# Test Plan for FIT Schedule Planner

## 1. Introduction

This test plan displays the intended approach towards the verification of all functionality and performance for the FIT Schedule Planner, which has the basic role of integrating class registration, degree evaluation, and progress tracking into a single interface. It details core main features and behaviors explained in the requirements document with the idea of assuring the system operates as expected-both with typical and atypical inputs.

## 2. Test Cases

### 2.1 Feature: Class Scheduling

Ensure the user can see and add classes to their schedule, and also handle time conflicts.

Test Case 1.1: AddingNew Class Successfully
    Input: User selects a class which is not already in their schedule
    Expected Output: The schedule updates and display the added class on their schedule

Test Case 1.2: Class Already Added
    Input: User selects a class that is already in their schedule
    Expected Output: The schedule is not changed. Ask the user that this class has been added in schedule already.
    Edge Case: Same class entries but with a different section or time

Test Case 1.3: Time Conflict
    Input: User chooses class to add, the class timings of which clash with the previously added class in his schedule
    Expected Output: The schedule is not changed. Ask the user that this class hasn't been added because it clashes with another class timings.

### 2.2 Feature: Classes Prerequisite Check

Ensure that upon adding of new class, all prerequisite requirements of classes are appropriately satisfied.

Test Case 2.1: User adds a Class which they Met Prerequisites
    Input: User attempted to add a class which prerequisites are met
    Expected Output: The schedule updates and display the added class on their schedule

Test Case 2.2: User adds a Class which they do Not Meet Prerequisites
    Input: User attempts to add a class which prerequisites are not met
    Expected Output: The schedule remains unchanged. Alert the user that he/she does not
    meet the prerequisites of the class they have chosen, and provide the prerequisites.

## 2.3 Feature: Class Filtering

Test that class filtering functions properly.

Test Case 3.1: No Filters Applied
    Input: No filters selected by user
    Expected Output: Display all classes in alphabetical/chronological order

Test Case 3.2: Professor Filter is Applied
    Input: User applies a professor filter
    Expected Output: List all classes taught by selected professor in alphabetical/
    chronological order

Test Case 3.3: Multiple Filters Applied
    Input: User selects 3000+ course number and RateMyProfessor score
    Expected Output: Display classes numbered 3000+, sorted by decreasing order of
    RateMyProfessor score

Test Case 3.4: Conflicting Criteria
    Input: User selects filters which return no results; such that a selected professor teaches
    no classes in the selected subject
    Expected Output: "No Classes Available.

## 2.4 Feature: Degree Evaluation and Progress Tracking

Degree evaluation and progress tracking works, displaying correct information.

Test Case 4.1: Successful Degree Evaluation Upload
    Input: User selects a program and uploads their CAPP Degree Evaluation successfully
    Expected Output: Checklist of suggested credits and classes

Test Case 4.2: No Program Selected
    Input: User does not select a program and uploads their CAPP Degree Evaluation
    Expected Output: Error message "A program must be selected."

Test Case 4.3: Failed Degree Evaluation Upload
     Input: User selects a program but the CAPP Degree Evaluation upload fails
     Expected Output: Error message "Something went wrong with your CAPP Degree
     Evaluation, please try again."

Test Case 4.4: No Program and Failed Upload
     Input: User does not select a program, and the CAPP Degree Evaluation upload fails
     Expected Output: Error message "A program must be selected."

## 3. Non-Functional Testing

Ensure the system responds promptly to user actions, efficiently handles data processing, and the user interface is intuitive and easy to navigate.

Test Cases:
-   Measure response times for user actions, aiming for responses within a few seconds
-   Confirm that closing the system clears all session data
-   Assess ease of navigation through different features
-   Verify the availability and clarity of contextual help throughout the interface