BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi
Autonomous Institute under Visvesvaraya Technological University, Belagavi)
"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur,
Ballar1-583 104 (Karnataka) (India)
Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

DEPARTMENT OF ARTIFICIAL INTELLEGENCE AND MACHINE LEARNING

A DBMS Mini Project Report

On

"ATM MANAGEMENT SYSTEM"

Submitted By

M DEEPA	3BR21AI055
DAMODHAR SINGH R	3BR21AI025
MALA M	3BR21AI059
MANOJ B	3BR21AI060

Under the Guidance of

Mrs. SYEDA BADRUNNISA BEGUM

Mrs. MANIKESHWARI

Dept of AI&ML,

BITM, Ballari.

Visvesvaraya Technological University

Belagavi, Karnataka

2023-2024

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi
Autonomous Institute under Visvesvaraya Technological University, Belagavi)
"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur,
Ballar1-583 104 (Karnataka) (India)
Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF ARTIFICAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify This that the DBMS Mini Project entitled “ATM MANAGEMENT SYSTEM” has been successfully presented by M DEEPA, DAMODHAR SINGH R, MALA M,MANOJ B bearing **3BR21AI055, 3BR21AI025, 3BR21AI059, 3BR21AI060** students of V semester B.E. for the partial fulfillment of the requirements for the award of **Bachelor Degree in Artificial intelligence and machine learning** of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY during the academic year 2023-2024

Signature of Guide(s)	Signature of HOD
Mrs.Syeda Badrunnisa Begum	Dr. BM Vidyavathi

ACKNOWLEDGEMENT

On presenting the **Database Management System Laboratory Mini Project** report on “ATM MANAGEMENT SYSTEM”, we feel great to express humble feelings of thanks to all who have helped directly or indirectly in the successful completion of project work

We would like to thank our respected guides, **Mrs. Syeda Badrunnisa** and **Mrs. Manikeshwari** Assistant Professor, Department of AIML, who have helped us a lot in completing this task and for giving continuous encouragement and guidance throughout the project work.

We would like to thank **Dr. BM Vidhyavathi**, Professor and HOD, Department of AIML BITM, Bellary and **Dr Yadavalli Basavaraj**, Principal, BITM, Bellary, for all their support and encouragement.

We are grateful to Department of Artificial intelligence and machine learning Engineering and our Institution, Ballari Institute of Technology and Management, Bellary for imparting us the knowledge with which we could do our best. Finally, we also would like to thank the whole teaching and non-teaching staff of Artificial intelligence and machine learning Department.

ABSTRACT

The ATM Management System is a comprehensive software solution designed to streamline and optimize the operation of Automated Teller Machines (ATMs). This system focuses on enhancing the efficiency, security, and overall management of ATM networks within financial institutions. Key features include robust transaction processing, advanced security protocols, real-time monitoring, user management, remote diagnostics, and seamless integration with external systems. The system aims to provide a user-friendly interface for both administrators and end-users, ensuring a smooth and secure experience. By leveraging cutting-edge technologies and adhering to industry standards, the ATM Management System contributes to the reliability, scalability, and overall effectiveness of ATM networks, meeting the evolving needs of the financial technology landscape.

TABLE OF CONTENTS

CHAPTER NO.	CONTENT	PAGE NO.
	Acknowledgement	I
	Abstract	II
	Table of Contents	III
CHAPTER 1	**INTRODUCTION**	**1-3**
	1.1 OVERVIEW	1
	1.2 PROBLEM STATEMENT	1
	1.3 DATABASE MANAGEMENT SYSTEM	2
	1.4 SQL	3
	1.5 HTML	3
CHAPTER 2	**REQUIREMENTS SPECIFICATION**	**4-6**
	2.1 OVERALL DESCRIPTION	4
	2.2 SPECIFIC REQUIREMENTS	4
	2.3 SOFTWARE REQURIMENTS	5
	2.4 TECHNOLOGY	6
CHAPTER 3	**DETAILED DESIGN**	**7-15**
	3.1 SYSTEM DESIGN	7
	3.2 ENTITY RELATIONSHIP DIAGRAM	12
	3.3 RELATIONAL SCHEMA	13
	3.4 DESCRIPTION OF TABLES	14
CHAPTER 4	**IMPLEMENTATIONS**	**16-18**
	4.1 MODULES AND THEIR DESCRIPTION	16
	4.2 TRIGGERS AND STORED PROCEDURES	17
	4.3 RESULT	18
CHAPTER 5	**TESTING**	**19-22**

5.1 SOFTWARE TESTING 19

5.2 MODULE TESTING AND INTEGRATION 20

5.3 LIMITATIONS 22

CHAPTER 6 SNAPSHOT 23-31

6.1 WELCOME PAGE 23

6.2 LOGIN PAGE 24

6.3 MENU PAGE 24

6.4 BALANCE ENQUIRY 25

6.5 WITHDRAW PAGE 25

6.6 PIN CHANGE PAGE 26

6.7 FAST CASH PAGE 27

6.8 TRANSACTION PAGE 28

6.9 DAILY LIMIT WARNING MESSAGE 29

6.10 ADMIN LOGIN PAGE 29

6.11 ADD CASH PAGE 30

6.12 TRANSACTION SUMMARY PAGE 31

CHAPTER 6 CONCULSION 32

CHAPTER 7 FUTURE ENHANCEMENTS 33

REFERENCES 34

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

Automated Teller Machines (ATMs) have revolutionized the way individuals access and manage their finances, providing convenient and round-the-clock banking services. The dynamic nature of the financial industry, coupled with technological advancements, has necessitated the development of sophisticated systems to efficiently manage and secure ATM networks. This introduction provides an overview of the pivotal role played by the ATM Management System in addressing the complexities associated with ATM operations within the framework of financial institutions. Over the years, ATMs have evolved from simple cash dispensers to multifunctional terminals capable of handling a diverse range of financial transactions. The widespread adoption of ATMs hassignificantly increased the demand for robust systems that can manage these networks effectively, ensuring seamless and secure operations. As financial institutions strive to provide enhanced services to their customers, the importance of an integrated and comprehensive ATM Management System becomes evident. This system goes beyond basic transaction processing, incorporating features that focus on security, real-time monitoring, user-friendly interfaces, and integration with external systems. With the rise of cyber threats and financial fraud, security is a top priority for both financial institutions and ATM users. The ATM Management System takes a proactive approach to address these concerns, implementing advanced security protocols such as biometric authentication, encrypted communication channels, and real-time monitoring to safeguard sensitive data and prevent unauthorized access.

1.2 PROBLEM STATEMENT

To design and develop an efficient ATM Management System using SQL to handle the various functionalities associated with automated teller machines (ATMs). The system will facilitate seamless and secure transactions for users while ensuring the integrity and confidentiality of their financial information

1.3 DATABASE MANAGEMENT SYSTEM

A Database Management System (DBMS) is a software suite designed to efficiently store, retrieve, and manage data in databases. It acts as an interface between the database and end users or applications, facilitating data organization, access, and manipulation while ensuring data integrity, security, and concurrency control.

Key components of a DBMS include:

1. Data Definition Language (DDL): Used to define the structure of the database schema, including tables, relationships, constraints, and indexes.

2. Data Manipulation Language (DML): Allows users to perform operations such as insertion, deletion, modification, and retrieval of data in the database.

3. Query Language: Enables users to retrieve specific data from the database using queries, often through SQL (Structured Query Language).

4. Transaction Management: Ensures the integrity and consistency of data by managing transactions, which are sequences of database operations treated as a single unit.

5. Concurrency Control: Handles simultaneous access to the database by multiple users or applications, preventing conflicts and ensuring data consistency.

6. Data Security: Implements measures to protect sensitive data from unauthorized access, including user authentication, access control, and encryption.

7. Backup and Recovery: Provides mechanisms for creating backups of the database to prevent data loss and restoring the database to a previous state in case of failures or disasters.

DBMS can be categorized into different types based on their data model (relational, hierarchical, network, object-oriented, etc.) and architecture (centralized, distributed, client-server, etc.). Popular examples of DBMS include MySQL, Oracle Database, Microsoft SQL Server, PostgreSQL, and MongoDB.

1.4 SQL

In an ATM management system, SQL serves as the backbone, storing user account details, transaction records, and ATM information. It facilitates secure authentication processes, ensuring user data integrity and confidentiality. SQL enables efficient management of ATM operations, including monitoring cash levels, location tracking, and maintenance scheduling. Additionally, it supports real-time transaction logging for accurate financial record-keeping and provides a platform for generating insightful reports and analytics to optimize system performance and enhance user experience.

1.5 HTML

HTML, short for Hypertext Markup Language, serves as the backbone of the World Wide Web, providing the fundamental structure for web pages. It uses a system of tags and attributes to define the various elements within a webpage, such as headings, paragraphs, links, images, and more. These elements are organized hierarchically, forming the structure of the document. HTML also enables the inclusion of multimedia content like images, audio, and video, as well as interactive features such as forms and buttons. With its simplicity and versatility, HTML lays the groundwork for creating engaging and accessible online experiences, shaping the way we interact with information on the internet.

CHAPTER 2: REQUIREMENTS SPECIFICATION

2.1 OVERAL DESCRIPTION

An ATM management system is a comprehensive software platform designed to oversee and optimize the operations of Automated Teller Machine (ATM) networks within a financial institution. This system integrates various functionalities to ensure seamless ATM functioning, efficient transaction processing, robust security measures, and effective resource management. It provides real-time monitoring and control capabilities, allowing administrators to remotely track ATM status, health, and performance. Additionally, the system facilitates secure transaction processing for banking services like cash withdrawals, balance inquiries, fund transfers, and bill payments while adhering to strict security protocols and regulatory standards. Furthermore, it optimizes cash management by forecasting cash demand, monitoring cash levels, and scheduling cash replenishments to prevent cash shortages or excesses. Maintenance, diagnostics, reporting, and compliance features are also included to enhance operational efficiency, security, and customer experience within the banking industry.

2.2 SPECIFIC REQUIREMENTS

An ATM management system must fulfill a range of specific requirements to effectively oversee and optimize Automated Teller Machine (ATM) operations within a financial institution. These requirements include robust user authentication mechanisms such as PIN verification and biometric authentication to ensure secure access to ATM services. The system should support various transaction types, including cash withdrawals, balance inquiries, fund transfers, and bill payments, while also enabling efficient cash management through monitoring cash levels, forecasting demand, and optimizing replenishment schedules. Security is paramount, necessitating features such as encryption of sensitive data, real-time transaction monitoring, and fraud detection algorithms. Remote monitoring and control capabilities allow administrators to track ATM performance, conduct maintenance tasks, and deploy software updates remotely. Comprehensive reporting and analytics tools provide insights into ATM performance and transaction trends, while ensuring compliance with regulatory standards such as PCI DSS and GDPR. Usability and accessibility are also critical, requiring user-friendly interfaces and adherence to accessibility standards to accommodate users with disabilities. Lastly, the system should be scalable and flexible to accommodate growth and changes in ATM networks, supporting seamless integration with banking systems and third-party services.

2.3 SOFTWARE REQUIREMENTS

Frontend:
HTML: HTML5 for structuring web pages.
CSS: CSS3 for styling the interface.
JavaScript: Optional for client-side interactivity
Backend:
PHP: PHP for server-side scripting.
Database: MySQL or PostgreSQL for data storage.
Web Server: Apache or Nginx to host the application.
Development Tools: Text editor or IDE for coding, Git for version control.
Security: Implement security measures such as input validation and secure authentication.
Compatibility: Ensure cross-browser compatibility and responsive design.
Documentation: Provide comprehensive documentation covering system architecture, installation, and user guides.

2.5 TECHNOLOGY

Frontend Technologies:
HTML5: Utilized for structuring the various elements and content of the web pages in the ATM management system.
CSS3: Employed for styling and formatting the HTML elements to enhance the visual presentation and user experience.
JavaScript (optional): While not essential, JavaScript can be utilized for client-side validation, dynamic content manipulation, and enhancing interactivity within the user interface.
Backend Technologies:
PHP: Used as the primary server-side scripting language for implementing the backend logic, processing requests, and interacting with the database.
MySQL or PostgreSQL: Chosen as the relational database management system (RDBMS) for storing and managing data related to ATM transactions, user accounts, system configurations, and logs.
Web Server:
Apache HTTP Server: Selected as the web server software to host the ATM management system, serving HTML, CSS, and PHP files to clients and handling incoming HTTP requests.
Development Tools:

Text Editor or IDE: Utilized for writing, editing, and managing the HTML, CSS, PHP codebase. Popular options include Visual Studio Code, Sublime Text, Atom, or PHPStorm.

Version Control System (optional): Git can be used for version control to track changes to the codebase, collaborate with other developers, and manage project history.

Security Measures:

Implementation of security measures such as input validation, data sanitization, secure authentication mechanisms, and protection against common web vulnerabilities like SQL injection and cross-site scripting (XSS).

Cross-Browser Compatibility:

Ensuring that the web application functions consistently across different web browsers and platforms, including Chrome, Firefox, Safari, Edge, and others.

Responsive Design:

Incorporating responsive design techniques using CSS frameworks like Bootstrap or Flexbox to ensure the ATM management system's optimal display and usability on various devices, including desktops, tablets, and smartphones.

CHAPTER 3: DETAILED DESIGN

3.1 SYSTEM DESIGN

1.System Overview:

Purpose:- The purpose of an ATM management system is to ensure the efficient and secure operation of Automated Teller Machines (ATMs) within a financial institution. This system streamlines operational processes such as cash management, transaction processing, and maintenance scheduling, optimizing ATM operations and reducing downtime. It facilitates secure transaction processing, implements robust security measures to prevent fraud, and provides tools for proactive maintenance and diagnostics to ensure ATM availability. Additionally, the system offers reporting and analytics capabilities to monitor ATM performance and compliance with regulatory standards. Overall, the ATM management system aims to enhance operational efficiency, security, and reliability, providing a seamless banking experience for customers while meeting the needs of financial institutions.

Key Features:

Remote Monitoring: Real-time monitoring of ATM status, health, and performance from a centralized location, allowing administrators to track operational metrics and identify issues promptly.

Transaction Processing: Facilitation of secure and efficient processing of various transactions, including cash withdrawals, balance inquiries, fund transfers, bill payments, and deposit processing.

Cash Management: Optimization of cash inventory through monitoring cash levels, forecasting demand, and scheduling replenishments to ensure ATMs have adequate cash supplies without excessive holding.

Security Measures: Implementation of robust security measures such as encryption, secure authentication mechanisms, transaction monitoring, and fraud detection algorithms to safeguard ATM transactions and customer data.

Maintenance and Diagnostics: Tools for proactive maintenance, fault detection, troubleshooting, and remote diagnostics to ensure optimal performance and minimize downtime.

Reporting and Analytics: Comprehensive reporting and analytics capabilities to monitor ATM performance, transaction trends, cash usage patterns, and operational efficiency, providing insights for optimization and decision-making.

Compliance Management: Features to ensure compliance with regulatory standards and industry guidelines governing ATM operations, security, data privacy, and financial transactions.

User Management: Administration of user access rights, permissions, and roles, ensuring proper access control and security within the ATM management system.

Integration with Banking Systems: Integration with the bank's core banking system and third-party services for seamless data exchange, transaction processing, and reporting.

2.System Architecture:

Presentation Layer:

This layer encompasses the user interface components that interact with end-users, including ATM screens, keypads, and receipt printers.

Technologies such as HTML, CSS, and JavaScript are used to create the user interface elements and enable interactivity.

User input and actions are transmitted to the application layer for processing.

Application Layer:

The application layer contains the business logic and processing components of the ATM management system.

It handles user requests, transaction processing, authentication, authorization, and integration with external systems such as banking databases and payment networks.

This layer is implemented using server-side technologies like PHP, along with frameworks and libraries for handling HTTP requests, routing, and data manipulation.

Database Layer:

The database layer stores and manages data related to ATM operations, transactions, user accounts, system configurations, and logs.

A relational database management system (RDBMS) like MySQL or PostgreSQL is commonly used for this purpose.

The database schema is designed to support efficient data storage, retrieval, and manipulation, with appropriate indexing and normalization to ensure data integrity and performance.

Integration Layer:

The integration layer facilitates communication between the ATM management system and external systems, such as the bank's core banking system, payment networks, and third-party services.

APIs, web services, or messaging protocols are used for data exchange and interoperability between different systems.

This layer ensures seamless integration with external systems for transaction processing, account management, and reporting.

Security Layer:

The security layer implements measures to protect the ATM management system from unauthorized access, data breaches, and fraudulent activities.

Security features include encryption of sensitive data, secure authentication mechanisms, access control, logging and auditing, intrusion detection, and monitoring for suspicious behavior.

Compliance with regulatory standards such as PCI DSS, GDPR, and ADA accessibility requirements is also addressed in this layer.

Infrastructure Layer:

The infrastructure layer includes the hardware and network infrastructure that supports the operation of the ATM management system.

This includes servers, storage devices, networking equipment, and other infrastructure components deployed in data centers or cloud environments.

High availability, scalability, and redundancy are key considerations in designing the infrastructure to ensure uninterrupted service and accommodate growth in transaction volumes.

3.Data Model:

Account: account type, and user ID and basic detail’s.

admin: Contains details of ATM administrators, including admin ID, PIN, and ATM ID.

atm: Stores information about Automated Teller Machines (ATMs), including ATM ID, location, and available cash.

bank: Stores information about banks, including bank name, bank ID, branch location, and associated account.

card: Stores information about ATM cards, including card number, PIN, balance, and user ID.

logs: Records logs of account balance changes, including log ID, user ID, updated balance, and creation date.

transaction: Stores information about transactions, including transaction date, transaction ID, status, type, and user ID.

user: Contains details of bank users, including user ID, first name, last name, address, contact number, and date of birth.

Relationships:

account and user: One-to-many relationship between users and their bank accounts.

admin and atm: One-to-many relationship between ATM administrators and ATMs they manage.

card and account: One-to-one relationship between ATM cards and bank accounts.

logs and account: One-to-many relationship between account logs and the corresponding bank account.

transaction and user: One-to-many relationship between users and their transactions.

transaction and atm: One-to-many relationship between ATM transactions and the corresponding ATM.

Constraints:

Foreign key constraints are defined to maintain referential integrity between related tables, ensuring that relationships between entities are enforced.

For example, the foreign key constraints in the card table reference the account table to ensure that the balance and user ID in the card table correspond to existing account records.

4.System Workflow:

User Authentication and Authorization:

Users log in to the online ATM management system using their credentials (pin number).

Upon successful authentication, the system verifies the user's role and permissions to determine their access level.

ATM Management:

Authorized administrators access the ATM management module to perform tasks such as adding cash and viewing recent transactions.

Transaction Processing:

Users initiate transactions such as cash withdrawals, balance inquiries, or fund transfers through the online ATM system.

The system validates the transaction request, checks the user's account balance, and processes the transaction accordingly.

For cash withdrawals, the system updates the user's account balance, deducts the withdrawn amount, and records the transaction in the database.

For other transactions, such as balance inquiries or fund transfers, the system retrieves and displays relevant account information to the user.

Account Management:

Users manage their bank accounts through the online ATM system, including viewing account details.

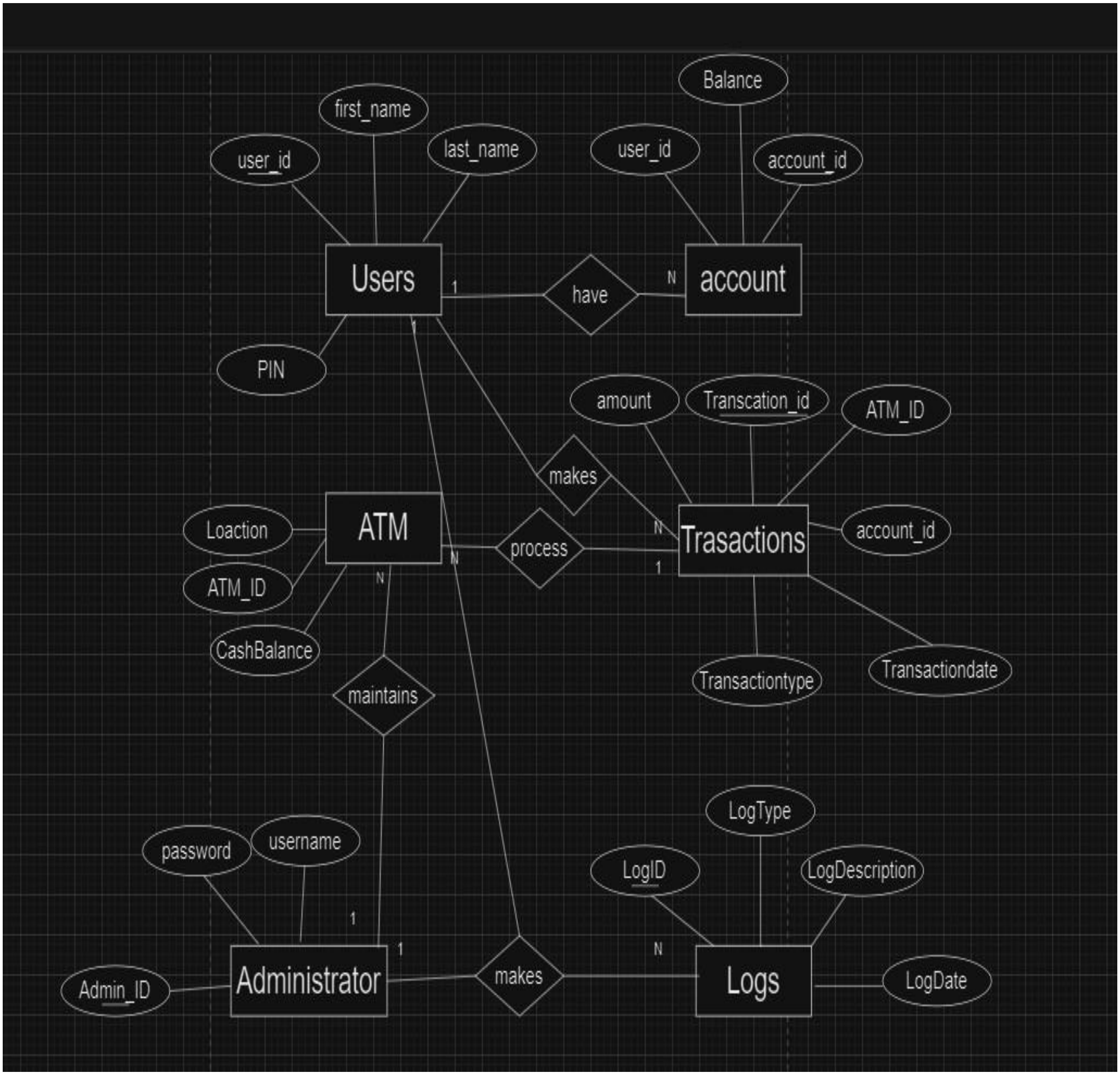
Transaction Logging and Reporting:

Each transaction processed through the online ATM system is logged in the database, capturing details such as transaction type, amount, timestamp, and user ID.

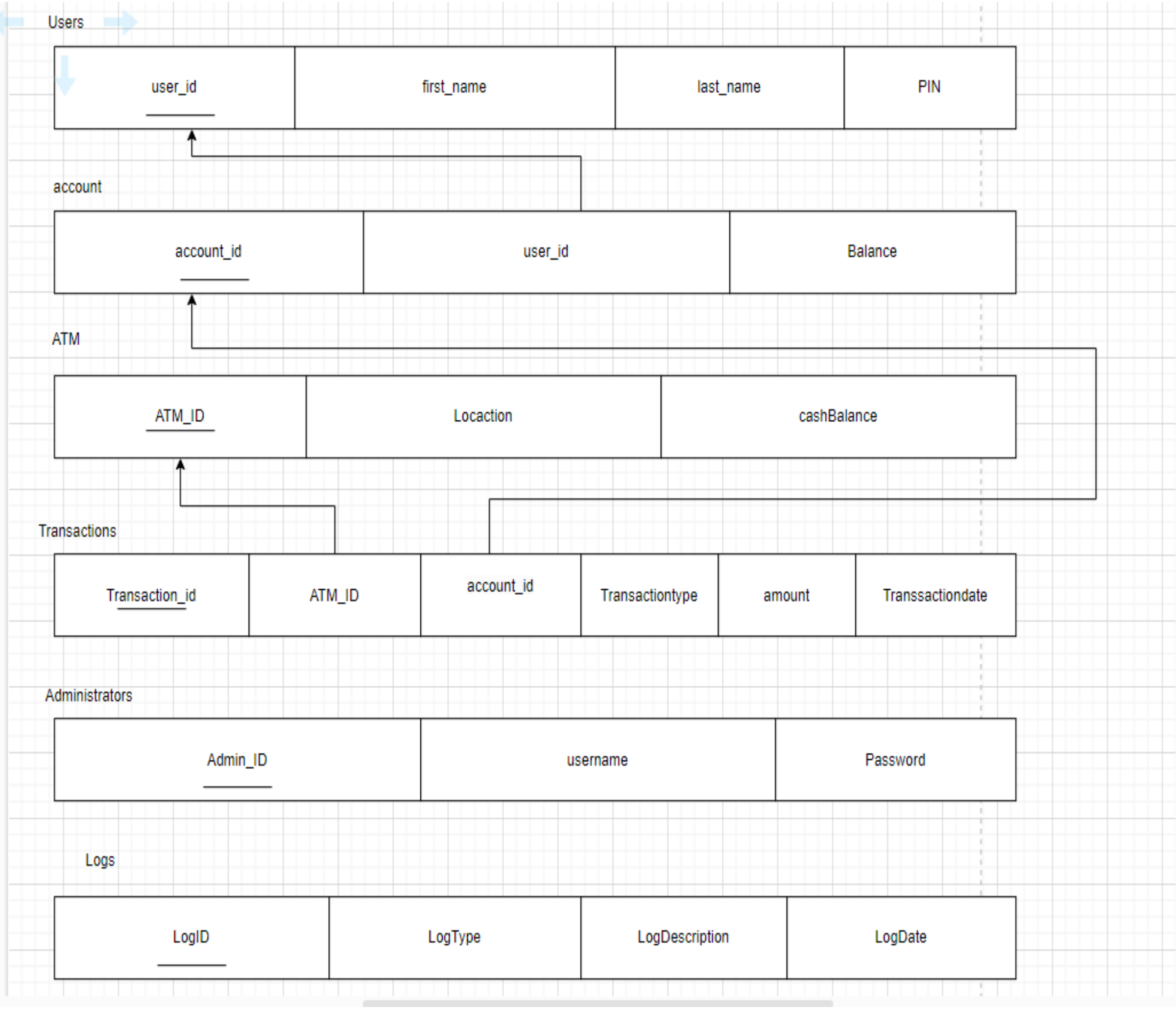
Transaction logs are used to generate reports on ATM usage, transaction volumes, cash flows, and user activity, providing insights for monitoring and analysis.

- Error Handling: Implement proper error handling mechanisms for database errors, invalid user input, and potential security threats.

2. ENTITY RELATIONSHIP DIAGRAM



3.3 RELATIONAL SCHEMA



3.5 DESCRIPTION OF TABLES

Account:

Fields: Account_number (Primary Key), Balance, Account_type, user_id (Foreign Key referencing user table)

Description: Stores information about bank accounts, including account number, balance, account type, and user ID.

admin:

Fields: admin_id (Primary Key), admin_pin, atm_id (Foreign Key referencing atm table)

Description: Contains details of ATM administrators, including admin ID, PIN, and ATM ID.

atm:

Fields: atm_id (Primary Key), atm_location, available_cash

Description: Stores information about Automated Teller Machines (ATMs), including ATM ID, location, and available cash.

bank:

Fields: Bank_name, Bank_id (Primary Key), Branch_Location, account (Foreign Key referencing account table)

Description: Stores information about banks, including bank name, bank ID, branch location, and associated account.

card:

Fields: card_no (Primary Key), card_pin, balance, user_id (Foreign Key referencing account table)

Description: Stores information about ATM cards, including card number, PIN, balance, and user ID.

logs:

Fields: id (Primary Key), user_id (Foreign Key referencing user table), balance, created_date

Description: Records logs of account balance changes, including log ID, user ID, updated balance, and creation date.

transaction:

Fields: transaction_date, transaction_id (Primary Key), transaction_status, transaction_type, user_id (Foreign Key referencing user table)

Description: Stores information about transactions, including transaction date, ID, status, type, and user ID.

user:

Fields: user_id (Primary Key), first_name, last_name, address, contact_no, DOB

Description: Contains details of bank users, including user ID, first name, last name, address, contact number, and date of birth.

CHAPTER 4: IMPLEMETATION

4.1 MODULE AND THEIR ROLES

User Authentication Module:

Role: Handles user authentication and authorization based on PIN.

Responsibilities:

Verifies user identity using their PIN.

Grants access to ATM functionalities based on PIN validation.

Transaction Processing Module:

Role: Processes user-initiated transactions at the ATM.

Responsibilities:

Validates transaction requests.

Facilitates balance inquiries, cash withdrawals, fast cash transactions, and PIN changes.

Updates account balances and transaction logs accordingly.

Account Information Module:

Role: Provides access to user account information.

Responsibilities:

Retrieves and displays account balance information upon request.

Allows users to view their account details before initiating transactions.

Cash Dispensing Module:

Role: Controls the dispensing of cash during transactions.

Responsibilities:

Manages the cash dispensing mechanism within the ATM.

Dispenses cash according to transaction requests while ensuring sufficient funds in the ATM.

PIN Change Module:

Role: Facilitates the process of changing the user's PIN.

Responsibilities:

Verifies the user's current PIN for authentication.

Allows users to input and confirm a new PIN for their account.

Updates the user's PIN in the system upon successful validation and confirmation.

Admin Management Module:

Role: Manages administrative functionalities for the ATM system.

Responsibilities:

Allows admin users to log in and access admin functionalities.

Enables admin users to add cash to the ATM when required.

Provides access to view transaction history for monitoring and audit purposes.

4.2 TRIGGERS AND STORED PROCEDURES

Triggers:

Transaction Logging Trigger: A trigger can be set up to automatically log every transaction performed on the system. This trigger would execute after an INSERT operation on the transaction table, capturing relevant details such as transaction type, amount, timestamp, and user ID. This helps in maintaining an audit trail for all transactions.

Balance Update Trigger: Another trigger can be implemented to update the account balance automatically after a successful transaction. This trigger would execute after an INSERT operation on the transaction table and would update the account balance based on the transaction type (e.g., deposit or withdrawal).

Stored Procedures:

Withdrawal Procedure: A stored procedure can handle the withdrawal process, ensuring that the user's account balance is updated correctly and that sufficient funds are available before processing the withdrawal. It would take inputs such as the user's account details and the withdrawal amount, perform the necessary validations, update the account balance, and log the transaction.

Pin Change Procedure: This procedure would allow users to change their PIN securely. It would verify the user's current PIN, prompt for a new PIN, and update the PIN in the database after ensuring it meets the required security criteria.

Cash Replenishment Procedure: For administrators, a stored procedure can facilitate the process of adding cash to ATMs. It would take inputs such as the ATM ID and the amount of cash to be added, update the available cash in the ATM, and log the replenishment transaction.

These triggers and stored procedures ensure that transactions are logged accurately, account balances are updated correctly, and critical operations such as withdrawals and PIN changes are executed securely. They enhance the functionality, security, and efficiency of the ATM management system.

4.3 RESULT

The ATM management system efficiently handles user transactions, ensuring secure authentication through PIN verification. It allows users to perform basic operations such as balance inquiries, cash withdrawals, and PIN changes seamlessly. Administrators can monitor and replenish cash levels in ATMs, while transaction logs maintain a comprehensive record of all system activities. This system enhances convenience for users and enables effective management of ATM resources, ensuring smooth operation and customer satisfaction.

CHAPTER 5: TESTING

5.1 SOFTWARE TESTING

1. Unit Testing:

   - Verify individual components or methods in isolation.
   - Use testing frameworks like JUnit to create and run unit tests.
   - Test cases should cover various scenarios, including boundary cases and edge cases.

2. Integration Testing:

   - Verify interactions between different components or modules.
   - Ensure proper communication between GUI elements and backend logic.
   - Validate the integration of database operations.

3. User Interface (UI) Testing:

   - Confirm that the graphical user interface functions as expected.
   - Use tools like Selenium or specialized GUI testing frameworks for Java Swing applications.
   - Verify proper button clicks, window openings, and data input.

4. Database Testing:

   - Check the interactions with the MySQL database.
   - Ensure that data is correctly inserted, checking balance and pin change etc in atm tables.
   - Validate the handling of SQL exceptions and errors.

5. Security Testing:

   - Evaluate the application's security measures, especially concerning user authentication and password handling.
   - Verify that sensitive information is stored securely in the database.
   - Check for SQL injection vulnerabilities in database queries.

6. Performance Testing:

- Assess the application's responsiveness and stability under different loads.
- Test concurrent user logins, diary entries, and content updates.
- Identify and address any performance bottlenecks.

7. Error Handling Testing:

- Validate the application's response to invalid inputs, such as incorrect user credentials.
- Check error messages and ensure they are informative without revealing sensitive information.

8. Usability Testing:

- Evaluate the overall user experience and interface design.
- Ensure that user interactions are intuitive and error messages are user-friendly.
- Check for consistency in GUI elements across different functionalities.

5.2 MODULE TESTING AND INTEGRATION

Module Testing:

User Authentication Module Testing:

Verify that user authentication works correctly by testing with valid and invalid PINs.

Test different scenarios such as expired cards, locked accounts, and incorrect PIN attempts to ensure proper error handling.

Ensure that user roles and permissions are enforced accurately.

Transaction Processing Module Testing:

Test each transaction type individually (e.g., balance inquiry, cash withdrawal, PIN change) to ensure they are processed accurately.

Validate the correctness of transaction logs and updates to account balances.

Test edge cases such as large withdrawals, insufficient funds, or network failures to ensure robustness.

Integration Testing:

User Authentication and Transaction Processing Integration:

Test the integration between the user authentication module and the transaction processing module to ensure that authenticated users can perform transactions seamlessly.

Verify that user authentication status is properly communicated to the transaction processing module.

Test scenarios where transactions are denied due to authentication failures.

Transaction Processing and Account Management Integration:

Test the integration between the transaction processing module and the account management module to ensure that account balances are updated accurately after transactions.

Verify that transaction details are logged correctly in the transaction history.

Test scenarios where account updates fail (e.g., due to database errors) and ensure proper error handling.

Scenario-based Testing:

Conduct scenario-based testing to simulate real-world usage scenarios such as cash withdrawals, balance inquiries, and PIN changes.

Test end-to-end scenarios involving multiple modules to ensure that the system behaves as expected from the user's perspective.

Include both normal and exceptional scenarios to validate system behavior under different conditions.

Mocking and Stubbing:

Use mocks and stubs to simulate the behavior of external dependencies such as database interactions or network calls during module testing.

This allows for isolated testing of individual modules without relying on the functionality of external components.

5.3 LIMITATIONS

Dependence on Internet Connectivity: Online ATM management systems rely on internet connectivity for operations. Any disruptions in internet connectivity can hinder the system's functionality, leading to service outages or delays in processing transactions.

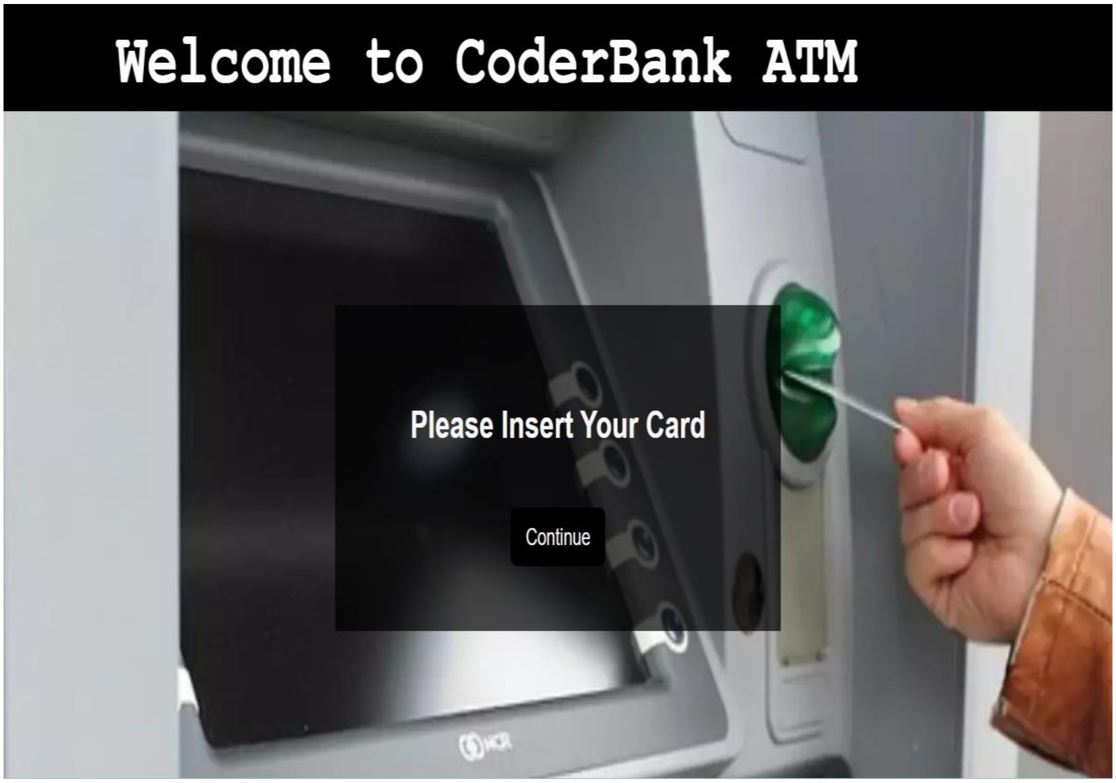
Security Concerns: Online systems are susceptible to security threats such as hacking, phishing attacks, malware, and data breaches. Ensuring robust security measures, including encryption, firewalls, and multi-factor authentication, is crucial to mitigate these risks.

Technical Issues: Online ATM management systems may encounter technical issues such as software bugs, system failures, or hardware malfunctions. Regular maintenance and updates are necessary to address these issues and maintain system reliability.

Dependency on Third-party Providers: Online ATM management systems may rely on third-party providers for services such as internet banking platforms, payment gateways, or cloud hosting services. Any disruptions or service outages from these providers can impact the system's availability and performance.

CHAPTER 6: SNAPSHOTS

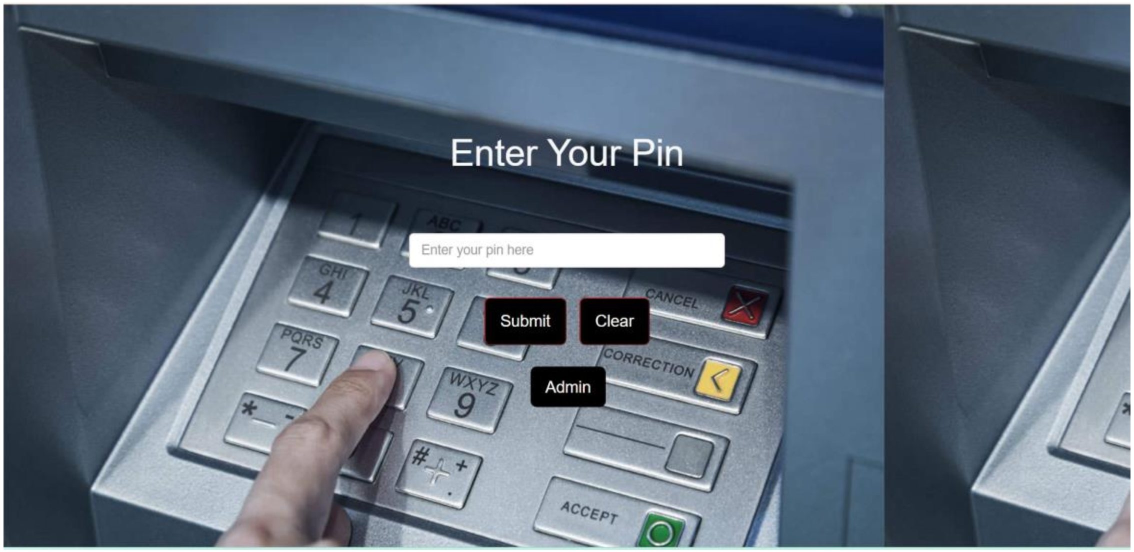
6.WELCOME PAGE



Welcome to CoderBank ATM

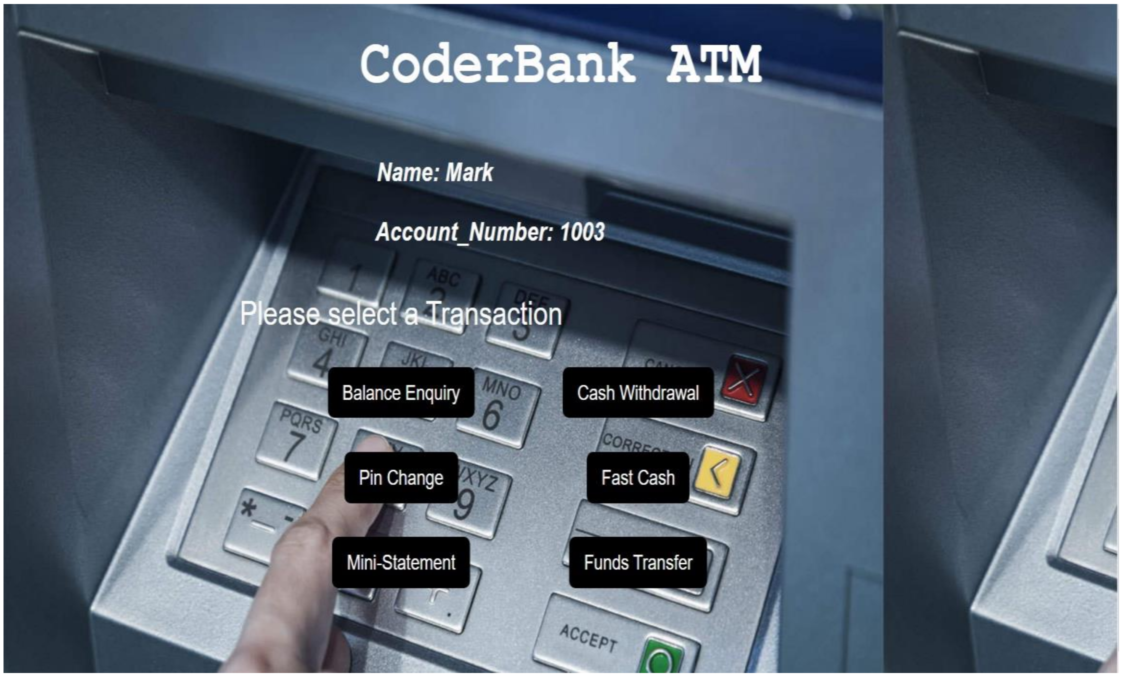
Please Insert Your Card

6.2 LOGIN PAGE



Enter Your Pin

6.3 MENU PAGE



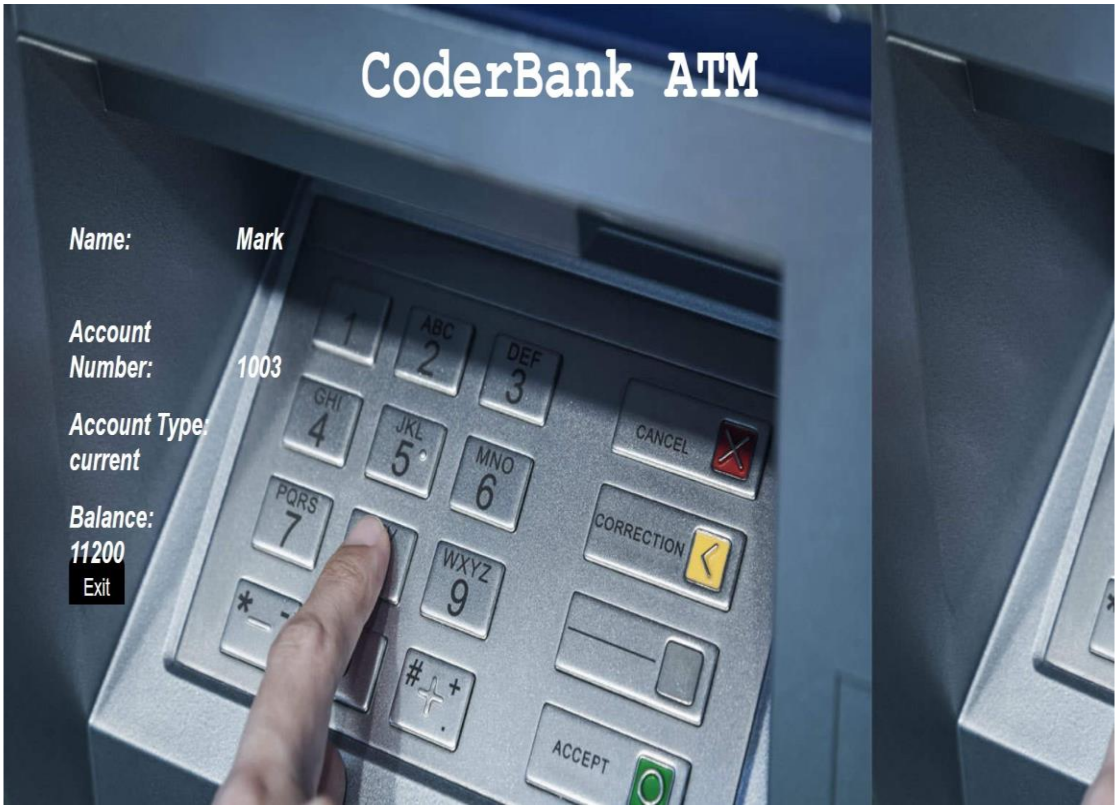
CoderBank ATM

Name: Mark

Account_Number: 1003

Please select a Transaction

6.4 BALANCE ENQUIRY



CoderBank ATM

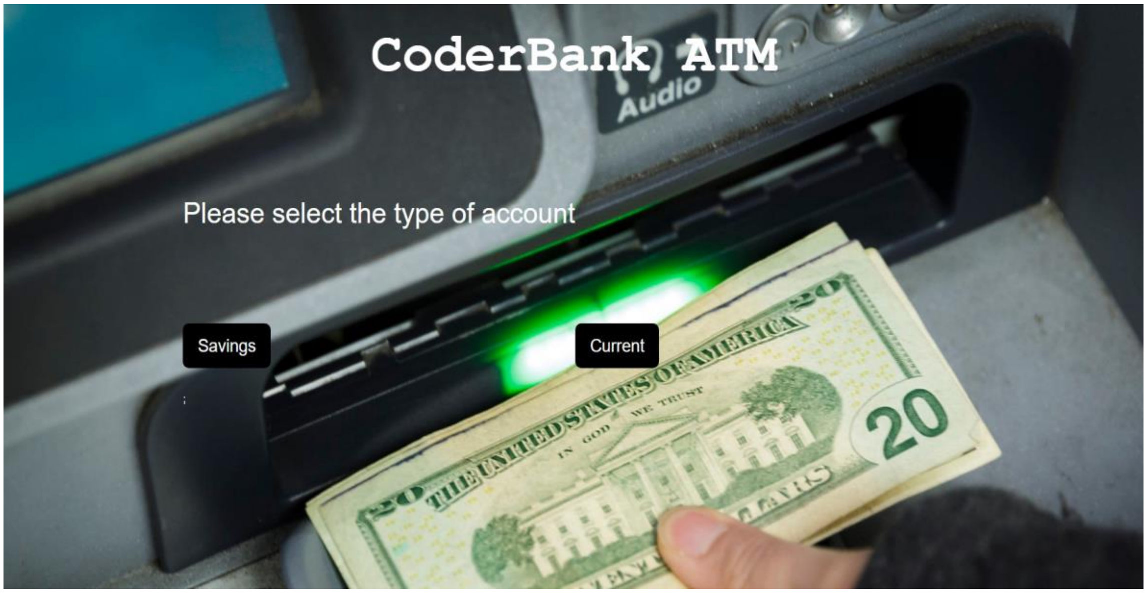
Name: Mark

Account Number: 1003

Account Type: current

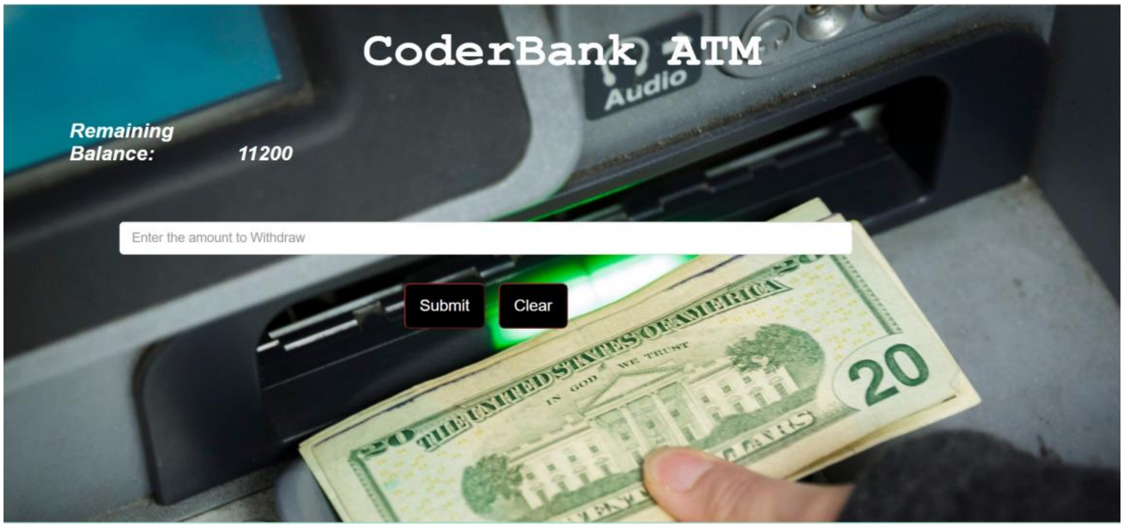
Balance: 11200

6.5 WITHDRAW PAGE



CoderBank ATM

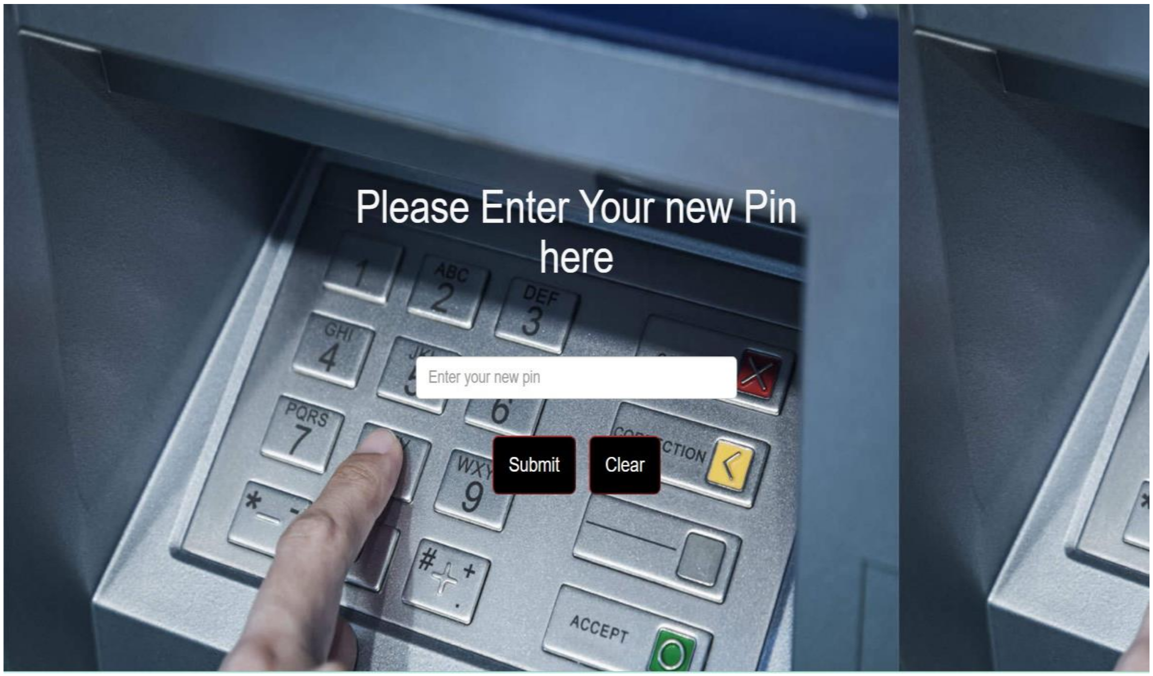
Please select the type of account



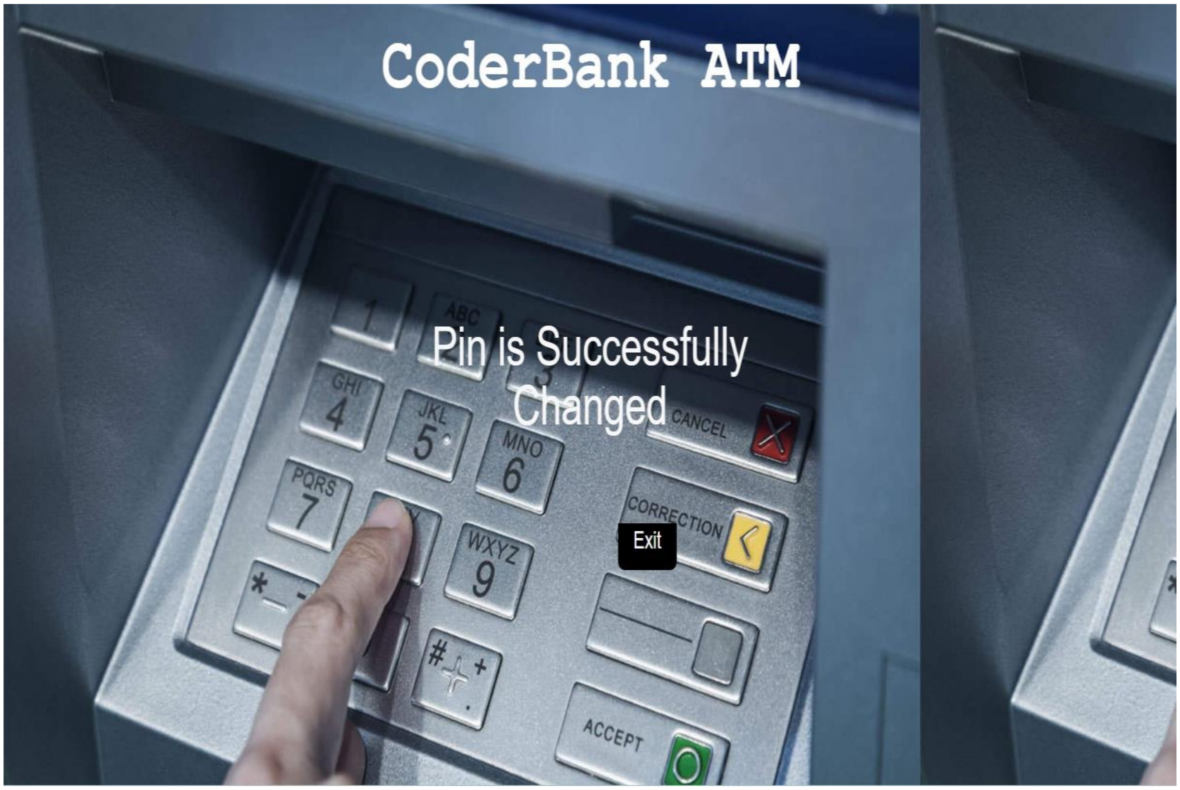
CoderBank ATM

Remaining Balance: 11200

6.6 PIN CHANGE PAGE



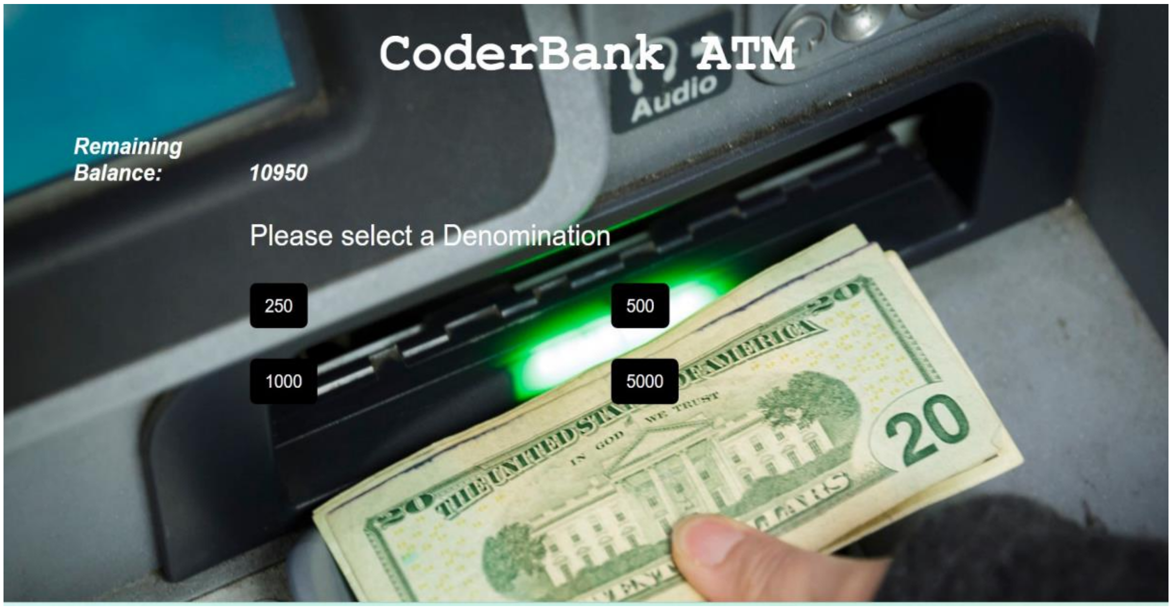
Please Enter Your new Pin here



CoderBank ATM

Pin is Successfully Changed

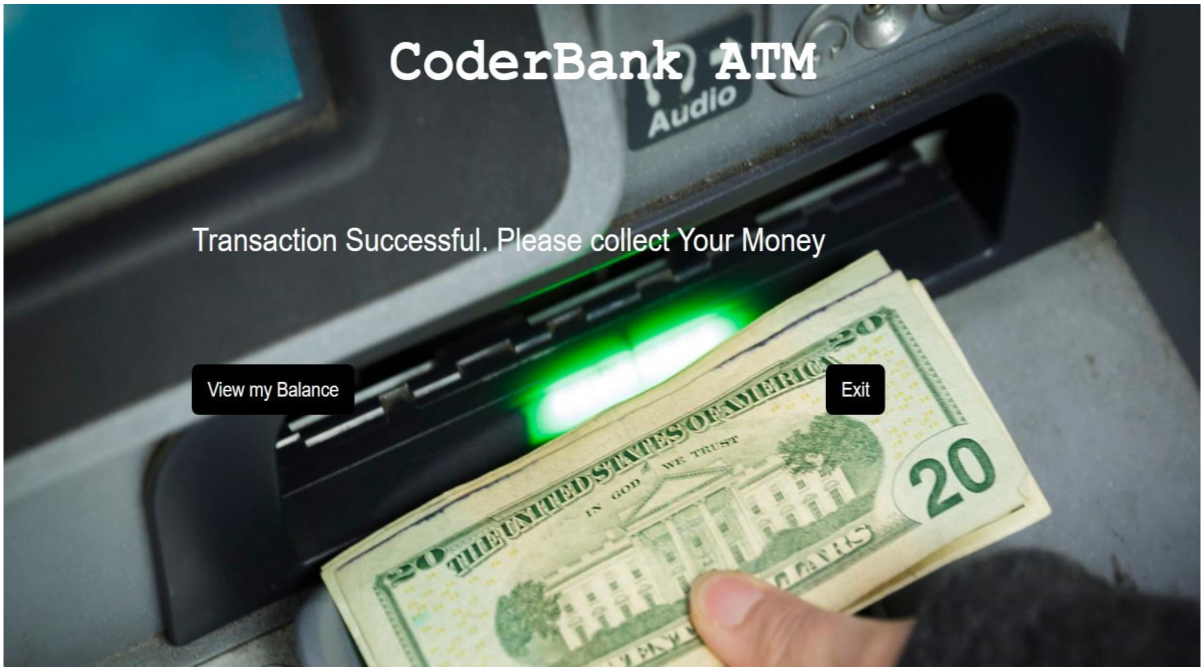
6.7 FAST CASH PAGE



CoderBank ATM

Remaining Balance: *10950*

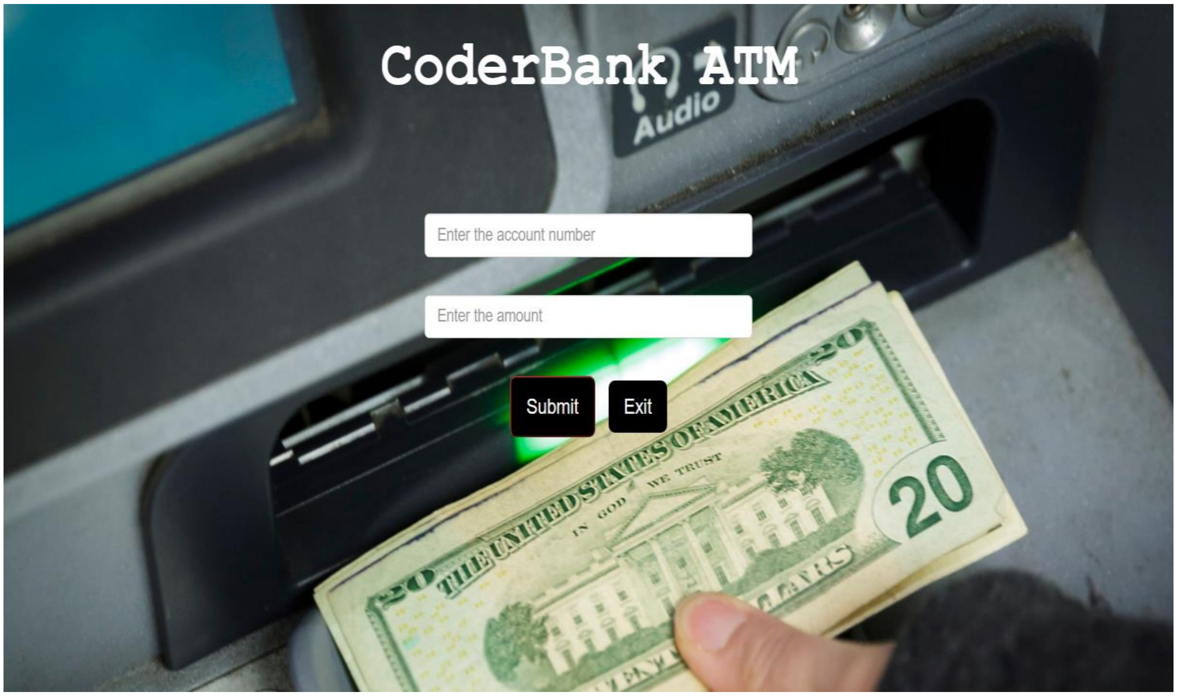
Please select a Denomination



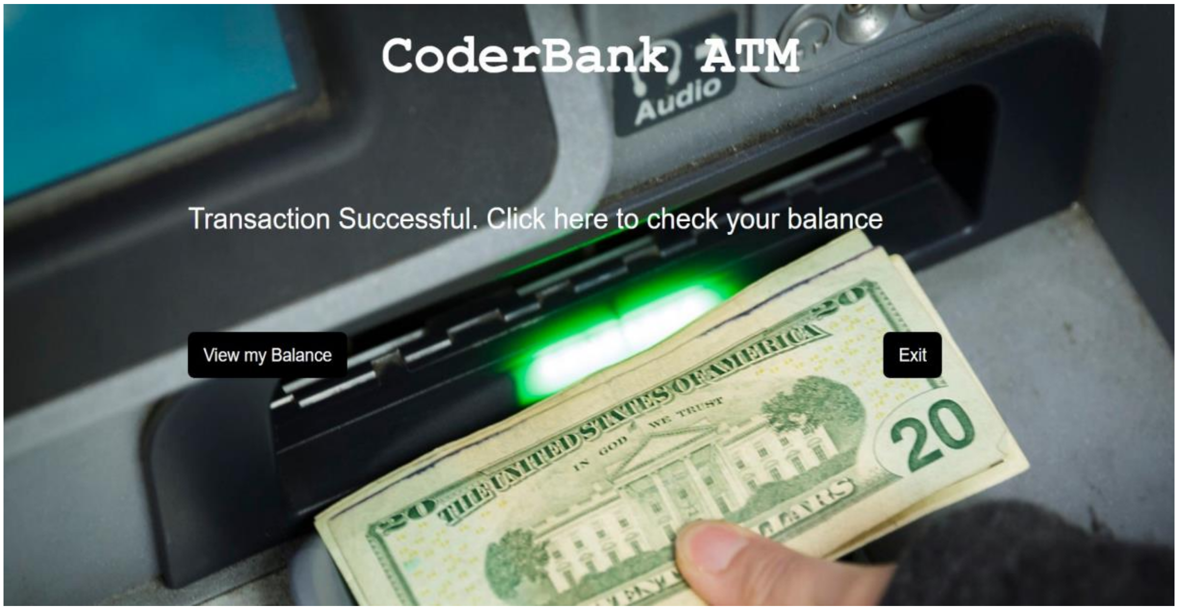
CoderBank ATM

Transaction Successful. Please collect Your Money

6.8 TRANSFER FUNDS PAGE

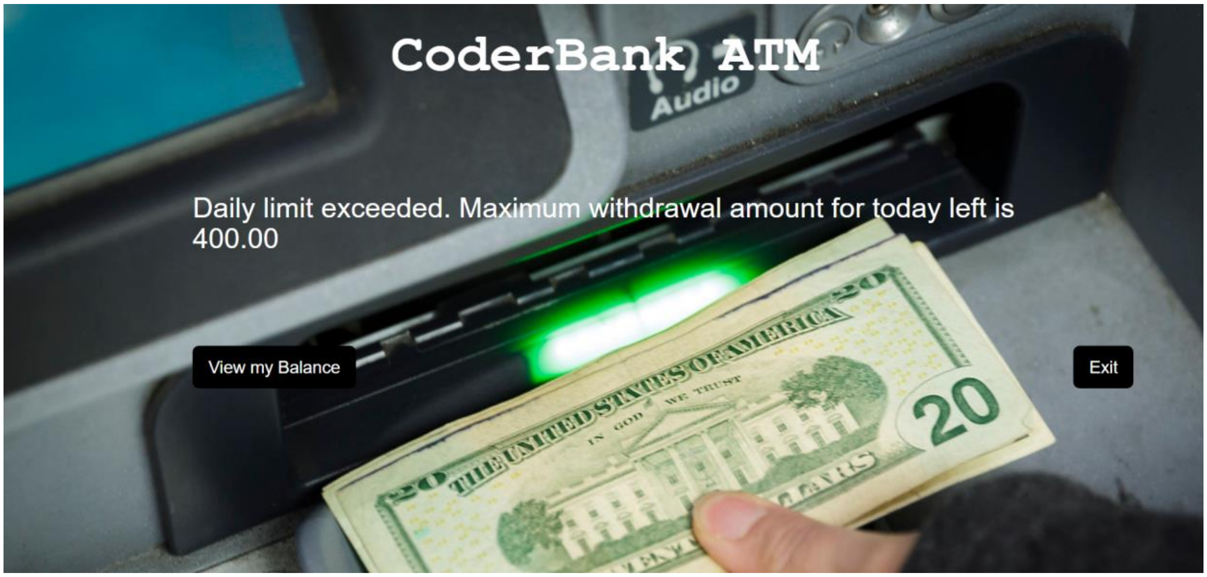


CoderBank ATM



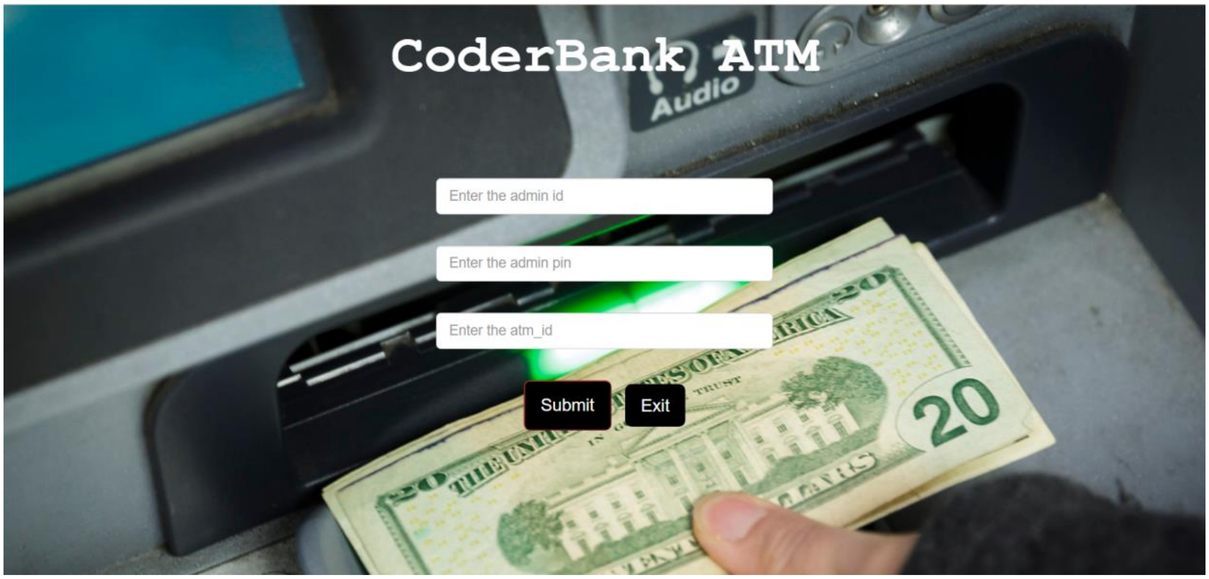
Transaction Successful. Click here to check your balance

6.9 DAILY LIMIT WARNING MESSAGE PAGE

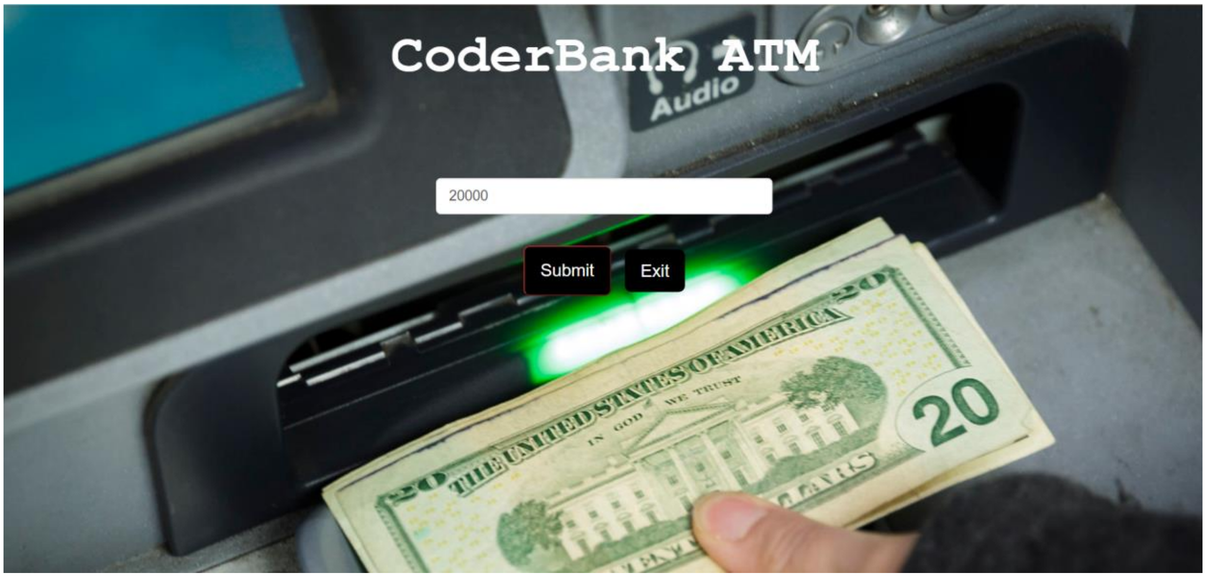


Daily limit exceeded. Maximum withdrawal amount for today left is 400.00

6.10 ADMIN LOGIN

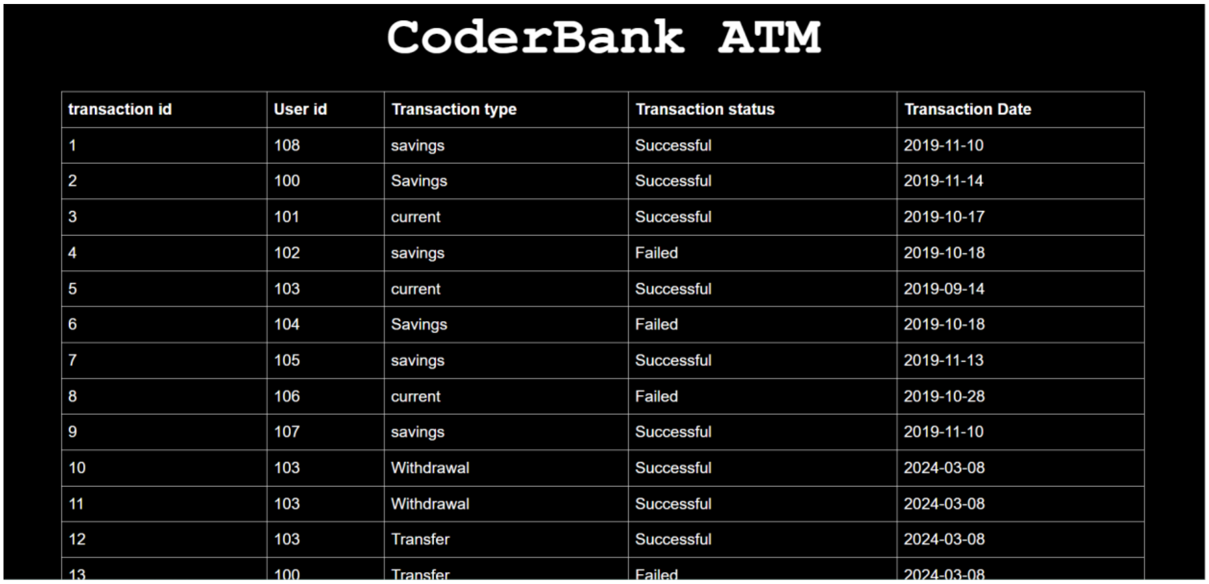


6.11 ADD CASH PAGE



20000

6.12 TRANSACTION SUMMARY



CoderBank ATM

transaction id	User id	Transaction type	Transaction status	Transaction Date
1	108	savings	Successful	2019-11-10
2	100	Savings	Successful	2019-11-14
3	101	current	Successful	2019-10-17
4	102	savings	Failed	2019-10-18
5	103	current	Successful	2019-09-14
6	104	Savings	Failed	2019-10-18
7	105	savings	Successful	2019-11-13
8	106	current	Failed	2019-10-28
9	107	savings	Successful	2019-11-10
10	103	Withdrawal	Successful	2024-03-08
11	103	Withdrawal	Successful	2024-03-08
12	103	Transfer	Successful	2024-03-08
13	100	Transfer	Failed	2024-03-08

CHAPTER 7: CONCLUSION

The online ATM management system represents a significant advancement in banking technology, offering unparalleled convenience, security, and efficiency for financial institutions and customers alike. By centralizing control and monitoring functions through web-based platforms, these systems streamline operations, optimize resource allocation, and enhance the overall user experience. Through features such as real-time transaction monitoring, remote diagnostics, and predictive maintenance capabilities, online ATM management empowers administrators to proactively address issues, minimize downtime, and ensure uninterrupted service delivery. Moreover, the integration of advanced technologies such as biometric authentication, machine learning algorithms, and blockchain-based security protocols further fortifies the system against emerging threats and challenges. As the digital landscape continues to evolve, online ATM management systems will remain instrumental in driving innovation, improving operational efficiency, and delivering superior banking services in the modern era.

CHAPTER 8: FUTURE ENHANCEMENTS

Innovative advancements can revolutionize ATM management systems, enhancing security, convenience, and user experience. Integrating biometric authentication methods like fingerprint recognition, facial recognition, or iris scanning adds an extra layer of security, ensuring reliable user authentication. Mobile wallet integration enables seamless transactions directly from users' mobile devices, eliminating the need for physical cards. AI-powered personalization leverages machine learning algorithms to analyze user behavior, offering personalized recommendations and tailored promotions. Enhanced fraud detection using AI algorithms detects and prevents fraudulent activities in real-time, safeguarding users' finances. Contactless transactions via NFC technology or QR codes provide a hygienic and convenient option for users. Exploring blockchain integration enhances data security and transparency, ensuring secure transaction processing and identity verification. Supporting multiple channels such as web, mobile apps, and social media platforms expands accessibility, while predictive maintenance algorithms ensure optimal ATM performance by proactively addressing hardware issues. These innovations collectively elevate the ATM experience, making it safer, more convenient, and technologically advanced.

REFERENCES

https://www.slideshare.net/khalidbazgamah/atm-project

https://www.bing.com/search?pglt=2211&q=atm+management+system+project&cvid

ATM Management System using html - GeeksforGeeks

Web development text book

DBMS notes