HEKA

Practice Application

Milestone Report II

CmpE352, Group 4

Spring 2022

Elif Tokluoğlu
Yusuf Bayındır
Berat Damar
Umut Deniz Şener
Sanberk Serbest
Hatice Kübra Aksu
Oğuzhan Tok
Erdinç Günaydın
Miraç Batuhan Malazgirt
Yiğit Can Özkaya

Table of Contents

Executive Summary

Description

HEKA is a basic blog application where users are able to post their ideas and experiences in a restricted social environment in which users can communicate via commenting under posts.

It serves as a native web application.

Overall Status

Team members came up with an idea for a basic blog application in the first meeting organized for the practice application. Requirement analysis started just after members decided to go after this idea.

The implementation phase was initiated no sooner than the requirement analysis using Django, a high-level Python web framework. Tasks were divided among team members and implemented one after another in accordance with requirements.

Team members learned two days before the deadline that Class Design, Use Case Design, and Sequence Diagrams should also be delivered via the Milestone Report II. Detailed design work was initiated a little bit late. Therefore, team members started to elicit inconsistencies and modify the requirements, design, and implementation in a way that eliminates these inconsistencies.

Beyond the basic functionality of the platform, members also tried to come up with some external APIs that facilitate the platform. Detailed explanations of APIs are listed below. More detailed documentation of APIs is presented in individual reports.

API URL(s)

- **BASE_URL/api/categories**
- **BASE_URL/api/posts**
- **BASE_URL/api/edit-profile/**
- **BASE_URL/api/all-profiles/**
- **BASE_URL/api/users**
- **BASE_URL/api/like/count/<post-id>**
- **BASE_URL/api/like/post/**
- **BASE_URL/api/comments-api/**

Basic Functionality of Our Project

Users are able to register, log in to platform, and log out from the platform. Users are able to create, edit, and delete posts. The platform also provides a restricted social environment in which users can comment under posts.

Posts are created by providing title, content, location, and category. Users are able to like or dislike posts but not both. Some external APIs such as BMI Calculator and Search for Disease for a given keyword are also presented in the platform.

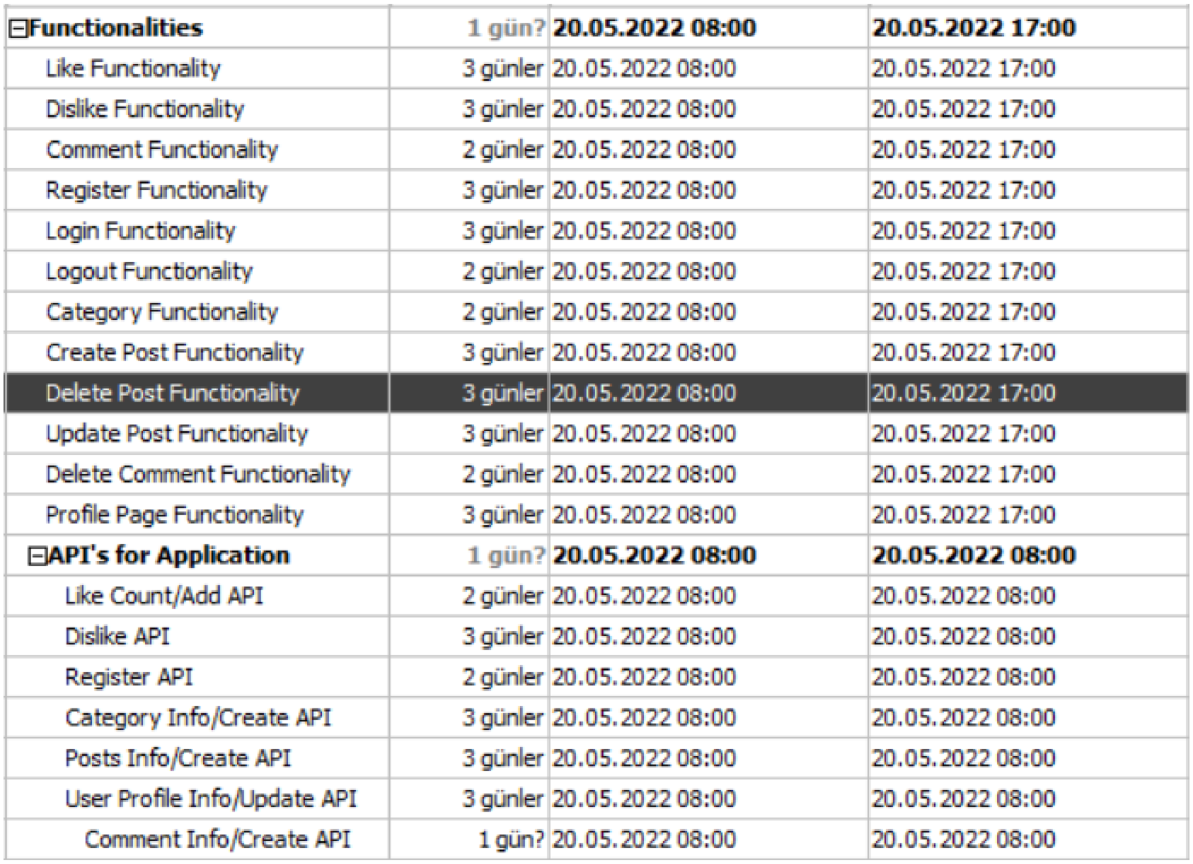
Challenges We Met As a Group

First of all we had to first determine our requirements. However since we have done that for the first part of the course it did not take so long to prepare requirements of this project. Secondly we had to make our application design. We designed our class, use-case and sequence diagrams. Again since we had done that in the first half of the course it did not take much to design the project.

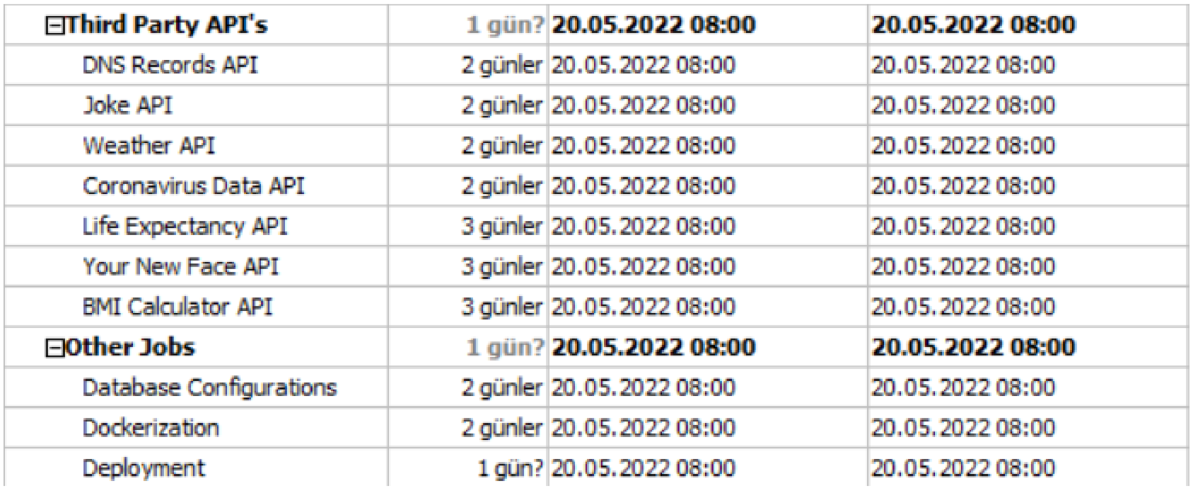
However hard part of the project started right after designing the project. We had to learn a lot of new tools. Firstly most of us did not do web development before this project. So we first learned the tools and technologies that are required for web development. Some group members helped other group members in this learning process.

After that, we decided to use 'develop' branch to push our changes in the project. However rarely do some pull requests conflicted with each other. So we had to learn carefully review our codes. Also, we take care of developing the API part of the project by helping and teaching each other. Additionally deployment part was also a little tricky for us but we handled it

Project Plan:



⊟Functionalities	**1 gün?**	**20.05.2022 08:00**	**20.05.2022 17:00**
Like Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
Dislike Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
Comment Functionality	2 günler	20.05.2022 08:00	20.05.2022 17:00
Register Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
Login Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
Logout Functionality	2 günler	20.05.2022 08:00	20.05.2022 17:00
Category Functionality	2 günler	20.05.2022 08:00	20.05.2022 17:00
Create Post Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
Delete Post Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
Update Post Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
Delete Comment Functionality	2 günler	20.05.2022 08:00	20.05.2022 17:00
Profile Page Functionality	3 günler	20.05.2022 08:00	20.05.2022 17:00
⊟API's for Application	**1 gün?**	**20.05.2022 08:00**	**20.05.2022 08:00**
Like Count/Add API	2 günler	20.05.2022 08:00	20.05.2022 08:00
Dislike API	3 günler	20.05.2022 08:00	20.05.2022 08:00
Register API	2 günler	20.05.2022 08:00	20.05.2022 08:00
Category Info/Create API	3 günler	20.05.2022 08:00	20.05.2022 08:00
Posts Info/Create API	3 günler	20.05.2022 08:00	20.05.2022 08:00
User Profile Info/Update API	3 günler	20.05.2022 08:00	20.05.2022 08:00
Comment Info/Create API	1 gün?	20.05.2022 08:00	20.05.2022 08:00



⊟Third Party API's	**1 gün?**	**20.05.2022 08:00**	**20.05.2022 08:00**
DNS Records API	2 günler	20.05.2022 08:00	20.05.2022 08:00
Joke API	2 günler	20.05.2022 08:00	20.05.2022 08:00
Weather API	2 günler	20.05.2022 08:00	20.05.2022 08:00
Coronavirus Data API	2 günler	20.05.2022 08:00	20.05.2022 08:00
Life Expectancy API	3 günler	20.05.2022 08:00	20.05.2022 08:00
Your New Face API	3 günler	20.05.2022 08:00	20.05.2022 08:00
BMI Calculator API	3 günler	20.05.2022 08:00	20.05.2022 08:00
⊟Other Jobs	**1 gün?**	**20.05.2022 08:00**	**20.05.2022 08:00**
Database Configurations	2 günler	20.05.2022 08:00	20.05.2022 08:00
Dockerization	2 günler	20.05.2022 08:00	20.05.2022 08:00
Deployment	1 gün?	20.05.2022 08:00	20.05.2022 08:00

List and status of deliverables

Name	Due	Delivery Date	Prepared By	Status
Requirements	10.05.2022	10.05.2022	Sanberk Serbest, Berat Damar, Oğuzhan Tok, Miraç Batuhan Malazgirt, Erdinç Günaydın, Yusuf Bayındır, Yiğit Can Özkaya	Done
Use Case Diagram	13.05.2022	13.05.2022	Sanberk Serbest, Berat Damar, Oğuzhan Tok, Miraç Batuhan Malazgirt, Erdinç Günaydın, Yusuf Bayındır, Yiğit Can Özkaya	Done
Class Diagram	13.05.2022	13.05.2022	Sanberk Serbest, Berat Damar, Oğuzhan Tok, Miraç Batuhan Malazgirt, Erdinç Günaydın, Yusuf Bayındır, Yiğit Can Özkaya	Done
Sequence Diagram	13.05.2022	13.05.2022	Sanberk Serbest, Berat Damar, Oğuzhan Tok, Miraç Batuhan Malazgirt, Erdinç Günaydın, Yusuf Bayındır, Yiğit Can Özkaya	Done
URI of the Tag of your project.	20.05.2022	20.05.2022	Sanberk Serbest, Berat Damar, Oğuzhan Tok, Miraç Batuhan Malazgirt, Erdinç Günaydın, Yusuf Bayındır, Yiğit Can Özkaya	Done

Code	20.05.2022	20.05.2022	Sanberk Serbest, Berat Damar, Oğuzhan Tok, Miraç Batuhan Malazgirt, Erdinç Günaydın, Yusuf Bayındır, Yiğit Can Özkaya	Done

Evaluation of Tools and Processes

Trello

Trello helps us to organize our tasks. Thanks to Trello, we are able to manage which task is being done by whom. It makes it easy to divide tasks into subteams. Every member of subteams can see which tasks should be done and which of them are currently in progress.

Slack

We used slack to communicate with TAs and Prof. Uskudarlı up to now. We know that Slack is a communication app that can be easily integrated with many different apps. It has a user-friendly and simplistic interface. Another useful feature is the persistent state; when you log in after some time, it shows you the last message you have seen. That makes it easier to track messages.

Github

Github is a platform where all the work that we have done gets together. We keep ourselves up to date by following the wiki page.It gives the opportunity to trace individuals by using Github's commit history. Using the issue system we can easily identify the undone issues, prioritize them and then focus them. That feature also traces the individual effort of every team member.

Draw.io

Draw.io is a useful platform providing users a free means for drawing charts, schematics, diagrams etc. It has various built-in templates allowing users to easily create visualizations of works with different formats and purposes. This tool enabled us to create our mock-ups for scenarios.

Lucidchart

Lucidchart is quite a useful proprietary platform that is used to allow users to collaborate on drawing and sharing charts and diagrams. One of the most significant advantages of Lucidchart is being easy to use. It is a well fit tool if you are working as a team to draw some diagrams or charts simultaneously. It gives enough properties when you use your boun email address to log in.

Microsoft Teams

We organize our meetings on Microsoft Teams.You can create a subchannel for a subteam. The sub teams are able to continue their task in their subchannel without confusion with other subteams. It provides simultaneous meetings at the same time. Every task meeting can be traced. Therefore, you can see who attends which meeting.

WhatsApp

We are using WhatsApp during an emergency situation to receive quick responses. WhatsApp is not an optimal solution to provide communication between team members since it does not have multiple channel support.Therefore, we try to avoid it unless we have to use it.

Django

We used Django as a framework. It is a free and open source web application framework written in Python. It provides simplicity, flexibility, reliability and scalability.Django offers an administrative interface right out of the box that is both professional and versatile. Django uses the MVP (Model View Template) architecture. The model is the layer where the database operations are performed. We model our database related operations in this layer and use this layer when it is available. SQL commands are not needed to create a database in Django.It makes it easy to develop without caring about tables. A database is created in a language specific to Django and Django is used with its own database comment. View is our development part. All our python codes are located here. It is a bridge between other layers, and we draw the necessary part with our Python codes and use it in this layer. Template is our design and presentation layer. The web page or elsewhere is concerned with how our page will look.

Docker

Docker works as a machine that runs python on it, it runs our django application. Also, we can reach our repository using Docker.Hub from anywhere and run it with one line code.Primarily, it enables easy deployment to the cloud. Beyond that, Docker technology is also more controllable and and is a microservices-based method focused on efficiency

MySQL

MySQL is a relational database management system based on SQL – Structured Query Language. It is an open source software supported by Oracle. You can store data in tables, create indexes on the data and query the data using SQL queries.It allows you to manage relational databases. The other advantage is that MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. Although we can not experience yet in that project, MySQL works very quickly and works well even with large datasets.

Work Done So Far

Berat Damar
- Attending all weekly meetings and customer meetings - I met with Sanberk and we decided core of our application(https://github.com/bounswe/bounswe2022group4/issues/117) - Researching on Django and Django Rest Framework - Researched Django unit tests - Writing registration API with GET and POST (https://github.com/bounswe/bounswe2022group4/issues/124) - Searching a external API and decided OpenWeather (https://github.com/bounswe/bounswe2022group4/issues/153) - Implementing login and logout functionality (https://github.com/bounswe/bounswe2022group4/issues/135) - Testing user registration system (https://github.com/bounswe/bounswe2022group4/issues/171) - Milestone 2 Evaluation of Tools (https://github.com/bounswe/bounswe2022group4/issues/202) - Create sequence Diagram for our app (https://github.com/bounswe/bounswe2022group4/issues/205) - Documenting individual report (https://github.com/bounswe/bounswe2022group4/issues/217)

Elif Tokluoğlu

Miraç Batuhan Malazgirt

- Determined and Wrote Requirements of Practice Application to Milestone Report

- Created Use Case Diagram for the design of Practice Application

- Wrote Parts of Executive Summary of Milestone II

- Implemented Coronavirus Data of Countries via External API
- Pull Request

- Implemented Categories and Posts of Categories to codebase
- Pull Request

- Category API Implemented and Frontend - API connection
- Pull Request

- Implemented Unit Tests
- Pull Request

Oğuzhan Tok

- Implementing DNS Records of Domains
  https://github.com/bounswe/bounswe2022group4/issues/137

- Implementing Upvote-Downvote Functionalities
  https://github.com/bounswe/bounswe2022group4/issues/144

- Implementing Like API
  https://github.com/bounswe/bounswe2022group4/issues/180

- Upvote & Downvote Functionality Revisiting
  https://github.com/bounswe/bounswe2022group4/issues/161

- Re-organizing Upvote-Downvote Branches
  https://github.com/bounswe/bounswe2022group4/issues/150

- Like Functionality and API Unit Tests
  https://github.com/bounswe/bounswe2022group4/issues/211

- Creating Project Plan
  https://github.com/bounswe/bounswe2022group4/issues/214

- Pull Requests

https://github.com/bounswe/bounswe2022group4/pull/139
https://github.com/bounswe/bounswe2022group4/pull/146
https://github.com/bounswe/bounswe2022group4/pull/149
https://github.com/bounswe/bounswe2022group4/pull/151
https://github.com/bounswe/bounswe2022group4/pull/176
https://github.com/bounswe/bounswe2022group4/pull/181

  - Researched about django rest framework
  - Attending meetings
  - Decisioning of implementing which framework will being used
  - Researching about git

Yusuf Bayındır

- Implementation of like/dislike functionality.
  https://github.com/bounswe/bounswe2022group4/issues/161
- Database configuration of the application.
  https://github.com/bounswe/bounswe2022group4/issues/156
- Search for Disease implemented using an external API.
  https://github.com/bounswe/bounswe2022group4/issues/179
- Class design of the practice application.
  https://github.com/bounswe/bounswe2022group4/issues/212
- Dislike API which returns the total number of likes given a post title with GET request.
  (Details are presented in the individual report.)
- Executive Summary of Milestone Report II.
  https://github.com/bounswe/bounswe2022group4/issues/199
- Test case for dislike functionality.
  (Details are presented in the individual report.)

Sanberk Serbest

I attended many long meetings, followed some web development tutorials and implemented some functionalities.

- Implemented home page

https://github.com/bounswe/bounswe2022group4/pull/123

https://github.com/bounswe/bounswe2022group4/issues/119

- Initial database
  https://github.com/bounswe/bounswe2022group4/issues/118
  https://github.com/bounswe/bounswe2022group4/pull/125

- Delete Post
  https://github.com/bounswe/bounswe2022group4/pull/130
  https://github.com/bounswe/bounswe2022group4/issues/121

- Update Post
  https://github.com/bounswe/bounswe2022group4/issues/122
  https://github.com/bounswe/bounswe2022group4/pull/131

- Page for posts of a specific user
  https://github.com/bounswe/bounswe2022group4/pull/134
  https://github.com/bounswe/bounswe2022group4/issues/132

- Split the posts into different pages
  https://github.com/bounswe/bounswe2022group4/issues/133

- External API
  https://github.com/bounswe/bounswe2022group4/issues/186
  https://github.com/bounswe/bounswe2022group4/pull/203

- Provide an API for posts
  https://github.com/bounswe/bounswe2022group4/issues/187
  https://github.com/bounswe/bounswe2022group4/pull/203

- Login check for creating/deleting posts
  https://github.com/bounswe/bounswe2022group4/issues/188

- Unit test additions
  https://github.com/bounswe/bounswe2022group4/issues/189
  https://github.com/bounswe/bounswe2022group4/pull/208

- Create Sequence Diagrams for the practice app
  https://github.com/bounswe/bounswe2022group4/issues/205
  https://lucid.app/lucidchart/53219fab-040f-4bfe-aea1-08c5c503ac34/edit?invitationId=inv_066f222c-4fbc-4212-ae4c-fcf8a552aea7

- Individual Report
  https://github.com/bounswe/bounswe2022group4/issues/218

Umut Deniz Şener

- Researched about github branches and commands
- Researched Django.models Library
- Implement comment system by using Django.models library (https://github.com/bounswe/bounswe2022group4/issues/147)
- Researched about Django Rest Framework
- Implement an Restful API for comment system
- Use that api in the system (https://github.com/bounswe/bounswe2022group4/issues/157) (https://github.com/bounswe/bounswe2022group4/issues/160)
- Researched about free 3rd party API's
- Implement a BMI Calculator system for the application (https://github.com/bounswe/bounswe2022group4/issues/168)
- Researched Django unit tests
- Writing unit test cases for comment system (https://github.com/bounswe/bounswe2022group4/issues/158)
- Writing unit test cases for comment api (https://github.com/bounswe/bounswe2022group4/issues/159)
- Writing unit test cases for bmi system and api (https://github.com/bounswe/bounswe2022group4/issues/168)
- Creating use case diagram for the practice application with Mirac Batuhan Malazgirt (https://github.com/bounswe/bounswe2022group4/issues/200)

Yiğit Can Özkaya

- Implement delete comment in application https://github.com/bounswe/bounswe2022group4/issues/164

- Implement Joke API to application
  https://github.com/bounswe/bounswe2022group4/issues/163
- Researched about API's and others
- Review merge requests
  https://github.com/bounswe/bounswe2022group4/pull/207
  https://github.com/bounswe/bounswe2022group4/pull/185
  https://github.com/bounswe/bounswe2022group4/pull/178
- Dockerization of application
  https://github.com/bounswe/bounswe2022group4/issues/183
- Deployment of application
  https://github.com/bounswe/bounswe2022group4/issues/184
- Researching about git
- Implementation of templates of delete comment and joke apı
- Researched about django rest framework
- Attending meetings
- Decisioning of implementing which framework will being used
- Implementation of redesigning of home and base template

Erdinç Günaydın

- Login user bug
  https://github.com/bounswe/bounswe2022group4/issues/175
- Practice-App Profile Get and Post Apis
  https://github.com/bounswe/bounswe2022group4/issues/163
- Researched about API's and others
- Review merge requests

https://github.com/bounswe/bounswe2022group4/pull/193
https://github.com/bounswe/bounswe2022group4/pull/178
https://github.com/bounswe/bounswe2022group4/pull/177

- Practice app category Revision
  https://github.com/bounswe/bounswe2022group4/issues/197
- Practice App- Unit Tests
  https://github.com/bounswe/bounswe2022group4/issues/215
- Researching about git
- Static ip to Dynamic
  https://github.com/bounswe/bounswe2022group4/issues/209
- External Aestatic API
  https://github.com/bounswe/bounswe2022group4/issues/196

- Researched about django rest framework
- Attending meetings
- Decisioning of implementing which framework will being used

Hatice Kübra Aksu

Deliverables

Requirements

Glossary

- User: A person who uses or interacts with the application.
- Guest: A person who uses or interacts with the application as an unregistered user.
- Category: Any medical specialty or subspecialty.
- Discussion Thread: Series of posts in forum or comment section that revolves around a single focused topic.
- Username: Identification used by the users to access the application.
- Password: Secret word or phrase that must be used to access the application
- Homepage: The introductory page of the application.
- Upvote: Feature that allows users to signal their approval or support for a post.
- Post: Piece of writings with attachments under categories that are intended to share a medical experience or domain-specific knowledge.

- Comment: Tool of communication and discussion under a post.
- Tool: Some medical tests like BMI calculator, Body Surface Area Calculator, Pregnancy Due Date Calculator etc.
- Timeline: Medium of displaying posts under followed categories and of followed health professionals.
- Location: The physical address of a user.

1.Functional Requirements

1.1 User Requirements

1.1.1 Authentication

- 1.1.1.1 Sign Up

  - 1.1.1.1.1 Users shall provide a username and e-mail address in the system.

– 1.1.1.1.2 Users shall set a password that is longer than 8 characters and contains at least one upper case letter, one lower-case letter, and one number.

– 1.1.1.1.3 Users shall confirm the password by writing the same password again.

- **1.1.1.2. Sign In**

– 1.1.1.2.1 Users shall provide their e-mails and passwords for signing in.

– 1.1.1.2.2 Users shall get a warning message which is " Your e-mail or password is wrong." if the provided e-mail or password is wrong.

– 1.1.1.2.3 Users shall see the home page after signing in successfully.

1.1.2. User Interactions

- 1.1.2.1 Forum

- 1.1.2.1.1 Users shall be able to create posts.
- 1.1.2.1.2 Users shall be able to comment under other users' post.
- 1.1.2.1.3 Users shall add a title while creating posts.
- 1.1.2.1.4 Users shall add content to their posts and comments.
- 1.1.2.1.6 Users shall select categories of their posts from a specified set of categories.
- 1.1.2.1.7 Users shall be able to share location in their posts and comments.
- 1.1.2.1.8 Users shall be able to read a post shared by another user.
- 1.1.2.1.9 Users shall be able to edit their posts and comments.
- 1.1.2.1.10 Users shall be able to delete their posts and comments.
- 1.1.2.1.12 Users shall be able to upvote other users' posts and comments.
- 1.1.2.1.16 Users shall be able to see all posts.

- 1.1.2.2 **Profile Pages**

- 1.1.2.2.1 Users shall have a profile page.
- 1.1.2.2.2 Users shall be able to log out from their profile pages.
- 1.1.2.2.3 Users shall see other users' user type from their profile pages.

- 1.1.2.2.4 Users shall see other users' past comments and post from their profile pages.

- 1.1.2.2.5 Users shall have their usernames visible on their profile pages.

1.1.2.4 Tool

- 1.1.2.4.1 Users shall be able to access Tools from the homepage.

- 1.1.2.4.2 Users shall be able to pass input if required by the tool.

1.2 System Requirements

1.2.5. Forum

- 1.2.5.1. The system shall show the number of upvotes given to a post or comment at the lower right corner of every post or comment.

- 1.2.5.3. The system shall show the category of a post at the upper-left corner of every post.

- 1.2.5.5. The system shall delete the whole discussion thread if the main post is deleted.

- 1.2.5.7. The system shall not delete the main post of a deleted comment.

- 1.2.5.9. The system shall provide healthcare-related categories to users.

- 1.2.5.10. The system shall display the posts of the current chosen category.

2.Non-Functional Requirements

2.1 Reliability Requirements

- 2.1.1. The system shall always run up to 2000 users.
- 2.1.2. The system shall be portable for website and application.
- 2.1.3. The system shall respond to any user no later than 4 seconds.

2.2 Availability Requirements

- 2.2.1. The system language shall be English.
- 2.2.2. The system shall be available as a native web application in browsers.

2.3 Security

- 2.3.1. The system shall prevent various cyber-attacks by providing SSL certificates to its users (HTTPS).
- 2.3.2. The system shall perform input validation properly wherever input is received such as forms, chatbot, etc.
- 2.3.3. The system shall provide an access control policy such that users cannot act outside of their intended permission.

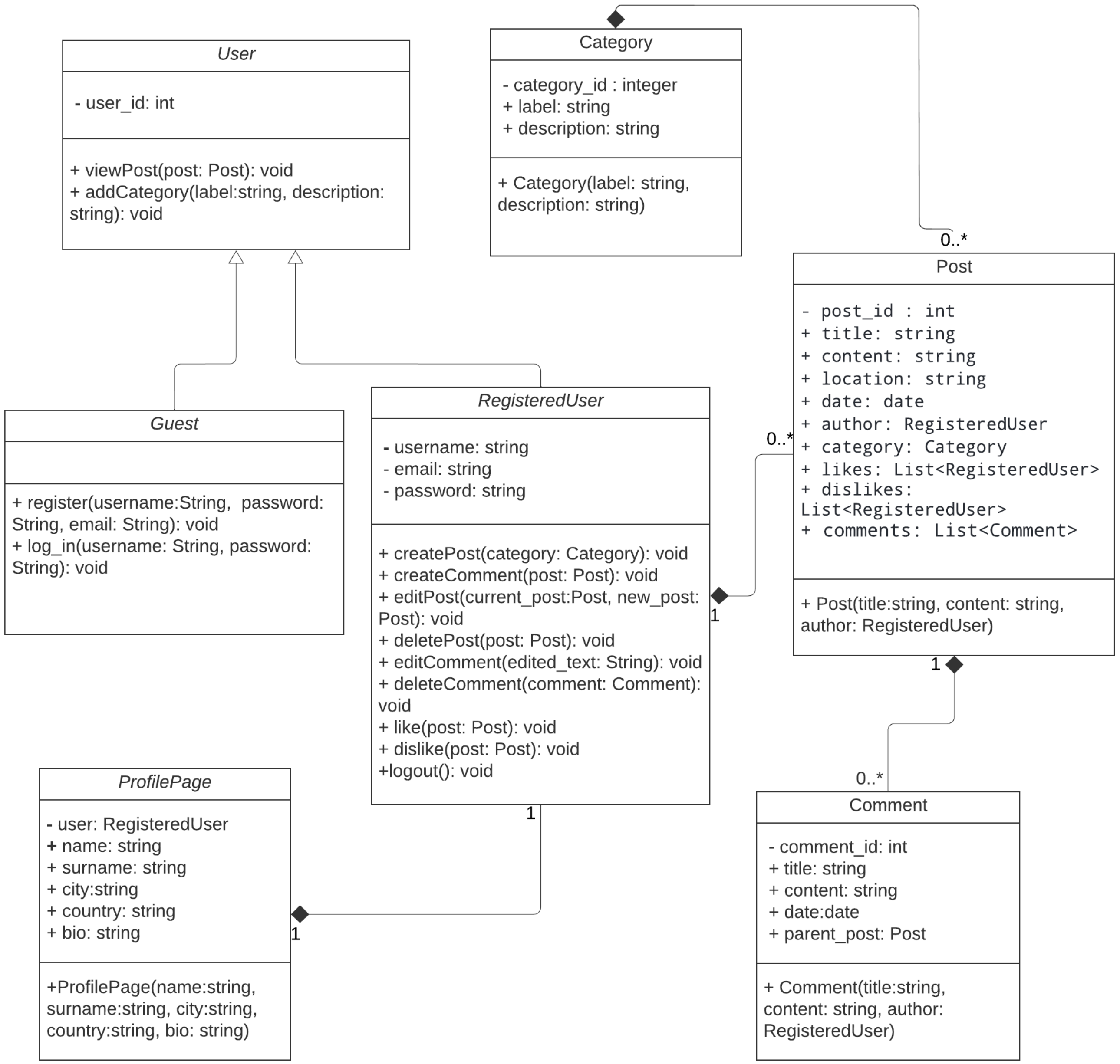
Diagrams

Use Case Diagram

To see in Lucidchart: link.

Class Diagram

To see in Lucidchart: link.


User
- user_id: int
+ viewPost(post: Post): void
+ addCategory(label:string, description: string): void
Category
- category_id : integer
+ label: string
+ description: string
+ Category(label: string, description: string)
0..*
Post
- post_id : int
+ title: string
+ content: string
+ location: string
+ date: date
+ author: RegisteredUser
+ category: Category
+ likes: List<RegisteredUser>
+ dislikes: List<RegisteredUser>
+ comments: List<Comment>
+ Post(title:string, content: string, author: RegisteredUser)
Guest
+ register(username:String, password: String, email: String): void
+ log_in(username: String, password: String): void
RegisteredUser
- username: string
- email: string
- password: string
+ createPost(category: Category): void
+ createComment(post: Post): void
+ editPost(current_post:Post, new_post: Post): void
+ deletePost(post: Post): void
+ editComment(edited_text: String): void
+ deleteComment(comment: Comment): void
+ like(post: Post): void
+ dislike(post: Post): void
+logout(): void
0..*
1
1
1
0..*
ProfilePage
- user: RegisteredUser
+ name: string
+ surname: string
+ city:string
+ country: string
+ bio: string
+ProfilePage(name:string, surname:string, city:string, country:string, bio: string)
1
Comment
- comment_id: int
+ title: string
+ content: string
+ date:date
+ parent_post: Post
+ Comment(title:string, content: string, author: RegisteredUser)

Sequence Diagrams

Link to new sequence diagrams: Practice App Sequence Diagrams: Lucidchart