Requirement Document: FIT Schedule Planner

1. Introduction

1.1 Purpose

The FIT Schedule Planner is designed to simplify the class registration process for students by integrating multiple existing systems into a single user-friendly interface. It aims to help students manage their class schedules, track their academic progress, and access all relevant information from one platform.

1.2 Scope

The FIT Schedule Planner will consolidate functionalities from PAWS, CAPP Degree Evaluations, FIT programs, and Rate My Professor. It will provide class schedule building, prerequisite checks, availability monitoring, class filtering, and progress tracking.

2. Functional Requirements

2.1 Feature: Class Scheduling

Users should be able to view available classes, add them to their schedule, and manage any time conflicts. Their schedule should be depicted as classes are added and time conflicts should be highlighted.

Sample Input/Output:
Input: User selects a class not already in their schedule and clicks “Add”
Output: “(Class name) was susseccefully added.” (Schedule depiction updates to show added class)
Input: User does not select any class and clicks “Add”
Output: “No class was selected, please try again.” (Schedule depiction does not change)
Input: User selects a class that is already in their schedule and clicks “Add”
Output: “(Class name) has already been added to your schedule.” (Schedule depiction does not change)
Input: User selects a class that overlaps another class time and clicks “Add”
Output: “(Class name) was susseccefully added, but there is a time conflict.” (Schedule depiction updates to show added class and highlights the overlapping classes)

2.2 Feature: Prerequisite Checking

When users add a class with prerequisites, the system should check if the user has met the requirements. If the requirements are met, then the class should be added to their schedule. If the requirements are not met, a popup should appear stating that the class has prerequisites, listing the prerequisites, and asking if the user would still like to add the class to their schedule (the system should ask since closed class forms exist and allow some students to waive prerequisites).

Sample Input/Output:
Input: User tries to add a class with prerequisites and they meet those prerequisites
Output: "(Class name) was susseccefully added." (Schedule depiction updates to show added class)
Input: User tries to add a class with prerequisites and they do not meet those prerequisites
Output: Popup appears, lists prerequisites, and asks user, "Would you still like to add (Class name) to your schedule?"

> Input: User clicks "No"
> Output: (Popup disappears)
> Input: User clicks "Yes"
> Output: "(Class name) was susseccefully added." (Schedule depiction updates to show added class and popup disappears)

2.3 Feature: Class Filtering

When searching for classes to add, users should be able to filter the list by subject, professor, times, course number, credit range, or RateMyProfessor score. These filters should help the user narrow down and organize the list of classes so they can find the classes they are looking for.

Sample Input/Output:
Input: User selects no filter and clicks "Apply Filters"
Output: All of the classes are displayed in alphabetical/chronological order
Input: User selects a professor filter and clicks "Apply Filters"
Output: All classes taught by that professor are displayed in alphabetical/chronological order
Input: User selects 3000+ course filter, RateMyProfessor filter, and clicks "Apply Filters"
Output: All classes that are numbered 3000 or more are displayed in decreasing RateMyProfessor score order (higher rates at top)
Input: User selects certain filters that do not contain a single class (for example: Dr. Chan teaching communication classes) and clicks "Apply Filters"
Output: The list will say "No Classes Available"

2.4 Feature: Degree Evaluation and Progress Tracking

Users should be able to view their progress toward their degree program. The system should be able to access FIT programs from the FIT website, access user’s CAPP Degree Evaluation (once credentials are provided), and combine them to generate a checklist. The checklist will show the needed credits for their program, show the credits they have fulfilled, and recommend credits they should aim to fulfill during the upcoming semester.

Sample Input/Output:
Input: User selects a program and successfully uploads their CAPP Degree Evaluation
Output: A checklist is depicted and certain classes are recommended
Input: User does not select a program and successfully uploads their CAPP Degree Evaluation
Output: “A program must be selected.”
Input: User selects a program and unsucessfully uploads their CAPP Degree Evaluation
Output: “Something went wrong with your CAPP Degree Evaluation, please try again.”
Input: User does not select a program and unsuccessfully uploads their CAPP Degree Evaluation
Output: “A program must be selected.”

3. Non-Functional Requirements

The system should respond to user actions within a couple seconds unless it is processing data. Processing data should not take much longer and a progress bar should be depicted to show that the system is working. Considering the fact that users will not need to connect to each other in order to use the system, the system should be able to handle any number of simultaneous users without performance degradation. No databases are required and therefore no information should be stored for security purposes. Once a user closes the system, all of that user’s information should disappear with them (especially their CAPP Degree Evaluation). the user interface should be consistent and easy to navigate with clear labels. Contextual help should be provided throughout the interface.

4. Constraints

The system should be accessible on both Windows and macOS machines. If developing a web application is decided, it should be compatible with major browser like Chrome and Safari. It must integrate existing systems for PAWS and CAPP Degree Evaluation. Development will involve a team of two developers and one project advisor.

5. Assumptions

It is assumed that users of this system will have access to their CAPP Degree Evaluation. It is also assumed that the tools and technologies required for the development of this project will be compatible with the user's system.

6. Acceptance Criteria

6.1 Functional Testing

All core features (schedule building, prerequisite checking, filtering, and progress tracking) must operate as described in functional requirements. The system should handle valid and invalid inputs appropriately, providing feedback when necessary.

6.2 Non-Functional Testing

Response times should meet the specified requirements. No personal data should be saved or stored anywhere as specified. Users should be able to navigate the interface without confusion.

7. Dependencies

- FIT Programs are up to date on their website
- FIT class times and availability are up to date on their website
- Abilty to access CAPP Degree Evalkuations
- RateMyProfessor scores are up to date on their website