System and Unit Test Report

A. User Story from Sprint 1: As a user, I want a website where I can see the list of courses for the current computer science catalog.

B. User Story from Sprint 1: As a user, I want the website to have a basic user-interface.

Components: App, Sign Up, NavBar, How-it-Works, Course Entry, Course Context (When you click on submit, the CourseContext file is in charge of defining where a class go, i.e. a class is a requirement, a capstone, DC, etc.) AuthContext (Firebase Authorization functions), UserContext (in charge of storing and providing current user data).

Scenario:

1) Start majorCoursePlanner App; select sign up; type
    - First name: John
    - Last Name: Doe
    - Password: password
    - Hit enter or click sign up
    - Click on the “How it Works” tab to see how the course planner works
    - Click on course entry to enter the classes you have taken already
    - Click on “I HAVE TAKEN ALL LOWER DIVISION COURSES” if you have taken all the lower division courses (i.e., you are a senior student looking for capstone and other upper division courses)
    - Click on done once you are done adding courses

Notes: Later in our process we decided to only let logged in users be able to use the app, so initial sprint stories don't indicate that a user needs to be logged in because it was decided in the following sprints.

A. User Story from Sprint 2: As a user, I want the website to be able to display graduation progress.

B. User Story from Sprint 2: As a user, I want the website to be able to save my current progress and plan.

Components: App, NavBar, SignIn, AuthContext (Firebase Authorization functions), UserContext (modify the user in database with new information), Dashboard, AccountSettings.

Scenario:

1) Start majorCoursePlanner App; if user already has an account
    - Select sign in
    - Enter email and password
    - Hit enter or click on sign in
    - Click Graduation Status
    - Click account settings
    - Select a major, seniority and status
    - Sign out
2) If user doesn't have an account, then follow the steps from sprint 1 and follow the same steps mentioned earlier in this scenario.

Notes: Student information is saved on a database, so even if they sign out, the information is still saved.

A. User Story from Sprint 3: As a user, I want to see recommend courses based on their workload, difficulty, and other factors.

B. User Story from Sprint 3: As a user, I want the website to have a full-featured user interface.

   Components: App, SignIn, UserContext, Dashboard, Next-Quarter-Planner, CourseContext, CourseDrawer

   Scenario:

   1) Start majorCoursePlanner App; if user already has an account
      - Select sign in
      - Enter email and password
      - Hit enter or click on sign in
      - Click on the next quarter planner
      - Based on the classes recommended, pick one
      - The classes are color coded to reference a specific requirement like Lower division requirements, DC, Capstone and upper division elective courses
      - After adding classes, click on cart
      - If you need to drop a class click on trash
      - Once you are done click on submit
      - Classes added will be displayed on the home page and will say ”In Progress”

A. User Story from Sprint 4: As a user, I want to see recommended courses and my graduation progress.

B. User Story from Sprint 4: As a user, I want the website to be easier to use than the official planner.

Components: App, SignIn, UserContext, Dashboard, Next-Quarter-Planner, CourseContext, CourseDrawer

Scenario:

1) Start majorCoursePlanner App; if user already has an account
    - Select sign in
    - Enter email and password
    - Hit enter or click on sign in
    - Click on next quarter planner
    - Look at the color-coded classes that easily differentiate the different requirements, take as many classes as you want.
    - Click on cart
    - Click on submit
    - Click on graduation status and see your graduation progress