University of Ghana

College of Basic and Applied Sciences

School of Engineering Sciences

CPEN 208: Introduction to Software Engineering

First Semester, 2024/2025

Project Report: Department Software System

Student Name: [Your Full Name]

Student Number: [Your Student Number]

Course: CPEN 208 - Introduction to Software Engineering

Date: June 2025

1. Introduction

This report presents the design and implementation of a software system intended to assist the Department of Computer Engineering at the University of Ghana. The system provides functionalities to manage student information, handle fee payments, enroll students in courses, and assign lecturers and teaching assistants (TAs) to courses. The solution involves two major components:

1. A PostgreSQL relational database
2. A Next.js 14 web application with login, registration, and dashboard features.

2. Assumptions

- Each student has a unique student number.
- Students may enroll in multiple courses.
- Fee payment records include the total fee and amount paid.
- A course may have multiple lecturers and one or more TAs.

- Lecturers and TAs have unique emails and names.

3. Database Design

3.1 Database Name and Schema

- Database Name: cped_system

- Schema Used: cped

3.2 Tables and Relationships

- students: Stores personal information

- fees_payments: Records student payments

- courses: Course data

- course_enrollment: Student-course mapping

- lecturers: Lecturer details

- teaching_assistants: TA details

- lecturer_course: Lecturer-course mapping

- lecturer_ta: Lecturer-TA mapping

3.3 Data Types and Constraints

Keys are auto-generated using SERIAL. Foreign key constraints are in place.

4. Sample Data Insertion

The database was populated using a sample set drawn from the CPEN 208 class, including students, courses, payments, and staff entries.

5. Functions and Views

A PL/pgSQL function named get_outstanding_fees() was created to compute each student's outstanding balance. The result is returned as a JSON array including student_number, student_name, total_fee, total_paid, and outstanding_balance.

6. Web Application (Next.js 14)

6.1 Framework

Next.js 14, a React-based full-stack framework, was used.

6.2 Application Features

- /register: User registration
- /login: User login
- /dashboard: Displays data

6.3 Authentication

Implemented using next-auth.

6.4 Backend Integration

Connected to PostgreSQL using the pg library.

7. Deployment and Repository

7.1 GitHub Repository

Includes:

- Source code

- SQL scripts

- PostgreSQL backup

- PDF report

Repository URL: [Insert GitHub Link]

7.2 Database Backup

Created via pgAdmin and stored in the repo.

8. Challenges and Resolutions

- Handling relationships and constraints in SQL

- Managing data fetching and rendering in Next.js

- Authentication configuration with next-auth

9. Conclusion

The system is a scalable and functional academic management tool. Future improvements may include an admin panel, access control, and notifications.

10. References

- PostgreSQL Documentation: https://www.postgresql.org/docs/
- Next.js: https://nextjs.org/docs
- pgAdmin: https://www.pgadmin.org/docs/
- next-auth: https://next-auth.js.org/