

---

## Design Document for FindFiesta

---

Group [MU-31](#)

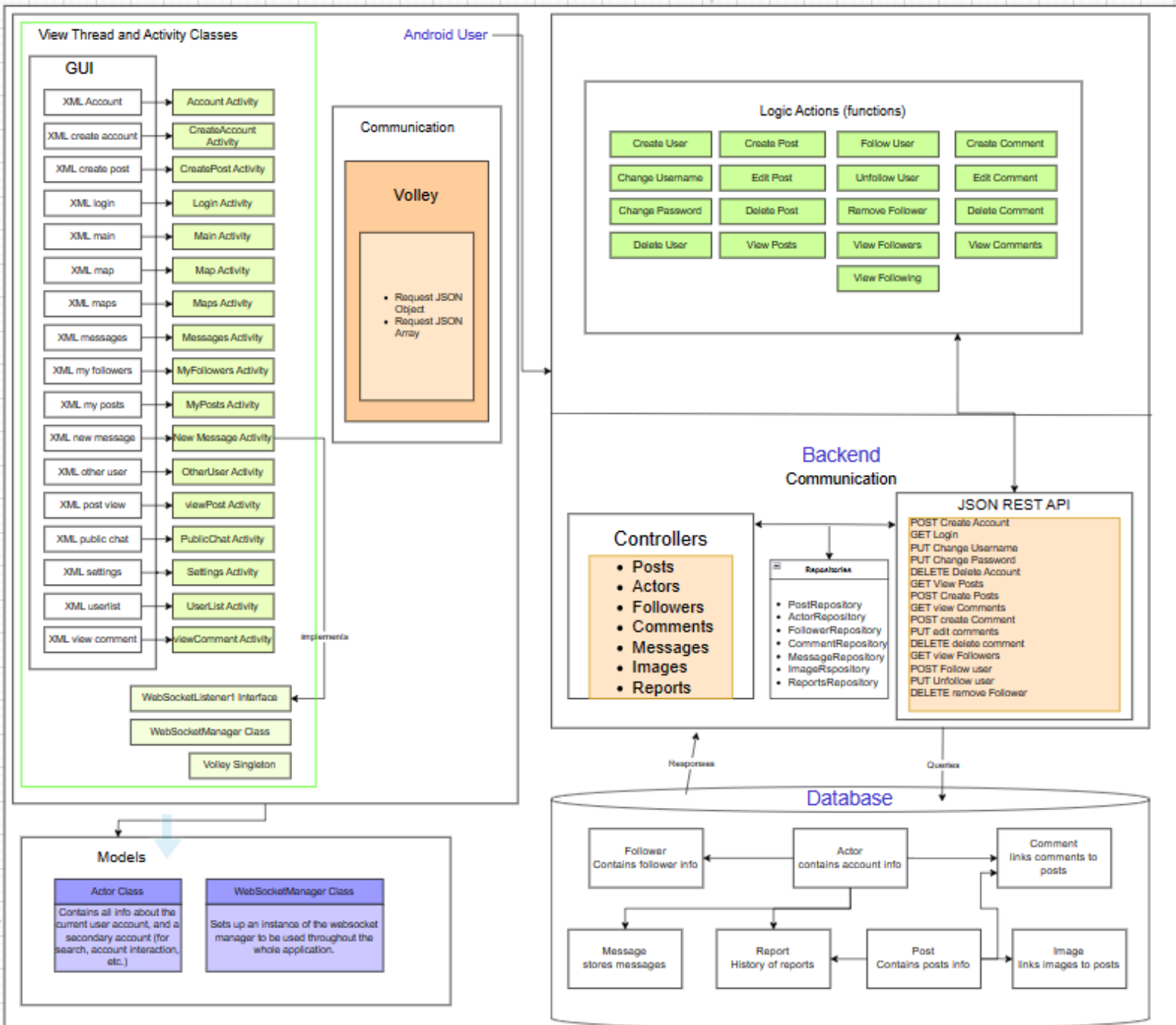
Kaitlyn Hoyme: 25% contribution

Omran Shareef: 25% contribution

Juan Figueroa: 25% contribution

Christopher Moseley: 25% contribution

# Block Diagram



# Design Description

## Frontend (currently implemented)

### CreateAccountActivity (user)

- Create account generates a page with the following elements
  - EditText: usernameEdit
  - EditText: emailEdit
  - EditText: passwordEdit
  - Button: CreateAccount
- Upon clicking the button 'CreateAccount' the values of the usernameEdit, emailEdit, and passwordEdit are sent as a POST request to the server. Similar examples happen when creating posts and comments on their respective pages.

### MainActivity (user)

1. Main activity generates a page with a listView called postList. When the page is generated it calls a JSON array request to get all posts by post ID
  2. Inside the JSON array request it calls a JSON object request to get the title and description of each post and puts the information into an arrayList
  3. The tiles of the listView then have their corresponding title and description attached and when they are clicked takes you to a separate view for individual posts
- A similar case happens when viewing an individual post with comments for said post

### viewPostActivity, viewCommentActivity (user)

When choosing to edit either a post or a comment the page hides and disables all the boxes displaying the items information. When submitting the updated information/details the page refreshes to display the now current information.

## Backend

### Communication

Mappings are used to update data using requests and responses.

- POST: sends info to the database such as creating an account (new user) or creating a post, etc.
- GET: retrieves info from the database, such as viewing followers, viewing posts, comments, etc.
- PUT: changes info in the database such as updating usernames, updating passwords, editing comments, etc.
- DELETE: gets rid of info in the database such as deleting accounts, deleting posts, deleting comments, etc.

### Controllers

Handles all the REST API requests, controls the interaction between frontend and backend.

- Actor - Contains the above mappings to create users, which contains one-to-many relationships with the below post, comment, follower, message and
- Post - Keeps track of all the contents of the posts like title, type, location, notes. It is connected to the images table to keep track of the posted images, as well as actor and comment to see who created the post and commented on it.
- Follower - Has a many to many relationship with actors to keep track of who is following who.
- Comment - Contains the above mappings to create posts with a one to many relationship with actors and post, users can comment other posts
- Message - many to many relationships with actors, any user can message any other user several times.
- Image - Every post can have many images, it is implemented with a one to many relationship with post.
- Report - Users are able to report other posts but only moderators can delete the post or reports after adding their justification.

## Tables and Fields

