Design Document for WingWatcher

Group MS 317

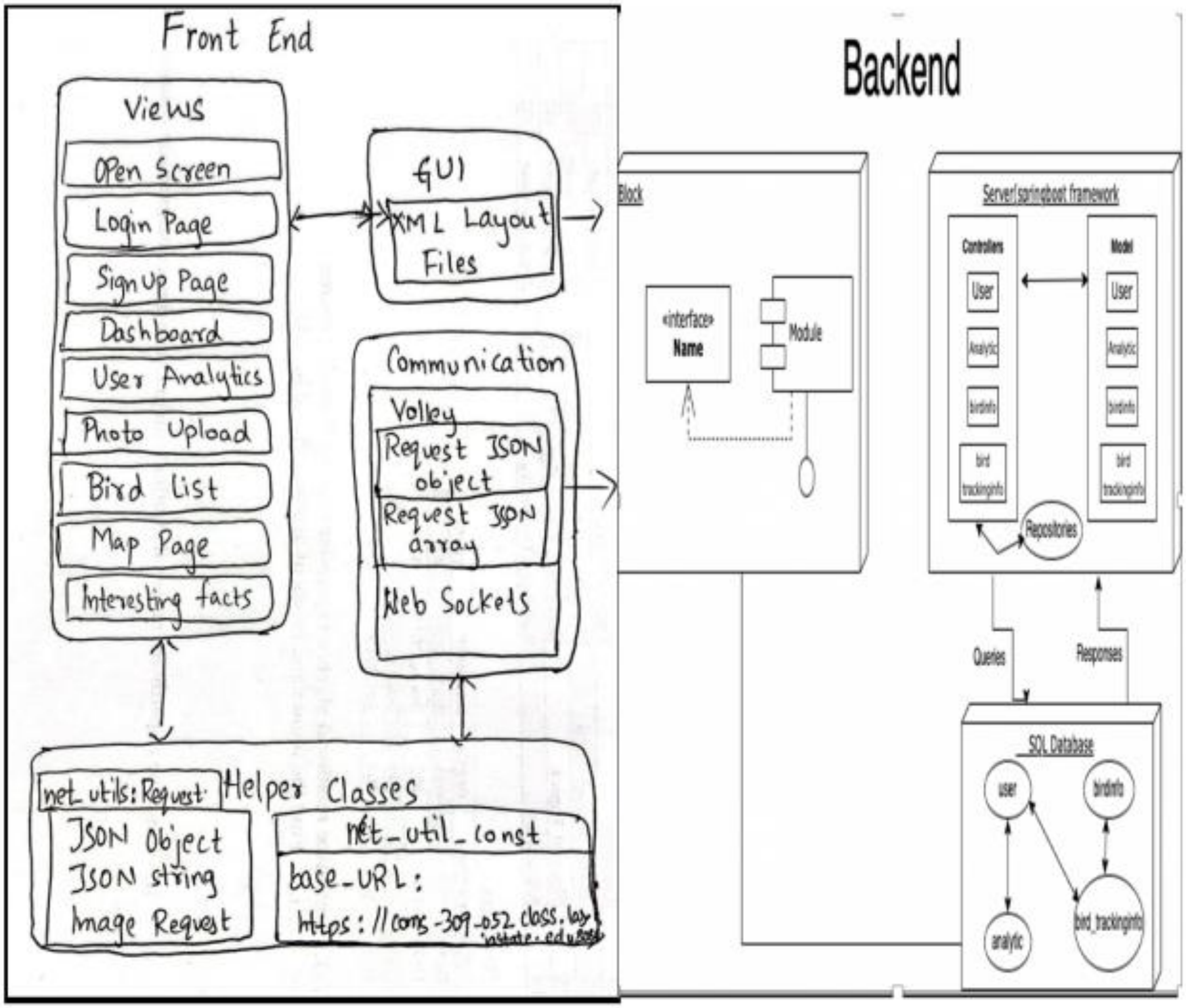
Brian Xicon: 25% contribution

Owen Kim: 25% contribution

Andrew De Gala: 25% contribution

Siddhartha Gudipudi: 25% contribution

Block Diagram



Block Diagram Description

Frontend:

Description / Functionality of the app: When the user starts the app, they are presented with the splash screen, followed by the login page; if they don't have an account, they can click the signup button there. If the user already has an account, they can click the login button and use the app. If clicked signup, the user gets navigated to the signup page and can create an account. Once the user has successfully logged into the app, the user will be navigated to the dashboard, which includes all the pages and cards for each. Once the user is on the dashboard page, they can click on the cards they want to go to, and upon clicking, the user is navigated to that page. We also have a navigator drawer. The navigation drawer includes home, app info, list of privileges, change password, and logout. If clicked logout, the user gets logged out and will be navigated back to the login page. To change the password, the user sees a pop-up for entering a new password twice to authenticate. Coming to the six cards we have, one is the user analytics, which gives information on the analytics and counts of the birds. The next one is the photo upload thing, which allows the users to upload a picture from the photo gallery, and then the information of the bird is displayed. The third one is the bird list page, where all the birds are displayed as a list, and there is also a search bar at the top so that the users can search for a particular bird they want to scout. Upon searching for the birds, there is also a learn more button that takes the users to the bird info page that gives the total information about the bird, and users can read about the bird they wish to. The next is the map page, where we integrated the Google Maps API, and there is a map that also shows the markers and names of the birds marked. The markers also have a button to make users navigate the bird list page and search for the birds they want. The next card is the interesting facts, with a button to generate the facts. When clicked on that button, the facts are displayed from there.

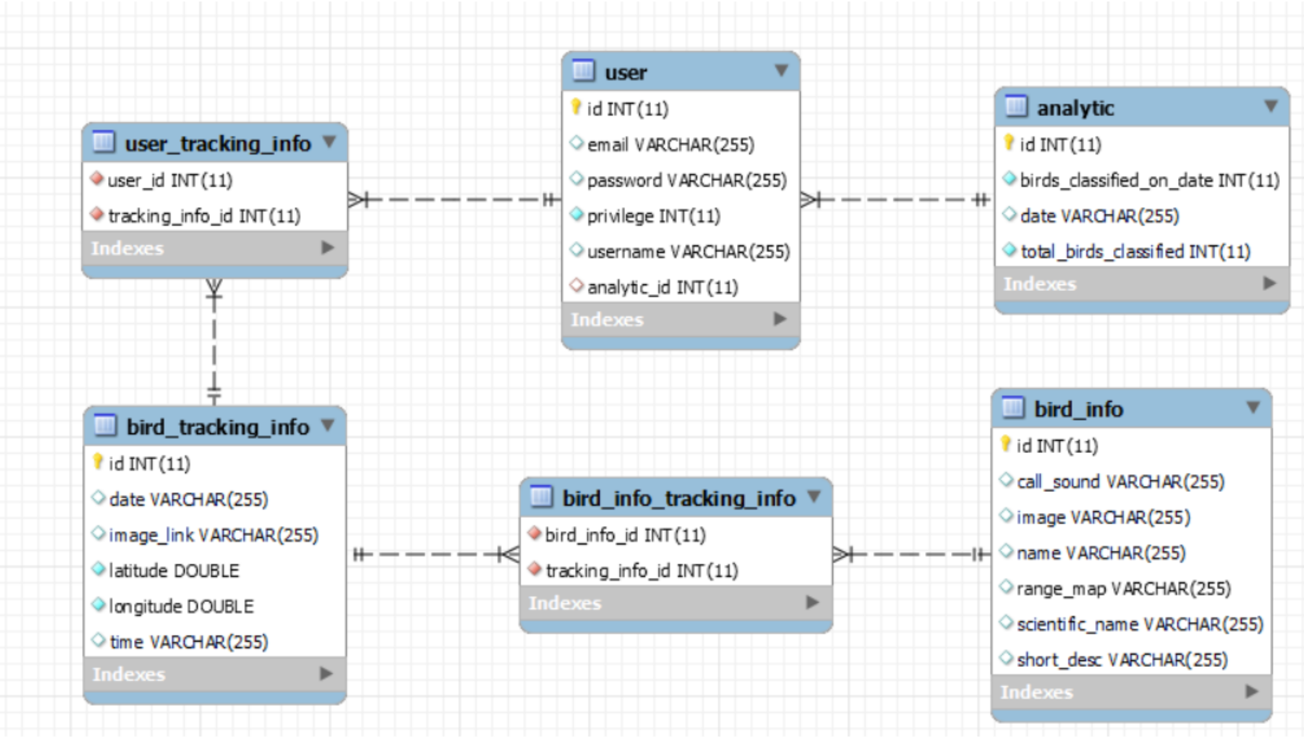
Server:

We utilize Spring Boot to act as our server. The server contains three core components: models, controllers, and repositories. Models are Java classes representing our database tables which are equipped with many essential functions like constructors, getters, and setters. Next, the controllers serve as the messenger between the actor and the server, this allows the actor to talk to the server which can then talk to our database to retrieve or manipulate various aspects of our database. Lastly, the way our server talks to our database is through the repositories. Which makes it easy for our CRUD commands to be converted to MySQL queries. An example of a typical call from the frontend to create a user would first be sent to the user’s controller which would then make a call to the model to create the variables needed with the data given, the repository will then allow the server to communicate with the database to save the added information.

Database:

Our database contains the tables of all the information we store. We have four tables altogether. These include the User, Analytic, Bird Info, and Bird Tracking Info tables. Our database acts as the main source of data for our application. We are using MySQL as the basis for our database. MySQL is a relational database that supports client-server architecture. With the relational database, we can have relationships between our tables, including a many-to-many relationship between Bird Tracking Info and User/Bird Info. User also has a one-to-one relationship with Analytic. This allows all our data to be connected to access information much more efficiently.

Database Schema


user_tracking_info
user_id INT(11)
tracking_info_id INT(11)
Indexes
user
id INT(11)
email VARCHAR(255)
password VARCHAR(255)
privilege INT(11)
username VARCHAR(255)
analytic_id INT(11)
Indexes
analytic
id INT(11)
birds_classified_on_date INT(11)
date VARCHAR(255)
total_birds_classified INT(11)
Indexes
bird_tracking_info
id INT(11)
date VARCHAR(255)
image_link VARCHAR(255)
latitude DOUBLE
longitude DOUBLE
time VARCHAR(255)
Indexes
bird_info_tracking_info
bird_info_id INT(11)
tracking_info_id INT(11)
Indexes
bird_info
id INT(11)
call_sound VARCHAR(255)
image VARCHAR(255)
name VARCHAR(255)
range_map VARCHAR(255)
scientific_name VARCHAR(255)
short_desc VARCHAR(255)
Indexes