

Elicitation of the Functional and Non-Functional Requirements of ATM Management System

Four Week Industrial Training on Software Development

(June 19, 2023- July 16, 2023)

Submitted by

Danish Naseem, Diploma in Computer Engg. III Year

University Polytechnic

Faculty of Engineering and Technology

Jamia Millia Islamia, New Delhi

A Central University



Division of Communications and Information Technology

Indraprastha Institute of Information Sciences Delhi

(A Unit of Indraprastha Institute of Information Sciences Private Limited, New Delhi, India)

June 2023

Table of Contents

Certificate.....	III
Abstract.....	IV
List of Figures.....	. V
CHAPTER 1	INTRODUCTION
1.1 Introduction.....	06
1.2 Problem Definition.....	07
1.3 Evidence of Problem Definition.....	07
1.4 Proposed Solution.....	07
1.5 Stakeholder Identification.....	07
1.6 Motivation.....	09
1.7 Contribution.....	09
CHAPTER 2	ELICITATION OF REQUIREMENTS
2.1 Introduction.....	10
2.2 Requirements Elicitation	11
2.3 Functional Requirements (FRs).....	12
2.3.1 Non-Functional Requirements (NFRs).....	12
2.3.2 List of all Possible FRs and NFRs of ATM Managements System.....	13
CHAPTER 3	UML MODELING
3.1 Modeling in Software Engineering.....	16
3.2 UML (Unified Modeling Language)	16
3.3 Flow Charts.....	17
3.4 Use Case Diagrams.....	17
3.5 Activity Diagrams.....	18
3.6 Use Case Diagram of ATM Management System.....	18
CHAPTER 4	IMPLEMENTATION
4.1 Introduction.....	21
4.2 Implementation of Login Module.....	21
4.3 Implementation of Account Options Module.....	25
4.4 Implementation of Withdrawal Module.....	28
CHAPTER 5	CONTRIBUTION AND FUTURE WORK
5.1 Contributions.....	31
5.2 Future Work.....	31
5.3 Conclusion.....	31
REFERENCES	27

Certificate

Abstract

The ATM System is the project which is used to access their bank accounts in order to make cash withdrawals, Balance enquiry etc. In this project I tried to show the working of a ATM system and cover the basic functionality of a ATM Management System.

To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also, to enable the user's work space to have additional functionalities which are not provided under a conventional banking project.

The ATM Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for ATM System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software.

This project is developed using HTML, CSS, and PHP languages. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyses the system requirements and then comes up with the requirements specifications. The system is then designed in accordance with specifications to satisfy the requirements.

The system design is then implemented with using HTML, CSS, and PHP languages. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating while the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

The ATM will provide the customer with a printed receipt for each successful transaction, showing the date, time, machine location, type of transaction, account(s), amount, and ending and available balance(s) of the affected account ("to" account for transfers).

List of Figures

Fig 3.2 Overview of Different Types of UML Diagram.....	16
Fig 3.3 Flow chart symbols & functions.....	17
Fig 3.4 Use Case Symbols & Functions.....	17
Fig 3.6 Use Case Diagram of ATM System.....	18
Fig 4.2 (a) User Login Page.....	25
Fig 4.2 (b) Account Page.....	28
Fig 4.2 (c) Withdrawal Page.....	30

CHAPTER 1:

INTRODUCTION

The objective of this chapter is to discuss about ATM Management System, Stakeholder, Identification of Stakeholder. The Motivation behind the work and contribution are also discussed at the end of this Chapter.

1.1 Introduction

Automated Teller Machine enables the clients of a bank to have access to their account without going to the bank. This is achieved only by development the application using online concepts.

When the product is implemented, the user who uses this product will be able to see all the information and services provided by the ATM, when he enters the necessary option and arguments. The product also provides services like request for cheques, deposit cash and other advanced requirement of the user. The data is stored in the database and is retrieved whenever necessary. The implementation needs ATM machine hardware to operate or similar simulated conditions can also be used to successfully use the developed product.

To develop this ATM system the entire operation has been divided into the following step:

1. Verification process
2. language, service, and account selection
3. Banking services
4. Transactions
5. Special services

The program is designed in such a way that the user must have card and pin number. Once verified, he is provided a menu and he/she had to enter the option provided in the menu. For example, when the user wants to view the list of payment history than he/she had to enter the option for payment history provided in the main menu. When the option is entered alone with the respective argument, then the payment history is displayed on the screen. The user also must be given option to browse through the pages like previous page, next page, etc. The user may experience a delay in retrieving or viewing the data, when there are many users logged on to the same bank branch system.

1.2. Problem Definition

The system mainly used by the bank clients. When a client comes to ATM centre to update and delete their account. It reduces the time consumption and lot of paperwork. For any single operation it involves numerous references and updating also takes subsequent changes in other places.

1.3. Evidence of problem definition

Now- a -days every one very busy in their work. So, they feel that the job must be easier so the system is used to reduce their work which is done in the ATM system. Instead of keeping lots of paper into a record or file and it may be missed somewhere so, this system helps to keep the record of the customer it also keeps the details of the customer. It is also easy to access.

1.4. Proposed solution

The system customer transactions, satisfies the requirements of the existing system in full-fledged manner. Through this system, customer can make fast transactions and view the last transactions easily.

1.5 Stakeholder Identification

In software development, a stakeholder refers to any individual or group that has an interest or involvement in a project. Stakeholders can be directly or indirectly affected by the project and can influence its outcome. They play a crucial role in the requirements gathering, decision-making, and overall success of a software development project.

How to identify Stakeholders?

- Who will use or affected by the software?
- Which department will use the software?
- Workflow of the software project?
- Legal restrictions and regulatory policies of the government.
- Any other vital people for the project.
- Who will gain or lose money due to the final product?
- Whose work will be drastically changed by the product?
- Whose career/reputation will be affected by the final product?
- Who has authority to make changes to the development plan?
- Whose support is vital to the success of the product?

In the context of an ATM (Automated Teller Machine) management system, the following are some stakeholders typically involved:

1. **Bank:** The bank or financial institution itself is a central stakeholder. They initiate and sponsor the project, providing the necessary resources, funding, and strategic direction. The success of the project directly impacts their customer experience, operational efficiency, and overall business performance.
2. **ATM Service Providers/Vendors:** Companies that manufacture, install, and maintain ATMs are crucial stakeholders. They collaborate with the bank to integrate their hardware with the ATM Management System, ensuring hardware compatibility, reliability, and optimal performance.
3. **Software Development Team:** The team responsible for designing, developing, and testing the ATM Management System software is a critical stakeholder. They translate the project requirements into a functional software solution that governs the ATM network's operations and services.
4. **IT and Technical Support Teams:** These teams are responsible for the ongoing maintenance, updates, and troubleshooting of the ATM Management System. They ensure that the system remains secure, up-to-date, and operational at all times.
5. **Security Experts:** Given the sensitive nature of financial transactions, security experts play a vital role in designing robust security protocols for the ATM Management System. They safeguard against fraud, hacking attempts, and data breaches to maintain the integrity and confidentiality of customer information.
6. **Regulatory Bodies and Compliance Officers:** In the heavily regulated financial industry, regulatory bodies, and compliance officers ensure that the ATM Management System adheres to industry standards, security regulations, and data protection laws. They ensure that the system meets legal requirements and mitigates risks.
7. **Network Administrators:** These professionals manage the network infrastructure that connects ATMs to the central system. They oversee network reliability, uptime, and connectivity, ensuring seamless communication between ATMs and the management system.
8. **End Users (Customers):** Customers who use ATMs are indirect stakeholders. The efficiency, accessibility, and security of the ATM Management System directly impact their banking experience. A well-functioning system provides a smooth and convenient self-service banking experience.
9. **Internal Stakeholders (Management):** The bank's senior management and executives are crucial stakeholders who provide strategic direction, allocate resources, and make high-level

decisions related to the project. They ensure that the project aligns with the organization's goals and objectives.

10. **Analytics and Reporting Teams:** These teams use data collected from the ATM Management System to generate insights about customer behaviour, transaction trends, and operational efficiency. Their findings can drive strategic decisions and process improvements.
11. **Training and Communication Teams:** These teams ensure that bank staff and customers are educated about the features, functionality, and proper use of the ATM Management System. Clear communication and training materials are essential for a successful rollout.
12. **Regulators and Auditors:** Depending on the jurisdiction, regulatory bodies and auditors may be stakeholders in the ATM management system. They oversee compliance with banking regulations, security standards, and audit the system's operation.

It is important to engage and communicate with stakeholders throughout the software development lifecycle to ensure that their needs are understood, expectations are managed, and the final system meets their requirements. Effective stakeholder management helps in delivering a successful ATM management system that satisfies the needs of all parties involved.

1.6 Motivation

The motivation behind implementing an ATM management system is to provide enhanced convenience and accessibility to customers, streamline banking operations, improve transaction speed and accuracy, strengthen security measures, and ensure compliance with regulatory standards. By deploying such a system, banks can offer self-service options, expand their service reach, optimize operational efficiency, reduce costs, and ultimately deliver a seamless and secure banking experience to their customers.

1.7 Contribution

Implementing an ATM management system contributes to the banking industry by enhancing customer convenience through 24/7 self-service access to banking services. It expands service reach to underserved areas, improves operational efficiency, and reduces costs through automation and optimized processes. The system ensures faster and more accurate transactions, enhances security measures, enables real-time monitoring and maintenance, and integrates seamlessly with core banking systems. By complying with regulations and industry standards, it installs trust and confidence in customers while driving overall efficiency and effectiveness in delivering banking services.

CHAPTER 2: - ELICITATION OF REQUIREMENTS

The Objectives of this Chapter is to discuss about Requirements Elicitation, Functional and Non-functional Requirements of ATM Management System.

2.1 Introduction

Elicitation of requirements is a crucial process in the field of software development, systems engineering, and other project management domains. It refers to the systematic gathering, clarification, and documentation of information about what stakeholders need from a particular system, product, or project. The goal of this process is to understand and define the needs, expectations, constraints, and objectives of the stakeholders accurately.

The requirements elicitation process involves various techniques and methods to gather information from stakeholders, including:

- i. Interviews: Direct discussions with stakeholders to understand their needs, preferences, and concerns.
- ii. Workshops: Group sessions where stakeholders can brainstorm and collaborate to identify requirements collectively.
- iii. Surveys and Questionnaires: Structured forms used to gather information from many stakeholders simultaneously.
- iv. Observation: Analysts observe how users work in their current environment to identify potential requirements.
- v. Prototyping: Building and testing quick prototypes to validate and clarify requirements with stakeholders.
- vi. Document Analysis: Reviewing existing documentation, reports, or records to identify potential requirements.
- vii. Focus Groups: Organized sessions where stakeholders can openly discuss their needs and expectations.
- viii. Use Cases and User Stories: Describing typical interactions and scenarios that users will encounter in the system.
- ix. Brainstorming: Encouraging stakeholders to generate new ideas and requirements.
- x. JAD (Joint Application Development) sessions: Intensive collaborative workshops with key stakeholders and development teams.

Throughout the elicitation process, it is essential to involve various stakeholders, such as end-users, customers, business analysts, domain experts, project managers, and developers. Effective communication and active listening are critical to understanding stakeholders' true needs and ensuring that requirements are captured accurately.

Once the requirements are elicited, they need to be analysed, prioritized, and documented clearly, concise, and unambiguous. This documentation serves as a foundation for the subsequent phases of the project, including design, development, testing, and deployment, ensuring that the final product meets stakeholders' expectations and delivers value to the organization or users.

2.2 Requirements Elicitation

Requirements elicitation for an ATM (Automated Teller Machine) management system involves identifying and documenting the functional and non-functional requirements of the software. Here are some key points to consider during the requirements elicitation process:

1. User Authentication and Authorization Module:
 - a) Responsible for verifying the identity of users through PIN codes, biometrics, or card authentication.
 - b) Manages user access rights and authorizations for various transactions.
2. Transaction Processing Module:
 - a) Handles different types of transactions, such as cash withdrawals, balance inquiries, funds transfers, bill payments, and card reissuance.
 - b) Interacts with the core banking system to validate and process transactions.
3. Cash Management Module:
 - a) Monitors cash levels in ATMs and forecasts cash replenishment requirements.
 - b) Generates alerts for low cash levels and initiates cash replenishment processes.
4. Security and Fraud Prevention Module:
 - a) Implements security measures to prevent unauthorized access, tampering, and fraud.
 - b) Monitors for suspicious activities and triggers alerts for potential security breaches.
5. User Interface Module:
 - a) Provides the graphical user interface (GUI) for customers to interact with the ATM.
 - b) Displays transaction options, prompts, and messages.

6. Card Management Module:
 - a) Manages card issuance, renewal, replacement, and deactivation.
 - b) Links customer accounts to their ATM cards.
7. Compliance and Regulatory Module:
 - a) Ensures that the ATM system adheres to industry standards and regulatory requirements related to data security, privacy, and accessibility.
8. Remote Management and Monitoring Module:
 - a) Allows remote administrators to manage and monitor ATMs without physically visiting each machine.
9. Reporting and Analytics Module:
 - a) Provides insights and analytics on ATM usage patterns, transaction trends, and operational performance.
10. Integration with Core Banking System:
 - a) Interfaces with the bank's core banking system to ensure accurate and up-to-date account information.

Remember that the specific requirements for an ATM management system may vary based on the context, business requirements, and technology stack used. It is essential to prioritize and tailor these requirements to meet the specific needs of the ATM management system you are developing.

2.3 Functional Requirements (FRs)

Functional requirements define what the software system should do. They describe the specific functionalities, behaviours, and operations that the software must perform to meet the user's needs. FRs are typically expressed in terms of specific actions, tasks, or operations that users or system components should be able to perform. Examples of functional requirements include user authentication, data input validation, transaction processing, report generation, and integration with other systems.

2.4 Non-Functional Requirements (NFRs)

Non-functional requirements specify the qualities or attributes that the software system should possess. Unlike functional requirements, which focus on what the system does, non-functional requirements define how the system should perform or behave. NFRs address aspects such as performance, security, reliability, usability, maintainability, scalability, and compliance.

Examples of non-functional requirements include response time, system availability, security measures, user interface design, error handling, and regulatory compliance.

Both functional and non-functional requirements are essential for capturing and documenting the complete set of specifications and expectations for a software system. They help guide the software development process, enable effective communication between stakeholders, and serve as a basis for system design, implementation, and testing activities.

2.5 List of all possible FRs and NFRs of ATM Management System are given below_

Functional Requirements (FRs):

- i. User Authentication and Authorization:
 - a) Users should be able to log in using a unique PIN code or biometric authentication.
 - b) The system should verify user identity and authorize transactions based on account permissions.
- ii. Cash Withdrawal:
 - a) Users should be able to withdraw cash from their accounts.
 - b) The system should validate the withdrawal amount against the account balance and transaction limits.
- iii. Balance Inquiry:
 - a) Users should be able to check the balance of their accounts.
 - b) The system should display the current account balance accurately.
- iv. Funds Transfer:
 - a) Users should be able to transfer funds between their accounts.
 - b) The system should validate the transfer details and update account balances accordingly.
- v. Card Issuance and Replacement:
 - a) Admins should be able to issue new ATM cards or replace existing ones.
 - b) The system should link the new card to the appropriate account and provide activation instructions.
- vi. Transaction History:
 - a) Users should be able to view a history of their recent transactions.
 - b) The system should display transaction details, dates, and locations.

- vii. Receipt Generation:
 - a) The system should generate transaction receipts for users after each successful transaction.
 - b) Receipts should include transaction details, account balances, and timestamps.
- viii. Account Linking (if applicable):
 - a) Users should be able to link multiple accounts to a single ATM card.
 - b) The system should allow users to select the account they wish to transact with.
- ix. Alerts and Notifications:
 - a) Users should receive alerts for successful transactions, low account balances, and other important notifications.
 - b) Admins should receive alerts for low cash levels, technical issues, and security breaches.
- x. Security Measures:
 - a) The system should implement encryption, secure communication protocols, and data protection measures.
 - b) Users' sensitive information and transaction data should be securely stored and transmitted.
- xi. Error Handling:
 - a) The system should provide clear error messages and guidance to users in case of transaction failures or errors.

Non-Functional Requirements (NFRs):

- i. Usability and User Experience:
 - a) The ATM interface should be user-friendly, intuitive, and easy to navigate.
 - b) Response times for user interactions should be fast, providing a seamless user experience.
- ii. Reliability and Availability:
 - a) The ATM system should be available for use by customers 24/7, with minimal downtime for maintenance.
 - b) The system should have a high uptime percentage, ensuring reliable access to banking services.
- iii. Performance and Response Time:
 - a) The system should process transactions, including cash withdrawals and balance inquiries, within a specified response time.

- b) Response times should be consistent even during peak usage periods.
- iv. Security:
 - a) The system should employ robust authentication and encryption mechanisms to ensure the security of user data and transactions.
 - b)** All user data, PINs, and transaction details should be encrypted and stored securely.

CHAPTER 3:

UML MODELING

The Objectives of this Chapter is to discuss about Unified Modelling Language (UML) and Use Case Diagram of ATM Management System.

3.1 Modeling in Software Engineering

In software engineering, modelling refers to the process of creating representations of a software system, its components, behaviours, and interactions using various modelling techniques and notations. These models serve as abstractions that aid in understanding, analysing, and documenting different aspects of the software system throughout its development lifecycle.

3.2 UML (Unified Modeling Language)

UML is a standardized modelling language that provides a set of notations to represent software systems. It offers various diagrams, such as use case diagrams, class diagrams, sequence diagrams, activity diagrams, and state machine diagrams, to depict different aspects of the system's structure, behaviour, and interactions.

UML contains two types are: -

1. Structural Diagrams – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

2. Behaviour Diagrams – Capture dynamic aspects or behaviour of the system.

Behaviour diagrams include: Use Case Diagrams, State Diagrams, Activity

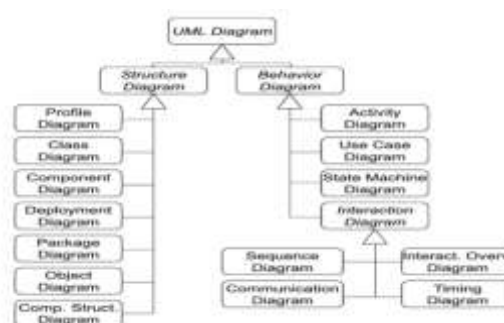


Fig 3.2 Overview of Different Types of UML Diagram

3.3 Flow Chart

A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This is a diagrammatic representation solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Flowcharts are used in analysing, designing, documenting, or managing a process or program in various fields.






Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Fig 3.3 Flow chart symbols & functions

3.4 Use Case Diagrams

Use case diagrams describe the interactions between system components and external actors (users, systems, or devices). They illustrate the functionalities or use cases provided by the system and the actors involved. Use case diagrams help in understanding system requirements and identifying the system's boundaries.


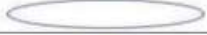
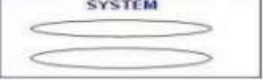
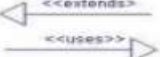
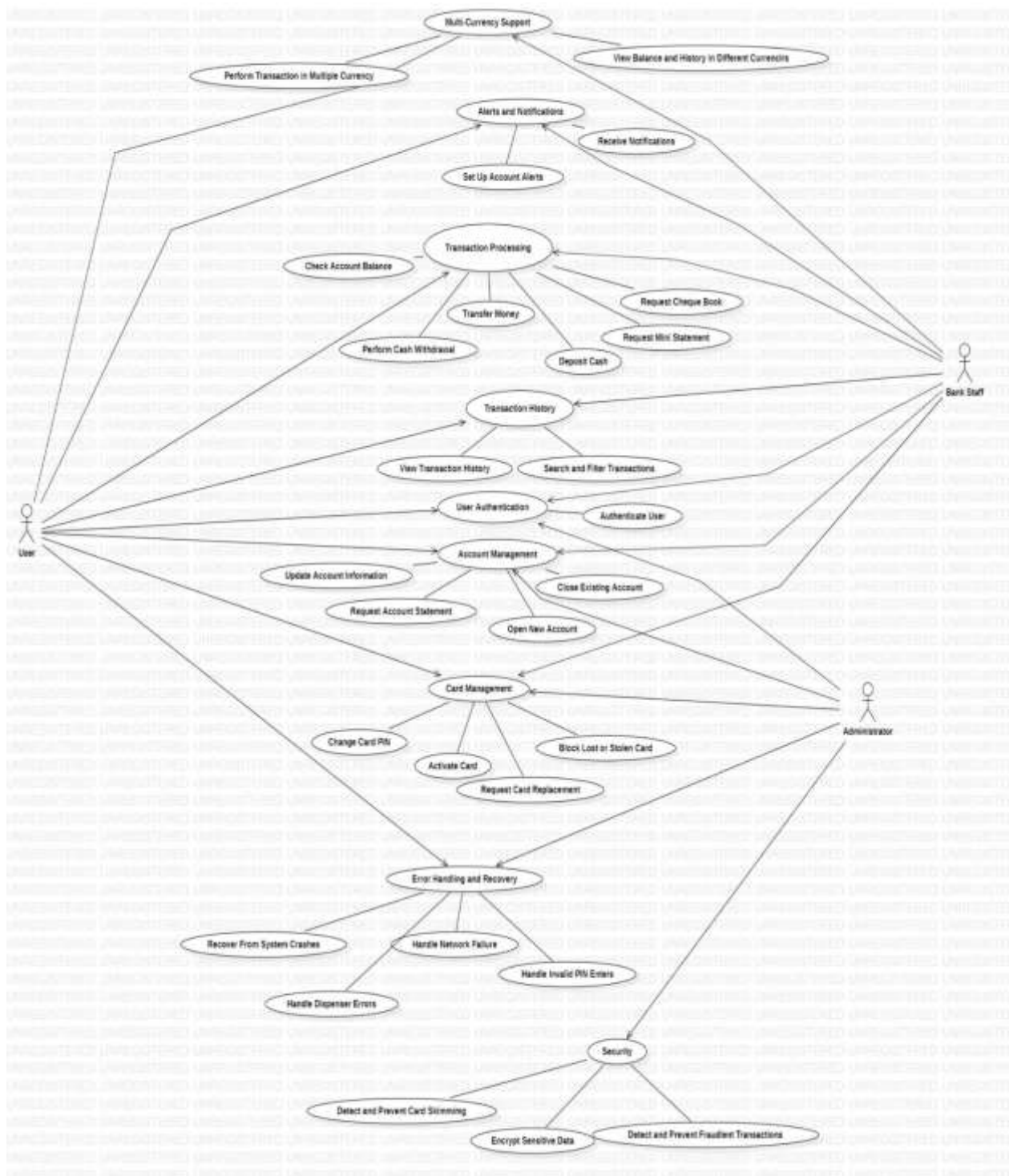
Symbol	Function
	Actor
	Use Case
	System
	Relationship

Fig 3.4 Use Case Symbols& Functions

3.5 Activity Diagrams

Activity diagrams depict the workflow or flow of activities within a system. They show the sequence of actions, decisions, and concurrency in a process. Activity diagrams are often used for business process modelling, system analysis, and high-level system design.

3.6 Use Case Diagram of ATM Management System



Use Case of ATM	
Actor	User, Bank staff, Administrator
Preconditions	<ul style="list-style-type: none"> • The ATM card and the PIN is the only way to access the ATM. • ATMs are more vulnerable to theft and burglary especially in remote areas due to a lack of proper security measures like a bank. • Customers who make online transactions using ATMs are likely to be affected by fraud there is a chance of stealing information from the hacker.
Postconditions	<ul style="list-style-type: none"> • User will login to the system using his card. • User must insert his card and add pin code in order to do transactions. • User can transfer cash to other accounts by entering the account number he wants to transfer. • User can view balance which is available in his respective account. • User can view past transactions.
Flow of events	<ul style="list-style-type: none"> • User inserts his bank card • If the bank card originates from a foreign country, then the ATM requests the user (on screen, in multiple languages) to select a language: <ul style="list-style-type: none"> a) User selects a language by pressing a key: ATM now uses keyboard and screen as input/output. b) User selects a language by speaking the name of the language: ATM detects the language and uses

	<p>voice recognition and synthesis as input/output.</p> <ul style="list-style-type: none"> • ATM requests user to specify the PIN • ATM contacts Bank with the encrypted authentication data, else ATM contacts the Card Consortium • If ATM receives a ‘failed authentication’ response then ATM: <ul style="list-style-type: none"> a) Alerts the user b) Ejects the card • ATM requests user to make a selection from a choice of operations. • If user selects operation by pressing a key then keyboard and screen are used, else if user speaks the operation then voice is used • The ATM receives the response and terminate the transections • ATM goes back to the main menu • User specifies no other transactions • ATM ejects the user’s car.
--	--

Table- 3.1

CHAPTER 4:

IMPLEMENTATION

The objective of this chapter is to discuss about Project Implementation Programming Languages, Database Connectivity and Coding of Login, Account Options and Withdrawal Module.

4.1 Introduction

I have implemented the Transaction Module of ATM System with the help of Html, CSS, JavaScript, and PHP. I use html, css, and js for the Graphical User Interface (GUI) through which user can perform transactions. The term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. PHP is used in the back-end to store the data in the database. It is open-source which means it is free to download and use. It is very simple to learn and use. The files have the extension “.php”.

4.2 Implementation of Login Module

Source code:

- **Login Page**

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>ATM Login</title>
```

```
</head>
```

```
<style>
```

```
  body {
```

```
    font-family: Arial, sans-serif;
```

```
    background-color: #f0f0f0;
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
    height: 100vh;
```

```
margin: 0;
}

.atm-container {
  background-color: #333;
  color: #fff;
  border-radius: 10px;
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.6);
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
}

.atm-screen {
  text-align: center;
  margin: 20px 0;
}

.atm-screen h1 {
  margin-top: 0;
}

button {
  padding: 10px 20px;
  font-size: 18px;
  border: none;
  border-radius: 5px;
  background-color: #007bff;
  color: #fff;
  cursor: pointer;
  width: 100px;
  height: 40px;
}
```

```

button a {
    color: #fff;
    text-decoration: none;
}

button:hover {
    background-color: #0056b3;
}
</style>

<body>
<div class="atm-container">
    <div class="atm-screen">
        <h1>Welcome to Your Bank</h1>
        <p>Please enter your account number and PIN:</p>
        <form action="indexatm.php" method="post">
            <input type="text" name="account_number" placeholder="Account Number"
required>
            <input type="password" name="pin" placeholder="PIN" required>
            <button type="submit"><a href="atm2.html">Login</a></button>
        </form>
    </div>
</div>
</body>
</html>
<?php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "atm";

$conn = new mysqli($servername, $username, $password, $dbname);

```

```

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$userAccounts = [
    ['account_number' => '123456', 'pin' => '1234', 'name' => 'Danish', 'balance' => 5000],
    ['account_number' => '789012', 'pin' => '5678', 'name' => 'Amanat', 'balance' => 7500],
];

// Get the submitted account number and PIN
$enteredAccountNumber = $_POST['account_number'];
$enteredPIN = $_POST['pin'];

// Check if the submitted account number and PIN match any user account
$authenticatedUser = null;
foreach ($userAccounts as $user) {
    if ($user['account_number'] === $enteredAccountNumber && $user['pin'] ===
$enteredPIN) {
        $authenticatedUser = $user;
        break;
    }
}

if ($authenticatedUser !== null) {
    // Authentication successful, redirect to the ATM interface with user data
    session_start();
    $_SESSION['user'] = $authenticatedUser;
    header('Location: atm2.html');
    exit();
} else {
    // Authentication failed, redirect back to the login page with an error message
    header('Location: indexatm.html?error=1');
    exit();
}

```

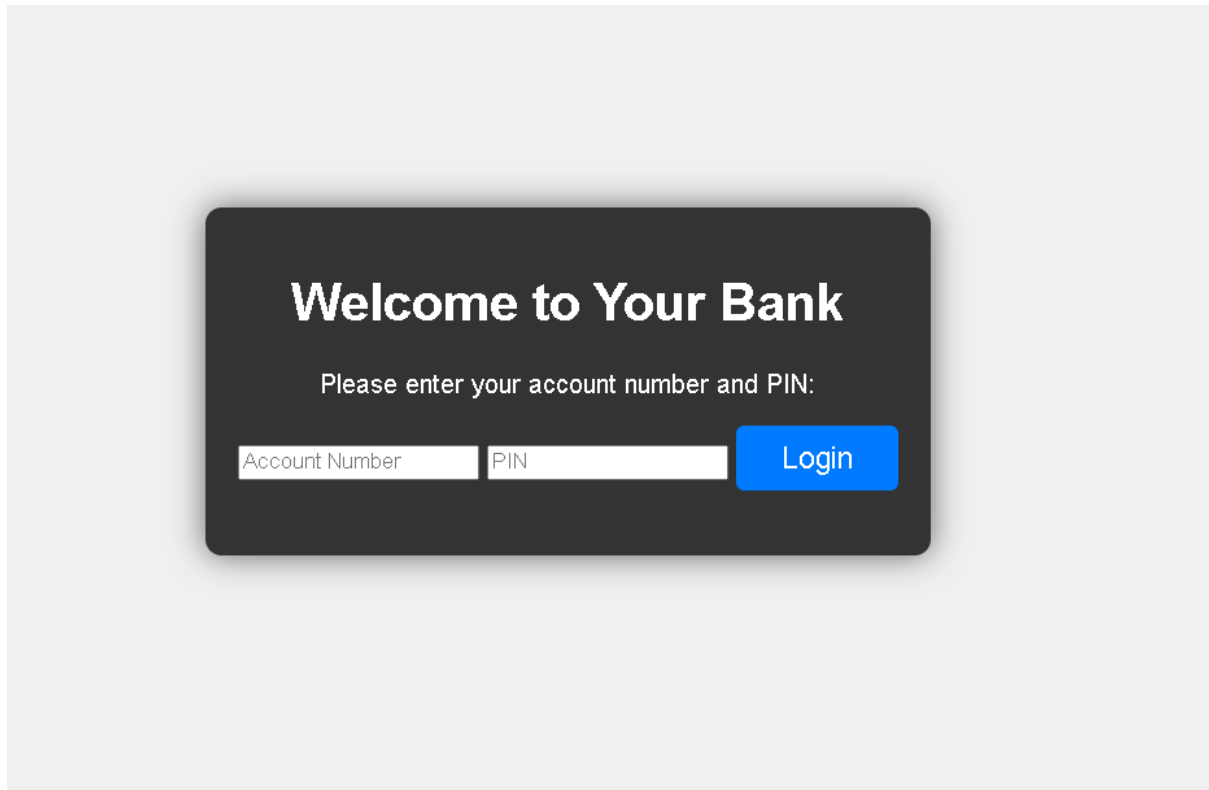



Fig 4.2 (a) User Login

4.3 Implementation of Account Options Module:

- **Account Page**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ATM Interface</title>
  <link rel="stylesheet" href="atm2.css">
</head>
<body>
  <div class="atm-container">
    <div class="atm-screen">
      <h1>Welcome to Your Bank</h1>
      <p>Please select an option:</p>
      <ul class="atm-options">
        <li><button><a href="withdraw1.html">Withdrawal</a></button></li>
        <li><button><a href="balance.php">Balance Inquiry</a></button></li>
        <li><button><a href="pin_change.php">Change PIN</a></button></li>
      </ul>
    </div>
  </div>

```

```

        <li><button><a href="Account.php">Account Details</a></button></li>
        <li><button><a href="Others.php">Others</a></button></li>
        <li><button><a href="generate_pin.php">Generate PIN</a></button></li>
    </ul>
</div>
</div>
</body>
</html>

body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}

.atm-container {
    background-color: #333;
    color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.6);
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 20px;
}

.atm-screen {
    text-align: center;
    margin: 20px 0;
}

```

```
.atm-screen h1 {  
    margin-top: 0;  
}  
  
.atm-options {  
    list-style-type: none;  
    padding: 0;  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: space-between;  
    width: 100%;  
}  
  
.atm-options li {  
    margin-bottom: 10px;  
    width: calc(50% - 10px);  
}  
  
button {  
    padding: 10px 20px;  
    font-size: 18px;  
    border: none;  
    border-radius: 5px;  
    background-color: #007bff;  
    color: #fff;  
    cursor: pointer;  
    width: 130px;  
    height: 60px;  
}  
  
button a {  
    color: #fff;  
    text-decoration: none;  
}
```

```

button:hover {
    background-color: #0056b3;
}

```

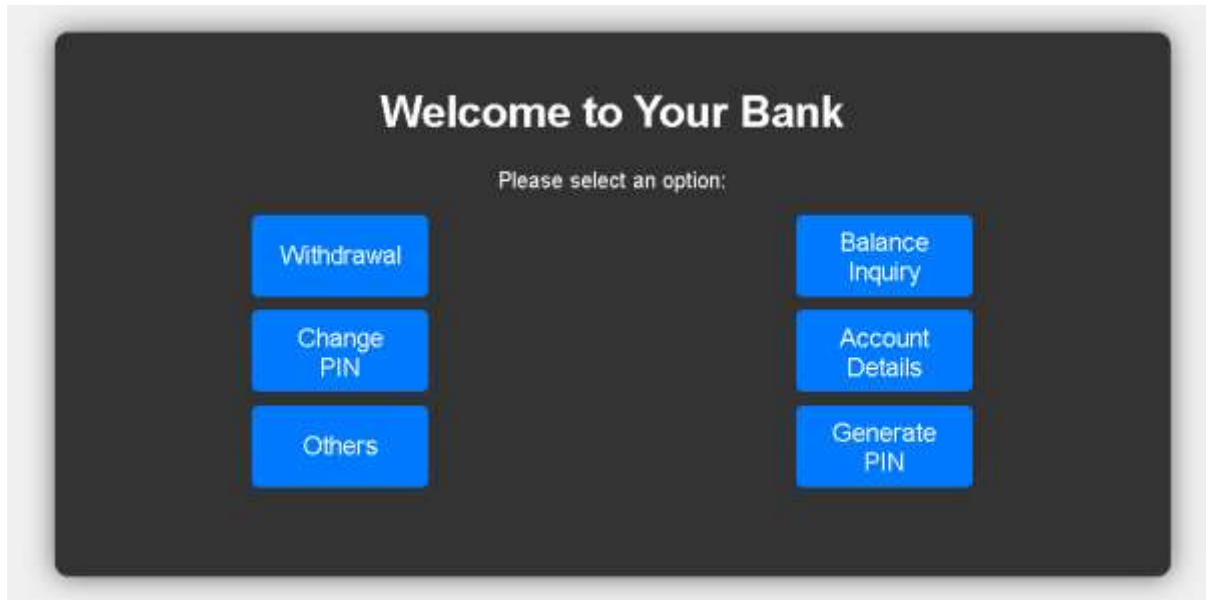


Fig 4.2 (b) Account Options

4.4 Implementation of Withdrawal Module

- **Withdrawal Page**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ATM Interface</title>
    <link rel="stylesheet" href="withdrawal.css">
    <link rel="stylesheet" href="atm2.css">
</head>
<body>
    <div class="atm-container">
        <div class="atm-screen">
            <h1>Welcome to Your Bank</h1>
            <p>Select an amount to Withdraw:</p>
            <form action="withdrawal.php" method="post">

```

```

        <input type="text" name="amount" placeholder="Enter amount" required>
        <button type="submit">Withdraw</button>
    </form>
</div>
<div class="atm-keypad">
    <button><a href="Cancel">Cancel</a></button>
</div>
</div>
</body>
</html>
input[type="text"] {
    width: 100%;
    padding: 10px;
    margin: 10px -10px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

```

```

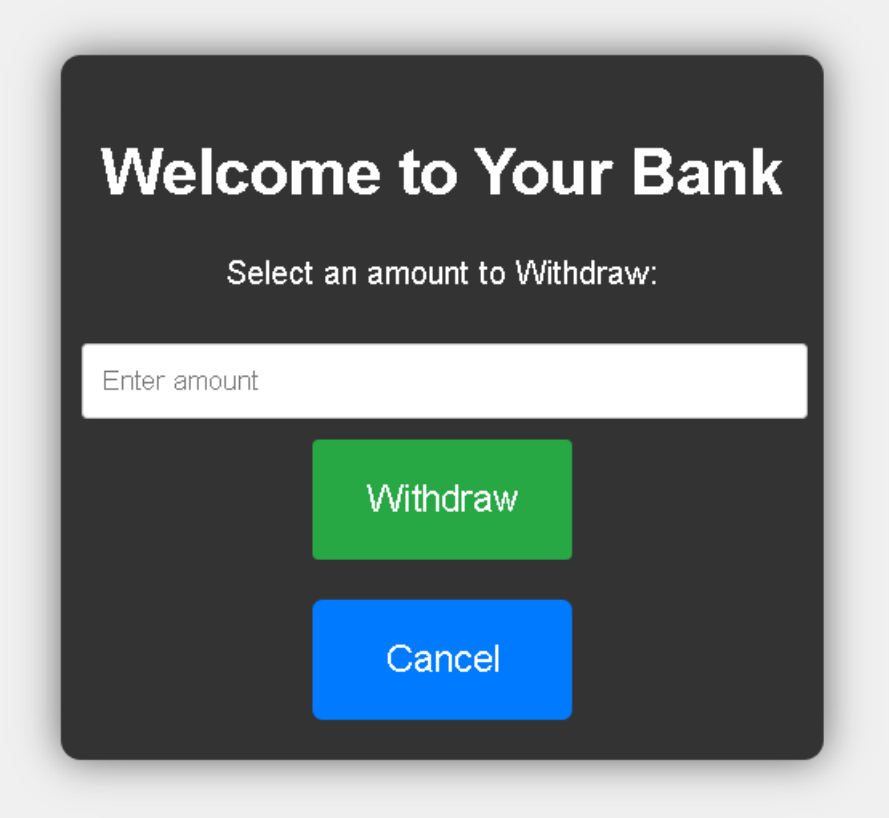
button[type="submit"] {
    padding: 10px 20px;
    background-color: #28a745;
    color: #fff;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

```

```

button[type="submit"]:hover {
    background-color: #218838;
}

```



The image shows a dark-themed user interface for a bank withdrawal. At the top, the text "Welcome to Your Bank" is displayed in a large, bold, white font. Below this, the instruction "Select an amount to Withdraw:" is shown in a smaller white font. A white rectangular input field with the placeholder text "Enter amount" is positioned below the instruction. At the bottom of the interface, there are two buttons: a green button labeled "Withdraw" and a blue button labeled "Cancel", both in white text.

Fig 4.2 (c) Withdrawal Page

CHAPTER 5: CONTRIBUTION AND FUTURE WORK

In this chapter we are going to discuss the Contribution, Future Work, and Conclusion of ATM Management System.

5.1 Contributions

I have analysed some existing systems, websites for the development of the ATM Management System. I have done requirement elicitation and have found a good set of requirements with the help of UML. I have made a use case diagram.

5.2 Future Work

I will continue my research on the proposed system. Since it was not possible to elicitate all the requirements as with time new requirements arise. I will try to find as many new requirements as possible for the development to make a successful software system

5.3 Conclusion

The execution of ATM protection by availing fingerprint also has the traditional verifying methods that were inputting the client's fingerprints, that is sent by the administrator and checked correctly. The protection feature was improved highly for the firmness and solidity of the client's identity. The complete system was constructed on a fingerprint system that makes the mechanism safe, dependable, and effortless to avail. This shall be the most favourable technology in electronic or digital money transactions.

References

- [1] Khalid Saleem, Department of Computer Engineering, Hyderabad Institute of Technology
- [2] Ankit, Department of Computer Science & Application, AIJHM College Rohtak
- [3] <https://www.w3schools.com>
- [4] javapoint.com