

# BIOMAP

## PROTOTYPE REPORT

GRAEME FAUL<sup>1</sup> & GREGORY LINKLATER<sup>1</sup>

May 16, 2016

### CONTENTS

1	Application Description	2
1.1	Assumptions	2
1.2	Target Audience	2
1.3	Context	2
1.4	Required Features	2
2	Feedback	3
2.1	False Assumptions	3
2.2	Changes to the System	3
A	Wireframes	5
A.1	Login	5
A.2	Home	5
A.3	Settings	6
A.4	Help	7
A.5	Record	8
B	UML Diagrams	9
B.1	Use Case Diagram	9
B.2	Activity Diagrams	10

### LIST OF FIGURES

Figure 1	Login Screen Wireframe	5
Figure 2	Home Screen Wireframe	5
Figure 3	Add Database Shortcut Screen Wireframe	6
Figure 4	Settings Screen Wireframe	6
Figure 5	User Help List and Detail Wireframes	7
Figure 6	Record List and Detail Wireframes	8
Figure 7	Use Case Diagram	9
Figure 8	Login Activity Diagram	10
Figure 9	Add/Edit Record Activity Diagram	10
Figure 10	Hamburger Activity Diagram	11

---

<sup>1</sup> Department of Computer Science, Rhodes University, Grahamstown, South Africa

## 1 APPLICATION DESCRIPTION

### 1.1 Assumptions

- Users will be using an Android device with the Jellybean operating system or newer.
- Users will have an existing account on the Animal Demography Unit (ADU).
- Users have at minimum a basic understanding of how to operate a smart phone.
- The system will utilise an API defined by the ADU to interact with the existing system.

### 1.2 Target Audience

Existing members of the ADU who have access to an Android smart phone.

### 1.3 Context

The ADU is an online database of fauna and flora sightings from around the world, but predominantly South Africa. The idea behind the database is to maintain digital records rather than physical ones, thus preventing the need of killing the specimen for archiving. The current system requires manual entry of data including location, date, specimen details and manual upload of media. This is an extremely slow and inefficient way of capturing new entries, and is prone to human error. This is where the idea of having an Android application which can easily generate records by auto completing much of the data required, and simplifying the capture of other data such as images and sounds.

### 1.4 Required Features

The application should be able to:

- Allow the user to authenticate into the application with the same credentials as their ADU account.
- Allow the user to create new record for any one of the ADU databases, the records should hold the exact same information as they would if they were created on the ADU.
- Automatically complete data where possible, such as location and date.
- Allow the user to override any auto completed data.
- Allow the user to submit completed record to the ADU for verification.
- Allow the user to edit/delete any non-submitted records and view any previously submitted records.
- Allow the user the option to only upload records when on wi-fi.

## 2 FEEDBACK

### 2.1 False Assumptions

After a conversation with *Prof. Craig Peter* it became clear that our initial assumptions of how the system should operate were completely incorrect; in particular the following assumptions that we were operating under:

- The backend of the system would be completely separate to the existing system and would need to be built separately. This assumption was based on the fact that it was highly unlikely that the system administrators of the ADU would allow us (students) to interact and effect possibly breaking changes to a production system. Since the initial prototype was created, it seems that there will be a development system created that will be functionally equivalent to the production system for which we will be given API documentation and the relevant credentials to access. As of the time of writing, we have not yet been given such documentation which may or may not affect how the final product will operate when compared to this prototype document.
- We would be able to add functionality to the existing system. This stems from the above point where we would be creating our own backend and could therefore define additional features. Ideas such as live communication with moderators regarding submissions, improved search functionality by indexing records through the use of “tags”, the use of notifications and correcting database schema errors such as the restriction of three images to a submission are all therefore useless for the design of this application due to lack of access to the backend system. According to *Prof. Peter* The restriction of the number of images per submission often leads to users submitting multiple records to cover all of the images the user has captured and the existing system has no way to link them. It is impossible to tell without more information, however this error is most likely an error in the database schema that has not been designed to conform to the 3<sup>rd</sup> normal form.
- The document containing a list of requests from *Prof. Peter* showed additional features that were not supported by the existing system. Due to us operating under the above assumptions; this further reinforced that we would be able to implement these additional features when creating our own backend. Now that we know differently; it is clear that some of the requirements that were requested will be impossible to implement without modification of the existing system and we will therefore not be able to meet them.

The above false assumptions have therefore since required us to completely redesign the system which is what is reflected in this document.

### 2.2 Changes to the System

From what we were able to salvage from our existing prototype, *Prof. Peter* had the following comments:

- The use of icons to select the database on the home screen (Figure 2 on page 5) seems to be a different approach; he seemed unsure of how

it would work and stipulated that it would be difficult to use if icons for all of the databases were present on the page, as most of the users seem to restrict themselves to a handful of complimentary databases. We have therefore decided to keep the icon-based home screen – as this decreases the amount of manual text entry on the part of the user when submitting a record – and instead have added a process for adding icons to the home screen from the list of existing databases. Due to the fact that we have not yet recieved API documentation the list of databases; the list (Figure 3 on page 6) will be built statically.

- The addition of the “My Records” screen (Figure 6a on page 8) received positive feedback as it seems that it is fairly difficult, when batch uploading a large amount images, to remember where exactly one is in the process. Initially the plan was have one screen that contained all the records submitted for all the databases the user contributes to, and records were distinguished by name, date and an icon showing which database the submission was contributing to. In order to make it easier for the user to remember; we changed this to be a separate records list per database and changed the icon on the left of the record heading to show the first image of that submission. Once again, due to the fact that we do not have any API documentation for the existing system as of yet, this list will not be definitive as the data it contains will be persisted locally on the device and will only contain the submissions that have been made from that particular device. There are no plans to add a “backup” feature for this.
- Regarding the editing and submission of records, we suggested the use of Google’s “Reverse Geocoding” Service<sup>1</sup> which would allow us to gain location data from GPS coordinates. Our suggestion received positive feedback however the point was made that it is absolutely necessary for the user to confirm the details from this process. As such; we added fields to manually edit the closest town, province and country should the user wish to.

Some of these changes have been reflected in this document and others have not due to time restrictions. The changes that are not reflected in this document will hopefully be present in the final product if there is sufficient time to include them.

---

<sup>1</sup> <https://developers.google.com/maps/documentation/javascript/examples/geocoding-reverse>

## A WIREFRAMES

### A.1 Login

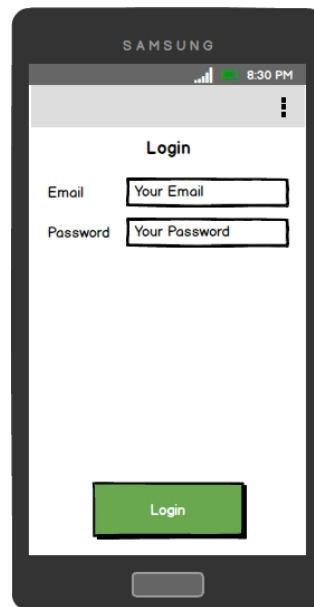


Figure 1: Login Screen Wireframe

### A.2 Home

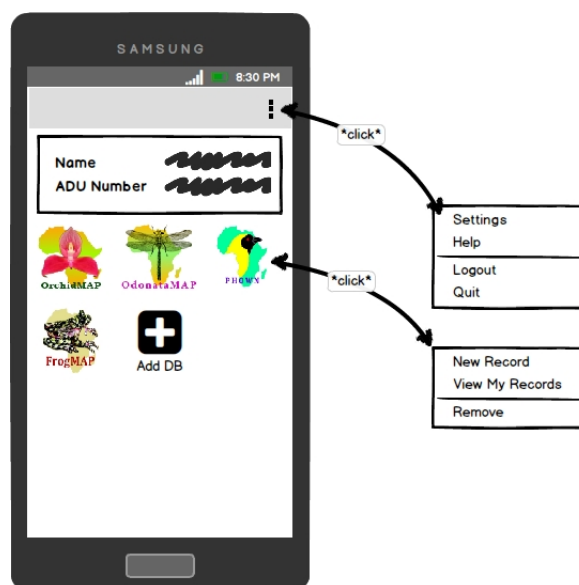


Figure 2: Home Screen Wireframe



Figure 3: Add Database Shortcut Screen Wireframe

### A.3 Settings

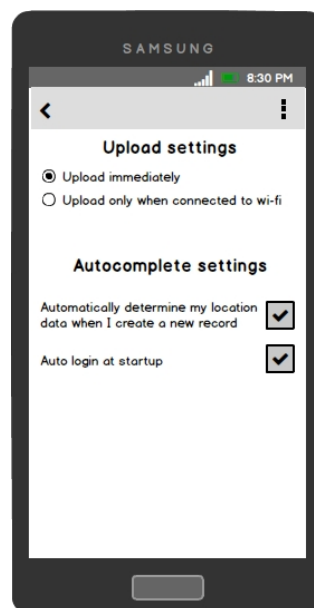
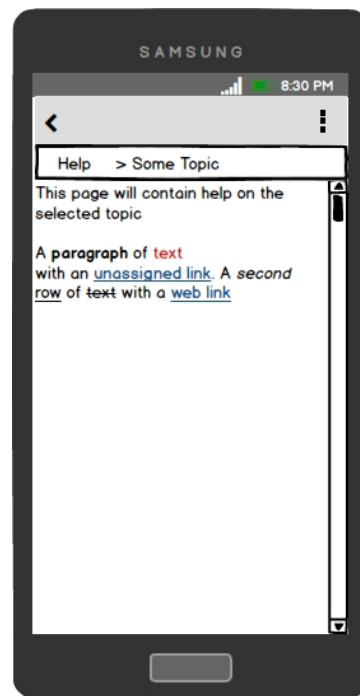


Figure 4: Settings Screen Wireframe

## A.4 Help



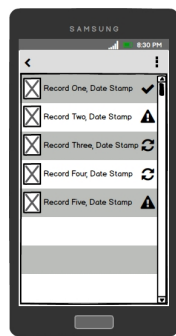
(a) Select Help Topic Screen



(b) View Help Topic Screen

Figure 5: User Help List and Detail Wireframes

## A.5 Record



The icon on the left shows the first picture of the submission.

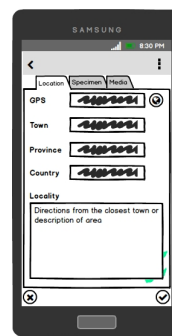
The icon on the right shows the current status of the record.

A check means that the record has been submitted to the ADJ.

An exclamation mark means the record is incomplete.

A refresh symbol means the record is complete and pending submission.

(a) Record List Screen Wireframe

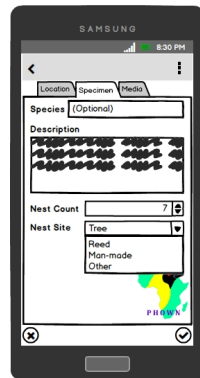


Pressing the globe button will open the google maps app and allow the user to select a location.

On first start up, the app will request that user's enable automatic geotagging of photos taken with the camera app.

The Town, Province and Country fields will be auto-populated (but editable) by google's reverse geocoding service.

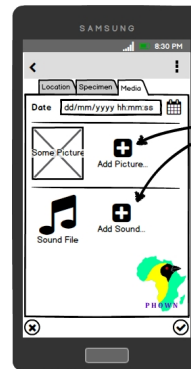
(b) Record Location Tab Wireframe



The nest related elements will only be visible for use with the PHDOWN database.

Other database specific fields will be visible here depending on which database is selected.

(c) Record Specimen Tab Wireframe



Clicking one of the buttons to add media will allow the user to capture new media or add existing media.

(d) Record Media Tab Wireframe

Figure 6: Record List and Detail Wireframes



## B UML DIAGRAMS

### B.1 Use Case Diagram

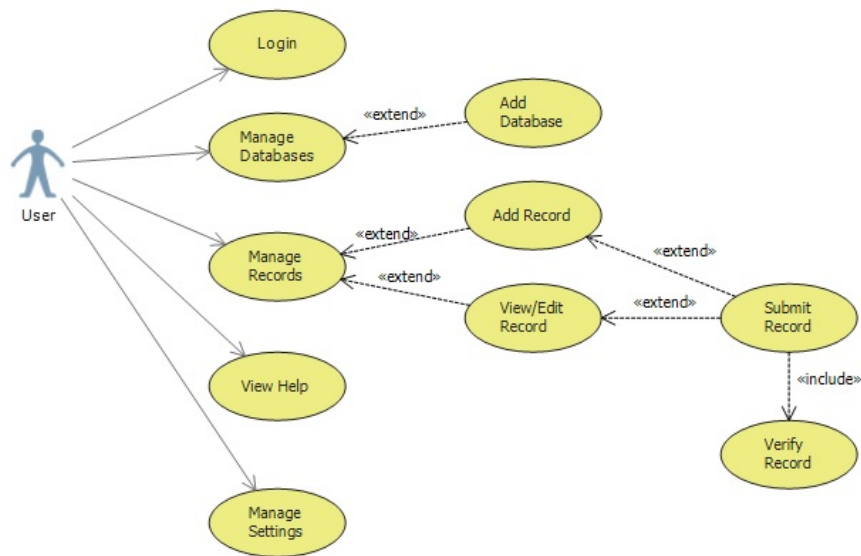


Figure 7: Use Case Diagram

## B.2 Activity Diagrams

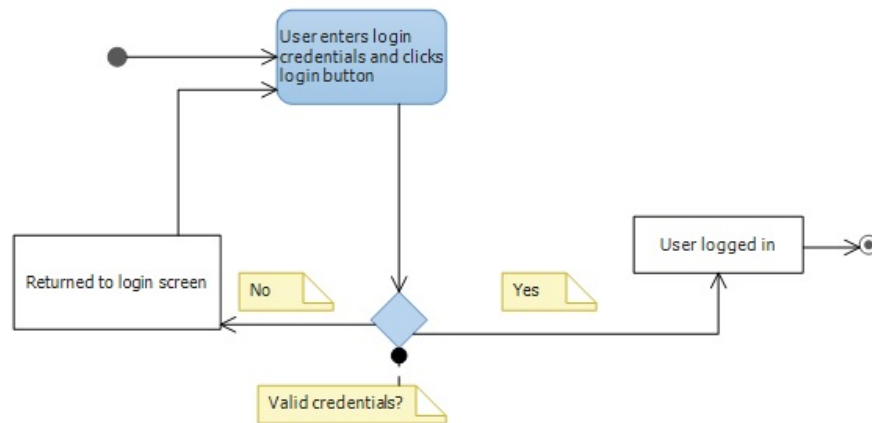


Figure 8: Login Activity Diagram

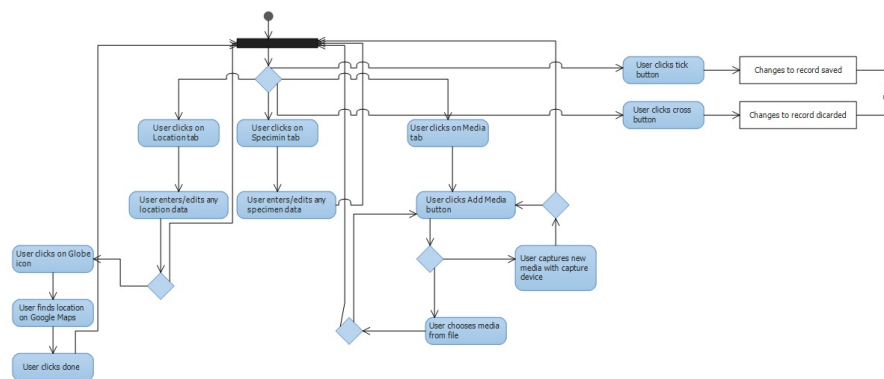


Figure 9: Add/Edit Record Activity Diagram

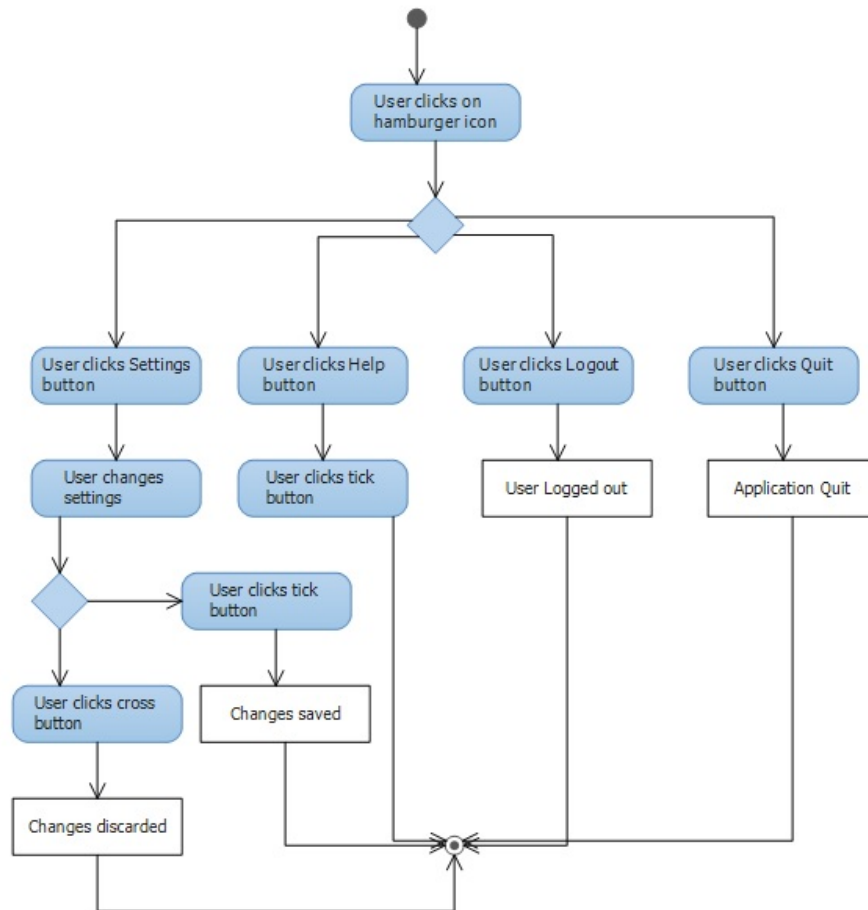


Figure 10: Hamburger Activity Diagram