

SW Engineering CSC 648/848 Section 4
Milestone 4 Submission Form
November 27, 2022

MyDay

Team 7

Nya Avelina Bautista – Team Lead
Paula Abigail Tam – Front End Lead
Christine J’Ursey – Back End Lead
Henry Cai – Github Master
Francis Ranallo – Scrum Master

History Table

Revision ID	Revision Date	Revised By
1	11/26/2022	Nya Bautista
2	12/04/2022	Nya Bautista

1) Product Summary

Our application, MyDay, is the new innovative, multi-functional scheduler for busy students at San Francisco State University. Our team at MyDay strives to utilize the functionalities of a calendar while taking into consideration routine breaks and loss of habits that may be forgotten in the future.

What Features Does MyDay Have?

- ❖ *Account Management* - The first page that the user will see when visiting MyDay for the first time is the Home page where the user will be able to register with their SFSU ID and password. Additionally, for recurring users, there is a separate login page. Once logged in, the user will be able to log out as well and they will know they are logged out once the Home page appears.
- ❖ *Scheduler* - Users are able to visualize their schedule by creating, deleting, and updating events in our Scheduler function. MyDay additionally gives users the option to visualize their schedule in a Daily, Weekly, or Monthly format. When creating an event, users can additionally create preferences that MyDay will use to suggest other events based on the user's event location, selection of Before/After relativity, and tags.
- ❖ *Suggested Events* - Once the user has inputted their schedule, MyDay will take into account the preferences that were inputted per each event and suggest locations or activities that fit the user's downtime and "empty spaces".
- ❖ *Map* - MyDay users also will be able to utilize our Map page. This page allows users to view locations to visit in between classes to rest, eat, study, or socialize. While viewing a map of SFSU that shows where their classes are, the users also will be given the walking distance and walking duration between their classes based on their schedules.

Why MyDay?

MyDay incorporates the functionalities of a calendar and a map into one application, but our team at MyDay highly encourages a healthy balance in between events in a college student's life. We understand how difficult it may be to remember to do the simplest and vital activities, such as eating or working out, due to the stress of being a college student. MyDay incorporates suggestions to give a light nudge to our users to be able to utilize what the campus has to offer.

Try Out MyDay Today! – <https://csc-648-team07-project.tk/>

2) QA Testing

❖ Unit Test

❖ [Github URL of Test Cases](#)

❖ Functional & Statement Coverage:

- Sign Up/Sign In - This test case was to show the functionality of the user being able to input their information and their ability to create an account. Additionally, the test should check an existing user's information and the same user to be able to log back in.
- Scheduler - This test case tested if the scheduler shows the correct information needed for the user to populate onto the scheduler. The test additionally checks if the scheduler displays the correct tabs such as a Daily View, Weekly View, and Monthly View. Additionally, the test checks for the correct preference tabs and the event APIs.
- Map - This test case checks for the correct display of the Google Map API. The test case additionally checks if the user is able to view the map correctly when the page is loaded.
- Suggestions - This test case checks if the suggestions change based on the location of the event the user is at and if the suggestions tab is visible to the viewer.
- NavBar - This test case checks if the NavBar is visible to the user and if the user is able to view the tabs implemented onto the NavBar.

❖ Integration Test

❖ [Bug Tracking System Link](#)

- If the link does not work, please use the following information to log into our project to view our Bug Tracking System.

- User: csc648.fa22.04.team07@gmail.com
- Password: p4\$\$w0rd++

- ❖ Coverage Analysis: For each of our P1 features, we tested from a user's perspective of the steps one would follow and the results we expect from each step. Additionally, we took into consideration of the user experience and if the tests passed or failed.

❖ Sign Up/ Sign In:

Test Case ID: 002	Description: Testing Registration		
Date Tested: 11/30/2022	Created By: Frankie		
Tester's Name: Frankie	Test Case (Pass/Fail): Pass		
Steps Followed:	Expected Results:	Actual Results:	Pass/Fail:
1. Enter test values into registration form 2. Click register button	1. Form data is returned	As expected	Pass
➤ 1. Click on the sign in button	1. Go to the sign in page	As expected	Pass

❖ Scheduler:

Test Case ID: 001	Description: Testing Scheduler		
Date Tested: 11/18/2022	Created By: Henry		
Tester's Name: Henry	Test Case (Pass/Fail): Pass		
Steps Followed:	Expected Results:	Actual Results:	Pass/Fail:
1. Go to home page containing scheduler	1. Display, headers, and column are visible and contains the correct text	As expected	Pass
1. Click on any cell in the scheduler to show a popup window for creating an event	1. Content of popup window is visible when the cell is clicked on	As expected	Pass
1. Click on "Add Prefs" button to view preferences	1. Preference button is visible. Elements in tags and relative dropdown are visible	As expected	Pass

➤
❖ Map:

Test Case ID: 001	Description: Testing the Google Maps API functionality		
Date Tested: 11/18/2022	Created By: Nya		
Tester's Name: Nya	Test Case (Pass/Fail): Pass		
Steps Followed:	Expected Results:	Actual Results:	Pass/Fail:
1. Maps page is visited by any browser.	1. Maps page is visible with the ability to input current location and destination.	As expected	Pass
2. Input a current location and destination location and loads correct, manageable route.	2. Current location and destination location are easily located on the API. Routes have a calculated walking distance and walking time shown.	As expected	Pass
3. Input two events on the scheduler to test the map displaying a calculation of the current location and destination location.	3. Map should show the event location and the location of the next event.	As expected	Pass



❖ Suggestions:

Test Case ID: 001	Description: Suggestions show up		
Date Tested: 11/27/2022	Created By: Christine J'usrey		
Tester's Name: Christine	Test Case (Pass/Fail): Pass		
Steps Followed:	Expected Results:	Actual Results:	Pass/Fail:
Login as 999888777 password Password1*	Login successful scheduler shows everyday has event called test from 12:00am-12:00am	As Expected	Pass
click button suggestions	It possible to click button	As Expected	Pass
drop down menu shows	Shows 1-5 suggestions or "No suggestions"	As Expected	Pass

❖ NavBar:

Test Case ID: 001	Description: Testing Navigation Bar		
Date Tested: 11/28/2022	Created By: Paula Abigail		
Tester's Name: Paula Abigail	Test Case (Pass/Fail): Pass		
Steps Followed:	Expected Results:	Actual Results:	Pass/Fail:
1. Click on the upper right icon. 2. Click on the upper right hand icon again.	1. Dropdown menu with options appear in the upper right of the screen. 2. Dropdown menu disappears and the Navigation Bar is in its default state.	As Expected	Pass
1. Click on the upper right icon. 2. Click on the first item of the dropdown menu. 3. Repeat steps 1-2 with the rest of the items in the dropdown menu.	1. The dropdown menu shows the options "Sign In", "Sign Up", and "About Us". 2. Clicking the first item, "Sign In", navigates to the Sign In page. 3. Clicking the second item, "Sign Up", navigates to the Sign Up page. 4. Clicking the third item, "About Us", navigates to the About Us page.	As Expected	Pass
1. As a signed in user, click the upper right icon.	1. The dropdown menu contains the items "Profile", "About Us", and "Sign Out".	As Expected	Pass

3) Code Review

a) Coding Style

- a. For our coding style, we utilized the [Google JavaScript Style Guide](#) where we specifically enforced the Naming conventions and Functions features. When we first began creating our application, we had similar coding styles and we did not need to enforce the coding style as much, since we were focusing on finalizing our functionalities of our application. We enforced our coding style predominantly when reviewing Github Pull Requests.

b) Github Pull Requests

1. [Scheduler](#)
2. [NavBar](#)
3. [Google Maps](#)
4. [Sign-In](#)
5. [Registration](#)
6. [Table Schemas](#)
7. [M3](#)
8. [CSS](#)
9. [Creating Event Fixes](#)
10. [User Registration](#)