CHAPTER 1: INTRODUCTION

1.1 Introduction

The Water Works project represents a paradigm shift in addressing the intricate challenges associated with water supply management in rural settings. By integrating a state of-the-art tech stack and employing innovative design principles, this project seeks to redefine how water resources are distributed and managed within village communities. Introduces a level of transparency and accountability previously unseen in traditional systems.

The project aims to empower village communities by providing them with a robust, user friendly, and technologically advanced platform for managing their water resources effectively, ultimately improving the quality of life for residents. In essence, Water Works is not just a project; it's a transformative force driving positive change in the realm of rural water supply management.

1.2 Existing System and The Need For System

The existing water management system faced several challenges that impacted the efficiency and effectiveness of water supply services. Prior to the proposed improvements, users experienced significant wait times due to inefficient queue management. Manual record-keeping processes introduced errors and delays, hindering the system's ability to provide accurate and timely information. Additionally, the absence of report generation capabilities limited the authorities' ability to analyze data and make informed decisions.

Errors and delays in tracking water consumption, service requests, and resource utilization hinder the system's ability to provide accurate and up-to-date information. The manual nature of these processes also poses challenges in maintaining an organized and transparent record system.

1.3 Limitations of existing system

i. **Lack of space** – It becomes a problem in itself to find space to keep the sheets of paper being generated as a result of the ongoing discussion. The documents being generated are too important to be ill-treated.

ii. **Filing poses a problem** – Filing the documents categorically is a time consuming and tedious exercise

iv. **Reviewing becomes time-consuming** – All the process done manually at the centers and all the records are maintained on the papers. So the maintenance of the record is very difficult in the departments and as well as it's very difficult for the workers to check the record. The Existing system is paper based, time consuming, monotonous, less flexible and provides a very hectic working schedule. The chance of loss of records is high and also record searching is difficult. Maintenance of the system is also very difficult and takes lot of time.

v. **Result Processing**: is slow due to paper work and requirement of staff.

1.4 Hardware and Software Requirements

Frontend Development:

• React.js: A JavaScript library for building user interfaces.

• Redux: A predictable state container for managing application state.

• HTML/CSS/JavaScript: For creating the user interface and implementing clientside functionality.

• UI/UX Design Tools: Such as Adobe XD, Sketch, or Figma for designing wireframes and mockups. JavaScript library for building user interfaces.

Backend Development:

• Spring Boot: A Java-based framework for building enterprise-level applications.

• Hibernate: An object-relational mapping (ORM) tool for Java applications.

• JDK: Required for Java Development.

• DBMS: Such as MySQL, PostgreSQL, or MongoDB for data storage.

• RESTful APIs: For communication between the frontend and backend components.

Development Tools:

• Integrated Development Environment (IDE): Such as IntelliJ IDEA, Eclipse, or Visual Studio Code for writing and debugging code.

• Version Control System: Such as Git for collaborative development and version control.

• Dependency Management: Tools like Maven or Gradle for managing project dependencies.

• Unit Testing Framework: Such as JUnit for writing and executing unit tests.

Hardware Requirements:

- Processor : 860 Mhz onwards
- RAM : 256 MB or higher.
- Hard Disk : 10 GB or higher.
- Keyboard : 102 keys Mouse : 2 Buttons or higher

CHAPTER 2: PROPOSED SYSTEM

The proposed water management system aims to address these challenges by providing a centralized platform for monitoring, maintenance, and equitable distribution of water resources. By integrating the existing decentralized sources into a unified framework, the system seeks to enhance community engagement.

The objective of WaterWorks application is to facilitate user friendly platform by reducing the manual effort. Overall, the Water Work Flows system addresses the pressing needs of rural water supply management by improving resource allocation, enhancing transparency and accountability, prioritizing user experience, enhancing operational efficiency, promoting sustainability, and leveraging technological advancements.

2.1 Objectives of Proposed System

- Improved water distribution efficiency.
- Enhanced employee productivity and accountability.
- Financial transparency for sustainable management.
- Utilization of cutting-edge technologies for real-time monitoring.
- Empowered communities through insightful reporting tools.

2.2 Scope of the System

- **User-Centric Approach**: The Subscriber Management module prioritizes the needs of individual users, fostering a sense of ownership and community engagement. This ensures that the project directly addresses the concerns and preferences of the residents.

- **Zone Management for Targeted Distribution**: By implementing Zone Management, Water Work Flows optimizes water distribution across specific geographic areas. This targeted approach minimizes resource wastage and tailors services to the unique requirements of each zone, thereby improving overall water access.

- **Employee Efficiency and Accountability**: Employee Management ensures that the workforce responsible for water supply operations is well-organized and equipped with the necessary tools. This enhances operational efficiency, reduces downtime, and contributes to better overall water service delivery.

- **Financial Transparency with Utility Management**: The project's Utility Management module tracks income sources, expenses, and product availability. This financial transparency is crucial for sustainable water management, enabling better budgeting and resource allocation.

- **Technology-Driven Transparency**: Leveraging modern technologies like React.js, Redux, Spring Boot, and Hibernate, Water Work Flows introduces transparency and accountability to rural water management. Real-time monitoring, data analytics, and automated processes enhance the efficiency of the system.

Community Empowerment through Reporting: The comprehensive Reporting tools empower communities with insights into income, expenses, and transactions. This information facilitates informed decision-making, fosters community participation, and establishes a foundation for long-term sustainability.

CHAPTER 3: REQUIREMENT DETERMINTION AND ANALYSIS

3.1 Feasibility Study:

Market Feasibility:

- Evaluate the need for improved water supply management in rural areas.
- Analyze the target market's demographics, including population size, geographic distribution, and water usage patterns.
- Identify existing water supply systems, their shortcomings, and areas for improvement.

Technical Feasibility:

- Assess the technical requirements and capabilities needed to develop and implement the Water Work Flows system.
- Evaluate the availability of technology infrastructure (e.g., internet connectivity, hardware) in rural areas to support the system.
- Determine the feasibility of integrating modern technologies (e.g., React.js, Spring Boot) into the system architecture.

Financial Feasibility:

- Estimate the costs associated with developing, implementing, and maintaining the Water Work Flows system.
- Identify potential sources of funding, including government grants, private investment, and community contributions.
- Conduct a cost-benefit analysis to determine the potential return on investment and the system's long-term financial sustainability.

Legal and Regulatory Considerations:

- Review relevant laws, regulations, and policies governing water supply management in the target region.
- Identify any legal barriers or compliance requirements that may impact the development and implementation of the system.
- Ensure that the project complies with data protection and privacy regulations, especially concerning the collection and management of user data.

Operational Feasibility:

- Assess the capacity and capabilities of local stakeholders (e.g., water utilities, community organizations) to support the implementation and operation of the system.
- Evaluate the readiness of end-users (e.g., residents, water service subscribers) to adopt and utilize the Water Work Flows system.

- Identify any potential operational challenges or barriers and develop strategies to mitigate them.

Social and Environmental Impact:

- Consider the social and environmental implications of implementing the Water Work Flows system.
- Assess how the project may improve access to clean water resources, enhance community empowerment, and contribute to environmental sustainability.
- Identify any potential negative impacts or risks associated with the project and develop mitigation measures.

Risk Assessment:

- Identify potential risks and uncertainties that may affect the success of the project, such as technical challenges, funding constraints, or resistance from stakeholders.
- Evaluate the likelihood and potential impact of each risk and develop contingency plans to mitigate them.
- Conduct a sensitivity analysis to assess how changes in key variables (e.g., project costs, user adoption rates) may affect the project's outcomes.

Conclusion and Recommendations:

- Summarize the findings of the feasibility study and provide recommendations regarding the viability and feasibility of the Water Work Flows project.
- Outline key actions and next steps required to move the project forward, including further research, stakeholder engagement, and project planning.

3.2 System Analysis

Requirements Gathering:

- Identify and document the functional and non-functional requirements of the Water Work Flows system.
- Conduct interviews, surveys, and workshops with stakeholders to gather input and feedback on system features and functionalities.
- Prioritize requirements based on their importance to stakeholders and their alignment with project objectives.

System Architecture Design:

- Develop a high-level system architecture that outlines the components, modules, and interactions of the Water Work Flows system.

- Determine the technologies, frameworks, and platforms that will be used to build and deploy the system.
- Define data structures, database schemas, and APIs necessary to support system functionality.

Data Modeling:

- Analyze the data requirements of the system and design a data model that captures the information needed to support various system functions.
- Define entities, attributes, relationships, and constraints within the data model.
- Consider data storage, retrieval, and security requirements when designing the data model.

Process Analysis:

- Analyze the workflows and processes involved in water supply management within rural communities.
- Identify opportunities to streamline processes, eliminate redundancies, and improve efficiency through automation.
- Map out the flow of information and tasks within the system to ensure that all requirements are addressed.

User Interface Design:

- Design user interfaces (UIs) that are intuitive, user-friendly, and accessible to the target audience.

- Create wireframes, mockups, and prototypes to visualize the layout, navigation, and functionality of the system.
- Incorporate feedback from stakeholders to refine the UI design and ensure that it meets their needs and preferences.

System Integration:

- Identify any existing systems or databases that need to be integrated with the Water Work Flows system.
- Define interfaces, protocols, and data formats for exchanging information between the new system and external systems.
- Conduct compatibility testing and validation to ensure that the integrated system functions correctly and meets performance requirements.

Risk Analysis:

- Identify potential risks and uncertainties that may impact the development, implementation, or operation of the Water Work Flows system.
- Assess the likelihood and potential impact of each risk and develop mitigation strategies to address them.
- Monitor risks throughout the project lifecycle and adjust mitigation measures as needed to minimize their impact.

Cost-Benefit Analysis:

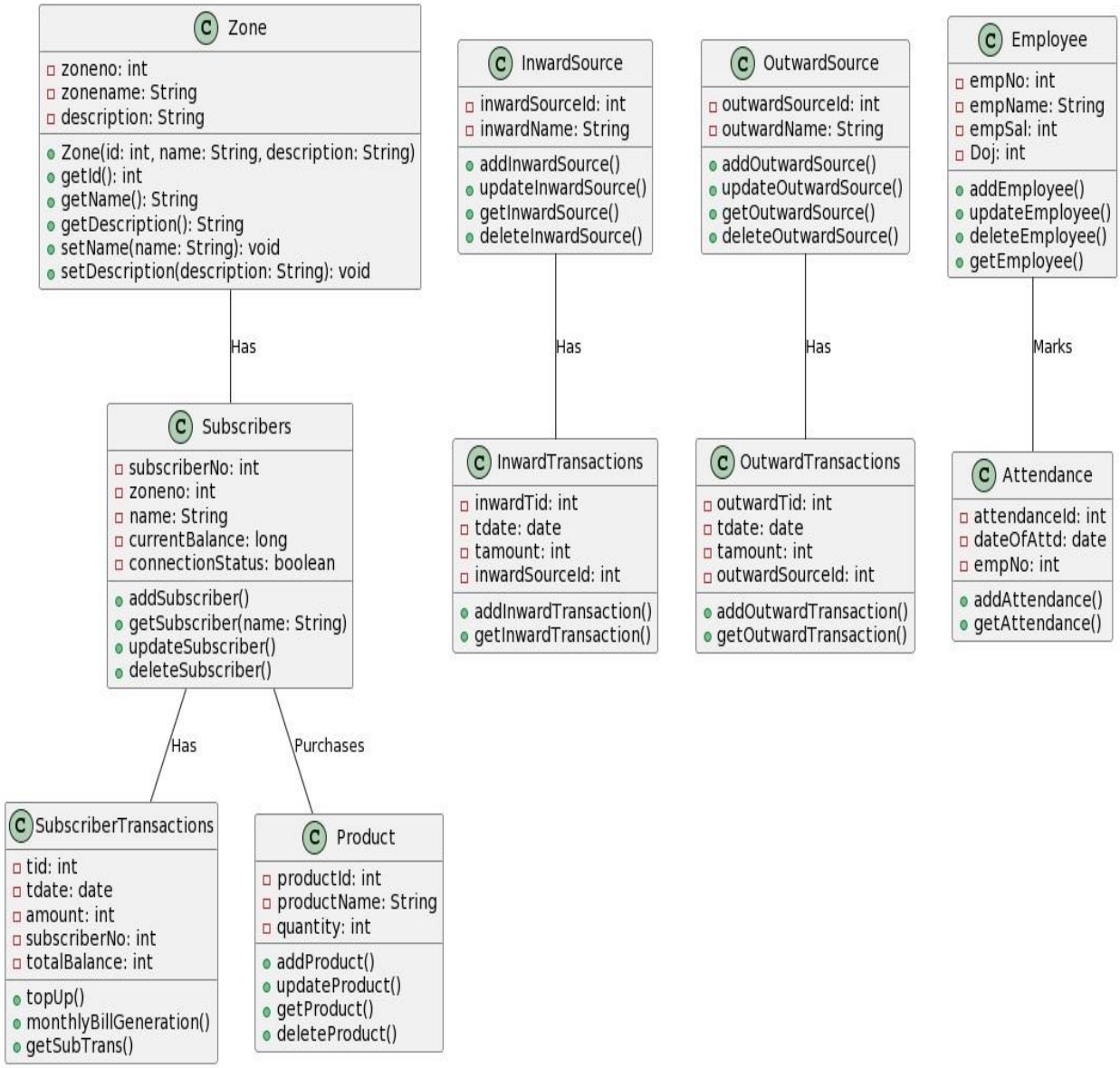
- Evaluate the costs associated with developing, implementing, and maintaining the Water Work Flows system.
- Estimate the potential benefits, such as improved water management, increased efficiency, and enhanced community empowerment.
- Conduct a cost-benefit analysis to assess the overall value proposition of the system and justify investment decisions.

By conducting a thorough system analysis, project stakeholders can gain a deep understanding of the requirements, design considerations, and potential risks associated with the Water Work Flows project. This analysis forms the foundation for successful system development and implementation, ensuring that the final product meets the needs of stakeholders and delivers tangible benefits to rural communities.

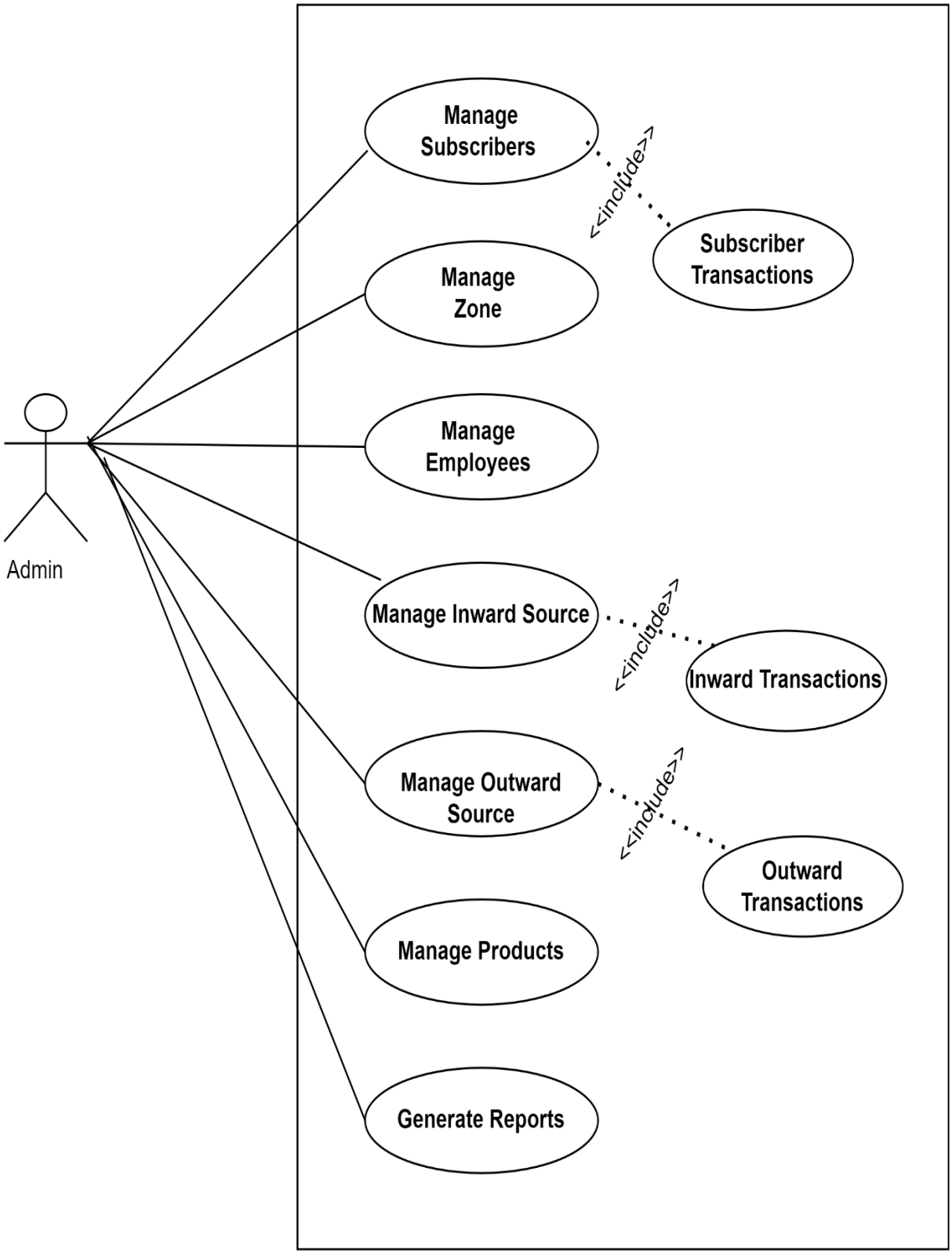
CHAPTER 4 : SYSTEM ANALYSIS AND DESIGN

4.1 Analysis Phase

4.1.1 Class Diagram:

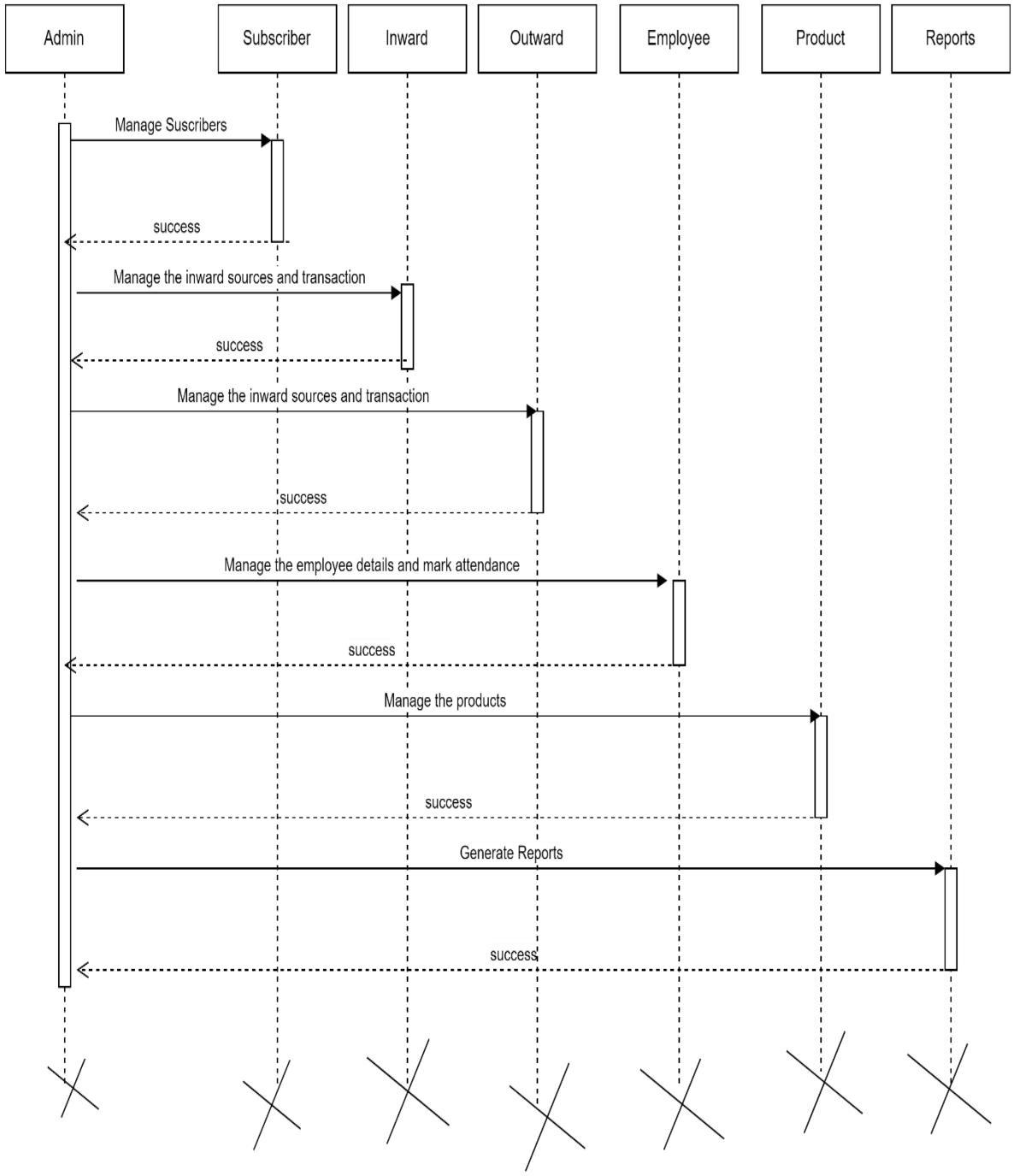


4.1.2 Use Case Diagram:

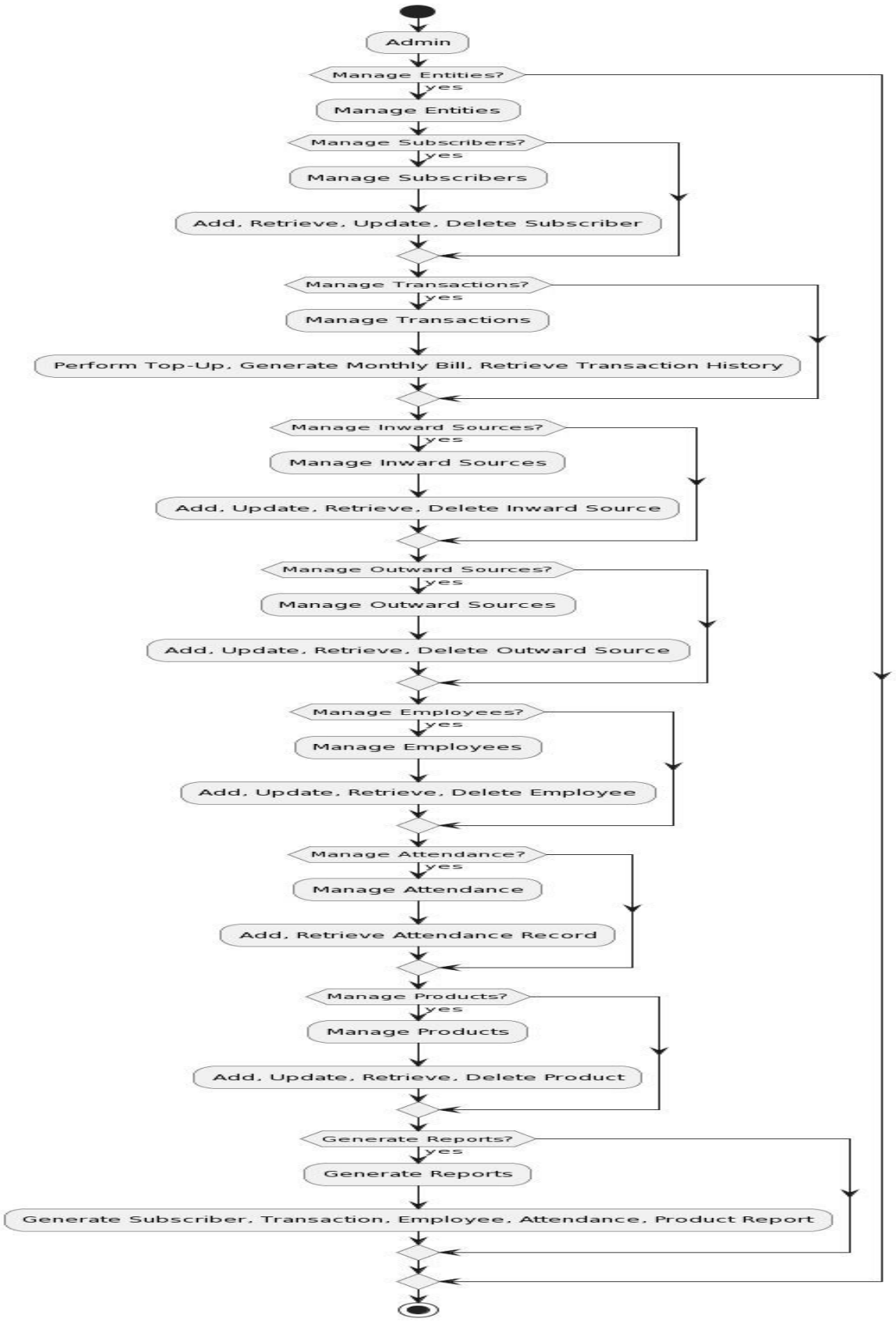


4.2 Design Phase

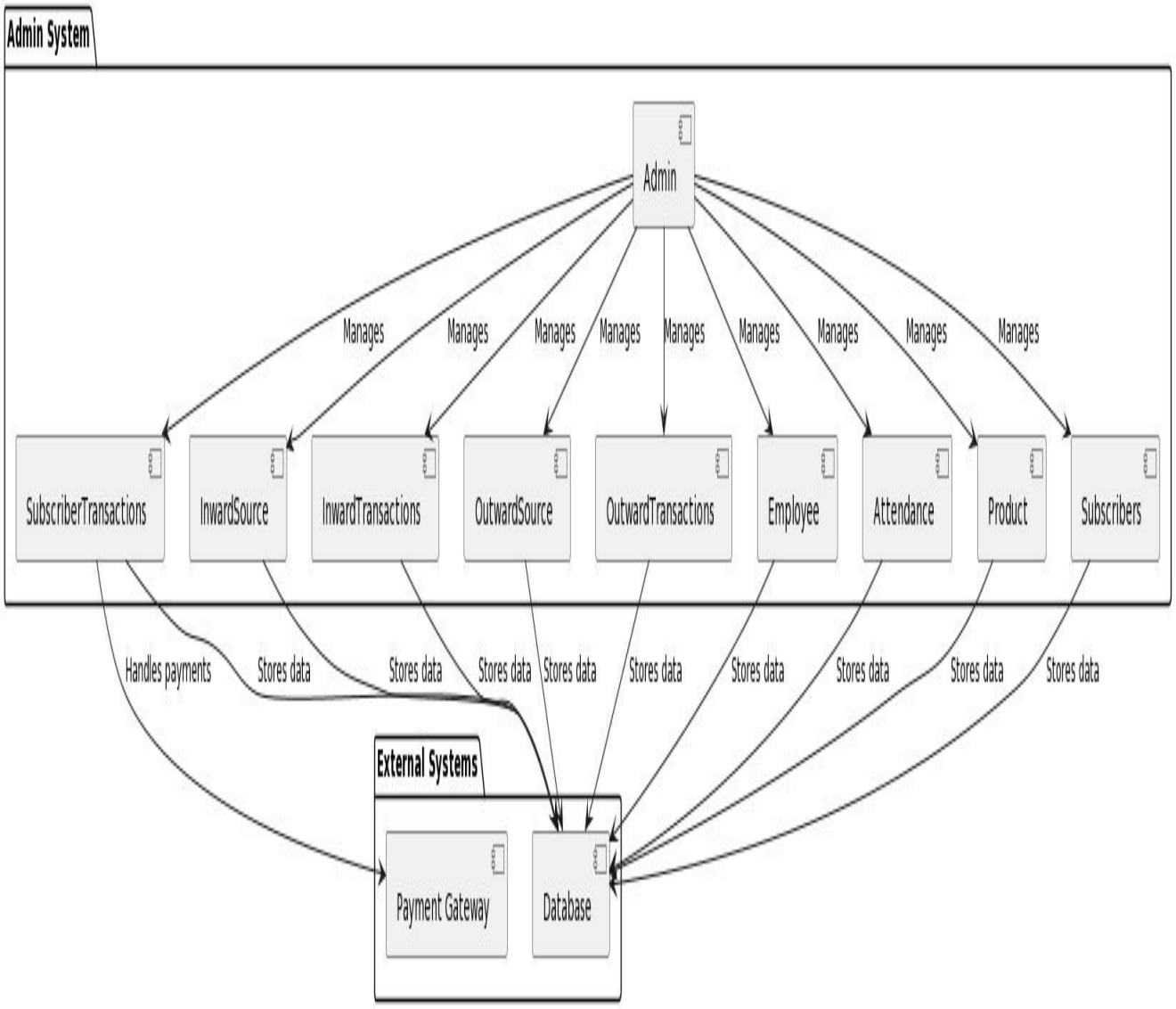
4.2.1 Sequence Diagram :



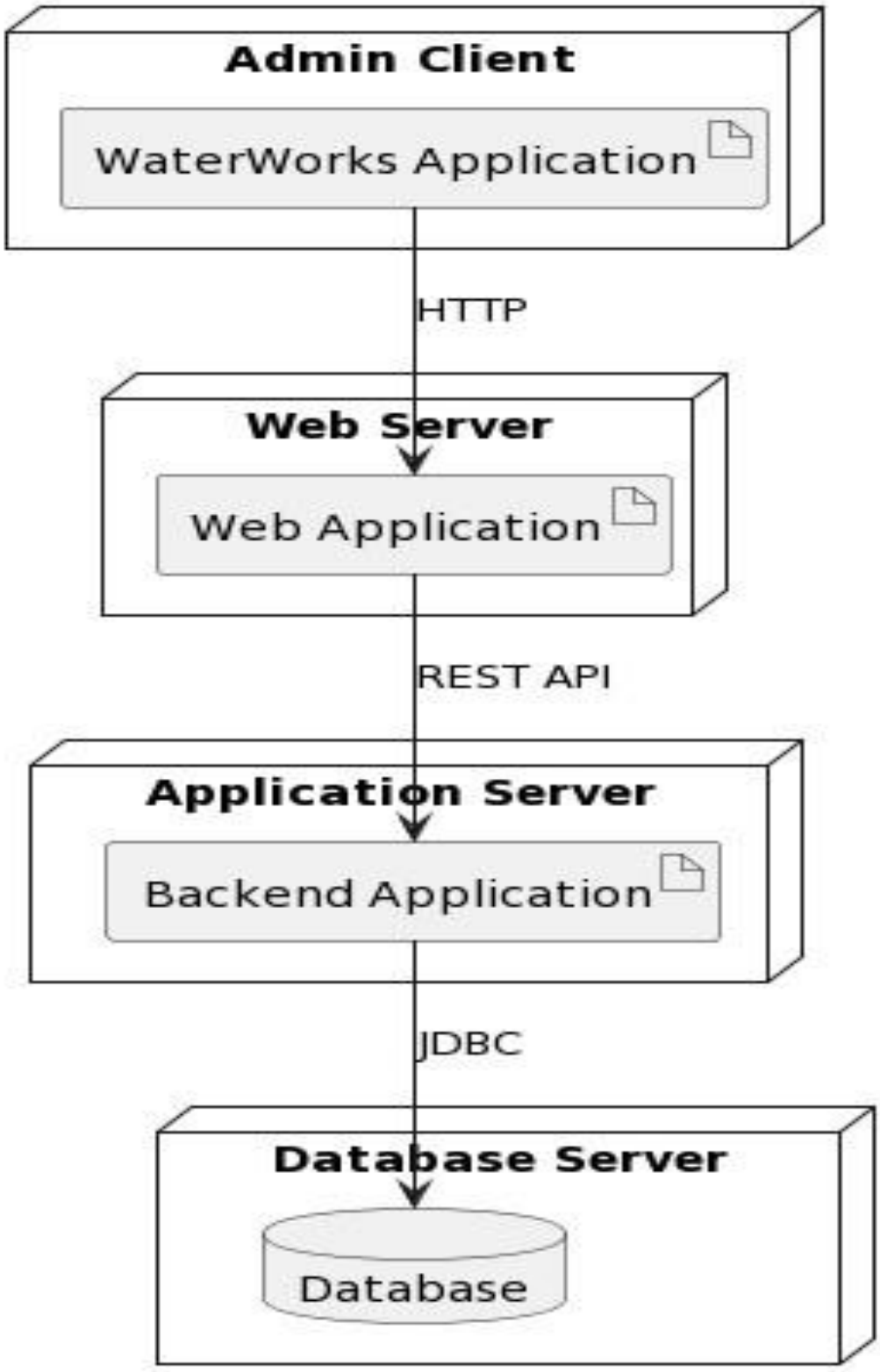
4.2.2 Activity Diagram :



4.2.3 Component Diagram :



4.2.4 Deployment Diagram :



4. 3 Module Specification

1. **SUBSCRIBER MANAGEMENT:**
    - Subscribers can easily add and view their information.
    - Provides functionalities for subscriber zone assignment and preferences.
    - Subscriber-specific features such as notifications and account details.

2. **ZONE AND PRODUCT MANAGEMENT:**
    - Defines and manages water supply zones for effective distribution.
    - Subscriber interactions are linked to specific zones, ensuring targeted communication.
    - Products: monitors the availability of water bottles and caps.

3. **EMPLOYEE MANAGEMENT:**
    - Streamlines the addition and viewing of employee details.
    - Incorporates sorting and live filtering options for efficient employee management.

4. **UTILITY MANAGEMENT:**

Inward: manages income sources, recording all incoming funds.

- Outward: tracks and records expenses, offering a comprehensive view of financial outflows.

5. **TRANSACTION MANAGEMENT:**

- Inward transactions: records all incoming cash flows, maintaining detailed transaction records.
- Outward transactions: manages and records all expenses, displaying a comprehensive view of financial outflows.

6. **TOP-UP AND SUBSCRIBER TRANSACTIONS:**

- Verifies users and facilitates account recharge through top-ups.
- Users can conveniently add funds by providing their subscriber number.
- Displays the subscriber transactions.

7. **REPORTS:**

- Income report: generates a detailed report of all income transactions.
- Expense report: provides an overview of all expenses incurred.

- Transaction report: displays detailed transaction information, including debits and credits.

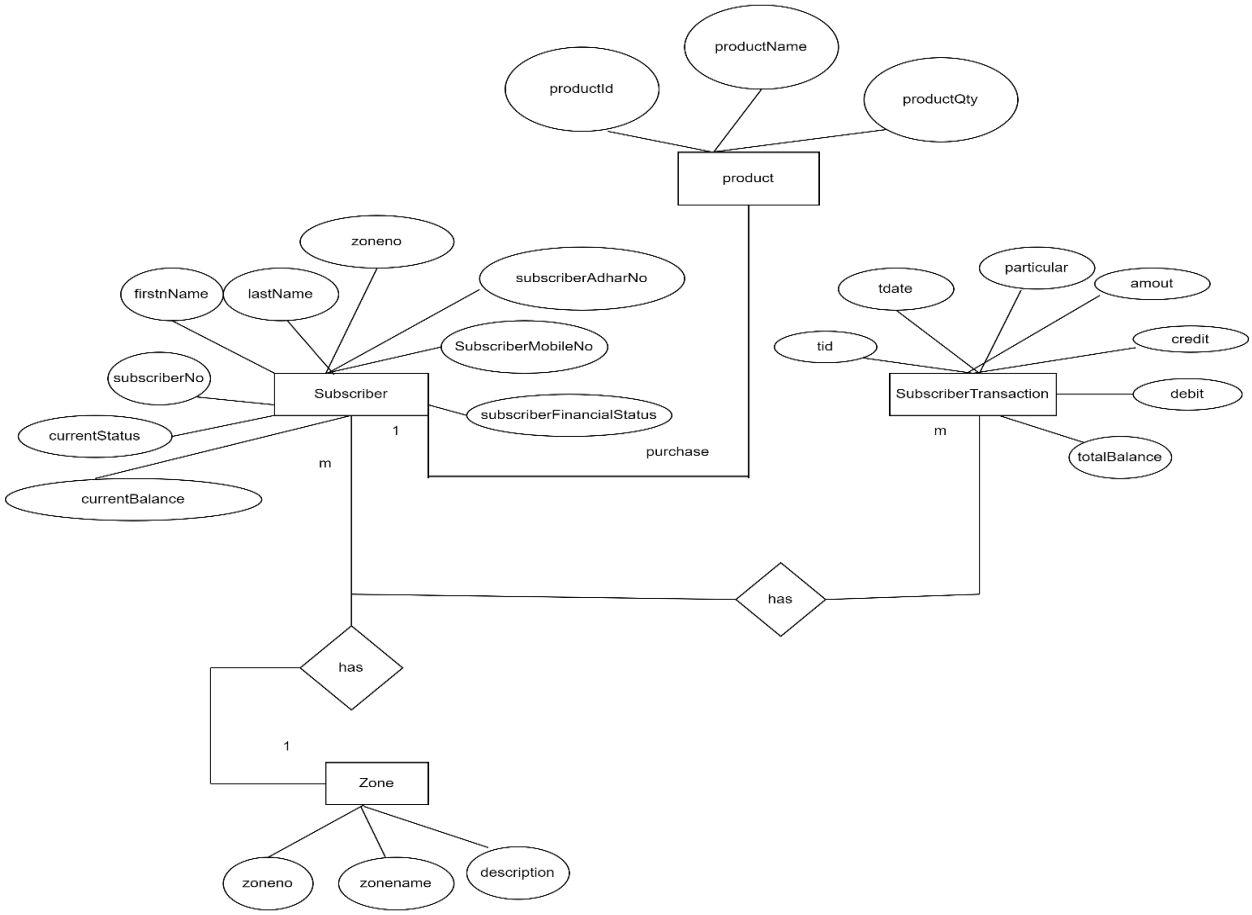
8. THIRD-PARTY MANAGEMENT:

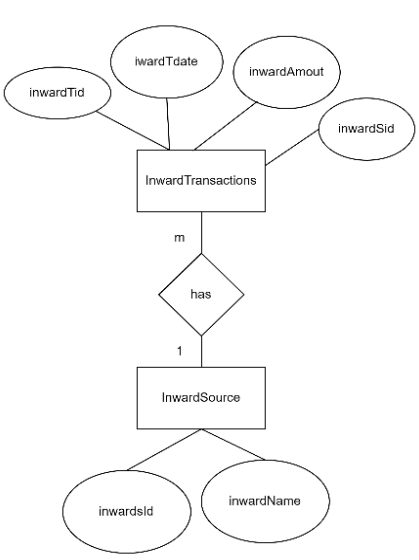
- Contains details of third-party workers such as plumbers and electricians.

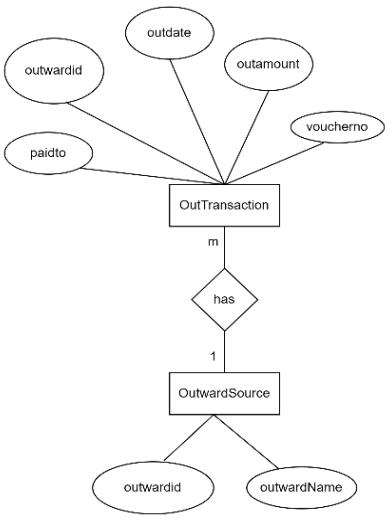
Streamlines the process of assigning and managing external services.

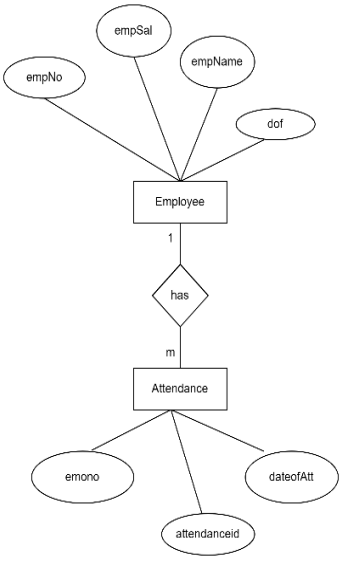
CHAPTER 5 : BACKEND SPECIFICATIONS

5. 1 ERD Diagram:









5. 2 Data Dictionary:

Table 1	Relationship	Table 2
Zone	1 – M	Subscriber
Subsciber	1 – M	Subscriber Transaction
Employee	1 – M	Attendance
Inward Source	1 – M	Inward Transaction
Outward Source	1 – M	Outward Transaction
Product	-	-
Third Party Contacts	-	-

Subscriber

Attribute	Data Type	Constraint	Description
subscriberNo	integer	Primary Key	Unique Identifier for each subscriber
lastName	varchar(40)	Not null	Last name of the subscriber
firstName	varchar(40)	Not null	First name of the subscriber
address	varchar(50)	Not null	Address
mobileno	varchar(10)	Not null	Mobile no of the subscriber
zoneno	integer	Foreign Key	To which zone the subscriber belongs
currentBalance	varchar(10)	Not null	Current balance of the Subscriber
connectionStatus	varchar(10)	Not null	Connection Status(active,blocked,disconnected)
financialStatus	boolean	Not null	Financial status (rich or poor)

Zone

Attribute	Data Type	Constraint	Description
zoneno	integer	Primary Key	Unique identifier for each zone
zoneName	varchar(40)	Not null	Name of the Zone

Subscriber Transaction

Attribute	Data Type	Constraint	Description
subtransId	integer	Primary Key	Unique Identifier for each subscriber transaction
tdate	date	Not null	Date when the transaction took place
tamount	varchar(40)	Not null	Amount to be added
particular	varchar(50)	Not null	Details of the transaction
totalBalance	varchar(10)	Not null	Total balance of the subscriber
subscriberNo	integer	Foreign Key	Links to the subscriber table

---

Inward Source

Attribute	Data Type	Constraint	Description
inwardSid	integer	Primary Key	Unique Identifier for each inward source
inwardName	varchar(40)	Not null	Name of the inward source

Inward Transaction

Attribute	Data Type	Constraint	Description
inwardTid	integer	Primary Key	Unique Identifier for each inward Transaction
tdate	date	Not null	Transaction date
tamount	varchar(20)	Not null	Amount to be transacted
inwardSid	integer	Foreign Key	Links to the inward source table.

Outward Source

Attribute	Data Type	Constraint	Description
outwardSid	integer	Primary Key	Unique Identifier for each outward source
outwardName	varchar(40)	Not null	Name of the outward source

Outward Transaction

Attribute	Data Type	Constraint	Description
outwardTid	integer	Primary Key	Unique Identifier for each outward Transaction
tdate	date	Not null	Transaction date
outamount	varchar(20)	Not null	Amount to be transacted
paidto	Varchar(40)	Not null	The details to whom the amount was paid
outwardSid	integer	Foreign Key	Links to the outward source table.

Employee

Attribute	Data Type	Constraint	Description
empNo	integer	Primary Key	Unique Identifier for each employee
Name	varchar(40)	Not null	Name of the employee
empDoj	date	Not null	Date of joining
address	varchar(50)	Not null	Address
mobileno	varchar(10)	Not null	Mobile no of the employee
salary	Varchar(30)	Not null	Salary of the employee

Attendance

Attribute	Data Type	Constraint	Description
attendanceId	integer	Primary Key	Unique Identifier for each attendance
dateOfAttd	date	Not null	Date of the Attendance
intime	time	Not null	Intime of the employee
outtime	time	Not null	Out time of the employee
Status	varchar(10)	Not null	Status whether the employee is present or absent
empNo	integer	Foreign Key	Links to the employee table

\

Product

Attribute	Data Type	Constraint	Description
productId	integer	Primary Key	Unique Identifier for each product
productName	Varchar(25)	Not null	Name of the product
quantity	Varchar(10)	Not null	Quantity of the product

Third Party Details

Attribute	Data Type	Constraint	Description
contactId	integer	Primary Key	Unique Identifier for each contacts
name	Varchar(50)	Not null	Name of the contact
mobileno	Varchar(10)	Not null	Mobile number
Profession	Varchar(20)	Not null	Profession of the contact

5. 3 Test Cases:

ADD SUBSCRIBERS

Sr. No.	Action	Expected Output	Actual Output	Status
1	Valid subscriber details	Display message: Record inserted successfully	Record Inserted Successfully	Pass
2	Invalid subscriber details	Display message: Record not inserted successfully	Record Not Inserted Successfully	Fail
3	Empty subscriber details	Should not be processed	Highlight empty field	Fail

UPDATE SUBSCRIBER

Sr. No.	Action	Expected Output	Actual Output	Status
1	Valid id entered	Fill remaining fields automatically	Fields are filled automatically	Pass
2	Invalid id entered	Display message: subscriber record is not present	Id Is Not Present In Database	Fail
3	Empty subscriber details	Should not be processed	Highlight empty field	Fail
4	Update subscriber details	Display message: Record updated successfully	Record Updated Successfully	Pass

DELETE SUBSCRIBER

Sr. No.	Action	Expected Output	Actual Output	Status
1	Valid id entered	Fill remaining fields automatically	Fields are filled automatically	Pass
2	Invalid id entered	Display message: Employee record is not present	ID Is Not Present In Database	Fail
3	Empty subscriber details	Should not be processed	Highlight empty field	Fail

4	Delete subscriber details	Display message: Record deleted successfully	Record Deleted Successfully	Pass

VIEW ALL THE SUBSCRIBERS

Sr. No.	Action	Expected Output	Actual Output	Status
1	Valid id or subscriber name entered	Display subscriber details	Display subscriber details	Pass
2	Invalid id or subscriber name entered	Display message:subscriber record is not present	No Record Found With Provided Information	Fail

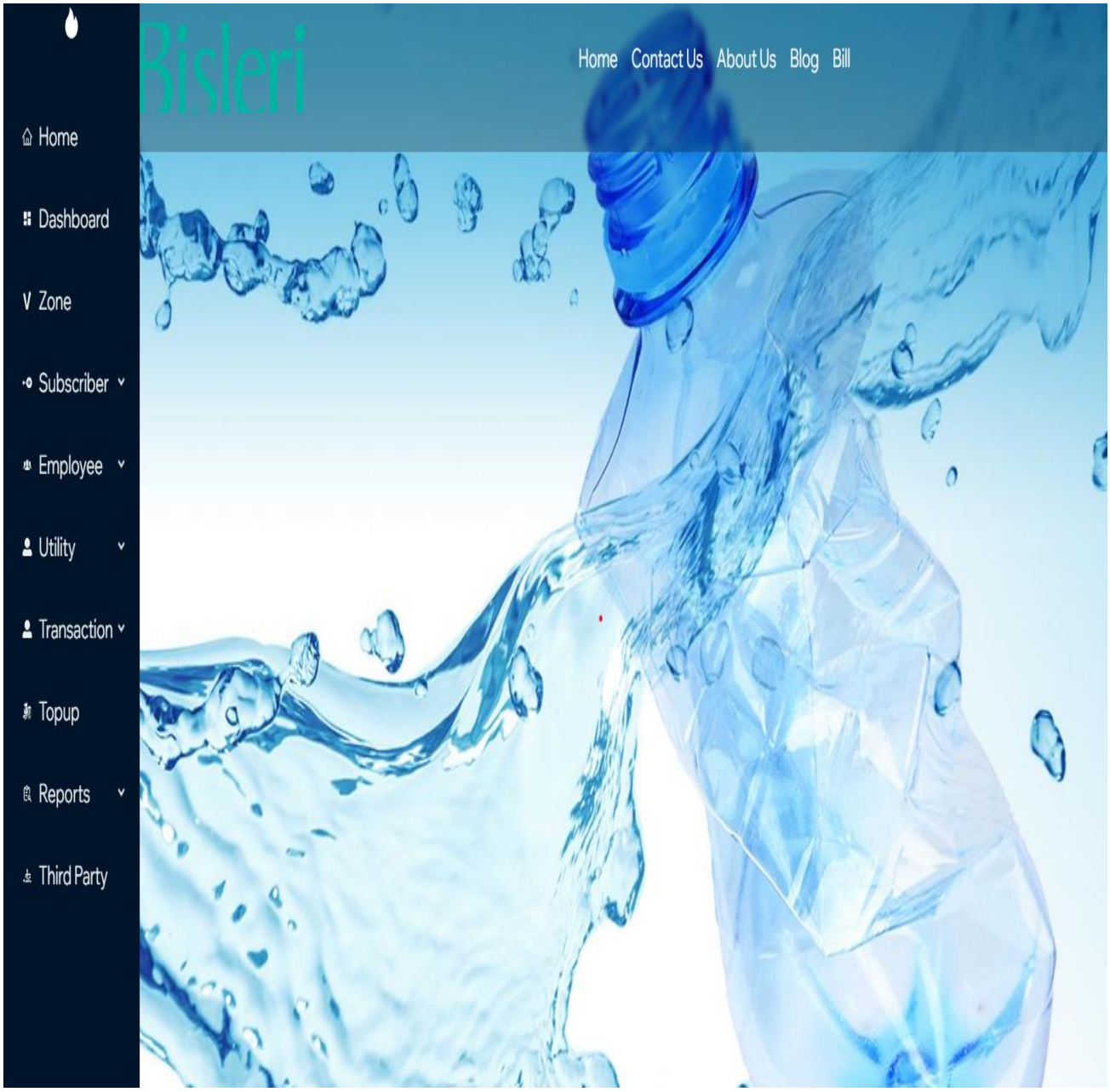
VIEW THE REPORTS OF THE SUBSCRIBER

Sr. No.	Action	Expected Output	Actual Output	Status
1	Select the id you want to generate the report	Generated	Report generated	Pass
2	Invalid details entered	Display message: Check valid details	Check Valid Details	Fail

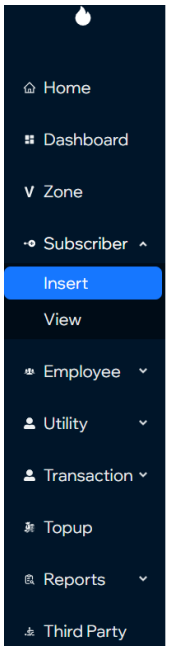
CHAPTER 6 :FRONTEND SPECIFICATIONS

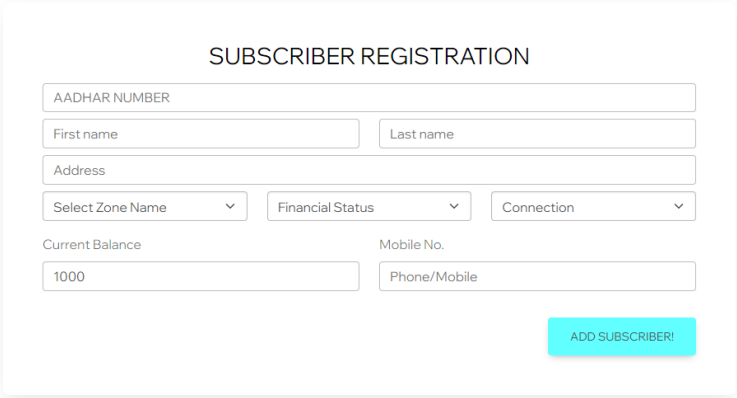
6.1 Input/Output Screens:

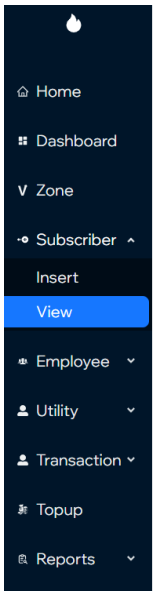
Home Page:

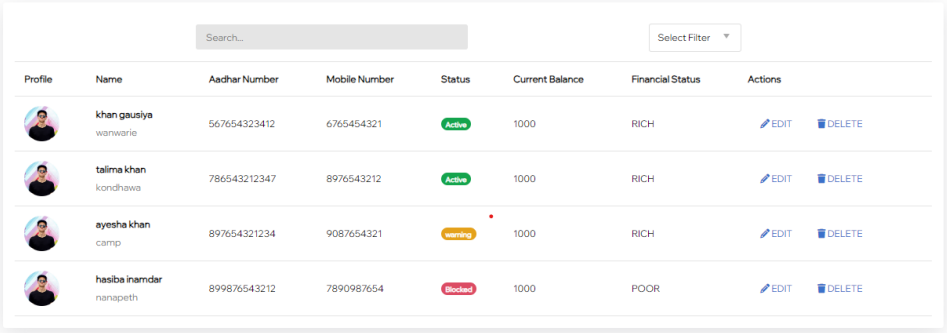


Subscriber
Insert/view

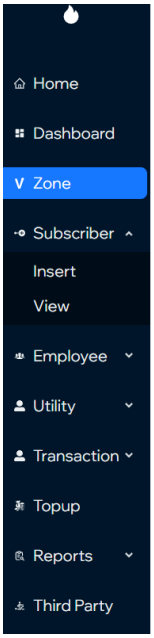


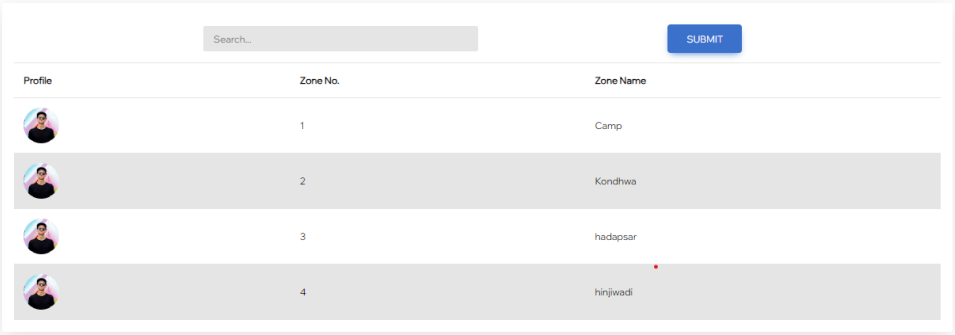




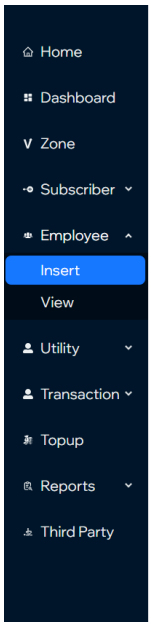


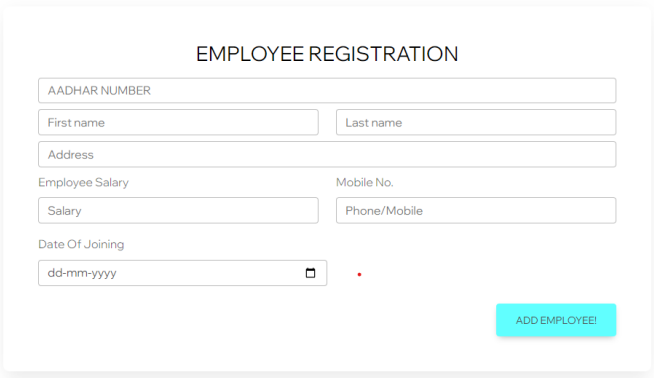
Zone

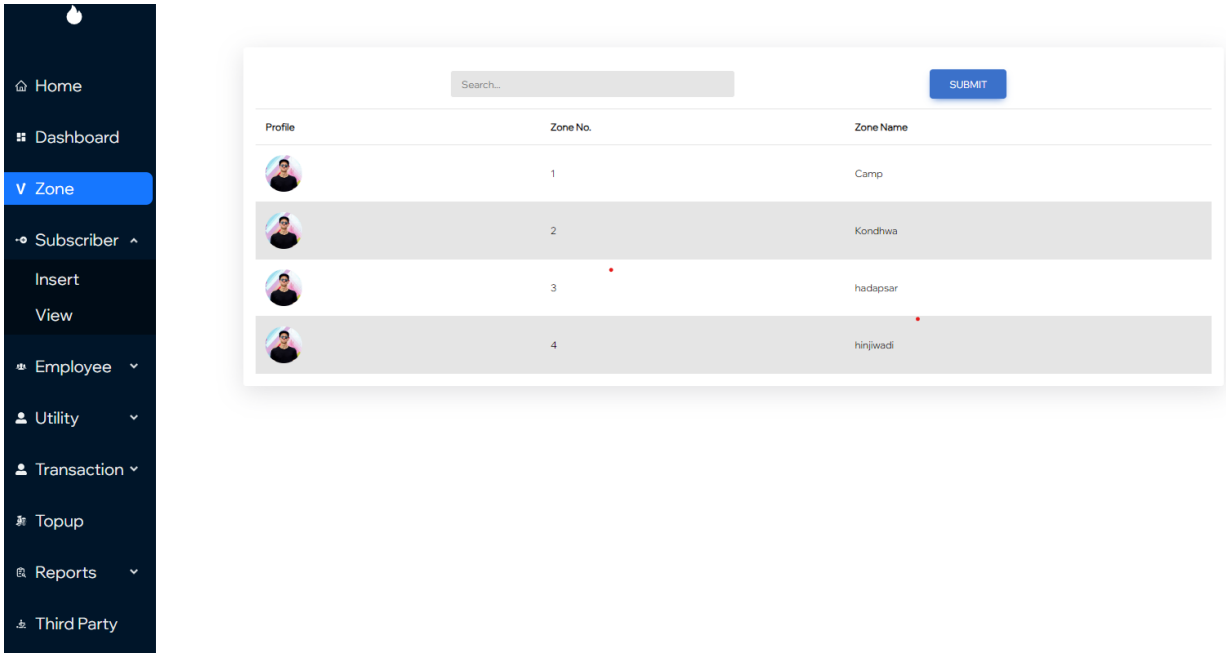




Employee: Insert/view

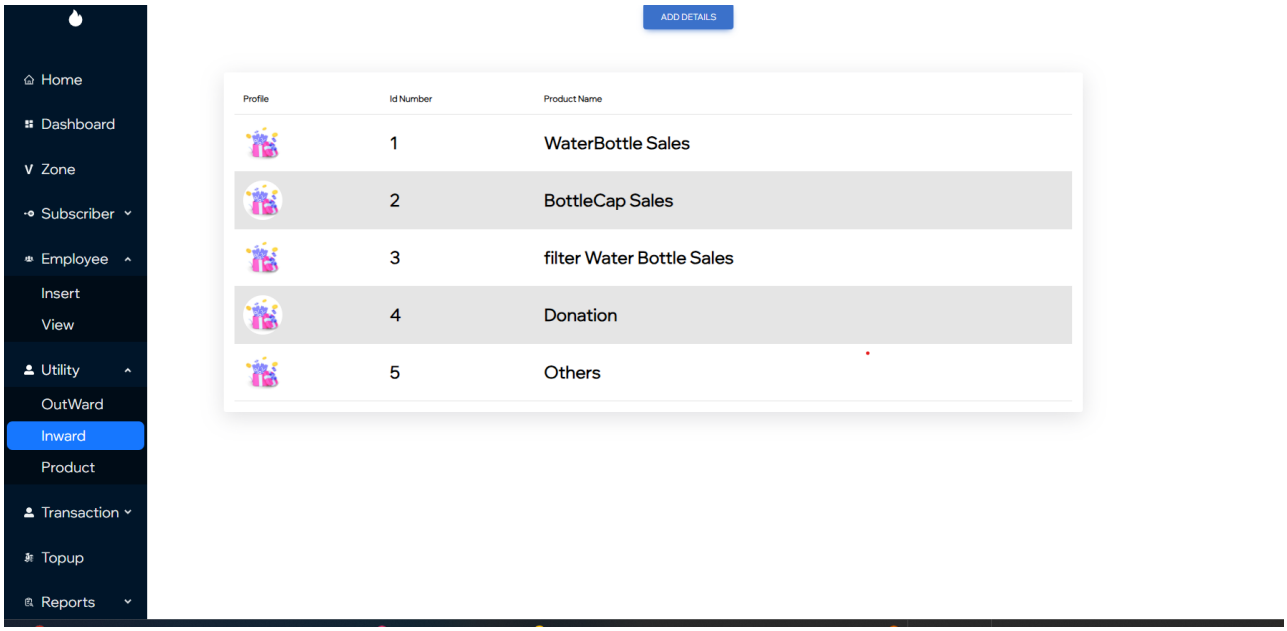


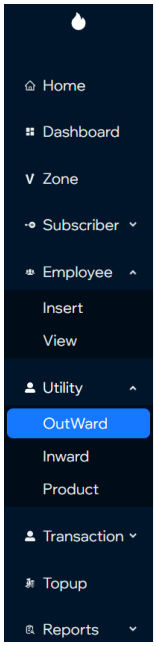




Utility:

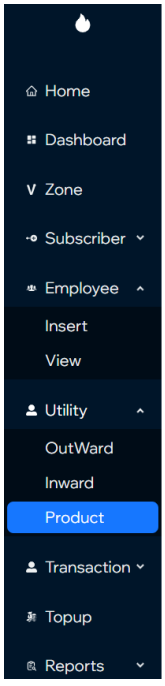
Inward Source/Outward Source/Product





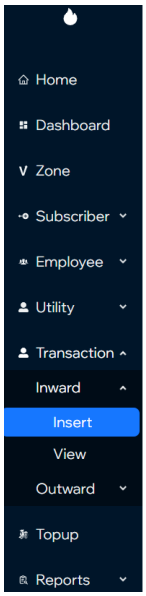
ADD DETAILS

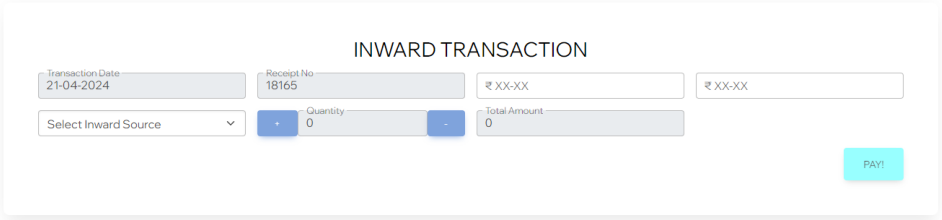
Profile	Id Number	Product Name
	1	WaterBottle Purchase
	2	BottleCap Purchase
	3	Electricity
	4	Emp Salary
	5	Others

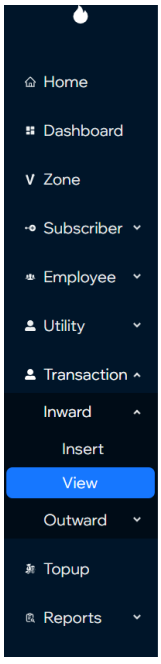


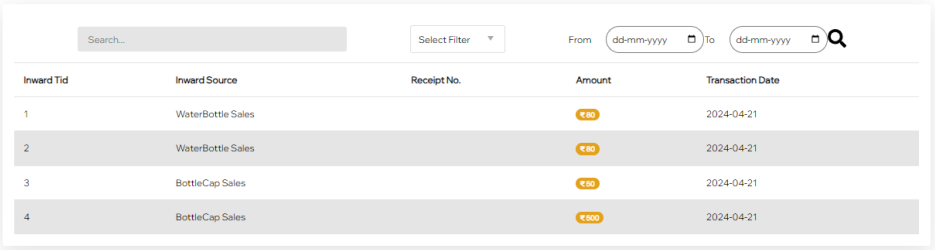
Profile	Id Number	Product Name	Quantity
	1	WaterBottle	104
	2	BottleCap	20

Transactions:
Inward Trans
Insert/view

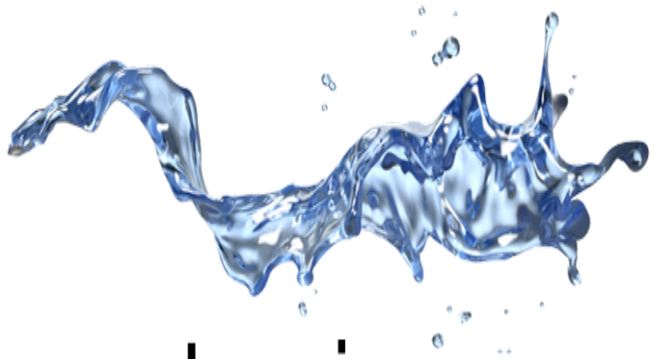








Inward Report pdf:



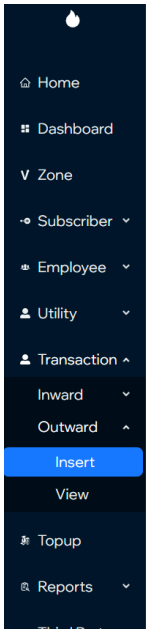
Invoice

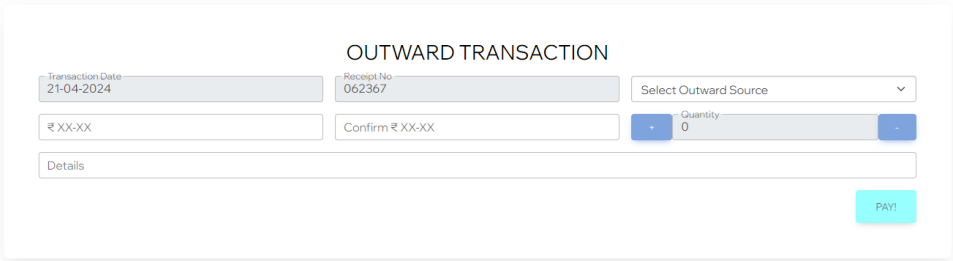
Inward Tid	Inward Source	Receipt No.	Amount	Transaction Date
1	WaterBottel Sales		160	2024-02-24
2	WaterBottel Sales		160	2024-02-24
3	BottleCap Sales		20	2024-02-24
4	BottleCap Sales		20	2024-02-24
5	BottleCap Sales		20	2024-02-24
6	BottleCap Sales		20	2024-02-24
7	BottleCap Sales		20	2024-02-24
8	BottleCap Sales		20	2024-02-24
9	BottleCap Sales		20	2024-02-24
10	BottleCap Sales		20	2024-02-24
11	BottleCap Sales		20	2024-02-24
12			20	2024-02-24
13			20	2024-02-24

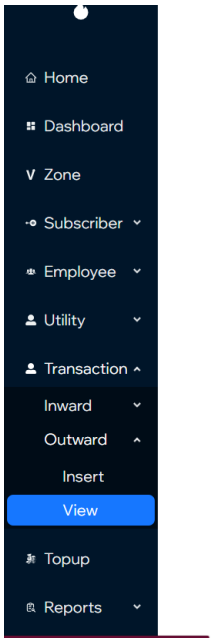
Authorized Signature

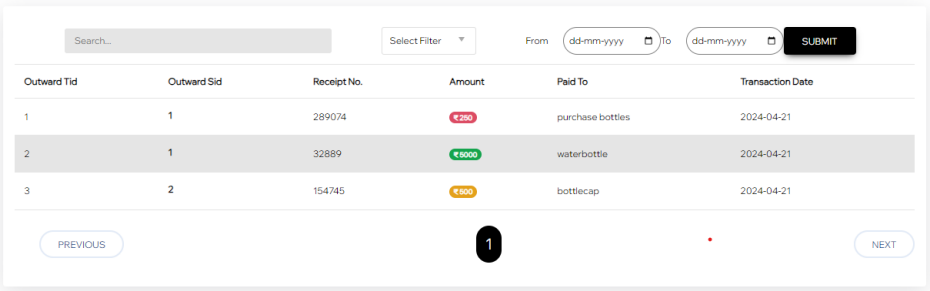
Outward:

Insert/view

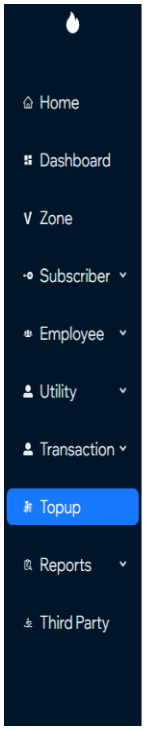


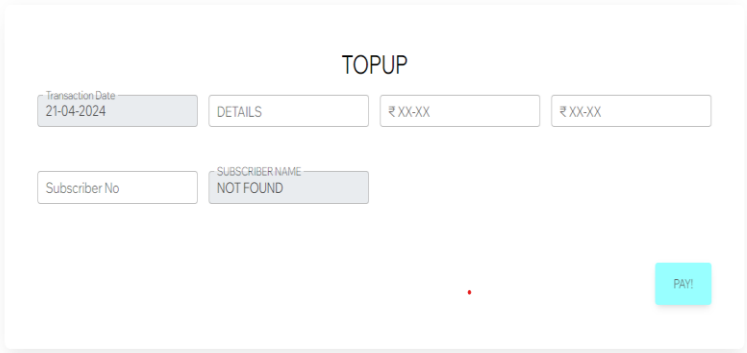




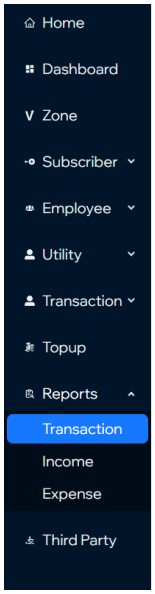


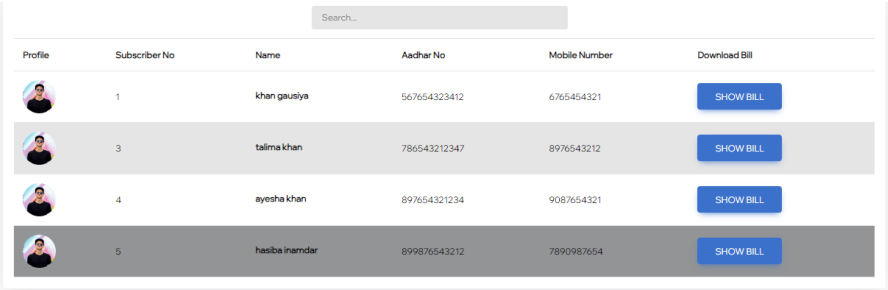
TopUp:



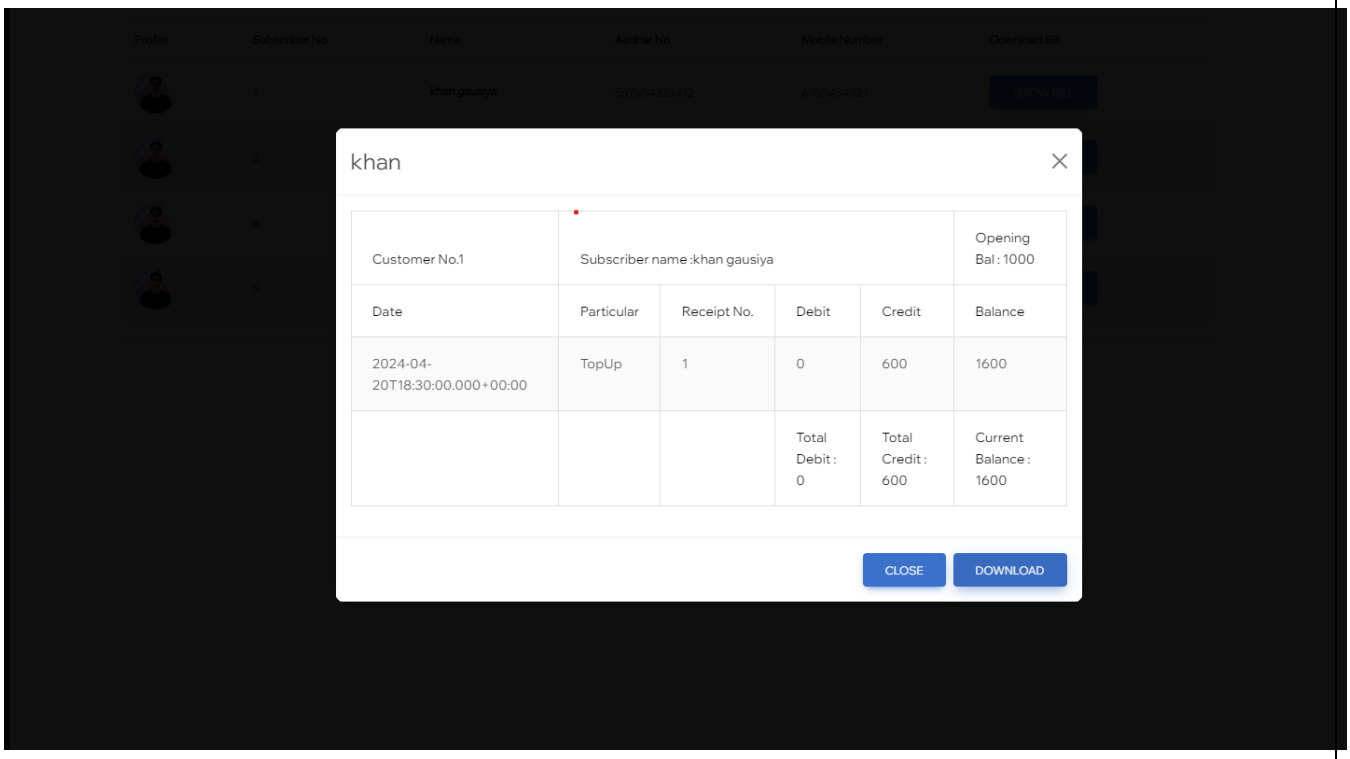


Report:
Transaction





Show Subscriber bills of transaction:



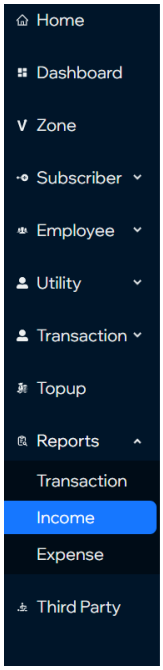
Downloaded Pdf:

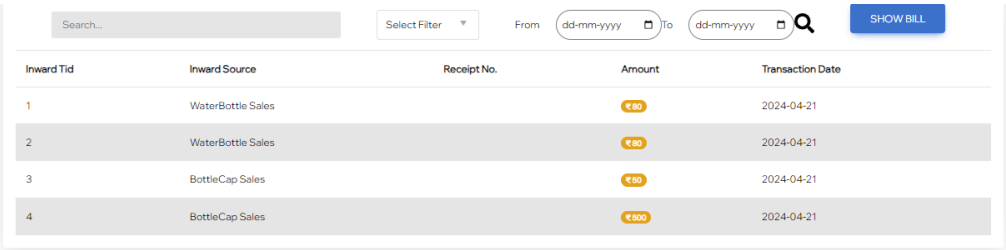
Invoice

Customer No.1	Subscriber name :khan gausiya				Opening Bal : 1000
Date	**Particular**	**Receipt No.**	**Debit**	**Credit**	**Balance**
2024-04-20T18:30:00.000+00:00	TopUp	1	0	600	1600
			Total Debit : 0	Total Credit : 600	Current Balance : 1600

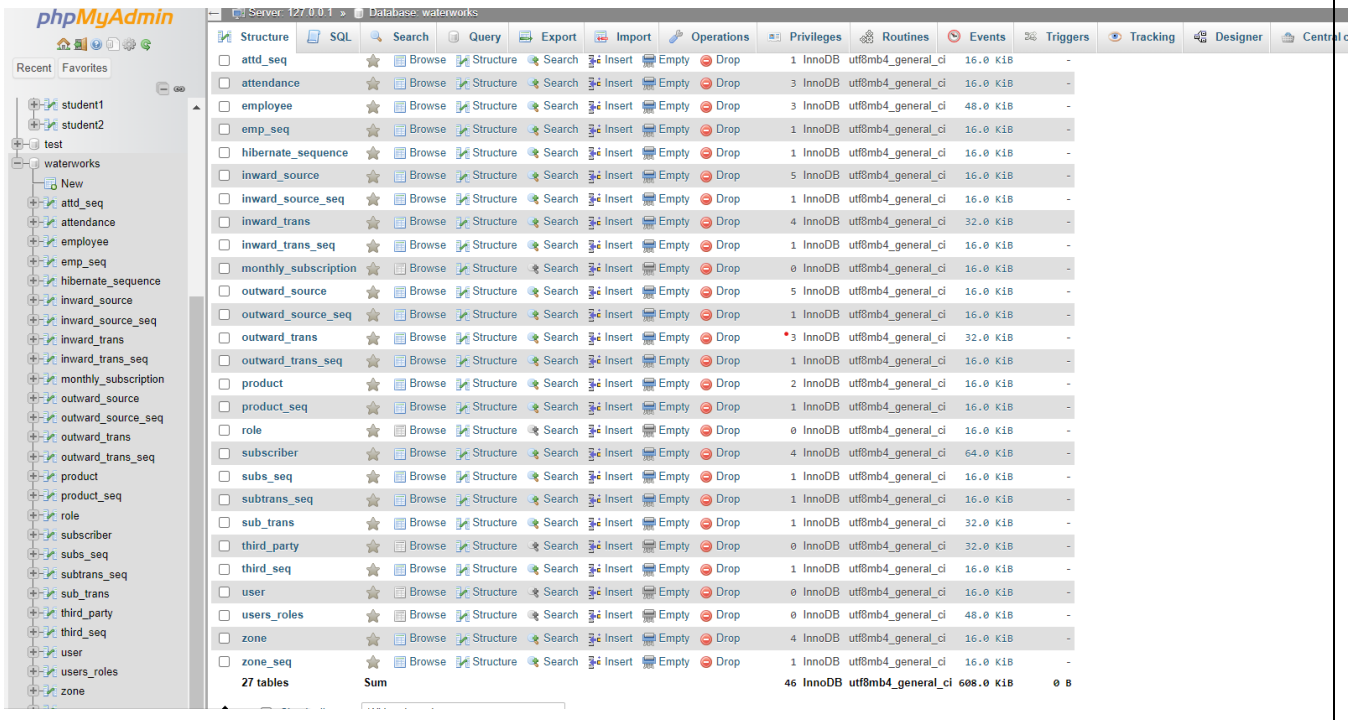
Authorized Signature

Income:





Database:



6.2 Sample Code:

```
@RestController
@RequestMapping("/waterwork")
@CrossOrigin("*")
public class WaterWorkController {

	@Autowired
	EmployeeService employeeService;

	@Autowired
	SubsService subsService;

	@Autowired
	ZoneService zoneService;

	@Autowired
	ThirdPartyService tpservice;

	@Autowired
	MonthlySubService msservice;

	@Autowired
	AttdService attdserv;

	@Autowired
	ProductService productService;


	//@Value("${myname}")
	//private String myName;


	// employee controller

	@RequestMapping("/welcome")
	public String greet()
	{
		return "login successfully";
	}
	@RequestMapping("/accessDenied")
	public String accessDenied()
	{
		return " accessDenied 403 ";
	}

	@RequestMapping("/add/insertEmployee")
	public Employee insertCourse(@RequestBody Employee employee) {

		return employeeService.insertEmployee(employee);
	}

	@RequestMapping("/delete/deleteEmployeeById")
	public String deleteEmployeeById(@RequestParam("empNo") long empNo) {
		employeeService.deleteEmployeeById(empNo);
		return " Record Deleted Succesfully";
	}
```

```
	@RequestMapping("/update/updateEmployeeById")
	public Employee updateEmployeeById(@RequestParam("empNo") long empNo,
@RequestBody Employee employee) {
		return employeeService.updateEmployeeById(empNo, employee);
	}

	@RequestMapping("/get/getEmployeeByName")
	public Employee getEmployeeByName(@RequestParam("empName") String
empName) {
		return employeeService.getEmployeeByName(empName);
	}

	@RequestMapping("/get/getByEmployeeNo")
	public Optional<Employee> getByEmployeeNo(@RequestParam("empNo") long
empNo) {
		return employeeService.getByEmployeeNo(empNo);
	}

	@RequestMapping("/get/getEmployeeByAdharNo")
	public Employee getEmployeeByAdharNo(@RequestParam("empAdharNo") long
empAdharNo) {
		return employeeService.getEmployeeByAdharNo(empAdharNo);
	}

	@RequestMapping("/get/getEmployeeByMobileNo")
	public Employee getEmployeeByMobileNo(@RequestParam("empMobileNo")
long empMobileNo) {
		return employeeService.getEmployeeByMobileNo(empMobileNo);
	}

	@RequestMapping("/get/getAllEmployee")
	public List<Employee> getAllEmployee() {
		return employeeService.getAllEmployee();
	}
	// gausiya//subscriber

	@RequestMapping("/add/addSubscriber")
	public Subscriber addSubscriber(@RequestBody Subscriber subscriber) {
		return subsService.addSubscriber(subscriber);
	}

	@RequestMapping("/get/getSubscriberByNo")
	public Subscriber getSubscriberByNo(@RequestParam("subscriberNo") int
subscriberNo) {
		return subsService.getSubscriberByNo(subscriberNo);
	}

	@RequestMapping("/get/getAllSubscriber")
	public List<Subscriber> getAllSubscriber() {
		return subsService.getAllSubscriber();
	}

	@RequestMapping("/get/getSubscribersbyzoneno")
	public List<Subscriber> getSubscriberByZoneno(@RequestParam("zoneno")
int zoneno) {
		return subsService.getSubscriberByZoneno(zoneno);
	}

	@RequestMapping("/get/getSubscriberByName")
```

```
	public Subscriber
getSubscriberByName(@RequestParam("subscriberFirstName") String
subscriberFirstName) {
		return subsService.getSubscriberByName(subscriberFirstName);
	}

	@RequestMapping("/get/getSubscriberBymobileno")
	public Subscriber
getSubscriberByMobileNo(@RequestParam("subscriberMobileNo") long
subscriberMobileNo) {
		return subsService.getSubscriberByMobileNo(subscriberMobileNo);
	}

	@RequestMapping("/get/getSubscriberByAdharNo")
	public Subscriber
getSubscriberByAdharNo(@RequestParam("subscriberAdharNo") long
subscriberAdharNo) {
		return subsService.getSubscriberByAdharNo(subscriberAdharNo);
	}

	@RequestMapping("/delete/deleteSubscriberById")//problem
	public String deleteSubscriber(@RequestParam("subscriberNo") int
subscriberNo) {
		//return "record deleted successfully";
		subsService.deleteSubscriber(subscriberNo);
		return "record deleted successfully";

	}



	@RequestMapping("/update/UpdatedRecord/{subscriberNo}")
	public  Subscriber updatedRecord(@PathVariable("subscriberNo") int
subscriberNo,
			@RequestBody Subscriber subscriberUser) {
		//return "update successfully ";
		return subsService.updatedRecord(subscriberNo, subscriberUser);

	}


	@RequestMapping("/delete/deleteSubscriberByadharno")
	public String deleteSubscriber(@RequestParam("subscriberAdharNo")
long subscriberAdharNo) {
		subsService.deleteSubscriberByAdharNo(subscriberAdharNo);
		return "record deleted successfully";

	}


	@RequestMapping("/get/getCrbBySubno")
	public int getCrbBySubno(@RequestParam("subscriberNo") int
subscriberNo) {
	return	subsService.getCrbBySubno(subscriberNo);


	}

	//Defaulters

	@RequestMapping("/get/getdefaultersByAmt")
```

```
	public List<Subscriber> getdefaultersByAmt(@RequestParam("amt") int
amt) {
	return	subsService.getdefaultersByAmt(amt);
		}

	@RequestMapping("/get/getdefaultersBySurname")
	public List<Subscriber>
getdefaultersBySurname(@RequestParam("lastName") String lastName) {
	return	subsService.getdefaultersBySurname(lastName);
		}

	@RequestMapping("/get/getdefaultersByZone")
	public List<Subscriber> getdefaultersByZone(@RequestParam("zoneno")
int zoneno) {
	return	subsService.getdefaultersByZone(zoneno);
		}



	// ZONE

	@RequestMapping("/add/addZone")
	public String addZone(

			@RequestParam("zonename") String zonename)

	{
		Zone zone = new Zone();
		zone.setZonename(zonename);
		zoneService.addZone(zone);
		return ("inserted successfully");

	}

	@RequestMapping("/get/getZone")
	public List<Zone> getZone() {
		return zoneService.getAllZone();
	}


	// ThirdParty // MonthlySubscription

	@RequestMapping("/add/addThirdParty")
	public ThirdParty addThirdParty(@RequestBody ThirdParty
addThirddParty) {
		return tpservice.addThirdParty(addThirddParty);

	}
```

CHAPTER 7: LIMITATIONS AND FUTURE ENHANCEMENTS

7.1 LIMITATIONS AND FUTURE SCOPE:

This project has a vast scope as many other functionalities can be added to it ,Also, to prevent this many other functionalities can be added to the user side especially to the subscriber like the subcriber has access to their account and can make the payment by themselves rather than contacting the admin.More creative way of showing reports and futher more modules can be added as per requirement. Payment Gateway can be added for ease of payment.

7.2 Conclusion:

After completion of this project we have concluded that this web application works as per the need and requirement of the client and is user friendly. Also this Water Works project helped me to understand the design, code and implementation processes which are performed while making any project. Many concepts were revised and many of them were very new which were learnt in making of this web application.

7.3 Bibliography

React.js Documentation –

- Official documentation for React.js library.
- Website: https://reactjs.org/docs/getting-started.html

Redux Documentation –

- Official documentation for Redux state management library.
- Website: https://redux.js.org/introduction/getting-started

Spring Boot Documentation –

- Official documentation for Spring Boot framework.
- Website: https://spring.io/projects/spring-boot

Hibernate Documentation –

- Official documentation for Hibernate ORM framework.
- Website: https://hibernate.org/orm/documentation/

"Water Supply and Sanitation in Rural Areas" by World Health Organization (WHO) and UNICEF.

- Report providing insights into challenges and best practices in rural water supply management.
- "Rural Water Supply Management: Challenges and Innovations"

- Academic paper or book discussing challenges and innovative approaches in rural water supply management.

"Community Engagement in Water Resource Management"

- Academic paper or book discussing the importance of community engagement in water resource management.

Interviews and consultations with experts in the field of water supply management, technology, and community development