Software Requirements Specification

for

Project Ovijog

By Team Connector

Noakhali Science and Technology University (NSTU)

Software Requirements Specification

for

Project Ovijog

Project Mentor

Md. Rafid Mostafiz
Lecturer
Institute of Information Technology
Noakhali Science and Technology University

Prepared by

Md. Armanur Rashid **ASH1925013M**
Arnab Dey **ASH1925024M**
Nayeem Khan **ASH1925027M**

Submission Date: 22.08.2022

Table of Contents

List of Tables

List of Figures

1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete **"Project Ovijog Application".** By presenting the problem statement in detail, the purpose of this document is to collect, evaluate, and provide a deeper understanding of the whole "**Project Ovijog**" application. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs. The detailed requirements of the **"Project Ovijog"** are provided in this document.

1.1 Problem Statement

Among the public universities, Noakhali Science and Technology University is very reputed. There are about 5500 to 6000 students studying here. There are many teachers teaching here. Many students of NSTU live in university halls. Most of the students live in Maijdee City. Students travel by university bus from Maijdee. Student face many problems in their day to day life. Some of those problems are directly connected to either higher authority of the university or the department they are admitted to. Moreover, there are many problems that are not disclosable publicly. But there is no platform in NSTU where the students can submit their problems without disclosing their identity.

Those who live in hall also faces various problem. Hall does not have good food facilities for students. The food in the campus hall canteen is very bad and the price is very high. Hall washroom is not cleaned properly. Besides, on rainy days, water accumulates on the street in the front of the hall. Sometimes various reptiles enter the hall including snakes.

There is no central cafeteria on campus. Students fell ill after eating unhealthy food from the local hotel at lunch time. There is no adequate lighting system in the campus. Outsiders come to the campus due to which girls are subjected to eve teasing in their own campus. Our respected teachers also face various problems. They face many problems including transportation, food problems, housing problems. They also do not always reach the administration about their problems.

Students can't always go to the authority with their problems. Many times, they go the department chairman with their problems. Chairman then asked them to submit a written application. He himself discussed the problem with the authority. They also often post in different groups on Facebook. These posts are usually not seen by the authorities. Due to which they are unable to report their problems to the authorities and fail to get their various rights and solve the problems. If all the students, teachers and authorities of the campus can be brought on one platform, then if everyone will easily know about the problem and the authority will be aware about the problem.

1.2 Purpose

The purpose of this document is to present a detailed description of the **Project Ovijog Application**. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to **Institute of Information Technology, Noakhali Science and Technology University** for its approval.

1.3 Project Scope

Our project name is “Project Ovijog”. This project can be used to post someone's problem. One can post their issue here and report any false posts as well. Most reported post will be deleted by the admin. The ability to vote for or against a post is also provided by this platform. It will mainly help to make a post trending. Problem filtering by category is an additional function. The user will feel smooth to use as a result. How many posts are made, how many are solved, and the total number of postings will be considered carefully.

1.4 Glossary

This section provides definitions for all document names, acronyms, and abbreviations. The application domain's terms and concepts are defined.

API	Application Programming Interface
SRS	Software Requirement Specification
UI	User Interface
XML	Extensible Markup Language
AJAX	Asynchronous JavaScript and XML
PHP	Hypertext Preprocessor
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language

1.5 References

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. Software Engineering 9th Edition by Lan Sommerville.
3. Requirements Engineering Fundamentals by Klaus Pohl.
4. Database System Concepts 6th Edition by Abraham Silber Schatz.

1.6 Overview

In the beginning, this document will give an overview of the project. The scope, purpose, problem statement, glossary, and references will all be covered. The system's stakeholders would then be mentioned after that. Implementation and Design Constrains part will describe User Interface Technology and Server-Side Technology. User interface design is the visual organization of the parts of a website or technical product that a user might interact with. When an application is used, behind-the-scenes operations are referred to as server-side development. The next section is the requirement specification. The functional requirements, data requirements, performance

requirements, capacity requirements, maintainability and supportability requirements, security requirements, usability requirements, requirements for human integrity, and legal requirements have all been covered in this section. Requirement Engineering Process is the next step. Requirements Engineering (RE) determines software requirements according to customer requirements or needs. Requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management are all parts of the requirements engineering process. The paragraph about requirement validation follows that. This section includes many forms of requirement validation, such as reviewing, testing, and simulating the requirements. The use case diagram that will demonstrate interactions between the use case and the stakeholder. All use cases will be described in the use case description. Following that, an activity diagram will mostly show how each use case has been implemented. The Appendix is the last section. It will discuss how the project's requirements were prioritized.

2. Stakeholders and Characteristics

2.1 Students

It is clear that students are the most important stakeholders. The most crucial factors in guaranteeing the quality of higher education are the caliber of the academic staff and the study curricula; the most crucial activities are the administration of the learning process and the delivery of the study curricula.

2.2 Teachers

The individuals who really carry out the task of educating kids are teachers. In their classrooms, they have a significant degree of power over what and how pupils are taught. Teachers believe they have the competence to know what to teach and how it should be taught to their pupils, which is another reason they feel strongly about education.

2.3 Academic Officials

The principal, associate principals, and other university executives are considered university administrators. They bear a heavy burden for the efficient and successful operation of the University. But in order for colleges to represent the ideals of their communities, university managers frequently need to listen to various stakeholders and solicit their opinions when making strategic decisions.

2.4 Developer

To be classified as stakeholders, persons who have a legitimate interest in the product or project must have legitimate interests. Because the developers have too much at stake, they are deemed stakeholders because they have a legitimate interest. As stakeholders, developers are responsible for software delivery and estimation on time.

3. Design and Implementation Constrains

We have employed design and implementation constraints to ensure the success of this project. It also refers to a tool that allows developers and testers to inspect and interact with the application's user interface (UI) elements.

3.1 User Interface Technology

The visual layout of the components that a user could interact with in a website or technical product is referred to as user interface design, or UI design. In other terms, it is a website's visual design.

3.1.1 Programming Language

JavaScript is an ECMAScript-compliant high-level, frequently just-in-time compiled language. It has first-class functions, dynamic typing, and prototype-based object orientation. It's multi-paradigm, allowing you to program in event-driven, functional, or imperative styles.

JavaScript XML is abbreviated as JSX. It's just a JavaScript syntactic extension. It allows us to create HTML directly in React (within JavaScript code). It is straightforward to generate a template in React using JSX, but it is not a simple template language; instead, it has all of JavaScript's capability.

It is faster than standard JavaScript because it optimizes when converting to standard JavaScript. Rather than dividing the markup and functionality in different files.

3.1.2 CSS Framework

Cascading Style Sheets (CSS) is a language for specifying the appearance of a document written in a markup language like HTML. Along with HTML and JavaScript, CSS is a key component of the World Wide Web. Semantic UI is a website using UI component framework. Developers may use Semantic UI to create websites with quick and clear HTML, as well as a fully mobile responsive experience. Semantic UI offers a React-integrated version called Semantic UI React, which includes the following functionalities:

- jQuery Free.
- Declarative API.
- Augmentation.
- Shorthand Props.
- Sub Components.
- Auto Controlled State

3.1.3 Bootstrap

Bootstrap is a sizable repository of reusable code that comes in handy for developers. It is a JavaScript, CSS, and HTML frontend development framework. Using Bootstrap, web developers and designers can easily create fully responsive websites. It might be regarded as the most well-liked CSS framework for creating mobile-first and responsive applications.

3.2 Server-Side Technology

Server-side development refers to the actions that take place behind the scenes when an application is used. It primarily focuses on databases, scripting, website architecture, backend logic, APIs, and Servers.

3.2.1 PHP

The general-purpose programming language (GPPL) is PHP. It is mostly used as a server-side scripting language for the creation of websites. Web development is also simplified by the PHP frameworks. This framework makes it easier to reuse existing code and eliminates the need to create lengthy, intricate code for web apps. The majority of PHP frameworks are free source and simple to use. Because PHP is open-source and cost-free, developers may install it easily and utilize it right away. All major operating systems, including Windows, Unix, Linux, etc., support PHP. Web applications created using PHP may simply operate on any platform. PHP makes a safe connection with databases and connects to them with ease. It features an integrated module that may be used to quickly connect to the database. The primary purposes of the PHP framework are to simplify the construction of web applications and to automatically maintain the code. The built-in tools and features of PHP frameworks make it simpler to defend online applications from outside assaults and security risks.

3.2.2 Database Server

A piece of hardware that runs database software is called a database server. Users and companies may store, manage, retrieve, update, or modify files, information logs, and other types of digital data with the use of database software. Large volumes of digital information may be easily stored, arranged, and maintained using database servers. Database servers perform by combining a database management system with memory and storage capacity for databases (DBMS).

4. Requirement Specification

All the requirements based on the elicitation process are described in this section.

4.1 Functional Requirement

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data the system should hold and the interfaces with the user.

4.1.1 User login and register

FR-1	User can register a new account and login to a registered account.		
Description	User should register his/her account for the first time and be able to log in to the account which was registered once. Already registered users will not face this stage.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	High

4.1.2 Create new post

FR-2	User can create a new post.		
Description	If the user faces any problem, user can post the problem by clicking on Add new post option.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	High

4.1.3 Edit post

FR-3	User can edit an already created post.		
Description	After the user posts from his account, the user can edit his/her post.		
Stakeholders	Students, teachers, Academic Officials	**Priority**	Low

4.1.4 Delete post

FR-4	User can delete an already created post.		
Description	After the user posts from his account, the user can permanently delete his/her post.		
Stakeholders	Students, teachers, Academic Officials	**Priority**	Low

4.1.5 Show Trending problems

FR-5	User can see the trending problems.		
Description	There will be posts on different types of issues. The trending issues will be shown on the user's Homepage.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	High

4.1.6 Quick and Easy Search

FR-6	Search posted problems according to user preference.		
Description	Users can search a specific problem using a search box and the system will provide results for search results.		
Stakeholders	Students, teachers, Academic Officials	**Priority**	Medium

4.1.7 Comment

FR-7	User can comment on posts.		
Description	When someone posts, other users can comment on that post. One can reply others comments.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	Low

4.1.8 Voting

FR-8	User can vote on posts.		
Description	When user's posts, if other users also face it or liked the post or the post is true, then other users will support the post by upvoting. But if other users think the post is fake then they can express their opinion by downvoting.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	high

4.1.9 Filtering problems by category

FR-9	Filtering problems by category.		
Description	Users can do category wise filtering. In this case, the system will show the posts of that category after filtering.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	Medium

4.1.10 Report

FR-10	Report against fake posts.		
Description	If someone has posted a fake post, or used indecent language in comments, then anyone can report against the user.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	Low

4.1.11 Mark as solved

FR-11	User can mark his post as Solved after the problem is solved.		
Description	If the problem posted by the user is solved by the authority, then the user will mark his post as solved.		
Stakeholders	Students, Teachers, Academic Officials	**Priority**	High

4.1.12 Admin Control

FR-12	Admin Control		
Description	There will be admin panel, they can control everything after logging in admin panel. Action will be taken by admin against anyone who posts fake or bad comments.		
Stakeholders	Admin	**Priority**	Medium

4.1.13 Log out

FR-13	User log out from their account.		
Description	The user will be able to log out of his/her account at the end of his need. Users will need to log in again for later use.		
Stakeholders	Students, Teachers, Academic Officials, Admin	**Priority**	Medium

4.2 Data Requirement

The Data Requirements section of the SRS provides information on the data used by the software application/system.

4.2.1 Password Protection

DR-1	Users' password will be protected		
Description	Hashing will be used to keep users' passwords protected.		
Stakeholders	Developers	**Priority**	High

4.3 Performance Requirement

It is important to maintain the performance of the software system. To ensure performance we maintain these steps:

4.3.1 Higher speed and low latency while searching, posting, and filtering

PR-1	Search, Post, and Filter with less buffering and load faster.
Description	Searching and filtering result will load faster and less buffering. It will take less time whenever user post something.

4.3.2 Capacity Requirements

This system can load up to thousands of user's information and thousands of posts information.

4.3.3 Safety Critical Requirement

There are no safety-critical requirements for our project.

4.3.4 Robustness or Fault-Tolerance Requirements

There are no Robustness or Fault-Tolerance Requirements for our project.

4.4 Maintainability and Supportability

4.4.1 Maintenance Requirements

MR-1	Make the code maintainable.		
Description	Code must be developed so that it can be modified later and will be readable.		
Stakeholders	Developers	**Priority**	High

4.4.2 Supportability Requirements

This system meets Testability, Maintainability, Compatibility, Configurability, Serviceability, and installation ability which are related to supportability requirements.

4.5 Security Requirements

Securing information is much more important for a system to get users' dependability. Here are some of them:

4.5.1 Access Requirements

For accessing information, the system will use some authorization techniques to ensure that the correct data is used by the correct user.

4.5.2 Integrity Requirements

Integrity requirements refer to a security system that ensures an expectation of data quality. It also ensures that all data of the system would never be exposed to malicious modification or accidental destruction. For preventing anonymous access to user passwords, the system will use an encryption technique called the Hash Function for encrypting user passwords.

4.5.3 Privacy Requirements

Privacy requirements are enhanced to protect stakeholders' privacy. In this way, all data or a partial part of data is going to be disclosed according to the system's privacy policy. To ensure privacy, the central database should be protected by the anonymous. Users are permitted to get access to those data which are being associated with them which can be ensured by the user login system.

4.6 Usability and Human Integrity Requirements

This system will provide a more user-friendly environment

4.6.1 Ease of Use Requirements

Our system will be easier to use by any type of people and they don't need any training to use the system.

4.6.2 Accessibility Requirements

To get access to the application, the application provides authorization/authentication. This application uses various modules.

SR-1	The system provides security strategies.
Description	The system is designed in a way that allows all modules to access a mechanism that provides security services.

4.7 Style Requirements

There are no style requirements in our system.

4.8 Legal Requirements

Legal requirements normally refer to the terms and conditions or privacy policy of any organization. The terms and conditions of our application are that no third-party software or person is allowed to use our data for their business purpose.

5. Requirement Engineering Process

Requirements Engineering (RE) determines software requirements according to customer requirements or needs. Requirements engineering process includes requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management.

5.1 Requirement Elicitation Techniques

Requirements elicitation is the practice of researching and finding system requirements for users, customers, and other stakeholders also referred to as "requirement gathering". Requirement elicitation can be done by contacting participants directly or by doing some research, analysis, and testing.

5.1.1 Hold Interviews

We hold discussions that can be held individually or with a small group of participants. They are an effective way to access services without spending a lot of time with participants because we meet with people to discuss only certain important requirements of this program. Negotiations are useful for obtaining individual requirements for members in organizing workshops where those members of the program come together to resolve any issues or conflicts. We mainly perform our interviews based on some specific criteria.

5.1.2 Perform Document Analysis

Existing documentation can help to show how systems are currently operating or what they are what I should do. Documents include written information about current programs, business processes, needs specifications, and competitor research. Review once textual analysis can help determine which performance should remain and functionality that isn’t in use. After the existing document. In analysis, we found several problems with the existing system.

5.1.3 System Interface Analysis

The first thing to do is to identify which systems the system-to-be shall communicate with. It could be a server on the Internet, a piece of software on the same host as the system-to-be, some hardware, or something completely different.

5.1.4 Distribute Questionnaires

The questionnaire is a useful way to investigate styles, changes in attitudes and users' ideas, and user satisfaction with priorities and preferences. Our lists of questions were as short as possible. The respondent may be tired or frustrated. Had a basic reason for all the questions as well as group

the topic areas together for the respondent to focus on. The main advantage of this survey responses was that they were collected in the usual way. Information was summarized by a large number of people.

5.2 Requirement Validation

Requirement validation ensures that the requirements are correct and reflect the quality you want from this program. In the beginning, our requirements looked good but when we read them and tried to work with them, they came out having ambiguities and gaps.

5.2.1 Review the Requirements

Negative peer review, especially the type of rigorous review called evaluation, is unique among the highest quality software processes available. We had a team of reviewers representing different perspectives and carefully examined written needs, analysis models, and related information on disability.

5.2.2 Test the Requirements

The test creates another view of the requirements. We also performed writing tests regarding assurance of whether the expected performance was found or not. Getting tested by the user needs to document the expected product behavior under specified conditions.

5.2.3 Simulate the requirements

To stimulate requirements, trading tools are available that we have used to simulate a proposed system in place or to add details of written requirements. The simulation takes prototyping to the next level.

6. Use Case Diagram

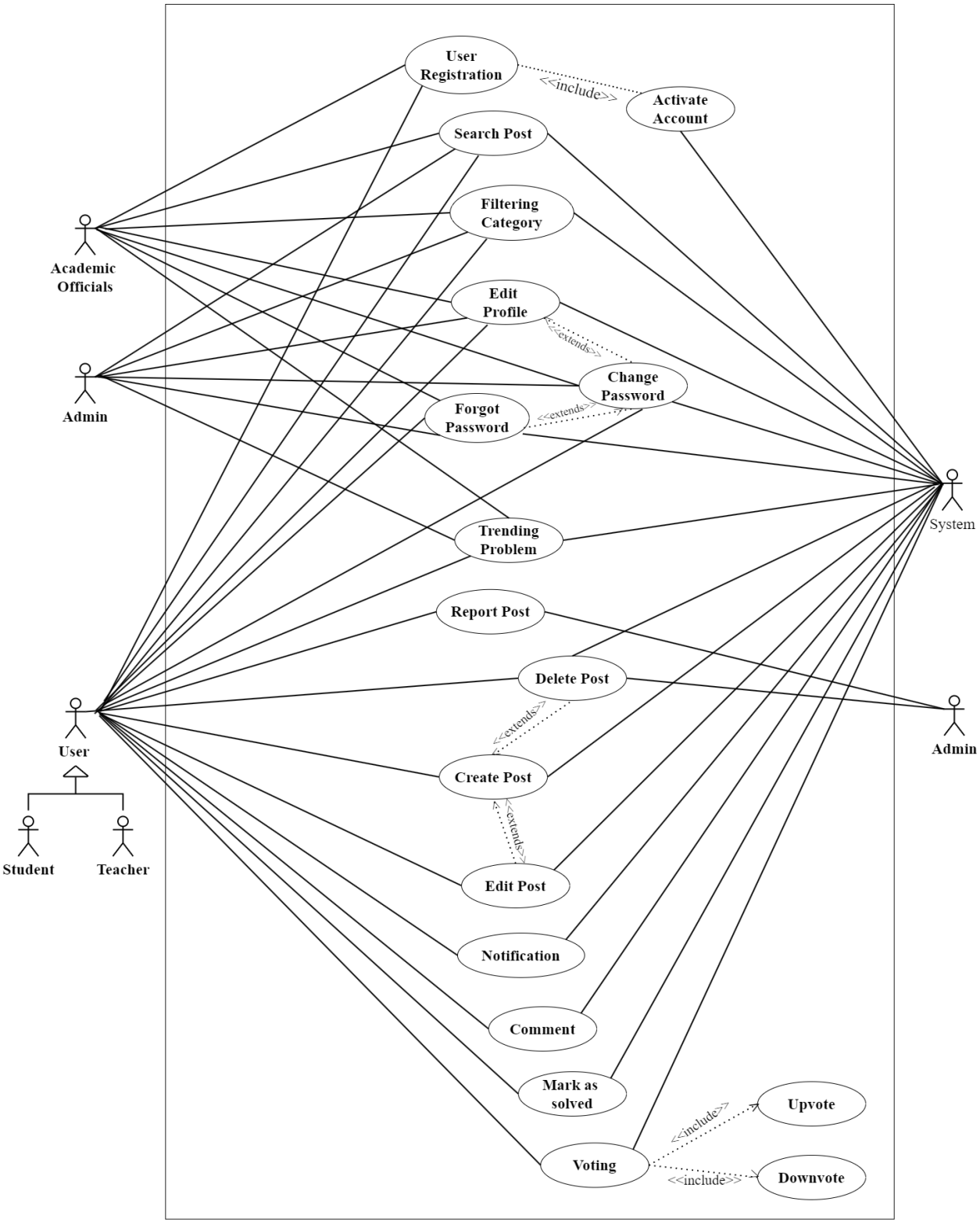

Figure 1: Use Case Diagram

7. Use Case Description

Table 1: Registration

Use Case	Users Registration	
Goal	User wants to create an account in the “Project Ovijog” System.	
Preconditions	N/A	
Success End Condition	A user account is created.	
Failed End Condition	User account is not created.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	“Create Account” button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User opens the “Project Ovijog” application.
	2	User clicks the “Create Account” button.
	3	User provides first name, last name, student ID, educational email, password, confirm password.
	4	Server checks Gmail is already existed or not.
	5	User activated the account.
	6	Account is created.
Alternative Flows	**Step**	**Branching Action**
	2a	User have an account
	3a	User does not provide information.
	4a	Server shows that information invalid or used before.
	4a1	User needs to change provided information.
Quality Requirements	**Step**	**Requirement**
	4	Server will respond within 3 to 5 seconds

Table 2: Activation Account

Use Case	Activate Account	
Goal	User activate account.	
Preconditions	Provide Gmail.	
Success End Condition	Successfully create account.	
Failed End Condition	Unable to create account.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	User needs to click on the provided link which was sent by the system in specified Gmail account.	
Main Success Flows	**Step**	**Action**
	1	User provides Gmail and password and clicked signup
	2	System sends an activation link to the given Gmail.
	3	User click on that link to activate account.
	4	Account activated.
Alternative Flows	**Step**	**Branching Action**
	2a	System doesn't send activation link
	2a1	User again provide information and signup
	3a	User doesn't click on activation link
	4a	Account is not activated.
Quality Requirements	**Step**	**Requirement**
	2	Server must send an activation link to the given Gmail within 5 minutes.

Table 3: Create Post

Use Case	Create Post	
Goal	Create a new post	
Preconditions	User Account	
Success End Condition	Successfully post about a problem	
Failed End Condition	Unable to post.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Create new post" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User clicked on "Create new post" button
	2	User select a category.
	3	User write the post in the text box
	4	clicked on "Post" button
	5	Post done.
Alternative Flows	**Step**	**Branching Action**
	2a	User try to post without selecting a category
	2a1	A popup will alert the user to select a category
	3a	User try to post without writing anything.
	3a1	A popup will alert the user to write something.
Quality Requirements	**Step**	**Requirement**
		N/A

Table 4: Edit Post

Use Case	Edit Post	
Goal	Edit previous posts	
Preconditions	There is already a Created post.	
Success End Condition	Successfully edit the post.	
Failed End Condition	Unable to edit.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Edit" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User select the post. Click on "Edit" button.
	2	Edit the post and clicked "Post" button.
	3	Post updated.
Alternative Flows	**Step**	**Branching Action**
	2a	User try to submit without writing anything.
	2a1	A popup will alert the user to write anything.
	3a	Post isn't updated.
Quality Requirements	**Step**	**Requirement**
		N/A

Table 5: Delete Post

Use Case	Delete Post	
Goal	Permanently delete previous post	
Preconditions	There is already a Created post.	
Success End Condition	Successfully delete the post.	
Failed End Condition	Unable to delete.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Delete" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User select the post. Click on "Delete" button.
	2	System ask confirmation.
	3	System deletes the post permanently.
Alternative Flows	**Step**	**Branching Action**
	2a	Post isn't deleted.
Quality Requirements	**Step**	**Requirement**
		Post will be deleted from the database and will not show in the future (Integrity).

Table 6: Comment

Use Case	Comment	
Goal	Comment on own or others post.	
Preconditions	There is already a Created post.	
Success End Condition	Successfully comment on posts	
Failed End Condition	Unable to comment	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"comment" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User goes to the post. Type comment and click "Comment" button.
	2	Notification sent to the user who post.
	3	User go to his post, comments on others comment.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

Table 7: Notification

Use Case	Notification	
Goal	Send notification.	
Preconditions	N/A	
Success End Condition	Successfully sent notification.	
Failed End Condition	Failed to send notification.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Signup" or "change password" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	There is a new activity in the existing post.
	2	Server notify the user who created the post.
	3	User notified.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
	2	Server will notify within 3 to 5 seconds

Table 8: Mark as solved

Use Case	Mark as solved	
Goal	Mark the problem as solved	
Preconditions	There is already a Created post.	
Success End Condition	Marked.	
Failed End Condition	Not marked.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Solved" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User feels that the problem that mentioned in the post is solved
	2	User click the "Solved" Button.
	3	System asks his confirmation.
	4	User confirmed.
	5	System removes the post from the Trending board
	6	The number of solving problem is increased.
Alternative Flows	**Step**	**Branching Action**
	4a	User not confirmed.
	4a1	User redirected to the previous page.
Quality Requirements	**Step**	**Requirement**
	5	System will response in a consistent way.

Table 9: Report

Use Case	Report	
Goal	Report against post.	
Preconditions	There is already a Created post.	
Success End Condition	Successfully reported against a user.	
Failed End Condition	Unable to report.	
Primary Actors: **Secondary Actors:**	User Admin	
Trigger	"Report" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users report to admin against a fake post
	2	Admin check the post
	3	Admin delete the post
Alternative Flows	**Step**	**Branching Action**
	3	Admin take action.
Quality Requirements	**Step**	**Requirement**
		N/A

Table 10: Voting

Use Case	Voting	
Goal	User vote on others post.	
Preconditions	N/A	
Success End Condition	Successfully vote on others post.	
Failed End Condition	Vote in not counted.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Vote" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User goes to a post. Give upvote on it if he/she liked it
	2	System increases the vote number.
Alternative Flows	**Step**	**Branching Action**
	1a	Give a downvote on the post
	2a	System decreases the vote number.
Quality Requirements	**Step**	**Requirement**
		N/A

Table 11: Search

Use Case	Search	
Goal	User wants to search a post	
Preconditions	N/A	
Success End Condition	User gets his/her desired post.	
Failed End Condition	The application server is unable to provide the desired post and shows “No Result”	
Primary Actors: **Secondary Actors:**	User System	
Trigger	The “Search box” needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	The user clicks the search box and provides desired keywords.
	2	Application server shows maximum possible results according to the keyword
	3	User looks for his/her desired post.
Alternative Flows	**Step**	**Branching Action**
	3a	Application server failed to show the desired result
	3a1	User provides another keyword
	3a2	User stops searching
Quality Requirements	**Step**	**Requirement**
	2	System shows the result quickly.

Table 12: Category Filtering

Use Case	Category Filtering	
Goal	User search a specific category to filter.	
Preconditions	N/A	
Success End Condition	User gets his/her desired categorical post.	
Failed End Condition	The application server is unable to provide the desired post and shows "No Result".	
Primary Actors: **Secondary Actors:**	User System	
Trigger	Select a category	
Main Success Flows	**Step**	**Action**
	1	User select a category then clicked search button.
	2	Application server shows maximum possible results according to the selected category.
	3	User looks for his/her desired post.
Alternative Flows	**Step**	**Branching Action**
	2a	Application server failed to show the desired result.
	3a	User select another category.
Quality Requirements	**Step**	**Requirement**
	2	System shows the result quickly.

Table 13: Forgot Password

Use Case	Forgot Password	
Goal	Create new password.	
Preconditions	N/A	
Success End Condition	Successfully create new password.	
Failed End Condition	Unable to create new password.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Forgot Password" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User click on "Forgot Password"
	2	System asks for user Gmail. User provide his/her Gmail
	3	System sends a link to that Gmail.
	4	User clicks on that link.
	5	System asks for new password and confirm password. User provide new password.
	6	Password changed.
Alternative Flows	**Step**	**Branching Action**
	3a	System not sent the link
	3a1	Users again request for change password.
Quality Requirements	**Step**	**Requirement**
	3	Server must send an activation link to the given Gmail within 5 minutes.

Table 14: Change Password

Use Case	Change password	
Goal	Change user password	
Preconditions	Go to the edit profile	
Success End Condition	Change user's current password	
Failed End Condition	Unable to change password	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Change password" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User click on "Change Password"
	2	System asks current password and new password
	3	User provide current and new password.
	4	User click the "Save" Button.
	5	System changes the password.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

Table 15: Edit Profile

Use Case	Edit Profile	
Goal	Update information.	
Preconditions	N/A	
Success End Condition	Successfully update information.	
Failed End Condition	Unable to update information.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Edit Profile" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User click the "Edit Profile".
	2	User click the "Change username".
	3	System asks new username. User provide new username.
	4	User click the "Save" Button
	5	System updates the username.
Alternative Flows	**Step**	**Branching Action**
	2	User click on change password
	3	System asks current password and new password
	3a	User provide current and new password.
Quality Requirements	**Step**	**Requirement**
		N/A

Table 16: Trending Problem

Use Case	Trending Problem	
Goal	User find the trending problem.	
Preconditions	Go to the dashboard.	
Success End Condition	Show the most voted posts in the trending board	
Failed End Condition	Unable to show.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	N/A	
Main Success Flows	**Step**	**Action**
	1	User login to his account.
	2	System shows the most upvoted post.
	3	Trending post marked as solved.
	4	System updates the board.
Alternative Flows	**Step**	**Branching Action**
	3a	Non-trending post get most upvote.
Quality Requirements	**Step**	**Requirement**
	4	Continuously updated the trending board.

8. Activity Diagram

Usecase 1: User Registration

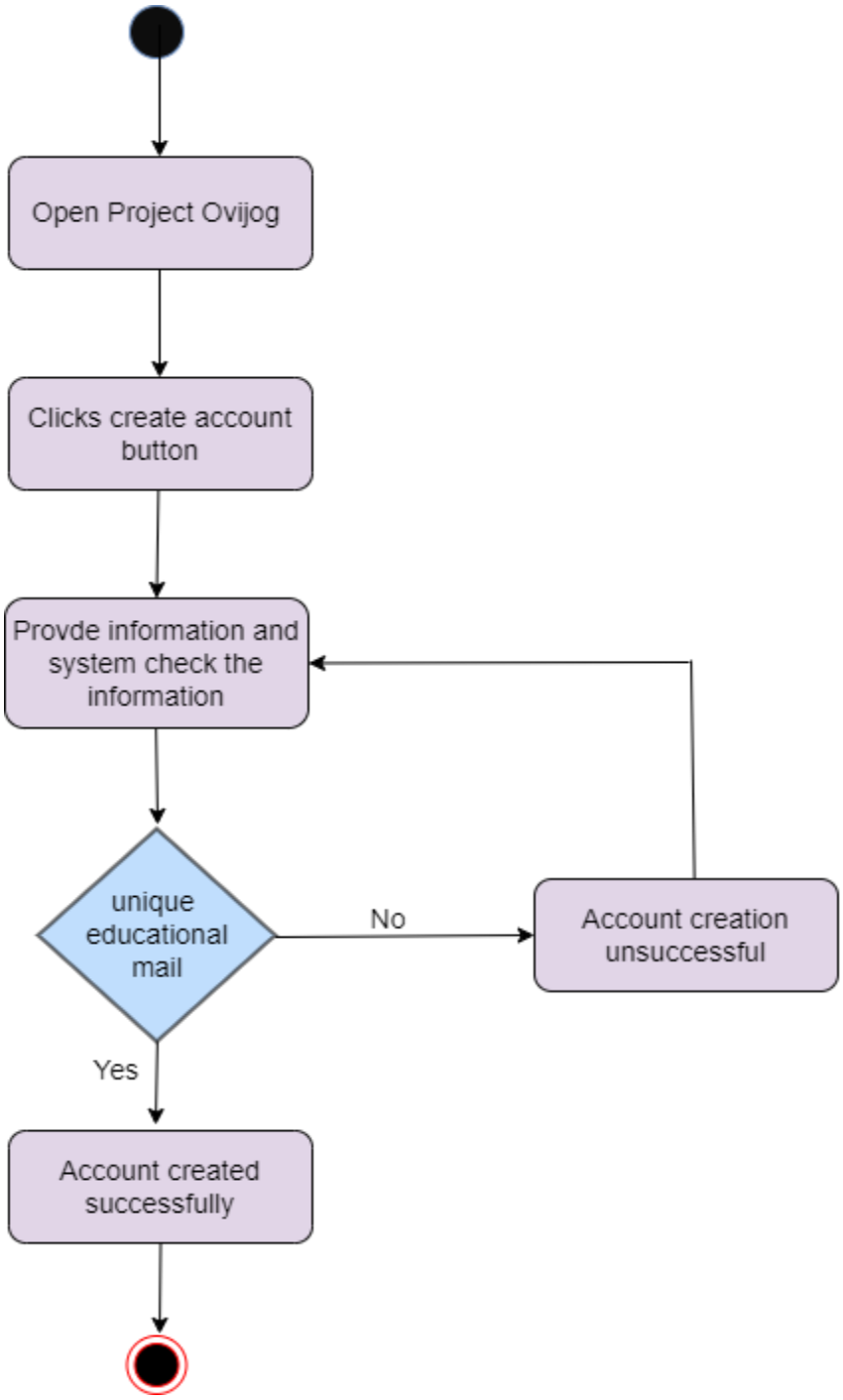

Figure 2: Users Registration

Usecase 2: Activate Account

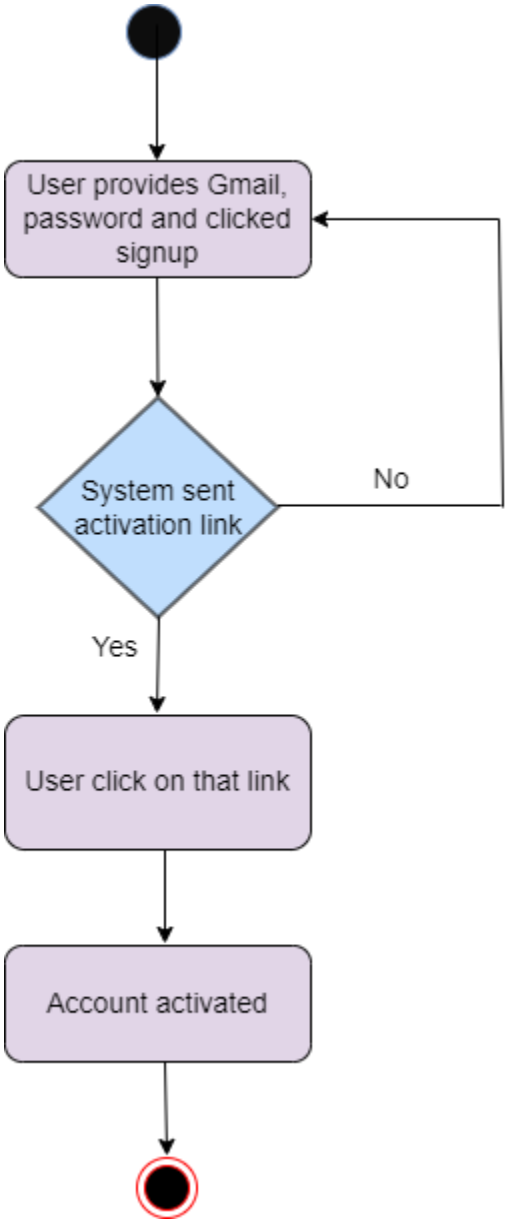

Figure 3: Activate Account

Usecase 3: Create Post

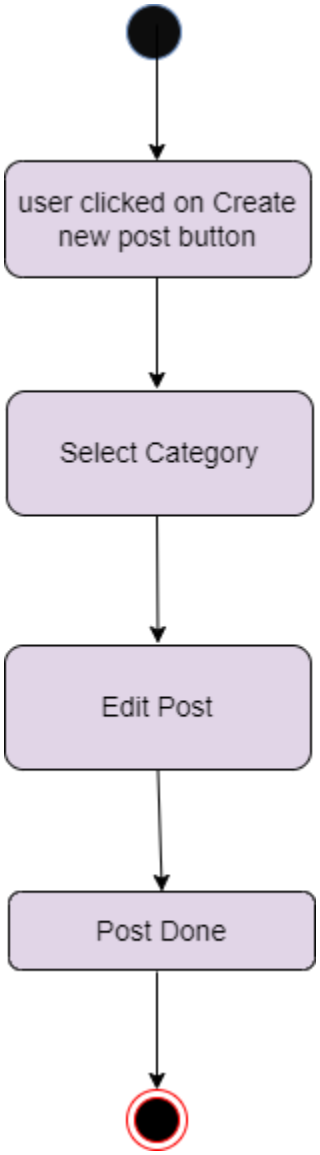

Figure 4: Create Post

Usecase 4: Edit Post

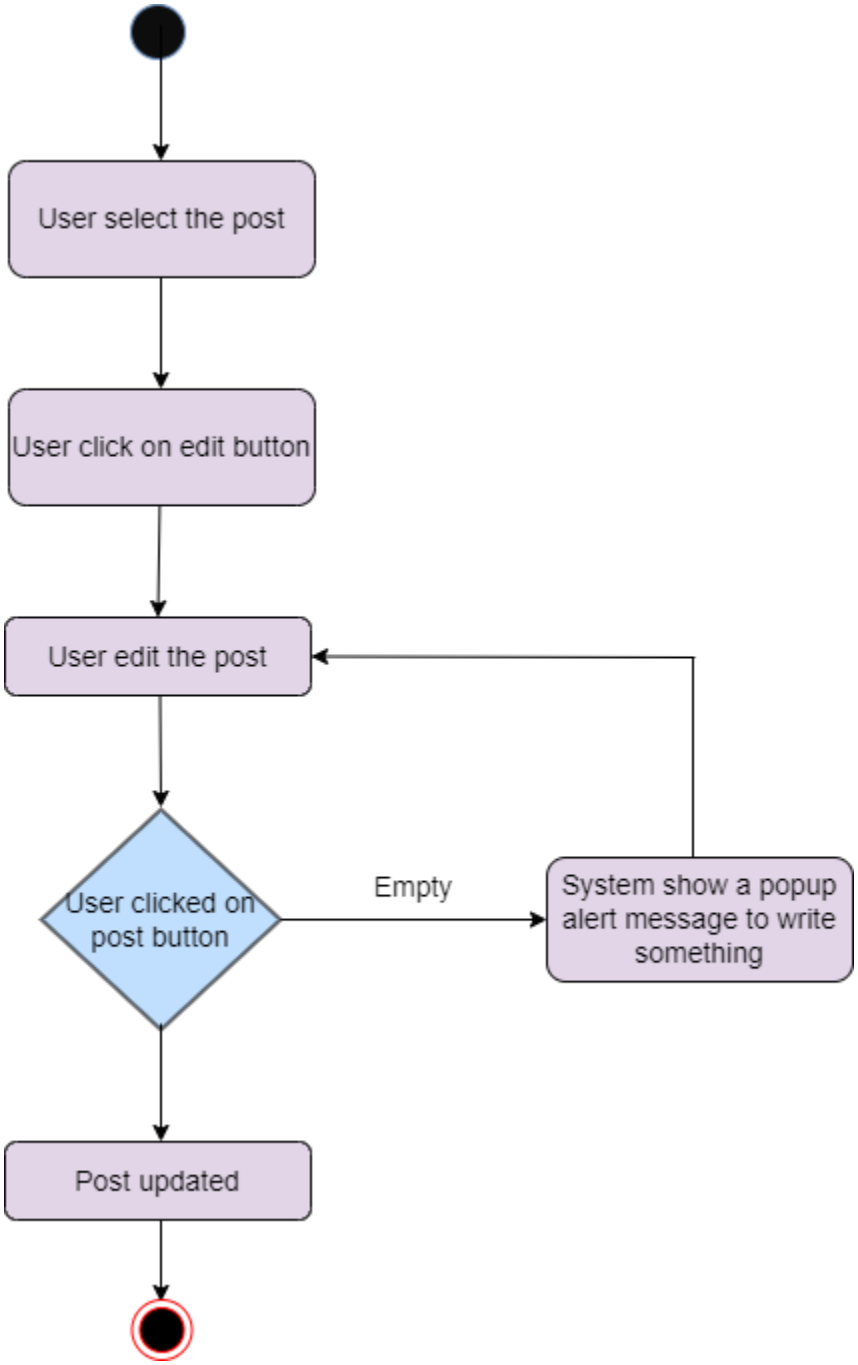

Figure 5: Edit Post

Usecase 5: Delete Post

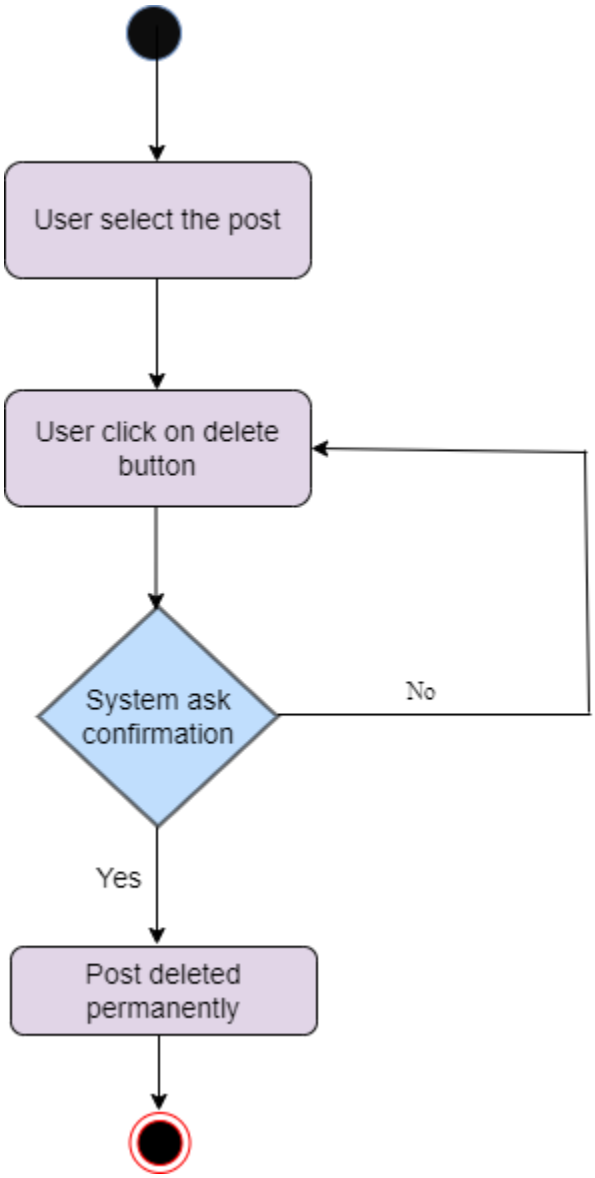

Figure 6: Delete Post

Usecase 6: Comment

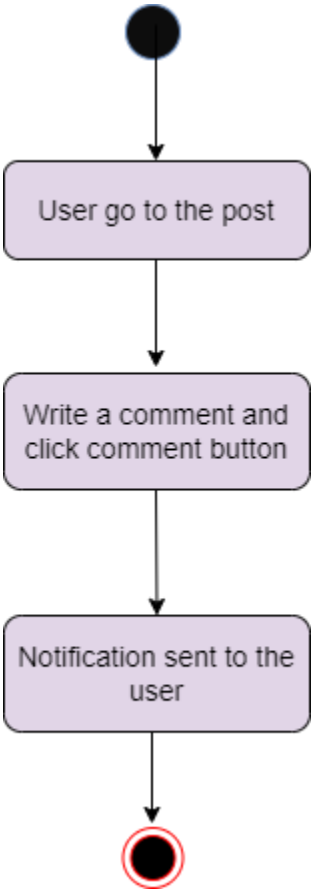

Figure 7: Comment

Usecase 7: Notification

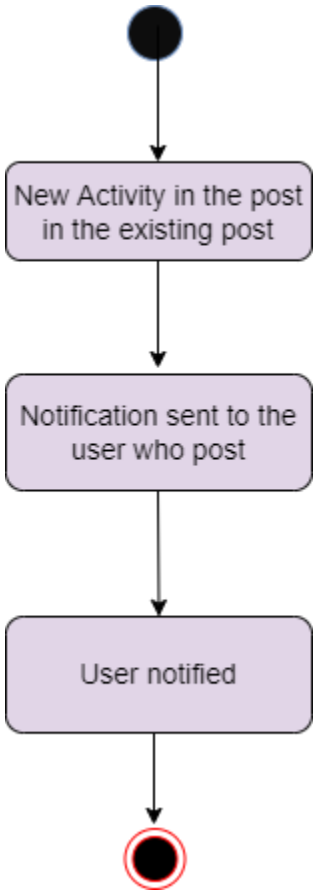

Figure 8: Notification

Usecase 8: Mark as Solved

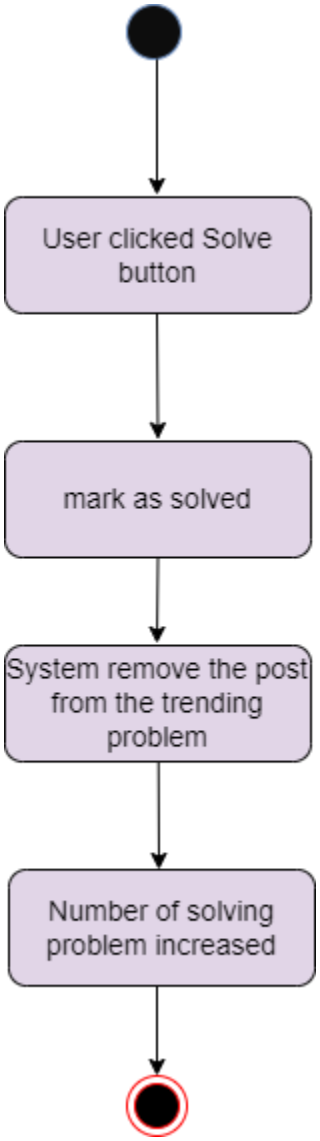

Figure 9: Mark as solved

Usecase 9: Report

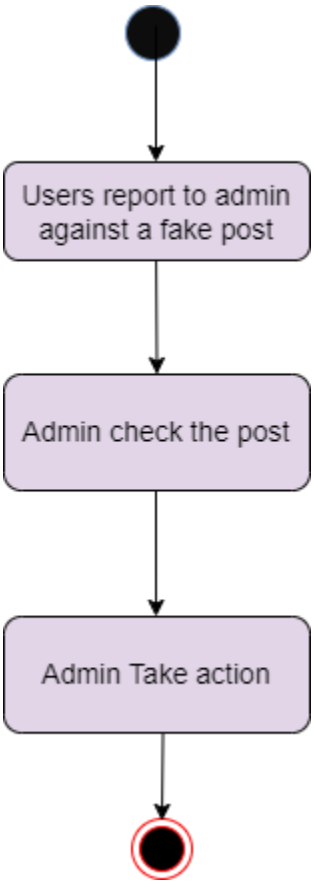

Figure 10: Report

Usecase 10: Voting

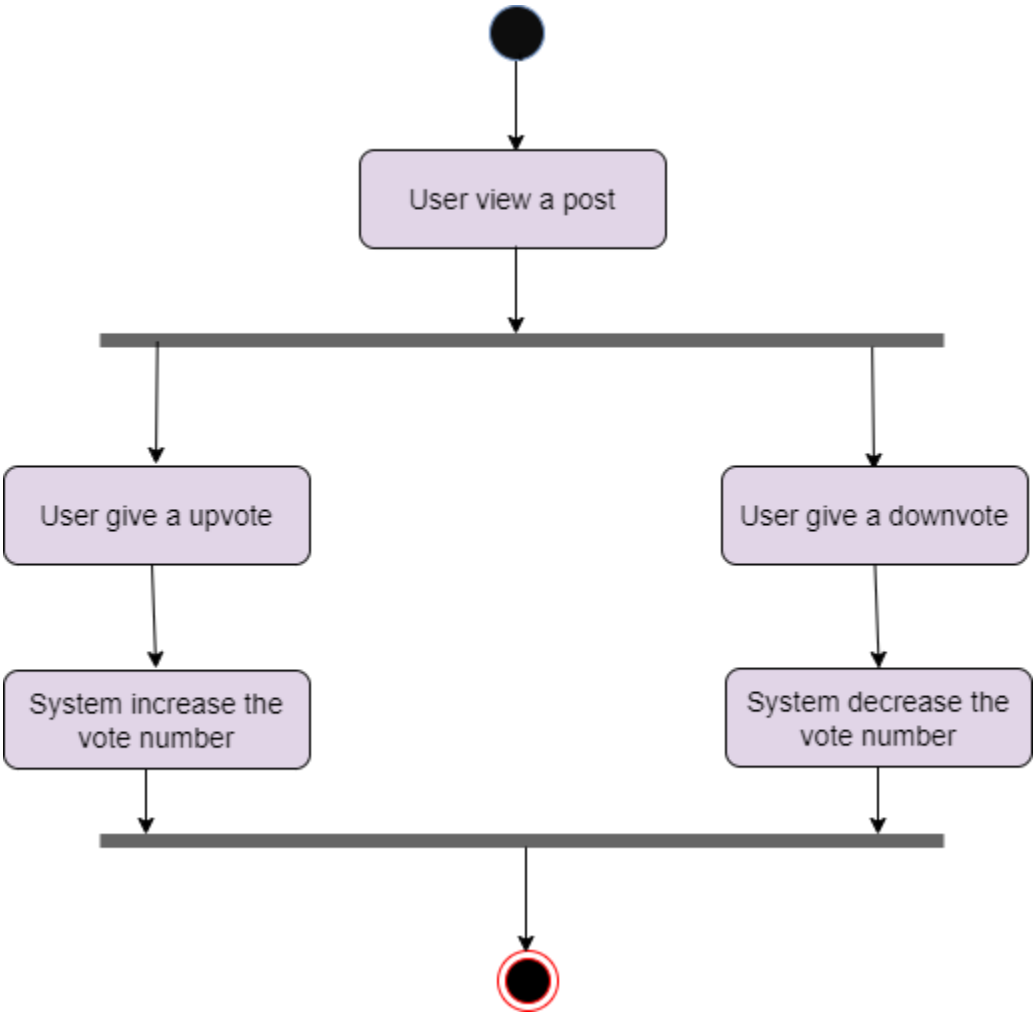

Figure 11: Voting

Usecase 11: Search

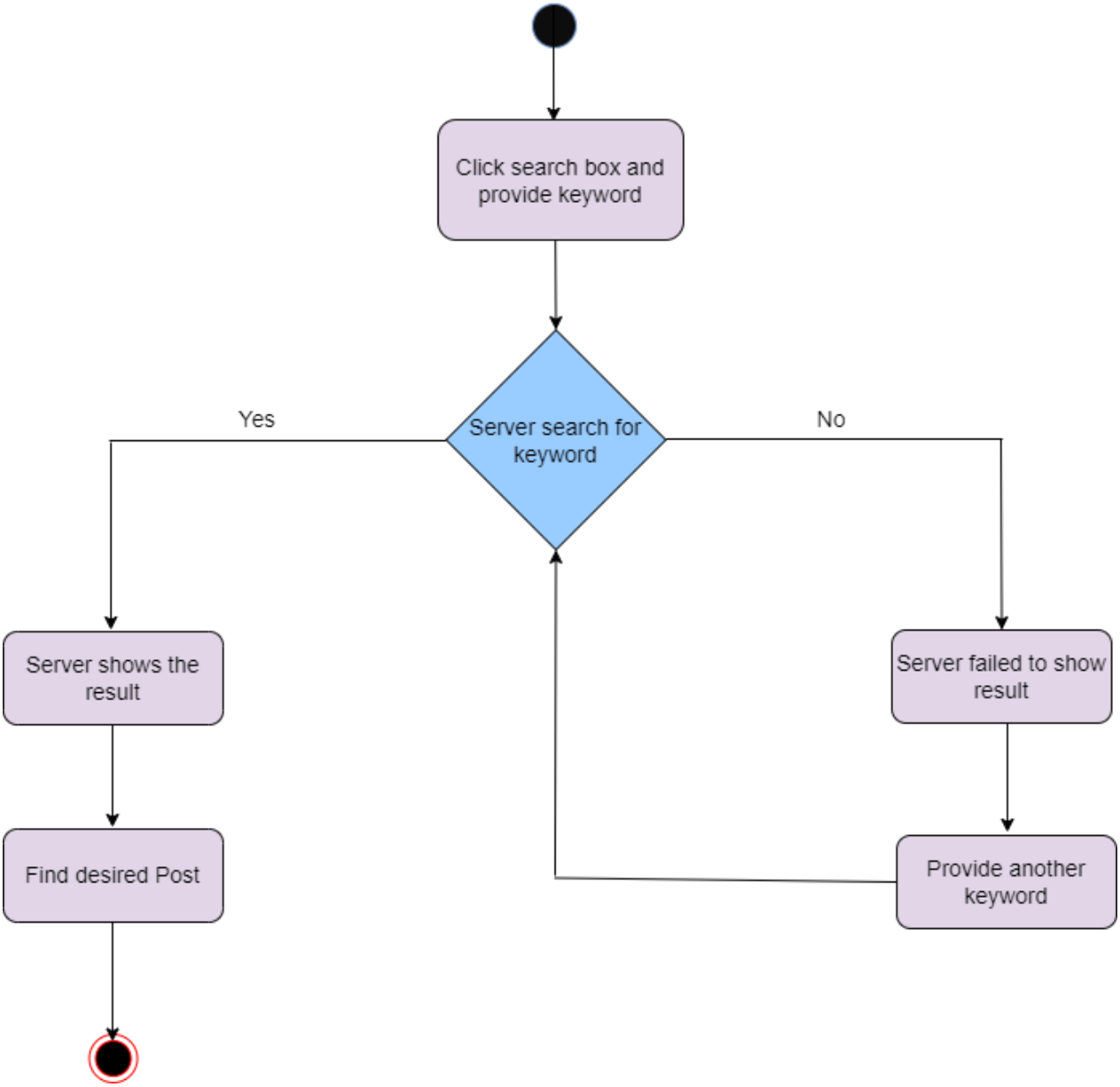

Figure 12: Search

Usecase 12: Filtering Category

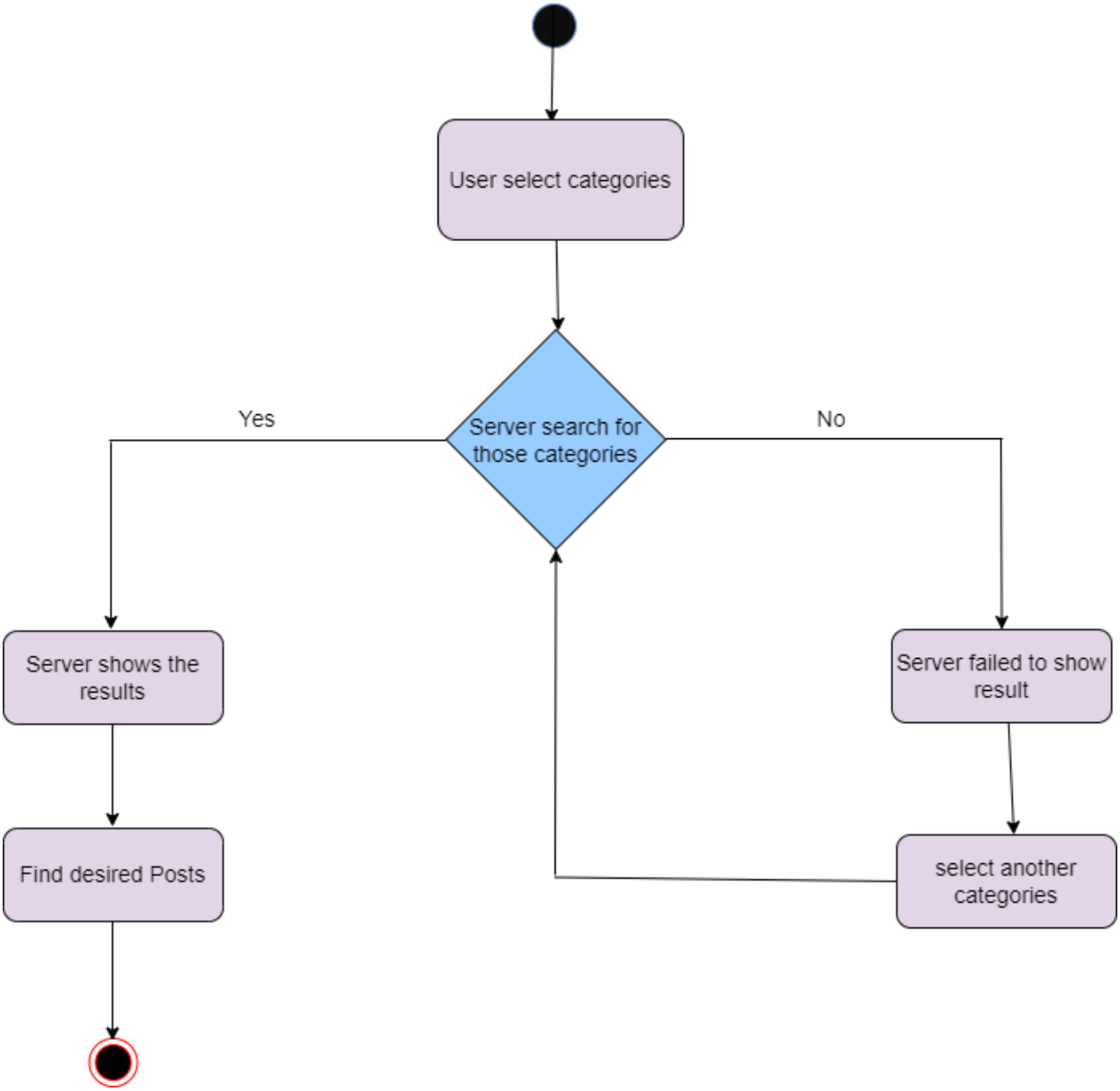

Figure 13: Filtering Category

Usecase 13: Forgot Password

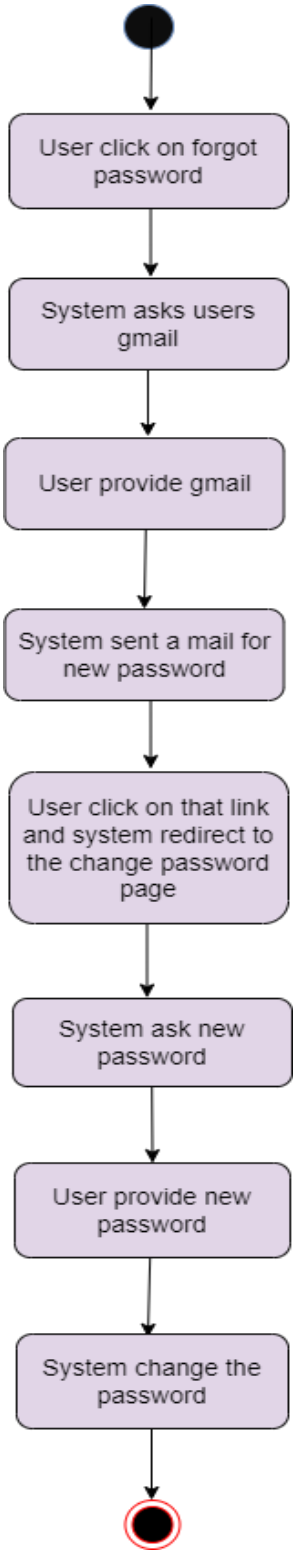

Figure 14: Forgot Password

Usecase 14: Change Password

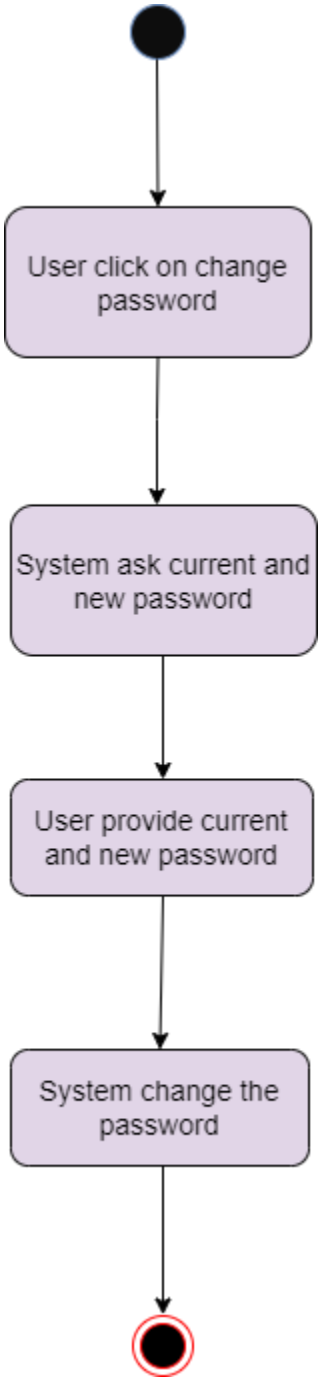

Figure 15: Change password

Usecase 15: Edit Profile

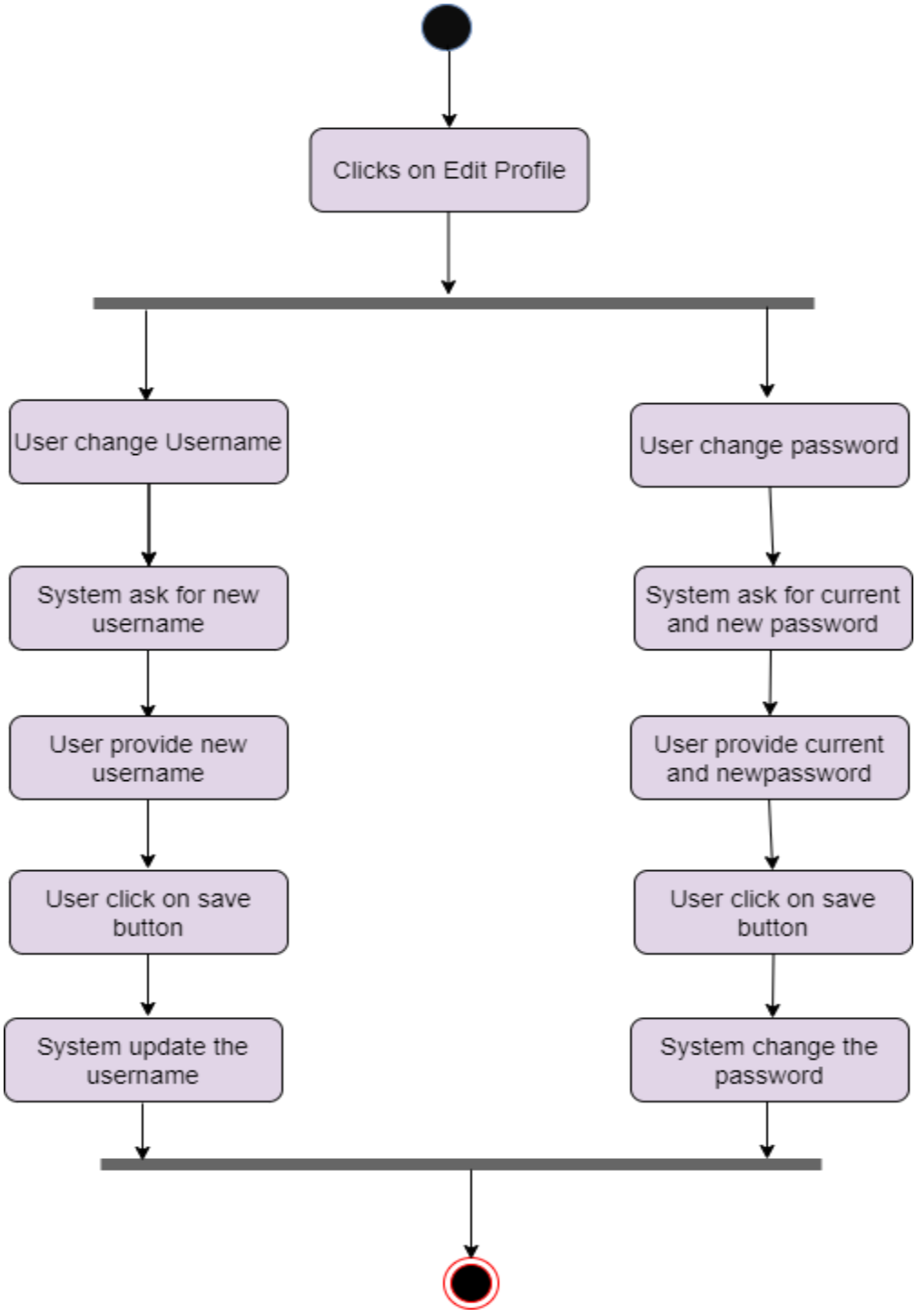

Figure 16: Edit Profile

Usecase 16: Trending Problem

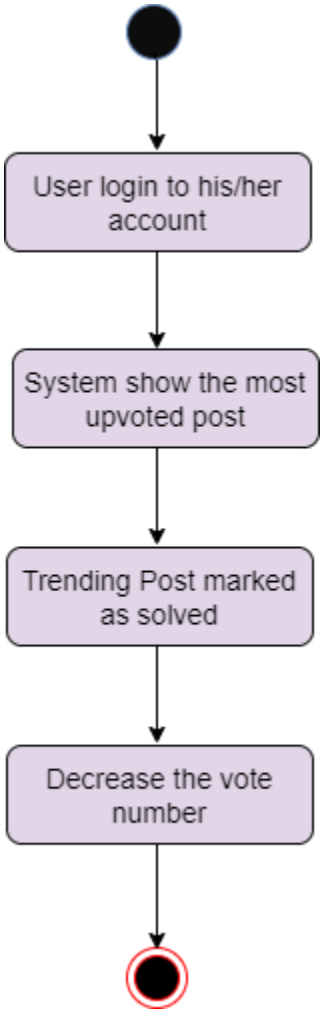

Figure 17: Trending Problem

9. Appendix

9.1 Prioritization of requirements

We've prioritized the functional requirements by following Three-level Scale technique.

9.1.1 Three-level Scale

When a Business Analyst categorizes the requirements in any of the ordering or ranking scale, it is subject to the analyst's understanding of the business. Many analysts suggest that this method has some drawbacks and advocate methods that have more than one scale.

9.1.2 Prioritization of the requirements of project Ovijog

FR1 – High priority: Its essential requirement for our system. User registration is obvious to run this application.

FR2: High priority: It is important to post if there is a problem. Otherwise, it will not come on the trending board. And not everyone will know.

FR3: Low priority: Editing post is not such necessary. However, if there is any type mistake, it is better to edit it for us to edit it for everyone's understanding.

FR4: Low priority: Deleting post is not such important. It will be convenient for everyone to see the problem if the post is not deleted.

FR5: High priority: The trending board will show the most upvoted posts. This is a high-level requirement. Through this it is known that most users are facing problem.

FR6 - Medium priority: Users can search a specific post using a search box and the system will provide results for search results.

FR7 - Low priority: Users can comment on someone's post if they want. There is no problem if you don't.

FR8 - High priority: Voting is important. The post will appear on the trending board only through voting.

FR9: -Medium priority: Users can do category wise filtering. In this case, the system will show the posts of that category after filtering.

FR10 - Low priority: If anyone thinks the post is fake then they can report it to the admin.

FR11 - Medium priority: If the posted problem is solved then the user will mark that post as solved. It will be easy for everyone to understand that problem is solved.

FR12 - Medium priority: Admin can handle many things. If someone reported to the admin against a fake post, admin can take further action including deleting the post.

FR13 - Medium priority: The user will be able to log out of his account at the end of his need. Users will need to login again for later use.

DR1 - High priority: User passwords will be kept protected using hashing. It is very important to keep user password protected.

PR1 - High priority: File will load faster and play with less buffering, because of the compression the application uses while uploading files.

MR1 - High priority: Code must be developed so that it can be modified later and will be readable.