Software Requirement and Design Specifications

[Blogging Website]

Version: [xx.xx]

Course Code	*CS3004*	
Instructor	*Mrs Javeria Farooq*	
Project Team	19K-0271 Ayesha Saif 19K-0249 Haris Aqeel 19K-1326 M.Jahanzeb	
Submission Date	*20-Dec-2021*	

Table of Contents

1. Introduction

1.1. Purpose of Document

To explain the software and design analysis of a web application.

1.2. Intended Audience

The designers and developers to understand the software and design analysis of this application.

1.3 Definition of Terms, Acronyms and Abbreviations

There are none.

1.4 Document Convention

Font-style: Arial (Italic)
Font_size:10

2. Overall System Description

2.1. Project Background

To make the process of writing blogs easy, fast and efficient by categorizing blogs under various categories.

2.2. Project Scope

The project allows a user to register and then login to the website in order to read and write blogs. These blogs can appear altogether on the home page of the website or they can be classified into the categories provided there.

2.3. Not In Scope

The project does not contain an admin module due to which content maintenance cannot be carried out.

2.4. Project Objectives

*The objective of the project is to allow users to login to the website and be able to create blogs belonging to different categories, provided there. These blogs consist of the number of views they get and they can later be updated or deleted completely by **the author** as well. Individuals who have not yet registered on to the websites can visit and partly utilize it as well, i.e. they can only **read** the blogs.*

2.5. Stakeholders

- *Users*
- *Backend team*
- *Frontend team*
- *Database Manager*

2.6. Operating Environment

The operating system used for our project is Windows 10. The software Microsoft Visual Studio Code is used for frontend and backend connectivity. MySQL is used for the database. JavaScript Node Modules are used, React and Node Js for Frontend and Backend Respectively.

2.7. System Constraints

- ***Software constraints :** Browser*
- ***Hardware constraints :** Must have a working system*
- ***User constraints**: The website can be used by people belonging to any age, race or occupation however, the website itself is based on the English Language creating a limiting factor in its reach.*

2.8. Assumptions & Dependencies

We are making the following assumptions:

- *Our database server is functioning 24/7.*
- *Our backend API is functioning 24/7.*

3. External Interface Requirements

3.1. Hardware Interfaces

PC, laptop or a smartphone

3.2. Software Interfaces

For Front End: React JS, the libraries are: mysql@2.15.0, body-parser@1.18.2, Bootstrap, FontAwesome, Use state.

For Backend, express@4.16.3, Node JS Libraries are express, axios, Use state.

3.3. Communications Interfaces

The user will need a Web Browser and a valid email.

4. Functional Requirements

4.1. Functional Hierarchy

[This section will give a big picture of overall system functionality. The main modules/features of system and their sub-functions will be described here in the form of a functional hierarchy so that, before getting into the use case, audience could grab the idea of overall system functions.]

4.2. Use Cases

4.2.1. [Title of use case]

[Use Case Diagram]
[Use Case Description]

Use Case Description: *It allows a user to create a blog according to the desired category.*	
Use Case name: *Create Blog*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *-*	
Pre-conditions: ▪ *Must be logged in.*	
Flow of Events: ***1.*** *User logs in* ***2.*** *Choses category* ***3.*** *Writes and publishes blog*	
Alternative and exceptional flows: ***4.1*** *Cannot write if the user is not logged in*	

Post-conditions: ▪ -

Use Case Description: *It allows a user to update an already existing blog that* ***they*** *wrote.*	
Use Case name: *Update Blog*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *-*	
Pre-conditions: ▪ *Must be logged in.* ▪ *Must be the author.*	
Flow of Events: ***1.*** *User logs in* ***2.*** *Choses blog* ***3.*** *Updates and publishes.*	
Alternative and exceptional flows: ***4.1*** *Cannot update if the user is not logged in* ***4.2*** *Cannot update unless the user is the author.*	

Post-conditions: ▪

Use Case Description: *It allows a user to delete an already existing blog that* ***they*** *wrote.*	
Use Case name: *Delete Blog*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *-*	
Pre-conditions: ▪ *Must be logged in.* ▪ *Must be the author.*	
Flow of Events: ***1.*** *User logs in* ***2.*** *Choses blog* ***3.*** *Deletes.*	
Alternative and exceptional flows: ***4.1*** *Cannot deleted if the user is not logged in* ***4.2*** *Cannot delete unless the user is the author.*	
Post-conditions: ▪ *Deleted blog will not be visible on the website.*	

Use Case Description: *It allows a user to like a blog.*	
Use Case name: *Like a Blog*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *-*	
Pre-conditions: ▪ *Must be logged in.*	
Flow of Events: ***1.*** *User logs in* ***2.*** *Choses blog* ***3.*** *Likes*	
Alternative and exceptional flows: ***4.1*** *Cannot like if the user is not logged in*	
Post-conditions: ▪ *The blog's like count will be incremented.*	

Use Case Description: *It allows a user to rate a blog out of a 10.*	
Use Case name: *Rating a blog*	

Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *-*	
Pre-conditions: ▪ *Must be logged in.*	
Flow of Events: ***1.*** *User logs in* ***2.*** *Choses blog* ***3.*** *Rates*	
Alternative and exceptional flows: ***4.1*** *Cannot rate if the user is not logged in*	
Post-conditions: ▪ *The blog's rate count will be incremented.* ▪ *The blog's average rate count will be affected.*	

Use Case Description: *It allows a user to write a review on another author's blog.*	
Use Case name: *Review a Blog*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	

Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *-*
Pre-conditions: ▪ *Must be logged in.*
Flow of Events: ***1.*** *User logs in* ***2.*** *Choses blog* ***3.*** *Writes a review*
Alternative and exceptional flows: ***4.1*** *Cannot review if the user is not logged in*
Post-conditions: ▪ *The blog's reviews will be visible to all the other users.*

Use Case Description: *It allows a user to write a blog according to one of the categories provided.*	
Use Case name: *Category of a Blog*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *Search*	

Pre-conditions: *-*
Flow of Events: ***1.*** *User logs in* ***2.*** *Search by category when reading* ***3.*** *Selects a category from the drop down when writing*
Alternative and exceptional flows: *-*
Post-conditions: ▪ *All blogs belonging to the chosen category will be displayed.*

Use Case Description: *It allows a user to view a blog.*	
Use Case name: *View a Blog*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *Search*	
Pre-conditions: *-*	

Flow of Events:

1. *Accesses website*

2. *Search by category when reading*

Alternative and exceptional flows: *-*

Post-conditions:

- *All blogs available will be displayed to the user.*

Use Case Description: *It allows a user to edit their profile details.*

Use Case name: *Update Profile*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	

Relationships ▪

Includes: *-*

- ***Extends:*** *-*

Pre-conditions: *Must be registered and logged in.*

Flow of Events:

1. *User logins*

2. *Accesses profile.*

3. *Clicks on "Update"*

4. *Makes changes*

5. *Clicks "OK"*
Alternative and exceptional flows: -
Post-conditions: ▪ *Changes have been made to the database.*

Use Case Description: *It allows a user to login to the website with privileges.*	
Use Case name: *Login*	
Primary actor: *User*	***Other actors:*** -
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *Email verification, Registration* ▪ ***Extends:*** *Logout*	
Pre-conditions: *Must be registered into the system and the email has been verified.*	

Flow of Events: ***1.*** *User enter the website's URL in the browser search bar.* ***2.*** *Clicks on "LOGIN"* ***3.*** *Enters login details*
Alternative and exceptional flows: *-*
Post-conditions: ▪ *User gets access to the website with additional features.*

Use Case Description: *It allows a user to register to the website and be able to login to it.*	
Use Case name: *Registration*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *Already existing* *email* ▪ ***Extends:*** *-*	
Pre-conditions: *-*	
Flow of Events: ***1.*** *User enter the website's URL in the browser search bar.* ***2.*** *Clicks on "REGISTRAION"* ***3.*** *Enters login details*	

4. *Logs into mentioned email for confirmation.*
Alternative and exceptional flows: *-*
Post-conditions: ▪ *User gets access to the website with additional features.*

Use Case Description: *It allows a user to logout of the website.*	
Use Case name: *Logout*	
Primary actor: *User*	***Other actors:*** *-*
Stakeholders: *User*	
Relationships ▪ ***Includes:*** *-* ▪ ***Extends:*** *-*	
Pre-conditions: *Must be logged into the system*	
Flow of Events: ***1.*** *User clicks on "LOGOUT"*	

Alternative and exceptional flows: -
Post-conditions: ▪ *User is taken to the home page with the additional restraint of only reading blogs.*

5. Non-functional Requirements

5.3. Security Requirements

For security purposes, the user attempting to register to the website must have a valid email. Once the user has provided a valid email, a verification email is sent to them and only after the user has provided confirmation via this email are they allowed access to the website.

SDS

6. System Architecture

The file architecture can be viewed on the following link:
https://github.com/Haris-sp4rk/Blogapp

6.2. Software Architecture

7. Design Strategy

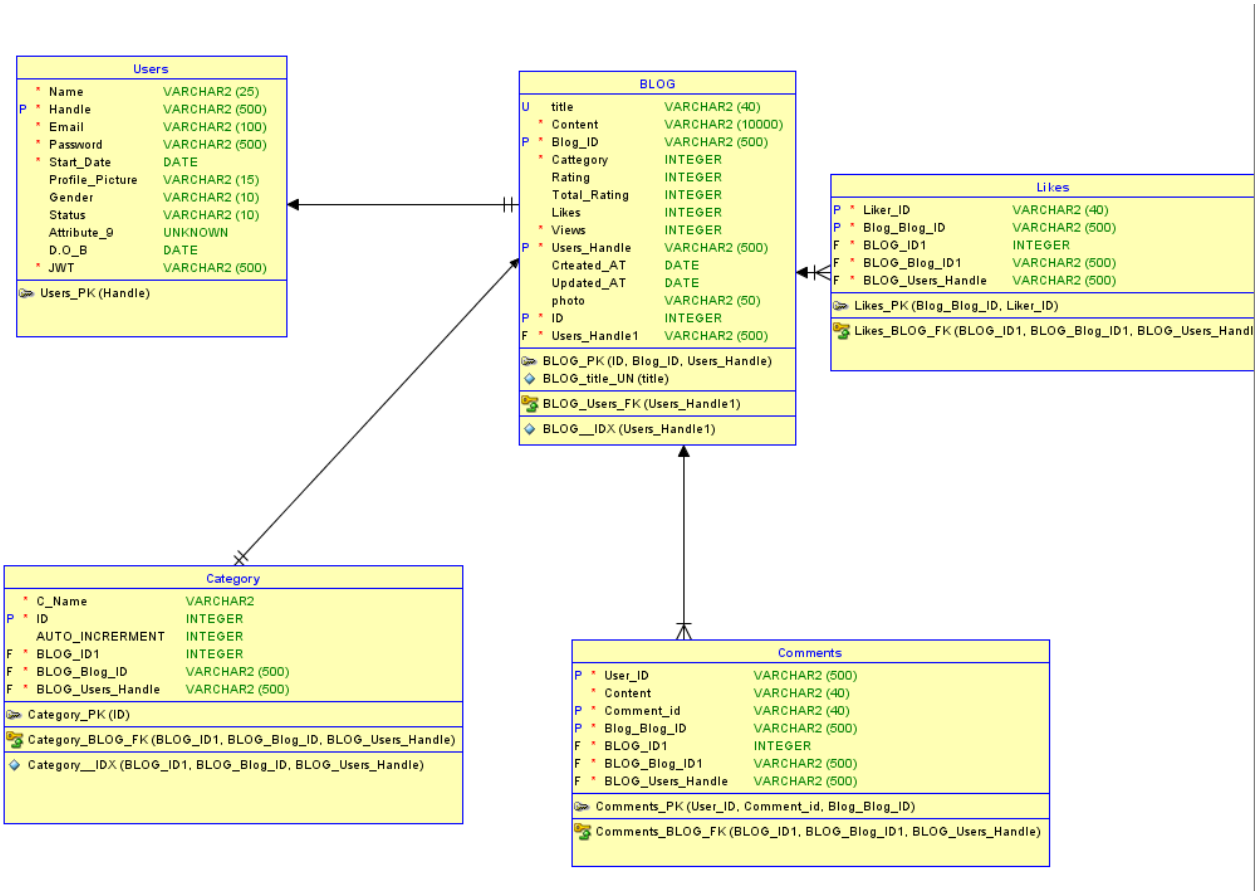
[Describe the design strategies or decisions that impact the overall organization of the system and its high-level structures. This information should provide the reader with insights into the key abstractions and mechanisms used in the system architecture.

For the strategy, discuss the reasoning employed (possibly referring to previously stated design goals and principles) and any trade-offs. Areas for consideration include:

- *Add admin panel to manage posts*
- *Adding AI models to predict user preferences*
- *Adding NoSQL database to manage big data*

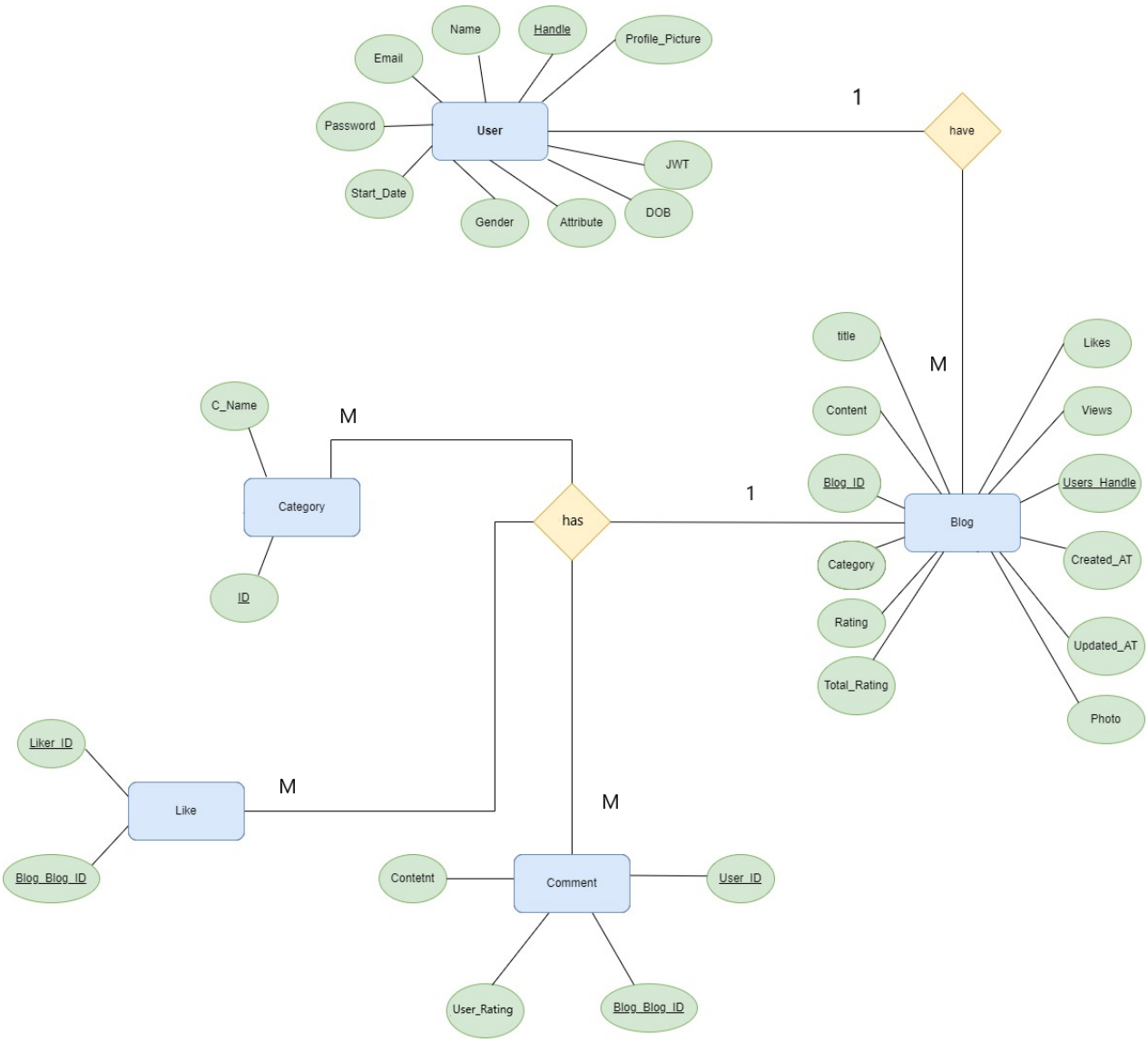
8. Detailed System Design

Class Diagram:



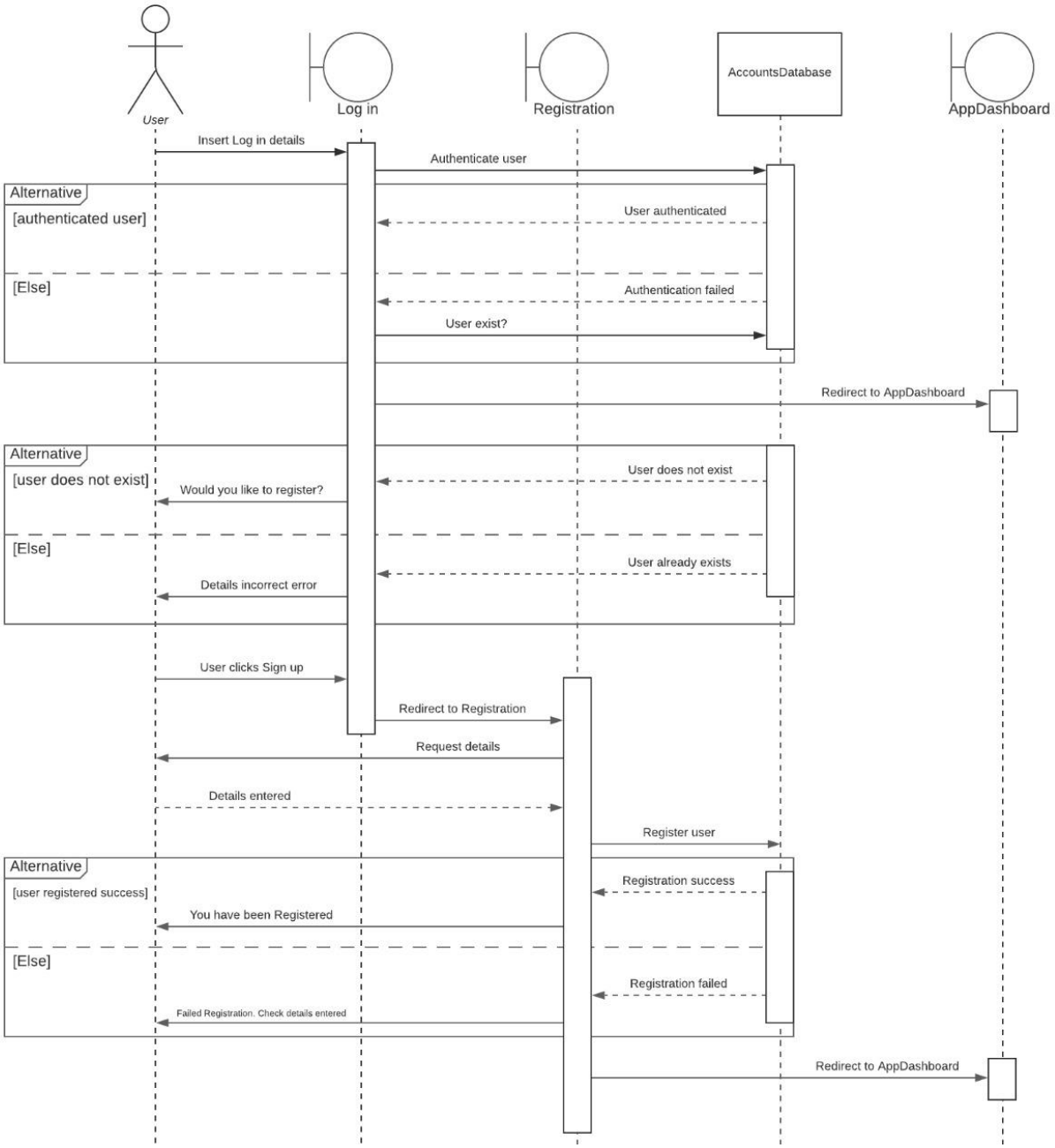
8.1. Database Design

8.1.1. ER Diagram



9. Application Design

9.1.2 Sequence Diagram 1


User
Log in
Registration
AccountsDatabase
AppDashboard
Insert Log in details
Authenticate user
Alternative
[authenticated user]
User authenticated
[Else]
Authentication failed
User exist?
Redirect to AppDashboard
Alternative
[user does not exist]
User does not exist
Would you like to register?
[Else]
User already exists
Details incorrect error
User clicks Sign up
Redirect to Registration
Request details
Details entered
Register user
Alternative
[user registered success]
Registration success
You have been Registered
[Else]
Registration failed
Failed Registration. Check details entered
Redirect to AppDashboard

Collaboration Diagram:

Diagram 1:

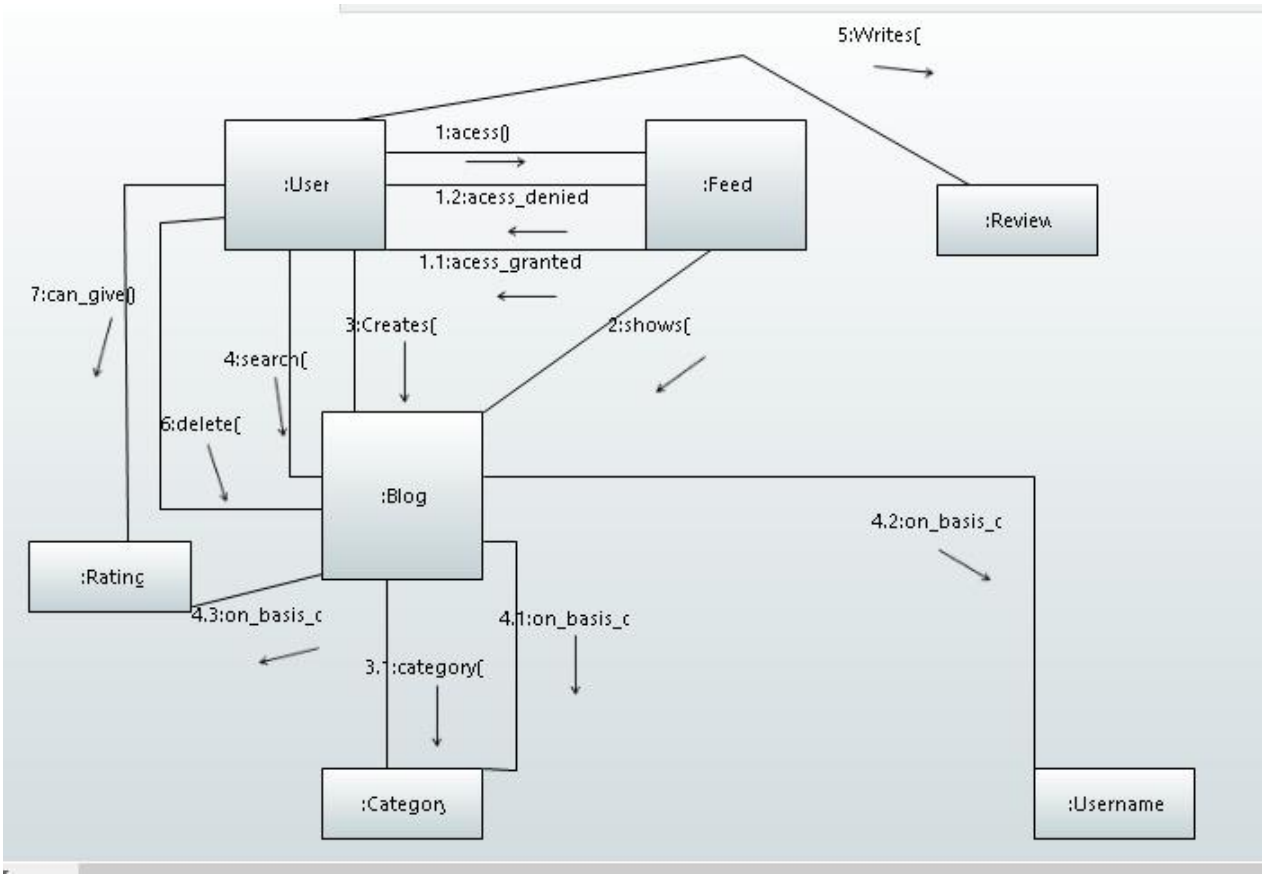
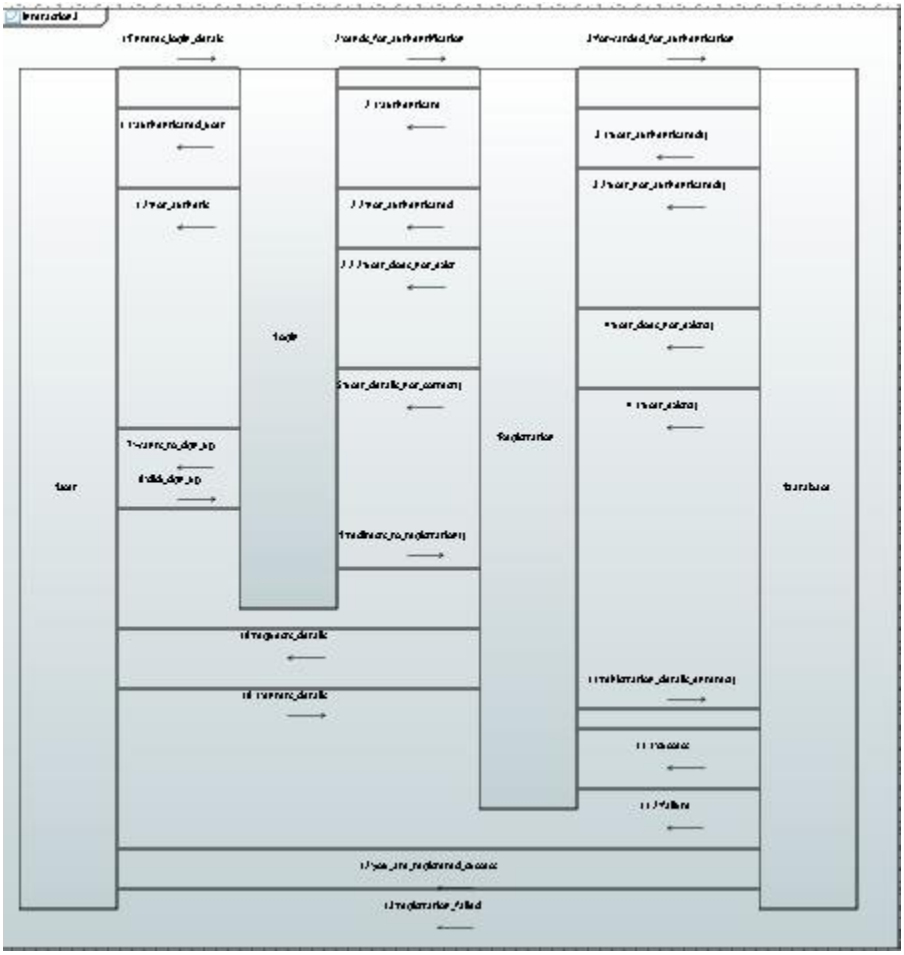
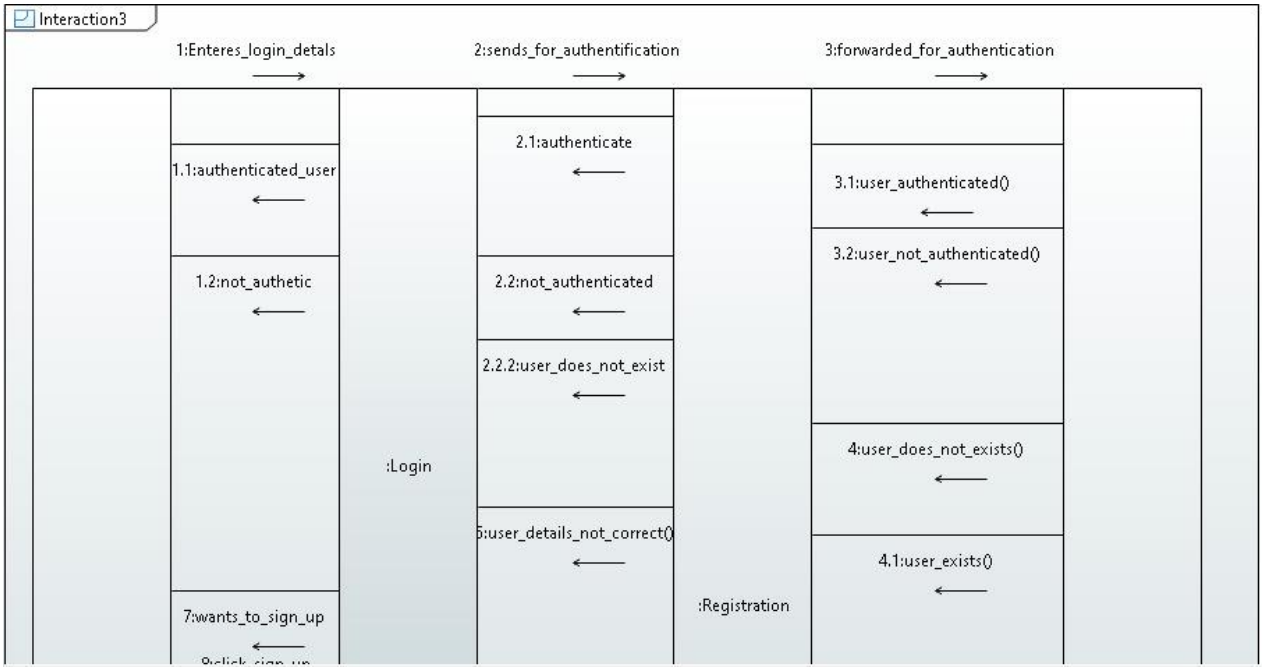
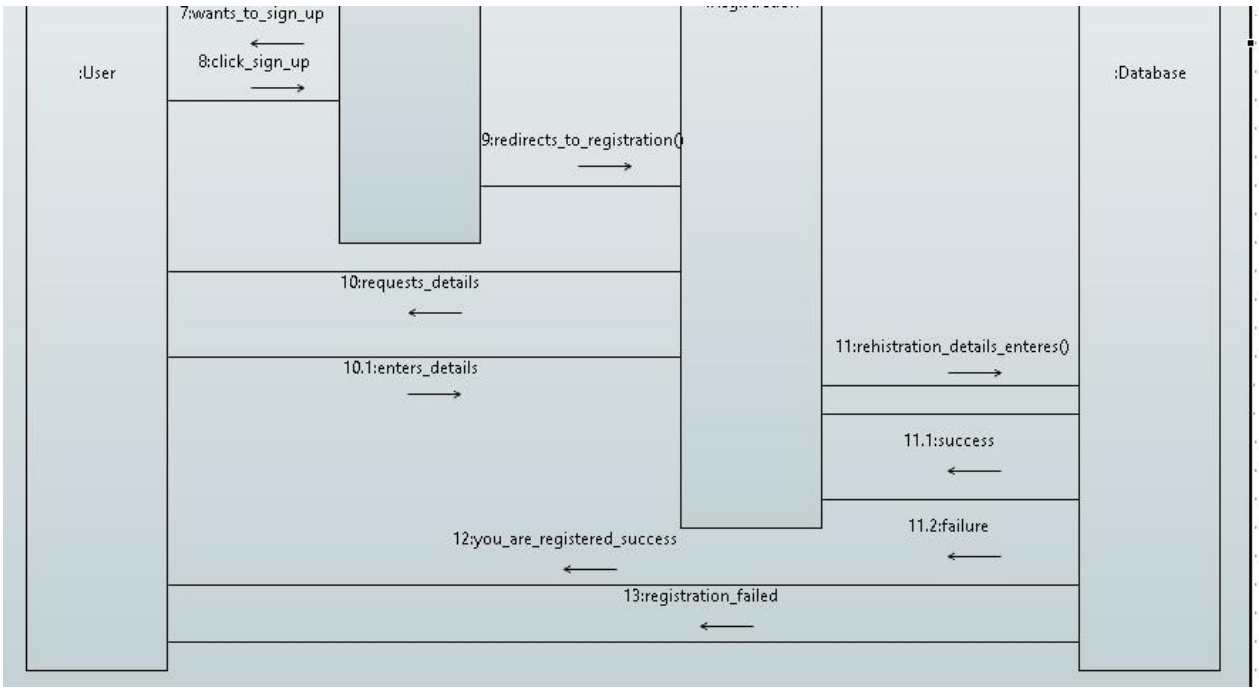


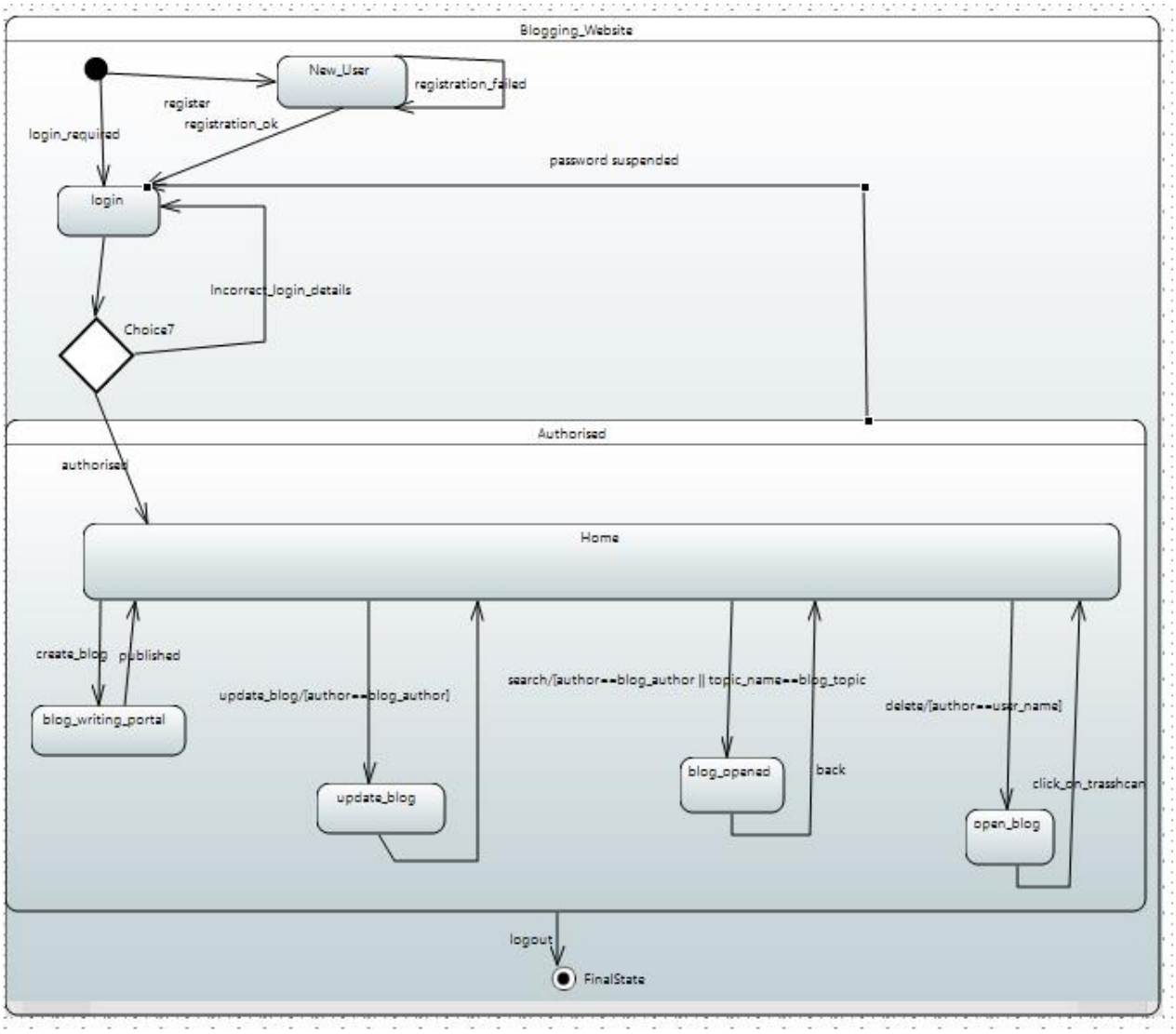
Diagram 2:



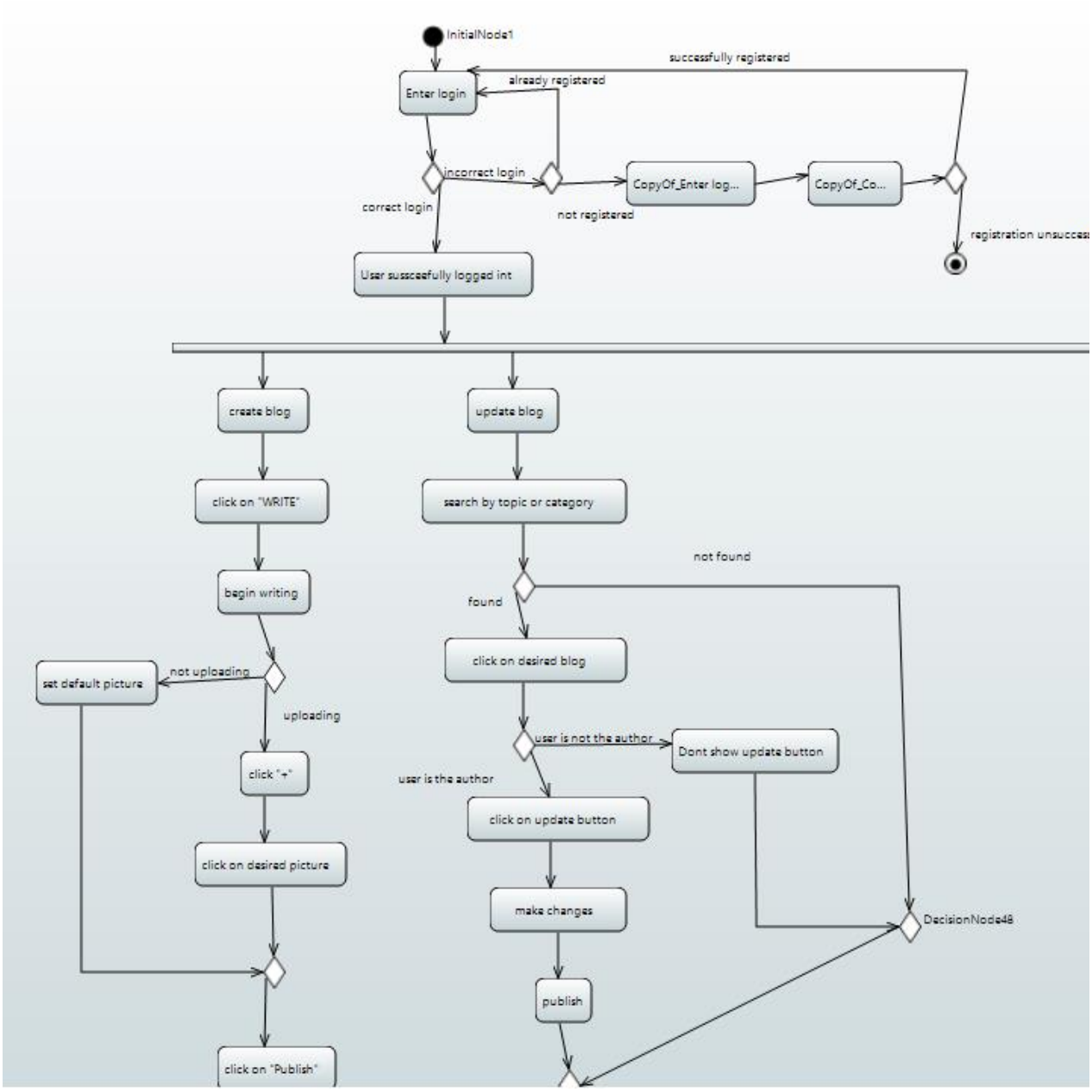

Interaction3
1:Enteres_login_detals
2:sends_for_authentification
3:forwarded_for_authentication
2.1:authenticate
1.1:authenticated_user
3.1:user_authenticated()
3.2:user_not_authenticated()
1.2:not_authetic
2.2:not_authenticated
2.2.2:user_does_not_exist
:Login
4:user_does_not_exists()
6:user_details_not_correct()
4.1:user_exists()
:Registration
7:wants_to_sign_up

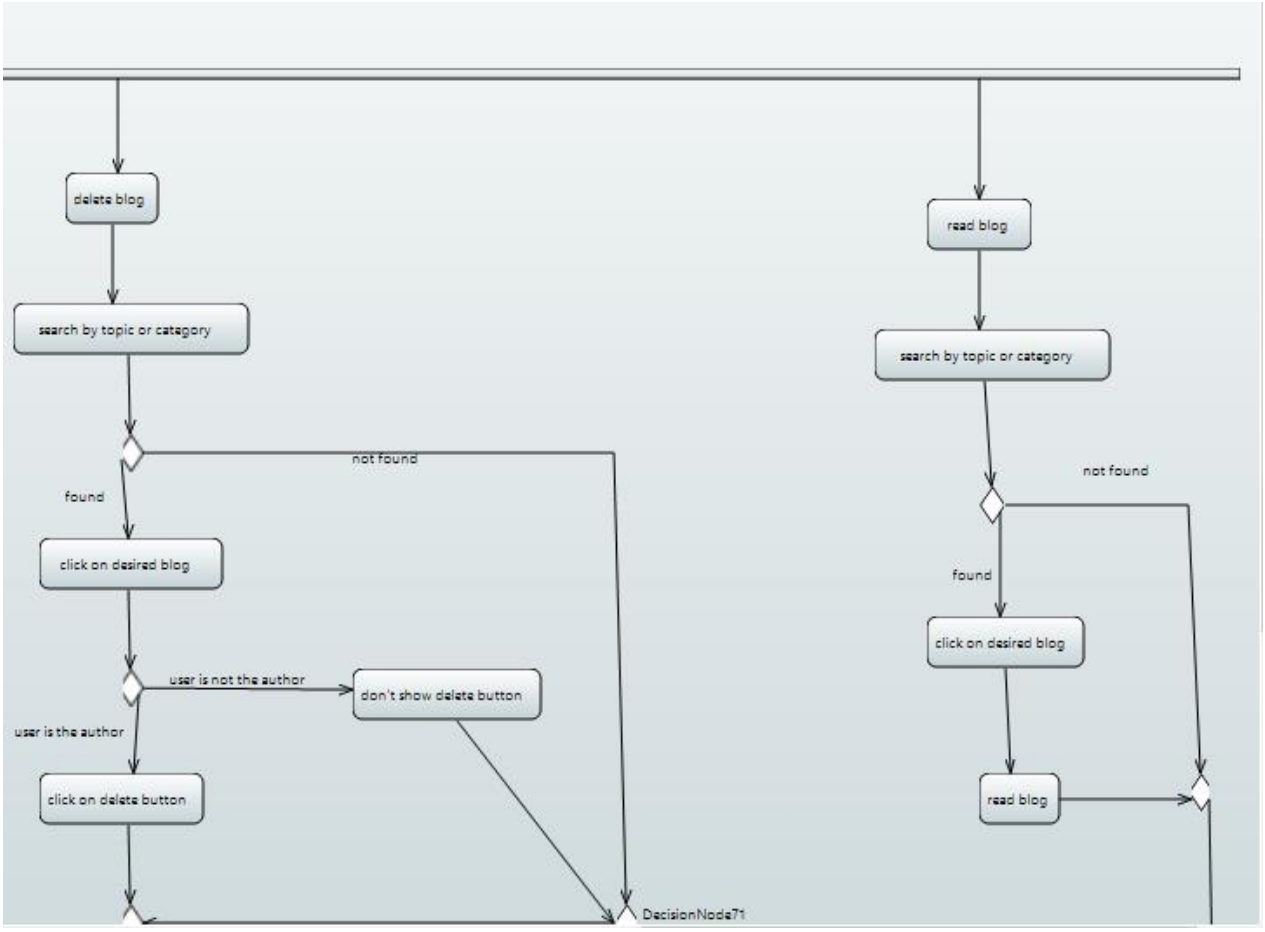

7:wants_to_sign_up
:User
8:click_sign_up
:Database
9:redirects_to_registration()
10:requests_details
11:rehistration_details_enteres()
10.1:enters_details
11.1:success
11.2:failure
12:you_are_registered_success
13:registration_failed

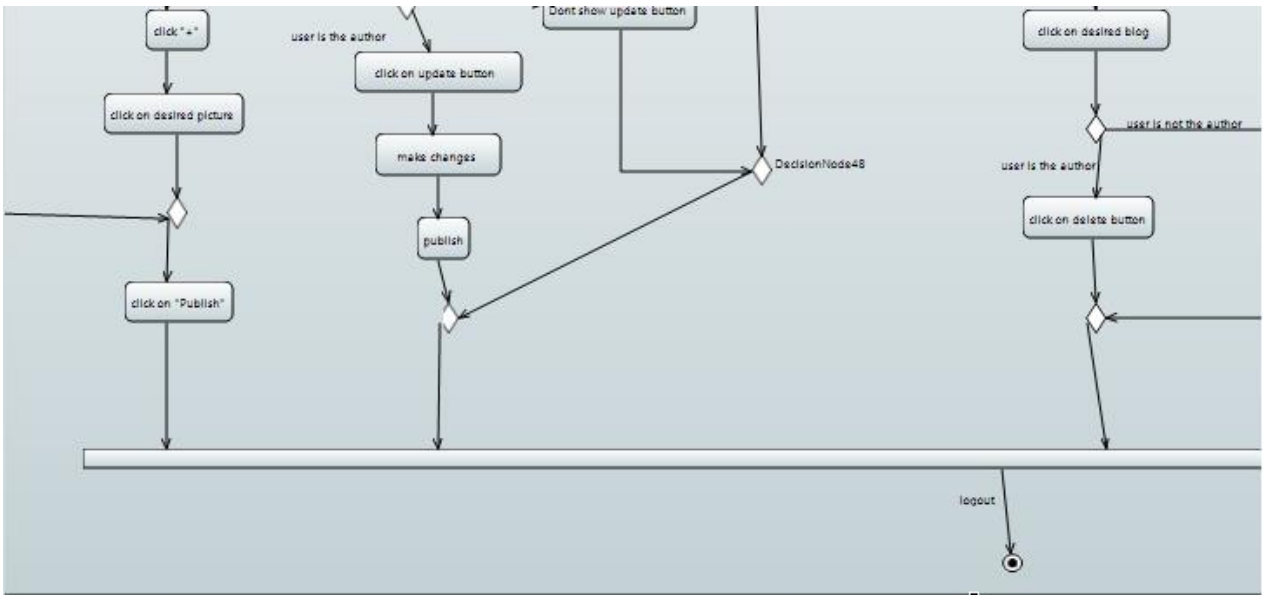
9.1.3. State Diagram

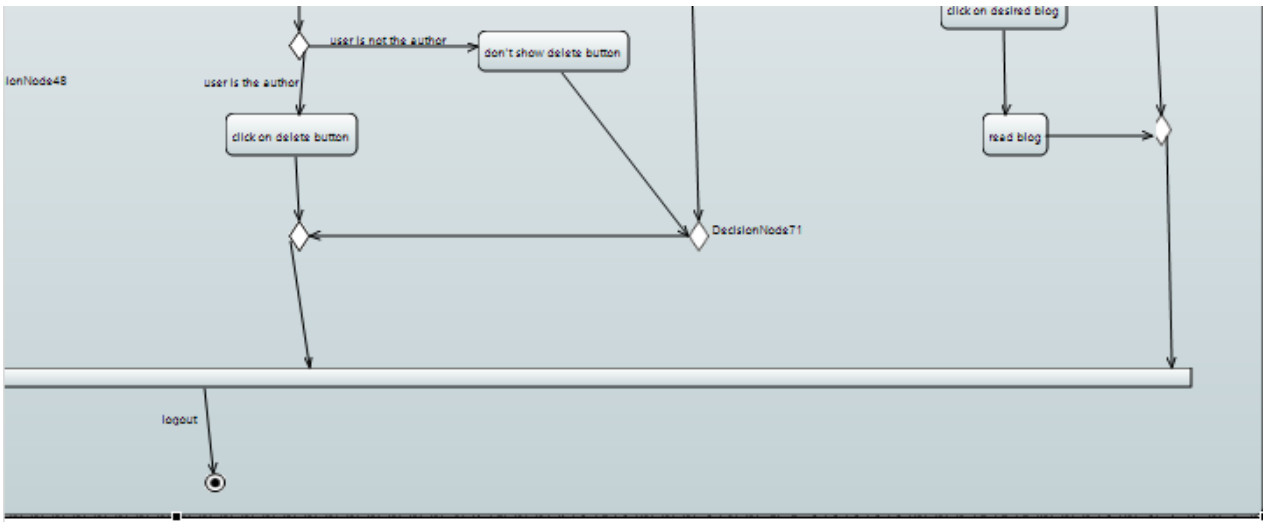


9.1.4. Activity Diagram




delete blog
search by topic or category
not found
found
click on desired blog
user is not the author
don't show delete button
user is the author
click on delete button
DecisionNode71
read blog
search by topic or category
not found
found
click on desired blog
read blog


click "+"
click on desired picture
click on "Publish"
user is the author
click on update button
make changes
publish
Dont show update button
DecisionNode48
click on desired blog
user is not the author
user is the author
click on delete button
logout


user is not the author
don't show delete button
click on desired blog
ionNode48
user is the author
click on delete button
read blog
DecisionNode71
logout