Software Requirements Specification

DegreeAdmin

[1.4]

[Samantha Barnum, Aiden Cox, Luke Dawson, Evan Timmons]

[08.30.25]

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Samantha Barnum	8/30/25	Filling out most of what is required.	1.0
Samantha Barnum	11/02/25	Filling out the User Interfaces section.	1.1
Luke Dawson	11/30/25	Revising and making final changes based on finalized front and back end scope	1.2
Luke Dawson	12/09/25	Made final changes before submission	1.3
Luke Dawson	12/11/25	Made a few small revisions	1.4

● Introduction

Purpose

This document defines the software requirements for the DegreeAdmin project, a web-based system that allows students to log in, select a combination of up to two majors and two minors, and generate a schedule that outlines the classes they will need to take based on Wright State's degree paths. Version 1.3, Fall 2025.

Document Conventions

- Plain English
- No special typography
- Requirements numbered REQ-x

● Overall Description

Product Perspective

New, self-contained product developed as part of CEG4110. It is not replacing an existing system.

Product Features

- Login Authentication
- Role Dashboards
    - Student Dashboard (functional)
    - Faculty Dashboard (non-functional, placeholder)
    - Admin Dashboard (non-functional, placeholder)
- Major/Minor selection
- Schedule Generation
- Update notifications

User Classes and Characteristics

- Student: undergrads, in need of a degree path

Operating Environment

Web app. Desktop browser only
Backend: Java 17, Spring
DB: SQL through MariaDB
Frontend: React

Design and Implementation Constraints

- Must use real WSU catalog data for up to 2 majors + 2 minors
    - Economics, Business Major
    - Entrepreneurship, Marketing Minor
- Limited to 13-week semester
- Manual testing allowed
- Requirements doc template must be followed

Assumptions and Dependencies

Course catalog comes from CSV files compiled based on the Wright State Academic Catalog.
Students/faculty/admin sample accounts are fixed.

● System Features

Feature 1: Authentication & Role Dashboards

●.1.1 Description and Priority

This feature allows all users (student, faculty, admins) to log into the system with unique credentials. Based on their role, the system will redirect them to their respective dashboard. Only the student dashboard contains functionality at this time. This feature is **High Priority**, as no other functionality can be accessed without it.

3.1.2 Stimulus/Response Sequences

- User enters username and password.
- System validates that the username is exactly 8 lowercase letters.
- System validates that the password has 8-12 characters, including 1 uppercase, 1 lowercase, 1 digit, and 1 symbol.
- If credentials fail validation, then an inline error message is displayed.
- If credentials pass validation, the user is logged in and redirected to the dashboard corresponding to their role.
- Session ends upon logout or browser close.

3.1.3 Functional Requirements

- **REQ-1**: The system shall enforce usernames to be exactly 8 lowercase characters.
- **REQ-2**: The system shall enforce passwords to be 8-12 characters with >=1 uppercase, >=1 lowercase, >=1 digit, >=1 symbol.
- **REQ-3**: The system shall display inline error messages for invalid login attempts (e.g., missing fields, bad credentials).
- **REQ-4**: The system shall route successfully authenticated users to their role-specific dashboard.
- **REQ-5**: The system shall terminate the user's session when the user logs out or closes the browser.
- **REQ-6**: The system shall provide three role dashboards: Student (functional), Faculty (non-functional), Admin (non-functional)

Feature 2: Degree Planner

3.2.1 Description and Priority

This feature generates a multi-semester plan (N semesters) for students based on selected majors/minors, as well as college and credit hour requirements. It is **High Priority**, as it forms the core of the project requirements.

3.2.2 Stimulus/Response Sequences

- Student selects 2 majors and 2 minors (or equivalent).
- System generates a term-by-term plan that respects college needs and overall credit hour requirements.

3.2.3 Functional Requirements

- **REQ-7**: The system shall allow students to select two majors and two minors from the catalog. Electives may be hardcoded as a fixed set, per Q&A.
- **REQ-8**: The system shall generate a semester-by-semester plan that satisfies degree and overall credit hour requirements.
- **REQ-9**: The system shall enforce a flexible credit load and label each semester based on how heavy the credit load is
- **REQ- 11**: The system will allow students to select whether or not they wish to take courses in the summer term

Feature 3: Co-op Requests & Notifications

3.3.1 Description and Priority

This feature allows students to request semester-long co-ops. This feature is **High Priority**.

3.3.2 Stimulus/Response Sequences

- Student selects a number of co-op semesters they will be engaging in
- System generates an adjusted plan reflecting the co-op term.

3.3.3 Functional Requirements

- **REQ-13**: The system shall allow students to request co-ops for a specific semester.
- **REQ-14**: The system shall generate an adjusted plan based on the co-op request.
- **REQ-17**: The system shall display notifications to students as a badge counter (e.g., “+1”) upon login.

● External Interface Requirements

User Interfaces

- **GUI Standards**
  - Material design-inspired component system with consistent spacing and typography.
  - Light and dark theme modes with user-toggleable preference (persists across sessions).
  - Responsive layout supporting desktop (>900px), tablet (720-900px), and mobile (<720px).

- **Screen Layout**
  - Persistent navigation bar across all authenticated pages containing: logo, role-specific navigation links, theme toggle, notifications bell, user info, and logout button.
  - Centered content area with maximum width constraints for readability.
  - Card-based content containers with consistent borders and shadows.
  - Active page highlighted in navigation using distinct background color.

- **Standard UI Components**
  - **Buttons**: Primary (accent-colored), secondary (neutral), and ghost (transparent) variants with consistent hover/focus states.
  - **Forms**: Labels positioned above inputs, inline validation messages, required field indicators.
  - **Notifications Bell**: Header icon displaying unread count badge; clicking opens dropdown popover with recent notifications.
  - **Theme Toggle**: Sun/moon icon button in header for switching between light and dark modes.
  - **Error Messages**: Inline display below form fields with distinct red background and clear text.

- **Screens Requiring User Interface** (see attached screenshots in Appendix):
  1. Login screen - username/password authentication with demo credentials.
  2. Student Dashboard - welcome message and feature tile grid.
  3. Degree Plan - major/minor selection form with generated plan results.
  4. Notifications Page - filterable list with unread highlighting.
  5. Faculty Dashboard - overview and navigation tiles (placeholder, non-functional).
  6. Admin Dashboard - system management tiles (placeholder, non-functional).

- **Keyboard Shortcuts**
  - Tab: Navigate between interactive elements.
  - Enter: Submit forms, activate buttons.
  - Escape: Close popovers and modals.

- **Accessibility Standards**
  - All interactive elements have visible keyboard focus indicators.
  - ARIA labels provided for icon-only buttons.
  - Semantic HTML with proper heading hierarchy.
  - Color contrast ratios meet WCAG AA standards.

Hardware Interfaces

No hardware requirements beyond standard PC.

Software Interfaces

- **React (v18+)**
    - **React Webpage**: Provides students with a user interface for all student functionality
    - **React Router**: Client-side routing for single-page application navigation.
- **MariaDB (Latest LTS)**: Access to SQL database formed from CSV files
- **Bruno (v.2.15.0)**: Allows HTTP requests to be sent to the backend to retrieve or change info
- **Spring Boot (v3+)**: Handles backend logic and HTTP requests to access the database
- **Java (v17+)**: The language that the backend logic is written in

Communications Interfaces

HTTP/HTTPS: Standard web protocols for client-server communication.

● Other Nonfunctional Requirements

Performance Requirements

Should generate a degree plan in under 2 seconds for the required data set.

Safety Requirements

- Low risk
- No real student data
- Only demo accounts.

Security Requirements

- Basic login enforcement (Username/Password requirement enforcement)
- Session ends on logout or browser close
- No persistent sessions

- **Key Resource Requirements**

Major Project Activities	Skill/Expertise Required	Internal Resource	External Resource
Frontend UI	React, CSS	Group Members	React libraries
Backend Logic	Spring Boot, Java	Group Members	Spring Boot, Java
DB setup	MariaDB, Bruno	Group Members, CSV data files	MariaDB, Bruno, WSU course catalog
Testing	Manual Testing	Group Members	
Documentation	Technical Writing	Group Members	