# CS221 Group Project

## Final Report

| | | |
|---|---|---|
| Project Team | Aldridge, William | Atkins, Max |
| | Barnes, Xander | Dart, Nicholas |
| | Harizanov, Maksim | O'Hare, James |
| | Pocock, Michael | Tuff, Sebastian |
| | Wilcock, Daniel | Wilmot, Andrew |
| Version | 0.1 | |
| Status | Pre-Release | |
| Date Published | February 12, 2015 | |
| Reference | SE_O2_FP_01 | |
| Department | Computer Science | |
| Address | Aberystwyth University<br>Penglais Campas<br>Ceredigion<br>SY23 3DB | |

# Contents

# 1 Management Summary

foobar

## 2 Project History

foobar

# 3 End of Project Report

foobar

# 4 Team Performance

foobar

# 5 Test Report

# 6 Maintenance Manual

## 6.1 Android

### 6.1.1 Brief Description of Final Product

The Android app first required user details including name, phone and email. It then also requires a (pre-made on the web site) site name which is downloaded from the API on app start up. The data is then validated and on successful validation will proceed on to a specimen entry screen.

The specimen entry screen takes a Latin plant name and a DAFOR scale represented by a slider. There is also an option for a scene and a specimen photo, which can be retaken as needed. Multiple Specimens can be added and reviewed in the specimen upload screen.

In the specimen upload screen, specimens can be edited or deleted by long pressing on the list. They can then be uploaded to the server.

### 6.1.2 Program Structure

The Android app is structured into several key areas:

*Activities*
Activity classes are bundled together and contain event listeners, “Async Tasks” and location listeners. They also contain all the validation for input fields as well as adding lists to “AutoCompleteTextBoxs”.

*Utilities*
The “Utils” package contains database utilities and classes for adding, searching, finding and deleting data from the databases. The following databases were implemented in the app:

1. Plants - Contains the Latin names
2. Specimens - Contains specimens the user has take for upload
3. Users - The information the user has entered for auto-completion
4. Sites - The sites that can be used, pulled from the API

There also exists a “OSGridReferHelper” class which is a third party class originally to be used to convert from GPS coordinates to a os grid reference for the specimen adder. It was decided that this feature was not required and so was not used.

*Data Classes*
These describe users, visits specimens and species and are used when reading/searching from and inserting into the database. The specimen class was not used as heavily as expected.

*Layout*
The layout of the Android activities are described in xml files, whose names follow the “activity_¡activityName¿” naming convention and are located in “src/main/res/layouts”.

### 6.1.3 Significant Algorithms

The following details significant algorithms that may require alterations in the future.

When the app starts, there is an initial “Async Task” thread spun off to download the sites list and update the database and to download the specimen list. This thread is located in the “LauncherActivity” class. This thread will check, upon download of the sites list, will add non-existent sites to the database. If the site list cannot be found, no viable error will be shown to the user, however the user may continue to use sites that already exist. The thread then proceeds to check if the latin names database is present and populated. If it contains entries, it will not download the Latin names list (as it is very large and slow to process, and so should be avoided). If it is not present, it will be downloaded and processed in a batch insertion to the database.

There is a location listener in the “SpecimenAdder” activity which gets the users location via gps when adding specimens. It does not however use course (network based) location to get this. This listener implements the standard Android interface “LocationListener”.

The databases are manipulated by Database Classes in the utilities package. These classes use the “DatabaseUtilities” class which inherit the standard “SQLiteOpenHelper” class. Tables are accessed via a class by their name eg “SiteDataSource” to access the sites table. These classes require a reference to the data class.

Upload of the final list of reserves is done in another “Async Task” in the “ReviewActivity” where a connection to the API is established and the data from the databases serialised to JSON and uploaded. At present the JSON is constructed manually. The list of specimens is then cleared (however the UI is not updated).

### 6.1.4 The Main Data Areas

The app makes extensive use of an “sqlite” database to store reserves, specimens, users and species. These are manipulated by classes for each table under the “Utils” package, which contain methods for selecting, inserting and deleting data from the databases, via the use of data classes related to the table’s content.

The “DataClasses” package contains classes primarily used for storing data to be entered into or read from the database. These classes contain getters and setters for the data. They also contain an “id” property which is only set when the data is read from the database, this is to allow for deletion of data via it’s unique id in the database.

Finally, we use “Adapters” in the android API to allow “AdapterViews” in the activity to access data. adapters

### 6.1.5 Files Accessed

The App accesses images located on the device which were taken in the specimen selection screen. These images are located on the root of the internal storage device.

The App also downloads the Latin names file from the remote location “http://nic-dart.co.uk/ nic/res/plantlist.json”, these are then processed into the sqlite database. This is done only in the init thread.

### 6.1.6 Interfaces

The app required a gps device to acquire location during the specimen addition process, as well as a camera if an image is desired.

### 6.1.7 Suggestions for Improvements

- Specimen and Scene images could be silhouetted and represent a scene and an specimen. IE not the same image.
- Deleting of multiple specimens
- Upload completion dialogue/confirmation
- Users and sites are specific to specimen, not read at upload time
- Auto complete lists not overlapped by keyboard
- Settings/Preferences menu or activity
- Custom GPS coordinates for specimen

### 6.1.8 Future Improvement Concerns

### 6.1.9 Limitations of the Program

- Deleting images before upload will most likely cause crash
- The "current user" details at the time of upload will be used.

### 6.1.10 Building from source

The source contains an "Android Studio" project file, this contains all the links to the files needed. It will require JDK (or OpenJDK 8) and the Android API and may need to be told where they reside. Building requires either a physical Android device of version 4.0.3 (API 15) or above, or alternatively a virtual Android device, created in the AVD.

# 7 Document History

| Version | Edit | Date | Persons |
|---|---|---|---|
| 0.1 | Initial Version | Febuary 12 2015 | nid21 |