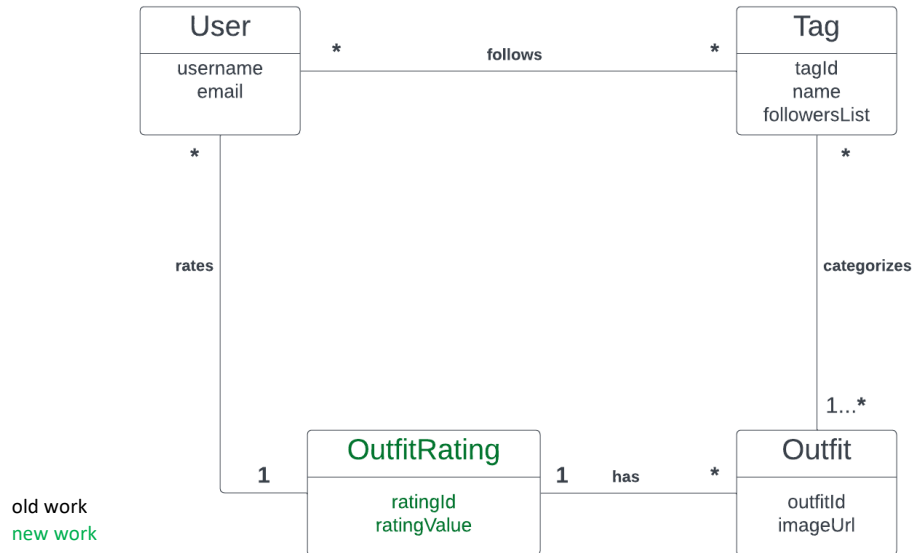


Deliverable – Sprint 2

1. Domain Model:



User: This entity represents a person who uses the application.

- A User can *rate* zero or more Ratings, hence “*” cardinal relationship.
- A User can *follow* zero or more Tags, hence “*” cardinal relationship.

Tag: This is a label that can be followed by zero or more users and can categorize one or more outfits.

- A Tag can *categorize* zero or more Outfits, hence “*”
- A tag can be *followed* by zero or more Users, hence “*”

Outfit: An ensemble that can be categorized by one or more tags and can have zero or more ratings associated with it.

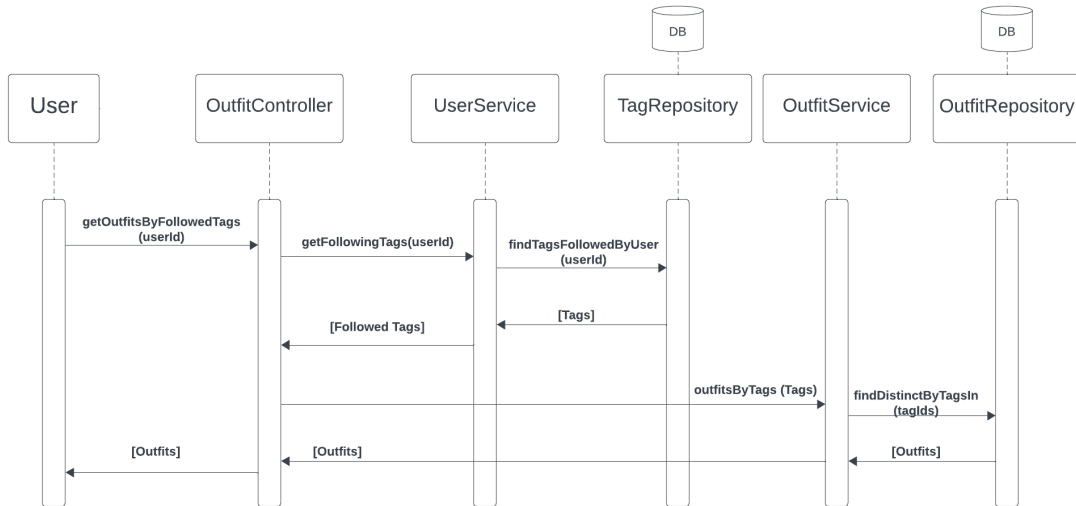
- Every outfit can have zero or more ratings, hence “*”
- Every outfit can be *categorized* with at least one tag, hence “1…*”

OutfitRating: A rating given by a user to an outfit. An outfit can have multiple ratings, but each rating is associated with exactly one outfit.

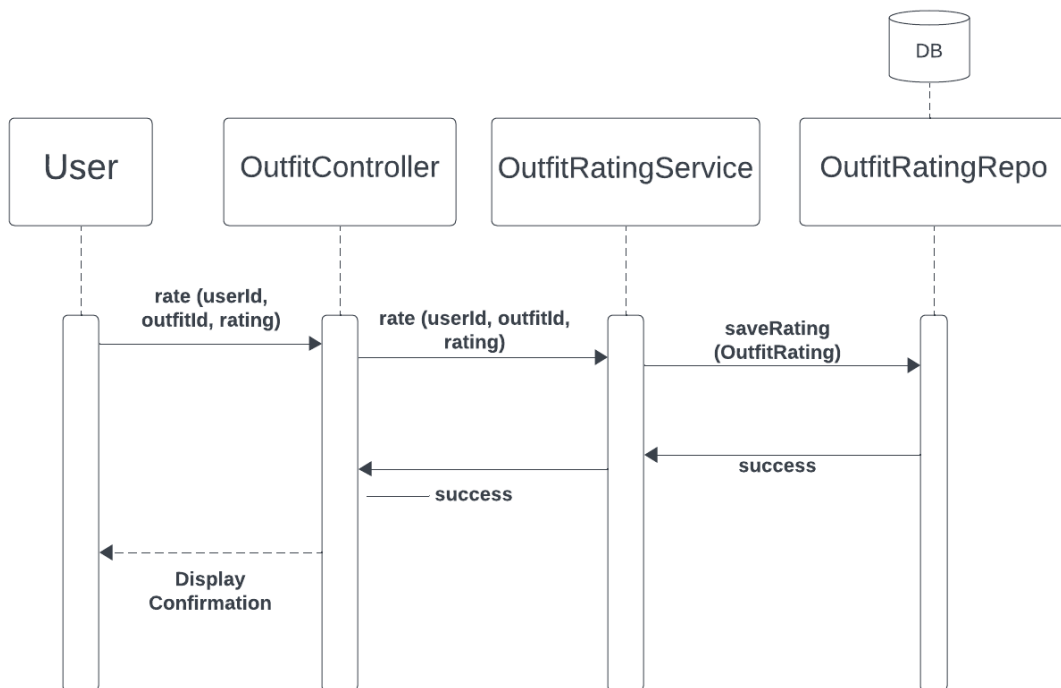
- Each OutfitRating is associated with one single user & single outfit, hence “1” cardinal relation with both

2. Interaction Diagrams

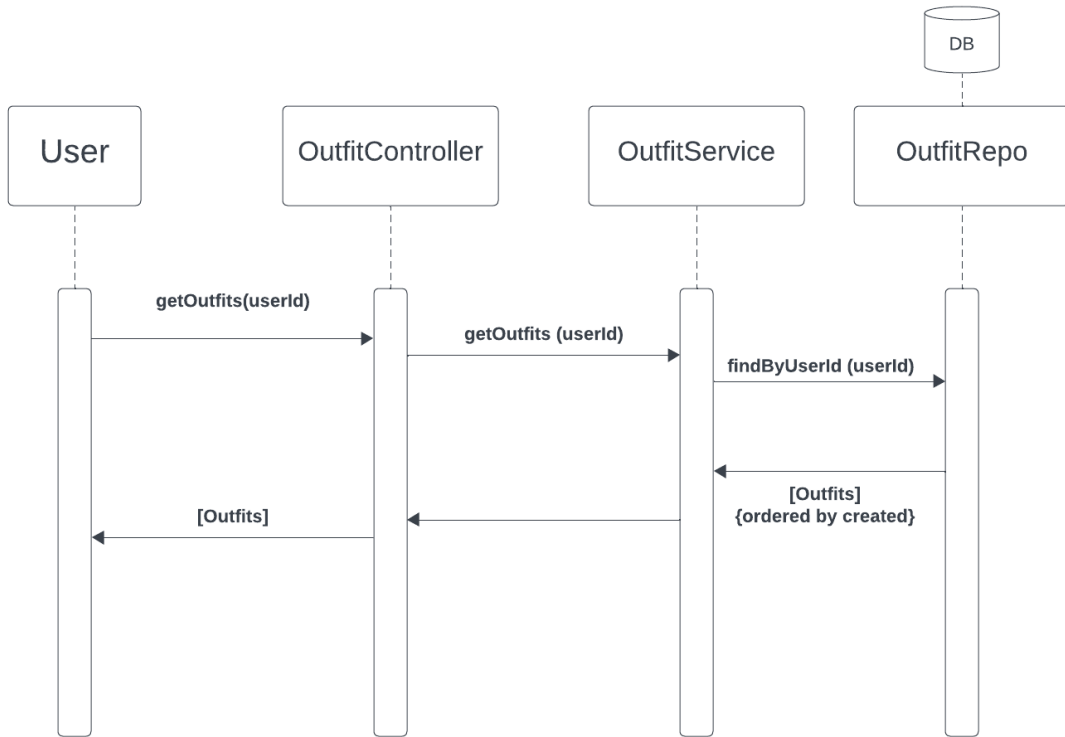
2.1 Interaction / Sequence Diagrams – Feed Page



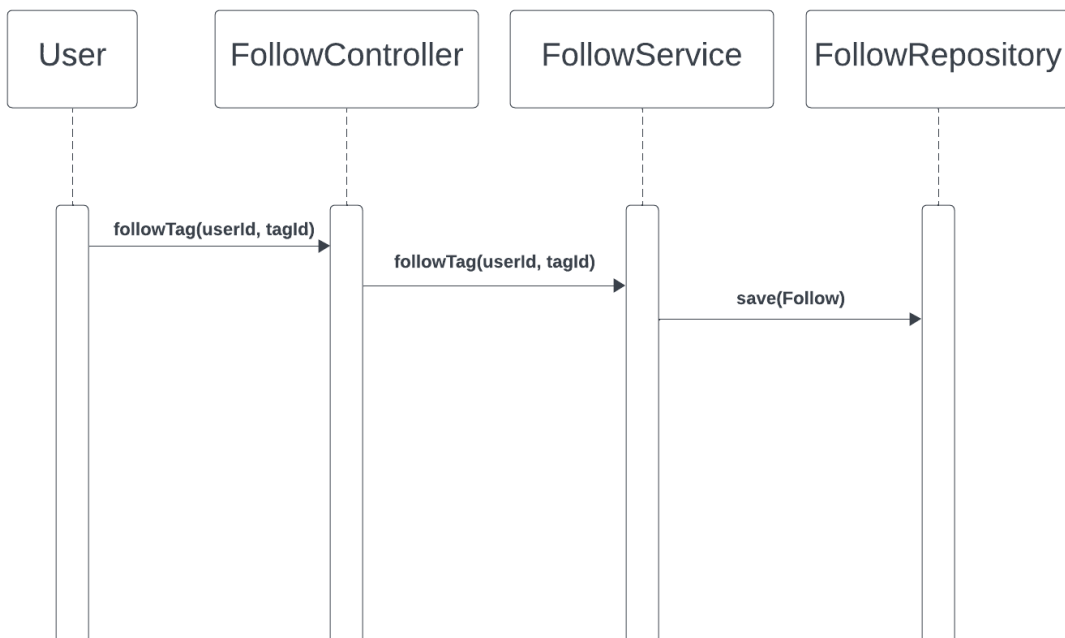
2.2 Interaction / Sequence Diagrams – Rate Outfit



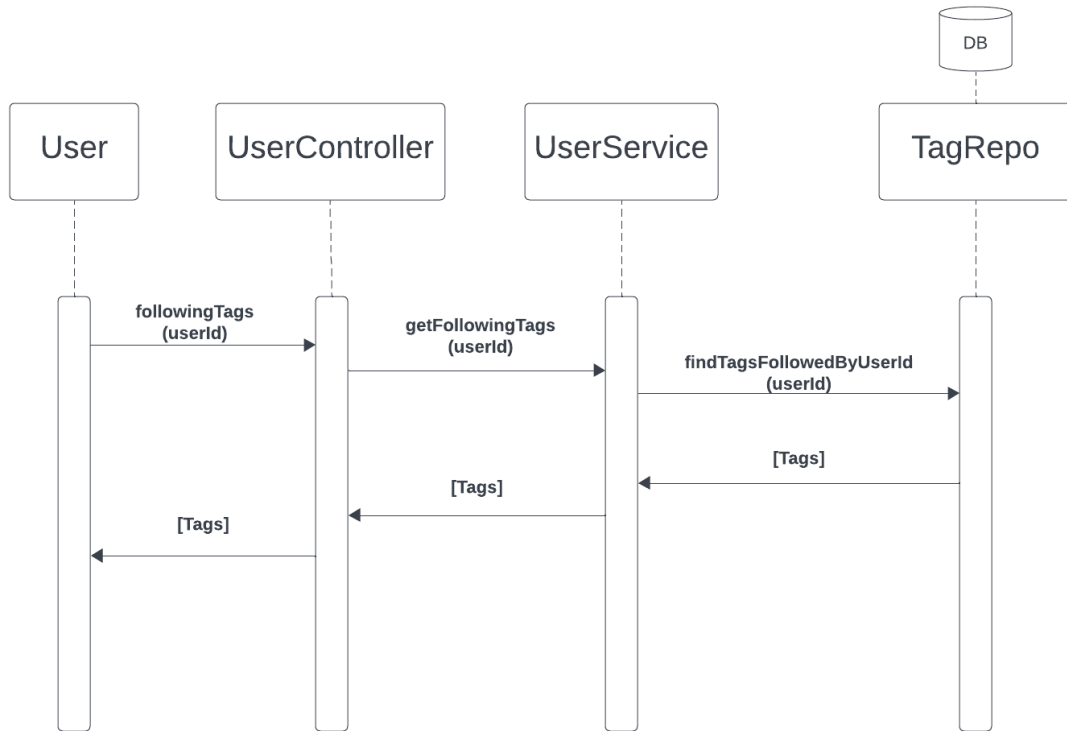
2.3 Interaction / Sequence Diagrams – Profile Page



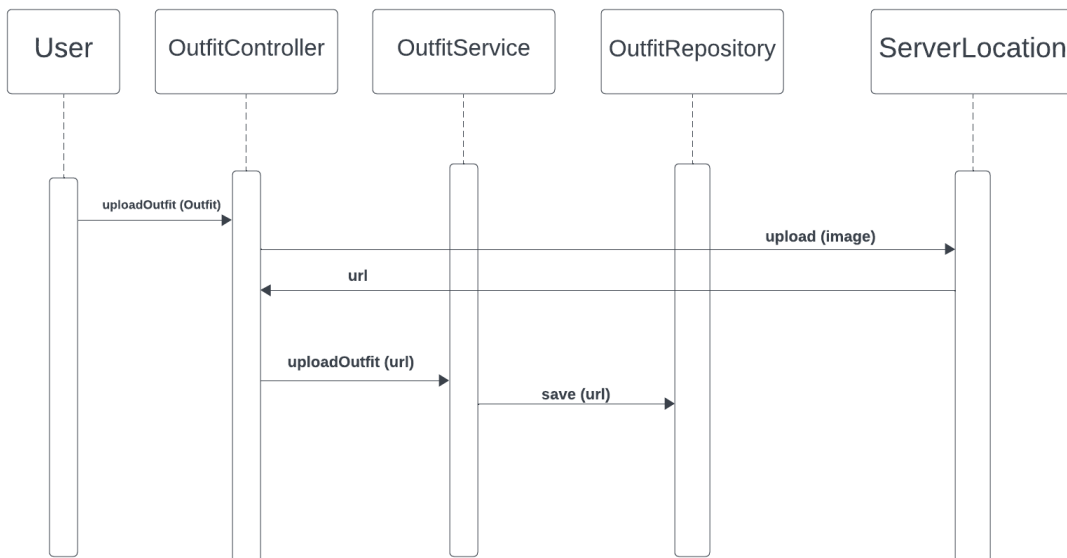
2.4 Interaction / Sequence Diagrams – Un/Follow Tag



2.5 Interaction / Sequence Diagrams – Not / Following

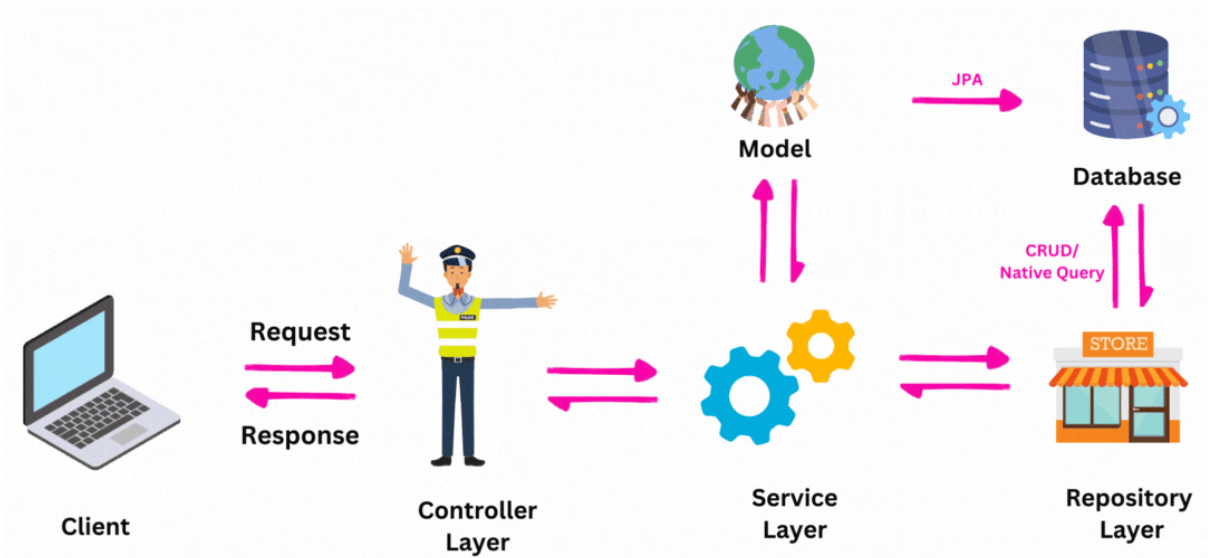


2.6 Interaction / Sequence Diagrams – Upload Outfit



3. Class Diagrams

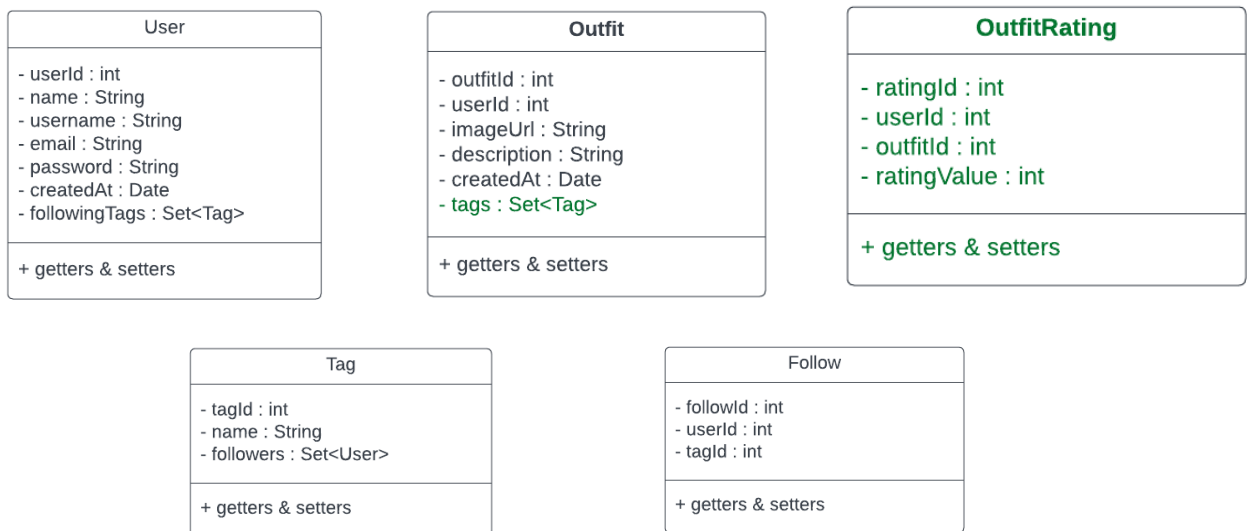
3.1 Spring – Boot High Level Architecture



source

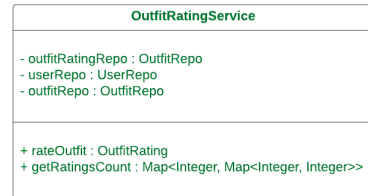
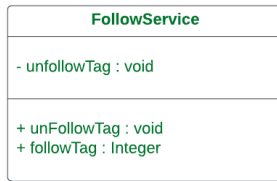
- This architecture primarily drives the notion of class diagrams behind our project.
- Like the above image, our classes are bifurcated into Model, Controller, Service & Repository

3.2 Model Classes – Entities

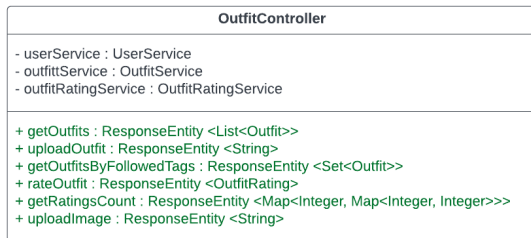
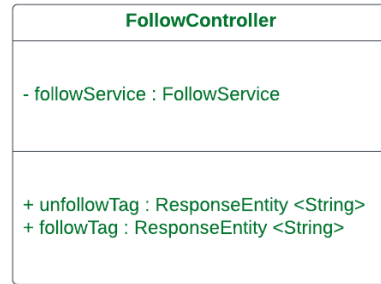
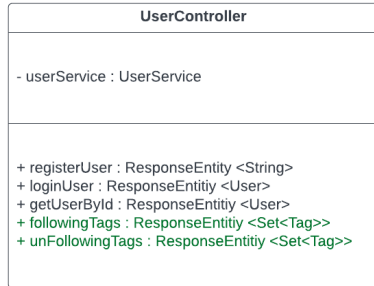


3.3 Service Layer

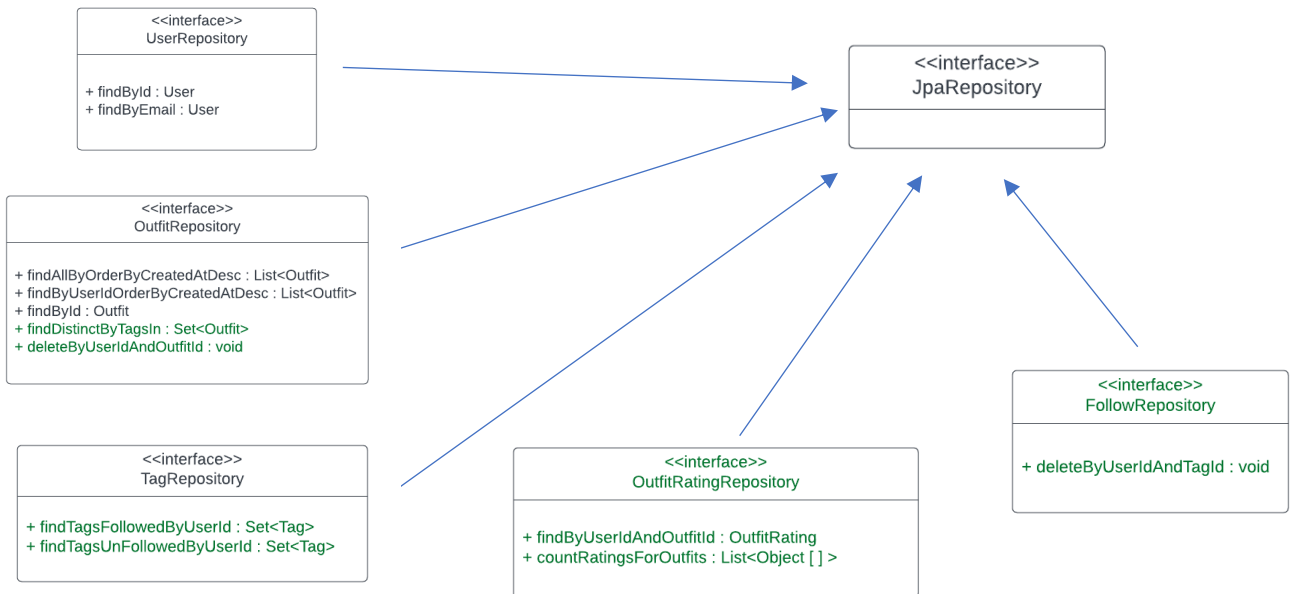




3.4 Controller Layer

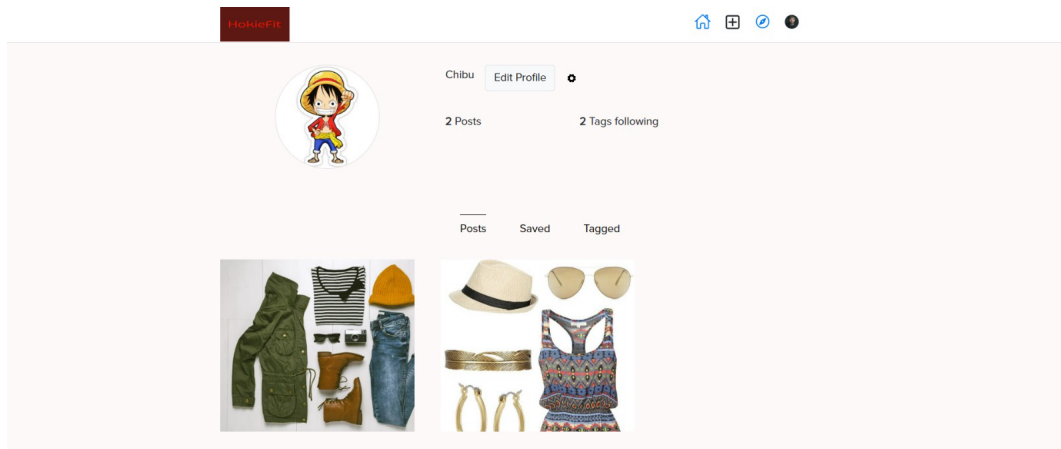


3.5 Repository Layer

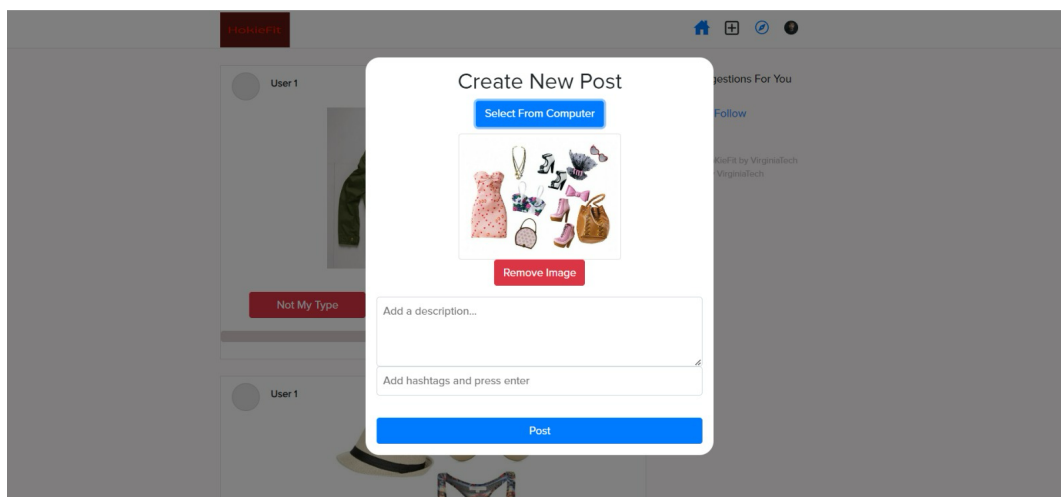


4. Demo

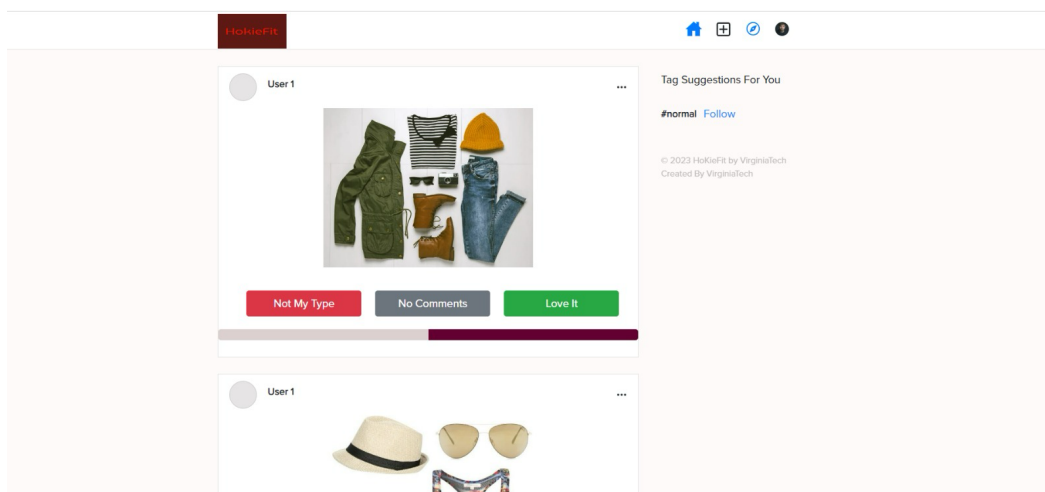
4.1 Profile Page



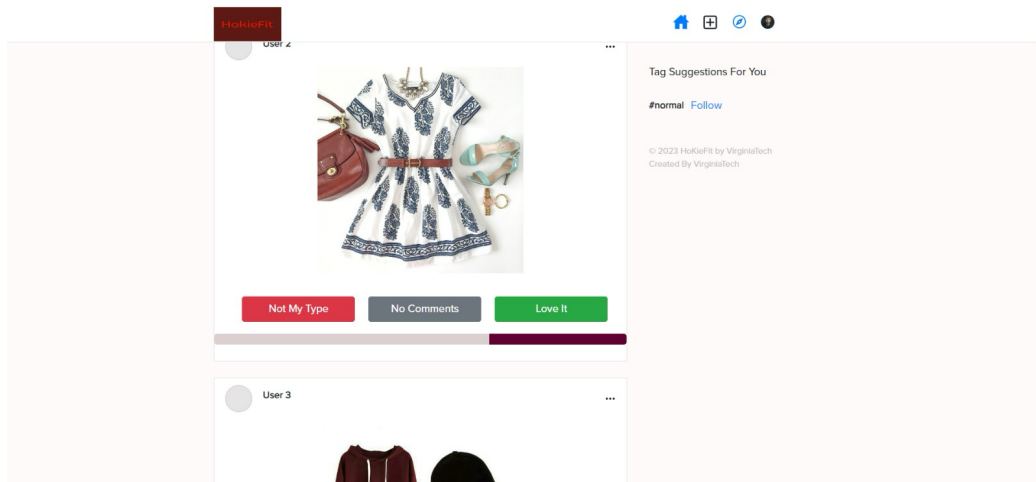
4.2 Upload Page



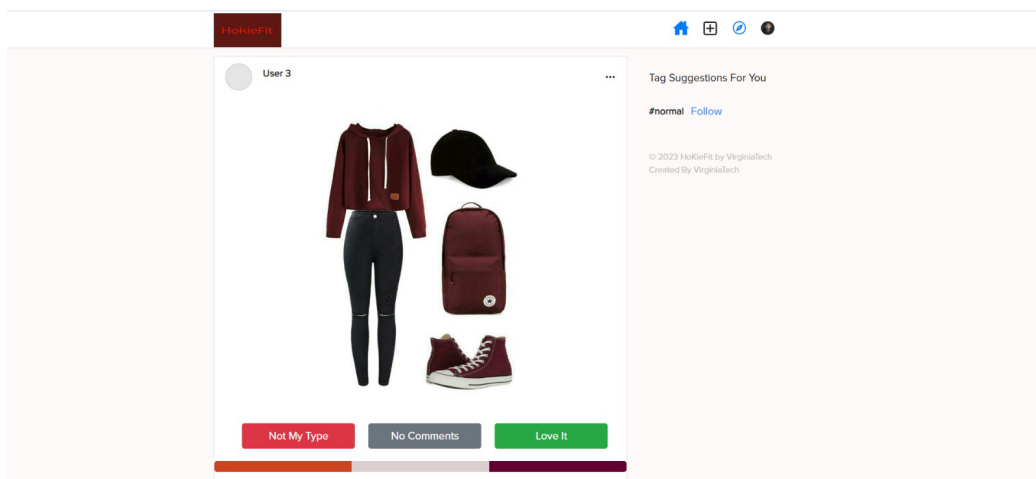
4.3 Feed Page 1



4.4 Feed Page 2



4.5 Feed Page 3



5. Scrum

5.1 Sprint 2 Backlog

5.1.1 "Feed" page

- *UI Design for Feed*
 - Assignee: Sai Pavan Bathala
 - Tasks:
 - Design "Feed" page UI - Estimated Time: 3 hours.
 - Implement "Feed" page React components - Estimated Time: 3 hours.
- *Backend for Feed*
 - Assignee: Sagar Atla
 - Tasks:
 - Develop endpoints for retrieving feed outfits - Estimated Time: 5 hours.

- Implement logic for personalized feed based on followed tags - Estimated Time: 4 hours.
-

5.1.2 Rate Outfits

- *Outfit Rating Feature*
 - Assignee: Mohith Kamanuru
 - Tasks:
 - Develop rating API endpoints (like, neutral, dislike) - Estimated Time: 3 hours.
 - Implement rating logic in the service layer - Estimated Time: 4 hours.

5.1.3 "Profile" page

- *Profile Page Development*
 - Assignee: Srikanth Karri
 - Tasks:
 - Design profile page UI - Estimated Time: 5 hours.
 - Implement profile page displaying user outfits - Estimated Time: 4 hours.

5.1.4 Following / Not Following Tags (Added New)

- *Tags Management Backend*
 - Assignee: Aruj Nayak
 - Tasks:
 - Create endpoints to display following/not following tags - Estimated Time: 3 hours.
 - Implement logic to manage user tags relations - Estimated Time: 3 hours.

5.1.5 Follow Tag / Unfollow Tag (Added New)

- *Tag Follow Feature*
 - Assignee: Sai Pavan Bathala
 - Tasks:
 - Design UI component for follow/unfollow tags - Estimated Time: 3 hours.
 - Implement follow/unfollow functionality on the frontend - Estimated Time: 3 hours.

5.1.6 (Backlog – sprint 1) Upload Outfit using a window

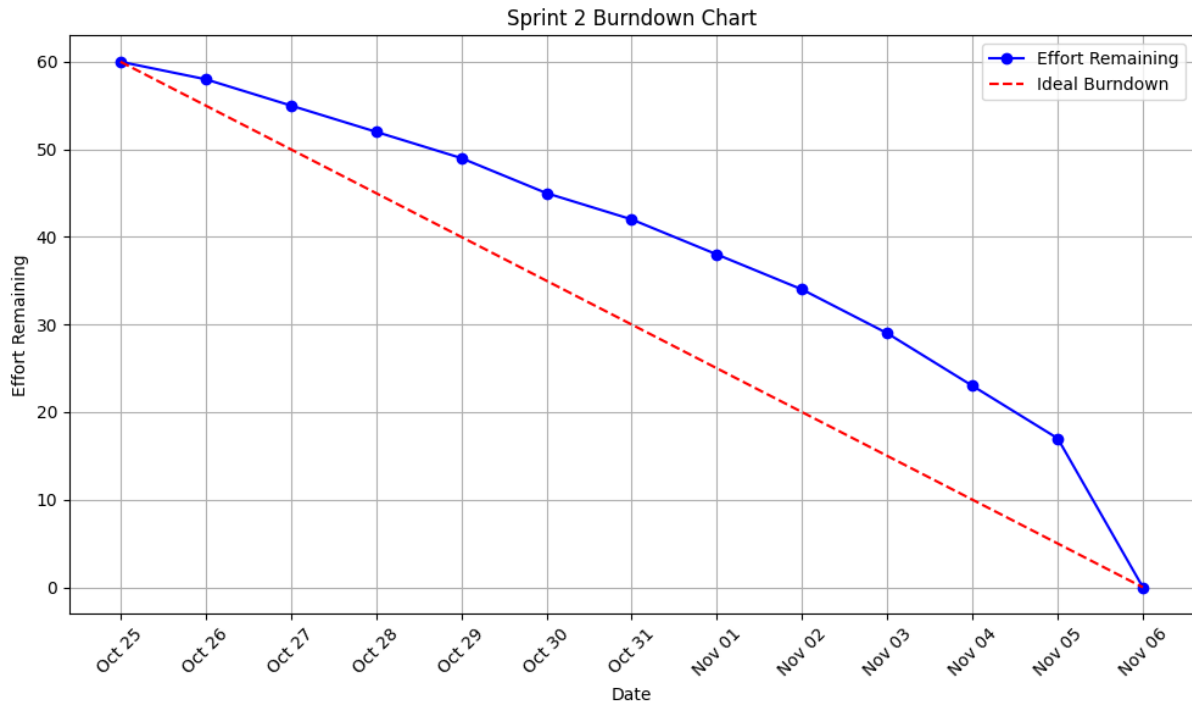
- *Upload Outfit Backend and UI*
 - Assignee: Sai Pavan Bathala
 - This user story was part of the backlog and is addressed in this sprint. Estimated Time: 6 hours.

5.1.7 Delete Outfit (Added New)

- *Outfit Deletion Functionality*
 - Assignee: Sagar Atla

- Tasks:
 - Develop endpoint for deleting outfits - Estimated Time: 2 hours.
 - Implement deletion logic on the backend - Estimated Time: 2 hours.

5.2 Burndown Chart



5.3 Sprint Retrospective

Accomplishments:

- Implemented "Feed" page which shows outfits based on followed tags.
- Added "Rate Outfits" feature allowing users to like, stay neutral, or dislike outfits.
- Created "Profile" page showcasing user's own outfit uploads.
- Developed functionality for users to follow or unfollow tags.
- Completed the backend work for the "Upload Outfit"
- We have been identify more important stories than what we had planned ultimately giving us a better product

Impediments:

- Incomplete UI for upload outfit and showing tags on a post
 - Difficulties in storing images and retrieval in front-end hence pushed to next sprint backlog
- We encountered some challenges coping with making the report, presentation, front-end and backend all in a due short sprint time.

Addressing Impediments:

- We are referring to Spring Boot documentations and YouTube tutorials to fix

What Went Well:

- Team collaboration and communication improved significantly
- We were able to coherently code front-end and backend together which reduced waiting time from both end substantially
- Adherence to best coding Spring Boot practices has simplified extending new functionalities
- Sprint planning was effective, leading to a clearer understanding of tasks

Improvements:

- Prioritize setting up automated testing to catch issues early.
- Allocate time for research and spike tasks to handle unforeseen technical challenges.

5.4 Product Backlog Updated

Newly Identified Stories [Please refer to sprint backlog for more information]

- Following / Not Following Tags (Added New)
- Follow Tag / Unfollow Tag (Added New)
- Delete Outfit (Added New)

These stories have been identified very important in the due process of developing such an application.

Dropped Stories

- Notifications on Likes and Saves
 - This user story has been dropped and will not be addressed in the scope of the project
 - This requires a huge backend amendment to include attributes like sessions, seen and new entities for notifications and probably cannot be addressed in such a short time frame

5.5 Sprint 3 Backlog

1. *Complete UI for showing Tags on an Outfit (carried from Sprint 2)*
 - Subtask: Design tag display layout on the frontend (UI) - 3 points
 - Subtask: Implement the API to retrieve tags for each outfit (Backend) - 3 points
 - Subtask: Integrate tag display functionality in the frontend - 2 points
2. *Complete UI bug fixing while uploading Outfit (carried from Sprint 2)*
3. *Face Detection API - ML*
 - Subtask: Research and choose a face detection library or service - 5 points
 - Subtask: Implement a face detection service (Backend) - 8 points
 - Subtask: Integrate face detection with the outfit upload process - 5 points
4. *Anonymous Username Generation (New)*
 - Subtask: Develop algorithm for generating unique anonymous usernames - 3 points
 - Subtask: Implement username generation in the account creation process - 3 points

5. *Search Functionality/Page (Big Story)*

- Subtask: Design search functionality UI - 3 points
- Subtask: Develop API for searching outfits (Backend) - 5 points
- Subtask: Integrate search API with frontend - 3 points

6. *Testing*

- Subtask: Write unit tests for new backend functionalities - 5 points
- Subtask: Conduct integration testing for the new search and tag features - 5 points
- Subtask: Perform usability testing on the updated UI - 3 points