



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

1	Introduction	5
1.1	Background and Motivation	6
1.2	Problem Statement	6
1.3	System Definition	7
1.4	System Development Process	7
1.5	Organization	9
2	Project Management	11
3	Requirement Gathering and analysis	12
4	Conceptual Modelling	13
5	Logical Modelling	14
6	Normalization	15
7	System Architecture	16
8	Implementation	17
9	Validation	18
10	Software Deployment	19
11	Conclusion and Future Work	20
12	Bibliography	21

List of Figures

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

List of Tables

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

Listings

1	A SQL query example	17
2	A BibTeX example	21

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.

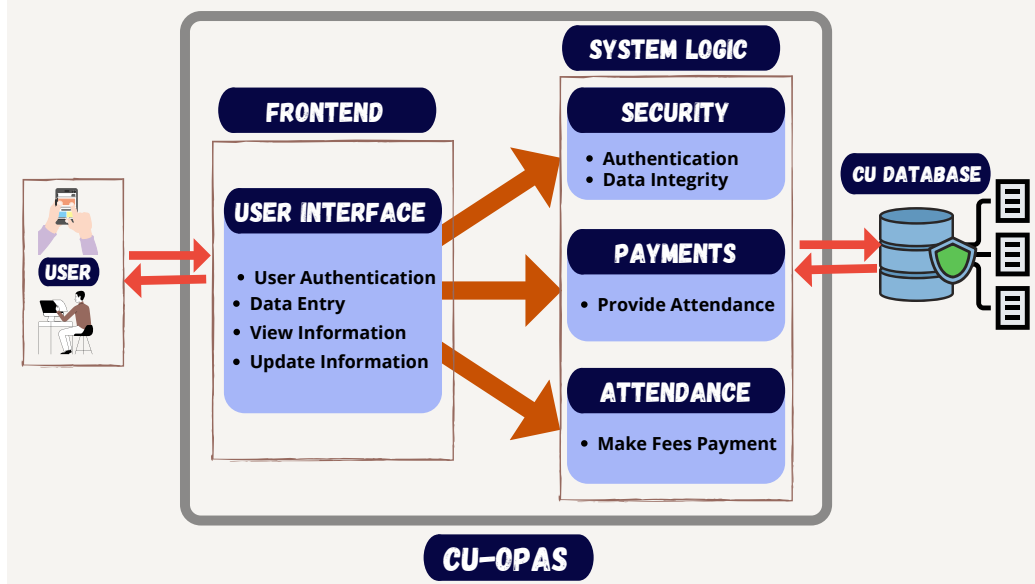


Figure 2: Architectural design of CU-OPAS

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 7 and section 8 gives the information about the whole system structure and how we implemented it. Section 9 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 10 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 11.

2 Project Management

Describe how the projects are resources are organized and managed in details, the roles of each members, used tools (Github, Trello ect.). See the scrum method.

3 Requirement Gathering and analysis

Explain how you gather the requirements of your problem: documentation, interviewing, survey, discussion, etc. Who are the stack-holders of your system?

4 Conceptual Modelling

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

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 System Architecture

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

8 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

9 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.

10 Software Deployment

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

11 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.

12 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.