Computer Science and Applications / Computer Information Systems / Software Engineering

Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology

The Hashemite University, Zarqa-Jordan

Online Attendance System

HU Chat Subsystem

By

Abdelaziz I. Al. Tally (1739073)

Khaled O. Eilayan (1738441)

Supervised by

Dr. Abdelrahman F. Al-Ghuwairi

Submitted in partial fulfillment of the requirements of B.Sc. Degree in Computer science

January 2021

It is hereby certified that the project titled ***Online Attendance System*** and ***HU Chat System***, submitted by ***Abdelaziz I. Al. Tally*** (1739073) and ***Khaled Eilayan*** (1738441) in partial fulfillment of the award of the Degree of Bachelor **Software Engineering**, **Business Information Technology** embodies original work done by them under my supervision.

Project Supervisor: Dr. Abdelrahman Al-Ghuwairi

Department of Software Engineering

ABSTRACT

Student attendance system deals with the maintenance of the students' attendance details. It generates the attendance of the student on basis of presence in class. It is maintaining daily basis of attendance; the staff will be provided with the separate username and password to make student attendance.

The Students will be responsible of making the attendance by signing into the official website of their university, and the attendance will be calculated in real time the student's attendance report will be recorded as soon as the lecture ends.

Also provides an easy, formal and fully secured way for communication between all users of the system to discuss Various topics by using online chat system.

DEDICATIONS

This project is proudly dedicated to my beloved university, all of its instructors and employees, especially our fellow Dr. Abdelrahman F. Al-Ghuwairi, and a special thanks to our families and friends who helped us along the way.

TABLE OF CONTENTS

Table of Contents

Abbreviations

1. Online Attendance System – (OAS)
2. Magnetic Identification Card – (MIC)
3. University Identification Card – (UIC)
4. Fingerprint Identification Card – (FPBI)
5. Integrated Automated Fingerprint Identification System – (IAFIS)
6. Automated System Detection – (ASD)

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The satisfaction that accompanies the successfulness completion of any task would be incomplete without mentioning the people whose ceaseless cooperation made it possible.

We are grateful to our project supervisor Dr. Abdelrahman F. Al-Ghuwairi for the guidance, inspiration and constructive suggestions that were helpful to us in the preparation of this project.

CHAPTER 1: INTRODUCTION

In this chapter we will give brief overview about the Online Attendance System (OAS), in specific; we will talk about our motive to produce this system, and the difficulties that faced the instructors and the students in various universities, also we will talk about what we would want to achieve after developing this system.

1.1 Background

Until now, most of the attendances in institutions are still recorded manually by calling out the name one by one by the lecturer. This seems to be wasting time especially when the number of students is huge. There is also another way that the lecturer passes the attendance sheet to student one by one to sign beside their name. However, this way will lead to one drawback which is the student can help to sign for the student that are absent. Therefore, the only solution to overcome this kind of problem is to manage the attendance automatically. In order to avoid this issue, wireless attendance identification system based on a specific password generated by the lecture instructor and the student personal device MAC Address. The MAC Address has a lot of pros, such as uniqueness and it lasts forever.

Also, we noticed that the communication between students and instructors has its drawbacks and given that nearly all of us are familiar with various chat systems (like WhatsApp, Messenger, etc.) why not develop a chat system that will be part of the Students' / Instructors' / Deans' respective portals.

So, we introduced a new chat platform to make the communication easier for all the members of the university.

1.2 Problem Statement

Attendance is a state of being present at a place. It represents how committed a person is in real life or working environment. Until now, the attendance is still being taken manually as we sign on the name list that lecturer prepared and submit after signing. In this circumstance, some students may help other absent students to mark their attendance. Therefore, an accurate system is needed to be invented to solve this problem. In order to have an accurate system, we have thought about installing devices to read magnetic cards in every class, an ID is necessary, it means every student will have their own University Identification Card (UIC) that can differentiate between students. Although it is already magnetic and is available in our University and ready to be used, but it is still not secure because the tools and devices that enable us to use this feature are not available at the present time. We also thought about introducing the Fingerprint Biometric Identification (FPBI) because it is unique for everyone, so it is safe to use, however, this system will need a lot of wires to connect if it is installed in every classroom in the University, and it will take a lot of time to take the (FPBI) of all the students.

Communication between students and instructors is vital, especially in these circumstances while having online classes.

Although a dedicated email has been given to each student by the university not so long ago. Yet, there’s still some students who communicate with their instructors on other platforms rather than reaching out to them by email.

Like sending messages to instructors on their private Facebook profile, or at WhatsApp, and recently on Microsoft Teams.

1.3 Objective(s)

The main objective of the project is to :

- Demonstrate an (OAS) that can automatically acquire students' attendance.
- Develop an easy, formal and secure way for communication between all users.

In order to accomplish this we have to:

a. Go with computerized system, instead of old-fashioned manual process.

b. Develop a verification system on attendance marking**.**

c. Acquire the data using MAC Address and store the attendance in the database.

d. Display the absence of every student via their Portal.

e. No paperwork, and no risk of errors which can be occurred while marking the attendance manually.

f. Reports are easily generated.

g. User Friendly: User Interface is very easy. Data storing and recovery is fast and secured.

h. A way to communicate between users especially the student and his instructor.

1.4 Report Organization

The rest of the report is organized as follows:

- ***Chapter 2*** *talks about a couple of solutions that serves the same purpose.*
- ***Chapter 3*** *lists the user requirements and the way they have been elicited.*
- ***Chapter 4*** *presents the design and components' architecture.*
- ***Chapter 5*** *contains an implementation, testing methodologies and cases on the system as a whole.*
- ***Chapter 6*** *presents the future work and concludes the report.*

CHAPTER 2: LITERATURE REVIEW

Attendance system has already been introduced to the world and put into use in two main ways: Verification Card Identification and the Fingerprint Biometric Identification. But it is difficult to implement these ways and use it in the University for reasons already stated before.

YEAR	Invention Name	AUTHOR(S)	OBJECTIVES	PROBLEMS
1950s	Magnetic Identification Card (MIC)	Forrest Corry Parry	Storing data by modifying the magnetism of tiny iron- based magnetic particles on a band of magnetic material on the card.	The student can help to sign for the student that are absent. The tools and devices that enable us to use this feature are not available at the present time.
1920s	Integrated Automated Fingerprint Identification System (IAFIS)	The New York Police, FBI	Uniquely identify a person since that not two humans have the same fingerprints.	This system will need lots of wire to connect if it is installed in every class in the University. It will take a lot of time to take the fingerprint biometric identification of all the students.

Table 1 : Possible Solutions Review

In general, using (MIC) and (IAFIS) meets our requirements and our vision for a better future, but sadly it is not in our capabilities at the moment to use it in the University.

We have taken a look at many other systems that aimed to accomplish the same idea but none of them were executed the right way, so we used the weaknesses in their systems to make our system more complete, secure and reliable.

Our system opens new horizons in the future to use either the Iris or Facial Recognition.

CHAPTER 3: SOFTWARE REQUIREMENTS

The SRS document itself states those functions and capabilities in precise and explicitly a software system must provide and state any required constraints by which the system must abide to. It is important to note that an SRS contains functional and non-functional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information.

3.1 Requirements Elicitation

As we discussed before the old method of taking attendance which is still used to this day, takes a lot of time and effort from instructors in universities, and cannot be relied on for the reasons mentioned before, so the process of gathering requirements is very important for this program because the problem is not just for one university but in most universities.

The process of eliciting requirements in this type of system takes a lot of time because you have to collect them from various instructors and students and make accurate statistics to find the satisfactory requirements that lead the program to success, and a result of its use and reliance on it in their universities ensures that the user has a comfortable and satisfying user experience.

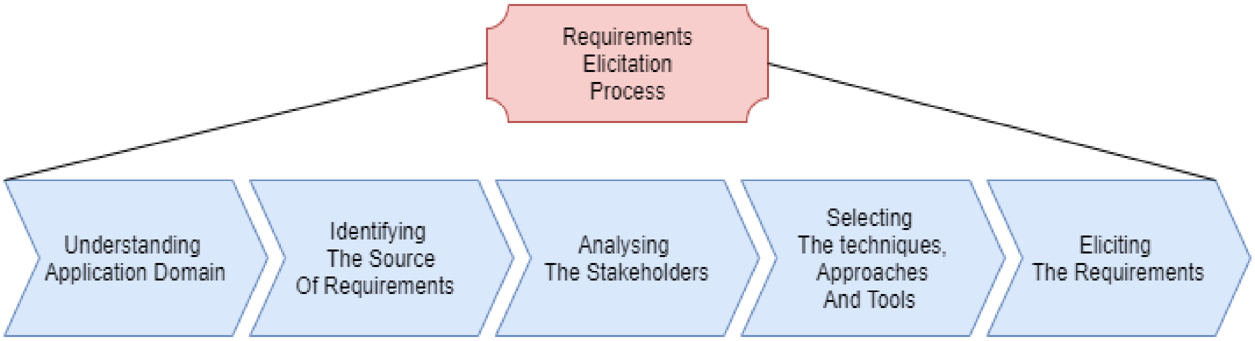

Requirements Elicitation Process
Understanding Application Domain
Identifying The Source Of Requirements
Analysing The Stakeholders
Selecting The techniques, Approaches And Tools
Eliciting The Requirements

Figure 1 : Requirements Elicitation Process

3.2 Functional User Requirements

They are what the system must do and can be thought of in terms of how the system responds to inputs. In the table below; we will be stating some of the functional requirements for the OAS.

No.	Requirements	Frequency of use	Users
1.	The system notifies the dean when a certain student's absence exceeds 20% of the lectures in the semester.	When necessary.	• System
2.	Signing in.	Upon the request.	• Faculty Dean • Instructors • Students
3.	The system should view the attendance for all users.	Upon the request.	• Faculty Dean • Instructors • Students
4.	The system should add new Instructors/Students.	Upon the request.	• Faculty Dean
5.	Add new Courses	Upon the request.	• Faculty Dean
6.	The instructor should be able to remove student's absence if he provides a valid reason for it	Upon the request.	• Instructors
7.	The student must be able to record his attendance for each lecture.	In each lecture.	• Students
8.	Student must be able to register in the courses he wants.	Registration period	• Students
9.	The student must be able to view his registration table for the current semester.	Always	• Students
10.	Messages between two users shouldn't be accessible by others	Always	• Faculty Dean • Instructors • Students

Table 2 : Functional User Requirements

The use case diagram below shows the actions that can be taken by each user.

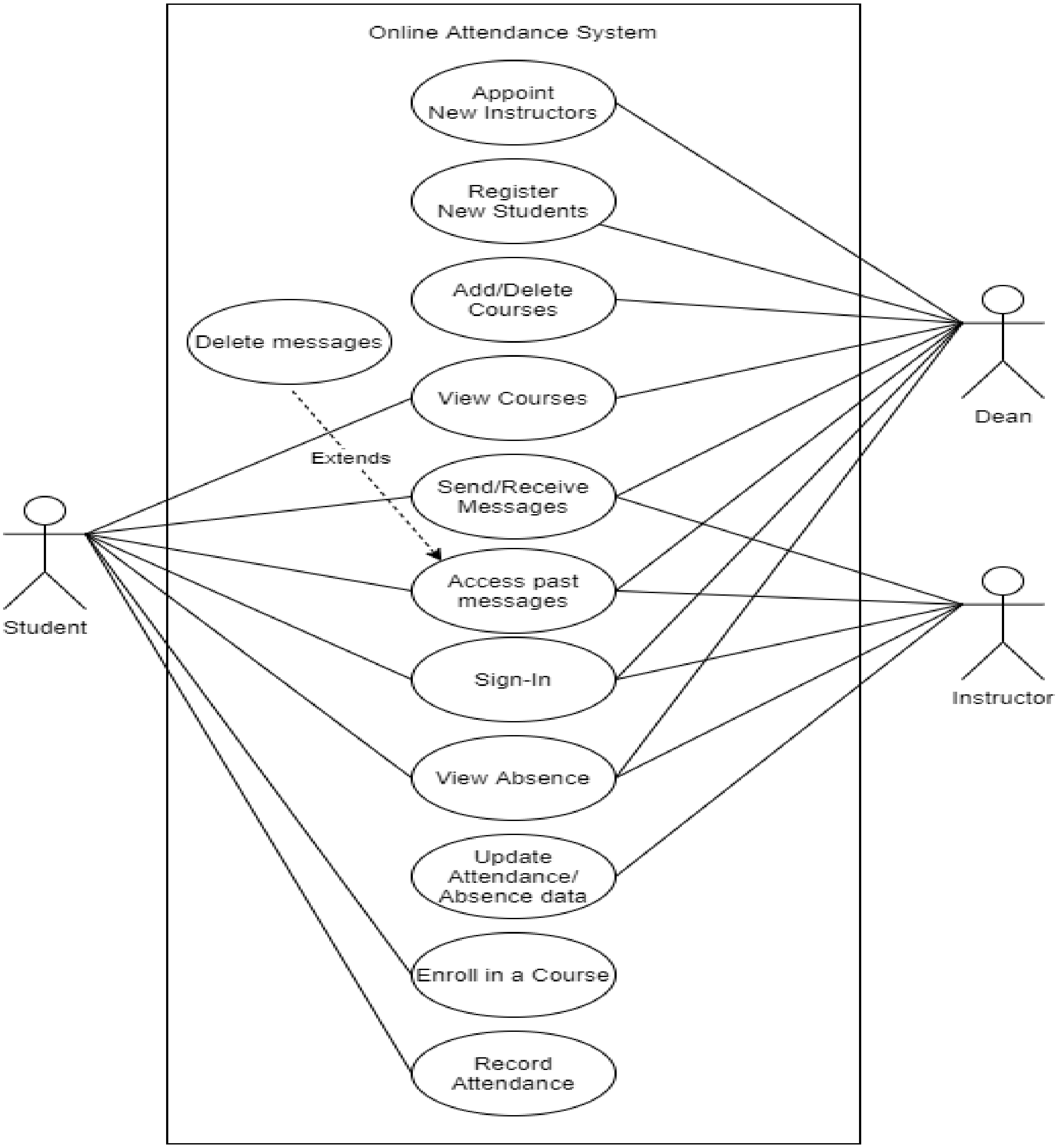


Figure 4 : Use Case Diagram

3.3 Non-Functional User Requirements

Broadly, functional requirements define what a system should do, on the other hand non-functional requirements define how a system is supposed to be and also can be omitted because it defines the quality of the implementation of the system.

The table below shows the complementary requirements, sorted by their priorities, to enhance the quality of the system.

Non-Functional Requirements Classifier	**Non-Functional Requirements**	**Priority Level**
Security	The student can only view his personal data.	HIGH
	The student can't submit the attendance for others.	
	Passwords are encrypted.	
Reliability	The system data should be accurate.	HIGH
Portability	Application must be able to run on different environments.	HIGH
Performance	Queries executed without delays.	MEDIUM
Availability	The system supposed to be available at all times.	MEDIUM
Usability	Ease of use.	MEDIUM
Maintenance	Any modification on the database or the system will be easy to handle.	MEDIUM
DESIGN (UI/ UX)	Interactions with the systems is straight forward and easy.	LOW

Table 3 : Non-Functional User Requirements

CHAPTER 4: SOFTWARE DESIGN

One of the most important factors that affect the program is the design that contains many types that work differently, some of which affect the user experience, such as the interface design, including what affects the flexibility of the system and its capabilities such as database design and the interaction between them, so the design must be good enough not to adversely affect the system.

4.1 System Architecture

System architecture is the ideal model that determines the structure and behavior of a system. This description is of a high standard that can demonstrate the relationship between the components that stimulate software and hardware and their communication. The design of the OAS and the Chat System is based on the client/server architecture. A web application serves as an interface between the users and the service provided by the system.

The first level of this architecture is the hardware component. The device here is the user device that will be connected to the system and other components. After that, the architectural level will go to the task of the process when the student tries to access the server through an inquiry or registration of attendance where it is linked with the server (which is connected to the database) via the Internet, then this data is received and processed and verified as correct as the server communicates with the database, the required tasks are executed accurately.

4.1.1 System Architecture

This figure shows the general architecture of the proposed system which includes the server, database and shows the interaction between them.

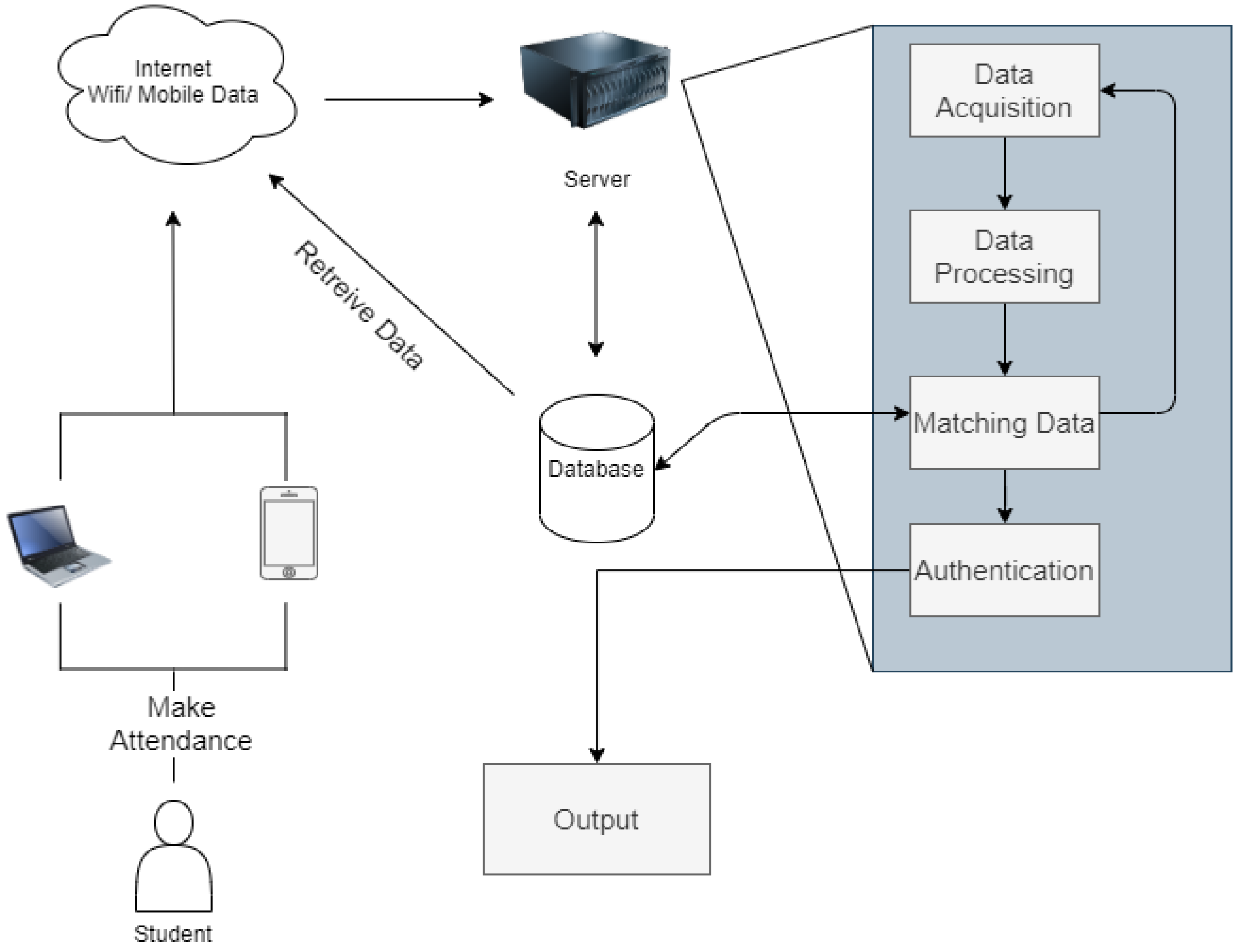


Figure 7 : OAS Architecture

4.1.2 Sequence Diagram

To get more understanding of the components of the system and the method of interaction between them and to see the details of the requests by the user and the method of completing them by the server and the database:

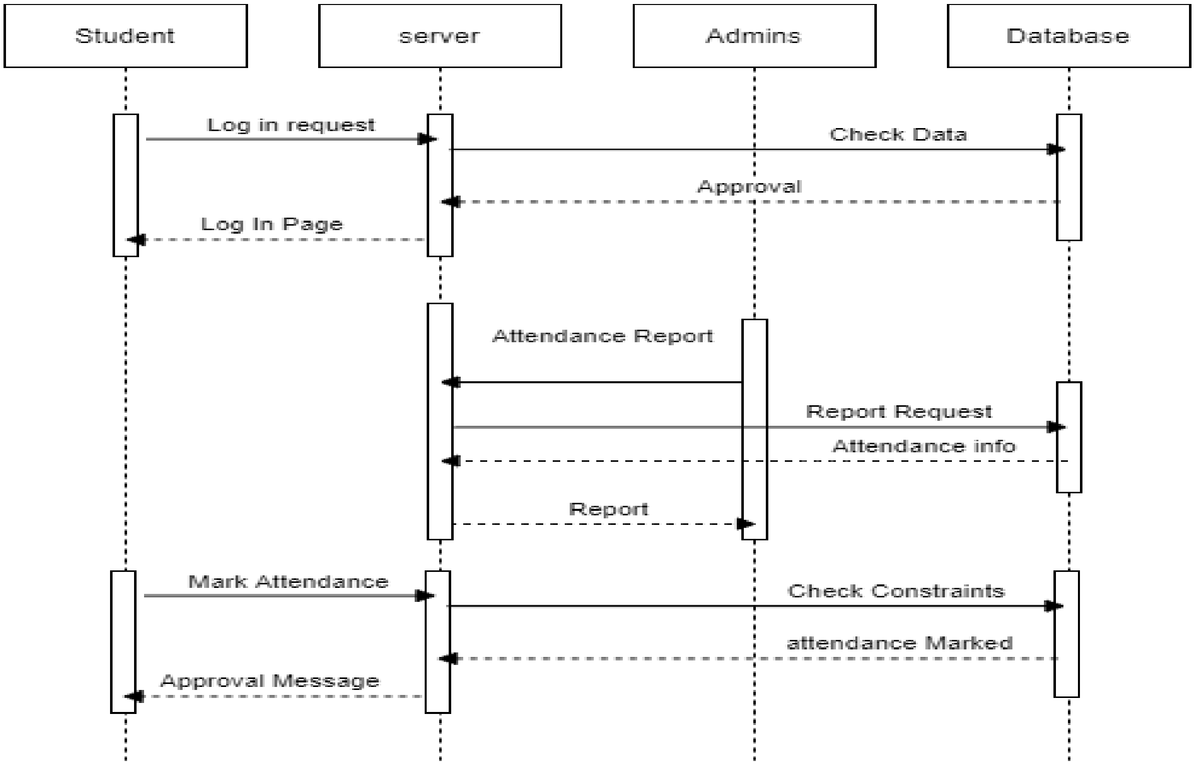


Figure 10 : Sequence Diagram

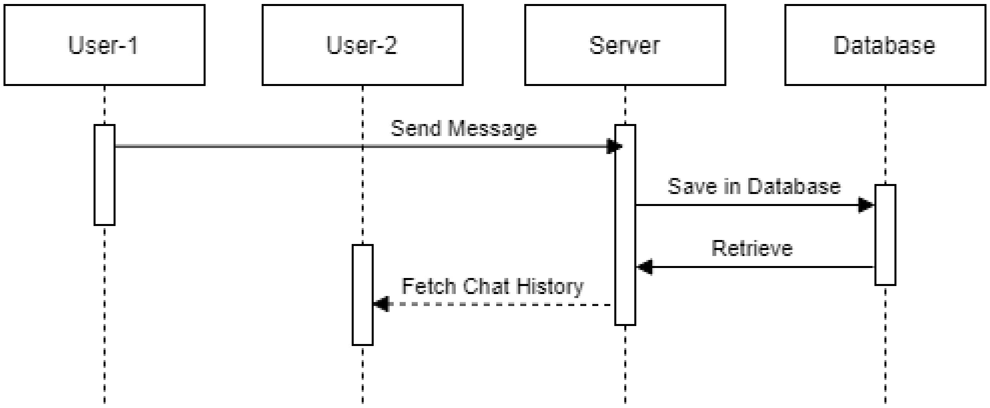


Figure 13 : Chat Sequence Diagram

4.2 Database Specification

4.2.1 ER Diagram

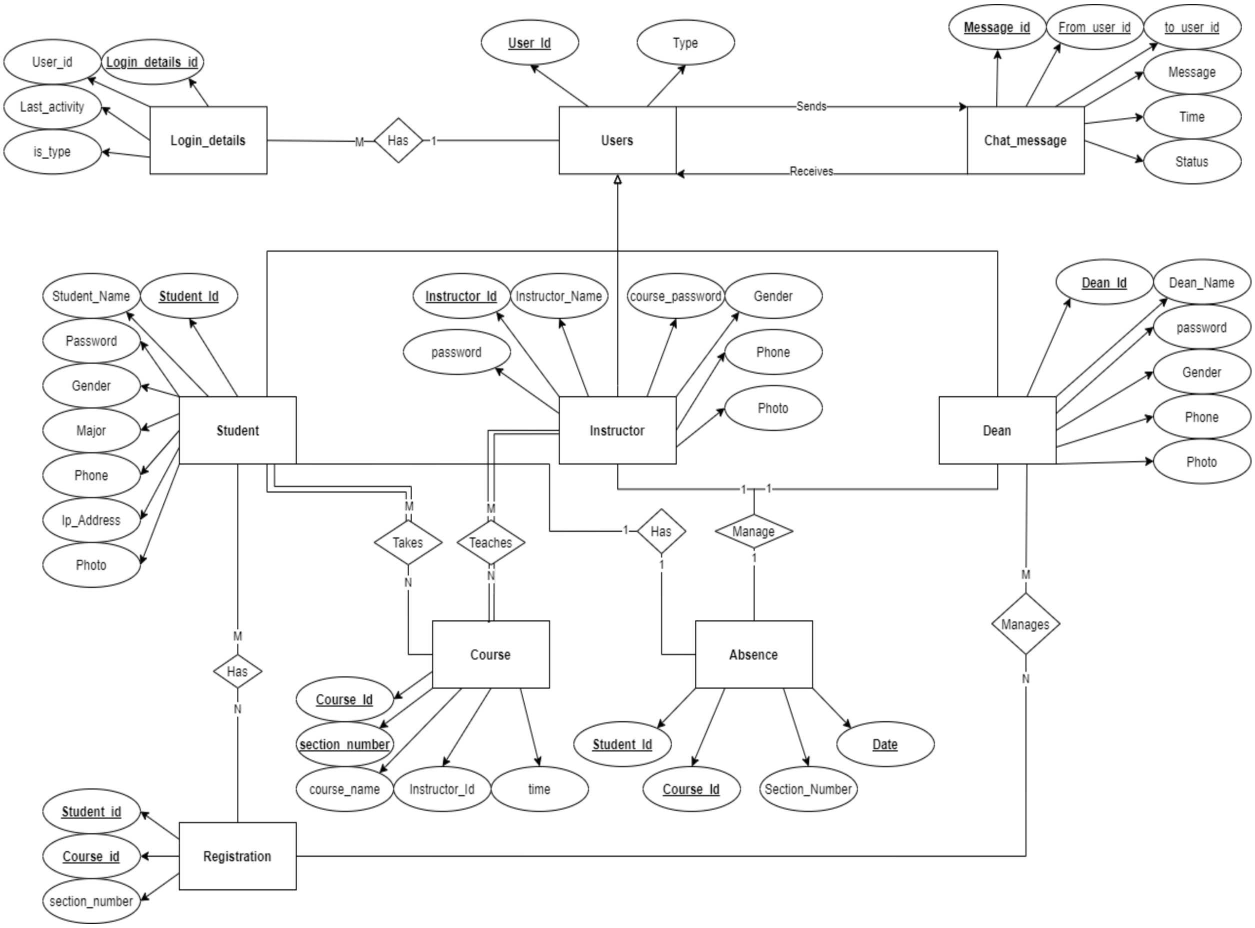
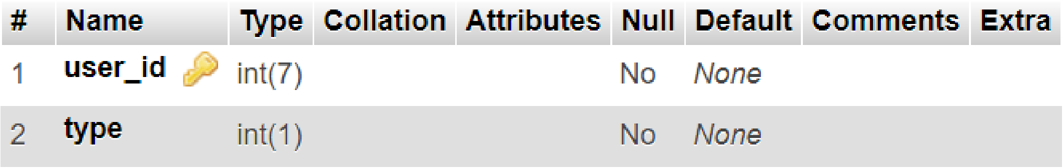


Figure 16 : ER Diagram

4.2.2 Database Tables Specification.

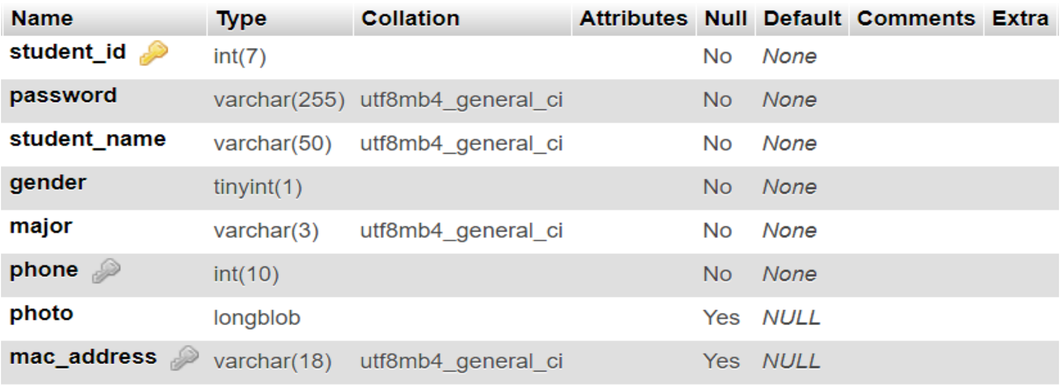
Our system contains 9 tables as shown down below:

1. Users
2. Student Table
3. Instructor Table
4. Dean Table
5. Course Table
6. Registration Table
7. Absence Table
8. Login_details
9. Chat_message



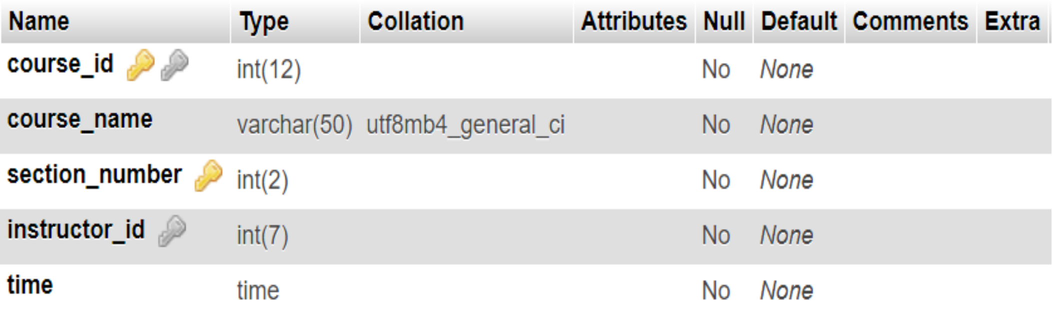
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	user_id 🔑	int(7)			No	*None*		
2	type	int(1)			No	*None*		

Figure 19 : Users Table



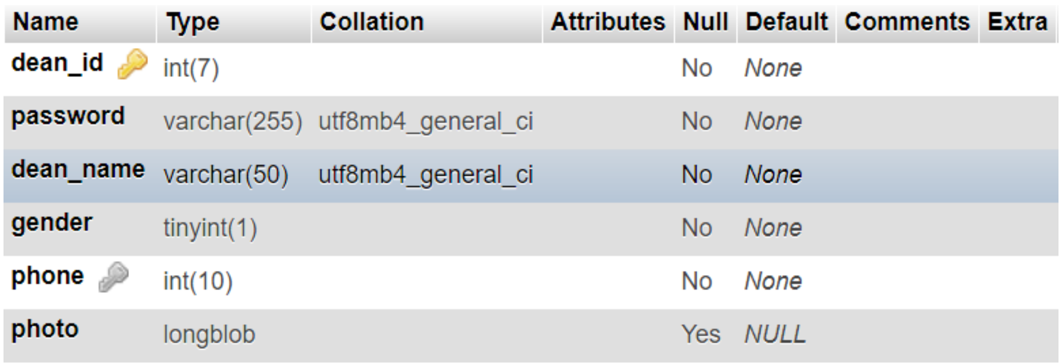
Name	Type	Collation	Attributes	Null	Default	Comments	Extra
student_id 🔑	int(7)			No	*None*		
password	varchar(255)	utf8mb4_general_ci		No	*None*		
student_name	varchar(50)	utf8mb4_general_ci		No	*None*		
gender	tinyint(1)			No	*None*		
major	varchar(3)	utf8mb4_general_ci		No	*None*		
phone 🔑	int(10)			No	*None*		
photo	longblob			Yes	*NULL*		
mac_address 🔑	varchar(18)	utf8mb4_general_ci		Yes	*NULL*		

Figure 22 : Student Table



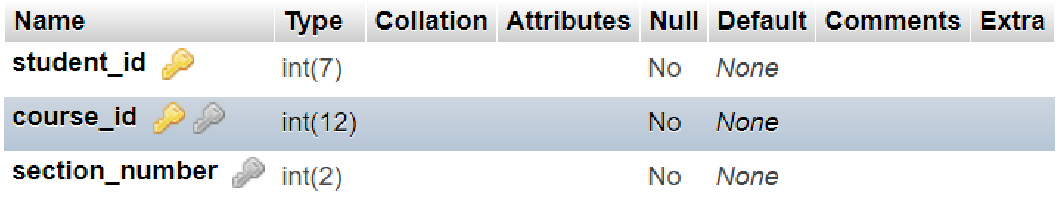
Name	Type	Collation	Attributes	Null	Default	Comments	Extra
course_id 🔑 🗝	int(12)			No	*None*		
course_name	varchar(50)	utf8mb4_general_ci		No	*None*		
section_number 🔑	int(2)			No	*None*		
instructor_id 🗝	int(7)			No	*None*		
time	time			No	*None*		

Figure 28 : Course Table



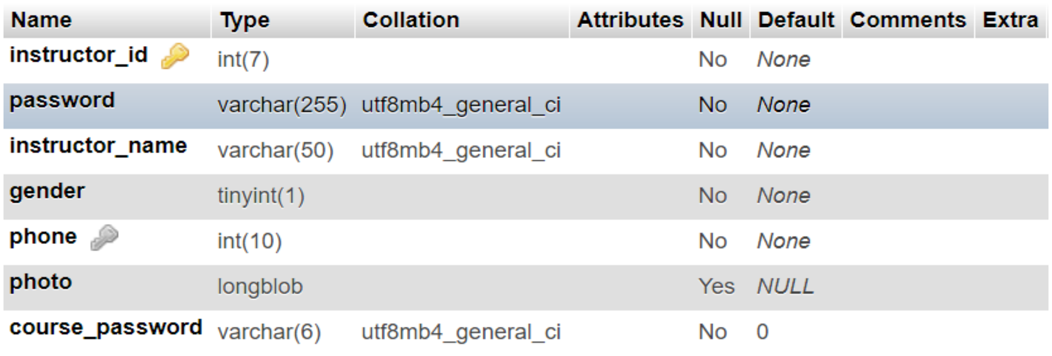
Name	Type	Collation	Attributes	Null	Default	Comments	Extra
dean_id 🔑	int(7)			No	*None*		
password	varchar(255)	utf8mb4_general_ci		No	*None*		
dean_name	varchar(50)	utf8mb4_general_ci		No	*None*		
gender	tinyint(1)			No	*None*		
phone 🗝	int(10)			No	*None*		
photo	longblob			Yes	*NULL*		

Figure 25 : Dean Table



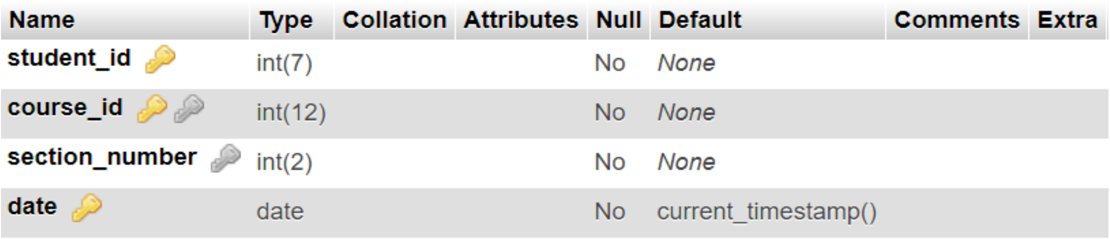
Name	Type	Collation	Attributes	Null	Default	Comments	Extra
student_id 🔑	int(7)			No	*None*		
course_id 🔑 🗝	int(12)			No	*None*		
section_number 🗝	int(2)			No	*None*		

Figure 31 : Registration Table



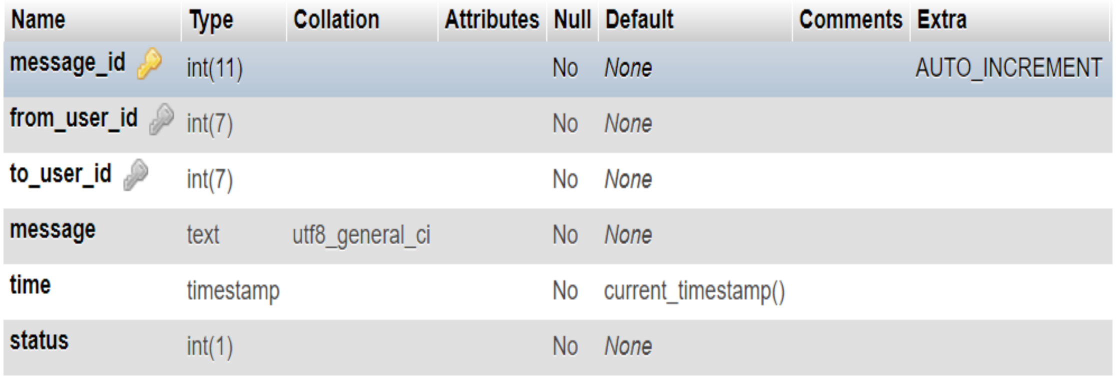
Name	Type	Collation	Attributes	Null	Default	Comments	Extra
instructor_id 🔑	int(7)			No	*None*		
password	varchar(255)	utf8mb4_general_ci		No	*None*		
instructor_name	varchar(50)	utf8mb4_general_ci		No	*None*		
gender	tinyint(1)			No	*None*		
phone 🗝	int(10)			No	*None*		
photo	longblob			Yes	*NULL*		
course_password	varchar(6)	utf8mb4_general_ci		No	0		

Figure 34 : Instructor Table



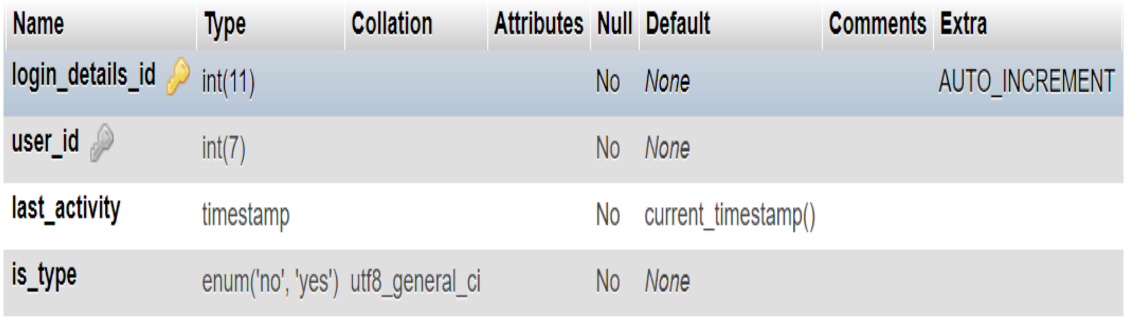
Name	Type	Collation	Attributes	Null	Default	Comments	Extra
student_id	int(7)			No	*None*		
course_id	int(12)			No	*None*		
section_number	int(2)			No	*None*		
date	date			No	current_timestamp()		

Figure 37 : Absence Table



Name	Type	Collation	Attributes	Null	Default	Comments	Extra
message_id	int(11)			No	*None*		AUTO_INCREMENT
from_user_id	int(7)			No	*None*		
to_user_id	int(7)			No	*None*		
message	text	utf8_general_ci		No	*None*		
time	timestamp			No	current_timestamp()		
status	int(1)			No	*None*		

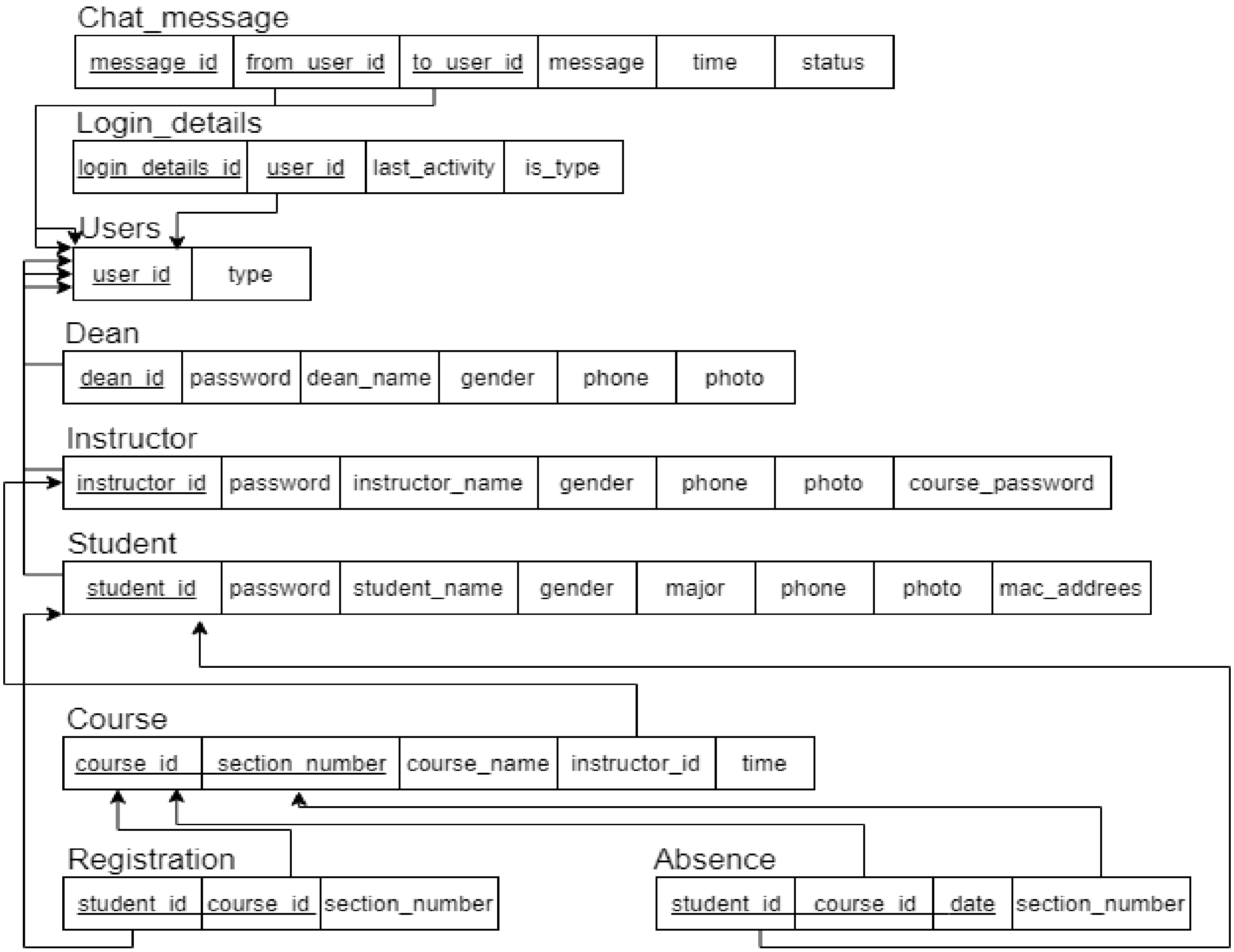
Figure 40 : Chat Message Table



Name	Type	Collation	Attributes	Null	Default	Comments	Extra
login_details_id	int(11)			No	*None*		AUTO_INCREMENT
user_id	int(7)			No	*None*		
last_activity	timestamp			No	current_timestamp()		
is_type	enum('no', 'yes')	utf8_general_ci		No	*None*		

Figure 43 : Login Details Table

4.2.3 Database Relational Schema.



Chat_message

message_id	from_user_id	to_user_id	message	time	status

Login_details

login_details_id	user_id	last_activity	is_type

Users

user_id	type

Dean

dean_id	password	dean_name	gender	phone	photo

Instructor

instructor_id	password	instructor_name	gender	phone	photo	course_password

Student

student_id	password	student_name	gender	major	phone	photo	mac_addrees

Course

course_id	section_number	course_name	instructor_id	time

Registration

student_id	course_id	section_number

Absence

student_id	course_id	date	section_number

Figure 46 : Database Relational Schema

4.2.4 Database Schema

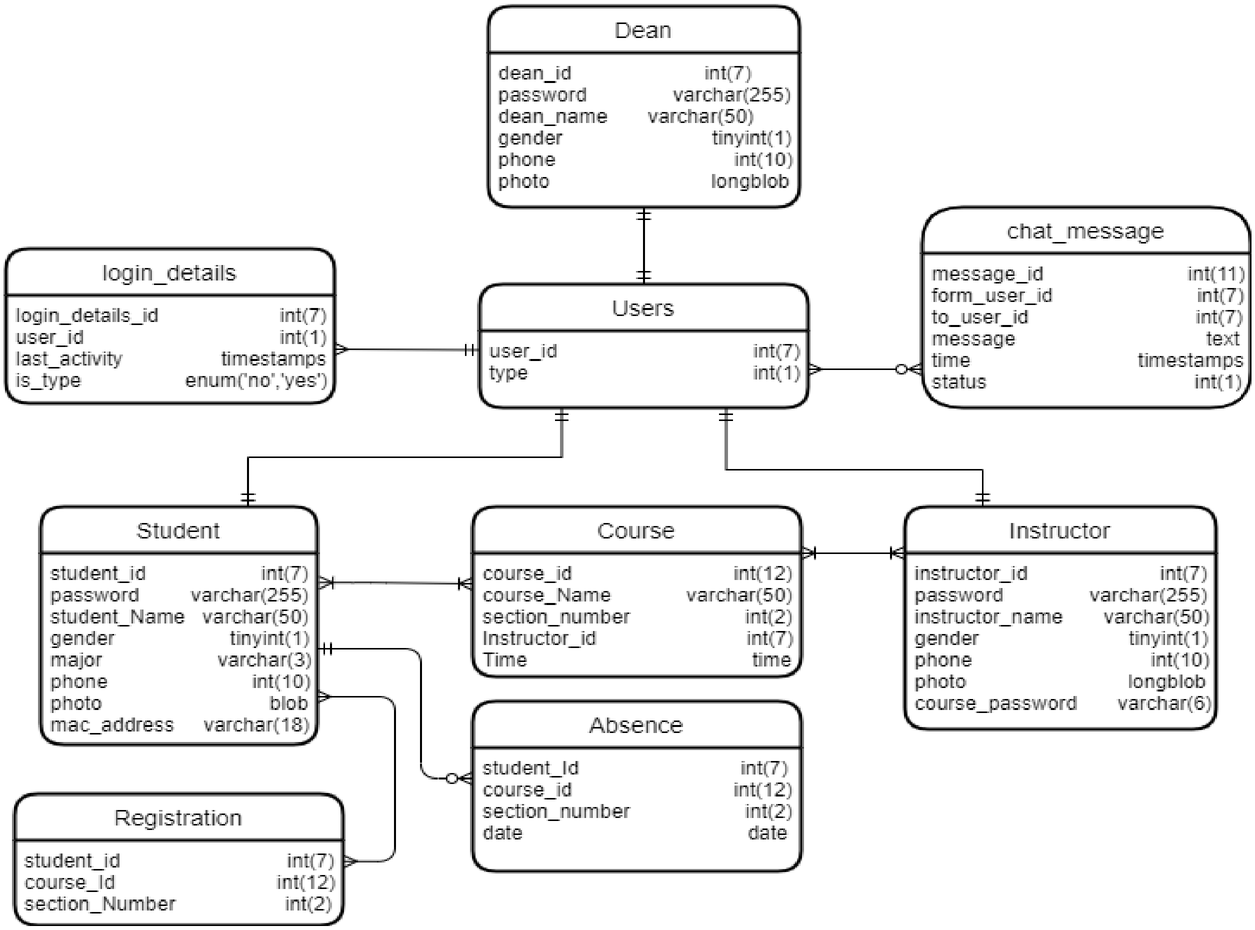


Figure 49 : Database Schema

4.3 Interface Specification

The UI will be user friendly, and the user will not have any difficulty interacting with it either browsing from their phone or laptop/desktop. The system will have a joint sign-in page for all users (Deans, Instructors, and Students).

4.3.1 Common components among all users.

This section as mentioned before definitely contains a sign-in page that includes (header, footer) and it is the first page that the user will face, whether a student, instructor or dean.

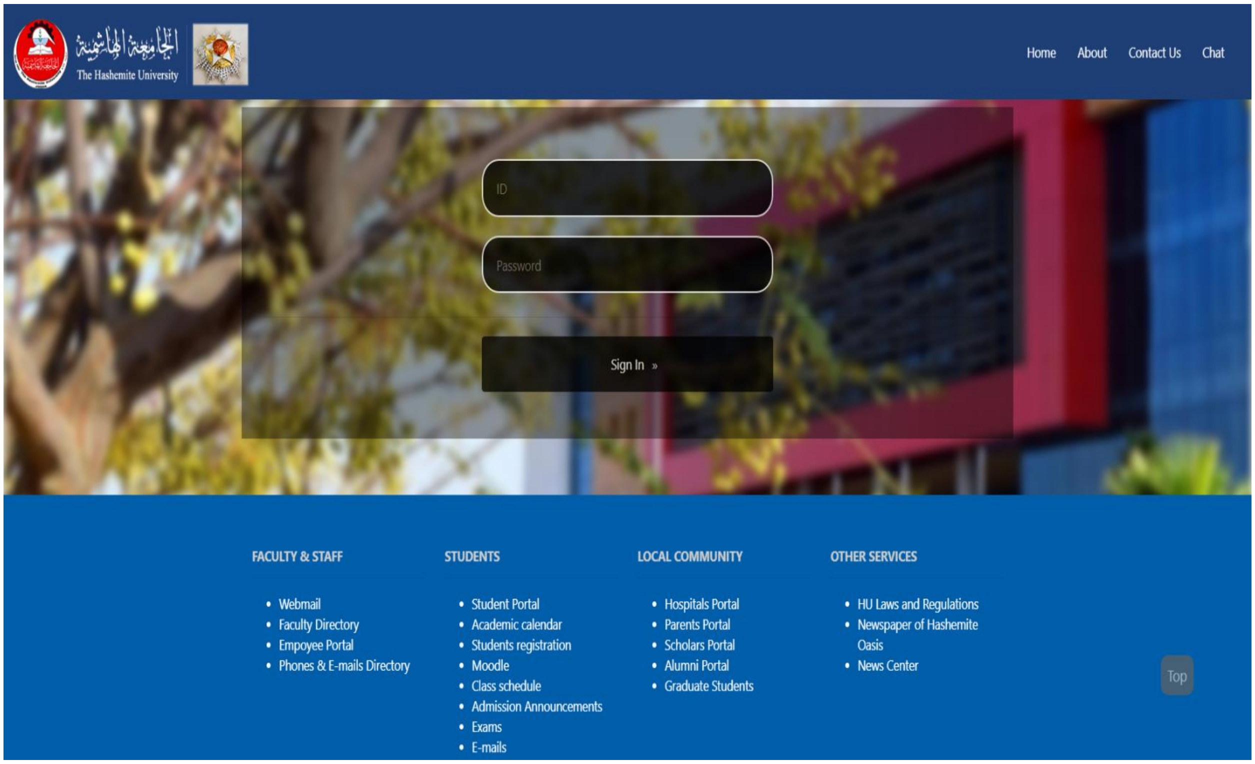


Figure 52 : Sign-in Page

1. As usual, the application header contains quick access buttons, In this situation…
    - ○ Home: Navigates the user to his respective home page.
    - ○ About: Information about the organization.
    - ○ Contact US: Give the user the possible ways to communicate with organization.
    - ○ Chat: Open the chat subsystem to communicate with other users.
2. The footer contains links to access various web pages of the HU Official Site.

The home pages for all users contains:

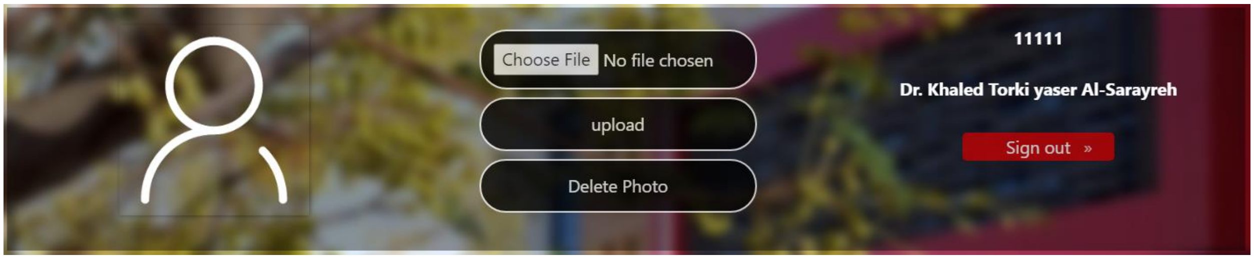
1. User Information.

   It contains his name, id and his specialization if he was a student.

2. Photo Settings

   The user can add or remove a personal photo from there.

3. Sign out Button



11111

Dr. Khaled Torki yaser Al-Sarayreh

Figure 55 : User Details

4.3.2 Dean Home Page.

As for the Dean's page, it contains many components that meet his needs and powers within the college like adding students, courses and instructors. Also, it has a separate section for the absence of students in any course in that semester.

1. Dean Forms.

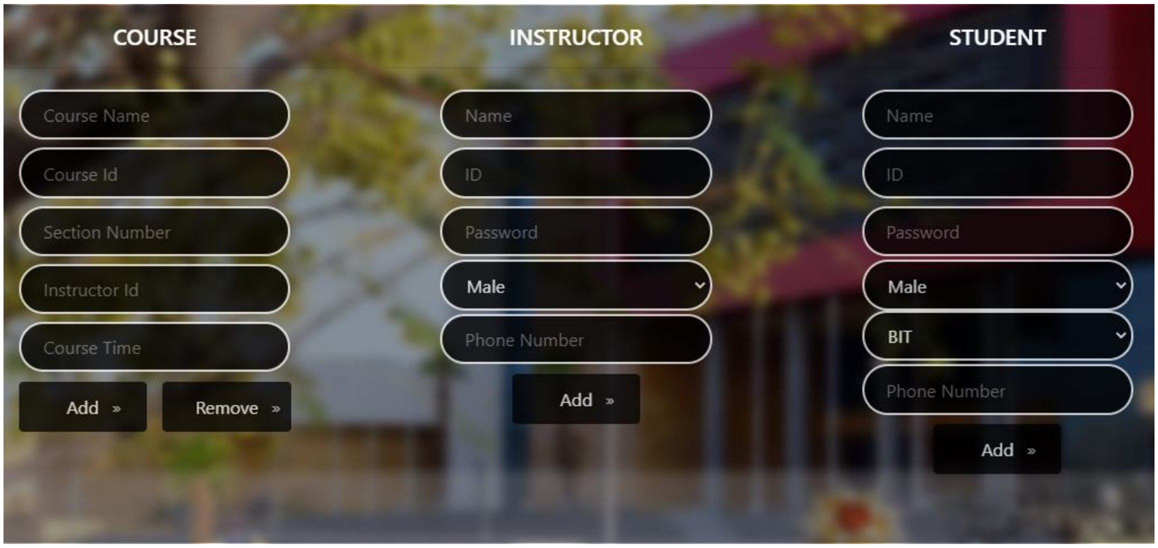
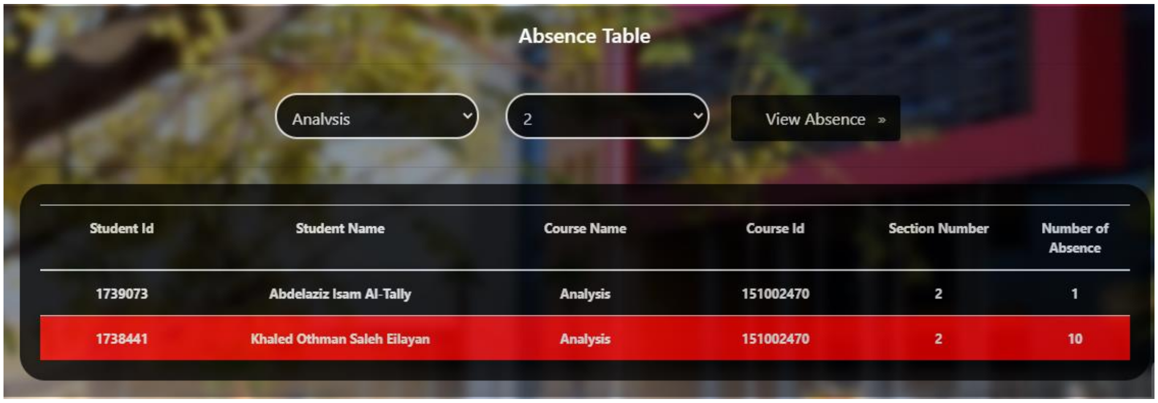


Figure 58 : Dean Forms

2. Absence Form: There is a feature that indicates to the dean which student exceeded the permitted absence percentage.



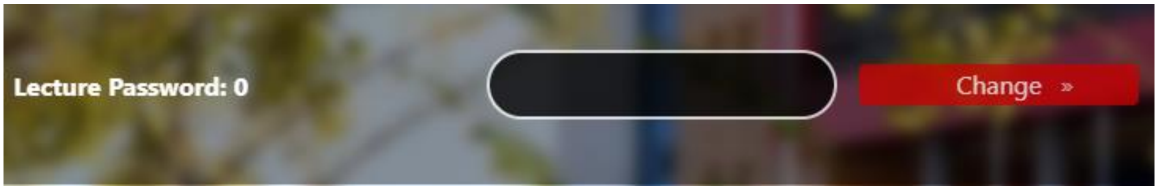
Student Id	Student Name	Course Name	Course Id	Section Number	Number of Absence
1739073	Abdelaziz Isam Al-Tally	Analysis	151002470	2	1
1738441	Khaled Othman Saleh Eilayan	Analysis	151002470	2	10

Figure 61 : Dean Absence Table

4.3.3 Instructors Home Page.

The instructors page contains three sections (Lecture Password, Registration Table, Absence Form of the students' in his lectures):

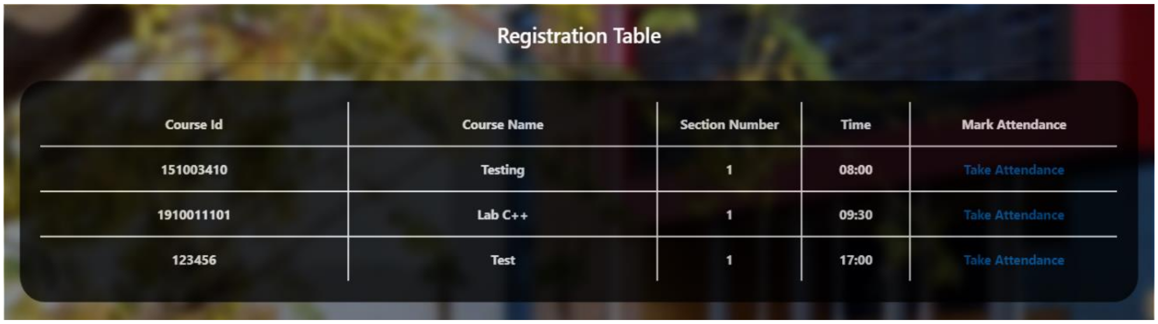
1. As we mentioned before the student will take attendance so the system will check lecture password modified by the course's instructor from this section.



Lecture Password: 0

Figure 64 : Lecture Password

2. Registration Section: this form showing the courses that instructor teaching in that semester and start taking attendance for current time lecture.

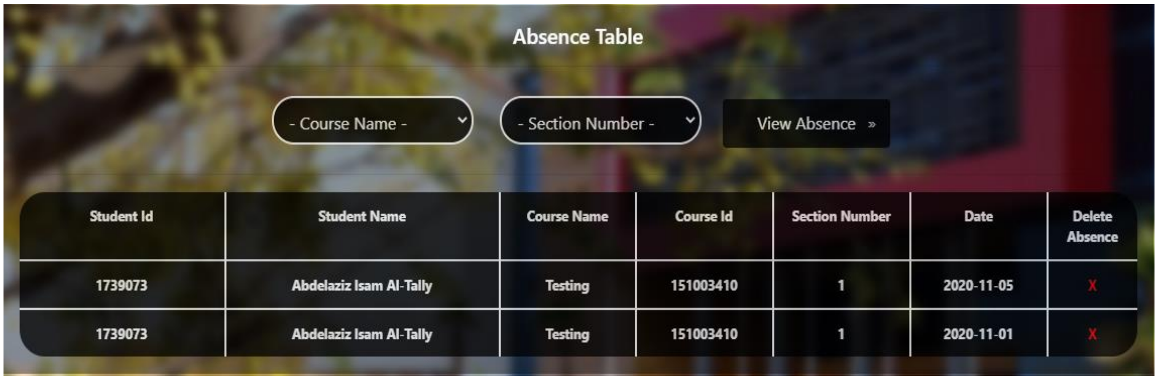


Registration Table

Course Id	Course Name	Section Number	Time	Mark Attendance
151003410	Testing	1	08:00	Take Attendance
1910011101	Lab C++	1	09:30	Take Attendance
123456	Test	1	17:00	Take Attendance

Figure 67 : Instructors Registration Table

3. Absence Table: this table show the student absence for his courses in details.



Absence Table

Student Id	Student Name	Course Name	Course Id	Section Number	Date	Delete Absence
1739073	Abdelaziz Isam Al-Tally	Testing	151003410	1	2020-11-05	X
1739073	Abdelaziz Isam Al-Tally	Testing	151003410	1	2020-11-01	X

Figure 70 : Instructors Absence Table

4.3.4 Students Home Page.

Students Home page also has three sections (Add courses, Registration, Absence)…

1. Add courses: from this section the student can add course to his registration table.

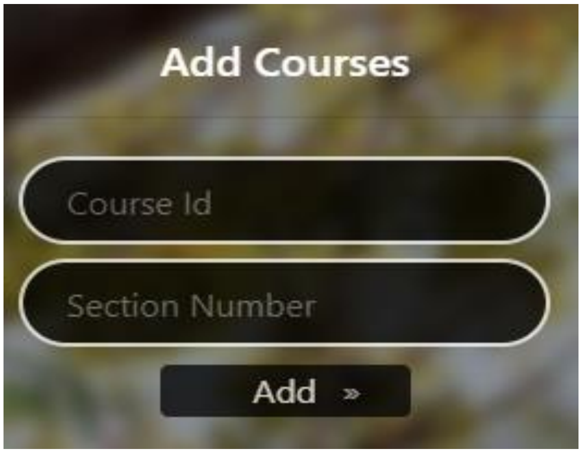
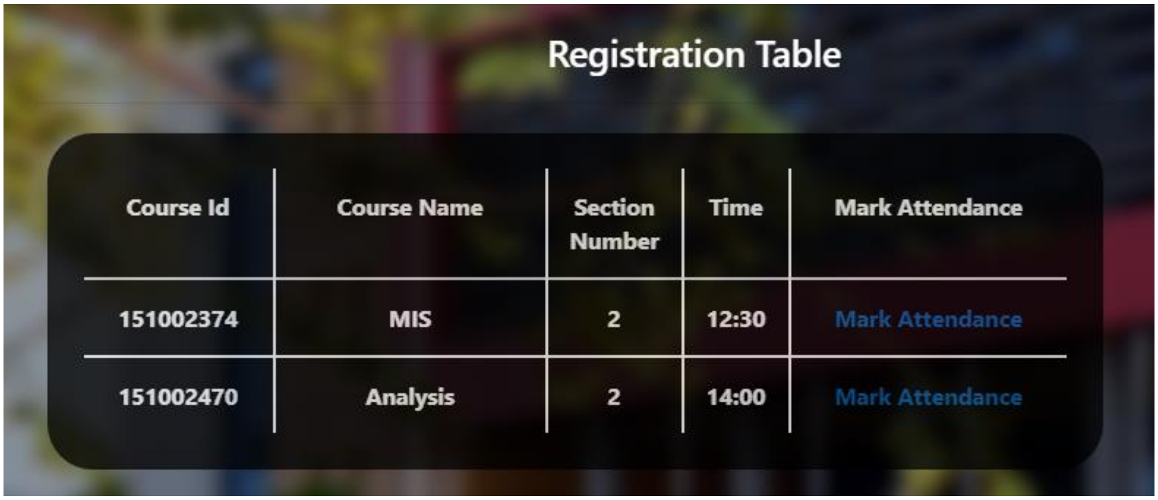


Figure 73 : Student Add Course

2. Registration Table: this section shows the courses he is taking in that semester and it contains mark attendance button to take attendance for current lecture.



Registration Table

Course Id	Course Name	Section Number	Time	Mark Attendance
151002374	MIS	2	12:30	Mark Attendance
151002470	Analysis	2	14:00	Mark Attendance

Figure 76 : Student Registration Table

After clicking on Mark Attendance this form will pop-up to enter that lecture password given by its instructor.

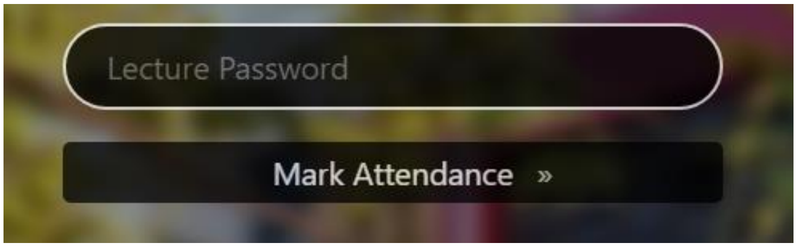
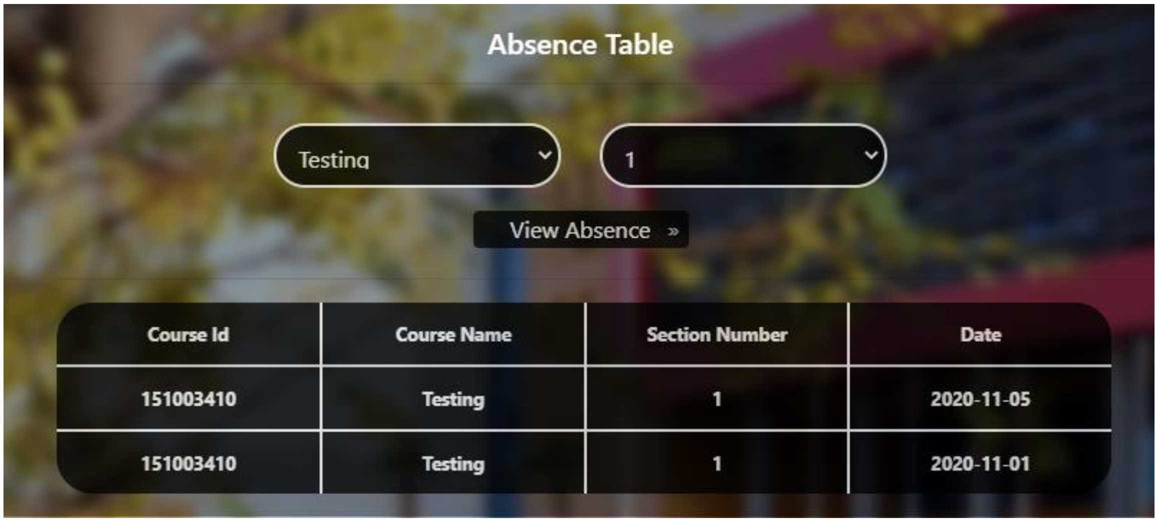


Figure 79 : Student Lecture Password

3. Absence Table: this form shows the detailed Absence for his lectures.



Absence Table

Testing

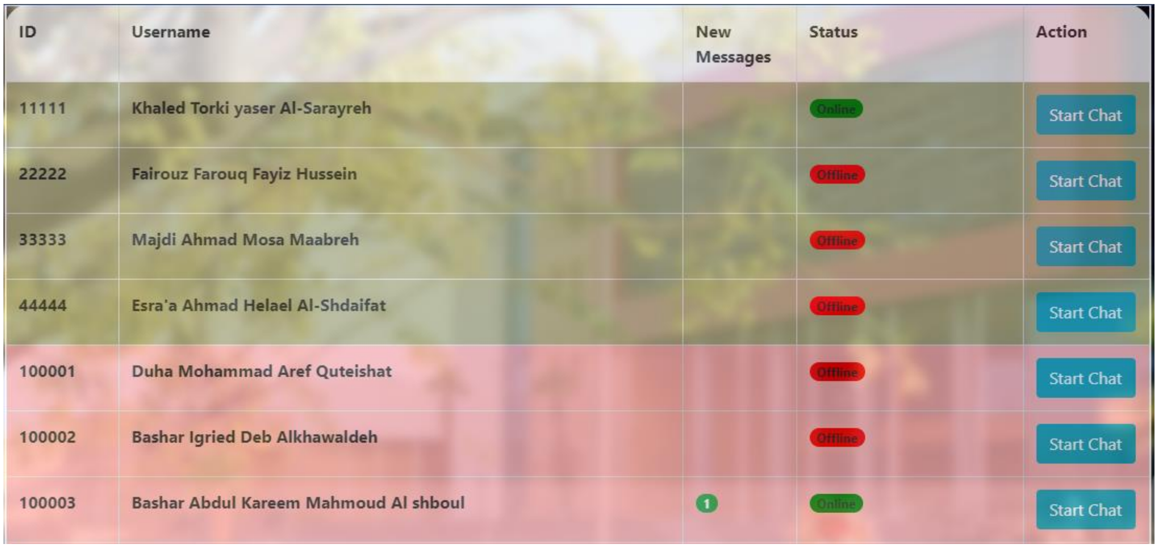
1

Course Id	Course Name	Section Number	Date
151003410	Testing	1	2020-11-05
151003410	Testing	1	2020-11-01

Figure 82 : Student Absence Table

4.3.5 Chat System.

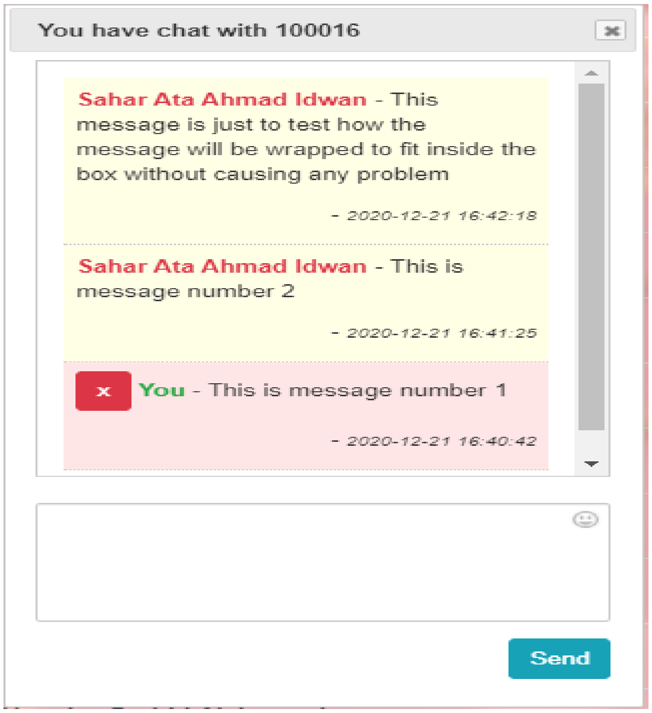
When the user clicks on chat button from the Navigation Bar, this table appears, user can start sending messages to any other user, see if he has received any messages and know online, offline users.



ID	Username	New Messages	Status	Action
11111	Khaled Torki yaser Al-Sarayreh		Online	Start Chat
22222	Fairouz Farouq Fayiz Hussein		Offline	Start Chat
33333	Majdi Ahmad Mosa Maabreh		Offline	Start Chat
44444	Esra'a Ahmad Helael Al-Shdaifat		Offline	Start Chat
100001	Duha Mohammad Aref Quteishat		Offline	Start Chat
100002	Bashar Igried Deb Alkhawaldeh		Offline	Start Chat
100003	Bashar Abdul Kareem Mahmoud Al shboul	1	Online	Start Chat

Figure 85 : Available users for Chat

After clicking on Start Chat button, the conversation will pop-up so you can see the chat history with the other user.



You have chat with 100016

Sahar Ata Ahmad Idwan - This message is just to test how the message will be wrapped to fit inside the box without causing any problem

- 2020-12-21 16:42:18

Sahar Ata Ahmad Idwan - This is message number 2

- 2020-12-21 16:41:25

x You - This is message number 1

- 2020-12-21 16:40:42

Figure 88 : User Conversation

4.4 Component Specification

Component Specification is a generic description of a component. It defines its dependencies to other components, what actions that can be performed on it and how those actions affect the dependencies. It also defines what artifacts the component can export to other components. In the component specifications diagram below, we tried to place some of the system components and to clarify the interaction between them.

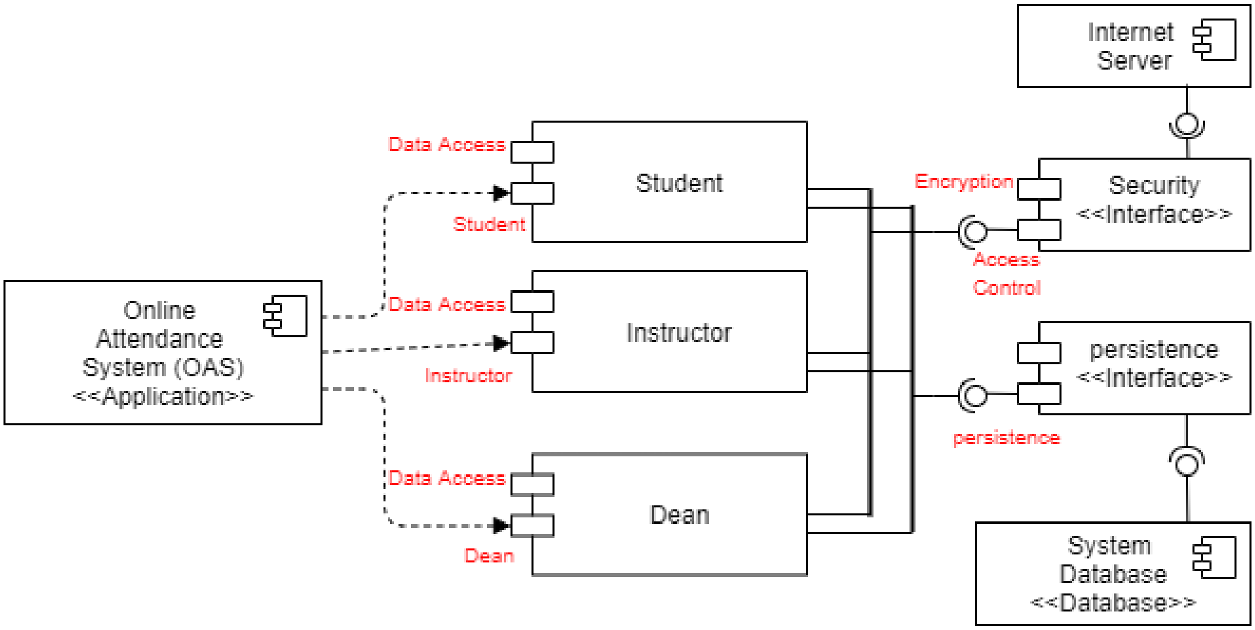


Figure 91 : OAS Component Specification

CHAPTER 5: SOFTWARE IMPLEMENTATION AND TESTING

A strategy for software testing must include low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

The test strategy occurs at a time when deadline pressure begins to rise, progress must be measurable, and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing.

5.1 Software Implementation

The software implementation stage involves the transformation of the software into one or more fabricated, integrated, and tested software configuration items that are ready for software acceptance testing. It refers to the process of adopting and integrating a software application into a business workflow. Implementation of new tools and software into an enterprise can be complex, depending on the size of the organization and the software.

Prior to implementation, the software should be selected by assessing needs, budget, potential benefits, obstacles, and so on. Once the solution is chosen, implementation can begin.

To avoid mistakes, the tests will be done by several steps, including:

1. Beta Software - Before installing at a large scale, test the software in advance with an in-house test kit that will be done by installing part of the system on universities for feedback.
2. Install - The IT department should work with the vendor to install the app across all target devices in the universities.
3. Preparation and Training - Once installed, develop a qualification program and software training program to ensure all users to take advantage of all functions and features.
4. Monitoring, Maintenance, and Follow-up - Monitor user feedback throughout the implementation process.

After the implementation of the program, we got some benefits for the system producer and the owner as below:

1. Cost-effective.
2. Improved reporting and planning.
3. Improved Efficiency.
4. Customer Service.
5. Improved Collaboration and Workflows.

5.2 Software Testing

Once the source code has been generated for a whole part of the system, it must be tested to uncover as many errors as possible in minimum amount of time and effort.

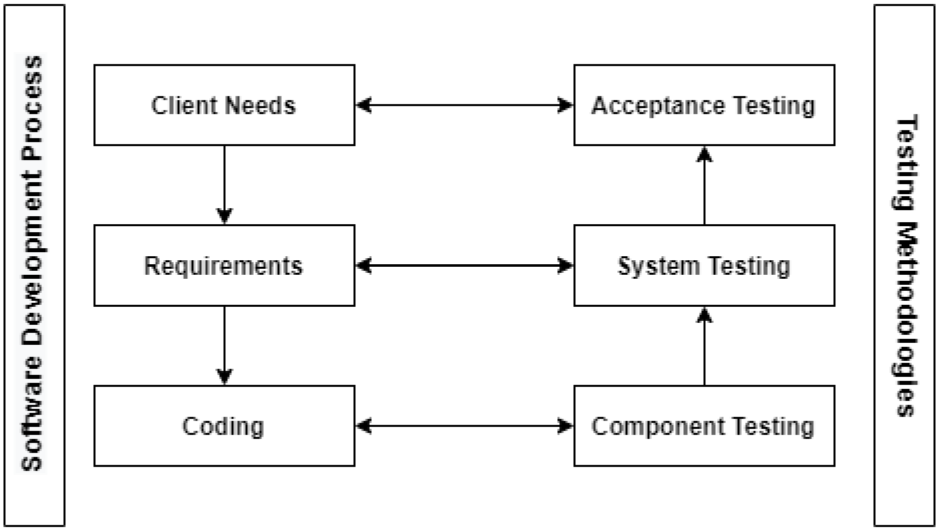


Figure 94 : Process of Software Development

Before we start the testing phase; we decided to go through with the Agile Testing methodology; so that whenever we implement a part of the code it will be tested immediately.

5.2.1 Component Testing

The step of focusing verification effort on the smallest unit of the software design. The component testing is a white-box testing approach. When looking back at figure we can see that User Acceptance Testing is utilized to identify and verify client needs.

The component testing for every module of the OAS and the Chat System is implemented by giving a correct manual input to the system, the input is stored in the database, and retrieved in future tests to verify that the system works perfectly.

Some of these components that we need to focus on while developing the system will be mentioned below along with their expected output/response.

ID	In Which Web Page	What to be Tested	Actual Response
1.	Sign-In Page	The Dean, Instructor, Student should be able to sign into their respective portals	The system directs each user to his respective portal.
2.	Instructor/ Student Database Table	To verify that the data is stored in the database after appointing a new Instructor or adding a new Student	All data entered by the user was processed and stored in the database
3.	Portal	View Attendance and absence: **Dean** → All Students in the faculty **Instructor** → All students who attend his courses **Students** → His own attendance in all courses	The dean can view the number of absences for each student. The instructor can view each absence for each student with its respective date. The student can view his absence in each of the courses he's enrolled in.
4.	Student's Portal	The Student should be able to Mark his attendance in a course he's attending only during a specific time.	The student can mark his attendance given that he has the lecture's password provided by the instructor.
5.	Chat System	• The Instructor/Dean should be able to communicate with all members. • The Student should be able to communicate with every Dean/Instructor in the faculty.	All messages were guaranteed to reach its desired user without any conflict.

Table 4 : Test Cases for Component Testing

5.2.2 System Testing

System testing is a series of different tests whose primary purpose is to fully exercise the computer-based system.

Our goal is to design a series of test cases that have a good capability of finding errors. Below we have chosen the best that helps us in our system under the circumstances such as:

1. **Usability Testing:** Mainly focuses on the user's ease of use of the application, flexibility in handling controls and ability of the system to meet its objectives
2. **Load Testing:** Is necessary to know that a software will perform well under real-life loads.
3. **Regression Testing:** Involves testing done to make sure that none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.
4. **Recovery Testing:** Done to demonstrate if a software solution is reliable, trustworthy and can successfully recoup from possible crashes.
5. **Migration testing:** Is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.
6. Testing every input in the application to check desired output.
7. Testing of the user’s experience with the system.

5.2.3 Acceptance Testing

Acceptance testing is no less important than any of the Unit or System testing, and for many reasons one might say that it is the most crucial stage of the testing phase.

The acceptance testing is a complicated stage in the software development, because it is done at an awkward time - at the end of the system development - which means that the QA team will start testing the system after the development team is done with it, that’s why we will be using the agile testing approach which means that the testing will be done in the same time the development team is done working on a whole part/unit of the system and the Acceptance Testing will be done on the newly implemented part/unit.

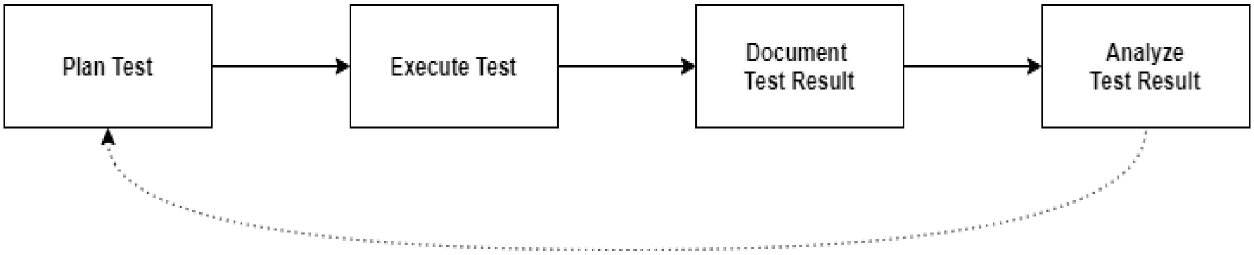


Figure 97 : User Acceptance Testing

CHAPTER 6: CONCLUSION

Online Attendance System (OAS) and the Chat System, a two web-based applications developed with HTML, CSS, Bootstrap, jQuery, AJAX and JavaScript as Front-End languages, PHP and JSON as a Back-End management language, Apache Server as a web server, and MySQL as a Database Management System to fit perfectly with the objectives to be achieved.

The OAS is very helpful in saving valuable time in universities' lectures, it can record attendance of students in a very convenient manner using their devices to prevent impersonation and reduce level of absence. Also, it reduces most human errors and helps to update and maintain attendance records.

The system was developed with professionalism and high accuracy, and accordingly the system proved its effectiveness in helping educational institutions in the process of registering student attendance, and this is what we aspired to from the beginning.

The Chat System will save time for all members in the university which undoubtedly easier and faster than emails, and at the same time more formal and professional than using already existed chat systems (WhatsApp, Messenger) which is developed to be personal and private.

6.1 Future Work

New additions will be implemented for the chat system, for example creating a group chat for each lecture and section will come handy for instructors and students also, such as posting news about upcoming quizzes, exams, deadlines.

A group chat for instructors may help them receive new rules and regulations issued by the faculty or the university itself.

This system contains a wide range of future developments and improvements that will achieve a quantitative leap in the attendance registration process in all educational institutions. In the near future, more capabilities will be added to the instructor through which he can impose penalties on students as a result of exceeding the number of absences allowed during the semester.

Based on our future aspirations, we can make the attendance unit more dynamic by using the facial recognition feature. The iris identification technique will also be used. Through these technologies, we will be able to transform the attendance registration process into a fully automated process without human intervention at all.

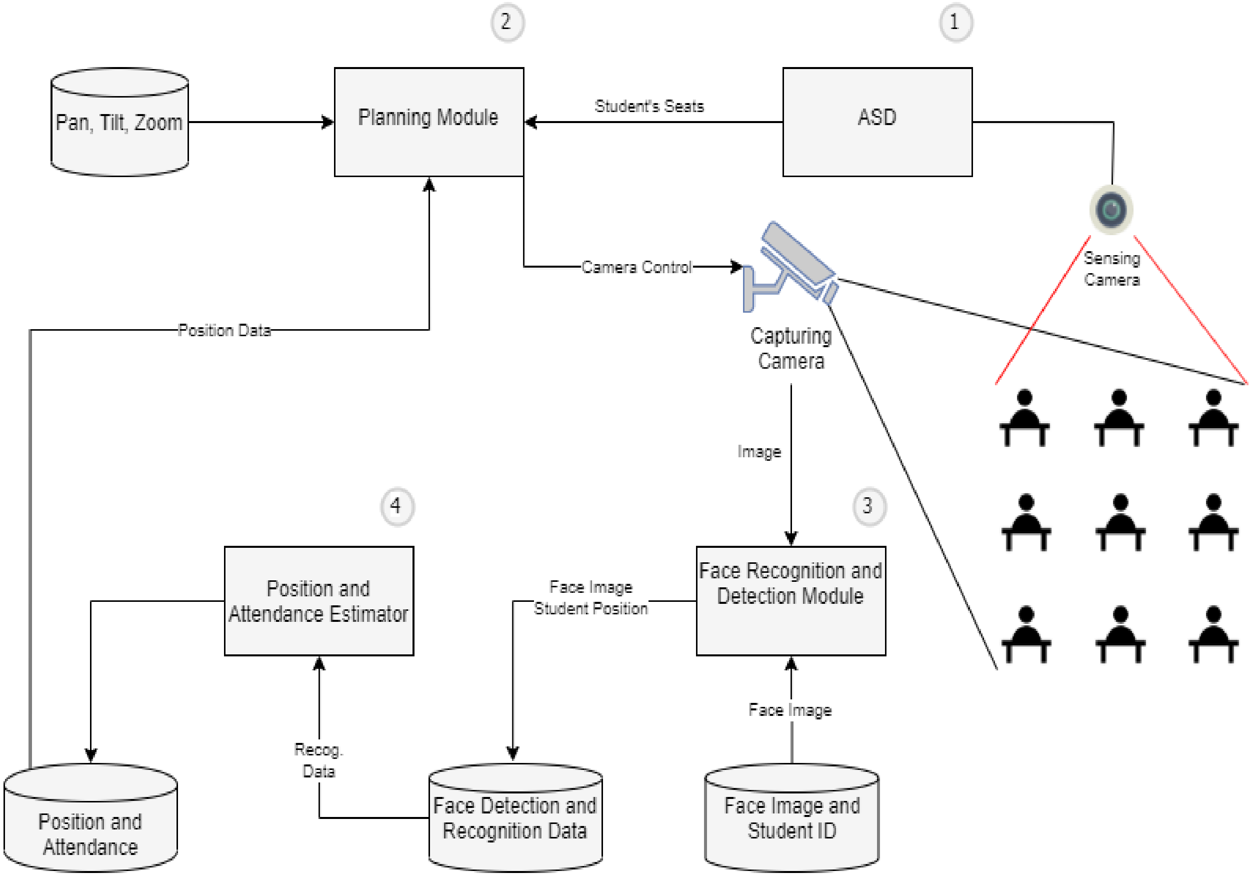


Figure 100 : Future Work Architecture

REFERENCES

a. Books

Hoffer, J. H, and George, J. F, Valacich, J. S (2014). Modern Systems Analysis and Design (7th Edition).

Paul C. Jorgensen. (2013). Software testing: a craftsman's approach (4th Edition). CRC Press.

Ramez Elmasri and Shamkant Navathe (2015). Fundamentals of Database Systems (7th edition). London, UK: Pearson Education.

b. Websites

Wales, Jimmy. "Software Design." Wikipedia. January 15, 2001.

https://en.wikipedia.org/wiki/Software design.

Mothasim M. tutorialspoint. Jan 1, 2006.

https://www.tutorialspoint.com/index.htm.

Refsnes Data. "SQL Tutorial" w3schools. 1998.

https://www.w3schools.com/sql/default.asp.