

# SW Engineering CSC648-848

Summer 2022

Milestone 2 V2

<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	Git Master	<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
07/19/2022	M3V1
07/19/2022	M2V2
07/19/2022	M1V2
07/07/2022	M2V1
06/21/2022	M1V1

## Table of Contents

<b>1. DATA DEFINITIONS</b>	<b>3</b>
<b>2. PRIORITIZED FUNCTIONAL REQUIREMENTS</b>	<b>7</b>
Priority 1	7
Priority 2	11
Priority 3	13
<b>3. UI MOCKUPS &amp; STORYBOARD</b>	<b>17</b>
<b>4. HIGH LEVEL DATABASE ARCHITECTURE &amp; ORGANIZATION</b>	<b>24</b>
<b>5. High Level APIs and Main Algorithms</b>	<b>30</b>
<b>6. HIGH LEVEL UML DIAGRAMS</b>	<b>31</b>
<b>7. HIGH LEVEL APPLICATION NETWORK AND DEPLOYMENT DIAGRAMS</b>	<b>32</b>
<b>8. IDENTIFY ACTUAL KEY RISKS FOR YOUR PROJECT AT THIS TIME</b>	<b>34</b>
<b>9. Project Management</b>	<b>35</b>
<b>10. 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. Look up the homepage and public events
  - 1.2. 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 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 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. **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 Full Name**
    - 9.1.2. **Description of Assistance**
    - 9.1.3. **Email:** User’s email id so that the support staff can contact the user.
  - 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. **Email:** User’s email id so that the support staff can contact the user.
9. **Backend Admin:** Work on technical issues users have reported via support staff.
  - 9.1. All backend admins have the following information:
    - 9.1.1. **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 **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
13. **Comment:** A piece of user-generated content attached to an event. A comment has all the below attributes
  - 13.1. **Created by:** who attached this comment.
  - 13.2. **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 the 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
16. **Emergency Contact:** Contact details of two persons, who are known to the registered user.
  - 16.1. Contact name
  - 16.2. Contact number
17. **User Events:** A type of event created by a registered user, but not associated with a group.
  - 17.1. The user who created the event is the creator of that event and may alter the event as they see fit.
18. **Posts:** A selection of text that can be associated with a group or event for all associated users to see.
  - 18.1. Posts must include text, but may also include images
  - 18.2. Posts can be edited and deleted by the user who created them.

## 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.
- 1.3 A general user shall be able to register.
- 1.4 A general user shall be able to create one account.
- 1.5 A general user shall be able to become one registered user.
- 1.7 A general user shall be able to request assistance from the PlayDate support staff for onboarding.

#### 2. Registered User

- 2.1 A registered user shall be able to log into their account.
- 2.2 A registered user shall have a profile.
- 2.3 A registered user shall be able to have a username.
- 2.4 A registered user shall be able to have an email address
- 2.6 A registered user shall be able to edit their Username in profile
- 2.7 A registered user shall be able to edit their Email in profile
- 2.9 A registered user shall be able to edit their Name in profile
- 2.11 A registered user shall be able to edit their Dependents' Name in profile
- 2.12 A registered user shall be able to edit their Dependents' Birth Date in profile
- 2.13 A registered user shall be able to edit their Dependents' Type in profile
- 2.14 A registered user shall be able to edit their Dependents' Interests in profile
- 2.17 A registered user shall be able to search for public events.
- 2.18 A registered user shall be able to view public events
- 2.19 A registered user shall be able to post on public events
- 2.21 A registered user shall be able to log out from the application.
- 2.22 A registered user shall have one or more dependents
- 2.23 A registered user shall be able to become a group user.
- 2.24 A registered user shall be able to create many groups.
- 2.25 A registered user shall be able to become a group admin
- 2.26 A registered user shall be able to search for groups based on search criteria of location.
- 2.27 A registered user shall be able to search for groups based on search criteria of interest.
- 2.28 A registered user shall be able to search for groups based on search criteria of group name.

- 2.29 A registered user shall be able to join many groups
- 2.30 A registered user shall be able to search for events from all groups they are a part of.
- 2.31 A registered user shall be able to join a group.
- 2.33 A registered user shall be able to browse groups
- 2.46 A registered user shall be able to request for technical assistance from support staff on product bugs
- 2.59 A registered user shall be able to edit their Birth Date in profile
- 2.60 A registered user shall be able to view public user events
- 2.61 A registered user shall be able to make their own user events public
- 2.62 A registered user shall be able to sign up for other users' events.
- 2.64 A registered user shall be able to cancel their sign up to an event

### 3. Group

- 3.1 A group shall have at least one group user.
- 3.2 A group shall have at least one group admin.
- 3.4 A group shall have 0 or more events
- 3.5 A group shall include the creator of the group

### 4. Group Users

- 4.1 A group user shall also be a registered user.
- 4.2 A group user shall be able to post on a group
- 4.3 A group user shall be able to edit their own post.
- 4.7 A group user shall be able to create group events.
- 4.8 A group user who creates a group event is the event's event admin
- 4.11 A group user shall be able to search for group events.
- 4.12 A group user shall be able to sign up for group events.
- 4.13 A group user shall be able to leave a group.
- 4.23 A group user shall be able to request for technical assistance from support staff on product bugs.

### 5. Group Admin

- 5.1 A group admin shall also be a registered user.
- 5.2 A group admin shall be able to administer at least one group
- 5.3 A group admin shall be able to add or remove many group members.
- 5.6 A group admin shall be able to delete group events
- 5.8 A group admin shall be able to invite a registered user to the group.

### 6. Account

- 6.1 An account shall be provided for each registered user
- 6.2 An account shall carry the profile of a user
- 6.3 An account shall associate with 1 to many roles.

7. Roles
  - 7.1 A role shall be used by 0 or more accounts
  - 7.2 A role shall allow the associated user to interact with the application
  - 7.3 A role of ‘registered user’ shall allow the account all the functionality of a registered user.
  - 7.4 A role of ‘group admin’ shall allot the account all the functionality of a group admin, but only for the group the account administrates
  - 7.5 A role of ‘support staff’ shall allow the account to be emailed for support concerns and to have all of the abilities of the support staff
  - 7.6 A role of ‘backend admin’ shall allow the account to have all the abilities of a backend admin
  - 7.7 A role of ‘group user’ shall allow the account to have all the abilities of a group user
8. Support Staff
  - 8.1 A support staff shall receive emails regarding user onboarding issues.
  - 8.2 A support staff shall receive help requests from general users
  - 8.3 A support staff shall receive emails regarding user technical issues.
  - 8.4 A support staff shall receive emails from registered users
9. Backend Admin
  - 9.9 A backend admin shall be able to access all the public content of events
10. Events
  - 10.1 An event shall have a name
  - 10.2 An event shall have a date and time
  - 10.3 An event shall have an address
  - 10.4 An event shall have a list of people currently RSVP’d
  - 10.5 An event shall be edited by the event admin
  - 10.6 An event shall set the event creator as the event admin
  - 10.7 An event shall be either a public event, a group event, or a user event
11. Public Events
  - 11.1 Public events can be searched for by general users
  - 11.2 Public events can be searched for by registered users
  - 11.3 Public events can be viewed by general users
  - 11.4 Public events can be viewed by registered users
  - 11.5 Public events can be posted on by registered users
12. Group Events
  - 12.1 A group event shall immediately accept sign-ups from group users of that group
  - 12.2 A group event can be edited by the event admin

- 12.3 A group event can be edited by the group admin
- 12.4 A group event can be created by any group user
- 12.5 A group event can be signed up for by group users of that group
- 12.6 A group event can be canceled by the group admin
- 12.7 A group event shall include a sign up by the event creator

### 13. Comment

- 13.1 A comment shall be created by a registered user
- 13.4 A comment shall include text
- 13.5 A comment shall include a datetime of when it was created
- 13.6 A comment shall be removed from the system if the post it was attached to is deleted

### 14. Dependents

- 14.1 A dependent shall have a name
- 14.2 A dependent shall have a birth date
- 14.3 A dependent shall have a type
- 14.4 A dependent shall have a list of interests
- 14.6 A dependent shall be managed by their associated registered user

### 15. Survey

*No Priority 1 Functional Requirements for 15. Survey*

### 16. Emergency Contacts

- 16.1 An emergency request shall be requested by any registered user.
- 16.3 An emergency request shall contain the registered user's event location and contact number

### 17. User Events

- 17.1 A user event shall have an event admin
- 17.7 A user event shall be able to be made public by the user who created it

### 18. Posts

- 18.1 A post shall be made on an event or group
- 18.2 A post shall contain text
- 18.5 A post on a group shall be created by a group user of the group

## **Priority 2**

### **1. General User**

1.6 A general user shall be able to upload proof of the parent of a kid or pet.

### **2. Registered User**

2.5 A registered user shall have an address in their profile

2.8 A registered user shall be able to edit their Address in profile

2.10 A registered user shall be able to edit their Birth Date in profile

2.15 A registered user shall be able to edit their Dependents' schedule

2.20 A registered user shall be able to comment on public event posts

2.32 A registered user shall be able to search for users nearby an address

2.34 A registered user shall be able to create user events.

2.35 A registered user shall be able to send notification to a selected number of registered users via application.

2.36 A registered user shall be able to search for user events based on location.

2.39 A registered user shall be able to post on a user event

2.40 A registered user shall be able to comment on a user event post

2.41 A registered user shall be able to request emergency assistance via PlayDate application.

2.50 A registered user shall be able to report other users by contacting support staff

2.54 A registered user shall be able to view their list of RSVP'd events

2.57 A registered user shall be able to add events to their favorites

2.63 A registered user shall be able to comment on other users' posts.

2.65 A registered user shall be able to search for friends by username

2.66 A registered user shall be able to search for friends by name

2.67 A registered user shall be able to accept an invitation to join a group

2.68 A registered user shall be able to upload a profile photo.

2.69 A registered user shall be able to edit their own post

2.70 A registered user shall be able to filter event searches by dependent type

2.71 A registered user signed up for an event shall be able to view the event attendees

2.72 A registered user shall be able to search for users by location

2.73 A registered user shall be able to delete their comments

2.74 A registered user shall be able to accept event invitations

2.75 A registered user shall be able to add two emergency contacts.

2.76 A registered user shall be able to edit emergency contacts.

2.77 A registered user shall be able to remove themselves from the application

### **3. Group**

3.3 A group shall have 0 or more comments

- 3.6 A group shall be able to be joined by request
- 3.7 A group shall be able to be joined by invite
- 3.8 A group shall contain no more than 50 group users

#### 4. Group Users

- 4.4 A group user shall be able to delete their posts
- 4.5 A group user shall be able to edit a post on a group
- 4.6 A group user shall be able to delete a post on a group
- 4.9 A group user shall receive a notification when a group event is created.
- 4.16 A group user shall receive a notification when a group receives a Post
- 4.17 A group user shall be able to send group invites to registered users.
- 4.26 A group user shall be able to include images in their posts
- 4.27 A group user shall be able to contact the support staff to report other group users
- 4.29 A group user shall be able to comment on group posts
- 4.30 A group user shall be able to view past group events

#### 5. Group Admin

- 5.4 A group admin shall be able to accept group join requests
- 5.5 A group admin shall be able to deny group join requests
- 5.9 A group admin shall be able to delete group posts

#### 6. Account

*No Priority 2 Functional Requirements for 6. Account*

#### 7. Roles

*No Priority 2 Functional Requirements for 7. Roles*

#### 8. Support Staff

*No Priority 2 Functional Requirements for 8. Support Staff*

#### 9. Backend Admin

- 9.4 A back end admin shall be able to remove users from a group

#### 10. Events

*No Priority 2 Functional Requirements for 10. Events*

#### 11. Public Events

*No Priority 2 Functional Requirements for 11. Public Events*

## 12. Group Events

12.8 A group event with no sign-ups shall be canceled

12.9 A group event shall be removed if there is no user signup apart from the creator of the event by the datetime of the event

## 13. Comment

13.2 A comment shall be edited by the user who created it

13.3 A comment shall be deleted by the user who created it

## 14. Dependents

14.5 Dependents shall have an availability schedule

## 15. Survey

*No Priority 2 Functional Requirements for 15. Survey*

## 16. Emergency Contacts

16.2 An emergency request shall be sent to the nearest police via 911 emergency helpline

## 17. User Events

17.2 A user event shall require that all sign ups be accepted or denied by the event admin.

17.3 A user event shall be searchable by all registered users

## 18. Posts

18.3 A post shall be able to contain images

18.4 A post shall be created by a registered user

18.6 A post on an event shall be created by any registered user with access to the event

18.7 A post shall be editable by the creator

## Priority 3

### 1. General User

*No Priority 3 Functional Requirements for 1. General User*

### 2. Registered User

2.16 A registered user shall be able to send a referral link to a friend

2.37 A registered user shall be able to send out user event invites to a filtered user list.

2.38 A registered user shall be able to schedule a recurring event.

2.42 A registered user shall be able to create a survey for an event.

2.43 A registered user who is attending the event shall be able to respond to surveys

corresponding to that event.

2.44 A registered user who created the survey shall be able to delete the survey.

2.45 A registered user who created the survey shall be able to modify it

2.47 A registered user shall be able to create 1 or more lists of friends.

2.48 A registered user shall be able to add friends to a list

2.49 A registered user shall be able to label the list of friends.

2.51 A registered user shall be able to set available events as ‘Interested’

2.52 A registered user shall be able to filter to RSVP’d events

2.53 A registered user shall be able to set available events as ‘Maybe’

2.55 A registered user shall be able to view their list of Interested events

2.56 A registered user shall be able to view their list of Maybe events

2.58 A registered user shall be able to view all of their favorite events.

### 3. Group

*No Priority 3 Functional Requirements for 3. Group*

### 4. Group Users

4.10 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.

4.14 A group user who is part of a group shall be able to view a heatmap of the group schedule.

4.15 A group user shall be able to update their availability on group heatmap.

4.18 A group user shall be able to create a survey for an event of their group

4.19 A group user shall be able to post surveys(polls) for group events

4.20 A group user who is attending the event shall be able to respond to surveys

4.21 A group user who created the survey shall be able to delete the survey.

4.22 A group user who created the survey shall be able to modify it

4.24 A group user who created the event shall be able to create a post to collect reviews after its occurrence.

4.25 A group user who attended the event shall be able to give a rating out of 5 stars

4.28 A group user shall be able to reply to group posts with emojis.

### 5. Group Admin

5.7 A group admin shall be able to delete any surveys on events of their group

### 6. Account

*No Priority 3 Functional Requirements for 6. Account*

### 7. Roles

*No Priority 3 Functional Requirements for 7. Roles*

8. Support Staff

*No Priority 3 Functional Requirements for 8. Support Staff*

9. Backend Admin

9.1 A backend admin shall be able to access the user verification portal.

9.2 A backend admin shall be able to verify the general user's identity to confirm his registration.

9.3 A back end admin shall be able to delete a group

9.5 A back end admin shall be able to remove posts from a group

9.6 A back end admin shall be able to remove comments from a group

9.7 A backend admin shall be able to access all the group content of comments

9.8 A backend admin shall be able to access all the group content of events

9.10 A backend admin shall be able to remove registered user from the application.

10. Events

*No Priority 3 Functional Requirements for 10. Events*

11. Public Events

11.6 A public event shall be set as 'Interested' by a registered user

11.7 A public event shall be set as 'Maybe' by a registered user

11.8 A public event shall be favorited by zero or more registered users

12. Group Events

12.10 A group event shall contain the cumulative average of all the ratings as the rating of that event.

12.11 A group event shall be set as 'Interested' by a group user

12.12 A group event shall be set as 'Maybe' by a group user

12.13 A group event shall be favorited by zero or more registered users

13. Comment

*No Priority 3 Functional Requirements for 13. Comments*

14. Dependents

*No Priority 3 Functional Requirements for 14. Dependents*

15. Survey

15.1 A survey shall be attached to a group event

15.2 A survey shall be created by a group user

15.3 A survey shall be responded to by a group user signed up for the group event the

survey is attached to

15.4 A survey shall be deleted by the group user that created the survey

15.5 A survey shall be deleted by the group admin

15.6 A survey shall consist of a set of choices and the number of times those choices have been chosen

## 16. Emergency Contacts

*No Priority 3 Functional Requirements for 16. Emergency Request*

## 17. User Events

17.4 A user event shall be set as ‘Interested’ by a registered user

17.5 A user event shall be set as ‘Maybe’ by a registered user

17.6 A user event shall be favorited by zero or more registered users

## 18. Posts

*No Priority 3 Functional Requirements for 18. Posts*

### 3. UI Mockups & Storyboard

#### 1. Use Case 1: New user registration

**User Case 2:**

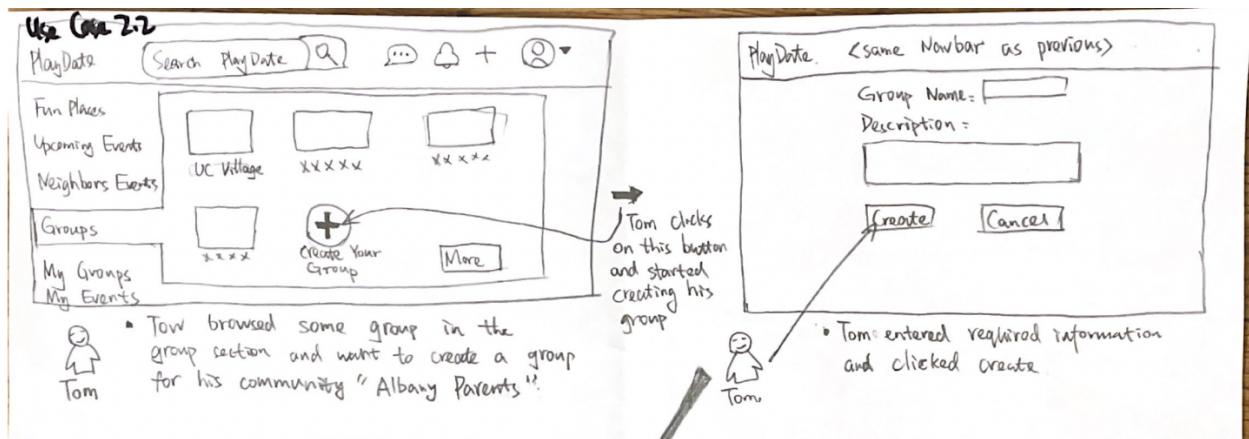
The storyboard consists of four hand-drawn wireframes connected by arrows, illustrating a user flow for new user registration.

- Home Screen:** Shows a search bar ("Enter Zip Code to Search") and buttons for "Sign Up", "Log In", and "Help". Below is a section titled "Fun Places" with "Upcoming Events" and three placeholder boxes labeled "xxxx", "xxxx", and "xxxxx". A "More Places" button is at the bottom right. An annotation notes "Many clicks on Sign Up".
- Sign Up Screen:** A "PlayDate Sign Up" form with fields for Email, Username, Password, and Re-enter password. It includes a "Sign Up" button, a link for "Already Got An Account?", and a "Log In" button.
- Log In Screen:** A "PlayDate" screen with a search bar ("Search PlayDate") and a "Log In" button. Below is a section titled "Fun Places" with "Upcoming Events" and three placeholder boxes labeled "xxxx", "xxxx", and "xxxxx". A "More" button is at the bottom right. An annotation notes "Many Entered all the required info and registered PlayDate".
- Profile Screen:** A "PlayDate" screen with a search bar ("Search For PlayDate"). It shows a post from "Helen Lee" with the text "Meet @ Ocean Park" and "Posted: Helen Lee". Below is a section titled "Her posts" with three placeholder boxes labeled "xxxx", "xxxx", and "xxxxx". A "More" button is at the bottom right. An annotation notes "After clicking on Helen's name, Mary is directed to Helen's personal page".

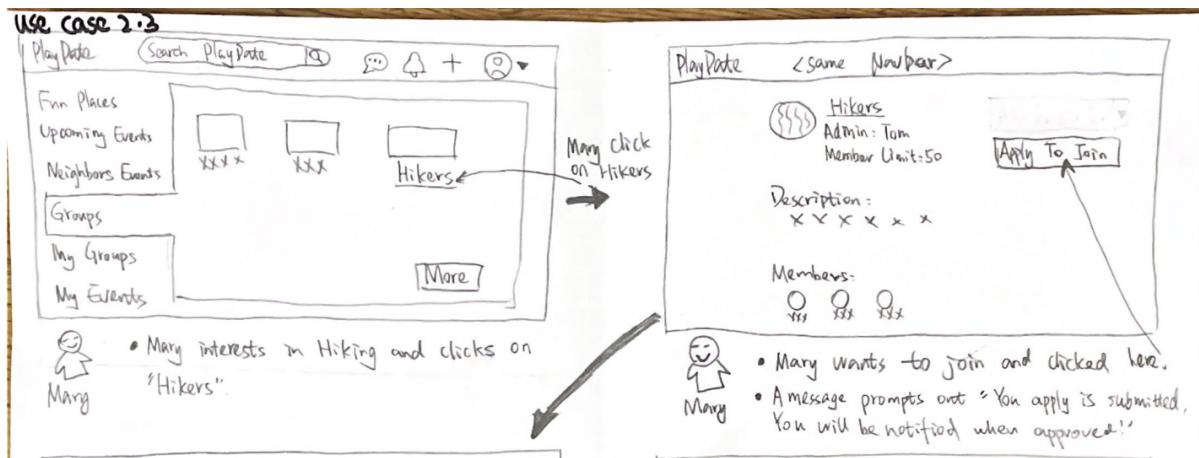
**Mary's Actions and Notes:**

- Home Screen: Mary can browse all listed fun places and upcoming events on PlayDate. She can also search for public info by zipcode.
- Sign Up Screen: Mary wants to register so that she can view more events.
- Log In Screen: After registration, Mary automatically logged in and see a different page.
- Profile Screen: 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.

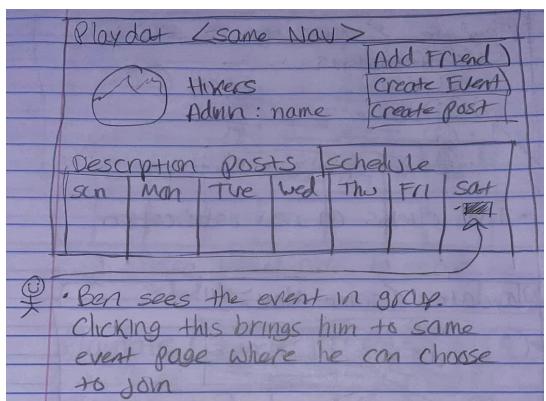
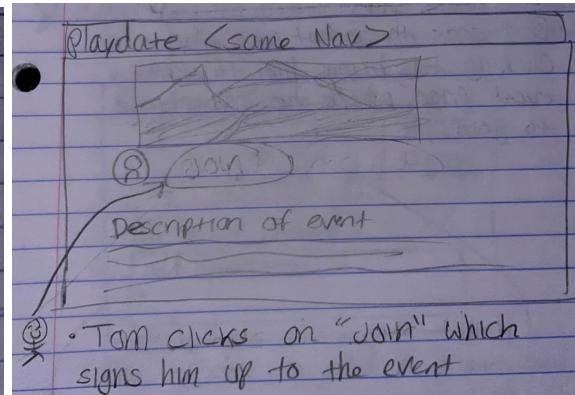
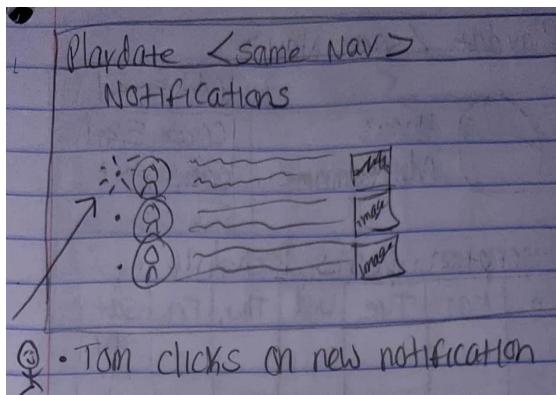
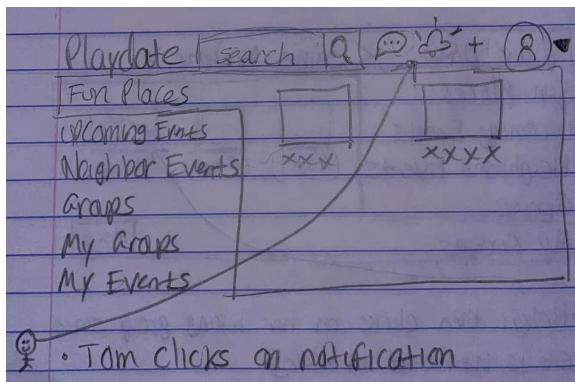
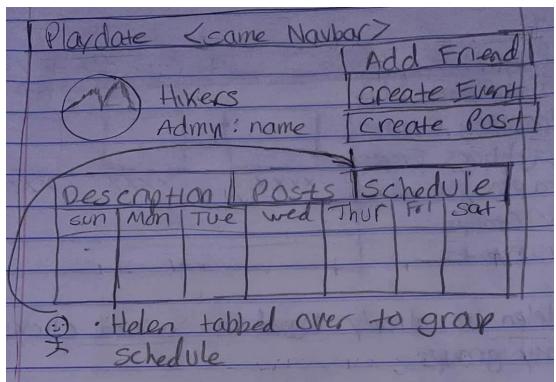
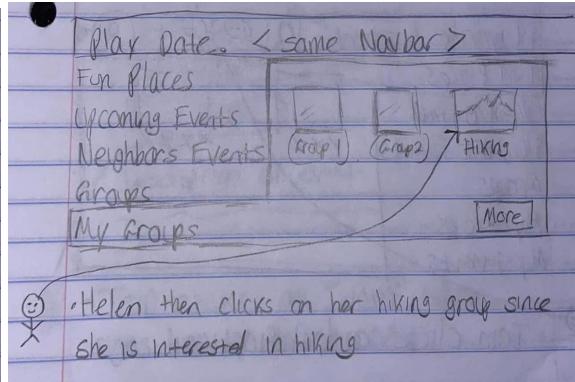
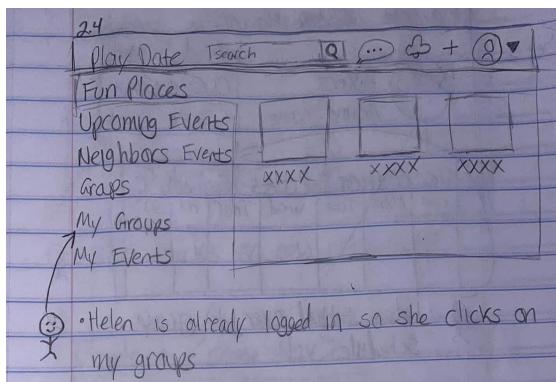
## 2. Use Case 2: Group Creation



### 3. Use Case 3: Joining group



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



## 5. Use Case 5: Creating Events

2.5 Playdate < same Nav >

Fun Places	<input type="text"/> event name	<input type="text"/> event name
Upcoming Events	Create Event	
Neighbor Events		
Groups		
My Groups		
My Events		

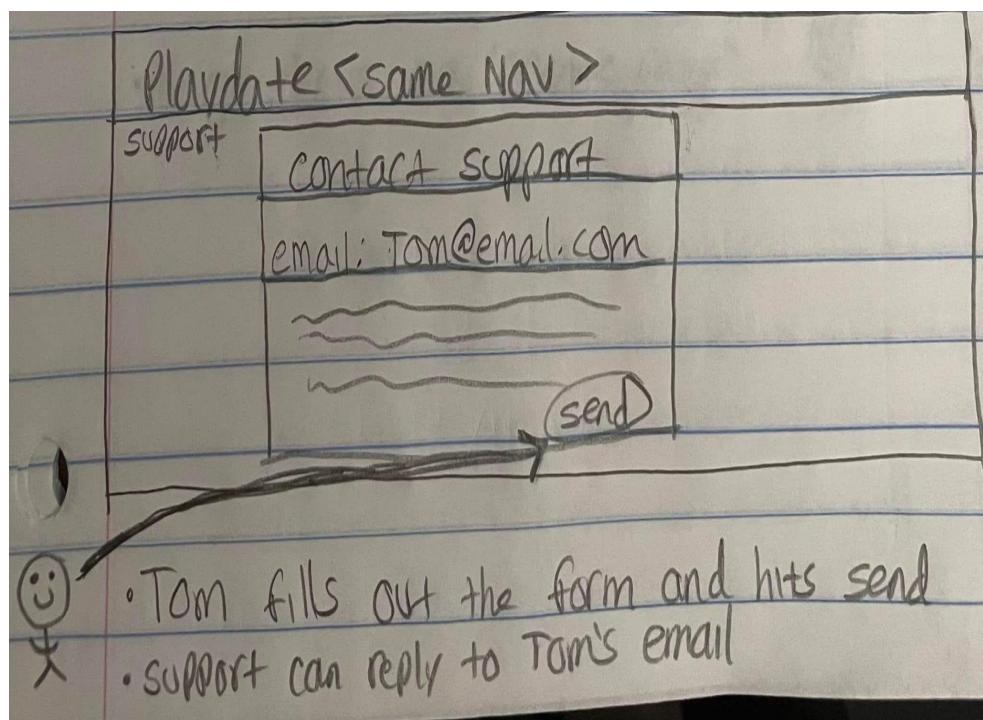
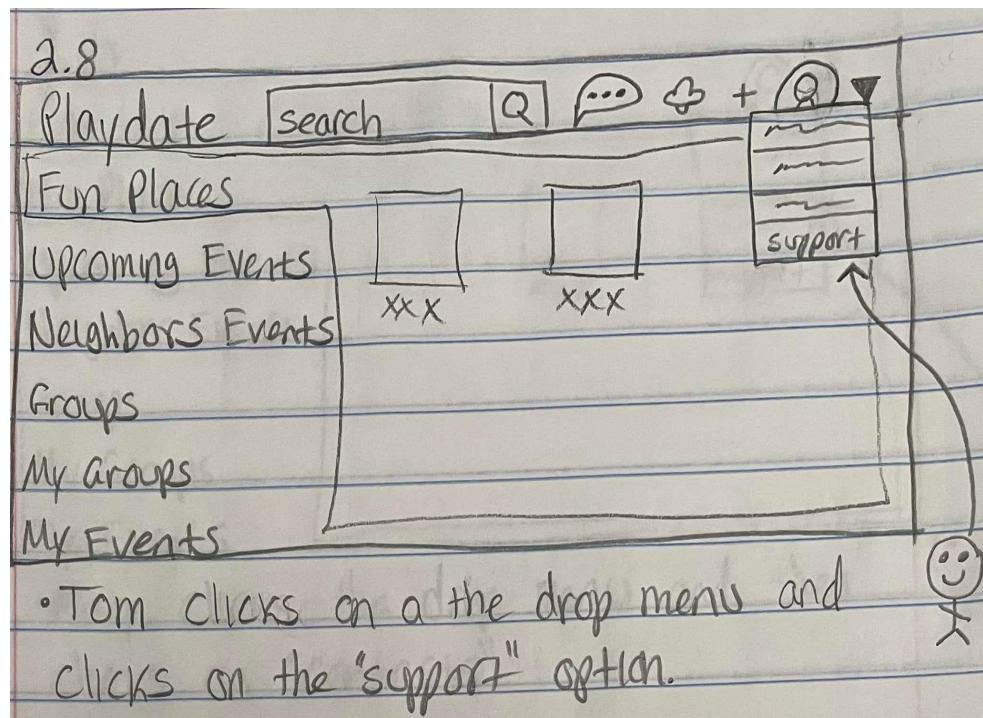
• On the "My Events" tab from main page Eric clicks on "Create Event" button.

Playdate < same Name >

<input type="checkbox"/> make public
<input type="checkbox"/> make private
+ add photo
(8) invite friends
add description...

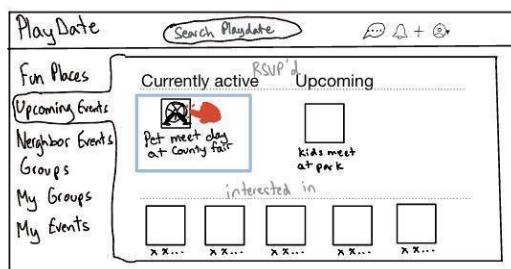
• Eric is able to add a photo and description to the event, both clickable.  
 • Eric is also able to add friends to an invite list, invites will be sent after event is created  
 • Eric checks the "make public" check box to make the post public

## 6. Use Case 8: Technical Support



## 7. 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

## 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)

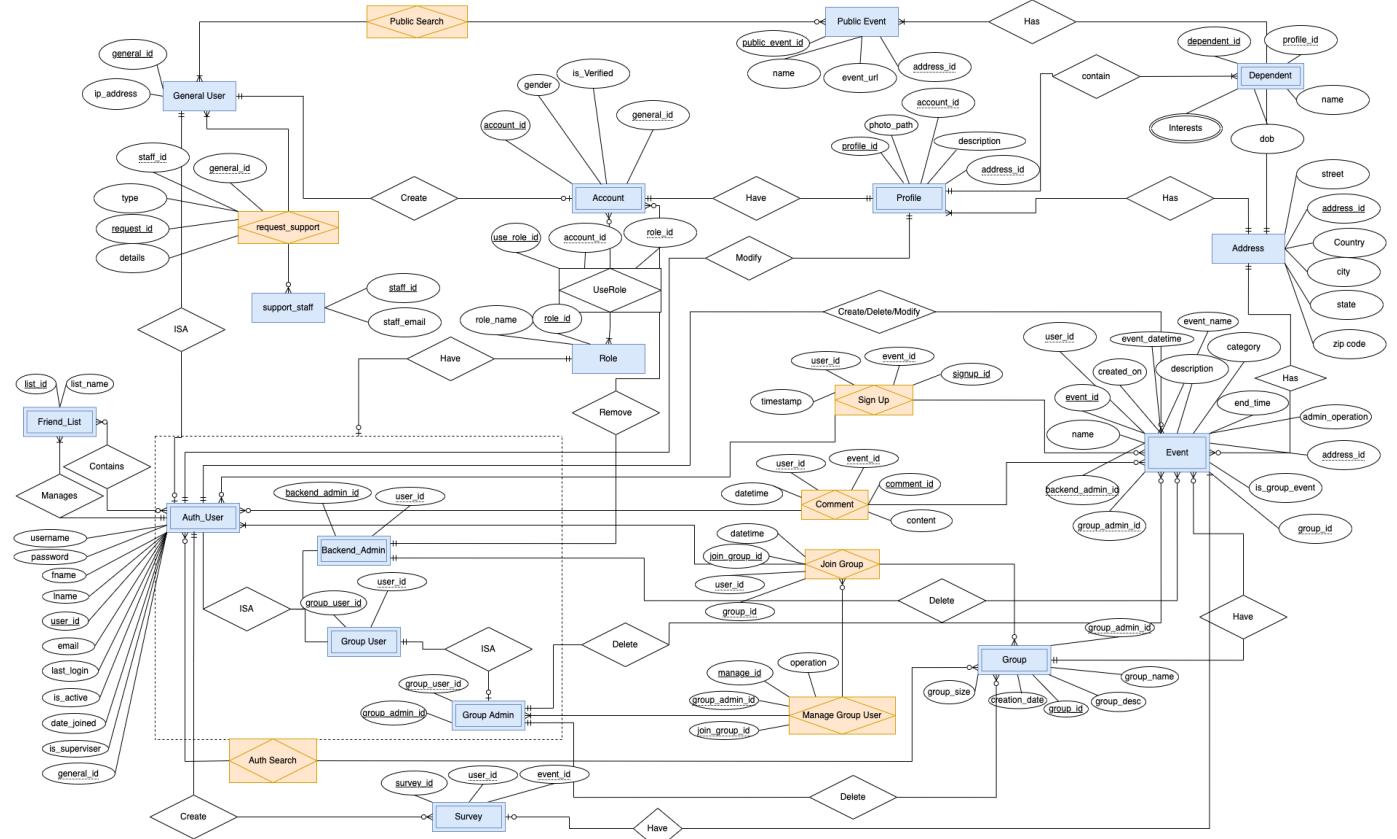
- \* group\_event\_id: strong key, numeric
- \* group\_id: weak key, numeric
- \* group\_user\_id: weak key, numeric
- \* content: alphanumeric
- \* datetime: datetime

xiii. Dependents (Weak)

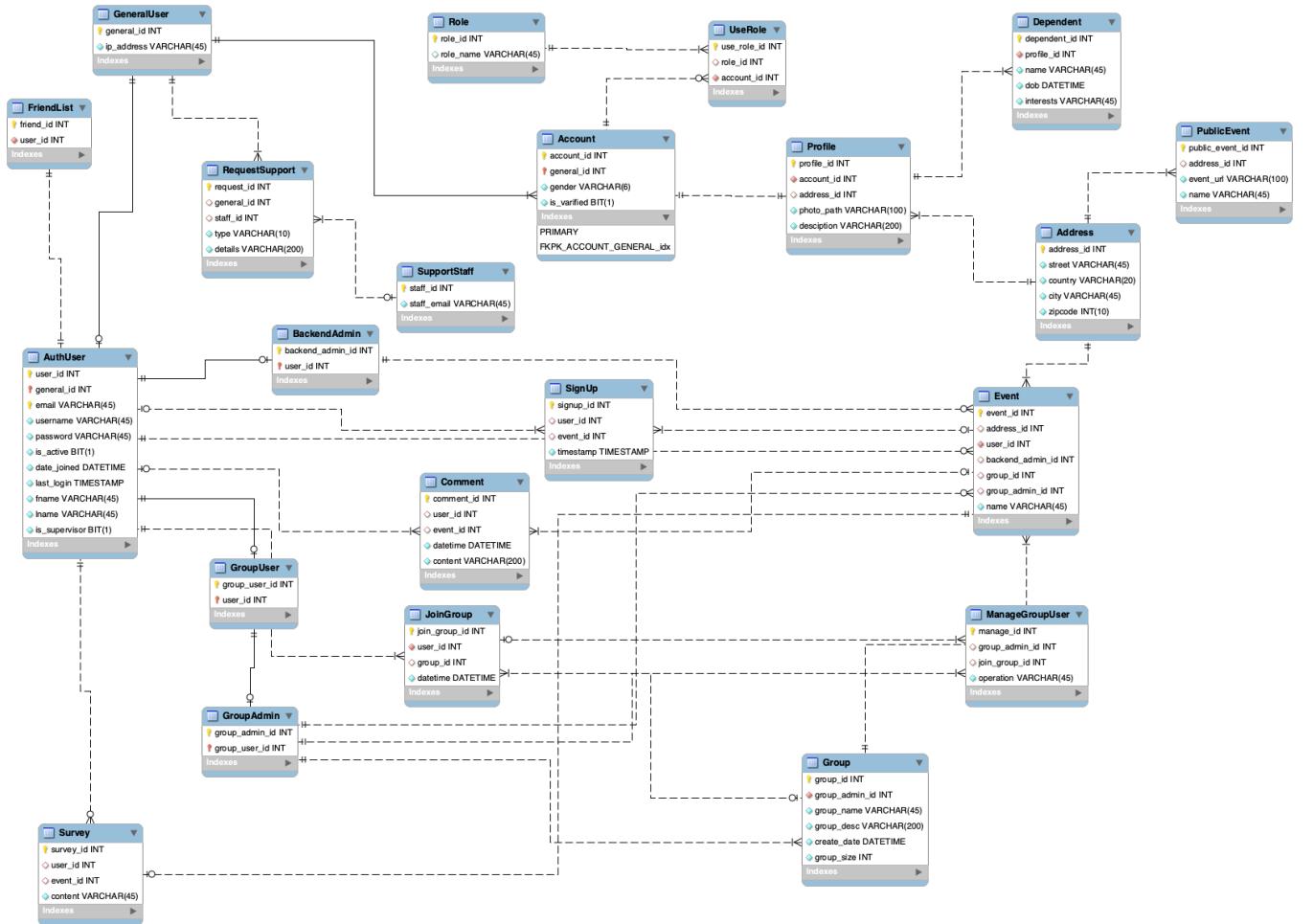
- \* dependent\_id: strong key, numeric
- \* profile\_id: weak key, numeric

- \* 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. Django stores the location/path of the file in the db so django knows where to access it.

## 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 keywords, like locations and names. After parsing and confirming validity of the keyword, we use Case-insensitive containment test ( `icontains` ) from Django QuerySet API. `icontains` is equivalent to SQL LIKE statement. `Icontain` looks for that keyword in the form of `LIKE %keyword%`. After the query set is built using `icontains`, it will run the query on the MySQL database to fetch results.

Steps:

- A. User will enter a keyword example ‘Palo’ in the search space.
- B. The keyword shall be used to build the QuerySet, where this keyword will be checked for its presence in country, state, city and street from the address table.

```
lookups= Q(address__city__icontains=keyword) |  
Q(address__zipcode__icontains=keyword) | Q(address__country__icontains=keyword) |  
Q(address__street__icontains=keyword)
```

- C. This queryset will perform a lookup on the mySQL database and return results from public events that map to the address table via foreign key of address\_id.

```
results= Publicevent.objects.filter(lookups)
```

- D. The result from the above query shall be used to render the search results of the user.

## 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.

The results from step C can be further filtered if the user selects the filter option. Here we again use the icontain to match the category of the filter.

```
results= Publicevent.objects.filter(lookups).filter(Q(category__icontains = 'kids'))
```

## 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:

- **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.

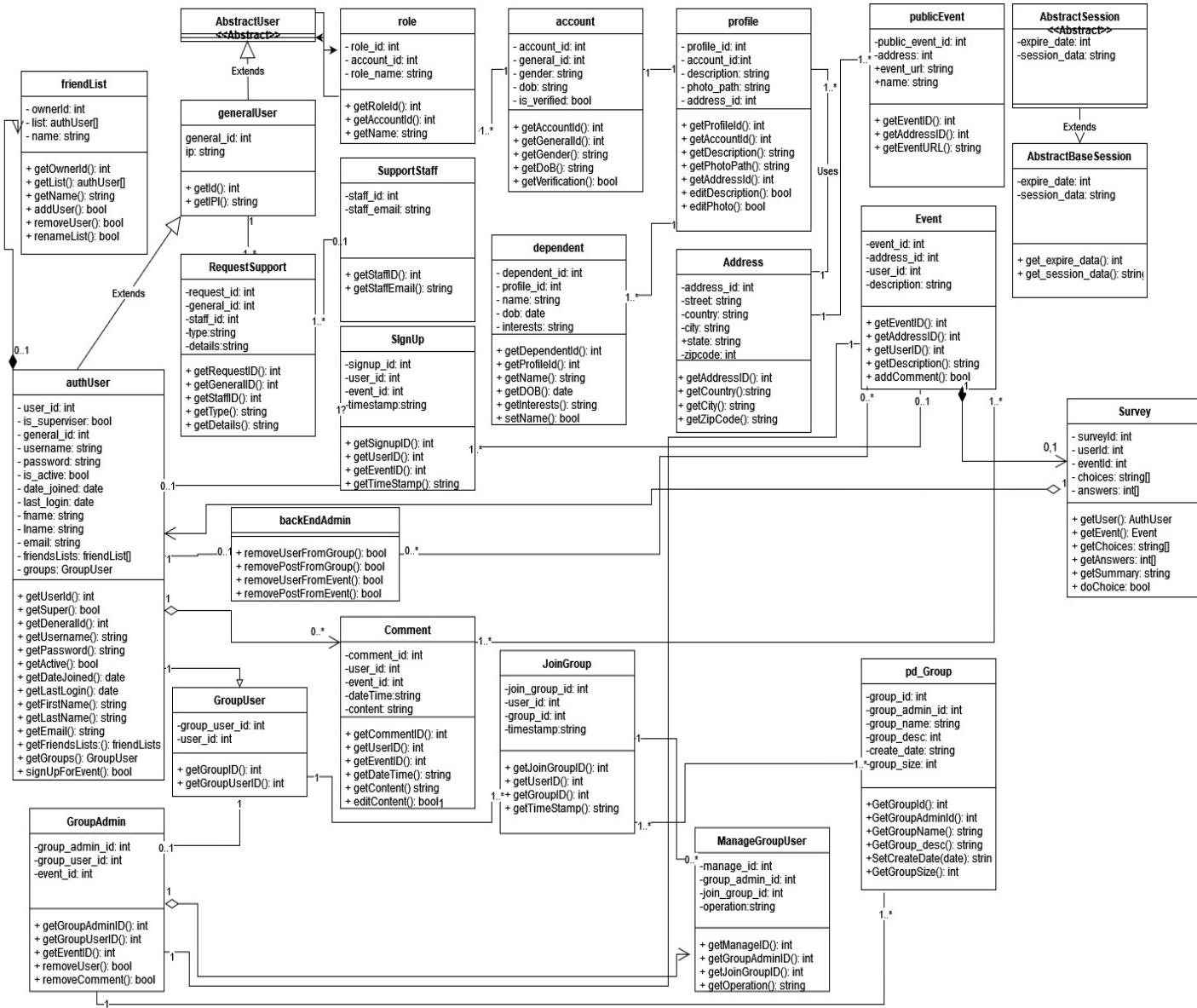
### Algorithms

We are using the Django QuerySet API, which performs lookup based on linear search algorithms. This is used to retrieve the search results from the MySQL database via object models by applying relevant filters based on the where clause of the MySQL queries. The search algorithm goes from one row to another linearly to find rows that match the filter and returns the results.

### Changes

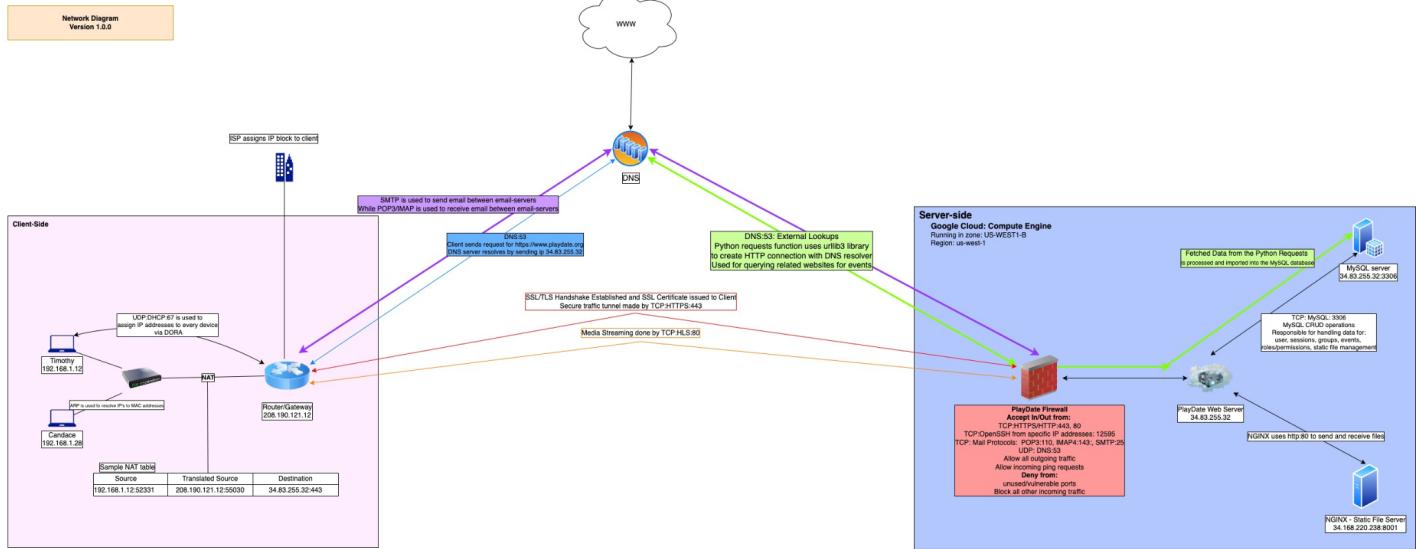
There have been no changes to our choice of software and frameworks between Milestone 1 and Milestone 2.

## 6. High Level UML Diagrams

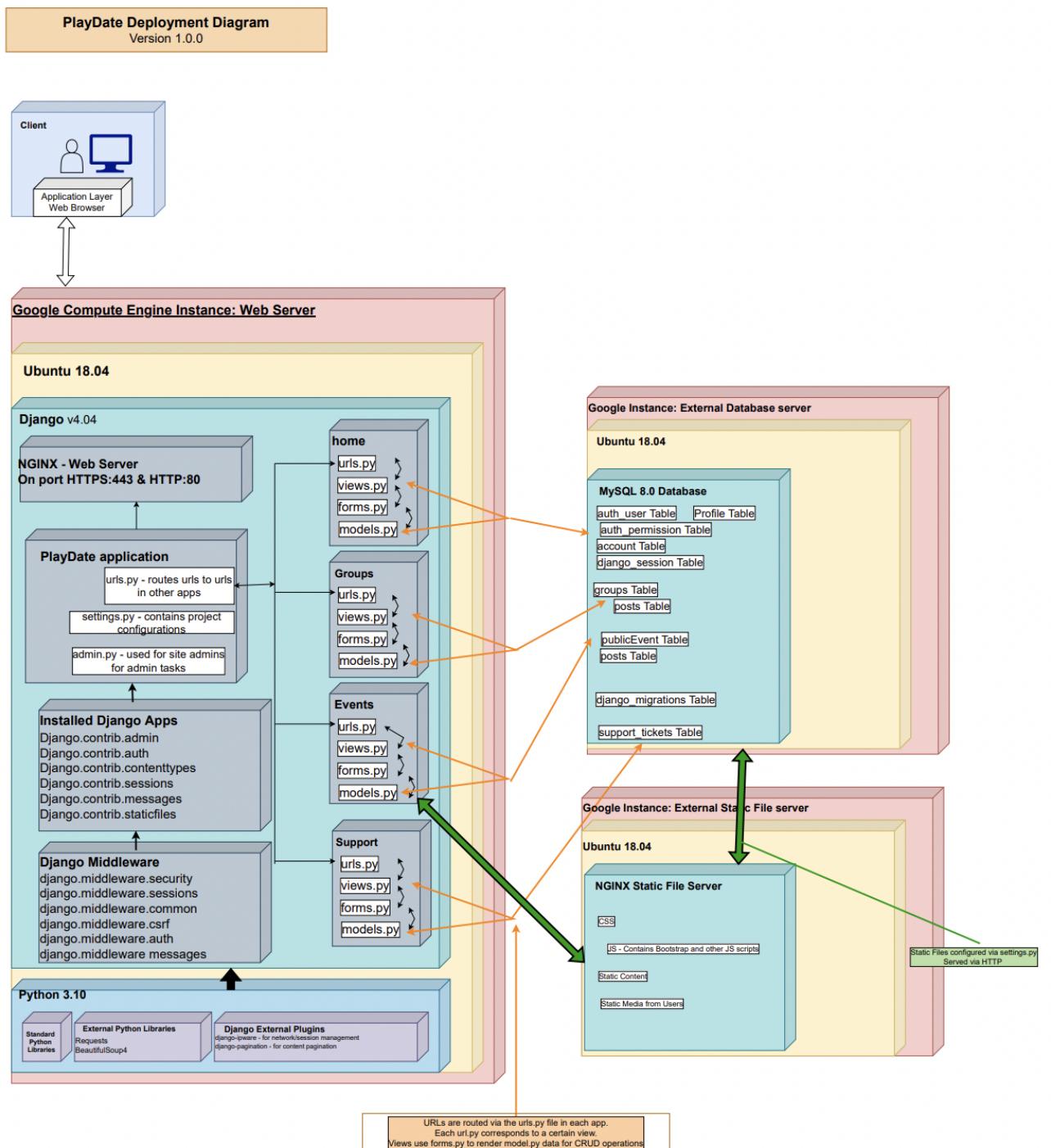


## 7. High Level Application Network and Deployment Diagrams

### 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 images that we currently have are from open source. But the legality of those are not verified by us. Hence they can form content risk to this project.

## 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 Backend Team	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 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 Implemented Login page and Home page, along

		<p>with logout feature.</p> <p>Contributed to front end of Home Page</p> <p>Wrote data definitions and modified Data definitions</p>
Will Plachno	Git Master Backend 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>