Pick-Up Sportz Sprint #1 Presentation

Chaz Del Prato, John Him,
Benjamin Seo, Jamil Khan,
Brandon Le, Christine Duong

Product Requirements Document

Goals

What is the purpose of this project?
The purpose of this project is to develop a mobile application for users to play a sport of their choosing with other people by connecting with others through the app.

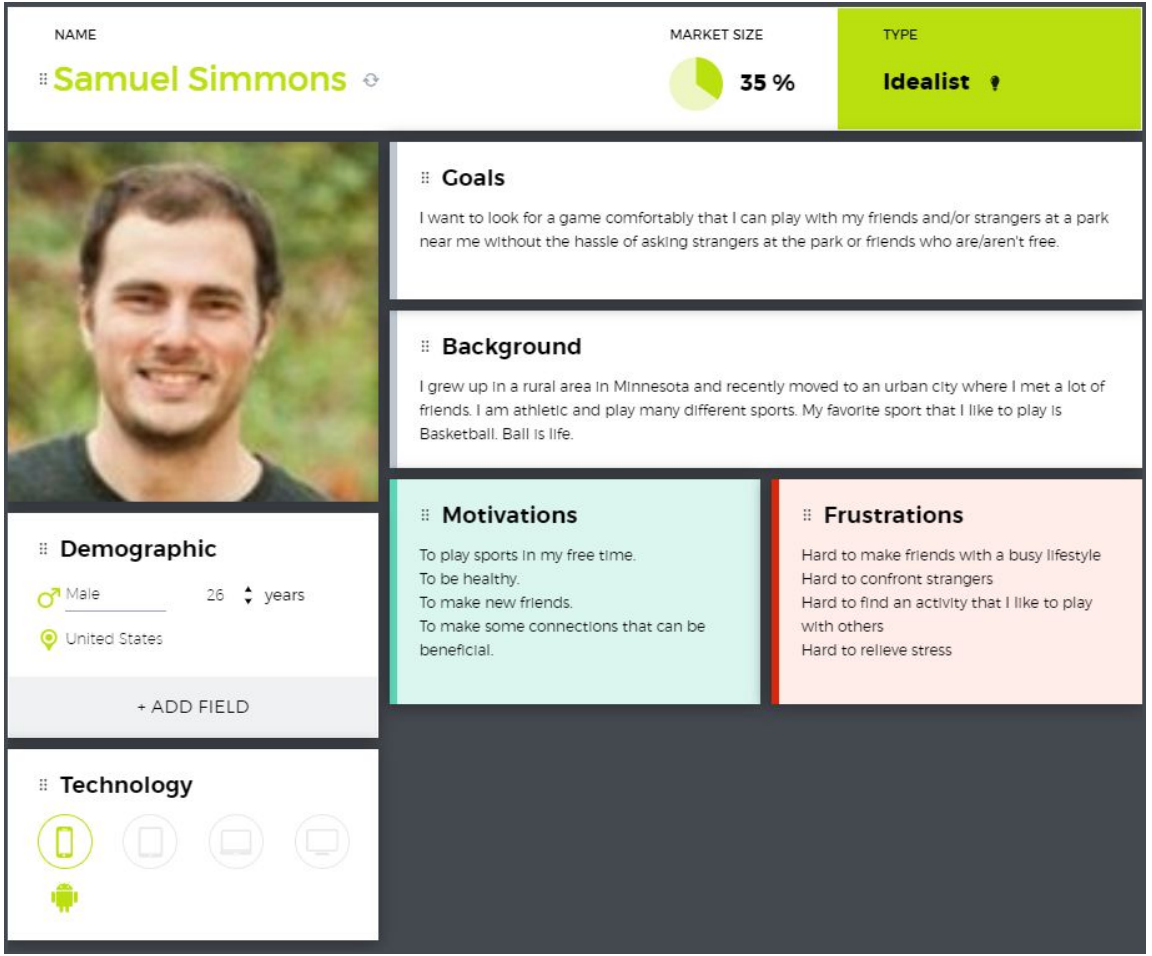
What are the problems it will solve?
The product of this project will solve user's problems such as not having anyone to play sports with, not being able to find areas for a specific sport, or even not having the required equipment to play a game.

How will it streamline or improve the current process or facilitate a new process?
With the Pick-Up Sportz Application, it eliminates the guessing normally required to determine if others are there and will allow the user to play any sport according to their schedule, instead of waiting for other people to be free. The application will also help users that do not have any equipment to play a sport, and link them with other users that do have equipment to start a game.

What is the product vision?
This product will fill the void in looking for a pick up game. The app will not only serve as a way for people to connect over sports in person, but be a mix between social media and real life interactions. Eventually through leagues set up by the app, amateur local leagues will become synonymous with the leagues run by Pick Up Sportz wherever league organized sports are played.

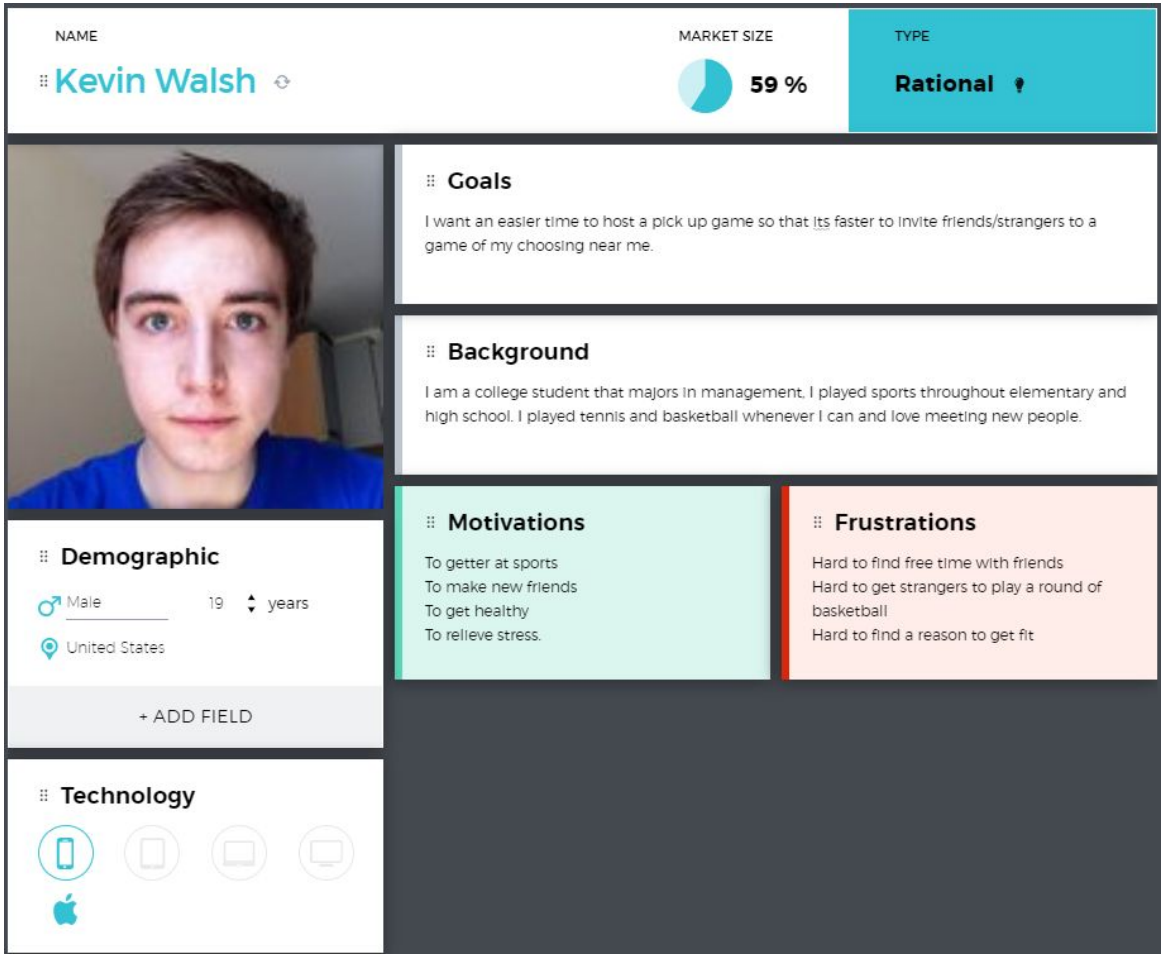
Persona 1

Samuel Simmons is a 26 year old male who grew up in rural Minnesota and recently moved to a new urban city, where he met many of his new friends. He enjoys playing basketball, as it is his favorite sport. He is motivated to be healthy, make new friends, and play sports in his free time. He has difficulty confronting strangers, finding an activity to play with others, and relieving his stress. Given his busy lifestyle, he is also frustrated in his attempts to make new friends. His goal is to look for a game comfortably, so that he can play with his friends or strangers at a park near him, without hassling them.



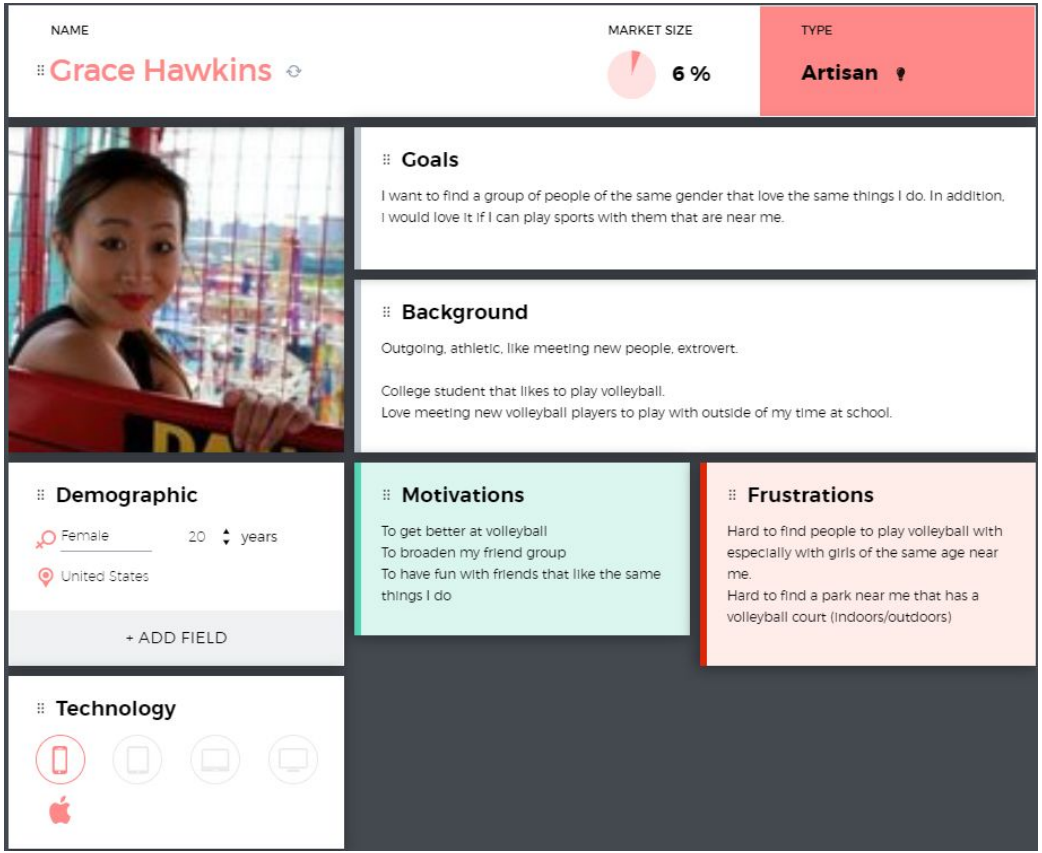
Persona 2

Kevin Walsh is a 19 year old male college student currently majoring in Human Resource Management who loves to meet new people. He has played sports throughout elementary and high school. His favorite sports to play are tennis and basketball. He is motivated to improve at sports, make new friends, be healthy, and relieve stress. He has difficulty finding free time to spend with his friends, getting strangers to play a round of basketball with him, and finding a reason to get in shape. His goal is to have an easier time to host a pickup game so that it is faster to invite friends or strangers to a game of his choosing near him.



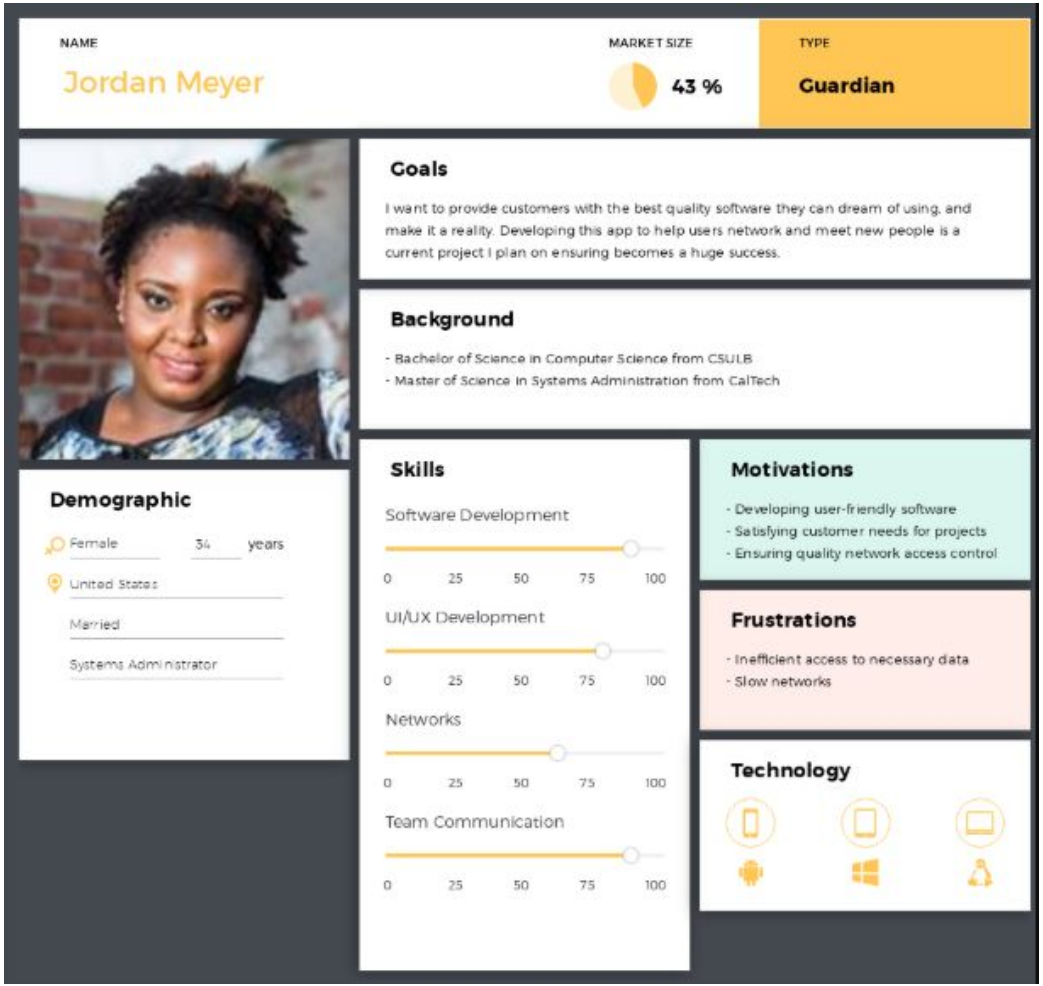
Persona 3

Grace Hawkins is a 20 year old female college student majoring in Molecular Biology, who likes to play volleyball. She is outgoing and athletic. She also likes to meet other volleyball players who she can play volleyball with outside of school. She is motivated to get better at volleyball, broaden her friend group, and have fun with friends who have similar interests. She has difficulty finding a nearby park with indoor/outdoor volleyball facilities, and finding other girls near her age with whom she can volleyball. Her goal is to find a group of people of the same gender that share her interests, and would also like to play sports at a place near to her.



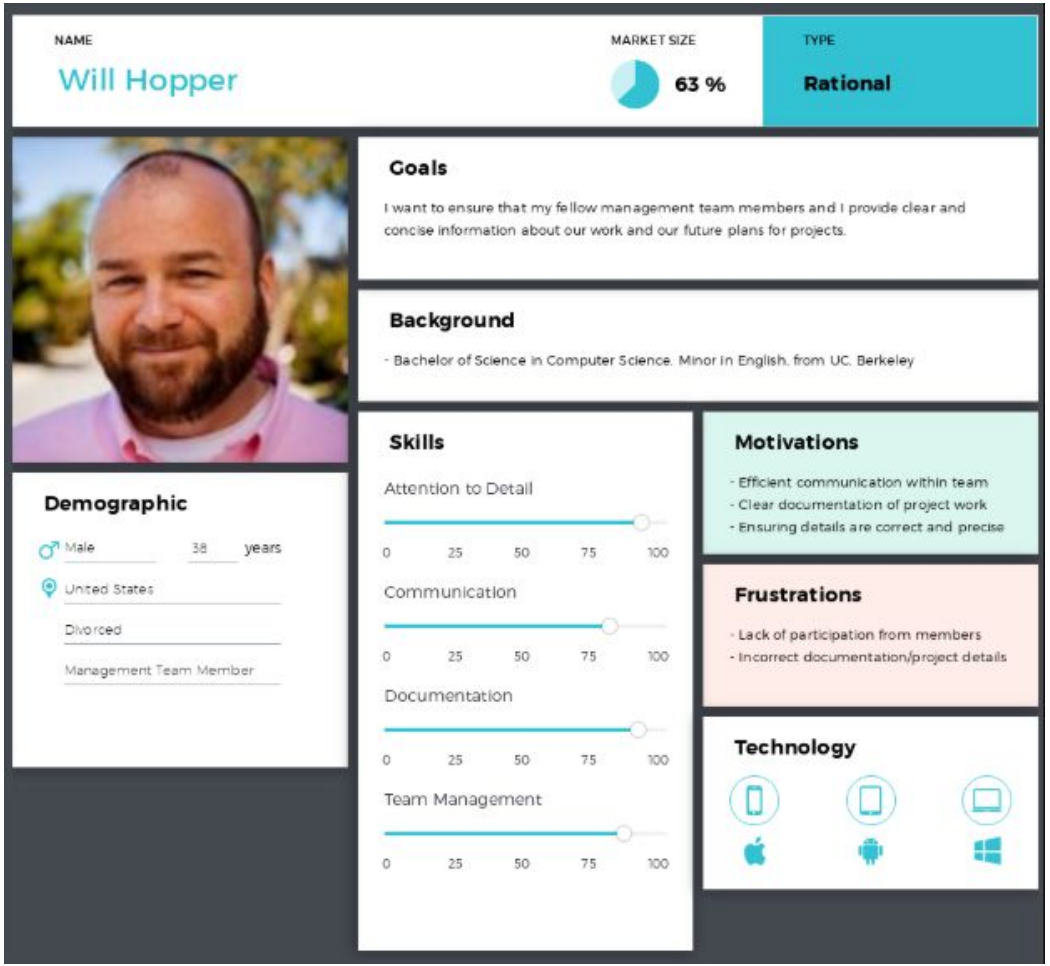
Persona 4 (updated on BRD)

Jordan Meyer is a 34 year old Systems Administrator who works within Amazon's Systems Administration Department. She is skilled in software development, UI/UX development, networks, and team communication. Her goal is to provide customers with the best quality software they wish to have, and make it a reality. She attended CSULB for a Bachelor of Science in Computer Science, as well as CalTech for a Master of Science in Systems Administration. She is motivated to develop user-friendly software, to satisfy customer needs, and to ensure quality network access control. She is frustrated by inefficient access to necessary data and slow networks. She uses technology such as Android, Windows, and Linux.



Persona 5(updated on BRD)

Will Hopper is a 38 year old Management Team Member working within Google's App Development Department. He is skilled in communication, documentation, and team management. He also has a high attention to detail. He wants to ensure that he and his fellow team members provide clear and concise information about their current work and their future plans for projects. He graduated from UC: Berkeley with a Bachelor of Science in Computer Science with an English minor. He is motivated by efficient communication within his team, clear documentation of current project work, and correct and precise details about current work. He is frustrated by a lack of participation by other team members, and incorrect documentation about project details. He uses technology such as iOS, Android, and Windows.



User Stories

Epic 1: As a management team, we want to be able to have the required documents for a software development process, so that our product is
successful.

- **SRP1:** As a member of the management team, I want to be able to have a Business Requirements Document, so that I can describe the characteristics of our proposed system.
- **SRP2:** As a member of the management team, I want to be able to have a Management Plan, so that I can understand how the project is executed, monitored, and controlled.
- **SRP3:** As a member of the management team, I want to be able to have an interactive Sprint Board, so that we can determine tasks that need to be completed for the current sprint.
- **SRP4:** As a member of the management team, I want to be able to have a Product Requirements Plan, so that we can understand the requirements for the product.
- **SRP5:** As a member of the management team, I want to be able to have an Architectural and Design Documents, so that we can understand the structure and layout of our product.

User Stories Cont.

Epic 2: As an administrator, we want to be able to have a backend, such as a server, database, and Machine Learning algorithms, working in synergy, so that our product performs at the highest level of design.

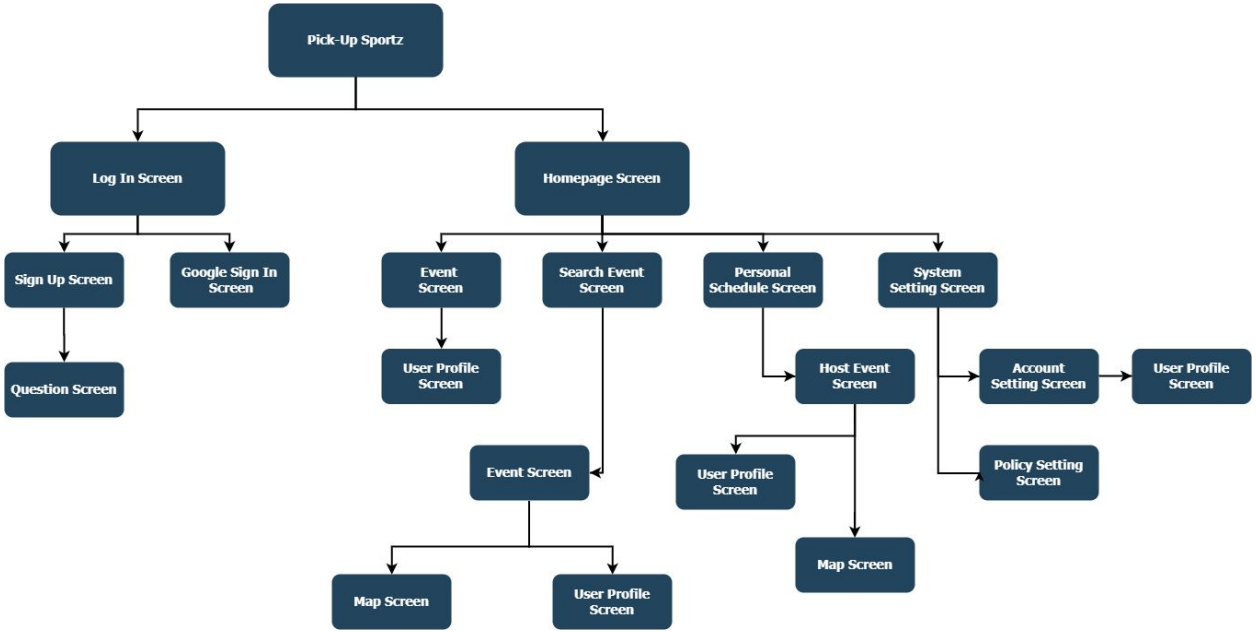
- **SRP1:** As an app admin, I want to be able to communicate with a database, so that I can keep records of all information within the app.
- **SRP2:** As an app admin, I want to be able to have our database communicate with our application, so that our database can start to save the users information.

User Stories Cont.

Epic 3: As a player, we want to be able to have an application that will allow us to create pick-up games for any sport and allow strangers to join our games, so that we can have fun, network, and exercise while playing our favorite sports.

- **SRP1:** As a player, I want to be able to create a profile that shows my age, skill level, and preferred sports, so that I get notified when a related game is started.
- **SRP2:** As a player, I want to be able to use a user's manual, so that I understand how to use the application.

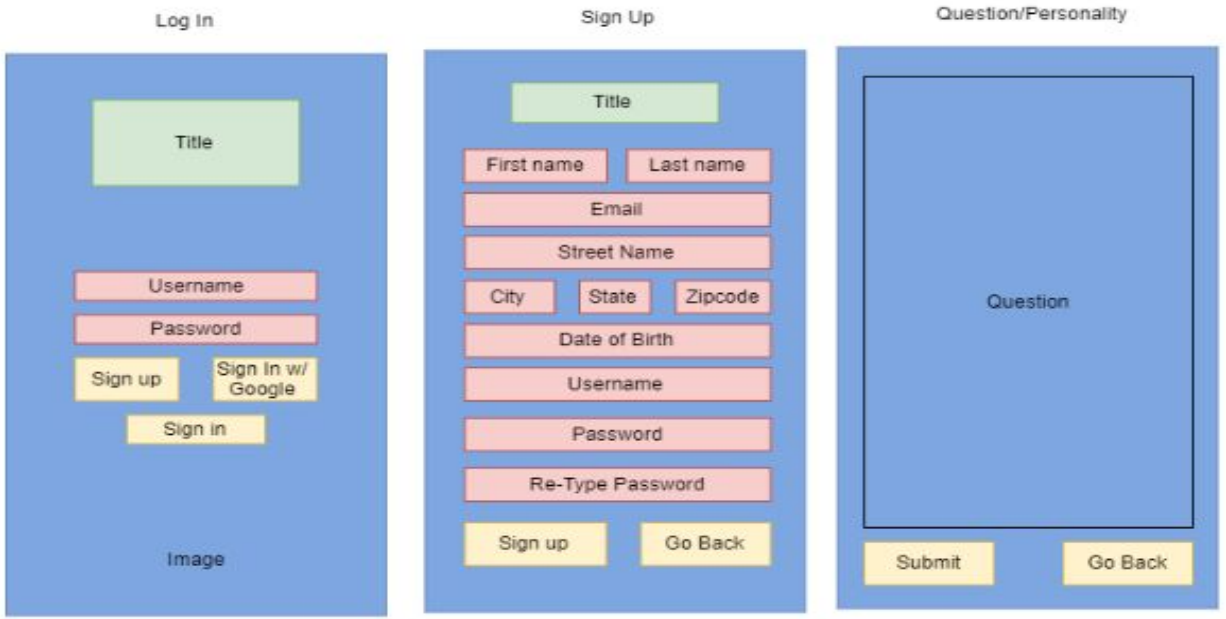
Server Sitemap

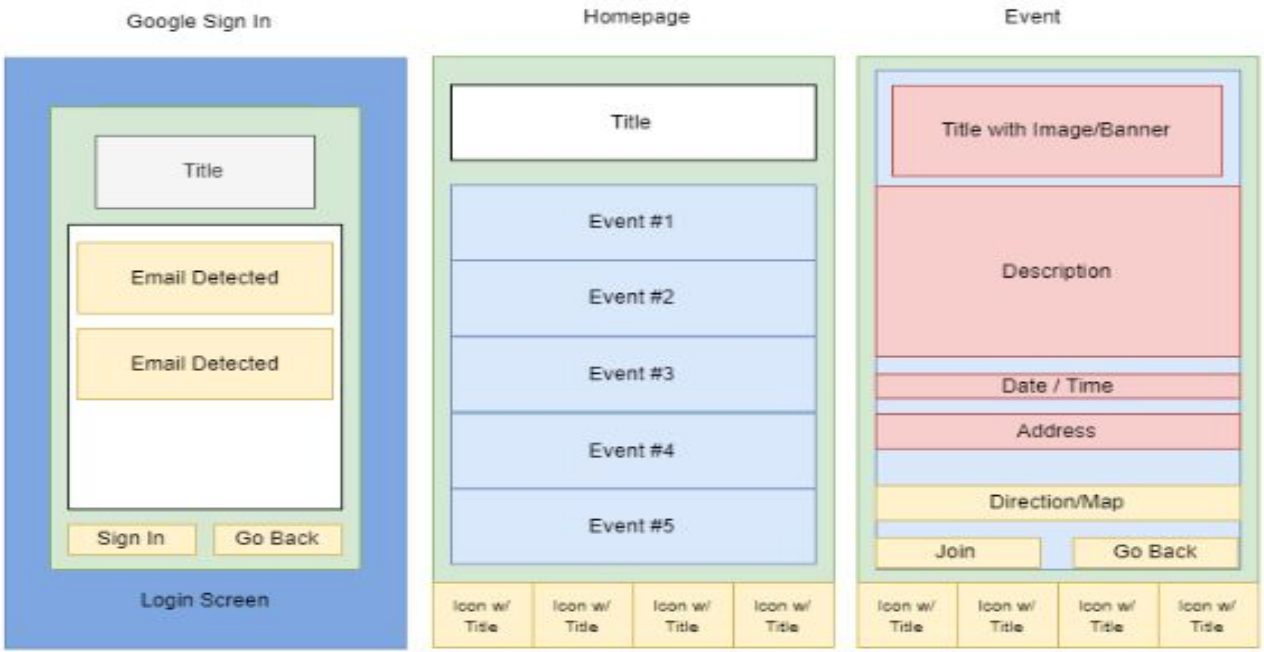


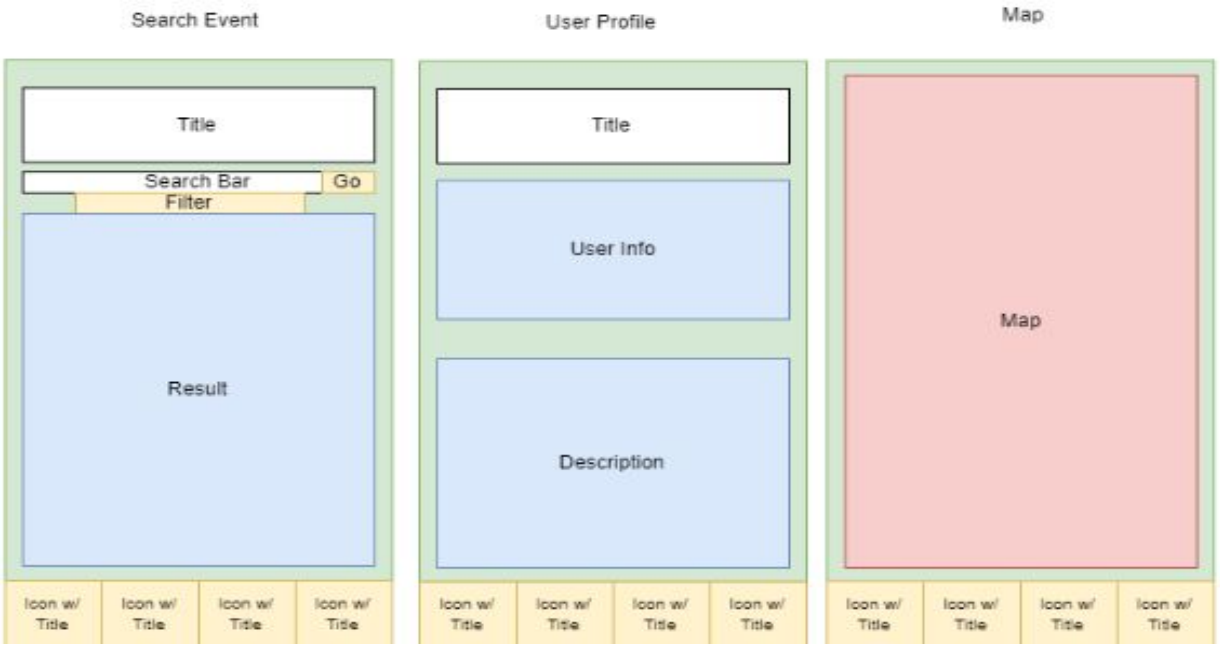
Page Description

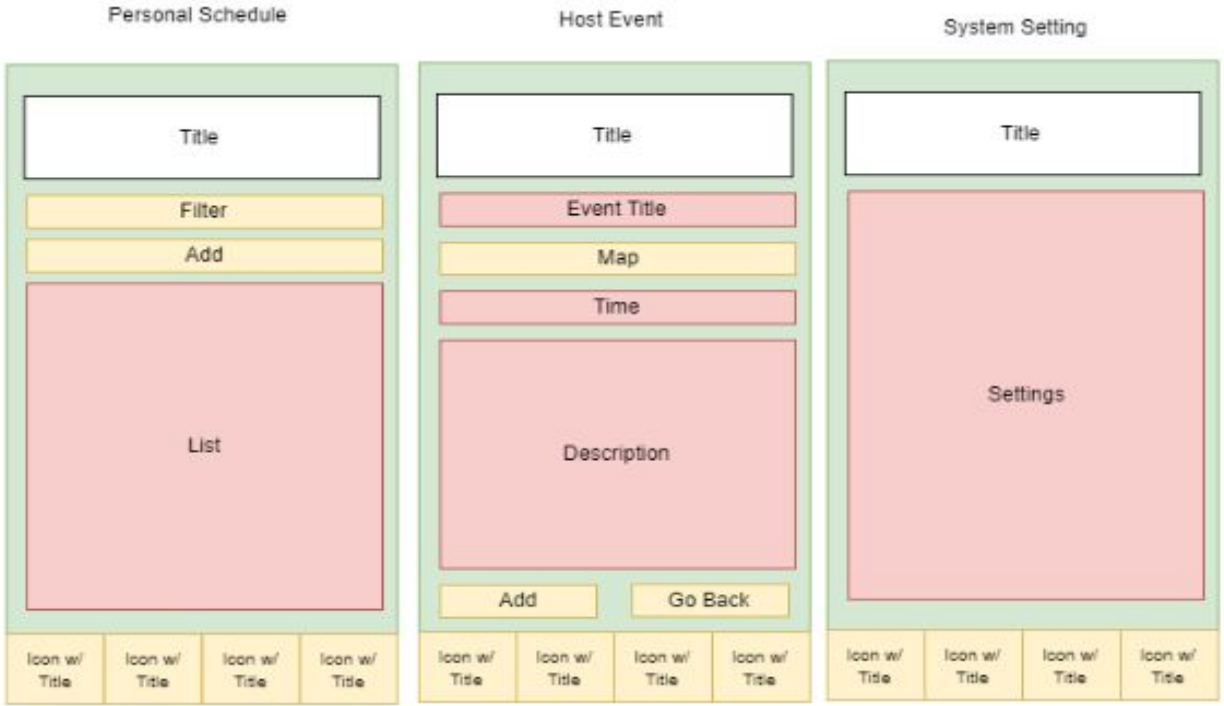
Page	Elements
Login	1. Title 2. Username 3. Password 4. Sign Up button 5. Google Sign In Button
Sign Up	1. Title 2. First name 3. Last name 4. Email 5. Street Name 6. City 7. State 8. Zip Code 9. Date of Birth 10. Username 11. Password 12. Sign Up button 13. Cancel/Back button
Question/Personality	1. Questionnaire 2. Submit button 3. Back button
Google Sign In	1. Title 2. Emails Detected 3. Sign In button 4. Back button
Homepage	1. Title 2. Events list
Event	1. Title with banner/image 2. Description 3. Date/time 4. Address 5. Directions/Map button 6. Join button 7. Back button
Search Event	1. Title 2. Search bar 3. Go to Event Details button 4. Filter button 5. Result list
User Profile	1. Title 2. User info 3. Description
Map	1. Geographical Image of Location
Personal Schedule	1. Title 2. Filter button 3. Add button 4. List of events in schedule
Host Event	1. Title 2. Event Title 3. Map button 4. Event Details (Time, Location, etc.) 5. Description 6. Add Event button 7. Back button
System Setting	1. Title 2. Settings Options
Account Setting	1. Title 2. Update First name 3. Update Last name 4. Update Email 5. Update Street name 6. Update City 7. Update State 8. Update Zip code 9. Update Date of birth 10. User profile button 11. Back button
Policy Setting	1. Title 2. Policy 3. Back button

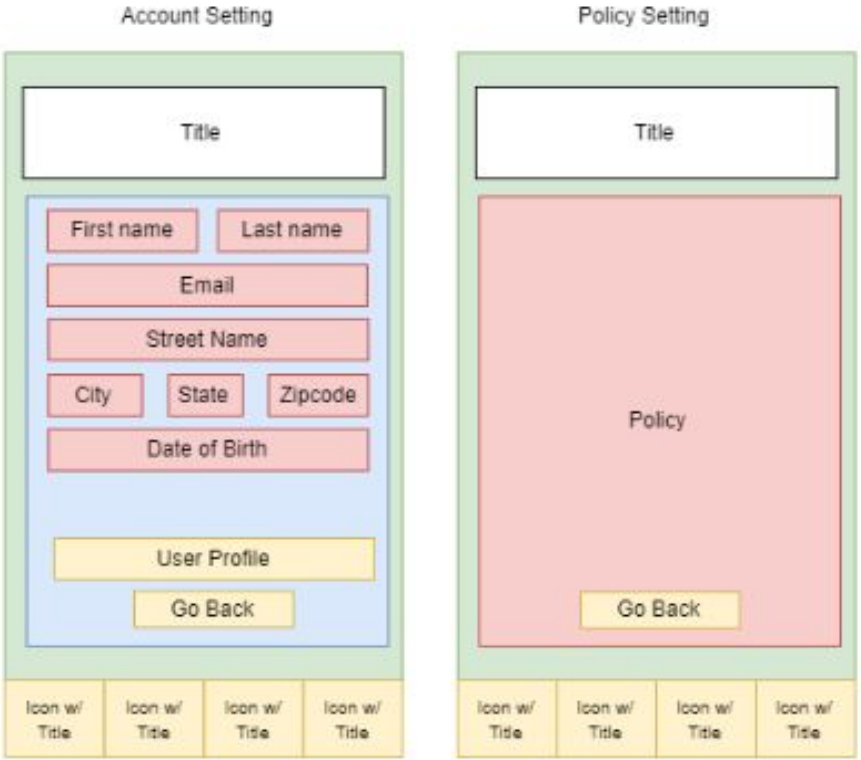
Wireframe


Log In
Title
Username
Password
Sign up
Sign In w/ Google
Sign in
Image
Sign Up
Title
First name
Last name
Email
Street Name
City
State
Zipcode
Date of Birth
Username
Password
Re-Type Password
Sign up
Go Back
Question/Personality
Question
Submit
Go Back


Google Sign In
Title
Email Detected
Email Detected
Sign In
Go Back
Login Screen
Homepage
Title
Event #1
Event #2
Event #3
Event #4
Event #5
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title
Event
Title with Image/Banner
Description
Date / Time
Address
Direction/Map
Join
Go Back
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title


Search Event
Title
Search Bar
Go
Filter
Result
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title
User Profile
Title
User Info
Description
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title
Map
Map
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title


Personal Schedule
Title
Filter
Add
List
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title
Host Event
Title
Event Title
Map
Time
Description
Add
Go Back
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title
System Setting
Title
Settings
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title


Account Setting
Title
First name
Last name
Email
Street Name
City
State
Zipcode
Date of Birth
User Profile
Go Back
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title
Policy Setting
Title
Policy
Go Back
Icon w/ Title
Icon w/ Title
Icon w/ Title
Icon w/ Title

Interfaces

- Users
  - Input Controls: buttons, text fields, toggles, date field, etc.
  - Navigational Components: Sliders, search field, tags, icon, etc.
  - Information Components: notifications, message boxes, etc.
- Admins
  - Similar to User Interfaces with extra interfaces such as buttons to disable some functions, monitor activity, etc.
- Channel Partners
  - Host an event with interfaces that are similar to User Interfaces
- External APIs
  - Interaction with the maps for parks, droppable pins to parks, etc.

Non-Functional Requirements

- Reliability - Information and data are reliable.
- Maintainability - Profile, schedule, and data are easily maintained.
- Accessibility - Application can be accessed via phones, tablets, and computers, each with different operating systems.
- Credibility - No scams or harm in info/data.
- Amiability - Focus on networking and making friends.
- Satisfiability - Easier on the eyes and stress free UI.
- Detectability - Detects bots and scams. In addition, smoother detection of events.
- Performability - Smoother and faster application performance

Performance Requirements

Web Based Requirements:

- Smooth flow of website depends heavily on the internet connection.
- Regulated online traffic for better performance.
- A strong security to prevent scammers and bots.
- A dedicated browser for better performance.
- A recommended operating system.
- Some plugins that can help better the performance.
- Dedicated web server to handle user, data traffic, and security.

Performance Requirements Cont.

Phone/Tablet Requirement:

- Specific operating system for better performance on data handling and UI.
- Minimum requirement hardware for better performance on data handling and UI.
- Dedicated mobile server, if it is on an app.
- Latest upgrade on tablets and phones for up-to-date versions
- Dedicated web server, if accessed on the web.
- Strong internet connection to access maps and other events.
- Security to prevent scammers, data leaks, and bots.

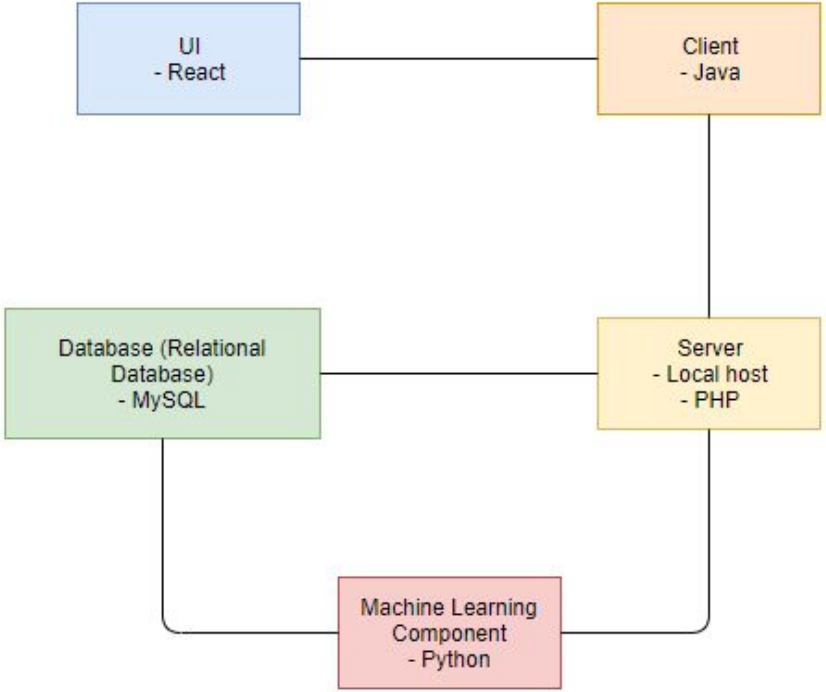
Future Iterations

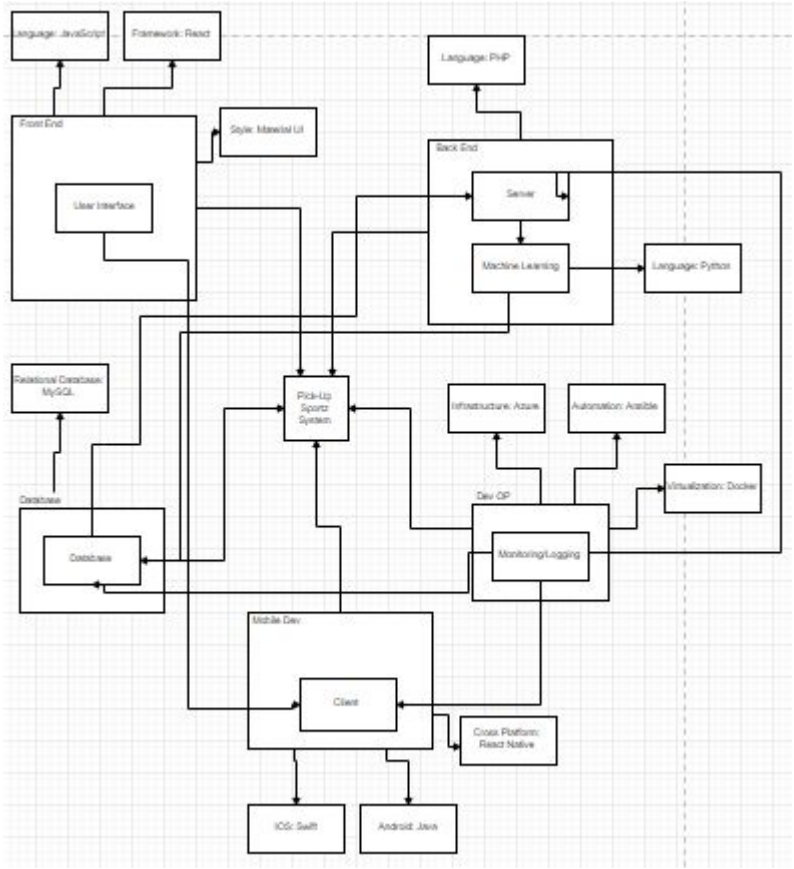
Future Iterations:

- Security of Database and other confidential data
- Logs/Monitoring choices
- CI/CD choices
- Process decisions

Architecture and Design Document

System Component Diagram





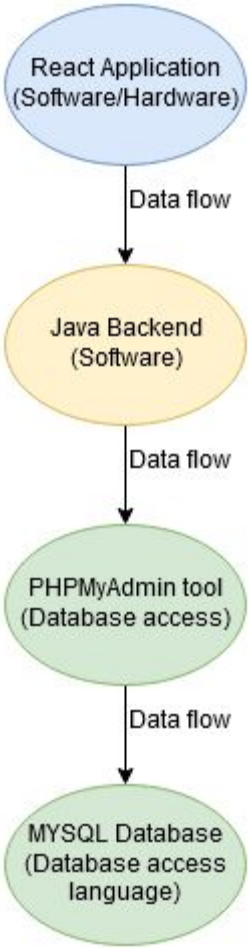
Number of Team Created Servers and APIs in the Architecture?
- 1 server and 1 unfinished API for right now.

User platform
- Mobile and Web application, because of Fullstack

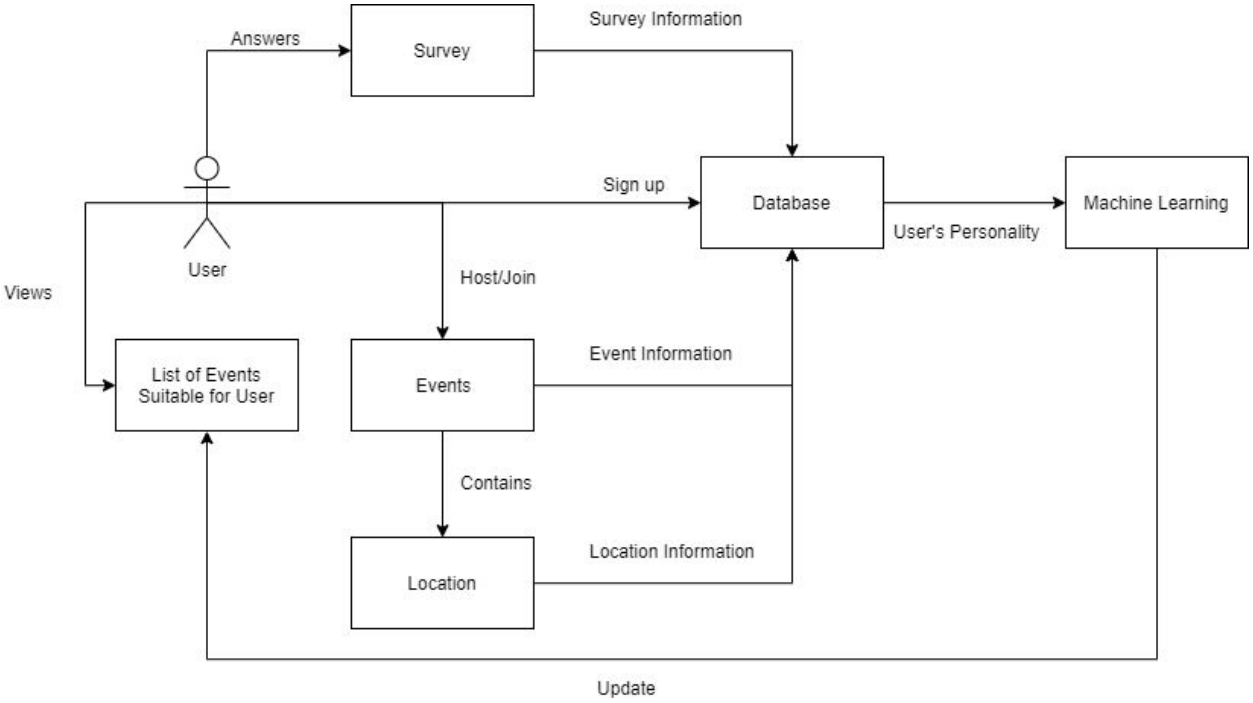
Language for main Server?
- For right now Java as a placeholder, using MySQL. Then PHP after we got some functionalities working.

Types of Databases planned?
- one, MySQL

Components Involved in Database Connectivity



Information Flow Diagram



Quality and Quantity Standard

To set up the front end of our application, we will be using JavaScript through the React Native framework. It will be set up with the Material UI style. By using React Native, there is less burden on us to handle the UI interactions and makes threading a lot easier.

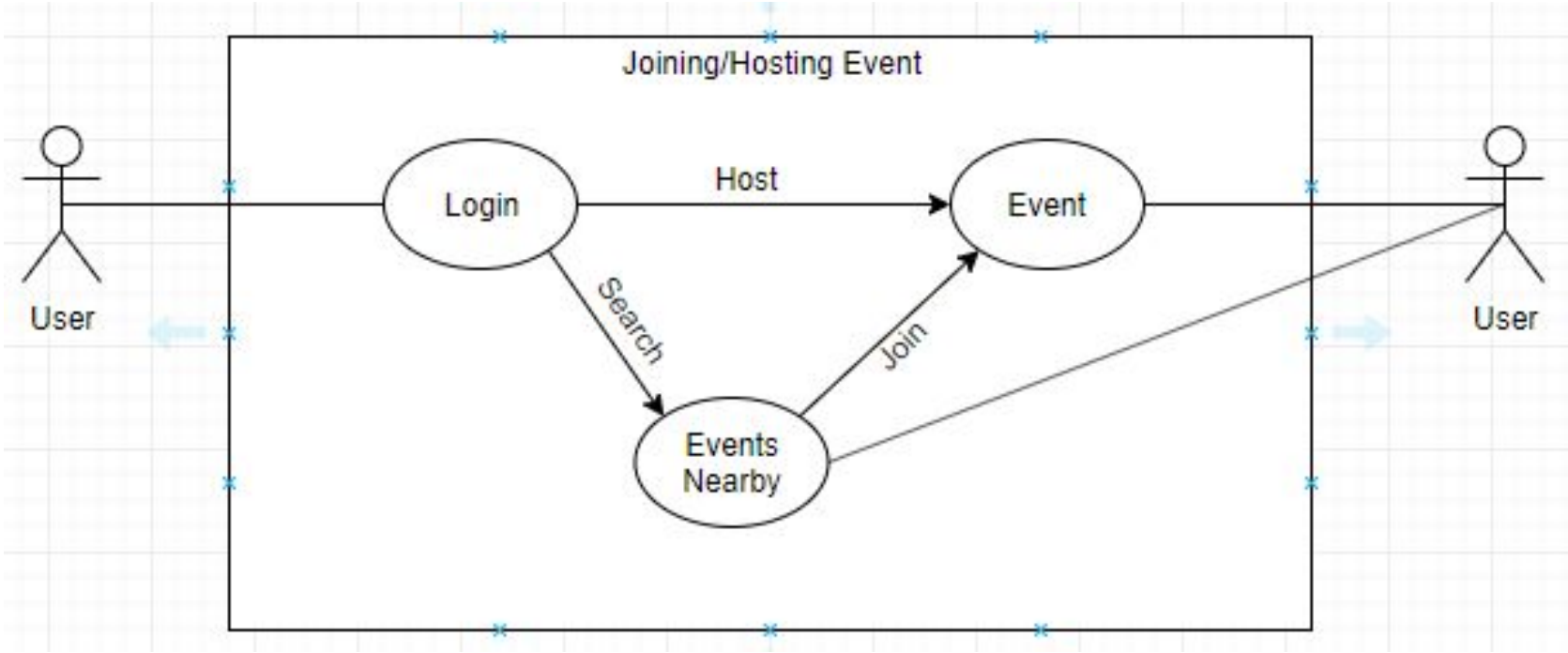
For mobile development, using React Native will allow us to develop the application for either the Android, iOS platform, or even both to allow the app to function across both platforms. If we choose to build the app solely on iOS then we will use Swift, and Java if we choose to make the app exclusively for Android.

To set up the back end of our application, the language we are planning to use is PHP. This will consist of a server component and our machine learning component written in Python that will communicate with our client, database, and our DevOps.

To set up the database component, we will use MySQL as a primary foundation for our relational database. The database will obtain data from numerous components such as, client, server, monitoring/logging, and machine learning components. From this, it will work hand-in-hand with our back end.

To set up the DevOp component, it will consist of an infrastructure using Azure with an Automation, Ansible, and a virtualization, Docker. This portion will consist of our monitoring/logging components that will communicate with our client, server, and database.

User Join/Host Use Case

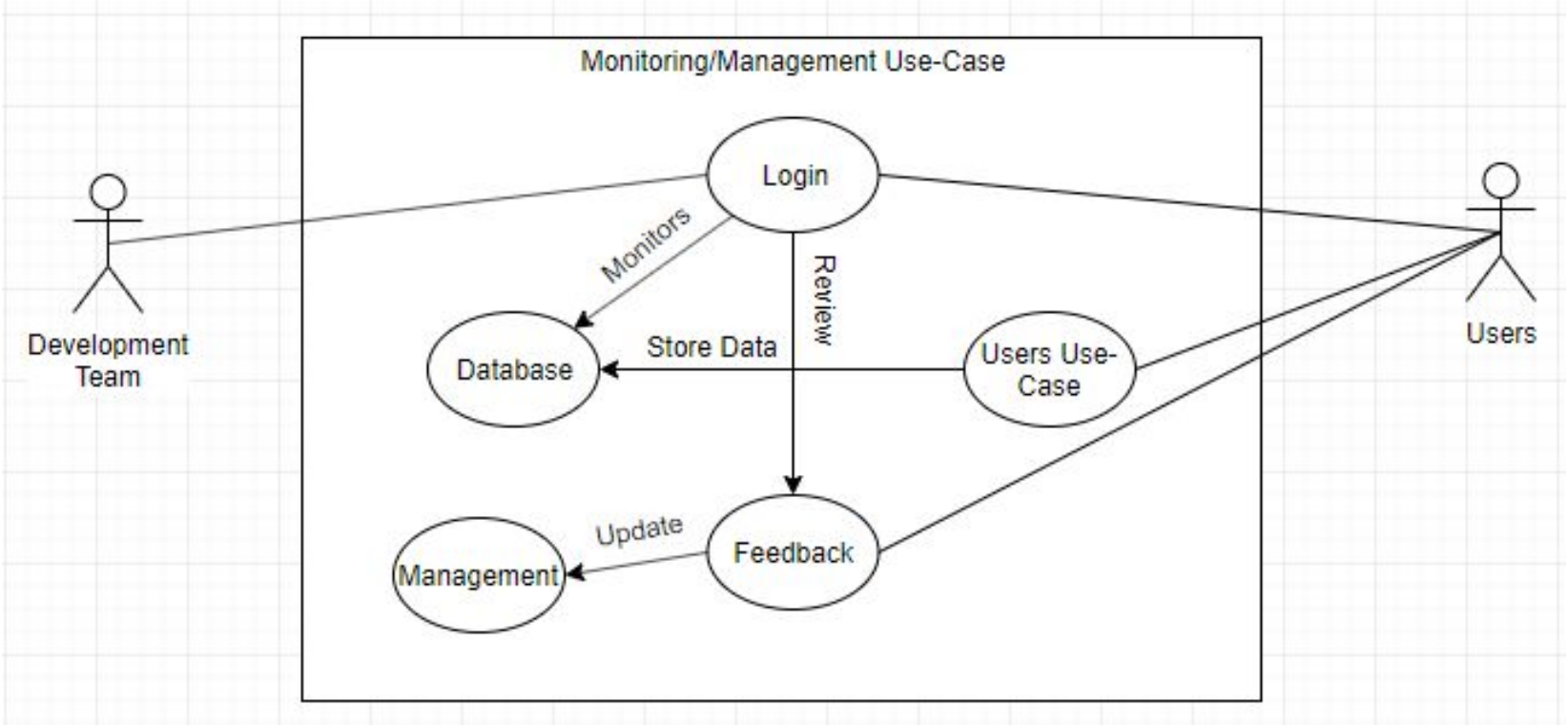


- Use-case field
  - This use case shows the action of the user when joining/hosting an event.
- Use-case name
  - Host/Join an event
- Subject area
  - Users Use Case
- Actors
  - User (left) is the actor that is hosting/joining a game
  - User (right) is the actor that is other user(s) that hosted or joined a game
- Use-case overview
  - A user (left) first logins then either search for an event nearby to join or host their own event.
  - A user (right) has either joined the event that the user (left) has hosted or joined the same event the user (left) has joined.
- Preconditions
  - User (left) must have signed up and answered the survey.
- Termination outcome
  - The use case might end when the event the user (left) joined is submitted to the system, i.e. the game has started.
  - The use case might end when the event the user (left) joined is cancelled.
- Condition affecting termination outcome
  - Cancelled event.
  - Event has passed the deadline.
  - Event has started

- Use-case description
  - Cancelled event.
    - The system will notify the users that the event has been cancelled and will close down the event.
  - Event has passed the deadline.
    - The system will notify the users that the event has passed the deadline, erasing the event from the database.
  - Event has started
    - The system will notify the users that the event has started and will save the event into the users schedule/database, while also erasing the event from the event nearby list.
- Use-case associations
  - Sign up Use Case
  - Check Schedule Use Case
  - Filter/Map Use Case
  - Look Up Other Users Profile Use Case
- Traceability to a list of other related documents, models, and products that are associated with this use case
  - Map API
  - Users Profile/History
  - Nearby Events
- Input summary
  - Event search bar/filter
  - Event descriptions/information
  - Login information

- Output summary
  - List of events nearby
  - Schedule
- Usability index (out of 10)
  - Satisfaction (7)
  - Importance (10)
  - Frequency (10)
- Use case notes
  - System must keep track of information of events, such as type of event, location, either hosted or joined, etc. so that the machine learning component can learn through the user’s personality which can result in better events suited for them.

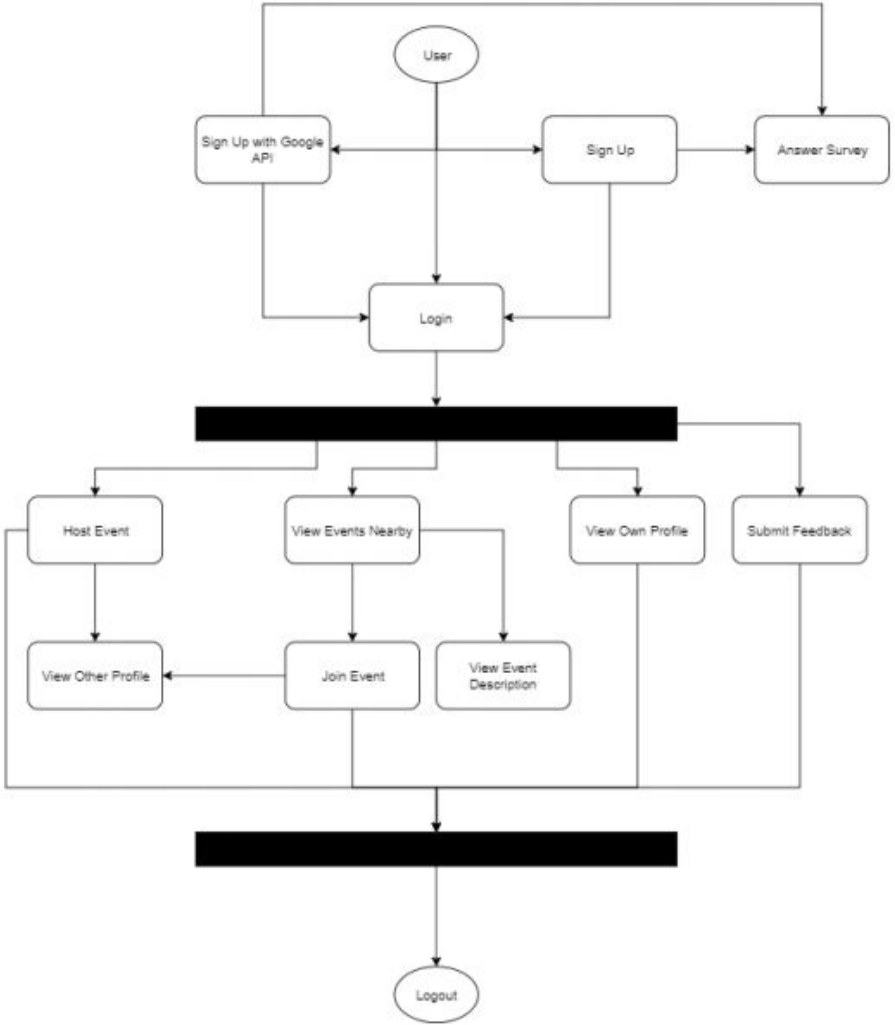
Monitoring/Management Use Case Diagram



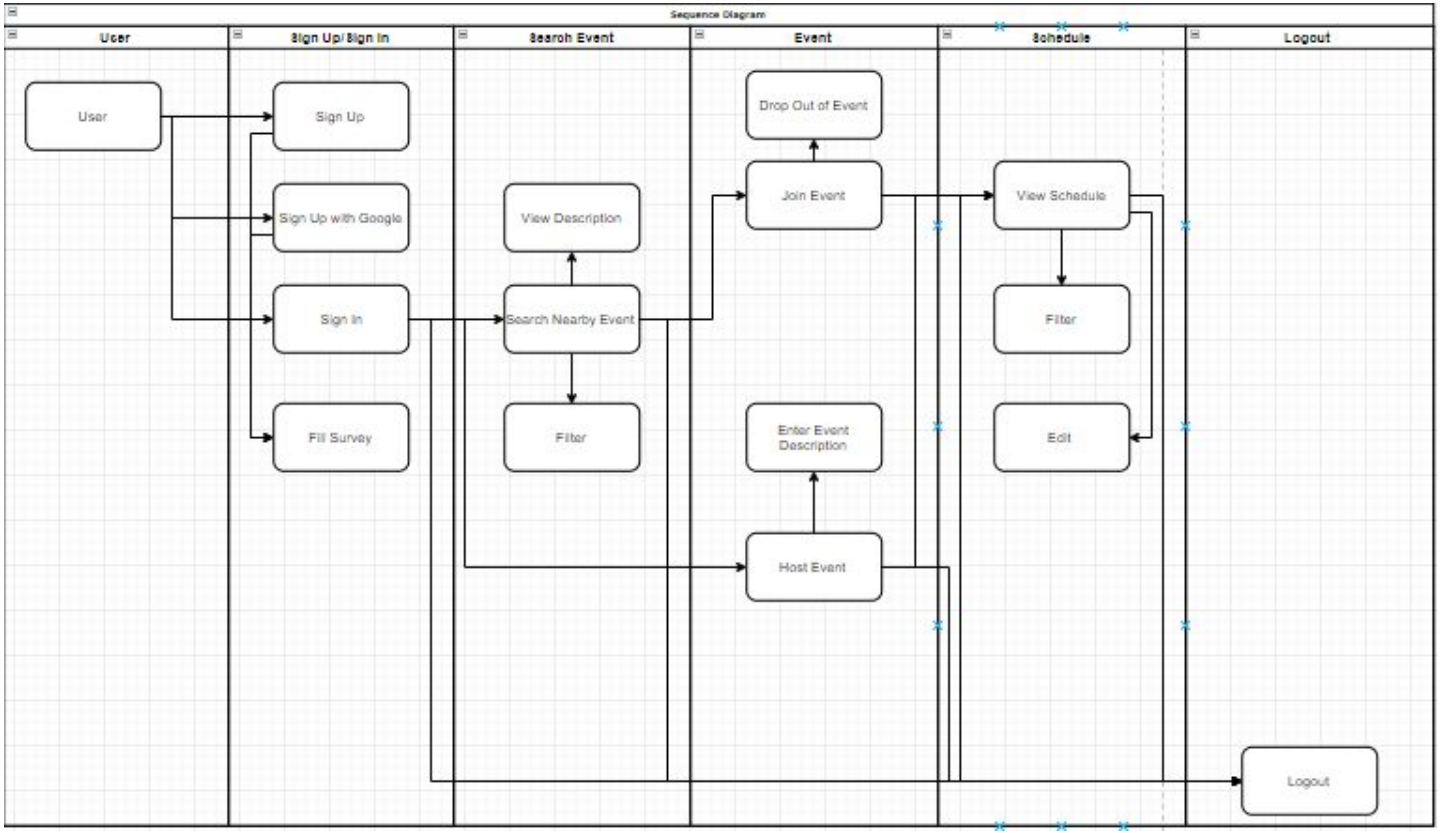
- Use Case Field
  - This use case shows what and how the development team monitors/manage in the system
- Use Case Name
  - Manage/Monitor the System
- Subject Area
  - Developers Use-Case
- Business Event
  - Login
- Actors
  - Development Team
  - Users
- Use-Case overview
  - The development team monitors the users activity; such as events joined/hosted, change in personal information, etc. while also managing feedback from users when they use the product.
- Preconditions
  - When users change their information or someway alter the database through using the application. In addition, when users submit a feedback/comment that has to do with some functionalities of the product.
- Termination Outcome
  - There is no termination outcome for monitoring and managing the application. There needs to be constant managing and monitoring.
- Condition Affecting Termination Outcome
  - No conditions

- Use Case Associations
  - User Use Case
- Input summary
  - Login information
  - Notification for change in system
  - Update Notification
  - Feedback reply
- Output summary
  - System Notification
  - Feedback reply
- Usability Index (out of 10)
  - Satisfaction (6)
  - Importance (10)
  - Frequency (10)
- Use case notes
  - This use case will allow developers to implement new functionalities according to user feedback. In addition, it will allow developers to secure the confidential information/data leaks.

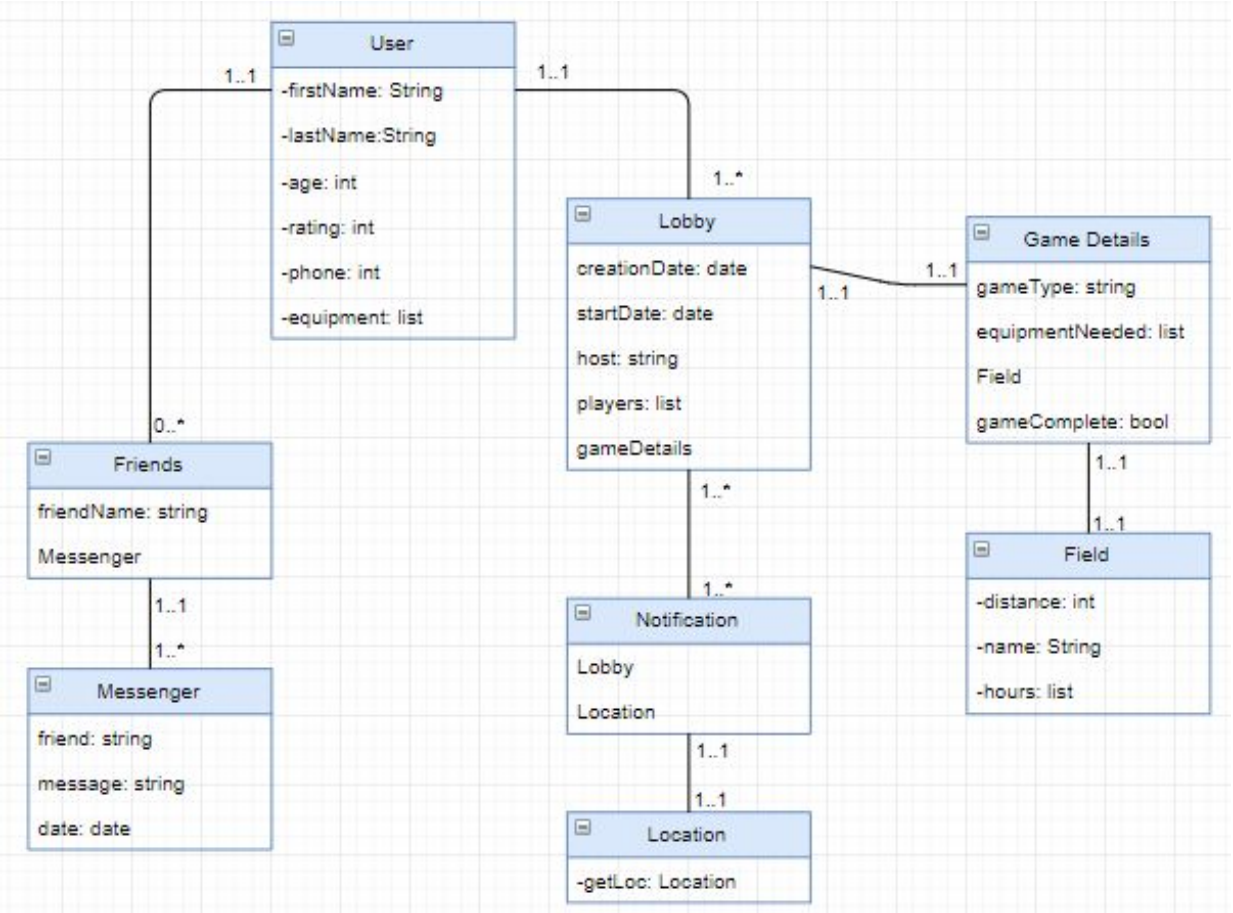
Activity Diagram



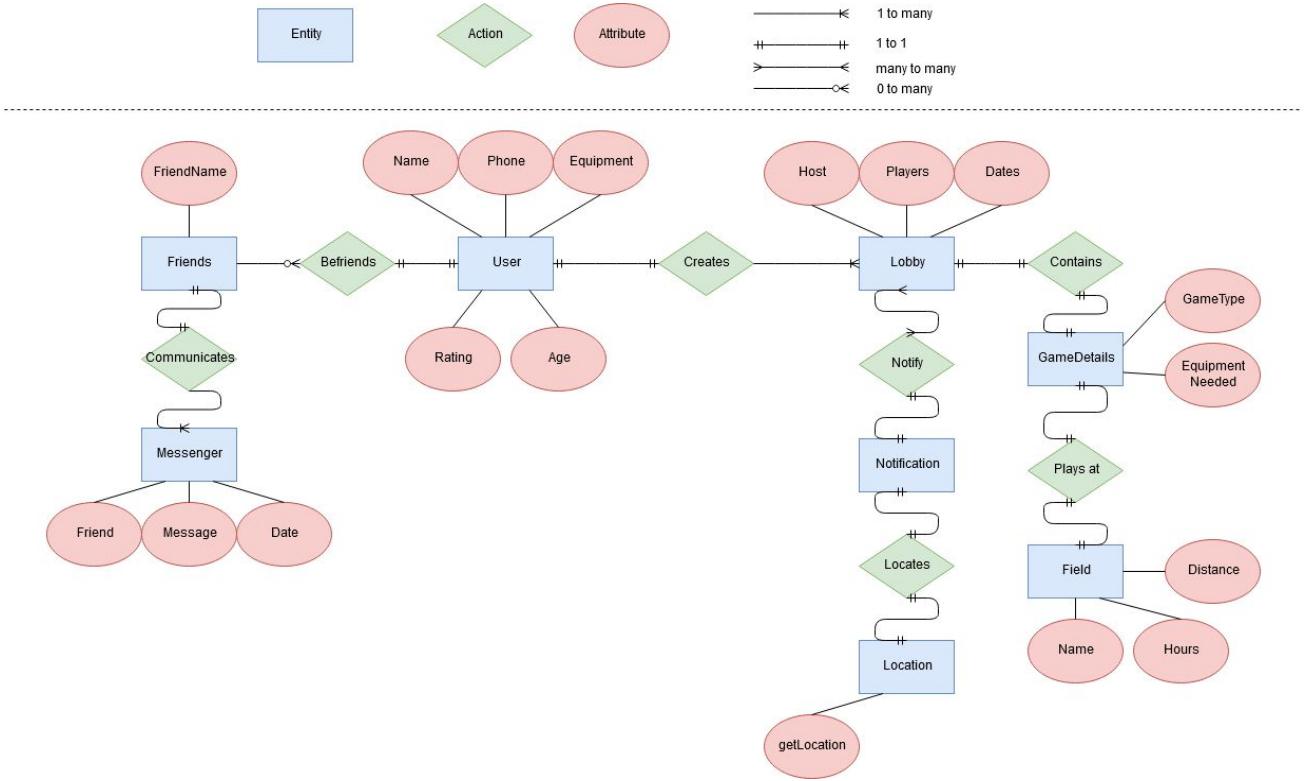
Sequence Diagram



UML Diagram


User
-firstName: String
-lastName:String
-age: int
-rating: int
-phone: int
-equipment: list
Lobby
creationDate: date
startDate: date
host: string
players: list
gameDetails
Game Details
gameType: string
equipmentNeeded: list
Field
gameComplete: bool
Friends
friendName: string
Messenger
Field
-distance: int
-name: String
-hours: list
Messenger
friend: string
message: string
date: date
Notification
Lobby
Location
Location
-getLoc: Location
1..1
1..*
0..*

Relational Diagram



Trade-Off Analyses

Architectural Choice:
Layered

Alternatives:
Interpreter (SQL) and Client-Server (message, email, notification)

Componentes:
UI, Client, Server, Database, Machine Learning.

Reason for Server Choice:
Something that we knew instead of something we didn't.

Criteria	Metrics	Weight	Max Value
Major Architecure Pattern(s)	Interpreter Pattern (SQL)	10%	25
	Client-Server Pattern (message, email)		25
	Layered Pattern (General Desktop Apps)		50
Total			100
Component Choices	UI	20%	10
	Client		15
	Server		15
	Database		20
	Machine Learning		40
Total			100
Language Choices	JavaScript	15%	10
	Java		20
	Python		50
	PHP		20
Total			100
Framework Choices	Data-Driven Framework	15%	25
	Hybrid Driven Testing Framework		50
	Behavior Driven Testing Framework		25
Total			100
Database Choices	MySQL	5%	100
Total			100
Server or Serverless Choices	MySQL Server	5%	100
Total			100
Front End Framework Choices	React	5%	100
Total			100
API Choices	Google Sign In	25%	50
	Google Maps		50
Total			100
Final Total		100%	

Machine Learning Write Up

Application Use Case
Whenever someone wants to join or host a pick up game ahead of time without the hassle of going to the nearest park where there are uncertain amount of people there willing to play a game with them.

Stakeholder's Benefits
Stakeholders will not waste time nor will they need to go far to find a pick up game. In addition, with the use of machine learning, users will find games that match their likings. This will result in better experience and faster networking.

ML Model vs. Our Objectives
Our machine learning model that we plan to use is supervised learning. It consists of a target/outcome variable which is to be predicted from a given set of predictors. Using these set of variables, we generate a function that maps inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Since our objective for the application of machine learning in our product is to find pickup games that are best suited for our users, this model works hand-to-hand with our objectives

Architectural Choices

- Web Server
    - We chose to use an interpreter pattern and client-server pattern architecture since it works well with our database, which is based on SQL, and messaging system, such as user-to-user communication and system messages. We also chose to use a layered pattern architecture since it works well with mobile applications.

Plan to Deploy

- Web API

Alternative Deployment

- None, cause our planned architecture seemed to cover all basis of our product.

Sprint Goals

- Create a Product Requirements Document
- Update Business Requirements Document
- Update Management Plan
- Start Architecture and Design Documents

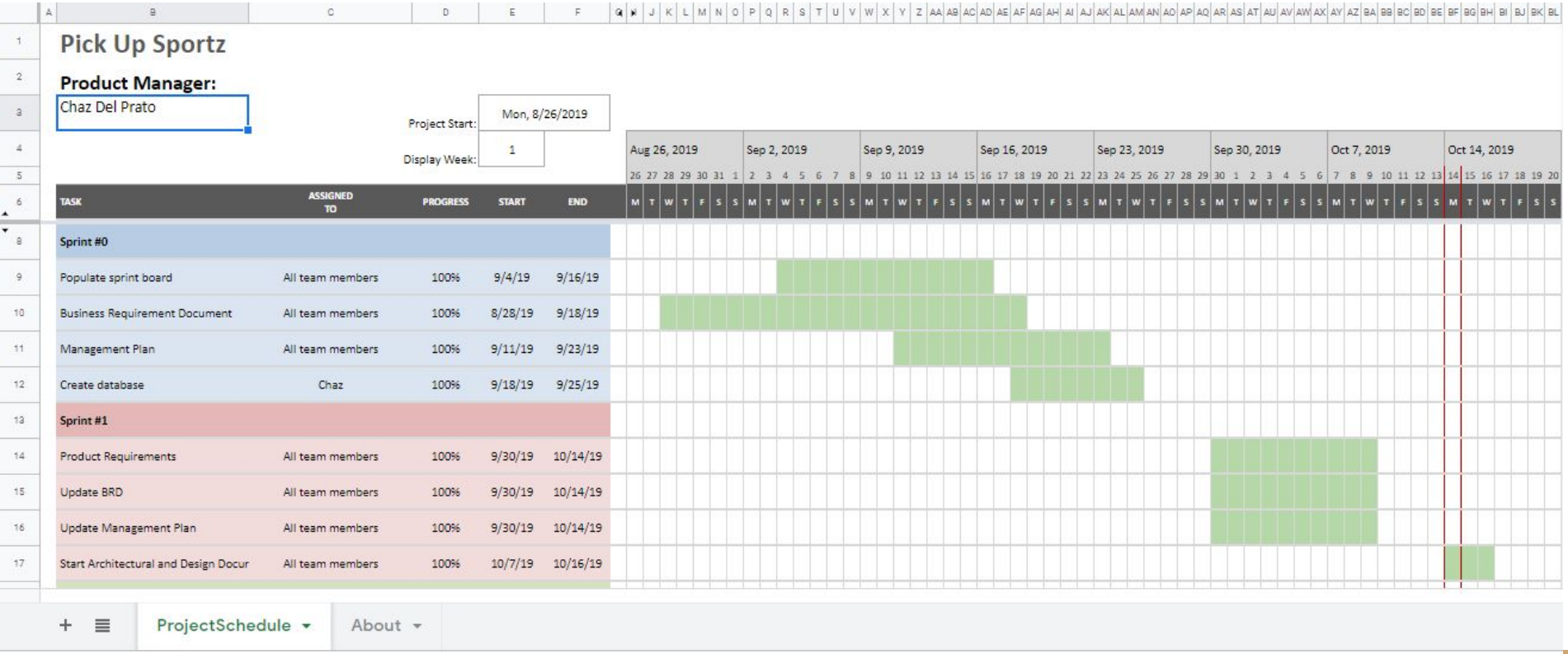
Major Points Updated on BRD

- Personas
- User Stories
- Market Segment

Major Points Updated on Management Plan

- Gantt Chart
- Project Tracking Matrix
- Sprint Board
- Burndown Chart
- Sprint Retrospective
- Lessons Learned from Agile Development

Updated Gantt Chart



Pick Up Sportz

Product Manager:

Chaz Del Prato

Project Start: Mon, 8/26/2019

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Sprint #0				
Populate sprint board	All team members	100%	9/4/19	9/16/19
Business Requirement Document	All team members	100%	8/28/19	9/18/19
Management Plan	All team members	100%	9/11/19	9/23/19
Create database	Chaz	100%	9/18/19	9/25/19
Sprint #1				
Product Requirements	All team members	100%	9/30/19	10/14/19
Update BRD	All team members	100%	9/30/19	10/14/19
Update Management Plan	All team members	100%	9/30/19	10/14/19
Start Architectural and Design Docur	All team members	100%	10/7/19	10/16/19

Aug 26, 2019 | Sep 2, 2019 | Sep 9, 2019 | Sep 16, 2019 | Sep 23, 2019 | Sep 30, 2019 | Oct 7, 2019 | Oct 14, 2019

ProjectSchedule | About

Updated Project Tracking Matrix

Project Template

Project Pick-Up Sportz

Project Manager Chaz Del Prato

Project Start Date Start of Fall Semester 2019

Project End Date End of Spring Semester 2020

OrangeScrum

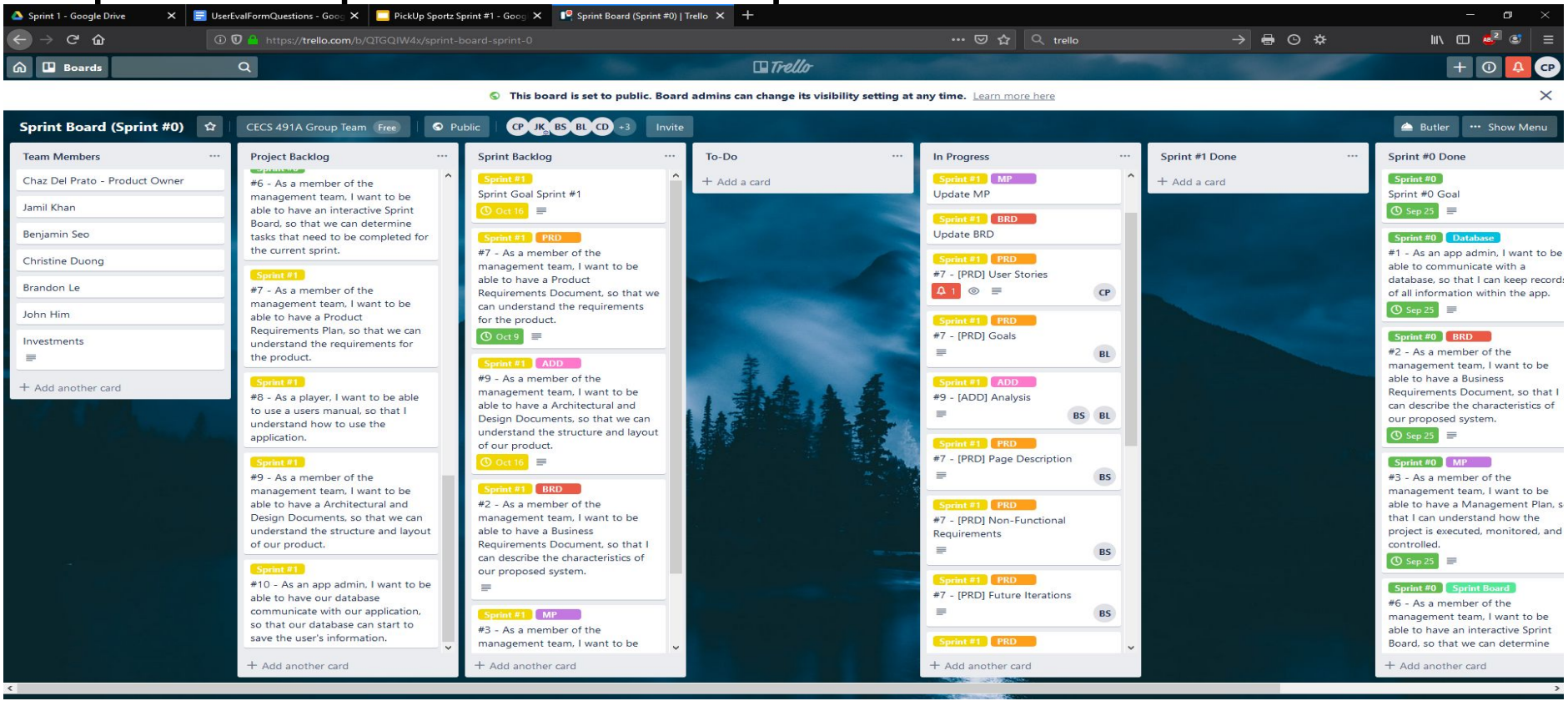
Task	Task Type	Task Status	Est SLOC	Actual SLOC	Priority	Assigned To	Assigned Date	Deadline	Estimated Hrs	Start Date	End Date	% Done	Actual Hr.
Populate Sprint Board	Sprint #0	Completed			High	Chaz Del Prato, Benjamin Seo, Brandon Le, John Him, Jamil Khan, Christine Duong	28-Aug-19	18-Sep-19	2	26-Aug-19	18-Sep-19	100%	3
Product Details	BRD	Completed			Medum	Chaz Del Prato	28-Aug-19	18-Sep-19	1	26-Aug-19	18-Sep-19	100%	1
Preface	BRD	Completed			Low	Benjamin Seo	28-Aug-19	18-Sep-19	1	16-Sep-19	18-Sep-19	100%	1
Appendix	BRD	Completed			Low	Benjamin Seo	28-Aug-19	18-Sep-19	1	16-Sep-19	18-Sep-19	100%	1
Mini Business Plan - Strategy	BRD	Completed			Medum	Brandon Le	28-Aug-19	18-Sep-19	2	26-Aug-19	18-Sep-19	100%	1
Mini Business Plan - SWOT Analysis	BRD	Completed			Medum	John Him	28-Aug-19	18-Sep-19	1	26-Aug-19	18-Sep-19	100%	1
Mini Business Plan - Market Segment	BRD	Completed			Medum	Brandon Le	28-Aug-19	18-Sep-19	2	26-Aug-19	18-Sep-19	100%	2
Mini Business Plan - Primary Market Research	BRD	Completed			High	Benjamin Seo	28-Aug-19	18-Sep-19	5	26-Aug-19	18-Sep-19	100%	4
Mini Business Plan - Secondary Market Research	BRD	Completed			High	Chaz Del Prato, Benjamin Seo, Christine Duong, Jamil Khan	28-Aug-19	18-Sep-19	8	26-Aug-19	18-Sep-19	100%	6
Mini Business Plan - Comercialization Strategy	BRD	Completed			Medum	John Him	28-Aug-19	18-Sep-19	2	26-Aug-19	18-Sep-19	100%	2
Mini Business Plan - Monetization Strategy	BRD	Completed			Medum	John Him	28-Aug-19	18-Sep-19	2	26-Aug-19	18-Sep-19	100%	2
Total Solution	BRD	Completed			Medum	Benjamin Seo	28-Aug-19	18-Sep-19	2	17-Sep-19	18-Sep-19	100%	1
Gantt Chart	MP	Completed			High	Brandon Le, Christine Duong	11-Sep-19	23-Sep-19	4	18-Sep-19	23-Sep-19	100%	1
Project Tracking Matrix	MP	Completed			High	Benjamin Seo	11-Sep-19	23-Sep-19	2	21-Sep-19	23-Sep-19	100%	2
Code Development Velocity Calculation	MP	Completed			High	Chaz Del Prato	11-Sep-19	23-Sep-19	0.5	18-Sep-19	23-Sep-19	100%	0.5
Sprint Board	MP	Completed			High	Jamil Khan	11-Sep-19	23-Sep-19	2	18-Sep-19	23-Sep-19	100%	2
Burndown Chart	MP	Completed			High	Jamil Khan	11-Sep-19	23-Sep-19	3	18-Sep-19	23-Sep-19	100%	2
Sprint Retrospective	MP	Completed			High	Benjamin Seo	11-Sep-19	23-Sep-19	1	21-Sep-19	23-Sep-19	100%	1
Sprint Review Records in an Appendix	MP	Completed			High	John Him	11-Sep-19	23-Sep-19	3	21-Sep-19	23-Sep-19	100%	3
Lessons Learned From Agile Devlopment	MP	Completed			High	John Him	11-Sep-19	23-Sep-19	1	21-Sep-19	23-Sep-19	100%	1
Fix Preface - Technology Innovation or Depth	BRD	Completed			Medum	Benjamin Seo	22-Sep-19	29-Sep-19	0.5	22-Sep-19	29-Sep-19	100%	0.5
Fix Strategy - Unfair Advantage, Customer Segment, Channels	BRD	Completed			Medum	Brandon Le	22-Sep-19	29-Sep-19	1	22-Sep-19	29-Sep-19	100%	1

Updated Project Tracking Matrix

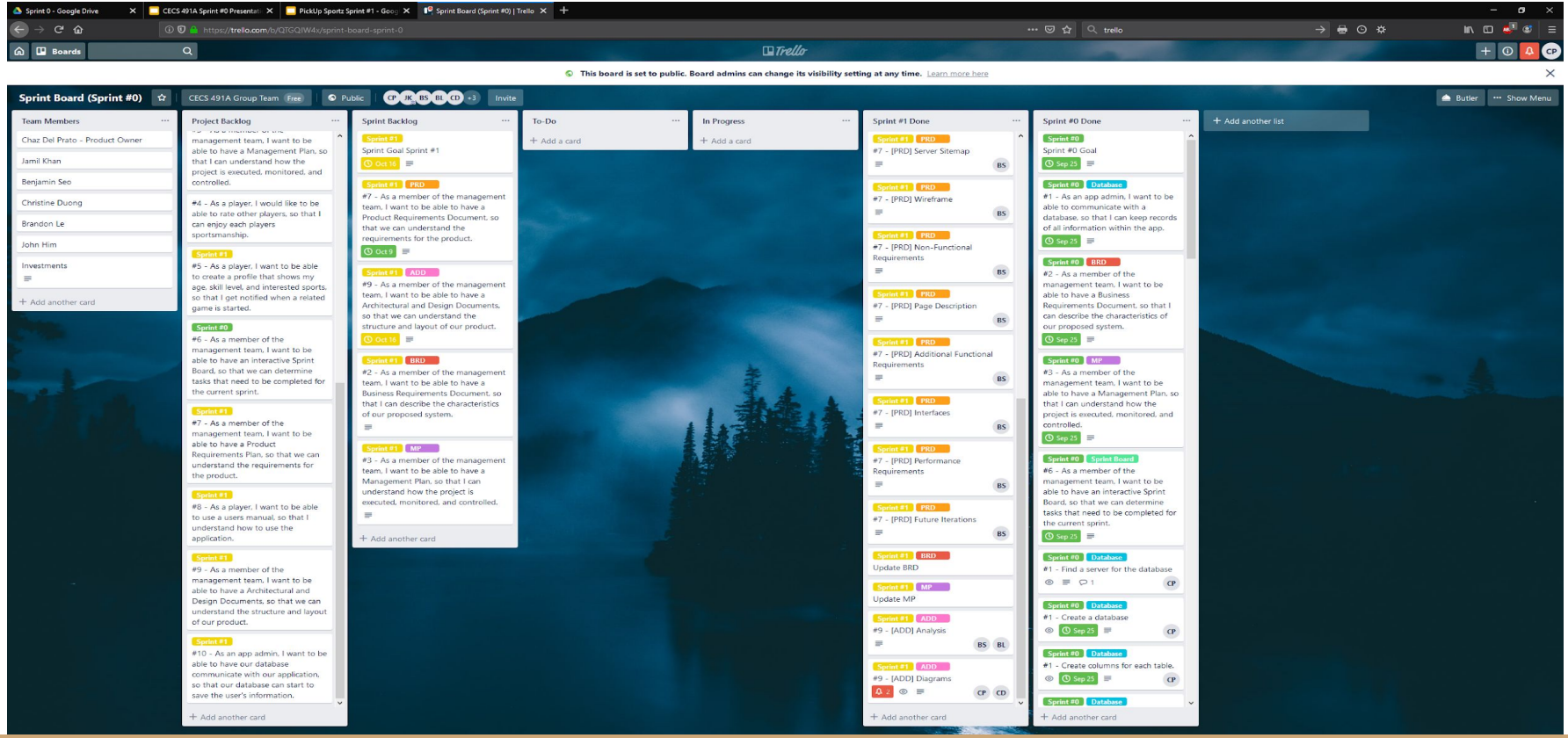
Fix Competitors	BRD	Completed			Medum	Benjamin Seo	22-Sep-19	29-Sep-19	0.5	22-Sep-19	29-Sep-19	100%	0.5
Fix Personas	BRD	Completed			Medum	Brandon Le	22-Sep-19	29-Sep-19	1	22-Sep-19	29-Sep-19	100%	1
Create Database	Sprint #0	Completed			High	Chaz Del Prato	18-Sep-19	29-Sep-19	8	18-Sep-19	29-Sep-19	100%	5
Fix Gant Chart	MP	Completed			Medum	Brendon Le, Christine Duong	22-Sep-19	29-Sep-19	1	22-Sep-19	27-Sep-19	100%	1
Goals	PRD	Completed			High	Brandon Le	30-Sep-19	14-Oct-19	0.5	30-Sep-19	14-Oct-19	100%	0.5
User Personas	PRD	Completed			High	Jamil Khan	30-Sep-19	14-Oct-19	0.5	30-Sep-19	14-Oct-19	100%	0.5
User Stories	PRD	Completed			High	Chaz Del Prato	30-Sep-19	14-Oct-19	0.5	30-Sep-19	14-Oct-19	100%	0.5
Server Sitemap	PRD	Completed			High	Benjamin Seo	30-Sep-19	14-Oct-19	1.5	30-Sep-19	14-Oct-19	100%	1.5
Page Description	PRD	Completed			High	Benjamin Seo	30-Sep-19	14-Oct-19	1	30-Sep-19	14-Oct-19	100%	1
Wireframe	PRD	Completed			High	Benjamin Seo	30-Sep-19	14-Oct-19	2	30-Sep-19	14-Oct-19	100%	3
Interfaces	PRD	Completed			High	Benjamin Seo	30-Sep-19	14-Oct-19	1	30-Sep-19	14-Oct-19	100%	1
Additional Functional Requirements	PRD	Completed			High	Benjamin Seo	30-Sep-19	14-Oct-19	1	30-Sep-19	14-Oct-19	100%	1
Non-Functional Requirements	PRD	Completed			High	Benjamin Seo	30-Sep-19	14-Oct-19	2	30-Sep-19	14-Oct-19	100%	2
Performance Requirements	PRD	Completed			High	Benjamin Seo	30-Sep-19	14-Oct-19	1	30-Sep-19	14-Oct-19	100%	1
Future Iterations	PRD	Completed			Low	Benjamin Seo	30-Sep-19	14-Oct-19	0.5	30-Sep-19	14-Oct-19	100%	0.5
Connect database to application	Database	In-Progress			Low	Chaz Del Prato	30-Sep-19	31-Oct-19	5	30-Sep-19		80%	0
Test data tranfer to and from database	Testing	In-Progress			Low	Chaz Del Prato	30-Sep-19	31-Oct-19	1	30-Sep-19		80%	0
Create user manual	Manual	Created			Low		30-Sep-19	31-Oct-19	3	30-Sep-19		0%	0
UI and Code Username and Password	Code	In-Progress			Low	Jamil Khan, Chaz Del Prato, John Him	30-Sep-19	31-Oct-19	1	30-Sep-19		0%	0
UI and Code User Details	Testing	In-Progress			Low	Jamil Khan, Chaz Del Prato, John Him	30-Sep-19	31-Oct-19	3	30-Sep-19		0%	0
Test Login and User Creation	Testing	Created			Low		30-Sep-19	31-Oct-19	2	30-Sep-19		0%	0
Diagrams Section	ADD	Completed			High	Christine Duong, Chaz Del Prato	7-Oct-19	14-Oct-19	7	7-Oct-19	14-Oct-19	100%	7
Analysis Section	ADD	Completed			High	Benjamin Seo, Brandon Le	7-Oct-19	14-Oct-19	6	7-Oct-19	14-Oct-19	100%	6

Total Hours Spent	72	Project Progress	90.22 %

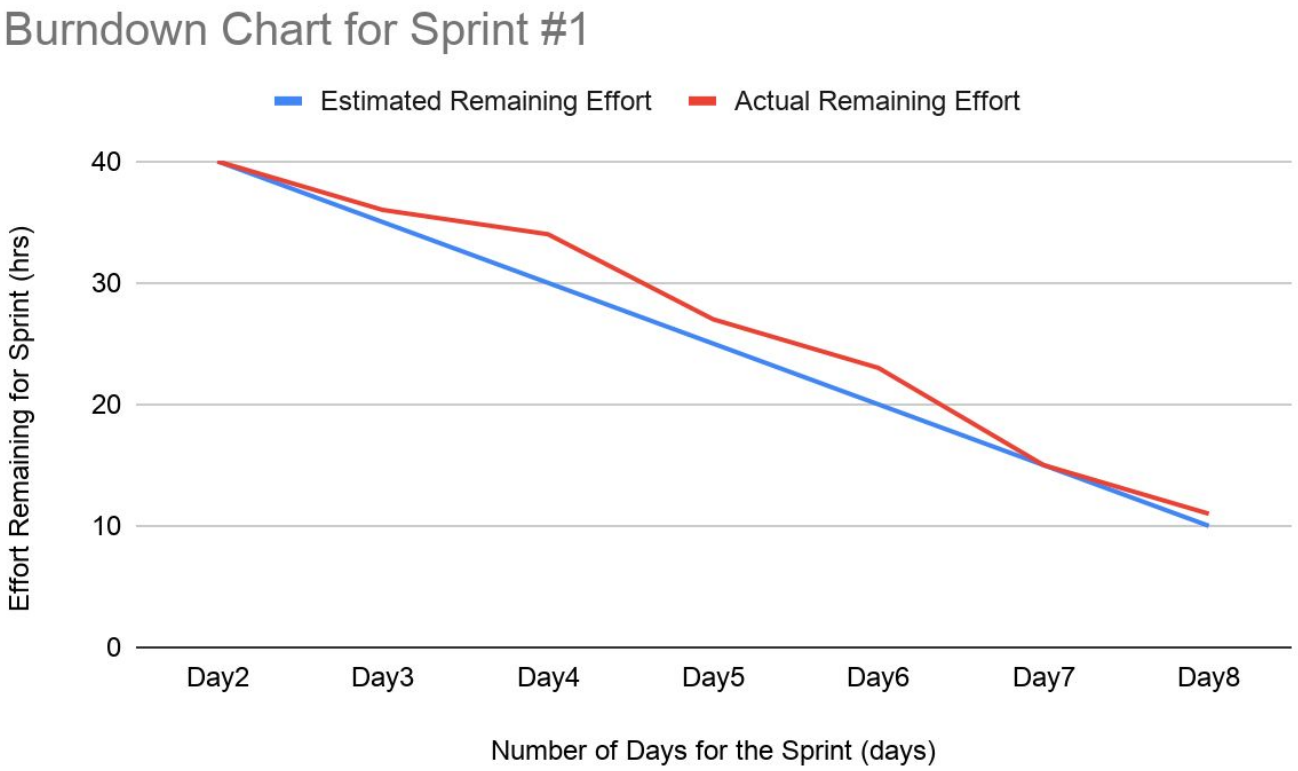
Updated Sprint Board (before)



Updated Sprint Board (after)



Updated Burn Down Chart



Sprint Retrospective

Our sprint goals consist of :

- **Updating our Business Requirement Document**
- **Updating our Management Plan**
- **Creating a Product Requirements Document**
- **Creating a Architecture and Design Document**

Our sprint goals have been met. Even though there were a bit a of hiccups along the way, we have completed the sprint goals on time.

Our team's total velocity is 54 hours since:
3 hours per week * 3 weeks * 6 members

Current hours worked: **~ 40 hours (Story Points)**

Our burndown chart for sprint #1 is looking good. With all the tasks that have been completed, we can clearly see the graph slowly heading down, signifying that we are almost done with a task.

Team velocity for next sprint:
Pay attention to assignment date and due date so burndown chart looks smoother.
Increase the hours per week

Sprint Retrospective Cont.

We decided to prioritize the user stories by their due dates. Since the majority of our user stories were due on the same day, creating the PRD and ADD were a higher priority than updating the previous documents.

Only a few In-Progress user stories related to code.

Tasks per User Story:

- Create PRD - 11 tasks
- Create ADD - 2 tasks
- Update BRD - 1 task
- Update MP - 1 task

What We Learned

- The types of components/architectures needed for a full stack
- The interaction between each major components
- The variety of framework and languages in major components
- Suitable frameworks and languages among each components
- Diagrams

What We Plan For Next Sprint

- Update BRD, PRD, MP, Architecture and Design Document later on for a more concrete layout of our product.
- Start Coding