|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Team:** The Wildlife Committee | | | | |
| **First name:**   Nicholas | **Last name:**   Ramnaraign | **York Email:**   nramnara@my.yorku.ca | **Lecture Section:**   N | **Lab Section:**   3 |
| **First name:**   Jade | **Last name:**   Pablo | **York Email:**   jade03@my.yorku.ca | **Lecture Section:**   N | **Lab Section:**   3 |
| **First name:**   Anania | **Last name:**   Tsegaye | **York Email:**   anania17@my.yorku.ca | **Lecture Section:**   N | **Lab Section:**   3 |
| **First name:**   Oluwafunmbi | **Last name:**   Akande | **York Email:**   funmbia@my.yorku.ca | **Lecture Section:**   N | **Lab Section:**   3 |
| **Project Title:** The Wildlife Guide | | | | |
| **Project Description:**  For our EECS1022 group project our team will be creating a mini-database app that will feature general information on popular animals. The app will function like a digital encyclopedia where the user will be able to input the name of an animal into the search, and the app will display all information relevant for that animal. The user will also have access to a full list of included animals if they wish to browse using that method. The information will include all general information for the animal (size, location, etc.), as well as a photo.  When the user launches the app they will be brought to a screen where they can search for the animal they are interested in. If the animal is available in the database then the user will be brought to that animal’s page, if the animal is not in the database then the user will be notified on the search screen. Each animal’s page will provide general information for the animal, as well as suggestions for similar animals located at the bottom of the page. The user also will be able to control which sections of information are displayed on the animals’ pages, this gives the user the control to tailor their experience within the app to better suit their interests. With this app the user will be given the ability to quickly access information on their favourite animals with ease. | | | | |

**Requirements Definition:**

1. When the user launches the app they will be brought to the ‘Search’ screen where they will be able to search for animals by selecting the search bar displayed on the screen.
2. The user will be able to use the search bar to find the animal they are interested in by typing in the animal’s name.
3. When the user enters an animal into the search bar they will be shown a list of animals that closest match their search term.
4. If no animal in the database matches the input of the user, text will be displayed on the screen saying that their search was not found in the database.
5. Once the user selects an animal that exists in the database, the animal’s photo and information will be displayed on the screen.
6. If the user is viewing an animal’s page there will be a button available to take them back to the previous screen they were viewing.
7. If the user is viewing an animal’s page, other animals will appear as buttons at the bottom of the page which are related to the current animal (by biome, taxonomy, etc.).
8. The user will always have access to a widget which they can use to quickly pull up a search bar, and access the settings menu.
9. If the user clicks on the ‘All Animals’ button, they will be brought to a page listing all animals, with their pictures and names displayed for the user.
10. If the user is on the ‘All Animals’ page, a dropdown menu will be available to give the user the ability to sort animals by different search criteria (biome, sorted alphabetically etc.).
11. If the user clicks on the ‘Settings’ button, they will be brought to a page where they can adjust settings relevant to the animal’s pages, as well as user settings for the app.
12. The settings page will allow the user to toggle on and off what information is displayed on each animal’s page, this will include description, colour, size, etc.
13. The settings page will also allow the user to change basic user settings such as dark/light mode and text size to allow a more comfortable viewing experience.

Graphical user interface, application

Description automatically generated**User Interface Wireframes:**

**Phase 3:**

So far in the phase 3 process our team has implemented the core functionality of the app. This includes being able to load a list of animals found in our database, and being able to select any of them to bring up an info page providing detailed facts about that animal. Our app first loads a list of available animals for the user to select. When the user selects one of those animals, the information on that animal is then loaded from our database into an Animal object, and then displayed on the Animal Info Page.

**Navigating through the app:**

Navigation through the app is achieved primarily through buttons. We have implemented multiple pages to our app (such as home, search, animal info, etc.) and getting to each page is accomplished through button presses.

The main functions of our app follow button presses from the home page to the animal page. Home Page -> Search Page -> Animal Info Page. Each button loads the next page.

**Displaying Information:**

Information on the animals is displayed on the Animal Info page. This information is read from our database, and that info is loaded into TextView boxes setup on the Animal Info page. The TextView boxes are all predetermined and will automatically populate itself with the correct category of information for the current animal.

**Storing and Retrieving Information on Animals:**

Since our app is a database the core functionality will require us to access and display information. This is achieved by storing the animal information for each animal in separate text files located in the “assets” folder of the project. This information is formatted in a way that can easily be parsed into String variables for each animal, and then assigned to the correct TextView box so the correct category of information is displayed. Each animal will also have an image associated with it, these images will also be located in the “assets” folder.

**Animal Class:**

The animals have their information stored in the “assets” folder before load, and has the info loaded to the TextView boxes on the Animal Page after the user selects the animal. To get from the “assets” folder to the TextView boxes the information is first parsed and stored into String variables. A class called Animal is used to hold this information. The Animal class has code to read the info from the “assets” folder, and then hold that information in variables to be assigned to TextView boxes. The code used to read the info is implemented using a BufferedReader. This reader will continue to read lines from the text document, and store each set of information into the appropriate Animal instance variable.