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

List of Figures

List of Tables

Listings

Abstract

The University of Chittagong's online payment and attendance system is the objective of this project report. A large number of students at the university pay all 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. Moreover, manually attendance system is still being practiced here. Both of this analog systems are 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. On the other hand, current manual attendance system consumes a significant amount of time everyday. Against this backdrop, we began work on a project to create an alternative payment and attendance system that would allow students to pay and show up for class online. This method ensures that all students are acquainted with the online payment processes. Additionally, taking attendance online will save time and classes will be more effective.

To maintain the system development process, we used the Software Development Life Cycle (SDLC) and the Scrum method to help team members work together. The system was developed using a JavaScript based framework called "React", which includes Cascading Style Sheets(CSS) for the front-end and "ExpressJs" for the back-end, as well as an Apache web server and a MySQL database server. Testing and validation of the system were also carried out by enabling users 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. Our system can develop this system 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 and the time-consuming manual attendance method.

1 Introduction

One of the most significant advantages of employing technologies over analog processes is that it is less expensive in terms of time. Conventionally, we continue to do all of the official activities including fees payment as well as attendance 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 imposed by the authorities. Furthermore, there is always the possibility of unsuccessful finishing of the task. Considering these circumstances, our objective is to create an application that will transfer these two procedures (payment and attendance system) to a digital platform, which will be able to complete the duties in the shortest amount of time imaginable and with the least amount of paper work possible. The system will 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 system would be user-friendly in it's 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.

1.1 Background and Motivation

Every year since the university's founding in 1966, students have increased by a small but steady margin. Even though the globe is being exposed to new technology daily, the method for receiving fees and the attendance system at the University of Chittagong have stayed the same. The dependency on a single branch of a single bank makes the payment process of tuition fees difficult to manage not only 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 year after year, university administrators and bank administrators have had enough of this terrible procedure. The same may be said regarding the old method of attendance. 25 to 30 percentile of a class period is devoted to traditionally attesting attendance.

If we take a closer look, we can see that these issues are interconnected. We want to create a system to tackle these difficulties while also processing primary data from both students and instructors. The system does not need paperwork to accomplish the functions and demands the most negligible physical presence from authorities and students. This system will convert both the payment and attendance systems to a one-click operation. Administrators' capability to query the necessary information will be more accessible than ever before. Students and administrators may use this system to run processes from any location, with an immediate time stamp to validate them.

1.2 Problem Statement

Develop a database system to handle online fees payment and attendances.

The system may use 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 needs to hold the records of payments already done and have to be done by a student. Also, implement an online attendance system using the information of courses from a teacher and the students studying the course on the same platform.

1.3 System Definition

Generally, an online application uses the internet, and users can have access globally. According to requirements, we are supposed to implement an online application named "Chittagong University Online Payment and Attendance System (CU-OPAS)." It will manage the student fees and attendances. A digital platform-based system will save time and provide a paperless working facility.

1.4 System Development Process

To develop our system, we use the system development life cycle (SDLC) process [https://g.co/kgs/nFcYCv]. Figure 1 shows the different steps of the SDLC process: Requirement Gathering, Requirement Analysis, Database Modeling, Architectural Design, Implementation, and Testing and Validation. It is a cyclic process and allows to include new requirements given by uses at the validation stage. SDLC provides an effective method and framework to develop a system. It also provides enables project planning, scheduling and estimating. As a result, developers can estimate costs and have a proper planning for project tracking and control. These increases the development speed and enables developers to design and build a high-quality system. Considering all of these facilities, we've chosen it.

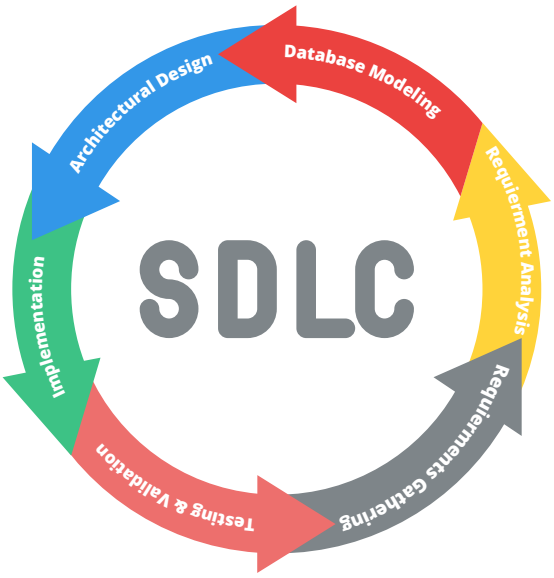

Figure 1: System Development Life Cycle (SDLC)

In the following, we briefly define the each step.

Requirement Gathering 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.

Requirement Analysis 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.

Database Modeling 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. [www.en.wikipedia.org/wiki/Data_model].

- 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.

Architectural Design The architecture of a system describes its major components, their relationships (structures), and how they interact with each other. It encompasses all of the system's components, as well as subsystems that handle all of the system's operations in their entirety. Figure 10 shows the architectural design of Chittagong University Online Payment and Attendance System's (CU-OPAS) architectural design.

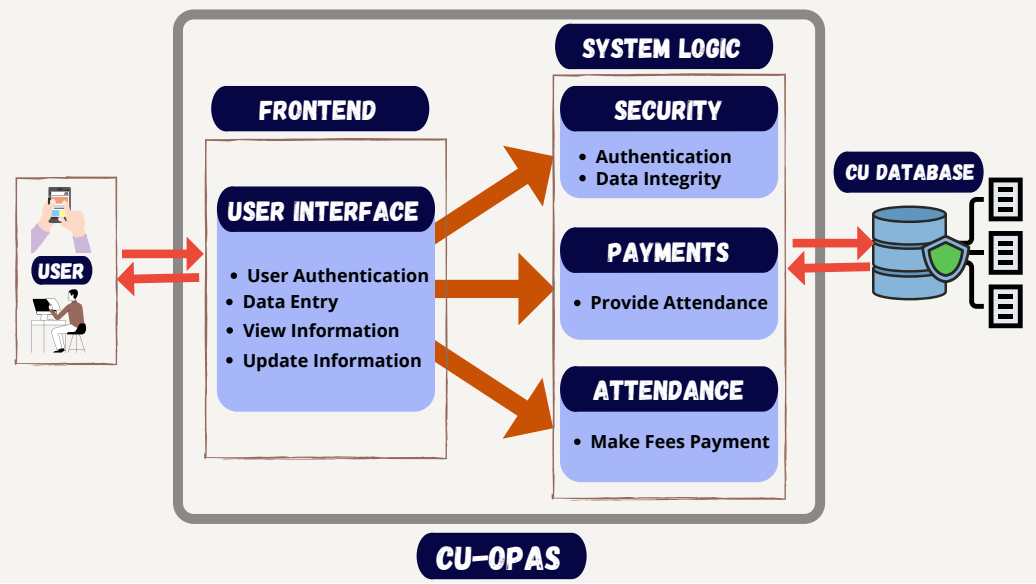

Figure 2: Architectural design of CU-OPAS

CU-OPAS uses request-response patterns to communicate with users. Any legitimate user of this system may request inquiries about payment and attendance activities from the system front-end. Then the system

generates a response communicating with database as shown in the figure.

Implementation 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.

Testing and Validation 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 provide an overview of how we designed the database and enhanced it step by step as most as possible. Section 8 and Section 9 provide 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

According to Section 1.4 we are using Software Development Life Cycle (SDLC) as our software development process. There are various method of SDLC. Among them we are using Agile SDLC method. Also, there are various Agile methodologies. Here we are using Scrum methodology. This can be shown as follows.

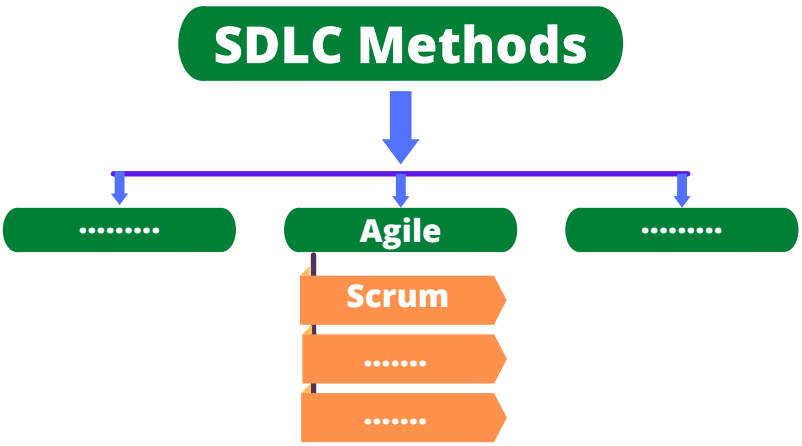

Figure 3: SDLC classification

The Agile SDLC model is a combination of iterative and incremental process models with a focus on process adaptability and customer satisfaction by rapid delivery of working software products. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross-functional teams working simultaneously on various areas like
-

- Planning/Requirement Gathering
- Requirements Analysis
- Database Modeling
- Architectural Design
- Implementation

- Testing and Validation

Ref: [www.tutorialspoint.com/sdlc/sdlc_agile_model.htm] Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Here is a graphical illustration of the Agile Model -

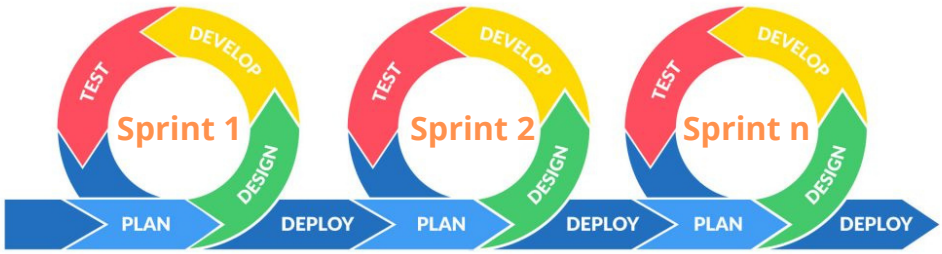

Figure 4: Agile Model

We define separate phases (sprints in Scrum terminology) for each and every steps of SDLC. For each phase we make a plan, design a work-flow, develop it, and finally test the output of the phase. If the output is correct according to the requirements, then we proceed for the next phase.

To complete each of these phases or sprints we followed the Scrum methodology. It actually allows implementing Agile development methodology. So it can be called a framework which enables an iterative and incremental development process. At the end of each step a usable product is delivered to a customer. Customer feedback helps reveal possible problems or change the initial development

plan if needed.
Here are the main roles involved in the development process, according to the Scrum model:

Product owner : The product owner takes care of the end user's interests. In our project *Tonmoy Chandro Das* plays this role of a product owner.

Scrum master : The Scrum master coordinates the whole development process. Another task is to make sure that Scrum is used properly and to hold regular Scrum meetings. In our case, *Md Masud Majumdar* plys the role of team leader from the beginning to the conclusion, and he planned and executed the whole project.

Scrum team : The Scrum team develops the product. Its main tasks are programming, analysis, testing, etc. Our scrum team contains all of the members of our database team including-

- Md. Masud Mazumder
- Tonmoy Chandro Das
- Palash Hossen
- Tareq Rahman Likhon Khan
- Hamza Mohtadee Ibne Mamun
- Nu Sai Mong Marma

Each scrum team member has significant contributions to various aspects of the overall project. For example we holds regular virtual meetings where all the teammates together discuses, came up with all the various ideas to design the database schema etc. The meetings we hold virtually through Google Meet and Zoom software (Online software for meetings) conducts *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 also looked after the documentation of this entire project. This documentation was prepared under the leadership of *Hamza Mohtadee Nafi*.

Md. Masud Mazumder and *Tonmoy Chandro Das* write codes to implement our system mostly.

Now, let's take a look at the main steps of the development process that Scrum consists of which we followed.

1. **Product Backlog Creation**
   In this process, we transformed the significance and functional details of the system into short stories. Every user story gets a unique ID. As a rule, user stories have the following format: As a [User Role], I want to [feature body] so that [User profit]. This list below shows how these stories can look like. These are actual product requirements that were implemented during the software developing process:

UserID	User Type	User Story
user-01	Student	I want to pay all my tuition fees via online from anywhere in the world.
user-02	Student	I want to see all of my previous transactions and due transactions that I need to complete.
user-003	Administrative personnel	I need to create, update and edit payments for each and every students.
user-04	Administrative personnel	I want to have track of the payments for all the students.
user-05	Teacher	I need to collect attendance online for each and every classes I take.

Table 2: User story example.

   After listing all the product backlog items, it's time to sort through them and prioritize the tasks that which tasks are more important than others. Most important one is ranked higher than less important one. This is called prioritization. We then make a list ordering according to the priority in descending order. This one is a continuous process. Because we continuously add a new item in the backlog list and prioritize it and update the listing.

2. **Sprint planning and creating backlog**
   Second step of Scrum methodology is sprint planning and creating sprint backlog. One of the most important task of this step is to make a duration of each sprint. As there is a time frame for the respective "Database Systems" course, we have determined our sprint duration for about 1 to 2 weeks. We have set a goal for each and every sprint. After that we our product owner determines importance of user story and Scrum team selects the most important user stories from the product backlog. Thus the sprint backlog is created which will be completed during the respective sprint.

3. **Working on sprint**
   This is a phase of practical application. The real stories is reorganized as a discrete task in the sprint backlog, which is where the actual work begins. In order to get started, we create a task board, also known as a Scrum board with a large number of cards in use. A scrum board is

mainly a collection of tasks of a sprint. Each task is represented by a single card in a scrum board. The cards can be arranged according to their importance. When work on a task has been started, the corresponding card is moved from the "To do" field to the "In progress" one. When work is completed, the card is moved to the "Testing" field, and after the task is successfully tested, the card goes to the "Done" field. An example of a Scrum board is shown below.

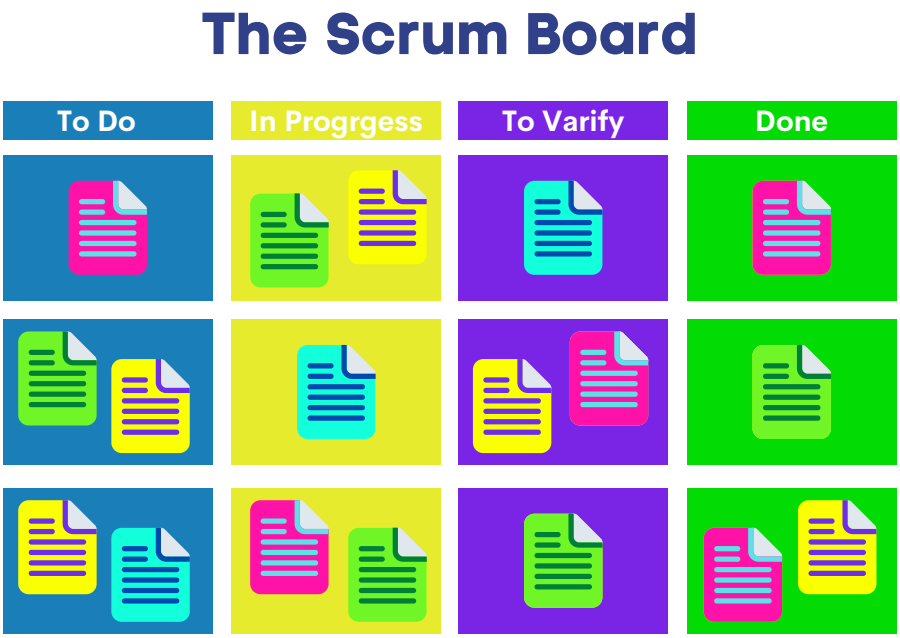

Figure 5: Project Management Agile Scrum Board Template

To fulfill the desire of a scrum board we use "Trello" which is an online tool to help creating Scrum board. Below is an example of a Scrum board in Trello of our project work.

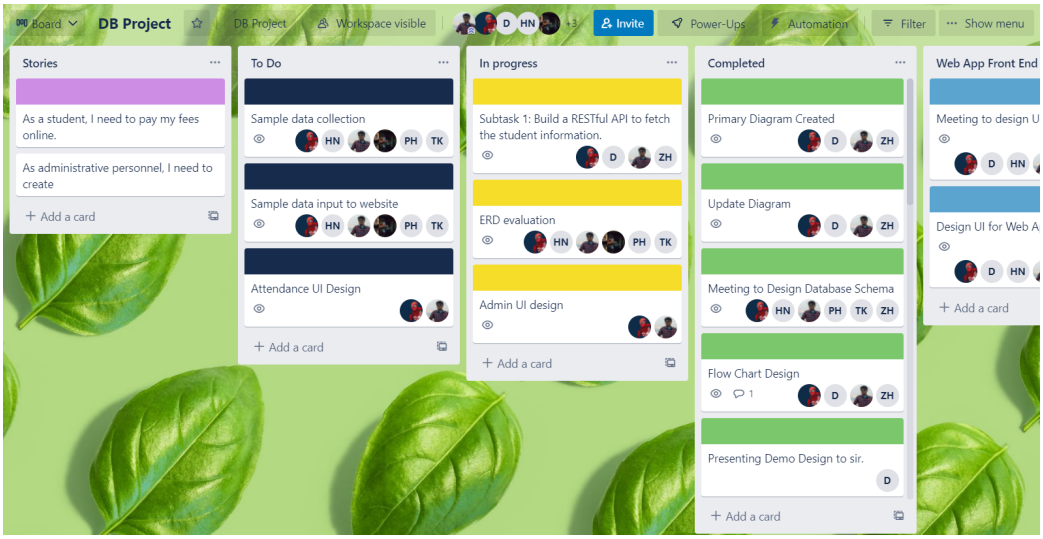

Figure 6: Example of Trello of CU-OPAS

We can easily maintain our scrum board using Trello. After completing a task on a specific field we can easily move it to another field using drag and drop property. This makes our work more easier. Aslo, a important property of Trello is that, we can assign each and every task with a deadline to a specific member of our scrum team. According to deadline, our members complete the task assigned to him.

4. **Testing and Product Demonstration**
   Generally, after each iteration, the development team creates a new version of a software product with increased value. Stack-holders and customers which includes students, teachers, and other relevant authorities reviews our system in this step. They provide their valuable comments. If every thing is positive, then the respective sprint is successful. Otherwise, we make some modification depending on the results of the study.

5. **Retrospective and the next sprint planning**
   After completing all the previous steps, it is necessary to examine what is gone well and what might be improved for the following level. We speaks about the lessons gained and the hazards of any specific challenges or problems that come up throughout the course of the project. We discusses our next sprint, which will consist of future development. An important feature is that at this stage it is the processes of work and interaction that are discussed in order to improve the work of the Scrum team as a whole. We finally conclude what went

well during the working process and what can be done better during future iteration. Then concentrate on the next sprint planning.

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.

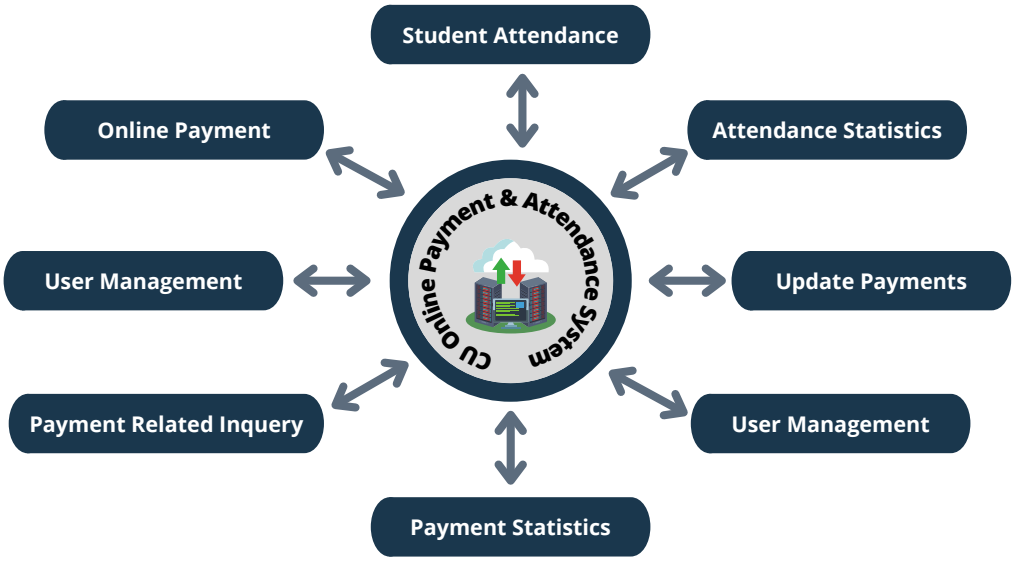

Figure 7: Context Diagram of CU-OPAS

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.

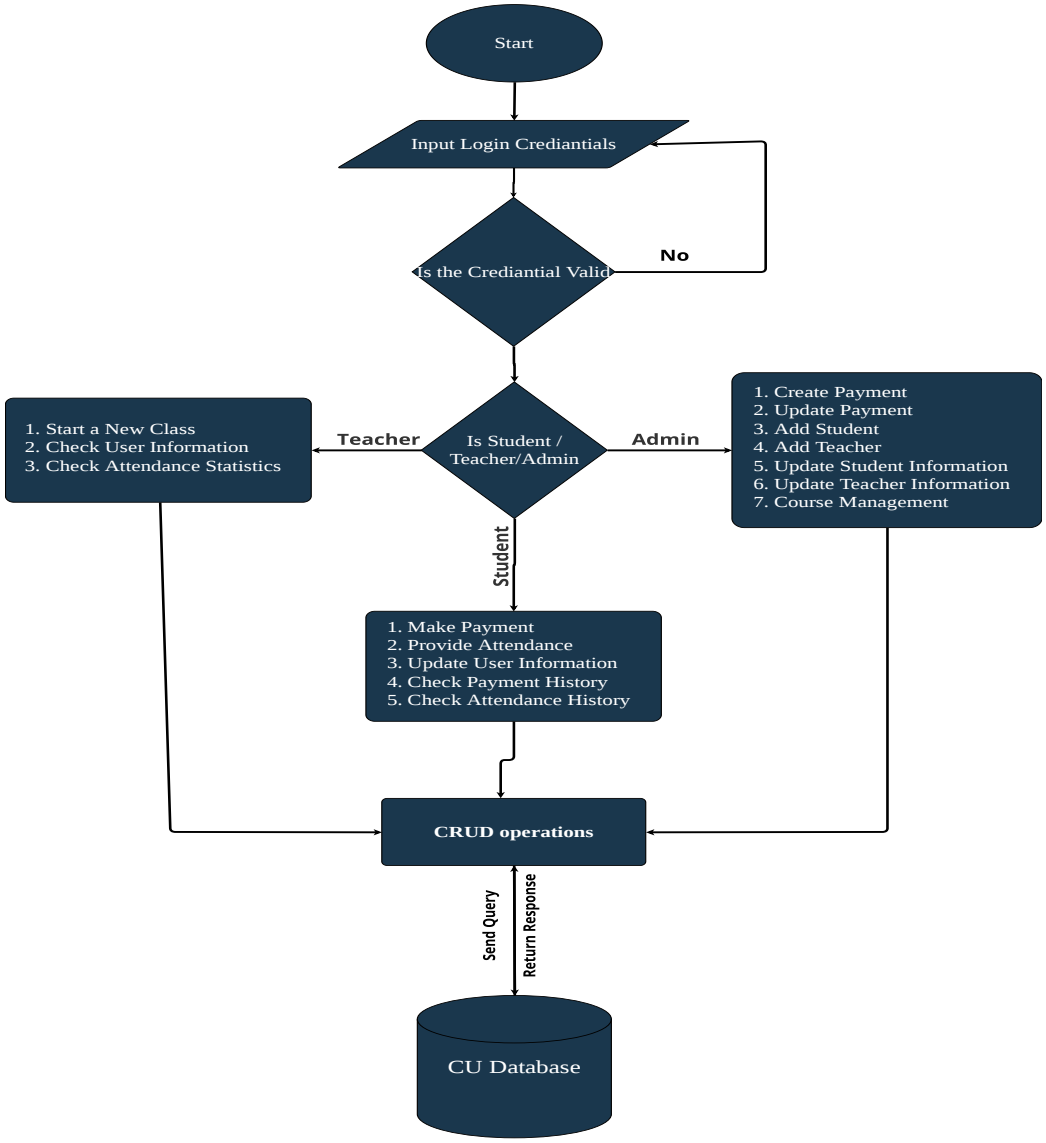

Figure 8: Flowchart of CU-OPAS

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

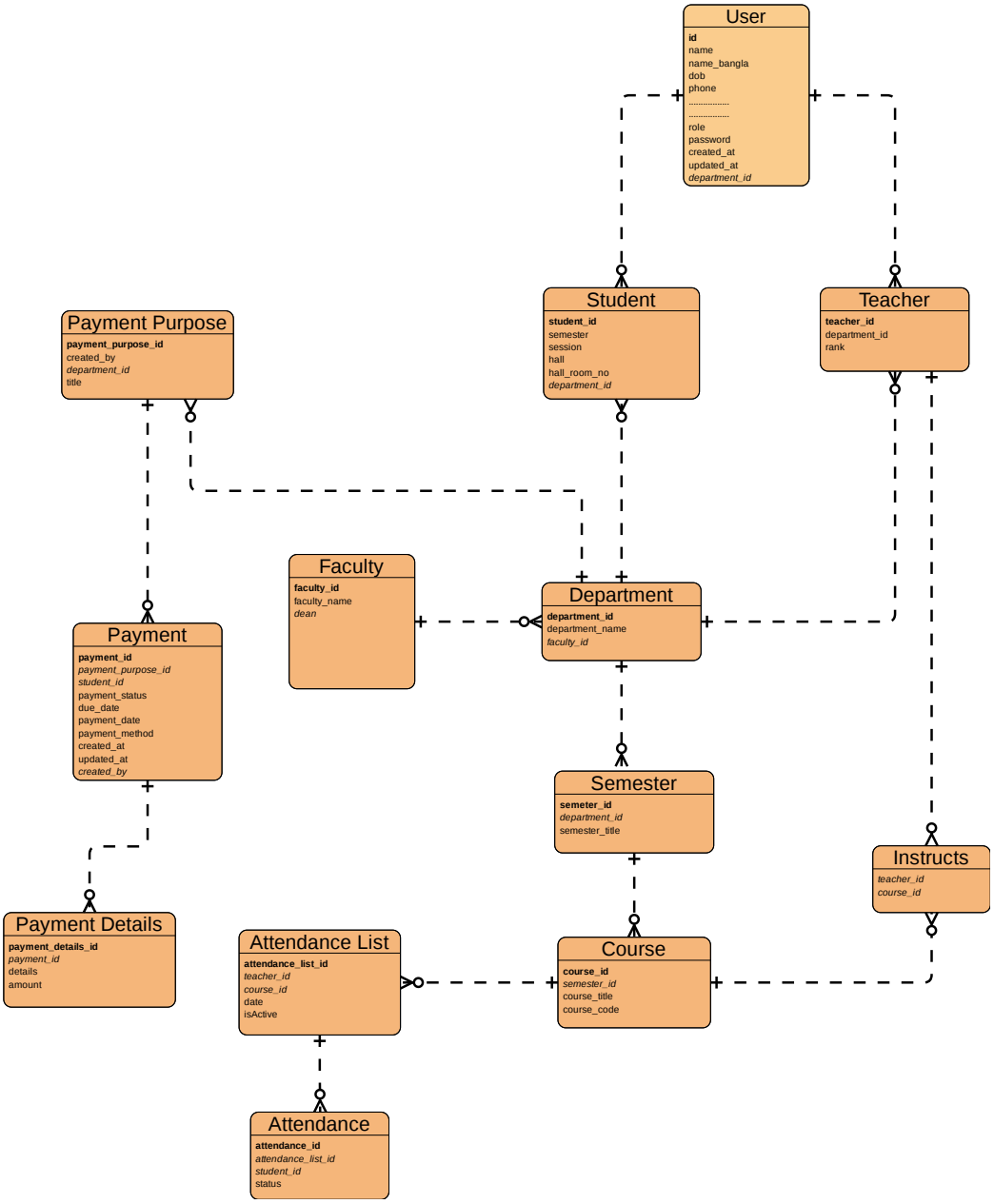

Figure 9: Entity Relationship Diagram of CU-OPAS

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

Data modeling is the practice of using words and symbols to represent data and how it flows in a simplified representation of a software system and the data pieces it includes. Data models serve as a roadmap for creating a new database or reengineering an existing one. A data model is a flowchart that depicts data items, their properties, and the relationships that exist between them. Before any code is developed, it allows data management and analytics teams to establish data needs for apps and uncover mistakes in development plans.
Data models can generally be divided into three categories, which vary according to their degree of abstraction. The process will start with a conceptual model, progress to a logical model and conclude with a physical model. We describe conceptual data modeling in this section.

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.

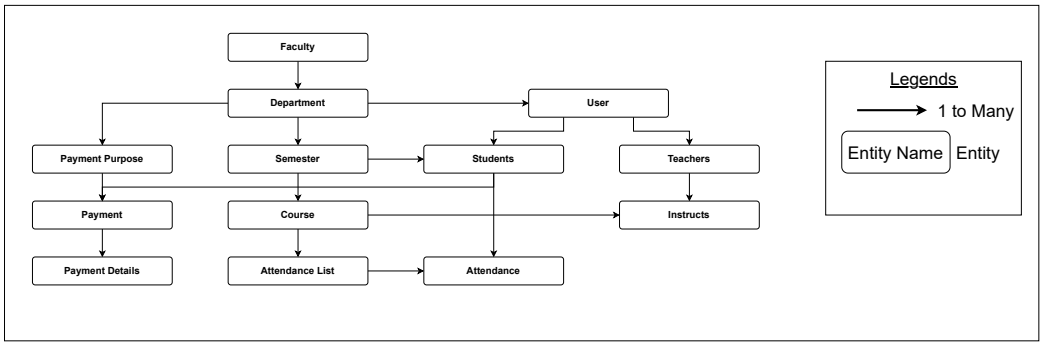

Figure 10: Conceptual Data Model of CU-OPAS

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.

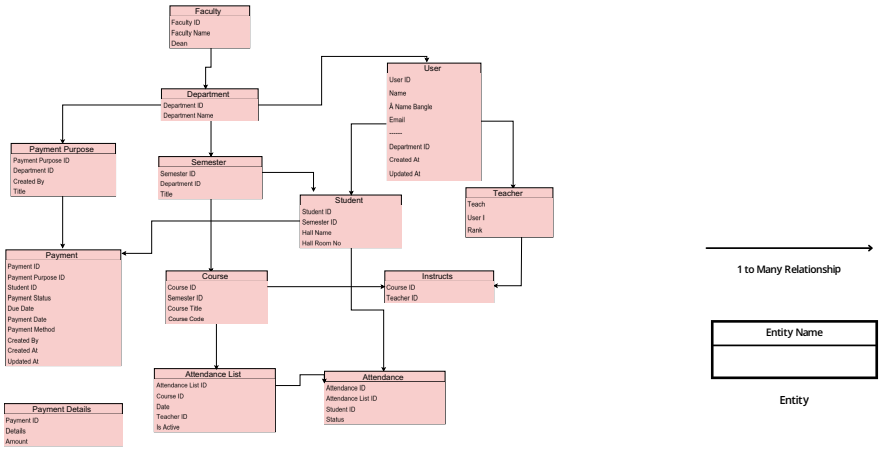

Figure 11: Logical Data Model of CU-OPAS

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

6 Normalization

Normalization is the process of structuring data in a database in order to eliminate data redundancy, insertion anomalies, update anomalies, and deletion anomalies. Normalization rules divide large tables into smaller tables and connect them using relationships. The purpose of normalization in SQL is to remove redundant (repetitive) data and ensure that the data is stored properly.
There are various database "Normal" forms. Each normal form has an importance which helps in optimizing the database to save storage and to reduce redundancies. Some of them are discussed in the following sub-sections.

6.1 Normal Forms

1. **1st Normal Form (1NF)**
   In this Normal Form, we address the issue of atomicity. In this context, atomicity indicates that the values in the table should not be further subdivided. To put it simply, a single cell cannot carry multiple values. The First Normal Form is violated when a table has a composite or multi-valued attribute. Therefore, a relation is in 1st Normal form if -

   - It contains only automic values.
   - Each Record needs to be unique and there are no repeating groups.

2. **2nd Normal Form (2NF)**
   The first criterion in the second NF is that the table be in the first NF. The table should also not contain any partial dependencies. In this case, partial dependence means that the appropriate subset of candidate keys determines a non-prime property. Therefore, a relation is in 2nd Normal form if -

   - It is in 1sr Normal Form.
   - There should not be any partial dependency of any column on primary key.Means the table have concatanated primary key and each attribute in table depends on that concatanated primary key.
   - All Non-key attributes are fully functionally dependent on primary key.If primary is is not composite key then all non key attributes are fully functionally dependent on primary key.

3. **3rd Normal Form (3NF)**
   The same criterion applies as previously, namely that the table must be

in 2NF before moving on to 3NF. Another requirement is that there be no transitive dependency for non-prime attributes. That is, non-prime attributes (those that do not constitute a candidate key) should not be reliant on other non-prime attributes in the same table. A transitive dependence is a functional dependency in which X $\rightarrow$ Z (X determines Z) indirectly, through X $\rightarrow$ Y and Y $\rightarrow$ Z (where Y $\rightarrow$ X is not the case). Therefore, a relation is in 3rd Normal form if -

- It is in Second normal form.
- There is no transitive functional dependency.

4. **Boyce-Codd Normal Form (BCNF)**
   BCNF The normal form is a more advanced variation of the third normal form. This type is intended to deal with analomies that cannot be dealt with in the third normal form. Dependencies between attributes belonging to candidate keys are not permitted in BCNF. It removes the limitation on non-key qualities from the third normal form.

   - In BCNF if every functional dependency A $\rightarrow$ B, then A has to be the Super Key of that particular table.

6.2 Normalizing the database into Boyce-Codd Normal Form

According to definition in 6.1 a relation is in BCNF iff (if and only if) every functional dependency A $\rightarrow$ B, then A has to be the Super Key of that particular relation. So, first of all, we let all the attributes are in a single relation $\boldsymbol{R}$ and we find all the functional dependencies.

For less complexity we are going to rename the attributes with uppercase letters here.

A = *user_id*
B = *name*
C = *email*
D = *role*
E = *student_id*
F = *session*
G = *hall*
H = *teacher_id*
I = *rank*
J = *faculty_name*
K = *dean*
L = *department_name*
M = *semester_name*

N = *course_name*
O = *course_code*
P = *purpose_title*
Q = *created_by*
R = *payment_details*
S = *payment_amount*
T = *due_date*
U = *payment_date*
V = *payment_method*
W = *class_date_time*
X = *class_is_active*
Y = *class_teacher*
Z = *attendance_status*

Now, ***R*** becomes as follows -

1. ***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 dependencies:
   {
   A → B C D, E → F G
   H → I, J → K
   N → O, P → Q
   R → S, W → X Y
   }

   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:

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

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

R_{15} (E, F, G)
FD:
{
$E \rightarrow F$
$E \rightarrow G$
}

$R1_{16}$ (A, E, H, J, L, M, N, P, R, T, U, V, W, Z)
FD:
{ }

R_1 (A, B, C, D)
FD:
{
$A \rightarrow B$
$A \rightarrow C$
$A \rightarrow D$
}

R_3 (W, X, Y)
FD:
{
$W \rightarrow X$
$W \rightarrow Y$
}

R_5 (R, S)
FD:$\{ \ R \rightarrow S \ \}$

R_7 (P, Q)
FD:$\{ \ P \rightarrow Q \ \}$

R_9 (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:
{
$A \rightarrow B$
$A \rightarrow C$
$A \rightarrow D$

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

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

R_1 (A, B, C, D)
{
$A \rightarrow B$
$A \rightarrow C$
$A \rightarrow D$
}

and:

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

In:

R_2 (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 R_2 is replaced by:

R_3 (W, X, Y)
FD:
{
$W \rightarrow X$
$W \rightarrow Y$
}

and:

R_4 (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:

R_4 (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 R_4 is replaced by:

R_5 (R, S)
FD:{ $R \rightarrow S$ }

and:

R_6 (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 R_6 is replaced by:

R_7 (P, Q)
FD:{ $P \rightarrow Q$ }

and:

R_8 (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:

R_8 (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 R_8 is replaced by:

R_9 (N, O)
FD:{ $N \rightarrow O$ }

and:

R_{10} (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:

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

the determinant of E $\rightarrow$ F is not a superkey and so R_{14} is replaced by:

R_{15} (E, F, G)
FD:
{
$E \rightarrow F$
$E \rightarrow G$
}

and:

R_{16} (A, E, H, J, L, M, N, P, R, T, U, V, W, Z)
FD:{}

The final results are:

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

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

R_{15} (E, F, G)
FD:
{
$E \rightarrow F$
$E \rightarrow G$
}

R_{16} (A, E, H, J, L, M, N, P, R, T, U, V, W, Z)
FD:{}

R_1 (A, B, C, D)
{
$A \rightarrow B$
$A \rightarrow C$
$A \rightarrow D$
}

R_3 (W, X, Y)
FD:
{
$W \rightarrow X$
$W \rightarrow Y$
}

R_5 (R, S)
FD:{ $R \rightarrow S$ }

R_9 (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.

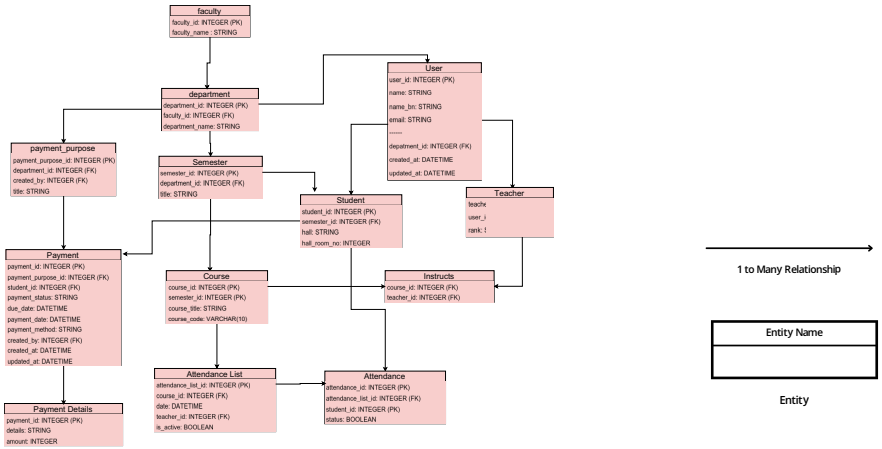

Figure 12: Physical Data Model of CU-OPAS

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*[1]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. API[2](*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.

[5]MySQL 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.

[6]An 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.

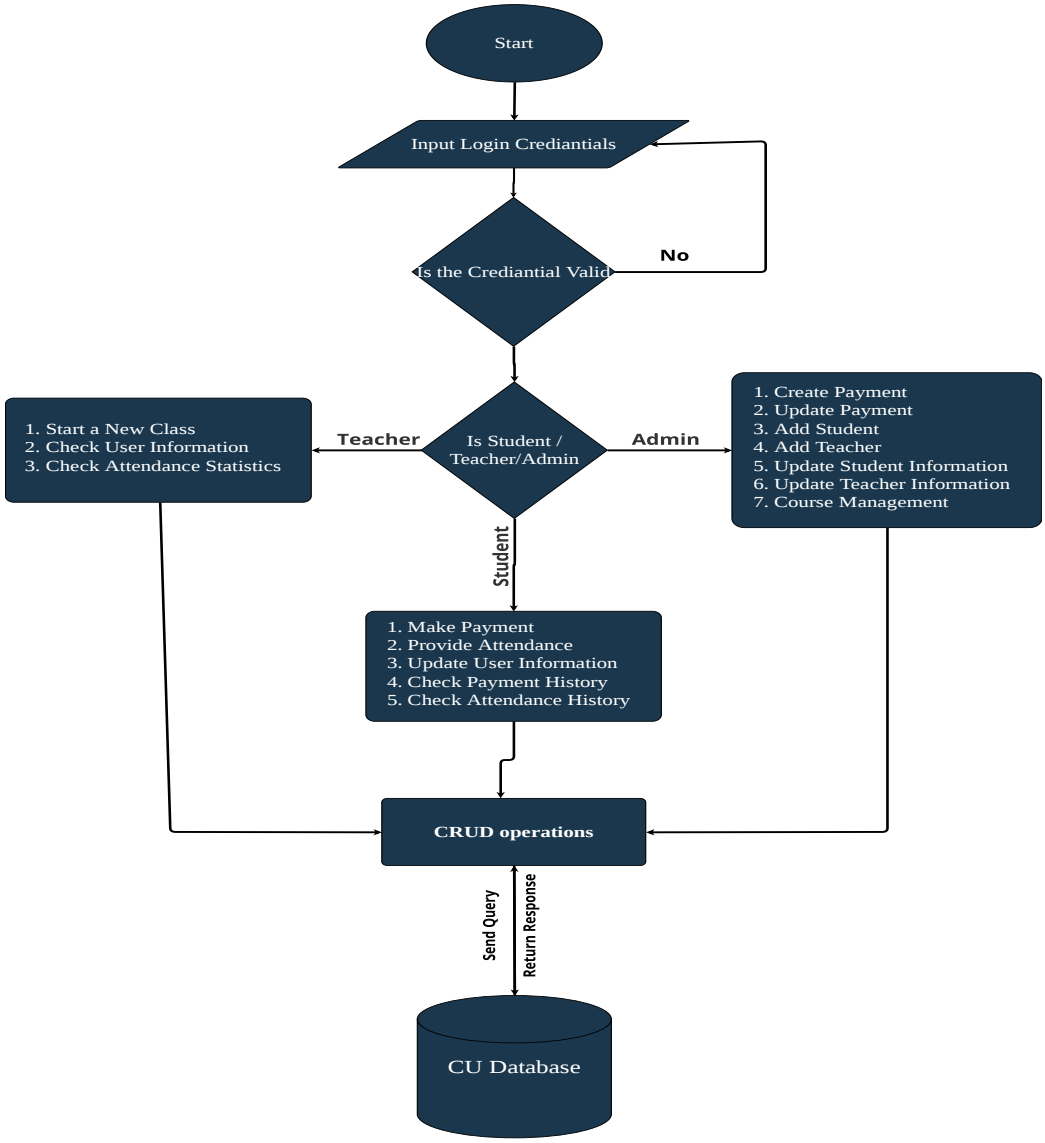

Figure 13: Work flow of the system

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).

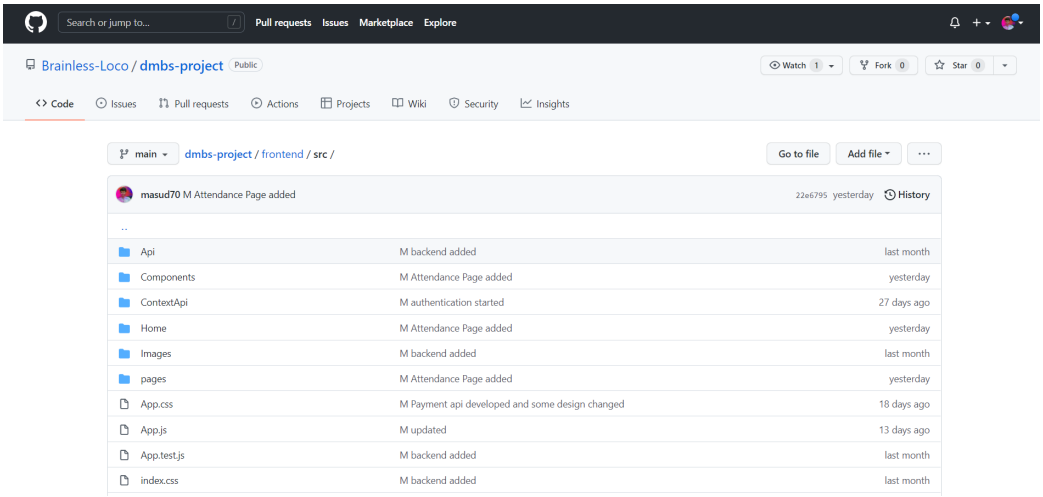

Figure 14: Github repository of CU-OPAS

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.

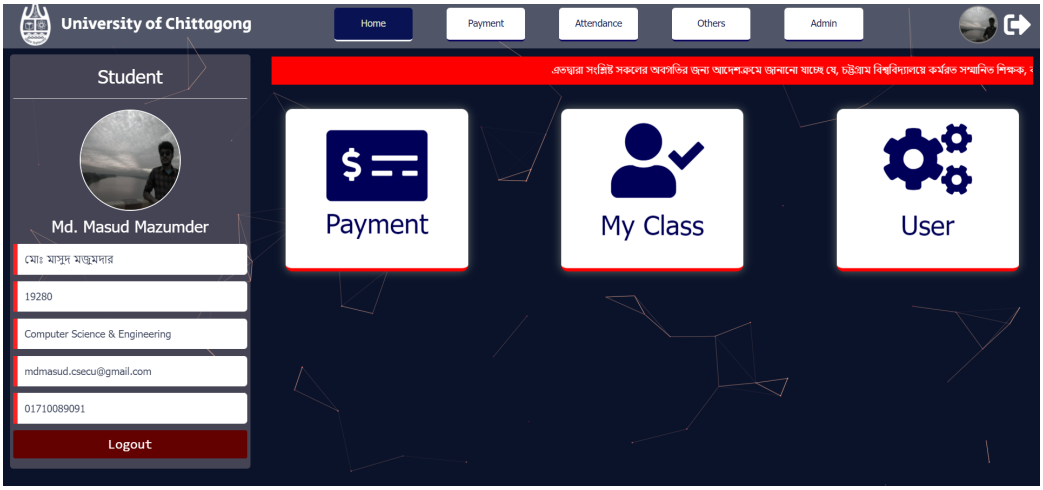

Figure 15: Home Page User Interface

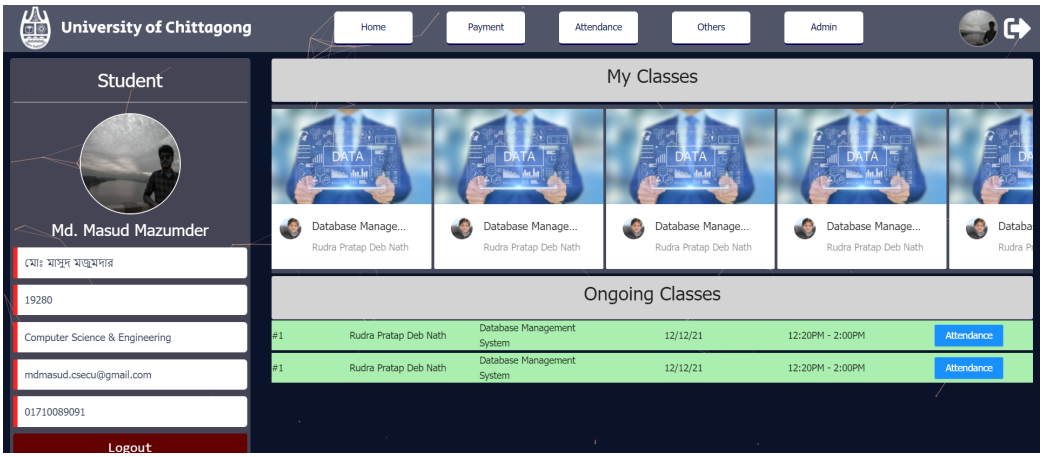

Figure 16: Student Attendance User Interface

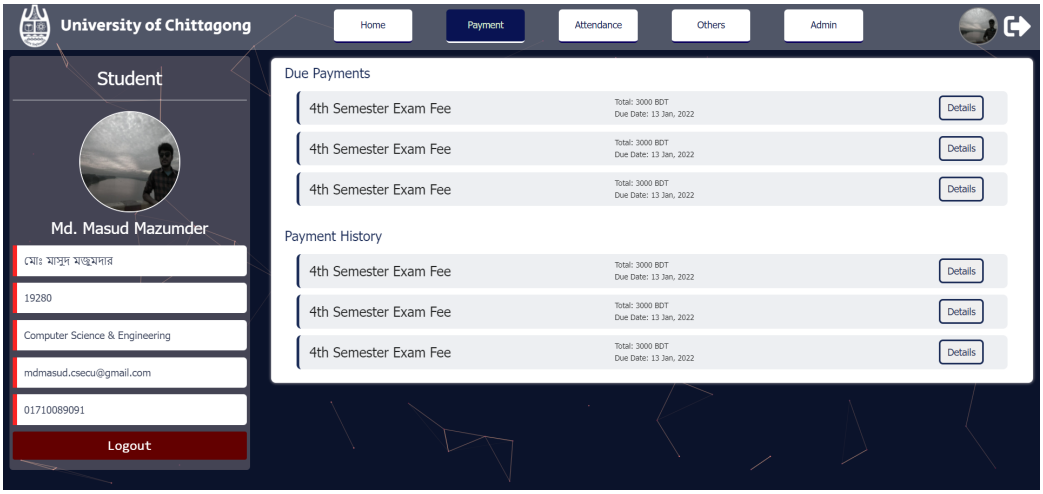

Figure 17: Payment History User Interface

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

Detals		**Amount**
Tuition Fee	-	350/-
Transportation Fee	-	320/-
BNCC Fee	-	300/-
Total	-	**970/-**

Figure 18: Payment Slip

Figure 19: Available Payment Methods

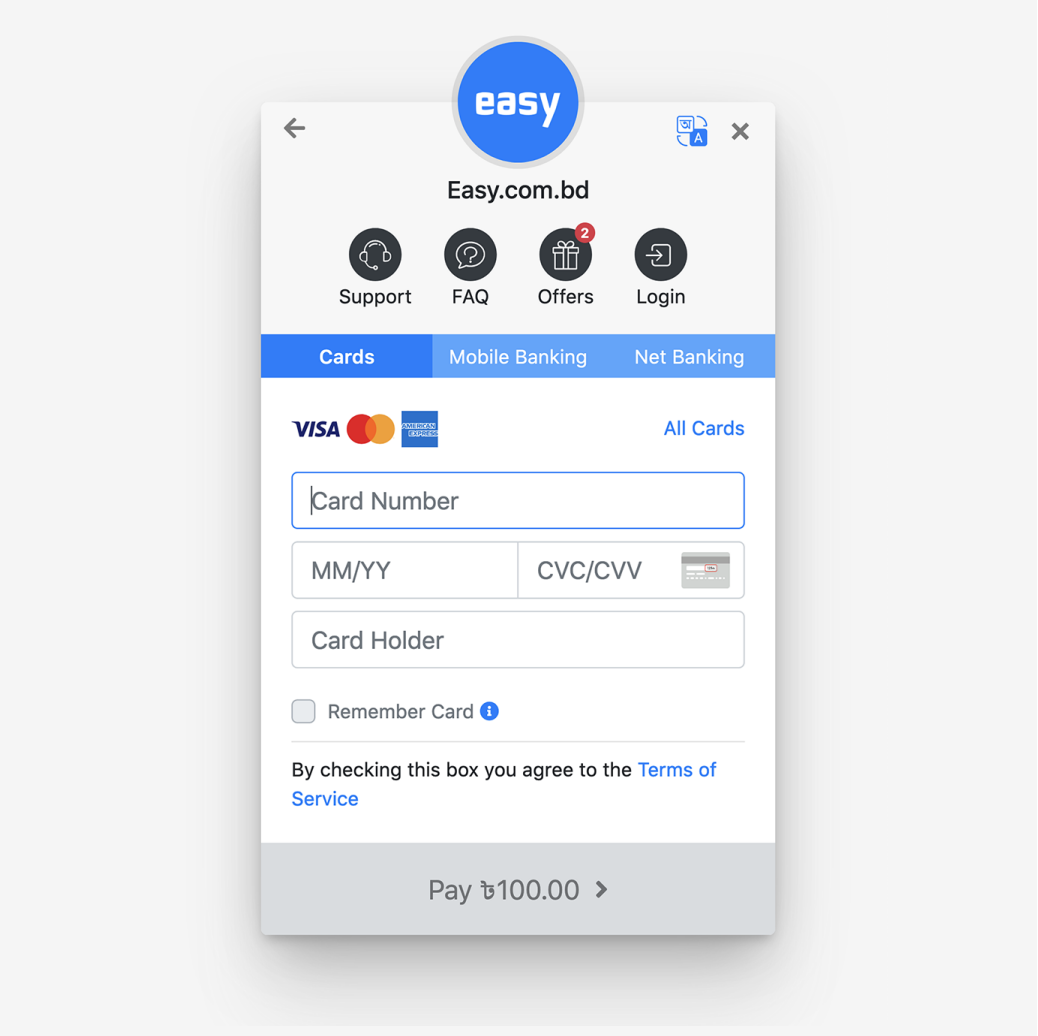

Figure 20: Payment Procedure

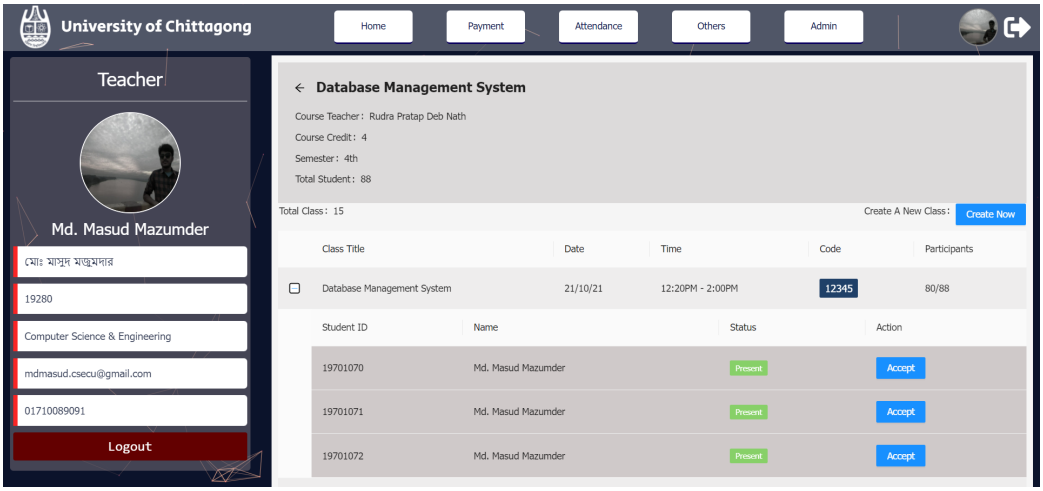

Figure 21: Teacher Attendance Interface

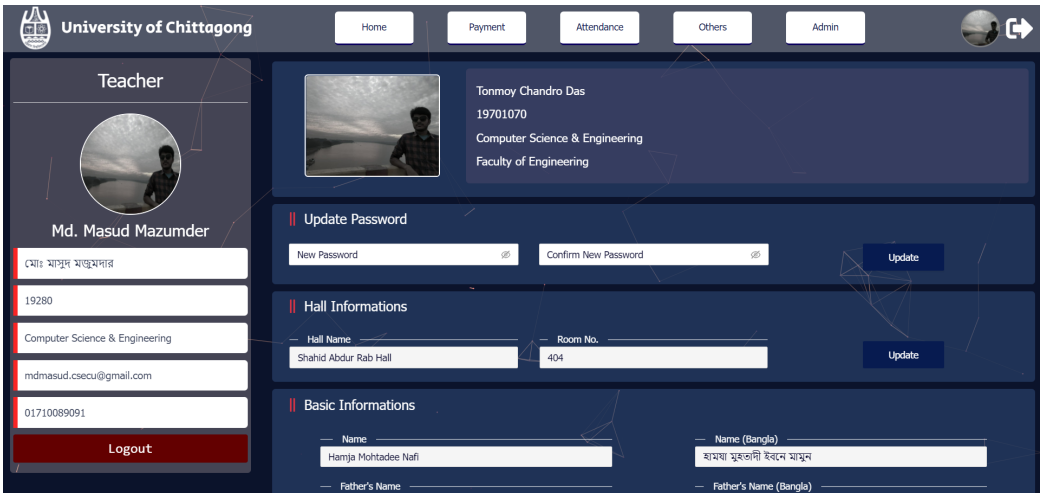

Figure 22: User Update Form

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,
```

```
    name_bangla,
    email,
    phone,
    dob,
    image,
    religion,
    father,
    mother,
    nationality,
    present_address,
    permanent_address,
    marital_status,
    PASSWORD,
STATUS
    ,
    role,
    department_name
FROM
    USER user_id = 101
```

Listing 1: A SQL query to find a User

```
SELECT
    student_id AS id,
    user_id,
    NAME,
    name_bangla,
    email,
    phone,
    dob,
    image,
    religion,
    father,
    mother,
    nationality,
    present_address,
    permanent_address,
    marital_status,
    PASSWORD,
STATUS
    ,
```

```
    role,
    SESSION,
    department_name,
    alloted_hall,
    semester_id
FROM
    student
NATURAL JOIN USER NATURAL JOIN
department WHERE user_id = 1
```

Listing 2: A SQL query to find a Student data

```
INSERT INTO payment_purpose(
    purpose_title,
    department_id,
    created_by,
    created_at
)
VALUES(?, ?, ?, ?)
```

Listing 3: A SQL query to insert payment purpose

```
SELECT
    *,
    (
    SELECT
        COUNT(*) AS total
    FROM
        student
    NATURAL JOIN semester NATURAL JOIN
    course WHERE course_id = ?
) AS total
FROM
    semester
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.

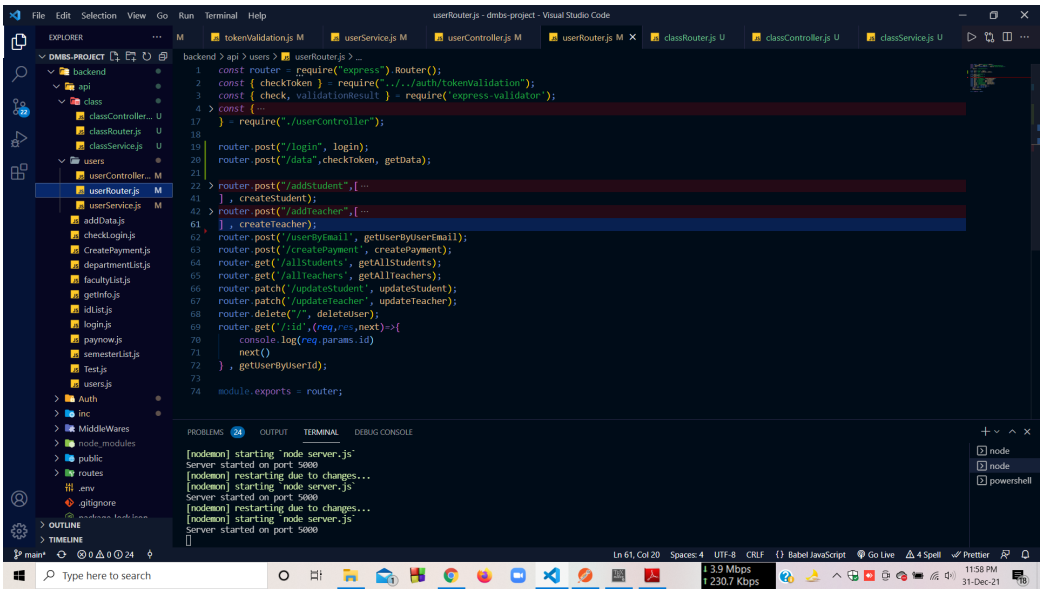

Figure 23: Backend API implementation (part-1)

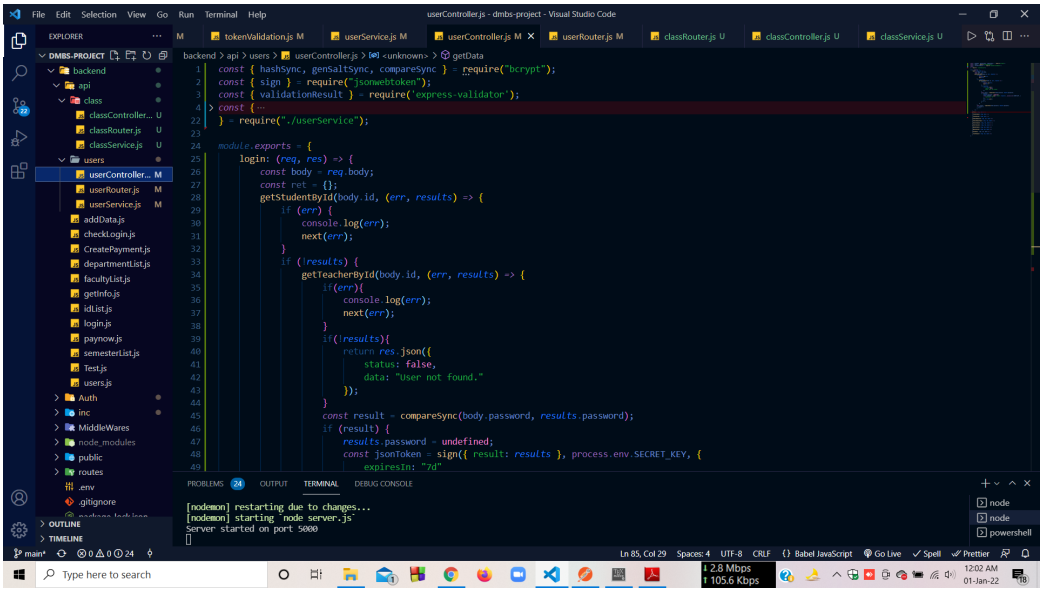

Figure 24: Backend API implementation (part-2)

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.