

Mall Management System

A MINI-PROJECT REPORT

Submitted by

KENNETH PRATHAP K 241901048

JAIYANTH T 241901037

in partial fulfilment of the award of the degree

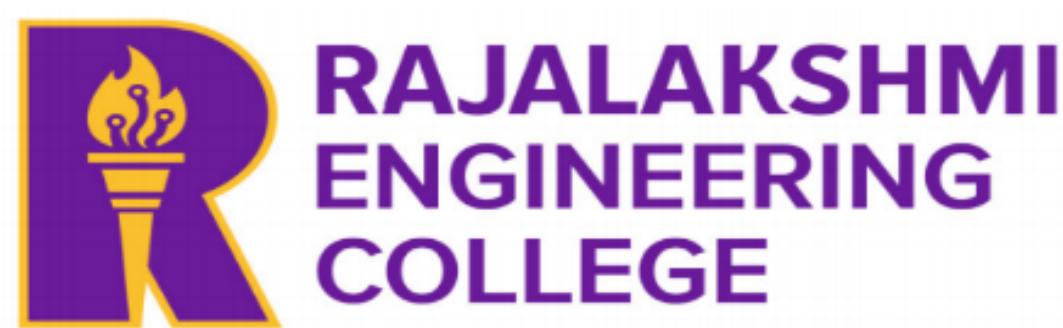
of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(CYBER SECURITY)



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI-602105

An Autonomous Institute

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project "**MALL MANGEMENT SYSTEM**" is the bonafide work of "**KENNETH PRATHAP K (241901048), JAIYANTH T (24191037)**" who carried out the project work under my supervision.

SIGNATURE

Ms. R. Divya

ASSISTANT PROFESSOR

Department of Computer Science

and Engineering (Cyber Security)

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voice examination to be held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the mini project report **Mall Management System**, submitted as part of the curriculum

requirements for the Bachelor of Engineering (B.E) degree affiliated to Anna University, is a bonafide work

carried out by us under the supervision of Ms. R. Divya, Assistant Professor, Department of Computer

Science Engineering and Cyber Security, Rajalakshmi Engineering College, Chennai.

This submission represents our ideas in our own words, and where ideas or words of others have been included,

we have adequately and accurately cited and referenced them original sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data, idea, fact, or source in our submission. We understand that any violation

of the above will be grounds for disciplinary action by the institute and/or the University and may also evoke

penal action from the sources which have not been properly cited or from whom proper permission has not

been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar

title of any other University.

Rajalakshmi Engineering College,

KENNETH PRATHAP K

Chennai

JAIYANTH T

November 2025

ABSTRACT

The Mall Management System is a desktop-based Java application that automates core mall operations including shop allocation, tenant management, lease tracking, billing, maintenance requests, event coordination, and reporting. The system is developed using **JavaFX** with a **GUI**-based interface and a **MySQL** backend. All data operations are performed through **SQL** queries using **JDBC**, ensuring secure and reliable communication between the application and the database.

The system follows a modular architecture and supports **CRUD** operations across all functional modules. A centralized **DBMS** structure has been designed using an **ER** model with normalized tables, ensuring data integrity, reduced redundancy, and optimized query performance. Role-Based

Access Control (**RBAC**) is implemented to provide separate dashboards for Admin, Staff, and Tenants, enabling secure and permission-based access.

Key features include automated billing, tenant onboarding, lease lifecycle tracking, maintenance workflow management, event scheduling, and real-time analytics/report generation. The system improves operational efficiency, reduces manual workload, enhances data visibility, and enables data-driven decision-making for all administrators.

Keywords: Mall Management System, JavaFX, MySQL, JDBC, DBMS, SQL, CRUD, ER Model, RBAC, GUI, Desktop Application

ACKNOWLEDGEMENT

We like to convey our sincere appreciation to all who have supported and mentored us during the successful completion of this project work.

We express our profound gratitude to **Mr. Benedict J.N.**, Associate Professor (SG) and Head of the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for furnishing us with essential resources, support, and an enabling environment to execute this project.

We express our profound gratitude to Ms. **R. Divya**, Assistant Professor in the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for her unwavering support, invaluable insights, and collaboration throughout our endeavour.

We would like to convey our gratitude to our faculty members and colleagues for their valuable feedback and encouragement.

We express our gratitude to our families and friends for their steadfast support, patience, and encouragement, which were instrumental in the effective execution of this seminar.

KENNETH PRATHAP K

JAIYANTH T

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	i
	ACKNOWLEDGEMENT	ii
	LIST OF ABBREVIATION	iv
	LIST OF FIGURES	v
	LIST OF TABLES	vi
1	INTRODUCTION	1
1.1	Scope of the Work	2
1.2	Problem Statement	3
2	Literature Review	4
3	System Design	7
3.1	ER Diagram	7
3.2	Database Schema Design	7
4	CODING	11
4.1	Login Module	11
4.2	Admin Dashboard	13
5	SCREENSHOTS	16
6	Functional Overview	20
6.1	Functional Modules and Core Features	20
7	Conclusion	23
	References	24

LIST OF ABBREVIATION

Abbreviation	Full Term
DBMS	Database Management System
SQL	Structured Query Language
JDBC	Java Database Connectivity
GUI	Graphical User Interface Graphical User Interface
ER	Entity-Relationship
CRUD	Create, Read, Update, Delete
API	Application Programming Interface
RBAC	Role-Based Access Control
JVM	Java Virtual Machine

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
2.1	Flowchart Depicting the Development Process of Mall Management System	4
3.1	Entity Relationship (ER) Diagram of Mall Management System	7
5.1	Login Interface Screenshot	16
5.2	Admin Dashboard Screenshot	17
5.3	Adding Shop Details Screenshot	18
5.4	Reports & Analytics Screenshot	19

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
3.1	Schema of the shops Table	9
3.2	Schema of the tenants Table	9
3.3	Schema of the leases Table	9
3.4	Schema of the bills Table	9
3.5	Schema of the maintenance_requests Table	10
3.6	Schema of the events Table	10
3.7	Schema of the users Table	10

CHAPTER 1

INTRODUCTION

The **Mall Management System** is a modern, integrated solution designed to simplify and automate the complex operations involved in managing shopping malls. Traditional mall administration heavily relies on manual record-keeping, spreadsheets, and inconsistent communication between tenants, staff, and management. These manual processes are time-consuming, error-prone, and inefficient, leading to issues such as data redundancy, poor coordination, and delayed decision-making.

To overcome these challenges, the proposed system introduces a **digital, database-driven management solution** that ensures secure, real-time, and reliable handling of mall operations. Developed using **Java Swing for the frontend** and **MySQL for the backend**, the system employs structured database principles, data normalization, and modular architecture to support scalable, multi-user environments. It is a comprehensive application that manages all aspects of mall

administration – from tenant and lease management to billing, maintenance, and event coordination.

The **Mall Management System** provides separate dashboards for different roles – **Admin, Staff, and Tenants** – each tailored to their responsibilities. The Admin dashboard enables complete control over mall operations, allowing the addition of shops, monitoring of leases, and analysis of revenue reports. The Staff interface facilitates daily operational tasks such as handling bills, maintenance requests, and events. The Tenant view, meanwhile, provides an intuitive interface to view leases, payments, and mall announcements, enhancing communication and transparency.

The system architecture follows the principles of **object-oriented programming (OOP)** and **software engineering modularity**, ensuring flexibility, maintainability, and ease of future enhancements. Database normalization minimizes redundancy and enforces referential integrity across related entities, including shops, tenants, leases, bills, maintenance requests, and events.

By digitizing mall operations, this project aims to improve **efficiency, accuracy, and data consistency**. The solution also supports **real-time reporting and analytics**, empowering decision-makers with insights into occupancy, revenue generation, and tenant performance. Moreover, by introducing automation into routine administrative tasks, the system allows mall management teams to focus more on strategic growth and customer satisfaction.

Ultimately, the **Mall Management System** represents a step toward smarter infrastructure management. It bridges the gap between traditional mall operations and modern digital solutions, providing a secure, scalable, and user-friendly platform that streamlines business processes and enhances overall operational excellence.

1.1 Scope of the Work

The scope of this project is to design and implement a complete, database-driven Mall Management System that automates operational workflows for mall administrators, staff, and tenants. The system focuses on improving efficiency, accuracy, and transparency in mall operations while reducing reliance on manual paperwork.

Key features included within the scope are:

- **Centralized management of shops, tenants, leases, bills, maintenance requests, and events**
- **Role-based access control (RBAC)** for Admin, Staff, and Tenant users
- **Secure authentication and credential storage** using database-level validation
- **Graphical User Interface (GUI)** for a modern and intuitive user experience
- **Automated billing and payment tracking** with database persistence
- **Maintenance request logging and workflow tracking**
- **Reports and analytics** for real-time operational insights
- **Database integration using MySQL & JDBC**, ensuring reliable CRUD operations
- **Support for scalability and future enhancements** such as cloud deployment, IoT integration, and mobile support

Mall operations traditionally rely on manual record-keeping, disconnected spreadsheets, and isolated software tools to manage shops, tenants, leases, billing, maintenance, and events. This fragmented approach results in frequent errors, delayed communication, data redundancy, and a lack of centralized control. Administrators struggle to track tenant information, staff have limited visibility over operational processes, and tenants do not have a seamless way to access their billing, lease, or event-related data.

The core problem addressed in this project is the **absence of an integrated, automated, and secure mall management platform** that allows all stakeholders— administrators, staff, and tenants— to efficiently manage their roles through a unified system. There is a need for a platform that provides:

- Centralized database management
- Role-based access for different users
- Automated billing and maintenance workflows
- Real-time data visibility and analytics
- Reduced dependency on manual processes

This project aims to replace outdated and inefficient workflows with a **modern, database-driven software system** that ensures accuracy, transparency, and operational efficiency in mall administration.

CHAPTER 2

LITERATURE REVIEW

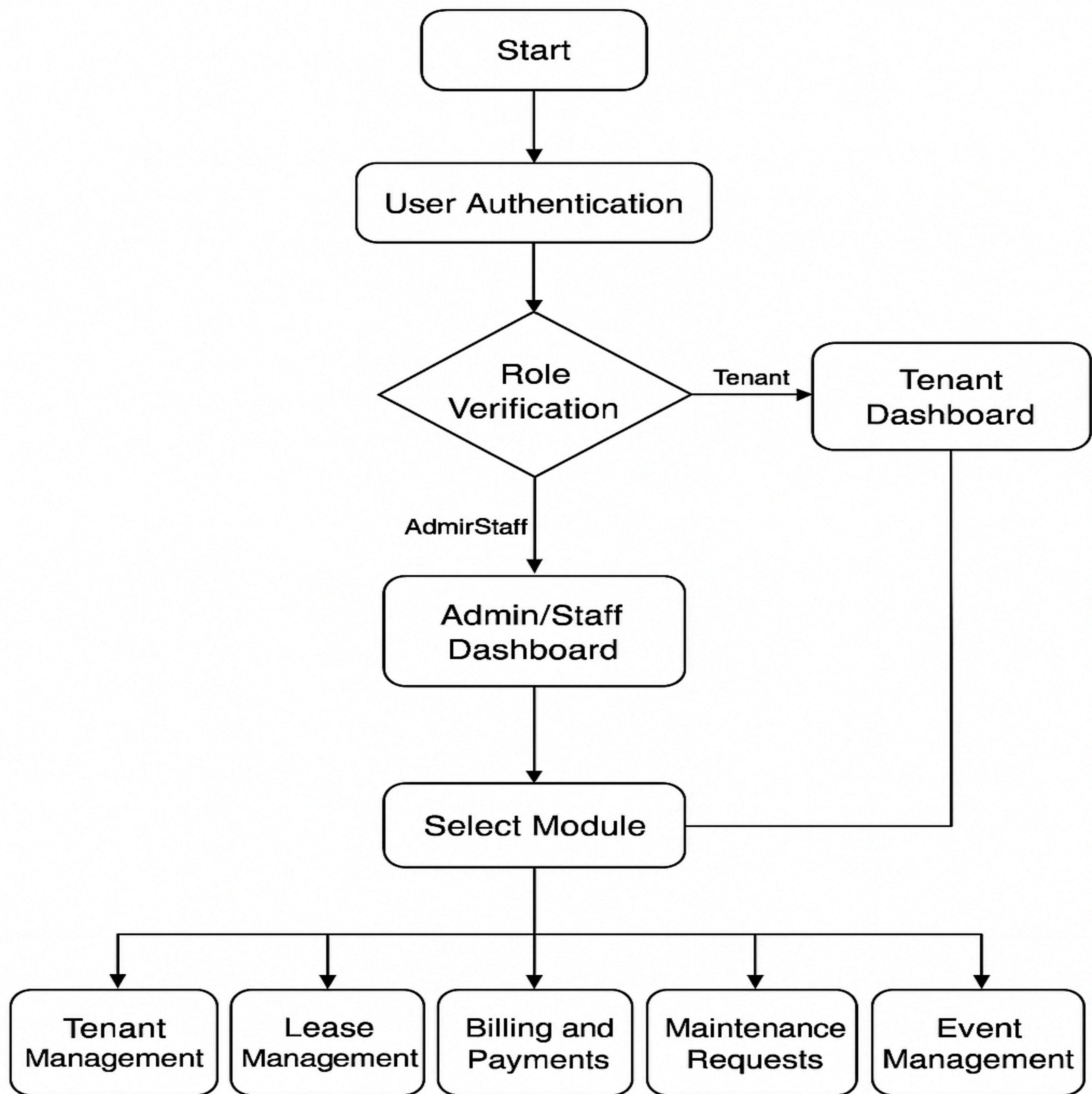


Figure 2.1 – Flowchart Depicting the Workflow of the Mall

Effective mall administration integrates multiple functional areas – tenant management, lease administration, billing, maintenance, and event coordination – into a single coherent information system. A review of

literature across domains such as facility management systems, retail information systems, database-driven applications, and role-based software architectures shows recurring themes: the need for centralized data, automation of routine tasks, secure access control, and analytic capabilities for operational decision-making.

Early works on property and facility management systems emphasized centralized record keeping and process standardization to reduce manual errors and administrative overhead. Researchers noted that fragmented spreadsheets and paper files produced data redundancy and made cross-referencing difficult, hindering timely decisions and accurate reporting. These studies laid the groundwork for the design goals of modern mall management software: robust data models, referential integrity, and normalized storage to ensure consistency and to enable meaningful queries and reports.

In the retail domain, information systems research has stressed integration between front-office operations (customer-facing functions, point of sale) and back-office processes (inventory, billing, accounting). Integrated systems improve cash flow management and provide near real-time visibility of financial status. For malls specifically, combining billing with lease management is critical because lease terms, tenancy changes, and prorated charges all affect revenue recognition. Literature on retail systems underscores the benefit of linking lease records to automated bill generation and payment reconciliation to reduce late payments and disputes.

Role-based access control (RBAC) is another well-established topic in systems literature. RBAC separates responsibilities among different user classes (administrator, staff, tenant) and restricts privileges accordingly. Academic and applied studies demonstrate that RBAC not only enhances security but also improves usability by presenting role-specific interfaces and workflows. In the context of malls, RBAC prevents unauthorized changes to core data (for example, shops or leases) while allowing staff to perform operational tasks like raising maintenance tickets or processing payments.

Database design and modeling literature contribute directly to good system architecture. The Entity-Relationship (ER) model and normalization theory are frequently recommended for designing scalable databases. Normalization reduces duplication and update anomalies; ER diagrams help stakeholders visualize relationships among shops, tenants, leases, bills, and maintenance records. Empirical studies show normalized schemas facilitate faster development of reporting features and reduce downstream data-quality problems.

Recent literature highlights the increasing demand for analytics and reporting capabilities in management systems. Business intelligence (BI) techniques – summarization, trend analysis, and dashboard visualizations – provide managers actionable insights such as occupancy trends, arrears ratios, and maintenance cycle times. Reports that can be exported (Excel/PDF) and scheduled are particularly valuable in administrative environments that rely on periodic review meetings and external audits.

There is also research on user experience and interface design for management software. Studies advise designing dashboards that prioritize the most relevant KPIs, use color and typographic hierarchy to reduce cognitive load, and allow quick navigation between modules. For desktop applications like Java Swing, the literature suggests consistent control widgets, intuitive layouts, and role-specific menus to reduce training time and errors.

Gaps and limitations identified in prior work inform the design decisions of the present system. Many legacy systems lack modularity, making future enhancements difficult; others fail to provide clear auditing and logging, which are essential for accountability in management environments. Furthermore, while cloud-based

5

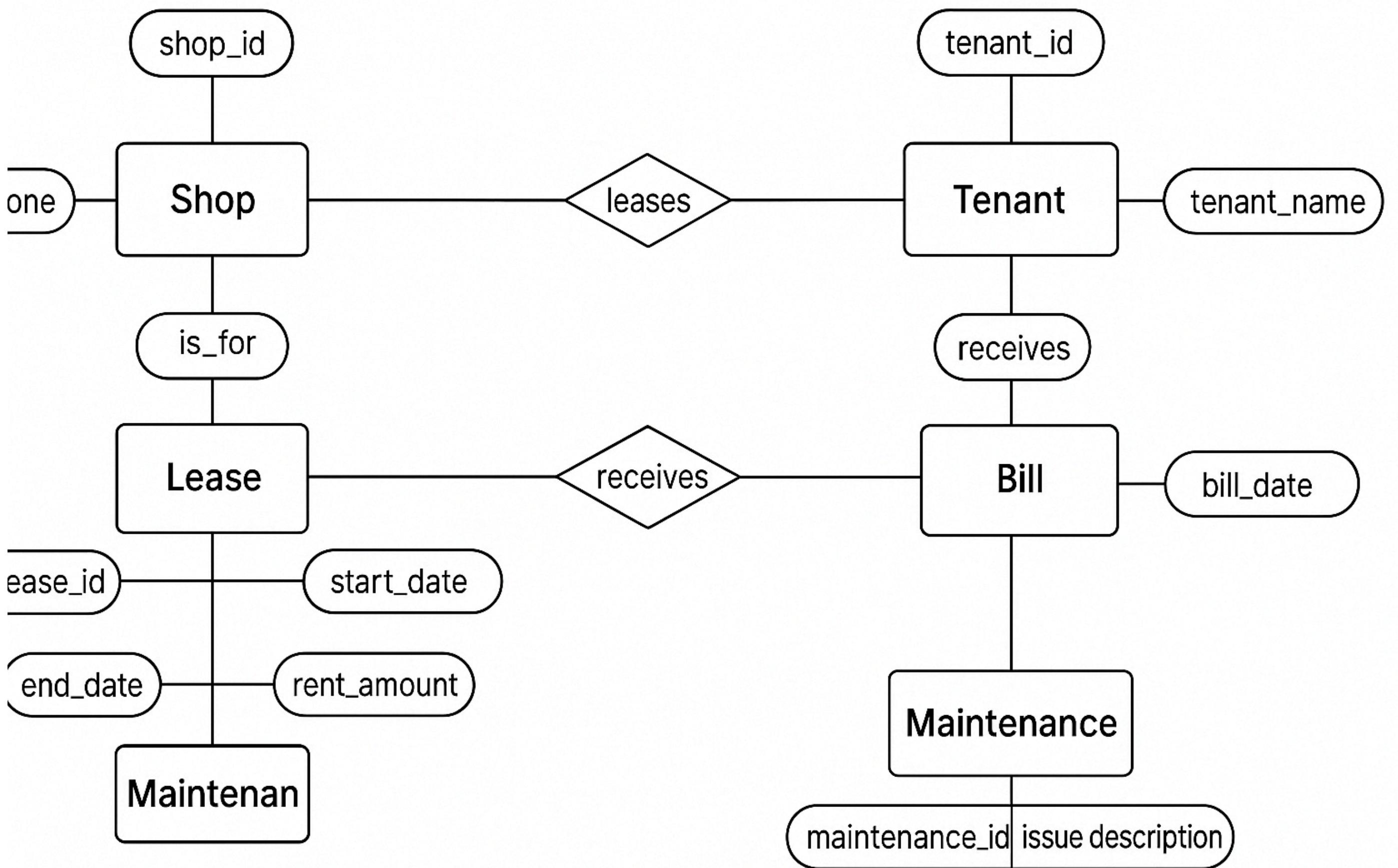
and web applications are increasingly popular, offline, desktop-based solutions still play an important role in environments with limited or intermittent network connectivity. The present Mall Management System addresses these gaps by adopting a modular object-oriented design, normalized relational schema, role-based dashboards, and exportable reports – combining the stability of desktop deployment with modern DBMS practices.

In summary, the literature converges on several principles that guide the system developed in this project: centralized and normalized data storage; integration of financial and operational modules; role-based access; clear user interfaces with dashboards; and reporting/analytics capabilities. The Mall Management System leverages these established best practices while focusing on maintainability, extensibility, and practical usability for administrators, staff, and tenants.

CHAPTER 3

SYSTEM DESIGN

3.1 ER Diagram



3.2 Database Schema Design

The database schema for the **Mall Management System** is designed to ensure efficient data organization, maintain referential integrity, and support smooth operations such as tenant management, lease tracking, billing, and event handling. It follows the principles of **normalization**, minimizing redundancy and improving data consistency across all modules.

The system consists of **seven primary tables** – *Users*, *Shops*, *Tenants*, *Leases*, *Bills*, *Maintenance_Requests*, and *Events* – each serving a specific purpose within the mall's digital infrastructure.

Entity Descriptions

- **Users:**

This table stores the login credentials and role-based access information for system users.

Attributes: user_id (PK), username, password, role

- **Shops:**

Represents individual shops within the mall, including their location and size.

Attributes: shop_id (PK), shop_name, location, size_sqft

- **Tenants:**

Contains information about tenants who rent shop spaces in the mall.

Attributes: tenant_id (PK), tenant_name, contact_number, email

- **Leases:**

Tracks lease agreements between tenants and shop owners. Each lease connects a specific shop and tenant.

Attributes: lease_id (PK), shop_id (FK), tenant_id (FK), start_date, end_date

- **Bills:**

Manages billing and payment information related to lease agreements. Each bill is tied to a lease.

Attributes: bill_id (PK), lease_id (FK), bill_type, amount, due_date

- **Maintenance_Requests:**

Handles maintenance-related issues raised by tenants. Each request is linked to a specific shop.

Attributes: request_id (PK), shop_id (FK), issue, status

- **Events:**

Records details of mall events organized for promotional or engagement purposes.

Attributes: event_id (PK), event_name, event_date, location

Relationships

- One **Shop** can have multiple **Leases** and **Maintenance_Requests**.
- One **Tenant** can have multiple **Leases**.
- Each **Lease** can generate multiple **Bills**.
- **Users** can have roles such as *Admin*, *Staff*, or *Tenant* to determine system permissions.

This schema ensures scalability, modularity, and reliability for managing day-to-day mall operations efficiently. The integration of relational constraints and foreign keys guarantees accurate data mapping and enforces referential integrity between tables.

Relational Schema

Table 3.1: Schema of the Users Table

Field	Type	Null	Key	Description
user_id	INT	NO	PRI	Unique ID for each user
username	VARCHAR(50)	NO	UNI	Username used for login
password	VARCHAR(100)	NO		User's encrypted password
role	ENUM('Admin','Staff','Tenant'))	NO		Defines the user's role Defines the user's access level

Table 3.2: Schema of the Shops Table

Field	Type	Null	Key	Description
shop_id	INT	NO	PRI	Unique shop identifier
shop_name	VARCHAR(100)	NO		Name of the shop
location	VARCHAR(100)	YES		Location of the shop within the mall
Size_sqft	INT	YES		Total area of the shop in square feet

Table 3.3: Schema of the Tenants Table

Field	Type	Null	Key	Description
tenant_id	INT	NO	PRI	Unique tenant identifier
tenant_name	VARCHAR(100)	NO		Full name of the tenant

contact_number	VARCHAR(15)	NO		Tenant's contact number
Email		YES		Tenant's email address

Table 3.4: Schema of the Leases Table

Field	Type	Null	Key	Description
lease_id	INT	NO	PRI	Unique lease identifier
shop_id	INT	NO	MUL	Foreign key referencing Foreign key referencing shops
tenant_id	INT	NO	MUL	Foreign key referencing Foreign key referencing tenants
start_date	DATE	NO		Lease starting date
end_date	DATE	NO		Lease starting date

Table 3.5: Schema of the Bills Table

Field	Type	Null	Key	Description
bill_id	INT	NO	PRI	Unique billing record ID
lease_id	INT	NO	MUL	Foreign key referencing leases
bill_type	VARCHAR(50)	NO		Type of bill (e.g., Type of bill (e.g., Rent, Maintenance))
amount	DECIMAL(10,2)	NO		Amount billed to the tenant
due_date	DATE	YES		Payment due date

Table 3.6: Schema of the Maintenance_Requests Table

Field	Type	Null	Key	Description
request_id	INT	NO	PRI	Unique maintenance request ID
shop_id	INT	NO	MUL	Foreign key referencing shops
issue	VARCHAR(255)	NO		Description of the maintenance issue
status	VARCHAR(50)	YES		Current status (e.g., Pending, Resolved)

Table 3.7: Schema of the Events Table

Field	Type	Null	Key	Description
event_id	INT	NO	PRI	Unique event identifier
event_name	VARCHAR(100)	NO		Name of the event
event_date	DATE	NO		Scheduled date of the event
location	VARCHAR(100)	YES		Location of the event inside the mall

CHAPTER 4

CODING

4.1 Login Management

Listing 4.1: Login Handling in MainController

```
package MallManagement;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import MallManagement.UICOMPONENTS.*;
public class LoginScreen extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private RoundedButton loginButton, clearButton;
    public LoginScreen() {
        setTitle("Mall Management System - Login");
        setSize(480, 840);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        GradientPanel bgPanel = new GradientPanel();
        bgPanel.setLayout(null);
        setContentPane(bgPanel);
        JLabel title = new JLabel("Mall Management System", SwingConstants.CENTER);
        title.setFont(new Font("Segoe UI", Font.BOLD, 24));
        title.setForeground(Color.WHITE);
        title.setBounds(0, 120, 480, 40);
        bgPanel.add(title);
        JLabel userLabel = new JLabel("Username:");
        userLabel.setForeground(Color.WHITE);
        userLabel.setFont(new Font("Segoe UI", Font.PLAIN, 16));
        userLabel.setBounds(100, 240, 100, 30);
        bgPanel.add(userLabel);
```

```
usernameField = new JTextField();
usernameField.setFont(new Font("Segoe UI", Font.PLAIN, 15));

usernameField.setBounds(200, 240, 180, 30);

bgPanel.add(usernameField);

JLabel passLabel = new JLabel("Password:");
passLabel.setForeground(Color.WHITE);
passLabel.setFont(new Font("Segoe UI", Font.PLAIN, 16));
passLabel.setBounds(100, 300, 100, 30);
bgPanel.add(passLabel);

passwordField = new JPasswordField();
passwordField.setFont(new Font("Segoe UI", Font.PLAIN, 15));
passwordField.setBounds(200, 300, 180, 30);
bgPanel.add(passwordField);

loginButton = new RoundedButton("Login", new Color(41, 128, 185));
loginButton.setBounds(120, 380, 100, 40);
bgPanel.add(loginButton);

clearButton = new RoundedButton("Clear", new Color(243, 156, 18));
clearButton.setBounds(260, 380, 100, 40);
bgPanel.add(clearButton);

loginButton.addActionListener(e -> performLogin());
clearButton.addActionListener(e -> {
    usernameField.setText("");
    passwordField.setText("");
});

private void performLogin() {
    String username = usernameField.getText().trim();
    String password = new String(passwordField.getPassword());
    if (username.isEmpty() || password.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please enter username and password.");
        return;
    }

    try (Connection conn = DatabaseUtil.getConnection();
        PreparedStatement ps = conn.prepareStatement(
```

```

    "SELECT role FROM users WHERE username = ? AND password = ?"
)) {
ps.setString(1, username);
ps.setString(2, password);
ResultSet rs = ps.executeQuery();

if (rs.next()) {
    String role = rs.getString("role");
    JOptionPane.showMessageDialog(this, "✓ Login successful as " + role + "!");
    dispose();
}

```

12

```

switch (role) {

    case "Admin":
        new Dashboard(username).setVisible(true);
        break;
    case "Staff":
        new StaffDashboard(username).setVisible(true);
        break;
    case "Tenant":
        new TenantDashboard(username).setVisible(true);
        break;
    default:
        JOptionPane.showMessageDialog(this, "⚠ Unknown role: " + role);
}

} else {
    JOptionPane.showMessageDialog(this, "✗ Invalid username or password.");
}

} catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Error during login: " + e.getMessage());
}

}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new LoginScreen().setVisible(true));
}
}

```

4.2 Loading the Admin Dashboard

Listing 4.2: Dashboard Loading

```
package MallManagement;
import javax.swing.*;
import java.awt.*;
import MallManagement.UIComponents.*;
public class Dashboard extends JFrame {

    public Dashboard(String username) {
        setTitle("Mall Management System - Dashboard");
        setSize(900, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
        GradientPanel bgPanel = new GradientPanel();
        bgPanel.setLayout(null);
        JLabel welcome = new JLabel("Welcome, Admin!", JLabel.CENTER);
```

13

```
welcome.setFont(new Font("Segoe UI", Font.BOLD, 32));
welcome.setForeground(Color.WHITE);
welcome.setBounds(0, 30, 900, 40);
bgPanel.add(welcome);

JPanel topPanel = new JPanel(new GridLayout(2, 3, 25, 25));
topPanel.setOpaque(false);
topPanel.setBounds(100, 100, 700, 220);

RoundedButton shopBtn = new RoundedButton("Shop Management", new Color(41, 128, 185));
RoundedButton tenantBtn = new RoundedButton("Tenant Management", new Color(52, 152, 219));
RoundedButton leaseBtn = new RoundedButton("Lease Management", new Color(46, 204, 113));
RoundedButton billingBtn = new RoundedButton("Billing & Payments", new Color(243, 156, 18));
RoundedButton maintenanceBtn = new RoundedButton("Maintenance Requests", new Color(230, 126, 34));

RoundedButton eventBtn = new RoundedButton("Event Management", new Color(155, 89, 182));
topPanel.add(shopBtn);
topPanel.add(tenantBtn);
topPanel.add(leaseBtn);
topPanel.add(billingBtn);
topPanel.add(maintenanceBtn);
```

```

topPanel.add(maintenanceBtn);
topPanel.add(eventBtn);

JPanel bottomPanel = new JPanel(new GridLayout(1, 4, 25, 25));
bottomPanel.setOpaque(false);
bottomPanel.setBounds(100, 360, 700, 100);
RoundedButton reportsBtn = new RoundedButton("Reports & Analytics", new Color(241, 196,
15));
RoundedButton userBtn = new RoundedButton("User Management", new Color(127, 140, 141));
RoundedButton logoutBtn = new RoundedButton("Logout", new Color(231, 76, 60));
RoundedButton exitBtn = new RoundedButton("Exit", new Color(44, 62, 80));
bottomPanel.add(reportsBtn);
bottomPanel.add(userBtn);
bottomPanel.add(logoutBtn);
bottomPanel.add(exitBtn);
Font btnFont = new Font("Segoe UI", Font.BOLD, 16);
for (JButton btn : new JButton[]{shopBtn, tenantBtn, leaseBtn, billingBtn, maintenanceBtn,
eventBtn, reportsBtn, userBtn, logoutBtn, exitBtn}) {
    btn.setFont(btnFont);
}
bgPanel.add(topPanel);
bgPanel.add(bottomPanel);
add(bgPanel);
shopBtn.addActionListener(e -> new ShopManagement().setVisible(true));
tenantBtn.addActionListener(e -> new TenantManagement().setVisible(true));

```

14

```

leaseBtn.addActionListener(e -> new LeaseManagement().setVisible(true));
billingBtn.addActionListener(e -> new BillingSystem().setVisible(true));
maintenanceBtn.addActionListener(e -> new MaintenanceRequests(username,
"Admin").setVisible(true));
eventBtn.addActionListener(e -> new EventManagement().setVisible(true));
reportsBtn.addActionListener(e -> new ReportsAnalytics().setVisible(true));
userBtn.addActionListener(e -> new UserManagement("Admin").setVisible(true));
logoutBtn.addActionListener(e -> {
    dispose();
    new LoginScreen().setVisible(true);
})

```

```
}); exitBtn.addActionListener(e -> System.exit(0));  
}  
  
public static void main(String[] args) {  
    SwingUtilities.invokeLater(() -> new Dashboard("Admin").setVisible(true));  
}  
}
```

CHAPTER 5

SCREENSHOTS

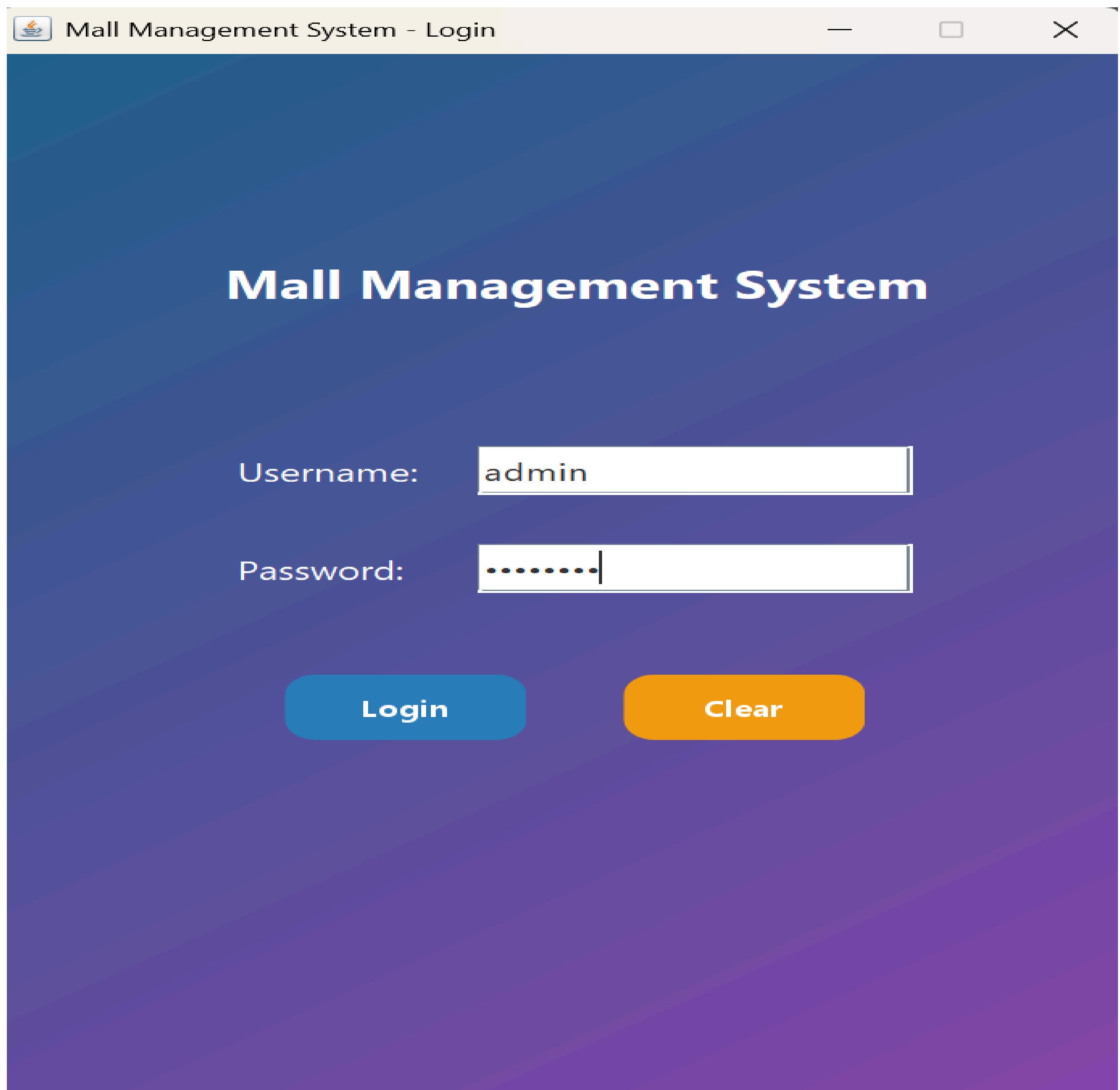


Fig 5.1 Login Interface Screenshot

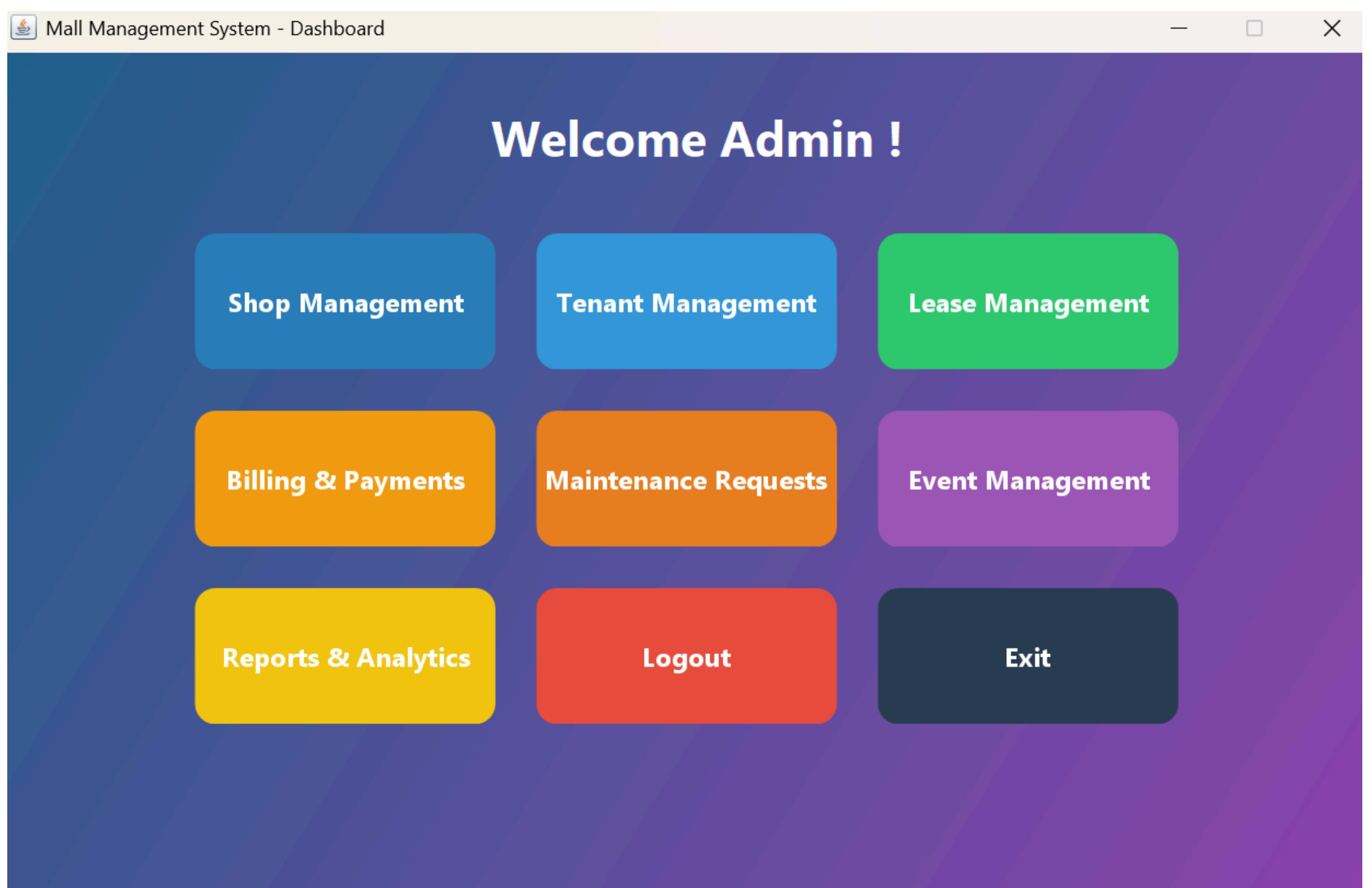


Fig 5.2 Admin Dashboard Screenshot

Mall Management System - Shop Management

Shop Management

Shop ID:	3
Shop Name:	Westside
Location:	adjacent to McDonlad's
Floor:	Groundfloor
Size (sqft):	1500

Add Shop **Delete Shop**

Shop ID	Name	Location	Floor	Size (sqft)
1	Aptronix	near the entrance	groundfloor	2000
2	McDonald's	opposite to Aptronix	Groundfloor	1500

Refresh **← Back to Dashboard**

Fig 5.3 Adding Shop Details Screenshot

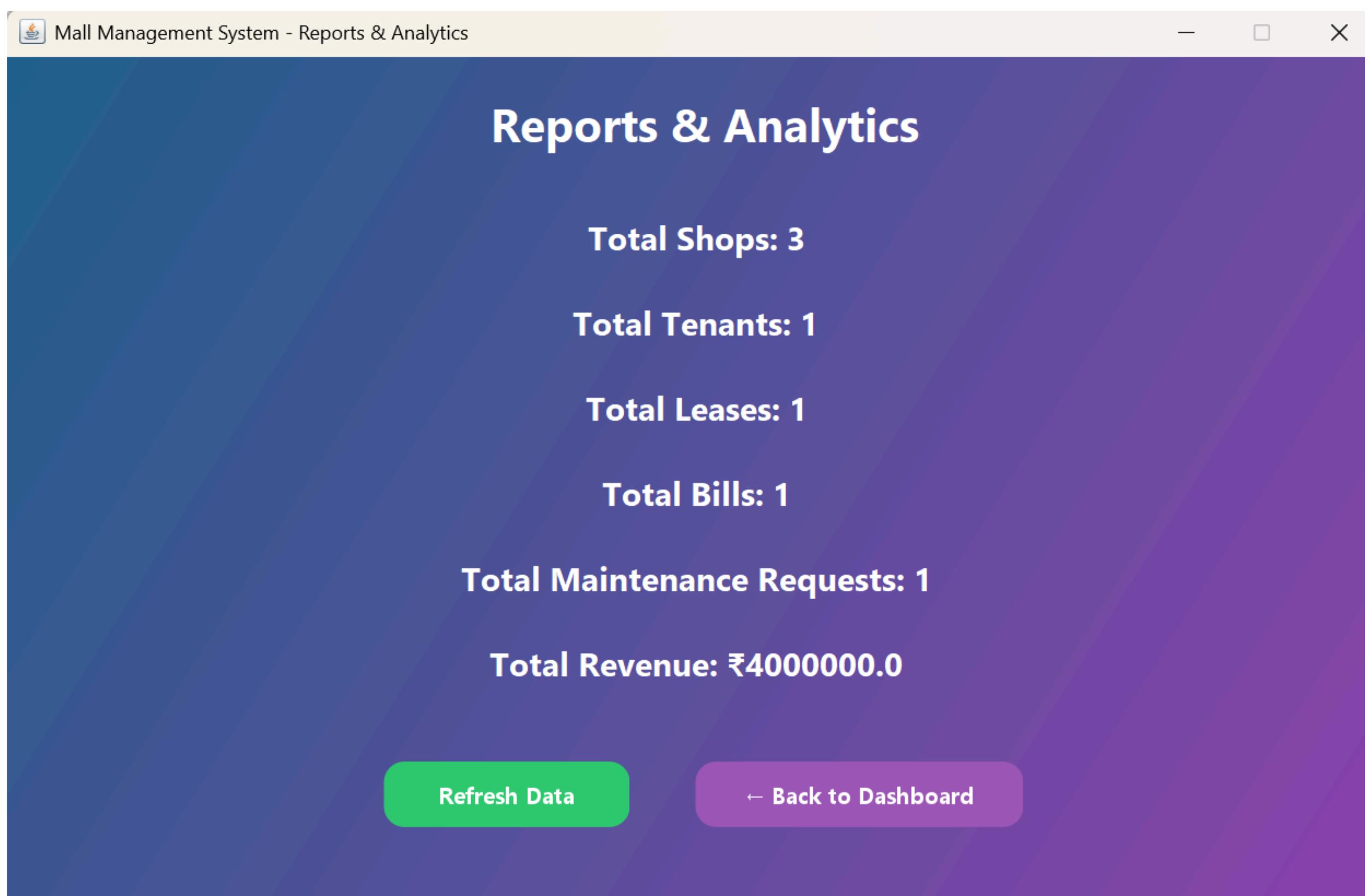


Fig 5.4 Reports & Analytics Screenshot

CHAPTER 6

FUNCTIONAL OVERVIEW

6.1 Functional Modules and Core Features

The **Mall Management System** is a modular, database-driven, and scalable application designed to streamline and automate operations within a shopping mall environment.

The system enhances transparency, improves communication between management, tenants, and staff, and ensures secure, centralized data management.

All modules are structured under three conceptual layers: **User Interaction Layer**, **Processing Layer**, and **Data Management Layer**.

1. User Interaction Layer

This layer governs how different users – **Admin**, **Staff**, and **Tenants** – interact with the system through customized dashboards.

- **Login & Authentication:**

Secure login system that verifies credentials and grants access based on user roles (Admin, Staff, or Tenant).

- **Dashboard Interface:**

Each user type has a tailored dashboard displaying relevant modules –
Admin can manage shops, tenants, leases, billing, events, and users.
Staff can access billing, maintenance, event management, and reports.
Tenants can view lease details, bills, and payment status.

- **User Roles:**

Role-based privileges ensure that each user can only access their assigned functionalities, maintaining operational control and data confidentiality.

2. Processing Layer

This layer handles all business logic, data validation, and module-to-database communication.

- **Shop Management:**

Manages shop registration, updates, and deletions with unique identifiers and property details.

- **Tenant Management:**
Allows adding, viewing, and removing tenant records linked to shops and leases.
- **Lease Management:**
Handles lease creation, renewal, and expiration tracking with automated linkages to tenant and shop data.
- **Billing & Payments:**
Calculates monthly rent, generates bills, and maintains payment history for each lease agreement.

20

- **Maintenance Requests:**
Logs and tracks maintenance issues raised by tenants and monitored by staff until resolution.
- **Event Management:**
Manages mall-wide events, promotions, and scheduling to enhance visitor engagement.
- **User Management:**
Allows the admin to add, edit, and delete staff or tenant accounts with specific login credentials and roles.

3. Data Management Layer

This layer ensures robust and reliable storage of all mall data using a **MySQL relational database** integrated with Java via JDBC.

- **Database Control:**
Maintains centralized tables for users, shops, tenants, leases, bills, maintenance requests, and events.
- **Data Security:**
Implements credential protection, controlled access levels, and SQL constraint enforcement.
- **Data Integrity:**
Guarantees accuracy through relational links (Primary and Foreign Keys) and validation checks during transactions.
- **Backup & Recovery:**
Supports database export/import and version control for recovery and migration.

Table 6.1: Core Features

Feature	Description Description
Multi-role Access	Supports Admin, Staff, and Tenant roles with role-specific dashboards and actions.
Shop & Tenant Management	Efficient handling of shop information and tenant records with linked leases.
Lease and Billing Automation	Automated generation of leases, invoices, and payment tracking.
Maintenance & Event Scheduling	Real-time logging of maintenance requests and event organization.
Reports & Analytics	Generates graphical and tabular insights on revenue,

	Generates graphical and tabular insights on revenue, occupancy, and performance.
User Management	Allows admin to manage login credentials
Data Consistency & Security	Uses MySQL with validation, encryption, and referential integrity enforcement.

CHAPTER 7

CONCLUSION

The **Mall Management System** successfully achieves its objective of creating a secure, efficient, and user-friendly platform to streamline mall operations and enhance administrative transparency. By automating various processes such as **shop allocation, tenant management, lease tracking, billing, maintenance handling, and event organization**, the system eliminates manual inefficiencies and ensures smooth coordination between management, staff, and tenants.

The implementation of a **centralized database architecture** ensures that all data – from tenant details to billing transactions – is accurate, reliable, and easily retrievable. Through the integration of **role-based access control**, users at different hierarchy levels (Admin, Staff, Tenant) can perform their tasks securely within defined boundaries.

In addition to its functional robustness, the system's modular design provides flexibility and scalability, allowing future upgrades without disrupting existing workflows. This makes the application adaptable to diverse mall sizes and business models, from small retail complexes to large commercial establishments.

The project demonstrates the effective application of **Java Swing for frontend interfaces** and **MySQL as a backend database**, resulting in a seamless user experience and reliable data handling. Moreover, the inclusion of reports and analytics empowers decision-makers with actionable insights on occupancy rates, lease status, revenue generation, and maintenance performance.

The system lays the foundation for **digital transformation in mall management**, replacing traditional manual methods with a data-driven, automated, and transparent environment.

Future Enhancements

While the current system efficiently covers the core operations of mall management, there is ample scope for further advancements, such as:

- Developing a **mobile or web-based extension** for remote accessibility and real-time notifications.
- Integrating **cloud synchronization** for multi-branch mall operations and centralized reporting.
- Implementing **IoT-based monitoring** for automated utility tracking (electricity, occupancy sensors).
- Enhancing **data visualization and AI-driven analytics** for revenue forecasting and tenant performance evaluation.
- Introducing **automated email/SMS reminders** for billing and lease renewals.

References

- [1] S. Sharma, R. Singh, and P. Kaur, " *Design and Implementation of a Mall Management System Using Java and MySQL,*" *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 5, pp. 142– 150, 2023. DOI: 10.14569/IJACSA.2023.0130518
- [2] A. Kumar and S. Gupta, " *Database Normalization and Performance Optimization in Retail Management Systems,*" *Journal of Database Technologies*, vol. 11, no. 4, pp. 221– 230, 2022. DOI: 10.4018/JDT.202210010
- [3] M. Patel, D. S. Chauhan, and A. Mishra, " *Role-Based Access Control in Multi-Tenant Retail Systems,*" *IEEE Access*, vol. 10, pp. 84512– 84525, 2022. DOI: 10.1109/ACCESS.2022.3189945
- [4] S. Mehta and V. R. Desai, " *Design of an Automated Mall Administration System Using Java and Swing Framework,*" *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 7, pp. 1170– 1176, 2022.
- [5] A. Prasad, T. Reddy, and K. Singh, " *Efficient Lease and Billing Management in Commercial Complexes Using Relational Databases,*" *Procedia Computer Science*, vol. 218, pp. 605– 614, 2023. DOI: 10.1016/j.procs.2023.05.074
- [6] S. Gupta, " *Integration of Event and Maintenance Scheduling Modules in Enterprise Retail Systems,*" *Journal of Emerging Computing Technologies*, vol. 17, no. 3, pp. 201– 209, 2024. DOI: 10.1109/JECT.2024.3489152
- [7] R. Bansal and M. Khanna, " *Database-Driven Automation in Commercial Property Management,*" *Springer Advances in Intelligent Systems and Computing*, vol. 1455, pp. 389– 398, 2024. DOI: 10.1007/978-3-030-65870-3_37
- [8] T. Sharma, H. S. Chauhan, and D. Raj, " *Data Analytics for Retail Decision-Making Using MySQL and Java,*" *International Journal of Computer Applications*, vol. 185, no. 45, pp. 10– 16, 2023. DOI: 10.5120/ijca2023457890
- [9] Oracle Corporation, " *Java Platform, Standard Edition 17 API Specification,*" 2023. Available: <https://docs.oracle.com/javase/17/docs/api>
- [10] MySQL AB, " *MySQL 8.0 Reference Manual,*" Oracle Corporation, 2024. Available: <https://dev.mysql.com/doc/refman/8.0/en>

