BANK MANAGEMENT SYSTEM

Submitted By: Prathmesh Sanap, Tanishq Bhitkar

Under the Guidance of: Kajal Moghe Mam

Department: Computer Engineering

College Name: Dr.D.Y.Patil Technical Campus Talegaon Varale

Date: [05/07/2025]

📜 Certificate Page

This is to certify that the project titled "Bank Management System" is a bonafide work carried out by **Prathmesh Sanap** and **Tanishq Bhitkar** under the guidance of **Kajal Moghe Mam** , in partial fulfillment of the requirements for the subject in the Department of Computer Engineering, **Dr.D.Y.Patil Technical Campus Talegaon Varale.**

🧾 Student Declaration

We hereby declare that the project titled "Bank Management System" submitted in partial fulfillment of the requirements for the subject is our original work and has not been submitted previously elsewhere.

Prathmesh Sanap

Tanishq Bhitkar

🙏 Acknowledgment

We would like to express our sincere gratitude to our guide **KAJAL MOGHE MAM** for their valuable guidance, encouragement, and continuous support throughout this project. We also extend our thanks to the Computer Engineering Department and our classmates for their suggestions and support.

🧠 Abstract

The Bank Management System is a Java-based mini project designed to simulate the basic operations of a banking system through a command-line interface. This project allows users to register with a full name, email, and password. Using JDBC, the credentials are securely stored in a MySQL database. The application validates if the user already exists to prevent duplicate entries. It demonstrates how Java can be effectively integrated with MySQL databases using JDBC for real-world applications. The project focuses on simplicity, modular design, and practical learning of CRUD operations and input handling.

📚 Table of Contents

🔍 Introduction

Banking operations are an essential part of daily life. This project simulates the basic functionality of a banking system using Java and JDBC. The motivation behind this project is to gain hands-on experience in managing databases using JDBC and understanding how backend systems interact with front-end inputs.

🎯 Objective

- Understand and implement JDBC connectivity
- Perform CRUD operations on a relational database
- Handle user input securely using Scanner
- Design modular and maintainable code structure

💻 System Requirements

Hardware:

- RAM: 4GB minimum
- Processor: Intel i3 or higher

Software:

- JDK 17 or higher
- IntelliJ IDEA Community Edition
- MySQL Server 8.x
- MySQL Connector/J (JDBC Driver)

🧰 Technology Stack

Java	Programming Language
JDBC	Database Connectivity
MySQL	Relational Database System
IntelliJ IDEA	Development Environment

🏗️ System Architecture

Flow: User Input → Java Application → JDBC Driver → MySQL Database

📊 Use Case Diagram

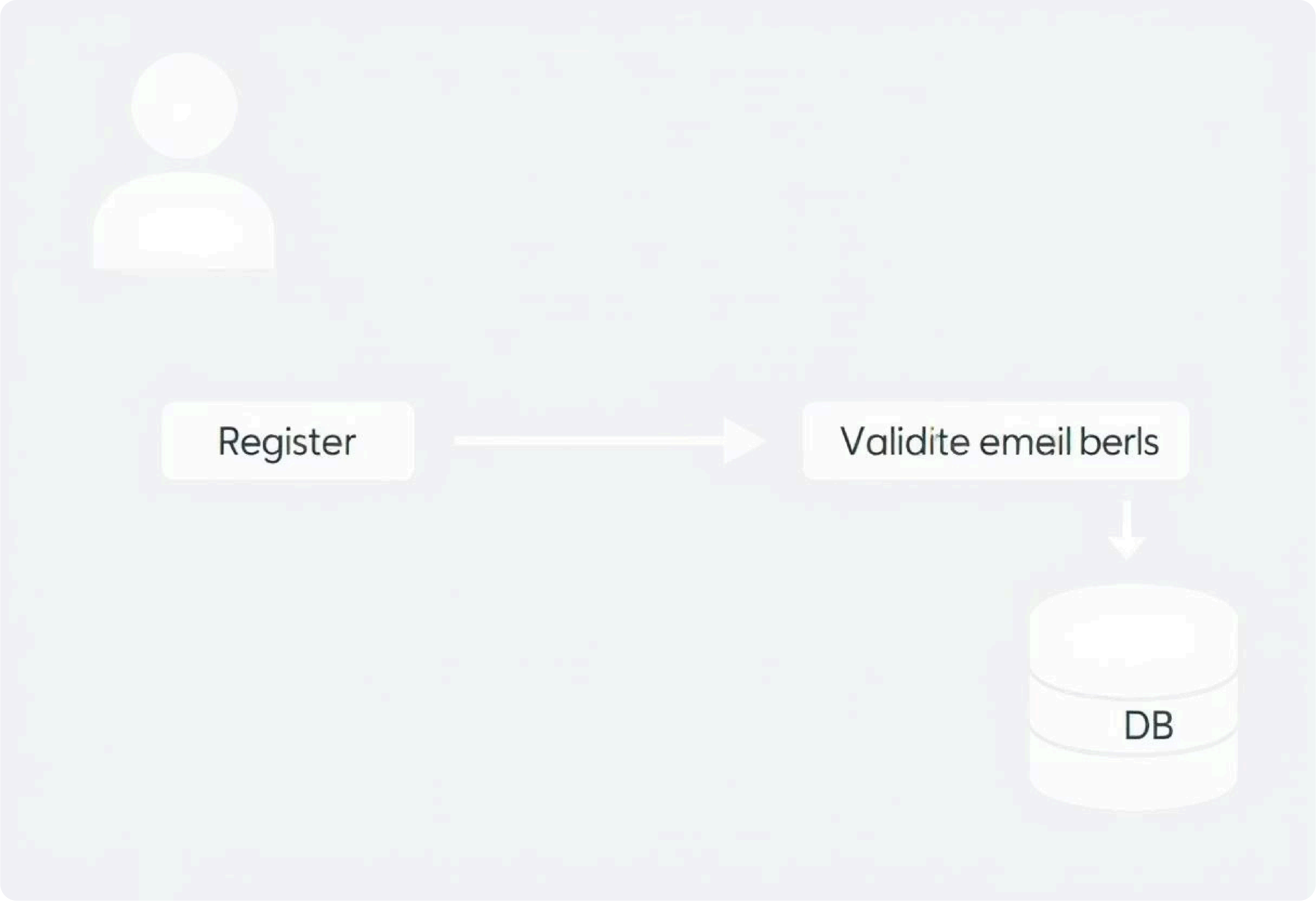
Actors:

- User

Use Cases:

- Register
- Validate Email
- Store to DB

📌 [Insert Use Case Diagram Image Here]



🧩 Modules & Features

bankingapp.java

- Loads JDBC Driver
- Connects to MySQL
- Calls User module

User.java

- Accepts input (name, email, password)
- Validates if email already exists
- Prepares for future login and transaction features

🗃️ Database Design

Table: users

id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(100)	NOT NULL
email	VARCHAR(100)	UNIQUE, NOT NULL
password	VARCHAR(100)	NOT NULL

SQL Query:

```
CREATE TABLE users (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100),
email VARCHAR(100) UNIQUE,
password VARCHAR(100)
);
```

🛠️ Implementation

Flow:

1. Scanner collects input
2. JDBC checks if email exists
3. If not, data is inserted into DB

Code Snippet:

```
Scanner scanner = new Scanner(System.in);
System.out.print("Email: ");
String email = scanner.nextLine();
PreparedStatement pst = connection.prepareStatement("SELECT * FROM users WHERE email = ?");
pst.setString(1, email);
ResultSet rs = pst.executeQuery();
```

🧪 Testing

TC1	Register new user	Success
TC2	Register existing user	Message: "User already exists"

📸 **[Attach IntelliJ terminal or MySQL Workbench screenshot here]**

⚠️ Limitations

- No password encryption
- No GUI interface
- Only registration module implemented

🚀 Future Enhancements

- Encrypt passwords using SHA-256
- Add login and session management
- Introduce GUI using Swing or JavaFX
- Add account and transaction modules

✅ Conclusion

This project provided practical knowledge of Java's interaction with databases using JDBC. It enhanced understanding of modular programming, user input validation, and backend development. It is a strong foundation for scalable banking software.

📚 References

- https://youtu.be/hlGoQC332VM?si=x0YE5OoDZkGki9qs :- MySql Java MySQL
- httpshttps://youtu.be/TcJZQvDE1ow?si=ieYZbydBtZW_47pc https://youtu.be/TcJZQvDE1ow?si=ieYZbydBtZW_47pc :- JDBC
- Google AI Studio