

Boğaziçi University
Software Engineering MS Program

SWE 599 Project, Fall 2017
Supervisor: Suzan Üsküdarlı

Motive:
Community Information Processing Project
Interim Report

06.11.2017
Revision 1.0

By: Yasemin Alpay

Student Id: 2015719012

Table of Contents

1. Introduction	2
2. Software Requirements	3
3. System Design.....	5
3.1. Technical Overview	5
3.2. System Architecture.....	5
3.3. API List.....	6
3.4. Class Diagram	7
4. Work Done So Far – Future Work	8

1. Introduction

The purpose of this project to develop a community information processing web and mobile application. It aims to provide users to create and organize content and connect with each other in meaningful ways and to create information paths that can be utilized for more effective and enjoyable collective action.

After users are logged in to the system, they can create interest specific groups (interests) or subscribe existing ones. They can create content in these interests and share their knowledge about the content or interest in pre-defined ways.

Features of the application as the following:

1. Creating interests
2. Searching interests
3. Viewing interest details
4. Subscribing to public or private interests
5. Creating content in the interests
6. Searching existing contents
7. Displaying contents
8. Commenting to contents
9. Annotating the contents
10. Displaying the annotations
11. Having a profile page

2. Software Requirements

Functional Requirements

User Account Operations

1. User shall be able to provide their username, email and password while creating a new account.
2. User shall be able to login to their account using username and password.
3. User shall be able to recover their password.

Creating Interests

4. User shall create interest specific groups (interests) providing the title, description, privacy of the interest (public or private) and members. Titles shall be unique.
5. User shall be the admin of the interest that they have created.
6. Admin of the interest shall be able to make other members admin.
7. Admin of the interest shall be able to add new members to the interest or remove members from the interest.
8. Admin of the interest shall be able to change the privacy setting of the interest, so they can set the privacy to public or private.
9. If the admin change the privacy of the interest to private, contents of the interest shall directly become private or vice versa.

Searching Interests

10. User shall search interests by title.

Viewing Interests

11. User shall be able to view interest details (title, description, privacy, and subscription status).

Subscribing to Interests

12. Users shall subscribe to an interest after searching them.
13. User shall send a notification to the admin of the interest to subscribe, if the interest is private.
14. User shall be able to subscribe to the interest without permission, if the interest is public.

Creating contents in the interests

15. Users shall create text or image contents in the interests providing their title, description, and tags

Searching content

16. User shall search content by title or tags.

Displaying content

17. User shall view the details (title, description, tags, creator, and vote count) of the content if the interest of the content is subscribed by the user or the content is public.

Voting content

18. Users shall up vote or down vote the contents.

Commenting the content

19. Users shall comment the content if they can see the content.

Annotating content

- 20. User shall be able to add annotations to the content that they have created or the contents created by other users.
- 21. Annotations shall be in the format of text or image.

Displaying annotations

- 22. User shall be able to see the existing annotations of content.

User profile

- 23. Users shall have profile page.
- 24. Profile page shall display the name of the user, interests, contents, annotations, and comments created by user.
- 25. User profile shall be private or public.

Non-functional Requirements

- 1. The system shall expose a Rest API allowing interaction with other systems.
- 2. System shall be able to provide Android and Web client.
- 3. The system shall use open source libraries and tools.
- 4. Contents shall be annotated regarding to W3 Web Annotation Standards through a text or image using JSON-LD.

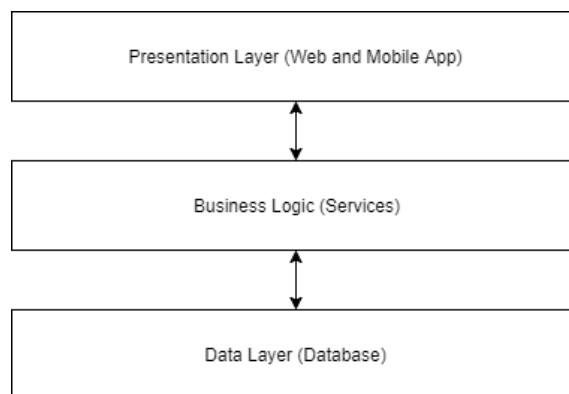
3. System Design

3.1. Technical Overview

Motive is a client-server application so it consists of a client side which is a mobile and web application and a server side exposing Restful API to be consumed by client side. Server (backend) side is on Spring Boot with MongoDB as NoSQL database.

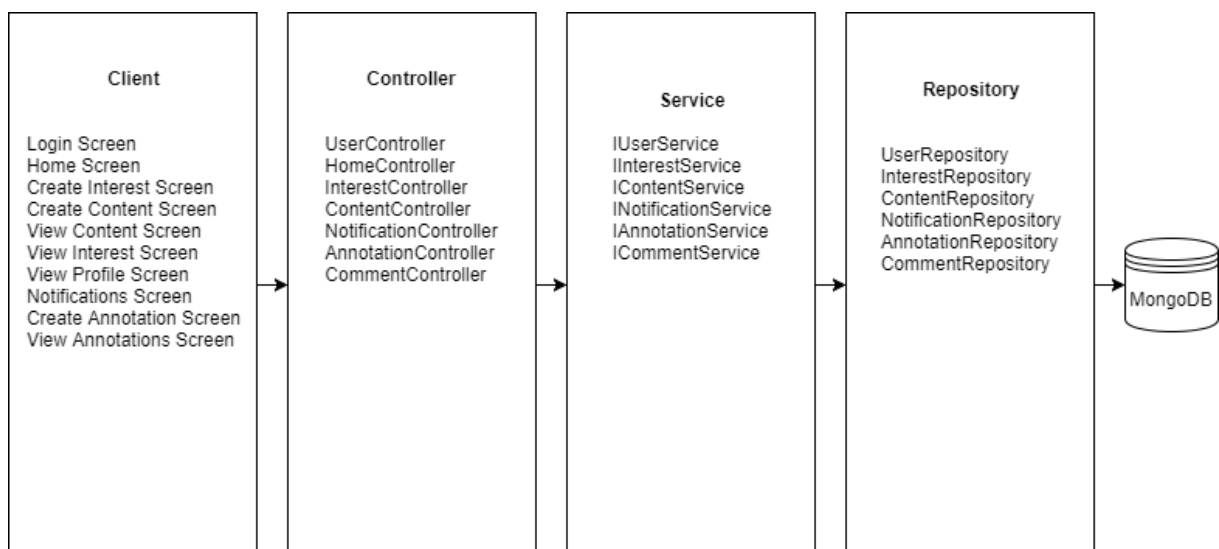
3.2. System Architecture

For architectural design, multi-tier architecture is going to be followed and depicted as the following:



Architectural Layer Diagram

Also, in implementation level, client will communicate to the server through the API (controllers in Spring Boot application) and controllers will manipulate data in the database with repository and service layer. Service layer provides abstraction in case of any change in database-specific repository layer:



Structure Diagram

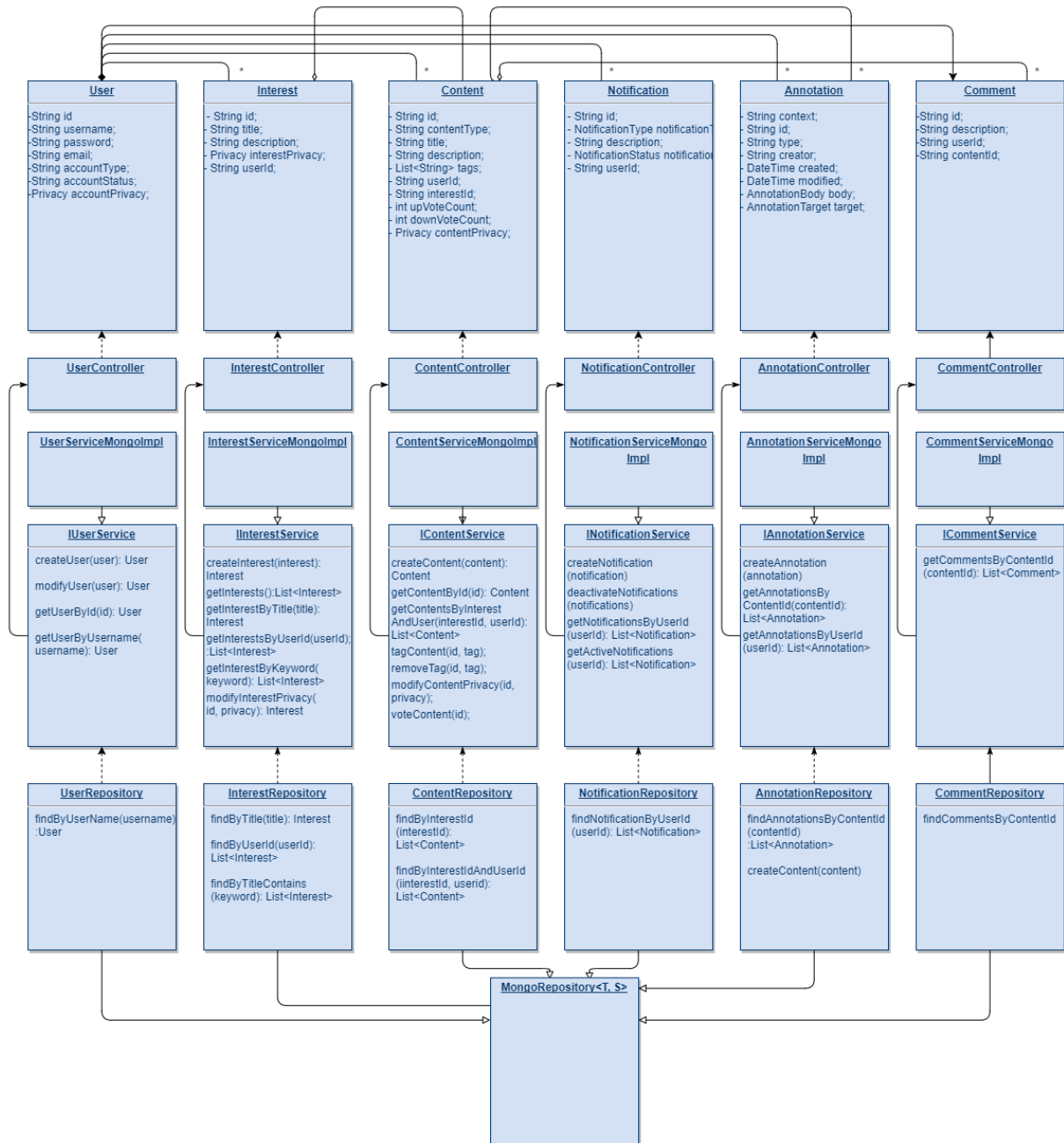
3.3. API List

API endpoint list to be exposed for usage of client side as the following:

- Create user
- Get user by username
- Create content
- Get content by ID
- Get content by interest and user
- Tag content
- Remove tag from content
- Up vote content
- Down vote content
- Modify privacy of content
- Create interest
- Get all interests
- Search interest by title
- Modify privacy of interest
- Create notification
- Get notifications by user id
- Get active notifications by user id
- Deactivate notifications of user
- Create comment
- Get comments by content id
- Create annotation
- Get annotation by content id

3.4. Class Diagram

Overall class diagram is depicted as the following:



Class Diagram

4. Work Done So Far – Future Work

1. Requirements are created.
2. GitHub repository is created. (<https://github.com/yaseminalpay/motive>)
3. Project plan with milestones are created in GitHub.
4. Progress is being followed with GitHub issues.
5. System is designed by creating architectural and structural diagrams, class diagram, and API doc.
6. API side is implemented with Spring Boot and MongoDB.
7. API side is deployed to AWS with a documentation in GitHub wiki. (<http://motive-env.dyabhv3spp.us-east-1.elasticbeanstalk.com/>)
8. Tests are being implemented for API side (still in progress).