Analysis and Design of Information Systems I

MTI social club

Supervised By:

- Dr. Marwa N. Refaie
- T.A.: Omar Khaled
- T.A.: Amal Mohamed

Submitted By :

Student Name
Mostafa Ahmed Ali Ahmed G7
El-Syed Ahmed Abdalla El-Borei G2

Table of Contents

Chapter 1

INTRODUCTION

In this project we considered a social club app for our University MTi. We want to give our community (MTI students) a way to communicate each other by a spatial app ,so we made a small version of an app call **MTI social club.**

Technologies/Techniques

We used deferent tools to complect this app in a web version, starting with the Database by using **Microsoft SQL Server** to save and modify the data of our users, for the front end we used **React** program base **typescript** and of course with **HTML** and **CSS** ,for delivering the data from our Database to the front end we used **ASP API**.

Features

The user can:

- **post his post easily.**
- **give the post alike.**
- **know the user level ,Name and maybe his image.**

Functional Requirements:

Admin

- Manage posts.
- Manage users.

Normal User

- Add, edit, delete posts.
- Change image.

Non-Functional Requirements:

Performance:

The web app should be able to handle a large number of concurrent users without slowing down or crashing. It should be optimized for fast loading times and minimal latency.

Security:

The web app should be secure against unauthorized access, hacking, and data breaches. It should use encryption for sensitive data, implement user authentication and authorization mechanisms, and follow best practices for secure coding and software design.

Scalability:

The web app should be designed to scale up or down easily to accommodate changing user needs and usage patterns. It should be able to handle increased traffic and usage without compromising performance or stability.

Availability:

The web app should be available to users 24/7, with minimal downtime for maintenance or upgrades. It should have a robust disaster recovery plan in place to ensure data integrity and availability in case of system failures or disasters.

Usability:

The web app should be easy to use, intuitive, and accessible to users of all abilities. It should follow established UX and UI design principles, be responsive to different screen sizes and devices, and provide clear and concise feedback to users.

Maintainability:

The web app should be easy to maintain, upgrade, and extend over time. It should use modular and reusable code, follow established coding standards and best practices, and be well-documented with clear instructions for developers and system administrators.

Basic Data

Table Name: department

- dep_id (int)
- dep_name (varchar(15))

Table Name: users

- userID (int)
- email (varchar(30))
- password (varchar(30))
- name (varchar(20))
- birth_date (date)
- phone (varchar(11))
- address (varchar(30))
- Level (int)
- user_role (varchar(9))
- Profile_photo (image)
- gender (char(6))
- dep_id (int)

Table Name: post

- post_id (int)
- userID (int)
- post_date (date)
- text (varchar(max))
- img (image)
- likes (int)

Table Name: staff

- userID (int)
- courses (varchar(15))

Table Name: stats

- Status_id (int)
- Status_val (varchar(15))

API Perform transactions:

- Add, update, delete ,check user exists ,login user check(users).
- Add, update, delete , get all posts ,search by user id(posts).
- The API handles converting the data string ,numerical and image data to json so it can be handled and used in the front end.

Login Example:

In this example show how the API handle the login action the front sends the user id and his password then the API chick if the data are valid from the Database if it is not valid the API send empty object of user if else the API send this user data in object called user.

the code :

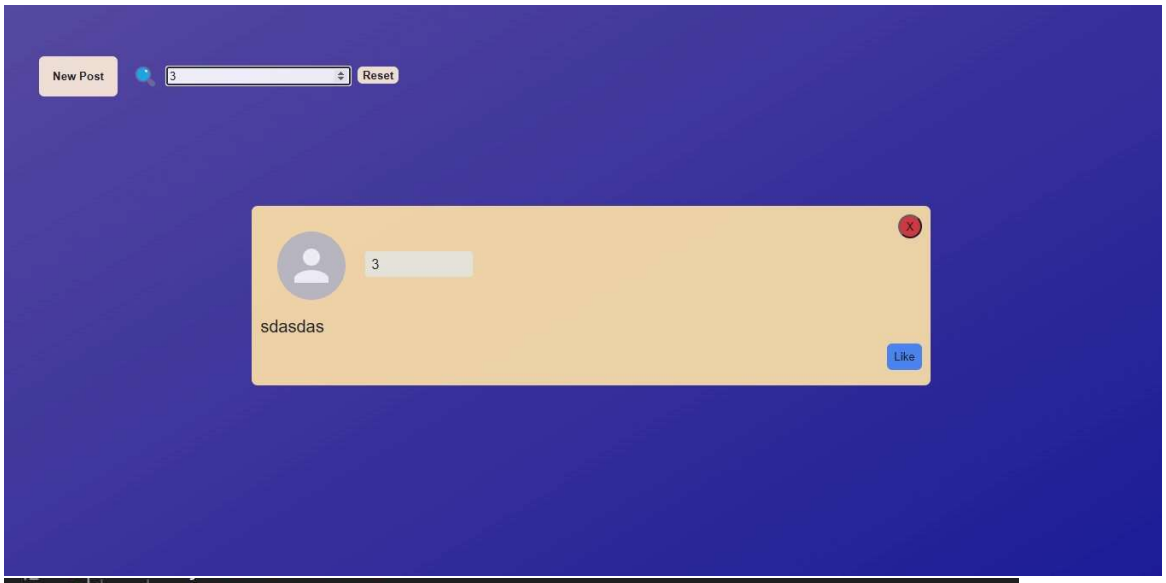
```
[HttpGet]
[Route("/login")]
0 references
public user GetLogin(int id,string pass)
{
    user userr = new user();
    SqlConnection con = new SqlConnection(sqlConnection);
    try
    {
        SqlCommand cmd = new SqlCommand($"SELECT * FROM users where userID ={id} and Password = '{pass}';", con);
        con.Open();

        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(ds);
        con.Close();

        if ((ds.Tables.Count > 0) && (ds.Tables[0].Rows.Count > 0))
        {
            return Get(id);
        }
    }
    catch (Exception)
    {
        con.Close();
    }
    return userr;
}
```

Report

Search for posts by user ID:

in this example the API receive the user id from the front end and search for all posts from this user and send it back to the front

The used quarry for the search :
“SELECT * FROM post where userID = {id}”




```
[Route("getpostID")]
[HttpGet]
0 references
public JsonResult GetPostID(int id)
{

    String query = $"SELECT * FROM post where userID = {id}";
    SqlConnection conn = new SqlConnection(sqlConnection);
    SqlCommand cmmd = new SqlCommand(query, conn);
    conn.Open();
    List<post> deps = new List<post>();
    SqlDataReader rdr = cmmd.ExecuteReader();
    if (rdr.HasRows)
    {
        post post;
        while (rdr.Read())
        {
            post = new post();
            post.postID = (int)rdr["post_id"];
            post.userID = (int)rdr["userID"];
            post.text = rdr["text"].ToString();
            post.postDate = (DateTime)rdr["post_date"];
            post.likes = (int)rdr["likes"];
            deps.Add(post);
        }
    }
    conn.Close();


    return new JsonResult(deps);

```

Chapter 2 : Analyses

Task Sheet:

Task 1: Research and Analysis

- Conduct research on best practices for documenting software projects
- Analyze the project requirements and determine the necessary documentation components
- Create a document outline and set project milestones

Task 2: Introduction and Technologies Overview

- Write an introduction to the project and its objectives
- Provide an overview of the technologies used in the project, including React, Microsoft SQL Server, and ASP API
- Describe the basic data structures used in the app (e.g. the user, post, and department tables)

Task 3: Functional and Non-Functional Requirements

- Document the functional requirements for the app, including user and admin roles, post management, and user authentication
- Document the non-functional requirements for the app, including performance, security, scalability, availability, usability, and maintainability

Task 4: Sitemap, Use Case, and Activity Diagrams

- Create a sitemap to illustrate the hierarchy and organization of the web app
- Develop use cases for the app to describe the interactions between users and the system
- Create activity diagrams to visualize the flow of user interactions within the app

Task 5: Test Cases and Sequence Diagrams

- Develop test cases to ensure the functionality and accuracy of the app
- Create sequence diagrams to illustrate the interactions between objects and components in the app

Task 6: EER Diagram, Mapping, and Skema

- Develop an EER diagram to illustrate the relationships between data tables in the app
- Map out the data flow and relationships between the API, database, and front-end components of the app
- Create a Skema to illustrate the database schema used in the app

Task 7: Database Design and Integration

- In this task, we will design and integrate the database for the MTI Social Club web app.
- We will start by creating the necessary tables and relationships in Microsoft SQL Server, using SQL commands to define the structure and constraints of the database.
- We will then integrate the database with the back-end components of the app using ASP.NET Web API, ensuring that data is stored and retrieved accurately and efficiently.

Task 8: Input and Output Interface Design

- Create mockups of the input and output interfaces for the app, including forms, tables, and charts
- Incorporate feedback from the instructor and revise the interface designs as necessary

Task 9: Issue Query Subsystems, Perform Transactions

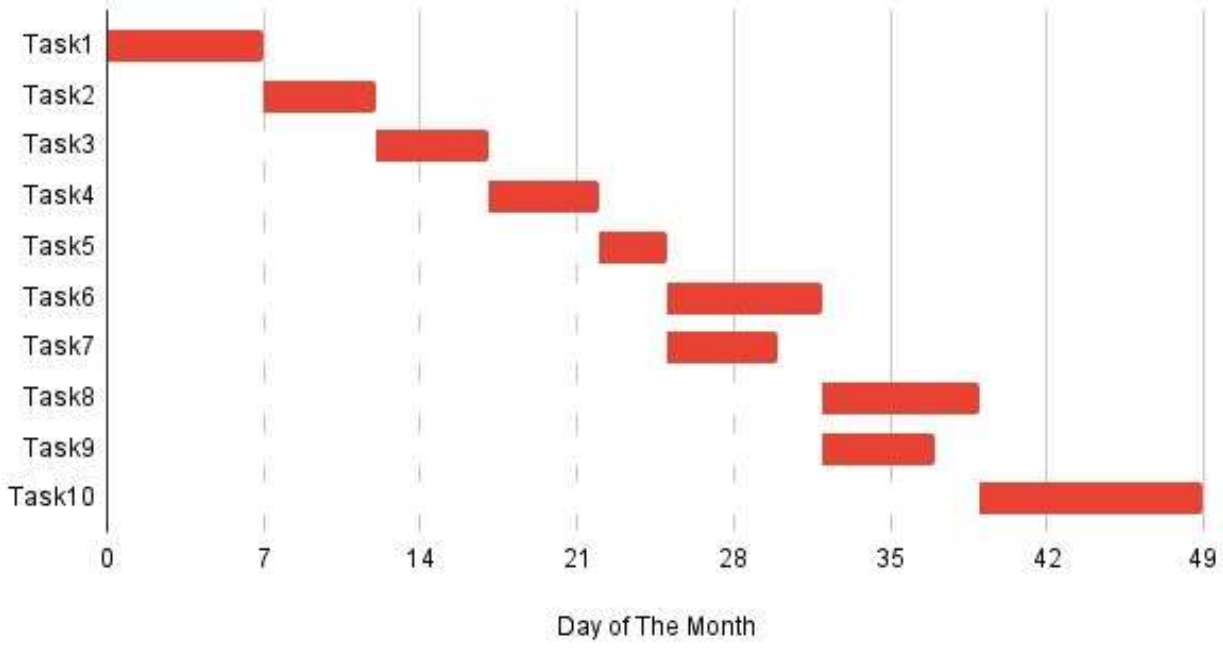
- Document the app's capabilities for issuing and processing user queries and requests
- Document the app's ability to perform transactions such

Task 10: Finalize and Review

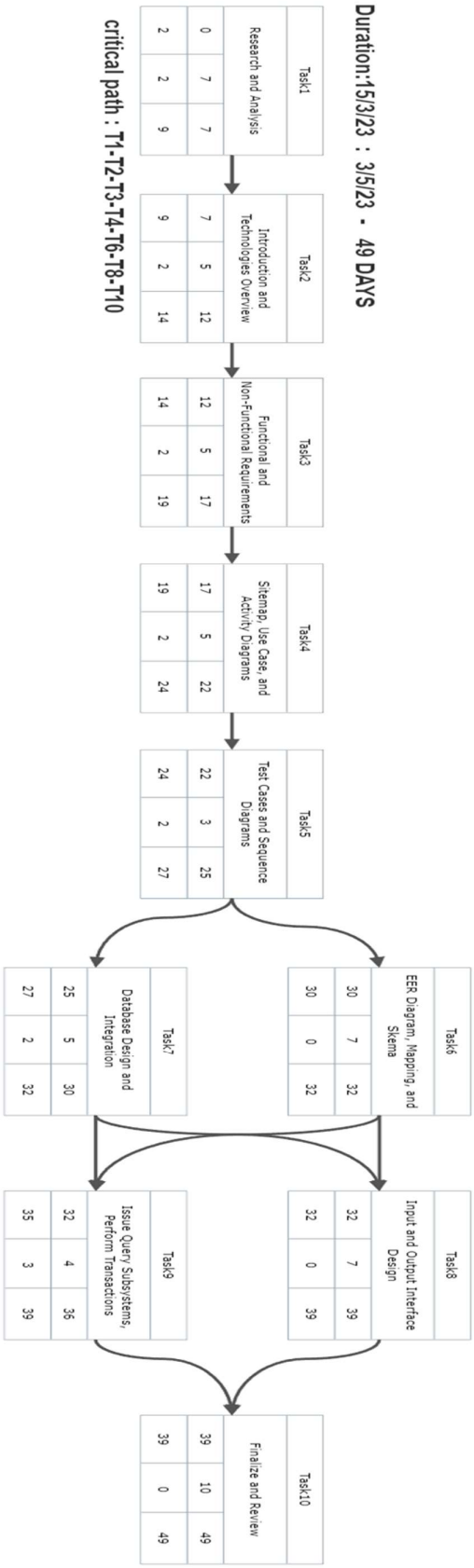
- Compile all documentation components into a comprehensive project report
- Review and revise the report for clarity and accuracy
- Submit the final report to the instructor for evaluation

Gantt Chart

TASK	START DATE	END DATE	Start on Day	DURATION(Days)
Task1	3/15/2023	3/22/2023	0	7
Task2	3/22/2023	3/27/2023	7	5
Task3	3/27/2023	4/1/2023	12	5
Task4	4/1/2023	4/6/2023	17	5
Task5	4/6/2023	4/9/2023	22	3
Task6	4/9/2023	4/16/2023	25	7
Task7	4/9/2023	4/14/2023	25	5
Task8	4/16/2023	4/23/2023	32	7
Task9	4/16/2023	4/21/2023	32	5
Task10	4/23/2023	5/3/2023	39	10

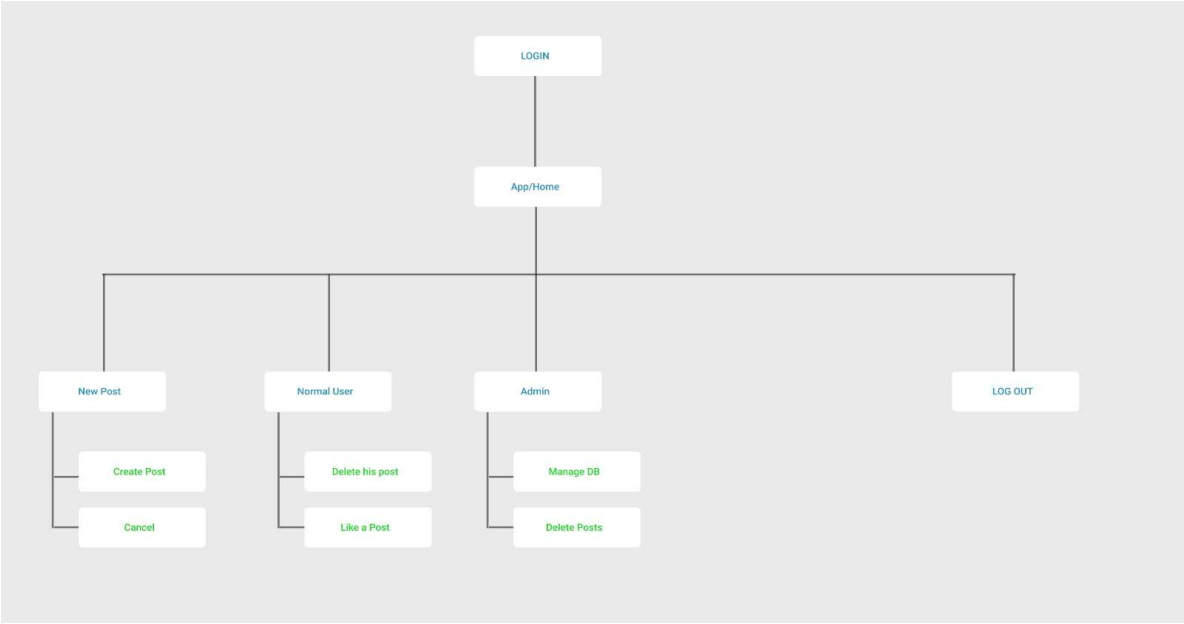


Network Diagram



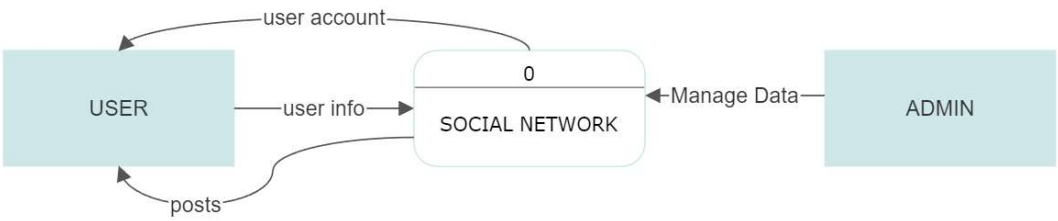
Sitemap

A sitemap is a list of all the pages on a website, which helps search engines and website visitors understand the structure and content of the site. It can improve the visibility of the website in search engine results and make it easier for visitors to navigate the site.

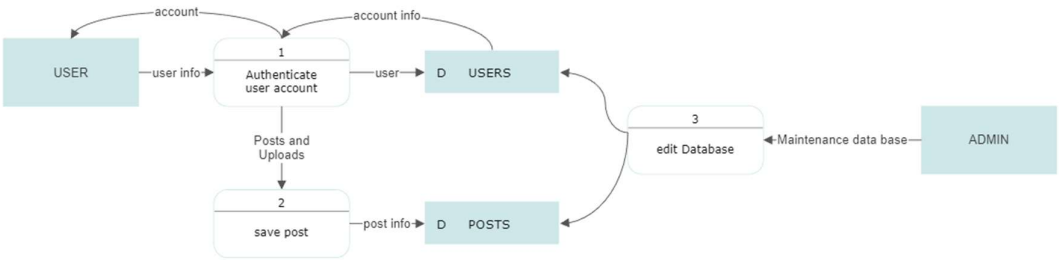


System Analysis

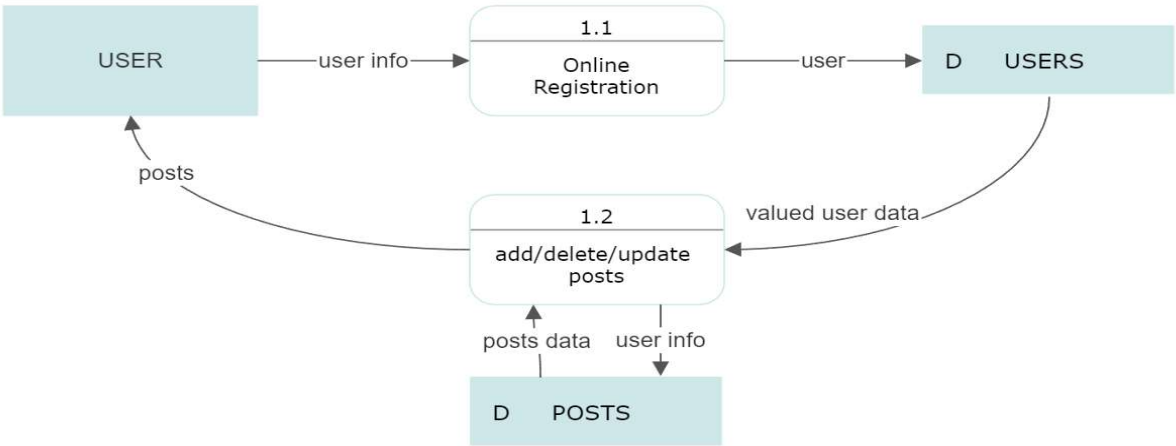
Context Diagram:



Data Flow Diagram:

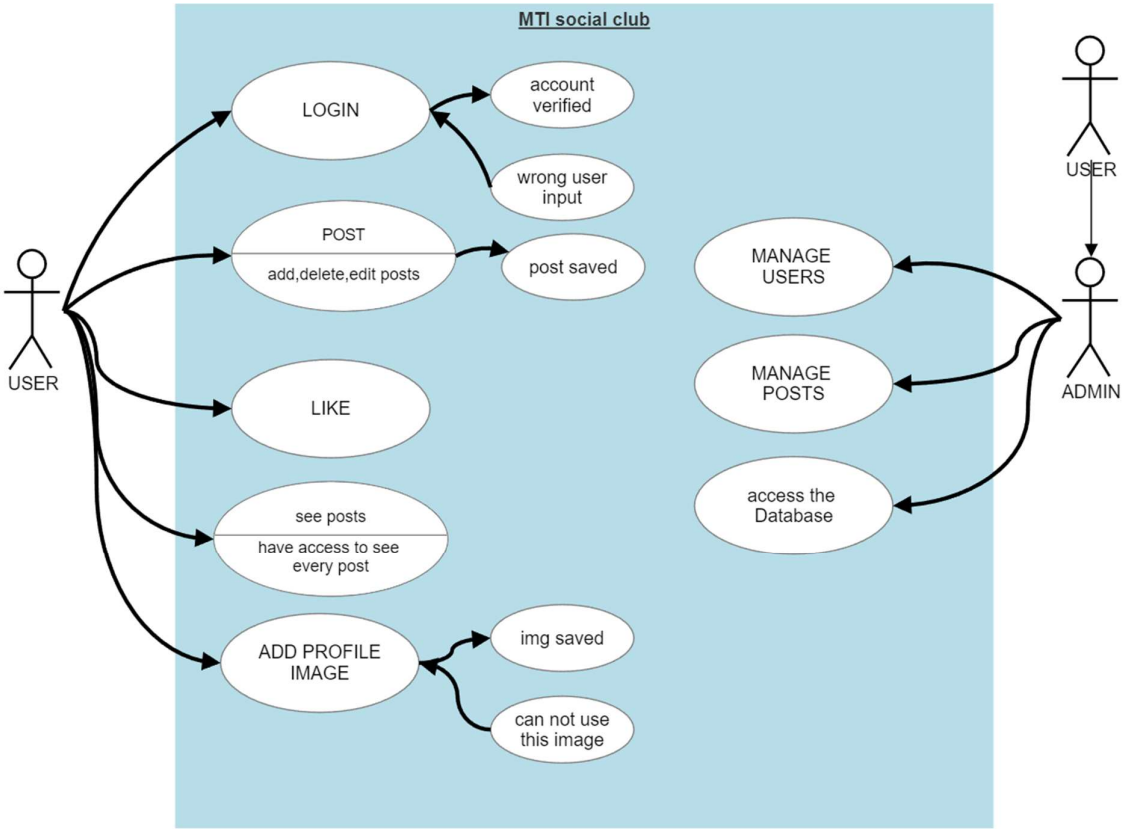


Level-1 DFD:



use case

A use case depicts a set of activities performed to produce some output result. Each use case describes how an external user triggers an event to which the system must respond.



First Actor: User

- Can Login.
- Can add, edit and see posts.
- Can like other posts.
- Can edit his image.

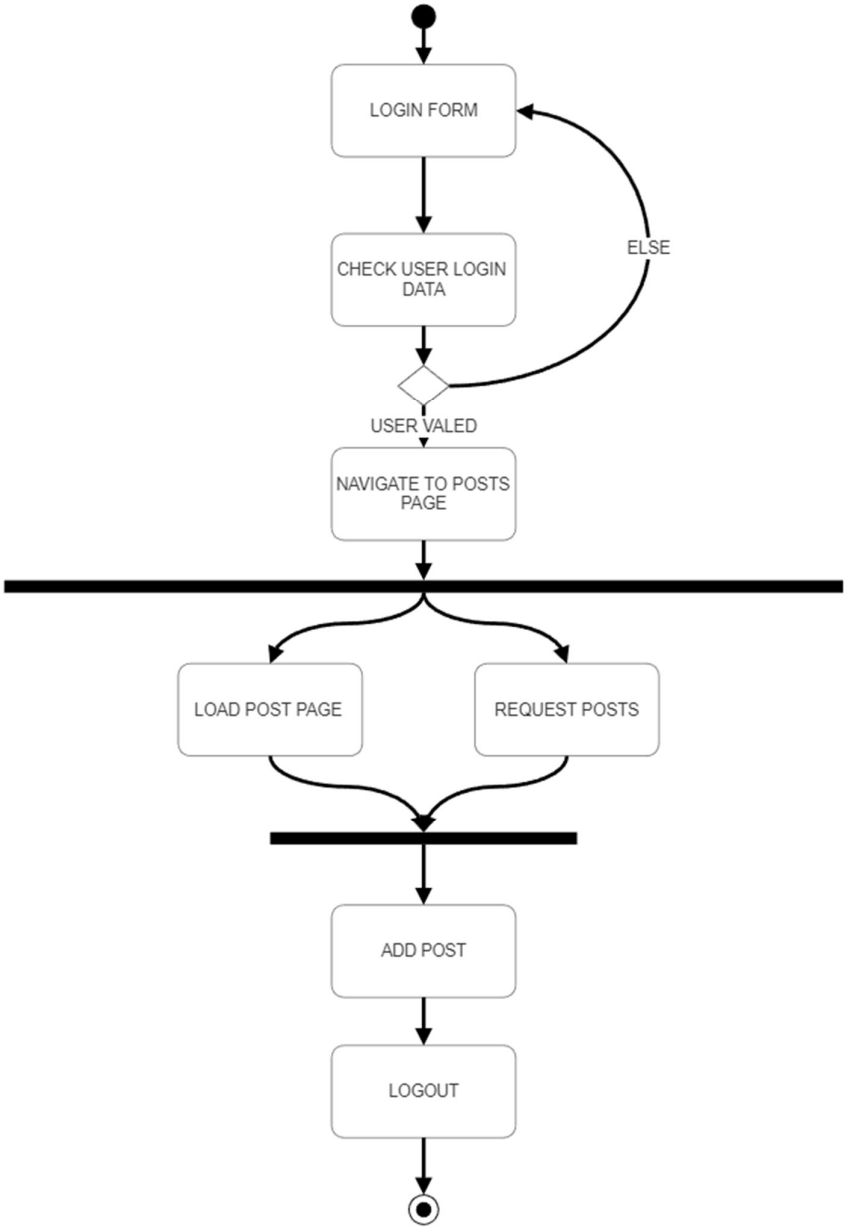
Second Actor: Admin

- Can do anything like a user.
- Can remove images and posts.

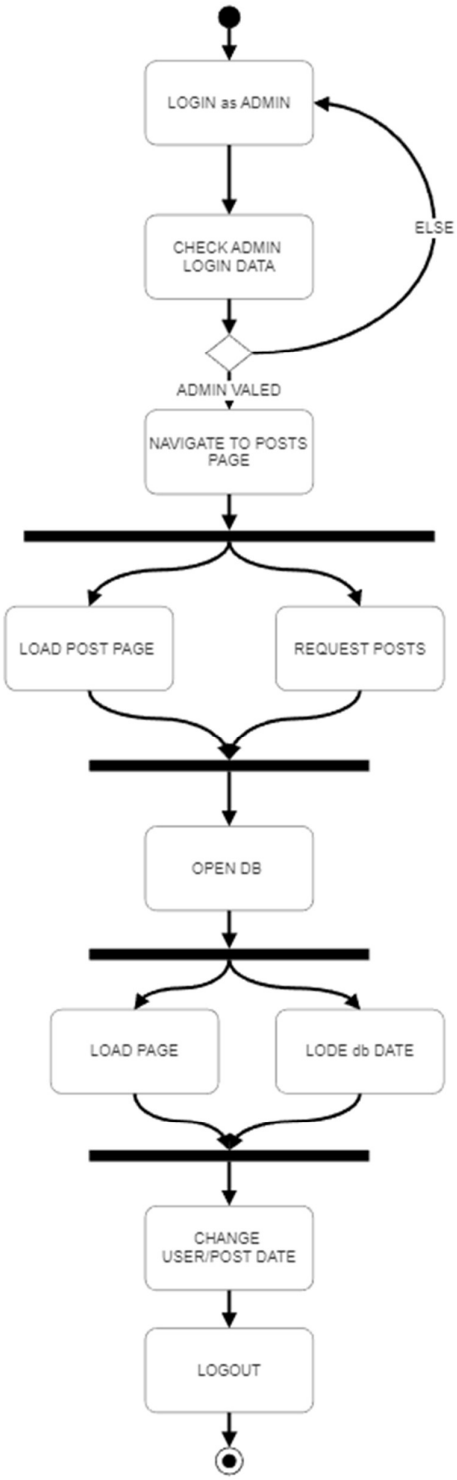
Activity Diagram

The activity diagram gives a wide look at the system, the activity diagram is a tool that can be offered to non-technical people to understand the system well.

Main activity diagram of the User:



Main activity diagram of the admin changing Data



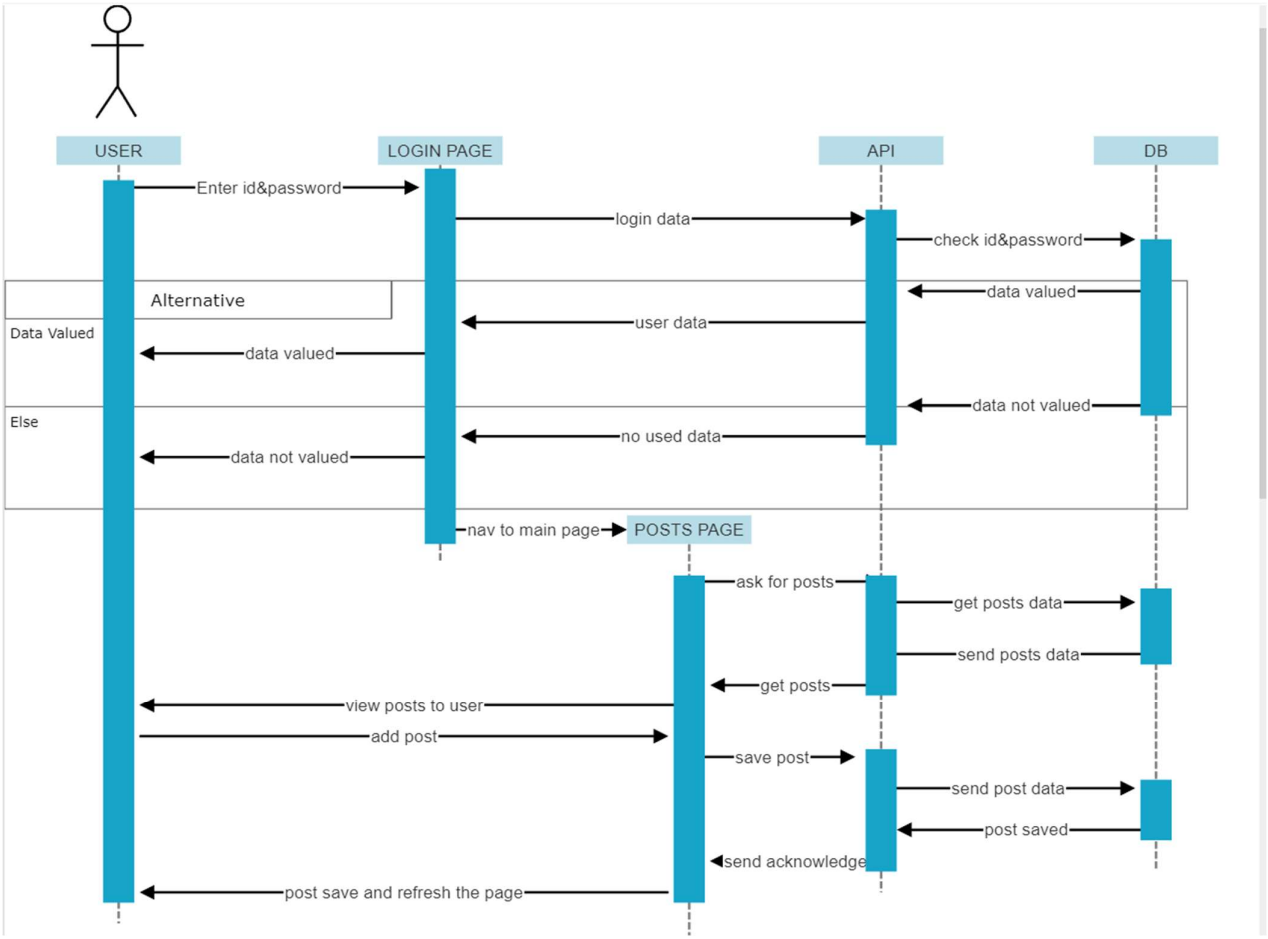
Test Case

Test.no	process	steps	expected	actual	stats	note
1	Add post	1-login 2-click post 3-save post	The post show in the front	The post show in the front	success	
2	Upload user image	1-login 2-click upload 3-chose image	The user image changed	The user image changed	success	
3	See the DB	1-login as admin 2-click on DB button	The data appeared	The data appeared	success	
4	Add image to the post	1- login 2-add post 3-select image 4-uplode	The image showed in the post	No image showed	failed	In progress to fix it
5	Delete post	1-Login as admin 2-choose post to delete 3-delete the post	The post removes from the DB and from the UI	The post removes from the DB and from the UI	success	

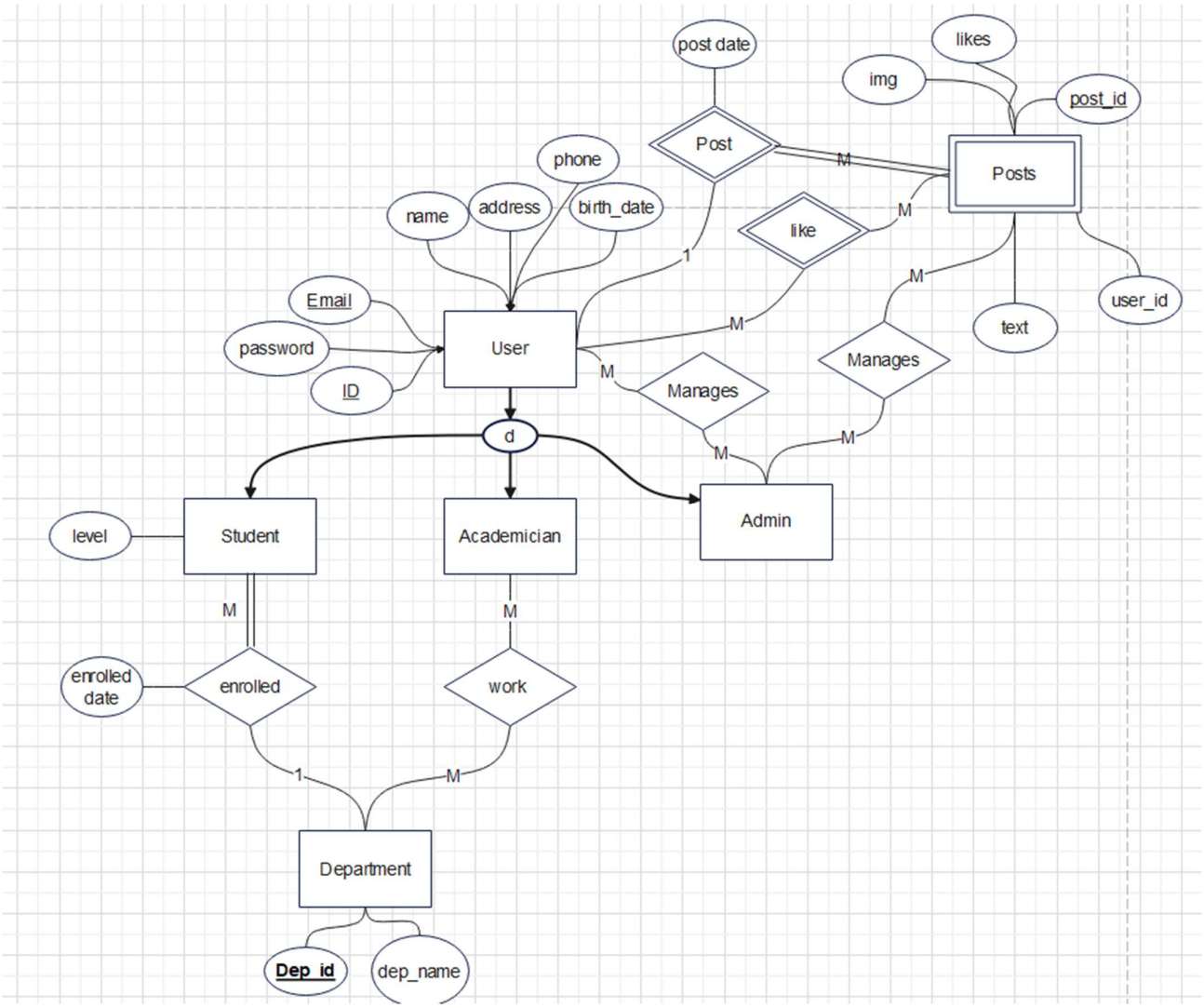
Sequence Diagrams

The sequence diagram shows interactions in the system between objects put together respecting the sequence. It shows the message exchanged between the objects and what kind of interactions are done. In the MTI social club application, we look at the exchange between the actors and the system as well as the database and the API.

Login as User:



EER Digram:


post date
likes
img
post_id
Post
Posts
phone
name
address
birth_date
like
Email
password
ID
User
text
user_id
Manages
Manages
d
Admin
level
Student
Academician
enrolled date
enrolled
work
Department
Dep_id
dep_name

Mapping :

Department

(dep_id , dep_name)

Users

(userID , email ,password , name ,birth_date ,phone ,address , Profile_photo, gender)

Student

(userID , enrolleddate , Level , dep_id)

Academician

(userID , dep_id)

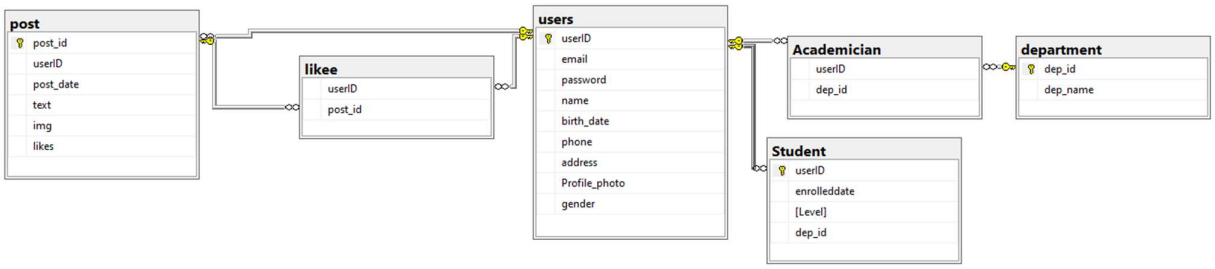
post

(Post_id , userID , post_date , text , img)

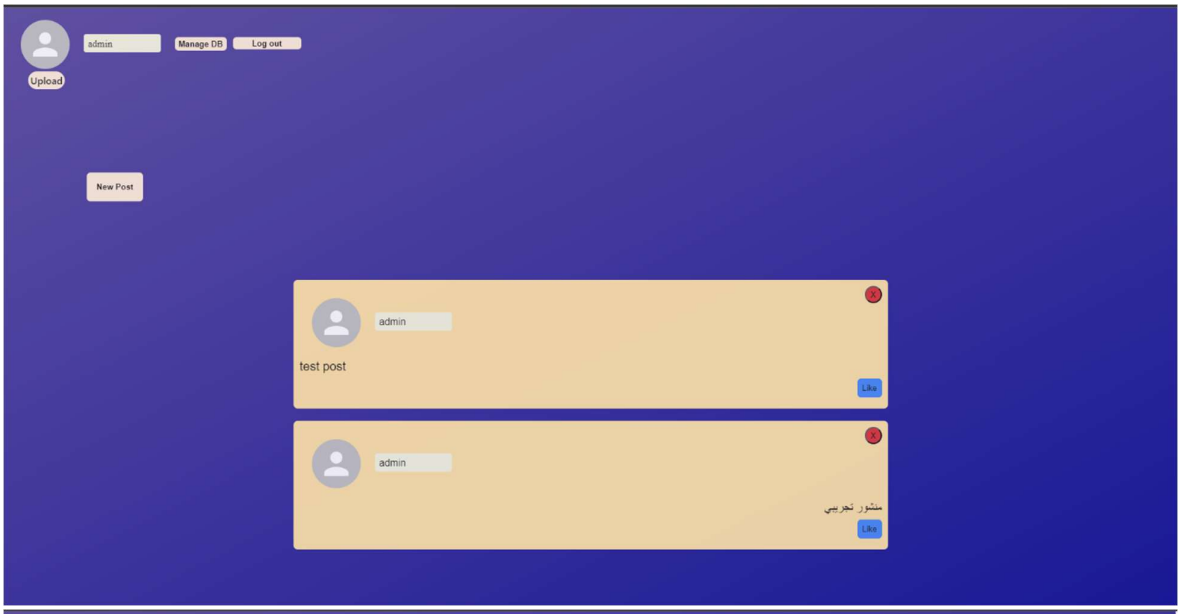
Like

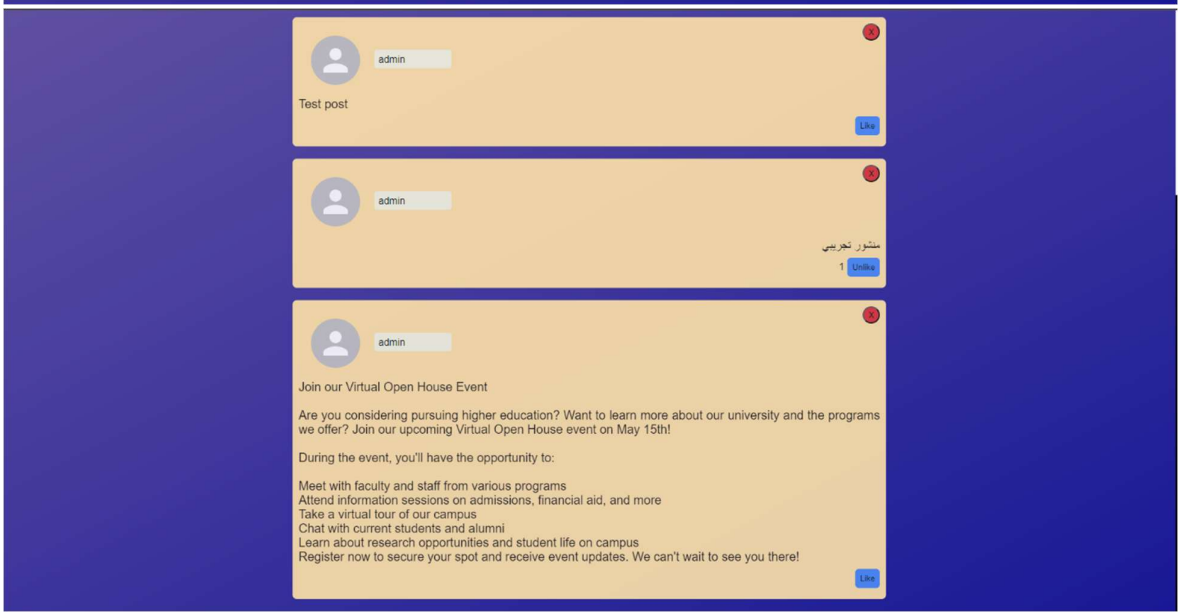
(userID, Post_id)

Skema:



Input / Output Interface Design


admin
Manage DB
Log out
Upload
New Post
admin
test post
Like
admin
منشور تجريبي
Like


admin
Test post
Like
admin
منشور تجريبي
1
Unlike
admin
Join our Virtual Open House Event
Are you considering pursuing higher education? Want to learn more about our university and the programs we offer? Join our upcoming Virtual Open House event on May 15th!
During the event, you'll have the opportunity to:
Meet with faculty and staff from various programs
Attend information sessions on admissions, financial aid, and more
Take a virtual tour of our campus
Chat with current students and alumni
Learn about research opportunities and student life on campus
Register now to secure your spot and receive event updates. We can't wait to see you there!
Like