# 

# A

**Progress Report**

**ATM Simulator**

Submitted to

**COLLEGE NAME**

For

degree programme By

# Name of the Student

(Calibri 20 size, Bold)

Roll No : -----------------

**College LOGO**

# Under the supervision of

(Signature) (signature)

# Name of Mentor/Supervisor Student Name

Designation Roll no



***Introduction- Organization***

# Future Finders is a ground-breaking platform that develops young Indian talent that is motivated to advance and forge successful careers in IT. We provide a variety of courses to help you launch your career and locate the employer that will assist you as you rise to the top. Our expertise with the latest tools and techniques, and the experience of our professional experts help us deliver high-end services to our esteemed clients. Future Finders educates students and developers about the most recent technologies that are now popular. We are working on distributing informational know-how and offering clients specialized services following global best practices. At Future Finders, we recognize your unrivalled skill and help you realize your creative ideas. Our sole goal is to provide students with cutting-edge practical skills that will enable them to swiftly and effectively adapt to the constantly evolving technologies found in the business world. At Future Finders, our goal is to raise educational standards via innovation in both quality and practical knowledge.



**MISSION**

* At Future Finders, our goal is to improve quality and practical skills while raising the bar for education. Future Finders' principal objective is to bridge the knowledge gap between what is being taught in schools and what the industry needs.

**VISION**

* Future Finders' vision is to provide students with cutting-edge practical skills so they can easily handle and swiftly acclimate to the constantly evolving technology in the corporate world.

**PHILOSOPHY**

# To impart hardcore practical quality training among students/developers about latest technologies trending today.

To share knowledge of information security and create awareness in the market. The solution to clients' as per the International standard practices and governance.

# To support good business practices through continual employee training and education

To equip a local team with a strong knowledge of international best practices and international expert support so as to provide practical advisories in the best interests of our clients

* JAVA
* FULL STACK
* C LANGUAGE
* C++ LANGUAGE
* CYBER SECURITY
* MERN STACK
* PYTHON
* DIGITAL MARKETING
* PHP
* NETWORKING
* ANDROID
* IOS
* .NET
* ORADE DBA
* SOFTWARE TESTING
* ETHICAL HACKING
* MACHINE LEARNING
* BIG DATA HADOOP
* NODE JS
* ANGULAR JS
* CLOUD COMPUTING
* AUTO CAD
* NETWORKING
* ROBOTICS
* VLSI – VHDL
* PCB
* MATLAB
* EMBEDDED SYSTEM
* IOT
* PLC / SCADE
* ARDUINO
  + NX- CAM
  + AUTOCAD ME
  + SOLID WORKS
  + CATIA
  + CREO
  + CNC PROGRAMMING
  + ANSYS
    - MARKETING
    - DIGITAL MARKETING
    - FINANCE
    - HR MANAGEMENT
  + AUTOCAD CE
  + PRIMA VERA
  + MX ROAD
  + 3DS MAX
  + REVIT ARCHITECTURE
  + STAAD PRO

CSE & IT

ECE & EE

MECHANICAL

CIVIL

MANAGEMENT

 Experience

 Goal Oriented

 A Streamlined / Quality-Driven Process  Talented Designers & Expert Developers

 Our Websites & E-marketing Platforms are Easy to Manage  We Are Dedicated to Our Client’s Success

## We focus on imparting practical skills to the trainees & not just theoretical knowledge. The courses are designed in this way at FUTURE FINDERS correspond to the standards of the corporate divisions and industries. Only through the acquisition of practical skills you can handle the everlasting technologies that venture out in real-time situations**.**

* At **FUTURE FINDERS,** Future Finders has committed staff members with organised learning curricula that can assist you in beginning your career in the most cutting-edge and successful industries.

* **Quality of the Product:** Our software service sector has been maintaining the highest international standards of quality.

## * **Live projects:** Working on active projects and working at a job aid in experiencing growth. In keeping with this, we assist you in developing greater self-assurance, which helps you succeed as quickly as feasible.

* **Global certification:** The courses that Future Finders offers make sure that you are certified and eligible to grab the best opportunities in your career.

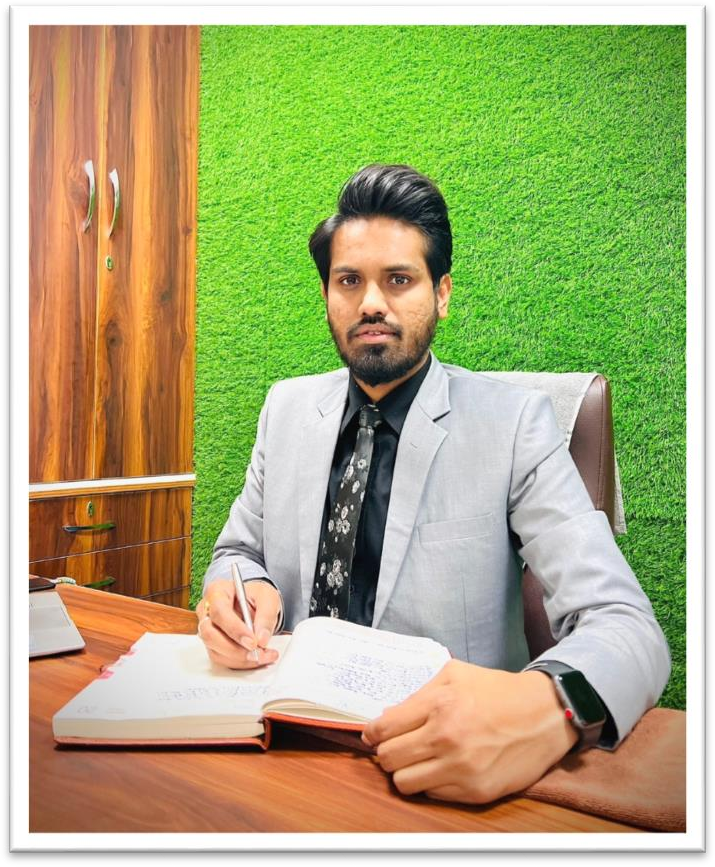
## * **Partnership:** FUTURE FINDERS, considers every client a partner. From the initial stages, you are closely involved into the procedure of technical classification, development, and testing.

* **Jobs & Career Prospective:** Future Finders views education and learning as investments in oneself. Consequently, the paths we have set forth ensure that your investment in us is a success



* In addition to a panel of eminent consultants and advisors, we have a dedicated pool of trained Developers and Trainer, investigators, working under the guidance of professional managers. **“A Ship is as good as the crew who sail her**.**”** Our Technical team of professionals handing, designing & delivering of projects has a strong presence in the North India & the US. Our engineers are already working on the latest technologies like **I-Phone & Android** Applications, **Robotics**, **VLSI-VHDL**, Embedded System, Networking and **Cloud computing.** Some of the key professionals and advisors are listed

### Mr. Bonish Singla: (Director)



* He is the backbone of FUTURE FINDERS, manage the company’s day to day affairs and a man with more than 9 years rich practical experience who believes in taking up new ventures and projects. He has been awarded many times for his exemplary work in process improvement for IT Service Delivery Domains. MASTERS in Computer applications and Certified from CU Certification. Holds total of 9 Years of rich experience including 5 Years in Information Security Implementation, Maintenance and Auditing and initial over 4 years of experience in Project Management, Client Relationship Management and Server, Desktop, and IT Service Delivery web designing.

### 

### Miss. Harjit Kaur :(Branch Manager)



* She has more than 5 years solid industrial experience in software companies and she is very innovative in her technical approach. She has completed her masters in MBA. She takes all the responsibilities and maintains staff by recruiting , selecting , orienting , and training employees and Accomplishes staff results by communicating job expectations , planning ,monitoring , and appraising job results.

### Miss. Harsimran: (HR)



* Human resources (HR) are the division of a business responsible for finding, recruiting, screening, and training job applicants. MBA in HR and marketing from (CU) Total of 5+ Years of rich experience HR departments also handle employee compensation, benefits, and terminations. HR departments must keep up to date with laws that can affect the company and its employees. She also assists with payroll management so employees receive their paychecks on time.

### 

### Miss. Isha Bala: (Technical Head)



* A technical lead, or tech lead, oversees the technical aspects of a software team. M. tech and diploma in (CSE) . She helps making architectural and design decisions, guiding team members, and supervising system modifications. Identify potential risk and forming contingency plan as fast as possible. Efficiently liaise with the team members, clients , and also the management .

### Miss. Nihirika: (Head Counselor)

### 

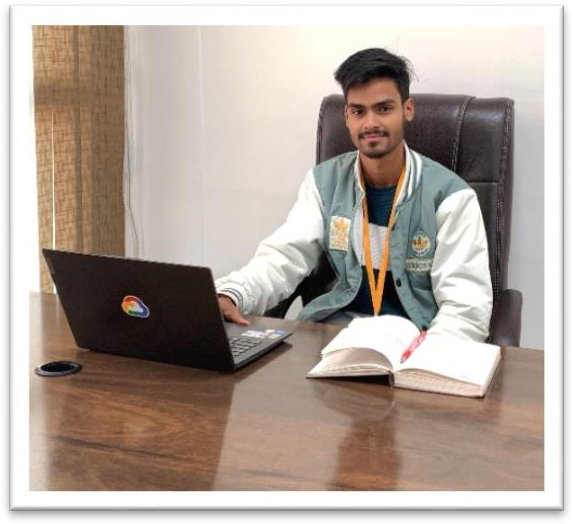
* Head Counselor, under the general direction of the Head - Sales and Marketing, provides leadership and direction to the Counseling Department and assumes responsibilities in developing, implementing, and evaluating the Company counseling and guidance program that includes academic, career, personal/social development. She completed her degree in B .Tech

### 

### Mr. Jaspal Singh: (Civil &Mechanical Head)

* He is leader of the team of civil, mechanical, and electrical engineers and responsible for the planning and analysis of the aspect of the construction that involves mechanical works . He has more than 37 years of experience in industrial field. He is providing services as a technical trainer for more than 8 years. He did his B. Tech in Mechanical Engineering from PEC (Punjab Engineering College). He has been awarded many times for his brilliant services.

### Mr. Chetan Kalra: (Digital Marketing Head)



* B.tech (CSE) – IKG-PTU, Experienced digital marketing manager with extensive experience building, maintaining, and running successful digital marketing campaigns from past 4 years. Bringing forth broad marketing knowledge, coupled with focused campaign experience. Adept at creating and implementing client- centered, successful campaigns, aimed at improving brand awareness and presence. Collaborative and creative manager accomplished at managing digital marketing presence content. Experienced in leading teams of marketing professionals to meet and exceed digital marketing goals.



Mr. Shivam: (Java Developer)

### 

Having 5+ years of experience in analysis, design, development, testing & implementation of complex software applications. B. Tech in CSE (CGC College) Experience and involvement in designing, implementing, and evaluating end-to-end systems using several Java frameworks and technologies like J2EE.

### Miss. Archana: (Full Stack Developer)



* Full Stack Developer with 6+ years of hands-on experience designing, developing, and implementing applications and solutions using a range of technologies and programming languages. B.tech (CSE) PU Certification Seeking to leverage broad development experience and hands-on technical

### 



Plot NO F-465, K&B Tower, Second Floor, Phase 8B, Mohali I (+91) 98559-08009 I [www.futurefinders.in](http://www.futurefinders.in/)

### Mr. Ashwani: (Automation Executive)

### 

* (EE) – CGC College Automation engineer with 4+ years of experience in a variety of industries. Passionate for developing and implementing process improvements through the use of robotics, PLCs, and HMIs. Demonstrated ability to lead cross-functional teams in the design, development, and deployment of manufacturing and process automation solutions.

### Mr. Ayush: (PHP Developer)



* PHP developer to manage our back-end services and ensure a seamless interchange of data between the server and our users. Bachelor's degree in computer programming, development and Certified from CU Certification PHP developer, responsible for developing and coding all server-side logic and required to maintain the central database and respond to requests from front-end developers

**Introduction of Tasks**

The development of an ATM Simulator in Java represents a significant undertaking in the realm of modern banking and financial services. This progress report provides an overview of the ongoing development, challenges encountered, milestones achieved, and future prospects of the ATM Simulator project. Rooted in the Java programming language, the project aims to replicate the functionalities of a real-world ATM system, offering a platform for experimentation, learning, and refinement.

The development process involves integrating various components, including user interface design, transaction processing logic, security protocols, and error handling mechanisms. Each aspect demands meticulous attention to detail and adherence to best practices in software engineering. Throughout development, the team has emphasized principles of modularity, scalability, and maintainability, adhering to object-oriented design principles and leveraging established frameworks and libraries.

The project team remains committed to delivering a robust, feature-rich ATM Simulator embodying reliability, security, and user-centric design principles. Through collaboration, feedback, and rigorous testing practices, the team aims to realize the full potential of the project.

The development process is driven by the collective ambition to innovate, educate, and empower, creating a platform that simulates real-world banking experiences and fosters a deeper understanding of software development principles and practices.

**Short Briefing: Activities performed till previous evaluation**

1. **Requirement Analysis:**
   * Conducted comprehensive analysis to identify the functional and non-functional requirements of the ATM Simulator.
   * Defined user stories and use cases to capture the system's behavior from the user's perspective.
2. **Design Phase:**
   * Designed the system architecture outlining the components and their interactions.
   * Created class diagrams, sequence diagrams, and other UML diagrams to visualize the system's structure and behavior.
   * Established design patterns and coding standards to ensure consistency and maintainability.
3. **Implementation:**
   * Developed core functionalities such as user authentication, transaction processing, and error handling.
   * Implemented a modular and scalable codebase adhering to object-oriented principles.
   * Integrated user interface components to provide a seamless and intuitive experience for users.
   * Incorporated logging mechanisms to track transactional data for auditing and analysis purposes.
4. **Testing:**
   * Conducted unit tests to validate individual components and ensure their correctness.
   * Performed integration testing to assess the interactions between different modules.
   * Identified and resolved bugs and issues through rigorous testing procedures.
   * Conducted user acceptance testing (UAT) to gather feedback and validate the system against user expectations.
5. **Documentation:**
   * Documented system requirements, design decisions, and implementation details for reference and future maintenance.
   * Prepared user manuals and guides to assist users in navigating the ATM Simulator effectively.
   * Compiled progress reports to track the project's evolution and communicate update

**Hardware/Software Requirements**

**Hardware Requirements:**

1. Processor: Intel Core i5 or equivalent processor (or higher recommended) for development and execution.
2. RAM: Minimum 4 GB of RAM (8 GB or more recommended) to ensure smooth operation during development and testing phases.
3. Storage: At least 100 MB of available disk space for storing the ATM Simulator application files and related resources.
4. Display: Minimum resolution of 1280x800 pixels recommended for optimal user interface display.
5. Input Devices: Standard keyboard and mouse for user interaction during development and testing.

**Software Requirements:**

1. Operating System: Compatible with Windows 10, macOS, or Linux distributions (Ubuntu, Fedora, etc.).
2. Java Development Kit (JDK): Version 8 or higher is required for compiling and running Java applications. It's recommended to use the latest stable version available.
3. Integrated Development Environment (IDE): A Java IDE such as IntelliJ IDEA, Eclipse, or NetBeans is recommended for software development and project management.
4. Version Control System: Git is recommended for version control to manage source code changes, collaboration, and tracking.
5. Build Tools: Apache Maven or Gradle can be used for building, packaging, and managing dependencies of the ATM Simulator project.
6. Database Management System: MySQL, PostgreSQL, or SQLite can be used to store transactional data and user information. JDBC drivers for the chosen database should be included in the project dependencies.
7. User Interface Framework: JavaFX or Swing can be utilized to develop the graphical user interface (GUI) for the ATM Simulator. Ensure that the necessary libraries and dependencies are included in the project configuration.
8. Logging Framework: Log4j or java.util.logging can be employed for logging system activities, errors, and transactions during runtime.

**Design**

**DATA FLOW DIAGRAM 1:**

+----------------------+

| ATM System |

+----------------------+

|

| User Actions

v

+----------------------------------------+

| User Interface |

+----------------------------------------+

|

| Transaction Data

v

+----------------------------------------+

| Transaction Processor |

+----------------------------------------+

|

| Database Access

v

+----------------------------------------+

| Database |

+----------------------------------------+

**DATA FLOW DIAGRAM 2:**

+-------------------+ +-------------------+

| User Interface | | Transaction |

| | | Processor |

+-------------------+ +-------------------+

| |

| Transaction Data | Transaction Data

v v

+-------------------+ +-------------------+

| Transaction | | Database |

| Selection | | Operations |

+-------------------+ +-------------------+

| Transaction Data |

v |

+-------------------+ +----------------------+

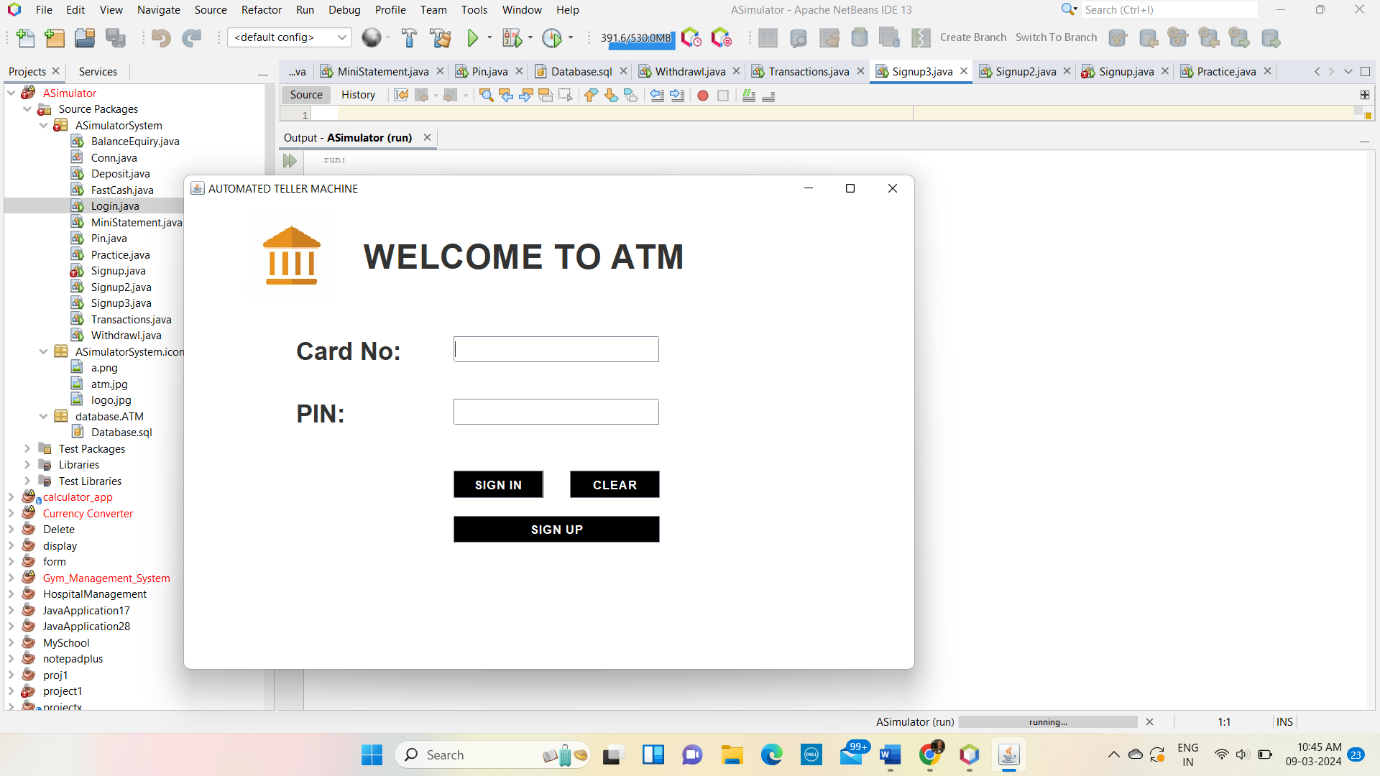
| Transaction | | User Accounts |

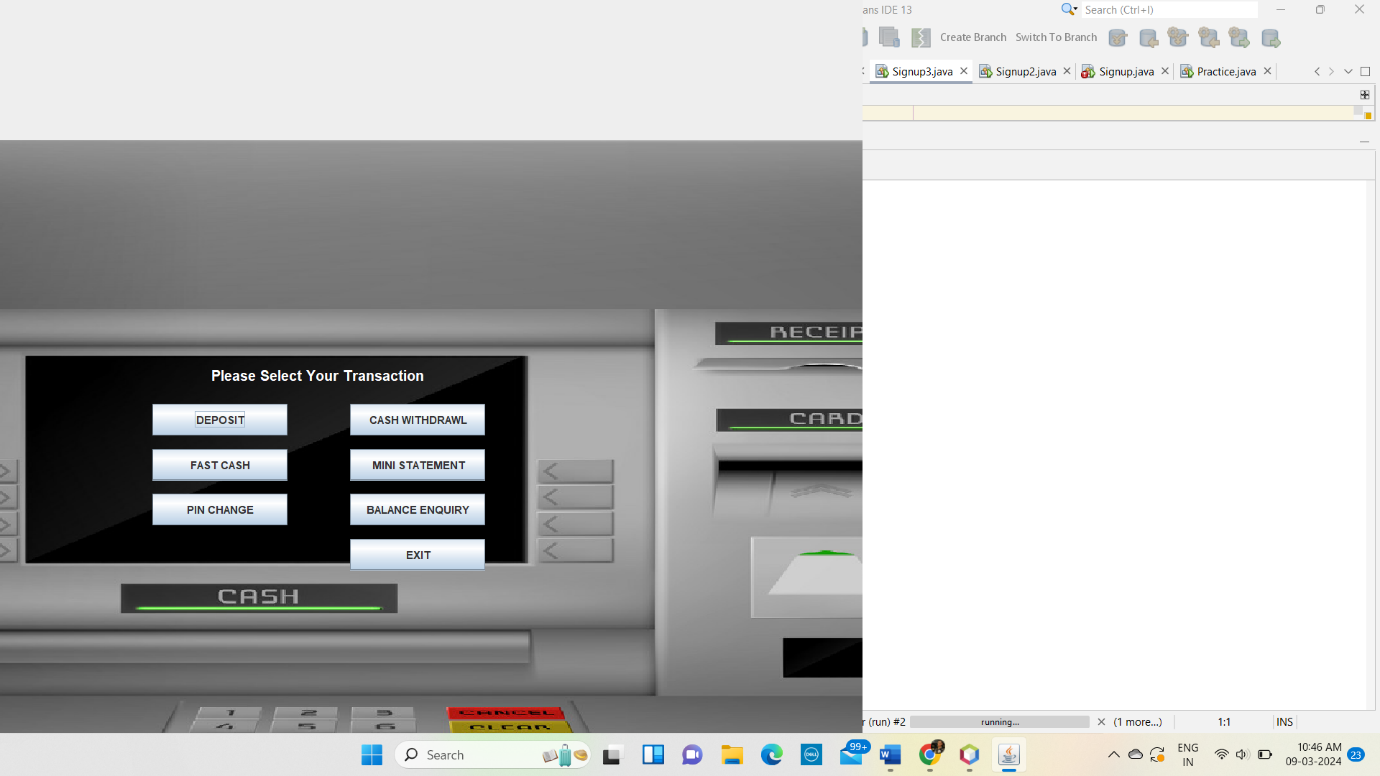
| Confirmation | | Transaction |

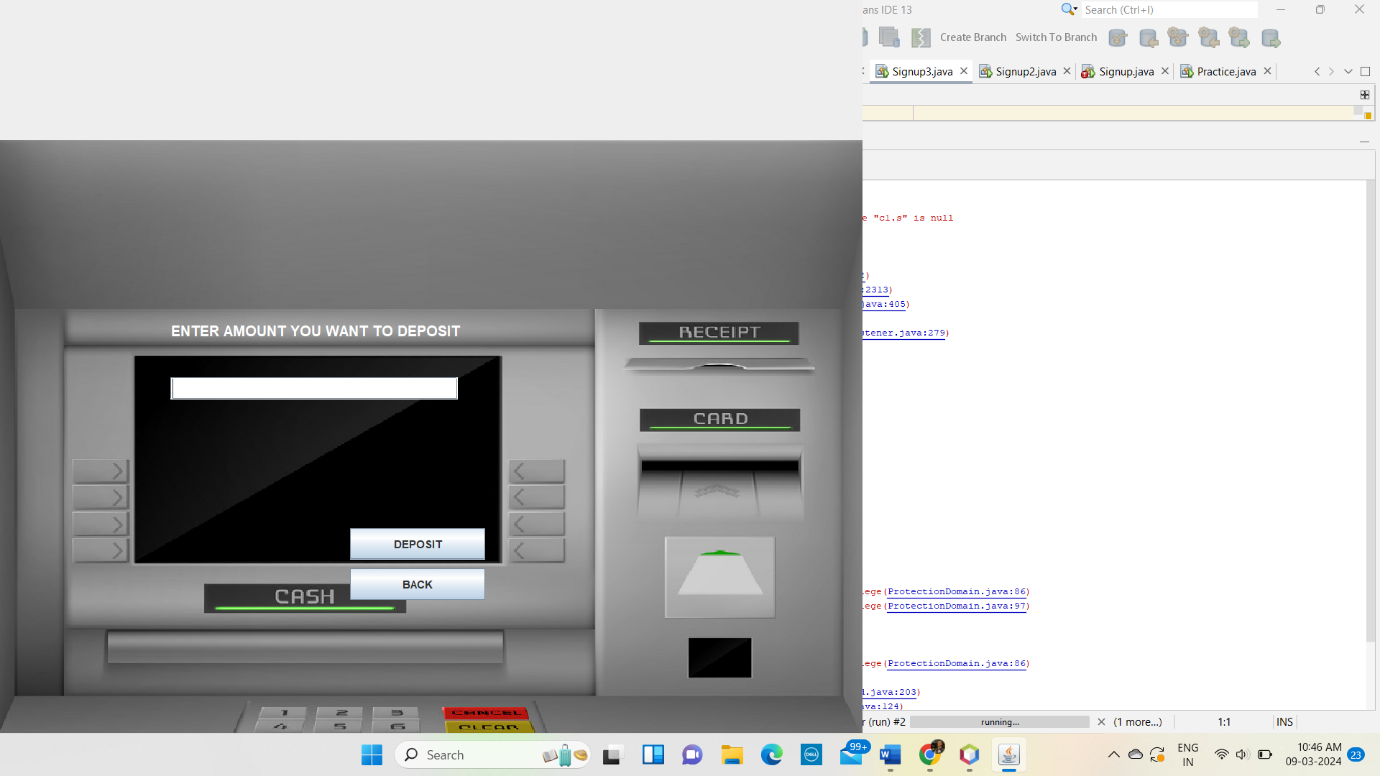
+-------------------+ | History |

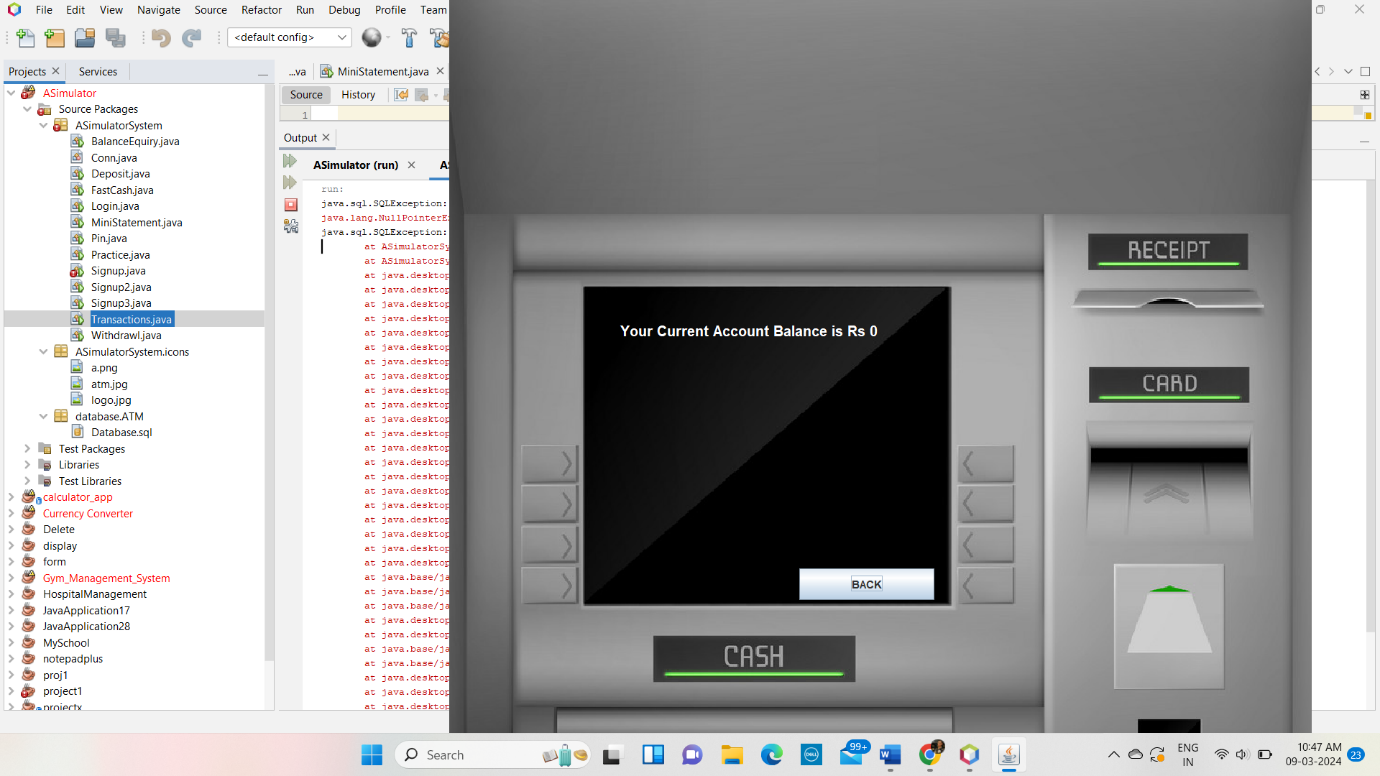
+----------------------+

**Interface Design**









**Detailed Description of New Functionalities added or activities underwent**

1. **User Authentication Enhancements**:
   * Implemented multi-factor authentication (MFA) to enhance security.
   * Integrated biometric authentication (fingerprint, facial recognition) for user identification.
   * Enhanced password hashing algorithms (e.g., bcrypt) for improved security against brute force attacks.
   * Implemented account lockout mechanisms after multiple failed login attempts to prevent unauthorized access.
2. **Transaction Processing Features**:
   * Added support for additional transaction types such as bill payments, account transfers, and mobile recharges.
   * Enhanced transaction validation mechanisms to prevent invalid or unauthorized transactions.
   * Implemented transaction scheduling for future-dated transactions and recurring payments.
   * Integrated real-time transaction notifications via email or SMS for user convenience and security alerts.
3. **User Interface Improvements**:
   * Redesigned the user interface for improved usability and accessibility.
   * Implemented customizable themes and layouts to cater to diverse user preferences.
   * Enhanced error handling and feedback mechanisms to provide informative messages to users.
   * Added support for multiple languages and localization to accommodate users from different regions.
4. **Transaction Logging and Auditing**:
   * Enhanced transaction logging capabilities to capture detailed information about each transaction, including user details, transaction type, timestamp, and location.
   * Implemented auditing mechanisms to track system activities and monitor for suspicious or unauthorized access attempts.
   * Integrated log aggregation and analysis tools to facilitate monitoring and troubleshooting of system issues.
5. **Performance Optimization**:
   * Conducted performance profiling and optimization to identify and address bottlenecks in the system.
   * Implemented caching mechanisms to improve response times and reduce database load.
   * Optimized database queries and indexing to enhance overall system performance and scalability.
   * Conducted stress testing and performance benchmarking to validate system performance under various load conditions.
6. **Security Enhancements**:
   * Implemented role-based access control (RBAC) to enforce granular access permissions based on user roles and privileges.
   * Enhanced data encryption protocols to protect sensitive user information during transmission and storage.
   * Conducted security assessments and penetration testing to identify and remediate potential security vulnerabilities.
   * Integrated security monitoring and intrusion detection systems to detect and respond to security threats in real-time.

**Code or Algorithm**

package ASimulatorSystem;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.sql.\*;

public class Transactions extends JFrame implements ActionListener{

JLabel l1;

JButton b1,b2,b3,b4,b5,b6,b7;

String pin;

Transactions(String pin){

this.pin = pin;

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/atm.jpg"));

Image i2 = i1.getImage().getScaledInstance(1366, 768, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l2 = new JLabel(i3);

l2.setBounds(0, 0, 960, 1080);

add(l2);

l1 = new JLabel("Please Select Your Transaction");

l1.setForeground(Color.WHITE);

l1.setFont(new Font("System", Font.BOLD, 16));

b1 = new JButton("DEPOSIT");

b2 = new JButton("CASH WITHDRAWL");

b3 = new JButton("FAST CASH");

b4 = new JButton("MINI STATEMENT");

b5 = new JButton("PIN CHANGE");

b6 = new JButton("BALANCE ENQUIRY");

b7 = new JButton("EXIT");

setLayout(null);

l1.setBounds(235,400,700,35);

l2.add(l1);

b1.setBounds(170,450,150,35);

l2.add(b1);

b2.setBounds(390,450,150,35);

l2.add(b2);

b3.setBounds(170,500,150,35);

l2.add(b3);

b4.setBounds(390,500,150,35);

l2.add(b4);

b5.setBounds(170,550,150,35);

l2.add(b5);

b6.setBounds(390,550,150,35);

l2.add(b6);

b7.setBounds(390,600,150,35);

l2.add(b7);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

b6.addActionListener(this);

b7.addActionListener(this);

setSize(960,1080);

setLocation(0,0);

setUndecorated(true);

setVisible(true);

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource()==b1){

setVisible(false);

new Deposit(pin).setVisible(true);

}else if(ae.getSource()==b2){

setVisible(false);

new Withdrawl(pin).setVisible(true);

}else if(ae.getSource()==b3){

setVisible(false);

new FastCash(pin).setVisible(true);

}else if(ae.getSource()==b4){

new MiniStatement(pin).setVisible(true);

}else if(ae.getSource()==b5){

setVisible(false);

new Pin(pin).setVisible(true);

}else if(ae.getSource()==b6){

this.setVisible(false);

new BalanceEnquiry(pin).setVisible(true);

}else if(ae.getSource()==b7){

System.exit(0);

}

}

public static void main(String[] args){

new Transactions("").setVisible(true);

}

}

package ASimulatorSystem;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.sql.\*;

public class Login extends JFrame implements ActionListener{

JLabel l1,l2,l3;

JTextField tf1;

JPasswordField pf2;

JButton b1,b2,b3;

Login(){

setTitle("AUTOMATED TELLER MACHINE");

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/logo.jpg"));

Image i2 = i1.getImage().getScaledInstance(100, 100, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l11 = new JLabel(i3);

l11.setBounds(70, 10, 100, 100);

add(l11);

l1 = new JLabel("WELCOME TO ATM");

l1.setFont(new Font("Osward", Font.BOLD, 38));

l1.setBounds(200,40,450,40);

add(l1);

l2 = new JLabel("Card No:");

l2.setFont(new Font("Raleway", Font.BOLD, 28));

l2.setBounds(125,150,375,30);

add(l2);

tf1 = new JTextField(15);

tf1.setBounds(300,150,230,30);

tf1.setFont(new Font("Arial", Font.BOLD, 14));

add(tf1);

l3 = new JLabel("PIN:");

l3.setFont(new Font("Raleway", Font.BOLD, 28));

l3.setBounds(125,220,375,30);

add(l3);

pf2 = new JPasswordField(15);

pf2.setFont(new Font("Arial", Font.BOLD, 14));

pf2.setBounds(300,220,230,30);

add(pf2);

b1 = new JButton("SIGN IN");

b1.setBackground(Color.BLACK);

b1.setForeground(Color.WHITE);

b2 = new JButton("CLEAR");

b2.setBackground(Color.BLACK);

b2.setForeground(Color.WHITE);

b3 = new JButton("SIGN UP");

b3.setBackground(Color.BLACK);

b3.setForeground(Color.WHITE);

setLayout(null);

b1.setFont(new Font("Arial", Font.BOLD, 14));

b1.setBounds(300,300,100,30);

add(b1);

b2.setFont(new Font("Arial", Font.BOLD, 14));

b2.setBounds(430,300,100,30);

add(b2);

b3.setFont(new Font("Arial", Font.BOLD, 14));

b3.setBounds(300,350,230,30);

add(b3);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

getContentPane().setBackground(Color.WHITE);

setSize(800,480);

setLocation(550,200);

setVisible(true);

}

public void actionPerformed(ActionEvent ae){

try{

if(ae.getSource()==b1){

Conn c1 = new Conn();

String cardno = tf1.getText();

String pin = pf2.getText();

String q = "select \* from login where cardnumber = '"+cardno+"' and pin = '"+pin+"'";

ResultSet rs = c1.s.executeQuery(q);

if(rs.next()){

setVisible(false);

new Transactions(pin).setVisible(true);

}else{

JOptionPane.showMessageDialog(null, "Incorrect Card Number or PIN");

}

}else if(ae.getSource()==b2){

tf1.setText("");

pf2.setText("");

}else if(ae.getSource()==b3){

setVisible(false);

new Signup().setVisible(true);

}

}catch(Exception e){

e.printStackTrace();

}

}

public static void main(String[] args){

new Login().setVisible(true);

}

}

package ASimulatorSystem;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.\*;

public class Deposit extends JFrame implements ActionListener{

JTextField t1,t2;

JButton b1,b2,b3;

JLabel l1,l2,l3;

String pin;

Deposit(String pin){

this.pin = pin;

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/atm.jpg"));

Image i2 = i1.getImage().getScaledInstance(1000, 768, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l3 = new JLabel(i3);

l3.setBounds(0, 0, 960, 1080);

add(l3);

l1 = new JLabel("ENTER AMOUNT YOU WANT TO DEPOSIT");

l1.setForeground(Color.WHITE);

l1.setFont(new Font("System", Font.BOLD, 16));

t1 = new JTextField();

t1.setFont(new Font("Raleway", Font.BOLD, 22));

b1 = new JButton("DEPOSIT");

b2 = new JButton("BACK");

setLayout(null);

l1.setBounds(190,350,400,35);

l3.add(l1);

t1.setBounds(190,420,320,25);

l3.add(t1);

b1.setBounds(390,588,150,35);

l3.add(b1);

b2.setBounds(390,633,150,35);

l3.add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

setSize(960,1080);

setUndecorated(true);

setLocation(0,0);

setVisible(true);

}

public void actionPerformed(ActionEvent ae){

try{

String amount = t1.getText();

Date date = new Date();

if(ae.getSource()==b1){

if(t1.getText().equals("")){

JOptionPane.showMessageDialog(null, "Please enter the Amount to you want to Deposit");

}else{

Conn c1 = new Conn();

c1.s.executeUpdate("insert into bank(pin,date,type,amount) values('"+pin+"', '"+date+"', 'Deposit', '"+amount+"')");

JOptionPane.showMessageDialog(null, "Rs. "+amount+" Deposited Successfully");

setVisible(false);

new Transactions(pin).setVisible(true);

}

}else if(ae.getSource()==b2){

setVisible(false);

new Transactions(pin).setVisible(true);

}

}catch(Exception e){

e.printStackTrace();

}

}

public static void main(String[] args){

new Deposit("").setVisible(true);

}

}

package ASimulatorSystem;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.Date;

import java.sql.\*;

public class Withdrawl extends JFrame implements ActionListener{

JTextField t1,t2;

JButton b1,b2,b3;

JLabel l1,l2,l3,l4;

String pin;

Withdrawl(String pin){

this.pin = pin;

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/atm.jpg"));

Image i2 = i1.getImage().getScaledInstance(1000, 768, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l3 = new JLabel(i3);

l3.setBounds(0, 0, 960, 1080);

add(l3);

l1 = new JLabel("MAXIMUM WITHDRAWAL IS RS.10,000");

l1.setForeground(Color.WHITE);

l1.setFont(new Font("System", Font.BOLD, 16));

l2 = new JLabel("PLEASE ENTER YOUR AMOUNT");

l2.setForeground(Color.WHITE);

l2.setFont(new Font("System", Font.BOLD, 16));

t1 = new JTextField();

t1.setFont(new Font("Raleway", Font.BOLD, 25));

b1 = new JButton("WITHDRAW");

b2 = new JButton("BACK");

setLayout(null);

l1.setBounds(190,350,400,20);

l3.add(l1);

l2.setBounds(190,400,400,20);

l3.add(l2);

t1.setBounds(190,450,330,30);

l3.add(t1);

b1.setBounds(390,588,150,35);

l3.add(b1);

b2.setBounds(390,633,150,35);

l3.add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

setSize(960,1080);

setLocation(0,0);

setUndecorated(true);

setVisible(true);

}

public void actionPerformed(ActionEvent ae){

try{

String withdraw = t1.getText();

Date date = new Date();

if(ae.getSource()==b1){

if(t1.getText().equals("")){

JOptionPane.showMessageDialog(null, "Please enter the Amount to you want to Withdraw");

}else{

Conn c1 = new Conn();

ResultSet rs = c1.s.executeQuery("select \* from bank where pin = '"+pin+"'");

int balance = 0;

while (rs.next()) {

if (rs.getString("type").equals("Deposit")) {

balance += Integer.parseInt(rs.getString("amount"));

} else {

balance -= Integer.parseInt(rs.getString("amount"));

}

}

if(balance < Integer.parseInt(withdraw)){

JOptionPane.showMessageDialog(null, "Insuffient Balance");

System.out.println(withdraw+" "+balance);

return;

}

// withdraw=0-withdraw;

c1.s.executeUpdate("insert into bank(pin,date,type,amount) values('"+pin+"', '"+date+"', 'Withdrawl', '"+withdraw+"')");

JOptionPane.showMessageDialog(null, "Rs. "+withdraw+" Debited Successfully");

setVisible(false);

new Transactions(pin).setVisible(true);

}

}else if(ae.getSource()==b2){

setVisible(false);

new Transactions(pin).setVisible(true);

}

}catch(Exception e){

e.printStackTrace();

System.out.println("error: "+e);

}

}

public static void main(String[] args){

new Withdrawl("").setVisible(true);

}

}

package ASimulatorSystem;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.ResultSet;

import javax.swing.\*;

//import java.util.\*;

class BalanceEnquiry extends JFrame implements ActionListener {

// JTextField t1, t2;

JButton b1; //,b2, b3;

JLabel l1;//, l2, l3;

String pin;

BalanceEnquiry(String pin) {

this.pin = pin;

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("ASimulatorSystem/icons/atm.jpg"));

Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l3 = new JLabel(i3);

l3.setBounds(0, 0, 960, 1080);

add(l3);

l1 = new JLabel();

l1.setForeground(Color.WHITE);

l1.setFont(new Font("System", Font.BOLD, 16));

b1 = new JButton("BACK");

setLayout(null);

l1.setBounds(190, 350, 400, 35);

l3.add(l1);

b1.setBounds(390, 633, 150, 35);

l3.add(b1);

int balance = 0;

try{

Conn c1 = new Conn();

ResultSet rs = c1.s.executeQuery("select \* from bank where pin = '"+pin+"'");

while (rs.next()) {

if (rs.getString("type").equals("Deposit")) {

balance += Integer.parseInt(rs.getString("amount"));

} else {

balance -= Integer.parseInt(rs.getString("amount"));

}

}

}catch(Exception e){}

l1.setText("Your Current Account Balance is Rs "+balance);

b1.addActionListener(this);

setSize(960, 1080);

setUndecorated(true);

setLocation(500, 0);

setVisible(true);

}

public void actionPerformed(ActionEvent ae) {

setVisible(false);

new Transactions(pin).setVisible(true);

}

public static void main(String[] args) {

new BalanceEnquiry("").setVisible(true);

}

}

**References**

1. Java Platform, Standard Edition (Java SE) Documentation - Oracle Corporation.
   * Website: https://docs.oracle.com/javase/
2. JavaFX Documentation - OpenJFX.
   * Website: https://openjfx.io/documentation.html
3. Apache Maven Documentation.
   * Website: https://maven.apache.org/guides/index.html
4. Git Documentation - Git SCM.
   * Website: https://git-scm.com/doc
5. JUnit Documentation.
   * Website: https://junit.org/junit5/docs/current/user-guide/
6. Log4j 2 Documentation.
   * Website: https://logging.apache.org/log4j/2.x/manual/index.html
7. Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (the Gang of Four).
8. Effective Java by Joshua Bloch.
9. Clean Code: A Handbook of Agile Software Craftsmanship by Robert C. Martin.
10. Java Concurrency in Practice by Brian Goetz et al.
11. MySQL Documentation - Oracle Corporation.
    * Website: https://dev.mysql.com/doc/
12. PostgreSQL Documentation.
    * Website: https://www.postgresql.org/docs/
13. SQLite Documentation.
    * Website: https://www.sqlite.org/docs.html
14. Software Engineering: A Practitioner's Approach by Roger S. Pressman.