ОТЧЕТ

О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ:

“Social networking project OnlyFriends”

Проект выполнял:

Чинь Дык Тханг

P3222

Преподаватель: Доц.

факультета ПИиКТ

Ефимчик Е.А

Санкт-Петербург

2021

Table of Contents

1 Introduction

The world we live in is more connected, with people living across various regions interacting in real-time without ever stepping outside their house. For people are expanding relationship day by day, a social network is always effectively playing an important role to help them keep in touch with each other. Inspired from Facebook, the largest social network in the World in term of number of users, we harboured a dream of creating a similar website broadening relationship and connecting them as well. That is the story of OnlyFriends, a friendly social networking website.

2 Analysis

2.1 About

A Web service that allows users to upload status, comment and share their daily thought.

2.2 Detail Description

The application should be supported the following requirements:

- User must register or login with existing account
- Content creator should be able to upload a photo and status.
- The user should be able to comment on other's post.
- The user should be able to follow other's profile.

2.3 Description of analogs:

- *Facebook* is a social networking site that makes it easy for you to connect and share with family and friends online. Originally designed for college students, Facebook was created in 2004 by Mark Zuckerberg while he was enrolled at Harvard University. By 2006, anyone over the age of 13 with a valid email address could join Facebook. Today, Facebook is the world's largest social network, with more than 1 billion users worldwide.

- *VK* is a Russian online social media and social networking service based in Saint Petersburg. VK is available in multiple languages but it is predominantly used by Russian-speakers.

- *Instagram* is an American photo and video sharing social networking service founded by Kevin Systrom and Mike Krieger. In April 2012, Facebook Inc. acquired the service for approximately US$1 billion in cash and stock

2.4 Comparative analysis of analogs:

Criterion	Facebook	Vkontakte	Instagram	OnlyFriends
Search engine	☑	☑	☑	✖
Developed for PC/mobile user	both	both	both	PC
Sharing status and commenting	☑	☑	☑	☑
Age restriction	From 18	From 18	From 18	Anyone
Administration system	☑	☑	☑	☑

Table 1: Comparative analysis of analogs

3 Use Case

3.1 List of Actors

- Admin: Administrator who runs the website has the rights to manage video/post, delete inappropriate comments, disable a user.
- User: To use the website the user needs to sign in the system. After logging in, the user will be able to add a status, comment on any post, manage his post or comment. Besides, the user can either see other profile or edit his profile.
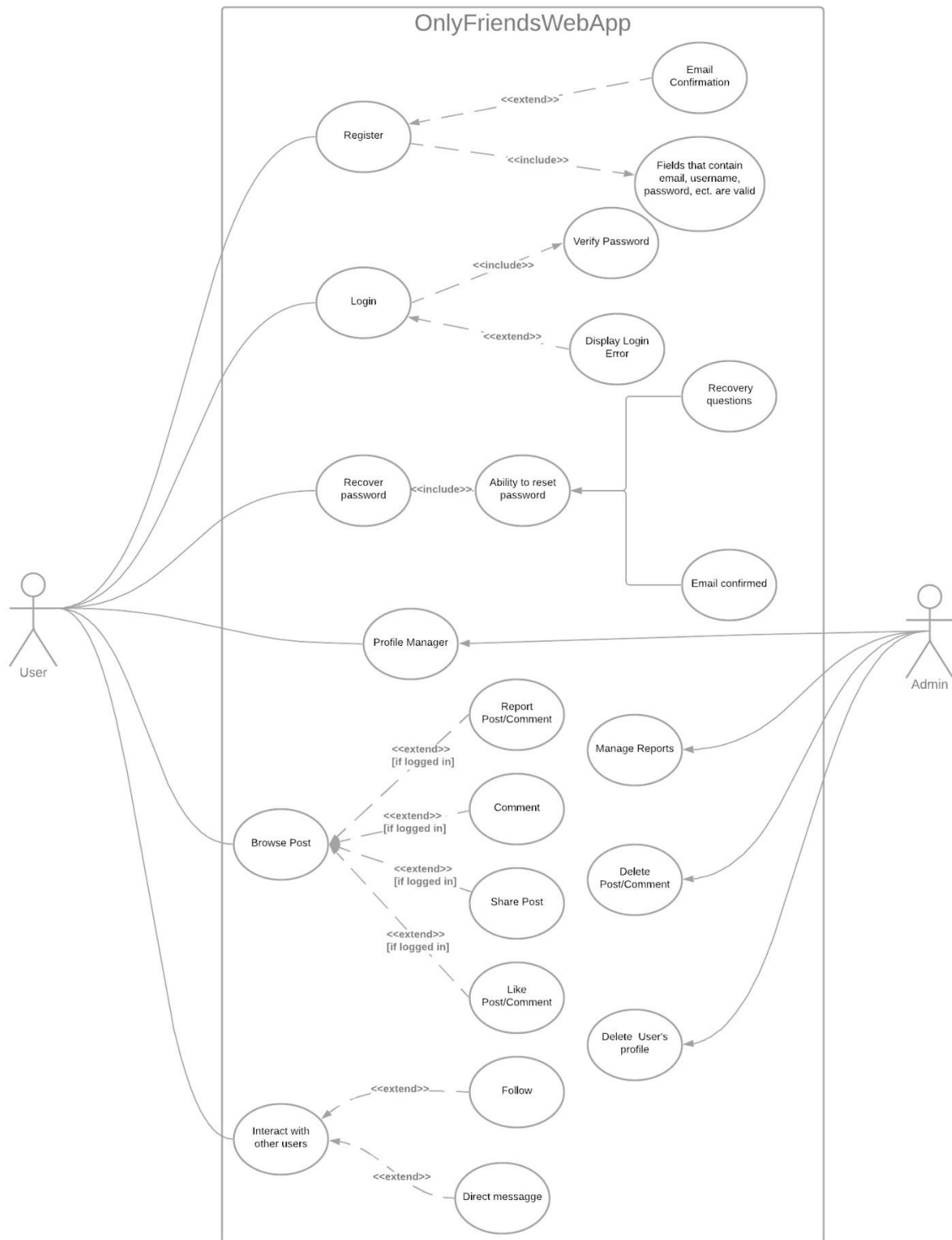
3.2 Use Case Diagram

Figure 1: Use Case Diagram

3.3 Use Case Specification

Use cases	Description	Actor	Preconditions	Main stream	Sub stream	Postconditions
Register	User can create a new account	User	User has no account and wishes to make a new one	User will fill out a registration form with basic information	If the information entered is not in the correct format or account already exists in the system, the user is forced to reenter it	If the information entered is successful, a new account will be created. In case the user entered email, the user needs to confirm that email.
Login	Sign in the website	User	User has a created account	User enters the login field with email or username and password	If the login information entered incorrectly, the user needs to fill in again.	With the correct login, the user will be directed to the main page of website and will be able to use some special features
Recover Password	To regenerate password	User	The user forgets password. This account must be set recovery questions or have email confirmed	The user may choose some recovery questions and answer them. All of which should be correct in two times. If the user has email confirmed, a verification code will be sent.	If the user gives incorrect answers more than two times, he needs to re-answer after 30 minutes. If the code entered by the user doesn't match the code sent by the system, the user can resend verification code after 2 minutes	Having confirmed by the system that the user owns this account, he can set a new password straight away.
Profile Manager	Edit profile, manager owned	Admin and User	User or Admin logged in the system	Add information, delete/edit	-	After user saves change, system update that

	posts			post, upload profile photo, change password, ect.		account's profile
Browse Post	Stream videos, see posts and read comments on the website	User	-	The user can scroll up and down to load new posts	If the user who hasn't logged in system interacts with post and other users, the website will torward to login/sign up page. For registered user, he can report violation, react to post, leave comments and share	-
Interact with other users	Conectivity between users	User	User logged in system	User should be able to follow other users and send direct messages	-	-
Manager Reports	Display violations	Admin	Logged in as admin	The admin should be able to read add violations reported by user	-	-

Table 2: Use Case Specification

4   Database Design

4.1   List of Entities, Properties

Entity	Property	Comment
User	user_id	Key property for this entity. User ID is generated automatically.
	email	User's email address.

	password	User's password.
	Name	User's name to be displayed on website.
	type_of_user	This property can be user or admin.
Profile	user_id	User's ID
	info	All information about user displayed like json file.
following_list	user_id	This is a foreign key property.
	follower_id	This is a foreign key property.
Post	post_id	Each post has a key ID.
	author_id	ID of the creator of this post.
	date	Date created.
	post_type	This property value can be TEXT or IMAGE_VIDEO
Text_post	post_id	Post ID
	post_text	Post content.
IMAGE_VIDEO_POST	post_id	Post ID
	post_text	Post content, including url to file location of image or video.
Comments	comment_id	Comment ID.
	post_id	Post ID that contains this comment
	commenter_id	ID of Commenter.
	comment_text	Content of this comment.
Likes	like_id	ID of this like
	like_on_id	Comment ID that receives reaction.
	like_by_id	who reacted to this.
	reaction_type	This field can take values of upvote(agree) or downvote(unsupport)

Table 3: List of Entities, Properties

Relationships:
- One-to-many relationship (user creates) between User and Post. A User may create several posts.
- One-to-many relationship (user comments) between User and Comments. A User may comment multiple times.
- One-to-many relationhsip (user likes) between User and Likes. A User may have various likes from post's comments.
- One-to-many relationship (comment_like) between Likes and Comments. A comment can receive numerous likes.
- One-to-one relationship (Has) between User and Profile. Each User has one and only one Profile.
- One-to-many relationship (post_comment) between Post and Comments. A Post can have lots of comments.

4.2   Entity Relationship
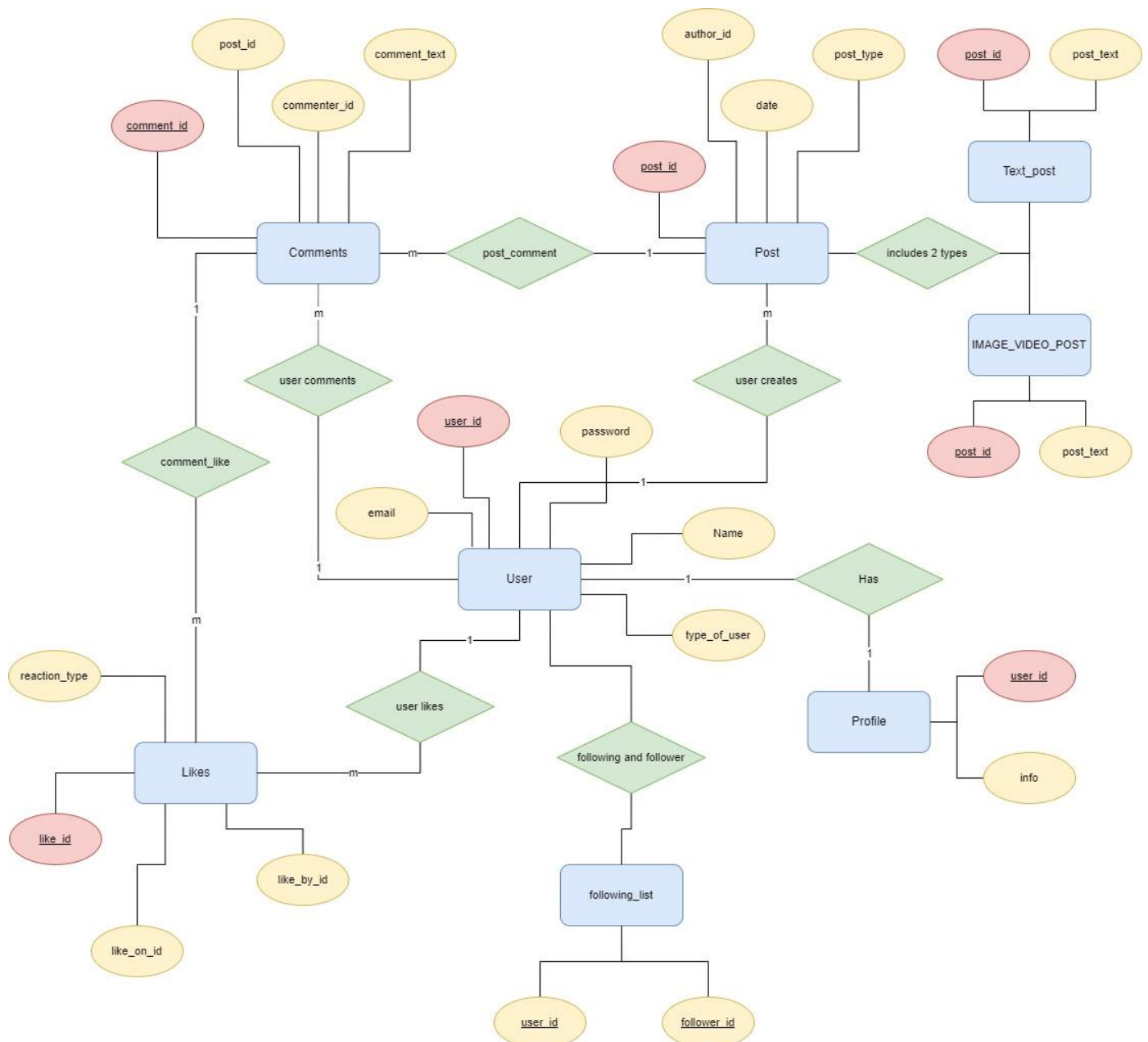
Figure 2: Entity Relationship

4.3   Database model
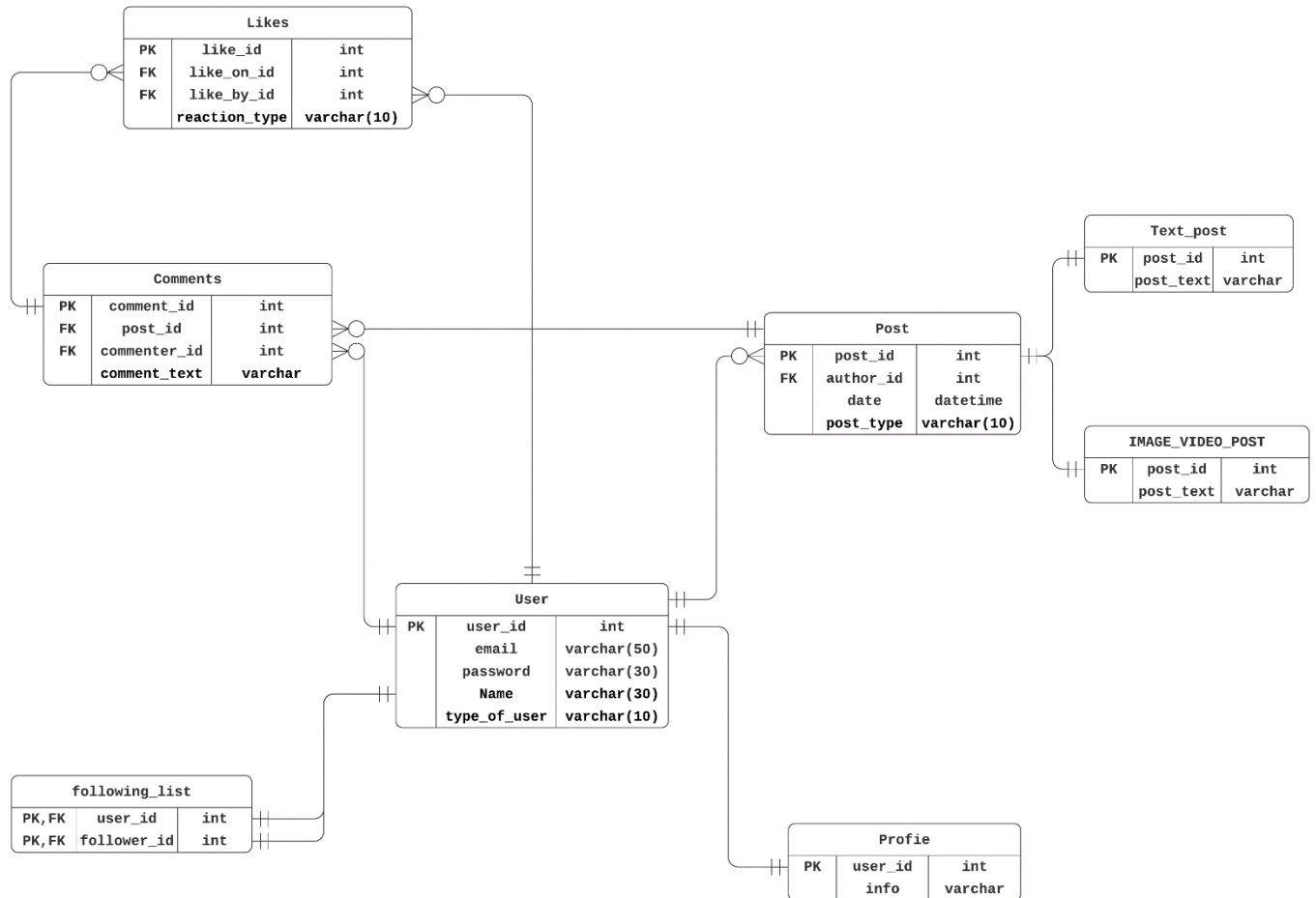
Figure 3: Database Model

5   User Interface

5.1   List of Interfaces

- **Not Authenticated**
  - Login page: Before authentication, if the user tries to access any url, it will automatically redirect to login page.
  - Register page.
  - Verification page
  - Message page: After registering, the user needs to click on the link that sent to the user's email in order to activate account. The link is basicly a request contains token as request param. After that, it will direct to the message page, whose content is:
    - Registration confirmed. You can now log in.
    - This token has expired.
    - Invalid user

  Besides, a message page with content "Access denied" will appear when an error code 403 occurs in sever.
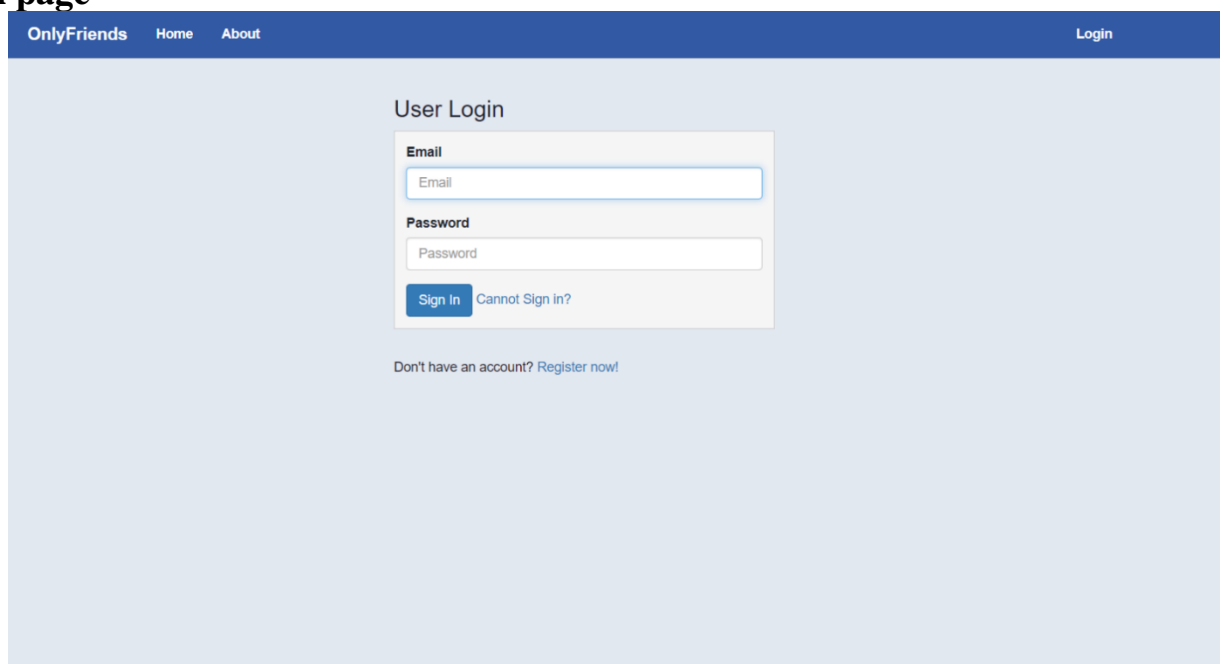
- **User**
  - News feed page: Display of statuses and comments on them. If a comment is commented by the user, there will be a link button "delete" allows the user to delete that comment. At the footer of a Status, there is a comment box, where the user can post a comment.
  - Add status page: When the user is authenticated, there will be a link button navigating to add status page. After clicking on addStatus(submit) button, a post request will be sent and the website will navigate to New feed page.
  - Edit status Page: It's quite similar to add status page.
  - About page: This is profile page showing information of the user. There will be an Edit button, navigating to edit profile page. If the user go to other profile page, there will be only display of information of that user without button Edit.
  - Edit Profile page: User can change personal information and change profile photo(avatar).
  - Home page: Display all statuses owned by the User. More over the user can add image to status, delete image of the status, edit status text and permanently delete that status.
  - Upload Image page: This page appears when the user changes profile photo and add image to his status.

- **Admin**
  - User Management Page: User with role ADMIN can see all user and disable any user.
  - Status Management Page: User with role ADMIN can see all statuses and comments, he should be able to delete them as well.

5.2 Interface Appearance

1. Login page

Figure 4: Login page

Figure 5: Login page with error

2. Register page

Figure 6: Register page

Figure 7: Register page with error

3. Verification page

Figure 8: Verification page

4. Message page

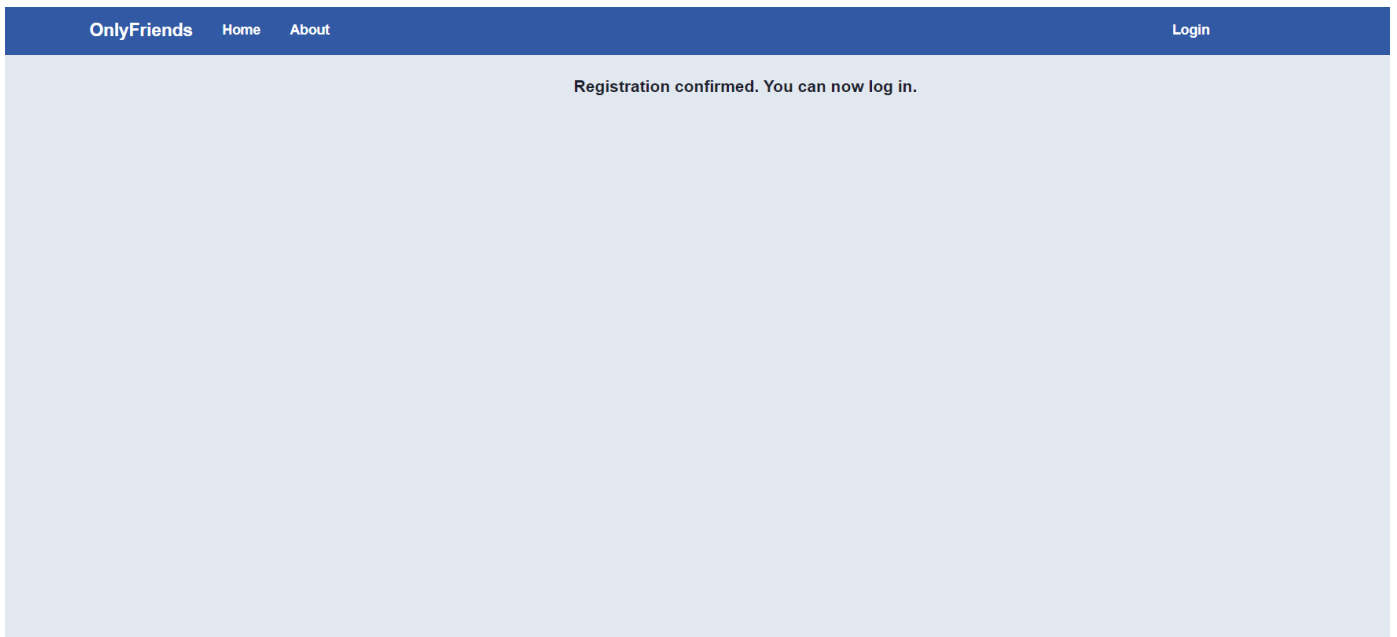Registration confirmed. You can now log in.

Figure 9: Message page

5. News feed page

Figure 10: News feed page

6. Add Status page
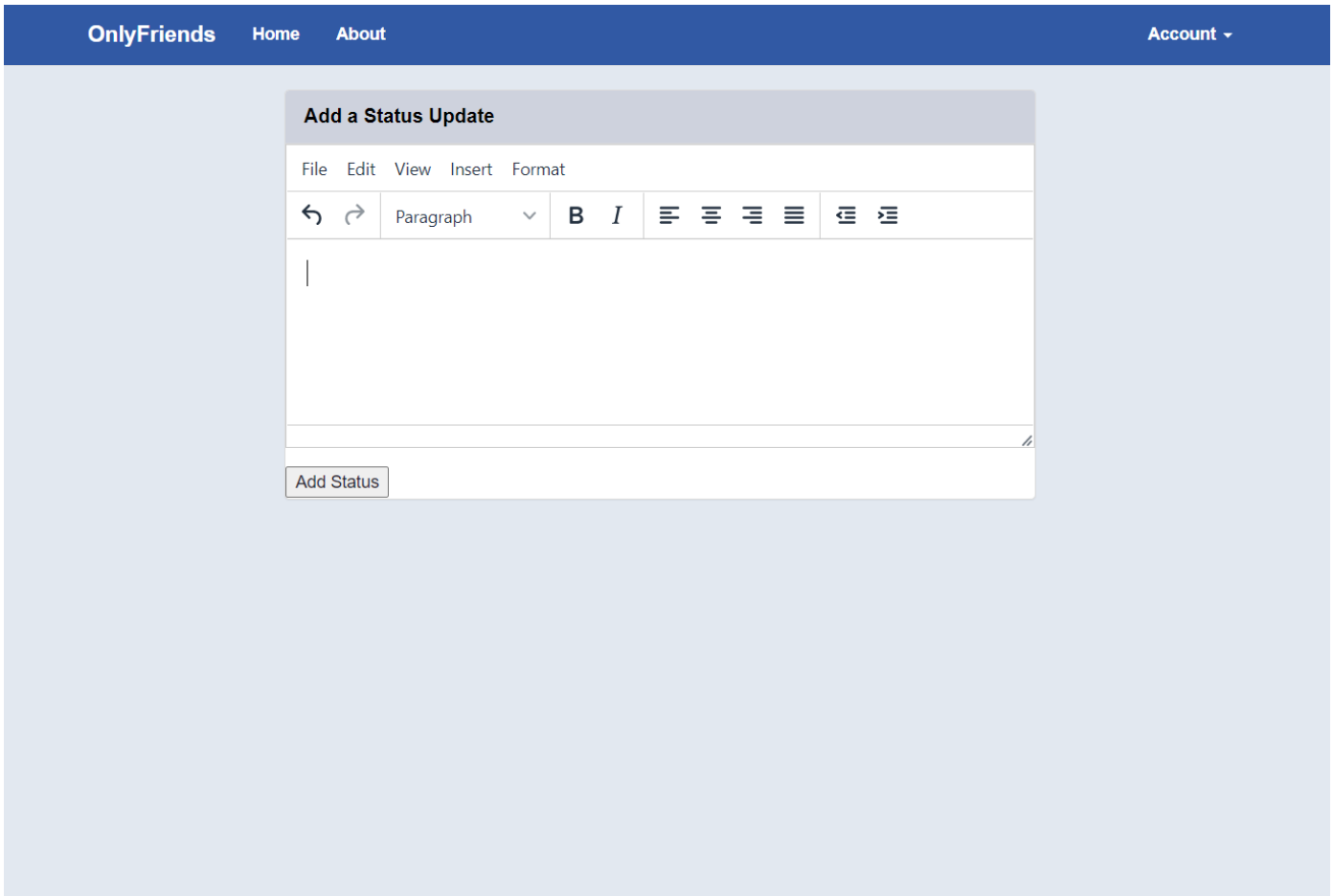
Figure 11: Add status page

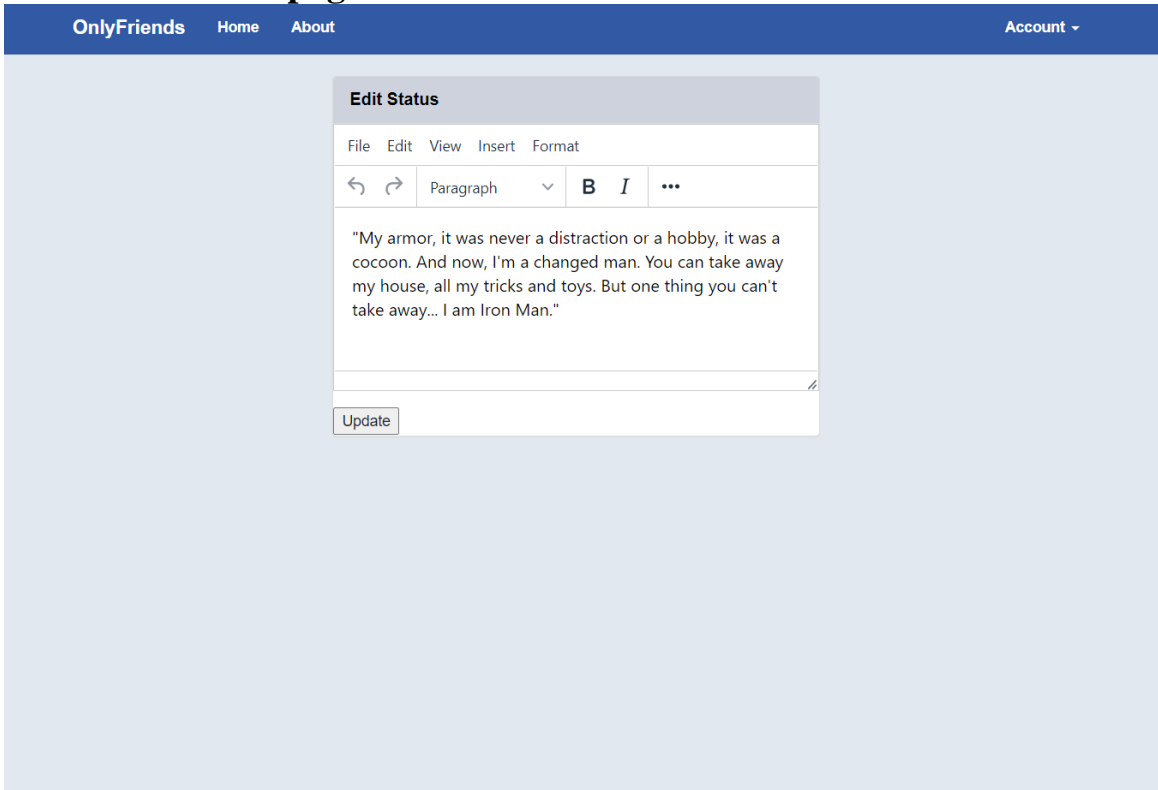# 7. Edit Status page

Figure 12: Edit status page

8. About page

Figure 13: About page

9. Edit Profile page

Figure 14: Edit profile page

16

10. Home page
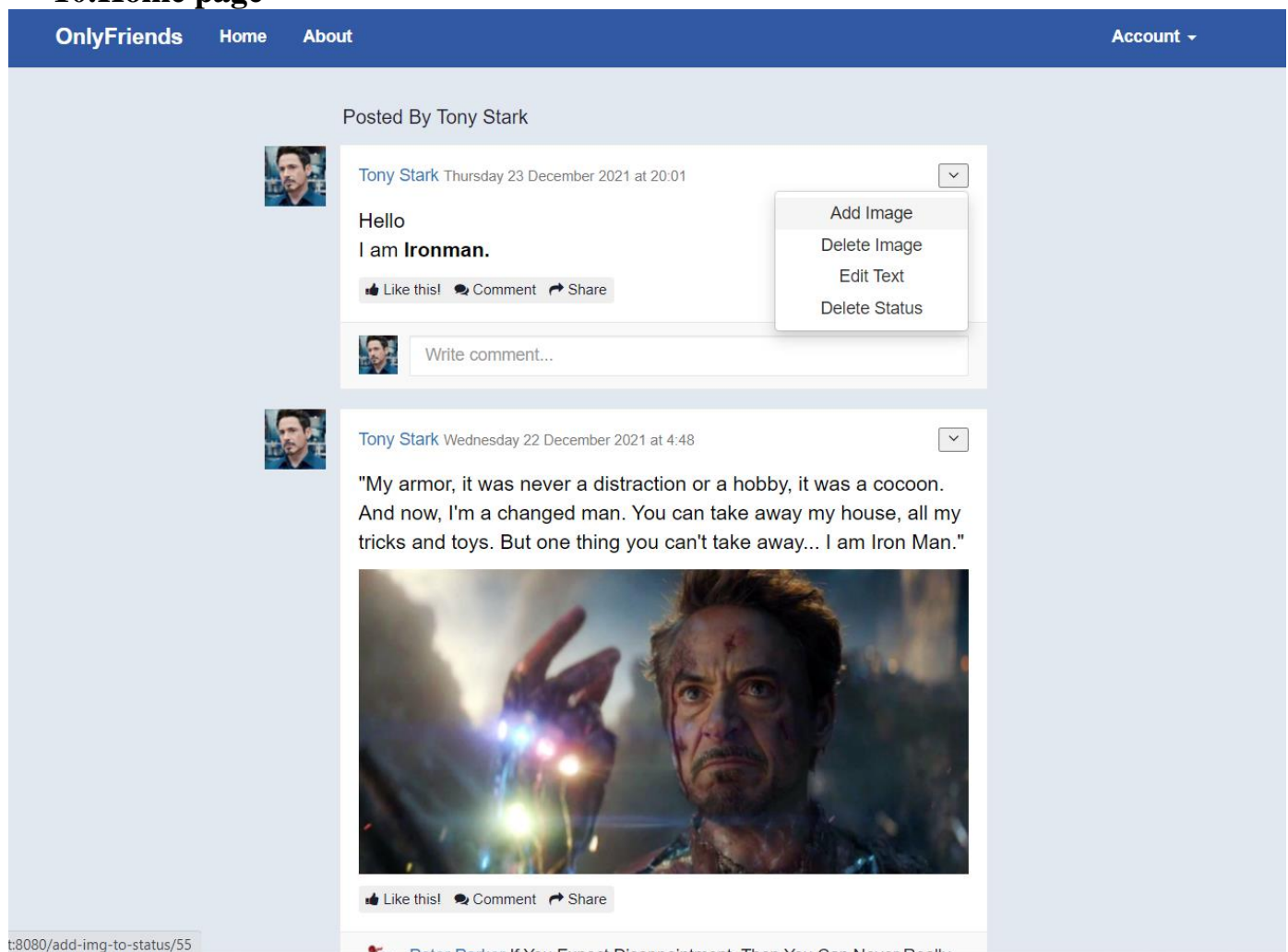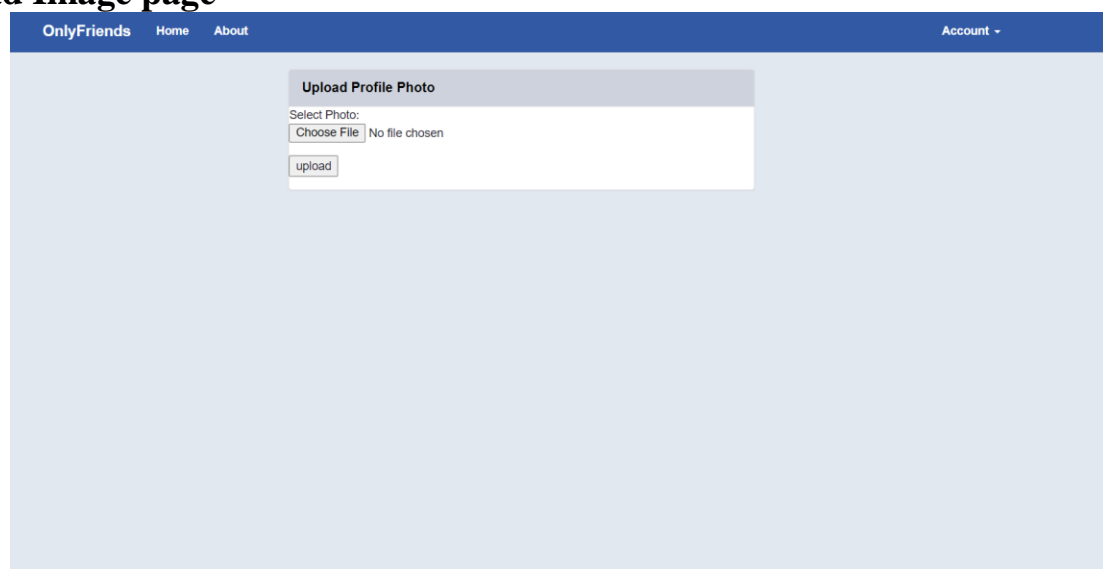

Figure 15: Home page

11. Upload Image page

Figure 16: Upload Image page

12. User Management page

Figure 17: User management page.

13. Status Management page

Figure 18: Status management page

5.3 Interface Activity Diagram
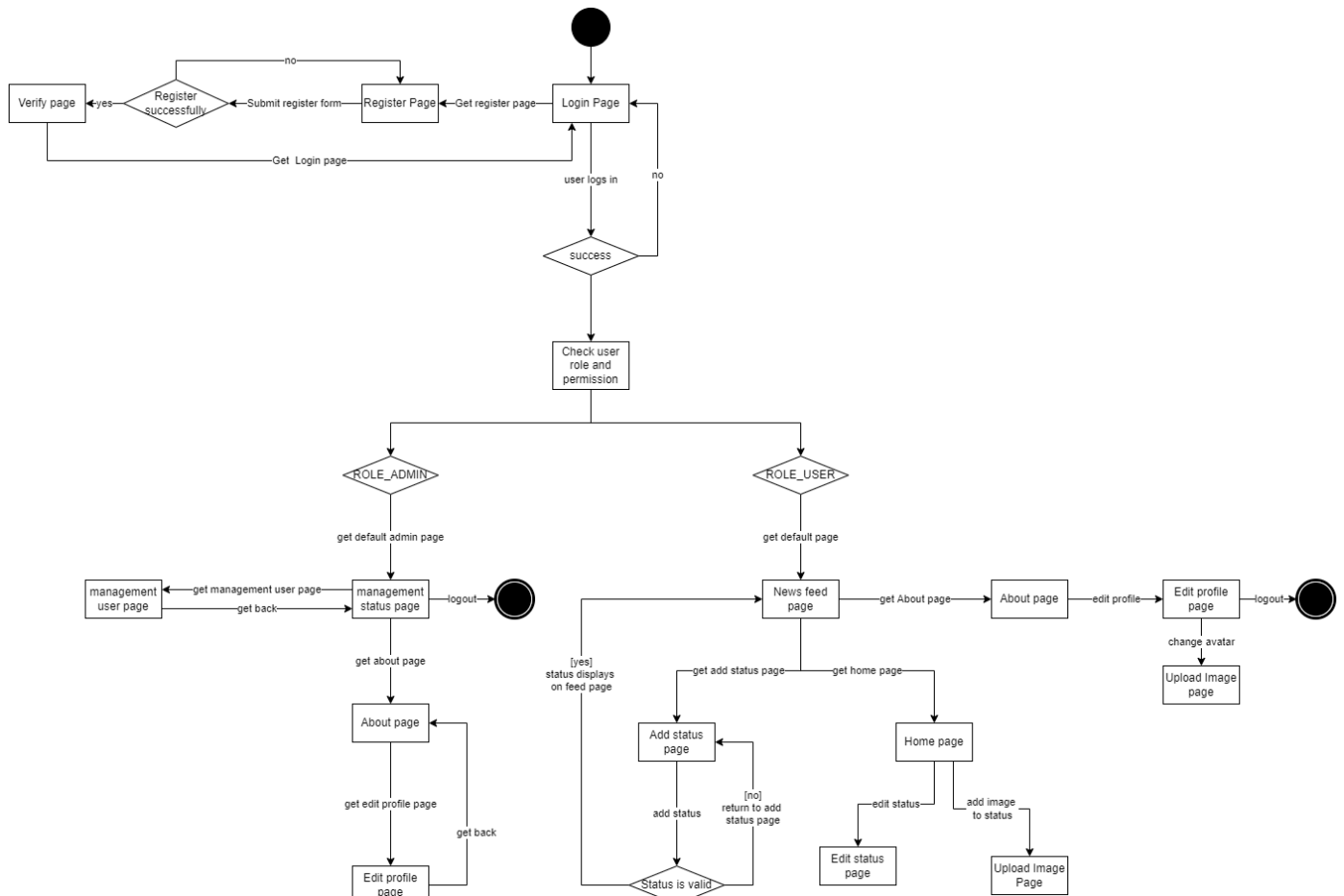
Figure 19: Interface activity diagram

6 Technologies

6.1 Intellij IDEA

IntelliJ IDEA enables developers to work more efficiently. They can write better code with consistent style, fewer bugs and exceptions, better referencing of the existing code base, better structure, and more.

6.2 Spring Framework

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.

6.3 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

6.4 Apache Tiles

Apache Tiles is a template composition framework. Tiles was originally built to simplify the development of web application user interfaces, but it is no longer restricted to the JavaEE web environment.

6.5 MySQL

Oracle MySQL Database Service is a fully managed database service that lets developers quickly develop and deploy secure, cloud native applications using the world's most popular open source database.

7 Project Implementation

For developing our web application, I built project on MVC architecture pattern using Spring framework. Apache Tiles allows to define page fragments with can be assembled into a complete page at runtime.

7.1   Structure
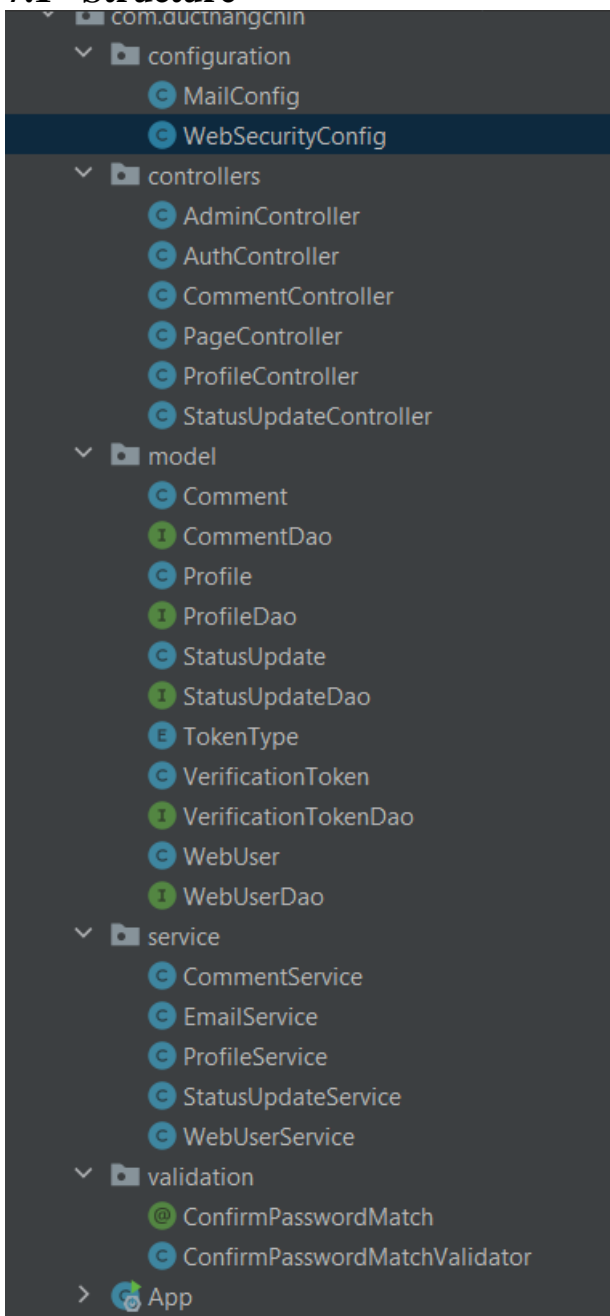
Figure 20: Backend directory structure.
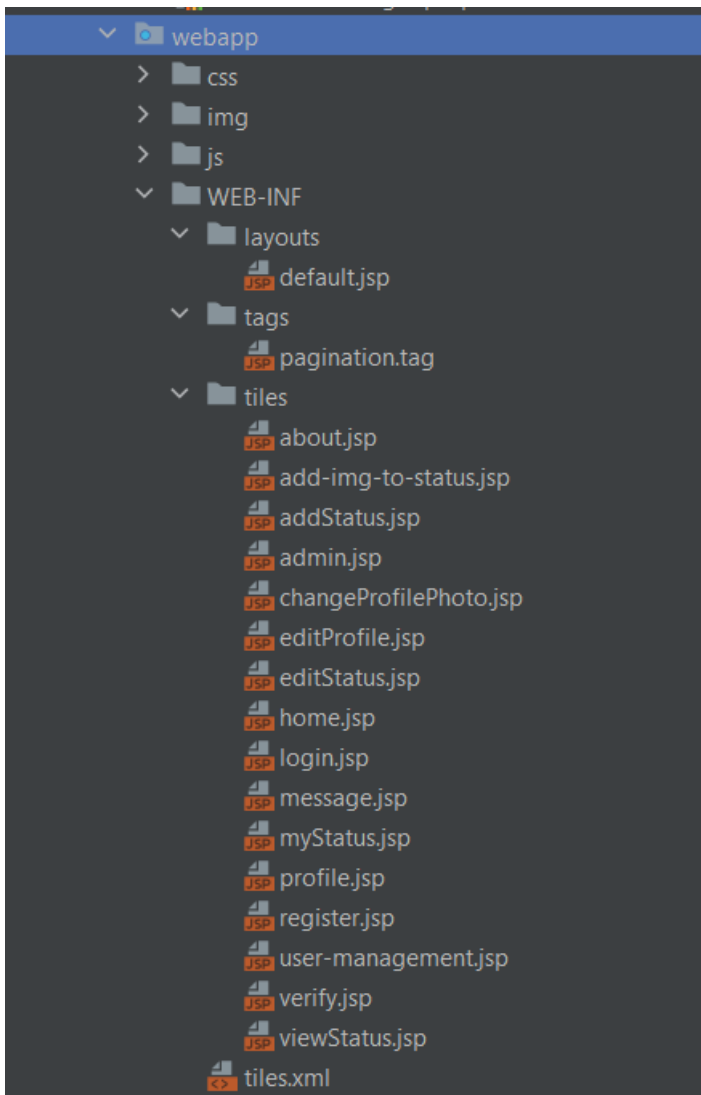
Figure 21: Frontend interface page.

```java
@SpringBootApplication
public class App extends SpringBootServletInitializer {

    public static void main(String[] args) { SpringApplication.run(App.class, args); }

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {
        return builder.sources(App.class);
    }

    @Bean
    public TilesConfigurer tilesConfigurer() {
        TilesConfigurer tilesConfigurer = new TilesConfigurer();

        String[] defs = {"/WEB-INF/tiles.xml"};

        tilesConfigurer.setDefinitions(defs);

        return tilesConfigurer;
    }

    @Bean
    public UrlBasedViewResolver tilesViewResolver() {
        UrlBasedViewResolver tilesViewResolver = new UrlBasedViewResolver();
        tilesViewResolver.setViewClass(TilesView.class);
        return tilesViewResolver;
    }

    @Bean
    PasswordEncoder getEncoder() { return new BCryptPasswordEncoder(); }
```

Figure 22: Spring Boot first configuration with some Beans defined.

7.2 Database

Tables_in_onlyfriends
verification
users
profile
post
▶ hibernate_sequence
comment

Figure 23: My Database list

I have id column with generated strategy AUTO, so MySQL generated hibernate_sequence
table:

```
1 •   show tables;
2 •   select * from hibernate_sequence;
3
```

next_val
69
69
69
69

Figure 24: Table hibernate_sequence

DB includes 5 connected table:

users:

id	email	enabled	password	role	username
1	ironman@mcu	1	$2a$10$qGz0BrfKEdps7XLSbcMbRu7utzbRVkrcv56LeySgLs6APZQHCmQvK	ROLE_USER	Tony Stark
5	spiderman@mcu	1	$2a$10$Sk4tLRoThPMX.6VAWPrjt.c.GpxDAxpu2xmgEuNkpC2X4f0Rw5UEK	ROLE_USER	Peter Parker
10	natasharomanoff@mcu	1	$2a$10$ldKvcgFAKZnP7H.UaXhlZOtGXnbbc7opem9K3eZmbwpnnKlUcvaM2	ROLE_USER	Natasha Romanoff
33	madisen1976@hotmail.com	1	$2a$10$Q7qO.j9J6yI0KCfO/xViau5NtnTLdebUEk3KTaXukq01ACHbV/XBi	ROLE_USER	Mary Oglesby
38	fred2009@yahoo.com	0	$2a$10$LD2xwaY0fr.rGyy20pPsZ.ihhoxjELxIkHuzwx19/N277n25PeJGO	ROLE_USER	Fredrick Mendoza
45	joan1998@hotmail.com	1	$2a$10$bm8Kd4eB2vXdtcOLbUIRruPufbK4hqbNJ.EOYp86lFkznV1UGkW6q	ROLE_USER	Susan Reynolds
56	chin@itmo	0	$2a$10$XouqduBvqqYq1BEgtlF9bO5gGzQfMsSKdlVPU88et3NRDlYjT/xf2	ROLE_USER	Chin Thang
59	trinhducthang201@gmail.com	1	$2a$10$f7p3aKlL6K7RCMErNYLgzui3Y64K4BqSJ0rRLudVZoYLMJx0B60de	ROLE_ADMIN	Duc Thang Chin
64	tyshawn2016@gmail.com	1	$2a$10$dkGN7KoHto43DIWhYn5Od.8ZHpTYJEZXifEs1d1sjafHHnU4.HB0K	ROLE_USER	Diana Cuthbert
NULL	NULL		NULL	NULL	NULL

Figure 25: Table users

post:

id	added	imgurl	text	owner_id
4	2021-12-22 04:48:18	/status/images.jpg	"My armor, it was never a distraction or a hobby, …	2
8	2021-12-22 05:03:22	/status/images (1).jpg	Spider-Man is a superhero created by writer-edito…	6
9	2021-12-22 03:05:52	/status/maxresdefault.jpg	<em>"What do you remember about your dad?"…	2
13	2021-12-22 15:30:23	/status/Natalia_Romanoff_(Earth-199999)_fro…	Years later, Romanoff was taking down a Ten Rin…	11
27	2021-12-22 23:16:34	/status/MV5BZWMyYzFjYTYtNTRjYi00OGExLWE…	No way home	6
32	2021-12-22 11:58:49	/status/download.jpg	<strong>What Does the Leftover Venom Goo Me…	6
36	2021-12-23 00:53:26	/status/g135fcbc4c6ab6a94bb6399004f98eb47…	<span style="font-family: 'Open Sans', Arial, san…	34
41	2021-12-23 01:04:30	/status/g4b98c02e58212a4c5443fbfcf475bf1b…	<span style="font-family: 'Open Sans', Arial, san…	39
48	2021-12-23 01:18:44	NULL	<span style="color: #555555; font-family: Arial, …	46
52	2021-12-23 01:23:02	/status/texts-from-my-dog1.jpg	<span style="color: #333333; font-family: Arial, …	39
68	2021-12-24 08:13:58	NULL	Blessed greater called so Place had a tree had mo…	65
NULL	NULL	NULL	NULL	NULL

Figure 26: Table post

comment:

	id	added	text	commenter_id	post_id
▶	1	2021-12-22 15:30:23	Internet tend to repeat predefined chunks as n...	11	4
	16	2021-12-22 18:15:15	Stark: "I want one"	2	13
	15	2021-12-22 17:36:29	A famous man once said, 'we create our own de...	2	8
	18	2021-12-22 18:18:32	This is an agent of some kind.	2	13
	63	2021-12-24 05:42:47	A famous man once said, 'we create our own de...	2	9
	20	2021-12-22 18:32:36	If You Expect Disappointment, Then You Can N...	6	4
	28	2021-12-22 23:28:50	wow yeah	6	27
	31	2021-12-22 23:55:19	We may think we have all the time in the world ...	6	9
	37	2021-12-23 00:53:48	Lorem ipsum dolor sit amet, consectetur adipisci...	34	36
	42	2021-12-23 01:07:38	My father used to say that it's never too late to...	39	9
	43	2021-12-23 01:09:10	Aliquam pulvinar sem fermentum enim aliquet ia...	39	36
	44	2021-12-23 01:09:53	Nullam at nibh mollis lacus porttitor varius.	39	13

Figure 27: Table comment

profile:

	id	about	address	avatarurl	fullname	phone	user_id
▶	2	Stark Industries	3426 Beeghley Street Heflin	/avatar/download (1).jpg	Tony Stark	256-201-2970	1
	6	Avenger Team	St. Petersburg, Russia	/avatar/spiderman.jpg	Peter Parker	8 (921) 943-05-96	5
	11	Assassin Spy	S.H.I.E.D	/avatar/maxresdefault (1).jpg	Natasha Romanoff	8 (921) 943-05-96	10
	34	Ophthalmic Medical T...	Lewisville, Texas	/avatar/female102235138208.jpg	Mary Oglesby	979-301-6142	33
	39	Hydrologist	Quincy, Massachusetts(MA)	/avatar/male1085128630737.jpg	Fredrick Mendoza	857-499-3755	38
	46	Library Assistant, Cl...	Chico, California(CA)	/avatar/female20171026040211670.jpg	Susan Reynolds	530-228-7916	45
	57	NULL	NULL	/avatar/default.png	Chin Thang	NULL	56
	60	Full Stack Developer	St. Petersburg, Russia	/avatar/photo1640308475.jpeg	Duc Thang Chin	8 (921) 943-05-96	59
	65	Materials Engineer	South Boston, Massachus...	/avatar/female20171026150268372.jpg	Diana Cuthbert	617-653-5914	64
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 28: Table profile

verification:

	id	expiry	token	token_type	user_id
*	NULL	NULL	NULL	NULL	NULL

Figure 29: Table verification

8 Conclusion

The idea of the original project is somewhat different from the product. I removed a few unnecessary features, and also revised the database and interface to be more suitable. After complete complementation, I launched the application and checked its performance, from which I am certain enough to conclude that project was successful.

9 Sources
https://www.jetbrains.com/idea/
https://spring.io/projects/spring-framework
https://tiles.apache.org/framework/index.html
https://getbootstrap.com/
https://www.mysql.com/