SW Engineering CSC648-848-05 Spring 2024

Application Title & Name : Teamup

Team : 5

Student Name	SFSU Email	GitHub	Discord	Role
Juan Estrada	jestradazuluaga@sfsu.edu	jjestrada2	juanjosee	Team-lead
Areeb Abbasi	aabbasi@sfsu.edu	areeeeb	_xertz	Backend-lead
Cole Chiodo	cchiodo@sfsu.edu	colechiodo	colechiodo	Docs-editor
Jaycee Lorenzo	jlorenzo3@sfsu.edu	jclorenz0	__jaycee	Frontend-lead
Martin Pham	mpham8@sfsu.edu	mar10fam	marnoki	Github-master
Kotaro Iwanaga	kiwanaga@sfsu.edu	iamkotaaa	kotaro8448	Database-admin

Milestone 3

4/4/2024

Version #	Submission Date
M3V1	04/25/2024
M2V2	04/18/2024
M2V1	04/04/2024
M1V2	03/21/2024
M1V1	03/01/2024

1. Data Definitions

1. User

a. Definition: Represents anyone who interacts with the app, including players and school administrators
b. Usage: Users can browse games, join or create games, and participate in tournaments. School administrators have additional privileges to create and manage tournaments.
c. Attributes: User ID, Name, Email, Password, Profile Picture (JPG, 320x320px), Skill Level, Sports Preferences, Availability.

2. Game

a. Definition: A scheduled sports activity that users can join or create.
b. Usage: Allows users to find, join, or post games matching their interests and schedules.
c. Attributes: Game ID, Sport Type, Location, Date & Time, Required Number of Players, Skill Level Preference, Equipment Details, Organizer (User ID).

3. Team

a. Definition: A group of users who join together to play a sport, either for casual games or within a tournament.
b. Usage: Facilitates team formation for both casual play and competitive tournaments.
c. Attributes: Team ID, Team Name, Members (List of User IDs), Sport Type, Skill Level.

4. Tournament

a. Definition: A competitive event organized by schools or universities, involving multiple games and teams.
b. Usage: Schools can create tournaments, and students can register as solo players, groups, or teams.
c. Attributes: Tournament ID, Name, Sport Type, Date & Time, Location, Registration Details, Tournament Format, Brackets, Standings.

5. Game Location

a. Definition: Physical place where games and tournaments are held.
b. Usage: Users can select locations for their games or view where tournament matches are taking place.
c. Attributes: Location ID, Name, Address, Facility Details, Parking Information.

6. Profile
    a. Definition: Detailed information about a user, including their sports preferences and skill levels.
    b. Usage: Helps in matching users with appropriate games and teams.
    c. Attributes: User ID, Skill Levels (per sport), Sports Preferences, Biography, Contact Information.
7. Admin
    a. Definition: A user who is associated with one school who can create tournaments.
    b. Usage: Admins are approved users with higher privileges. They are able to create tournaments for the school they are associated with.
    c. Attributes: Admin ID, User ID, School ID
8. School
    a. Definition: Location that can host Tournaments.
    b. Usage: Locations that Admins can host Tournaments at.
    c. Attributes: School ID, Name, Location.
9. Sport
    a. Definition: The activity that will be played during a game.
    b. Usage: When creating a game, a sport is chosen that will be played. Users can also define their sport preferences.
    c. Attributes: Sport ID, Name, Description.
10. Review
    a. Definition: A rating given from one user to another.
    b. Usage: Users can rate other users based on their performance or actions after they have played a game together.
    c. Attributes: Review ID, Rating, Description, User ID.
11. Sport Level
    a. Definition: A level of expertise
    b. Usage: Users can filter games based on the level of expertise of a specific game.
    c. Attributes: User ID, Sport ID, Level.

User Types and Privileges

● Player: Regular users who can browse and join games, create games, and register for tournaments. They can also be part of or form teams.
● School Administrator: Users with the authority to create and manage

tournaments on behalf of their institution. They have additional access to tournament management tools.

Registration Info

- Upon account registration, the user must provide the following items of information:
    - Full Name, Username, Email, Password, Date of Birth, Gender, Phone Number.

Usage in Documentation and Development

- These terms and their definitions will be used consistently across all project documentation, user interfaces, software components, and database designs.
- The division of user types into Players and School Administrators informs the app's functionality and access control mechanisms.
- The attributes listed for each entity provide a high-level overview of the data model and serve as a guide for database design and API development.

2. Prioritized Functional Requirements

Priority 1:

User:

1. Users shall be able to search for games based on sport type.
2. Users shall be able to search for games based on location
3. Users shall be able to search for games based on time.
4. Users shall be able to search for games based on player skill level.
5. Users shall be able to join a team in a game
6. Users shall create new games.
7. Users shall specify the sport, location, time, and number of players needed for a game.
8. Users shall be allowed to update their account information.
9. Users shall have the option to manually log out from their accounts.
10. User shall be able to customize their profiles with avatars and bios.
11. Users shall be able to use filters by gender for search games.
12. Users shall be able to rate other players based on skill level.
13. Users shall be able to rate facilities for game locations
14. Users shall earn badges for achievements

Game:

15. Game listing shall provide facility location.
16. Game listing shall provide player profiles.
17. Game listing shall provide facility game rules.

Priority 2 :

User:

18. Users shall receive confirmation emails containing game details.
19. Users shall receive notifications containing game participation instructions.
20. Users shall receive reminders to registered participants before the game start date.
21. Team members shall be able to communicate to coordinate logistical issues related to games.
22. Participants shall receive notifications of their game schedules, including date, time, opponent, and location.
23. The app shall recognize the outstanding performances of top participants.

24. Users shall have the opportunity to provide feedback and evaluations on the game.
25. Users shall have privacy settings to control who can see their profile and game activity.
26. Users shall be able to use optional two-factor authentication (2FA).
27. Users shall see leaderboards for various metrics like most games played,
28.

Priority 3 :

Game:

29. Games shall be able to provide suggestions for indoor locations in case of predicted bad weather.
30. Games shall include options for renting necessary sports equipment.
31. Games shall support calendar integrations for managing team schedules.
32. Games shall include weather forecasts for the scheduled time and location.

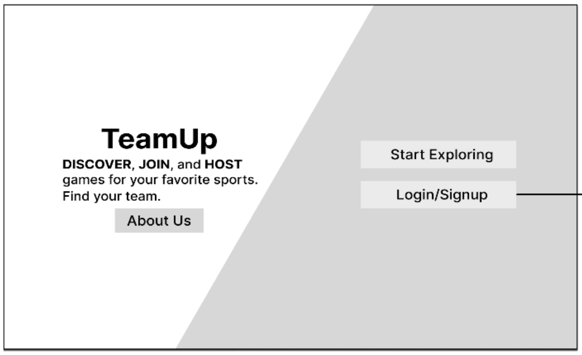
Tournament:

33. Tournaments shall offer options to livestream games.
34. Tournaments shall generate game schedules based on the number of teams, available time slots, and tournament format.
35. Admins shall have the ability to update or change the tournament schedules.
36. The tournament shall enforce a code of conduct policy for game participants.
37. Tournament violations of the code of conduct shall be reported.
38. Admins shall be able to manage team rosters, set up team events, and send out team announcements.

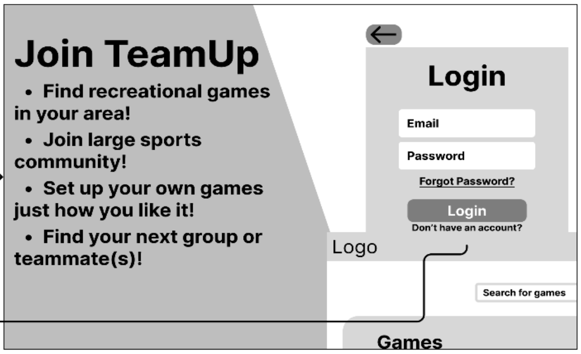
3. Wireframes Based on your Mockups/Storyboards

Use Case 1

- **Darryl wants a way to meet friends at SFSU**
- **He meets people by signing up with TeamUp and joining games**

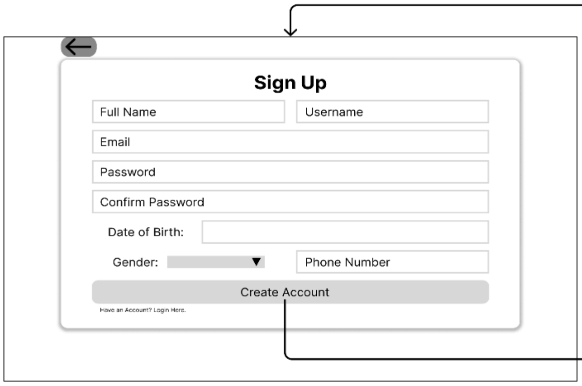

TeamUp
DISCOVER, JOIN, and HOST games for your favorite sports. Find your team.
About Us
Start Exploring
Login/Signup

Darryl clicks login/signup

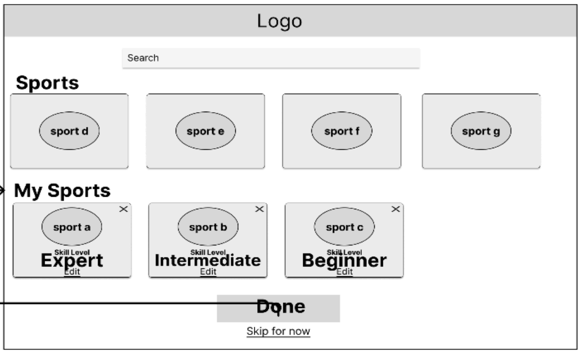

Join TeamUp

- Find recreational games in your area!
- Join large sports community!
- Set up your own games just how you like it!
- Find your next group or teammate(s)!

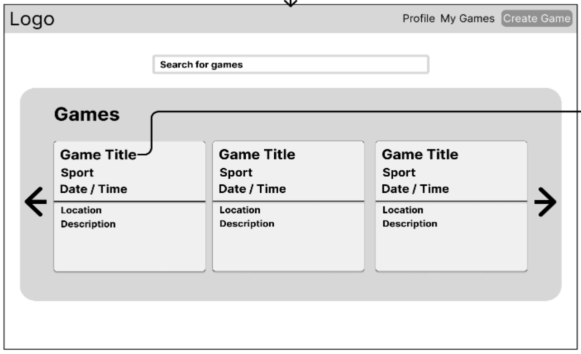
Darryl doesn't have an account so goes to sign up



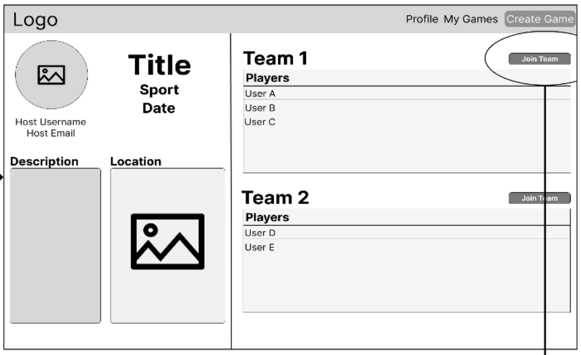
Clicks create account after entering info



Selects his favorite sports and continues to home page



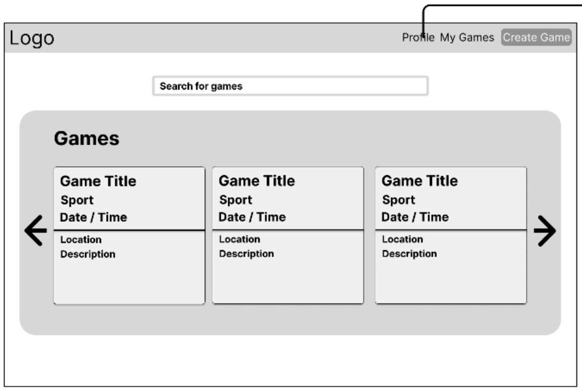
Clicks on a game hosted by other SFSU students



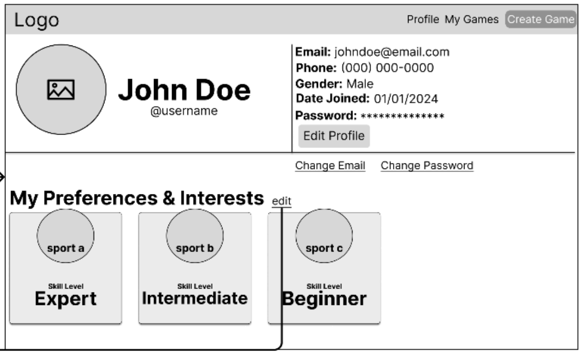
Joins a team

Use Case 2

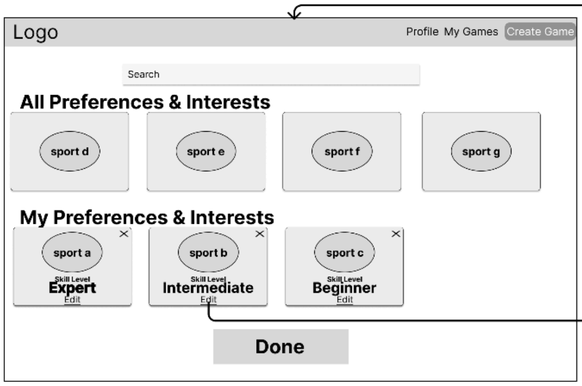
- **Alex is an existing member on TeamUp**
- **He believes it is necessary to adjust his skill level**
- **After adjusting, he has a much more enjoyable experience playing with other players with similar skill level**



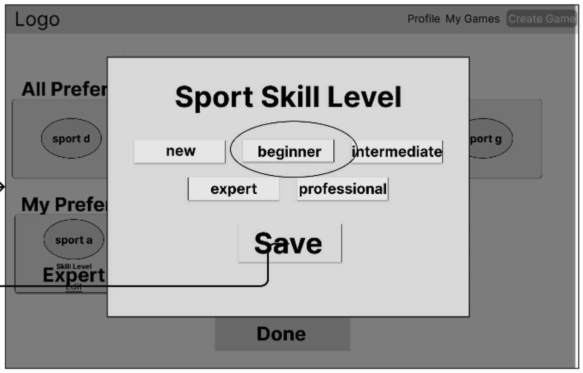
Alex goes to his profile



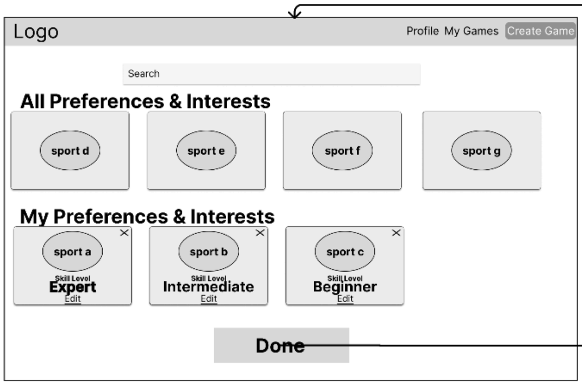
Clicks edit to change his skill level



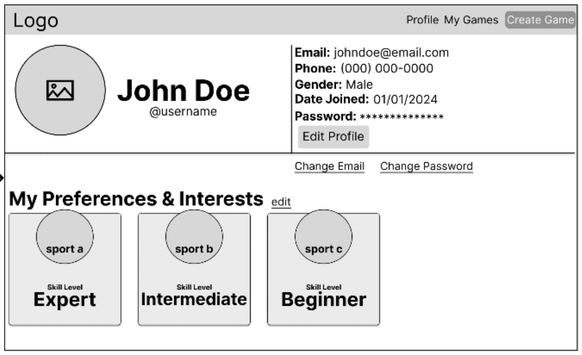
Clicks edit on the sport he wants to change



Selects beginner and saves

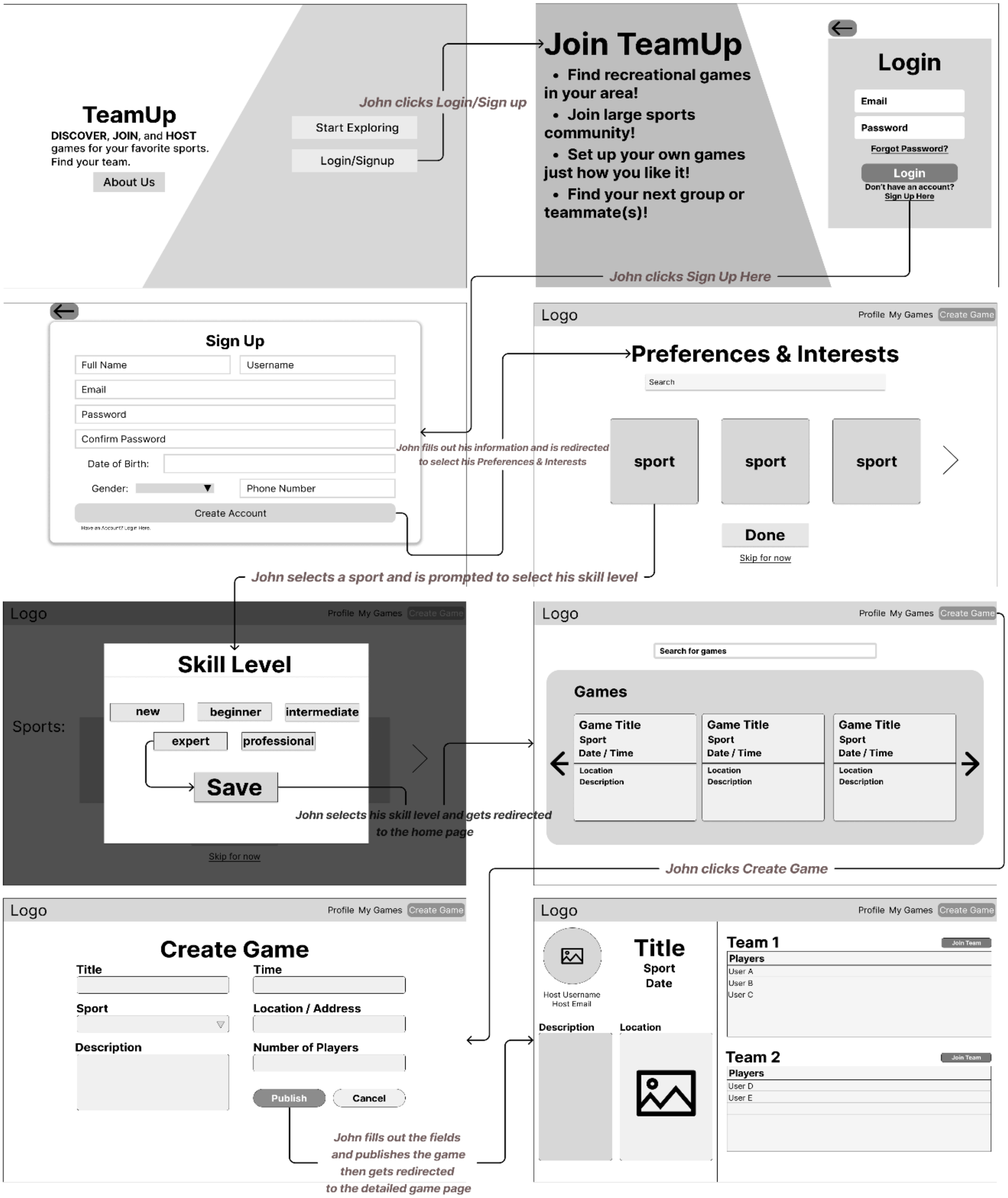


Clicks done and sees changes back in his profile page



Use Case 3

- **John wants to set up basketball games with other high skill level players**
- **John gets introduced to TeamUp**
- **John Registers and creates a game cater to more experienced players**



John clicks Login/Sign up

John clicks Sign Up Here

John fills out his information and is redirected to select his Preferences & Interests

John selects a sport and is prompted to select his skill level

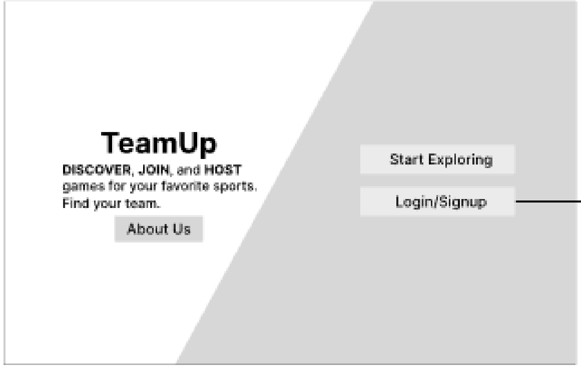
John selects his skill level and gets redirected to the home page

John clicks Create Game

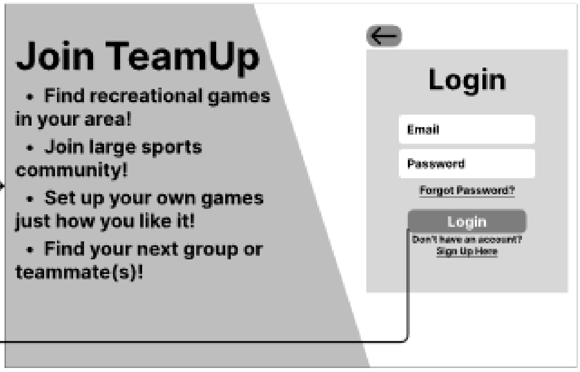
John fills out the fields and publishes the game then gets redirected to the detailed game page

Use Case 4

- **Mary has a Tennis Tournament soon**
- **She wants to get extra practice in using TeamUp**

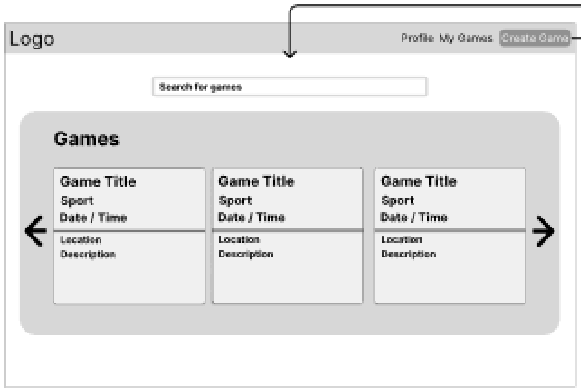

TeamUp
DISCOVER, JOIN, and HOST games for your favorite sports. Find your team.
About Us
Start Exploring
Login/Signup

Mary Clicks Login.

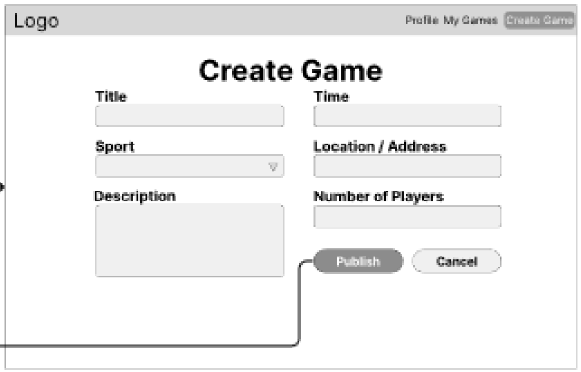

Join TeamUp

- Find recreational games in your area!
- Join large sports community!
- Set up your own games just how you like it!
- Find your next group or teammate(s)!

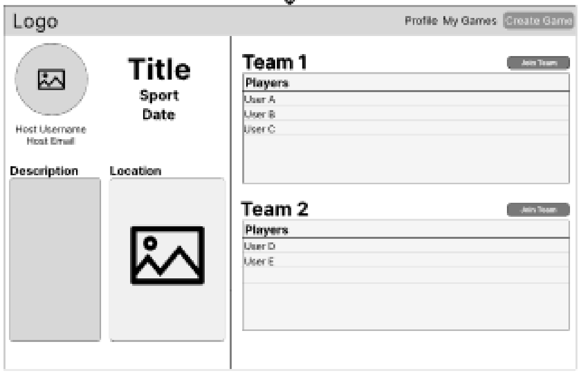
Mary Enters her Information and Logs in.



Mary Clicks Create Game.



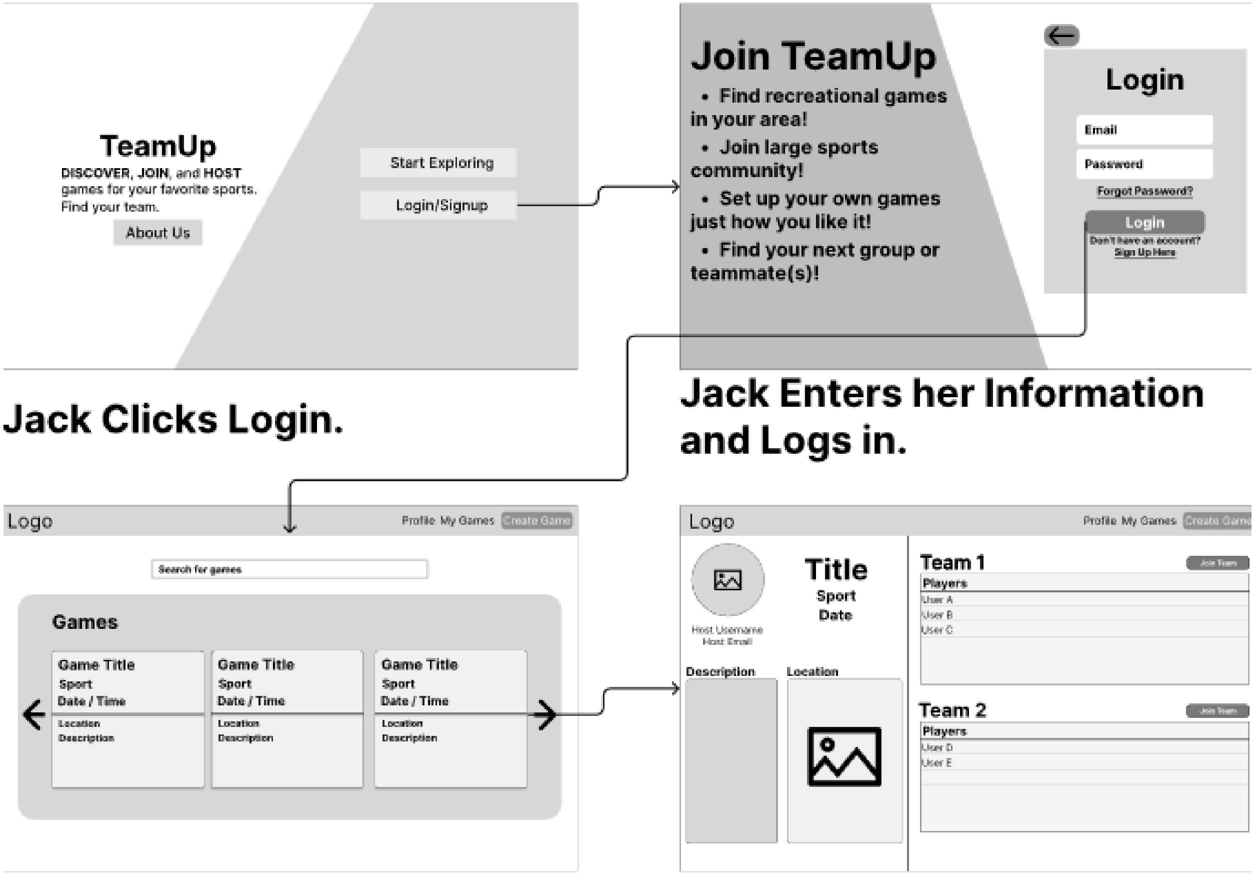
Mary Enters the required game info and clicks Publish.



Mary's game is now created. She waits for an opponent to join.

Use Case 5

- **Jack has moved to a new city**
- **Jack wants to find a casual basketball match to join and play.**



Jack Clicks Login.

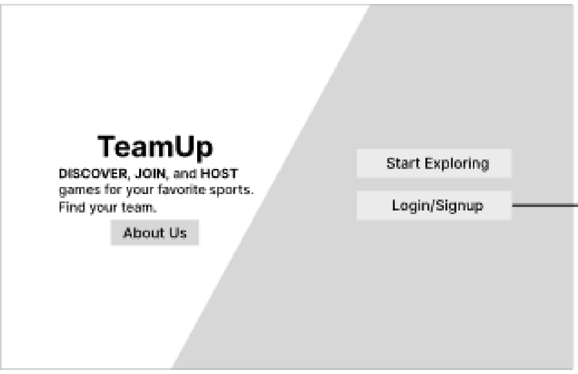
Jack Enters her Information and Logs in.

Jack Clicks A Game that Interests him.

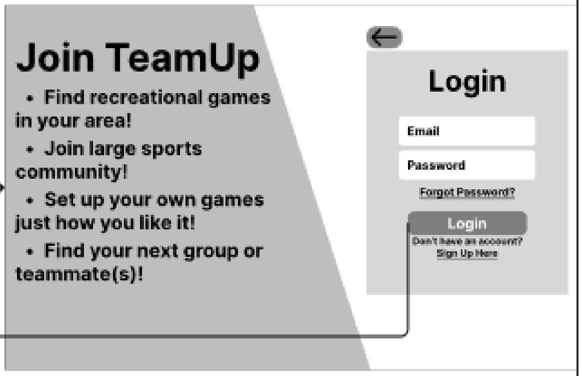
The description of the game details the skill level. So Jack joins the game.

Use Case 6

- **Tyler has joined a game using TeamUp**
- **His roommate, Chris, also wants to play on the same team.**

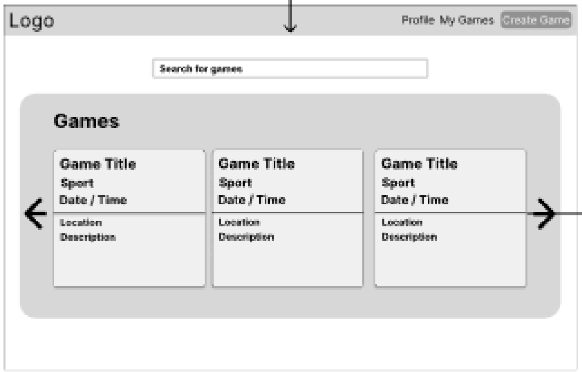

TeamUp
DISCOVER, JOIN, and HOST games for your favorite sports. Find your team.
About Us
Start Exploring
Login/Signup

Chris Clicks Login.

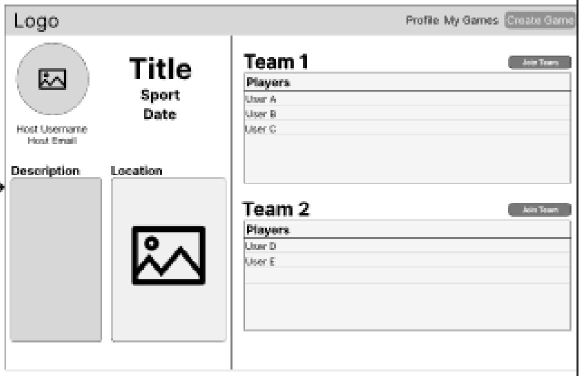


- Find recreational games in your area!
- Join large sports community!
- Set up your own games just how you like it!
- Find your next group or teammate(s)!

Chris Enters her Information and Logs in.



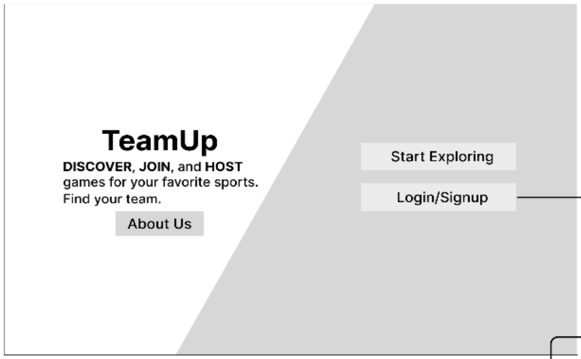
Chris Clicks The Game that Tyler Joined



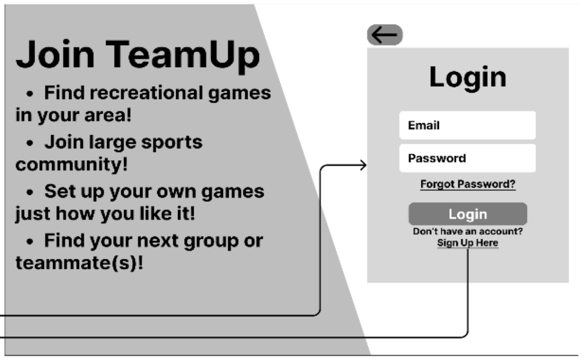
Chris Joins the team that Tyler is on

Use Case 7

- **Jane moves to San Francisco**
- **Jane plays cricket, but cannot find any games nearby**
- **Jane comes across TeamUp online and creates an account**

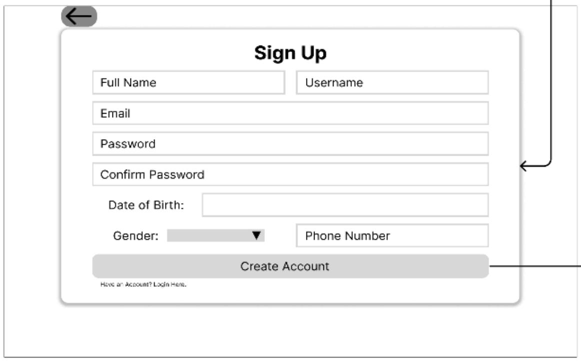

TeamUp
DISCOVER, JOIN, and HOST games for your favorite sports. Find your team.
About Us
Start Exploring
Login/Signup

Jane clicks login/sign up

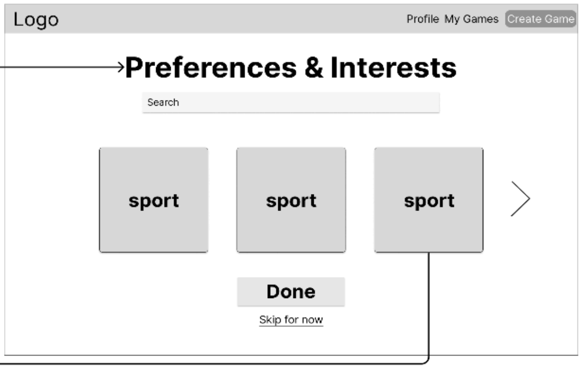

Join TeamUp

- Find recreational games in your area!
- Join large sports community!
- Set up your own games just how you like it!
- Find your next group or teammate(s)!

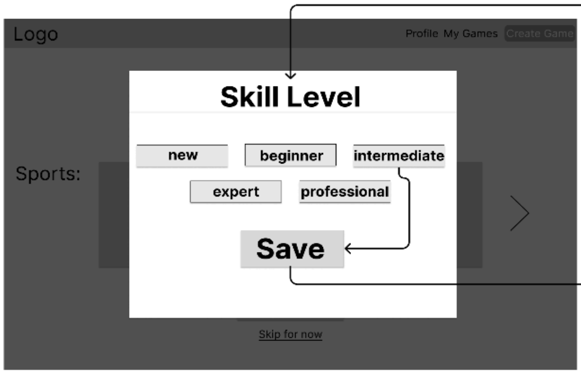
Jane clicks sign up to create an account



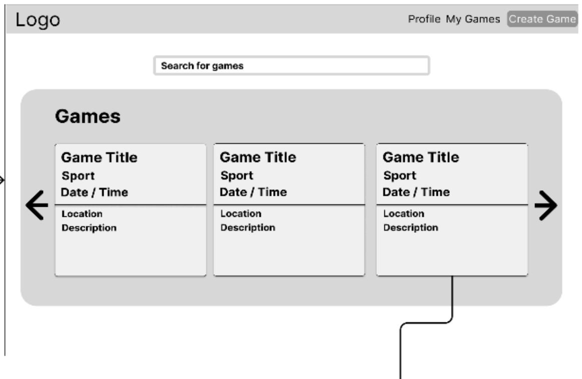
Jane fills the fields to create an account and is redirected to select her preferences

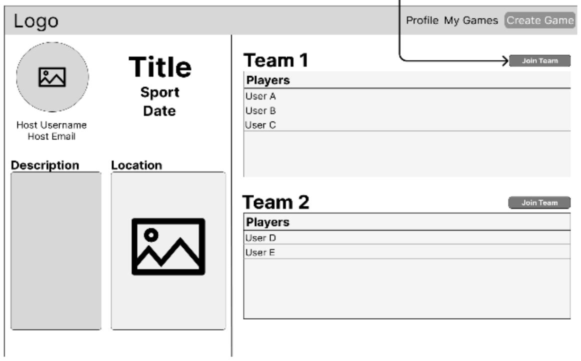


Jane selects her preferred sport and skill level for that sport



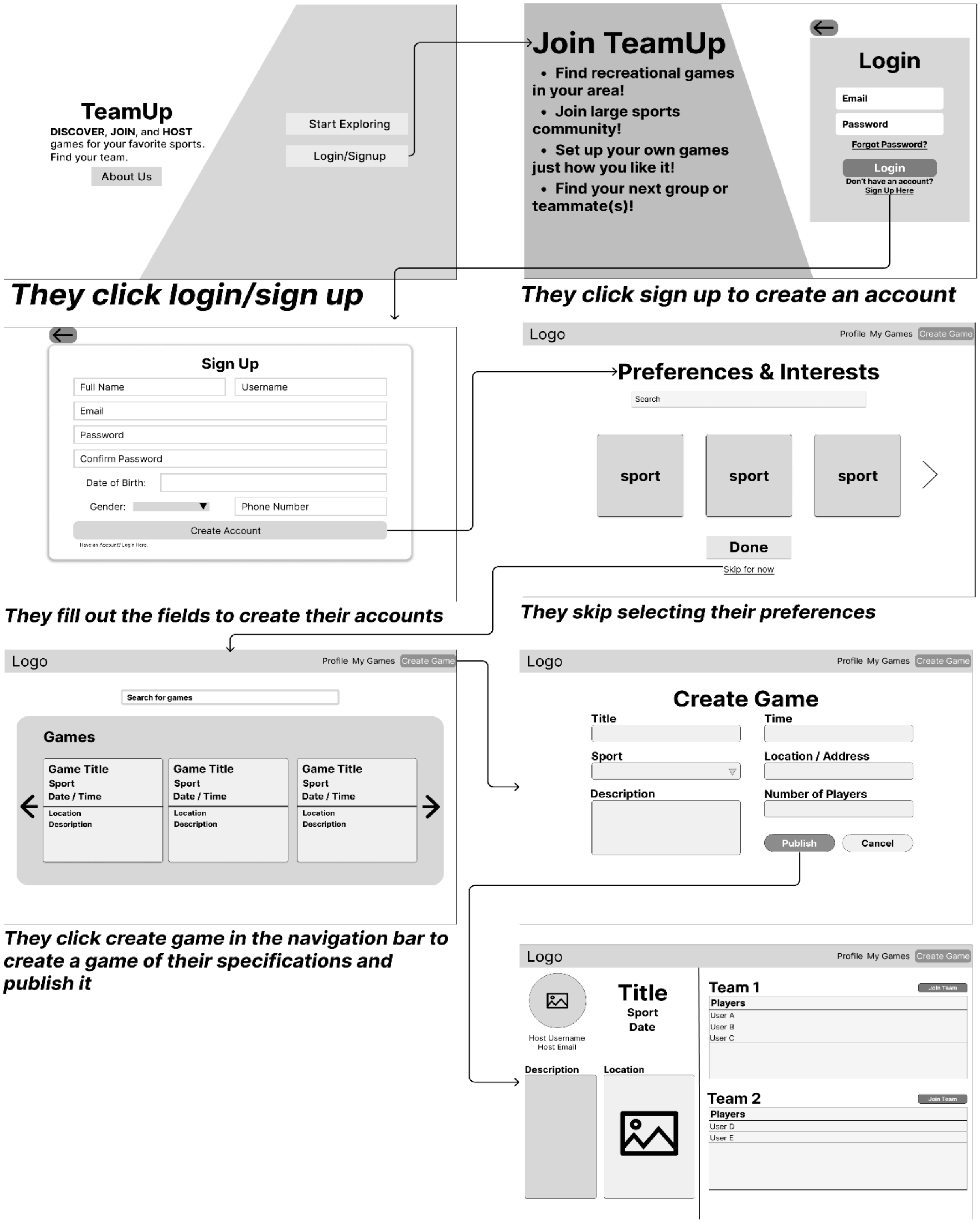
Jane is redirected to the home page and finds a game to join





Use Case 8

- **Trevor and Noah have trouble finding badminton games in their city**
- **They discover TeamUp and find that they can set up and find games that suit their availability**
- **Trevor and Noah go through the same registration process**



They click login/sign up

They click sign up to create an account

They fill out the fields to create their accounts

They skip selecting their preferences

They click create game in the navigation bar to create a game of their specifications and publish it

4. High-level database architecture and organization

Functional requirements:

1. **User**
    - 1.1 a user shall be able to join multiple games.
    - 1.2 a user shall have only one account with an unique email.
    - 1.3 a user shall host multiple games.
    - 1.4 a user shall have many reviews
    - 1.5 a user shall join many teams
    - 1.6 a user shall have many teammates
    - 1.7 a user shall have one role
    - 1.8 a user shall have many sport levels.
    - 1.9 a user shall have many sport preferences.
    - 1.10 a user shall have many tokens.
    - 1.11 a user shall write many reviews.

2. **Sport**
    - 2.1 a sport shall be played by many teams.
    - 2.2 a sport shall be chosen by many games.
    - 2.3 a sport shall be chosen as a preference by many users.

3. **Admin**
    - 3.1 an admin is a user.
    - 3.2 an admin shall manage only one school.
    - 3.3 an admin shall be able to create many tournaments.

4. **Game**
    - 4.1 a game shall be hosted by only one user.
    - 4.2 a game shall have multiple players.
    - 4.3 a game shall specify a type of sport.
    - 4.4 a game shall specify the location.
    - 4.5 a game shall be played by at least one team.

5. **Team**
    - 5.1 a team shall have at least one user.
    - 5.2 a team shall specify a type of sport.
    - 5.3 a team shall play many games.
    - 5.4 a team shall have only one name.

6. **School**
    - 6.1 a school shall have only one admin.
    - 6.2 a school shall host many tournaments.

7. **Game_Location**
    - 7.1 a game location shall be chosen by many games.
    - 7.2 a game location shall belong to only one city.

8. **Region**
    - 8.1 a region shall have multiple cities.

9. **City**
    - 9.1 a city shall belong to only one region.
    - 9.2 a city shall have multiple game locations.

10. **Review**
    - 10.1 a review shall belong to only one user.
    - 10.2 a review shall be written by only one user.

11. **Token**
    - 11.1 a token shall belong to only one user.

Non-functional requirements:

1. **Performance**
    - 1.1 The database shall retrieve data fast.
    - 1.2 The database system shall support concurrent transactions.
    - 1.3 Database queries and server-side operations shall be optimized to minimize latency and ensure smooth user experience.

2. **Storage**
    - 2.1 The database shall support persistent storage
    - 2.2 The database shall store data effectively

3. **Security**
    - 3.1 Data such as password and personal information shall be encrypted
    - 3.2 The access control for personal information shall be strictly managed.

- 3.3 Data collected shall only be used for making game and tournament management easy and shall not be shared with third parties without user consent.
- 3.4 Accessing to database shall be limited to specific users such as admin

4. **Scalability**
    - 4.1 The database shall scale easily as the service grows.
    - 4.2 The database shall maintain scalable performance when the accesses increase

5. **Capability**
    - 5.1 The database shall be available all the time
    - 5.2 The database system shall support regular backups of data.

6. **Reliability**
    - 6.1 The system shall be available 99.9% of the time during peak usage hours.
    - 6.2 The application shall be resilient to server failures and able to recover gracefully without data loss.
    - 6.3 Maintenance shall be done during off-peak hours

7. **Usability**
    - 7.1 The user interface shall be easy to use for admins.
    - 7.2 The speed of retrieving data from the database shall be fast for users.

Our group decided to use MySql as our DBMS because it is high performance, high scalability, strong security, and can run on multiple platforms. Also, most people in our group are already familiar with it.

Entity description:

1. **User (Strong)**
    - user_id: PK, numeric
    - name: composite (first name, last name), alphanumeric
    - username: alphanumeric
    - email: alphanumeric, email
    - password: alphanumeric
    - dob: multi-value, date

- Gender: char
- Phone_number: string
- created_at: date
- updated_at: date
- isEmailVerified: boolean
- role: enum

User - Admin is a One-to-One relationship.
User - Review is a Many-to-One relationship.
User - Sport is a Many-to-Many relationship.
User - Game is a Many-to-One relationship.
User - Team is a Many-to-Many relationship.

2. **Sport (Strong)**
    - sport_id: PK, numeric
    - sport_name: alphanumeric
    - description: alphanumeric

Sport - User is a Many-to-Many relationship.
Sport - Game is a Many-to-One relationship.
Sport - Team is a Many-to-One relationship.

3. **Admin (Weak)**
    - admin_id: PK, numeric
    - user_id: FK, numeric
    - school_id : FK, numeric

Admin - School is a One-to-One relationship.
Admin - User is a One-to-One relationship.

4. **Game (Weak)**
    - game_id: PK, numeric
    - sport_id: FK, numeric
    - date_time: date
    - number_of_players: Numeric
    - game_location_id: FK, numeric
    - description: alphanumeric
    - fee: alphanumeric
    - gender: alphanumeric
    - age_group: numeric

- user_id: FK, numeric
- team_id: FK, numeric

Game - User is a One-to-Many relationship.
Game - Sport is a One-to-Many relationship.
Game - Game Location is a One-to-Many relationship.
Game - Team is a Many-to-Many relationship.

5. **Team (Weak)**
    - team_id: PK, numeric
    - name: alphanumeric
    - sport_id: FK, numeric
    - user_id: FK, numeric

Team - User is a Many-to-Many relationship.
Team - Sport is a One-to-Many relationship.
Team - Game is a Many-to-Many relationship.

6. **School (Strong)**
    - school_id: PK, numeric
    - name: alphanumeric
    - location: alphanumeric

School - Admin is a One-to-One relationship.

7. **Game_Location (Weak)**
    - location_id: PK, numeric
    - name: alphanumeric
    - address: alphanumeric
    - description: alphanumeric
    - parking: alphanumeric
    - fee: alphanumeric
    - map_url: alphanumeric
    - city_id: FK, numeric

Game Location - City is a One-to-Many relationship.
Game Location - Game is a Many-to-One relationship.

8. **Region (Strong)**
    - region_id: PK, numeric

Region - City is a Many-to-One relationship.

9. **City (Weak)**
    - region_id: FK, numeric
    - city_id: PK, numeric

City - Region is a One-to-Many relationship.
City - Game Location is a Many-to-One relationship.

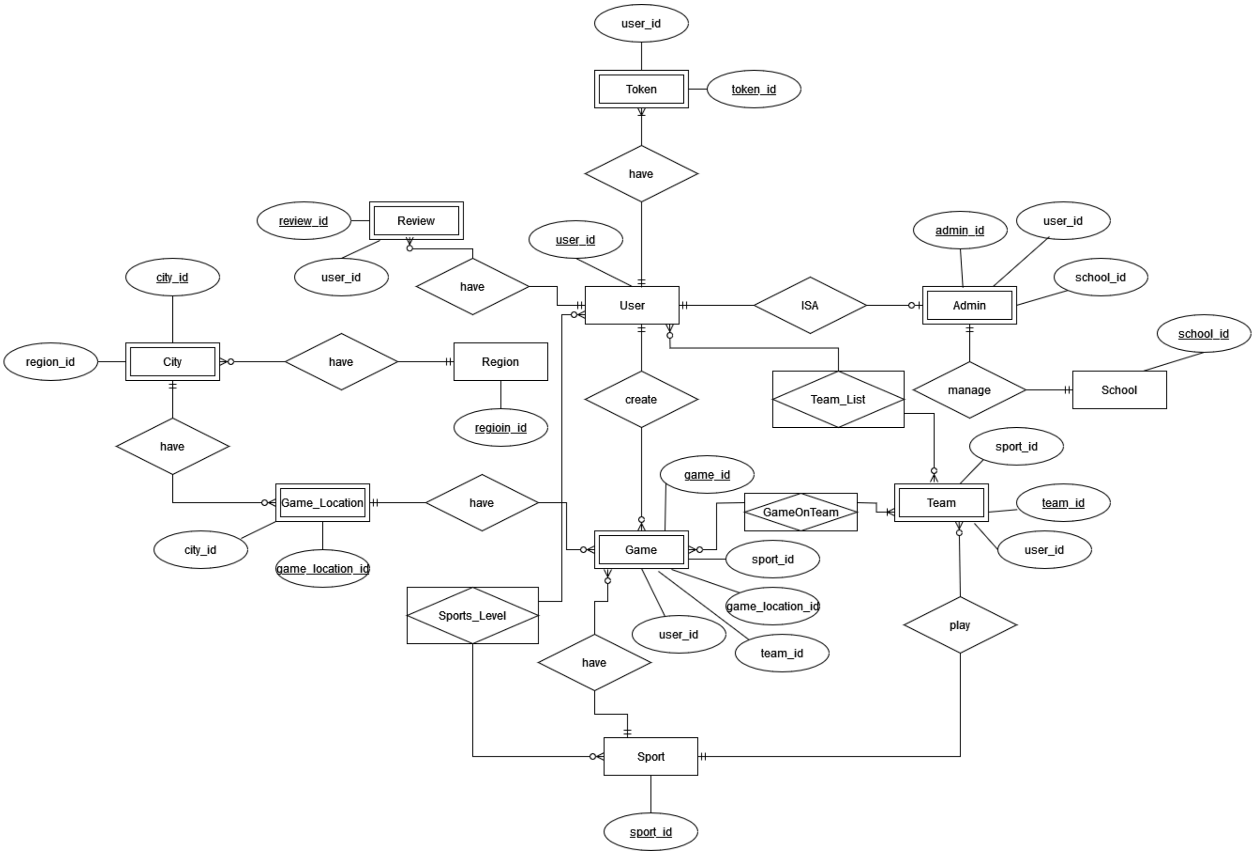
10. **Review (Weak)**
    - review_id: PK, numeric
    - user_id: FK, numeric
    - rating: alphanumeric
    - description: alphanumeric

Review - User is a One-to-Many relationship.

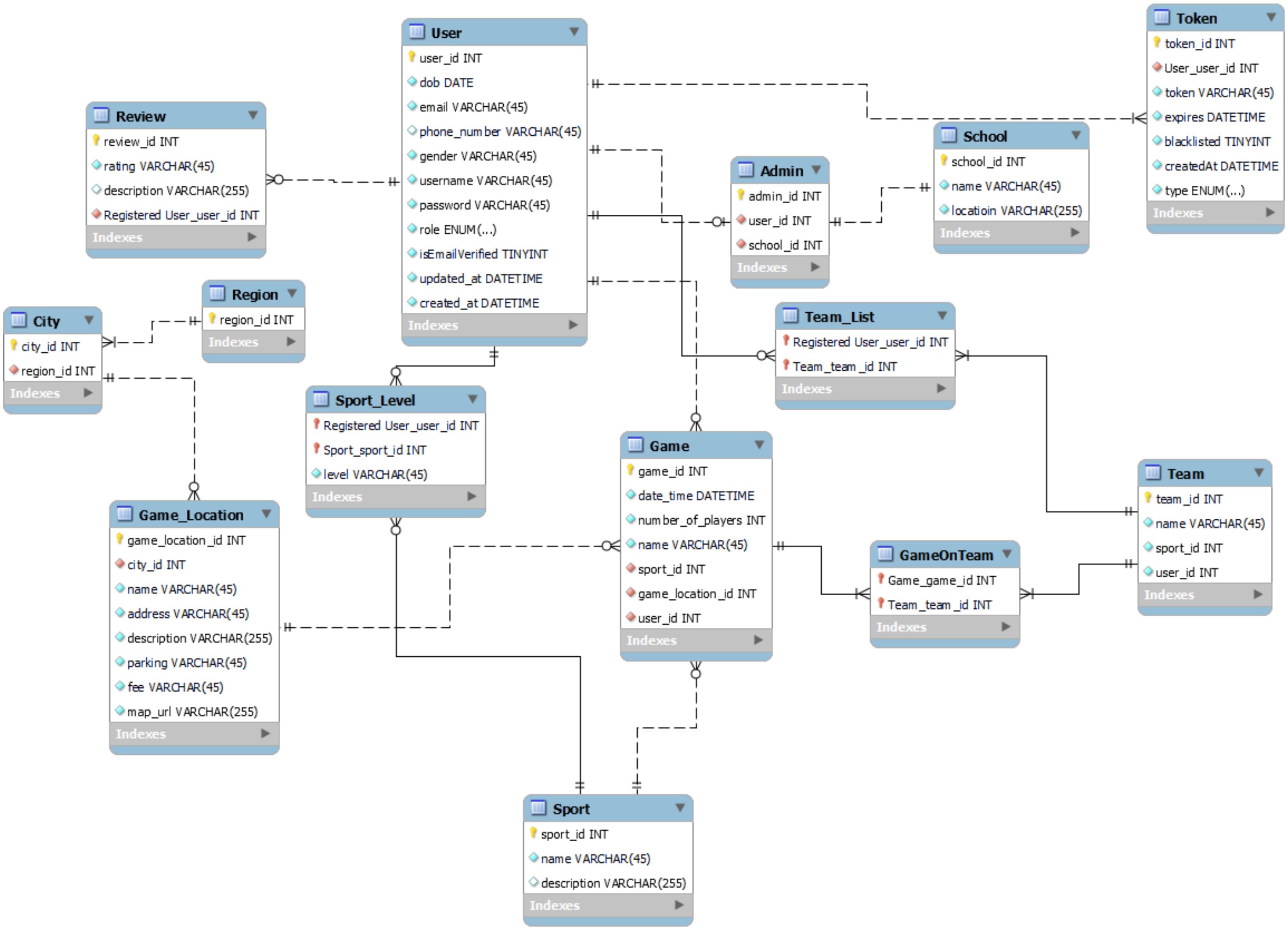
11. **Token (Weak)**
    - token_id: PK
    - user_id: FK
    - token: alphanumeric
    - type: enum
    - expires: date
    - blacklisted: boolean
    - createdAt: date

Token - User is a One-to-Many relationship.

ERD:

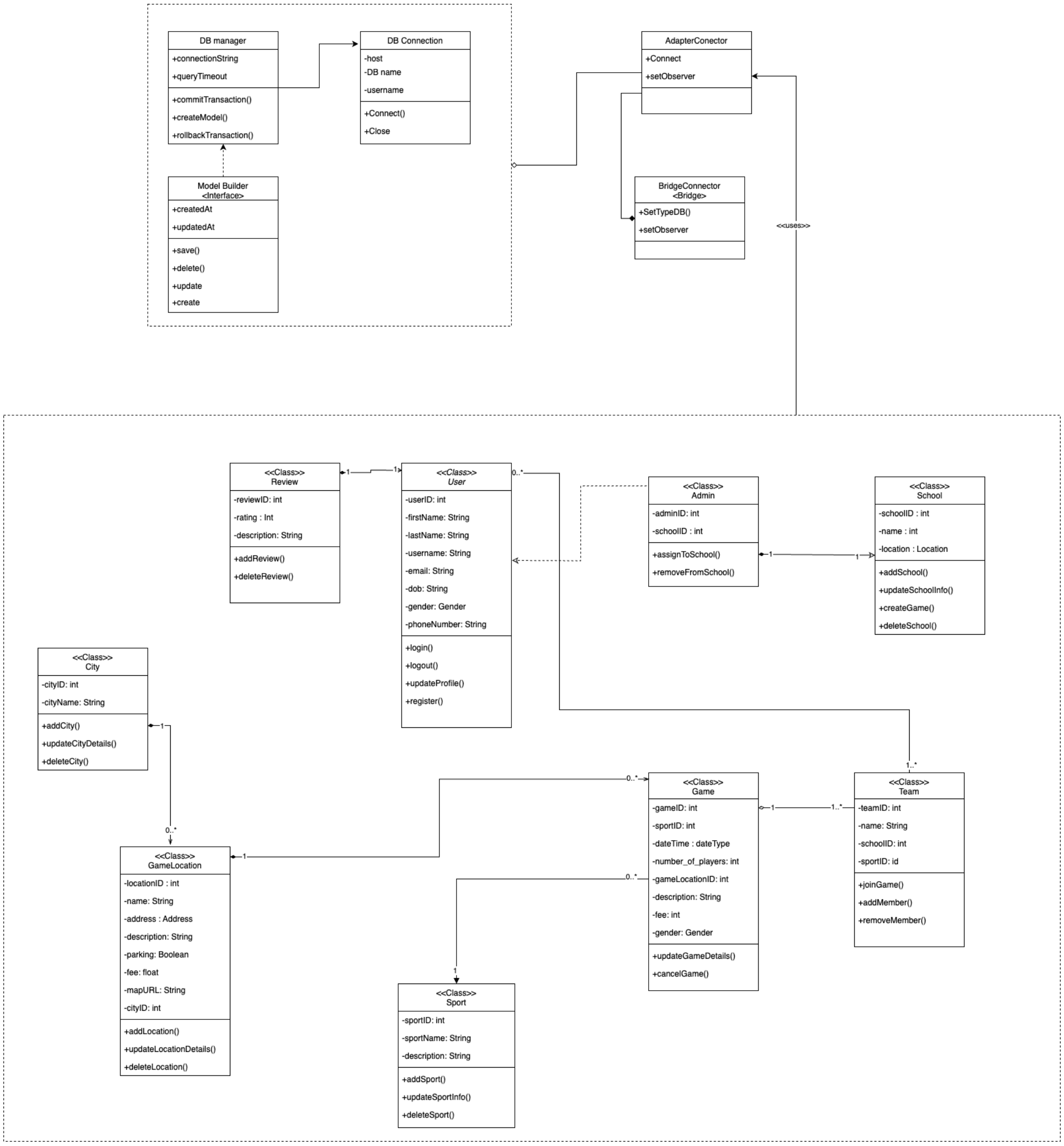

user_id
Token
token_id
have
review_id
Review
user_id
admin_id
user_id
user_id
have
school_id
city_id
User
ISA
Admin
school_id
region_id
City
have
Region
manage
School
create
Team_List
regioin_id
have
sport_id
game_id
Game_Location
have
GameOnTeam
Team
team_id
city_id
game_location_id
Game
sport_id
user_id
game_location_id
Sports_Level
play
user_id
team_id
have
Sport
sport_id

EER:

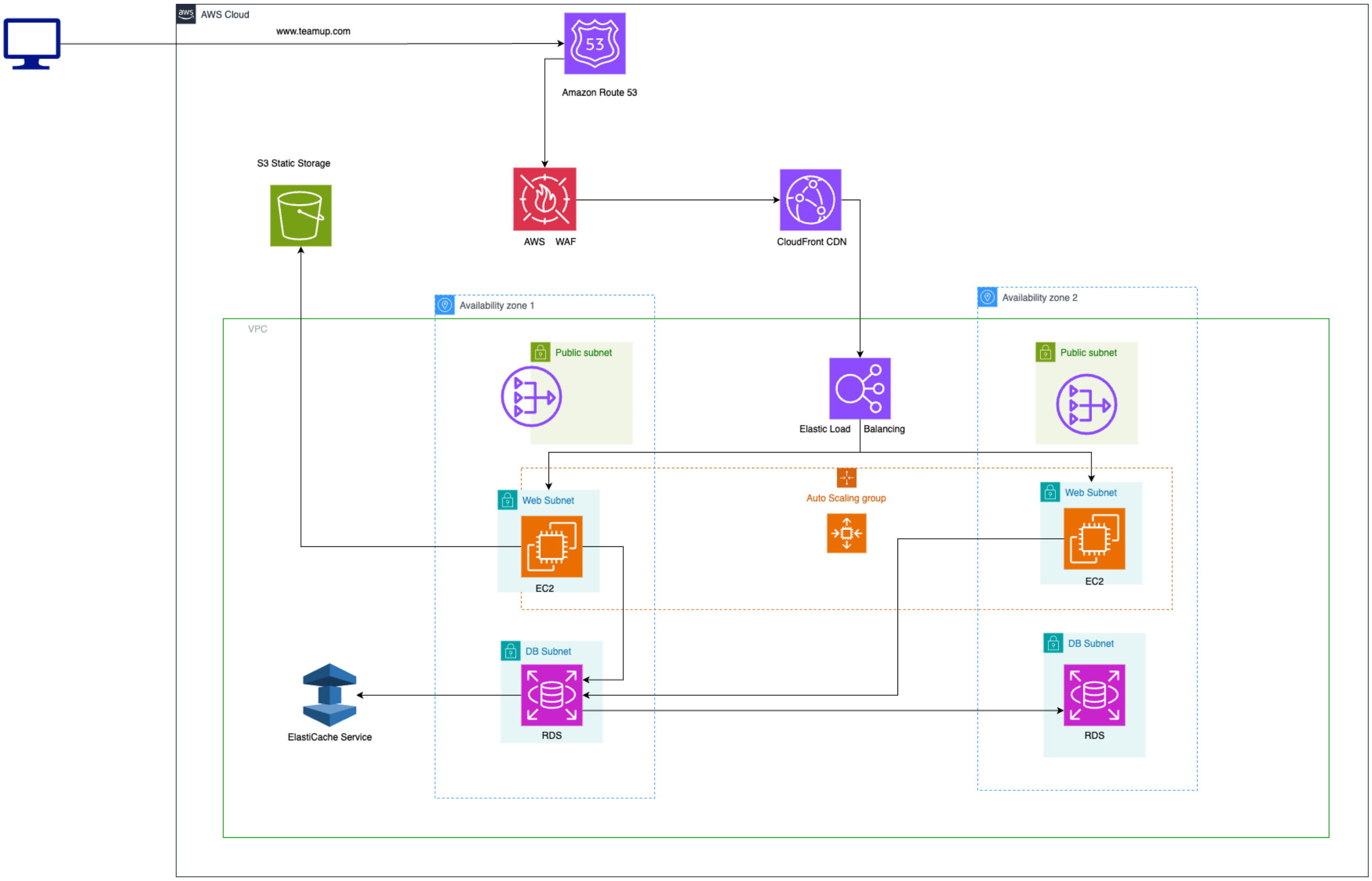

User
user_id INT
dob DATE
email VARCHAR(45)
phone_number VARCHAR(45)
gender VARCHAR(45)
username VARCHAR(45)
password VARCHAR(45)
role ENUM(...)
isEmailVerified TINYINT
updated_at DATETIME
created_at DATETIME
Indexes
Token
token_id INT
User_user_id INT
token VARCHAR(45)
expires DATETIME
blacklisted TINYINT
createdAt DATETIME
type ENUM(...)
Indexes
Review
review_id INT
rating VARCHAR(45)
description VARCHAR(255)
Registered User_user_id INT
Indexes
School
school_id INT
name VARCHAR(45)
locatioin VARCHAR(255)
Indexes
Admin
admin_id INT
user_id INT
school_id INT
Indexes
Region
region_id INT
Indexes
City
city_id INT
region_id INT
Indexes
Team_List
Registered User_user_id INT
Team_team_id INT
Indexes
Sport_Level
Registered User_user_id INT
Sport_sport_id INT
level VARCHAR(45)
Indexes
Game
game_id INT
date_time DATETIME
number_of_players INT
name VARCHAR(45)
sport_id INT
game_location_id INT
user_id INT
Indexes
Team
team_id INT
name VARCHAR(45)
sport_id INT
user_id INT
Indexes
Game_Location
game_location_id INT
city_id INT
name VARCHAR(45)
address VARCHAR(45)
description VARCHAR(255)
parking VARCHAR(45)
fee VARCHAR(45)
map_url VARCHAR(255)
Indexes
GameOnTeam
Game_game_id INT
Team_team_id INT
Indexes
Sport
sport_id INT
name VARCHAR(45)
description VARCHAR(255)
Indexes

5. High-Level Diagrams

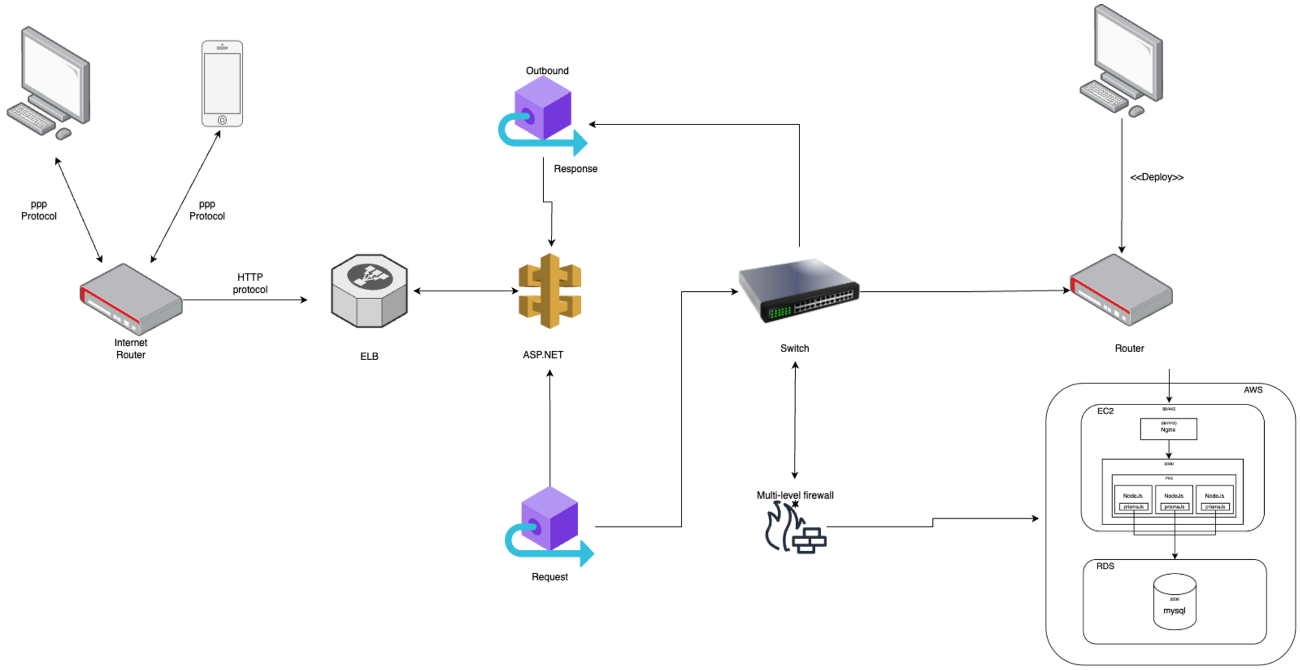
UML Class diagrams



Scalability Diagram



Application Network and Deployment Design:



6. Detailed list of contributions

Juan Estrada	9	● Data Definitions V2 ● Application Network and Deployment Design. ● [API] Game Listing Details. ● [API] Get Joined Games
Kotaro Iwanaga	9	● [API] Get Joined Games ● [API] Get, update, and delete user's preferences ● [API] for getting games user-hosted.
Cole Chiodo	9	● ⅓ Figma wireframes ● My profile page ● My games page
Martin Pham	9	● ⅓Figma wireframes ● Signup page v2 ● Preferences page ● Editing Preferences page ● Migration of existing pages to nextjs
Jaycee Lorenzo	9	● ⅓Figma wireframes ● Detail game page ● Create game page ● Card component ● Landing page v2
Areeb abbasi	8	● Automatic deployment