Software Engineering Group 11
SE_11_PP
Project Plan

Author:	Tom Raikes (tor10), Alan Spence (als48), Aled Davies (add20), Theo Goree (tcg2), Richard Chowne (rhc15), Qiaoyang Zheng (qiz), Jack Skitt (jas78), Gavin Reynolds (gar18), Elliot Oram (elo9), Aloysius Fernandes (alf33), Kieran Dunbar (kid10)
Configuration Ref:	SE_11_PP
Date:	2014-10-30
Version:	1.1
Status:	Release

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Table of Contents

1. Introduction 4
1.1. Purpose of Document 4
1.2. Scope 4
1.3. Objectives 4
2. Overview 5
2.1. System Architecture Diagram 5
2.2. Client 6
2.2.1. Field Entry 6
2.2.2. Field Recall 6
2.2.3. Upload Screen 6
2.2.4. Settings 6
2.2.5. Model 6
2.2.6. Local Database 7
2.3. Server 7
2.3.1. Request Handler 7
2.3.2. Model 7
2.3.3. Update Handler 7
2.4. Web 7
2.4.1. Browse 8
2.4.2. Record View 8
2.5. Database 8
3. Use-Cases 9
3.1. Use-Case Diagram 9
3.2. Use-Case Descriptions 10
3.2.1. Android Application (RPSRrec) 10
3.2.2. Web Application (RPSRview) 11
3.2.3. Server Application (RPSRsrv) 11
4. User Interface Designs 12
4.1. Colour Swatches 12
4.2. Website Basic Concepts 13
4.2.1. Page Structure 13
4.2.2. Link Styles 13
4.2.3. Footer Links 14
4.2.4. Navigation Links 14
4.3. Web User Interface Designs 14
4.3.1. Index Login Page 14

4.3.2. Registration Page 15
4.3.3. You Page – Account 16
4.3.4. View Page - Database 17
4.3.5. New Page – Database Entry 18
4.3.6. Logout – You Page 19
4.3.7. About Page 20
4.4. Database Schema 21
4.4.1. Schema Diagram 21
4.5. Description 21
4.5.1. Species Tables 21
4.5.2. Reserves Table 21
4.5.3. Recordings Table 21
4.5.4. Species Occurrence Table 22
4.5.5. Users Table 22
4.6. Android User Interface Design 22
4.6.1. Main Screen 22
4.6.2. Login Screen 23
4.6.3. Recording Screen 24
5. Gantt chart 25
5.1. Term 1 25
5.2. Term 2 25
5.3. Gantt chart key 25
6. Risk Assessment 26
6.1. Absence 26
6.2. Programming Issues 27
6.3. Customer Related Issues 27
6.4. Software Issues 28
6.5. Dependencies 28
6.6. Additional Issues 29
6.7. Risk Assessment Key 29
7. REFERENCES 30
8. DOCUMENT HISTORY 30

1. Introduction

1.1. Purpose of Document

The purpose of this document is to show that we have considered the clients requirement specification and have created an initial design for the system architecture and laid out some objective and target dates to achieve these goals within. Our goals will be displayed using a Gantt chart (Page 25).

1.2. Scope

This document aims to cover our Plan of how to approach the Project including designs, milestones and risk analysis of our proposed system.

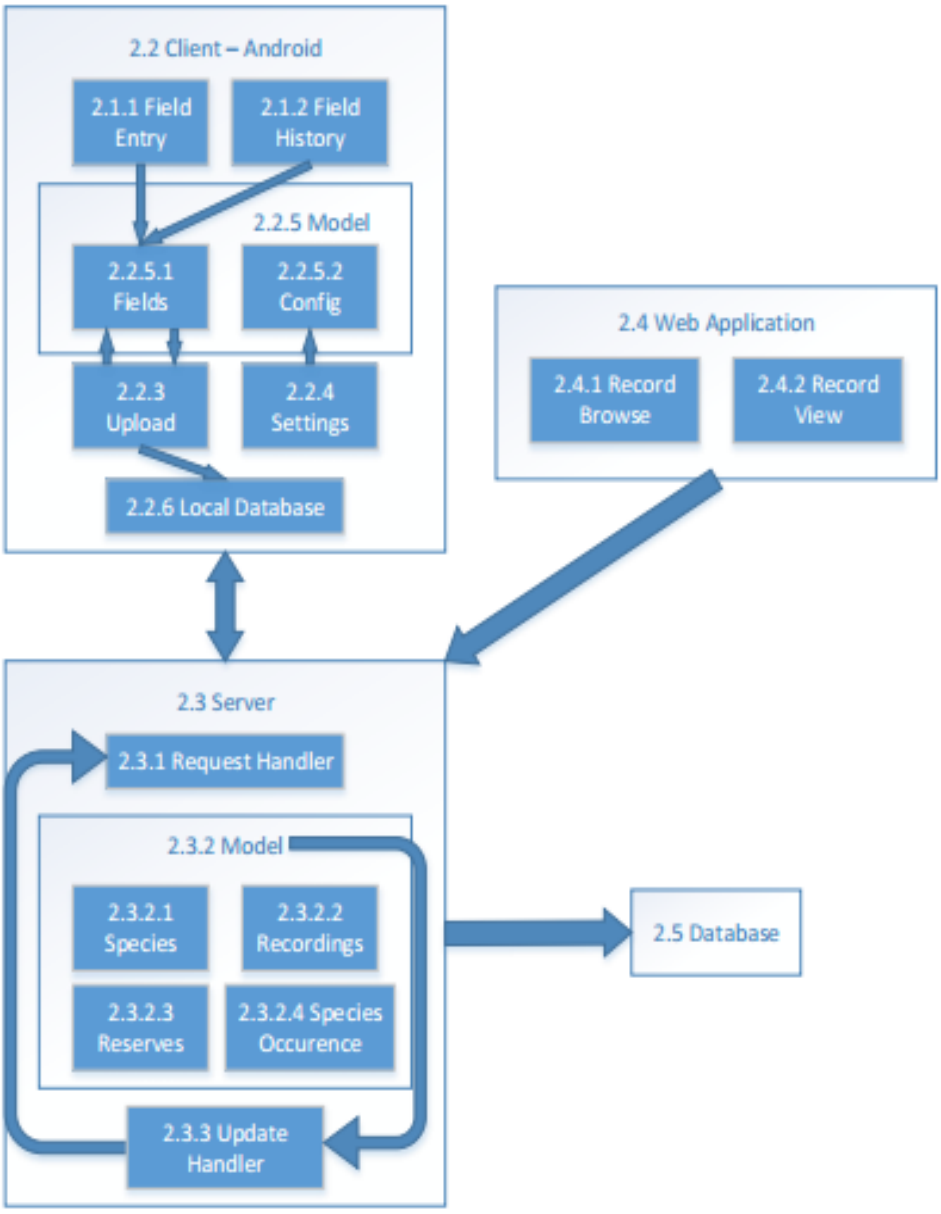
1.3. Objectives

This document covers:

- The general proposed architecture of our system.
- How we believe the user will interface with our system (use-cases).
- Our designs on the UI (both web and Android).
- The milestones and objectives for our Project (Gantt chart).
- A risk analysis of our Project.

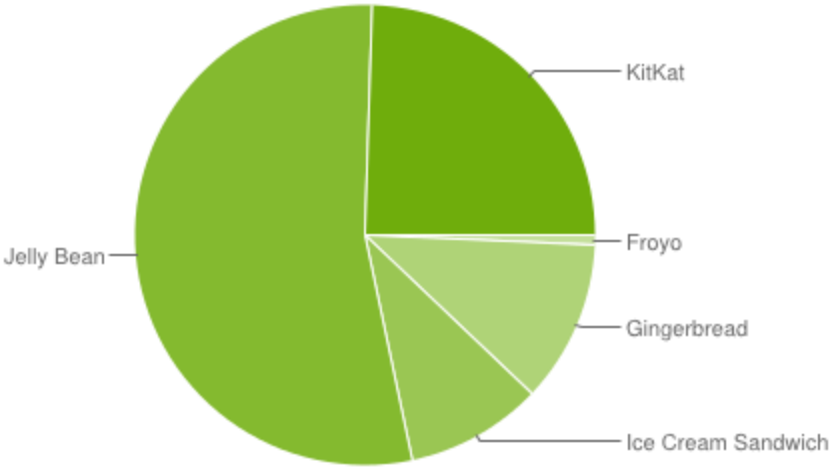
2. Overview

2.1. System Architecture Diagram



2.2. Client

Our Android application will be designed for Android version 4.0+, as we feel older versions are losing their market share [1]. Android version 4.0+ has 87.9% market share, made up of Ice Cream Sandwich, KitKat and Jellybean. The client describes the Android app and all its features.



2.2.1. Field Entry

This will be the primary screen for inputting data into the local database stored on the Android device. This can be accessed from the initial screen of the application.

2.2.2. Field Recall

This allows the user to view previous entries, accessing the same 'fields' model that the field entry screen has access to. This is also available from the initial screen of the application.

2.2.3. Upload Screen

This screen directly accesses the 'fields' model. Any field items recording by the Field entry screen (2.2.1) will be added to a queue ready to be uploaded to the external database. This screen should also have the feature to browse records and choose which to upload to the server and potentially being able to filter by date.

2.2.4. Settings

This screen interfaces with the local configuration file to allow users to store preferences locally on the device and have them remembered (i.e. Prefer Wi-Fi over Mobile network).

2.2.5. Model

This describes local content stored on the device, whether that's the configuration file or the entry fields ready to be uploaded to the database.

2.2.5.1. Model – Fields

These are the fields ready to be uploaded to the database, they will be stored locally until an internet connection can be accessed to upload records.

2.2.5.2. Model – Config

This is the locally stored user preferences file, this will be accessible locally and potentially via the database (client may wish to switch devices).

2.2.6. Local Database

This is not 'strictly' a database but a collection of records that will be stored ready to be sent to the server. Records can only be sent when an internet connection is achieved.

2.3. Server

This will utilise the universities LAMP server, this will handle communication between the application and database. This server will not physically contain the database due to security implications.

2.3.1. Request Handler

The request handler deals with communication between the client and server, its primary function is to send data back and forth between the client and server. Included in its functionality is the ability to check incoming data for errors and to confirm to the client that data had been received and added to the database.

2.3.2. Model

Transforms raw data from client into database readable information. This information is then passed to the update handler.

2.3.2.1. Model – Species

Sorts data related to species.

2.3.2.2. Model – Recordings

Sorts data related to recordings.

2.3.2.3. Model – Reserves

Sorts location data and reserve data.

2.3.2.4. Model – Species Occurrence

Sorts data related to species occurrence.

2.3.3. Update Handler

Takes information from the model (2.3.2) and creates a SQL statement using the received information ready to be sent to the database in a database readable format. Sends data to the request handler to confirm that information has been added to the database.

2.4. Web

The web application makes use of PHP to retrieve data from the database and display that as information on the web app. The PHP will also send data to the server and we will use JQuery to Login. It will also make use of HTML and CSS to make the website look professional and user friendly. We will design our web application to support modern web browsers, particularly those that hold a larger market share. The image below is taken from W3 schools and is their report on the market share of the most popular browsers [2]. We can interpret from the following information that our priorities will be Google Chrome and Firefox.

Browser Statistics

2014	Chrome	Internet Explorer	Firefox	Safari	Opera
September	59.6 %	9.9 %	24.0 %	3.6 %	1.6 %
August	60.1 %	8.3 %	24.7 %	3.7 %	1.8 %
July	59.8 %	8.5 %	24.9 %	3.5 %	1.7 %
June	59.3 %	8.8 %	25.1 %	3.7 %	1.8 %
May	59.2 %	8.9 %	24.9 %	3.8 %	1.8 %
April	58.4 %	9.4 %	25.0 %	4.0 %	1.8 %
March	57.5 %	9.7 %	25.6 %	3.9 %	1.8 %
February	56.4 %	9.8 %	26.4 %	4.0 %	1.9 %
January	55.7 %	10.2 %	26.9 %	3.9 %	1.8 %

2.4.1. Browse

This module of the web application will allow the user to browse records which have been retrieved from the database, the user will option to search/filter the records to make the browsing more user friendly.

2.4.2. Record View

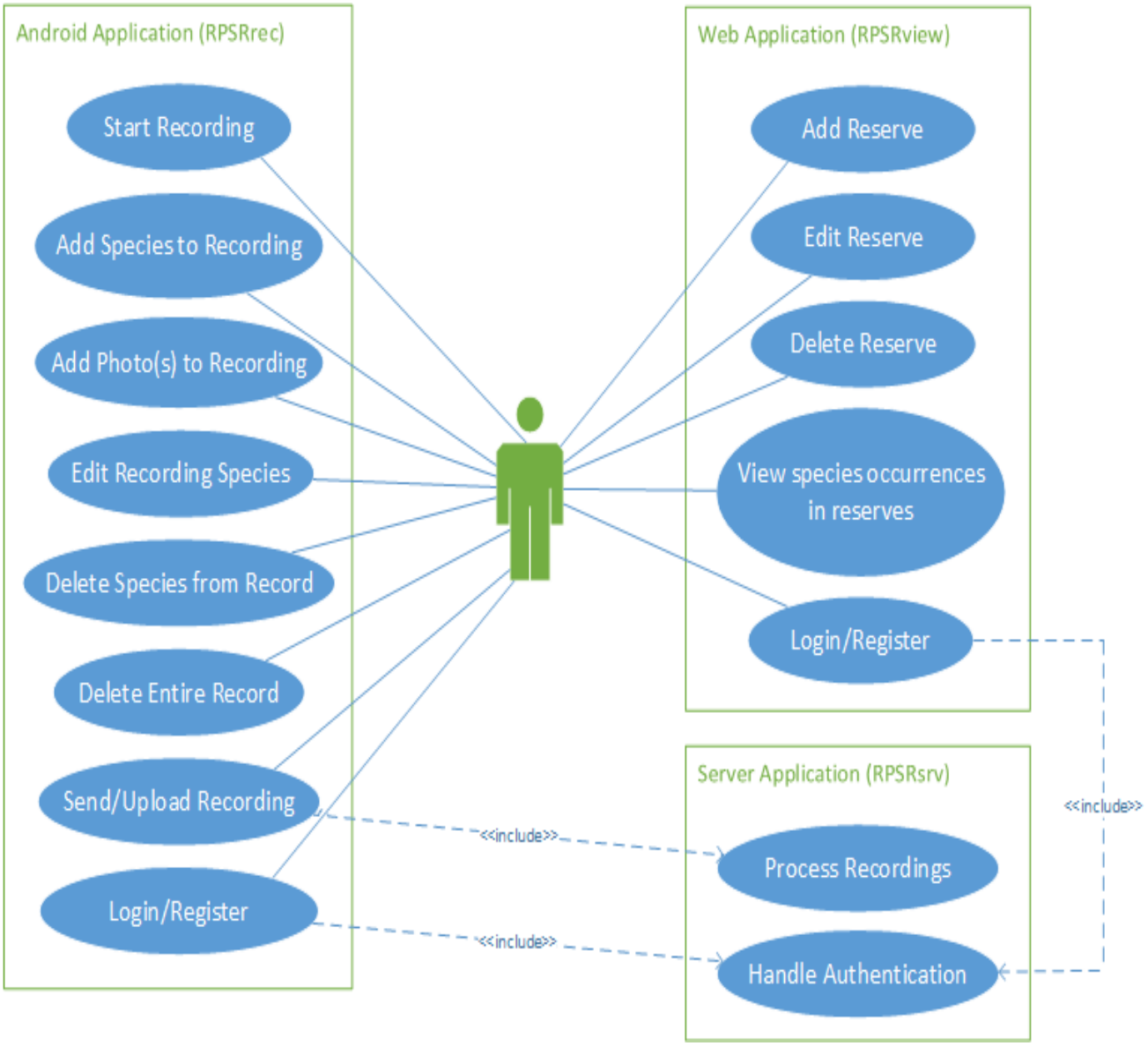
This module is a more detailed view of a particular species.

2.5. Database

This will be where processed information will be stored, it will either be MySQL or PostgreSQL. It will consist of several tables each containing data relevant to the table.

3. Use-Cases

3.1. Use-Case Diagram



3.2. Use-Case Descriptions

3.2.1. Android Application (RPSRrec)

Requirement	**Use Case**	**Description**
FR1 and FR2	Start Recording	User should be able to start a recording by entering details of the reserve. He/she may also have to Login or Register (see use case below).
FR3 and FR4	Add Species to Recording	User should be able to add a new or existing species to the recording by entering details such as location, abundance and any comments.
FR4	Add Photo(s) to Recording	User should also be able to add general and specimen photos to the species occurrence in the recording.
FR5	Edit Recording Species	The user should also be able to edit any of the species specified in a particular recording.
FR5	Delete Species from Record	The user should be able to delete a species occurrence from a recording.
FR5	Delete Entire Record	The user should also be able to delete an entire recording, including all of its associated species occurrences.
FR6	Send/Upload Recording	The user should be able to submit details of the recording through to the server (RPSRsrv) once finished.
N/A - Optional Feature	Login/Register	The user should be able to register for an account and subsequently be able to login using that account to use the application.

3.2.2. Web Application (RPSRview)

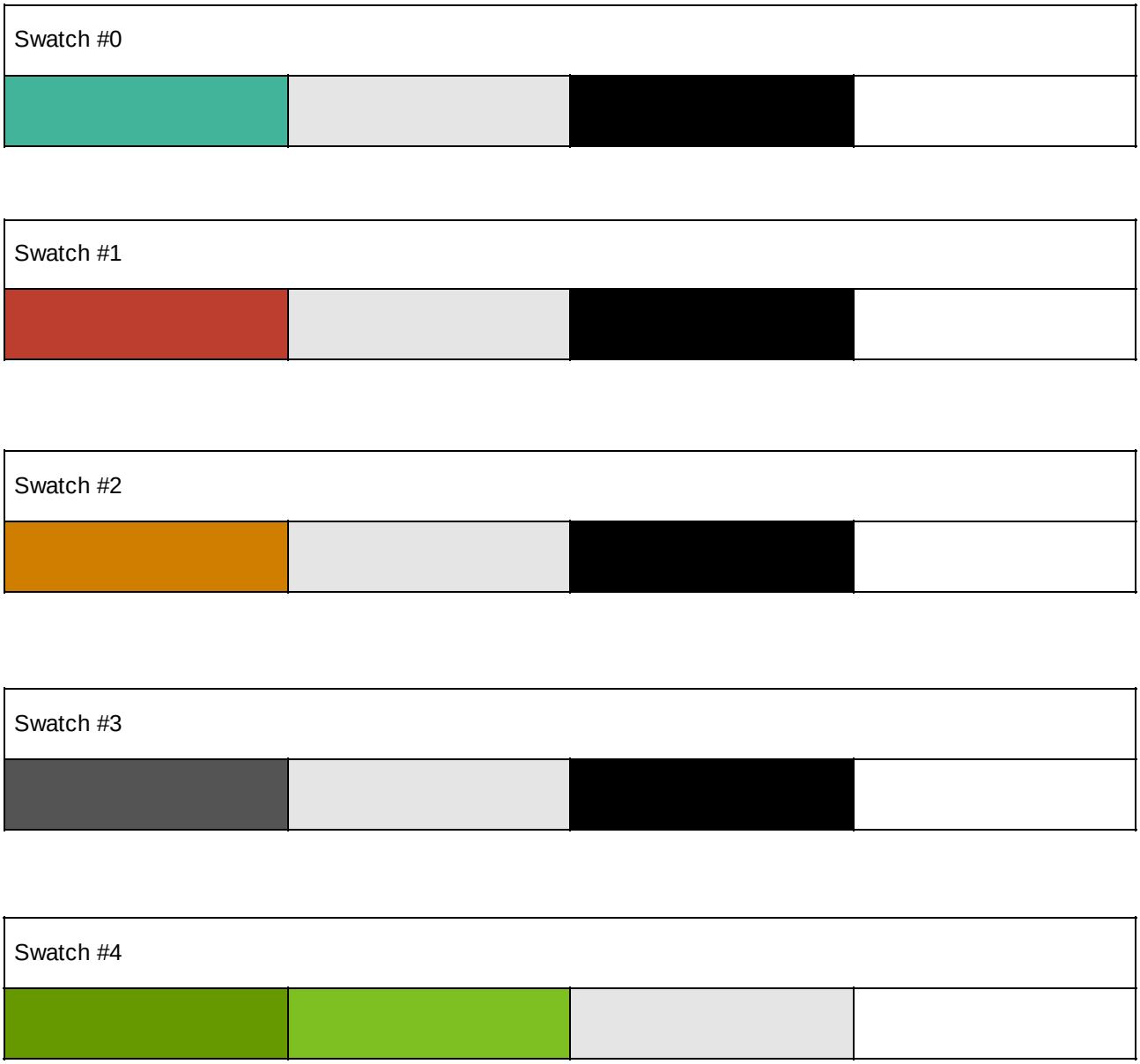
Requirement	Use Case	Description
FR8	Add Reserve	User should be able to create a new reserve by entering details such as its name, location and description.
FR8	Edit Reserve	The user should also be able to edit a reserve's details.
FR8	Delete Reserve	The user should also be able to delete a reserve from the system.
FR9	View species occurrences in reserves	The user should be able to view all recordings (including all details of any species occurrences in that recording) from a particular reserve.
N/A - Optional Feature	Login/Register	The user should be able to register for an account and subsequently be able to login using that account to use the application.

3.2.3. Server Application (RPSRsrv)

Requirement	Use Case	Description
FR7	Process Recordings	When the user submits a recording (or multiple recordings), the server should process these and store them in the database.
N/A - Optional Feature	Handle Authentication	When the user attempts to log in or register, the server should handle this by either checking login details or by adding a new user to the database.

4. User Interface Designs

4.1. Colour Swatches



Hex References
References in order as to how the swatch is organised (left to right)

Swatch #0

#42B49A #E5E5E5 #000000 #FFFFFF

Swatch #1

#BD4030 #E5E5E5 #000000 #FFFFFF

Swatch #2

#D17F00 #E5E5E5 #000000 #FFFFFF

Swatch #3

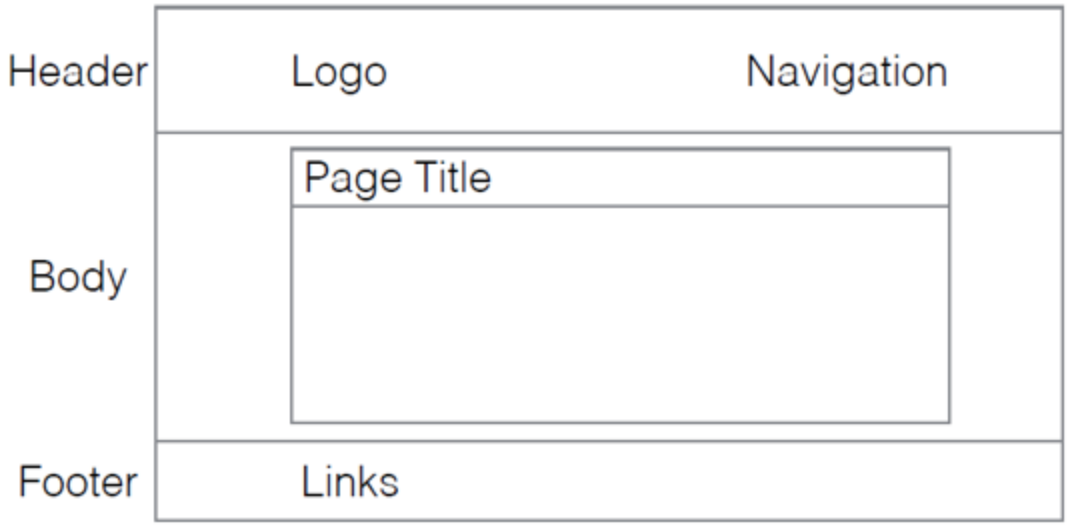
#545454 #E5E5E5 #000000 #FFFFFF

Swatch #4

#669900 #7FC123 #E5E5E5 #FFFFFF

4.2. Website Basic Concepts

4.2.1. Page Structure



4.2.2. Link Styles

- a:link (white)
- a:visited (white)
- a:hover (white)
- a:active (grey)
- Standard link (white)
- When on Page (grey)

4.2.3. Footer Links

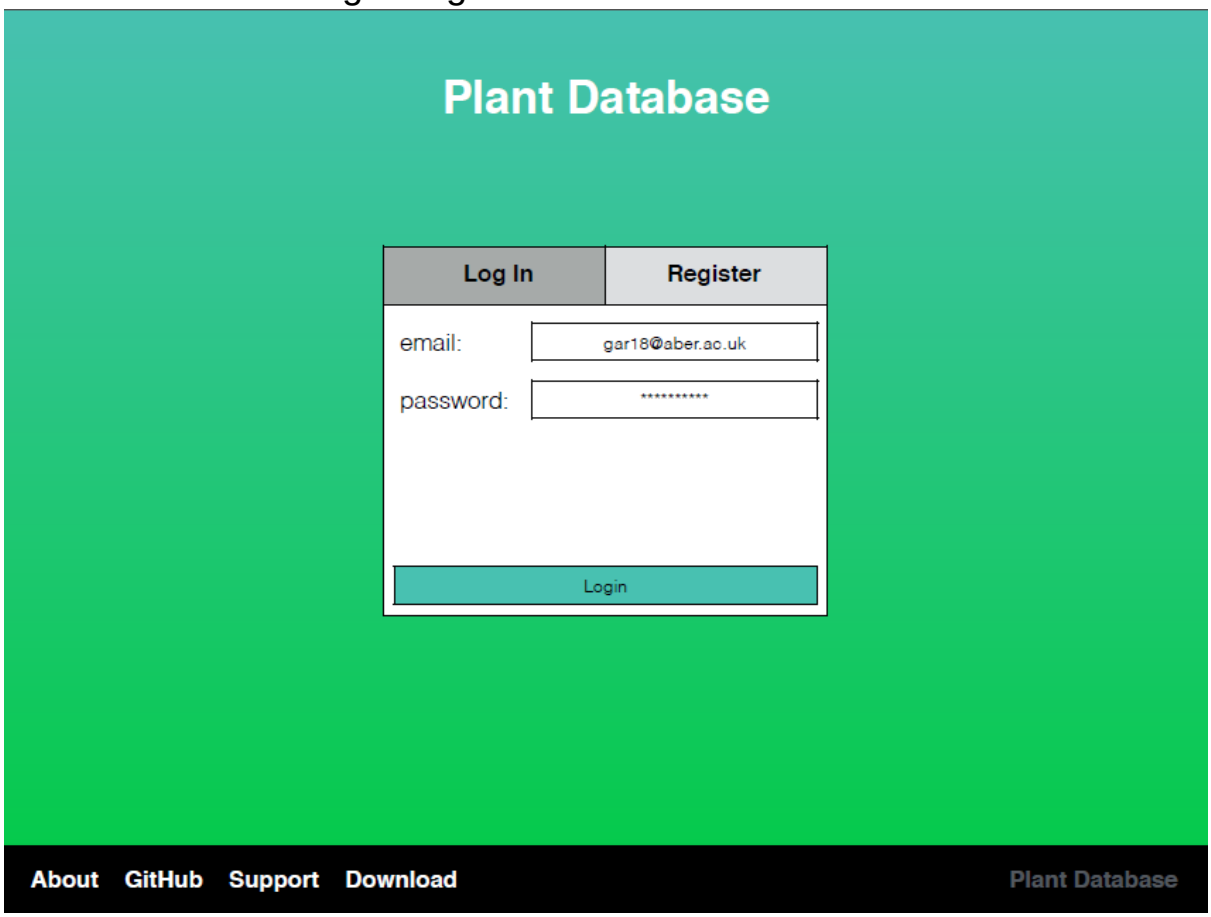
- About links to the About page
- GitHub links to the GitHub Repository
- Support will link direct to an email address

4.2.4. Navigation Links

- You
- View
- New

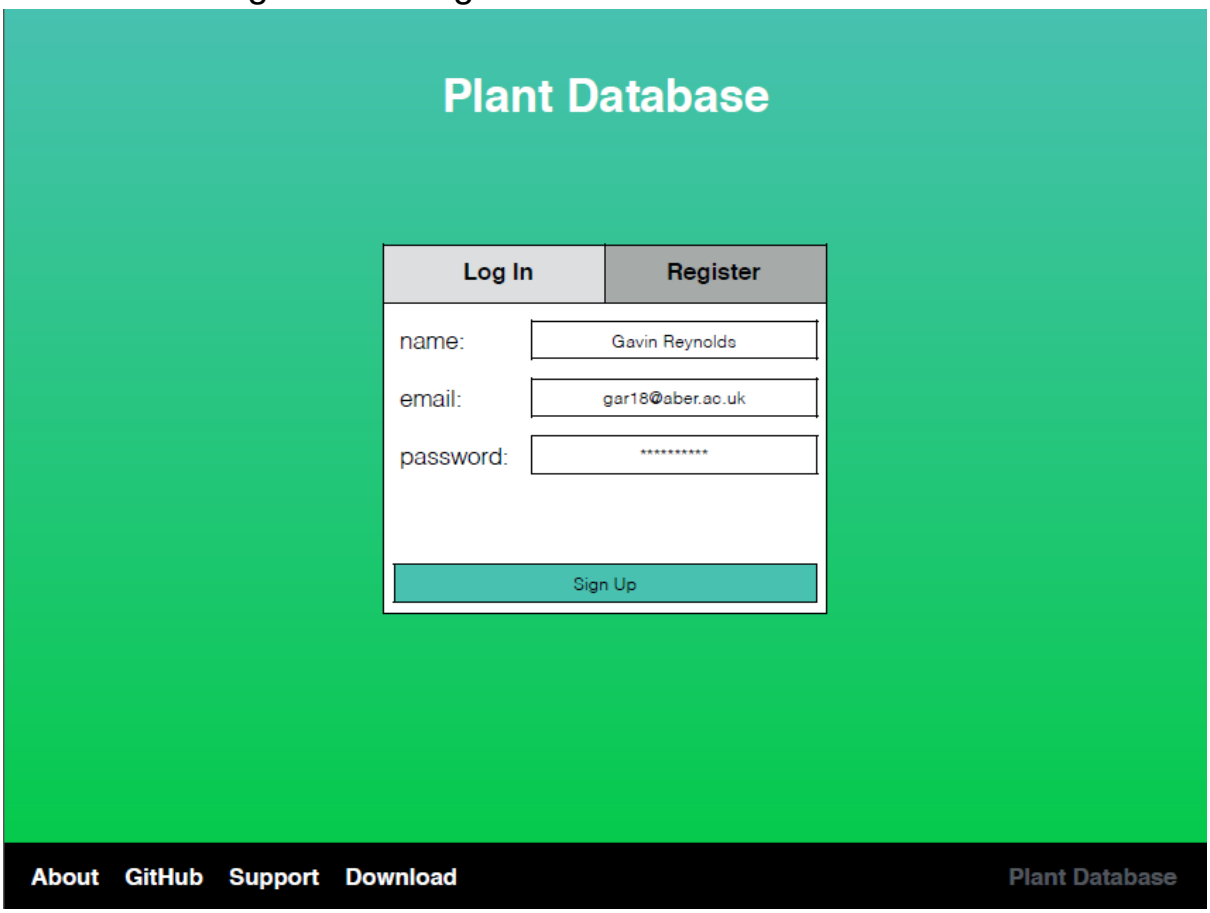
4.3. Web User Interface Designs

4.3.1. Index Login Page



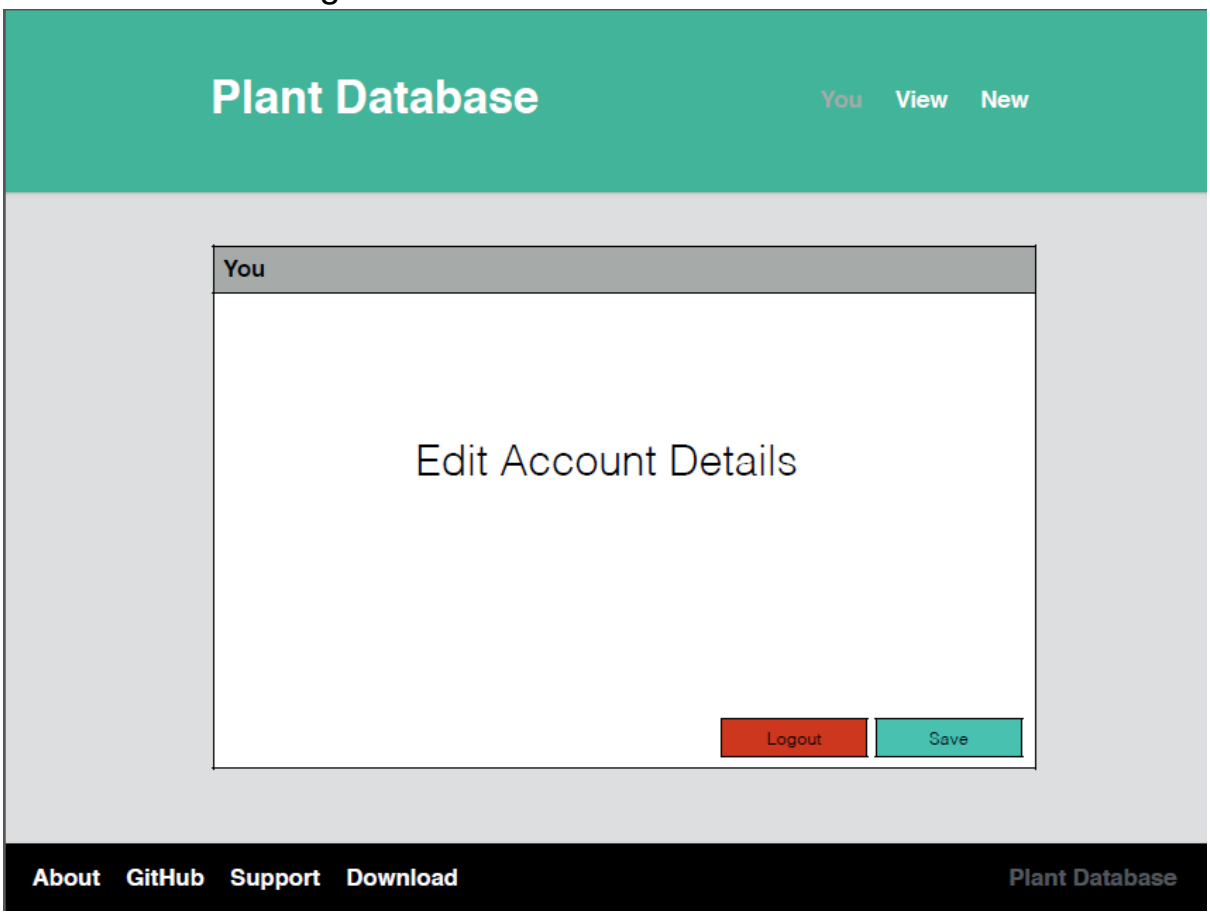
This Is the Main Login page where the user has entered their email address and password they would now click Login to proceed to the main website.

4.3.2. Registration Page



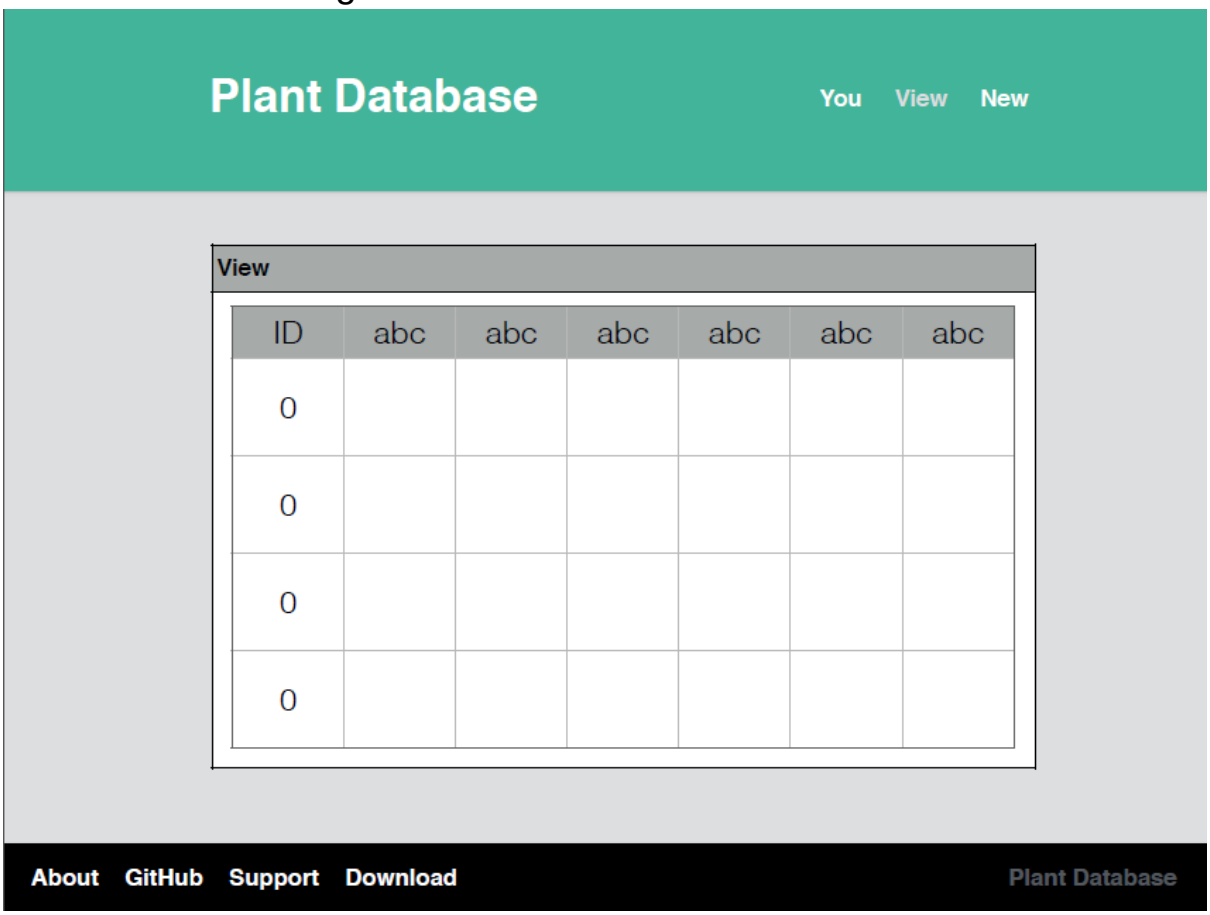
If the User does not have an account they will click the Register tab and then need to enter their Name, email and password and click Sign Up to register an account. After this their data will be sent to a database which then can be used to allow the user to Login.

4.3.3. You Page – Account



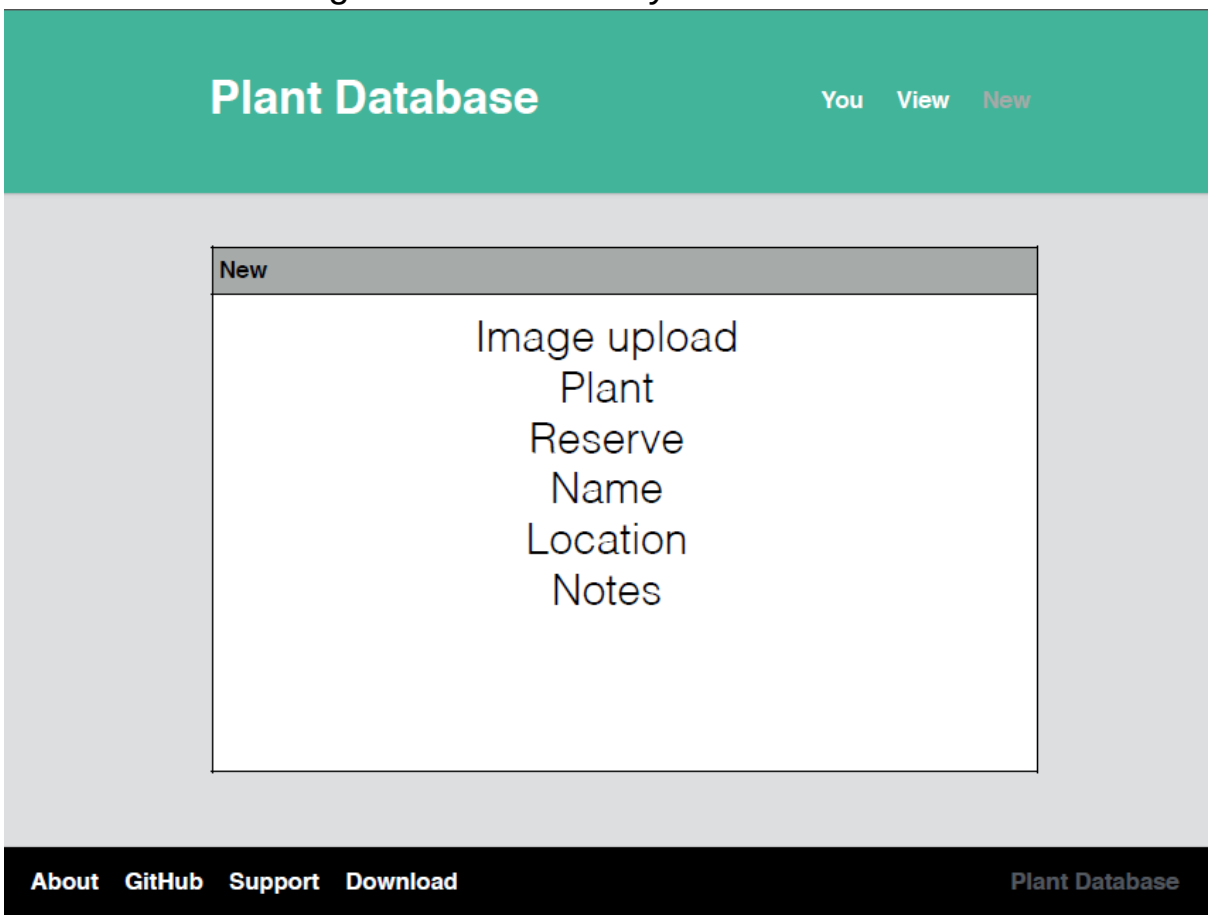
The user is brought to the You page by default. The user can edit their name, email and password here.
They can save their details here once edits have been made.

4.3.4. View Page - Database



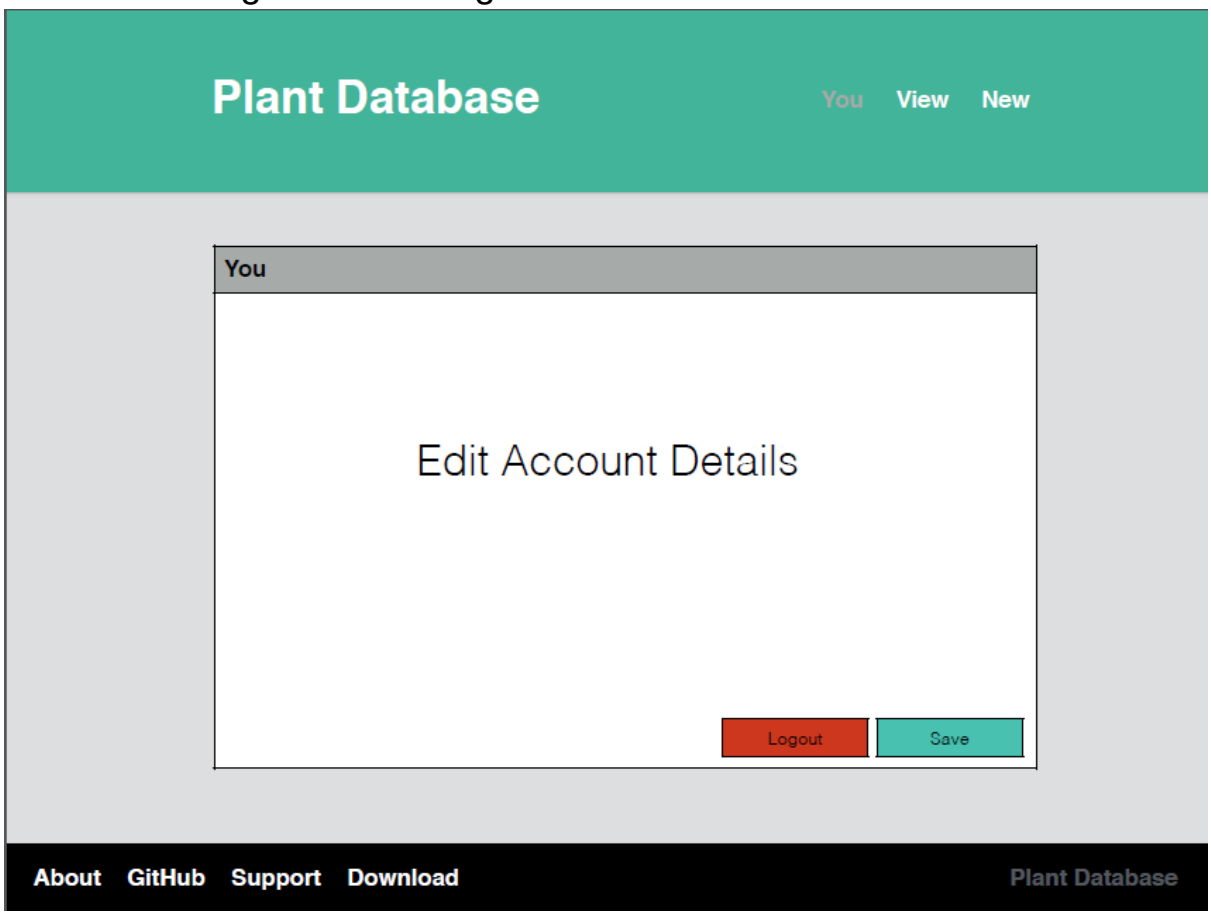
The View page allows the user to view the database records.

4.3.5. New Page – Database Entry



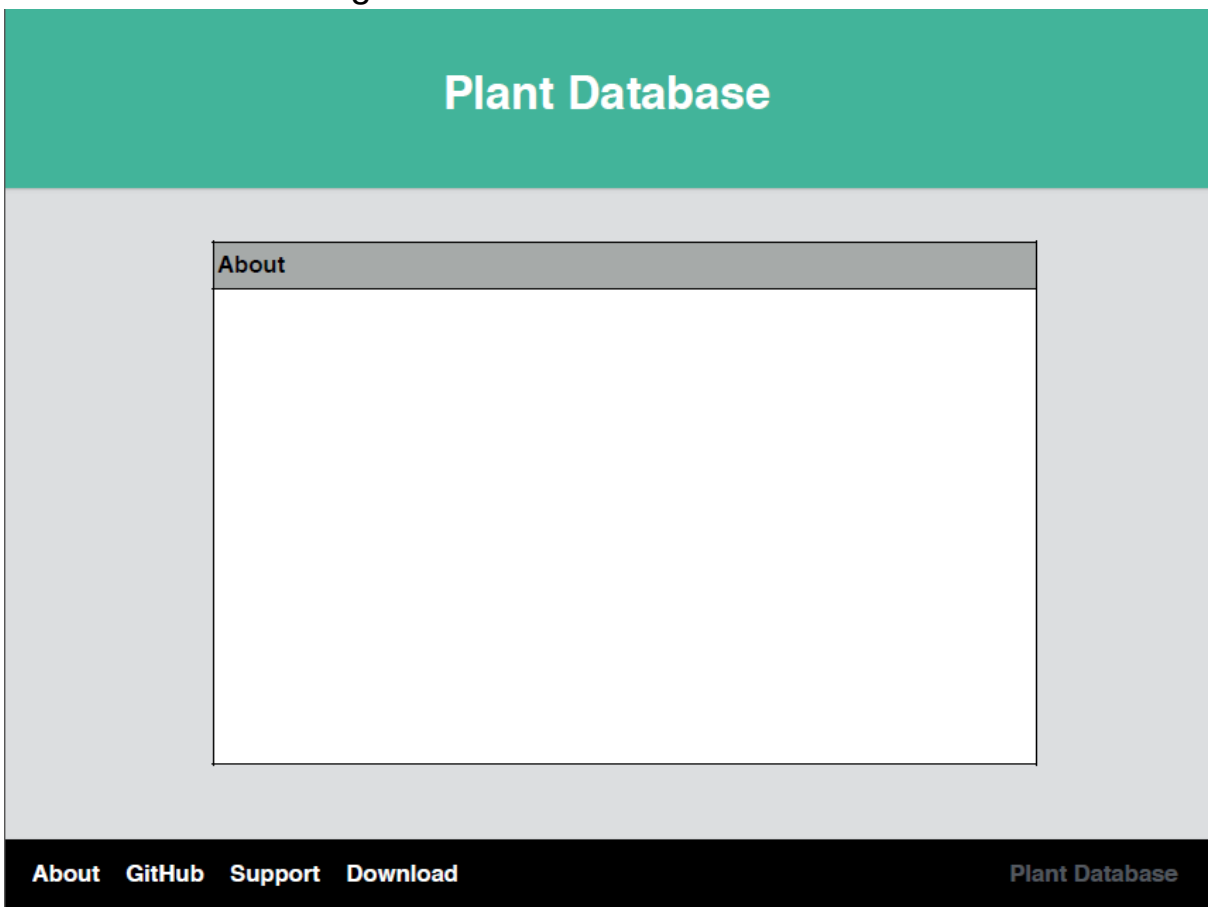
The New page allows the user to create a new record. This includes an image, the plant name, which reserve it was found on, its common name, its location and any extra notes.

4.3.6. Logout – You Page



Back to the You page now and the user can click Logout to logout when they are ready.

4.3.7. About Page

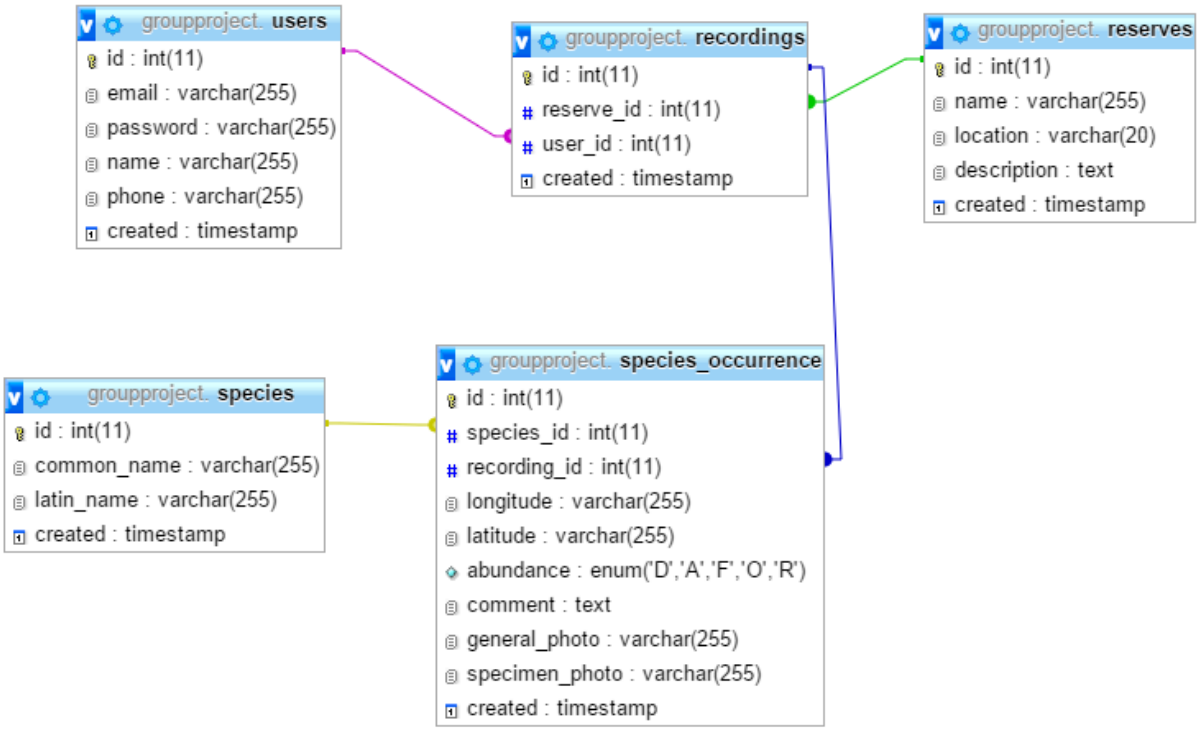


The footer has links on it too. They are present both when logged in and logged out. This is shown as if the user is logged out and wanting to read about the Plant Database application. There is also a download link to download the App.

4.4. Database Schema

4.4.1. Schema Diagram

Below is a graphical view the DB schema along with any relations. This was created using a MySQL database.



4.5. Description

4.5.1. Species Tables

So this table would contain both the common and Latin names for the species, which we can obtain from the BSBI list. These would be stored as varchars. It also contains the ID as a primary key and a created timestamp.

4.5.2. Reserves Table

The reserves table would contain the name, location (OS grid ref as per spec) and description of the reserve. The name and location would be stored as varchar and the description would be stored as text. As before it would contain the ID as the primary key and a created timestamp.

4.5.3. Recordings Table

The recordings table would contain a reserve ID which would be related to the primary key of the reserves table. It also contains the user's ID which would be related to the primary key of the users table. As before it also contains its own ID as the primary key and a created timestamp.

4.5.4. Species Occurrence Table

The species_occurrence table contains both the species ID and the recording ID which are both related to the relevant tables. This would contain each individual occurrence of a species on a per recording basis. This way we would be able to get all the different species recorded on a particular recording and/or on a particular reserve. This table also contains the longitude and latitude of where the species was found, the abundance, a comment, and URLs (?) to both the general and specimen photos. The longitude, latitude and photos will be stored as varchars whereas the comment will be stored as text. In the case of abundance I have used enum as we're choosing from a specific range of options. As before it also contains ID as its primary key and a created timestamp.

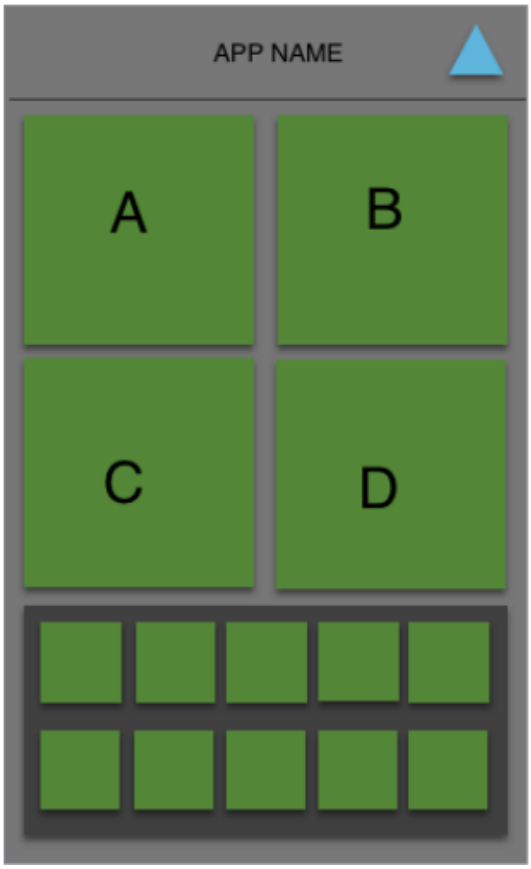
4.5.5. Users Table

The users table contains the email, phone number and name of the user along with his/her encrypted password. All of these fields are stored as varchar (including phone number due to the fact that it starts with a 0). As before this table also contains ID as its primary key and a created timestamp.

4.6. Android User Interface Design

4.6.1. Main Screen

This is the main title screen it holds the various control buttons for the outline, this allows the users to see all the options available to them. It can also be used for the management of data and a visual representation of currently stored local data. There's a login button at the top to allow a user to quickly load their application data.



4.6.1.2. Section A/B

Upload button/create new plant recording.

Section B is used to manage previous records

4.6.1.3. Section C

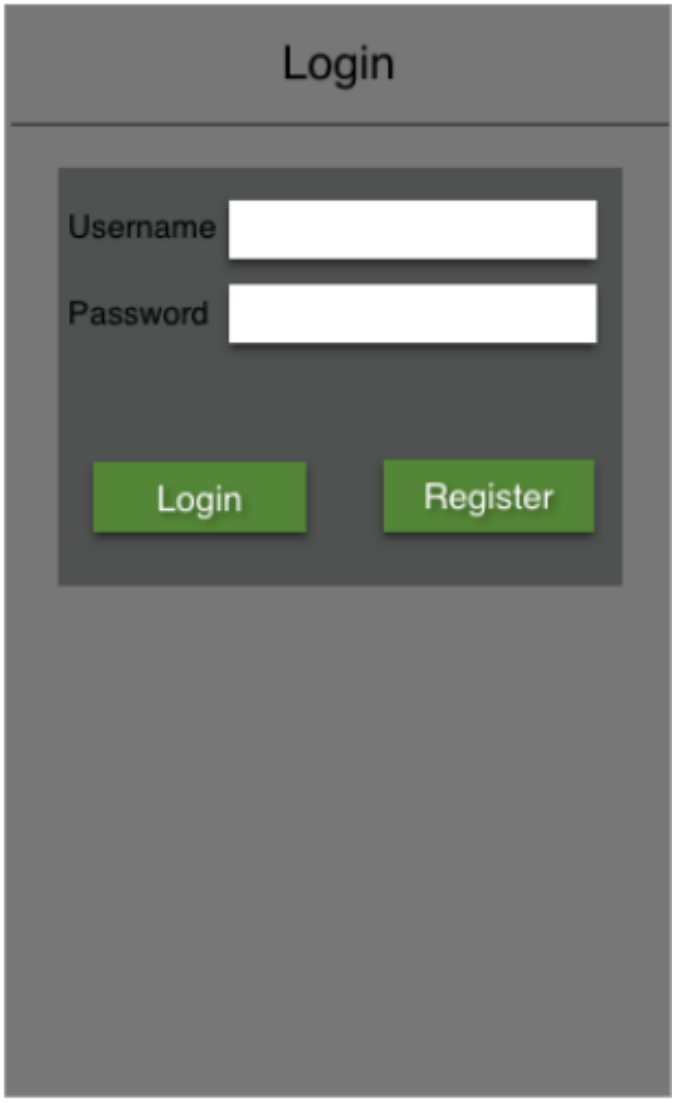
Used for Plant Creation

4.6.1.4. Section D

Plant Data Management

4.6.2. Login Screen

Standard login screen allows for the registration or login of accounts.



4.6.3. Recording Screen

This allows for the recording and submission of plant data. The image moves left and right in order to switch between close up images and area images, for multiple images you can wipe down on the category.

The plant and reserve boxes allow you to search the relevant databases for the best result, the search screens should be a standard list menu with a search box at the top.

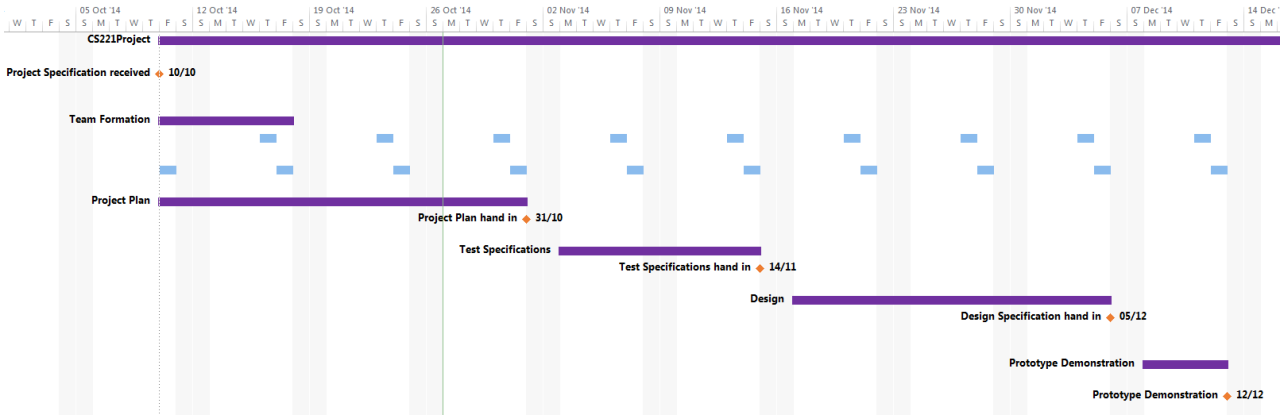
The name box allows the user to enter a name for the record they are adding.

The location box has two options a GPS location pulled from the phones sensors or a manual location entering, this is needed for recordings added offsite and for devices that do not have a GPS sensor.

The notes box allows the botanist to add any extra details that they feel is relevant to the location

5. Gantt chart

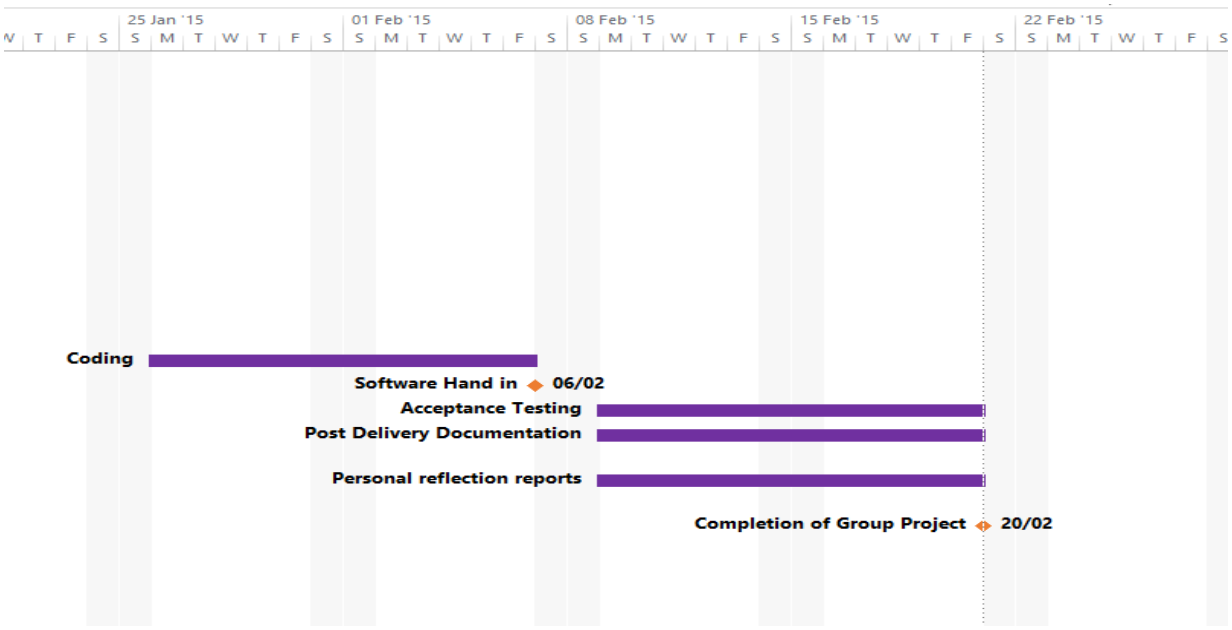
5.1. Term 1



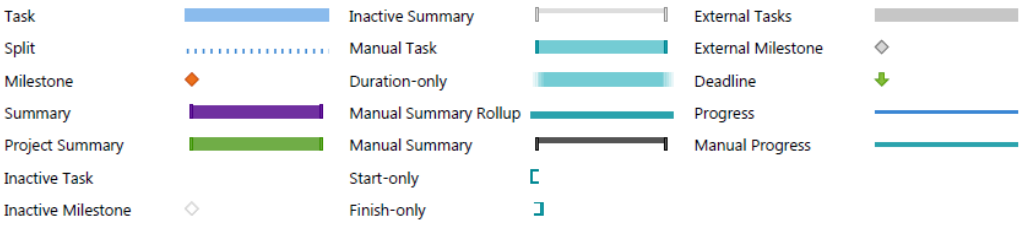
The first row of tasks repeating on Thursdays is a weekly unofficial meeting for the group.

The Second row of tasks repeating on Fridays is a weekly official meeting for the group with Project Manager Chris Loftus.

5.2. Term 2



5.3. Gantt chart key



6. Risk Assessment

6.1. Absence

Event	Likelihood	Severity	Risk	Who it affects	Mitigation
Project Manager absence	0.1	0.6	6%	Team	Transfer the Manager's role to the Project leader ensuring he is aware of the duties involved in managing the task. Contact the Manager with the meeting minutes.
Project Leader absence	0.2	0.3	14%	Team	Make sure all group members are aware of what their upcoming duties are for the Project. The session will be run by the Deputy Project Leader.
Quality Assurance Manager absence	0.2	0.5	10%	DQA & PL	Deputy Quality Assurance Manager is to take over the writing for meeting minutes and other Quality Assurance Manager activities. The QA Manager is to inform the group prior to the meeting that they will not be able to attend.
Deputy Manager absence	0.3	0.3	9%	QA & PL	Providing that the specific Manager is present, little action should need to be taken other than that which would be if a regular team member was absent. If necessary a temporary Deputy Manager will be asked to take over the duties for a meeting.
Team member absence	0.5	0.3	15%	Team	Team members are expected to make contact in advance of missing a meeting. Furthermore, they should ensure they read the minutes from the meeting and are clear on their upcoming tasks for the week. If multiple meetings are missed and work is not completed, it may become necessary to divide the absent members workload to other team members.
Loss of communication with team member	0.3	0.7	21%	PL	Attempt to contact the member in question in person. If no contact can be made then inform the Project Manager of the situation. The Project Manager should then carry out appropriate action regarding the continued absence.

6.2. Programming Issues

Event	Likelihood	Severity	Risk	Who it affects	Mitigation
Network connection handling	0.3	0.4	12%	Team	The code to handle connecting to the network is a fairly key point of the Project. Some team members are currently performing spike work to investigate the feasibility of implementing a queue like structure to handle delayed uploading.
Poor communication between sub teams	0.3	0.6	18%	Sub Teams	If the sub teams are not working closely with one another it may cause problems when it comes to linking sections of the Project. The Team leader and sub team leaders are in charge of ensuring team collaboration is good. This should ensure integrating each teams software should go well.

6.3. Customer Related Issues

Event	Likelihood	Severity	Risk	Who it affects	Mitigation
Change of specification	0.1	0.9	9%	Team	If the specification under goes drastic changes then the group will schedule a meeting as soon as possible to address the severity of the changes. If no meeting time can be established, the changes will be assessed by the available members of the group and all others will be emailed and informed of the investigations outcome.
Additional features added by customer	0.2	0.7	14%	Team	If additional features are added by the client, the team in charge of the section those changes refer to will be immediately contacted. The new features will then become a matter of priority to implement providing the Project has reached the stage where they should be implemented.
Tailoring the Project for the audience	0.4	0.6	24%	Team	Attempted frequent contact with the client will be advised in order to ensure the end deliverable matches their needs. Furthermore, the audience needs should be taken into consideration at every stage of development.

6.4. Software Issues

Event	Likelihood	Severity	Risk	Who it affects	Mitigation
GitHub downtime	0.1	0.9	9%	Team	Take regular backups of data stored on GitHub to ensure Project data can be restored. If GitHub is consistently down and the repository cannot be accessed, the group will consider alternative solutions such as BitBucket or other.
Communication downtime	0.1	0.8	8%	Team	Try to establish a physical meeting with as many member of the group as possible. The meeting will be used to establish a new method of communication.
Database server downtime	0.2	0.8	16%	Client	Have a backup database to ensure no loss of data. Inform users with a message if the database cannot be accessed.
Website server downtime	0.2	0.8	16%	Client	Have a message displayed to the user if the website is currently down due to maintenance or other circumstances.

6.5. Dependencies

Event	Likelihood	Severity	Risk	Who it affects	Mitigation
Lack of Android testing devices	0.1	1	10%	Team	Most members of the group own an Android device. Those that do not will have access to the emulator available with the eclipse IDE.
Over dependency on singular member	0.2	0.8	16%	Team	The Project Manager is responsible for ensuring that the work load is evenly spread between all group members. This should ensure that the group can complete all aspects of the Project evenly.

6.6. Additional Issues

Event	Likelihood	Severity	Risk	Who it affects	Mitigation
Feature Creep	0.3	0.5	15%	Client	The Project Manager and Project Leader should be monitoring the progress of the Project overall to ensure that no feature creep takes place.
Drifting from the initial specification	0.2	0.8	16%	Client	As the Project is broken down into smaller subsections it will be harder for the whole group to go off on a tangent. Furthermore, The Project Manager and Project Leader will monitor this to ensure it does not take place.
Poor quality of work	0.4	0.6	24%	Team	All Work should be submitted earlier than the date of the deliverable to either the QA or Deputy QA Manager for checking. Any work that is not satisfactory will be amended by the producer or a member of the QA team.
Team members missing deadlines	0.3	0.2	24%	Team	All deadlines are set out well in advance of their completion date. Group members are highly encouraged to email Managers with concerns on meeting deadlines or producing content.
Underestimation of activity timings	0.3	0.4	12%	Team	Monitoring from the Project Leader should ensure that the group remains on schedule. If certain tasks take longer than expected extra group members can be allocated to help.

6.7. Risk Assessment Key

Risk	**Acceptable**	**Low**	**Medium**	**High**	**Severe**
Percentage (%)	0-20	21-35	36-50	51-75	76-100

7. REFERENCES

[1] https://developer.Android.com/about/dashboards/index.html (21/10/14)

[2] http://www.w3schools.com/browsers/browsers_stats.asp (30/10/14)

8. DOCUMENT HISTORY

Version	CCF No.	Date	Changes made to document	Changed by
1.0	N/A	28/10/14	Document Created and Structured Into new Template Supplied by als48. Document Put into Review	Tcg2
1.1	#1,#2,#3,#4,#5,#6,#7	30/10/14	Document Reviewed and Issues assigned. Changes made and document placed in Release Status	Tcg2