



---

# 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	Signature	Date	Supervisor Approval
19701070	Md. Masud Mazumder			
19701066	Tonmoy Chandro Das			
19701068	Tareq Rahman Likhon Khan			
19701040	Hamza Mohtadee Ibne Mamun			
19701019	Polash Hossen			
187010**	Nu-Sai Mong Marma			

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
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 . . . . .	9
<b>2</b>	<b>Project Management</b>	<b>11</b>
<b>3</b>	<b>Requirement Gathering and analysis</b>	<b>15</b>
3.1	Requirement Gathering . . . . .	15
3.1.1	Functional Requirements: . . . . .	15
3.1.2	Non-Functional Requirements: . . . . .	16
3.2	Requirement Analysis . . . . .	17
<b>4</b>	<b>Conceptual Modelling</b>	<b>21</b>
<b>5</b>	<b>Logical Modelling</b>	<b>22</b>
<b>6</b>	<b>Normalization</b>	<b>23</b>
<b>7</b>	<b>Physical Modelling</b>	<b>24</b>
<b>8</b>	<b>System Architecture</b>	<b>25</b>
<b>9</b>	<b>Implementation</b>	<b>26</b>
<b>10</b>	<b>Validation</b>	<b>27</b>
<b>11</b>	<b>Software Deployment</b>	<b>28</b>
<b>12</b>	<b>Conclusion and Future Work</b>	<b>29</b>
<b>13</b>	<b>Bibliography</b>	<b>30</b>

# List of Figures

1	Software Development Life Cycle (SDLC) . . . . .	7
2	Architectural design of CU-OPAS . . . . .	9
3	Context Diagram of CU-OPAS . . . . .	17

4	Flowchart of CU-OPAS . . . . .	18
5	Entity Relationship Diagram of CU-OPAS . . . . .	19
6	Conceptual Data Model of CU-OPAS . . . . .	21
7	Logical Data Model of CU-OPAS . . . . .	22
8	Physical Data Model of CU-OPAS . . . . .	24

## List of Tables

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

## Listings

1	A SQL query example . . . . .	26
2	A BibTeX example . . . . .	30

## **Abstract**

This project report is mainly focused on the online payment and attendance system of the University of Chittagong. In the university, numerous students pay all the university fees through bank drafts to the university's accounts in a specific bank branch which doesn't facilitate online systems. This analog payment system is not efficient enough especially during periods of examinations when most of the students are to pay examination fees. This process is characterized by long queues, too much waiting by the students, and congestion at banks where payments are made. On the other hand, manually attendance-taking still exists here which is excessively time-consuming. It was upon such background that we embarked on the project to develop an alternative system that enables online payment & attendance by students and their sponsors. Using this system every student will be comfortable with the online payment procedures. Also, it will save a lot of time by taking attendance online. The system was implemented using a JavaScript-based framework named "React" including Cascading Style Sheets for frontend and "ExpressJs" for backend, Apache web server, MySQL database server. System testing and validation were also done by allowing users of the system to interact with it using test data. Currently, the system is only limited to the payment and attendance system. In the future, this system can be expanded with various online systems like No Objection Certificate (NOC), Student Management System, Employee Management System, and so on. The result of the project is an online payment and attendance system for the University of Chittagong that provides relief of the long endured problems of the current modes of payment at the university as well as a time-consuming manual attendance system.

# 1 Introduction

One of the major benefits of using technologies is that it costs less time and instance in most of the cases to perform the same operation than with analog process. Conventionally we still perform all the official tasks through paper work. It not only consumes more time, but also makes students dependent on authority and on specific time period set by the authority. Also, there are always chances of being unsuccessful to complete the process with or without wrong information. In addition, taking attendance manually is also a time-consuming method. Our goal is to develop an application that aims to shift these two process into a digital platform which will obviously be able to perform the given tasks in the shortest time possible with no paper work. The system should be able to let the students get the liberty from the tormenting limitations of the administration and also to let the authority to make their responsibility easier. It goes without saying that the application would be a user-friendly application to both students and authority.

This document objectifies a recording of a strategic and creative process focused on clearly outlining issues, goals as well as overview of the application representing the narrative from the beginning to the end. Any person willing to use as well as develop the system would be able to do so by the help of this documentation.

The objective 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

Since 1966, the establishment of the university, each year the number of students are increasing. Whereas day by day, the world is being introduced with new technologies, the procedure to receive fees and attendance system remained same in University of Chittagong. Being dependent on one specific branch of an individual bank, the payment system of fees has become troublesome not only to the students but also to the administration. Students find this analog system dreadful while they are binded with a time period of a day to submit the fee during class days or a time near the examination. University Authority as well as bank authority have been seemed to be tired of this miserable process as the number of students has a positive growing rate each year. Same goes for the traditional attendance system. Teachers have been recorded to bestow 25 to 30 percentage of a class period to attest the attendance conventionally.

If we look closely, these problems are related to each other. We intend to develop a system that can solve these problem while processing with some basic data of both students and teachers. The system will requires no paper work and least possible physical presence of authority as well as students to perform the operations. Both payment and attendance system will become an one click process throughout the development of the application. Administrations will find it easier to query the needed information than ever before. Students as well as administration may operate the processes from anywhere at anytime with an instant time in this application.

## 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 used to handle the payment system of fees by the students and attendance system. The application should transform and shift the analog payment system into digital platform and prevent the time-consuming process.”*

### 1.4 System Development Process

There are several phases for a software development life Cycle (SDLC). Requirement gathering and analysis is the foremost and most important phase among them. During this phase, the client expresses the expectations of the project including who will use the product, how the customer will use the product, and the particular data included with any exceptional client requirements identified with the product. To accumulate prerequisites, project administrators ordinarily utilize a few methods. Interviews, surveys, observations, workshops are the most widely recognized of them. Determined to assemble requirements, we used to interview and questionnaire techniques.

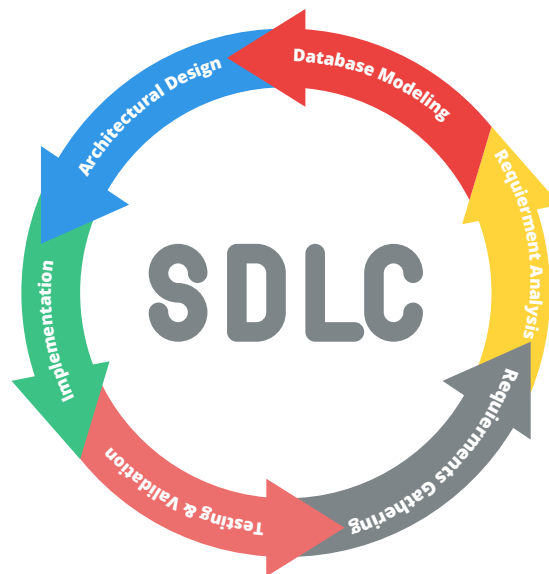


Figure 1: Software Development Life Cycle (SDLC)

Requirements investigation is critical and fundamental movement later requirement assembled. We break down, refine, and investigate the assembled requirements to make predictable and unambiguous necessities. This movement audits all requirements and may give a graphical perspective on the whole



framework. Later the consummation of the examination, it is normal that the understandability of the task might improve essentially. Here, we may likewise utilize the connection with the client to explain points of disarray and to comprehend which requirements are a higher priority than others.

Database modeling also called data modeling, is the process of creating a data model for the data to be stored in a database. It is the third phase of SDLC. This data model is a conceptual illustration of data objects, relationships between them, and the rules. The Data Model is characterized as a theoretical model that arranges information depiction, information semantics, and consistency limitations of the information. The information model underlines what information is required and how it ought to be coordinated rather than what tasks will be performed on information.

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.

System architecture is an illustrative design that depicts the structure of a system. It contains all the components of the system including subsystems that completely address the functionalities of the system. CU-OPAS is a web-based application system that works with request-response patterns. Using the frontend user interface, any authentic user of this system can request queries regarding payment and attendance operation to the backend API. Then the backend API interacts with the database server and generates a response according to queries that are sent back to the frontend.

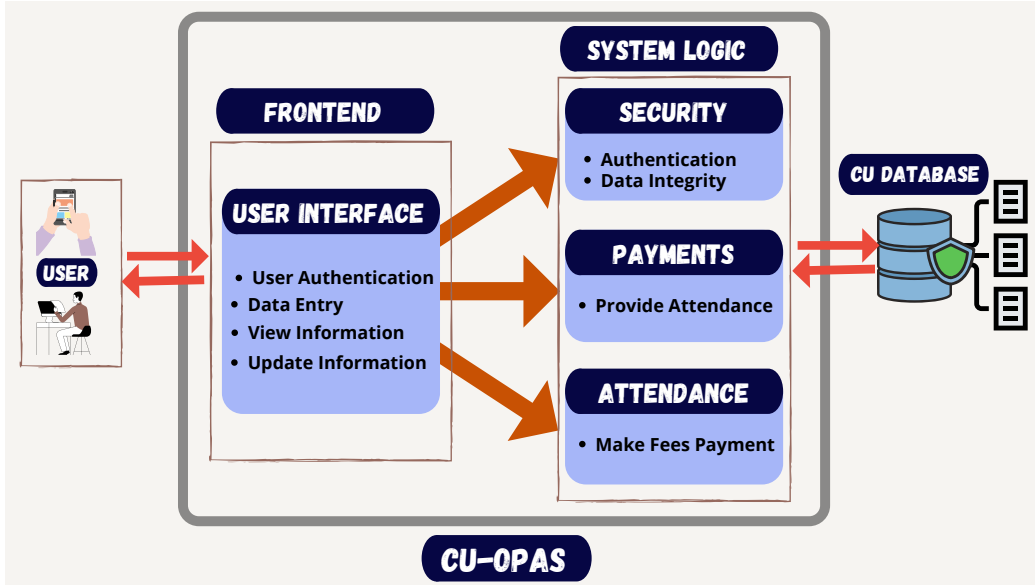


Figure 2: Architectural design of CU-OPAS

After successfully completing the architectural design and user testing, implementation begins. The physical design of the system takes place in this phase. It is the SDLC's third phase. During this phase, we used a variety of programming languages, frameworks, tools, and internet resources to build our system.

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

To implement the system we choosed web platform since it will be easier to use with any kind of devices. It will requires only a browser and internet to use the application and one can use it from anywhere at anytime he wishes to. The application will require a frontend which is to interact with it's user through browser and backend which will perform functionalities and some will make some operations into the database based on the query that is selected by the user.

Developing this application would requires a huge time if we would have wished to complete it individually. So we initially planned to split the entire project into certain branches with small or medium scaled tasks and distribute them among all of our teammates. After the completion we planned to assemble all the branches from our teammates and merge them together to build the final system successfully. It is clearly visible that we were moving forward with Scrum<sup>1</sup> method while completing the project. Following scrum model provision we made following 5 steps or phases mentioned below to execute our project.

### 1. Product Backlog Creation

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

If we think of a regular student being whose semester final exam is in a few days, has to pay the semester fee in a specific time period set by the administrator. This causes students suffering and directly effects on their preparation. Besides, in conventional way of transaction, university is fully dependent on one specific bank branch which causes an exhausting and tormenting situation when a huge number of students wishes to pay the fees on the same day. Many students even fail to submit the fee on time as they can't make any transaction on non-working hours. Teachers are also recorded to lose about one-third of the course period in taking attendance manually which effects on the concentration of students. On the other hand administration often find it tiresome to update data of students manually in papers. It not only consumes huge time, but also requires physical presence of a good number of person to complete.

These reasons influenced our team to develop a single application with the knowledge of our database course that is able to solve all of the issues mentioned above.

### 2. Sprint planning and creating backlog

The next stage was to do the sprint backlog creation for which our team had to select the important user stories and made them into smaller tasks. We made plans on how to get the task completed. Also, our team did prioritize the necessary tasks to accomplish the project more smoothly. The sprint duration lasted about 9 to 10 days which was long enough to allow the developer to work thoroughly.

### 3. Working on sprint

This was a practical phase. The actual stories were moved as small tasks in the sprint backlog where the actual work started. To begin with, a task board also called Kanban board in Trello<sup>2</sup> was made with a lot of cards in used. The cards used to specify the details about the tasks such as assignee, work details, due date or the time duration, etc. This cards also were used to record our meeting schedule and details as well as future updates. The task board consisted of the following columns the "To Do" lists, "Work In Progress" and then "Testing" and "Work Done" columns. A typical task board is shown in the diagram below.

In scrum method, a project is usually done through merging the contributions of a whole team being led a leader among them. In our case, Md Masud Majumdar was the leader among us, who planned and led the team from beginning to end. Each team member had contributed in parts of the entire 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 Meet<sup>3</sup> 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 finally done by Palash Hossen. Later it was modified by Nu Sai Mong Marma.

While developing the system, we used Github<sup>4</sup> 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 about the documentation of this entire project. This documentation was prepared under the leadership of Hamza Mohtadee.

### **Picture Thakbe**

## **4. Testing and Product Demonstration**

This phase is basically the modification phase based on the review of stack holders of our system. The tasks completed were to be realized as a working product with full life cycle testing. Every sprint that was completed was demonstrated to the stack holders i.e. students, teachers and concerned authority for their acceptance and their viewpoint on the complete solution. We received cordial reviews from the stack holders and in most of the cases they expressed their pleasures on the solution given by us. Being asked for suggestion, a part of them gave us some suggestion for modifications on several parts of the application. We obviously implemented many of those and integrated to the final system.

## **5. Retrospective and the next sprint planning**

The result of this step was to discuss what had gone well and what could be improved for the next level. Also we discussed the lessons learned and the pitfalls of any particular issues or problems related to this project. We planned our next sprint that is future work. We aim to integrate more functions into the system and withing a time, make it a reliable fully functioned university management system. With the current knowledge and more study, our team is quite confident to make that possible. This will results a paperless administration. All the official tasks will require less time than ever. In the meantime, we will assuredly maintain the system to make it an uninterrupted service provider application.

From the beginning of the planning to the final execution of the system, each and every member of our team has equally likely contributed to the project. We tried to make a better system than current one and it was finally

possible as together all of us went through hundreds of ideas. We tried to make the best use of current technologies to complete this project. During the progress, we learned a lot of new things while being stuck on problems we faced never before. Without proper project management principles, projects could be handled haphazardly and are at a much higher risk of project failure, delay in the project, and being over budget. Knowing the fundamentals of project management improved our chances of completing a project successfully.

---

<sup>4</sup>Scrum 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.

<sup>4</sup>Trello 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](http://www.trello.com)]

<sup>4</sup>Google 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](http://www.meet.google.com)]

<sup>4</sup>GitHub 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](http://www.github.com)]

## 3 Requirement Gathering and analysis

Requirement gathering and analysis is the foremost step of SDLC. In this step, we used some requirement gathering techniques like interviews, surveys, and observations. We have partitioned all of the requirements into two separate types which are functional requirements and non-functional requirements.

### 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

Post requirement gathering, the step is to analyze it. We have completed this step following the Structured Analysis Methodologies which focuses on the functional decomposition and data-flow analysis. After the brainstorming session of our team members, we come up with a context diagram.

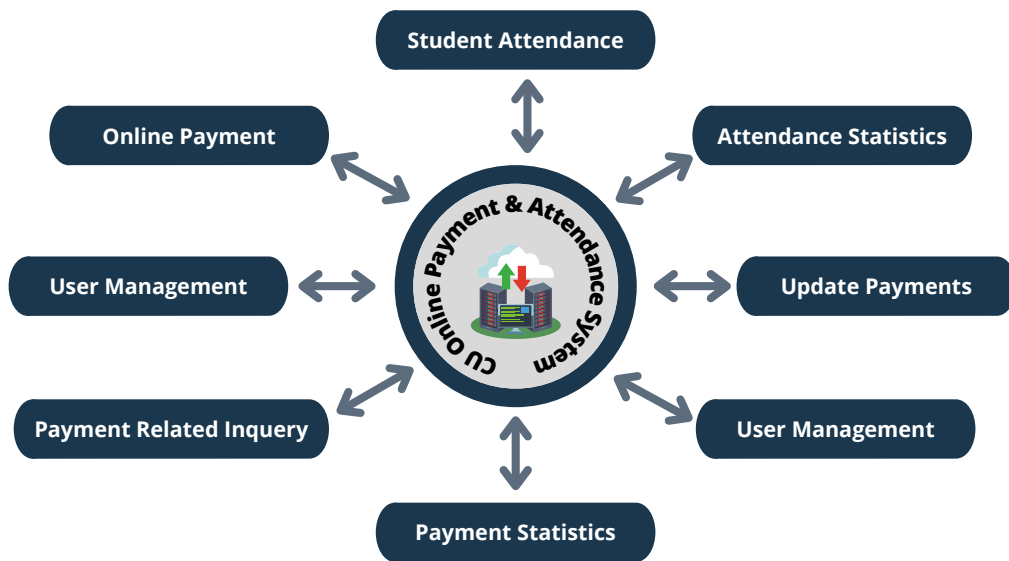


Figure 3: Context Diagram of CU-OPAS

We represented it to the authority as well as our supervisor. Then it was modified a little bit. Finally, we finalized a model of the system which includes all the requirements of administrative officials, teachers, and students which is ready to implement. Then, to define the flow of activities, we created a flowchart.

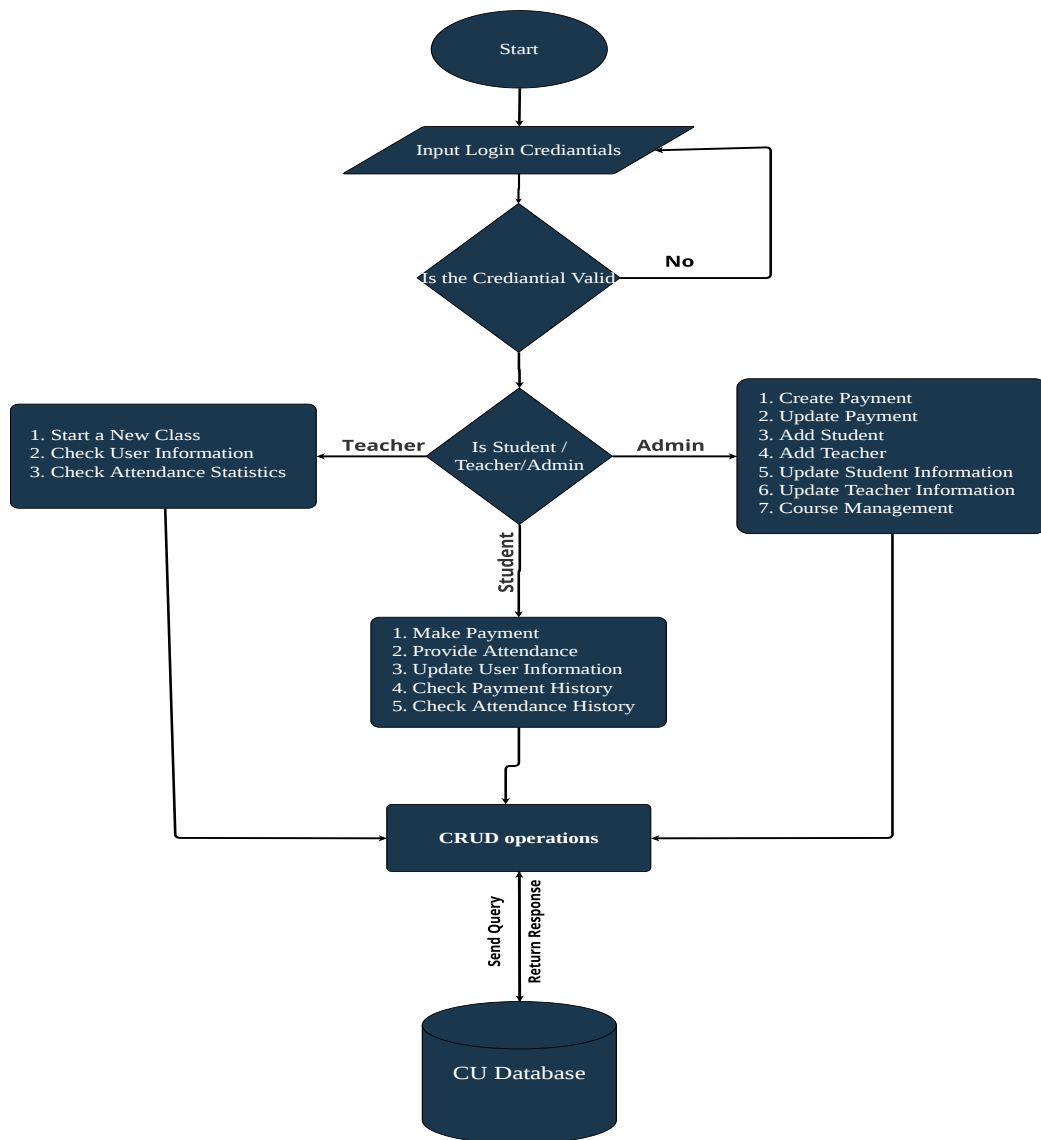


Figure 4: Flowchart of CU-OPAS

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

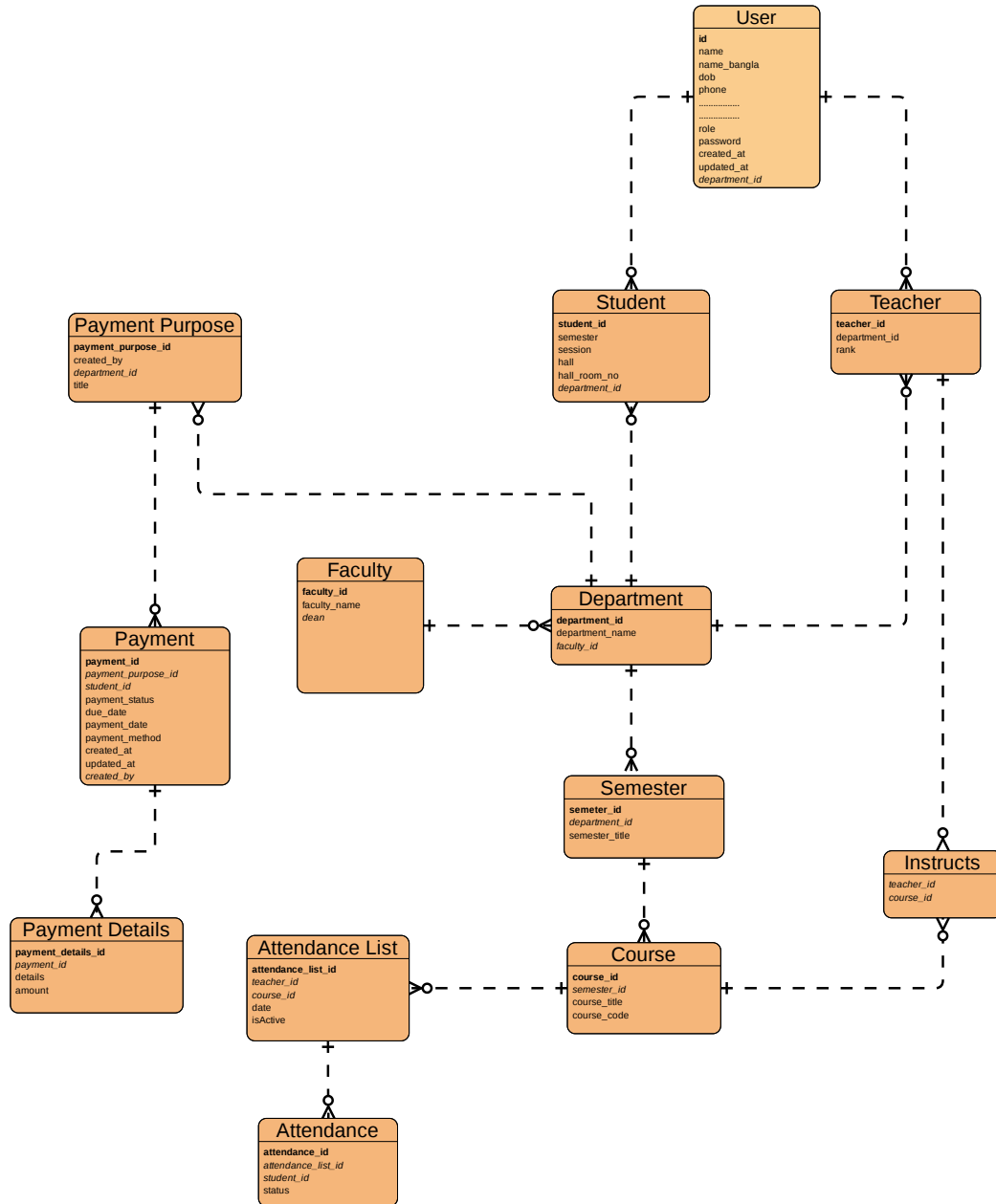


Figure 5: Entity Relationship Diagram of CU-OPAS

Stakeholders are the personnel connected to the project. They are concerned about the outcome of the project. Stakeholders can also be a group, company, customers, suppliers, etc. Stakeholders have a direct or indirect impact on the project. According to this stakeholders are two types i.e. Primary and Secondary. Supervisors, team members, teachers, administrative officials

are primary stakeholders, and students, media, other teams including their members are secondary stakeholders in this project.

## 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.

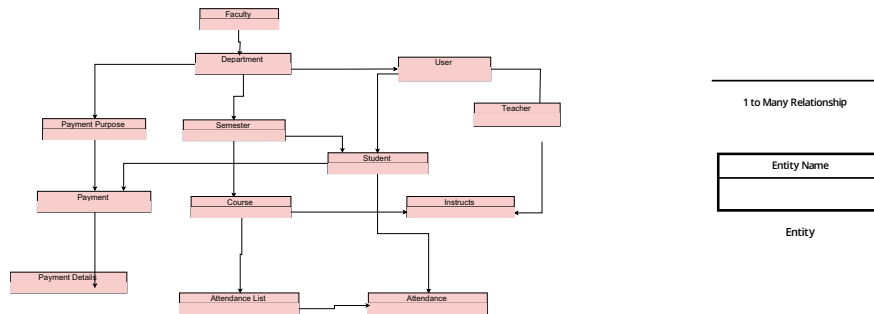


Figure 6: Conceptual Data Model of CU-OPAS

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.

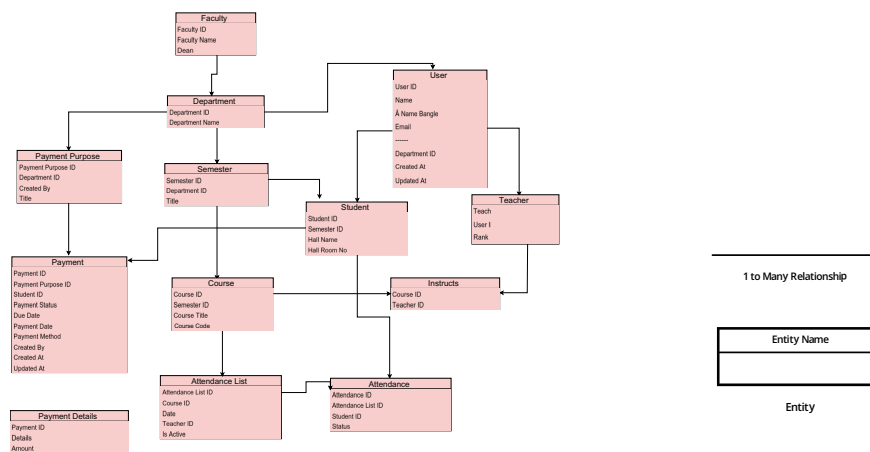


Figure 7: Logical Data Model of CU-OPAS

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

## 6 Normalization

From your Relational model, find the functional dependencies of each relation schema and show that they are normalized upto 3NF or BCNF.



## 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.

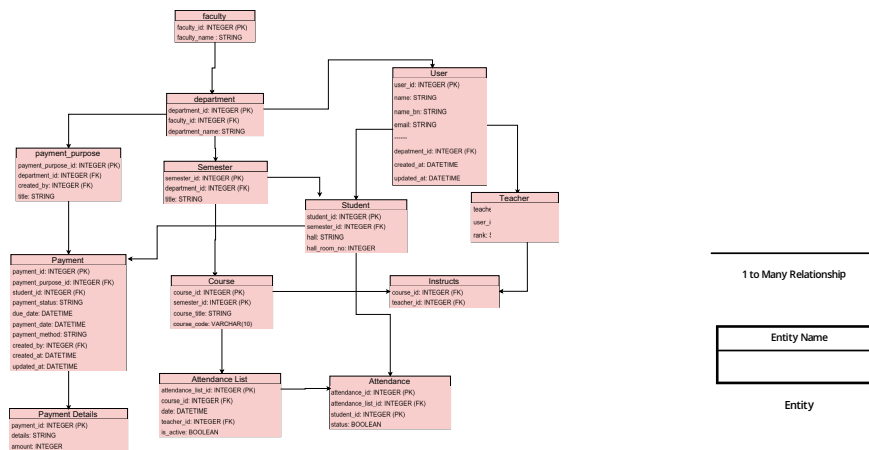


Figure 8: Physical Data Model of CU-OPAS

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

## 8 System Architecture

Describe the architecture of your system using a figure: Describe how each component of the architecture communicate.

## 9 Implementation

Give some code snippet of each component you outlined in your System architecture. Some DDL query example. Use the listing environment for writing code. Listing 1 shows an SQL query.

```
1 select distinct name
2 from instructor
3 where salary > some( select salary
4                       from instructor
5                       where dept_name='CSE');
```

Listing 1: A SQL query example

## 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.

## 11 Software Deployment

Describe how to install and configure your system so that a non-technical user can use your system.

## **12 Conclusion and Future Work**

Write the conclusion of your project: what was the problem? what the developed solution offers, Significance of the project, limitations of the project and future work.

## 13 Bibliography

To add bibliography in your document, use the following steps:

1. First create a .bib file in the same directory where your .tex file is (in our case, the file name is references.bib). Also place the bibliography style file in the same directory. In our case, we are using the ios1.bst style file. We include the following commands in the .tex file for the style file and bib file:

```
\bibliographystyle{ios1}  
\bibliography{references}
```

2. Import the BibTeX of your book or paper from Google Scholar or other sources into your .bib file. An example of BibTeX is shown in Listings 2.

```
1 @article{kopka1995guide ,  
2   title={A Guide to  $\LaTeX$ },  
3   author={Kopka, H and Daly, PW},  
4   year={1995},  
5   publisher={Citeseer}  
6 }
```

Listing 2: A BibTeX example

3. Then, use the name of the BibTeX (in Listing 2, the name is kopka1995guide) in the text of your .tex document where you want to refer it.
4. After saving your .tex document, execute the PDFLaTeX option one time; then execute the BibTeX option; then again execute the PDFLaTeX option for twice; finally, execute the QuickBuild option. Now your document refer the corresponding book or paper.