DesignDocument_final.md

Project Design Document

Your Project Title

Prepared by:

- Sarah Listzwan , Worcester Polytechnic Institute
- Zachary Serocki , Worcester Polytechnic Institute
- Dylan Schmit , Worcester Polytechnic Institute
- Nikhil Gangaram , Worcester Polytechnic Institute

Course : CS 3733 - Software Engineering

Instructor: Sakire Arslan Ay

Table of Contents

- 1. Introduction
- 2. Software Design
  - 2.1 Database Model
  - 2.2 Modules and Interfaces
  - 2.2.1 Overview
  - 2.2.2 Interfaces
  - 2.3 User Interface Design
- 3. References
- Appendix: Grading Rubric

Document Revision History

Name	Date	Changes	Version
Revision 1	2025-2-12	Initial draft	1.0
Revision 2	2025-2-19	Revision	2.0

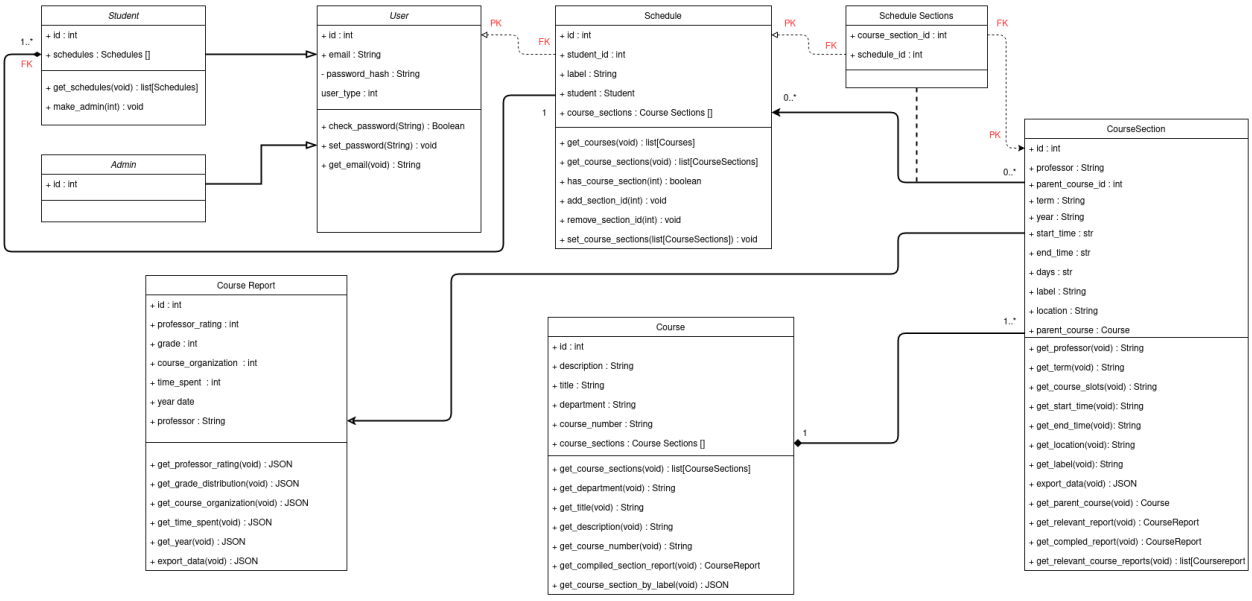
1. Introduction

We aim to introduce the design of the application, Personal Integrated Planner for WPI (PIPW). This application seeks to provide a class planning service for the Worcester Polytechnic Institute Community. Students can use this resource to plan their classes, and view the past relevant course reports in the same location. The design of this application is outlined in this document.

2. Software Design

2.1 Database Model

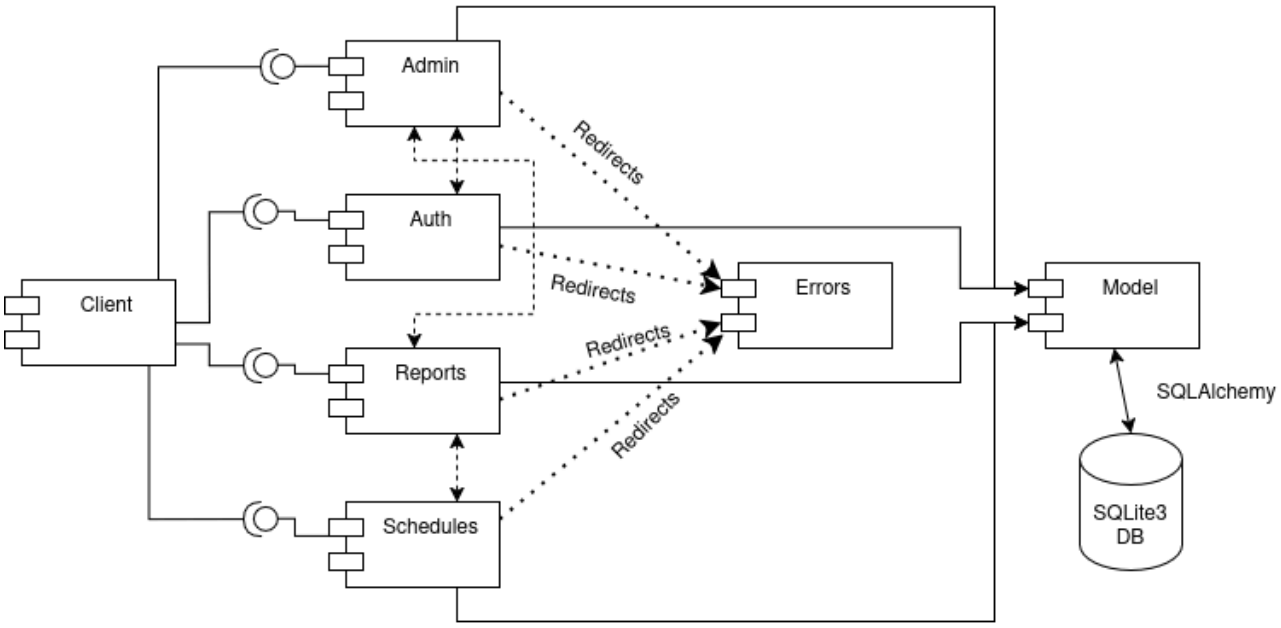
Name	Description
User	is a table of generic users
Student	inherits from user accesses scheduling pages
Admin	inherits from user accesses admin pages
Schedule	allows users to save the schedules that they have built
Course	Generic head that contains general course info
Course Section	Contains information about a specific section of a course (lab, lecture, or discussion), in a specific term
Course Reports	holds the previous course reports it isn't tied to a course section because the course sections might be removed



The user model stores the users information, the Student and Admin tables inherit from the User model and allow the program to determine redirect users to the right place. Schedules hold a collection of course sections and other relevant information data. Course sections is a collection of all the relevant data and is similar in nature to the WPI Lecture, Lab and Discussion sections. Course sections make up courses which is a overarching head that hold the Identical data of the course sections this minimizes the data footprint. Course reports share information from a course section and hold the student opinions on the course.

2.2 Modules and Interfaces

2.2.1 Overview



The admin module handles the admin pages. It allows admins to view all courses and edit courses. It allows the admin to search all courses and filter based on department, professor, or term. It also allows admins to edit courses.

The reports module holds the past course reports. The past course reports are then displayed visually for the user. The course reports are also sorted and filtered based on relevance.

The schedules module handles the user editing and creating course schedules. The user can view their current schedule in an easy-to-use calendar view, search for new course sections, add/remove course sections from their schedule, and clear their schedule.

The auth module handles users registration and login. It allows new user to register. It also allows login with both email and Single Sign On (SSO).

The errors module handles 404 and 500 errors to display a page to the user. The user can then navigate back to a known page.

2.2.2 Interfaces

2.2.2.1 Schedules Routes

	Methods	URL Path	Description
1.	GET	/schedules/view/	view schedule page
2.	GET	/schedules/<id>/data	get front-end schedule data for schedule id

	Methods	URL Path	Description
3.	GET	/schedules/search_courses/<query>	get course sections that match the query
4.	POST	/schedules/<id1>/add/<id2>	add course section with id2 to schedule with id1
5.	POST	/schedules/<id1>/remove/<id2>	remove course section with id2 from schedule with id1
6.	POST	/schedules/<id>/clear	clear schedule with id
7.	POST	/schedules/<id>/optimize	optimize schedule with id

2.2.2.2 Reports Routes

	Methods	URL Path	Description
1.	GET	/reports/<section_id>/view	show all relevant course reports for section with id
2.	GET	/reports/<id>/data	get data for report with id

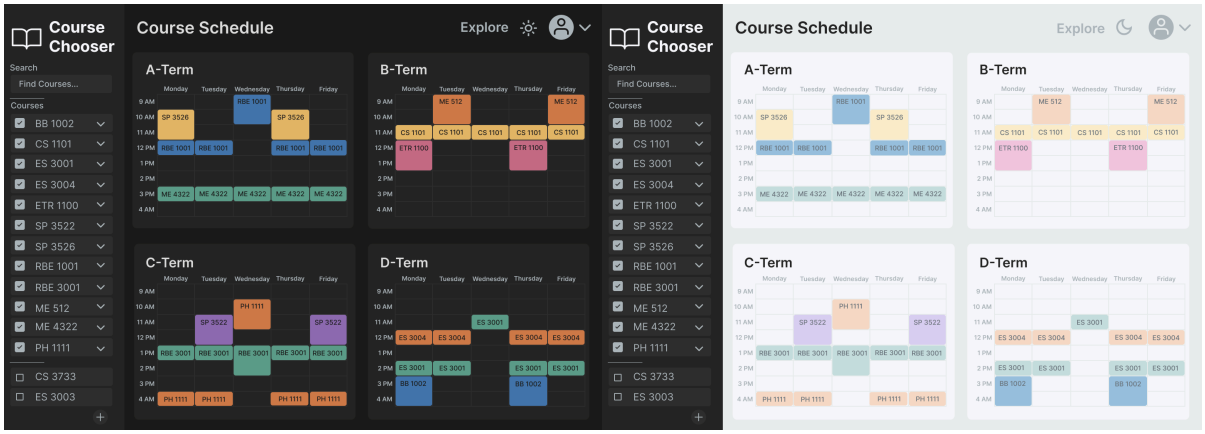
2.2.2.3 Auth Routes

	Methods	URL Path	Description
1.	GET, POST	/student/register	register as a student
2.	GET, POST	/user/login	log in to the app
3.	GET	/user/logout	log out of the app

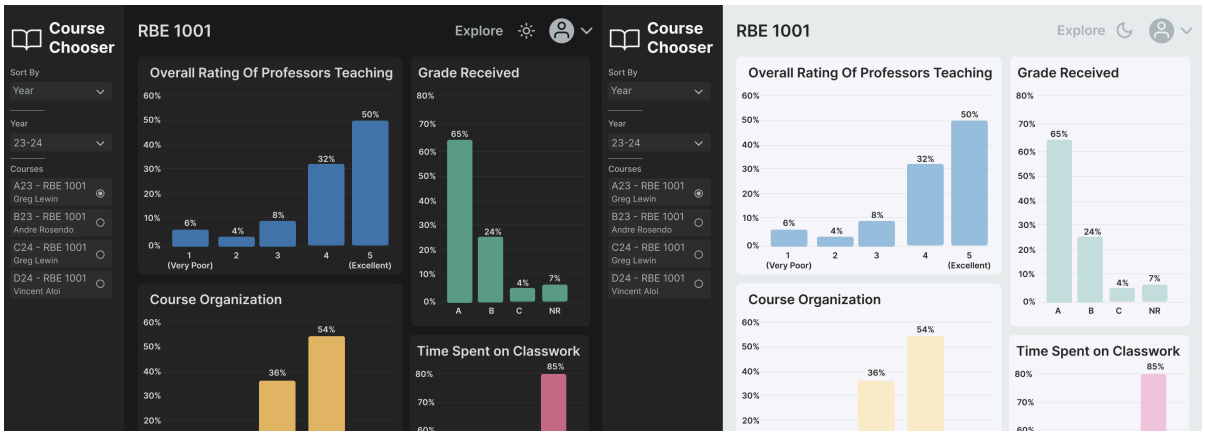
2.2.2.4 Admin Routes

	Methods	URL Path	Description
1.	POST	/admin/sync	sync course data from workday database
2.	GET	/admin/courses	get all course information for the admin view
3.	POST	/admin/courses/<course_id>/edit	edit a course and/or it's course listings

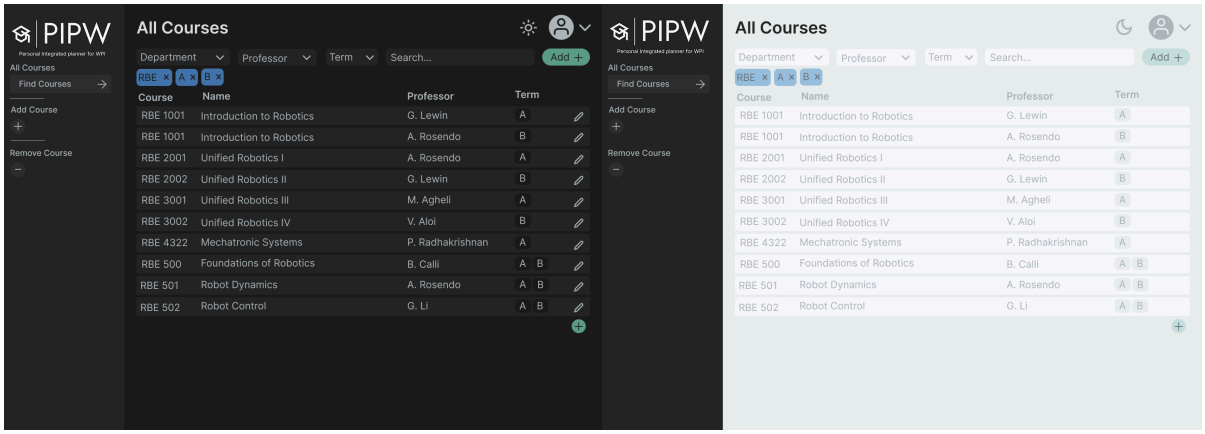
2.3 User Interface Design



Students can add classes and to their schedules. They can also choose sections to add. These classes are then displayed visually on their calendar.



Students can click on course sections on their schedule. This allows them to view the previous course reports relevant to that course (ie. same professor, same class, etc.)



Administrators can view all courses. They can filter by department, professor, or term.

3. References

Stanford OnCourse

- Developers: William Huang, Niall Kehoe, and Aly Sultan

WPI Planner

- Developers: Henrique Polido, Ryan Anthony, Douglas Lally
- Source code: https://github.com/Nican/wpischeduler
- Modified to work with Workday in 2021 by Jordyn McKeen and Nick Markou
- Source code: https://github.com/Jmckeen8/wpiplanner