

Team 11's Sprint 1 Requirements Artifact for Schedule Builder - Addison Bartelli, Anya Combs, Janna Dungao, Hunter Long, Elizabeth Miller, Marco Martinez Reyez

## 1. System Overview

The schedule builder will provide an intuitive user interface that allows users to build schedules at the University of Kansas for past, current, or future semesters. It will allow users to save built schedules and edit the saved schedules. It will also prevent users from adding courses with conflicting meeting times.

## 2. Functional Requirements

As a user, I want to see a login page where I can sign up for an account and login to a created account.

### 2.1. Account Creation – Janna

- FR1.1: The system shall provide a button in the login page to direct the user to an account creation page.
- FR1.2: The system shall provide three forms for the user's email, password, and password confirmation, as well as a submit button.
- FR1.3: The system shall send a confirmation email to the user and require confirmation before being able to login.
- FR1.4: The system shall confirm that the password meets the minimum requirements (lowercase letters, uppercase letters, digits, and symbols).
- FR1.5: The system shall confirm that the password matches the password confirmation.

### 2.2. Account Login – Janna

- FR2.1: The system shall provide a login page with forms for the email associated with the user's account and the account's password, as well as a submit button.
- FR2.2: The system shall provide a reset password feature.
- FR2.3: The system shall provide a sign out button.

### 2.3. Account Authentication and Security – Janna

- FR3.1: The system shall use jwt tokens that expire within a reasonable time.
- FR3.2: The system shall use the NextAuth and Supabase for account authentication and storing account data securely.

As a user, I want to be able to see my selected courses and have options to manage them.

### 2.4. Display Selected Courses – Anya

- FR4.1: The system will display all selected courses to the right of the schedule grid.
- FR4.2: The display will have a delete button for each course.

As a user, I want to be informed when I cannot add a course due to scheduling conflicts.

### 2.5. Display Conflict Message – Anya

- FR5.1: The system will display a message if all offerings of a course conflict with the current schedule.

As a user, I want to be able to search for courses via typing into a search bar.

#### 2.6. Course Search Bar – Marco

- FR6.1 The user shall input a string or characters or the exact class number into a search bar, then they shall be able to see and click on the courses that match their search.

As a user, I want to be able to see all the information related to the course.

#### 2.7. Course Information Dropdown – Marco

- FR 7.1 The user shall click on one of the results from the search bar and be redirected to a popup that displays extended course information. Through this popup, the user will also be able to add courses to their schedule.

As a user, I want to be able to add and remove courses.

#### 2.8. Add/Delete Courses – Marco

- FR 8.1 The user shall be able to add courses to a schedule by clicking on the ‘add’ button on a queried course search result. The user shall be able to delete a course from their schedule by clicking on the ‘delete’ button on the selected course.

As a user, I want the program to have functionality for saving my schedules online.

#### 2.9. Save Schedule – Addison

- FR9.1: The system shall be able to read the current schedule and save it to the database for the authenticated user

As a user, I want the program to have functionality editing or overwriting my saved schedules.

#### 2.10. Edit Saved Schedule – Addison

- FR10.1: The system shall be able to overwrite schedules using the save functionality

As a user, I wish to be able to delete my schedules from the online database.

#### 2.11. Delete Schedule – Addison

- FR11.1: The system shall be able to delete schedules from the database if they belong to the authenticated user

As a developer, I want a dedicated database hosted by Supabase to store and fetch data from.

#### 2.12. Course Information Database – Hunter

- FR12.1: The database shall be accessible through the Supabase API.

#### 2.13. User Information Database – Hunter

- FR13.1: The database shall store information on each user to facilitate accounts and storing schedules.

#### 2.14. Add Additional Course Information to Database – Hunter

- FR14.1: Add course names, credit hours, and descriptions to database

As a user, I want to interact with a clean and easy-to-use interface.

#### 2.15. UI Improvements – Elizabeth, Janna

- FR15.1: The system shall provide a clean user interface that is simple, cohesive, and complete.
- FR15.2: The system shall provide a UI that is intuitive and does not require instruction.