Table of Contents

1 Introduction .........................................................................................................................4
  1.1 Goals and Objectives .................................................................................................4
  1.2 Statement of Scope .....................................................................................................5
  1.3 Software Context .........................................................................................................6
  1.4 Major Constraints .......................................................................................................6
2 Usage Scenario ....................................................................................................................7
  2.1 User Profiles ...............................................................................................................7
  2.2 Use-Cases ....................................................................................................................8
    2.2.1 Create Game ..........................................................................................................8
    2.2.2 Place Waypoint ....................................................................................................9
    2.2.3 End Game ...........................................................................................................10
    2.2.4 View Statistics/Leaderboard ...............................................................................11
    2.2.5 Join Game ...........................................................................................................12
    2.2.6 Ping ....................................................................................................................13
  2.3 Special Usage Considerations ....................................................................................14
3 Data Model and Description ...............................................................................................15
  3.1 Data Description ..........................................................................................................15
    3.1.1 Data Objects .......................................................................................................15
    3.1.2 Relationships ......................................................................................................17
    3.1.3 Entity-Relationship Diagram ..............................................................................19
    3.1.4 Data Dictionary ....................................................................................................19
4 Functional Model and Description ......................................................................................20
  4.1 Function Descriptions .................................................................................................20
    4.1.1 Start Program .......................................................................................................20
    4.1.2 View Games .........................................................................................................20
    4.1.3 Create Game ........................................................................................................21
    4.1.4 Join Game ............................................................................................................22
    4.1.5 Manage Game .....................................................................................................23
    4.1.6 Ping ......................................................................................................................25
    4.1.7 Exit Program ........................................................................................................26
  4.2 Software Interface Description .....................................................................................27
    4.2.1 External Machine Interfaces ...............................................................................27
    4.2.2 External System Interfaces ..................................................................................27
1 Introduction

This section will provide an overall description of the project and pave a path for the rest of the document. This section will go over goals for the project and objectives we intend to achieve with this software. The rest of the document will describe in detail the functionality and behavior of the software that is being designed.

1.1 Goals and Objectives

The overall scope for this project is to develop an iPhone application that will involve multiple people to play a game with each other. The main objective of the game is to find out where the host player places multiple waypoints until the last one is located. This will happen through clues which will be pictures. The user will then use the picture as well as a written clue to try and figure out where the host has placed a way point. Using the GPS built-in to the iPhone, the person will “ping” to see how close they are to the waypoint. Once the player is within an appropriate distance of the waypoint the ping will return that they are in the correct spot and they are given the next picture and clue corresponding to the next waypoint. Once a player has completed finding all of the waypoints the game will end and the winner will be displayed to all players and the host.

There a couple of main goals that we hope to accomplish with this project. The first goal would be to successfully complete a program using a programming platform that no member of the group is familiar with. Another main goal is to have the software be very user friendly to the point where a non-tech savvy person could handle using the program with minimal issues. The last goal we want to be successful with is integrating this program with the built-in GPS of the iPhone.
1.2 Statement of Scope

This piece of software that we are designing will have several parts integrated with each other. The first is the program app itself on the iPhone which will have to interact with the GPS system installed in the phone itself. This program will be interacting with the a web server that we will set up online which will keep track of different games that are being played. The web server will also keep track of all important objects associated with each game such as photos and clues and coordinates of the waypoints. The biggest challenge will be to integrate the web server with the program and have it function well.

Major inputs that will be required for this program will first come from the host. Since the host is the start of every game we will need his inputs first. The host will need to place his waypoints which can be a single waypoint to as many as the host would like to place. Each waypoint should be at a different GPS coordinate. This list of locations will be sent to the web server to be stored and checked when it is needed. Once the host has all positions they wish to have they will have to initiate the start of the game. Once the game is initiated players may join as they wish throughout the game. Inputs come in from the players in the form of pings. These pings will be sent to the web server with the coordinates given by the GPS which are then compared to the coordinates for the given waypoint. The response that will be given will be a hot or cold statement explaining you are getting closer or farther to the waypoint. The total pings used by each user will be counted and that is determined who is the winner of the game. The game is also able to be terminated at the host’s discretion. The players can also leave the game whenever they choose to.
1.3 **Software Context**

In terms of this software in the “big picture” we are looking for this to be used by any iPhone user. It would be nice to get this involved with game center which will let a user know of any of his friends that have this application on their device. This will make match making for the games a lot easier then inviting people to join your game. If the product turns out complete by the end of the semester then we will look to make it a downloadable app for the iPhone. One of the market constraints will be to deny people with the iPad and the iTouch from downloading it. The iPhone is the only Apple product that has working GPS already built-in. Since the game is largely based on the use of the GPS system it would not make sense for having users play the game without having access to the GPS.

1.4 **Major Constraints**

There will be several constraints on this software when it comes to the design and testing of this software. One of the constraints was talked about in the previous section. This would be allowing only people with an iPhone to download the app because all other Apple products do not have a working GPS system in place. The game works around the GPS and relies heavily on the GPS to be fully functional and accurate. We only have access to two iPhones within our group making it a little difficult to test a game that would involve multiple players along with the host. The only scenario we will be able to test will be one host and one player. Another constraint will be testing it in any area. We will probably be constrained to testing this software in the Morgantown area because of transportation and time issues not allowing us to be able to get outside of the area and testing it in different environments.
2 Usage Scenario

This section of the document will provide all of the use-cases that were extracted during elicitation. This section will provide in depth description on the different interactions with the software.

2.1 User Profiles

This piece of software will include only two different types of user profiles: host and player.

**Host:** The host user will be the one who establishes a new game. Their job is to open a new game and set the waypoints for the rest of the users to search for. Along with each waypoint, the host will take snapshots and write text clues to help guide the users in the right direction of where they placed the waypoint to be located. The host will be able to view statistics of the game that they are currently hosting. The statistics will include how many times each of the players have “pinged” the game and how many waypoints the user has successfully located. The leader board will be displayed in list form sorted first by the highest number of waypoints located at the top to the lowest. From there, it will then sort by the number of pings. The person at the top of the list will have the most waypoints found as well as the lowest number of pings indicating that they are currently in the lead of the game.

**Player:** The player user will be able to join an already created game. The objective of the player user will be to successfully locate the waypoints placed by the host and to do it in the shortest amount of pings. The player will have a few options they can do. They will be able to ping the system to see if they are hot or cold from the waypoint’s location. Along with the ping, they will be able to view the clue as many times as they like. The last feature they will have is to be able to view the leader board/statistics which will display the same thing as the host as well as give them their current ranking in the game.
2.2 Use-Cases

2.2.1 Create Game

Use Case: Create Game

Primary Actor: Host

Stakeholders and Interest:

Host – wants to create a new game for people to find his/her waypoints
Player – wants the host to create a game so they can start the search

Entry Conditions:

- The host is logged in and has the application for the iPhone

Trigger Condition:

- Host is logged into the app

Exit Conditions:

- None

Flow of events:

1. User logs into the system
2. User hits the create game button
3. Game is created

Exceptions:

- User is using an iPad or an iTouch which does not have a GPS
2.2.2 Place Waypoint

**Use Case:** Place Waypoint

**Primary Actor:** Host

**Stakeholders and Interest:**

Host – wants to place waypoints for players to locate

Player – needs locations to find provided by the host

**Entry Conditions:**

- The host is logged in and has created a game lobby

**Trigger Condition:**

- Host is logged into the app and has already created a new game

**Exit Conditions:**

- User finishes placing their desired waypoint

**Flow of events:**

1. User logs into the new game created
2. User walks to the spot where they wish to drop a waypoint
3. User uploads an image and/or a text clue for their location

**Exceptions:**

- User has not created a new game
- User is not using a device with a working GPS
2.2.3  End Game

**Use Case:** End Game

**Primary Actor:** Host

**Stakeholders and Interest:**

Host – wants to terminate the game and declare a winner

Player – find out who wins the game

**Entry Conditions:**

- The host is logged in and has a game in progress

**Trigger Condition:**

- Host is logged into the app, has an existing game in progress, and clicks the end game option

**Exit Conditions:**

- User confirms that they want to end the game

**Flow of events:**

1. User enters the lobby of the existing game
2. User clicks on the end game button
3. User confirms they would like to end the game
4. Game terminates and announces a winner

**Exceptions:**

- User has not started a new game
- User is not using a device with a working GPS
- User is not hosting any games at the moment
2.2.4 View Statistics/Leader board

**Use Case:** View Statistics/Leader board

**Primary Actor:** Host, Player

**Stakeholders and Interest:**

Host – wants to view who is in the lead of their hosted game

Player – find out their personal rank in the game/see the complete leader board

**Entry Conditions:**

- The player or host is logged into a current game session

**Trigger Condition:**

- Player or host is logged into a game and clicks on the view statistics/leader board button

**Exit Conditions:**

- User clicks on the back button

**Flow of events:**

1. User enters the lobby of the existing game
2. User clicks on the view statistics button

**Exceptions:**

- User is not in a game
- User is not using a device with a working GPS
2.2.5 Join Game

**Use Case:** Join Game

**Primary Actor:** Player

**Stakeholders and Interest:**
- Host – wants players to join their created game
- Player – wants to join in a new game

**Entry Conditions:**
- The player is logged into the app

**Trigger Condition:**
- Player click on the join game button

**Exit Conditions:**
- User chooses the game to join

**Flow of events:**
1. User enters the app
2. Chooses the join game button
3. Chooses from a list of suggested games to join in their general area
4. Game is joined once choice is selected

**Exceptions:**
- User is not using a device with a working GPS
2.2.6 Ping

**Use Case:** Ping

**Primary Actor:** Player

**Stakeholders and Interest:**

Host – wants to see how close players are getting to waypoints

Player – wants to check to see if they are on the waypoint or how close they are

**Entry Conditions:**
- The player is logged into a current game session

**Trigger Condition:**
- Player is logged into a game and clicks on the ping button

**Exit Conditions:**
- User clicks on the back button

**Flow of events:**
1. User enters the lobby of the existing game
2. User clicks on the ping button

**Exceptions:**
- User is not in a game
- User is not using a device with a working GPS
2.3 **Special Usage Considerations**

There are a couple of special requirements that will be needed for this software. The first one would involve that anyone who would want to use this software would require an iPhone with a working GPS component built-in. The game that this document is describing will not be able to properly function without this feature. The only piece of Apple products that includes the GPS is the iPhone. The iTouch as well as the iPad or any other apple product will not work with this application. Another special requirement to use this software would be to make sure people in the same general area are using this software. It would not be feasible for a user to join a game that the host placed waypoints hundreds of miles away. The last special requirement would be to have an account on game center. Our intentions for this software are to link up with game center to be able to do matchmaking with your friends. It will also help with finding games local to where you are geographically located.
3 Data Model and Description

This section describes the information domain of the software.

3.1 Data Description

There will be several pieces of data that will be kept for tracking. The data objects that will be involved with this program will be host, players, pings, games, waypoints, and ranks.

3.1.1 Data Objects

**host:** This data object will hold the information of the host of the game

Information included with the host:

Host name; gameID; number of waypoints placed;

**Player:** This data object will hold information of the player in the game

Information included with the player:

Player name; gameIDs; number of pings in game; current rank;

**Pings:** This data object will hold information of the pings from the players

Information included with the ping:

Number of pings; location of pings;

**Game:** This data object will hold information of the game

Information included with the game:

gameID; host name; number of players; playerIDs; waypoint location; number of waypoints;
**Waypoints:** This data object will hold information of the waypoints placed by the host

Information included with the waypoint:

Waypoint number; waypoint location; waypoint clue; waypoint picture;

**ranks:** This data object will hold information of the ranks of the players

Information included with the rank:

playerID; current rank;
3.1.2 Relationships

**Host:**

Players: Has many players join his/her game
Pings: NONE
Game: Hosts a game
Waypoint: Places waypoints
Ranks: NONE

**Player:**

Host: Joins a hosts game
Pings: creates a ping on the location
Game: Joins a game
Waypoint: NONE
Ranks: Receives a rank

**Ping:**

Host: NONE
Player: created by a player
Game: NONE
Waypoint: NONE
Ranks: NONE

**Game:**

Host: Created by Host
Player: joined by player
Ping: NONE
Waypoint: NONE
Ranks: NONE

**Waypoint:**
Host: Host creates Waypoints
Player: NONE
Ping: NONE
Game: NONE
Ranks: NONE

**Ranks:**
Host: NONE
Player: Player obtains a Rank
Ping: NONE
Game: NONE
Waypoint: NONE
3.1.3 Entity-Relationship Diagram

3.1.4 Data Dictionary
1. Host name – String indicating the host’s username
2. gameID – random integer ID number assigned to each created game
3. number of waypoints placed – integer number indicating the number of waypoints
   placed by the host of the specific game
4. Player name – string indicating the player’s username
5. number of pings in game – counter keeping track of the number of pings executed
   by the player
6. current rank – integer assigned to the current rank of the player
7. location of pings – GPS coordinates of each ping executed by the user
8. number of players – current number of players playing in the game
9. waypoint location – GPS coordinates to the current waypoint
10. Waypoint number – number assigned to a specific waypoint given by the host
11. waypoint clue – the text clue given by the host for the specific waypoint
12. waypoint picture – picture given as an additional clue for the specific waypoint
4 Functional Model and Description

4.1 Function Descriptions

4.1.1 Start Program

4.1.1.1 Initializes the program, bringing the user to the main display window where they can choose to create a new game or join any game they may have been invited to.

4.1.1.2 No flow diagram needed.

4.1.1.3 No inputs. Outputs may include the user deciding to create a game, join a game, manage a game, or exit the program.

4.1.1.4 Makes a call to display the main page which shows the user any games they may be involved with (either host or player) and provides them options to create a new game or join a game they have been invited to play (which is visible to the user on the main display).

4.1.1.5 The application will be required to link up with our provided game server (where games will be stored and hosted from) as well as linking up with Game Center on the iPhone. Both of these may result in slow loading and if either fails to connect, some usability of the application will be forfeit until a connection is established.

4.1.1.6 Application must be as small and efficient as possible.

4.1.2 View Games

4.1.2.1 Retrieves any games that the user is involved with. Sorts them based on whether or not the user is each game’s host or a player in that game. Displays the games to the main display. Called whenever the program starts up or the user returns to the main display. Accessing a displayed game will allow the user to continue or manage that game depending on their privileges for each game.

4.1.2.2
4.1.2.3 Inputs: User account (Game Center account). Outputs: games the user is involved with along with any games the user has been invited to play in. Formatted and displayed to the main display depending on the user's privileges with each game.

4.1.2.4 Makes a request to the game server, requesting a list of games that the provided user account is involved with.

4.1.2.5 The connection to the game server is required to view games. An unsuccessful or slow connection will have undesirable results on loading time.

4.1.2.6 The connection, request, and request fulfillment should be quick and hardly noticeable to the user.

4.1.3 Create Game

4.1.3.1 Allows the user to set up a new instance of scavenger hunt that his or her friends can later join and play in. Brings the user to the create game display where they can set waypoints (minimum one waypoint per game required), cancel the creation, or finish the creation and post the game up after inviting friends to play.

4.1.3.3 Inputs: User determines that they wish to create a game. Outputs: Newly created game containing at least one waypoint and optional clues sent to game server. Game invites sent to selected friends.
4.1.3.4 This function will have a variety of other functions involved. The user will be required to set waypoints using the set waypoint function as well as provide any picture or textual clues as to where each waypoint is. Setting a waypoint involves storing the current GPS location of the user and ultimately sending that to the game server. The user can cancel the game creation at any time, returning them to the main display. Once all the desired waypoints are set (they are numbered by the order they are entered), the user selects to create the game. Doing so will create the game on the game server until the host decides to close the game. Afterwards the user can invite friends to play the game via the manage game function from the main display.

4.1.3.5 Recording the current GPS location of the user may prove troublesome, especially in locations with poor cellular reception. Therefore, placing waypoints will be only possible with a good GPS signal. As well, posting the game to the server may take time if the user does not have a strong connection to any network.

4.1.3.6 Creating a game should be simple and intuitive. Waypoints for players will be numbered by the order the host inputs them (i.e. the last waypoint entered is the "finish"). A minimum of one waypoint per game will be required to post a game and perhaps a maximum will be set in the future. Inputting clues are optional to the host and should also be editable via the manage game function.

4.1.4 Join Game

4.1.4.1 Puts the user into the selected game. Displays to them the current statistics and leader board for that game and provides options to either start the game (activating a timer and ping counter) or drop out from the game (removing them from the selected game and returning to the main display).
4.1.4.3 Inputs: The selected game, user account. Outputs: Details for the selected game, the first waypoint to start the game.

4.1.4.4 Sub functions include the ability to drop out of the game (also done via the manage games function) and to start the game. Starting the game loads the user to the ping display with the first provided waypoint as their ultimate destination. Starting the game also creates specific ping counters and timers for each game. As well, the user must be shown the current leader board for each game and some basic statistics before they start the game. These statistics include but are not limited to: distance to first waypoint, overall distance of entire game, host name, etc. These statistics are to help the user decide if they wish to join the game.

4.1.4.5 Connecting to the game server is dependent upon the user’s connection speed. A slow speed will be undesirable for optimal enjoyment.

4.1.4.6 The request to join the game and successful joining should happen quickly behind the scenes, allowing the user to engage in the game near immediately upon selecting to join.

4.1.5 Manage Game

4.1.5.1 Activated anytime the user selects a game displayed on the main display. Displays different options depending on the user’s privileges for that game. If the user is a player in the game, their options are limited to continuing the game (directing them to the ping display with that game’s current waypoint loaded as the destination) or dropping out of the game. As a host, the user can edit the waypoints (allows the host to upload a picture or clue for each waypoint; may allow for adding and deletion of waypoints in the future) or close the game altogether (removing it from the game server with a polite notification to each player upon updating their main display).

4.1.5.2
4.1.5.3 Inputs: Selected picture, textual clue, request for deletion, request to drop. Outputs: Notification to players about game's deletion upon main display update, updated clues for waypoints.

4.1.5.4 The manage game function requires a multitude of sub functions. The user’s privileges must be obtained for the game first to determine whether or not the user has access to host-related functions. Once privileges are determined, then other functions may also be called. For instance, a player can decide to drop from the game. Doing so would require a request sent to the game server and then the game server must update the list of current players for that game. A player can also decide to continue playing the game, in which case the game will direct the user to the ping display with that game’s current waypoint set as the destination. As a host, a function call to edit the provided waypoints may be called. To handle this, the host will be shown a list of the current waypoints (showing their number, attached picture if present, and attached clue if present). Selecting a waypoint will allow the host to alter the picture or textual clue associated with that waypoint. As well, the host can elect to close the game prematurely for any reason. Doing so entails a request sent to the game server to delete the game (confirmation required) and the subsequent termination of the game and notification to its players.

4.1.5.5 Connection to the game server is the only real limitation.

4.1.5.6 Managing games should be as simple and intuitive as possible.
4.1.6 Ping

4.1.6.1 This is the core functionality of the game itself. The user is presented with a simple touch screen UI that when touched, sends out an audible ping. The user's current GPS position is registered and compared to the current waypoint destination of the game. A quick mathematic formula determines how long the user waits before an audible return ping is heard (informing the user via deduction of how much further the waypoint is). If the user is at the current waypoint they need to be, pinging the application will result in the game updating to include that the user has reached the waypoint and the destination is changed to the next waypoint for that game (if present). If the waypoint reached is the final waypoint for the game, the pinging will result in the game ending (for that user) and the user's statistics (ping count and elapsed time) are uploaded to the game server and displayed to the user. Any changes in the leader boards are made at this point.

4.1.6.2

4.1.6.3 Inputs: Current GPS location. Outputs: audible return ping, updated destination along with notification of reaching prior waypoint, final game statistics.

4.1.6.4 Pinging requires many functions. It will involve the transmitting of the current GPS location (after retrieving it, of course), the calculating of the return ping time based on distance to current waypoint destination, the updating of waypoints on the condition of arriving at the current destination, and transmission of the user’s final game statistics to be displayed and compared to the leader board of the game.
4.1.6.5 Connection issues may appear based on the user’s connection speed and the speed of the GPS at identifying the user’s location and distance to the destination.

4.1.6.6 Being the main portion of our game, the pinging should respond almost immediately with results. There should be no lag between touching the screen and calculating the time for the return ping. Any lag here may result in confusion with the user in terms of their relative proximity to the current waypoint destination.

4.1.7 Exit Program

4.1.7.1 Called whenever the user manually terminates the program, this function will release all temporary resources being used. Restarting the program will result in loading of the main display and a reloading of the games and current state of the user in each game (stored on the game server).

4.1.7.2 No flow diagram needed.

4.1.7.3 Inputs: desire to terminate program. Outputs: none (resources freed).

4.1.7.4 No sub functions needed other than those that will be freeing resources as necessary.

4.1.7.5 Terminating the program should not affect any current states the user has in any game (i.e. it should not reset what waypoint they are currently on).

4.1.7.6 Resources must be freed and the program should terminate cleanly.
4.2 Software Interface Description

4.2.1 External Machine Interfaces

4.2.1.1 Our application will require connection with our constructed game server. Currently, our plan is to create a custom game server using one of our local machines. Some direct communication between users' iPhones may also be required.

4.2.2 External System Interfaces

4.2.2.1 As stated above, our application will be networked with a custom game server. This server will be hosted on one of our local machines. The game server will store games and any information relevant to each game. This includes but is not limited to: users currently playing the game, host user, series of waypoints, user state inside the game, leader board statistics. The series of waypoints stored will include the order in which they are to be reached (ordered by original input), any picture that is to be associated with the waypoint (uploaded by host only), and any textual clue provided by the host. User states consist of which waypoints the user has currently reached and which waypoint that user is currently in pursuit of.

4.2.3 Human Interface

4.2.3.1 Main Display - This is the home screen of our application. It is the first screen the user will see upon activating the program and logging into Game Center. At the very top of the screen is a button to create a new game. The screen will display a list of games they are currently playing in followed by a list of games that they are the host of (or managing). As well, there will be displayed any games that they have been invited to in between these two lists with a visual cue to inform the user. Selecting a game brings the user to either the manage game screen or the game lobby screen.
4.2.3.2 Game Lobby - For each game, there is a game lobby. Selecting a game the user is currently playing brings the user to its game lobby screen. The game lobby displays the game's host name at the top (along with the game's name if present) and the current leader board for the game. At the bottom, the user is presented with three buttons: "Back", "Drop Out", and "Continue." Clicking back returns the user to the main display. Clicking the drop out button will remove the user from that game (until invited back by the host) and return the user to the main display. Clicking continue will bring the user into the ping display to play the game.

4.2.3.3 Manage Game Screen - When the user is a host for a game and selects a game they are currently hosting, they are brought to the managing screen. Here they see the game’s name at the top, the current leader board, and a list of players currently engaged in the game along with their status (not started, playing, finished). Next to the game’s name at the top, the host has an option to invite friends linked to their Game Center account. Clicking this button will bring up a display where the user can select friends to send an invite to. At the bottom, the host has three options. They can edit the waypoints, taking them to the edit waypoint screen, they can return to the main display, or they can choose to end the game. Ending the game will remove the game from the server and notify the players when they update their main display.

4.2.3.4 Edit Waypoints Screen - If the host decides to perform some minor changes to the waypoints in their game, they are sent here. Here, a listing of each waypoint involved with the game is shown along with a slot for any pictures associated with the waypoint and a slot for any textual clue. Clicking either of these slots will allow the host to modify the associated picture or clue. The host can only select pictures already stored on the iPhone (in-app camera functionality might be applied later). Two buttons are provided at the bottom: "Back" and "Apply." Selecting "Back" will return the user to the manage game screen. Selecting "Apply" at the bottom of the screen will commit the changes to the game server.
4.2.3.5 Create Game Screen - This screen is used when a user wishes to create their own Scavenger Hunt game. The display is simple; prompting the user for a quick name for the game first and then proceeding to the setting of waypoints. The screen will display the current GPS coordinates of the user and provide options for either setting a waypoint at the current location, finish creating the game (note that one waypoint must first be set to finish creation), or cancel creating the game. When a user decides to set a waypoint, the GPS coordinates are stored and the user is prompted with a screen where they can set a picture to that waypoint and a textual clue if desired. They then select to either add another waypoint (returning to the original create game screen) or finish creating the game. When the user selects to finish creating the game, the game is uploaded to the game server and the user is returned to the main display which will update to show their new hosted game. Selecting the game brings them to the manage game screen where they can invite friends.

4.2.3.6 Ping Display - The screen associated with our game. Simply, this screen will display the picture associated with the current waypoint destination (if there is no picture, the screen will default to a radar background) along with a text field at the top containing the provided textual clue (if no clue is provided, the game's name will be displayed instead). At the very bottom, the user has a simple back button they can use to return to the game lobby. Touching the screen anywhere besides that button results in the game’s pinging mechanic. An audible ping is heard and depending on the user’s relative proximity to the next destination a return ping is heard thereafter.
4.3 **Control Flow Description**

4.3.1 Ideally, a typical game creation and management would proceed as follows. The user starts the application, logging into Game Center and connecting with the game server. An update request is sent to the game server containing the user's account information (namely, their account number). The game server responds by providing a list of any games the user is either playing in, hosting, or invited to. The main display is updated to show these games, if present. The user selects to create a new game at the top of the screen and is brought to the create game screen. The user is prompted for a game name and creates one, proceeding to the setting of waypoints. The GPS is activated and the current GPS coordinates of the user are displayed. The user selects to set a waypoint at the current location and is shown a screen where they can input a previously stored picture and/or textual clue. The user opts to insert a picture clue and selects a picture of Jerry West's statue. The user then types in "Associated with gladiators in the old days" as the textual clue. The user decides that this game will only have one waypoint and selects finish instead of adding another waypoint. The game is stored and sent to the game server, containing a list of all waypoints in the game (only one) and all the clues associated with each waypoint. The user is returned to the main display and their newly created game is shown under the heading "Managed Games." The user then selects the game and is taken to the manage game screen. Here they see any players in the game (none yet) and various options. They decide to invite their friend SampleFriend to the game so they select the option at the top to invite friends. In the prompt (fueled by Game Center) they select SampleFriend from the list and select invite. A push notification is sent to SampleFriend and the user is returned to the manage game screen. The user decides to terminate the game and selects the option to delete the game. A confirmation is required and provided. The game is then removed from the game server (SampleFriend will not have the game listed on the main display upon logging in) and the user is returned to the main display where their game is now gone.
4.3.2 For a user wishing to play a game, ideally it would proceed as follows. The user receives a push notification informing them that they have been invited to play a game. The user starts the application and after logging into Game Center and connecting to our game server, the list of games they are involved with are updated. One game under the heading "Active Games" has a visual cue around it, signifying that the user has been invited to join that game. The user selects the game and is brought to the join game screen. Here they see the game’s name, its host, the current leader board, and the current distance to the first waypoint. The user decides to join the game and selects "Start Game." The first waypoint is loaded as the current destination and the user is brought to the ping display where they see a picture of Jerry West’s statue and a textural clue at the top stating "Associated with gladiators in the old days." The user deduces that the waypoint must be the Coliseum and heads there. As they get to the Coliseum streetlight, the user taps the screen to ping. The current GPS coordinates of the user are compared behind the scenes to the GPS coordinates of the current waypoint destination (stored on the game server) and the time between the return ping is calculated. Very quickly, the user hears the return ping, indicating they are very close to the waypoint destination. The user walks up to the entrance of the Coliseum and taps the screen again. This time the GPS coordinates match and the user is notified that they reached the waypoint. The user then clicks "Ok" on the prompt and the user state on the game server is updated to reflect that the user completed the first waypoint and is now on the next. The next waypoint destination is downloaded, along with any clues associated with it, and the ping display updates accordingly. This continues until the user reaches the final waypoint. Upon reaching the final waypoint and pinging, the user is notified that they have finished the game. Clicking "Ok" this time brings the user to the game lobby screen where they see their final game statistics (number of pings and elapsed time) and the game’s leader board (updated when the player finishes). Clicking "Continue" from here results in a notification that the game is complete for that user.
5 Behavioral Model and Description

5.1 Description for Software Behavior

5.1.1 Events

5.1.1.1 Starting the Program - Opening the program will be the first behavior. Doing such will quickly create a connection between the application and the game server, transmitting the user's account number and retrieving any games the user is involved with. The main display is updated to reflect this and the system enters a waiting state, listening for input by the user.

5.1.1.2 Creating a Game - Selecting to create a new game will bring the system into the creation state, where the user is to provide details of the game such as clues. Most importantly, the creation state involves the user selectively placing waypoints as they desire and as prompted by the creation state. When finished, the game (including waypoint locations) is uploaded to the game server where other players may view it from their applications and the host's state is returned to the main display.

5.1.1.3 Joining a Game - When a user selects a game they are not a host of or actively participating in already, they are put in the joining state. Here they are shown information pertaining to the selected game and can decide to either join the game or return to the main display (waiting state). Selecting to join the game will put the player in the playing state for that game.

5.1.1.4 Playing a Game - While in the playing state, the user can view the clues associated with their current waypoint destination and can ping the consoler to determine relative proximity to said destination. Pinging involves some internal states described below. At any time during the playing state, the user may opt to temporarily leave the game (being sent to the game lobby or main display, or waiting state).
5.1.1.5 Game Lobby - After joining a game and while not actively pinging/playing, the user is brought to the game lobby state anytime they select a game they are still participating in. Here they can view the current statistics and leader board as well as decide to continue playing, drop out from the game, or return to the main display. Dropping out from the game will remove the user from the current participants registered with the game, return the user to the main display, and remove the user from the playing state of that game.

5.1.1.6 Pinging - When a user pings to determine their closeness to the next waypoint, a series of internal states and actions occur. First, the origin ping is sent. The system will check the current GPS coordinates of the user against the coordinates of the current waypoint destination. If they are relatively close enough, the system will notify the user of their success and advance them to the next waypoint stage/state for that game (updating the picture and clue respectively). If that was the final waypoint for the game, then the user is informed of such and returned to the game lobby where they can view their final statistics and the updated leader board. If the user is not close enough to the current waypoint, then the system calculates the time between the origin ping and the return ping based on distance to the next waypoint. The system enters a waiting state while waiting for the timer to finish. Then a return ping is heard, and the user keeps searching for the waypoint.

5.1.1.7 Manage Game - As the host of the game, a user may decide to edit the clues associated with specific waypoints in their game, or the user may wish to end the game before all players have completed it. Doing so will result in the game being removed from the game server and all associated players in the game being removed from the playing state of that game and returned to the main display.

5.1.2 States

5.1.2.1 Waiting State - The default state whenever a user is not involved with any games. They are allowed access to the main display, where they can view games they have been invited to. This is also the state of the main display, where functionality is determined by the actions of the user (i.e. selecting a game to join, manage, or creating a new game).
5.1.2.2 Hosting State - For each game, there is a host (the creator of the game). That user is put in the hosting state for that game. In this state, they are allowed managing actions related to that game. This includes deleting the game and editing waypoint clues.

5.1.2.3 Playing State - For each game, there will be players playing the game (if invited by the host). A player that joins a game is put in the playing state of that game. In this state, their ping count and elapsed time are constantly being monitored as well as the player being provided options to actively play the game or drop from the game. Note that the playing state does not mean the player is currently pinging. The player may be on the main display (waiting for input state) but the player is still noted as currently engaged in the selected game.

5.1.2.4 Creation State - When a user decides to create a game from the main display, they are brought to the creation screen and placed in the creation state of the system. Here they set waypoints and their respective clues until they are ready to finish creating the game (or cancel). Upon finishing creation, the game is posted on the game server, the user is put into the hosting state for that game, and the user returns to the main display.

5.1.2.5 Pinging State - When a player in a game wishes to actively participate in the game (either via "Start Game" or "Continue"), the user is placed into the pinging state. In this state, the user activates pings that are sent to the game server. These pings contain the user’s current GPS coordinates and are used to determine if the user has reached the current waypoint destination for that game. If so, then the user’s state inside that game (by default, each player starts at waypoint or state one) is advanced to the next waypoint, or state, if present. If that was the final waypoint for the game, then the user is removed from the pinging state and playing state for that game and placed in the finished state as well as returned to the game lobby.
5.1.2.6 Finished State - When a player completes the game, they are brought to the finished state for that game. In this state, the user is still free to visit the game lobby and view their statistics and the current leader board. However, they can no longer drop from the game (they aren't in it anymore) or choose to continue playing it via pinging.
5.2 State Transition Diagrams

5.2.1 Overall

5.2.2 Playing/Pinging
*Note that "Next Waypoint Stage" would be identical to the functions performed at the "Current Waypoint Stage." *
5.3 Control Specification

[Flowchart diagram showing decision points and paths for different user interactions related to game creation and management, including Main Display, Create a Game, Display Games, Game Creation, Game Lobby, Manage Game, Pinging, At Correct Location, Is This the Final Location, and Advance to Next Waypoint Stage.]
6 Restrictions, Limitations, and Constraints

The application will be restricted to only iPhone and iPad 3G users for two reasons. An iPhone/iPad application is not easily transferable to other smartphones. This is because Apple uses its own proprietary language called Xcode which it does not allow other companies to support. If it wasn’t for Apple’s domination of the social media aspect of smartphones we may have chosen a less restrictive platform. The second reason the application will be restricted to iPhone or iPad 3G users is that the application relies heavily on GPS functionality. iPod touch and iPad Wi-Fi users will not be able to use the product if they purchase it so we must either make any potential owners aware that they can only use this software if they have GPS functionality or make it so that only iPhone and iPad 3G users can purchase it.

We also have a few limitations to our testing process that at this time we are unsure of whether or not they can be overcome. One of those limitations is the limited number of iPhones we have to test with. We currently have two available and are looking to have more made available in the future. Our other limitation is based on location. Since the system needs to function differently in different areas (example: Games in New York shouldn’t show up for me in Morgantown) it would be helpful to be able to travel for testing. Unfortunately, budget and time issues may make this impossible.
7 Validation Criteria

The approach to software validation is described.

7.1 Classes of tests

7.1.1 Create Scavenger Hunt

A user will choose “create a hunt”, fill in all game information, and set at least one point for the hunt.

7.1.2 View Hunt

A user will select a hunt from the lobby.

7.1.3 Play Hunt

Once a user is viewing a hunt they will press “start the hunt” and begin. They will ping the system until they have found all the saved locations. Once they’ve found the last location they will see their score and choose whether or not they would like to post it as their score for the game.

7.1.4 Ping Location

While attempting a hunt the user will always have the option to ping his current location.

7.2 Expected software response

7.2.1 Create Scavenger Hunt

Any user, including the one who created the hunt, should be able to view the hunt’s information and choose to play it. In addition all information should match the creators input.

7.2.2 View Hunt

The user should be able to see basic hunt information such as creator’s name, number of points, and clues. The user should also be able to see a table with high scores for that hunt. Finally every user should be given the option to return to the lobby or start the chosen hunt.
7.2.3 Play Hunt

Completion only has an effect when the user decides to post there score. When that occurs the player’s score should be listed on the high score table when the game is selected.

7.2.4 Ping Location

If they are close enough to the next location they will get a confirmation that they found it and move on to the next location and clue. If they are not close enough a sonar ping will sound.

7.3 Performance bounds

7.3.1 Create Scavenger Hunt

A game created in Chicago should not be viewable and therefore playable by someone in Morgantown.

7.3.2 View Hunt

7.3.3 Play Hunt

A player’s score will not show up if ten better scores have already been posted.

7.3.4 Ping Location

The length of the sonar ping will be proportional to the distance the user is away from the next location.
8 Appendices

Presents information that supplements the Requirements Specification

8.1 System traceability matrix

A matrix that traces stated software requirements back to the system specification.

8.2 Product Strategies

This product is primarily a form of social media so word of mouth advertising should work after we have entered the market. To get the word of mouth started we will provide the application for free until a sufficient number of users has downloaded it. Once that occurs we will put a price on it and new users who wish to play with their friends will be our primary customer.