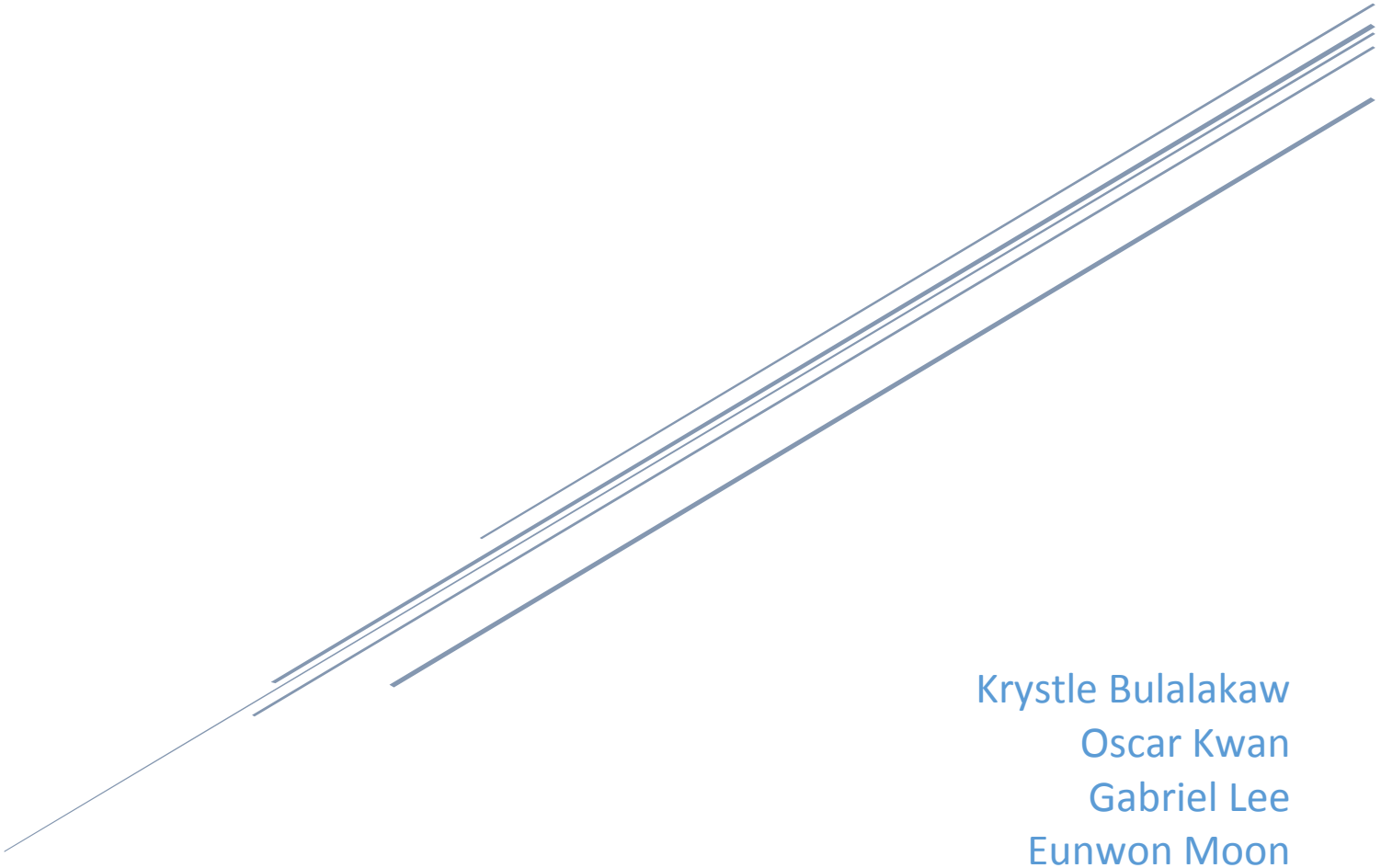
TEST DOCUMENT

COMP4985 Android GPS Project



Krystle Bulalakaw
Oscar Kwan
Gabriel Lee
Eunwon Moon

Table of Contents

Introduction

This document explains the testing strategy for the Android GPS project. It covers all the main elements (coverage, strategy, methods), as well as the list of test cases that the project will be tested against.

The focus of the testing will be the core functionalities of the project via manual testing. The testing will be performed by each individual members responsible for their own sections of the project.

Test Plan

This section covers the details of the testing that will be performed.

Test Coverage

The testing will cover all aspects of the application: server, Android app, and the webpages. The focus, however, will the core functionality of the project such as the server receiving datagrams and the web page displaying the location with markers.

Test Strategy

All the testing will be performed manually. Due to the straightforwardness of the project, unit testing and automated testing will only be "more trouble than its worth".

Test Schedule

The testing is to be performed when all of the development has been completed.

Test Cases

Test ID	*Test Description*	*Prereqs*	*Test Data*	*Test Procedure*	*Expected Result*	*Pass/Fail*
				Server		
S1	The server starts properly.	N/A	Java udps_sql 51234	1. Run the runnable JAR file with a port number.	The server will start normally and listen for datagrams.	**Pass**

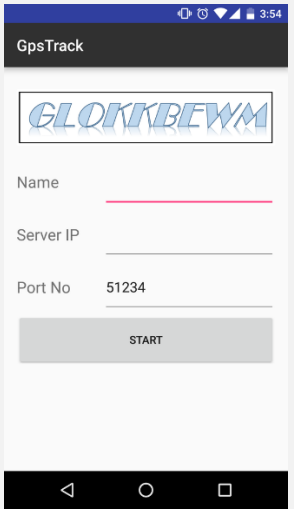
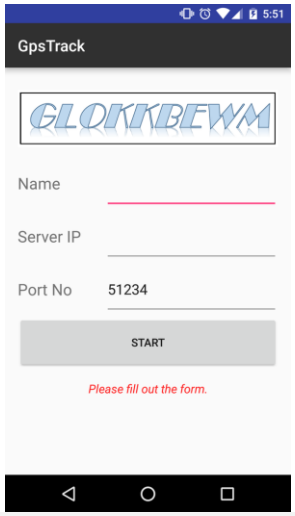
S2	The server returns displays usage if port is not defined.	N/A	java udps_sql	1.	Run the runnable JAR file without a port number.	The server will exit immediately after displaying the usage.
						`ubuntu@ip-172-31-26-0:~/java$ java -jar udps_sql.jar`
						`Usage Error : java udps <port>`
						`ubuntu@ip-172-31-26-0:~/java$`
							Pass

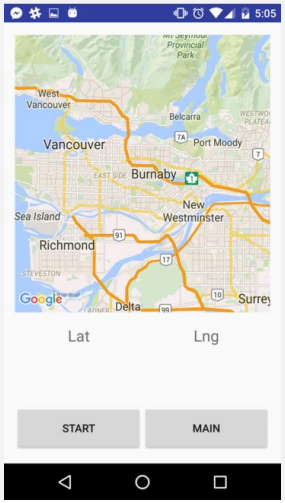
S3	The server is able to receive data from the clients.	The server is running.	N/A	1.	An android app client starts sending datagrams.	The server displays data received from the client.
						`ubuntu@ip-172-31-26-0:~/java$ java -jar udps_sql.jar 51234`
						`Connecting to database...`
						`Listening on port: 51234`
						`Datagram from: /142.232.141.162:41216`
						`Message: INSERT INTO `markers`(`name`, `lat`, `lng`, `ip`) VALUES ('moon',49.24989628,-123.00190143,'142.232.141.162');`
						`Listening on port: 51234`
						`Datagram from: /142.232.141.162:41216`
						`Message: INSERT INTO `markers`(`name`, `lat`, `lng`, `ip`) VALUES ('moon',49.24987450,-123.00163280,'142.232.141.162');`
						`Listening on port: 51234`
							Pass

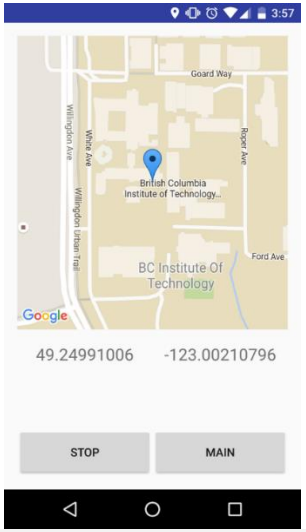
S4	The server parses the client's data correctly.	The server is running.	N/A	1.	An android app client starts sending datagrams.	The server parses the client's data; timestamp, longitude, latitude, name.	**Pass**
S5	The server updates the database with the client data.	The server is running.	N/A	1.	An android app client starts sending datagrams.	The MySQL database is updated with the data from the client. id | name | lat | lng | ip 2016-03-21 09:33:03 | krystle | 49.24991000 | -123.00211000 | 142.232.131.217 2016-03-21 09:39:43 | Eunwon | 49.24986607 | -123.00197086 | 142.232.141.162 2016-03-21 09:40:13 | Eunwon | 49.24986607 | -123.00197086 | 142.232.141.162 2016-03-21 09:40:43 | Eunwon | 49.24986607 | -123.00197086 | 142.232.141.162	**Pass**

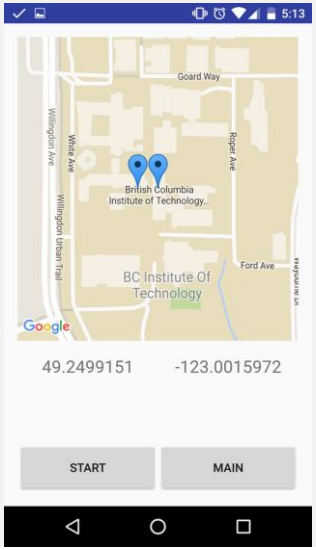
S6	The server timeout if no datagram has been received.	The server is running.	N/A	1.	Wait 100 seconds.	The server's datagram socket timeout and exit out of the program.
						`ubuntu@ip-172-31-26-0:~/java$ java -jar udps_sql.jar 51234`
						`Connecting to database...`
						`Listening on port: 51234`
						`Socket timed out!`
							Pass

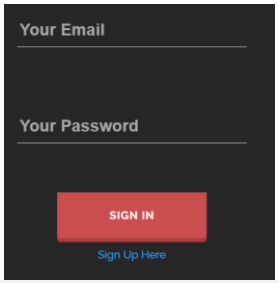
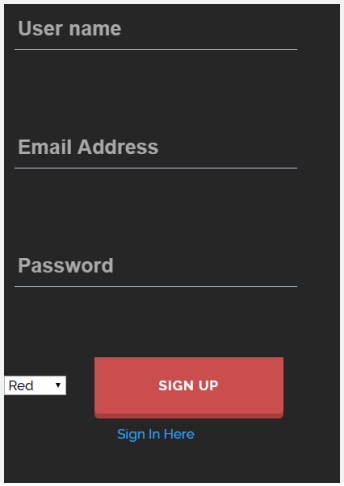
Android App

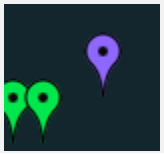
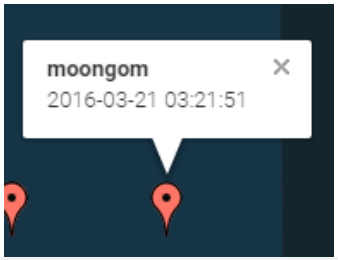
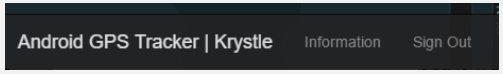
A1	The app loads properly.	N/A	N/A	1. Start the app.	The app loads properly to the connection configuration screen. 	**Pass**
A2	The app prevents users from using the app without the Name field.	The app is running and at the connection configuration screen.	N/A	1. Empty the *Name* field. 2. Select *Start*.	The app informs the user to fill out the form. 	**Pass**
A3	The app prevents	The app is running and	N/A	1. Empty the *Server IP* field.	The app informs the user to fill out the form.	**Pass**

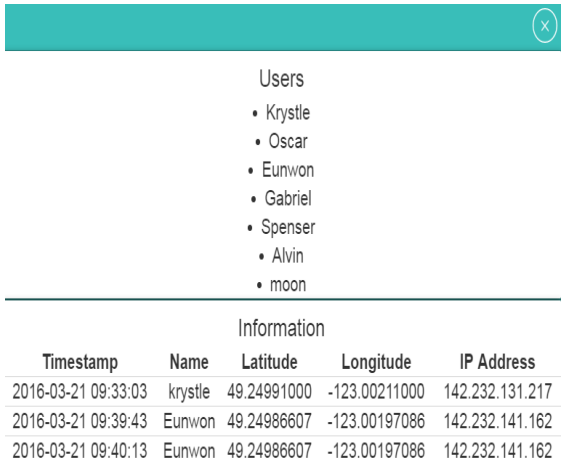
	users from using the app without the Server IP field.	at the connection configuration screen.		2. Select *Start*.			
A4	The app prevents users from using the app without the Port No field.	The app is running and at the connection configuration screen.	N/A	1. Empty the *Port No* field. 2. Select *Start*.		The app informs the user to fill out the form.	**Pass**
A5	The app starts properly with all connection forms filled.	The app is running and at the connection configuration screen.	Name: testuser Server IP: 52.37.226.120 Port No: 51234	1. Input “testuser” to *Name*. 2. Input “52.37.226.120” to *Server IP*. 3. Input “51234” to *Port No*. 4. Select *Start*.		The app brings the user to the Google Maps page. 	**Pass**
A6	The app is able to send datagrams to the server.	The app is running and at the Map page.	N/A	1. Select *Start*.		The app begins to send datagrams to the server.	**Pass**



A7	The app is able to stop sending datagrams to the server.	The app is running and at the Map page sending datagrams.	N/A	1.	Select *Stop*.	The app stops sending datagrams to the server. 	**Pass**
A8	Selecting Main brings the user back	The app is running and	N/A	1.	Select *Main*.	The app brings the user back to the connection configuration screen for new connection.	**Pass**

	to the connection configuration screen.	at the Map page.				
				Web page		
W1	The webpage brings the user to the login page.	N/A	N/A	1. Open the web page.	The web page requires the user to login before usage. 	**Pass**
W2	The webpage allows new users to register an account.	N/A	Name: Test Email: test@test.test Password: test	1. Input "Test" to *Name* field. 2. Input "test@test.test" to *Email* field. 3. Input "test" to *Password* field. 4. Select *OK*.	The webpage allows the user to register for an account. 	**Pass**
W3	The webpage allows	The user already has	Name: Test	1. Input "Test" to *Name* field.	The user is able to login and is redirected to the map page.	**Pass**

	existing user to login.	an existing account.	Email: test@test.test Password: test	2. Input "test@test.test" to *Email* field. 3. Input "test" to *Password* field. 4. Select *OK*.		
W4	The webpage displays different users in different color markers.	The webpage is at the Map page.	N/A	1. Start the server. 2. Start the app. 3. Start the app on another device.	The two devices are displayed in different colors. 	**Pass**
W5	When click on marker, name and timestamp displays.	Must have markers on webpage	N/A	1. Open webpage 2. Login 3. Click on marker 4. Observe popup window	Name and timestamp is displayed 	**Pass**
W6	User displayed on bottom of webpage	Must have user logged in	N/A	1. Open webpage 2. Login 3. Observer bottom bar	Info displayed correctly. 	**Pass**
W7	Clients tracking info displayed	Must have logged in. Must have markers.	N/A	1. Open webpage 2. Login 3. Clik info on bottom bar 4. Observe modal window	Modal window displays all registered users, and info of all markers.	**Pass**



Timestamp	Name	Latitude	Longitude	IP Address
2016-03-21 09:33:03	krystle	49.24991000	-123.00211000	142.232.131.217
2016-03-21 09:39:43	Eunwon	49.24986607	-123.00197086	142.232.141.162
2016-03-21 09:40:13	Eunwon	49.24986607	-123.00197086	142.232.141.162