HARIS AZIZ-036

iftikhar hussain-018

MUNEEB AHMED-042

**electricity billing system**

**Project Proposal: Smart Electricity Billing System with Advanced Analytics**

This proposal outlines a project for a "Smart Electricity Billing System" tailored to the requirements of the "Software Construction and Development (SCD) Lab" course. The project aims to build a maintainable Java application that not only automates the billing process but also incorporates advanced features like real-time data analysis and visualization, as highlighted in the benchmarking image provided.

**1. Project Goal & Objectives**

The goal is to design and develop an end-to-end "Electricity Billing System" using core software engineering principles and practices learned in the SCD lab. The system will manage customer data, record meter readings, calculate bills, generate invoices, and track payments. A key objective is to address code complexity, changeability, and reusability through object-oriented design and refactoring.

**2. Primary Features**

The system will be developed with a focus on a user-friendly and well-integrated interface.

* **Customer & User Management:** Implement a user-friendly interface with CRUD operations (Create, Read, Update, Delete) for customers and two roles: Admin and Operator.
* **Meter Reading & Billing:** A module to input and store meter readings and an engine to accurately calculate bills based on consumption, including taxes and late fees.
* **Invoice & Payment History:** The system will generate printable invoices and track payment statuses (Paid/Unpaid/Partial).
* **Real-time Analysis & Reports:** Implement an admin dashboard that provides real-time analysis of key metrics such as monthly revenue and outstanding payments.
* **Integration:** The application will have seamless integration with a SQL database for efficient data storage.

**3. Proposed Tech Stack**

The project will use technologies and tools specified in the course profile.

* **Language:** Java
* **IDE:** NetBeans
* **Database:** SQL Server
* **Tools:**
  + **Version Management:** GitHub for version control and software release management.
  + **Testing:** JUnit for unit testing.

**4. Weekly Action Plan (Aligned with Course Profile)**

This plan follows the "Weekly Course Outline" from the course profile to ensure all course activities and learning outcomes are addressed.

* **Week 1:** **Project Charter:** Finalize the project proposal and initiation document, defining the problem, scope, and objectives of the electricity billing system.
* **Week 2:** **Software Version Management:** Create the Git repository, establish branching policies, and set up the project skeleton with core dependencies.
* **Week 3:** **UML Modeling:** Create UML diagrams for the system, including class diagrams, use case diagrams, and sequence diagrams, for the core modules.
* **Week 4:** **OOP Concepts & Code Debugging:** Implement core OOP principles like abstraction and inheritance while developing the basic customer management module. Begin debugging to ensure the code is error-free.
* **Week 5:** **Code Refactoring:** Refactor the existing code for the customer and meter modules to improve its design and maintainability.
* **Week 6:** **Unit Testing using JUnit:** Implement JUnit tests for the billing calculation engine and other critical components to ensure correctness and stability.
* **Week 7:** **Design Patterns:** Apply design patterns like the Singleton and Factory patterns to improve the system's architecture and code quality.
* **Week 8:** **JSP & JDBC:** Design the first Java web page and establish its connectivity with the SQL Server database.
* **Week 9:** **Exception Handling:** Implement comprehensive exception handling mechanisms across the project, especially for database operations and user input.
* **Week 10:** **Software Configuration Management:** Manage project configurations and dependencies to ensure a consistent build environment for all team members.
* **Week 11:** **Integration Testing:** Perform integration and system-level testing to ensure all modules work together as a cohesive whole.
* **Week 12:** **Software Reviews:** Conduct personal and peer reviews of the project code to identify and fix any remaining issues.
* **Week 13:** **JAR File Creation:** Prepare the final project for distribution by creating a runnable JAR file.
* **Week 14:** **Final Project Demonstration:** Present the project, showcasing the final working system and its features.