



Online Payment and Attendance System

Database Project Report
Group-05

Report submitted December 26, 2021

A project submitted to Dr. Rudra Pratap Deb Nath, Associate Professor, Department of Computer Science and Engineering, Chittagong University (CU) in partial fulfillment of the requirements for the Database Systems Lab course. The project is not submitted to any other organization at the same time.

Table 1: Details of Group-05

Roll / ID	Name	Sigature	Date	Supervisor Approval
19701070	Md. Masud Mazumder	Masud	31-12-21	
19701066	Tonmoy Chandro Das	Tonmoy	31-12-21	
19701068	Tareq Rahman Likhon Khan	Tareq	31-12-21	
19701040	Hamza Mohtadee	Hamza	31-12-21	
19701019	Palash Hossen	Palash	31-12-21	
18701045	Nu Shai Mong Marma	Mong	31-12-21	

Contents

1	Introduction	5
1.1	Background and Motivation	6
1.2	Problem Statement	6
1.3	System Definition	7
1.4	System Development Process	7
1.5	Organization	10
2	Project Management	11
3	Requirement Gathering and analysis	16
3.1	Requirement Gathering	16
3.1.1	Functional Requirements:	16
3.1.2	Non-Functional Requirements:	17
3.2	Requirement Analysis	18
4	Conceptual Modelling	22
5	Logical Modelling	23
6	Normalization	24
7	Physical Modelling	33
8	System Architecture	34
9	Implementation	36
9.1	Frontend	36
9.2	Database	42
9.3	Backend	44
10	Validation	46
11	Software Deployment	48
12	Conclusion and Future Work	50

List of Figures

1	Software Development Life Cycle (SDLC)	7
2	Architectural design of CU-OPAS	9

3	Project Management Agile Scrum Board Template	13
4	Context Diagram of CU-OPAS	18
5	Flowchart of CU-OPAS	19
6	Entity Relationship Diagram of CU-OPAS	20
7	Conceptual Data Model of CU-OPAS	22
8	Logical Data Model of CU-OPAS	23
9	Physical Data Model of CU-OPAS	33
10	Work flow of the system	35
11	Github repository of CU-OPAS	36
12	Home Page User Interface	37
13	Student Attendance User Interface	37
14	Payment History User Interface	38
15	Payment Slip	39
16	Available Payment Methods	40
17	Payment Procedure	41
18	Teacher Attendance Interface	42
19	User Update Form	42
20	Backend API implementation (part-1)	45
21	Backend API implementation (part-2)	45

List of Tables

1	Details of Group-05	1
---	-------------------------------	---

Listings

1	A SQL query to find a User	42
2	A SQL query to find a Student data	43
3	A SQL query to insert payment purpose	44
4	A SQL query to find course data	44

Abstract

The University of Chittagong's online payment and attendance system is the focus of this project report, which is primarily concerned with that system. A large number of students at the university pay all of the university fees using bank drafts to the institution's accounts at a particular bank branch that does not allow for the use of internet payment methods. This analog payment mechanism is inefficient, particularly during examination seasons, when the majority of students are required to pay examination fees. It is marked by lengthy lines, excessive waiting on the part of students, and congestion at the banks where payments are made throughout this procedure. Alternatively, manually recording attendance continues to be practiced here, and it is quite time-consuming to do so. It was against this backdrop that we began work on the project to establish an alternate payment and attendance system that would allow students and their sponsors to pay and show up for class online. This method ensures that all students are acquainted with the online payment processes. Using this technology, Additionally, taking attendance online will save a significant amount of time. The system was developed using a JavaScript-based framework termed "React," which included Cascading Style Sheets for the frontend and "ExpressJs" for the backend, as well as an Apache web server and a MySQL database server, among other technologies. Testing and validation of the system were also carried out by enabling users of the system to engage with it while interacting with test data. For the time being, the system is solely capable of handling the payment and attendance systems. This system has the potential to be developed in the future to include numerous online systems such as the No Objection Certificate (NOC), the Student Management System, the Employee Management System, and so on. The project's outcome is an online payment and attendance system for the University of Chittagong, which alleviates the long-standing challenges associated with the university's present ways of payment as well as the time-consuming manual attendance method.

1 Introduction

One of the most significant advantages of employing technologies is that, in most circumstances, it is less expensive in terms of time and instantaneous time in most of the cases to do the same task than it would be with an analog process. Conventionally, we continue to do all of the official activities via the use of paper documents. In addition to increasing time consumption, it also makes students more reliant on authority and the particular time limits impose by the authorities. Furthermore, there is always the possibility of being unsuccessful in finishing the procedure, whether or not the information is incorrect. In addition, manually recording attendance is a time-consuming process that takes a long time. Our objective is to create an application that will transfer these two procedures to a digital platform, which will, of course, be able to complete the duties in the shortest amount of time imaginable and with the least amount of paper work possible. The system should be able to provide students with the independence from the tormenting limits of the administration while also allowing the authorities to make their responsibilities more manageable for themselves. For both students and administrators, it goes without saying that the program would be user-friendly in its design.

This document is a record of a strategic and creative process that was focused on clearly describing concerns and objectives, as well as providing an overview of the application that represented the narrative from the beginning to the completion of the process. Anyone interested in using or developing the system would be able to do so with the assistance of this documentation.

The purpose of this course is to develop a database application system by applying the theories, methodologies, tools, and technologies we learnt in CSE 413 that is Database System course.

1.1 Background and Motivation

Every year since the university's founding in 1966, the number of students has increased by a small but steady margin. Despite the fact that the globe is being exposed to new technology on a daily basis, the method for receiving fees and the attendance system at the University of Chittagong have stayed the same. This is due to the fact that it is reliant on a single branch of a single bank, making the payment system for tuition difficult to manage not just for students but also for the administration. Students despise this analog approach, especially since they are restricted to a one-day time frame to deposit the money during class days or in the weeks leading up to the test. As the number of students continues to rise at a good pace year after year, it seems that university administrators, as well as bank administrators, have had enough of this terrible procedure. The same may be said regarding the old method of attendance. Teachers have been seen to devote 25 to 30 percentile of a class period to attesting attendance in the traditional manner.

If we take a closer look, we will see that these issues are interconnected. We want to create a system that can tackle these difficulties while also processing some basic data from both students and instructors. To accomplish the functions, the system will not need any paper work and will demand the least amount of physical presence from authorities as well as students. Throughout the development of the application, both the payment and attendance systems will become one-click operations. The capability for administrations to query the necessary information will be more accessible than ever before. Students and administrators may use this program to run processes from any location at any time, with an immediate time stamp to validate it.

1.2 Problem Statement

To develop a database system that can be used to handle payment system of fees of student and attendance system online.

The system will be using student data such as *name*, *student id*, some institutional data such as *course*, *department*, some teacher's data such as *teacher's name*, *the department he belongs to* and *which courses he teaches* and so on. The system will also hold the records of payments already done or have to be done by a student. With the information of courses from a teacher and the students studying the course, the attendance system will be implemented in the same application.

1.3 System Definition

“An online application that is used to manage the payment system for fees by students as well as the attendance system for such students. Transforming and shifting the analog payment system to a digital platform should be accomplished via the application, avoiding the time-consuming approach.”

1.4 System Development Process

A process known as the software development life cycle is divided into various parts (SDLC). The requirements collection and analysis step is the most critical element of the whole process. It is at this phase that the client explains his or her expectations for the project, including who will use the product, how the customer will use the product, and the specific data that will be included with any unusual client needs that have been recognized. Project managers often use a variety of approaches to gather the requirements for their projects. Interviews, surveys, observations, and workshops are some of the most well-known types of research methods. We employed interview and questionnaire methodologies to gather the requirements because we were determined to do so.

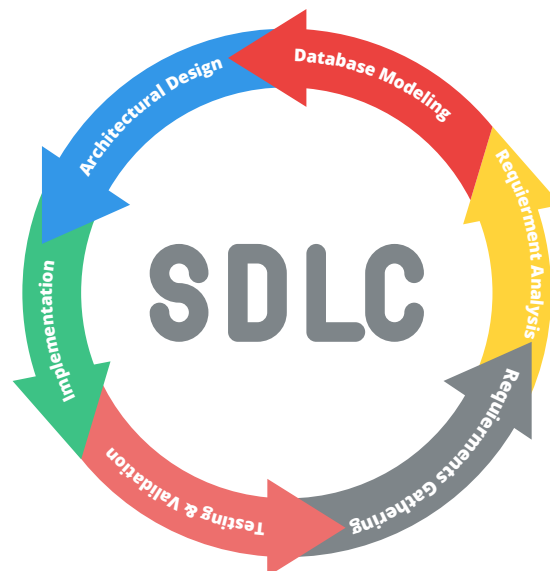


Figure 1: Software Development Life Cycle (SDLC)

The research of requirements is vital, as is the basic movement that occurs once the requirements are gathered. We disassemble, improve, and study the

needs that have been gathered in order to generate predictable and clear requirements. As part of this effort, all requirements are audited and a graphical view on the whole framework is provided. Afterwards, when the assessment has been completed, it is common for the task's understandability to significantly increase in terms of comprehension. In this case, we may also make use of the client's relationship to describe areas of disarray and determine which needs are of more importance than others.

When it comes to database modeling, it's also referred to as data modeling since it involves establishing a data model for the data that will be kept in the database. The third phase of the SDLC is known as the design phase. This data model is a conceptual representation of data items, the connections that exist between them, and the rules that govern them. The Data Model is defined as a theoretical model that organizes information representation, information semantics, and consistency limits of the information in a way that is understandable to the user. The information model emphasizes what information is necessary and how it should be organized, rather than what actions will be done on it, as opposed to the tasks themselves..

There are basically three types of data models. These are conceptual data models, logical data models, and physical data models, each with a specific purpose.

- The conceptual data model mainly defines what the system contains. This model is commonly created by business stakeholders and Data Architects. The intention is to put together, scope, and characterize business ideas and rules.
- The logical data model defines how the system should be implemented independently of the DBMS. This model is regularly made by Data Architects and Business Analysts. The object is to foster a specialized guide of rules and information structures.
- The physical data model depicts how the system gonna be implemented using a particular DBMS. This model is commonly made by DBA and engineers. The intention is the actual implementation of the database.

A system architecture is a representation of the structure of a system that is intended to be used as an illustration. It encompasses all of the system's components, as well as subsystems that handle all of the system's operations in their entirety. CU-OPAS is a web-based application system that uses

request-response patterns to communicate with users. Any legitimate user of this system may request inquiries about payment and attendance activities from the backend API by using the frontend user interface on the system's front end. After that, the backend API communicates with the database server and creates a response in response to the queries that are provided back to the frontend application.

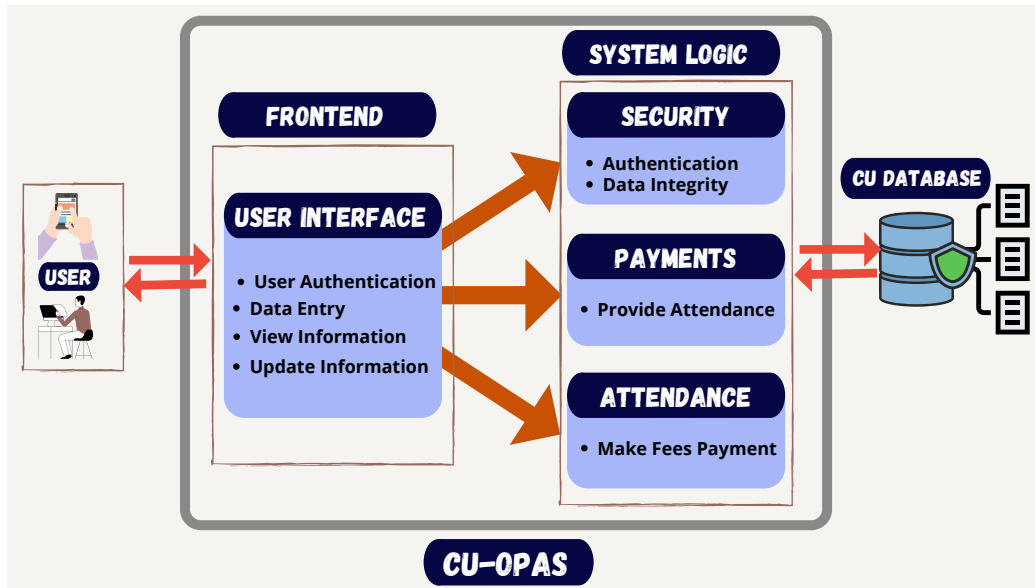


Figure 2: Architectural design of CU-OPAS

The implementation process starts when the architectural design and user testing have been completed successfully. It is at this phase when the physical design of the system is completed. It is the third phase of the SDLC. As part of this phase, we employed a range of programming languages, frameworks, tools, and online resources to construct our system from scratch.

The final and most crucial phase of the SDLC is validation. This step is critical in ensuring that not just the proper product, but also a high-quality product is produced. Validation can determine whether or not the system meets end-user requirements. We've tested our system in a variety of ways for this aim, including Unit Testing, Integration Testing, System Testing, and Acceptance Testing.

1.5 Organization

Section 1 gives an overview of this project. this section also narrates the project from start to the end briefly. Section 2 describes how the project and the resources are managed. The next section 3 refers to the results of the analysis of the information gathered from the surveys, interviews and discussion with some group of students, teachers and administrative officers. The following section 4, section 5 and section 6 gives the overview of how we designed the database and enhanced it step by step as most as possible. Section 8 and section 9 gives the information about the whole system structure and how we implemented it. Section 10 says how we validated the system with real user data with an statistics on consumed time, cost, user satisfaction between the previous system and this system. Section 11 is about the process to install and configure the system so that even a non-technical person can use the system easily. Finally, the conclusion and the pointers to the future work are outlined in Section 12.

2 Project Management

We chose a web-based platform to develop the system since it would be simpler to use with a variety of devices in general. To utilize the program, all that will be required is a web browser and access to the internet, and the user will be able to access it from anywhere at any time. Several components will be required for the program, including a frontend, which will communicate with its user through a browser, as well as a backend, which will provide certain functionality as well as do some database activities in response to a query that has been chosen by the user.

If we attempted to finish this application on our own, it would take a significant amount of time. Thus, we had originally intended to divide the whole project into many branches, each of which would include little or medium-sized assignments, and then divide those duties throughout our full team of collaborators. Following the end of the project, we planned to combine all of the branches from our teammates and merge them together in order to successfully develop the final system. Although we were working on the project, it is evident that we were using the Scrum¹ method to keep things moving ahead. In order to complete our project in accordance with the scrum model's specifications, we followed the five processes or phases listed below.

1. Product Backlog Creation

In this process, we transformed the significance and functional details of the system into short stories

As an example, consider a standard student whose semester final test is scheduled for a few days from now. He/she will be required to pay the semester fee within a certain time frame determined by the administration. Students suffer as a result of this, and their preparedness is negatively impacted. Furthermore, while using the traditional method of transaction, a university is completely reliant on a single bank branch, which creates a tiring and agonizing scenario when a large number of students seek to pay their tuition fees on the same day as the university. Several students even fail to submit their fees on time since they are unable to conduct any financial activities during non-working hours. Teachers are also said to have lost around one-third of the course session while manually collecting attendance, which has an adverse impact on the focus of pupils. On the other aspect, management often complains about how time-consuming it is to manually update student data in papers. It not only takes a long time to finish, but it also needs the actual presence of a huge number of individuals in order to

be successful.

These considerations inspired our team's decision to design a single application using the information obtained from our database course that is capable of resolving all of the challenges discussed above.

2. Sprint planning and creating backlog

It was then time for the establishment of the sprint backlog, for which our team had to choose the most essential user stories and split them down into smaller tasks. Contingency plan was developed for how we would execute the work. In addition, our team prioritized the activities that needed to be completed in order to complete the project more efficiently. The sprint lasted around 9 to 10 days, which was sufficient time for the developers to complete their task completely.

3. Working on sprint

This was a phase of practical application. The real stories were reorganized as discrete tasks in the sprint backlog, which is where the actual work began. In order to get started, a task board, also known as a Kanban board in Trello², was created with a large number of cards in use. Using the cards, you may describe the specifics of the tasks, such as who will be assigned to the job, what will be done, when it will be completed, and so on. Additionally, we utilized these cards to keep track of our meeting schedule and information, as well as any future revisions. In addition to the "To Do" lists, the task board also had columns for "Work In Progress," "Testing," and "Work Done," as well as a column for "Testing." A typical task board is shown in the diagram below.

In the scrum approach, a project is often completed by combining the efforts of a large group of people who are all overseen by a single individual. In our situation, *Md Masud Majumdar* was the team's leader from the beginning to the conclusion, and he planned and executed the whole project. Each team member has made significant contributions to various aspects of the overall project.

We held regular virtual meetings where all the teammates together, came up with all the various ideas to design the database schema. The meetings we held virtually through Google Meet³ and Zoom software were led by *Tareq Rahman Khan*. Within a week, we decided to draw

the first E-R Diagram on our ideas which was fianlly done by *Palash Hossen*. Later it was modified by *Nu Sai Mong Marma*.

While developing the system, we used Github⁴ to make the collaboration among teammates and to integrate all the assigned task and make the final system. Github not only allowed us to collaborate with our teammates but also to control the modifications of the application. Since the start of the project, our team made about 100 of commits into Github repository. All the commits are from *Tonmoy Chandro Das* and *Md. Masud Majumdar* as frontend and backend functionalities were developed by them. To interact with github, we used git in our local machine and Visual Studio Code as code editor.

While developing the system, we also looked after the documentation of this entire project. This documentation was prepared under the leadership of *Hamza Mohtadee Nafi*.

The Scrum Board



Figure 3: Project Management Agile Scrum Board Template

4. Testing and Product Demonstration

Our system's stack holders have been reviewed, and thus this phase is mainly the modification phase depending on the results of the study. The activities done were to be turned into a workable product that would be subjected to full life cycle testing. Every sprint that was done was displayed to the stack holders, who included students, teachers, and other relevant authorities, in order to get their approval and opinion on the overall solution. We got positive feedback from the stack holders, and in the majority of instances, they stated their satisfaction with the solution that we provided. In response to our request for recommendations, a portion of them provided us with some suggestions for improvements to numerous aspects of the application. We definitely implemented a significant number of them and integrated them into the final system.

5. Retrospective and the next sprint planning

After this phase was completed, it was necessary to examine what had gone well and what might be improved for the following level. We also spoke about the lessons gained and the hazards of any specific challenges or problems that came up throughout the course of the project. We discussed our next sprint, which will consist of future development. We want to include additional features into the system over time, and eventually transform it into a dependable and fully working university administration system. Our team is sure that they will be able to do this with the present information and more research. As a consequence, there will be no need for paper in administration. All of the formal duties will take less time than they have in the past. In the interim, we will undoubtedly maintain the system in order to ensure that it continues to function as an uninterrupted service provider application.

Each and every member of our team has made an equal contribution to the project, from the beginning of the planning phase to the completion of the system's final implementation. We attempted to create a better system than the present one, and after hundreds of proposals were discussed and rejected by all of us, we were eventually successful. In order to accomplish this project, we attempted to make the best possible use of existing technology. During the course of the project, we discovered a great deal of new information while being stuck on challenges we had never encountered before. Projects might be conducted carelessly if adequate project management principles are not followed, putting them at a much greater risk of failure, delay, and going over budget. Knowing the foundations of project management increased our

chances of effectively completing a project.

¹Scrum is an agile development methodology used in the development of Software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project.

²Trello is a web-based, Kanban-style, list-making application and is developed by Trello Enterprise, a subsidiary of Atlassian. It is a visual collaboration tool that enables one to organize and prioritize projects in a fun, flexible, and rewarding way. A Trello board is a series of lists, with a bunch of cards attached and packed full with powerful features and automation. [website: www.trello.com]

³Google Meet is a video-communication service developed by Google. It is one of two apps that constitute the replacement for Google Hangouts, the other being Google Chat. [website:www.meet.google.com]

⁴GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features. It lets us and others work together on projects from anywhere. [website: www.github.com]

3 Requirement Gathering and analysis

The collecting and analysis of requirements is the first phase in the SDLC process. The methodologies employed in this stage included interviews, questionnaires, and observational data collection to obtain requirements. We have divided all of the needs into two categories: functional requirements and non-functional requirements. Functional requirements are those that are necessary for the application to operate.

3.1 Requirement Gathering

3.1.1 Functional Requirements:

To collect functional requirements, we meet some teachers from our university and interviewed them. The following are some of their requirements:

- Less time-consuming attendance system
- Attendance statistics in detail should be easy to calculate (From teachers and students ends)
- Attendance system must be transparent
- Students records should be printable
- Less paperwork and one-click procedure

Then we meet some administrative officers to meet their requirements and interviewed them. Their requirements include the following things:

- Students must be able to enter transaction data into a user interface accepts transaction data.
- Students will be able to make fee payments online.
- Students should be able to receive feedback on the online payment.
- If the fee payment transaction is successful, students can view, print, or save the payment receipt.
- Financial officers will be permitted to lead look on the data of individuals online payment information.
- The finance officer will be able to view statistics of all payments made through the system.

- Financial officials will provide login information so that everyone can safely use the system.
- Finance officials will be able to see fee payments in an editable manner.

Secondly, we conducted a survey among all the students of the University of Chittagong to meet their requirements. According to the survey we have come out with some requirements such as:

- Less time-consuming attendance and payment system
- Being free from working hours
- Get rid of the tiresome process of queueing
- Expanding the ways of transaction
- Being independent of a specific bank branch
- Being able to make transactions from anywhere

3.1.2 Non-Functional Requirements:

According to the interviews of administrative officers teachers, a survey conducted on students of the University of Chittagong, our observation we find out some non-functional requirements which are as followings:

- The system must be easy to operate.
- User interface should be simple and understandable for both experienced and inexperienced users.
- The system must be secured enough.
- Speed of the system should be as fast as possible.
- It should be cross-platform compatible.
- Reports should be provided in a variety of formats, such as tables and graphs, for simple management visualization.
- A standard graphical user interface that facilitates online data entry, modification, update, and deletion.

3.2 Requirement Analysis

Following the collection of requirements, the next stage is to examine them. We have accomplished this stage in accordance with Structured Analysis Methodologies, which focuses on the functional decomposition and data-flow analysis, among other things. Following our team's brainstorming session, we came up with a context diagram that everyone can understand.

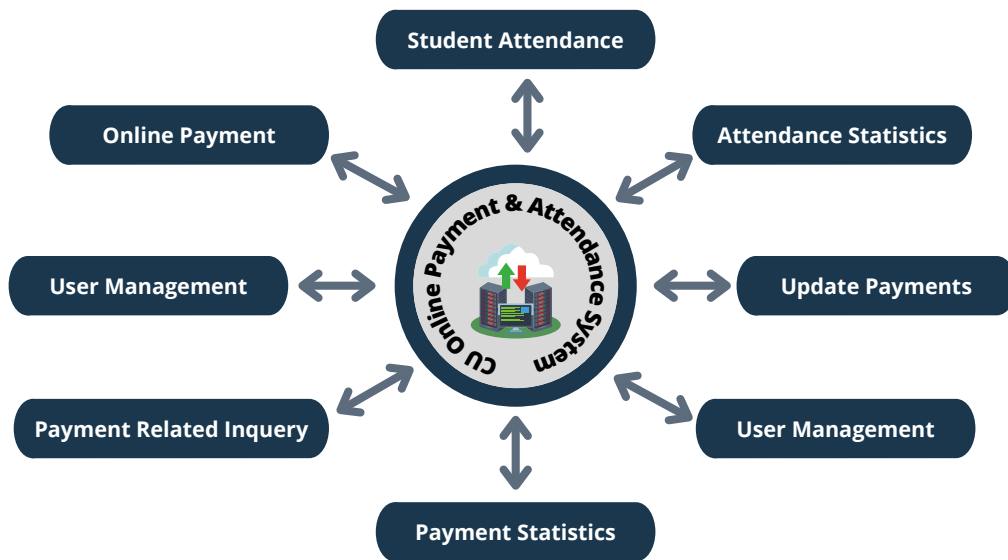


Figure 4: Context Diagram of CU-OPAS

We brought it up in front of the appropriate authorities as well as our supervisor. Then it was slightly tweaked to make it more user-friendly. Finally, we completed a model of the system that meets all of the needs of administrative authorities, instructors, and students, and it is ready for implementation. A flowchart was then built in order to define the sequence of actions.

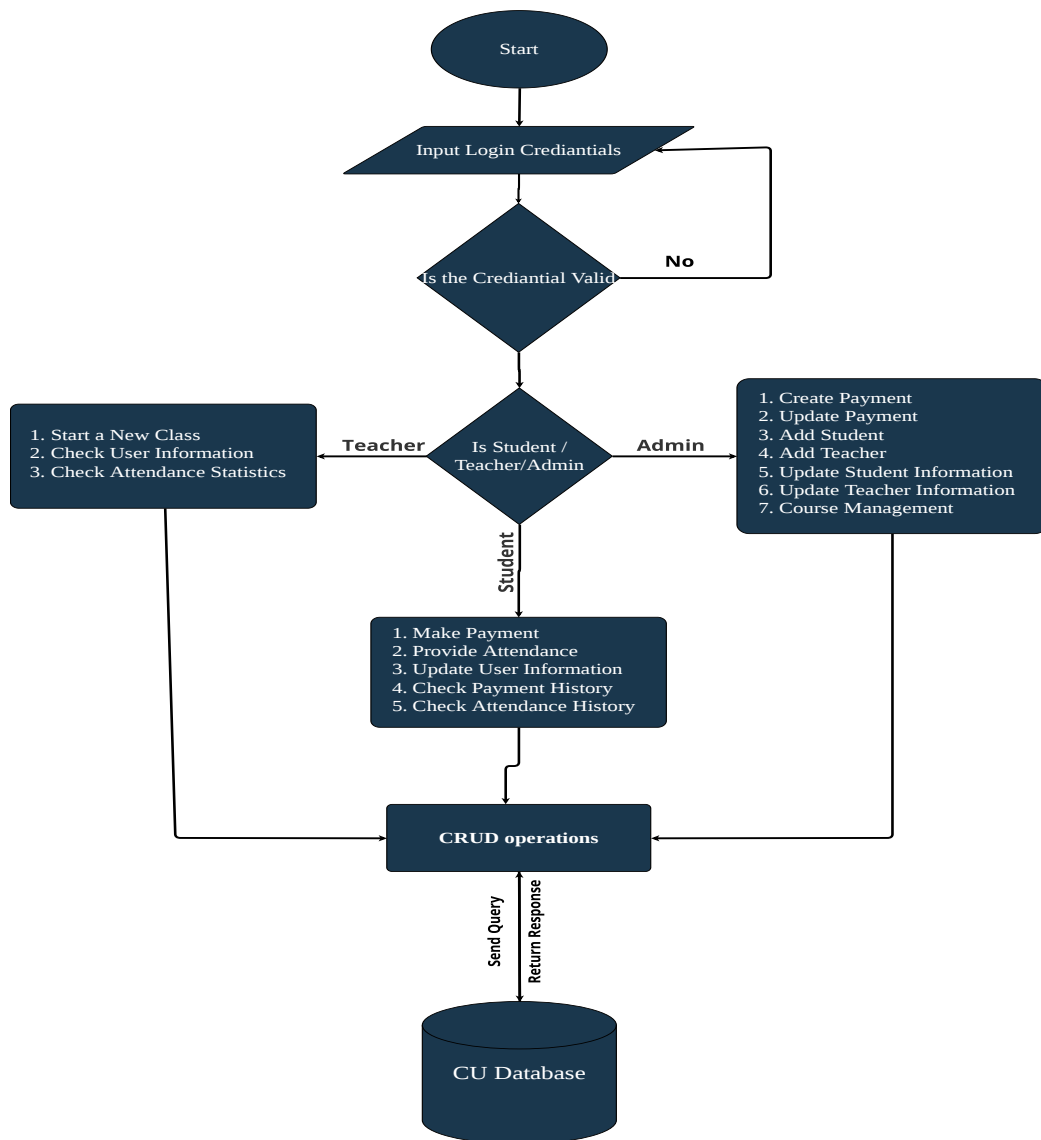


Figure 5: Flowchart of CU-OPAS

Finally, we have a completed entity relationship diagram that is ready for implementation.

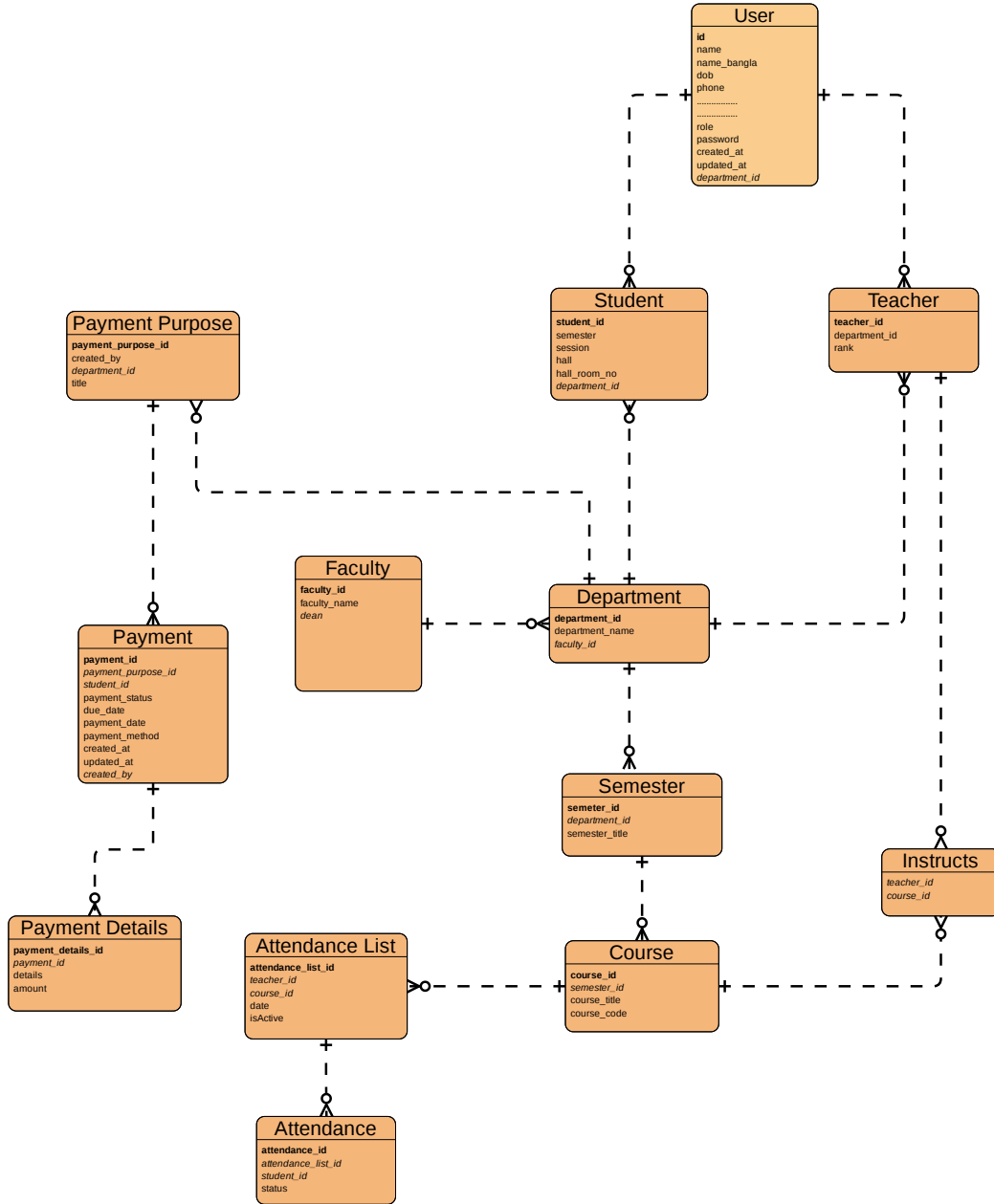


Figure 6: Entity Relationship Diagram of CU-OPAS

Stakeholders are the individuals that are involved in the project. There is apprehension regarding the project's result among them. Participants may also include a group, a firm, customers, suppliers, and other parties. Stakeholders have an effect on the project, either directly or indirectly. According to this classification, there are two sorts of stakeholders: primary and secondary.

In this project, the key stakeholders are supervisors, team members, teachers, and administrative authorities, while the secondary stakeholders include students, the news media, and other teams, including their members.

4 Conceptual Modelling

In software development, conceptual data modeling is a concept that represents the relationships between different entities in a database. This data model is generated as a result of the requirement analysis and is the most basic type of data modeling. Therefore, this paradigm, which is known as extremely abstract, is easy to understand by any technician or non-technical person. In this data model, the entity's attributes may or may not be present. Business stakeholders and data architects are frequently involved in the creation of this data model. We analyzed our project requirements and developed a very basic data model (Conceptual Data Model), which is shown below.

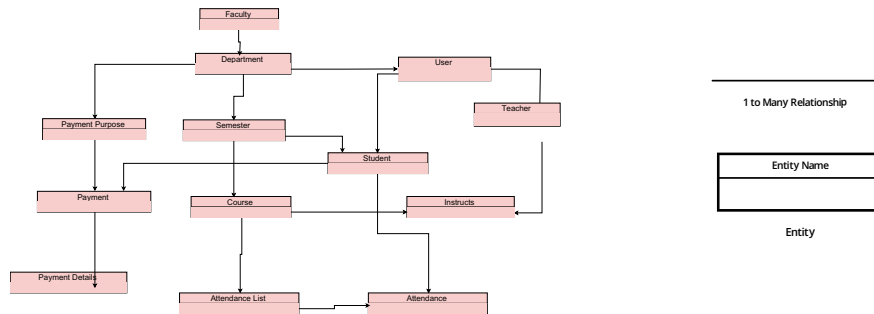


Figure 7: Conceptual Data Model of CU-OPAS

Conceptually model your database using an E-R diagram. Use the legends in your diagram. Write how you find the entity types, relationships, and attributes from Section 3

5 Logical Modelling

The logical data model (LDM) is the conceptual data model's enlarged format. It describes how to set up a system that is not particular to any database. It primarily establishes the data elements and the relationships between them. A logical data model is the foundation of the physical data model (FDM). Attributes, primary keys, foreign keys, relationship cardinality, and descriptive entities and classes are all described in an LDM. This data model clearly defines all of the relationships between the entities. As a result, anyone may convert an LDM to an FDM in any database management system. This data model is often created by data architects and business analysts.

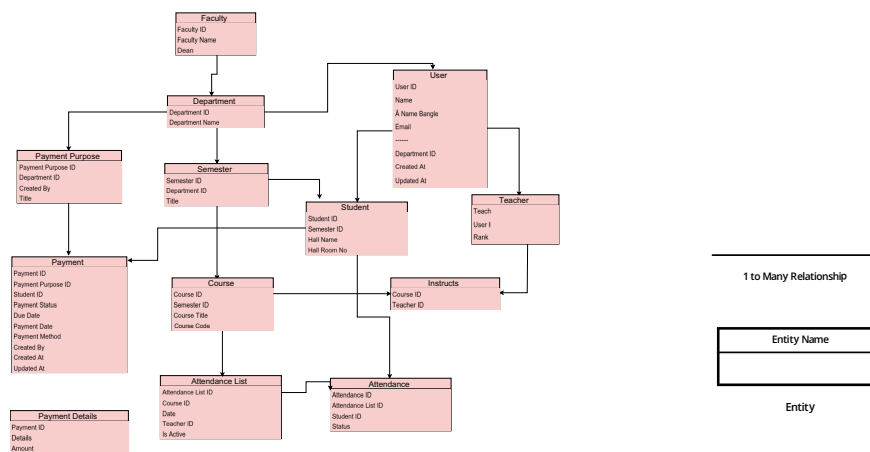


Figure 8: Logical Data Model of CU-OPAS

Write a short description of Relation model. Write a how you convert your E-R model in Relational model

6 Normalization

For less complexity we are going to rename the attributes with uppercase letters here. Following the statement, let,

A = <i>user_id</i>	N = <i>course_name</i>
B = <i>name</i>	O = <i>course_code</i>
C = <i>email</i>	P = <i>purpose_title</i>
D = <i>role</i>	Q = <i>created_by</i>
E = <i>student_id</i>	R = <i>payment_details</i>
F = <i>session</i>	S = <i>payment_amount</i>
G = <i>hall</i>	T = <i>due_date</i>
H = <i>teacher_id</i>	U = <i>payment_date</i>
I = <i>rank</i>	V = <i>payment_method</i>
J = <i>faculty_name</i>	W = <i>class_date_time</i>
K = <i>dean</i>	X = <i>class_is_active</i>
L = <i>department_name</i>	Y = <i>class_teacher</i>
M = <i>semester_name</i>	Z = <i>attendance_status</i>

The Boyce-Codd Normal form of

R(A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z)

Functional Dependency:

{
 $A \rightarrow B \ C \ D$
 $E \rightarrow F \ G$
 $H \rightarrow I$
 $J \rightarrow K$
 $N \rightarrow O$
 $P \rightarrow Q$
 $R \rightarrow S$
 $W \rightarrow X \ Y$
}

is:

R₁₁ (J, K)

FD:

{ $J \rightarrow K$ }

{

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

}

R₁₃(H, I)

FD:

{ $H \rightarrow I$ }

R₃ (W, X, Y)

FD:

{

$W \rightarrow X$

$W \rightarrow Y$

}

R₁₅ (E, F, G)

FD:

{

$E \rightarrow F$

$E \rightarrow G$

}

R₅ (R, S)

FD:{ $R \rightarrow S$ }

**R₁₆ (A, E, H, J, L, M,
N, P, R, T, U, V, W,
Z)**

FD:

{ }

R₇ (P, Q)

FD:{ $P \rightarrow Q$ }

R₁ (A, B, C, D)

FD:

R₉ (N, O)

FD:{ $N \rightarrow O$ }

The dependencies are preserved.

Here is the transcript of the algorithm:

In:

**R (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R,
S, T, U, V, W, X, Y, Z)**

FD:

{

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

$$\begin{aligned}
&E \rightarrow F \\
&E \rightarrow G \\
&H \rightarrow I \\
&J \rightarrow K \\
&N \rightarrow O \\
&P \rightarrow Q \\
&R \rightarrow S \\
&W \rightarrow X \\
&W \rightarrow Y \\
&\}
\end{aligned}$$

The determinant of $A \rightarrow B$ is not a superkey and so R is replaced by:

$$\begin{aligned}
&\mathbf{R_1 (A, B, C, D)} \\
&\{ \\
&A \rightarrow B \\
&A \rightarrow C \\
&A \rightarrow D \\
&\}
\end{aligned}$$

and:

$$\begin{aligned}
&\mathbf{R_2 (A, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U,} \\
&\mathbf{V, W, X, Y, Z)} \\
&\text{FD:} \\
&\{ \\
&W \rightarrow X \\
&W \rightarrow Y \\
&R \rightarrow S \\
&P \rightarrow Q \\
&N \rightarrow O \\
&J \rightarrow K \\
&H \rightarrow I \\
&E \rightarrow F \\
&E \rightarrow G \\
&\}
\end{aligned}$$

In:

R₂ (A, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z)

FD:

{
 $W \rightarrow X$
 $W \rightarrow Y$
 $R \rightarrow S$
 $P \rightarrow Q$
 $N \rightarrow O$
 $J \rightarrow K$
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

the determinant of $W \rightarrow X$ is not a superkey and so R₂ is replaced by:

R₃ (W, X, Y)

FD:

{
 $W \rightarrow X$
 $W \rightarrow Y$
 }

and:

R₄ (A, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, Z)

FD:

{
 $R \rightarrow S$
 $P \rightarrow Q$
 $N \rightarrow O$
 $J \rightarrow K$
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

in:

R₄ (A, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, Z)

FD:

{
 $R \rightarrow S$
 $P \rightarrow Q$
 $N \rightarrow O$
 $J \rightarrow K$
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

the determinant of $R \rightarrow S$ is not a superkey and so R₄ is replaced by:

R₅ (R, S)

FD:{ $R \rightarrow S$ }

and:

R₆ (A, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, T, U, V, W, Z)

FD:

{
 $P \rightarrow Q$
 $N \rightarrow O$
 $J \rightarrow K$
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

the determinant of $P \rightarrow Q$ is not a superkey and so R₆ is replaced by:

R₇ (P, Q)

FD:{ $P \rightarrow Q$ }

and:

R₈ (A, E, F, G, H, I, J, K, L, M, N, O, P, R, T, U, V, W, Z)

FD:

{
 $N \rightarrow O$
 $J \rightarrow K$
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

In:

R₈ (A, E, F, G, H, I, J, K, L, M, N, O, P, R, T, U, V, W, Z)

FD:

{
 $N \rightarrow O$
 $J \rightarrow K$
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

the determinant of $N \rightarrow O$ is not a superkey and so R₈ is replaced by:

R₉ (N, O)

FD:{ $N \rightarrow O$ }

and:

R₁₀ (A, E, F, G, H, I, J, K, L, M, N, P, R, T, U, V, W, Z)

FD:

{
 $J \rightarrow K$
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$

}

the determinant of $J \rightarrow K$ is not a superkey and so R_{10} is replaced by:

$R_{11} (J, K)$
 FD: { $J \rightarrow K$ }

and:

$R_{12} (A, E, F, G, H, I, J, L, M, N, P, R, T, U, V, W, Z)$
 FD:
 {
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

In:

$R_{12} (A, E, F, G, H, I, J, L, M, N, P, R, T, U, V, W, Z)$
 FD:
 {
 $H \rightarrow I$
 $E \rightarrow F$
 $E \rightarrow G$
 }

the determinant of $H \rightarrow I$ is not a superkey and so R_{12} is replaced by:

$R_{13} (H, I)$
 FD: { $H \rightarrow I$ }

and:

$R_{14} (A, E, F, G, H, J, L, M, N, P, R, T, U, V, W, Z)$
 FD:
 {
 $E \rightarrow F$
 $E \rightarrow G$
 }

In:

R₁₄ (A, E, F, G, H, J, L, M, N, P, R, T, U, V, W, Z)
FD:
 $\{$
 $E \rightarrow F$
 $E \rightarrow G$
 $\}$

the determinant of $E \rightarrow F$ is not a superkey and so R₁₄ is replaced by:

R₁₅ (E, F, G)
FD:
 $\{$
 $E \rightarrow F$
 $E \rightarrow G$
 $\}$

and:

R₁₆ (A, E, H, J, L, M, N, P, R, T, U, V, W, Z)
FD: $\{\}$

The final results are:

R₁₁ (J, K)
FD: $\{ J \rightarrow K \}$

R₁₃ (H, I)
FD: $\{ H \rightarrow I \}$

R₁₅ (E, F, G)
FD:
 $\{$
 $E \rightarrow F$
 $E \rightarrow G$
 $\}$

R₁₆ (A, E, H, J, L, M, N, P, R, T, U, V, W, Z)
FD: {}

R₁ (A, B, C, D)
{
 $A \rightarrow B$
 $A \rightarrow C$
 $A \rightarrow D$
}

R₃ (W, X, Y)
FD:
{
 $W \rightarrow X$
 $W \rightarrow Y$
}

R₅ (R, S)
FD: { $R \rightarrow S$ }

R₉ (N, O)
FD: { $N \rightarrow O$ }

7 Physical Modelling

Physical Data Modeling (PDM) is the final step in data modeling. A PDM is primarily concerned with the implementation of a database-specific data model. As a result, it depicts how the model will be created in the database. For a non-technical individual, this data model is difficult to comprehend. This data model is required to create a query for CRUD activities. We created a Physical Data Model based on Logical Data Modeling (LDM) and Normalization. We utilized the MYSQL database server and applied the FDM reconsideration to it in this case. The name convention for entities and attributes was "Snake Case." For example, we used "student id" for the attribute name "Student ID." The Physical Data Model we used in our MYSQL database server is shown below.

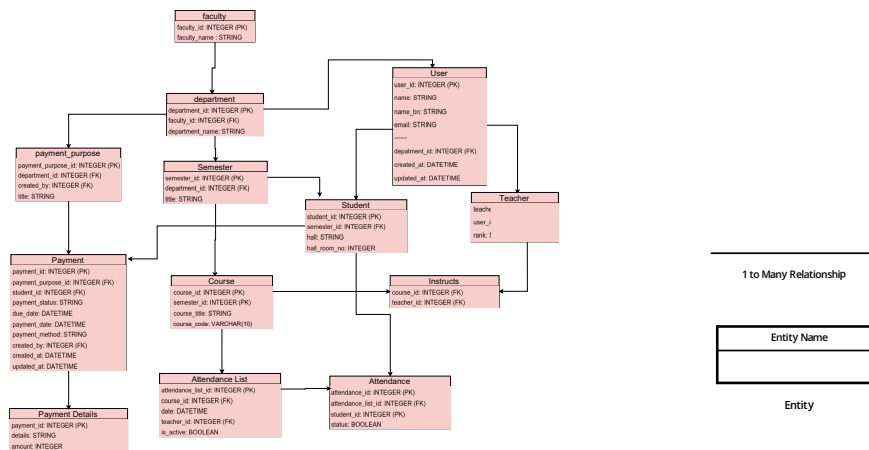


Figure 9: Physical Data Model of CU-OPAS

Write a short description of Relation model. Write a how you convert your E-R model in Relational model

8 System Architecture

Due to the fact that it is a web platform, the system is divided into two parts: the frontend and the backend. Frontend refers to the component of the system that will be visible to the user on the screen of a mobile device or computer. Whereas the back end refers to parts of the system or a program's code that allow it to operate and that cannot be accessed by a user. The back end is also called the data access layer of a system and includes any functionality that needs to be accessed and navigated to by digital means. The frontend is designed with more than 25 libraries including leading frontend library *React.js*, *MaterialUI*, *Ant-Design*, **Bootstrap** and so on. These libraries are written on *Javascript* programming language and is able to be integrated with any other programming language based backend libraries. We used more than 15 Javascript based frameworks such as *Express Js*. For database management, we used *MySQL*⁵ which is an open-source relational database management system. When a user tries to log in to the system with credentials, the the system catches the input data and send those to the backend first for certifying the authorized user. Backend functions then connect with the database server, and certifies the user.

All of the front-end information that a user sees is retrieved from the database and sent to the browser using backend functionalities. On the dashboard, different options are shown based on the roles of the users. CRUD (Creat, Read, Update, Delete) operations, that may be performed on the selections of the user from the dashboard. API⁶(*Application Programming Interface*) allows the backend to retrieve and alter data into the database through SQL queries. The following figure is a diagrammatic representation of the work-flow of the system.

⁵MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

⁶An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification.

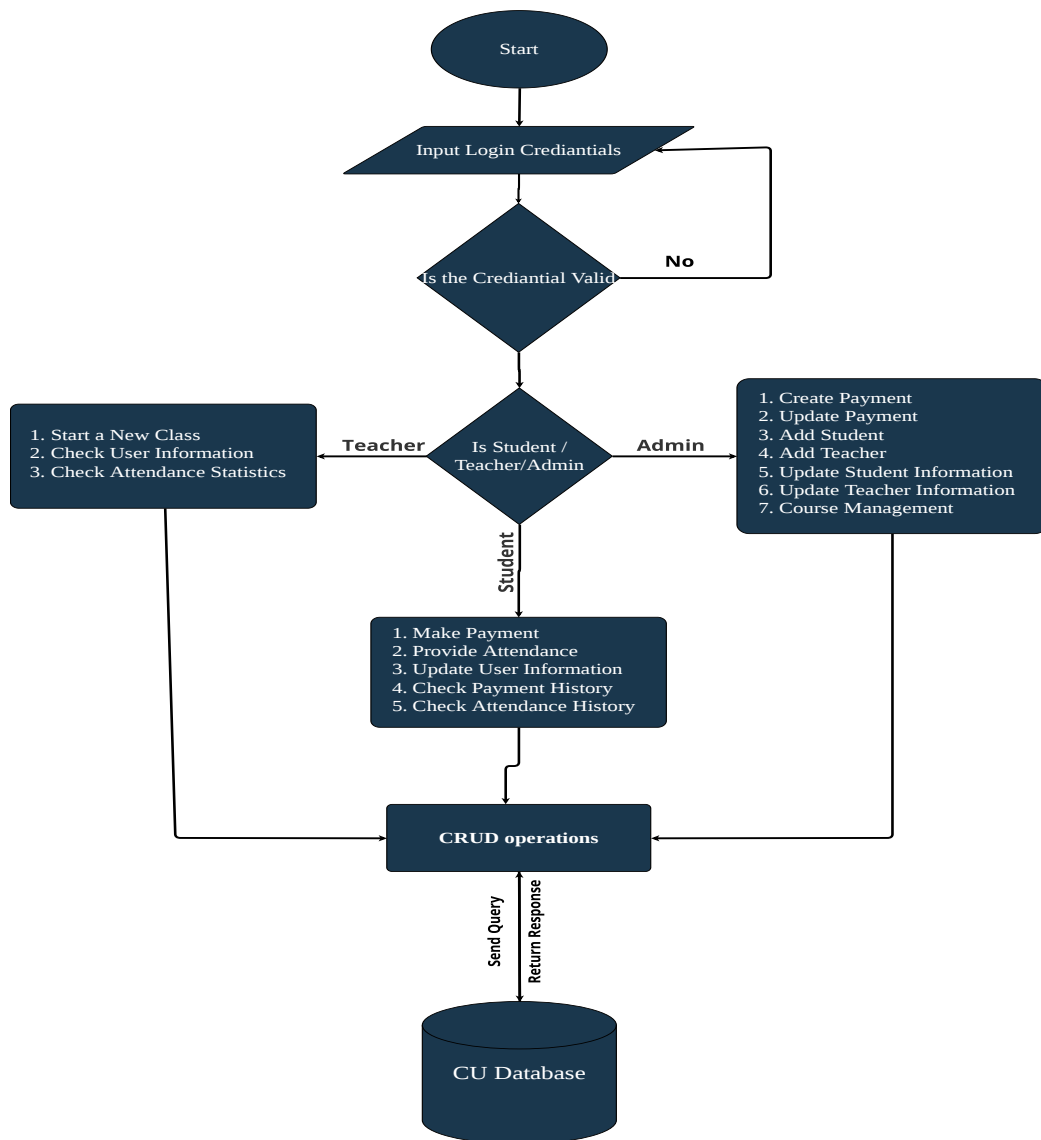


Figure 10: Work flow of the system

9 Implementation

Implementation is the last part before testing. It is divided into two parts: frontend Implementation and Database or Backend Implementation. Our coding environment was "VS CODE". It is a trendy code editor authored by Microsoft. For the version controlling system, we used GitHub. GitHub is the best version controlling system in the current world. Our online GitHub repository is located at "https://github.com/Brainless-LoCo/dmbs-project". We used "Node Package Manager" (npm).

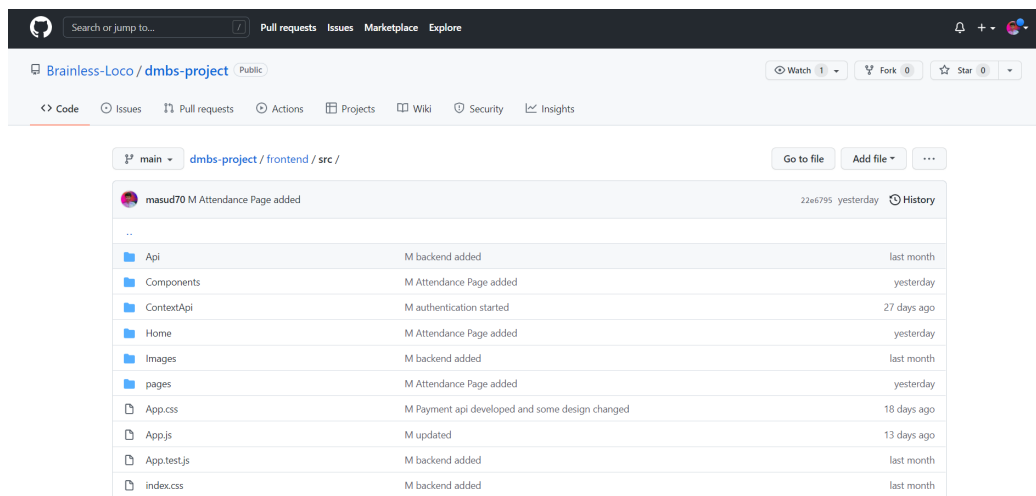


Figure 11: Github repository of CU-OPAS

9.1 Frontend

As mentioned before, to implement the frontend, we used Javascript based framework named "React", including various libraries like 'moment js', 'antd', 'material UI', 'react spinners', 'jquery' etc. some of the frontend UI design of our system is mentioned below.

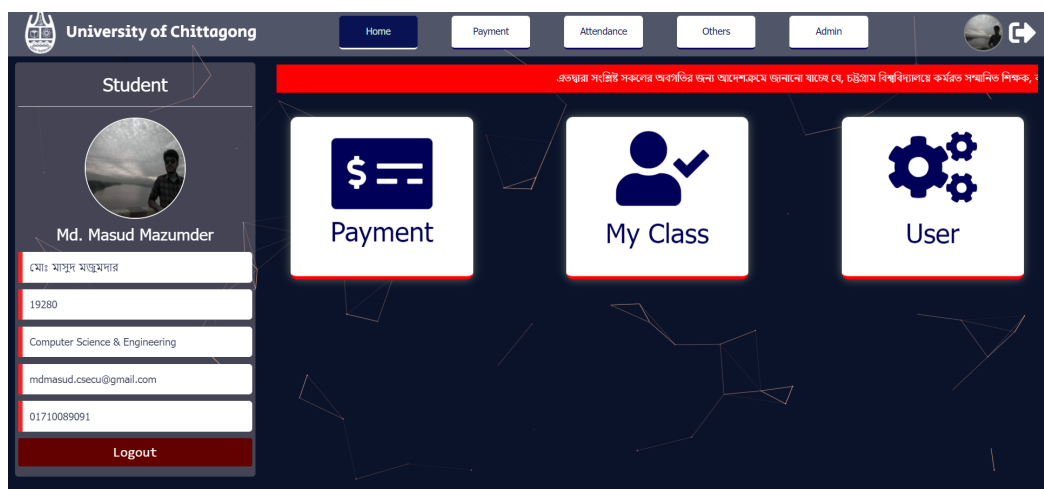


Figure 12: Home Page User Interface

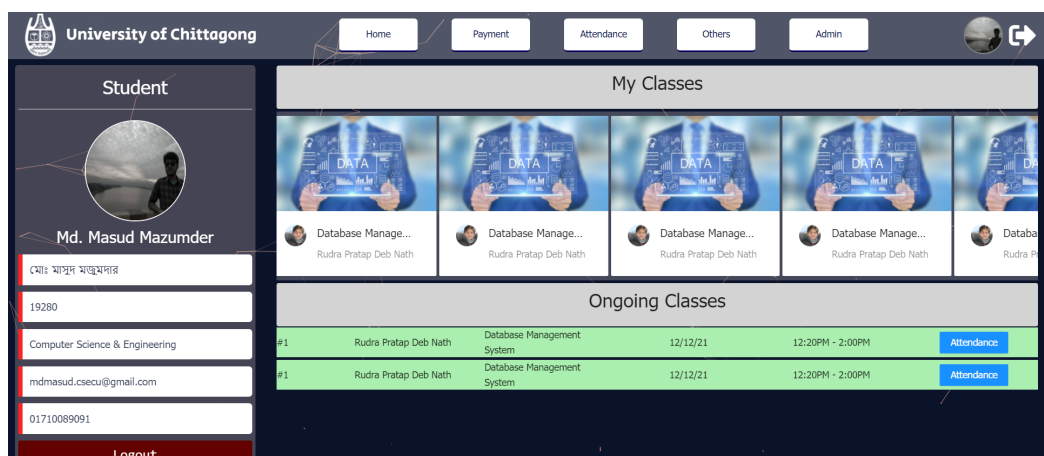


Figure 13: Student Attendance User Interface

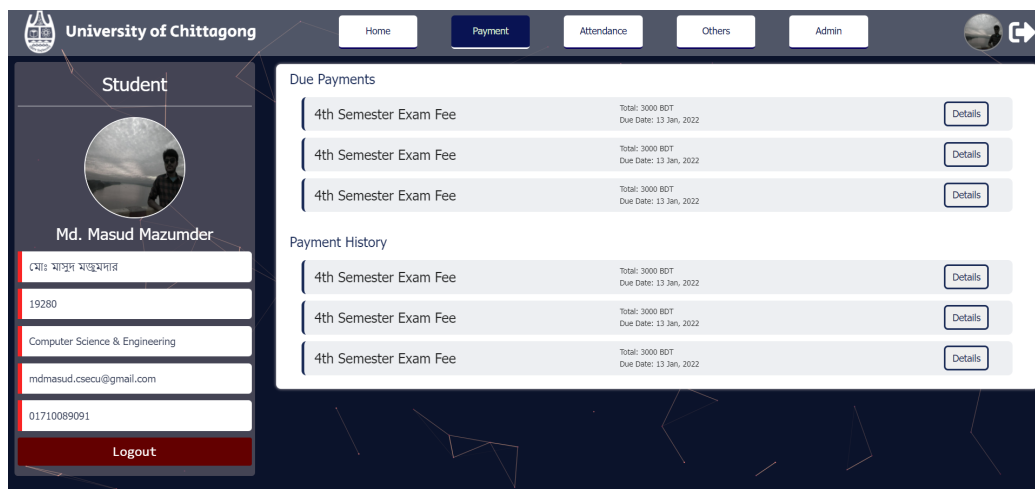


Figure 14: Payment History User Interface



University of
Chittagong

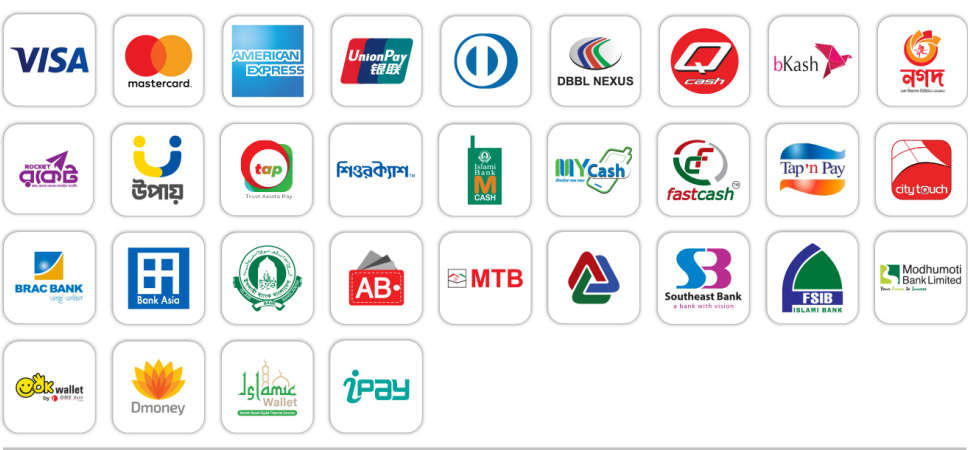
Money Receipt



Student's information		
Name	:	Tonmoy Chandro Das
Department	:	Computer Science & Engineering
Student ID	:	19701066
Session	:	2018-2019
Last Date	:	21/12/21
Payment Status	:	Unpaid
Payment Details		
Details		Amount
Tuition Fee	-	350/-
Transportation Fee	-	320/-
BNCC Fee	-	300/-
Total	-	970/-

Figure 15: Payment Slip

Pay With



Verified By **SSLCOMMERZ**

Figure 16: Available Payment Methods

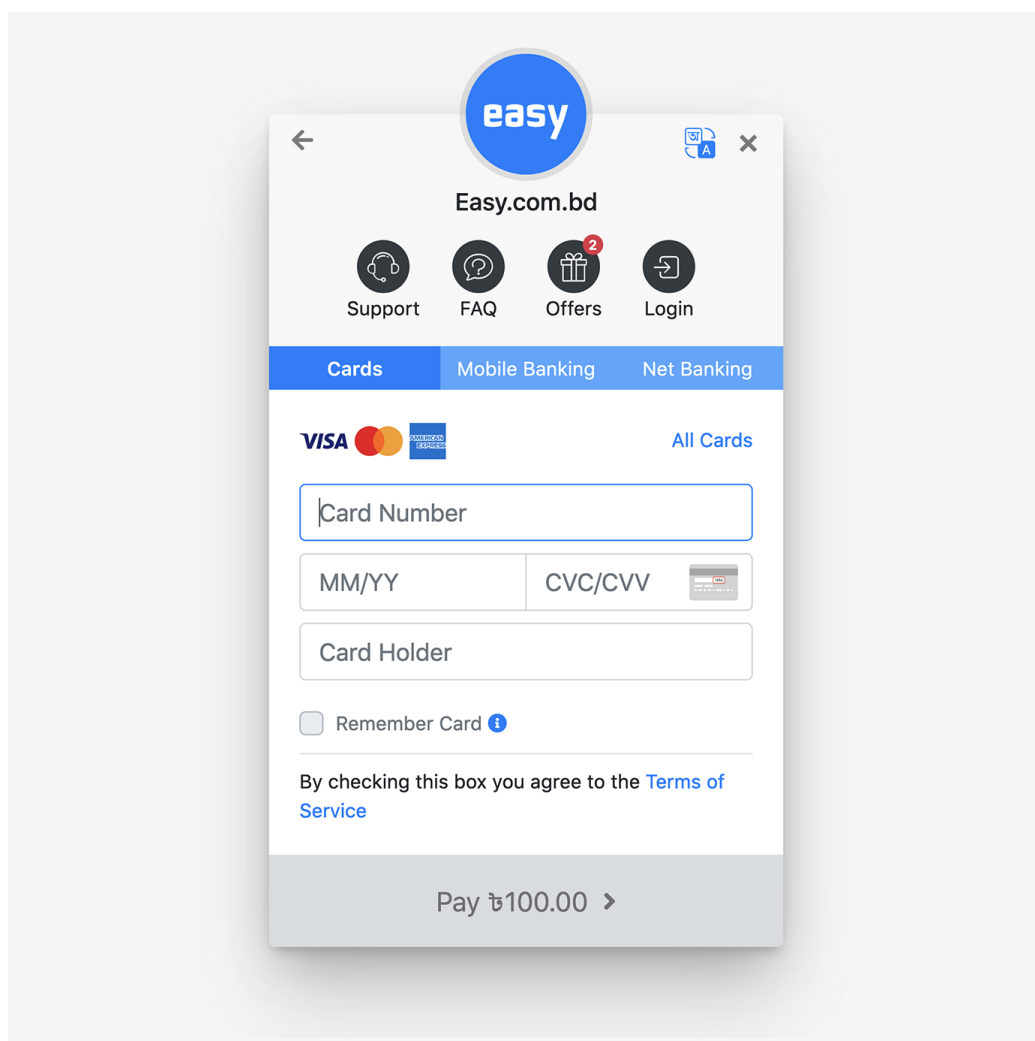


Figure 17: Payment Procedure

University of Chittagong

Home | Payment | Attendance | Others | Admin

Teacher

Md. Masud Mazumder

মোঃ মাসুদ মাসুদমার

19280

Computer Science & Engineering

mdmasud.csecu@gmail.com

01710089091

Logout

< Database Management System

Course Teacher : Rudra Pratap Deb Nath
Course Credit : 4
Semester : 4th
Total Student : 88

Total Class : 15 Create A New Class: [Create Now](#)

Class Title	Date	Time	Code	Participants
<input type="checkbox"/> Database Management System	21/10/21	12:20PM - 2:00PM	12345	80/88

Student ID	Name	Status	Action
19701070	Md. Masud Mazumder	Present	Accept
19701071	Md. Masud Mazumder	Present	Accept
19701072	Md. Masud Mazumder	Present	Accept

Figure 18: Teacher Attendance Interface

University of Chittagong

Home | Payment | Attendance | Others | Admin

Teacher

Md. Masud Mazumder

মোঃ মাসুদ মাসুদমার

19280

Computer Science & Engineering

mdmasud.csecu@gmail.com

01710089091

Logout

Tonmoy Chandro Das
19701070
Computer Science & Engineering
Faculty of Engineering

Update Password

New Password Confirm New Password [Update](#)

Hall Informations

Hall Name Room No. [Update](#)

Basic Informations

Name Name (Bangla)
 Father's Name Father's Name (Bangla)

Figure 19: User Update Form

9.2 Database

We implemented our database using the MYSQL server. MYSQL is an open-source relational database that is fast, reliable, and easy to use. It uses a particular query language called "Structured Query Language" or SQL. Listing 4 shows an SQL query.

```

1 SELECT
2     user_id ,
3     NAME ,

```

```

4      name_bangla ,
5      email ,
6      phone ,
7      dob ,
8      image ,
9      religion ,
10     father ,
11     mother ,
12     nationality ,
13     present_address ,
14     permanent_address ,
15     marital_status ,
16     PASSWORD ,
17 STATUS
18     ,
19     role ,
20     department_name
21 FROM
22     USER user_id = 101

```

Listing 1: A SQL query to find a User

```

1 SELECT
2     student_id AS id ,
3     user_id ,
4     NAME ,
5     name_bangla ,
6     email ,
7     phone ,
8     dob ,
9     image ,
10    religion ,
11    father ,
12    mother ,
13    nationality ,
14    present_address ,
15    permanent_address ,
16    marital_status ,
17    PASSWORD ,
18 STATUS
19     ,

```

```

20     role ,
21     SESSION ,
22     department_name ,
23     allotted_hall ,
24     semester_id
25 FROM
26     student
27 NATURAL JOIN USER NATURAL JOIN
28 department WHERE user_id = 1

```

Listing 2: A SQL query to find a Student data

```

1 INSERT INTO payment_purpose(
2     purpose_title ,
3     department_id ,
4     created_by ,
5     created_at
6 )
7 VALUES(?, ?, ?, ?)

```

Listing 3: A SQL query to insert payment purpose

```

1 SELECT
2     *,
3     (
4     SELECT
5         COUNT(*) AS total
6     FROM
7         student
8     NATURAL JOIN semester NATURAL JOIN
9     course WHERE course_id = ?
10 ) AS total
11 FROM
12     semester
13 NATURAL JOIN course WHERE course_id = ?

```

Listing 4: A SQL query to find course data

9.3 Backend

Also, for backend implementation, we used the Javascript-based framework "Express JS". This is the most popular backend library in the current world.

We've also used some dependencies like 'MySQL', 'Moment JS', 'bcrypt', 'jsonwebtoken' etc. Some DML code snippets are as below.

```

1 const router = require("express").Router();
2 const { checkToken } = require("../auth/tokenValidation");
3 const { check, validationResult } = require("express-validator");
4 const { ... } = require("../userService");
5 const { ... } = require("../userController");
6
7 router.post("/login", login);
8 router.post("/data", checkToken, getData);
9
10 router.post("/addStudent", [
11   ], createStudent);
12 router.post("/addTeacher", [
13   ], createTeacher);
14 router.post("/userByEmail", getUserByEmail);
15 router.post("/createPayment", createPayment);
16 router.get("/allStudents", getAllStudents);
17 router.get("/allTeachers", getAllTeachers);
18 router.patch("/updateStudent", updateStudent);
19 router.patch("/updateTeacher", updateTeacher);
20 router.delete("/", deleteUser);
21 router.get("/:id", (req, res, next) => {
22   console.log(req.params.id);
23   next();
24 }, getUserById);
25
26 module.exports = router;

```

Figure 20: Backend API implementation (part-1)

```

1 const { hashSync, gensaltSync, compareSync } = require("bcrypt");
2 const { sign } = require("jsonwebtoken");
3 const { validationResult } = require("express-validator");
4 const { ... } = require("../userService");
5
6 module.exports = {
7   login: (req, res) => {
8     const body = req.body;
9     const res = {};
10    getStudentById(body.id, (err, results) => {
11      if (err) {
12        console.log(err);
13        next(err);
14      }
15      if (results) {
16        if (results) {
17          return res.json({
18            status: false,
19            data: "User not found."
20          });
21        }
22        const result = compareSync(body.password, results.password);
23        if (result) {
24          results.password = undefined;
25          const jwtToken = sign({ result: results }, process.env.SECRET_KEY, {
26            expiresIn: "7d"
27          });
28          res.json({
29            status: true,
30            data: jwtToken
31          });
32        }
33      }
34    });
35  },
36   getStudentById: (body.id, (err, results) => {
37     if (err) {
38       console.log(err);
39       next(err);
40     }
41     if (results) {
42       return res.json({
43         status: false,
44         data: "User not found."
45       });
46     }
47     const result = compareSync(body.password, results.password);
48     if (result) {
49       results.password = undefined;
50       const jwtToken = sign({ result: results }, process.env.SECRET_KEY, {
51         expiresIn: "7d"
52       });
53       res.json({
54         status: true,
55         data: jwtToken
56       });
57     }
58   })
59 };

```

Figure 21: Backend API implementation (part-2)

10 Validation

Show that users are satisfied with your product. You can also give a user manual here describing how to use your system (process of completion of different tasks using your system) You can use some matrices (time, cost, resource etc.) to compare your system with the previous system.

After finalizing the system in accordance with the team's plan, we showed it to the students and teachers and asked them to test it out for themselves. We were fortunate in that we didn't spot any problems with the system when testing all of the functions using actual data. We conducted user surveys to get insight into their perceptions of the system, and we were offered some recommendations for future enhancements as a result. Our testers provided us with a wide variety of valuable insights, and the end result was pretty exciting. Students and teachers experienced less struggles as a result of the developed system as compared to the current system.

The basic differences we noted from the insights and thoughts of testers between the current system and our developed system is shown below.

- **Time**

Out of all the facts, our project's main concern was to reduce the time to perform the same operations. We have seen in the statistics that it costs 98 percent less time to submit the tuition fee through our system than current analog system. The time is optimized as students don't have to stand in a queue, bank officers don't have to attest the payments manually, and students can pay their fees anytime they wish.

- **Efficiency**

The newly developed system is highly efficient than previous system. Authority has expressed their satisfaction with the system as they don't have to calculate data manually anymore in case they use this system. This system is proven to be better as there are almost zero percentage chance to be unsuccessful to submit the on due time.

- **Cashless**

Moving with the digital world, this system introduces the university with a cashless payment system.

- **Independence**

The system only depends on an online payment gateway to perform the transaction which allows users to make payment through any medium such as bank, mobile bank etc. In this way the university will get rid of dependency on specific bank branch. As the current system fully

depends on the bank branch, it also depends on a certain period of whole daytime that is working period of the bank. But in our system, there are no chances to depend on any specific time period. One can pay fees any time they wish.

Student and administrations don't have to be present physically to complete transactions anymore. In our system, both students and administrations can perform tasks from anywhere just using the internet.

- **Time saving with attendance system**

It has been seen that teachers can now save 25 to 30 percentage of the class session if they start attesting attendance through our system. It not only saves time but also reduce the discomfort of taking attendance of teachers. Students were seemed happy to use the new attendance system.

- **Data Analysis**

In current system, administrations need to calculate several data manually which costs a good amount of time and mental work. But in our proposed system, those data are automatically generated by the system and also represented graphically which will help authority to reduce the difficulties to make decisions. It should be noted that, in current system, calculations can go wrong as it requires more people to complete.

Since a vast number of users expressed the satisfaction with the developed system, administrations may take necessary decisions to replace the current system with the new one.

11 Software Deployment

To use the system, one must be familiar with the internet. The system is available through the internet from any device with a browser. In the browser, if one enters the website address that is `student.cu.ac.bd` of the system, the browser will redirect to the landing page of the website. The landing page, contains a basic form of two fields which are *ID* and *Password*. Every student will be provided with a distinct ID for academic identity purposes right at the time of being admitted to the university, and with that, he will be able to log in to the system. Similarly, all the teachers and admins will be provided with their own IDs when they are appointed to the role. There are no sign-up option in the website since all the information of newly admitted students and newly appointed teachers will be added to database by an admin from the authority. So only the permitted or authorized students and teachers will be able to use the system.

After completing the login form, the user will be led to a dashboard based on his role. For a student role, one will get three options which are *Payments*, *Attendance*, *Profile Update* on the dashboard. A teacher will have *Attendance*, *Profile Update* option on the dashboard. An admin however, will have access to add a new Student, a new teacher, update any information of a student or teacher, add a new fee, see and analyze the statistical data of attendance or payments and so on. An admin will also be able to add or update a payment for a specific department or a group of students of that department or even for a specific student. The payment will be automatically added to the profile of those students. Students will be able to see all the payment lists that are to be paid and all the payment lists that have already been paid from their page. They will be able to download and print the receipt of any payment from their profile which will obviously contain the payment status that is *paid* or *unpaid*.

In order to certify attendance, teachers will be able to create a class session, and the system will produce a temporary code for students to use. As a temporary measure, the student will find a class session on their account, and if they enter the code they received in that session into that class throughout the class period, the system will automatically verify his or her attendance. If the teacher and administration want it, the system will display all of the statistical data that has been generated automatically.

The system will have two types of admin which we primarily named as *Super Admin* and *Sub Admin*. When someone from the department administration or a faculty member acts as sub admin, they will only be allowed to alter or update data that pertains to his or her department such as information that belongs to student or teachers, student scholarship

information etc. A super admin, on the other hand, is someone who has the authority to edit, change, or update any data in the system. A super admin will have the ability to create new administrators, add officially new appointed teachers, cancel or alter any payment fees, and see all of the statistical data provided by the system and so on. In addition, it should be emphasized that all actions performed by an admin will be logged in the database with an accurate time stamp, and a super admin will be able to see the details of that activity.

The main links to get to the required pages are located in the menubar at the top of the website, which may be accessed from any page at any time. It has a number of essential links in the footer section that will guide viewers to the university's official website, the institution's Facebook page, the notice board, and other relevant resources.

Any time a user desires, he or she may log out of the system by clicking the *Log Out* button located on the sidebar of the system. Any authorized user can log in and log out of the system any time from anywhere and that is what makes the system independence in terms of location and flexible.

12 Conclusion and Future Work

The risks avoidance and reduction strategies that could be implemented to ensure project success is: Defining project key success factors at the beginning, then throughout the development stages, each is resolved by designing the appropriate policies, regulations, system model, and database tables design. For the development of an online payment system to handle tuition fees as well as other payments and attendance systems to reduce the time consumption as well as to make attendance taking procedure more flexible and more transparent which integrates the current university system, we started with the enlisting the issues and difficulties faced by both students and administration while using the current analog system and then came up with the various ideas to solve these problems. While choosing the best idea among those, we took into consideration the system being uncomplicated to use, transparent, flexible, being unbounded from time, physical presence, being usable with multiple possible ways of transactions, region independent. Compared with traditional payment methods such as pay-by-check, pay-by-phone, or wire transfer, online payment is considered more time- and cost-efficient, convenient, and flexible. The system not only offers these mentioned properties but also offers an integrated attendance system that can replace the current analog time-consuming system. Our developed system proposes a paperless attendance attesting system that is easy to use and offers more features and automatically generated statistics, usually done manually.

In an educational institute like the University of Chittagong, where the administration has to deal with a massive number of students, it has been an exigency situation to replace the analog system with a digital one. Our system ensures the students' gratification and ensures the liberty of administration from being dependent on a specific bank as an agent to make the transaction with students. Once teachers start using the newly built attendance system, they can save even more time for teaching, as seen in the statistics. Administrations will be able to see more statistical data on both payments and attendance, which could be helpful to make significant and critical decisions. The purpose of this project is to resolve the issues with the current system and replace the analog procedure of both payment and attendance with an efficient, flexible, convenient, and secured system and our developed system promises to offer these core features as we have used the latest technologies to build it.

Despite being efficient, reliable, flexible, and secured, we found two limitations in this newly developed system. Stack-holders, i.e., students, teachers, and administrations, must be experienced in using the internet as a prerequisite to use this system. Otherwise, the system may appear difficult to use sometimes. Following the prerequisite, the major limitation of this system

is that it requires the internet to use it. As the system is developed as a web application, it can be accessed only via browsers. Although any devices such as mobile, tablets, and desktops with any operating system can access this system via browser, without the internet, the browser will not be able to access the system. Whereas the internet makes this system convenient and flexible, it makes the system depends on it.

When the system was given to use primarily to experience by the students and teachers on testing purposes, they expressed their satisfaction and suggested integrating more features into it. We aim to integrate the whole student management system into it to make all the administrative operations paperless and digital. While moving forward to the coming days, the system will require more modifications and updates as the number of students as well as teachers will always increase.