SOSY Scheduler

Requirements and Specification Document

12/03/2024, Version 1

24/03/2024, Version 2

Project Abstract

SOSY Scheduler is an online web application catered for students in the Software Systems (SoSy) Major at Simon Fraser University. Building upon the foundation of their current course offerings display (https://sfussss.org/courses), SoSy Scheduler enhances the existing webpage by providing personalized degree planning. Students can view required SoSy courses, track academic progress, plan across multiple semesters, and preview upcoming courses for semester schedules. The planning experience that this application provides aims to alleviate the stress of a student's university journey by facilitating an all-in-one seamless navigation.

Competitive Analysis

GoSFU's MySchedule tool is the primary schedule planning platform for SFU Students. It offers upcoming semester courses, visual scheduling, and the ability to save multiple schedules. However, its limitation to the current semester can pose a challenge for students. Moreover, while there are several ways to access a list of required courses for the SoSy degree, the process often entails extensive navigation. Addressing these issues, SoSy Scheduler enables planning across multiple semesters with the reference of degree checklists all in one place.

Customer

Name: Joshua Li

Role: President of Software Systems Student Society (SSSS)

SSSS is a student-led society at Simon Fraser University, where it helps students and faculty by addressing issues and concerns between the students and faculty, building community in and around our society, and sharing resources provided by students and external organizations.

Email: joshua_li@sfu.ca

User Stories

Type of User: Student User (SFU student in Software Systems Major)

Story: A regular student wants to log in.

Iteration: 1

Subtype: Login

Triggers/PreConditions: From the root page, press on a login.

Actions/Postconditions: If successful login, take to their landing page. If unsuccessful login, return and notify the user.

Tests: Program tests whether valid login or not and handles either case.

Type of User: Student User (SFU student in Software Systems Major)

Story: A regular student wants to log out.

Iteration: 2

Subtype: Logout

Triggers/PreConditions: The user must have had a successful login (see story above). The user then clicked on the logout.

Actions/Postconditions: The user's session is successfully terminated. The page is then rerouted back to the index.

Tests: As a precondition, the program must test that it is in a valid session. As a postcondition, the program must test that it logged out properly.

Type of User: Student User (SFU student in Software Systems Major)

Story: A regular student wants to view a list of completed and uncompleted SOSY courses specified as such.

Iteration: 2

Subtype: Course Viewing

Triggers/PreConditions: The user must be logged in.

Actions/Postconditions: The user has a list of completed and uncompleted SOSY courses that they can update as complete and incomplete (see stories below).

Tests: As a precondition, the program must test that it is in a valid session. As an invariant condition, the program must test that the completed and uncompleted SOSY list make up the complete SOSY list.

Type of User: Student User (SFU student in Software Systems Major)

Story: A regular student wants to mark a class as complete.

Iteration: 2

Subtype: Course Viewing

Triggers/PreConditions: The user must have had a successful login (see story above). The user then clicked on an incomplete class from their course list.

Actions/Postconditions: The user's account is updated with the newly completed course. The view of completed and uncompleted SOSY courses is reloaded (see story above).

Tests: As a precondition, the program must test that it is in a valid session. As an invariant condition, the program must test that the completed and uncompleted SOSY list make up the complete SOSY list.

Type of User: Student User (SFU student in Software Systems Major)

Story: A regular student wants to mark a class as incomplete.

Iteration: 2
Subtype: Course Viewing
Triggers/PreConditions: The user must have had a successful login (see story above). The user then clicked on a complete class from their course list.
Actions/Postconditions: The user's account is updated removing the completed course. The view of completed and uncompleted SOSY courses is reloaded (see story above).
Tests: As a precondition, the program must test that it is in a valid session. As an invariant condition, the program must test that the completed and uncompleted SOSY list make up the complete SOSY list.

Type of User: Student User (SFU student in Software Systems Major)
Story: A regular student wants to register a new account.
Iteration: 2
Subtype: Registration
Triggers/PreConditions: From the root page, click on registration.
Actions/Postconditions: User inputs unique email and passwords and submits. If successfully created, notify and reroute to login (see story above). If not successfully created, route back to the registration page.
Tests: A registering user is checked to have a valid username ending with @sfu.ca and a password that includes an uppercase, lowercase, and a number. Program handles all cases of incorrect username or password as an unsuccessful case and notifies the user.

Type of User: Administrator
Story: An administrator wants to log into their account
Iteration: 1
Subtype: Login.
Triggers/PreConditions: From the root page, press on login.
Actions/Postconditions: If successful login, take to their landing page. If unsuccessful login, return and notify the user.
Tests: Program tests whether valid login or not and handles either case.

Type of User: Administrator
Story: A regular student wants to log out.
Iteration: 2
Subtype: Logout
Triggers/PreConditions: The user must have had a successful login (see story above). The user then clicked on the logout.
Actions/Postconditions: The user's session is successfully terminated. The page is then rerouted back to the index.

Tests: As a precondition, the program must test that it is in a valid session. As a postcondition, the program must test that it logged out properly.

Type of User: Administrator

Story: An administrator wants to see the current list of users in the database.
Iteration: 2
Subtype: Administration
Triggers/PreConditions: The user must have had a successful login (see story above).
Actions/Postconditions: The user is provided with a list of current users in the system.
Tests: As a precondition, the program must test that it is in a valid administration session.

Type of User: Administrator

Story: An administrator wants to remove a user.
Iteration: 2
Subtype: Administration
Triggers/PreConditions: The user must have had a successful login (see story above). The user then clicked on a user from the user list.
Actions/Postconditions: The database is updated removing the user. The view of users is reloaded (see story above).
Tests: As a precondition, the program must test that it is in a valid session. Upon receiving a user to remove, the program tests that it is a current valid user and handles the case when the administrator is selected. The program must ensure that the administrator is not deleted.

Iteration 2 Major Features

Flowing over from iteration 1, the login/logout/register feature is now complete with session verification. The admin account has the username adminHere@sfu.com and the password Hello 123. Please create your new account with a random sfu.com email to test internal functionality.

For the administrator, a current list of users is dynamically provided. From this list, the administrator can click on a user to remove them from the database. The administrator themselves cannot be removed.

For the users, a current list of all SOSY courses completed and uncompleted is provided. From this list, users can click on an uncompleted course to log it as completed. Additionally from this list, users can click on completed courses to log them as uncompleted. This

information is saved and carries over from session to session. All SOSY courses are either completed or uncompleted.

Iteration 2 Velocity

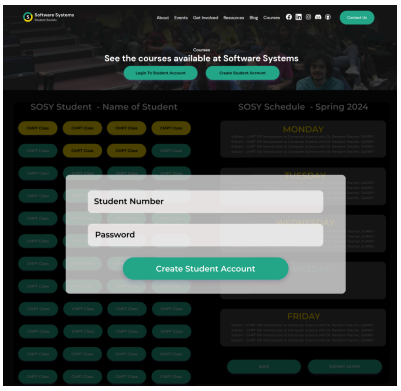
The velocity for Iteration 2 sped up quite considerably from Iteration 1 as there are now 8 additional story points completed in this iteration all of which have set up the back-end data needed to create the front-end scheduling view for Iteration 3. The velocity for Iteration 3 is estimated to be quite quicker since it requires less information from the database. The velocity of the development increased as the team members became more comfortable working on different parts of the project simultaneously and the bottleneck of setting up the system was cleared.

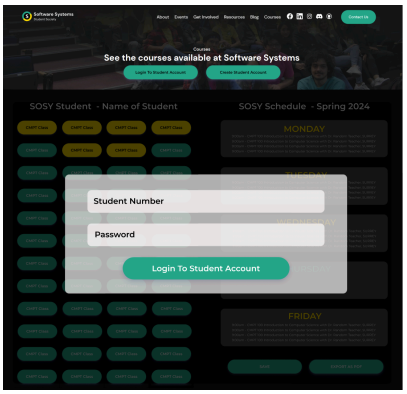
Development did briefly slow when the Render database and site thus delaying pushes to git. Team Members instead developed functionalities on local devices using locally saved data before pushing to git after Render.com recovered.

User Interface Requirements

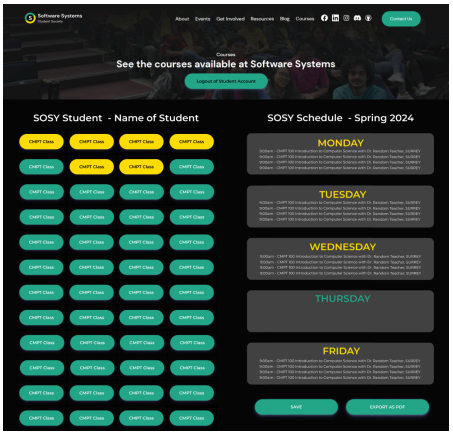
The User Interface must have all of the following views/methods on the site.

1. A method to log in and/or create a new user. In the implementation, this method can be accessed via two buttons on the main page of the site.

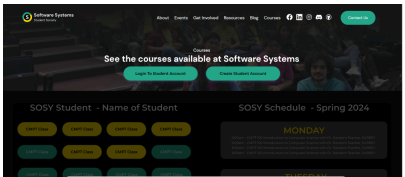


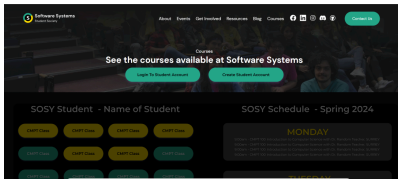


2. A view to see a complete list of courses completed and to complete a Software Systems degree. In this implementation, a list of courses as individual buttons is displayed on the main page with completed courses in yellow and uncompleted courses in green.



3. A view to see classes scheduled over multiple different academic calendar terms. In this implementation, a weekly schedule shows selected classes with timings for a certain term.





4. A method to save and/or export the created schedule. In the implementation, a button is used to generate either result from the back-end

5. A method to view and select from all lecture and lab offerings of a given course for a given term. In the implementation, users clicking on a course (2) while in a given term (3) will result in a list of lecture offerings followed by lab offerings in that semester.