Software Requirements Specification

Degree Admin

1.0.2

Mohamed Elkharraz

Jonathan Wasky

Mark Harshbarger

Brandon Whittaker

11/17/2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Jonathan Wasky	12/10/25	Filling out remaining fields in the document.	1.0.0
Jonathan Wasky	12/11/25	Expanding upon a few fields to better document the content and features of the Degree Admin project	1.0.1
Mark Harshbarger	12/11/25	Added Other Nonfunctional Requirements	1.0.2

1. Introduction

Purpose

The purpose of this project is to facilitate the tracking of students' college semesters and allow the advisors to approve or deny the students' schedule. Students are able to make schedules for different selections of Major(s) and/or Minor(s). The student can select the max amount of credit hours they want to take per semester, and a schedule will be generated for them. The student can also add a semester for a co-op. This allows flexibility for different students' needs. Faculty have to approve the students' schedules, which ensures that the student is on the right path to success. The system includes administrators who are able to modify information regarding other accounts.

Document Conventions

Degree Admin was originally created prior to the writing of this document. All requirements outlined here are already reflected within the current implementation of the software. As the software continues to evolve, any new features or updates should be documented here to ensure the requirements remain accurate and complete.

References

The team's official website and development repository can be located at the following links:

- Team website: https://markharshbarger.github.io
- Repository: https://github.com/Wamski/CEG-4110

2. Overall Description

Product Perspective

Degree Admin is software built to replace an existing University's scheduling system and portal. It is designed to streamline and modernize student/faculty workflows when scheduling their degrees. Degree Admin can be adopted and customized by institutions around the world.

Product Features

The standard version of Degree Admin allows for:

- Student Accounts
    - Select the Major(s) and/or Minor(s) they would like to take.
    - (*Optional*) Select the semester they would like to take a Co-Op.
    - Select the number of semester hours they would like to take per semester.
    - Generate a schedule.
    - View their schedule on a semester basis.
    - View course details.
        - Number of credits
        - The level of the course
        - A Description of the course
- Faculty Accounts
    - Can view a notification center that lets them know if a student assigned to them has a pending schedule.
    - Can enter a student ID to view that student’s information.
        - Total Credit Hours
        - Total Semesters
        - Full Schedule (Pending or Approved)
    - Can approve or deny pending schedules.
- Admin Accounts
    - Select a user account and edit the following:
        - Usernames
        - Passwords
        - Advisor/Faculty assigned to the account.
        - Deactivate/Reactivate the account.
    - Create a new user with the following:
        - Username
        - Password
        - Account level:
            - 1: Student
            - 2: Faculty
            - 3: Administrator

User Classes and Characteristics

The users for this project are as follows: students, faculty, and administrators.

Students:

Students can make a schedule for themselves by stating the max number of credits they would like to have per semester, along with their major(s) and minor(s). The unique schedule will then be generated. Students can also request overrides for their classes and request co-ops. If the student does take a co-op, then the schedule would also change accordingly. Students should cover all types of users with varying computer skills, there should not be a significant learning curve to interact with the student controls. Students have the lowest account/control level of 1.

Faculty:

Faculty are able to accept or reject students' requests for overrides and co-ops, while also being able to see the students' schedule that was generated. Faculty, like students should cover all types of users with varying computer skills, the interface should be easy to use. Faculty have a moderate amount of control with an account/control level of 2.

Administrators:

Admins are able to create, delete, and modify accounts. Admins are also responsible for assigning faculty to advise students. Lastly, the system locks out users after 3 failed password attempts for enhanced security. Admins are then able to unlock user's accounts and/or change the password in the event of a malicious actor. Administrators should have a good knowledge of how the account levels work, with 1 being the lowest (students) and 3 being the highest (admins).

Operating Environment

Degree Admin has been designed and tested with the following operating systems in mind:

- Microsoft Windows 10/11
- Mac OS (14.8+)
- Linux distributions
    - Linux Mint

Degree Admin also requires the following:

- PostgreSQL
- Minimum of 100 MB of storage

Design and Implementation Constraints

Degree Admin is open source and can run on most platforms. The frontend was written in ReactJS with Vite, while the backend uses python in conjunction with PostgreSQL. To be able to run Degree Admin, there must be a device serving the following: PostgreSQL for the database, FastAPI for the api endpoints, and React for the frontend. All three of these services must be running simultaneously for Degree Admin to function as intended.

Assumptions and Dependencies

- Working database using PostgreSQL
- Frontend service is running.
- Backend service with Fast API is running.

3. System Features

3.1 System Feature 1 – Login Page

The Login Page is a JSX component in the Degree Admin frontend. It uses a form submitting a username and password to request an account token and type.

3.1.1 Description and Priority

The login page is the primary landing page that users first see when accessing Degree Admin. Depending on a user's account type, received by signing in, an associated dashboard will be displayed. This feature is a high priorty for the Degree Admin project.

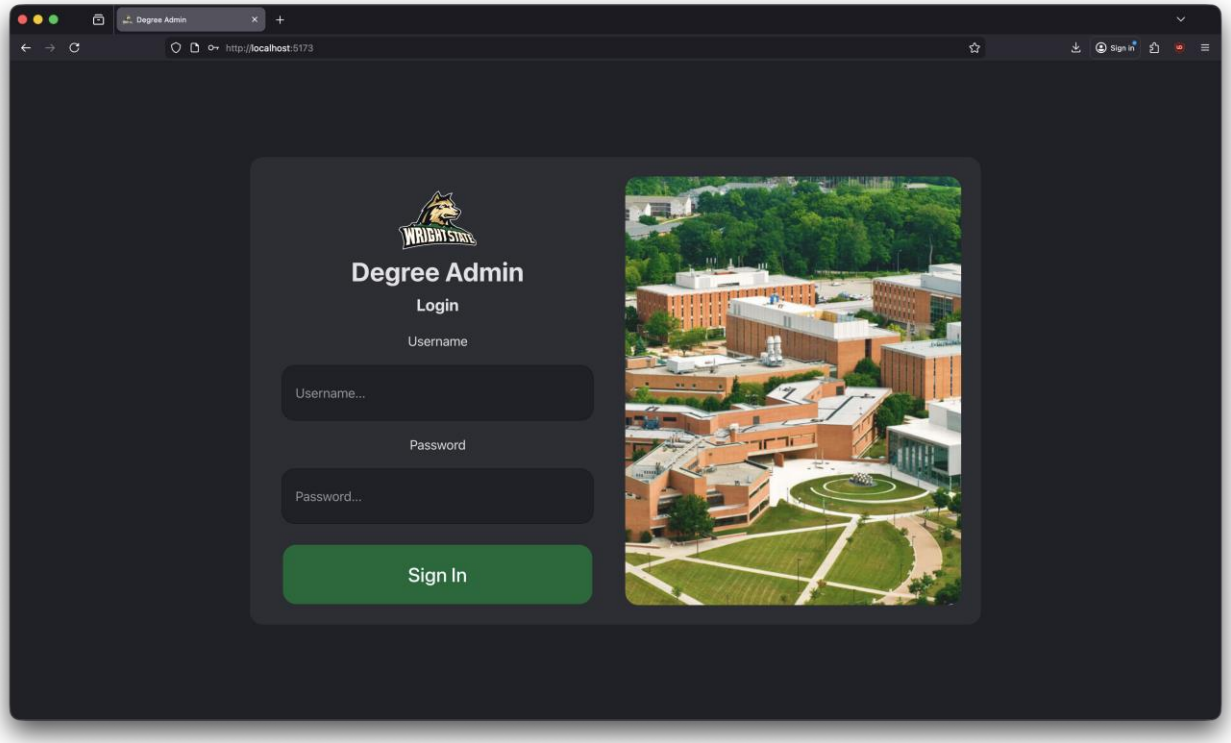


Figure 1 - A screenshot of the login page.

3.1.2 Stimulus/Response Sequences

The user has to enter a username and password in the input fields and then click on the "Sign In" button to access their content. When a user selects an input field, it is outlined in the primary green color to show that it is active. The sign in button changes to the secondary color of gold when hovered over to signal to the user that it can be pressed.

3.1.3 Functional Requirements

REQ-1: The user must enter both a valid username and password to be granted access into their respective dashboard.

REQ-2: The user can only attempt to successfully access the account 3 times. After that, admins will be alerted, and the account will be deactivated.

3.2 System Feature 2 – Student Dashboard

The student account type has access to a student dashboard which can generate courses from set parameters.

3.2.1 Description and Priority

The student dashboard is where students may generate a course schedule. Students are able to decide the amount of credit hours per semester they would like to take, the major(s) and minor(s) they would like to take, as well as if they would like to take a Co-Op. This feature is a high priority for the Degree Admin project.

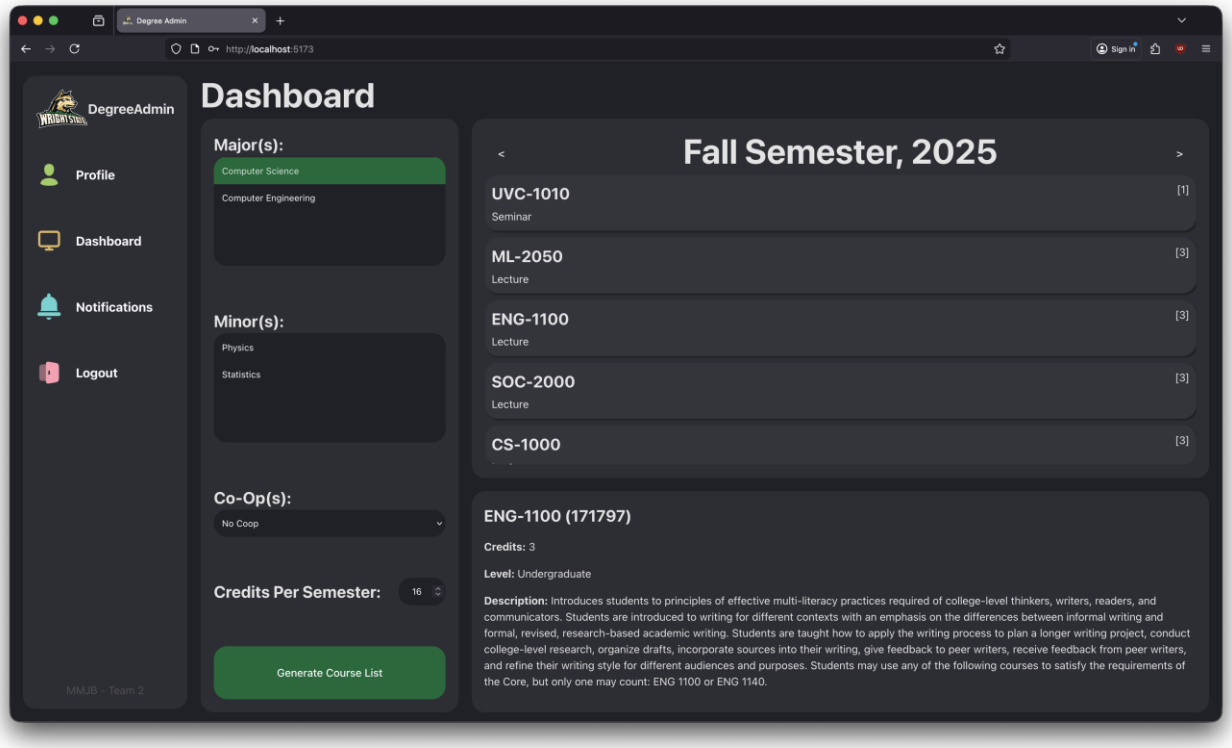


Figure 2 - A screenshot of the Student Dashboard

3.2.2 Stimulus/Response Sequences

The student must select one or more major(s) and optional minor(s). To select multiple majors or minors, the student must hold the control key (Windows) or the command key (MacOS) when selecting each. From this point the student can select a semester they would like to take a Co-Op, if they would like to continue without taking one, they would leave the field in the position of “No Coop”. Next, the student would enter the number of credits they would like to take per semester.

With all of this information entered, the student can select “Generate Course List,” this will display a pending schedule for the student to look through as they wait for faculty approval.

Within the pending schedule, the student is able to select a course to get more information and details.

3.2.3 Functional Requirements

REQ-1: The student must select a minimum of 1 major. Co-ops, and minor(s) are optional.

REQ-2: The student must enter the number of credit hours per semester.

3.3 System Feature 3 – Faculty Dashboard

The faculty account type has access to the Faculty Dashboard which can view/approve/deny a selected student’s schedule.

3.3.1 Description and Priority

The faculty dashboard is where faculty are able to view/approve/deny a selected student’s schedule. The faculty must enter in the student’s ID to accomplish this. This feature is a high priority for the Degree Admin project.

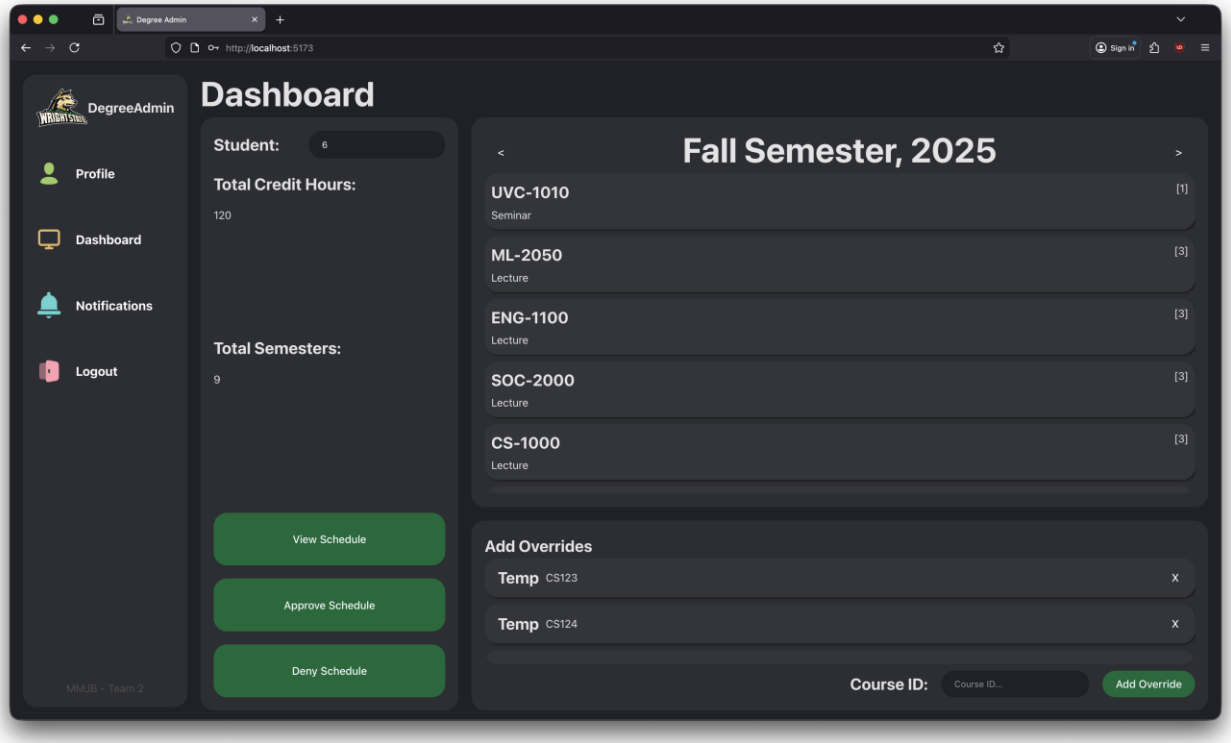


Figure 3 - A screenshot of the Faculty Dashboard

3.3.2 Stimulus/Response Sequences

The faculty must enter a student ID before being able to view/approve/deny a schedule. Once the student ID is inputted the faculty can click a few options:

1. View Schedule: This displayed the pending or approved student schedule for the faculty to review.
2. Approve Schedule: This approves the student's schedule changing it from pending to approved.
3. Deny Schedule: This removes the student's pending schedule.

3.3.3 Functional Requirements

REQ-1: The faculty must be able to enter a student ID.
REQ-2: The faculty must be able to approve or deny the student's course schedule.

3.4 System Feature 4 – Admin Dashboard

The admin account type has access to the admin dashboard which allows administrators to manipulate attributes on different accounts.

3.4.1 Description and Priority

The admin dashboard is the place where administrators are able to modify usernames, passwords, and assigned faculty with all user accounts. Administrators also are able to create new accounts and assign them an account level (Student/Faculty/Admin). This feature a medium priority for the Degree Admin project.

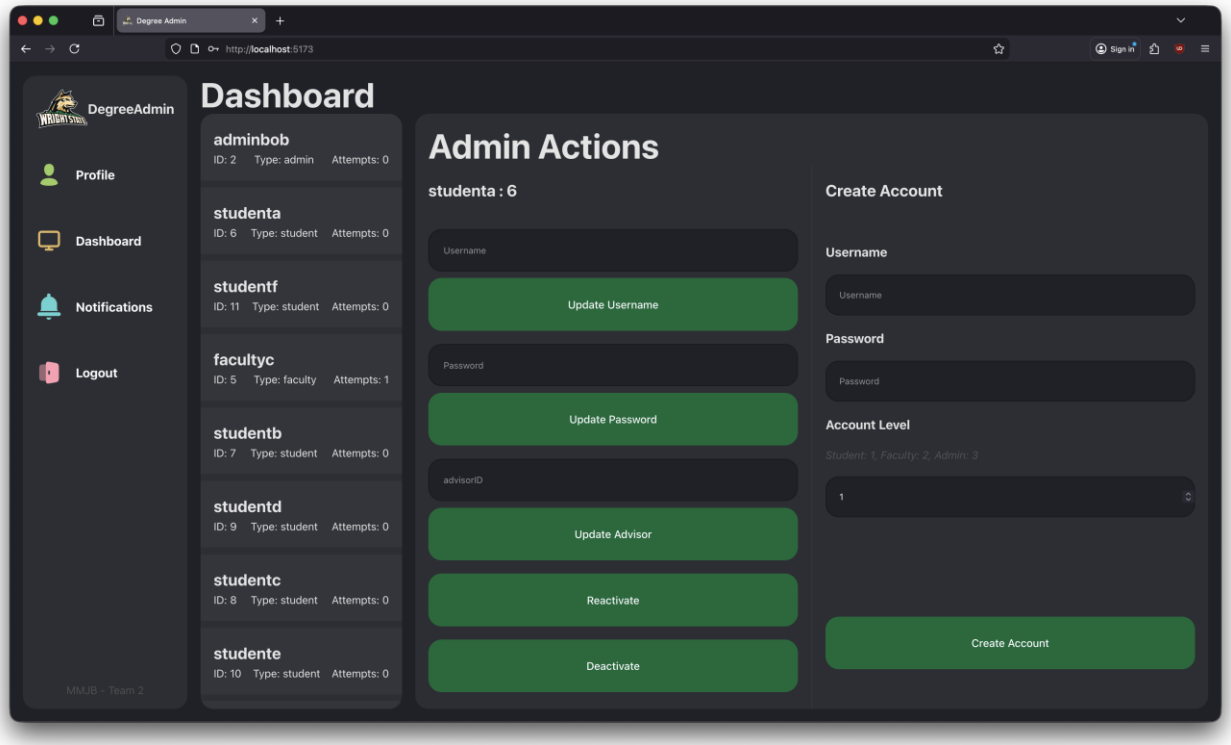


Figure 4 - A screenshot of the Admin Dashboard

3.4.2 Stimulus/Response Sequences

To modify attributes associated with accounts, the administrator must first select an account from the left panel. Once an account is selected, the administrator is able to modify the following for the account: Username, Password, and Advisor. The administrator is also able to reactivate and deactivate the account. Reactivating account typically will be required when a user locks themselves out after 3 incorrect login attempts.

To create an account the administrator must set a unique username and password. Then the administrator must select an account level: student = 1, faculty = 2, administrator = 3. With all of this information, the administrator can create the account.

3.4.3 Functional Requirements

REQ-1: The administrator must be able to enter in an account ID to be able to modify attributes.

REQ-2: The administrator must be able to enter in new attributes into input fields to update the respective attributes associated with an account ID.

REQ-3: The administrator must be able to enter in a new username and password into input fields to create a new account.

REQ-4: The administrator must be able to select an account level when creating a new account.

3.5 System Feature 5 – Profile Page

Each account type has access to the profile page which displays quick info.

3.5.1 Description and Priority

The profile page is the place where users can get quick info, this includes major(s) and minor(s) as well as their class level (freshman, sophomore, junior, senior). This feature is a low priority for the Degree Admin project.

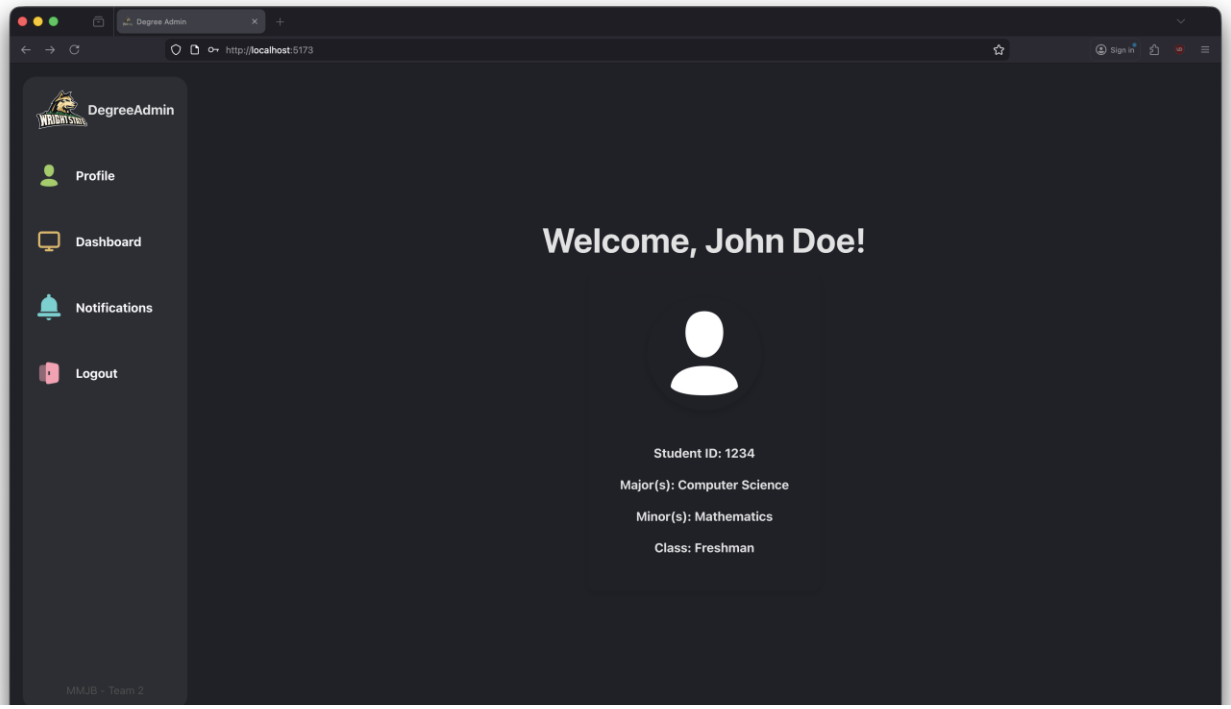


Figure 5 - A screenshot of the Profile Page

3.5.2 Stimulus/Response Sequences

The user must select the profile link with the green icon on the side bar to view the profile page.

3.5.3 Functional Requirements

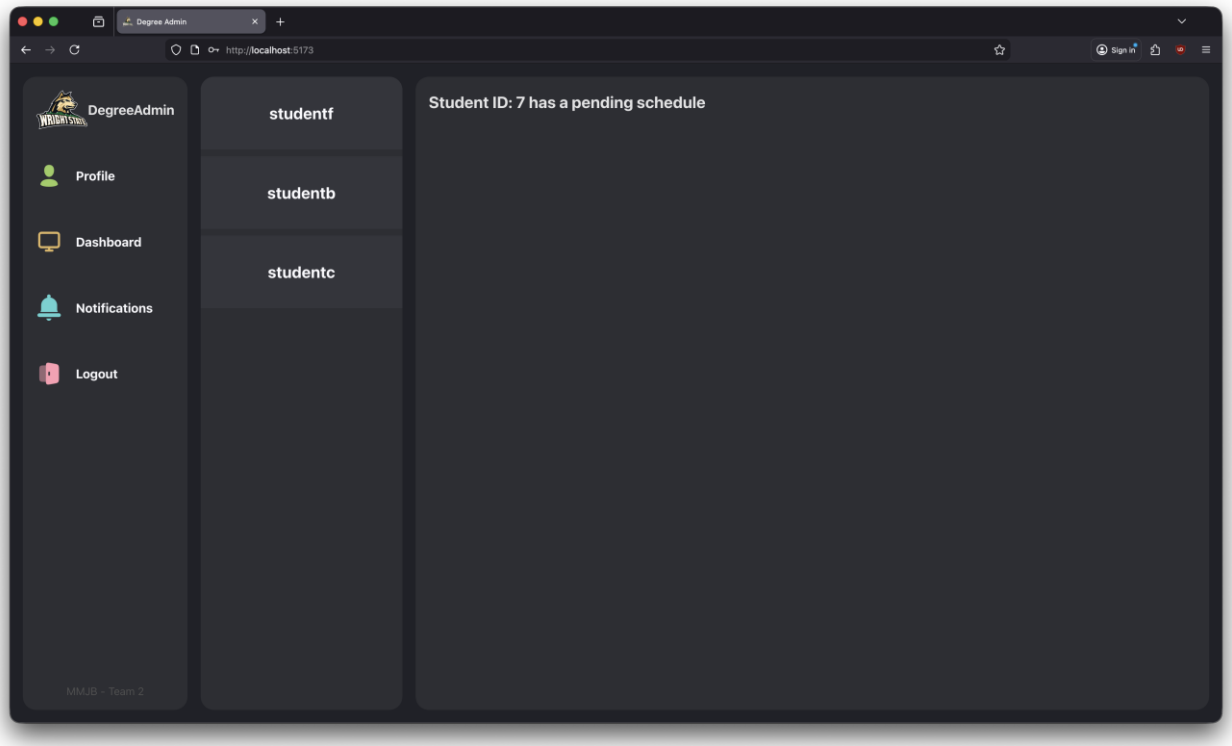
REQ-1: The database must provide the user's selected major(s) and minors.
REQ-2: The database must provide the class year that the user is (freshman/sophomore/junior/senior).

3.6 System Feature 6 – Notification Page

The faculty account type is the only account type that can access the notification page.

3.6.1 Description and Priority

The notification page is the location where faculty members are able to see their assigned student's request for a course schedule. This is a medium priority for the Degree Admin project.



3.6.2 Stimulus/Response Sequences

For the notification area to be used, a student assigned to the faculty account must generate a schedule. Once a student generates a schedule, their account shows up in the

notification page. A faculty can select an account from this list to receive the student ID.

3.6.3 Functional Requirements

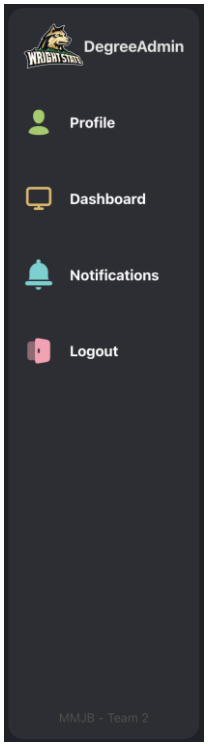
REQ-1: The account type that is accessing the notification page must be faculty.
REQ-2: The faculty must have students assigned to them to have accounts with pending schedules be visible in the notification area.

4. External Interface Requirements

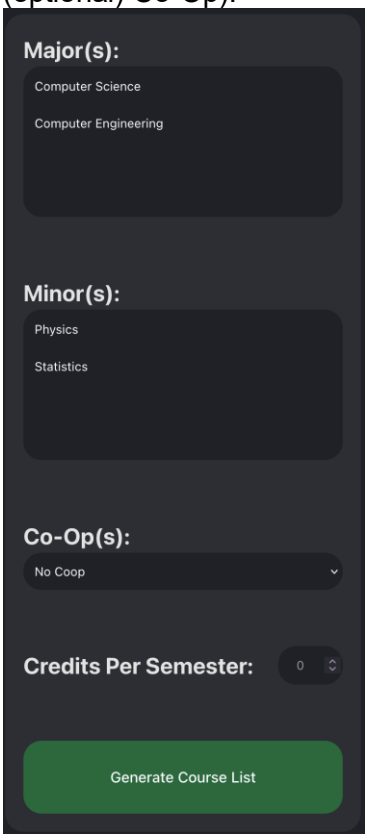
User Interfaces

Degree Admin features a simple and easy to use user interface. The most used features with the Degree Admin GUI are as shown:

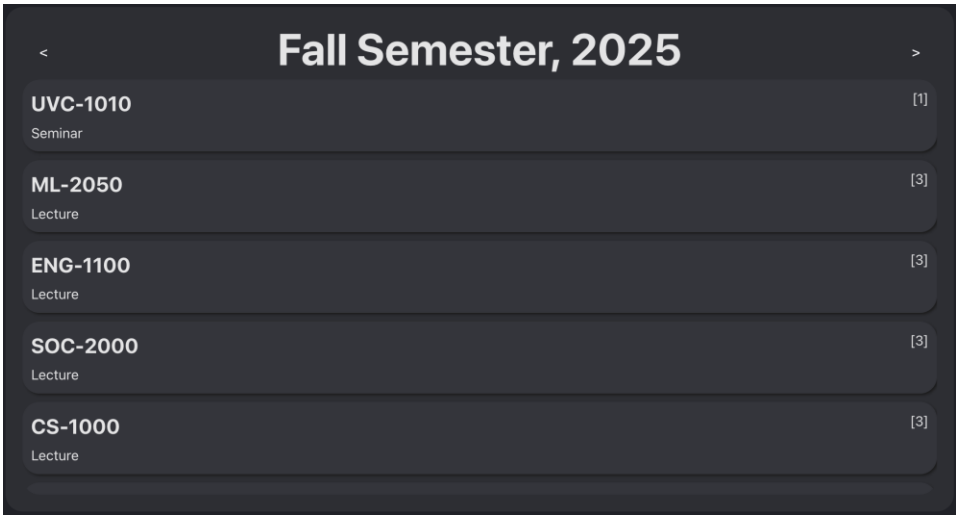
- The side navigation bar:



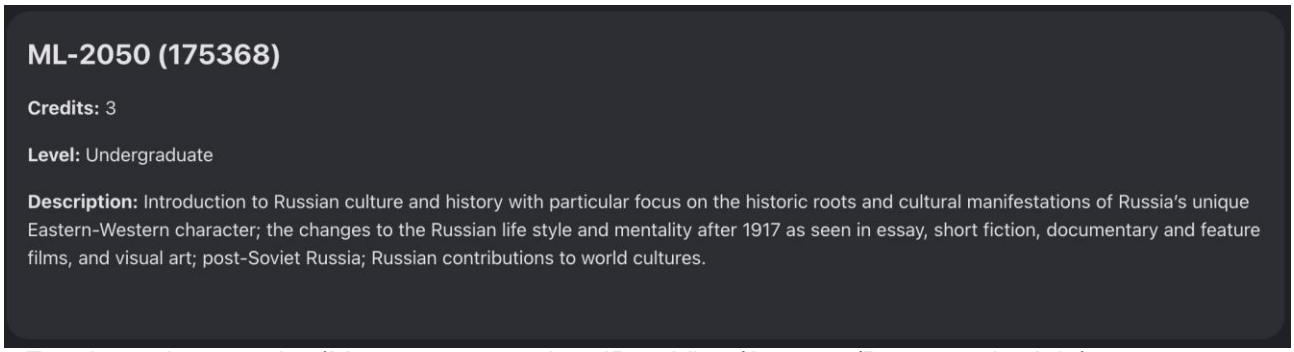
- Student course generator – (Select Major(s) | Minor(s) | Number of credits per semester | (optional) Co-Op):



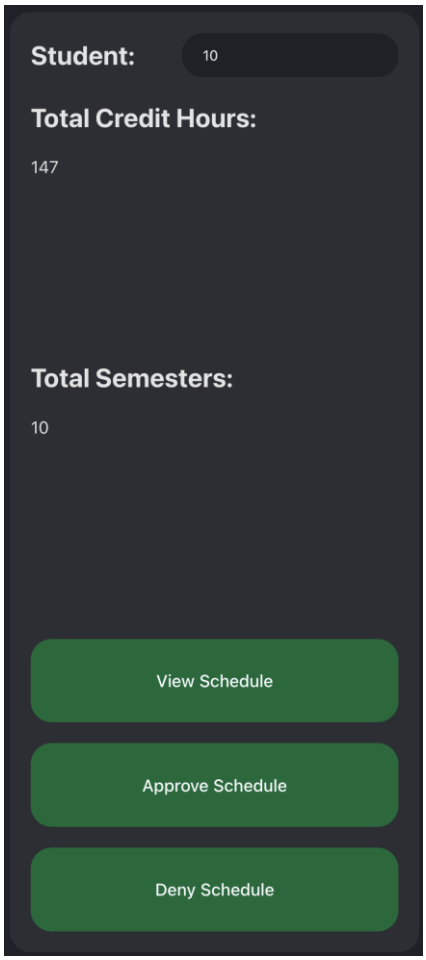
- Course Schedule:



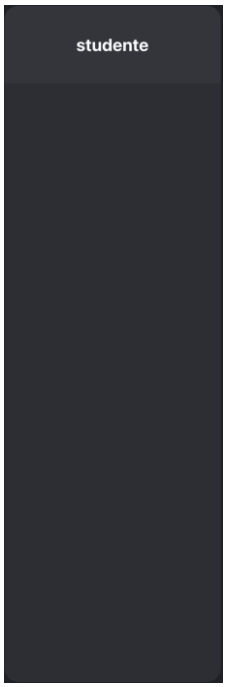
- Student course details:



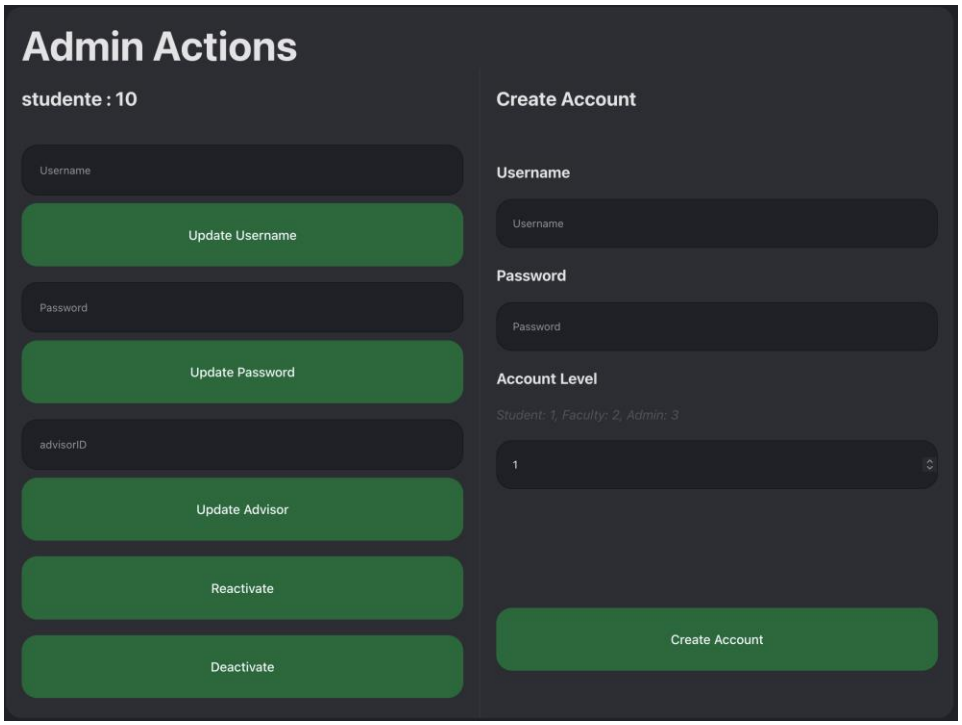
- Faculty action panel – (Must enter a student ID to View/Approve/Deny a schedule)



- Notification selector:



- Admin Actions – (Can update Usernames, passwords, and advisors. As well as creating new accounts):



Hardware Interfaces

N/A

Software Interfaces

Degree Admin is compatible with every system that supports PostgreSQL, Fast API, Vite, and ReactJS. That means it is independent from the operating system of the computer system on which it runs. PostgreSQL serves as the database portion of Degree Admin, while Vite+ReactJS serves the frontend user interface. FastAPI is used as the middleware for Degree Admin connecting the frontend to the backend database.

Degree Admin is capable of running on any platform that supports the following: PostgreSQL, FastAPI, Vite+ReactJS.

Communications Interfaces

Degree Admin is a web application, so network communication and a web browser are necessary to access and use it. The frontend portion of Degree Admin uses a JavaScript library named ‘AXIOS’ to send HTTP requests to the API so that it can serve accurate and formatted information.

To use Degree Admin, a user must sign in with a valid username and password, this information is provided and checked by the backend. If the information is valid, the frontend receives a response containing an account token that is used throughout the application to authorize http requests.

5. Other Nonfunctional Requirements

Performance Requirements

There are no performance requirements for the users of Degree Admin to have long as they are accessing Degree Admin with a device that can run a web browser. The server used to run Degree Admin should be tested to ensure that the hardware can handle the expected load required for each organization. The application is considered real time.

Safety Requirements

The application uses an algorithm to generate course schedules based on a student's selected requirements for major(s), minor(s), and max credit hours. An advisor must approve students' generated schedules. This ensures that a knowledgeable person can answer any questions the student may have regarding different major(s) and minor(s) and workload expected for different credit hour ranges.

Security Requirements

Degree Admin does introduce security requirements by using the OAuth 2.0 protocol for account authentication via username and password. Admin accounts are the only account able to view create, delete, and update student/faculty users. Faculty accounts are only able to view student accounts they manage. This ensures that only necessary accounts are able to view/manage other accounts, thus providing better security.

6. Key Resource Requirements

Major Project Activities	Skill/Expertise Required	Internal Resource	External Resource
Database	Backend Engineer	Mark Harshbarger (Full Time)	N/A
Course Generator	Algorithm Specialist	Mark Harshbarger (Full Time)	N/A
Integrations	API Specialist	Mark Harshbarger (Full Time), Brandon Wittaker (Full Time)	N/A
Login Page	UI/IX Engineer	Jon Wasky (Full Time)	N/A
Frontend Dashboards	UI/UX Engineer	Jon Wasky (Full Time), Mohamed Elkharraz (Full Time)	N/A
Faculty Notification Page	UI/IX Engineer	Jon Wasky (Full Time)	N/A
Profile Page	UI/IX Engineer	Mohamed Elkharraz (Full Time)	N/A
Frontend Wireframes	UI/UX Engineer	Jon Wasky (Full Time)	N/A