

# SW Engineering CSC648-848

Summer 2022

<b>“PlayDate” Application — by Team 03 (the “Babysitters”)</b>		
<b>Name</b>	<b>Job</b>	<b>Email</b>
Soujanya Ravindra Nayak	Team Lead	<a href="mailto:soujanyaravindra@gmail.com">soujanyaravindra@gmail.com</a>
Margaret De La Torre	Frontend Lead	<a href="mailto:mdelato1@mail.sfsu.edu">mdelato1@mail.sfsu.edu</a>
Andy Cho	Backend Lead	<a href="mailto:andrewyunejo@gmail.com">andrewyunejo@gmail.com</a>
Martin Salvatierra	Frontend Team	<a href="mailto:martysaljhost@gmail.com">martysaljhost@gmail.com</a>
Qin Geng	Frontend Team	<a href="mailto:gengqin50@gmail.com">gengqin50@gmail.com</a>
William Plachno	Backend Team	<a href="mailto:wiplachno@gmail.com">wiplachno@gmail.com</a>
Victor Callejas	Frontend Team	<a href="mailto:vcallejas@mail.sfsu.edu">vcallejas@mail.sfsu.edu</a>

Date	Version
07/07/2022	M2V1
06/21/2022	M1V1

## Table of Contents

<b>DATA DEFINITIONS</b>	<b>3</b>
<b>PRIORITIZED FUNCTIONAL REQUIREMENTS</b>	<b>7</b>
<b>UI MOCKUPS &amp; STORYBOARD</b>	<b>11</b>
<b>HIGH LEVEL DATABASE ARCHITECTURE &amp; ORGANIZATION</b>	<b>24</b>
<b>HIGH LEVEL APIs AND MAIN ALGORITHMS</b>	<b>31</b>
<b>HIGH LEVEL UML DIAGRAMS</b>	<b>32</b>
<b>HIGH LEVEL APPLICATION NETWORK AND DEPLOYMENT DIAGRAMS</b>	<b>33</b>
<b>IDENTIFY ACTUAL KEY RISKS FOR YOUR PROJECT AT THIS TIME</b>	<b>34</b>
<b>PROJECT MANAGEMENT</b>	<b>35</b>
<b>DETAILED LIST OF CONTRIBUTION</b>	<b>36</b>

## 1. Data Definitions

1. **General users:** Can browse the homepage, view and search for public events.
  - 1.1. **general\_id:** there will be a general\_id generated every time a general user interacts with “PlayDate” apps.
  - 1.2. Look up the homepage and public events
  - 1.3. Search for public events
2. **Registered users:** A user shall be able to look up the website and search for public events. Once a user login the system, they shall be able to search for other users and their posted events. A user shall be able to leave comments or sign up for other users' events. They shall also be able to post, edit, delete their own event activities, join groups, and sign up for public user activities.
  - 2.1. Look up the homepage and public events: same as a general use
  - 2.2. Search for public events: same as a general use
  - 2.3. Login System
    - 2.3.1. **Account:** already had an account
    - 2.3.2. **Email/Username:** need a unique email/username for login
    - 2.3.3. **Password:** need a password to login
  - 2.4. Search for and sign up for **events**
  - 2.5. **Log out** the system
  - 2.6. Leave **comments** on events
  - 2.7. Create/Edit/Delete **events**
  - 2.8. My Events: There shall be a link “**My Events**” which links to a web page of all the user's created and sign-upped events
3. **Group:** Group is where people of similar interests form a circle to create and attend events together.
  - 3.1. All groups consist of below information:
    - 3.1.1. **Name:** Group name
    - 3.1.2. **Group admin:** who created this group and can administer it.
4. **Group users:** Users who have joined a specific group and have more privilege than general users in terms of viewing and subscribing to group events which are private to the group.
  - 4.1. A group user is also a **registered user** and has all the attributes and privileges same as a registered user.
  - 4.2. A group user can comment and sign up for **group events** that are private to this group.
  - 4.3. **Sign out** the group
  - 4.4. Create/Edit/Delete **group events**

- 4.5. **My Groups:** There shall be a link “**My Groups**” which links to a web page of all the user’s joined groups
5. **Group Admin:** Administrator of a group, who is also the creator of the group and has the rights to delete inappropriate group events, to add users into the group, and remove group users who violate “PlayDate” terms of use.
- 5.1. A group user is also a **registered user** and has all the attributes and privileges same as a registered user.
  - 5.2. A group admin can delete **group events** if they’re inappropriate
  - 5.3. A group admin can **remove group users** if they violate “PlayDate” terms of use
  - 5.4. A group admin shall have an **group\_admin\_id** attribute to denote the identification
  - 5.5. A group admin can add registered users into the group
6. **Account:** general users can register the “PlayDate” system, and every user will have an account.
- 6.1. Each account will contain an **account\_id** and a **profile**
  - 6.2. A profile contains basic information of a user including **name**, **username**, **DOB**, **address**, and **dependents info**
  - 6.3. An account can use **several roles**, like **registered user**, **admin**, **group user**, and **group admin**, to denote which user is related to this account
7. **Roles:** including **registered user**, **admin**, **group user**, and **group admin**. Every registered user has an account and every account has one or more roles.
8. **Support Staff:** are the ones with whom the users can connect incase of any issues with the application by choosing Help on web application. The issue can be classified as one of below:
- 8.1. All support staff consist of below information:
    - 8.1.1. **Staff\_id:** identifies each staff
    - 8.1.2. **Name:** staff name
  - 8.2. **Request assistance:** The users who need assistance with usability or onboarding can contact the support team. The request consists of:
    - 9.1.1. **User Name**
    - 9.1.2. **Description of Assistance**
    - 9.1.3. **Mode of contact & Details** (phone or email and it’s details)
  - 8.3. **Report users/groups:** Registered users can report other users or groups that do not follow community guidelines or are causing bad experiences. User will click on the report option on the user profile or group profile of the user/group that needs to be reported and provide a reason in the description box.
  - 8.4. **Report bugs:** Users can report any bugs with the application and the service team will contact the user and get it fixed. User needs to provide below details while reporting bug:

- 8.4.1. **Description of Bug**
  - 8.4.2. **Contact Details**
- 9. **Backend Admin:** Work on technical issue users have reported via support staff.
  - 9.1. All backend admins have the following information:
    - 9.1.1. **Backend\_admin\_id:** identifies each admin
    - 9.1.2. **Name:** Staff name
  - 9.2. Backend admins will update background-checking when a user registers the “PlayDate” system. On successful verification, backend admins initiate the account activation. An admin is also a **registered user** and has all the attributes and privileges same as a registered user.
  - 9.3. Backend admin can delete **events** if they’re inappropriate
  - 9.4. Backend admin can **remove registered users** if they violate “PlayDate” terms of use
  - 9.5. Admin shall have an **admin\_id** or **is\_admin** attribute to denote the identification
- 10. **Events:** An event is a combination of date and place where a group of registered users can meet with their dependents such as children or pets. An event is created by a registered user and is open to only registered users on PlayDate. All events consist of below information:
  - 10.1. **Name:** Event name
  - 10.2. **Created by:** Event is created by which registered user
  - 10.3. **Address:** Venue of the event
  - 10.4. **Time and Date:** Details on when the event is scheduled to occur
- 11. **Public Events:** Upcoming public events are just seeders which are posted for general users to view and do a search. These public events are scrapped from other websites to give “PlayDate” general users an idea of what events will be happening around them and they can take their children or pets there. All public events consist of below information:
  - 11.1. **Name:** Event name
  - 11.2. **Address:** Venue of the event
  - 11.3. **Time and Date:** Details on when the event is scheduled to occur
- 12. **Group Event:** An event tied to a specific group. The group users of the group the event is attached to may register their attendance or *sign up*. While registration is open to the entire group, there is still an internal list of group users confirmed as group. A group event is an event, it has all the attributes same as an event
  - 12.1. **Name:** Event name
  - 12.2. **Created by:** Event is created by which registered user
  - 12.3. **Address:** Venue of the event
  - 12.4. **Time and Date:** Details on when the event is scheduled to occur
  - 12.5. **Group\_id:** denote in which group this group event is generated, and only groups users from this group can view and sign up for these events.

13. **Comment:** A piece of user-generated content attached to an event. A comment has all the below attributes
  - 13.1. **Event\_id:** under which event this comment is attached
  - 13.2. **Created by:** who attached this comment.
  - 13.3. **Time and Date:** Details on when the comment is attached
14. **Dependents:** Children or pets that are under the preview of a user. A dependent has all the below attributes.
  - 14.1. **Name:** Name of dependent
  - 14.2. **Age:** Age of dependent will be used in case of matching playdates of same age.
  - 14.3. **Interests:** These are likes based on which the registered users want to match dependents with playdates
15. **Survey:** A survey is generated by a group user to know about the group users' preferences on food or drinks. A survey can better help with a successful event. A survey has all the below attributes.
  - 15.1. **Topic:** topic of the survey
  - 15.2. **Generator:** who generated this survey
  - 15.3. **Event\_id:** related to which event, this survey is generated
16. Emergency Contact: Contact details of two persons, who are known to the registered user.
  - 16.1. Contact name
  - 16.2. Contact number

## 2. Prioritized Functional Requirements

### Priority 1

1. General User
  - 1.1. A general user shall be able to view public events.
  - 1.2. A general user shall be able to search for public events based on location.
  - 1.3. A general user shall be able to filter search results of public events based on category of kids.
  - 1.4. A general user shall be able to filter search results of public events based on category of pets.
  - 1.5. A general user shall be able to register.
  - 1.6. A general user shall be able to create one account.
  - 1.7. A general user shall be able to become only one registered user.
  - 1.8. A general user shall be able to upload proof of the parent of a kid or pet.
  - 1.9. A general user shall be able to request assistance from the PlayDate support staff for onboarding.
  - 1.10. A general user shall be able to view Terms of Use.
  - 1.11. A general user shall be able to view Privacy Notice.
  - 1.12. A general user shall be able to agree to Terms of Use while signing up.
  - 1.13. A general user shall be able to agree to Privacy Notice while signing up.
2. Registered User
  - 2.1. A registered user shall be able to view Terms of Use.
  - 2.2. A registered user shall be able to view Privacy Notice.
  - 2.3. A registered user shall be able to log into their account.
  - 2.4. A registered user shall have a profile.
  - 2.5. A registered user shall be able to have a username.
  - 2.6. A registered user shall be able to have an email address.
  - 2.7. A registered user shall have an address in their profile.
  - 2.8. A registered user shall be able to add many dependents.
  - 2.9. A registered user shall be able to edit their Username in profile
  - 2.10. A registered user shall be able to edit their Email in profile.
  - 2.11. A registered user shall be able to edit their Address in profile
  - 2.12. A registered user shall be able to edit their Name in profile
  - 2.13. A registered user shall be able to add many dependents to their profile..
  - 2.14. A registered user shall be able to delete their dependents from their profile.
  - 2.15. A registered user shall be able to edit their Dependents' Name in profile
  - 2.16. A registered user shall be able to edit their Dependents' Birth Date in profile
  - 2.17. A registered user shall be able to edit their Dependents' Type in profile
  - 2.18. A registered user shall be able to edit their Dependents' Interests in profile
  - 2.19. A registered user shall be able to edit their Birth Date in profile
  - 2.20. A registered user shall be able to edit their Dependents' schedule
  - 2.21. A registered user shall be able to become a group user.
  - 2.22. A registered user shall be able to create many groups.
  - 2.23. A registered user shall be able to become a group admin.

- 2.24. A registered user shall be able to search for groups based on location
- 2.25. A registered user shall be able to search for groups based on interest
- 2.26. A registered user shall be able to search for groups based on group name
- 2.27. A registered user shall be able to join many groups.
- 2.28. A registered user shall be able to search for events from all groups they are a part of.
- 2.29. A registered user shall be able to request to join a group.
- 2.30. A registered user shall be able to search for other registered users based on location.
- 2.31. A registered user shall be able to create public events.
- 2.32. A registered user shall be able to search for other registered users via a filter of common interests.
- 2.33. A registered user shall be able to be part of a group.
- 2.34. A registered user shall be able to send notification to a selected number of registered users via application.
- 2.35. A registered user shall be able to search for public events based on location.
- 2.36. A registered user shall be able to request for technical assistance from playdate support staff on product bugs.
- 2.37. A registered user shall be able to log out.
- 2.38. A registered user shall be able to view public events
- 2.39. A registered user shall be able to filter the search result of public events based on category of kids.
- 2.40. A registered user shall be able to filter the search result of public events based on category of pets.

### 3. Group admin

- 3.1. A group admin shall also be a registered user.
- 3.2. A group admin shall also be a group user.
- 3.3. A group admin shall be able to administrate at least one group
- 3.4. A group admin shall be able to add or remove many group users from the group.
- 3.5. A group admin shall be able to accept the join request of the registered user.
- 3.6. A group admin shall be able to deny the join request of the registered user.
- 3.7. A group admin shall be able to delete an event created by the users of their group.

### 4. Group User

- 4.1. A group user shall be able to sign up for many group events.
- 4.2. A group user shall be able to sign off on many group events.
- 4.3. A group user shall be able to delete the group events created by them.
- 4.4. A group user shall also be a registered user.
- 4.5. A group user shall be able to create an event in a group.
- 4.6. A group user shall be able to post the event in the group.
- 4.7. A group user shall be able to request for technical assistance from playdate support staff on product bugs.

- 4.8. A group user who creates a group event is that event's event admin
  - 4.9. A group user shall be able to leave a group
5. Support staff
  - 5.1. A support staff shall be able to respond to queries from general users.
  - 5.2. A support staff shall be able to respond to queries from registered users.
  - 5.3.
  - 5.4. Support staff shall be able to receive emails from registered users with issues details.
  - 5.5. Support staff shall be able to receive emails from general users with issues details.
6. Public Events
  - 6.1. A Public Event shall be viewed by General Users
  - 6.2. A Public Event shall be viewed by Registered Users
  - 6.3. A Public Event shall be searched for by General Users
  - 6.4. A Public Event shall be searched for by Registered Users
  - 6.5. A Public Event shall be searched based on location search criteria.
  - 6.6. A Public Event shall be filtered based on type of Pets.
  - 6.7. A Public Event shall be filtered based on type of Kids.
7. Dependents
  - 7.1. Dependents shall have a name
  - 7.2. Dependents shall have a birth date
  - 7.3. Dependents shall have a type
  - 7.4. Dependents shall have a list of interests
  - 7.5. Dependents shall have an availability schedule
  - 7.6. Dependents shall be managed by their associated registered user
  - 7.7. A dependents Name shall be editable by the Registered User who manages the dependent
  - 7.8. A dependents Birth Date shall be editable by the Registered User who manages the dependent
  - 7.9. A dependents Type shall be editable by the Registered User who manages the dependent
  - 7.10. A dependents Interests shall be editable by the Registered User who manages the dependent
  - 7.11. A dependents Schedule shall be editable by the Registered User who manages the dependent
8. Groups
  - 8.1. A group shall have at least one group user.
  - 8.2. A group shall have at least one group admin.
  - 8.3. A group shall be able to have zero or more posts.
  - 8.4. A group shall be able to have zero or more comments on each post.
  - 8.5. A group shall be able to be formed by 50 group users.
  - 8.6. A group shall be able to have 0 or more group events.
  - 8.7. A group shall be searched by Registered User based on group location
  - 8.8. A group shall be searched by Registered User based on group interest

- 8.9. A group shall be searched for by Registered Users according to group name
- 8.10. A group shall be joined by many registered users who will become group users.
- 8.11. A group shall have a list of Registered Users which can be modified by the group admin
- 8.12. A group shall allow for events to be created with all group users as possible attendees.
- 8.13. A group shall include posts by it's Group Users
  
- 9. Group Events
  - 9.1. A group event shall be able to immediately accept signup from group users of that group.
  - 9.2. A group event shall be able to be edited by the group user who created that group event.
  - 9.3. A group event can be deleted by the group user who created that group event.
  - 9.4. A group event can be searched based on whether the User doing the searching belongs to the group the event belongs to.
  - 9.5. A group event shall be signed up by group users till the maximum capacity is reached.
  - 9.6. A group event can be deleted by the group admin.
  - 9.7. A group event shall be able to allow group users to sign up.
  
  - 9.8. A group event with no sign-ups by start of group event time shall be able to be canceled.

## **Priority 2**

- 1. Registered User
  - 1.1. A registered user shall be able to send out public event invites to a filtered user list.
  - 1.2. A registered user shall be able to request emergency assistance via PlayDate application.
  - 1.3. An emergency request shall be sent to the nearest police via 911 emergency helpline.
  - 1.4. An emergency request shall contain registered user's event location and contact number.
  - 1.5. A registered user shall be able to create a survey for an event.
  - 1.6. A registered user who is attending the event shall be able to respond to surveys corresponding to that event.
  - 1.7. A registered user who created the survey shall be able to delete the survey.
  - 1.8. A registered user who created the survey shall be able to modify it.
  - 1.9. A registered user shall be able to report other users by contacting playdate support staff.
  - 1.10. A registered user shall be able to set interest on any upcoming events that they have access to.
  - 1.11. Interested events of a registered user shall be listed under the interested category.
  - 1.12. A registered user shall be able to filter to RSVP'd events
  - 1.13. A registered user when unsure of their attendance shall be able to set 'Maybe' on any upcoming events that they have access to.
  - 1.14. An event that is set 'Maybe' shall be listed under the 'Maybe' category.
  - 1.15. A registered user shall be able to view their list of RSVP'd events

- 1.16. A registered user shall be able to view their list Interested events
    - 1.17. A registered user shall be able to view their list of Maybe events
  - 1.18. A registered user shall be able to add friends to their account.
  - 1.19. A registered user shall be able to send a referral link to a friend.
2. Group admin
    - 2.1. A group admin can send invites of group join to a selected list of registered users.
    - 2.2. A group admin shall be able to delete any surveys on events of their group.
  3. Group User
    - 3.1. A group user shall be able to create a survey for an event of their group.
    - 3.2. A group user shall be able to post surveys(polls) for group events in the group.
    - 3.3. A group user who is attending the event shall be able to respond to surveys corresponding to that group event.
    - 3.4. A group user who created the survey shall be able to delete the survey.
    - 3.5. A group user who created the survey shall be able to modify it.
    - 3.6. A group user who receives a notification regarding the creation of a group event shall be able to sign up for the event through the notification.
    - 3.7. Posts shall raise notification to group users.
    - 3.8. A group user shall be able to see the notification of the event in their account.
    - 3.9. A group user shall be able to contact the playdate support staff to report other group users.
    - 3.10. A group user shall be able to create a post in a group.
    - 3.11. A group user shall be able to reply to group posts with emojis.
    - 3.12. A group user shall be able to upload image to their post
    - 3.13. A group user shall be able to rate an event after the event is completed.
    - 3.14. A group user shall be able to send group invite to registered users to join the group.
  4. Support Staff
    - 4.1. Support staff shall be able to verify the concern raised by group users by reporting other group users.
  5. Emergency contact
    - 5.1. An emergency contacts shall be able to receive emergency request from the registered user
    - 5.2. An emergency contact shall be able to receive registered users' event location and phone number.

### **Priority 3**

1. General User
  - 1.1. A general user shall be able to comment on public events
2. Registered User

- 2.1. A registered user shall be able to comment on public events
  - 2.2. A registered user can schedule a recurring event.
  - 2.3. A registered user shall be able to create a list of friends.
  - 2.4. A registered user shall be able to label the list of friends.
  - 2.5. A registered user shall be able to add events to their favorites
  - 2.6. A registered user shall be able to view all of their favorite events.
3. Group User
  - 3.1. A group user who is part of a group shall be able to view a heatmap of the group schedule.
  - 3.2. A group user shall be able to update their availability on group heatmap.
  - 3.3. A group user shall be able to sign up for an event via notification on their account.
  - 3.4. A group user who created the event shall be able to create a post to collect reviews after its occurrence.
  - 3.5. A group user who attended the event shall be able to give a rating out of 5 stars on the event by clicking on stars.
  - 3.6. An event shall contain the cumulative average of all the ratings as the rating of that event.
  - 3.7. 7.3 A group user shall be able to edit their posts
  - 3.8. 7.4 A group user shall be able to delete their posts
  - 3.9. 7.5 A group user shall be able to edit a comment on a group
  - 3.10. 7.6 A group user shall be able to delete a comment on a group
4. Backend Admin
  - 4.1. A backend admin shall be able to access the user verification portal.
  - 4.2. A backend admin shall be able to verify the general user's identity to confirm his registration.
  - 4.3. A backend admin shall be able to access all the group content of posts
  - 4.4. A backend admin shall be able to access all the group content of events
  - 4.5. A backend admin shall be able to access all the public content of events
  - 4.6. A backend admin shall be able to remove registered user from the application.
  - 4.7. A backend admin shall be able to remove group user from the group.
  - 4.8. A backend admin shall be able to delete a group.
5. Group
  - 5.1. A group shall have a heatmap of availability of it's group users.

### 3. UI Mockups & Storyboard

#### 1. Use Case 1: Registration

**User Case 1:**

**PlayDate** Enter Zip Code to Search  Sign Up Log In Help

Fun Places  
Upcoming Events

Mary

- Mary can browse all listed fun places and upcoming events on PlayDate.
- She can also search for public info by zipcode.
- Mary wants to register so that she can view more events.

Many clicks on "Sign Up" button

PlayDate Sign Up

Email:   
Username:   
Password:   
Re-enter password:

Sign Up Log In Already Got An Account?

Mary Entered all the required info and registered PlayDate.

After registration, Mary automatically logged in and see a different page.

PlayDate Search For PlayDate  +

Fun Places  
Upcoming Events  
Neighbor Events  
Groups  
My Groups  
My Events

Mary

- After logging in, Mary browsed some Neighbors Events, and find a post from her friend, Helen Lee.
- Mary liked Helen's post, but didn't sign up because of time conflict. She then clicked on Helen's name.

Many clicks on Helen's name

PlayDate Search For PlayDate  +

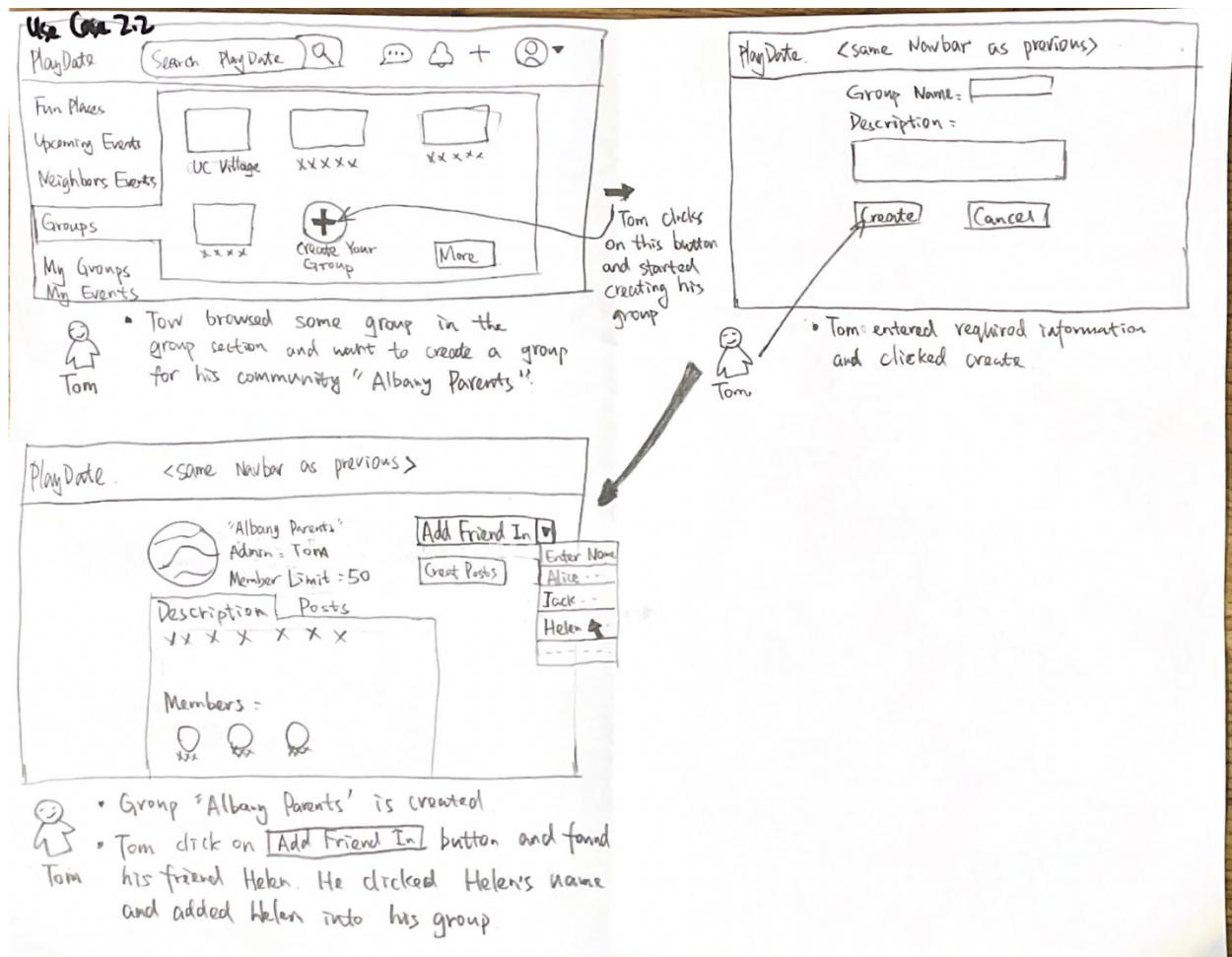
Helen Lee Add Friend

Her posts!

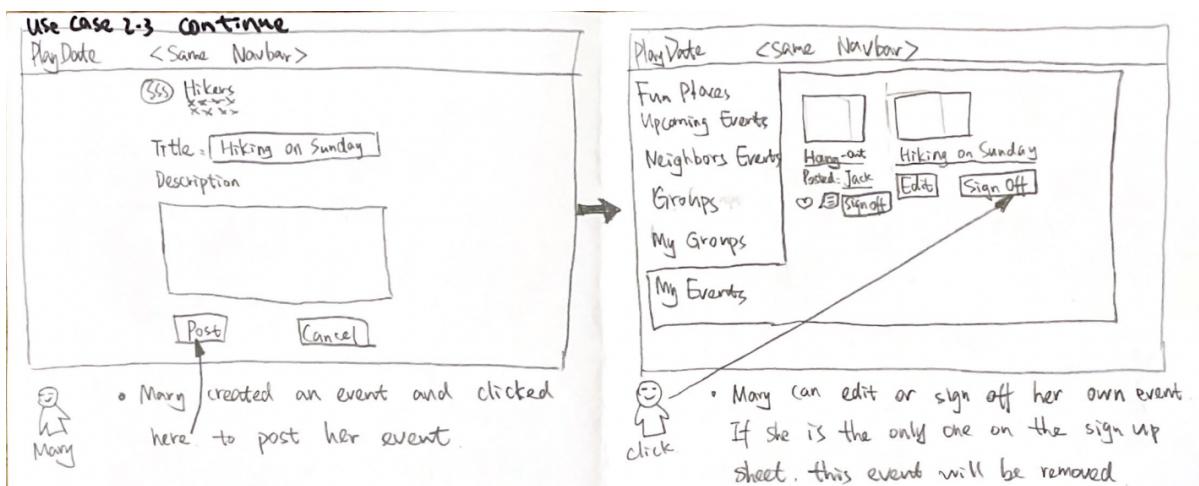
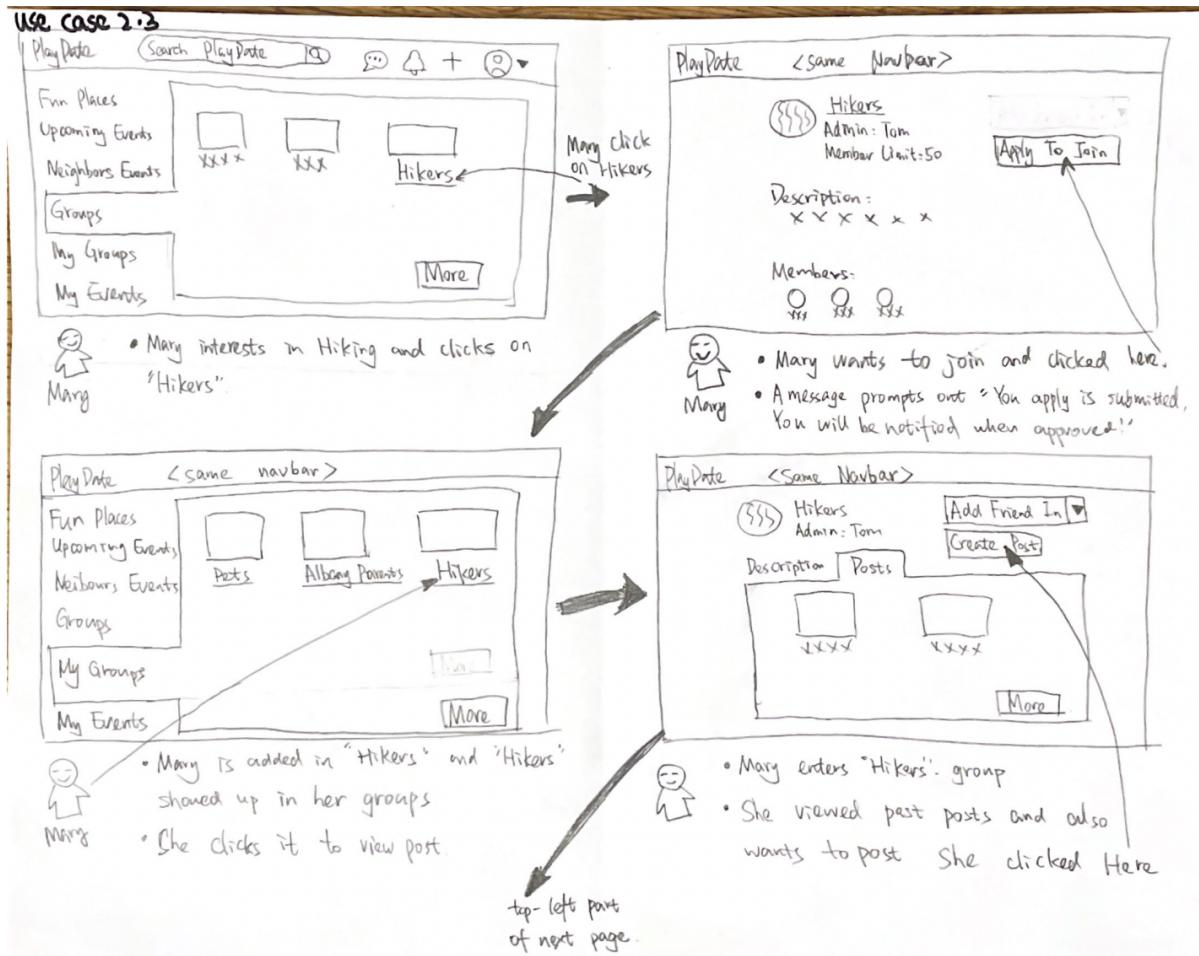
Mary

- After clicking on Helen's name, Mary is directed to Helen's personal page.
- Mary can view all Helen's post pics and can add her as friend.

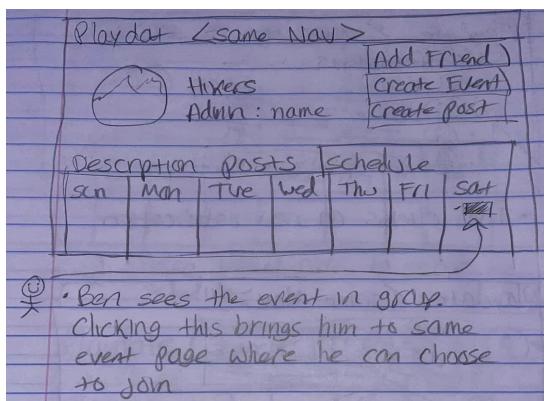
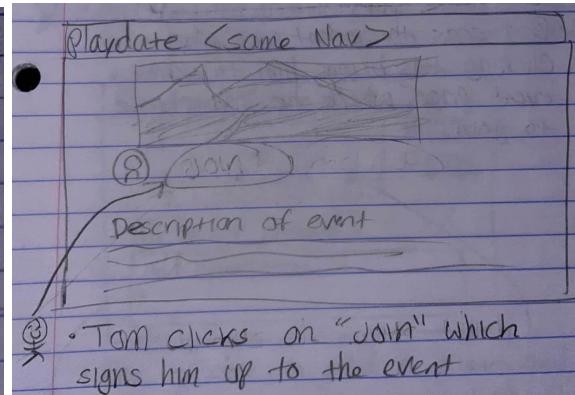
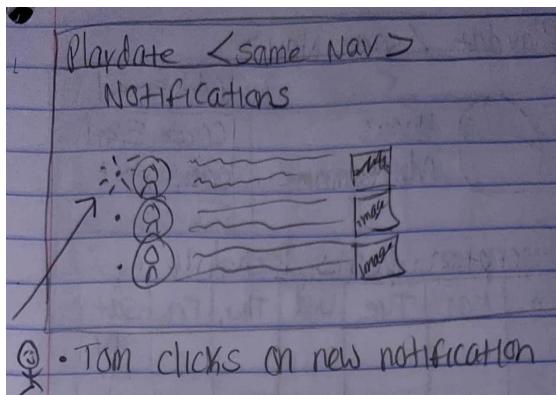
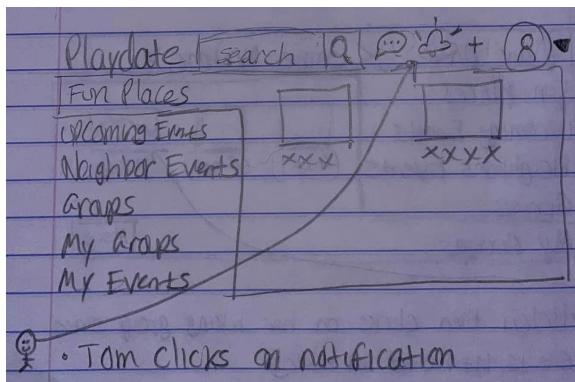
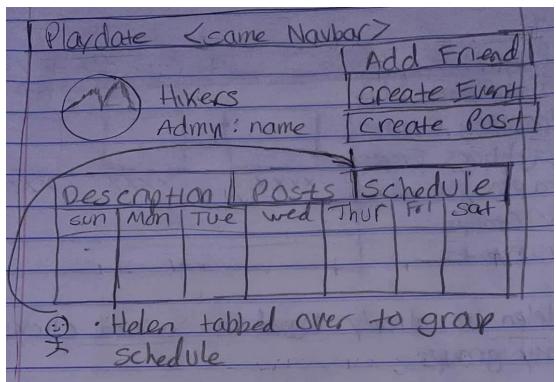
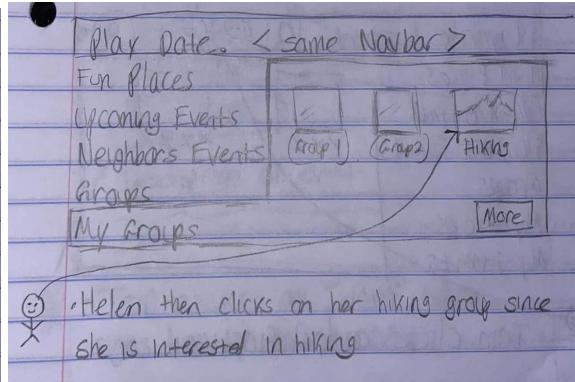
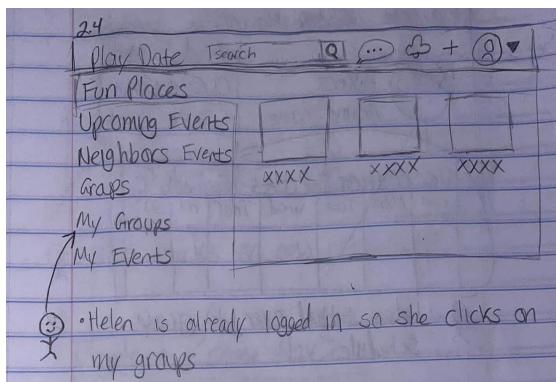
## 2. Use Case 2: Group Creation



### 3. Use Case 3: Joining group



#### 4. Use Case 4: Creating Group Events



## 5. Use Case 5: Creating Public Events

2.5

Playdate < same Nav >

- Fun Places
- Upcoming Events
- Neighbor Events
- Groups
- My Groups
- My Events

event name      event name

• On the "My Events" tab from main page Eric clicks on "Create Event" button.

Playdate < same Name >

+ add photo

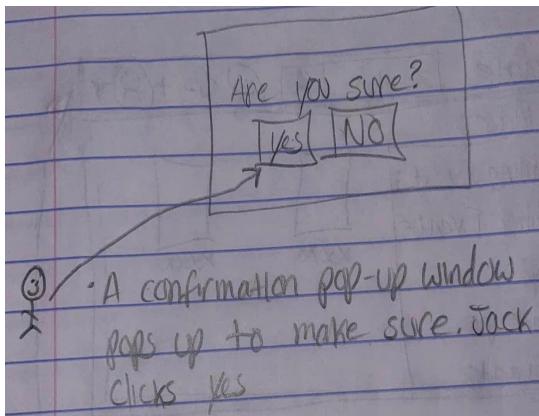
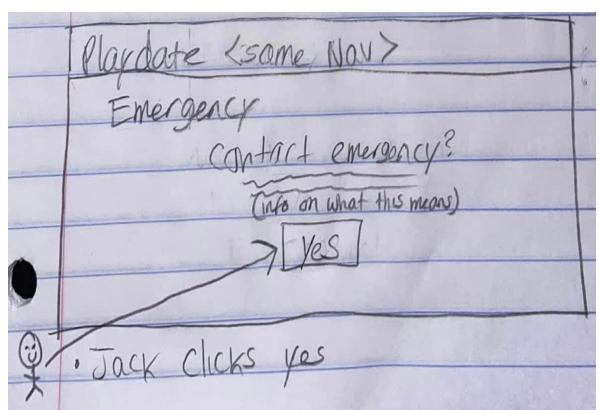
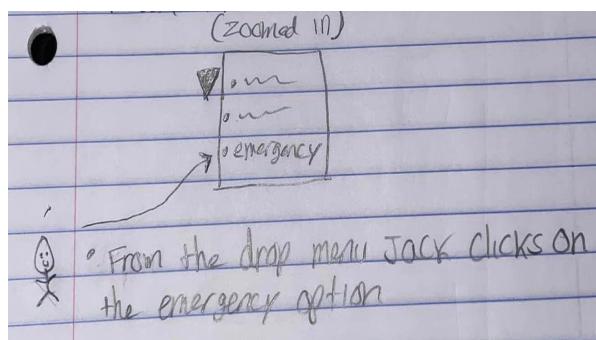
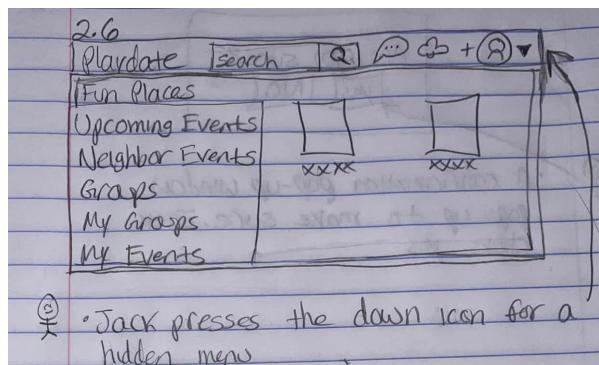
make public  
 make private

(8) invite friends

add description...

• Eric is able to add a photo and description to the event, both clickable.  
• Eric is also able to add friends to an invite list, invites will be sent after event is created  
• Eric checks the "make public" check box to make the post public

## 6. Use Case 6: Emergency Assistance



## 7. Use Case 7: Surveys for the event

PlayDate		
Fun Places	<input type="checkbox"/>	<input type="checkbox"/>
Upcoming Events	<input type="checkbox"/>	<input type="checkbox"/>
Neighbor Events	Dog Spa	Dog Walk
Groups	<input type="checkbox"/>	<input type="checkbox"/>
My Groups		Dog park visit
My Events	<input type="checkbox"/>	<input type="checkbox"/>
	More	
	Birthday Party	Study Circle
Jeff	clicks	

Jeff visit upcoming events on his Playdate account. he clicks on BirthDay Party event.

Jeff doesn't what to take to birthday party of his child's friend so he decides to create a survey.

PlayDate		
Fun Places	<input type="checkbox"/>	<input type="checkbox"/>
Upcoming Events	<input type="checkbox"/>	<input type="checkbox"/>
Neighbor Events	<input type="checkbox"/>	<input type="checkbox"/>
Groups	<input type="checkbox"/>	<input type="checkbox"/>
My Groups		
My Events		
	TPmeB Venue	
	Create Survey	
	View Survey	
	Cancel	

The application takes Jeff to 'BirthDay Party' Event Page. The Page has all details of the event . On the Right Side he clicks on create survey .

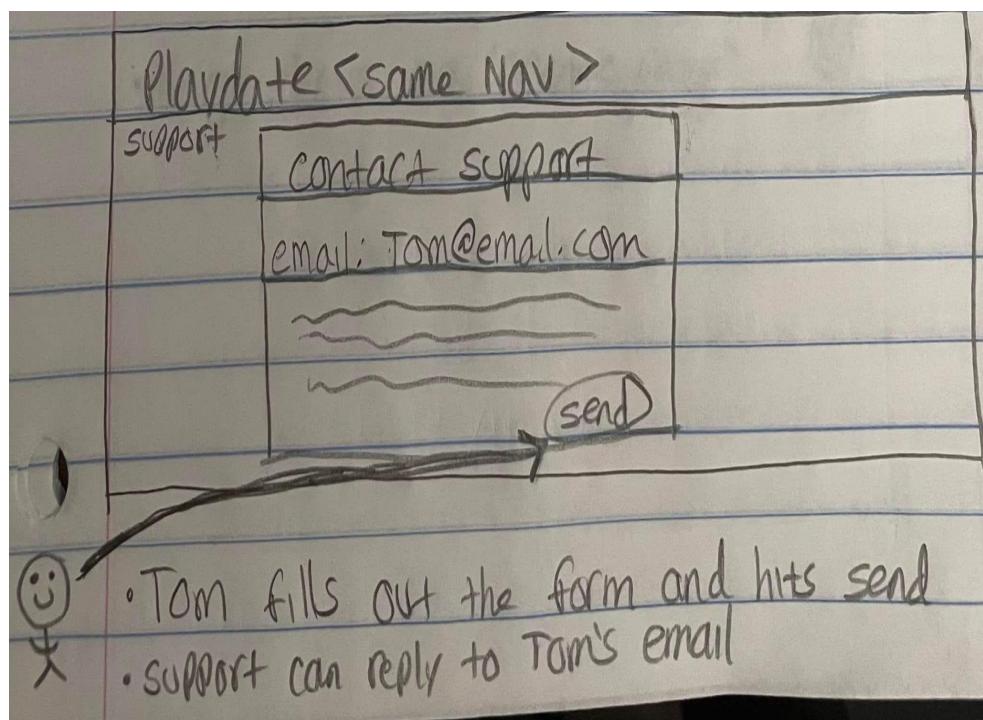
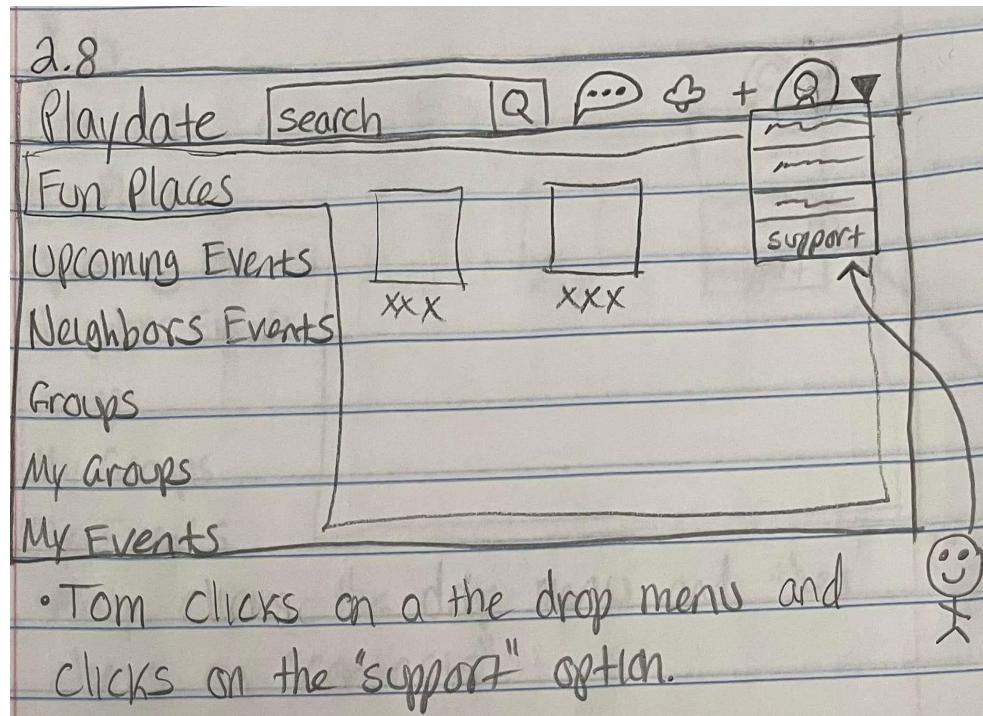
Play Date		
Fun Place	Search	
Upcoming Events	BirthDay Party	
Neighbor Events		
Groups		
My Groups		
My Events		
	What beverage do kids like?	
	<input type="radio"/> O xxxx xx	
	<input type="radio"/> O xxxx xx	
	<input checked="" type="radio"/> O xxxx xx	
	Add Question	
	Upload Image	
	Submit	

Jeff Adds question using 'add question' button, which creates a text field for question & Radio button for response . Next he clicks submit to confirm survey creation .

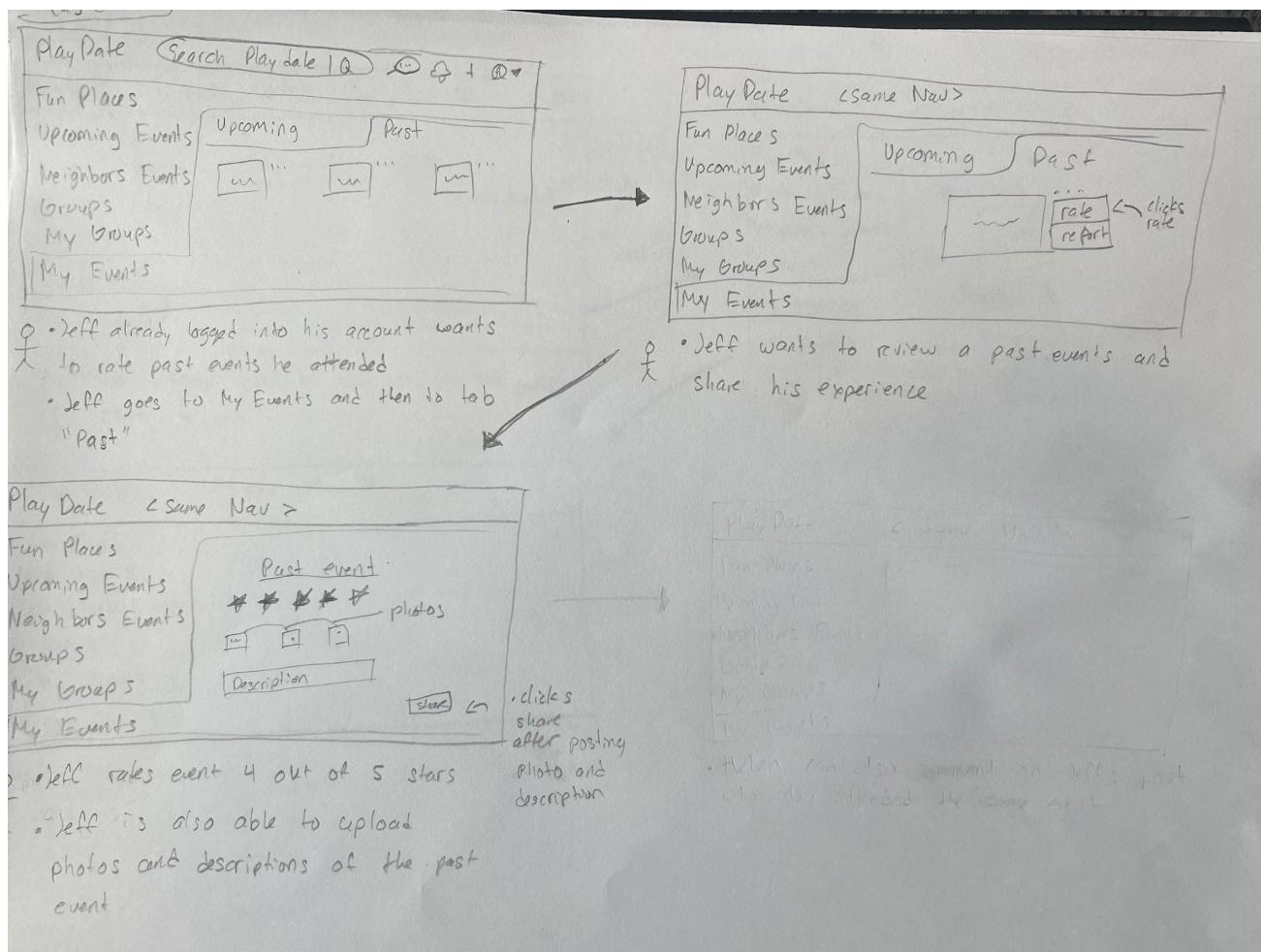
PlayDate		
Fun Places	Search	
Upcoming Events		
Neighbor Events		
Groups		
My Groups		
My Events		
	What beverage do kids like	
	<input type="radio"/> O xxxx xx	
	<input type="radio"/> O xxxx xx	
	<input type="radio"/> O xxxx xx	
	Submit	
	Cancel	

Helen one of group event attendee responds to survey and clicks on submit . Her response is now saved for Jeff to view later .

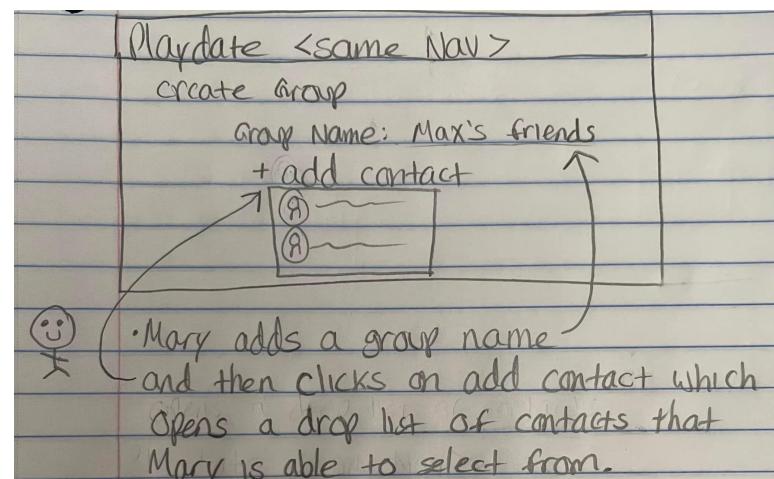
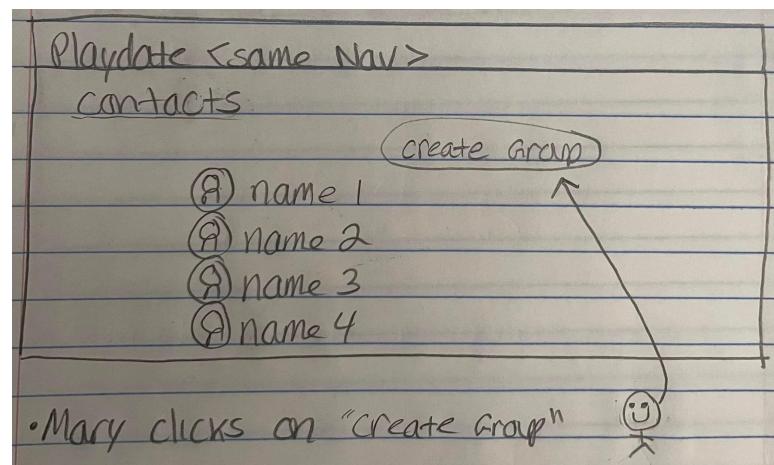
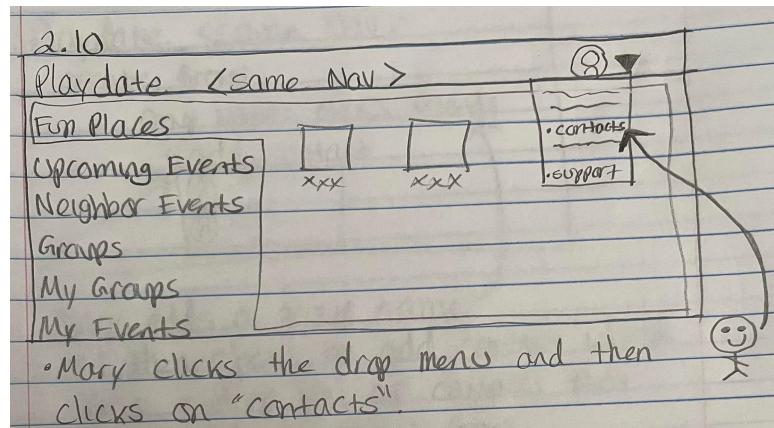
## 8. Use Case 8: Technical Support



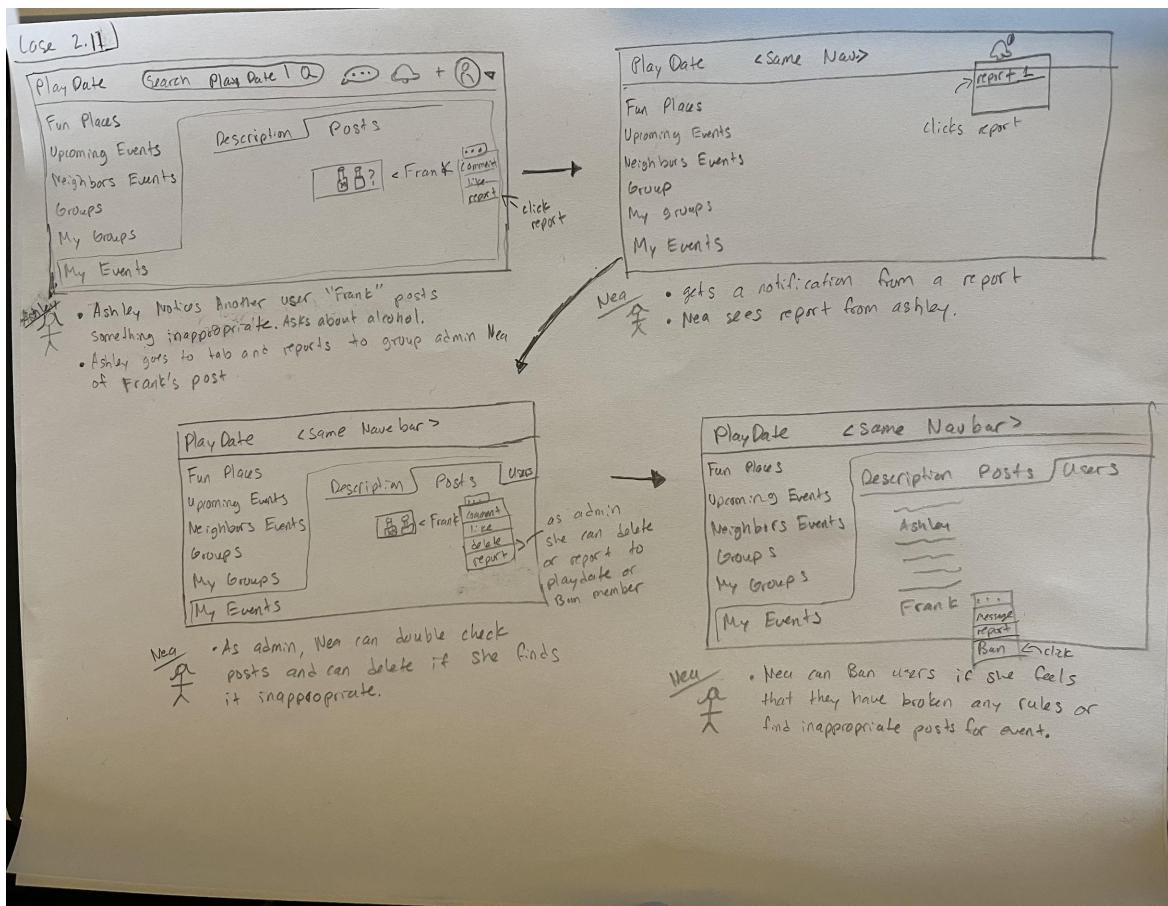
## 9. Use Case 9: Reviewing Event



## 10. Use Case 10: Grouping together friends list

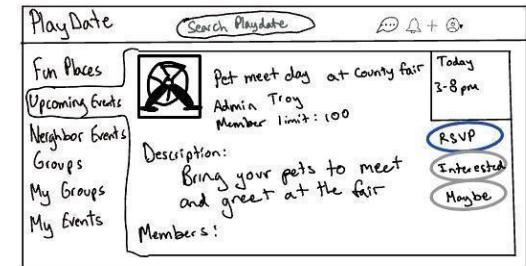
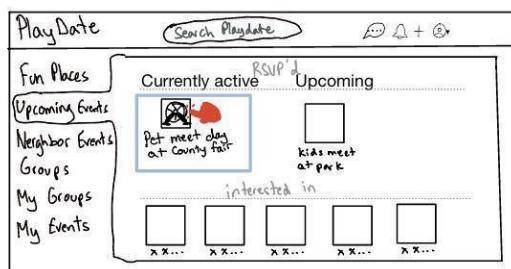


### **11. Use Case 11: Report content or user**



## 12. Use Case 12: View RSVP'd event

USE CASE 2.12

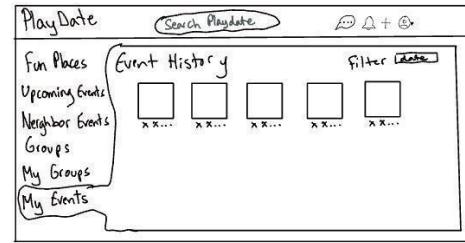
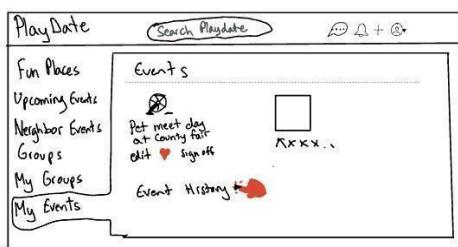


? Joe has a bunch of different events coming up that he RSVPed to as "going" but doesn't want to have to search past all the events that he has listed in general, ie. ones that he responded to as "interested" or "maybe".

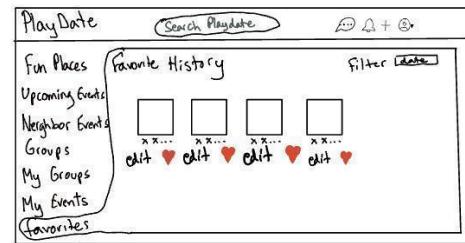
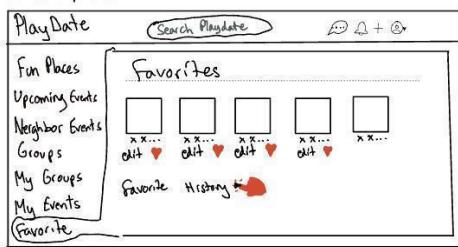
So he goes clicks on current event which lists only the events that he has RSVPed and has further details about the event

### 13. Use Case 13: Save Favorite Events

#### Use Case 2.13



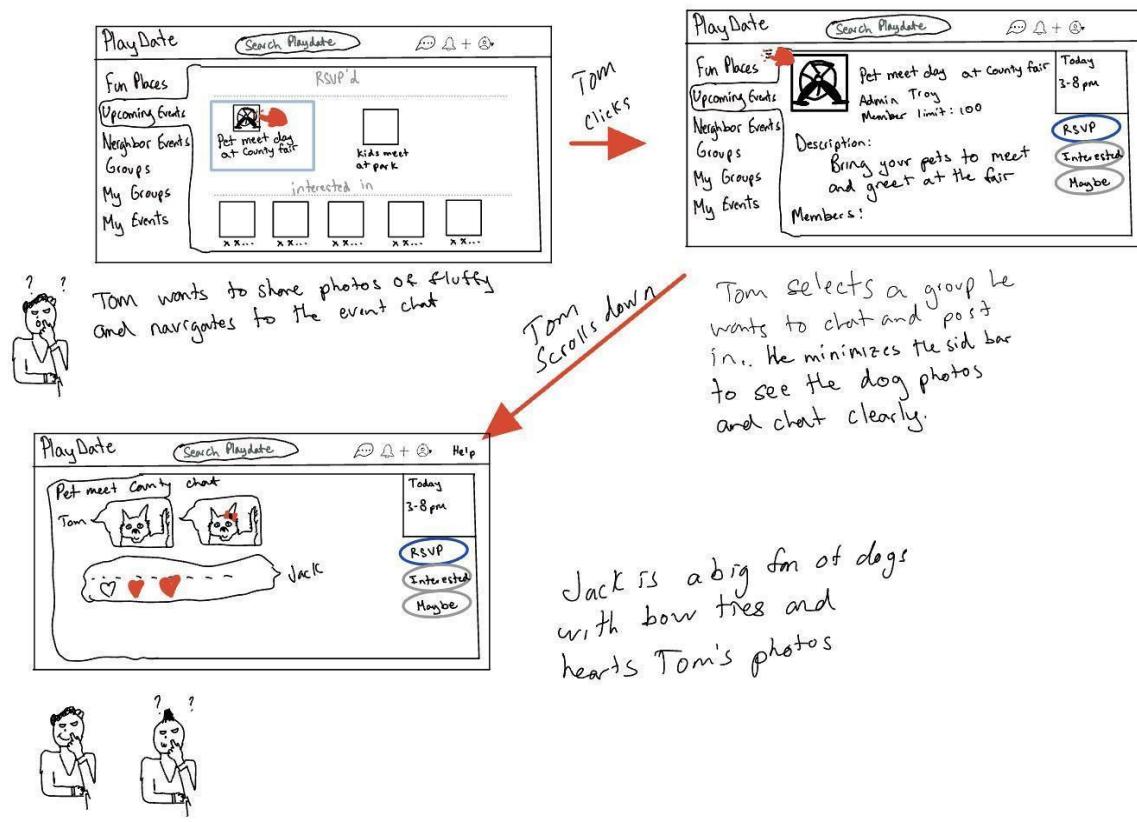
Tim loves the app and has already gone to many events. He wants a way to look back at not only his past events but to have an easy way to look back on all of his favorite past events. He should be able to create a list of favorites including past and present.



Clicks favorite history and has the option to filter it based on date, location,etc..

## 14. Use Case 14: Post in the group

### USE CASE 2.14



## 4. High level database architecture & organization

### 1. DB Organization

#### 1.1. Business Rules

##### i. General User

A general user shall be able to become a registered user.

A general user shall be able to create only one account.

##### ii. Registered User

A registered user shall be a general user.

A registered user shall have only one account.

A registered user shall be able to create/delete/edit many events.

A registered user shall be able to sign up for many events.

A registered user shall be able to join many groups.

##### iii. Admin

An admin shall be a registered user.

An admin shall be able to delete many events.

An admin shall be able to remove many registered users.

##### vi. Group User

A group user shall be a registered user.

A registered user shall be able to create/delete/edit many group events.

A registered user shall be able to sign up for many group events.

##### v. Group Admin

A group admin shall be a registered user.

A group admin shall be able to delete many group events.

A group admin shall be able to remove many group users.

##### vi. Account

An account shall be created by one and only one general user.

An account shall use many roles.

##### vii. Roles

A role shall be used by 0 or more accounts.

##### viii. Group

A group shall have at least one group user.

A group shall have one group admin.

A group shall have many group events.

ix. Support Staff

A support staff shall be contacted by many registered users.

A support staff shall be able to contact at least one technical staff when a technical issue is reported.

x. Public Events

A public event shall be viewed by many users.

A public event shall be searched by many users.

xi. Events

An event shall be created by only one registered user.

An event shall be signed-up by 0 or more registered users.

xii. Group Events

A group event shall be created by only one group user.

A group event shall be signed-up by 0 or more group users.

xiii. Dependents

A dependent shall be had by only one registered user.

## 1.2. Entities

i. General User (Strong)

\* general\_id: key, numeric

\* ip\_address: alphanumeric

ii. Registered User (Weak)

\* registered\_user\_id: strong key, numeric

\* general\_id: weak key, numeric

iii. Admin (Weak)

\* admin\_id: strong key, numeric

\* registered\_user\_id: weak key, numeric

vi. Group User (Weak)

\* group\_user\_id: strong key, numeric

\* registered\_user\_id: weak key, numeric

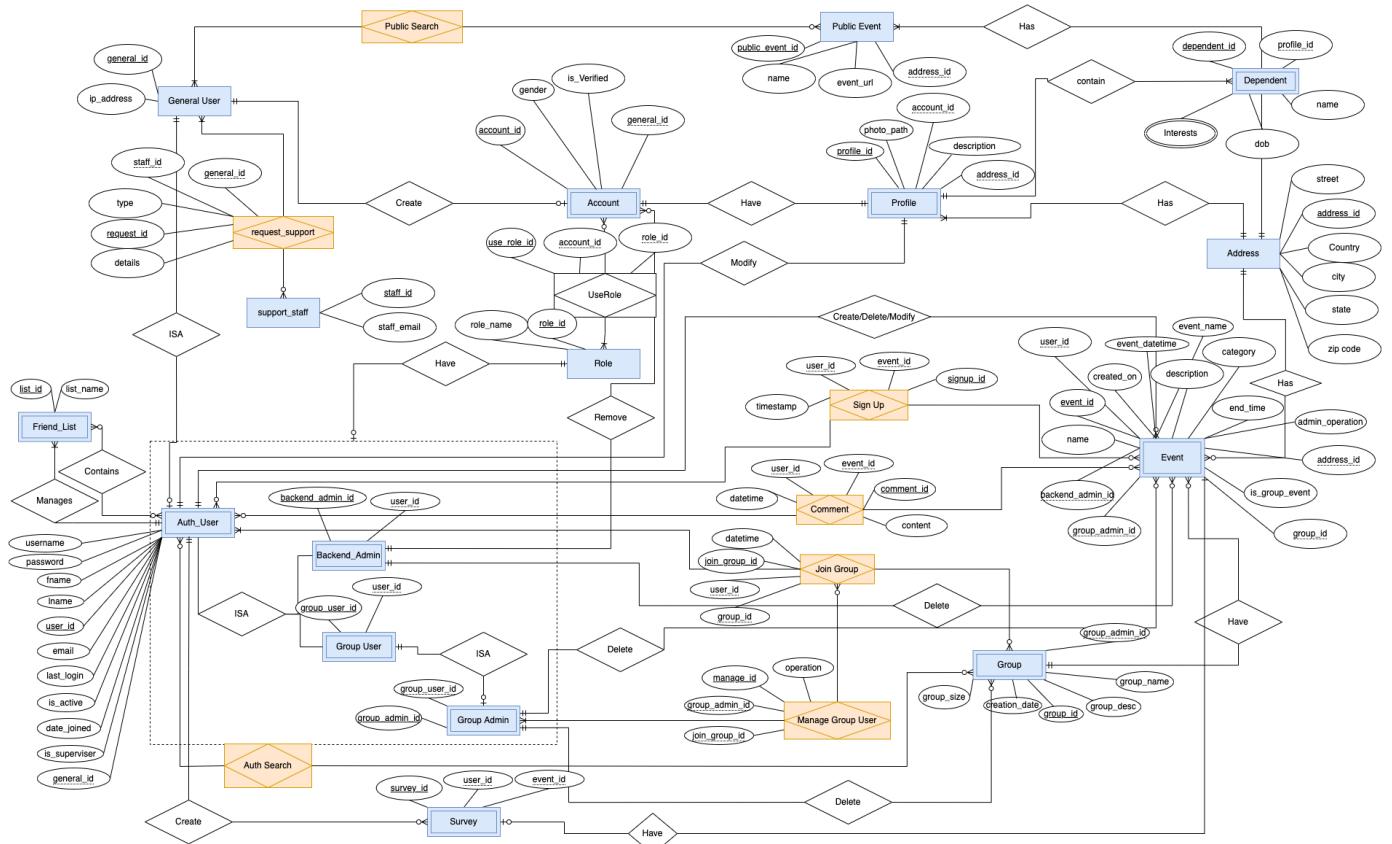
v. Group Admin (Weak)

\* group\_admin\_id: strong key, numeric

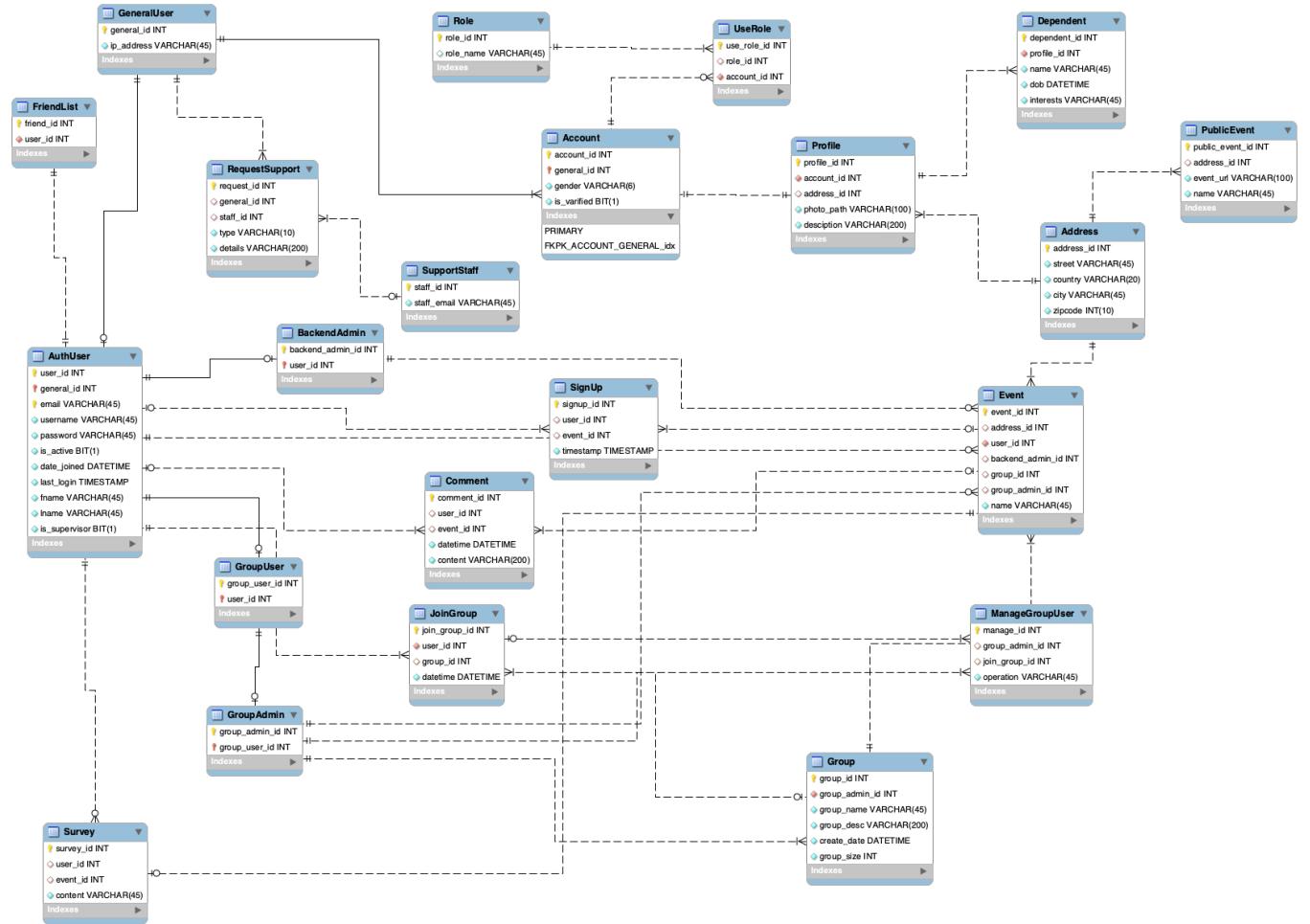
- \* registered\_user\_id: weak key, numeric
- vi. Account (Weak)
  - \* account\_id: key, numeric
  - \* user\_id: key, numeric
  - \* role\_id: key, numeric
- vii. Roles (Strong)
  - \* roles\_id: key, numeric
  - \* roles\_name: alphabetical
  - \* description: alphabetical
- viii. Group (Weak)
  - \* group\_id: strong key, numeric
  - \* group\_admin\_id: weak key, numeric
  - \* group\_name: alphanumeric
- ix. Support Staff (Strong)
  - \* staff\_id: key, numeric
  - \* staff\_name: alphanumeric
  - \* email: key, alphanumeric
- x. Public Events (Strong)
  - \* event\_id: strong key, numeric
  - \* content: alphanumeric
  - \* address: composite, street, city, state, zip code
  - \* datetime: datetime
- xi. Events (Weak)
  - \* event\_id: strong key, numeric
  - \* user\_id: weak key, numeric
  - \* content: alphanumeric
  - \* datetime: datetime
- xii. Group Events (Weak)
  - \* group\_event\_id: strong key, numeric
  - \* group\_id: weak key, numeric
  - \* group\_user\_id: weak key, numeric
  - \* content: alphanumeric
  - \* datetime: datetime
- xiii. Dependents (Weak)

- \* dependent\_id: strong key, numeric
- \* profile\_id: weak key, numeric
- \* interest: alphanumeric, composite
- \* dob: date
- \* name: alphanumeric, composite, first name, last name

### 1.3.ERD



## 1.4.Database Model (EER)



## 1.5. DBMS

We used MySQL Workbench to build our database model and generated our database generation script with forward engineering, so we directly used MySQL to generate our database model.

## 2. Media Storage

Our images and contexts will be stored in file systems.

## 3. Search/Filter architecture

### 1. Search Algorithm:

The search algorithm will consist of user input into the search bar on UI. The user may input event attributes, like locations and names. After parsing and confirming validity of the input, multiple queries will be sent to the database via Django QuerySet API . All

valid queries will return a list of event objects. All event objects will be used to render the web page.

2. Filter categories: The search can be further filtered for various categories, using the drop down menu which will append to query sets generated in the search algorithm.

## 5. High Level APIs and Main Algorithms

### APIs

Our application will be very important to advertisers, users, and others. As such, we want to include an API that allows for those clients to interact with our system without going through the html. We will include the following APIs:

- **Login:** A user will be able to login by supplying us with their username and password. This functionality will be required for all users of the API. The response will indicate whether the login was successful.
- **Create a Public Event:** Through the API, public events may be created. To do so, we require a name, a description, a category, an address, a date, and a URL. Public events submitted this way will still need to be moderated by our back-end staff. The response will indicate whether the request was parsed correctly by the application.
- **Create a Group Event:** If the logged in user is the admin of a group, they shall be able to use our API to create an event for that group. This will require the user to supply the group name, the event name, an event description, the address, the start time, the end time, and the category. The response will indicate if everything was successfully parsed.

With this functionality provided, there are key functions which will not be part of the API, requiring all users to do this through the HTML side of our application. Those functions include registering a user, creating a group, and other easily abused parts of the application.

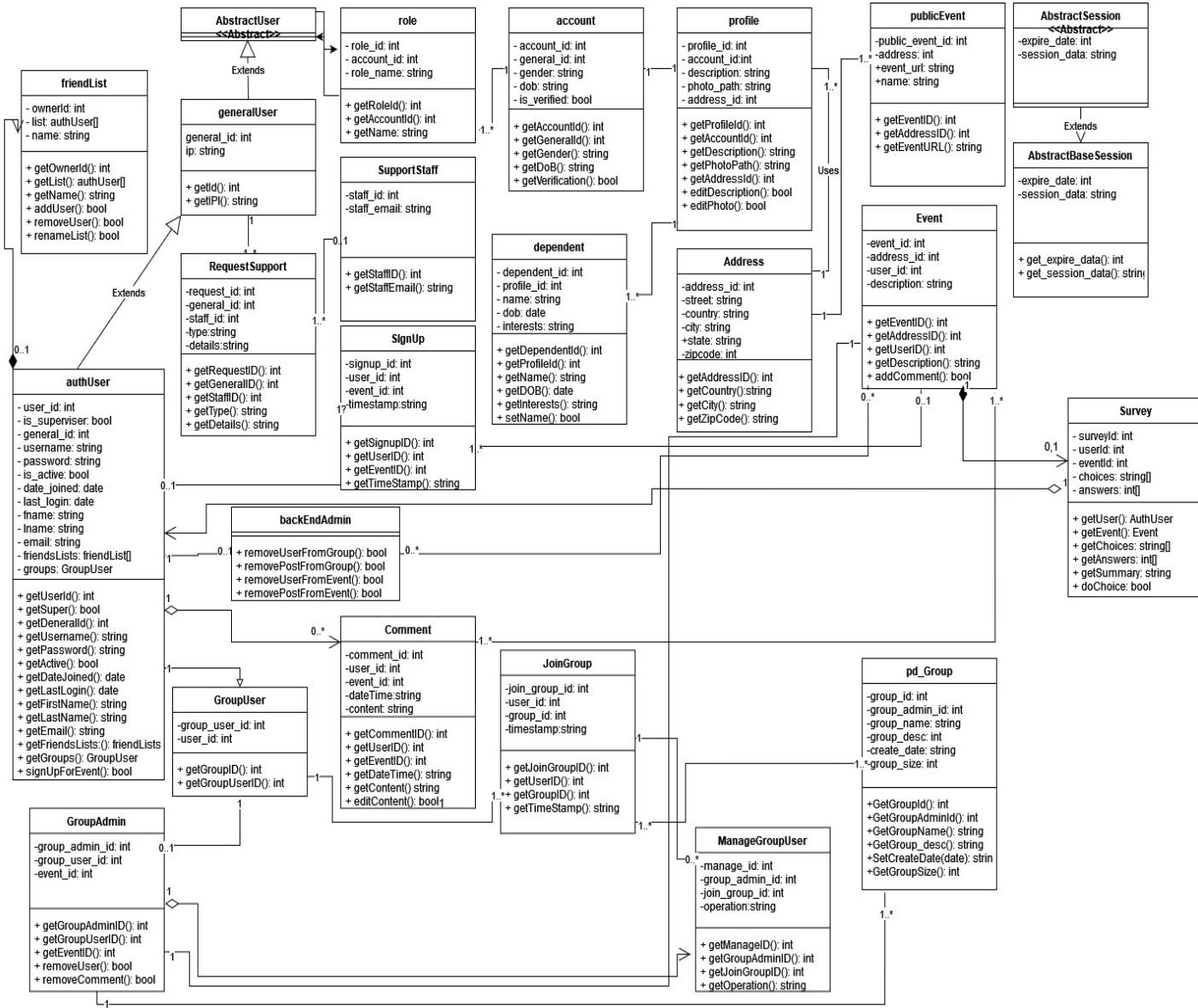
### Algorithms

We are implementing a search system so that users can search events and groups. This search algorithm depends on the similarity concept, where the results are retrieved in the order of best match first and least match last. To perform the database queries we use the Django QuerySet API.

### Changes

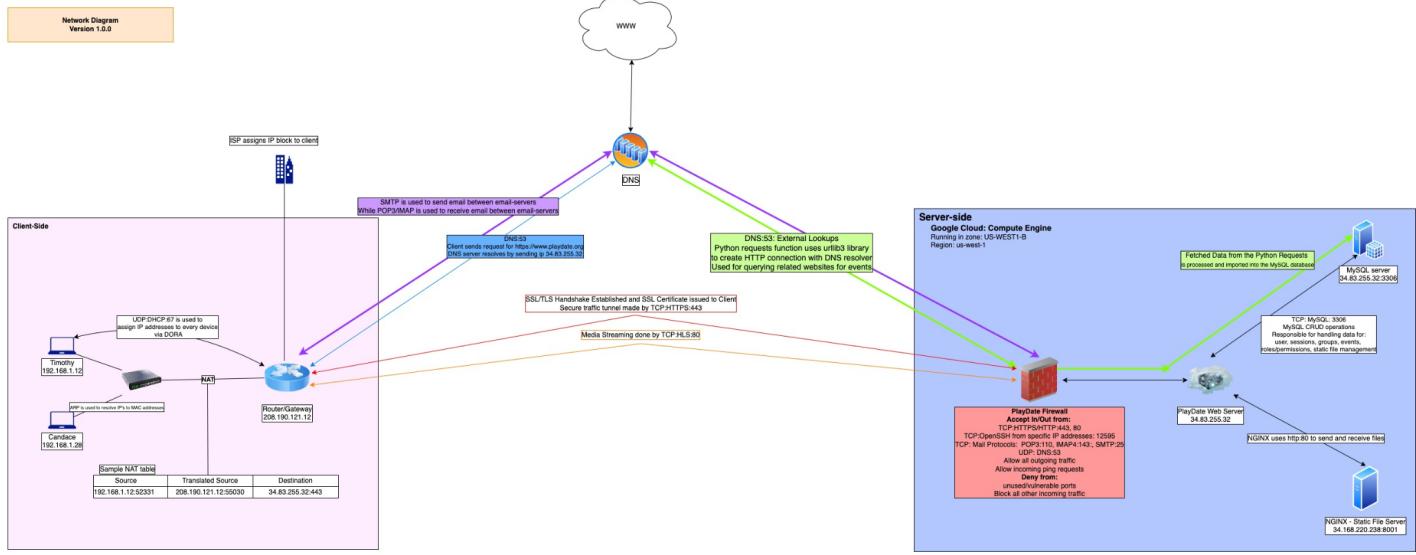
There have been no changes to our choice of software and frameworks between Milestone 1 and Milestone 2.

## 6. High Level UML Diagrams

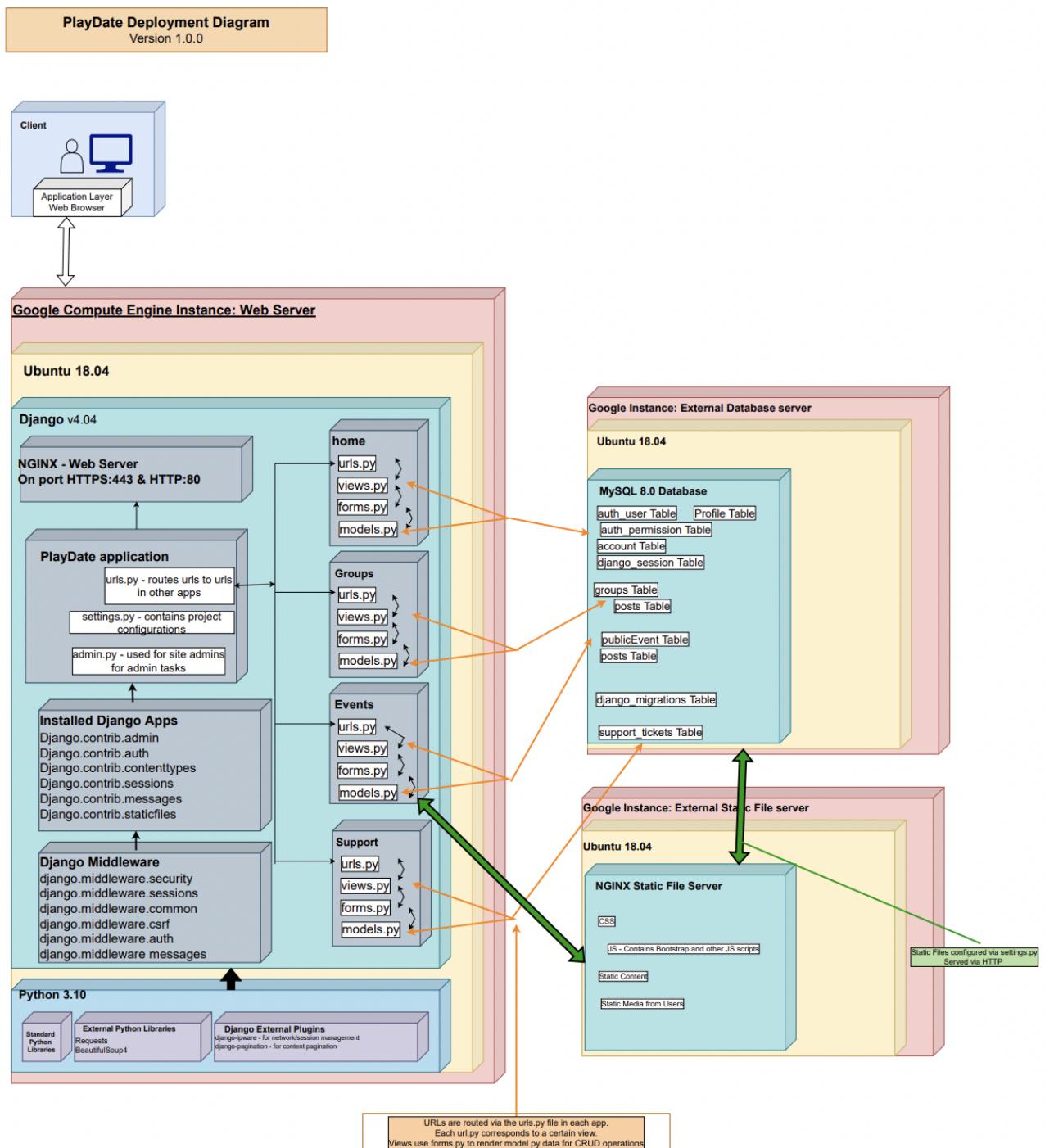


# 7. High Level Application Network and Deployment Diagrams

## 1. Network Diagram



## 2. Deployment Diagram



## 8. Identify actual key risks for your project at this time

- **Skill risks:** Some of the team members are new to Django, Git and Python. Other members who know more are helping each other in ramping up.
- **Schedule risks:** Some of the team members have a tight schedule. But after getting feedback for Milestone 1 our group realized that we could have done better perhaps if we met more often just like the group that did well. We at first did not plan to meet on weekends but have changed that since. We often do not all have the same free time so we have been meeting for about an hour and a half after class on Tuesdays and Thursdays but going forward we have decided it will be okay to meet more often even if not all can show up. Going forward we plan to record our meetings so that the people that can't make it won't miss out completely. For those that do miss a meeting can also add input via our Discord if they wish to speak on what was discussed while they were gone.
- **Technical risk:** We all, or most of us seemed to have some amount of technical difficulty while trying to get set up with our project. Making sure everything was installed correctly was an issue that the backend team, and especially Andy, all helped with. When any technical issue did come up, and they often did, we shared screenshots in our Discord to help each other out. The number of technical issues did take away a good amount of time that we could have used building our project, but I think we are all aware that technical issues come with the territory and learning how to fix these types of issues are part of the learning process for us. We will continue to help each other via Discord moving forward and factor technical risks into our work time.
- **Teamwork risk:** All team members are ready to contribute, but the skill gap makes it difficult for equal contribution. The most skilled team members could be more overworked compared to the less skilled team members. We can solve this by having more people working on harder tasks to lessen the workload.
- **Legal/content risks:** The content that we currently have are from open source. So we shouldn't face any Legal risks.

## 9. Project Management

The project is divided into five milestones which act as major checkpoints in completion of the project. At each milestone we collect feedback from the CTO and revise the milestone, along with maintaining consistency of the next milestone with the previous milestone. To ensure efficient usage of time and implementation, we divide all the tasks of a milestone further into two minor checkpoints such that the first checkpoint is achieved in mid way to milestone and next checkpoint before milestone submission. During these minor checkpoints, we will organize everyday scrum meetings of 30 minutes to track the team's progress. In each of these sprints, team members can voice if they need assistance with the task and more hands will be added to solve the hurdles. Everyone needs to share their to-dos before wrapping the meeting.

We use Trello to make Kanban style lists, which will help the entire team in understanding their duties for that checkpoint and also keep track of progress. There are two types of list, one for Milestone document and other for application implementation. For each checkpoint, tasks are listed under Milestone document and Prototype. These tasks are assigned to one or more team members based on their field of expertise along with a deadline to accomplish it. The team will work parallelly to complete them. Post completion of tasks, they will move the task to the completed list.

After checkpoint 2, we revise the implementation and documentation. Team lead and rest of the members give feedback to each other on tasks that are completed. Post testing the product if there is no further feedback, in agreement with everyone, we submit the milestones.

## 10. Detailed List of contribution

Name	Role	Contribution
Soujanya Ravindra Nayak	Team Lead Document Contributor Git Master	Assigned Tasks of Milestone 2 and updated Lists on Trello Implemented storyboards and ERD Implemented Search and Filter functionality of Home Page. Proofreading and feedback on M2 document to team members and revision of M2 document. Feedback on UI of HomePage Wrote data definitions and modified Data definitions
Margaret De La Torre	Front-end lead Document Contributor	Designed storyboards Contributed to revising functional requirements Contributed to front end of Home Page Analyzed risks and updated the document Wrote data definitions
Andy Cho	Back-end lead Document Contributor	Implemented the backend of signup process in Home Page Managed GANTT chart Contributed in prioritizing and revising functional requirements. Designed ERD and UML Designed and created Network diagram Designed and created Deployment diagram Server maintenance and deployment Assisted other team mates in debugging development environments. Wrote data definitions
Martin Salvatierra	Front-end Team Document Contributor	Designed story boards Contributed in prioritizing and revising functional requirements. Designed UML diagram Wrote data definitions
Qin Geng	Front-end Team Document Contributor	Designed Story boards 2.1 to 2.3 Contributed to revision of functional requirements. Contributed to proof reading of M2 document and gave feedback. Designed ERD and Database Models Implemented header, general_navbar, and registered-navbar template

		<p>Implemented Login page and Home page, along with logout feature.</p> <p>Contributed to front end of Home Page</p> <p>Wrote data definitions and modified Data definitions</p>
Will Plachno	Back-end Team Document Contributor	<p>Contributed in prioritizing and revising functional requirements.</p> <p>Designed ERD and UML diagram</p> <p>Implemented backend of registration(sign up)</p> <p>Finished High-Level APIs and Algorithms section</p> <p>Assisted other team mates in debugging development environments.</p>
Victor Callejas	Backend Team Document Contributor	<p>Designed story boards and UML diagram</p> <p>Contributed to revision of functional requirements.</p> <p>Contributed to proof reading of M2 document and gave feedback.</p>