

A
Training Report
On
ELECTRICITY BILLING SYSTEM

**Submitted for partial fulfillment of the award of degree of
Bachelor of Technology (B.Tech)
in
Computer Science & Engineering**



Submitted By:

Siya(2022026869)

Submitted to:

Department of Computer Science & Engineering

**Seth Jai Parkash Mukand Lal Institute of Engineering &
Technology (JMIT), Radaur**

Affiliated to Kurukshetra University Kurukshetra

ALMA INFORMATION TECHNOLOGY (I)

A Division of ALMA Limited (An ICAL Australia 3 Star Rated and ISO 9001:2015 certified Organization)
Member of all India IT Association (AIITA)



Ref:AMK/TRN/0807/2K24

Date: 10/08/2024

To Whom it May Concern

This is to certify that **Ms. Siya** student of B.Tech CSE under roll no 1222169 of JMIT RADAUR (YAMUNA NAGAR), has done her 6 weeks Practical Training during the period of July, 2024 to August, 2024 in the partial fulfillment of the Degree course.

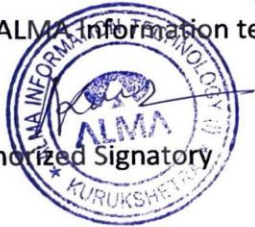
It is also certified that she learned **Programming in Java** in her training and has been completed the Project to the satisfaction of her Team Leader. It is to be noted that all the data used is solely entitled for the academic project only, no information should be copied/published anywhere without the consent of the company in any form.

She has been very sincere and her performance throughout has been good. Wishing her all the best in her professional career.

Sincerely

For ALMA Information technology (I)

Authorized Signatory



Workstation : SCF 1719/6, flyover Service Lane Jyoti Nagar , Red Road Kurukshetra-136118 (HARYANA) INDIA
Ph : +91 1744 358 134, +91 9 8125 7 7745, Email : kurukshetra@alma.in
Reg. Office : 18/3, "Alma House", Pardeshipura Nr. Electronic Complex, Indore (M.P.) INDIA
Ph : + 91 731 4055550, 4055551, 4055505, 4222242, 4222252, 4288812, 4290002
Fax : + 91 731 2573779 | Email : info@alma.in | www.alma.in



Declaration

I hereby certify that the work which is being presented in the Training Report entitled, “**Electricity Billing System**” by me, **Siya (2022026869)** in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Computer Science & Engineering at Seth Jai Parkash Mukand Lal Institute of Engineering & Technology (JMIT), Radaur** (Affiliated to Kurukshetra University Kurukshetra) is an authentic record of my own work carried out under the supervision of **Er. Ajay Mahn**.

Siya

Siya

2022026869

Acknowledgement

The writing of this training report has been assisted by the generous help of many people. I feel that I was very fortunate to receive assistance from them. I wish to express my sincere appreciation to them.

First and foremost, I am indebted to my principal supervisor, **Er. Ajay Mahn** of **ALMA** who has been very supportive at every stage of my preparations. I wish to express my utmost gratitude to him/her for his/her invaluable advice and patience in reading, correcting and commenting on the drafts of the report and, more importantly, for his/her generosity which I have received throughout my training program.

I would like to acknowledge and extend my heartfelt gratitude to **Dr. Gaurav Sharma**, Head of Department who provided me the golden opportunity to undergo and complete this training program.

I wish to express my thanks to **Er. Rajiv Bansal** who also helped me in conducting this study and encouraged me throughout this journey.

Finally, I am particularly indebted to my dearest parents/guardians as without their generous assistance and love; this dissertation could never have been completed.

Siya

2022026869

Preface

Objectives of the Training:

The objective of the Java training program was to equip participants with a robust understanding of Java programming fundamentals and practical application skills. This training covered core concepts such as object-oriented programming (OOP), inheritance, polymorphism, encapsulation, and abstraction, which are essential for building scalable and maintainable applications. Additionally, the training delved into advanced topics, including exception handling, collections framework, multi-threading, file handling, and database connectivity, providing a well-rounded foundation in Java.

The training was designed to enhance participants' problem-solving abilities, with a focus on writing efficient, clean, and optimized code. Hands-on projects and coding exercises enabled participants to implement learned concepts, reinforcing theoretical knowledge through practical application. The curriculum also included debugging techniques and best practices to develop high-quality, error-free applications.

The end objectives of the training are to obtain an In-depth knowledge to:

- Understand Core Java Concepts: Gain a solid grasp of fundamental Java programming concepts, including data types, variables, and control structures.
- Learn Object-Oriented Programming (OOP): Master OOP principles such as inheritance, encapsulation, polymorphism, and abstraction to design structured and reusable code.
- Explore Java Libraries and APIs: Familiarize with Java's standard libraries and APIs, including Collections, for efficient data handling and manipulation.
- Handle Exceptions: Understand and implement exception handling to manage runtime errors and ensure program stability.
- Implement Multi-Threading: Learn multi-threading techniques to develop applications that perform multiple tasks simultaneously, improving efficiency.
- Develop File I/O Skills: Acquire knowledge in reading, writing, and managing files, crucial for data processing and storage.
- Connect with Databases: Learn how to use JDBC to connect Java applications to databases, allowing for effective data storage and retrieval.
- Enhance Problem-Solving Abilities: Strengthen logical thinking and problem-solving through real-world coding exercises and projects.
- Follow Debugging and Best Practices: Develop debugging skills and follow coding standards for writing clean, maintainable, and optimized code.

Table of Contents

Certificate of Training (Photo Copy Only).....	i
Declaration.....	ii
Acknowledgement	iii
Preface (Objective of the Training)	iv
Table of Contents.....	v
Profile of the Company & Training Division	vi
Scheduled Chart of Training Activities	vii
List of Figures.....	viii
List of Tables.....	ix
List of Abbreviations.....	x

Chapters (As per the actual):

- Chapter 1:** Introduction
- Chapter 2:** Academic & Technical Importance of the Learning/Observed Sequence
- Chapter 3:** Detail Study of Processes/ Protocols/ Methods/ Languages/ Case Domain
- Chapter 4:** Proposed System Architecture and Design of the Problem (Flow Charts/ DFDs, if any)
- Chapter 5:** Prototyped Module Implementations (Pseudo-codes/Algorithms) / Inspection Results
- Chapter 6:** Future Scope & Limitations
- Chapter 7:** Learning Outcomes & Conclusion
- Chapter 8:** Bibliography (No referencing to Wikipedia) in Proper Formatting
- Chapter 9:** Appendix (containing code/ algorithms along with the Results/ Screen Prints (If any) *

Company Profile

A heritage of over 15 years in the services, starting from 1999, Alma is one among the pioneers in this field with over 0.7 millions satisfied customers & over one Hundred IT Workstations to its credit. Our customers' experience is through our people and their performance, which result in an outstanding track record of success. These are the achievements that influence customers to trust us. The relationships which cause our customers to return to us year after year are built by our people. A heritage of over 15 years in the Services, starting from 1999, Alma is one among the Pioneers in this field with over 0.7 millions satisfied Students & over one thousand IT Learning Centres to its credits. Alma also offers many tailor made computer courses that are part of International Certification from Mauritius and Australia. It also has ISO and ICAL certifications for providing quality services to the user.

Alma also offers many tailor made computer courses that are part of International Certification from Mauritius and Australia. It also has ISO and ICAL certifications for providing quality services to the user. Alma operates through multi-locations offices in India and international offices across the world. The institute provides a rigorous, intellectually challenging curriculum with flexibility to allow students to tailor their education to meet their professional and personal interests and goals.

Alma has emerged as one of the best source of technical education in India in just ten years. It provides a collaborative learning environment that weaves technology and management into everyday life. Addition to that Alma serves specialized courses in UG & PG of reputed universities and international certificate in IT & Management to the deserving scholars.

Alma is known for philanthropic approach among youth, offering them all computer courses in nominal fee-structure. It provides a collaborative learning environment that weaves technology and management into everyday life. Addition to that Alma serves specialized courses in UG & PG of reputed universities and international certificate in IT & Management to the deserving scholars. We actively explore and develop new methodologies such as alternative teaching techniques and learning approaches that promote thinking skills and lifelong learning habits.

To spread over the Computer Literacy and Computer Education throughout the country especially in the backward and progressive countries and development of Information Technology Exchange Programs over the world. Alma has successfully provided IT services to many of the world's most reputed enterprises. Satisfy the high standards of our clients for quality, service and delivery.

TRAINING SCHEDULE

w.e.f. July 10, 2024

Time Duration	Contents
Introductory Week (11-07-2024 to 12-07-2024)	Introduction of java , Concepts of JVM ,Keywords ,Constant ,Variables and data types ,Operators and expressions, Control statements, Conditional statements, loops and iterations , Class definition, Creating objects.
WEEK-1 (14-07-2024 to 20-07-2024)	Constructors ,Defining and calling methods, Method overloading, Creating an array, one and two dimensional array, string array, Wrapper classes, Packages and interfaces ,Exception handling, Swing, Applet, Icons and labels, Textfields, Buttons and Button class.
WEEK-2 (21-07-2024 to 27-07-2024)	Checkbox, Radio buttons, The container, Panel ,Windows, Frame classes, Combo box, Tabbed panes, Scroll panes, Trees ,tables, Introduction to servlets ,Life cycle of servlets ,Creating and running servlet ,Reading the servlet parameter, Reading initialization parameter.
WEEK-3 (28-07-2024 to 3-08-2024)	Packages-javax.servlet package, Handling HTTP, Request and Response(GET/POST request),Cookies and session tracking.
WEEK-4 (4-08-2024 to 10-08-2024)	Advance java: Collection, list, map ,tree, hashing JDBC: JDBC fundamentals ,Establishing Connectivity and working with connection interface, working with statements ,Creating and Executing SQL statements, working with Result set object & Result set meta data.
WEEK-5 (11-08-2024 to 17-08-2024)	Project on electricity Billing System, Splash class, Main class, Signup class, Login class ,Bill Details class ,Calculate bills class.
WEEK-6 (18-08-2024 to 24-08-2024)	Deposit bill class, database class, meter info class, pay bill class, payment bill class, generate bill class, update information class, view information class, completion of project.

List of Figures

Figure	Title of the figure
Figure 1.1	Intoduction of JAVA
Figure 2.1	Technical importance of JAVA
Figure 3.1	Processes in JAVA
Figure 3.2	Protocols in JAVA
Figure 4.1	TCP/IP Model
Figure 6.1	Future Scope
Figure A.1	DFD of Electricity Billing System (zero level)
Figure A.2	DFD of Electricity Billing System (first level)
Figure A.3	DFD of Electricity Billing System (second level)

CHAPTERS

Chapter-1: Introduction

This chapter introduces the objectives, scope, and background of the Java training program. Java, a versatile and widely-used programming language, is the foundation of many enterprise applications and digital platforms. The purpose of this training was to gain in-depth knowledge of Java programming concepts and practical skills for developing robust applications. Java's platform independence, strong object-oriented principles, and rich libraries make it essential for building applications across various domains, from web development to mobile applications.



Figure 1.1 : Introduction of JAVA

Training program aimed to equip learners with a comprehensive understanding of core Java fundamentals, covering topics such as variables, data types, control structures, and object-oriented programming (OOP) principles. Additionally, the training included hands-on exercises with Java's extensive libraries and APIs, such as the Collections Framework, which are essential for efficient data handling. This approach emphasized writing clean, modular, and efficient code aligned with industry standards.

Throughout the training, a combination of theoretical and practical sessions allowed learners to develop and implement concepts. The course culminated in the creation of prototype modules, enabling participants to apply their learning in real-world scenarios.

The Introduction chapter thus serves to outline the rationale behind choosing Java, the scope of the training, and the skills targeted for development. This foundational knowledge in Java prepares trainees to confidently tackle complex programming challenges, setting the stage for more advanced learning in Java frameworks and enterprise application development.

Chapter-2:Academic & Technical importance of Learning/Observed Sequencee

The Java training program is of considerable academic and technical importance due to its relevance in both foundational programming education and industry applications. Academically, Java is a language that reinforces core programming concepts such as data types, control structures, and object-oriented principles like inheritance, polymorphism, and encapsulation. Learning Java provides a solid basis for understanding advanced programming paradigms, making it an essential part of a computer science curriculum. Java's syntax and structure make it accessible for beginners, while its extensive libraries and robust APIs allow students to explore more complex topics, such as multithreading, data management, and exception handling.

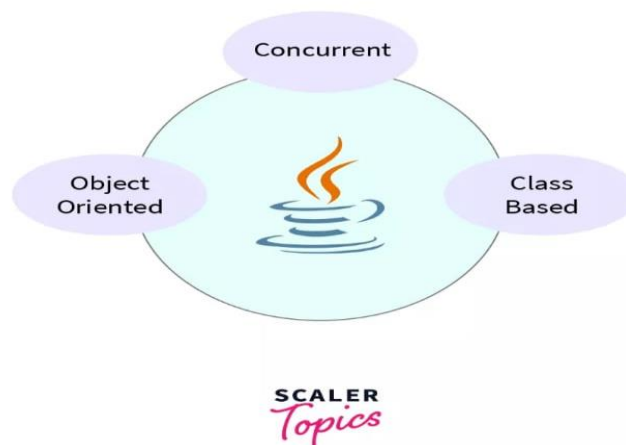


Figure 2.1 : Technical importance of JAVA

Technically, Java is a versatile language that is widely used in industry for developing web applications, mobile apps, enterprise software, and even embedded systems. Its platform independence and rich set of libraries make Java ideal for creating reliable, scalable, and maintainable applications. Through this training, learners observe and practice a structured sequence of concepts and hands-on exercises, from basic syntax to advanced applications. This observed learning sequence ensures a progressive understanding of Java's capabilities, fostering problem solving skills and preparing learners for real-world programming challenges. Mastery of Java positions individuals for success in software development.

Chapter-3: Detail study of Case Domain

Processes/Protocols/Methods/CaseDomain

- **Processes in Java: Compilation & Interpretation:** Understanding the process of converting Java code into bytecode using the Java compiler, followed by execution on the Java Virtual Machine (JVM), ensuring platform independence. **Garbage Collection:** Automated memory management in Java through garbage collection, which reclaims memory from unused objects. **Exception Handling:** Using try-catch blocks to ensure stability and error recovery in application.

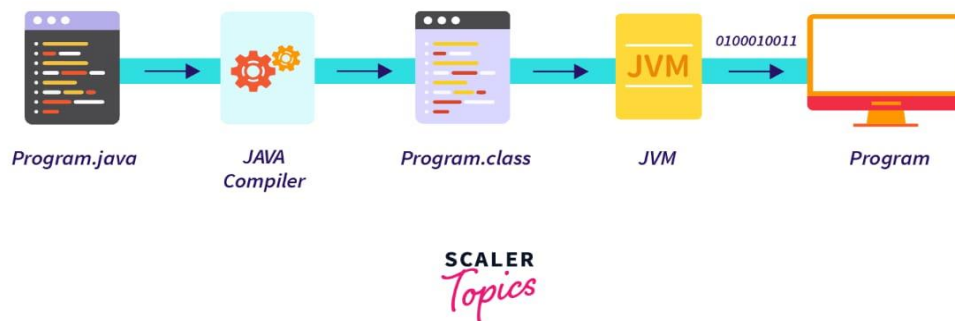


Figure 3.1 : Processes in JAVA

- **Protocols: TCP/IP:** Emphasis on networking protocols, such as TCP/IP, used in Java to facilitate communication between different devices over the internet or local networks. **HTTP/HTTPS:** Protocols used for web communication in Java, enabling the development of client-server applications, such as web browsers or Restful APIs.

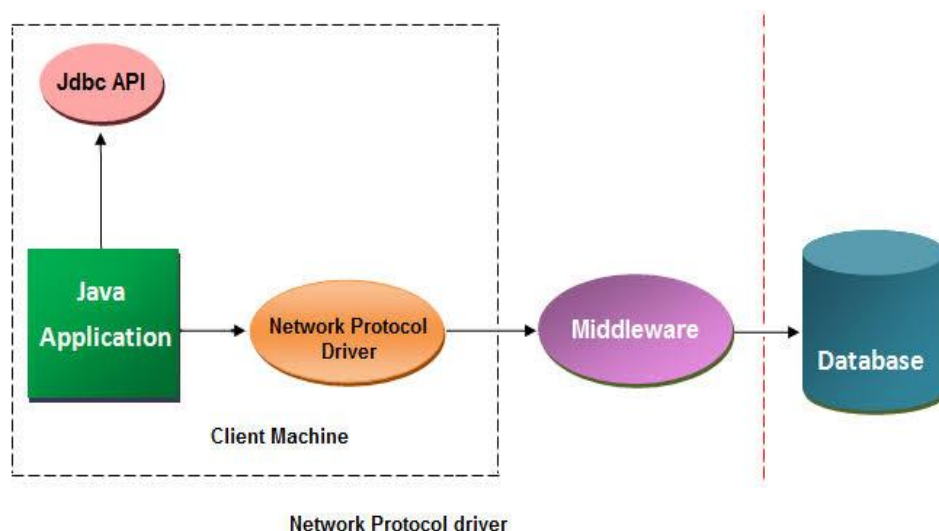


Figure 3.2 : Protocols in JAVA

- **Methods: Core Java Methods:** Key methods in Java's libraries for tasks like data manipulation (e.g., String manipulation), file handling, and input/output operations. **API Integration:** Utilizing Java APIs for additional functionality like database connectivity, GUI creation, and networking.
- **Languages: Java:** Primary focus on the Java programming language, learning syntax, OOP concepts, and libraries. **SQL:** Introduction to SQL for database interaction through Java, using JDBC for connecting Java applications to databases.



- **Case Domain:** Enterprise Application Prototype: A hands-on project, such as an inventory management system, simulating real-world applications. Trainees applied Java concepts in a practical case domain to implement business log id, database connectivity, and user interfaces.

Chapter 4: Proposed System Architecture and Design of the Problem

The system architecture defines the structure of the software solution, specifying how different components interact with each other. For a Java-based application, this could involve defining layers for data handling, business logic, and user interface.

Proposed System Architecture

- **User Interface (UI):** The system's front-end allows user interaction. It could be a command-line interface (CLI) or a graphical user interface (GUI) built using JavaFX or Swing. Components: Forms, Buttons, Text Boxes, Menus, etc.
- **Business Logic Layer:** This layer implements the core functionality of the application (e.g., processing input, managing data, etc.). In Java, this is typically where classes and methods are defined to handle core application logic. Components: Java classes, Methods, Algorithms.
- **Networking Layer:** In cases where the application communicates over a network, this layer handles protocol implementations like TCP/IP or HTTP/HTTPS. Components: Sockets, Web Services, REST API endpoints.

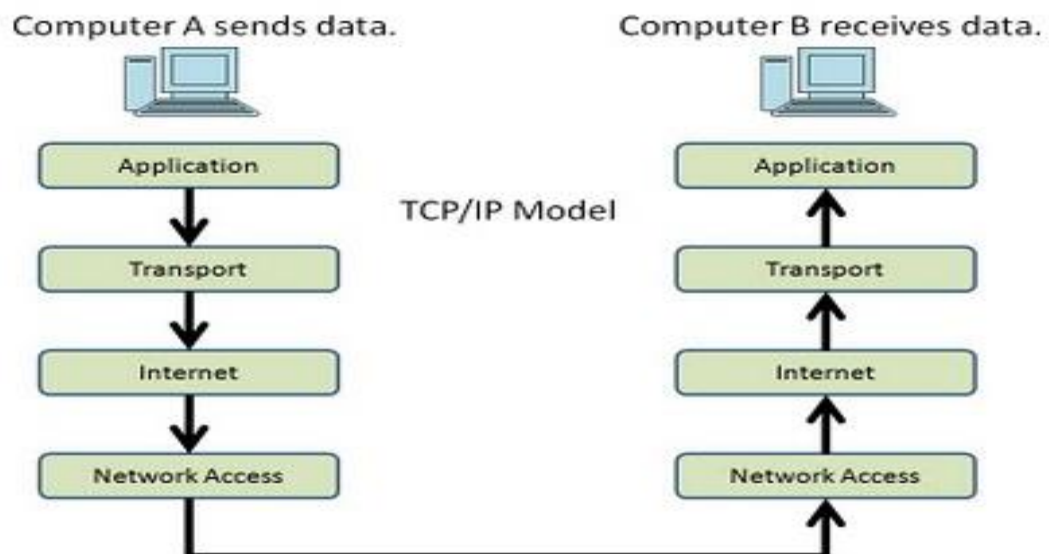


Figure 4.1 : TCP/IP Model

- **Data Layer**: Responsible for database interaction, storing and retrieving data using JDBC or other persistence methods. Components: Database, Data Access Objects (DAO).

Chapter 5 : Prototyped Module Implementations / Inspection Results

Module Implementations

Module Overview: The prototype module was developed to showcase the functionality of the Java application, focusing on key processes such as user authentication, database interaction, and data processing.

Core Features Implemented :User Authentication: Implemented login functionality with username and password validation. Database Connectivity: Integrated Java Database Connectivity (JDBC) for storing and retrieving user data. Data Processing: Developed functions for processing user input, performing calculations, or querying the database based on user interactions. User Interface: A simple command-line interface (CLI) or graphical user interface (GUI) for users to interact with the application.

Design Approach: Utilized object-oriented programming principles like classes and methods for modular design. Designed the system to follow a layered architecture: UI layer, Business Logic Layer, and Data Layer.

Technologies Used :Java: Core language for logic and functionalities. JDBC: For connecting to and interacting with the database.

Inspection Results

Code Quality: The code followed Java best practices, including proper indentation, meaningful variable/method names, and modular design for readability and maintainability.

Functionality :The core functionalities, including user login, data retrieval from the database, and error handling, worked as intended .The user interface was intuitive, and user inputs were processed without errors.

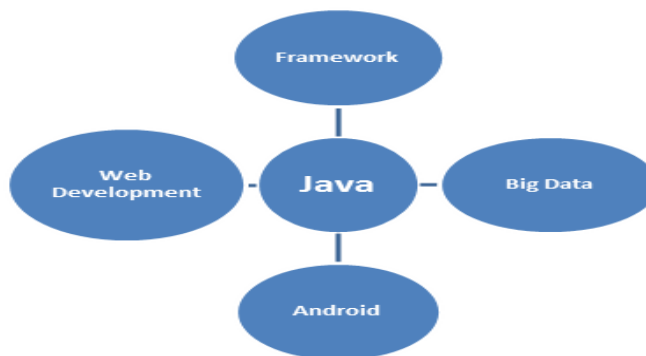
Performance :The application handled basic operations efficiently with minimal delays. There were no major performance bottlenecks for small to medium-sized datasets.

Error Handling :Proper exception handling was implemented to manage runtime errors such as invalid database connections or incorrect user inputs.

Usability :The user interface (CLI/GUI) was straightforward and user-friendly. However, some improvements could be made to enhance the overall user experience, such as adding input validation and error messages.

Chapter - 6 : Future Scope and Limitations

- **Development in Enterprise Applications**: Java continues to dominate the enterprise software sector, particularly for large-scale applications. With its robustness, security features, and scalability, Java remains a preferred choice for building complex, distributed systems such as banking, ecommerce platforms, and cloud-based solutions.
- **Android Development** :Java has been a foundational language for Android app development. While Kotlin is now officially recommended, Java will continue to play a vital role in maintaining and developing Android apps for the foreseeable future due to its vast libraries, frameworks, and compatibility.
- **Microservices Architecture** :Java is increasingly being used for building microservices-based applications. With frameworks like Spring Boot, Java offers a powerful ecosystem for creating small, independently deployable services that communicate over the network, making it an excellent choice for cloud-based architectures.
- **Big Data and Analytics**: Java is widely used in Big Data frameworks such as Apache Hadoop and Apache Spark. Java's performance, portability, and the availability of big data libraries make it a solid choice for building data processing and analytics pipelines.
- **Artificial Intelligence and Machine Learning** :While Python is popular for AI and ML, Java is steadily gaining ground in this field. With libraries like Deeplearning, Weka, and MOA, Java is increasingly being utilized for developing machine learning models and AI applications.



- **Internet of Things (IoT)** :Java's portability and scalability make it well-suited for IoT applications. Java is already being used to build IoT solutions in industries such as smart homes, healthcare, and automotive.

Limitations of Java

- **Performance Overhead** :Java is not as fast as some lower-level languages like C or C++ because it runs on the Java Virtual Machine (JVM) rather than directly on hardware. While improvements in JIT (Just-In-Time) compilers have mitigated some performance issues, Java can still be slower for applications that require high performance computing.
- **Memory Consumption**: Java applications tend to consume more memory compared to applications written in other languages. The automatic garbage collection mechanism, while efficient, also contributes to higher memory overhead, which can be a disadvantage in resource-constrained environments.
- **Verbosity** :Java's syntax can be verbose, requiring more lines of code for simple operations compared to modern programming languages like Python or Kotlin. This can increase development time and reduce code maintainability in some scenarios.
- **Limited Functional Programming Support**: While Java has introduced functional programming features like lambdas and streams, it is primarily an object-oriented language, which can make it less intuitive for developers who are accustomed to purely functional programming languages.
- **Complexity in GUI Development** :Java's GUI libraries, such as Swing and AWT, are considered outdated and less flexible than modern web-based UI frameworks. Although JavaFX is an improvement, it still falls behind popular web technologies like React or Angular for building user interfaces.

Chapter 7: Conclusion

- **Solid Java Foundation:** The training provided a strong foundation in Java programming, preparing me for further exploration of more advanced Java topics and technologies.
- **Increased Confidence:** The hands-on approach boosted my confidence in using Java for building real-world applications, including database-driven systems and user interfaces.
- **Application in Real-World Scenarios:** The training bridged theoretical knowledge with practical skills, enabling me to develop applications and solve complex problems.
- **Prepared for Future Learning:** Equipped with foundational knowledge and skills, I am now better prepared to explore Java frameworks, cloud computing, and other advanced areas in software development.
- **Future Scope:** The training opened doors for further specialization in areas such as Android development, web services, and enterprise applications.
- **Overall Development:** The Java training has significantly enhanced my programming skills, contributing to my growth as a software developer.

Chapter 8: Bibliography

1. Books:

- Herbert Schildt, "Java: The Complete Reference," 11th Edition, McGraw-Hill Education, 2018.
- Joshua Bloch, "Effective Java," 3rd Edition, Addison-Wesley, 2018.
- Kathy Sierra & Bert Bates, "Head First Java," 2nd Edition, O'Reilly Media, 2005.
- Brian Goetz, "Java Concurrency in Practice," Addison-Wesley, 2006.

2. Online Documentation:

- Oracle Corporation, "The Java™ Tutorials," Oracle, <https://docs.oracle.com/javase/tutorial/>.
- Oracle Corporation, "Java SE Documentation," <https://docs.oracle.com/en/java/javase/>.

3. Websites:

- Stack Overflow, "Java Programming," Stack Exchange, <https://stackoverflow.com/questions/tagged/java>.
- GeeksforGeeks, "Java Programming Language," GeeksforGeeks, <https://www.geeksforgeeks.org/java/>.

4. Other Resources:

- JetBrains, "IntelliJ IDEA Documentation," JetBrains, <https://www.jetbrains.com/idea/documentation/>.
- Apache Software Foundation, "JDBC API," Apache, <https://jdbc.apache.org/>.

Chapter 9 : Appendix

Splash class:

Package electricity billing system;

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
public class Splash extends JFrame {
```

```
    Splash(){
```

```
        ImageIcon imageicon= newImageIcon(ClassLoader.getResource("icon/splash/Splash.jpg"));
```

```
        Image imageOne = imageicon.getImage().getScaledInstance(600,400,Image.SCALE_DEFAULT);
```

```
        ImageIcon imageIcon2 = new ImageIcon(imageOne);
```

```
        JLabel imageLabel = new JLabel(imageicon);
```

```
        add(imageLabel);
```

```
        setLocation(300,400);    // For Display Location
```

```
        setSize(400,400);    //Size of Screen
```

```
        setVisible(true);    // Frame Visibility default is Hidden.
```

```
        try{
```

```
            Thread.sleep(3000);
```

```
            setVisible(false);
```

```
            new Login();
```

```
        }catch(Exception e){
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
    public static void main(String[] args){
```

```
        new Splash();
```

```
    }
```

```
}
```

Main Class:

```
package electricitybillingsystem;
import com.mysql.cj.log.Log;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MAIN extends JFrame implements ActionListener {
    String accType,Meter_pass;
    MAIN(String acctype,String Meter_pass){

        this.accType=acctype;
        this.Meter_pass=Meter_pass;
        setExtendedState(JFrame.MAXIMIZED_BOTH);

        ImageIcon imgicon = new ImageIcon(ClassLoader.getResource("icon/ebs.png"));
        Image image = imgicon.getImage().getScaledInstance(1530,830,Image.SCALE_DEFAULT);
        ImageIcon imgicon2 = new ImageIcon(image);
        JLabel imglabel = new JLabel(imgicon2);
        add(imglabel);

        JMenuBar menuBar = new JMenuBar();
        setJMenuBar(menuBar);

        JMenu menu = new JMenu("Menu");
        menu.setFont(new Font("serif",Font.PLAIN,18));

        JMenuItem newCustomer = new JMenuItem("New Customer");
        newCustomer.setFont(new Font("monospace",Font.PLAIN,14));
        ImageIcon customerimg = new
ImageIcon(ClassLoader.getResource("icon/newcustomer.png"));
        Image customerImage =
customerimg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
        newCustomer.setIcon(new ImageIcon(customerImage));
        newCustomer.addActionListener(this);
        menu.add(newCustomer);

        JMenuItem customerDetail = new JMenuItem("Customer Details");
        customerDetail.setFont(new Font("monospace",Font.PLAIN,14));
        ImageIcon customerDetail1 = new
ImageIcon(ClassLoader.getResource("icon/customerDetails.png"));
```

```

Image customerDetailImg2 =
customerDetail1.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
customerDetail.setIcon(new ImageIcon(customerDetailImg2));
customerDetail.addActionListener(this);
menu.add(customerDetail);

JMenuItem depositDetails = new JMenuItem("Deposit Details");
depositDetails.setFont(new Font("monospace",Font.PLAIN,14));
ImageIcon depositimg = new
ImageIcon(ClassLoader.getResource("icon/depositdetails.png"));
Image depositimg2 = depositimg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
depositDetails.setIcon(new ImageIcon(depositimg2));
depositDetails.addActionListener(this);
menu.add(depositDetails);

JMenuItem calculateBill = new JMenuItem("Calculate Bill");
depositDetails.setFont(new Font("monospace",Font.PLAIN,14));
ImageIcon calculateBillImg = new
ImageIcon(ClassLoader.getResource("icon/calculatorbills.png"));
Image calculateBillImg2 =
calculateBillImg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
calculateBill.setIcon(new ImageIcon(calculateBillImg2));
calculateBill.addActionListener(this);
menu.add(calculateBill);

JMenu info = new JMenu("Info");
info.setFont(new Font("serif",Font.PLAIN,18));

JMenuItem upinfo = new JMenuItem("Update Information");
upinfo.setFont(new Font("monospace",Font.PLAIN,14));
ImageIcon infoimg = new ImageIcon(ClassLoader.getResource("icon/refresh.png"));
Image upinfo2 = infoimg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
upinfo.setIcon(new ImageIcon(upinfo2));
upinfo.addActionListener(this);
info.add(upinfo);

JMenuItem viewInfo = new JMenuItem("View Information");
viewInfo.setFont(new Font("monospace",Font.PLAIN,14));
ImageIcon viewinfoImg= new
ImageIcon(ClassLoader.getResource("icon/information.png"));
Image viewinfoImg2=
viewinfoImg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
viewInfo.setIcon(new ImageIcon(viewinfoImg2));

```



```
viewInfo.addActionListener(this);  
info.add(viewInfo);
```

```
JMenu user = new JMenu("User");  
user.setFont(new Font("serif",Font.PLAIN,18));
```

```
JMenuItem payBill = new JMenuItem("Pay Bill");  
payBill.setFont(new Font("monospace",Font.PLAIN,14));  
ImageIcon payBillImg= new ImageIcon(ClassLoader.getResource("icon/pay.png"));  
Image payBillImg2= payBillImg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);  
payBill.setIcon(new ImageIcon(payBillImg2));  
payBill.addActionListener(this);  
user.add(payBill);
```

```
JMenuItem billDetails = new JMenuItem("Bill Details");  
billDetails.setFont(new Font("monospace",Font.PLAIN,14));  
ImageIcon billDetailsImg= new ImageIcon(ClassLoader.getResource("icon/detail.png"));  
Image billDetailsImg2=  
billDetailsImg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);  
billDetails.setIcon(new ImageIcon(billDetailsImg2));  
billDetails.addActionListener(this);  
user.add(billDetails);
```

```
JMenu bill = new JMenu("Bill");  
bill.setFont(new Font("serif",Font.PLAIN,18));
```

```
JMenuItem genBill = new JMenuItem("Generate Bill");  
genBill.setFont(new Font("monospace",Font.PLAIN,14));  
ImageIcon genBillimg = new ImageIcon(ClassLoader.getResource("icon/bill.png"));  
Image genBillimg2 = genBillimg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);  
genBill.setIcon(new ImageIcon(genBillimg2));  
genBill.addActionListener(this);  
bill.add(genBill);
```

```
JMenu utility = new JMenu("Utility");  
utility.setFont(new Font("serif",Font.PLAIN,18));
```

```
JMenuItem notepad = new JMenuItem("Notepad");  
notepad.setFont(new Font("monospace",Font.PLAIN,14));  
ImageIcon notepadImg = new ImageIcon(ClassLoader.getResource("icon/notepad.png"));  
Image notepadImg2 =  
notepadImg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
```

```
notepad.setIcon(new ImageIcon(notepadImg2));
notepad.addActionListener(this);
utility.add(notepad);
```

```
JMenuItem calc = new JMenuItem("Calculator");
calc.setFont(new Font("monospace",Font.PLAIN,14));
ImageIcon calcImg = new ImageIcon(ClassLoader.getResource("icon/calculator.png"));
Image calcImg2 = calcImg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
calc.setIcon(new ImageIcon(calcImg2));
calc.addActionListener(this);
utility.add(calc);
```

```
JMenu exit = new JMenu("LOG OUT");
exit.setFont(new Font("serif",Font.PLAIN,18));
```

```
JMenuItem exititem = new JMenuItem("LOG OUT");
exititem.setFont(new Font("monospace",Font.PLAIN,14));
ImageIcon exitimg = new ImageIcon(ClassLoader.getResource("icon/exit.png"));
Image exitimg2 = exitimg.getImage().getScaledInstance(20,20,Image.SCALE_DEFAULT);
exititem.setIcon(new ImageIcon(exitimg2));
exititem.addActionListener(this);
exit.add(exititem);
```

```
if(acctype.equals("Admin")) {
    menuBar.add(menu);
}else {
    menuBar.add(bill);
    menuBar.add(user);
    menuBar.add(info);
}
menuBar.add(utility);
menuBar.add(exit);
```

```
setLayout(new FlowLayout());
setVisible(true);
```

```
}
```

```
@Override
```

```

public void actionPerformed(ActionEvent e) {
    String msg = e.getActionCommand();
    if(msg.equals("New Customer")){
        new newCustomer();
    }else if(msg.equals("Customer Details")){
        new CustomerDetails();
    }else if(msg.equals("Deposit Details")){
        new DepositDetails();
    }else if(msg.equals("Calculate Bill")){
        new CalculateBill();
    }else if(msg.equals("View Information")){
        new viewInformation(Meter_pass);
    }else if(msg.equals("Bill Details")){
        new billDetails(Meter_pass);
    }else if(msg.equals("Calulator")){
        try{
            Runtime.getRuntime().exec("calc.exe");
        }catch(Exception E){
            E.printStackTrace();
        }
    }else if(msg.equals("Notepad")) {
        try {
            Runtime.getRuntime().exec("notepad.exe");
        } catch (Exception E) {
            E.printStackTrace();
        }
    } else if (msg.equals("LOG OUT")) {
        setVisible(false);
        new Login();
    } else if (msg.equals("Pay Bill")) {
        new payBill(Meter_pass);
    } else if (msg.equals("Update Information")) {
        new updateInformation(Meter_pass);
    }else if (msg.equals("Generate Bill")) {
        new generateBill(Meter_pass);
    }
}

public static void main(String[] args){
    new MAIN("", "");
}
}

```

Database Class:

```
package electricitybillingsystem;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

public class DataBase {
    Connection con;
    Statement stmt;
    DataBase(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/BILL__SYSTEM","root","siyaban1705#");
            stmt= con.createStatement();
        }catch(Exception e){
            e.printStackTrace();

        }
    }
}
```

BillDetail Class :

```
package electricitybillingsystem;

import net.proteanit.sql.DbUtils;

import javax.swing.*;
import javax.xml.crypto.Data;
import java.awt.*;
import java.sql.ResultSet;

public class billDetails extends JFrame {

    String Meter;
    billDetails(String Meter){

        this.Meter=Meter;
        setSize(700,650);
        setLocation(400,150);
        setLayout(null);
        getContentPane().setBackground(Color.WHITE);

        JTable table = new JTable();
        try{
            DataBase c = new DataBase();
            String query_bill = "select * from bill where meterNum = '"+Meter+"'";
            ResultSet rs = c.stmt.executeQuery(query_bill);
            table.setModel(DbUtils.resultSetToTableModel(rs));

        }catch(Exception e){
            e.printStackTrace();
        }

        JScrollPane sp = new JScrollPane(table);
        sp.setBounds(0,0,700,650);
        add(sp);

        setVisible(true);
    }

    public static void main(String[] args) {
        new billDetails("");
    }
}
```

Update Information Class:

```
package electricitybillingsystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.RectangularShape;
import java.sql.ResultSet;

public class updateInformation extends JFrame implements ActionListener {
    JLabel nametext;
    JTextField addressText,cityText,stateText,emailText,phoneText;
    String meter;
    JButton update,cancel;
    updateInformation(String meter){
        this.meter=meter;
        setBounds(400,150,777,450);
        getContentPane().setBackground(new Color(229,255,227));
        setLayout(null);

        JLabel heading = new JLabel("Update Customer Information");
        heading.setBounds(50,10,400,40);
        heading.setFont(new Font("serif",Font.BOLD,20));
        add(heading);

        JLabel name = new JLabel("Name");
        name.setBounds(30,70,100,20);
        add(name);

        nametext = new JLabel("");
        nametext.setBounds(150,70,200,20);
        add(nametext);

        JLabel meterNo = new JLabel("Meter Number");
        meterNo.setBounds(30,110,100,20);
        add(meterNo);

        JLabel meterText = new JLabel("");
        meterText.setBounds(150,110,100,20);
        add(meterText);

        JLabel address = new JLabel("Address");
        address.setBounds(30,150,100,20);
```

```
add(address);
```

```
addressText = new JTextField();  
addressText.setBounds(150,150,200,20);  
add(addressText);
```

```
JLabel city = new JLabel("City");  
city.setBounds(30,190,100,20);  
add(city);
```

```
cityText = new JTextField();  
cityText.setBounds(150,190,200,20);  
add(cityText);
```

```
JLabel state = new JLabel("State");  
state.setBounds(30,230,100,20);  
add(state);
```

```
stateText = new JTextField();  
stateText.setBounds(150,230,200,20);  
add(stateText);
```

```
JLabel email = new JLabel("Email");  
email.setBounds(30,270,100,20);  
add(email);
```

```
emailText = new JTextField();  
emailText.setBounds(150,270,200,20);  
add(emailText);
```

```
JLabel phone = new JLabel("Phone");  
phone.setBounds(30,310,100,20);  
add(phone);
```

```
phoneText = new JTextField();  
phoneText.setBounds(150,310,200,20);  
add(phoneText);
```

```
try{  
    DataBase c = new DataBase();  
    ResultSet resultSet = c.stmt.executeQuery("select * from newCustomer where meterNo =  
    '"+meter+"'");  
    if (resultSet.next()){  
        nametext.setText(resultSet.getString("name"));  
        meterText.setText(resultSet.getString("meterNo"));  
    }  
}
```

```

        addressText.setText(resultSet.getString("address"));
        cityText.setText(resultSet.getString("city"));
        stateText.setText(resultSet.getString("state"));
        emailText.setText(resultSet.getString("email"));
        phoneText.setText(resultSet.getString("phoneNo"));
    }
} catch (Exception e){
    e.printStackTrace();
}

update = new JButton("Update");
update.setBackground(new Color(33,106,145));
update.setForeground(Color.white);
update.setBounds(50,360,120,25);
update.addActionListener(this);
add(update);

cancel = new JButton("Cancel");
cancel.setBackground(new Color(33,106,145));
cancel.setForeground(Color.white);
cancel.setBounds(200,360,120,25);
cancel.addActionListener(this);
add(cancel);

ImageIcon imageIcon = new ImageIcon(ClassLoader.getResource("icon/update.png"));
Image image = imageIcon.getImage().getScaledInstance(400,410,Image.SCALE_DEFAULT);
ImageIcon imageIcon1 = new ImageIcon(image);
JLabel imgLabel = new JLabel(imageIcon1);
imgLabel.setBounds(360,0,400,410);
add(imgLabel);

setVisible(true);

}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource()==update){
        String saddress = addressText.getText();
        String scity = cityText.getText();
        String sstate = stateText.getText();
        String semail = emailText.getText();
        String sphone = phoneText.getText();
    }
}

```



```
try{
    DataBase c = new DataBase();
    c.stmt.executeUpdate("update newCustomer set address='"+saddress+"', city='"+scity+"',
state='"+sstate+"', email='"+semail+"', phoneNo='"+sphone+"' where meterNo='"+meter+"'");

    JOptionPane.showMessageDialog(null,"User Information Updated Successfully");
    setVisible(false);
}catch (Exception E){
    E.printStackTrace();
}
}else {
    setVisible(false);
}
}

public static void main(String[] args) {
    new updateInformation("");
}
}
```

SignUp Class:

```
package electricitybillingsystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.ResultSet;

public class Signup extends JFrame implements ActionListener {
    Choice loginASCho;
    TextField meterText,EmployerText,userNameText,nameText,passwordText;
    JButton create,back;
    Signup(){
        super("Signup Page");
        getContentPane().setBackground(new Color(168,203,255));

        JLabel createAs = new JLabel("Create Account As");
        createAs.setBounds(30,50,125,20);
        add(createAs);

        loginASCho = new Choice();
        loginASCho.add("Admin");
        loginASCho.add("Customer");
        loginASCho.setBounds(170,50,120,20);
        add(loginASCho);

        JLabel meterNo = new JLabel("Meter Number");
        meterNo.setBounds(30,100,125,20);
        meterNo.setVisible(false);
        add(meterNo);

        meterText = new TextField();
        meterText.setBounds(170,100,125,20);
        meterText.setVisible(false);
        add(meterText);

        JLabel Employer = new JLabel("Employer ID");
        Employer.setBounds(30,100,125,20);
        Employer.setVisible(true);
        add(Employer);

        EmployerText = new TextField();
        EmployerText.setBounds(170,100,125,20);
        EmployerText.setVisible(true);
        add(EmployerText);
```

```
JLabel userName = new JLabel("UserName");
userName.setBounds(30,140,125,20);
add(userName);
```

```
userNameText = new TextField();
userNameText.setBounds(170,140,125,20);
add(userNameText);
```

```
JLabel name = new JLabel("Name");
name.setBounds(30,180,125,20);
add(name);
```

```
nameText = new TextField("");
nameText.setBounds(170,180,125,20);
```

```
meterText.addFocusListener(new FocusListener() {
    @Override
    public void focusGained(FocusEvent e) {

    }
}
```

```
    @Override
    public void focusLost(FocusEvent e) {
        try{
            DataBase c = new DataBase();
            ResultSet resultSet = c.stmt.executeQuery("select * from Signup where meter_no
=" + meterText.getText());
            if (resultSet.next()){
                nameText.setText(resultSet.getString("name"));
            }

        }catch (Exception E){
            E.printStackTrace();
        }

    }
});
```

```
add(nameText);
JLabel password = new JLabel("Password");
password.setBounds(30,220,125,20);
add(password);
```

```
passwordText = new TextField();
passwordText.setBounds(170,220,125,20);
add(passwordText);
```

```
loginAScho.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent e) {
        String user = loginAScho.getSelectedItem();
        if (user.equals("Customer")){
            Employer.setVisible(false);
            nameText.setEditable(false);
            EmployerText.setVisible(false);
            meterNo.setVisible(true);
            meterText.setVisible(true);
        }else {
            Employer.setVisible(true);
            EmployerText.setVisible(true);
            meterNo.setVisible(false);
            meterText.setVisible(false);
        }
    }
});
```

```
create = new JButton("Create");
create.setBackground(new Color(66,127,219));
create.setForeground(Color.black);
create.setBounds(50,285,100,25);
create.addActionListener(this);
add(create);
```

```
back = new JButton("Back");
back.setBackground(new Color(66,127,219));
back.setForeground(Color.black);
back.setBounds(180,285,100,25);
back.addActionListener(this);
add(back);
```

```
ImageIcon boyIcon = new ImageIcon(ClassLoader.getResource("icon/boy.png"));
Image boyImg = boyIcon.getImage().getScaledInstance(250,250,Image.SCALE_DEFAULT);
ImageIcon boyIcon2 = new ImageIcon(boyImg);
JLabel boyLabel = new JLabel(boyIcon2);
boyLabel.setBounds(320,30,250,250);
add(boyLabel);
```

```

setSize(600,380);
setLocation(500,200);
setLayout(null);
setVisible(true);
}
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource()== create){
        String sloginAs = loginASCho.getSelectedItem();
        String susername = userNameText.getText();
        String sname = nameText.getText();
        String spassword = passwordText.getText();
        String smeter = meterText.getText();
        try{
            DataBase c = new DataBase();
            String query= null;
            if (sloginAs.equals("Admin")) {
                query = "insert into Signup values('"+ smeter + "', '"+ susername + "', '"+ sname + "', '"+
spassword + "', '"+ sloginAs + "')";
            }else {
                query = "update Signup set username = '"+suserame+"', pass = '"+spassword+"', usertype
= '"+sloginAs+" where meter_no = '"+smeter+"'";
            }
            c.stmt.executeUpdate(query);

            JOptionPane.showMessageDialog(null,"Account Created");
            setVisible(false);
            new Login();

        }catch (Exception E){
            E.printStackTrace();
        }
    } else if (e.getSource()==back) {
        setVisible(false);
        new Login();
    }
}

public static void main(String[] args) {
    new Signup();
}
}

```

NewCustomer Class:

```
package electricitybillingsystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

public class newCustomer extends JFrame implements ActionListener {

    JLabel heading,meterNum,addr,city,state,email,phNum,meterNumText;
    TextField nameText,addrText,cityText,stateText,emailText,phNumText;
    JButton next,cancel;

    newCustomer(){
        super("New Customer");

        JPanel panel = new JPanel();
        panel.setLayout(null);
        panel.setBackground(new Color(252,186,3));
        add(panel);

        heading = new JLabel("New Customer");
        heading.setBounds(180,10,200,20);
        heading.setFont(new Font("Tahoma",Font.BOLD,20));
        panel.add(heading);

        JLabel customerName = new JLabel("New Customer");
        customerName.setBounds(50,80,100,20);
        panel.add(customerName);

        nameText = new TextField();
        nameText.setBounds(180,80,150,20);
        panel.add(nameText);

        meterNum = new JLabel("Meter Number");
        meterNum.setBounds(50,120,100,20);
        panel.add(meterNum);

        meterNumText = new JLabel();
        meterNumText.setBounds(180,120,150,20);
        panel.add(meterNumText);
```

```
Random random = new Random();  
long number = random.nextLong()% 1000000;  
meterNumText.setText("    "+Math.abs(number));
```

```
addr = new JLabel("Address");  
addr.setBounds(50,160,100,20);  
panel.add(addr);
```

```
addrText = new TextField();  
addrText.setBounds(180,160,150,20);  
panel.add(addrText);
```

```
city = new JLabel("City");  
city.setBounds(50,200,100,20);  
panel.add(city);
```

```
cityText = new TextField();  
cityText.setBounds(180,200,150,20);  
panel.add(cityText);
```

```
state = new JLabel("State");  
state.setBounds(50,240,100,20);  
panel.add(state);
```

```
stateText = new TextField();  
stateText.setBounds(180,240,150,20);  
panel.add(stateText);
```

```
email = new JLabel("E-Mail");  
email.setBounds(50,280,100,20);  
panel.add(email);
```

```
emailText = new TextField();  
emailText.setBounds(180,280,150,20);  
panel.add(emailText);
```

```
phNum = new JLabel("Phone Number");  
phNum.setBounds(50,320,100,20);  
panel.add(phNum);
```

```
phNumText = new TextField();  
phNumText.setBounds(180,320,150,20);  
panel.add(phNumText);
```

```
next = new JButton("Next");
next.addActionListener(this);
next.setBounds(90,390,100,25);
panel.add(next);
```

```
cancel = new JButton("Cancel");
cancel.addActionListener(this);
cancel.setBounds(220,390,100,25);
panel.add(cancel);
setLayout(new BorderLayout());
add(panel,"Center");
```

```
ImageIcon img1 = new ImageIcon(ClassLoader.getResource("icon/boy.png"));
Image img2 = img1.getImage().getScaledInstance(230,200,Image.SCALE_DEFAULT);
ImageIcon img3= new ImageIcon(img2);
JLabel imglable = new JLabel(img3);
add(imglable,"West");
```

```
setSize(700,500);
setLocation(400,200);
setVisible(true);
}
```

@Override

```
public void actionPerformed(ActionEvent e) {
    if(e.getSource()==next){
        String sname= nameText.getText();
        String smeter = meterNumText.getText();
        String saddress = addrText.getText();
        String scity = cityText.getText();
        String sstate = stateText.getText();
        String eemail = emailText.getText();
        String sphone = phNumText.getText();
```

```
        String queryCustomer = "insert into newCustomer
values('"+sname+"','"+smeter+"','"+saddress+"','"+scity+"','"+sstate+"','"+eemail+"','"+sphone+"')";
        String querySignup = "insert into Signup values ('"+smeter+"','"+sname+"','"+sphone+"')";
        try{

            DataBase c = new DataBase();
            c.stmt.executeUpdate(queryCustomer);
            c.stmt.executeUpdate(querySignup);
```



```
        JOptionPane.showMessageDialog(null,"Customer Details Added Successfully");
        setVisible(false);
        new MeterInfo(smeter);
    }catch(Exception E){
        E.printStackTrace();
    }
}else {
    setVisible(false);
}
}

public static void main(String[] args) {
    new newCustomer();
}
}
```

DFD of Electricity Billing System:

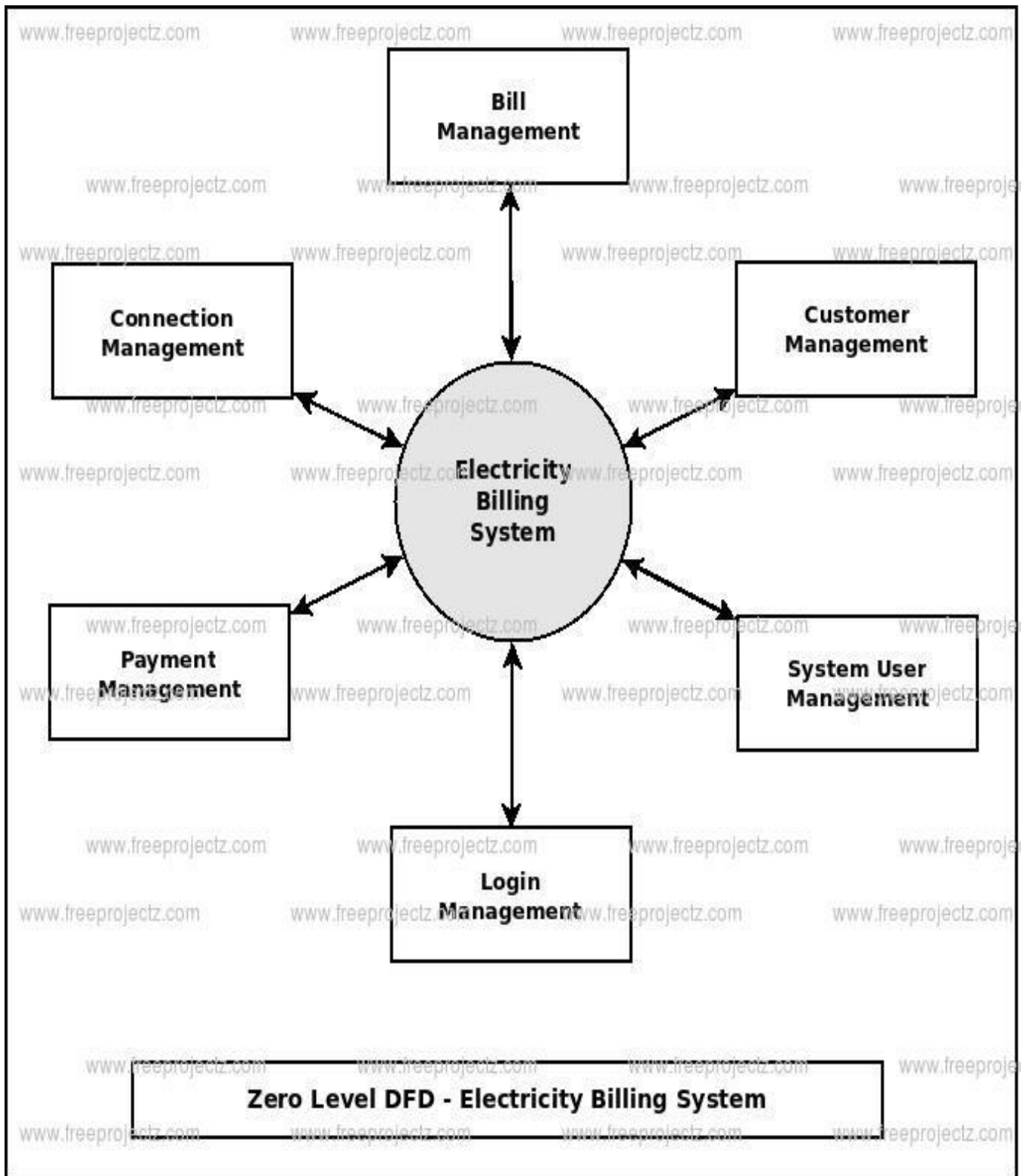


Figure A.1

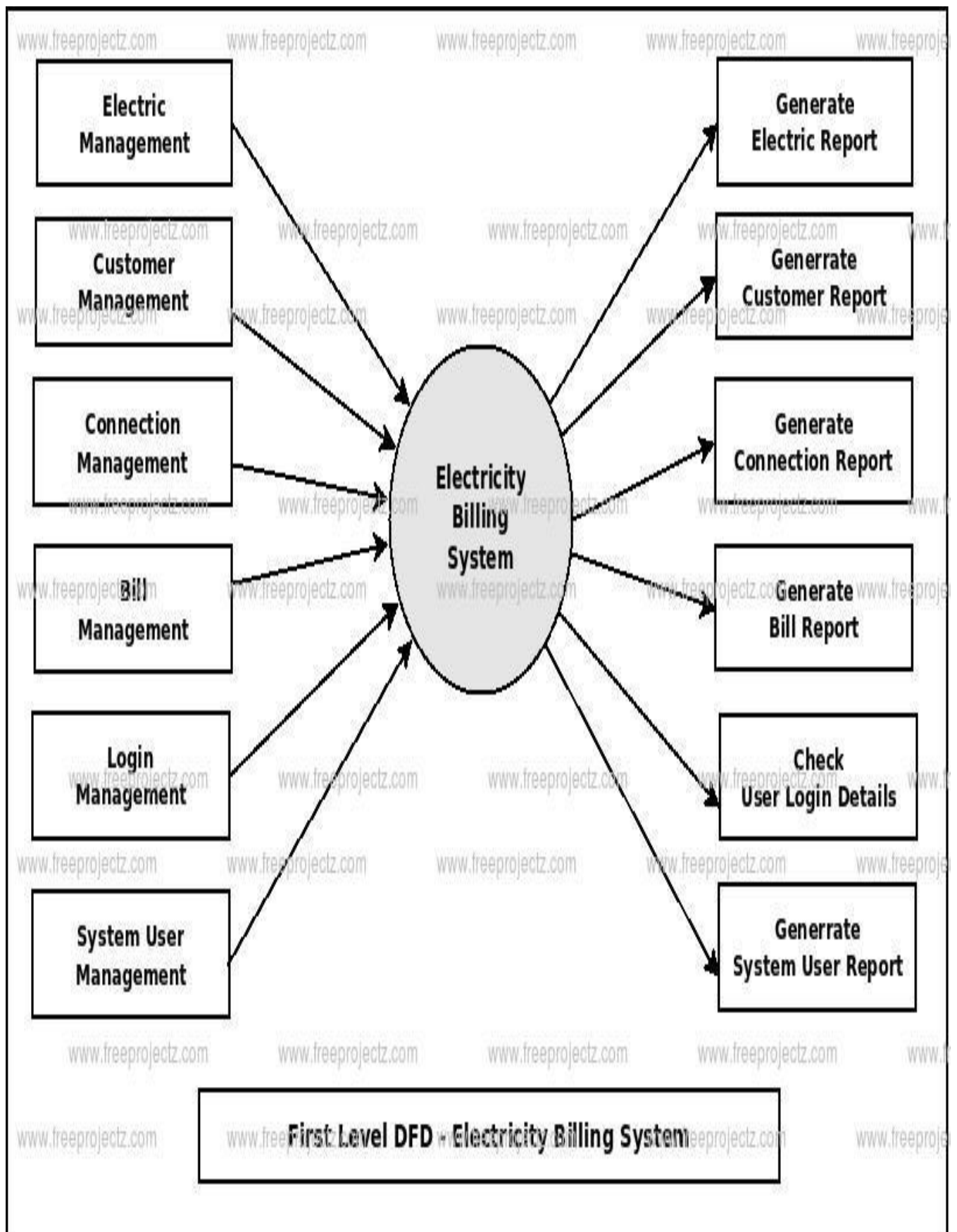


Figure A.2

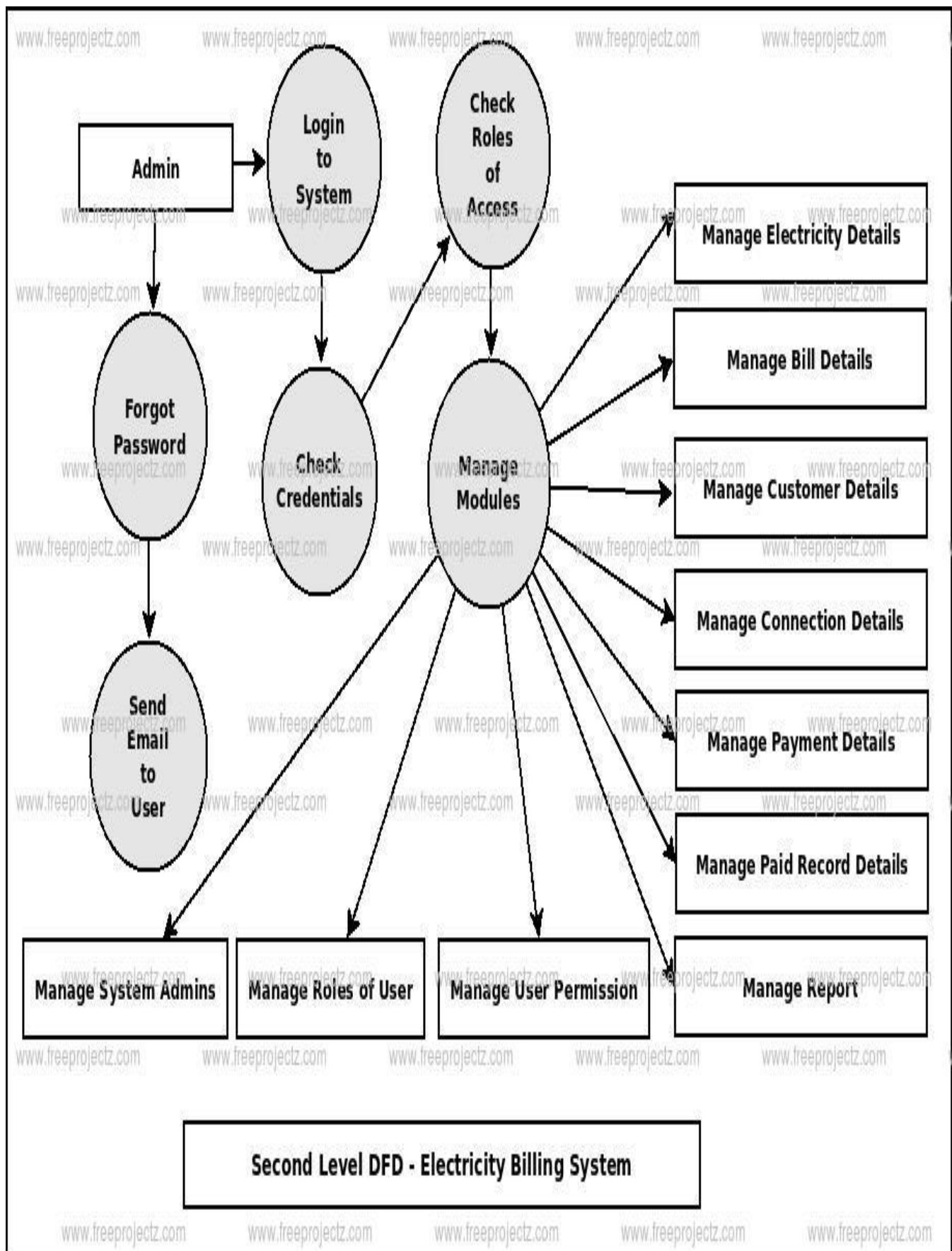
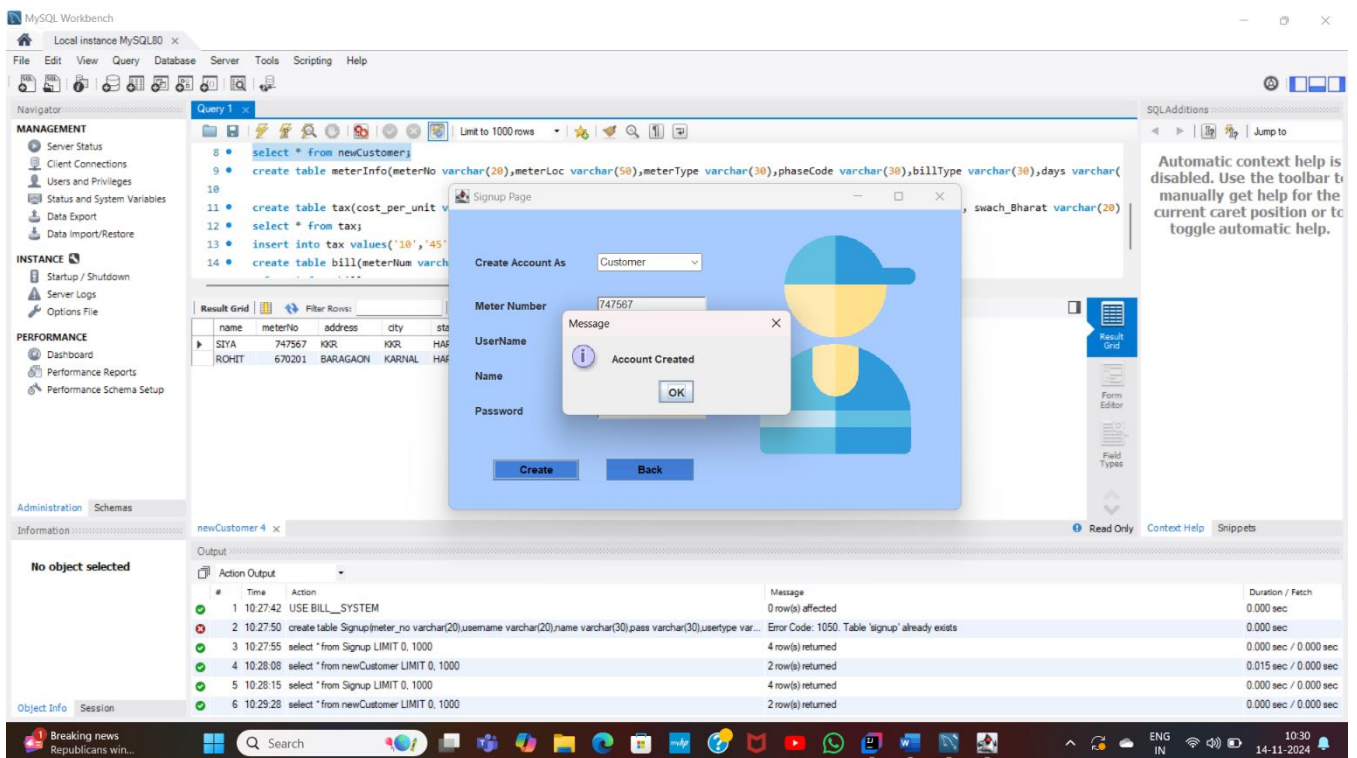
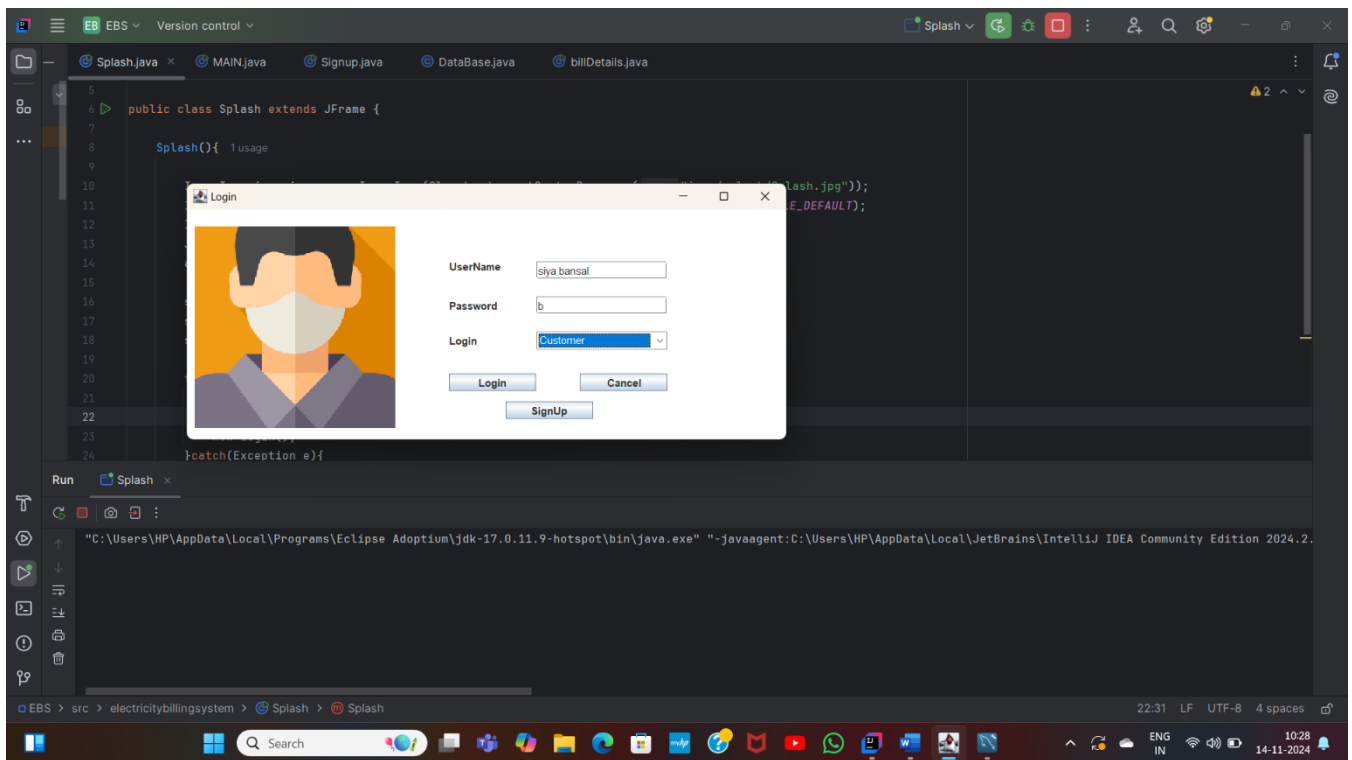


Figure A.3

Outputs:



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Object Info Session

Query 1

```
1 CREATE DATABASE BILL_SYSTEM;
2 USE BILL_SYSTEM;
3 create table Signup(meter_no varchar(20),username varchar(20),name varchar(30),pass varchar(30),usertype varchar(20));
4 select * from Signup;
5 create table newCustomer(name varchar(30),meterNo varchar(20),address varchar(50),city varchar(30),state varchar(20),email varchar(30),phoneNo
```

Result Grid

meter_no	username	name	pass	usertype
747567	siya	SIYA	b	Customer
SSS	SIYABANSAL	888	Admin	
670201	RR	ROHET	1234	Customer
radhika	radhika	12345	Admin	
sanchana	sanchana	bansal	Admin	
191840	siya	siya bansal	12345	Admin
580234	tanu12	tanu	bansal	Admin
874882	amit			

Signup 1

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:32:29	CREATE DATABASE BILL_SYSTEM	Error Code: 1007. Can't create database 'bill_system'; database exists	0.000 sec
2	16:32:33	select * from Signup LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
3	16:32:41	select * from newCustomer LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
4	16:32:48	USE BILL_SYSTEM	0 row(s) affected	0.000 sec
5	16:32:55	select * from Signup LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

16:33 23-11-2024

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Object Info Session

Query 1

```
1 CREATE DATABASE BILL_SYSTEM;
```

Login

UserName: radhika

Password: 12345

Login: Admin

Login Cancel

SignUp

Signup 6

Output

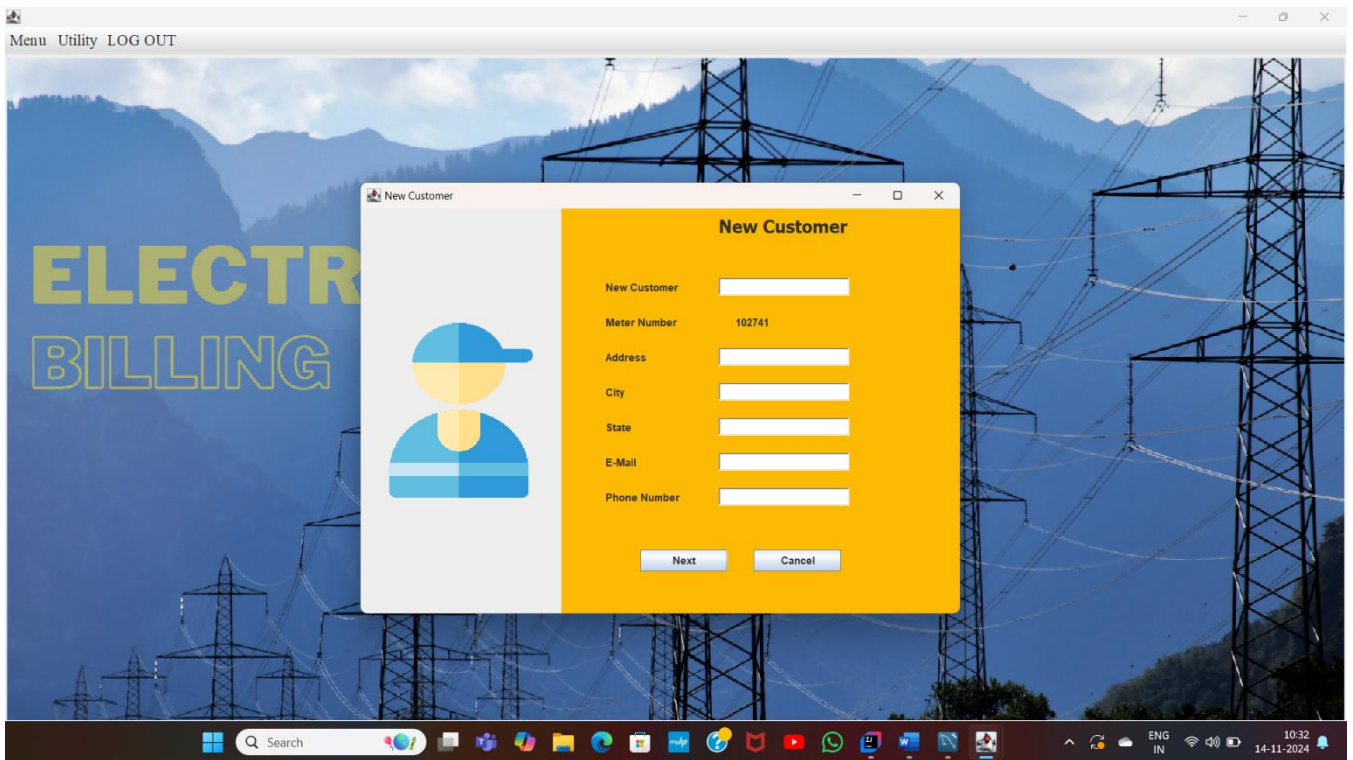
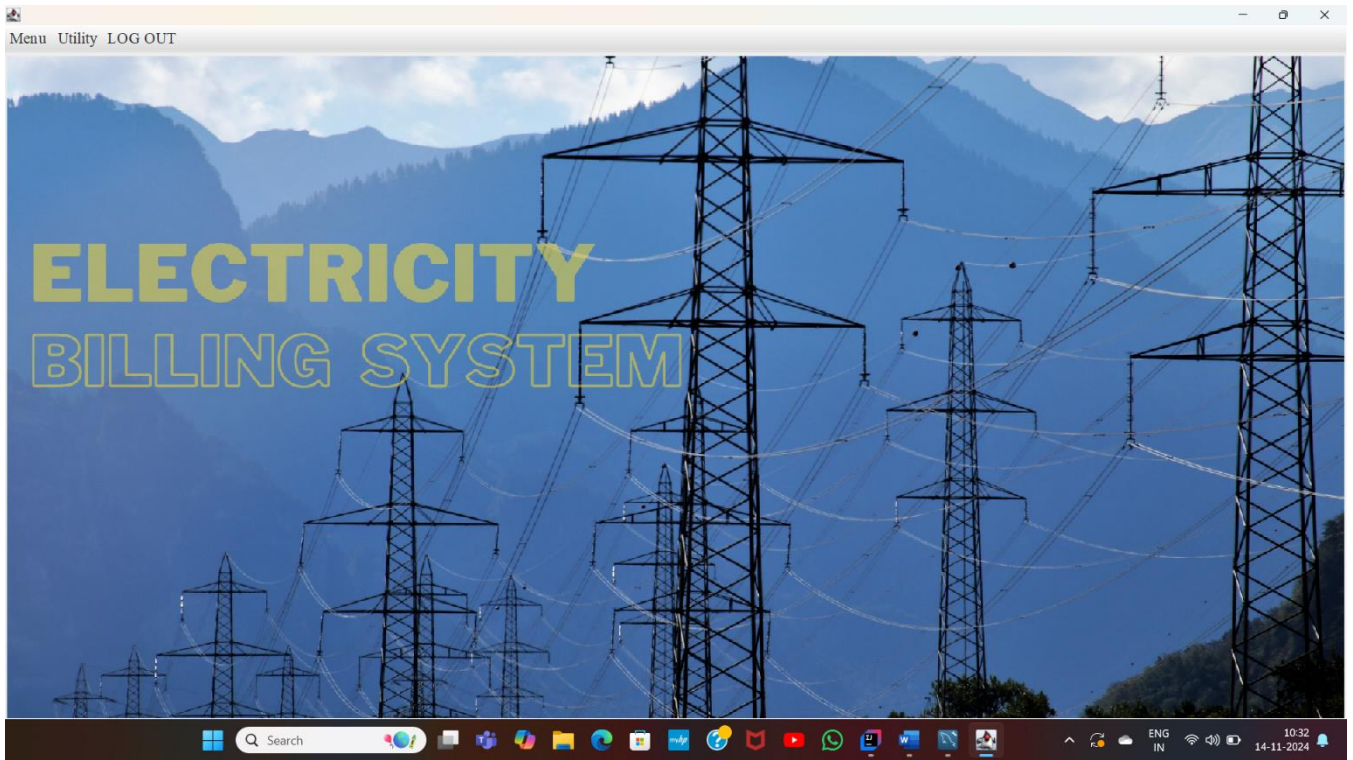
Action Output

#	Time	Action	Message	Duration / Fetch
4	10:28:08	select * from newCustomer LIMIT 0, 1000	2 row(s) returned	0.015 sec / 0.000 sec
5	10:28:15	select * from Signup LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
6	10:29:28	select * from newCustomer LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
7	10:30:40	USE BILL_SYSTEM	0 row(s) affected	0.000 sec
8	10:30:44	select * from Signup LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
9	10:32:14	select * from Signup LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

10:32 14-11-2024



MenuUtilityLOG OUT

ELECTR
BILLING

Customer Details

Search By Meter Number747567

Search By NameSIYA

Search

Print

Close

name	meterNo	address	city	state	email	phoneNo
SIYA	747567	KKR	KKR	HARYANA	12311	111111
ROHIT	670201	BARAGAON	KARNAL	HARYANA	@GMAIL.COM	234234523

Search

14-11-2024

MenuUtilityLOG OUT

ELECTR
BILLING

Deposit Details

Search By Meter Number747567

Search By NameJanuary

Search

Print

Close

meterNum	month	unit	totalBill	status
747567	January	123	1376	Not Paid
747567	January	200	2146	Not Paid

Search

14-11-2024

