**Software Engineering CSC 648**

**Project Event Monkey**

**Section 1 Team 2**

Austin Ocampo - *Team Lead*

Micheal Hua - *Scrum Master*

Michael Maksoudian - *Back-End*

Sajan Gurung - *Back-End*

Robin Rillon – *Front-End*

Matthew Lee – *GitHub Master*

Milestone 2

October 19, 2022

**History Table**

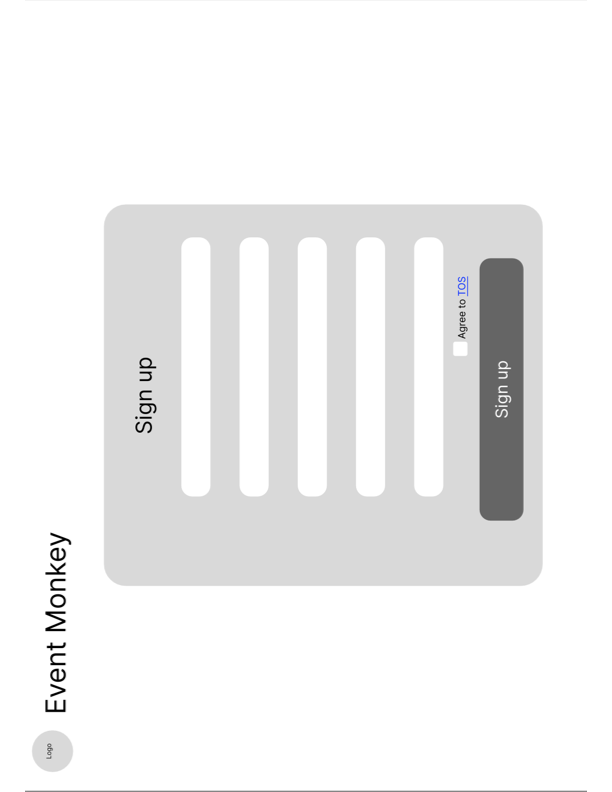
| **Date**: | **Revision Iteration:** | **Revision Notes:** |
| --- | --- | --- |
| Oct 19, 2022 | Original | N/A |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Data Definitions V2**

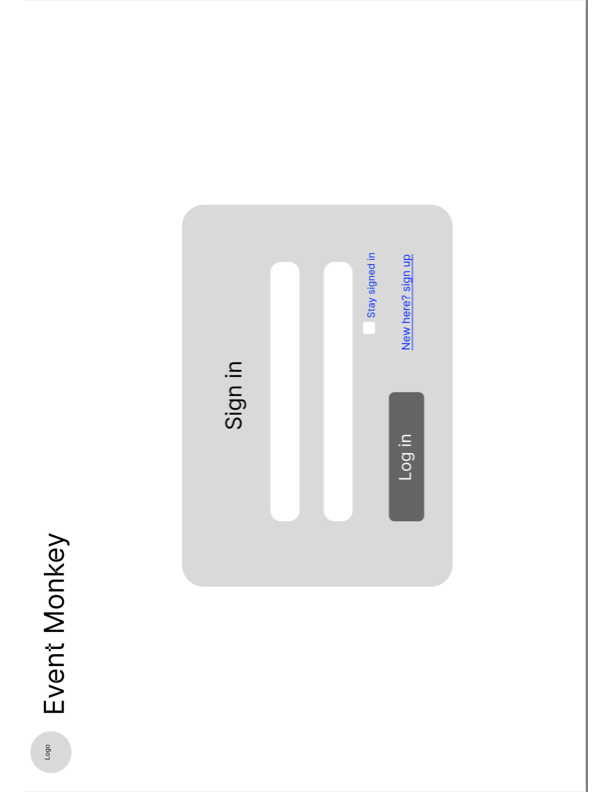
| **Main Data Item** | **Sub-Items** | **Description** |
| --- | --- | --- |
| Event | * name * description * dates * priceRanges * images * classifications | A container holding event details |
| Classification | * Segment   + the primary genre for an event. This genre is broad (Music, Sports, Arts, etc) * Genre   + the secondary genre for an event. This genre is more specific (Rock, Classical, Animation, etc) * Subgenre   + the tertiary genre of an event. This genre adds more detail to the *Genre* (Alternative Rock, Ambient Pop, etc) | A container holding a *Segment*, *Genre*, and *Subgenre* |
| Image | * ratio * width * height * url | Meta-data for url based images used for events and user profile pictures |
| Notification | * date * title * description | A message created for a *User* |
| User | * type (‘ORGANIZER’ or ‘ATTENDEE’) * email * password * username * profileImage * notifications | An abstract type to identify a person using the app. A user can be an *Organizer* or an *Attendee* |
| Organizer | In addition to *User* sub-items,   * ownedEvents | A type of user who can publish *Events* |
| Attendee | In addition to *User* sub-items,   * interests * favorites | A type of user who can search and add *Events* to their favorites list to attend |

1. **Functional Requirements V2**
2. Login page
3. Sign up page
4. Edit profile button
5. Change profile picture
6. Change description
7. Home page
8. Add events
9. Search events
10. Filter/sorting section
11. Sections for popular types of events
12. In homepage, scroll left and right recommended events
13. Friend
14. Notification tab
15. Send events to friends
16. Save events
17. Favorites list
18. Calendar
19. Upcoming events tab
20. Mobile alert
21. Light/dark mode
22. Help page
23. Sign up/ Sign in (Priority: high, user story id: 3)
    1. If the user would like a personalized experience, the user should be able to sign up and log in to personalize their profile
24. Make an event (Priority: medium, user story id: 1,2)
    1. To make an event, users must register and be registered as an event host to create events.
25. Modify an event (Priority: medium, user story id: 1,2)
    1. If anything changes regarding the event, users listed as an event host should be able to modify and update any changes such as whether or not an event has been canceled, if the time of the event has changed, any age requirements for the event, and much more.
26. Participate in events (Priority: low, user story id: 3,4)
    1. To participate in some events that are free and open for everyone, users should register with their name, email address, or some way to contact them about the event.
27. Customizing user profile (Priority: high, user story id: 1)
    1. Users should be able to customize their profiles to their fullest extent. Contents such as bio, description, and common interests should be filled out to better analyze their type of events.
28. Searching events (Priority: medium, user story id: 3,4)
    1. Everyone including registered users and guests without credentials should be able to search for events. Searches can be very specific or very broad. For example, users can search something like “sports” or “football” or “49ers” or “49ers vs Seahawks”
29. Popular categories on top bar (Priority: medium, user story id: 1)
    1. The top bar will have popular categories so that it’s easier for users to specifically check out any events currently happening or that will be upcoming within the category. If the users select “Sports”, they should be able to find all events within a date range. There will also be categories within that will specify it even further. Using the example, if the user wants to check out events for basketball or hockey, they could click on the categories matching the title.
30. Customized recommended section on home page (Priority: low, user story id: 1)
    1. Just like Netflix, based on the users’ interest or the events that users have interacted with, the home page will make a recommended section that users can scroll left to right to search for new events that they could be interested in.
31. Calendar/ Scheduler (Priority: low, user story id: 1)
    1. Users should be able to view a calendar that lists upcoming events that the user will be attending. This will allow the user to be organized and make sure that nothing overlaps the events
32. Notifications (Priority: medium, user story id: 3)
    1. Users should be able to have access to a notifications tab on the top bar so that they could get recommendations or receive notifications for an event they would be attending as a reminder and so that any changes made by the event host would also be notified by the attendees.
33. Favorites (Priority: medium, user story id: 4)
    1. Users should be able to save and favorite events so that if they would like to register for the event again, they could easily find it on their favorites page.
34. Help page (Priority: high, user story id: 1)
    1. If anyone has trouble accessing the website, anyone could go to the help page and get assistance there.
35. Event Randomizer (Priority: low, user story id: 3)
    1. For anyone who is ambitious, adventurous, and would like to participate in an event that’s out of their comfort zone. This event randomizer will help with that. Just as how google has an “I’m feeling …” feature, the event listener will be the same, but randomizing categories of events
36. Friends (Priority: 3)
    1. With a social media-like friends system, users should be able to send events to friends and see if they’re interested to go as well. These friends could be actual friends or those who share similar interests as the user.
37. Sign-up/Sign-in - (priority: 1)
    1. Sign-up page (priority: 1)
       1. Directs to a page that asks for interests (priority: 2)
    2. Sign-in page (priority: 1)
       1. Keep me logged in checkbox (priority: 3)
       2. Sign in with Google, Apple, etc… (priority: 3)
       3. New to website that directs to sign-up page (priority: 2)
    3. Forgot password link (priority: 1)
       1. Maybe a page that asks verification questions (priority: 2)
    4. Forgot username (priority: 3)
38. Search Events - (priority: 1)
    1. Related results will show based on search (priority: 2)
    2. Filter search by categories (like Amazon) (priority: 3)
39. Adding Events - (priority: 1)
    1. If not signed in, directs to sign in page (priority: 1)
    2. Populate information (priority: 1)
       1. Event name (priority: 1)
       2. Location (priority: 1)
       3. Date and time (priority: 1)
       4. Host/company (priority: 1)
       5. Pricing for different seats (priority: 2)
    3. Verify with event host/company (priority: 3)
40. Customizing User Profile - (priority: 1)
    1. Text box to fill description section (priority: 1)
    2. Change username/nickname (priority: 1)
    3. Change password, but must enter the old password (priority: 1)
    4. Change any interests (priority: 1)
    5. Change email/ phone number. Sends verification message (priority: 2)
    6. 2 factor authentication? (priority: 3)
    7. Upload profile picture (priority: 2)
41. Favorites - (priority: 2)
    1. Create collections (priority: 1)
       1. add/ remove events from collections (priority: 2)
    2. Remove event from favorites (priority: 1)
    3. Undo? (in case accidentally removes event) (priority: 3)
    4. Filter by: (priority: 1)
    5. Sort by: (priority: 1)
    6. Default categories? (priority: 3)
42. Notification - (priority: 1)
    1. Delete notifications (priority: 1)
    2. Click on the notifications title header to have it drop down for more info (priority: 1)
    3. Search notifications (priority: 1)
    4. Sort by: (priority: 1)
    5. Click on notification to direct them to the content (priority: 2)
43. **UI Mockups and Storyboards (High Level Only)**

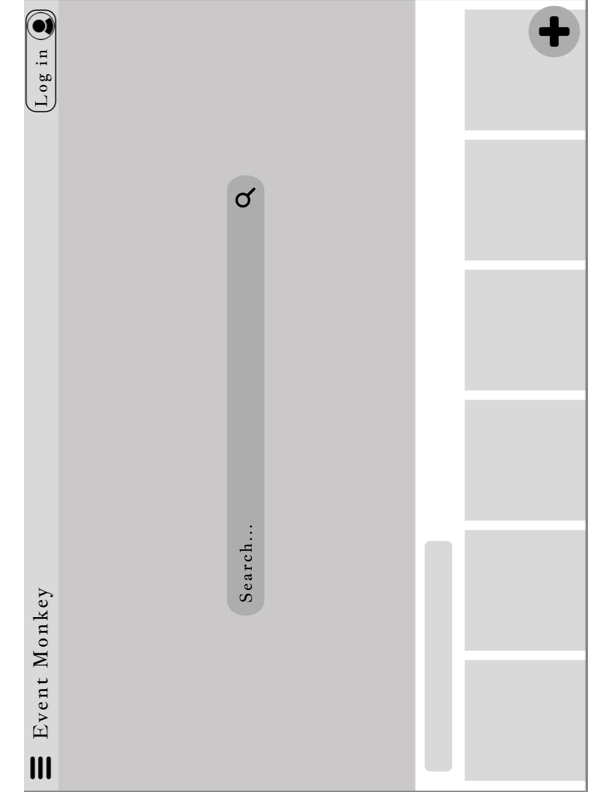
**Sign Up Page:**



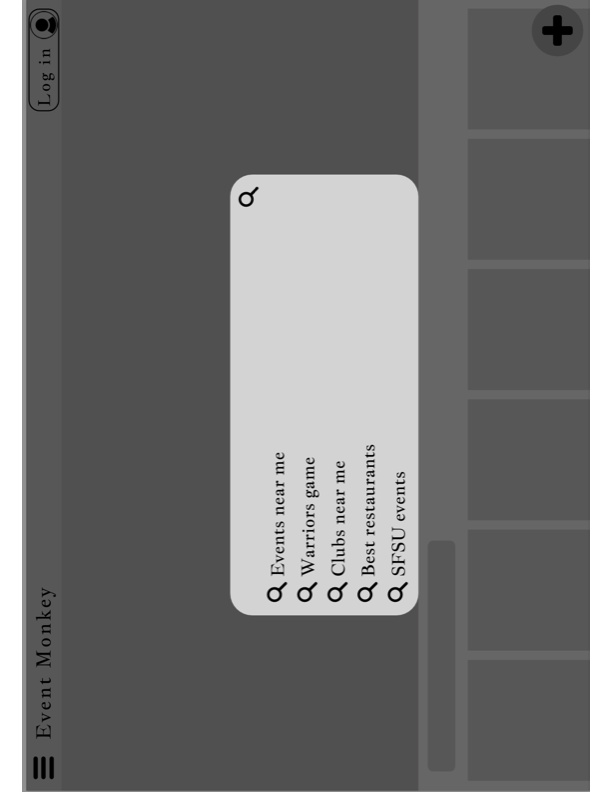
**Sign In Page:**

****

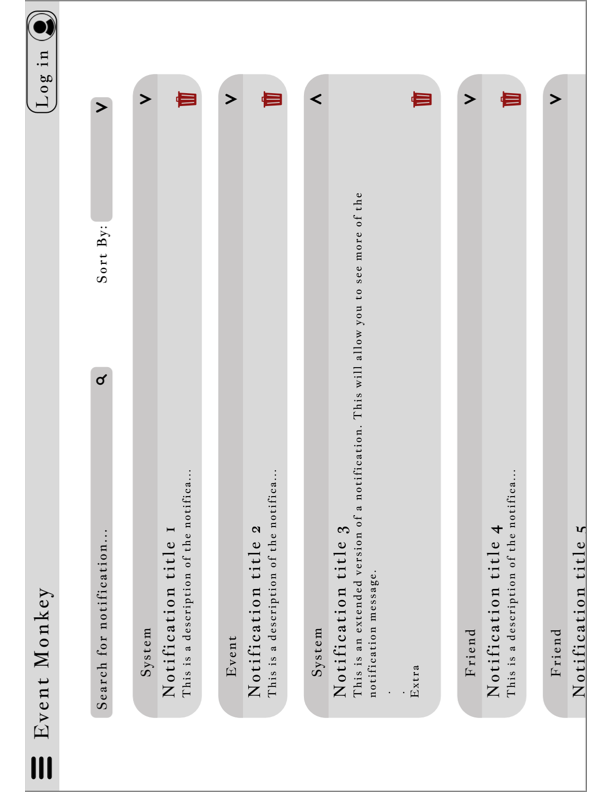
**Homepage:**

****

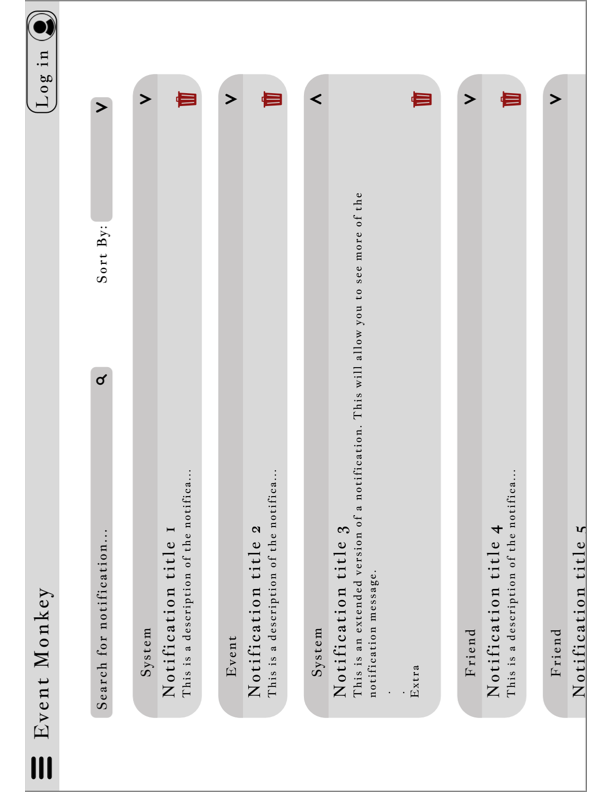
**Search Events:**

****

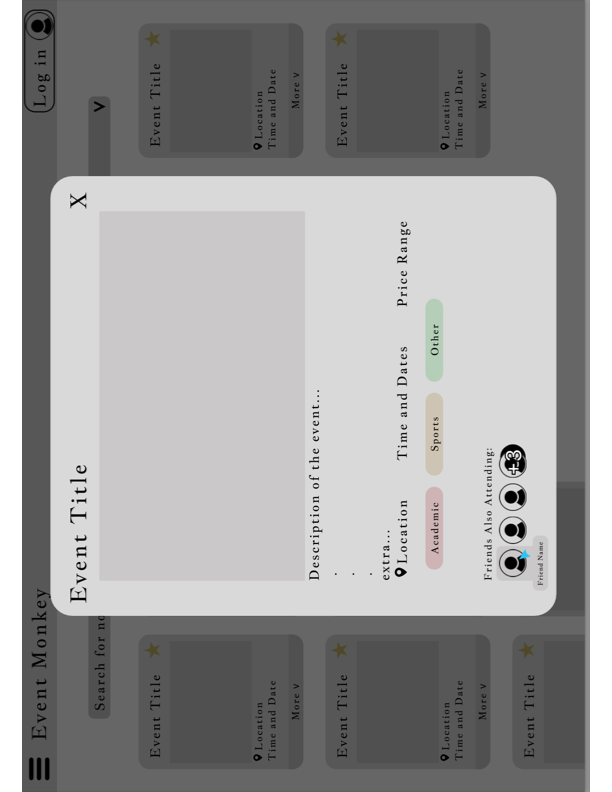
**Notifications:**

****

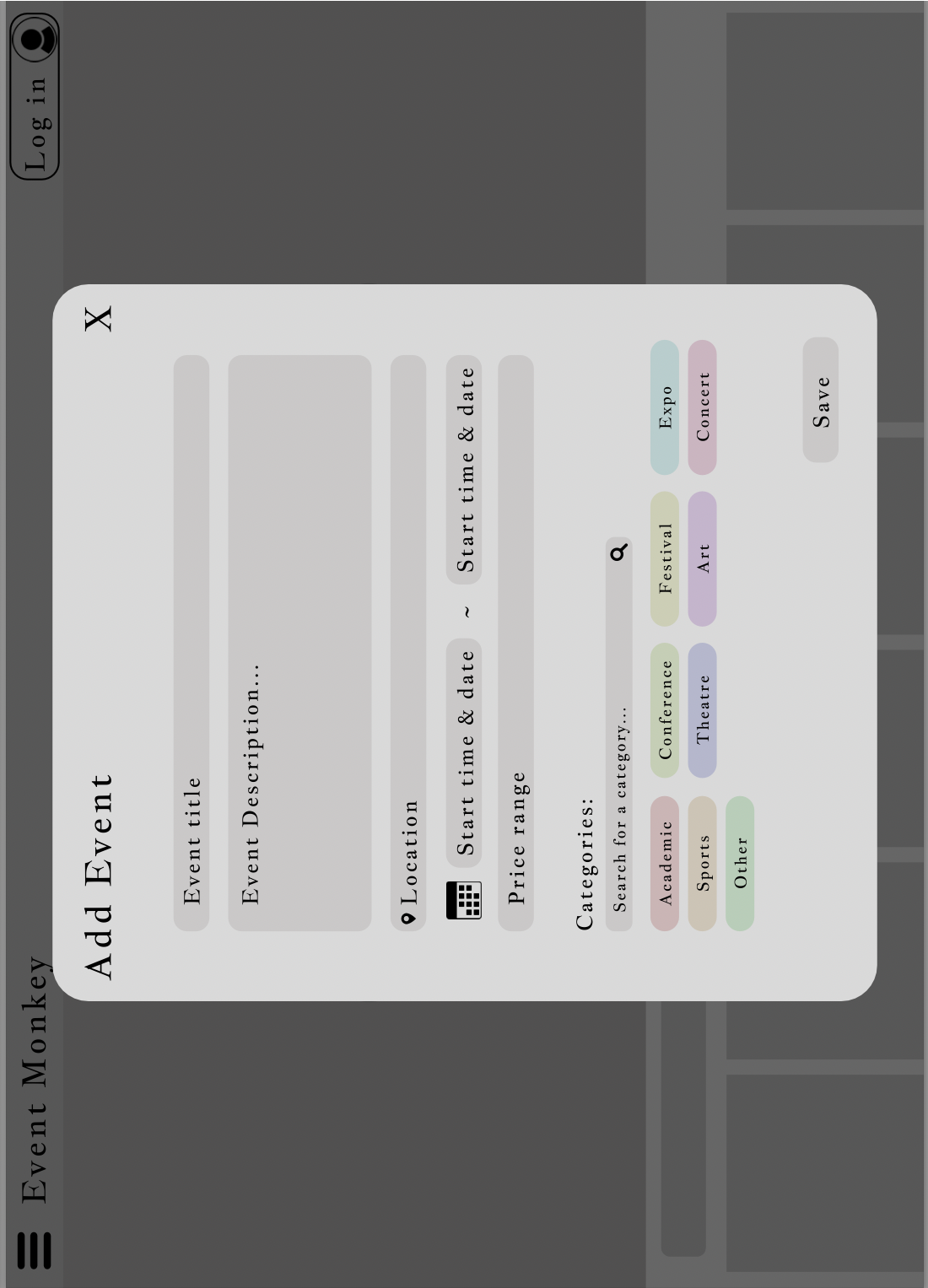
**Favorites:**

****

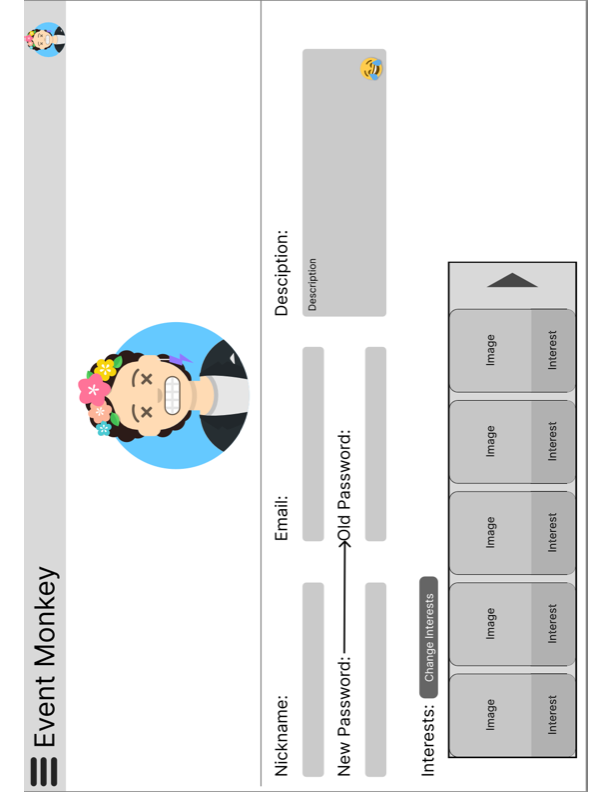
**Event Page:**

****

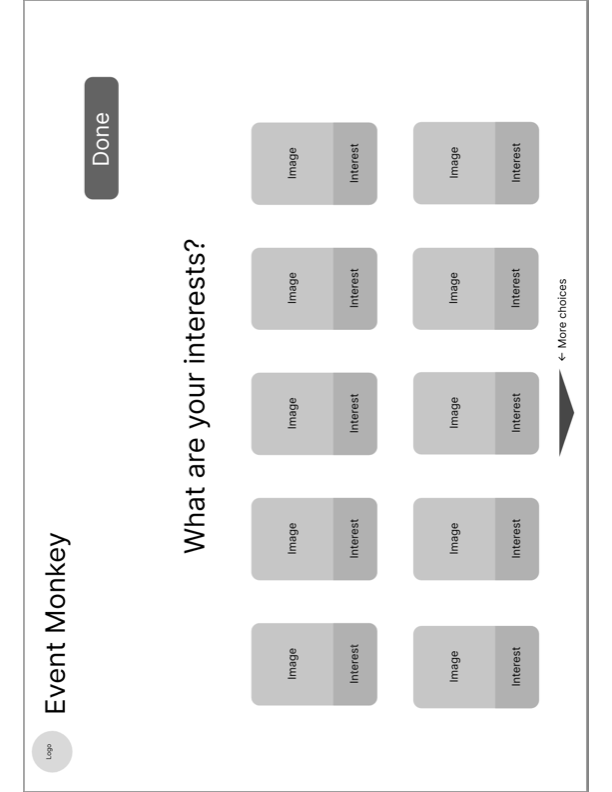
**Add Event:**



**Customize User Profile:**

****

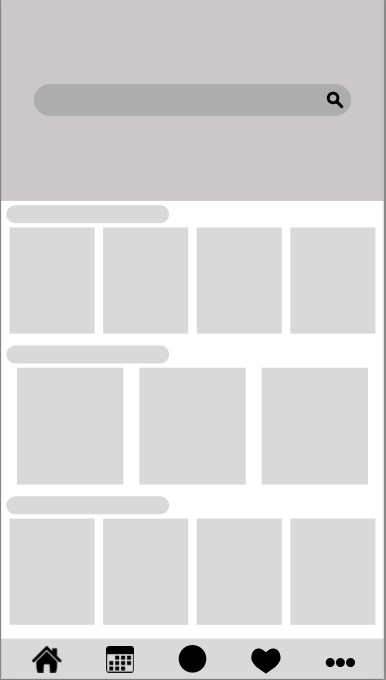
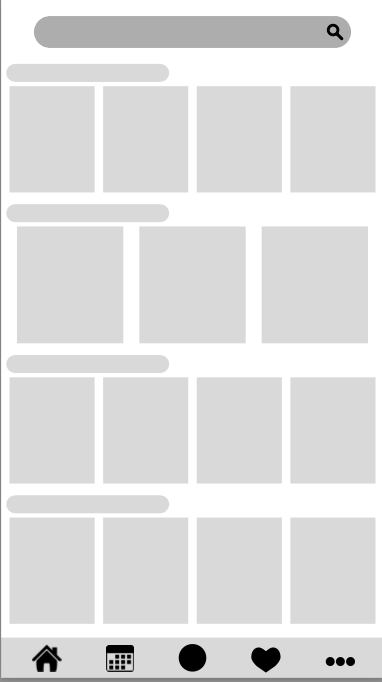
**Interests:**

****

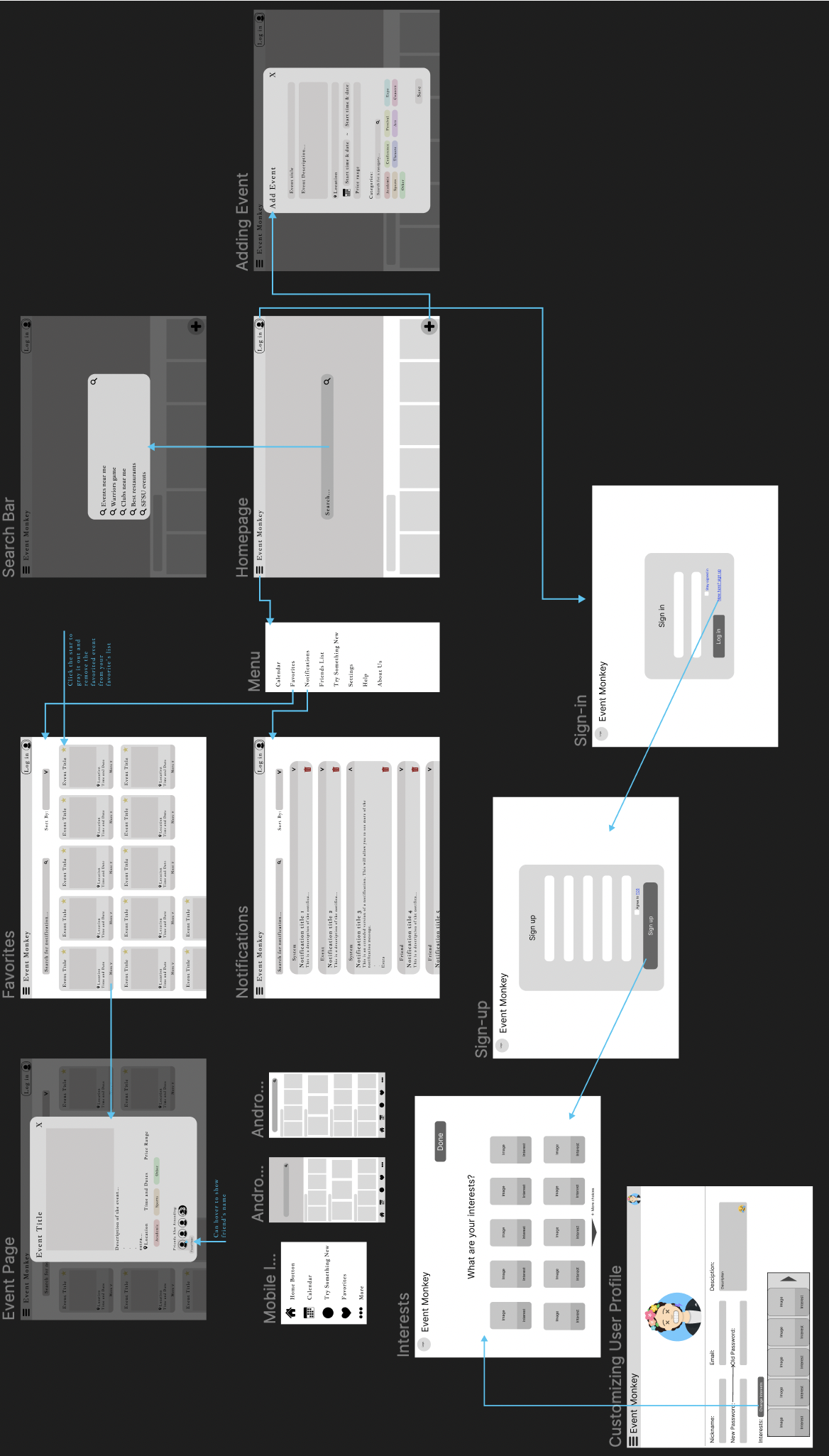
**Mobile Icons:**



**Android Banner/No Banner:**

**UX Flowchart:**



1. **High Level Architecture, Database Organization**

Database Organization

Graphical user interface, text, application, chat or text message, Teams

Description automatically generated

| **Table Name** | **Add / Delete / Search Architecture** | **Functional Requirement** |
| --- | --- | --- |
| Event | Add / Delete / Search | Searching for events by its attributes; Organizers can add events |
| Event\_List | Add / Delete | Events favorited by attendee user types are added/deleted  Events created by organizers are added/deleted from this list |
| User | Add / Search | Sign-up will add a user; Sign-up and sign-in will search for existing users |
| Attendee\_Interest\_List | Add / Delete | Attendee interests can be added and removed as a profile customization option |
| Notification | Add / Delete / Search | Notifications are added for both user types and can be searched through/deleted by them |
| Image | Add / Delete | Adding events; Customizing user profile |

**API**

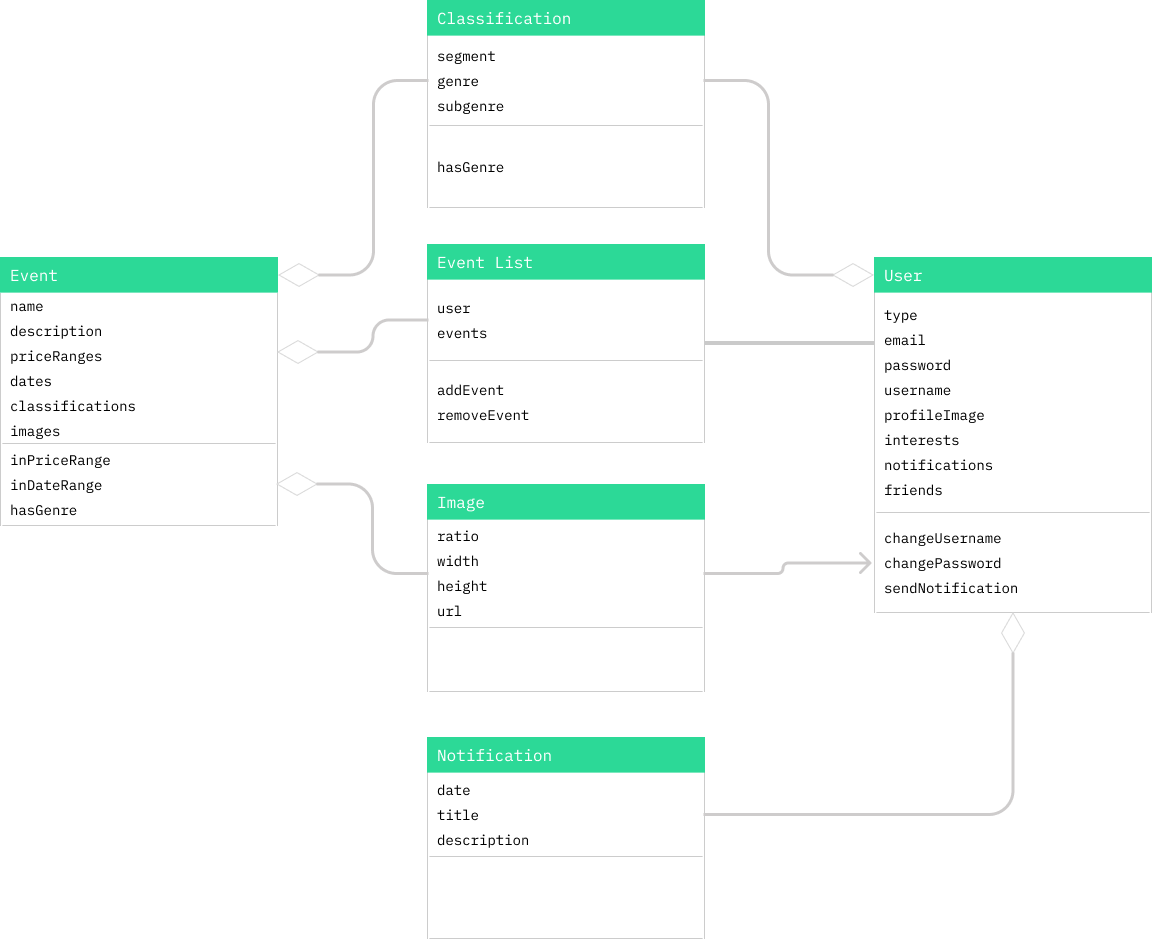
The 3rd party api we are using is TicketMaster. This api provides event details and can find suggestions based on an array of genres. The api can also provide classification details and relevant images. The api is intended to be used in our project when searching for events and when finding recommended events for users.

Our Express server will provide routes for http requests sent by the React client. The client will send requests through http for signing up, signing in, event searches, and account management.

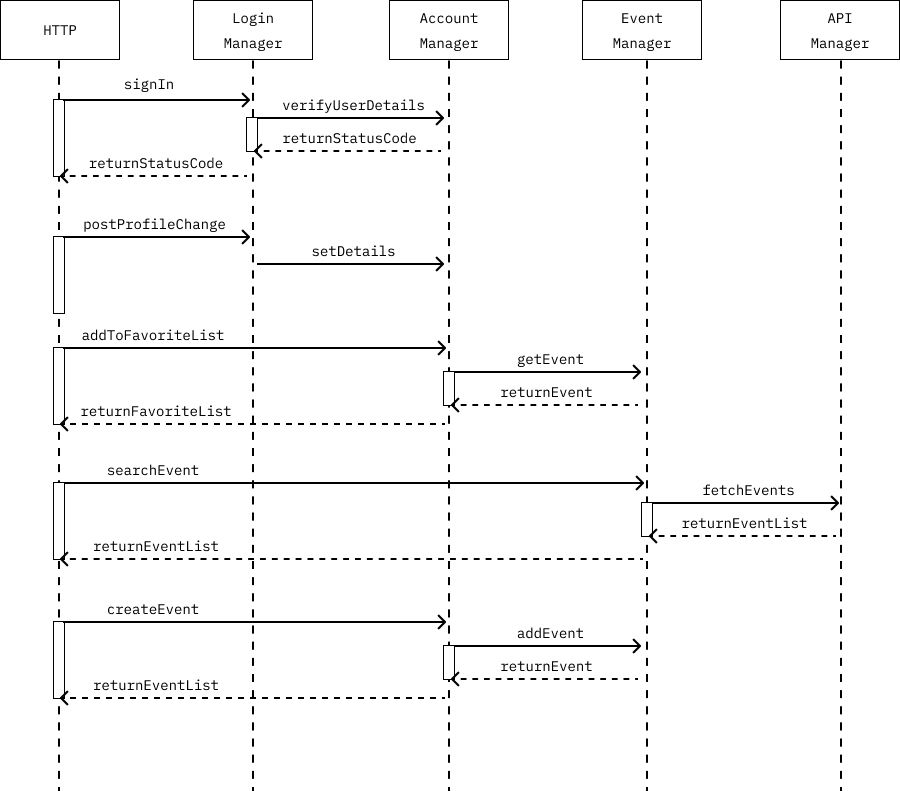
| **Http API Request** | **Description** |
| --- | --- |
| /api/event | Requests a list of events from our database combined with TicketMaster results. This api request is intended to have parameters such as ‘genre’ and ‘title’ to filter the list. |
| /api/event/create | Requests to add an event to our database. Http content type is intended to be in JSON format with event details. |
| /api/user | GET requests user data from our database. Parameters can be added to the request for getting specific user data. PUT will request to modify user data. Http content type is intended to be in JSON format |
| /api/user/signin | A sign-in request, with required account details as parameters. |
| /api/user/signup | A sign-up request, with required account details as parameters. |

1. **High Level UML Diagrams**

Class Diagram



Sequence Diagram



1. **Identify actual key risks for your project at this time**

Skill Risk:

A skill risk we have encountered is integrating our front-end to our database using React. We ran into issues connecting the react client to our API and have been researching methods to correct the issue. We have many resources such as articles from Free Code Camp and StackOverflow on how to connect the two together which has been helpful.

* Schedule risks
  + If a teammate has something going on that collides with the proposed meeting time but doesn’t tell the team, it might cause delays and teammates to wait for a response back if their code also has to do with the person missing.
    - Resolution:
      * Communicate with the team about missed days and report back ASAP
* Teamwork risks
  + If several teammates are working on the same section of the code, it can cause confusion and delay the scheduled times.
    - Resolution:
      * Let other teammates know what part is being done so the team knows what part not to code
  + Miscommunication between the front-end and back-end leaders can cause the final product to not mesh well and the quality of the product will fall
    - Resolution:
      * Hold meetings every week to keep track of each teammate’s progress
      * Ask questions if necessary

Legal/Content Risks:

Legal and content risks we may run into is copyright on images such as our Curious George monkey and also when we pull events such as NBA games. The NBA logo as well as other sport organization logo might be copyrighted so we must do further research to ensure we are using company and organization logos properly and legally.

1. **Project Management**

During Milestone 2 our team was able to adequately assess and implement the tasks assigned to them. Our plan consisted of the initial dissemination of tasks that needed to be accomplished by the end of Milestone 2. Our biweekly meetings (in addition to class meetings) were used to update the team on the statuses of each assigned task as well as provide a time to discuss any issues that have come up since the last meeting. Our tasks were managed using our Scrum Board which is shared via Notion. It shows what tasks have been assigned as well as the current progress status of each. Our team has found that it has been significantly helpful to have two methods of tracking progress. The Scrum Board serves as a platform where each member can visually see the progress made and the weekly meetings are helpful as it allows each lead to elaborate more on the progress made or issues encountered. Overall, the team has been outstanding in the way they have managed and conducted their assigned roles.