

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 no 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	<p>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>
Host Points	<p>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>

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

Hand-drawn UI mockups for a registration system, showing the Login Page and Sign up Page.

Login Page

Navigation: Unigator | Search | Q | Login | Sign Up

Login

SFSU E-mail:

Password:

Login:

Sign up for a new account — [Link to sign up page](#)

Sign up Page

Navigation: Unigator | Search | Q | Login | 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:

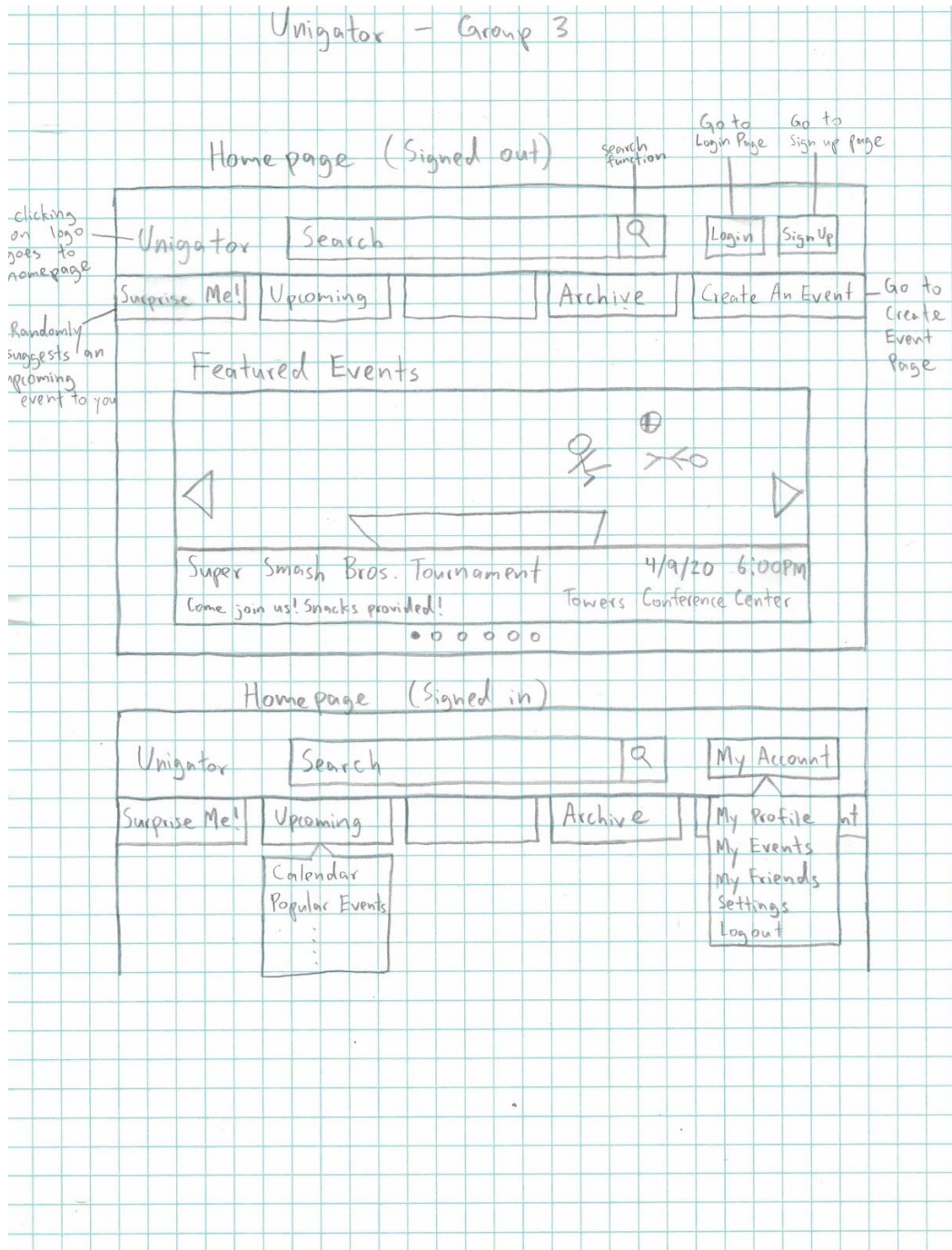
Username:

Password:

Sign Up:

I already have an account — [Link to login page](#)

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
Month Day Year

Start Time: 06 : 00 AM/PM

End Time: 09 : 00 AM/PM

Event Type: Recreation ▾

Description: Join us for a fun evening of Super Smash Bros. Ultimate! Controllers are provided. Light snacks and refreshments provided.

Photo(s):

Admission: ☒ Free ☐ Enter price: \$

Location: Towers Conference Center

Submit Event

Your event will be submitted for approval by Unigator administrators within 48 hours.

My Profile Page

My Profile Page

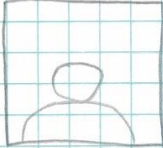
Unigator

Search

Q

My Account

Jo_Smith27







Change Profile Picture

Information (edit)

Name: John Smith (Private)
Username: Jo-Smith27
Email: jsmith@mail.sfsu.edu (private)

My Friends (22)

See All

Saved Events (2)

See All

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

My Friends Page

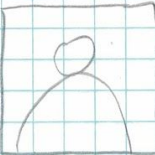
Unigator

Search







Q

My Account

Jo_Smith27



My Friends

Firstname Lastname	Firstname Lastname
 ~ ~ ~	 ~ ~ ~
 ~ ~ ~	 ~ ~ ~
 ~ ~ ~	 ~ ~ ~

Saved Event Page

My Events Page (Saved)

Unigator

My Events

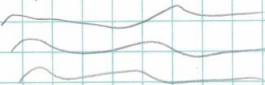
Date and Time	Event	Location
3/20/20 6:00 PM	Concert	Cesar Chavez Student Center
3/27/20 12:00 PM	Baseball Game Description: Sonoma State @ SF State Spend your Friday afternoon cheering on your SF State Gators baseball team!	Maloney Field

Click on
an event
to expand
to see
picture and
description

My Events Page (Hosting)

Unigator

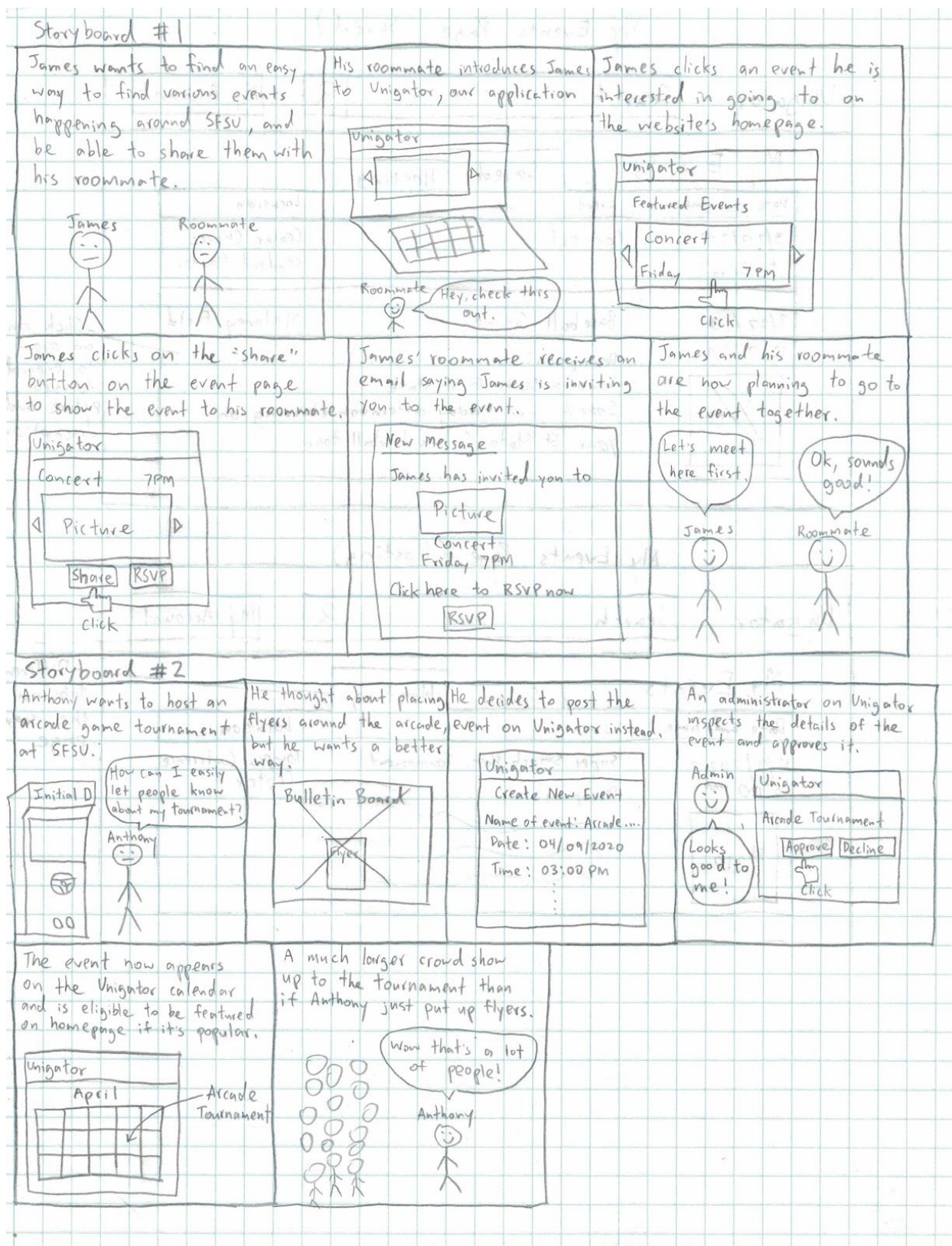
My Events

Date and Time	Event	Location
4/9/2020 6:00 PM	Super Smash Bros. Tournament Description: 	Towers Conference Center

Photo

Different
tabs on
the same
page

Storyboards



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

Filters 1. Tech Fair 3/21/20

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 Events

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 11AM

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.

Steve

Tech Fair

→

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

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.

Facebook

Twitter

Instagram

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 started 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.



Are you going to the tournament?



Yes!

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

Unigator

Super Smash Bros

Search Results

~~~~~

Click

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 see 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

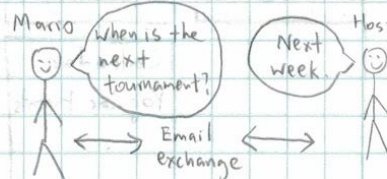
SFSU Email

Username

Password

Sign Up

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





## Storyboard #7

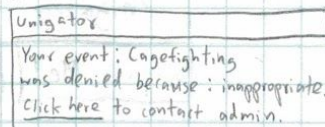
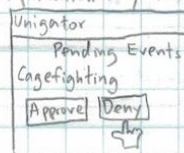
Luigi is an administrator at Unigator. He has to approve all events before they go live on the website.



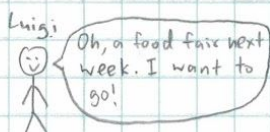
After logging into his account with admin privileges, he can see a list of pending events. He approves events as he sees fit.



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.

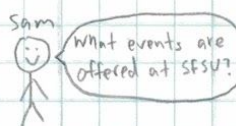


After completing his work, Luigi will logout of his account.

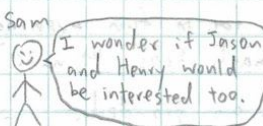


## Storyboard #8

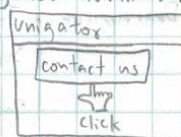
Sam is a student at a local community college. He is interested in attending cultural events at SFSU.



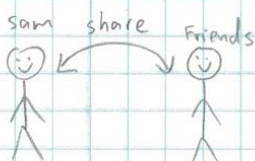
He finds 2 events he wants to attend with his friends at SFSU.



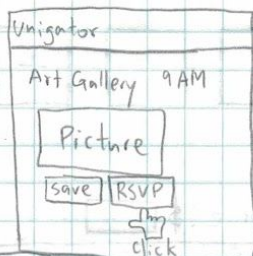
Since Sam is not a SFSU student, he needs to request an account from an Unigator administrator.



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.

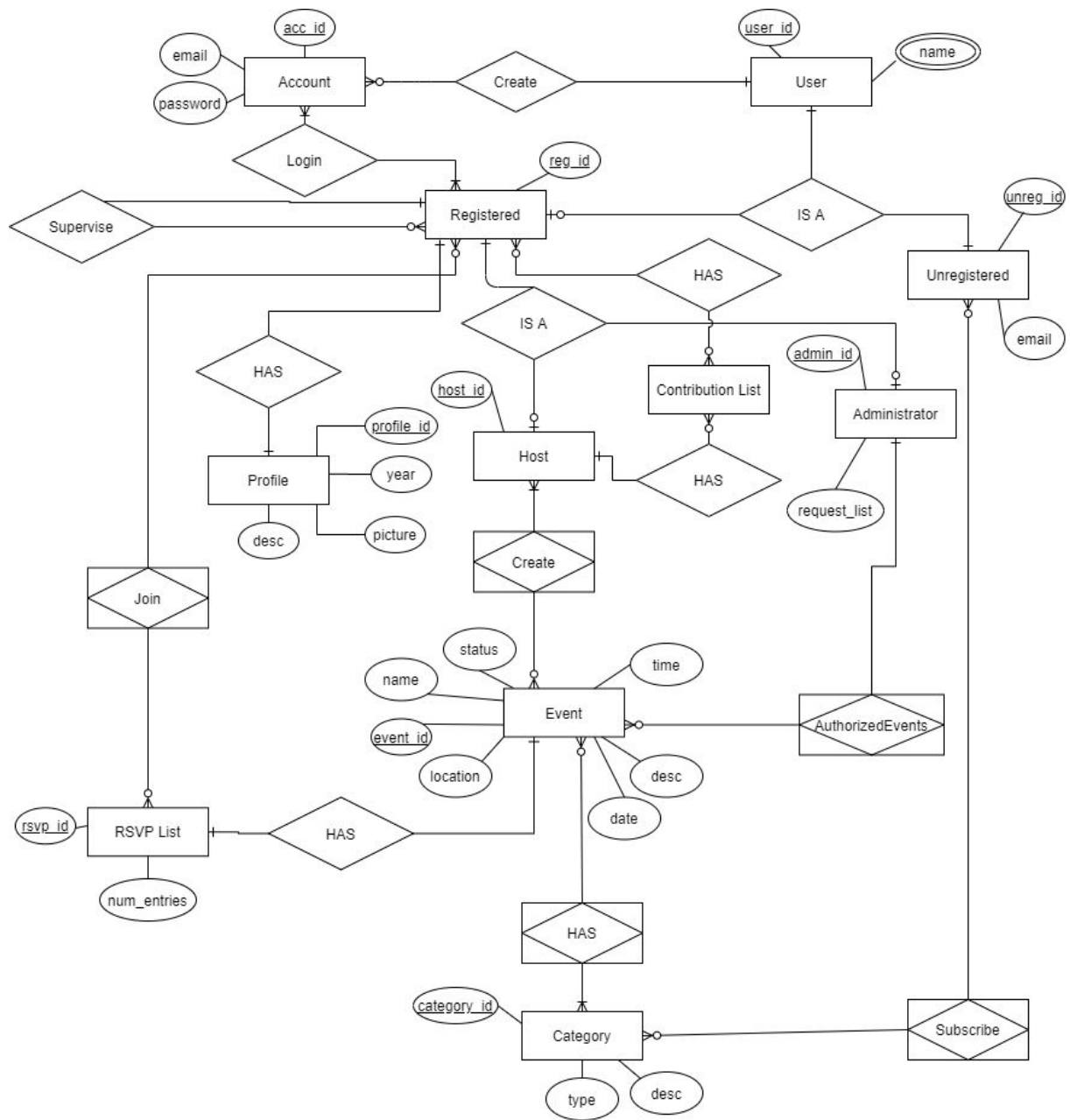


# High level Database Architecture and Organization

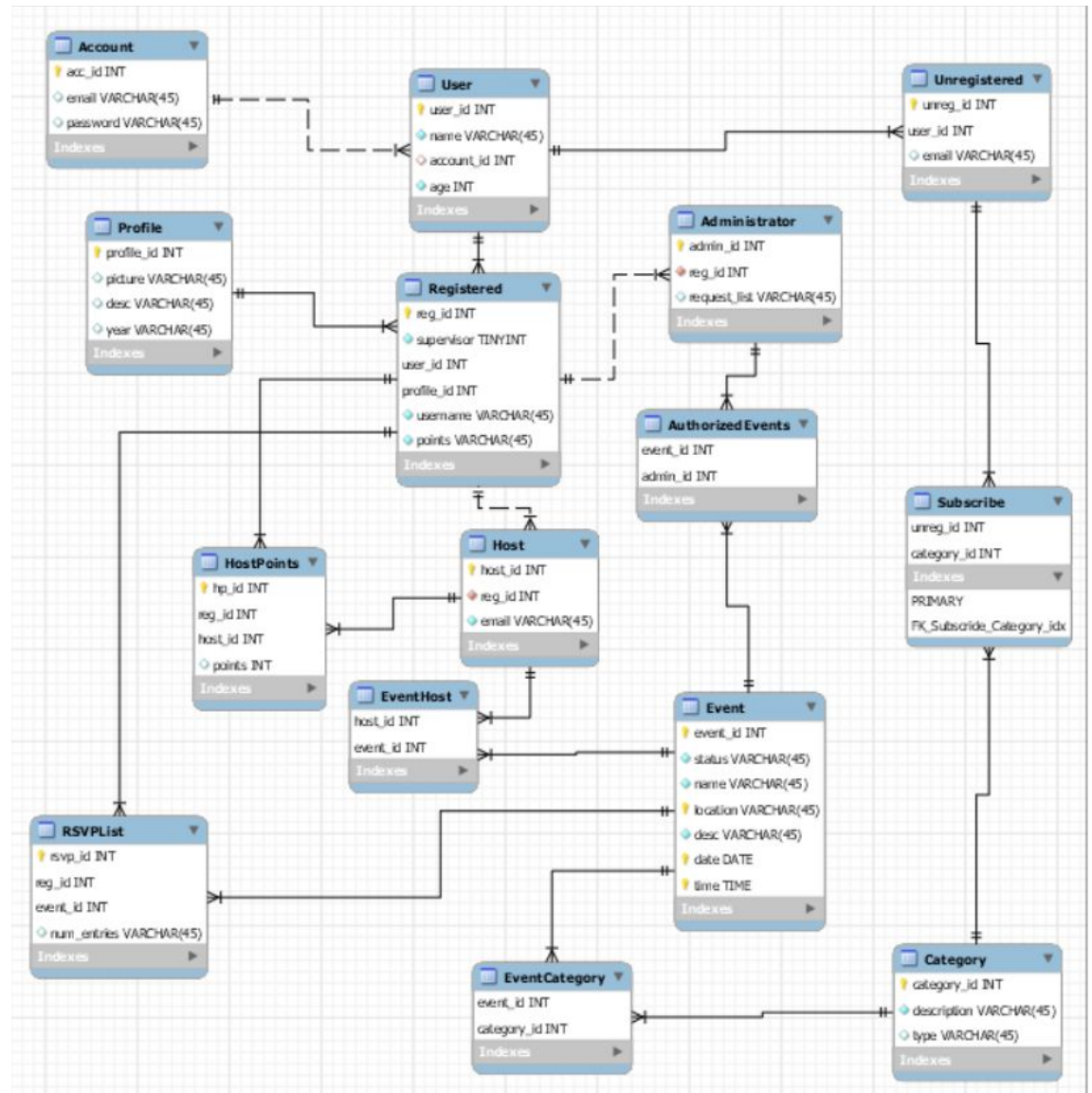
- Database Organization
  - Business Rules
    1. Unregistered Users can create at most one account on the application per SFSU email.
    2. Registered Users can log in into the application on multiple devices.
    3. An event shall only have at most one RSVP list.
    4. Multiple Registered Users can 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 once 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.

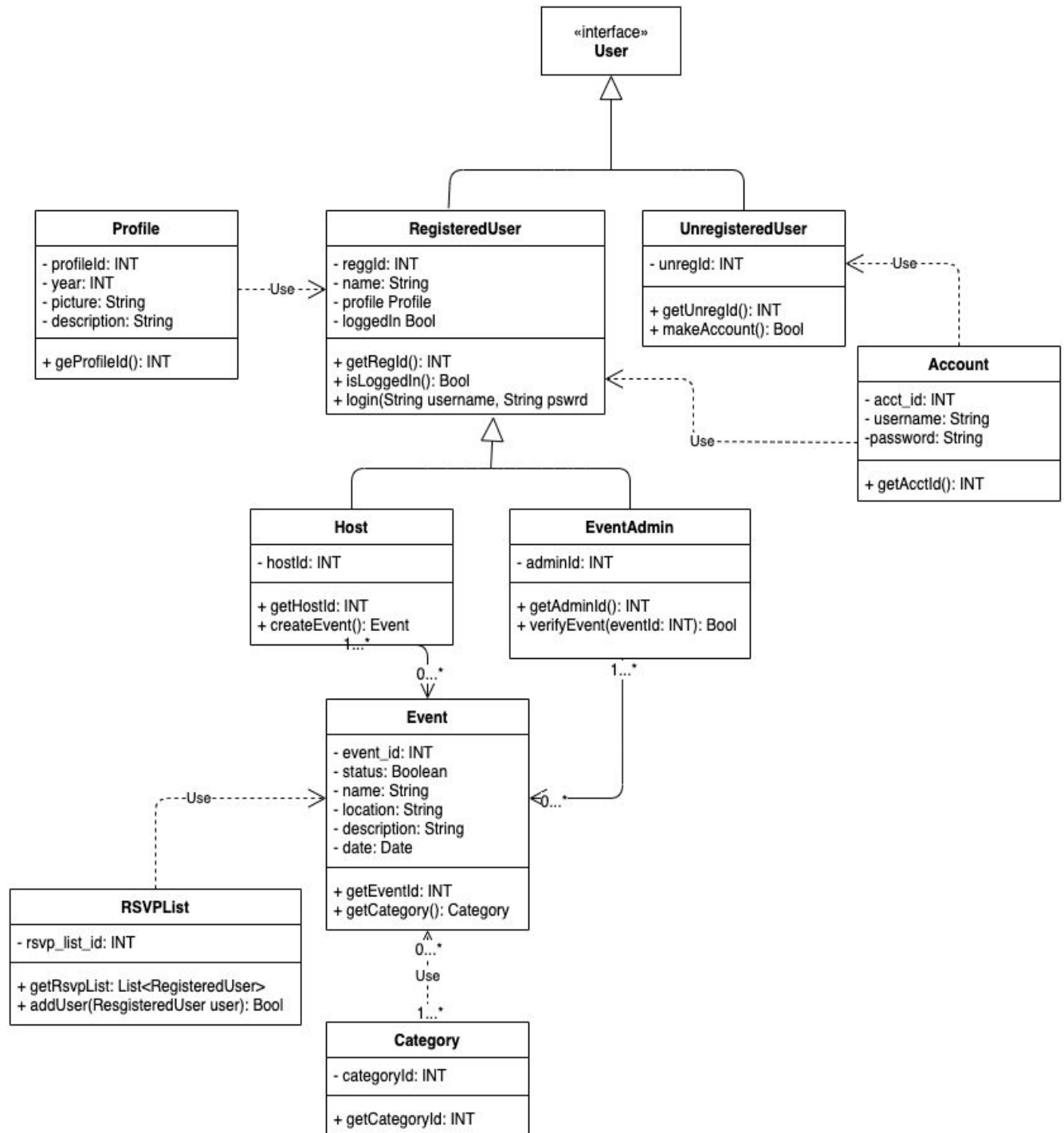
## EventRSVP

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 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.

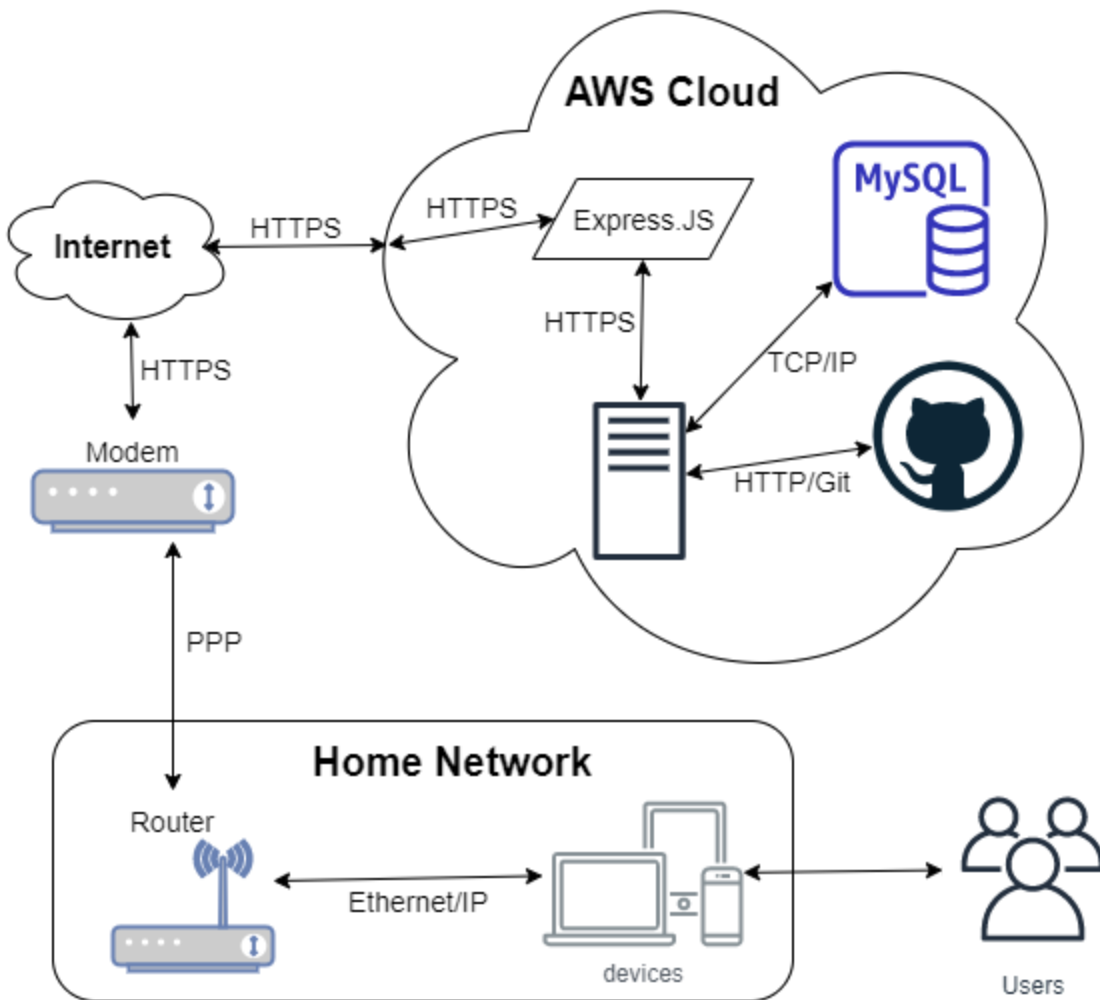


# High Level UML Diagram

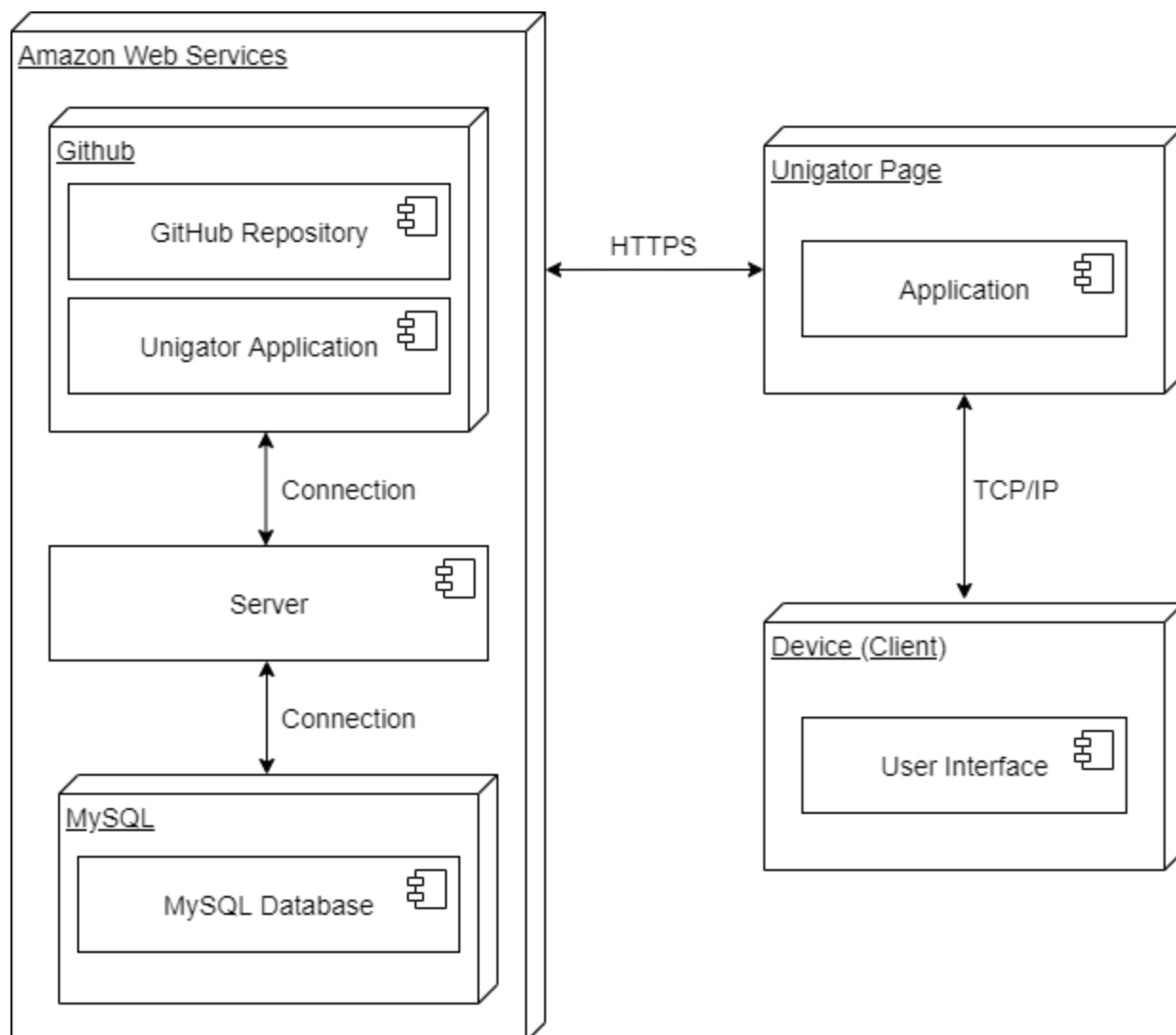


# High Level Application Network and Deployment Diagrams

## 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 it's 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.