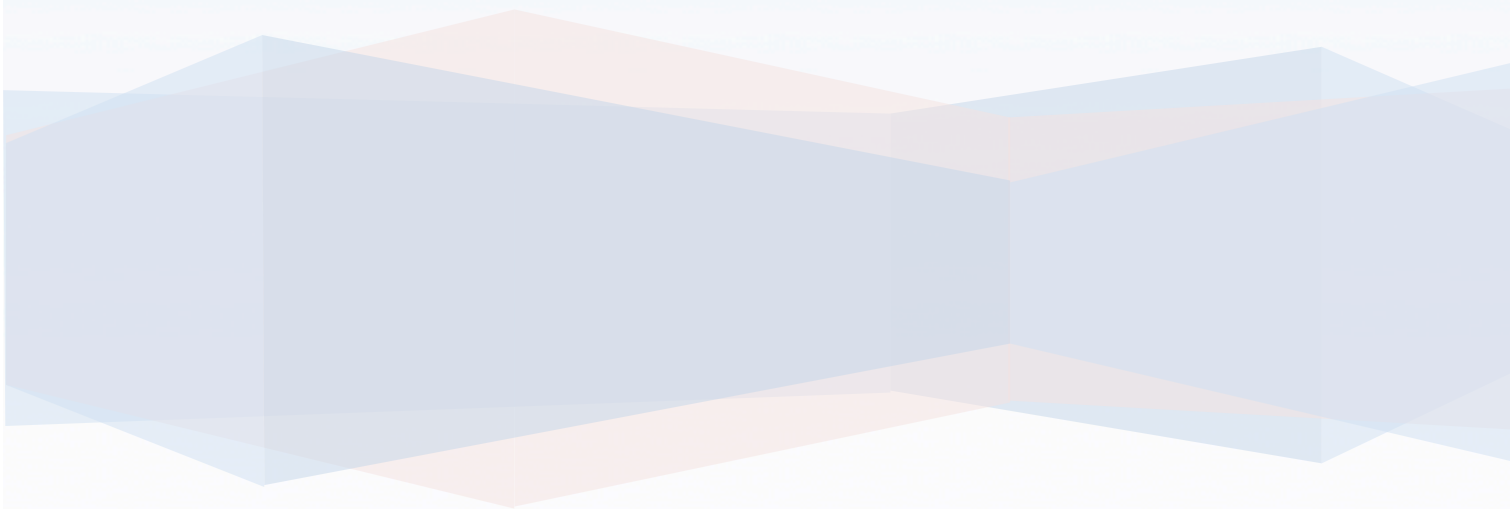


# Functional Specification

Animal Tracker And Health Monitoring Application

Gearoid Lacey



## Table of Contents

<b>Abstract</b> .....	<b>3</b>
<b>Application Purpose</b> .....	<b>4</b>
<b>Who Is The Application Intended For?</b> .....	<b>4</b>
<b>FURPS+</b> .....	<b>5</b>
Functionality .....	5
Usability.....	5
Reliability.....	5
Performance .....	5
Supportability.....	6
+ .....	6
<i>Security Requirements</i> .....	6
<b>Use Case Diagrams</b> .....	<b>7</b>
<b>Brief Use Cases</b> .....	<b>8</b>
Register.....	8
Login .....	8
Logout.....	8
Change Password .....	8
Display Animal Path .....	<b>Error! Bookmark not defined.</b>
Display Current Location .....	<b>Error! Bookmark not defined.</b>
Isolate Animal .....	9
Add Animal Profile.....	9
Process Login .....	9
Store Animal Data .....	10
Retrieve Animal Data.....	10
Store User Credentials.....	10
Set GPS Intervals.....	10
Create Boundary.....	10
Retrieve GPS Data.....	11
Store GPS Configuration .....	11
Record GPS Coordinates.....	11
<b>Iteration Plan</b> .....	<b>12</b>
Iteration One .....	12
Iteration Two .....	12
Iteration Three .....	12
<b>Bibliography</b> .....	<b>13</b>

## **Abstract**

The purpose of this document is to provide a functional specification for an animal tracker and health-monitoring app. The aim of this document is to include brief use cases, which describe the basic interactions between the application user, the application and the cloud. This document will also outline some of the desired non-functional requirements for the application. This document will also provide an Iteration plan for the project detailing the expected deliverables for each of the three iterations.

## **Application Purpose**

The purpose of this application (app) is to provide an animal tracking and health monitoring application along with a GPS tracker to perform a proof of concept regarding animal health. This will entail tracking an animals location at set intervals which are then sent to a database that is interfaced using the android application. The main objective behind this project is to try prove the possibility of determining the source of an animals sickness and or injury based on their previous locations.

## **Who Is The Application Intended For?**

It is envisaged that this application could be used on any animal theoretically with the main categories of use being in Agriculture and Equine. As agriculture has probably only started to adapt to the addition of technology in recent years it is expected that the application interface should be very user friendly. One area where this application and tracker could become quite useful is in tracking the source of bovine tuberculosis. Bovine Tuberculosis (TB) within a heard can cause havoc for a farmer. When an animal or animals are deemed to be carrying TB the farmer is then put in 'Lock Down' which restricts them from selling any animals. This can prove quite difficult for beef farmers especially as the Lock Down period lasts for 6 months meaning they have no source of revenue outside of their single farm payment. Therefore, as a proof of concept project, the aim is to aid the livestock owners in determining the cause of the animals illness by analyzing their movements.

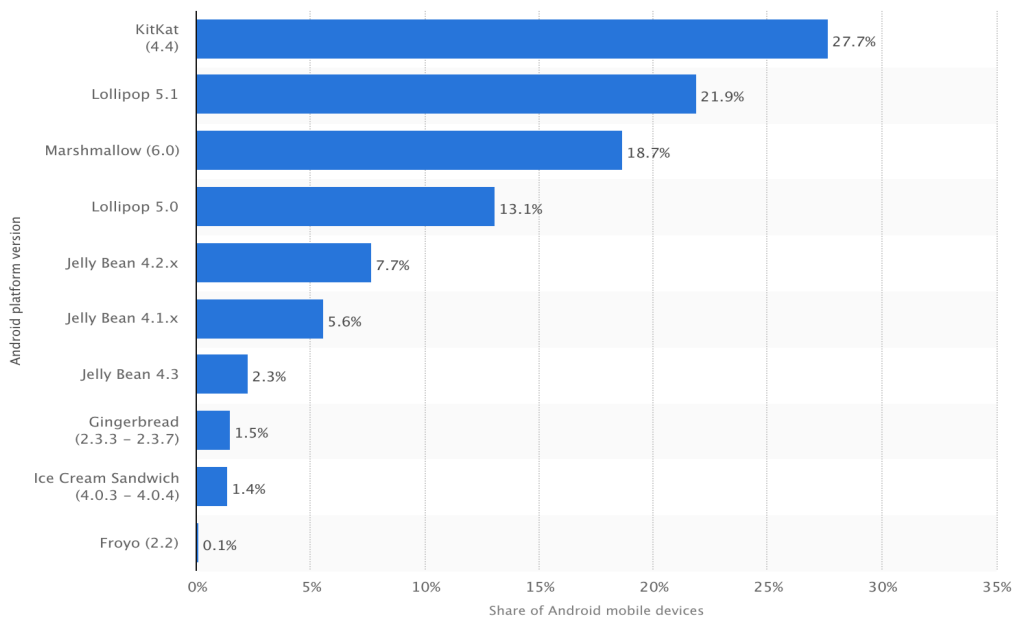
## FURPS+

### Functionality

- One of the main non-functional requirements for this application is it must have an easy to use interface. As technology is less prominent in rural area's it is key that the interface should be very intuitive to accommodate users who are not particularly adept at working with technology.

### Usability

- As previously stated this application is intended for the Android and iOS mobile platforms. It is proposed that the application should be compatible with all Android versions from 4.4 up due to their larger market share which can be seen in the diagram below and also be compatible with iOS 9 and up. ([Statista](#))



- Furthermore it is proposed that the application should be supported on all Android phones with a screen size greater than 4 inches.
- Additionally it is expected that a new user should be able to register and login to the main menu in less than 2 minutes.

### Reliability

- Based on the high android application failure rates documented in Blancco Technology Group Trend Report ([Blancco Technology Group, 2016](#)) it is proposed that the application created during the course of this project should have a crash rate of less than 5%.

### Performance

- It is proposed that the Application should have a maximum response time of 2 seconds for any operation implemented by the user.

### **Supportability**

- The main concept of supportability is being able to make necessary changes without further implications to the rest of the application. As this app is aimed at more than one industry, it must be kept relatively generic to facilitate many animals i.e. the app should not be explicitly developed for cows for example.

+

### **Security Requirements**

- It is proposed that requiring the user to enter a login ID and password will restrict access to user data when they open the app.

## Use Case Diagrams

The following use cases outline the functionality for the Android Application and remote GPS tracker.



## Brief Use Cases

### Register

**Name:** Register

**Actors:** Application User, Database System

**Description:** This use case begins when the application user chooses to create, an account on the android application. The application users will be required to create a user ID and password to register on the application. The use case ends when the application user has finished their work regarding their account.

### Login

**Name:** Login

**Actors:** Application User, Database System

**Description:** This use case begins when the application user opens the app. When the application user has opened the app they will be prompted to enter in their user ID and password. The user then clicks the login button located directly beneath the input fields and is redirected the application home screen.

### Logout

**Name:** Logout

**Actors:** Application User

**Description:** This use case begins when the application user wishes to logout of the app. The application user will proceed by clicking on the menu button located on the top left corner of each screen. A menu bar will then appear on the left hand side of the screen with the logout option at the bottom of the menu. The application user will then click the logout option which will display a prompt requesting them to confirm that they wish to logout. The user will then click the confirm logout button on this prompt. The application user is now logged out of the application.

### Change Password

**Name:** Change Password

**Actors:** Application User, Database System

**Description:** This use case begins when the application user navigates to the login screen and selects the Change Password button. After clicking the Change Password button a new screen is displayed where the application user inputs their old password followed by their new one. The application user then clicks the Request Change button that changes their password and returns them to the login screen.



### Analyse Paths

**Name:** Analyse Paths

**Actors:** Application User, Database System

**Description:** This use case begins when the application user selects the display animal path option in the application menu. The application user is then required to select the animal whose path data they wish to view. They then press the view button which displays the animals path data on a map.

### Display Current Location

**Name:** Display Current Location

**Actors:** Application User, Database System

**Description:** This use case begins when the application user selects the display animal path option in the application menu. The application user is then required to select the animal whose current location they wish to view. They then press the view button which displays the animals current location data on a map.

### Isolate Animal

**Name:** Isolate Animal

**Actors:** Application User, Database System

**Description:** This use case begins when the application user selects the Isolate Animal functionality in the application menu. The application user selects the animal they wish to isolate. They then press the save button to add the animal to a list of sick/injured animals in isolation.

### Add Animal Profile

**Name:** Add Animal Profile

**Actors:** Application User, Database System

**Description:** This use case begins when the application user chooses to add an animal profile in the application. Details such as the animals tag number(optional), type, weight, breed, gender and age are stored for each animal. The use case ends when the application user has finished their work regarding their animals.

### Process Login

**Name:** Process Login

**Actors:** Android Application System

**Description:** This use case begins when an application user requests login on the android application. The application users access credentials are then verified. The use case ends when the application user has or has not been granted access into the android application.

### Store Animal Data

**Name:** Store Animal Data

**Actors:** Android Application System

**Description:** This use case begins when an application user requests to change an animal's details on the android application. The database system stores the entered details. The use case ends when the application user has been notified of the success or failure of the current action.

### Retrieve Animal Data

**Name:** Retrieve Animal Data

**Actors:** Android Application System

**Description:** This use case begins when an application user requests to view an animal's details on the android application. The database system retrieves the information for the selected animal which includes information such as the animal's tag number, weight, breed, gender, age and location data for example. The use case ends when the application user is able to view the animal's data.

### Store User Credentials

**Name:** Store User Credentials

**Actors:** Android Application System

**Description:** This use case begins when an application user has requested to update their account details on the android application. This may include information such as their user ID, farm type and number of animals stored. The use case ends when the application user has been notified of the success or failure of the current action.

### Set GPS Intervals

**Name:** Set GPS Intervals

**Actors:** Application User

**Description:** This use case begins when an application user wishes to set the GPS Intervals. From the drop down menu the application user will select the Set GPS Intervals option. The application user will then select an animal from a drop down list and enter the length of time they want between each tracking request. The use case ends when the application user has set the GPS Intervals.

### Create Boundary

**Name:** Set Boundary

**Actors:** Application User, Database System

**Description:** This use case begins when an application user wishes to set the boundary for their animal. From the drop down menu the application user will select the Create Boundary option. The application user will then select an animal from a drop down list and enter the radius of the boundary they want to set for the animal from their current location. The use case ends when the application user has created the Boundary for the specified animal.

### Retrieve GPS Data

**Name:** Retrieve GPS Data

**Actors:** Android Application System

**Description:** This use case begins when an application user requests to view an animals details on the android application. The database system retrieves the information for the selected animal which includes information such as the animals tag number, weight, breed, gender, age and location data for example. The use case ends when the application user is able to view the animals data.

### Store GPS Configuration

**Name:** Store GPS Configuration

**Actors:** Android Application System

**Description:** This use case begins when an application user requests to store their GPS configuration on the android application. The database system retrieves the GPS information for the selected animal which includes information such as the intervals for GPS tracking. The use case ends when the application has stored the animals location with the specified intervals.

### Record GPS Coordinates

**Name:** Record GPS Coordinates

**Actors:** GPS Tracking Device, Database System

**Description:** This use case begins when the tracking device begins to record the GPS coordinates. The database system retrieves and stores the GPS information for the selected animal. The use case ends when the GPS coordinates have been recorded for the animal.

## Iteration Plan

### Iteration One

The main objectives for iteration one are as follows:

- Using PhoneGap implement the Login screen, the register screen and the change password screen.
- Using pythonanywhere create database tables to store user login details and user registration details.
- Using python implement the backend for the application which will be used to communicate with the MySQL database on pythonanywhere.
- Using PhoneGap Build produce a prototype android and iOS application with the above functionality.

### Iteration Two

The main objectives for iteration one are as follows:

- Using PhoneGap implement the Add Animal screen, the Display Current Location screen and the Set GPS Intervals screen.
- It is anticipated that the initial setup to allow the NanoTracker send GPS data to the database will also be implemented.
- Using python further implement the backend for the application which will be used to communicate with the MySQL database on pythonanywhere.
- Using PhoneGap Build produce a prototype android and iOS application with the above functionality.

### Iteration Three

The main objectives for iteration three are as follows:

- Using PhoneGap implement the Add Animal Path screen, the Create Boundary screen and the Isolate Animal screen.
- It is anticipated the functionality that allows the application change parameters on the NanoTracker will be implemented during this iteration.
- Using python further implement the backend for the afore mentioned functionality which will be used to communicate with the MySQL database on pythonanywhere.

## Bibliography

Blancoo Technology Group. (2016, May). *State of Mobile Device Performance and Health*. Retrieved November 1, 2016, from Blancoo Technology Group:  
<http://download.blancco.com/download/state-of-mobile-device-performance-and-health-report-q1-2016.pdf>

Statista. (n.d.). *Android version market share distribution among smartphone owners as of September 2016*. Retrieved November 1, 2016, from Statista:  
<https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/>