Software Engineering Software Requirements Specification (SRS) Document

Prepared by :
1-Anas Hesham Mohamed.
2-Toka Abdelgawad Abdallah.
3-Hussien Mostafa Mahmoud.
4-Habiba Emad Elden Abdallah.
5-Norhan Ahmed Saber.

➢ **Contents**

1. Introduction

1.1   Documentaion Purpose

The purpose of this document is to provide a detailed and complete specification of Social media platform for the Faculty of Engineering, Helwan University , and describe and define the requirements of the project and to spell out the system's functionality and its constrains.

1.2   Target Users

This document is created based on the requirments analysis that had been done by the system administrators and the requirments gathered from different developers and system contributers to make an application that help them over come the roadblocks they faced during their learning journey.

1.3 Problem definition

Nowadays individuals of the same community are facing several problems in communicating and sharing data together ; so we are in a need for a system that connects individuals of the same comunity together.

1.4 Scope of the problem

We are looking forward to facilitate the communication between people whether of the same community or not with a targeted social platform avoiding all other similar systems' disadvantages.

1.5 Similar systems

- Microsoft Teams
  Disadvantages :
  - It has no feeds.
  -Can't share data with others (there is no profile for each individual).
-LinkedIn
  Disadvantages :
  - It doesn't have Feature of online meeting .
  - UI of LinkedIn is complex.

2. User Requirements

1.1 Functional Requirements

1- User Management:
It is the function of dealing with users and perform all the required functions on them like the (CRUD) operations and gives the ability of connecting 2 users togeather by creating a friendship between them

2-Authentication and Authorization:
- User authentication is the process of verifying the identity of a user when that user logs in to the system.

- Authorization is the process of granting or denying access to the system resources and functionality based on the polices assigned to the authenticated user role, it mostly happens after Authentatication.

3- Creating Posts:
The system allows users to create their own posts and share it on their profile or in a certain Community (Group), the post can have custom text and **attached Files.**

3-Uploading Files:

Gives the ability for users to share /add files on their posts to be avaliable on their profile, friends' feeds or groups, the file can have any type of extension and can be downloaded later by the post viewer.

4-Creating communities:

The system allows managers to create communites (groups) and add different people to join this group and be able to post on it and upload their files

1.2 Non Functional Requirements:

● Performance:

System should be able to handle all the users capacity simultaneously including all the operations that users can make.

● Maintainability:

The system is developed using framework and technologies that have a wide community, which makes it very maintainable.

● Availability:

The system is avaliable 24 hours.

3.System Users

1- **Adminstrator**: An admin system user can create system users, add accounts, assign permissions and more.

2- **Groups Managers** : they decide what information is approved or removed according to standards and guidelines so the manages groups media and members.

3- **Regular system user**: Regular system users can only access the assets they have permission for example for that they can join groups- share media - connect with others - view other's profile.

5. Software Requirement and Specifications

In order for the system to have all the user functional requirements, it had to implement this services :

1- User Management Services

It is the ability of managing users and act on them in a way where you can do the following operations:
- Create User (Sign Up)
- Delete User
- View User
- View All Users
- Update User
- View All Friends
- View All Groups
- Add Friend
- Remove Friend
- Mute Friend
- Add Post in Group

2-Posts Services:

The system allows users to create posts and manage it using the following operations (functions) :
- Create Post
- Delete Post
- Edit Post
- Upload Files to Post
- View User's all Posts
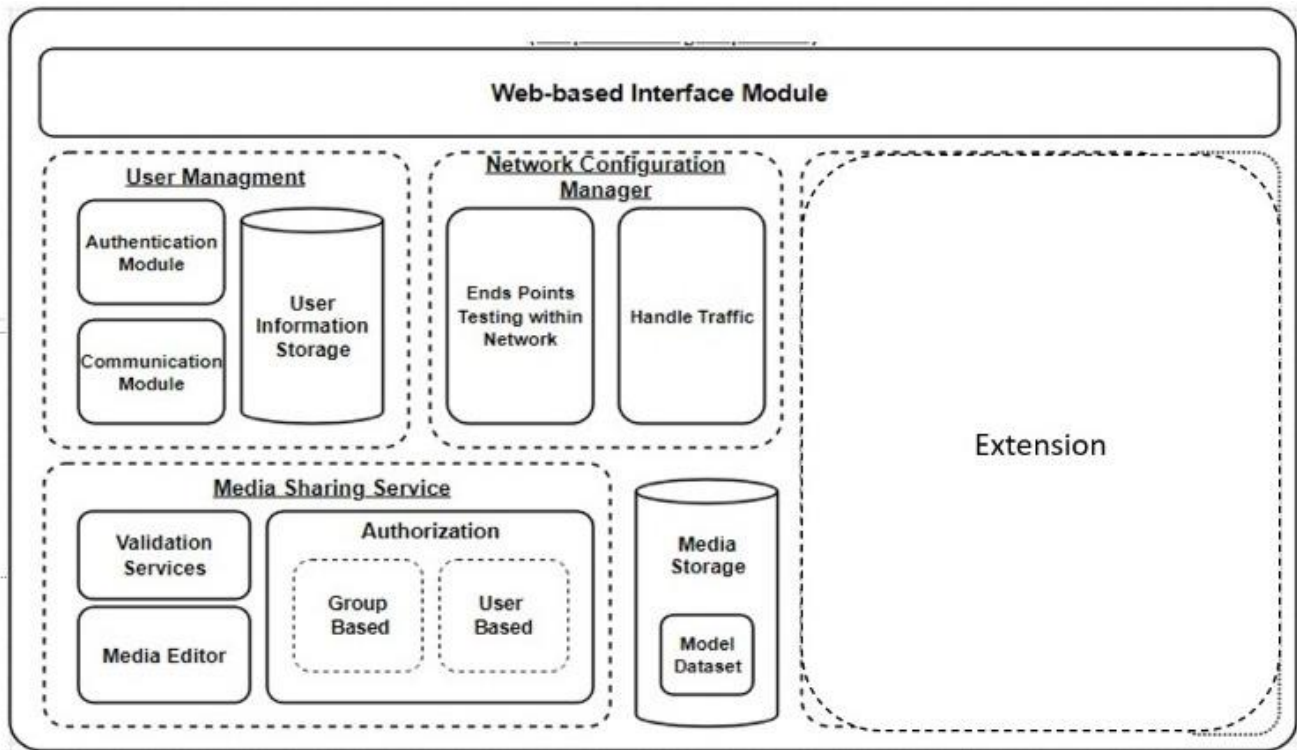- View Post's all Files
- Download Post's Files

3-Groups Services:

The system allows managers to create groups and add members to it in order to have a community and get users to post in it, the groups have the following operatoins (functions):

- Create Group
- Delete Group
- View All Groups
- View Group
- Add Member to Group
- Remove Member from Group
- View all Group's Members
- View all Group's Posts
- View Member's Posts (posted in the Group)
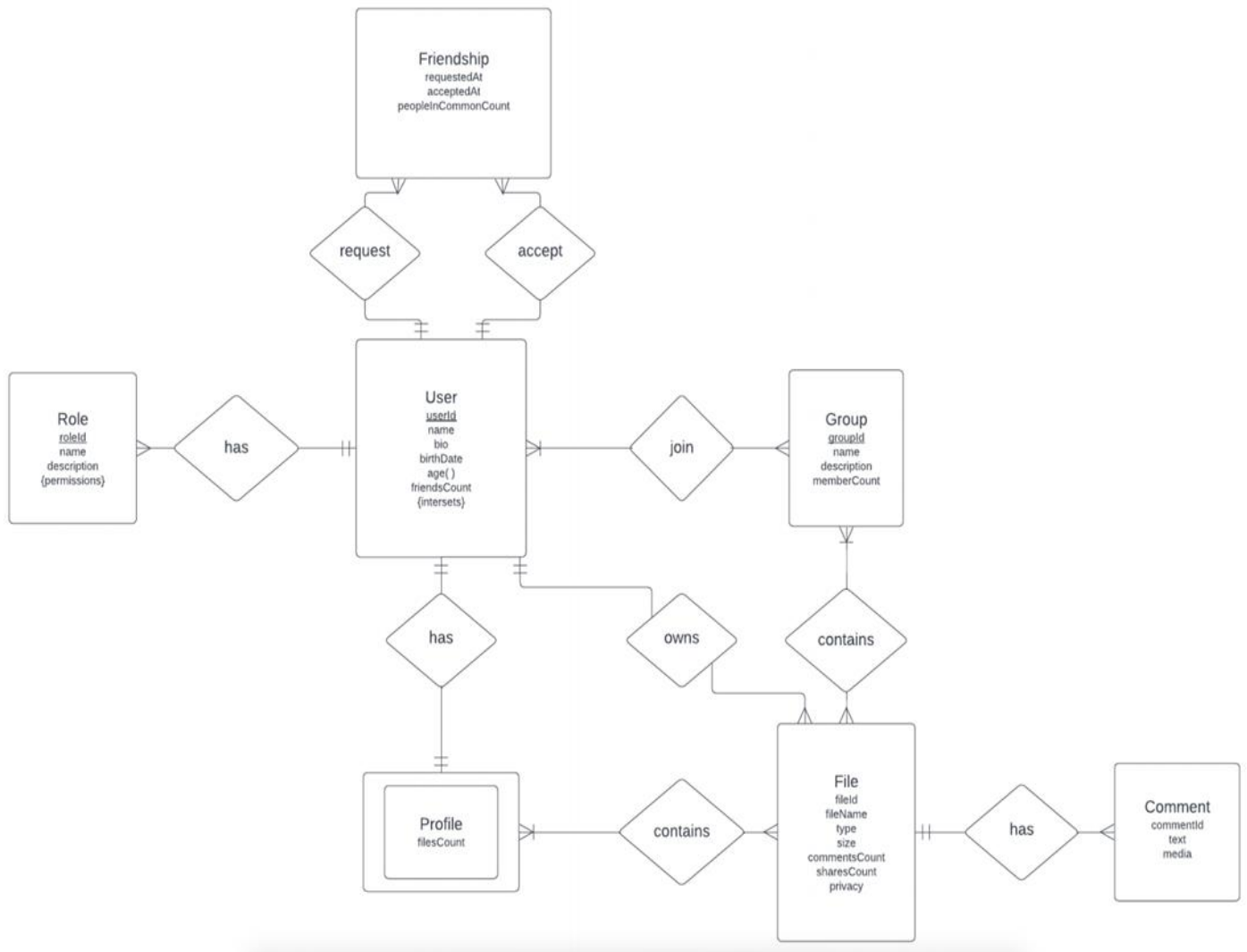- Remove Post from Group
- Mute Member

5. System Modeling

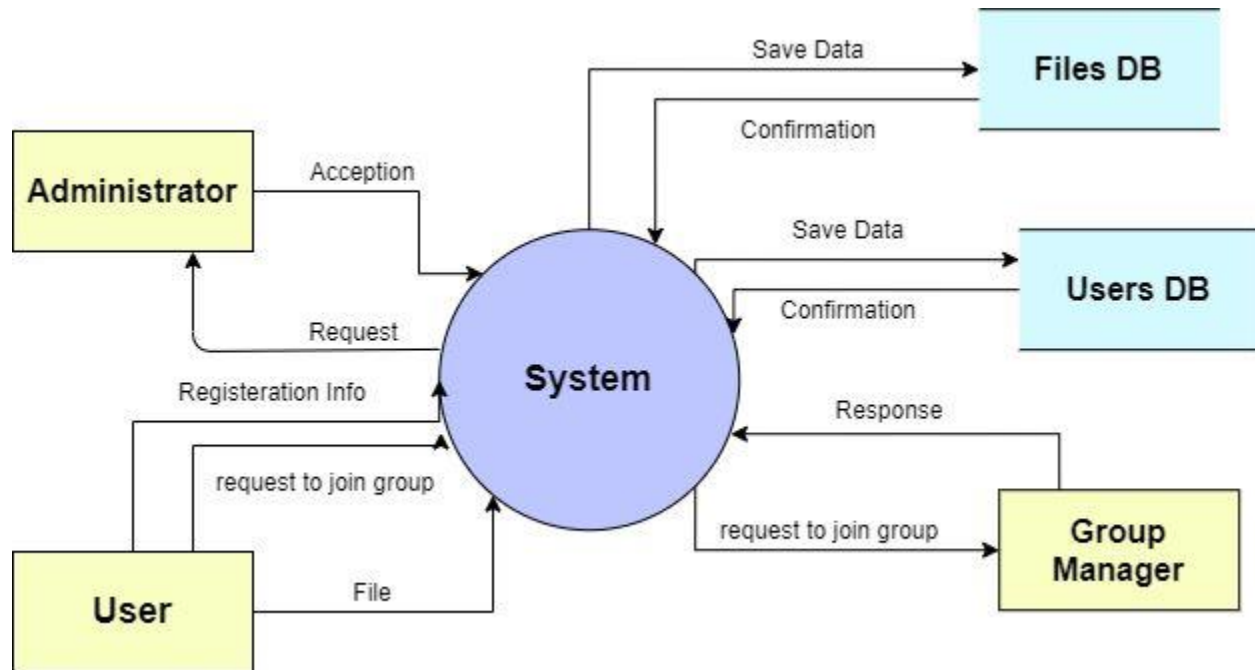## 1- The System Modular Archeticture

Figure(1): The whole system primiry modular archeticture

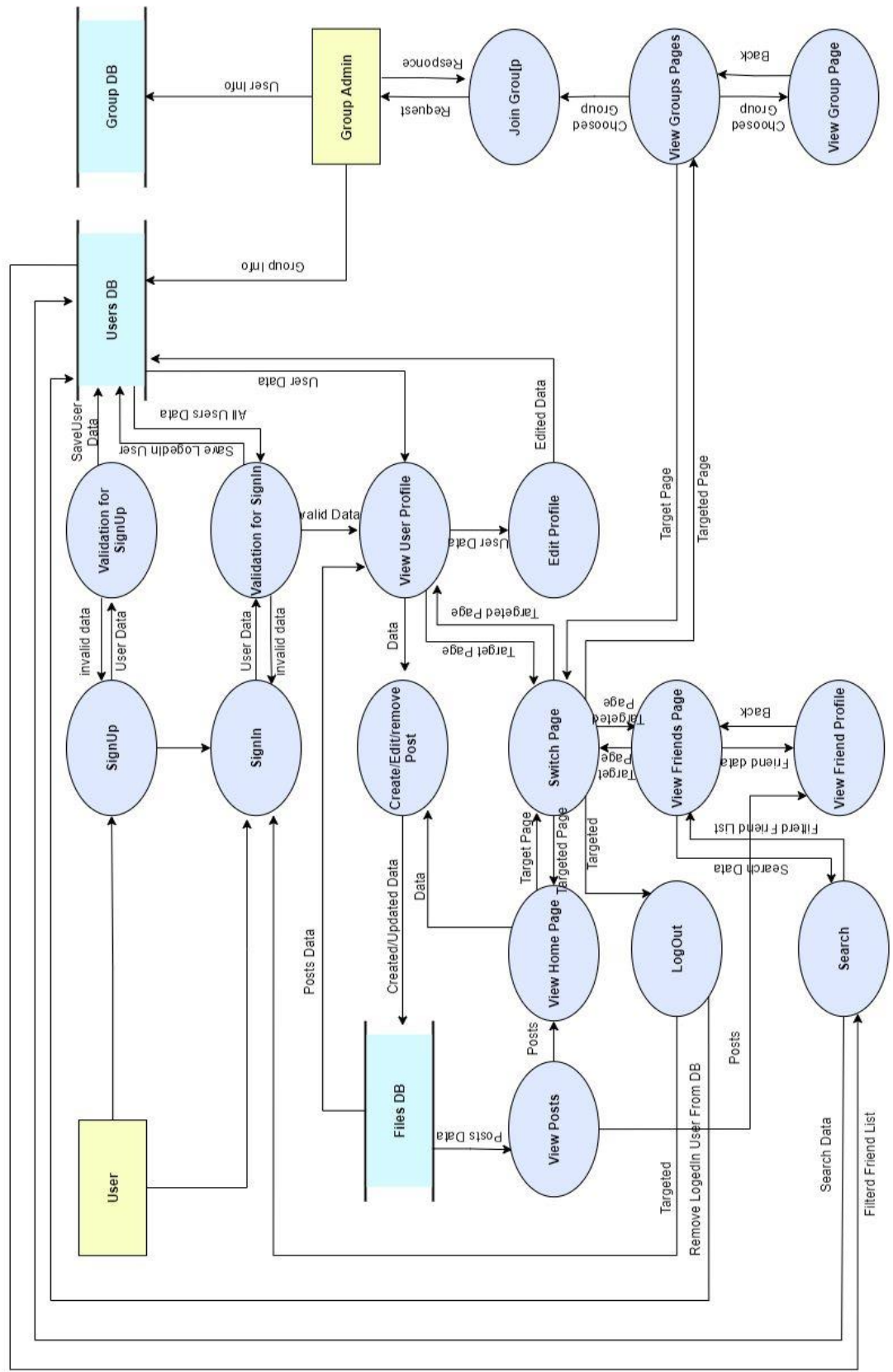# 2- Entity Relationship Diagram:

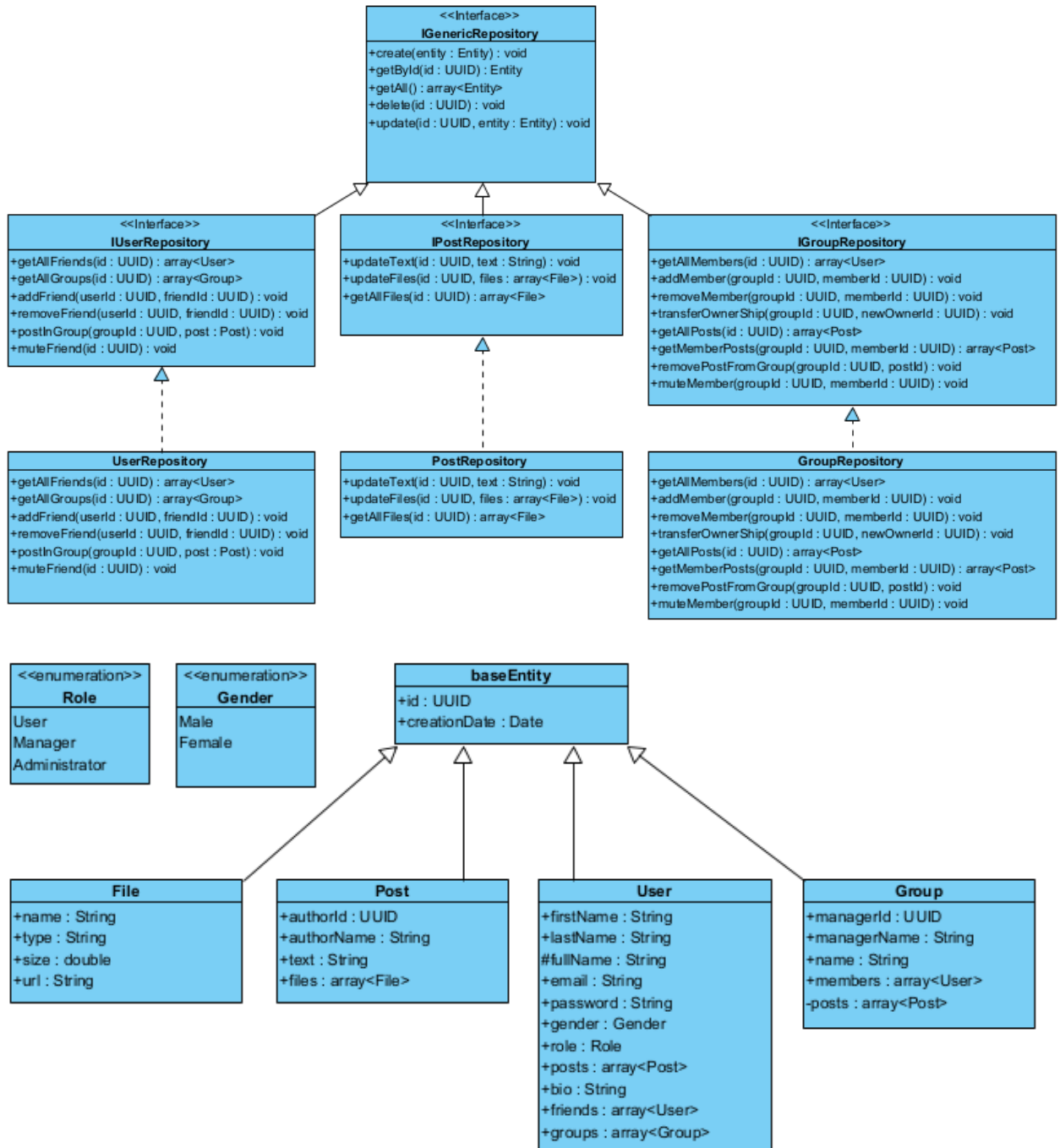Figure(2): The Entity Relationship Diagram (ERD)

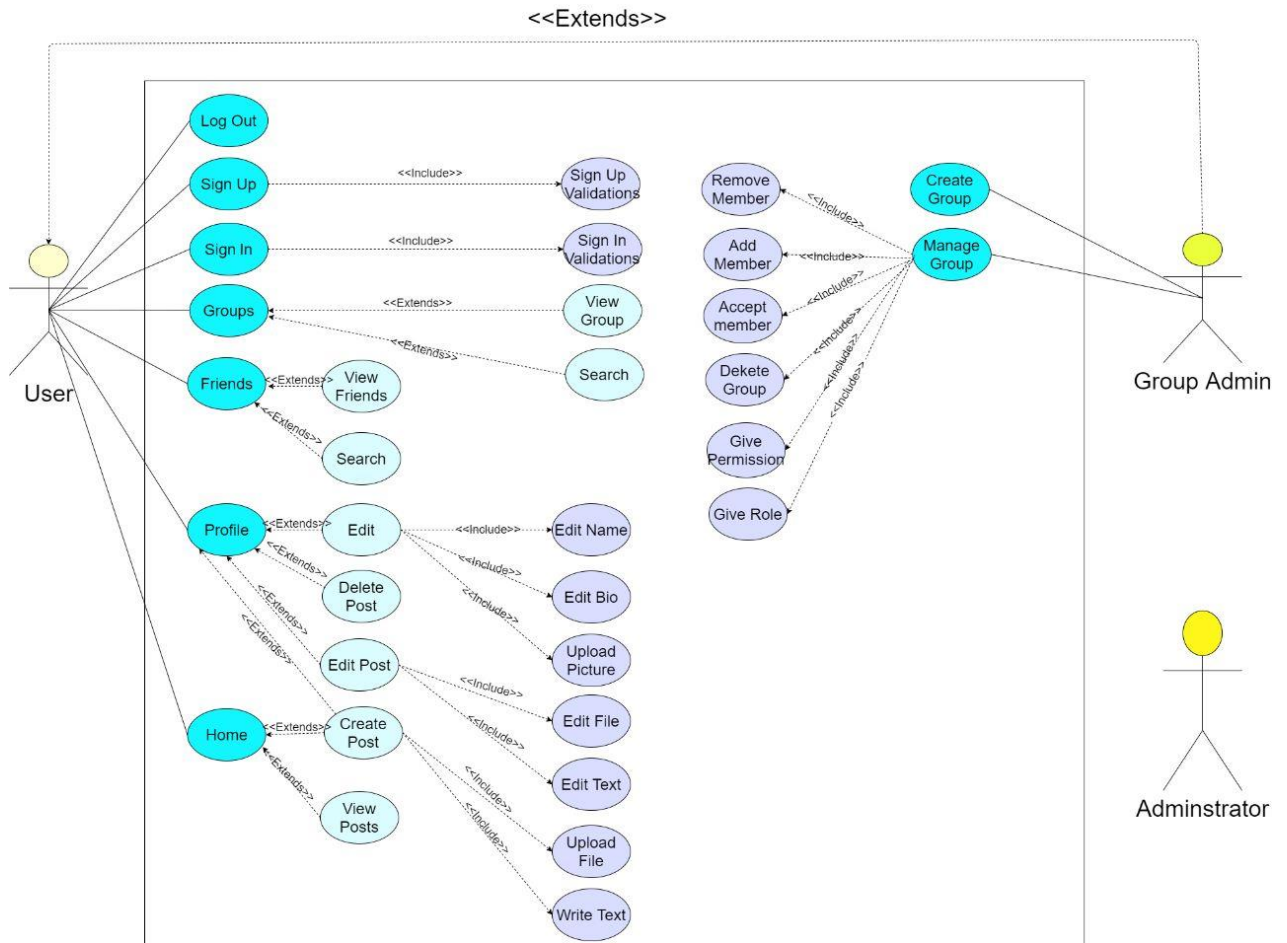# 3- Data Flow Diagram:

Figure(3): Data Flow Diagram Level 0
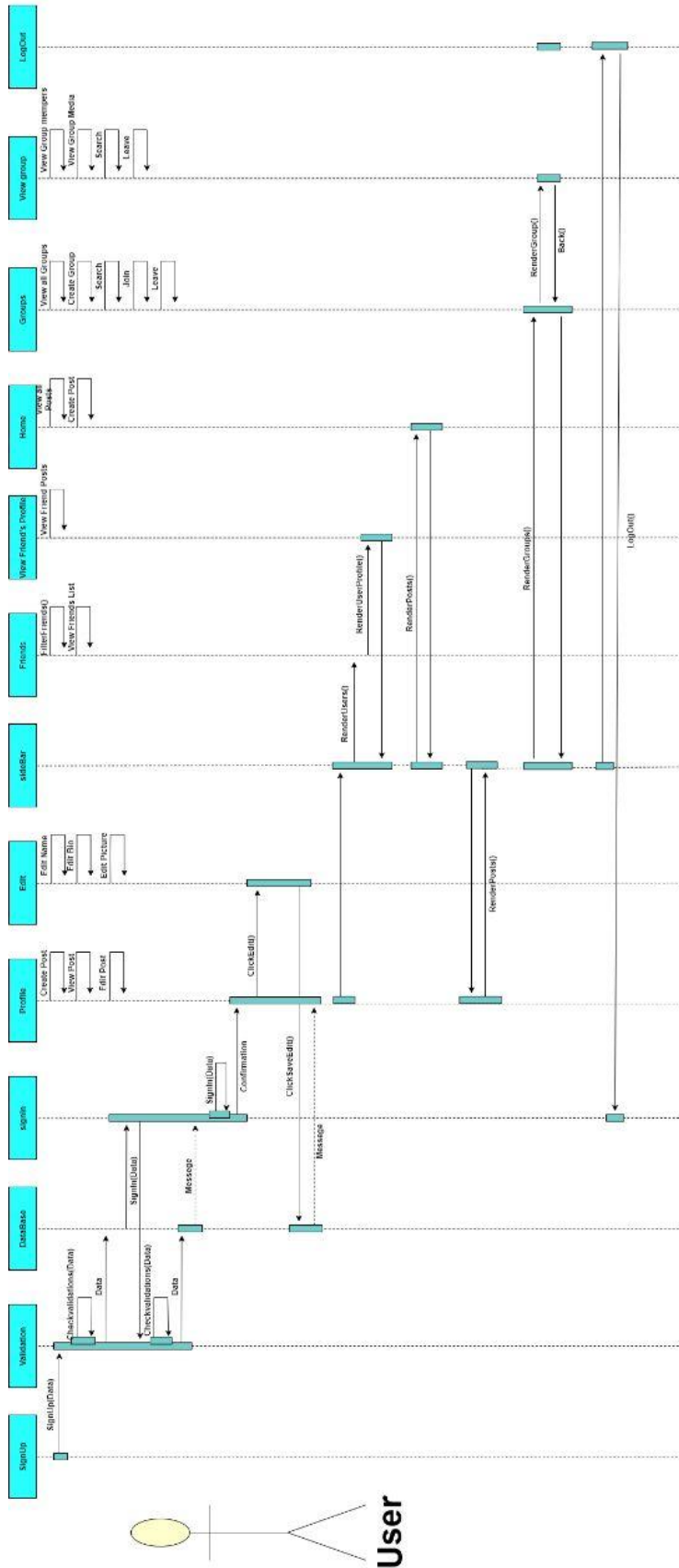
Figure(4): Data Flow Diagram Level 1

4- UML Class Diagram:

<<Interface>> IGenericRepository
+create(entity : Entity) : void
+getById(id : UUID) : Entity
+getAll() : array<Entity>
+delete(id : UUID) : void
+update(id : UUID, entity : Entity) : void

<<Interface>> IUserRepository
+getAllFriends(id : UUID) : array<User>
+getAllGroups(id : UUID) : array<Group>
+addFriend(userId : UUID, friendId : UUID) : void
+removeFriend(userId : UUID, friendId : UUID) : void
+postInGroup(groupId : UUID, post : Post) : void
+muteFriend(id : UUID) : void

<<Interface>> IPostRepository
+updateText(id : UUID, text : String) : void
+updateFiles(id : UUID, files : array<File>) : void
+getAllFiles(id : UUID) : array<File>

<<Interface>> IGroupRepository
+getAllMembers(id : UUID) : array<User>
+addMember(groupId : UUID, memberId : UUID) : void
+removeMember(groupId : UUID, memberId : UUID) : void
+transferOwnerShip(groupId : UUID, newOwnerId : UUID) : void
+getAllPosts(id : UUID) : array<Post>
+getMemberPosts(groupId : UUID, memberId : UUID) : array<Post>
+removePostFromGroup(groupId : UUID, postId) : void
+muteMember(groupId : UUID, memberId : UUID) : void

UserRepository
+getAllFriends(id : UUID) : array<User>
+getAllGroups(id : UUID) : array<Group>
+addFriend(userId : UUID, friendId : UUID) : void
+removeFriend(userId : UUID, friendId : UUID) : void
+postInGroup(groupId : UUID, post : Post) : void
+muteFriend(id : UUID) : void

PostRepository
+updateText(id : UUID, text : String) : void
+updateFiles(id : UUID, files : array<File>) : void
+getAllFiles(id : UUID) : array<File>

GroupRepository
+getAllMembers(id : UUID) : array<User>
+addMember(groupId : UUID, memberId : UUID) : void
+removeMember(groupId : UUID, memberId : UUID) : void
+transferOwnerShip(groupId : UUID, newOwnerId : UUID) : void
+getAllPosts(id : UUID) : array<Post>
+getMemberPosts(groupId : UUID, memberId : UUID) : array<Post>
+removePostFromGroup(groupId : UUID, postId) : void
+muteMember(groupId : UUID, memberId : UUID) : void

<<enumeration>> Role
User
Manager
Administrator

<<enumeration>> Gender
Male
Female

baseEntity
+id : UUID
+creationDate : Date

File
+name : String
+type : String
+size : double
+url : String

Post
+authorId : UUID
+authorName : String
+text : String
+files : array<File>

User
+firstName : String
+lastName : String
#fullName : String
+email : String
+password : String
+gender : Gender
+role : Role
+posts : array<Post>
+bio : String
+friends : array<User>
+groups : array<Group>

Group
+managerId : UUID
+managerName : String
+name : String
+members : array<User>
-posts : array<Post>

Figure(5): The UML Class Diagram of the system
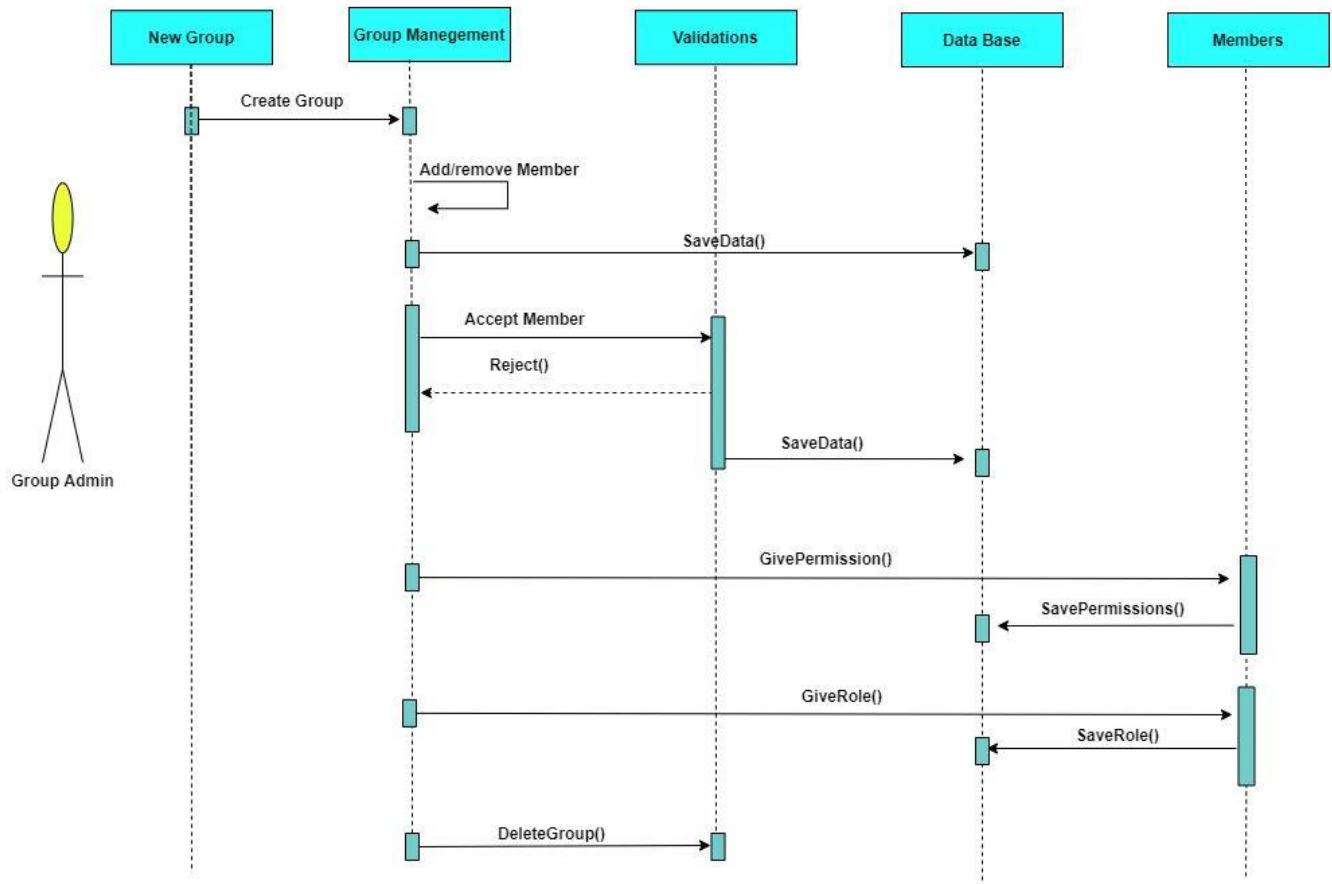
12

5- Use Cases:

Figure(6): Use Cases Diagram

6- Sequence Diagram:

Figure(7): User Sequence Diagram
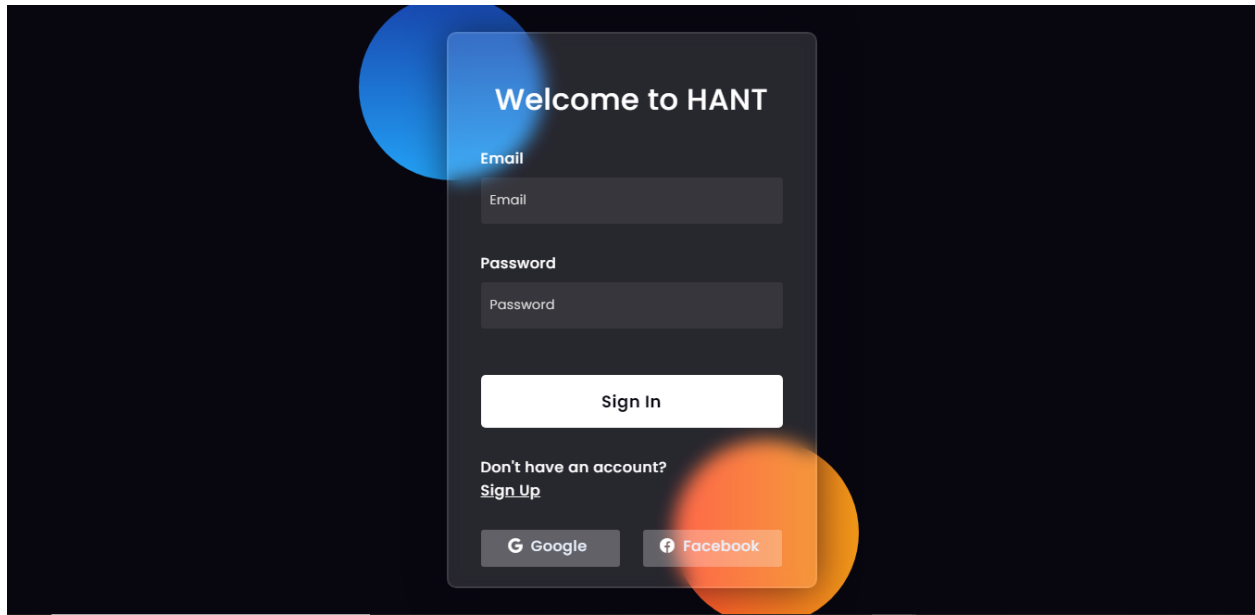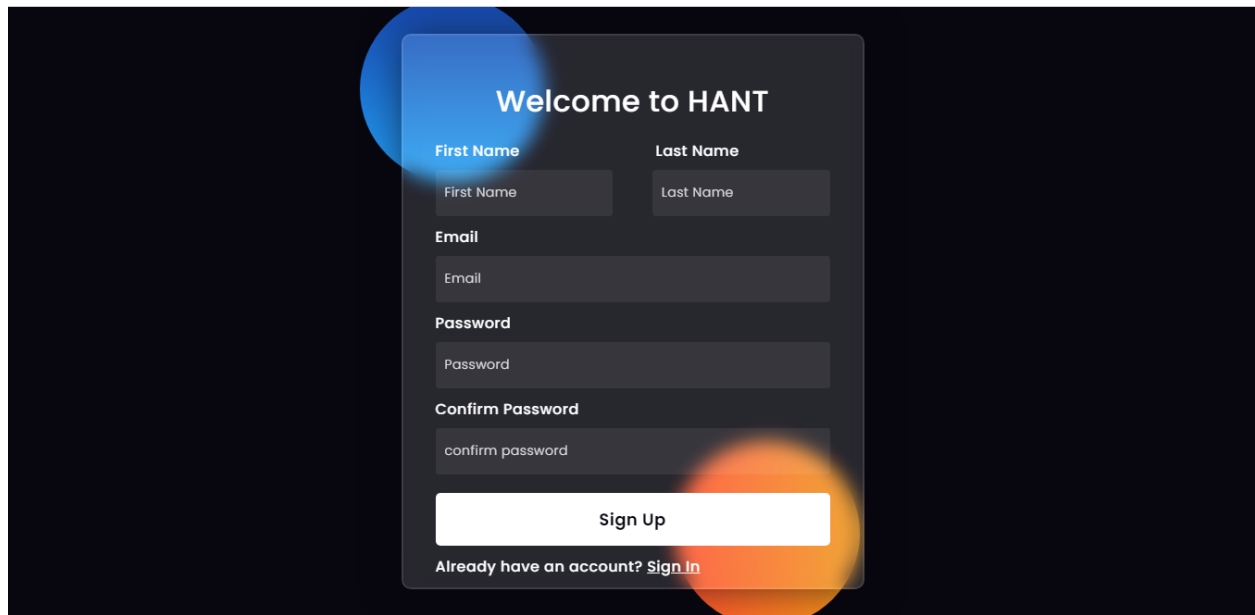
Figure(8): Group Admin Sequence Diagram

6. Future Work

1- AI modeling to filter photos based on facial Recognition.
2- Photo editor.
3- Online meeting.
4- Online compiler to run programs on System.
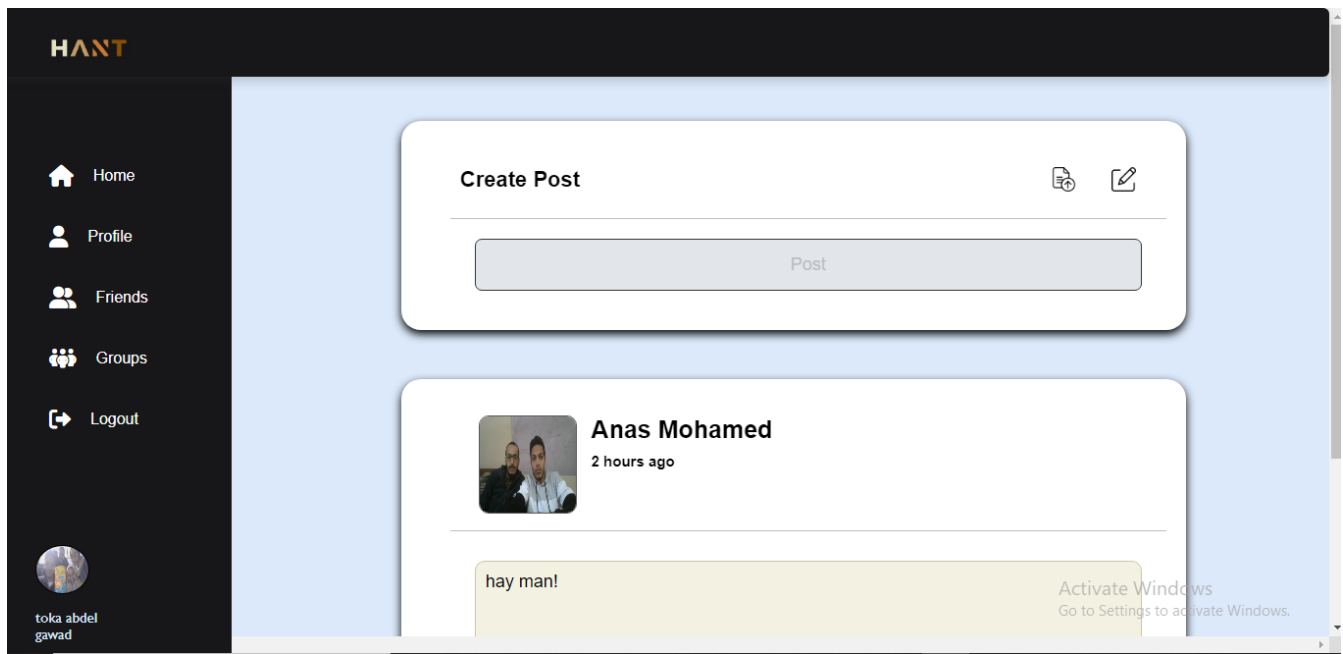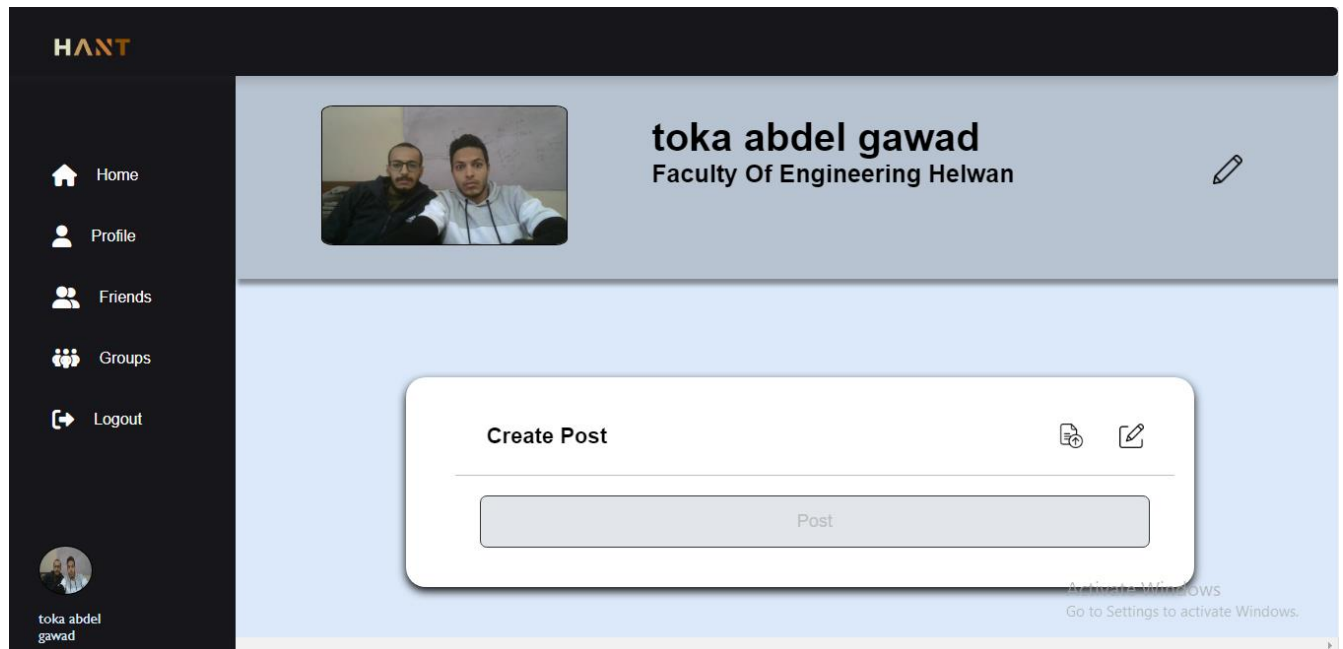5- Version controlling for developing.

7. Screenshots of the Project
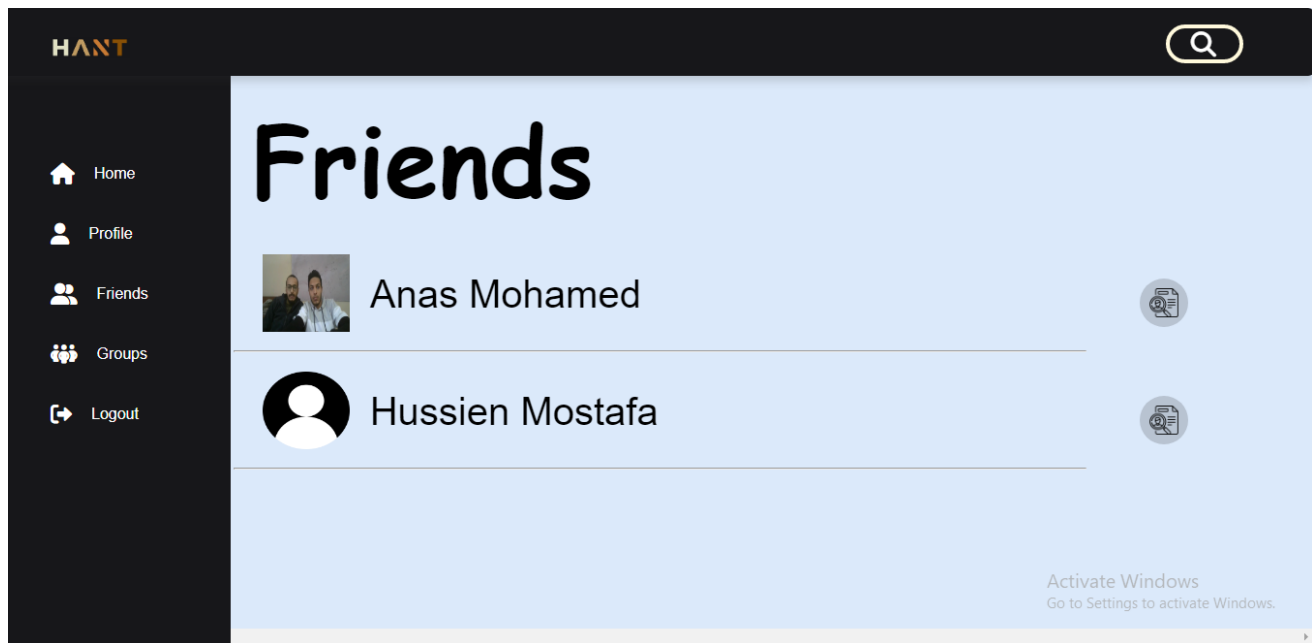

Figure(9): Screenshot of the Login Page

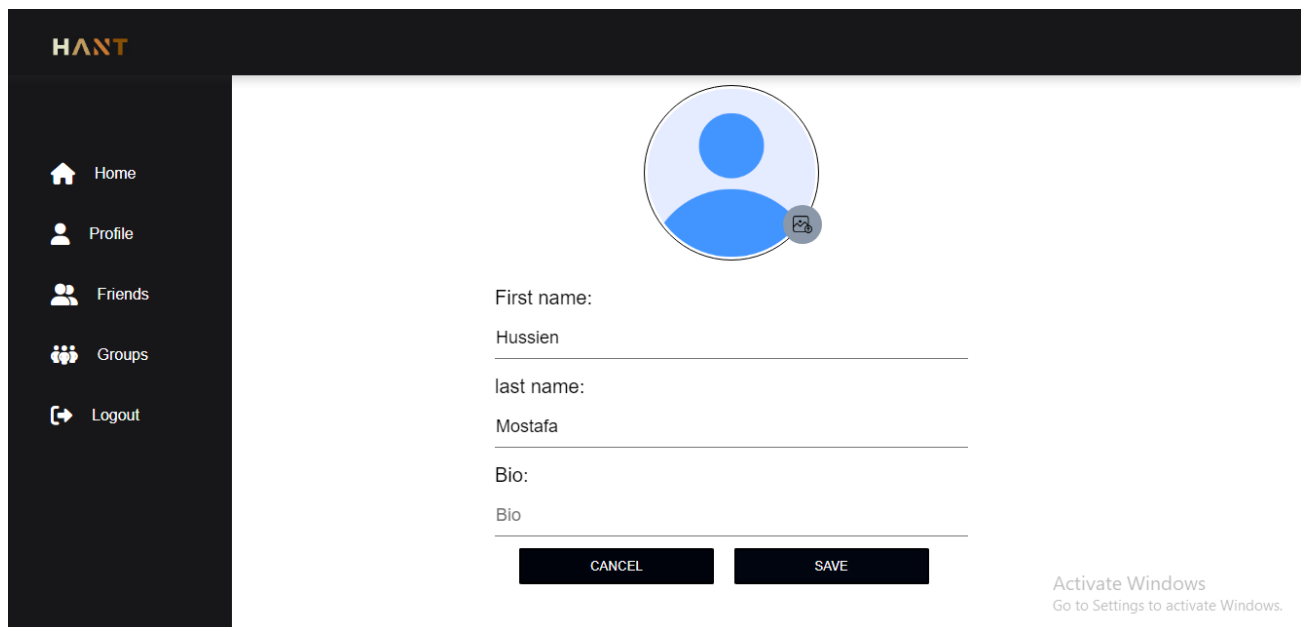Figure(10): Screenshot of the Sign Up Page

Figure(11): Screenshot of the home page of the application

Figure(12): Screenshot of the Profile Page

Figure(13): Screenshot of the Friends Page

Figure(14): Screenshot of the Edit Profile Page