IBDP Computer Science Internal Assessment

Appendix

Runxi Yu

October 12, 2024

Contents

1. Backend source code

1.1. config.go

```go
/*
 * Handle configuration
 *
 * Copyright (c) 2024  Runxi Yu <me@runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "bufio"
        "fmt"
        "os"

        "git.sr.ht/~emersion/go-scfg"
)

/*
 * We use two structs. The first has all of its values as pointers, and scfg
 * unmarshals the configuration to it. Then we take each value, dereference
 * it, and throw it into a normal config struct without pointers, reporting
 * missing values.
 * We should probably use reflection instead.
 */

var configWithPointers struct {
        URL    *string `scfg:"url"`
        Prod   *bool   `scfg:"prod"`
        Listen struct {
                Proto *string `scfg:"proto"`
                Net   *string `scfg:"net"`
                Addr  *string `scfg:"addr"`
                Trans *string `scfg:"trans"`
                TLS   struct {
                        Cert *string `scfg:"cert"`
                        Key  *string `scfg:"key"`
                } `scfg:"tls"`
        } `scfg:"listen"`
        DB struct {
                Type *string `scfg:"type"`
                Conn *string `scfg:"conn"`
        } `scfg:"db"`
        Auth struct {
                Client    *string `scfg:"client"`
                Authorize *string `scfg:"authorize"`
                Jwks      *string `scfg:"jwks"`
                Token     *string `scfg:"token"`
                Secret    *string `scfg:"secret"`
                Expr      *int    `scfg:"expr"`
        } `scfg:"auth"`
        Perf struct {
                SendQ               *int  `scfg:"sendq"`
                MessageArgumentsCap *int  `scfg:"msg_args_cap"`
                MessageBytesCap     *int  `scfg:"msg_bytes_cap"`
                ReadHeaderTimeout   *int  `scfg:"read_header_timeout"`
                UsemDelayShiftBits  *int  `scfg:"usem_delay_shift_bits"`
                PropagateImmediate  *bool `scfg:"propagate_immediate"`
        } `scfg:"perf"`
        Req struct {
                Y9 struct {
                        Sport    *int `scfg:"sport"`
                        NonSport *int `scfg:"non_sport"`
                } `scfg:"y9"`
                Y10 struct {
                        Sport    *int `scfg:"sport"`
                        NonSport *int `scfg:"non_sport"`
                } `scfg:"y10"`
                Y11 struct {
                        Sport    *int `scfg:"sport"`
                        NonSport *int `scfg:"non_sport"`
                } `scfg:"y11"`
                Y12 struct {
                        Sport    *int `scfg:"sport"`
                        NonSport *int `scfg:"non_sport"`
                } `scfg:"y12"`
        } `scfg:"req"`
}

var config struct {
        URL    string
        Prod   bool
        Listen struct {
                Proto string
                Net   string
                Addr  string
                Trans string
                TLS   struct {
                        Cert string
                        Key  string
                }
        }
        DB struct {
                Type string
                Conn string
        }
        Auth struct {
                Client    string
                Authorize string
                Jwks      string
                Token     string
                Secret    string
                Expr      int
        }
        Perf struct {
                SendQ               int
                MessageArgumentsCap int
                MessageBytesCap     int
                ReadHeaderTimeout   int
                UsemDelayShiftBits  int
                PropagateImmediate  bool
        }
        Req struct {
                Y9 struct {
                        Sport    int
                        NonSport int
                }
                Y10 struct {
                        Sport    int
                        NonSport int
                }
                Y11 struct {
                        Sport    int
                        NonSport int
                }
                Y12 struct {
                        Sport    int
                        NonSport int
                }
        }
}

func fetchConfig(path string) (retErr error) {
        defer func() {
                if retErr != nil {
                        retErr = wrapError(errCannotProcessConfig, retErr)
                }
```

```
}()

f, err := os.Open(path)
if err != nil {
        return wrapError(errCannotOpenConfig, err)
}

err = scfg.NewDecoder(bufio.NewReader(f)).Decode(&configWithPointers)
if err != nil {
        return wrapError(errCannotDecodeConfig, err)
}

if configWithPointers.URL == nil {
        return fmt.Errorf("%w: url", errMissingConfigValue)
}
config.URL = *(configWithPointers.URL)

if configWithPointers.Prod == nil {
        return fmt.Errorf("%w: prod", errMissingConfigValue)
}
config.Prod = *(configWithPointers.Prod)

if configWithPointers.Listen.Proto == nil {
        return fmt.Errorf("%w: listen.proto", errMissingConfigValue)
}
config.Listen.Proto = *(configWithPointers.Listen.Proto)

if configWithPointers.Listen.Net == nil {
        return fmt.Errorf("%w: listen.net", errMissingConfigValue)
}
config.Listen.Net = *(configWithPointers.Listen.Net)

if configWithPointers.Listen.Addr == nil {
        return fmt.Errorf("%w: listen.addr", errMissingConfigValue)
}
config.Listen.Addr = *(configWithPointers.Listen.Addr)

if configWithPointers.Listen.Trans == nil {
        return fmt.Errorf("%w: listen.trans", errMissingConfigValue)
}
config.Listen.Trans = *(configWithPointers.Listen.Trans)

if config.Listen.Trans == "tls" {
        if configWithPointers.Listen.TLS.Cert == nil {
                return fmt.Errorf(
                        "%w: listen.tls.cert",
                        errMissingConfigValue,
                )
        }
        config.Listen.TLS.Cert = *(configWithPointers.Listen.TLS.Cert)

        if configWithPointers.Listen.TLS.Key == nil {
                return fmt.Errorf(
                        "%w: listen.tls.key",
                        errMissingConfigValue,
                )
        }
        config.Listen.TLS.Key = *(configWithPointers.Listen.TLS.Key)
}

if configWithPointers.DB.Type == nil {
        return fmt.Errorf("%w: db.type", errMissingConfigValue)
}
config.DB.Type = *(configWithPointers.DB.Type)

if configWithPointers.DB.Conn == nil {
        return fmt.Errorf("%w: db.conn", errMissingConfigValue)
}
config.DB.Conn = *(configWithPointers.DB.Conn)

if configWithPointers.Auth.Client == nil {
        return fmt.Errorf("%w: auth.client", errMissingConfigValue)
}
config.Auth.Client = *(configWithPointers.Auth.Client)

if configWithPointers.Auth.Authorize == nil {
        return fmt.Errorf("%w: auth.authorize", errMissingConfigValue)
}
config.Auth.Authorize = *(configWithPointers.Auth.Authorize)

if configWithPointers.Auth.Jwks == nil {
        return fmt.Errorf("%w: auth.jwks", errMissingConfigValue)
}
config.Auth.Jwks = *(configWithPointers.Auth.Jwks)

if configWithPointers.Auth.Token == nil {
        return fmt.Errorf("%w: auth.token", errMissingConfigValue)
}
config.Auth.Token = *(configWithPointers.Auth.Token)

if configWithPointers.Auth.Secret == nil {
        return fmt.Errorf("%w: auth.secret", errMissingConfigValue)
}
config.Auth.Secret = *(configWithPointers.Auth.Secret)

if configWithPointers.Auth.Expr == nil {
        return fmt.Errorf("%w: auth.expr", errMissingConfigValue)
}
config.Auth.Expr = *(configWithPointers.Auth.Expr)

if configWithPointers.Perf.SendQ == nil {
        return fmt.Errorf(
                "%w: perf.sendq",
                errMissingConfigValue,
        )
}
```

```
config.Perf.SendQ = *(configWithPointers.Perf.SendQ)

if configWithPointers.Perf.MessageArgumentsCap == nil {
        return fmt.Errorf(
                "%w: perf.msg_args_cap",
                errMissingConfigValue,
        )
}
config.Perf.MessageArgumentsCap = *(configWithPointers.Perf.MessageArgumentsCap)

if configWithPointers.Perf.MessageBytesCap == nil {
        return fmt.Errorf(
                "%w: perf.msg_bytes_cap",
                errMissingConfigValue,
        )
}
config.Perf.MessageBytesCap = *(configWithPointers.Perf.MessageBytesCap)

if configWithPointers.Perf.ReadHeaderTimeout == nil {
        return fmt.Errorf(
                "%w: perf.read_header_timeout",
                errMissingConfigValue,
        )
}
config.Perf.ReadHeaderTimeout = *(configWithPointers.Perf.ReadHeaderTimeout)

if configWithPointers.Perf.UsemDelayShiftBits == nil {
        return fmt.Errorf(
                "%w: perf.usem_delay_shift_bits",
                errMissingConfigValue,
        )
}
config.Perf.UsemDelayShiftBits = *(configWithPointers.Perf.UsemDelayShiftBits)

if configWithPointers.Perf.PropagateImmediate == nil {
        return fmt.Errorf(
                "%w: perf.propagate_immediate",
                errMissingConfigValue,
        )
}
config.Perf.PropagateImmediate = *(configWithPointers.Perf.PropagateImmediate)

if configWithPointers.Req.Y9.Sport == nil {
        return fmt.Errorf(
                "%w: req.y9.sport",
                errMissingConfigValue,
        )
}
config.Req.Y9.Sport = *(configWithPointers.Req.Y9.Sport)
if configWithPointers.Req.Y9.NonSport == nil {
        return fmt.Errorf(
                "%w: req.y9.non_sport",
                errMissingConfigValue,
                )
        }
        config.Req.Y9.NonSport = *(configWithPointers.Req.Y9.NonSport)
        if configWithPointers.Req.Y10.Sport == nil {
                return fmt.Errorf(
                        "%w: req.y10.non_sport",
                        errMissingConfigValue,
                )
        }
        config.Req.Y10.Sport = *(configWithPointers.Req.Y10.Sport)
        if configWithPointers.Req.Y10.NonSport == nil {
                return fmt.Errorf(
                        "%w: req.y10.non_sport",
                        errMissingConfigValue,
                )
        }
        config.Req.Y10.NonSport = *(configWithPointers.Req.Y10.NonSport)
        if configWithPointers.Req.Y11.Sport == nil {
                return fmt.Errorf(
                        "%w: req.y11.sport",
                        errMissingConfigValue,
                )
        }
        config.Req.Y11.Sport = *(configWithPointers.Req.Y11.Sport)
        if configWithPointers.Req.Y11.NonSport == nil {
                return fmt.Errorf(
                        "%w: req.y11.non_sport",
                        errMissingConfigValue,
                )
        }
        config.Req.Y11.NonSport = *(configWithPointers.Req.Y11.NonSport)
        if configWithPointers.Req.Y12.Sport == nil {
                return fmt.Errorf(
                        "%w: req.y12.sport",
                        errMissingConfigValue,
                )
        }
        config.Req.Y12.Sport = *(configWithPointers.Req.Y12.Sport)
        if configWithPointers.Req.Y12.NonSport == nil {
                return fmt.Errorf(
                        "%w: req.y12.non_sport",
                        errMissingConfigValue,
                )
        }
        config.Req.Y12.NonSport = *(configWithPointers.Req.Y12.NonSport)

        return nil
}
```

1.2. confirm.go

```go
/*
 * Confirmation checking
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
)

func getConfirmedStatus(ctx context.Context, userID string) (confirmed bool, retErr error) {
        err := db.QueryRow(
                ctx,
                "SELECT confirmed FROM users WHERE id = $1",
                userID,
        ).Scan(&confirmed)
        if err != nil {
                retErr = wrapError(errUnexpectedDBError, err)
        }
        return
}
```

1.3. course_types_groups.go

```go
/*
 * Course types and groups
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "fmt"
)

/* Course types, e.g. Sport */

const (
        sport    string = "Sport"
        nonSport string = "Non-sport"
)

var courseTypes = map[string]struct{}{
        sport:    {},
        nonSport: {},
}

func checkCourseType(ct string) bool {
        _, ok := courseTypes[ct]
        return ok
}

type userCourseTypesT map[string]int

func getCourseTypeMinimumForYearGroup(yearGroup, courseType string) (int, error) {
        switch yearGroup {
        case "Y9":
                switch courseType {
                case sport:
                        return config.Req.Y9.Sport, nil
                case nonSport:
                        return config.Req.Y9.NonSport, nil
                default:
                        return 0, errInvalidCourseType
                }
        case "Y10":
                switch courseType {
                case sport:
                        return config.Req.Y10.Sport, nil
                case nonSport:
                        return config.Req.Y10.NonSport, nil
                default:
                        return 0, errInvalidCourseType
                }
        case "Y11":
                switch courseType {
                case sport:
                        return config.Req.Y11.Sport, nil
                case nonSport:
                        return config.Req.Y11.NonSport, nil
                default:
                        return 0, errInvalidCourseType
                }
        case "Y12":
                switch courseType {
                case sport:
                        return config.Req.Y12.Sport, nil
                case nonSport:
                        return config.Req.Y12.NonSport, nil
                default:
                        return 0, errInvalidCourseType
                }
        default:
                return 0, errNoSuchYearGroup
        }
}

/* Course groups, e.g. MW1 */

type userCourseGroupsT map[string]struct{}

func checkCourseGroup(cg string) bool {
        _, ok := courseGroups[cg]
        return ok
}

const (
        mw1 string = "MW1"
        mw2 string = "MW2"
        mw3 string = "MW3"
        tt1 string = "TT1"
        tt2 string = "TT2"
        tt3 string = "TT3"
)

var courseGroups = map[string]string{
        mw1: "Monday/Wednesday CCA1",
        mw2: "Monday/Wednesday CCA2",
        mw3: "Monday/Wednesday CCA3",
        tt1: "Tuesday/Thursday CCA1",
        tt2: "Tuesday/Thursday CCA2",
        tt3: "Tuesday/Thursday CCA3",
}
```

```go
/* Populate both */

func populateUserCourseTypesAndGroups(
        ctx context.Context,
        userCourseTypes *userCourseTypesT,
        userCourseGroups *userCourseGroupsT,
        userID string,
) error {
        rows, err := db.Query(
                ctx,
                "SELECT courseid FROM choices WHERE userid = $1",
                userID,
        )
        if err != nil {
                return wrapError(
                        errUnexpectedDBError,
                        err,
                )
        }
        for {
                if !rows.Next() {
                        err := rows.Err()
                        if err != nil {
                                return wrapError(
                                        errUnexpectedDBError,
                                        err,
                                )
                        }
                        break
                }
                var thisCourseID int
                err := rows.Scan(&thisCourseID)
                if err != nil {
                        return wrapError(
                                errUnexpectedDBError,
                                err,
                        )
                }
                var thisGroupName, thisTypeName string
                _course, ok := courses.Load(thisCourseID)
                if !ok {
                        return fmt.Errorf(
                                "%w: %d",
                                errNoSuchCourse,
                                thisCourseID,
                        )
                }
                course, ok := _course.(*courseT)
                if !ok {
                        panic("courses map has non-\"*courseT\" items")
                }
                thisGroupName = course.Group
                thisTypeName = course.Type
                if _, ok := (*userCourseGroups)[thisGroupName]; ok {
                        return fmt.Errorf(
                                "%w: user %v, group %v",
                                errMultipleChoicesInOneGroup,
                                userID,
                                thisGroupName,
                        )
                }
                (*userCourseGroups)[thisGroupName] = struct{}{}
                (*userCourseTypes)[thisTypeName]++
        }
        return nil
}
```

1.4. courses.go

```
/*
 * Course data structures and locking
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "fmt"
        "sync"
        "sync/atomic"

        "github.com/coder/websocket"
)

type courseT struct {
        /*
         * Selected is usually accessed atomically, but a lock is still
         * necessary as we need to sequentialize compare-with-Max-and-increment
         * operations.
         * We put Selected before other values to ensure 64-bit alignment on
         * all systems, because it needs to be accessed atomically. See the
         * "Bugs" section of sync/atomic.
         */
        Selected     uint32 /* atomic */
        SelectedLock sync.Mutex
        ID           int
        Max          uint32
        Title        string
        Type         string
        Group        string
        Teacher      string
        Location     string
        CourseID     string
        SectionID    string
        Usems        sync.Map /* string, *usemT */
}

var courses sync.Map /* int, *courseT */

var numCourses uint32 /* atomic */

const staffDepartment = "Staff"

/*
 * Read course information from the database. This should be called during
 * setup.
 */
func setupCourses(ctx context.Context) error {
        rows, err := db.Query(
                ctx,
                "SELECT id, nmax, title, ctype, cgroup, teacher, location, course_id,
                 ↪  section_id FROM courses",
        )
        if err != nil {
                return wrapError(errUnexpectedDBError, err)
        }

        for {
                if !rows.Next() {
                        err := rows.Err()
                        if err != nil {
                                return wrapError(
                                        errUnexpectedDBError,
                                        err,
                                )
                        }
                        break
                }
                currentCourse := courseT{} //exhaustruct:ignore
                err = rows.Scan(
                        &currentCourse.ID,
			&currentCourse.Max,
			&currentCourse.Title,
			&currentCourse.Type,
			&currentCourse.Group,
			&currentCourse.Teacher,
			&currentCourse.Location,
			&currentCourse.CourseID,
			&currentCourse.SectionID,
		)
		if err != nil {
			return wrapError(errUnexpectedDBError, err)
		}
		if !checkCourseType(currentCourse.Type) {
			return fmt.Errorf(
				"%w: %d %s",
				errInvalidCourseType,
				currentCourse.ID,
				currentCourse.Type,
			)
		}
		if !checkCourseGroup(currentCourse.Group) {
			return fmt.Errorf(
				"%w: %d %s",
				errInvalidCourseGroup,
				currentCourse.ID,
				currentCourse.Group,
			)
		}
		err := db.QueryRow(
			ctx,
			"SELECT COUNT (*) FROM choices WHERE courseid = $1",
			currentCourse.ID,
		).Scan(&currentCourse.Selected)
		if err != nil {
			return wrapError(
				errUnexpectedDBError,
				err,
			)
		}
		courses.Store(currentCourse.ID, &currentCourse)
		atomic.AddUint32(&numCourses, 1)
	}

	return nil
}

func (course *courseT) decrementSelectedAndPropagate(
	ctx context.Context,
	conn *websocket.Conn,
) error {
	func() {
		course.SelectedLock.Lock()
		defer course.SelectedLock.Unlock()
```

```
                atomic.AddUint32(&course.Selected, ^uint32(0))
        }()
        go propagateSelectedUpdate(course)
        err := sendSelectedUpdate(ctx, conn, course.ID)
        if err != nil {
                return wrapError(
                        errCannotSend,
                        err,
                )
        }
        return nil
}
```

1.5. database.go

```
/*
 * Database handling
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"

        "github.com/jackc/pgx/v5/pgxpool"
)

var db *pgxpool.Pool

const pgErrUniqueViolation = "23505"

/*
 * This must be run during setup, before the database is accessed by any
 * means. Otherwise, db would be a null pointer.
 */
func setupDatabase() error {
```

```go
        var err error
        if config.DB.Type != "postgres" {
                return errUnsupportedDatabaseType
        }
        db, err = pgxpool.New(context.Background(), config.DB.Conn)
        if err != nil {
                return wrapError(errUnexpectedDBError, err)
        }
        return nil
}
```

1.6. endpoint_auth.go

```go
/*
 * Custom OAUTH 2.0 implementation for the CCA Selection Service
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "encoding/json"
        "errors"
        "fmt"
        "net/http"
        "net/url"
        "strings"
        "time"

        "github.com/MicahParks/keyfunc/v3"
        "github.com/golang-jwt/jwt/v5"
        "github.com/jackc/pgx/v5/pgconn"
)

var myKeyfunc keyfunc.Keyfunc
```

```go
const tokenLength = 20

/*
 * These are the claims in the JSON Web Token received from the client, after
 * it redirects from the authorize endpoint. Some of these fields must be
 * explicitly selected in the Azure app registration and might appear as
 * zero strings if it hasn't been configured correctly.
 */
type msclaimsT struct {
        Name  string `json:"name"`  /* Scope: profile */
        Email string `json:"email"` /* Scope: email   */
        Oid   string `json:"oid"`   /* Scope: profile */
        jwt.RegisteredClaims
}

func generateAuthorizationURL() (string, error) {
        nonce, err := randomString(tokenLength)
        if err != nil {
                return "", err
        }
        /*
         * Note that here we use a hybrid authentication flow to obtain an
         * id_token for authentication and an authorization code. The
         * authorization code may be used like any other; i.e., it may be used
         * to obtain an access token directly, or the refresh token may be used
         * to gain persistent access to the upstream API. Sometimes I wish that
         * the JWT in id_token could have more claims. The only reason we
         * presently use a hybrid flow is to use the authorization code to
         * obtain an access code to call the user info endpoint to fetch the
         * user's department information.
         */
        return fmt.Sprintf(
                "https://login.microsoftonline.com/ddd3d26c-b197-4d00-a32d-1ffd84c0c295/oauth2/authoriz
                config.Auth.Client,
                config.URL,
                nonce,
        ), nil
}

/*
 * Handles redirects to the /auth endpoint from the authorize endpoint.
 * Expects JSON Web Keys to be already set up correctly; if myKeyfunc is null,
 * a null pointer is dereferenced and the thread panics.
 */
func handleAuth(w http.ResponseWriter, req *http.Request) (string, int, error) {
        if req.Method != http.MethodPost {
                return "", http.StatusMethodNotAllowed, errPostOnly
        }

        err := req.ParseForm()
        if err != nil {
                return "", http.StatusBadRequest, wrapError(errMalformedForm, err)
        }
```

```
returnedError := req.PostFormValue("error")
if returnedError != "" {
        returnedErrorDescription := req.PostFormValue("error_description")
        return "", http.StatusUnauthorized, wrapAny(
                errAuthorizeEndpointError,
                returnedError+": "+returnedErrorDescription,
        )
}

idTokenString := req.PostFormValue("id_token")
if idTokenString == "" {
        return "", http.StatusBadRequest, wrapAny(
                errInsufficientFields,
                "id_token",
        )
}

claimsTemplate := &msclaimsT{} //exhaustruct:ignore
token, err := jwt.ParseWithClaims(
        idTokenString,
        claimsTemplate,
        myKeyfunc.Keyfunc,
)
if err != nil {
        return "", http.StatusBadRequest, wrapError(
                errCannotParseClaims,
                err,
        )
}

switch {
case token.Valid:
        break
case errors.Is(err, jwt.ErrTokenMalformed):
        return "", http.StatusBadRequest, wrapError(
                errJWTMalformed,
                err,
        )
case errors.Is(err, jwt.ErrTokenSignatureInvalid):
        return "", http.StatusBadRequest, wrapError(
                errJWTSignatureInvalid,
                err,
        )
case errors.Is(err, jwt.ErrTokenExpired) ||
        errors.Is(err, jwt.ErrTokenNotValidYet):
        return "", http.StatusBadRequest, wrapError(
                errJWTExpired,
                err,
        )
default:
        return "", http.StatusBadRequest, wrapError(
                errJWTInvalid,
                err,
        )
}

claims, claimsOk := token.Claims.(*msclaimsT)

if !claimsOk {
        return "", http.StatusBadRequest, errCannotUnpackClaims
}

authorizationCode := req.PostFormValue("code")

accessToken, err := getAccessToken(req.Context(), authorizationCode)
if err != nil {
        return "", -1, err
}

department, err := getDepartment(req.Context(), *(accessToken.Content))
if err != nil {
        return "", -1, err
}

switch {
case department == "SJ Co-Curricular Activities Office □□□□□□□□□":
        department = staffDepartment
case department == "Y9" || department == "Y10" ||
        department == "Y11" || department == "Y12":
default:
        return "", http.StatusForbidden, errUnknownDepartment
}

cookieValue, err := randomString(tokenLength)
if err != nil {
        return "", -1, err
}

now := time.Now()
expr := now.Add(time.Duration(config.Auth.Expr) * time.Second)
exprU := expr.Unix()

cookie := http.Cookie{
        Name:     "session",
        Value:    cookieValue,
        SameSite: http.SameSiteLaxMode,
        HttpOnly: true,
        Secure:   config.Prod,
        Expires:  expr,
} //exhaustruct:ignore

http.SetCookie(w, &cookie)

_, err = db.Exec(
        req.Context(),
                "INSERT INTO users (id, name, email, department, session, expr, confirmed)
                 ↪  VALUES ($1, $2, $3, $4, $5, $6, false)",
                claims.Oid,
                claims.Name,
                claims.Email,
                department,
                cookieValue,
                exprU,
        )
        if err != nil {
                var pgErr *pgconn.PgError
                if errors.As(err, &pgErr) && pgErr.Code == pgErrUniqueViolation {
                        _, err := db.Exec(
                                req.Context(),
                                "UPDATE users SET (name, email, department, session, expr) =
                                 ↪  ($1, $2, $3, $4, $5) WHERE id = $6",
                                claims.Name,
                                claims.Email,
                                department,
                                cookieValue,
                                exprU,
                                claims.Oid,
                        )
                        if err != nil {
                                return "", -1, wrapError(errUnexpectedDBError, err)
                        }
                } else {
                        return "", -1, wrapError(errUnexpectedDBError, err)
                }
        }

        http.Redirect(w, req, "/", http.StatusSeeOther)

        return "", -1, nil
}

func setupJwks() error {
        var err error
        myKeyfunc, err = keyfunc.NewDefault([]string{config.Auth.Jwks})
        if err != nil {
                return wrapError(errCannotSetupJwks, err)
        }
        return nil
}

/*
 * Fetch the department name of the user, mostly to identify which grade
 * a student is in. This expects an accessToken obtained from the OAUTH 2.0
 * token endpoint obtained via an authorization code. It might also be able
 * to use this as part of a hybrid flow that directly provides access tokens,
 * but this flow seems to be only usable for single-page applications according
 * to the Azure portal.
 */
```

```go
func getDepartment(ctx context.Context, accessToken string) (string, error) {
	req, err := http.NewRequestWithContext(
		ctx,
		http.MethodGet,
		"https://graph.microsoft.com/v1.0/me?$select=department",
		nil,
	)
	if err != nil {
		return "", wrapError(errCannotGetDepartment, err)
	}
	req.Header.Set("Authorization", "Bearer "+accessToken)

	client := &http.Client{} //exhaustruct:ignore
	resp, err := client.Do(req)
	if err != nil {
		return "", wrapError(errCannotGetDepartment, err)
	}
	defer resp.Body.Close()

	var departmentWrap struct {
		Department *string `json:"department"`
	}

	decoder := json.NewDecoder(resp.Body)
	err = decoder.Decode(&departmentWrap)
	if err != nil {
		return "", wrapError(errCannotGetDepartment, err)
	}

	if departmentWrap.Department == nil {
		/*
		 * This is probably because the response does not contain a
		 * "department" field, which hopefully doesn't occur as we
		 * have specified $select=department in the OData query.
		 */
		return "", wrapError(
			errCannotGetDepartment,
			errInsufficientFields,
		)
	}

	return *(departmentWrap.Department), nil
}

type accessTokenT struct {
	Content          *string `json:"access_token"`
	Error            *string `json:"error"`
	ErrorDescription *string `json:"error_description"`
	ErrorCodes       *[]int  `json:"error_codes"`
}

func getAccessToken(
	ctx context.Context,
        authorizationCode string,
) (accessTokenT, error) {
        var accessToken accessTokenT
        v := url.Values{}
        v.Set("client_id", config.Auth.Client)
        v.Set("scope", "https://graph.microsoft.com/User.Read")
        v.Set("code", authorizationCode)
        v.Set("redirect_uri", config.URL+"/auth")
        v.Set("grant_type", "authorization_code")
        v.Set("client_secret", config.Auth.Secret)
        req, err := http.NewRequestWithContext(
                ctx,
                http.MethodPost,
                config.Auth.Token,
                strings.NewReader(v.Encode()),
        )
        if err != nil {
                return accessToken,
                        wrapError(errCannotFetchAccessToken, err)
        }
        resp, err := http.DefaultClient.Do(req)
        if err != nil {
                return accessToken,
                        wrapError(errCannotFetchAccessToken, err)
        }
        defer resp.Body.Close()

        decoder := json.NewDecoder(resp.Body)
        err = decoder.Decode(&accessToken)
        if err != nil {
                return accessToken,
                        wrapError(errCannotFetchAccessToken, err)
        }
        if accessToken.Error != nil || accessToken.ErrorCodes != nil ||
                accessToken.ErrorDescription != nil {
                if accessToken.Error == nil || accessToken.ErrorCodes == nil ||
                        accessToken.ErrorDescription == nil {
                        return accessToken, errCannotFetchAccessToken
                }
                return accessToken,
                        fmt.Errorf(
                                "%w: %v",
                                errTokenEndpointReturnedError,
                                *accessToken.ErrorDescription,
                        )
        }
        if accessToken.Content == nil {
                return accessToken,
                        fmt.Errorf(
                                "error extracting access token: %w",
                                errInsufficientFields,
                        )
        }
```

```
	return accessToken, nil
}
```

1.7. endpoint_export_choices.go

```
/*
 * Staff page
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
	"encoding/csv"
	"net/http"
	"strings"
)

func handleExportChoices(w http.ResponseWriter, req *http.Request) (string, int, error) {
	_, _, department, err := getUserInfoFromRequest(req)
	if err != nil {
		return "", -1, err
	}
	if department != staffDepartment {
		return "", http.StatusForbidden, errStaffOnly
	}

	type userCacheT struct {
		Name       string
		StudentID  string
		Department string
	}
	userCacheMap := make(map[string]userCacheT)

	rows, err := db.Query(req.Context(), "SELECT userid, courseid FROM choices")
	if err != nil {
```

```
                return "", -1, wrapError(errUnexpectedDBError, err)
        }
        output := make([][]string, 0)
        for {
                if !rows.Next() {
                        err := rows.Err()
                        if err != nil {
                                return "", -1, wrapError(errUnexpectedDBError, err)
                        }
                        break
                }
                var currentUserID, currentUserName, currentStudentID, currentDepartment string
                var currentCourseID int
                err := rows.Scan(&currentUserID, &currentCourseID)
                if err != nil {
                        return "", -1, wrapError(errUnexpectedDBError, err)
                }
                currentUserCache, ok := userCacheMap[currentUserID]
                if ok {
                        currentUserName = currentUserCache.Name
                        currentDepartment = currentUserCache.Department
                        currentStudentID = currentUserCache.StudentID
                } else {
                        var currentUserEmail string
                        err := db.QueryRow(
                                req.Context(),
                                "SELECT name, email, department FROM users WHERE id = $1",
                                currentUserID,
                        ).Scan(
                                &currentUserName,
                                &currentUserEmail,
                                &currentDepartment,
                        )
                        if err != nil {
                                return "", -1, wrapError(errUnexpectedDBError, err)
                        }
                        before, _, found := strings.Cut(currentUserEmail, "@")
                        if found {
                                currentStudentID, _ = strings.CutPrefix(before, "s")
                        } else {
                                currentStudentID = currentUserEmail
                        }
                        userCacheMap[currentUserID] = userCacheT{
                                Name:       currentUserName,
                                StudentID:  currentStudentID,
                                Department: currentDepartment,
                        }
                }

                _course, ok := courses.Load(currentCourseID)
                if !ok {
                        return "", -1, wrapAny(errNoSuchCourse, currentCourseID)
                }
```

```
		course, ok := _course.(*courseT)
		if !ok {
			panic("courses map has non-\"*courseT\" items")
		}
		if course == nil {
			return "", -1, wrapAny(errNoSuchCourse, currentCourseID)
		}
		output = append(
			output,
			[]string{
				currentUserName,
				currentStudentID,
				currentDepartment,
				course.Title,
				course.Group,
				course.SectionID,
				course.CourseID,
			},
		)
	}

	w.Header().Set("Content-Type", "text/csv; charset=utf-8")
	w.Header().Set("Content-Disposition", "attachment;filename=cca_choices.csv")
	csvWriter := csv.NewWriter(w)
	err = csvWriter.Write([]string{
		"Student Name",
		"Student ID",
		"Grade/Year",
		"Group/Activity",
		"Container",
		"Section ID",
		"Course ID",
	})
	if err != nil {
		return "", -1, wrapError(errHTTPWrite, err)
	}
	err = csvWriter.WriteAll(output)
	if err != nil {
		return "", -1, wrapError(errHTTPWrite, err)
	}
	csvWriter.Flush()
	if csvWriter.Error() != nil {
		return "", -1, wrapError(errHTTPWrite, err)
	}
	return "", -1, nil
}
```

1.8. endpoint_export_students.go

```
/*
 * Staff page
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "encoding/csv"
        "net/http"
        "strconv"
)

func handleExportStudents(w http.ResponseWriter, req *http.Request) (string, int, error) {
        _, _, department, err := getUserInfoFromRequest(req)
        if err != nil {
                return "", -1, err
        }
        if department != staffDepartment {
                return "", -1, errStaffOnly
        }

        rows, err := db.Query(req.Context(), "SELECT name, email, department, confirmed FROM
         ↪  users")
        if err != nil {
                return "", -1, wrapError(errUnexpectedDBError, err)
        }
        output := make([][]string, 0)
        for {
                if !rows.Next() {
                        err := rows.Err()
                        if err != nil {
                                return "", -1, wrapError(errUnexpectedDBError, err)
                        }
                        break
                }
                var currentUserName, currentEmail, currentDepartment string
                var currentConfirmed bool
                err := rows.Scan(
                        &currentUserName,
                        &currentEmail,
```

```go
			&currentDepartment,
			&currentConfirmed,
		)
		if err != nil {
			return "", -1, wrapError(errUnexpectedDBError, err)
		}

		if currentDepartment == staffDepartment {
			continue
		}

		output = append(
			output,
			[]string{
				currentUserName,
				currentEmail,
				currentDepartment,
				strconv.FormatBool(currentConfirmed),
			},
		)
	}

	w.Header().Set("Content-Type", "text/csv; charset=utf-8")
	w.Header().Set("Content-Disposition", "attachment;filename=cca_students.csv")
	csvWriter := csv.NewWriter(w)
	err = csvWriter.Write([]string{
		"Student Name",
		"Student ID",
		"Grade/Year",
		"Group/Activity",
		"Container",
		"Section ID",
		"Course ID",
	})
	if err != nil {
		return "", -1, errHTTPWrite
	}
	err = csvWriter.WriteAll(output)
	if err != nil {
		return "", -1, errHTTPWrite
	}
	csvWriter.Flush()
	if csvWriter.Error() != nil {
		return "", -1, errHTTPWrite
	}

	return "", -1, nil
}
```

1.9. endpoint_index.go

```go
/*
 * Index page
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
	"errors"
	"log"
	"net/http"
	"sync/atomic"
)

func handleIndex(w http.ResponseWriter, req *http.Request) (string, int, error) {
	_, username, department, err := getUserInfoFromRequest(req)
	if errors.Is(err, errNoCookie) || errors.Is(err, errNoSuchUser) {
		authURL, err2 := generateAuthorizationURL()
		if err2 != nil {
			return "", -1, err2
		}
		var noteString string
		if errors.Is(err, errNoSuchUser) {
			noteString = "Your browser provided an invalid session cookie."
		}
		err2 = tmpl.ExecuteTemplate(
			w,
			"login",
			struct {
				AuthURL string
				Notes   string
			}{
				authURL,
				noteString,
			},
		)
```

```
        if err2 != nil {
                log.Println(err2)
                return "", -1, wrapError(errCannotWriteTemplate, err2)
        }
        return "", -1, nil
} else if err != nil {
        return "", -1, err
}

/* TODO: The below should be completed on-update. */
type groupT struct {
        Handle  string
        Name    string
        Courses *map[int]*courseT
}
_groups := make(map[string]groupT)
for k, v := range courseGroups {
        _coursemap := make(map[int]*courseT)
        _groups[k] = groupT{
                Handle:  k,
                Name:    v,
                Courses: &_coursemap,
        }
}
courses.Range(func(key, value interface{}) bool {
        courseID, ok := key.(int)
        if !ok {
                panic("courses map has non-\"int\" keys")
        }
        course, ok := value.(*courseT)
        if !ok {
                panic("courses map has non-\"*courseT\" items")
        }
        (*_groups[course.Group].Courses)[courseID] = course
        return true
})

if department == staffDepartment {
        err := tmpl.ExecuteTemplate(
                w,
                "staff",
                struct {
                        Name   string
                        State  uint32
                        Groups *map[string]groupT
                }{
                        username,
                        state,
                        &_groups,
                },
        )
        if err != nil {
                return "", -1, wrapError(errCannotWriteTemplate, err)
        }
        return "", -1, nil
}

if atomic.LoadUint32(&state) == 0 {
        err := tmpl.ExecuteTemplate(
                w,
                "student_disabled",
                struct {
                        Name       string
                        Department string
                }{
                        username,
                        department,
                },
        )
        if err != nil {
                return "", -1, wrapError(errCannotWriteTemplate, err)
        }
        return "", -1, nil
}
sportRequired, err := getCourseTypeMinimumForYearGroup(department, sport)
if err != nil {
        return "", -1, err
}
nonSportRequired, err := getCourseTypeMinimumForYearGroup(department, nonSport)
if err != nil {
        return "", -1, err
}

err = tmpl.ExecuteTemplate(
        w,
        "student",
        struct {
                Name       string
                Department string
                Groups     *map[string]groupT
                Required   struct {
                        Sport    int
                        NonSport int
                }
        }{
                username,
                department,
                &_groups,
                struct {
                        Sport    int
                        NonSport int
                }{sportRequired, nonSportRequired},
        },
)
if err != nil {
        return "", -1, wrapError(errCannotWriteTemplate, err)
```

```
	}
	return "", -1, nil
}
```

1.10. endpoint_newcourses.go

```
/*
 * Overwrite courses with uploaded CSV
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
	"context"
	"encoding/csv"
	"errors"
	"fmt"
	"io"
	"net/http"
	"strings"
	"sync/atomic"

	"github.com/jackc/pgx/v5"
)

func handleNewCourses(w http.ResponseWriter, req *http.Request) (string, int, error) {
	if req.Method != http.MethodPost {
		return "", http.StatusMethodNotAllowed, errPostOnly
	}

	_, _, department, err := getUserInfoFromRequest(req)
	if err != nil {
		return "", -1, err
	}
	if department != staffDepartment {
		return "", http.StatusForbidden, errStaffOnly
```

```
}

if atomic.LoadUint32(&state) != 0 {
        return "", http.StatusBadRequest, errDisableStudentAccessFirst
}

/* TODO: Potential race. The global state may need to be write-locked. */

file, fileHeader, err := req.FormFile("coursecsv")
if err != nil {
        return "", http.StatusBadRequest, wrapError(errFormNoFile, err)
}

if fileHeader.Header.Get("Content-Type") != "text/csv" {
        return "", http.StatusBadRequest, errNotACSV
}

csvReader := csv.NewReader(file)
titleLine, err := csvReader.Read()
if err != nil {
        return "", http.StatusBadRequest, wrapError(errCannotReadCSV, err)
}
if titleLine == nil {
        return "", -1, errUnexpectedNilCSVLine
}
if len(titleLine) != 8 {
        return "", -1, wrapAny(errBadCSVFormat, "expecting 8 fields on the first line")
}
var titleIndex, maxIndex, teacherIndex, locationIndex,
        typeIndex, groupIndex, sectionIDIndex,
        courseIDIndex int = -1, -1, -1, -1, -1, -1, -1, -1
for i, v := range titleLine {
        switch v {
        case "Title":
                titleIndex = i
        case "Max":
                maxIndex = i
        case "Teacher":
                teacherIndex = i
        case "Location":
                locationIndex = i
        case "Type":
                typeIndex = i
        case "Group":
                groupIndex = i
        case "Section ID":
                sectionIDIndex = i
        case "Course ID":
                courseIDIndex = i
        }
}

if titleIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Title")
}
if maxIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Max")
}
if teacherIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Teacher")
}
if locationIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Location")
}
if typeIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Type")
}
if groupIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Group")
}
if courseIDIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Course ID")
}
if sectionIDIndex == -1 {
        return "", http.StatusBadRequest, wrapAny(errMissingCSVColumn, "Section ID")
}

lineNumber := 1
ok, statusCode, err := func(ctx context.Context) (retBool bool, retStatus int, retErr
 ↪  error) {
        tx, err := db.Begin(ctx)
        if err != nil {
                return false, -1, wrapError(errUnexpectedDBError, err)
        }
        defer func() {
                err := tx.Rollback(ctx)
                if err != nil && (!errors.Is(err, pgx.ErrTxClosed)) {
                        retBool, retStatus, retErr = false, -1,
                         ↪  wrapError(errUnexpectedDBError, err)
                        return
                }
        }()
        _, err = tx.Exec(
                ctx,
                "DELETE FROM choices",
        )
        if err != nil {
                return false, -1, wrapError(errUnexpectedDBError, err)
        }
        _, err = tx.Exec(
                ctx,
                "DELETE FROM courses",
        )
        if err != nil {
                return false, -1, wrapError(errUnexpectedDBError, err)
        }
```

```go
for {
	lineNumber++
	line, err := csvReader.Read()
	if err != nil {
		if errors.Is(err, io.EOF) {
			break
		}
		return false, -1, wrapError(errCannotReadCSV, err)
	}
	if line == nil {
		return false, -1, wrapError(errCannotReadCSV,
		 ↪  errUnexpectedNilCSVLine)
	}
	if len(line) != 8 {
		return false, -1, wrapAny(errInsufficientFields, fmt.Sprintf(
			"line %d has insufficient items",
			lineNumber,
		))
	}
	if !checkCourseType(line[typeIndex]) {
		return false, -1, wrapAny(errInvalidCourseType,
			fmt.Sprintf(
				"line %d has invalid course type
				 ↪  \"%s\"\nallowed course types: %s",
				lineNumber,
				line[typeIndex],
				strings.Join(getKeysOfMap(courseTypes), ", "),
			),
		)
	}
	if !checkCourseGroup(line[groupIndex]) {
		return false, -1, wrapAny(errInvalidCourseGroup,
			fmt.Sprintf(
				"line %d has invalid course group
				 ↪  \"%s\"\nallowed course groups: %s",
				lineNumber,
				line[groupIndex],
				strings.Join(getKeysOfMap(courseGroups), ", "),
			),
		)
	}
	_, err = tx.Exec(
		ctx,
		"INSERT INTO courses(nmax, title, teacher, location, ctype,
		 ↪  cgroup, section_id, course_id) VALUES ($1, $2, $3, $4, $5,
		 ↪  $6, $7, $8)",
		line[maxIndex],
		line[titleIndex],
		line[teacherIndex],
		line[locationIndex],
		line[typeIndex],
		line[groupIndex],
				line[sectionIDIndex],
				line[courseIDIndex],
			)
			if err != nil {
				return false, -1, wrapError(errUnexpectedDBError, err)
			}
		}
		err = tx.Commit(ctx)
		if err != nil {
			return false, -1, wrapError(errUnexpectedDBError, err)
		}
		return true, -1, nil
	}(req.Context())
	if !ok {
		return "", statusCode, err
	}

	courses.Clear()
	err = setupCourses(req.Context())
	if err != nil {
		return "", -1, wrapError(errWhileSetttingUpCourseTablesAgain, err)
	}

	http.Redirect(w, req, "/", http.StatusSeeOther)

	return "", -1, nil
}
```

1.11. endpoint_state.go

```
/*
 * Let staff update state
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main
```

```go
import (
        "net/http"
        "strconv"
)

func handleState(w http.ResponseWriter, req *http.Request) (string, int, error) {
        _, _, department, err := getUserInfoFromRequest(req)
        if err != nil {
                return "", http.StatusUnauthorized, err
        }
        if department != staffDepartment {
                return "", http.StatusForbidden, errStaffOnly
        }

        basePath := req.PathValue("s")
        newState, err := strconv.ParseUint(basePath, 10, 32)
        if err != nil {
                return "", http.StatusBadRequest, wrapError(errInvalidState, err)
        }
        err = setState(req.Context(), uint32(newState))
        if err != nil {
                return "", http.StatusBadRequest, wrapError(errCannotSetState, err)
        }

        http.Redirect(w, req, "/", http.StatusSeeOther)
        return "", -1, nil
}
```

1.12. endpoint_ws.go

```go
/*
 * WebSocket endpoint handler
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main
```

```
import (
        "log"
        "net/http"

        "github.com/coder/websocket"
)

/*
 * Handle requests to the WebSocket endpoint and establish a connection.
 * Authentication is handled here, but afterwards, the connection is really
 * handled in handleConn.
 */
func handleWs(w http.ResponseWriter, req *http.Request) {
        wsOptions := &websocket.AcceptOptions{
                Subprotocols: []string{"cca1"},
        } //exhaustruct:ignore
        c, err := websocket.Accept(
                w,
                req,
                wsOptions,
        )
        if err != nil {
                wstr(
                        w,
                        http.StatusBadRequest,
                        "this endpoint only supports valid WebSocket connections:
                         ↪   "+err.Error(),
                )
                return
        }
        defer func() {
                _ = c.CloseNow()
        }()

        userID, _, department, err := getUserInfoFromRequest(req)
        if err != nil {
                err := writeText(req.Context(), c, "U")
                if err != nil {
                        log.Println(err)
                }
                return
        }

        err = handleConn(req.Context(), c, userID, department)
        if err != nil {
                err := writeText(req.Context(), c, "E :"+err.Error())
                if err != nil {
                        log.Println(err)
                }
                return
        }
}
```

1.13. errors.go

```go
/*
 * Error definitions
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "errors"
        "fmt"
)

var (
        errCannotSetupJwks                  = errors.New("cannot set up jwks")
        errInsufficientFields               = errors.New("insufficient fields")
        errCannotGetDepartment              = errors.New("cannot get department")
        errUnknownDepartment                = errors.New("unknown department")
        errCannotFetchAccessToken           = errors.New("cannot fetch access token")
        errTokenEndpointReturnedError       = errors.New("token endpoint returned error")
        errCannotProcessConfig              = errors.New("cannot process configuration file")
        errCannotOpenConfig                 = errors.New("cannot open configuration file")
        errCannotDecodeConfig               = errors.New("cannot decode configuration file")
        errMissingConfigValue               = errors.New("missing configuration value")
        errInvalidCourseType                = errors.New("invalid course type")
        errInvalidCourseGroup               = errors.New("invalid course group")
        errMultipleChoicesInOneGroup        = errors.New("multiple choices per group per user")
        errUnsupportedDatabaseType          = errors.New("unsupported db type")
        errUnexpectedDBError                = errors.New("unexpected database error")
        errCannotSend                       = errors.New("cannot send")
        errCannotGenerateRandomString       = errors.New("cannot generate random string")
        errContextCanceled                  = errors.New("context canceled")
        errCannotReceiveMessage             = errors.New("cannot receive message")
        errNoSuchCourse                     = errors.New("reference to non-existent course")
        errInvalidState                     = errors.New("invalid state")
        errCannotSetState                   = errors.New("cannot set state")
        errWebSocketWrite                   = errors.New("error writing to websocket")
        errHTTPWrite                        = errors.New("error writing to http writer")
        errCannotCheckCookie                = errors.New("error checking cookie")
        errNoCookie                         = errors.New("no cookie found")
        errNoSuchUser                       = errors.New("no such user")
        errNoSuchYearGroup                  = errors.New("no such year group")
        errPostOnly                         = errors.New("only post is supported on this
         ↪  endpoint")
        errMalformedForm                    = errors.New("malformed form")
        errAuthorizeEndpointError           = errors.New("authorize endpoint returned error")
        errCannotParseClaims                = errors.New("cannot parse claims")
        errCannotUnpackClaims               = errors.New("cannot unpack claims")
        errJWTMalformed                     = errors.New("jwt token is malformed")
        errJWTSignatureInvalid              = errors.New("jwt token has invalid signature")
        errJWTExpired                       = errors.New("jwt token has expired or is not yet
         ↪  valid")
        errJWTInvalid                       = errors.New("jwt token is somehow invalid")
        errStaffOnly                        = errors.New("this page is only available to
         ↪  staff")
        errDisableStudentAccessFirst        = errors.New("you must disable student access
         ↪  before performing this operation")
        errFormNoFile                       = errors.New("you need to select a file before
         ↪  submitting the form")
        errNotACSV                          = errors.New("the file you uploaded is not a csv
         ↪  file")
        errCannotReadCSV                    = errors.New("cannot read csv")
        errBadCSVFormat                     = errors.New("bad csv format")
        errMissingCSVColumn                 = errors.New("missing csv column")
        errUnexpectedNilCSVLine             = errors.New("unexpected nil csv line")
        errWhileSetttingUpCourseTablesAgain = errors.New("error while setting up course tables
         ↪  again")
        errCannotWriteTemplate              = errors.New("cannot write template")
        // errInvalidCourseID                  = errors.New("invalid course id")
)

func wrapError(a, b error) error {
        if a == nil && b == nil {
                return nil
        }
        return fmt.Errorf("%w: %w", a, b)
}

func wrapAny(a error, b any) error {
        if a == nil && b == nil {
                return nil
        }
        return fmt.Errorf("%w: %v", a, b)
}
```

1.14. main.go

```go
/*
 * Main listener
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "crypto/tls"
        "embed"
        "flag"
        "html/template"
        "io/fs"
        "log"
        "net"
        "net/http"
        "time"
)

var tmpl *template.Template

//go:embed build/static/* templates/*
//go:embed build/iadocs/*.pdf build/iadocs/*.htm build/iadocs/*.html
//go:embed build/docs/*
var runFS embed.FS

//go:embed go.* *.go
//go:embed docs/* iadocs/*
//go:embed frontend/* templates/*
//go:embed README.md LICENSE Makefile .editorconfig .gitignore .gitattributes
//go:embed scripts/* sql/*
var srcFS embed.FS

func main() {
        var err error

        var configPath string

        flag.StringVar(
                &configPath,
                "c",
                "cca.scfg",
                "path to configuration file",
        )
        flag.Parse()

        if err := fetchConfig(configPath); err != nil {
                log.Fatal(err)
        }

        log.Println("Setting up templates")
        tmpl, err = template.ParseFS(runFS, "templates/*")
        if err != nil {
                log.Fatal(err)
        }

        log.Println("Registering static handle")
        staticFS, err := fs.Sub(runFS, "build/static")
        if err != nil {
                log.Fatal(err)
        }
        http.Handle("/static/",
                http.StripPrefix(
                        "/static/",
                        http.FileServer(http.FS(staticFS)),
                ),
        )

        log.Println("Registering iadocs handle")
        iaDocsFS, err := fs.Sub(runFS, "build/iadocs")
        if err != nil {
                log.Fatal(err)
        }
        http.Handle("/iadocs/",
                http.StripPrefix(
                        "/iadocs/",
                        http.FileServer(http.FS(iaDocsFS)),
                ),
        )

        log.Println("Registering docs handle")
        docsFS, err := fs.Sub(runFS, "build/docs")
        if err != nil {
                log.Fatal(err)
        }
        http.Handle(
                "/docs/",
                http.StripPrefix(
                        "/docs/",
                        http.FileServer(http.FS(docsFS)),
                ),
        )
```

```go
log.Println("Registering source handle")
http.Handle(
	"/src/",
	http.StripPrefix(
		"/src/", http.FileServer(http.FS(srcFS)),
	),
)

log.Println("Registering handlers")
http.HandleFunc("/ws", handleWs)
setHandler("/{$}", handleIndex)
setHandler("/export/choices", handleExportChoices)
setHandler("/export/students", handleExportStudents)
setHandler("/auth", handleAuth)
setHandler("/state/{s}", handleState)
setHandler("/newcourses", handleNewCourses)

var l net.Listener

switch config.Listen.Trans {
case "plain":
	log.Printf(
		"Establishing plain listener for net \"%s\", addr \"%s\"\n",
		config.Listen.Net,
		config.Listen.Addr,
	)
	l, err = net.Listen(config.Listen.Net, config.Listen.Addr)
	if err != nil {
		log.Fatalf(
			"Failed to establish plain listener: %v\n",
			err,
		)
	}
case "tls":
	cer, err := tls.LoadX509KeyPair(
		config.Listen.TLS.Cert,
		config.Listen.TLS.Key,
	)
	if err != nil {
		log.Fatalf(
			"Failed to load TLS certificate and key: %v\n",
			err,
		)
	}
	tlsconfig := &tls.Config{
		Certificates: []tls.Certificate{cer},
		MinVersion:   tls.VersionTLS13,
	} //exhaustruct:ignore
	log.Printf(
		"Establishing TLS listener for net \"%s\", addr \"%s\"\n",
		config.Listen.Net,
		config.Listen.Addr,
		)
		l, err = tls.Listen(
			config.Listen.Net,
			config.Listen.Addr,
			tlsconfig,
		)
		if err != nil {
			log.Fatalf(
				"Failed to establish TLS listener: %v\n",
				err,
			)
		}
	default:
		log.Fatalln("listen.trans must be \"plain\" or \"tls\"")
	}

	log.Println("Setting up database")
	if err := setupDatabase(); err != nil {
		log.Fatal(err)
	}

	log.Println("Loading state")
	if err := loadState(); err != nil {
		log.Fatal(err)
	}

	log.Println("Setting up courses")
	err = setupCourses(context.Background())
	if err != nil {
		log.Fatal(err)
	}

	log.Println("Setting up JWKS")
	if err := setupJwks(); err != nil {
		log.Fatal(err)
	}

	if config.Listen.Proto == "http" {
		log.Println("Serving http")
		srv := &http.Server{
			ReadHeaderTimeout: time.Duration(
				config.Perf.ReadHeaderTimeout,
			) * time.Second,
		} //exhaustruct:ignore
		err = srv.Serve(l)
	} else {
		log.Fatalln("Unsupported protocol")
	}
	if err != nil {
		log.Fatal(err)
	}
}
```

1.15. misc_utils.go

```
/*
 * Utility functions
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
	"crypto/rand"
	"encoding/base64"
	"log"
	"net/http"
)

func wstr(w http.ResponseWriter, code int, msg string) {
	w.Header().Set("Content-Type", "text/plain; charset=utf-8")
	w.WriteHeader(code)
	_, err := w.Write([]byte(msg))
	if err != nil {
		log.Printf("Error wstr'ing to writer: %v", err)
	}
}

/*
 * Generate a random url-safe string.
 * Note that the "sz" parameter specifies the number of bytes taken from the
 * random source divided by three and does NOT represent the length of the
 * encoded string. It's divided by three because we're using base64 and it's
 * ideal to ensure that the entropy remains consistent throughout the string.
 */
func randomString(sz int) (string, error) {
	r := make([]byte, 3*sz)
	_, err := rand.Read(r)
	if err != nil {
		return "", wrapError(errCannotGenerateRandomString, err)
	}
```

```
        return base64.RawURLEncoding.EncodeToString(r), nil
}

func getKeysOfMap[K comparable, V any](i map[K]V) []K {
        o := make([]K, 0, len(i))
        for k := range i {
                o = append(o, k)
        }
        return o
}
```

1.16. session.go

```
/*
 * Session checking functions
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "errors"
        "net/http"

        "github.com/jackc/pgx/v5"
)

func getUserInfoFromRequest(req *http.Request) (userID, username, department string, retErr
 ↪  error) {
        sessionCookie, err := req.Cookie("session")
        if errors.Is(err, http.ErrNoCookie) {
                retErr = wrapError(errNoCookie, err)
                return
        } else if err != nil {
                retErr = wrapError(errCannotCheckCookie, err)
                return
        }
        err = db.QueryRow(
                req.Context(),
                "SELECT id, name, department FROM users WHERE session = $1",
                sessionCookie.Value,
        ).Scan(&userID, &username, &department)
        if err != nil {
                if errors.Is(err, pgx.ErrNoRows) {
                        retErr = errNoSuchUser
                        return
                }
                retErr = wrapError(errUnexpectedDBError, err)
                return
        }
        return
}
```

1.17. sethandler.go

```
/*
 * HTTP handler setting
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "log/slog"
        "net/http"
)

func setHandler(pattern string, handler func(http.ResponseWriter, *http.Request) (string, int,
 ↪  error)) {
        http.HandleFunc(pattern, func(w http.ResponseWriter, req *http.Request) {
                msg, statusCode, err := handler(w, req)
                if err != nil {
                        if statusCode == -1 || statusCode == 0 {
```

```
					statusCode = 500
				}
				slog.Error(
					"handler",
					"path", req.URL.Path,
					"status", statusCode,
					"error", err,
				)
				if msg != "" {
					wstr(w, statusCode, msg+"\n"+err.Error())
				} else {
					wstr(w, statusCode, err.Error())
				}
			} else if msg != "" {
				if statusCode == -1 || statusCode == 0 {
					statusCode = 200
				}
				wstr(w, statusCode, msg)
			}
		})
}
```

1.18. state.go

```
/*
 * Handle the unified global state
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
	"context"
	"errors"
	"sync/atomic"

	"github.com/jackc/pgx/v5"
```

```
)

/*
 * 0: Student access is disabled
 * 1: Student have read-only access
 * 2: Student can choose courses
 */
var state uint32 /* atomic */

func loadState() error {
	var _state uint32
	err := db.QueryRow(
		context.Background(),
		"SELECT value FROM misc WHERE key = 'state'",
	).Scan(&_state)
	if err != nil {
		if errors.Is(err, pgx.ErrNoRows) {
			_state = 0
			_, err := db.Exec(
				context.Background(),
				"INSERT INTO misc(key, value) VALUES ('state', $1)",
				_state,
			)
			if err != nil {
				return wrapError(errUnexpectedDBError, err)
			}
		} else {
			return wrapError(errUnexpectedDBError, err)
		}
	}
	atomic.StoreUint32(&state, _state)
	return nil
}

func saveStateValue(ctx context.Context, newState uint32) error {
	_, err := db.Exec(
		ctx,
		"UPDATE misc SET value = $1 WHERE key = 'state'",
		newState,
	)
	if err != nil {
		return wrapError(errUnexpectedDBError, err)
	}
	return nil
}

func setState(ctx context.Context, newState uint32) error {
	switch newState {
	case 0:
		cancelPool.Range(func(_, value interface{}) bool {
			cancel, ok := value.(*context.CancelFunc)
			if !ok {
				panic("chanPool has non-\"*contect.CancelFunc\" values")
```

```
                                }
                                (*cancel)()
                                return false
                        })
        case 1:
                propagate("STOP")
        case 2:
                propagate("START")
        default:
                return errInvalidState
        }
        err := saveStateValue(ctx, newState)
        if err != nil {
                return err
        }
        atomic.StoreUint32(&state, newState)
        return nil
}
```

1.19. usem.go

```
/*
 * Increase-unblocking capped semaphores
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

/*
 * usemT is basically a semaphore capped at 1. Adding is always non-blocking;
 * adding it multiple times without a read in between is equivalent to setting
 * it once. Reading blocks after the first read after the last set.
 */

type usemT struct {
        ch (chan struct{})
}
```

```
func (s *usemT) init() {
	s.ch = make(chan struct{}, 1)
}

func (s *usemT) set() {
	select {
	case s.ch <- struct{}{}:
	default:
	}
}
```

1.20. ws_connection.go

```
/*
 * WebSocket connection routine
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
	"context"
	"errors"
	"fmt"
	"sync"
	"sync/atomic"
	"time"

	"github.com/coder/websocket"
)

type errbytesT struct {
	err   error
	bytes *[]byte
}
```

```
var usemCount int64 /* atomic */

/*
 * This is more appropriately typed as uint64, but it needs to be cast to int64
 * later anyway due to time.Duration, so let's just use int64.
 */

/*
 * The actual logic in handling the connection, after authentication has been
 * completed.
 */
func handleConn(
        ctx context.Context,
        c *websocket.Conn,
        userID string,
        department string,
) (retErr error) {
        send := make(chan string, config.Perf.SendQ)
        chanPool.Store(userID, &send)
        defer chanPool.CompareAndDelete(userID, &send)

        reportError := makeReportError(ctx, c)
        newCtx, newCancel := context.WithCancel(ctx)

        _cancel, ok := cancelPool.Load(userID)
        if ok {
                cancel, ok := _cancel.(*context.CancelFunc)
                if ok && cancel != nil {
                        (*cancel)()
                }
                /* TODO: Make the cancel synchronous */
        }
        cancelPool.Store(userID, &newCancel)

        defer func() {
                cancelPool.CompareAndDelete(userID, &newCancel)
                if errors.Is(retErr, context.Canceled) {
                        /*
                         * Only works if it's newCtx that has been canceled
                         * rather than the original ctx, which is kinda what
                         * we intend
                         */
                        _ = writeText(ctx, c, "E :Context canceled")
                }
                /* TODO: Report errors properly */
        }()

        /* TODO: Tell the user their current choices here. Deprecate HELLO. */

        usems := make(map[int]*usemT)

        /* TODO: Check if the LoadUint32 here is a bit too much overhead */
        atomic.AddInt64(&usemCount, int64(atomic.LoadUint32(&numCourses)))
```

```
courses.Range(func(key, value interface{}) bool {
        /* TODO: Remember to change this too when changing the courseID type */
        courseID, ok := key.(int)
        if !ok {
                panic("courses map has non-\"int\" keys")
        }
        course, ok := value.(*courseT)
        if !ok {
                panic("courses map has non-\"*courseT\" items")
        }
        usem := &usemT{} //exhaustruct:ignore
        usem.init()
        course.Usems.Store(userID, usem)
        usems[courseID] = usem
        return true
})

defer func() {
        courses.Range(func(key, value interface{}) bool {
                _ = key
                course, ok := value.(*courseT)
                if !ok {
                        panic("courses map has non-\"*courseT\" items")
                }
                course.Usems.Delete(userID)
                return true
        })
        atomic.AddInt64(&usemCount, -int64(atomic.LoadUint32(&numCourses)))
}()

usemParent := make(chan int)
for courseID, usem := range usems {
        go func() {
                for {
                        select {
                        case <-newCtx.Done():
                                return
                        case <-usem.ch:
                                select {
                                case <-newCtx.Done():
                                        return
                                case usemParent <- courseID:
                                }
                        }
                        time.Sleep(
                                time.Duration(
                                        atomic.LoadInt64(&usemCount)>>
                                                config.Perf.UsemDelayShiftBits,
                                ) * time.Millisecond,
                        )
                }
        }()
}
```

```go
var userCourseGroups userCourseGroupsT = make(map[string]struct{})
var userCourseTypes userCourseTypesT = make(map[string]int)
err := populateUserCourseTypesAndGroups(newCtx, &userCourseTypes, &userCourseGroups,
 ↪  userID)
if err != nil {
        return reportError(
                fmt.Sprintf(
                        "cannot populate user course types/groups: %v",
                        err,
                ),
        )
}

/*
 * Later we need to select from recv and send and perform the
 * corresponding action. But we can't just select from c.Read because
 * the function blocks. Therefore, we must spawn a goroutine that
 * blocks on c.Read and send what it receives to a channel "recv"; and
 * then we can select from that channel.
 */
recv := make(chan *errbytesT)
go func() {
        for {
                /*
                 * Here we use the original connection context instead
                 * of the new context we just created. Apparently when
                 * the context passed to Read expires, the connection
                 * gets closed, which makes it impossible for us to
                 * write the context expiry message to the client.
                 * So we pass the original connection context, which
                 * would get canceled anyway once we close the
                 * connection.
                 * See: https://github.com/coder/websocket/issues/242
                 * We still need to take care of this while sending so
                 * we don't infinitely block, and leak goroutines and
                 * cause the channel to remain out of reach of the
                 * garbage collector.
                 * It would be nice to return the newCtx.Err() but
                 * the only way to really do that is to use the recv
                 * channel which might not have a listener anymore.
                 * It's not really crucial anyways so we could just
                 * close this goroutine by returning here.
                 */
                _, b, err := c.Read(ctx)
                if err != nil {
                        /*
                         * TODO: Prioritize context dones... except
                         * that it's not really possible. I would just
                         * have placed newCtx in here but apparently
                         * that causes the connection to be closed when
                         * the context expires, which makes it
                         * impossible to deliver the final error
                         * message. Probably need to look into this
                         * design again.
                         */
                        select {
                        case <-newCtx.Done():
                                _ = writeText(
                                        ctx,
                                        c,
                                        "E :Context canceled",
                                )
                                /* Not a typo to use ctx here */
                                return
                        case recv <- &errbytesT{err: err, bytes: nil}:
                        }
                        return
                }
                select {
                case <-newCtx.Done():
                        _ = writeText(ctx, c, "E :Context canceled")
                        /* Not a typo to use ctx here */
                        return
                case recv <- &errbytesT{err: nil, bytes: &b}:
                }
        }
}()

for {
        var mar []string
        select {
        case <-newCtx.Done():
                /*
                 * We select this context done channel when entering
                 * other cases too (see below) because we need to
                 * make sure the context cancel works even if both
                 * the cancel signal and another event arrive while
                 * processing a select cycle.
                 */
                return wrapError(
                        errContextCanceled,
                        newCtx.Err(),
                )
        case sendText := <-send:
                select {
                case <-newCtx.Done():
                        return wrapError(
                                errContextCanceled,
                                newCtx.Err(),
                        )
                default:
                }

                err := writeText(newCtx, c, sendText)
                if err != nil {
```

```
                return err
        }
case courseID := <-usemParent:
        select {
        case <-newCtx.Done():
                return wrapError(
                        errContextCanceled,
                        newCtx.Err(),
                )
        default:
        }

        err := sendSelectedUpdate(newCtx, c, courseID)
        if err != nil {
                return wrapError(
                        errCannotSend,
                        err,
                )
        }
        continue
case errbytes := <-recv:
        select {
        case <-newCtx.Done():
                return wrapError(
                        errContextCanceled,
                        newCtx.Err(),
                )
        default:
        }

        if errbytes.err != nil {
                return wrapError(
                        errCannotReceiveMessage,
                        errbytes.err,
                )
                /*
                 * Note that this cannot return newCtx.Err(),
                 * so we handle the error reporting in the
                 * reading routine
                 */
        }
        mar = splitMsg(errbytes.bytes)
        switch mar[0] {
        case "HELLO":
                err := messageHello(
                        newCtx,
                        c,
                        reportError,
                        mar,
                        userID,
                )
                if err != nil {
                        return err
                }
        case "Y":
                err := messageChooseCourse(
                        newCtx,
                        c,
                        reportError,
                        mar,
                        userID,
                        &userCourseGroups,
                        &userCourseTypes,
                )
                if err != nil {
                        return err
                }
        case "N":
                err := messageUnchooseCourse(
                        newCtx,
                        c,
                        reportError,
                        mar,
                        userID,
                        &userCourseGroups,
                        &userCourseTypes,
                )
                if err != nil {
                        return err
                }
        case "YC":
                err := messageConfirm(
                        newCtx,
                        c,
                        reportError,
                        mar,
                        userID,
                        department,
                        &userCourseTypes,
                )
                if err != nil {
                        return err
                }
        case "NC":
                err := messageUnconfirm(
                        newCtx,
                        c,
                        reportError,
                        mar,
                        userID,
                )
                if err != nil {
                        return err
                }
        default:
                return reportError("Unknown command " + mar[0])
```

```
                        }
                }
        }
}

var cancelPool sync.Map /* string, *context.CancelFunc */

var chanPool sync.Map /* string, *chan string */
```

1.21. ws_utils.go

```
/*
 * WebSocket auxiliary functions
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "fmt"
        "log"
        "sync/atomic"

        "github.com/coder/websocket"
)

/*
 * The message format is a WebSocket message separated with spaces.
 * The contents of each field could contain anything other than spaces,
 * The first character of each argument cannot be a colon. As an exception, the
 * last argument may contain spaces and the first character thereof may be a
 * colon, if the argument is prefixed with a colon. The colon used for the
 * prefix is not considered part of the content of the message. For example, in
 *
 *    SQUISH POP :cat purr!!
 *
 * the first field is "SQUISH", the second field is "POP", and the third
 * field is "cat purr!!".
 *
 * It is essentially an RFC 1459 IRC message without trailing CR-LF and
 * without prefixes. See section 2.3.1 of RFC 1459 for an approximate
 * BNF representation.
 *
 * The reason this was chosen instead of using protobuf etc. is that it
 * is simple to parse without external libraries, and it also happens to
 * be a format I'm very familiar with, having extensively worked with the
 * IRC protocol.
 */

/*
 * Split an IRC-style message of type []byte into type []string where each
 * element is a complete argument. Generally, arguments are separated by
 * spaces, and an argument that begins with a ':' causes the rest of the
 * line to be treated as a single argument.
 */
func splitMsg(b *[]byte) []string {
        mar := make([]string, 0, config.Perf.MessageArgumentsCap)
        elem := make([]byte, 0, config.Perf.MessageBytesCap)
        for i, c := range *b {
                switch c {
                case ' ':
                        if (*b)[i+1] == ':' {
                                mar = append(mar, string(elem))
                                mar = append(mar, string((*b)[i+2:]))
                                goto endl
                        }
                        mar = append(mar, string(elem))
                        elem = make([]byte, 0, config.Perf.MessageBytesCap)
                default:
                        elem = append(elem, c)
                }
        }
        mar = append(mar, string(elem))
endl:
        return mar
}

func baseReportError(
        ctx context.Context,
        conn *websocket.Conn,
        e string,
) error {
        err := writeText(ctx, conn, "E :"+e)
        if err != nil {
                return fmt.Errorf("error reporting protocol violation: %w", err)
        }
        err = conn.Close(websocket.StatusProtocolError, e)
        if err != nil {
                return fmt.Errorf("error closing websocket: %w", err)
        }
        return nil
}

type reportErrorT func(e string) error

func makeReportError(ctx context.Context, conn *websocket.Conn) reportErrorT {
        return func(e string) error {
                return baseReportError(ctx, conn, e)
        }
}

func propagateSelectedUpdate(course *courseT) {
        course.Usems.Range(func(key, value interface{}) bool {
                _ = key
                usem, ok := value.(*usemT)
                if !ok {
                        panic("Usems contains non-\"*usemT\" value")
                }
                usem.set()
                return true
        })
}

func sendSelectedUpdate(
        ctx context.Context,
        conn *websocket.Conn,
        courseID int,
) error {
        _course, ok := courses.Load(courseID)
        if !ok {
                return fmt.Errorf("%w: %d", errNoSuchCourse, courseID)
        }
        course, ok := _course.(*courseT)
        if !ok {
                panic("courses map has non-\"*courseT\" items")
        }
        if course == nil {
                return fmt.Errorf("%w: %d", errNoSuchCourse, courseID)
        }
        selected := atomic.LoadUint32(&course.Selected)
        err := writeText(ctx, conn, fmt.Sprintf("M %d %d", courseID, selected))
        if err != nil {
                return fmt.Errorf(
                        "error sending to websocket for course selected update: %w",
                        err,
                )
        }
        return nil
}

func propagate(msg string) {
        chanPool.Range(func(_userID, _ch interface{}) bool {
```

```
		ch, ok := _ch.(*chan string)
		if !ok {
			panic("chanPool has non-\"*chan string\" key")
		}
		select {
		case *ch <- msg:
		default:
			userID, ok := _userID.(string)
			if !ok {
				panic("chanPool has non-string key")
			}
			log.Println("WARNING: SendQ exceeded for " + userID)
		}
		return true
	})
}

func writeText(ctx context.Context, c *websocket.Conn, msg string) error {
	err := c.Write(ctx, websocket.MessageText, []byte(msg))
	if err != nil {
		return wrapError(errWebSocketWrite, err)
	}
	return nil
}
```

1.22. wsmsg_choose.go

```
/*
 * Handle the "Y" message for choosing a course
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
	"context"
	"errors"
	"fmt"
	"strconv"
	"sync/atomic"
	"time"

	"github.com/coder/websocket"
	"github.com/jackc/pgx/v5"
	"github.com/jackc/pgx/v5/pgconn"
)

func messageChooseCourse(
	ctx context.Context,
	c *websocket.Conn,
	reportError reportErrorT,
	mar []string,
	userID string,
	userCourseGroups *userCourseGroupsT,
	userCourseTypes *userCourseTypesT,
) error {
	if atomic.LoadUint32(&state) != 2 {
		err := writeText(ctx, c, "E :Course selections are not open")
		if err != nil {
			return wrapError(
				errCannotSend,
				err,
			)
		}
		return nil
	}

	select {
	case <-ctx.Done():
		return wrapError(
			errContextCanceled,
			ctx.Err(),
		)
	default:
	}

	if len(mar) != 2 {
		return reportError("Invalid number of arguments for Y")
	}
	_courseID, err := strconv.ParseInt(mar[1], 10, strconv.IntSize)
	if err != nil {
		return reportError("Course ID must be an integer")
	}
	courseID := int(_courseID)

	_course, ok := courses.Load(courseID)
	if !ok {
		return reportError("no such course")
	}
	course, ok := _course.(*courseT)
```

```
if !ok {
        panic("courses map has non-\"*courseT\" items")
}
if course == nil {
        return reportError("couse is nil")
}

if _, ok := (*userCourseGroups)[course.Group]; ok {
        err := writeText(ctx, c, "R "+mar[1]+" :Group conflict")
        if err != nil {
                return wrapError(
                        errCannotSend,
                        err,
                )
        }
        return nil
}

err = func() (returnedError error) {
        tx, err := db.Begin(ctx)
        if err != nil {
                return reportError(
                        "Database error while beginning transaction",
                )
        }
        defer func() {
                err := tx.Rollback(ctx)
                if err != nil && (!errors.Is(err, pgx.ErrTxClosed)) {
                        returnedError = reportError(
                                "Database error while rolling back transaction in defer
                                 ↪  block",
                        )
                        return
                }
        }()

        _, err = tx.Exec(
                ctx,
                "INSERT INTO choices (seltime, userid, courseid) VALUES ($1, $2, $3)",
                time.Now().UnixMicro(),
                userID,
                courseID,
        )
        if err != nil {
                var pgErr *pgconn.PgError
                if errors.As(err, &pgErr) &&
                        pgErr.Code == pgErrUniqueViolation {
                        err := writeText(ctx, c, "Y "+mar[1])
                        if err != nil {
                                return fmt.Errorf(
                                        "error reaffirming course choice: %w",
                                        err,
                                )
                        }
                        return nil
                }
                return reportError(
                        "Database error while inserting course choice",
                )
        }

        ok := func() bool {
                course.SelectedLock.Lock()
                defer course.SelectedLock.Unlock()
                /*
                 * The read here doesn't have to be atomic because the
                 * lock guarantees that no other goroutine is writing to
                 * it.
                 */
                if course.Selected < course.Max {
                        atomic.AddUint32(&course.Selected, 1)
                        return true
                }
                return false
        }()

        if ok {
                go propagateSelectedUpdate(course)
                err := tx.Commit(ctx)
                if err != nil {
                        err := course.decrementSelectedAndPropagate(ctx, c)
                        if err != nil {
                                return wrapError(
                                        errCannotSend,
                                        err,
                                )
                        }
                        return reportError(
                                "Database error while committing transaction",
                        )
                }

                /*
                 * This would race if message handlers could run
                 * concurrently for one connection.
                 */
                (*userCourseGroups)[course.Group] = struct{}{}
                (*userCourseTypes)[course.Type]++

                err = writeText(ctx, c, "Y "+mar[1])
                if err != nil {
                        return wrapError(
                                errCannotSend,
                                err,
                        )
                }
```

```
                        if config.Perf.PropagateImmediate {
                                err = sendSelectedUpdate(ctx, c, courseID)
                                if err != nil {
                                        return wrapError(
                                                errCannotSend,
                                                err,
                                        )
                                }
                        }
                } else {
                        err := tx.Rollback(ctx)
                        if err != nil {
                                return reportError(
                                        "Database error while rolling back transaction due to
                                         ↪  course limit",
                                )
                        }
                        err = writeText(ctx, c, "R "+mar[1]+" :Full")
                        if err != nil {
                                return wrapError(
                                        errCannotSend,
                                        err,
                                )
                        }
                }
                return nil
        }()
        if err != nil {
                return err
        }
        return nil
}
```

1.23. wsmsg_confirm.go

```
/*
 * Handle the "C" message
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "fmt"
        "sync/atomic"

        "github.com/coder/websocket"
)

func messageConfirm(
        ctx context.Context,
        c *websocket.Conn,
        reportError reportErrorT,
        mar []string,
        userID string,
        department string,
        userCourseTypes *userCourseTypesT,
) error {
        _ = mar

        if atomic.LoadUint32(&state) != 2 {
                err := writeText(ctx, c, "E :Course selections are not open")
                if err != nil {
                        return wrapError(
                                errCannotSend,
                                err,
                        )
                }
                return nil
        }

        select {
        case <-ctx.Done():
                return wrapError(
                        errContextCanceled,
                        ctx.Err(),
                )
        default:
        }

        for courseType := range courseTypes {
                minimum, err := getCourseTypeMinimumForYearGroup(department, courseType)
                if err != nil {
                        return reportError("Invalid year group or course type, something is
                         ↪  broken")
                }
                if (*userCourseTypes)[courseType] < minimum {
                        return writeText(
                                ctx,
                                c,
                                fmt.Sprintf(
                                        "RC :Cannot confirm choices: You chose %d out of
                                         ↪  required %d of type %s",
                                        (*userCourseTypes)[courseType],
                                        minimum,
                                        courseType,
                                ),
                        )
                }
        }

        _, err := db.Exec(
                ctx,
                "UPDATE users SET confirmed = true WHERE id = $1",
                userID,
        )
        if err != nil {
                return reportError("error updating database setting confirmation")
        }

        return writeText(
                ctx,
                c,
                "YC",
        )
}
```

1.24. wsmsg_hello.go

```
/*
 * Handle the "HELLO" message
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main
```

```go
import (
	"context"
	"strings"
	"sync/atomic"

	"github.com/coder/websocket"
	"github.com/jackc/pgx/v5"
)

func messageHello(
	ctx context.Context,
	c *websocket.Conn,
	reportError reportErrorT,
	mar []string,
	userID string,
) error {
	_ = mar

	select {
	case <-ctx.Done():
		return wrapError(
			errContextCanceled,
			ctx.Err(),
		)
	default:
	}

	rows, err := db.Query(
		ctx,
		"SELECT courseid FROM choices WHERE userid = $1",
		userID,
	)
	if err != nil {
		return reportError("error fetching choices")
	}
	courseIDs, err := pgx.CollectRows(rows, pgx.RowTo[string])
	if err != nil {
		return reportError("error collecting choices")
	}

	if atomic.LoadUint32(&state) == 2 {
		err = writeText(ctx, c, "START")
		if err != nil {
			return wrapError(errCannotSend, err)
		}
	}

	confirmed, err := getConfirmedStatus(ctx, userID)
	if err != nil {
		return err
	}
	if !confirmed {
```

```
                err = writeText(ctx, c, "NC")
                if err != nil {
                        return wrapError(errCannotSend, err)
                }
        } else {
                err = writeText(ctx, c, "YC")
                if err != nil {
                        return wrapError(errCannotSend, err)
                }
        }

        err = writeText(ctx, c, "HI :"+strings.Join(courseIDs, ","))
        if err != nil {
                return wrapError(errCannotSend, err)
        }

        return nil
}
```

1.25. wsmsg_unchoose.go

```
/*
 * Handle the "N" message for unchoosing a course
 *
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "strconv"
        "sync/atomic"

        "github.com/coder/websocket"
)

func messageUnchooseCourse(
	ctx context.Context,
	c *websocket.Conn,
	reportError reportErrorT,
	mar []string,
	userID string,
	userCourseGroups *userCourseGroupsT,
	userCourseTypes *userCourseTypesT,
) error {
	if atomic.LoadUint32(&state) != 2 {
		err := writeText(ctx, c, "E :Course selections are not open")
		if err != nil {
			return wrapError(
				errCannotSend,
				err,
			)
		}
		return nil
	}

	select {
	case <-ctx.Done():
		return wrapError(
			errContextCanceled,
			ctx.Err(),
		)
	default:
	}

	if len(mar) != 2 {
		return reportError("Invalid number of arguments for N")
	}
	_courseID, err := strconv.ParseInt(mar[1], 10, strconv.IntSize)
	if err != nil {
		return reportError("Course ID must be an integer")
	}
	courseID := int(_courseID)

	_course, ok := courses.Load(courseID)
	if !ok {
		return reportError("no such course")
	}
	course, ok := _course.(*courseT)
	if !ok {
		panic("courses map has non-\"*courseT\" items")
	}
	if course == nil {
		return reportError("couse is nil")
	}

	ct, err := db.Exec(
		ctx,
		"DELETE FROM choices WHERE userid = $1 AND courseid = $2",
		userID,
			courseID,
		)
		if err != nil {
			return reportError(
				"Database error while deleting course choice",
			)
		}

		if ct.RowsAffected() != 0 {
			err := course.decrementSelectedAndPropagate(ctx, c)
			if err != nil {
				return wrapError(
					errCannotSend,
					err,
				)
			}

			_course, ok := courses.Load(courseID)
			if !ok {
				return reportError("no such course")
			}
			course, ok := _course.(*courseT)
			if !ok {
				panic("courses map has non-\"*courseT\" items")
			}
			if course == nil {
				return reportError("couse is nil")
			}

			if _, ok := (*userCourseGroups)[course.Group]; !ok {
				return reportError("inconsistent user course groups")
			}
			delete(*userCourseGroups, course.Group)
			(*userCourseTypes)[course.Type]--
		}

		err = writeText(ctx, c, "N "+mar[1])
		if err != nil {
			return wrapError(
				errCannotSend,
				err,
			)
		}

		return nil
}
```

1.26. wsmsg_unconfirm.go

```
/*
 * Handle the "C" message
 *
```

```go
 * Copyright (C) 2024  Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

package main

import (
        "context"
        "sync/atomic"

        "github.com/coder/websocket"
)

func messageUnconfirm(
        ctx context.Context,
        c *websocket.Conn,
        reportError reportErrorT,
        mar []string,
        userID string,
) error {
        _ = mar

        if atomic.LoadUint32(&state) != 2 {
                err := writeText(ctx, c, "E :Course selections are not open")
                if err != nil {
                        return wrapError(
                                errCannotSend,
                                err,
                        )
                }
                return nil
        }

        select {
        case <-ctx.Done():
                return wrapError(
                        errContextCanceled,
                        ctx.Err(),
                )
        default:
	}

	_, err := db.Exec(
		ctx,
		"UPDATE users SET confirmed = false WHERE id = $1",
		userID,
	)
	if err != nil {
		return reportError("error updating database setting confirmation")
	}

	return writeText(
		ctx,
		c,
		"NC",
	)
}
```

1.27. go.mod

```
module git.sr.ht/~runxiyu/cca

go 1.23.1

require (
	git.sr.ht/~emersion/go-scfg v0.0.0-20240128091534-2ae16e782082
	github.com/MicahParks/keyfunc/v3 v3.3.5
	github.com/coder/websocket v1.8.12
	github.com/golang-jwt/jwt/v5 v5.2.1
	github.com/jackc/pgx/v5 v5.7.1
)

require (
	github.com/MicahParks/jwkset v0.5.20 // indirect
	github.com/jackc/pgpassfile v1.0.0 // indirect
	github.com/jackc/pgservicefile v0.0.0-20240606120523-5a60cdf6a761 // indirect
	github.com/jackc/puddle/v2 v2.2.2 // indirect
	golang.org/x/crypto v0.28.0 // indirect
	golang.org/x/sync v0.8.0 // indirect
	golang.org/x/text v0.19.0 // indirect
	golang.org/x/time v0.7.0 // indirect
)
```

1.28. go.sum

```
git.sr.ht/~emersion/go-scfg v0.0.0-20240128091534-2ae16e782082
 ↪  h1:9Udx5fm4vRtmgDIBjy2ef5QioHbzpw5oHabbhpAUyEw=
git.sr.ht/~emersion/go-scfg v0.0.0-20240128091534-2ae16e782082/go.mod
 ↪  h1:ybgvEJTIx5XbaspSviB3KNa6OdPmAZqDoSud7z8fFlw=
github.com/MicahParks/jwkset v0.5.20 h1:gTIKx9AofTqQJ0srd8AL7ty9NeadP5WUXSPOZadTpOI=
github.com/MicahParks/jwkset v0.5.20/go.mod h1:q8ptTGn/Z9c4MwbcfeCDssADeVQb3Pk7PnVxrvi+2QY=
github.com/MicahParks/keyfunc/v3 v3.3.5 h1:7ceAJLUAldnoueHDNzF8Bx06oVcQ5CfJnYwNt1U3YYo=
github.com/MicahParks/keyfunc/v3 v3.3.5/go.mod h1:SdCCyMJn/bYqWDvARspC6nCT8Sk74MjuAY22C7dCST8=
github.com/coder/websocket v1.8.12 h1:5bUXkEPPIbewrnkU8LTCLVaxi4N4J8ahufH2vlo4NAo=
github.com/coder/websocket v1.8.12/go.mod h1:LNVeNrXQZfe5qhS9ALED3uA+l5pPqvwXg3CKoDBB2gs=
github.com/davecgh/go-spew v1.1.0/go.mod h1:J7Y8YcW2NihsgmVo/mv3lAwl/skON4iLHjSsI+c5H38=
github.com/davecgh/go-spew v1.1.1 h1:vj9j/u1bqnvCEfJOwUhtlOARqs3+rkHYY13jYWTU97c=
github.com/davecgh/go-spew v1.1.1/go.mod h1:J7Y8YcW2NihsgmVo/mv3lAwl/skON4iLHjSsI+c5H38=
github.com/golang-jwt/jwt/v5 v5.2.1 h1:OuVbFODueb089Lh128TAcimifWaLhJwVflnrgM17wHk=
github.com/golang-jwt/jwt/v5 v5.2.1/go.mod h1:pqrtFR0X4osieyHYxtmOUWsAWrfe1Q5UVIyoH402zdk=
github.com/jackc/pgpassfile v1.0.0 h1:/6Hmqy13Ss2zCq62VdNG8tM1wchn8zjSGOBJ6icpsIM=
github.com/jackc/pgpassfile v1.0.0/go.mod h1:CEx0iS5ambNFdcRtxPj5JhEz+xB6uRky5eyVu/W2HEg=
github.com/jackc/pgservicefile v0.0.0-20240606120523-5a60cdf6a761
 ↪  h1:iCEnooe7UlwOQYpKFhBabPMi4aNAfoODPEFNiAnClxo=
github.com/jackc/pgservicefile v0.0.0-20240606120523-5a60cdf6a761/go.mod
 ↪  h1:5TJZWKEWniPve33vlWYSoGYefn3gLQRzjfDlhSJ9ZKM=
github.com/jackc/pgx/v5 v5.7.1 h1:x7SYsPBYDkHDksogeSmZZ5xzThcTgRz++I5E+ePFUcs=
github.com/jackc/pgx/v5 v5.7.1/go.mod h1:e7O26IywZZ+naJtWWos6i6fvWK+29etgITqrqHLfoZA=
github.com/jackc/puddle/v2 v2.2.2 h1:PR8nw+E/1w0GLuRFSmiioY6UooMp6KJv0/61nB7icHo=
github.com/jackc/puddle/v2 v2.2.2/go.mod h1:vriiEXHvEE654aYKXXjOvZM39qJ0q+azkZFrfEOc3H4=
github.com/pmezard/go-difflib v1.0.0 h1:4DBwDE0NGyQoBHbLQYPwSUPoCMWR5BEzIk/f1lZbAQM=
github.com/pmezard/go-difflib v1.0.0/go.mod h1:iKH77koFhYxTK1pcRnkKkqfTogsbg7gZNVY4sRDYZ/4=
github.com/stretchr/objx v0.1.0/go.mod h1:HFkY916IF+rwdDfMAkV7OtwuqBVzrE8GR6GFx+wExME=
github.com/stretchr/testify v1.3.0/go.mod h1:M5WIy9Dh21IEIfnGCwXGc5bZfKNJtfHm1UVUgZn+9EI=
github.com/stretchr/testify v1.7.0/go.mod h1:6Fq8oRcR53rry900zMqJjRRixrwX3KX962/h/Wwjteg=
github.com/stretchr/testify v1.8.1 h1:w7B6lhMri9wdJUVmEZPGGhZzrYTPvgJArz7wNPgYKsk=
github.com/stretchr/testify v1.8.1/go.mod h1:w2LPCIKwWwSfY2zedu0+kehJoqGctiVI29o6fzry7u4=
golang.org/x/crypto v0.28.0 h1:GBDwsMXVQi34v5CCYUm2jkJvu4cbtru2U4TN2PSyQnw=
golang.org/x/crypto v0.28.0/go.mod h1:rmgy+3RHxRZMyY0jjAJShp2zgEdOqj2AO7U0pYmeQ7U=
golang.org/x/sync v0.8.0 h1:3NFvSEYkUoMifnESzZl15y791HH1qU2xm6eCJU5ZPXQ=
golang.org/x/sync v0.8.0/go.mod h1:Czt+wKu1gCyEFDUtn0jG5QVvpJ6rzVqr5aXyt9drQfk=
golang.org/x/text v0.19.0 h1:kTxAhCbGbxhK0IwgSKiMO5awPoDQ0RpfiVYBfK860YM=
golang.org/x/text v0.19.0/go.mod h1:BuEKDfySbSR4drPmRPG/7iBdf8hvFMuRexcpahXilzY=
golang.org/x/time v0.7.0 h1:ntUhktv3OPE6TgYxXWv9vKvUSJyIFJlyohwbkEwPrKQ=
golang.org/x/time v0.7.0/go.mod h1:3BpzKBy/shNhVucY/MWOyx10tF3SFh9QdLuxbVysPQM=
gopkg.in/check.v1 v0.0.0-20161208181325-20d25e280405/go.mod
 ↪  h1:Co6ibVJAznAaIkqp8huTwlJQCZ016jof/cbN4VW5Yz0=
gopkg.in/yaml.v3 v3.0.0-20200313102051-9f266ea9e77c/go.mod
 ↪  h1:K4uyk7z7BCEPqu6E+C64Yfv1cQ7kz7rIZviUmN+EgEM=
gopkg.in/yaml.v3 v3.0.1 h1:fxVm/GzAzEWqLHuvctI91KS9hhNmmWOoWu0XTYJS7CA=
gopkg.in/yaml.v3 v3.0.1/go.mod h1:K4uyk7z7BCEPqu6E+C64Yfv1cQ7kz7rIZviUmN+EgEM=
```

2. Frontend source code

2.1. eslint.config.js

```
export default [
    {
        files: ["*.js"],
        languageOptions: {
            globals: {
                document: "readonly",
                alert: "readonly",
                WebSocket: "readonly"
            }
        }
    },
    {
        rules: {
            indent: ["error", "tab"],
            "no-negated-in-lhs": "error",
            "no-cond-assign": ["error", "except-parens"],
            curly: ["error", "all"],
            "object-curly-spacing": ["error", "always"],
            "computed-property-spacing": ["error", "never"],
            "array-bracket-spacing": ["error", "never"],
            eqeqeq: ["error", "smart"],
            "no-unused-expressions": "error",
            "no-sequences": "error",
            "no-nested-ternary": "error",
            "no-unreachable": "error",
            "wrap-iife": ["error", "inside"],
            "no-caller": "error",
            quotes: ["error", "double"],
            "no-undef": "error",
            "no-unused-vars": [
                "error",
                {
                    args: "all",
                    argsIgnorePattern: "^_"
                }
            ],
            "operator-linebreak": ["error", "after"],
            "comma-style": ["error", "last"],
            camelcase: [
                "error",
                {
                    properties: "never"
                }
            ],
            "dot-notation": [
                "error",
                {
                    allowPattern: "^[a-z]+(_[a-z]+)+$"
                }
            ],
            "max-len": [
                "error",
                {
                    code: 200,
                    ignoreComments: true,
                    ignoreUrls: true,
                    ignoreRegExpLiterals: true
                }
            ],
            "no-mixed-spaces-and-tabs": "error",
            "no-trailing-spaces": "error",
            "no-irregular-whitespace": "error",
            "no-multi-str": "error",
            "comma-dangle": ["error", "never"],
            "comma-spacing": [
                "error",
                {
                    before: false,
                    after: true
                }
            ],
            "space-before-blocks": ["error", "always"],
            "space-in-parens": ["error", "never"],
            "keyword-spacing": [2],
            "template-curly-spacing": ["error", "always"],
            semi: ["error", "never"],
            "semi-spacing": [
                "error",
                {
                    before: false,
                    after: true
                }
            ],
            "no-extra-semi": "error",
            "space-infix-ops": "error",
            "eol-last": "error",
            "lines-around-comment": [
                "error",
                {
                    beforeLineComment: true
                }
            ],
            "linebreak-style": ["error", "unix"],
            "no-with": "error",
            "brace-style": "error",
            "space-before-function-paren": ["error", "never"],
            "no-loop-func": "error",
            "no-spaced-func": "error",
            "key-spacing": [
                "error",
                {
                    beforeColon: false,
                    afterColon: true
                }
            ],
            "space-unary-ops": [
                "error",
                {
                    words: false,
                    nonwords: false
                }
            ],
            "no-multiple-empty-lines": 2
        }
    }
]
```

2.2. student.js

```
/*
 * Copyright (c) 2024 Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

document.addEventListener("DOMContentLoaded", () => {
    const socket = new WebSocket("wss://cca.runxiyu.org/ws")

    /*
     * TODO I want to make this easily configurable somehow, but I'm unsure
     * how to fill things into JavaScript. A few possible solutions:
     * - Replace this string during build time
     *   This is suboptimal because users should be able to replace it
     *   during runtime, as the binary is supposed to be decoupled from
     *   particular instances.
     * - Replace this string while setting the static handler.
     *   This is a bit more involved because it requires messing with io.fs;
     *   I also don't know a way to cleanly escape it.
     * - Indicate this string somewhere in the template (perhaps via
     *   a JavaScript variable that we could access).
     *   This is probably the way to go, especially since html/template
     *   provides contextual escaping.
     */

    socket.addEventListener("open", function() {
        let gstate = 0
        let ustate = 0
        let _handleMessage = event => {
            let msg = new String(event?.data)

            /*
             * Standard IRC Message format parsing without IRCv3
             * tags or prefixes.  It's a simple enough protocol
             * format suitable for our use-case.  No need for
             * protobuf or anything else nontrivial.
             */
            let mar = msg.split(" ")
            for (let i = 0; i < mar.length; i++) {
                if (mar[i].startsWith(":")) {
                    if (i === mar.length - 1) {
                        mar[i] = mar[i].substring(1)
                        break
                    }
                    mar[i] = mar[i].substring(1) + " " +
                        mar.slice(i + 1).join(" ")
                    mar.splice(i + 1)
                    break
                }
            }

            switch (mar[0]) {
            case "E": /* unexpected error */
                alert(`The server reported an unexpected error, "${ mar[1] }". The system might
                 ↪  be in an inconsistent state.`)
                break
            case "HI":
                document.querySelectorAll(".need-connection").
                    forEach(c => {
                        c.style.display = "block"
                    })
                document.querySelectorAll(".before-connection").
                    forEach(c => {
                        c.style.display = "none"
                    })
                if (mar[1] !== "") {
                    let courseIDs = mar[1].split(",")
                    for (let i = 0; i < courseIDs.length; i++) {
                        document.getElementById(
                            `tick${ courseIDs[i] }`
                        ).checked = true
                        {
                            let courseType = document.
                                getElementById(`type${ courseIDs[i] }`).
                                textContent
                            document.getElementById(`${ courseType }-chosen`).
                                textContent = parseInt(document.
                                    getElementById(`${ courseType }-chosen`).
                                    textContent) + 1
                        }
                        if (gstate === 1) {
                            document.getElementById(
                                `tick${ courseIDs[i] }`
                            ).disabled = false
                            if (parseInt(document.getElementById("Sport-chosen").textContent)
                             ↪  >=
                                parseInt(document.getElementById("Sport-required").textContent)
                                 ↪  &&
                                parseInt(document.getElementById("Non-sport-chosen").textContent)
                                 ↪  >=
                                parseInt(document.getElementById("Non-sport-required").textContent))
                                 ↪  {
                                document.getElementById("confirmbutton").disabled = false
                            }
                        }
                    }
                }
                if (ustate === 1) {
                    document.querySelectorAll(".confirmed-handle").forEach(c => {
                        let handle = c.textContent
                        document.getElementById(`confirmed-name-${ handle }`).textContent = ""
                        document.getElementById(`confirmed-type-${ handle }`).textContent = ""
                        document.getElementById(`confirmed-teacher-${ handle }`).textContent =
                         ↪  ""
                        document.getElementById(`confirmed-location-${ handle }`).textContent =
                         ↪  ""
                        document.querySelectorAll(".coursecheckbox").forEach(d => {
                            if (d.dataset.group === handle && d.checked) {
                                document.getElementById(`confirmed-name-${ handle
                                 ↪  }`).textContent =
                                    d.dataset.title
                                document.getElementById(`confirmed-type-${ handle
                                 ↪  }`).textContent =
                                    d.dataset.type
                                document.getElementById(`confirmed-teacher-${ handle
                                 ↪  }`).textContent =
                                    d.dataset.teacher
                                document.getElementById(`confirmed-location-${ handle
                                 ↪  }`).textContent =
                                    d.dataset.location

                                /* TODO: break */
                            }
                        })
                    })
                    document.querySelectorAll(".unconfirmed").forEach(c => {
                        c.style.display = "none"
                    })
                    document.querySelectorAll(".confirmed").forEach(c => {
                        c.style.display = "block"
                    })
                    document.querySelectorAll(".neither-confirmed").forEach(c => {
                        c.style.display = "none"
                    })
                }
                break
            case "U": /* unauthenticated */
                /* TODO: replace this with a box on screen */
                alert("Your session is broken or has expired. You are unauthenticated and the
                 ↪  server will reject your commands.")
                break
            case "N":
                document.getElementById(`tick${ mar[1] }`).
                    checked = false
                document.getElementById(`tick${ mar[1] }`).
                    indeterminate = false
                {
                    let courseType = document.getElementById(`type${ mar[1] }`).
                        textContent
                    document.getElementById(`${ courseType }-chosen`).textContent =
                        parseInt(document.
                            getElementById(`${ courseType }-chosen`).
                            textContent) - 1
                }
                if (parseInt(document.getElementById("Sport-chosen").textContent) <
                    parseInt(document.getElementById("Sport-required").textContent) ||
                    parseInt(document.getElementById("Non-sport-chosen").textContent) <
                    parseInt(document.getElementById("Non-sport-required").textContent)) {
                    document.getElementById("confirmbutton").disabled = true
                }
                break
            case "M":
                document.getElementById(`selected${ mar[1] }`).
                    textContent = mar[2]
                if (
                    mar[2] === document.getElementById(`max${ mar[1] }`).textContent &&
                    !(document.getElementById(`tick${ mar[1] }`).checked)
                ) {
                    document.getElementById(`tick${ mar[1] }`).disabled = true
                } else if (gstate === 1) {
                    document.getElementById(`tick${ mar[1] }`).disabled = false
                }
                break
            case "R": /* course selection rejected */
                document.getElementById(`coursestatus${ mar[1] }`).
                    textContent = mar[2]
                document.getElementById(`coursestatus${ mar[1] }`).
                    style.color = "red"
                document.getElementById(`tick${ mar[1] }`).
                    checked = false
                document.getElementById(`tick${ mar[1] }`).
                    indeterminate = false
                if (mar[2] === "Full") {
                    document.getElementById(`tick${ mar[1] }`).
                        disabled = true
                }
                break
            case "Y": /* course selection approved */
                document.getElementById(`coursestatus${ mar[1] }`).
                    textContent = ""
                document.getElementById(`coursestatus${ mar[1] }`).
                    style.removeProperty("color")
                document.getElementById(`tick${ mar[1] }`).
                    checked = true
                document.getElementById(`tick${ mar[1] }`).
                    indeterminate = false
                {
                    let courseType = document.getElementById(`type${ mar[1] }`).
                        textContent
                    document.getElementById(`${ courseType }-chosen`).textContent =
                        parseInt(document.
                            getElementById(`${ courseType }-chosen`).
                            textContent) + 1
                }
                if (parseInt(document.getElementById("Sport-chosen").textContent) >=
                    parseInt(document.getElementById("Sport-required").textContent) &&
                    parseInt(document.getElementById("Non-sport-chosen").textContent) >=
                    parseInt(document.getElementById("Non-sport-required").textContent) &&
                    gstate === 1) {
                    document.getElementById("confirmbutton").disabled = false
                }
                break
            case "STOP":
                gstate = 0
                document.getElementById("stateindicator").textContent = "disabled"
                document.getElementById("confirmbutton").disabled = true
                document.getElementById("unconfirmbutton").disabled = true
                document.querySelectorAll(".coursecheckbox").forEach(c => {
                    c.disabled = true
                })
                break
            case "START":
                gstate = 1
                document.getElementById("unconfirmbutton").disabled = false
                document.querySelectorAll(".courseitem").forEach(c => {
                    if (c.querySelector(".selected-number").textContent !==
                        c.querySelector(".max-number").textContent ||
                        c.querySelector(".coursecheckbox").checked) {
                        c.querySelector(".coursecheckbox").disabled = false
                    }
                })
                if (parseInt(document.getElementById("Sport-chosen").textContent) >=
                    parseInt(document.getElementById("Sport-required").textContent) &&
                    parseInt(document.getElementById("Non-sport-chosen").textContent) >=
                    parseInt(document.getElementById("Non-sport-required").textContent)) {
                    document.getElementById("confirmbutton").disabled = false
                }
                document.getElementById("stateindicator").textContent = "enabled"
                break
            case "YC":
                ustate = 1
                document.querySelectorAll(".confirmed-handle").forEach(c => {
                    let handle = c.textContent
                    document.getElementById(`confirmed-name-${ handle }`).textContent = ""
                    document.getElementById(`confirmed-type-${ handle }`).textContent = ""
                    document.getElementById(`confirmed-teacher-${ handle }`).textContent = ""
                    document.getElementById(`confirmed-location-${ handle }`).textContent = ""
                    document.querySelectorAll(".coursecheckbox").forEach(d => {
                        if (d.dataset.group === handle && d.checked) {
                            document.getElementById(`confirmed-name-${ handle }`).textContent =
                                d.dataset.title
                            document.getElementById(`confirmed-type-${ handle }`).textContent =
                                d.dataset.type
                            document.getElementById(`confirmed-teacher-${ handle
                             ↪  }`).textContent =
                                d.dataset.teacher
                            document.getElementById(`confirmed-location-${ handle
                             ↪  }`).textContent =
                                d.dataset.location

                            /* TODO: break */
                        }
                    })
                })
                document.querySelectorAll(".unconfirmed").forEach(c => {
                    c.style.display = "none"
                })
                document.querySelectorAll(".confirmed").forEach(c => {
                    c.style.display = "block"
                })
                document.querySelectorAll(".neither-confirmed").forEach(c => {
                    c.style.display = "none"
                })
                break
            case "NC":
                ustate = 0
                document.querySelectorAll(".unconfirmed").forEach(c => {
                    c.style.display = "block"
                })
                document.querySelectorAll(".confirmed").forEach(c => {
                    c.style.display = "none"
                })
                document.querySelectorAll(".neither-confirmed").forEach(c => {
                    c.style.display = "none"
                })
                break
            case "RC":
                alert(mar[1])
                break
            default:
                alert(`Invalid command ${ mar[0] } received from socket. Something is wrong.`)
            }
        }
        socket.addEventListener("message", _handleMessage)
        let _handleClose = _event => {
            document.querySelectorAll(".need-connection").forEach(c => {
                c.style.display = "none"
            })
            document.querySelectorAll(".broken-connection").
                forEach(c => {
                    c.style.display = "block"
                })
        }
        socket.addEventListener("close", _handleClose)
        socket.send("HELLO")
    })

    document.querySelectorAll(".coursecheckbox").forEach(c => {
        c.addEventListener("input", () => {
            if (c.id.slice(0, 4) !== "tick") {
                alert(`${ c.id } is not in the correct format.`)
                return false
            }
            switch (c.checked) {
            case true:
                c.indeterminate = true
                document.querySelectorAll(".coursecheckbox").forEach(d => {
                    if (d.checked === true &&
                        d.dataset.group === c.dataset.group &&
                        c.id !== d.id) {
                        d.indeterminate = true
                        socket.send(`N ${ d.id.slice(4) }`)
                    }
                })
                socket.send(`Y ${ c.id.slice(4) }`)
                break
            case false:
                c.indeterminate = true
                socket.send(`N ${ c.id.slice(4) }`)
                break
            default:
                alert(`${ c.id }'s "checked" attribute is ${ c.checked } which is invalid.`)
            }
            return false
        })
    })

    document.getElementById("confirmbutton").addEventListener("click", () => {
        socket.send("YC")
    })
    document.getElementById("unconfirmbutton").addEventListener("click", () => {
        socket.send("NC")
    })

    document.querySelectorAll(".script-required").forEach(c => {
        c.style.display = "block"
    })
    document.querySelectorAll(".script-unavailable").forEach(c => {
        c.style.display = "none"
    })
})
```

2.3. style.css

```
/*
 * Copyright (c) 2024 Runxi Yu <https://runxiyu.org>
 * SPDX-License-Identifier: AGPL-3.0-or-later
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */

/*
 * TODO: Remove all uses of !important. These are obviously bad practice, but
 * it's not always trivial to get the precedence right.
 */

:root {
        --primary-bg: white;
        --primary-fg: #212529;
        --border: #ced4da;
        --anchor-underline-color: lightgray;
        --anchor-color: #0062cc;
        --theme: #0062cc;
        --theme-contrast: white;
        --box: #f2f2f2;
        --box-contrast: var(--primary-fg);
        --button: #e2e2e2;
        --button-contrast: var(--primary-fg);
        --danger: #d32535;
        --danger-contrast: white;
        --white: white;
        --white-contrast: #222222;
        --header-fg: black;
        --header-bg: #f2f2f2;
}

@media (prefers-color-scheme: dark)  {
        :root {
                --primary-bg: #212529;
                --primary-fg: #f8f9fa;
                --border: #495057;
                --anchor-underline-color: #4F4F4F;
                --anchor-color: #3294fe;
                --theme: #0062cc;
                --theme-contrast: #f8f9fa;
                --box: #30363B;
                --box-contrast: #f8f9fa;
                --button: #40464B;
                --button-contrast: #f8f9fa;
                --danger: #d32535;
                --danger-contrast: #f8f9fa;
                --white: #202020;
                --white-contrast: #f8f9fa;
                --header-fg: #f8f9fa;
                --header-bg: #30363b;
        }
}

html {
        font-family: system-ui, sans-serif;
        line-height: 1.2;
        background-color: var(--primary-bg);
        color: var(--primary-fg);
}

body {
        margin: 0;
        padding: 0;
}

main,
body > section,
.reading-width,
footer {
        margin: 1rem auto;
        padding-left: 1rem;
        padding-right: 1rem;
        max-width: 60rem;
}
```

```
/*
 * For accessibility reasons, we still want anchors to be underlined, but
 * perhaps not as profound of an underline as the default.
 */
a {
        color: var(--anchor-color);
        text-decoration: underline;
        text-decoration-color: var(--anchor-underline-color);
}

/*
 * However, although the site title will be an anchor, it should not be
 * underlined.
 */
#site-title {
        text-decoration: none;
}

/*
 * Navigation is a simple bulleted list with bullets in the middle only.
 * This should probably be revamped.
 */
nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        display: flex;
}
nav ul > li {
        display: inline-block;
}
nav ul > li:not(:last-child)::after {
        content:"\2000"
}

/*
 * The header should stick to the top of the page.
 */
header {
        position: -webkit-sticky;
        position: sticky;
        top: 0;
        left: 0;
        color: var(--header-fg);
        background-color: var(--header-bg);
        z-index: 1000;
        width: 100%;
}

/*
 * We don't want underlined anchors in the header in general, since it should
 * be obvious that things in it are links.
 */
header a {
        text-decoration: none;
        color: var(--header-fg);
}
.header-content {
        padding-left: 1rem;
        padding-right: 1rem;
        max-width: 60rem;
        display: flex;
        justify-content: space-between;
        align-items: center;
        margin: 0 auto;
}
header img {
        vertical-align: middle;
        max-height: 40px;
}
header h1 {
        font-size: 25px; /* TODO: Specifying font sizes in pixels is bad */
}

/*
 * The table, the most important element in my site design...
 */
table {
        margin-top: 0.4em;
        margin-bottom: 0.4em;
        border-collapse: collapse;
        border: 1px solid var(--border);
}
table.wide {
        width: 100%;
}
th[scope~="row"] {
        text-align: left;
}
th[scope~="col"] {
}
td {
        border: 1px solid;
        text-align: left;
        height: 1.25rem;
        border: 1px solid var(--border);
        padding: 3px 5px;
}
table.fat td {
        padding: 6px 5px;
}
td.th-like, th {
        background-color: var(--box) !important;
        border: 1px solid var(--border);
        font-weight: bold;
        padding: 3px 5px;
}
th.min, td.min {
        width: 0;
        min-width: fit-content;
        white-space: nowrap;
}

/* Workaround for https://bugzilla.mozilla.org/show_bug.cgi?id=217769 */
tbody:empty {
        display: none;
}
table > tbody:after {
        content: "";
}

/*
 * Input elements, which are usually in tables anyway
 */
textarea {
        box-sizing: border-box;
        background-color: var(--box);
        resize: vertical;
}
textarea,
input[type=text],
input[type=password] {
        font-family: sans-serif;
        font-size: smaller;
        background-color: var(--box);
        color: var(--box-contrast);
        border: none;
        padding: 0.3rem;
        width: 100%;
        box-sizing: border-box;
}
input[type=file] {
        width: fit-content;
}
td.tdinput, th.tdinput {
        padding: 0rem !important;
}
td.tdinput textarea,
td.tdinput input[type=text],
td.tdinput input[type=password],
th.tdinput textarea,
th.tdinput input[type=text],
th.tdinput input[type=password] {
        background-color: transparent !important;
}

/*
 * Button definitions.
 *
 * Each button should contain the .btn class and a .btn-type class, where type
 * is one of primary, danger, white, and normal.
 */
.btn-primary {
        background: var(--theme);
        color: var(--theme-contrast);
        border: var(--border) 1px solid;
        font-weight: bold;
}
.btn-danger {
        background: var(--danger);
        color: var(--danger-contrast);
        border: var(--border) 1px solid;
        font-weight: bold;
}
.btn-white {
        background: var(--white);
        color: var(--white-contrast);
        border: var(--border) 1px solid;
        font-weight: bold;
}
.btn-normal,
input[type=file]::file-selector-button {
        background: var(--button);
        border: var(--border) 1px solid !important;
        color: var(--button-contrast);
        font-weight: bold;
}
.btn,
input[type=submit],
input[type=file]::file-selector-button {
        display: inline-block;
        width: auto;
        min-width: fit-content;
        border-radius: 0;
        padding: .1rem .75rem;
        font-size: 0.9rem;
        transition: background .1s linear;
        cursor: pointer;
}
input[type=file].no-file-selector-button::file-selector-button {
        display: none;
}
a.btn {
        text-decoration: none;
}


/*
 * Multiple columns, flexible wrapping
 */
.multicols {
        display: flex;
        flex-direction: row;
        @media(max-width: 50rem) {
                flex-wrap: wrap;
                gap: 0rem;
        }
        gap: 2rem;
        align-items: stretch;
}

.multicols div {
        min-width: 18em;
        /* max-width: 40rem; */
        width: 100%;
        margin-left: auto;
        margin-right: auto;
}

/*
 * Spanning elements across a flex container with equal space in between
 */
.flex-justify {
        display: flex;
        justify-content: space-between;
        align-items: center;
        margin: 0 auto;
        border: none;
}

.message-box {
        margin: auto;
        max-width: 30rem;
        border: solid 1px var(--border);
        background-color: var(--box);
        padding: 0rem 1rem;
}

table.table-of-courses {
        width: 100%;
}

:disabled {
        background: repeating-linear-gradient(
                135deg,
                grey,
                grey 5px,
                dimgrey 5px,
                dimgrey 10px
        );
}


/*
```

```
 * .need-connection is the content that should actually display when we are
 * connected via WebSocket. The JavaScript would change display from none to
 * block when fully connected to WebSocket.
 */
.need-connection {
        display: none;
}

/*
 * Same for script-required, though the JavaScript hides this as soon as it's
 * loaded.
 */
.script-required {
        display: none;
}

/*
 * .broken-connection displays a message telling users to refresh the page,
 * after their WebSocket connection breaks. It should be hidden by default.
 */
.broken-connection {
        display: none;
}

.unconfirmed {
        display: none;
}
.confirmed {
        display: none;
}
.neither-confirmed {
        display: block;
}

/*
 * This site heavily uses CSS styling to display and hide messages, so by
 * default we put a big warning about CSS being broken, which disappears
 * once the main CSS, i.e. this file, is completely loaded. Therefore it's
 * probably best to put this at the bottom of this file.
 */
.broken-styling-warning {
        display: none;
}
```

3. HTML templates

3.1. login.html

```
{{- define "login" -}}
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      Authentication required &ndash; CCA Selection System
    </title>
    <link rel="stylesheet" href="/static/style.css" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="description" content="Authentication Page for the YK Pao School CCA Selection
     ↪  System" />
    <style>
      #login-box {
        margin: auto;
        max-width: 30rem;
      }
    </style>
  </head>
  <body>
    <header>
      <div class="header-content">
        <div class="header-left">
          <h1><a id="site-title" href="./">CCA Selection System</a></h1>
        </div>
        <div class="header-middle">
          <nav>
            <ul>
              <li>
                <a href="./">Home</a>
              </li>
              <li>
                <a href="./docs/">Docs</a>
              </li>
              <li>
                <a href="./iadocs/">IA</a>
              </li>
              <li>
                <a href="./src/">Source</a>
              </li>
            </ul>
          </nav>
        </div>
        <div class="header-right">
          <p>Unauthenticated</p>
        </div>
      </div>
    </header>
    <main>
      <div id="login-box">
        <p>
        {{- if eq .Notes "" -}}{{- .Notes -}}{{- end -}}
        </p>
        <p>
          You have not authenticated. You must sign in to use this service.
        </p>
        <p>
          <a class="btn btn-primary" href="{{- .AuthURL -}}">Sign in with Microsoft</a>
        </p>
      </div>
    </main>
  </body>
</html>
{{- end -}}
```

3.2. staff.html

```
{{- define "staff" -}}
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      Staff Home &ndash; CCA Selection System
    </title>
    <link rel="stylesheet" href="/static/style.css" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="description" content="YK Pao School CCA Selection System" />
  </head>
  <body>
    <div style="font-size: 150%; color: red; font-weight: bold;"
     ↪ class="broken-styling-warning">
      The fact that you see this message means that the CSS styling information for this site
       ↪ is not loading correctly, and usability would be severely impacted. Check your
       ↪ network connection, and if this issue persists, you should contact the system
       ↪ administrator.
    </div>
    <header>
      <div class="header-content">
        <div class="header-left">
          <h1><a id="site-title" href="./">CCA Selection System</a></h1>
        </div>
        <div class="header-middle">
          <nav>
            <ul>
          <li>
            <a href="./">Home</a>
          </li>
          <li>
            <a href="./docs/">Docs</a>
          </li>
          <li>
            <a href="./iadocs/">IA</a>
          </li>
          <li>
            <a href="./src/">Source</a>
          </li>
        </ul>
      </nav>
    </div>
    <div class="header-right">
      <p>{{- .Name }} (Staff)</p>
    </div>
  </div>
</header>
<div class="reading-width" id="wip-notice">
  <p>
  This site is still a work in progress and may contain bugs! Please contact <a
   ↪  href="mailto:s22537@stu.ykpaoschool.cn">Runxi Yu</a> for any issues.
  </p>
</div>
<div class="reading-width">
  <p><a href="./export/choices" class="btn-normal btn">Export all choices as a
   ↪  spreadsheet</a></p>
  <p><a href="./export/students" class="btn-normal btn">Export student confirmed status as
   ↪  a spreadsheet</a></p>
  {{- if ge .State 1 }}
  <p><a href="./state/0" class="btn-danger btn">Disable student access</a></p>
  {{- if ge .State 2 }}
  <p><a href="./state/1" class="btn-danger btn">Stop course selections</a></p>
  {{- else }}
  <p><a href="./state/2" class="btn-primary btn">Start course selections</a></p>
  {{- end }}
  {{- else }}
  <p><a href="./state/1" class="btn-primary btn">Enable student access</a></p>
  {{- end }}
  <table class="table-of-courses">
    <colgroup>
      <col style="width: 5%;" />
      <col style="width: 5%;" />
      <col style="width: 5%;" />
      <col/>
      <col style="width: 15%;" />
      <col style="width: 15%;" />
      <col style="width: 15%;" />
    </colgroup>
    <thead>
      <tr>
        <th scope="col">ID</th>
        <th scope="col">Used</th>
        <th scope="col">Max</th>
        <th scope="col">Name</th>
        <th scope="col">Type</th>
        <th scope="col">Teacher</th>
        <th scope="col">Location</th>
      </tr>
      <tr>
        <th colspan="7" class="tdinput">
          <input type="text" id="search" placeholder="Search..." />
        </th>
      </tr>
    </thead>
    <tbody>
      {{- range .Groups }}
      <tr><th colspan="7">{{ .Name }}</th></tr>
      {{- range .Courses }}
      <tr class="courseitem" id="course{{.ID}}" data-group="{{.Group}}">
        <th scope="row">
          {{.ID}}
        </th>
        <td>
          <span id="selected{{.ID}}">{{.Selected}}</span>
        </td>
        <td>
          <span id="max{{.ID}}">{{.Max}}</span>
        </td>
        <td>{{.Title}}</td>
        <td id="type{{.ID}}">{{.Type}}</td>
        <td>{{.Teacher}}</td>
        <td>{{.Location}}</td>
      </tr>
      {{- end }}
      {{- end }}
    </tbody>
    {{- if eq .State 0 }}
    <tfoot>
      <tr>
        <td class="th-like" colspan="7">
          <form method="POST" enctype="multipart/form-data" action="/newcourses">
            <div class="flex-justify">
              <div class="left">
              </div>
              <div class="right">
                <input title="Upload course list (CSV)" type="file" id="coursecsv"
                 ↪  name="coursecsv" accept=".csv" />
                <input type="submit" value="Delete all choices and reset courses"
                 ↪  class="btn btn-danger" />
              </div>
            </div>
          </form>
        </td>
```

```
          </tr>
        </tfoot>
        {{- end }}
      </table>
    </div>
    <script>
      document.addEventListener("DOMContentLoaded", () => {
        const search = document.getElementById("search")
        search.addEventListener("input", () => {
          const s = search.value.toLowerCase().trim().normalize('NFD')
          document.querySelectorAll(".courseitem").forEach(c => {
            c.hidden = (!c.textContent.toLowerCase().normalize('NFD').includes(s)) && (s.length
             ↪   > 0)
          })
        })
      })
    </script>
  </body>
</html>
{{- end -}}
```

3.3. student.html

```
{{- define "student" -}}
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      {{ .Name }} &ndash; CCA Selection System
    </title>
    <link rel="stylesheet" href="/static/style.css" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="description" content="YK Pao School CCA Selection System" />
  </head>
  <body>
    <div style="font-size: 150%; color: red; font-weight: bold;"
     ↪  class="broken-styling-warning">
      The fact that you see this message means that the CSS styling information for this site
       ↪  is not loading correctly, and usability would be severely impacted. Check your
       ↪  network connection, and if this issue persists, you should contact the system
       ↪  administrator.
    </div>
    <header>
      <div class="header-content">
        <div class="header-left">
          <h1><a id="site-title" href="./">CCA Selection System</a></h1>
        </div>
        <div class="header-middle">
          <nav>
            <ul>
              <li>
            <a href="./">Home</a>
          </li>
          <li>
            <a href="./docs/">Docs</a>
          </li>
          <li>
            <a href="./iadocs/">IA</a>
          </li>
          <li>
            <a href="./src/">Source</a>
          </li>
        </ul>
      </nav>
    </div>
    <div class="header-right">
      <p>{{- .Name }} ({{ .Department -}})</p>
    </div>
  </div>
</header>
<div class="reading-width" id="wip-notice">
  <p>
  This site is still a work in progress and may contain bugs! Please contact <a href="mailto:sj-cca@ykpaoschool.cn">the CCA department</a> for CCA selection issues or <a href="mailto:s22537@stu.ykpaoschool.cn">Runxi Yu</a> for website issues.
  </p>
</div>
<div class="script-unavailable message-box">
  <p>
  JavaScript is required to use this page. One of the following conditions are present:
  </p>
  <ul>
    <li>
      The JavaScript hasn&rsquo;t finished loading.
    </li>
    <li>
      JavaScript is not supported by your browser.
    </li>
    <li>
      JavaScript is disabled/blocked.
    </li>
  </ul>
  <p>
  All JavaScript hosted on this site are licensed under the GNU Affero General Public License, version 3.0 or any later version.
  </p>
</div>
<div class="script-required">
  <div class="before-connection message-box">
    <p>
    Attempting to establish an WebSocket connection.
    </p>
    <p>
```

```html
  If this message does not disappear soon, it means that one of the following conditions
   ↪  are true:
  </p>
  <ul>
    <li>
      Your browser does not <a href="https://caniuse.com/websockets">support
       ↪  WebSocket</a>, or they are being blocked.
    </li>
    <li>
      The server is overloaded or encountered an error.
    </li>
    <li>
      The network is over-saturated, or you just have a bad network.
    </li>
  </ul>
</div>
<div class="broken-connection message-box">
  <p>
  Your WebSocket connection has been closed. This means that one of the following
   ↪  occurred:
  </p>
  <ul>
    <li>
      You logged in on another session.
    </li>
    <li>
      CCA staff disabled the student portal.
    </li>
    <li>
      The network is over-saturated and connections cannot be maintained.
    </li>
    <li>
      There was an internal server error that closed your connection.
    </li>
    <li>
      There was an error in the JavaScript running on your browser.
    </li>
  </ul>
  <p>
  If you believe that your networking is in good condition, you may wish to report this
   ↪  to the system administrator.
  </p>
  <p>
  <a href="javascript:window.location.reload(true)" class="btn btn-primary">Reconnect</a>
  <!-- TODO: Add a quicker reconnect -->
  </p>
</div>
<div class="need-connection">
  <div class="reading-width">
    <p>
    Course selections are <span style="font-weight: bold;"
     ↪  id="stateindicator">disabled</span>.
    </p>
```

```
<div class="neither-confirmed">
  <p>
  (Still loading...)
  </p>
</div>
<div class="confirmed">
  <table class="table-of-choices">
    <thead>
      <tr>
        <th scope="col">Group</th>
        <th scope="col">Name</th>
        <th scope="col">Type</th>
        <th scope="col">Teacher</th>
        <th scope="col">Location</th>
      </tr>
    </thead>
    <tbody>
      {{- range .Groups }}
      <tr>
        <th class="confirmed-handle" id="confirmed-handle-{{ .Handle }}"
         ↪  scope="row">{{ .Handle }}</th>
        <td id="confirmed-name-{{ .Handle }}"></td>
        <td id="confirmed-type-{{ .Handle }}"></td>
        <td id="confirmed-teacher-{{ .Handle }}"></td>
        <td id="confirmed-location-{{ .Handle }}"></td>
      </tr>
      {{- end }}
    </tbody>
    <tfoot>
      <tr>
        <td class="th-like" colspan="7">
          <div class="flex-justify">
            <div class="left">
            </div>
            <div class="right">
              <button id="unconfirmbutton" class="btn-danger btn"
               ↪  disabled>Unconfirm</button>
            </div>
          </div>
        </td>
      </tr>
    </tfoot>
  </table>
</div>
<div class="unconfirmed">
  <table class="table-of-courses">
    <colgroup>
      <col style="width: 5%;" />
      <col style="width: 5%;" />
      <col style="width: 5%;" />
      <col/>
      <col style="width: 15%;" />
      <col style="width: 15%;" />
      <col style="width: 15%;" />
    </colgroup>
    <thead>
      <tr>
        <th scope="col">Tick</th>
        <th scope="col">Used</th>
        <th scope="col">Max</th>
        <th scope="col">Name</th>
        <th scope="col">Type</th>
        <th scope="col">Teacher</th>
        <th scope="col">Location</th>
      </tr>
      <tr>
        <th colspan="7" class="tdinput">
          <input type="text" id="search" placeholder="Search..." />
        </th>
      </tr>
    </thead>
    <tbody>
      {{- range .Groups }}
      <tr><th colspan="7">{{ .Name }}</th></tr>
      {{- range .Courses }}
      <tr class="courseitem" id="course{{.ID}}" data-group="{{.Group}}">
        <th style="font-weight: normal;" scope="row">
          <input aria-label="Enroll in course" class="coursecheckbox" type="checkbox"
              ↪  id="tick{{.ID}}" name="tick{{.ID}}" value="tick{{.ID}}"
              ↪  data-group="{{.Group}}" data-type="{{.Type}}" data-title="{{.Title}}"
              ↪  data-teacher="{{.Teacher}}" data-location="{{.Location}}" disabled
              ↪  ></input>
          <span id="coursestatus{{.ID}}"></span>
        </th>
        <td>
          <span class="selected-number" id="selected{{.ID}}">{{.Selected}}</span>
        </td>
        <td>
          <span class="max-number" id="max{{.ID}}">{{.Max}}</span>
        </td>
        <td>{{.Title}}</td>
        <td id="type{{.ID}}">{{.Type}}</td>
        <td>{{.Teacher}}</td>
        <td>{{.Location}}</td>
      </tr>
      {{- end }}
      {{- end }}
    </tbody>
    <tfoot>
      <tr>
        <td class="th-like" colspan="7">
          <div class="flex-justify">
            <div class="left">
              Sport: <span id="Sport-chosen">0</span>/<span id="Sport-required">{{
               ↪  .Required.Sport }}</span>,
                        Non-sport: <span id="Non-sport-chosen">0</span>/<span
                         ↪  id="Non-sport-required">{{ .Required.NonSport }}</span>
                      </div>
                      <div class="right">
                        <button id="confirmbutton" class="btn-primary btn"
                         ↪  disabled>Confirm</button>
                      </div>
                    </div>
                  </td>
                </tr>
              </tfoot>
            </table>
          </div>
        </div>
      </div>
    </div>
    <script src="static/student.js"></script>
    <script>
      document.addEventListener("DOMContentLoaded", () => {
        const search = document.getElementById("search")
        search.addEventListener("input", () => {
          const s = search.value.toLowerCase().trim().normalize('NFD')
          document.querySelectorAll(".courseitem").forEach(c => {
            c.hidden = (!c.textContent.toLowerCase().normalize('NFD').includes(s)) && (s.length
             ↪  > 0)
          })
        })
      })
    </script>
  </body>
</html>
{{- end -}}
```

3.4. student_disabled.html

```
{{- define "student_disabled" -}}
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      {{ .Name }} &ndash; CCA Selection System
    </title>
    <link rel="stylesheet" href="/static/style.css" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="description" content="YK Pao School CCA Selection System" />
  </head>
  <body>
    <div style="font-size: 150%; color: red; font-weight: bold;"
     ↪  class="broken-styling-warning">
```

```
      The fact that you see this message means that the CSS styling information for this site
       ↪  is not loading correctly, and usability would be severely impacted. Check your
       ↪  network connection, and if this issue persists, you should contact the system
       ↪  administrator.
    </div>
    <header>
      <div class="header-content">
        <div class="header-left">
          <h1><a id="site-title" href="./">CCA Selection System</a></h1>
        </div>
        <div class="header-middle">
          <nav>
            <ul>
              <li>
                <a href="./">Home</a>
              </li>
              <li>
                <a href="./docs/">Docs</a>
              </li>
              <li>
                <a href="./iadocs/">IA</a>
              </li>
              <li>
                <a href="./src/">Source</a>
              </li>
            </ul>
          </nav>
        </div>
        <div class="header-right">
          <p>{{- .Name }} ({{ .Department -}})</p>
        </div>
      </div>
    </header>
    <div class="reading-width" id="wip-notice">
      <p>
      This site is still a work in progress and may contain bugs! Please contact <a
       ↪  href="mailto:s22537@stu.ykpaoschool.cn">Runxi Yu</a> for any issues.
      </p>
    </div>
    <div class="reading-width">
      <p>
      Student access is currently disabled.
      </p>
      <p>
      Please check back later.
      </p>
    </div>
  </body>
</html>
{{- end -}}
```

4. Build system and auxiliary scripts

4.1. Makefile

```
# TODO: Use some variables to clean up the massive documentation file specifiers

.PHONY: cca default minifier iadocs docs build_iadocs build_docs

default: dist/cca docs iadocs

cca: dist/cca

docs: dist/docs/admin_handbook.html dist/docs/handbook.css dist/docs/cca.scfg.example

iadocs: dist/iadocs/index.html dist/iadocs/cover_page.htm dist/iadocs/appendix.pdf
 ↪  dist/iadocs/crita_planning.pdf dist/iadocs/critb_design.pdf
 ↪  dist/iadocs/critb_recordoftasks.htm dist/iadocs/critc_development.pdf
 ↪  dist/iadocs/critd_functionality.pdf dist/iadocs/crite_evaluation.pdf

# Final binary which tries to embed stuff
dist/cca: go.* *.go build/static/style.css build/static/student.js templates/*
 ↪  build/docs/admin_handbook.html build/docs/handbook.css build/docs/cca.scfg.example
 ↪  build/iadocs/index.html build/iadocs/cover_page.htm build/iadocs/appendix.pdf
 ↪  build/iadocs/crita_planning.pdf build/iadocs/critb_design.pdf
 ↪  build/iadocs/critb_recordoftasks.htm build/iadocs/critc_development.pdf
 ↪  build/iadocs/critd_functionality.pdf build/iadocs/crite_evaluation.pdf .editorconfig
 ↪  .gitignore .gitattributes scripts/* sql/* docs/* iadocs/* README.md LICENSE Makefile
        mkdir -p dist
        go build -o $@

# Documentation
dist/docs/%: build/docs/%
        mkdir -p dist/docs
        cp $< $@
build/docs/%.html: docs/%.html
        mkdir -p build/docs
        minify --html-keep-end-tags --html-keep-document-tags -o $@ $<
build/docs/handbook.css: docs/handbook.css
        mkdir -p build/docs
        minify -o $@ $<
build/docs/cca.scfg.example: docs/cca.scfg.example
        mkdir -p build/docs
        cp $< $@

# IA documentation
dist/iadocs/%.pdf: build/iadocs/%.pdf
        mkdir -p dist/iadocs
        cp $< $@
dist/iadocs/%.htm: build/iadocs/%.htm
        mkdir -p dist/iadocs
        cp $< $@
dist/iadocs/%.html: build/iadocs/%.html
        mkdir -p dist/iadocs
        cp $< $@
build/iadocs/%.htm: iadocs/%.htm
        mkdir -p build/iadocs
        minify --html-keep-end-tags --html-keep-document-tags -o $@ $<
build/iadocs/index.html: build/iadocs/cover_page.htm
        cp $< $@
build/iadocs/%.pdf: iadocs/%.tex build/iadocs/header.texinc
        mkdir -p build/iadocs
        lualatex -interaction batchmode -output-directory=build/iadocs $<
        lualatex -interaction batchmode -output-directory=build/iadocs $<
build/iadocs/appendix.pdf: iadocs/appendix.tex build/iadocs/source.gen build/iadocs/agpl.texinc
        mkdir -p build/iadocs
        lualatex -interaction batchmode -shell-escape -output-directory=build/iadocs $<
        lualatex -interaction batchmode -shell-escape -output-directory=build/iadocs $<
build/iadocs/source.gen: go.* *.go frontend/*.css frontend/*.js templates/* scripts/latexify-source.sh docs/* sql/* scripts/* iadocs/*.tex iadocs/*.texinc
        mkdir -p build/iadocs
        scripts/latexify-source.sh
build/iadocs/%.texinc: iadocs/%.texinc
        mkdir -p build/iadocs
        cp $< $@

# Temporary files in build/ to be embedded into the final binary
build/static/style.css: frontend/style.css
        mkdir -p build/static
        minify -o $@ $<
build/static/student.js: frontend/student.js
        mkdir -p build/static
        minify -o $@ $<

# External dependencies
minifier:
        go install github.com/tdewolff/minify/v2/cmd/minify@latest
```

4.2. deploy.sh

```
#!/bin/bash

set -xeu

home_dir="${BASH_SOURCE[0]%/*}/.."
cd "$home_dir"

make build/static/student.js
gsed -i 's;wss://cca.runxiyu.org/ws;wss://dev.runxiyu.org/ws;g' build/static/student.js
```

```
mv dist/cca dist/ccae
GOOS=linux GOARCH=amd64 make dist/cca

rm build/static/student.js

rsync -v dist/cca root@runxiyu.org:/srv/dev/cca

ssh root@runxiyu.org pkill cca

rm dist/cca
mv dist/ccae dist/cca
```

4.3. eslint.sh

```
#!/bin/sh

set -e

cd frontend

eslint . "$@"
```

4.4. latexify-source.sh

```
#!/bin/bash

set -eu

targetfile="$(realpath -- build/iadocs/source.gen)"

printf '\n' > "$targetfile"

printfile() {
        lang="$1"
        tabsize="$2"
        base="$3"
        shift 3
        for i in "$@"
        do
                printf '\\section{%s}\n' "$(sed 's/_/\\_/g' <<< "$i")" >> "$targetfile"
                printf '\\inputminted[breaklines, tabsize=%s]{%s}{%s/%s}\n' "$tabsize" "$lang"
                 ↪  "$base" "$i" >> "$targetfile"
        done
}

printf '\\chapter{Backend source code}\n' >> "$targetfile"
printfile go 8 ./ *.go
printfile text 8 ./ go.*

printf '\\chapter{Frontend source code}\n' >> "$targetfile"
```

```
cd frontend
printfile javascript 4 ./frontend *.js
printfile css 8 ./frontend *.css

printf '\\chapter{HTML templates}\n' >> "$targetfile"
cd ../templates
printfile html 2 ./templates *.html

printf '\\chapter{Build system and auxiliary scripts}\n' >> "$targetfile"
cd ..
printfile makefile 8 ./ Makefile
cd scripts
printfile bash 8 ./scripts *.sh

printf '\\chapter{SQL scripts}\n' >> "$targetfile"
cd ../sql
printfile postgresql 8 ./sql *.sql

printf '\\chapter{Production documentation}\n' >> "$targetfile"
cd ../docs
printfile html 2 ./docs *.html
printfile css 8 ./docs *.css
printfile text 8 ./docs *.csv cca.scfg.example

printf '\\chapter{IA documentation}\n' >> "$targetfile"
cd ../iadocs
printfile latex 8 ./iadocs *.tex *.texinc
```

4.5. lint.sh

```
#!/bin/sh
set -e
golangci-lint run --color=always --enable-all
 ↪  --disable=wsl,funlen,exportloopref,gomnd,execinquery,godox,lll,gochecknoglobals,depguard,cyclop,gos
```

5. SQL scripts

5.1. drop.sql

```
DROP TABLE choices;
DROP TABLE users;
DROP TABLE courses;
DROP TABLE misc;
```

5.2. schema.sql

```
CREATE TABLE courses (
        id INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
        nmax INTEGER NOT NULL,
        title TEXT NOT NULL,
        teacher TEXT NOT NULL,
        location TEXT NOT NULL,
        ctype TEXT NOT NULL,
        cgroup TEXT NOT NULL,
        course_id TEXT NOT NULL,
        section_id TEXT NOT NULL
);
CREATE TABLE users (
        id TEXT PRIMARY KEY NOT NULL, -- should be UUID
        name TEXT NOT NULL,
        email TEXT NOT NULL,
        department TEXT NOT NULL,
        session TEXT,
        expr BIGINT, -- seconds
        confirmed BOOLEAN NOT NULL
);
CREATE TABLE choices (
        PRIMARY KEY (userid, courseid),
        seltime BIGINT NOT NULL, -- microseconds
        userid TEXT NOT NULL, -- should be UUID
        FOREIGN KEY(userid) REFERENCES users(id),
        courseid INTEGER NOT NULL,
        FOREIGN KEY(courseid) REFERENCES courses(id),
        UNIQUE (userid, courseid)
);
CREATE TABLE misc (
        key TEXT PRIMARY KEY NOT NULL,
        value INTEGER NOT NULL
);
```

6. Production documentation

6.1. admin_handbook.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>CCA Admin Handbook</title>
    <link rel="stylesheet" href="./handbook.css" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
  </head>
  <body>
    <header>
      <h1>
        CCA Admin Handbook
      </h1>
    </header>
    <main>
      <h2>Introduction</h2>
      <p>
      This handbook guides you in installing, configuring, and managing your CCA Selection
       ↪  System (CCASS) instance.
      </p>
      <h2>Downloading</h2>
      <p>
      You may obtain a stable or development version. The stable version is recommended for
       ↪  production.
      </p>
      <ul>
        <li>To obtain a stable version, go to the <a
         ↪  href="https://git.sr.ht/~runxiyu/cca/refs">release page</a> and download the latest
         ↪  version that is not a pre-release.</li>
        <li>To obtain an unstable development version, clone the development repository at <a
         ↪  href="https://git.sr.ht/~runxiyu/cca"><code>https://git.sr.ht/~runxiyu/cca</code></a>,
         ↪  or download the latest development snapshot&rsquo;s tarball at <a
         ↪  href="https://git.runxiyu.org/ykps/cca.git/snapshot/cca-master.tar.gz"><code>https://git.ru
      </ul>
      <h2>External dependencies</h2>
      <p>
      You need a <a href="https://go.dev">Go</a> toolchain, <a
       ↪  href="https://pygments.org/">Pygments</a>, <a
       ↪  href="https://www.gnu.org/software/make/">GNU make</a>, <a
       ↪  href="https://tug.org/texlive/">TeX Live</a> and <a
       ↪  href="https://github.com/tdewolff/minify">minify</a>. If you have everything else,
       ↪  you could install minify via <code>make minifier</code>, which would build and
       ↪  install it with your Go toolchain.
      </p>
      <p>
      The Go toolchain will fetch more dependencies. You may wish to set a Go proxy (such as
       ↪  via <code>export GOPROXY='https://goproxy.io'</code>) if it stalls or is too slow.
       ↪  This is likely necessary for users in Mainland China due to firewall restrictions.
      </p>
      <h2>Building</h2>
      <p>Just type <code>make</code>.</p>
      <p>
      The built files will appear in <code>dist/</code>. The binary, with all runtime resources
       ↪  other than the configuration file embedded, is located at <code>dist/cca</code>. A
       ↪  minified copy of the documentation, including a sample configuration file, is located
       ↪  at <code>dist/docs/</code>.
      </p>
      <h2>Configuration</h2>
      <p>
      Copy <a href="./cca.scfg.example">the example configuration file</a> to
       ↪  <code>cca.scfg</code> in the working directory where you intend to run CCASS. Then
       ↪  edit it according to the comments, though you may wish to pay attention to the
       ↪  following:
      </p>
      <ul>
        <li>CCASS natively supports serving over clear text HTTP or over HTTPS. HTTPS is
         ↪  required for production setups as Microsoft Entra ID does not allow clear-text HTTP
         ↪  redirect URLs for non-<code>localhost</code> access.</li>
        <li>Note that CCASS is designed to be directly exposed to clients due to the lacking
         ↪  performance of standard reverse proxy setups, although there is nothing that
         ↪  otherwise prevents it from being used behind a reverse proxy. Reverse proxies must
         ↪  forward WebSocket connection upgrade headers for all requests to the
         ↪  <code>/ws</code> endpoint.</li>
        <li>You must <a
         ↪  href="https://portal.azure.com/#view/Microsoft_AAD_RegisteredApps/ApplicationsListBlade">cr
         ↪  an app registration on the Azure portal</a> and complete the corresponding
         ↪  configuration options.</li>
        <li><code>perf/sendq</code> should be set to roughly the number of expected students
         ↪  making concurrent choices.</li>
      </ul>
      <h2>Database setup</h2>
      <p>
      A working PostgreSQL setup is required. It is recommended to set up UNIX socket
       ↪  authentication and set the user running CCASS as the database owner while creating
       ↪  the database.
      </p>
      <p>
      Before first run, run <code>psql <i>dbname</i> -f sql/schema.sql</code> to create the
       ↪  database tables, where <code><i>dbname</i></code> is the name of the database.
      </p>
      <p>
      Using the same database for different versions of CCASS is currently unsupported,
       ↪  although it should be trivial to manually migrate the database.
      </p>
    </main>
  </body>
```

```
</html>
```

6.2. handbook.css

```
:root {
	--theme: #0062cc;
	--anchor-underline-color: lightgray;
}

html {
	font-family: system-ui, sans-serif;
	padding: 0rem 10px;
	margin: auto;
	max-width: 50rem;
}

a {
	color: var(--theme);
	text-decoration: underline;
	text-decoration-color: var(--anchor-underline-color);
}
```

6.3. courses_example.csv

```
Title,Max,Teacher,Location,Type,Group,Section ID,Course ID
Floorball,65535,Chen Xiaojia,Gym,Sport,MW3,FL,FL
Fakeball,65535,Foo Bar,Gym,Sport,MW1,FK,FK
Chamber Music,65535,Bar Baz,Music Rooms,Non-sport,MW1,CM,CM
Programming Club,65535,Foo Baz,Library,Non-sport,MW1,PC,PC
Flag Football,65535,Ding Zhaoyuan,Pitch,Sport,MW2,FF,FF
Table Tennis,65535,Somebody,Table Tennis Room,Sport,MW2,TT,TT
Spinning,65535,Somebody,Spinning Room,Sport,MW2,SP,SP
Some Music,65535,Eeeee,Music Rooms,Non-sport,TT1,SM,SM
Some More Music,65535,Eeeee,Music Rooms,Non-sport,TT1,SMM,SMM
Math and Computer Science,65535,Jeff Zhang,2307,Non-sport,TT2,MC,MC
Basketball,65535,Somebody,Gym,Sport,TT2,BB,BB
Chinese Drama,65535,Monica Chen (?),Black Box (?),Non-sport,TT2,CD,CD
Actually Flag Football,30,Hmm,Pitch,Sport,TT3,AFF,AFF
```

6.4. cca.scfg.example

```
# Which URL are we accessible at? This is used to determine the redirect URL
# and some user-accessible URLs.
url http://localhost:5555

# Should we run in production mode? This causes the Secure flag to be set on
# cookies and may come with other production-related changes in the future.
prod false
```

```
listen {
        # Which protocol are we listening for? Currently only ”http” is
        # supported because it is difficult to configure FastCGI to work with
        # WebSockets.
        proto http

        # Which network backend should we use? This is usually set to ”tcp”
        # for plain TCP, and ”unix” for UNIX domain sockets.
        net tcp

        # What is the address we should listen at? This is usually set to
        # something like ”:5555” for TCP on all interfaces, and a file path for
        # UNIX domain sockets.
        addr 127.0.0.1:5555

        # Which transport should we use? Currently only ”plain” and ”tls” are
        # supported.
        trans plain

        # If ”trans” is set to ”tls”, this block must be configured:
        tls {
                # Where is the file containing the TLS certificate? You should
                # use a certificate that includes the entire CA chain, which
                # should usually be called ”fullchain.pem” for certificates
                # obtained from LE.
                cert /etc/letsencrypt/live/cca.runxiyu.org/fullchain.pem

                # Where is the file containing the TLS private key?
                key /etc/letsencrypt/live/cca.runxiyu.org/privkey.pem
        }
}

db {
        # What type of database should we use? Currently, only ”postgres” is
        # supported.
        type postgres

        # What is the connection string to database?
        # Example: postgresql:///cca?host=/var/run/postgresql
        conn postgresql:///cca?host=/var/run/postgresql
}

auth {
        # What is our OAUTH2 client ID?
        client e8101cb5-84a3-49d7-860b-e5a75e63219a

        # What is the OAUTH 2.0 authorize endpoint?
        authorize
         ↪  https://login.microsoftonline.com/ddd3d26c-b197-4d00-a32d-1ffd84c0c295/oauth2/v2.0/authoriz

        # What is the OAUTH 2.0 token endpoint?
        token
         ↪  https://login.microsoftonline.com/ddd3d26c-b197-4d00-a32d-1ffd84c0c295/oauth2/v2.0/token
```

```
	# What is the URL to the JSON Web Key Set?
	jwks https://login.microsoftonline.com/common/discovery/keys

	# What is the client secret? Certificates are not supported yet.
	secret something

	# How long, in seconds, should cookies last?
	expr 604800
}

# The following block contains some tweaks for performance.
perf {
	# How many arguments' space should we initially allocate for each
	# message?
	msg_args_cap 4

	# How many bytes should we initially allocate for each argument in a
	# message?
	msg_bytes_cap 5

	# How long should we wait to complete reading HTTP headers, before we
	# time out? Note that a large value may cause the server to be
	# vulnerable to Slow Loris attacks.
	read_header_timeout 5

	# The number propagation interval per course per connection is
	# ((course count * connection count) >> usem_delay_shift_bits)
	# milliseconds. You may configure it here. A smaller value (i.e. longer
	# delay) could cause more latency in how the numbers update, but a
	# larger value (i.e. shorter delay) could cause too much lock
	# contention and degrade the system usability overall.
	usem_delay_shift_bits 5

	# Should we send a course's member count to a user as soon as they
	# choose the course? Setting this to true may provide a better
	# user experience but would have a major performance impact.
	propagate_immediate true

	# How long should the send queue be, for messages sequentially
	# propagated through a queue, rather than usems?
	senq 10
}

# Minimum course requirements for each year group
req {
	y9 {
		sport 2
		non_sport 1
	}
	y10 {
		sport 2
		non_sport 1
```

```
        }
        y11 {
                sport 1
                non_sport 1
        }
        y12 {
                sport 1
                non_sport 1
        }
}
```

7. IA documentation

7.1. appendix.tex

```
\documentclass[numbers=endperiod, parskip=half-]{scrreprt}

\title{Appendix}
\subject{IBDP Computer Science Internal Assessment}
\author{Runxi Yu}
\date{\today}

\usepackage{fontspec}
\setmainfont{TeX Gyre Termes}
\setsansfont{TeX Gyre Heros}
\setmonofont{Inconsolatazi4}
\usepackage{unicode-math}
\setmathfont{TeX Gyre Termes Math}
\usepackage{microtype}

\usepackage{xcolor}
\usepackage[colorlinks, allcolors={white!40!blue!50!black}]{hyperref}

\usepackage{minted}

\usepackage{geometry}
\geometry{
        margin=0.5in,
        includefoot,
}
\usepackage{multicol}

\makeatletter
% https://tex.stackexchange.com/questions/390842
% Two-column table of contents
\let\@starttocorg\@starttoc
\def\@starttoc#1{%
        \begin{multicols*}{2}%
                \@starttocorg{#1}%
        \end{multicols*}}%
\makeatother

\begin{document}
\maketitle

\tableofcontents

\input{source.gen}
```

```
\chapter{Copyright license}

\input{agpl.texinc}

\end{document}
```

7.2. crita_planning.tex

```
\input{header.texinc}

\title{Criterion A: Planning}

\begin{document}
\maketitle

\section{The scenario}

My school offers co-curricular activities (CCAs) to students after academic
periods from Monday to Thursday. Most of those activities are relatively
limited in space, and my campus has approximately 586 students across 4 year
groups.

Before I joined the school, the CCA department has already been using
\href{https://www.schoolsbuddy.com}{SchoolsBuddy} as a CCA selection interface
for students. However, SchoolsBuddy has the following problems when used at the
scale and to the requirements of our school:

\begin{itemize}
        \item When too many students attempt to choose one course at a time,
                most of their attempts would be confirmed and recorded in the
                database, and the CCA department staff must notice them
                after-the-fact that their attempt at choosing a course was
                unsuccessful. At that point, there would be much fewer other
                CCAs with available places too, leaving the student with
                insufficient choices---after they have been told by the system
                that their place has already been confirmed.
        \item The only way to sign in to SchoolsBuddy, at least for our school's
                configuration, is to log on to PowerSchool and click the
                SchoolsBuddy Single Sign-on link. This generally works well,
                except for the fact that PowerSchool only allows approximately
                300 simultaneous sessions, which means that there would be
                approximately 200 students unable to choose CCAs on
                SchoolsBuddy.
        \item Even after logging in, the SchoolsBuddy web page is extremely
                bloated. It takes 16.5 MiB to get completely load the
                SchoolsBuddy home page,        and an additional 8 MiB per additional
                page. This takes about 8 seconds on a relatively good connection
                without congestion. But when CCA selection starts and everyone
                is trying to use student WiFi to log on at the same time,
                loading each page could take well more than a minute.
        \item Some parts of the interface is unintuitive to students. We have
```

```
                6 CCA slots: MW1, MW2, MW3, TT1, TT2, and TT3, which basically
                mean ``CCA Period $n$''.
        \item The selection system does not enforce CCA hours requirements, and
                the CCA office's staff must manually verify that students have
                completed CCA choices to the year group's requirements, by
                literally printing out the spreadsheet to paper and reading
                through them.
        \item The school has to pay SchoolsBuddy an expensive subscription fee.
\end{itemize}

I am developing this project to replace this legacy system and to improve the
user experience for both the CCA department and individual students.

\section{Rationale for the proposed solution}

I have consulted with the school's IT department and confirmed that, with the
approval of appropriate faculty such as the Head of Co-curricular Activities,
the IT can provision a virtual machine on the school's LAN, running a suitable
server operating system such as Alpine Linux or Debian, to run the solution
that I develop. I am also capable of running the solution on my own hardware in
my dorm if necessary for beta and acceptance testing.

I am relatively experienced in developing low-latency network applications such
as IRC software, and I am comfortable reading specifications of network
protocols on various layers of the OSI model. I am somewhat familiar with
developing web applications in the context of our school's environment, and I
have previously developed a library for web services written in
\href{https://go.dev}{Go} to interface with our school's Microsoft Entra ID
system for authentication (previously known as Azure Active Directory).

The program does not need input data during the development process. During
production, all data is automatically retreived from Microsoft Entra ID and the
Microsoft Graph API via delegated access once a student has logged in via OAuth
2.0; in practice, this data includes the year group (grade level), name,
student number, and email address, all of which are publicly available to any
student via the Microsoft Entra ID portal.

There are no special security considerations other than various standard ones
present when working with web applications. Care must be taken not to leak
client secrets used in the OAuth 2.0 authorization code flow, although leakage
thereof is considered inconsequential as an authorization code would be
required anyway. Cookies must be protected against cross-origin request forgery
and should have \texttt{httponly} and \texttt{secure} flags. It should be made
impossible for a student to spoof another student's course choices, provided
that the victim's school login credentials haven't been already leaked.

\section{Success criteria}

The product shall present to students and administrators an accessible and
easy-to-use web interface for choosing and managing CCAs. It shall address each
of the issues of SchoolsBuddy as presented above.

\begin{itemize}
```

```
        \item When too many students attempt to choose one course at a time,
            their attempts are sequentially processed, and those that
            exceed the CCA's member limit are properly rejected.
        \item It should be possible to log in via Microsoft Entra ID.
        \item The web page must be lightweight. The login page shall be
            preferrably no more than 15 KiB. The course selection page
            shall be preferrably no more than 50 KiB.
            If a compressing content encoding such as \texttt{gzip} or
            \texttt{deflate} is used, the values above refer to the size of
            resources after decompression. If minification is used, the
            values above refer to the size of resources after minification.
            The experience should be fast and it should not take excessive
            resources on the host server.
        \item The course selection categories shall be relatively intuitive.
            Students may choose from a table of dropdown choices, where
            there is one dropdown per CCA slot; or they may choose from a
            set of tables, where one table represents all CCAs in one
            CCA slot.
        \item The selection system must enforce CCA hours requirements.
        \item The selection system must be able to take a CSV of CCAs with
            fields such as period, location, teacher, and member cap. It
            must then be able to populate its own CCAs table that it
            presents to the students with information in the CSV.
        \item The selection system must be able to export student choices as a
            CSV containing the following fields:
            \begin{itemize}
                \item Student name
                \item Numeric student ID
                \item Student year group (grade level)
                \item CCA name
                \item CCA period
            \end{itemize}
            These may then be used to trivially import choices to
            PowerSchool. It would be best if the PowerSchool API could be
            directly accessed to insert the courses, but the API is not
            publicly documented.
\end{itemize}

\end{document}
```

7.3. critb_design.tex

```
\input{header.texinc}

\title{Criterion B: Design}

\begin{document}
\maketitle

\section{Design methodologies}

\section{Draft design}
```

```
\section{Testing plan}

\end{document}
```

7.4. critc_development.tex

```
\input{header.texinc}

\title{Criterion C: Development}

\begin{document}
\maketitle

% Remember to demonstrate algorithmic thinking.

\section{Structure and justification}

\section{Algorithmic thinking}

\section{Development techniques}

\section{Existing tools}

The following tools, libraries, and other materials were used in the
development of this product.

\begin{itemize}
        \item The \href{https://go.dev}{Go programming language}'s
                \href{https://go.dev/ref/spec}{specification} was referred to,
                especially for documentation on how channel operations work;
                its GC toolchain (the most widely-used reference
                implementation) is used as the compiler during development and
                production; its standard library is used extensively in the
                program.

        \item \href{https://gobyexample.com}{Go by Example} was referred to for
                documentation on command-line flags and contexts.

        \item \href{https://developer.mozilla.org/en-US/}{MDN Web Docs} was
                used as my primary source of JavaScript documentation.

        \item \href{https://godocs.io}{A hosted fork of gddo} was used as my
                primary source of Go documentation, along with using the
                \texttt{go doc} command as part of the GC toolchain.

        \item \href{https://git.sr.ht/~emersion/go-scfg}{scfg}
                written by \href{https://emersion.fr}{Simon Ser} is used to
                parse configuration files.

        \item \href{https://github.com/MicahParks/keyfunc}{keyfunc}
                written by \href{https://micahparks.com/}{Micah Parks} is used
                to update the JSON Web Key Set to validate JSON Web Tokens for
                user authentication.

        \item \href{https://github.com/coder/websocket}{websocket} maintained
                by \href{https://coder.com/}{coder} is used for bi-directional
                communication.

        \item A minimal variant of the
                \href{https://www.rfc-editor.org/rfc/rfc1459#section-2.3}{RFC1459
                IRC message format} is used as the message format in
                client-server communication.

        \item \href{https://github.com/golang-jwt/jwt}{golang-jwt} is used to
                parse and validate JSON Web Tokens for user authentication.

        \item \href{https://github.com/google/uuid}{uuid} by Google is used to
                generate
                \href{https://www.rfc-editor.org/rfc/rfc9562.html}{UUIDs}
                during testing.

        \item \href{https://github.com/jackc/pgx}{pgx} by
                \href{https://jackchristensen.com/}{Jack Christensen} is used
                to establish a connection with the PostgreSQL database backend.

        \item \href{https://www.postgresql.org/}{PostgreSQL} is used as a
                database backend.

        \item \href{https://golangci-lint.run/}{golangci-lint} is used as a
                linter to detect programming errors.

        \item \href{https://neovim.io/}{neovim} is used as a text editor; its
                LSP client was used to connect to
                \href{https://pkg.go.dev/golang.org/x/tools/gopls}{gopls} for
                error detection and documentation provision while editing code.

\end{itemize}


\end{document}
```

7.5. critd_functionality.tex

```
\input{header.texinc}

\title{Criterion D: Functionality}

\begin{document}
\maketitle



\end{document}
```

7.6. crite_evaluation.tex

```
\input{header.texinc}

\title{Criterion E: Evaluation}

\begin{document}
\maketitle



\end{document}
```

7.7. agpl.texinc

```
\begin{center}
        {\LARGE\bfseries GNU Affero General Public License}

        Copyright \copyright\  2007 Free Software Foundation, Inc. \url{https://fsf.org/}

        \bigskip
        Everyone is permitted to copy and distribute verbatim copies of this\\
        license document, but changing it is not allowed.
\end{center}

\textbf{Preamble}

The GNU Affero General Public License is a free, copyleft license
for software and other kinds of works, specifically designed to ensure
cooperation with the community in the case of network server software.

The licenses for most software and other practical works are
designed to take away your freedom to share and change the works.  By
contrast, our General Public Licenses are intended to guarantee your
freedom to share and change all versions of a program--to make sure it
remains free software for all its users.

When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights
with two steps: (1) assert copyright on the software, and (2) offer
you this License which gives you legal permission to copy, distribute
and/or modify the software.

A secondary benefit of defending all users' freedom is that
improvements made in alternate versions of the program, if they
```

```latex
receive widespread use, become available for other developers to
incorporate.  Many developers of free software are heartened and
encouraged by the resulting cooperation.  However, in the case of
software used on network servers, this result may fail to come about.
The GNU General Public License permits making a modified version and
letting the public access it on a server without ever releasing its
source code to the public.

The GNU Affero General Public License is designed specifically to
ensure that, in such cases, the modified source code becomes available
to the community.  It requires the operator of a network server to
provide the source code of the modified version running there to the
users of that server.  Therefore, public use of a modified version, on
a publicly accessible server, gives the public access to the source
code of the modified version.

An older license, called the Affero General Public License and
published by Affero, was designed to accomplish similar goals.  This is
a different license, not a version of the Affero GPL, but Affero has
released a new version of the Affero GPL which permits relicensing under
this license.

The precise terms and conditions for copying, distribution and
modification follow.

\begin{center}
        {\Large\scshape Terms and Conditions}
\end{center}

\begin{enumerate}

\addtocounter{enumi}{-1}

\item Definitions.

``This License'' refers to version 3 of the GNU Affero General Public License.

``Copyright'' also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

``The Program'' refers to any copyrightable work licensed under this
License.  Each licensee is addressed as ``you''.  ``Licensees'' and
``recipients'' may be individuals or organizations.

To ``modify'' a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a ``modified version'' of the
earlier work or a work ``based on'' the earlier work.

A ``covered work'' means either the unmodified Program or a work based
on the Program.

To ``propagate'' a work means to do anything with it that, without
```

```latex
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

To ``convey'' a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays ``Appropriate Legal Notices''
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

\item Source Code.

The ``source code'' for a work means the preferred form of the work
for making modifications to it.  ``Object code'' means any non-source
form of a work.

A ``Standard Interface'' means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

The ``System Libraries'' of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
``Major Component'', in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

The ``Corresponding Source'' for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.
```

```
The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

The Corresponding Source for a work in source code form is that
same work.

\item Basic Permissions.

All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

\item Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

\item Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
```

```
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

\item Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:
  \begin{enumerate}
  \item The work must carry prominent notices stating that you modified
  it, and giving a relevant date.

  \item The work must carry prominent notices stating that it is
  released under this License and any conditions added under section
  7.  This requirement modifies the requirement in section 4 to
  ``keep intact all notices''.

  \item You must license the entire work, as a whole, under this
  License to anyone who comes into possession of a copy.  This
  License will therefore apply, along with any applicable section 7
  additional terms, to the whole of the work, and all its parts,
  regardless of how they are packaged.  This License gives no
  permission to license the work in any other way, but it does not
  invalidate such permission if you have separately received it.

  \item If the work has interactive user interfaces, each must display
  Appropriate Legal Notices; however, if the Program has interactive
  interfaces that do not display Appropriate Legal Notices, your
  work need not make them do so.
\end{enumerate}
A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
``aggregate'' if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

\item Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:
  \begin{enumerate}
  \item Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by the
  Corresponding Source fixed on a durable physical medium
  customarily used for software interchange.

  \item Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by a
  written offer, valid for at least three years and valid for as
  long as you offer spare parts or customer support for that product
  model, to give anyone who possesses the object code either (1) a
  copy of the Corresponding Source for all the software in the
  product that is covered by this License, on a durable physical
  medium customarily used for software interchange, for a price no
  more than your reasonable cost of physically performing this
  conveying of source, or (2) access to copy the
  Corresponding Source from a network server at no charge.

  \item Convey individual copies of the object code with a copy of the
  written offer to provide the Corresponding Source.  This
  alternative is allowed only occasionally and noncommercially, and
  only if you received the object code with such an offer, in accord
  with subsection 6b.

  \item Convey the object code by offering access from a designated
  place (gratis or for a charge), and offer equivalent access to the
  Corresponding Source in the same way through the same place at no
  further charge.  You need not require recipients to copy the
  Corresponding Source along with the object code.  If the place to
  copy the object code is a network server, the Corresponding Source
  may be on a different server (operated by you or a third party)
  that supports equivalent copying facilities, provided you maintain
  clear directions next to the object code saying where to find the
  Corresponding Source.  Regardless of what server hosts the
  Corresponding Source, you remain obligated to ensure that it is
  available for as long as needed to satisfy these requirements.

  \item Convey the object code using peer-to-peer transmission, provided
  you inform other peers where the object code and Corresponding
  Source of the work are being offered to the general public at no
  charge under subsection 6d.
  \end{enumerate}

A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

A ``User Product'' is either (1) a ``consumer product'', which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, ``normally used'' refers to a
typical or common use of that class of product, regardless of the status
```

```latex
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

``Installation Information'' for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

\item Additional Terms.

``Additional permissions'' are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:
  \begin{enumerate}
  \item Disclaiming warranty or limiting liability differently from the
  terms of sections 15 and 16 of this License; or

  \item Requiring preservation of specified reasonable legal notices or
  author attributions in that material or in the Appropriate Legal
  Notices displayed by works containing it; or

  \item Prohibiting misrepresentation of the origin of that material, or
  requiring that modified versions of such material be marked in
  reasonable ways as different from the original version; or

  \item Limiting the use for publicity purposes of names of licensors or
  authors of the material; or

  \item Declining to grant rights under trademark law for use of some
  trade names, trademarks, or service marks; or

  \item Requiring indemnification of licensors and authors of that
  material by anyone who conveys the material (or modified versions of
  it) with contractual assumptions of liability to the recipient, for
  any liability that these contractual assumptions directly impose on
  those licensors and authors.
  \end{enumerate}

All other non-permissive additional terms are considered ``further
restrictions'' within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

\item Termination.
```

```
You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

\item Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

\item Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

An ``entity transaction'' is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.
```

```
You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

\item Patents.

A ``contributor'' is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's ``contributor version''.

A contributor's ``essential patent claims'' are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, ``control'' includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

In the following three paragraphs, a ``patent license'' is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To ``grant'' such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  ``Knowingly relying'' means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
```

```latex
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

A patent license is ``discriminatory'' if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

\item No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

\item Remote Network Interaction; Use with the GNU General Public License.

Notwithstanding any other provision of this License, if you modify the
Program, your modified version must prominently offer all users interacting
with it remotely through a computer network (if your version supports such
interaction) an opportunity to receive the Corresponding Source of your
version by providing access to the Corresponding Source from a network
server at no charge, through some standard or customary means of
facilitating copying of software.  This Corresponding Source shall include
the Corresponding Source for any work covered by version 3 of the GNU
General Public License that is incorporated pursuant to the following
paragraph.

Notwithstanding any other provision of this License, you have permission to
link or combine any covered work with a work licensed under version 3 of
the GNU General Public License into a single combined work, and to convey
the resulting work.  The terms of this License will continue to apply to
the part which is the covered work, but the work with which it is combined
will remain governed by version 3 of the GNU General Public License.

\item Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of
the GNU Affero General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU Affero General
Public License ``or any later version'' applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU Affero General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU Affero General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

\item Disclaimer of Warranty.

\begin{sloppypar}
 THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
 APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE
 COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ``AS IS''
 WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,
 INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE
 RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.
 SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL
 NECESSARY SERVICING, REPAIR OR CORRECTION.
\end{sloppypar}

\item Limitation of Liability.

 IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
 WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES
 AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR
 DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
 DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM
 (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
 INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE
 OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH
 HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
 DAMAGES.

\item Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

\begin{center}
{\Large\scshape End of Terms and Conditions}

\bigskip
How to Apply These Terms to Your New Programs
\end{center}

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the ``copyright'' line and a pointer to where the full notice is found.

\begin{verbatim}
<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
\end{verbatim}

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer
network, you should also make sure that it provides a way for users to
get its source.  For example, if your program is a web application, its
```

```
interface could display a ``Source'' link that leads users to an archive
of the code.  There are many ways you could offer source, and different
solutions will be better for different programs; see section 13 for the
specific requirements.

You should also get your employer (if you work as a programmer) or
school, if any, to sign a ``copyright disclaimer'' for the program, if
necessary.  For more information on this, and how to apply and follow
the GNU AGPL, see \url{https://www.gnu.org/licenses/}.

\end{enumerate}
```

7.8. header.texinc

```
% This file is to be \input{}'ed into the preamble of other documents

\documentclass[numbers=endperiod, parskip=half-]{scrartcl}

\subject{IBDP Computer Science Internal Assessment}
\author{Runxi Yu}
\date{\today}

\usepackage{fontspec}
\setmainfont{TeX Gyre Termes}
\setsansfont{TeX Gyre Heros}
\setmonofont{Inconsolatazi4}
\usepackage{unicode-math}
\setmathfont{TeX Gyre Termes Math}
\usepackage{microtype}

\usepackage{xcolor}
\usepackage[colorlinks, allcolors={white!40!blue!50!black}]{hyperref}
```

8. Copyright license

GNU Affero General Public License

Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions.

   "This License" refers to version 3 of the GNU Affero General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

   The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

   The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

   The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

   All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

   c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

   d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

   A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

   You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this

License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

    A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

    A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

    Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

    In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

    If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

    If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Remote Network Interaction; Use with the GNU General Public License.

    Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

    THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License
```

```
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a “Source” link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see https://www.gnu.org/licenses/.