StudyScapes
7/9/20
Version 003

Requirements and Specification Document

Project Abstract

This application, **StudyScapes**, will allow users to **login** to their account and view the Simon Fraser University campus as a virtual map. Login **views** will differ for faculty and students (and admins). This uses the **Web API for Google Maps** for location and time updates. As a **real-time feature**, this application will use **Socket.io** to play real-time multiplayer games. This application enables quick meetups and easy navigation for users. The application is not just practical, but fun! The incorporated minigames help the user kill time while waiting between meetups or interact with other students (they can even battle their professors!).

Customers

Faculty: Teachers and TAs at Simon Fraser University that would like to interact in a convenient way with students, plan meetings, coordinate meeting locations, schedule or find events on campus, and enjoy minigames.

Students: Students at Simon Fraser University that would like to communicate with all their professors or TAs in one convenient application. They would like to meet with other students or faculty at SFU, get assistance in finding their way around the school, find specific buildings or rooms, schedule or find events on campus, meet other students, or just have fun through minigames.

Competitive Analysis

What makes StudyScapes so unique is that it applies specifically to the Simon Fraser University Burnaby Campus. It incorporates ideas from many other successful programs into one main application for ease of access. Although Google Maps is useful, it does not provide a detailed view of the Burnaby Campus. StudyScapes will elaborate on the map with this campus in mind, allowing for much more extensive interaction and location finding around the University. SFU Snap was an inspiration for this, as it helped students to search for and travel between rooms and buildings across campus. StudyScapes expands this to allow for scheduling events and meetings around campus with students and faculty alike. It also provides icebreakers for students in the form of minigames and different views for faculty and students, so that each user sees the information that directly applies to their own needs.

StudyScapes
7/9/20
Version 003

Requirements and Specification Document

Main Features - Epics

Map: Provide an interactive map that updates to user location so they can find other users and locations on the Simon Fraser University Campus.

Meetings: Enable faculty and students to easily interact and meet up for studying or course/career discussion. They can also chat with each other through the app.

Events: Have events held by clubs, faculty or student societies show on the map as icons in real-time, so users can quickly learn about the activity on campus.

Minigames: Allow students and faculty to interact in mini games with those nearby based on location. This would allow users to play small various games that can act as icebreakers between their fellow peers.

User Stories

Actors:

Faculty (Professors and Teaching Assistants) that can view and cancel meetings. They can also interact with students, view the map, view and schedule events, and play minigames. Students can view, request, reschedule, and cancel meetings with Faculty and other students. They can view and schedule events, view the map for specific rooms and buildings, and play minigames.

A test login for faculty is username: QWERTY and password: pass2

Students can view, request, reschedule, and cancel meetings with Faculty and other students. They can view and schedule events, view the map for specific rooms and buildings, and play minigames.

A test login for student is username: ASDF and password: pass1

Admin can view all database content including usernames, roles, meetings, and much more. The admin account is used for verification and debugging purposes. For the sake of security and privacy of users, the admin can only view hashed passwords.

A test login for admin is username: ADMIN and password: ADMIN

Requirements and Specification Document

Current Iteration Stories

Story #21

Name/Description: A student wants to meet with a professor

Actors: Student 'Maddie' and Professor 'Bud'

Triggers/Preconditions: Maddie accesses the meetup page and schedules a meeting with Bud

Actions/Postconditions: Displays a form for Maddie to specify the meeting users, date, location and time, then sends a request to Bud

Acceptance Tests:

- Meetup form is displayed properly
- Bud is an existing faculty member and a request can be sent to him
- Meeting location is valid and can be chosen
- Form can be submitted
- Meeting data is stored in the meetup table

Iteration: *completed in iteration 3*

Story #10

Name/Description: Viewing upcoming events on campus.

Actors: A user (student/faculty)

Triggers/Preconditions: User clicks the "View Upcoming Events" on a page.

Actions/Postconditions: Shows a page with a table where the rows are Events showing its name, host group, website link, location, time, and date in chronological order.

Acceptance Tests:

- An Events table is displayed with information described in postconditions.
- Events are listed in chronological order (closest to farthest from current time).
- Events prior to current time are not shown.
- Cancelled events have colored text (red) and shows "CANCELLED" under "time" ("CANCELLED" should be reflected in the database under "time" also).

Iteration: to be implemented in *iteration 002, with additions in iteration 3*

Requirements and Specification Document

Story #22

Name/Description: View error page when a user has no meetups.

Actors: A student named Huck

Triggers/Preconditions: Huck clicks the meetups page, but doesn't have any meetups

Actions/Postconditions: Redirects Huck to an error page, with links to request a meeting or return to the home page.

Acceptance Tests:

- An error page is displayed
- The back button is shown
- The request meeting button is shown

Iteration: *additions completed in iteration 3*

Story #14

Name/Description: A new student is looking to make new friends at SFU, wants to play a game.

Actors: A student, Sophia, wants to play a game with someone.

Triggers/Preconditions: Student Justin clicks Minigames tab on their page.

Actions/Postconditions: Shows different types of minigames and allows them to play with people nearby.

Acceptance Tests:

- Able to connect with other players
- Game logic is correct
- Able to choose what type of game to play
- Able to correctly interact with game buttons.

Iteration: to be implemented in *iteration 003*

Story #15

Name/Description: Viewing events happening today on the map.

Actors: A user (student/faculty)

Triggers/Preconditions: The user goes to the campus map page (not viewing room finder)

Actions/Postconditions: The map loads showing SFU campus. Events listed to happen today are shown with event markers at their respective locations.

Acceptance Tests:

- The map is rendered correctly, showing SFU campus.
- Each event marker represents an event happening *today*.
- Each event marker is shown at their correct locations.
- Clicking on a marker displays detailed information regarding the event.

Iteration: to be implemented in *iteration 003*

Story #16

Name/Description: Hiding a type of marker on the map.

Actors: A user (student/faculty)

Triggers/Preconditions: The user clicks on any of the box from a group of checkbox options to for hiding a type of marker on the map (i.e. hide location markers, hide event markers, etc.).

Actions/Postconditions: The map is re-rendered with the desired markers to be hidden gone.

Acceptance Tests:

- The map is re-rendered with the same position and zoom level as prior to triggering the event.
- The desired markers to be hidden are not seen while the others are still shown.
- The hidden markers cannot be interacted with.

Iteration: to be implemented in *iteration 003*

Story #17

Name/Description: A student has a 5-hour break between two classes, and he/she wants to pass some time while being on the campus

Actors: Stacey, a fourth-year student

Triggers/Preconditions: Stacey logs in to StudyScapes and opens the mini games

Actions/Postconditions: Display different options for mini games

Acceptance Tests:

- Game services are enabled

- Scores can be found and display

Iteration: to be implemented in *iteration 003*

Story #18

Name/Description: A new student wants to meet other students with similar interest on the Burnaby campus

Actors: New student named 'Patrick'

Triggers/Preconditions: Patrick logs in to StudyScapes and opens the events page

Actions/Postconditions: After finding a suitable event he can now meet students who share same interests as him

Acceptance Tests:

- Different events are displayed to the users
- Students can drop into their desirable event

Iteration: *completed in iteration 3*

Story #19

Name/Description: Viewing the user's schedule timetable in the Dashboard

Actors: A user (student/faculty)

Triggers/Preconditions: User logs into StudyScapes or goes to Dashboard page

Actions/Postconditions: The Dashboard is shown with a schedule timetable on the page. The schedule visually represents a weekly calendar, displaying meetings that the user has committed to participate in along with events.

Acceptance Tests:

- The schedule shows 7 days ahead including the current day.
- A schedule timetable is displayed with participating meetings at their correct positions.
- Canceled meetings are not shown on timetable.

Iteration: to be implemented in *iteration 003*

Story #20

Name/Description: A student is looking for their friend on campus but is unsure where they are.

Requirements and Specification Document

Actors: New student named “Adam”

Triggers/Preconditions: Adam logs in to StudyScapes and opens the View My Location page

Actions/Postconditions: After opening the View My Location page, Adam is able to see buildings around campus and can easily locate his friend in order to meet up with him.

Acceptance Tests:

- Able to see important SFU buildings on the map.
- Able to view the building acronyms on each marker on the map.
- Location updates as the user moves.

Iteration: completed in *iteration 003*

Story #23

Name/Description: A faculty member wants to check for unconfirmed meetings

Actors: A professor names “Marshal”

Triggers/Preconditions: Marshal sees a pending meeting on his meetups page and clicks to confirm the meeting

Actions/Postconditions: The meeting is confirmed and this change is also updated for the other users in the meeting

Acceptance Tests:

- The meetups table can be seen, with just meetings corresponding to Marshal
- Marshal is able to type a meeting ID into the input box
- The confirm button calls a query to update the corresponding row
- is_pending is changed to false for the chosen meeting

Iteration: completed in *iteration 003*

Story #24

Name/Description: A faculty member wants to remove a cancelled meetup

Actors: A professor names “Anabelle”

Requirements and Specification Document

Triggers/Preconditions: Anabelle views her meetups and enters the meetup ID for a cancelled meetup

Actions/Postconditions: The meeting is removed from her meetup page and this change is also updated for the other users in the meeting

Acceptance Tests:

- The meetups table can be seen, with just meetings corresponding to Anabelle
- Anabelle is able to type a meeting ID into the input box
- The remove button calls a query to remove the corresponding row
- the meetup is completely removed from the database

Iteration: completed in *iteration 003*

Story #25

Name/Description: A student incorrectly fills out a meetup request form

Actors: A student 'Audie'

Triggers/Preconditions: Audie accesses the meetup page and incorrectly fills out a form to meet with a professor, by forgetting to including any users

Actions/Postconditions: The form redirects to an error page and the entry is ignored

Acceptance Tests:

- Meetup form is displayed properly
- The program does not crash
- An error page is loaded
- The error page has a link to return to the meetup request page

Iteration: *completed in iteration 3*

Story #26

Name/Description: A student wants to meet with his professor but doesn't know her ID

Actors: Student 'Wade' and faculty member 'Tammy'

Triggers/Preconditions: Wade logs in to view a meetup request page

Actions/Postconditions: Wade can view the list of users and find his professor's ID

Acceptance Tests:

- Meetup form is displayed properly
- A database table is displayed, showing all users and their roles
- The table is ordered by user role

Iteration: *completed in iteration 3*

Story #27

Name/Description: A faculty member tries to remove a meetup, but choses a non-cancelled one

Actors: Faculty member 'Frita'

Triggers/Preconditions: Frita logs in to view her meetups and tries to remove one that hasn't been cancelled yet

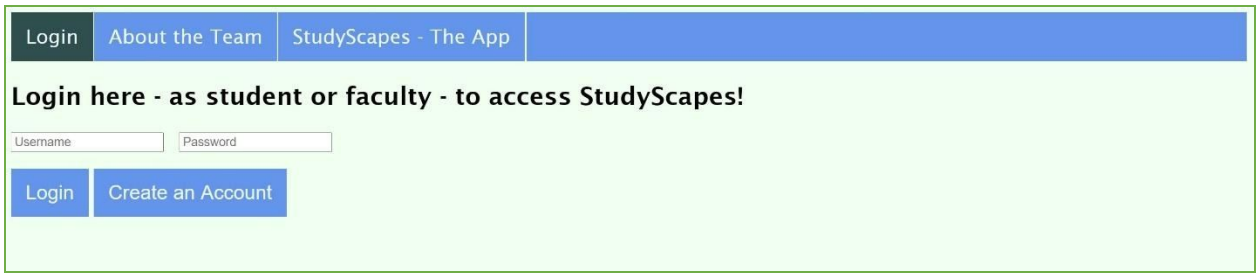
Actions/Postconditions: The meetup is not removed and Frita is redirected to an error page

Acceptance Tests:

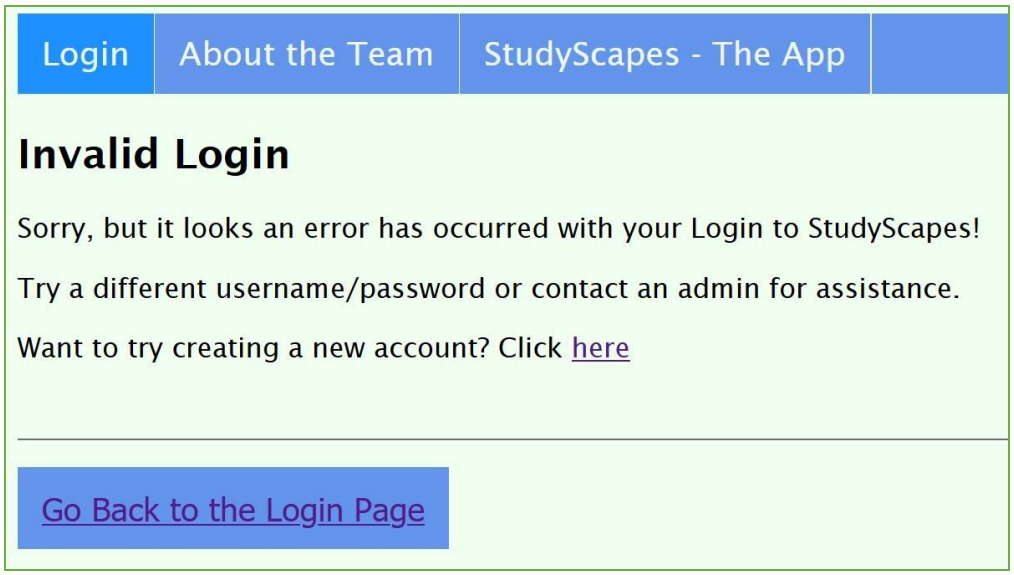
- Meetup table is displayed properly
- Frita is able to enter a meetup ID
- The meetup is correctly determined to be still active and not yet cancelled
- Frita is redirected to an error page
- The meetup is not removed from the database

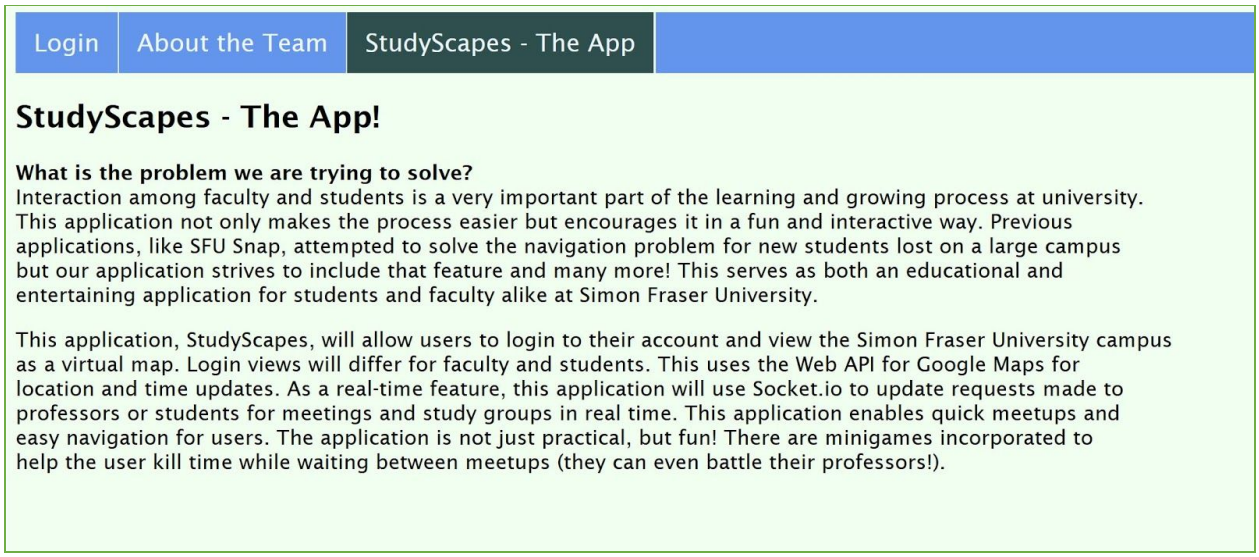
Iteration: *completed in iteration 3*

User Interface Requirements

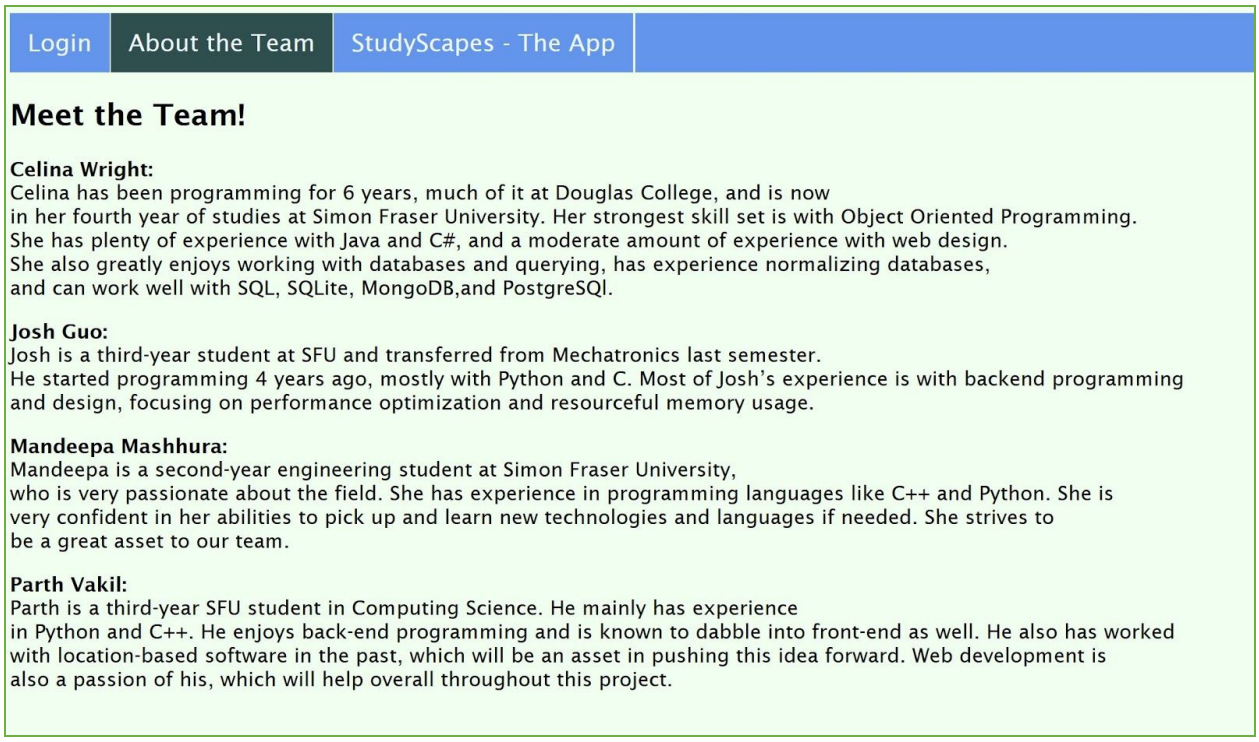


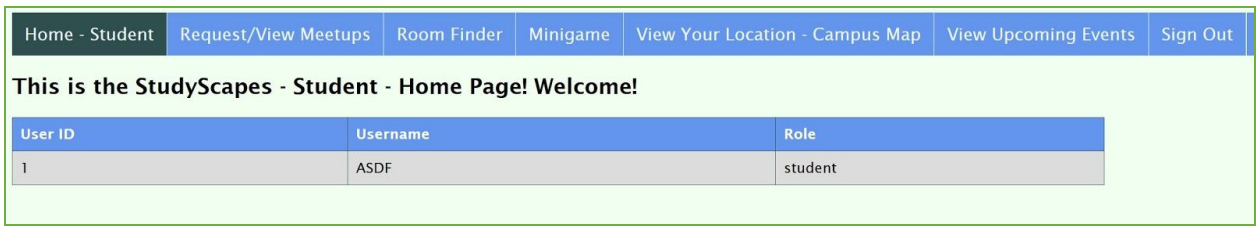
Requirements and Specification Document



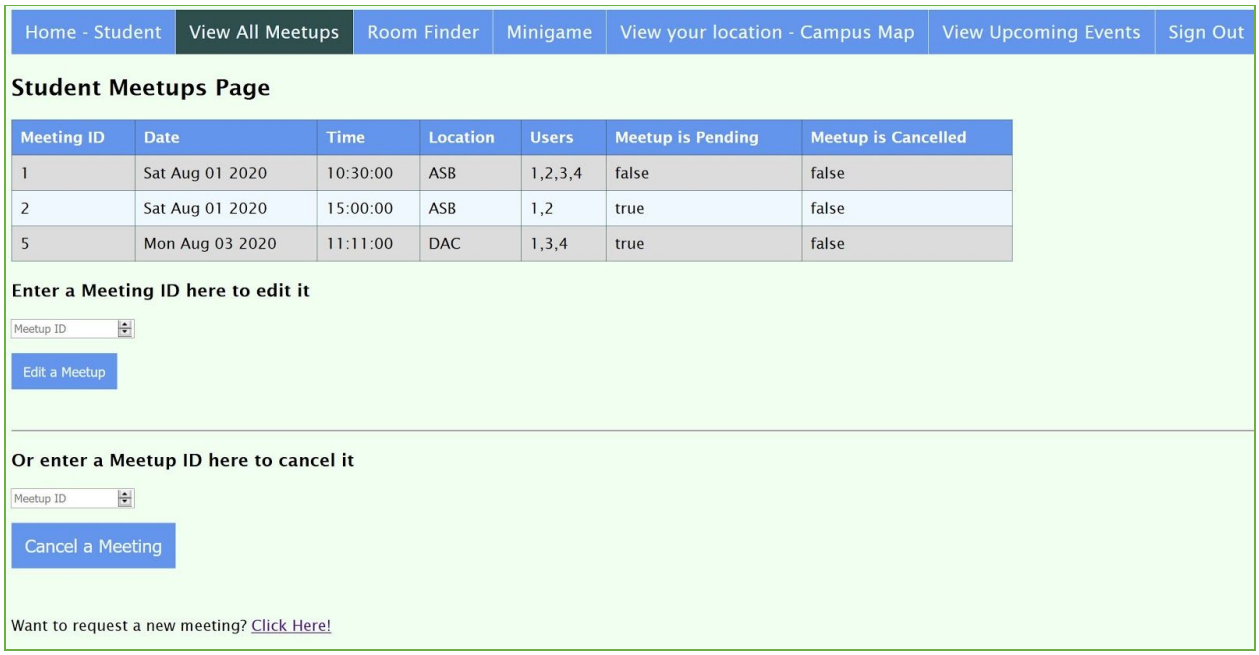


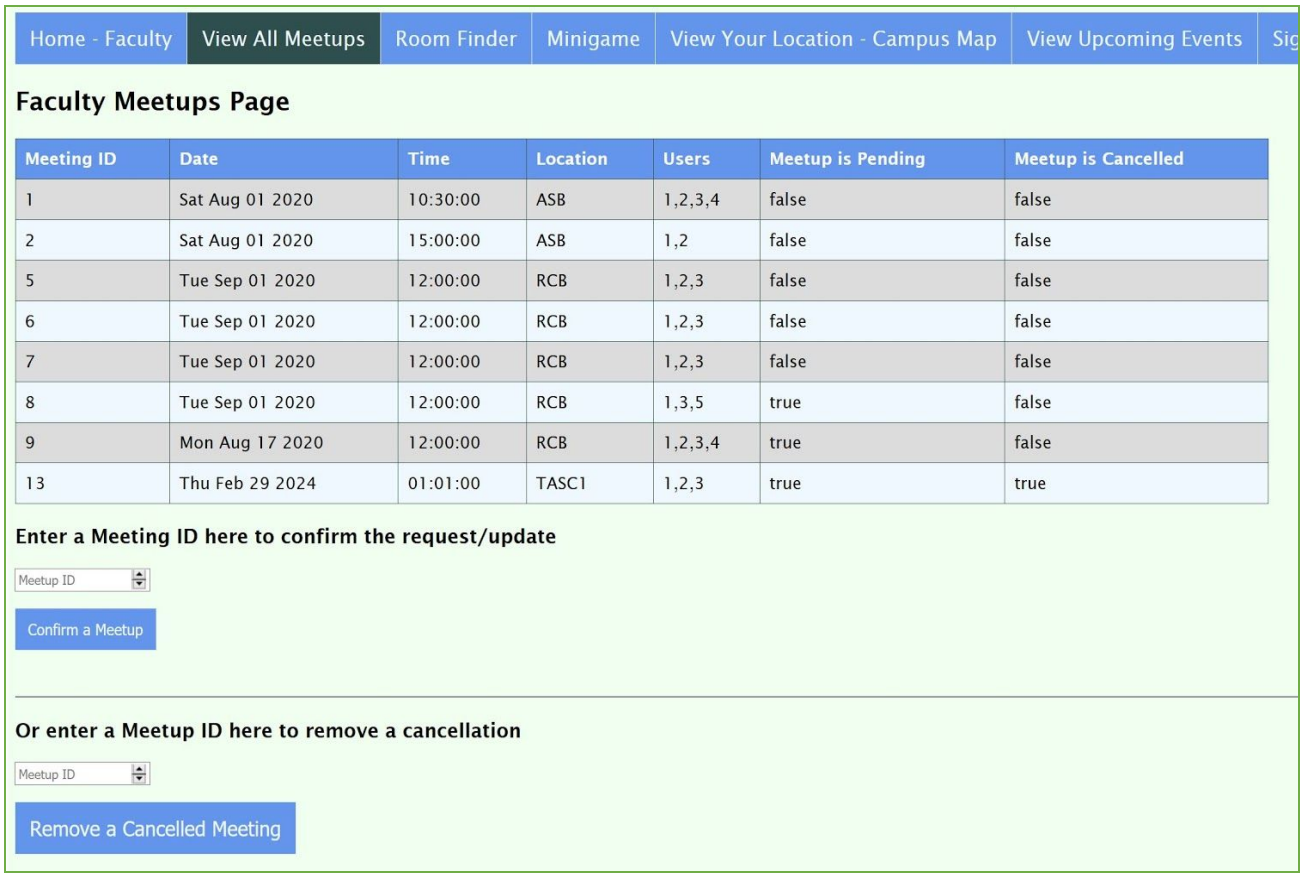
Requirements and Specification Document



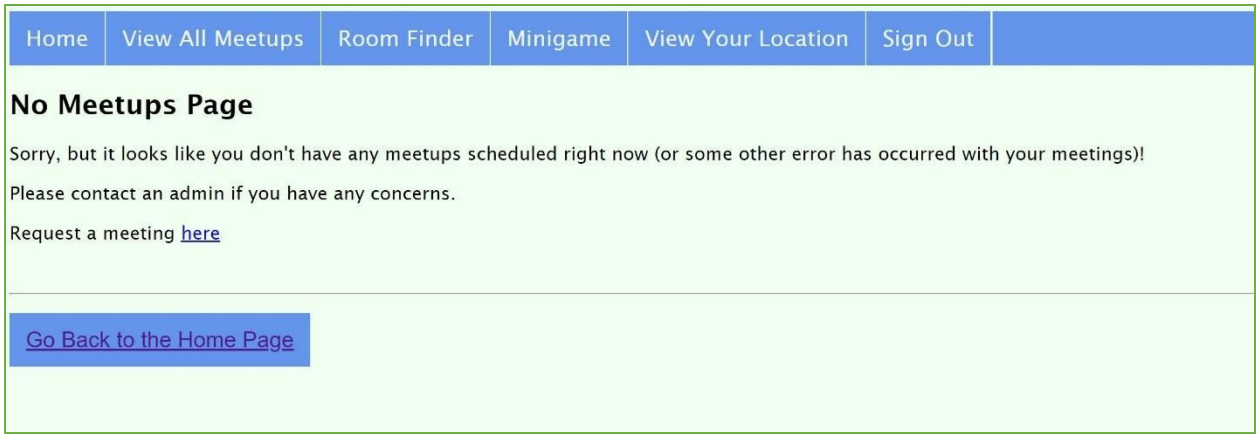


Requirements and Specification Document

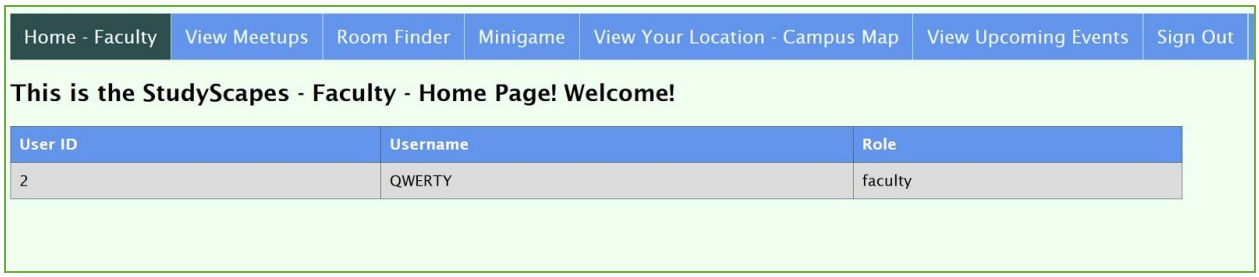


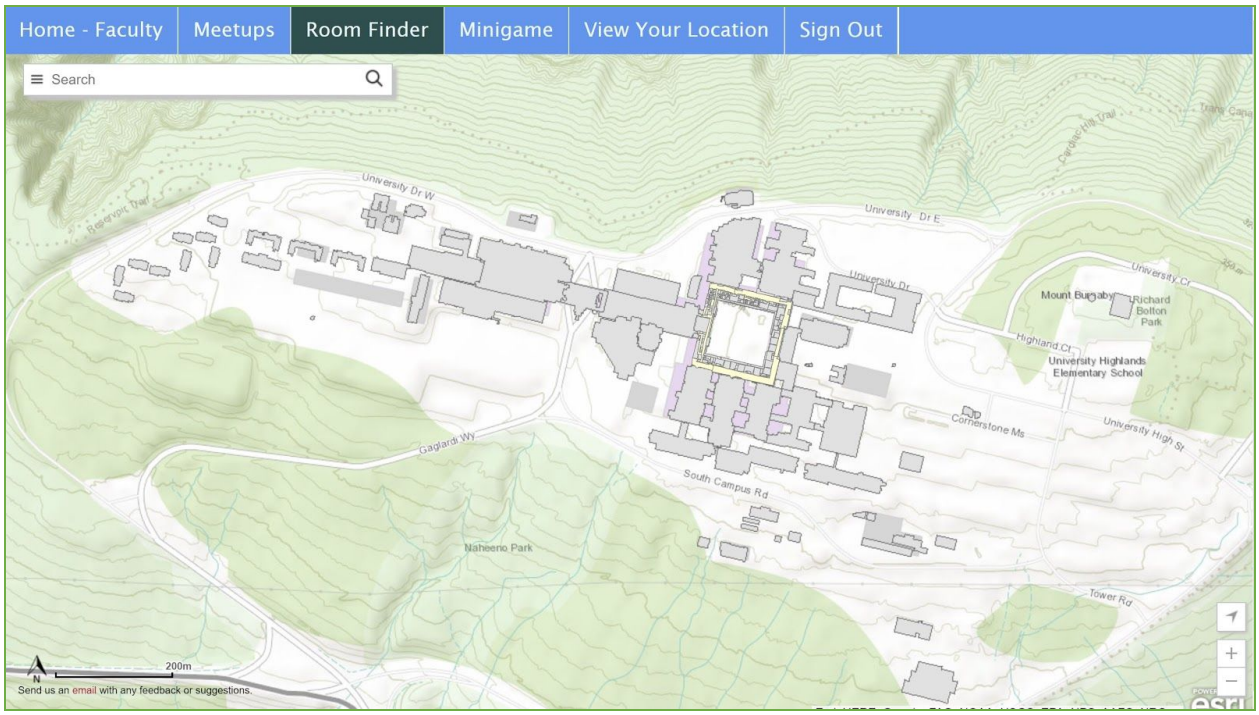


Requirements and Specification Document

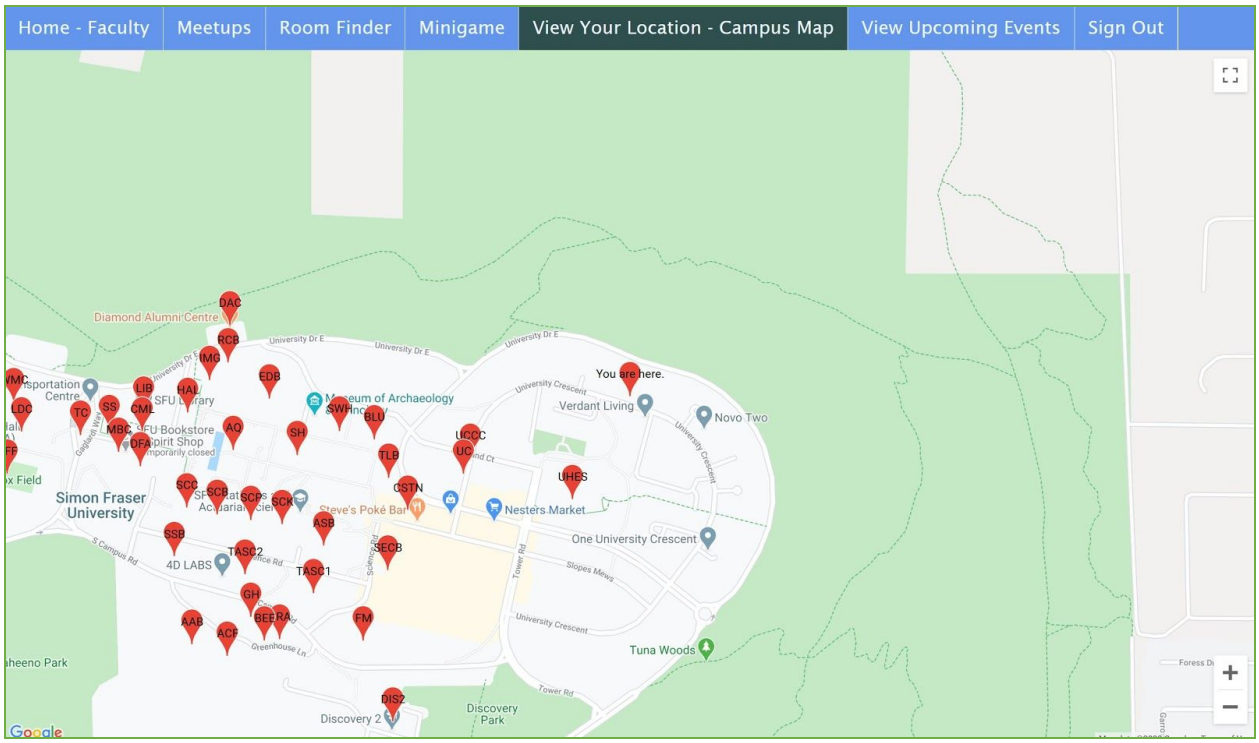


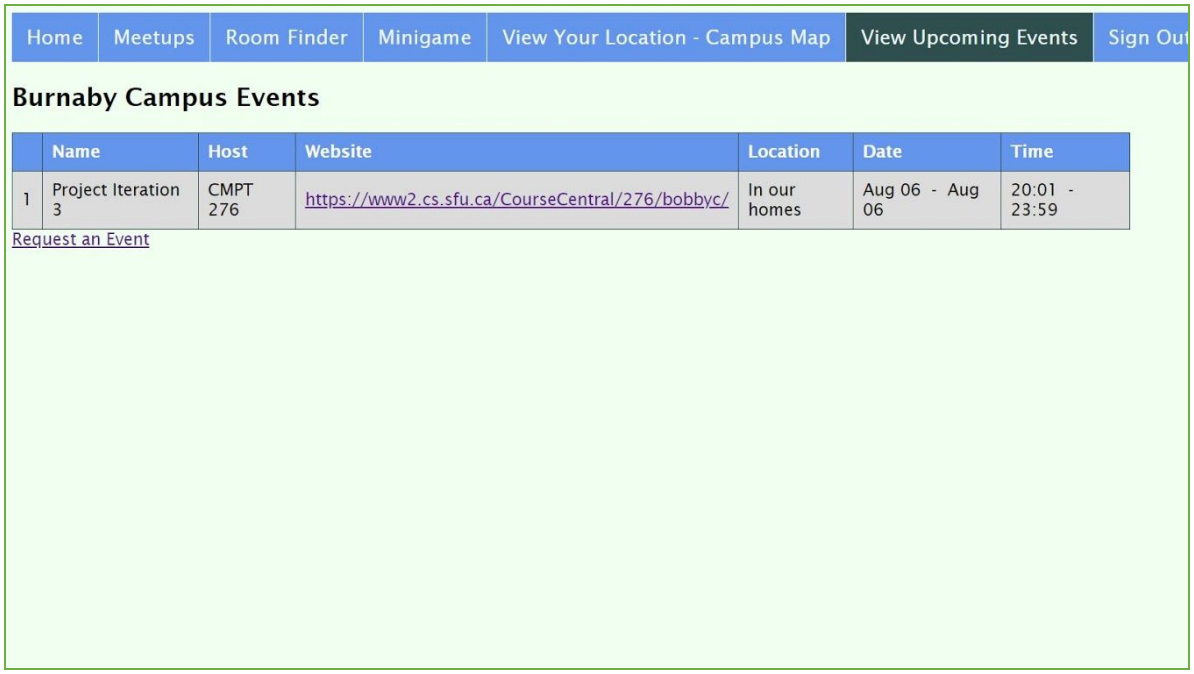
Requirements and Specification Document



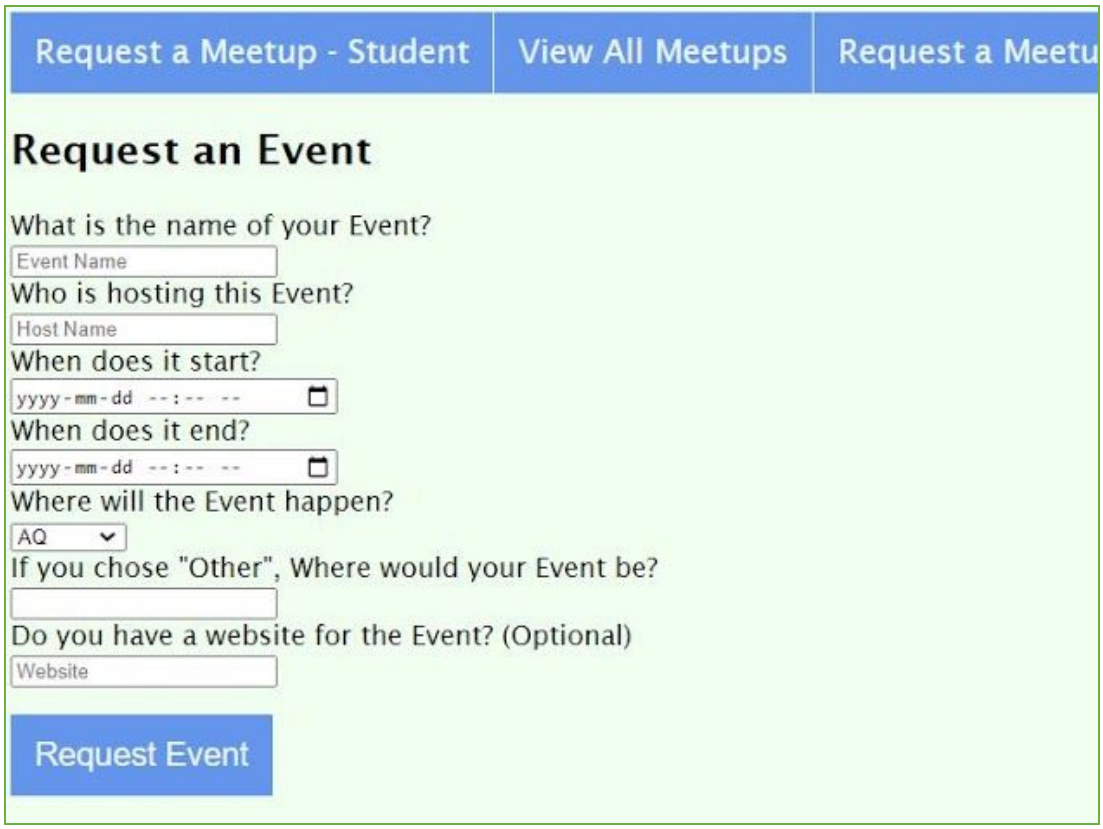


Requirements and Specification Document





Requirements and Specification Document



Requirements and Specification Document

View All User Logins | View All Meetups | View Upcoming Events | Sign Out

Admin Home Page

User ID	Username	Password - Should be Hashed!	Role
1	ASDF	$2b$05$JWgDYqbwDxjAJTZlKOMXsu7L4HwHlrXQAxpi6Ol2hogvbavhZo.Me	student
2	QWERTY	$2b$05$B/UDY1YPGJqhVx7m.N9DyOCHZfri6XdNgt9B9wVmlPg1PlqfJFAaS	faculty
3	sob	$2b$05$IsT.57uJjxOzmUb.KLW7I.09Kcr.63xoiZTCG00b/QqHsveLj0Ngu	student
4	parth	$2b$05$cRL/pJUSeVeDZYwfTEEBA.9d0h7jk0hulp4cu26Fe3t1SB8KmALCq	student
5	parth-faculty	$2b$05$flJqI0KUKFF.P5G0TnvWzetT4ULZHjOfrLnfyHDolusVyaYNzLrSq	faculty
8	vbatvia	$2b$05$uqkpSt5OB49WSJxCx3vAK.c.rtSlZHlVyuWdxjZzpMD9HXz.wgMqm	student
9	hamid	$2b$05$ECOJ4ZRjXTOM6fJ2pkHprep/mWHatvstLZwY5dizPO2g97no.sJgG	student
10	students	$2b$05$W9gpW.kfI1xpFIQtAGvfGe.OOH0chf/Q2yeeUr0Hz1MfSn7kxogky	student
11	bc	$2b$05$iRTEJgzEAuPlQyMQrL53wuOsRbFTJ3pUBE5NLS4PT.P/hDg/z5TpC	faculty
12	Phyberosis	$2b$05$fsFth/VGrWEgxNkuiT/jB.pT1cWZbq3TNyzdIqkIIdYDlCID1Wgte	faculty
14	v	$2b$05$xDS8XEFiiLYt6S4D5JCLseO8juALVpntj0m70kWdAZZ4VVS.SQvLq	faculty
20	ADMIN	$2b$05$BUjm.KDO8f3IhYg0U2ng.O.RkeOWW3nI6b3DtMAtZ4ZJALdcsILni	admin

View All User Logins | View All Meetups | View Upcoming Events | Sign Out

Admin Meetup Page

Meeting ID	Date	Time	Location	Users	Meetup is Pending	Meetup is Cancelled
1	Sat Aug 01 2020	10:30:00	ASB	1,2,3,4	false	false
2	Sat Aug 01 2020	15:00:00	ASB	1,2	false	false
3	Sun Aug 02 2020	09:30:00	ASB	3,4	false	true
4	Sun Aug 02 2020	10:30:00	AQ	15,14	false	false
5	Tue Sep 01 2020	12:00:00	RCB	1,2,3	false	false
6	Tue Sep 01 2020	12:00:00	RCB	1,2,3	false	false
7	Tue Sep 01 2020	12:00:00	RCB	1,2,3	false	false
8	Tue Sep 01 2020	12:00:00	RCB	1,3,5	true	false
9	Mon Aug 17 2020	12:00:00	RCB	1,2,3,4	true	false
13	Thu Feb 29 2024	01:01:00	TASC1	1,2,3	true	true

Requirements and Specification Document

Velocity Measurement

Story Points		
Story Number	**Points**	**Iteration**
1	2	Iteration 1
2	3	Iteration 1
3	3	Iteration 1
4	3	Iteration 2
5	2	Iteration 2
6	2	Iteration 2
7	2	Iteration 2
8	2	Iteration 2
9	1	Iteration 2
10	2	Iteration 2
11	1	Iteration 2
12	3	Iteration 2
13	1	Iteration 2
14	5	Iteration 3
15	2	Iteration 3
16	4	Iteration 3
17	2	Iteration 3
18	1	Iteration 3
19	6	Iteration 3
20	5	Iteration 3
21	3	Iteration 3
22	2	Iteration 3
23	2	Iteration 3
24	2	Iteration 3
25	1	Iteration 3
26	1	Iteration 3
27	2	Iteration 3
28		
29		
30		
31		

Velocity

Iteration 1 points:
8 in 2 weeks
= 4 points on average

Iteration 2 points:
19 in 2 weeks
= 9.5 point on average

Iteration 3 points:

Requirements and Specification Document

The Database Plan– Iterations 2 & 3

StudyScapes Database					
login		**meetup**		**event**	
uid	serial (PK)	mid	serial (PK)	eid	serial (PK)
username	char(20)	users	int[]	name	varchar
password	char(60)	date	date	host_name	varchar
role	char(20)	time	time	url	varchar
		location	varchar	location	varchar
		is_pending	boolean	start_datetime	timestamp
		is_cancelled	boolean	end_datetime	timestamp

Disclaimers

The Room Finder Tab was not created by our team. This was an application created by SFU that we have integrated into our project in order to make it more whole. We obtained permission from Professor Chan to include this in our application.

Resources

These are works that were referenced in the creation of this project

1.

URLs

GitHub Repository: *https://github.com/Guojiaxi/sfu-cmpt276proj.git*

Heroku link: *https://cmpt276proj-jlguo.herokuapp.com/*

Heroku Git Link: *https://git.heroku.com/cmpt276proj-jlguo.gi*

Features Board

COMPLETED	IN PROGRESS	TO BE IMPLEMENTED	PREREQUSITE(S) NEEDED	TO BE DESIGNED	BROKEN /REDESIGN

StudyScapes
7/9/20
Version 003

Requirements and Specification Document

map	Place map markers	Event markers	Plotted meetings (schedule timetable)	Meeting notifying system	Chat
Log in and sign up system	User location tracking	Dashboard schedule timetable	Plotted participating events (schedule timetable)	Event notifying system	
Login database		meeting invitation/response		Meeting creation UI	
Logged-in tracking		minigames		Event posting UI	
Sign-up system				Event submission approval system	
Room finder				User profile page	
Server-side storage for events (database)					
Server-side storage for meetings (database)					

Features to have done tomorrow:

- Tic Tac Toe Minigame with socket.io
- UI and Sidebar for the Find My Location

Requirements and Specification Document

- Add events to map
- Finish Meetups

Demo deadline: Saturday!