**Functional Requirements Template - Urban Hunt**Jonathan Petersen, Anthony Kiesel, Kenny Fryar-Ludwig, Shawn Fairbourn, Kris Mygrant, Hira Ahmed

**1. Introduction and Context**

This system is an extensible app for Android smartphones that allows users to connect and work together in real-time. The initial implementation of this app will be directed toward real-world games that are augmented by the use of this system. When players open the app, they have the ability to connect to other Urban Hunt users. The initial game mode will allow players to assign teams and set an objective.

This system will allow users playing games via phone to connect together like never before. It will bring a new digital dimension to classic outdoor games. Humans will no longer be able to lie that they haven't been infected by a zombie. Fitness users will be able to run faster and harder as they are chased by digital zombies. Users will learn about classic games and their rules that they may not have known the joy of before. Users will now be able to continue staring at their phone and get exercise at the same time.

This app addresses the problem of players keeping track of other night game colleagues. It also provides a solution to locating where team members may be, other members, locations of game play, and boredom.

**2. Users and their Goals**

**Actors:**

* Host User
* Client Users
* Google Maps
* Third-party “Multiplayer” server
* Android™ Smart Device (Phone/Tablet)

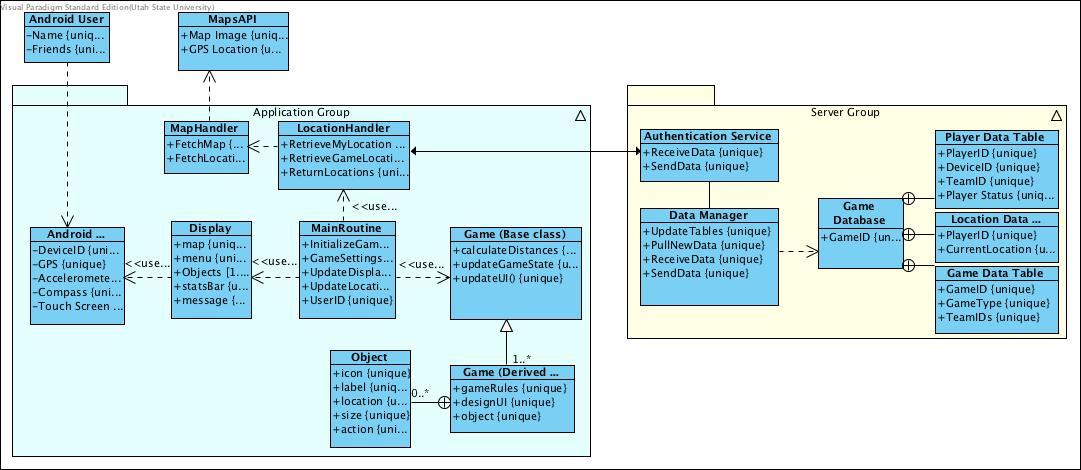
**Users:** Fitness Junkies, Android users, “Night Games” people

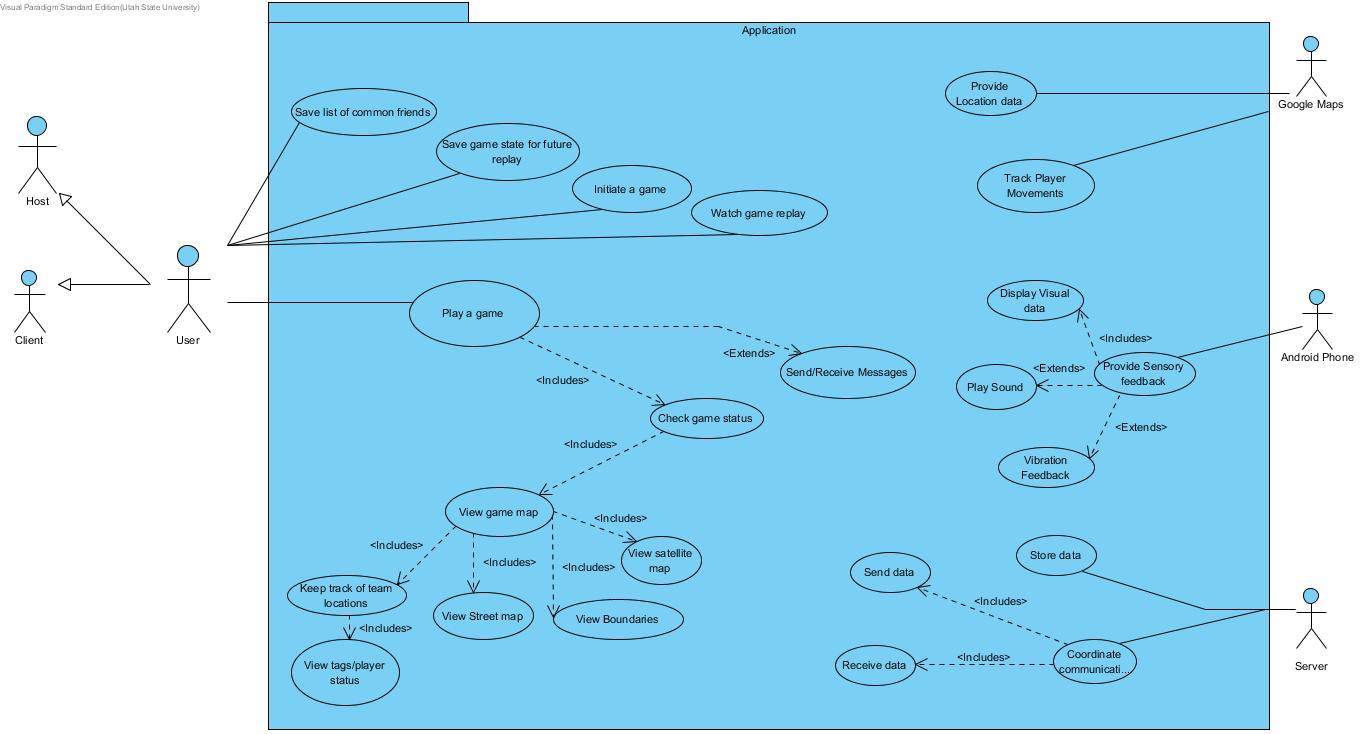
**Connected Systems:** MySQL server, Google Maps, Android Devices

**Use Cases:**

1. Play a variety of group games (Users)
   1. Track team locations
   2. View tags/Player status
   3. Send/recv messages to/from team
   4. Send/recv messages to/from all players
   5. View digital boundaries
   6. View street map
   7. View satellite map of local area
   8. Update group on current objectives
   9. Initiate game
   10. Save game state for future replay
   11. Replay Mode
   12. Save list of commonly connected friends
2. Track independent GPS stats (Google Maps)
   1. Provide location data
   2. Track player movements
3. Provide Sensory Feedback (Phone)
   1. Vibration Feedback
   2. Play sound
   3. Display Visual data
4. Server
   1. Store data
   2. Send data
   3. Receive Data

**Class Diagram**





**3. Functional Requirements**

1. *Create a game*
   1. *We will need a screen that allows users to host a new game.*
   2. *System to allow users to join an existing game*
   3. *System to keep track of who is currently active in a given game*
2. *Play a game (Application)*
   1. *A set of rules for specific types of games*
   2. *An engine to provide that set of rules to current players.*
   3. *Verification that users play by the rules*
   4. *Record current location from phone*
   5. *Display the location of other players*
   6. *Display the objectives of the current game*
   7. *System to determine when a game is finished.*
   8. *Sensory feedback (Visual, audio, haptic) based off of game status*
   9. *Network connection component to talk to Location Server*
   10. *System to determine actual location from GPS coords*
   11. *System to determine how users are interacting (collisions)*
   12. *Capability to save game data to resume play later*
3. *Keep Statistics about current / past games (Fitness Users)*
   1. *Local storage of distance travelled (calories burned?)*
   2. *Feature to replay the locations of users after completion of a game*
4. *Location server to keep track of player location / data*
   1. *Authentication component to prevent cheating*
   2. *Data storage for Player Location*
   3. *Data storage for Player Teams*
   4. *Data Storage for Games being played*
   5. *Data storage for messages / other misc information about a game*
5. *Receive messages from the host of the game*
6. *AI component to simulate other players*
7. *A splash screen to keep already whipped user engaged.*

**4. Non-functional Requirements**

1. *Development will follow the Agile method, including weekly meetings.*
2. *The system will be backed up, including backups on BitBucket/DigitalOcean*
3. *Any changes made to the project will have to be unanimously voted on.*
4. *Reasonably responsive GPS location for all players*
5. *Should be functional on all supported Android platforms.*
6. *Will be accurately documented in code and in the application/comments*
7. *System will be capable of supporting an arbitrary number of users*
8. *System will be able to support users globally*

**5. Future Features**

*Powerups*

*Facebook Integration*

*Game-Specific Achievements*

**6. Glossary**

GPS - Global Positioning System

Game - A set of objectives for users to complete

Android Phone - Device running Android OS with GPS functionality.