Test Plan for DrunkDrivingApp

James Holliday

Introduction

A test plan for my DrunkDrivingApp which is being developed for a module in University. The project has followed a methodology of Agile and features a login system, SQLite database and uses the Google API.

|  |  |
| --- | --- |
| In Scope | Out of Scope |
| * Account Registration | * Account Management (Changing Password) |
| * Account Login | * Retrieval of Taxi Numbers |
| * Add drivers to database | * Dialing Taxi Numbers |
| * Removing Drivers from Database | * Adding Pictures |
| * Logout | * Viewing Pictures |

Objective

Make sure that the app under test meets the needs of the functional and non functional requirements whilst also meets the quality specifications defined in the project plan. Bugs and Issues should be fixed before final delivery.

|  |  |
| --- | --- |
| Roles and Responsibilities | Test Deliverables |
| * Software Developer (Myself) | * Test Plan |
| * Software Tester (Myself) | * Test Case |
|  | * Bug Report |

Approach

The approach for the testing will be to use Unit tests to check the functionality of the databasehelper class. This will ensure the database is being created, data can be retrieved and that data can be removed. Automated UI tests will also be used to check the User Interface of logging in, adding drivers and removing drivers.

Assumption + Risks

The assumption is that the tests will be able to register an account and login to access the application. The application will also be able to add and remove drivers via unit tests and via the automated AI tests. Due to the nature of the app the risk is that the database is shared between the final release and testing so make sure that the device is wiped prior to testing to ensure the database is “first time setup”.

Resources

* Android Studio
* JUnit4 (Whitebox Unit Testing)
* Android Espresso Automated UI Testing (Blackbox Testing)