

# **Architecture Design Document**

## **ATM Interface in Java (Console Based Application)**

<b>Written By:</b>	Sohel Datta
<b>Document Version:</b>	2.0

**Document Version Control:**

Version	Author	Date	Description
1.0	Sohel Datta	02-01-2025	First Draft
1.5	Sohel Datta	02-01-2025	Revised with feedback
2.0	Sohel Datta	02-01-2025	Final Draft

## Contents

<b>Document Version Control .....</b>	<b>2</b>
<b>Abstract .....</b>	<b>4</b>
<b>1. Introduction .....</b>	<b>5</b>
<b>1.1 What is an Architecture Design Document? .....</b>	<b>5</b>
<b>1.2 Scope .....</b>	<b>5</b>
<b>1.2.1 Definition of the System Architecture .....</b>	<b>5</b>
<b>1.2.2 Deployment strategies .....</b>	<b>6</b>
<b>1.2.3 Proposed solutions to meet business requirements .....</b>	<b>6</b>
<b>1.2.4 User interaction workflows .....</b>	<b>7</b>
<b>1.2.5 Identification of KPIs for performance measurement. ....</b>	<b>7</b>
<b>2. Architecture .....</b>	<b>7</b>
<b>3. Deployment.....</b>	<b>8</b>
<b>4. Proposed Solution .....</b>	<b>8</b>
<b>5. User I/O Workflow .....</b>	<b>8</b>
<b>6. KPI .....</b>	<b>9</b>

## Abstract

This Architecture Design Document outlines the architectural framework and deployment strategy for the proposed system. It provides a comprehensive overview of the architecture, deployment processes, proposed solutions, user input/output workflows, and key performance indicators (KPIs) necessary for measuring success in a structured manner. It provides a concise overview of the proposed system's architectural framework and deployment strategy. It outlines the key components, deployment processes, proposed solutions, user interaction workflows, and the key performance indicators (KPIs).

## 1. Introduction

### 1.1. What is an Architecture Design Document?

An Architecture Design Document (ADD) is a formal document that describes the architecture of a system. It serves as a blueprint for both the development and implementation phases, ensuring that all stakeholders have a clear understanding of the system's structure, components, and interactions.

### 1.2. Scope

The scope of this document includes:

#### 1.2.1. Definition of the system architecture

The system architecture for an ATM interface in a Java console-based application defines the overall structure and components of the system. It typically includes:

- **User Interface:** The console-based interface that allows users to interact with the ATM system, including options for balance inquiry, cash withdrawal, and deposit.
- **Business Logic Layer:** This layer processes user requests, performs validations (e.g., checking account balance), and executes transactions.
- **Data Access Layer:** Responsible for interacting with the database or data storage where user account information and transaction records are stored.
- **Error Handling Mechanism:** Ensures that any issues during transactions are managed gracefully, providing feedback to the user.

### 1.2.2. Deployment strategies

Deployment strategies outline how the ATM application will be set up and made operational. For a console-based application, this may include:

- **Local Deployment:** The application can be run on local machines for testing purposes. Each developer can run their instance to test features.
- **Production Environment:** Once tested, the application may be deployed on a server that interfaces with an actual banking system or a simulated environment for real-time transactions.
- **Updates and Maintenance:** Strategies for updating the application code and maintaining the server environment to ensure uptime and security.

### 1.2.3. Proposed solutions to meet business requirements

The proposed solutions focus on fulfilling the business needs of an ATM interface, such as:

- **User Authentication:** Implementing secure login mechanisms to verify user identity through PINs.
- **Transaction Processing:** Providing functionalities for various transactions like withdrawals, deposits, and balance inquiries.
- **User Feedback:** Ensuring that users receive clear messages regarding transaction success or failure, as well as prompts for further actions.

#### 1.2.4. User interaction workflows

User interaction workflows describe how users will navigate through the ATM interface:

- **Login Process:** Users enter their PIN to access their accounts.
- **Main Menu Navigation:** After logging in, users are presented with options (e.g., Withdraw Cash, Check Balance).
- **Transaction Execution:** Users select an option, follow prompts (e.g., entering withdrawal amount), and receive confirmation or error messages based on their actions.
- **Logout Process:** Users can log out after completing their transactions to ensure security.

#### 1.2.5. Identification of KPIs for performance measurement.

Key Performance Indicators (KPIs) help measure the effectiveness of the ATM interface:

- **Transaction Success Rate:** The percentage of successful transactions compared to total attempts.
- **Average Transaction Time:** The average time taken to complete a transaction from start to finish.
- **User Satisfaction Score:** Feedback collected from users regarding their experience using the ATM interface.
- **Error Rate:** The frequency of errors encountered during transactions, which helps identify areas needing improvement.

## 2. Architecture

The architecture section details the overall structure of the system, including:

- **Components:** Overview of all major components and their interactions.
- **Technologies:** Description of technologies used (e.g., databases, frameworks).
- **Diagrams:** Visual representations (e.g., UML diagrams) illustrating the architecture.

### 3. Deployment

This section outlines the deployment strategy, including:

- **Environment Setup:** Description of development, testing, and production environments.
- **Deployment Process:** Steps for deploying the application, including any automation tools or scripts used.
- **Rollback Procedures:** Strategies for reverting to previous versions in case of deployment failures.

### 4. Proposed Solutions

This section presents the proposed solution to address the identified business problems:

- **Overview:** A high-level description of the solution.
- **Features:** Key features that will be implemented.
- **Benefits:** Explanation of how the solution meets business needs and improves efficiency.

### 5. User I/O Workflow

The User Input/Output workflow outlines how users will interact with the system:

- **User Roles:** Description of different user roles and their permissions.
- **Workflow Diagrams:** Visual representation of user interactions with the system.
- **Input Methods:** Types of inputs users can provide (e.g., forms, APIs).



## 6. KPI

Key Performance Indicators (KPIs) are defined to measure the success of the system:

- Usage Metrics: Number of active users, frequency of use.
- Performance Metrics: System response times, error rates.
- Business Impact Metrics: Revenue generated, customer satisfaction ratings.

This document serves as a foundational reference for stakeholders involved in the project and will be updated as necessary throughout the project lifecycle.