

# CS221 Group Project

## Project Plan

|                |  |   |
|----------------|--|---|
| Project Team   | Aldridge, William<br>Barnes, Xander<br>Harizanov, Maksim<br>Pocock, Michael<br>Wilcock, Daniel | Atkins, Max<br>Dart, Nicholas<br>O'Hare, James<br>Tuff, Sebastian<br>Wilmot, Andrew |
| Version        | 0.5  |   |
| Status         | Pre-Release  |   |
| Date Published | February 3, 2015   |   |
| Reference      | SE_O2_PM_00  |   |
| Department     | Computer Science   |   |
| Address        | Aberystwyth University<br>Penglais Campus<br>Ceredigion<br>SY23 3DB                            |   |

Copyright ©Aberystwyth University 2014

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                       | <b>4</b>  |
| 1.1      | Purpose of this Document . . . . .        | 4         |
| 1.2      | Scope . . . . .                           | 4         |
| 1.3      | Objectives . . . . .                      | 4         |
| <b>2</b> | <b>Overview</b>                           | <b>5</b>  |
| 2.1      | Platforms . . . . .                       | 5         |
| 2.1.1    | Android . . . . .                         | 5         |
| 2.1.2    | LAMP Server . . . . .                     | 5         |
| 2.2      | High Level Architecture . . . . .         | 6         |
| 2.2.1    | Android Application (RPSRrec) . . . . .   | 6         |
| 2.2.2    | Website (RPSRview) . . . . .              | 6         |
| 2.2.3    | Server (RPSRsrv) . . . . .                | 7         |
| 2.2.4    | Interaction of Components . . . . .       | 7         |
| 2.2.5    | External Interface Requirements . . . . . | 7         |
| 2.2.6    | Performance Requirements . . . . .        | 7         |
| 2.2.7    | Design Constraints . . . . .              | 7         |
| 2.2.8    | Miscellaneous Requirements . . . . .      | 7         |
| 2.3      | Target Audience . . . . .                 | 7         |
| <b>3</b> | <b>Use Cases</b>                          | <b>9</b>  |
| 3.1      | Use Case Diagram . . . . .                | 9         |
| 3.2      | Use Case Descriptions . . . . .           | 10        |
| 3.2.1    | Android OS User . . . . .                 | 10        |
| 3.2.2    | Website user . . . . .                    | 10        |
| 3.2.3    | Admin . . . . .                           | 11        |
| <b>4</b> | <b>User Interface Designs</b>             | <b>12</b> |
| 4.1      | Android Interface . . . . .               | 12        |
| 4.1.1    | New Site Visit . . . . .                  | 12        |
| 4.1.2    | Adding a new species . . . . .            | 12        |
| 4.1.3    | Editing and saving a site visit . . . . . | 12        |
| 4.2      | Web Interface . . . . .                   | 13        |
| 4.2.1    | Web Homepage . . . . .                    | 13        |
| 4.2.2    | Plant Database Page . . . . .             | 13        |
| 4.2.3    | Add Record Page . . . . .                 | 13        |
| 4.2.4    | Plant Details Overlay . . . . .           | 13        |
| 4.2.5    | Plant Location Overlay . . . . .          | 13        |
| 4.2.6    | Remove Plant Record . . . . .             | 13        |
| 4.3      | Mockups . . . . .                         | 13        |
| <b>5</b> | <b>Gantt Chart</b>                        | <b>21</b> |

|   |           |
|---|-----------|
| <b>6 Risk Analysis and Mitigation</b>               | <b>22</b> |
| 6.1 Ongoing Risk Assessments . . . . .              | 22        |
| 6.2 Documentation . . . . .                         | 23        |
| 6.3 Software Development And Deliverables . . . . . | 23        |
| <b>7 References</b>                                 | <b>24</b> |
| <b>8 Document History</b>                           | <b>25</b> |

# 1 Introduction

## 1.1 Purpose of this Document

This document has been commissioned to show our current understanding of the client's requirement specification of Reserve Plant Species Recording ("RPSR"). The project encompassing the Android Application Framework and modern Web Technologies which will provide a monitoring system for definable Areas of Interest, into a series of basic objectives and milestones. RPSR has three main component parts, an Android Application, a Website and a Database.

The document will give a high level overview of the time line for developing and testing RPSR, an estimate of the final User Interface design, a probabilistic risk assessment, and a use case breakdown.

The client is to read this document and confirm that their requirements have been understood and correctly interpreted by the team.

## 1.2 Scope

The document comprises the high level design and development plan for RPSR, it should not be referred to for final software design structure. This document will follow the design specification.

This document provides an overview of the system we have interpreted for the client, including our choice of platform (where applicable), high-level architecture and will also included a section dedicated out understanding of the target audience of the application.

A use case diagram and corresponding table will show how the actors of the system are expected to interact with elements of the system. The document will also contain a probabilistic risk assessment for the project in full. This document will provide a first draft of the UI designs and are subject to alterations at a later date.

## 1.3 Objectives

The objective of this document is to show our initial plan for development of RPSR, from initial UI designs and user interaction plans, through a breakdown of deliverable dates on a Gantt chart and the expected issues that could arise during development. This is as follows:

1. Provide a high level overview of system design and interaction
2. Define the technologies used for the system
3. Display the initial web UI
4. Display the initial Android UI
5. Breakdown the project into milestones
6. Define a limited scope risk assessment for the project and the plans to mitigate the risks

## 2 Overview

### 2.1 Platforms

#### 2.1.1 Android

The client placed a specific request for the application to run on Android devices. As yet no response has been received for a minimum version, so in keeping with more up to date releases, we will use Android API 15 (Android 4.0.3 which encompasses the vast majority of new smart phones)

#### 2.1.2 LAMP Server

We will be using a LAMP (Linux, Apache2, MySQL, and PHP) ready server. Linux will be version Gentoo 3.12, MySQL will be version 14.14, PHP will be version 5.5.18. This will provide us with the tools ready to develop the website, and interact with the database. PHP was chosen to handle the server side processing of received data due to it being free and wide availability. The language is also covered in other modules during the project time line, meaning it will be fresh in the minds of the web team. MySQL is the most commonly used database software on web servers and is available on the university servers.

References for Version: Linux: <https://www.gentoo.org/> Apache2: <http://httpd.apache.org/> MySQL: <http://www.mysql.com/> PHP: <http://php.net/>

## 2.2 High Level Architecture

The system consists of the following high-level elements:

### 2.2.1 Android Application (RPSRrec)

The application fulfils the following roles:

- |     |   |
|-----|---|
| FR1 | Startup of software   |
| FR1 | Allow the user to record a new visit each time they complete one  |
| FR2 | Collect information about a new visit from the user   |
| FR2 | Collect time and date information from the phone for the recordings   |
| FR3 | Allow the user to select a species from the database  |
| FR3 | Add a species to a recording  |
| FR4 | Take a photo using an Android device by capturing a new photo or selecting one from the device's library  |
| FR4 | Obtain location data from the GPS unit within and Android device to include in the recording  |
| FR4 | Allow the user to enter data for each species<br>RESTATEMENT OF FUNCTIONAL REQUIREMENTS ???   |
|     | <ul style="list-style-type: none"> <li>• Typical location</li> <li>• Abundance using “DAFOR” scale</li> <li>• Free text comment</li> <li>• Photo of a general scene at a typical location</li> <li>• Photo's of the specimen</li> <li>• Allow the user to enter a name of a new species if not currently available</li> </ul> |
| FR5 | Allow the user to edit and delete local (not yet uploaded) recordings and the species data within them  |
| FR5 | These local recordings will be stored on the device storage with SQLite until they are ready to be sent to the server.  |
| FR6 | Upload the collected recordings to the remote database server whenever a network connection becomes available   |

The underlying platform of execution for this subsystem is the Android operating system.

### 2.2.2 Website (RPSRview)

The website fulfills the following roles:

- |     |  |
|-----|--|
| FR3 | Add a species to a recording                     |
| FR4 | Allow the user to enter data for each species    |
| FR8 | Provide a detailed view of individual recordings |
| FR8 | Enable maintenance of the recordings database    |
| FR9 | Browse and search uploaded recordings            |

The platform of execution of this subsystem will be the LAMP stack, making use of PHP as a scripting language and Apache2 as a HTTP server.

### 2.2.3 Server (RPSRsrv)

The server plays an essential middle-man role in the system, providing persistent storage for RPSRview and RPSRrec, and allows for exchange of data (recordings) between the two.

The server fulfills the following roles:

- |     |   |
|-----|---|
| FR7 | Provide a public Web API to be used by the website and the mobile application, enabling safe HTTP access to stored recordings |
| FR7 | Provide a MySQL database for the Web API to use as a data store   |
| FR7 | Ensure data integrity and security  |

The platform of execution of this subsystem will be the LAMP stack. PHP, the language, and Apache2, the HTTP server, will support the Web API, while MySQL will provide the database back-end.

### 2.2.4 Interaction of Components

### 2.2.5 External Interface Requirements

- |      |   |
|------|---|
| EIR1 | The program should be intuitive to regular computer users |
|------|---|

### 2.2.6 Performance Requirements

Reasonable expectations of the relevant software parts of the product:

- |     |   |
|-----|---|
| PR1 | User input should be reflected on screen within one second  |
| PR2 | Software products should run appropriately on their respective platforms: <ul style="list-style-type: none"><li>• The app on android devices</li><li>• Apache and php on the web server</li></ul> |

### 2.2.7 Design Constraints

Features and limitations set forth by the user or implied by reasonable implementation:

- |     |   |
|-----|---|
| DC1 | Java must be used for all Android development by corporate policy. All Java will be built in the Android Studio IDE                                 |
| DC2 | The API will be developed in php and will be server-side only   |
| DC3 | Functionality of software must be shown by exploration of at least 2 reserves, with at least 2 recording visits with overlapping species recordings |

### 2.2.8 Miscellaneous Requirements

- |      |  |
|------|--|
| MSC1 | Project will be developed in line with Group Project QA guidelines |
|------|--|

## 2.3 Target Audience

The client stated:

*The system will be used by naturalists who are familiar with standard computer interfaces. They are concerned with accuracy of recording and they may have to operate in difficult weather conditions and in remote locations.*

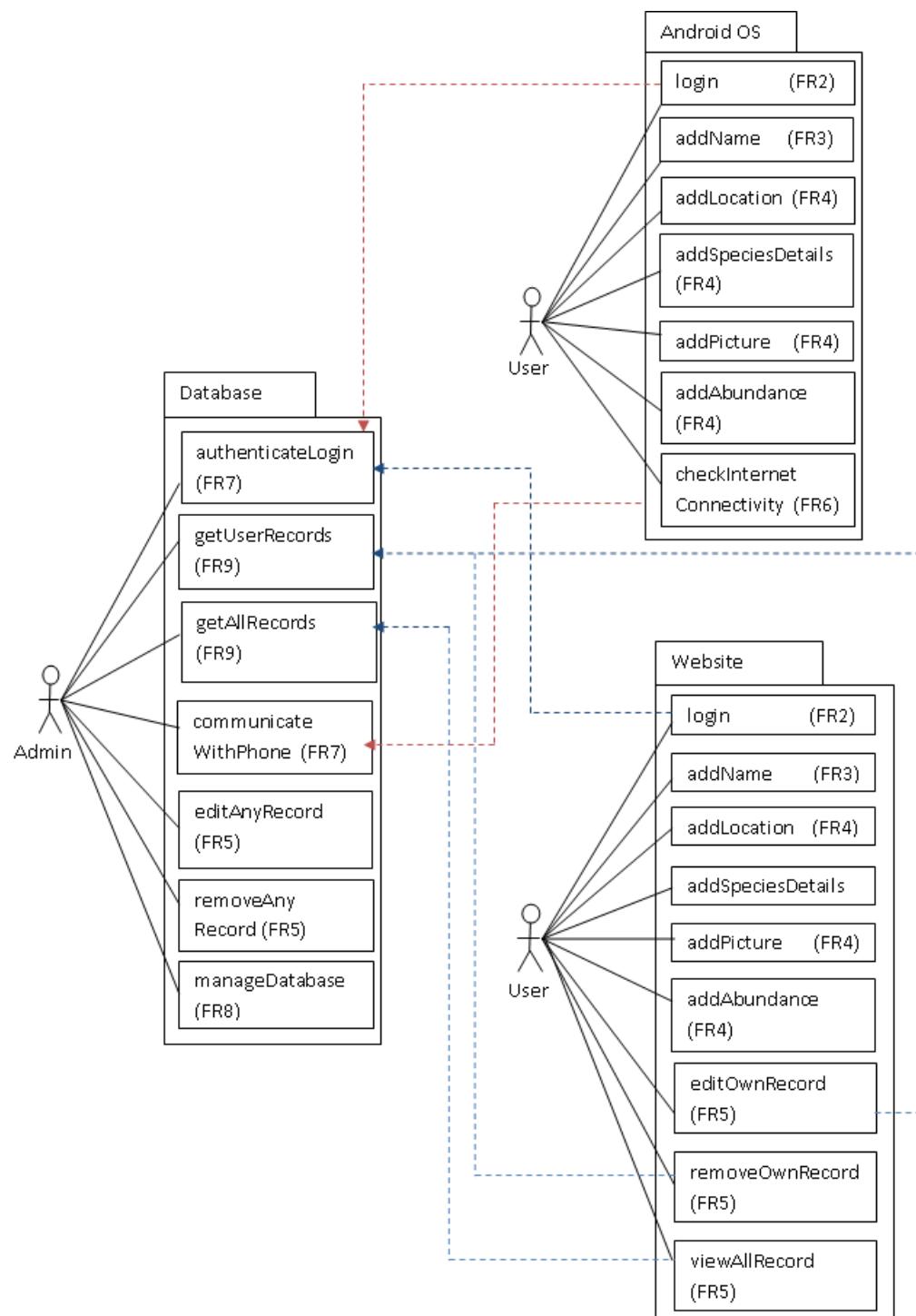
From this we understood the users will be competent in the basic interaction with Android and Web User Interfaces. We have designed the application to have large input areas to aid in data entry when outdoors but also recognize that devices used have limited screen real estate so user interfaces should be simple and uncluttered. We have also understood the user may be entering data in areas where a real time connection to the server (RPSRsrv) is not possible so data collected should be stored until a connection can be established.

The system may also be used for education purposes within a school environment and so we will also target teachers and young students. From this we understood that these users will also benefit from the above requirements since this kind of app would be used on school trips.

We also believe the app could be beneficial in research and so the app must also look professional and efficient.

### 3 Use Cases

#### 3.1 Use Case Diagram



## 3.2 Use Case Descriptions

### 3.2.1 Android OS User

|     |                           |   |
|-----|---------------------------|---|
| FR2 | login                     | User should be able to log into the application by entering a valid username  |
| FR3 | addName                   | User should be able to add the name of a plant to the database  |
| FR4 | addLocation               | User should be able to add the location of a plant to the database  |
| FR4 | addSpeciesDetails         | User should be able to add a description of the plant to the database   |
| FR4 | addPicture                | User should be able to add a picture of the plant to the database   |
| FR4 | addAbundance              | User should be able to record the level of abundance of the plant within the area   |
| FR6 | checkInternetConnectivity | The application should be able to know if it is connected to the internet and is able to send records to the database. If not the data is stored in local storage |

### 3.2.2 Website user

|     |                   |  |
|-----|-------------------|--|
| FR2 | login             | User should be able to log into the application by entering a valid username           |
| FR3 | addName           | User should be able to add the name of a plant to the database                         |
| FR4 | addLocation       | User should be able to add the location of a plant to the database                     |
| FR4 | addSpeciesDetails | User should be able to add a description of the plant to the database                  |
| FR4 | addPicture        | User should be able to add a picture of the plant to the database                      |
| FR4 | addAbundance      | User should be able to record the level of abundance of the plant within the area      |
| FR5 | editOwnRecord     | User should be able to make changes to any record that they have added to the database |
| FR5 | removeOwnRecord   | User should be able to remove any record that they have added to the database          |
| FR5 | viewAllRecord     | User should be able to view any record in the database through the website             |

### 3.2.3 Admin

|     |                      |  |
|-----|----------------------|--|
| FR7 | authenticateLogin    | The server will allow a user to log onto the application through their phone or through the website providing that the user has entered correct username and/or password. Using a database which contains details of valid usernames and passwords |
| FR9 | getUserRecords       | When the user wants to view edit or delete records they have entered the server will need to call them from the database   |
| FR9 | getAllRecords        | The server will send information about all the records in the database to the user upon request  |
| FR9 | communicateWithPhone | Server must be able to send and receive data from the phone of records being entered   |
| FR5 | editAnyRecord        | The website admin should be able to edit any record entered in the database by any user  |
| FR5 | removeAnyRecord      | The website admin should be able to remove any record entered into the database by any user  |
| FR8 | manageDatabase       | The database administrator will be able to log into the database and manage/maintain it  |

## 4 User Interface Designs

### 4.1 Android Interface

This section displays the envisioned design of the Android Application layout (Fig 1).

#### 4.1.1 New Site Visit

The first thing that will be presented to the user on entry to the app will be the option to create a new visit by pushing the button provided. This will then load the Basic Information page.

The user will then be prompted to input their basic information such as their name, phone number, email and site location for reference. These are to be validated by the server upon the push of the Next button. Add a species page will then be displayed to the user.

Further fields may be required to be added at a later date due to additional requirements given by the client or by a change during further development in the design.

There is no requirement in the spec to remember the users details on the app so currently the user will have to fill in their details each time they make a recording.

#### 4.1.2 Adding a new species

The first input asked of the user is to select a species to record using an integrated search function. This will not only help the user find the species they are looking for, but also allow the user to add a new species to the database if required (Fig 2).

An option to provide a GPS signal is given which will link up with the GPS in the android device to provide a location. Options are given to provide pictures of the site and specimen either from the camera or the gallery on the device. A field for adding notes will be provided that may be useful to the record (Fig 3).

#### 4.1.3 Editing and saving a site visit

The edit and save page provides the options to the user to view and edit the recordings made. The Edit Species link allows the user to access a list of the species pages they have added to the visit. This takes the user back to the Add a Species page to edit or delete an entry.

Users are given the option to delete the visit which will be met with a prompt to confirm or cancel the delete.

The final link is to save all recordings, giving permission to upload the data to the database at an appropriate time.

## 4.2 Web Interface

This section displays the envisioned design of the Website Layout.

### 4.2.1 Web Homepage

The Homepage has a central search bar to search the database for visit records and species information. The navigation bar at the top remains consistent over all pages with breadcrumbs just below to ensure ease of navigation. There is a link to the site map in the footer which also remains consistent. There will be a heading welcoming the user, alongside information regarding the nature and purpose of the website (Fig 4).

### 4.2.2 Plant Database Page

The plant database provides access to the entire database using the search bar to search plants by name. The search results will be displayed as a list. At the side there is a sorting bar which contains various drop down menus of search filters to refine the search if required. The user can then click a record which will be linked to the plants details (4.7). The Add Plant button will link the user to the Add record page (Fig 5).

### 4.2.3 Add Record Page

This page allows the user to add a new record to the database. The user will be asked to provide information such as scientific name, pronunciation, common name, origin and location alongside an image that can be uploaded to the database upon clicking the save button (Fig 6).

### 4.2.4 Plant Details Overlay

The specific plant page allows the user to access data about individual plants. Above the picture there are links to edit and remove the records giving the user the ability to update the data about the plant. A Where to find button links to a map of the location based from GPS coordinates (Fig 7).

### 4.2.5 Plant Location Overlay

Upon clicking the Where to Find button, a map will open displaying the location of the selected plant using the location from the database. Note: Use of Google Maps (Fig 8).

### 4.2.6 Remove Plant Record

If the user chooses to remove a record they can do so by pressing the Remove button where they will be prompted to verify their choice before the entry is deleted (Fig 9).

## 4.3 Mockups

The image shows two side-by-side screenshots of the Botany App interface. The left screenshot displays a yellow button labeled "Start New Site Visit". The right screenshot shows a "Basic Information" screen with fields for Name, Phone Number, Email, and Site Location, followed by a "Next" button.

Figure 1: An example new site visit form

The image shows two side-by-side screenshots of the Botany App interface. The left screenshot shows a "Add a Species" screen with fields for Species (with a dropdown menu "Please Select" and an "If other please specify" input), GPS Location (with a "Add GPS Location" button), Abundance (with a dropdown menu "Please Select" and two buttons for "Add a scene Picture" and "Add a Specimen Picture"), and Notes (with a text input). The right screenshot shows a similar "Add a Species" screen with a dropdown menu listing species like Cy, Cydamen, Cydamen hederifolium, Cynodon dactylon, and Cynoglossum germanicum, along with buttons for "Add a scene Picture" and "Add a Specimen Picture", a "Notes" input, and a "Confirm" button.

Figure 2: An example new species form

The image shows a single screenshot of the Botany App interface. It displays a "Edit and save" screen with three buttons: "Edit Species", "Delete Visit", and "Save Visit".

Figure 3: An example site/visit confirmation screen

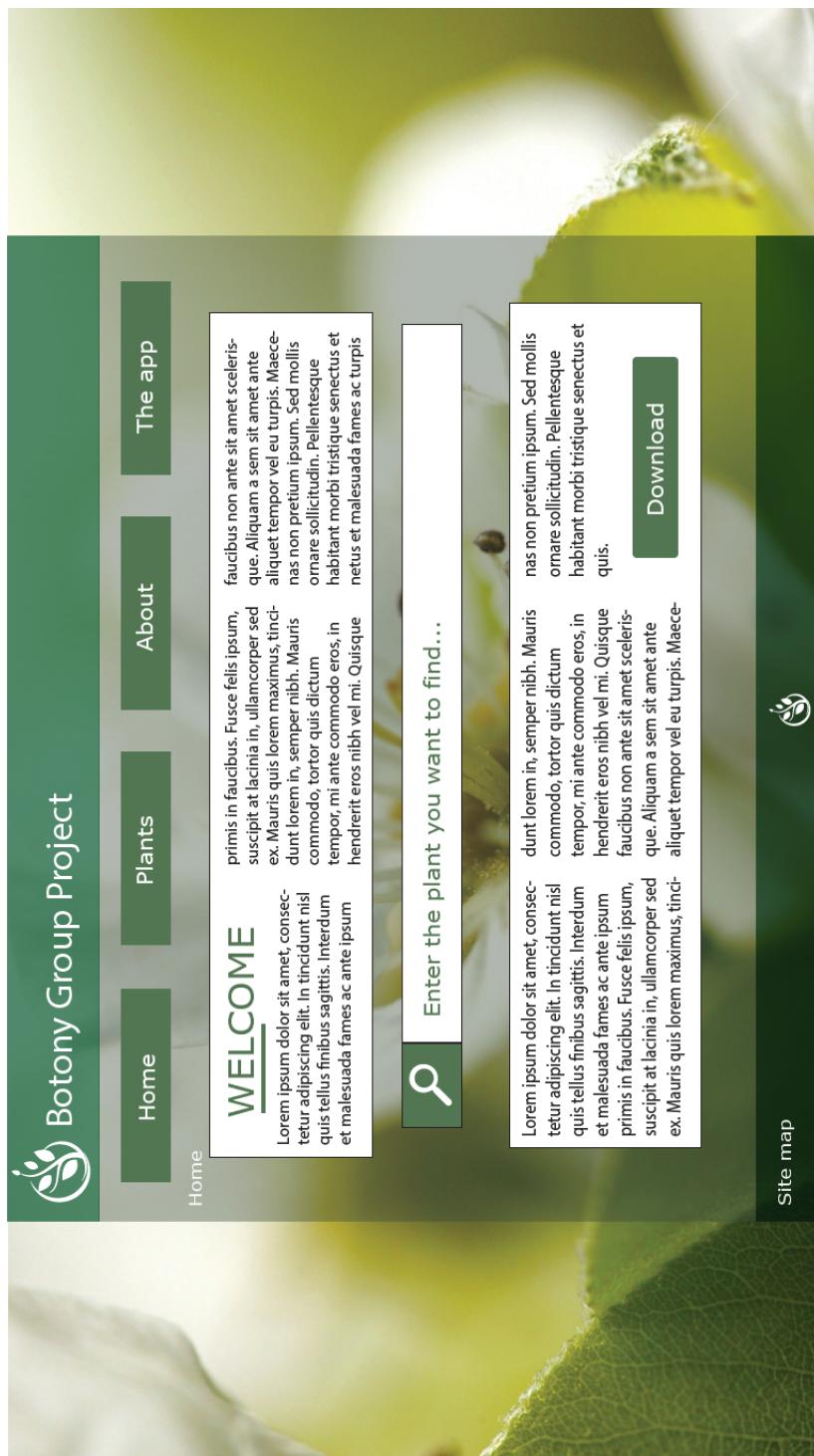


Figure 4: index.php

The screenshot shows the 'Plant database' search interface. At the top, there's a navigation bar with links for Home, Plants, About, and The app. Below the navigation is a search bar with a magnifying glass icon containing the placeholder text 'Enter the plant you want to find...'. To the right of the search bar is a button labeled 'Add Plant' with a green background and white text. Further right is a button labeled 'Words or phrases' with a downward arrow icon. Below the search bar, there's a table with two columns: 'Scientific name' and 'Common name'. The 'Common name' column lists 16 entries, all of which are 'Plant'. At the bottom of the search interface, there are sections for 'Sorting' (with dropdown menus for 'Plant type', 'Flowering time', 'Flower color', 'Fruit color', 'Landscape use', and 'Hardiness Zone'), 'Location' (with dropdown menus for 'All' and 'Site map'), and a footer with a small plant icon.

Figure 5: An example of the plant database

The image shows a detailed view of a green leaf with prominent venation, which serves as the background for the website's interface.

**Botony Group Project**

Home > Plant database > Add record

[Home](#) [Plants](#) [About](#) [The app](#)

Add record

Scientific Name:  
Pronunciation:  
Common Name:  
Family Name:  
Plant Type:  
Habit:  
Form:  
Texture:  
Mature Size:  
Height:  
Spread:  
Origin:  
Hardiness Zone:  
Exposure:  
Soil:  
Landscape Uses:  
Location:

**Upload image**

**Save** **Cancel**

[Site map](#)

Figure 6: An example plant entry in the site

The screenshot shows a web application interface for managing plant records. At the top, there's a navigation bar with a logo, the text "Botony Group Project", and links for "Home", "Plants", "About", and "The app". Below the navigation, a breadcrumb trail reads "Home > Plant database > Edit record". The main content area is titled "Edit record" and contains a form for a plant entry. The form includes fields for Scientific Name ("Zantedeschia aethiopica"), Pronunciation ("Zan-tuh-DESKee-un-huh-thee-OH-pee-uh-kuh"), Common Name ("Calla lily"), Family Name ("Araceae"), Plant Type ("Herbaceous perennial"), Habit ("Upright"), Form ("Vase"), Texture ("Medium - coarse"), Mature Size ("0.7 - 1.3m"), Height ("0.3 - 0.6m"), Spread ("Fast"), Origin ("South Africa"), Hardiness Zone ("Zone 8a (-12 to -9.5°C)"), Exposure ("Full sun"), Soil ("Cultivar or foliage, Perennial border"), Landscape Uses ("Google Images"), and Location (""). There's also a placeholder image for a plant photo, an "Upload image" button, and "Save" and "Cancel" buttons. A "Site map" link is visible at the bottom left.

Figure 7: An example of a plant entry in edit mode

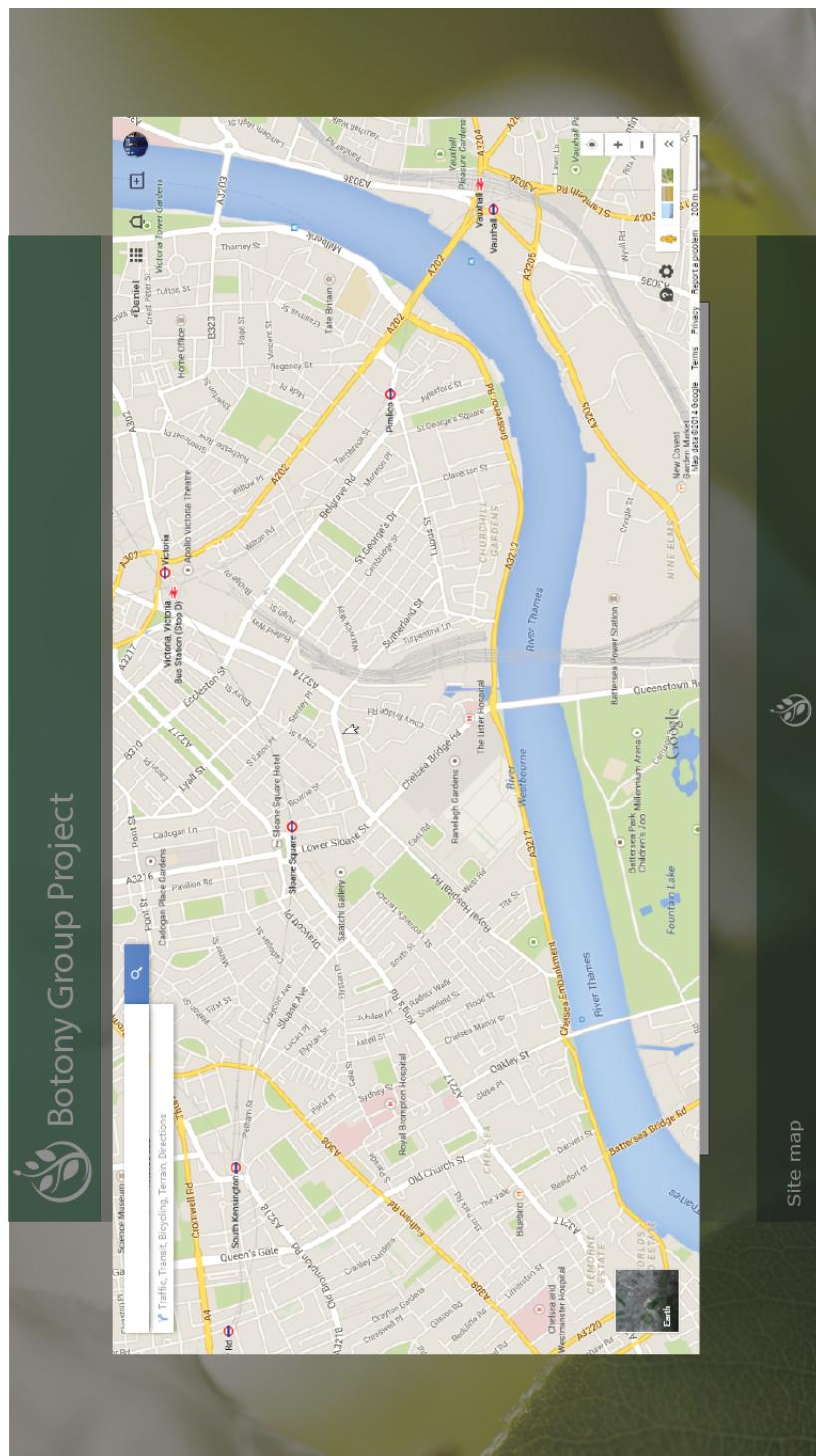


Figure 8: An example map to display recording locations

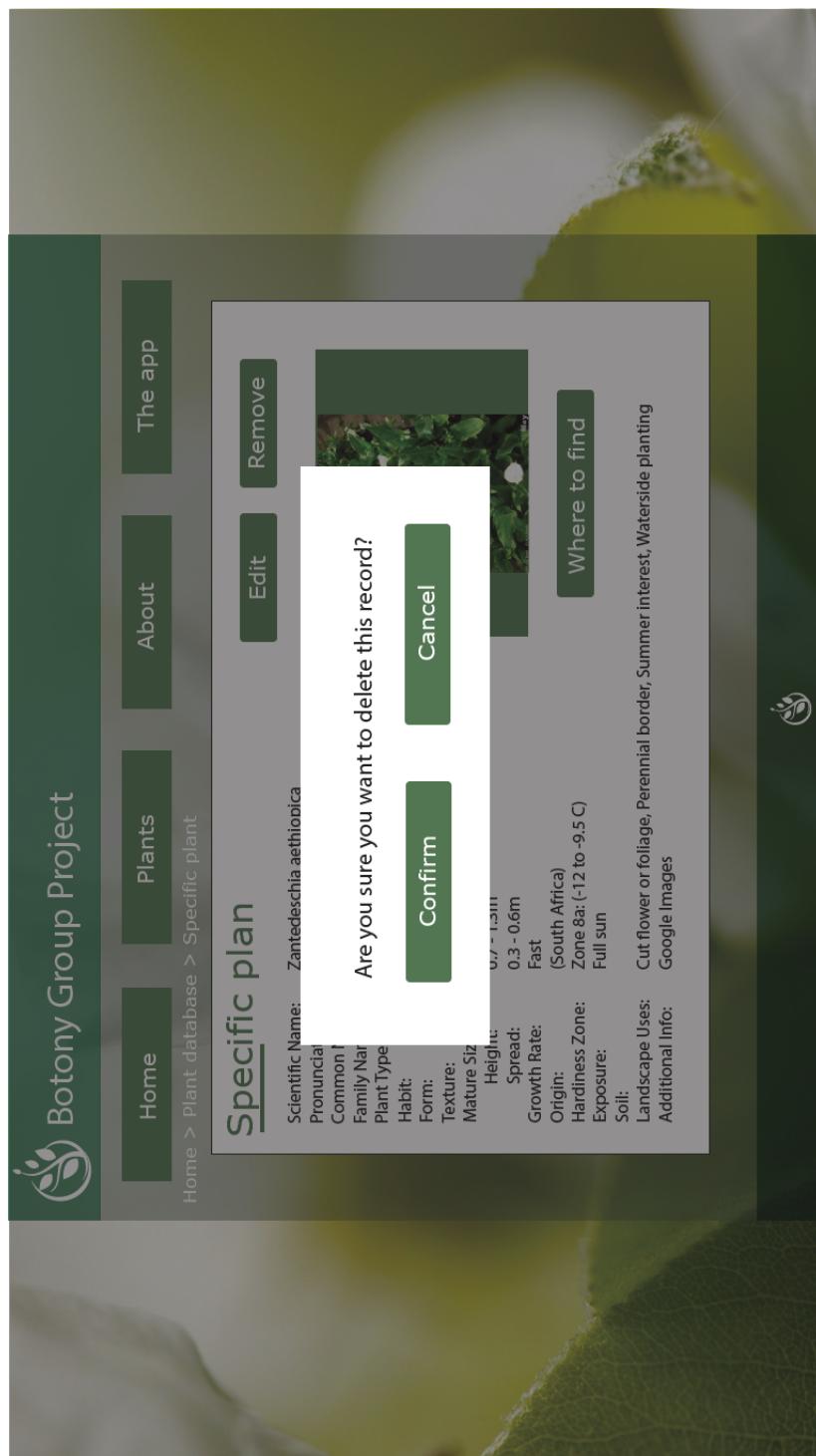
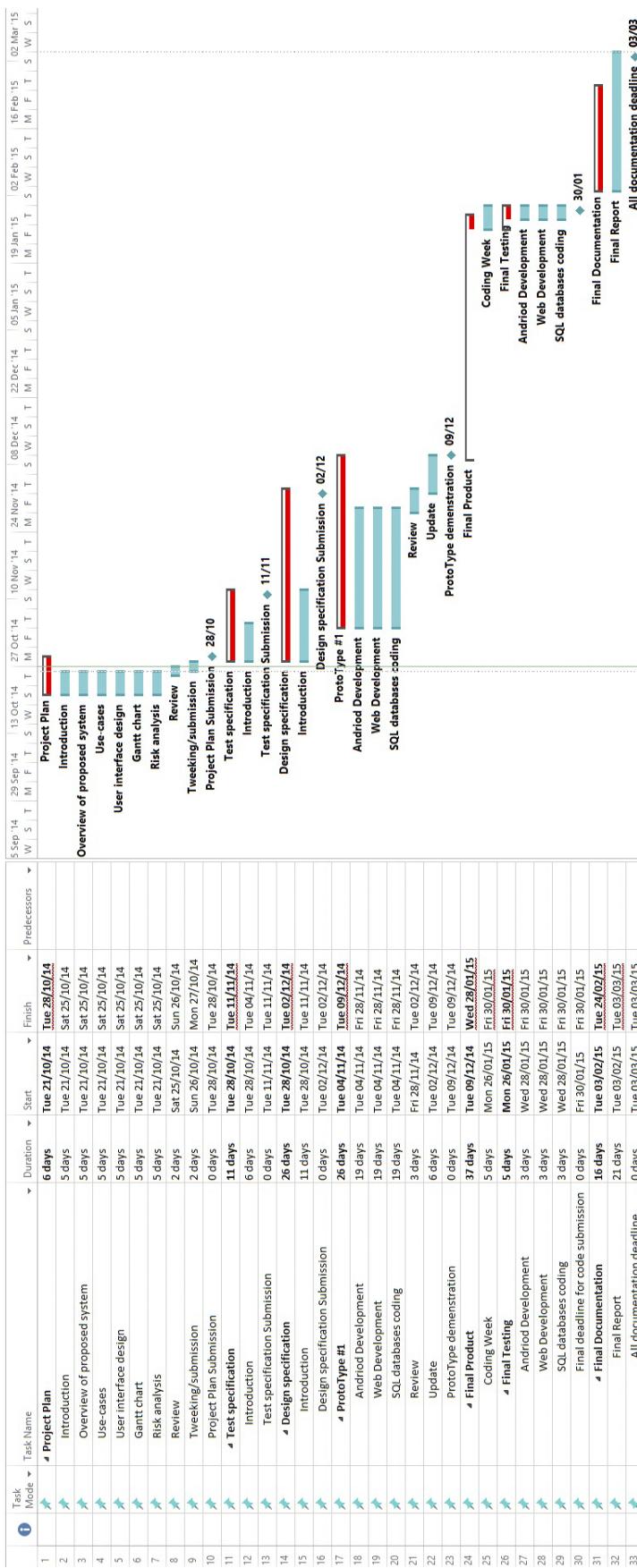


Figure 9: An example delete confirmation dialog

## 5 Gantt Chart



## 6 Risk Analysis and Mitigation

### 6.1 Ongoing Risk Assessments

| Ongoing                          | Likelihood | Magnitude | Risk | Mitigation  |
|----------------------------------|------------|-----------|------|---|
| Planned Team Member Absence      | 2          | 2         | 4    | Meetings scheduled in advance with plenty of time for members to state if there is a problem attending. Missing member required to read minutes and report to relevant manager. |
| Planned Project Leader Absence   | 2          | 3         | 5    | Meeting to be chaired by Deputy Project Leader instead.   |
| Planned QA Manager Absence       | 2          | 3         | 5    | QA Questions and Decisions to be made by Deputy QA Manager instead.   |
| Unplanned Team Member Absence    | 3          | 2         | 5    | Missing member required to read minutes and report to manager. Persistent lack of attendance to result in being carded.   |
| Unplanned Project Leader Absence | 3          | 3         | 9    | Persistent lack of attendance to result in Deputy taking over   |
| Unplanned QA Manager Absence     | 3          | 3         | 9    | Persistent lack of attendance to result in Deputy taking over   |
| Github Downtime                  | 1          | 2         | 2    | Local copies to be used.  |
| Github Failure                   | 1          | 3         | 3    | Use backup to host locally.   |
| Long term illness                | 1          | 3         | 3    | Member to report to relevant manager in advance   |
| Member dropout                   | 1          | 4         | 4    | Member to report to relevant manager in advance   |
| Unknown Technologies             | 4          | 3         | 3    | Prototypes and experimentation to be tried before committing to using unknown technologies  |

## 6.2 Documentation

| Documentation   | Likelihood | Magnitude | Risk | Mitigation  |
|---|------------|-----------|------|---|
| Individual parts of document submitted late             | 2          | 3         | 5    | Internal Deadline set as Friday before the tutorial meeting. Team Members to report to relevant manager as soon as an issue with making the deadline is apparent. |
| Individual parts of document submitted with low quality | 2          | 3         | 5    | Documents to be read by entire team   |
| Human Errors  | 3          | 2         | 5    | Documents should be reviewed by relevant managers and the QA manager  |

## 6.3 Software Development And Deliverables

| Development and Delivery                         | Likelihood | Magnitude | Risk | Mitigation   |
|--|------------|-----------|------|--|
| Slipping from Project Time line                  | 2          | 3         | 5    | Detailed gant charts and time predictions should be kept and viewable by all   |
| Missing or incomplete parts of implementation    | 2          | 4         | 5    | Project Leader to make sure tasks exist for all objectives. QA Manager to ensure work is progressing satisfactory. Extensive testing to make sure all features work as expected. |
| Feature Creep                                    | 2          | 2         | 4    | Teams to maintain communication with managers. QA Manager to maintain adherence to objectives.   |
| Implementation not working as expected by client | 2          | 3         | 5    | Project Leader to practice strong expectation management.  |
| Client requirements change                       | 2          | 2         | 4    | Keep regular contact with the client   |
| Member dropout during Work Week                  | 1          | 4         | 4    | Longer hours will have to be worked by the rest of the team. If possible, Team Member to walk the person picking up their work through their work.                               |
| Central Server down                              | 2          | 2         | 22   | Transfer more work to the git repository and work on local machines. Temporary Databases can be set up if needed.  |

## 7 References

- Requirements Specification : Provided by the client
- Quality Assurance Plan : C.J. Price, B.P.Tiddeman, N.W. Hardy (2014-09-23) SE.QA.01 Version 1.9
- Project Management Standards : C.J. Price (2014-09-23) SE.QA.02 Version 1.9
- General Documentation Standards : C. J. Price, N.W.Hardy & B.P.Tiddeman (2014-09-23) SE.QA.03 1.7
- Design Specification Standards : C. J. Price, N.W.Hardy & B.P.Tiddeman (2014-09-29) SE.QA.05A 1.7
- Project Plan Specification Standards :B. P. Tiddeman (2014-09-23) SE.QA.05b 1.2
- Test Procedure Standards : C. J. Price, N.W.Hardy and B.P.Tiddeman (2014-09-23) SE.QA.06 1.8
- Review Standards : C. J. Price, N.W.Hardy and B.P.Tiddeman (2014-09-26) SE.QA.07 1.6
- Operating Procedures and Configuration Management Standards : C. J. Price , N.W. Hardy & B.P.Tiddeman (2014-09-23) SE.QA.08 1.8
- Java Coding Standards : C. J. Price, A. McManus (2013-09-29) SE.QA.09 1.7
- Producing a Final Report : C. J. Price, N.W. Hardy and B.P.Tiddeman (2013-02-11) SE.QA.11 1.7

## 8 Document History

| Version | Edit                                       | Date             | Persons |
|---------|--|------------------|---------|
| 0.1     | Initial Version                            | October 21 2014  | jsm14   |
| 0.2     | Document First Draft                       | October 27 2014  | jsm14   |
| 0.3     | Client first release                       | October 28 2014  | jsm14   |
| 0.4     | Spellchecking and Document standardization | October 29 2014  | nid21   |
| 0.5     | Document review                            | November 28 2014 | nid21   |
| 0.6     | Further document review                    | January 22 2015  | anw46   |