



FUNCTIONAL SPECIFICATION

**DEPARTMENT OF ELECTRONIC ENGINEERING
UNIVERSITY OF YORK**

MENG YEAR 3

SOFTWARE ENGINEERING GROUP PROJECT

**PENELOPE
JANUARY 2023**

(CURRENT DOCUMENT VERSION 3.0.0)

Document update history:

Version 1.0.0

Created by Ana Monteiro and Ophelia Korontinis on Nov 5, 2022

Description: First draft of Functional Specification Document (FSD)

Version 2.0.0

Modified by Connall Shurey on Nov 11, 2022

Description: Overhaul of FSD Structure and content

Reviewed By Ana Monteiro and Ophelia Korontinis on Nov 17, 2022

Version 3.0.0

Modified by Connall Shurey, Ana Monteiro, and Ophelia Korontinis on Jan 19, 2023

Description: Updated according to received feedback

Reviewed By Connall Shurey, Ana Monteiro, and Ophelia Korontinis on Jan 19, 2023

Contents List

1. Introduction	1
1.1. Project and Scope	1
2. System Overview	1
2.1. Application Diagrams	2
2.1.1. Interface Diagram	2
2.1.2. Data Flow Diagram	2
2.2. User Roles and Responsibilities / Authority Requirements	3
2.3. Dependencies	3
3. Functional Specifications	4
3.1. User Client Application	4
3.2. Admin Client Application	5
4. System Configurations	5
5. Non-Functional Requirements	6
6. Deployment	6
References	6

1. Introduction

1.1. Project and Scope

Our aim is to develop and sell an educational Android application to provide students of campus-based universities, and local members of the public, a means of identifying and learning about the local avian wildlife.

By providing potential future students with information on the wildlife that exists on campus, we are giving them the opportunity to fall in love with the university even prior to beginning their studies. Therefore, our target audience includes students that have interest in studying in a nature-surrounded environment in the UK.

In this project's scope, we will initially focus on supplying information regarding wildlife at The University of York. This will include images, videos, sounds and text which will act as a template that can be used to expand our application to cover any university in the UK. Users will be able to use the application as a journey of discovery, identifying different species of birds around the campus and learning about their characteristics, diet, location, and fun facts. Users that purchase a guide licence will be able to load data from their desktop, allowing student societies and third-party providers to get involved in expanding the database of existing species and information.

Universities would benefit greatly from the application by increasing the likelihood that a potential student will choose the university as their place of study. Students seeking a calm, peaceful and natural environment that gives them the opportunity to release the stress associated with studying a degree will be motivated by the familiarity with local wildlife gained through our application. The application would also increase the university's revenue by attracting more bird watchers and curious members of the community to the campus and therefore income from sources such as parking and on-campus cafes would increase, while also engaging the community in the university, furthering its reputation.

2. System Overview

The system consists of an android user application, a desktop admin application and a database hosted on a remote server.

On the main user client, developed for Android, the user is presented with the home page containing a list of local avian wildlife species upon launching the application. From this page the user can change their location or view information about the application. Choosing a bird presents the user with a bird profile page containing the related information. Basing this application on android use allows the user to easily access the application while on the go.

A secondary guide admin client, developed for desktops, allows guides to create, read, update, and delete information from the database. Due to this client giving access to the application's database, appropriate security measures need to be taken to ensure the validity of the user's intentions. This includes requiring an API key to access the client. Basing this application on desktop use allows for ease when editing information and adding content from file browsers.

Having an admin key on the admin client will give the required access for creating API keys and creating/updating locations.

2.1. Application Diagrams

2.1.1. Interface Diagram

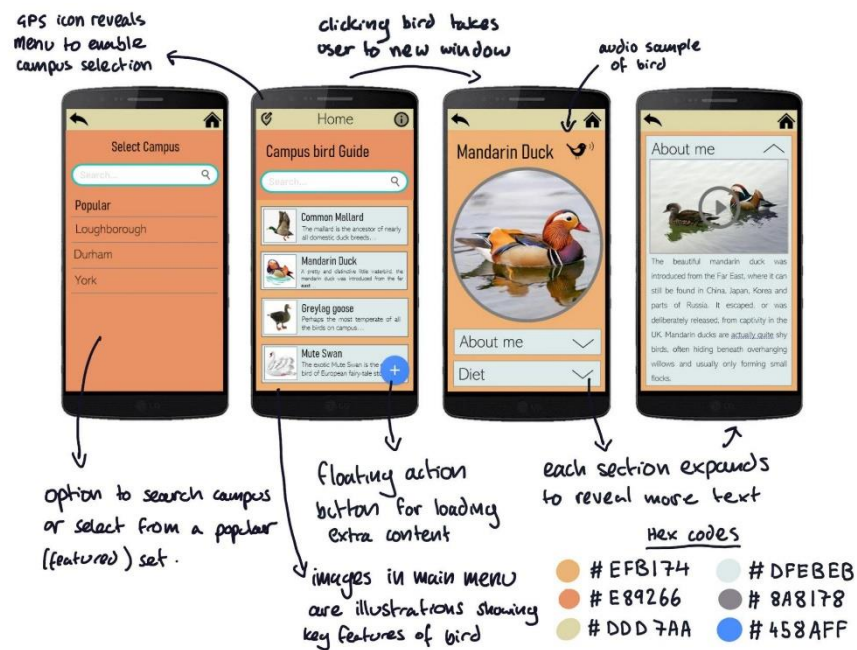


Figure 1 - Indicative interface diagram (demo of the app)

2.1.2. Data Flow Diagram

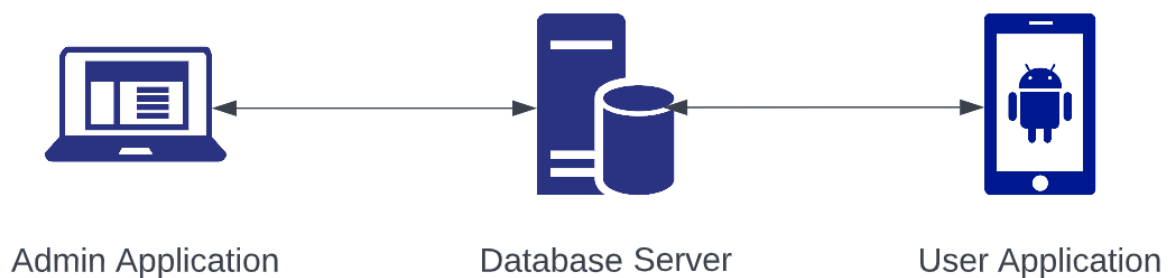


Figure 2 - Data Flow Diagram

2.2. User Roles and Responsibilities / Authority Requirements

User / Role	Example	Frequency of Use	Security/Access, Features used	Additional Notes
User	Students, Bird watchers	Frequent	Can view bird information.	
Admin	Groundskeepers, Bird Experts	Occasional	Can edit/create duck content. Requires API Key	
System Admin	Developers	Rare	Can create API keys and edit/add locations. Requires system access	

2.3. Dependencies

The functionality of the live application will depend on:

1. The media handlers written by a contracted third party
2. The server host being fully functional and having a stable connection (if the hosting platform crashes then the server host would crash)
3. Future proof code that will not go out of date as new Java and library versions are released
4. Complying with android/iOS app guidelines

3. Functional Specifications

3.1. User Client Application

Purpose / Description	<p>Specifications related to the android client used by regular users.</p> <p>This includes the following pages:</p> <ul style="list-style-type: none">● Loading Page<ul style="list-style-type: none">○ This is the page the user will see before the content of the app is loaded● Landing Page/ Home Screen<ul style="list-style-type: none">○ Includes the list of birds○ Includes a button that enables the user to load additional content (additional birds)● Individual Bird Page<ul style="list-style-type: none">○ Includes all the information about the specific bird chosen by the user as well as some images and videos of the bird seen on campus● Campus Selection Page<ul style="list-style-type: none">○ Allows user to switch location/campus
Use Case	<p>This client will be used to identify birds and view related information</p>
Functional Requirements / User Stories	<ul style="list-style-type: none">● As a User, I can see a list of birds by location on the user app home page which are stored on the database● As a User, I can change the location on the user app to change which birds are viewed from the database● As a User, I can see details about a particular bird in the list sent from the database● As a User, I can load a third-party provided bird presentation file into the user app● As a User, I can see instructions on the user app describing how to use the user app

3.2. Admin Client Application

Purpose / Description	Specifications related to the desktop client used by regular admins and system admins.
Use Case	<p>The regular admins will be able to create, read, update, and delete information about birds in the database.</p> <p>The system admins will be able to create API keys and new locations as well as create, read, update, and delete information about the birds in the database.</p>
Functional Requirements / User Stories	<ul style="list-style-type: none">• As a regular admin, through the admin app, I can create, read, update, and delete a bird entry in the database• As a regular admin, through the admin app, I can create, read, update, and delete a bird's assets• As a System Admin, I can create, read, update, and delete a location on the database using the admin client• As a System Admin, I can create, read, update, and delete API keys on the database using the admin client

4. System Configurations

To configure application:

User Client:	No configuration necessary. Download the app and launch to browse
Regular Admin Client:	Requires input of an API key
System Admin Client:	Requires An API key which has system admin level of access which can only be generated by System Admin Clients. The first system admin key must be manually generated through the server, however every key after would be generated by existing system admins.

5. Non-Functional Requirements

On the bird page of the user client, the following media is required:

Media Type	Media Description
Text	<ul style="list-style-type: none">• Names of birds• Description of birds• Preferred location• Diet• Fun facts
Sound	<ul style="list-style-type: none">• Characteristic sound of each bird
Images & Video	<ul style="list-style-type: none">• Images and videos capturing each bird in its natural element

6. Deployment

Applications will be deployed on Amazon Web Services (AWS) as they can deliver a reliable service while adhering to suitable sustainability standards (see sustainability bill [[1](#)]). The entire server application and database can be implemented on AWS (or any other web hosting service if there is a price advantage), so our servers will be centralised and easier to maintain.

We will deploy the user app on the android Google store for users and will manually send the admin application to admins with their respective key.

References

1. Amazon, “Sustainability in the Cloud”, Amazon Web Services, [Online]. Available: <https://sustainability.aboutamazon.co.uk/environment/the-cloud?energyType=true>