IMAT3451

Final Year Project Dissertation

Title:

<example title page>

Research & Development of a Butterfly Identification Android Application

Your name

Your P number

Supervised by:

Dr Richard Bates

Date

Contents Index

A contents page or two, main section headings with page numbers

<example table of contents>

Table of Contents

Figures and Tables Index

A contents page or index of the tables and figures

*<Headings and text in **Black** probably should appear in your report>*

<Text in Green and the illustrations are just my waffle to give you suggestions>

<The report is made up of two parts:

- *A main section covering the process - [what did you do, why did you do it, how did you do it]*
- *and the Appendices - [showcasing the product - design diagrams, code snippets, test results, screen shots] >*

<This following is the main section with a 10,000 word limit>

<Written in 3rd person 'The app was developed, code was written...'>

Introduction

- Around 200-600 words, sentences rather than bullet points.
- This project was done because…
- Your motivation...
- A need...
- Imagine somebody is reading your report and knows nothing about your project, give them an idea what it's about

<Simple example>

<Some of this can be from your first report>

Butterfly spotting is a popular hobby enjoyed by many people, and one of the attractions of this hobby is identifying butterfly species. Identification can be logged along with time, date and location. This data is of use to conservationists and other concerned bodies.

Traditionally this logging is done by using a book of photographs and recording the sighting on paper logs, which are then shared by email. This results in a cumbersome and time consuming process.

This butterfly identification and central logging process was an application development opportunity, and the aim of this project was to automate this entire process by producing a butterfly identification smartphone app. The research and development of this butterfly identification app is the subject of this dissertation.

Aims and Objectives

- Around 1-2 pages, more if you need
- The aim of this project was to… It accomplished...
- Look at you Project Contract and Interim Report for ideas
- Imagine somebody is reading your report and knows nothing about your project, go through each key aim or objective of the project

<Example few paragraphs, write much more than this>

<Some of this can be from your first report>

The main aim of the project was to develop a butterfly identification application. The app would automate the identification process by allowing the user to photograph a butterfly, then identify the butterfly by connecting the app to a database of butterfly identification pictures. The app would use an online service to make a best guess attempt to identify the butterfly. The app would then return the result to the user and also share the identification results including location date and time via a shared web application.

There were several main objectives to developing the app. Image acquisition, connecting to an online services, image identification, location determination, and finally results saving and sharing via an online database. Each of these objectives were met, the app passed testing and was released to the Android Marketplace, making over 100 downloads in the first month.

A sample screenshot of the final app is shown below, figure X. Further images appear in the Appendix X.

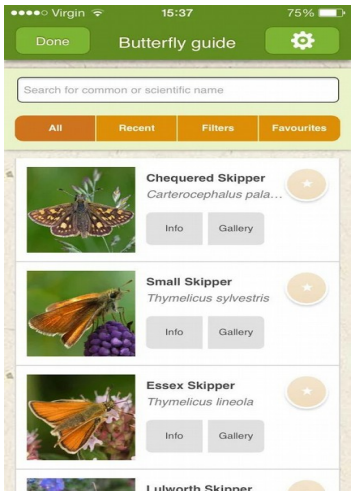

Figure X: Butterfly app

Research

- This is a summary of your literature review
- 8 to 16 pages, more if you need
- Use References if you quote something, and put the in the References section later in the report
- If your had a few summary tables, then use them
- If your had a few survey results tables, then use them
- This research is from your first report literature review, put that in the Appendix, talk about it here
- You can also rewrite your original literature review if you like
- Talk about your findings from your literature review
- Imagine somebody is reading your report and knows nothing about your research, or what you have researched about your project area, give them an idea what you have learnt.

<Example few paragraphs, write much more and use more figures and tables than this>

<add your own headings to suit your project>

<you don't need to use my headings or ideas here, feel free to do what you feel!>

Before developing the app it was clear that the development required research into the various features of the app. A history of butterfly identification techniques was required, covering how manual systems work, and comparing them to the proposed machine image identification techniques proposed for the app.

Comparison of existing machine based image identification systems

A comparison of the reliability or accuracy of identification systems was required. A literature review was conducted on these techniques, see Appendix X. The review found that manual techniques were generally reliable, but dependant on experience, with novices taking some time to identify images and being fairly unreliable in correctly identifying butterflies. In contrast machine based systems offer a reasonably high level of accuracy,

matching that of experience manual users. This indicated that an automated system may achieve the accuracy of an experienced user, and would exceed that of a novice manual user. Thus the development of this app could aid in the identification of butterflies.

Choice of image Identification system

Image identification systems were analysed, Appendix X, and Google systems were chosen due to their close integration with Android. A sharing web app choices were surveyed and Firebase chosen for speed and integration into Android, Appendix X.

The image recognition method found by the research to be most suitable was the Google Cloud Vision API. This had features such as:

- can understand the content of an image by encapsulating powerful machine learning model
- easy to use REST API.
- classifies images into thousands of categories (e.g., "sailboat", "lion", "Eiffel Tower")
- close integration into Android

Similar apps and functionality

Research was also conducted in the literature review on the capabilities and functionality of existing butterfly or similar identification apps, this resulted in a feature list of best features to be incorporated into this app. This is shown in Appendix X and summarised in figure X. Here the most popular or prevalent features are each app are summarised, with the research showing that low latency high speed image capture, and machine based identification were most popular.

App name \Attribute	Camera	Manual based ID	Machine based ID	Location logging	Sharing online	Low capture latency
But Fly ID		x		x		
Wings	x	x	x		x	x
ID that Fly	x		x			x
Utterly Butterfly		x	x	x		x
This app	x		x	x	x	x

Figure X. Most desired app features from competitor apps

User profiles and app targeting

Research was also carried out into what system opportunities and constraints were available for the app. This analysis is shown in Appendix X. This research showed that for maximum user uptake targeting Android phones with their large market share would be a good choice for the app. It showed that a certain level of application backwards compatibility would allow more user uptake, hence the development would target the lowest level of Android that would still allow the app to features to function correctly.

Development environments

Research was conducted into what development methods and environment to use, with Android Studio being selected as the IDE and an iterative development method chosen.

User survey results

Finally, the research also looked at the needs of users with a short survey of amateur butterfly spotters usual methods and what methods they might like from this app, Appendix X. These results are shown in Appendix X. The main features asked for were speed of taking the photograph in case the subject moved, and ease and reliability of the identification. The ability to share results instantly was also very popular.

Analysis

- Summary of your research and literature review
- What does the research mean
- How does the research influence how your app works
- What are the results of your research
- 4 to 6 pages, more if you need
- This research is from your first report literature review conclusions, put that in the Appendix, talk about it and expand it here
- You can also rewrite your original literature review conclusions if you like
- Imagine somebody is reading your report and knows nothing about what your research means, so you have found things out but what does it mean?

<Example few paragraphs, write much more and consider using figures and tables>

<add your own headings to suit your project>

<you don't need to use my headings or ideas here, feel free to do what you feel!>

Performance of machine object recognition

The research showed that machine based identification techniques could perform as well as experienced human amateurs, so the main focus of the app was aimed at maximising image identification. This meant high resolution images must be captured.

Effects of image capture time

A second finding from the research was that users required a fast image capture time, this raises a problem as higher resolution images take longer to capture. A compromise was to design the app so that the image resolution, and hence capture time and accuracy of identification could be chosen by the user to suit the environment they were in at any time.

Popular features and user preferences

The research also showed that other completing or similar apps that were popular tended to have fast capture times and machine based image recognition.

Finally the needs of the potential users were addressed by adding sharing features to the app.

Design

- How does the research and analysis lead to your app design
- App design sketches
- Design diagram snippets
- Put the full diagrams in the Appendix
- 8 to 16 pages, more if you need
- You may already have some design documentation
- Typical examples include database design (ERD, schema, etc), UI design, UML design, system/software architecture, etc. Anything that showcases the design of your system
- These designs will be placed in the appendices with snippets shown here
- Imagine somebody is reading your report and knows nothing about your app design, explain how you designed your app to them

<Example few paragraphs, write much more and use more figures and tables than this>

<add your own headings to suit your project>

<you don't need to use my headings or ideas here, feel free to do what you feel!>

Overall Design Aims

The app design was inspired by the aims and objectives of the project and constructed from the results of the research and analysis. The main features determined were:

- Image capture via the phone camera, allowing choice between speed and image resolution
- Butterfly image identification via an online service, preferably using Google image analysis
- Geolocation, time/date recording of where and when the identification took place
- Account creation and logging in/out of a shared online butterfly information database
- Creation of an online database for sharing butterfly identification data, location, time, date, preferably using Firebase

Functional requirements

<these are from your first report, put them in the Appendix, talk about them here and add a few samples, describe how you made them here>

Functional requirements for the app were constructed based on the user needs from the app, and the design research and analysis. These requirements are shown in Appendix X.

A sample functional and non-functional requirement table is shown in Table X. This shows the requirements to send and retrieve the captured image.

Requirement ID	Description	Functional requirements	Non-functional Requirements
1	Image identification functions	Maintains connection to Google Servers	Must have a latency < 10 seconds
2		Sends camera image to Google Servers	
3		Identifies image	Must be >80% accuracy

Table X. Main image identification Functional Requirements

User interface designs

Initial user interface designs were sketched to give basic concept drawings of the app. These are shown in Appendix X. An iterative user interface design methodology was adopted where basic sketches were evolved into more detailed user interface designs. All designs aim to be Android Material Design compliant, though allowing for some flexibility in design to suit the nature of the app. A sample sketch of a first iteration is shown, Figure X.

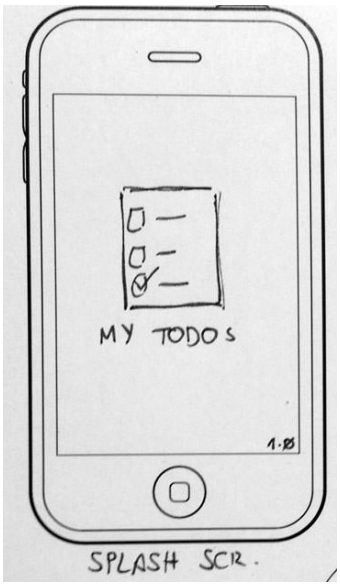

Figure X. sample interface design sketched

This shows the basic identification interface concept. Note that Material Design has been use to size and position the buttons.

<find something to criticise, or make comments on, talk about things that went well and things that went wrong, you get marks for explaining what you did, for example> *there were issues fitting all the required buttons onto a single screen so two screens were used for the image capture and image identification requirements. The original and final screens are shown in figures x and x.*

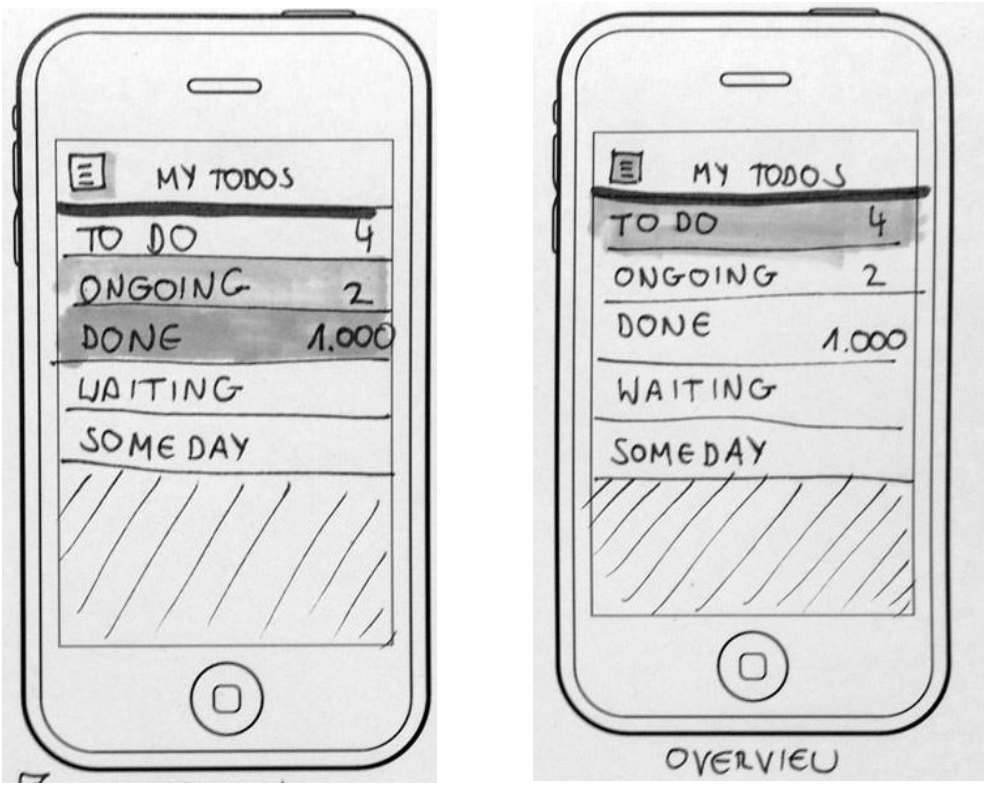

Figure x, before left, after redesign right

App interaction flow

The interaction flow was sketched out to determine the best flow based on the user requirements from the survey, see Appendix X, and the basic app functional requirements. These diagrams are shown in Appendix X. The flow of the main app is shown in figure X.

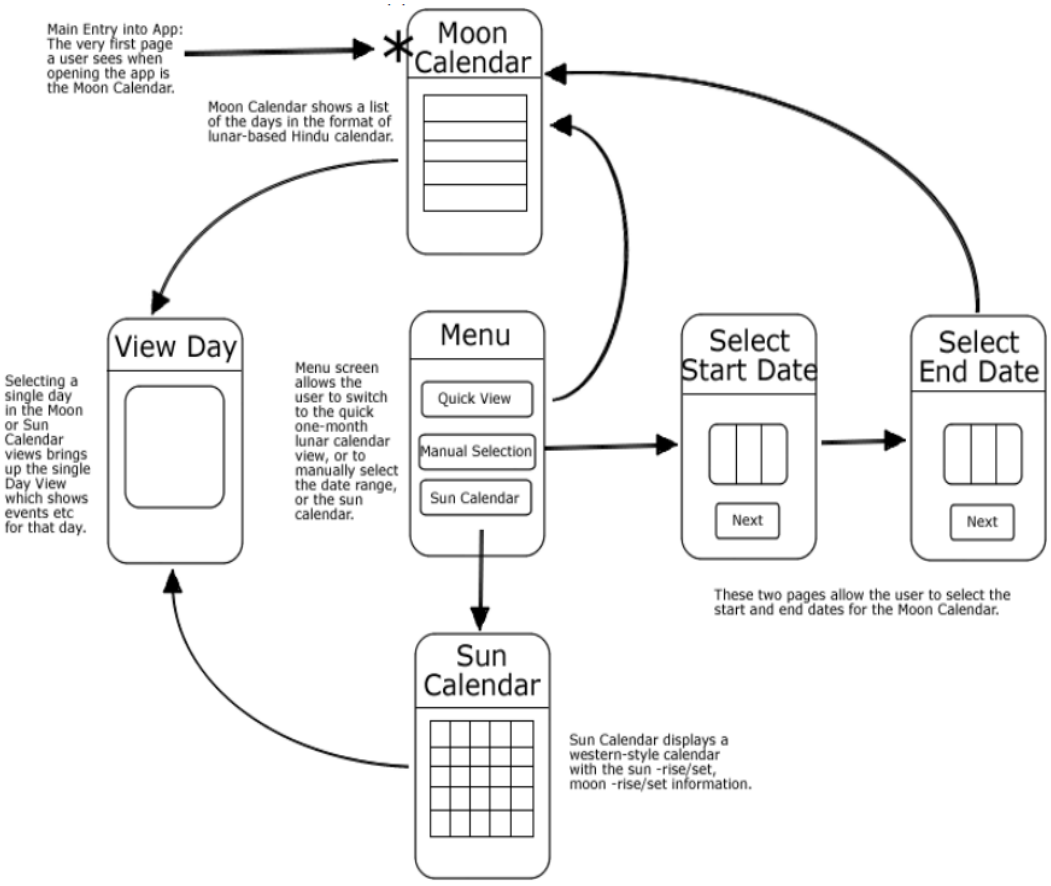

Figure X. App interaction flow

The app flow aimed to minimise the number of screens the user needed to navigate through to complete the tasks required. The flow was also summarised using flow charts, see Appendix X, a sample of the main app flow is shown in figure X.

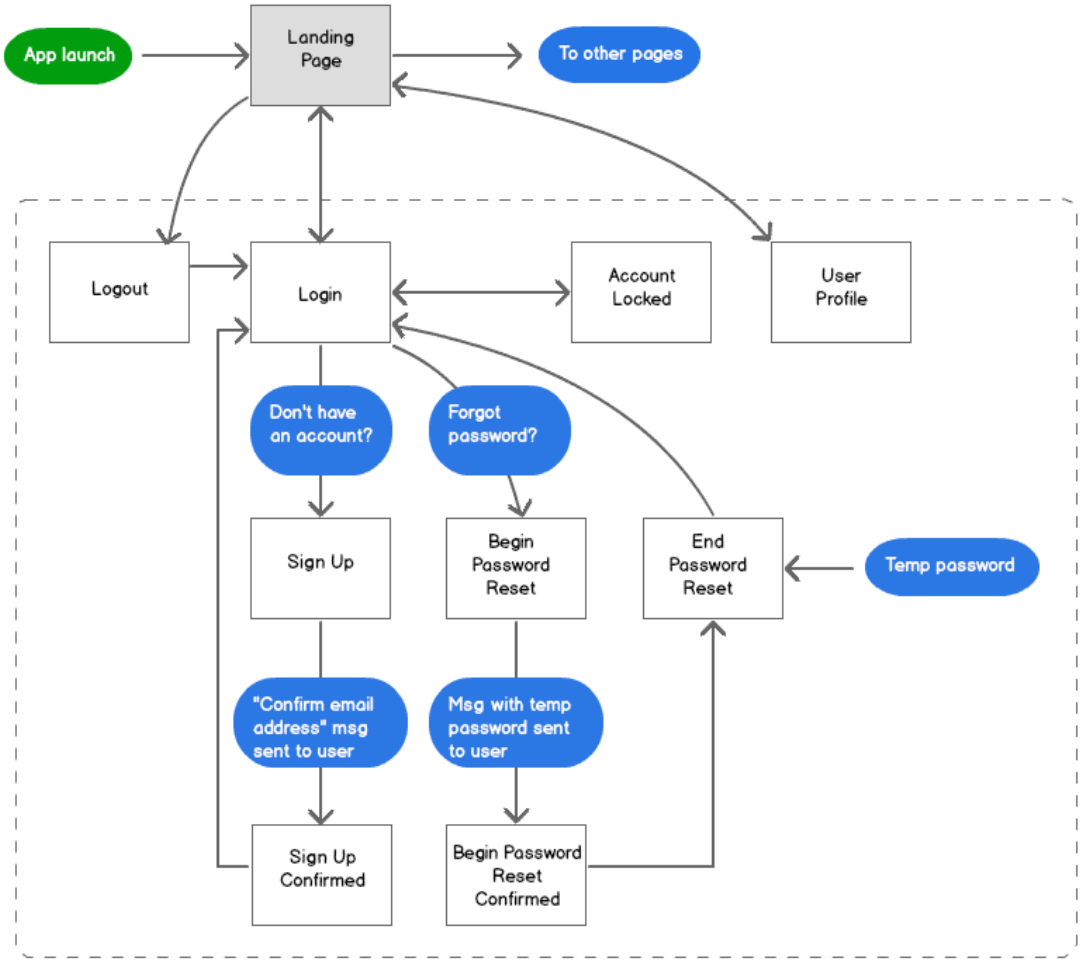

Figure X, app main interaction flow

<find something to criticise, or make comments on, talk about things that went well and things that went wrong, you get marks for explaining what you did, for example> *The flow originally designed, Appendix X, caused an additional page to be displayed between login and the home screen every time the app was redisplayed, this was removed by moving the login to only be required if the phone was shut down, rather than every time the app appeared on the home screen of the phone.*

Use Case Diagrams

Use Case diagrams were constructed to visualize the functional requirements of the app, they helped to show how the functional requirements join together when the app was being used.

The Use Case then led to the definition of the Classes and Functions of the app, and the relationship they had between them. Appendix X shows the main Use Case Diagram, with a sample for the main image identification system shown in figure X

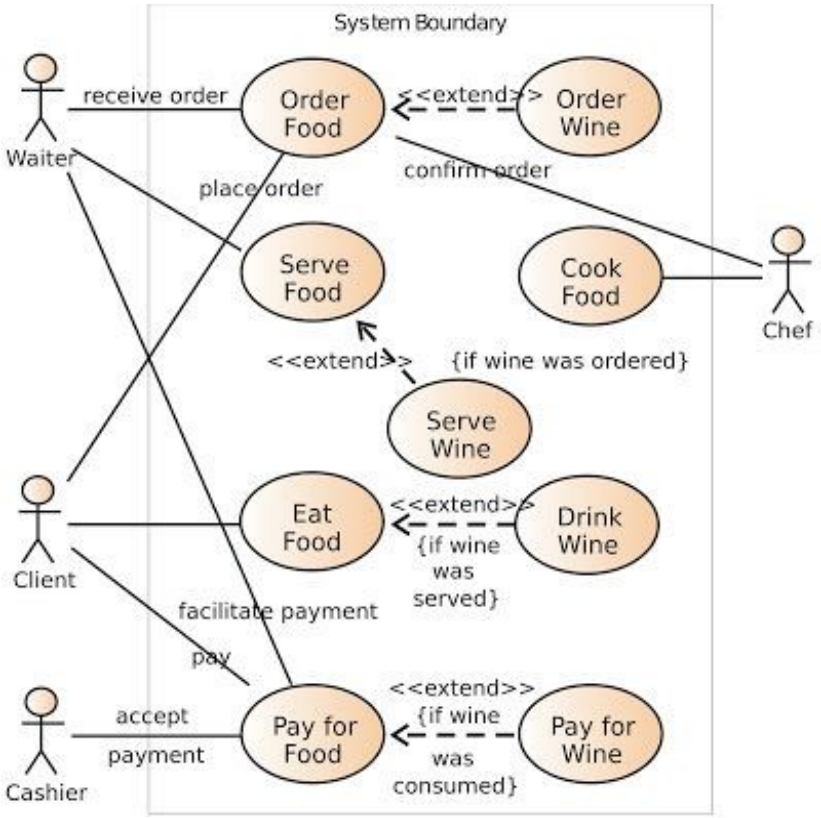

Figure X. Use Case diagram for the image identification functions

<find something to criticise, or make comments on, talk about things that went well and things that went wrong, you get marks for explaining what to did, for example>
Problems were found with the initial diagram, Appendix X, where there were conflicts in the identification screen meaning additional send and receive buttons were needed, adding complexity to the user interaction flow. The UCD was redrawn, Appendix X, to remove this problem, this revised UCD is shown in figure 5.

Derivation of Classes and Class diagrams

The Use Case Diagrams and Functional Requirements were used to help form Class Diagrams for the app. These class diagrams visualised the relationships and dependencies among the classes and defined the Classes of the app. A simple design was used where each main function of the app was represented by a main class, with the implementation of that feature implemented by sub classes of the main class. The full class diagram is shown in Appendix X, and figure X shows the main classes for each feature of the app.

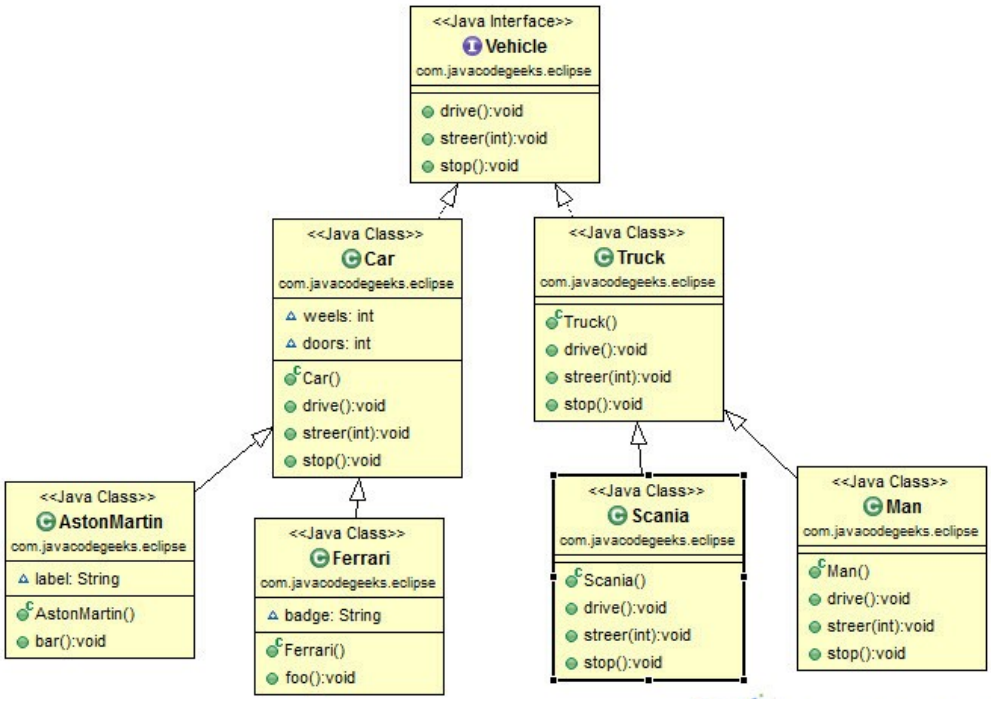

Figure x. Abbreviated main class diagram

<find something to criticise, or make comments on, talk about things that went well and things that went wrong, you get marks for explaining what to did, for example>
After the initial class diagram, Appendix X, it was found that the user login was originally part of the user share data class, such that the login class was a sub-class of the main share class. Moving log to a separate main class gave a more coherent class structure that allowed the login and share classes to be developed separately before incorporation into the final app, see Appendix X for the original and revised diagrams.

Database Entity Relationship design <if you have a database>

The app required a database to hold the identification and location user data. This was held in the Cloud using Firebase. A ERD was constructed for Firebase, the basic diagram is shown in Figure x, and the full diagram in Appendix X.

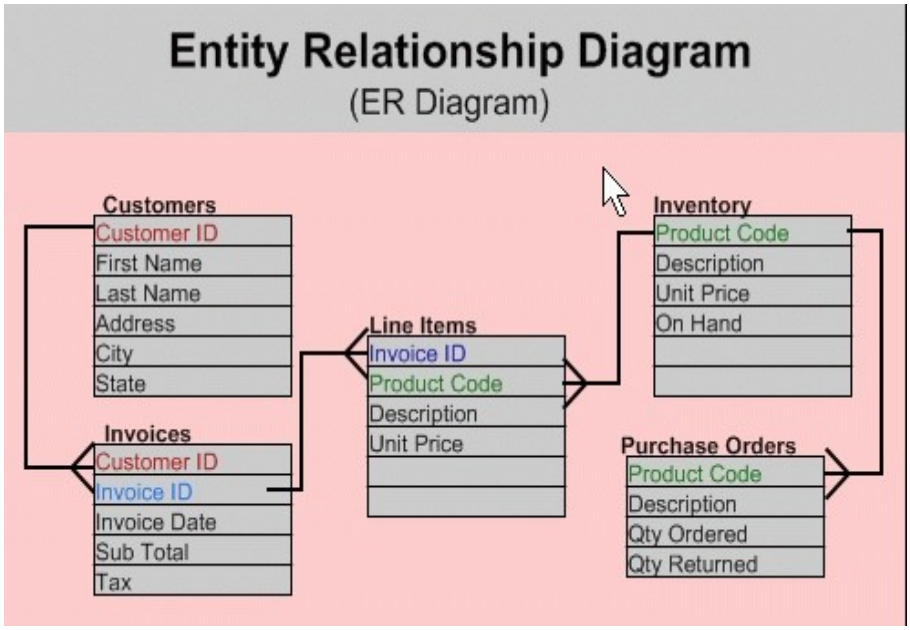

Figure X. Simplified Entity relationship diagram for Firebase.

Firebase used a JSON format, and required a simple structure for each user. Firebase allows near instant variable synchronisation across all apps connected to the server, allowing all instances of the app to be synchronised if required. This allowed the near instant sharing of butterfly identifications to all app users.

<find something to criticise, or make comments on, talk about things that went well and things that went wrong, you get marks for explaining what to did, for example>
The database required a key for the username, and also location and the butterfly identified, so other users could search for butterfly and location. This required login security for the user identity, but sharing publicly to other users the location and butterfly identified. To do this the database had separate public keys for butterfly and location. This was implemented in firebase...

Sequence Diagram design

The app required a sequence diagram for the flow of interaction between the app and the Firebase database. The main diagram is shown in Appendix X, an example is shown in Figure X where the log in sequence to the app is shown.

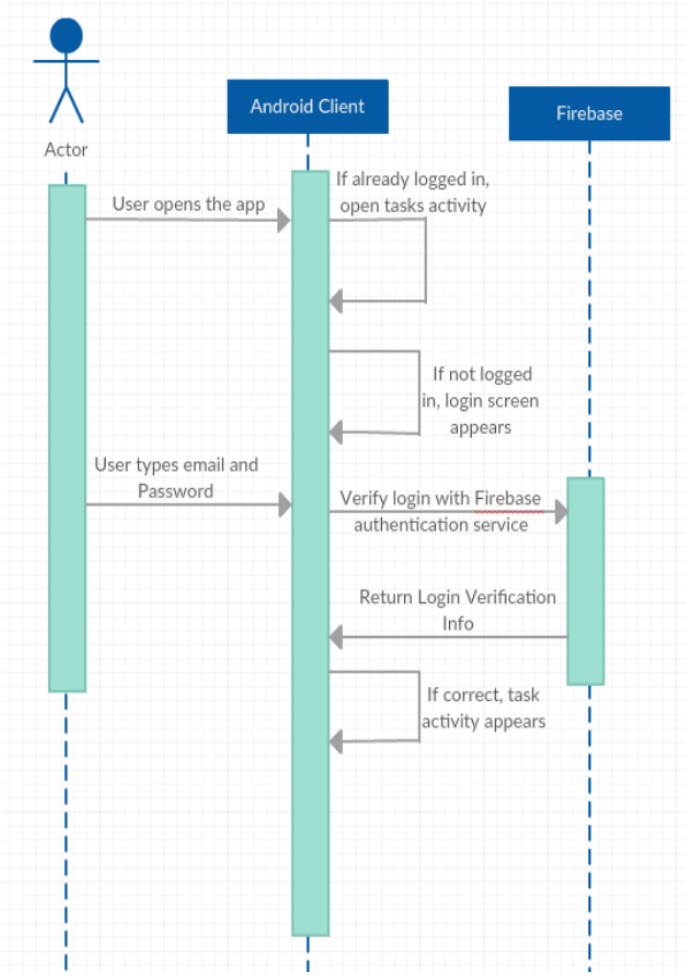

Figure X. Login sequence diagram

<find something to criticise, or make comments on, talk about things that went well and things that went wrong, you get marks for explaining what to did, for example>
The sequence diagrams, Appendix X, showed that there could be conflict if two or more users were accessing the database simultaneously, this could lead to contention with database entities being locked by Firebase when the app attempts to update them. This was solved by adding a lock and flag to the app to indicate if the database update was successful or not, and if it should keep attempting until the entry is updated...

Programming language and architecture

<Talk about choice of language, IDE, architecture etc> Android was chosen as the programming language, this is based on Java. A development architecture of MVC, Model View Controller was chosen to separate the core underlying code from the Android interface design, this allowed a range of possible interface designs without the need to alter the underlying core code. The level of Android chosen was Lollipop, with the choice of user base shown inn Appendix X.

Software development methodology

<Talk about choice of methodology SCRUM, SSADM waterfall, incremental design etc> The project required several different modules to be incorporated, with the possible risk of one of more not necessarily working as planned. This meant changes may happen in the project as it proceed. Hence e the project had some inherent risk. To counter this risk an Agile project development approach was adopted. This was coupled with an incremental and iterative development life cycle, allowing development to be flexible and allowing the testing of individual modules before incorporating into the main project as they became available. For example, a basic camera image capture module was constructed and tested, before being added to the main app. This approach was adopted for all of the core functions of the app.

<Anything else you can think of that's to do with design of your app>

Implementation

- Describe how you wrote your code
- This can be a long section, 10 to 20 pages
- break it down into modules and describe them
- For each module show code snippets, class diagrams, etc to explain how it works
- One of the most important parts of your report is the implementation chapter, which should provide a step by step account of how you developed your software.
- There will be different ways to structure this chapter and it will take some consideration, e.g. front-end, back-end; different users; different components.
- You can include fragments of code and screenshots (which may reflect the application during development), but may also refer to the appendices.
- You do not want to clutter your main report with large amounts of source code but can put key components, functions, classes, etc, into your appendices and refer to them.
- Do NOT put all of your source code in the appendices, only important / technically accomplished areas.
- Your appendices may also contain screenshots for your entire application that can be referred to.
- Imagine somebody is reading your report and knows nothing about how you wrote your code, explain it here

<Example few paragraphs, write much more and consider using figures and tables>

<add your own headings to suit your project>

<you don't need to use my headings or ideas here, feel free to do what you feel!>

Overall Implementation Aims

The development was broken down into the five main functions or modules of the app, image capture, identification, geolocation and date and time, account creation and log in/out, and finally online sharing. Each of these were developed and tested separately before incorporation into the final completed app.

This section will document the development of each module individually, before showing how each feature was brought together to form the final app.

1. Image Capture Module Development

The image capture required use of the phone camera together with a capture screen that would have low latency, as defined by the user survey, Appendix X.

<Talk about satisfying your functional requirements> This module satisfied the functional requirement X 'Latency of the camera' , from the functional requirements of the app, see Appendix X.

Requirement ID	Description	Functional requirements	Non-functional Requirements
1	capture image	invoke camera display image respond to capture button capture image	Low latency, <100ms for capture
2		save image to bitmap	none

Table X. Main image capture functional requirements

<Talk about interesting code snippets> The image capture function required access to the camera capture functions of Android. These were accessed by the XXXX function, the critical code is shown in Figure X. ***<code should be fully commented at the end of lines, showing that you understand what it does>***

main.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/frameLayout">

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/gridLayout"
        android:layout_gravity="center"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Login:"
            android:id="@+id/textView"
            android:layout_row="0"
            android:layout_column="0"
            android:paddingRight="20dp" />
```

Design Text

Figure X, camera capture code

<Talk about class design choices and use class diagram snippets>

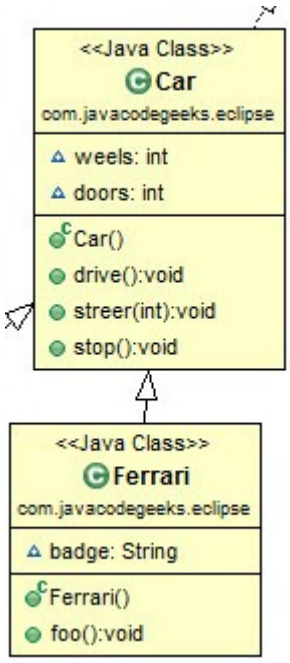

Figure XX. Capture class and camera handler subclass

<Talk about sequence diagram snippets>

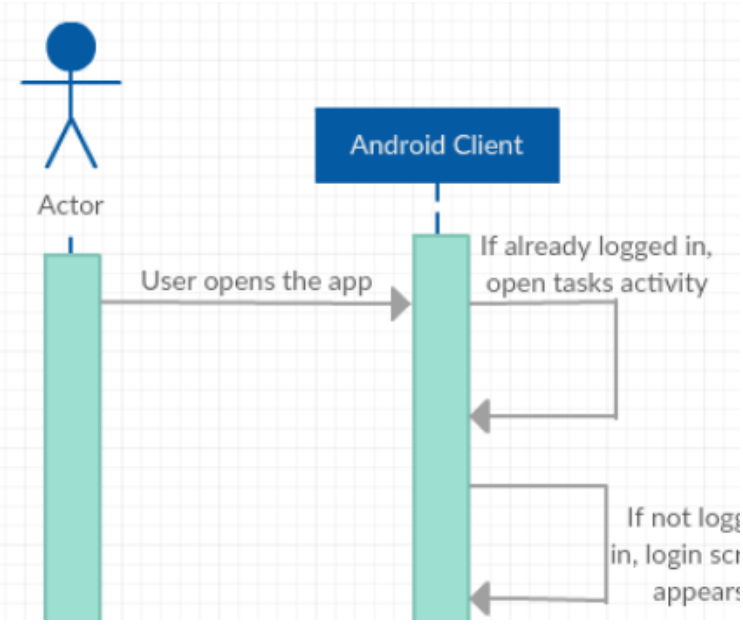

Figure XX. Camera handler image acquisition sequence diagram

<Talk about interface design implementation and choices> The capture module interface was implemented with two design choices, one with a single capture button, and one with additional buttons for resolution and image quality, figure X. The buttons to choose resolution, exposure, focus, flash were either displayed on a separate page, on on-screen with eh camera images. Trialling both interfaces it was felt that the interface with the additional buttons was more useful and this was incorporated into the main app.

Figure X. Camera capture screen options, left one button, right many buttons

The module was tested for errors and the results shown in the test table, Appendix X, with the main results shown in Table X. Here the camera image is captured correctly, and allows the user to re-take the image or send the image for analysis.

Test	Input	Expected outcome	Actual outcome	Pass/Fail
Show camera image	Invoked by selecting capture button	Camera image displayed. Capture button displayed	Camera image displayed. Capture button displayed	Pass
Capture image	Capture button selected	Camera captures image. Saves to bitmap.	Camera captures image. Saves to bitmap.	Pass
Time to capture image after button press	button press	<100ms latency	>100ms latency, typically 200ms	Fail

Table X, unit test of image capture module

<find something to criticise, or make comments on, talk about things that went well and things that went wrong, you get marks for explaining what you did, for example> *The capture function required more lines of code than originally thought, and took longer than expected, the original project Gantt chart, Appendix X, was updated to reflect this delay, with a new chart produced, Appendix X.*

The capture function had a latency greater than 100ms, but after changes to the code, Appendix X, this was brought down to 120ms, which was deemed acceptable.

Test	Input	Expected outcome	Actual outcome	Pass/Fail
Time to capture image after button press	button press	<100ms latency	120ms latency	Acceptable fail

Table X, unit re-test of image capture module

Summary

The image capture module proved to be difficult to implement due to unexpected additional code required to setup the camera and reduce the camera capture time. The module passed testing satisfactorily, although with a slightly longer latency than required. It is thought upgrading the phone to faster hardware might solve this issue.

<that's the end of describing this module, now repeat for the remaining modules>

2. Butterfly image identification

3. Geolocation, mapping and time/date recording

4. Account creation and logging in/out

5. Creation of an online database using Firebase

Final App Development

<After describing the individual modules, now talk about putting it all together>
Once the individual modules were completed and passed testing, the final app was assembled. The final app is shown in Appendix X, and screen shots of the capture and identification screen shown if Figure X.

Figure X, Capture left, and Identification screen right.

<write some more on how well it went together, any problems, solutions etc>

Publication on the App Store

The finished app was submitted to the App Store and accepted with alterations, see Appendix X. The app has been downloaded over 100 times in the first month (references).

Test Results

- Test results for the app, tests based on functional requirements
- Test tables in the appendix, snippets here
- Think of a test for each Functional Requirement
- Test all of the things your app does, text boxes, buttons etc
- Put the test results in the Appendix
- Show snippets from the testt results here

<Example few paragraphs, write much more and consider using figures and tables>

<add your own headings to suit your project>

<you don't need to use my headings or ideas here, feel free to do what you feel!>

The modules of the app all passed their individual test. The completed app was tested for overall functionality and passed all tests apart from camera latency. The test results are shown in Appendix X. Table X shows the main test results for the app.

Test task	Functional Requirement	Input	Actual Result	Pass/Fail	Comments
Show camera image	Requirement 6.1 Show camera image	Camera button selected	Camera image displayed. Capture button displayed	Pass	Buttons may need different layout to be easier to select
Capture image	Requirement 6.2 Image capture	Capture button selected	Camera captures image. Saves to bitmap.	Pass	
Identify image	Requirement 6.3 Identify image	Image from camera	Butterfly name	Pass	Name retrieved from google server
Share image	7.1 share	Butterfly name	Successful share	Pass	
Login user	4.1 login	User name and password	Login	Pass	Text may need to be larger
Etc...					

Table X. Test results of the completed app

User Manual

- Write a short user manual
- Use screen shots
- Put it in the Appendix
- Mention it here

<Keep it to the point, simple>

A user manual was written to cover how to operate the app. The full manual is set out in Appendix X. The manual covers all aspects of capturing and identifying a butterfly, together with how to create an account and share your identifications. A sample entry is shown below in figure X.

User Manual

Capturing an Image

1. Press the capture button on the home screen
2. The capture a screen is shown you will see a camera image
3. Point the camera at the butterfly
4. Let the camera focus
5. Press the capture button
6. The captured image will be displayed

Identification

1. Once an image is captured you can identify it or try again
2. to try again press the capture button again
3. to identify press the Identify button

Figure X. User manual entry

Risk Analysis

- Did you risk mitigation measures work
- Did you lose data
- Did GitHub work correctly
- Did your backups work
- Use the risk analysis from your Project Contract

<Talk about data security>

All project files were linked to GitHub to minimise any potential data loss, this worked successfully as a version was lost due to a disk failure, and this was available via GitHub.

Project Planning

- Did you risk mitigation measures work
- Did you implement all the core functions
- Did your time management work

<Talk about how your plan went, did you need to change it?> The project Gantt chart was updated to keep the project on course several times. At one point illness caused a week delay, Appendix X, The capture function required more lines of code than originally thought, and took longer than expected, the original project Gantt chart, Appendix X, was updated to reflect this delay, with a new chart produced, Appendix X.

These changes to the Gantt chart, by compressing other activities such as minimising the development time for less critical functions, and reducing the complexity of the sharing system, allowed the project to remain on schedule.

Critical Evaluation

- 2-4 pages
- You can use the first person from now onwards - ‘I thought, I tested...’
- A talk and summary about how the whole project progressed
- Talk about how you overcame problems
- Talk about fulfilling the Project Contract
- The majority of the main report looks back at what you have done in past tense.
- The critical evaluation is usually written in present/future tense, as you are evaluating what you have achieved and also looking ahead to the future.
- There are different ways to structure this but one approach is to revolve it around the objectives in your project contract.
- You should write about what is both good and bad about your software (and project experience more generally) – try and focus on the positives, but clearly do not neglect the obvious negatives or enhancements.
- Obvious areas are your software, research, requirements, etc. But you can also consider how you managed the project, how the plans evolved, the chosen methodology, development tools you used, etc.
- Also consider future enhancements to your software.

<Example few paragraphs, write much more>

The butterfly identification app was competed on time and passed all testing.

It fulfilled all but one of the functional requirements, with some difficulties in coding of the sharing functions meaning that only simple sharing of data was possible.

Also the camera capture speed was slightly below the requirements.

The app worked well on test images.

Delays were caused by problems in coding but were overcome by time management changes.

Overall I felt the project went well and...

Lessons learned were...

Summary

<200-600 words, more if you need to. Wrap it up. How did it go, were you happy, what will you do next with the app!>

References

<List of references probably quoted mostly from the Research and Analysis sections>

Appendix Index

<A contents page or index of the Appendices>

Appendix

<no page or words limit for this section>

<to contain, in roughly this order>

- Project Contract
- Ethical Review
- BCS Checklist
- Global Checklist
- Project Plan
    - Gantt Chart and other project planning presentations
        - landscape, big and easy to read
        - include any other planning graphics here
        - include updated Gantt charts ***<show original and updated charts>***
- Literature Review ***<you can rewrite your old one>***
- Any survey or questionnaire results
- Functional Requirements ***<you can rewrite your old one>***
- Design Sketches

- Design flowcharts
- Design Use Cases
- Design Class Diagrams
- Design Data Structures
- Design ERD’s
- Design Sequence Diagrams
- Any other design diagrams
- Code listings ***<snippets, not all of it, with explanations>***
- Brief user manual
- Periodic Progress Reports ***<don’t forget, you get marks for these>***
    - In order, oldest first.

That’s it! Put it together as a single pdf. Must be submitted via TurnItIn.