



# Software requirement specification document for project Blog

Version 1.0 approved

Prepared by Mohamed Alqblawi

TEAM RRRRR 404!

1/5/2022

# Table of Contents

<b>Table of Contents .....</b>	<b>2</b>
<b>1. Introduction.....</b>	<b>3</b>
1.1 Purpose.....	3
1.2 Product Scope .....	3
<b>2. Overall Description.....</b>	<b>3</b>
2.1 Product Perspective.....	3
2.2 Product Functions .....	3
2.3 User needs.....	3
2.4 Constrains .....	4
2.5 Assumptions and Dependencies.....	4
<b>3. System features and requirements .....</b>	<b>4</b>
3.1 <b>Functional requirements</b> .....	4
3.2 <b>Functional requirements</b> .....	4
3.2.1 Performance Requirements .....	4
3.2.2 Safety and security Requirements .....	4
3.2.3 Reliability .....	4
3.2.4 Usability .....	5
<b>4. System Design.....</b>	<b>5</b>
4.1 UML Diagram.....	5
4.2 High level use casse diagram of blog.....	5
4.3 Login dashboard.....	5
4.3.1 Use case diagram of login .....	5
4.3.2 Collaboration Diagram of login .....	6
4.4 Register .....	6
4.4.1 Use case diagram of Register.....	6
4.4.2 Collaboration Diagram of Register .....	6
4.5 Manage Posts .....	7
4.5.1 Use case diagram of MP .....	7
4.5.2 Collaboration Diagram of MP.....	7
4.5.2.1 MP-create post collaboration diagram .....	7
4.5.2.2 MP-edit post collaboration diagram.....	7
4.5.2.3 MP-delete post collaboration diagram .....	7
4.6 Manage Comments MC .....	7
4.6.1 Use case diagram of MC.....	7
4.6.2 Collaboration Diagram of MP.....	8
4.6.2.1 MC-Add Comment collaboration diagram .....	8
4.6.2.1 MC-Edit Comment collaboration diagram.....	8
4.7 Manage Reacts MR.....	8
4.7.1 Use case diagram of MR.....	8
4.7.2 Collaboration Diagram of MR.....	8
4.7.2.1 MR -Add Comment collaboration diagram .....	8
4.7.2.1 MR -Edit Comment collaboration diagram.....	8
4.8 Search.....	8
4.8.1 Use case diagram of search.....	8
4.8.2 Collaboration Diagram of search .....	9
4.9 Entity Relationship Digram .....	9
4.9.1 List of entities .....	9
4.9.2 Relationship with entities.....	9
4.9.2.1 Posts – Comments.....	9
4.9.2.2 Comments itself .....	9
4.9.2.3 Posts – reacts .....	9
4.9.2.4 Posts – Postreacts .....	9
4.9.2.5 Reacts – Postreacts.....	9
4.9.2.6 Users – Postreacts .....	10
4.9.2.7 Users – Posts .....	10
4.9.2.8 Users – Comments .....	10
4.9.2.9 Categories – Posts .....	10
4.9.2.10 Posts – postcategories .....	10
4.9.2.11 Categories – postcategories.....	10
4.9.2 ER-Diagram .....	11

# 1. Introduction

The following section provides overview the software requirements Specifications (SRS) for the blog 4All.

## 1.1 Purpose

The purpose of this document is to specify the requirements that must be fulfilled by the blog aggregation, preservation, management, and dissemination system which will be developed in the Blog 4All project. Thereby, the requirements describe functional and qualitative necessities of the system but do not postulate any demands regarding the design of the system.

## 1.2 Product Scope

the description of requirements should help the intended users to express their needs. Therefore, the requirements must be understandable by system users. On the other hand, the specification of requirements facilitates the development of the system by providing measurable demands. It narrows down the number of possible designs. Therefore, the specification must be understandable by developers and the requirements must be formulated in a way that allows the assessment of the fulfilment of requirements.

# 2. Overall Description

## 2.1 Product Perspective

The software described this SRS the software for a blog 4All. The system merges various hardware and software elements.

## 2.2 Product Functions

System users will be able to put comments, reacts if the type of plan basic is he is premium he can create post twice a month.

System user and admins will be able to search for any and put comments and reacts.

System admin will be able to create with category, edit, delete posts, view statistics and pages, change format, theme, and sittings.

## 2.3 User needs

USER	USER DESCRIPTION	USER NEEDS
Author (Admin)	Also called 'admin'; refers to the person with full access of the website's admin interface.	<ul style="list-style-type: none"><li>- The admin should be able to add /remove registered website superusers /admins</li><li>- The admin needs to have access to the message's eft by the visitors through the "Contact Me" form.</li><li>- The admin must be able to post and edit content (text, images, videos, links) on the website</li></ul>
Visitor (users basic)	Also called 'reader'; refers to any person who visits and views the webpage	<ul style="list-style-type: none"><li>- The visitor must be able to seamlessly view and navigate all the webpages of the website regardless of browser or device used</li><li>- The visitor should be able to contact the author for inquiries or comments through a "Contact Me" form</li></ul>
Visitor (users premium)	the same as the previous with create post twice a month	the same as the previous

## 2.4 Constraints

The software is constrained by the shared internet connection which should be reliable. Also, the capacity of the database, so it should have a big free space.

## 2.5 Assumptions and Dependencies

- **Assumptions:**

- a) **express Development:**

The backend code of the blog will be developed and maintained using the NodeJS express framework. The system will utilize a relational database to store and manage its data via the built-in sequelize ORM. At the time of development, the express framework version to be used is express 4.18 which features an admin interface.

- b) **react Framework:**

React framework will be utilized for the styling and dynamic features of the website for it to be more responsive and mobile-compatible. At the time of development, react 18 or higher will be utilized for the front-end development of the website.

## 3. System features and requirements

### 3.1 Functional requirements

The primary functional requirements of the blog, which are based on the user needs, are as follows:

- a) Viewing and Navigating the Site. Visitors should be able to view the site as intended through a mobile or desktop browser
- b) Contacting the Author. Visitors should be able to contact the author through contact form.
- c) Adding, Editing, Removing Posts. The author should be able to add and/or edit blog posts specifically on the Home.
- d) Administrators and users can use the search feature.

### 3.2 Non- Functional requirements

#### 3.2.1 Performance Requirements

The system should be reliable and compatible with all modern browsers.

#### 3.2.2 Safety and security Requirements

- The system must handle safe login and logout through session and hashing technology should be used handle the secure login for users.
- The password must be more than 8 characters.

#### 3.2.3 Reliability

This program should be reliable and provide catching of exceptions so that unintended results do not occur such as system crashes, or data validation failures.

### 3.2.4 Usability

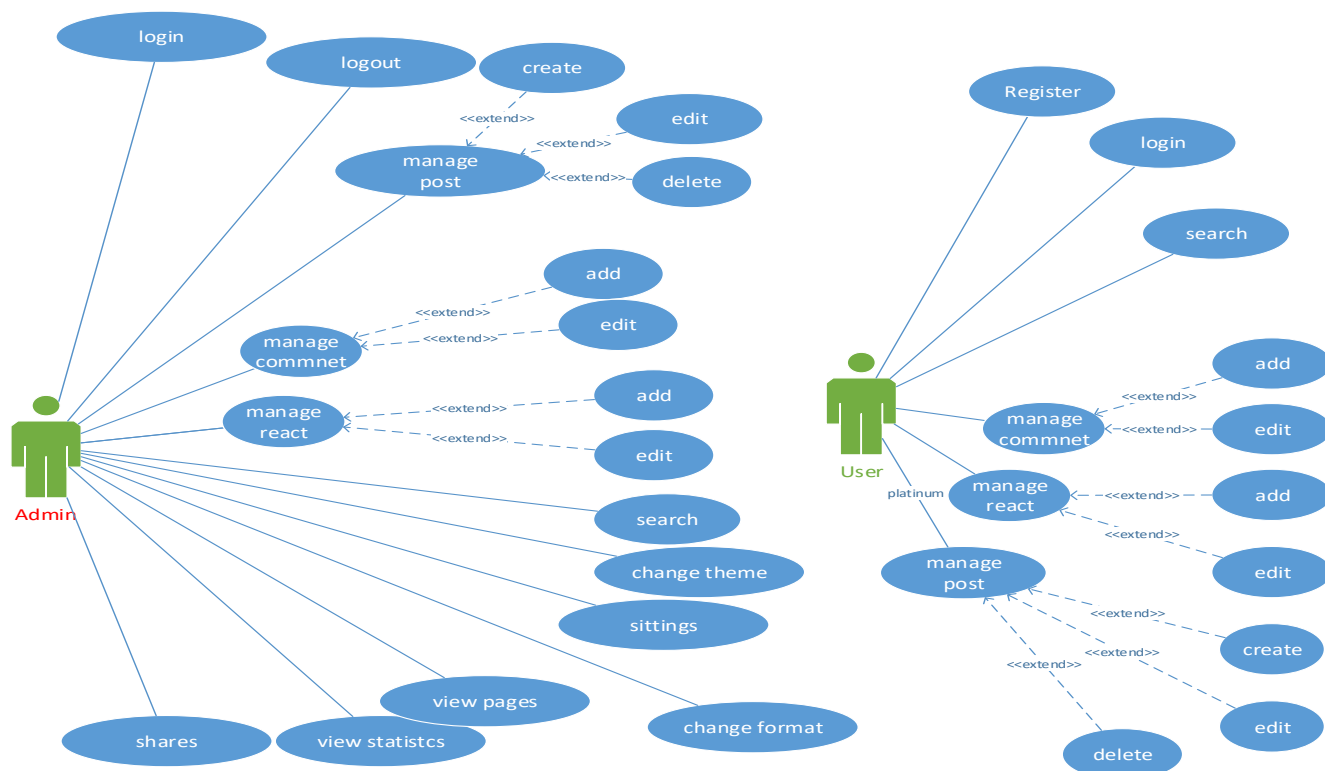
The system should have user-friendly interface.

## 4. System Design

### 4.1 UML Diagram

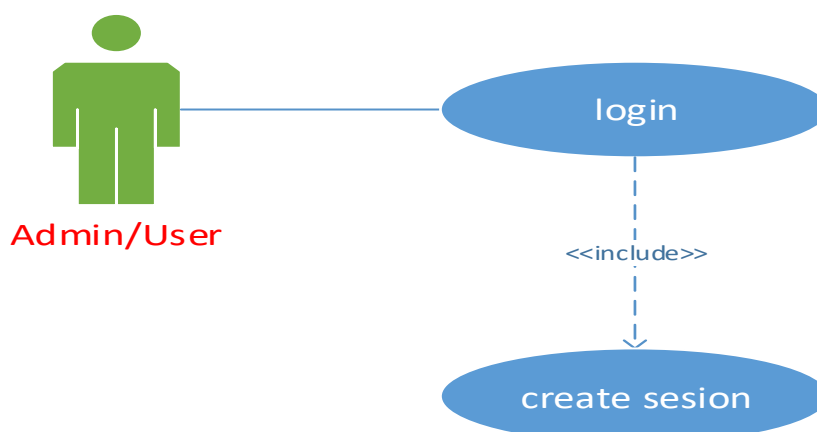
For this project only 3 diagram of UML diagram will be used. They are Use case, collaboration, and entity relationship diagram.

### 4.2 High level use case diagram of Blog

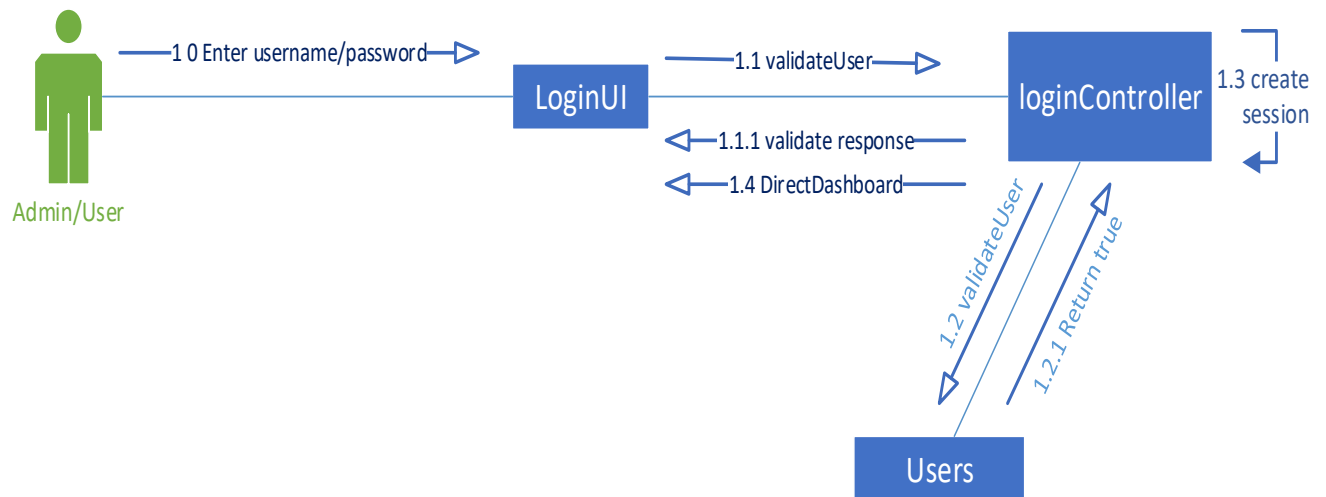


### 4.3 Login dashboard

#### 4.3.1 Use case diagram of login

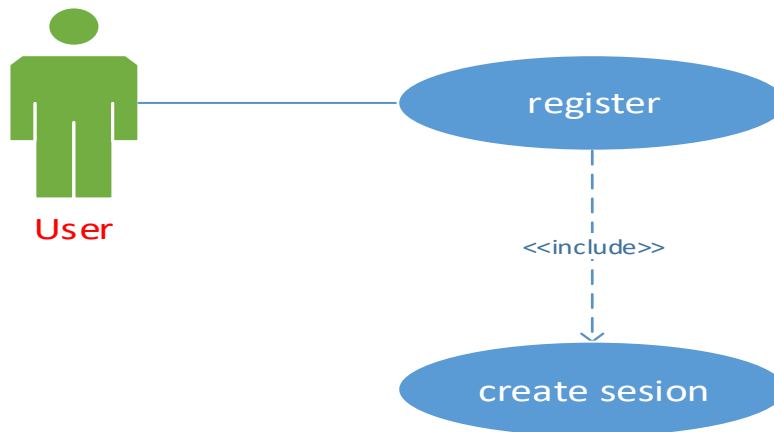


### 4.3.2 Collaboration Diagram of login

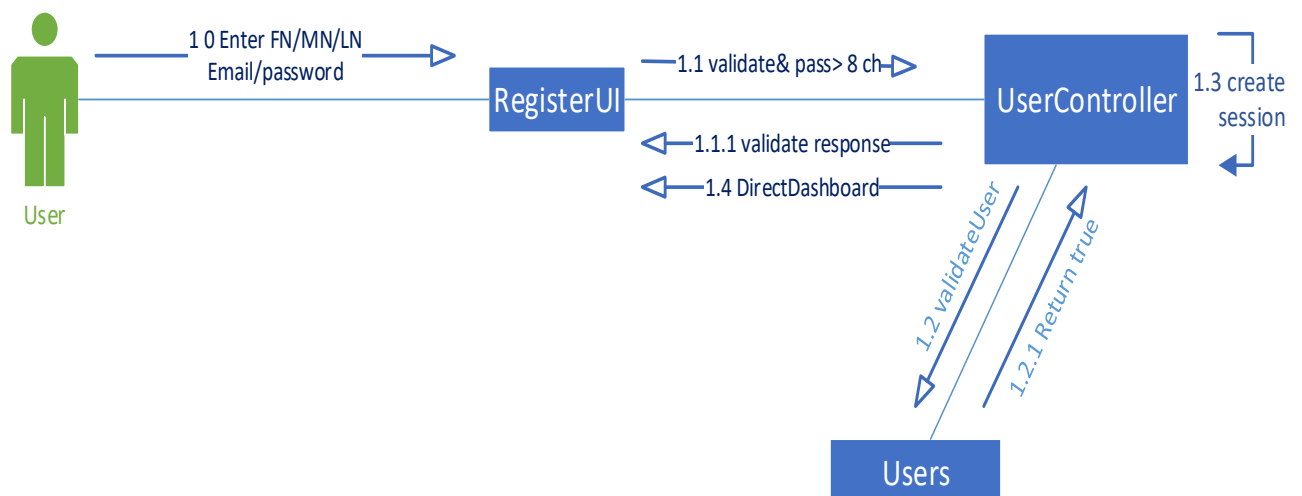


## 4.4 Register

### 4.4.1 Use case diagram of register

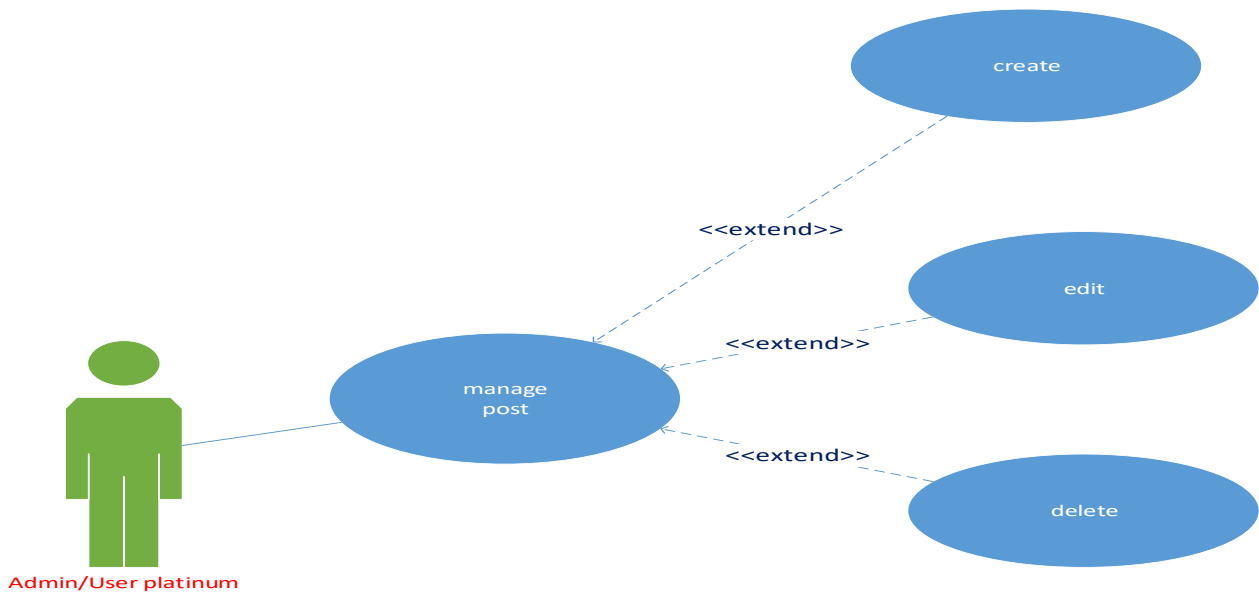


### 4.4.2 Collaboration Diagram of register



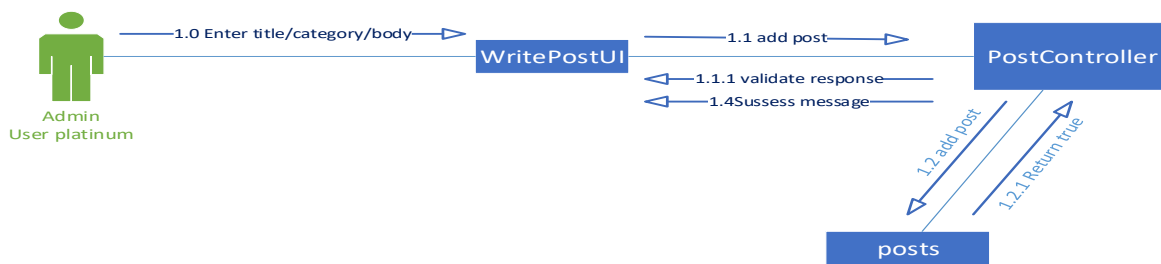
## 4.5 Manage Posts

### 4.5.1 Use case diagram of MP

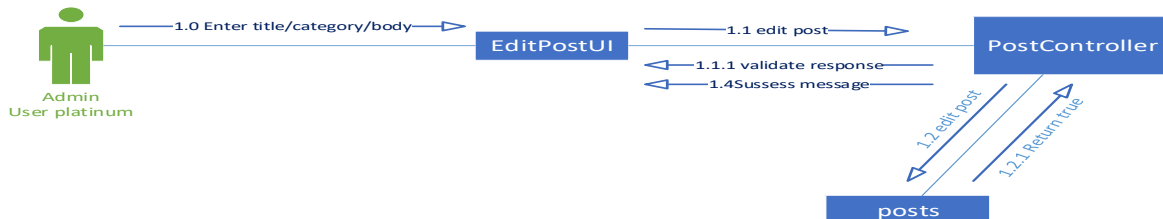


### 4.5.2 Collaboration Diagram of MP

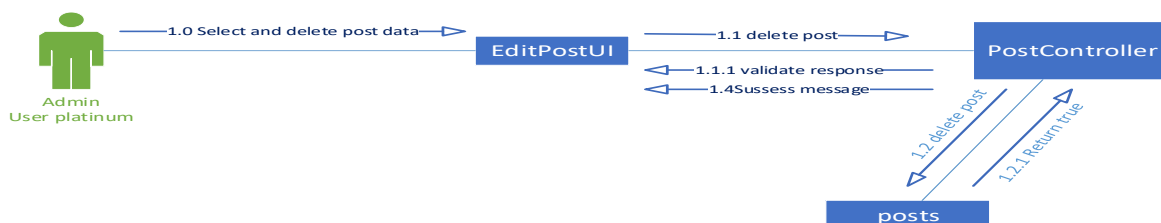
#### 4.5.2.1 MP-create post collaboration diagram



#### 4.5.2.2 MP-Edit post collaboration diagram



#### 4.5.2.3 MP-Delete post collaboration diagram



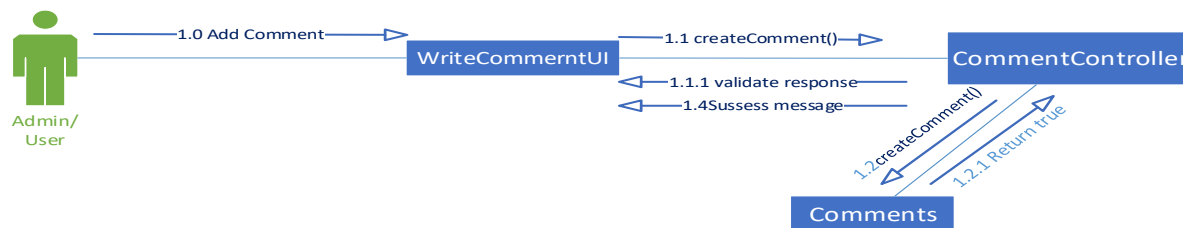
## 4.6 Manage Comments MC

### 4.6.1 Use case diagram of MC

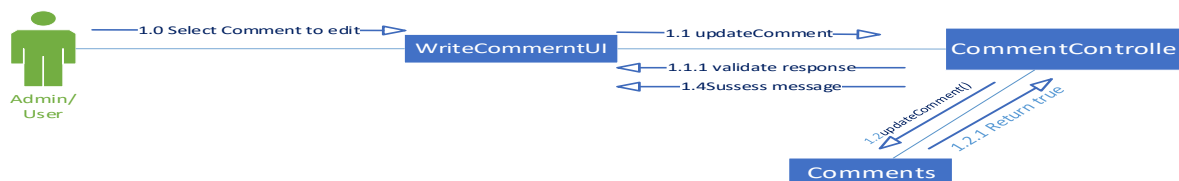


## 4.6.2 Collaboration Diagram of MC

### 4.6.2.1 MC-Add Comment collaboration diagram

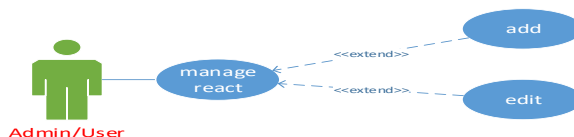


### 4.6.2.2 MC-Edit Comment collaboration diagram



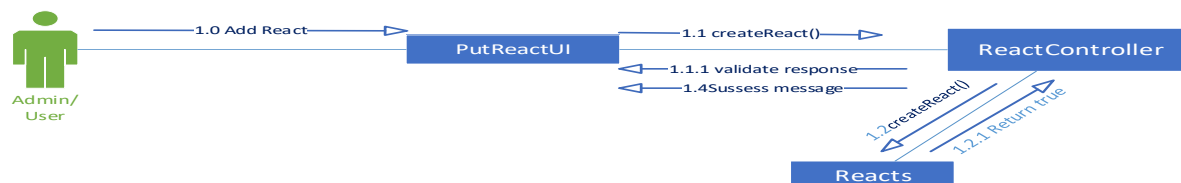
## 4.7 Manage Reacts MR

### 4.7.1 Use case diagram of MR

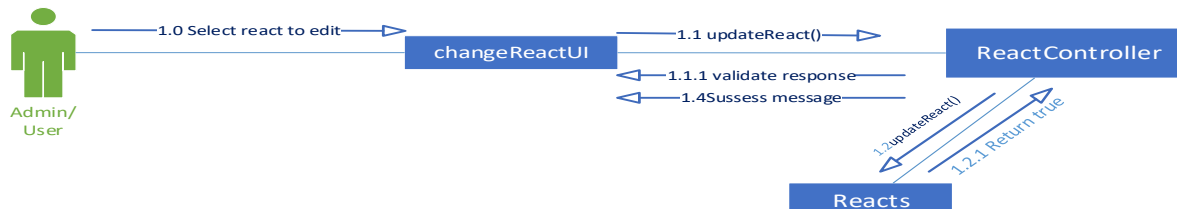


### 4.7.2 Collaboration Diagram of MR

#### 4.7.2.1 MR-Add react collaboration diagram

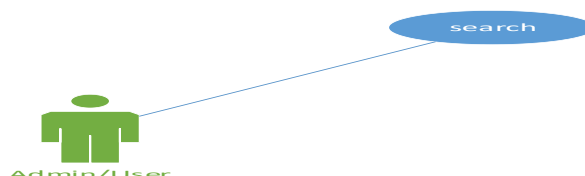


#### 4.7.2.2 MR-Edit React collaboration diagram



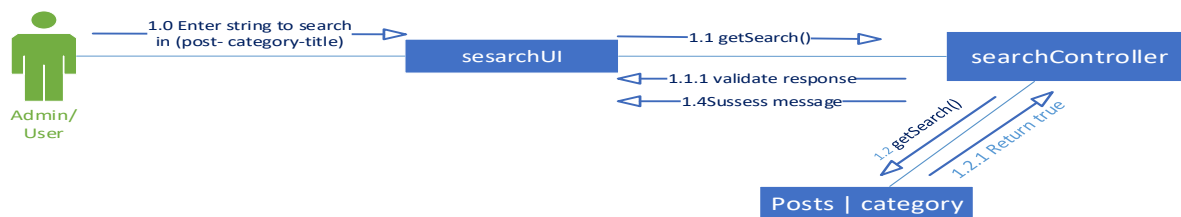
## 4.8 Search

### 4.8.1 Use case diagram of search



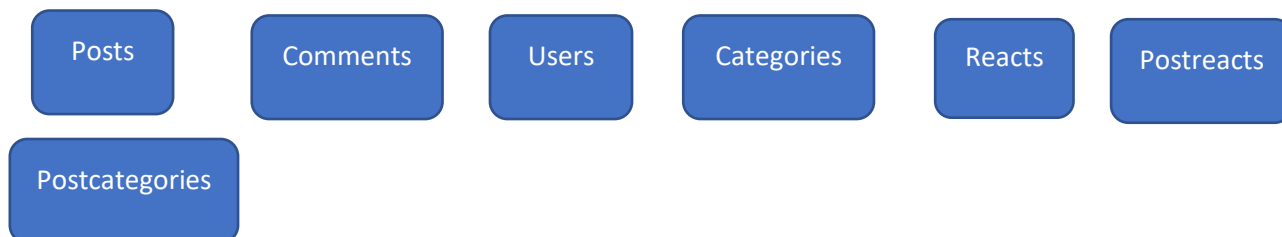


## 4.8.2 Collaboration Diagram of RC



## 4.9 Entity Relationship Diagram

### 4.9.1 List of Entities



### 4.9.2 Relationship with entities

#### 4.9.2.1 Posts – Comments



- Comments have one posts.
- One post has many comments.

#### 4.9.2.2 Comments itself



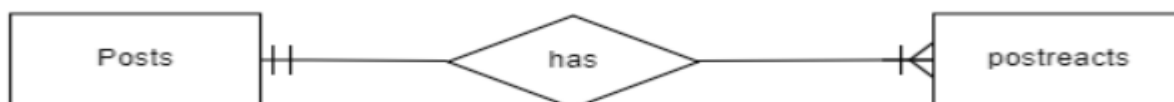
- One comment has many comments.

#### 4.9.2.3 Posts-Reacts



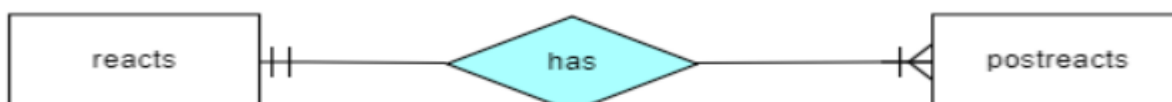
- one post has many reacts.
- One reacts has many posts.

#### 4.9.2.4 Posts-Postreacts



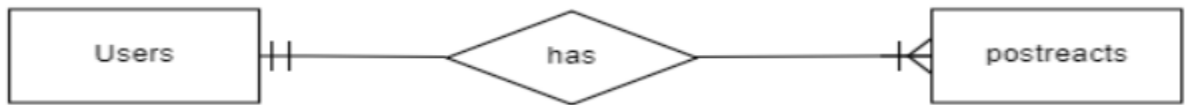
- one post has many Postreacts.
- One Postreacts has one posts.

#### 4.9.2.5 Reacts-Postreacts



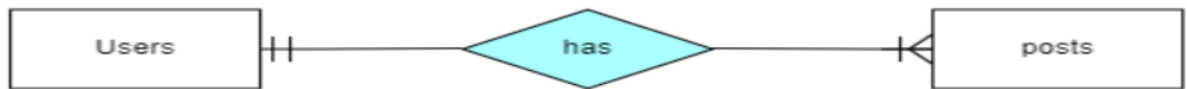
- one reacts has many Postreacts.
- One Postreacts has one react.

#### 4.9.2.6 Users-Postreacts



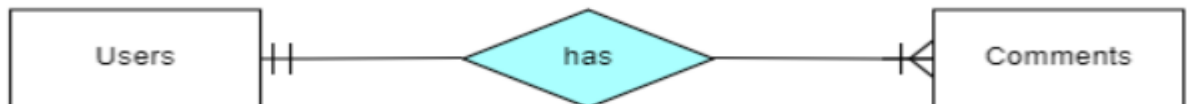
- one Users has many Postreacts.
- One Postreacts has one Users.

#### 4.9.2.7 Users-Posts



- one Users has many Posts.
- One Posts has one Users.

#### 4.9.2.8 Users-Comments



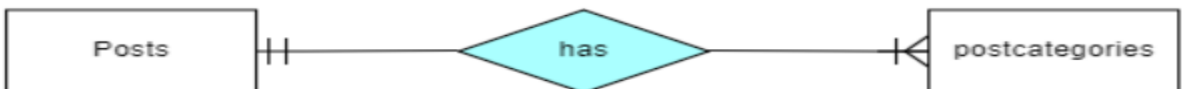
- one Users has many Comments.
- One Comment has one Users.

#### 4.9.2.9 Categories-Posts



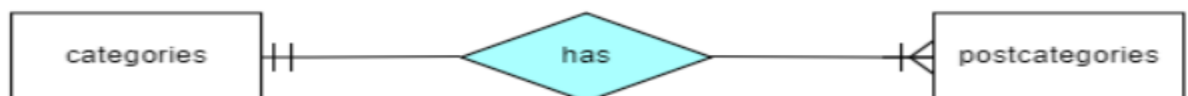
- one post has many categories.
- One category has many posts.

#### 4.9.2.10 Posts- postcategories



- one Post has many Postcategories.
- One Postcategories has one Post.

#### 4.9.2.11 categories- postcategories



- one category has many Postcategories.
- One Postcategories has one category.

### 4.9.3 ER-Diagram

