

# **1 Introduction**

In many universities, students are still required to visit banks physically to pay their semester fees. This traditional method is time-consuming, inconvenient, and often involves long queues and manual paperwork. As a result, students waste valuable academic time and face unnecessary stress.

Additionally, most official university notices and updates are shared through Facebook groups. Many students miss important information because they do not regularly check social media, or important posts get buried among other content. This creates communication gaps between the university and students.

To solve these problems, an online system is proposed that allows students to pay fees digitally and receive all official notifications in one platform.

## **2 Overview of the System**

The proposed system is a web-based and mobile-friendly platform designed for university students and administrators. It will provide a centralized system for fee payment, academic document access, and official communication.

### **2.1 Student Features**

Students will be able to:

- Pay semester fees using digital payment services such as bKash and Nagad
- Receive automatic digital payment receipts
- Download admit cards online
- View official university notices and announcements

### **2.2 Administrator Features**

University administrators will be able to:

- Manage student payment records

- Verify transactions
- Upload notices and announcements
- Generate reports

The system will ensure secure transactions, accurate record-keeping, and easy access to important information.

### **3 Problem Statement and Analysis**

Currently, students must visit banks physically to pay semester fees, which causes long waiting times, transportation costs, and unnecessary stress. Manual payment systems are prone to errors, delays in verification, and loss of payment records. Students often face difficulties in collecting receipts and confirming successful payments.

Moreover, important university notices are mainly shared through Facebook groups. Many students miss these updates due to irregular social media usage, inactive accounts, or posts being buried under other content. This leads to misinformation, missed deadlines, and poor communication between students and the administration.

The absence of a centralized digital platform results in inefficiency, lack of transparency, and dependency on manual processes. Therefore, there is a strong need for an integrated online system that can manage fee payments and official notifications in a secure and organized manner.

## **4 Objectives and Scope**

### **4.1 Objectives**

The main objectives of this system are:

- To reduce the need for physical bank visits
- To save time and effort for students and staff
- To provide a secure and reliable online payment system
- To ensure that students receive all important notices on time

- To digitize administrative processes

## **4.2 Scope**

The scope of this system includes:

- Online fee payment using bKash and Nagad
- Automatic generation of digital receipts
- Admit card download facility
- Centralized notice board
- User authentication for students and administrators
- Payment and notice management system

The system does not cover academic grading, course registration, or learning management features in its initial version.

## **5 Problem Statement and Analysis**

Currently, students must visit banks physically to pay semester fees, which causes long waiting times, transportation costs, and unnecessary stress. Manual payment systems are prone to errors, delays in verification, and loss of payment records. Students often face difficulties in collecting receipts and confirming successful payments.

Moreover, important university notices are mainly shared through Facebook groups. Many students miss these updates due to irregular social media usage, inactive accounts, or posts being buried under other content. This leads to misinformation, missed deadlines, and poor communication between students and the administration.

The absence of a centralized digital platform results in inefficiency, lack of transparency, and dependency on manual processes. Therefore, there is a strong need for an integrated online system that can manage fee payments and official notifications in a secure and organized manner.

## 6 Requirement Analysis

Requirement analysis defines the services and constraints of the proposed system. It helps in understanding what the system should do and how it should perform.

### 6.1 Functional Requirements

The system must be able to:

- Allow students to register and log in securely
- Enable students to pay semester fees using bKash and Nagad
- Generate and store digital payment receipts
- Allow students to download admit cards
- Display all official notices and announcements
- Allow administrators to verify payments
- Enable administrators to upload, update, and delete notices
- Generate payment and activity reports
- Maintain student and transaction records

### 6.2 Non-Functional Requirements

The system should ensure:

- **Security:** Protection of user data and secure payment processing
- **Performance:** Fast response time and smooth transaction handling
- **Reliability:** Continuous availability with minimum downtime
- **Usability:** User-friendly interface for students and staff
- **Scalability:** Ability to handle increasing number of users
- **Maintainability:** Easy system updates and error fixing

## 7 System Design

System design describes the overall architecture of the system and explains how different components interact to fulfill the functional requirements. It provides a clear view of data storage, system behavior, and user interactions.

### 7.1 Database Design


The database is designed to store and manage all system-related data in a structured and secure manner. It ensures data integrity, consistency, and efficient access to information.

The main entities of the database include:

- **Student:** Stores basic student information such as ID, name, email, date of birth, and address.
- **Admin:** Stores administrator information including ID, name, and email.
- **Payment:** Stores payment details such as payment ID, amount, payment date, approval status, and associated student ID.
- **Receipt:** Stores digital receipt information generated after successful payment.
- **Notice:** Stores official university notices including title, content, posting date, and expiry date.
- **AdmitCard:** Stores admit card information such as exam name, exam date, exam location, and validity status.

Each entity is uniquely identified using a primary key. Relationships between entities are maintained using foreign keys to ensure referential integrity.

Field Name	Data Type	Key	Description
id	string	PK	Unique admit card ID
student_id	string	FK	References Student(id)
exam_name	string	—	Exam name
exam_date	date	—	Exam date
exam_location	string	—	Exam center
valid	boolean	—	Admit card validity

 Relationship: One Student — Many AdmitCards

Field Name	Data Type	Key	Description
id	string	PK	Unique receipt ID
payment_id	string	FK (Unique)	References Payment(id)
receipt_date	date	—	Receipt issue date
receipt_number	string	—	Receipt number

🔗 Relationship: One Payment → One Receipt

Field Name	Data Type	Key	Description
id	string	PK	Unique notice ID
admin_id	string	FK	References Admin(id)
title	string	—	Notice title
content	string	—	Notice content/detail
posted_date	date	—	Date of posting
expiry_date	date	—	Expiry date of notice

🔗 Relationship: One Admin – Many Notices

Field Name	Data Type	Key	Description
id	string	PK	Unique payment ID
amount	decimal	—	Payment amount
payment_date	date	—	Date of payment
approved	boolean	—	Payment approval status
student_id	string	FK	References Student(id)

🔗 Relationship: One Student → Many Payments

Field Name	Data Type	Key	Description
id	string	PK	Unique admin identifier
name	string	—	Admin name
email	string	—	Admin email

Field Name	Date Type	Key	Description
id	string	PK	Unique student identifier
name	string	—	Student full name
email	string	—	Student email address
date_of_birth	date	—	Student date of birth
address	string	—	Student address

Figure 1: Logical Database Schema

Figure below illustrates the Entity Relationship (ER) diagram representing the core entities of the system and their relationships.

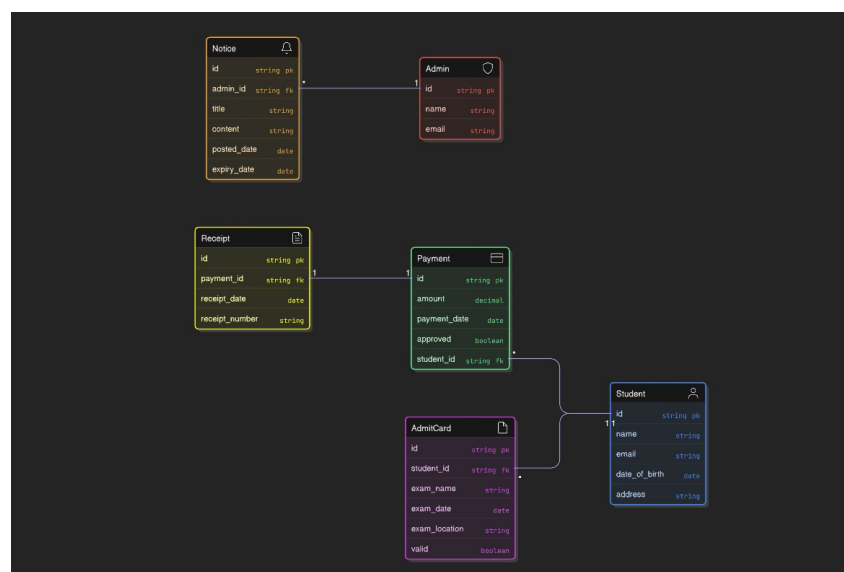


Figure 2: Entity Relationship (ER) Diagram