Base Run Requirements Document

By Reno Sarge, Michael Shliselberg, Patrick Lowry, Daniel Bergman, and Jordan Kaplan

1. Introduction

We are creating the Android application Base Run as a project for the Fall 2016 Course: Software Intensive Engineering ECE373 taught by Professor David Irwin. Base Run is a team-based smartphone GPS game where teams compete to capture and defend bases in order to strategically out-score one another in a set location and time limit. The user will choose the following parameters: location, time limit, radius of where bases can be, and an amount of bases (up to a specific maximum bound). The purpose of Base Run is to use the smartphone GPS as a way to create a new, fun, and easy game for users to quickly learn and play.

2. User Requirements

2.1 Software Interfaces

These are external systems/libraries that you have to interact with:

- Android OS (API level 16 or above) for client
- Unix Operating System on Server
- Android API (API level 16 with associated Android libraries)
- Google Maps API for displaying client location and base locations

2.2 User Interfaces

User inputs on a smartphone screen via buttons will manipulate the client side of the product, allowing the creation of games, capturing of information for these games, and joining created games. All of this information will be sent to the server and pertinent game data will be sent from clients to the server, and then distributed after verification of the data.

2.3 User Characteristics

The intended user base is people between the ages of 16-25 who have an Android OS cell phone, who have internet access, and enjoy outdoor activities. Essentially any user regardless of age who has an Android OS cell phone can play use our app.

2.4 Assumptions and dependencies

The user will have the following:

- Internet access
- Android OS cell phone that has access to the Google Maps API

3. System Requirements

3.1 Functional Requirements

- A user should be able to view open sessions

- A user should be able to join open sessions
- The server should generate an ID code for every user in a game session
- The server should generate a base layout for any specific game session
- The server should randomly divide the group of people into 2 teams
- The server should generate an updated session with a point count that the user can view
- The server should generate a list of users currently in a game
- The user, within a game, should be able to view their position as well as the bases, and a color to depict whose team has control of it

3.2 Non-Functional Requirements

Property	Measure
Speed	Processed transactions/second:360 processed client/server transactions/second User/event response time: 90 ms Screen refresh time 100 ms
Size	Units are in Megabytes Client: 20 Mbytes Server: 13 Mbytes
Ease of use	Training time: 5 minutes Number of help frames: 3 frames
Reliability	Mean time to failure: 3 seconds to failure Probability of unavailability: Low, uses mobile Internet on cellular phones Rate of failure occurrence: 1 failure every 4000 transactions, 2.78 seconds Availability: High, any location with mobile internet
Robustness	Time to restart after failure: 200 ms Percentage of events causing failure: .1% Probability of data corruption on failure: .001%
Portability	Percentage of target dependent statements: 50% Number of target systems: 2, Android, Unix

3.2.1 Software Quality Attributes

Fault Tolerance:

- Should be able to handle any disconnections due to inconsistent data from mobile data
- Measured by dropped data when it is expected or partial data streams

Reliability:

- Should be able to recover from any corrupted data by calling for an update from the server/client
- Measured by the amount of corrupted calls to replace server/client data