

# JUNIVERSE

RAHAF MOATASEM SALEEM MOHAMMED 0215756

HEBA WALID KREISHAN 0207326

ZAINA BASEM ABDEL HAFEZ 2200374

Project supervisors: Aseel Al-Anani, Rola Al-Khalid





**JUNIVERSE**

## **ABSTRACT**

“A social & education hub platform for the students of king Abduallah II school of information technology, where students can share or publish files or information that helps their colleagues' students. The project will be a web application that students can access through their browser on their personal computers. The web application will include a profile for each student after they register, through which they can access a set of centers or "hubs" as we will call them on our web application. Each center is a useful source of information for the students. Some of the hubs the web app is going to include social hub, where students can communicate with each other's, ask questions and help their colleagues find answers, another example is a file sharing hub “files hub”, where students can access files related to a specific course or share useful files for that course, benefiting everyone. The site will include a lot of useful hubs that're going to help the students throughout their journey in university.”

## **ACKNOWLEDGEMENT**

First, we would like to thank the University of Jordan, the King Abdullah II School of Information Technology, and the Computer Information Systems Department for giving us the chance to work on this project and for providing the support and tools we needed.

We would also like to give special thanks to our supervisors, Dr. Aseel Al-Anani and Dr. Rola Al-Khalid, for their help, support, and useful advice during the whole project. They guided us step by step, and we really appreciate their efforts.

We also want to thank all the academic staff who supported us and helped us learn throughout our university years.

And big thank you to our friends and classmates who helped us, gave us feedback, and supported us along the way.

Finally, we are very thankful to our families for their love, patience, and encouragement during this journey.

# CONTENTS

1. CHAPTER ONE: INTRODUCTION .....	1
Project motivation.....	1
Problem statement.....	1
Project aim .....	1
Objectives .....	2
Project scope.....	3
Software and hardware requirements .....	1
Project limitations .....	1
Project expected output.....	1
Project schedule .....	2
Report outline .....	2
2. CHAPTER TWO: SIMILAR EXISTING SYSTEMS.....	4
Introduction.....	4
Existing systems .....	4
Problems of Existing Systems .....	4
Overall solution approach.....	5
Summary .....	5
3. CHAPTER THREE: SYSTEM REQUIREMENTS ENGINEERING AND ANALYSIS.....	7
Introduction.....	7

Technical study .....	7
Requirement gathering techniques.....	9
Targeted users and stakeholders .....	10
Functional requirements .....	11
Non-functional requirements .....	17
Usability and user experience goals.....	18
4. CHAPTER FOUR: SYSTEM DESIGN .....	19
Context diagram.....	19
Data flow diagram .....	20
Entity relationship diagram (ERD) .....	21
User case diagrams .....	22
Sequence diagram .....	28
Class diagram.....	44
Graphical user interface design.....	45
5. CHAPTER FIVE: SYSTEM IMPLEMENTATION .....	66
Introduction.....	66
Database implementation.....	67
backend implementation .....	68
Frontend implementation .....	70
Graphical User Interfaces .....	73
6. CHAPTER SIX: SYSTEM TESTING AND INSTALLATION .....	84

7. CHAPTER SEVEN: PROJECT CONCLUSION AND FUTURE WORK.....	117
Introduction:.....	117
Overall strengths: .....	117
Overall weakness: .....	118
Future work:.....	118
Summary:.....	118
8. REFERENCES .....	119
9. Appendix .....	120

# LIST OF FIGURES

Figure 4-1: context diagram .....	19
Figure 4-2: data flow diagram.....	20
Figure 4-3: ERD.....	21
Figure 4-4:use case diagram for common cases between all actors .....	22
Figure 4-5: use case diagram for common cases between users part 2 .....	23
Figure 4-6: common admin, moderator, student, use case diagram.....	24
Figure 4-7: use case diagram for admin only cases.....	25
Figure 4-8: use case diagram for common mod and admin cases AND moderator cases .....	26
Figure 4-9: therapist use case diagram .....	27
Figure 4-10: moderator use case diagram .....	27
Figure 4-11: login sequence diagram .....	28
Figure 4-12:upload profile picture sequence diagram.....	28
Figure 4-13: send public message sequence diagram .....	29
Figure 4-14: send private message sequence diagram .....	29
Figure 4-15: upload file sequence diagram .....	30
Figure 4-16: add news sequence diagram .....	30
Figure 4-17: change roles sequence diagram .....	31
Figure 4-18: delete folder sequence diagram .....	31
Figure 4-19: change bio sequence diagram .....	32
Figure 4-20: delete file sequence diagram .....	32
Figure 4-21: delete news sequence diagram.....	33
Figure 4-22: add folder sequence diagram .....	33
Figure 4-23: accept or reject file sequence diagram.....	34
Figure 4-24: delete file sequence diagram .....	34
Figure 4-25: delete events sequence diagram.....	35
Figure 4-26: delete message sequence diagram .....	35
Figure 4-27: check task sequence diagram.....	36
Figure 4-28: add task sequence diagram .....	36
Figure 4-29: update quote sequence diagram.....	37
Figure 4-30: delete task sequence diagram .....	37
Figure 4-31: edit public message sequence .....	38
Figure 4-32: attach file sequence diagram .....	38

Figure 4-33: edit private message sequence diagram.....	39
Figure 4-34: delete public message sequence diagram .....	39
Figure 4-35: delete private message sequence diagram .....	40
Figure 4-36: update file sequence diagram .....	40
Figure 4-37: update note sequence diagram.....	41
Figure 4-38: update folder sequence diagram .....	41
Figure 4-39: add note sequence diagram.....	42
Figure 4-40: update news sequence diagram .....	42
Figure 4-41: delete note sequence diagram.....	43
Figure 4-42: class diagram .....	44
Figure 4-43: profile page design prototype .....	45
Figure 4-44: public chat design prototype.....	45
Figure 4-45: file sharing hub, folders design prototype .....	46
Figure 4-46: file sharing hub, design prototype .....	46
Figure 4-47: mental health hub, design prototype.....	47
Figure 4-48: news hub, design prototype .....	47
Figure 4-49: events hub, design prototype .....	48
Figure 4-50: e-card design prototype .....	48
Figure 4-51: file management design prototype.....	49
Figure 4-52: news management prototype design.....	49
Figure 4-53: students management prototype design.....	50
Figure 4-54: moderators management prototype design.....	50
Figure 4-55: student profile page design prototype.....	51
Figure 4-56: student social hub prototype .....	51
Figure 4-57: student file sharing hub design prototype.....	52
Figure 4-58: student file sharing hub part 2 design prototype .....	52
Figure 4-59: student mental health hub design prototype .....	53
Figure 4-60: student news design prototype .....	53
Figure 4-61: student events design prototype .....	54
Figure 4-62: student e-card design prototype.....	54
Figure 4-63: therapist profile page design prototype .....	55
Figure 4-64: news hub design prototype .....	56
Figure 4-65: therapist chats design prototype .....	56
Figure 4-66: moderator profile page design prototype.....	57
Figure 4-67: moderator social hub design prototype.....	57

Figure 4-68: moderator file sharing hub design prototype .....	58
Figure 4-69: moderator public chat design prototype .....	58
Figure 4-70: moderator news design prototype.....	59
Figure 4-71: moderator event design prototype .....	59
Figure 4-72: moderator pending files management design prototype.....	60
Figure 4-73: moderator e-card design prototype .....	60
Figure 4-74: login page design prototype.....	61
Figure 4-75: profile dashboard design prototype .....	62
Figure 4-76: last files design prototype.....	62
Figure 4-77: help box design prototype .....	63
Figure 4-78: footer design prototype.....	63
Figure 4-79: logo design .....	64
Figure 4-80: logo design part 2 .....	65
Figure 5-1: private chat JPA entity .....	67
Figure 5-2: message JPA entity.....	67
Figure 5-3: message service code snippet .....	68
Figure 5-4: therapist note code snippet .....	69
Figure 5-5: file service code snippet .....	69
Figure 5-6: App.js code snippet .....	70
Figure 5-7: formLogin.js code snippet.....	71
Figure 5-8: news management.js code snippet.....	71
Figure 5-9: therapistnotes.js code snippet .....	72
Figure 5-10: social hub.js code snippet.....	72
Figure 5-11: homepage UI screenshot .....	73
Figure 5-12: about us UI screenshot .....	74
Figure 5-13: login UI screenshot.....	74
Figure 5-14: profile UI screenshot .....	75
Figure 5-15: public chat UI screenshot .....	76
Figure 5-16: mental health hub UI screenshot .....	76
Figure 5-17: private chat with a therapist UI screenshot.....	77
Figure 5-18: news hub UI screenshot.....	78
Figure 5-19: filesharing hub UI screenshot .....	79
Figure 5-20: filesharing hub UI screenshot 2.....	79
Figure 5-21: file management UI screenshot .....	80
Figure 5-22: e-card UI screenshot.....	80

Figure 5-23: therapist chats UI screenshot .....	81
Figure 5-24: therapist notes UI screenshot.....	81
Figure 5-25: student management UI screenshot .....	82
Figure 5-26: moderators managements UI screenshot .....	82
Figure 5-27: banned users management.....	83
Figure 9-1: survey results.....	120

# LIST OF TABLES

Table 1-1: needed software .....	1
Table 1-2: needed hardware .....	1
Table 1-3: project schedule .....	2
Table 3-1: stakeholders .....	10
Table 3-2: functional requirements .....	11
Table 3-3: non-functional requirements .....	17
Table 6-1: home page test cases .....	84
Table 6-2: login page test cases.....	85
Table 6-3: profile page, test cases .....	86
Table 6-4: social hub, test cases .....	90
Table 6-5: mental health hub, test cases.....	91
Table 6-6: news hub, test cases .....	93
Table 6-7: file sharing hub, test cases .....	96
Table 6-8: E-card test cases.....	104
Table 6-9: News management Test Cases.....	105
Table 6-10:students management, test cases.....	107
Table 6-11: moderators management, test cases .....	108
Table 6-12: banned users management, test cases .....	109
Table 6-13: file management, test cases.....	110
Table 6-14: therapist chats, test cases .....	112
Table 6-15: therapist notes, test cases .....	114
Table 6-16: notifications, test cases .....	115

## **1. CHAPTER ONE: INTRODUCTION**

### **Project motivation**

The motivation for the project is to build a platform for KASIT students where they can feel at home and find a supportive digital environment for them and help students to connect, share knowledge, and collaborate across various subjects and activities in a structured, meaningful way

### **Problem statement**

Students at King Abdullah II College currently lack a centralized, accessible platform for academic and social engagement, which limits opportunities for collaborative learning, peer-to-peer support, and efficient resource sharing

To address these challenges, we propose developing a web-based social educational platform that provides students with a digital space to communicate, collaborate, and share resources by establishing "hubs" dedicated to various purposes, such as a social hub for discussion and a file-sharing hub for academic resources. We aim to enhance the college's learning environment and foster a stronger, more connected community. This platform will empower students to support each other, creating a more inclusive and resource-rich academic experience.

### **Project aim**

The aim of this project is to create an interactive web-based platform for the students of King Abdullah II College, designed to facilitate academic collaboration, resource sharing, and community building, by fostering a collaborative online environment, the platform will not only support academic success but also cultivate a sense of community, enabling students to interact meaningfully and support one another both socially and academically. Ultimately, the goal is to

enhance students' learning experiences, encourage peer engagement, and streamline access to valuable educational resources within a single, accessible digital space.

## **Objectives**

**Enable Knowledge Sharing:** using a digital platform for students at King Abdullah II College to share and exchange educational resources.

**Support Academic and Social Interaction:** by enabling students to interact socially and academically through hubs

**Enhance Student Involvement:** by providing a comfortable and user-friendly environment that encourages student participation and involvement in college events, discussions, and study groups.

**Promote Collaborative Learning:** by encouraging direct learning among students

**Support Mental Well-being:** by offering a dedicated mental health hub, promoting student wellness and support.

**Share College News and Updates:** by keeping students informed about the latest college events, news, and important announcements

## **Project scope**

"JUniVerse" aims to develop a website designed for students at King Abdullah II College, accessible via their web browsers. The application will feature the following capabilities:

### **Hubs:**

- 1. Social Hub:** A platform for students to engage in discussions and ask questions
- 2. File Sharing Hub:** An area for students to upload course-related files
- 3. Mental Health Hub:** Guidance on mental health topics provided to students.
- 4. Events Hub:** A section providing details about upcoming college events.
- 5. News Hub:** Updates college activities and announcements.
- 6. E-Card Hub:** a digital copy

**Admin and Moderator Controls:** Administrators and moderators will have the ability to manage user roles, oversee content, and add or remove resources within the hubs.

**Interactive Messaging:** Students can post messages in the hubs to discuss and help one another.

## **Software and hardware requirements**

### **Needed software**

The following table 1.1 represents the software needed.

*Table 1-1: needed software*

<b>name</b>	<b>usage</b>
<b>IntelliJ</b>	Back-end, Java development IDE, using spring framework.
<b>VS code</b>	Front-end, React app development IDE.
<b>PostgreSQL</b>	As a DBMS
<b>Browser</b>	To run the web application
<b>Swagger</b>	To test our backend functionality, and document it, it works as communication point between frontend and backend developers.
<b>Notion</b>	For teammates communication and organization
<b>StarUML &amp; Lucid Chart</b>	For sequence, class and other UML diagrams
<b>PowerPoint and Canva</b>	For designing UI & poster & other design requirement.

## Needed hardware

*Table 1-2: needed hardware*

<b>name</b>	<b>usage</b>	<b>number</b>	<b>price</b>
<b>laptops</b>	<p>Laptops with internet access and a modern web browser, to develop the application using the programs mentioned.</p> <p>Specifications:</p> <p>CPU: Intel Core i5</p> <p>Storage: 512GB</p> <p>RAM: 16GB.</p> <p>OS: windows</p>	3	500-600 JD for each
<b>Server (if system gets deployed)</b>	<p>To install our system on a server with the following features:</p> <ul style="list-style-type: none"> <li>• 32 CPU cores</li> <li>• 128 GB RAM</li> <li>• 1.5 TB storage</li> </ul>	1	\$219.99 monthly

## **Project limitations**

- Only students from King Abdullah II College will have access.
- The website should be ready 25/may/2025.
- Development costs should not exceed 10000 JD.

## **Project expected output**

The project's expected output is a web application that is fully functioning, the website is intended to be able to serve the needs of KASIT students, with its fully functioning hubs where each hub helps with a specific thing, including the social hub, file sharing hub, mental health hub, events hub, news hub, knowledge hub, the application is going to have three type of users, each one of them serves different topic, for example the social hub is a general chat between students where they can communicate their problems or chat regularly, file sharing hub where students can help each other with studying materials, mental health hub is a hub generated by the moderators and admins where they can post mental health related topics to help students go through their university life with support, events hub is for announcing new events, news hub is for news in the KASIT school, knowledge hub is going to be about IT related topics to enlighten students to learn more about their field and spread knowledge.

## Project schedule

Table 1-3: project schedule

ID	Task Name	Duration	Dependency
T1	Project idea and brainstorming	5 Days	-
T2	Planning the project	6 Days	T1
T3	Hardware and software requirements	1 Day	T2
T4	Feasibility Study	2 Days	T3
T5	Functional Requirements	8 Days	T4
T6	Non-functional Requirements	4 Days	T5
T7	Context diagram	1 Day	T5
T8	Dataflow diagram	2 Days	T7
T9	Entity Relationship Diagram	2 Days	T8
T10	Use-Case diagrams	3 Days	T4
T11	Sequence Diagram	3 Days	T10
T12	Graphical user interface (GUI)	30 Days	T11
T13	Implementation	150 Days	T12

## Report outline

- **Chapter one:** Shows a preamble in section 1.1, the project motivation is elaborated in Section 1.2, in addition, the problem statement is stated in Section 1.3, Section 1.4 lists the project aim and objectives, whereas the project scope is identified in section 1.5, Section 1.6 provide the project software and hardware requirement and Section 1.7 highlights the project limitation, furthermore, project expected output is addressed in Section 1.8, in section 1.9 the project is scheduled, and the report outline is finally present in Section 1.10.
- **Chapter Two:** Shows an introduction about existing systems in Section 2.1, the existing

system is elaborated in Section 2.2, in addition, the difference between the two systems is in Section 2.3, whereas the summary is identified in Section 2.4

- **Chapter Three:** shows an introduction in Section 3.1, feasibility study is elaborated in Section 3.2, in addition, and Requirements Elicitation Techniques in Section 3.3, the targeted users are stated in Section 3.4, Section 3.5 shows Functional requirements definition and functional requirement specification is identified in Section 3.6, Section 3.7 highlights the nonfunctional requirements, furthermore, the summary output is addressed in Section 3.8
- **Chapter Four:** Shows an introduction in Section 4.1, Context Diagram elaborated in Section 4.2, the Data Flow Diagram (DFD) in Section 4.3, Section 4.4 provides the Architectural Design, and the Entity Relationship Diagram is stated in Section 4.5, in addition, the Use-Case Diagram is stated in Section 4.6, and Section 4.7 include UML Sequence Diagram, Finally Section 4.8 provides Graphical User Interface design.
- **Chapter Five:** Shows system implementation (database, backend and front end) and technologies required.
- **Chapter Six:** Shows system installation guide.
- **Chapter Seven:** Shows future project work and stretch goals.

## **2. CHAPTER TWO: SIMILAR EXISTING SYSTEMS**

### **Introduction**

This chapter mentions similar existing systems that are related to our system, highlights the problems associated with these systems, and presents an overall solution approach.

### **Existing systems**

In our field, while platforms like **Facebook Groups, Google Drive, and messaging apps like WhatsApp** are used by students for sharing information and files, they lack integration and specific features for academic collaboration.

Also, **learning management systems like Moodle** are controlled by faculty and do not encourage peer-to-peer interaction or social engagement.

These limitations create an opportunity to develop "JUniVerse" as an all-in-one platform to meet students' academic and social needs.

### **Problems of Existing Systems**

#### **1. Lack of Centralized Communication for Students:**

Many current systems, such as social media platforms or university forums, do not offer a dedicated space for students to communicate in an organized way. Students are forced to use different platforms, like social media groups or personal messaging, which can be disorganized and difficult to manage.

## **2. Missing Information About Events and News:**

Students often miss important college events and updates because there isn't one platform where all this information is shared.

## **3. Difficulty in Accessing Course-Related Files:**

Students find it hard to access course materials and files. There's no simple platform to share study materials.

## **4. Lack of Mental Health Support:**

Many student platforms don't offer mental health support. Students facing stress or emotional issues have no easy way to find help.

## **Overall solution approach**

JUniVerse will solve these problems by providing different hubs to the students:

1. The **Social Hub** will allow students to communicate and share information easily.
2. The **File Sharing Hub** will help students upload and access course materials quickly.
3. The **Mental Health Hub** will provide resources and support for students in need.
4. The **Events Hub and News Hub** will keep students informed about college events and updates.

## **Summary**

In summary, JUniVerse was created to help students work together and get support by fixing the problems with current systems. Unlike platforms like Facebook or Moodle, JUNiVerse brings together social, academic, and mental health support in one easy-to-use platform. It provides a

space for chatting, sharing files, and getting event updates, making it easier and more enjoyable for students to stay connected and organized.

### **3. CHAPTER THREE:**

## **SYSTEM REQUIREMENTS ENGINEERING AND ANALYSIS**

#### **Introduction**

This chapter serves as the foundation, providing a comprehensive understanding of the system's requirements, ensuring a clear roadmap for design and implementation. By dissecting connectivity, security, user interfaces, and hardware components, we pave the way for a robust and effective solution.

We ensure that our project is not just a concept but a well-thought-out and technically feasible solution. This section acts as a guide for developers, detailing the specific needs and functionalities that will bring our innovative pharmacy vending machine system to life.

#### **Technical study**

##### **1. Website Development:**

- Developing a website to facilitate student interaction, resource sharing, and communication requires a solid understanding of modern web development technologies. For the **front-end**, we will utilize **React**, a powerful JavaScript library for building dynamic and interactive user interfaces. React allows us to create reusable components and ensures an efficient, smooth user experience.
- The backend of our website is powered by **Java**, a versatile programming language that integrates seamlessly with various libraries and frameworks to handle complex server-side functionalities.

- The framework we'll use is **Spring Boot**, which enables efficient connection and management of front-end elements using Java. The **Spring Web library** will serve as the bridge between the front-end and back-end using RESTful Api.
- For dependency management, we rely on **Maven** to ensure all required libraries and dependencies are properly configured.

## 2. Development Tools:

- The development team utilizes **Visual Studio Code (VSCode)** for React-based front-end development and **IntelliJ IDEA** for back-end development.
- Both tools offer a rich set of features, such as code completion, debugging tools, and Maven integration (IntelliJ IDEA), improving the efficiency and productivity of the development process.

## 3. Database Management:

- The database for the Social Hub Platform is designed to efficiently manage the storage and retrieval of student data, such as posts, messages, and resources.
- We implement the database using **Spring Data JPA (Java Persistence API)**, which simplifies and optimizes database operations in Java-based applications.
- For the database system, we use **PostgreSQL**, known for its scalability, advanced features, and compliance with SQL standards. Spring Boot provides flexibility to integrate other database systems in the future if needed.

## 4. Scalability and performance:

- Employing scalable interface and optimizing website performance.

## **Requirement gathering techniques**

### **Survey**

The survey conducted for this project is attached in the appendix for reference. It includes the questions and responses collected to support the findings discussed in this project.

-

You can check them at last pages 120 in the appendix.

## **Targeted users and stakeholders**

- **KASIT students**

The users targeted in our system JUNiVerse are mostly students from the King Abdullah ii school for information and technology, to make their university life easier, make them exchange files and information, so everyone feels included and part of the KASIT community. **They'll be playing roles like moderators**, normal students. Therefore, our targeted users are **students**.

- **KASIT professors**

Coming to the admins, the targeted users are going to be professors, in the faculty, so they can manage moderators, students and the actions they make.

- **University Therapists**

Coming to the mental health hub therapist, they should be an authorized therapists who work in the university of Jordan, they'll play an important role in helping KASIT students preserve a healthy psychological condition. The following table 3-1 sums up the main stakeholders and targeted users of the system

*Table 3-1: stakeholders*

<b>ID</b>	<b>Stakeholder</b>
1	Moderators
2	Admins (professors)
3	University therapists
4	KASIT Students

## Functional requirements

The following table 3-2 represents the functional requirements the system should perform.

*Table 3-2: functional requirements*

FR-ID	Name	ST	Description
	<b>login</b>	All users	<b>Users access the system by entering the right credentials, then they'll be directed to their profile page</b>
	<b>Change bio</b>	All users	<b>Allow users to change their bio.</b>
	<b>Change profile picture</b>	All users	<b>Allow users to change their profile picture.</b>
	<b>Change cover picture</b>	All users	<b>Allow users to change their cover picture.</b>
	<b>Remove the profile picture</b>	All users	<b>Allow users to remove their profile picture.</b>
	<b>Remove the cover picture</b>	All users	<b>Allow users to remove their cover picture.</b>

	<b>Add task</b>	<b>All users</b>	<b>Allow users to add tasks to their to do list</b>
	<b>Delete task</b>	<b>All users</b>	<b>Delete tasks from their profile page</b>
	<b>Update tasks</b>	<b>All users</b>	<b>Users can check or uncheck tasks</b>
	<b>Update quote</b>	<b>All users</b>	<b>Users can update a quote</b>
	<b>Send/Receive public messages</b>	<b>All users</b>	<b>Users can chat on both public and private chat</b>
	<b>Delete a public message</b>	<b>Admins/moderators.</b>	<b>Admins or mods can delete any public message in a role hierarchy manner.</b>
	<b>Delete message as a message owner</b>	<b>All users</b>	<b>All users can delete their own messages in public or private chat</b>

	<b>Edit message as a message owner</b>	<b>All users</b>	<b>All users can edit their own messages in public or private chat</b>
	<b>Add folder</b>	<b>Admins.</b>	<b>Add folders to the file sharing hub.</b>
	<b>Delete folders</b>	<b>Admins.</b>	<b>Delete folders from the file sharing hub.</b>
	<b>Update folder</b>	<b>Admins</b>	<b>Update folder name and folder description</b>
	<b>Upload file</b>	<b>Students/ admins/ moderators</b>	<b>Uploading material related files to a specific material folder.</b>
	<b>Delete file</b>	<b>Admins/ moderators</b>	<b>Delete files from material's folder.</b>
	<b>Update file</b>	<b>Admins/ moderators</b>	<b>Update file's name or description</b>

	<b>Download uploaded files</b>	<b>All users (except therapist)</b>	<b>All users can download a file that has been uploaded to a specific folder</b>
	<b>View a file</b>	<b>All users (except therapist)</b>	<b>All users can view a file before downloading.</b>
	<b>Accept File</b>	<b>Moderators</b>	<b>Accept a pending file</b>
	<b>Reject File</b>	<b>Moderators</b>	<b>Reject a pending file</b>
	<b>Add events</b>	<b>Admins</b>	<b>Add an event to the calendar to notify all users.</b>
	<b>Delete events</b>	<b>Admins</b>	<b>Delete events from the news hub.</b>
	<b>Add news</b>	<b>Admins</b>	<b>Add news to the news hub.</b>
	<b>Delete news</b>	<b>Admins</b>	<b>Delete news from the news hub.</b>

	<b>Update news</b>	<b>Admins</b>	<b>Update news from the news hub.</b>
	<b>Ban users</b>	<b>Admins</b>	<b>ban users and remove them for the platform</b>
	<b>Unban users</b>	<b>Admins</b>	<b>Unban users and return them to the platform</b>
	<b>Promote to moderator</b>	<b>Admins</b>	<b>Promoting regular students to moderators.</b>
	<b>Demoting to student</b>	<b>Admins</b>	<b>Demoting moderators to students.</b>
	<b>Attach file to private chat</b>	<b>All users</b>	<b>All users can attach files to private chat</b>
	<b>Add note</b>	<b>therapist</b>	<b>Therapist can add a note related to a specific chat</b>
	<b>Delete note</b>	<b>Therapist</b>	<b>Therapist can delete a note that he created</b>

	<b>Edit note</b>	<b>Therapist</b>	<b>Therapist can edit a note after adding it</b>
	<b>Receive notification</b>	<b>All users</b>	<b>All users receive a notification if a new message on private chat is sent, a file is accepted/rejected, or a new event is coming</b>

## Non-functional requirements

The system ensures the following non-functional requirements are met as shown in table 3-3,

note that priorities are **ordered** from **1 the highest** to lowest increasingly.

*Table 3-3: non-functional requirements*

NFR-ID	Description	Priority	note
NFR-1	<b>Security:</b> secure Authentication and Authorization	1	-
NRF-2	<b>Performance:</b> ensuring fast performance and responses and handling good number of users.	1	-
NRF-3	<b>Reliability:</b> The system should be available 99.9% of the time.	1	-
NRF-4	<b>Usability:</b> a user-friendly interface that is intuitive and easy to navigate, and feedback to some parts makes it easier.	2	-
NRF-5	<b>Maintainability:</b> the system is easy to maintain, and the code is readable, using clean architecture in the back end, and MVC pattern to connect layers; makes the system easy to maintain by other developers, also easy to refactor system parts, using react also adds another value to easier maintainability.	1	e.g.: We're documenting the system to make it easier to maintain!!

## **Usability and user experience goals**

- **Usability**

The platform should be simple and intuitive, with clear navigation and easily understood buttons and sections. Error messages should be clear and offer helpful suggestions, and tasks should be efficient. The platform must be responsive across devices, ensuring that content is easily readable.

- **Users experience goals**

To improve the user experience, the platform should be interactive to increase engagement.

Community interactions should encourage practice and collaboration. Positive feedback mechanisms and personalized learning goals allow users to have a better experience. The layout should provide easy access to resources, and content should be useful.

## 4. CHAPTER FOUR: SYSTEM DESIGN

### Context diagram

A context diagram provides a high-level overview of the system, illustrating its interactions with external entities. It highlights the system boundaries and the flow of information between the system and its environment

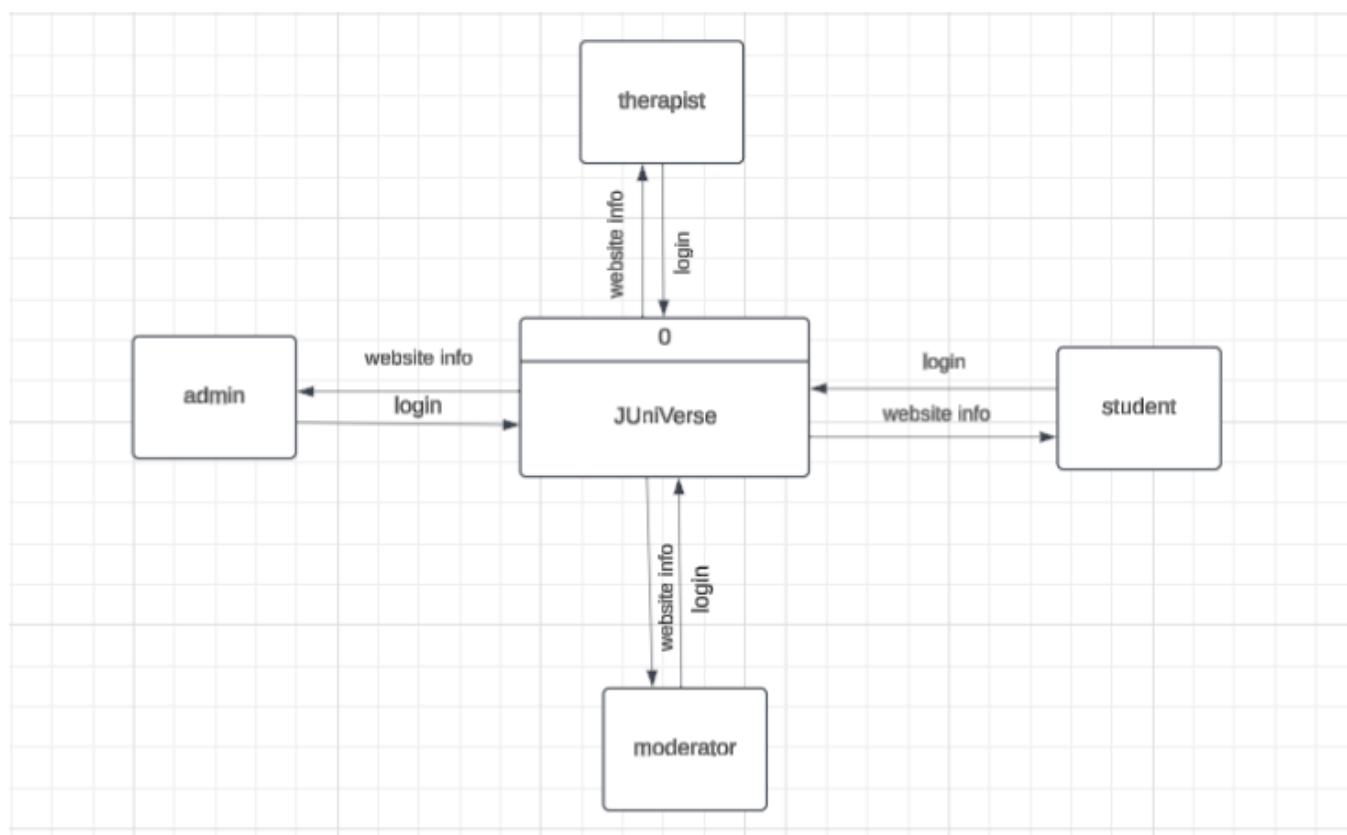


Figure 4-1: context diagram

## Data flow diagram

A Data Flow Diagram (DFD) visually represents the flow of data within a system, illustrating how information is processed and transferred between various components.

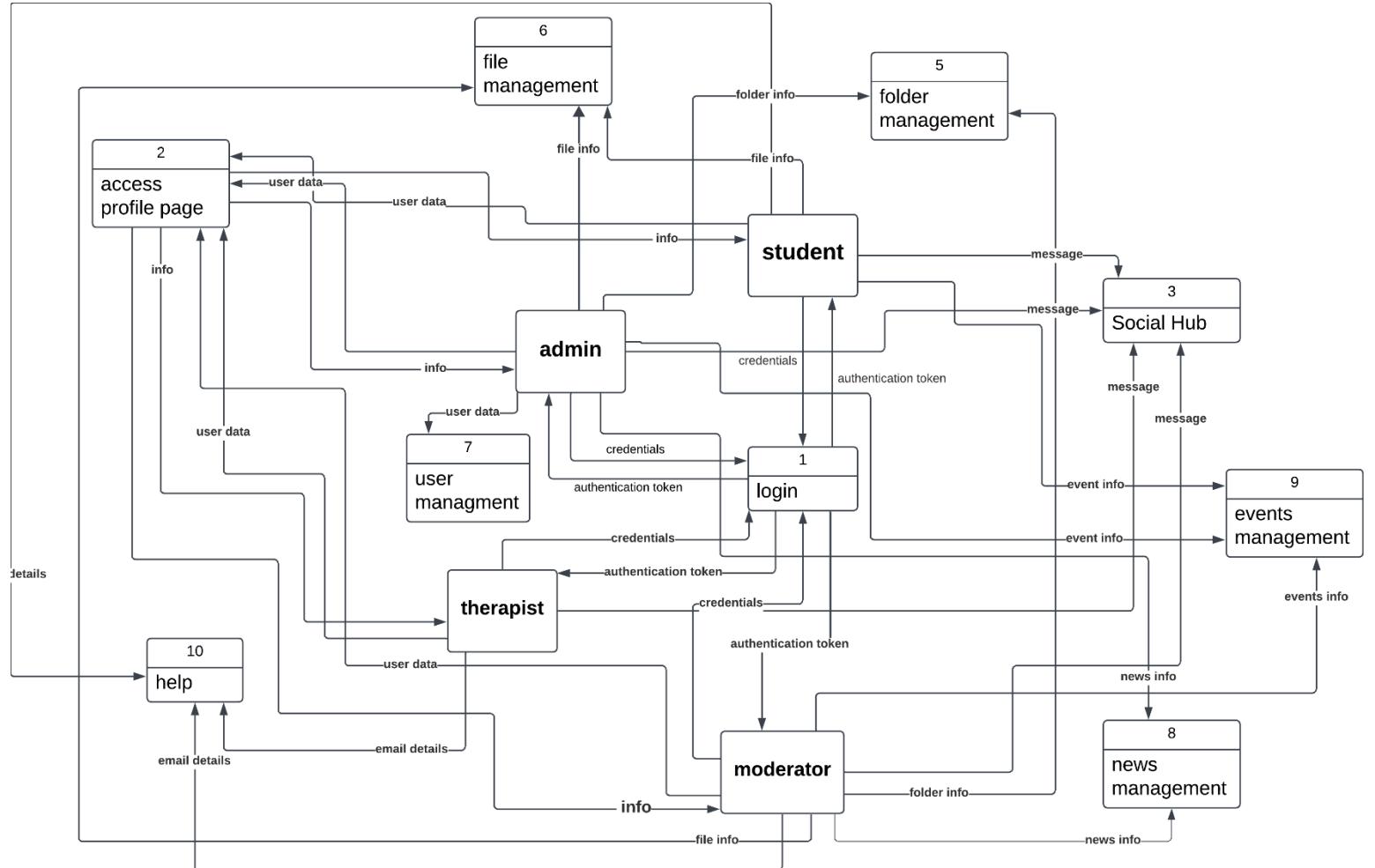


Figure 4-2: data flow diagram

## Entity relationship diagram (ERD)

The following diagram represents the ERD for the system.

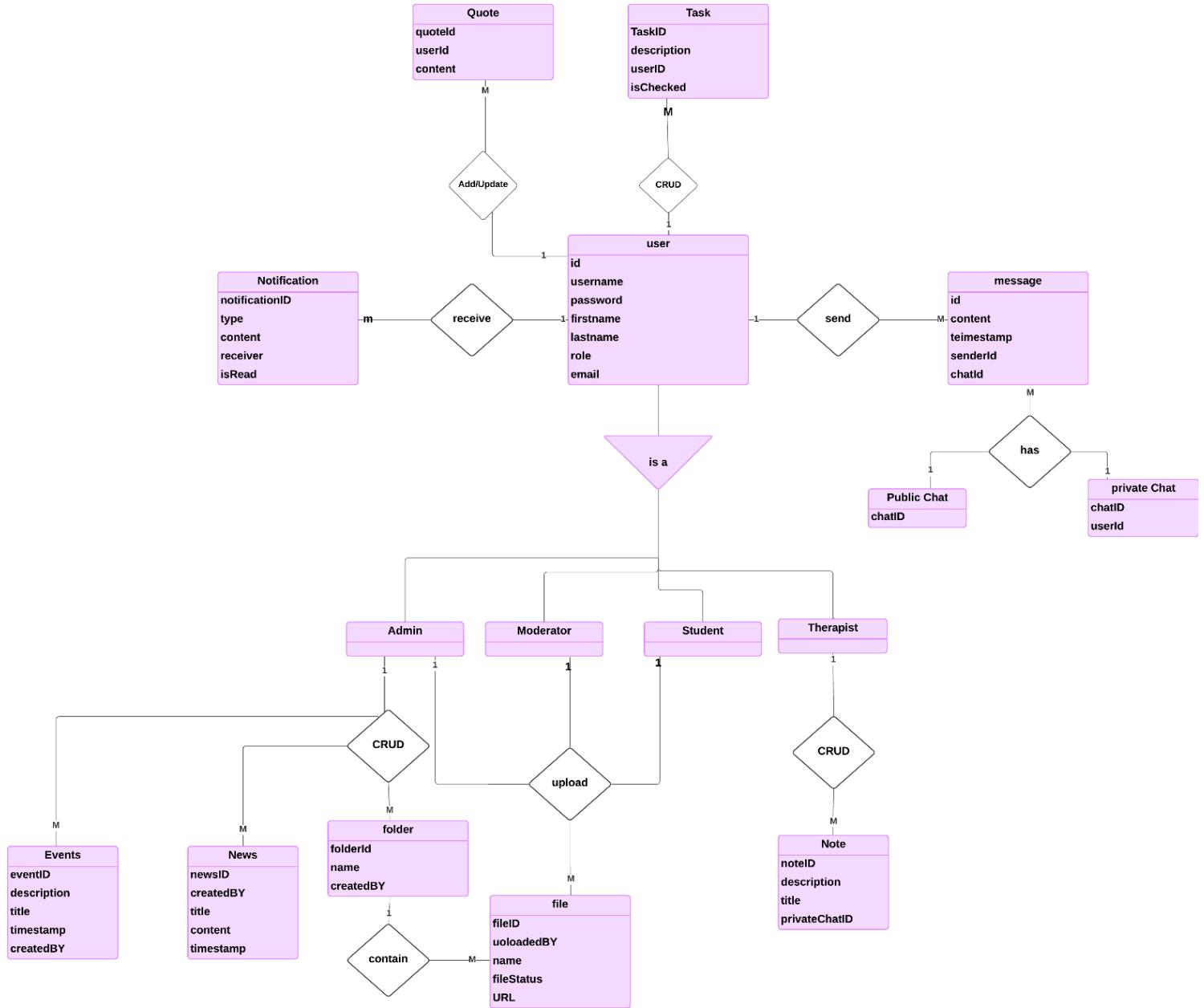


Figure 4-3: ERD

## User case diagrams

The following diagram represents the use case diagram for the common use cases between all actors in the system.

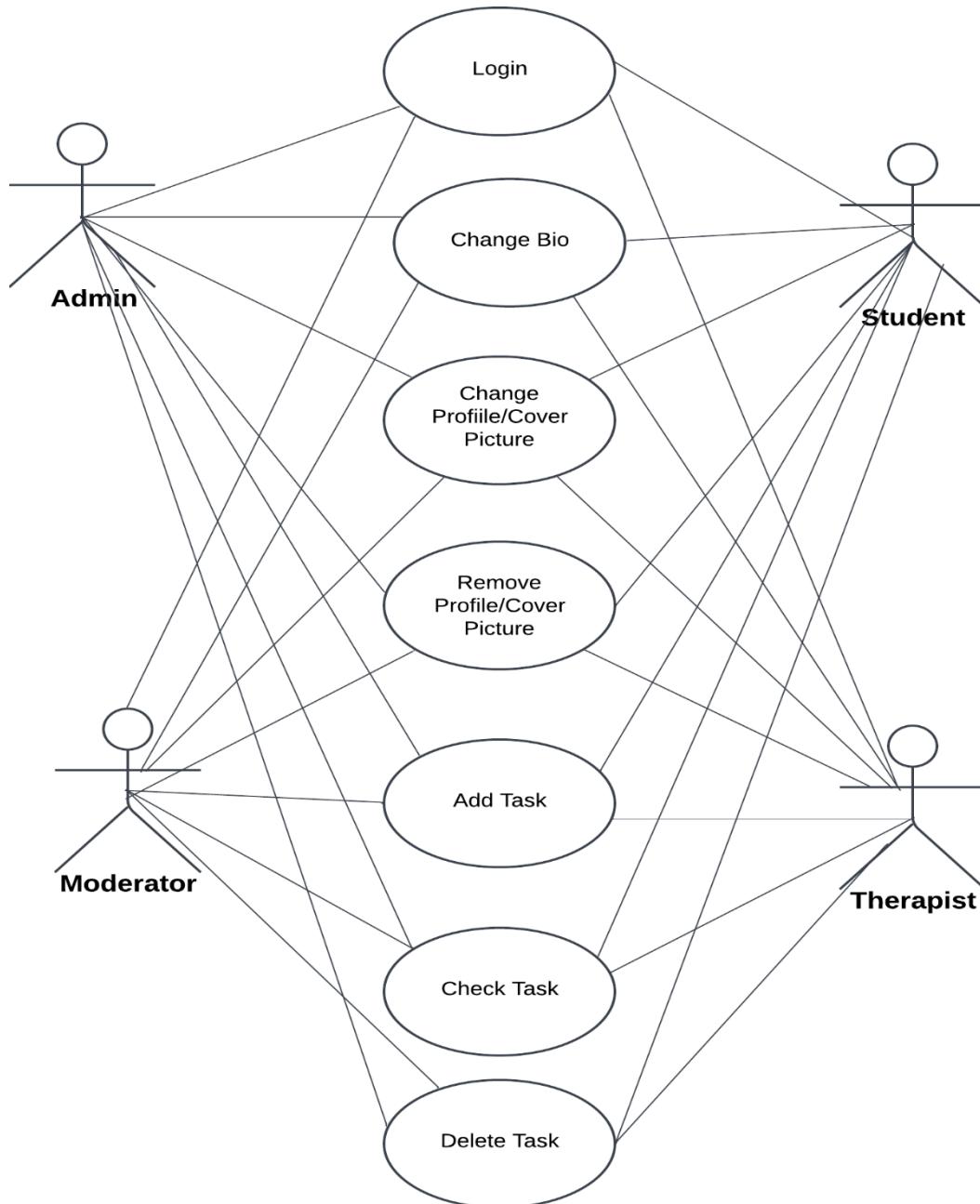


Figure 4-4:use case diagram for common cases between all actors

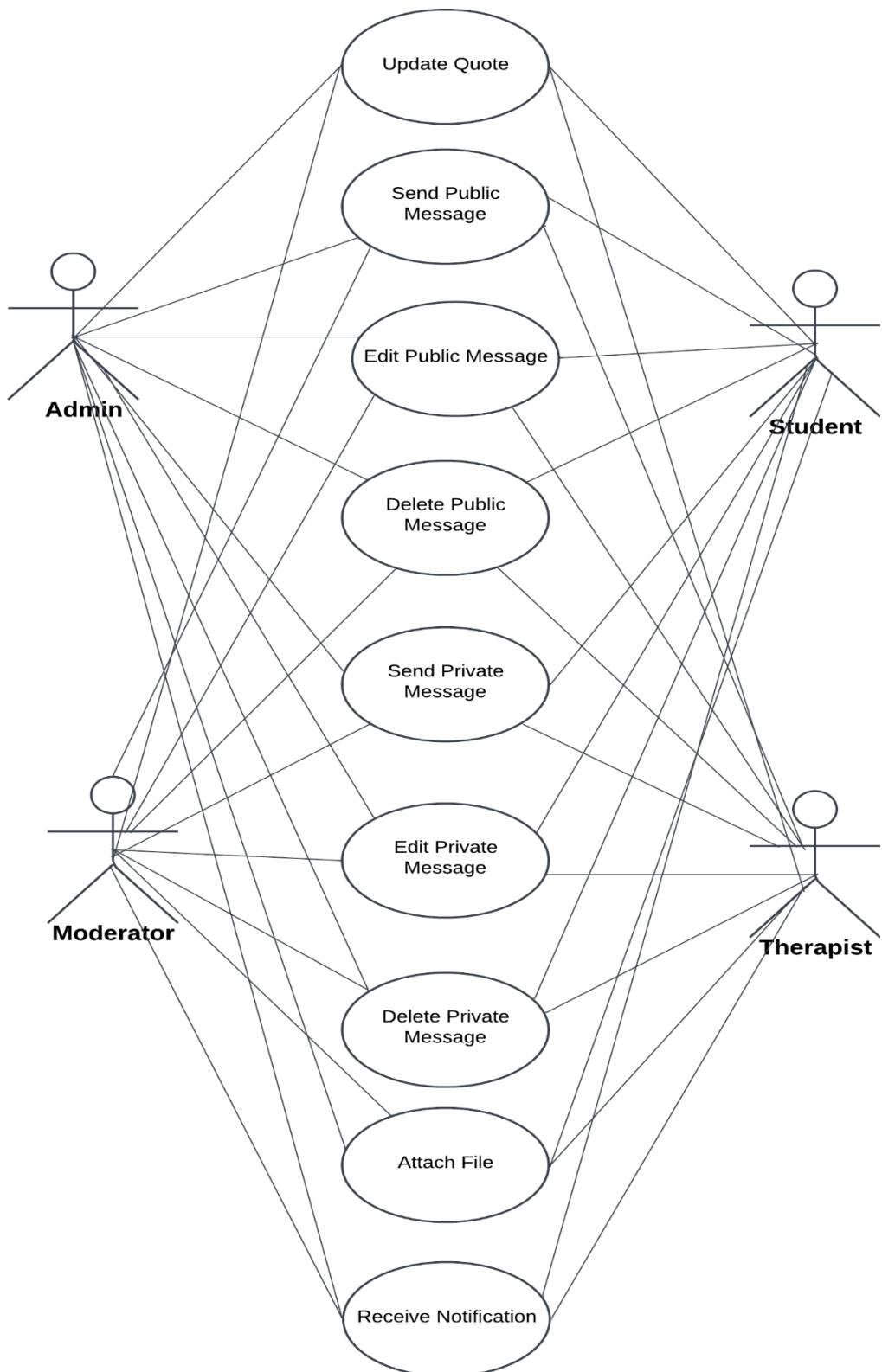


Figure 4-5: use case diagram for common cases between users part 2

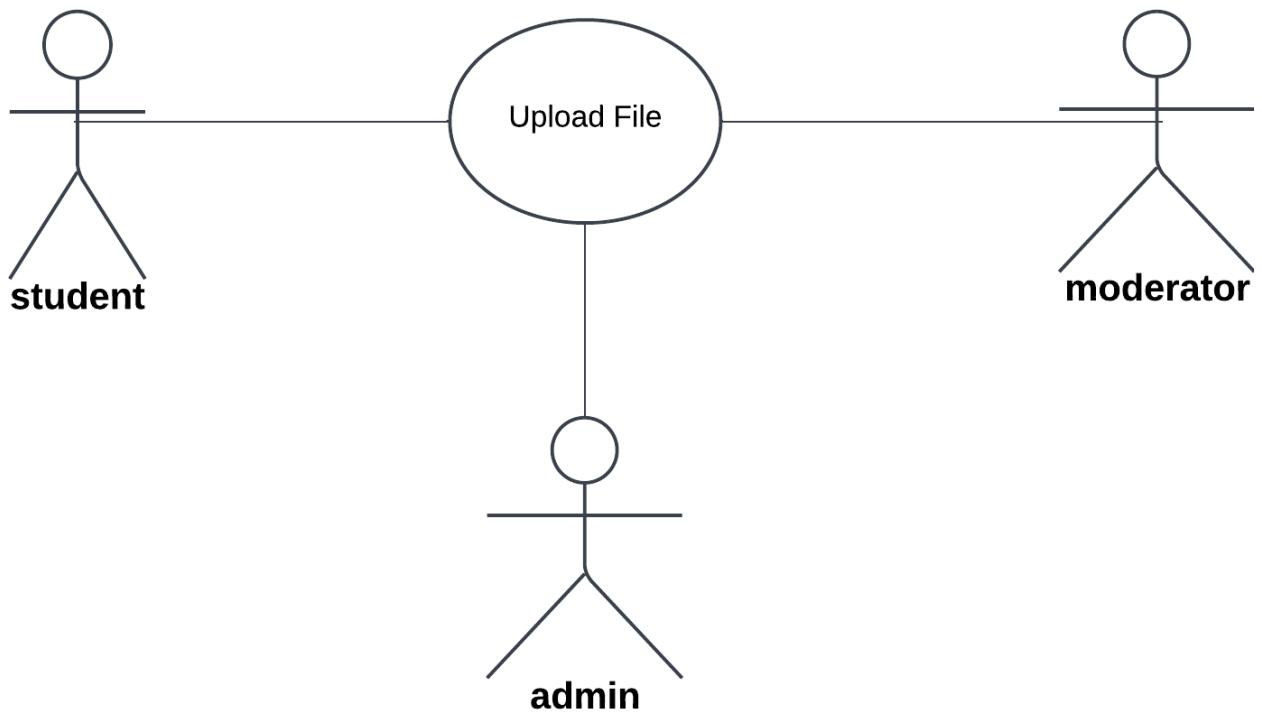


Figure 4-6: common admin, moderator, student, use case diagram

The following diagram represents the use case diagram for the admin use cases only.

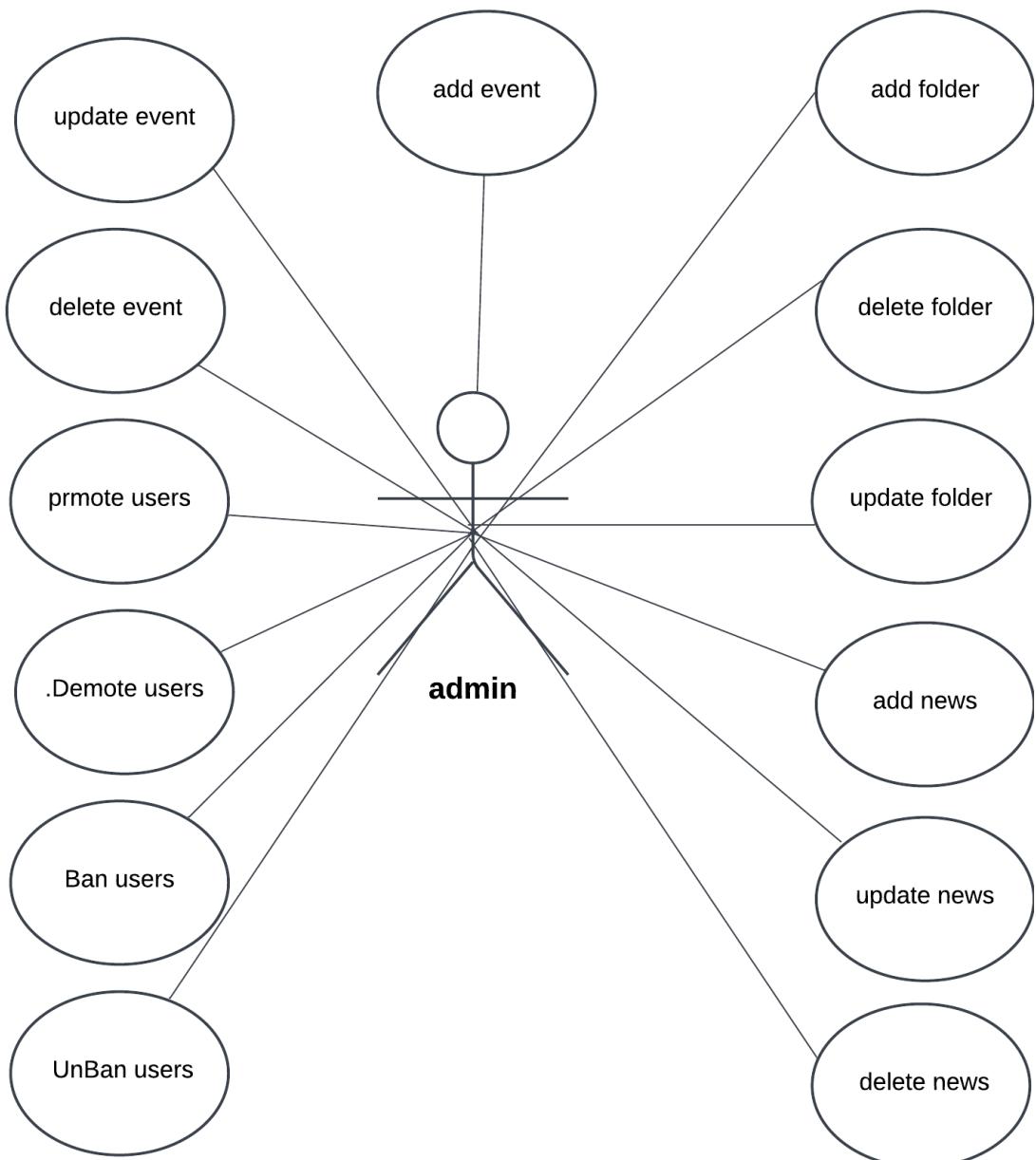


Figure 4-7: use case diagram for admin only cases

The following diagram represents the use case diagram for moderator use cases and common moderator and admin use cases.

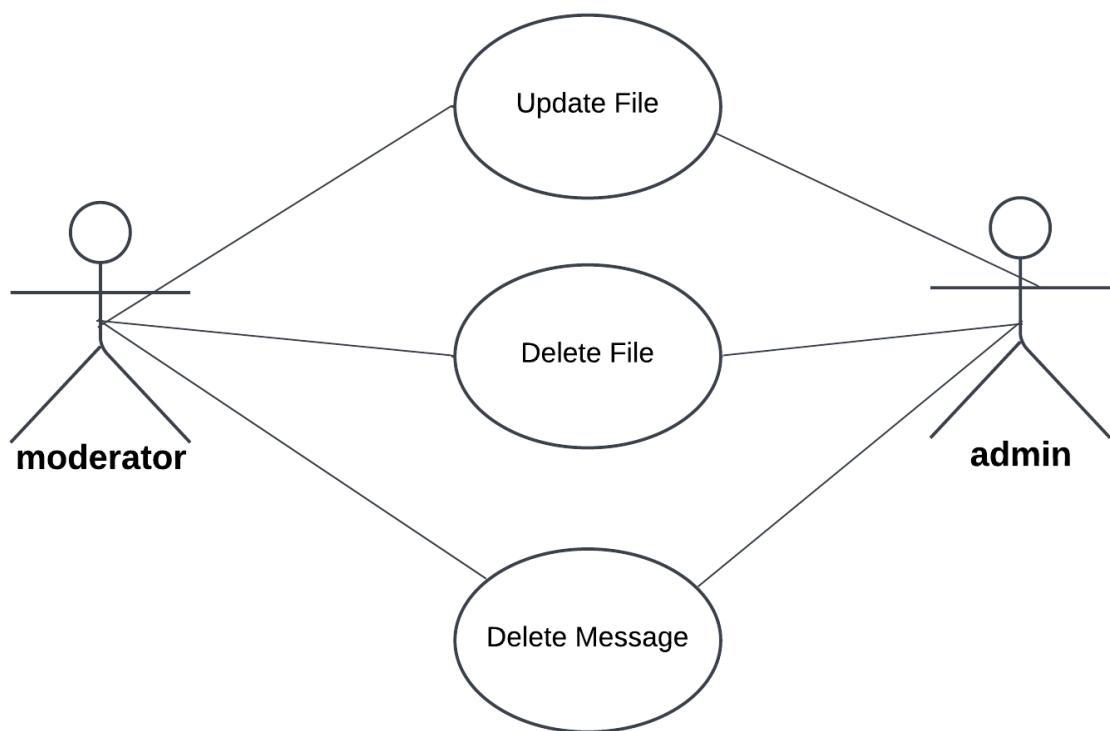


Figure 4-8: use case diagram for common mod and admin cases AND moderator cases

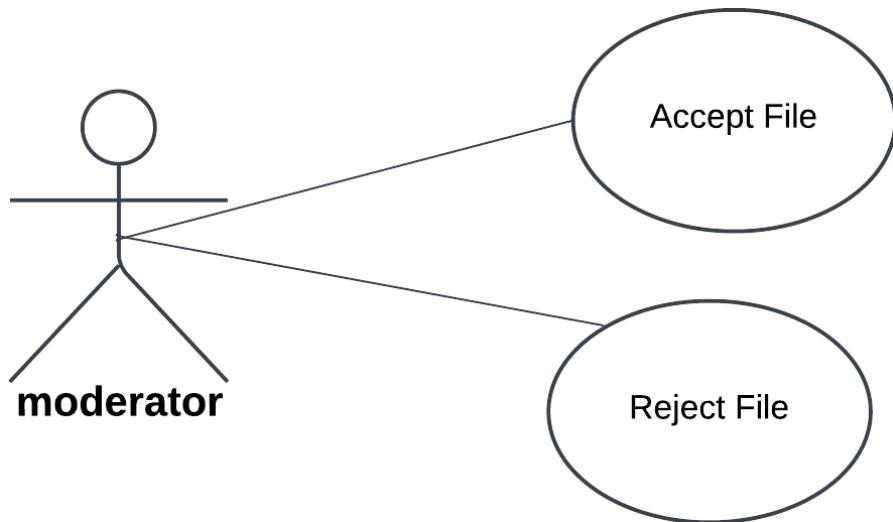


Figure 4-10: moderator use case diagram

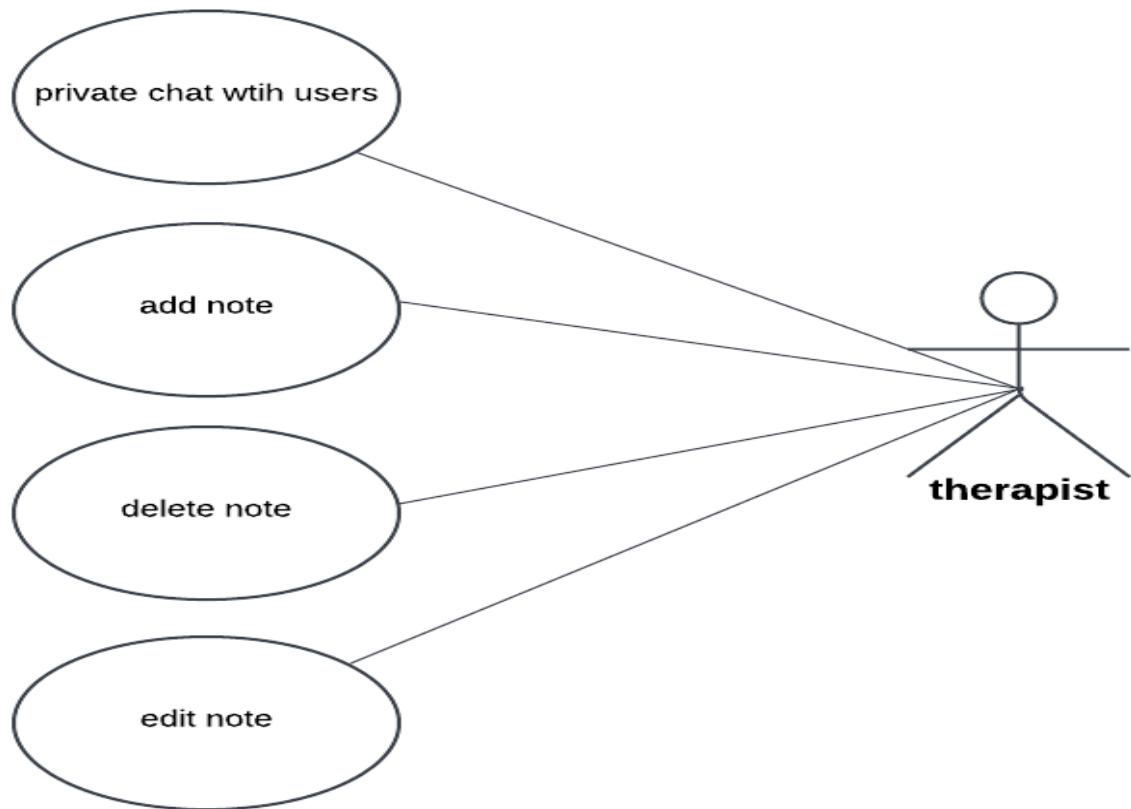


Figure 4-9: therapist use case diagram

## Sequence diagram

The following diagram represents the login sequence diagram.

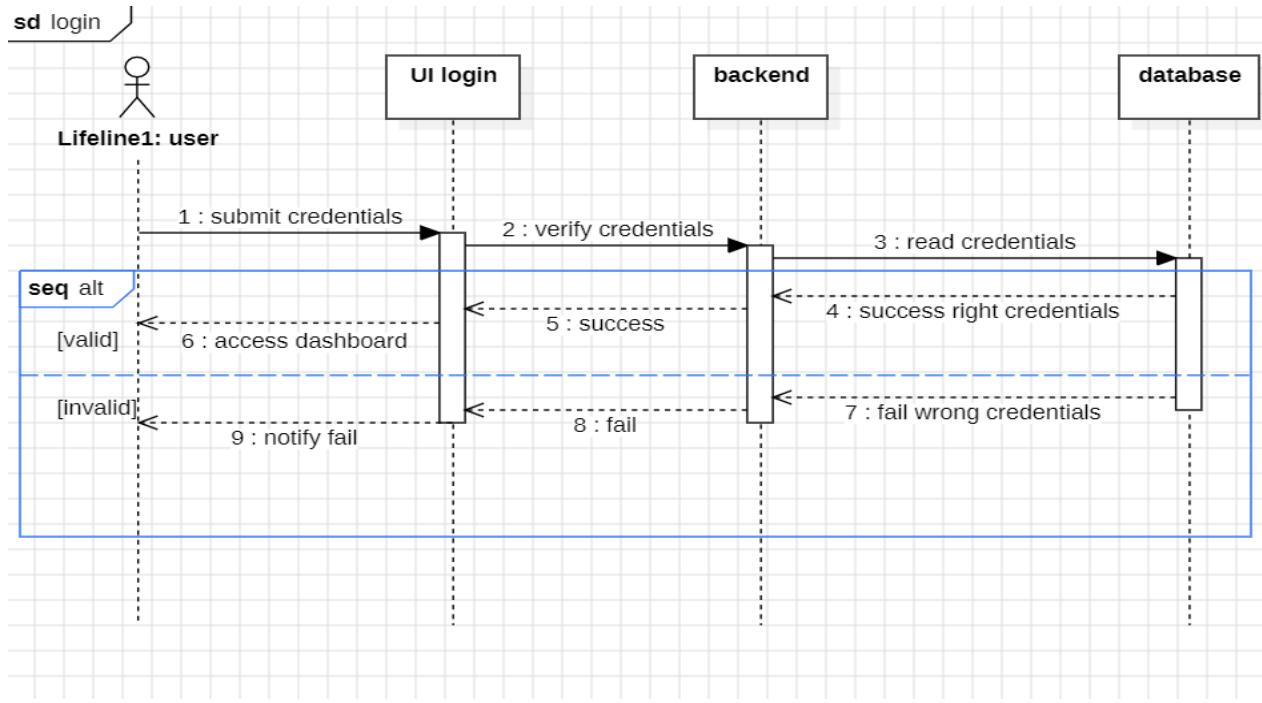


Figure 4-11: login sequence diagram

The following diagram represents

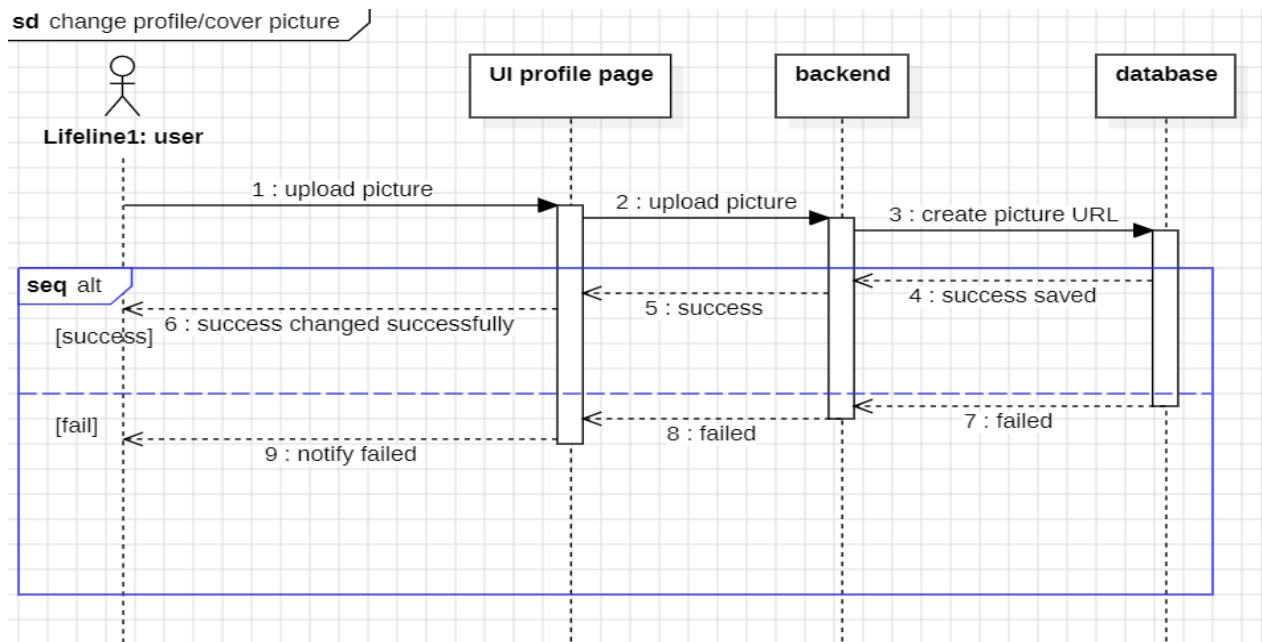


Figure 4-12:upload profile picture sequence diagram

The following diagram represents the sending of public message sequence diagram.

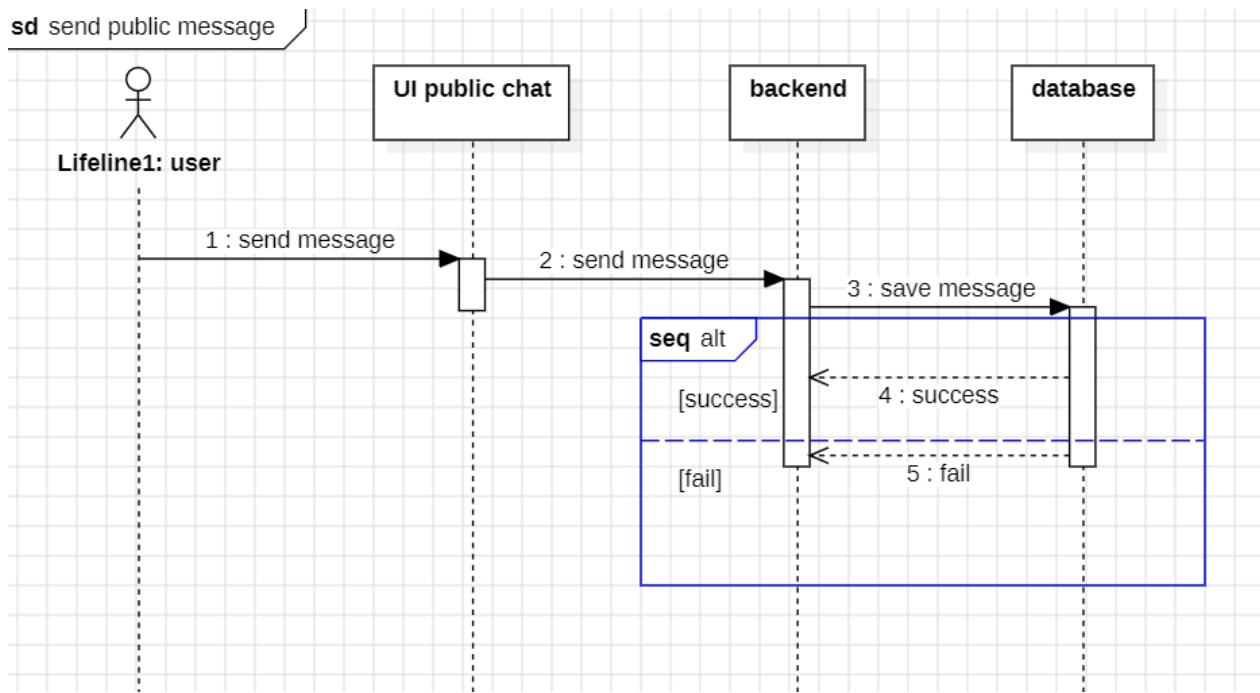


Figure 4-13: send public message sequence diagram

The following diagram represents send private message sequence diagram

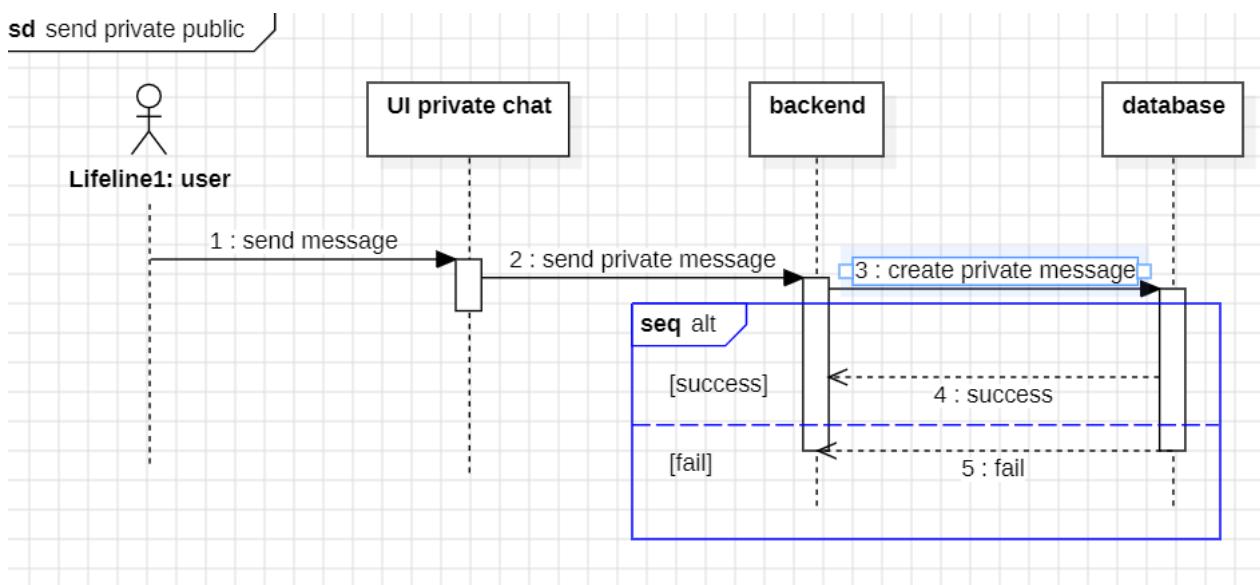


Figure 4-14: send private message sequence diagram

The follow diagram represents the upload file sequence diagram

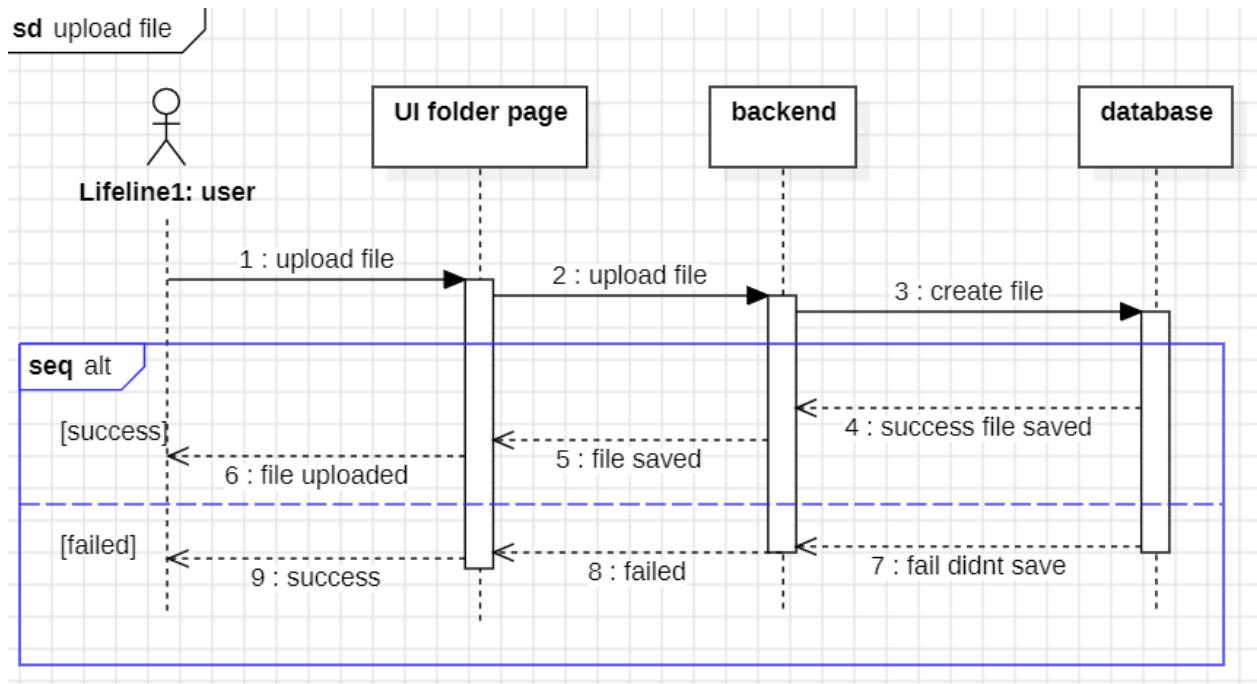


Figure 4-15: upload file sequence diagram

The following diagram represents the add news sequence diagram

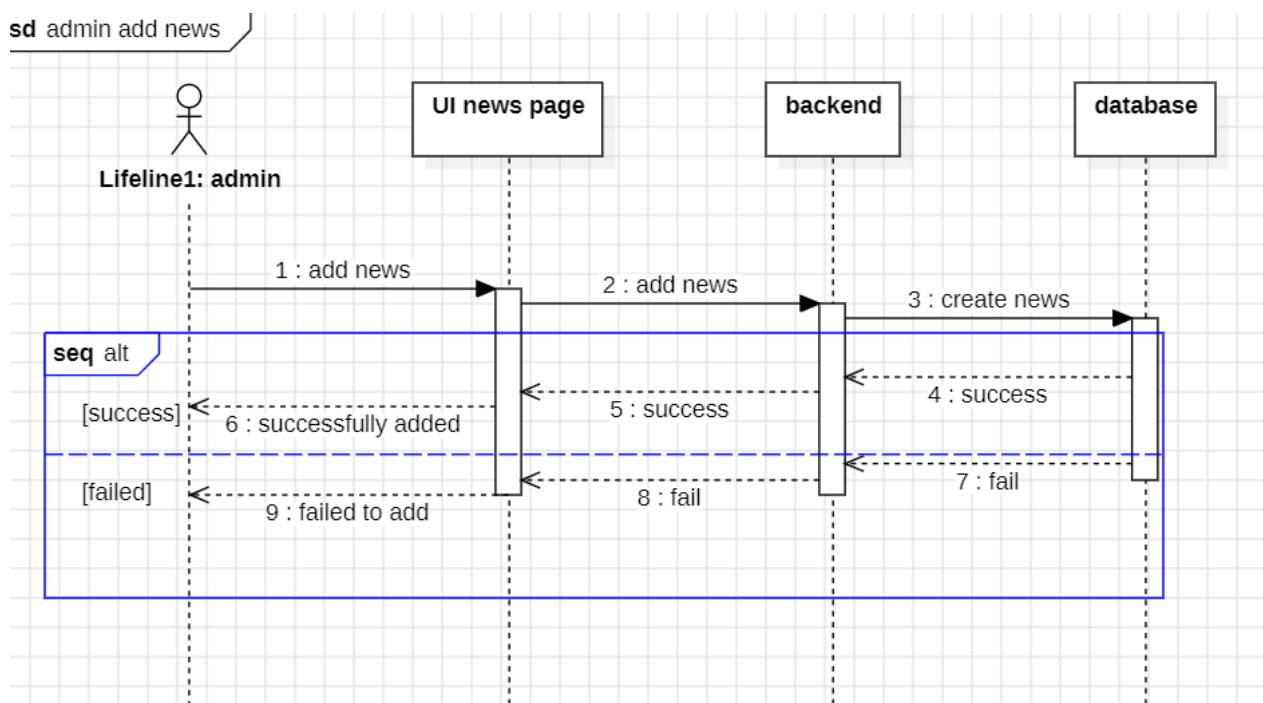


Figure 4-16: add news sequence diagram

The following diagram represents the change roles sequence diagram

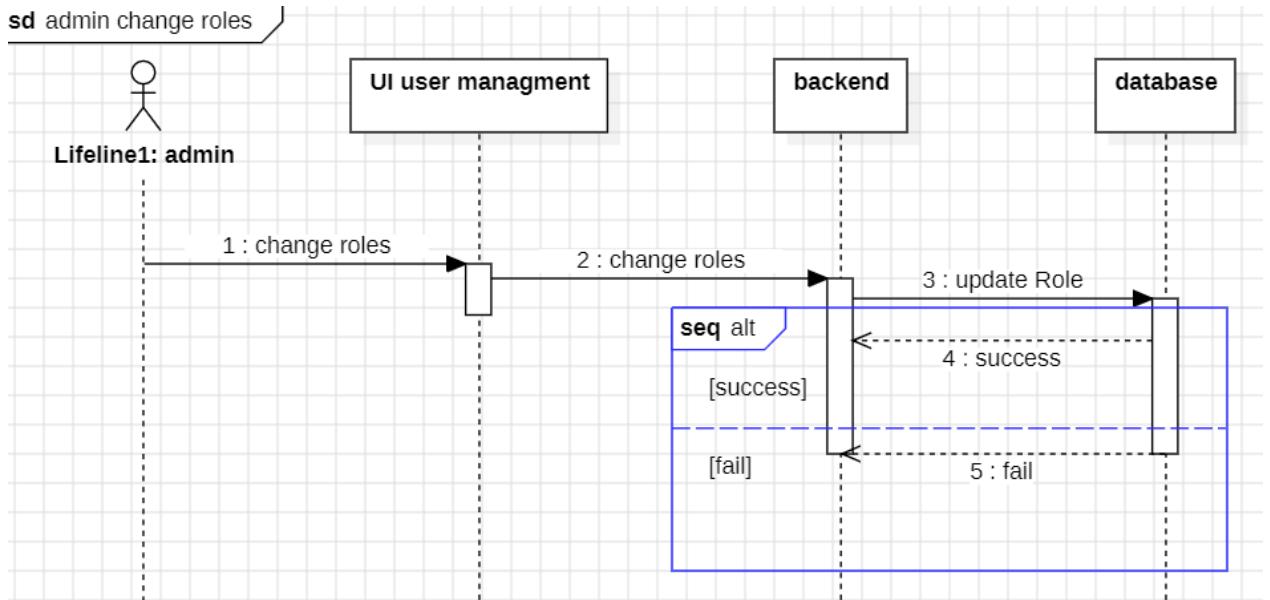


Figure 4-17: change roles sequence diagram

The following diagram represents the delete folder sequence diagram.

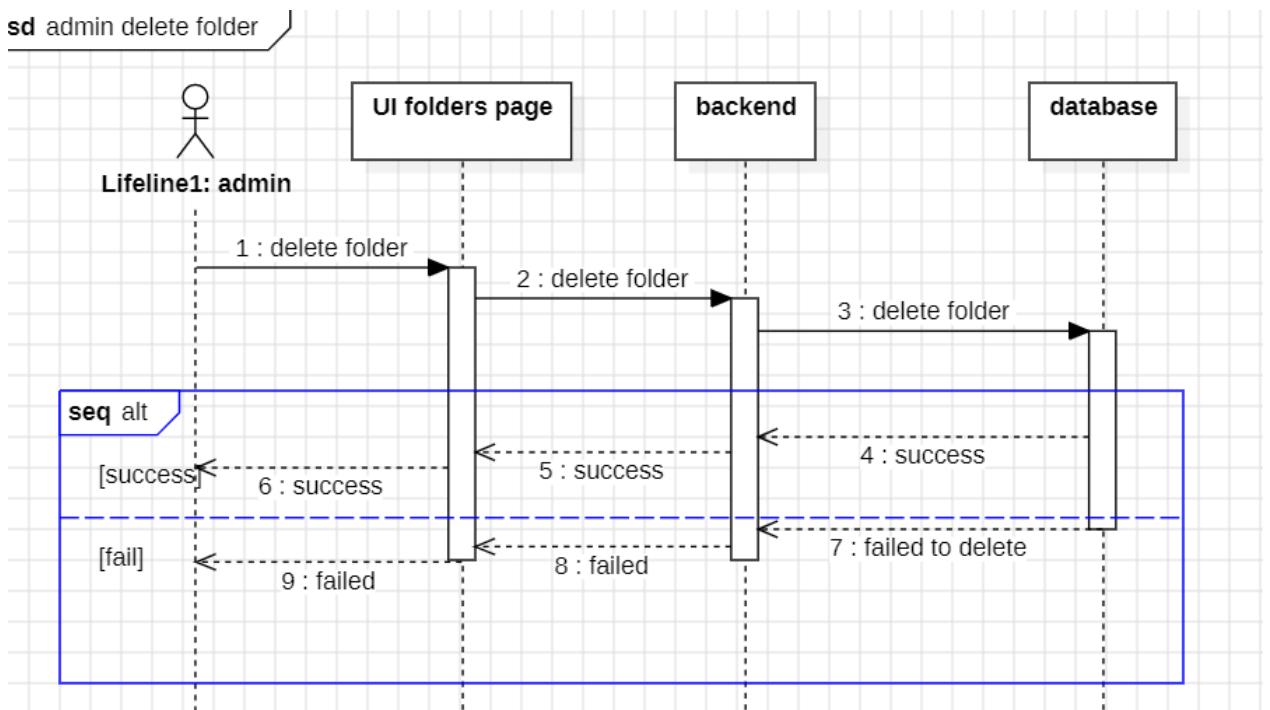


Figure 4-18: delete folder sequence diagram

The following diagram represents the change bio sequence diagram.

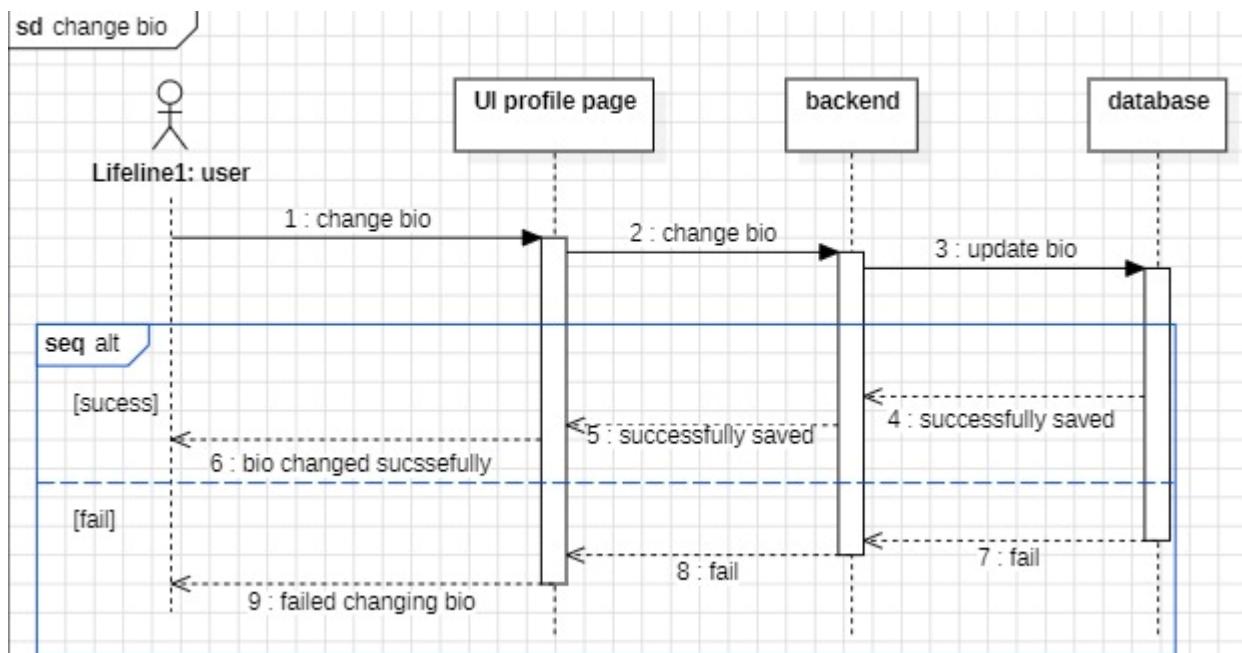


Figure 4-19: change bio sequence diagram

The following diagram represents the deleted personal files

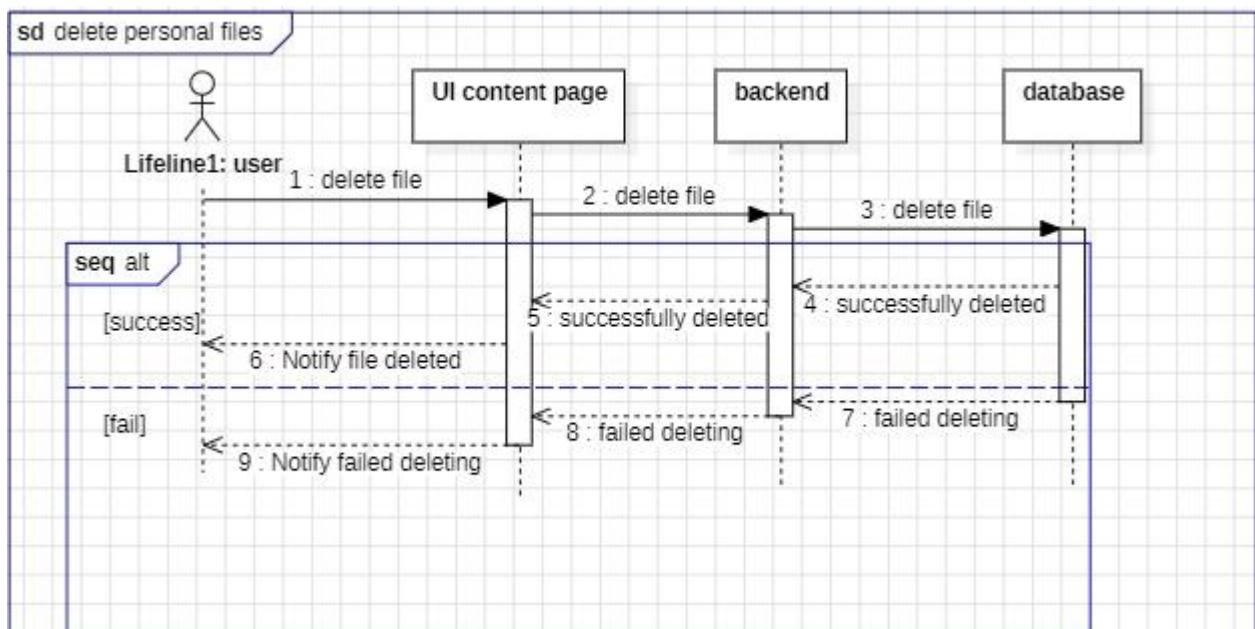


Figure 4-20: delete file sequence diagram

The following diagram represents the delete news sequence diagram

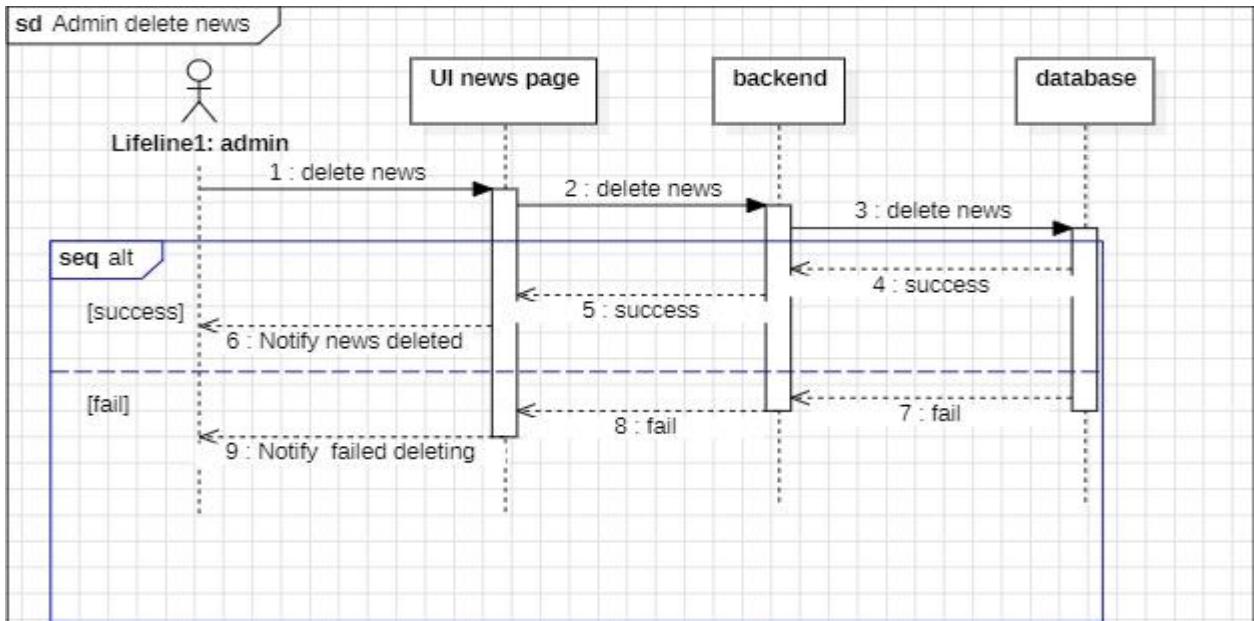


Figure 4-21: delete news sequence diagram

The following diagram represents the add folder sequence diagram

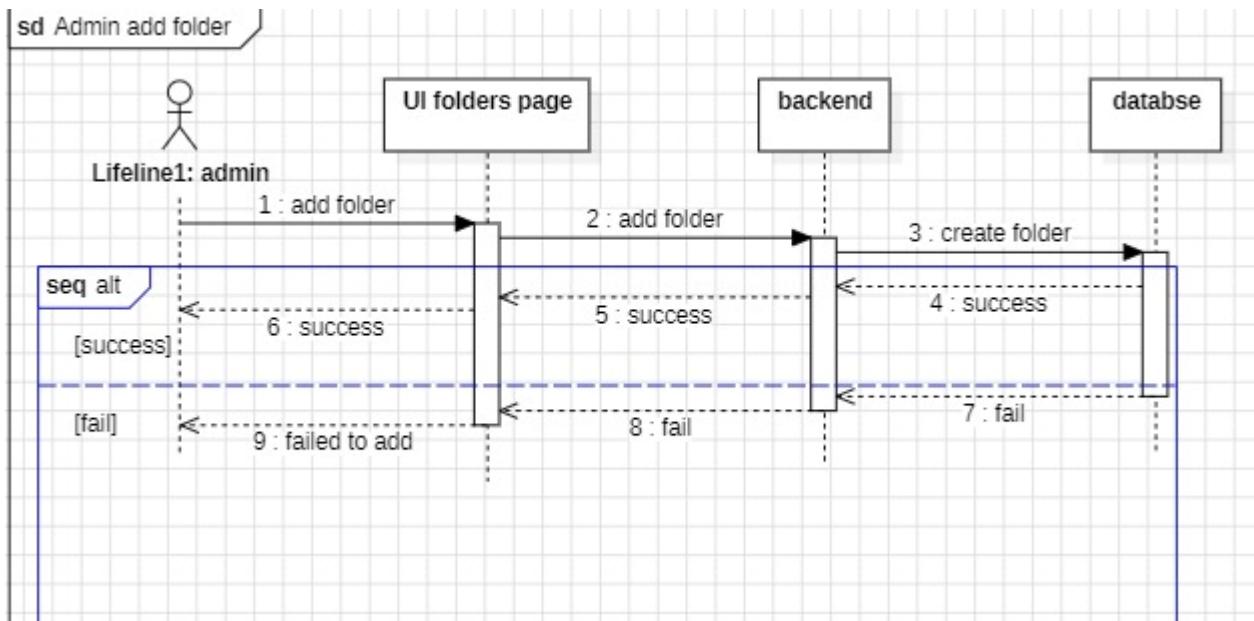


Figure 4-22: add folder sequence diagram

The following diagram represents the accept/reject file sequence diagram

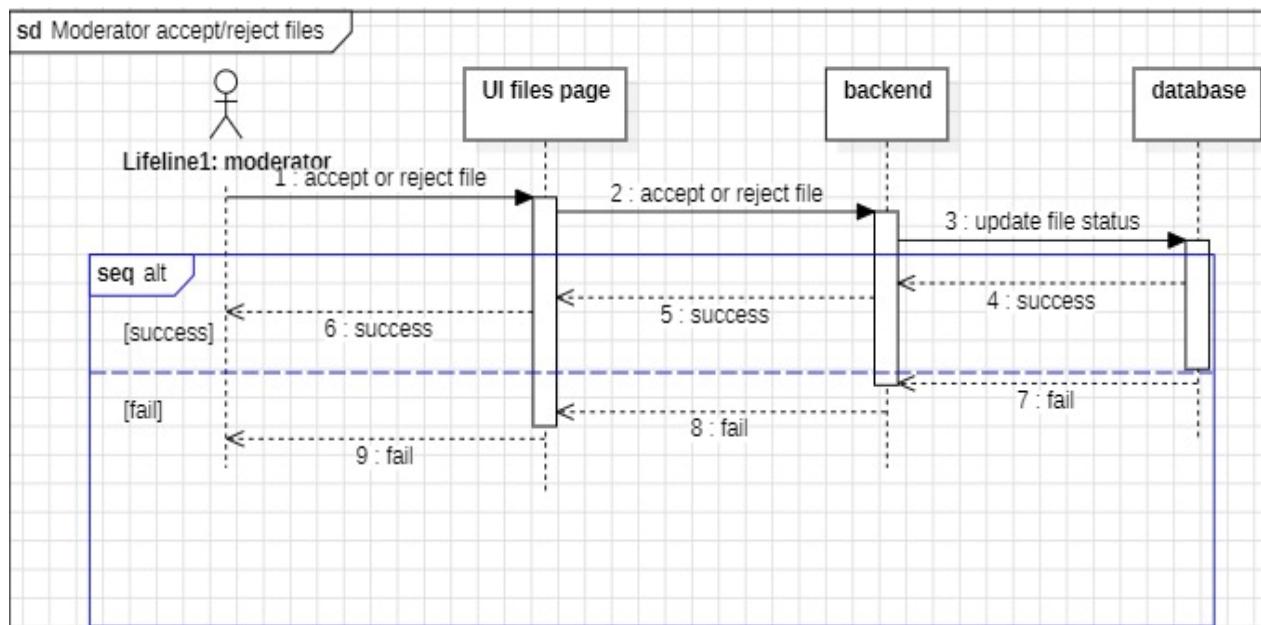


Figure 4-23: accept or reject file sequence diagram

The following diagram represents the delete file sequence diagram

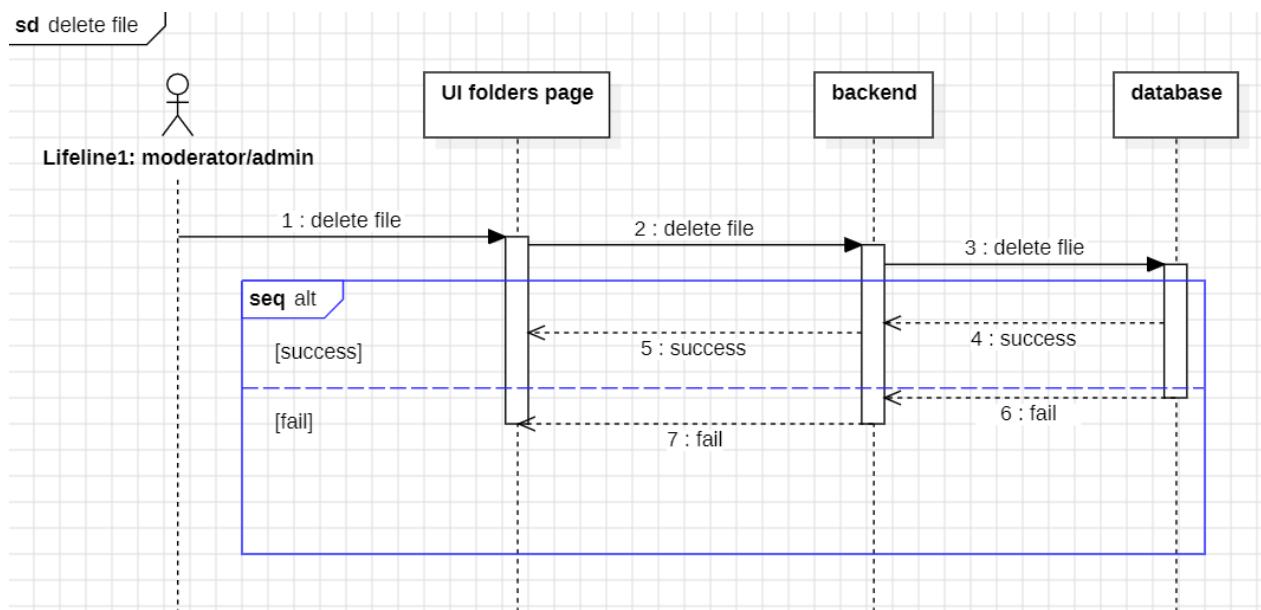


Figure 4-24: delete file sequence diagram

The following diagram represents the delete events sequence diagram

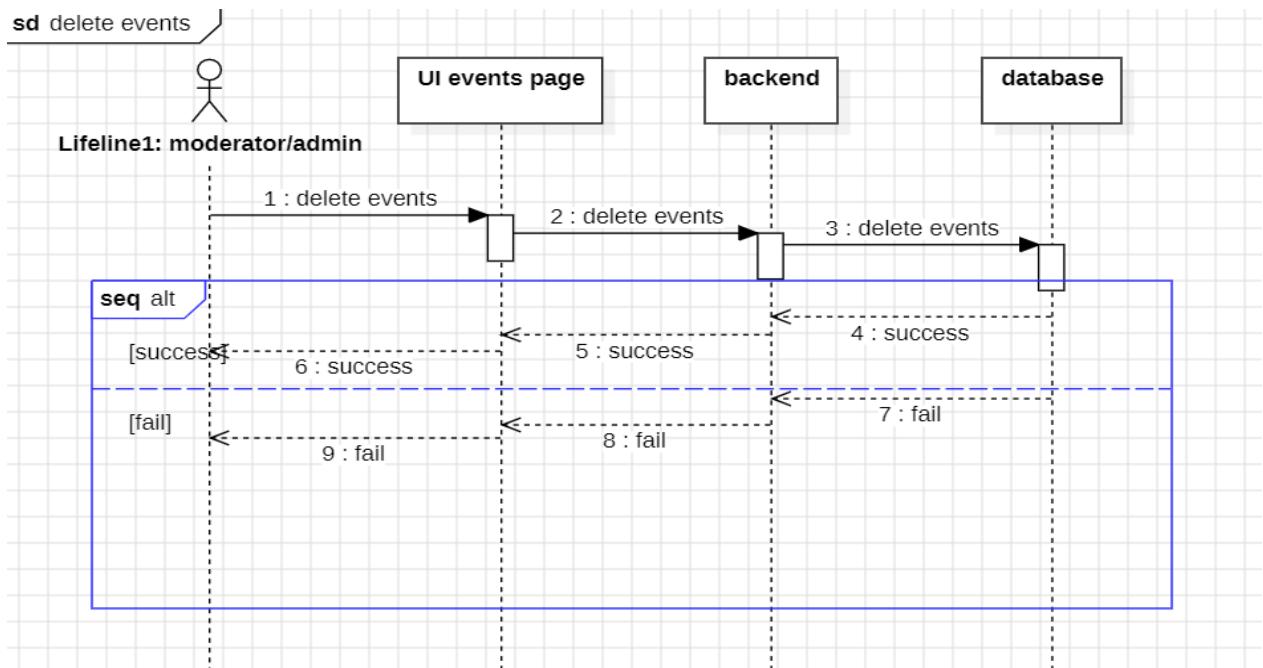


Figure 4-25: delete events sequence diagram

The following diagram represents the admin/moderator delete messages

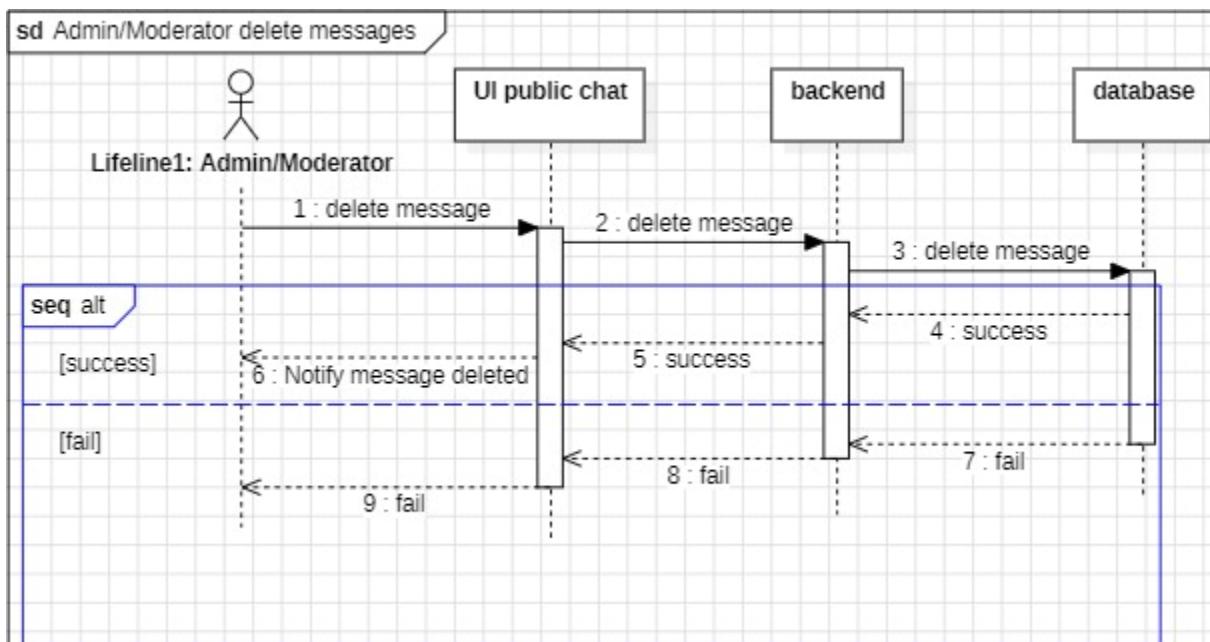


Figure 4-26: delete message sequence diagram

The following diagram represents the check task process

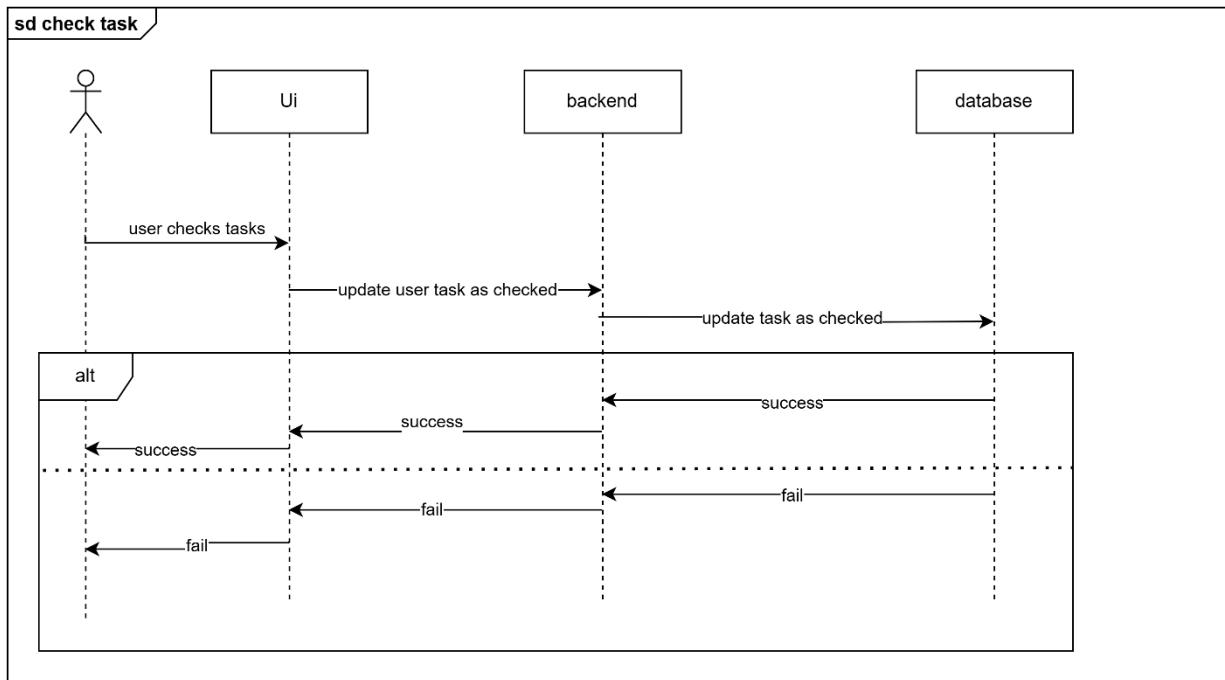


Figure 4-27: check task sequence diagram

The following diagram represents the add task process

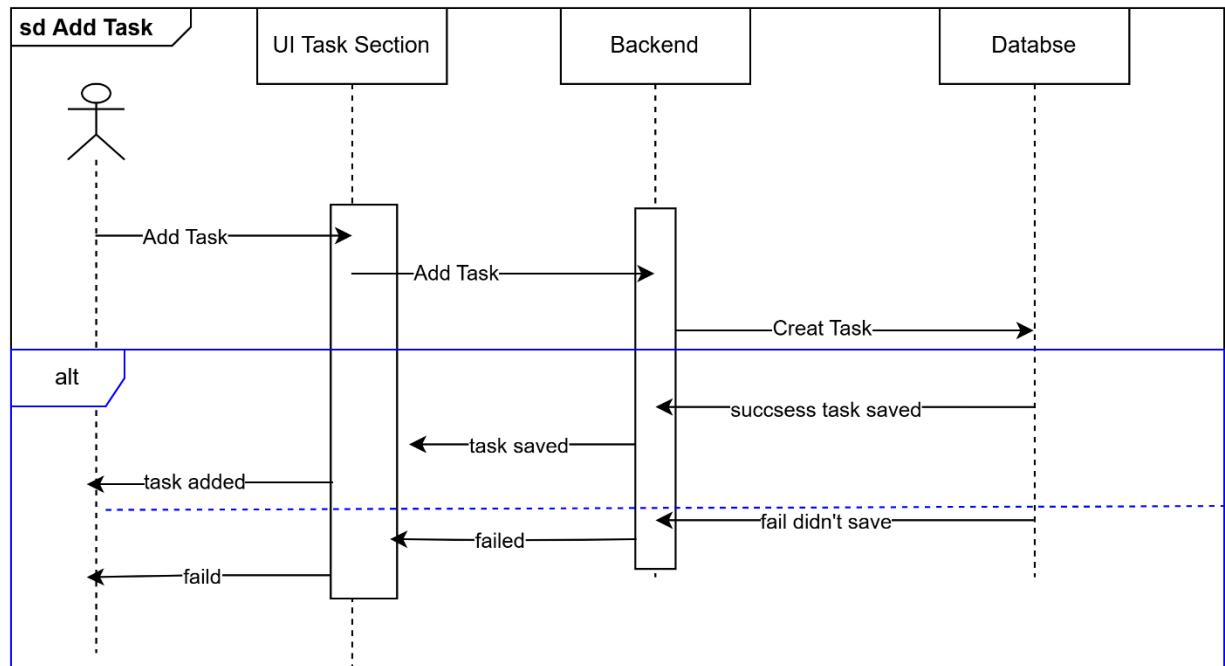


Figure 4-28: add task sequence diagram

The following diagram shows the update quote

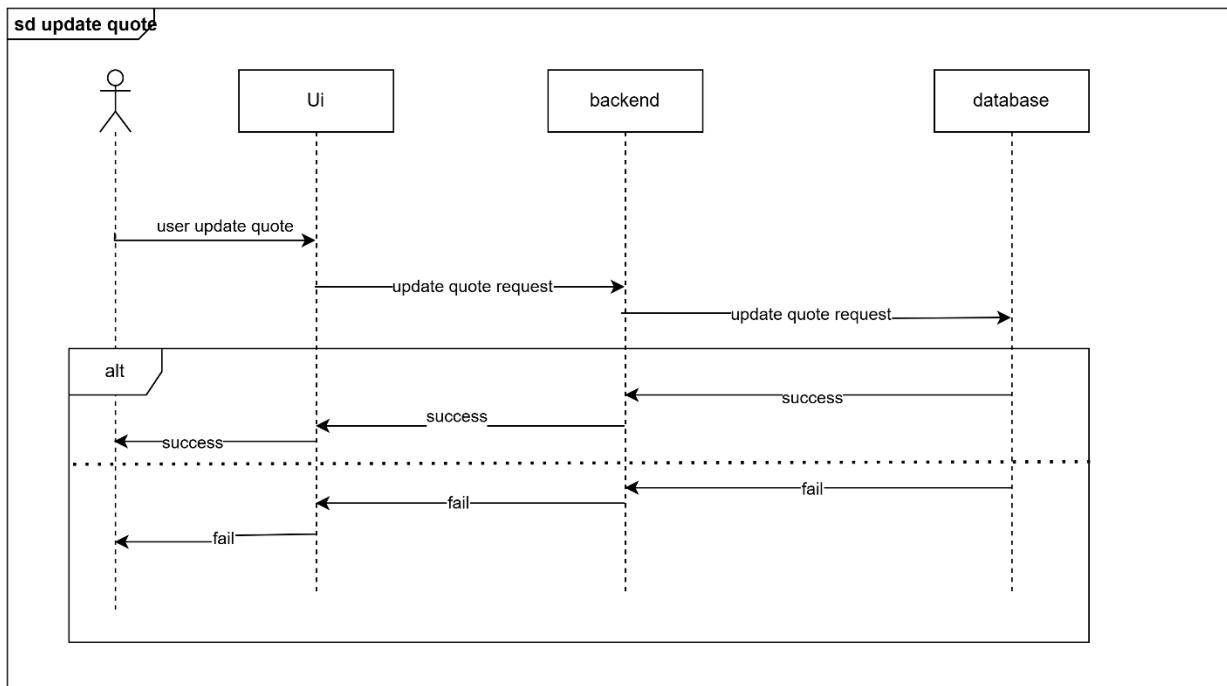


Figure 4-29: update quote sequence diagram

The following diagram shows the delete task sequence diagram

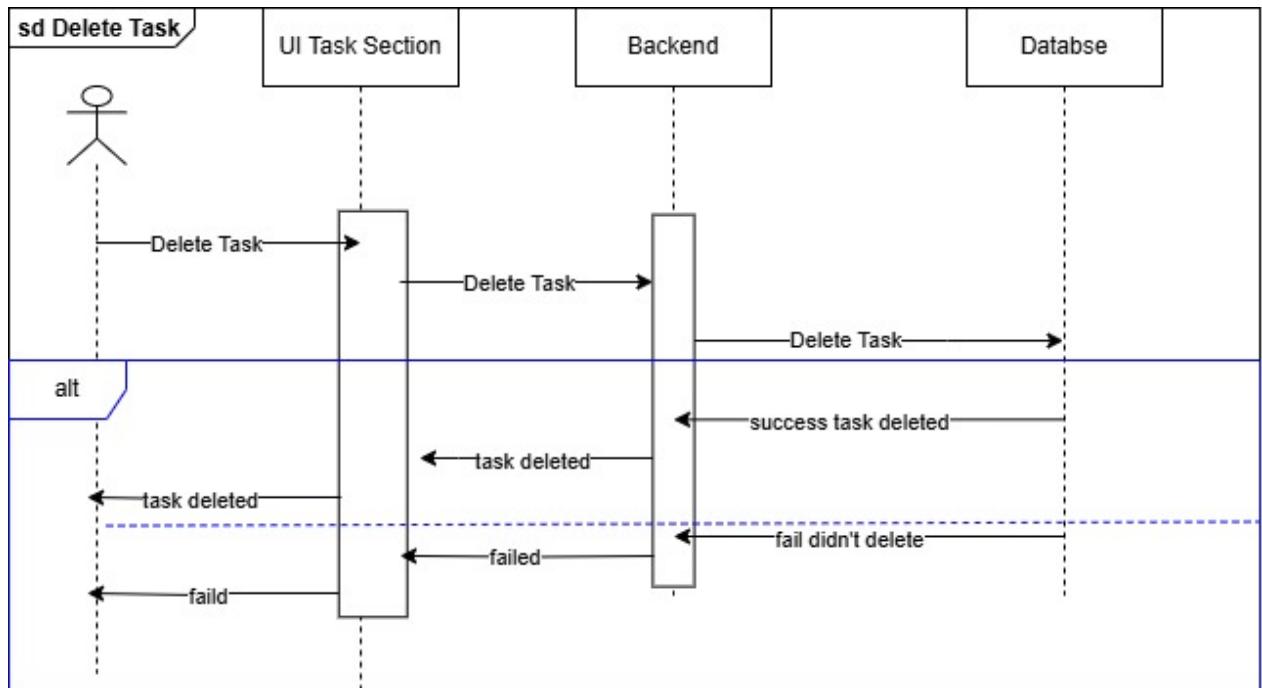


Figure 4-30: delete task sequence diagram

The following figure shows the edit public message sd

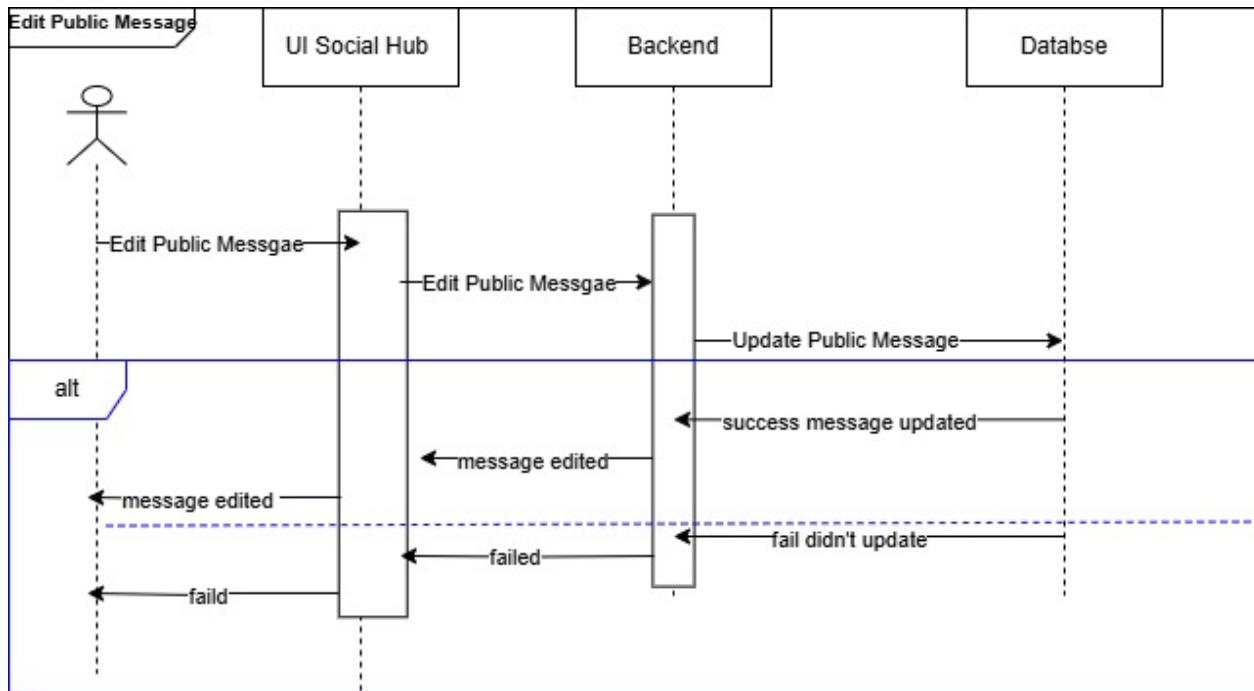


Figure 4-31: edit public message sequence

The following figure shows the attach file sequence diagram

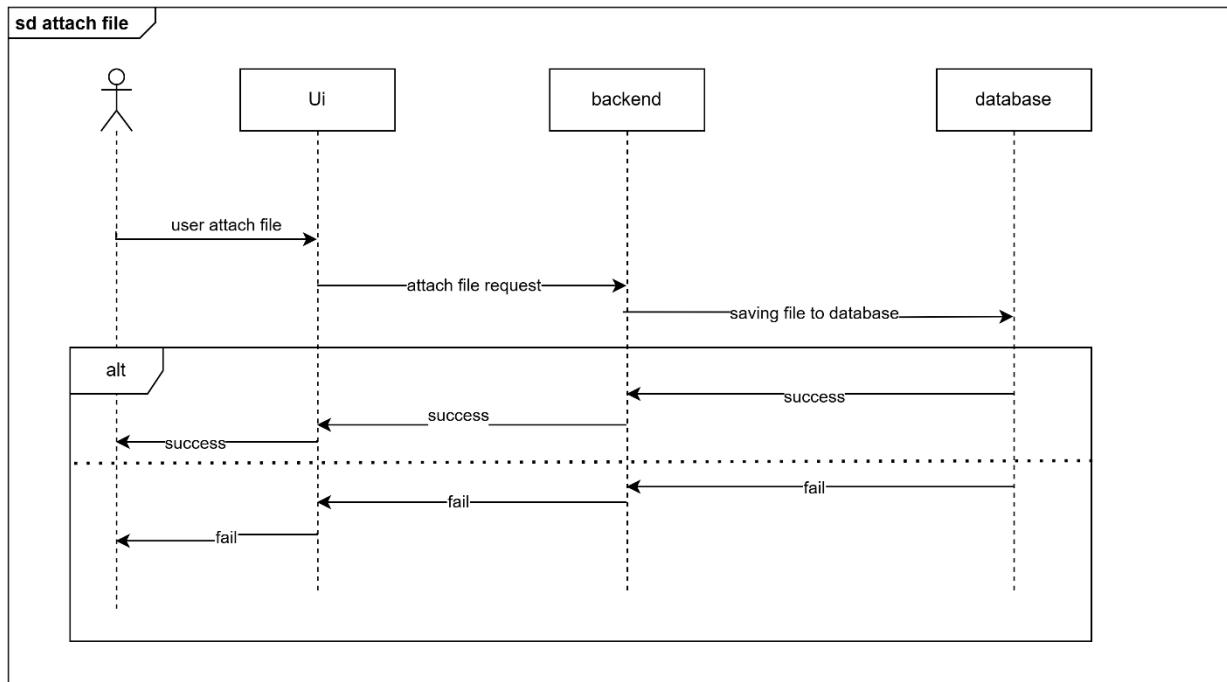


Figure 4-32: attach file sequence diagram

The following figure represents edit message in private chat sd

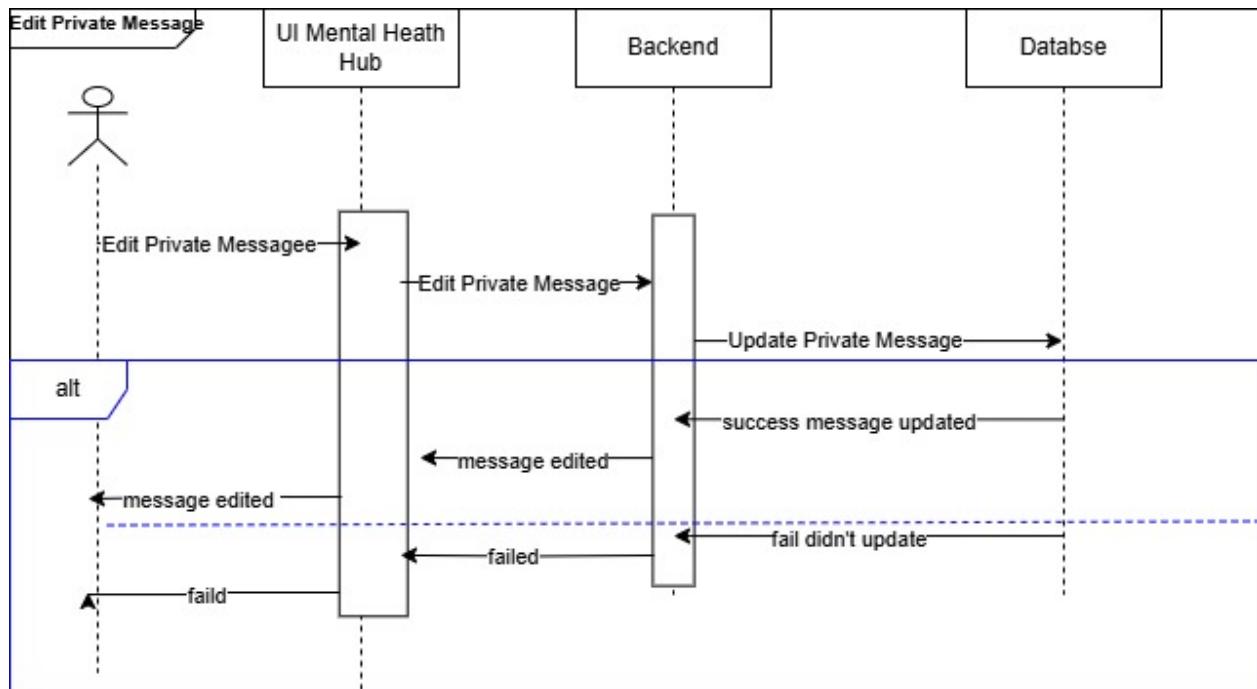


Figure 4-33: edit private message sequence diagram

The following figure represents the deleted public message sequence diagram

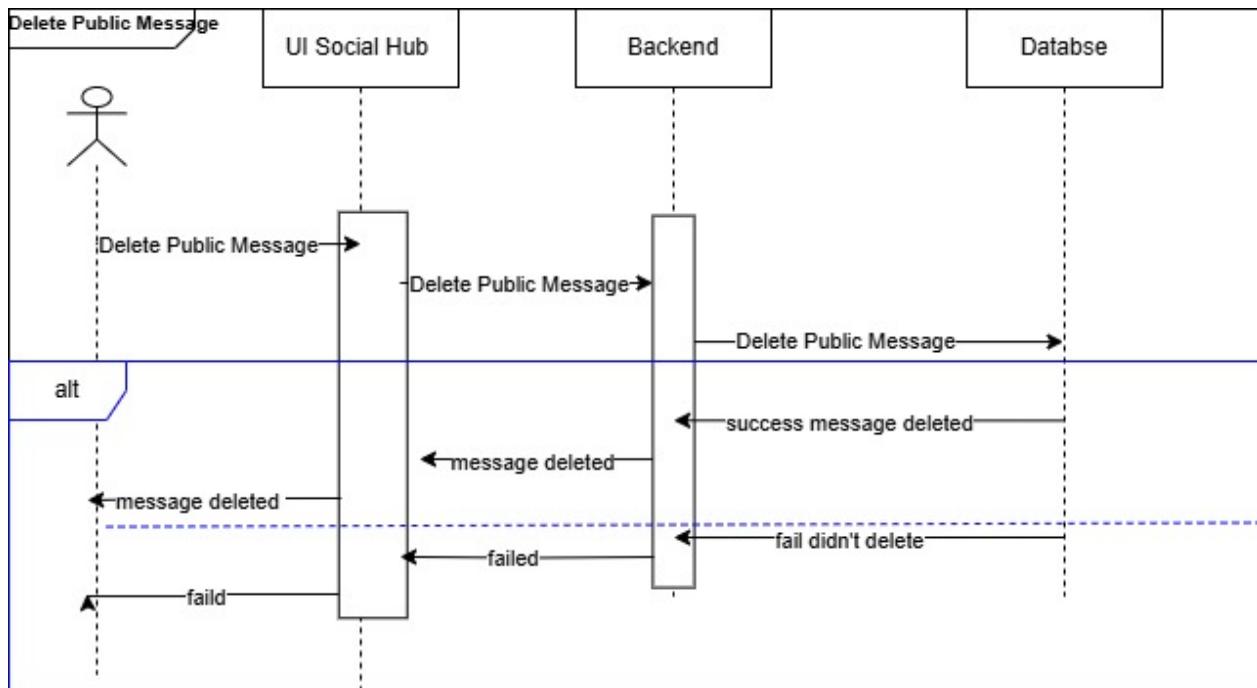


Figure 4-34: delete public message sequence diagram

The following figure represents the deleted private message

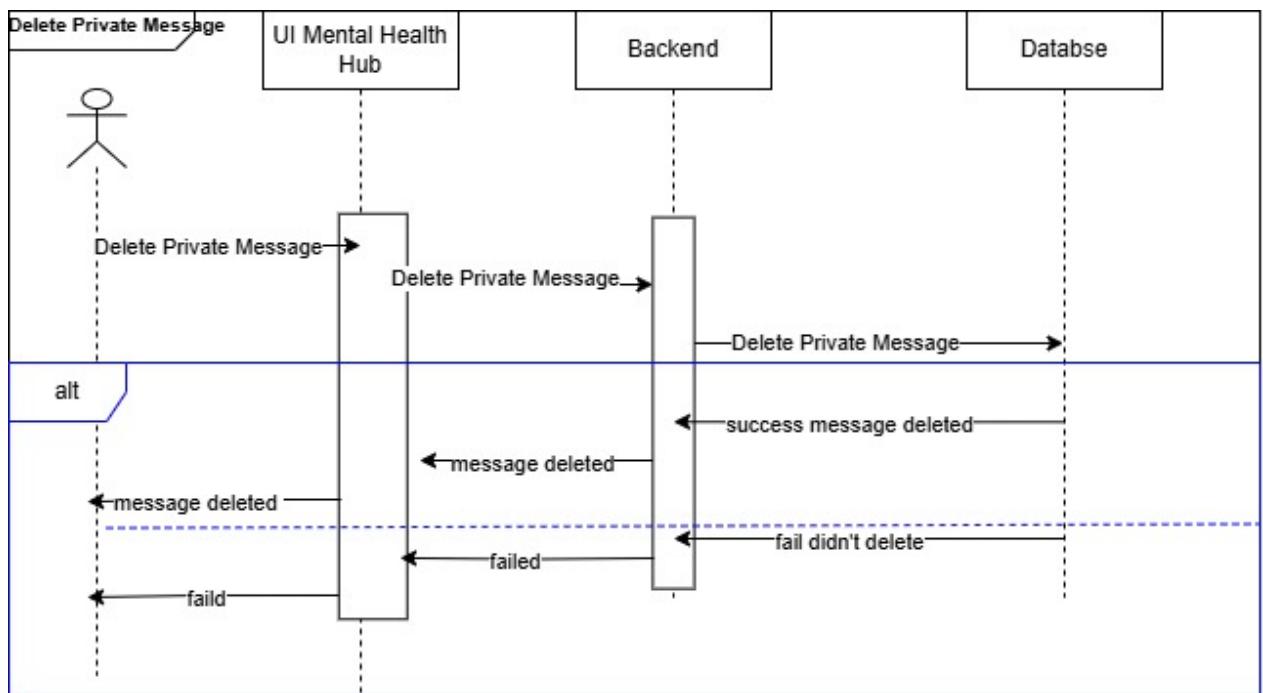


Figure 4-35: delete private message sequence diagram

The following figure represents the update file sequence diagram

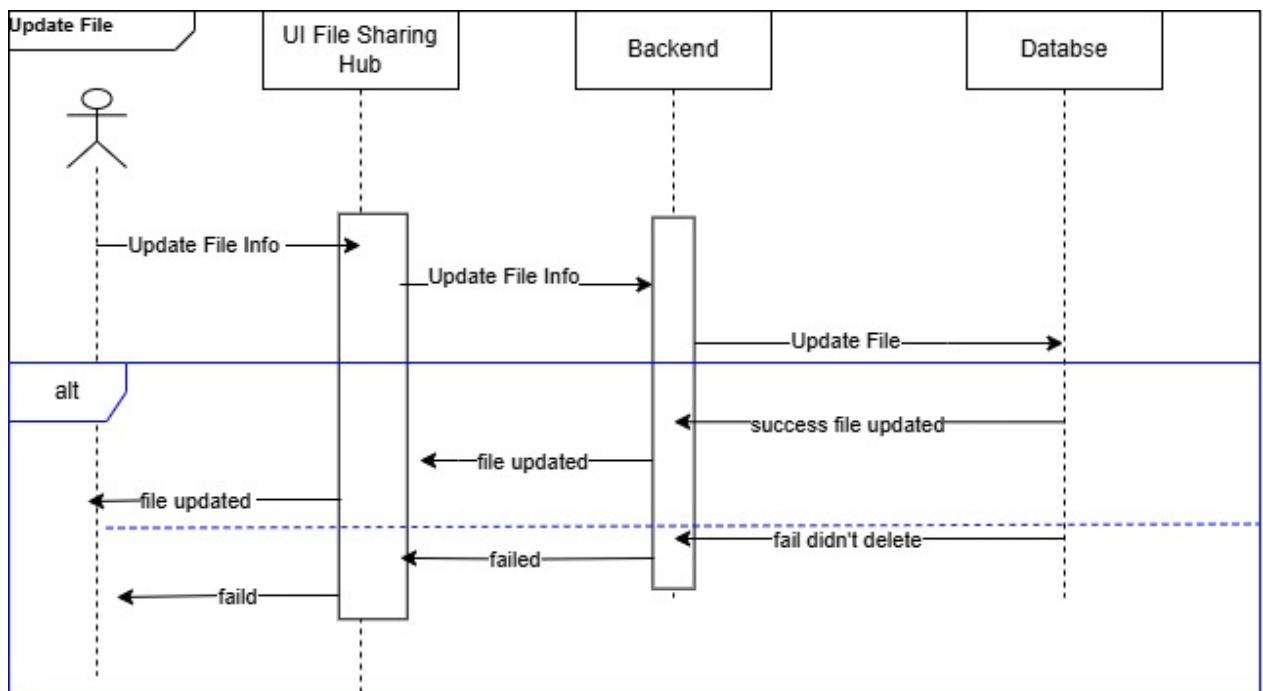


Figure 4-36: update file sequence diagram

The following figure represents the update note sequence diagram

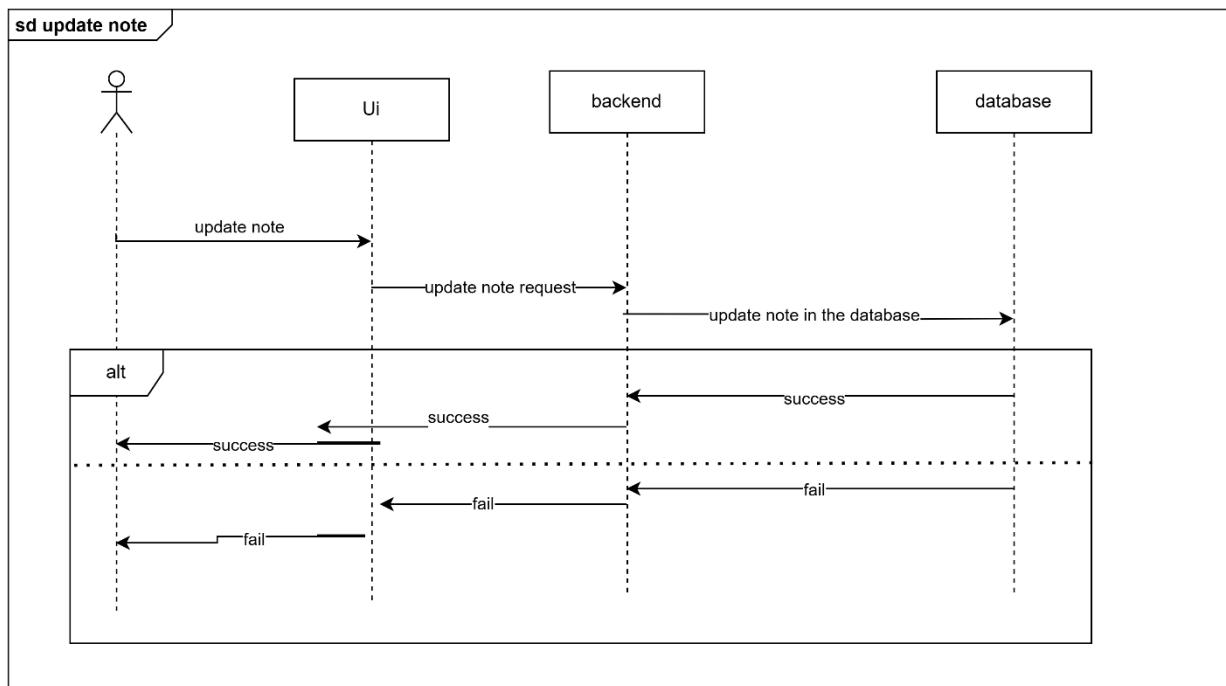


Figure 4-37: update note sequence diagram

The following figure represents the update folder

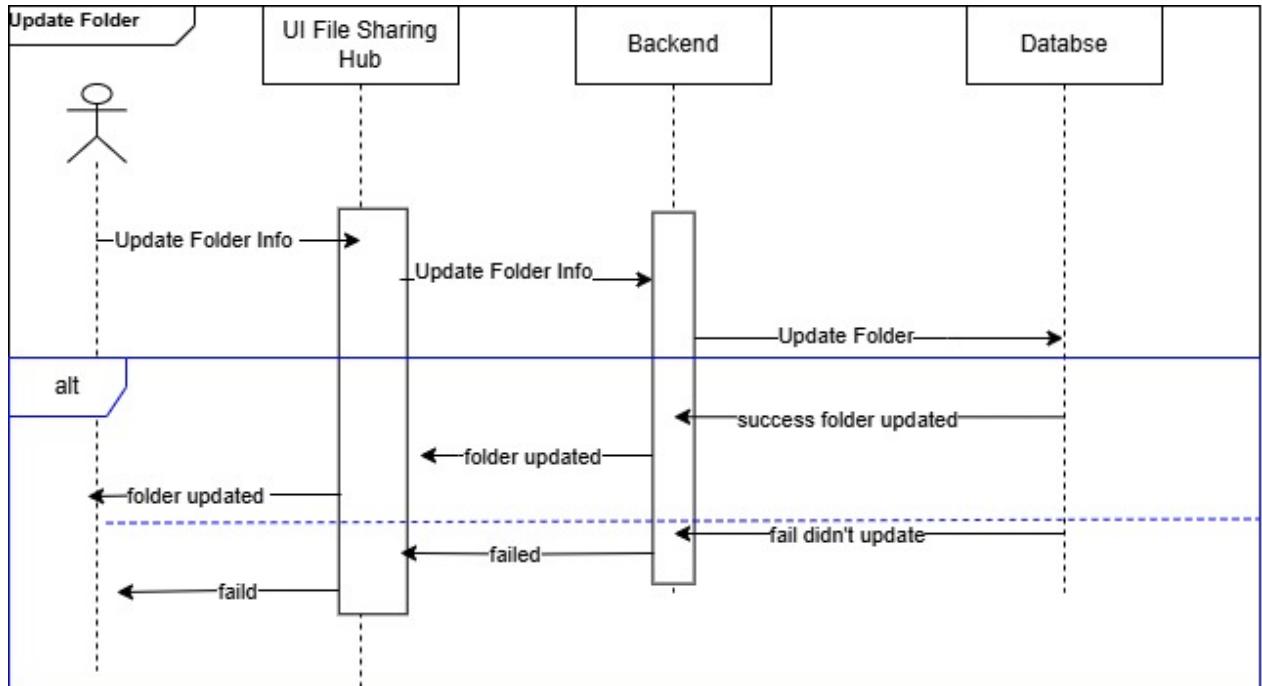


Figure 4-38: update folder sequence diagram

The following figure represents the add note sequence diagram

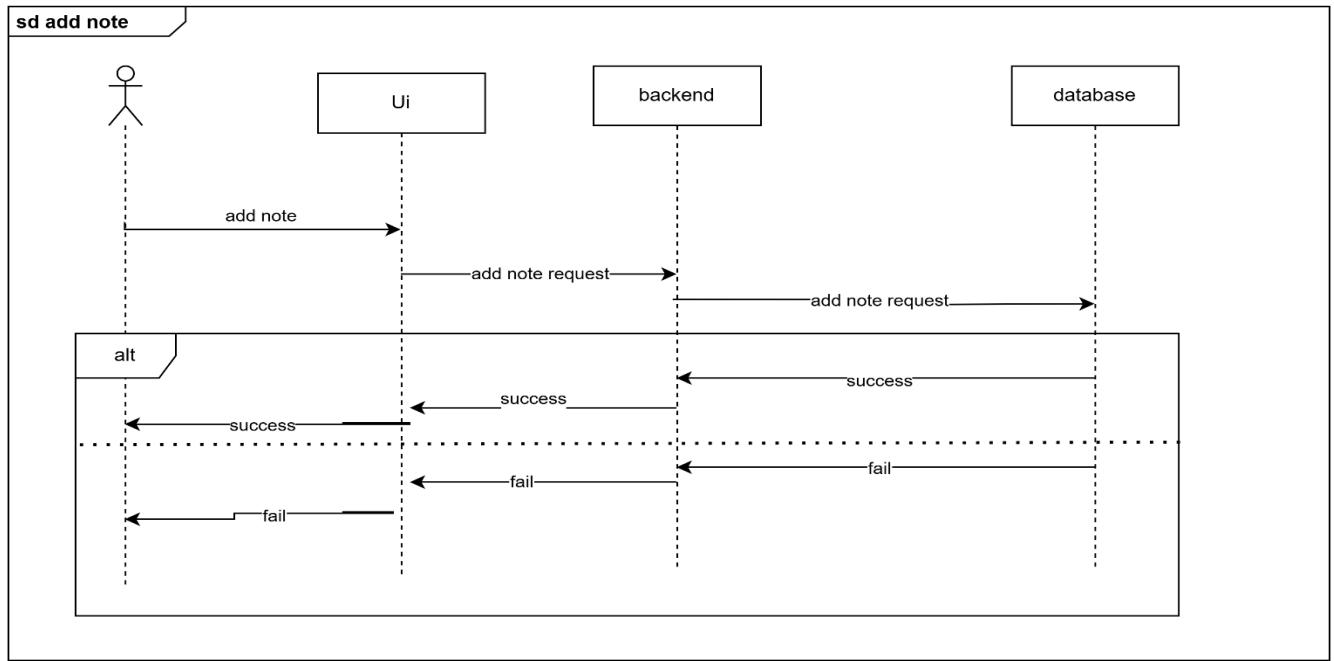


Figure 4-39: add note sequence diagram

The following figure represents the update news

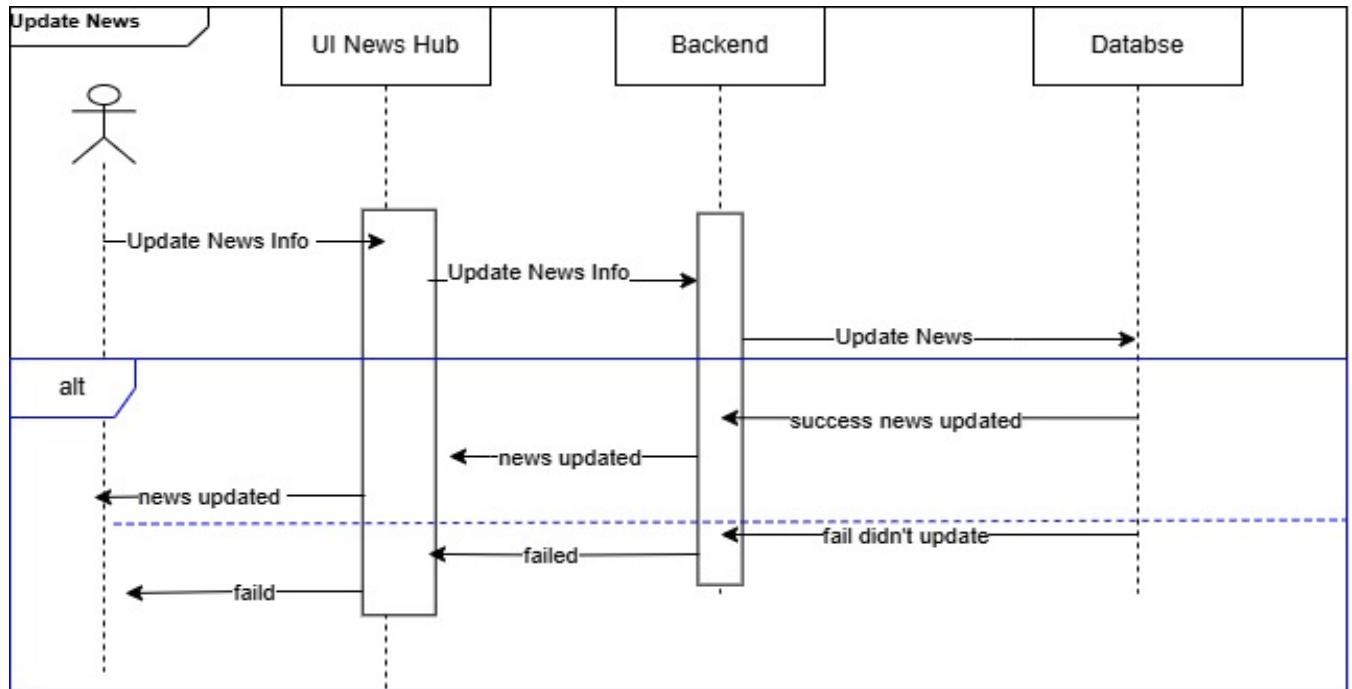


Figure 4-40: update news sequence diagram

The following figure represents the delete note sequence diagram

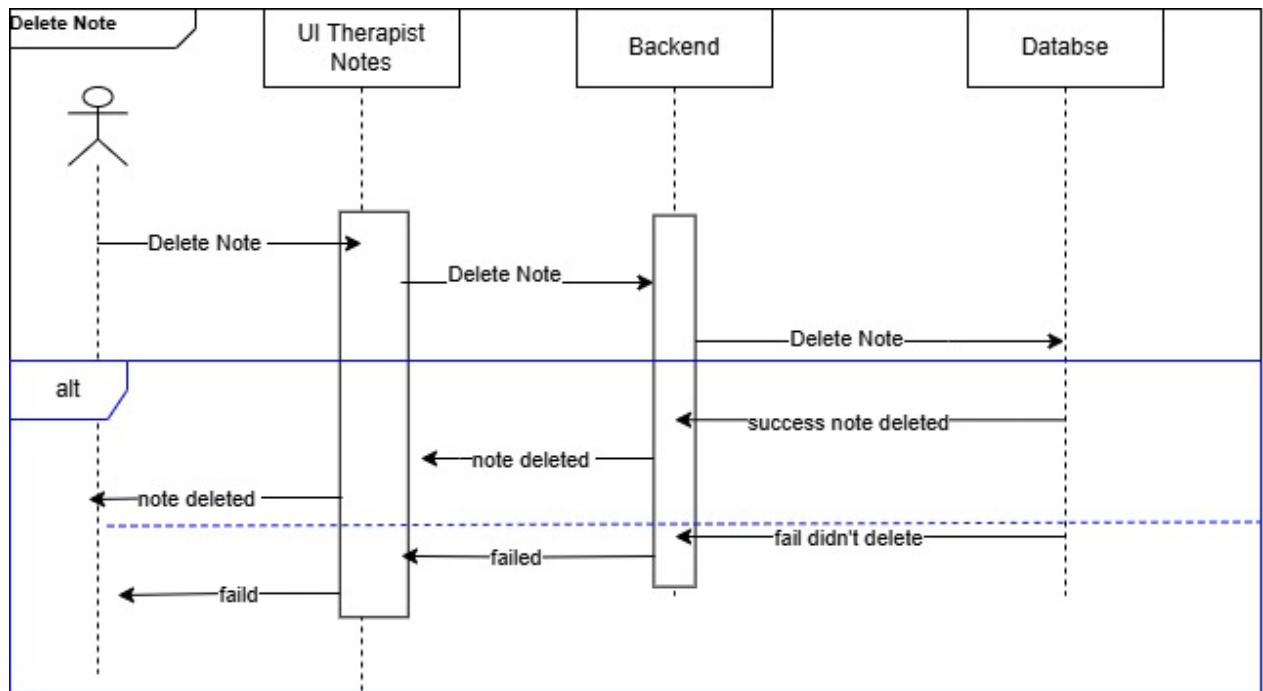


Figure 4-41: delete note sequence diagram

## Class diagram

The following diagram represents the class diagram for our system

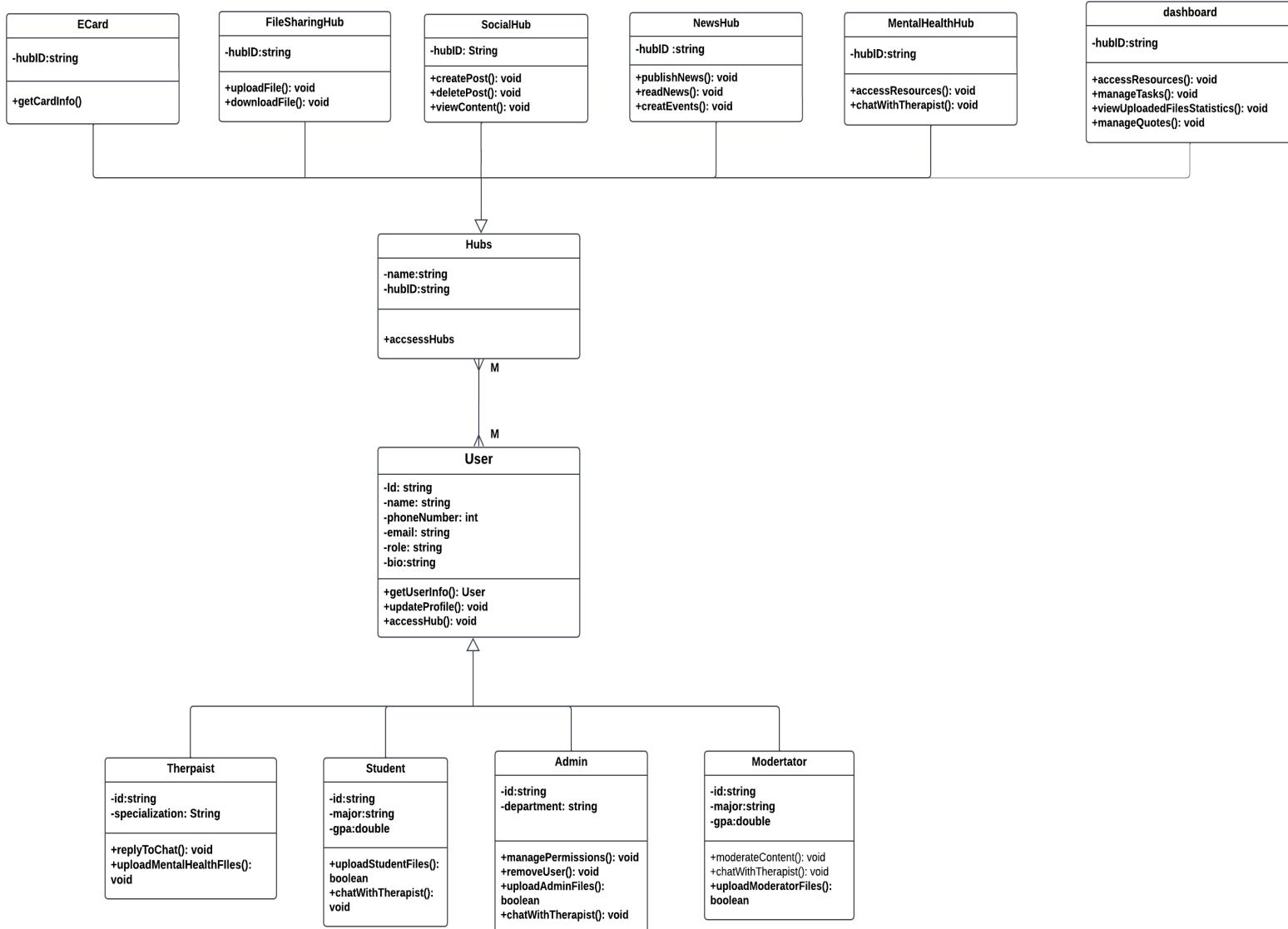


Figure 4-42:class diagram

## Graphical user interface design

**Admin pages:** the following figures represent the admin interfaces prototypes 2

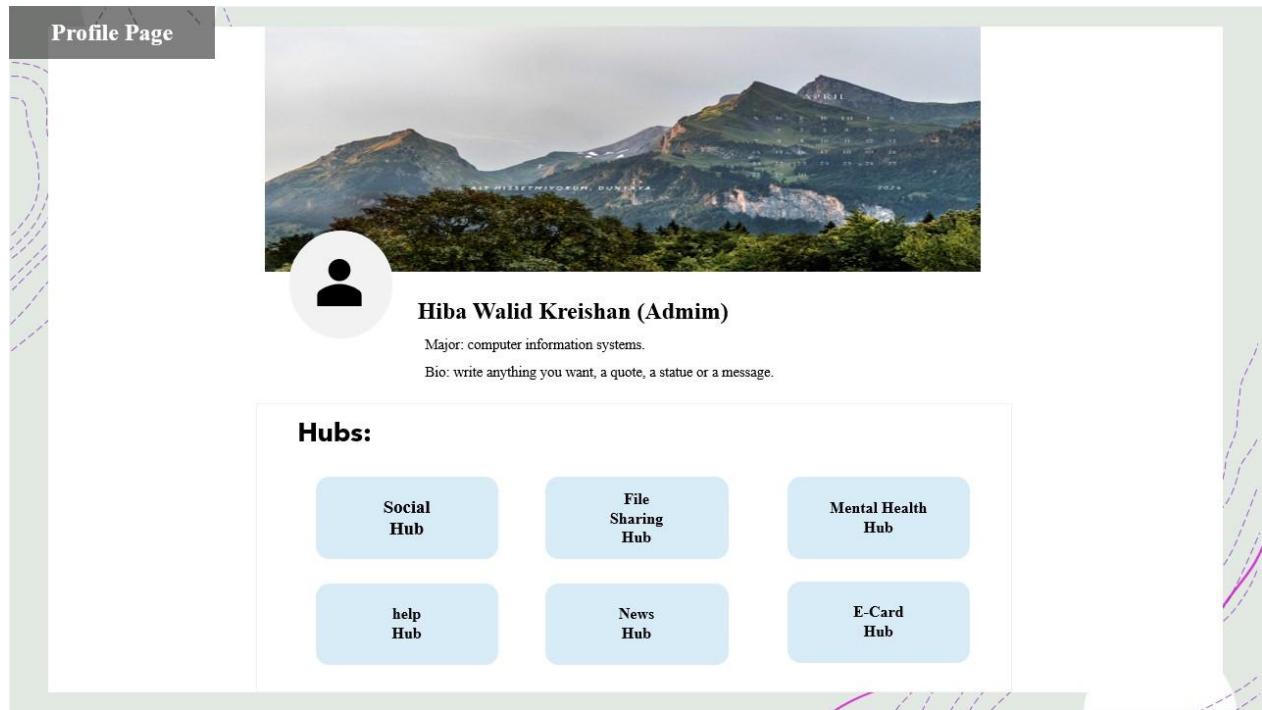


Figure 4-43: profile page design prototype

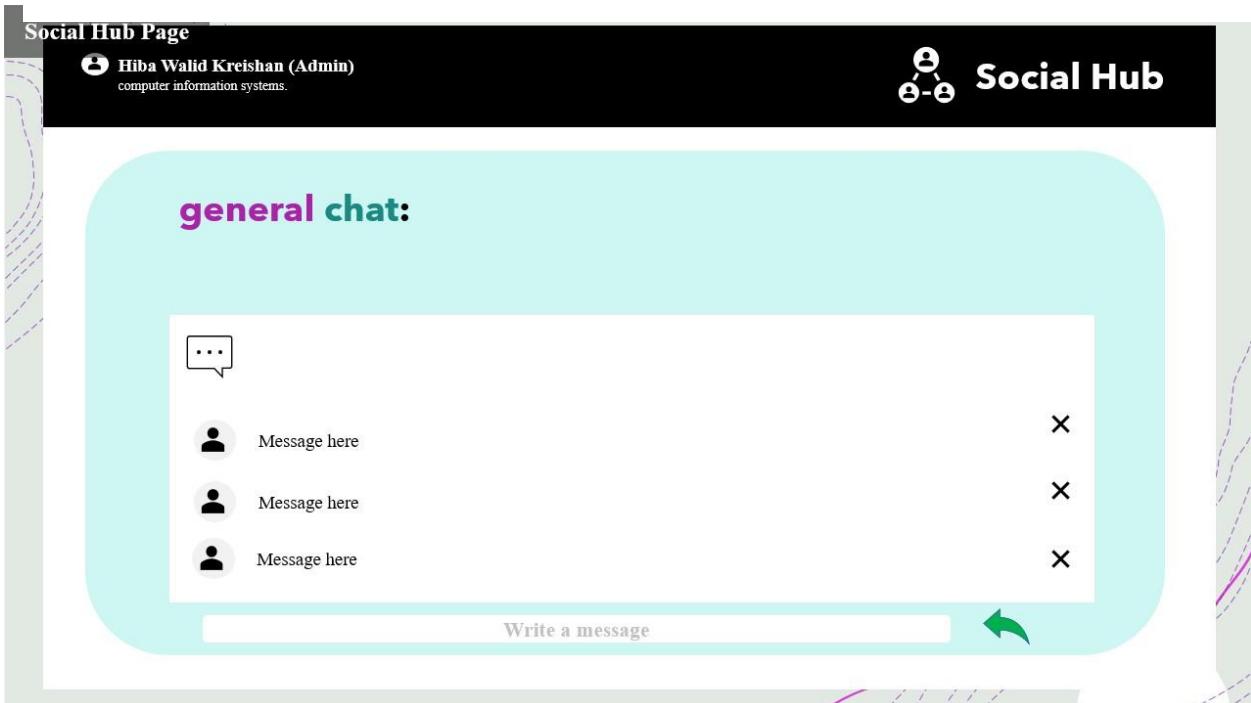


Figure 4-44: public chat design prototype

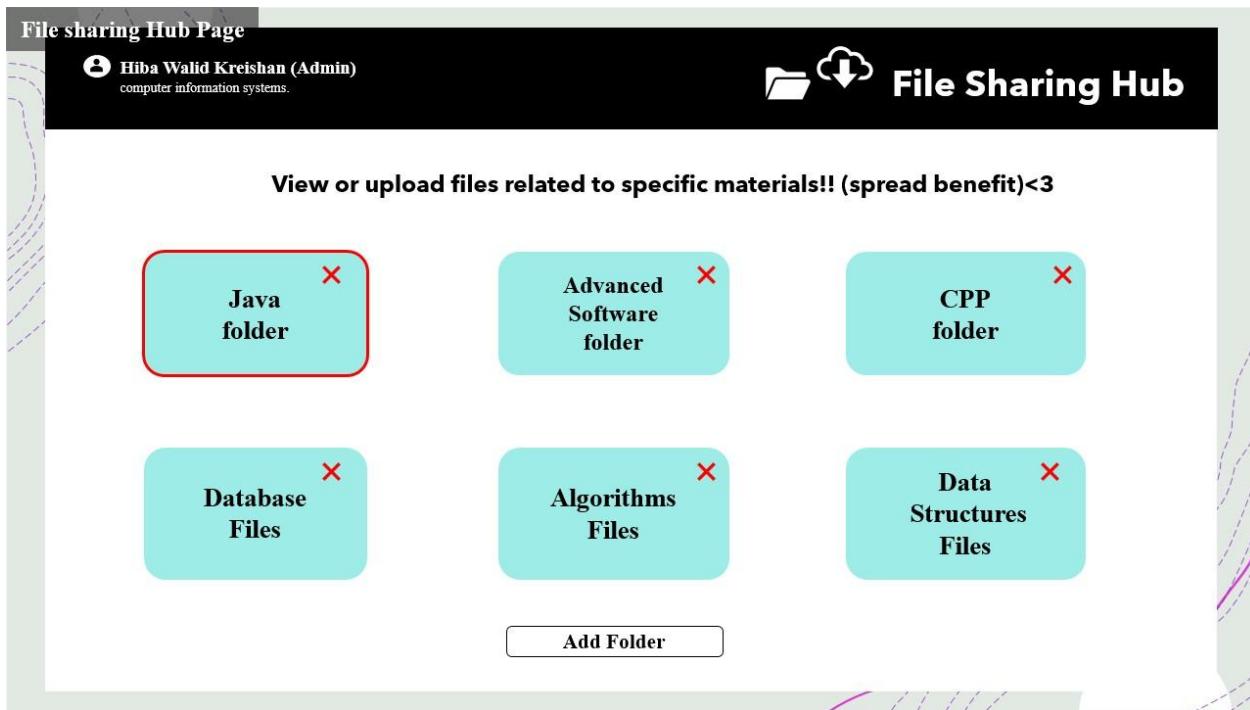


Figure 4-45: file sharing hub, folders design prototype

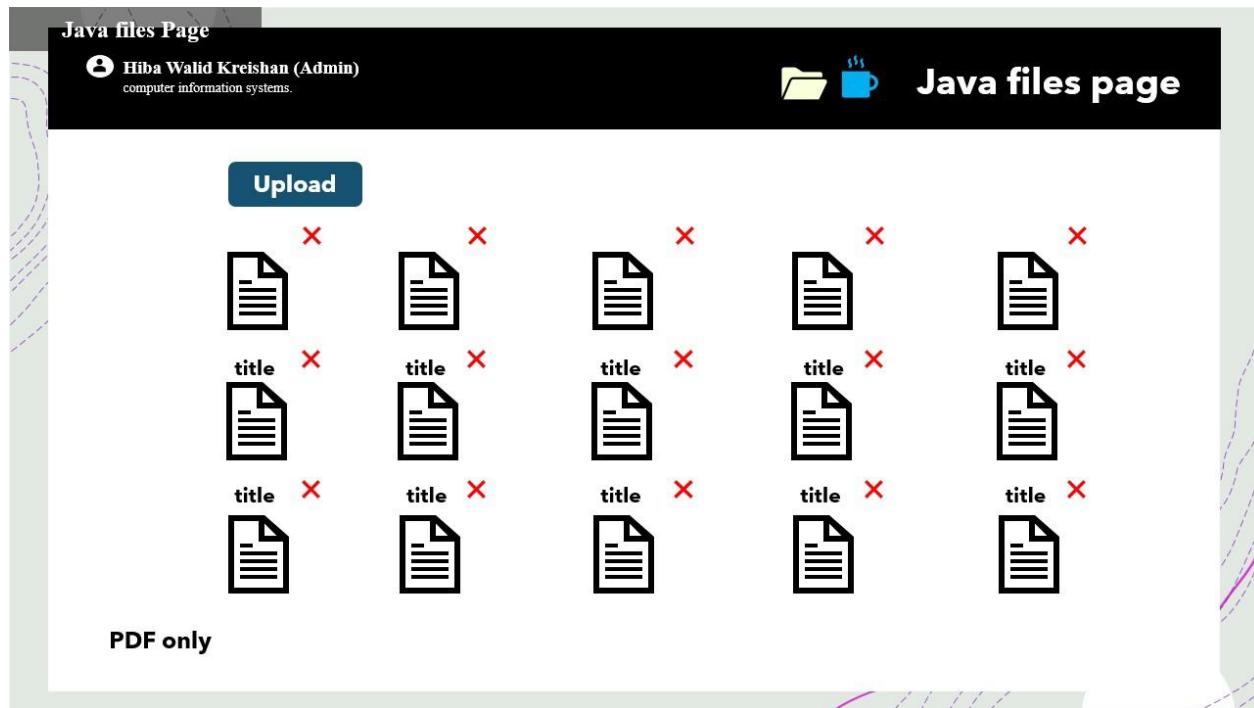


Figure 4-46: file sharing hub, design prototype

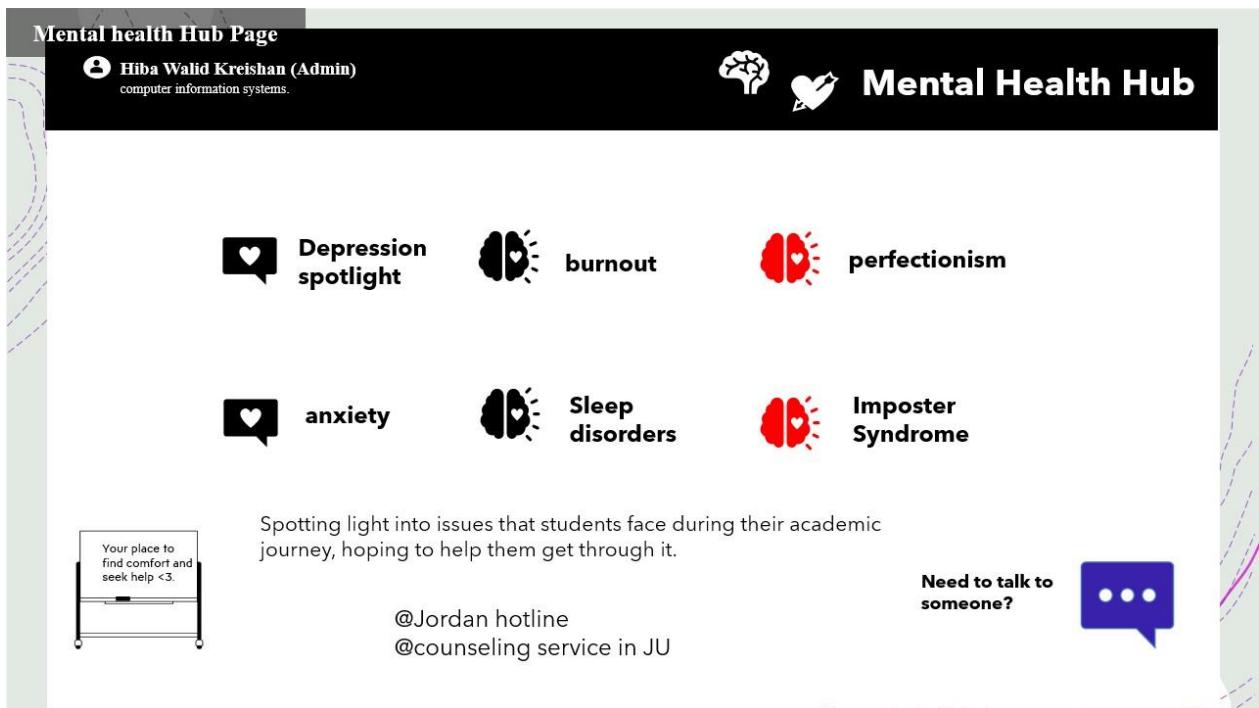


Figure 4-47: mental health hub, design prototype

6

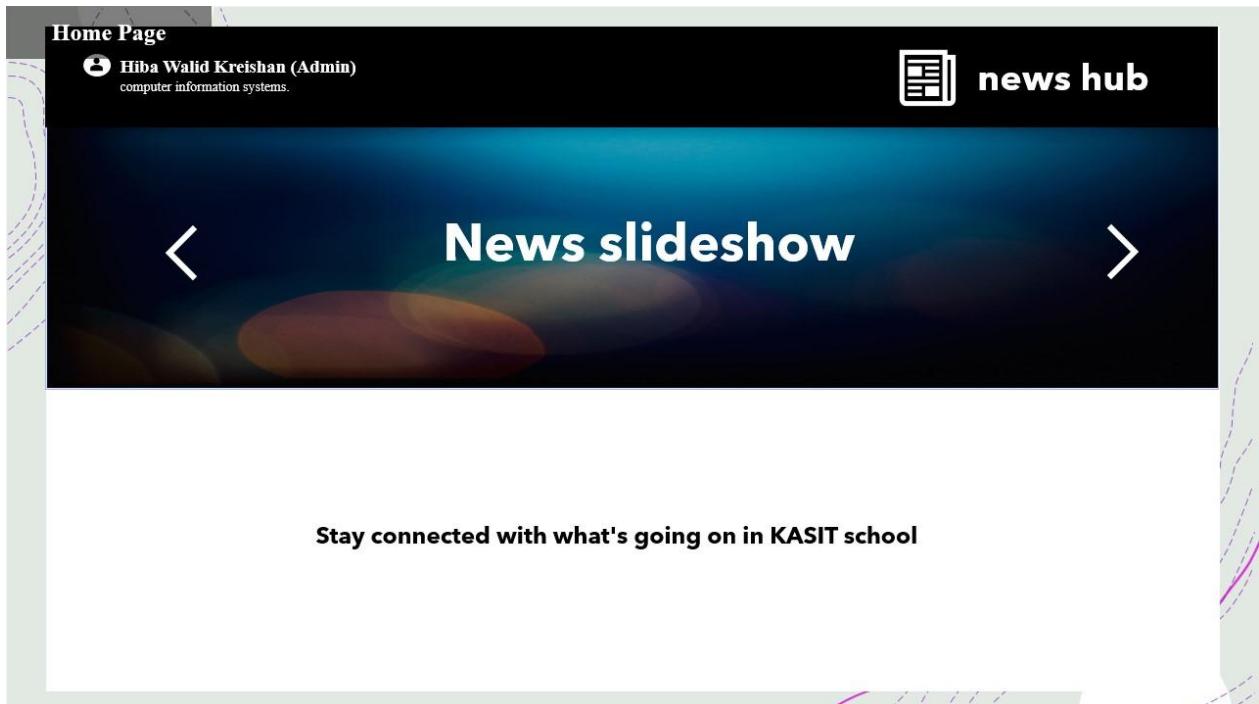


Figure 4-48:news hub, design prototype

7

The screenshot shows a web-based application interface titled "Home Page 2". At the top left is the user information "Hiba Walid Kreishan (Admin)" and "computer information systems.". On the right, there is a news icon and the text "News page 2". Below this, a section titled "Upcoming events" contains a calendar grid for the month. The grid shows days from 1 to 30. Specific events are marked: "event1" (red X) on day 1, "event2" (red X) on day 14, and "event3" (purple X) on day 23. A button labeled "Add Events" is located in the top right corner of the calendar area.

1 event1 X	2	3	4	5	6
7	8	9	10	11	12
13	14 event2 X	15	16	17	18
19	20	21	22	23 event3 X	24
25	26	27	28	29	30

Figure 4-49: events hub, design prototype

8

The screenshot shows a web-based application interface titled "E-Card Hub Page". At the top left is the user information "Zaina basem abdel hafez (admin)" and "computer information systems.". On the right, there is a clock and calendar icon and the text "E-Card Hub". The main content area features a green rounded rectangle containing the text "LINC" in red, followed by "JUS0215756", "Zaina Basem Abdel Hafez", and "King Abdallah school". To the right of this text is a black profile picture placeholder and two overlapping circular icons, one red and one yellow.

Figure 4-50: e-card design prototype

9

**Events Hub Page**

Rahaf Moatasem Saleem (student)  
computer information systems.

**Contents Hub**

**Submitted files:**

File	State	delete
FILE URL	Pending, Rejected, Accepted.	<b>DELETE</b>

Figure 4-51: file management design prototype

10

**News Hub Page**

Hiba Walid Kreishan (Admin)  
computer information systems.

**News management Hub**

Add News

New's id	content	Title	Timestamp	action
1	content	title	time	<b>Delete News</b>
2	content	title	time	<b>Delete News</b>
3	content	title	time	<b>Delete News</b>
4	content	title	time	<b>Delete News</b>
5	content	title	time	<b>Delete News</b>

Figure 4-52: news management prototype design

**Student Management Page**

Hiba Walid Kreishan (Admin)  
computer information systems.

**Students Management**

### List of Students

Student's ID	Student's First Name	Student's Last Name	Student's Email	Action
1	Hiba	Kreishan	H.M@gmail.com	<a href="#">Delete</a> <a href="#">Make moderator</a>
2	Rahaf	Moatasem	R.M@gmail.com	<a href="#">Delete</a> <a href="#">Make moderator</a>
3	Zaina	Abdelhafez	Z.A@gmail.com	<a href="#">Delete</a> <a href="#">Make moderator</a>
4	Ali	Barakat	A.B@gmail.com	<a href="#">Delete</a> <a href="#">Make moderator</a>
5	Mohammad	Shaker	M.S@gmail.com	<a href="#">Delete</a> <a href="#">Make moderator</a>

Figure 4-53: students management prototype design

**Moderators Management Page**

Hiba Walid Kreishan (Admin)  
computer information systems.

**Moderators Management**

### List of moderators

Moderator's ID	Moderator's First Name	Moderator's Last Name	Moderator's Email	Action
1	Hiba	Kreishan	H.M@gmail.com	<a href="#">Delete</a> <a href="#">Make student</a>
2	Rahaf	Moatasem	R.M@gmail.com	<a href="#">Delete</a> <a href="#">Make student</a>
3	Zaina	Abdelhafez	Z.A@gmail.com	<a href="#">Delete</a> <a href="#">Make student</a>
4	Ali	Barakat	A.B@gmail.com	<a href="#">Delete</a> <a href="#">Make student</a>
5	Mohammad	Shaker	M.S@gmail.com	<a href="#">Delete</a> <a href="#">Make student</a>

Figure 4-54: moderators management prototype design

**Student pages:** the following interfaces represent the student interface design

12

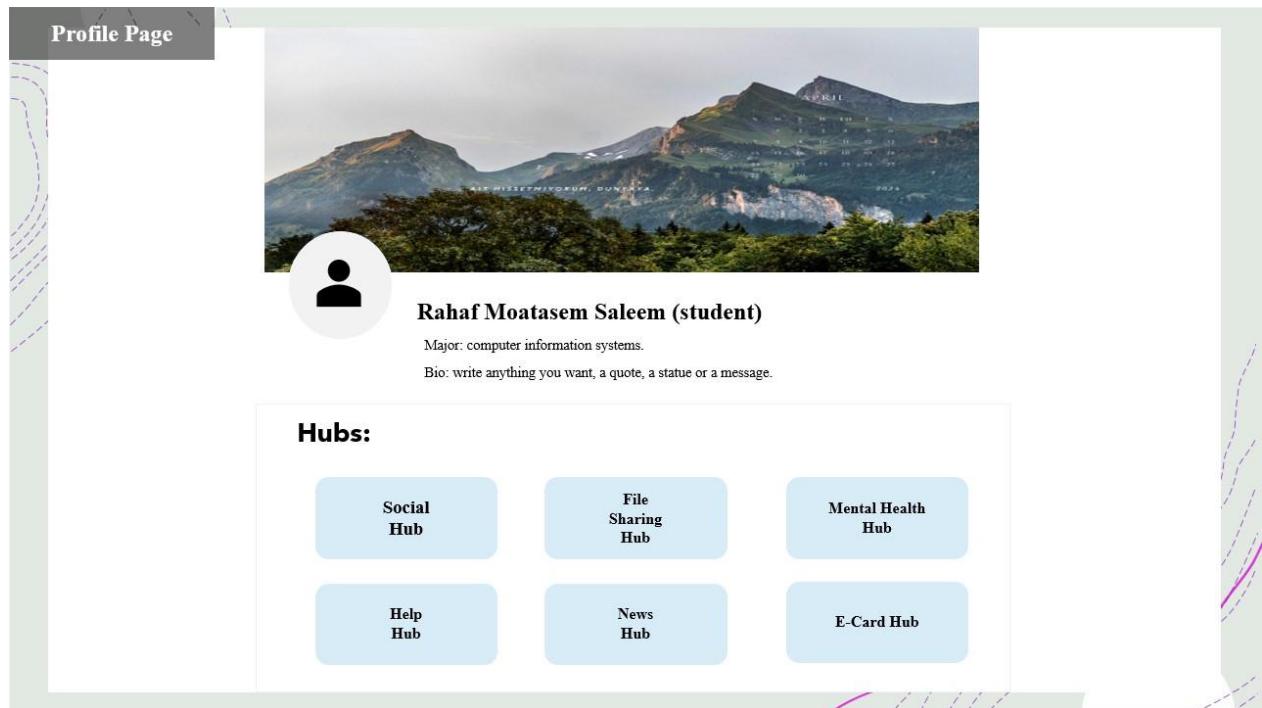


Figure 4-55: student profile page design prototype

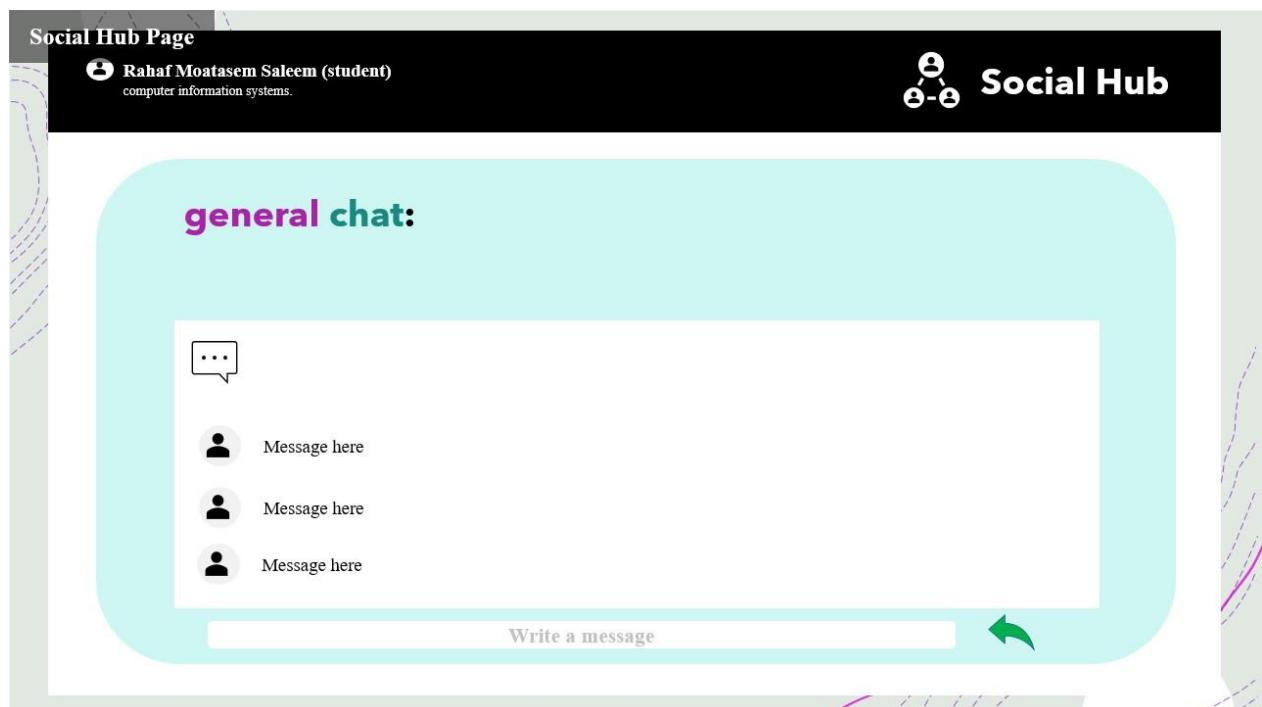


Figure 4-56: student social hub prototype

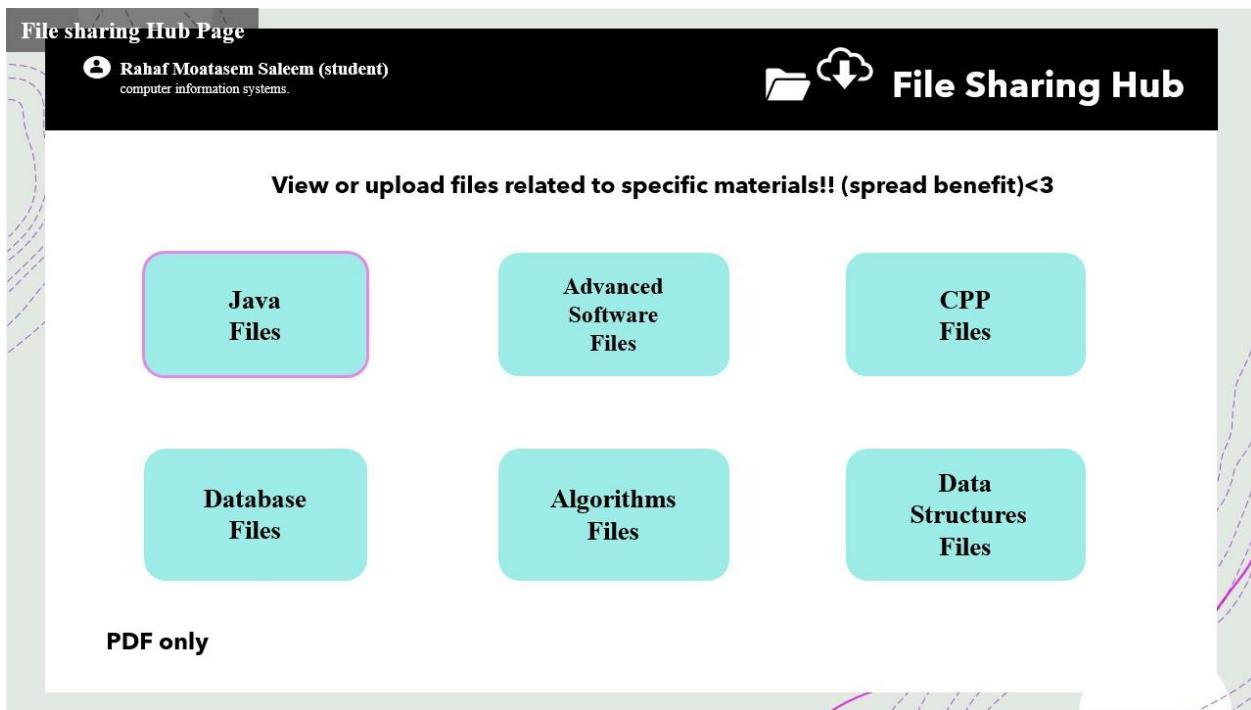


Figure 4-57: student file sharing hub design prototype

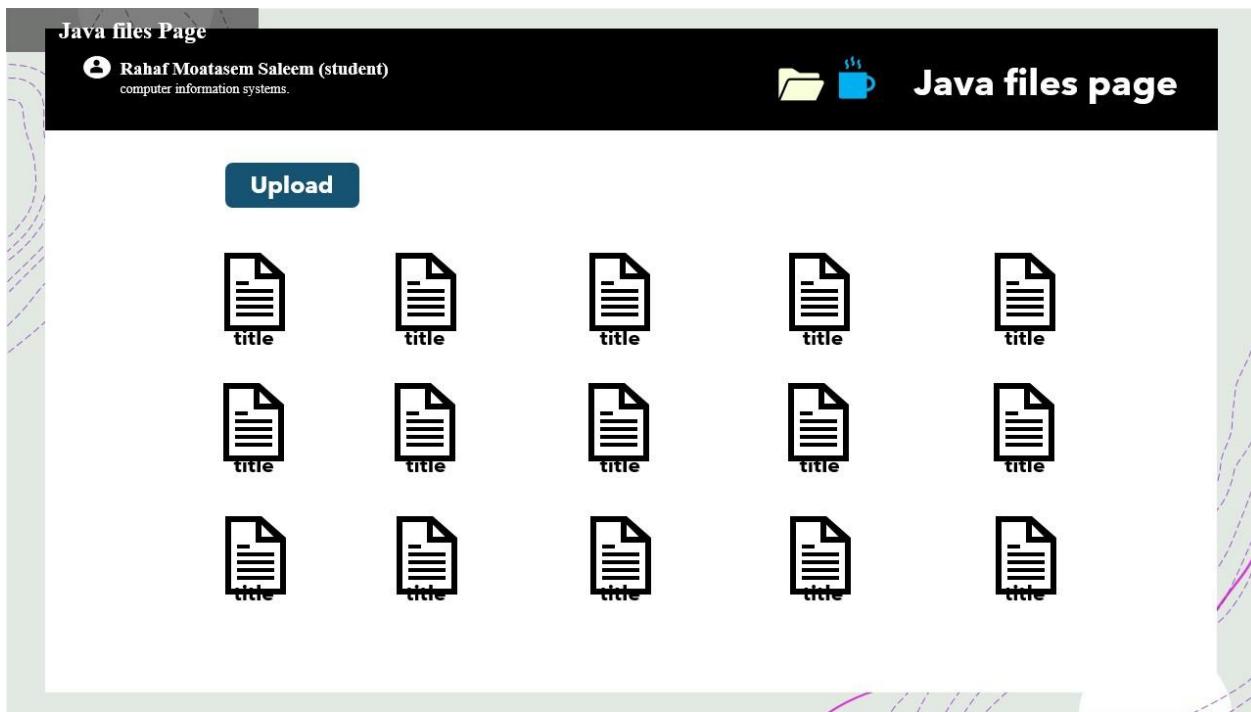


Figure 4-58: student file sharing hub part 2 design prototype

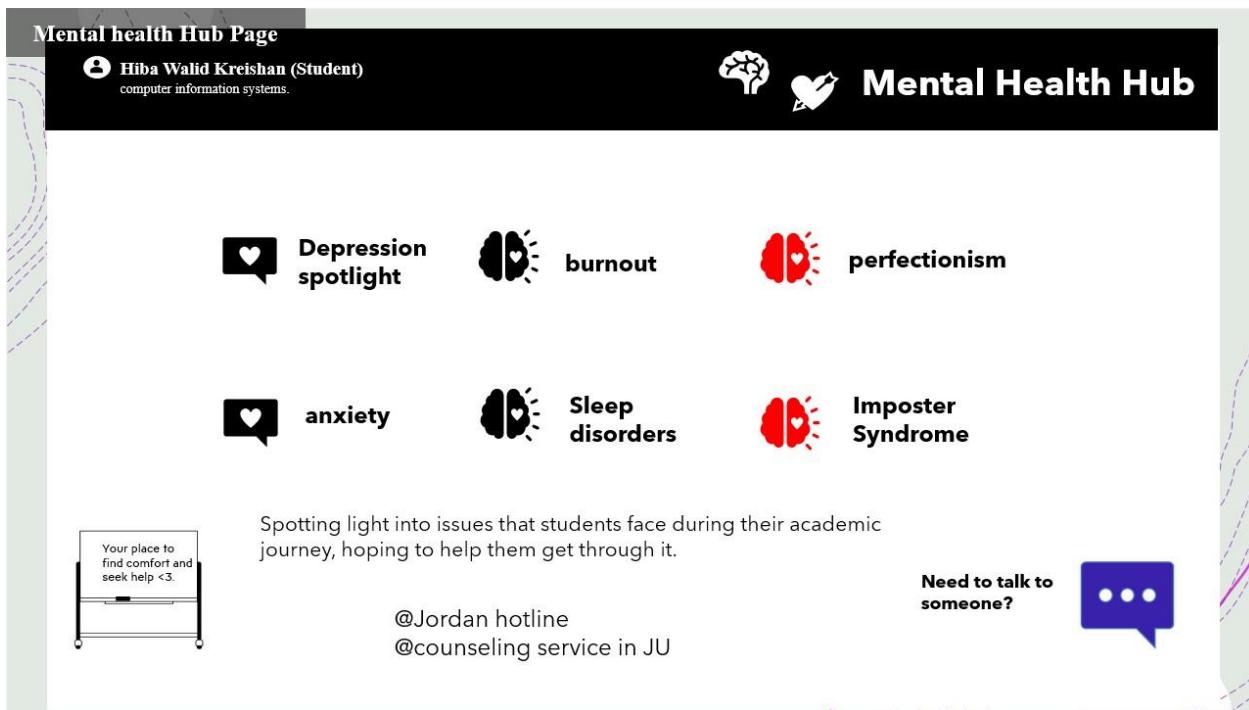


Figure 4-59: student mental health hub design prototype

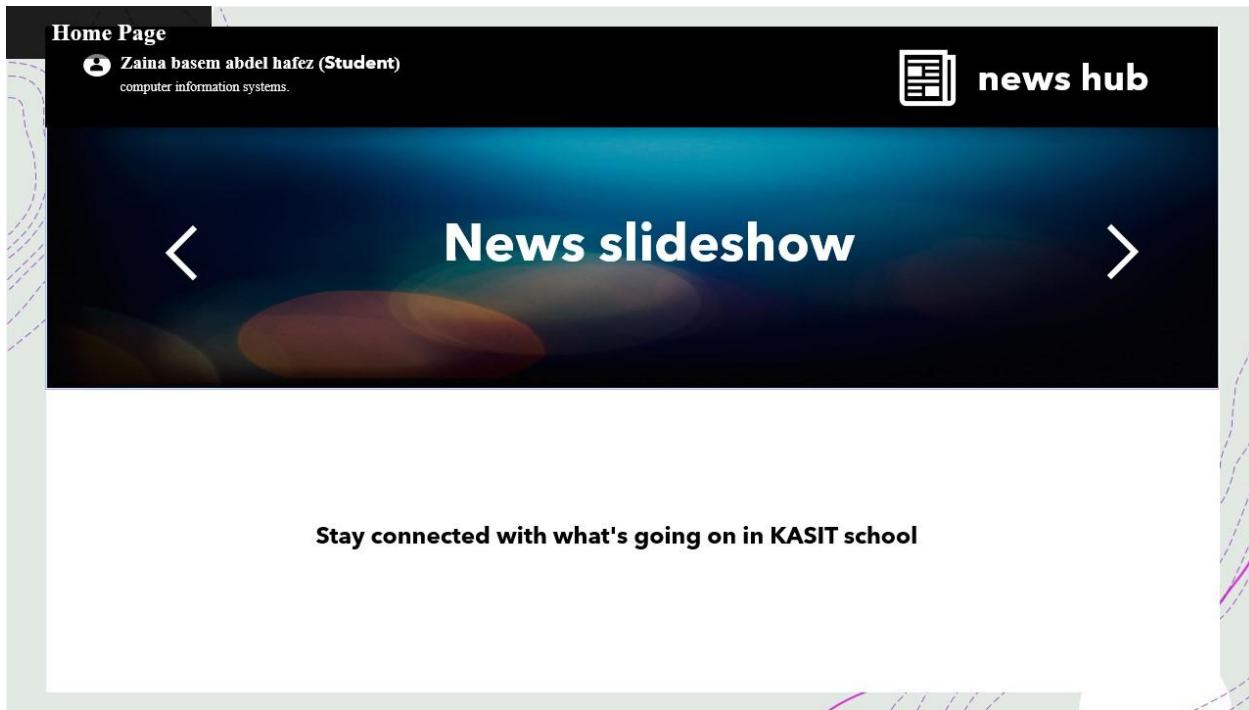


Figure 4-60: student news design prototype

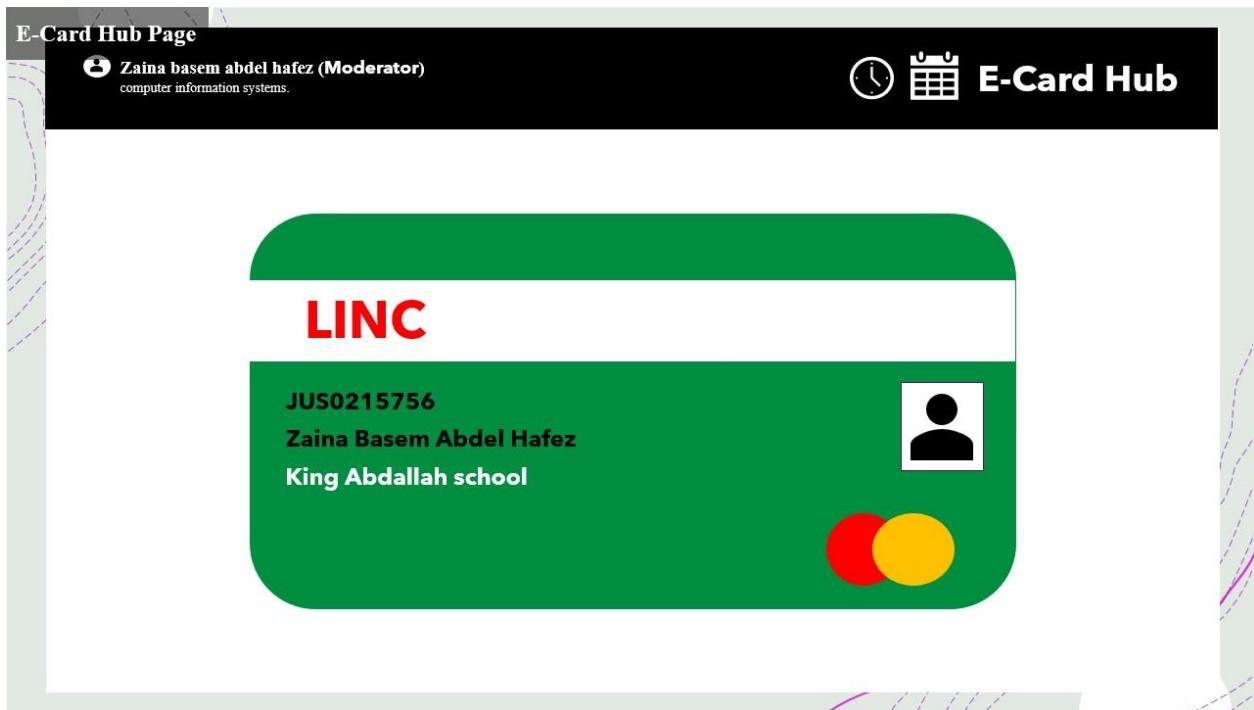
7



The screenshot shows a 'Events Hub Page' interface. At the top, there's a user profile for 'Rahaf Moatasem Saleem (student)' with the subtitle 'computer information systems.' To the right is a logo consisting of a clock and a calendar icon next to the text 'Events Hub'. Below the header, a section titled 'Upcoming events' features a circular icon with a clock symbol. A 5x6 grid calendar is displayed, showing dates from 1 to 30. Specific dates are highlighted with event labels: 'event1' in green on the 1st, 'event2' in red on the 14th, and 'event3' in purple on the 23rd.

Figure 4-61: student events design prototype

8



The screenshot shows an 'E-Card Hub Page' interface. At the top, there's a user profile for 'Zaina basem abdel hafez (Moderator)' with the subtitle 'computer information systems.' To the right is a logo consisting of a clock and a calendar icon next to the text 'E-Card Hub'. The main feature is a large, rounded rectangular card. The card has a green background and contains the text 'LINC' in red at the top. Below it, the card displays a student's information: 'JUS0215756', 'Zaina Basem Abdel Hafez', and 'King Abdallah school'. To the right of the text is a small black silhouette icon inside a white square. At the bottom right of the card is a graphic of a red and yellow circle, resembling a Mastercard logo. The card is set against a light gray background with decorative dashed lines on the sides.

Figure 4-62: student e-card design prototype

**Therapist pages:** The following interfaces represent the therapist interface design

1

**Profile Page**



A circular placeholder icon for a profile picture.

**Rahaf moatasem saleem (therapist)**

Major: computer information systems.  
Bio: write anything you want, a quote, a statue or a message.

**Hubs:**

- Therapists Chats
- E-Card Hub
- help Hub

*Figure 4-63: therapist profile page design prototype*

3



Stay connected with what's going on in KASIT school

Figure 4-64: news hub design prototype

4

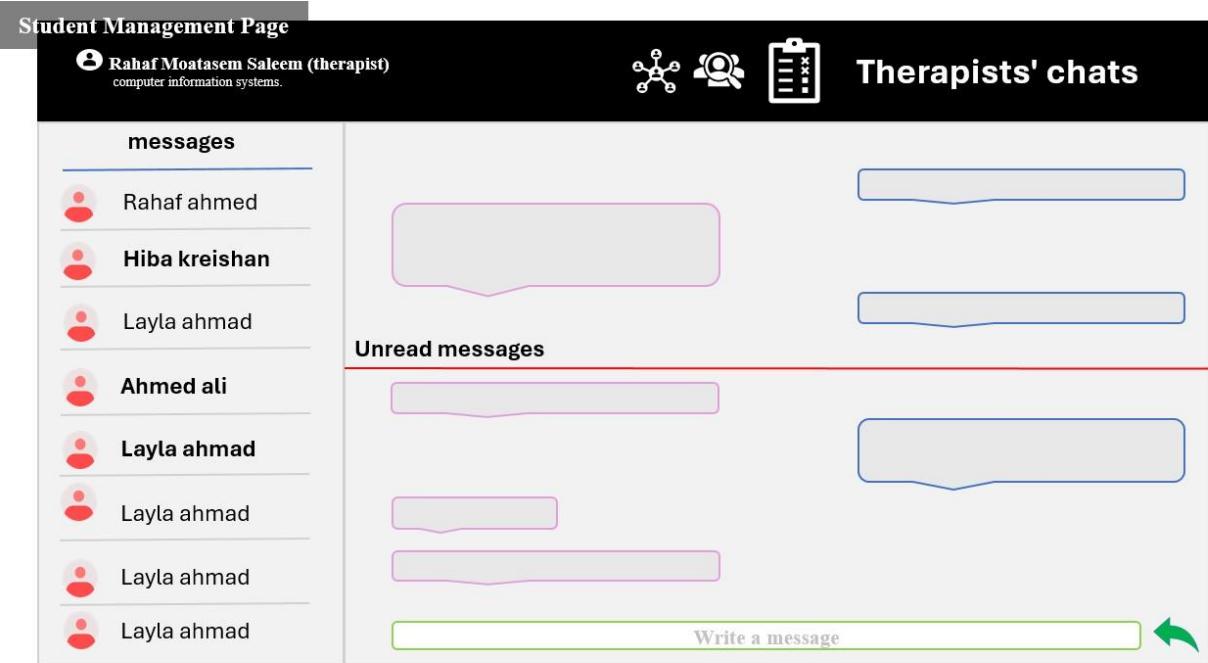


Figure 4-65: therapist chats design prototype

**Moderator pages:** The following interfaces represent the moderator interface design

1

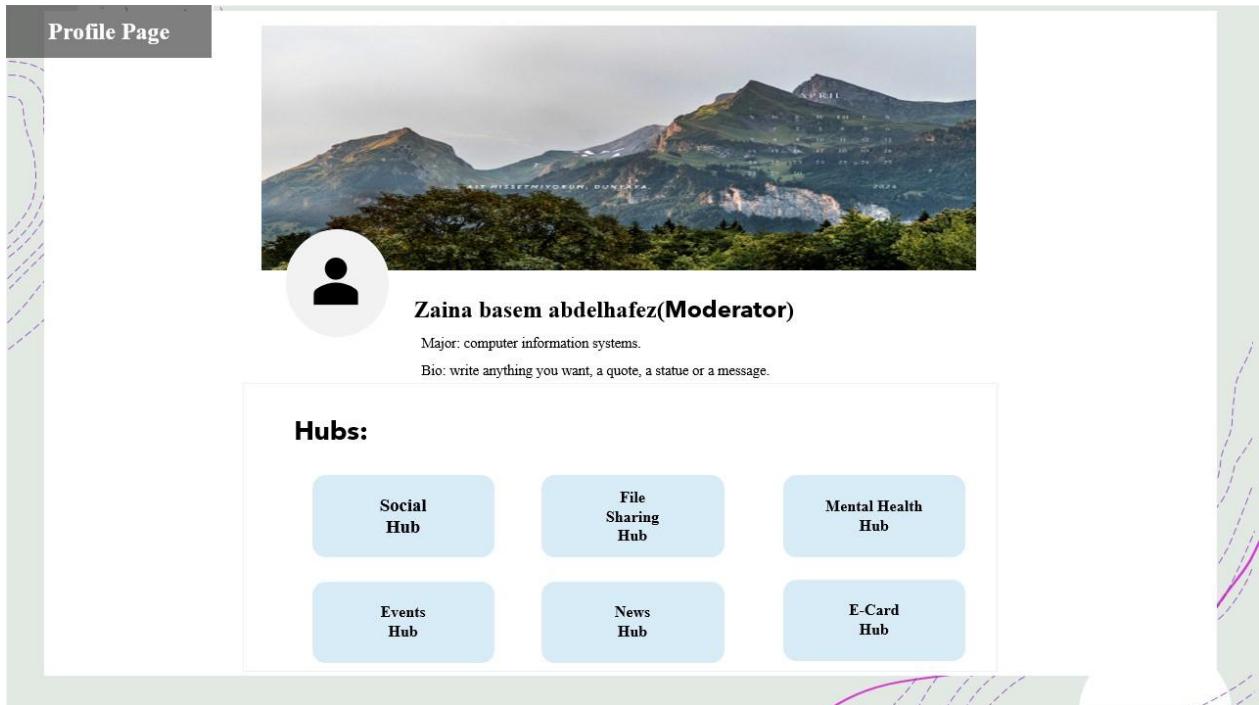


Figure 4-66: moderator profile page design prototype

4

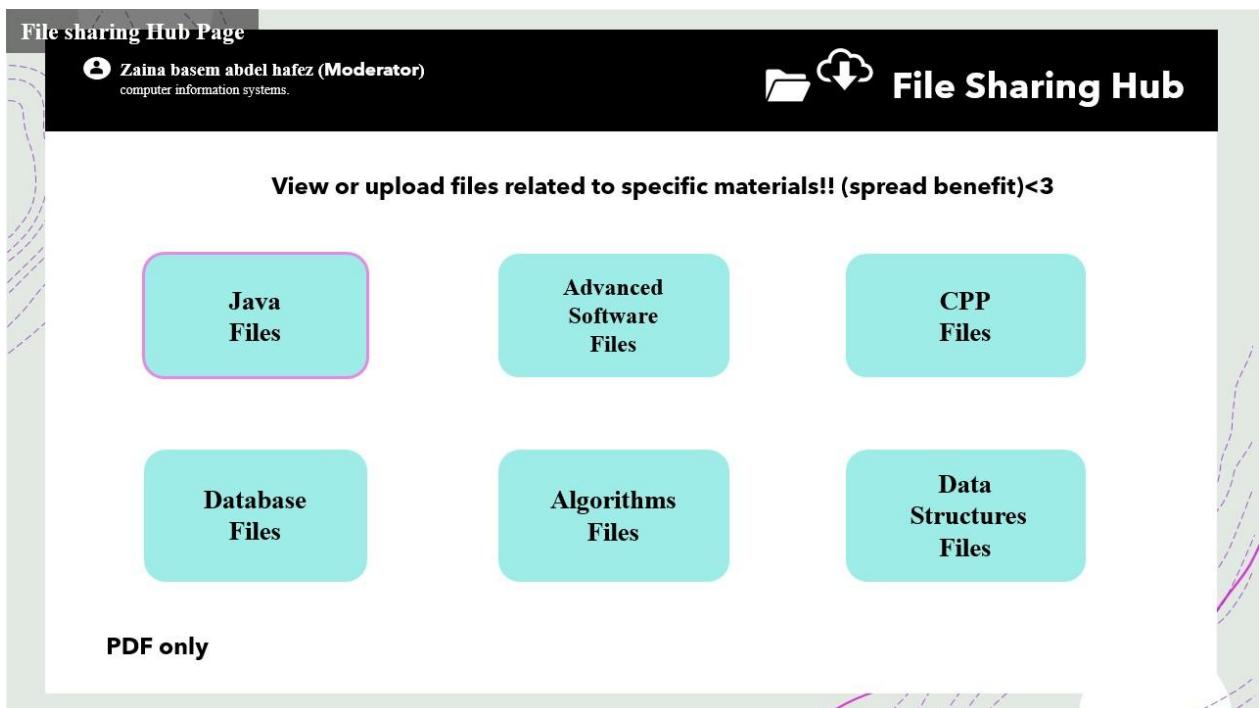


Figure 4-67: moderator social hub design prototype

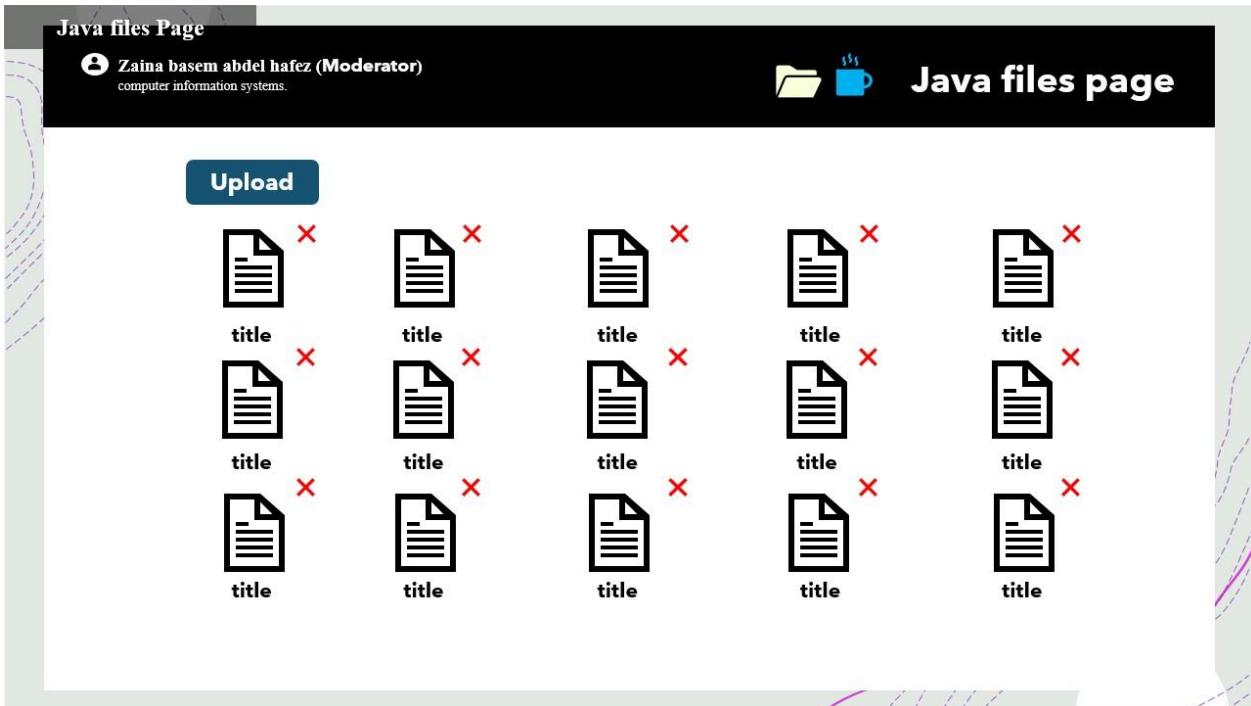


Figure 4-68: moderator file sharing hub design prototype

5

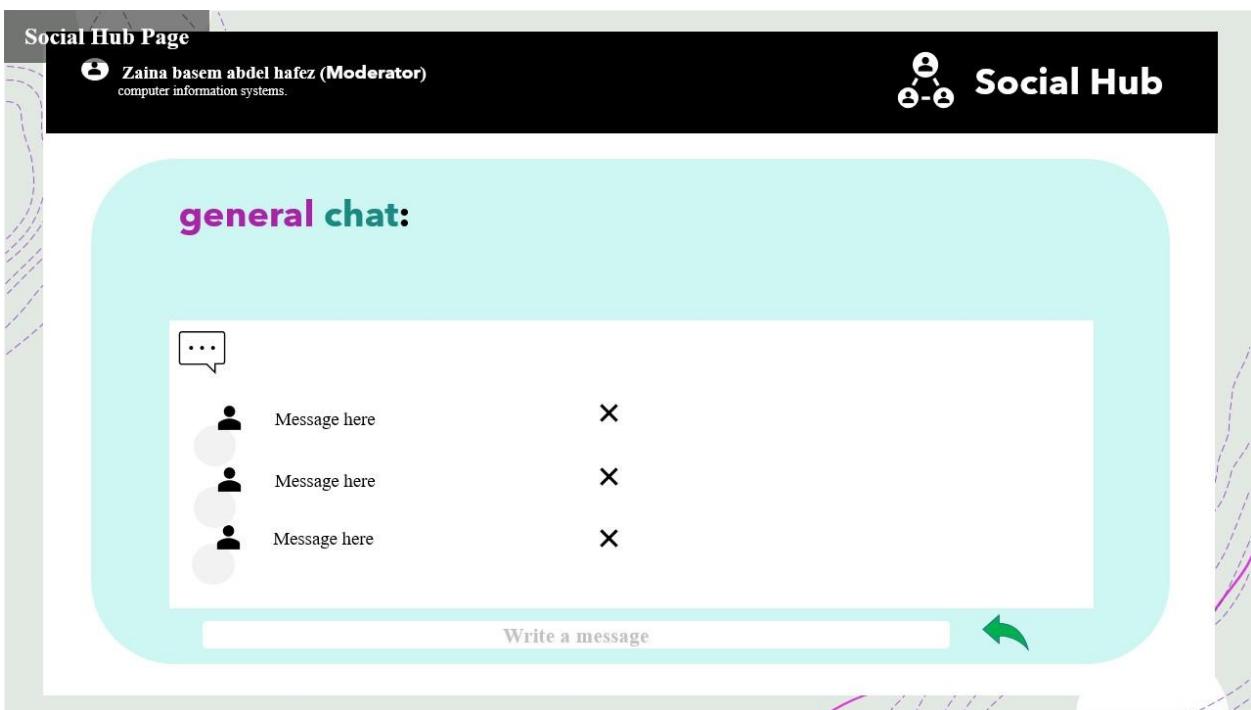


Figure 4-69: moderator public chat design prototype

6

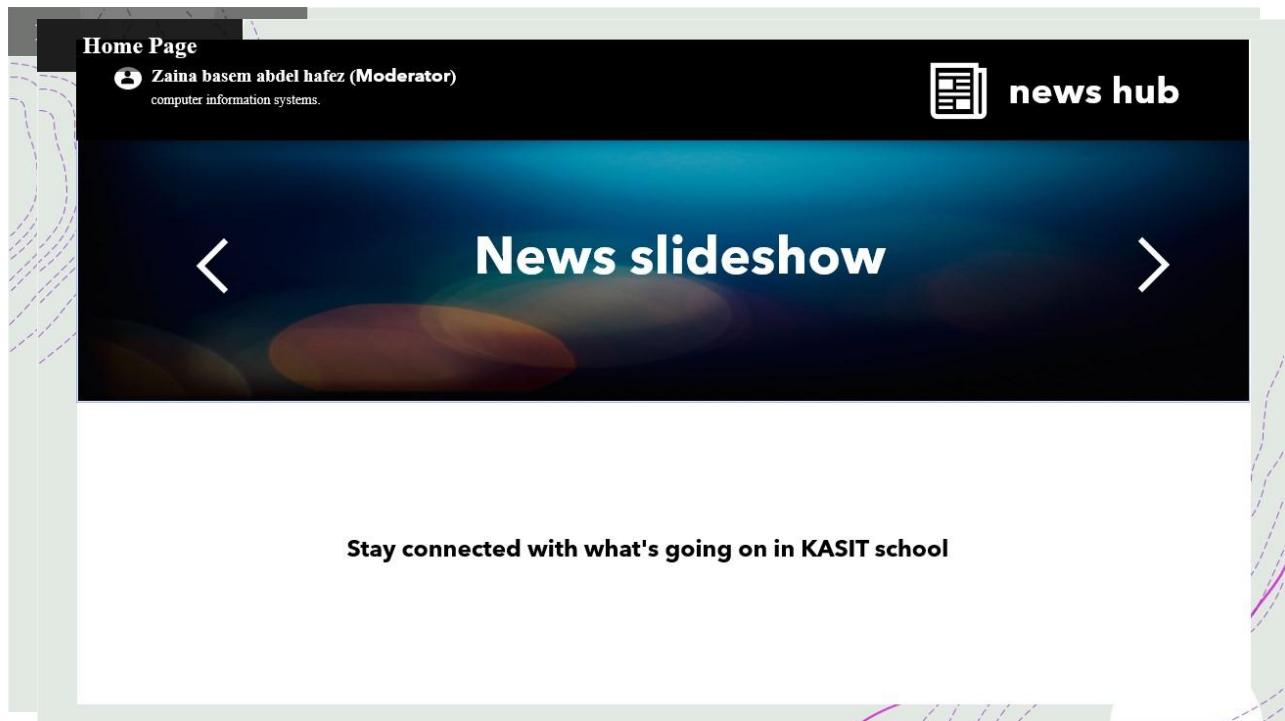


Figure 4-70: moderator news design prototype

7

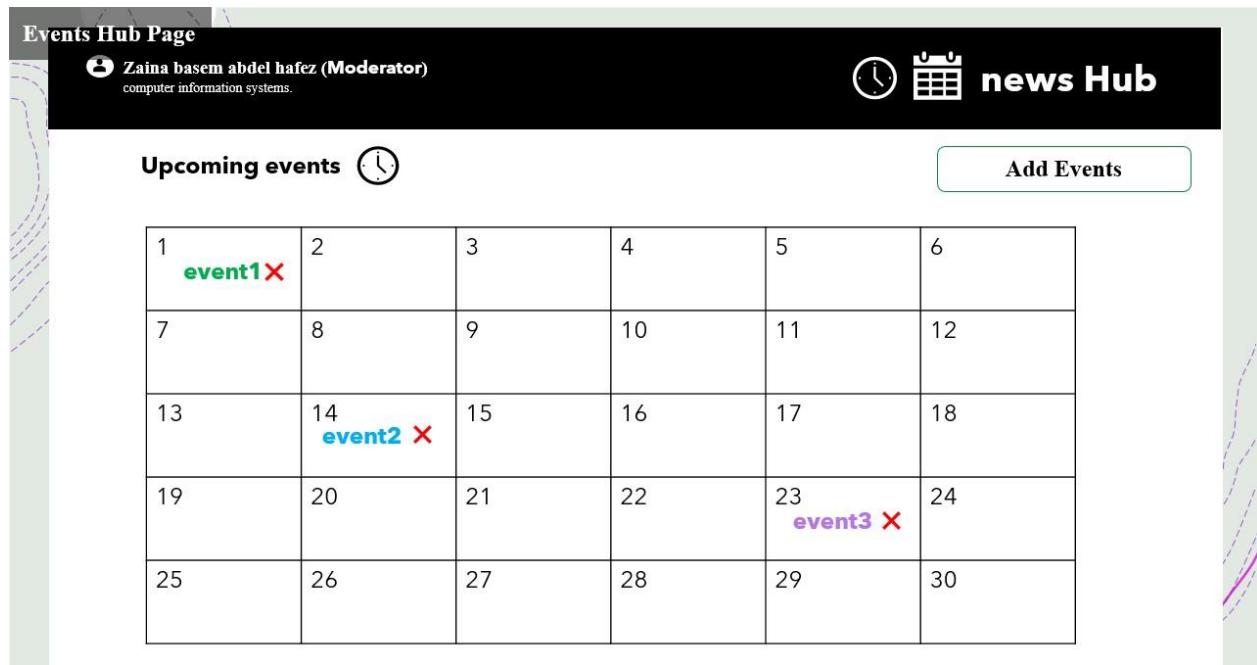


Figure 4-71: moderator event design prototype

**Java files Page**

Zaina basem abdel hafez (Moderator)  
computer information systems.

**pending files management**

Uploader ID	File	Action
IDS	FILE URL	Reject, Accept

Figure 4-72: moderator pending files management design prototype

**E-Card Hub Page**

Zaina basem abdel hafez (Moderator)  
computer information systems.

**E-Card Hub**

**LINC**

JUS0215756  
Zaina Basem Abdel Hafez  
King Abdallah school




Figure 4-73: moderator e-card design prototype

## Login pages



Figure 4-74: login page design prototype

## Dashboard pages

1

Layla Hassan (STUDENT)

Major: BIT  
Bio:sen benim ictegim segarinin domanisin... gecmis degil bugun gibi yasiyorum hala seni, sen benim sarkilarsin

Hubs:

**Today's quote**

"When we strive to become better than we are, everything around us becomes better too."

**To-Do list**

Figure 4-75: profile dashboard design prototype

2

**Accepted files**

File name	uploaded at	accepted at

**Pending files**

File name	uploaded at

Figure 4-76: last files design prototype

3

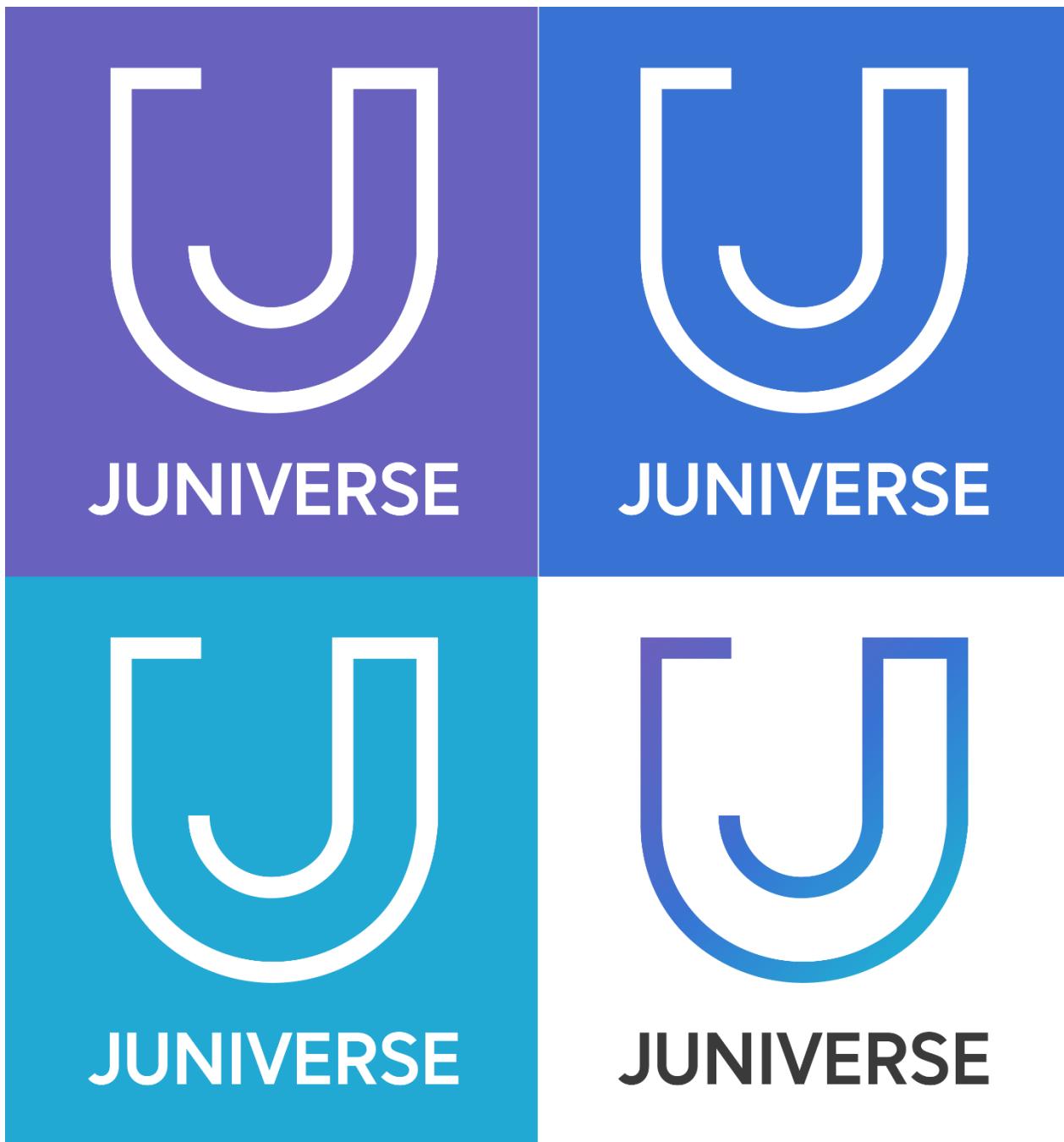


4

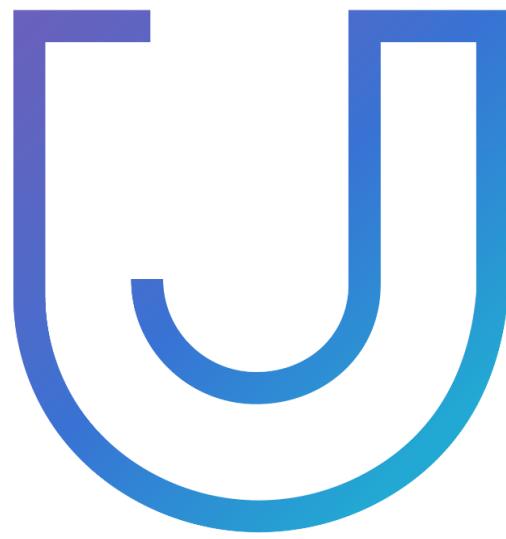


Figure 4-78: footer design prototype

**Logo design**



*Figure 4-79: logo design*



# JUNIVERSE



HEX: #6861bd



HEX: #3873d4



HEX: #22a9d3

*Figure 4-80: logo design part 2*

## **5. CHAPTER FIVE: SYSTEM IMPLEMENTATION**

### **Introduction**

In this chapter, the website implementation is discussed. In section 5.1 we will show the database implementation, meanwhile, the backend implementation is shown in section 5.2, and finally the frontend implementation 5.3.

### **Source code**

Here are the links to our GitHub repositories:

Front-End GitHub Link: <https://github.com/HibaKreishan2002/JUniVerse.git>

Back-End GitHub Link: <https://github.com/rahafduglass/juniverse-backend.git>

## Database implementation

We used JPA from spring data to manage our database. The following code represents the most important parts and entities in our JPA code

```
@Entity(name = "message") 24 usages ✘ rahafulduglass
@AllArgsConstructor
@NoArgsConstructor
@Data
@Table(name = "message")
public class MessageEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(nullable = false)
    private String content;

    @Column(nullable = false)
    private LocalDateTime timestamp;

    @Column(nullable = true)
    private Boolean isRead;

    @Enumerated(EnumType.STRING)
    @Column(nullable = false)
    private ChatType chatType;

    @Enumerated(EnumType.STRING)
    @Column(nullable = false)
    private MessageStatus status;

    @ManyToOne
    @JoinColumn(name = "sender_id", nullable = false)
    private SysUserEntity sender;

    @ManyToOne
    @JoinColumn(name = "receiver_id", nullable = true)
    private SysUserEntity receiver;

    @ManyToOne
    @JoinColumn(name = "private_chat_id", nullable = true)
    private PrivateChatEntity privateChat;

    @ManyToOne
    @JoinColumn(name = "deleted_by")
    private SysUserEntity deletedBy;

    private Boolean isFile;

    @OneToOne
    private FileEntity file;
}
```

Figure 5-1: message JPA entity

```
@Entity(name="private_chat") ✘ rahafulduglass
@Data
@AllArgsConstructor
@NoArgsConstructor
@Table(name="private_chat")
public class PrivateChatEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @OneToOne(fetch = FetchType.EAGER) //each user has one private chat
    @JoinColumn(name = "user_id", nullable = false)
    private SysUserEntity user;

    @ManyToOne(fetch = FetchType.EAGER) //one therapist has many private chat
    @JoinColumn(name = "therapist_id", nullable = false)
    private SysUserEntity therapist; //it'll be set from domain as static dat
}
```

Figure 5-2: private chat JPA entity

Other entities are implemented just like that, our final JPA entities are 11.

## backend implementation

Our backend code was layered and organized in a very clean way, we separated domain, persistence and presentation layers, it was based on spring boot APIs and utils which helped us so much and saved us a lot of time. Thank u spring :D <3

We'll give a snippet of some of our services code:

```
public MessageModel sendMessageToTherapist(String content) throws Exception { 1 usage ▲ rahafduglass

    if (content.isEmpty())
        throw new Exception("can't send empty message");

    SysUserEntity currentUser = identityProvider.currentIdentity();
    SysUserEntity therapist = sysUserRepository.findByUsername("omar_khaled").get();
    PrivateChatModel privateChat = privateChatRepository.findById(currentUser.getId());

    if (privateChat == null) {
        privateChat = privateChatService.createChatBetween(currentUser.getId(), therapist.getId());
    }

    MessageModel messageModel = new MessageModel();
    messageModel.setContent(content);
    messageModel.setSenderUsername(currentUser.getUsername());
    messageModel.setSenderId(currentUser.getId());
    messageModel.setReceiverUsername(therapist.getUsername());
    messageModel.setReceiverId(therapist.getId());
    messageModel.setChatType(ChatType.PRIVATE);
    messageModel.setIsRead(false);
    messageModel.setTimestamp(LocalDateTime.now());
    messageModel.setPrivateChatId(privateChat.getId());
    messageModel.setStatus(MessageStatus.SENT);

    return sendMessage(messageModel);
}

public List<MessageModel> getAllPrivateMessages(Long chatId) { 1 usage ▲ rahafduglass
    List<MessageModel> privateMessages = messageRepository.findAllByPrivateChatId(privateChatRepository.findById(chatId).getId());
    privateChatService.markChatAsRead(chatId);
    return privateMessages;
}

public List<MessageModel> getAllPrivateMessages() { 1 usage ▲ rahafduglass
    Long chatId = privateChatRepository.findById(identityProvider.currentIdentity().getId());
    List<MessageModel> privateMessages = messageRepository.findAllByPrivateChatId(chatId);

    privateChatService.markChatAsRead(chatId);

    return privateMessages;
}

public MessageModel sendMessageFromTherapist(String content, String receiverUsername, Long privateChatId) throws Exception { 1 usage ▲ rahafduglass

    if (content.isEmpty())
        throw new Exception("can't send empty message");

    SysUserEntity therapist = identityProvider.currentIdentity();
    SysUserEntity receiver;
```

Figure 5-3: message service code snippet

```

private final FileRepository fileRepository;
private final IdentityProvider identityProvider;
private final FolderRepository folderRepository;
private final NotificationService notificationService;

public boolean addFile(FileModel fileModel, String fileAsBase64) throws IOException {
    validateFile(fileModel);

    SysUserEntity currentUser = identityProvider.currentIdentity();

    fileModel.setStatus(currentUser.getRole() == UserRole.STUDENT ? FileStatus.PENDING : FileStatus.ACCEPTED);
    fileModel.setOwnerId(currentUser.getId());
    fileModel.setOwnerUsername(currentUser.getUsername());
    fileModel.setUploadDate(LocalDateTime.now());
    fileModel.setPath("src\\main\\resources\\juniverse_files\\folders\\" + fileModel.getFolderId() + "\\");
    FileModel savedFile = fileRepository.addFile(fileModel);

    byte[] decodedFile = Base64.getDecoder().decode(fileAsBase64);
    String filePath = "src\\main\\resources\\juniverse_files\\folders\\" + fileModel.getFolderId() + "\\" + savedFile.getId() + "." + savedFile.getExtension();

    FileOutputStream fileOutputStream = new FileOutputStream(filePath);
    fileOutputStream.write(decodedFile);
    fileOutputStream.close();

    return true;
}

private void validateFile(FileModel fileModel) {}

public List<FileModel> getAcceptedFiles(Long folderId) {}

private void validateFolder(Long folderId) {}

```

Figure 5-5: file service code snippet

```

@Service 2 usages ✘ rahafduglass
@RequiredArgsConstructor
public class TherapistNoteService {

    private final NoteRepository noteRepository;
    private final TherapistNoteMapper therapistNoteMapper;
    private final IdentityProvider identityProvider;

    public boolean addNote(TherapistNoteModel therapistNoteModel) { 1 usage ✘ rahafduglass
        therapistNoteModel.setTherapistId(identityProvider.currentIdentity().getId());
        return noteRepository.addNote(therapistNoteModel);
    }

    public List<TherapistNoteModel> getNotes(Long chatId) { return noteRepository.getNotes(chatId); }

    public boolean updateNote(Long noteId, String title, String description) { 1 usage ✘ rahafduglass
        TherapistNoteModel therapistNoteModel = TherapistNoteModel.builder()
            .id(noteId)
            .title(title)
            .description(description)
            .build();
        return noteRepository.updateNote(therapistNoteModel);
    }

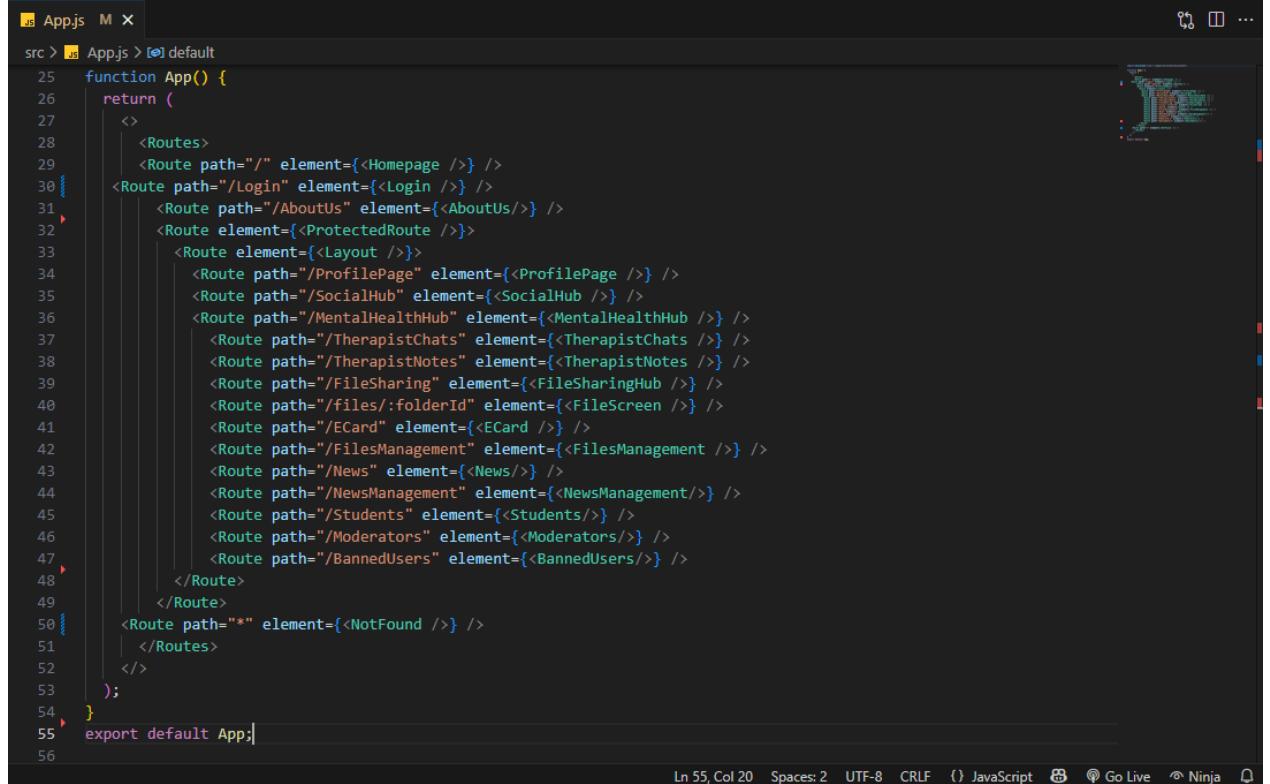
    public boolean deleteNote(Long noteId) { return noteRepository.deleteMessage(noteId); }
}

```

Figure 5-4: therapist note code snippet

## Frontend implementation

We organized our frontend using React.js and MUI by creating separate reusable components for each part of the UI, like buttons, forms, and cards. This made our code cleaner, easier to manage, and more consistent. We also grouped related files into folders to keep everything well structured.



```
App.js  default
src > App.js > default
25  function App() {
26    return (
27      <>
28        <Routes>
29          <Route path="/" element={<Homepage />} />
30          <Route path="/Login" element={<Login />} />
31          <Route path="/AboutUs" element={<AboutUs/>} />
32          <Route element={<ProtectedRoute />}>
33            <Route element={<Layout />}>
34              <Route path="/ProfilePage" element={<ProfilePage />} />
35              <Route path="/SocialHub" element={<SocialHub />} />
36              <Route path="/MentalHealthHub" element={<MentalHealthHub />} />
37              <Route path="/TherapistChats" element={<TherapistChats />} />
38              <Route path="/TherapistNotes" element={<TherapistNotes />} />
39              <Route path="/FileSharing" element={<FileSharingHub />} />
40              <Route path="/files/:folderId" element={<FileScreen />} />
41              <Route path="/ECard" element={<ECard />} />
42              <Route path="/FilesManagement" element={<FilesManagement />} />
43              <Route path="/News" element={<News />} />
44              <Route path="/NewsManagement" element={<NewsManagement />} />
45              <Route path="/Students" element={<Students />} />
46              <Route path="/Moderators" element={<Moderators />} />
47              <Route path="/BannedUsers" element={<BannedUsers />} />
48            </Route>
49          </Route>
50        <Route path="*" element={<NotFound />} />
51      </Routes>
52    </>
53  );
54 }
55 export default App;
```

Ln 55, Col 20 Spaces: 2 UTF-8 CRLF {} JavaScript ⚡ Go Live ⚡ Ninja ⚡

Figure 5-6: App.js code snippet

A screenshot of a code editor showing the `formLogin.js` file. The code is written in JavaScript and uses React components. It includes state management with `useState` and `useEffect`, form handling with `useForm`, and axios requests for logging in. The code editor interface shows tabs, status bar, and a sidebar.

```
src > components > FormLogin.js > FormLogin
74 function FormLogin () {
75     const [showPassword, setShowPassord] = useState(false);
76     const [remember, setRemember] = useState(true);
77
78     const handleTogglePassword = () => setShowPassord(!showPassword);
79     const handleToggleRemember = () => setRemember(!remember);
80
81     const navigate = useNavigate();
82
83     // hook form
84     const {
85         register,
86         handleSubmit,
87         formState: { errors },
88     } = useForm({
89         defaultValues: {
90             email: "",
91             password: "",
92             rememberUser: true,
93         },
94     });
95
96     useEffect(()=>{
97         sessionStorage.clear();
98     },[])
99     // prevent Default
100    const preventDefault = (e) => e.preventDefault();
101
102    // form submit
103
104    const onSubmit = (data) => {//This method we use it to handle Login Button
105        JuUniVerseAxios.post("/auth/signIn",{username:data.email,password:data.password}).then(res=>{
106            ...
107        })
108    }
109
110    ...
111
112    ...
113
114    ...
115
116    ...
117
118    ...
119
120    ...
121
122    ...
123
124    ...
125
126    ...
127
128    ...
129
130    ...
131
132    ...
133
134    ...
135
136    ...
137
138    ...
139
140    ...
141
142    ...
143
144    ...
145
146    ...
147
148    ...
149
150    ...
151
152    ...
153
154    ...
155
156    ...
157
158    ...
159
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
229
230
231
232
233
234
235
236
237
238
239
239
240
241
242
243
244
245
246
247
248
249
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
387
388
389
389
390
391
392
393
394
395
396
397
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
847
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
868
869
869
870
871
872
873
874
875
876
877
878
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
946
947
947
948
949
949
950
951
952
953
954
955
956
957
957
958
959
959
960
961
962
963
964
965
966
967
967
968
969
969
970
971
972
973
974
975
976
977
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1001
1002
1003
1003
1004
1005
1005
1006
1007
1007
1008
1009
1009
1010
1011
1011
1012
1013
1013
1014
1015
1015
1016
1017
1017
1018
1019
1019
1020
1021
1021
1022
1023
1023
1024
1025
1025
1026
1027
1027
1028
1029
1029
1030
1031
1031
1032
1033
1033
1034
1035
1035
1036
1037
1037
1038
1039
1039
1040
1041
1041
1042
1043
1043
1044
1045
1045
1046
1047
1047
1048
1049
1049
1050
1051
1051
1052
1053
1053
1054
1055
1055
1056
1057
1057
1058
1059
1059
1060
1061
1061
1062
1063
1063
1064
1065
1065
1066
1067
1067
1068
1069
1069
1070
1071
1071
1072
1073
1073
1074
1075
1075
1076
1077
1077
1078
1079
1079
1080
1081
1081
1082
1083
1083
1084
1085
1085
1086
1087
1087
1088
1089
1089
1090
1091
1091
1092
1093
1093
1094
1095
1095
1096
1097
1097
1098
1099
1099
1100
1101
1101
1102
1103
1103
1104
1105
1105
1106
1107
1107
1108
1109
1109
1110
1111
1111
1112
1113
1113
1114
1115
1115
1116
1117
1117
1118
1119
1119
1120
1121
1121
1122
1123
1123
1124
1125
1125
1126
1127
1127
1128
1129
1129
1130
1131
1131
1132
1133
1133
1134
1135
1135
1136
1137
1137
1138
1139
1139
1140
1141
1141
1142
1143
1143
1144
1145
1145
1146
1147
1147
1148
1149
1149
1150
1151
1151
1152
1153
1153
1154
1155
1155
1156
1157
1157
1158
1159
1159
1160
1161
1161
1162
1163
1163
1164
1165
1165
1166
1167
1167
1168
1169
1169
1170
1171
1171
1172
1173
1173
1174
1175
1175
1176
1177
1177
1178
1179
1179
1180
1181
1181
1182
1183
1183
1184
1185
1185
1186
1187
1187
1188
1189
1189
1190
1191
1191
1192
1193
1193
1194
1195
1195
1196
1197
1197
1198
1199
1199
1200
1201
1201
1202
1203
1203
1204
1205
1205
1206
1207
1207
1208
1209
1209
1210
1211
1211
1212
1213
1213
1214
1215
1215
1216
1217
1217
1218
1219
1219
1220
1221
1221
1222
1223
1223
1224
1225
1225
1226
1227
1227
1228
1229
1229
1230
1231
1231
1232
1233
1233
1234
1235
1235
1236
1237
1237
1238
1239
1239
1240
1241
1241
1242
1243
1243
1244
1245
1245
1246
1247
1247
1248
1249
1249
1250
1251
1251
1252
1253
1253
1254
1255
1255
1256
1257
1257
1258
1259
1259
1260
1261
1261
1262
1263
1263
1264
1265
1265
1266
1267
1267
1268
1269
1269
1270
1271
1271
1272
1273
1273
1274
1275
1275
1276
1277
1277
1278
1279
1279
1280
1281
1281
1282
1283
1283
1284
1285
1285
1286
1287
1287
1288
1289
1289
1290
1291
1291
1292
1293
1293
1294
1295
1295
1296
1297
1297
1298
1299
1299
1300
1301
1301
1302
1303
1303
1304
1305
1305
1306
1307
1307
1308
1309
1309
1310
1311
1311
1312
1313
1313
1314
1315
1315
1316
1317
1317
1318
1319
1319
1320
1321
1321
1322
1323
1323
1324
1325
1325
1326
1327
1327
1328
1329
1329
1330
1331
1331
1332
1333
1333
1334
1335
1335
1336
1337
1337
1338
1339
1339
1340
1341
1341
1342
1343
1343
1344
1345
1345
1346
1347
1347
1348
1349
1349
1350
1351
1351
1352
1353
1353
1354
1355
1355
1356
1357
1357
1358
1359
1359
1360
1361
1361
1362
1363
1363
1364
1365
1365
1366
1367
1367
1368
1369
1369
1370
1371
1371
1372
1373
1373
1374
1375
1375
1376
1377
1377
1378
1379
1379
1380
1381
1381
1382
1383
1383
1384
1385
1385
1386
1387
1387
1388
1389
1389
1390
1391
1391
1392
1393
1393
1394
1395
1395
1396
1397
1397
1398
1399
1399
1400
1401
1401
1402
1403
1403
1404
1405
1405
1406
1407
1407
1408
1409
1409
1410
1411
1411
1412
1413
1413
1414
1415
1415
1416
1417
1417
1418
1419
1419
1420
1421
1421
1422
1423
1423
1424
1425
1425
1426
1427
1427
1428
1429
1429
1430
1431
1431
1432
1433
1433
1434
1435
1435
1436
1437
1437
1438
1439
1439
1440
1441
1441
1442
1443
1443
1444
1445
1445
1446
1447
1447
1448
1449
1449
1450
1451
1451
1452
1453
1453
1454
1455
1455
1456
1457
1457
1458
1459
1459
1460
1461
1461
1462
1463
1463
1464
1465
1465
1466
1467
1467
1468
1469
1469
1470
1471
1471
1472
1473
1473
1474
1475
1475
1476
1477
1477
1478
1479
1479
1480
1481
1481
1482
1483
1483
1484
1485
1485
1486
1487
1487
1488
1489
1489
1490
1491
1491
1492
1493
1493
1494
1495
1495
1496
1497
1497
1498
1499
1499
1500
1501
1501
1502
1503
1503
1504
1505
1505
1506
1507
1507
1508
1509
1509
1510
1511
1511
1512
1513
1513
1514
1515
1515
1516
1517
1517
1518
1519
1519
1520
1521
1521
1522
1523
1523
1524
1525
1525
1526
1527
1527
1528
1529
1529
1530
1531
1531
1532
1533
1533
1534
1535
1535
1536
1537
1537
1538
1539
1539
1540
1541
1541
1542
1543
1543
1544
1545
1545
1546
1547
1547
1548
1549
1549
1550
1551
1551
1552
1553
1553
1554
1555
1555
1556
1557
1557
1558
1559
1559
1560
1561
1561
1562
1563
1563
1564
1565
1565
1566
1567
1567
1568
1569
1569
1570
1571
1571
1572
1573
1573
1574
1575
1575
1576
1577
1577
1578
1579
1579
1580
1581
1581
1582
1583
1583
1584
1585
1585
1586
1587
1587
1588
1589
1589
1590
1591
1591
1592
1593
1593
1594
1595
1595
1596
1597
1597
1598
1599
1599
1600
1601
1601
1602
1603
1603
1604
1605
1605
1606
1607
1607
1608
1609
1609
1610
1611
1611
1612
1613
1613
1614
1615
1615
1616
1617
1617
1618
1619
1619
1620
1621
1621
1622
1623
1623
1624
1625
1625
1626
1627
1627
1628
1629
1629
1630
1631
1631
1632
1633
1633
1634
1635
1635
1636
1637
1637
1638
1639
1639
1640
1641
1641
1642
1643
1643
1644
1645
1645
1646
1647
1647
1648
1649
1649
1650
1651
1651
1652
1653
1653
1654
1655
1655
1656
1657
1657
1658
1659
1659
1660
1661
1661
1662
1663
1663
1664
1665
1665
1666
1667
1667
1668
1669
1669
1670
1671
1671
1672
1673
1673
1674
1675
1675
1676
1677
1677
1678
1679
1679
1680
1681
1681
1682
1683
1683
1684
1685
1685
1686
1687
1687
1688
1689
1689
1690
1691
1691
1692
1693
1693
1694
1695
1695
1696
1697
1697
1698
1699
1699
1700
1701
1701
1702
1703
1703
1704
1705
1705
1706
1707
1707
1708
1709
1709
1710
1711
1711
1712
1713
1713
1714
1715
1715
1716
1717
1717
1718
1719
1719
1720
1721
1721
1722
1723
1723
1724
1725
1725
1726
1727
1727
1728
1729
1729
1730
1731
1731
1732
1733
1733
1734
1735
1735
1736
1737
1737
1738
1739
1739
1740
1741
1741
1742
1743
1743
1744
1745
1745
1746
1747
1747
1748
1749
1749
1750
1751
1751
1752
1753
1753
1754
1755
1755
1756
1757
1757
1758
1759
1759
1760
1761
1761
1762
1763
1763
1764
1765
1765
1766
1767
1767
1768
1769
1769
1770
1771
1771
1772
1773
1773
1774
1775
1775
1776
1777
1777
1778
1779
1779
1780
1781
1781
1782
1783
1783
1784
1785
1785
1786
1787
1787
1788
1789
1789
1790
1791
1791
1792
1793
1793
1794
1795
1795
1796
1797
1797
1798
1799
1799
1800
1801
1801
1802
1803
1803
1804
1805
1805
1806
1807
1807
1808
1809
1809
1810
1811
1811
1812
1813
1813
1814
1815
1815
1816
1817
1817
1818
1819
1819
1820
1821
1821
1822
1823
1823
1824
1825
1825
1826
1827
1827
1828
1829
1829
1830
1831
1831
1832
1833
1833
1834
1835
1835
1836
1837
1837
1838
1839
1839
1840
1841
1841
1842
1843
1843
1844
1845
1845
1846
1847
1847
1848
1849
1849
1850
1851
1851
1852
1853
1853
1854
1855
1855
1856
1857
1857
1858
1859
1859
1860
1861
1861
1862
1863
1863
1864
1865
1865
1866
1867
1867
1868
1869
1869
1870
1871
1871
1872
1873
1873
1874
1875
1875
1876
1877
1877
1878
1879
1879
1880
1881
1881
1882
1883
1883
1884
1885
1885
1886
1887
1887
1888
1889
1889
1890
1891
1891
1892
1893
1893
1894
1895
1895
1896
1897
1897
1898
1899
1899
1900
1901
1901
1902
1903
1903
1904
1905
1905
1906
1907
1907
1908
1909
1909
1910
1911
1911
1912
1913
1913
1914
1915
1915
1916
1917
1917
1918
1919
1919
1920
1921
1921
1922
1923
1923
1924
1
```

A screenshot of a code editor window titled "TherapistNotes.js M X". The code is written in JavaScript and defines a component named "TherapistNotes". It uses various state hooks like useState and useEffect. A specific line of code, "useEffect(() => { ... })", is highlighted with a blue background. The status bar at the bottom shows "Ln 25, Col 23" and other standard editor settings.

```
src > pages > Therapist > TherapistNotes.js > TherapistNotes
1 > import { useEffect, useState } from "react";...
8
9
10 function TherapistNotes() {
11   const [dataStudent, setDataStudent] = useState([]);
12   const [searchQuery, setSearchQuery] = useState("");
13   const [selectedUser, setSelectedUser] = useState(null);
14   const [notes, setNotes] = useState({}); // Initialize notes as an object to store notes by user ID
15   const [openModal, setOpenModal] = useState(false);
16   const [noteTitle, setNoteTitle] = useState("");
17   const [noteDescription, setNoteDescription] = useState("");
18   const [txtModelBtn, setTxtModelBtn] = useState("Save"); // Note state
19   // add them for 3 dots
20   const [anchorEl, setAnchorEl] = useState(null);
21   const [selectedNote, setSelectedNote] = useState(null);
22   // add them for 3 dots
23
24
25 //Get all users data| (Chats)
26 > useEffect(() => { ...
39 }, []);
40
41 //Filtering users for teh Search
42
43
44 > const filteredUsers = dataStudent.filter((user) => { ...
51 });
52 > const getNotes = async () => { ...
66 > useEffect(() => { ...
67 }, [selectedUser]);
75
76
```

Figure 5-9: therapistnotes.js code snippet

A screenshot of a code editor window titled "SocialHub.js M X". The code is written in JavaScript and defines a component named "SocialHub". It uses useState and useEffect hooks. A specific line of code, "const handleMenuClose = () => { ... }", is highlighted with a blue background. The status bar at the bottom shows "Ln 26, Col 4" and other standard editor settings.

```
src > pages > SocialHub > SocialHub.js > SocialHub
12 function SocialHub() {
13
14
15   const [data, setData] = useState([]);
16   const [messageInfo, setMessageInfo] = useState(null);
17   const [menuAnchorEl, setMenuAnchorEl] = useState(null);
18   const [selectedMsg, setSelectedMsg] = useState(null);
19   const chatContainerRef = useRef(null);
20
21   const handleMenuClick = (event, msg) => {
22     setMenuAnchorEl(event.currentTarget);
23     setSelectedMsg(msg);
24   };
25
26 > const handleMenuClose = () => { ...
29 };
30
31 const handleSendMessage = () => {
32   if (messageInfo) {
33     JUniVerseAxios.put(`public-chat/${messageInfo.id}`, { content: messageInfo.content })
34       .then(() => {
35         setMessageInfo(null);
36         setRefreshPage(refreshPage + 1);
37       })
38       .catch(console.log);
39   } else {
40     JUniVerseAxios.post("public-chat/message", { content: message })
41       .then(() => {
42         setMessage("");
43         setRefreshPage(refreshPage + 1);
44       })
45       .catch(console.log);
46   }
47 }
```

Figure 5-10: social hub.js code snippet

## Graphical User Interfaces

This section presents high-fidelity GUI designs of our website, these designs give a clear and detailed view of how the final website looks and works, including the layout, colors, buttons, and navigation. They help explain the user experience and show what users can expect when using the platform.



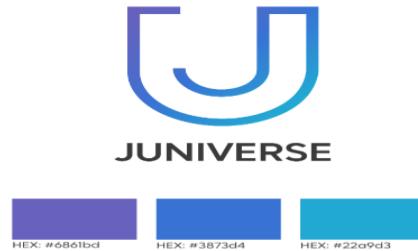
Figure 5-11: homepage UI screenshot

## About Us

Welcome to JUniVerse, a platform designed to empower and connect the bright minds of the King Abdullah II School of Information Technology. Our goal is to create a friendly online space where students can work together, share knowledge, and come up with new ideas. We want to help make your learning and social life even better.

JUniVerse is a comprehensive platform designed to support your academic and personal journey. It provides you with the opportunity to chat with others, exchange files, access mental health support, and stay up-to-date with the latest news, all in one place. Whether you're looking to connect with peers, seek guidance, or stay informed, JUniVerse helps you explore new ideas, grow personally and academically, and connect with the resources that matter most.

[About the logo](#)



### Logo Concept Explanation

The colors in the JUNIVERSE logo are inspired by the galaxy—glowing, colorful, and full of energy. Just like space is full of stars and planets, JUNIVERSE is full of ideas, creativity, and connection.

This platform is made for KASIT students to have their own universe—a place where they can learn, share, collaborate, and grow together during their university journey.

Figure 5-12: about us UI screenshot

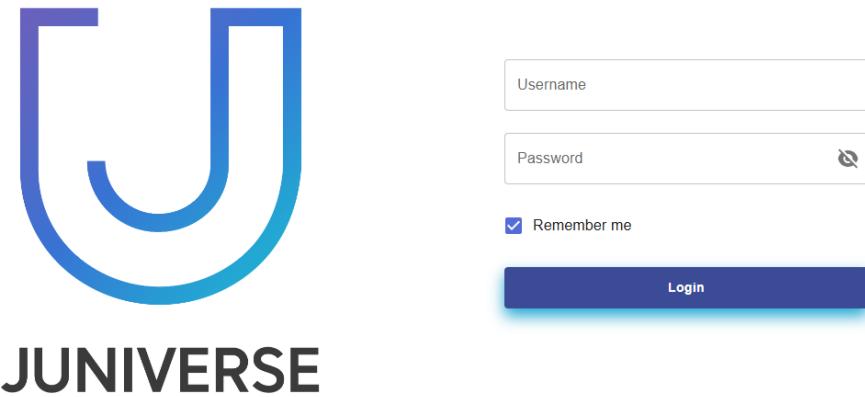


Figure 5-13: login UI screenshot

The screenshot shows a user profile page. At the top, there's a navigation bar with the Juniverse logo, social links, mental health news, file sharing, and e-card options. A notification bell icon with a '9' is visible. The main header features a laptop icon with the code 'console.log('LOVE');' and the word 'CODING' in large white letters. Below the header is a circular profile picture of a young man named Hassan Fadi (STUDENT). His bio information is listed as 'Major: BIT' and 'Bio: i love codin'. To the right is an 'EDIT PROFILE' button. The 'Today's quote' section contains a quote by Steve Jobs: "When we strive to become better than we are, everything around us becomes better too." There's a text input field for updating the quote and a blue 'Update Quote' button. The 'To-Do list' section has a placeholder 'Add a task...' and a blue 'Add Task' button.

This screenshot shows two file management sections. On the left, the 'Uploaded Files' section displays a table with columns for ID, File Name, Uploaded At, and Accepted At. It lists two files: 'All2.png' (uploaded at 5/26/2025 10:50:23 PM, accepted at 5/26/2025 10:50:57 PM) and 'AWS\_Certificatepdf.pdf' (uploaded at 5/26/2025 10:49:57 PM, accepted at 5/26/2025 10:50:53 PM). On the right, the 'Pending Files' section shows a similar table with four entries: 'aws-academy-graduate-aws-acade...', 'Examples.pptx', 'Chapter1.docx', and 'Excel.xlsx', all uploaded at 5/26/2025 10:51:23 PM. Both sections include navigation arrows and page numbers (1-2 of 2 for uploads, 1-4 of 4 for pending).

Figure 5-14: profile UI screenshot

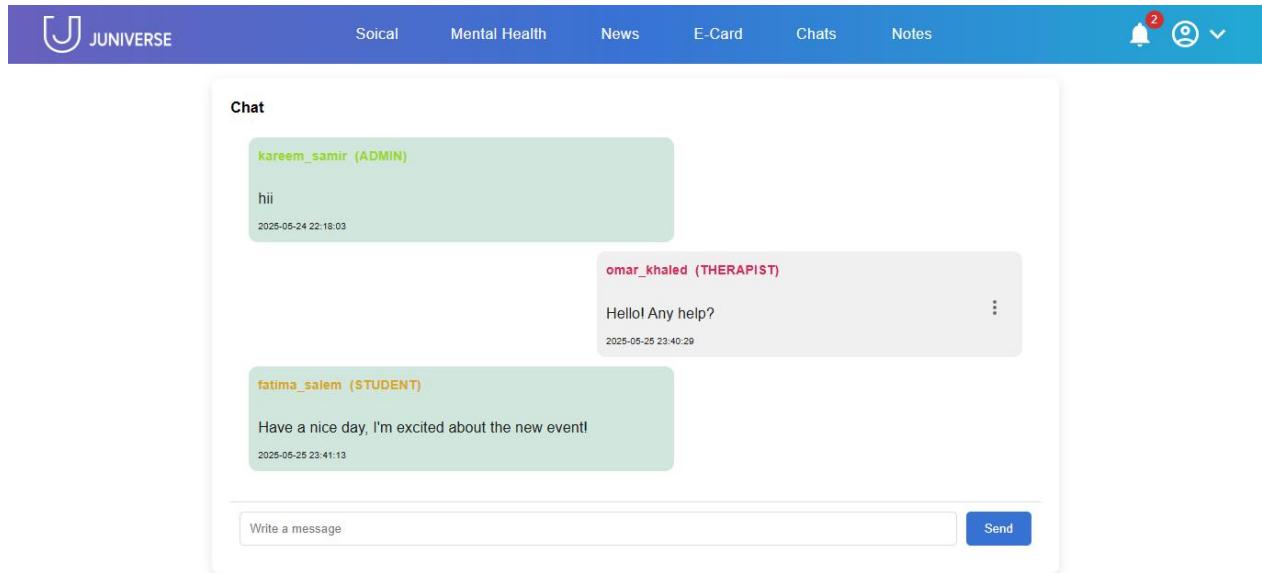


Figure 5-15: public chat UI screenshot

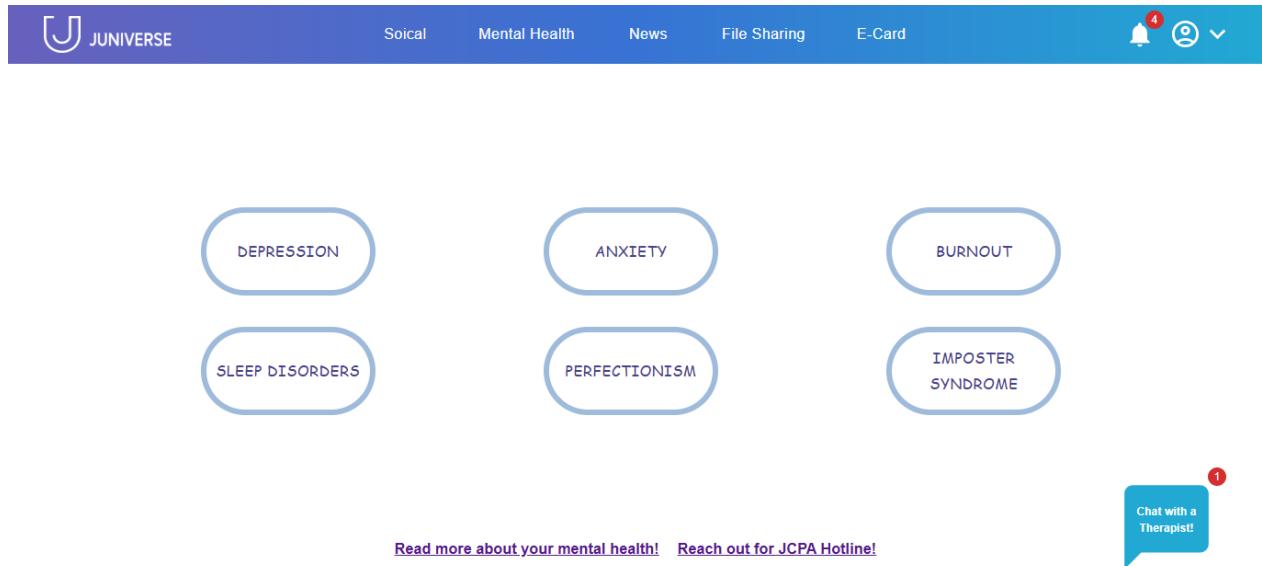


Figure 5-16: mental health hub UI screenshot

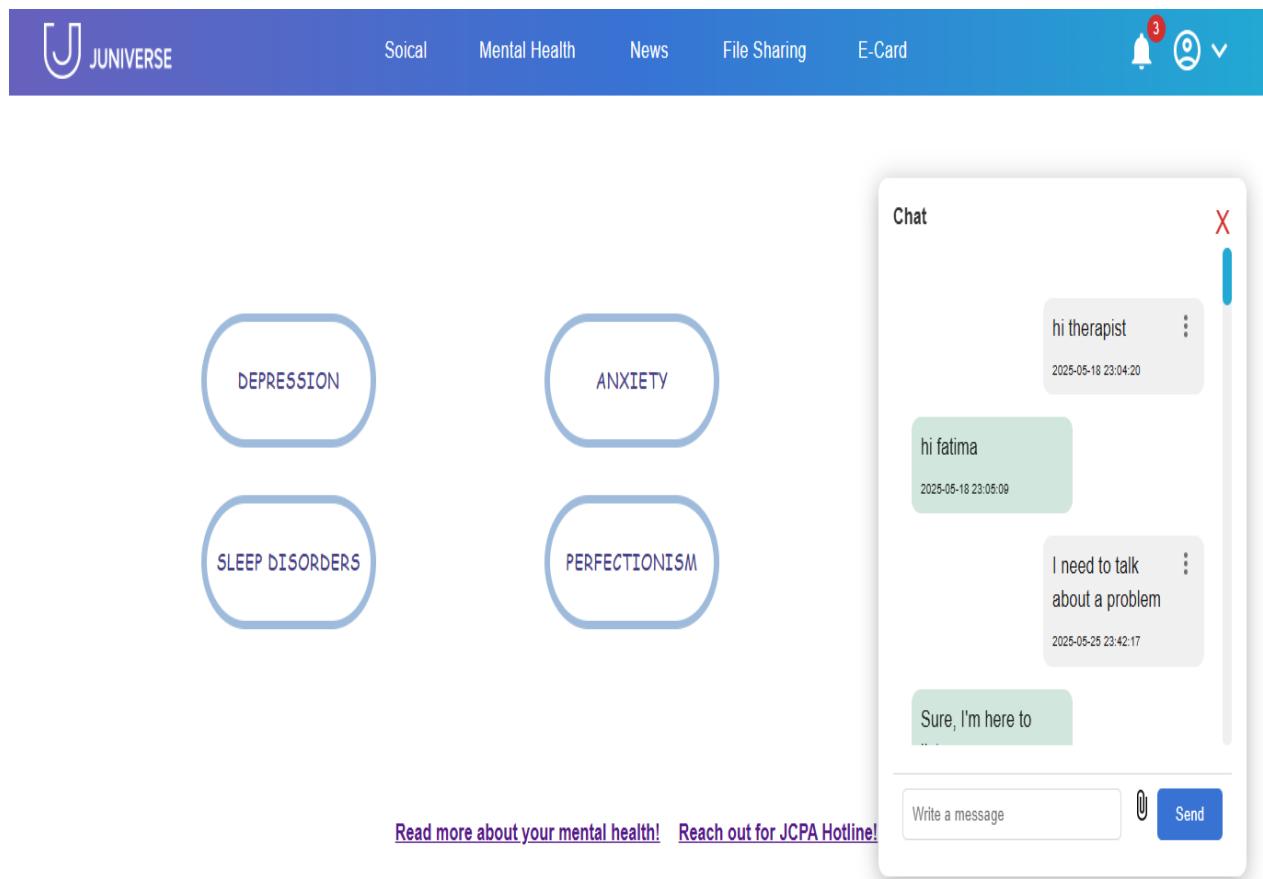
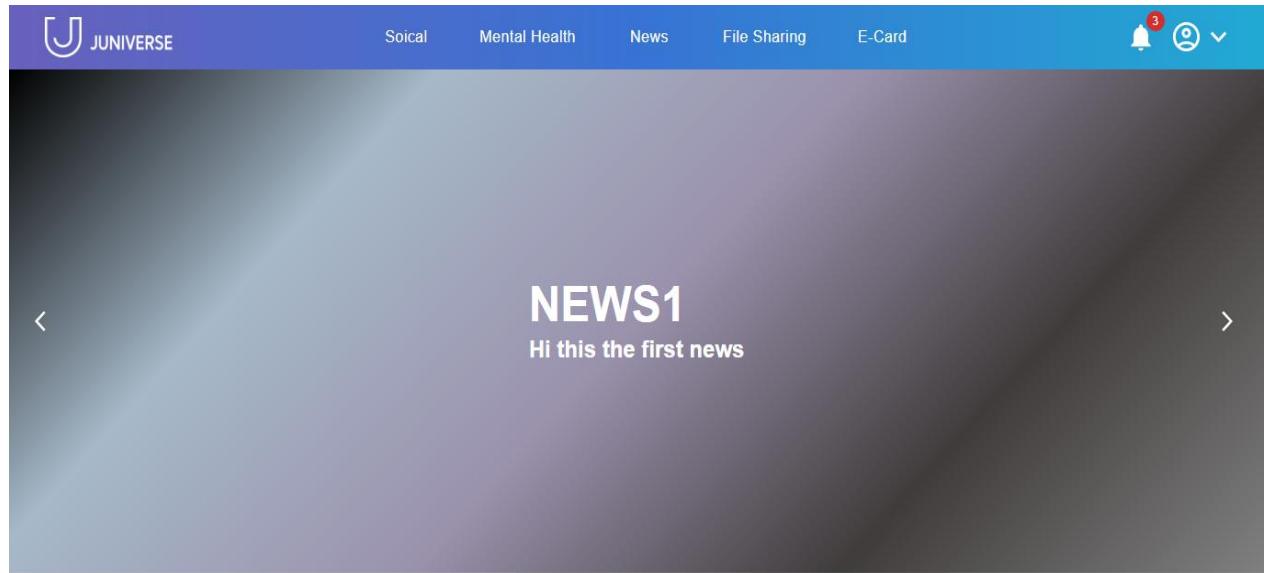


Figure 5-17: private chat with a therapist UI screenshot



Stay connected with what's going on in KASIT school!

May 2025							Month	
CURRENT		BACK	NEXT					
Sun	Mon	Tue	Wed	Thu	Fri	Sat		
	27	28	29	30	01	02	03	
	04	05	06	07	08	09	10	
	11	12	13	14	15	16	17	
<b>asdasd</b> asd Location: asdads		مسنونه المنسون Location: المسنون						
	18	19	20	21	22	23	24	
<b>Celebrate</b> JORDAN Location: JU		<b>asasdasdasd</b> asdasdasdasd Location: asdasdasd						
	25	26	27	28	29	30	31	

Figure 5-18: news hub UI screenshot



Figure 5-19: filesharing hub UI screenshot

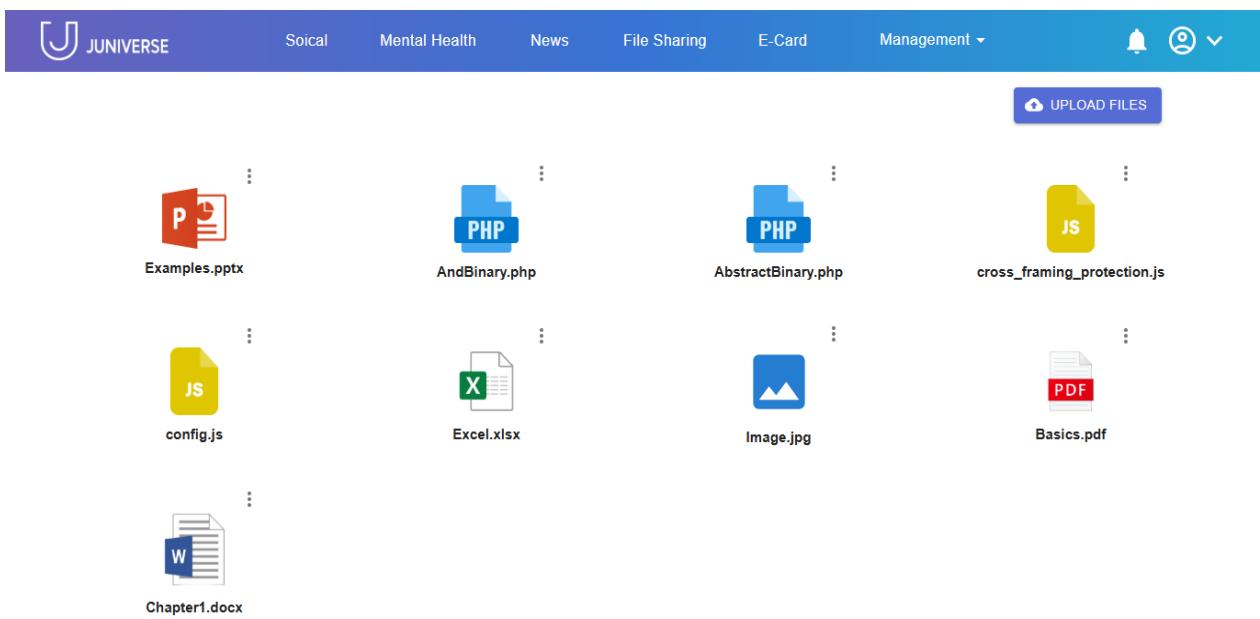


Figure 5-20: filesharing hub UI screenshot 2

The screenshot shows the Juniverse platform's file management interface. At the top, there is a blue header bar with the Juniverse logo, navigation links for Social, Mental Health, News, File Sharing, E-Card, and Files Management, and a user profile icon with a notification count of 6.

The main content area features a table with the following columns: Uploader, Folder Name, File, Download, and Action. The table lists four entries:

Uploader	Folder Name	File	Download	Action
hana_nour	HCI	adpic.png	<a href="#">DOWNLOAD</a>	<a href="#">ACCEPT</a> <a href="#">REJECT</a>
fatima_salem	C++	Basics.pdf	<a href="#">DOWNLOAD</a>	<a href="#">ACCEPT</a> <a href="#">REJECT</a>
fatima_salem	C++	AWS_Academy_Graduate___AWS_Acad...	<a href="#">DOWNLOAD</a>	<a href="#">ACCEPT</a> <a href="#">REJECT</a>
hana_nour	HCI	Chapter1.docx	<a href="#">DOWNLOAD</a>	<a href="#">ACCEPT</a> <a href="#">REJECT</a>

At the bottom right of the table area, there is a page navigation indicator showing "1-4 of 4" with left and right arrows.

Figure 5-21: file management UI screenshot

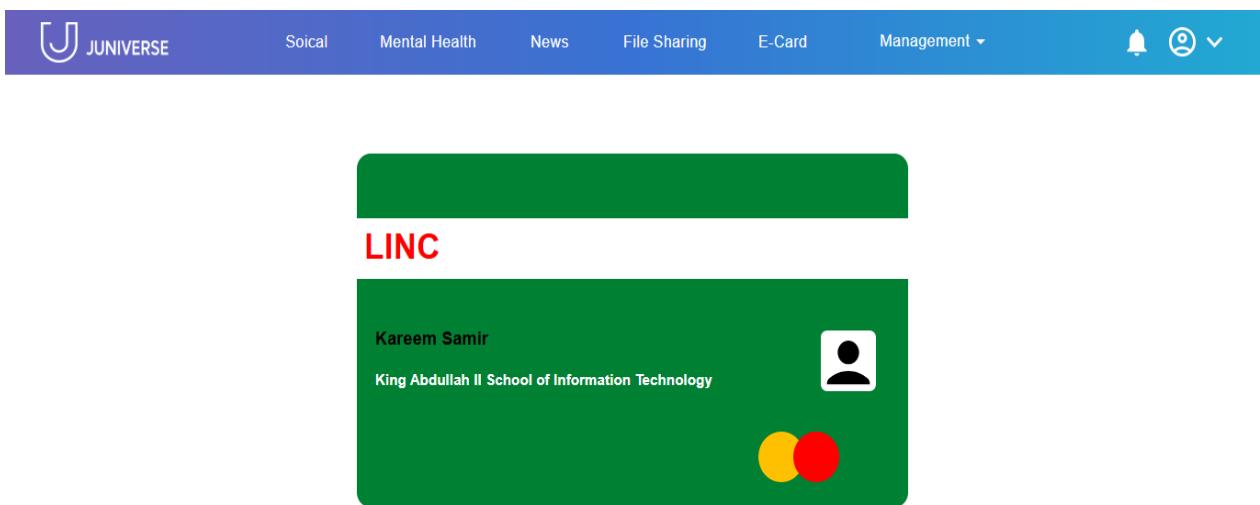


Figure 5-22: e-card UI screenshot

Figure 5-23: therapist chats UI screenshot

Figure 5-24: therapist notes UI screenshot

The screenshot shows a Juniverse student management interface. At the top, there's a navigation bar with links for Soical, Mental Health, News, File Sharing, E-Card, and Management. On the far right of the bar are icons for notifications, user profile, and a dropdown menu.

The main content area displays a table with the following columns: Student's id, First Name, Last Name, Email, and Action. The table contains four rows of student data:

Student's id	First Name	Last Name	Email	Action
2074531	Hana	Nour	hana.nour@example.com	<b>BAN</b> <b>PROMOTE</b>
2170906	Amal	Rami	amal.rami@example.com	<b>BAN</b> <b>PROMOTE</b>
2286741	Reham	Samir	reham.samir@example.com	<b>BAN</b> <b>PROMOTE</b>
2528110	Fatima	Salem	fatima.salem@example.com	<b>BAN</b> <b>PROMOTE</b>

At the bottom of the table, there's a page indicator "1-4 of 4" followed by navigation arrows.

Figure 5-25: student management UI screenshot

The screenshot shows a Juniverse moderators management interface. The layout is similar to Figure 5-25, with a top navigation bar and a main content area featuring a table.

The main content area displays a table with the following columns: Moderator's id, First Name, Last Name, Email, and Action. The table contains seven rows of moderator data:

Moderator's id	First Name	Last Name	Email	Action
2190598	Sara	Tariq	sara.tariq@example.com	<b>BAN</b> <b>DEMOTE</b>
2254155	Hassan	Fadi	hassan.fadi@example.com	<b>BAN</b> <b>DEMOTE</b>
2201702	Tamer	Nabil	tamer.nabil@example.com	<b>BAN</b> <b>DEMOTE</b>
2086763	Fady	Ibrahim	fady.ibrahim@example.com	<b>BAN</b> <b>DEMOTE</b>
2411823	Ahmed	Zaki	ahmed.zaki@example.com	<b>BAN</b> <b>DEMOTE</b>
2254070	Nadia	Hisham	nadia.hisham@example.com	<b>BAN</b> <b>DEMOTE</b>
2311623	Youssef	Mohamed	youssef.mohamed@example.com	<b>BAN</b> <b>DEMOTE</b>

At the bottom of the table, there's a page indicator "1-7 of 7" followed by navigation arrows.

Figure 5-26: moderators management UI screenshot

ID	First Name	Last Name	Email	Role	Action
2109015	Dina	Mahmoud	dina.mahmoud@example.com	MODERATOR	<button>UNBAN</button>
2166874	Layla	Hassan	layla.hassan@example.com	STUDENT	<button>UNBAN</button>
2522135	Mustafa	Rafik	mustafa.rafik@example.com	STUDENT	<button>UNBAN</button>

1-3 of 3 < >

Figure 5-27: banned users management

## 6. CHAPTER SIX:

### SYSTEM TESTING AND INSTALLATION

*Table 6-1: home page test cases*

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>role</b>	<b>Pass/Fail</b>
Home page, case 1	Ensuring opening home page.	-	Show home page, with the welcoming video, the login button on the top right and the footer in the bottom.	All users	pass
Home page, case 2	Ensuring routing to Instagram page.	-	Routing user to JUniVerse Instagram page	All users	pass
Home page, case 3	Ensuring opening about us page	-	About us page with information about JUniVerse	All users	pass
Home page, case 4	Ensuring route to login page	-	Routing user to the login page	All users	pass

Table 6-2: login page test cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass /Fail</b>
Login page, case 1	Ensure that username filed is required	Empty username	Validation error with the following message “Enter a valid username”	All users	pass
Login page, case 2	Ensure that password filed required	Empty password	Validation error with the following message “Enter a valid password”	All users	pass
Login page, case 3	Ensuring that credentials are valid	Invalid credentials	Unauthorized error with the following message “Invalid password or username”	All users	pass
Login page, case 4	Storing credentials in session storage when user checks remember me	Credentials	Remember user when they’re trying to login again	All users	pass
Login page, case 5	Ensure routing to profile page once user clicks login with valid credentials	Valid credentials	Routing to the user profile page	All users	pass

Table 6-3: profile page, test cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
<b>Profile Page Case 1</b>	Ensuring when writing a bio, the written bio is shown	Bio (Text)	The written bio is shown	All users	pass
<b>Profile Page Case 2</b>	Ensuring when choosing a profile picture, the profile picture is shown	Profile Picture (Image)	The chosen profile picture is shown	All users	pass
<b>Profile Page Case 3</b>	Ensuring when choosing a cover picture, the cover picture is shown	Cover Picture (Image)	The chosen cover picture is shown	All users	pass
<b>Profile Page Case 4</b>	Ensuring when clicking on “Cancel” the Bio, Profile picture and cover picture aren’t changed.	Bio, Profile Picture or Cover Picture	The Bio, Profile picture and cover picture aren’t changed.	All users	pass

<b>Profile Page</b> <b>Case 5</b>	Ensuring when clicking on “Save” the updated field is changed	Bio, Profile Picture or Cover Picture	Fields are updated	All users	pass
<b>Profile Page</b> <b>Case 6</b>	Ensuring when clicking on “Remove” profile picture is removed	-	Profile Picture is removed	All users	pass
<b>Profile Page</b> <b>Case 8</b>	Ensuring when clicking on “Remove” cover picture is removed	-	Cover Picture is removed	All users	pass
<b>Profile Page</b> <b>Case 9</b>	Ensuring that the bio field is required	Empty Bio	Validation error with the following message “Bio cannot be empty”	All users	pass

<b>Profile Page</b>	Ensuring when changing the quote, it's updated with the new written one	Quote (Text)	The quote is updated	All users	pass
<b>Profile Page</b>	Ensuring that the user cannot update and empty quote	Empty quote	Quote is not updated	All users	pass
<b>Profile Page</b>	Ensuring when clicking on “Add Task” the written task is added and shown on the To Do List	Task (Text)	Task is added	All users	pass
<b>Profile Page</b>	Ensuring that no task is added when clicking on “Add Task” but leaving the Task field empty	Empty task	Task is not added	All users	pass
<b>Profile Page</b>	Ensuring when checking a task, it's checked and struck through	-	Task is checked and struck through	All users	pass

<b>Profile Page</b>	Ensuring the user can delete a task when clicking on it and clicking delete	-	Task is deleted	All users	pass
<b>Profile Page</b>	Ensuring the pending files are displayed to the user with their details	-	Pending files are displayed	Student	pass
<b>Profile Page</b>	Ensuring the uploaded files are displayed to the user with their details	-	Uploaded files are displayed	Admin, Moderator and Student	pass

Table 6-4: social hub, test cases

Test case/scenario	Objective	Test data	Expected result	Role	Pass/Fail
Social hub, case 1	Ensure message is sent once user clicks <b>send</b>	Not empty Message	Message is shown in chat	All users	pass
Social hub, case 2	Ensure empty messages are not <b>sent</b>	Empty message	Message is not sent, nothing happens	All users	pass
Social hub, case 3	Ensure message is updated once user clicks on <b>edit</b>	Updated message	Edited messages are shown in chat	All users	pass
Social hub, case 4	Ensure admin can <b>delete</b> students and moderator messages	-	Messages get deleted once admin deletes them	Admin	pass
Social hub, case 5	Ensure moderators can <b>delete</b> their own messages and students' message	-	messages get deleted once mod deletes them	Moderator	pass

Table 6-5: mental health hub, test cases

Test case/scenario	Objective	Test data	Expected result	Role	Pass/Fail
Mental health hub, case 1	Ensure popping boxes are popping once clicked on	-	Popped boxes, with texts	All users	<b>pass</b>
Mental health hub, case 2	Ensure users can send message in private chat	message	Message shows in private chat box	Students, moderators, admins	<b>pass</b>
Mental health hub, case 3	Ensure users can edit their message after sending	Edited message	Edited messages are updated in chat	Students' moderators, admins	<b>pass</b>
Mental health hub, case 4	Ensure users can attach files	Attached file	Attached files shown in chat	Students' moderators, admins	<b>pass</b>
Mental health hub, case 5	Ensure files are viewed in another tab once user clicks view	-	File is opened in another tab	Students' moderators, admins	<b>pass</b>

Mental health hub, case 6	Ensure files are downloaded once users clicks download	-	Files downloaded in user pc downloads folder	Students' moderators, admins	<b>pass</b>
Mental health hub, case 7	Ensure mental health links are working when clicked	-	Route user to mental health links.	All users	<b>pass</b>

Table 6-6: news hub, test cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>role</b>	<b>Pass/Fail</b>
News hub, case 1	Ensuring news is switched once user clicks news arrows	-	Next news is shown	All users	pass
News hub, case 2	Ensure users can navigate through events with back and next arrows	-	Next pages in events	All users	pass
News hub, case 3	Ensure admins can add new event	Valid event form data	Event saved and added to events calendar	admin	pass
News hub, case 4	Ensure admin can't pick date and time from the past in event form	Invalid date and time	Validation error with the following message “can't add date from past”	admin	pass
News hub, case 5	Ensure the admin can't leave the	Empty title	Validation error with the	admin	pass

	Title field empty in events form		following message “Title is required”		
News hub, case 6	Ensure the admin can't leave the Description field empty in events form	Empty description	Validation error with the following message “Description is required”	admin	pass
News hub, case 7	Ensure the admin can't leave the Location field empty in events form	Empty location	Validation error with the following message “Location is required”	admin	pass
News hub, case 8	Ensure the admin can't leave the Location field empty in events form	Empty date	Validation error with the following message “Date is required”	admin	pass

News hub, case 9	Ensure the admin can't select a past time in Date field in events form	Past Date	Validation error with the following message "Time must be in the future"	admin	pass
News hub, case 10	Ensure events are deleted successfully	-	Event is deleted and removed from the events calendar	admin	pass

Table 6-7: file sharing hub, test cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>role</b>	<b>Pass/Fail</b>
File Sharing Hub, case 1	Ensuring when clicking on “Add Folder” a folder creation form opens.	-	Folder Creation Form is opened	Admin	pass
File Sharing Hub, case 2	Ensure that the user can't leave “Folder Name” field empty.	Empty Folder Name	Validation error with the following message “Please fill folder name”	Admin	pass
File Sharing Hub, case 3	Ensure that the user can't leave “Folder Description” field empty	Empty Folder Description	Validation error with the following message “Please fill folder description”	Admin	pass
File Sharing Hub, case 4	Ensuring when clicking on “Cancel” no folder is created.	Folder Name and Folder Description	No new folder is created	Admin	pass

File Sharing Hub, case 5	Ensure creating a new folder when clicking on “Save” with filling the required fields of folder creation form	Folder Name and Folder Description	A new folder is created	Admin	pass
File Sharing Hub, case 6	Ensuring when clicking on “Edit” a folder editing form is opened, allowing the user to edit the fields.	-	Folder Editing Form is opened	Admin	pass
File Sharing Hub, case 7	Ensuring validating the user when editing the folder with an empty folder name	Empty folder name	Validation error with the following message “Please fill folder name”	Admin	pass
File Sharing Hub, case 8	Ensuring validating the user when editing the folder with an empty folder description	Empty folder description	Validation error with the following message “Please fill	Admin	pass

			folder description”		
File Sharing Hub, case 8	Ensuring clicking on “Save” after editing folder name and description updates the data	Folder name and folder description	Folder’s name and description are updated successfully	Admin	Pass
File Sharing Hub, case 9	Ensuring clicking “Cancel” during edit does not apply any changes	Folder name and folder description	Folder name and description remain the same	Admin	Pass
File Sharing Hub, case 10	Ensuring clicking on “Delete” shows a confirmation message	-	A confirmation message appears to the user	Admin	Pass
File Sharing Hub, case 11	Ensuring clicking on “Delete” on the confirmation message, deletes the folder	-	The folder is deleted	Admin	Pass

File Sharing Hub, case 12	Ensuring clicking on “Cancel” on the confirmation message, doesn't delete the folder	-	The folder is NOT deleted	Admin	Pass
<b>File Sharing Hub,</b> <b>case 13</b>	Ensuring clicking on a folder shows all the files uploaded on the clicked folder.	-	Files are showed and displayed to users	Admin Moderator and Student	Pass
File Sharing Hub, case 14	Ensuring when clicking on “Upload File” a file upload form opens	-	The file upload form is opened	Admin Moderator and Student	Pass
File Sharing Hub, case 15	Ensure the user can't upload files without choosing one.	No file chosen	Validation error with the following message “Please select a file first!”	Admin Moderator and Student	Pass
File Sharing Hub, case 16	Ensuring user can't upload Unsupported file type	Unsupported file type, for example (exe)	Validation error with the following message:	Admin Moderator and Student	Pass

			“Unsupported file type!”		
File Sharing Hub, case 17	Ensure that the user can't leave “File Name” field empty.	Empty file name	Validation error with the following message “Please fill file name”	Admin Moderator and Student	Pass
File Sharing Hub, case 18	Ensure that the user can't leave “File Description” field empty.	Empty file description	Validation error with the following message “Please fill file description”	Admin Moderator and Student	Pass
File Sharing Hub, case 19	Ensuring when clicking on “Cancel” on file upload form no file is uploaded	File name and file description	No file is uploaded	Admin Moderator and Student	Pass
File Sharing Hub, case 20	Ensuring when clicking on “Save” on file upload form,	File name and file description	The chosen file is uploaded	Admin, Moderator	Pass

	the chosen file is uploaded				
File Sharing Hub, case 21	Ensuring notifying the user after clicking “Save” on file upload form that the file will be moderated before uploaded	File name and file description	Notify the user with the following message: “Your file is being moderated!”	Student	Pass
File Sharing Hub, case 22	Ensuring when clicking on “Edit” and edit file form is opened.	-	Edit file form is opened	Admin and Moderator	Pass
File Sharing Hub, case 23	Ensuring validating the user when trying to edit a file with an empty name	Empty file name	Validation error with the following message “Please fill file name”	Admin and Moderator	Pass
File Sharing Hub, case 24	Ensuring validating the user when trying to edit a file with an empty description	Empty file description	Validation error with the following message “Please fill	Admin and Moderator	Pass

			file description”		
File Sharing Hub, case 25	Ensuring clicking on “Save” after editing file name and description updates the data	File name and file description	Files’ name and description are updated successfully	Admin and Moderator	Pass
File Sharing Hub, case 26	Ensuring clicking on “Cancel” after editing file name and description doesn’t update the data	File name and description	File’s name and description remain the same	Admin and Moderator	Pass
File Sharing Hub, case 27	Ensuring clicking on ”view” opens the file in another tab	-	File is opened another tab	Admin, Moderator and Student	Pass
File Sharing Hub, case 28	Ensuring clicking on ”download” downloads the file	-	The file is downloaded	Admin, Moderator and Student	Pass
File Sharing Hub, case 29	Ensuring clicking on “Delete” on file shows a	-	A confirmation message	Admin and Moderator	Pass

	confirmation message		appears to the user		
File Sharing Hub, case 30	Ensure clicking on “Delete” on the confirmation message, deletes the file.	-	File is deleted	Admin and Moderator	Pass
File Sharing Hub, case 31	Ensure clicking on “Cancel” on the confirmation message, doesn't delete the file.	-	File is NOT deleted	Admin and Moderator	Pass

Table 6-8: E-card test cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
E-Card Hub, case 1	Ensuring displaying the full name of the logged user on the E-Card	-	Full Name is displayed on E- Card	All users	pass

Table 6-9: News management Test Cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
News management, case 1	Ensuring the user is able to fill “News title and content after clicking on “ADD NEWS”	-	Display news form with required fields (Title and Content)	Admin	Pass
News management, case 2	Ensure news cannot be added when the title or the content is empty	Empty Title or Content	Save button is disabled. News is not added.	Admin	Pass
News management, case 3	Ensure news is added successfully after clicking “Save” with valid title and content	Title and Content	Save button is enabled. News is added and published in the news hub.	Admin	Pass
News management, case 4	Ensuring when clicking on “Cancel” no news is added	Title and Content	No news is added	Admin	Pass
News management, case 5	Ensure edited news cannot be saved with empty title or content	Empty Title or Content	No news is edited and updated.	Admin	Pass

News management, case 6	Ensure news is updated successfully after clicking “Save” with valid title and content	Title and Content	Save button is enabled. News is updated successfully.	Admin	Pass
News management, case 7	Ensuring when clicking on “Cancel” discards changes and the news is not edited.	Title and Content	No news is edited or updated.	Admin	Pass
News management, case 8	Ensuring the user is notified with a confirmation message when clicking on “Delete”	-	A confirmation message about deleting the news is displayed	Admin	Pass
News management, case 9	Ensure no news is deleted if deleted is canceled	-	News remains in the list.		
News management, case 10	Ensure news is deleted when confirmed	-	The news is removed from the news hub.	Admin	Pass

*Table 6-10:students management, test cases*

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
Students' management, case 1	Ensure students are promoted to moderator when promotion is confirmed	-	Students are removed from Students List and added to Moderators List.	Admin	Pass
Students' management, case 2	Ensure when clicking on “Cancel” the student remains in Students list and isn't promoted	-	Student isn't promoted	Admin	Pass
Students' management, case 3	Ensure students are banned and removed from list when ban is confirmed	-	Students are banned from the platform.	Admin	Pass
Students' management, case 4	Ensure no action occurs if the ban is canceled and the student remains in Students List	-	Students are not banned and kept on the platform.	Admin	Pass

*Table 6-11: moderators management, test cases*

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
Moderators' management, case 1	Ensure moderator is demoted to students when demotion is confirmed	-	Moderator is removed from Moderators List and added to Students List.	Admin	Pass
Moderators' management, case 2	Ensure when clicking on “Cancel” the moderator remains in Moderators list and isn't demoted.	-	Moderator isn't demoted	Admin	Pass
Moderators' management, case 3	Ensure moderator is banned and removed from list when ban is confirmed	-	Moderators are banned.	Admin	Pass
Moderators' management, case 4	Ensure no action occurs if the ban is canceled and moderators remain in Moderators List	-	Moderators are not banned and kept on the platform.	Admin	Pass

*Table 6-12: banned users management, test cases*

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
Banned users Management, case 1	Ensure STUDENT is unbanned and returned to students list when Unban is confirmed	-	Students return to the platform and appear in students list	Admin	Pass
Banned users management, case 2	Ensure MODERATOR is unbanned and returned to moderators list when Unban is confirmed	-	Moderator is returned to the platform and appears in moderators list	Admin	Pass
Banned users' management, case 3	Ensure when clicking on “Cancel” the user remains in Banned users list and isn't unbanned from the platform.	-	Users are still banned from the platform.	Admin	Pass

Table 6-13: file management, test cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
Files management, Case 1	Ensure when clicking on a file, it opened in a new tab to be viewed	-	File is opened in a new tab	Moderator	Pass
Files management, Case 2	Ensure clicking on “Download” starts downloading the file	-	File is downloaded	Moderator	Pass
Files management, Case 3	Ensure the file is accepted after user clicks on “Accept” and confirms with “Yes”	-	File is accepted and published on File Sharing Hub	Moderator	Pass
Files management, Case 4	Ensure the file is NOT accepted after user clicks on “Cancel”	-	Files are NOT accepted and remain on the pending files list.	Moderator	Pass

Files management, Case 5	Ensure the file is rejected after user clicks on “Reject” and confirms with “Yes”	-	The file is rejected.	Moderator	Pass
Files management, Case 6	Ensure the file is NOT rejected after user clicks on “Cancel”	-	Files are NOT rejected and remain on the pending files list.	Moderator	Pass

Table 6-14: therapist chats, test cases

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
Therapist chats, case 1	Ensure all therapist chats are retrieved once therapists enter chats tab	-	List all initiated therapist chats	therapist	pass
Therapist chats, case 2	Ensure the therapist can search users via the search bar successfully	-	The searched user chat should be shown on top.	therapist	pass
Therapist chats, case 3	Ensure the therapist can send messages successfully to the other party	Valid message	Message should be sent, and displayed in chat in the sender side	Therapist	pass
Therapist chats, case 4	Ensure the therapist couldn't send an empty message	Empty message	Message should not be sent.	Therapist	pass
Therapist chats, case 5	Ensure the therapist can edit his message once sent	Edited message	The message should be updated in database then displayed in chat	Therapist	pass

Therapist chats, case 6	Ensure the therapist can delete their own messages once sent	-	Message should be deleted from database accordingly removed from chat	Therapist	pass
Therapist chats, case 7	Ensure the therapist can attach files to chat	file	File should be attached and displayed in chat	Therapist	pass
Therapist chats, case 8	Ensure the therapist can view or download the files attached in the chat successfully	-	Files should be viewed on other tabs/ downloaded in therapist downloads folder	Therapist	pass
Therapist chats, case 9	Ensure the therapist can't attach unsupported file extensions to chat	Unsupported file extension	User should be notified “unsupported file format”	Therapist	pass

*Table 6-15: therapist notes, test cases*

<b>Test case/scenario</b>	<b>Objective</b>	<b>Test data</b>	<b>Expected result</b>	<b>Role</b>	<b>Pass/Fail</b>
Therapist notes, case 1	Ensure all users that have private chats have a note record in the notes tab	-	All user notes are retrieved and displayed	Therapist	pass
Therapist notes, case 2	Ensure the therapist can create a note successfully	Valid note title and description	Note should be displayed with other chat notes	Therapist	pass
Therapist notes, case 3	Ensure the therapist can delete their notes	-	Note should be deleted	Therapist	Pass
Therapist notes, case 4	Ensure the therapist can edit their notes	Updated note	Note should be updated and displayed to therapist	Therapist	Pass
Therapist notes, case 5	Ensure the therapist can't upload empty note title or description	Empty note title or description	Upload button should deactivate if data is empty	therapist	pass

Table 6-16: notifications, test cases

Test case/scenario	Objective	Test data	Expected result	Role	Pass/Fail
Notifications, case 1	Notify all users when an event is added	New event	All users have a notification of the added event	All users	pass
Notifications, case 2	Notify the student when their file gets rejected	-	User get notified of the rejected file	students	pass
Notifications, case 3	Notify the student when their file gets accepted	-	User get notified of the accepted file	students	pass
Notifications case 4	Notify users when private messages from therapist is sent them	-	Users get notified of the message	Students, moderators, admins	pass
Notification case 5	Notify the therapist when users contact him	-	The Therapist gets notification of the message	Therapist	pass
Notifications, case 6	Ensure users can navigate to the specific hub the notification occurred at	-	Navigate the user to the specific page.	All users	pass

Notification case 7	Ensure notification gets read once clicked at	-	Notification marked as read, and moved to read notifications list	All users	pass
Notification case 8	Ensure all notification gets read once clicked at 'read all'	-	All the notifications are moved to the read list	All users	pass

## User Manual

Here is a link to our user manual, which provides a simple guide to help you navigate and use our website with ease:

[https://drive.google.com/file/d/1kzM9RfJ2y\\_16gYK\\_ME5JnGTKLMs-J0FN/view?usp=sharing](https://drive.google.com/file/d/1kzM9RfJ2y_16gYK_ME5JnGTKLMs-J0FN/view?usp=sharing)

## **7. CHAPTER SEVEN: PROJECT CONCLUSION AND FUTURE WORK**

### **Introduction:**

In JUniVerse, we built a web platform for students at King Abdullah II School of Information Technology. The goal was to help students connect, share useful resources, and support each other in both academic and social ways. This section talks about what went well, what could be better, and what we plan to do in the future.

### **Overall strengths:**

One of the best things about our project is how it brings everything together in one place. Students can ask questions, share files, read college news, check events, and even get mental health support — all through different hubs.

We also made sure different users (like Admins, Moderators, Therapists, and Students) have different roles and permissions, which helps keep the platform safe and organized.

Overall, the platform makes it easier for students to help each other and feel like part of a community.

### **Overall weakness:**

Even though the project works well, there are a few things that could be improved. Right now, it only works as a website, so students who prefer using their phones might find it less easy to use.

### **Future work:**

In the future, we plan to turn this platform into a mobile app, so users can use it more easily on their phones.

We also plan to expand the platform to include support for graduates. This would give them a space to stay connected, share job opportunities, and offer advice to current students. It would help build a stronger, long-term community that goes beyond just the classroom years.

### **Summary:**

In conclusion, this project gives students one place where they can learn, share helpful resources, and support each other during their time at university, it helps students work together and stay connected, in both their studies and social life.

We are proud of what we have built so far and believe it can make student life better.

Moving forward, we aim to continue enhancing the system with additional features and improvements that will make it even more effective and user-friendly.

## **8. REFERENCES**

- React.js Documentation: <https://react.dev/>
- JSON Web Tokens (JWT) Documentation: <https://jwt.io/introduction>
- Material User Interface (MUI) Documentation: <https://mui.com/>
- Spring Boot Documentation: <https://docs.spring.io/spring-boot/index.html>
- Sommerville, Ian, Software engineering, 10th edition, 2015, Welly:

## 9. Appendix

Select your major:

22 responses

 Copy chart

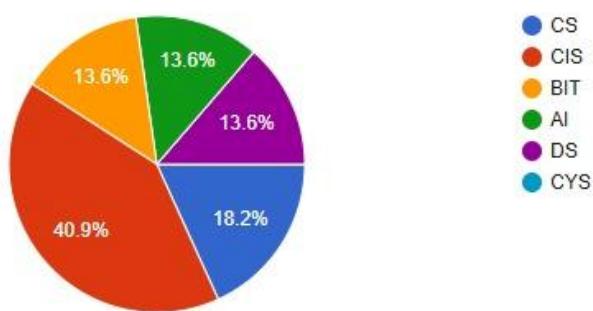
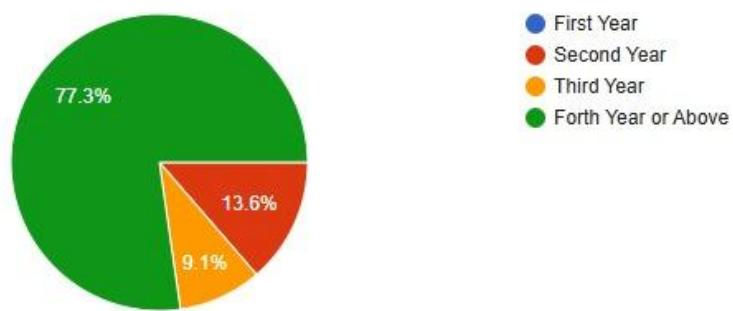


Figure 9-1: survey results

Select your academic year:

22 responses

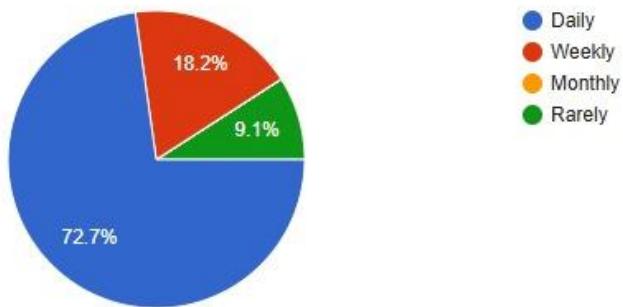
 Copy chart



 Copy chart

### How often do you use online platforms for educational purposes?

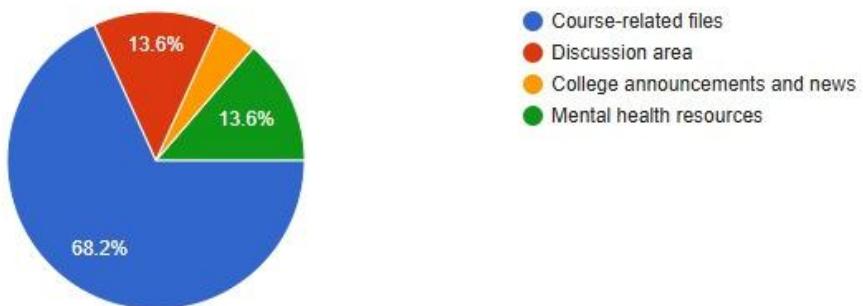
22 responses



 Copy chart

### What type of content do you find most helpful in an online platform?

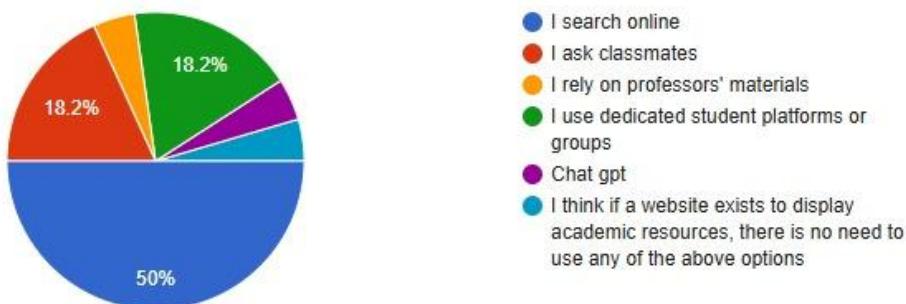
22 responses



 Copy chart

### How do you usually find academic resources (e.g., course materials, study guides)?

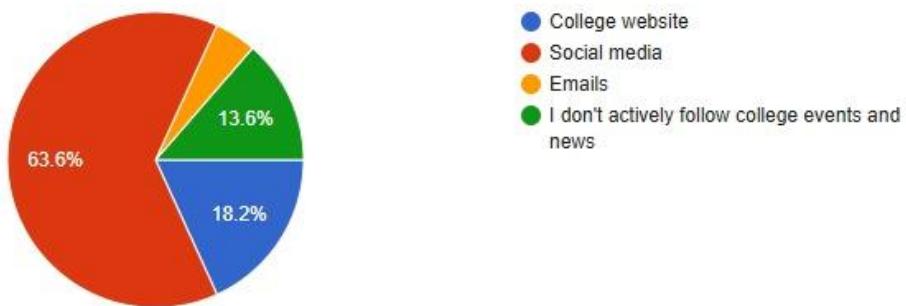
22 responses



 Copy chart

### How do you currently stay updated on college events and news?

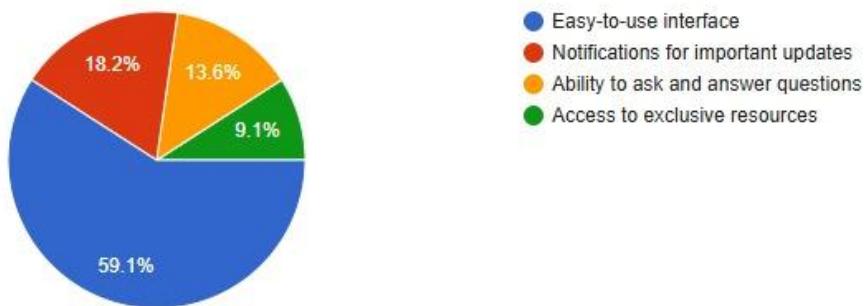
22 responses



**What features would encourage you to engage more on an online platform for students?**

 Copy chart

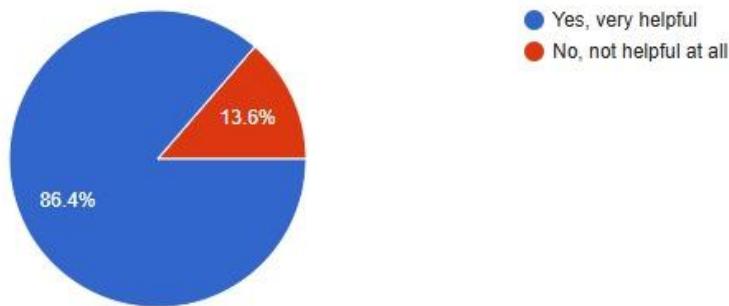
22 responses



**Would you find it helpful to have a space where students can ask questions and get answers from their peers?**

 Copy chart

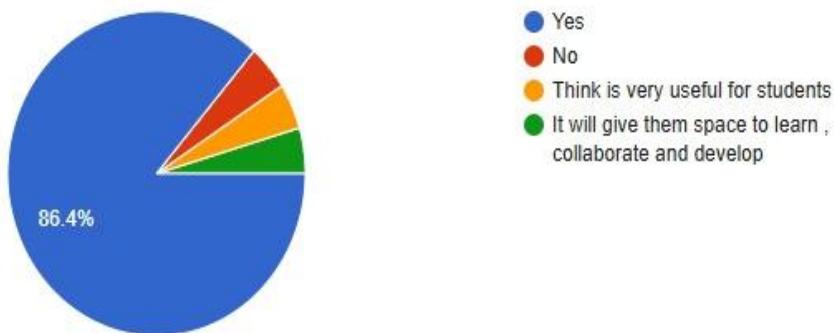
22 responses



 Copy chart

**Do you think a platform with social, academic, and mental health hubs could benefit students at King Abdullah II School of Information Technology?**

22 responses



**The End**