Final Project for SW Engineering Class CSC 648-848 Section 01 Summer 2021

Team 03

PlayDate

Name	Job	Email
Soujanya Ravindra Nayak	Team Lead	soujanyaravindra@gmail.com
Margaret De La Torre	Frontend Lead	mdelato1@mail.sfsu.edu
Andy Cho	Backend Lead	andrewyunejo@gmail.com
Martin Salvatierra	Frontend Team	martysaljhost@gmail.com
Qin Geng	Frontend Team	gengqin50@gmail.com
William Plachno	Git Master	wjplachno@gmail.com
Victor Callejas	Frontend Team	vcallejas@mail.sfsu.edu

Application Url: http://34.83.255.32:8000/

August 02, 2022

Table of Contents

1. Product Summary

PlayDate

Social Media has become a big part of society where adults can find like-minded friends. But what about kids and pets? They, too, need someone of their kind with whom they can spend quality time. Parents try to find playdates for their children and pets, but they are always worried about child safety. There are some applications in the market that try to solve the issue of finding playdates and activities but none ensures guaranteed safety from unauthorized users. How much do we really know about people whom we invite to be our kids' playdates?

“PlayDate” is aiming to help parents solve their pain points while looking for activities and companions for their children and pets. Parents can register in PlayDate and can either join or create events for their kids and pets. All the users of PlayDate need to undergo background verification to be able to create any event or sign up for other user’s events. This way PlayDate ensures the safety of your kids and pets.

Functions:

- General users can view public events on PlayDate that are available to all users.
- General users shall also be able to search for public events based on location.
- General users shall be able to register on PlayDate by completing the signup page.
- General users shall be able to request assistance from playdate support staff by filling the contact form and providing their name, email id and description of assistance.
- General users shall be able to use chat bot to get directions on using our application.
- Registered users shall be able to use their login credentials and login to PlayDate application.
- Registered users shall be able to view and edit their profile where they can add dependents and update their personal information.
- Registered users need to upload verification documents on profile to become verified users. Hence the registered user shall be able to upload photo identity as verification document on their profile.
- Registered users shall be able to view public events.
- Registered users shall be able to log out from their account on PlayDate.
- Registered users who are not verified shall not be able to create events or create groups. They shall also not be able to join groups or RSVP to events created by other users.
- Registered users shall be able to view events created by other PlayDate users under Events on PlayDate.
- Registered users who are verified, can register to many events that they are interested in.
- Registered users who are verified shall be able to create their own events.

- Registered users who are verified shall be able to delete the events created by them.
- Registered users can view all of their events that they created and RSVP'd on My Events
- Registered users shall be able to create many groups to form small communities of similar interests and become a group admin.
- Registered users shall be able to search the groups based on interests, location, group name.
- Registered users shall also be able to join many groups created by other users.
- Registered users shall be able to view all groups that they are part of, on My Groups.
- Registered users shall be able to cancel their signup to any event
- Each group user can create an event for their group, which will be available only to that group's users.
- Group users can RSVP to the group events.
- Group admin who created the group shall be able to delete any event from the group.
- Group users shall be able to post and comment in the group to share their thoughts with their group.
- Group users shall be able to leave the group if they are no longer interested in that group.
- Group admin shall be able to remove any user from their group.

Uniqueness:

Unlike adults, kids need an extra safe environment. Hence the unique feature of our application lies in its business logic of user verification which is incorporated to ensure safety from scammers. Post registering on our application, users need to mandatorily upload their identity document to their profile, which will be analyzed by PlayDate backend admins to permit the user to use all the functionalities of our application. If the user does not complete the verification process they will be restricted from core functions like create and join events and groups provided by PlayDate.

Application url: http://34.83.255.32:8000/

2. Milestone documents

SW Engineering CSC648-848

Summer 2022

PlayDate — by Team 03 (the "Babysitters")		
Name	**Job**	**Email**
Soujanya Ravindra Nayak	Team Lead	soujanyaravindra@gmail.com
Margaret De La Torre	Frontend Lead	mdelato1@mail.sfsu.edu
Andy Cho	Backend Lead	andrewyunejo@gmail.com
Martin Salvatierra	Frontend Team	martysaljhost@gmail.com
Qin Geng	Frontend Team	gengqin50@gmail.com
William Plachno	Git Master	wjplachno@gmail.com
Victor Callejas	Frontend Team	vcallejas@mail.sfsu.edu

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

1. Summary

Social Media is prevalent nowadays while people can be connected and communities can be expanded promptly. Meanwhile, children and pets also need friends and companions even though they don't know how to use social media softwares. "**PlayDate**" is the right application to help children and pets find suitable events in which they can take activities and find more friends. "PlayDate" is aiming to help parents solve their pain points on looking for activities and companions for their children and pets.

"PlayDate" has the following outstanding advantages why parents will choose us:

1. **General User Friendly**

   "PlayDate" application provides upcoming public events as seeders which is open for all users to browse and search. General users also get a chance to obtain valuable information from "PlayDate".
2. **Safe Environment With Identity Verification**

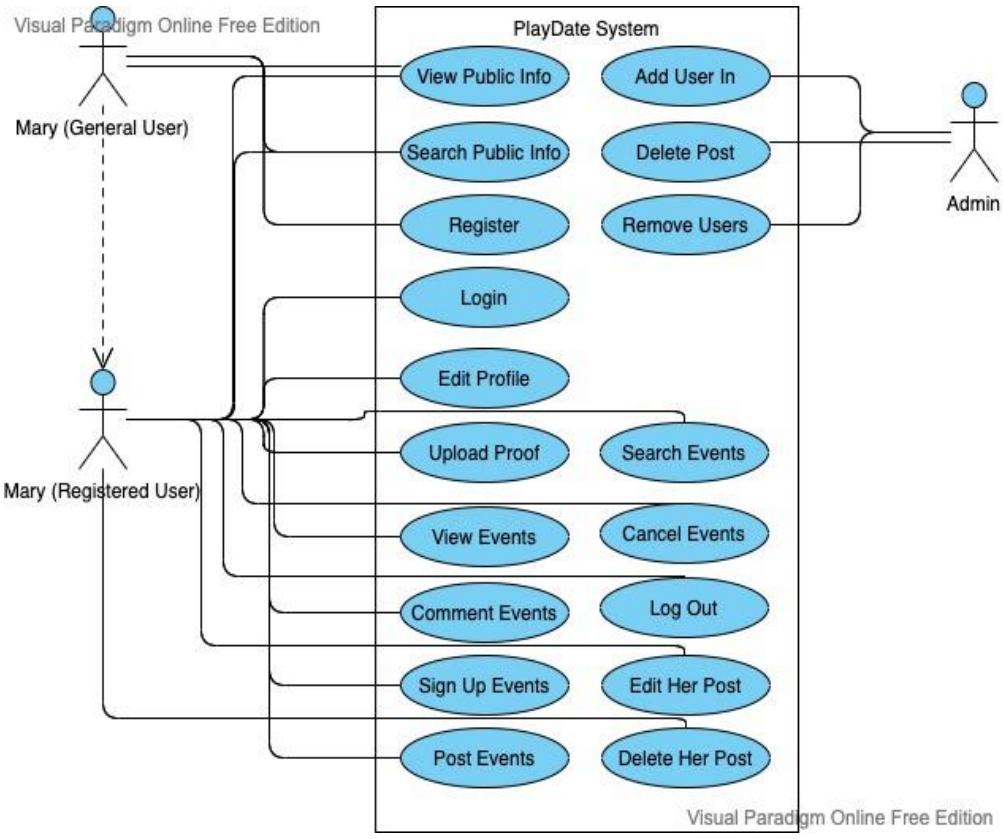
   "PlayDate" is focused on creating a safe environment for all registered users. Every registered user will have an ID Verification by the "PlayDate" team to make sure this community is totally safe to our users and their family.
3. **Connecting Neighbors**

   A registered user of "PlayDate" can search for other users based on locations which can help them find out their neighbors with similar family structure, like children and pets, and similar hang-out places. "PlayDate" can connect neighbors and provide much more opportunities for them to interact.
4. **Make Community a Big Family**

   Registered users living in the same community or having the same hang-out events are welcome to create groups. Groups will be enlarged with more and more similar users where they will share their activities and experiences. "PlayDate" will make each community a big family by providing a platform for people to share and care.
5. **Users' Data is Protected**

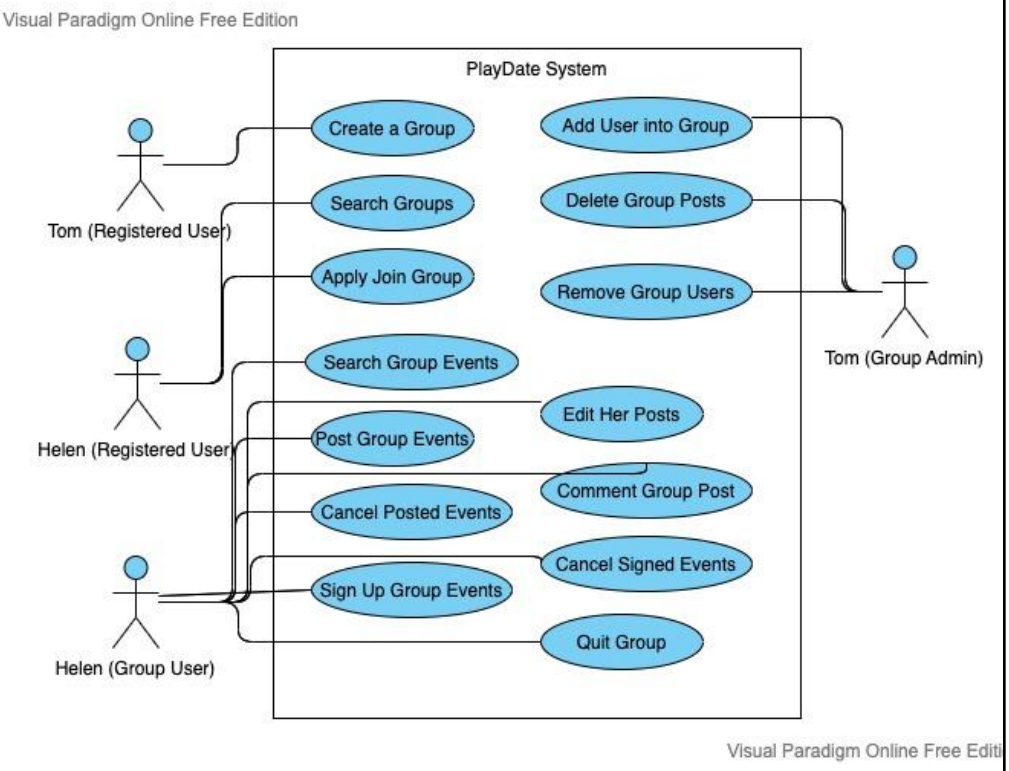
   "PlayDate" takes the responsibility to protect each registered user's data. All registered users' information will be protected and can only be seen by other registered users.

2. Main Use Cases

Title	1. New user registration
Actors	Mary (General User)
Description	Mary likes to take her young children to parks, libraries, museums, or other outdoor activities, but she always has difficulty finding outdoor playdates for her children. Mary's friend, Helen, introduces her to the "PlayDate" on which she goes through public events which seem fascinating, but find out our other parents' schedule and activities to help her find a playdate for her children. Mary thinks this app is fantastic, but she can only view public events before she registers. As Mary uses a referral link shared by Helen, she only needs to provide proof that she has kids to be registered and can have a free use of this app. After registering, Mary can login to and interact with other parents in the application.
Diagram	
Requirements	**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.

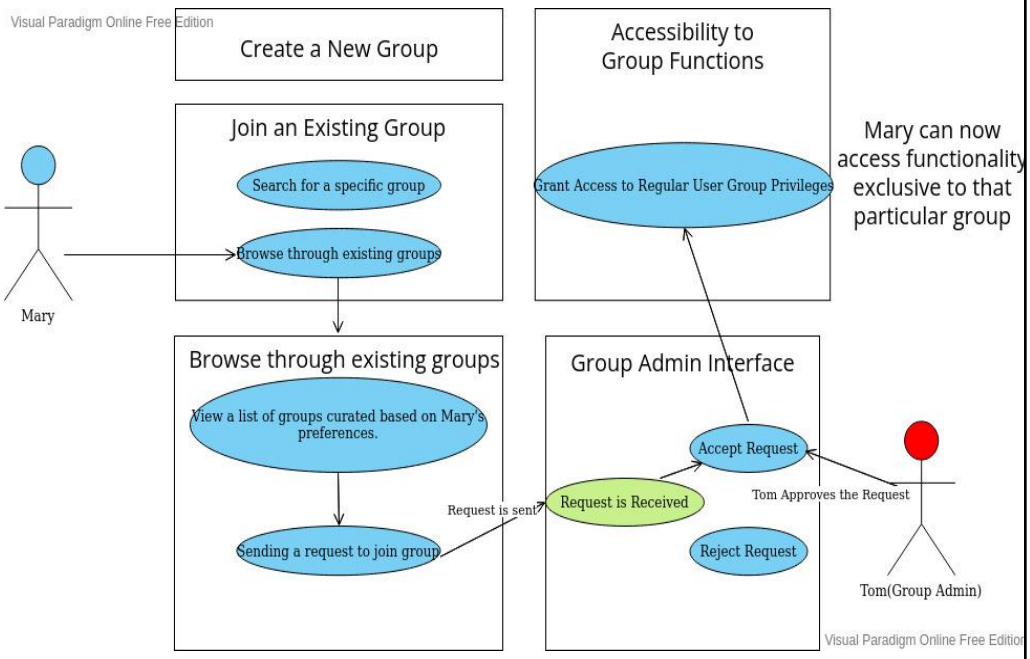
	1.4 A general user shall be able to create one account. 1.5 A general user shall be able to become only one registered user. 1.6 A general user shall be able to upload proof of the parent of a kid or pet. 1.7 A general user shall be able to request assistance from the PlayDate support staff for onboarding. **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 have a username 2.4 A registered user shall have an email address 2.5 A registered user shall have an 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 2.8 A registered user shall be able to edit their Address in profile 2.9 A registered user shall be able to edit their Name in profile 2.10 A registered user shall be able to edit their Birth Date 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 2.15 A registered user shall be able to edit their Dependents' schedule 2.16 A registered user shall be able to send a referral link to a friend. 2.17 A registered user shall be able to search for public events based on location. 2.18 A registered user shall be able to view public events 2.19 A registered user shall be able to comment on public events 2.21 A registered user shall be able to log out 2.22 A registered user shall have one or more dependents **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. **11. Public Events** 11.1 Public events can be searched for by general users

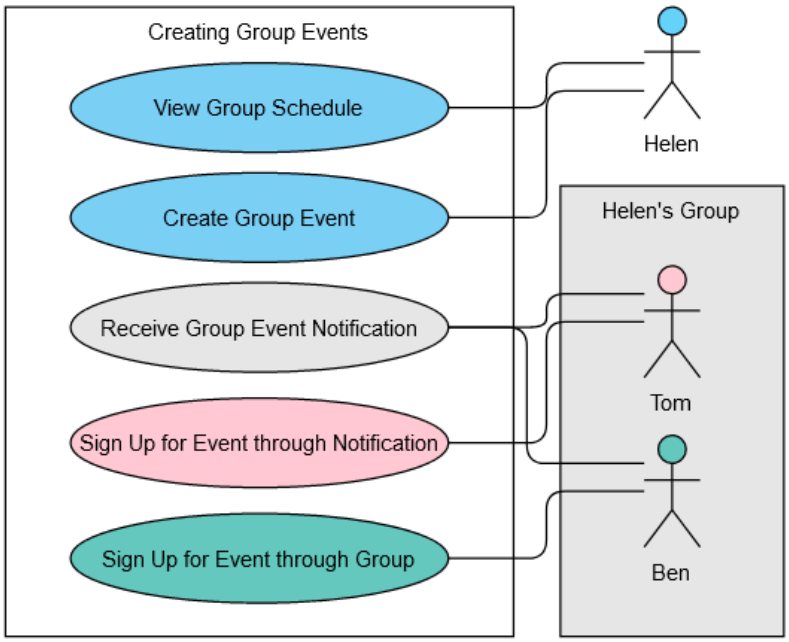
	11.2 Public events can be searched for by registered users 11.3 Public events can be viewed by general users 11.4 Public events can be viewed by registered users 11.5 Public events can be posted on by registered users **8. Support Staff** 8.1 Support staff shall receive emails regarding user onboarding issues. 8.2 Support staff shall receive help requests from general users **12. Dependents** 12.1 Dependents shall have a name 12.2 Dependents shall have a birth date 12.3 Dependents shall have a type 12.4 Dependents shall have a list of interests 12.5 Dependents shall have an availability schedule 12.6 Dependents shall be managed by their associated registered user

Title	2. Group Creation
Actors	Tom (Group admin), Helen(Registered user)
Description	Some parents Helen, Tom living in the same community want to share some group activities and want this information to be private only in the group. Tom takes the lead and creates a group. He becomes the admin for this group. He adds other parents from the same community by searching for the users based on his community address, where he finds Helen and adds her to the group. Tom gets to know that he can add a maximum of 50 parents only and he group is full. Helen is a member in Tom's group.
Diagram	
Requirements	**2. Registered User** 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 location 2.27 A registered user shall be able to search for groups based on interest 2.28 A registered user shall be able to search for groups based on group name 2.29 A registered user shall be able to join many groups.

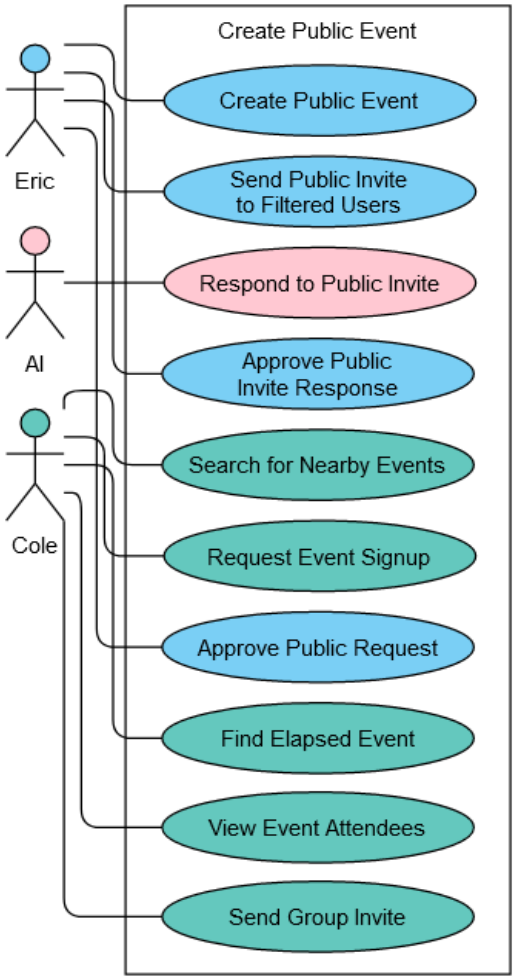
	2.30 A registered user shall be able to search for events from all groups they are a part of. 2.31 A registered user shall be able to apply to join an existing group 2.32 A registered user shall be able to search for users nearby an address **9. Back End Admin** 9.3 A back end admin shall be able to delete a group 9.4 A back end admin shall be able to remove users from 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 **4. Group User** 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 posts 4.4 A group user shall be able to delete their posts 4.5 A group user shall be able to edit a comment on a group 4.6 A group user shall be able to delete a comment on a group 4.7 A group user shall be able to create a group event 4.8 A group user who creates a group event is the event creator of that event 4.9 A group user shall receive a notification when a group event is created. 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. 4.11 A group user shall be able to search for group events 4.12 A group user shall be able to sign up for an event 4.13 A group user shall be able to leave a group **5. Group Admin** 5.1 A group admin shall also be a registered user. 5.2 A group admin shall administer at least one group 5.3 A group admin shall be able to add or delete many group members. 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 5.6 A group admin shall be able to delete group events **3. Group**

	3.1 A group shall have at least one group user. 3.2 A group shall have at least one group admin. 3.3 A group shall have 0 or more comments 3.4 A group shall have 0 or more events 3.5 A group shall include the creator of the group 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 **12. Group Event** 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 user who created that event 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

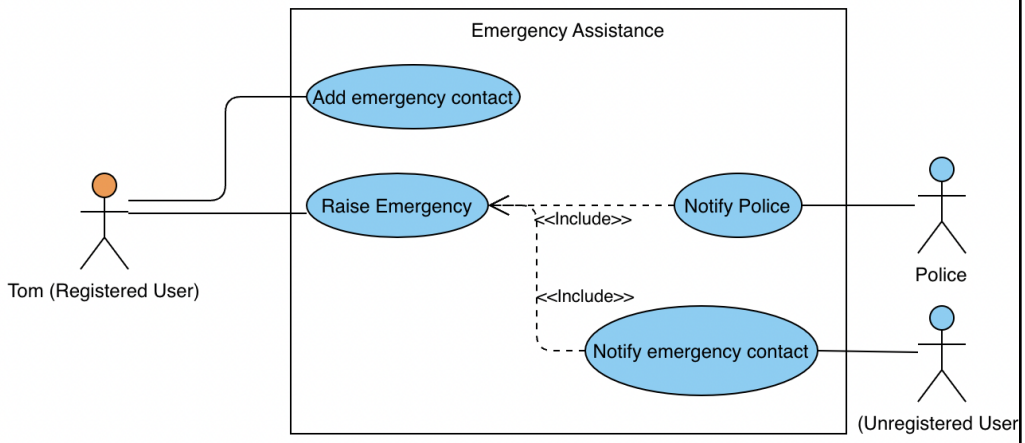
Title	3. Joining Group
Actors	Mary(Registered user), Tom(Group admin)
Description	Mary is a new user on PlayDate. She can either stay without groups, or create or join existing groups. She wants to join an existing group to get started and hence goes through the groups available on PlayDate. The groups are suggested based on common interests she mentions on the profile. Mary finds a group of “hikers”. She thinks she best fits into that group and sends a request to join the group. Tom, who is the admin of that group, approves her request. Now Mary is part of that group and can view all activities of that group.
Diagram	
Requirements	**2. Registered User** 2.33 A registered user shall be able to browse groups **12. Group Event** 12.7 A group event shall include a sign up by the event creator

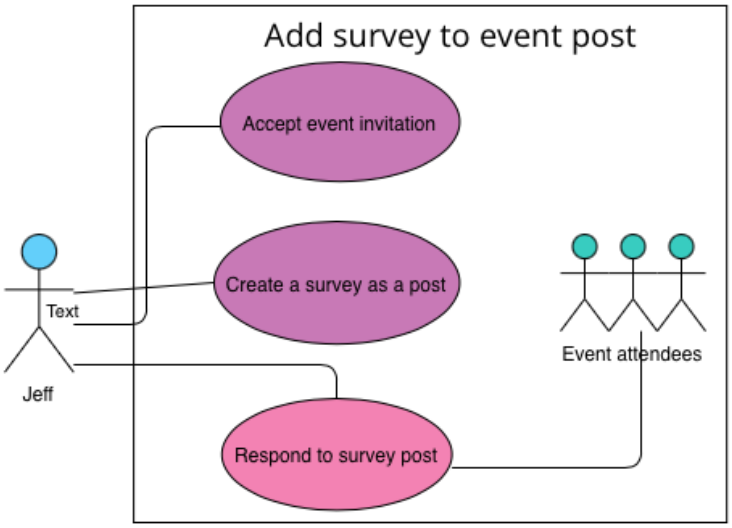
Title	4. Creating Group Events
Actors	Helen, Tom, Ben (Group users)
Description	Helen planned to hike with her son and needed some company for her son. She checks on the group members schedule to see if anyone is available and choses suitable time for the hike. She creates an activity of hiking on Sunday. Helen can either post this only to the group she is part of or invite other users by making the event public. She decided to post it in her group which triggers notification to group members. Tom sees the notification and uses it to sign up for the event. Ben, who missed his notification, sees the event listing on the group and uses that to sign up for the event. Many more parents signed up for this activity. If Helen is the only person on the sign up sheet by the start of the event, this activity post will be removed.
Diagram	
Requirements	**4. Group User** 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. 4.16 A group user shall receive a notification when a group receives a Post.

	12. Group Event 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.

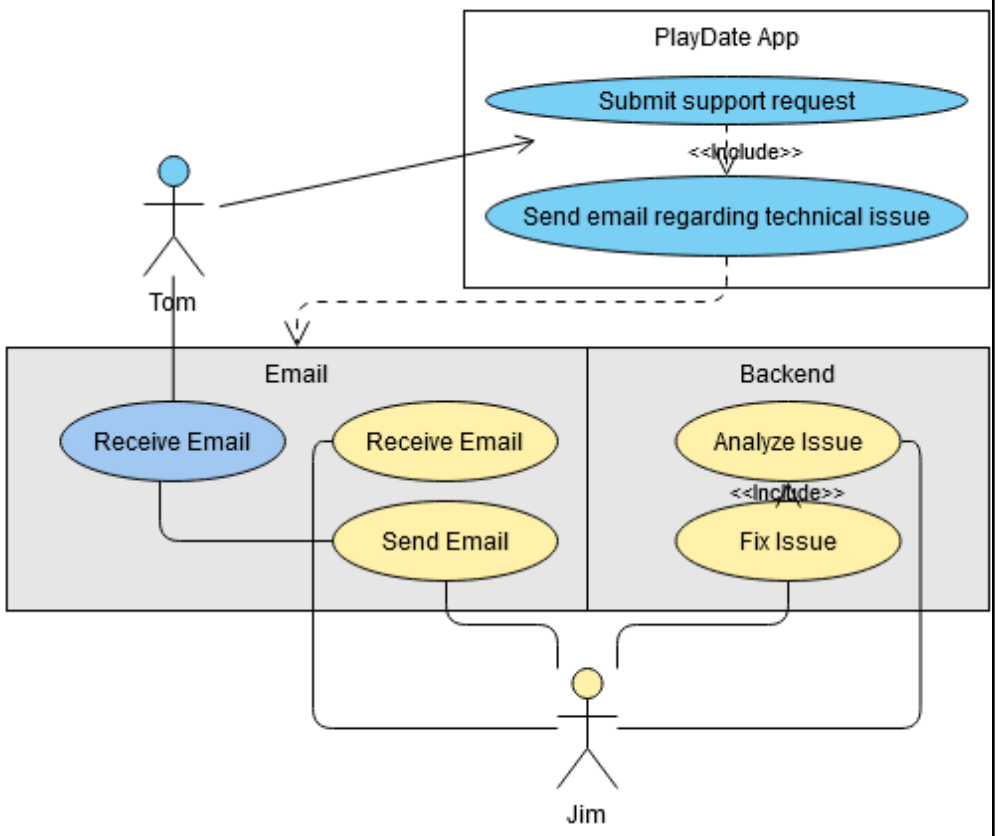
Title	5. Creating Events
Actors	Eric, Al, Cole (Registered Users)
Description	Eric decides he wants to put together a local dog show, mainly because he wants to show off his favorite dog, Barkour. After putting in some planning, Eric creates an event at the lawn of the local library (indeed, after getting the necessary permits!). While he is part of a local dog group, he does want to send out invitations to dog owners in the area, so he sends out a public notification to any dog owners within 10 miles of the event. Al, who loves her pet dog Rune and lives 5.4 miles away, gets the notification and decides to join the event. Eric approves Al's invitation. Cole, who does not have a dog, but does have a child named Patty, searches for nearby events. They decide they want to take Patty to the dog show, so they also try to join the event. Eric, receiving Cole's request, decides to accept the invite. Everyone goes to the event, at which Patty seems to connect with Rune very well, so Cole finds the event listing and sends an invite to Al to join their local group.
Diagram	

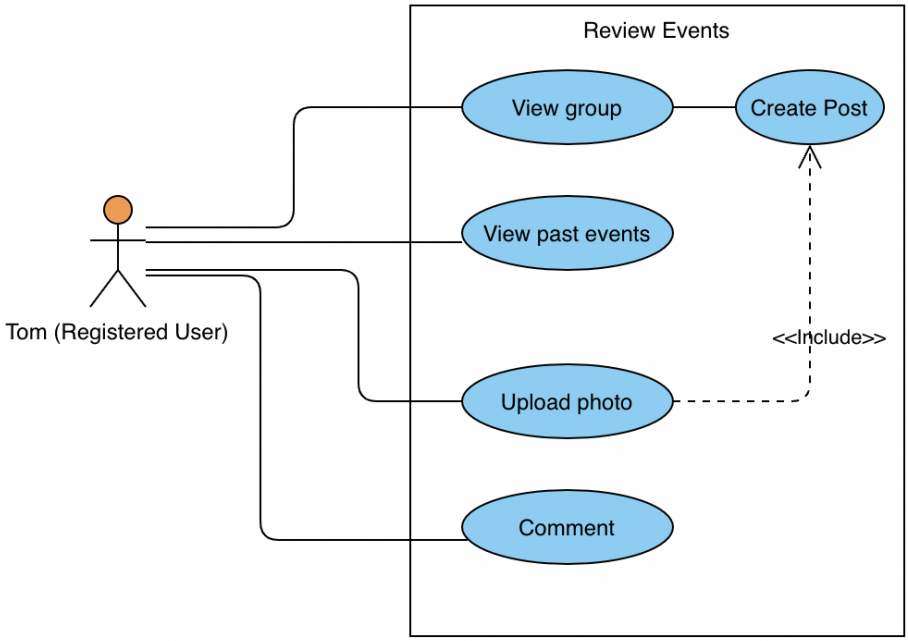
Requirements	**2. Registered User** 2.34 A registered user shall be able to create user events. 2.35 A registered user shall be able to send notification to a selected number of registered users via application. 2.36 A registered user shall be able to search for user events based on location. 2.37 A registered user shall be able to send out user event invites to a filtered user list. 2.38 A registered user can schedule a recurring event. 2.39 A registered user can post on a user event 2.40 A registered user can comment on a user event post **4. Group User** 4.17 A group user shall be able to send group invites to registered users. **17. User Event** 17.1 A user event shall have a user who created that event 17.2 A user event shall require that all sign ups be accepted or denied by the user who created that event 17.3 A user event shall be searchable by all registered users

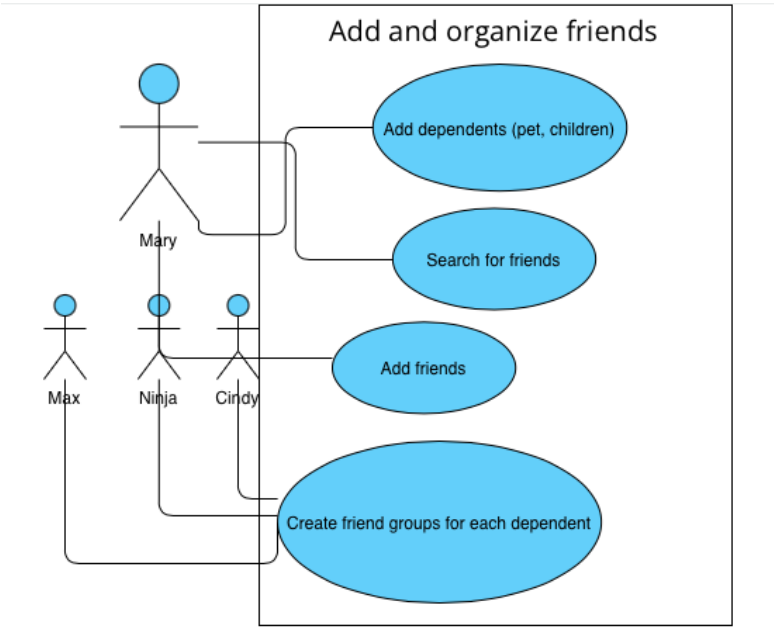
Title	6. Emergency Assistance
Actors	Jack (Registered Users), Police Station
Description	Jack wants to take his son for hiking in Berkeley Hill. He posted this activity on “PlayDate” one day before, but didn’t get any playdates to go with, so he decided to go alone. During their hiking, his son was trapped by a tree root and fell down into a pit and broke his leg. Jack remembers the emergency button on “PlayDate”, he quickly pressed that button and jumped down to check his son. The app automatically sends an emergency message to the nearest police station according to Jack’s schedule which has a location on it along with his contact number.
Diagram	
Requirements	**2. Registered User** 2.41 A registered user shall be able to request emergency assistance via PlayDate application. **16. Emergency Request** 16.1 An emergency request shall be requested by any registered user. 16.2 An emergency request shall be sent to the nearest police via 911 emergency helpline. 16.3 An emergency request shall contain the registered user’s event location and contact number.

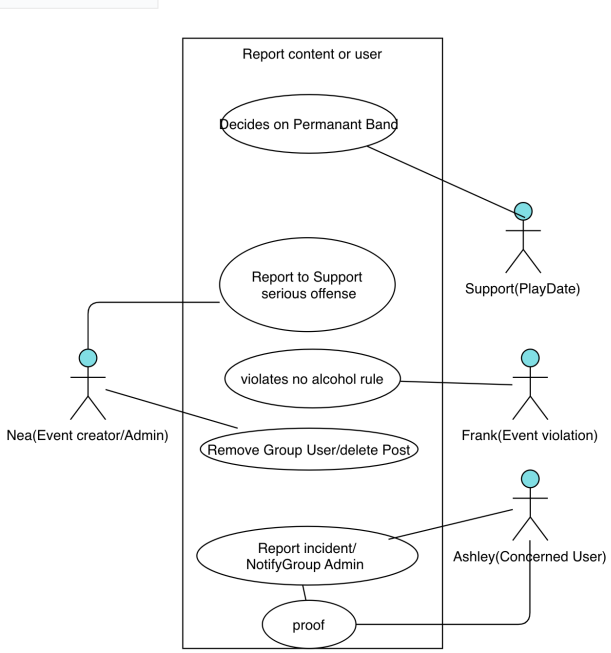
Title	7. Surveys for the event
Actors	Jeff (Registered User)
Description	Jeff was invited to a kids birthday party where everyone was asked to help bring food or beverages to the party. Jeff was going to bring soda but wasn't sure which type of soda people liked best. He knew it was going to be a pretty big party so instead of asking in a post and having to read and count comments, Jeff created a survey that makes it easy to track a count of responses.
Diagram	
Requirements	**2. Registered User** 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. **5. Group Admin** 5.7 A group admin shall be able to delete any surveys on events of their group. **4. Group User** 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 in the group. 4.20 A group user who is attending the event shall be able to respond to surveys corresponding to that group event. 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. **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

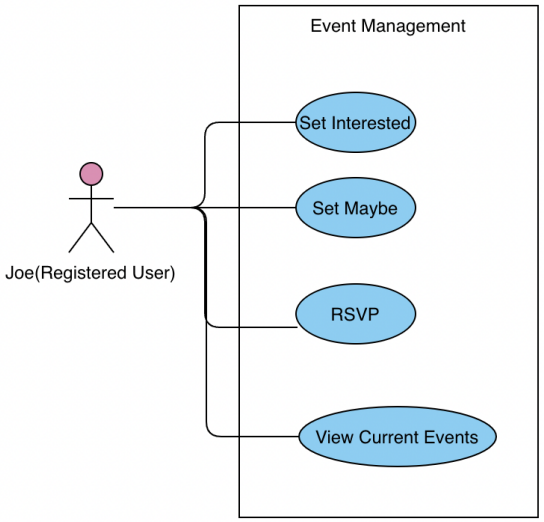
Title	8. Technical Support
Actors	Tom (Registered Users), Jim (PlayDate Support Staff)
Description	Tom wants to share his dog-walk activity with other pet owners, but he keeps failing to post it. Tom finds a support link and uses the built in form to send an email to the support staff. Jim, a member of the support team, fixes the issue and notifies Tom by email. Tom's event goes swimmingly.
Diagram	
Requirements	**2. Registered User** 2.46 A registered user shall be able to request for technical assistance from support staff on product bugs. **8. Support Staff** 8.3 Support staff shall receive emails regarding user technical issues. 8.4 Support staff shall receive emails from registered users **4. Group User** 4.23 A group user shall be able to request for technical assistance from support staff on product bugs.

Title	9. Reviewing an event
Actors	Tom, Helen (Registered Users)
Description	Jeff really loved the event, so after the event Jeff wanted to share his experience with others in the group who couldn't make it due to unavoidable reasons. So he creates a post where he uploads photos and some description on his experience. He also rates the event as 4 stars of 5. Helen, who also attended the event, commented on Jeff's post that she loved it.
Diagram	
Requirements	**4. Group User** 4.24 A group user who created the event shall be able to create a postto 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. 4.26 A group user shall be able to include images in their comments **12. Group Event** 12.10 A group event shall contain the cumulative average of all the ratings as the rating of that event.

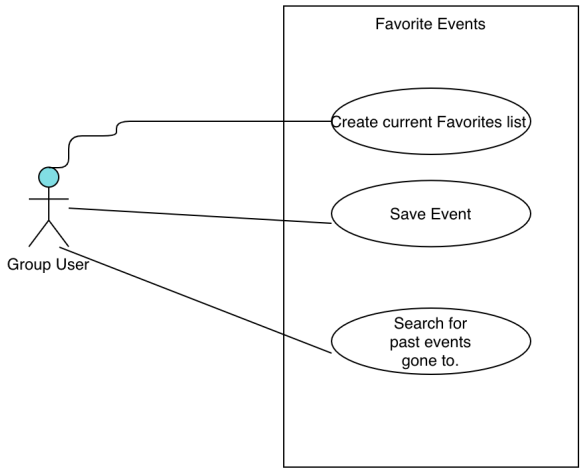
Title	10. Grouping friends into list
Actors	Mary (Registered User)
Description	Mary has a dog Ninja, a son Max, and a daughter Cindy. Her kids are of different ages, Max 5 and Cindy 15. Because of this they mostly have different friend groups that do different activities but they share a small number of some of the same friends. Some of her kid's friends like going to the park with their dog so that they can all play together. Mary wants to be able to group different friends together so that she can keep track of which kids are friends of one of her kids or both.
Diagram	
Requirements	**2. Registered user** 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.

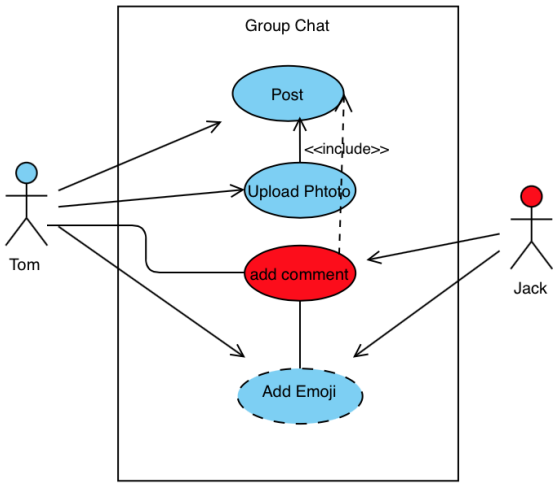
Title	11. Report content or user
Actors	Ashley, Frank (Registered User), Nea (Group Admin), PlayDate (Support)
Description	Ashley was invited to an event for a pool party for the kids. She noticed that Frank posted something about alcoholic beverages being brought to the event. Ashley doesn't think that it is appropriate for Frank to bring alcohol to the event so she reports the content to the event creator, Nea. This report was not a serious report to the app so Ashley decided to report only to Nea and not PlayDate. After the event, Frank continued to irritate others in the group and this annoyed Nea and she decided to report Frank to PlayDate. Nea was given a notice about the report so that the posts could be double checked by her and she can decide if she thinks the subject is appropriate for the event or not.
Diagram	
Requirements	**2. Registered User** 2.50 A registered user shall be able to report other users by contacting support staff. **9. Backend Admin** 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 9.9 A backend admin shall be able to access all the public content of

	events 9.10 A backend admin shall be able to remove registered user from the application. **4. Group User** 4.27 A group user shall be able to contact the support staff to report other group users.

Title	12. View RSVP'd events
Actors	Joe (Registered User)
Description	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 to current events which lists only the events that he has RSVPed
Diagram	
Requirements	**2. Registered User** 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' 2.54 A registered user shall be able to view their list of RSVP'd events 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 **11. Public Events** 11.6 Public events shall be set as 'Interested' by a registered user 11.7 Public events shall be set as 'Maybe' by a registered user **12. Group Events** 12.11 Group events shall be set as 'Interested' by a group user 12.12 Group events shall be set as 'Maybe' by a group user

	17. User Events 17.4 User events shall be set as ‘Interested’ by a registered user 17.5 User events shall be set as ‘Maybe’ by a registered user

Title	13. Save Favorite Events
Actors	Tim(Registered User)
Description	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.
Diagram	
Requirements	**2. Registered User** 2.57 A registered user shall be able to add events to their favorites 2.58 A registered user shall be able to view all of their favorite events. **11. Public Events** 11.8 A public event shall be favorited by zero or more registered users **12. Group Event** 12.13 Group events shall be favorited by zero or more registered users **17. User Event** 17.6 User events shall be favorited by zero or more registered users

Title	14. Chat with other users in the group
Actors	Tom(Registered User), Jack(Registered User)
Description	Tom has 2 dogs and is a long time user of PlayDate. He loves to share photos of his new dog fluffy. So he posts the photos in his group chat. In response to that Jack shared photos of his dogs too and many commented with heart emojis.
Diagram	
Requirements	**4. Group User** 4.28 A group user shall be able to reply to group posts with emojis.

3. Main data items and entities

1. **General users**: Can view, search public events of organizations but not public events created by users.
2. **Registered users**: Can view, post, edit, delete event activities, join groups and sign up for public user activities.
3. **Group**: A collection of registered users who share an interest and whose dependents are more likely to meet.
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.
5. **Group Admin**: Administrator or a group, which was created by him/her and has rights to add and remove group users of that group.
6. **Account:** general users can register the "PlayDate" system, and every user will have an account.
7. **Roles:** including general user, registered user, admin, group user, and group admin. Every user has an account and a role.
8. **Support Staff**: Service team is the support team with whom the users can connect incase of any issues with application
9. **Admin:** Admin is responsible for background-checking when a user registers the "PlayDate" system. Admin shall also check appropriateness of posts and can delete inappropriate posts and remove users.
10. **Events**: An event is a combination of date and place where a group or collection of users can meet.
11. **Seeders/Public Event**: available locations for children/pets hang-out, like parks, libraries, book stores, and museums, and upcoming public events. These seeders are posted for general users to view and do a search.
12. **Group Event**: An event tied to a specific group.
13. **Comment**: A piece of user-generated content attached to a post.
14. **Dependents**: Children or pets that are under the purview of a user
15. **Survey**: A poll for asking group users questions regarding an event
16. **Emergency Contact**: Each registered user shall have the ability to call for emergency services based on their location.
17. **User Events**: A type of event created by a registered user, but not associated with a group.
18. **Post**: A piece of user-generated content attached to a group or event.

4. Functional Requirements

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.
   1.6 A general user shall be able to upload proof of the parent of a kid or pet.
   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.
   2.5 A registered user shall have an address in their profile.
   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
   2.8 A registered user shall be able to edit their Address in profile
   2.9 A registered user shall be able to edit their Name in profile
   2.10 A registered user shall be able to edit their Birth Date 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
   2.15 A registered user shall be able to edit their Dependents' schedule
   2.16 A registered user shall be able to send a referral link to a friend.
   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
   2.20 A registered user shall be able to comment on public event posts
   2.21 A registered user shall be able to log out
   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 location
   2.27 A registered user shall be able to search for groups based on interest

2.28 A registered user shall be able to search for groups based on 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 apply to join an existing group
2.32 A registered user shall be able to search for users nearby an address
2.33 A registered user shall be able to browse groups
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.
2.37 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.
2.39 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.
2.42 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.
2.46 A registered user shall be able to request for technical assistance from support staff on product bugs.
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.
2.50 A registered user shall be able to report other users by contacting support staff.
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’
2.54 A registered user shall be able to view their list of RSVP’d events
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
2.57 A registered user shall be able to add events to their favorites
2.58 A registered user shall be able to view all of their favorite events.
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 user’s events
2.63 A registered user shall be able to comment on other users’ posts
2.64 A registered user shall be able to cancel their sign up to an event

2.65 A registered user shall be able to search for users by username
2.66 A registered user shall be able to search for users 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**

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

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 posts
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
4.7 A group user shall be able to create a group event
4.8 A group user who creates a group event is the event creator of that group
4.9 A group user shall receive a notification when a group event is created.
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.
4.11 A group user shall be able to search for group events
4.12 A group user shall be able to sign up for an event
4.13 A group user shall be able to leave a group
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.
4.16 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.
4.18 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.
4.23 A group user shall be able to request for technical assistance from support staff on product bugs.
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.
4.26 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.
4.28 A group user shall be able to reply to group posts with emojis.
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**

5.1 A group admin shall also be a registered user.
5.2 A group admin shall administer at least one group
5.3 A group admin shall be able to add or remove many group members.
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
5.6 A group admin shall be able to delete group events
5.7 A group admin shall be able to delete any surveys on events of their group.
5.8 A group admin shall be able to invite a registered user to the group
5.9 A group admin shall be able to delete group posts

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**

9.1 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
9.4 A back end admin shall be able to remove users from 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
9.9 A backend admin shall be able to access all the public content of events
9.10 A backend admin shall be able to remove registered user from the application.

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 user who created that event
10.6 An event shall set the event creator as the user who created that event
10.7 An event shall be either a public event, a group event, or a user event

11. Public Events

11.1 A public event can be searched for by general users
11.2 A public event can be searched for by registered users
11.3 A public event can be viewed by general users
11.4 A public event can be viewed by registered users

11.5 A public event can be posted on by registered users
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 Event**

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 user who created that event
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
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
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. **Comments**

13.1 A comment shall be created by a registered user
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
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
14.5 A dependent shall have an availability schedule
14.6 A dependent shall be managed by their associated registered user

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 Request**

16.1 An emergency request shall be requested by any registered user.
16.2 An emergency request shall be sent to the nearest police via 911 emergency helpline.
16.3 An emergency request shall contain the registered user's event location and contact number.

17. **User Event**

17.1 A user event shall have a user who created that event
17.2 A user event shall require that all sign ups be accepted or denied by the user who created that event
17.3 A user event shall be searchable by all registered users
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
17.7 A user event shall be able to be made public by the user who created it

18. **Post**

18.1 A post shall be made on an event or group
18.2 A post shall contain text
18.3 A post shall be able to contain images
18.4 A post shall be created by a registered user
18.5 A post on a group shall be created by a group user of the group
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

5. Non-functional Requirements

1. Privacy:
    1.1. Every general user shall sign PlayDate Terms of Service and Privacy when they register.
    1.2. Group admin shall sign an agreement to be responsible for appropriate data sharing in groups.
    1.3. Users violating PlayDate Terms of Service should be warned for the first time and removed for the second time.
2. Usability:
    2.1. Users shall receive online help from support for any assistance on the application.
3. Response time:
    3.1. General User shall receive an update on his/her background verification in less than 24 hours of registering on PlayDate.
    3.2. Support staff shall respond to users within 12 hours of raising request.
    3.3. Support staff shall send a notification to the backend admin within 30 minutes of receiving reports on inappropriate posts or technical issues.
4. Security:
    4.1. PlayDate org should do background-checking when a general user registers on PlayDate.
    4.2. Information should be securely transmitted to the database server without any changes in information.
5. Compatibility:
    5.1. Application should be supported on Mac via browsers of versions, Chrome >= 60, Safari >= 12
    5.2. Application shall be supported on Windows via browsers of versions, Chrome >= 60
6. Business need
    6.1. Application should use googleapis to find nearby places to visit.
7. Data Storage
    7.1. The application’s back-end servers should never display a customer’s password.
    7.2. The application’s back-end servers should only be accessible to authenticated backend admins.
    7.3. User data should be deleted from the backend servers if the user deletes their account.
8. Legal, marketing, copyright
    8.1. Application should display the disclaimers, copyright of the advertisers

while advertising on the application.

8.2. Application should use only open source images in the web application.

6. Competitive analysis

Competitor/ Feature	**510 family**	**Meetup.com**	**Facebook events**	**Nextdoor**	**Play:Date**
Strengths	Events available with information on the front page. Easy interface set by different topics by age, location, activities	+Able to create groups, public/private events +Able to find groups, people, events pertaining to certain specifications(age, interests, friends) +Notification System +Messaging System +local guides: informational posts that can help fuel tourism. +Reporting System that can help mitigate against unsafe posts and members +Multiplatform(web, ios, android)	Large user base, offers a variety of different content and events	Local News and connection suggestions. Strong community engagement	Easy to learn, good onboarding, kid-oriented
Weaknesses	-restricted to specific area	Pricing restrictions Website requires no identity verification.	Too many ads	Restricted to surrounding area	Mobile only, swipe system means one match at a time, profiles may be far away, simple chat
Pricing	Free	Free-Tier Paid-Tier Option 1: $14.99/ month for a group	Free	Free	Free

		upto 50 members and 3 co-organizers Option 2: $19.99/month with no restrictions			
Security	Email only	Simple login through Google, Apple, Facebook	Two-factor authentication and encrypted messaging with FB Messenger	Simple login through Google, Apple or Facebook	All you need is a phone number
Social Media	Facebook, Instagram, Pinterest, and twitter	Strong social media(Instagram, FB, twitter, youtube) presence in alignment with current-day diversity standards.	One of the most used social media platforms that also owns other popular social media platforms, like Instagram	Local news and comments are similar to facebook for community	Pic, age, general location, likes/dislike (pretty thorough on this)
Onboarding experience	No assistance and user needs to self learn about application	Modular and easy to follow website Onboarding assistance is available	Helpful hints and pop up bubbles	Easy to create and look up local events	Tutorial slides and profile creation sequence.

Features	510 family	Meetup	Facebook events	Nextdoor	Play:Date (myplaydateapp)	PlayDate
Private and public events	++	++	+	++	-	++
Reporting & Moderation	+	++	++	+	+	++
Privacy & Visibility Tools	-	++	+	+	-	+
Emergency Tools	-	-	+	+	-	++
Communication	++	++	++	+	+	+
Contact Suggestions	++	++	++	++	++	+
Data Security	-	+	+	-	-	++
Usability & Flexibility	+	++	++	+	++	+
User Verification	-	-	+	-	+	++

++ Superior +Feature present -Feature doesn't exist

Analysis

1. 510 family: 510 family helps parents find fun events for their kids.
    - This application is restricted to kids and not available for pets whereas our application caters to events of pets along with kids.
    - This application only provides information on events and does not allow users to sign up. This means the users wouldn't know when the event has closed the registration. To resolve this issue we in PlayDate let users create events and track the number of users signing up and users can see the number of users who have already signed up.

2. Meetup: Application for users to meet and create events.
    - Firstly, Meetup caters to a variety of audiences. When we looked for events based on various states, the number of events for kids in California was around 10-12, which is a very small number. As the platform is open to a wide variety of

audience, this can compromise with the safety of kids, due to which many parents may not prefer to use this application to find playdates for kids and pets.
    - This application is free to create accounts but is priced based on tier subscriptions. This can discourage the users from using the application. Whereas PlayDate does not charge its users and let them enjoy the process of creating and attending events for free.

3. Facebook.com: Social media platform for making friends.
    - Facebook is the most popular and strong competitor for PlayDate. Mainly because of the number of users who are already onboarded to the platform due to its existence for a long time. To tackle this PlayDate needs to provide experiences that Facebook currently doesn't provide.
    - Facebook is an open platform for anyone to join and same goes with the Facebook groups. Users can form groups and also create events. Users can see the number of users who have signed up. The group admins let people join groups, where neither facebook nor the group admins track the spam users from joining groups and group events. To solve this issue PlayDate lets only authorized users sign in the application, either by undergoing background verification or recommended by existing PlayDate users.
    - Facebook search for groups and events is based on the group name and event name consisting of the search keywords. Which means the event names that are not consisting of those keywords will not appear in search results. Whereas PlayDate lets events have tags which the creator of the event adds. For example an event can be named xyz and can have tags of hikes, trails, mist, yosemite etc, so that other users searching for these keywords can be easily mapped to relevant events irrespective of what the event is named.

4. NextDoor: Social network for local events and news sharing
    - Nextdoor has faced a lot of criticism for phony childcare providers. This is mainly due to the lack of user verification. PlayDate solves this problem with its secure onboarding, which lets only legit users use the application.

5. myPlayDateapp: Application for parents to find playdates for kids.
    - This application does not implement its features on web applications but is available on Appstore and Google Play.
    - The application focuses on pairing up users. Our focus is on larger groups and creating events for people to sign up for.
    - The application has a very small social media presence and few users onboarded. Many users have complained that the search returns very few matches, meaning there is a small user base. This drawback of myPlayDateapp can make us capture

the market. Our application will partner with public institutions, who can showcase their public events on our Public events page, which is available for all general users without login. This will encourage users by providing good event content and increase chances of general users signing up on our application.

- The application does not provide onboarding assistance. Whereas our application will allow all the users to be able to contact support staff for any assistance with using the application.

Pricing: Similar to most of our competitors, PlayDate doesn't charge their users, which will encourage more users to join our platform. The revenue model lies on the advertisements in the application. Advertisers in the domain of pets and kids shall be able to display their products on our web application. The products shall be displayed on the pages of events based on type of events. If the event is related to trek then trekking items shall be advertised on the events page, so that users can choose to buy those products that they might need for that event. By this the users can find beneficial materials for their events and at the same time we will be able to market products of advertisers.

7. High-level system architecture and technologies used

- **Client Layer**
  - Users can access the application via browser using web url http://34.168.80.213:8000/home/about/ which is accessible from on premise as well from google cloud.
  - Users can access limited functionalities of the application directly which includes About page, public events and Support.
  - To be able to access all the functionalities, the user can Register and Login on the web application, post which user will have access to Profile, Groups, Group & Private Events, depending on the role of the user.
  - Users can only interact with the user interface and cannot access business logic or any data stored on the server.

- **Server Layer**
  - Web application is hosted on Google compute engine on Google cloud. Here the business logic of the web application is stored.
  - Web application uses Django Framework. Django framework routes the url to the respective user interface (views).
  - Business logic performs read, write, update and delete operations on the Data Layer via MySQLclient connector in mysql.connector.django
  - Googleapis will be used for recommendation logic of nearby places

- **Data Layer**
  All the application & user data is stored on the MySQL database. MySQL server resides on the same server as the web application, which increases speed of operations.
  Sensitive data like passwords will be encrypted and stored on the server.
  Stores information like:
  - User
    - Personal Identification Data: Name, Date Of Birth, Address, Profile photo, proof for background verification
    - Authentication details: Username, Password, gmail account
    - User shared data: Images, Text, event demographics
    - Permissions
  - Public Events- Event information from google maps
  - Groups- Demographics
  - Advertisement Details
    - Text / Media
    - Vendor Details

Software stack

- Compute server: Google Compute Engine
    - Name: team03-main-01
    - Machine-type: e2-micro
    - vCPU: 1 shared core
    - Memory: 1gb Storage: 30gb
    - OS: Ubuntu 18.04
- Web server: Gunicorn 20.1.0
- Database: MySQL 8.0.29
- Server-side language: Python 3.10 with Django 4.0.5 Framework
- Front-end: Bootstrap 5
- IDE: VisualCode
- Version Control: GitHub

8. Checklist

Team found a time slot to meet outside of the class Github master chosen

- Done

Team decided and agreed together on using the listed SW tools and deployment server

- Done

Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing

- Done

Team lead ensured that all team members read the final M1 and agree/ understand it before submission

- Done

Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)

- Done

9. Team contribution

Name	Role	Contribution
Soujanya Ravindra Nayak	Team Lead Document Contributor Backend Team	- Organized meetings and brainstormed project ideas and implementaion. - Created initial django project skeleton and pushed it to github. - Created a view for About Page and view page for my bio. - Proof verified web application. - Contributed to M1 document in format, summary, use cases, diagrams, requirements and system architecture. - Monitored timely completion of tasks by the team members. - Revised M1V2 based on feedback
Margaret De La Torre	Front-end lead Document Contributor	- Helped brainstorm ideas for project - Helped write use cases, diagrams, and requirements - Helped with reminding the team about use cases needing to be more like stories rather than short descriptions - Helped with competitive analysis - Contributed to group discussions during team meeting
Martin Salvatierra	Front-end Document Contributor	- Applied three brainstorming topics for full application - Helped to write 2 use case, diagram creation, and requirements - Addition of three functional requirements - Ask for elaboration on multiple parts of project during team meetings - Going over javascript,html,css, bootstrap for future - over 10 hours of python practice on own to learn python - filled in about me page
Andy Cho	Back-end lead Document Contributor	- Applied constructive criticism to brainstorming topics - Helped write use cases, diagrams, and requirements, -Helped productivity by taking availability-schedules, posting resources

		- Created cloud server with necessary software to match our project needs - Helped with competitive analysis - Revised M1V2 based on feedback
Qin Geng	Front-end Document Contributor	- Finished two use cases (2.1 and 2.2) with diagrams and contributed to corresponding entity list, functional requirements, and non-functional requirements. - Wrote the first version of the “about us” django application to help team members learn about django projects. Also modified the second version application to help team members get a unified template to work on. - With help from the back-end lead, wrote an instruction document on how to deploy our application on a google cloud server step by step to help team members understand the deployment procedure. - Engaged in every team meeting, contributed to brainstorming ideas and problem discussion. - Revised M1V2 based on feedback
William Plachno	Git Master Back-end Team Document Contributor	- Helped with use cases - Made use case diagrams for UC2.4, UC2.5, and UC2.8 - Edited document - Contributed to group discussions during team meetings - Filled in about me page - Added and edited data entities and requirements - Helped with conceptualizing data entities and requirements - Downloaded and reviewed Play:Date app for Competitive Analysis - Revised M1V2 based on feedback
Victor	Front-end Document Contributor	-helped brainstorm ideas on project -contributed on cast studies -wrote and pushed about me -participated and contributed in meetings that was able to attend but was caught up on chats either in groups or individual messages. -contributed to use cases 2.12 and 2.14 and made the diagrams

SW Engineering CSC648-848

Summer 2022

Milestone 2 V2

“PlayDate” Application — by Team 03 (the “Babysitters”)		
Name	**Job**	**Email**
Soujanya Ravindra Nayak	Team Lead	soujanyaravindra@gmail.com
Margaret De La Torre	Frontend Lead	mdelato1@mail.sfsu.edu
Andy Cho	Backend Lead	andrewyunejo@gmail.com
Martin Salvatierra	Frontend Team	martysaljhost@gmail.com
Qin Geng	Frontend Team	gengqin50@gmail.com
William Plachno	Git Master	wjplachno@gmail.com
Victor Callejas	Frontend Team	vcallejas@mail.sfsu.edu

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

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

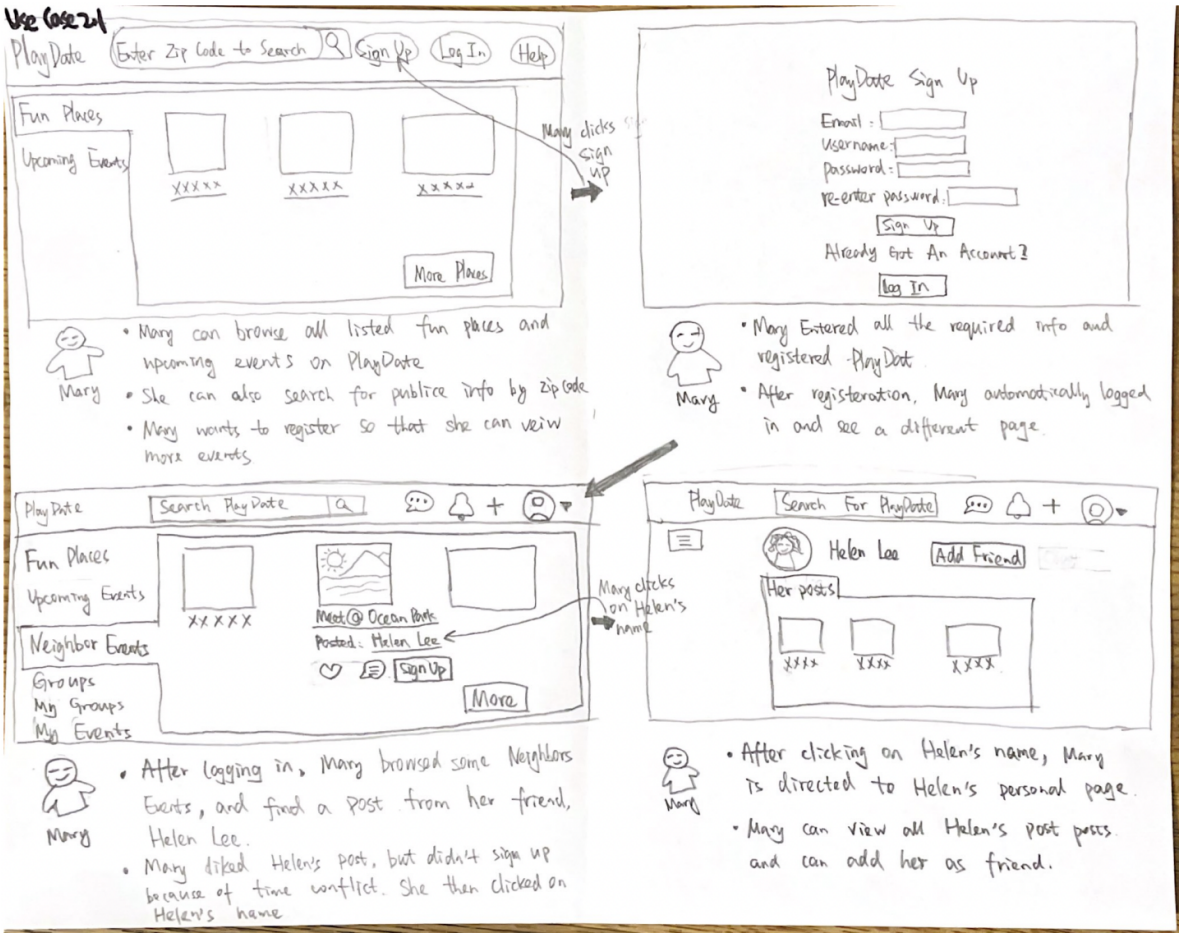
16. Emergency Contacts
No Priority 3 Functional Requirements for 16. Emergency Request

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

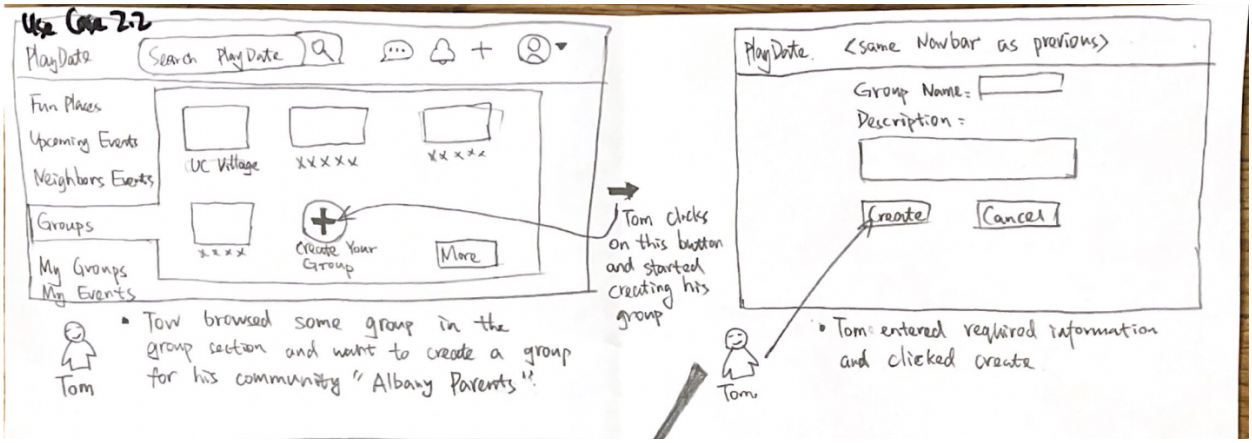
18. Posts
No Priority 3 Functional Requirements for 18. Posts

3. UI Mockups & Storyboard

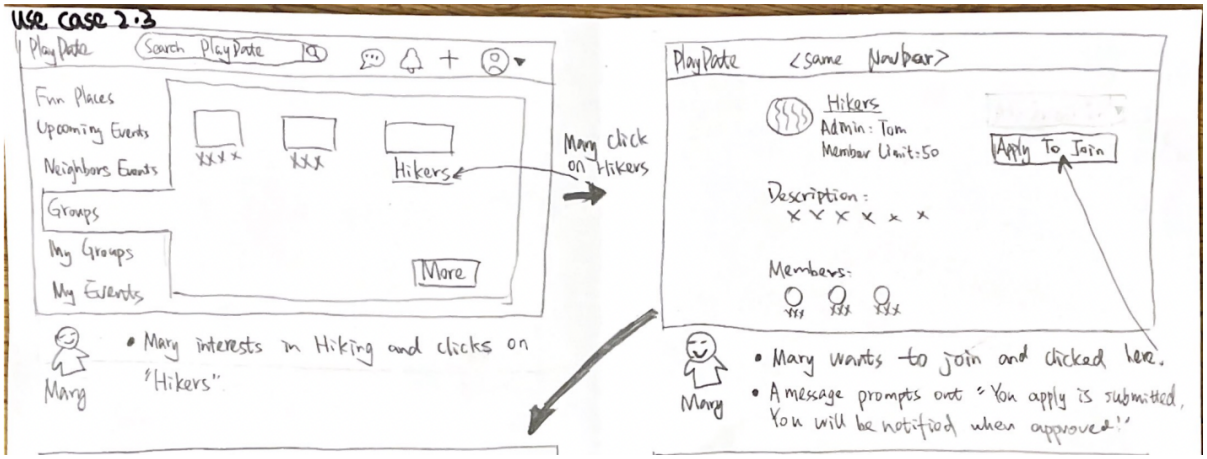
1. Use Case 1: New user registration



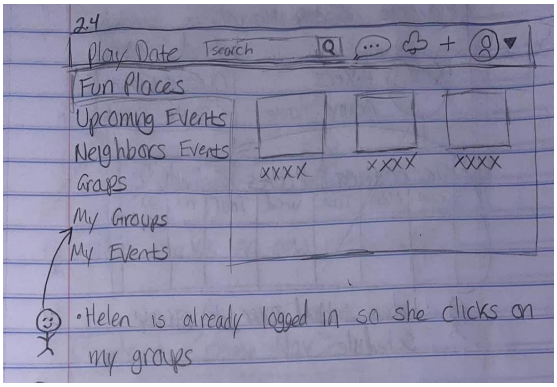
2. Use Case 2: Group Creation

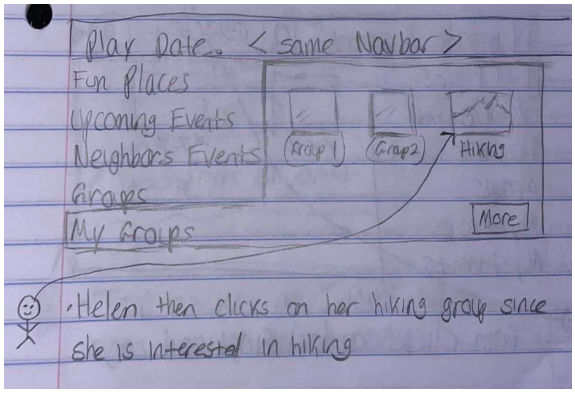


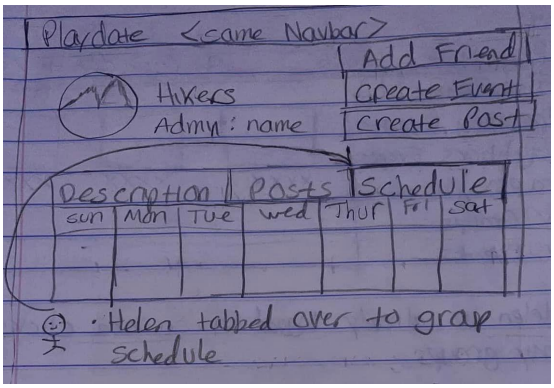
3. **Use Case 3: Joining group**

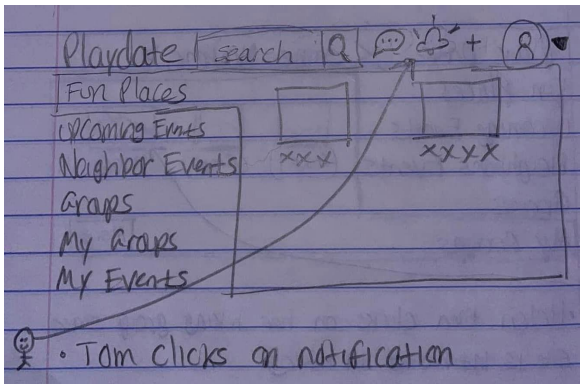


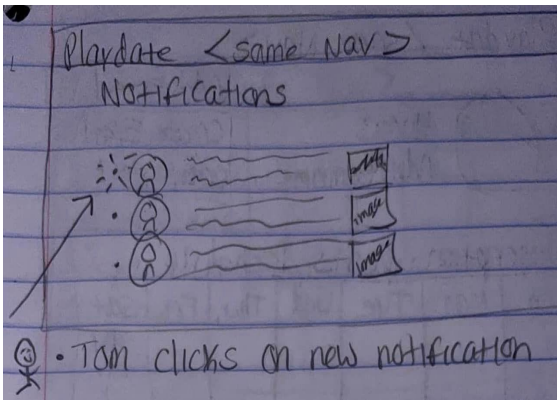
4. Use Case 4: Creating Group Events

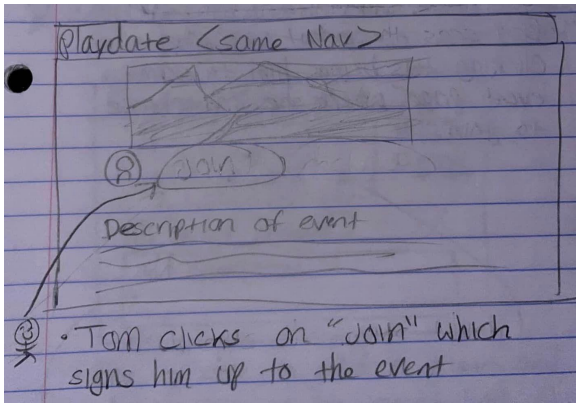


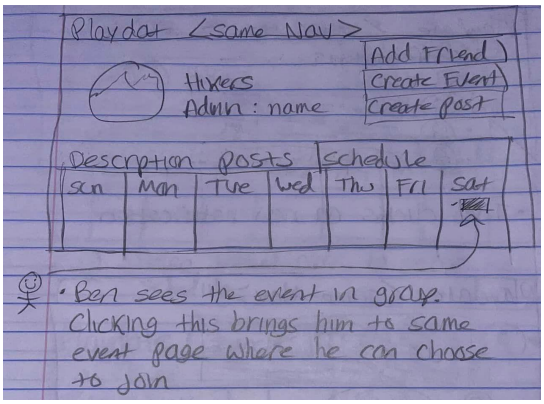




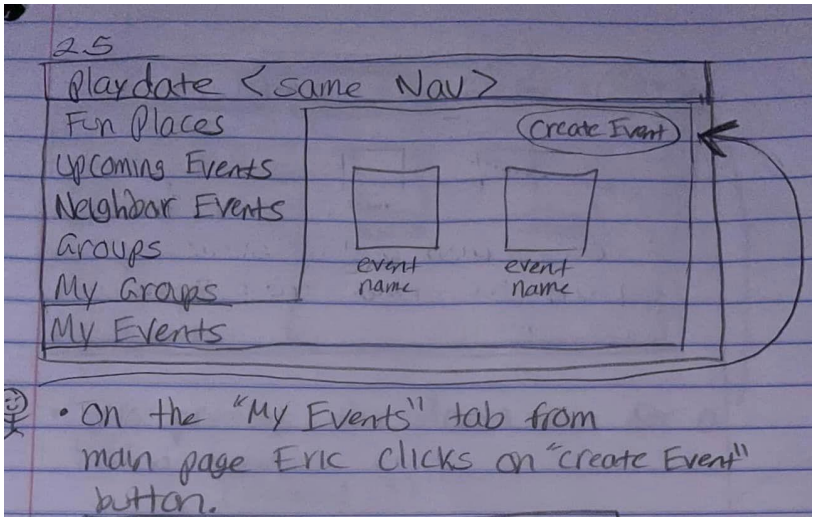


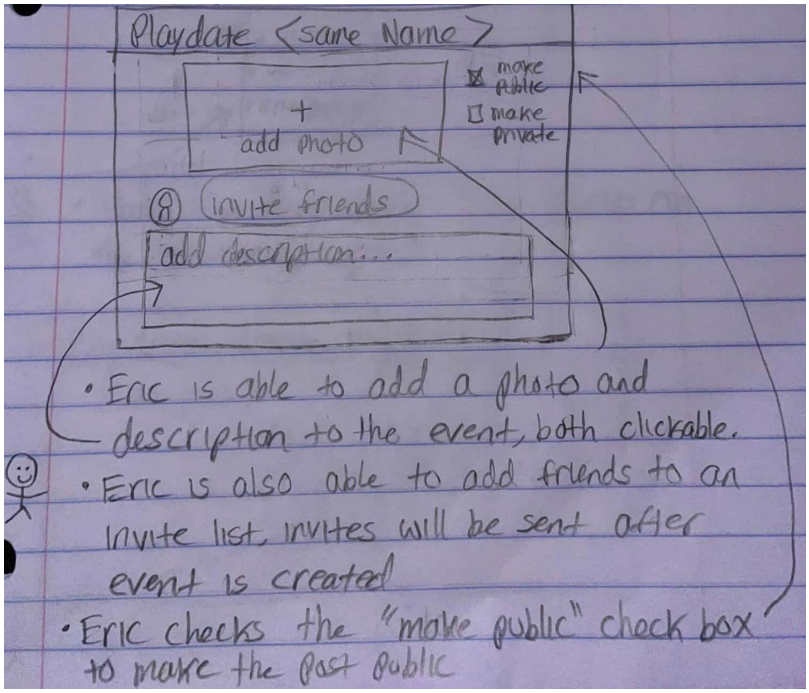




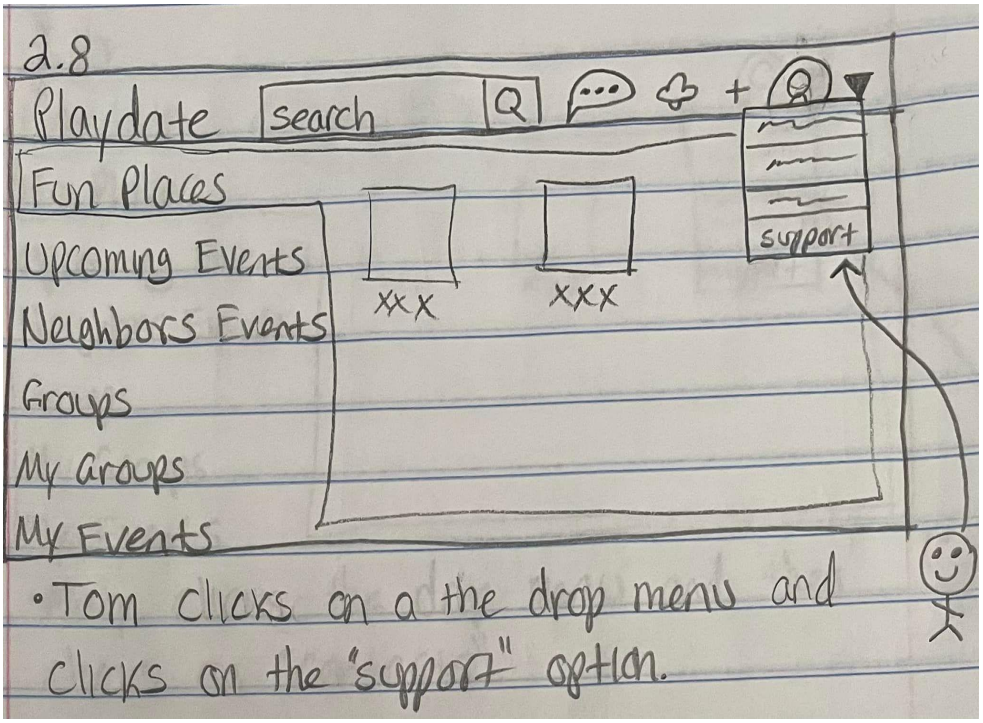


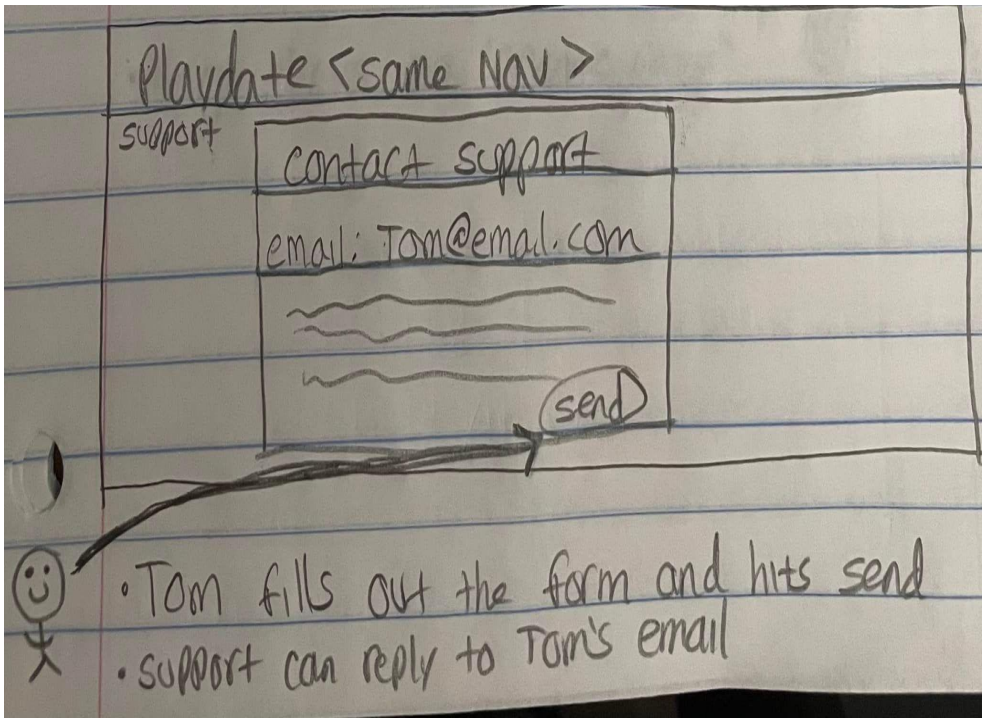
5. Use Case 5: Creating Events



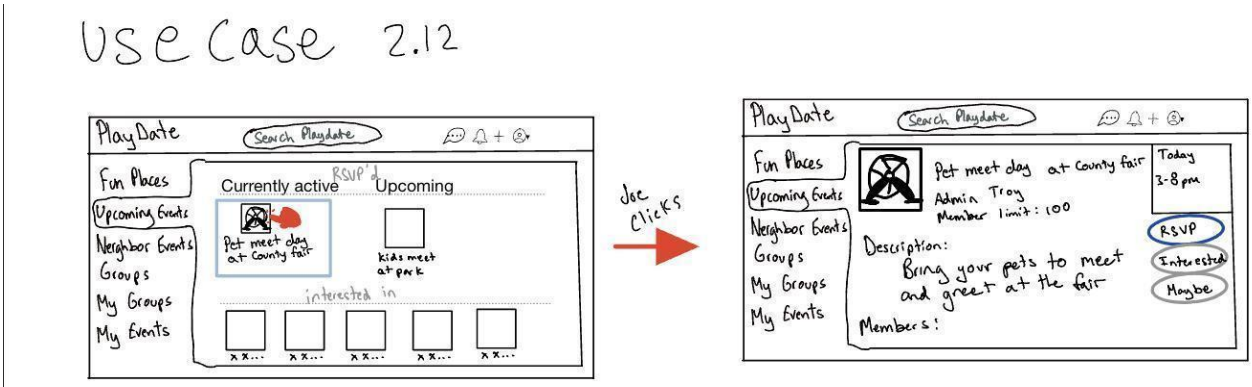


6. **Use Case 8: Technical Support**





7. Use Case 12: View RSVP'd event



?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

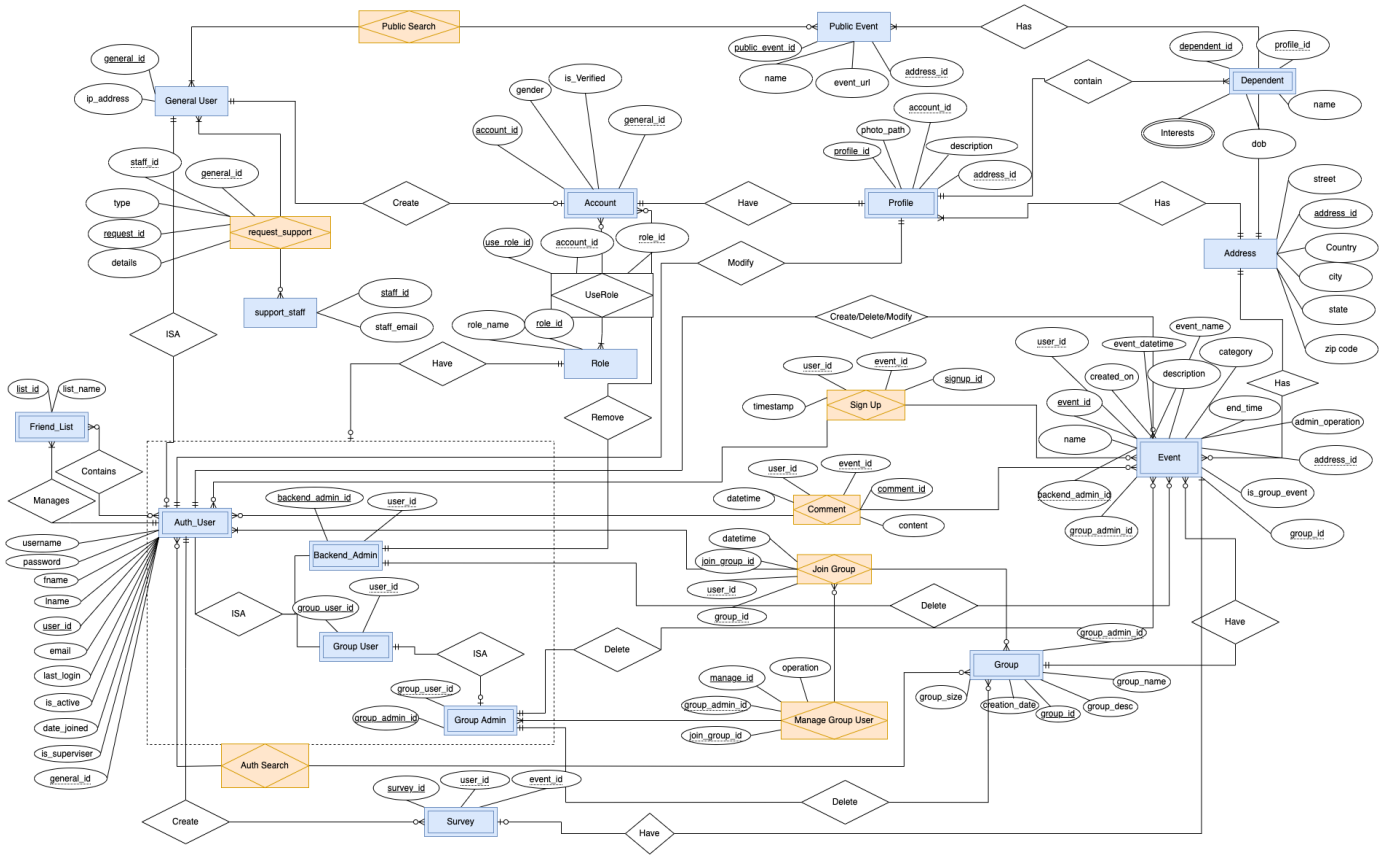
xi. Events (Weak)
* event_id: strong key, numeric
* user_id: weak key, numeric
* content: alphanumeric
* datetime: datetime

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

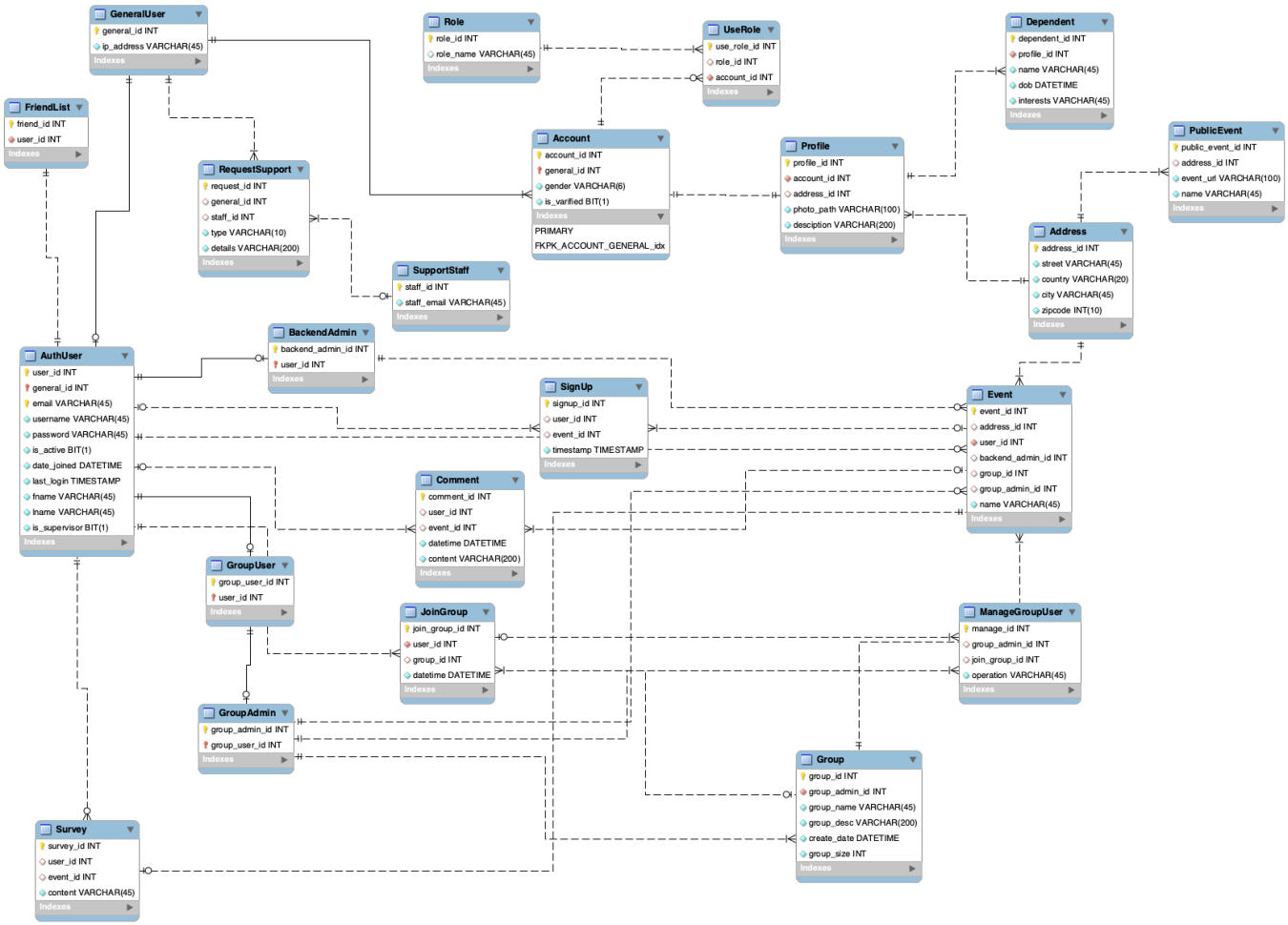
xiii. Dependents (Weak)
* dependent_id: strong key, numeric
* profile_id: weak key, numeric

* 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 icontain, 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 it’s 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:**C 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.

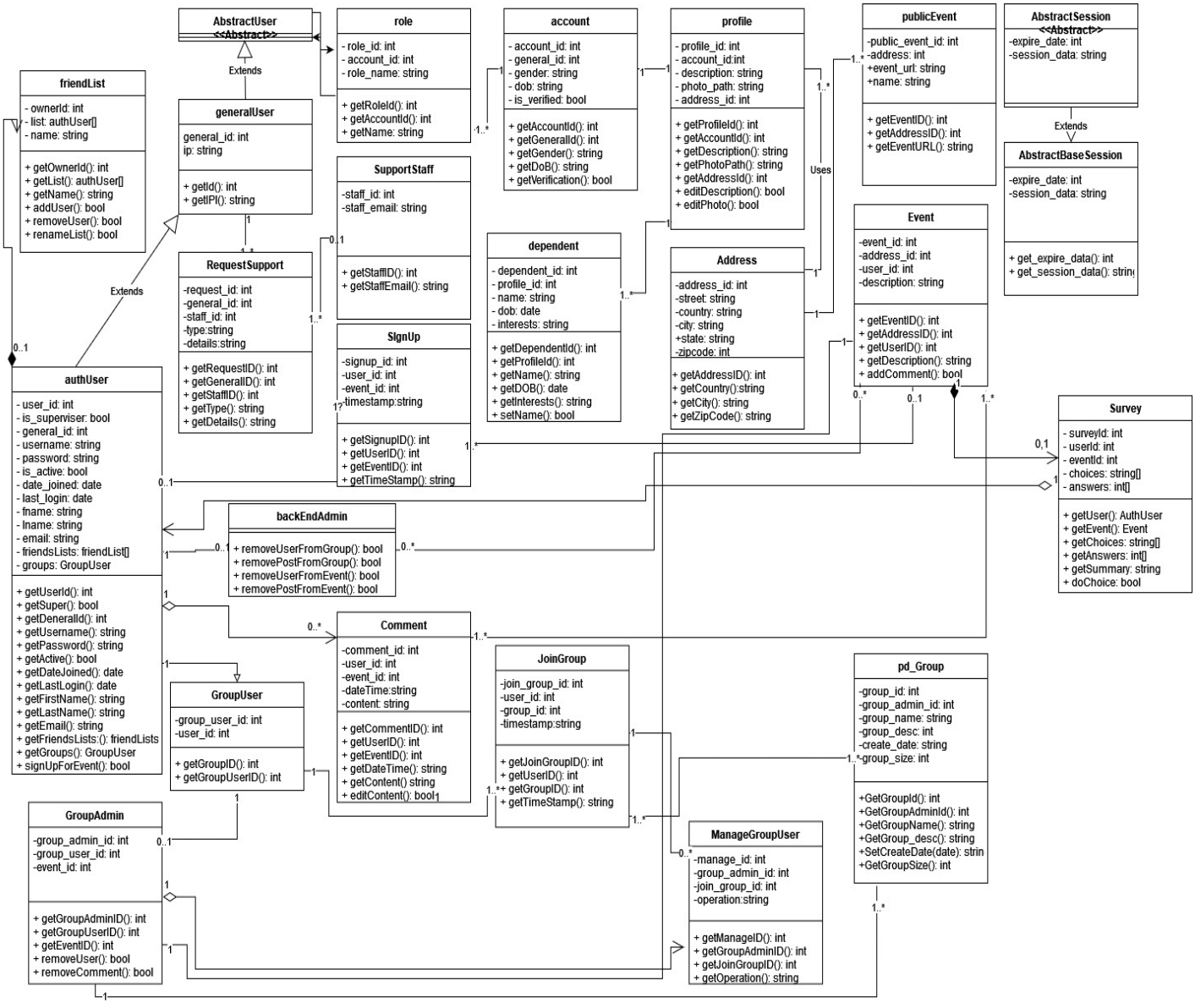
Algorithms

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

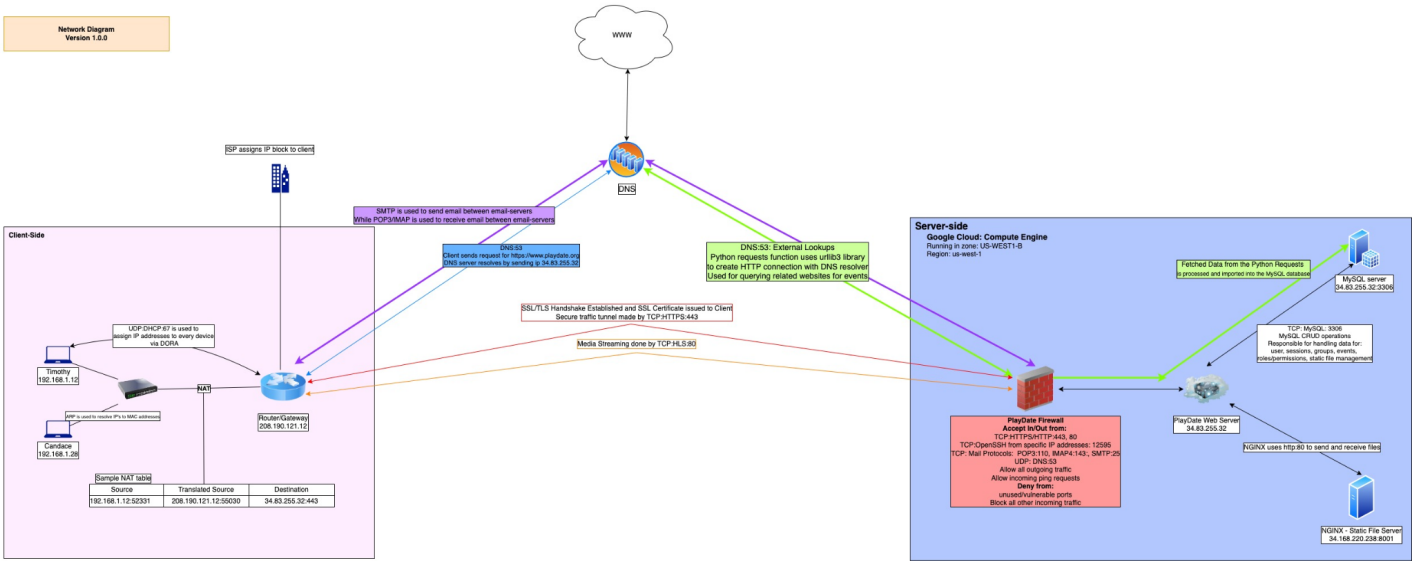
There have been no changes to our choice of software and frameworks between Milestone 1 and Milestone 2.

6. High Level UML Diagrams

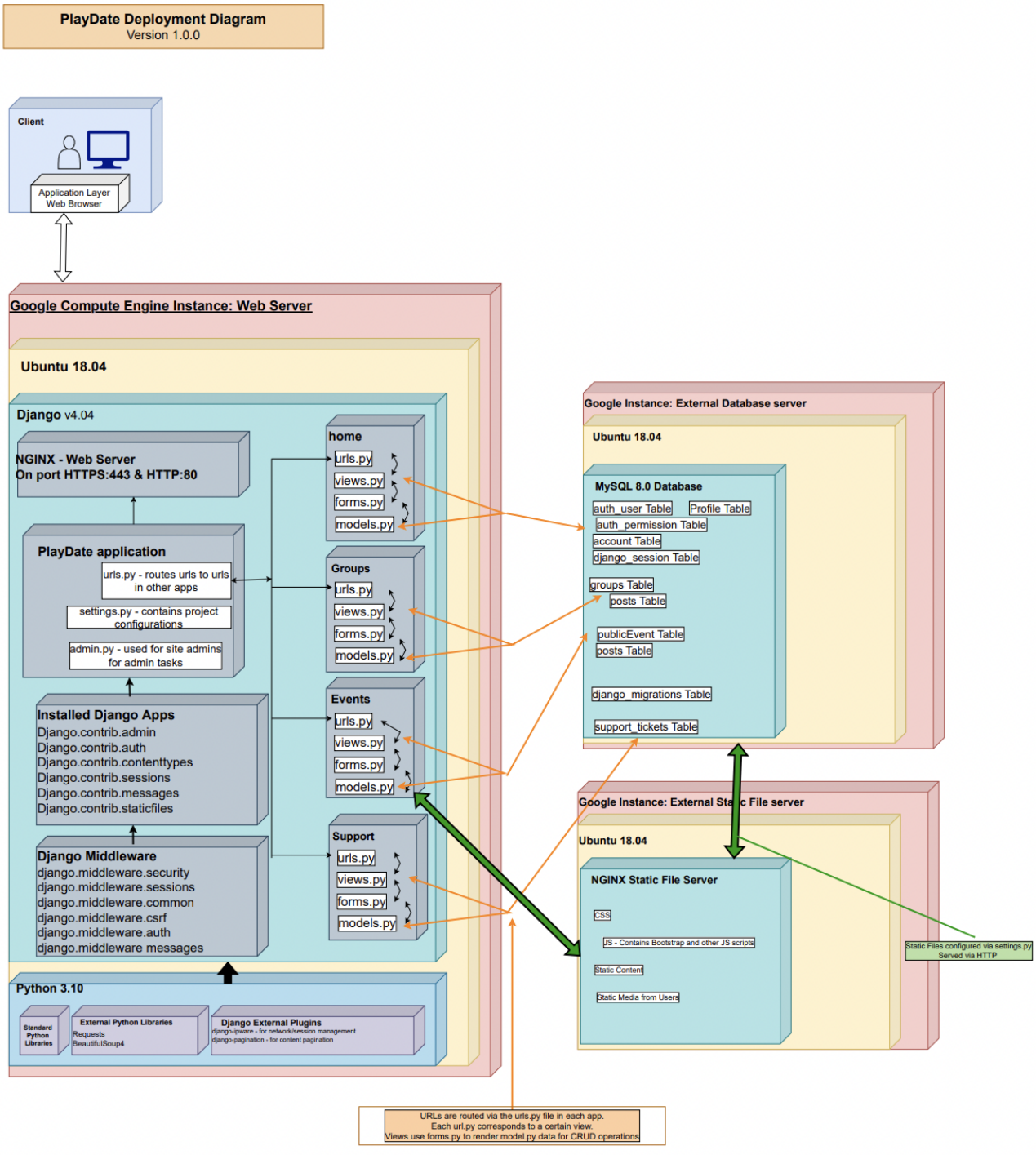


7. High Level Application Network and Deployment Diagrams

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 parallely 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

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

SW Engineering CSC648-848

Summer 2022

Milestone 3 V2

“PlayDate” Application — by Team 03 (the “Babysitters”)		
Name	**Job**	**Email**
Soujanya Ravindra Nayak	Team Lead	soujanyaravindra@gmail.com
Margaret De La Torre	Frontend Lead	mdelato1@mail.sfsu.edu
Andy Cho	Backend Lead	andrewyunejo@gmail.com
Martin Salvatierra	Frontend Team	martysaljhost@gmail.com
Qin Geng	Frontend Team	gengqin50@gmail.com
William Plachno	Git Master	wjplachno@gmail.com
Victor Callejas	Frontend Team	vcallejas@mail.sfsu.edu

Date	Version
07/28/2022	M3V2
07/19/2022	M3V1
07/19/2022	M2V2
07/19/2022	M1V2
07/07/2022	M2V1
06/21/2022	M1V1

Table of contents

1. Data Definitions V3

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. Functional Requirements V3

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

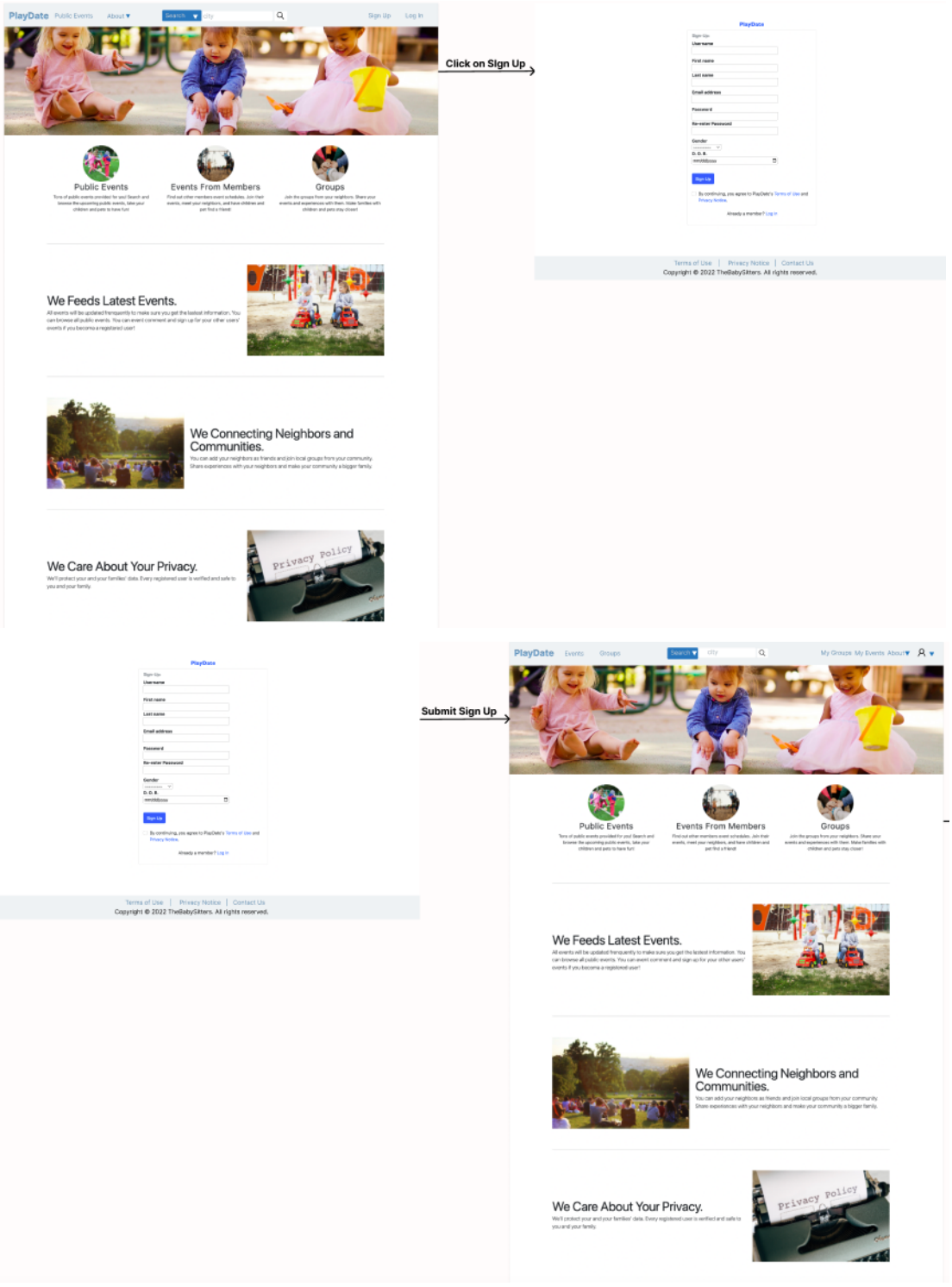
16. Emergency Contacts
    No Priority 3 Functional Requirements for 16. Emergency Request

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

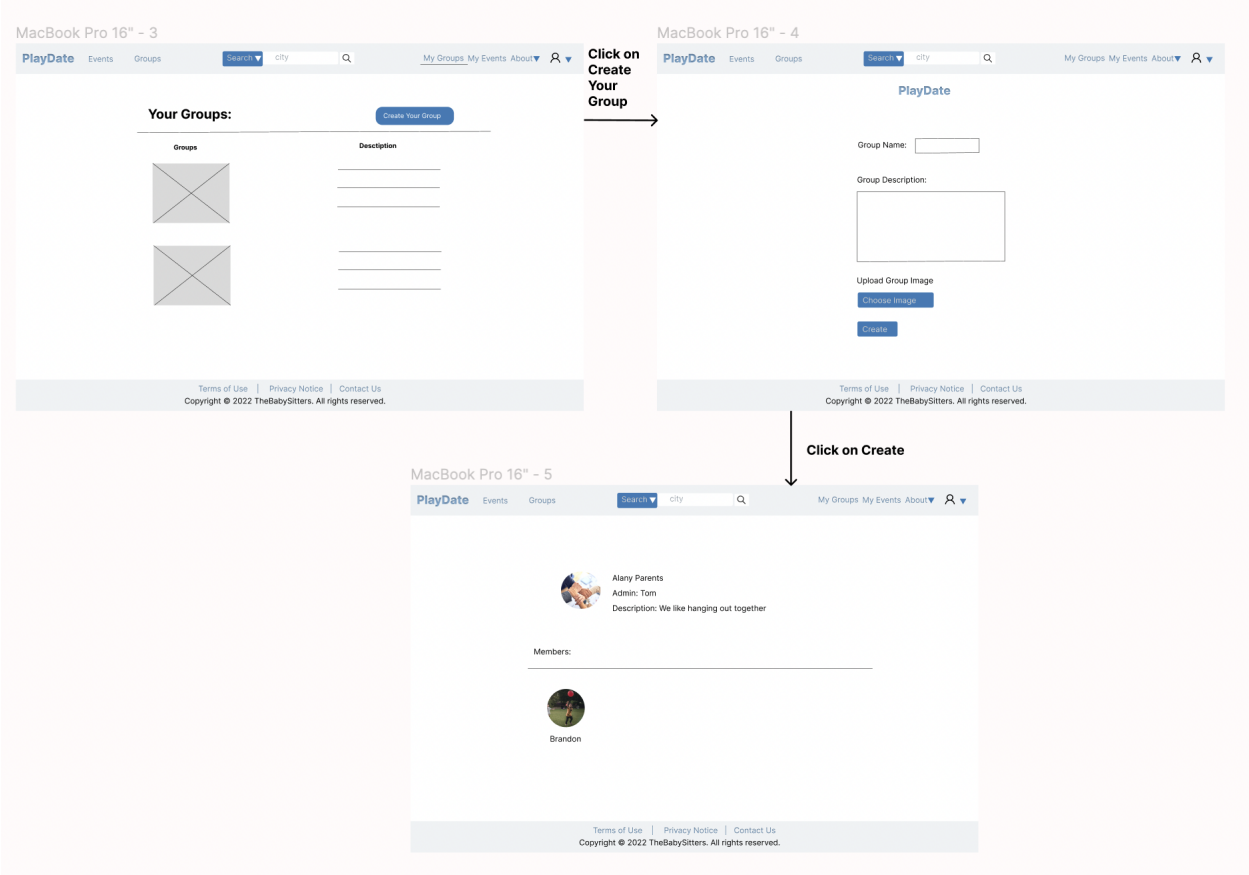
18. Posts
    No Priority 3 Functional Requirements for 18. Posts

3. Wireframes Based on Mockups/Storyboards V2

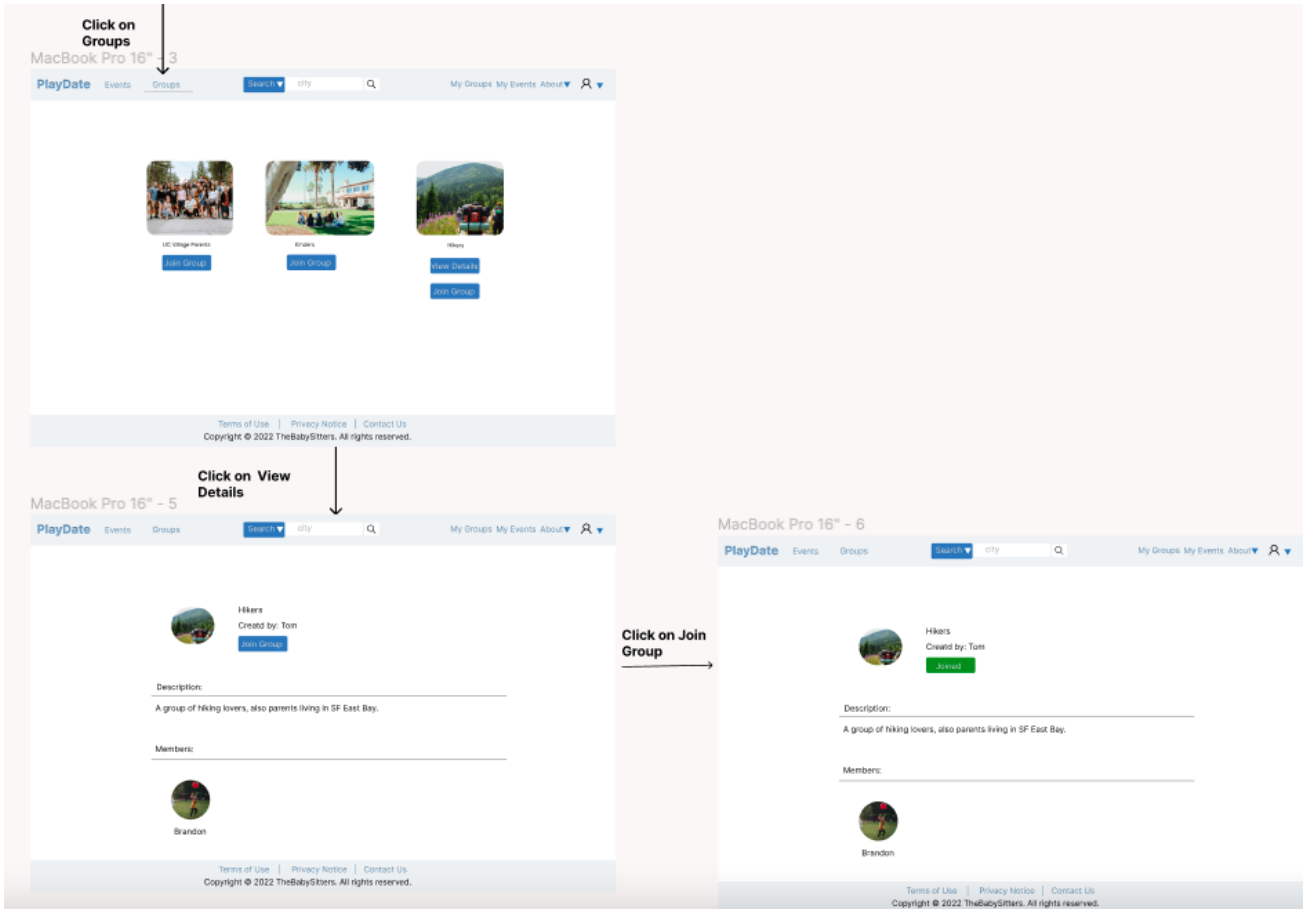
1. Use Case 1: New user registration



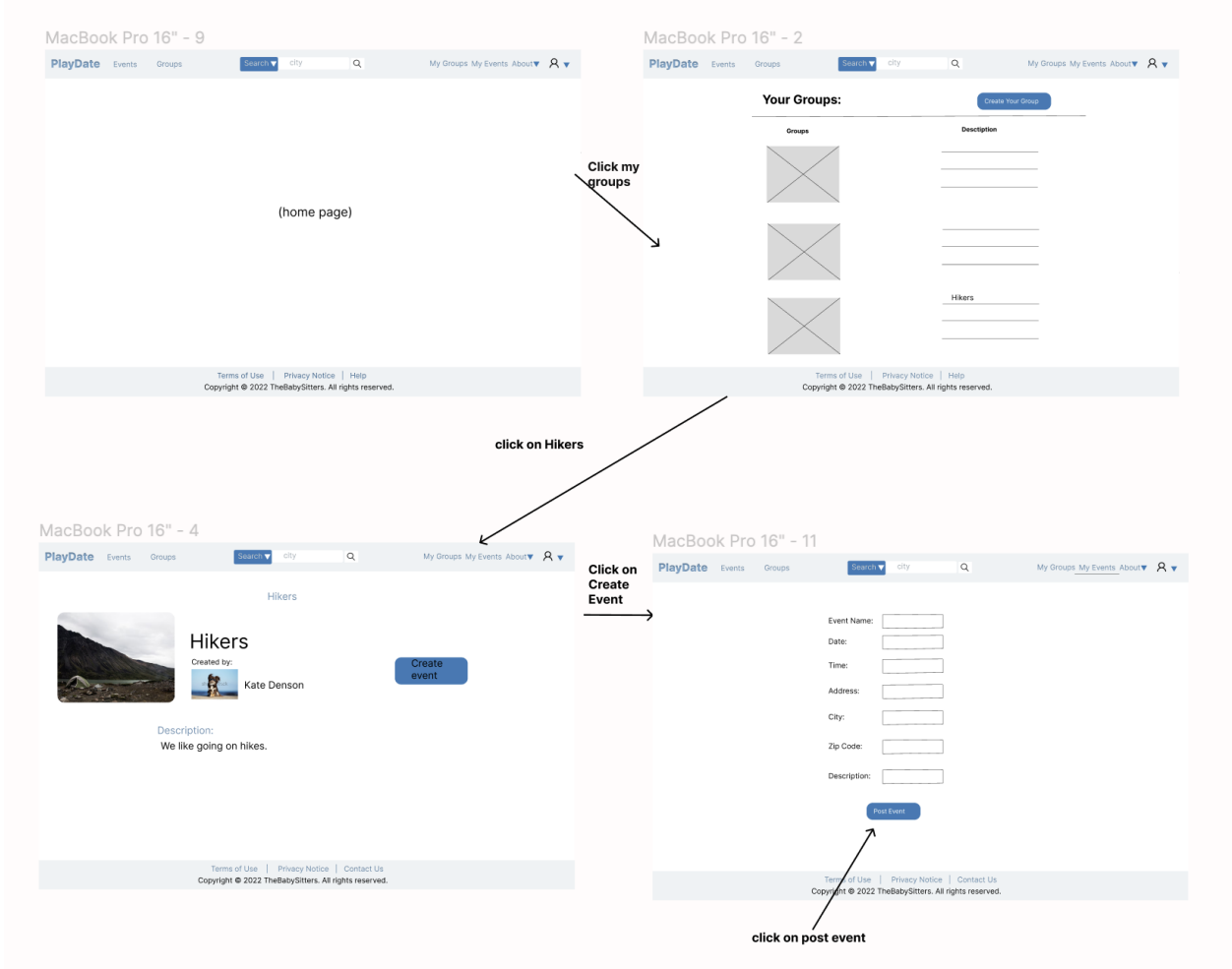
2. Use Case 2: Group Creation



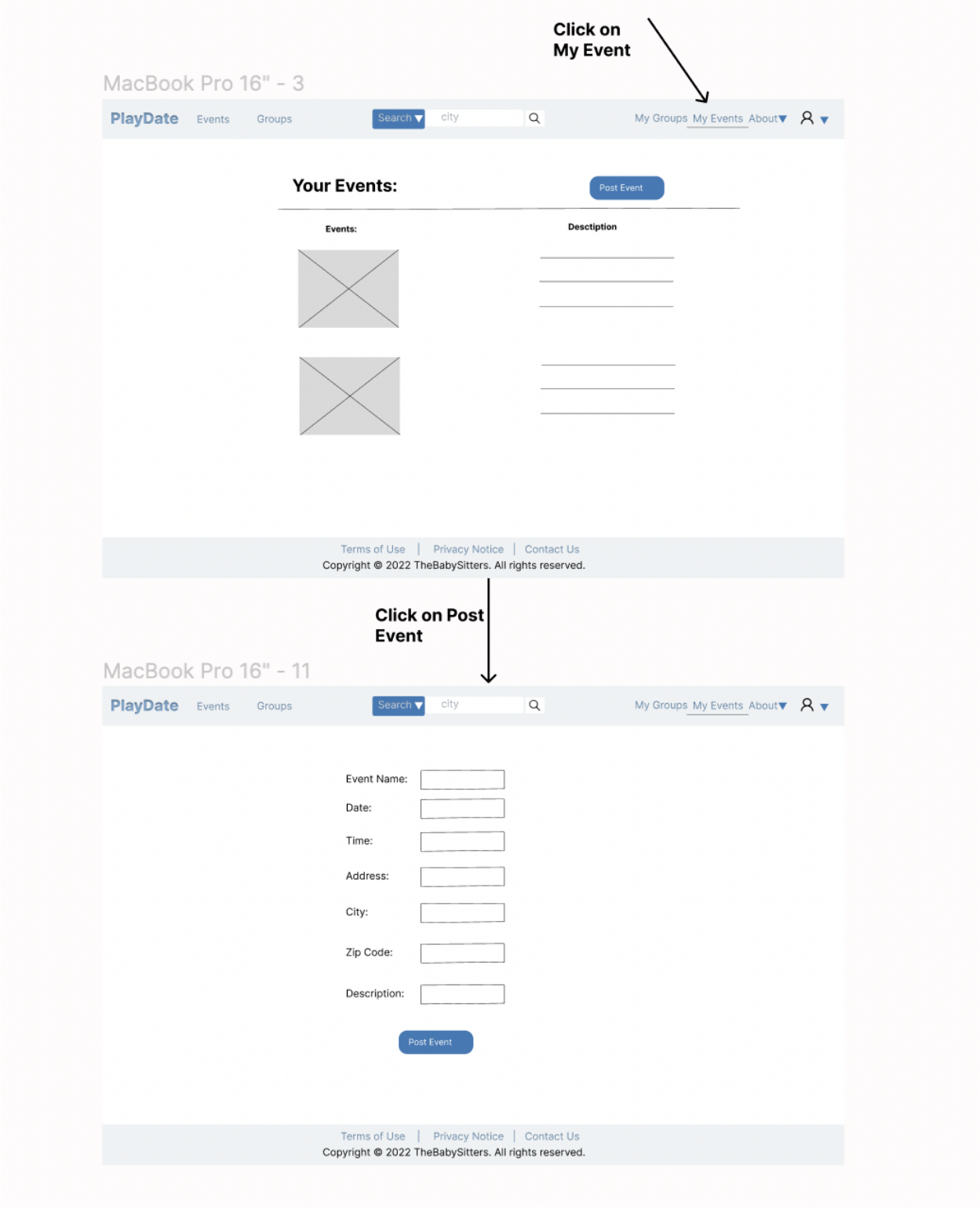
3. Use Case 3: Joining group



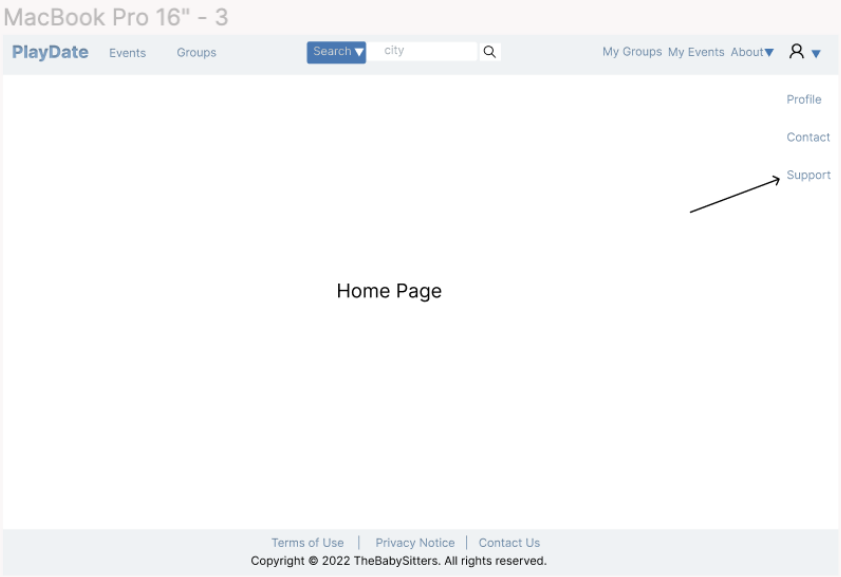
4. Use Case 4: Creating Group Events



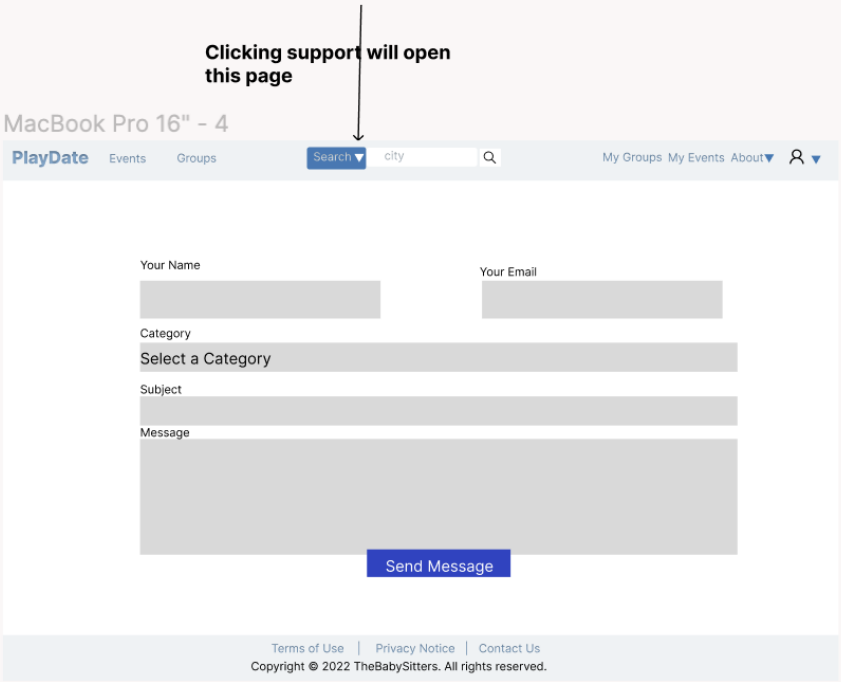
5. Use Case 5: Creating Events



6. Use Case 8: Technical Support



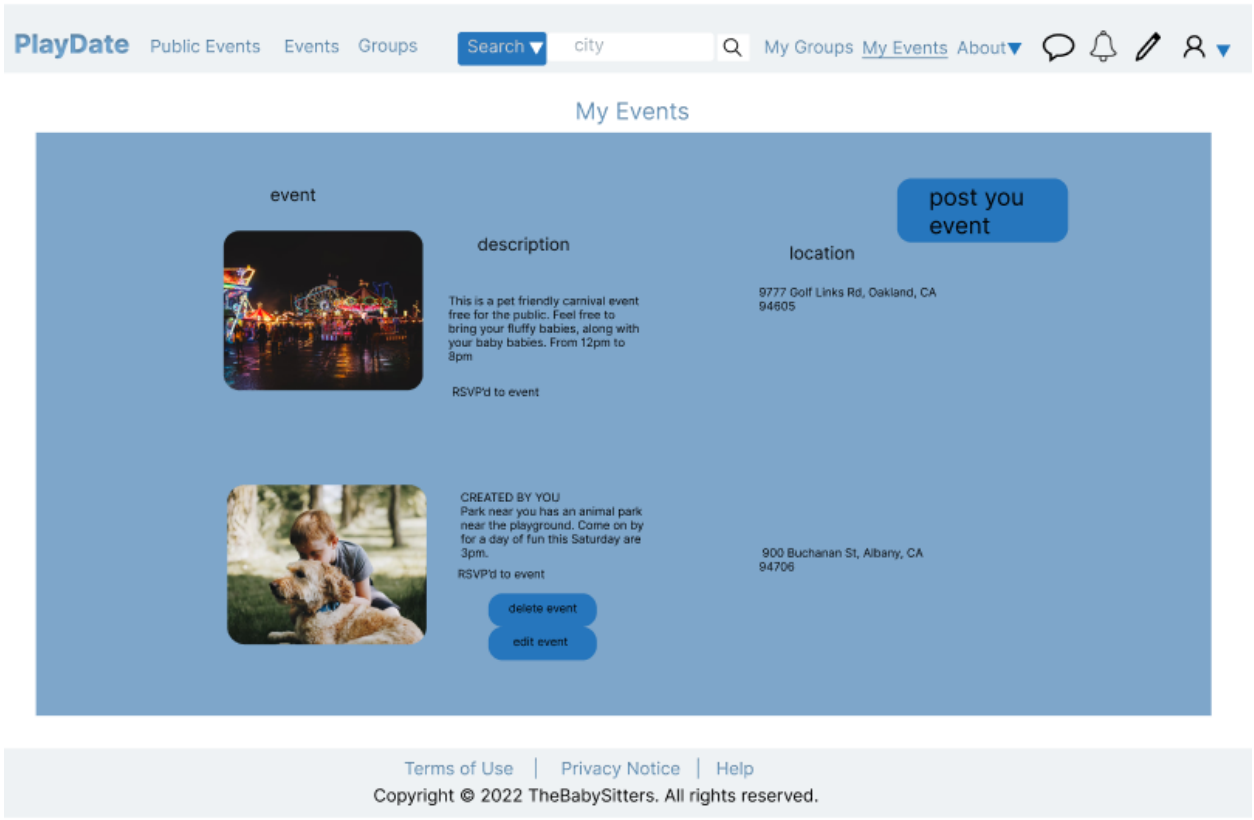
Tom clicks on drop bar menu and clicks on support tab



Tom fills out the form and hit “Send Message” when done to contact support. Support Team will get this information and will respond to email ID

7. Use Case 12: View RSVP’d event

Click on MyEvents



4. High level database architecture and organization V2

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

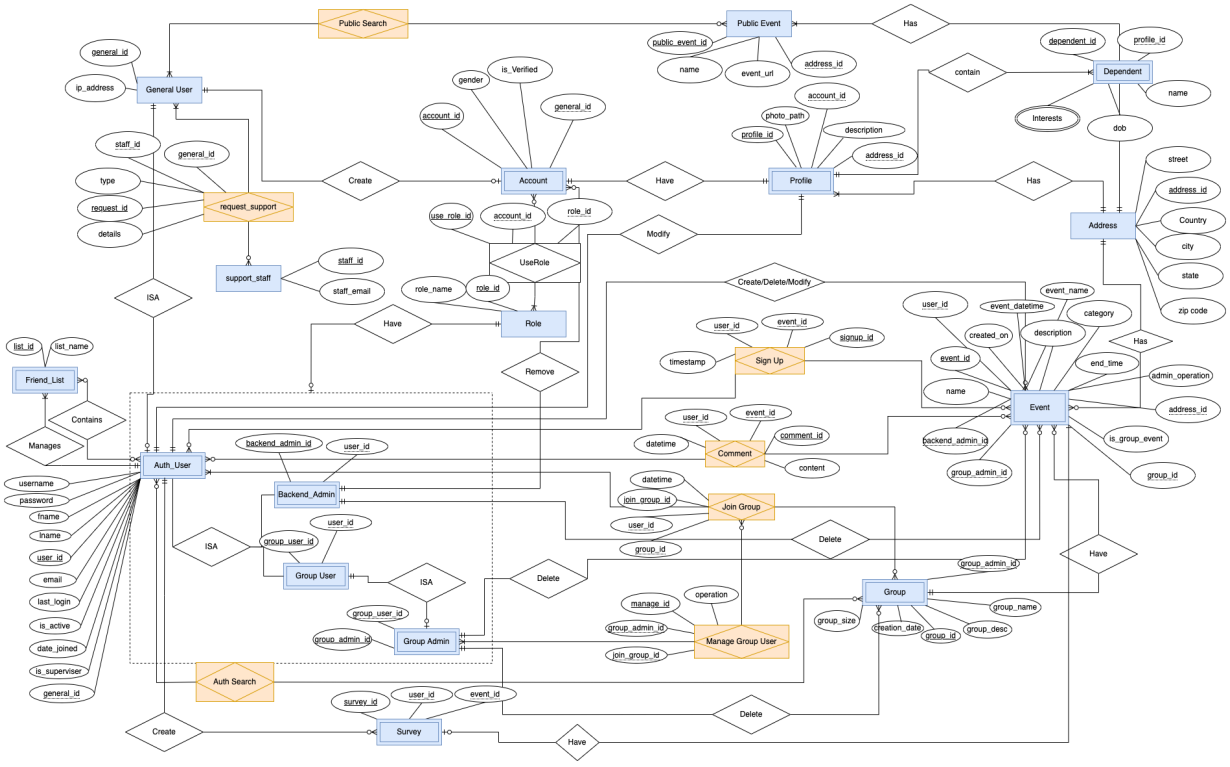
xi. Events (Weak)
* event_id: strong key, numeric
* user_id: weak key, numeric
* content: alphanumeric
* datetime: datetime

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

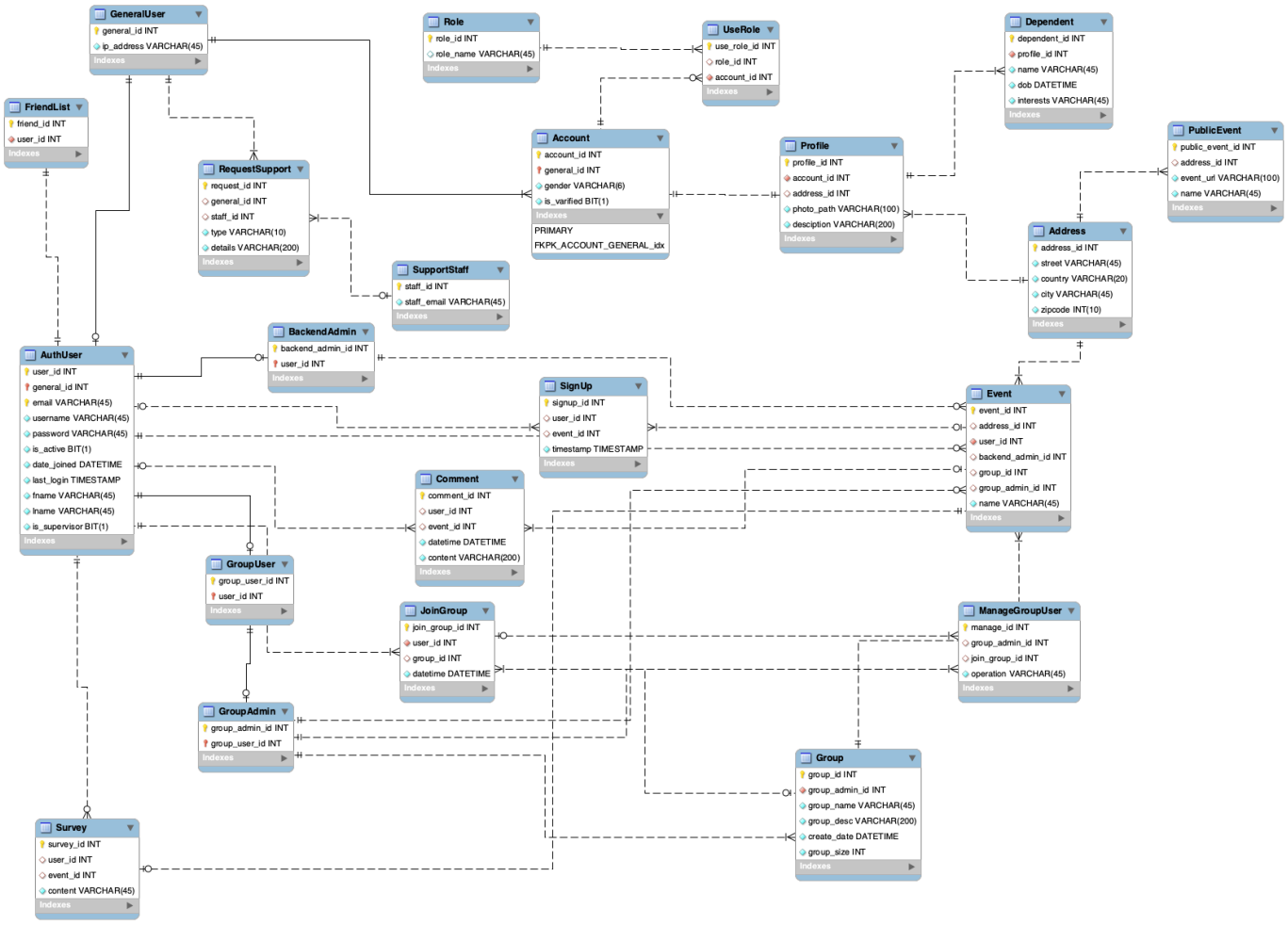
xiii. Dependents (Weak)
* dependent_id: strong key, numeric
* profile_id: weak key, numeric

* 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 will be stored in file systems. Django stores the location/path of the file in the database so that django knows where to access the images.

3. Search/Filter architecture

3.1 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 icontain, 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 it’s 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.

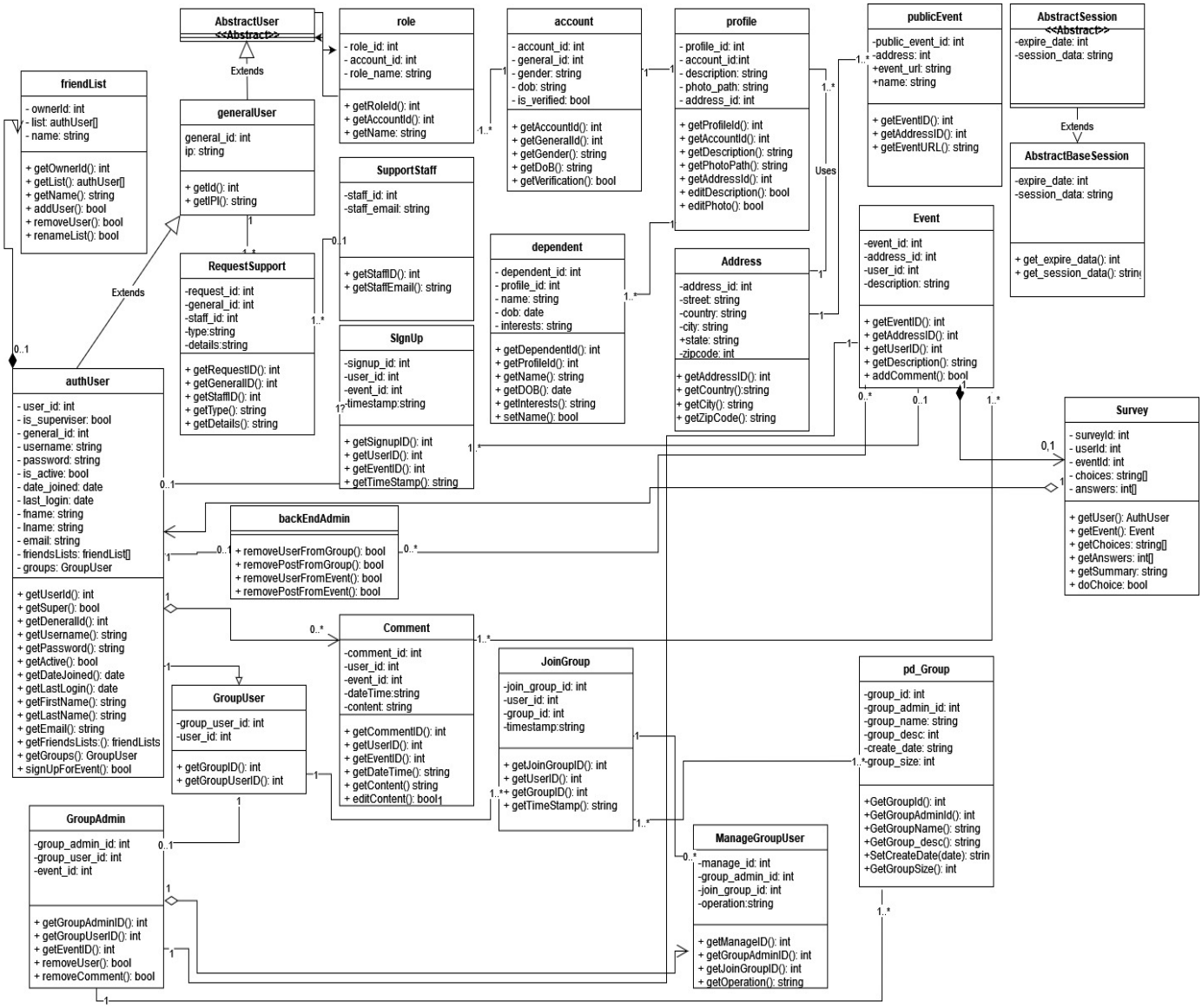
   results= Publicevent.objects.filter(lookups)

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

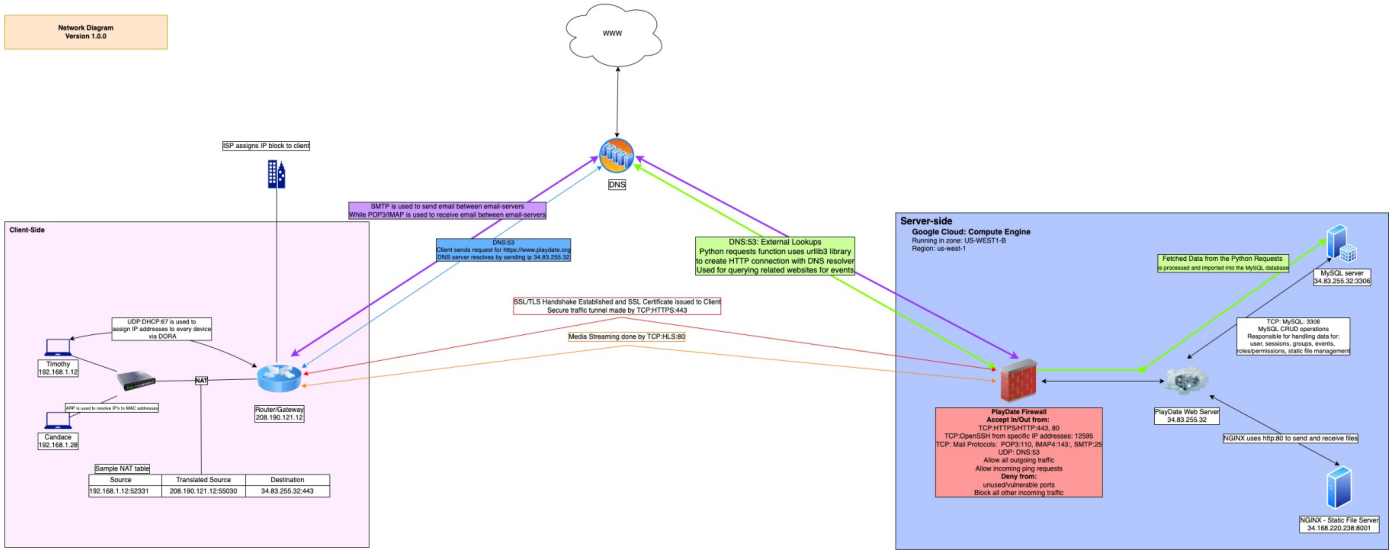
3.2 Filter categories:
The search can be further filtered for various categories, using the drop down menu which will append to query sets generated in the search algorithm.
The results from step C can be further filtered if the user selects the filter option. Here we again use the icontain to match the category of the filter.
results= Publicevent.objects.filter(lookups).filter(Q(category__icontains = 'kids'))

5. High Level Diagrams V2

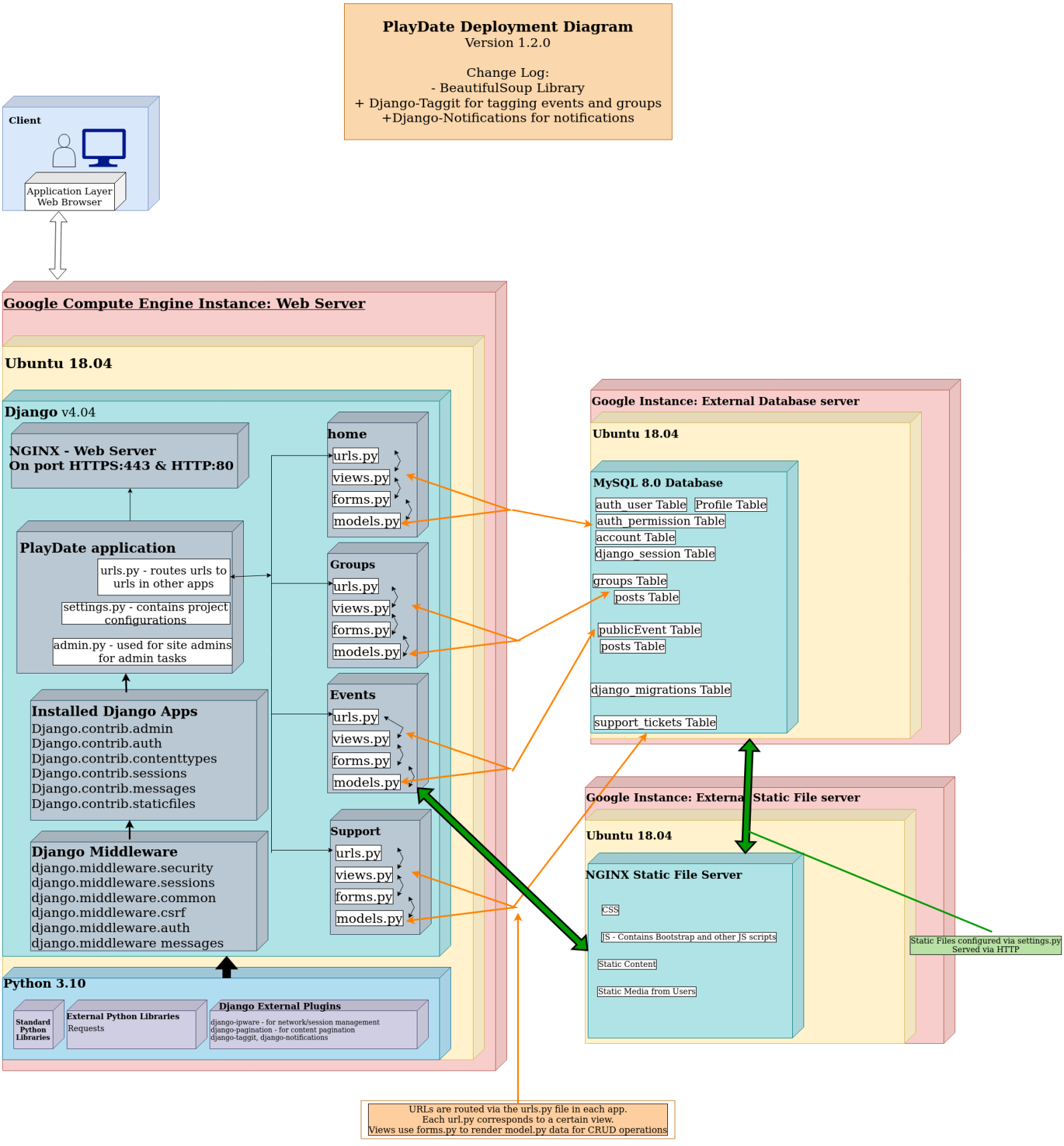
1. UML Diagram



2. Network Diagram



3. Deployment Diagram



6. List of Contributions

Name	**Role**	**Contribution**
Soujanya Ravindra Nayak	Team Lead Document Contributor Backend Team	● Organized meetings and divided tasks among team and followed up on the tasks ● Provided feedback on UI to improve them ● Implemented modifications to UI on home navbar and search ● Refined the functional requirements based on feedback from M1 & M2 and applied it to M3. ● Assisted frontend team with their git issues.
Margaret De La Torre	Front-end lead Document Contributor	● Helped brainstorm ideas implementation of UI design ● Designed wire diagrams ● Implemented UI of Group pages ● Followed up with the front end team on implementations.
Martin Salvatierra	Front-end Document Contributor	● Helped brainstorm ideas implementation of UI design ● Designed wire diagrams ● Implemented UI of User Profile page, Help Page, MyGroups Page, group search. ● Recorded and shared meeting videos. ● Noted to-do's in frontend meetings
Andy Cho	Back-end lead Document Contributor	● Provided feedback on UI ● Implemented modifications to UI for search ● Implemented backend of events and profile ● Deployed the code to server ● Updated deployment diagram
Qin Geng	Front-end Document Contributor	● Helped brainstorm ideas implementation of UI design ● Designed wire diagrams ● Implemented Home page, public events, my events, groups, my groups ● Noted to-do's in frontend meeting
William Plachno	Git Master Backend Team Document	● Refined the functional requirements based on feedback from M1 & M2 and applied it to M3.

	Contributor	● Provided feedback on UI ● Implemented modifications to Contact Support ● Updated UML diagram
Victor	Front-end Document Contributor	● Helped brainstorm ideas implementation of UI design ● Designed wire diagrams ● Implemented UI of Privacy & Terms of Use ● Implemented UI of Create event

SW Engineering CSC648-848

Summer 2022

Milestone 4

PlayDate — by Team 03 (the "Babysitters")		
Name	**Job**	**Email**
Soujanya Ravindra Nayak	Team Lead	soujanyaravindra@gmail.com
Margaret De La Torre	Frontend Lead	mdelato1@mail.sfsu.edu
Andy Cho	Backend Lead	andrewyunejo@gmail.com
Martin Salvatierra	Frontend Team	martysaljhost@gmail.com
Qin Geng	Frontend Team	gengqin50@gmail.com
William Plachno	Git Master	wjplachno@gmail.com
Victor Callejas	Frontend Team	vcallejas@mail.sfsu.edu

Date	Version
07/28/2022	M4 V1
07/28/2022	M3 V2
07/19/2022	M3V1
07/19/2022	M2V2
07/19/2022	M1V2
07/07/2022	M2V1
06/21/2022	M1V1

Table of contents

1. Product Summary

PlayDate

Social Media has become a big part of society where adults can find like-minded friends. But what about kids and pets? They, too, need someone of their kind with whom they can spend quality time. Parents try to find playdates for their children and pets, but they are always worried about child safety. There are some applications in the market that try to solve the issue of finding playdates and activities but none ensures guaranteed safety from unauthorized users. How much do we really know about people whom we invite to be our kids' playdates?

"PlayDate" is aiming to help parents solve their pain points while looking for activities and companions for their children and pets. Parents can register in PlayDate and can either join or create events for their kids and pets. All the users of PlayDate need to undergo background verification to be able to create any event or sign up for other user's events. This way PlayDate ensures the safety of your kids and pets.

Functions:

1. General users can view public events on PlayDate that are available to all users.
2. General users shall also be able to search for public events based on location.
3. General users shall be able to register on PlayDate by completing the signup page.
4. General users shall be able to request assistance from playdate support staff by filling the contact form and providing their name, email id and description of assistance.
5. General users shall be able to use chat bot to get directions on using our application.
6. Registered user shall be able to use their login credentials and login to PlayDate application.
7. Registered users shall be able to view and edit their profile where they can add dependents and update their personal information.
8. Registered users need to upload verification documents on profile to become verified users. Hence the registered user shall be able to upload photo identity as verification document on their profile.
9. Registered users shall be able to view public events.
10. Registered users shall be able to log out from their account on PlayDate.
11. Registered users who are not verified shall not be able to create events or create groups. They shall also not be able to join groups or RSVP to events created by other users.
12. Registered users shall be able to view events created by other PlayDate users under Events on PlayDate.
13. Registered user who are verified, can register to many events that they are interested in.
14. Registered users who are verified shall be able to create their own events.
15. Registered users who are verified shall be able to delete the events created by them.
16. Registered users can view all of their events that they created and RSVP'd on My Events
17. Registered users shall be able to create many groups to form small communities of similar interests and become a group admin.

18. Registered users shall be able to search the groups based on interests, location, group name.
19. Registered users shall also be able to join many groups created by other users.
20. Registered users shall be able to view all groups that they are part of, on My Groups.
21. Registered users shall be able to cancel their signup to any event
22. Each group user can create an event for their group, which will be available only to that group's users.
23. Group users can RSVP to the group events.
24. Group admin who created the group shall be able to delete any event from the group.
25. Group users shall be able to post and comment in the group to share their thoughts with their group.
26. Group users shall be able to leave the group if they are no longer interested in that group.
27. Group admin shall be able to remove any user from their group.

Uniqueness:

Unlike adults, kids need an extra safe environment. Hence the unique feature of our application lies in its business logic of user verification which is incorporated to ensure safety from scammers. Post registering on our application, users need to mandatorily upload their identity document to their profile, which will be analyzed by PlayDate backend admins to permit the user to use all the functionalities of our application. If the user does not complete the verification process they will be restricted from core functions like create and join events and groups provided by PlayDate.

Application url: http://34.83.255.32:8000/

2. Usability Test

2.1. Usability Test Plan

2.1.1. Purpose

As an application for parents of children and pets, “PlayDate” pays much attention to user’s and their families’ security. One of our superior features is “User Verification”, which will be tested in this section. “User Verification” will be tested since we want to make sure this superior feature is working well to compete with our competitors. There are 5 major functions related to this superior feature to be tested: upload identification image, sign up for events, join groups, create events, and create groups.

2.1.2. Problem Statement and Objective

- Upload Identification Image: we will test if a user can easily upload the identification photo for verification. A verified user can obtain more privileges, like sign up events and create groups, and it’s important for them to upload the identification photo successfully.
- Sign Up For Event: After the verification, we will test if a verified user can successfully sign up for events on “PlayDate”. This “sign up” function is a core privilege on our website and it’s important for users to successfully implement it.
- Join Groups: After the verification, we will test if a verified user can successfully join a group on “PlayDate”. This “join group” function is a core privilege on our website and it’s important for users to successfully implement it.
- Create Event: After the verification, we will test if a verified user can successfully create an event on “PlayDate”. This “create event” function is a core privilege on our website and it’s important for users to successfully implement it.
- Create Groups: After the verification, we will test if a verified user can successfully create a group on “PlayDate”. This “create group” function is a core privilege on our website and it’s important for users to successfully implement it.

2.1.3. User Profile

Users for the Usability test are registered users who haven’t been verified yet.

2.1.4. Method (test design)

Usability test with metrics of Effectiveness, Efficiency, and Satisfaction.

2.1.5. Task list

- Upload Identification Photo
- Sign Up For Event
- Join Groups
- Create Event
- Create Groups

2.1.6. Test Environment

Test host server website with Google Chrome Incognito on Mac.

2.1.7. Test Monitor Role

Front-end team member

2.1.8. Evaluation measures

Effectiveness & Efficiency form and Satisfaction Questionnaire

2.2. Usability Test Table for Effectiveness & Efficiency

# pages	Test Case	% completed	errors	comments	% time to complete	# steps	Amount of time
1 or 3	Upload Identification Photo	100%	No errors	The acceptable photo formats are: apng, avif, gif, jpeg, jpg, png, webp. If an unverified user tries to create an event or a group, the user will be redirected to the verification step.	0	3	30 seconds
2	Sign Up Event	95%	No errors. But after signing up for an event, the event page format changed.	Only unverified users can sign up for and view details of an event.	5	2	30 seconds
2	Join Group	95%	When unverified users joined a group, “404 error” popped up.	No errors for verified users joining a group. When unverified users join a group, it should show a page rather than 404.	5	2	30 seconds
2	Create Event	98%	No errors. But Individual event page need more styling, the time of events are all “midnight” right now	Only verified users can create an event, unverified users shall be redirected to the verification step when trying to create an event.	5	2	1 minute
2	Create Group	98%	No errors, but need more styling. Couldn’t see my group picture in the group page. “Create Group” button out of page border.	Only verified users can create a group, unverified users shall be redirected to the verification step when trying to create a group. Both groups and individual group pages need more styling.	5	2	1 minute

2.3. Task Description

Task1	Description
Task	Upload Identification Photo http://34.83.255.32:8000/profile/
Machine State	Host Server Running
Successful Completion Criteria	Identification photo uploaded and “Awaiting approval” message showed up.
Benchmark	Completed in 30 seconds

Task2	Description
Task	Sign Up Event http://34.83.255.32:8000/events/members-events/
Machine State	Host Server Running
Successful Completion Criteria	Sign up for an event and can view this event’s individual page.
Benchmark	Completed in 30 seconds

Task3	Description
Task	Join Group http://34.83.255.32:8000/groups/
Machine State	Host Server Running
Successful Completion Criteria	Join a group and can view this group’s individual page.
Benchmark	Completed in 30 seconds

Task4	Description
Task	Create Event http://34.83.255.32:8000/events/my-events/
Machine State	Host Server Running
Successful Completion Criteria	Create an event and can view this event's individual page.
Benchmark	Completed in 1 minute

Task5	Description
Task	Create Group http://34.83.255.32:8000/groups/myGroup/
Machine State	Host Server Running
Successful Completion Criteria	Create a group, can view this group's individual page, and is the admin of this group.
Benchmark	Completed in 1 minute

2.4.User Satisfaction

	Strongly disagree				Strongly agree
1. The port for uploading the identification photo is easy to find.	1	2	3	4	5
2. You can upload different formats of images.	1	2	3	4	5
3. The photo uploading process is easy to implement.	1	2	3	4	5
4. You can review info of all events.	1	2	3	4	5
5. It's easy to sign up for an event.	1	2	3	4	5
6. You can easily get the info of your signed up event.	1	2	3	4	5
7. You can review info of all groups.	1	2	3	4	5
8. It's easy to join a group.	1	2	3	4	5
9. You can easily get the info of groups you joined.	1	2	3	4	5
10. It's easy to create an event.	1	2	3	4	5
11. You can enter details about the event you want to create.	1	2	3	4	5
12. You can see your event on our website after you created it.	1	2	3	4	5
13. It's easy to create a group.	1	2	3	4	5
14. You can enter details about the group you want to create.	1	2	3	4	5
15. You can see your group on our website after you created it.	1	2	3	4	5

16. You are satisfied with the look and feel of the user interface that you used.	Strongly disagree				Strongly agree
	1	2	3	4	5
17. Comments or Advices:					

Average Agreement on User Satisfaction Survey
Survey repository:
https://drive.google.com/file/d/1iH16FyUmyE06jC6T-B_r-9TRNeol26e0/view?usp=sharing

Average Agreement Table:

Statement number	Average User Response
1	4
2	4
3	5
4	4
5	4
6	4
7	3
8	4
9	3
10	4
11	4
12	3
13	4
14	4
15	4
16	4

3. QA Test Plan

HW and SW setup:

Testing hardware 1 - MAcBook Pro 2019

Software 1 - MacOs Monterey Version 12.4

Testing hardware 2 - MacBook Air 2017

Software 2 - macOS Monterey Version 12.3.1

App link: http://34.83.255.32:8000/

TEST 1: Usability

2.1 Users shall receive online help from support for any assistance on the application.

Test Plan Outline:

This test will be performed on the home page where the chat bot is located on the bottom right. The test environment for 2.1 QA test is performed on Chrome and Safari. The action should take a minute or two to complete, this includes the typing and submission process. The risks that the test may run into is if the user submits repeated text or if the automated message lags to reply or simply does not work.

Test Number	Test Title	Test Description	Test input	Expected correct output	Pass/Fail
1	Test typical	Input a typical expected user input.	“hi”	Thanks for the message. Our team is offline. 💤 We will contact you back as soon as we can. 🙂	Pass
2	Test symbols	Test is non typical user input such as numbers and symbols.	Enter, “1!@$@2^2^3433&#%&”, into the input field for the chat help bot.	Auto reply saying something like “Thanks for the message. Our team is offline. 💤 We will contact you back as soon as we can. 🙂”	Fail
3	Test alt language	Test using other languages in input..	Enter, “私は助けが必要です”, into the input field for the chat help bot.	Auto reply saying something or auto message saying that they will reply when they can.	Pass

TEST 2: Security

4.2 - Information should be securely transmitted to the database server without any changes in information.

Test objectives:

This test is to make sure that user input is correctly sent properly to the database and we also want to make sure that password input is type in securely. This test should take about one minute to perform input and about another minute to check the database. A risk of this test is if the test fails we might have to fix data in the backend if incorrect tables are affected.

Test Number	Test Title	Test Description	Test input	Expected correct output	Pass/Fail
1	Password test	Test to see if a password is always hidden when entered.	Enter, "password123", into the password text box when creating an account.	"••••••••" SHA encrypted password. Database has an encrypted password.	Pass
2	Email test	Test to see if email was correctly sent to the database under auth_user table.	Enter "mail@email.com", in the email form when creating an account	In the database "mail@email.com ", is correctly placed into the email section on the auth_user table.	Pass
3	User name test	Test to see if the user name was correctly sent to the data under auth_user table.	Enter "MargaretDLT" under "user name" when creating an account.	In the database "MargaretDLT" was correctly placed under the "username" section of the auth_user table.	Pass

TEST 3: Compatibility
5.1 - Application should be supported on Mac via browsers of versions, Chrome >= 60, Safari >= 12

Test objectives:
The test requires that a standard input is inserted in the navigation search bar for the same results. The search result should work on compatible computers with chrome,firefox, and Safari. The test should take about a minute to verify that results are present for what is being looked for. The risks are that the user users an unsupported browser that fails to render a search.

Test Number	Test Title	Test Description	Test input	Expected correct output	Pass/Fail
1	Chrome	Test if app looks good and search function works well on Chrome.	Enter, “San Francisco”, into the search bar.	Shows propers list of events	Pass
2	Safari	Test if app looks good and search function works well on Safari	Enter, “ ”, into the search bar.	Display a message saying “No results found please try again”	Fail
3	Chrome	Test if app looks good and search function works well on Firefox	Enter, “ ”, into the search bar.	Display a message saying “No results found please try again”	Fail

TEST 4: Data Storage

7.2 - The application's back-end servers should only be accessible to authenticated backend admins.

Test objectives:

The test is to make sure that user input cannot affect the backend database. If a user were to input text that would be similar to backend code, then the backend should remain unaffected. This test should take about a minute to perform the input and then another minute to check the backend. The risks are that the backend might be affected by user input during this test.

Test Number	Test Title	Test Description	Test input	Expected correct output	Pass/Fail
1	Username login with SQL injection	User inputs text similar to a query in username form for login.	"5 OR 1=1" which corresponds to SELECT * FROM Users WHERE UserId = 105 OR 1=1;	User shouldn't be logged in	Pass
2	Username create	User inputs text similar to a query as their username when creating an account.	CREATE DATABASE injectionTest2;	"The username must consist of letters, digits, +, -, .,@,and _." , nothing was created in the database	Pass
3	Create event description	User inputs text similar to a query as their created event description.	CREATE DATABASE injectionTest3;	Uses "CREATE DATABASE injectionTest3;" as description and does not affect the database	Pass

TEST 5: Data Storage
7.1. The application's back-end servers should never display a customer's password.

Test objectives:
The test requires a new user to register an account using different sizes and characters. The password is encrypted by Django using the sha_256 algorithm. The action should take a minute or two to complete, this includes the typing and submission process. The risks that the test may run into are if the user submits more than what we tried of 170 characters and therefore might not be able to work. Another risk would be if the test fails we might be able to see other user passwords.

Test Number	Test Title	Test Description	Test input	Expected correct output	Pass/Fail
1	user1	Test a password that matches requirements.	"Password321!"	An encrypted password in the database on the "auth_user" table.	Pass
2	user2	Test a password that is really long.	"123456789qwertyuiopasdfghjklzxcvbn123456789qwertyuiopasdfghjklzxcvbn123456789qwertyuiopasdfghjklzxcvbn123456789qwertyuiopasdfghjklzxcvbn123456789qwertyuiopasdfghjklzxcvbn"	An encrypted password in the database on the "auth_user" table.	Pass
3	user3	Test a password with a bunch of characters	"aglas#$^#$^#@$"	An encrypted password in the database on the "auth_user" table.	Pass

4. Code Review

1. Coding Style

- Indentation: We are using Python and it uses indentation to indicate control structures, so correct indentation is required. By doing this, the need for bracketing with curly braces is eliminated.
- Header comments: Having an introductory header comment is very important. Django organizes into 'applications'. Our project has the home app, the events app, and the groups app, each of which have a set of standardized files which share names. Having a comment at the top is the perfect place to mention which file you are looking at. It is also good practice to provide a sense of the purpose of that file and what the big ideas are that have an effect on the organization of the file itself.
- Camel case: We use camel case styling for definition and variable names.
- Code management:
    - Our project is divided into multiple applications like: playdate, home, events, groups for structured management.
    - Each application has templates and views.
    - All the frontend code is stored in the templates folder of each application
    - All the backend logic is written in views.py for each application.
    - All the database classes are defined in models.py of respective applications which can be imported into other applications when needed.
    - All the forms are defined in forms.py
    - Navigation tabs, footers, and other reusable html snippets that are common to multiple pages are stored globally in the playdate application's templates folder and are included in other applications, without the need to re-write the headers and footers.
    - All the css, javascript, and media files are stored in static folders for access from all html files.

2. Peer review

Repository for Code Review

https://github.com/andrewyjo/CSC648-Team03-CodeReview

Peer review by other team

Team 01 Code Review of Team 03

Part 1

home/views.py:

- Excellent header providing a short but detailed description of the particular file
- Good organization of imports

- Plenty of informative inline comments
- Consider separating parts of the code with white space rather than simply comments to make it less crowded and easier on the eyes to read
- Good use of standard naming conventions

home/HTML/:

- Good headers
- Consider having minimal inline style. Inline style avoids content and design from being separated. Use CSS files instead. For ex: everything related to font, color, align, etc should be in CSS file not inline, makes the code very crowded
- Consider separating divs with white space to make it less crowded and easier on the eyes to read
- Good use of standard naming conventions
- Very few informative comments
- Consider adding <footer> to a separate view
- Consider putting functions to a separate file

Part 2

events/views.py:

- Excellent header providing a short but detailed description of the particular file
- Good organization of imports
- Utilizes white space well, seems less packed, and is simpler to read.
- Plenty of informative inline comments
- Good use of standard naming conventions

Peer review by other team member

Part 1:

Home/html

- Would want footer as a separate file for better organization.
- Easy to understand and follow

Home/views

- Header, helpful to know which page we are in to edit
- Imports organized and easy to find
- Informative comments
- You would want to reduce the number of if statements. Make it more modular by adding functions. Want to make it easier to debug and modify.

Part2:

events/views

Organization:

- Header, helpful to know which page you are in for editing.
- Imports organized
- Inline comments informative
- Probably same as previous feedback, to reduce number of if statements if possible and make it modular
- Email logic as a separate function for contact support is an idea if possible

5. Self-check on best practices for Security

5.1. List of major assets we are protecting

- Passwords: Passwords are encrypted before storing them on the database.
- Database:
    - Input data validation: Text inputs are protected against SQL injections using the ORM model of Django by validating them.
    - Image validation: Image uploads are validated for proper file type to avoid user uploading harmful files.
- User identity:
    - Users have unique login credentials to login into the application.
    - Each user is assigned a role that differentiates backend admins from users of the application to avoid superior privileges to users of the application.

5.2 Password Encryption:

In Django we use the PBKDF2 algorithm with a SHA256 hash, a password stretching mechanism recommended by NIST.

Password encryption algorithm:

In django we use make_password to encrypt the password.

```
    user.password = make_password(password)
    user.save(using=self._db)
    return user
```

It validates if the password field is empty then returns an unusable string.
If the password is not abiding by password requirements, it raises an error.
If the password is valid then it calls PBKDF2PasswordHasher. This definition uses the pbkdf2_sha256 algorithm which executes 320000 iterations to generate a hash of the password.

```
def make_password(password, salt=None, hasher="default"):
    if password is None:
        return UNUSABLE_PASSWORD_PREFIX + get_random_string(
            UNUSABLE_PASSWORD_SUFFIX_LENGTH
        )
    if not isinstance(password, (bytes, str)):
        raise TypeError(
            "Password must be a string or bytes, got %s." % type(password).__qualname__
        )
    hasher = get_hasher(hasher)
    salt = salt or hasher.salt()
```

```
return hasher.encode(password, salt)
```

In database table password is stored as below:

	id	password	last_login	is_superuser	username
▶	1	pbkdf2_sha256$260000$C02XC61bvVa8nMwK…	2022-07-28 09:03:04.723318	0	andy
	2	pbkdf2_sha256$260000$Fzday2kSLylZNLrLBE…	2022-07-28 13:57:04.849691	1	playdateadmin
	3	pbkdf2_sha256$260000$R47T7JRzA1aTKG3qr…	2022-07-28 04:38:22.178469	0	soujanya
	4	pbkdf2_sha256$260000$v1WgRmy9RpWcng8…	2022-07-28 03:54:42.355237	0	andy2
	5	pbkdf2_sha256$260000$qf5IRHk1l4ekUurwdYc…	2022-07-28 16:23:08.033585	0	wjplachno
	6	pbkdf2_sha256$260000$zh1l8JD8nh5BbIGu6C…	2022-07-28 04:53:08.174907	0	anna45
	7	pbkdf2_sha256$260000$PYNjslEVzS3KoevNV…	2022-07-28 05:03:05.273630	0	sana4536
	8	pbkdf2_sha256$260000$2RutipNnuZTTwEHQ…	2022-07-28 14:06:20.267441	0	qin6
	9	pbkdf2_sha256$260000$apDaUqSgxqmUi7Wm…	2022-07-28 13:58:19.063756	0	qin7
	NULL	NULL	NULL	NULL	NULL

5.3 Input data validation:

SQL Injections are a big security hazard. When you read in input from the client and try to stick it in the database, a nefarious user can set variables so that, instead of just the variables going into the database getting modified, the SQL query itself gets modified. This can lead to automatic password verification and data security breaches.

Thankfully, the Django framework was designed with SQL Injection protection in mind. Django provides the Object Relational Model layer with functionality for both defining the database as well as making database calls secure. Whenever a sql query gets run by the ORM, it double checks all variables for extra SQL. If it finds it, it removes the sql from the variable and only uses the portion not involved in the SQL attack.

While there are ways to side-step the ORM, our application never does. As such, we will be protected from SQL injection consistently.

We also protect ourselves by validating images that get uploaded. We require that all images uploaded to our server be less than 6.5MB, and that they are of a particular format. We allow images of .apng, .avif, .gif, .jpeg, .jpg, .png, and .webp format. We do these checks both client-side (using javascript inside of playdate/templates/imageUpload.html) as well as server-sde using custom validators on forms.

5.4 Search validation

Steps taken to validate search:

1. Firstly the input is validated for empty input. If input is empty search is not performed

```
{% if submitbutton == 'Search' and request.GET.q != '' %}
```

```
{% if results %}

<div class="result-body">
```

2. Search keywords are used in the query. We use the Django ORM model to make queries to the database. ORM removes SQL parts from the user input before checking for results from MySQL.

```
if request.method == 'GET':
    query = request.GET.get('q')
    filter = request.GET.get('category')

    submitbutton = request.GET.get('submit')
    print(query)

    if query is not None:
        if filter == 'All':
            # query databse to check if matching city, zipcode, or street

            lookups = Q(address__city__icontains=query) | Q(address__zipcode__icontains=query) | Q(
                address__country__icontains=query) | Q(address__street__icontains=query)

            results = Publicevent.objects.filter(lookups)

        elif filter == 'Kids':
            # query databse to check if matching city, zipcode, or street

            lookups = Q(address__city__icontains=query) | Q(address__zipcode__icontains=query) | Q(
                address__country__icontains=query) | Q(address__street__icontains=query)

            results = Publicevent.objects.filter(
                lookups).filter(Q(category__icontains='kids'))
            print(filter)
        else:
            lookups = Q(address__city__icontains=query) | Q(address__zipcode__icontains=query) | Q(
                address__country__icontains=query) | Q(address__street__icontains=query)

            results = Publicevent.objects.filter(
                lookups).filter(Q(category__icontains='pets'))
            print(filter)

        context = {'results': results,
```

```
                'submitbutton': submitbutton}
        return render(request, 'events/events.html', context)
```

3. If the search keywords do not match any results from the database, then we return no results found.

```
{% else %}
  <!-- input not found -->
  <div class="grid-container">
    <div class="gal-detail thumb" style="font-size: 130%;">
      <Header>No results found please try again</Header>
    </div>
  </div>
```

6. Self-check: Adherence to original Non-functional Requirements

1. Privacy:
    1.1. Every general user shall sign PlayDate Terms of Service and Privacy when they register. - Done
    1.2. Group admin shall sign an agreement to be responsible for appropriate data sharing in groups. - On Track
    1.3. Users violating PlayDate Terms of Service should be warned for the first time and removed for the second time. - Issue, this non functional requirement is dependent on the functional requirement 2.50 which is part of priority 2.

2. Usability:
    2.1. Users shall receive online help from support for any assistance on the application. - Done
3. Response time:
    3.1. General User shall receive an update on his/her background verification in less than 24 hours of registering on PlayDate. - Done
    3.2. Support staff shall respond to users within 12 hours of raising request. - Issue, implemented by the business and not in the application.
    3.3. Support staff shall send a notification to the backend admin within 30 minutes of receiving reports on inappropriate posts or technical issues. - Issue, implemented by the business and not in the application.
4. Security:
    4.1. PlayDate org should do background-checking when a general user registers on PlayDate.- Done
    4.2. Information should be securely transmitted to the database server without any changes in information. - Done
5. Compatibility:
    5.1. Application should be supported on Mac via browsers of versions, Chrome >= 60, Safari >= 12 - Done
    5.2. Application shall be supported on Windows via browsers of versions, Chrome >= 60 - Done
6. Business need
    6.1. Application should use googleapis to find nearby places to visit. - Issue, it is dependent on a functional requirement which is not implemented in priority 1.
7. Data Storage
    7.1. The application’s back-end servers should never display a customer’s Password. - Done
    7.2. The application’s back-end servers should only be accessible to authenticated backend admins. - Done
    7.3. User data should be deleted from the backend servers if the user deletes their account. - On Track
8. Legal, marketing, copyright
    8.1. Application should display the disclaimers, copyright of the advertisers

while advertising on the application. - Issue, currently there are no advertisements on the application.
8.2. Application should use only open source images in the web application. - Done

7. List of Contributions

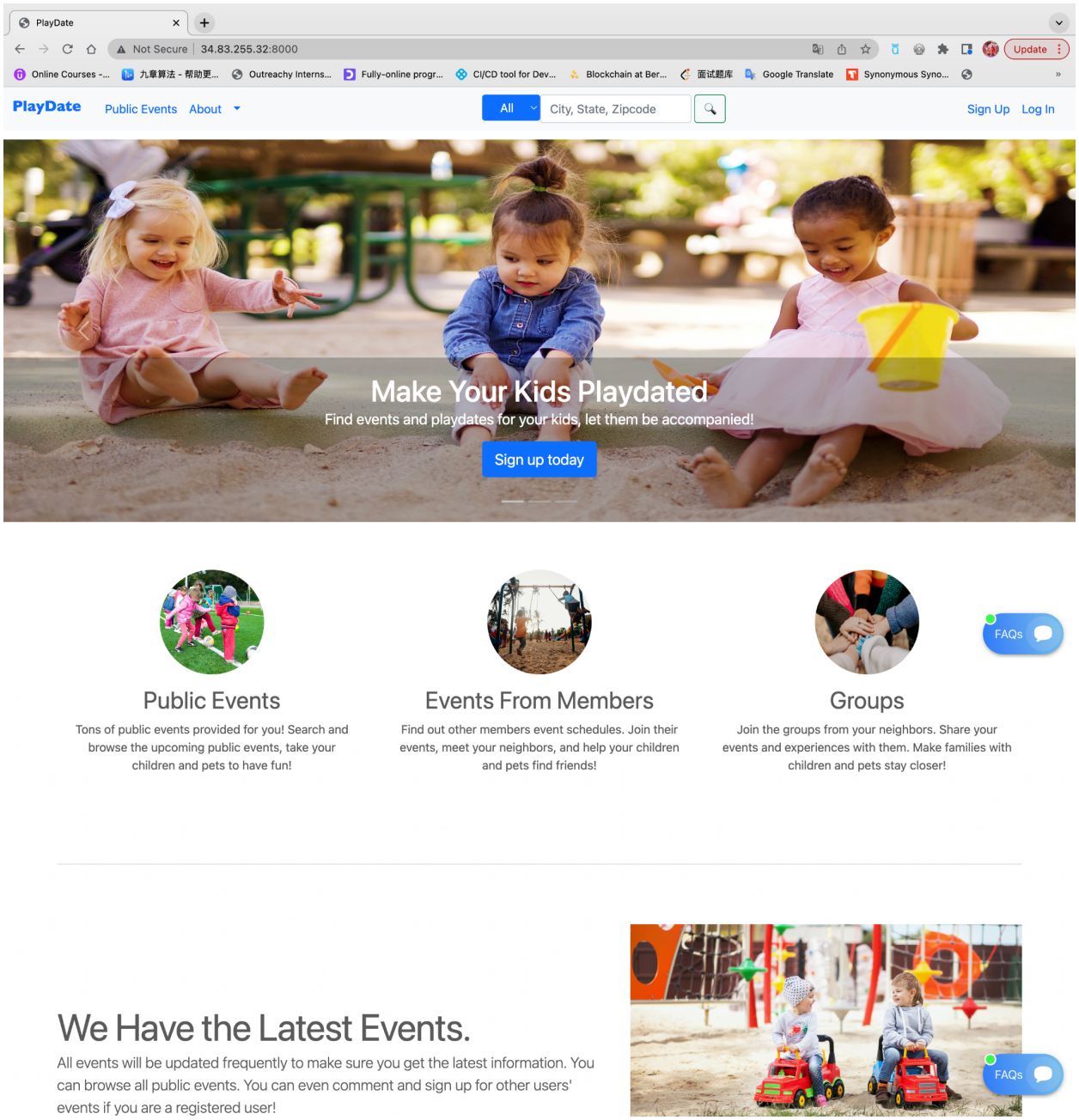
Name	**Role**	**Contribution**
Soujanya Ravindra Nayak	Team Lead Document Contributor Backend Team	● Organized meetings and divided tasks among team and followed up on the tasks ● Revised M3 V2 with feedback on M3 V1 ● Contributed to summary and QA testing ● Implemented backend of events for our superior feature ● Collected user survey
Margaret De La Torre	Front-end lead Document Contributor	● Performed QA testing ● Implemented frontend based on feedback of M3 ● Collected user survey
Martin Salvatierra	Front-end Document Contributor	● Performed QA testing ● Implemented frontend based on feedback of M3 ● Collected user survey
Andy Cho	Back-end lead Document Contributor	● Contributed to code review ● Implemented backend of groups for our superior feature
Qin Geng	Front-end Document Contributor	● Performed Usability testing ● Implemented frontend based on feedback of M3 ● Collected user survey
William Plachno	Git Master Backend Team Document Contributor	● Contributed to QA testing ● Contributed to Self check on security ● Implemented backend of use verification for our superior feature ● Contribute to code review ● Collected user survey
Victor	Front-end Document Contributor	● Performed usability testing ● Contributed to code review ● Implemented frontend based on feedback of M3

3. Screenshots of Final Product

Our final product can be divided into three parts based on the users identification:
general user (non-registered user), **registered user (but non-verified)**, **verified users**

3.1. As a general user (non-registered user), you can get reach to following pages:

3.1.1 Home Page:



We Connect Neighbors and Communities.

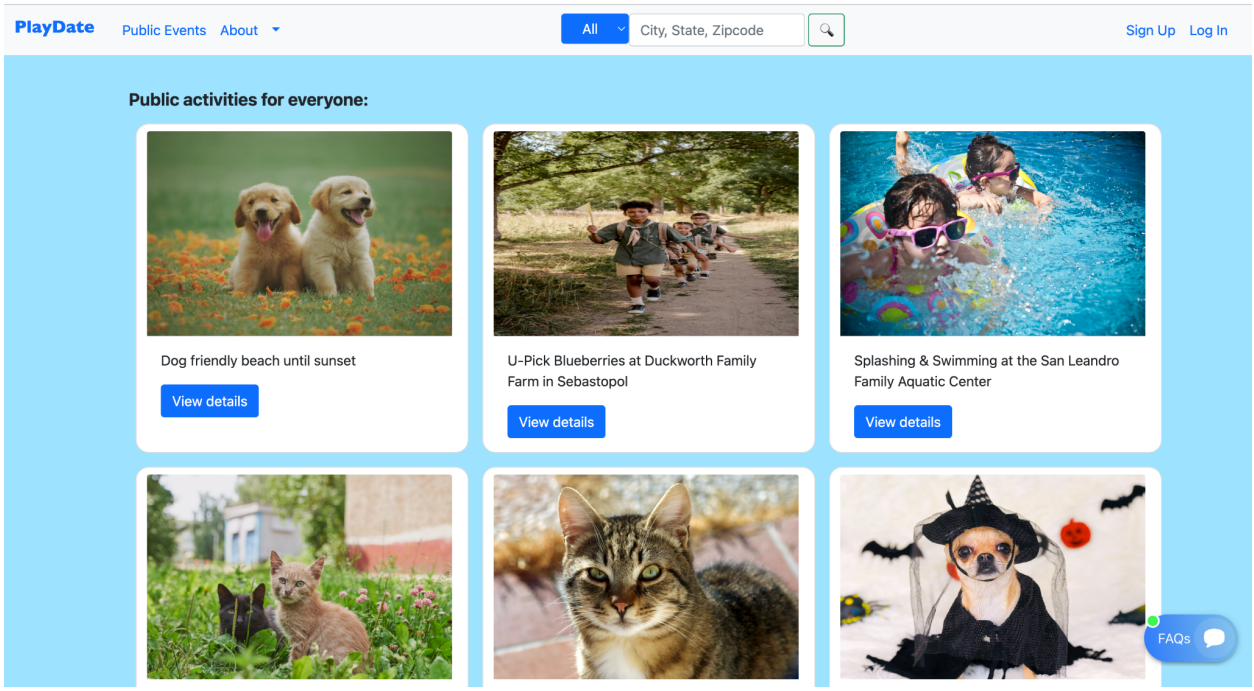
You can add your neighbors as friends and join local groups from your community. Share experiences with your neighbors and make your community a bigger family.

We Protect Your Data.

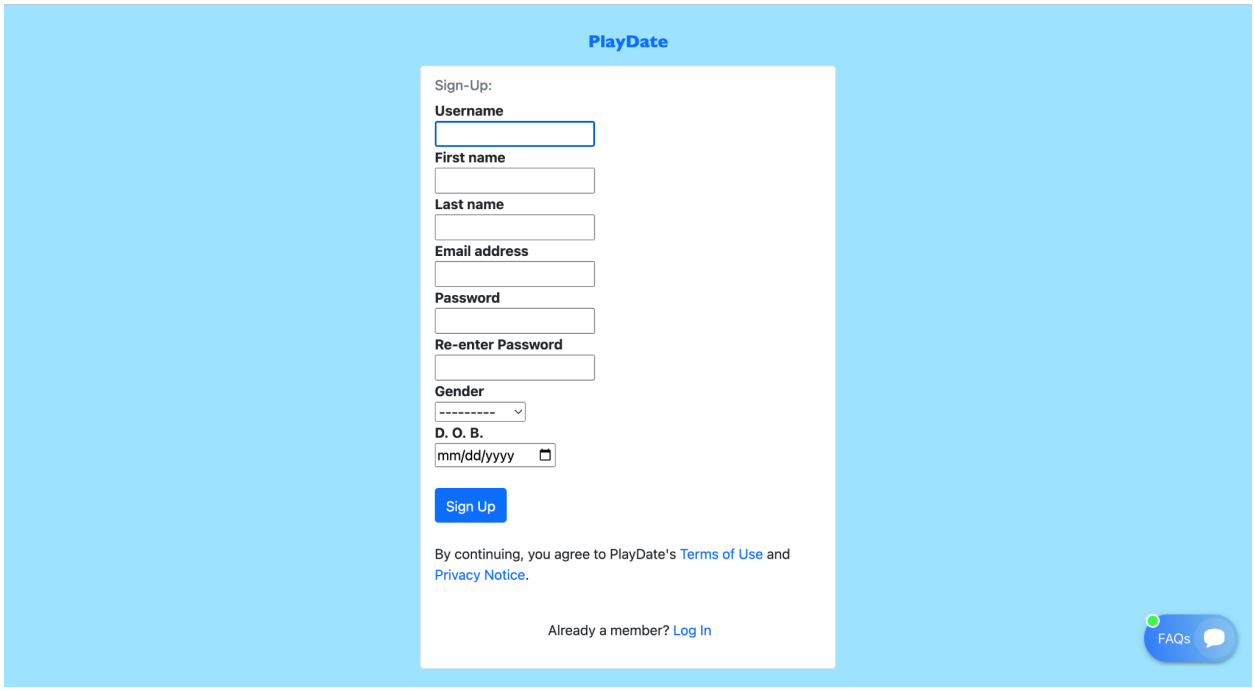
We care about your and your family's privacy and will protect your data. Every registered user is verified and safe to you and your family.

FAQs

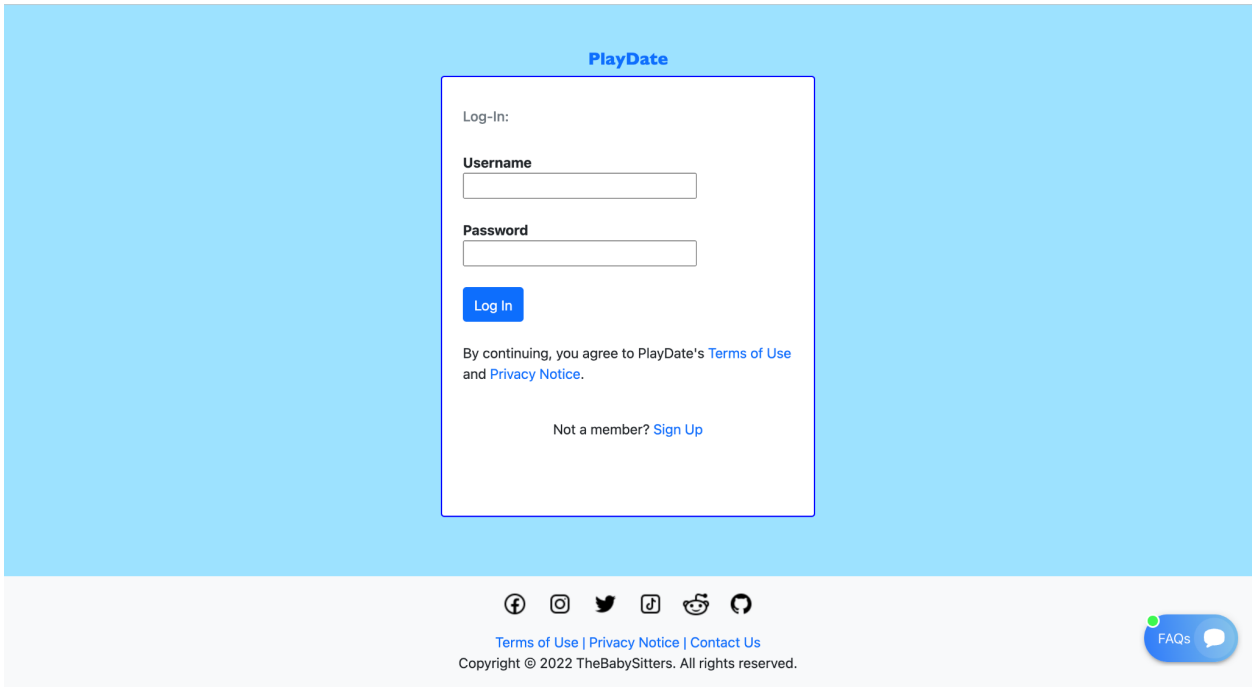
3.1.2 Public Event Page:



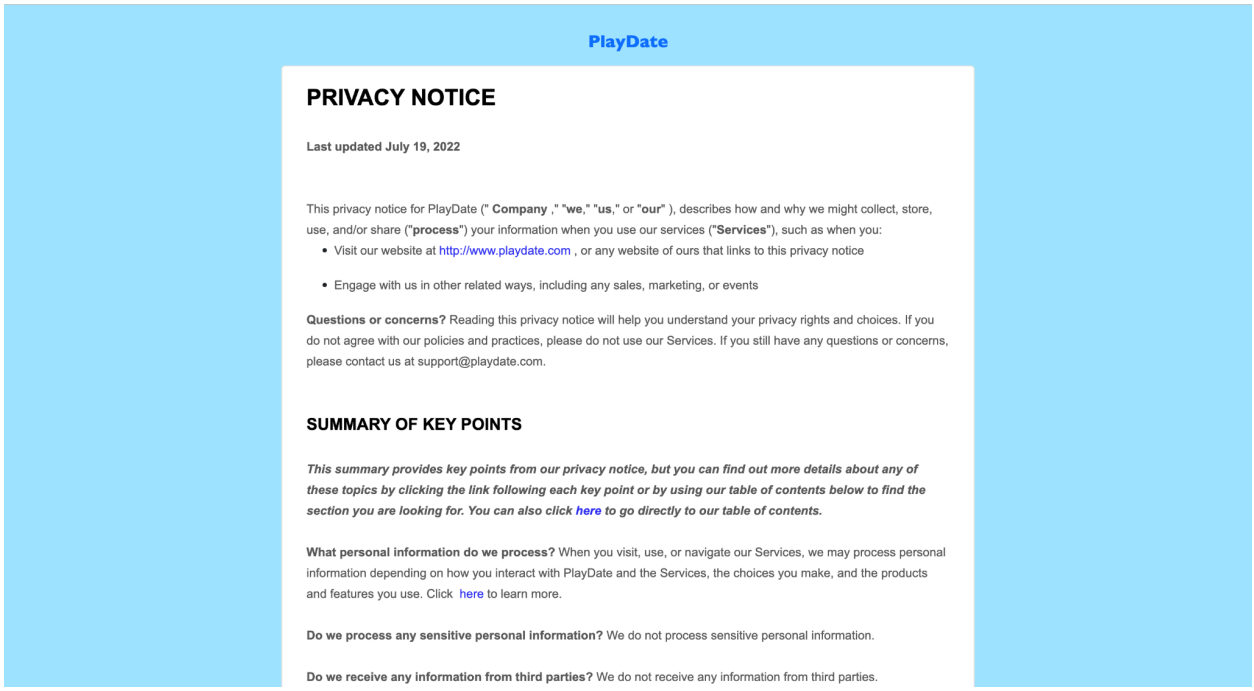
3.1.3 Sign Up Page:



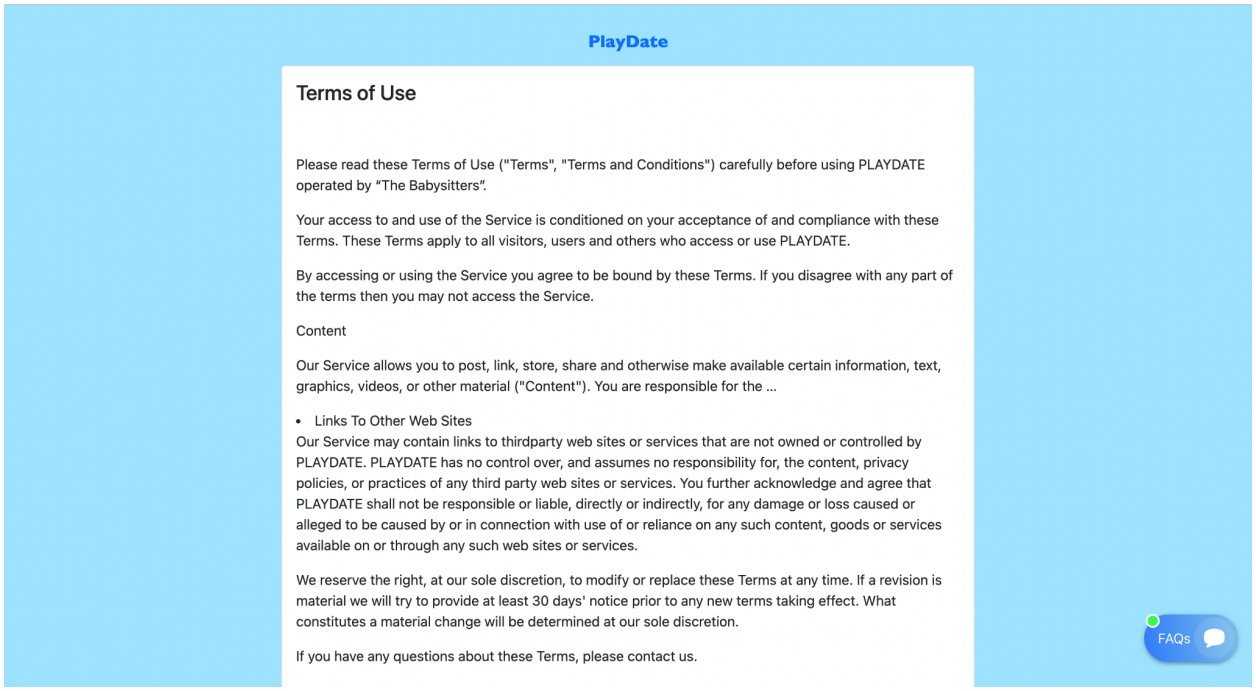
3.1.4 Log In Page:



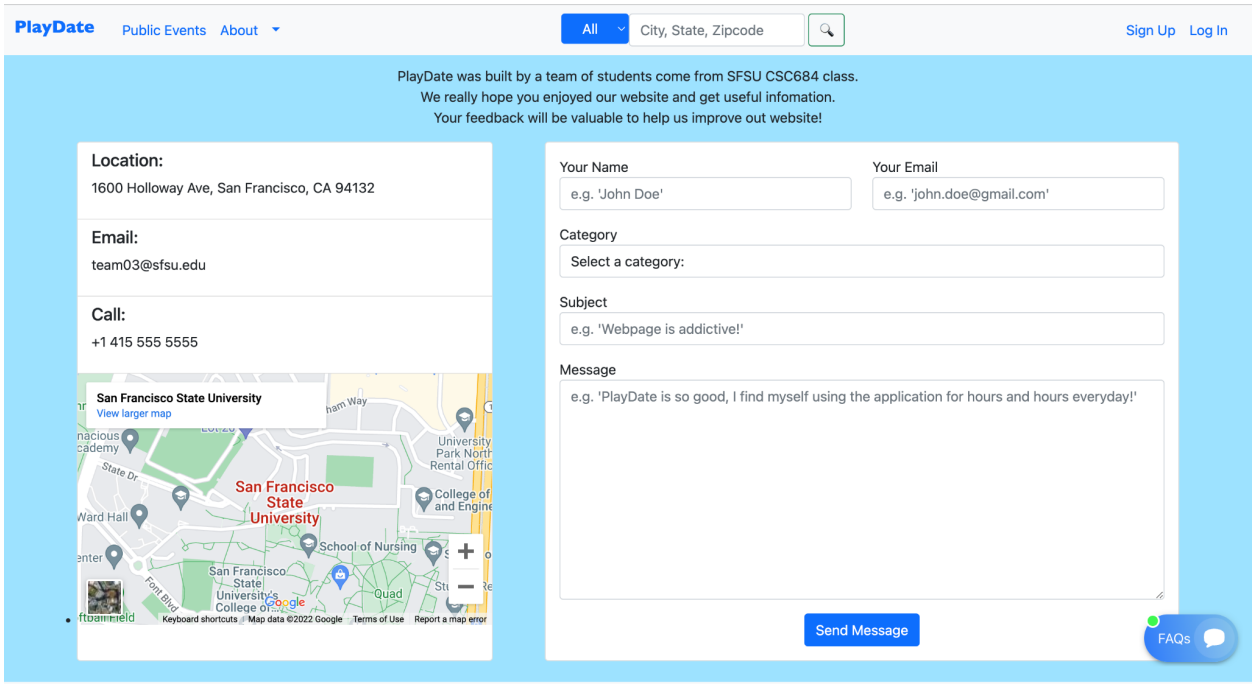
3.1.5 Privacy Notice:



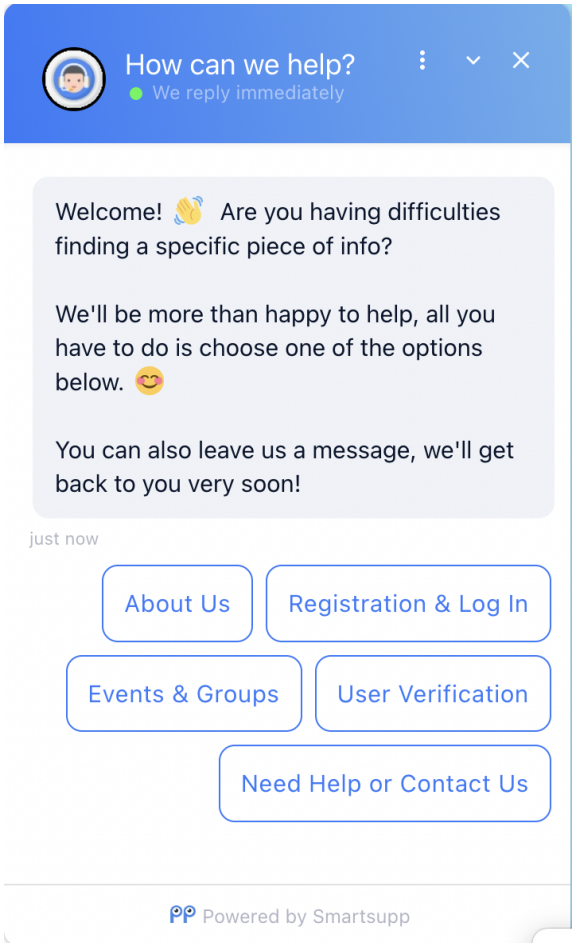
3.1.6 Terms of Use:



3.1.7 Contact Support:

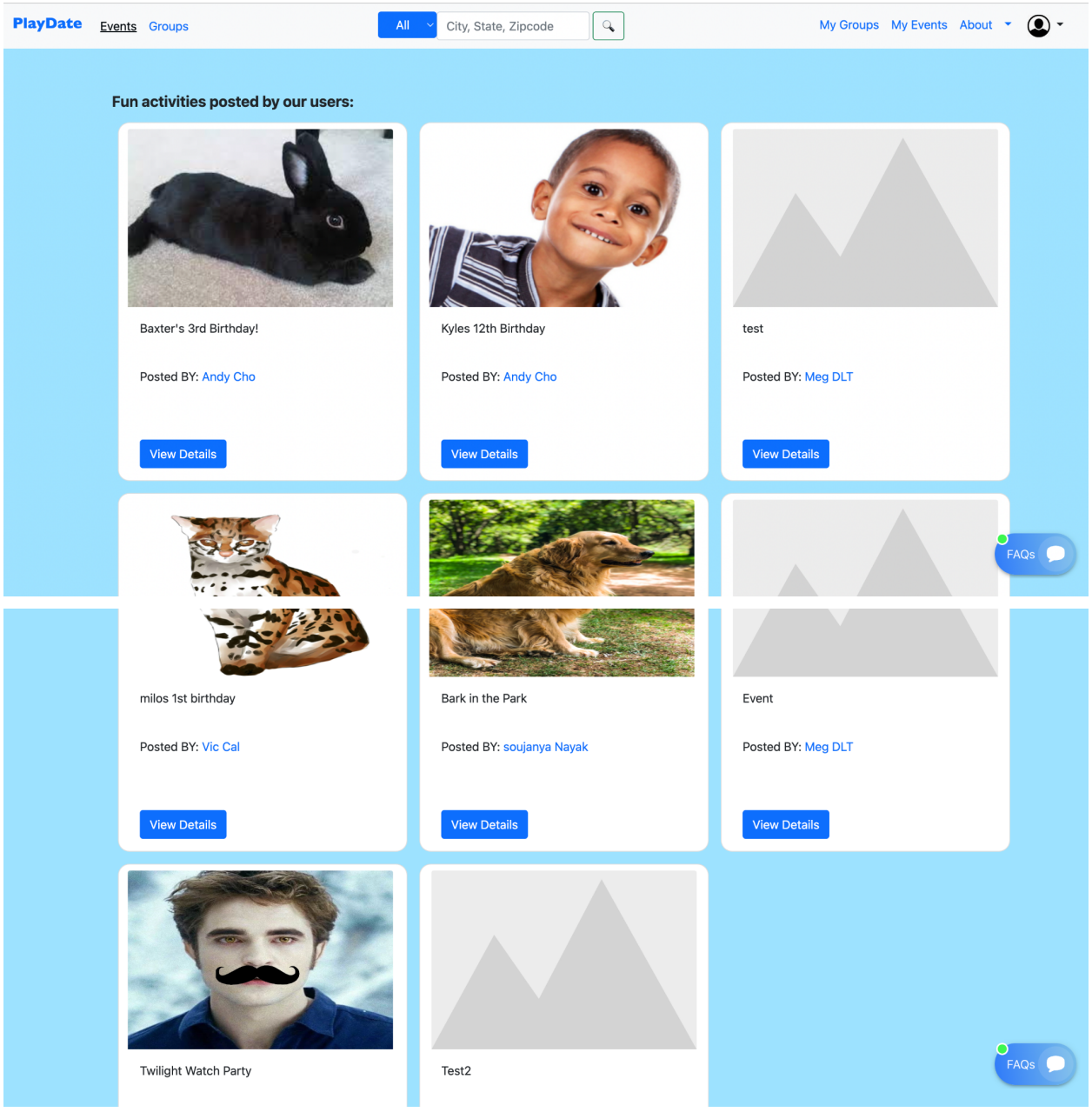


3.1.8 FAQs Bot

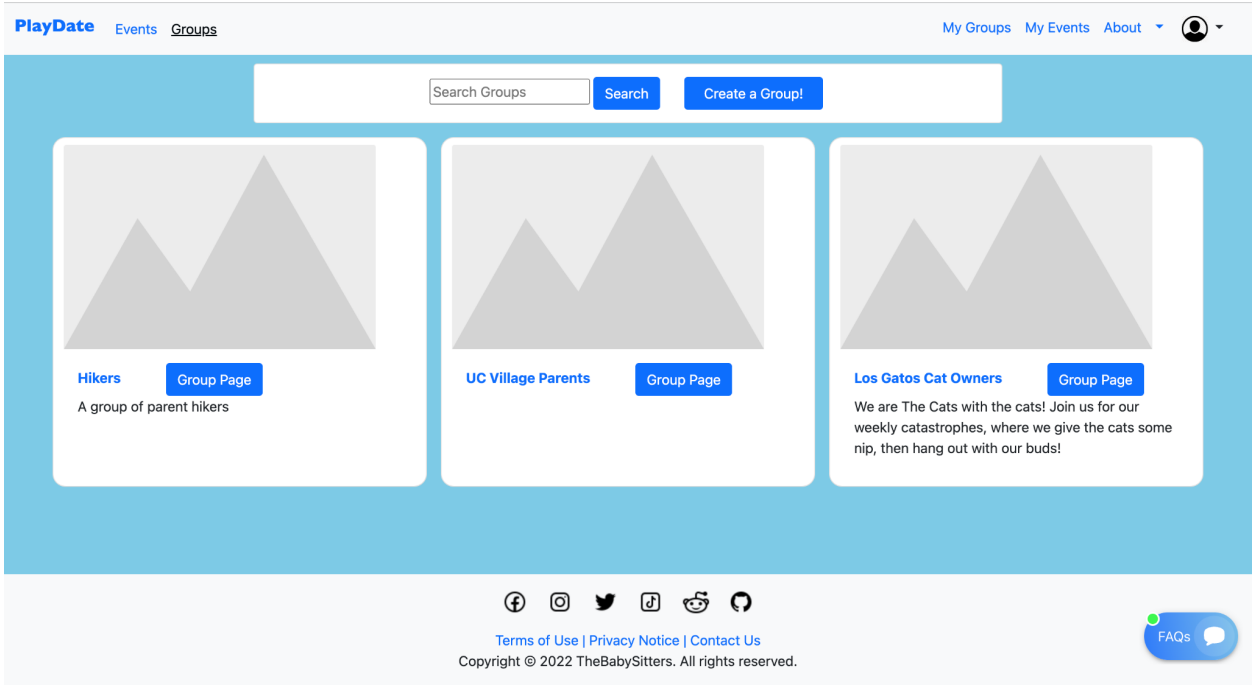

How can we help?
We reply immediately
Welcome! 👋 Are you having difficulties finding a specific piece of info?
We'll be more than happy to help, all you have to do is choose one of the options below. 😊
You can also leave us a message, we'll get back to you very soon!
just now
About Us
Registration & Log In
Events & Groups
User Verification
Need Help or Contact Us
Powered by Smartsupp

3.2. As a registered user (but non-verified), you can access the following more pages:

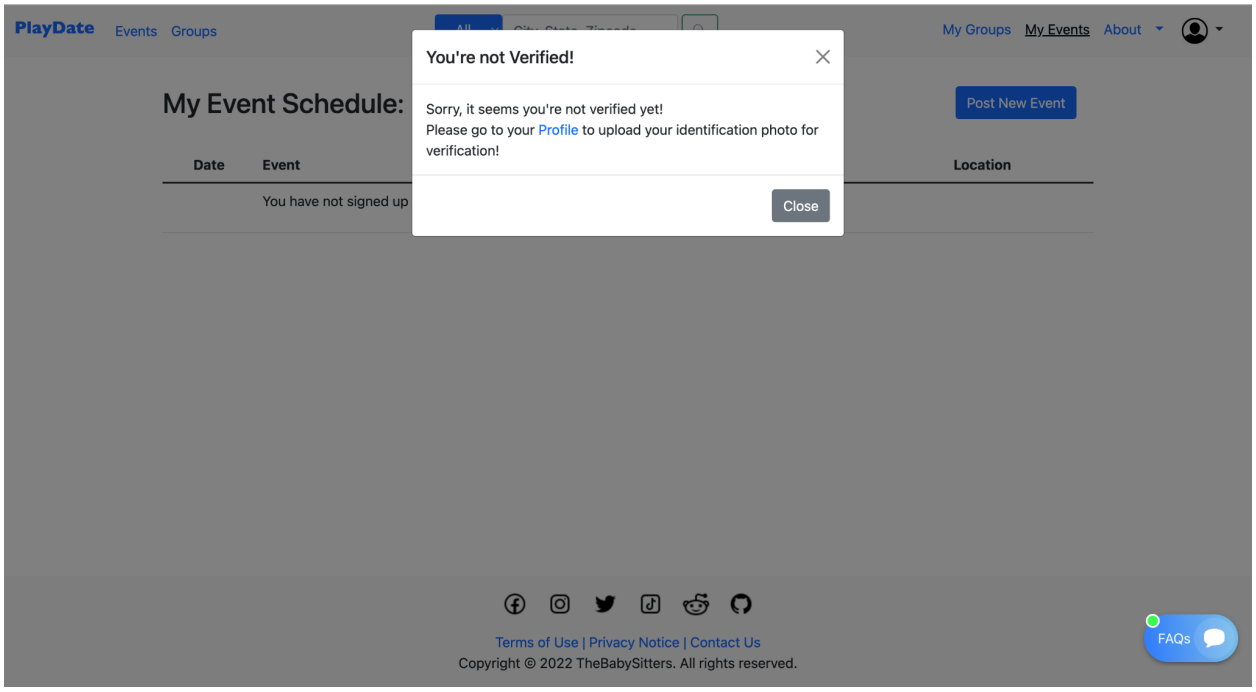
3.2.1 Events Page:



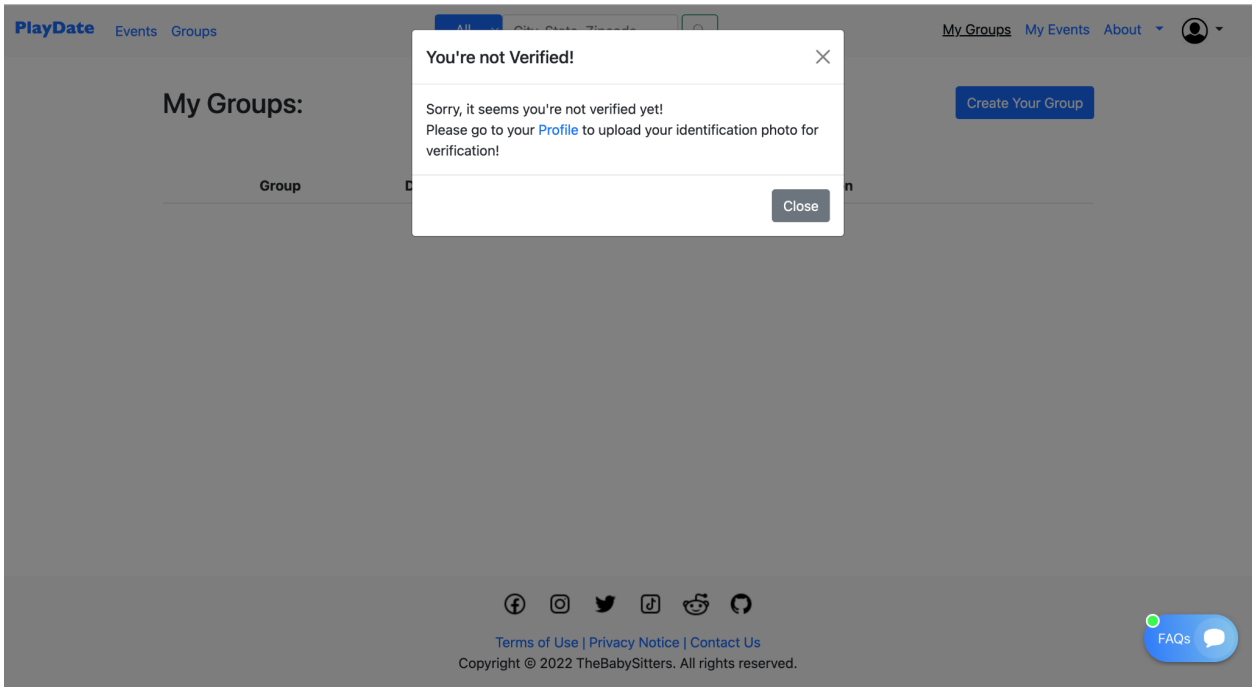
3.2.2 Groups Page:



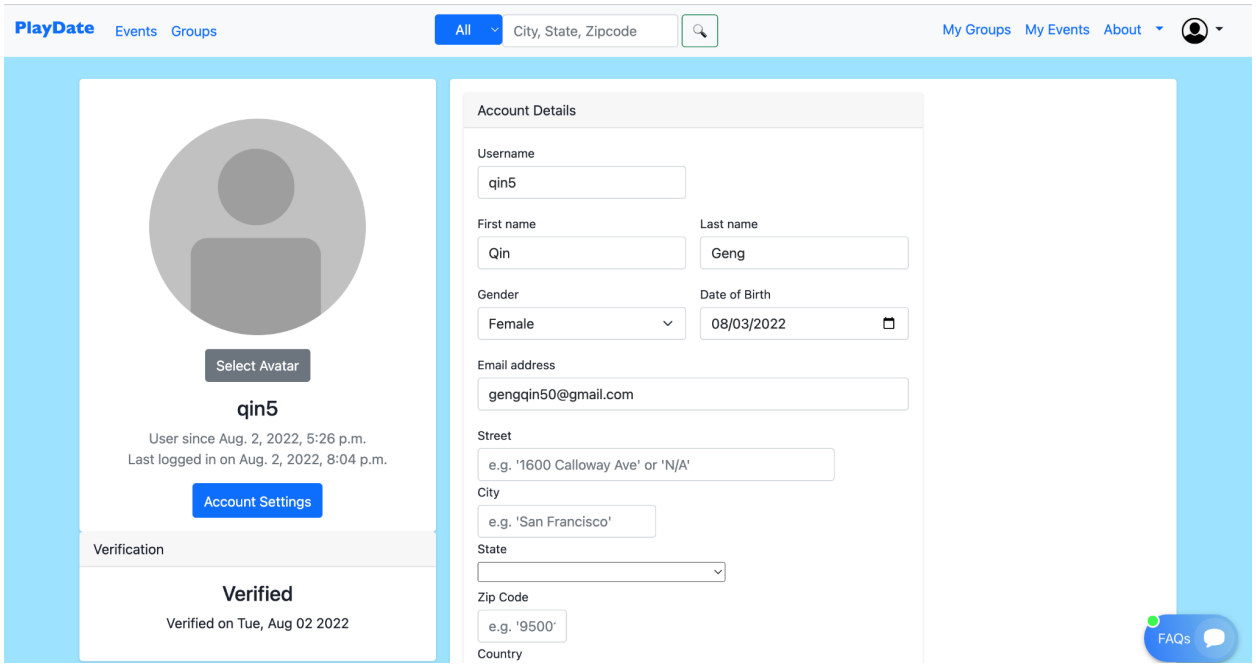
3.2.3 You can't Create Event

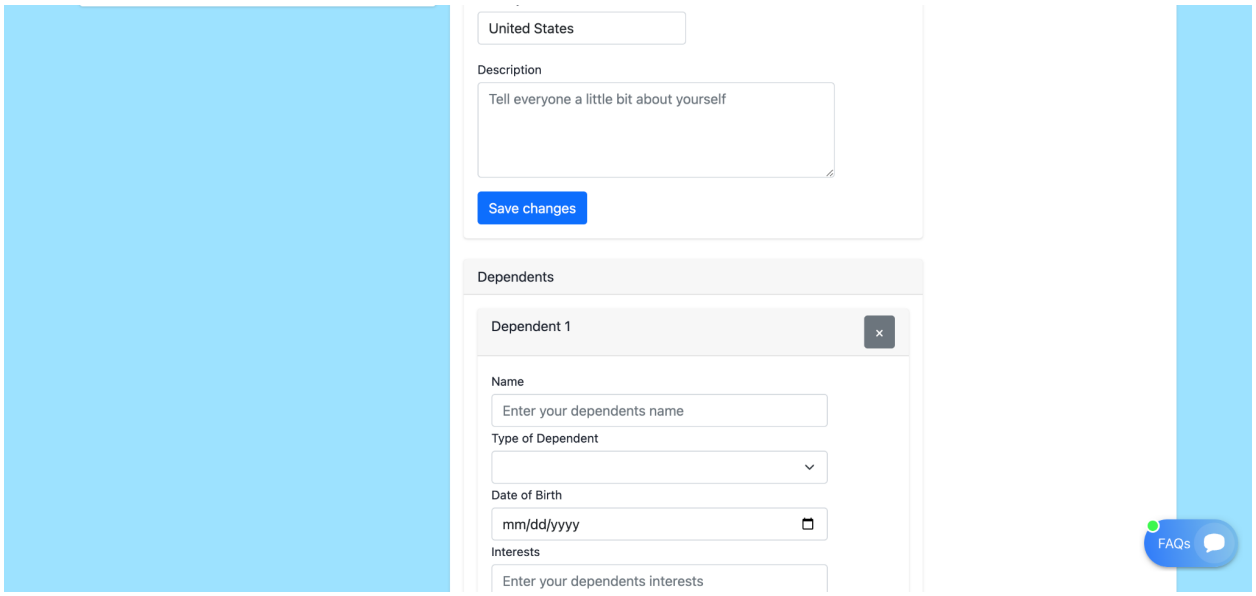


3.2.4 You can’t Create Group



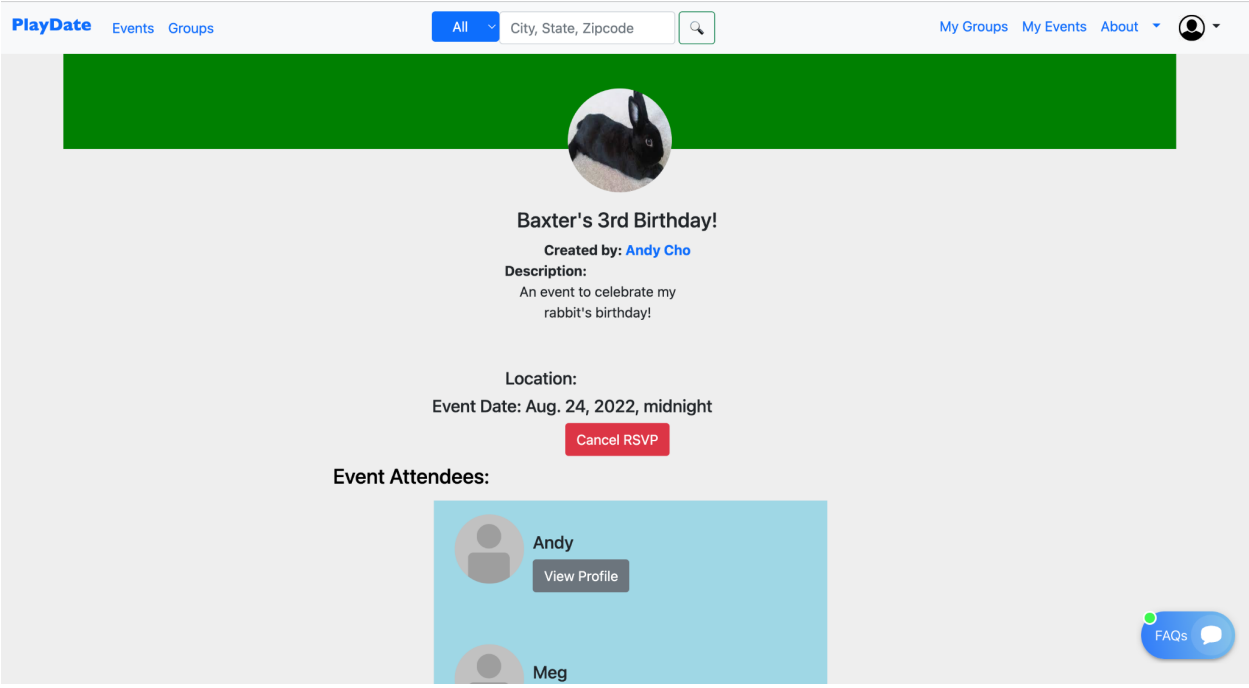
3.2.5 Profile Page



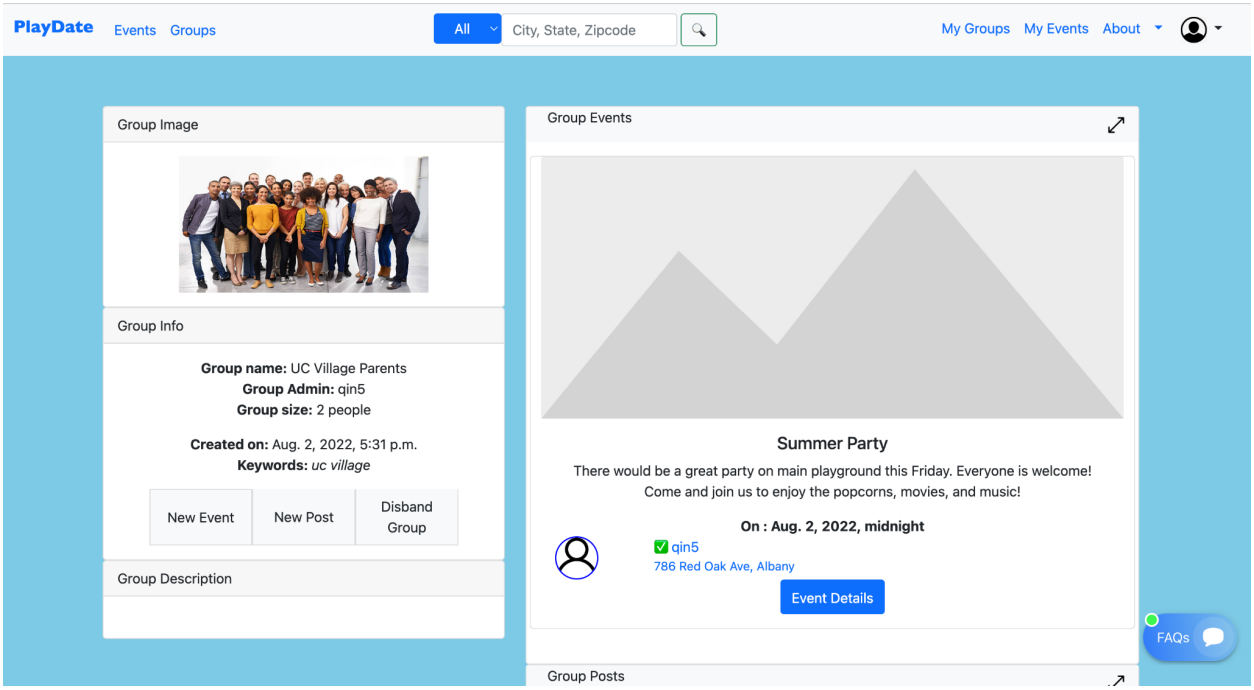


3.3. As a verified user, you can access the following more pages:

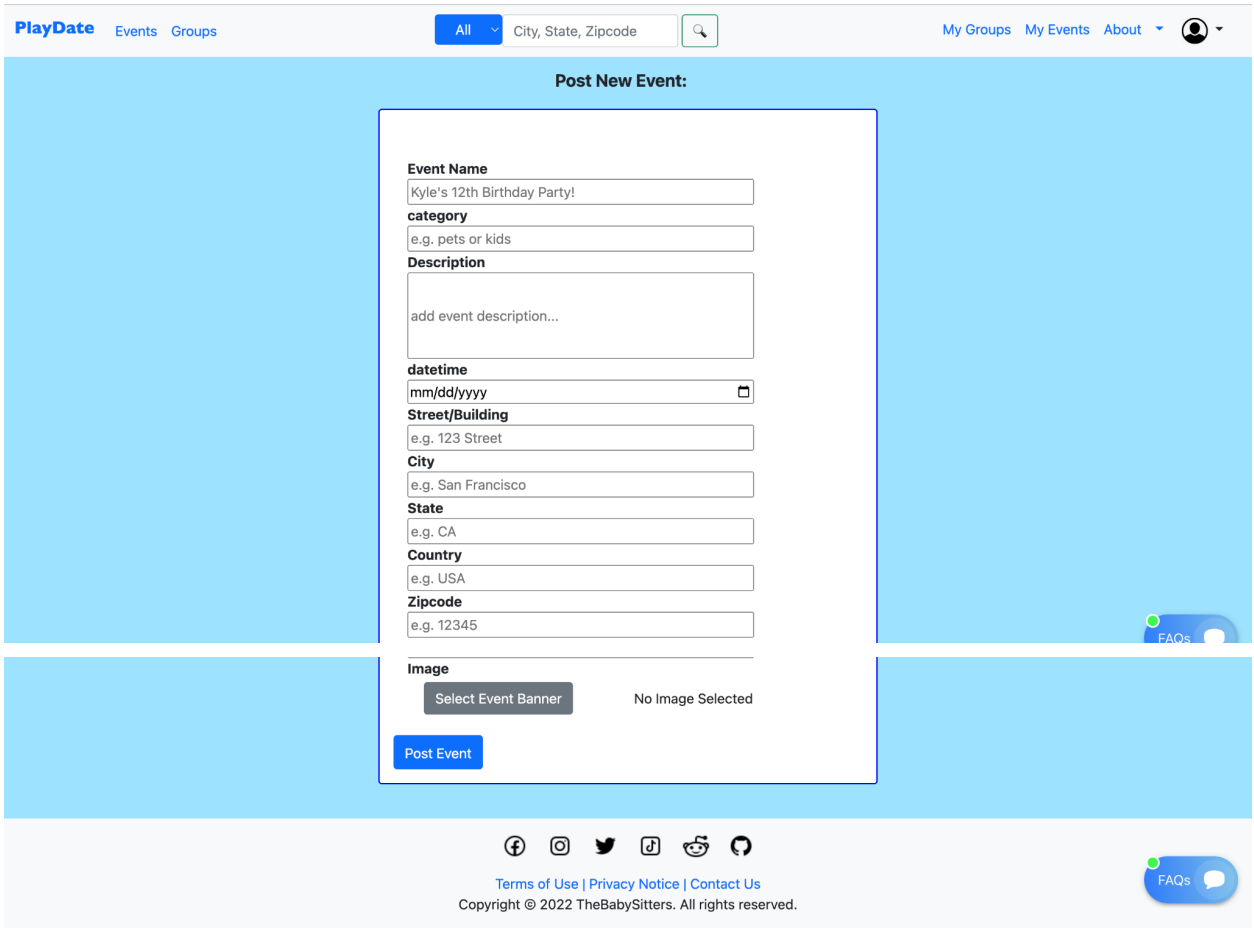
3.3.1 Join Event



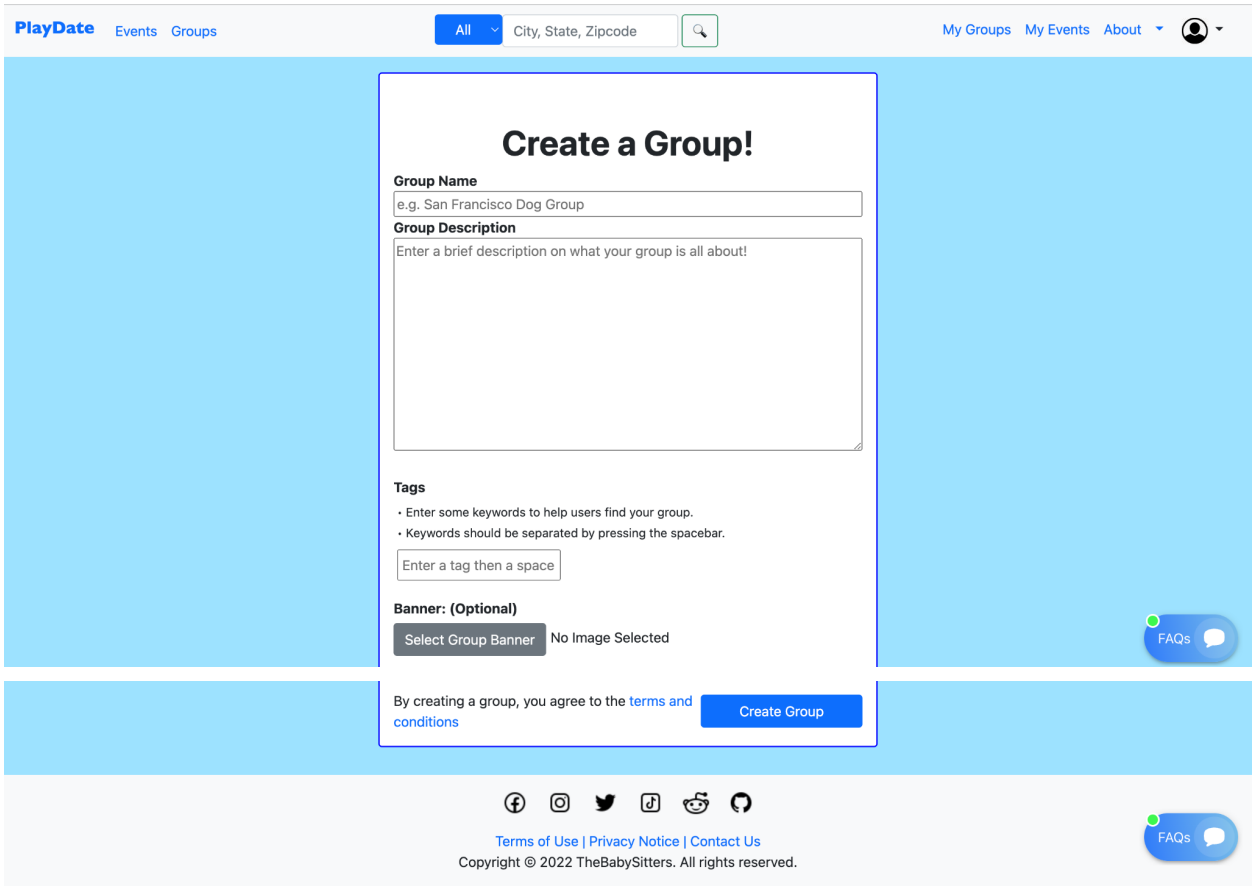
3.3.2 Join Group



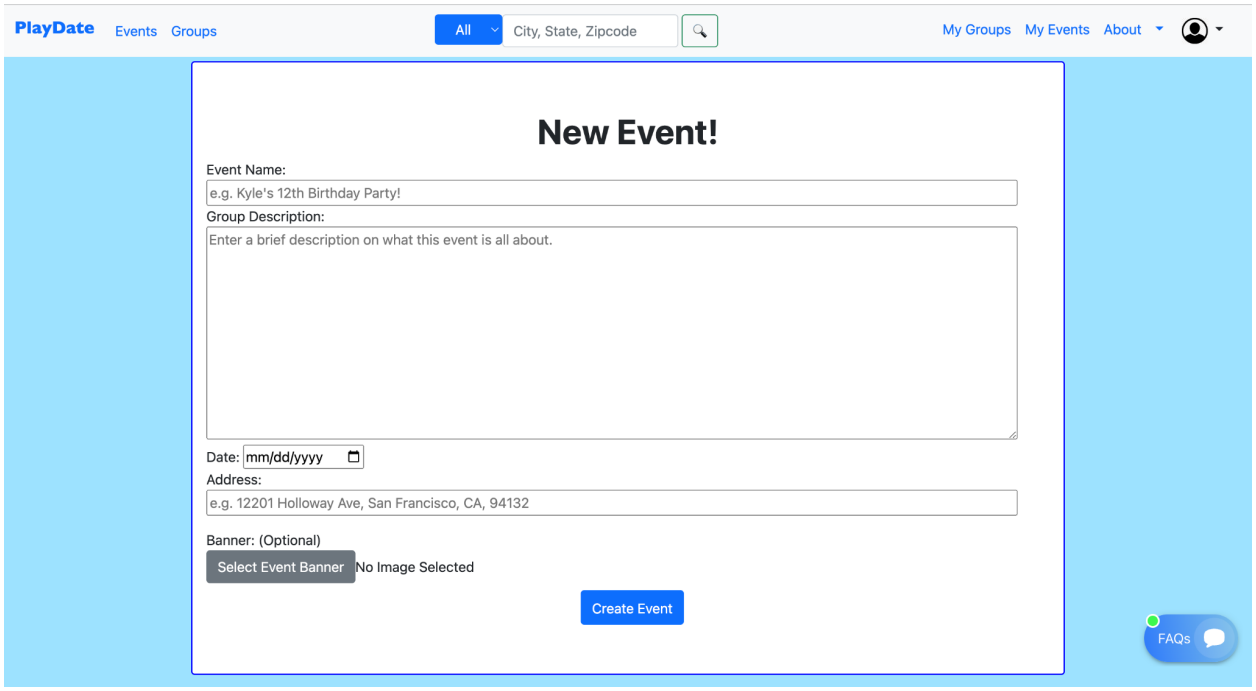
3.3.3 Create Event



3.3.4 Create Group

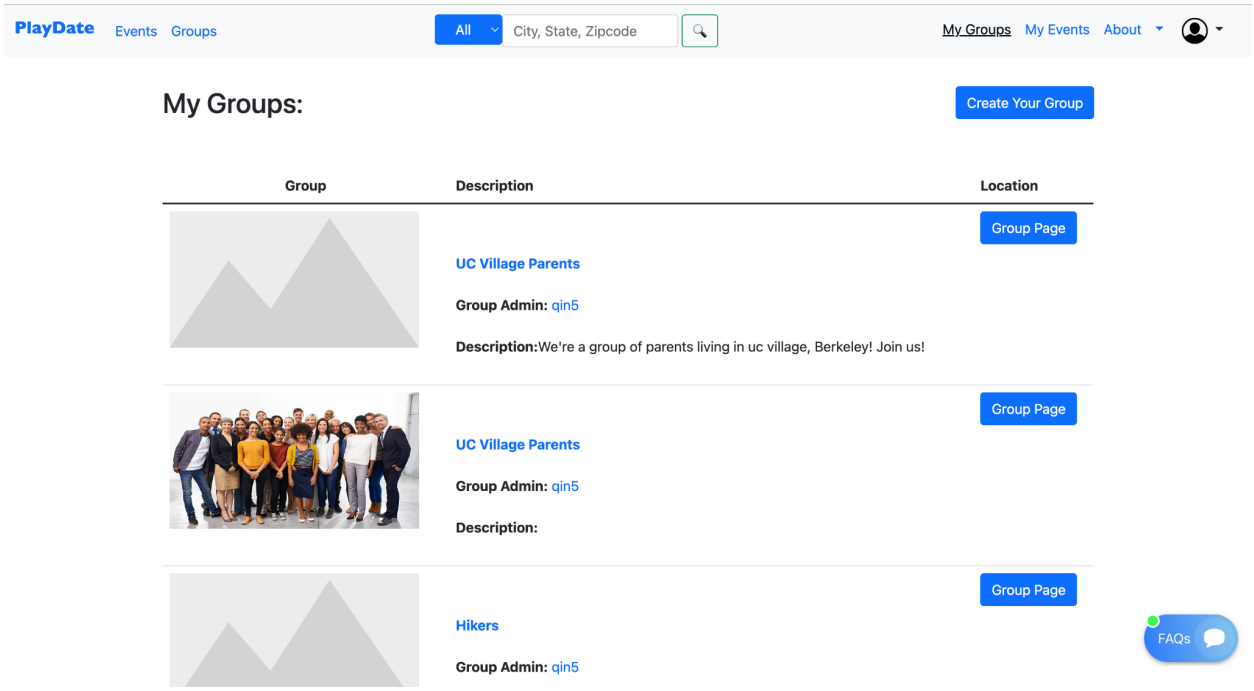


3.3.5 Create Group Event

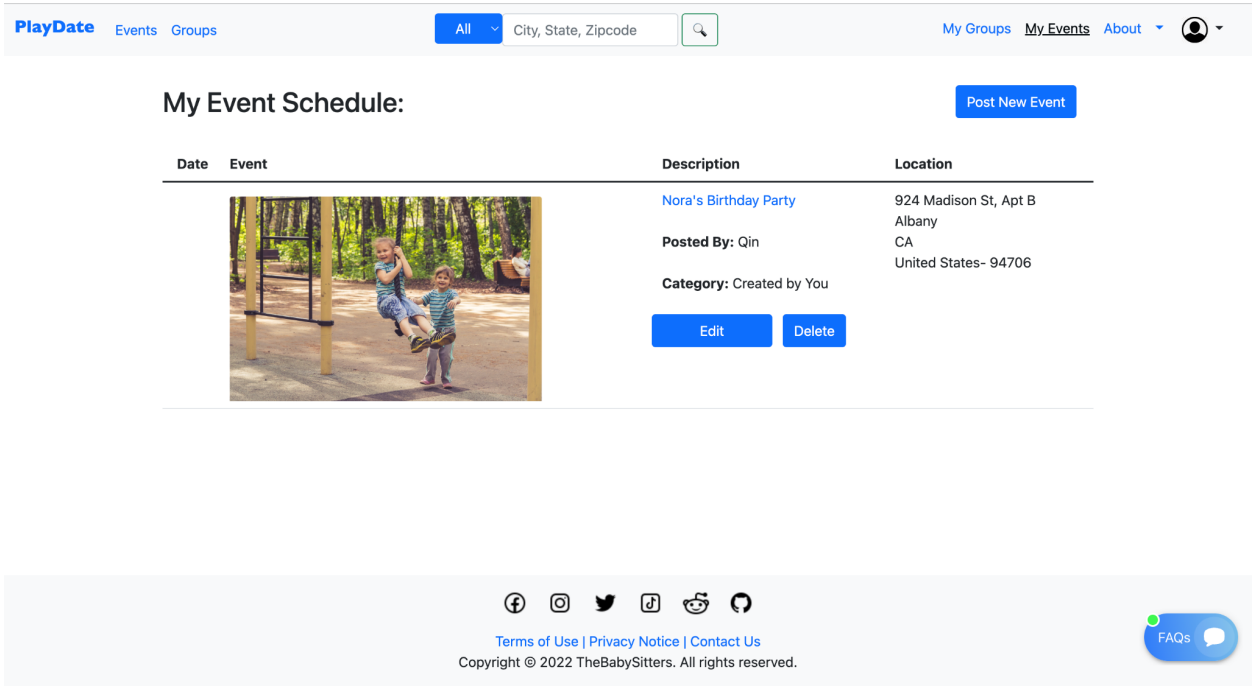


3.3.6 Create Group Post

3.3.7 My Groups

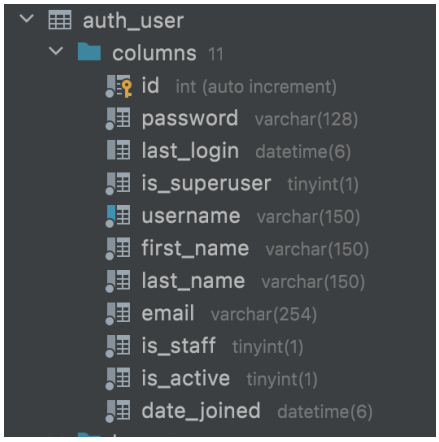


3.3.8 My Events

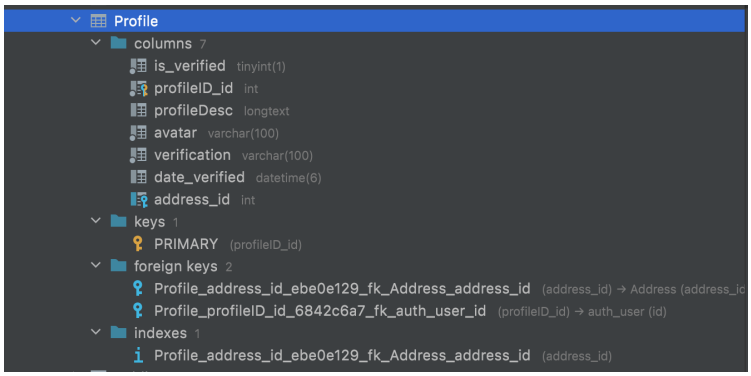


4. Screenshots of key DB Tables

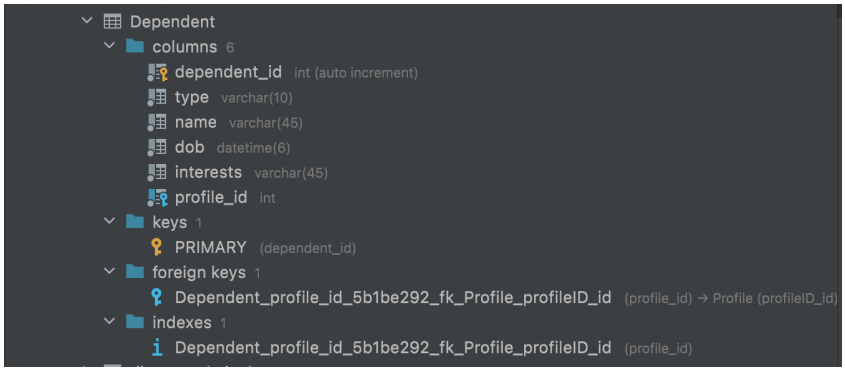
1. User



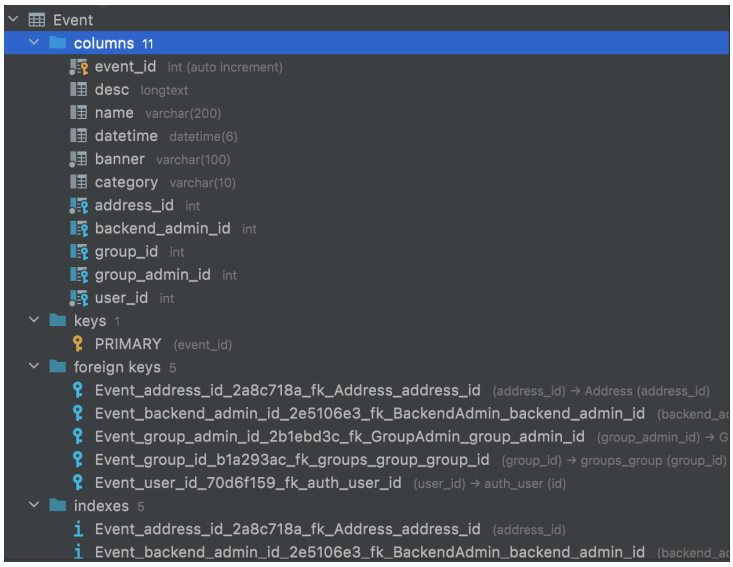
2. Profile



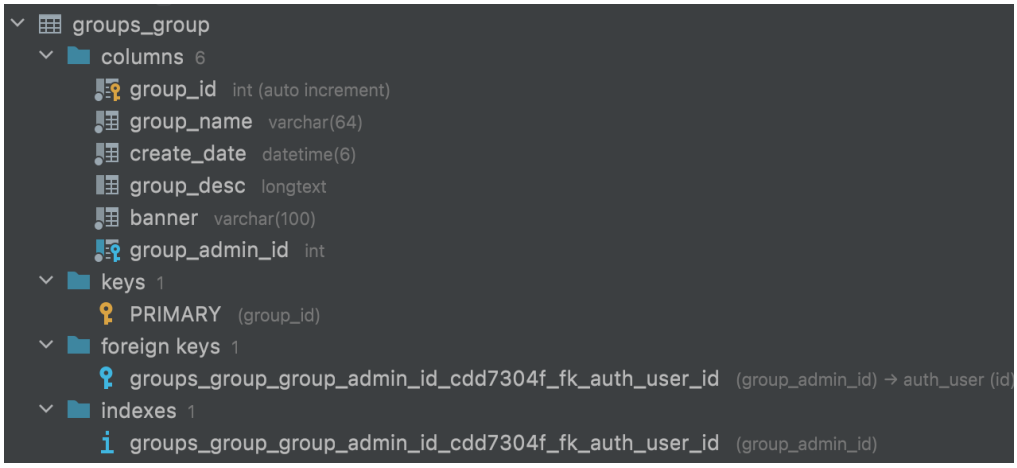
3. Dependents



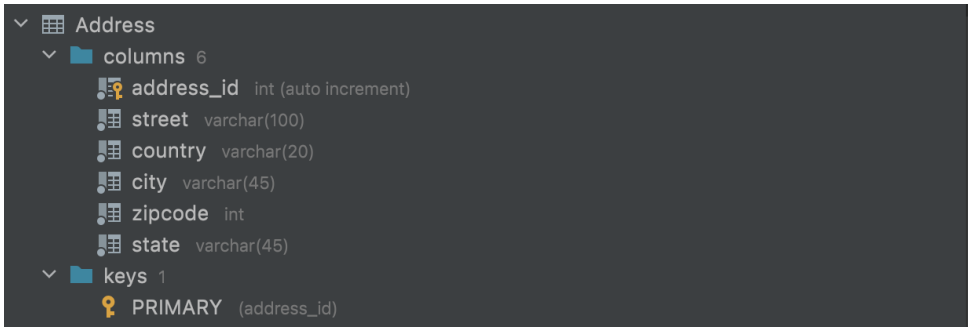
4. Event



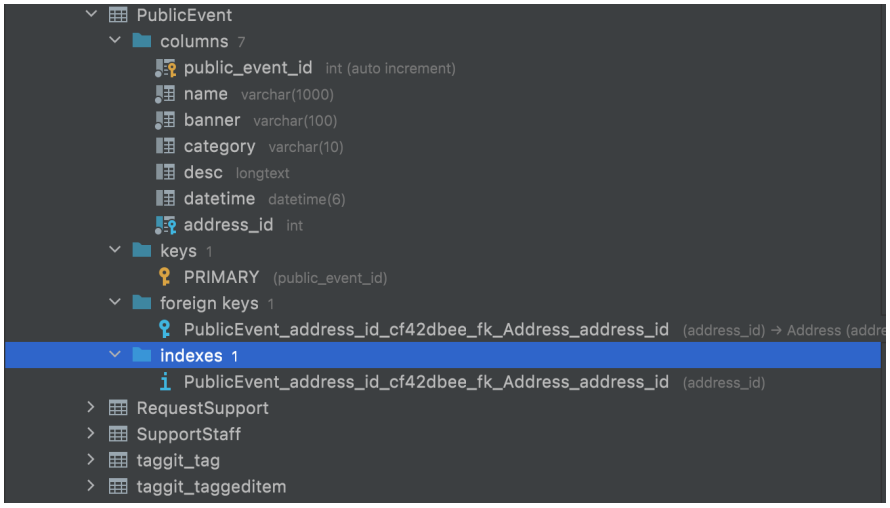
5. Group



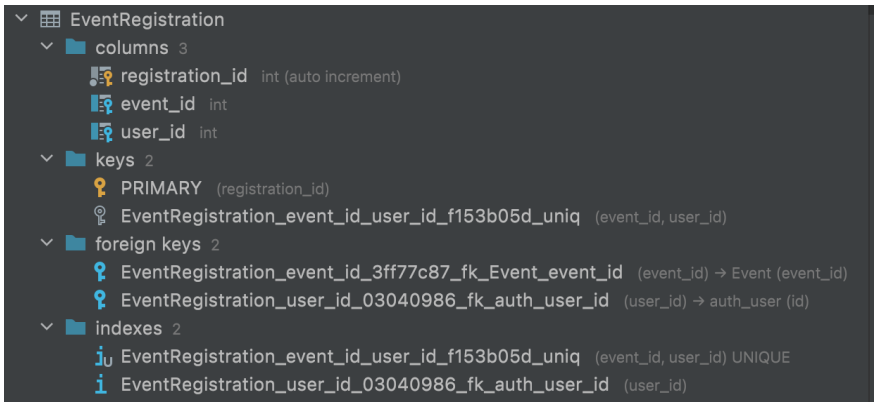
6. Address



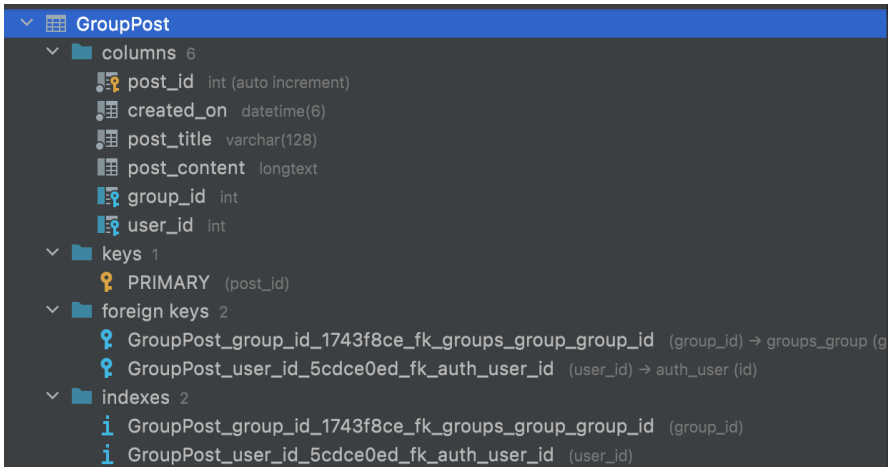
7. Public Event



8. Event Registration



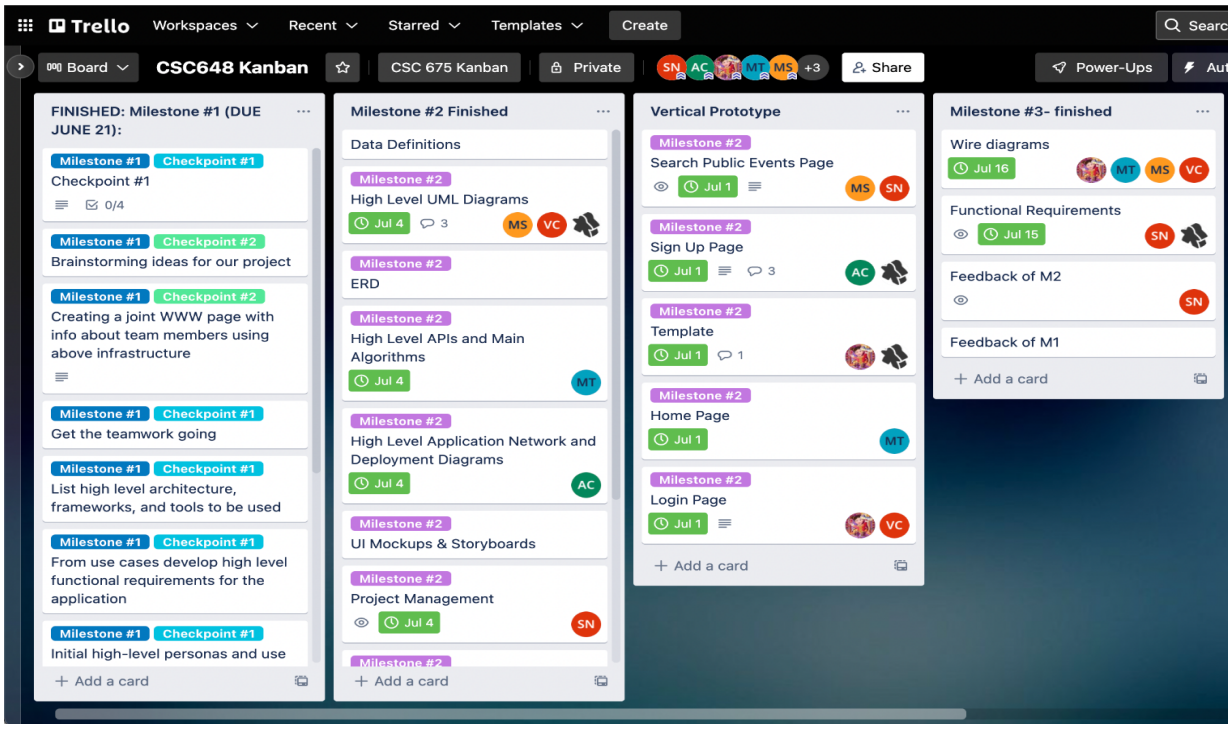
9. Group post

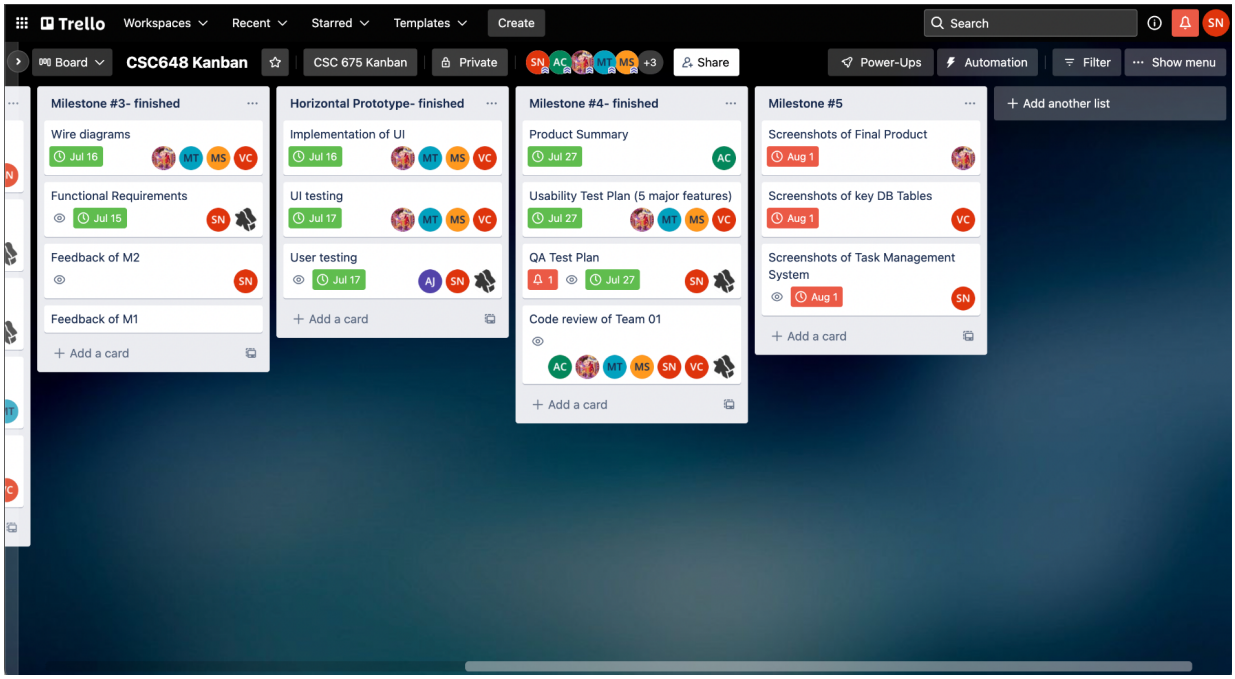


10. Support staff

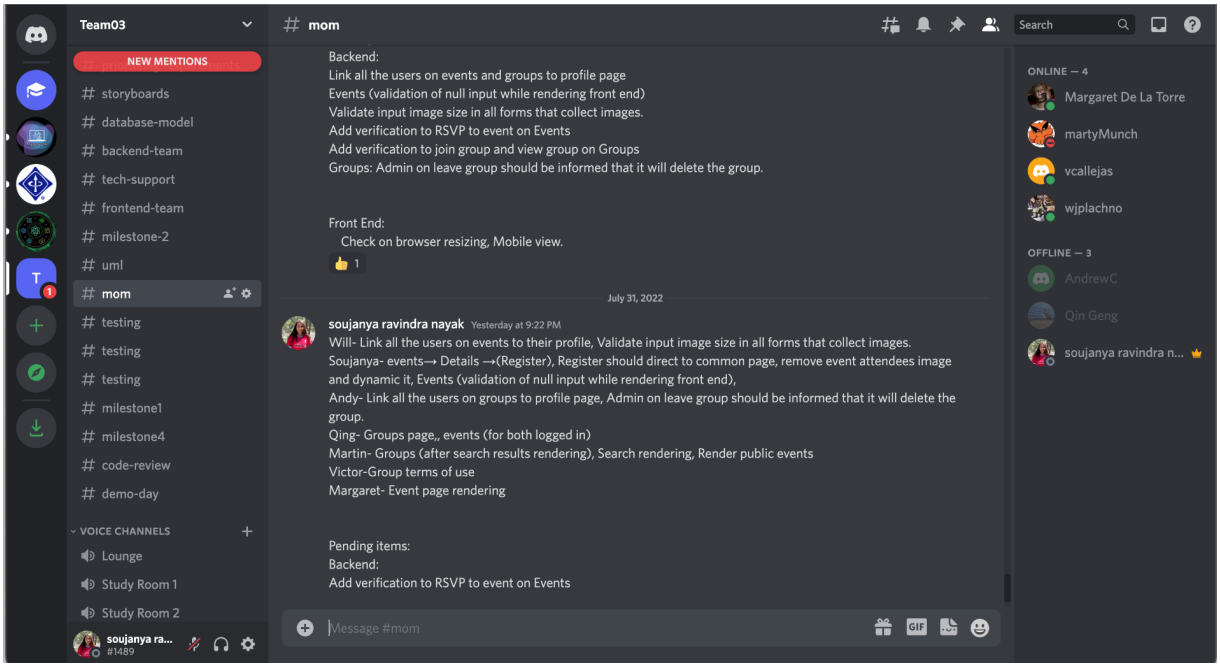
5. Screenshots of Task Management System

Trello: We used Trello to make Kanban style lists for each Milestone





Discord: We make minutes of meeting every scrum and write tasks for each team member for that day. This is later posted on Discord.



6. Team member contributions

Team members' contribution, based on their contribution to brainstorming in scrums, code implementation and milestones.

Score 1: no work at all…. Score 10: full contribution

Name	Role	Contribution Score
Soujanya Ravindra Nayak	Team Lead Document Contributor Backend Team	10
Margaret De La Torre	Front-end lead Document Contributor	10
Martin Salvatierra	Front-end Document Contributor	10
Andy Cho	Back-end lead Document Contributor	10
Qin Geng	Front-end Document Contributor	10
William Plachno	Git Master Backend Team Document Contributor	10
Victor	Front-end Document Contributor	9

7. Post Analysis

This project has been a great learning experience for all of us. We were so excited to be part of this project that we started off by adding many features to our application without being able to judge the possibility of implementation in such a short term. As we started implementing, we realized it was possible to implement but we ran out of time.

As software engineers, we need to be open to new technologies. There will not be a perfect team where everyone knows everything, and it is a tough call to decide on technologies. Many of us were new to various tools and languages, but the team was open to learning and adapting due to which we opted for a language and framework that was completely new to half of the team. This decision was taken with the idea of the remaining half supporting the team. And this decision was proven right, with the way the team supported each other, and we made a front-end team and backend team based on people skills.

During our first milestone, we had sufficient time after implementation, that could have been used to onboard the team with the software stack. This affected our milestone 2, where we had to work on ramping up everyone, helping and debugging the development environments on installation of tools along with working towards Milestone 2.

Our initial milestone was not so good but that helped us shape the rest of the software engineering journey. We received valuable feedback from professor, team members and we also looked at how other teams were working it out. We adapted good practices, some of which were everyday scrum, constructive feedback from team members. We started scrums every day, where we discussed progress on tasks and to-dos for each developer. We divided the tasks such that we had 4 front end developers and 3 backend developers working parallelly for faster implementation.

We improvised our ways of execution to make efficient progress. We divided tasks such that more than one person was working on every item of Milestone, so that we had multiple perceptions and brainstorming. One of our team members created a troubleshooting guide to help others with their issues. Minutes of the meeting were recorded and shared with everyone, so that none is left out if they couldn't attend scrums.

We could make it better, but time has always been a constraint to our ambitious project. If something that could have been changed would be on adhering to timelines. Many times, we were submitting milestones during the last minute of submission, which caused a lot of stress and last-minute issues on the servers. This could have been avoided if we had completed the tasks sooner. But I believe moving forward our projects will be more planned and time driven to deliver all tasks on time.