

---

**Development of Counseling Hour Management System for DIU  
BY**

**Tahzid Mahmud Suvo  
ID: 171-15-1272**

**Emon Kumar Das  
ID: 171-15-1300  
AND**

**Md. Tanvir Hasan Anik  
ID: 171-15-1307**

This Report Presented in Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

**Dr. S. M. Aminul Haque**  
Associate Professor  
Department of CSE  
Daffodil International University

Co-Supervised By

**Mr. Tajim Md. Niamat Ullah Akhund**  
Lecturer  
Department of CSE  
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY  
DHAKA, BANGLADESH**

**DECEMBER 2020**

## **APPROVAL**

This project titled “**Development of Counseling Hour Management System for DIU**”, submitted by Tahzid Mahmud Suvo ID: 171-15-1272, Emon Kumar Das ID: 171-15-1300, and Md. Tanvir Hasan Anik ID: 171-15-1307 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on

## **BOARD OF EXAMINERS**

---

**(Name)**

**Chairman**

**Designation**

Department of CSE  
Faculty of Science & Information Technology  
Daffodil International University

---

**(Name)**

**Internal Examiner**

**Designation**

Department of CSE  
Faculty of Science & Information Technology  
Daffodil International University

---

**(Name)**

**External Examiner**

**Designation**

Department of -----  
Jahangirnagar University

## DECLARATION

We hereby declare that this project has been done by us under the supervision of honorable **Dr. S. M. Aminul Haque, Associate Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

**Supervised by:**

---

**Dr. S. M. Aminul Haque**

Associate Professor  
Department of CSE  
Daffodil International University

**Co-Supervised by:**

---

**Mr. Tajim Md. Niamat Ullah Akhund**

Lecturer  
Department of CSE  
Daffodil International University

**Submitted by:**

---

**Tahzid Mahmud Suvo**

ID: 171-15-1272  
Department of CSE  
Daffodil International University

---

**Emon Kumar Das**

ID: 171-15-1300  
Department of CSE  
Daffodil International University

---

**Md. Tanvir Hasan Anik**

ID: 171-15-1307  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty God for His divine blessing makes us possible to complete the final year project successfully.

We grateful and wish our profound indebtedness to **Dr. S. M Aminul Haque, Associate Professor, Department of CSE**, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Research and Development*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Prof. Dr. Syed Akhter Hossain, Head, Department of CSE**, for his kind help to finish our project and also to other faculty members and the staff of the CSE department of Daffodil International University.

We would like to thank our entire course mate at Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

## ABSTRACT

Daffodil International University is one of the top-ranked universities in Bangladesh with a lot number of students and faculties. Day by day Daffodil International University is going to reach a significant landmark position in the area of teaching. The relationships and understandings between the students and the faculties of the university are playing a vital role in these achievements. Faculties are maintaining scheduled classes and counseling time for the students. They also manage many responsible activities for the university's education management. Often students have required some directions, advice, and some solves to their educational problems. They also need to follow-ups and instructions for their projects, researches, and internships. For this requirement, counseling from faculties hasn't any other alternative way to exclude. Proper management for this with flexible time management for both faculties and students is essential. Giving major priority on this dual side affirmation of available meeting time from their both we have developed this project titled "Development of Counseling Hour Management System for DIU". Also, this project includes some helpful services like files, resources, and notices sharing by faculties and admins with many functional live messaging and useful notifications. Tracking all the information on the counseling for both students and faculties can possible with this application. All in one, this project is deployable to a production server environment for a great schedule maintaining, tracking, and connecting services for counseling management.

## LIST OF TABLES

<b>TABLES</b>	<b>PAGE</b>
Table 3.2.1 Requirement Collection and Analysis	11
Table 3.3.2.1 Use Case Description for Login	18
Table 3.3.2.2 Use Case Description for Sign Up	19
Table 3.3.2.3 Use Case Description for Log Out	20
Table 3.3.2.4 Use Case Description for Manage Profile	21
Table 3.3.2.5 Use Case Description for Request Counseling	22
Table 3.3.2.6 Use Case Description Search for Faculty	23
Table 3.3.2.7 Use Case Description for View Counseling List	24
Table 3.3.2.8 Use Case Description for View Request List	25
Table 3.3.2.9 Use Case Description for Manage Request	26
Table 3.3.2.10 Use Case Description for View Notification	27
Table 3.3.2.11 Use Case Description for Manage Message	28
Table 3.3.2.12 Use Case Description for Manage Counseling Hours	30
Table 3.3.2.13 Use Case Description for Manage Counseling Meetings	31
Table 3.3.2.14 Use Case Description for Manage Resource Files	32
Table 3.3.2.15 Use Case Description Search for Students	33
Table 3.3.2.16 Use Case Description View Student Profile	34
Table 3.3.2.17 Use Case Description View Faculty Profile	35
Table 3.3.2.18 Use Case Description Manage Faculty Profile	36
Table 3.3.2.19 Use Case Description manage Student Profile	37
Table 3.3.2.20 Use Case Description Generate Report	38
Table 5.5 Test Result	69

## LIST OF FIGURES

FIGURES	PAGE
Figure 3.1.1 Business Process Model	10
Figure 3.3.1 Student Use Case Diagram	15
Figure 3.3.2 Faculty Use Case Diagram	16
Figure 3.3.3 Admin Use Case Diagram	17
Figure 3.4 Logical Data Model	40
Figure 4.1 Landing Page of the Counseling Portal	42
Figure 4.2 Registration Page	43
Figure 4.3 Login Page	44
Figure 4.4 Home Page for Students	44
Figure 4.5 Get Counseling Page -Request Counseling	45
Figure 4.6 Get Counseling Page -Basic Information	46
Figure 4.7 Request List Page	46
Figure 4.8 All Meetings Page	47
Figure 4.9 Notifications Drop Down List	47
Figure 4.10 Chat Page	48
Figure 4.11 Faculty Home Page	49
Figure 4.12 Request List Page	49
Figure 4.13 All Meeting page	50
Figure 4.14 Chat Page	51
Figure 4.15.1 New Upload Folder Page	51
Figure 4.15.2 Upload Files by Drag and Drop	52
Figure 4.15.3 All Uploaded Folders	52
Figure 4.15.4 Admin Home Page	53
Figure 4.16.1 Faculty List Page	53
Figure 4.16.2 Faculty Profile Page	54
Figure 4.17 Faculty Report Page	54

Figure 4.18 Report Generate Page	55
Figure 4.19 Student List Page	55
Figure 4.20 Chat page	56
Figure 5.1.1 Entity Relationship Diagram	61
Figure 5.1.2 Migration Confirmation Feedback	63
Figure 5.1.3 Migrated Table List Feedback	63
Figure 5.2.1 Mobile Responsive Landing Page	64
Figure 5.2.3 Mobile Responsive Login page	65
Figure 5.2.4 Mobile Responsive Menu	65
Figure 5.3.1 Toaster Alert for Successful Interaction	66
Figure 5.3.2 Delete Modal Confirmation Interaction	66
Figure 5.3.3 Delete Confirmation Toaster Alert	67
Figure 5.3.4 User Details Interactive Model	67
Figure 5.3.5 Live Update of Notification	68



## TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Board of examiners	i
Declaration	ii
Acknowledgments	iii
Abstract	iv
List of Tables	v
List of Figures	vi
<b>CHAPTER</b>	
<b>CHAPTER 1: Introduction</b>	<b>1-3</b>
1.1 Introduction	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	3
<b>Chapter 2: Background</b>	<b>4-9</b>
2.1 Introduction	4
2.2 Related Works	6
2.3 Comparative Studies	6
2.4 Scope of the Problem	7
2.5 Challenges	8

**Chapter 3: Requirement Specification** **10-41**

3.1 Business Process Modeling	10
3.2 Requirement Collection and Analysis	11
3.3 Use Case Modeling and Description	15
3.4 Logical Data Model	40
3.5 Design Requirements	41

**Chapter 4: Design Specification** **42-60**

4.1 Front-end Design	42
4.2 Back-end Design	57
4.3 Interaction Design and UX	58
4.4 Implementation Requirements	59

**Chapter 5: Implementation and Testing** **61-69**

5.1 Implementation of Database	61
5.2 Implementation of Front-end Design	64
5.3 Implementation of Interactions	65
5.4 Testing Implementation	68
5.5 Test Results and Reports	69

<b>Chapter 6: Conclusion and Future Scope</b>	<b>70</b>
6.1 Discussion and Conclusion	70
6.2 Scope for Further Developments	70

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 Introduction:**

Daffodil International University assures quality education for the students. A great responsibility comes to the faculties for maintaining this quality education. The faculties of the university always maintain scheduled classes just in time and participate in many other activities for the university. Students require some directions, advice, information from faculties. Students also have to follow up on some guidelines from their final year research, project, and internship. For these needs and requirements, proper management for counseling is introduced by giving priority to both faculties and students' available time affirmation.

Students can request a specific time of meeting for counseling from faculties available counseling time slots, and faculties can readvise, accept, or reject the counseling request. If faculties have not available at that time for the counseling, they can readvise students another time without accepting it. Students will confirm or dismiss the readvised time as they are free of that time or not. By this dual affirmation process, flexible time management is introduced with a practical solution for better time management.

All the information on the counseling for both students and faculties can be tracked with report generation. Necessary notifications are used for all the interactions that occurred inside the project via bot mail and application.

Some additional and relevant features are also added with this application like a fully functional chat to always keep connected the students and faculties. Faculties and admins can upload files and resources with information so that students can easily access necessary resources from the applications.

All in one, this is a complete project base on the idea of how to serve faculties and students of Daffodil International University during their counseling hour and keep everyone connected.

## **1.2 Motivation:**

There are several motivations behind the development of this project, major motivations behind this project are given below:

- ❖ As the number of faculty members and students is increasing rapidly, it becomes very tough to keep track of the counseling hour of the respective faculty members.
- ❖ Sometimes, a faculty member is too busy with a list of students leading others to wait in a queue.
- ❖ Faculty members can also have an idea about how many students want to have counseling.
- ❖ Fast communication with a faculty member and a student is possible.

## **1.3 Objective:**

Our objectives for this development project are provided below:

- ❖ To provide an online means of counseling hour information of faculty members.
- ❖ To avail a booking system to ensure the physical meeting.
- ❖ To keep track of the counseling status of the students.

## **1.4 Expected Outcome:**

Expectations from this project are listed below:

- ❖ Faculty and Students will have a better platform for communication regarding counseling.
- ❖ Less time consuming and much efficient execution of counseling procedure.

- ❖ Live update of the counseling status via live chat.

### **1.5 Report Layout:**

In this report, we have included six-chapter in total. A summary about this six-chapter is given below:

- i) In the first chapter, we have given a list of the motivation, objective, expected outcomes to get an overview introduction of the development project.
- ii) In the second chapter, we have discussed detailed background about the project and its related works, comparative studies, and challenges.
- iii) In Chapter three we have presented a detailed diagram of the Business Process Model, Requirement Collection and Analysis, Use Case Modeling and Description, Logical Data Model, and Design Requirements.
- iv) In chapter four we have detail discussed Design Specification of the front end, interaction, and UX.
- v) In chapter five we have discussed the implementation and testing processes. Database implementation, front-end design implementation, interaction implementation, testing implementation. results and reports.
- vi) In the last chapter, Chapter no. 6 we have discussed the conclusion and the future scope of the project.

## **CHAPTER 2**

### **BACKGROUND**

#### **2.1 Introduction**

Like other aspects of the study or rather student life counseling with the teachers is also important since it is pretty easy to get lost in the valley of knowledge and get distracted. So, to guide us on the right path and clear our doubts we need our teachers. Here counseling plays an important role since students can discuss and clear their doubts on certain topics through counseling. Counseling also plays a great role in the mental welfare of a student. Daffodil International University is a leading institution among the private Universities of Bangladesh hence it has a large number of students and the number is getting increased every year due to the enrollment of students in a vast variety of programs that this University offers. Now due to the increasing number of student's teachers and students can't interact properly hence problems are occurring regarding counseling hour like students can't keep track of the teacher's counseling hours and often find themselves in front of a locked cabin or in a long queue to get some counseling in the other hand teachers are facing some problems as well like they are spending a day without any appointments or sometimes they are facing a long queue of students to meet them or they are not even prepared for the topic student to want to discuss this is both frustrating and a waste of time for both parties. All of this is happening due to the lack of information about both parties. To solve these problems, we are developing a dynamic web application Counseling Hour Management System for DIU to enhance the current counseling hour management system.

This Application has 3 roles and they are:

1. Faculty
2. Student
3. Admin

In this application, all the users (Faculty, students) have to sign in first with using their designated varsity mail id since a verification mail will be sent for security purposes and after that users can log in and they will be redirected to their respective dashboards to explore the application. They can edit or update their profile details.

Students can search for faculty through their name or by email id after selecting their desired faculty they will be presented with the list of slots for counseling for a meeting. Then the students will request for a counseling time from the list of counseling hours to request a counseling student must give a short description of the meeting intention for more flexibility student can also request a time of their liking for counseling and they can even cancel the request system will always send a mail and a notification to the associated faculty account about the request the faculty can review the request and can also review the student details then the faculty can accept or reject a request for counseling or re-advise a new time for counseling. Students will get the corresponding notification and mail about the request, if the faculty gives a re-advised time the student can accept or reject time-based on his/her time of liking. After the request is accepted by both parties the meeting time is set and before (10 minutes) the meeting both gets a remainder mail and notification for the upcoming meeting. Faculty can then mark the meeting arrived or not based on the student's response. Faculty can also share resource files with the students on the platform. Admin is an independent actor here admin can login with set credentials and then can search for faculty or students can view their profile and remove a faculty or student from the system if they want. Admin gets the facility to get the monthly report about counseling of the faculties or students. But admin has no kind of access in counseling matters between the faculty and student to keep this matter total privacy.



## 2.2 Related Works

Counseling hour management is a specialized case of an appointment management system which is pretty common among first-world countries as well as second world countries. From hospitals to corporate offices they use some sort of appointment management system to manage the time of individuals. However, Hospitals use this type of system more than other sectors for booking appointments for doctors.

In third world countries like ours have already started to use these types of systems as well mostly in hospitals though. In our country, some well-known hospitals are using the appointment system like Centre for the Rehabilitation of the Paralyzed (CRP), Square Hospital, Apollo Hospital, and some other big hospitals are using appointment management systems.

Some foreign embassy is also using online appointment systems like the Appointment system of the German Foreign office – Dhaka and some similar types of organizations are using similar systems.

## 2.3 Comparative Studies

From the above discussion, we can see that similar types of works have already been done in different organizations but none of them are generalized so we had to develop our own from scratch. The systems we have stated are for booking and managing appointments but they not very user-friendly and flexible they are pretty static as well as not enough interactive.

If we take the most reputed hospital as the representative among stated ones above the Apollo hospital's doctor appointment system, we can see that it has fixed slots for appointments to request from but our application gives students this feature to request for an appointment at the time of their liking with the fixed slots.

The system has a restricted communication between the doctors and the patients but our system gives the users this flexibility of communication we have live chat implemented in

our application so both student or teacher can send messages to each other user convenience is the key aspect of our application.

Although our application is well enough we have some limitations as well due to limited resources we had to use only free and open source technologies only our database and server both are free so they have limited capacity so it can't support large traffic properly in the worst-case scenario the server might crash. We have used a service for implementing live response but we are using the free version that limits are message capabilities and connection capabilities as well in case of large traffic the system will fail.

In our future development of this application, if we get enough resources, we will be able to diminish all these problems and may add new features to improve the system.

## **2.4 Scope of the Problem:**

Since DIU is a leading university with a good reputation in the field of technology usages. All the classrooms are equipped with modern teaching mediums and fully digitized security and management system, this aspect (counseling) of management should be digitized as well. With this current method of counseling hour management system, there are a few problems like this whole process is manual since it is both less efficient and has a high risk or errors like the faculty has no idea who is going to come for a meeting with what topics in mind so the faculties are almost always unprepared and since the process is fully manual it is pretty easy for our faculties to lost track of all these meetings and the record they keep may get messy and full of errors. The same goes for students they often forget to attend a meeting or appear in a rush-hour or wrong in an off day due to the lack of information about the counseling hour status of the faculties. The admin panel or the HR department is also at a disadvantage here as well since they can not get a good report either of the faculties or of the students. So, this manual system presents a lot of problems most important among them is time wastages and mass communication. Since the faculties can't utilize their time

properly to give the students full support and convenience the university administration is forced to recruit more faculty members which are costly and a bit unnecessary. After conducting a little research and discussing the fact with the faculties and students we have confirmed these facts to be true. So, we are developing this application to solve this problem and to make a new section of our varsity digitized. With the help of this application, all the students will be able to see all the counseling hours of all faculties as well as the general information about them that diminish the information gap that the old system has. Students can book an appointment and give a description of the meeting intent that helps the faculty to be prepared mentally for the meeting and since both parties get a list of all the meetings and alert reminders from the system that helps them to keep track of the meetings. More overall this is done through the application here comes the benefit of the administration as well since they can get all the records and can generate a report on this. So, this is beneficial for all the parties involved in this system.

To solve existing problems and adding new features to the existing system of managing counseling hours we are developing this application.

## **2.5 Challenges:**

One of the most important challenges among the array of challenges that we have faced during the development of the application is we are developing a web application but nowadays most of the students especially nontechnical background one's use mobile phones namely android OS equipped so they are used to android applications rather than a web application but to give multi-platform support it was necessary to develop the application as a web-based one. So, the popularization of the system is a bit hard and the fact of the availability of the internet remains a great challenge since our country doesn't have free internet or rather internet access available everywhere.

Then gathering requirements for developing this application since our teacher are very busy and it is hard to get an appointment and the other end-users our students were not able to give much good opinion or information about the idea.

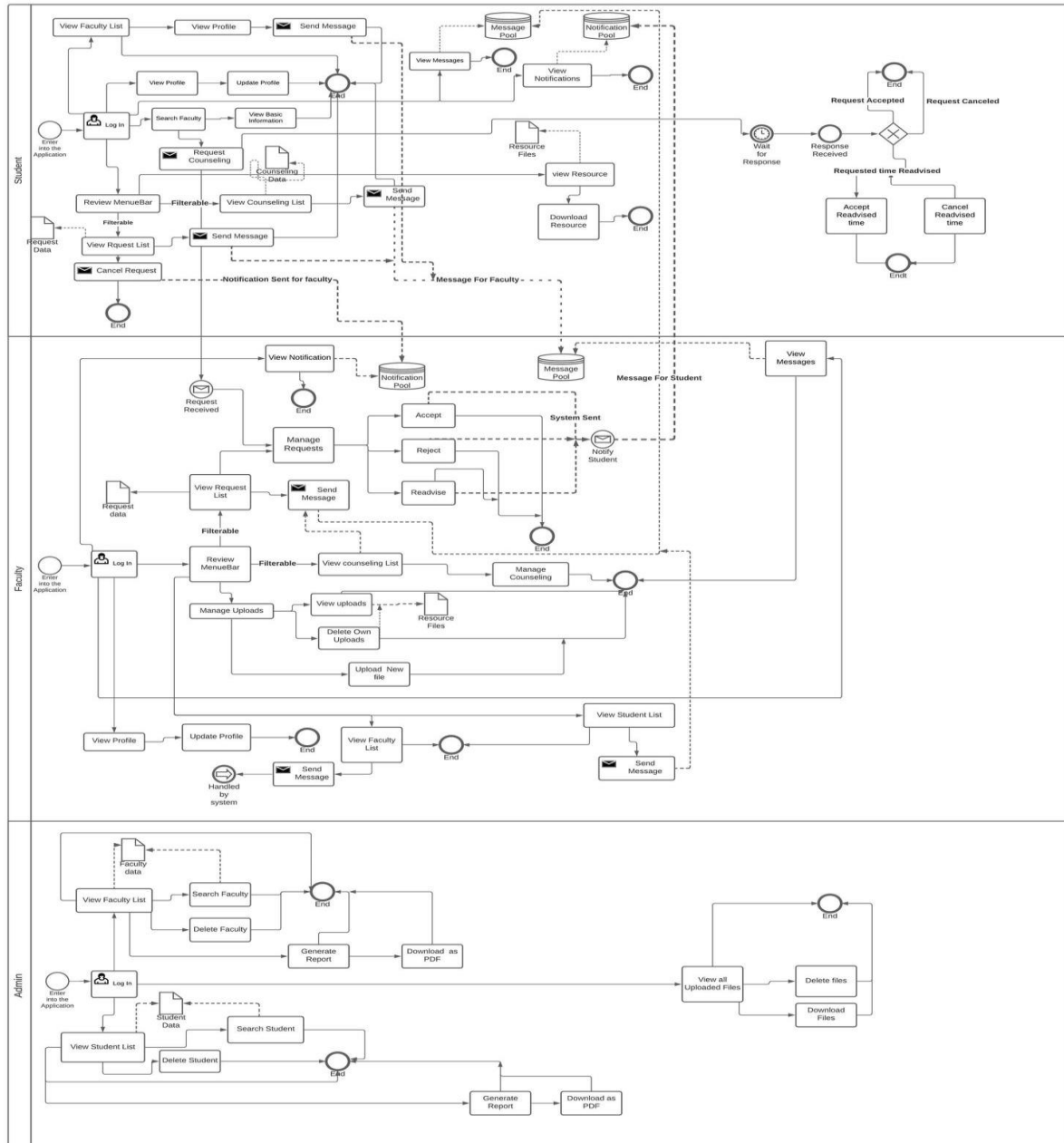
Then gathering data for test purposes was pretty tough since we tried to use real data about our teachers and students to see how this application performs in a real-life scenario and see the application in action and the development itself gave us some challenges as well since we had to learn new technologies to implement features for the application.

Finally, the maintenances and further upgrade will need some funding since we have limited resources and financial sufficiency.

## Chapter 3

### Requirement Specification

#### 3.1 Business Process Modeling



**Fig 3.1.1: Business Process Model**

### 3.2 Requirement Collection and Analysis:

In this project, there are three kinds of actors.

- i) Faculty
- ii) Student
- iii) Admin

We have gathered requirements from faculty by discussing detail about the objective, motive, and outcome of this project. And observe the student requirements closely from their side. And necessary admin requirements are fulfilled in this project by giving full control of monitoring and report generation methodology. The functional requirements for the user and its annalistic description are given below in the tabular form:

**Table 3.2.1: Requirement Collection and Analysis**

SL no.	ACTOR	Requirement Collection	Requirement Analysis
1	Faculty	Registration, Login & Profile-management	Faculty have to secure and ensure his information to use the application by a GUI. And the system has to auto-detect his identity that the user is a faculty. It also needs to be assured that the person bearing the university provided id.
2	Student	Registration, Login & Profile-management	The student has to secure and ensure his information to use the application by a GUI. And the system has to auto-detect his identity that

			the user is a student. It also needs to be assured that the person bearing the university provided id.
3	Admin	Login	Admin must have special access to the system so that he can completely monitor all the things and able to manage the system.
4	Faculty	Counseling Time Slot	Faculty need to add their counseling time slot so that students can get their information.
5	Student	Counseling Time Slot	Students need to get faculties necessary information and available counseling time slot. From the information, GUI students can easily connect faculty via chat.
6	Student	Apply for Counseling	Students can apply for counseling at a specific with a description. It will be grant as a counseling request. And also, the student can be able to quickly communicate with faculty via live chat.

7	Faculty	Response for Counseling	Faculty will Response to the counseling request from students and will be able to accept, reject, or readvise the student's expected time.
8	Student	Feedback to the response	The student will confirm the readvise time or cancel it.
9	Faculty & Admin	Upload files and resources	Faculty and admin can upload and share the necessary files and resources.
10	Admin	File and resources delete	Admin will have special access to delete any of the files and resources.
11	Faculty	File and resources delete	The faculty will have access to delete only the files he uploaded.
12	Student	Download	Students can download the necessary resources.
13	All	Chat	Every one of the systems inside the application can communicate with each other via live chat.
14	Admin	Report	Admin can generate a report for both teachers and



			students for all the counseling information.
15	System	Alert	The system will automatically alert all the users before the meeting started. And all interactions for users will be notified via notification and mail.

This is all the requirement analysis for the project from the users and all the management system needs.

### 3.3 Use Case Modeling and Description:

#### 3.3.1 Use Case Modeling:

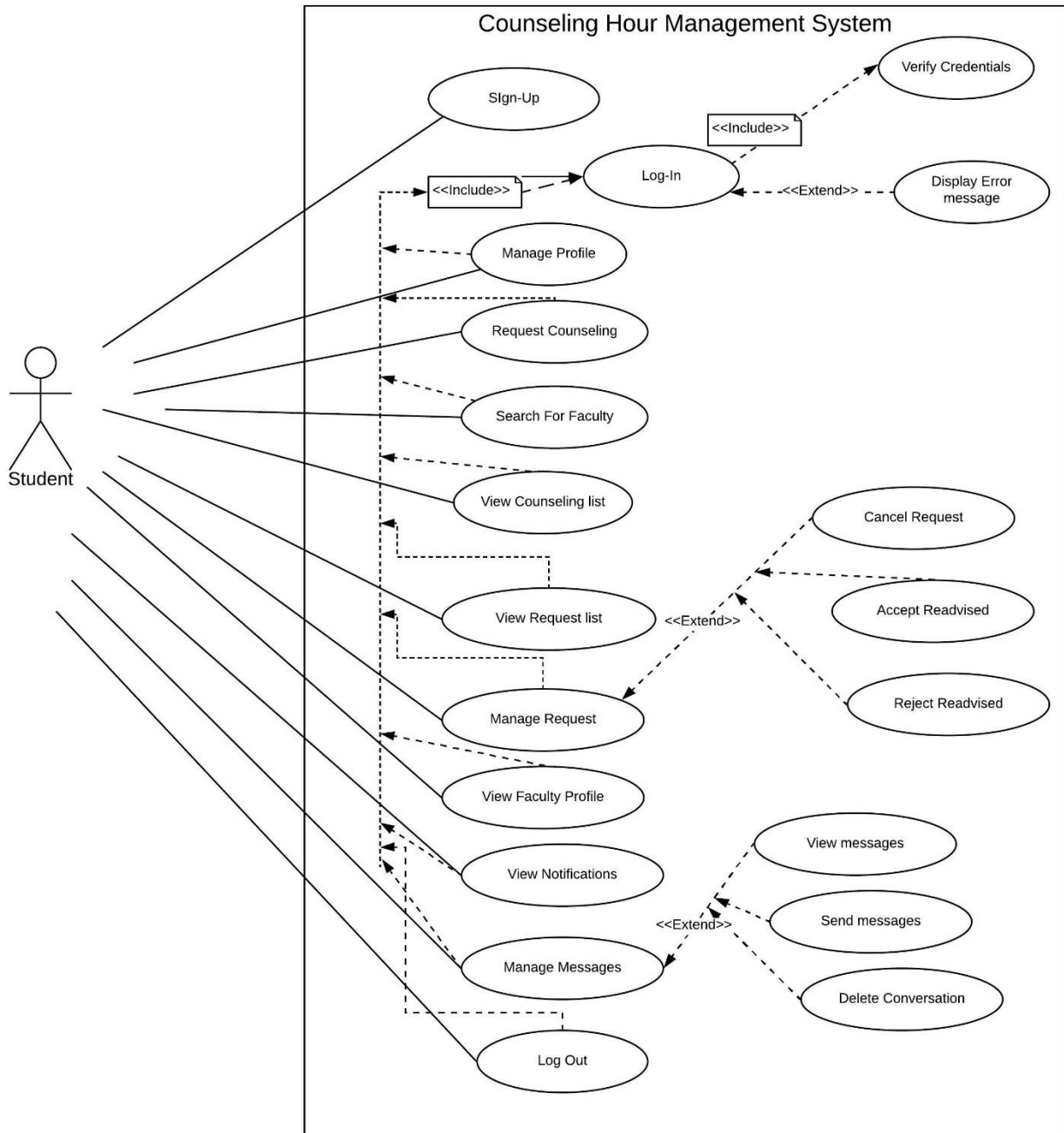


Fig 3.3.1: Student use Case Diagram

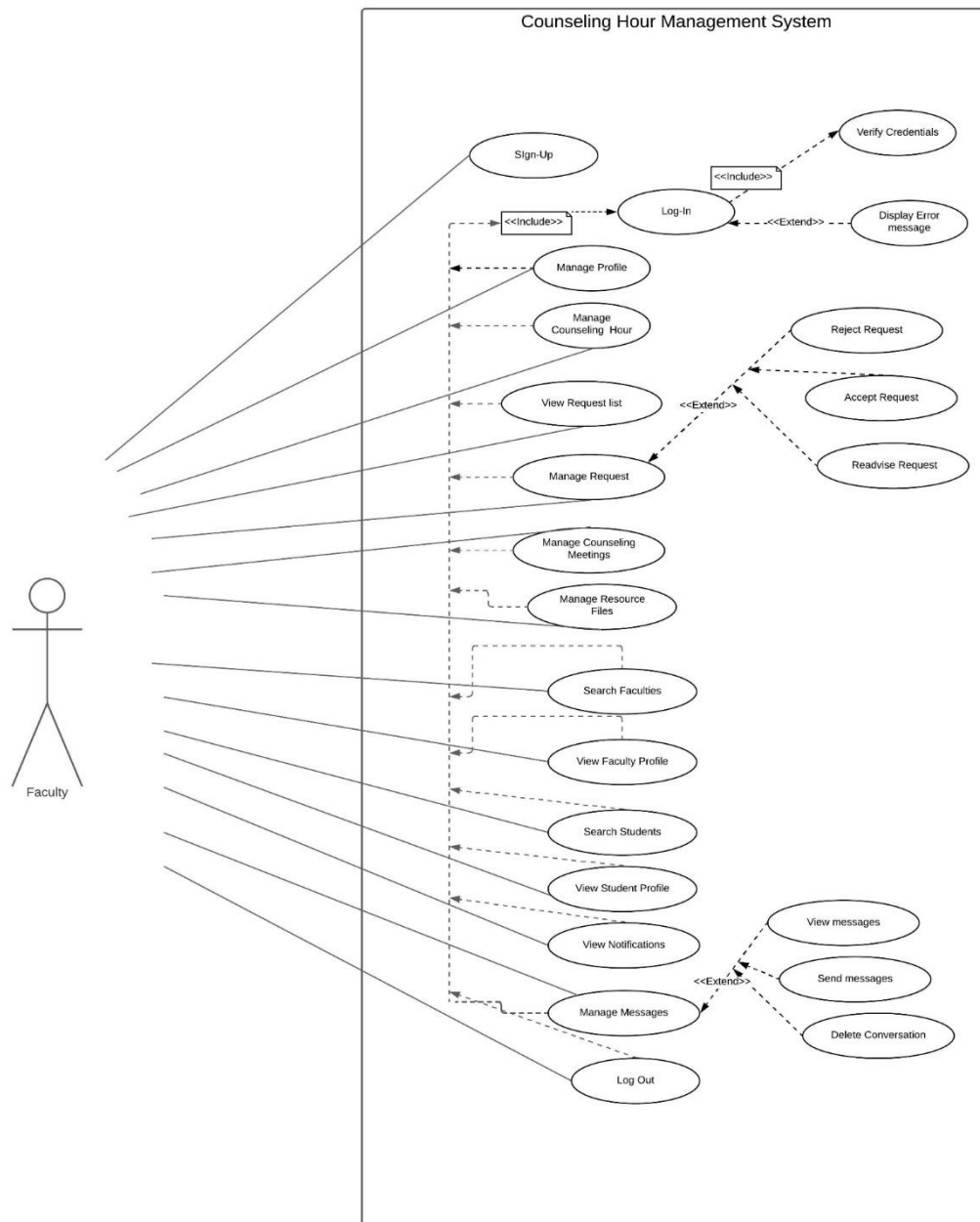


Fig 3.3.2: Faculty Use Case Diagram

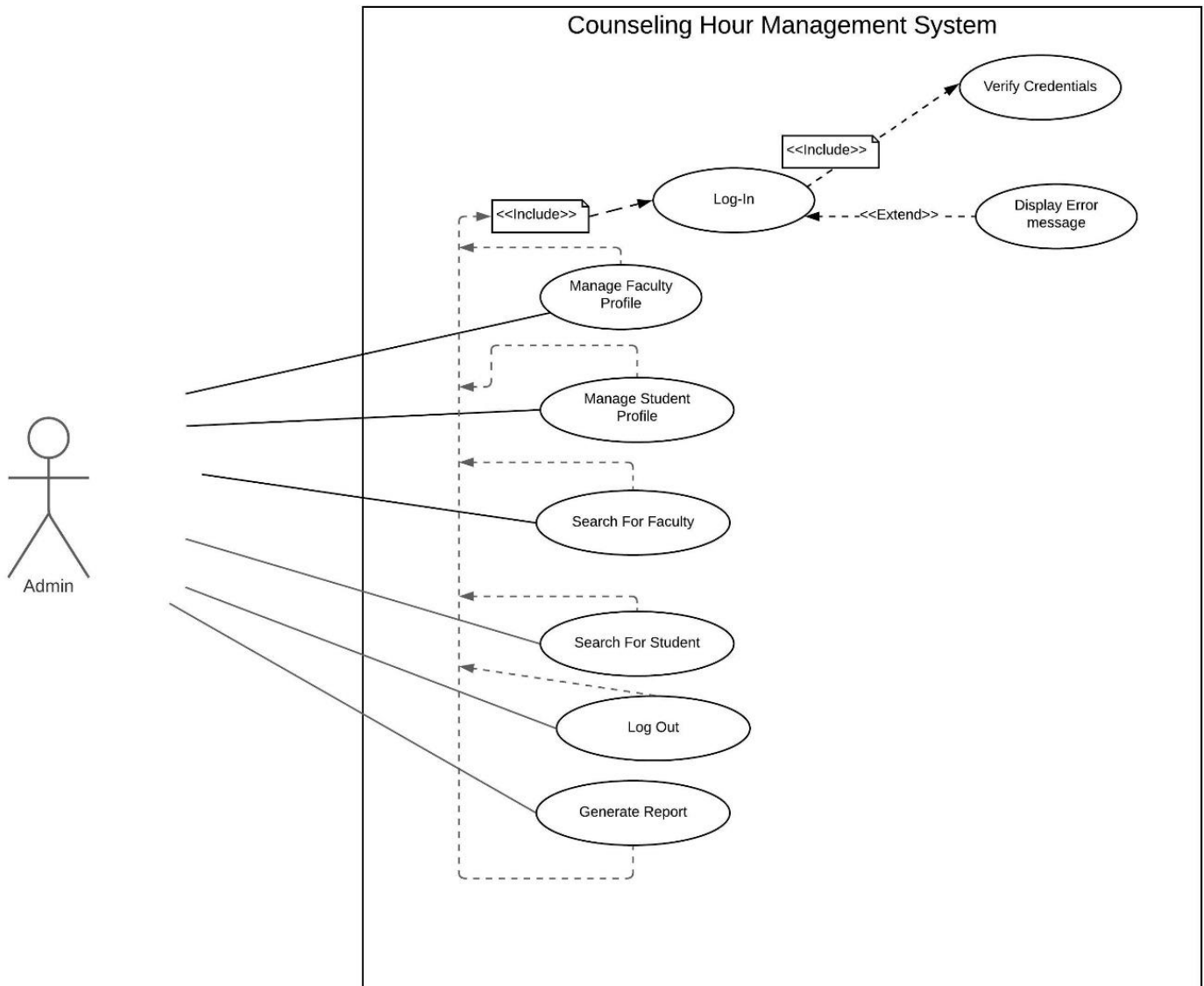


Fig3.3.3: Admin Use Case Diagram

### 3.3.2 Use Case Description:

**Table 3.3.2.1: Use Case description for login**

Use Case ID:	UC-1
Use Case Name:	Log in to the System
Created By:	Md. Tanvir Hasan Anik
Date Created:	10-9-2020
Actors:	Student, Admin, Faculty
Description:	Users will fill in the log in form using their credentials and log in successfully before start using the application.
Pre-condition:	<ol style="list-style-type: none"> <li>1. The user has an account</li> <li>2. The user is not already logged in</li> <li>3. The user is filling in the right form for log in</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. The user is logged in into the system</li> <li>2. User has access to all the designated functionalities of the system</li> </ol>
Include:	N/A
Normal Flow:	<ol style="list-style-type: none"> <li>1. User access the URL</li> <li>2. Fill up the login form with email and password</li> <li>3. The system authenticates the user</li> <li>4. The user gains access to all the functionalities</li> </ol>

**Table 3.3.2.2: Use Case description for Sign Up**

Use Case ID:	UC-2
Use Case Name:	Sign Up
Created By:	Md. Tanvir Hasan Anik
Date Created:	10-9-2020
Actors:	Student, Faculty
Description:	Users will fill in the Sign-up form with university-provided email and id, name, password, contact number, and an optional field picture then the system will send a verification mail in that email and after verifying the email the user account will be created and the user can log in and get access to all the functionalities
Pre-condition:	<ol style="list-style-type: none"> <li>1. User has a university-provided email address</li> <li>2. User has an id provided by the university</li> <li>3. User has access to the provided email inbox</li> <li>4. User filled up the right form and all the input fields with proper information</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. User successfully created an account user is logged in to the system</li> <li>2. User can log out and log-in into the system</li> <li>3. The user has access to all the designated functionalities.</li> </ol>
Include:	N/A
Normal Flow:	<ol style="list-style-type: none"> <li>1. User access the URL</li> </ol>

	<ol style="list-style-type: none"> <li>2. Fill up the sign-up form with email and password and other credentials</li> <li>3. The system sends a verification mail</li> <li>4. The user gains access to all the functionalities</li> </ol>
--	---

**Table 3.3.2.3: Use Case description for Log Out**

Use Case ID:	UC-3
Use Case Name:	Log Out from the System
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Student, Admin, Faculty
Description:	Users press the log out button and the session is destroyed and the user is logged out from the application and lost access to all the functionalities of the application
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. Has signed up</li> <li>3. Pressed the log out button</li> </ol>
Post-condition:	N/A
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. User access the URL</li> <li>2. Press the log out button</li> <li>3. The system kicks out the user from the system</li> </ol>

**Table 3.3.2.4: Use Case description for Manage Profile**

Use Case ID:	UC-4
Use Case Name:	Manage Profile
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Student, Faculty
Description:	Users press the My profile button and the user will be redirected to a new page for viewing the user's profile where the user can see all his information that he had provided in the system and user can both edit and update information in their profile
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. The user pressed the My Profile option</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. The user has access to all the profile information.</li> <li>2. Users will be able to add, edit, or update information.</li> </ol>
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. User select my profile option and immediately redirected to a new page that shows all the information about the user that he/has already added into the system</li> <li>2. User view own information and can add or edit own information</li> </ol>



**Table 3.3.2.5: Use Case description for Request Counseling**

Use Case ID:	UC-5
Use Case Name:	Request Counseling
Created By:	Emon Kumar Das
Date Created:	10-9-2020
Actors:	Student
Description:	The user (student) will search for a faculty he/she wants to get an appointment from then the user will send a request for a counseling meeting by selecting a time and giving a short description. The time can be both from the given time slots or time of liking.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. Search and select a faculty for sending the request</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. User can now wait for a response</li> <li>2. If the time is a re-advised one student can cancel or accept the new time slot at will</li> </ol>
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. User search for the faculty and select the one he/she wants</li> <li>2. Counseling time is selected and then fill the description input field and submit</li> </ol>

**Table 3.3.2.6: Use Case Description Search for Faculty**

Use Case ID:	UC-6
Use Case Name:	Search for Faculty
Created By:	Emon Kumar Das
Date Created:	10-9-2020
Actors:	Student, Faculty, Admin
Description:	The user (student) will search for a faculty he/she wants to get an appointment from then the user will select from the suggested names. Faculty and admin can view information about the faculty and send a message as well.
Pre-condition:	1. Logged in into the system.
Post-condition:	1. User can view the faculty profile and details 2. The user (student) can send a request or cancel a request for counseling. 3. User can send message
Include:	Log in
Normal Flow:	1. User search for the faculty by name and select the one he/she wants 2. After selection, the faculty information and time slots for counseling are presented in front of the student to see.

**Table 3.3.2.7: Use Case description for View Counseling List**

Use Case ID:	UC-7
Use Case Name:	View Counseling List
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Student, Faculty
Description:	User press the counseling option from the navigation bar then a list appears with filter options on this context and then selecting any option among them shows all the counseling meeting list with related information based on the filtering criteria user (student, faculty) can also send a message from here to the opposition party.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. Request from the requests list is accepted by both parties.</li> </ol>
Post-condition:	N/A
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user selects the counseling the option from the navigation bar and can filter based on needs</li> <li>2. View the list can refresh also can view the details of the student/faculty and all the information about the meeting.</li> </ol>

	3. Can also send a message with the other end-user as well.
--	---

**Table 3.3.2.8: Use Case description for View Request List**

Use Case ID:	UC-8
Use Case Name:	View Request List
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Student, Faculty
Description:	User (student, faculty) select the Requests option from the navigation bar then a sub-menu appears from where the user can select to apply a filter in the results a list with all the information like the status, time, etc., about the request and the other end-user is shown as well one can send message to other if they want.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. The user has already made some requests(student) or there are some pending requests (faculty) in the list to show.</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. User (student)can now wait for a response</li> <li>2. Faculty and user both can see the information about the requests and get access to send a message.</li> </ol>
Include:	Log in

Normal Flow:	1. The user Selects the option requests and gets access to all the necessary information about the request in a list form.
--------------	--

**Table 3.3.2.9: Use Case description for Manage Requests**

Use Case ID:	UC-9
Use Case Name:	Manage Requests
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Student, Faculty
Description:	User (student, faculty) select the Requests option from the navigation bar then a sub-menu appears from there the user can select to apply a filter in the results a list with all the information like the status, time, etc., about the request and the other end-user is shown as well one can send message to other if they want. Here both faculty and student have different action options like a student can cancel a request or accept or reject a re-advised time on the other hand a faculty can accept the pending request or reject or re-advise a new time.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. The user has already made some requests(student) or there are some pending requests(faculty) in the list to show.</li> </ol>

Post-condition:	<ol style="list-style-type: none"> <li>1. User (student) can now wait for a response</li> <li>2. Faculty and user both can perform some actions on the request like accept, cancel, and many more.</li> </ol>
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user selects the option requests and gets access to all the necessary information about the request in a list form.</li> <li>2. Both can perform their respective actions on every request in the request list</li> </ol>

**Table 3.3.2.10: Use Case description for View Notifications**

Use Case ID:	UC-10
Use Case Name:	View Notifications
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Student, Faculty
Description:	User (student, faculty) select the bell icon and get access to the list of all notifications sent by the system on every event system to send a notification containing a brief description to keep the users updated. Users can mark the notifications as read so the already seen ones don't show up again.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> </ol>

	2. The user has already made (student) some action or some action is waiting to be responded(faculty).
Post-condition:	1. User mark the notifications as read 2. View notification list
Include:	Log in
Normal Flow:	1. The user Selects the bell icon and a list appears as a drop-down containing brief information about some event or action. 2. User can scroll and view the notifications and mark them as read so they are shifted from unread to read the list of notifications

**Table 3.3.2.11: Use Case description for Manage Messages**

Use Case ID:	UC-11
Use Case Name:	Manage Messages
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Student, Faculty
Description:	User (student, faculty) select the envelope icon then the users will be redirected to a new page where he/she can see all the messages that have been sent by the user or gotten from others. Users can enter into the conversation and the message will be marked as seen user can also delete the conversation as well. User can even search for new contact among other users of the

	system and can send a message to start a new conversation with them even users can send messages to their self.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. User has already made has some message in the chat list already</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. User mark the messages as reading</li> <li>2. User has the access to all his/her conversations and he/she can delete a conversation or the whole chat list</li> <li>3. Can search and send a new message to others or even message to own self</li> </ol>
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user selects the envelope icon then get access to all the messages</li> <li>2. User can search if want to send a message if that not already exists in the conversation list.</li> <li>3. User can delete or start a new conversation with other users or own self</li> </ol>



**Table 3.3.2.12: Use Case description for Manage Counseling Hours**

Use Case ID:	UC-12
Use Case Name:	Manage Counseling Hour
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Faculty
Description:	User (faculty) after log in will be redirected to a new page where he/she can view the list of counseling hour slots the user has already published user can delete a slot or add a new one, to do so user have to select a valid day and time interval and then register the slot
Pre-condition:	1. Logged in into the system
Post-condition:	1. The user has the access to delete a counseling hour slot or to add a new one.
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. Users log in to the system and get access to a page where users can see all the list of counseling hour slots previously enlisted by him/her.</li> <li>2. Users can delete a slot.</li> <li>3. Users can add a new one by putting a time interval a day.</li> </ol>

**Table 3.3.2.13: Use Case description for Manage Counseling Meetings**

Use Case ID:	UC-13
Use Case Name:	Manage Counseling Meetings
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Faculty
Description:	Users press the counseling option from the navigation bar then a list appears with filter options on this context and then selecting any option among them shows all the counseling meeting lists with related information based on the filtering criteria user can also send messages from here to the opposition party. Users can mark the meeting status as arrived and not-arrived based on the student's arrival.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. Request from the requests list is accepted by both parties.</li> </ol>
Post-condition:	N/A
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user selects the counseling the option from the navigation bar and can filter based on needs</li> <li>2. View the list can refresh also can view the details of the student/faculty and all the information about the meeting.</li> </ol>

	3. Can also send a message with the other end-user as well.
--	---

**Table 3.3.2.14: Use Case description for Manage Resource Files**

Use Case ID:	UC-14
Use Case Name:	Manage Resource Files
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Faculty, Admin
Description:	The user selects the uploads or resource files option from the navigation bar from there they can filter to see all the existing files that have already been uploaded in the system or can add a new file. Admin has the authority to delete any one's file but faculty can only delete his/her files
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system</li> <li>2. Some files have already been uploaded into the system</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. User-based on the authority has the option to download a file</li> </ol>
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. The user selects the uploads option and can upload a new file or see their uploaded files or all the uploaded files</li> </ol>

	<ol style="list-style-type: none"> <li>2. User(faculty) can delete his/her own files admin can delete any files</li> <li>3. User can download the files</li> </ol>
--	--

**Table 3.3.2.15: Use Case description Search for Student**

Use Case ID:	UC-15
Use Case Name:	Search for Student
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Faculty, Admin
Description:	Users will search for a student based on his/her name or id or email to see a matched list of students from there user can see all the information of the student and can send a message is they want to.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system.</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. Users can view the student profile and details.</li> <li>2. User can send message</li> </ol>
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. User search for the student based on his/her name or id or email and select the one he/she wants</li> <li>2. After selection, the student's information is displayed to the user and can also send a message.</li> </ol>

**Table 3.3.2.16: Use Case description View Student Profile**

Use Case ID:	UC-16
Use Case Name:	View Student Profile
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Faculty, Admin
Description:	Users will search for a student based on his/her name or id or email to see a matched list of students from there user can view the profile and see all the information of the student and can send a message is they want to.
Pre-condition:	1. Logged in into the system.
Post-condition:	1. Users can view the student profile and details. 2. User can send message
Include:	Log in
Normal Flow:	1. User search for the student based on his/her name or id or email and select the one he/she wants 2. After selection, the student's information is displayed to the user and can also send a message.

**Table 3.3.2.17: Use Case description View Faculty Profile**

Use Case ID:	UC-17
Use Case Name:	View Faculty Profile
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Faculty, Admin, student
Description:	Users will search for a Faculty based on his/her name or id or email to see a matched list of students from there user can view the profile and see all the information of the faculty and can send a message is they want to.
Pre-condition:	1. Logged in into the system.
Post-condition:	1. Users can view the faculty profile and details. 2. User can send message
Include:	Log in
Normal Flow:	1. User search for the faculty based on his/her name or id or email and select the one he/she wants

	2. After selection, the faculty's information is displayed to the user and can also send a message.
--	---

**Table 3.3.2.18: Use Case Description Manage Faculty Profile**

Use Case ID:	UC-18
Use Case Name:	Manage Faculty Profile
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Admin
Description:	The user will search for a Faculty based on his/her name or id or email to see a matched list of students from there user can view the profile and see all the information of the faculty and can delete the profile from including the user account from the system.
Pre-condition:	1. Logged in into the system.
Post-condition:	1. Users can view the faculty profile and details. 2. User can delete the faculty from the system
Include:	Log in

Normal Flow:	<ol style="list-style-type: none"> <li>1. User search for the faculty based on his/her name or id or email and select the one he/she wants</li> <li>2. After selection, the faculty's information is displayed to the user and can delete the faculty from the system database</li> </ol>
--------------	---

**Table 3.3.2.19: Use Case Description Manage Student Profile**

Use Case ID:	UC-19
Use Case Name:	Manage student Profile
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Admin
Description:	The user will search for a student based on his/her name or id or email to see a matched list of students from there user can view the profile and see all the information of the student and can delete the profile from including the user account from the system.
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system.</li> </ol>
Post-condition:	<ol style="list-style-type: none"> <li>1. Users can view the student profile and details.</li> <li>2. User can delete the student from the system</li> </ol>



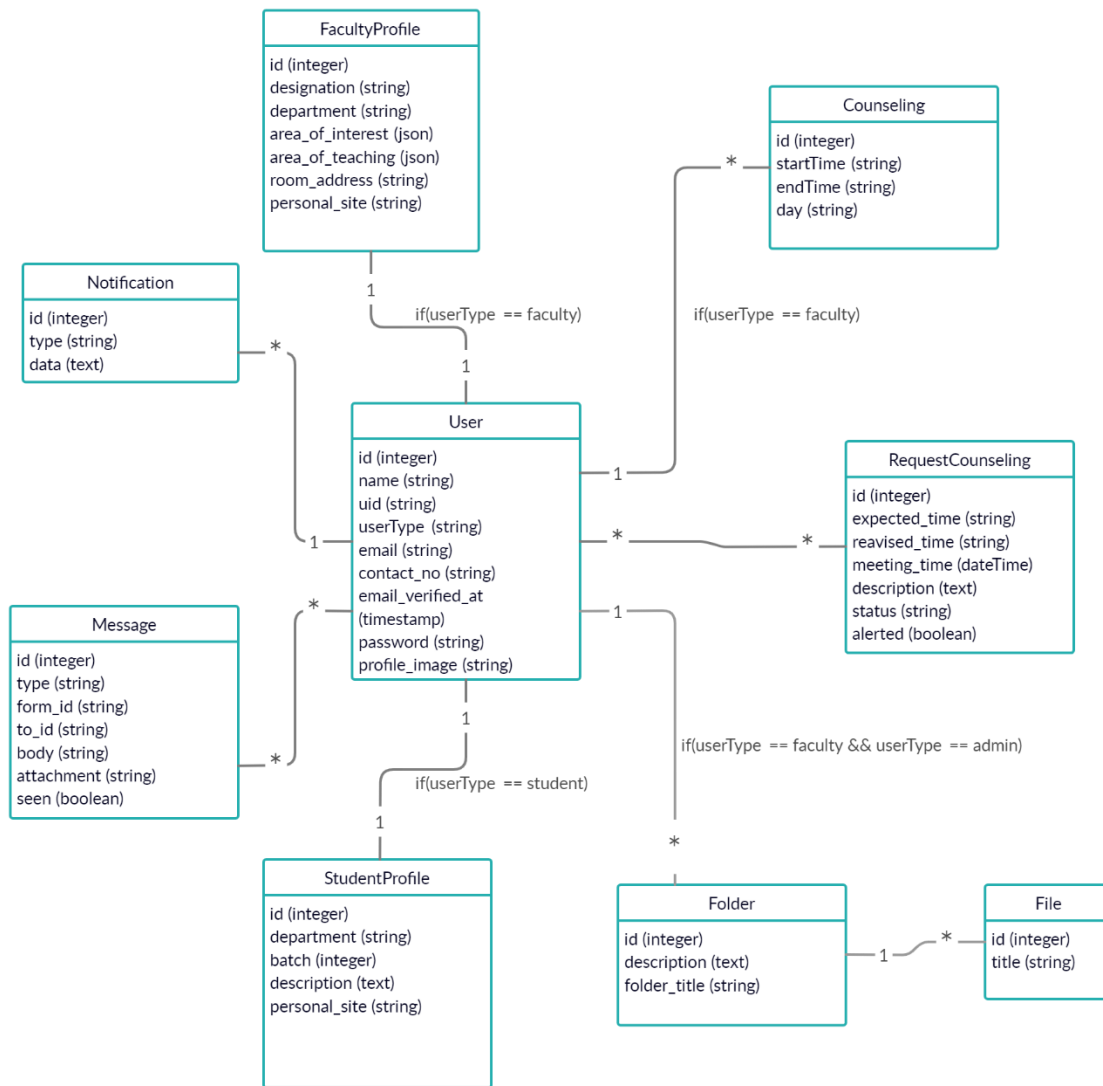
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"> <li>1. User search for the student based on his/her name or id or email and select the one he/she wants</li> <li>2. After selection, the student's information is displayed to the user and can delete the faculty from the system database</li> </ol>

**Table 3.3.2.20: Use Case Description Generate Repot**

Use Case ID:	UC-20
Use Case Name:	Generate Report
Created By:	Tahzid Mahmud Suvo
Date Created:	10-9-2020
Actors:	Admin
Description:	User will search for a Student or faculty based on his/her name or id or email see a matched list of students or faculties from there user can view the profile and see all the information of the student or faculty and can generate a report of his/her activity in a monthly basis
Pre-condition:	<ol style="list-style-type: none"> <li>1. Logged in into the system.</li> </ol>

Post-condition:	<ol style="list-style-type: none"><li>1. User can see a report in tabular format</li><li>2. User can download the report in a pdf format if wants</li></ol>
Include:	Log in
Normal Flow:	<ol style="list-style-type: none"><li>1. User search for the student or faculty based on his/her name or id or email and select the one he/she wants</li><li>2. After selection of a month, the student's or faculty's information about the activity of that month in a tabular format</li><li>3. User download the report in a pdf format as well</li></ol>

### 3.4 Logical Data Model:



**Fig 3.4: Logical Data Model**

### 3.5 Design Requirements:

From the discussion so far, we can see there are a lot of functionalities in our application and some of them are must-have according to the requirement analysis and they are the following:

The most important function of our system is the ability to give the students information about the counseling hour and based on that information the ability to request an appointment at a designated time or request a time other than the designated slots at a time of their liking student will also have the ability to cancel the request in the other end teachers will have the ability to accept, reject or even re-advise the time of request and student can accept or reject that if both parties are agreed then the meeting is registered in the list. Besides this, the list of all requests and appointments that have been conducted and are going to be conducted in the future all are listed and both teachers and students will be able to review at any time.

Other than that, all the actions will be synchronized through notifications and emails for both end-users, live interaction is our key strength for this application there will be a live chat feature that enables the students and teachers to interact with each other at any time live. For the administrator there will a supreme feature to erase the whole data about any user may that be a teacher or a student along with their user profile and for monitoring and for checking the progress purposes there will be a feature to generate a report and it will be filterable based on the month.

Our application is fully customizable since all the aspects of this application will be dynamic to allow future customization and implement new features in the future on a broader scale since this a kinda prototype this can't serve too many users due to the current lack of resources. But this application will be easy to scale because for the sake of scalability we have used all the popular web technologies like PHP, JavaScript, and MySQL that are being used in the majority of the scalable web applications.

## CHAPTER 4

### DESIGN SPECIFICATION

#### 4.1 Front-End Design:

The main aim of our project is to manage counseling hour properly that's why when a user enters our application, first of all, they see the landing of the counseling portal (figure 4.1) where they can sign up into our application or if they are already registered then they can log in just by clicking on the login button. As our application is used both by students, teachers as well as admins that's why for using this application they must have to log in to our application.

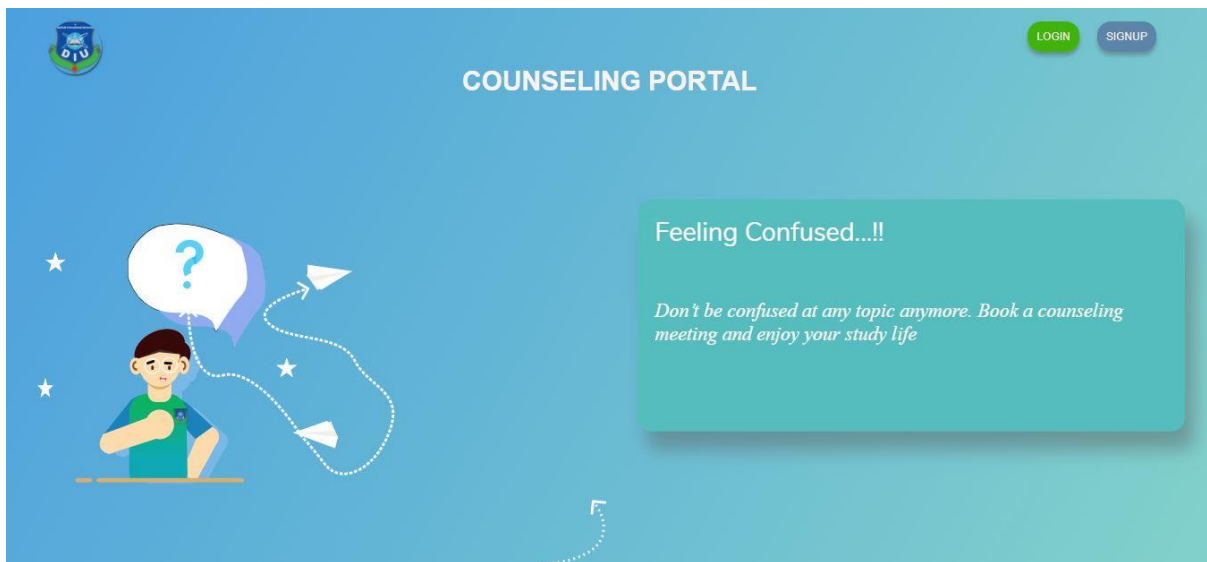


Figure 4.1: Landing page of The Counseling Portal

Figure 4.2 shows our registration page where they can see all the required fields for registering themselves such as name, id, email-id, contact number, password, profile image (optional). So, they can easily register themselves by putting in all the necessary information.

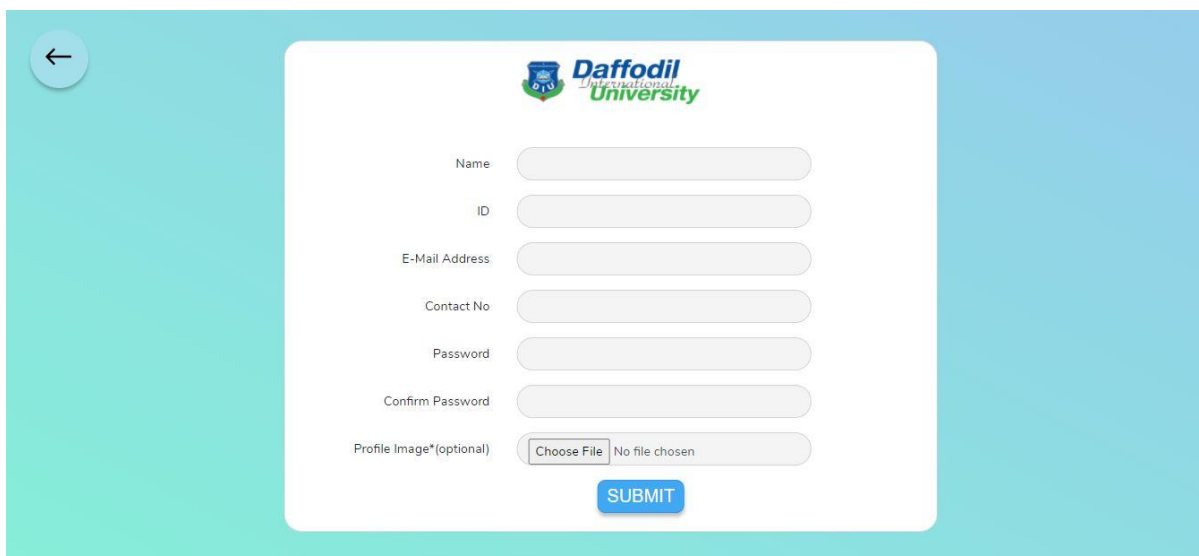
The image shows a registration form for Daffodil International University. The form is centered on a light blue background. At the top left of the form area is a circular back arrow button. The form itself is a white rounded rectangle containing the university's logo at the top center. Below the logo are several input fields: Name, ID, E-Mail Address, Contact No, Password, and Confirm Password. Each field has a light gray placeholder text. At the bottom of the form is a 'Profile Image\*(optional)' section with a 'Choose File' button and the text 'No file chosen'. A blue 'SUBMIT' button is located at the bottom right of the form.

Figure 4.2: Registration page

After registering themselves, they can go back to the landing page of the counseling portal and visit our login page (figure 4.3) where they see all the required fields for login into our application. They can easily log in to our application by giving their login information such as registered email addresses and passwords. If anytime they forgot their registered password then they can easily reset their password from our reset password page.

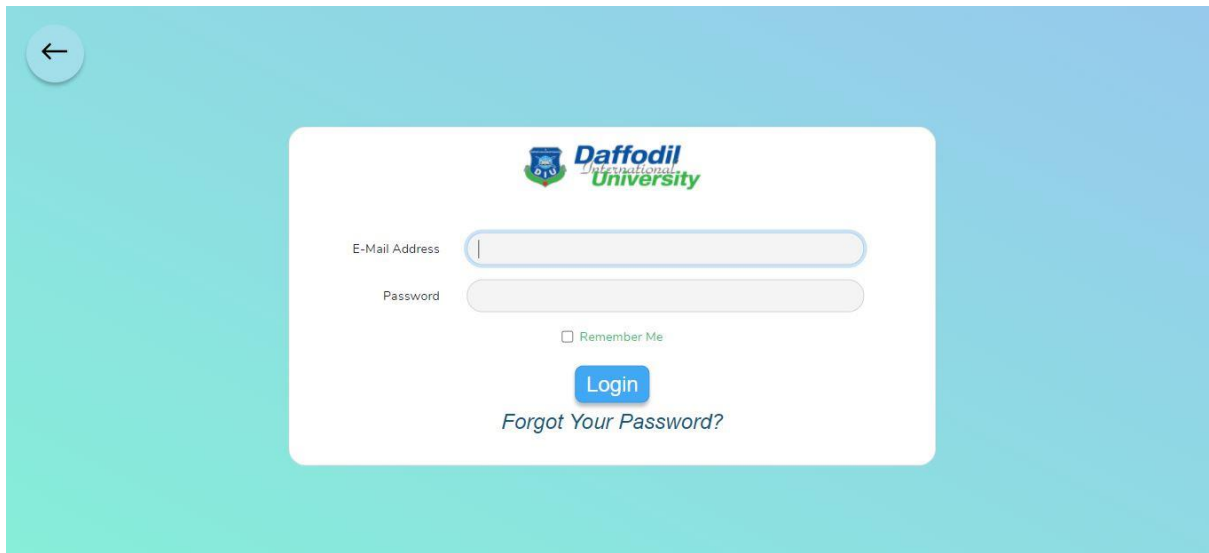


Figure 4.3: Login Page

After login in our website, they will come to their respective student home page (figure 4.4). Figure 4.4 is the homepage for students means if the user registered themselves as a student then after login, they can see this page where they see different menu bars such as home, resources, requests, counseling. They can also see the option for the apply counseling where they can find their respective teacher just by entering their name in the search faculty name here field.

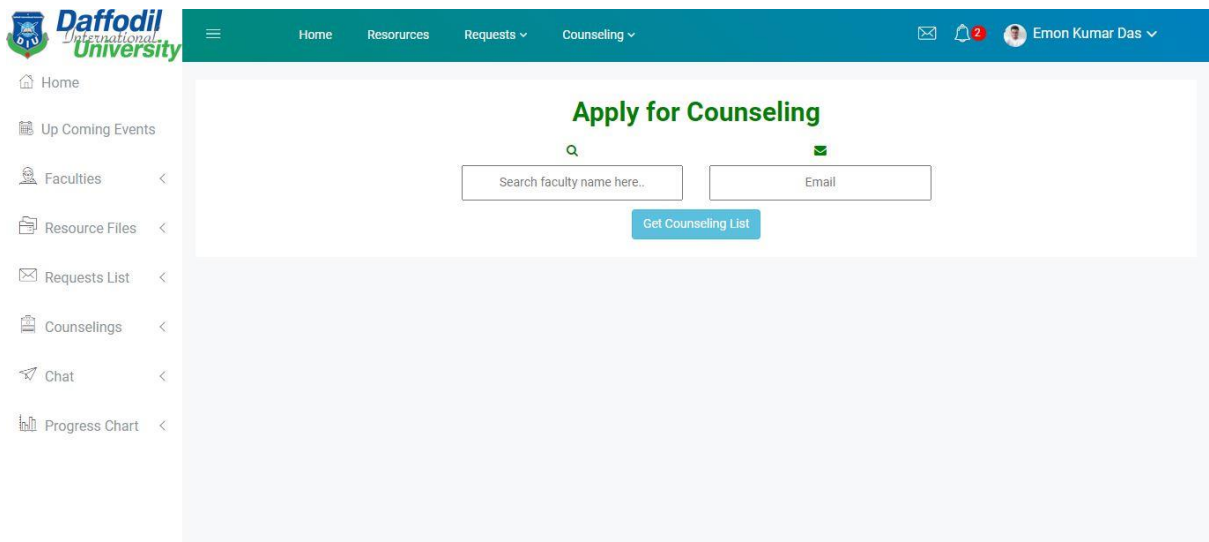


Figure 4.4: Home Page for Students

In our get counseling page (figure 4.5), they can see the counseling time slots of the respective teachers, as well as they, will also see the counseling request form. According to the respected time slots, students can request counseling time from the counseling request form. They can also get a short description form where they can give a short description of their problem which is less than 500 words.

Day	Start Time	End Time
Saturday	10:00 AM	11:00 AM
Sunday	12:00 AM	1:00 AM

Figure 4.5: Get Counseling Page (Request Counseling)

They can also see all the necessary information about their respective teacher such as name, email-id, designation, phone, department, room address, area of interest from the basic information tab (figure 4.6)



**Daffodil International University**

Home Resources Requests Counseling

Home Request Counseling **Basic Information**

Up Coming Events

Faculties <

Resource Files <

Requests List <

Counselings <

Chat <

Progress Chart <

**Name**  
Dr. S. M. Aminul Haque

**Designation**  
Associate Professor

**Email ID**  
aminul.cse@daffodilvarsity.edu.bd

**Phone**  
01847140129

**Department**

**Room Address**

Figure 4.6: Get Counseling Page (Basic Information)

Here our request list page (figure 4.7) shows them all the counseling request details such as when they are requested for counseling, their respected teacher's name, expected time, readvised time, counseling status, details & contact, and action. From here a student can easily understand that their counseling request has been accepted, readvised, rejected, or pending. They can also see them separately from the request menu bar.

**Daffodil International University**

Home Resources Requests Counseling

Home Request List Refresh List

Requested at	Name	Expected Time	Readvised Time	Status	Details & Contact	Action
29/09/20 05:34 AM	Dr. S. M. Aminul Haque	29/09/2020 11:34 AM		Pending		
29/09/20 05:32 AM	Dr. S. M. Aminul Haque	29/09/2020 2:15 PM		Rejected		
29/09/20 05:30 AM	Dr. S. M. Aminul Haque	30/09/2020 10:30 AM	30/09/2020 12:00 PM	Readvised		✓
29/09/20 05:27 AM	Dr. S. M. Aminul Haque	29/09/2020 12:20 PM		Accepted		

Up Coming Events

Faculties <

Resource Files <

Requests List <

Counselings <

Chat <

Progress Chart <

Figure 4.7: Requests List Page

In the same way, they can also see all the meetings from the all meeting page (figure 4.8). Here they can see all their accepted meeting with their respected teachers as well as meeting

time and details & contact. They can also see new upcoming meetings, arrived meetings, and not-arrived meetings from the counseling dropdown menu.

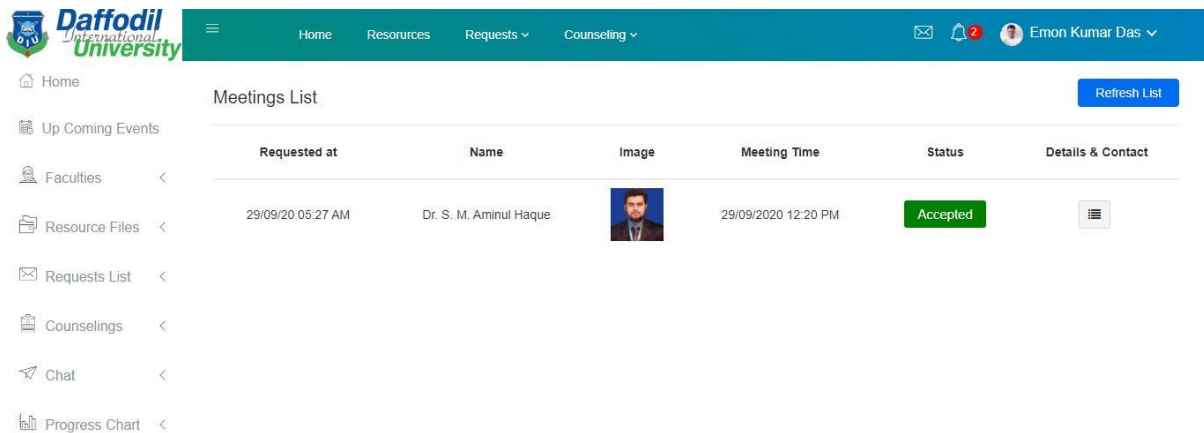


Figure 4.8: All Meetings Page.

Students also get notifications and mail into their university email account about their counseling request status. If their respected teacher accepted their counseling request then will get a notification that their counseling request is accepted and in the same way, if it is rejected, readvised then also they will get a notification and an email as we see in figure 4.9.

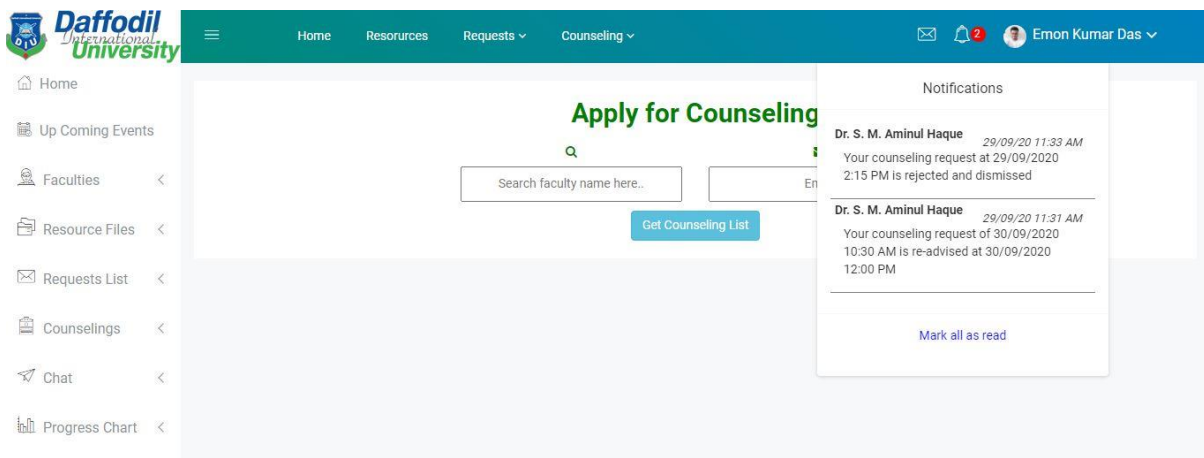


Figure 4.9: Notifications drop-down list

Students can also get the opportunity to chat with their respective teachers from the live chat the chatting interface look like this (figure 4.10). Let's say they face a problem and they need to get an instant solution then can easily solve their issue through chatting with their respected teacher or they can also chat with their respected teacher about their counseling problem.

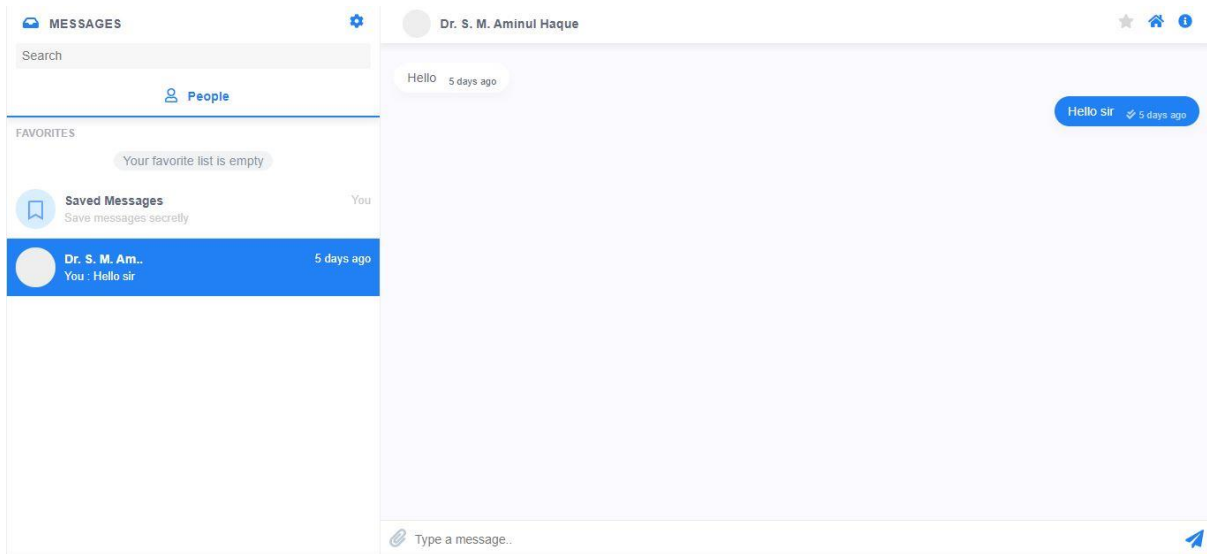


Figure 4.10: Chat Page

In the same way if our respected teacher login our application they can see our faculty home page (figure 4.11) where they can see add counseling hour from where they can add counseling hour. They can set dates, counseling start time, and end time. They can see all the counseling list from this page,

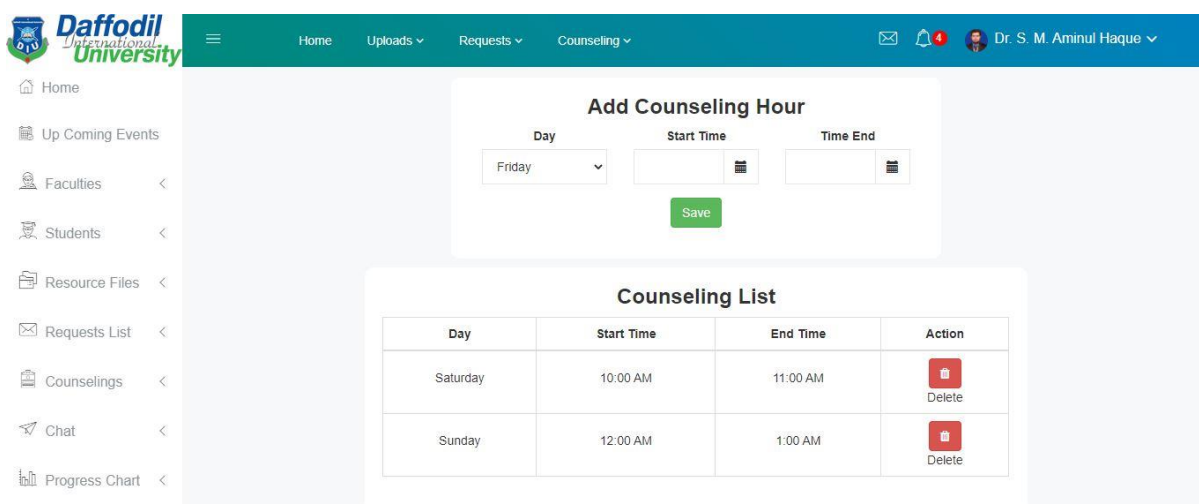


Figure 4.11: Faculty Home page

Teachers can also see all the request from the request list page (figure 4.12). They can see all the information such as request time, the name of the student, student's id, expected time, readvised time, counseling status, details & contact, and action. They can get all the options such as accept option, reject option, readvised option in that form. They can also see all the counseling status differently from the request menu bar.

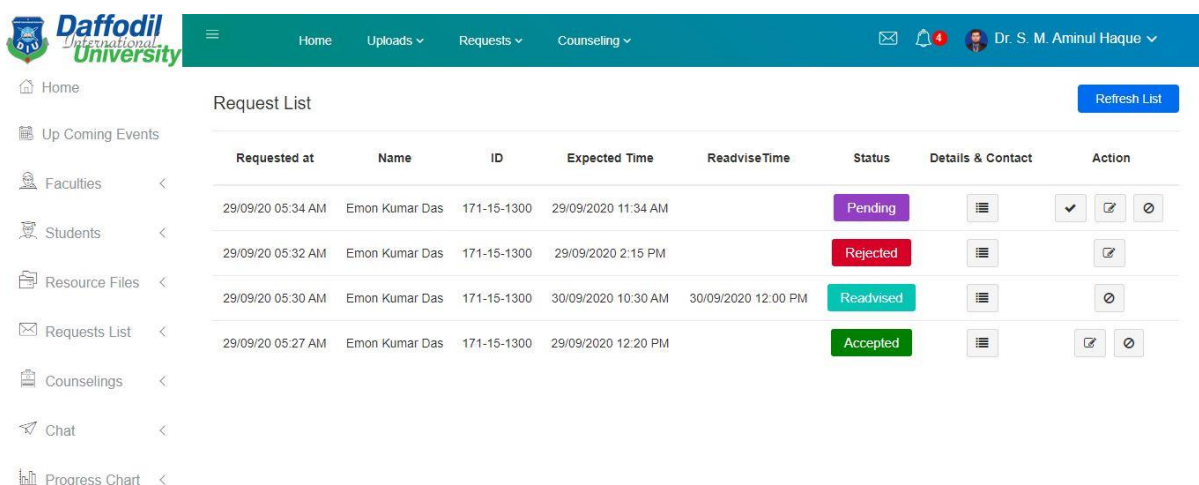


Figure 4.12: Request List Page

In the same way, teachers can also see all their meetings with their beloved students from the all meetings page (figure 4.13). From this page, they can see all the detailed information about the student whom they must meet. They can also see upcoming meetings, on arrival meetings, arrived, and not-arrived meeting from the counseling menu bar. They will also get all the counseling request notification from the notification icon and they can also see the option to mark them as all read.

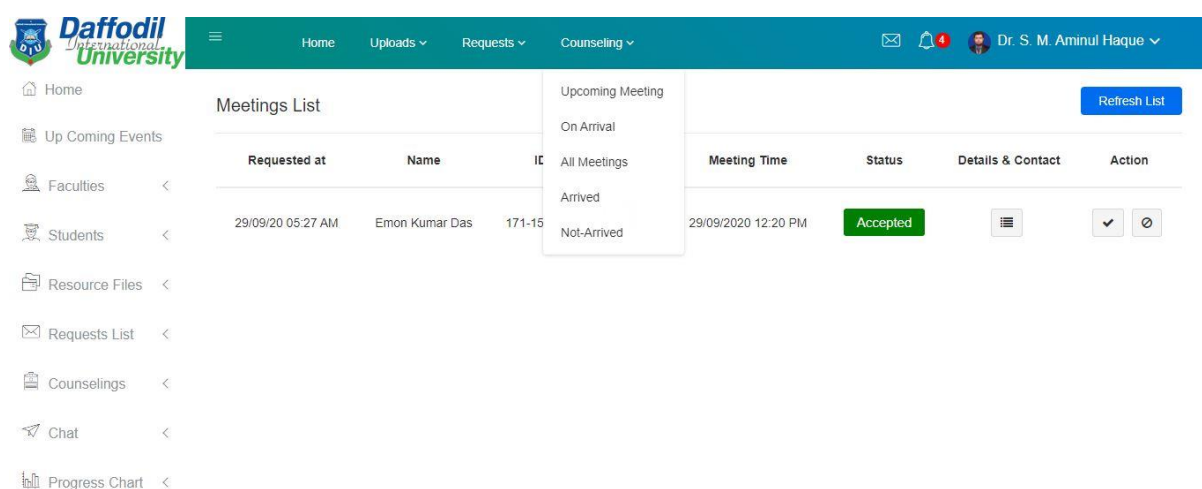


Figure 4.13: All Meeting Page

If our respective teachers want to talk any with their students then they can also get that option from our chat page (figure 4.14). Let's say a student is requested for counseling and a teacher accepts his/her request but at that time an emergency meeting arrives. So, in that situation, the teacher can easily message the students and solve the issue.

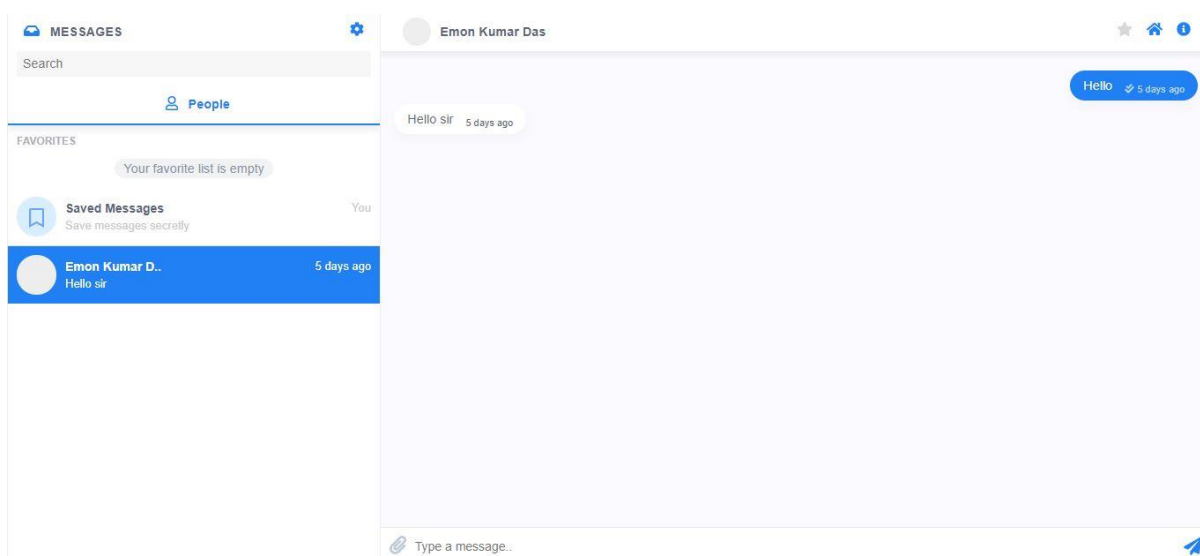


Figure 4.14: Chat Page

Teachers also get the option to upload their necessary files from the upload page by a title and a short description (Figure 4.15.1).

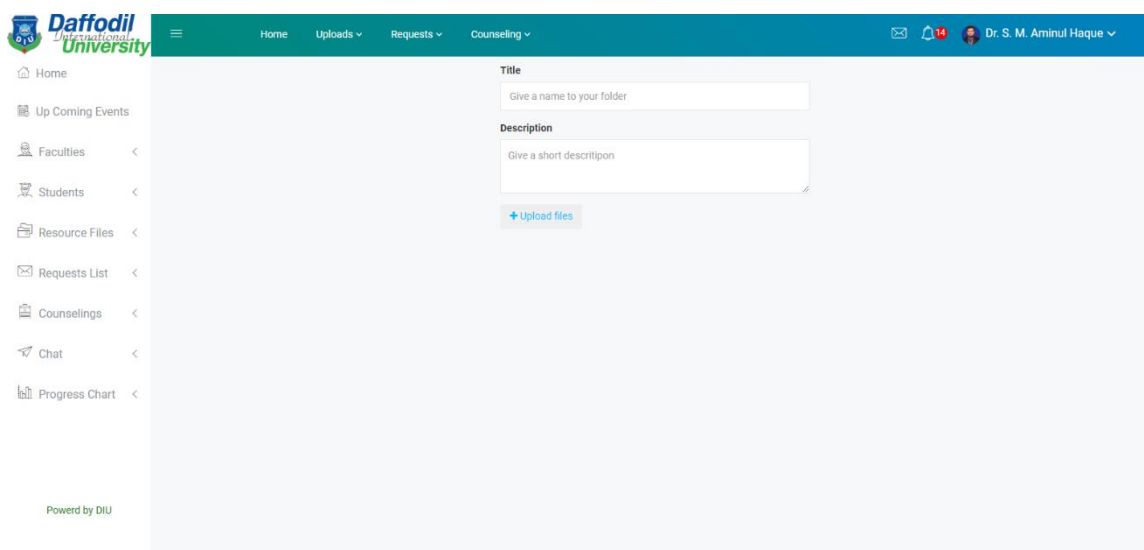


Figure 4.15.1: New Upload Folder Page

They can upload single or multiple files from their desktop just by doing drag and dropping. They can also see all uploads as well as their upload from the upload menu bar.

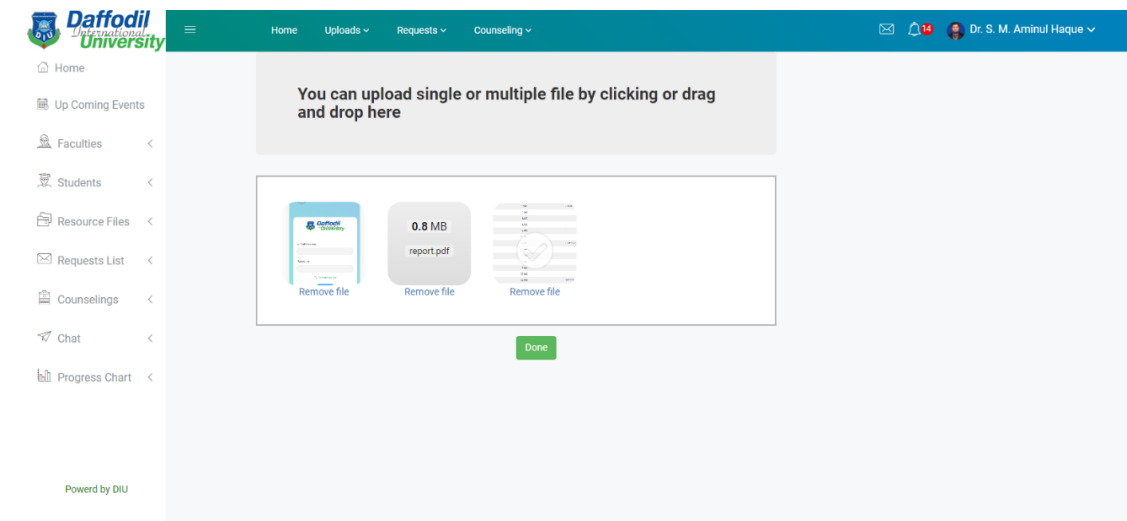


Figure 4.15.2: Upload Files by drag and drop

If the faculty wants to delete the folder they can do it from my uploads menu. Students will not have any delete options. But admin can delete all the files and faculties can delete only the file they uploaded (figure 4.15.3).

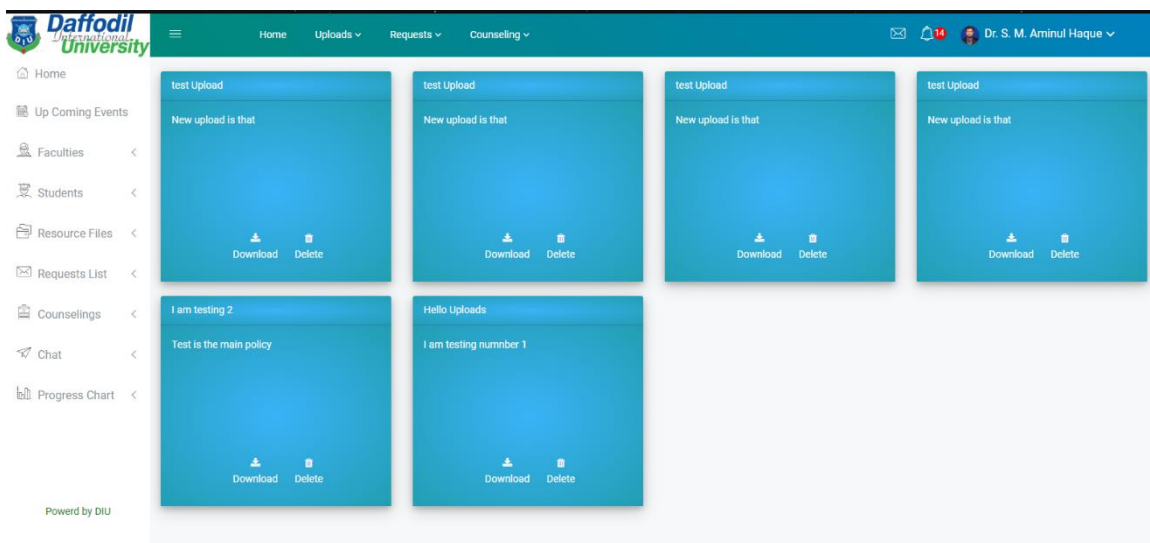


Figure 4.15.3: All Uploaded Folders

If the admin login into our application as admin then they can see the admin home page (figure 4.15.4). On this page, they can get the option to see all the faculty information as

well as all the student information. They can also see all their uploads from the upload menu. They can also see all the notification from the notification icon. If they don't have to read these notifications then they can get the option to mark them as all read.

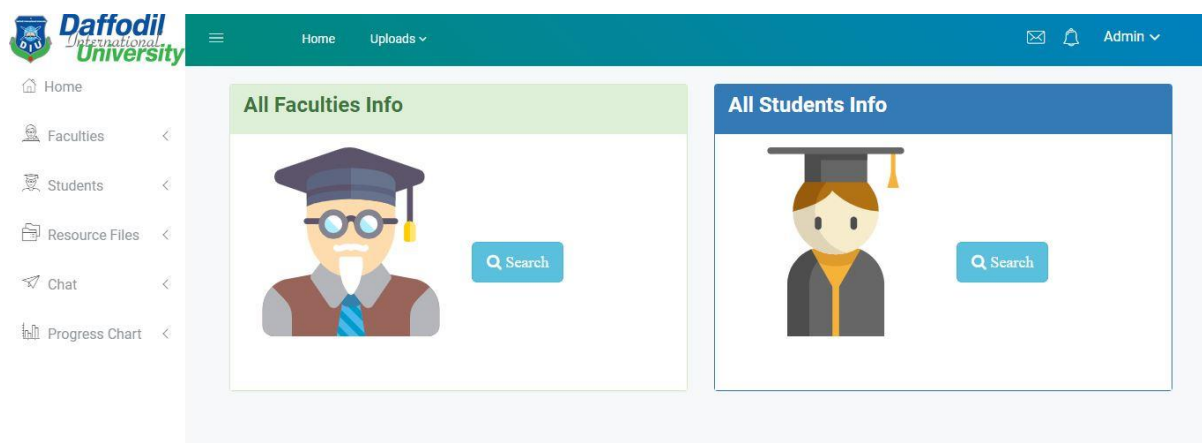


Figure 4.15.4: Admin Home Page

Now if admin wants to see all the information about faculty then they can see from the faculty list page (figure 4.16) where they can see all the information about faculty members such as their name, id, email, contact, profile.

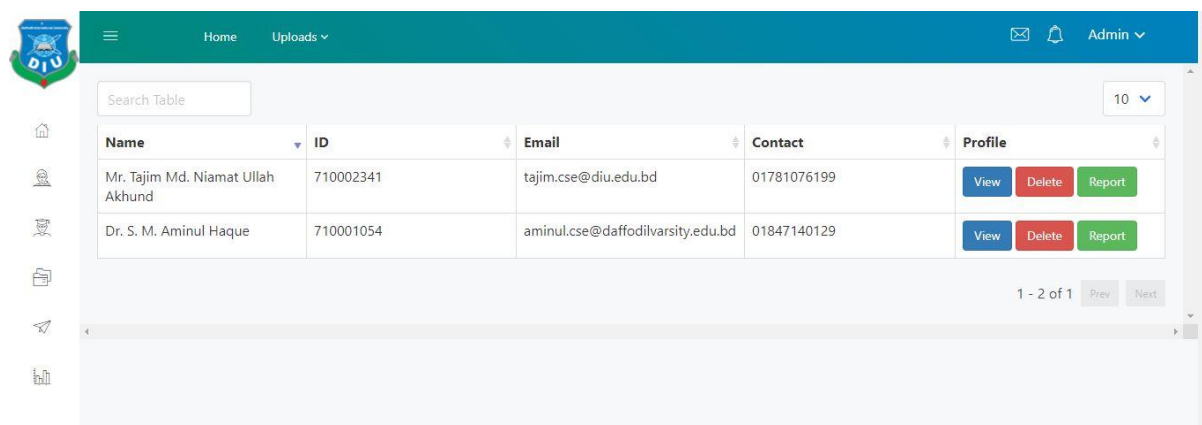


Figure 4.16.1: Faculty List page

If admin wants to see any faculty member profile then they can also see their profile from the view option. Here admin can see all the detailed information about the faculty member



such as name, email, contact number, designation, room address, area of interest and if the faculty member has any personal website then they can also see their website.

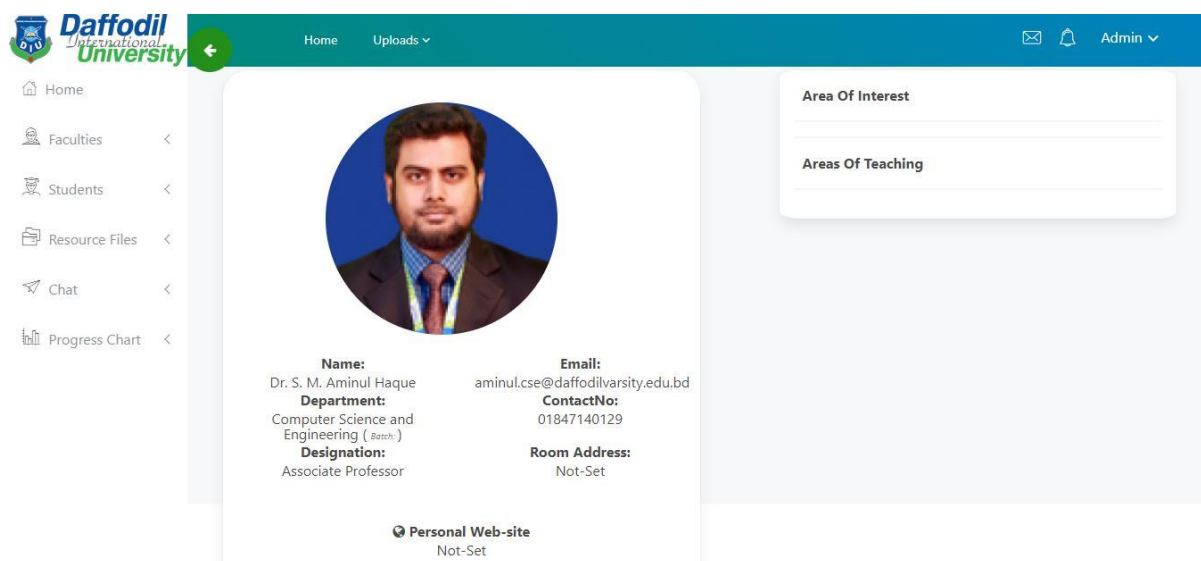


Figure 4.16.2: Faculty Profile Page

If admin wants to generate any report about any faculty member then they can get the option from the faculty report page (figure 4.17) where they see the faculty member name, they get an option to select a specific month and year.

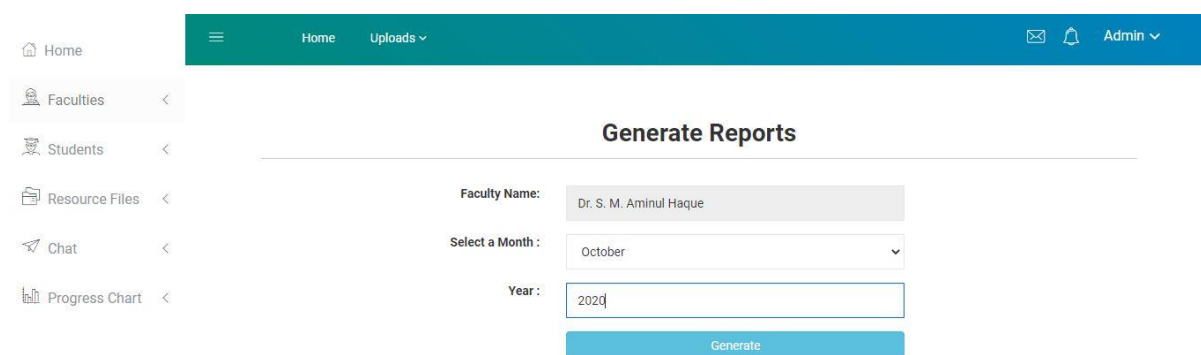


Figure 4.17: Faculty Report Page

They can also see the report from the report to generate the page (figure 4.18). Where they get the option to download the full report of that faculty member as a pdf format or if they want to see the summary of that pdf then they also get that option too.



Figure 4.18: Report Generate Page

In the same way, admin can also see student details information from the student list page (figure 4.19) where they can see then details about these students such as students name, id, email, contact as well as their profile. They can get the option to view, delete, and generate a report in the same way they got it for teachers.

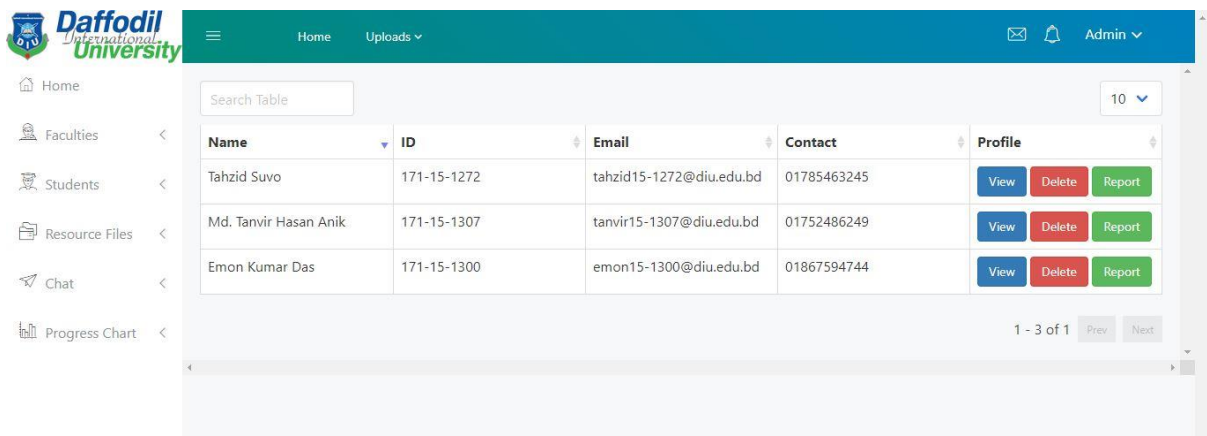


Figure 4.19: Student List Page

As faculty members and students can upload their new file in the same way admin also get the option to upload their new files from the upload menu bar. They can also see all the uploads as well as their new uploads from the upload menu bar. Admins also have the opportunity to chat with any faculty members as well as students. If they want to discuss an urgent issue, they can do so from the chat page (figure 4.20).

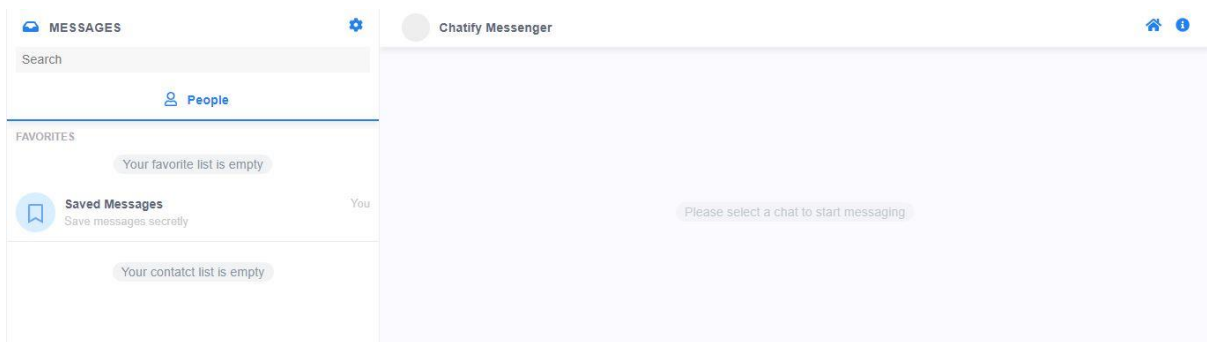


Figure 4.20: Chat Page

All the front end designs are responsive and have support for all the browsers and devices with JavaScript supports. This application also has responsive support to mobile and smart devices with HTML and JavaScript support.

## 4.2 Back-End Design

We have used a traditional stack of technologies that are used all over the world by the big companies for the reliability and safety they offer and last but not the least for scalability the support they give since our application is not fully equipped for supporting a huge number of traffics at this moment so it needs scalability in the future for the enhancement of this application the stack we used for backed is the following:

- Server-side Language: PHP.
- Frame-Work: Laravel which is the most used PHP framework for data extensive applications.
- Database: MySQL
- Server: Linux native LAMP stack

We have used PHP as our server-side language because of the enrichment of libraries and helper methods and the object-oriented feature it offers other than that it is a veteran of web technologies and still standing high above other newer server-side languages. World's 60% of the websites and web applications use are still running PHP in their back-end.

We choose Laravel for the out-of-the-box security and scalability it gives and Laravel is the preferred choice for the PHP community for large scalable projects that is prone to scalability in the future and our application is sure to scale in the future if needed.

In our application, we used some service worker type implementation since we have a reminder feature in our application hence the server has to run a service worker in the back in to resolve this issue, we are using crontab to serve cronjobs of our application.

We have used MySQL DBMS as our database for the sake of relational database instead of a NO-SQL DBMS since our application uses a lot of relational tables to resolve data management and to build up the relation between the data to extract and fetch data as desired.

### **4.3 Interaction Design And UX**

Interaction means reciprocal action or influence. Interaction design means ongoing interaction between the users and the products. so when we start designing interactions for this application then we are trying to make it feel natural.

We are always trying to make the words very simple so that our user can easily interact with this. We try to use short and most commonly used words so that they feel comfortable when they use our application. When we design our interaction then we continuously ask ourselves “What would a helpful application do?”, “what would a thoughtful, considerate interaction feel like?” For this reason, the form that we are used in our application is very well known. After seeing this form, a user can easily understand what they should do. We make our application responsive so that when a user uses our application, they feel very comfortable. Again, the loading time of the application is very fast, it can move one page to another page in some millisecond.

For designing the UX part of our application, we are doing some online surveys, we are taking some interviews we can make the best UX part for our application. We also made some kind of testing like usability testing for a better understanding of how our application UX design parts are working.

## 4.4 Implementation Requirement

Here we are developing a web-based application and for making this project we need to use lots of front-end language, back-end language, frameworks, and also databases so that it looks more attractive and it works very smoothly.

In the front-end part of our application, we use some well-known front-end language like HTML (Hyper Text Markup Language), CSS (Cascading Style Sheet), JavaScript. We also use some of the well known frameworks such as Bootstrap, libraries such as jQuery, Ajax, and open standard file formats such as JSON. And all live updates of notification and Data tables are implemented with the JavaScript framework Vue JS.

As we all know that for giving a proper structure of any web-based application we must have to use HTML that's why we use HTML for giving our application proper structure. HTML is used for making a proper static page of our web application. After that, we use CSS so that we can provide a look and feel of the content that is written in HTML. By using CSS, we change the font size, font color, background of our web page and we also set proper margin and to our webpage. But for converting our project static to dynamic as well as interactive web use JavaScript so that it can give the best user experience and our user feels comfortable to interact with our application. Then we use bootstrap (CSS framework) which is used to make our website responsive and cross-browser compatibility. We also used jQuery so that our plugin works nicely and properly. In the back-end part, we use a server-side language PHP (Hypertext Preprocessor) and its framework Laravel for making our application dynamic. We also use the MySQL server as a database tool. Mainly we use Laravel because it is an open-source PHP web framework and it follows the MVC (model-view-controller) architecture. We use Laravel because our project is very large and it contains a lot of functionality and for developing large and robust applications Laravel is best because it is reliable and it is open source means its source code is available for everyone. Again, it is also best for authentication and it can easily integrate with mail services. Laravel also provides us a lot of built-in functionality such as authentication, caching, session, etc. That makes the development process much faster. And for adding, updating, deleting, reading data from our database we use MySQL which is a relational

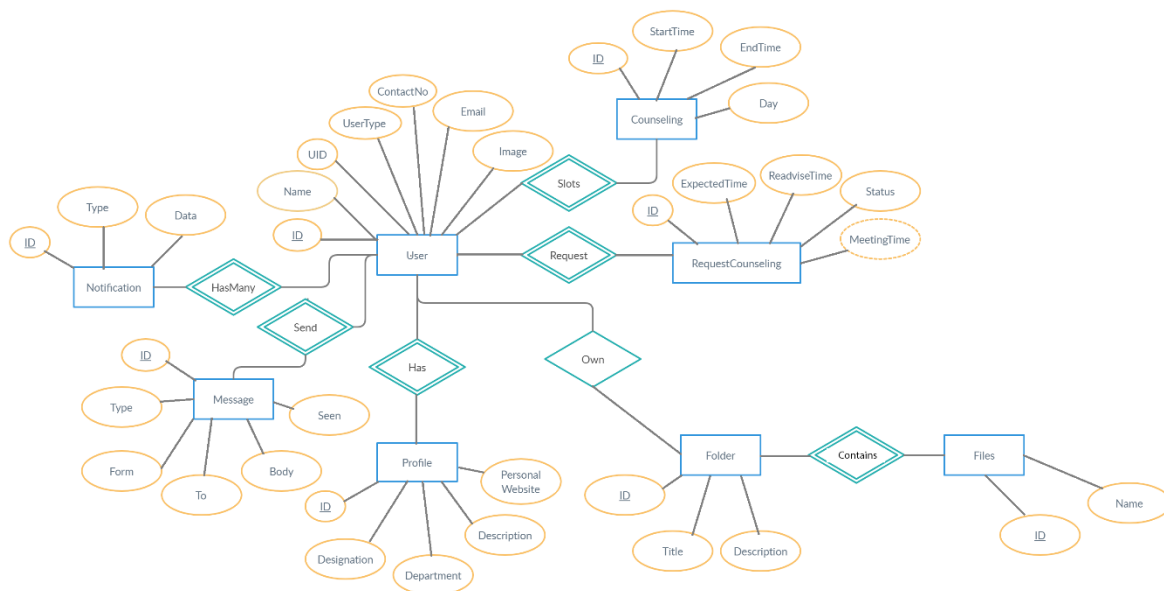
database management system. We use MySQL because it is lightweight and it is completely free. It is also cross-platform means we can use MySQL on a windows server and also Linux server.

## CHAPTER 5

### Implementation and Testing

#### 5.1 Implementation of Database:

For our project database, we have used the MySQL database, and for the server-side interactions, we have used one of the famous frameworks of PHP, The Laravel framework. We have modeled our database following the rules and regulations and helping of the framework. Forget a scenario of the database tables and its attributes a simplified ER relation diagram for this database has been given below:



**Fig 5.1.1: Entity Relationship Diagram**

As we have worked under Laravel Framework the database implementation and the configuration has become very easy. Step-by-step description and command line of the



implementation of the database are given on the next page. Any server with MySQL services will be able to implement our database as follows.

First, we have to log in to the MySQL server as a user by its username and password with the privilege of Data Manipulation and Data Definition.

- i) we have to create a database as our wished name by the command given below.

```
CREATE DATABASE database_name;
```

- ii) From our project directory, we have to configure the environment file of the project. The environment file of the project is named the **.env** file. In the **.env** file, we have to give the database name and the servers IP address information as follows with database username and password.

```
DB_CONNECTION=mysql
```

```
DB_HOST= servers_ip_address
```

```
DB_PORT=3306
```

```
DB_DATABASE= database_name
```

```
DB_USERNAME= username_of_database_access
```

```
DB_PASSWORD= password_for_the_access
```

- iii) By giving accessing information to the database in the **.env** file we have to just run a simple command from our projects directory under any apache server.

```
php artisan migrate
```

It will give feedback that migrated successfully just like below:

```
$ php artisan migrate
Migration table created successfully.
```

Fig5.1.2: Migration Confirmation Feedback

All the list of the tables created will appear also on the screen:

```
$ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.03 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (0.01 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0.01 seconds)
Migrating: 2019_09_22_192348_create_messages_table
Migrated: 2019_09_22_192348_create_messages_table (0.03 seconds)
Migrating: 2019_10_16_211433_create_favorites_table
Migrated: 2019_10_16_211433_create_favorites_table (0.02 seconds)
Migrating: 2019_10_18_223259_add_avatar_to_users
Migrated: 2019_10_18_223259_add_avatar_to_users (0.01 seconds)
Migrating: 2019_10_20_211056_add_messenger_color_to_users
Migrated: 2019_10_20_211056_add_messenger_color_to_users (0.01 seconds)
Migrating: 2019_10_22_000539_add_dark_mode_to_users
Migrated: 2019_10_22_000539_add_dark_mode_to_users (0 seconds)
Migrating: 2019_10_25_214038_add_active_status_to_users
Migrated: 2019_10_25_214038_add_active_status_to_users (0 seconds)
Migrating: 2020_02_03_002747_create_counselings_table
Migrated: 2020_02_03_002747_create_counselings_table (0.03 seconds)
Migrating: 2020_02_21_164626_create_faculty_profiles_table
Migrated: 2020_02_21_164626_create_faculty_profiles_table (0.03 seconds)
Migrating: 2020_04_01_151251_create_notifications_table
Migrated: 2020_04_01_151251_create_notifications_table (0.03 seconds)
Migrating: 2020_04_04_162316_create_request_counselings_table
Migrated: 2020_04_04_162316_create_request_counselings_table (0.05 seconds)
Migrating: 2020_06_15_100513_create_folders_table
Migrated: 2020_06_15_100513_create_folders_table (0.03 seconds)
Migrating: 2020_07_10_150054_create_file_uploads_table
Migrated: 2020_07_10_150054_create_file_uploads_table (0.03 seconds)
Migrating: 2020_07_18_134652_create_student_profiles_table
Migrated: 2020_07_18_134652_create_student_profiles_table (0.03 seconds)
```

Fig 5.1.3: Migrated table list feedback.

Through this process, we can easily implement the database. It will automatically generate the database tables and attributes with necessary relations.

## 5.2 Implementation of Front-end Design:

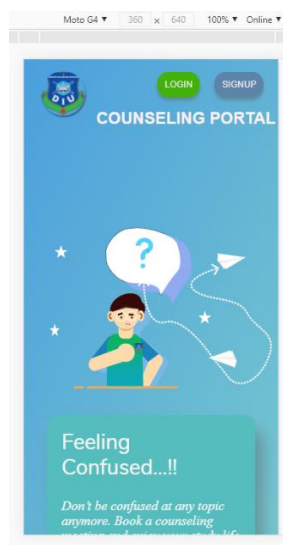
The implementation of the front end is done with the help of HTML, CSS, Bootstrap, jQuery, and some libraries, plugins, and frameworks of JavaScript. Live notification updates and data tables are implemented with the framework of JavaScript Vue JS. File Uploads are done with the help of drop-zone JS.

The project is fully responsive with all size of desktop screen and mobile screen that has the JavaScript support.

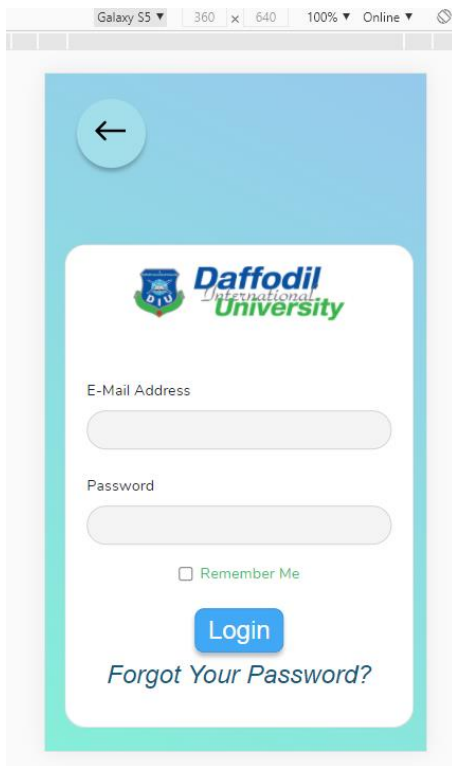
For it is media responsive UI and UX for all devices that support JavaScript plugin it is ready to deploy to the online

A mobile menu has been introduced for mobile applications for more interactive uses in smartphones.

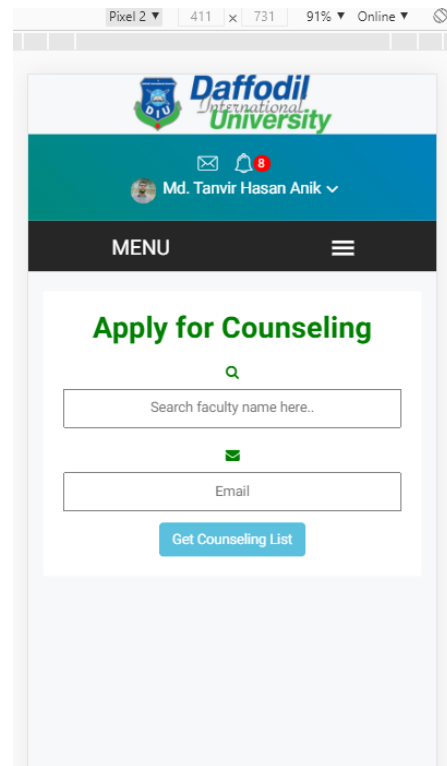
Example:



**Fig 5.2.1: Mobile Responsive Landing Page**



**Fig 5.2.3: Mobile Responsive Login Page**

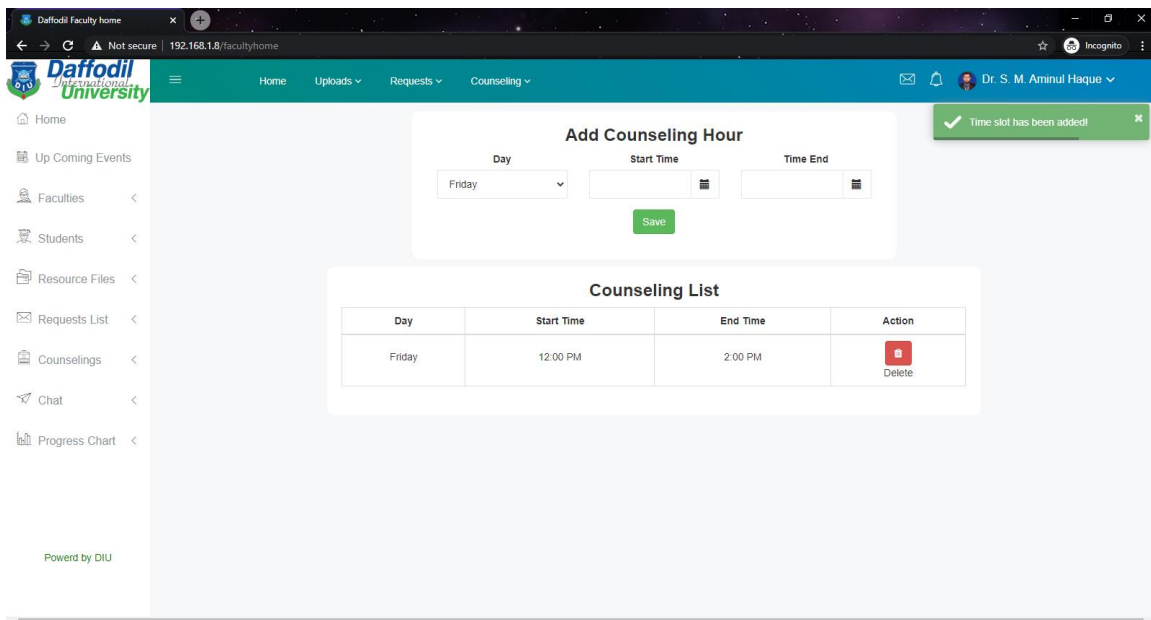


**Fig 5.2.4: Mobile Responsive Menu**

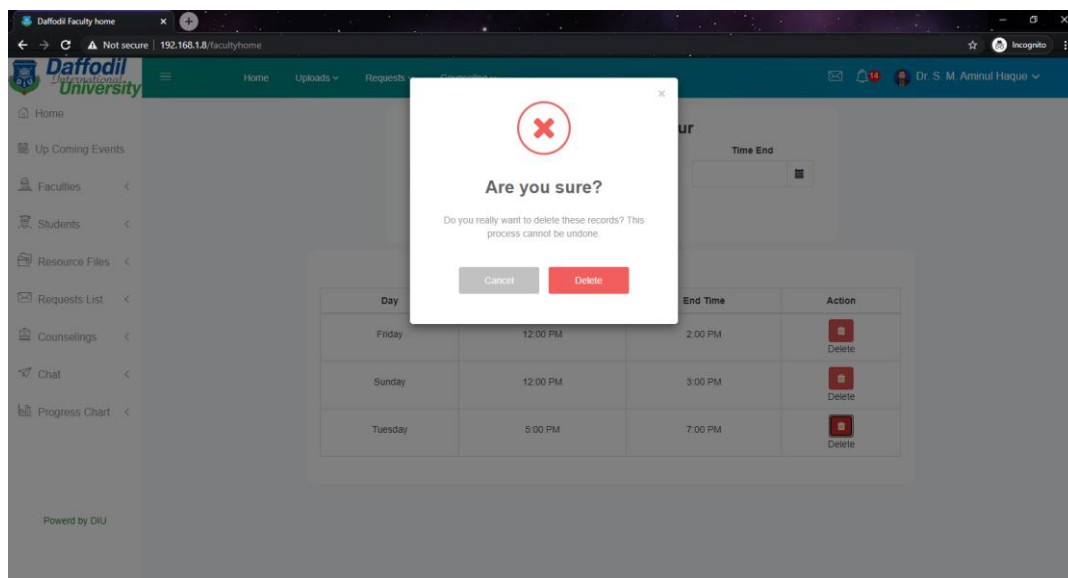
Any size screen of the modern desktops can get a simple, easy to use and user-friendly front-end user experience from this application alongside smartphones.

### 5.3 Implementation of Interaction:

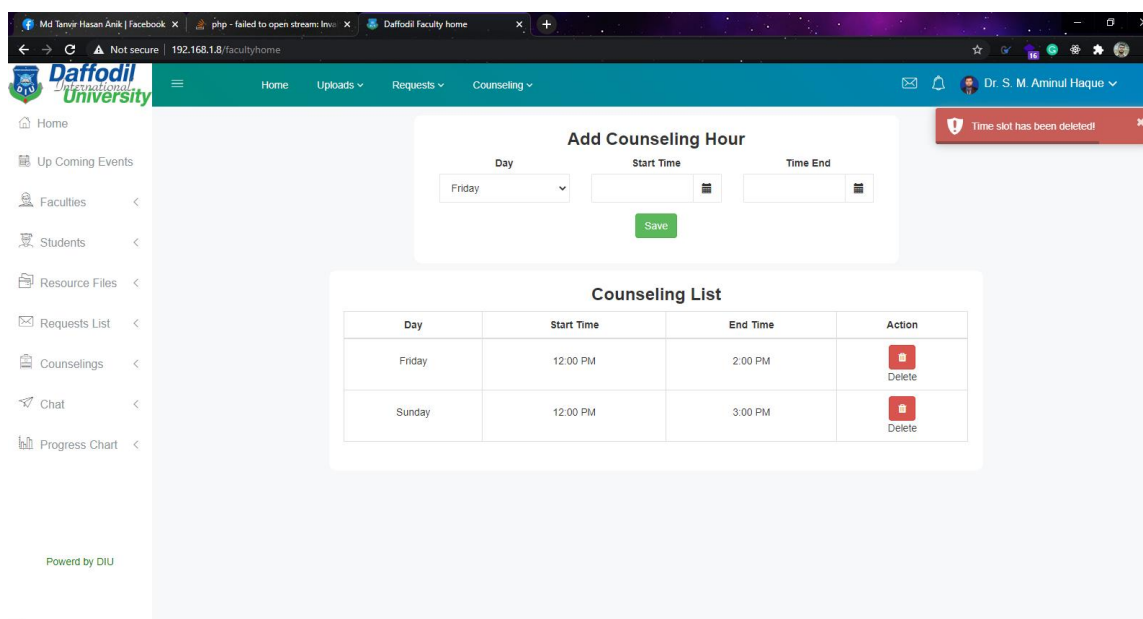
The application user interface will give a user-friendly interaction like necessary alerts, notifications, and modal boxes. The server-side response will give feedback throw toaster alert and modal dialogs. Necessary instruction and vocabulary are easy to understand to interact with the application.



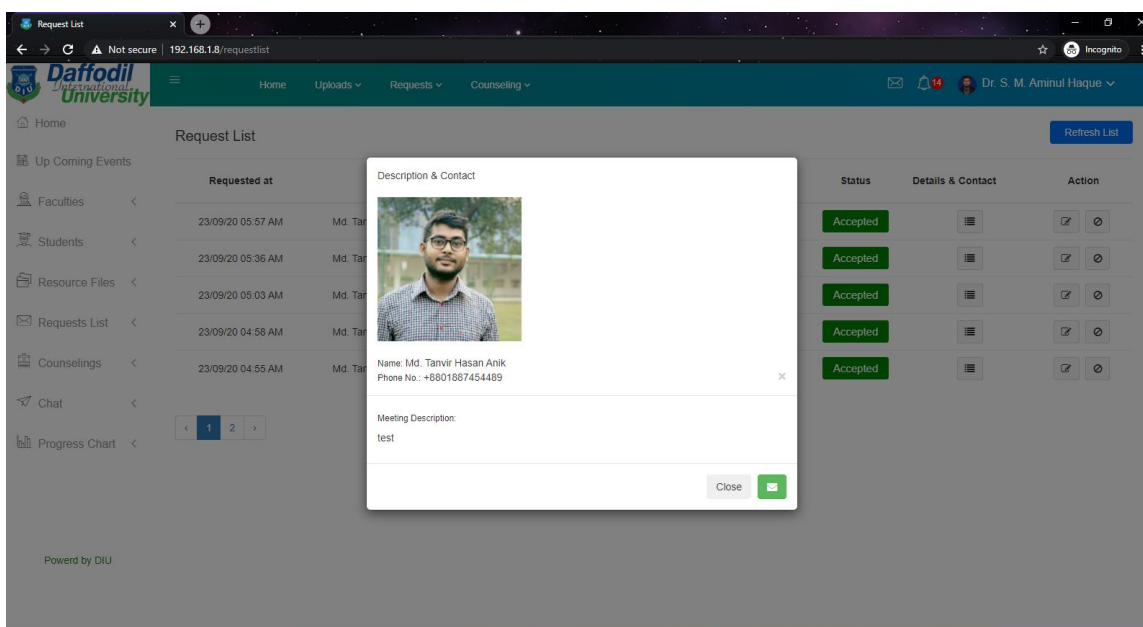
**Fig 5.3.1: Toaster Alert for successful interaction**



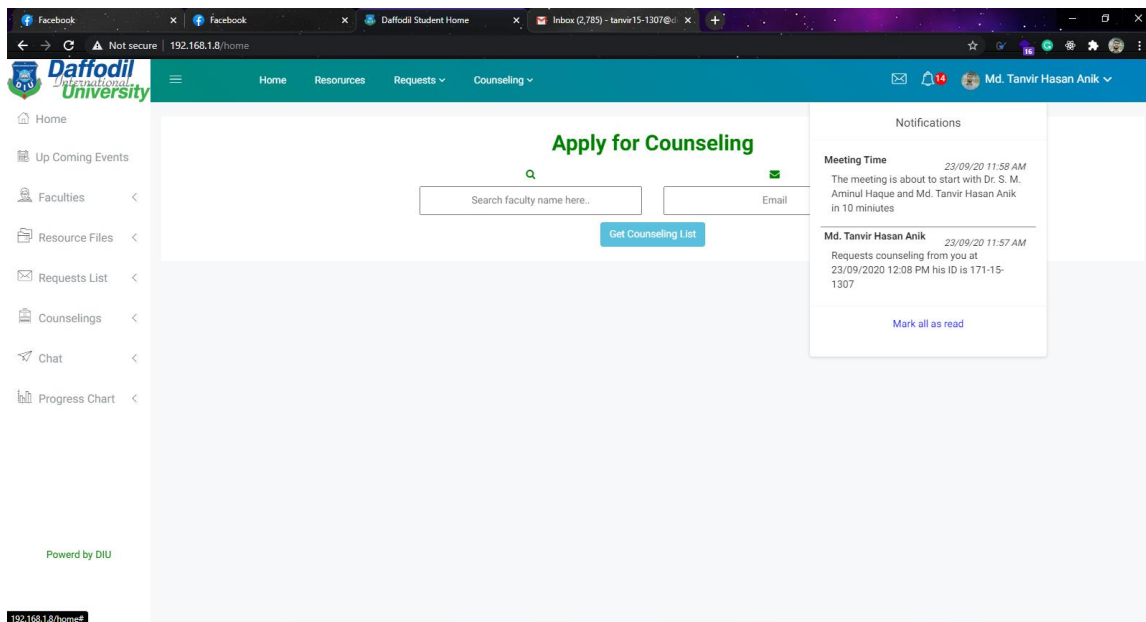
**Fig 5.3.2: Delete Modal Confirmation Interaction**



**Fig 5.3.3: Delete Confirmation Toaster Alert**



**Fig 5.3.4: User Details Interactive Modal**



**Fig 5.3.5: Live update of notification**

A server-side confirmation for all the interaction will be notified to the user. The Session for every user is set for 2 hours from login. If the user is not active for more than two hours than the user will automatically be logged out by the application server.

#### **5.4 Testing Implementation:**

Every part of the code's test is done as soon as it is written. So, the unit test has been done wisely for the project. All the interaction and responses are re-checked that it is working correctly.

We have followed and analyses all the requirements for the project and testing implementation has assured that all the functionality is working exactly the users require.

The performance, scalability, and security of the project is improved with the help of the framework.

### 5.5 Test Results and Reports:

We have gone through several tests for the overall projects. And the result of the test is okay and passed by the project a list of test cases and its result is given in the tabular form here in the below.

**Table 5.5: Test Results**

SL no.	Name of Test	Report and Result
1.	Functional Test.	Okay and Passed.
2.	Security Test	Okay and Passed.
3.	Scalability Test	Okay and Passed.
4.	Performance Test	Okay and Passed.
5.	Regression Testing	Okay and Passed.

So testing is giving reports and results that the overall project is passed the quality assurance and any needs and supports in the future can be assured for this project.



## CHAPTER 6

### CONCLUSION AND FUTURE SCOPE

#### 6.1 Discussing and conclusion:

The counseling hour management system is a web-based application that is mainly used to manage to counsel properly. Where students can apply for their respective teacher counseling, they can also see the counseling time slot and according to the counseling time slot, they can apply for counseling with a short description of their problem.

In our application, faculty members can add their counseling hours. They can also see all the counseling request which comes from the students. They can also upload their necessary documents into our application. Again, admin can also use our application. By using this application admin can easily see all the faculty information as well as student information. For developing this project, we had to learn so many languages like HTML, CSS, JavaScript, PHP as well as some well-known frameworks such as bootstrap, Laravel, etc. Our counseling hour management system is developed by using these above languages, framework, libraries, and also, by maintaining interaction and UX design so that our user feels very comfortable while they use our application.

#### 6.2 Scope for future development

We implement all the necessary functionality in our counseling hour management system and it is almost ready to use for counseling. But still, it has a lot of scope for future development. Again, if these actors want to get any additional functionality then we will implement this. We will also try to implement live audio and video call so that students and teachers can interact online and solve their issue very easily. We will also try to implement group chat like messenger so that a group of people also interact with one another, again they can also discuss their problem in their respective groups.