Rayat Shikshan Sanstha's

C. D. Jain College Of Commerce, Shrirampur

A Project Report On

"Stock Management System"

Submitted to

University of Pune

In Partial Fulfilment of The Requirement of

BBA(CA)- III

(Bachelor of Business Administration and Department of Computer Application)

Submitted By

Mr. Sarode Siddharth Ravindra and Mr. Sadavarte Gopal Ravindra

Under a Guidance of

Prof. Mr. Shaikh A. A. Sir

During The Academic Year **2024-25**

RAYAT SHIKSHAN SANTHA'S

C. D. JAIN COLLEGE OF COMMERCE, SHRIRAMPUR

CERTIFICATE

(Department of BBACA)

This is certify that Project entitled **Stock Management System** "Submitted by **Mr. Sarode Siddharth Ravindra and Mr. Sadavarte Gopal Ravindra** student of **BBA(CA)-III** (Bachelor Of Business Administration and Computer Application) had satisfactorily completed the project during the academic year **2024-25.**

Date: / / 2025

Project Guide

Prof. Mr. Shaikh A. A. Sir

Head of Department

Prof. Mr. Shaikh A. A. Sir

Internal Examiner

External Examiner

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organization. We would to kind to extend our sincere thanks to all of them.

First and foremost, we want to thanks Prof. Mr. Shaikh A. A. Sir HOD (BBACA) in C. D. Jain College, Shrirampur for giving us an opportunity to work on this project.

We are highly indebted to Prof. Mr. Shaikh A. A. Sir (Project Guide) for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in the project.

We would like to express our gratitude towards our parents & members of Information technology department for their kind co-operation and encouragement which help us in completion of this project.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us with their abilities.

Place: Shrirampur **Signature**

Date: / / 2025 Mr. Sarode Siddharth Ravindra

Signature

Mr. Sadavarte Gopal Ravindra

DECLARATION

We **Mr. Sarode Siddharth Ravindra** and **Mr. Sadavarte Gopal Ravindra** students of **BBA(CA)-III**, C.D Jain college Of Commerce, Shrirampur declare that the project entitled **Stock Management System** Have been completed successfully & this project is submitted towards the partial fulfillment of the requirement of the degree of BBA(CA).This project is not submitted for any other degree, Diploma or other similar title or prize in any other university.

Place: Shrirampur **Signature**

Date: / / 2025 Mr. Sarode Siddharth Ravindra

Signature

Mr. Sadavarte Gopal Ravindra

Index

Sr. No	Index Name	Page No
1	**Introduction:** 1.1 Motivation 1.2 Problem system 1.3 Purpose/objective & goal 1.4 Literature Survey 1.5 Project scope & limitations	6-7
2	**System Analysis:** 2.1 Existing System 2.2 Scope and limitations of system 2.3 Project Perspective & feature 2.4 Stakeholders 2.5 Requirement Analysis – Functional Requirements Performance Requirements Security Requirements	8-10
3	**System Design:** 3.1 Design Constraints 3.2 System Model DFD 3.3 Data Model 3.4 User Interfaces	11-27
4	**Implementation Details:** 4.1 Software/hardware specifications	28
5	**Outputs and Reports Testing:** 5.1 Output 5.2 Data Validation 5.3 Test cases	29
6	**Conclusion and Recommendations**	30
7	**Future scope**	31
8	**Bibliography and References**	32

1. Introduction

The Stock Management System is designed to simplify stock management. It helps businesses efficiently track stock levels, generate reports, and manage employee salaries. By replacing traditional manual tasks with a computer-based system, it reduces errors, saves time, and improves accuracy in stock handling and financial records.

The system automates complex calculations, minimizes paperwork, and enhances overall efficiency. Built using Core Java and PostgreSQL, this project provides a reliable and user-friendly solution for managing stocks and related business operations.

1.1 Motivation

Manual stock management is time-consuming and prone to errors. Traditional methods involve paperwork and complex calculations, leading to inefficiencies. This project aims to automate stock tracking, report generation, and salary management, making the process faster and more accurate. Using Core Java and PostgreSQL, we ensure a secure, efficient, and user-friendly system to improve business productivity.

1.2 Problem Statement

Many businesses still rely on manual stock management, Which leads to errors, inefficiency, and time-consuming processes. Tracking inventory, calculation stock levels, and managing employee salaries manually increases the risk of miscalculations and data loss. This project aims to develop a computerized system to automate stock tracking, report generation, and salary management, ensuring accuracy, efficiency, and ease of use.

1.3 Objective and Goals

Objective -

The main objective of this project is to develop a computerized Stock Management System that simplifies stock tracking, automates report generation, and manages employee salaries efficiently. It aims to eliminate manual errors, reduce paperwork, and improve business productivity by providing a secure and user-friendly system.

Goals -

1. **Automate Stock Management –** Track stock levels in real-time production.
2. **Reduce Manual Errors –** Minimize mistakes in calculations and record-keeping.
3. **Generate Reports Efficiently –** Provide automated stock reports, employee salary report etc.
4. **Save Time and Effort –** Speed up stock-related operations and reduce dependency on paperwork.
5. **Improve Business Productivity –** Enhance workflow efficiency with a structured and reliable system.

1.4 Literature Survey

1. **Existing Stock Management Methods –**
    - Traditional stock management relies on manual registers, spreadsheets, or basic software.
    - Manual record-keeping leads to errors, data loss, and inefficiencies.
2. **Research on Automated Systems –**
    - Studies show that automated inventory systems improve stock accuracy and reduce human effort.
    - Many businesses use ERP (Enterprise Resource Planning) based inventory management but these are costly and complex.
    - There is a need for affordable, user-friendly, and customizable stock management solutions.
3. **Need For a New System –**
    - Existing solutions may not be cost-effective or easy to use, especially for small businesses.
    - This project aims to develop a stock management system using Core Java and PostgreSQL.
    - The system will provide automation, efficient stock handling, reducing manual work and improving business productivity.

1.5 Project Scope and Limitations.

Project Scopes –

1. **Stock Management –** Tracks stock levels, product details, and inventory updates efficiently.
2. **Report Generation –** Provides reports on stock status, sales, and employee salaries.
3. **Employee Management –** Maintains employee details and salary records.
4. **Automation & Efficiency –** Reduces manual errors, saves time, and streamlines inventory operations.
5. **Secure Database Storage –** Uses PostgreSQL for data storage, ensuring security and reliability.
6. **Future Scalability –** Can be extended to include multi-user roles, cloud storage, and AI-based analytics.

Limitations –

1. **Manual Data Entry –** Stock updates and employee details need to be entered manually.
2. **Single-User Access –** In its basic version, the system may not support multiple user roles or remote access.

2. Systems Analysis

2.1 Existing System

The existing stock management system relies on traditional paper-based record-keeping. Employee management, including salary processing, attendance tracking, and other records, is maintained manually, making it difficult to manage. Calculations are complex and time-consuming, often leading to inefficiencies and errors.

2.2 Scope and limitations of system

Scope:

1. Stock Tracking – Inventory levels are recorded manually, with updates based on physical verification.
2. Transparency - The records are maintained in front of others, there for the transparency and clarity are maintained.

Limitations :

1. Time-Consuming Processes - Manual entry and calculations slow down operations.
2. Error-Prone - Human errors in stock tracking, payroll calculations, and attendance management are common.
3. Data Loss & **Security** Issues - Paper records are vulnerable to loss, damage, or unauthorized access.
4. Difficult Report Generation - Preparing reports requires manual data compilation, which is inefficient and prone to mistakes.

2.3 Project perspective, features

Introduction

Stock Management System simplifies stock tracking, reporting, and salary management. It replaces manual tasks, reducing errors and saving time. Automates calculations, cuts paperwork, and boosts efficiency. Built with Core Java & PostgreSQL, it is a reliable, user-friendly solution for business ops.

Purpose

The purpose of this project is to develop a centralized and automated system for managing stock and employee records efficiently. It aims to reduce manual errors, enhance data security, and ensure real-time updates for inventory tracking, payroll processing, and attendance management. By streamlining these operations, the system improves accuracy, saves time, and simplifies report generation for better decision-making.

Objectives

The main objective of this project is to develop a computerized Stock Management System that simplifies stock tracking, automates report generation, and manages employee salaries efficiently. It aims to eliminate manual errors, reduce paperwork, and improve business productivity by providing a secure and user-friendly system.

Scope

1. Stock Management – Tracks stock levels, product details, and inventory updates efficiently.
2. Report Generation – Provides reports on stock status, sales, and employee salaries.
3. Employee Management – Maintains employee details and salary records.
4. Automation & Efficiency – Reduces manual errors, saves time, and streamlines inventory operations.
5. Secure Database Storage – Uses PostgreSQL for data storage, ensuring security and reliability.
6. Future Scalability – Can be extended to include multi-user roles, cloud storage, and AI-based analytics.

Features

1. Data Export
2. User-Friendly Interface
3. Database Integration
4. Employee Salary Management
5. Stock Inventory Management

2.4 Stakeholders

1. Admin

2.5 Requirement Analysis:

Functional Requirements:

1. User Management
   - Login/logout for admin and employees.
   - Role-based access (Admin, Manager, Employee).
   - Add, edit, or delete user accounts.
2. Stock Management
   - Add, update, and delete stock details
   - Search and filter products by size or date.
3. Reporting
   - Generate stock, and financial reports.
   - Export reports in PDF.

4. Salary Management
    - Calculate employee salaries based on role/hours.
    - Generate salary slips.

Performance Requirements:

1. System Response Time
    - Should load data within 2-3 seconds (for up to 10,000 records).
2. Database Efficiency
    - PostgreSQL should efficiently manage CRUD operations (Create, Read, Update, Delete).
    - Backup and restore data without significant downtime.
3. Scalability
    - Support future expansion (e.g., adding more products/users).

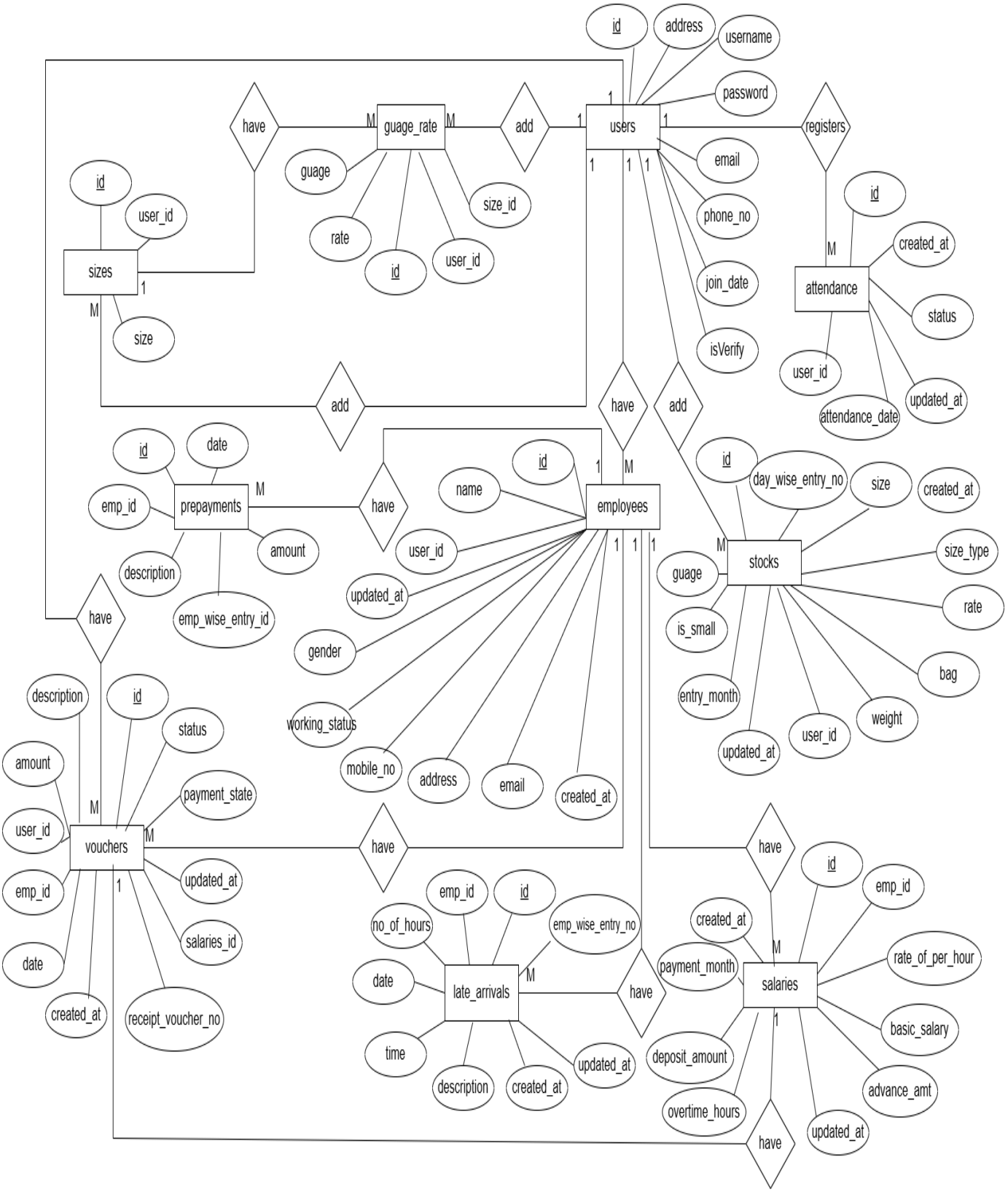
Security Requirements

1. Authentication & Authorization
    - Password-protected login with encryption (e.g., SHA-256).
    - Role-based permissions (e.g., only admin can modify salaries).
2. Data Protection
    - Prevent SQL injection attacks.
    - Sensitive data (salaries, passwords) stored securely.
3. Audit & Logs
    - Maintain logs of user activities (e.g., login attempts, stock changes).
    - Backup data weekly to prevent loss.

3. System Design

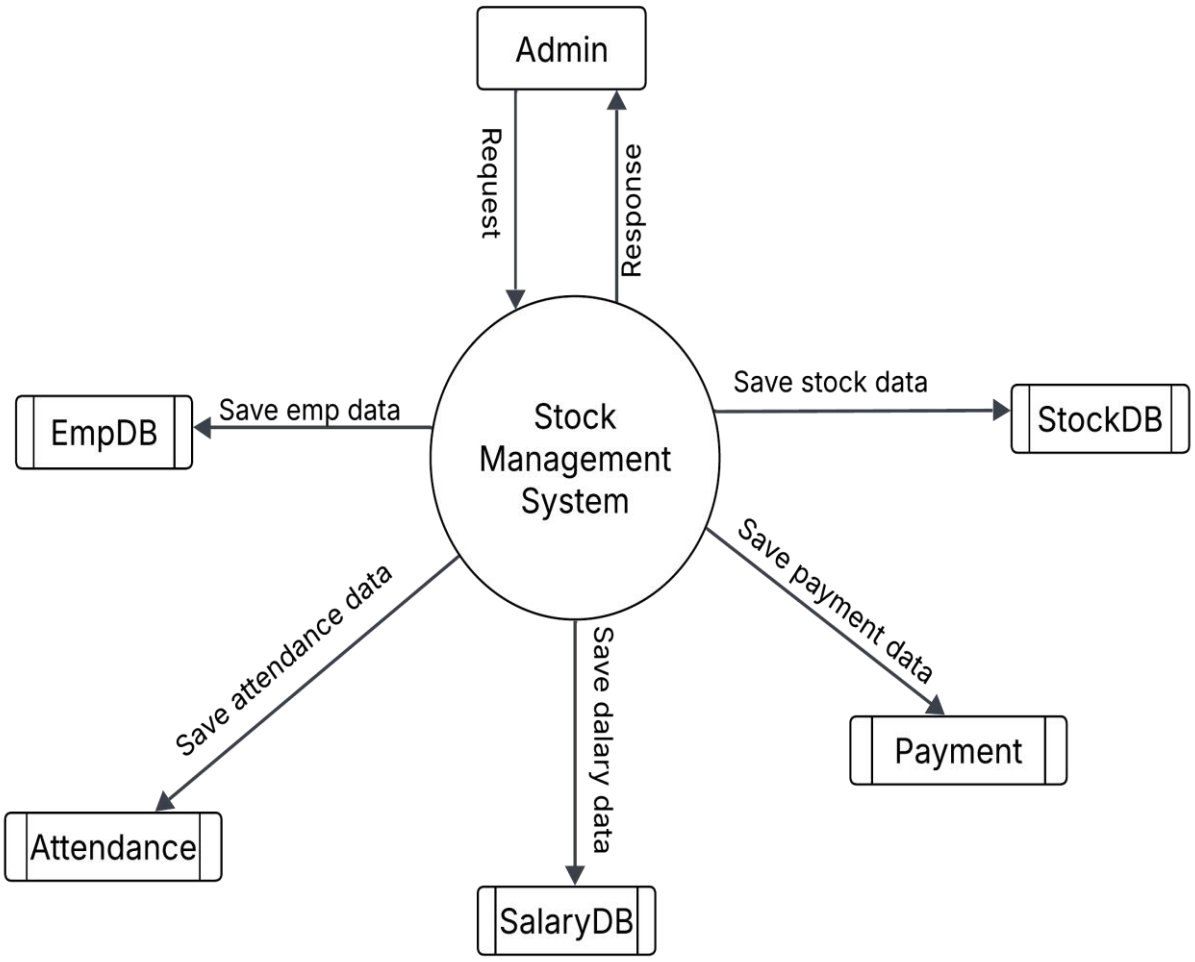
3.1 Design constraints

ER Diagram

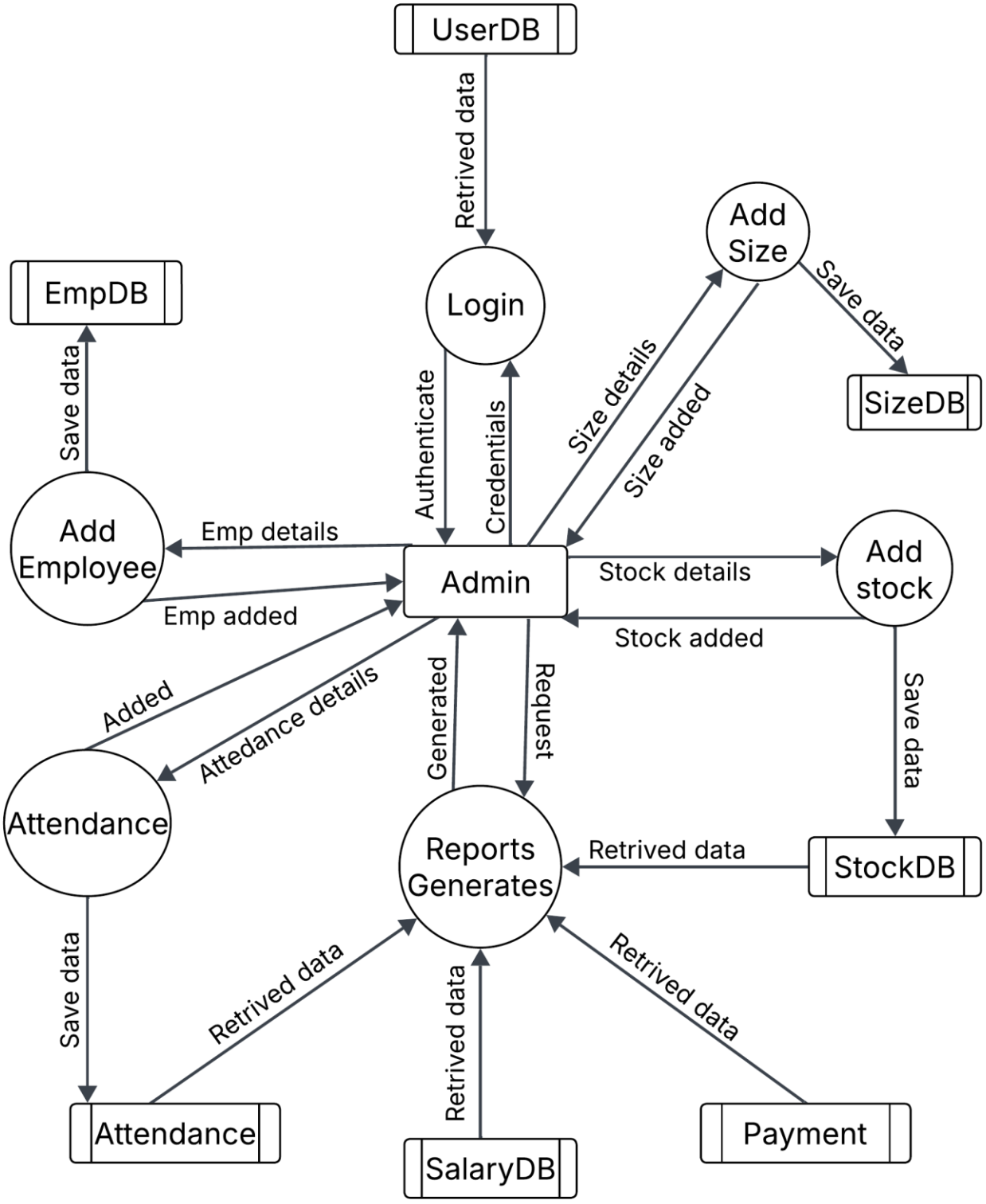


3.2 System Model: DFD

Context level DFD

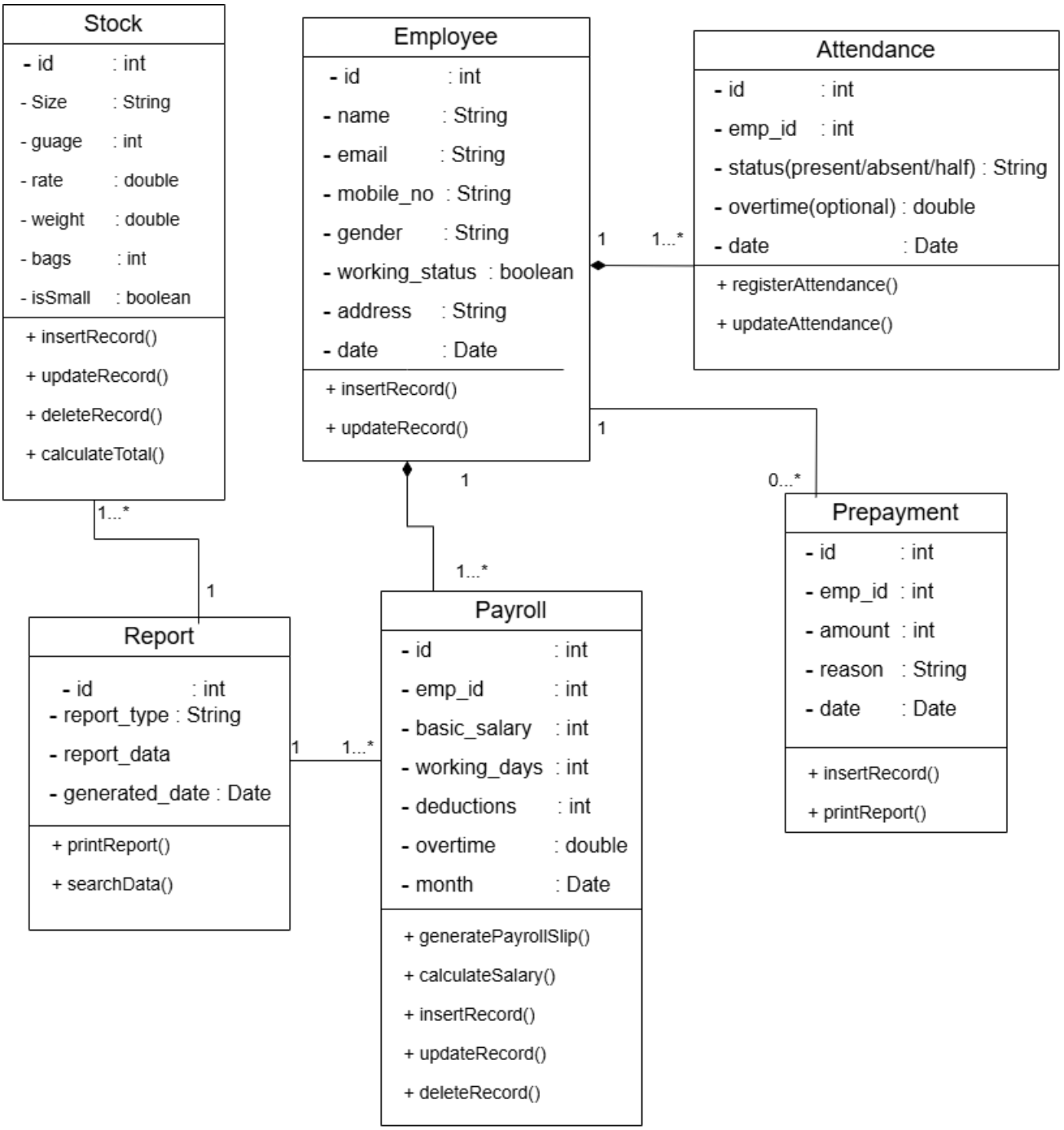


First level DFD

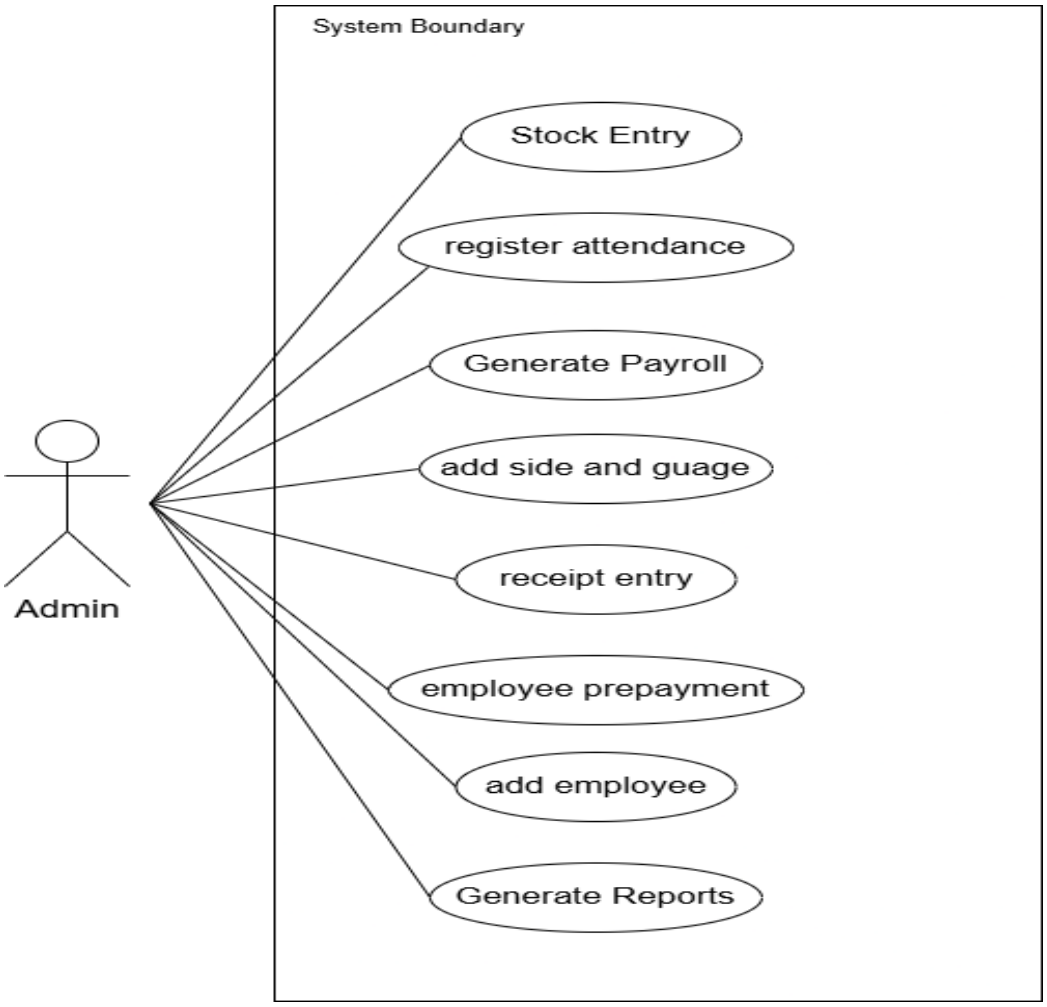


3.3 UML Diagrams

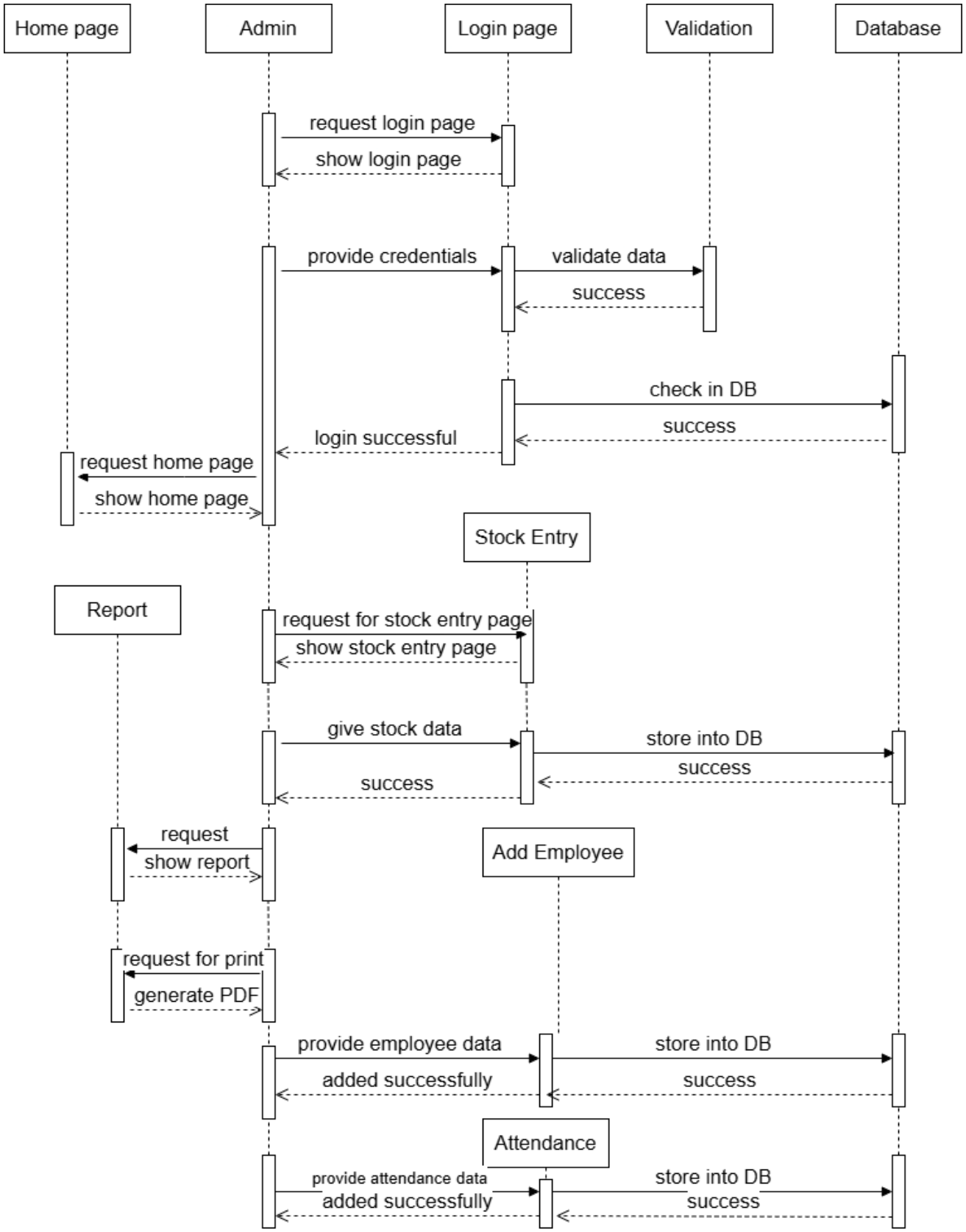
1. Class Diagram



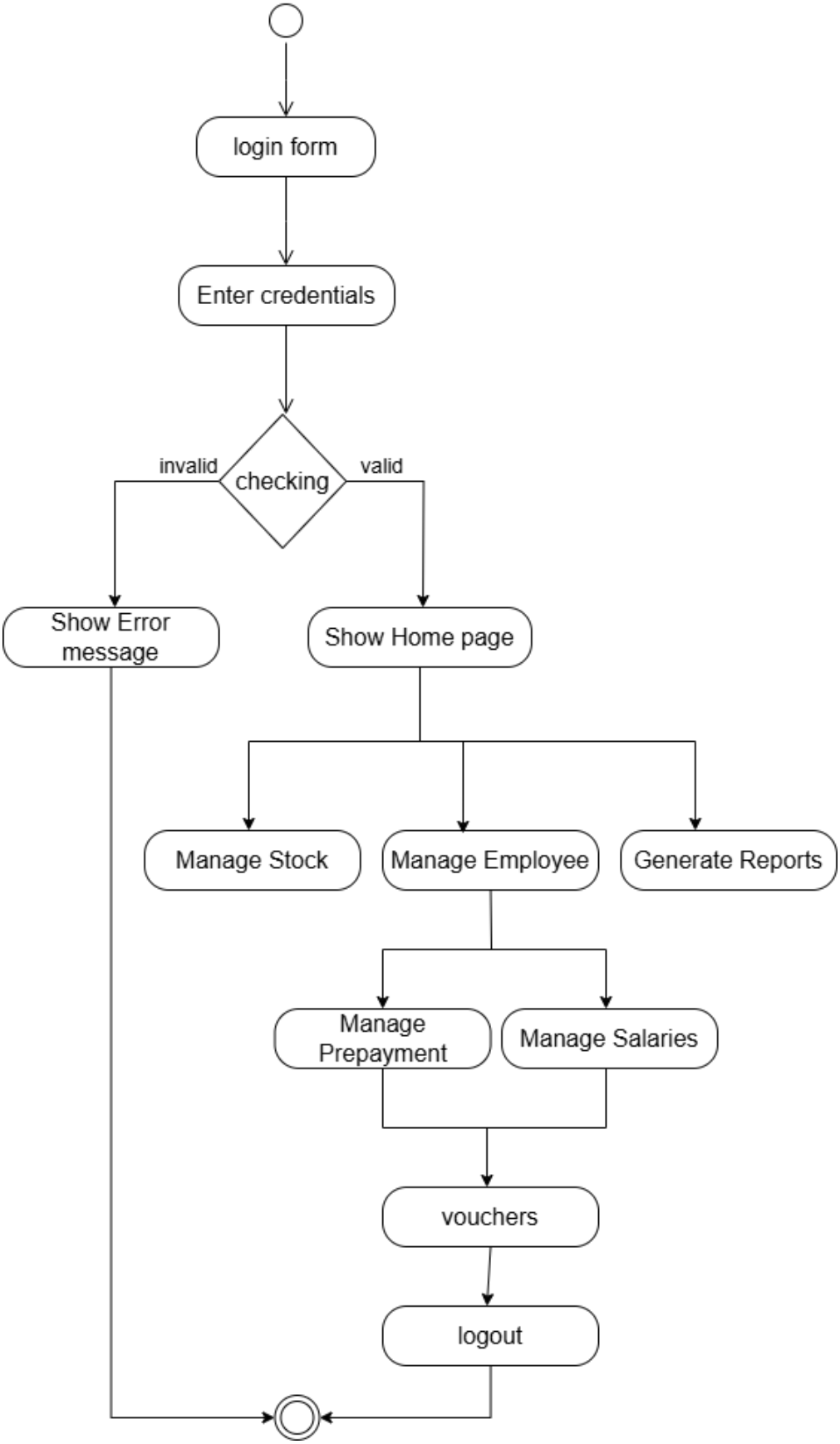
2. Use Case Diagram



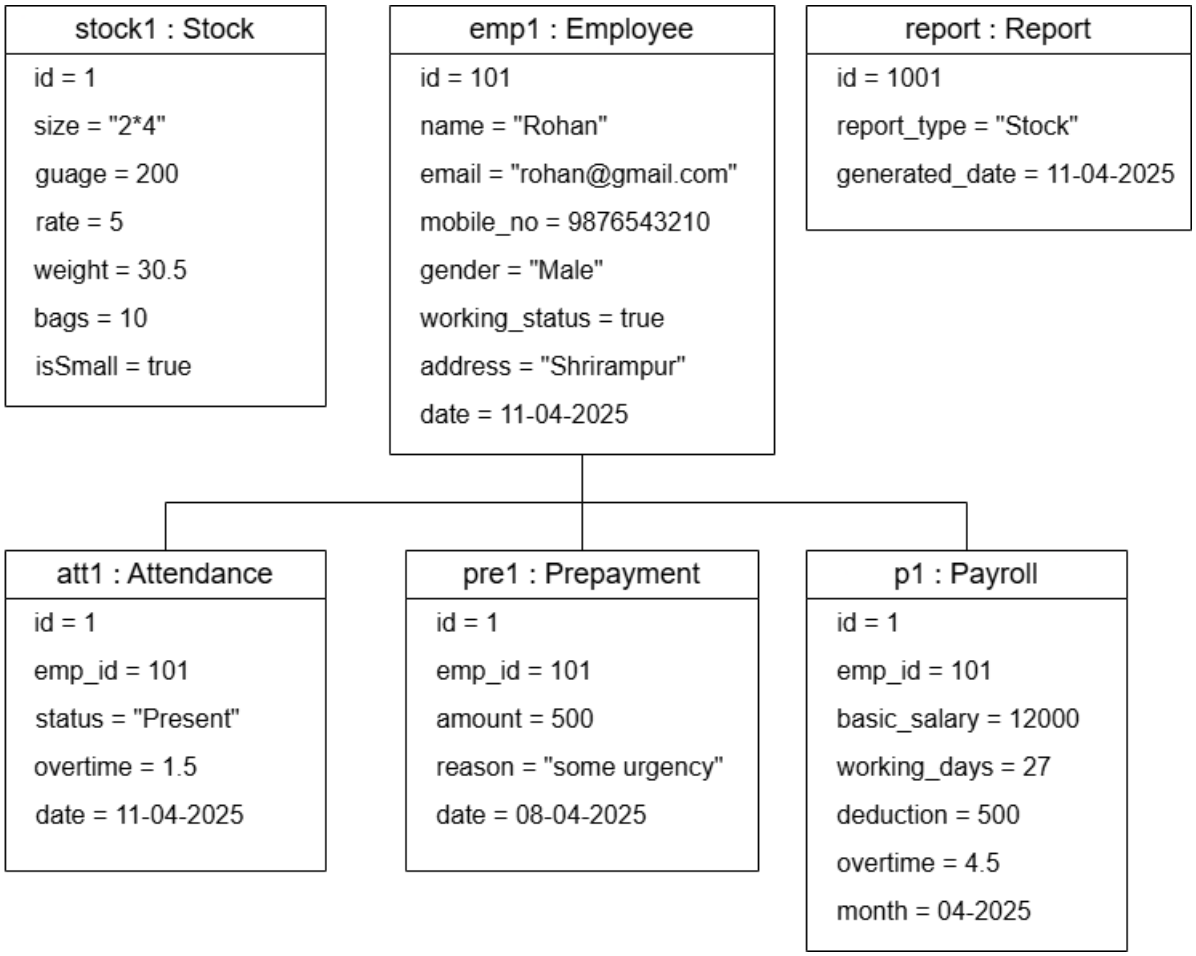
3. Sequence Diagram



4. Activity Diagram



5. Object Diagram



3.3 Data Model

Table name 1. users

table_name name	column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
users	id	1	[null]	NO	bigint
users	username	2	[null]	NO	text
users	password	3	[null]	NO	text
users	email	4	[null]	YES	text
users	phone_no	5	[null]	NO	text
users	join_date	6	[null]	NO	date
users	address	7	[null]	NO	text
users	isVerify	8	false	NO	boolean

Table name 2. stocks

column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('stocks_id_seq'::regclass)	NO	bigint
day_wise_entry_no	2	[null]	YES	integer
size	3	[null]	YES	text
size_type	4	[null]	YES	text
rate	5	[null]	YES	integer
bag	6	[null]	YES	integer
weight	7	[null]	YES	double precision
guage	8	[null]	YES	integer
is_small	9	false	YES	text
entry_month	10	[null]	YES	date
created_at	11	[null]	YES	timestamp without time zone
updated_at	12	[null]	YES	timestamp without time zone
user_id	13	[null]	YES	bigint

Table name 3. attendances

column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('attendance_id_seq'::regclass)	NO	bigint
status	2	[null]	YES	text
created_at	3	[null]	YES	timestamp without time zone
updated_at	4	[null]	YES	timestamp without time zone
attendance_date	5	[null]	YES	date
user_id	6	101	YES	bigint

Table name: 4. vouchers

column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('vouchers_id_seq'::regclass)	NO	bigint
status	2	[null]	NO	text
payment_state	3	[null]	NO	text
description	4	[null]	NO	text
amount	5	[null]	NO	numeric
user_id	6	[null]	NO	bigint
emp_id	7	[null]	YES	bigint
date	8	[null]	YES	date
created_at	9	[null]	YES	timestamp without time zone
updated_at	10	[null]	YES	timestamp without time zone
receipt_voucher_no	11	[null]	YES	integer
salaries_id	12	[null]	YES	bigint

Table name 5. salaries

column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('salaries_id_seq'::regclass)	NO	bigint
emp_id	2	[null]	NO	bigint
basic_salary	3	[null]	NO	numeric
advance_amt	4	[null]	YES	numeric
created_at	5	[null]	YES	timestamp without time zone
updated_at	6	[null]	YES	timestamp without time zone
payment_month	7	[null]	YES	date
deposit_amount	8	[null]	YES	integer
overtime_hours	9	[null]	YES	double precision
rate_of_per_hour	10	[null]	YES	integer

Table name 6. prepayments

column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('prepayments_id_seq'::regclass)	NO	integer
emp_id	2	[null]	YES	integer
amount	3	[null]	YES	integer
date	4	[null]	YES	date
emp_wise_entry_id	5	[null]	YES	integer
description	6	'-'::text	YES	text

Table name 7. late_arrivals

column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('late_arrivals_id_seq'::regclass)	NO	bigint
emp_id	2	[null]	YES	integer
no_of_hours	3	[null]	YES	double precision
date	4	[null]	YES	date
time	5	[null]	YES	text
created_at	6	CURRENT_TIMESTAMP	YES	timestamp without time zone
updated_at	7	CURRENT_TIMESTAMP	YES	timestamp without time zone
description	8	'-'::text	YES	text
emp_wise_entry_no	9	[null]	YES	integer

Table name 8. guage_rate

column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('guage_rate_id_seq'::regclass)	NO	integer
rate	2	[null]	YES	double precision
guage	3	[null]	YES	double precision
user_id	4	[null]	YES	bigint
size_id	5	[null]	YES	bigint

Table name 9. employees

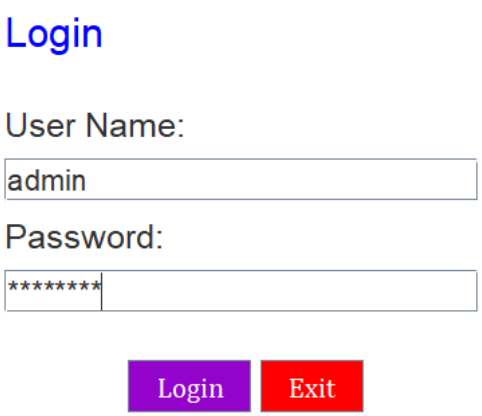
column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('employees_id'::regclass)	NO	bigint
name	2	[null]	NO	text
address	3	[null]	NO	text
mobile_no	4	[null]	NO	text
email	5	[null]	YES	text
gender	6	[null]	NO	text
created_at	7	[null]	NO	timestamp without time zone
updated_at	8	[null]	YES	timestamp without time zone
user_id	9	[null]	YES	bigint
working_status	10	[null]	YES	boolean

Table name 10. sizes

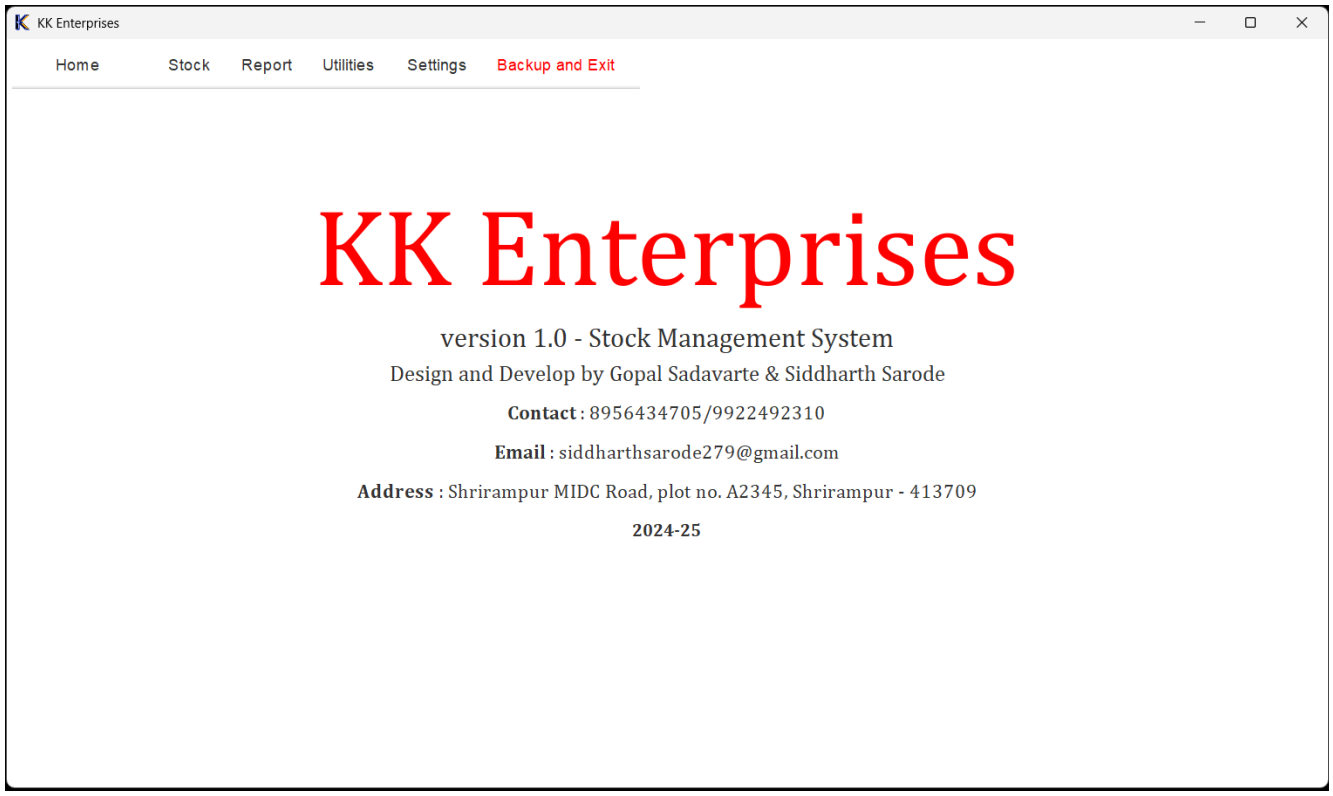
column_name name	ordinal_position integer	column_default character varying	is_nullable character varying (3)	data_type character varying
id	1	nextval('sizes_id_seq'::regclass)	NO	integer
size	2	[null]	YES	character varying
user_id	3	[null]	YES	bigint

3.4 User Interfaces

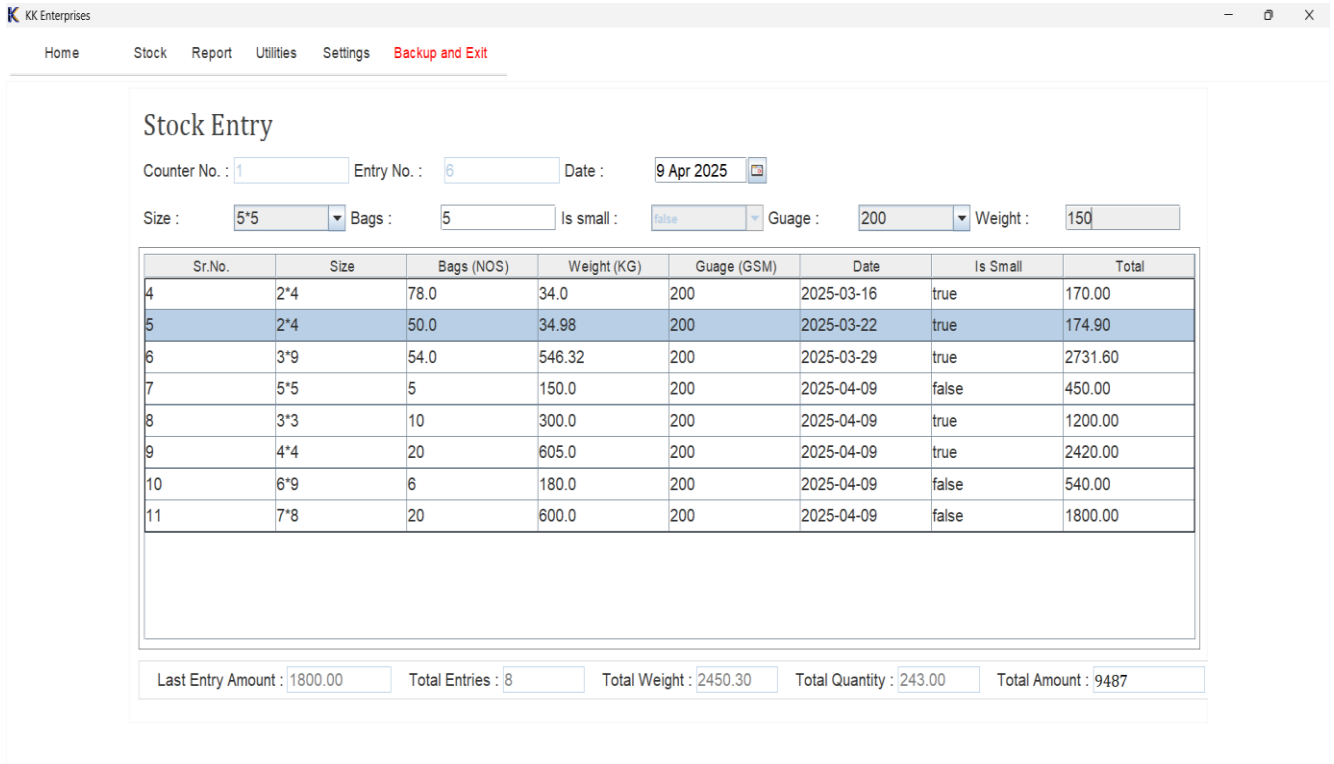
1. Admin login



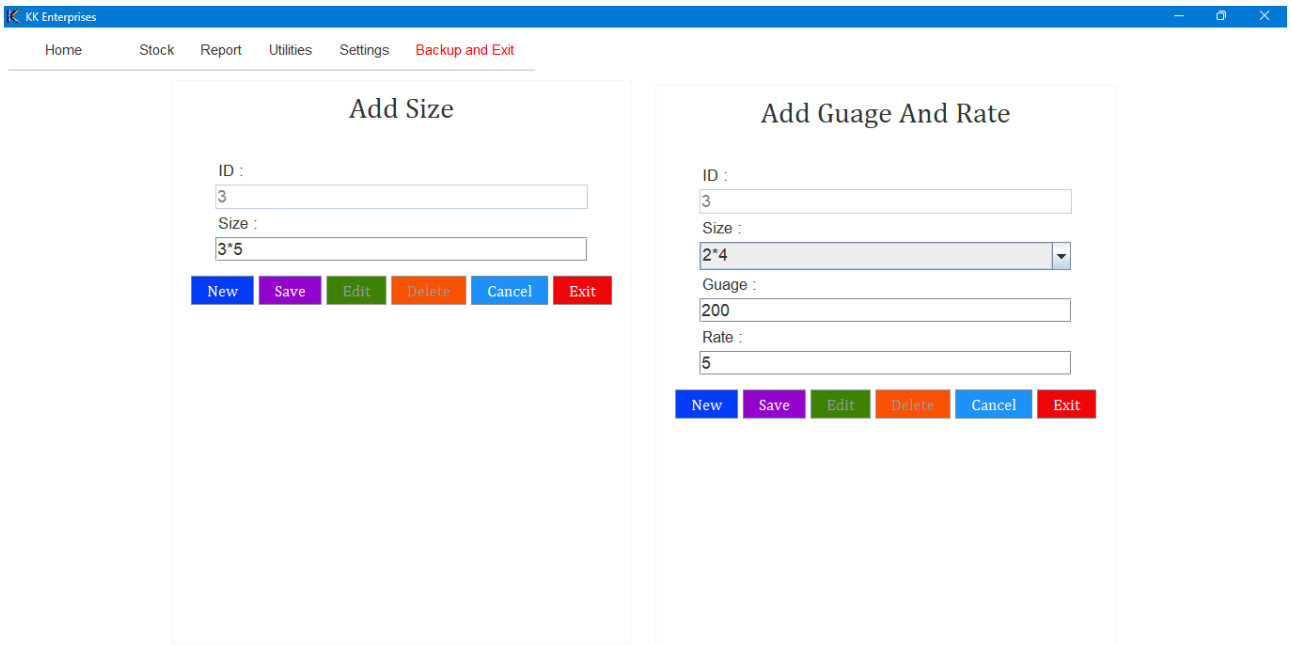
2. Home Screen



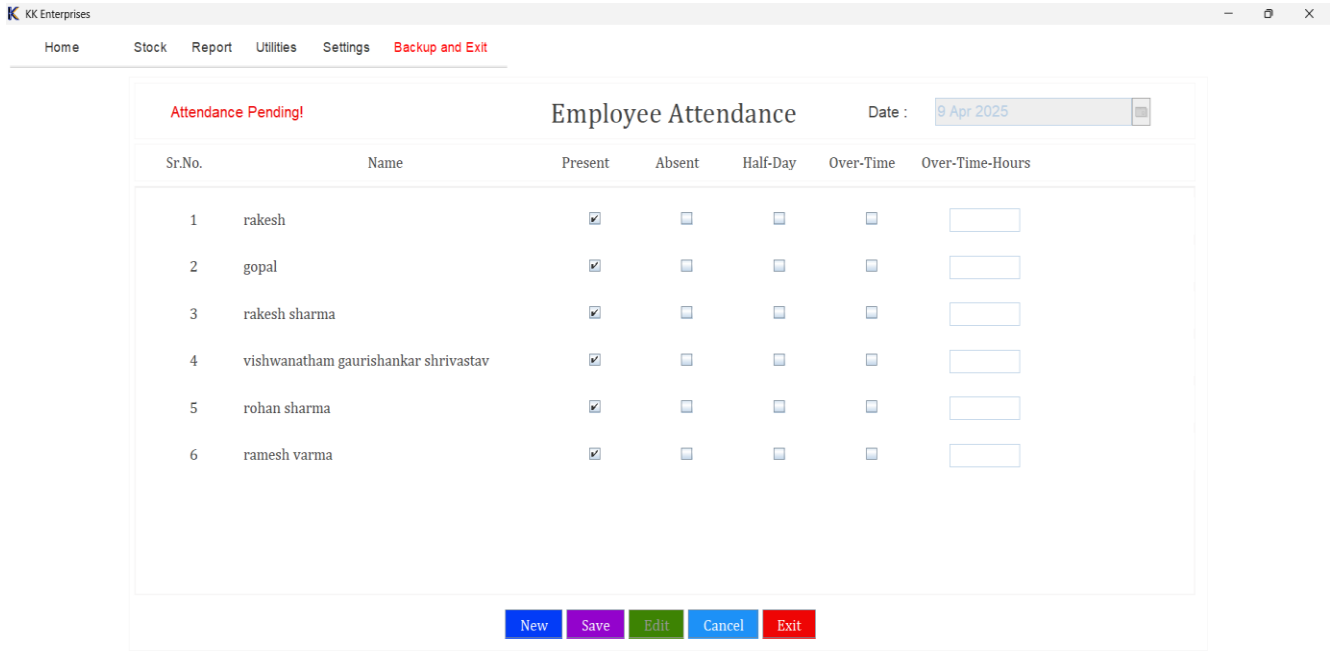
3. Stock entry screen



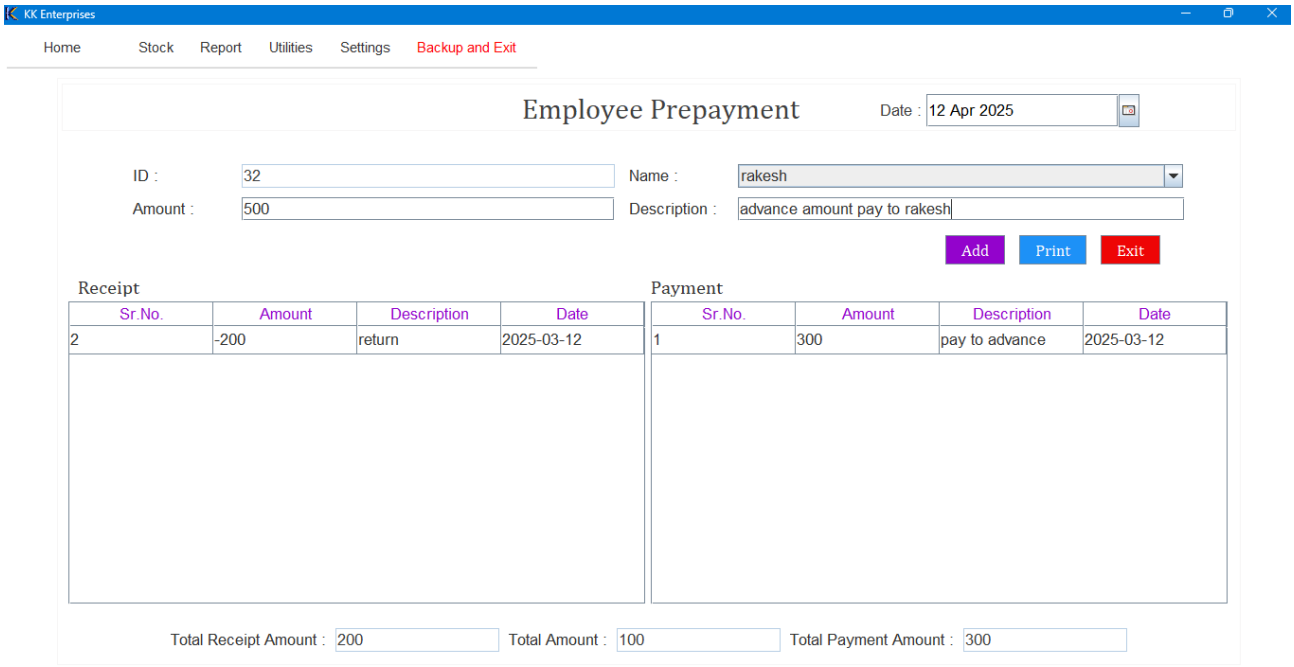
4. Sizes entry screen



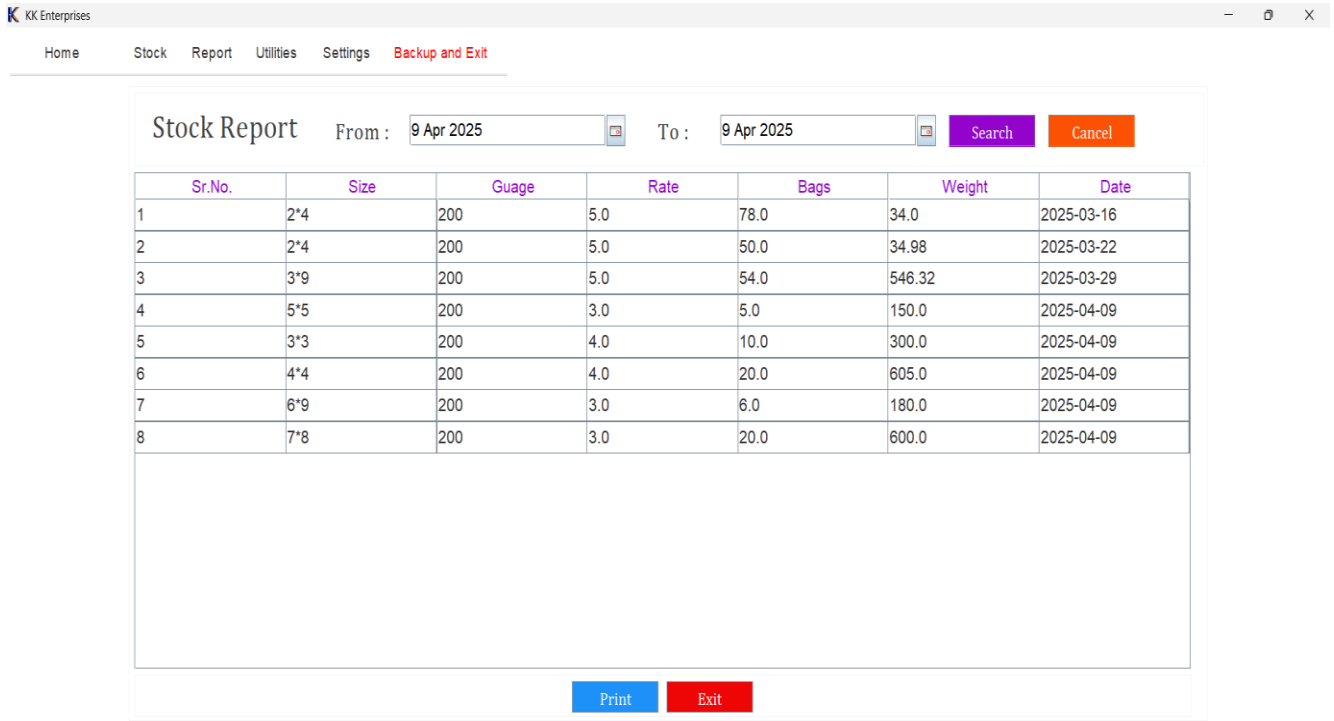
5. Employees attendance screen



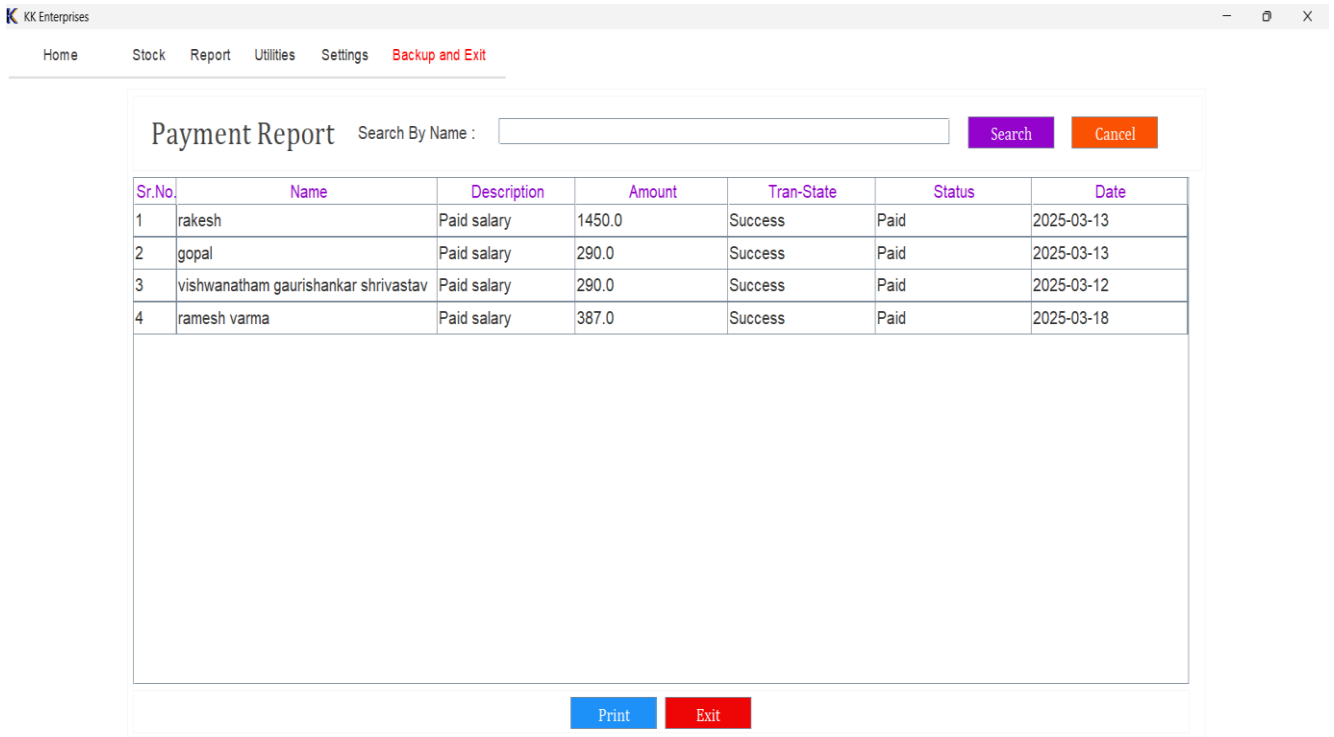
6. Employee Pre-payments screen



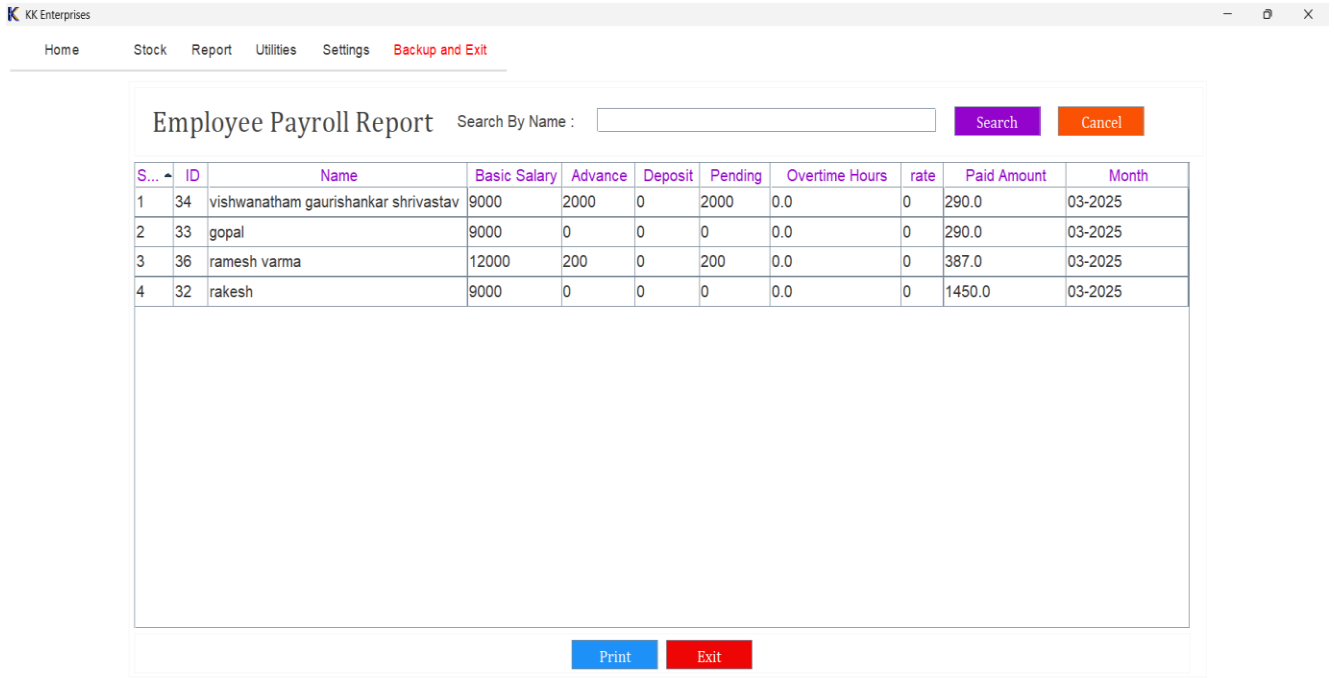
7. Stock report screen



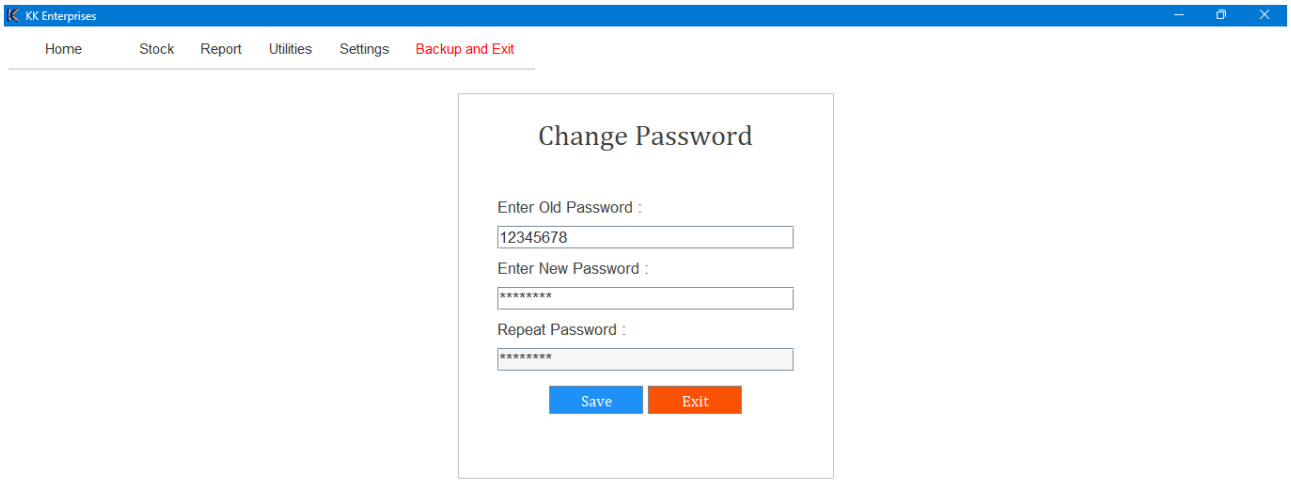
8. Payment Report Screen



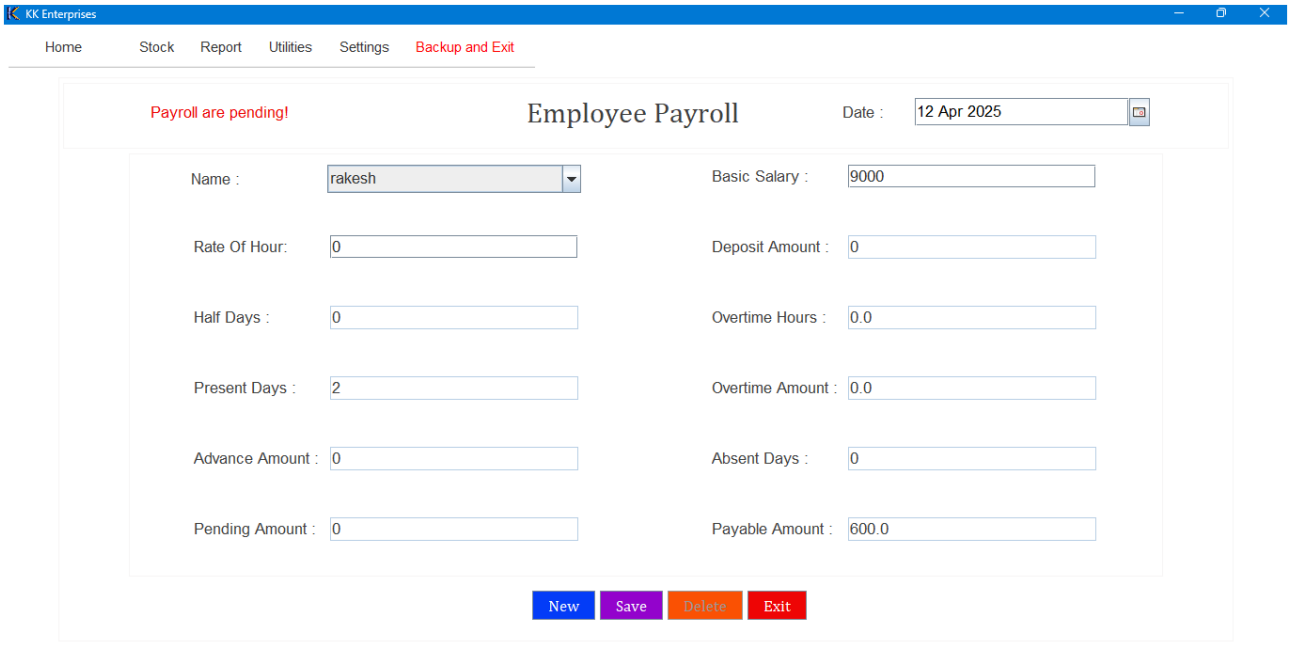
9. Employees Payrolls Screen



10. Change Password screen



11. Payroll screen



4. Implementation Details

4.1 Software/Hardware Specification:

- Device: Acer DESKTOP-C7KL3CM
- Processor: 11th Gen Intel(R) Core(TM) i3 gen.
- Input Devices: Basic keyboard and touch pad or mouse.
- Output Devices: Standard colour monitor
- Operating System: 64-bit Windows 11
- Front-End: Core Java, AWT
- Back-End: Advanced Java, Postgresql database

Needed Configuration

- RAM: At least 2GB
- Windows or iOS latest version will make smooth experience.
- **Hardware**: hard-disk should be greater than 200MB
- Java Runtime Environment (JRE)
- Postgresql database setup

5. Outputs and Reports Testing

5.1 Outputs

- Stock levels are updated correctly after sales and purchases.
- Employee salary reports reflect accurate calculations based on attendance, bonuses, and deductions.
- Date-wise and category-wise reports filter data correctly.
- Exported reports (PDF, CSV) are readable and properly formatted.

5.2 Data validation

- Preventing negative stock quantities.
- Ensuring correct date formats and numeric values in fields.
- Avoiding duplicate entries for products or employees.
- Validating salary input fields before processing payroll.

5.3 Test cases

Test case for admin login

TC No.	Description	Test steps	Expected Result	Pass / Fail
1	Verify valid admin credentials login successfully	1. Enter valid admin Username 2. Enter valid Password 3. Click on the login button	Home screen is displayed	Pass
2	Verify login with invalid admin username	1. If enter invalid admin Username 2. Enter valid Password 3. Click on the login button	Error message indicating invalid username is displayed	Pass
3	Verify login with invalid admin password	1. Enter valid admin Username 2. If enter invalid password 3. Click on the login button	Error message indicating invalid password is displayed	Pass

6. Conclusion and Recommendations

6.1 Conclusion

The Stock Management System successfully automates the core functions of stock tracking, report generation, and employee salary management. By replacing the manual, error-prone processes with a computerized solution, the system significantly improves accuracy, reduces workload, and enhances overall efficiency.

Developed using Core Java for the front-end and PostgreSQL for the database, the system ensures secure data handling and user-friendly interaction. It minimizes paperwork and streamlines complex calculations, making it a reliable solution for small to medium-sized businesses seeking efficient stock and payroll management.

6.2 Recommendations

1. Multi-user Access – Add role-based access for admin, manager, and staff-level users.
2. Real-time Notifications – Alert users for low stock levels or pending salaries.
3. Cloud Backup – To ensure data recovery and access from anywhere.
4. Mobile App Support – Extend the system's accessibility by developing a mobile version.
5. Data Analytics and Forecasting – Use charts and graphs to visualize stock trends and predict future requirements.

7. Future Scope

1. Cloud-Based Deployment

Hosting the application on the cloud (e.g., AWS, Azure) to enable real-time access from multiple locations and devices.

2. Mobile Application Integration

Developing Android and iOS applications to allow users to manage stocks and view reports on the go.

3. AI-Powered Analytics

Implementing AI algorithms to analyze stock patterns, predict future demands, and optimize stock levels accordingly.

4. Third-Party Integration

Enabling integration with third-party software like accounting tools (e.g., Tally, QuickBooks) and e-commerce platforms.

5. Enhanced Security Feature

Adding encryption, two-factor authentication (2FA), and role-based access control to further protect sensitive business data.

6. Multi-Language Support

Offering the system in multiple languages to cater to users from different regions and improve user adoption.

7. Custom Report Builder

Allowing users to create their own report formats with selected data fields and filters.

8. SMS/Email Notification System

Automating alerts for low inventory, pending payments, and salary disbursements through SMS or email.

8. Bibliography and References

8.1 Bibliography

1. Programming with Java by E. Balagurusamy
2. Lecture notes

8.2 References

1. https://docs.oracle.com/javase/8/docs/ — Official Java Documentation
2. https://www.postgresql.org/docs/ — PostgreSQL Official Documentation