

1.0 - Title

**COMP 3160 - Mobile App  
Development 2  
Project - Manhunt  
T0066654, Gavin Edwards**

## **2.0 - Table Of Contents**

<b>1.0 - Title</b>	<b>1</b>
<b><u>2.0 - Table Of Contents</u></b>	<b><u>2</u></b>
<b><u>3.0 - Introduction</u></b>	<b><u>3</u></b>
<u>3.1 - Transhuman Games</u>	<u>3</u>
<u>3.2 - Hunter</u>	<u>3</u>
3.3 - Manhunt	3
3.4 - Difference	4
<b>4.0 - System Documentation</b>	<b>5</b>
4.1 - Cost & Profit	5
4.2 - Design & Structure	5
4.3 - Known Bugs & Technical Debt	6
4.4 - Future Development Goals	7
<b>5.0 - User Documentation</b>	<b>8</b>
5.1 - Installation	8
5.2 - Pre Game Requirement	8
5.3 - How To Play	9

## **3.0 - Introduction**

### **3.1 - Transhuman Games**

Hello,

My name is Gavin Edwards and I am the founder of Transhuman Games.

Transhuman games hopes to combine the games of our childhood with the technology of the present.

This is our second app that does this our first being the classic hidden role game Handshake Murder. We converted the game from the physical handshakes to a more covid friendly digital handshake game, along the way we solved a few of the biggest issues with the original. Someone had to not play so that a murderer could be selected, someone had to manage the game time and each person was responsible for tracking there own death timer. In our version we used internal timers to manage the clock and the vibration function to notify users of their demise. We also used a random number generator so that everyone got to participate.

### **3.2 - Hunter**

Hunter is a classic game where players group up and select a few people out of the group to be hunters and the rest are hunted. The hunters will usually give the hunted a head start at which point they will run and hide. Once the head start ends the hunter would wander around and attempt to track down the hunted. If they find any they chase them down and if they manage to tag them they become hunters as well. This goes on until all players are hunters or until the designated time ends.

### **3.3 - Manhunt**

Manhunt is our version of Hunter with a few notable changes to allow for a greater variety of play. The way ours works is similar:

1. All players join a lobby.
2. The game starts and a set percentage of players are assigned the role of hunter.
  - a. They will then be told to wait well, a 2 minute timer counts down.
  - b. Then they will spend the next 15 minutes tracking people down and attempting to tag them.
  - c. Every minute the position of each hunted player will update in their display.
  - d. If they do tag someone they will select the tag button and input the tag code provided by the tagged player.

3. Everyone else will have the role of hunted.
  - a. They will have a 2 minute timer to run and hide.
  - b. They will spend the next 15 minutes hiding and running from the hunters.
  - c. The position of every hunter will update on their map as the hunters move.
  - d. If they are tagged they will need to select their tag button and provide the code to the hunter.
4. Once tagged a hunted will become a hunter and proceed to follow steps 2.b forward for the remaining time.
5. Eventually either enough players will have been tagged or the timer will run out and all players will be moved to a Victory screen that will allow them to either rematch or return to the main menu.

### 3.4 - Difference

The core benefit of the app is that it allows people to play a more intricate game of cat & mouse. You could purposely use your last known position to pull the hunter one way all well maneuvering around them and avoiding being seen. Our version also prevents players from getting bored as one of the fatal flaws with Hunter is that once hidden sometimes you never have to move and end up waiting out the timer in a single spot. Combine this with the fact it naturally reduces the strength & safety of hiding high up or inside objects it results in a more engaging experience for all involved. The true core benefit though is it will allow people to utilize the inside of buildings thus making the game significantly more playable during bad weather as well as introducing whole new avenues of play.

## 4.0 - System Documentation

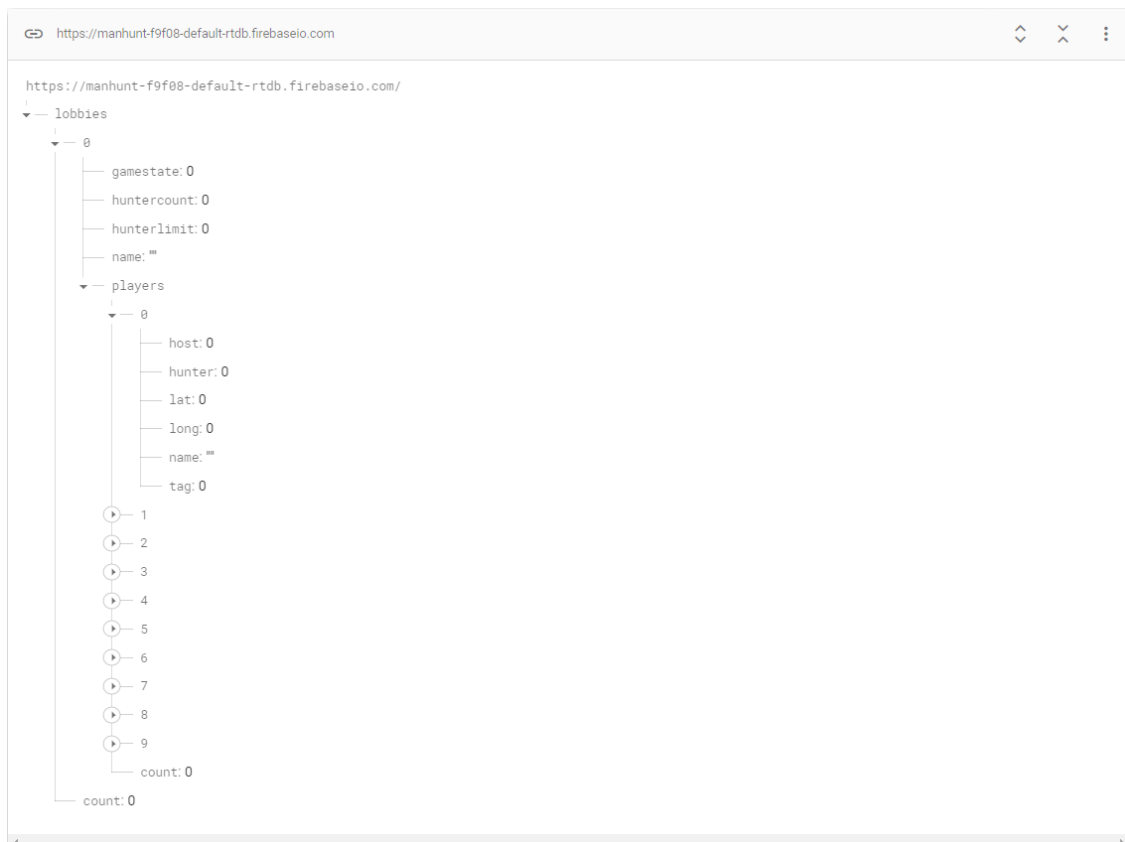
### 4.1 - Cost & Profit

Luckily the only cost for upkeep of this application is the Firebase Realtime Database which scales as 200k connections per database, \$5/GB stored, \$1/GB downloaded. Currently having tested approximately 100 lobbies I have used 2 connections, 773b storage, and 1.22mb downloaded. Assuming 10 players per lobby we should be able to host 20,000 lobbies, using 0.01546GB stored and 24.4 GB downloaded. Should result in a cost of under \$30 per full database. Based on google ad sense calculator 20,000 users would return between just over \$1000 and just over \$2000 based on user content category.

### 4.2 - Design & Structure

The application runs off of a Firebase Realtime Database, a countdown timer and the google maps API.

#### Firestore Realtime Database



The Realtime Database structure is built like above.

Lobbies hold each lobby as well as a count of existing lobbies.

Each lobby holds many keys and values:

Gamestate tells the application where the lobby is currently in its lifespan.

0 is the lobby.

1 is how to switch to the role page.

2 is how we navigate from role to play.

3 is the timer has run out.

4 is hunter count = hunterlimit.

Huntercount is calculated by the app by finding out what the ceiling of 10% of player count is at game start and is then used to know how many players should start as hunters. It is also used to track the amount of hunters in the game.

Hunterlimit is used to refer against the huntercount and tell if the hunters have hit the ceiling of 90% of player count and thus win.

Name is used to hold the lobby name which is then populated on both host and join lobby screens.

Players are used to host the player objects and the playercount.

Playercount is used to calculate many variables.

Player is a host of 6 values:

Host is the only one who controls gamestate timers.

Hunter determines your hunted or hunter status.

Lat is used to store a player's latitude.

Long is used to store a player's longitude.

Name stores the player's imputed name.

Tag is a randomly generated number for when tagged.

## Countdown Timer

```
new CountdownTimer(counttime, 1000)
{
    public void onTick(long millisUntilFinished)
    {
        min = time / 60;
        second = time - (min * 60);
        timer.setText(min + ":" + second);
        time--;
        getDeviceLocation();
    }
}
```

```

public void onFinish()
{
    if (host == 1) {
        gameState.setValue(3);
    }
}
}.start();

```

The CountdownTimer is used to trigger events every second. Currently it is in charge of the display timer and the map position updates.

## Google Maps API

```

@SuppressLint("MissingPermission")
@Override
public void onMapReady(@NonNull GoogleMap Map)
{
    this.map = Map;
    map.setMyLocationEnabled(true);
    map.getUiSettings().setAllGesturesEnabled(false);
    map.getUiSettings().setZoomGesturesEnabled(false);
    map.getUiSettings().setScrollGesturesEnabledDuringRotateOrZoom(false);
    map.getUiSettings().setScrollGesturesEnabled(false);
    getDeviceLocation();
}

private void getDeviceLocation()
{
    @SuppressLint("MissingPermission") Task<Location> locationResult =
    fusedLocationProviderClient.getLastLocation();
    locationResult.addOnCompleteListener(this, new OnCompleteListener<Location>() {
        @Override
        public void onComplete(@NonNull Task<Location> task) {
            if (task.isSuccessful())
            {
                // Set the map's camera position to the current location of the device.
                lastKnownLocation = task.getResult();
                if (lastKnownLocation != null) {
                    lat = lastKnownLocation.getLatitude();
                    lng = lastKnownLocation.getLongitude();
                }
            }
        }
    });
}

```

```

        map.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(lat, lng),
DEFAULT_ZOOM));
        playerIndex.child("lat").setValue(lat);
        playerIndex.child("long").setValue(lng);
    }
} else
{
    map.moveCamera(CameraUpdateFactory
        .newLatLngZoom(defaultLocation, DEFAULT_ZOOM));
    map.getUiSettings().setMyLocationButtonEnabled(false);
}
}
});
}

```

These two functions are my main uses of the google maps API.  
 OnMapReady creates the map and sets its settings then calls the other.  
 GetDeciveLocation collects the user's location and updates it to the Database.

## 4.3 - Known Bugs & Technical Debt

Whole application uses hard coded reference to lobbies/0 when it should be dynamic.  
 Whole application should use System.currentTimeMillis() to create the timer each time  
 rather than passing a timer sharedreference. Currently the timers break if a players  
 phone locks.

Host.java's host button should create a lobby rather than modify the existing one.  
 Host.java's host button should query all lobby names to confirm that the one it's trying to  
 create doesn't already exist.

Join.java's join button should query for a lobby based on the name typed rather then  
 just defaulting to lobbies/0.

HostLobby.java should have a recyclerview with a list of player names.

JoinLobby.java should have a recyclerview with a list of player names.



HunterPlay.java & HuntedPlay.java should have maps that update with database information.

Results.java's rematch button should reset the lobby and then update but should take into account if a non-host joins before the host. Potentially just saying the host has not restarted yet.

Should also reference saved preferences for name & lobby references to rejoin.

Results.java's main button should delete the lobby allowing a new one to be created in its place.

## 4.4 - Future Development Goals

We would like to add many things:

1. Settings
  - a. Timer Lengths
  - b. Hunter Count
  - c. Hunter Limits
  - d. Geofence Distance
2. Altitude
3. Banner Ads
4. Sound Effects
5. Geo Fencing
6. In Game Selfies
7. Clickable Player Indicators

## 5.0 - User Documentation

### 5.1 - Installation

As the app isn't currently on the App Store so we will need to follow some slightly different then normal installation instructions.

1. Open "Settings in your phone.
2. Click "Apps"
3. Click "Special App Access"
4. Click "Install Unknown Apps"
5. Click on your Browser of choice
6. Switch the "Allow from this source" radial button
7. Open "Chrome"
8. Paste  
[https://drive.google.com/file/d/1T8muhw7A6uJvG\\_dPfaW-PRBPZZIKoll3/view?usp=sharing](https://drive.google.com/file/d/1T8muhw7A6uJvG_dPfaW-PRBPZZIKoll3/view?usp=sharing)
9. Download and install this file.
10. When you open the app, allow "Location" permissions.

### 5.2 - Pre Game Requirement

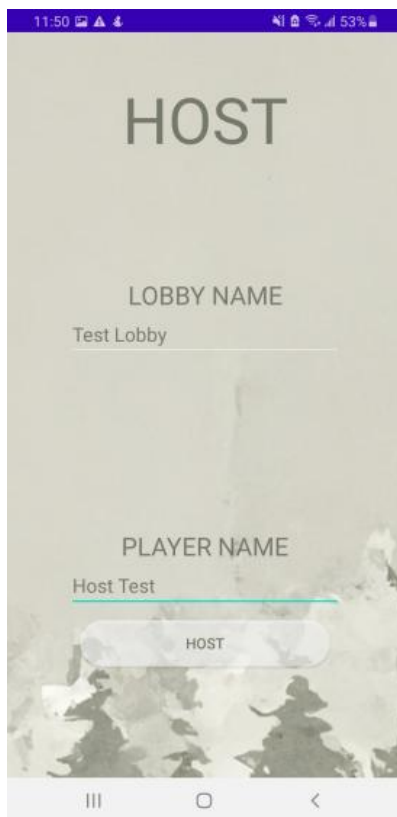
As this game is a combination of physical and digital play there are a few things to consider.

1. Make sure everyone is wearing appropriate footwear and clothes for the weather.
2. Make sure to bring water to drink as running around will result in a strong thirst.
3. Set up the area of play that your group is comfortable using.
4. Make sure any special rules have been established and are clear.  
Ex.
  - a. No Climbing
  - b. No Running inside buildings

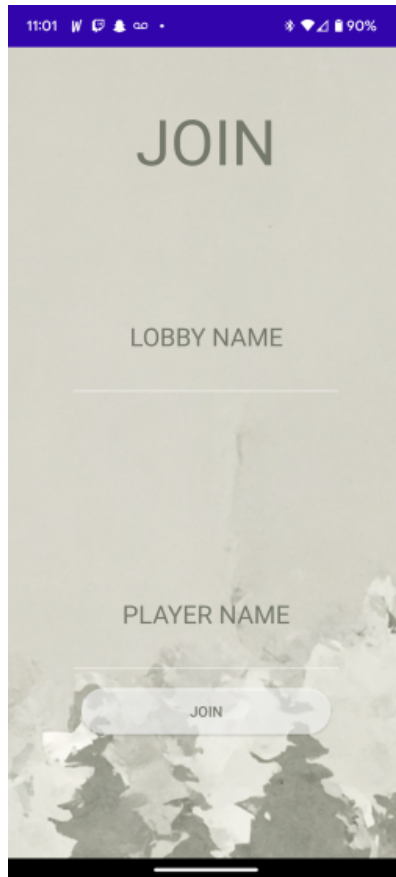
## 5.3 - How To Play



When the app opens you are presented with two options. Host means you want to be in control of the lobby. Currently this just means you choose when the game starts, in the future this role will have control over all game session settings. Join means you want to join an existing lobby. This is the easier option for new players.



The Host Page presents you with two options. One to input a lobby name and one to input your player name. Once they are filled in press the Host button to create the lobby.



The Join page has two options.

Lobby Name is for the name of the existing lobby you are trying to join.

Player Name is for your player name.

Once both are filled in click the Join button and the App will search for a lobby with the name you have written.



Hosting the lobby and joining the lobby produces a very similar page each holding a space for a player list.

The core difference is that if you are hosting the lobby then you will have a start button.

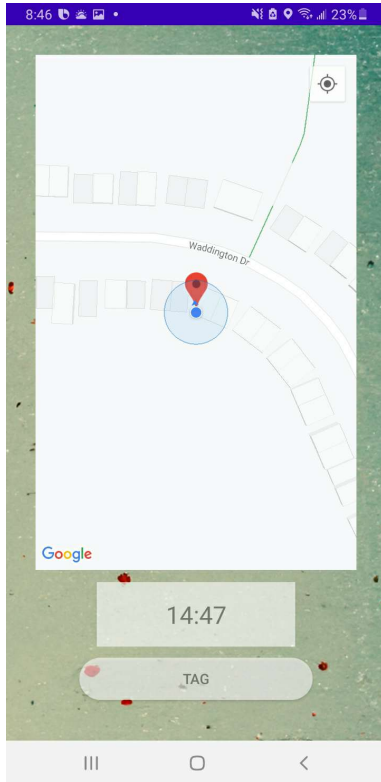
Press this when all players are in the lobby.



The Application will assign most players with the role of Hunted and show you the following page with the instructions to “Run And Hide” as well as a timer. This is to give you a headstart to get away from the hunter(s) as they will have to wait where they are well you attempt to find a space to start the game in. Currently the head start is 2 minutes.



Approximately 10% of players will be assigned the role of Hunter and will be informed they should “Stay Where You Are”. This is to allow the Hunted a chance to get further away and find some space to hide.

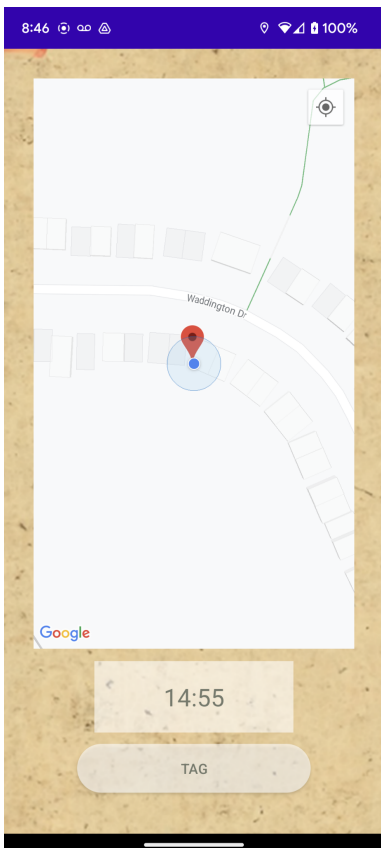


After the timer runs out the Hunted players will be provided this view featuring a timer and tag button.

It also will feature a map view containing up to date information on the position of all hunters compared to yourself.

Once the timer hits 0 the Hunted will have won so long as there are more than 10% of players left unhunted.

The tag button will open a view that is used if you are tagged.



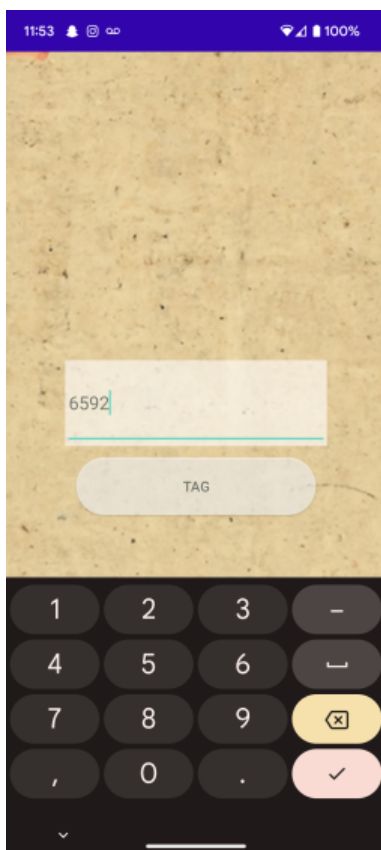
If you were assigned the role of hunter or have been tagged successful you will be shown a similar view with a timer and tag button.

The map this time would feature your position and every minute it will briefly show the position of all hunted targets. If the timer gets to 0 and the hunters are not 90% of the total players you will have lost.

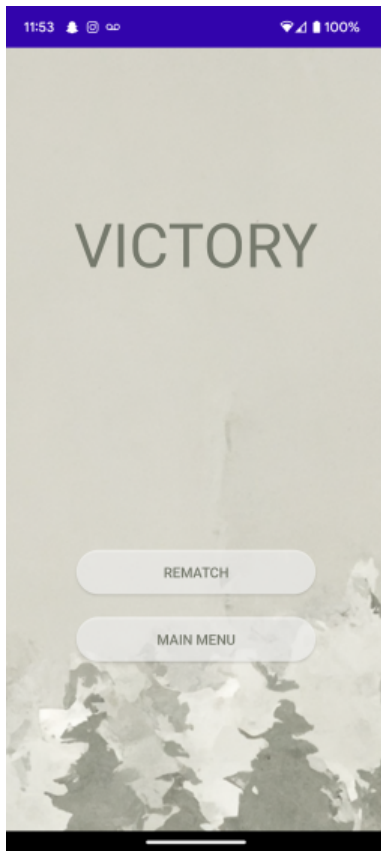
The tag button is for if you track and tag a player.



If you are tagged by another player and click the tag button on the Hunted page it will present you with this page. It features your personal TAG number which you will need to provide to the person who tagged you. Once they have imputed it successfully you will switch to the role of hunter as well.



When a hunter tags someone they will click the tag button and input the TAG code supplied by the other player. Then click the Tag button. The system will confirm the code and shift the other player to the hunter role.



Once either 90% of players have been switched to the role of hunter or the timer runs out you will be moved to the results screen.

For Hunters to win they need to tag 90% of the players before the timer runs out.

For the Hunted to win they will need over 10% of the players to remain hidden til the timer runs out.

Once presented with the result screen press either button to navigate off.

Rematch will put you back in the same lobby but you will need to wait for the host to select it first.

Main Menu will bring you to the starting page where you will be able to select whether or not you want to host or join.