

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	3
2. Main Use Cases	4
3. Main data items and entities	26
4. Functional Requirements	27
5. Non-functional Requirements	34
6. Competitive analysis	36
7. High-level system architecture and technologies used	41
8. Checklist	43
9. Team contribution	44

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	<p>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.</p>
Diagram	<pre> graph LR subgraph PlayDate_System [PlayDate System] direction TB UC1[View Public Info] UC2[Search Public Info] UC3[Register] UC4[Login] UC5[Edit Profile] UC6[Upload Proof] UC7[View Events] UC8[Comment Events] UC9[Sign Up Events] UC10[Post Events] UC11[Add User In] UC12[Delete Post] UC13[Remove Users] UC14[Search Events] UC15[Cancel Events] UC16[Log Out] UC17[Edit Her Post] UC18[Delete Her Post] end MGU[Mary General User] -.-> UC1 MGU -.-> UC2 MGU -.-> UC3 MGU -.-> UC4 MRU[Mary Registered User] -.-> UC5 MRU -.-> UC6 MRU -.-> UC7 MRU -.-> UC8 MRU -.-> UC9 MRU -.-> UC10 MRU -.-> UC14 MRU -.-> UC15 MRU -.-> UC16 MRU -.-> UC17 MRU -.-> UC18 Admin -.-> UC11 Admin -.-> UC12 Admin -.-> UC13 </pre> <p>Visual Paradigm Online Free Edition</p>
Requirements	<p>1. General User</p> <ul style="list-style-type: none"> 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

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

Title	2. Group Creation
Actors	Tom (Group admin), Helen(Registered user)
Description	<p>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.</p>
Diagram	<p>Visual Paradigm Online Free Edition</p> <p>Visual Paradigm Online Free Edition</p>
Requirements	<p>2. Registered User</p> <p>2.23 A registered user shall be able to become a group user.</p> <p>2.24 A registered user shall be able to create many groups.</p> <p>2.25 A registered user shall be able to become a group admin.</p> <p>2.26 A registered user shall be able to search for groups based on location</p> <p>2.27 A registered user shall be able to search for groups based on interest</p> <p>2.28 A registered user shall be able to search for groups based on group name</p> <p>2.29 A registered user shall be able to join many groups.</p>

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

	<p>3.1 A group shall have at least one group user.</p> <p>3.2 A group shall have at least one group admin.</p> <p>3.3 A group shall have 0 or more comments</p> <p>3.4 A group shall have 0 or more events</p> <p>3.5 A group shall include the creator of the group</p> <p>3.6 A group shall be able to be joined by request</p> <p>3.7 A group shall be able to be joined by invite</p> <p>3.8 A group shall contain no more than 50 group users</p> <p>12. Group Event</p> <p>12.1. A group event shall immediately accept sign-ups from group users of that group</p> <p>12.2 A group event can be edited by the user who created that event</p> <p>12.3 A group event can be edited by the group admin</p> <p>12.4 A group event can be created by any group user</p> <p>12.5 A group event can be signed up for by group users of that group</p> <p>12.6 A group event can be canceled by the group admin</p>

Title	3. Joining Group
Actors	Mary(Registered user), Tom(Group admin)
Description	<p>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.</p>
Diagram	<p>Visual Paradigm Online Free Edition</p>
Requirements	<p>2. Registered User 2.33 A registered user shall be able to browse groups</p> <p>12. Group Event 12.7 A group event shall include a sign up by the event creator</p>

Title	4. Creating Group Events
Actors	Helen, Tom, Ben (Group users)
Description	<p>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.</p>
Diagram	<pre> graph LR subgraph "Creating Group Events" VGS([View Group Schedule]) CGE([Create Group Event]) RGEN([Receive Group Event Notification]) SUNE([Sign Up for Event through Notification]) SUG([Sign Up for Event through Group]) end Helen((Helen)) subgraph "Helen's Group" Tom((Tom)) Ben((Ben)) end Helen --- VGS Helen --- CGE Helen --- RGEN Helen --- SUNE Helen --- SUG Tom --- RGEN Tom --- SUNE Ben --- SUG </pre> <p>The diagram illustrates the process of creating group events. It features a central box titled 'Creating Group Events' containing five use cases: 'View Group Schedule' (blue), 'Create Group Event' (blue), 'Receive Group Event Notification' (grey), 'Sign Up for Event through Notification' (pink), and 'Sign Up for Event through Group' (teal). To the right, there are three actors: Helen (blue head), Tom (pink head), and Ben (teal head). Helen is connected to all five use cases. Tom is connected to 'Receive Group Event Notification' and 'Sign Up for Event through Notification'. Ben is connected to 'Sign Up for Event through Group'. A grey box labeled 'Helen's Group' contains Tom and Ben, indicating they are members of the group Helen created.</p>
Requirements	<p>4. Group User</p> <p>4.14 A group user who is part of a group shall be able to view a heatmap of the group schedule.</p> <p>4.15 A group user shall be able to update their availability on group heatmap.</p> <p>4.16 A group user shall receive a notification when a group receives a Post.</p>

	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	<p>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.</p>
Diagram	<pre> sequenceDiagram actor Eric actor Al actor Cole Eric->>CreatePublicEvent: Create Public Event Eric->>SendPublicInvite: Send Public Invite to Filtered Users Al->>RespondToPublicInvite: Respond to Public Invite Eric->>ApprovePublicInviteResponse: Approve Public Invite Response Cole->>SearchForNearbyEvents: Search for Nearby Events Cole->>RequestEventSignup: Request Event Signup Eric->>ApprovePublicRequest: Approve Public Request Cole->>FindElapsedEvent: Find Elapsed Event Cole->>ViewEventAttendees: View Event Attendees Cole->>SendGroupInvite: Send Group Invite SendGroupInvite-->>Al: </pre> <p>The diagram illustrates the interactions between three actors (Eric, Al, and Cole) and a series of use cases. Eric initiates the process by creating a public event and sending out public invites. Al responds to a public invite, which Eric then approves. Cole searches for nearby events and requests to join. Eric approves Cole's request. Cole then finds the event listing, views attendees, and sends a group invite to Al.</p>

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

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	<pre> sequenceDiagram actor Tom as Tom (Registered User) participant EmergencyAssistance as Emergency Assistance participant Police as Police participant UnregisteredUser as (Unregistered User) Tom->>EmergencyAssistance: Add emergency contact EmergencyAssistance->>Tom: Raise Emergency EmergencyAssistance->>Police: Notify Police EmergencyAssistance->>UnregisteredUser: Notify emergency contact Police-->>EmergencyAssistance: <<Include>> UnregisteredUser-->>EmergencyAssistance: <<Include>> </pre> <p>The diagram illustrates the Emergency Assistance process. It features a system boundary labeled "Emergency Assistance" containing four use cases: "Add emergency contact", "Raise Emergency", "Notify Police", and "Notify emergency contact". The actors involved are "Tom (Registered User)", "Police", and "(Unregistered User)". Tom interacts with the system by adding an emergency contact and raising an emergency. The system then notifies the police and the emergency contact. The "Notify Police" and "Notify emergency contact" use cases include the "Raise Emergency" use case, as indicated by dashed arrows with the stereotype <<Include>>.</p>
Requirements	<p>2. Registered User 2.41 A registered user shall be able to request emergency assistance via PlayDate application.</p> <p>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.</p>

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	<pre> sequenceDiagram actor Jeff actor Attendees as Event attendees participant U1 as Accept event invitation participant U2 as Create a survey as a post participant U3 as Respond to survey post Jeff->>U1 U1->>U2 Jeff->>U2 U2->>U3 Attendees->>U3 </pre> <p>The diagram is a UML Use Case Diagram titled "Add survey to event post". It features two actors: "Jeff" (a stick figure with a blue head) and "Event attendees" (three stick figures with green heads). There are three use cases represented by ovals: "Accept event invitation" (purple), "Create a survey as a post" (purple), and "Respond to survey post" (pink). The relationships are as follows: Jeff is associated with "Accept event invitation" and "Create a survey as a post". "Accept event invitation" is associated with "Create a survey as a post". "Event attendees" is associated with "Respond to survey post". "Create a survey as a post" is associated with "Respond to survey post". A message labeled "Text" is shown between Jeff and "Create a survey as a post".</p>
Requirements	<p>2. Registered User</p> <p>2.42 A registered user shall be able to create a survey for an event.</p> <p>2.43 A registered user who is attending the event shall be able to respond to surveys corresponding to that event.</p> <p>2.44 A registered user who created the survey shall be able to delete the survey.</p> <p>2.45 A registered user who created the survey shall be able to modify it.</p> <p>5. Group Admin</p> <p>5.7 A group admin shall be able to delete any surveys on events of their group.</p> <p>4. Group User</p> <p>4.18 A group user shall be able to create a survey for an event of their group.</p> <p>4.19 A group user shall be able to post surveys(polls) for group events in the group.</p> <p>4.20 A group user who is attending the event shall be able to respond to surveys corresponding to that group event.</p> <p>4.21 A group user who created the survey shall be able to delete the</p>

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	<pre> sequenceDiagram actor Tom participant PlayDateApp as PlayDate App participant EmailSystem as Email participant BackendSystem as Backend actor Jim Tom->>PlayDateApp: Submit support request activate PlayDateApp PlayDateApp->>EmailSystem: <<include>> Send email regarding technical issue deactivate PlayDateApp EmailSystem->>BackendSystem: Receive Email activate EmailSystem EmailSystem->>BackendSystem: Send Email deactivate EmailSystem BackendSystem->>Jim: Analyze Issue activate BackendSystem BackendSystem->>Jim: <<include>> Fix Issue deactivate BackendSystem Jim->>EmailSystem: activate Jim Jim->>Tom: deactivate Jim </pre> <p>The diagram illustrates the technical support process. It features two actors: Tom (a registered user) and Jim (a support staff member). The process is divided into three main components: the PlayDate App, the Email system, and the Backend system. Tom initiates the process by submitting a support request through the PlayDate App. This triggers an include relationship to the 'Send email regarding technical issue' use case within the Email system. The Email system then receives the email and sends it to the Backend system. The Backend system analyzes the issue and includes a 'Fix Issue' use case. Finally, Jim, the support staff, receives the notification and sends a response back to Tom via the Email system.</p>
Requirements	<p>2. Registered User 2.46 A registered user shall be able to request for technical assistance from support staff on product bugs.</p> <p>8. Support Staff 8.3 Support staff shall receive emails regarding user technical issues. 8.4 Support staff shall receive emails from registered users</p> <p>4. Group User 4.23 A group user shall be able to request for technical assistance from support staff on product bugs.</p>

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	<pre> sequenceDiagram actor Tom as Tom (Registered User) participant ReviewEvents as Review Events ReviewEvents->>ViewGroup: View group ReviewEvents->>ViewPastEvents: View past events ReviewEvents->>UploadPhoto: Upload photo ReviewEvents->>Comment: Comment ReviewEvents->>CreatePost: Create Post UploadPhoto-->>CreatePost: <<Include>> </pre> <p>The diagram illustrates the 'Review Events' process. An actor, Tom (Registered User), interacts with a system boundary labeled 'Review Events'. Inside this boundary, there are five use cases: 'View group', 'View past events', 'Upload photo', 'Comment', and 'Create Post'. Tom is connected to all five use cases by solid lines. Additionally, a dashed line with an open arrowhead labeled '<<Include>>' connects the 'Upload photo' use case to the 'Create Post' use case, indicating that creating a post includes the action of uploading a photo.</p>
Requirements	<p>4. Group User</p> <p>4.24 A group user who created the event shall be able to create a post to collect reviews after its occurrence.</p> <p>4.25 A group user who attended the event shall be able to give a rating out of 5 stars.</p> <p>4.26 A group user shall be able to include images in their comments</p> <p>12. Group Event</p> <p>12.10 A group event shall contain the cumulative average of all the ratings as the rating of that event.</p>

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	<pre> graph LR subgraph "Add and organize friends" UC1([Add dependents (pet, children)]) UC2([Search for friends]) UC3([Add friends]) UC4([Create friend groups for each dependent]) end Mary((Mary)) --- UC1 Mary --- UC2 Max((Max)) --- UC4 Ninja((Ninja)) --- UC4 Cindy((Cindy)) --- UC4 UC1 --- UC2 UC2 --- UC3 UC3 --- UC4 </pre> <p>The diagram illustrates the 'Add and organize friends' process. It features four use cases: 'Add dependents (pet, children)', 'Search for friends', 'Add friends', and 'Create friend groups for each dependent'. The actors involved are Mary, Max, Ninja, and Cindy. Mary is connected to the first two use cases. Max, Ninja, and Cindy are connected to the fourth use case. The use cases are interconnected in a sequence: 'Add dependents' leads to 'Search for friends', which leads to 'Add friends', which finally leads to 'Create friend groups for each dependent'.</p>
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	<p>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.</p>
Diagram	<pre> graph LR subgraph "Report content or user" UC1([Decides on Permanent Band]) UC2([Report to Support serious offense]) UC3([violates no alcohol rule]) UC4([Remove Group User/delete Post]) UC5([Report incident/ NotifyGroup Admin]) UC6([proof]) UC1 --- UC2 UC2 --- UC3 UC3 --- UC4 UC4 --- UC5 UC5 --- UC6 end Nea[Nea(Event creator/Admin)] Frank[Frank(Event violation)] Ashley[Ashley(Concerned User)] Support[Support(PlayDate)] Nea --- UC1 Nea --- UC2 Nea --- UC4 Frank --- UC3 Ashley --- UC5 Ashley --- UC6 Support --- UC2 </pre> <p>The diagram illustrates the 'Report content or user' process. It features a central box containing six use cases: 'Decides on Permanent Band', 'Report to Support serious offense', 'violates no alcohol rule', 'Remove Group User/delete Post', 'Report incident/ NotifyGroup Admin', and 'proof'. The actors involved are Nea (Event creator/Admin), Frank (Event violation), Ashley (Concerned User), and Support (PlayDate). Nea is connected to 'Decides on Permanent Band', 'Report to Support serious offense', and 'Remove Group User/delete Post'. Frank is connected to 'violates no alcohol rule'. Ashley is connected to 'Report incident/ NotifyGroup Admin' and 'proof'. Support is connected to 'Report to Support serious offense'. The use cases are interconnected in a sequence: 'Decides on Permanent Band' leads to 'Report to Support serious offense', which leads to 'violates no alcohol rule', which leads to 'Remove Group User/delete Post', which leads to 'Report incident/ NotifyGroup Admin', which leads to 'proof'.</p>
Requirements	<p>2. Registered User 2.50 A registered user shall be able to report other users by contacting support staff.</p> <p>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</p>

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

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	<pre> graph LR subgraph Event_Management [Event Management] direction TB UC1([Set Interested]) UC2([Set Maybe]) UC3([RSVP]) UC4([View Current Events]) end Joe((Joe(Registered User))) --- UC1 Joe --- UC2 Joe --- UC3 Joe --- UC4 </pre>
Requirements	<p>2. Registered User</p> <p>2.51 A registered user shall be able to set available events as ‘Interested’</p> <p>2.52 A registered user shall be able to filter to RSVP’d events</p> <p>2.53 A registered user shall be able to set available events as ‘Maybe’</p> <p>2.54 A registered user shall be able to view their list of RSVP’d events</p> <p>2.55 A registered user shall be able to view their list of Interested events</p> <p>2.56 A registered user shall be able to view their list of Maybe events</p> <p>11. Public Events</p> <p>11.6 Public events shall be set as ‘Interested’ by a registered user</p> <p>11.7 Public events shall be set as ‘Maybe’ by a registered user</p> <p>12. Group Events</p> <p>12.11 Group events shall be set as ‘Interested’ by a group user</p> <p>12.12 Group events shall be set as ‘Maybe’ by a group user</p>

	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	<pre> sequenceDiagram actor Group User participant Favorite Events Group User->>Favorite Events: Create current Favorites list Group User->>Favorite Events: Save Event Group User->>Favorite Events: Search for past events gone to. </pre> <p>The diagram shows a use case box titled "Favorite Events" containing three use cases: "Create current Favorites list", "Save Event", and "Search for past events gone to.". An actor labeled "Group User" is connected to each of these three use cases by lines.</p>
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	<p>The diagram illustrates the interactions between two actors, Tom and Jack, and a set of use cases within a boundary labeled "Group Chat".</p> <ul style="list-style-type: none">Actors: Tom (blue head) and Jack (red head).Use Cases:<ul style="list-style-type: none">Post (blue oval)Upload Photo (blue oval)add comment (red oval)Add Emoji (blue oval with a dashed border)Relationships:<ul style="list-style-type: none">Tom is associated with Post, Upload Photo, add comment, and Add Emoji.Jack is associated with add comment and Add Emoji.A dashed arrow labeled <<include>> points from Upload Photo to Post.A solid arrow points from add comment to Add Emoji.
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 of 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 in case 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	<p>Events available with information on the front page.</p> <p>Easy interface set by different topics by age, location, activities</p>	<p>+Able to create groups, public/private events</p> <p>+Able to find groups, people, events pertaining to certain specifications(age, interests, friends)</p> <p>+Notification System</p> <p>+Messaging System</p> <p>+local guides: informational posts that can help fuel tourism.</p> <p>+Reporting System that can help mitigate against unsafe posts and members</p> <p>+Multiplatform(web, ios, android)</p>	<p>Large user base, offers a variety of different content and events</p>	<p>Local News and connection suggestions. Strong community engagement</p>	<p>Easy to learn, good onboarding, kid-oriented</p>
Weaknesses	<p>-restricted to specific area</p>	<p>Pricing restrictions</p> <p>Website requires no identity verification.</p>	<p>Too many ads</p>	<p>Restricted to surrounding area</p>	<p>Mobile only, swipe system means one match at a time, profiles may be far away, simple chat</p>
Pricing	<p>Free</p>	<p>Free-Tier</p> <p>Paid-Tier</p> <p>Option 1: \$14.99/month for a group</p>	<p>Free</p>	<p>Free</p>	<p>Free</p>

		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) presene 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 (myplaydate app)	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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Applied constructive criticism to brainstorming topics - Helped write use cases, diagrams, and requirements, -Helped productivity by taking availability-schedules, posting resources

		<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> -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