Requirements

Functional Requirements (MoSCoW Method)

ID	Description	Priority
1	User Registration/Sign-Up:	M
1.1	The system must ask for a unique username for the user to register.	M
1.2	The system must require a valid and verifiable email address.	M
1.3	The system must require a password that is six characters long and contains at least one numerical character.	M
1.4	Users should be able to register as players or hosts.	M
2	Login/Logout:	M
2.1	The system should allow users to log in using either their username or email, along with their password.	M
2.2	The user should be able to log out once she has used the system.	M
3	Profile Management :	M
3.1	The user must be able to edit/update the following fields: ● Name ● Username ● Profile picture ● Bio ● Player type ● Password ● Default location ● Contact information	M
4	Account Management	M
4.1	The system should allow users to delete their accounts.	M
4.2	The system should allow users to retrieve forgotten password.	M
5	Event Management:	M

5.1	The system should allow users to create a game event.	M
5.2	The user should be able to edit the following details about the event: ● Date ● Time ● Location ● Fee per person ● Description ● Number of players per team ● Type - kickabout/competitive/tournament	M
5.3	Organizers should be able to cancel events.	M
6	Participation:	M
6.1	The users should be able to join games when a vacant spot is available.	M
6.2	Users should be able to vacate their spots.	M
6.3	The system should not allow users to participate in more than one event at the same time.	M
6.4	The system should allow users to search for events based on : ● Date ● Time ● Location	M
7.	Hosting and Tournaments	M
7.1	Official companies should be able to register their facilities as event locations.	M
7.2	Users should be able to set-up tournaments with the following fields: ● Date ● Time ● Location ● Number of teams	M
7.3	Users should be able to cancel tournaments.	M
7.3	Users should be able to opt into teams on a tournament-to-tournament basis.	M
7.4	Users should be able to opt out of teams.	S

7.5	Users should be able to view locations.	
8	Additional features and recommendation engine	C
8.1	Users could be able to view and/or follow other players and locations.	C
8.2	The system could recommend events based on data from previous participation and location.	C
8.3	Users could send an emergency alert when an event is in need of players or due to last minute cancellations.	C
8.4	Organizers could invite players to join events and tournaments based on recommendations by the system.	C
8.5	Report absentees who do not show up for events	C

Documented Use Case

Search and Join Game	
Preconditions	1. Server is in a working state. 2. User is already registered. 3. User is logged into the system. 4. Events have been previously created in the system.
Flow of events	1. User clicks on search button. 2. User enters the following: 2.1 Date 2.2 Time 2.3 Location 3. The system displays results. 4. User selects an event from search results. 5. The system displays event details 6. From here the user can either: 5.1 Join the event 5.1.1 Go back to step #2 to update search criteria 5.1.2 Go back to step #3 to select a different event 7. The system updates the event details after user joins. 8. The system shows updated event details and successful participation by the user.
Postconditions	1. The database is updated as per event and user details. 2. The user is marked as busy during the time of event. 3. The buyer receives a confirmation email.

NOUN/VERB Analysis

We elicit the nouns and verbs relevant to our application's class diagram

Class Candidates (Nouns)

Player	Moderator	Location
Host	Game	Tournament
Location	Team	User
Engine	System	Arena

Operation Candidates (Verbs)

registerAccount()	editProfile()	registerArena()
emailValidation()	editGame()	editArena()
sendRegConfirmation()	editTournament()	searchGame()
login()	logout()	selectEvent()
organizeGame()	organizeTournament()	joinGame()
openInMaps()	updateLocation()	leaveGame()
getLocation()	joinTeam()	joinTournament()
changePassword()	leaveTeam()	leaveTournament()
deleteAccount()	removePic()	uploadPic()
addTeam()	removeTeam()	addPlayer()
removePlayer()	verifyLocation()	

Responsibility-Driven Analysis

In addition to noun/verb analysis we employ responsibility-driven analysis to outline Classes, their Responsibilities and their Collaborators (CRC Cards).

System	
Responsibilities	**Collaborators**
Main overall hub for the functionality of the web service. Includes event organization, user registration and profiles, and all other tasks outlined by the requirement elicitation.	Player,Location, Game, Host, Arena, Tournament, Moderator, Team, Engine, Organizer

Player	
Responsibilities	**Collaborators**
Holds the information for a user of type player. This user can can register and edit their profile, participate and organize events and opt out of them.	Location, Game, Arena, Tournament, Team, Engine,Organizer

Host	
Responsibilities	**Collaborators**
Holds the information for a user of type host. This user can can register and edit their profile. The main functionality accessible to this user is the registration of an arena. They can organize and cancel events.	Location, Game, Arena, Tournament, Moderator, Engine, Organizer, System

Location	
Responsibilities	**Collaborators**
Maintain data for GPS location of different users, arenas and events.	Game, Arena, Tournament, Engine, Organizer, System

Moderator	
Responsibilities	**Collaborators**
Validate registration of users and new arenas.	Player, Host, Arena, System

Game	
Responsibilities	**Collaborators**
Maintain data for game events which includes date, time, location, description and participants.	Player, Host, Organizer, Arena, Tournament, Engine, Location, System

Tournament	
Responsibilities	**Collaborators**
Maintain data for tournament events which includes date, time, location, description, teams, fixtures and individual participants.	Player, Host, Organizer, Arena, Team, Engine, Location, System, Game

Team	
Responsibilities	**Collaborators**
Maintain data for teams which are in a tournament, including all the individual participants.	Player, Tournament, System

User	
Responsibilities	**Collaborators**
Maintain data for a user who organizes events such as games and tournaments.	Player, Host, System

Arena	
Responsibilities	**Collaborators**
Maintain data such as GPS location and description for the actual location that the games are held. This would include mostly registered places by hosts and/or frequently used locations.	Player, Host, Organizer, Engine, Location, System

Engine	
Responsibilities	**Collaborators**
Responsible for analysing data and user behaviour to recommend and display best options for games, arenas and tournaments.	Player, Host, Arena, Tournament, Location, System