

程序设计实习大作业报告——选课网++

——OES 队（沈韬、欧阳霏、卢恒毅）

简介

本项目为一套功能完备的综合选课系统,集选课管理、课程测评与智能选课辅助于一体,面向高校师生用户,支持多角色登录、多学期状态管理与全过程数据持久化,致力于打造灵活、高效、智能的选课平台。系统主要包含三个核心模块:选课模块、课程测评模块与智能选课辅助系统模块,辅以教务管理界面与学生主界面构成整体框架。

在选课模块中,系统根据用户角色提供差异化功能。学生用户可浏览课程信息,进行投点选课、课程收藏及查询历史选课记录;教务用户则具备课程增删改查、批量导入、手工选课、抽签操作与名单导出等权限。课程数据以统一结构组织,涵盖编号、名称、时间、授课教师、限选人数、中文/英文简介等字段,通过 `CourseInfo` 类封装为对象,在模块间传递时通过 JSON 实现序列化与反序列化。学生选课过程中的投点情况与中签结果存储于 `UserInfo` 实例中,并通过投点值构建抽签池,采用 `QRandomGenerator` 完成抽签,配合 `QProgressBar` 动态显示抽签进度。

课程测评模块允许学生对已选课程进行评论与量化评分,评价维度包括听感、作业负担、考核难度与优先级,每条评论可被点赞/点踩以标记其参考价值,最终影响该评价在列表中的显示顺序与该课程在推荐过程中的评分权重。测评数据结构定义在 `coursecomment.h` 中,通过 `commentLoader` 进行读写管理,并与历史选课模块联动,仅允许对已选过课程进行评价。

智能选课辅助系统是本项目的创新亮点之一。该模块在用户输入个性化偏好(如偏好评分高/任务少的课程、避免时间冲突、避免重复修读)后,结合选课时间、已修课程与抽签策略,采用动态规划算法生成多组推荐排课方案与投点分配建议。该过程综合考虑课程评分、已选人数与最大容量等因素,提升推荐合理性与可中签率。此外,系统使用 `sigmoid` 函数估算投点中签概率,实时反馈用户投点行为带来的期望收益。

系统支持多角色登录,登录界面区分学生与教务身份验证逻辑,学生通过密码认证,教务用户除固定密码外还需输入系统动态生成的四位令牌进行二次校验,提升管理安全性。用户注册功能采用模态对话框实现,注册信息写入本地 JSON 文件,并在系统加载时同步更新。主界面设计强调模块化、清晰性与实时反馈,学生端通过 `QStackedWidget` 管理主页、选课、收藏夹、课表与历史模块,界面左侧为模块导航栏,底部显示当前学期、选课学期与抽签状态,增强系统可解释性。

在 UI 实现方面,系统大量使用 Qt 的 `QTableWidget`、`QLabel`、`QPushButton` 等控件,配合自定义样式表提升用户体验。课表模块中课程块绘制基于背景图+图层结构,时间解析算法支持单双周与节次识别,并通过课程索引值生成哈希色调,确保同课程颜色一致而不同课程区分明显。收藏机制与选课、课程详情等功能间通过信号槽机制联动,确保操作状

态的即时反馈。历史选课模块可展示所有学期选课记录，并为课程测评提供入口与上下文信息。

教务界面采用分模块管理，涵盖课程管理、抽签、手工选课、令牌刷新等功能。所有操作均基于当前系统学期状态进行权限控制，防止误操作或越权访问。选课数据与系统状态通过 `teacher_state.json` 与 `courses.json` 实现本地持久化存储，并支持随时间演进灵活切换“当前学期”“选课学期”“待选课学期”，为系统运行提供时序管理支持。

类的实现

登录界面 (LoginWindow)

主要功能：

提供选课系统的用户登录入口，支持学生与教务两种角色的身份验证。学生用户可通过用户名和密码登录系统；教务用户需输入固定账号密码并额外验证动态令牌，以提升安全性。此外，界面中还集成了注册入口，允许新学生用户注册账户信息。

实现细节：

采用 `QLineEdit`、`QComboBox` 和 `QPushButton` 等控件搭建界面，并结合样式表美化交互体验。系统通过 `roleBox` 控制登录角色的选择，动态切换令牌输入框的可见性。当登录身份为学生时，系统从 `users.json` 文件中读取用户数据，解析为 `QJsonObject`，并比对用户名与密码字段，一致则构造 `UserInfo` 对象并发出登录成功信号；若验证失败则显示错误提示。教务用户登录采用固定的用户名密码 (`admin/adminpass`) 配合 `TeacherInfo` 类中的动态令牌校验。注册功能由 `RegisterWindow` 触发，若成功返回，则将新用户数据写入 `users.json` 文件，采用读取-合并-重写的方式更新本地数据，防止覆盖其他用户信息。登录界面还提供了“显示密码”按钮，实现密码明文/密文切换，提高使用便利性。

注册界面 (RegisterWindow)

主要功能：

为学生用户提供账户注册功能，支持填写用户名、密码、真实姓名、学号、年级与院系等基本信息，并完成信息校验与身份确认。在确保用户填写内容完整、学号唯一、密码一致的前提下，生成 `UserInfo` 实例并提交给主程序，实现新用户的添加与持久化。

实现细节：

基于 `QDialog` 构建，采用垂直布局管理多个输入框组件，包括 `QLineEdit` 用于用户名、密码、确认密码、真实姓名和学号输入，`QComboBox` 用于年级与院系选择，并通过 `QCompleter` 实现对院系名称的智能补全。注册逻辑中，系统首先检查所有字段是否填写完整，并验证两次密码输入是否一致。随后，调用 `isIndexConflict()` 方法判断输入的学号是否已存在于 `users.json` 文件中，若冲突则阻止注册。信息无误后，通过弹窗提示用户确认身份信息（院系、年级与姓名），以防误操作；若用户点击“确认”，则将注册信息打包为 `UserInfo` 对象，并通过 `accept()` 将其提交给主窗口，由后者负责保存数据。

学生信息管理 (UserInfo)

主要功能:

用于管理学生用户的基本信息与选课相关数据, 包括用户名、密码、年级、学院、身份索引、真实姓名等基本属性, 以及当前课程、收藏课程、历年课程记录、课程投点与抽签结果等动态信息。支持学生选课、投点抽签、历史归档、收藏管理等核心操作。

实现细节:

通过构造函数初始化用户基本信息, 并加载默认课程填充当前课程列表, 同时为每门课程设置初始投点值。内部使用 QMap 管理课程投点与抽签结果, 提供如 getPointForCourse() 和 setPointForCourse() 等方法实现数据读写。支持 toJson() 和 fromJson() 进行完整序列化与反序列化, 涵盖当前课程、收藏夹和历史课程等内容, 确保用户数据持久保存与恢复。通过 archiveCurrentCourses() 可归档当前学期课程至历史记录, 便于多学期展示与管理。

学生主界面 (MainWindow)

主要功能:

为学生用户提供一个功能完备、模块化集成的选课操作主界面, 支持主页浏览、课程选择、智能选课、收藏管理、课表查看、历史记录等多种功能模块切换, 并实时展示当前学期与选课状态。

实现细节:

使用 QStackedWidget 实现模块页面的切换, 配合左侧 QListWidget 构建导航侧边栏, 模块包括主页、选课页面、智能选课系统、收藏夹、选课列表(课表与课程信息切换)、历史选课等。构造函数中首先加载课程与评论数据, 并生成对应的模块页面, 包括 HomePage、CourseSelection、ClassQuestionnaire、CourseListWidget、TimetablePage 和 HistoryPage。其中课程收藏更新通过信号槽机制联动, 确保收藏夹页面始终显示最新状态。

根据 TeacherInfo 中记录的选课学期信息判断是否为有效选课时间段, 并在非选课期间禁用“选课”、“智能选课”、“收藏夹”等按钮, 通过按钮状态与 Tooltip 文本引导用户行为。课表模块与课程列表模块实现双向联动, 在切换时同步刷新显示当前选课情况, 调用 refreshCourses() 与 setCourses() 方法进行数据更新。侧边栏底部附有三个标签, 分别展示当前学期、当前选课学期与抽签状态, 增强系统时序信息的可见性。

首页界面 (HomePage)

主要功能:

为学生用户提供系统欢迎界面和使用指引, 展示用户基本信息与各模块简介, 帮助用户快速了解平台的核心功能和操作流程。

实现细节:

通过 QVBoxLayout 构建页面主布局, 顶部展示欢迎语与用户个人信息 (包括用户名、真实姓名、学号、年级与学院), 内容区域使用 QScrollArea 滚动显示平台各模块功能简介, 包括选课、智能推荐、收藏、课程列表和历史记录等模块的功能说明, 采用 HTML 富文本支持图标、加粗与段落排版, 使说明更具可读性。底部右侧添加**平台 LOGO**, 并在 paintEvent 中重绘背景图, 增强界面视觉美感。整体设计强调信息完整性与美观性的统一。

课程表界面 (Timetable&TimetablePage)

主要功能:

用于以课表视图展示学生当前所选课程的时间安排, 帮助用户直观了解每日课程分布。页面提供 “⇌” 按钮实现与课程列表页面的切换, 并根据是否完成抽签决定是否筛选显示中签课程。

实现细节:

TimetablePage 作为界面容器, 内部嵌套一个自定义组件 CustomTimetableWidget, 并提供一个切换按钮 switchButton, 用于发出信号跳转至课程列表页面。setCourses 方法用于加载用户的选课数据, 若当前为抽签后阶段, 则根据 UserInfo 中的中签记录进行过滤, 仅展示中签课程。该逻辑依赖 TeacherInfo 判断系统抽签状态, 通过布尔变量判断是否调用筛选逻辑。

课表绘制组件中, 首先设定背景为一张空白课表图 (SVG), 然后通过 overlayLayer 图层在其上绘制课程块。对传入的每门课程解析其时间字符串 (如 “星期三(第 3 节-第 5 节)(单)”), 提取出星期几、节次范围、单双周限制等信息, 并计算对应位置。每门课程用 QLabel 展示, 课程块内显示课程名称、周次与教师姓名, 并为不同课程赋予基于哈希的色调生成颜色, 保证同课程颜色一致、不同课程有区分。

选课模块(CourseSelection)

主要功能:

点击上方 “课程检索” 按钮, 在搜索框输入关键字可进行课程搜索, 点击每条搜索结果右侧三个按钮分别可以进行查看详情、添加到收藏夹、选课操作, 鼠标悬浮在评分栏的图标上可便捷查看评测相关指标。在查看详情页面可以查看课程基本信息、中英文介绍以及课程测评信息。

点击上方 “已选课程” 按钮, 可查看当前选课情况, 点击每一条目右侧的退课按钮可移除有关课程

实现细节:

相关文件: courseSelection, courseEnrolledPage, courseSearchPage, courseDetailPage。

重要类: CourseSelection, CourseSearchPage, CourseEnrolledPage,

CourseDetailPage 用于实现相关界面以及同数据交互。

历史选课(HistoryPage)

主要功能：

显示历史选课记录。进入此模块可进入历史选课学期查看选课历史记录；并作为测评系统的入口。历史记录中的课程栏目右侧具有撰写测评按钮，学生可从听感、作业、考核、优先级四个层面进行课程评分，并支持留言。

实现细节：

- 测评功能

相关文件：commentLoader, coursecomment, 定义了 comment 的数据结构，并支持从 json 文件的写入导出。

课程列表界面 (CourseListWidget)

主要功能：

用于展示当前选课课程、历史课程或收藏课程，支持投点操作、课程评价、选中情况查看与页面视图切换，帮助学生全面管理当前与历史课程数据。

实现细节：

根据构造参数 type 区分模块用途（0 表示选课列表，1 表示历史列表，2 表示收藏夹），设置不同标题与列结构。主布局采用 QVBoxLayout 管理表格与顶部操作栏，若为选课列表则额外展示投点情况与中签概率估算。表格通过 QTableWidgetItem 实现，支持课程信息展示、列宽设置与行高自适应。

选课列表中，末列支持动态投点并实时更新中签概率提示，点数超限时弹出警告框；若已抽签，则显示“已选上/未选上”。历史列表中每门课可进行课程测评，评价内容通过对话框采集并写入 JSON 文件中，支持内容与四项指标打分。

投点操作通过 QSpinBox 与“投点”按钮联动实现，点击后自动更新用户状态及底部提示栏（已选学分、剩余点数、预计未中签学分），抽签概率估算由 estimateSelectionProbability() 提供，具体通过 sigmoid 函数拟合。模块底部通过按钮支持切换至课表视图，适配不同操作场景。

智能选课系统 (ClassQuestionnaire)

主要功能：

根据提供的需要的学分数量、各类课程数量以及当前已选课程，结合对“听感”“给分”等六个不同维度的需求权重，在保证课程时间不会发生冲突的前提下，生成一份尽可能符合需求的选课方案。

实现细节：

定义 checkTimeString 函数，通过课程的时间字符串判断课程之间是否存在时间冲突；定义 CourseWithLoss 结构体存储课程及其不同维度的“不符合度”，并通过对课程类型、学分等的特判清洗掉不符合需求的课程。

在 resultPage 函数中使用动态规划算法找出符合要求的选课方案，同时为了避免数组空间过大导致程序无法运行，利用滚动数组优化空间复杂度，保证程序的正常运行。

课程信息存储 (CourseInfo)

主要功能：

提供课程信息结构定义及与 JSON 数据的双向转换能力，支持从 JSON 对象解析为 CourseInfo 实例、将课程对象序列化回 JSON，确保课程数据在各模块之间的读取、保存与交换过程中格式统一、结构完整。

实现细节：

定义 CourseInfo 结构体字段，涵盖课程编号、名称、班号、类型、学分、时间信息、授课教师列表、中英文简介、最大/当前人数等。函数 parseCourseFromJson() 从 QJsonObject 中提取各字段构造课程实例，支持 time 和 teacher 字段以 QJsonArray 格式解析为 QStringList 存储；CourseInfo::toJson() 成员函数用于将课程结构反序列化为 QJsonObject，自动组织为统一键值结构。

教务信息存储 (TeacherInfo)

主要功能：

存储当前选课学期 (enrollmentTerm)、当前上课学期 (CurrentTerm)、下个选课学期 (UpcomingTerm) 等信息实现学期状态的时序化管理，从而为选课阶段的切换及各项功能的启用与控制提供支持。

实现细节：

通过维护学期信息、教务令牌、抽签状态等关键变量，并借助 teacher_state.json 文件实现对这些状态的实时读取、持久存储与动态更新，确保系统在不同阶段下功能的正确切换与执行。

教务窗口 (TeacherWindow)

主要功能：

维护**编辑课程列表**（可选择手动添加课程信息或使用 csv 文件导入、编辑课程信息、删除课程等操作维护下个选课学期的课程列表，仅能在选课开始前进行）、**手工选课**（允许教务在选课期间，通过输入学生姓名与课程编号，手动为学生添加课程。）、**修改当前学期**（支持推进学期、开始选课、停止选课等关键操作，确保系统时序管理的准确性）、**抽签**（在选课结束后，根据学生投点数进行抽签分配课程，抽签结果自动写入学生信息中）、**导出**

名单（仅在抽签结束后进行，可一键将每门课的信息导出至 curriculum 文件夹下的各 csv 文件中，打开可查看选中该门课的学生学号等信息）、**刷新令牌**（随机生成四位数字令牌，用于教务人员身份验证及登录管理）六个教务功能，共同构建起完整的教务操作体系，为选课流程提供全面支持与控制。

实现细节：

整体采用 QStackedWidget 管理多个页面，主界面通过图标按钮进入不同子功能模块，每个功能页面根据操作权限（由当前/选课/下学期状态决定）动态启用或禁用。

编辑课程列表功能中，支持从 csv 文件批量导入课程信息，自动解析字段填充 CourseInfo 实例，并存储为 json，也支持手动添加课程。支持对已有课程进行选择、编辑与批量删除，界面联动更新表格状态，并自动重新编号索引号。所有课程数据统一存储在 courses.json 中，读写通过 QJsonDocument 实现持久化。

手工选课功能中，教务通过输入学生学号与课程索引号进行选课。实现学生数据的查找与修改，并追加课程至用户的 currentCourses 字段。

修改当前学期功能中，实现三类操作：推进当前学期、启动下一轮选课、结束当前选课。所有操作通过 TeacherInfo 类封装逻辑并更新 teacher_state.json 状态文件。操作后界面同步更新当前学期、选课学期和待选课学期状态标签与按钮启用状态。

抽签功能中，针对人数超限课程，从已报名学生中按投点数构建抽签池，通过 QRandomGenerator 随机抽签。采用 QProgressBar 显示抽签进度，抽签结果写入每位学生的 courseLottery 字段。完成后标记“已抽签”状态，刷新按钮和界面提示。

导出名单功能中，读取所有课程与学生数据，为每门课程生成 .csv 名单文件，保存至 curriculum 文件夹下，记录包含选课成功学生的学号、姓名、年级与院系，并使用 QProgressBar 实时展示导出进度。

刷新令牌功能中，一键生成随机四位数字作为教务登录验证码，便于校验管理，并通过 TeacherInfo 类接口保存至状态文件，更新后弹窗提示新令牌值。

所有操作按钮基于学期状态判断权限是否开放，避免越权或错误时序操作。无权限操作将按钮设为不可用并提示“当前不可用”，避免误点。

小组分工

沈韬（队长）：教务系统及其他辅助页面设计、视频配音

欧阳霁：选课页面

卢恒毅：智能选课系统

创新之处

1.利用哈希值为课程自动生成颜色，确保视觉一致性与区分度

在课表绘制中，对每门课程的课程编号或名称取哈希值，映射为固定色调，从而使同一

门课颜色恒定、不同课程颜色有区分，有效提升课表视觉识别效率。

2. 抽签功能支持“投点+中签概率估算”结合

选课投点操作中，系统通过用户设置的点数和课程热度，调用基于 sigmoid 函数拟合的 `estimateSelectionProbability()` 方法，动态估算中签概率并实时提示，为学生提供理性策略引导，降低试错成本。

3. 利用 QCompleter 实现院系名智能补全功能

在注册界面，针对院系输入字段配合 `QComboBox + QCompleter` 使用，根据用户输入前缀自动匹配提示已有院系名称，提高注册效率并减少拼写错误带来的院系信息异常。

4. 教务操作权限基于状态标签自动启用/禁用

如“编辑课程”“抽签”等按钮，仅在满足状态（如非选课中、未抽签）下启用，通过 `TeacherInfo` 中读取状态变量动态控制按钮启用状态与 `ToolTip` 提示，避免操作越权。

5. 高可靠性 JSON 数据存储与读取机制

所有用户与课程数据均使用 json 文件格式存储，配合 `fromJson()/toJson()` 接口封装，支持结构化持久化存储；写入操作采用“读取→合并→覆盖”方式防止误写或数据丢失。

6. 利用动态规划算法及滚动数组优化进行智能选课

智能选课模块中的选择课程使用了动态规划算法，具体实现上根据课程的评分和输入的六个方面的需求权重，计算出每门课程的“不符合度”，并使用动态规划算法求得在一定的学分数和课程数下“不符合度”取得最小值时的选课方案。同时使用滚动数组优化空间复杂度保证程序的正常运行。

总结与反思

本项目围绕“选课系统”的核心需求，搭建了一套功能较为完善的学生选课平台，涵盖了选课操作、课程测评、智能推荐等多个模块，基本实现了从课程管理到用户使用体验的完整闭环。我们在开发过程中不仅实现了常规的登录注册、课程查询与筛选、历史记录管理，还引入了诸如“投点抽签”“课程测评打分”“选课智能辅助”等相对进阶的功能，增强了系统的实用性与交互性。

尽管本项目已经完成了核心功能的开发，但在未来的完善与扩展上，我们还有许多可以继续努力的方向：

1. 引入数据库支持，提升数据管理效率

当前系统基于 JSON 文件进行数据存储，适用于小规模部署。后续可考虑引入 SQL 数据库系统，进一步规范数据结构、提升查询性能，并支持更复杂的数据分析。

2. 完善智能选课推荐系统

智能选课功能目前仍处于初级阶段，主要以硬约束与简单权重为主。未来可以尝试引入推荐算法，如协同过滤、基于测评数据的偏好建模等，让系统根据用户历史选课、课程评分与个性偏好，自动生成更合理、更贴合需求的推荐方案。

3. 优化界面交互体验

部分页面的布局、信息展示形式仍有改进空间。后续可进一步提升界面美观性与操作便利性，比如通过动画反馈、主题自定义、课程颜色智能调整等增强用户体验。

4.提升权限与时序控制的灵活性

当前选课权限控制逻辑主要集中在教务端状态的管理上，较为固定。将来可以引入更灵活的权限配置机制，例如允许设置多次选课时间窗口，实现更精细的管理策略。

5.增强课程测评系统与数据挖掘能力

若能进一步打通用户评价与课程安排之间的桥梁，如对低评分课程发出调整提醒，或对高评分课程自动推荐给新生，将为课程管理与学生决策提供更多数据支持。