

# Week 4 Report

Jessica Braddon-Parsons  
Cameron Dykstra  
Umesh Ravji  
James Rumbal

# Table of Contents (Click to follow links to sections)

## Process

### Documenting Requirements

- Scope

- Definitions

- Requirements

- Priority

- Non Functional Requirements

- Support Time

- Use Cases

### Project Infrastructure

### Life Cycle Model and Project Plan

- Gantt Chart

- Primary Focus of Group Members

### Technology selection

### Architecture / high level design

- Component Diagram

- Database Design

### Risk Management

## Quality Assurance

- Testing

- Issue Tracking Policy

## Product

# Process

## Documenting Requirements

### Scope

The app we are developing during this project will run on Apple and Android mobile devices and will allow users to more easily locate traps for pest capture, and quickly report any species caught or damage found. It will also allow users to upload an image with this data if they wish. When their device has a network connection the created data will be synchronised with the server so all the information for a trap line is kept in one place. This database of information will be accessible from a website where users are able to export line data to CSV files. Users with administrative powers will also be able to add new lines, add, remove, and edit traps, and otherwise manage the data. There is currently no planned maintenance beyond the end of the semester, and the website will not provide visual representations of the data (such as graphs). This scope is subject to change as the semester progresses and we are able to talk to the customer and better understand their needs and requirements.

### Definitions:

**Priority:** 1 is highest, 4 is lowest - In lieu of meeting with the client, priorities are based roughly on creating a minimal viable prototype and order of prerequisites.

**Users:** Anyone who will use the app

**Authorised users:** Users with an admin login where they are an admin for the relevant line

**Input data:** Information a user has entered while walking a line (such as what was found at each trap, the geolocation of a new trap...)

Requirements	Priority
The system should allow authorised users to add new lines	1
The system should allow authorised users to add a new trap to an existing line	1
The system should allow authorised users to configure existing lines	3
The system should provide password protection for lines	3
The system should indicate how near users are to a trap	2
The system should capture trap related input data including maintenance carried out and details of species caught	1
The system should allow input data to be stored locally on the user's device until they are connected to a network	1
The system should share the input data from the user's device to the database when there is network connectivity	1
The system should allow the user to include photos in the input data	4
The system should allow users to export data relating to trap and line location, trap condition, species found in traps, and other collected information, to CSV files	3
The system should provide a web interface allowing users to download data in CSV files	4
The system should cache map data on the user's device so network connectivity is not required while walking the line	2
The system will include a database to store data relating to trap and line location, trap condition, species found in traps, and other collected information	1

## Non Functional Requirements

- The app will run on iOS and Android and is backwards compatible (e.g., works on older versions of these OS such as iOS7 and Android Kitkat)

## Support Time

For information on when different features will be implemented, please see the project plan further down, where requirements tiers refer to the assigned priority of the requirement.

## Use Cases

ID	<b>UC-1</b>
Name	<b>Add new trap line</b>
Description	Authorised user adds a new trap line to the system
Primary Actor	User
Preconditions	
Postconditions	There is a new line with no traps, and the user is entered into that line
Main Success Flow	<ol style="list-style-type: none"><li>1. User chooses to create a new line</li><li>2. System prompts user for admin login</li><li>3. User enters admin login credentials</li><li>4. System displays prompts for trap line details</li><li>5. User enters for trap line information</li><li>6. System displays "new line created message"</li><li>7. System enters active mode for selected line</li></ol>
Sub-Flows / Extensions	<ol style="list-style-type: none"><li>5.a. Add password to line<ol style="list-style-type: none"><li>5.a.1. User ticks "password protect" box</li><li>5.a.2. System displays additional inputs for password and password confirmation</li></ol></li></ol>
Alternate / Exceptional Flows	<ol style="list-style-type: none"><li>3.a. Incorrect credentials<ol style="list-style-type: none"><li>3.a.1. System prompts user to re-enter credentials or cancel</li></ol></li></ol>

ID	UC-2
Name	Select existing line
Description	User selects an existing line to use
Primary Actor	User
Preconditions	User has password for the line (if it requires a password)
Postconditions	The user is entered into the existing line
Main Success Flow	<ol style="list-style-type: none"> <li>1. System displays trap line selection screen</li> <li>2. User searches or scrolls through available trap lines and selects a trap line</li> <li>3. System prompts user for trap line password if required</li> <li>4. User enters trap line password</li> <li>5. System prompts user for which route they want to take</li> <li>6. User specifies route</li> <li>7. System enters active mode for selected route</li> </ol>
Sub-Flows / Extensions	
Alternate / Exceptional Flows	<ol style="list-style-type: none"> <li>4.a. Incorrect password <ol style="list-style-type: none"> <li>4.a.1. System prompts user to re-enter password or cancel</li> </ol> </li> </ol>

ID	UC-3
Name	Add trap
Description	User adds a new trap to an existing line
Primary Actor	User
Preconditions	
Postconditions	The selected line has a new trap
Main Success Flow	<ol style="list-style-type: none"> <li>1. User selects to add a new trap</li> <li>2. System prompts user for admin login</li> <li>3. User enters admin login credentials</li> <li>4. System displays trap line selection screen</li> <li>5. User searches or scrolls through available trap lines and selects a trap line</li> <li>6. System displays prompt for entering new trap information</li> <li>7. User enters trap information</li> <li>8. System saves trap information</li> <li>9. System displays active mode for selected route</li> </ol>
Sub-Flows / Extensions	
Alternate / Exceptional Flows	<ol style="list-style-type: none"> <li>3.a. Incorrect credentials <ol style="list-style-type: none"> <li>3.a.1. System prompts user to re-enter credentials or cancel</li> </ol> </li> </ol>

ID	UC-4
Name	Alert that user is near trap
Description	The app alerts the user that they are near a trap
Primary Actor	App
Preconditions	The user has selected a line
Postconditions	
Main Success Flow	<ol style="list-style-type: none"> <li>1. System detects user's location</li> <li>2. System creates an audio and/or visual alert that the trap is near</li> <li>3. User acknowledges alert</li> <li>4. System provides audio and/or visual direction towards the trap</li> <li>5. User indicates they have found the trap</li> </ol>
Sub-Flows / Extensions	<ol style="list-style-type: none"> <li>4.a. User walks past trap               <ol style="list-style-type: none"> <li>4.a.1. System creates audio/visual alert that the user has walked too far</li> </ol> </li> </ol>
Alternate / Exceptional Flows	



ID	UC-5
Name	Add data for trap
Description	The user adds trap related data into the system
Primary Actor	User
Preconditions	User has selected an existing trap line and has been alerted to the placement of the trap
Postconditions	
Main Success Flow	<ol style="list-style-type: none"> <li>1. User enters that they have found the trap</li> <li>2. User enters the species found in the trap</li> <li>3. User enters that they are finished with the trap</li> <li>3. System displays location information for the next trap</li> </ol>
Sub-Flows / Extensions	<ol style="list-style-type: none"> <li>2.a. User chooses to add image <ol style="list-style-type: none"> <li>2.a.1. System displays camera</li> <li>2.a.2. User takes photo</li> <li>2.a.3. Photo is attached to data files</li> </ol> </li> <li>2.b. User chooses to add maintenance information <ol style="list-style-type: none"> <li>2.b.1. System displays maintenance options</li> <li>2.b.2. User selects appropriate option(s)</li> <li>2.b.3. System returns user to previous screen</li> </ol> </li> </ol>
Alternate / Exceptional Flows	<ol style="list-style-type: none"> <li>1.a. User cannot find trap <ol style="list-style-type: none"> <li>1.a.1. User enters that the trap cannot be found</li> <li>1.a.2. System displays location information for the next trap</li> </ol> </li> </ol>

ID	UC-6
Name	Send data to database
Description	Send collected data to the database
Primary Actor	System
Preconditions	The user has completed the line and the data they entered is stored locally on their device
Postconditions	The data that was stored on the device is in the database of information collected
Main Success Flow	<ol style="list-style-type: none"> <li>1. The app detects network connectivity (eg wifi or cellular data)</li> <li>2. The system compresses and encodes the data</li> <li>3. The system sends the data over the network to the database</li> <li>4. The data is decompressed and added to the database</li> </ol>
Sub-Flows / Extensions	
Alternate / Exceptional Flows	

## Project Infrastructure

We have decided to use:

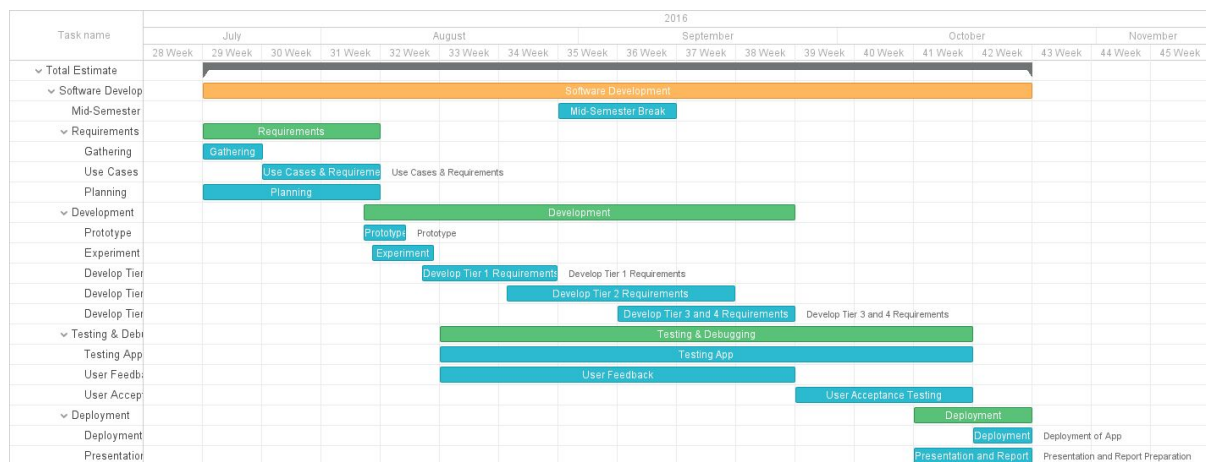
- Github - repository and issue tracking  
We considered both GitHub and BitBucket. Both offered private repositories (once we received a student licence for GitHub), but GitHub has more comprehensive tools such as tracking who edited which files and when, so it is easier to keep track of which group member did what work.
- Trello - workflow management  
Trello was selected as it provides a flexible and easily customisable Kanban board for use with our agile workflow management.
- Slack - communication  
Slack has been selected for team communication as it easily allows discussion to be separated by topic in different channels.
- Zapier - Integration of GitHub, Trello, and Slack  
GitHub, Trello, and Slack were also chosen because Zapier could be used to easily integrate them together. Through Zapier, a new card is created in Trello and a new channel in Slack for every new issue in GitHub, with consistent naming between the three tools. This makes it very clear where the task or discussion for that issue should be found. This integration of tools was chosen over JIRA as JIRA wasn't offered as a cloud service on an academic licence, and even if we paid for it, the system was not intuitive to use so would have consumed more time to learn to use.

- Gluon - cross platform mobile app development  
Gluon was chosen because it uses Java, which all group members are familiar with, offers academic licences, and facilitates creating mobile apps which run on iOS and Android which are the platforms specified in the requirements.
- Gradle - automation and deployment  
Gradle is built into Gluon and Gluon is built to work with Gradle, so we are using Gradle to make the app development as simple and smooth as possible, using the tools as they were intended rather than trying to use another automation tool.
- Travis-CI - building  
Travis-CI can be integrated with GitHub and Slack to automate our testing. Whenever a push is made to the repository, Travis-CI will automatically run build and test scripts, and return the results of running the scripts on Slack.
- PythonAnywhere - server  
PythonAnywhere has a free tier for hosting and includes a MySQL database, which are our primary requirements. We also have experience using it in the past.

## Life Cycle Model and Project Plan

We will be using the agile methodology Kanban. We have selected this over other agile methodologies because it is the most flexible, so it allows for our plan changing at any time to suit the requirements and needs of the end users.

### Gantt Chart



See image in full [here](#)

### Primary Focus of Group Members

We have decided to assign a primary focus for each team member, so they will predominantly work on that area, but will also work on other aspects of the project as necessary to ensure we all have a good understanding of the system and to ensure the project can be completed on time.

Jessica and Umesh - App development and testing using Gluon

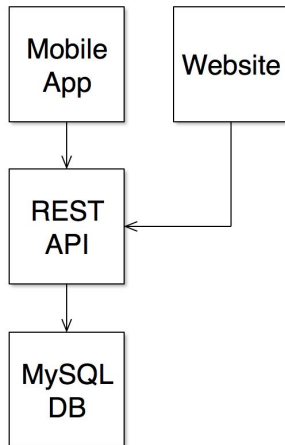
James and Cameron - Server and database development and testing

# Technology selection

We have decided to use:

- MySQL  
The industry standard for data storage and access, and all group members have experience using it.
- Java  
Java has been selected to develop the app because all group members are familiar with the language, and the Gluon tool we are using uses Java to create an app which can be deployed across all mobile platforms.
- Python  
Python has been selected to develop the server because all group members are familiar with it, and the group members who will be focusing on the server side development have some experience doing this in Python. Using Python also allows us to use PythonAnywhere to host the server for free.
- IntelliJ  
All group members are using the IntelliJ IDE as we are familiar with it and it provides plugins for Gluon. There is also support and additional features offered for Gluon when using IntelliJ, and we are able to use the professional version of IntelliJ through student licensing.
- JUnit4  
JUnit4 is a testing library for Java, that will be used to test the functionality of our application.
- JaCoCo  
JaCoCo is a Java code coverage library that will be used to support our tests and tell us what parts of code we have and haven't tested.
- PMD  
PMD can be used as a code sniffer, to make sure that good coding standards are used to make sure that code complexity is reduced to a minimum, and can include extra rules to check against like optimisation rules.
- JavaDocs  
In-built into Gradle, JavaDocs is used to develop the documentation for the application.
- Flask  
Some group members are already familiar with Flask, and it has a library 'flask\_restful' which provides a simple REST API framework.
- unittest  
The standard testing library in Python, will be used for testing the server-side code.

## Architecture / high level design



## Component Diagram

MySQL DB:

- Provides a SQL interface for storing and querying data.

REST API:

- Requires a SQL database backend.
- Provides a REST API for creating/querying lines, traps and catches.

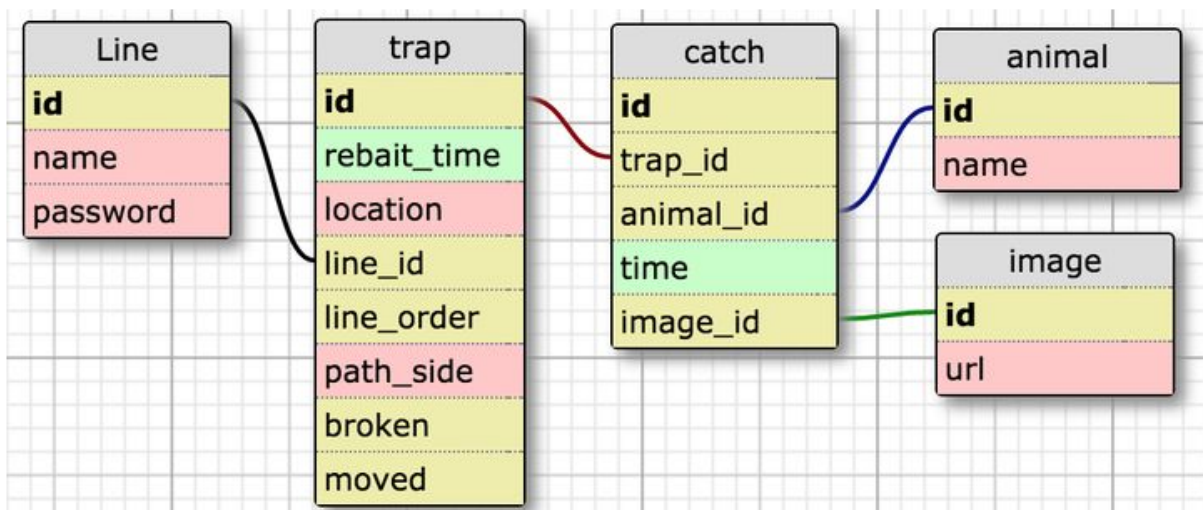
Mobile App:

- Requires API access to a database of lines, traps, and catches.
- Provides end-user functionalities, described in more detail in the 'Requirements' section of this document.

Website:

- Requires API access to a database of lines, traps, and catches.
- Provides end-user functionalities, described in more detail in the 'Requirements' section of this document.

## Database Design



## Risk Management

Risk	Severity	Probability	Strategy
Accurate GPS coordinates not available	High	Medium	Design the app so it gives general directions to traps, and informs users that it may not be accurate. Also attempt to use any other tools (such as built in gyroscope and accelerometer) to replace or aid GPS where needed
Information loss due to computer failure or losing track of information recorded on paper	Very high	Low	Ensure all digital data is regularly saved and backed up (e.g. pushed to repository and pulled to computers), and that important information on paper is replicated digitally
Unable to complete some requirements	Medium	Low	Requirements are prioritised so that high priority requirements are implemented first. This means that if we have hold ups and get behind schedule, the requirements that aren't completed will be the low priority ones, which are less important to the app and end users.
A group member is unable to complete work for some period of time due to sickness, injury, or other absence	Medium	Medium	All work will be divided between group members, with more than one person working on each aspect of the system or at least knowledgeable about each module. This makes it is easy for members to work on items that can't be completed by the member initially working on them. Also allow for buffer room to complete tasks so the final delivery won't be compromised

## Quality Assurance

### Testing

Automatic testing will be applied through Travis-CI and on the server. This will check for passing unit tests, which will be developed alongside the app and server. We will also complement this with manual tests for aspects such as system integration between the app and the server.

We will attempt to get feedback from end users throughout development to ensure we are meeting their requirements. We will also perform user acceptance testing at the end of development to confirm that we have delivered all of the requirements.

## Issue Tracking Policy

When a new issue is found, a group member will create a new issue in GitHub. Zapier will cause a new card to be created in Trello and a new channel in Slack. All communication related to the issue will be carried out in the Slack channel. The card in Trello will be moved to an appropriate place in the backlog of tasks to do, based on its urgency and importance. If necessary, it will also be broken down into smaller tasks. These tasks will then be picked up and moved to the WIP list like any other task in the Kanban board (in order of backlog, by someone who has the knowledge/skill to complete it), and the GitHub issue will be assigned to the group member(s) working on it. Once the task(s) has been completed and moved to the “Done” list, it will also be marked as closed in GitHub by the last person to certify it is complete.

## Product

Please visit [Marvel App](#) to view our interface prototype. This has been checked against the use cases and matches all that it demonstrates. (Some, such as sending the data to the database, are not demonstrated in the prototype, so have not been checked.)

We are also able to demonstrate a proof of concept prototype where the Google Maps API has been used to display a map in the mobile app.