



**KLE Technological
University** | Creating Value
Leveraging Knowledge

Billboard Management System

MASTER OF COMPUTER APPLICATIONS

of

KLE TECHNOLOGICAL UNIVERSITY

by

Sadanand M

**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
KLE TECHNOLOGICAL UNIVERSITY**

Vidyanagar, Hubballi-580031 Karnataka.

November- 2024



CERTIFICATE

This is to certify that the project work entitled **“Billboard Management System”** submitted in partial fulfilment of the requirements for the award of degree of **Master of Computer Applications of KLE Technological University, Hubballi, Karnataka** is a result of the work carried out by **Billboard management System Team** during the academic year **2023-2024**.

Under the guidance of:

Dr. P.R. Patil

Head of Department

Professor, MCA Dept.

Chapter No.	Table of Contents	Page No.
1.	Introduction	1
	1.1 Literature review	.
	1.1.1 Existing System	
	1.1.2 Proposed System Overview	
	1.2 Challenges/Motivation	.
	1.3 Objectives of the project	
	1.4 Problem definition	
2.	Proposed System	.
	2.1 Description of proposed system with simple block diagram	
	2.2 Description of Target users	
	2.3 Advantages/applications of proposed system	
	2.4 Scope	
3.	Software Requirement Specification	
	3.1 Overview of SRS	
	3.2 Requirement Specifications	
	3.2.1 Functional Requirements	
	3.2.2 Use case diagrams	
	3.2.3 Use Case descriptions using scenarios, strictly as per Pressman Template	
	3.2.4 Nonfunctional Requirements	
	3.2.4.1 Performance requirements	
	3.2.4.2 Safety requirements	
	3.2.4.3 Security Requirements	
	3.2.4.4 Usability	

- 3.4 Software and Hardware requirement specifications
- 3.5 GUI of proposed system (snap shots) navigation from Home screen to end results
- 3.6 Acceptance test plan

4 System Design

- 4.1 Architecture of the system (explanation)
- 4.2 Level 0 DFD (Context Level)
- 4.3 Level 1 DFD
- 4.4 Decomposition DFD
- 4.5 Class Diagram
- 4.6 Sequence Diagram
- 4.7 Activity Diagram
- 4.8 ER diagram and SQL schema
- 4.9 Data structure used

5 Implementation

- 5.1 Proposed Methodology (Explain the methodology with diagram
Algorithms explanation)
- 5.2 Modules
(Briefly describe modules
Modules descriptions
Module name ()
Input :
Output :
Code or pseudocode)

6 Testing

6.1 Test plan and test cases (tested as per acceptance test plan)

7 Results & Discussions

(Discuss the results and its interpretation.

Snap shots with brief explanation about the results

8 Conclusion and future scope

- Conclude the project by mentioning the features or services you have
Implemented.
- In Future work explain how the use of your work can be enhanced

9 References/Bibliography

10 Appendix

- A Glossary
- B Description on Technology used
- C Explanation on Tools

1. Introduction

Billboards are a powerful and effective medium for outdoor advertising, widely utilized by companies to increase brand visibility and reach diverse audiences. Managing billboard advertising, however, is a complex task involving multiple stakeholders, from property owners and advertising agencies to client companies and operational staff. Traditional management methods often rely on manual processes, leading to inefficiencies, scheduling conflicts, and inconsistent tracking of billboard availability. As a result, advertisers face challenges in optimizing billboard utilization, ensuring timely updates, and providing transparent billing and reporting to clients.

The **Billboard Management System** is a comprehensive, web-based platform designed to modernize the management of billboard advertising by centralizing data and automating critical tasks. This system addresses the specific needs of property managers, agencies, and clients by providing a unified interface to manage billboard availability, track client campaigns, and generate real-time reports. The proposed solution will streamline operations, enhance communication between stakeholders, and support role-based access to ensure data security and efficient workflow management.

With modules dedicated to agency management, client coordination, property handling, and user access control, the Billboard Management System aims to reduce operational overhead, prevent scheduling conflicts, and offer a user-friendly experience for all involved parties. By leveraging modern web technologies and a scalable database structure, the system will provide a reliable, flexible solution for managing billboard advertising in today's fast-paced market.

1.1 Literature review

1.1.1 Existing System

Current billboard management systems often rely on manual processes or traditional database systems that lack integration and automation. Typically, agencies manage billboard advertisements through physical records, spreadsheets, or standalone software without seamless data sharing between departments. These limitations result in inefficiencies, such as delayed updates, inaccurate availability tracking, and difficulty in scheduling and reporting.

Most existing systems also have minimal support for real-time availability, centralized client management, or automated billing and reporting functionalities. Additionally, they often lack user access levels, making it challenging to control the access and edit rights of various stakeholders, such as staff, agencies, and clients. As a result, there is a need for a more streamlined, digital solution that centralizes all aspects of billboard management while ensuring data security and ease of access.

1.1.2 Proposed System Overview

The proposed Billboard Management System is a web-based platform designed to centralize and automate billboard management for agencies, clients, and other stakeholders. This system integrates modules for property management, agency and staff coordination, client company management, and user management, all overseen by a team leader for smooth inter-module collaboration.

Key features of the system include:

- **Real-Time Availability Tracking:** Displays current status and availability of billboards to prevent double bookings and improve resource utilization.
- **Centralized Client and Property Management:** Allows users to manage properties, agencies, and client relationships from a single interface, promoting seamless data flow and efficient scheduling.
- **Role-Based Access Control:** Restricts access based on user roles (e.g., agency staff, property owners, clients), ensuring data privacy and operational security.
- **Automated Billing and Reporting:** Facilitates financial operations with automated invoicing, billing, and customizable reporting features, providing agencies and clients with timely financial information.

The proposed system's architecture supports both frontend and backend development, with a focus on providing a user-friendly interface and efficient database handling. It also aims to integrate a comprehensive dashboard that gives the management team insights into occupancy rates, revenue projections, and campaign effectiveness. This system is designed to enhance operational efficiency, improve client satisfaction, and reduce manual workload, providing a modernized, scalable solution for billboard management.

1.2 Challenges/Motivation

Billboard advertising management involves complex coordination among multiple stakeholders—property owners, advertisers, and agencies—each requiring reliable, up-to-date information. Key challenges include:

- **Manual Processes:** Traditional management relies heavily on manual updates, spreadsheets, and standalone systems that lack integration, resulting in inefficiencies and errors.
- **Scheduling Conflicts:** Managing billboard availability without real-time tracking often leads to overbooking and double bookings, frustrating clients and reducing revenue.
- **Limited Transparency:** Lack of automated billing and reporting hampers visibility into finances, impacting trust and client satisfaction.
- **Data Security Concerns:** Sensitive information, including client details and financial records, is often inadequately protected, risking data breaches and unauthorized access.

These challenges motivate the need for a streamlined, automated system that enhances efficiency, accuracy, and security in billboard management.

1.3 Objectives of the Project

The Billboard Management System aims to address the above challenges by:

1. **Automating Core Processes:** Streamline property, client, and advertisement management, minimizing manual data entry and reducing errors.



2. **Implementing Real-Time Tracking:** Provide instant updates on billboard availability to prevent double-booking and optimize resource allocation.
3. **Enabling Role-Based Access:** Improve data security through role-specific access control, ensuring that only authorized users access certain information.
4. **Enhancing Financial Transparency:** Automate invoicing, billing, and reporting to deliver clear financial insights, improving client trust and operational efficiency.
5. **Centralizing Data Management:** Provide a unified platform for all stakeholders, allowing seamless data flow and easier management of multiple billboards and contracts.

1.4 Problem Definition

The primary problem is the lack of an integrated, digital system to manage billboard advertisements effectively. The project seeks to design and implement a comprehensive, web-based Billboard Management System capable of:

- Managing entities such as property owners, billboard companies, clients, advertisements, and branches.
- Handling critical operations like scheduling, payments, contract management, and real-time availability tracking.
- Enhancing data security, access control, and usability for a diverse group of stakeholders.

By achieving these objectives, the system will simplify billboard management, improve client satisfaction, and provide a modern solution for the dynamic needs of the advertising industry.

2. Proposed System

The proposed Billboard Management System is a web-based application designed to streamline billboard management through automation, centralization, and enhanced security. This system consolidates property, client, advertisement, and financial management within a single interface, allowing seamless coordination and real-time updates.

2.1 Description of Proposed System with Simple Block Diagram

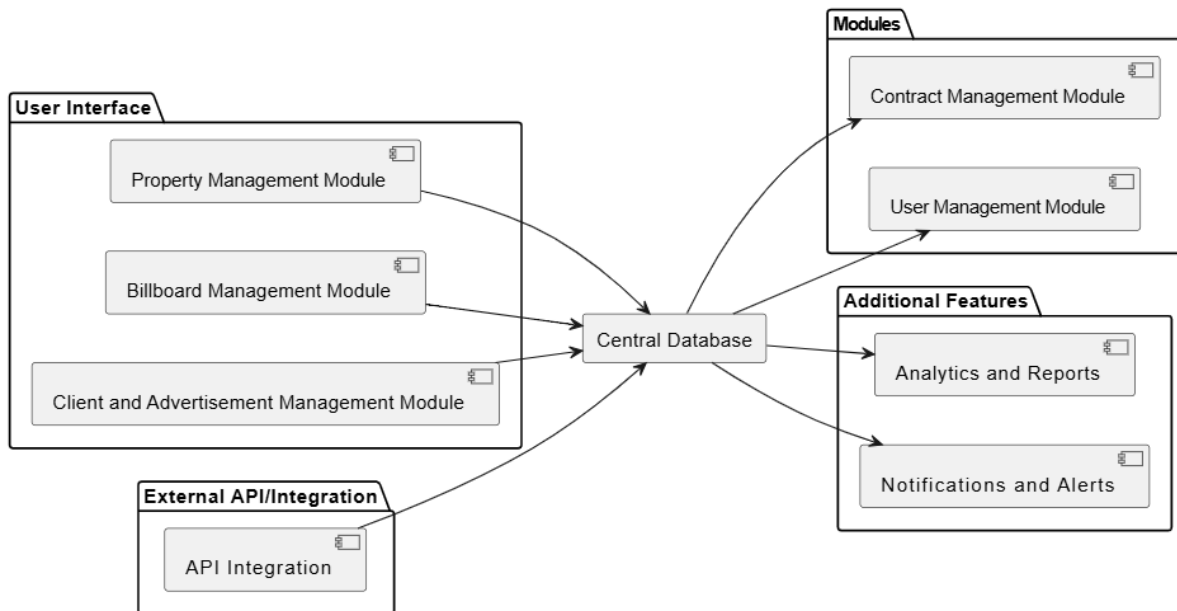
System Description:

The Billboard Management System is built as a MERN (MongoDB, Express.js, React, Node.js) stack application, leveraging modern web technologies for scalability, performance, and ease of use. The system comprises multiple modules designed to manage essential aspects of billboard operations:

- **Property Management Module:** Manages properties owned by property owners, tracking details like location and ownership.
- **Billboard Management Module:** Records each billboard's details, including location, dimensions, and availability status.
- **Client and Advertisement Management Module:** Manages client details and advertisements, including scheduling, types, and content.
- **Contract Management Module:** Tracks contracts, including start and end dates, financial terms, and billing schedules.

- **Finance Module:** Manages invoices, payments, and transaction history.
- **User Management Module:** Allows role-based access to ensure data security and restrict access based on roles.

Block Diagram:



2.2 Description of Target Users

The Billboard Management System is designed to serve various stakeholders in billboard management, including:

- **Property Owners:** Individuals or entities owning billboards who need to monitor and manage their assets.
- **Agencies:** Advertising agencies responsible for placing advertisements on billboards, scheduling, and managing client relationships.
- **Clients:** Businesses or brands advertising on billboards, needing access to scheduling, billing, and performance reports.
- **Staff and Administrators:** Personnel who require access to manage daily operations, such as handling finances, contracts, and user roles.
- **System Administrators:** Individuals overseeing system operations, managing access control, and ensuring data security.

Each user type has distinct access privileges tailored to their roles to facilitate smooth, secure operations.

2.3 Advantages/Applications of the Proposed System

Advantages:

- **Operational Efficiency:** Automation of routine tasks reduces manual workload, increases data accuracy, and allows staff to focus on higher-value activities.
- **Real-Time Data Access:** Real-time billboard availability and booking updates minimize scheduling conflicts and improve client satisfaction.

- **Centralized Management:** Combines property, client, advertisement, and financial data in one system, allowing easier data tracking and retrieval.
- **Role-Based Access Control:** Provides security by restricting access to sensitive data based on user roles, protecting clients and agencies.
- **Automated Billing and Reporting:** Simplifies financial management with automated invoicing and customizable reporting, ensuring transparency and accuracy.

Applications:

- **Billboard and Advertising Management:** The system supports advertising agencies and billboard companies in handling campaigns, contracts, and resource allocations.
- **Client Relationship Management (CRM):** Allows agencies to manage client details, contracts, and billing more effectively, fostering better client relations.
- **Real Estate Management:** Property owners can use the system to manage and track billboard assets and revenue.

2.4 Scope

The Billboard Management System is a comprehensive solution, but its scope is defined as follows:

- **In Scope:**
 - Management of billboards, clients, advertisements, and contracts.
 - Real-time availability tracking and scheduling.
 - Automated billing, invoicing, and financial reporting.
 - Role-based access control for enhanced security.
 - Scalable architecture supporting future expansion of modules or functionalities.
- **Out of Scope:**
 - Detailed customer relationship analytics (beyond basic CRM features).
 - Direct management of physical billboard maintenance or inspections.
 - Integration with non-web-based platforms or standalone offline systems.
 - Extensive client marketing or campaign analysis tools outside of the billboard context.

This system will cater to billboard management's core operational needs, providing a scalable framework to support a growing number of users, billboards, and advertisements.

3. Software Requirement Specification (SRS)

The Software Requirement Specification (SRS) defines the detailed requirements for the Billboard Management System, outlining both functional and non-functional needs to ensure the system meets its intended purpose.

3.1 Overview of SRS

The Billboard Management System aims to provide a centralized, web-based platform for managing billboard properties, advertisements, clients, and finances. This document specifies the software and hardware requirements, functional and non-functional specifications, use case scenarios, and performance and security needs. By defining these elements, the SRS serves as a blueprint for development, guiding design, testing, and deployment.

3.2 Requirement Specifications

This section contains detailed specifications for each system requirement, including functional and non-functional needs to ensure the system meets both user and performance expectations.

3.2.1 Functional Requirements

The functional requirements describe the primary actions the system should support:

1. **Property Management:** Allow users to add and manage billboard properties, including details like location, dimensions, and status.
2. **Billboard Management:** Enable tracking of billboard availability, booking status, and associated visibility data.
3. **Client Management:** Manage client profiles, including contract history, contact details, and industry type.
4. **Advertisement Management:** Support creating and managing advertisements, covering languages, ad types, and scheduling.
5. **Contract Management:** Track contract details, including start/end dates, amounts, and terms of payment.
6. **Staff Management:** Record and manage staff details, including assigned branches, contact details, and bank information.
7. **Financial Management:** Enable tracking of payments, invoicing, account balances, and transactions.
8. **User Management and Access Control:** Enforce role-based access, restricting certain functionalities based on user role.

3.2.2 Use Case Diagrams

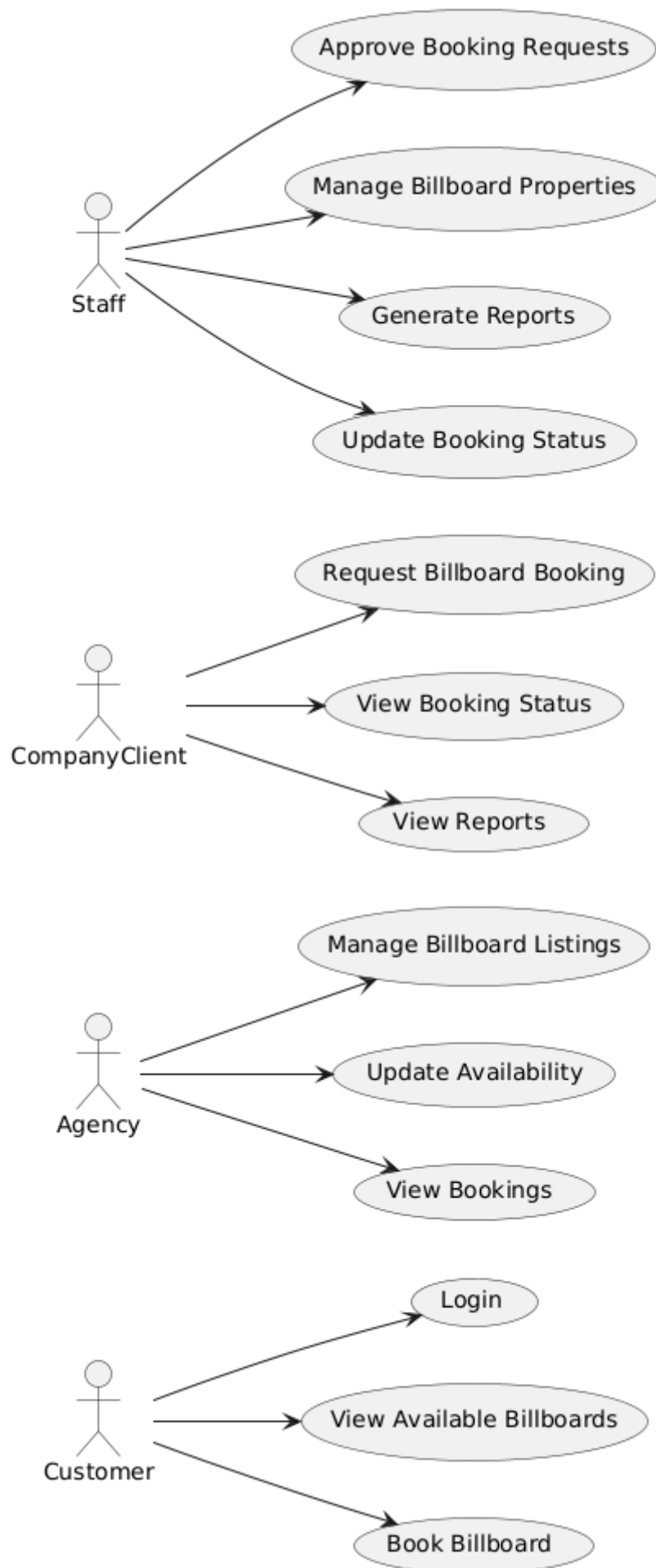
The use case diagram provides a visual representation of the system's interactions with different user roles, showcasing the relationships between the system and its users (e.g., property owner, agency staff, client, and admin). The diagram highlights key functionalities, including property management, client management, and financial transactions.

3.2.3 Use Case Descriptions Using Scenarios

For each use case, we use the Pressman template to define actors, scenarios, preconditions, and flow of events.

Example Use Case Description (Property Management Module)

- **Use Case Name:** Add Billboard Property
- **Actors:** Staff, Company/client, Agency, Customer
- **Goal:** To add a new billboard property to the system
- **Preconditions:** The user must have admin access.
- **Scenario:**
 1. User logs into the system and navigates to the Property Management module.
 2. User selects “Add New Property” and enters property details (location, dimensions, owner).
 3. User confirms the details and saves the property.
 4. System confirms the addition and updates the database.



3.2.4 Non-functional Requirements

3.2.4.1 Performance Requirements

- **Response Time:** The system should load dashboard data within 2 seconds for regular usage and respond to data queries within 5 seconds.
- **Scalability:** The system should handle up to 1000 active users and 5000 records in the database without performance degradation.

3.2.4.2 Safety Requirements

- **Data Backup:** Regular database backups should be performed daily to prevent data loss in case of system failure.
- **Redundancy:** Cloud backup storage to be enabled for key data.

3.2.4.3 Security Requirements

- **Authentication:** Implement multi-factor authentication for admin users.
- **Authorization:** Role-based access control to restrict user permissions.
- **Data Encryption:** Sensitive data, including financial records and client information, should be encrypted in the database.

3.2.4.4 Usability

- **User Interface:** The system interface should be user-friendly, with intuitive navigation and clear labeling.
- **Responsiveness:** Compatible with desktop, tablet, and mobile devices to ensure ease of access.

3.4 Software and Hardware Requirement Specifications

Software Requirements:

- **Frontend:** React.js, HTML5, CSS3, JavaScript, Bootstrap
- **Backend:** Node.js, Express.js
- **Database:** MongoDB (local instance at mongodb://localhost:27017/billboard for testing)
- **Development Tools:** VSCode, Git, GitHub, Thunder Client
- **Others:** NPM/Yarn, Chrome/Firefox/Edge browser

Hardware Requirements:

- **Processor:** Intel Core i5 or equivalent
- **RAM:** Minimum 8 GB (16 GB recommended)
- **Storage:** 256 GB SSD
- **Internet:** High-speed connection

3.5 GUI of Proposed System

3.6 Acceptance Test Plan

The acceptance test plan defines criteria for testing the system against functional and non-functional requirements. Key test cases will ensure that the system:

1. **Meets all functional requirements** (property management, client handling, advertisement scheduling).
2. **Passes security checks** for data encryption and access control.
3. **Performs within acceptable response times** for both general and peak usage.
4. **Complies with usability standards**, providing a responsive, accessible user interface.

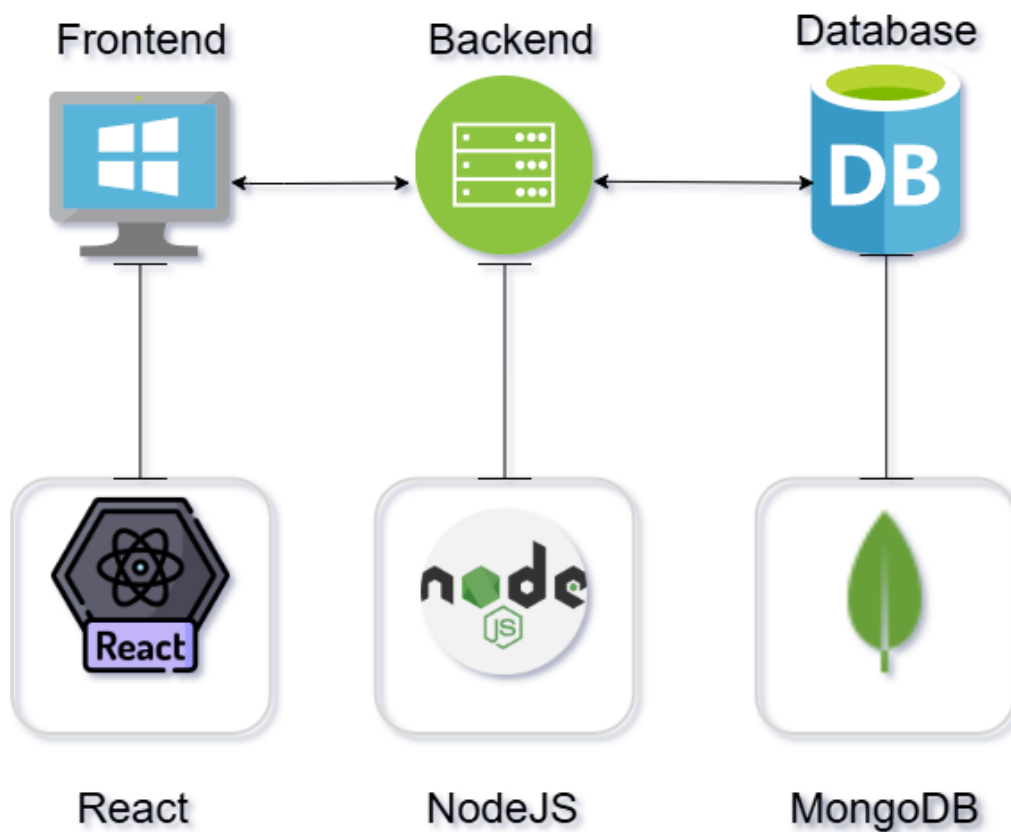
Testing phases will include unit, integration, and user acceptance tests to validate system functionality and reliability.

This SRS provides a clear framework for developing, testing, and implementing a reliable, secure, and user-friendly Billboard Management System.

4.SYSTEM DESIGN

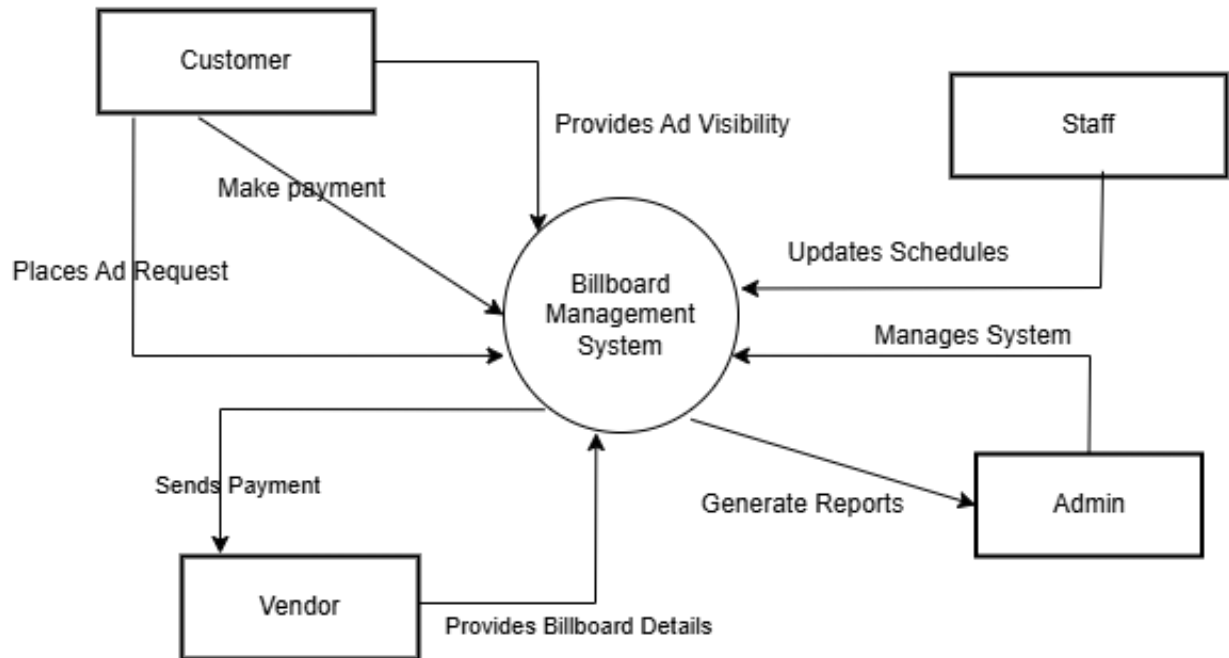
4.1 Architectural Design

1. **Property Module:** Manages properties and their owners.
2. **Billboard Module:** Tracks billboards, including details like location, dimensions, and status.
3. **Client and Advertisement Module:** Manages clients and the advertisements they own.
4. **Contract Module:** Tracks details of advertisement contracts.
5. **Finance Module:** Manages payment details, invoices, and transaction records.
6. **Staff and Branch Module:** Manages details related to staff members and branches.

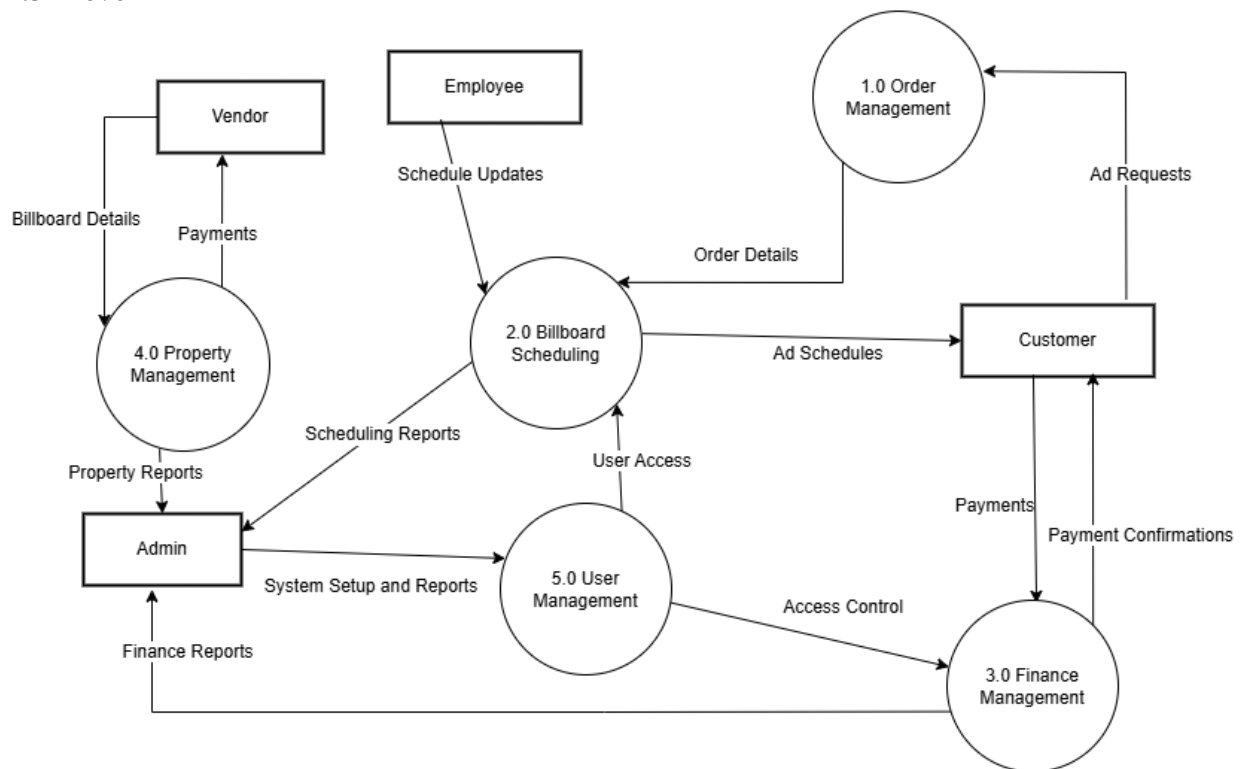


Architectural Design

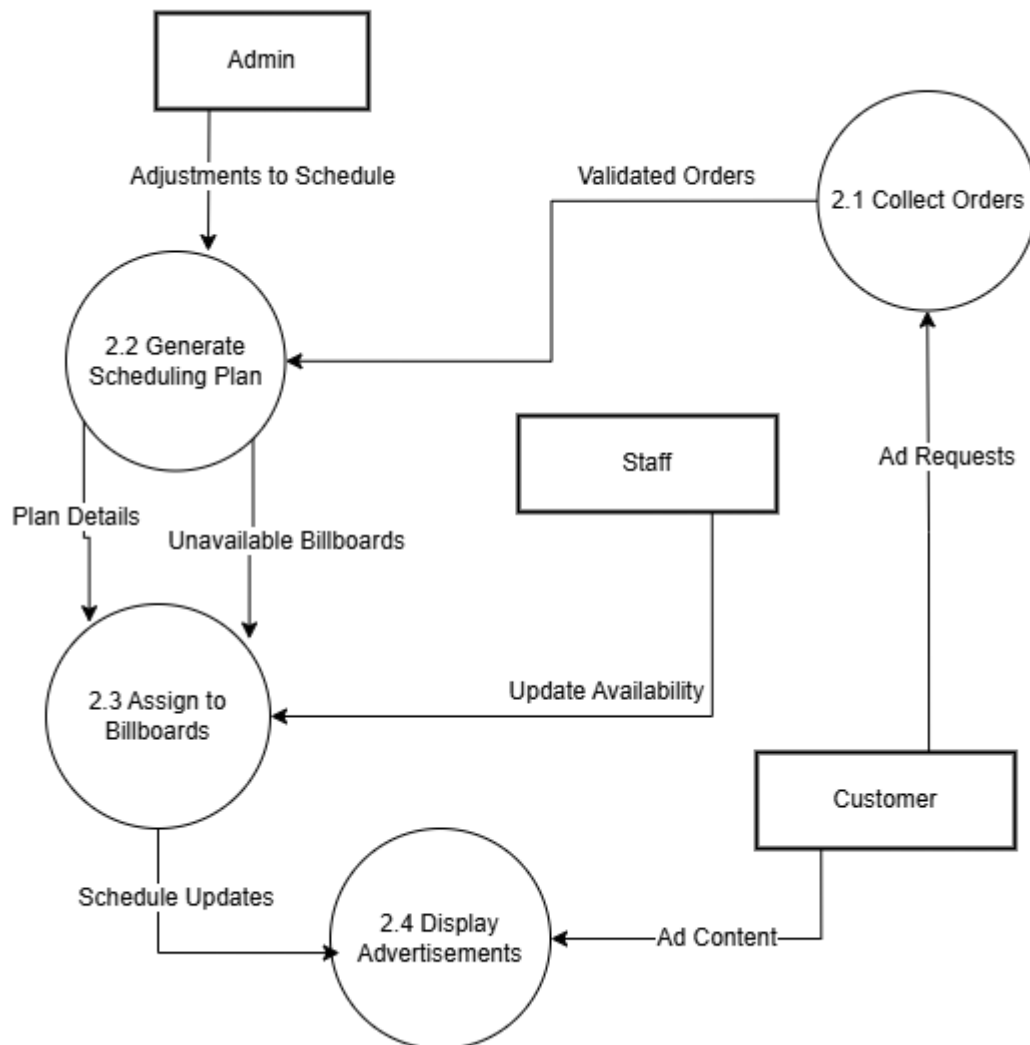
4.2 Level 0 (Context – Level) DFD



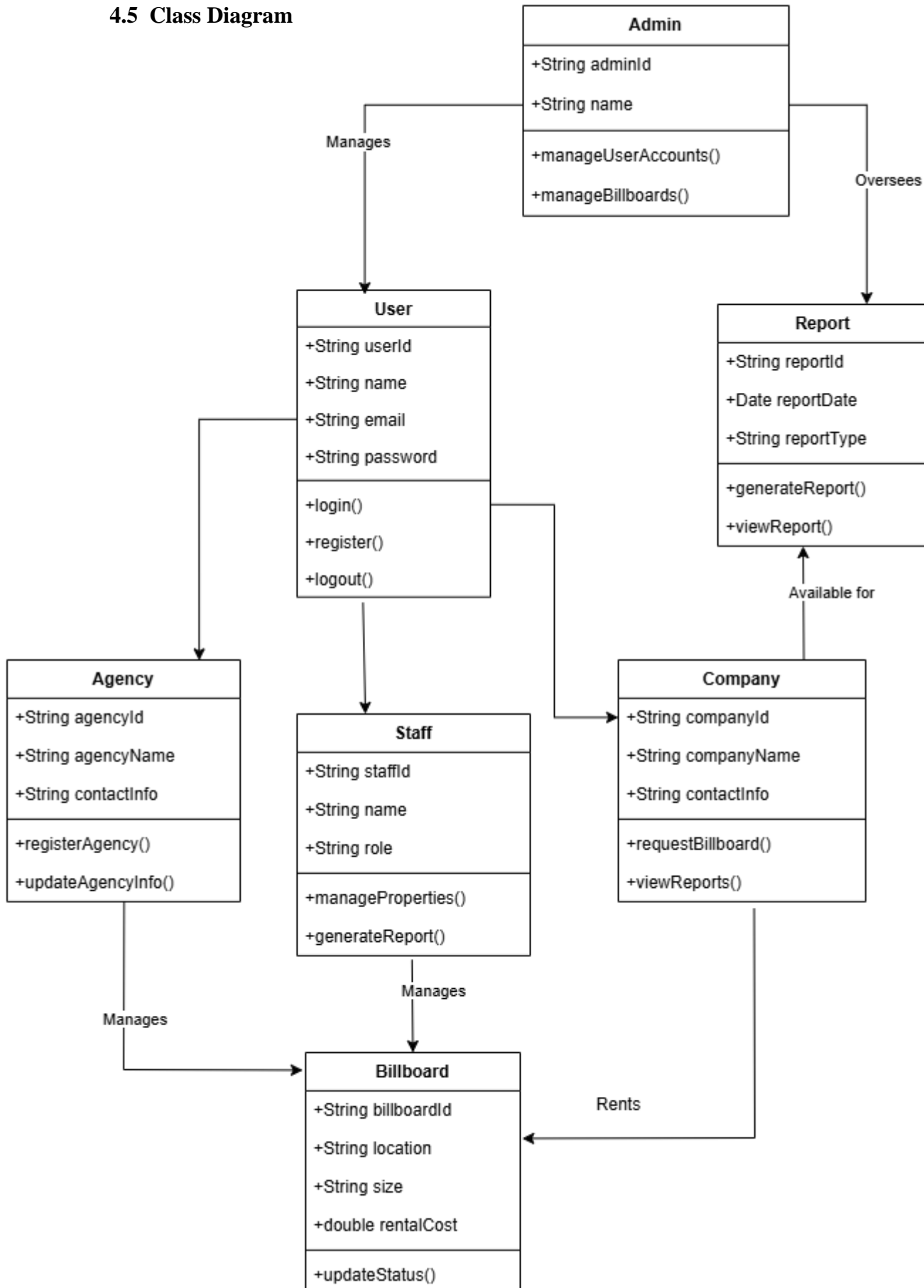
4.3 Level 1 DFD



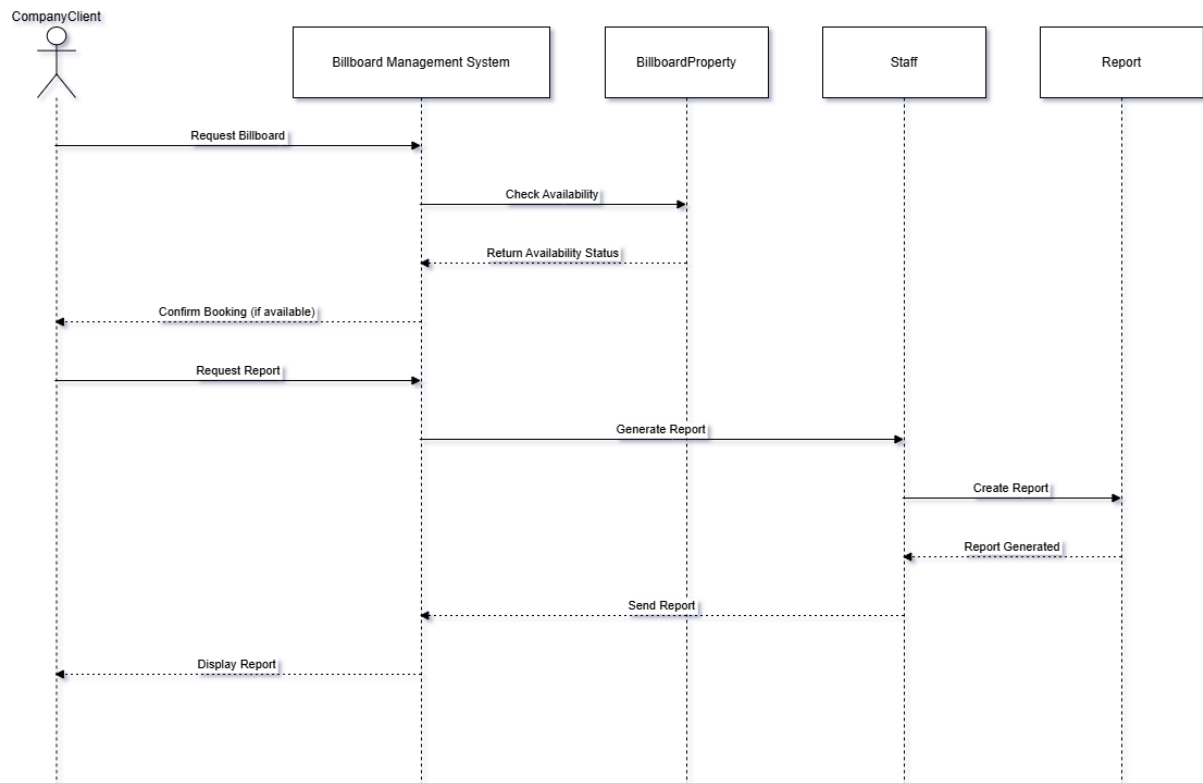
4.4 Decomposition DFD



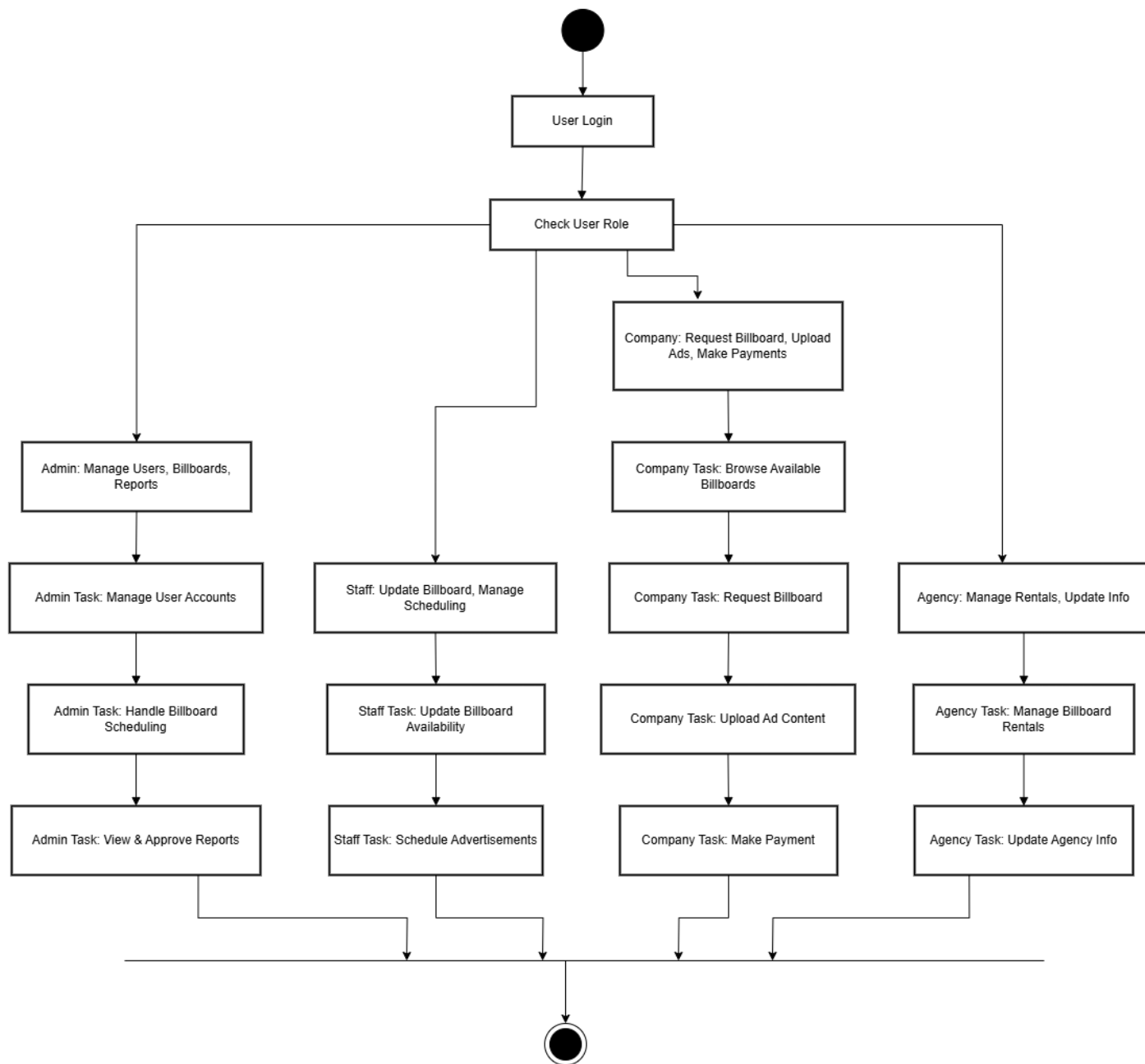
4.5 Class Diagram



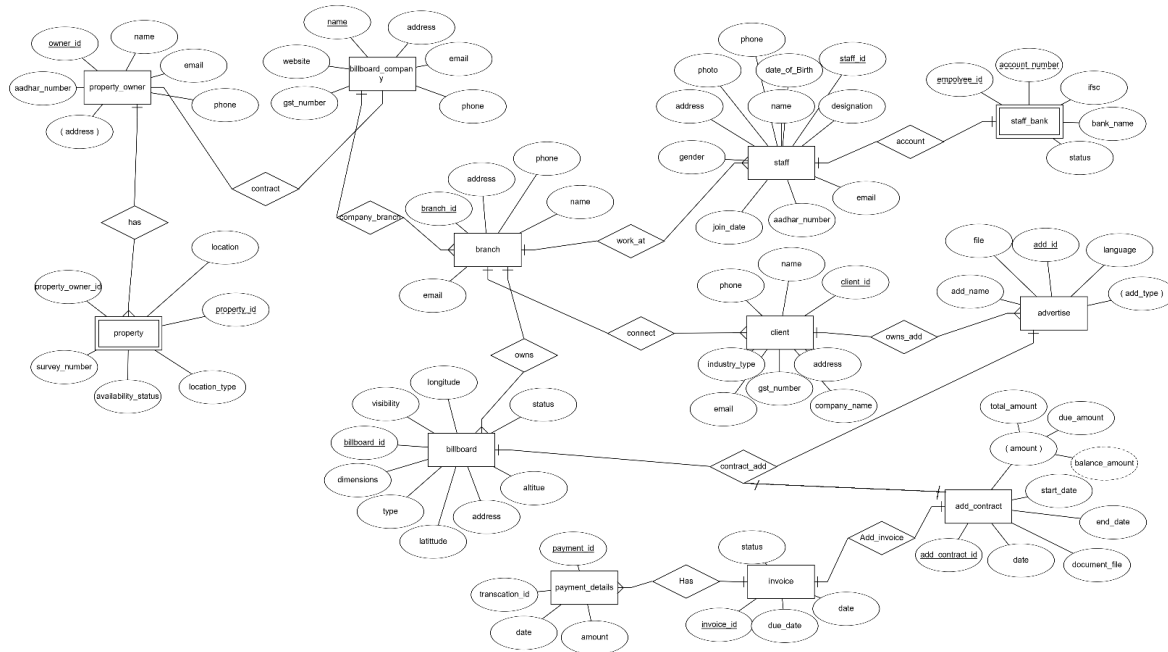
4.6 Sequence diagram



4.7 Activity Diagram



4.8 ER diagram and SQL schema



SQL Structure (Schema)

-- Table: Property Owner

```
CREATE TABLE PropertyOwner (
    owner_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    aadhar_number VARCHAR(12),
    email VARCHAR(100),
    phone VARCHAR(15),
    address TEXT
);
```

-- Table: Billboard Company

```
CREATE TABLE BillboardCompany (
    company_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    gst_number VARCHAR(15),
    email VARCHAR(100),
    phone VARCHAR(15),
    address TEXT,
```

```
website VARCHAR(100)

);

-- Table: Property
CREATE TABLE Property (
    property_id INT AUTO_INCREMENT PRIMARY KEY,
    property_owner_id INT,
    survey_number VARCHAR(50),
    location_type VARCHAR(50),
    availability_status BOOLEAN,
    longitude DECIMAL(10, 8),
    latitude DECIMAL(10, 8),
    address TEXT,
    FOREIGN KEY (property_owner_id) REFERENCES PropertyOwner(owner_id)
);

-- Table: Billboard
CREATE TABLE Billboard (
    billboard_id INT AUTO_INCREMENT PRIMARY KEY,
    property_id INT,
    type VARCHAR(50),
    dimensions VARCHAR(50),
    visibility VARCHAR(50),
    altitude DECIMAL(6, 2),
    status VARCHAR(50),
    FOREIGN KEY (property_id) REFERENCES Property(property_id)
);

-- Table: Staff
CREATE TABLE Staff (
    staff_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
```




```
aadhar_number VARCHAR(12),  
gender VARCHAR(10),  
date_of_birth DATE,  
designation VARCHAR(50),  
join_date DATE,  
email VARCHAR(100),  
phone VARCHAR(15),  
address TEXT  
);
```

-- Table: Client

```
CREATE TABLE Client (  
    client_id INT AUTO_INCREMENT PRIMARY KEY,  
    company_name VARCHAR(100),  
    industry_type VARCHAR(50),  
    gst_number VARCHAR(15),  
    email VARCHAR(100),  
    phone VARCHAR(15),  
    address TEXT  
);
```

-- Table: Advertisements

```
CREATE TABLE Advertisement (  
    add_id INT AUTO_INCREMENT PRIMARY KEY,  
    client_id INT,  
    add_name VARCHAR(100),  
    add_type VARCHAR(50),  
    language VARCHAR(50),  
    file TEXT,  
    FOREIGN KEY (client_id) REFERENCES Client(client_id)  
);
```

-- Table: Advertisement Contracts

```
CREATE TABLE AdvertisementContract (  
    add_contract_id INT AUTO_INCREMENT PRIMARY KEY,  
    add_id INT,  
    start_date DATE,  
    end_date DATE,  
    document_file TEXT,  
    FOREIGN KEY (add_id) REFERENCES Advertisement(add_id)  
);
```

-- Table: Payments

```
CREATE TABLE Payments (  
    payment_id INT AUTO_INCREMENT PRIMARY KEY,  
    transaction_id VARCHAR(50),  
    amount DECIMAL(10, 2),  
    date DATE  
);
```

-- Table: Invoice

```
CREATE TABLE Invoice (  
    invoice_id INT AUTO_INCREMENT PRIMARY KEY,  
    add_contract_id INT,  
    total_amount DECIMAL(10, 2),  
    due_amount DECIMAL(10, 2),  
    balance_amount DECIMAL(10, 2),  
    due_date DATE,  
    date DATE,  
    FOREIGN KEY (add_contract_id) REFERENCES AdvertisementContract(add_contract_id)  
);
```

Schema:

1. Advertisement



add_id	client_id	add_name	add_type	language	file
--------	-----------	----------	----------	----------	------

2. AdvertisementContract

add_contract_id	add_id	start_date	end_date	document_file
-----------------	--------	------------	----------	---------------

3. Billboard

billboard_id	property_id	typee	dimensionsns	visibility	altitude	status
--------------	-------------	-------	--------------	------------	----------	--------

4. BillboardCompany

company_id	name	gst_number	email	phone	address	website
------------	------	------------	-------	-------	---------	---------

5. Client

client_id	company_name	industry_type	gst_number	email	phone	address
-----------	--------------	---------------	------------	-------	-------	---------

6. Property

property_id	property_owner_id	survey_number	location_type	availability_status	longitud
-------------	-------------------	---------------	---------------	---------------------	----------

7. PropertyOwner

owner_id	name	aadhar_number	email	phone	address
----------	------	---------------	-------	-------	---------

8. Staff

Staff_id	Name	Adhaar_no	gender	dateofbirth	designation	joinDate	email	phNo	address
----------	------	-----------	--------	-------------	-------------	----------	-------	------	---------

9. Invoice

invoice_id	ad_contract_id	total_amt	due_amt	balance_amt	due_date	date
------------	----------------	-----------	---------	-------------	----------	------

10. Payments

payment_id	transaction_id	amount	date
------------	----------------	--------	------

4.9 Data structure used

The Billboard Management System uses MongoDB for a NoSQL, document-based data structure:

- **Document Model:** Each record is stored as a document (JSON-like structure) that can contain nested fields, allowing for a flexible schema.
- **Collections:** Collections represent tables for entities like Properties, Clients, Advertisements, and Transactions.
- **Indexes:** To improve query performance, indexes are used on frequently accessed fields, such as clientID and propertyID.

MongoDB's flexibility suits the dynamic requirements of billboard management, enabling easy updates to schema without complex migrations.

5. Implementation

This section outlines the implementation approach, including methodology, modules, and code structure.

5.1 Proposed Methodology

The **Proposed Methodology** follows an **Agile** development process, utilizing iterative cycles to gradually develop, test, and refine the system.

1. **Requirements Gathering:** Define requirements with user stories and functionality prioritization.
2. **Design Phase:** Create wireframes, data models, and system diagrams for planned modules.
3. **Development Phase:** Follow Agile sprints for coding each module, with continuous integration.
4. **Testing Phase:** Conduct unit tests after each sprint and a final system-wide test for overall functionality.
5. **Deployment:** Release the system in stages, initially deploying to a test server before moving to production.

5.2 Modules

The Billboard Management System includes several core modules. Each module has specific inputs, processes, and outputs.

Module Descriptions:

1. Property Management Module

- **Input:** Property details (ID, location, owner information)
- **Output:** Stored property details, retrieval for display on the dashboard.
- **Code/Pseudocode:**

python

Copy code

```
function addProperty(propertyID, location, ownerID):
    # Create new property record
```



```
property = {  
    "propertyID": propertyID,  
    "location": location,  
    "ownerID": ownerID  
}  
db.properties.insert(property)  
return "Property added successfully"
```

2. Billboard Management Module

- **Input:** Billboard details (ID, property ID, availability, dimensions)
- **Output:** Updated billboard status and information
- **Pseudocode:**

python

Copy code

```
function updateBillboard(billboardID, availability):  
    # Locate and update billboard status  
    db.billboards.update({"billboardID": billboardID}, {"$set": {"availability": availability}})  
    return "Billboard updated"
```

3. Client Management Module

- **Input:** Client information (ID, name, contact, industry type)
- **Output:** Client information stored and retrieved for contract management.
- **Pseudocode:**

python

Copy code

```
function addClient(clientID, name, contactInfo):  
    client = {  
        "clientID": clientID,  
        "name": name,  
        "contactInfo": contactInfo  
    }  
    db.clients.insert(client)  
    return "Client added successfully"
```

4. Financial Management Module

- **Input:** Transaction details (invoice ID, amount, client ID)
- **Output:** Financial records updated and invoices generated.
- **Pseudocode:**

python

Copy code

```
function generateInvoice(clientID, amount):  
    invoice = {  
        "clientID": clientID,  
        "amount": amount,  
        "status": "Pending"  
    }  
    db.invoices.insert(invoice)  
    return "Invoice generated"
```

6. Testing

6.1 Test Plan and Test Cases

The testing plan ensures each module is tested for functional accuracy, reliability, and security. Acceptance tests check that requirements match real-world scenarios.

Example Test Cases:

1. **Property Management Test:**
 - **Description:** Add new property and verify it in the database.
 - **Expected Outcome:** Property details should be accurately stored.
2. **Billboard Availability Test:**
 - **Description:** Update billboard status to "Occupied."
 - **Expected Outcome:** Billboard status changes in the database.
3. **Client Information Retrieval Test:**
 - **Description:** Retrieve a client's information by ID.
 - **Expected Outcome:** Accurate client data is returned.

7. Results & Discussions

This section provides a summary of results with screenshots from key screens of the system.

Discussion: The system successfully automates billboard management tasks, allowing users to:

- Track billboard availability in real time.
- Store, update, and retrieve client and property details.
- Generate invoices and track payments efficiently.

Snapshots: Screenshots from the interface include the **Dashboard**, **Client Management Screen**, **Billboard Details Page**, and **Invoice Generation** page, each with brief descriptions.

8. Conclusion and Future Scope

Conclusion

The Billboard Management System centralizes and automates billboard management, offering features like real-time availability tracking, role-based access, and automated billing. The system reduces manual workload, enhances transparency, and improves operational efficiency for billboard advertising management.

Future Work

Future enhancements could include:

- **Machine Learning:** Integrate predictive analytics to optimize billboard placements and maximize ad reach.
- **Mobile Application:** Develop a mobile app version for remote access and management.
- **Integration with External APIs:** Link with external ad-tracking systems to provide real-time data on advertisement performance and engagement.
- **Scalable Cloud Infrastructure:** Scale the system with cloud computing solutions to manage larger datasets and complex operations more efficiently.

These enhancements would make the Billboard Management System more robust, providing a comprehensive solution for modern billboard advertising management.

IMPLEMENTATION

6.1 Introduction to New Technologies used in this project

6.2 Source code of challenging modules in this project