Faculty of Information Technology

Computer Science Department

Computer Science

Free Distribution Resource System(FDRS)

Graduation Project (1/2) Report

Prepared by

Student ID	Student Name
202020067	Anas alseid
201920074	Wasef Joyousi
202020088	Saif Karborani

Supervised By

Dr. Ashraf Odeh

To Obtain

BSc In Computer Science

1ST Semester / 2023/2024

Group No : CS-23-1-1-Serial(Replace serial by group serial number)

Copyright © 2023-2024 - All rights Reserved

Middle East University

Declaration

We hereby acknowledge that the work presented in this document report and the ideas based upon are the group members own unless stated otherwise and properly cited in the text and referenced at the end of the document.

Student ID	Student Name	Signature	Date
202020067	Anas alseid		
201920074	Wasef Joyousi		
202020088	Saif Karborani		

Faculty of Information Technology

Graduation Project (1/2) Report

Supervisor Approval موافقة المشرف
Approval For Submission I certify that this project report entitled "**FREE DISTRIBUTION RESOURCE SYSTEM**" was prepared by **Anas Alseid** , **Wasef Jayousi and Saif Karborani** has met the required standard fro submission in partial fulfillment of the requirements for the degree of Bachelor of science in Computer Science at MEU Approved by Signature : ..... Supervisor : Dr ..... Date:.....

Acknowledgements الشكر و العرفان

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project starting from the guidance of our supervisor

Dr. Ashraf Odeh . And …..Dr. Hani Al-Bloush and the team work with my colleagues who worked with this project Mrs/Miss !!!!

Abstract (English) المستخلص (إنجليزي)

Abstract (Arabic) المستخلص (عربي)
عنوان المشروع **المستخلص**

Table of Contents

List of Figures 9

List of Tables 11

Terminologies 12

Chapter 1 : Introduction 13

1. Background of problem statement 13

2. Description of the Problem 14

3. Description of the Suggested Solution 14

4. Literature Review (related work) 15

5. Technology and tools to be used 16

Chapter 2 : Project Plan 17

1. Project Objectives 17

2. Project Scope 17

3.Software Process Model 18

4. Project Schedule 19

5.Project Schedule chart(s) (Bar/Gantt chart) - Including Team Structure and Roles 20

Chapter 3 : Requirements and analysis 21

1. Functional Requirements 21

2. Non-functional Requirements 22

Chapter 4 : Architecture and Design 24

1. Architecture 24

2.Use Case Diagram 27

3.Use case Description/Flow of Events 28

3. Sequence Diagrams 39

4. Activity diagram 46

5. Class Diagram 56

6. Entity Relationship Diagram 57

List of Figures

Figure 2.1 19
Figure 4.1.1 24
Figure 4.2.1 27
Figure 4.3.1 39
Figure 4.3.2 39
Figure 4.3.3 40
Figure 4.3.4 41
Figure 4.3.5 41
Figure 4.3.6 42
Figure 4.3.7 42
Figure 4.3.8 43
Figure 4.3.9 43
Figure 4.3.10 44
Figure 4.3.11 44
Figure 4.3.12 45
Figure 4.3.13 45
Figure 4.4.1 46
Figure 4.4.2 47
Figure 4.4.3 48
Figure 4.4.4 49
Figure 4.4.5 50

Figure 4.4.6 50
Figure 4.4.7 51
Figure 4.4.8 52
Figure 4.4.9 53
Figure 4.4.10 53
Figure 4.4.11 54
Figure 4.4.12 55
Figure 5.1 56
Figure 6.1 57

List of Tables

Table 4.2.1 28
Table 4.2.2 29
Table 4.2.3 30
Table 4.2.4 31
Table 4.2.5 32
Table 4.2.6 33
Table 4.2.7 34
Table 4.2.8 35
Table 4.2.9 36
Table 4.2.10 37
Table 4.2.11 38

Terminologies

pertinent

Chapter 1 : Introduction

This chapter is about illustrating the description of the challenge of Solving the absence of a Free Distribution Resource System and the solution, related work to be done and the technology and tools that were going to use later so we can implement our website

1. Background of problem statement

Students and Researchers in universities currently face a significant challenge: Accessing reliable and reviewed current academic references. Many students struggle to find pertinent study materials that could enhance their academic achievements. The core issue lies in the fact that many libraries lack resource tailored to specific majors and fields of study. Moreover, numerous academic materials are expensive and not easily accessible to students, hindering their ability to obtain the most recent updates or editions of academic materials, adversely affecting the quality of student studies. There lies an opportunity to develop a system that addressees thee needs of various majors by offering an extensive collection of academic references. A system enriched with content generated by students would be an invaluable asset, enabling them to share resources and experiences with their peers. Moreover, a system that consistently integrates the latest research and information can aid students in staying informed in their studies, thereby enhancing their academic performance.

2. Description of the Problem

The critical issue we face is the absence of a "Free Distribution Resource System" , crucial for providing unrestricted access to academic materials. To address this, a digital platform can be developed, offering free access to a wide range of academic resources tailored to various fields of study. This system should streamline the distribution of current, peer-reviewed academic content, ensuring that students and researchers can easily obtain the latest information and research in their respective domains. By eliminating financial barriers, this platform aims to democratize education, fostering a more inclusive and equitable academic environment.

3. Description of the Suggested Solution

The Free Distribution Resource System (FDRS) is an innovative platform tailored for university students across majors like Information Technology, Business, Engineering, Medicine, Law, and Social Sciences. It hosts an expansive database of student-contributed academic materials, including books, videos, research papers, and more. This community-driven approach ensures a diverse and up-to-date library. Key to FDRS is its user collaboration feature, where students upload and review resources, fostering a rich learning environment. The system tracks user interactions, offering personalized histories and the ability to rate and review materials. Organized by major and equipped with advanced search capabilities, FDRS makes finding specific resources straightforward and efficient. Enhancing its responsiveness, FDRS now includes a feedback feature for users to report unallocated resources. This not only aids in continuously updating the

database but also ensures the system evolves with the academic needs of its users. By integrating this feedback loop, FDRS strengthens its role as a collaborative, adaptive educational tool in the academic landscape.

4. Literature Review (related work)

In this related work, we are going to see and maybe upgrade on other students project that made this project similar

Related work:

In this related work we have looked on another project that is similar to our Problem and we found that their system is mainly about book issues for students, what is meant here by book issues are the librarian (Admin) writes your name and details and gives you the book & tells you the time period of returning the book, so this is in real life library. but in the system it actually does a lot more, the admin can monitor the whole system .It also has facility of an online notice board where teachers and student can put up information about workshops being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the workshop can add it to the notice board. The students when they log in and have an account they can see list of books issued and its date , their return date and the can request from the librarian which is the admin to add new books by filling a request form the student made [1].

We will make ours much easier to navigate and probably more modern. Their system is about a student user and a teacher user and they both have similar features, Our system improvement would be rather than a student that has to log in and a teacher that has also to log in, we can put a guest automatic log

in with less features so that with anybody wants to enter the website without logging in he can still do the core thing that the system provides and also our system is not about subscriptions that the user has to pay to get the book , it's about a free platform of resources.

Because taking feedbacks (Requests) from large number of people is an important in the field of Business, we can take this method and add it to our website so we can make our website more attractable to users.

5. Technology and tools to be used

1. VSCODE : We used VSCODE as our work environment to

2. GITHUB : The main use of GitHub is collaborate our work together

3. External embedded applications

4. Visual Paradigm

5. FRONTEND: React Framework

6. BACKEND : Node.Js

7. DATABASE : MongoDB

Chapter 2 : Project Plan

1. Project Objectives

1. Design an advanced search functionality that allows users to find resources based on specific criteria such as subject, source, author or publication date.
2. Implement a system that enables users to easily share and upload their academic resources, fostering a collaborative educational environment.
3. To have and up-to-date database that caters the user needs.
4. To develop and design an application that is lightweight, compatible with a wide variety of devices and a user-friendly interface.
5. To validate and test all aspects of the designing and the development processes

2. Project Scope

The Free Distribution Resource System (FDRS) is envisioned as a cutting-edge digital library, providing a collaborative platform for university students to freely upload and exchange academic resources aligned with their courses and majors. This project aims to construct a robust application with advanced search capabilities, allowing users to pinpoint resources by subject, course, author, or publication date. Our objective is to maintain a contemporary database responsive to user needs, designed with a lightweight architecture for compatibility across various devices, and a focus on a user-friendly interface to facilitate ease of access.

We operate under the assumption that our users—students and faculty—are adept with basic computer and internet functions, have reliable internet

access, are keen on contributing and utilizing academic materials, and will commit to our resource-sharing guidelines. The content will be compatible with standard user software.

The success of this project is dependent on a stable, scalable server infrastructure capable of managing concurrent user requests, a reliable database system for efficient storage and retrieval of academic resources, and ongoing access to up-to-date materials. We seek collaborative engagement with university authorities for content verification and promotional activities, plus ongoing technical support to ensure smooth system operations.

Our deliverables include a fully operational web-based platform for resource sharing, a secure user authentication and authorization system, a vast database of academic materials such as books, videos, and research papers, alongside user-friendly navigation. Comprehensive documentation for users and system maintenance, a feedback system for users to suggest new resources or features, and regular system updates and enhancements driven by user input and technological progress are also part of our commitment to delivering an exceptional tool for the academic community.

3.Software Process Model

The Software Development Life Cycle (SDLC) is an organized approach to crating high-quality software efficiently and cost-effectively in minimal time its primary aim is to deliver exceptional software that surpasses customer expectations. The SDLC methodically details each phase of the development process, with specific processes and deliverables for each stage. Following the SDLC framework accelerates the development pace and reduces risks and expenses compared to the production methodologies. This approach is particularly suitable for our project, aligning well with our system's requirements and objectives

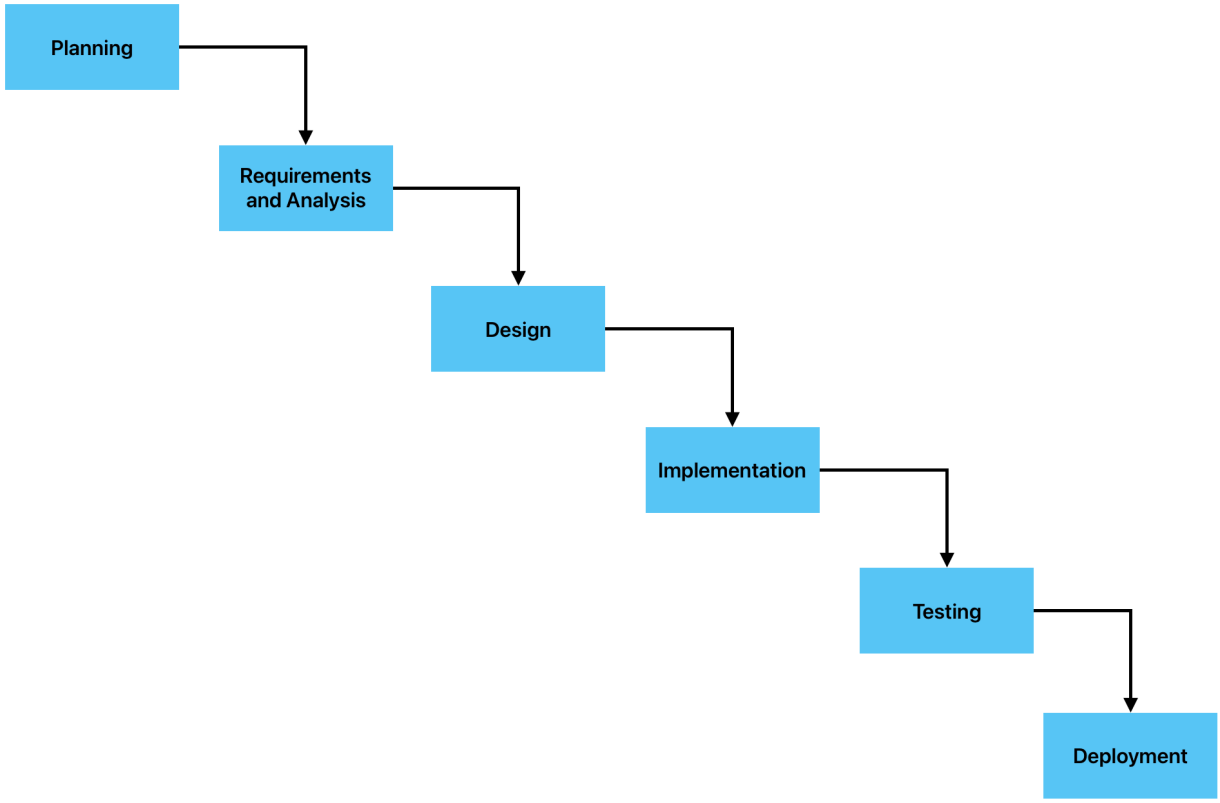

Figure 2.1

4. Project Schedule

5.Project Schedule chart(s) (Bar/Gantt chart) - Including Team Structure and Roles

Chapter 3 : Requirements and analysis

1. Functional Requirements

1. **User Authentication and Authorization** : Users should be able to create an account, login, and access only the features that they are authorized to use.

2. **Recourse Catalog** : The system should provide a searchable catalog of all recourse available in the FDRS along with their metadata such as title, author, genre, publication year, publisher, and availability status.

3. **Recourse view** : Users should be able to view and read recourse online or download them for offline reading.

4. **Recourse Search** : Users should be able to search for recourse by title, author, keyword, or category.

5. **Search Feedback Integration**: When users search for resources that are not available in the FDRS catalog, they should be prompted with an option to send feedback directly from the search text .

6. **User Interaction** : Users should be able to rate, review, and comment on recourse, as well as share them on social media.

7. **Admin Panel** : The system should have an admin panel to manage users, recourse, and get feedback from users for unfound search resources.

8. **User Profile Management :** Users should be able to manage their profile information, update their preferences, and view their reading history.

9. **User Upload** : Users should be able to upload resources on the website.

10. **User Favorite List** : Users should be able to add their favorite Resource on their List.

11. **Forget password** : User can regain access to their account by resetting their forgotten password.

2. Non-functional Requirements

1. **Performance** : The system should handle many concurrent users and transactions without slowing down or crashing, with fast page load times and smooth scrolling for online reading.

2. **Availability** : The system should be available 24/7, with minimal downtime for maintenance or upgrades, and with high availability to ensure users can access the system whenever they need it.

3. **Reliability** : The system should be reliable and provide accurate information, without losing data or causing errors, with frequent backups to ensure that data is not lost.

4. **Scalability** : The system should be able to scale up or down as needed to accommodate changes in user traffic or library collections, with the ability to add more servers and storage space as required.

5. **Security** : The system should be secure, with measures in place to protect user data, prevent unauthorized access, and detect and respond to security breaches, with SSL encryption and strong password policies.

6. **Usability** : The system should be easy to use, with an intuitive user interface and clear instructions for performing tasks, with clear and concise user documentation available.

7. **Compatibility** : The system should be compatible with different web browsers and operating systems, and support different languages and character sets, with a responsive design to ensure that it works on different devices.

8. **Maintainability** : The system should be easy to maintain and upgrade, with clear documentation and well-structured code, with change management policies in place to prevent unplanned changes.

Chapter 4 : Architecture and Design

1. Architecture

Overview

This section delineates the layered architecture of the Free Distribution Resource System (FDRS), ensuring a seamless user experience and robust system functionality. The architecture is composed of three primary layers—Presentation Layer, Application Layer (inclusive of Business Logic), and Data Layer.

Architecture Diagram

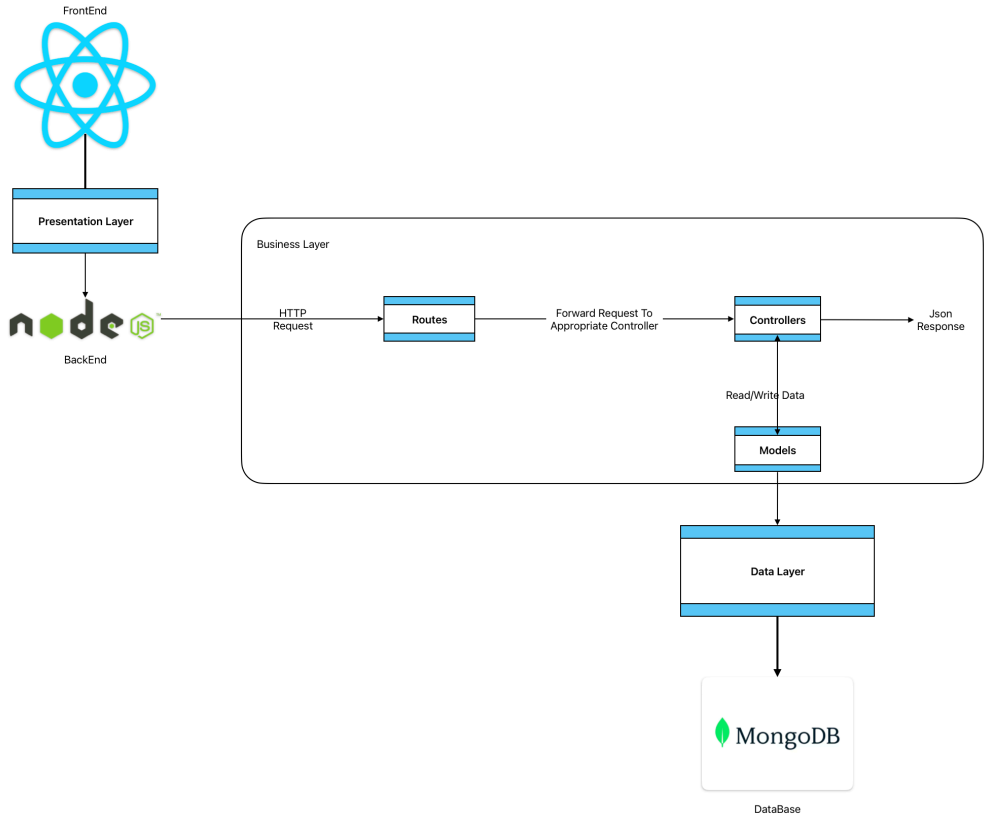

Figure 4.1.1

High-Level Architecture of FDRS

Presentation Layer (Frontend)

Description

Serving as the user-system interface, the Presentation Layer is developed with the React.js framework, known for its responsive and dynamic capabilities.

Components

Browser: Users access the application through web browsers that render the user interface.

User Interface: It includes dashboards, resource listings, and configurable settings.

Client-Side Logic: Implemented in JavaScript, with React facilitating dynamic updates without page reloads.

Application and Business Logic Layer (Backend)

Description

The backend, powered by Node.js, processes user requests, enacts business logic, and interfaces with the Data Layer.

Components

Server: The Node.js server handles HTTP requests and generates responses.

Business Logic: Algorithms and data processing are executed here.

Application Services: Includes additional services like email notifications.

Data Layer (Database)

Description

MongoDB, a NoSQL database, is utilized for data storage and retrieval, forming the Data Layer.

Components

Database Management System (DBMS): MongoDB manages data transactions.

Data Models

Data Flow

Data flow is initiated from the Presentation Layer by user interactions, creating HTTP requests handled by the Node.js server, which, in turn, interacts with MongoDB to retrieve or modify data, with responses relayed back dynamically.

2.Use Case Diagram

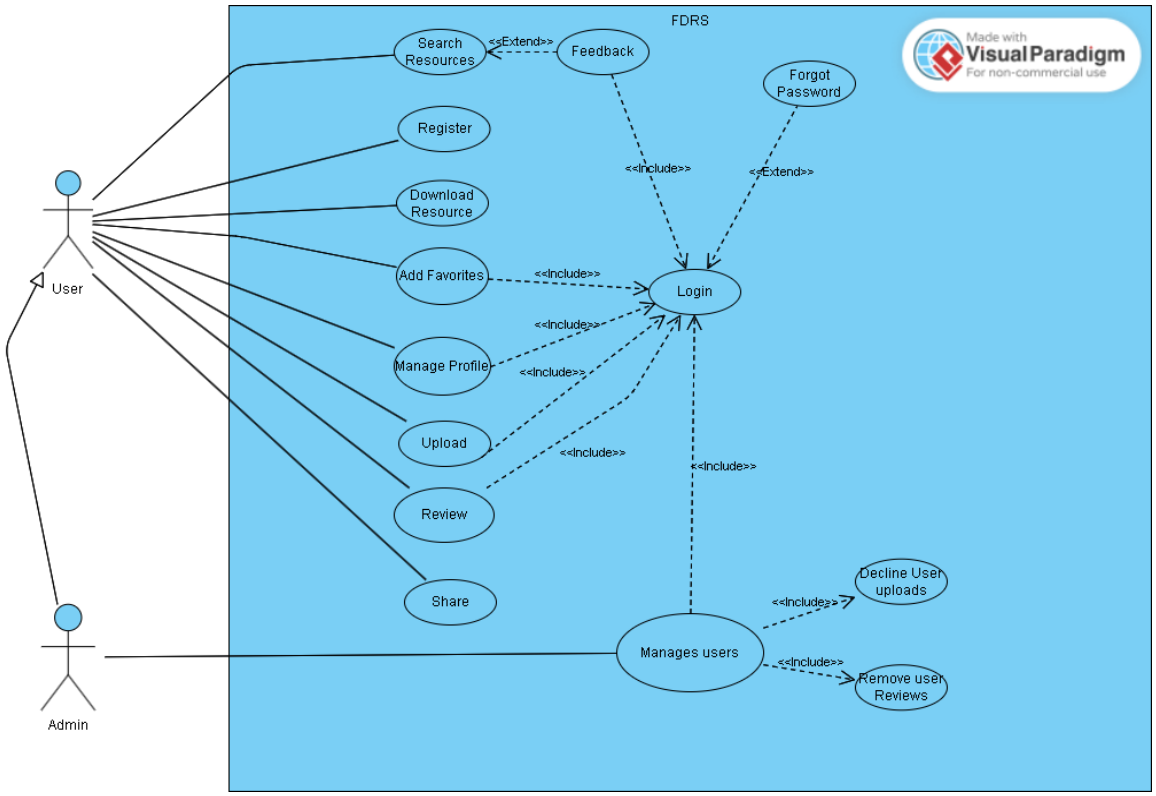

Figure 4.2.1

3.Use case Description/Flow of Events

Use Case name:	Login
Use Case ID:	1 | High
Description:	This use case describes how the actor logs into the System.
Primary Actor:	User/admin.
Secondary Actor:	Registration system.
Pre-condition:	The user must register (use case must be performed first).
Main flow of events:	User/admin: 1. The system requests that the actor enter his/her E-mail and password. 2. The actor enters his/her E-mail and password and presses 'login' option. 3. The system validates the entered E-mail and password and logs the actor into the system (Checks against the database). Sub-Flow(Forgot-password) : 1. The system requests that the user enter his email 2. User enter his email 3. The system sends an email to the user mailbox
Alternative Flow of events:	• The system displays an error message that the entered E-mail and/or password is invalid. • The system requests that the actor re-enters his/her E-mail and password.
Post-condition:	• The user that logins in will be able to do his features (review, upload, track list). • The admin That login in will have the Extra feature of managing the users.

Table 4.2.1

Use Case name:	Register(Create new account)
Use Case ID:	1 | High
Description:	This use case describes how an actor creates a new user account.
Primary Actor:	User.
Secondary Actor:	Registration system.
Pre-condition:	None.
Main flow of events:	1. The actor clicks on (Register) option from the home page. 2. The system prompts the actor to enter his/her Username, E-mail and password and confirm-password. 3. The system validates the entered E-mail and password via the registration system meaning checking if the email is reused and password match in DBthen redirects the actor to the login page (A1).
Alternative Flow of events:	A1: 1. The system displays an error message that the password and confirm-password do not match. 2. The system displays an error message That password must be at least 8 minimum length and min-length for username(12) and email(15) 3. The system requests that the actor re-enters his/her username, E-mail and password. 4. The system displays to the user that the Email or username is already in use.
Post-condition:	Added the user to the database.

Table 4.2.2

Use Case name:	Search
Use Case ID:	2 | Low
Description:	This use case describes how the actor Search through resources.
Primary Actor:	User/admin.
Secondary Actor:	...
Pre-condition:	None.
Main flow of events:	User/admin: 1. The user uses the search bar to type the resources that he is looking for based on title or author name. 2. The system will scan the database based on the search. 3. Then the system returns the desired Search resource based on the title or author name.
Alternative Flow of events:	• The system tells the user that the search is not found. • The system also tells the user if he would like to submit a feedback.
Post-condition:	The user should then be able to see the resource.

Table 4.2.3

Use Case name:	Download
Use Case ID:	2 | Low
Description:	This use case describes how the actor can Download a resource from the website
Primary Actor:	User
Secondary Actor:	
Pre-condition:	None
Main flow of events:	1. The user selects a resource to download. 2. The system searches for the resource through the DB using an ID and gets the file-path. 3. The system downloads the resource through the file system (uploads) using the file path and directly auto downloads through the web.
Alternative Flow of events:	• The systems return to the user that the resource isn't available to download (not
Post-condition:	The user has the resource on his device, drive.

Table 4.2.4

Use Case name:	Add Favorites
Use Case ID:	1 | High
Description:	This use case describes how the user can add his Favorite list of resources
Primary Actor:	User/admin
Secondary Actor:	
Pre-condition:	The user should have an account and logged in
Main flow of events:	1. The user selects the resource. 2. The user clicks on add to favorites. 3. The system adds the favorite resource to the DB.
Alternative Flow of events:	• The system displays an error message.
Post-condition:	The user can view his favorite list.

Table 4.2.5

Use Case name:	Upload
Use Case ID:	1 | High
Description:	This use case describes how the user can upload resources on the website
Primary Actor:	Users/admin.
Secondary Actor:	Admin.
Pre-condition:	Logged in.
Main flow of events:	1. The user clicks on the upload button. 2. The website displays a proper form with fields for the user to upload what kind of resource. 3. The user inputs the fields (title, author, description, pdf and image) and press upload. 4. The system validates and proceeds to the database to add the resource temporally. 5. The admin reviews it and accepts (A1) ,but if user is an admin then it is uploaded automatically 6. The system returns to the user that upload is successful and has been added to the website.
Alternative Flow of events:	• The system returns that the upload is invalid meaning that the input must meet minimum length and only pdf and (jpg, jpeg, png) is allowed. • The admin Declines the upload and it is deleted from Db and upload directory (backend) (A1).
Post-condition:	Users can see the uploaded resource.

Table 4.2.6

Use Case name:	Review
Use Case ID:	1 | High
Description:	The use case describes how the user can review on the resource, in which he can put a comment.
Primary Actor:	User/comment.
Secondary Actor:	Admin.
Pre-condition:	Logged in.
Main flow of events:	1. User selects the resource to review. 2. The website displays a comment form with a field for the user to enter their comment and submits. 3. The system saves the review in the DB. 4. The system returns the review and displays it on
Alternative Flow of events:	• The review may be reviewed by the admin and deletes it. • The comment must have a minimum length of 1
Post-condition:	Users can see the review on the resource.

Table 4.2.7

Use Case name:	Share
Use Case ID:	2 | Low
Description:	This use case describes how the user can share the resource
Primary Actor:	User/admin
Secondary Actor:	
Pre-condition:	None.
Main flow of events:	1. The user selects the resource that he wants to share. 2. The system returns a link of the resource to the user. 3. The user copies the link.
Alternative Flow of events:	• The system fails providing the link.
Post-condition:	The system delivers The link to the user.

Table 4.2.8

Use Case name:	Manage users
Use Case ID:	1 | High
Description:	This use case describes how the admin validates the users uploads and reviews.
Primary Actor:	Admin.
Secondary Actor:	User.
Pre-condition:	Logged in as an admin.
Main flow of events:	1. The system returns to the admin the upload's of the users. 2. Admin accepts them. 3. The system sends an email to the user that their upload has been approved. 1. As for the review (comments of users) 2. The admin clicks on the users comment 3. Hit delete button 4. System returns that comment has been deleted.
Alternative Flow of events:	• The admin declines. • The system sends an email to the user that the upload has been rejected.
Post-condition:	The admin can Reject users uploads and reviews.

Table 4.2.9

Use Case name:	Manage profile
Use Case ID:	1 | High
Description:	This use case describes how the user can edit his username or password.
Primary Actor:	User.
Secondary Actor:	
Pre-condition:	Logged in
Main flow of events:	1. The user clicks on user icon or manages profile button 2. The system displays the user information. 3. The user edits their profile information and saves. 4. The system updates the user username or email in DB. 5. The system return to the user that the Profile
Alternative Flow of events:	• The system returns that the edit is invalid minimum length. • The system returns that the username or email is already in use.
Post-condition:	User is able to change his profile settings.

Table 4.2.10

Use Case name:	Feedback
Use Case ID:	2 | Low
Description:	
Primary Actor:	User.
Secondary Actor:	Admin
Pre-condition:	Search and log in use case must be performed first.
Main flow of events:	1. User searches and resource is not found. 2. User Shares his feedback by clicking a button 3. The system will catch the result text. 4. The system will store it in the database for
Alternative Flow of events:	• System fails to get user feedbacks. • System returns that the input is invalid must be a minimum length.
Post-condition:	User will be able to share his feedbacks.

Table 4.2.11

3. Sequence Diagrams

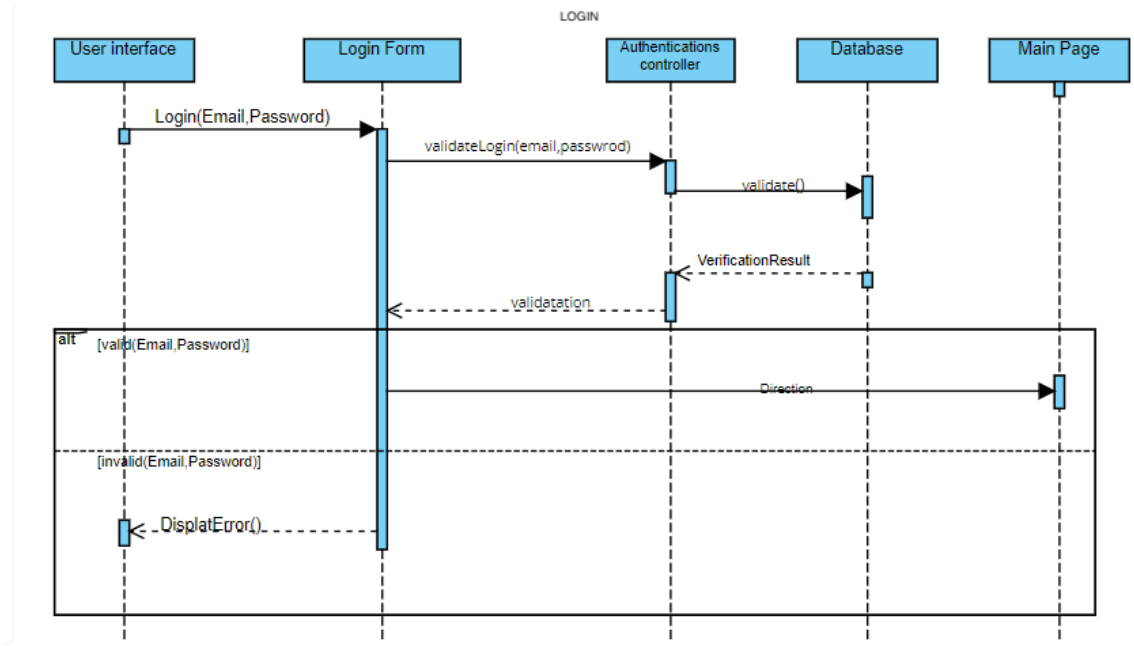

Figure 4.3.1

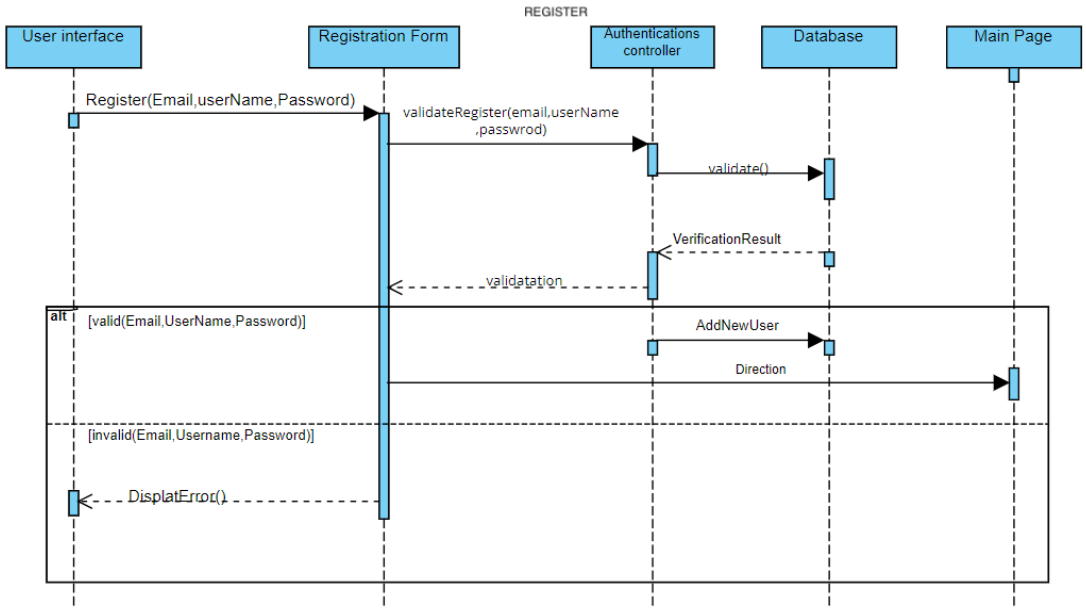

Figure 4.3.2

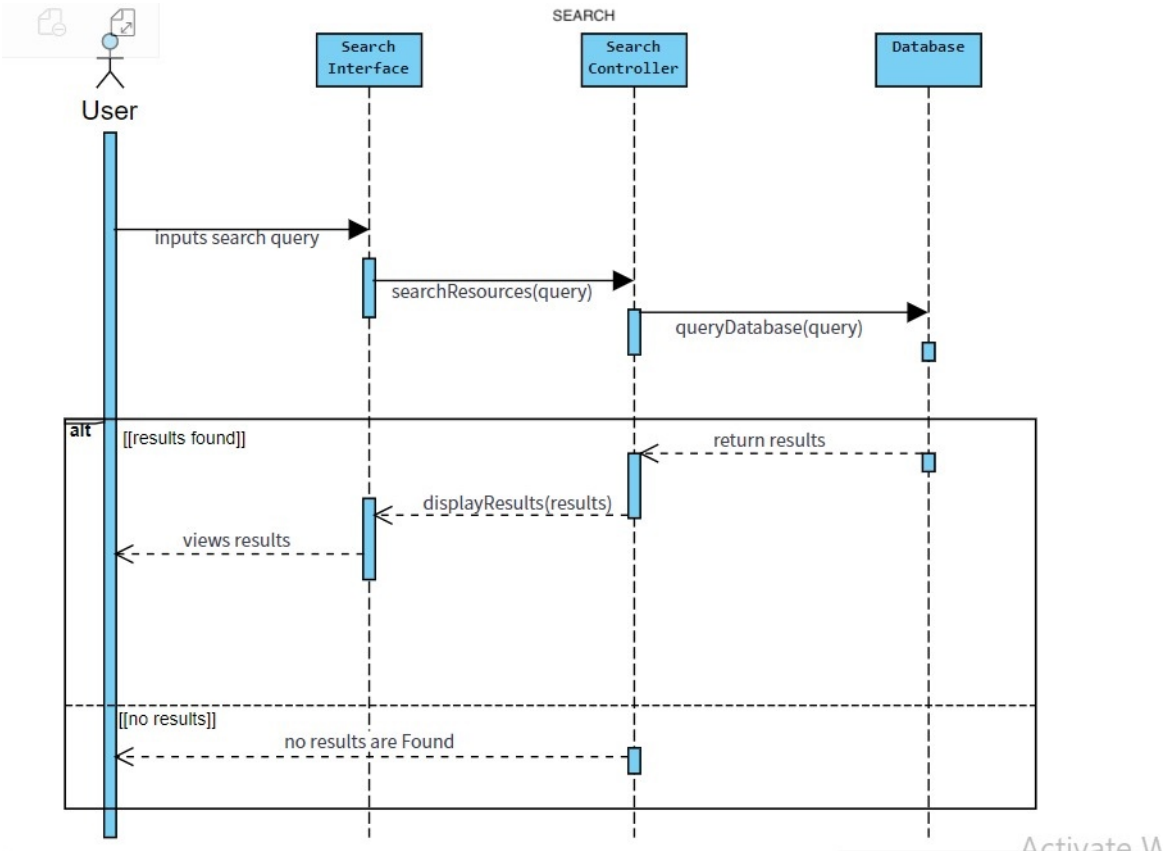
SEARCH
User
Search Interface
Search Controller
Database
inputs search query
searchResources(query)
queryDatabase(query)
alt
[[results found]]
return results
displayResults(results)
views results
[[no results]]
no results are Found

Figure 4.3.3

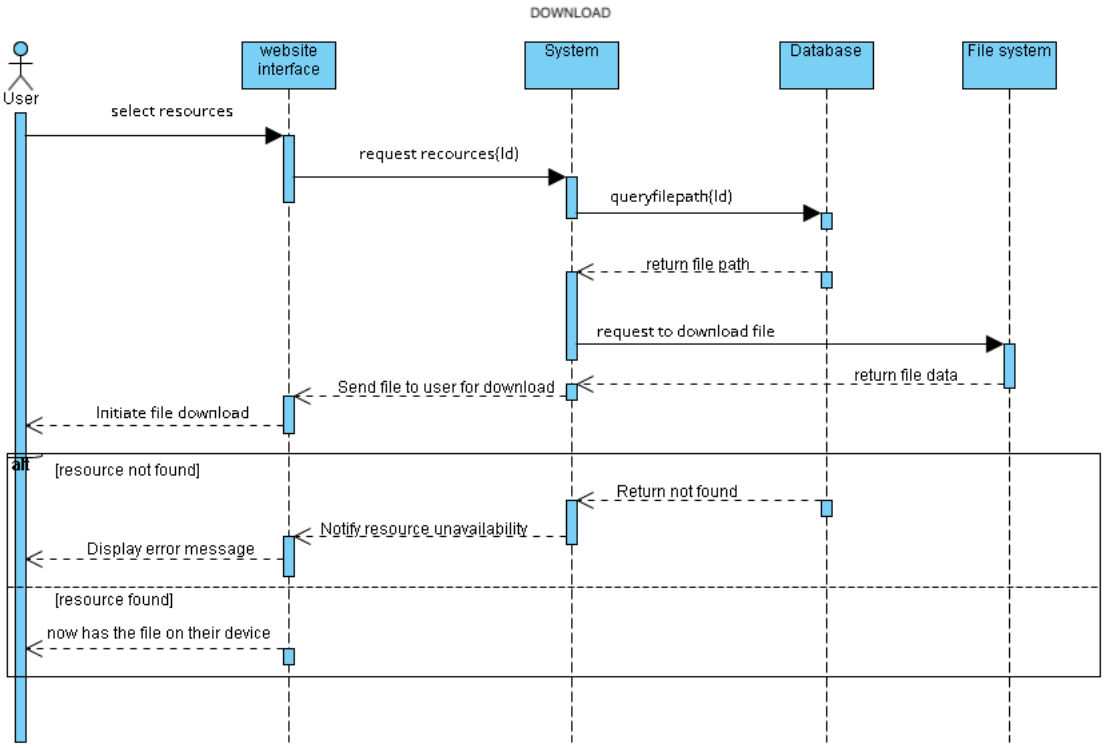

Figure 4.3.4

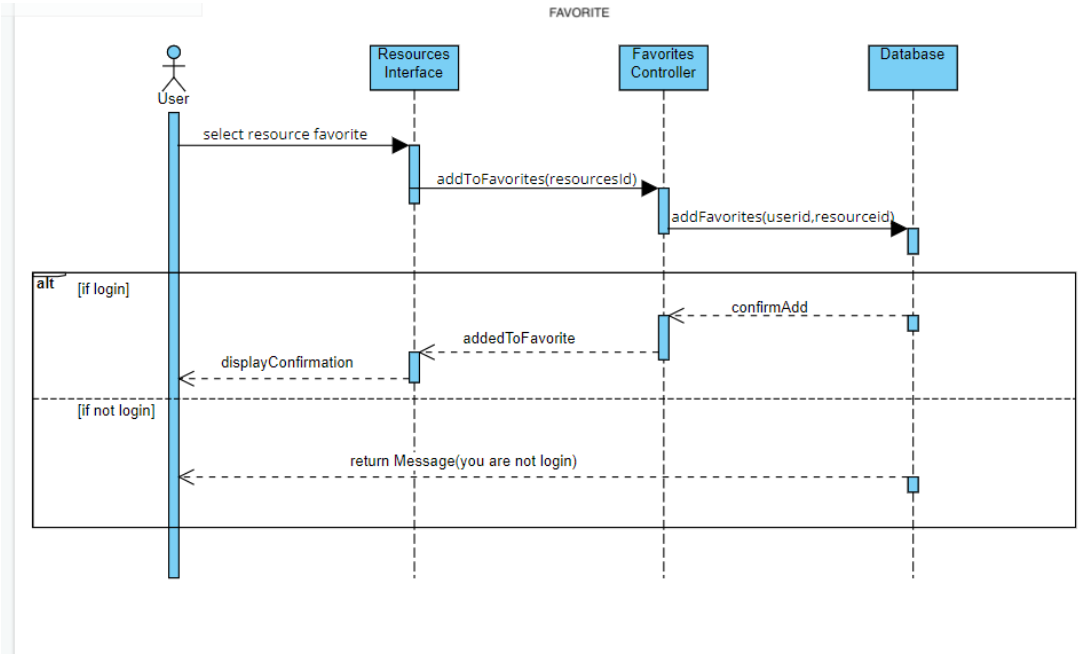

Figure 4.3.5

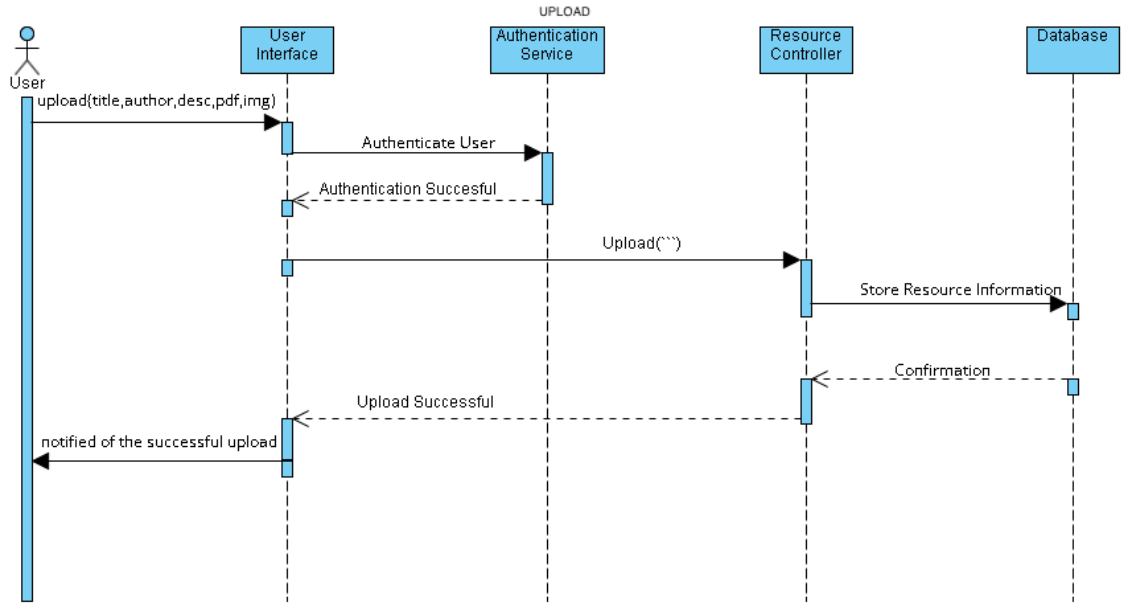

Figure 4.3.6

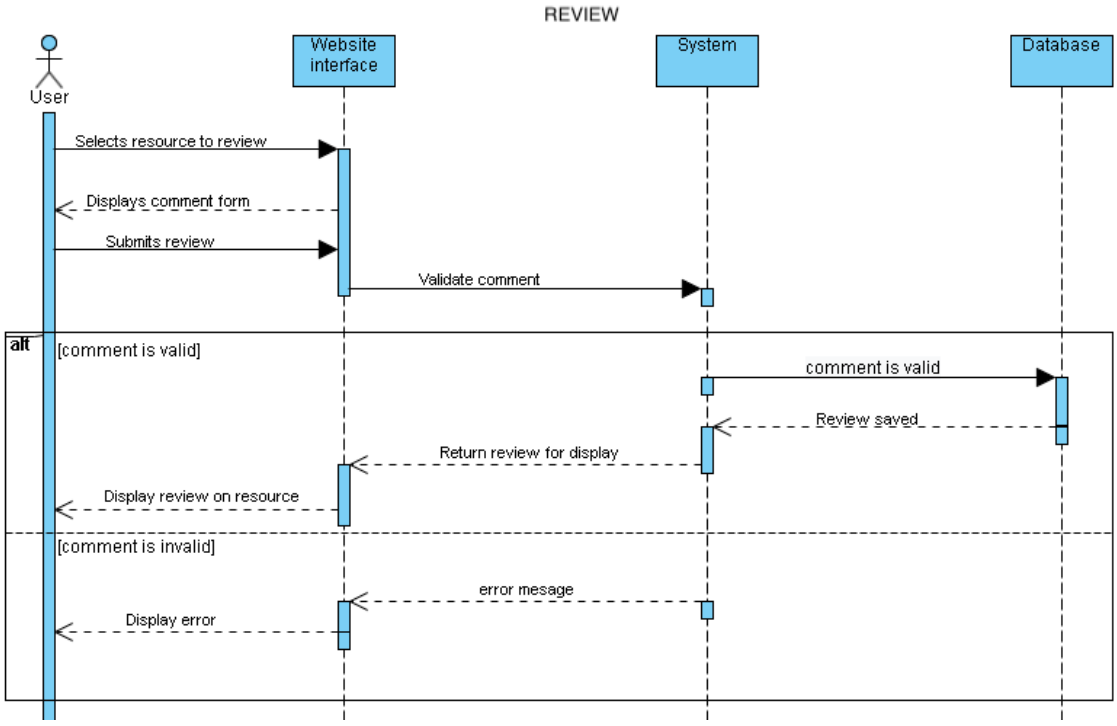

Figure 4.3.7

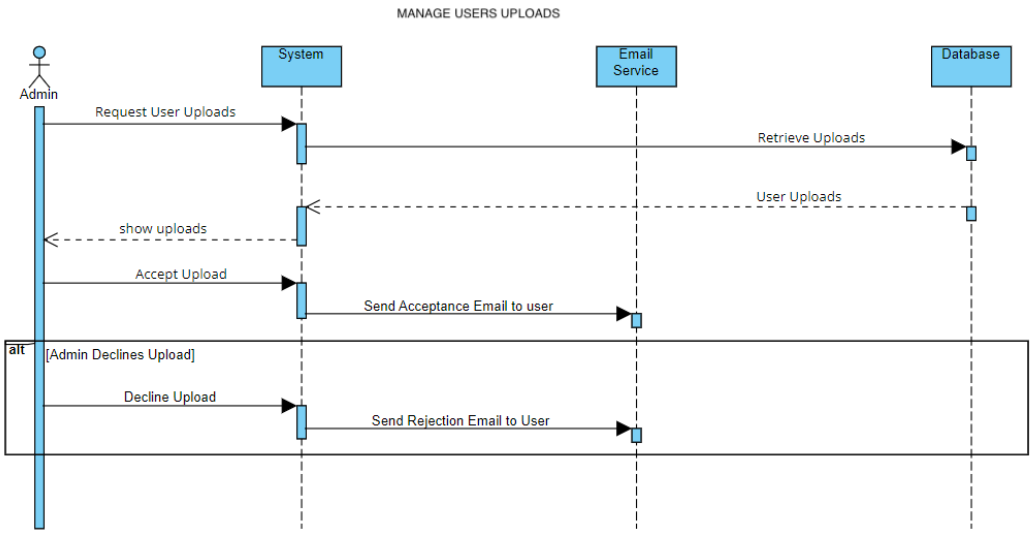

Figure 4.3.8

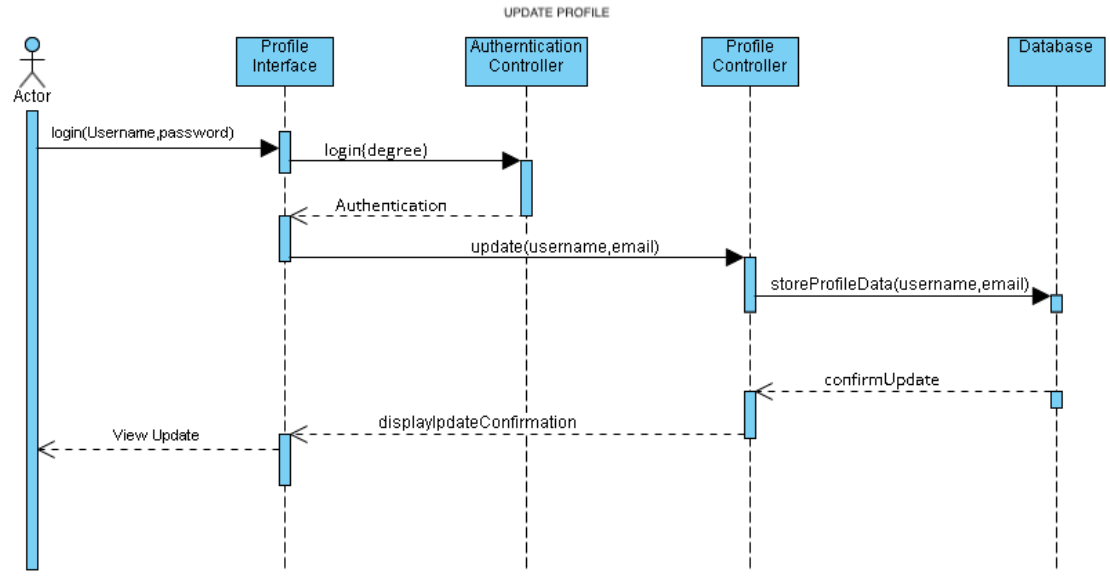

Figure 4.3.9

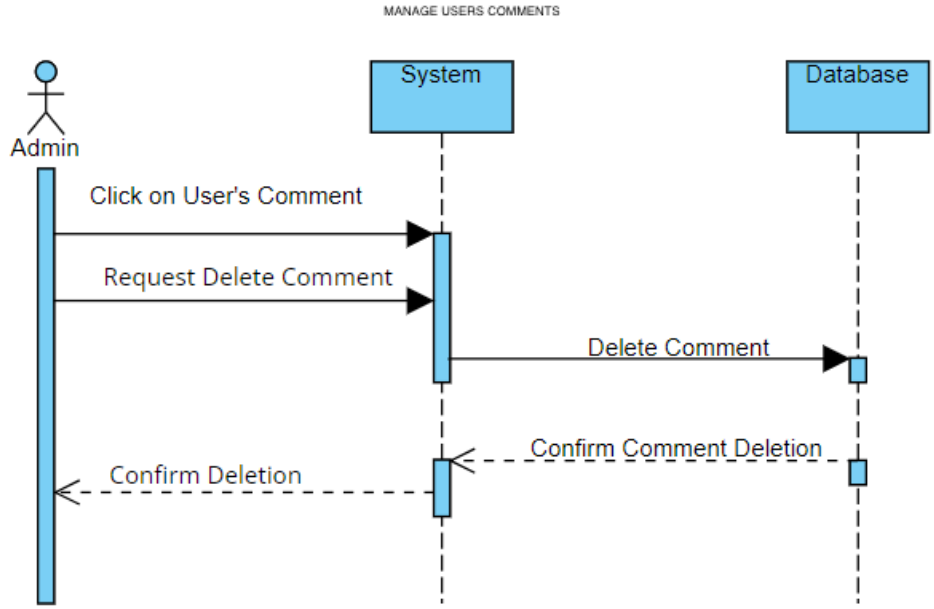

Figure 4.3.10

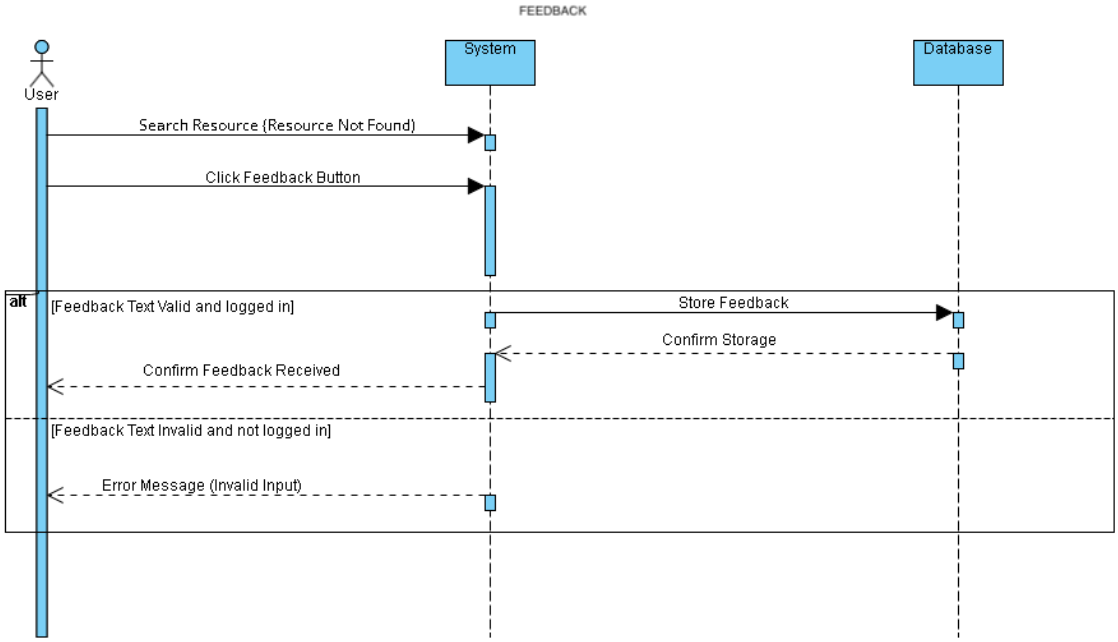

Figure 4.3.11

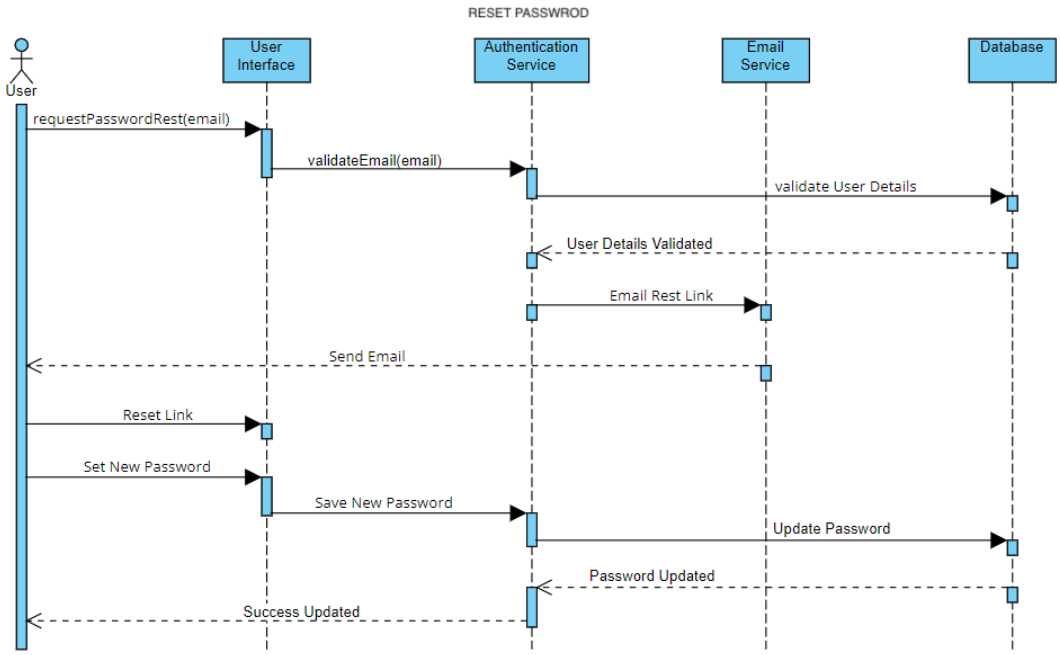

Figure 4.3.12

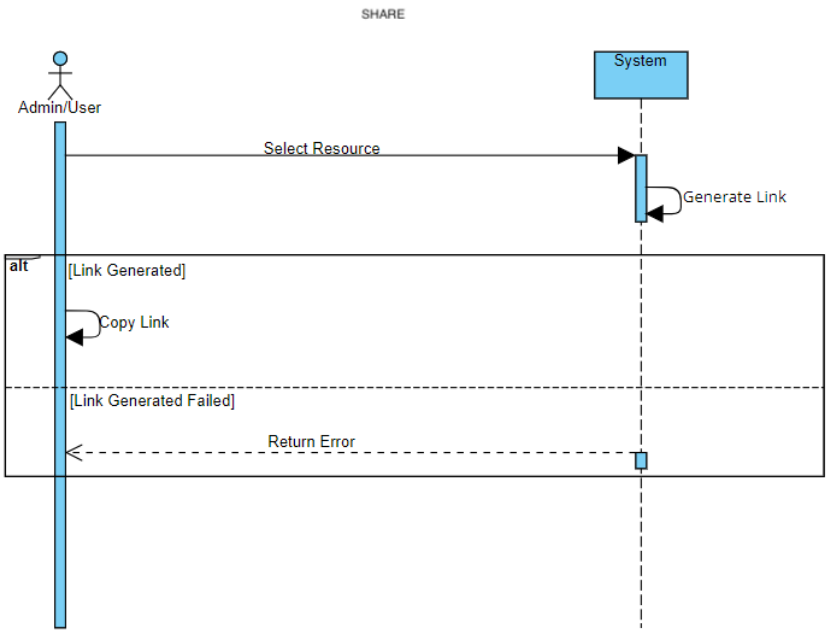

Figure 4.3.13

4. Activity diagram

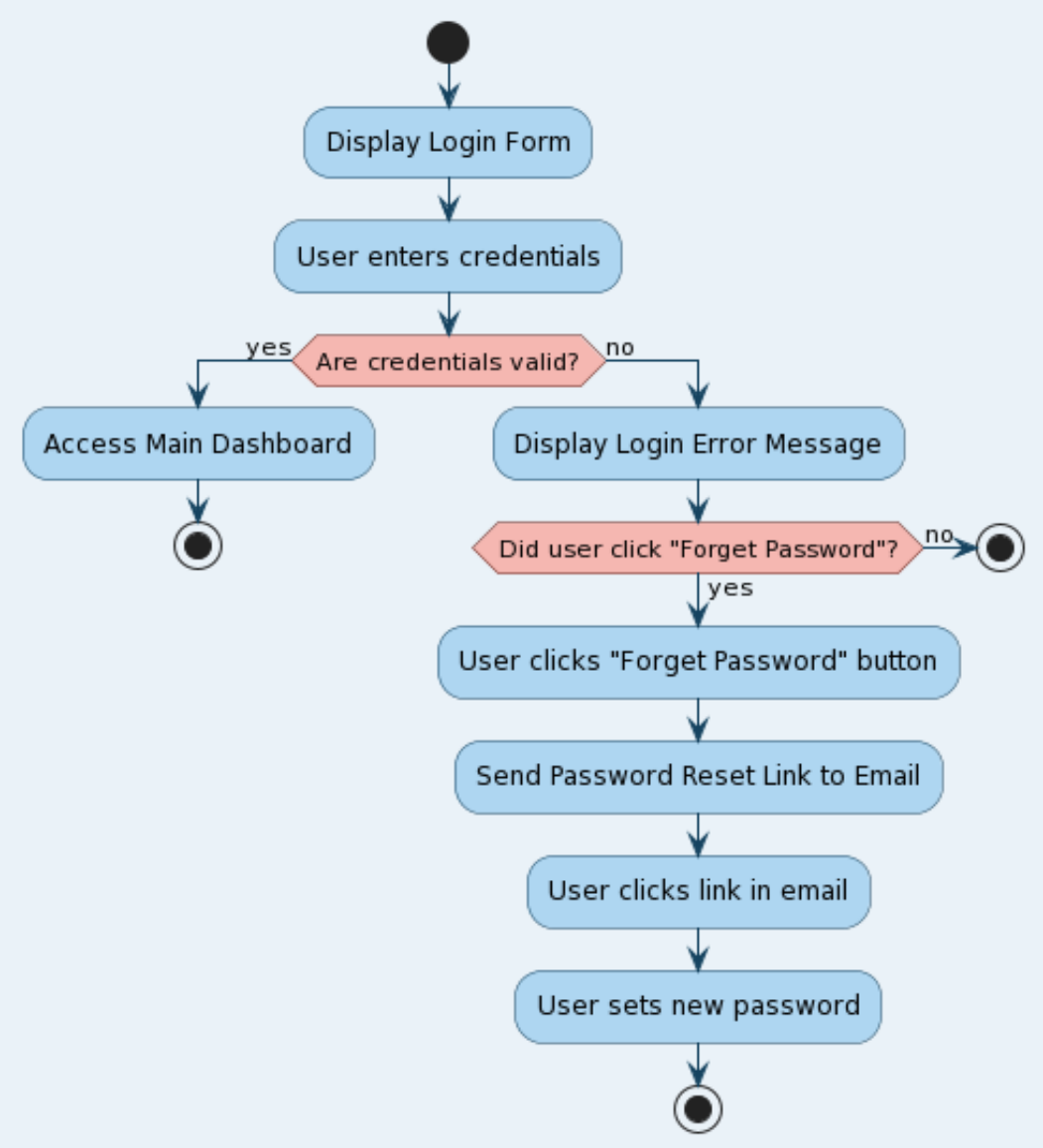

Figure 4.4.1

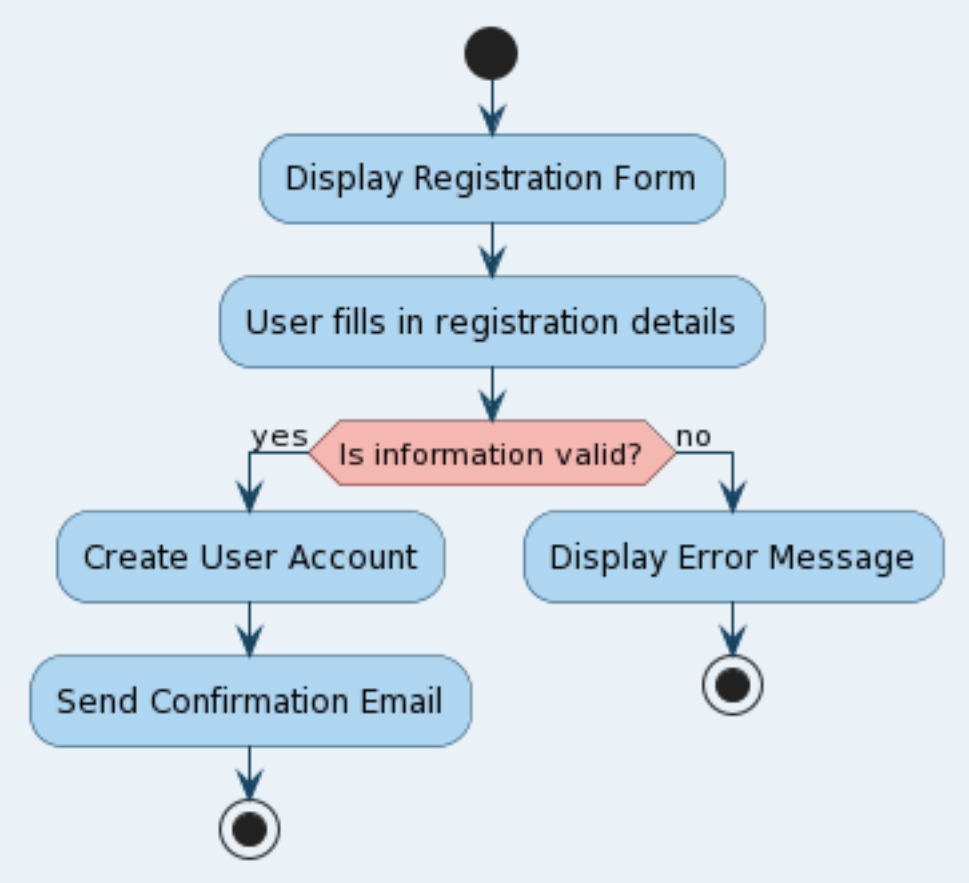
Display Registration Form
User fills in registration details
yes
Is information valid?
no
Create User Account
Display Error Message
Send Confirmation Email

Figure 4.4.2

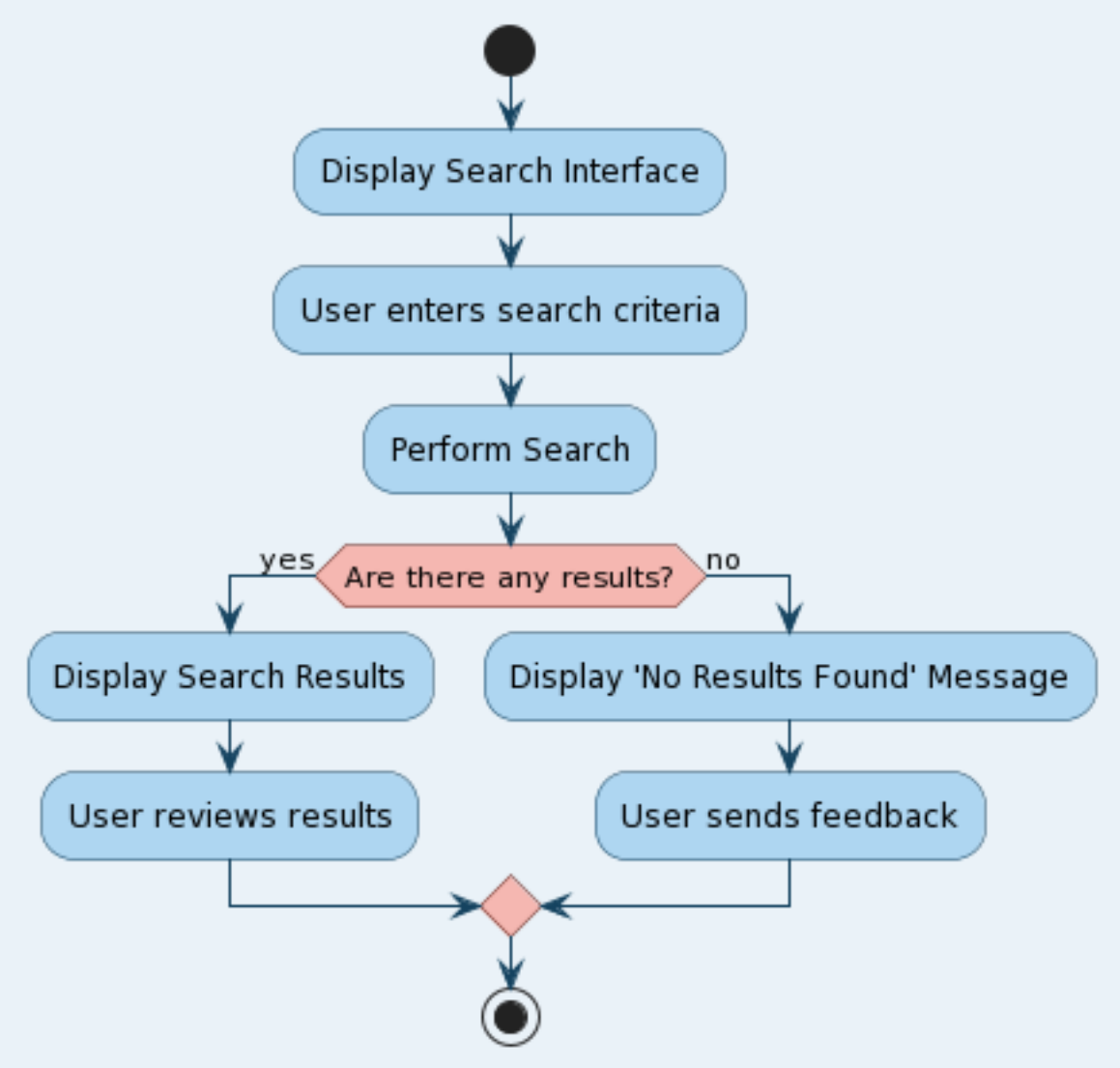
Display Search Interface
User enters search criteria
Perform Search
Are there any results?
yes
no
Display Search Results
Display 'No Results Found' Message
User reviews results
User sends feedback

Figure 4.4.3

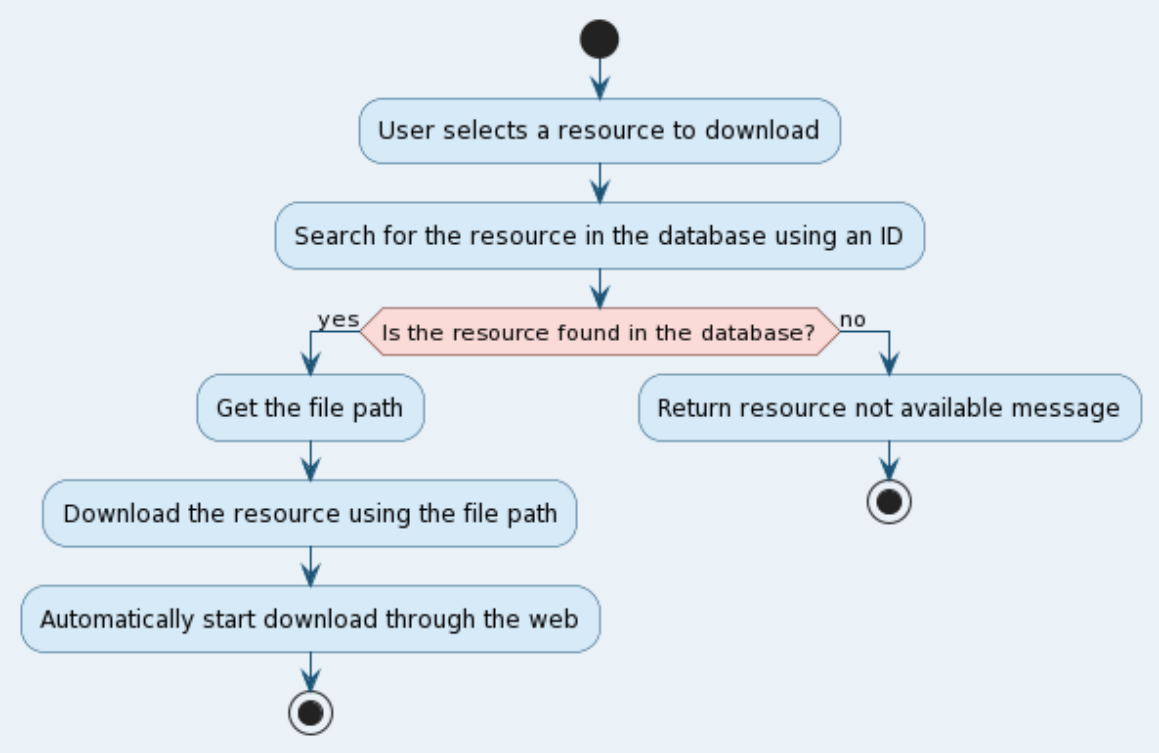
User selects a resource to download
Search for the resource in the database using an ID
Is the resource found in the database?
yes
no
Get the file path
Return resource not available message
Download the resource using the file path
Automatically start download through the web

Figure 4.4.4

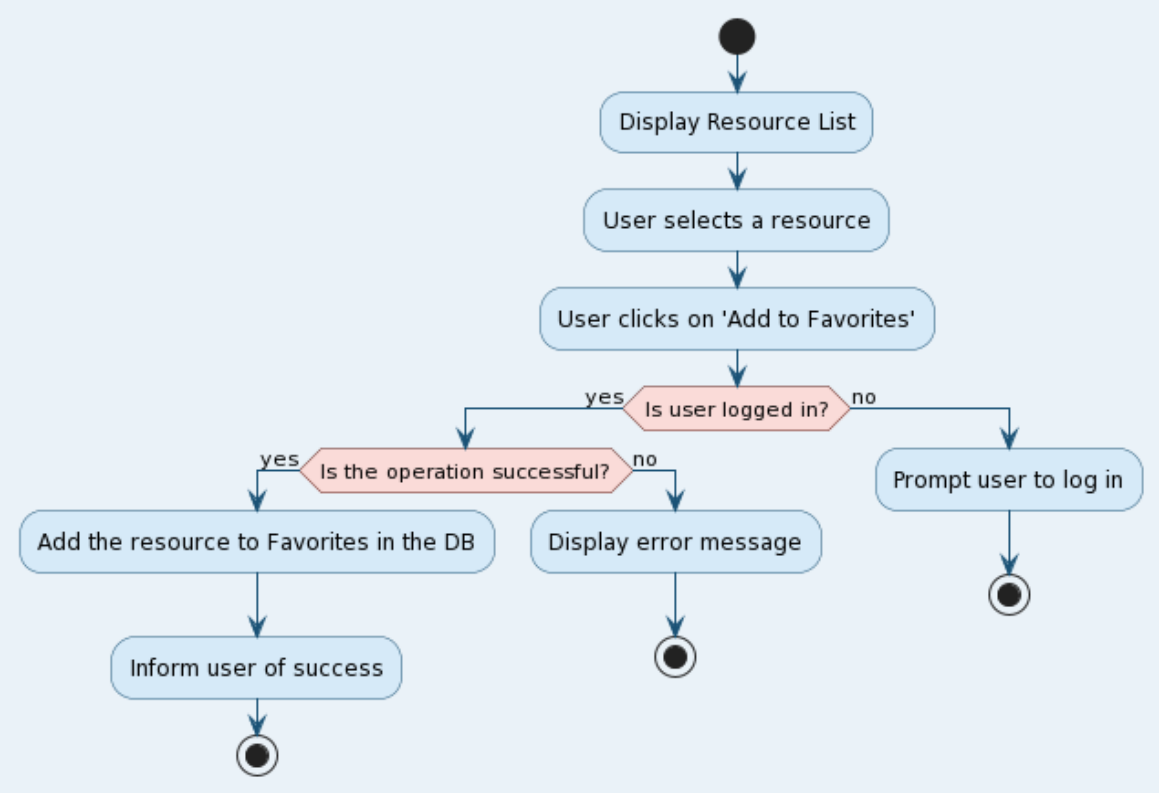

Figure 4.4.5

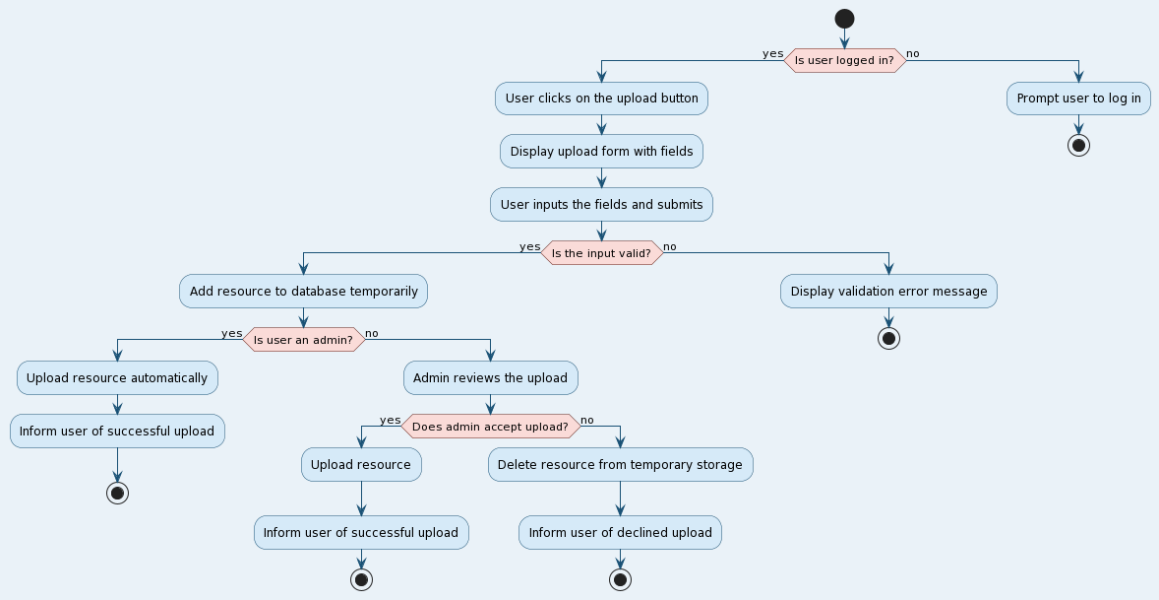

Figure 4.4.6

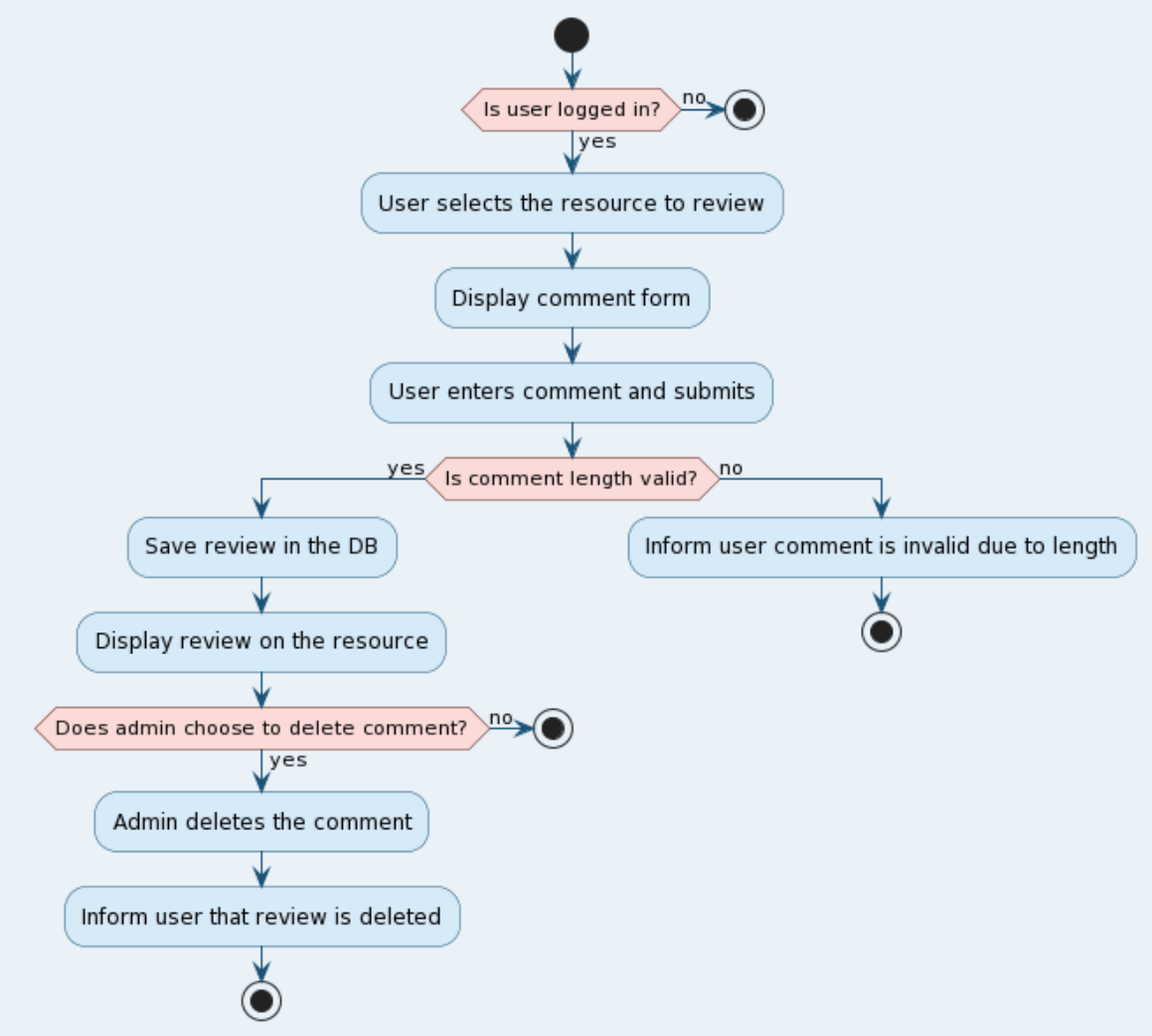
Is user logged in?
no
yes
User selects the resource to review
Display comment form
User enters comment and submits
yes
Is comment length valid?
no
Save review in the DB
Inform user comment is invalid due to length
Display review on the resource
Does admin choose to delete comment?
no
yes
Admin deletes the comment
Inform user that review is deleted

Figure 4.4.7

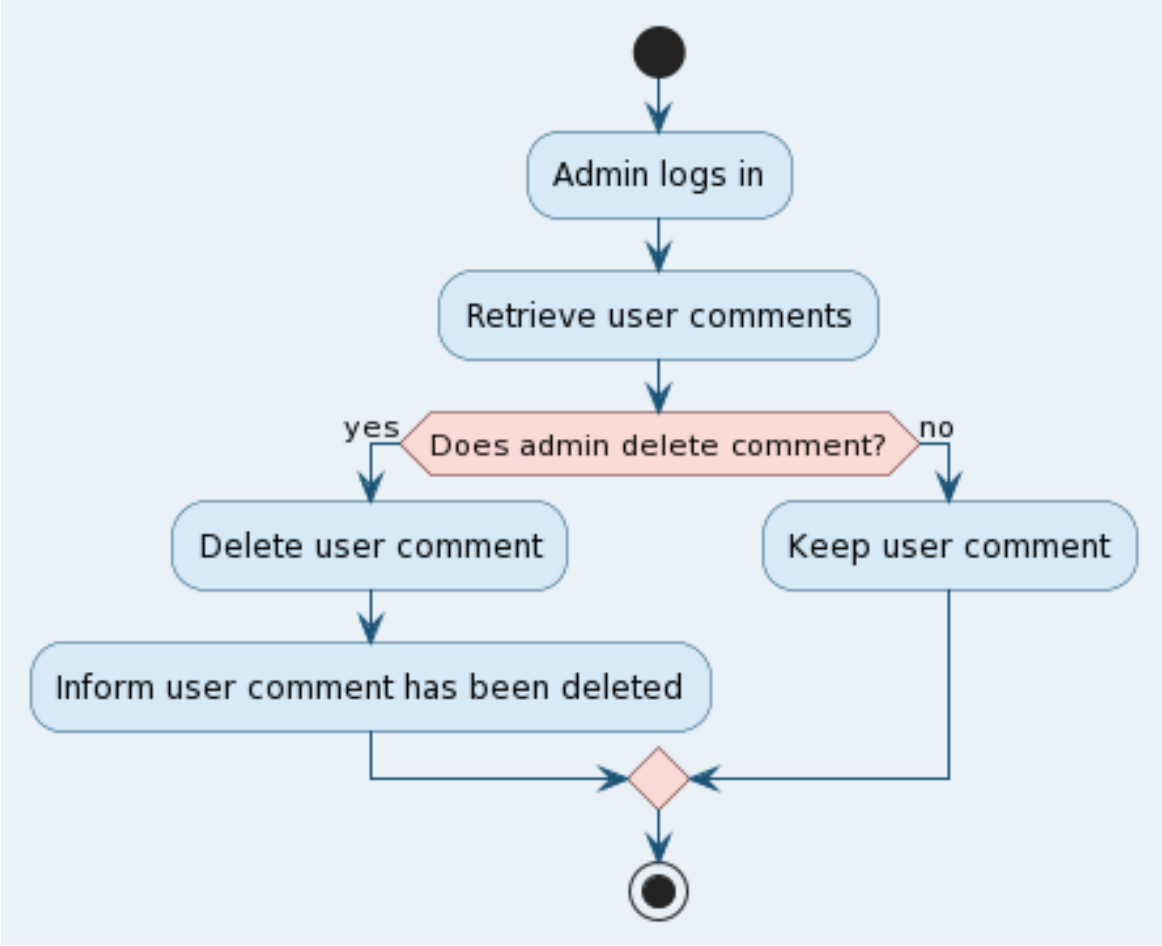
Admin logs in
Retrieve user comments
yes
Does admin delete comment?
no
Delete user comment
Keep user comment
Inform user comment has been deleted

Figure 4.4.8

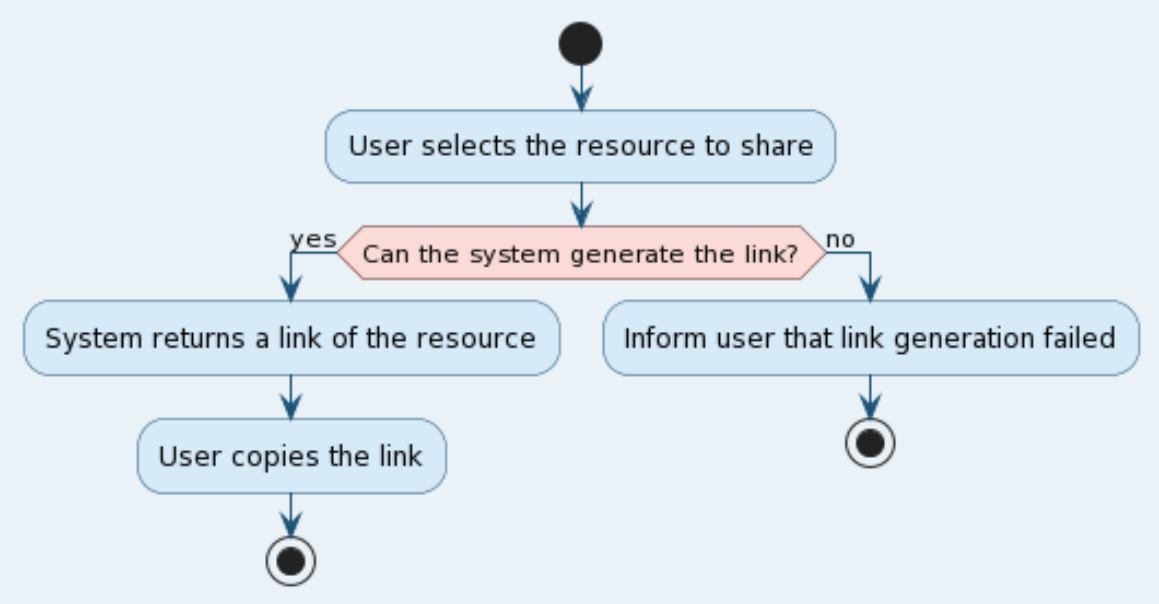

Figure 4.4.9

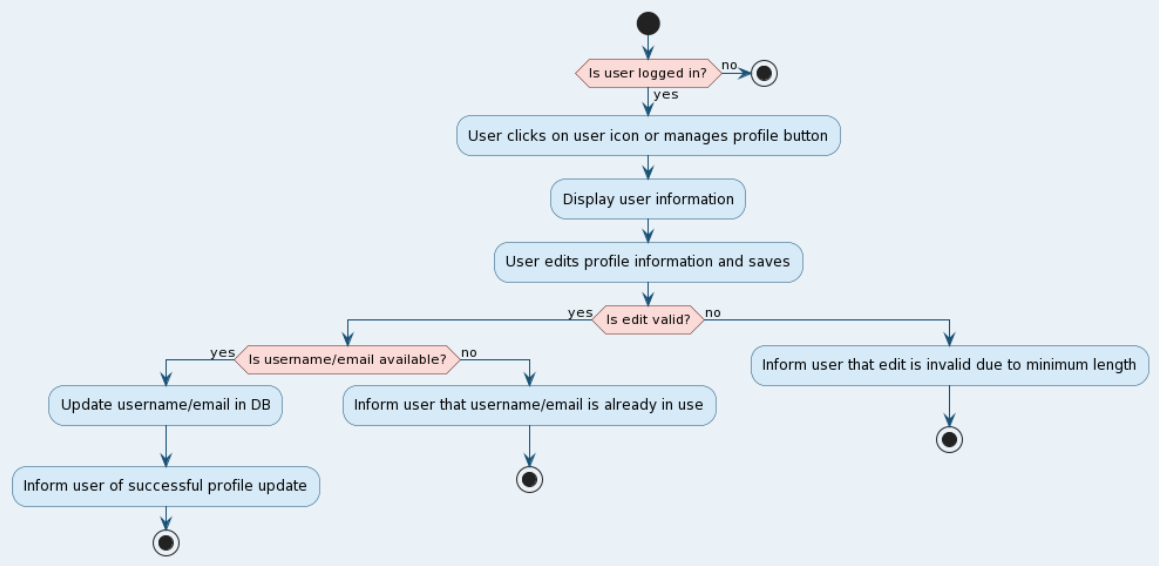

Figure 4.4.10

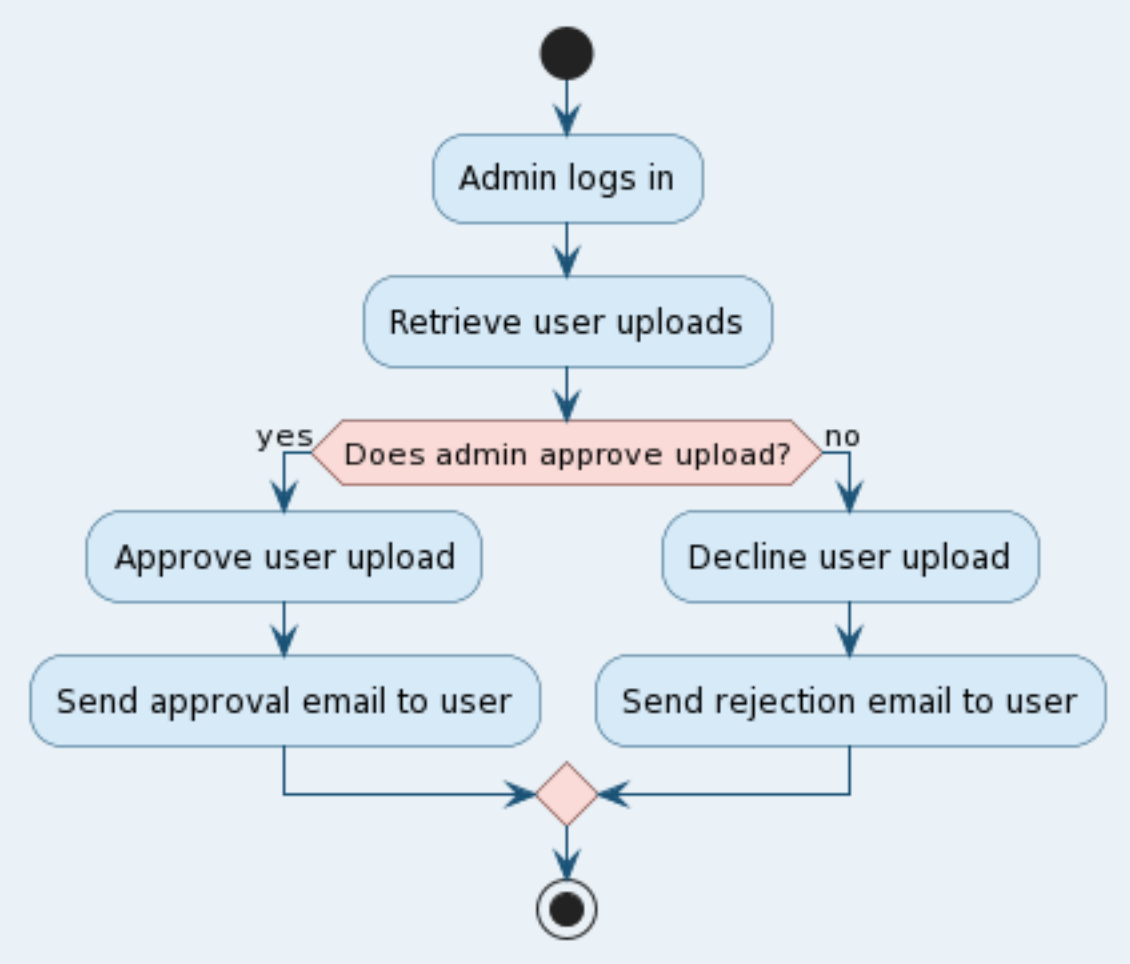
Admin logs in
Retrieve user uploads
yes
Does admin approve upload?
no
Approve user upload
Decline user upload
Send approval email to user
Send rejection email to user

Figure 4.4.11

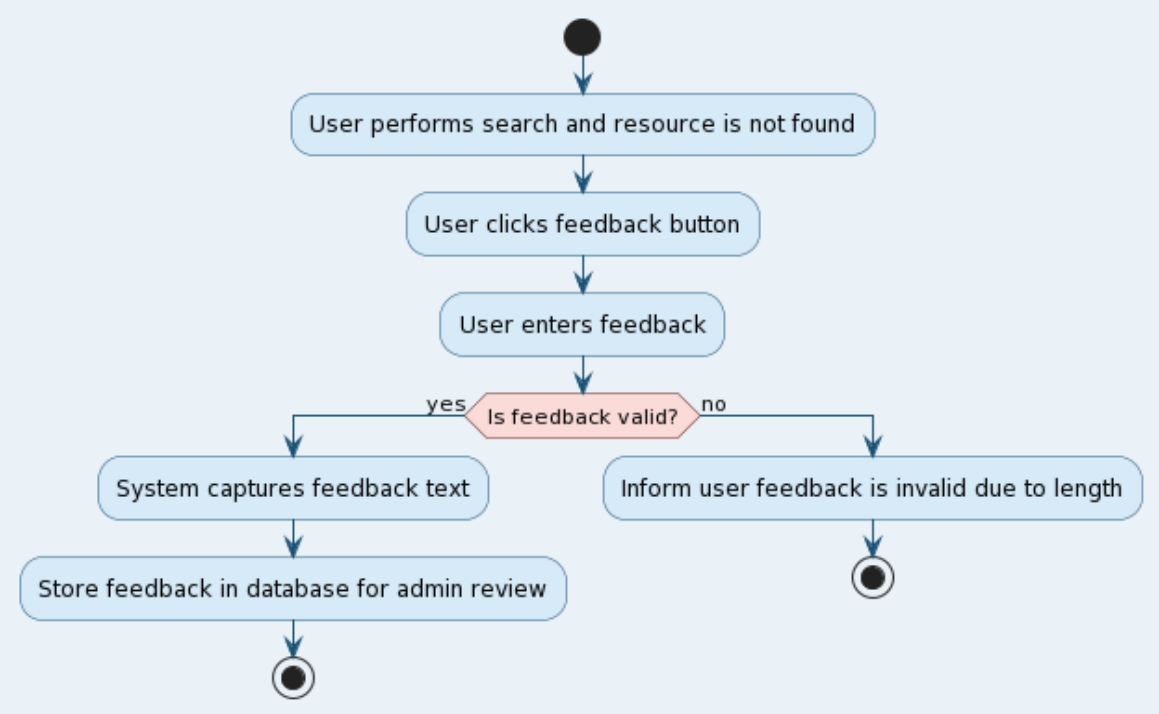
User performs search and resource is not found
User clicks feedback button
User enters feedback
Is feedback valid?
yes
no
System captures feedback text
Store feedback in database for admin review
Inform user feedback is invalid due to length

Figure 4.4.12

5. Class Diagram

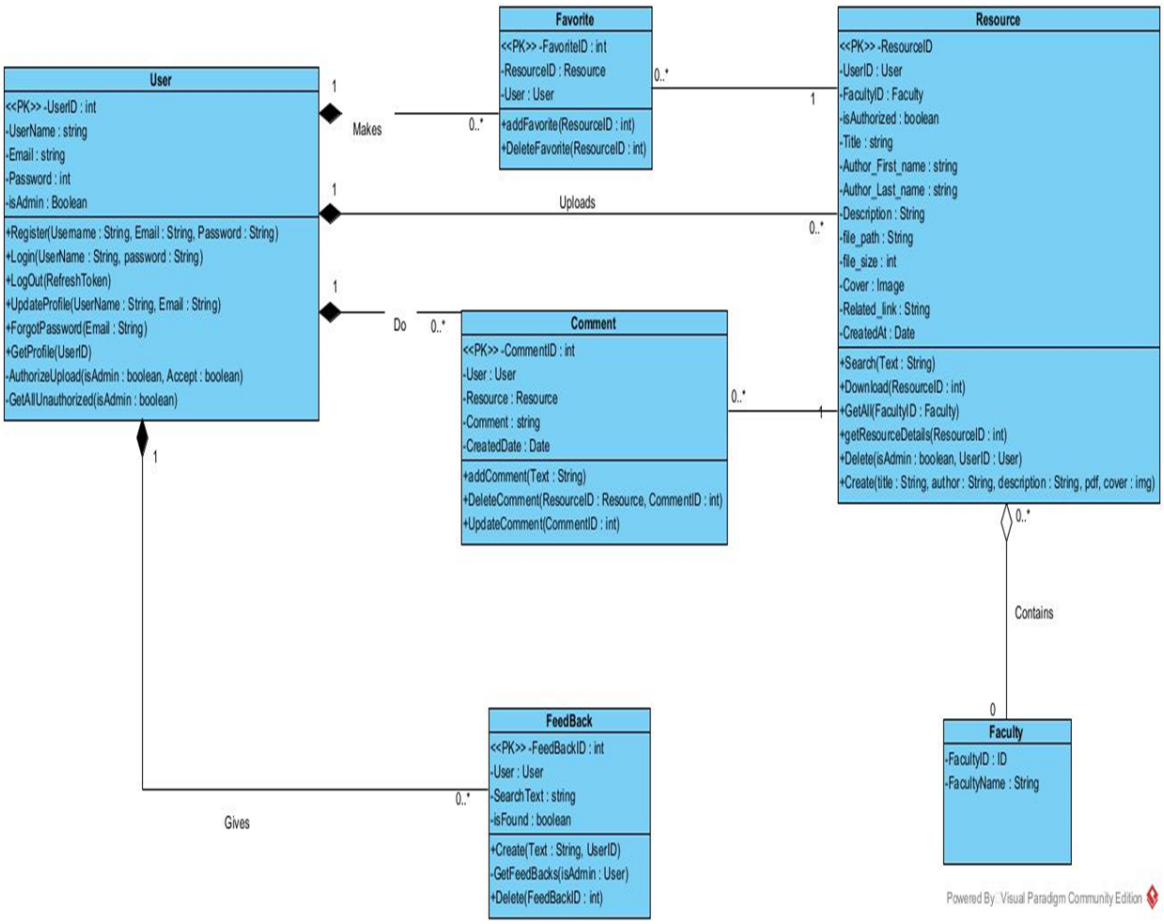

Figure 5.1

6. Entity Relationship Diagram

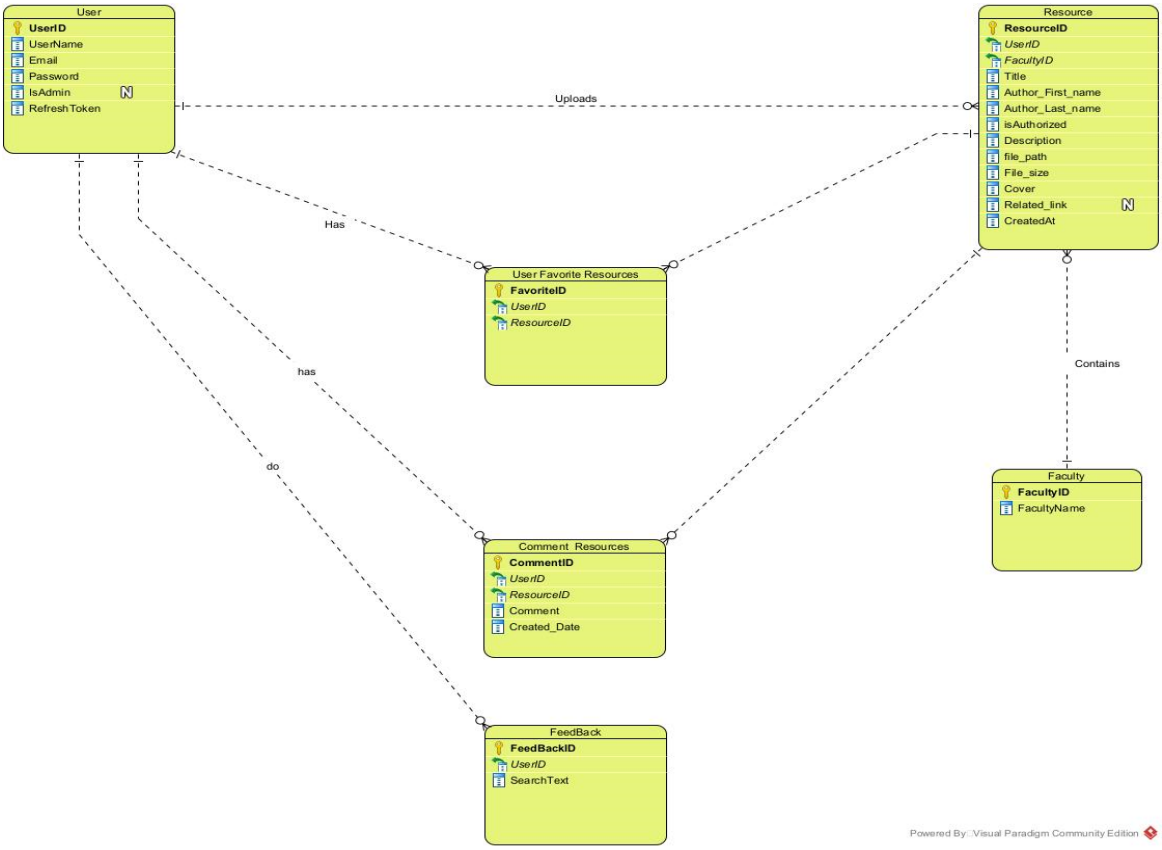

Figure 6.1