

SW Engineering - CSC648/848

Spring 2020



Team 03

Milestone 5

Lionel Wong	lwong7@mail.sfsu.edu	Team Lead
Jorge Landaverde	jlandaverde@mail.sfsu.edu	Lead Backend Engineer
Gordon Lam	glam1@mail.sfsu.edu	Backend Engineer
Mitul Savani	msavani@mail.sfsu.edu	Lead Frontend Engineer/Git Master
Kevin Huang	khuang2@mail.sfsu.edu	Frontend Engineer
Jack Kower	jkower@mail.sfsu.edu	Frontend Engineer

Milestone	Date
Milestone05 Version01	5/21/2020
Milestone04 Version02	5/20/2020
Milestone04 Version01	5/10/2020
Milestone03 Version02	5/8/2020
Milestone03 Version01	4/23/2020
Milestone02 Version02	4/15/2020
Milestone02 Version01	3/25/2020
Milestone01 Version02	3/12/2020
Milestone01 Version01	3/5/2020

Table Of Contents

I.	Product Summary	2
II.	Milestone Documents	
	A. Milestone 1	4
	B. Milestone 2	28
	C. Milestone 3	58
	D. Milestone 4	82
III.	Screenshots of Final Product	96
IV.	Screenshots of Key DB Tables	99
V.	Screenshots of Task Management System	101
VI.	Team Member Contributions	102
VII.	Post Analysis	107

Product Summary



1. San Francisco State University Students and Faculty are able to register for Unigator accounts using their school emails.
2. Users are only able to log in with their school associated email. The email needs to be verified before logging into Unigator.
3. Users on Unigator are able to search for available events by name.
4. Users are able to search through past events.
5. Users can also filter search results by Categories.
6. Users can view all information about an event.
7. Registered Users can log in or log out of their account.
8. Registered Users are able to save(Star) any event they are interested
9. Registered Users are able to RSVP to any event.
10. Registered Users are able to submit a “Create Event” request and that will be sent to an Administrator for approval. Information such as description, date, time, location and images are to be provided when creating an event.
11. Registered Users are able to become an Event Host.
12. Registered Users have their own unique profiles, which displays the user’s information.
13. Registered Users can edit their profiles.
14. Registered Users are able to view other user’s profiles.
15. Registered Users are able to view an Event Host’s email.
16. Registered Users can access the point shop and purchase customizable options by spending their accumulated points,
17. Registered Users can earn points by interacting with events on Unigator.
18. Administrators are able to view requests.
19. Administrators are able to approve or deny requests.
20. Administrators are able to ban a user.

21. Administrators are able to delete an event.
22. Administrators are able to send requests for changes to an Event Host.
23. Administrators have all options unlocked for the point shop.
24. Event Host shall be able to edit all of their event's information.
25. Event Host shall be able to delete their event.
26. Event Host are able to grant RSVPed users points.

Unigator aims to be a unique platform for anyone connected to San Francisco State University by uniting everyone together to enjoy and share their ideals through events. Any events concerning San Francisco State University can be found on Unigator and created to promote social engagement with the school and its students. Unigator unites San Francisco State University peers through events and social engagements.

URL: <http://13.52.231.107:3006/home>

Milestone Documents

SW Engineering - CSC648/848

Spring 2020



Team 03

Milestone 1

Team Lead - Lionel Wong, lwong7@mail.sfsu.edu

Backend Lead - Jorge Landaverde

Backend Engineer - Gordon Lam

Front End Lead - Mitul Savani

Front End Engineer - Kevin Huang/Jack Kower

Git Master - Mitul Savani

Milestone	Date submitted
Milestone01 Version02	3/12/2020
Milestone01 Version01	3/5/2020

Table Of Contents

I.	Executive Summary	2
II.	Main Use Cases	3
III.	List of main Data Items and Entities	10
IV.	Initial List of Functional Requirements	12
V.	List of Non-Functional Requirements	15
VI.	Competitive Analysis	18
VII.	High-level system Architecture and Technologies Used	21
VIII.	Team	22
IX.	Checklist	23

Executive Summary

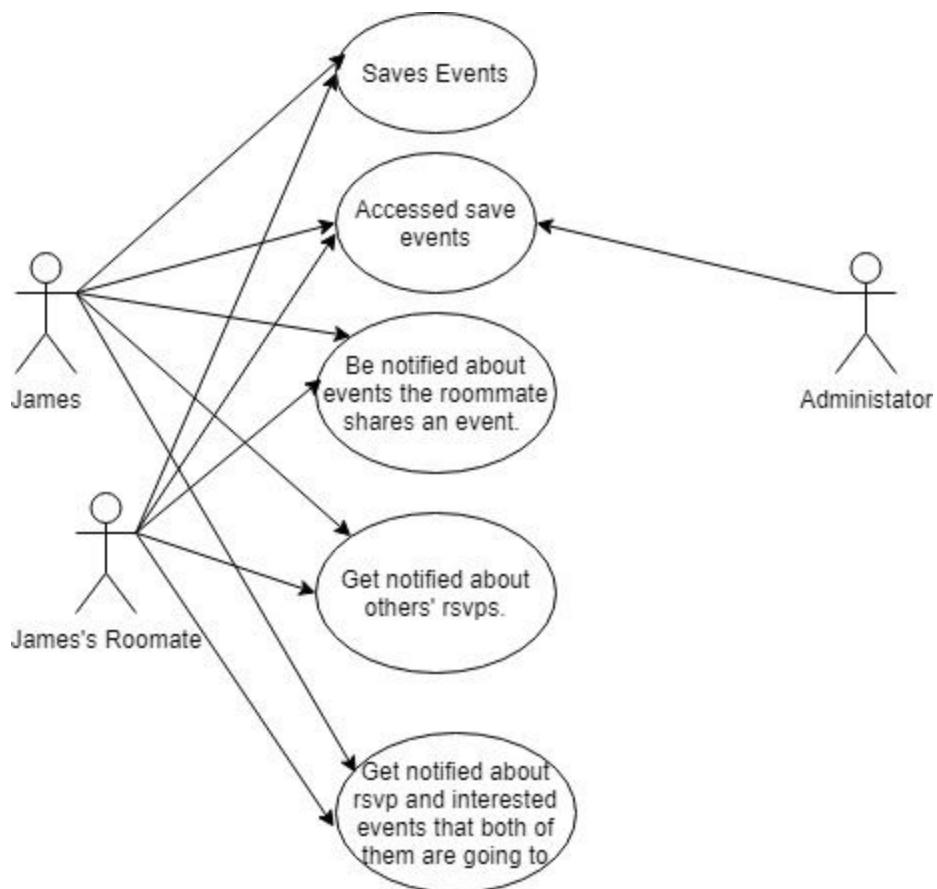
Unigator is a website dedicated to viewing and organizing San Francisco State University (SFSU) campus events. We aim to provide a platform for SFSU students to easily and clearly browse through university events in one central platform. There are numerous events in SFSU constantly ongoing everyday that don't get noticed and allow funding to be wasted. The current SFSU events' website is clearly lacking in terms of information and functionality. The events we would be displaying are not limited to certain elements, such as education or recreation. Events displayed cover all range of activities, any organization or students that wants to set up an event can use Unigator as a platform to advertise, manage, host or attend events. When anyone wants to know or look up an event at SFSU, Unigator is the place to go.

Unigator would take in all the ongoing events at SFSU, whether it's an event created by an organization or by a student. Unigator will prioritize displaying the information of the events in a simple and clear manner. Users would also be able to efficiently manage events and potentially invite or notify others about scheduled events. Unigator aims to push students to explore more and promote engagement. This would improve students' self esteem and decrease stress. The events are not tied down to sponsored organization, any scale of event can be posted. This will help the users to clearly identify which events they are interested in. Unigator will be one central platform that will provide all the essential information to make informed decisions. Organizations may sponsor events on the site, and SFSU may sponsor Unigator as it is a platform aimed towards SFSU students by expanding their horizons on culture and diversity. Unigator is aiming to unite students by allowing them to find new experiences to connect with other students.

Main Use Cases

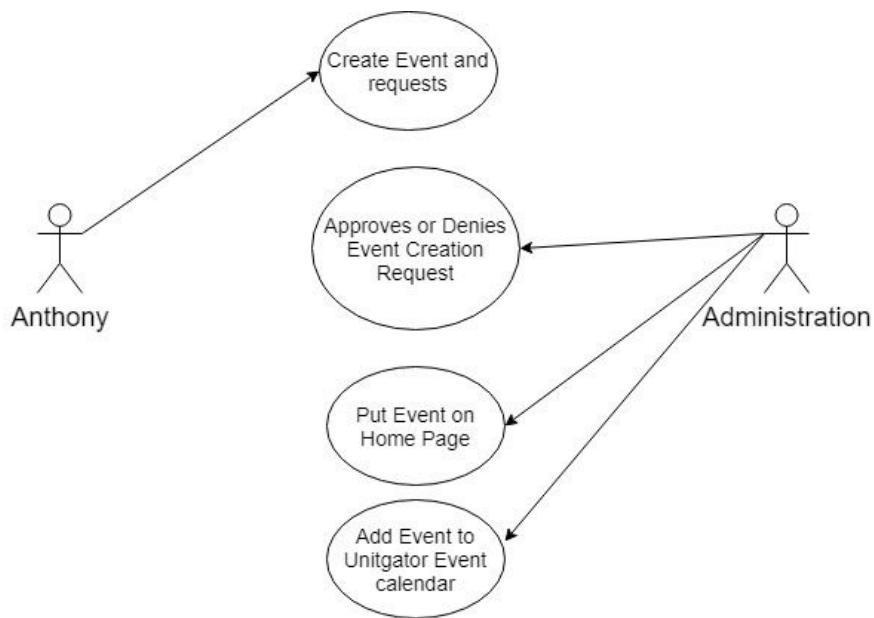
- James is a 22 year old college student. He is an environmental science major at San Francisco State University. He stays on campus with one other roommate, both James and his roommate are not from around San Francisco and are relatively new to the area and school. James only works part time as a cashier on campus which leaves him some free time. He wanted to explore and get to know things that are happening in the school. James is unsure though where to get info about events that are happening in the school as the school website is cluttered with other information. James's roommate recommended our online application "Unigator", which easily provides all the essential info and details of any event that are **currently planned or ongoing** in the school. James needs a way to let his roommate know of any events that he is interested in, and his roommate should be able to get **notifications on events** James has **shared with him**, so that they can both plan to attend the event.

Actors: James, Registered User



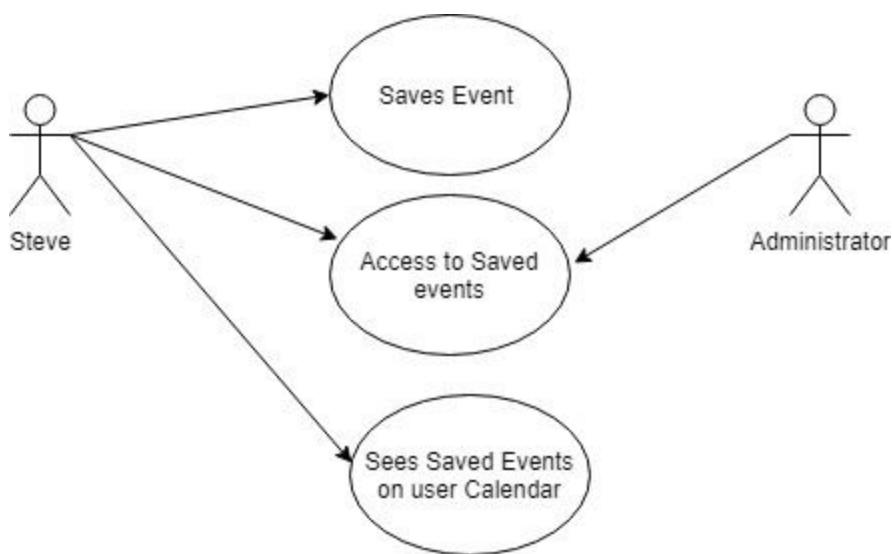
2. Anthony is a student organizer at San Francisco State University. He is part of an organization that owns and maintains the arcade on campus. Many people frequent the arcade at school almost everyday, some machines/games in particular have regulars too. They expressed their desire to have a local tournament with the machines, Anthony agrees and **needs a way to create and let people know of such an event**. The school website doesn't provide a good way to market the event, so Anthony resorted to placing flyers around the arcade to promote the event. Anthony found out about Unigator and decided to use it to **host the event**. Unigator helps promote the event easier by possibly being **featured on the homepage and calendar of the site when approved by administers**.

Actors: Anthony, Event Host



3. Steve is an aspiring software engineer studying in San Francisco State University. Steve needs to find out about possible opportunities to further his career and improve his resume. The school site provides info about most events but Steve is only interested in technology based events, the site doesn't have a **way to filter the events**. A friend recommended the application Unigator, which allows him to **browse** through events and **filter them by technology category**, and provides Steve with all the **necessary information of events** in San Francisco State University. Steve **wanted to save** all the events that seem interesting to him so he can look over and **organize** later on. Steve found out that he will need to **register an account with his school email** in order to use those features. Steve registered with his school email is able to **save the events** and better organize his schedule. Steve also registered his account with his own **unique username**.

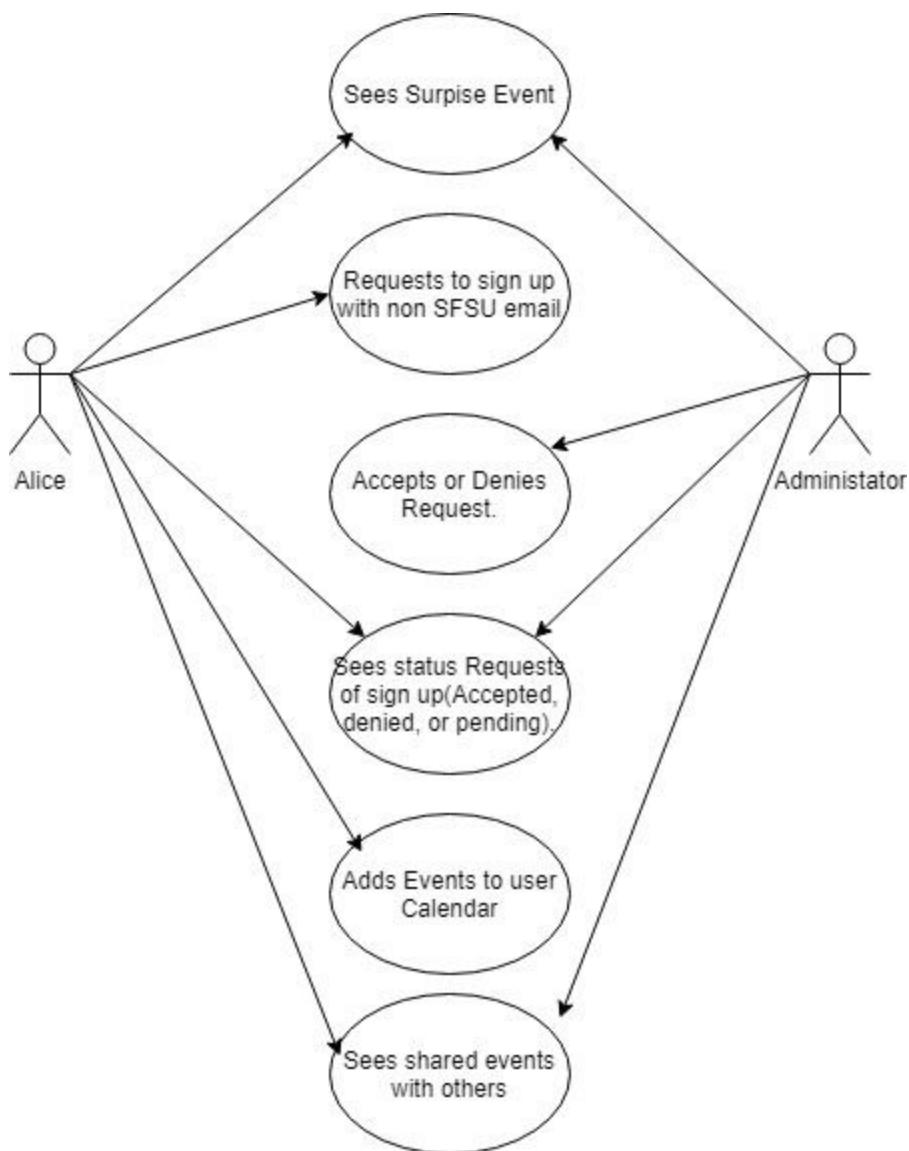
Actors: Steve, Registered User



4. Alice is a high school student that plans to attend San Francisco State University in the future. She wants to know more about the school by going to some events in the school, but is unsure of what events she's looking for specifically. The school website is complex, Alice has a hard time navigating the website just to find information about events. She then stumbled upon Unigator, an application about events happening **in San Francisco State University**. Unigator provides a **“surprise” function**, which randomly recommends an event to the user. Alice used this function and got recommended to a social event that she's interested in. She wanted to **save the info** onto a calendar but she needed an account. She is **unable to register an account as it can only be created with the**

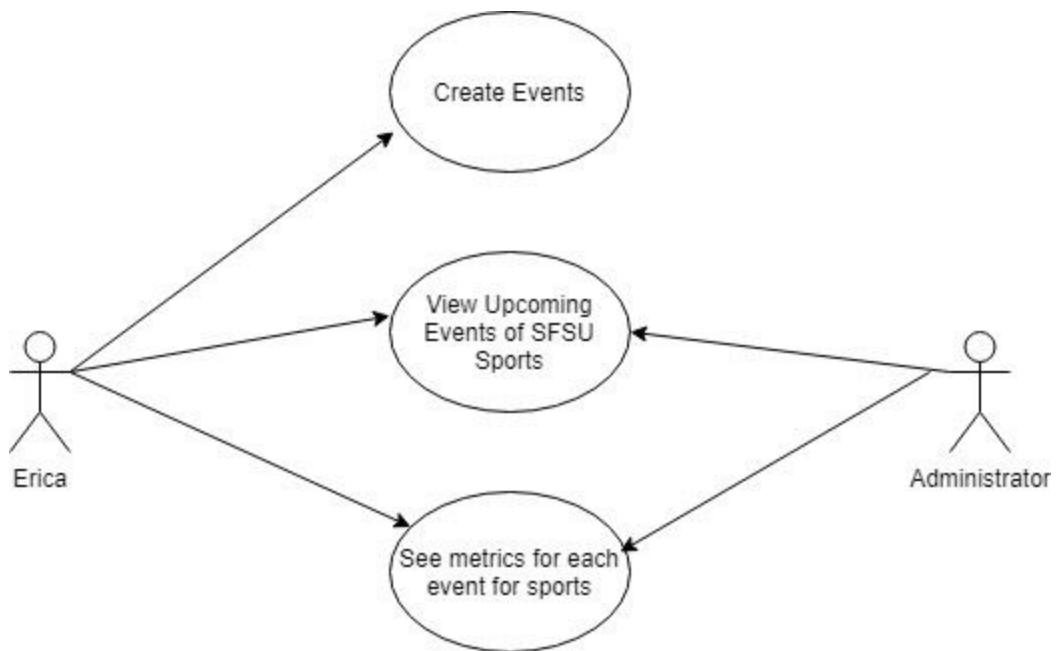
university's email. Therefore, while Alice is able to **browse through events** and learn about it, she has to write down all the relevant info about it. Because Alice is **not a registered user**, she is **not able to share an event's info** and her engagement of it with her social media platform. Alice learns that there are two different ways to become a registered user; she can either create an account with a sfsu email, or she can **submit a request to an administrator** to be able to **register with a personal email**. After being approved by an administrator, Alice is able to create an account without having an sfsu email and is now able to use the **share function** to let her friends and family know what events she is interested in.

Actors: Alice, Unregistered User

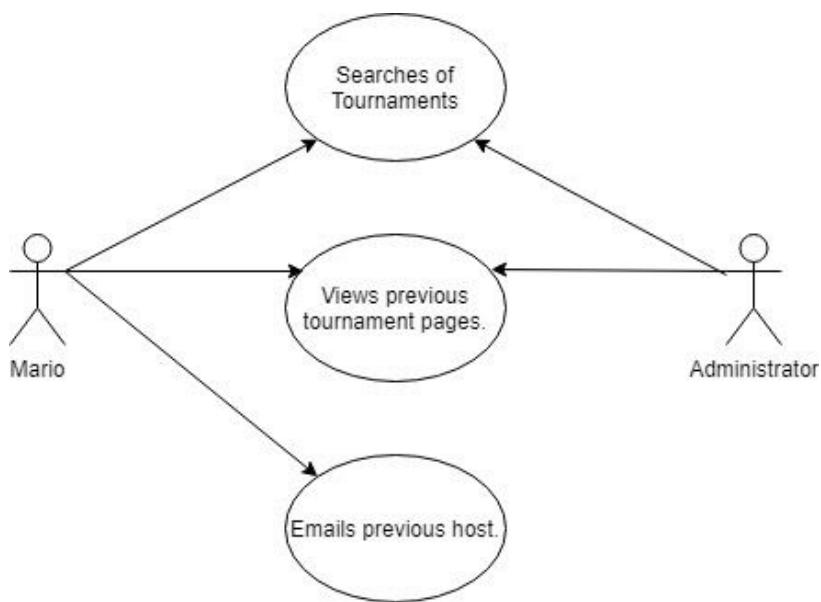


5. Erica is an organizer for the athletic teams in San Francisco State University. She is responsible for entering all of the dates and times for various sporting events taking place on campus. Previously, sporting event schedules are displayed only on the SF State Gators athletics website, but ever since discovering Unigator, Erica has been **posting the events** there as well. The old SF State Gators website only shows the time, date, and location of sporting events. It doesn't offer an option to **reserve tickets online** ahead of time. Tickets are distributed at the gate to sporting facilities only. SFSU students are allowed to attend sporting events for free, so registered student accounts on Unigator can **RSVP and reserve a digital ticket** for free ahead of time, eliminating the need to pull out their student ID card at the ticket booth. Unregistered users can still purchase a ticket in person at the event. Erica can use ticket reservation numbers to gauge student interest ahead of time.

Actors: Erica, Event host

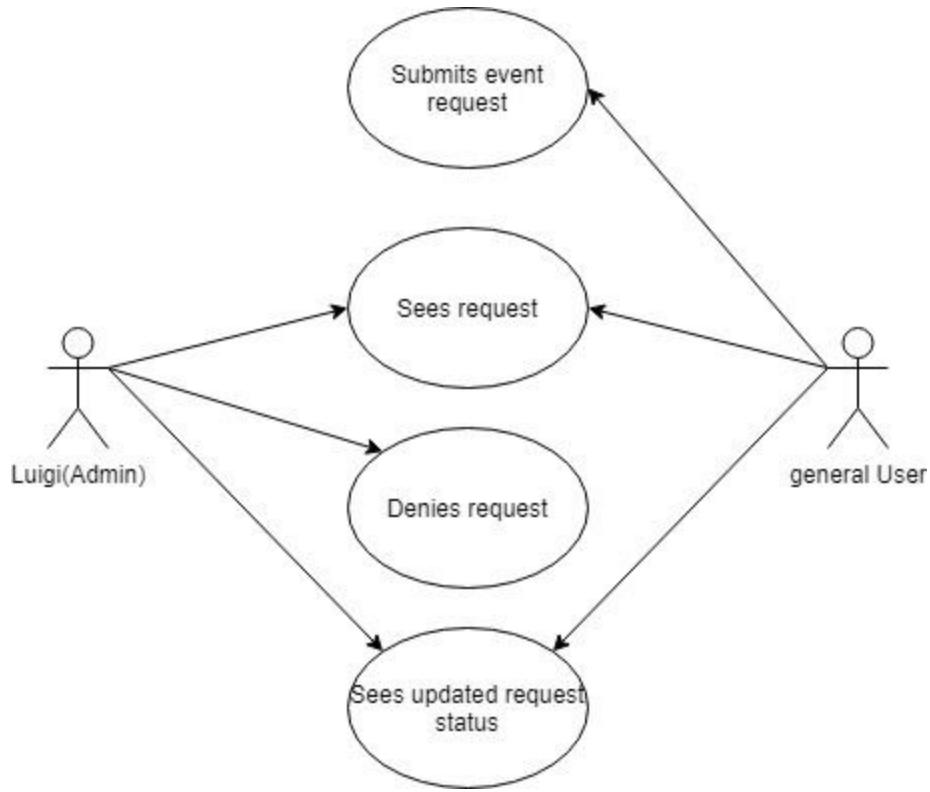


6. Mario is a freshman at San Francisco State University, on his way to class he overhears some other students talking about a Super Smash Bros tournament on campus, a game which he plays and would like to check out himself. Mario goes to unigator and uses the **search bar** to find information about future tournaments, but the only results which **show up are of past tournaments**. Still wanting to get information about the next tournament, Mario **views one of the past tournament's event pages** where he finds the **email** of the previous **event host** and is only visible if he is logged in. Mario proceeds to create an account where he is able to view the ex-host's email and get into contact with them.
- Actors: Mario, Unregistered User



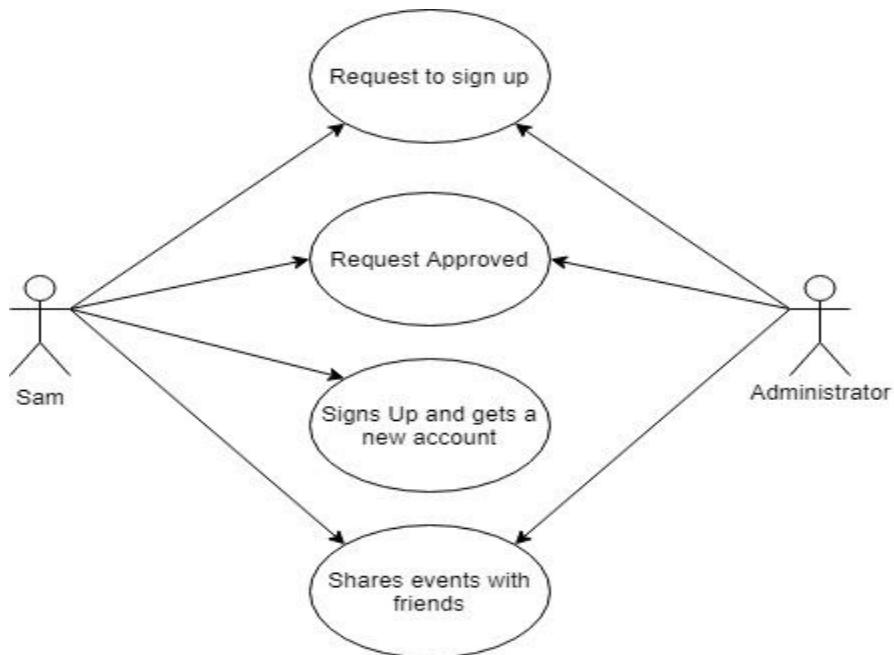
7. Luigi works for Unigator as an **administrator**. All events need to go through Luigi in order to be **approved**. Luigi **logs onto** his account, which has the **administrator privileges**. Luigi goes through the **event approval list** and approves those he sees fit. Luigi came across an event request that was unreasonable and confusing, he **rejected** the proposal and a **notification was sent** with explanation of why it was rejected. After Luigi has done his job, Luigi just casually **browse through events** like any other users. After going through the list, Luigi **logs out** of his admin account.

Actors: Luigi, Administrator



8. Sam is a student at a local community college. He is interested in cultural events. He sees 2 events he wants to go to with his friends at SFSU. Since Sam doesn't go to SFSU, he can not create an account. So he requests to sign up with a non SFSU email from the administrator. His request gets approved and he is able to share the events he is interested in with his SFSU friends on social media and is able to RSVP to the events.

Actors: Sam, Unregistered User



List of main data items and entities

Data Entities	Definition
Unregistered User	Users that are not registered with the site, able to browse through events and look at the information, but are unable to utilize some functions of Unigator, such as bookmarking the event and sharing it.
Registered User	Users that are registered with the site with their SFSU email. Able to browse through events and save them into their own personal calendar and share the events with social media. Able to update their status whether they are attending the event.
Events	Activities that are displayed on Unigator that takes place in SFSU.
Event Host	The registered user that is hosting the event, provides all the essential info of their event such as date, time, location and description.
Event Guest	A registered user that declared that they are attending the event. This allows the event host to gauge how much attendees the event might potentially have.
Administrator	A higher privilege user that oversees the site. Able to decide whether an event is to be listed or not. Able to ban users and send warnings.
Approved Background Credentials	Email needs to be verified as a SFSU email.
Event Organizers	An organization that coordinates the event and sets up all the required fields. Make sure that the location and time is suitable to host the event and communicate with the school if there is any disagreement.
Event Location	The location of a specific event.

Event Time	The time of a specific event.
Event List	A list that contains all of the events currently scheduled or ongoing in SFSU.
RSVP List	A list showing all registered users attending an event.
Activity Points	Points that act as a currency to be used in the point shop.

Initial list of functional requirements

Unregistered Users

1. Unregistered Users shall be able to sign up with their SFSU email.
2. Unregistered Users shall not be able to sign up without their SFSU email.
3. Unregistered Users shall be able to browse through the event list.
4. Unregistered Users shall be able to see all the essential information about an event.
5. Unregistered Users shall not be able to view the email of event hosts.
6. Unregistered Users shall not be able to bookmark/star an event.
7. Unregistered Users shall not be able to RSVP to an event.
8. Unregistered Users shall not be able to view an event's RSVP list.
9. Unregistered Users shall not be able to report a(n) event.
10. Unregistered Users shall not be able to report a(n) user.
11. Unregistered Users shall not be able to share with their social media.
12. Unregistered Users shall not host or create an event.
13. Unregistered Users shall not be able to utilize the calendar function.
14. Unregistered Users shall be able to view past events.
15. Unregistered Users shall be able to make a request to the Administrator to register.
16. Unregistered Users shall not be able obtain points based on interactions on the site.
17. Unregistered Users shall not be able to earn points.
18. Unregistered Users shall be able to view the points shop.
19. Unregistered Users shall not be able to purchase from the points shop.
20. Unregistered Users shall not be able to edit their profile.
21. Unregistered Users shall not be able to view a user's profile.

Registered Users

22. Registered Users shall be able to log in with their account.
23. Registered Users shall be able to log out of their account.
24. Registered Users shall be able set a unique username.
25. Registered Users shall be able to request to change their unique username.
26. Registered Users shall be able to browse through the event list.
27. Registered Users shall be able to utilize the personal calendar function.
28. Registered Users shall be able to see all the essential information about the event.
29. Registered Users shall be able to view the email of event hosts.

30. Registered Users shall be able to view past events.
31. Registered Users shall be able to bookmark/star an event.
32. Registered Users shall be able to un-bookmark/un-star an event.
33. Registered Users shall be able to RSVP to an event.
34. Registered Users shall be able to un-RSVP to an event.
35. Registered Users shall be able to view an event's RSVP list.
36. Registered Users shall be able to report a(n) event.
37. Registered Users shall be able to report a(n) user.
38. Registered Users shall be able to share with their social media.
39. Registered Users shall be able to host or create an event.
40. Registered Users shall be able to earn points.
41. Registered Users shall be able to view the points shop.
42. Registered Users shall be able to purchase from the points shop.
43. Registered Users shall be able to switch to previous options in the points shop.
44. Registered Users shall be able to edit their profile.
45. Registered Users shall be able to view a user's profile.

Administrators

46. Administrators shall be able to log in with their account.
47. Administrators shall be able to log out of their account.
48. Administrators shall be able to view all event proposals.
49. Administrators shall be able to approve an event proposal.
50. Administrators shall be able to reject an event proposal.
51. Administrators shall be able to request/demand changes from an event host.
52. Administrators shall be able to remove an event.
53. Administrators shall be able to see all the information about an event.
54. Administrators shall be able to view an event's RSVP list
55. Administrators shall be able to view the email of event hosts.
56. Administrators shall be able to view past events.
57. Administrators shall be able to bookmark/star an event.
58. Administrators shall be able to un-bookmark/un-star an event.
59. Administrators shall be able to RSVP to an event.
60. Administrators shall be able to un-RSVP to an event.
61. Administrators shall be able to view an event's RSVP list.
62. Administrators shall be able to view all requests to use a non-SFSU email to register.
63. Administrators shall be able to approve a request to use a non-SFSU email to register.

64. Administrators shall be able to deny a request to use a non-SFSU email to register.
65. Administrators shall be able to view all requests from users to change their username.
66. Administrators shall be able to approve a user's request to change their username.
67. Administrators shall be able to deny a user's request to change their username.
68. Administrators shall be able to change any registered user's username.
69. Administrators shall be able to ban a registered user.
70. Administrators shall be able to earn points.
71. Administrators shall have all purchases in the point shop unlocked.
72. Administrators shall be able to edit their profile
73. Administrators shall be able to view a user's profile.
74. Administrators shall be able to request a user make changes to their profile.

Event Host

75. Event Host shall be a registered user.
76. Event Host shall be able to edit the event that they are hosting.
77. Event Host shall be able to remove the event that they are hosting.
78. Event Host shall be able to request special changes from Administrator.
79. Event Host shall have the same functionality as a registered user.
80. Event Host shall be able to transfer/assign a new event host.

List of non-functional requirements

- 1) Security:
 - a) Can only start an event page if you are logged into your account.
 - b) You can only create an account with a SFSU email.
 - c) Able to request change of password.
 - d) Passwords will be encrypted.
 - e) Have password requirements for creating an account and resetting password.
- 2) Audit:
 - a) The user may request access to create an account without SFSU email.
 - b) The administrator may delete any event from the event page.
 - c) The administrator may view all RSVP accounts of any event.
 - d) The administrator may view and approve any event requests.
 - e) New events must be approved by the administrator.
- 3) Response Time:
 - a) The initial load up time on the site shall fall under 5 seconds.
 - b) Retrieving data from the database shall be fast and efficient.
- 4) Capacity:
 - a) The application shall be able to hold at least 50 events within the span of two weeks.
 - b) The application shall save a log of past events within 2 months.
 - c) The application shall be capable of handling accounts of at least all the SFSU students.
 - d) The application shall be made in a way that it's scalable, in case of and further feature updates are planned.
- 5) Reliability:
 - a) Downtime maintenance shall be done within 5 hours.
 - b) Downtime maintenance shall restrict users from using the application to prevent any false information.
 - c) The application shall not crash under any circumstances.
 - d) Registered Users shall be informed of maintenance via an announcement on the main page, or through email.
- 6) Recovery:
 - a) If any severe problems arise in the application, the application shall be down for repairs.
 - b) Application downtime shall not exceed 24 hours.

- c) If the application is down, Registered User data and Events data shall be safe.
- 7) Data Integrity:
- a) All Data shall be backed up every 72 hours.
 - b) The users shall be prompted when they wish to delete or edit data.
 - c) Any purchases in the point shop shall be saved with the Registered User's account.
 - d) Account recovery can be requested to the Administrator.
 - e) Image sizes shall not exceed 4 megabyte.
 - f) Only image format of JPG, JPEG and PNG shall be accepted.
- 8) Platform:
- a) The application shall be compatible with Chrome browser version 80.0.3987.132 and earlier.
 - b) The application shall be compatible with Firefox browser version 74.0 and earlier.
 - c) The application shall be compatible with Safari browser version 13.0 and earlier.
 - d) The application shall scale correctly with the screen size of the device its on
 - e) The application shall be compatible with any client OS that supports Chrome, Firefox, or Safari browsers.
 - f) The application shall be able to account for any compatibility issues as a result of browser or OS updates.
- 9) Coding Standards:
- a) Comments should accompany the code in order to understand functionality.
 - b) Errors shall not disrupt the application in a way that is detrimental.
 - c) Code needs to be reviewed and tested by all members of the group before going into the master branch.
 - d) Internal and external error shall be recorded in a log for future reference.
 - e) Any error or bugs that affect the user should be notified to them through their email.
 - f) Error debugging shall be done without interfering with the application's functionality.
 - g) All priority one requirements shall be implemented and working before launching the application.
- 10) Look and Feel Standards:

- a) The application should follow the aesthetics of San Francisco State University's site.
 - b) Registered Users profiles are unique and customizable by the user.
 - c) The application shall appear simple in order to avoid any confusion to the users.
 - d) Registered Users needs shall be prioritized over Unregistered Users.
 - e) The application should be automatically resized based on the screen size of the device.
 - f) Have a loading toggle wheel.
 - g) Have Links to Unigator social media in footer.
 - h) Follow the color scheme of SFSU with purple and gold.
- 11) Internalization / Localization:
- a) English shall be the default language of the application.
 - b) The application shall only allow SFSU students to register accounts.
 - c) Non-SFSU students shall be able to make requests.
- 12) Website Policies:
- a) The application's policies shall be visible via a link on the homepage.
 - b) The application shall not ask for any sensitive information.
 - c) Registered User info shall be kept confidential.
 - d) A verification of the email shall be done in the register section.
 - e) Users shall understand and read through all the policies and abide by it when using the application.

Competitive analysis

Feature/ Company	EventBrite	SFSU University Calendar	Associated Students Events	Gator X	Unigator
Strengths	Shows almost all events in SF with important info. Clear and simple to use UI.	Shows events in SFSU, provides all relevant info. Date search filter. Fast response time.	Clear and spacious UI, with relevant info displayed on the homepage. Displays info relevant to students of SFSU.	Lists out Events events by date. Good Events Wall	Displays all events scheduled or ongoing in San Francisco State University
Weaknesses	Doesn't show smaller scale events, such as in-school events. Only promotes large events by huge organizations.	Mostly small words, no save features for SFSU account. Only shows events by organizers and not by students. No map and calendar feature.	Not mainly focusing on events, but rather all aspects concerning the school in general. Doesn't show much info and uses google calendar in order to display more info. Too many menus, confusing.	Poor graphics and poor UI. Hard to find and access the application.	Does not provide info about any event taking place outside of campus.
Pricing	Depends on the events	Free	Free	Free	Free
Social Media	Integration with social media platforms.	None	Only points to their specific social media page, which	None	Shares the event to social media

			posts certain info.		platform
--	--	--	---------------------	--	-----------------

+ feature exists ++ superior - does not exist	EventBrite	SFSU University Calendar	Associated Students Events	GatorX	Unigator
Calendar System	+	-	+	+	+
General info	+	+	+	+	+
First Glance Info	+	-	-	+	++
Checklist/Bookmark	+	-	-	-	+
Pictures of Event	+	-	-	+	+
Creating an Event	+	+	+	+	+
Simple and Easy to use UI	-	-	-	+	+
Account Required to use certain features	+	-	-	+	+
Map System	+	-	-	+	+
Filter System	+	+	-	+	+
Event Suggestions	+	-	+	-	+
Search Bar/Browse	+	-	+	+	+
Social Media Interactiveness	+	-	-	-	+
Point System and Shop	-	-	-	-	++

The advantages of using our application Unigator over others is that it prioritizes events based on the San Francisco State University campus. Unigator gives the user a central dynamic platform to get all their event needs. Other platforms generally focus on a broad range of events spanning the whole city, but our platform is targeted directly to San Francisco State University students. Students will be able to see all events concerning San Francisco State University on Unigator. Students will also be able to access more features, giving them more dynamic control over their account. Unigator stands out as a unique platform due to its unique point system, which registered users are able to spend in the point shop. The point shop provides various customization options for the registered users profile. Points are accumulated through interaction on Unigator and are spent on the point shop for better representing the uniqueness of a profile.

High-level system architecture and technologies used:

- Application's Backend: Node Javascript
- Application's Frontend: React Javascript
- Database: MySQL
- Cloud: Amazon Web Services
- Development Environment: Visual Studio/Webstorm
- OS Architecture: Ubuntu 18.04

Team

Role	Name
Team Lead	Lionel Wong
Github Master	Mitul Savani
Backend Lead	Jorge Landaverde
Backend engineer	Gordon Lam
Frontend Lead	Mitul Savani
Frontend engineer	Kevin Huang/Jack Kower

Checklist

• Team found a time slot to meet outside of the class	DONE
• Github master chosen	DONE
• Team decided and agreed together on using the listed SW tools and deployment server	DONE
• Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	DONE
• Team lead ensured that all team members read the final M1 and agree/ understand it before submission	DONE
• Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	DONE

SW Engineering - CSC648/848

Spring 2020



Team 03

Milestone 2

Lionel Wong	lwong7@mail.sfsu.edu	Team Lead
Jorge Landaverde	jlandaverde@mail.sfsu.edu	Lead Backend Engineer
Gordon Lam	glam1@mail.sfsu.edu	Backend Engineer
Mitul Savani	msavani@mail.sfsu.edu	Lead Frontend Engineer/Git Master
Kevin Huang	khuang2@mail.sfsu.edu	Frontend Engineer
Jack Kower	jkower@mail.sfsu.edu	Frontend Engineer

Milestone	Date
Milestone02 Version02	4/15/2020
Milestone02 Version01	3/25/2020
Milestone01 Version02	3/12/2020
Milestone01 Version01	3/5/2020

Table Of Contents

I.	Data Definitions V2	2
II.	Functional Requirements V2	3
III.	UI Mockups and Storyboards	8
IV.	High level Database Architecture and Organization	17
V.	High Level APIs and Main Algorithms	22
VI.	High Level UML Diagram	24
VII.	High Level Application Network and Deployment Diagrams	25
VIII.	Actual Key Risks of the Project	27
IX.	Project Management	29

Data Definitions V2

Data Entities	Definition
User	A general user that is browsing the internet.
Account	An account that is unique to a user, which will be registered with email and password.
Unregistered User	Users that are not registered with the site, able to browse through events and look at the information, but are unable to utilize some functions of Unigator, such as bookmarking the event and sharing it.
Registered User	Users that are registered with the site with their SFSU email. Able to browse through events and save them into their own personal calendar and share the events with social media. Able to update their status whether they are attending the event. Every registered user will have a display name.
Events	Activities that are displayed on Unigator that takes place in SFSU. Every event will have to have a name and important information regarding the event; such as who the event host is, location, time, etc. An event can be of single occurrence or recurring.
Event Host	A registered user that is hosting/registering a(n) event, and provides all the essential information of the event; such as date, time, location and description. Will also be the contact listed for said event.
RSVP List	A list showing all registered users who plan to attend an event. A user can RSVP to multiple events even if there is a time conflict.
Administrator	A higher privilege user that oversees the site. Able to decide whether an event is to be listed or not. Able to ban users and send warnings. Moderates the site by utilizing special permissions only available to an admin account. Maintains the integrity of the site while keeping sure everything is in order.
Category	The categories that an event falls under. For example, an art gallery will be put under the 'Art' category. A recruiting event from a technology company would be under the 'Technology' category.

Activity Points	Every registered user will have their own wallet of Activity Points. These are points that act as a currency to be used in a point shop. Each profile represents a unique representation of a user. The profile can be customized with various options, options that can be purchased from the point shop page with earned activity points. Activity points are earned by interacting with the site and being active, example being checking the RSVP list and attending the event, bookmarking an event, and creating an event etc.
Host Points	Every Host has a list of users that have attended any of their events. The host will be awarding points during their meetings for attendance or participation. These points will be used to reward attendees and increase events' attendees count. The host can enable this feature and these points are only specific to this specific hosts' store.

Functional Requirements V2

Priority 1

Requirements
1) Unregistered Users
1.1) Unregistered Users shall be able to sign up with their SFSU email.
1.2) Unregistered Users shall be able to browse through the event list.
1.3) Unregistered Users shall be able to see all the essential information about an event.
1.4) Unregistered Users shall be able to view past events.
2) Registered Users
2.1) Registered Users shall be able to log in with their account.
2.2) Registered Users shall be able to log out of their account.
2.3) Registered Users shall be able set a unique username.
2.4) Registered Users shall be able to request to change their unique username.
2.5) Registered Users shall be able to browse through the event list.
2.6) Registered Users shall be able to utilize the personal calendar function.
2.7) Registered Users shall be able to see all the essential information about the event.
2.8) Registered Users shall be able to view the email of event hosts.
2.9) Registered Users shall be able to view past events.
2.10) Registered Users shall be able to bookmark/star an event.
2.11) Registered Users shall be able to un-bookmark/un-star an event.
2.12) Registered Users shall be able to RSVP to an event.
2.13) Registered Users shall be able to un-RSVP to an event.

2.14) Registered Users shall be able to view an event's RSVP list.
2.15) Registered Users shall be able to edit their profile.
2.16) Registered Users shall be able to view a user's profile.
2.17) Registered Users shall be able to share with their social media.
2.18) Registered Users shall be able to host or create an event.
2.19) Registered Users shall be able to earn points.
2.20) Registered Users shall be able to view the points shop.
2.21) Registered Users shall be able to purchase from the points shop.
2.22) Registered Users shall be able to switch to previous options in the points shop.
3) Administrators
3.1) Administrators shall be able to log in with their account.
3.2) Administrators shall be able to log out of their account.
3.3) Administrators shall be able to view all event proposals.
3.4) Administrators shall be able to approve an event proposal.
3.5) Administrators shall be able to reject an event proposal.
3.6) Administrators shall be able to remove an event.
3.7) Administrators shall be able to see all the information about an event.
3.8) Administrators shall be able to view an event's RSVP list
3.9) Administrators shall be able to view the email of event hosts.
3.10) Administrators shall be able to view past events.
3.11) Administrators shall be able to bookmark/star an event.
3.12) Administrators shall be able to un-bookmark/un-star an event.
3.13) Administrators shall be able to RSVP to an event.

3.14) Administrators shall be able to un-RSVP to an event.
3.15) Administrators shall be able to view an event's RSVP list.
3.16) Administrators shall be able to ban a registered user.
3.17) Administrators shall have all purchases in the point shop unlocked.
3.18) Administrators shall be able to edit their profile
3.19) Administrators shall be able to view a user's profile.
3.20) Administrators shall be able to request/demand changes from an event host.
4) Event Host
4.1) Event Host shall be a registered user.
4.2) Event Host shall be able to edit the event that they are hosting.
4.3) Event Host shall be able to remove the event that they are hosting.
4.4) Event Host shall have the same functionality as a registered user.
4.5) Event Host shall have the same functionality as a registered user.
4.6) Event Host shall be able to enable host points feature
4.7) Event Host shall be able to grant RSVPed users points

Priority 2

Requirements
1) Unregistered Users
1.1) Unregistered Users shall be able to view the points shop.
1.2) Unregistered Users shall be able to make a request to the Administrator to register.
2) Registered Users
2.1) Registered Users shall be able to report a(n) event.

2.2) Registered Users shall be able to report a(n) user.

3) Administrators

3.1) Administrators shall be able to view all requests from users to change their username.

3.2) Administrators shall be able to approve a user's request to change their username.

3.3) Administrators shall be able to deny a user's request to change their username.

3.4) Administrators shall be able to change any registered user's username.

3.5) Administrators shall be able to request a user make changes to their profile.

4) Event Host

4.1) Event Host shall be able to request special changes from the Administrator.

4.2) Event Host shall be able to transfer/assign a new event host.

Priority 3

1) Unregistered Users

1.1) Unregistered Users shall be able to make a request to an Admin to create an account without an SFSU email.

3) Administrators

3.1) Administrators shall be able to view all requests to use a non-SFSU email to register.

3.2) Administrators shall be able to approve a request to use a non-SFSU email to register.

3.3) Administrators shall be able to deny a request to use a non-SFSU email to register.

UI Mockups and Storyboards

UI Mockups

Registration

Login Page

Unigator	Search	<input type="text"/> Q	<input type="button" value="Login"/>	<input type="button" value="Sign Up"/>
----------	--------	------------------------	--------------------------------------	--

Login

SFSU E-mail	
Password	
<input type="button" value="Login"/>	
Sign up for a new account — Link to sign up page	

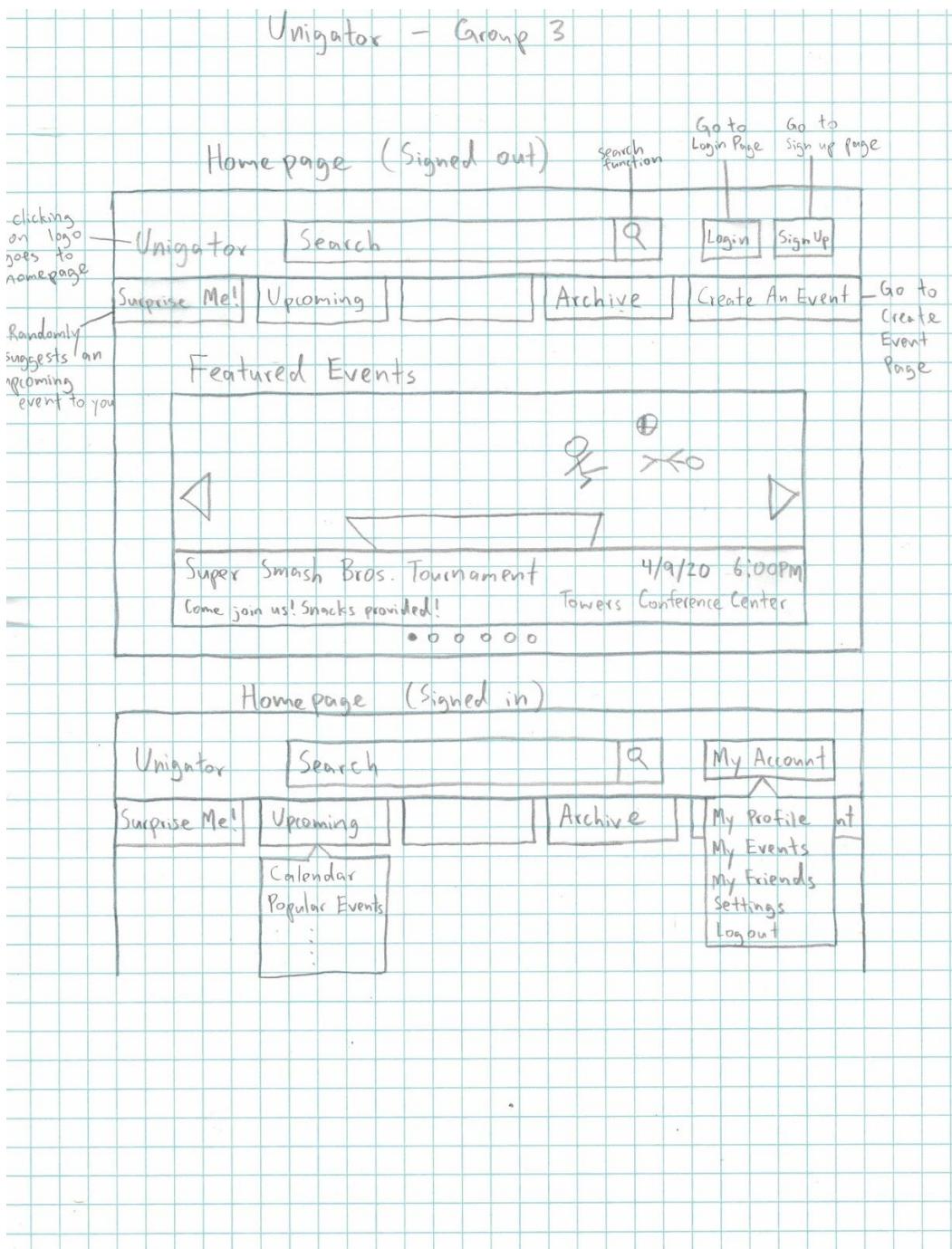
Sign up Page

Unigator	Search	<input type="text"/> Q	<input type="button" value="Login"/>	<input type="button" value="Sign Up"/>
----------	--------	------------------------	--------------------------------------	--

Sign Up

NOTE: Only SFSU Students can sign up on this page. Non-SFSU students can request an account by contacting an administrator: Contact Us		Link to contact administrator
SFSU E-mail		
Username		
Password		
<input type="button" value="Sign Up"/>		Link to login page
I already have an account —		

Home Page



Create Event Page

Create Event Page
Page only available while logged in.

Unigator	Search	Q	My Account
Surprise Me!	Upcoming		Archive Create An Event
Create a New Event			
Name of event : Super Smash Bros. Tournament Date : 04 / 09 / 2020 <small>Month Day Year</small> Start Time : 06 : 00 AM/PM End Time : 09 : 00 AM/PM Event Type : Recreation <input checked="" type="checkbox"/>			
Description : Join us for a fun evening of Super Smash Bros. Ultimate! Controllers are provided. Light snacks and refreshments provided.			
Photo(s) : <input type="button" value=" "/> <input type="button" value=" "/> <input type="button" value=" "/> <input type="button" value=" +"/>			
Admission : <input checked="" type="checkbox"/> Free <input type="checkbox"/> Enter price : \$ <input type="text"/>			
Location : Towers Conference Center			
<input type="button" value="Submit Event"/>			
Your event will be submitted for approval by Unigator administrators within 48 hours.			

My Profile Page

My Profile Page

Unigantor Search My Account

Jo_Smith27

Information (edit)

Name: John Smith (Private)
Username: Jo_Smith27
Email: josmith@mail.sfsu.edu (private)

Change Profile Picture

My Friends (22) See All

Goes to my friend page

Saved Events (2) See All

Goes to my events page

3/20/20 Concert 6:00PM
3/27/20 Baseball Game 12:00PM

My Friends Page

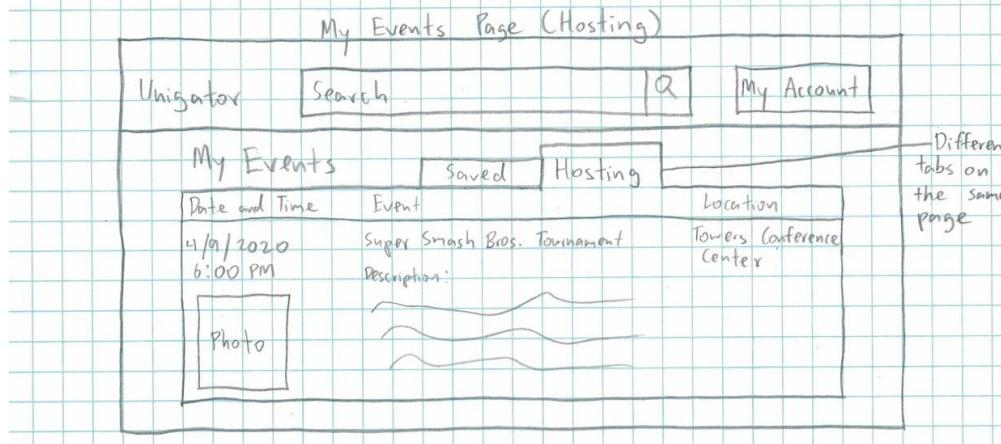
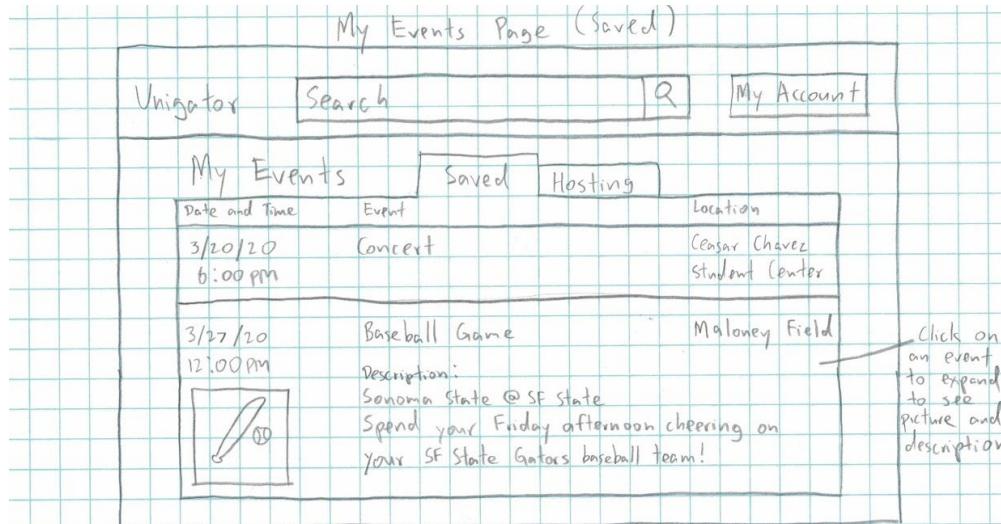
Unigantor Search My Account

Jo_Smith27

My Friends

Firstname Lastname Clicking on a friend takes you to their profile

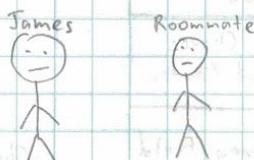
Saved Event Page



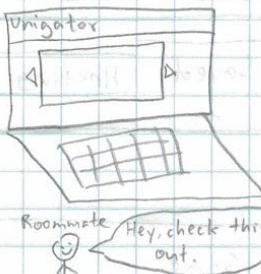
Storyboards

Storyboard #1

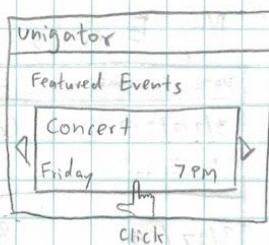
James wants to find an easy way to find various events happening around SFSU, and be able to share them with his roommate.



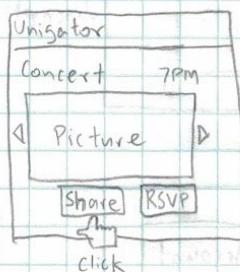
His roommate introduces James to Unigator, our application



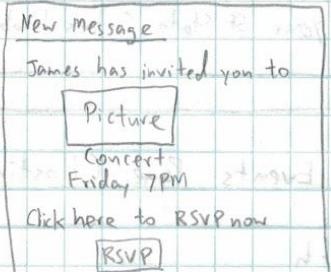
James clicks an event he is interested in going to on the website's homepage.



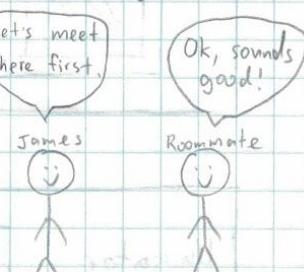
James clicks on the "share" button on the event page to show the event to his roommate.



James' roommate receives an email saying James is inviting you to the event.



James and his roommate are now planning to go to the event together.



Storyboard #2

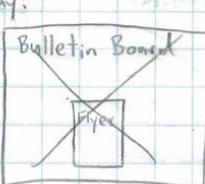
Anthony wants to host an arcade game tournament at SFSU.



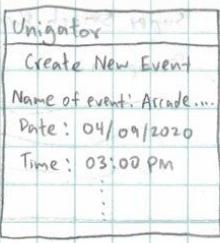
(How can I easily let people know about my tournament?)

Anthony

He thought about placing flyers around the arcade, but he wants a better way.



He decides to post the event on Unigator instead.



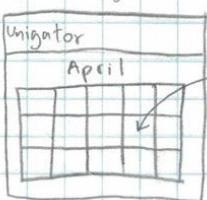
An administrator on Unigator inspects the details of the event and approves it.

Admin

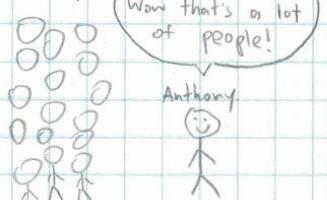


Looks good to me!

The event now appears on the Unigator calendar and is eligible to be featured on homepage if it's popular.



A much larger crowd shows up to the tournament than if Anthony just put up flyers.



Storyboard #3

Steve wants to look for technological events on campus to help further his career and improve his resume.

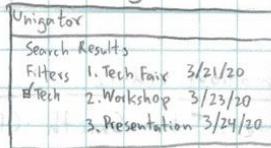
Steve  where can I find tech related events in SFSU?

He is directed to sign up for an account using his SFSU email before he can use the save function.



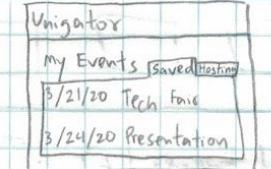
Unigator
Sign Up
SFSU Email
Username
Password
Sign Up

Steve goes on Unigator and filters events by "technology" category only. He sees a list of all upcoming tech events.



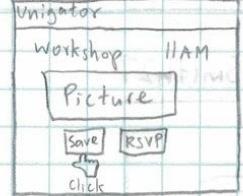
Unigator
Search Results
Fitness 1. Tech Fair 3/21/20
Tech 2. Workshop 3/23/20
3. Presentation 3/24/20

After signing up, Steve can go to the "My Events" page to view all of the events he has saved.



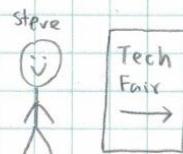
Unigator
My Events
Saved
3/21/20 Tech Fair
3/24/20 Presentation

Steve goes to the event page of the events he's interested in attending. He's able to save events for future reference.



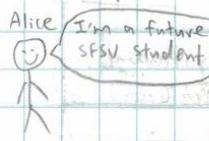
Unigator
Workshop HAM
Picture
Save RSVP
Click

Now Steve has a clear idea of what tech events are being offered on campus and which events he wants to attend.



Storyboard #4

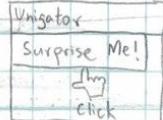
Alice is a high school student planning on attending SFSU. She wants to learn more about the school by attending events on campus.

Alice  I'm a future SFSU student

She doesn't know what to look for exactly, and the SFSU website is confusing to navigate.

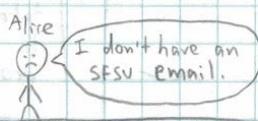


Alice stumbles upon Unigator, which provides a "Surprise Me" feature, which randomly suggests upcoming events.



Unigator
Surprise Me!
Click

Alice wants to save the events she finds onto her personal calendar, but realizes she needs an Unigator account. She cannot directly make an account without an SFSU email.

Alice  I don't have an SFSU email.

She also cannot share events on social media without an account.

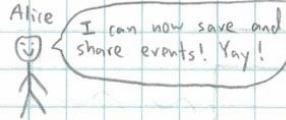


Alice discovers she can contact an Unigator administrator to request to register for an account using her personal email.



Unigator
Contact Us
Click

After her request has been approved, Alice can now create an account without an SFSU email.

Alice  I can now save and share events! Yay!

Alice is now able to let her friends and family about the events she is attending.



Unigator
Share
Click

Storyboard #5

Erica is an organizer for sporting events in SFSU. She is responsible for entering event schedules to the SF State Gators athletics website.

Erica



She discovers Unigator and starts posting the events there as well.

Unigator
Create a New Event
Name: Volleyball
Date: 5/3/20
⋮

The old SF State Gators website only shows the date, time, and locations of the events. It doesn't provide a way to reserve tickets ahead of time.

SF State Gators
Volleyball Schedule
⋮

SFSU students are allowed to get a free ticket to all sporting events. The only option used to be distributing tickets physically at the event gate.

Admit One

Students can now RSVP on Unigator and use their confirmation page as a digital ticket for the event.

Unigator
RSVP Confirmed!
Show this page as your ticket.

Erica can use these RSVP numbers to gauge interest in the events before they occur.

Unigator
RSVP count: 243

Storyboard #6

Mario is a freshman at SFSU. He hears other students talking about a Super Smash Bros. tournament, a game he likes.

Student
Are you going to the tournament?Student
Yes!

He goes to Unigator and uses the search bar to find more information on future tournaments.

Unigator
Super Smash Bros
Search Results
⋮

However, Mario only finds past events from the search results, not upcoming tournaments.

Mario
I want to find future events, not old ones.

He clicks on one of the past events and sees an option to see the email of the event host. Mario needs to be logged into an Unigator account to see the email.

Unigator
Super Smash Bros.
January 30, 2020
Host Information
Requires Unigator account to reveal info

Mario creates a new Unigator account.

Unigator
Sign Up
SFSU Email
Username
Password
Sign Up

Mario is now able to see the former host's email to ask for more information about future events.

Mario
When is the next tournament?Host
Next week

Email exchange

Storyboard #7

Luigi is an administrator at Unigator. He has to approve all events before they go live on the website.

After logging into his account with admin privileges, he can see a list of pending events. He approves events as he sees fit.

Luigi will deny events that are unreasonable, inappropriate, or confusing. The host will receive an email with an explanation why it was denied.

Luigi can browse the Unigator site just like any other registered user, aside from his admin privileges.	After completing his work, Luigi will log out of his account.	

Storyboard #8

Sam is a student at a local community college. He is interested in attending cultural events at SFSU.

Sam: What events are offered at SFSU?

He finds 2 events he wants to attend with his friends at SFSU.

Sam: I wonder if Jason and Henry would be interested too.

Since Sam is not a SFSU student, he needs to request an account from an Unigator administrator.

Once his request has been approved, Sam can now share events on Unigator with his friends at SFSU.	Sam can now RSVP to events he wants to attend.	

High level Database Architecture and Organization

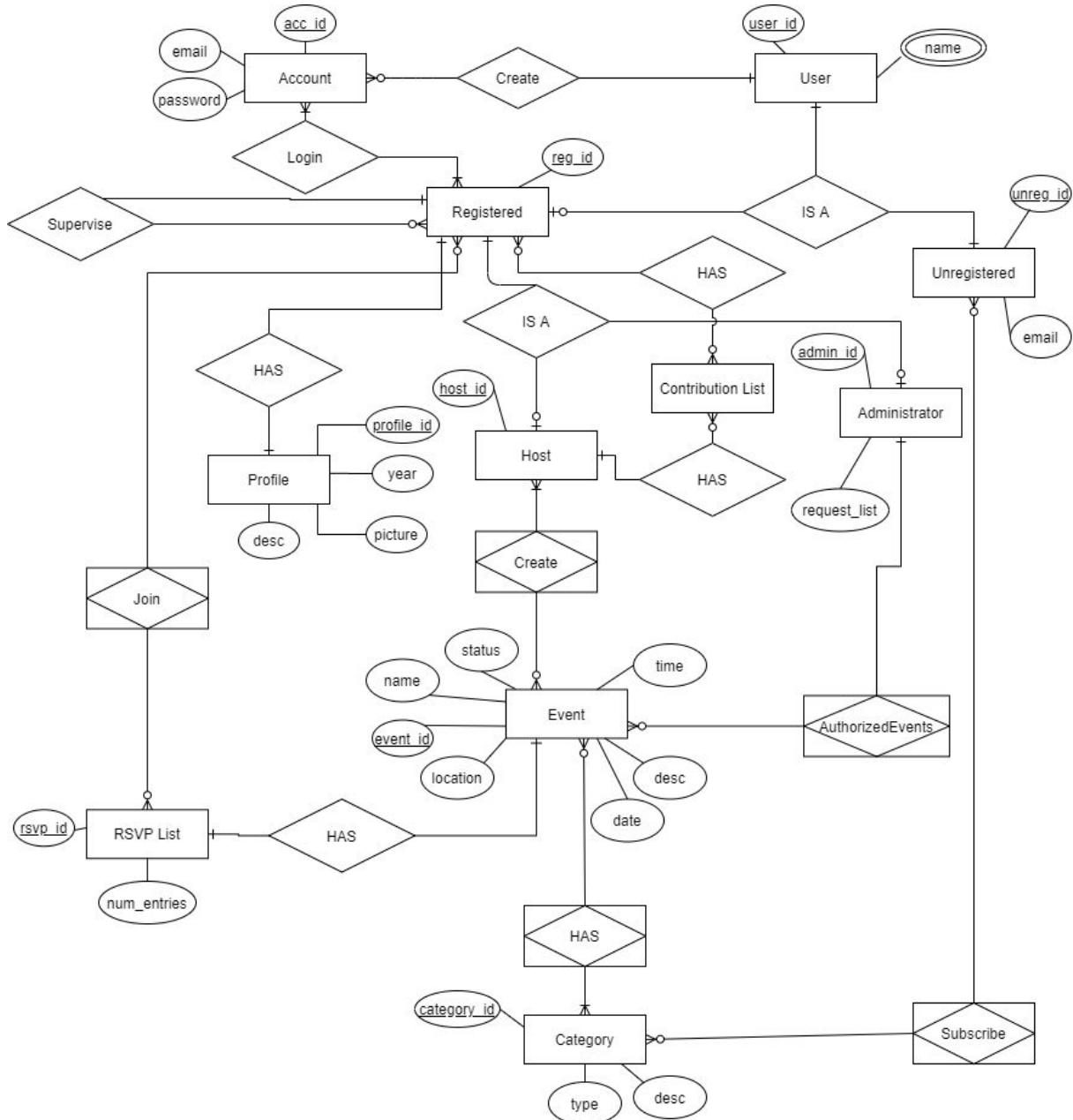
- Database Organization
 - Business Rules
 1. Unregistered Users can create at most one account on the application per SFSU email.
 2. Registered Users can log in into the application on multiple devices.
 3. An event shall only have at most one RSVP list.
 4. Multiple Registered Users can RSVP to multiple events.
 5. An event shall have at least one event host.
 6. Admins can approve/deny multiple event proposals.
 7. Registered Users can at most have one of each customizable option at a time.
 8. Registered Users can book/star multiple events.
 9. An event host can host multiple events.
 10. Registered Users can have multiple pending requests.
 - Entities and Attributes Relationship
 - Entities:
 - Account:
Acc_id: PK, e-mail, password
 - User:
User_id: PK, name, age
 - Registered:
reg_id:PK, username, points
 - Unregistered:
unreg_id:PK, email
 - Host:
host_id:PK, email
 - Event:
Event_id: PK, name, status, location, desc, date, time

- Profile:
profile_id: PK, year, picture, desc
- RSVP List:
rsvp_id: PK, event_id: FK, num_entries
- Administrator
admin_id: PK, request_list
- Category:
Category_id: PK, type, desc

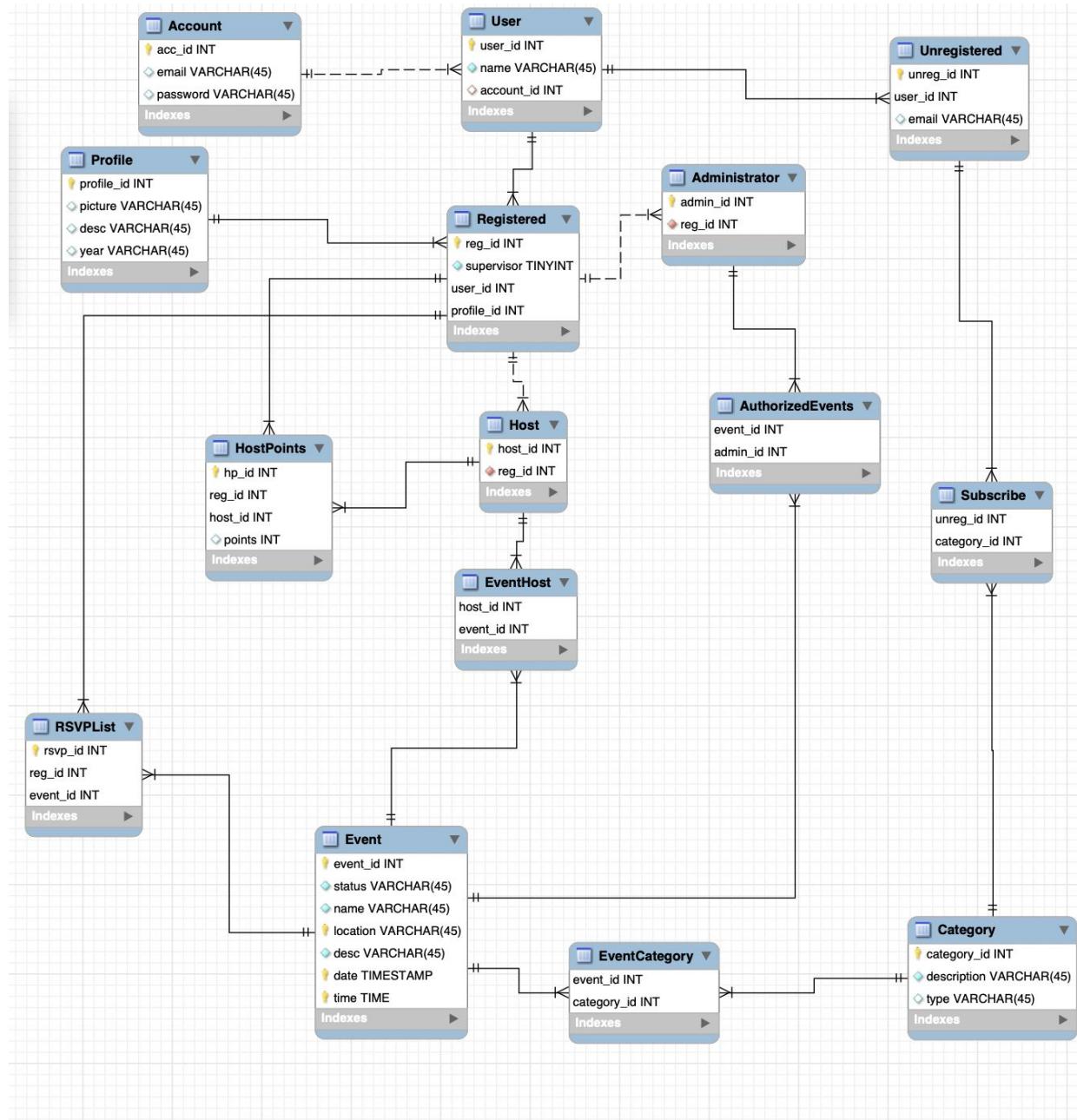
□ Relationships:

- Create
- Join
- RSVP
- HAS
- View
- AuthorizedEvent
- Login
- ISA
- Subscribe
- Supervise

Entity Relationship Diagram (ERD)



Database Model



- DBMS Used
MySQL will be the DBMS that our team would be utilizing, as it is one of the most commonly used DBMS.
- Media Storage
Any images or video will be stored in file systems. Media information will be kept using a table with media_id, attributes: media_id, type, file name, url and will be accessed through the server with the url and media_id.

- **Search/Filter Architecture and Implementation**

The algorithm will utilize substring to determine the events. Search items would be organized by categories, the date, relevancy and name. The user should be able to sort how they want the events to be searched and displayed. The database terms are the name of the event, event category, and date.

High Level APIs and Main Algorithms

Most of the endpoints require the user's unique id for authentication purposes and also to determine if the user is logged on to be able to make decisions, we need to check this before retrieving data from the database.

APIs

Login

A way of authenticating users before using features only available to registered users. This endpoint will take a username and an encrypted password in json format.

The backend server will retrieve information from a mysql database and compare the entered information and the information saved. If the information matches, the server will return “successful login”.

Register User

A way for users to register for an account. The backend should receive information in a JSON format. The JSON shall contain all information required to register an account. This information includes: email, password, first name, last name, and school year.

There would be multiple checks incorporated into this api, one of them will be checking that an email is not already in the database and checking that the user submitted a valid school year. The server will return “Registration Successful” and if not it will return the reason why it was not a valid request. All of this information will be in json format.

Create Event

A way for users to create an event. Registered Users can submit a request for an event, providing all essential information that pertains to the event. An event is required to have time, date, description, location, and category.

Once submission of an event request is done, it will have to go through the process of approval by the Administrators. Once submitted, the user may view the status of their event on their profile page, which would be

displayed under “My Events”. An email notification would be sent to the user regarding the status of their events one review is done by the Administrator.

Approve Event

Only users who are Administrators will be able to approve events. A request to approve an event will be submitted upon creation of each event. Manual review of the information provided along with verification of legitimacy of event location, compliance with campus guidelines (if location is on campus), and the Administrator’s discretion will determine whether an event will be approved.

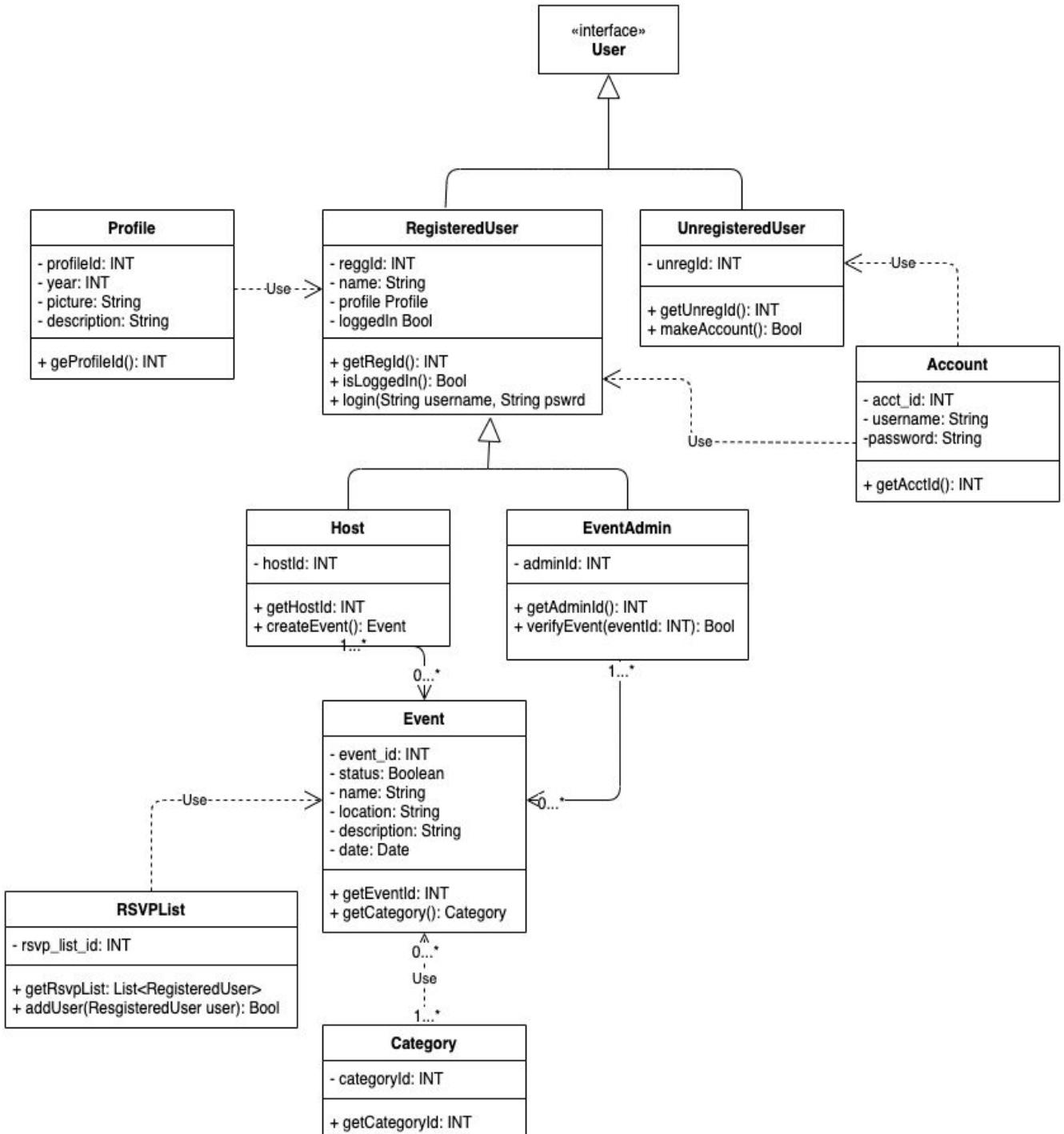
Post review, the Administrator will update the status of the event from Pending to Approved/Denied/Requires Revision. If an event requires revision, an email will be sent indicating the reason(s) why and possible additional feedback.

EventRSVP

Every event has an RSVP list showing all users that are planning to attend that specific event. Once an event is approved, the RSVP list function would become available to users to view and join. The RSVP list would display each user’s name.

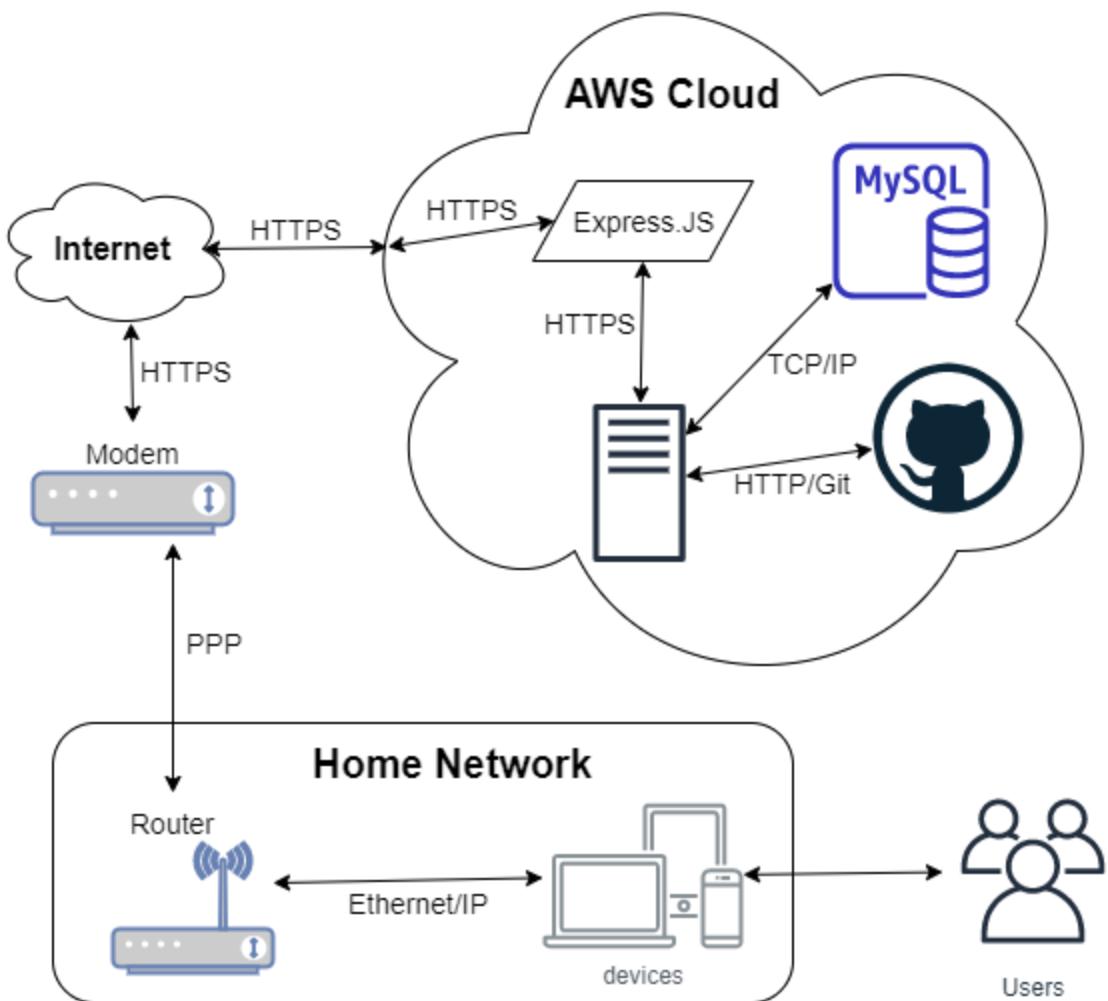
- Non-Trivial Algorithm or Process
 - **gettingRandomEvent:** This endpoint will return an event that we the user is not specifically requesting. The event will be chosen by checking previous events that the user has attended. We mainly care about the type of event therefore, we will be using the category as the key element to determine which event to show the user. On top of the category, we will also check for events that may be coming up soon and that have a good number of people attending. This will allow the user to make a decision based on people attending and previously shown interest.

High Level UML Diagram

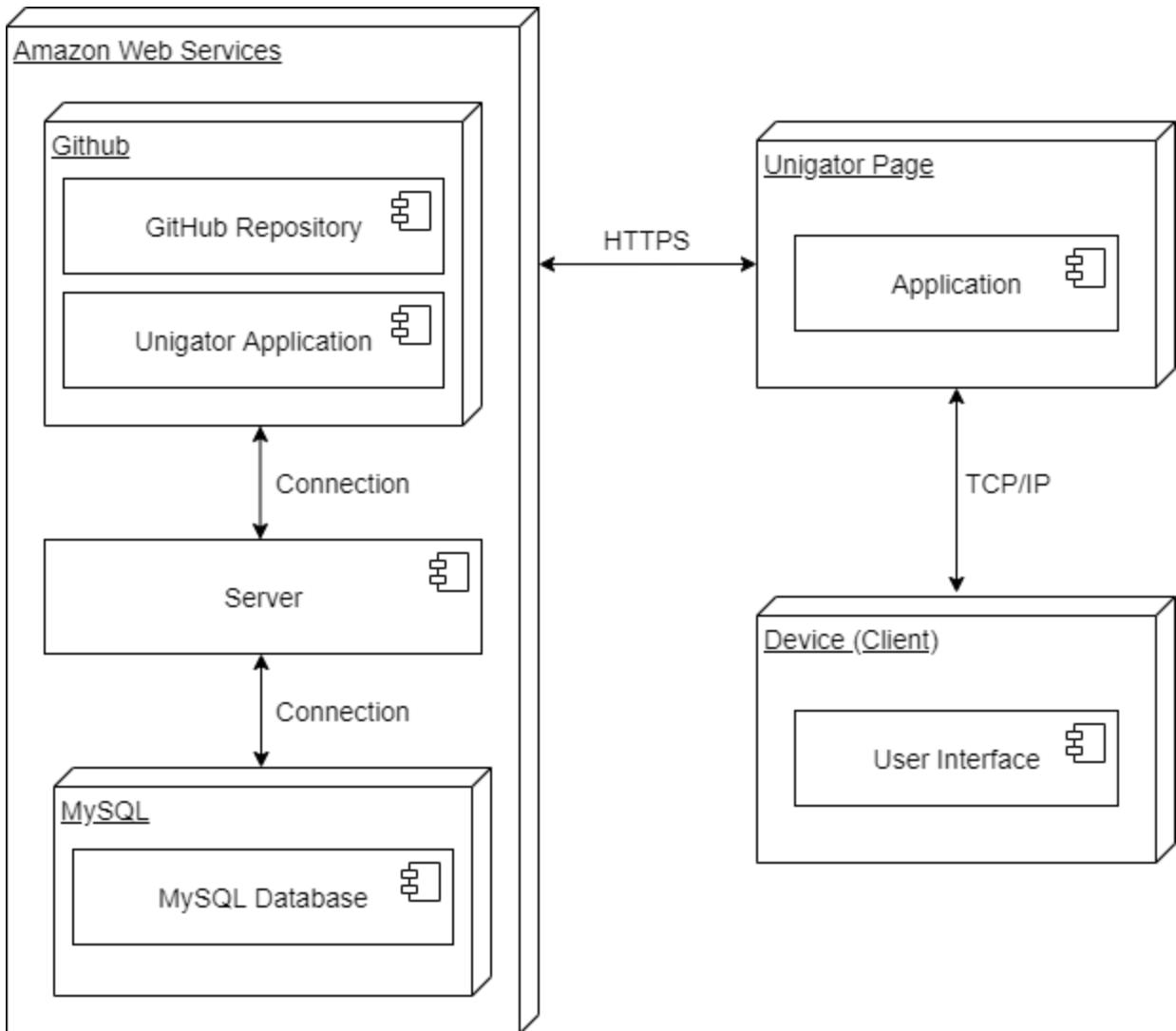


High Level Application Network and Deployment Diagrams

Application Networks Diagram



Deployment Diagram



Actual Key Risks of the Project

- Skill Risks
 - Practically everyone on our team is new to using cloud web services. Setting up a server/place to host our application was already challenging to start with. Some team members have little to no knowledge on javascript, as that is our chosen language, we all would need to learn and understand javascript and the frameworks that we will be utilizing for our application.
 - None of our team have any prior involvement in database management, so we would need to learn all the basics of using a database and how to deploy it to our application. Some members are currently taking an Introduction to Database Systems class in order to understand the concept of database.
- Schedule Risks
 - Each member of the team has varying schedules so it was very difficult to accommodate everyone's preferred meeting time. Meeting together as a whole time was almost impossible, even if we were to meet up as a whole, it would only last a short time as most of the time some of our members had classes or other business to deal with.
 - With the current coronavirus situation going on around the world and especially in California, it has greatly affected everyone's schedule. The campus is closed and our weekly in class meetings were moved to an online format and we are told to adhere to social distancing. Meeting up in person as a team is extremely difficult or even impossible at this current situation.
 - We have moved our weekly meetings to an online voice chatting service, "Discord". Since our meetings are now conducted with voice only, it's difficult to clearly voice out an opinion or talk about one's ideas. Not everyone on the team has access to a mic all the time and everyone is available at different intervals of time.
- Technical Risks
 - Due to the coronavirus situation plaguing the world at the moment, our university has suspended all types of events and gatherings in our university. This poses a problem to our application as our application is about events happening on campus. We hope the situation improves in

the coming months so that our application can take information about events by the campus.

- For most of the team members, this would be their first time working with a team with each member contributing to the code. Understanding Github and what it means to code together is something we need to work on and understand more.
- Teamwork Risks
 - As it currently stands, it's difficult and confusing to clearly communicate with each other on what tasks each member would be working on. Currently we use Stack as our main way of communicating and assigning tasks. Some members are occasionally unavailable, leaving the team blank on their progress of a certain task.
 - Making sure the team clearly understands the tasks and assignments they are assigned is based purely off trust and belief at the moment.
- Legal/Content Risks
 - Our application is based on information and data of the university's events and its students and staff, getting permission to utilize data of the campus would prove difficult. Our system plans to utilize our university's email as the main way to register and login, getting approval for this type of system might pose a risk to our application.

Project Management

We tried to meet up in person or online as much as possible, but it has been proven difficult with the world's current predicament of coronavirus. Initially, our primary way of communication was through Slack, but we needed a way to verbally voice out our opinions and collaborate. We ultimately moved over to Discord, as that platform provided us with voice chatting and screen sharing, giving us the opportunity to work on something together while efficiently communicating with each other. Unfortunately, not everyone can be on at the same time, so it was pretty difficult to discuss anything as a whole team. Our team is most likely going to stick with this platform to conduct any work together as a team, as it seems like meeting in person is highly unlikely at this moment.

Making sure everyone on the team understood what Milestone 2 was about and how it matters as a whole was important. Asking questions about any concepts that we are unfamiliar with and brainstorming as a team was crucial in this Milestone. Clearly knowing what tasks each member of the team is assigned to is important, to avoid any confusion or similar work being done on the same section.

Milestone 2 can be divided into sections, and assigning sections as tasks to each team member. For example, the UI Mockup and Storyboard section is handled by the front end team, and the Database Organization section is handled by the back end team. We worked together as a team for the rest of the other sections. For now we can only base our hopes in a team member accomplishing their tasks on trust and communication. We would need a new and concrete way to accurately assign tasks to members without any confusion. We are currently to utilize Trello, as that platform seems like a great way to keep track of the team's tasks.

SW Engineering - CSC648/848

Spring 2020



Team 03

Milestone 3

Lionel Wong	lwong7@mail.sfsu.edu	Team Lead
Jorge Landaverde	jlandaverde@mail.sfsu.edu	Lead Backend Engineer
Gordon Lam	glam1@mail.sfsu.edu	Backend Engineer
Mitul Savani	msavani@mail.sfsu.edu	Lead Frontend Engineer/Git Master
Kevin Huang	khuang2@mail.sfsu.edu	Frontend Engineer
Jack Kower	jkower@mail.sfsu.edu	Frontend Engineer

Milestone	Date
Milestone03 Version02	5/8/2020
Milestone03 Version01	4/23/2020
Milestone02 Version02	4/15/2020
Milestone02 Version01	3/25/2020
Milestone01 Version02	3/12/2020
Milestone01 Version01	3/5/2020

Table Of Contents

I.	Data Definitions V3	2-5
II.	Functional Requirements V3	6-9
III.	Wireframes Based on your Mockups/Storyboards V2	10-18
IV.	High level database architecture and organization V2	19
V.	High Level Diagrams V2	20-22
VI.	Customer Feedback	23

Data Definitions V3

Data Entities	Details
User	<p>Description</p> <ul style="list-style-type: none"> - A general user that is browsing the internet. <p>Requirements</p> <ul style="list-style-type: none"> - Able to browse the internet <p>Sub-Data items</p> <ul style="list-style-type: none"> - name
Account	<p>Description</p> <ul style="list-style-type: none"> - An account that is unique to a user, which will be registered with email and password. <p>Requirements</p> <ul style="list-style-type: none"> - Account shall only be created with an SFSU email <p>Sub-Data items</p> <ul style="list-style-type: none"> - email - password
Unregistered User	<p>Description</p> <ul style="list-style-type: none"> - Users that are not registered with the site, able to browse through events and look at the information, but are unable to utilize some functions of Unigator, such as bookmarking the event and sharing it. <p>Requirements</p> <ul style="list-style-type: none"> - Can sign-up to become a registered user using their SFSU email. <p>Sub-Data items</p> <ul style="list-style-type: none"> - email
Registered User	<p>Description</p> <ul style="list-style-type: none"> - Users that are registered with the site with their SFSU email. Able to browse through events and save them into their own personal calendar and share the events with social media. Able to update their status whether they are attending the event. Every registered user will have a display name.

	<p>Requirements</p> <ul style="list-style-type: none"> - Able to report another user. <p>Sub-Data items</p> <ul style="list-style-type: none"> - name - email - password
Event	<p>Description</p> <ul style="list-style-type: none"> - Activities that are displayed on Unigator that takes place in SFSU. Every event will have to have a name and important information regarding the event; such as who the event host is, location, time, etc. An event can be of single occurrence or recurring. <p>Requirements</p> <ul style="list-style-type: none"> - An event host is needed - Needs to be approved by an Administrator - Needs to fall under at least one category. <p>Sub-Data items</p> <ul style="list-style-type: none"> - name - status - location - time - desc - date
Event Host	<p>Description</p> <ul style="list-style-type: none"> - A registered user that is hosting/registering a(n) event, and provides all the essential information of the event; such as date, time, location and description. Will also be the contact listed for said event. <p>Requirements</p> <ul style="list-style-type: none"> - Be able to provide necessary information about the event when requested. - Is able to create an event when desired <p>Sub-Data items</p> <ul style="list-style-type: none"> - name - email

RSVP List	<p>Description</p> <ul style="list-style-type: none"> - A list showing all registered users who plan to attend an event. A user can RSVP to multiple events even if there is a time conflict. <p>Requirements</p> <ul style="list-style-type: none"> - Based on its own specific event. <p>Sub-Data items</p> <ul style="list-style-type: none"> - num_entries
Administrator	<p>Description</p> <ul style="list-style-type: none"> - A higher privilege user that oversees the site. Able to decide whether an event is to be listed or not. Able to ban users and send warnings. Moderates the site by utilizing special permissions only available to an admin account. Maintains the integrity of the site while keeping sure everything is in order. <p>Requirements</p> <ul style="list-style-type: none"> - Able to do anything a registered user can. - Able to view request list. - Able to respond to requests. - Able to request changes from a user/host. <p>Sub-Data items</p> <ul style="list-style-type: none"> - name - email - password
Category	<p>Description</p> <ul style="list-style-type: none"> - The categories that an event falls under. For example, an art gallery will be put under the 'Art' category. A recruiting event from a technology company would be under the 'Technology' category. <p>Requirements</p> <ul style="list-style-type: none"> - Choosable from predefined list. <p>Sub-Data items</p> <ul style="list-style-type: none"> - type - desc

Activity Points	<p>Description</p> <ul style="list-style-type: none"> - Every registered user will have their own wallet of Activity Points. These are points that act as a currency to be used in a point shop. Each profile represents a unique representation of a user. The profile can be customized with various options, options that can be purchased from the point shop page with earned activity points. Activity points are earned by interacting with the site and being active, example being checking the RSVP list and attending the event, bookmarking an event, and creating an event etc. <p>Requirements</p> <ul style="list-style-type: none"> - Able to be earned through specified activities on the website. - Able to be spent in the points shop. <p>Sub-Data items</p> <ul style="list-style-type: none"> - points
Host Points	<p>Description</p> <ul style="list-style-type: none"> - Every Host has a list of users that have attended any of their events. The host will be awarding points during their meetings for attendance or participation. These points will be used to reward attendees and increase events' attendees count. The host can enable this feature and these points are only specific to this specific hosts' store. <p>Requirements</p> <ul style="list-style-type: none"> - Is enabled by Event Host <p>Sub-Data items</p> <ul style="list-style-type: none"> - points

Functional Requirements V3

Priority 1

Requirements
1) Unregistered Users
1.1) Unregistered Users shall be able to sign up with their SFSU email.
1.2) Unregistered Users shall be able to browse through the event list.
1.3) Unregistered Users shall be able to see all the essential information about an event.
1.4) Unregistered Users shall be able to filter the event list.
1.5) Unregistered Users shall be able to view past events.
1.6) Unregistered Users shall be able to subscribe to Category for updates
2) Registered Users
2.1) Registered Users shall be able to log in with their account.
2.2) Registered Users shall be able to log out of their account.
2.3) Registered Users shall have a name (first name and last name).
2.4) Registered Users shall be able to browse through the event list.
2.5) Registered Users shall be able to filter the event list.
2.6) Registered Users shall be able to see all the essential information about the event.
2.7) Registered Users shall be able to view the email of event hosts.
2.8) Registered Users shall be able to view past events.
2.9) Registered Users shall be able to bookmark/star an event.
2.10) Registered Users shall be able to un-bookmark/un-star an event.
2.11) Registered Users shall be able to RSVP to an event.

2.12) Registered Users shall be able to un-RSVP to an event.
2.13) Registered Users shall be able to view an event's RSVP list.
2.14) Registered Users shall be able to edit their profile.
2.15) Registered Users shall be able to view a user's profile.
2.16) Registered Users shall be able to become host.
2.17) Registered Users shall be able to earn activity points.
2.18) Registered Users shall be able to view the points shop.
2.19) Registered Users shall be able to purchase from the points shop.
2.20) Registered Users shall be able to switch to previous options in the points shop.
3) Administrators
3.1) Administrators shall be able to log in with their account.
3.2) Administrators shall be able to log out of their account.
3.3) Administrators shall be able to view event requests.
3.4) Administrators shall be able to approve an event request.
3.5) Administrators shall be able to reject an event request.
3.6) Administrators shall be able to remove an event.
3.7) Administrators shall be able to see all the information about an event.
3.8) Administrators shall be able to view the email of event hosts.
3.9) Administrators shall be able to view past events.
3.10) Administrators shall be able to ban a registered user.
3.11) Administrators shall have all purchases in the point shop unlocked.
3.12) Administrators shall be able to request/demand changes to an event.
3.13) Administrators shall have all the functionality of a registered user.

4) Event Host

- 4.1) Event Host shall be a registered user.
- 4.2) Event Host shall be able to create an event.
- 4.3) Event Host shall be able to edit the event that they are hosting.
- 4.4) Event Host shall be able to remove the event that they are hosting.
- 4.5) Event Host shall have the same functionality as a registered user.
- 4.6) Event Host shall be able to enable host points feature.
- 4.7) Event Host shall be able to grant RSVPed users host points

Priority 2

Requirements

1) Unregistered Users

- 1.1) Unregistered Users shall be able to view the points shop.

2) Registered Users

- 2.1) Registered Users shall be able to request to change their name.
- 2.2) Registered Users shall be able to report a(n) event.
- 2.3) Registered Users shall be able to report a(n) user.
- 2.4) Registered Users shall be able to utilize the personal calendar function.
- 2.5) Registered Users shall be able to share with their social media.

3) Administrators

- 3.1) Administrators shall be able to view all requests from users to change their username.
- 3.2) Administrators shall be able to approve a user's request to change their username.

- | |
|--|
| 3.3) Administrators shall be able to deny a user's request to change their username. |
| 3.4) Administrators shall be able to change any registered user's username. |
| 3.5) Administrators shall be able to request a user make changes to their profile. |
| 3.6) Administrators shall be able to view reported events. |
| 3.7) Administrators shall be able to view reported users. |

4) Event Host

- | |
|--|
| 4.1) Event Host shall be able to request special changes from the Administrator. |
| 4.2) Event Host shall be able to transfer/assign a new event host. |

Priority 3

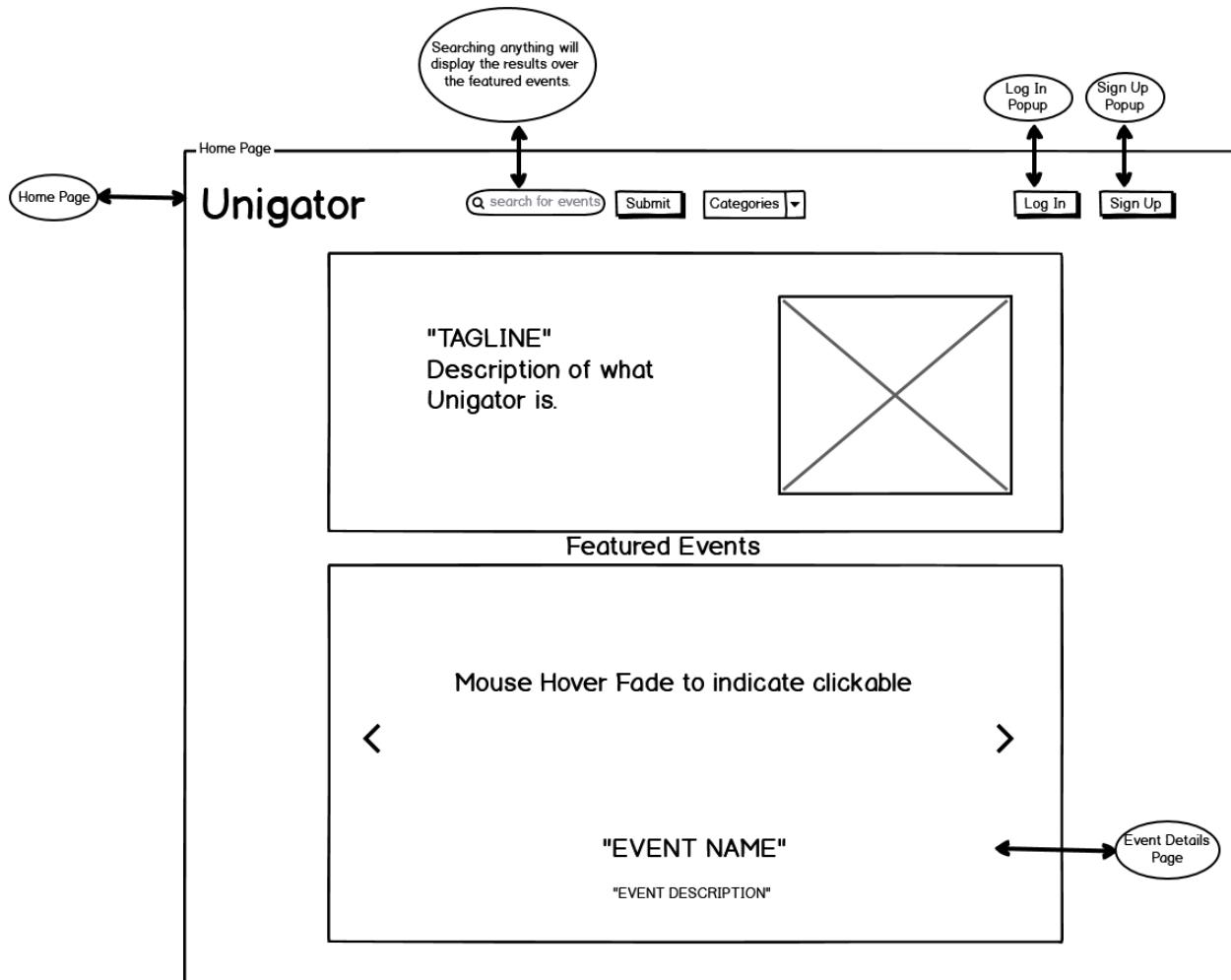
1) Unregistered Users

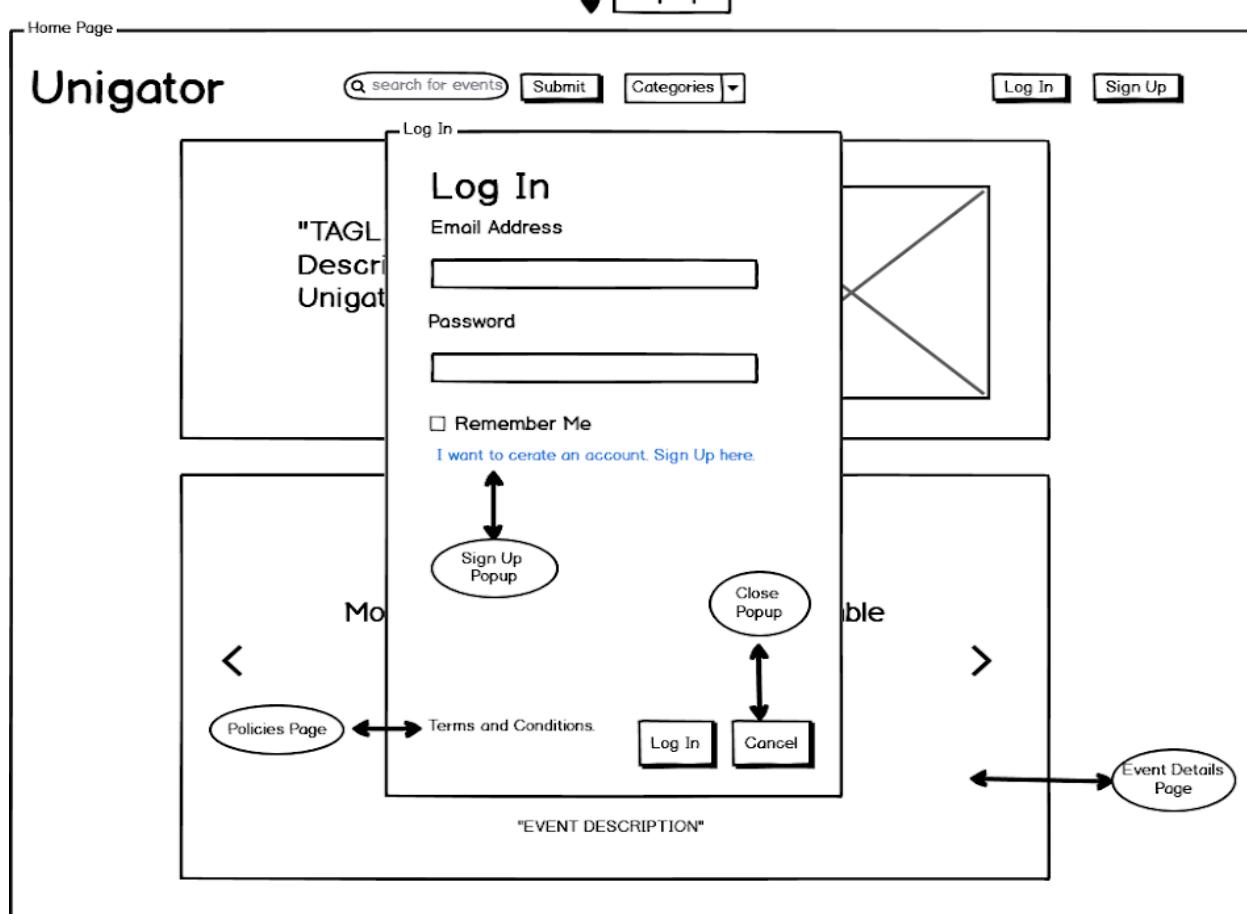
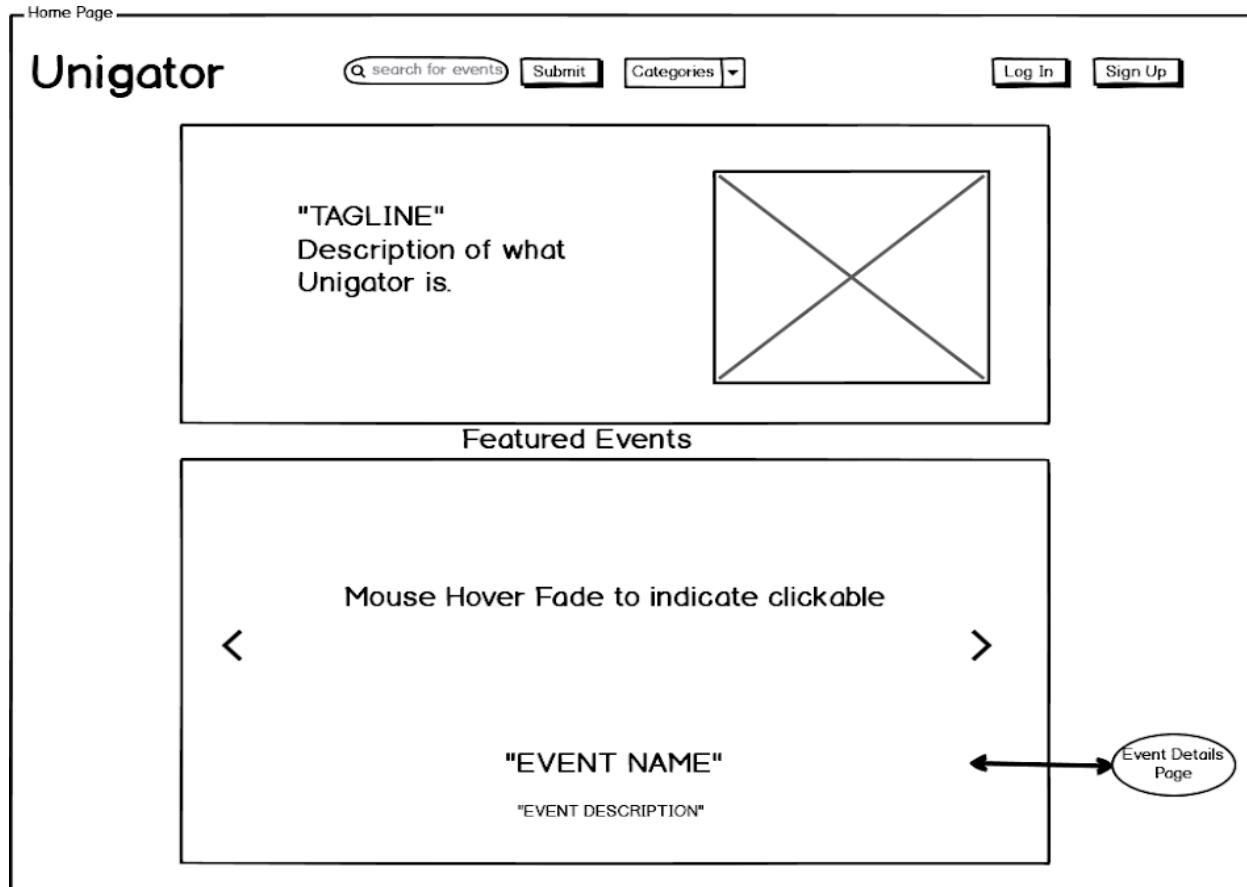
- | |
|---|
| 1.1) Unregistered Users shall be able to make a request to an Admin to create an account without an SFSU email. |
|---|

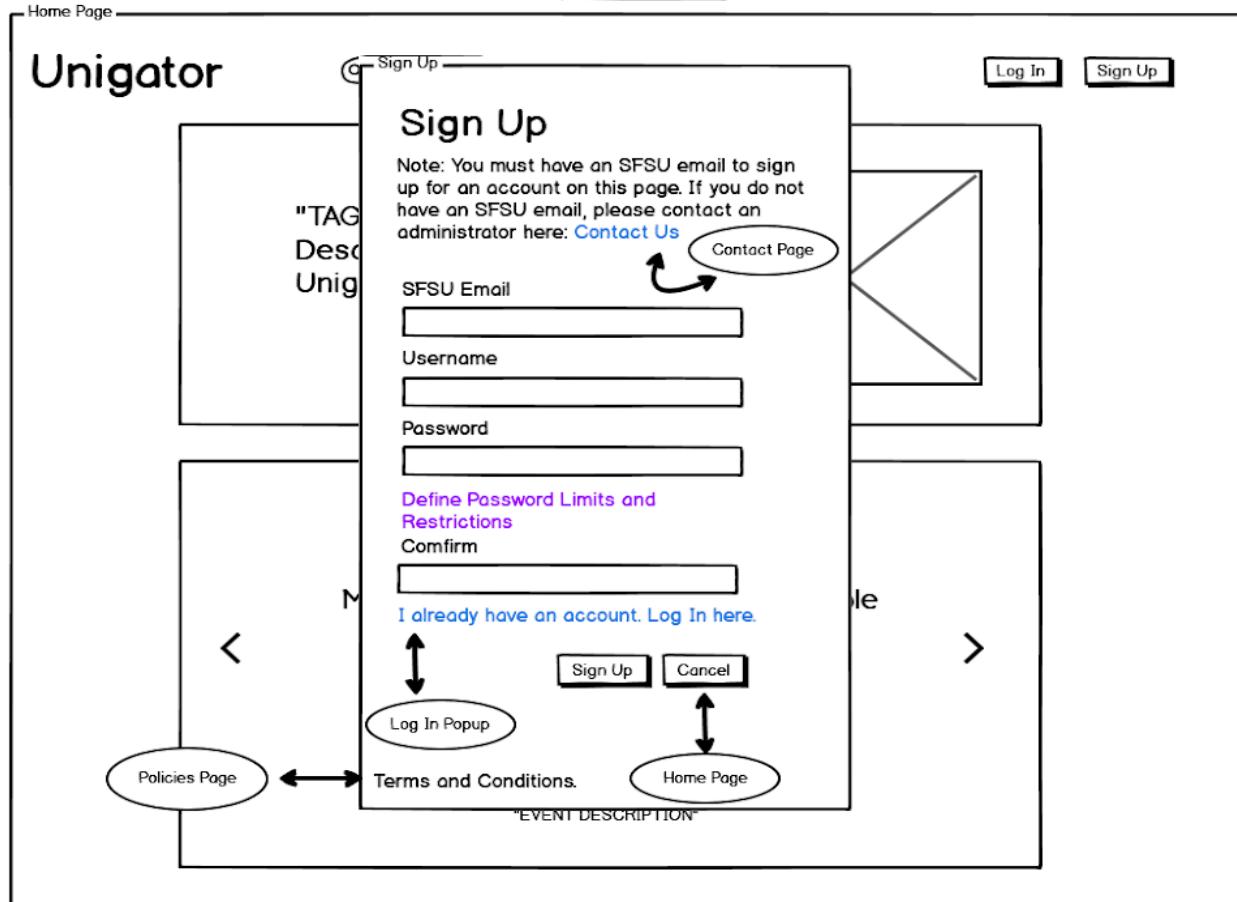
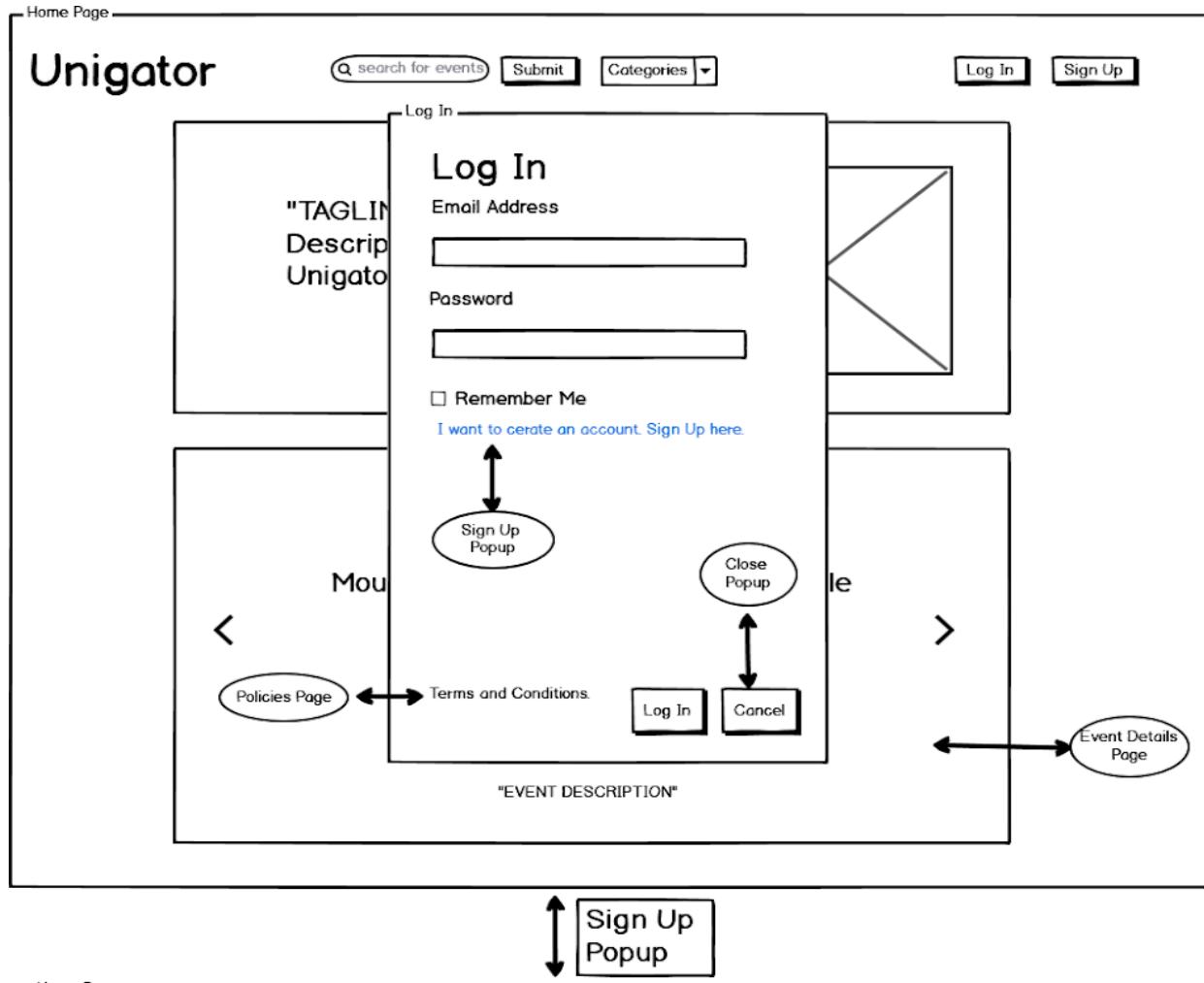
3) Administrators

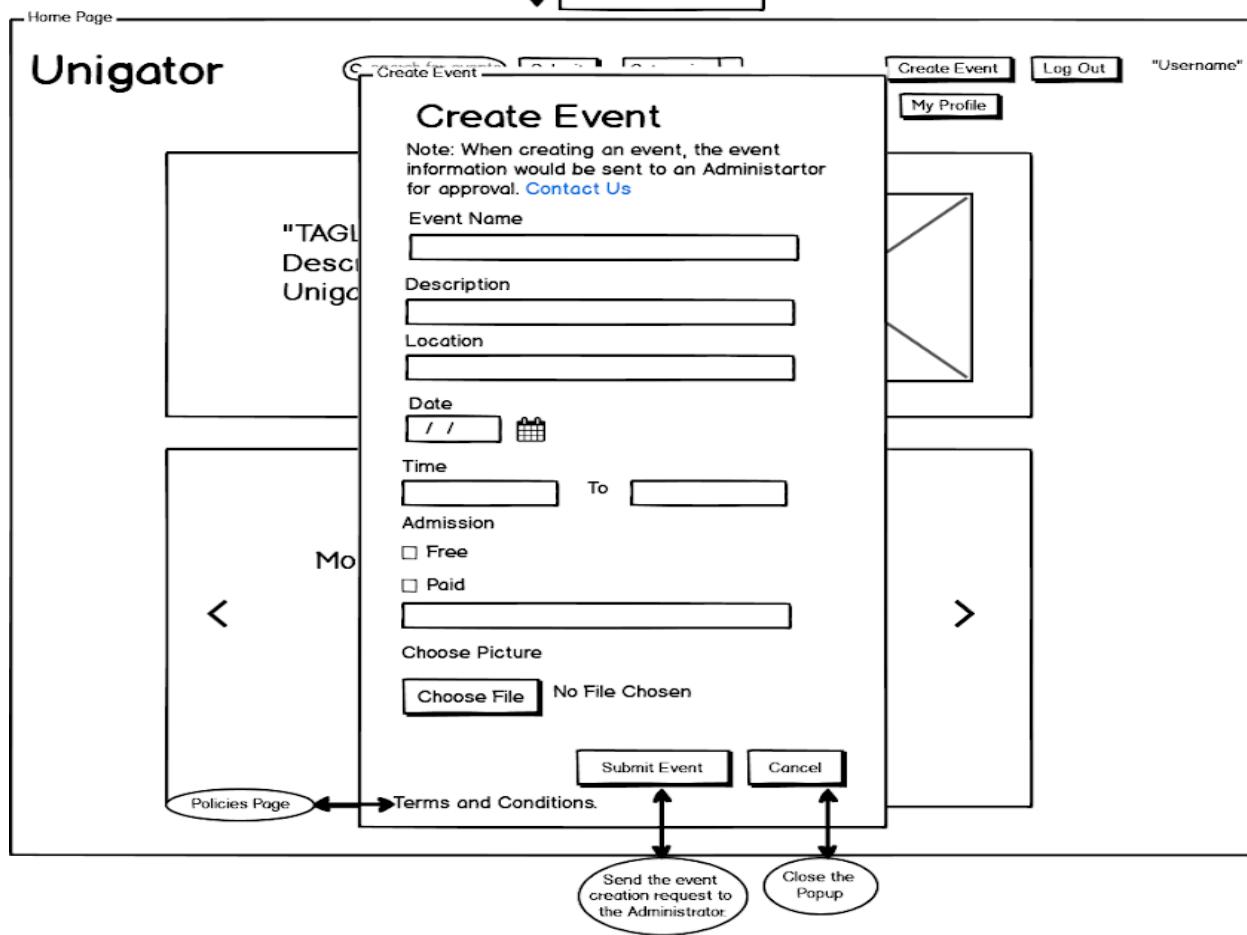
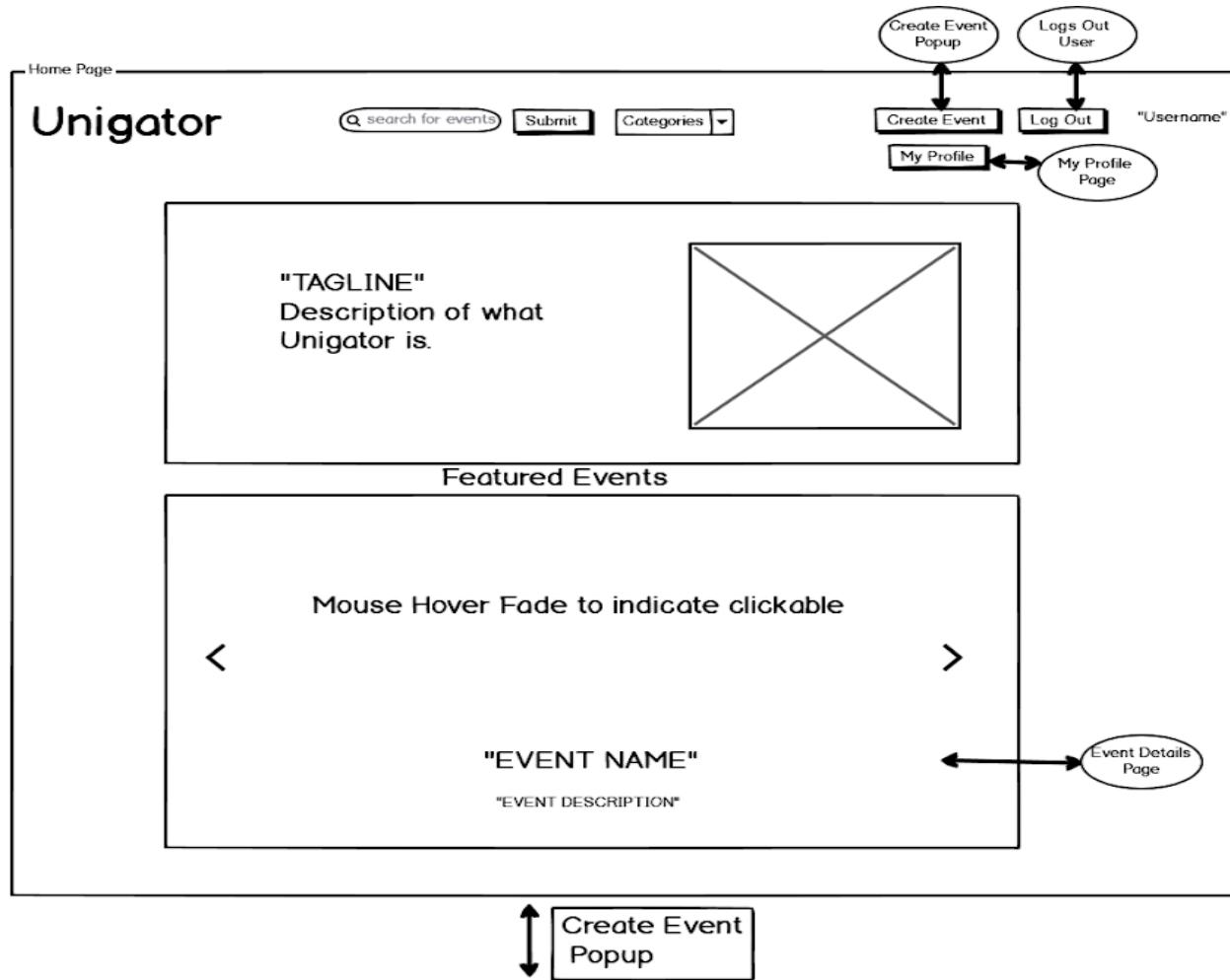
- | |
|---|
| 3.1) Administrators shall be able to view all requests to use a non-SFSU email to register. |
| 3.2) Administrators shall be able to approve a request to use a non-SFSU email to register. |
| 3.3) Administrators shall be able to deny a request to use a non-SFSU email to register. |

Wireframes Based on your Mockups/Storyboards V2









Home Page

Unigator

search for events Submit Categories Log In Sign Up

"TAGLINE"
Description of what Unigator is.

Featured Events

Mouse Hover Fade to indicate clickable

< >

"EVENT NAME"
"EVENT DESCRIPTION"

Event Details Page

Event Details Page

Event Details

Unigator

search for events Submit Categories Create Event Log Out "Username"

"My Profile" My Profile Page

Event Details

"Event Name"
"Date" "Time" "Location"

Save Share Report Reports the Event to an Admin

Event Host: "Name"
Contact: "Email"
Phone Number: "#"

RSVP

Event Description

Description of a specific event.

Event Details

Unigator

search for events Submit Categories ▾ Create Event Log Out "Username"

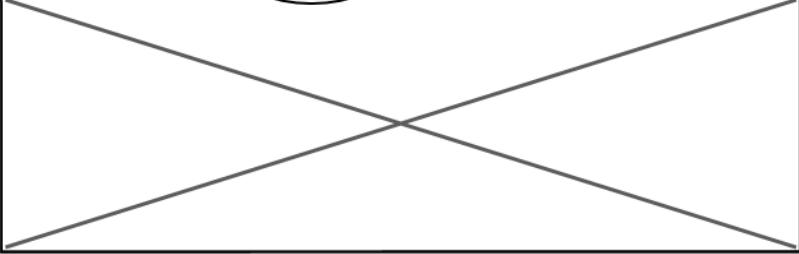
My Profile My Profile Page

Event Details

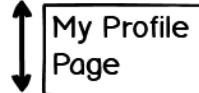
"Event Name"
"Date" "Time" "Location"
Save Share Report Reports the Event to an Admin

Event Host: "Name"
Contact: "Email"
Phone Number: "#"

RSVP



Event Description
Description of a specific event.



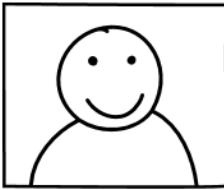
My Profile

Unigator

search for events Submit Categories ▾ Create Event Log Out "Username"

My Profile Edit profile Store Store Page

Welcome, "NAME"



User Information

Username:
Email:
School Year:
Points Accumulated:
Preferred Categories:
Description:

My Events

"SHOWS EVENT USER SAVED AND RSVPed"

↓
Clicking any of these will prompt the user to share.

↓

↓

↓

↓

My Profile

Unigator

search Submit Categories ▾ Create Event Log Out "Username"

My Profile Edit profile Store

Welcome, "NAME"



User Information

Username:
Email:
School Year:
Points Accumulated:
Preferred Categories:
Description:

My Events

"SHOWS EVENT USER SAVED AND RSVPED"

f i t



Store

Unigator

search Submit Categories ▾ Create Event Log Out "Username"

My Profile

"Amount of Points User has" \$

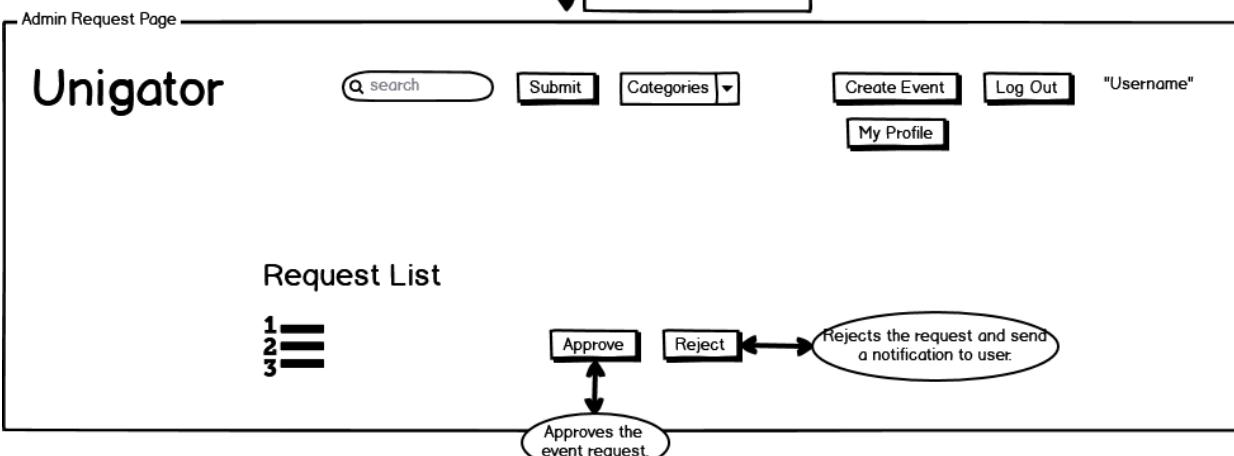
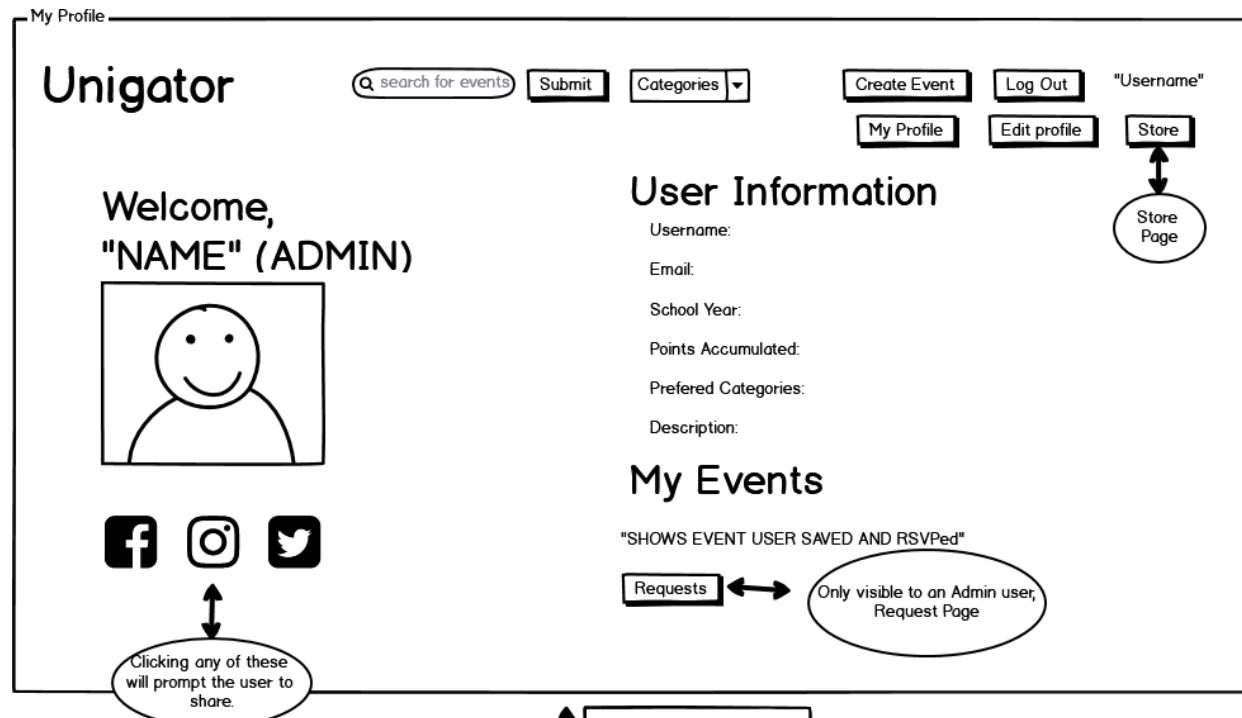
Unigator Store ⓘ

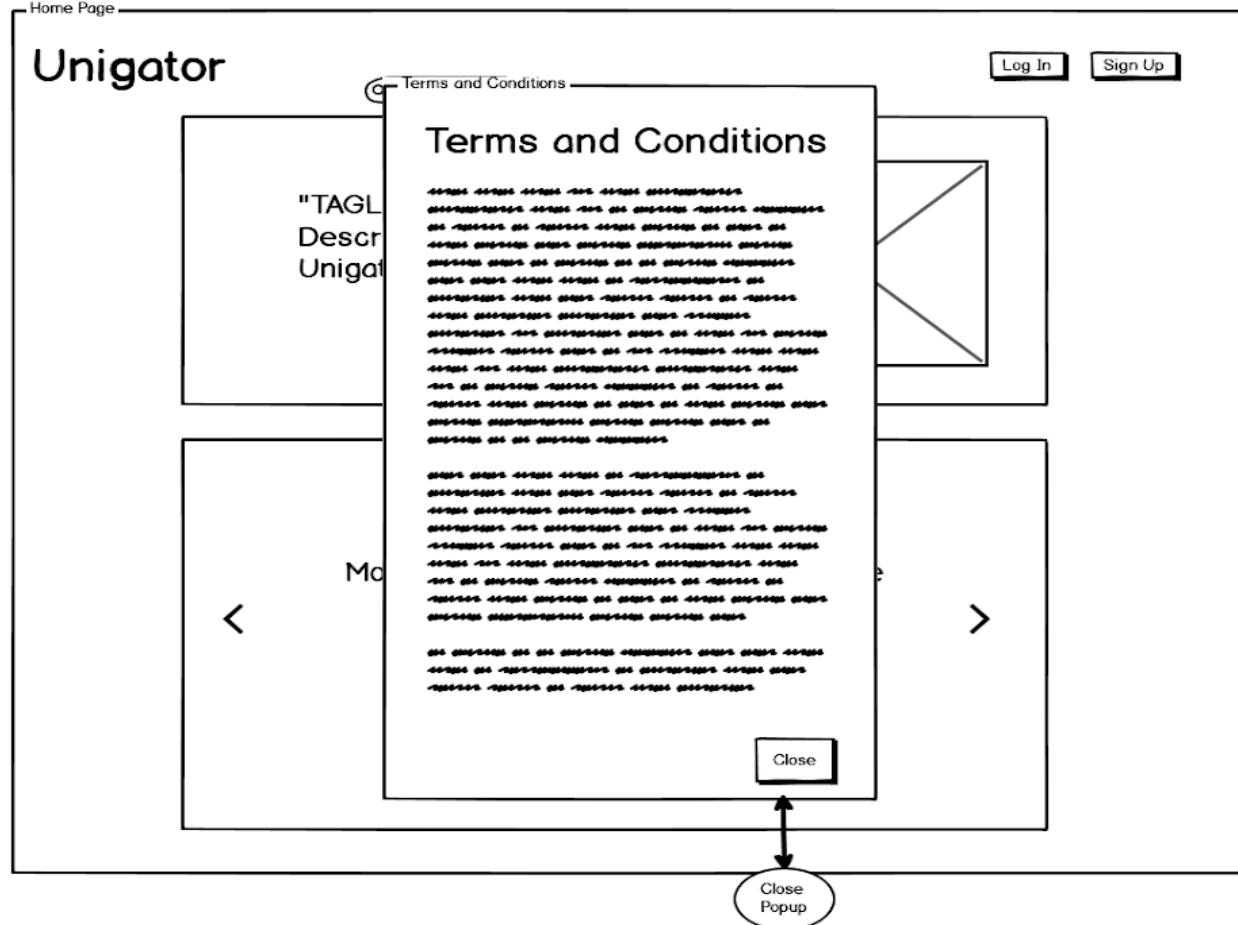
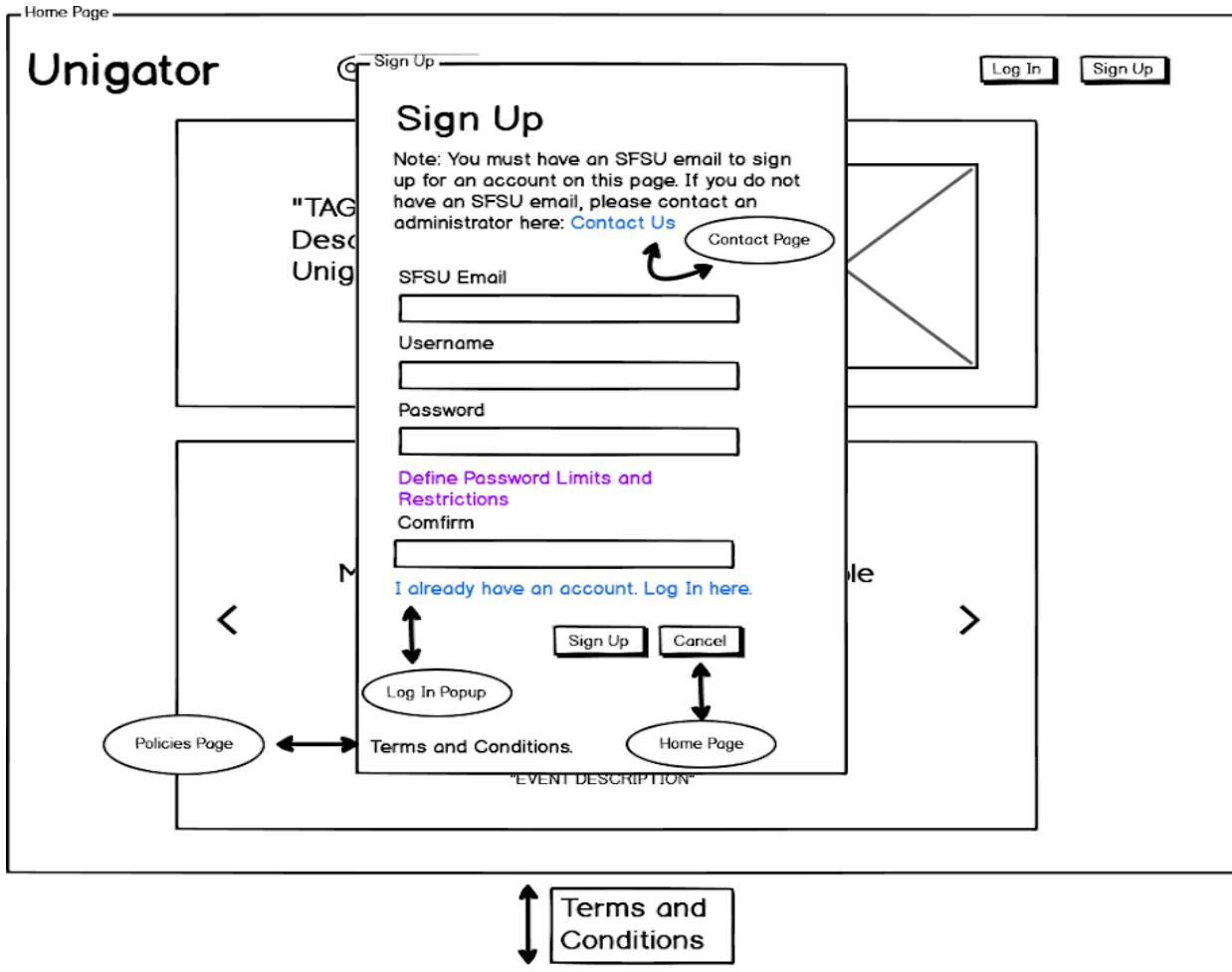
Details of what the store does.

Description of items

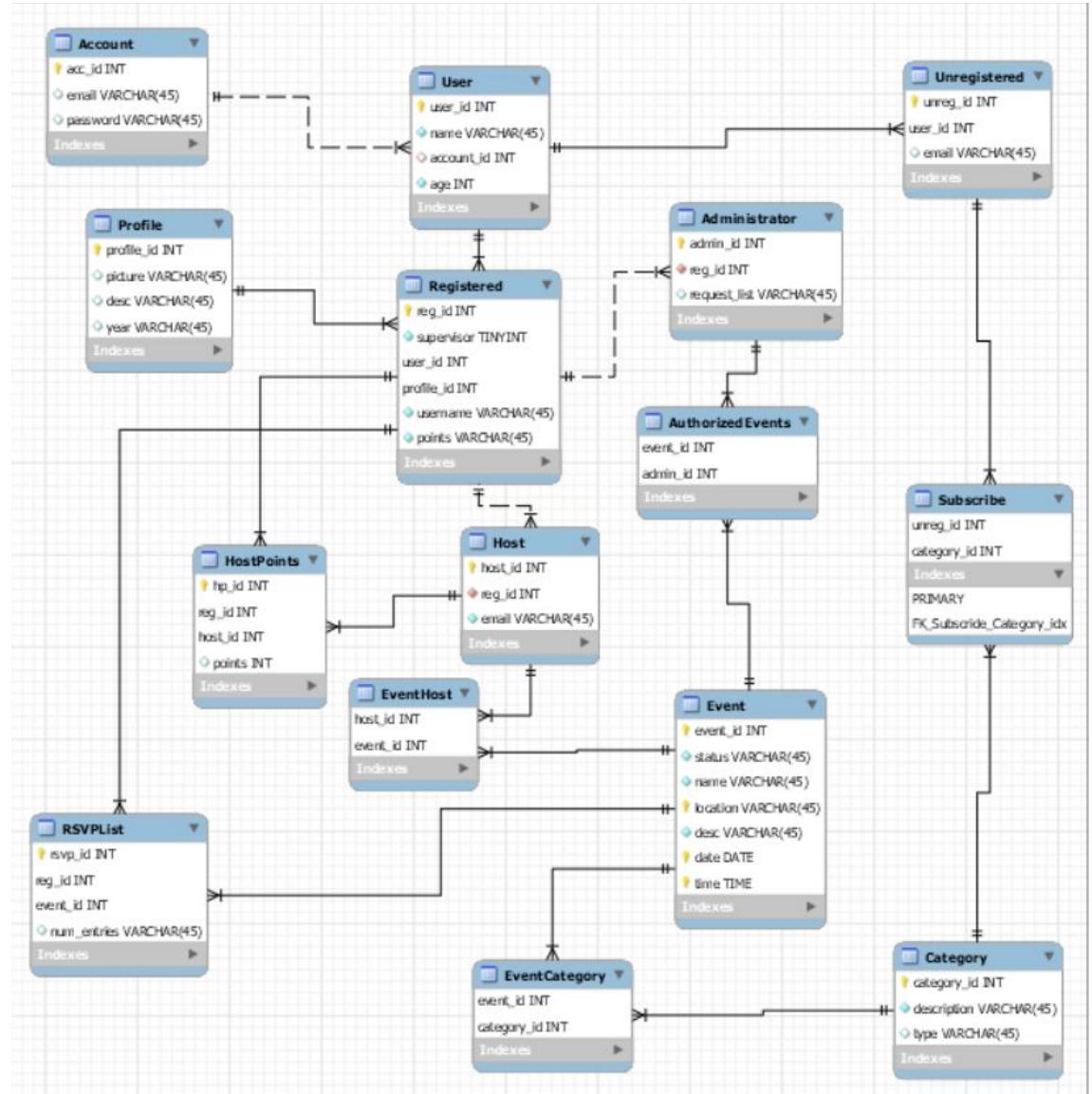
An example of what is being displayed. Purchase \$	An example of what is being displayed. Purchase \$	An example of what is being displayed. Purchase \$
An example of what is being displayed. Purchase \$	An example of what is being displayed. Purchase \$	An example of what is being displayed. Purchase \$





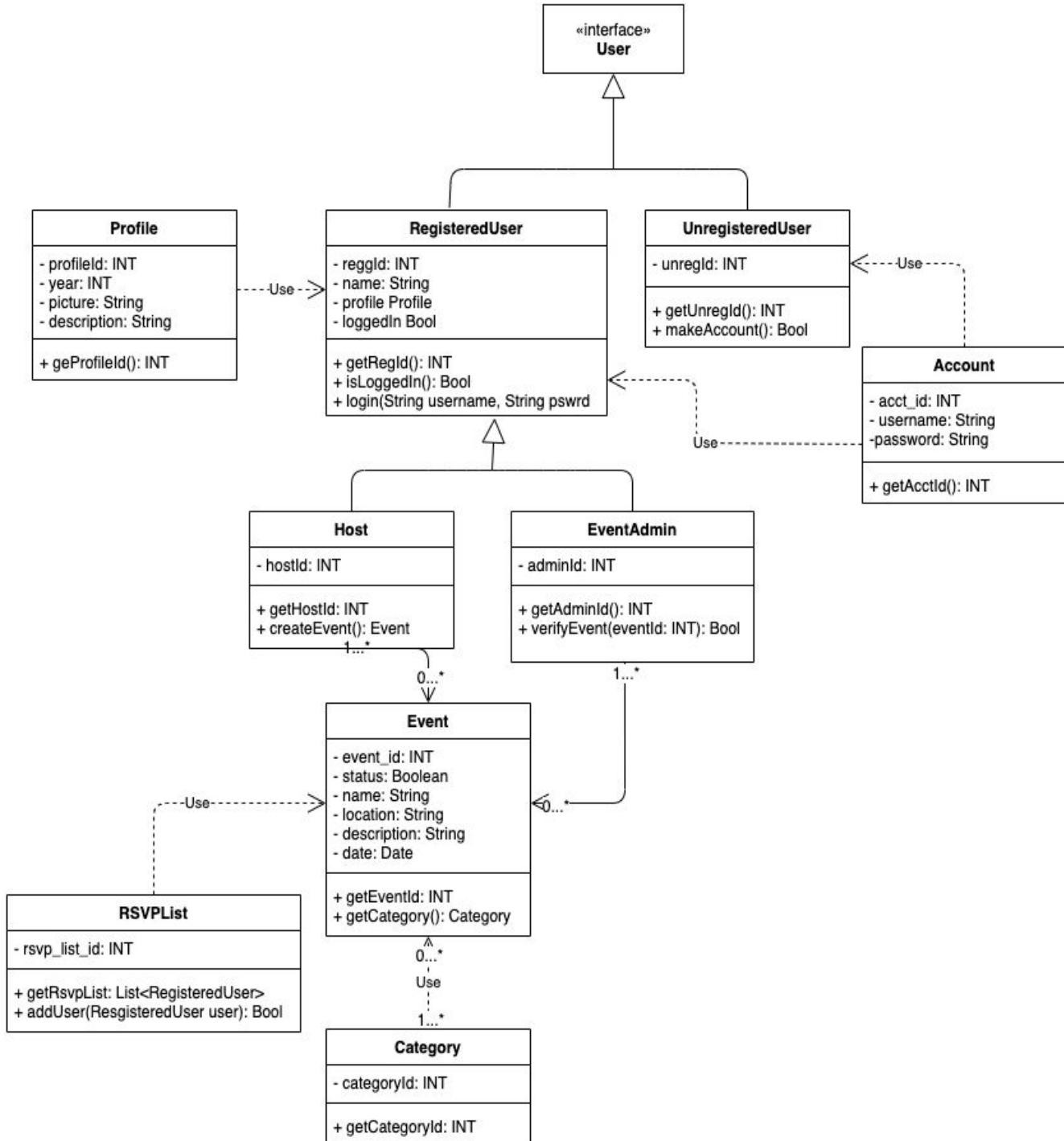


High level database architecture and organization V2

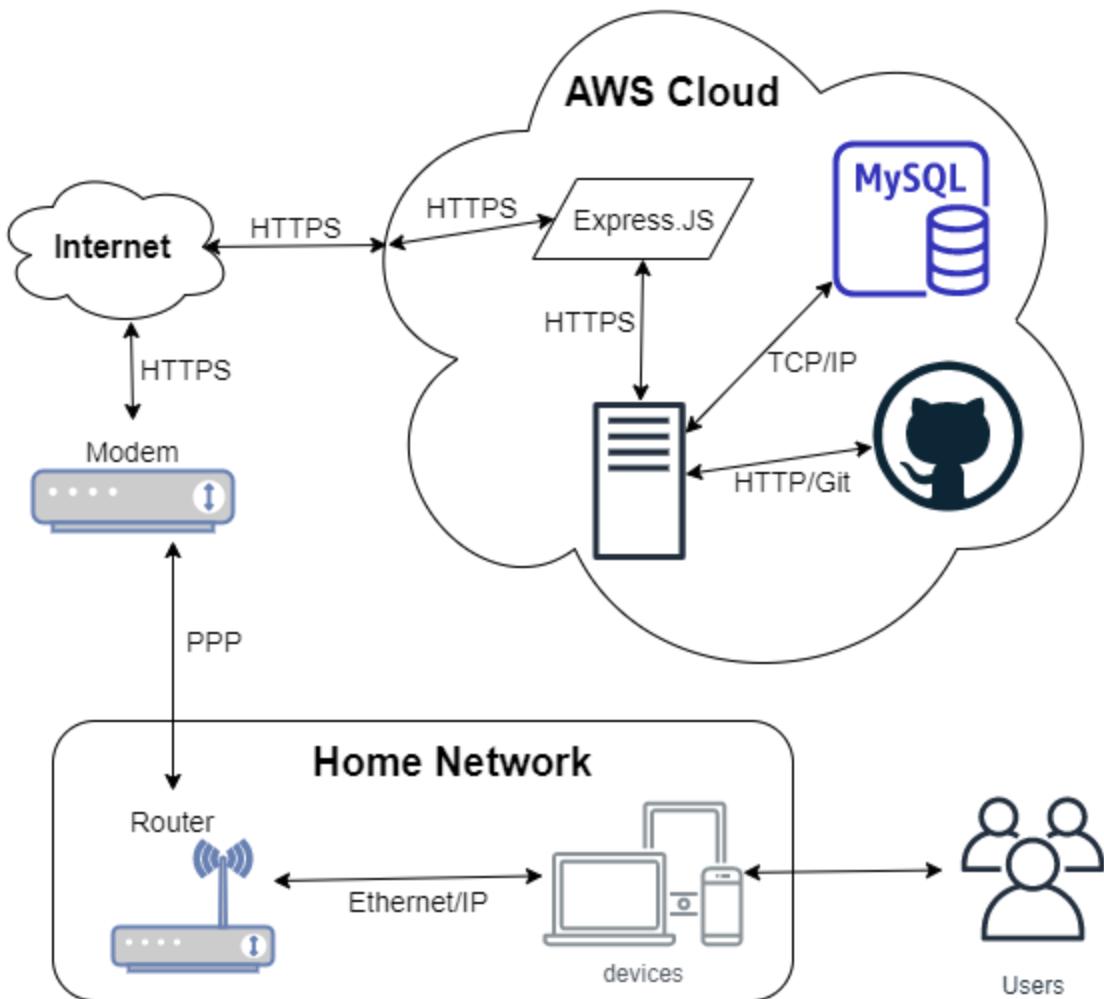


High Level Diagrams V2

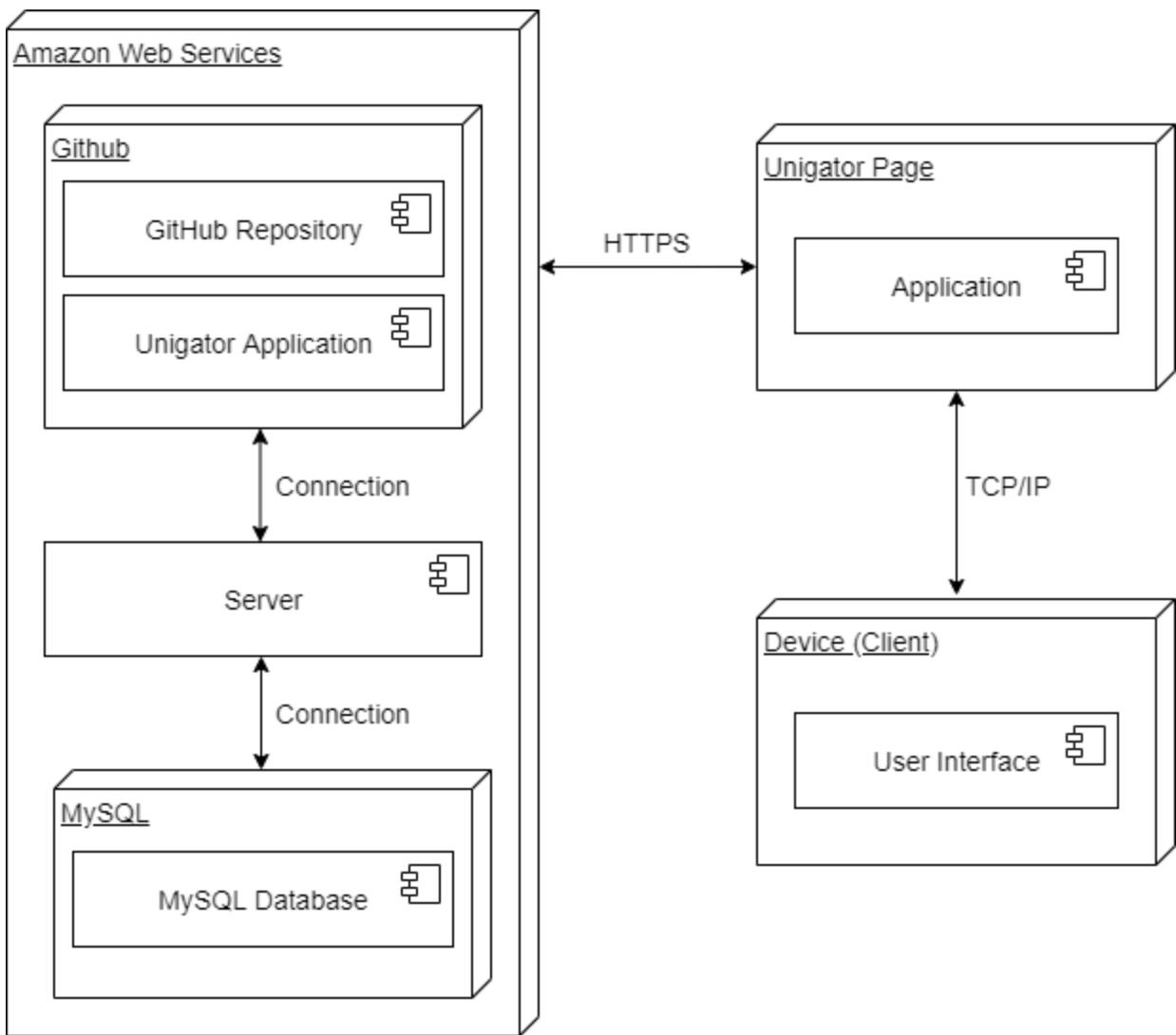
High Level UML Diagram



Application Networks Diagram



Deployment Diagram



Customer Feedback

- Description on homepage explaining what Unigator is about for users.
- Cleaning up the website to be more clear to the users.
- Remove the Home and About Nav Bar on the top.
- Placeholder on search bar.
- Changing the font size of wordings of the Pop-ups.
- Define the conditions and limits for setting the password.
- Adding Terms and Conditions.
- Clearly letting the user know that they are on the event details page.
- Hook the user's attention.
- Change the None in “Categories” to either All Categories or Blank.

SW Engineering - CSC648/848

Spring 2020



Team 03

Milestone 4

Lionel Wong	lwong7@mail.sfsu.edu	Team Lead
Jorge Landaverde	jlandaverde@mail.sfsu.edu	Lead Backend Engineer
Gordon Lam	glam1@mail.sfsu.edu	Backend Engineer
Mitul Savani	msavani@mail.sfsu.edu	Lead Frontend Engineer/Git Master
Kevin Huang	khuang2@mail.sfsu.edu	Frontend Engineer
Jack Kower	jkower@mail.sfsu.edu	Frontend Engineer

Milestone	Date
Milestone04 Version02	5/20/2020
Milestone04 Version01	5/10/2020
Milestone03 Version02	5/8/2020
Milestone03 Version01	4/23/2020
Milestone02 Version02	4/15/2020
Milestone02 Version01	3/25/2020
Milestone01 Version02	3/12/2020
Milestone01 Version01	3/5/2020

Table Of Contents

I.	Product Summary	2-3
II.	Usability Test Plan	4-5
III.	QA Test Plan	6-7
IV.	Code Review	8-9
V.	Self Check on Best Practices for Security	10
VI.	Self-check: Adherence to original Non-functional specs	11-13

Product Summary



1. San Francisco State University Students and Faculty are able to register for Unigator accounts using their school emails.
2. Users are only able to log in with their school associated email. The email needs to be verified before logging into Unigator.
3. Users on Unigator are able to search for available events by name.
4. Users are able to search through past events.
5. Users can also filter search results by Categories.
6. Users can view all information about an event.
7. Registered Users can log in or log out of their account.
8. Registered Users are able to save(Star) any event they are interested
9. Registered Users are able to RSVP to any event.
10. Registered Users are able to submit a “Create Event” request and that will be sent to an Administrator for approval. Information such as description, date, time, location and images are to be provided when creating an event.
11. Registered Users are able to become an Event Host.
12. Registered Users have their own unique profiles, which displays the user’s information.
13. Registered Users can edit their profiles.
14. Registered Users are able to view other user’s profiles.
15. Registered Users are able to view an Event Host’s email.
16. Registered Users can access the point shop and purchase customizable options by spending their accumulated points,
17. Registered Users can earn points by interacting with events on Unigator.
18. Administrators are able to view requests.
19. Administrators are able to approve or deny requests.

20. Administrators are able to ban a user.
21. Administrators are able to delete an event.
22. Administrators are able to send requests for changes to an Event Host.
23. Administrators have all options unlocked for the point shop.
24. Event Host shall be able to edit all of their event's information.
25. Event Host shall be able to delete their event.
26. Event Host are able to grant RSVPed users points.

Unigator aims to be a unique platform for anyone connected to San Francisco State University by uniting everyone together to enjoy and share their ideals through events. Any events concerning San Francisco State University can be found on Unigator and created to promote social engagement with the school and its students. Unigator unites San Francisco State University peers through events and social engagements.

URL: <http://13.52.231.107:3006/home>

Usability Test Plan

Test Objective

The Unigator team would test for event search, event creation, applying purchased items on the point shop, saving an event and RSVPing to an event. Unigator aims to hold information on numerous events, thus the need for searching a specific event is crucial. Interaction and sociability is what makes up an event, making sure that the user has a way to create an event is vital in promoting socialization. Unigator boasts a unique system where users are able to customize their profile to their liking, with our point shop. Enabling/Disabling customizable options for a user's profile. RSVPing to an event gives the Event host a range of how many users might be showing up to their events.

Test Description

- System Setup
 - Unigator is deployed on Amazon Web Services
- Starting Point
 - Users should be on the Unigator home page
- Intended Users
 - SFSU students are the intended users.
- URL to be Tested
 - <http://13.52.231.107:3006/home>
- What is to be Measured
 - User Satisfaction

Usability Task Description

- Task 1: Searching for an event from the database by name
- Task 2: Creating an event and providing information
- Task 3: Applying changes to User's profile
- Task 4: Enabling/Disabling customizable options
- Task 5: RSVPing to an event

Test / Use Case	% Completed	Error	Comments
Search Event	100%	None	Searching for a specific event is working.
Create Event	100%	None	Currently duplicates of events can be created.
Purchasing customizable options for profile	100%	None	The functionality is there, now we just need it to be shown on the frontend
Enabling/Disabling customizable options	100%	Only one “type” of customizable option can be enabled.	Still trying to fix the errors at the moment
RSVPing to an Event	80%	None	Work in progress

Questionnaire

The process of searching for specific event(s) was simple and fast:

Strongly Agree Agree Neutral Disagree Strongly Disagree

The information pertaining to a specific event was very clear and easily identifiable:

Strongly Agree **Agree** Neutral Disagree Strongly Disagree

The process of creating an event(s) was fast and simple:

Strongly Agree Agree **Neutral** Disagree Strongly Disagree

The overall responsiveness of the website was fast and simple:

Strongly Agree Agree Neutral Disagree Strongly Disagree

The point shop was an intriguing way to customize the profile to one's liking:

Strongly Agree Agree **Neutral** Disagree Strongly Disagree

QA Test Plan

Test Objective

As Unigator is a platform about events, the purpose of this test will primarily be focused on the events itself. The functionality that we will be testing are as follows: Event Creation, Event Editing, Event Deletion, Event Searching and Event Response (RSVP).

Hardware and Software Setup

Amazon Web Services instance is currently running Unigator. React.js for the frontend and Node.js on the backend with MySQL database. Google Chrome would be the default browser to be tested on.

Feature to be Tested

- Event Creation
- Event Editing
- Event Deletion
- Event Searching
- Event Response (RSVP)

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	Event Creation	Registered User creates an event.	To create an event, the following info is needed: Name, Description, Location, Date, Time, and optionally an image.	Event Created and searchable from the search bar and is in the database.	Pass
2	Event Editing	Event Host editing information on their hosted event.	Goes to the event details page of the created event, and edit information on the event, such as the time and location. Then update it.	Able to change information pertaining to the event and saved.	Fail

3	Event Deletion	Event Host deleting their event.	Goes to the event details and deletes the event. Which would also delete it from the database.	The event is non-searchable anymore, since it's deleted from the database.	Half-Pass
4	Event Searching	Registered User searches for an event.	Searches for "Mobile", for any event which has "Mobile" in it on the search bar.	The specific event searched for popped up on the results, and clicking on it will lead to the Event details page.	Pass
5	Event Response (RSVP)	Registered User RSVPing to an event.	A registered user will RSVP to an event on the event details page.	The user that RSVPed to an event is visible under the RSVP list on the event details page.	Half-Pass

Code Review

We conducted our code review through GitHub comments in the pull requests. In this specific instance, Gordon Lam was having his code reviewed by Jorge Landaverde. Since each of us have our own coding styles, our code varies a lot, especially compared with front-end and back-end, the coding styles vary.

landaverde-jorge reviewed 6 days ago

[View changes](#)

application/backend/db/index.js

```
130 +     if (err) {
131 +         return reject(err);
132 +     }
133 +     return resolve(results);
```

landaverde-jorge 6 days ago

check what you are sending back with console.log()

Reply...

Unresolve conversation LamGordon marked this conversation as resolved.

application/backend/db/index.js Outdated

landaverde-jorge 6 days ago

check what you are sending back with console.log()

Reply...

Unresolve conversation LamGordon marked this conversation as resolved.

application/backend/db/index.js Outdated

```
182 +     if (user_pointBalance < item_cost) { //check if user has enough
183 +         return reject({ error: "Insufficient amount of points."});
184 +     }
185 +     db.query(`INSERT INTO unigator.PurchasedItems (user_id, item_id, enab`
```

landaverde-jorge 6 days ago

create a separate function for reusability, also need to consider race conditions.

- What if you do not have the required information by when you need to update?

Reply...

Unresolve conversation LamGordon marked this conversation as resolved.

This is just an example of our code, the code needs to further documented by having more comments.

```

const unigatordb = {};

unigatordb.events = () => {
    return new Promise( executor: (resolve, reject) => {
        db.query( permissionDesc: `SELECT * FROM unigator.Event WHERE date > ?`, [dateFormat(new Date(), 'yyyy-mm-dd HH:MM:ss')], (err, results) => {
            if (err) {
                return reject(err);
            }
            return resolve(results)
        })
    });
}

unigatordb.pastEvents = () => {
    return new Promise( executor: (resolve, reject) => {
        db.query( permissionDesc: `SELECT * FROM unigator.Event WHERE date < ?`, [dateFormat(new Date(), 'yyyy-mm-dd HH:MM:ss')], (err, results) => {
            if (err) {
                return reject(err);
            }
            return resolve(results)
        })
    });
}
}

//no check if item is already purchased yet, either implement here or in frontend.
unigatordb.pointShopBuyItem = (user_id, item_id, item_cost) => {           //used to buy an item from the points shop for current user
    return new Promise( executor: async (resolve, reject) => {
        try {
            if (user_id==null) {
                return reject( reason: { error: "Please login if you wish to make a purchase." });
            }
            let current_user_info = await unigatordb.getUserInfoFromUserId(user_id);
            let user_pointBalance = current_user_info[0].point_balance;
            if (user_pointBalance < item_cost) {           //check if user has enough points to make purchase.
                return reject( reason: { error: "Insufficient amount of points." });
            }
            else if (user_pointBalance > item_cost) {
                db.query( permissionDesc: `INSERT IGNORE INTO unigator.PurchasedItems (user_id, item_id, enabled) VALUES (?, ?, 0)` , [user_id, item_id], async(err, results) => {
                    if (err) {
                        reject( reason: { error: "System was unable to add this item to your account." });
                    }
                    await unigatordb.updatePointBalance(user_id, (-1*item_cost));
                    return resolve( value: { message: "Purchase Successful: The item you selected has been added to your account" })
                })
            }
        } catch (e) {
            reject(e)
        }
    });
}
}

unigatordb.eventsByDate = (date) => {
    return new Promise( executor: (resolve, reject) => {
        db.query( permissionDesc: `SELECT * FROM unigator.Event WHERE date = ?` , [date], (err, results) => {
            if (err) {
                return reject(err);
            }
            return resolve(results)
        })
    });
}

unigatordb.eventInsert = (name, location, desc, date, time) => {
    return new Promise( executor: (resolve, reject) => {
        db.query( permissionDesc: `INSERT INTO unigator.Event (event_id, name, location, desc, date, time) VALUES = ?` ,
            [name, location, desc, date, time], (err, results) => {
                if (err) {
                    return reject(err);
                }
                return resolve(results[0])
            })
    });
}
}

```

Self-check on Best Practices for Security

The major assets that Unigator will be protecting are our user's password, email addresses. Sensitive information such as those are kept in the database and has encryption. The bcrypt node package allows us to encrypt our user's password, so any sensitive information would be kept confidential. We plan to only allow logged in users to view Event Host's information.

```
unigatordb.createAccount = async (email, password) => {
  return new Promise( executor: async (resolve, reject) => {
    password = await bcrypt.hash(password, saltRounds);
    db.query( permissionDesc: `INSERT INTO unigator.Account (email, password) VALUES(?,?)` ,
      [email, password], (err, results) => {
        if (err) {
          reject( reason: { error: "Email already in use" });
        } else {
          resolve(results.insertId);
        }
      });
  })
}
```

Self-check on Adherence to Original Non-Functional Specs

- 1) Security:
 - a) Can only create an event page if you are logged into your account.
(DONE)
 - b) You can only create an account with a SFSU email.(DONE)
 - c) Able to request change of password.(ON TRACK)
 - d) Passwords will be encrypted.(DONE)
 - e) Have password requirements for creating an account and resetting password.(ON TRACK)
- 2) Audit:
 - a) The user may request access to create an account without SFSU email.(ON TRACK)
 - b) The administrator may delete any event from the event page.(ON TRACK)
 - c) The administrator may view all RSVP accounts of any event.(DONE)
 - d) The administrator may view and approve any event requests.(ON TRACK)
 - e) New events must be approved by the administrator.(ON TRACK)
- 3) Response Time:
 - a) The initial load up time on the site shall fall under 5 seconds.(DONE)
 - b) Retrieving data from the database shall be fast and efficient.(DONE)
- 4) Capacity:
 - a) The application shall be able to hold at least 50 events within the span of two weeks. (DONE)
 - b) The application shall save a log of past events within 2 months.(ISSUE)
 - c) The application shall be capable of handling accounts of at least all the SFSU students. (DONE)
 - d) The application shall be made in a way that it's scalable, in case of and further feature updates are planned.(DONE)
- 5) Reliability:
 - a) Downtime maintenance shall be done within 5 hours.(DONE)
 - b) Downtime maintenance shall restrict users from using the application to prevent any false information.(DONE)
 - c) The application shall not crash under any circumstances.(DONE)
 - d) Registered Users shall be informed of maintenance via an announcement on the main page, or through email.(ON TRACK)
- 6) Recovery:

- a) If any severe problems arise in the application, the application shall be down for repairs.(DONE)
 - b) Application downtime shall not exceed 24 hours. (DONE)
 - c) If the application is down, Registered User data and Events data shall be safe.(DONE)
- 7) Data Integrity:
- a) All Data shall be backed up every 72 hours. (ON TRACK)
 - b) The users shall be prompted when they wish to delete or edit data.(ON TRACK)
 - c) Any purchases in the point shop shall be saved with the Registered User's account.(DONE)
 - d) Account recovery can be requested to the Administrator.(ON TRACK)
 - e) Image sizes shall not exceed 4 megabyte.(DONE)
 - f) Only image format of JPG, JPEG and PNG shall be accepted.(DONE)
- 8) Platform:
- a) The application shall be compatible with Chrome browser version 80.0.3987.132 and earlier.(DONE)
 - b) The application shall be compatible with Firefox browser version 74.0 and earlier.(DONE)
 - c) The application shall be compatible with Safari browser version 13.0 and earlier.(DONE)
 - d) The application shall scale correctly with the screen size of the device its on. (ISSUE)
 - e) The application shall be compatible with any client OS that supports Chrome, Firefox, or Safari browsers. (DONE)
 - f) The application shall be able to account for any compatibility issues as a result of browser or OS updates. (ON TRACK)
- 9) Coding Standards:
- a) Comments should accompany the code in order to understand functionality.(DONE)
 - b) Errors shall not disrupt the application in a way that is detrimental.(DONE)
 - c) Code needs to be reviewed and tested by all members of the group before going into the master branch.(DONE)
 - d) Internal and external error shall be recorded in a log for future reference.(ON TRACK)
 - e) Any error or bugs that affect the user should be notified to them through their email.(ON TRACK)

- f) Error debugging shall be done without interfering with the application's functionality.(ON TRACK)
 - g) All priority one requirements shall be implemented and working before launching the application.(ON TRACK)
- 10) Look and Feel Standards:
- a) The application should follow the aesthetics of San Francisco State University's site.(ON TRACK)
 - b) Registered Users profiles are unique and customizable by the user.(DONE)
 - c) The application shall appear simple in order to avoid any confusion to the users.(ON TRACK)
 - d) Registered Users needs shall be prioritized over Unregistered Users.(DONE)
 - e) The application should be automatically resized based on the screen size of the device.(ON TRACK)
 - f) Have a loading toggle wheel.(ON TRACK)
 - g) Have Links to Unigator social media in footer. (ON TRACK)
 - h) Follow the color scheme of SFSU with purple and gold.(DONE)
- 11) Internalization / Localization:
- a) English shall be the default language of the application.(DONE)
 - b) The application shall only allow SFSU students to register accounts.(ON TRACK)
 - c) Non-SFSU students shall be able to make requests.(ON TRACK)
- 12) Website Policies:
- a) The application's policies shall be visible via a link on the homepage.(ON TRACK)
 - b) The application shall not ask for any sensitive information.(DONE)
 - c) Registered User info shall be kept confidential.(DONE)
 - d) A verification of the email shall be done in the register section.(ON TRACK)
 - e) Users shall understand and read through all the policies and abide by it when using the application.(ON TRACK)

Screenshots of Final Product

The banner features a large blue and teal circular background. In the center, there is a white square containing the "SE STATE" logo with a yellow alligator. Above the logo, the text "A Place to Unite San Francisco State University Patrons!" is displayed in bold white font. Below the logo, a smaller text reads: "Take part in our events focused platform to connect and socialize with your fellow Gators!"

The page displays a grid of six event cards, each featuring a thumbnail image of a crowded hall with purple lighting and balloons.

Mobile 101 #1 Thorton Hall Share Learn More	Mobile 101 #2 Thorton Hall Share Learn More	Teacher Workshop: Women in Disney History Thorton Hall Share Learn More
Gap Year Lunch Panel Thorton Hall Share Learn More	Theater Workshop Thorton Hall Share Learn More	Fall Sports Kick-Off Thorton Hall Share Learn More

Unigator

Search:

Event Details

Mobile dev 101

4/7/2020 9:20 AM Towers Conference Center

Event Description

Lets learn how to create a mobile app

Unigator

Search:

Red

Orange

Yellow

Green

Blue

Lucida Console



Search:
[Create Event](#) [Log in](#) [Sign Up](#)

My Profile

Welcome,
JohnSmith123!



Share your profile:

User Information

Username: JohnSmith123
 Email: jsmith@mail.sfsu.edu
 School Year: Class of 2021
 Points Accumulated: 1300
 Preferred Categories: Technology, Recreation
 Description: Hi, I am JohnSmith123. I like pizza.

My Events

Create Event ×

Event Name

Location

Date X Delete

Time X Set

Category

Description

Choose picture
 No file chosen

Screenshots of Key DB Tables

Account Table

acc_id	email	password
24	sdfsdf@sfsu.edu	\$2b\$05\$c8UZ4r6umZx2ETZfr1zerOyxTBCyg40/CZdtwfgphCcT1mrgVTSa2
25	arodgers@sfsu.edu	\$2b\$05\$HdAYyxr8xpTuWUTSB4gsDuwfHe8PooR3PWjrtQ443bUMtUIZfxnoi
26	testaccount@me.com	\$2b\$05\$.eGU4adYUrRzjNQE94JM5ul2sH38TTe11QHYUXvxJWtgSaUxVCG8q
27	jorge@me.com	\$2b\$05\$qurSbrO.Gk8MOrN/BQa.FenZ9Q4aEa9jGGXlqqDyczmDThAUaT.sS
28	San	\$2b\$05\$ZioFekX4d94JRP1Yx/M3k.JAxcpYuAAbeGdtbZVr4ZHkdz7BXW2sm
29	test@test.com	\$2b\$05\$5p.DNFf7TceQNFmbM/diCeVK9H2VfzsYMGsGOasinXgi6AS7QhGIS
32	aar	\$2b\$05\$yXfoNurwlKudvH/z2A5Wx.mo7FfiAGnm4AMZNKyG6wW7LRotUKqN.
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL

Event Table

event_id	status	name	location	description	date	time	image
1	pending	Mobile 101 #1	Thorton Hall	Come learn how to create your first mobile app	2020-06-07 00:00:00	18:00:00	NULL
2	pending	Mobile 101 #2	Thorton Hall	Come learn how to create your first mobile app	2020-06-08 00:00:00	18:00:00	NULL
3	pending	Teacher Workshop: Wo...	Thorton Hall	Come find out how women have been involved in differ...	2020-06-09 00:00:00	18:00:00	NULL
4	pending	Gap Year Lunch Panel	Thorton Hall	Description of Gap Year Lunch Panel	2020-06-10 00:00:00	18:00:00	NULL
5	pending	Theater Workshop	Thorton Hall	Description of Theater Workshop	2020-06-11 00:00:00	18:00:00	NULL
6	pending	Fall Sports Kick-Off	Thorton Hall	Description of Fall Sports Kick-Off	2020-06-12 00:00:00	18:00:00	NULL
7	pending	Game Competition	Thorton Hall	Description of Game Competition	2020-06-13 00:00:00	18:00:00	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Category

category_id	description	type
1	Explore new technologies and enjoy a good time	Technology
2	Explore new topics that will provide you with	Education
3	Find a new hobby or a new buddy to try new th	Extracurricular
4	Come out and try out a new sport or play with	Sport
5	Get your creative side exposed to new ways of	Creativity
6	Find new people who you can find new adventur	Outdoors
NULL	NULL	NULL

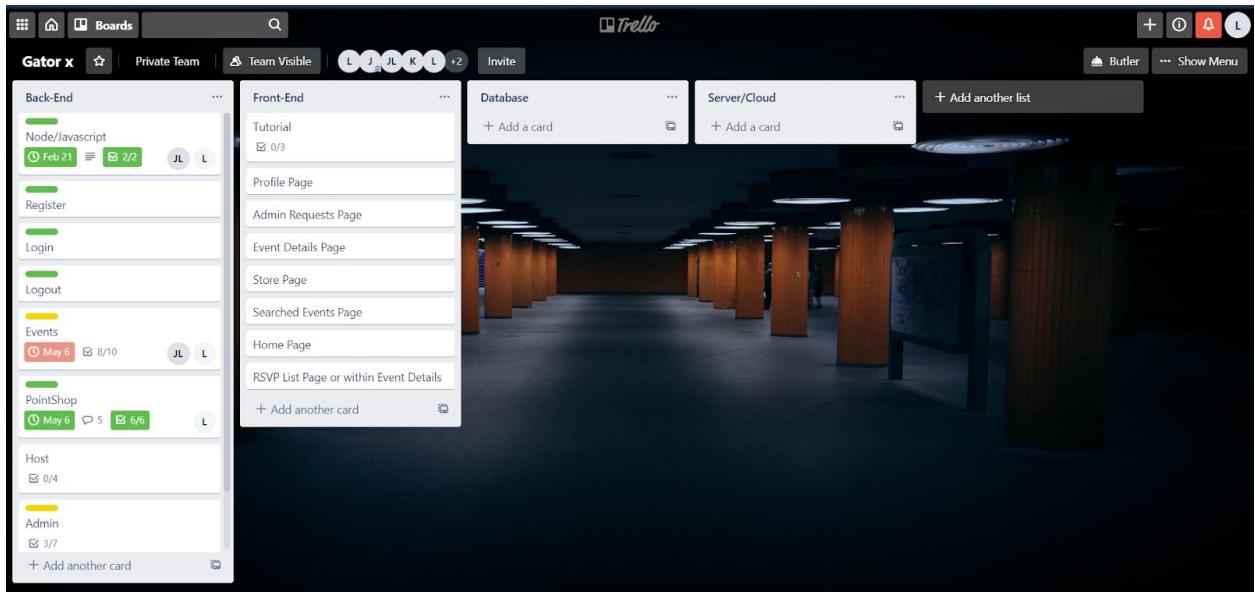
Event Host

host_id	user_id	event_count	hp_enabled
1	16	7	1
NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL

Shop items

item_id	name	cost	value	type
3	Yellow	50	color: Yellow;	color
4	Green	50	color: MediumSeaGreen;	color
5	Blue	50	color: DodgerBlue;	color
6	Lucida Console	100	font-family: "Lucida Console";	font
7	Courier New	100	font-family: "Courier New";	font
8	Comic Sans MS	100	font-family: "Comic Sans MS";	font
9	Georgia	100	font-family: Georgia;	font
10	Arial Black	100	font-family: "Arial Black";	font
11	Italics	200	font-style: italic;	style
12	Bold	250	font-weight: bold;	weight
NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Screenshots of Task Management System



Team Contributions



Lionel Siu Foong Wong

Wed 5/20/2020 1:03 PM

To: Gordon Huynh Lam; Kevin Wing Geet Huang; Mitul Odhavjibhai Savani; Jorge Antonio Landaverde; Jack Alexander Kower



Hello Team 03,

These are my contributions for CSC648

Team Lead Role:

Assigned team members to their respective roles. Communicated with the team the best that I can with the team through Slack. Set up group meetings and discussions as a team weekly. Cleared up any confusion any team members may had on a particular subject. Tried to make sure everyone stays on tasks.

Milestone 1:

Wrote about 6 use cases, helped came up with some requirements, and added on the competitive analysis part.

Milestone 2:

Started the initial draft of our ERD, Database Model, and Deployment Diagram. Filled out Key Risks, and Project Management and re tuned the requirements.

Helped set up the vertical prototype.

Milestone 3:

Adding more information on Data Definitions and did all of the Wireframes. Helped implement some UI for the horizontal prototype. Checked up on the team and organized meetings.

Milestone 4:

Added more details on the Homepage of the application. Did most of the documentation of Milestone 4, tested some functionalities and inquire on outsider's opinions on the application.

Milestone 5:

Completed the Milestone 5 documentation and did some final help on the application.

Had 13 commits in total.

Thank You, Lionel.



Gordon Huynh Lam

Wed 5/20/2020 10:43 PM

To: Gordon Huynh Lam



Hello Gordon,

I agree with your contributions.

Thank You, Lionel.

...



Gordon Huynh Lam

Wed 5/20/2020 10:40 PM

To: Lionel Siu Foong Wong; Kevin Wing Geet Huang; Mitul Odhavjibhai Savani; Jorge Antonio Landaverde; Jack Alexander Kower



Hello,

These are a list of my contributions for out project in CSC648

Role:

Backend Engineer - task was to mainly work on back-end portion of code alongside the back-end lead, and assist in documentation during the early stages of this project when there was little to no back-end coding required.

Milestone 1:

Worked on writing Use Cases, also worked on main data items and entities, initial list of functional requirements, and list on non-functional requirements along with everybody in the group.

Milestone 2:

Worked slightly on each: Data Definitions, Functional Requirements, Database Organization and ERD.

Milestone 3:

Did not do much during M3 other than slightly helping with Data Entries.

Milestone 4:

Little to no work on documentation portion of M4. Worked on coding back-end during this time; mostly working on Point Shop.

Milestone 5:

No work on documentation of M5. Continued coding back-end of project; Finishing Point Shop functionality; Host Funcionality, Admin Functionality, and a little work with Events functionality.

Total number of meaningful commits in development branch at the time is ~20.

Thank You, Gordon Lam



Lionel Siu Foong Wong

Wed 5/20/2020 10:46 PM

To: Kevin Wing Geet Huang

Hello Kevin,

I agree with your contributions.

Thank You, Lionel.

...



Kevin Wing Geet Huang

Wed 5/20/2020 10:41 PM

To: Lionel Siu Foong Wong

Cc: Gordon Huynh Lam; Mitul Odhayjibhai Savani; Jorge Antonio Landaverde; Jack Alexander Kover

Hello Team 3,

These are my contributions for CSC 648.

Milestone 1:

- Helped write use cases
- Helped come up with functional and nonfunctional requirements

Milestone 2:

- Designed and hand drew all mockups and storyboards
- Helped organize functional requirements by priority
- Remade the application network diagram and deployment diagram after feedback from the professor for the document
- Worked on frontend development for vertical prototype

Milestone 3:

- Worked on frontend development for horizontal prototype
- Worked with Lionel (team lead) and Mitul (front end lead) to refine the UI designs for the application and wireframes for the documentation

Milestone 4:

- Continued working on frontend development

Milestone 5:

- Continued working on frontend development

I have 24 commits to the development branch as of 5/20/2020 10:30 PM.

Feel free to let me know if you disagree with me or if there's something I forgot to add.

Thank you,

Kevin Huang



Lionel Siu Foong Wong
Thu 5/21/2020 1:03 PM
To: Jorge Antonio Landaverde



Hello Jorge,
I agree with your contributions and the valid criticisms, I felt like you did more than you stated though.

Thank You, Lionel.

...



Jorge Antonio Landaverde
Thu 5/21/2020 12:52 PM
To: Lionel Siu Foong Wong; Gordon Huynh Lam; Kevin Wing Geet Huang; Mitul Odhavjibhai Savani; Jack Alexander Kower



Hello,

My contributions to the team project are as follow:

Milestone 1
Milestone 2
Milestone 3
Milestone 4
Milestone 5

For all of the milestones my main focus has been getting the team to move forward. From setting up the initial application AWS EC2 Instance, and setting up the database along with other things. I have worked in most of the documents to determine the requirements for the application and the needs for the data that we will be storing. I have worked in both front-end and back-end as have set Github requirements so the project can flow as smoothly as possible. My overall contribution to the project has been huge since I have taken care of making sure that everyone is doing something that will move the project forward.

I believe most of the other team members could have done a better job (Mitul, Kevin, Gordon). You guys could have put more effort into the project earlier in the semester and this would have helped us move faster and get us to a project that is more than just passing. I believe Jack could have told us that he was not gonna be present and therefore not rely on him to do any work. Would that have been a big difference? maybe not since we still slacked off and put very little effort into this project. I tried my best to help everyone in the team with their requirements but there was a huge lack of communication from everyone except Lionel.

 Lionel Siu Foong Wong
Thu 5/21/2020 2:43 PM
To: Jack Alexander Kower

Hello Jack,
I disagree with your contributions, you might've submitted your work on somewhere, but we don't see it on our github repo. You were missing in terms of communication for practically the whole time and did not communicate with the team at all. You did help out for Milestone 1's documentation though, but after that you went missing throughout the whole semester.

Thank You, Lionel.

...

 Jack Alexander Kower
Thu 5/21/2020 2:36 PM
To: Mitul Odhaijibhai Savani; Lionel Siu Foong Wong
Cc: Jorge Antonio Landaverde; Kevin Wing Geet Huang; Gordon Huynh Lam

Here are my contributions to team 03:

Milestone 1:
For milestones I did a little bit of everything. I added to the Main uses cases. Added more situations that met not have been initially seen and helped handle planning for exceptions. I added a lot to security and how the software should worry about user needs and making each user feel safe. I added the comparative product of SFSU's GatorX and compare the entities to Unigator and should be improved on. I helped come up with some of the high-level technologies.

Milestone 2:
For milestone 2, I developed a high-level User Interface design. I implemented text, storyboarding, and user experiences mock-ups about how each page should operate, the design choices for each page, and connection for all the pages as one application. I went more in-depth with how to have key and reusable components such as navigation bar and each page's layout.

Milestone 3:
In milestone 3, I went more into detail about the user needs and experience and how to make the navigation of the app more accessible for a wider range of our user base. I went into detail about searching and the horizontal web app layout. This eventually leads to me working on the prototype, where I designed the practical layout of the Home page and the design and aesthetic that best fit the user with our technologies.

Milestone 4:
For Milestone 4, I added to the front page and the log in pop up. I edited the color and layout, Providing specific solutions to the navigation bar. I rearranged the order of the buttons that had made the most psychological sense for our web traffic and visitors.

Milestone 5:
And finally for Milestone 5, I wrapped up all the other work I did with some last-minute edits and used principles of UX design and Human-Computer Interaction to double-check that all the aesthetic of the application was working correctly and fit our user base as best as possible.

I had a total of 4 Commits. I also contributed extensively to the mockups and design of the software. Thank you to everybody who worked on this project, especially Lionel and our great and lovely professor.

Best,
Jack Kower

...

 Jack Alexander Kower
Thu 5/21/2020 2:50 PM
To: Lionel Siu Foong Wong

Lastly, just to add to my role: Front End Developer:

As a front end developer, I had the job of doing both development and UX design. I developed mock-ups, storyboards, and the overall aesthetic for the user. As a developer, I worked with front end web technologies to help put the designs into practice as the development.

...

 Jack Alexander Kower
Thu 5/21/2020 2:47 PM
To: Lionel Siu Foong Wong

Thank you for your feedback. Yes I should have done more work. It was hard with my schedule, but that is no excuse. I did very little at the end with Milestone 4 and basically no work the last week. I believe I did a good amount of work on the first two milestones and did some parts of Milestone 3. My contributions to Milestone 3 were very little and I recognize that. Kevin and I did do all the work for storyboarding and mock-ups. Thank you for the input.

Best,
Jack Kower

...

 Lionel Siu Foong Wong
Thu 5/21/2020 2:52 PM
To: Mitul Odhavjibhai Savani

Hello Mitul,
I disagree with some of the contributions, for Milestone 2, the vertical prototype was done mostly by Kevin around the last day or day before, Milestone 3 I made all of the wireframes. For Milestone 4 and 5, you were kind of missing for a while, you did some frontend work for Milestone 5 but, it didn't get pushed to the repo in time. I do agree that you've been very helpful on Github though, looking over requests and teaching us the way of how Github works in general.

Thank You, Lionel.

...

 Mitul Odhavjibhai Savani
Thu 5/21/2020 2:40 PM
To: Jack Alexander Kover; Lionel Siu Foong Wong
Cc: Jorge Antonio Landaverde; Kevin Wing Geet Huang; Gordon Huynh Lam

Hello Team 03 of CSC548,

This email contains my contributions to our project.

Role:
GitHub Master: My task was to mainly assist some of my teammates to fully understand the overflow of how GitHub is used for such huge engineering projects, which includes how to make and delete branches, writing relevant commit messages, pushing and creating a pull request.

FrontEnd Lead: My task was to analyze the project details and assign task/sprints to front-end engineers based on the time period that was given to us. With that being said, I assisted other engineers with their specific front-end framework queries.

Milestone 1:
I worked on analyzing the use cases, the functional requirement for the project along with my other teammates.

Milestone 2:
Continued reviewing the functional requirement and overlooked how the front-end will interact with our database model. Based on that information, I collaborated with the frontend team to make a storyboard and initialized the repository, tested, and imported third party libraries/packages for the vertical prototype.

Milestone 3:
Created wireframes and assigned tasks to team members to reflect the wireframes on the frontend to finish the horizontal prototype. Meanwhile I was also overlooking the GitHub repository to test and review pull requests that were coming in to give a signal to merge the branches to our team lead.

Milestone 4:
I continued working on the frontend side and resolved some of the feedbacks that were given by the professor on the vertical prototype.

Milestone 5:
Worked on the remaining use cases and improved UI.

I had in total 11 commits
Thank you, Professor, and my team members.

...

Post Analysis

The experience in this class was unlike anything I could've imagined it turning out to be. A lot of things have happened throughout this whole semester. The global pandemic happened, which forced us to move to a fully online environment in the first month of classes. The main challenge that we faced was trying to adapt to the new type of environment everyone is facing and not being able to see each other. Communication was easily the most challenging aspect about this whole ordeal. We can't meet up with each other in person anymore, which made discussion very hard. It was my first experience in managing a team in a real industry manner. It was a daunting task, I had much to learn about managing and communicating with the team, and not be afraid to speak up when necessary. At times we had disagreements with each other, and I only wish that we could've resolved it better. It was difficult to get our ideals across to each other, or even get a response on a message. Everyone was practically new to the technology we had to use, all of us had to learn about new environments and technologies to use in this class. Making sure that everyone stays on task was also difficult, as we can only delegate tasks to our team members and hope that they fulfill it.

Next time when these kinds of challenges arise, we hope to do communication better at the start and clearly get across our ideals to each other. There were times where a team member had a misunderstanding of what had to be done, and we never noticed it until it was too late. We should've dealt with tasks separately in a more coordinated way and spent more time on it. We should've used our time more efficiently, whether it be learning about new technologies or future tasks. I learned a lot from this class, especially on how to properly manage a team. I may not be the best team lead, but my team understood the situation we were facing from the beginning of the class. Overall, the team helped me along the way, giving encouragement and advice. We did the best we could as a team, considering the circumstances we've had, and that there were some members that went missing for quite some time, the current state of our project is the result of it.