SW Engineering CSC 648/848-04

Team 04
Leiyi Gao: Team Leader
Justin Mao: Backend Leader
Yinyin Wu: Scrum Master
Michael Han: Git Master
Nicholas Hamada: Frontend Leader

Milestone 2

10/17/2022

Project Name: RateMyResume

Revision History:
10/17/2022 - 1st draft

1. Data Definitions V2

Entity 1: **User**
Items: **Each user will have a unique account ID which will be linked to any resumes they post and linked to an email provided by the user which is stored in the mongo db database.**
Usage: **authentication, post and comments identification**
Privilege(s): **basic user**
Attributes: **userid, username, creation date, email**

Entity 2: **Post**
Items: **Each user can post their resume, and the data is also stored in the MongoDB Database. The resume is also linked with that user, as well as user's authentication such as user name, user email etc. Each user can comment on other users' resume, it could be suggestions, error checking or give out any specific current market information associated with the resume. All this information is also stored in the MongoDb database.**
Usage: **Contains resume, get feedback from other users through comments, templates that can be used by other users**
Attributes: **Postid, title, description, date, author, resume, comments, rating**

2. Functional Requirements V2

- **0001: Creating an account**
  - Description: Users should be able to create an account so they can share their resumes and comment on other resumes.
  - Priority: 1 – Must have
  - User stories: All three personas (John Smith, Jen Shing, and Jun Singh) will use this feature.
- **0002: Login**
  - Description: Users should be able to log in to their account after their account has been created.
  - Priority: 1 – Must have
  - User stories: All three personas will use this feature.

- **0003: Search**
    - Description: Users should be able to search for resumes based on tags or categories.
    - Priority: 3 – Opportunistic
    - User stories: All three personas will use this feature.
- **0004: Posting/uploading resumes**
    - Description: Users should be able to post or upload their resumes to the app. They should also be able to post their resumes anonymously.
    - Priority: 1 – Must have
    - User stories: All three personas will use this feature.
- **0005: Commenting**
    - Description: Users should be able to comment on each other's resumes to give feedback.
    - Priority: 1 – Must have
    - User stories: All three personas will use this feature.
- **0006: Likes/upvoting**
    - Description: Users should be able to like/upvote other people's resumes.
    - Priority: 1 – Must have
    - User stories: All three personas will use this feature.
- **0007: Deleting resumes**
    - Description: Users should be able to delete their own resumes after posting them.
    - Priority: 3 – Opportunistic
    - User stories: All three personas will use this feature.

3. UI Mockups and Storyboards

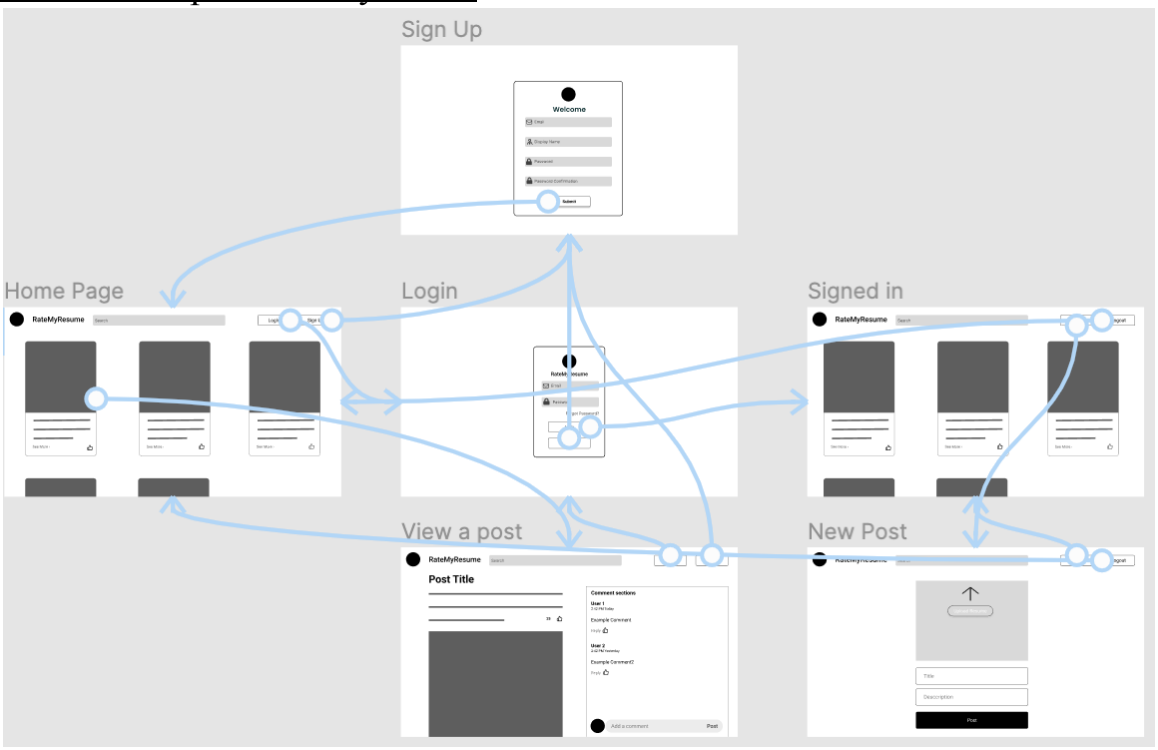

Figure 1. High-level storyboards

4. High level Architecture, Database Organization

- *DB organization*:

Table/collection 1
Title: User
Schema:

```
{
   "userid": { "bsonType": "objectId" },
   "username": { "bsonType": "string" },
   "email": { "bsonType": "string" },
   "privilege": { "bsonType": "double" }
}
```

Table/collection 2
Title: Post
Schema:

```
{
   "postid": { "bsonType": "objectId" },
   "author": { "bsonType": "string" },
   "creation_date": { "bsonType": "string" },
   "title": { "bsonType": "string" },
   "description": { "bsonType": "string" },
   "resume": { "bsonType": "string" },
```

```
  "ratings": { "bsonType": "double" },
  "comments": { "bsonType": "string" },
}
```

- *Add/Delete/Search architecture*:
  User: add user, delete user, update user username/email
  Post: add post, search post, update post comments, update post rating

- Internal APIs
  - **Signup**
    type: POST
    route: /user/signup
    backend function: createUser(email, username)
    responses: successful operation - status code 201, redirect to login
    failed due to duplicate email/username - status code 409
    failed due to server error - status code 500
  - **login**
    type: GET
    route: /user/login/
    backend function: login(username, password)
    responses: successful operation - status code 200, redirect to home
    failed due to invalid username/password - status code 401
    failed due to server error - status code 500
  - **create a post**
    type: POST
    route: /post/create
    backend function: createPost(user, date, resume)
    responses: successful operation - status code 201, redirect to post
    failed due to missing parameter - status code 400
    failed due to server error - status code 500
  - **view post**
    type: GET
    route: /post/view/{POST_ID}
    backend function: viewPost(postId)
    responses: successful operation - status code 200, redirect to post
    failed due to post not found - status code 404
    failed due to server error - status code 500
  - **comment**
    type: POST
    route: /post/{POST_ID}/comment
    backend function: viewPost(postId)
    responses: successful operation - status code 201, refresh page
    failed due to authorization issue - status code 401
    failed due to server error - status code 500
  - **upvote**
    type: POST

route: /post/{POST_ID}/upvote
backend function: upvote(postId, username)
responses: successful operation - status code 204
failed due to authorization issue - status code 401
failed due to server error - status code 500

- **search**
  type: GET
  route: /post/search/
  backend function: upvote(postId, username)
  responses: successful operation - status code 204
  failed due to authorization issue - status code 401
  failed due to server error - status code 500
- **delete post**
  type: POST
  route: /post/{POST_ID}/delete
  backend function: delete(postId, username)
  responses: successful operation - status code 204
  failed due to authorization issue - status code 401
  failed due to server error - status code 500

5. High Level UML Diagrams

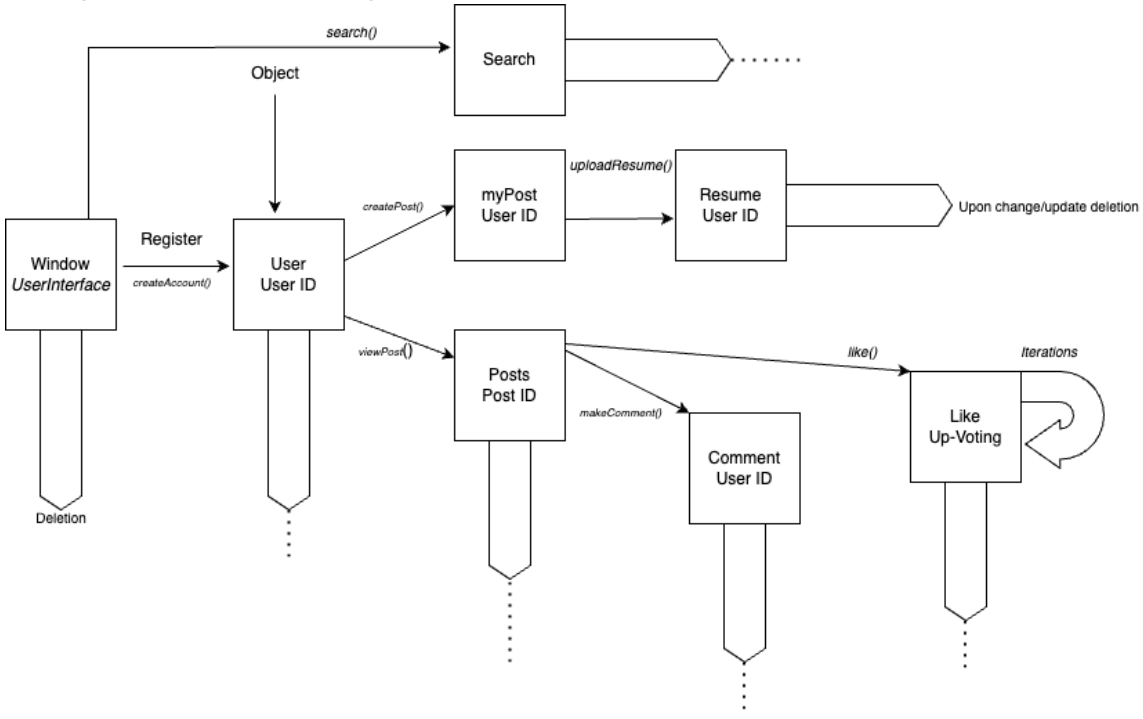

Figure 2. High-level sequence diagram

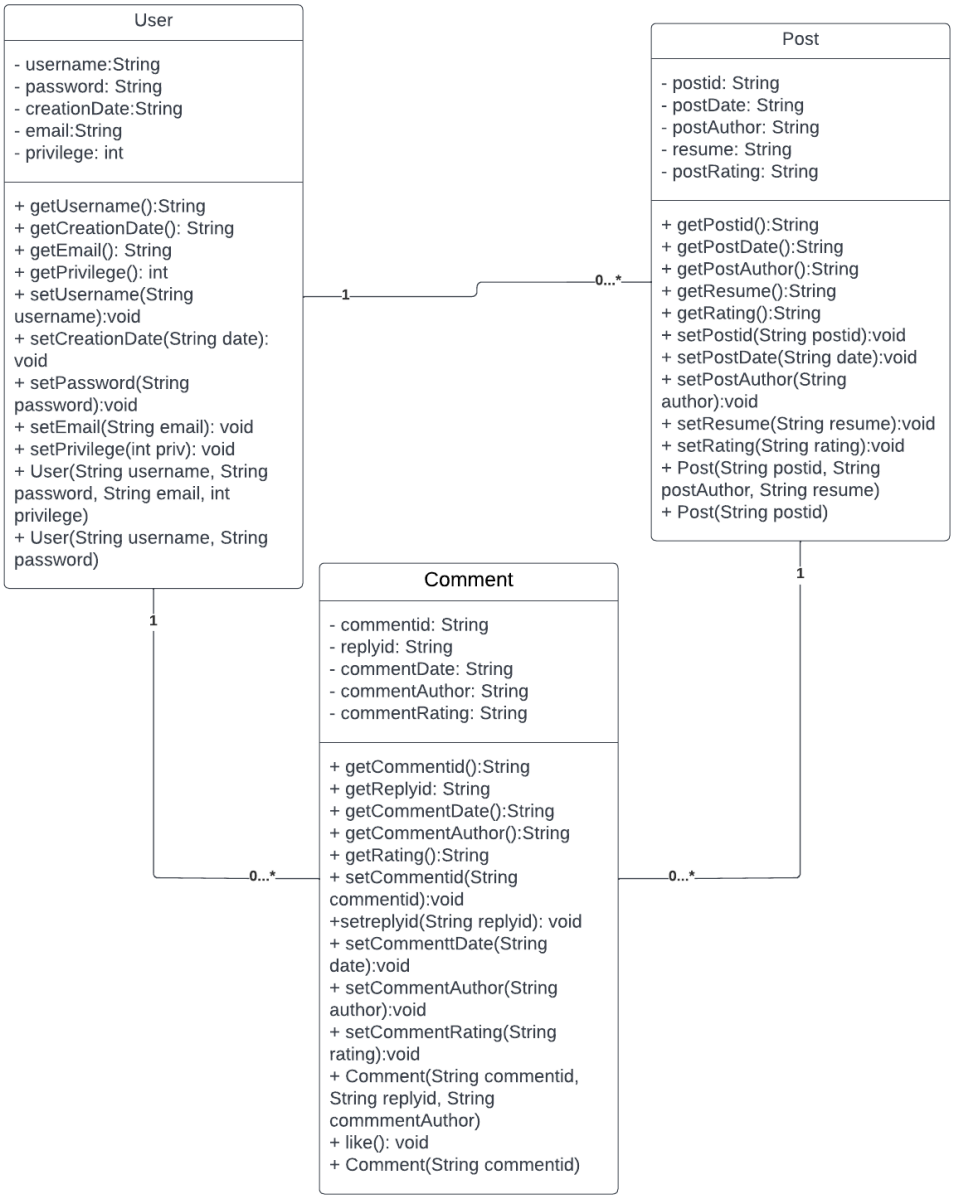

Figure 3. UML class diagram

6 .Key Risks

Schedule Risks: Schedule conflicts sometimes happen. Everyone will be notified on Discord and either the meeting is rescheduled or the person comes in late and tries to catch up with everyone. If there's an emergency, team members should be notified therefore an adjusted study plan can be made to deliver results on time.

Technical Risks: When researching a particular tool, technique, or concept takes longer than intended, the feature milestone may be pushed back. Plan ahead of time, browse over many resources before getting started, and feel free to exchange study resources in a team discussion.

Resolution up-to-date: We as a team always figure out a solution, we conduct the small meetings within our group to meet the task requirement, and the team members always communicate in discord if an emergency situation comes up.

7. Project management

Each member is on track for their process, and we share everyone's task within our team with 100% transparency. Our Scrum Master keeps us on page as a team. As far as right now, no one in the team has missed any meeting, and all team members show up in the meeting on time. Furthermore, we set up multiple small meetings within our team, such as 2 people working in a couple sessions.

We spread out our tasks into several subtasks in divide-and-conquer fashion to complete the main project. Originally, we were planning to use Atlassian's Jira, but we felt the tasks for Milestone 2 were too small that using project management tools might not be beneficial and could actually add additional workload instead.