

coachGrasin

CPD

Software Requirements Specification – BANKING MANAGEMENT SYSTEM application

**VERSION 1.0**

APPROVAL SIGNATURES

| CONTRACTOR | | DATE |
| --- | --- | --- |
| <<Deliverable Owner>>  <<John Doe, Manager>> | <<Signature>> |  |
| <<Vendor Project Manager>> | <<Signature>> |  |
| <<Vendor Managing Director>> | <<Signature>> |  |

| **PROJECT MANAGER** | | | | |
| --- | --- | --- | --- | --- |
| Date: | 12.11.2024 | To: | **Nisarg Nirmalkumar,** Project Manager | |
|  | I approve this deliverable and have no further questions or comments. | | | |
|  | I approve this deliverable conditionally, contingent on the review and approval of the following corrections (see comments). | | | |
|  | I reject this deliverable for the following reasons identified (see comments). | | | |
|  | | | | |
| **<<SIGNATURE>>** | | | | **<<DATE>>** |
| Comments | | | | |
|  | | | | |

DOCUMENT HISTORY

| **DOCUMENT APPROVAL HISTORY** | |
| --- | --- |
| Prepared By |  |
| Reviewed By |  |
| Approved By |  |

| **DOCUMENT REVISION HISTORY** | | | |
| --- | --- | --- | --- |
| Date | Document Version | Revision Description | Author |
| 12/11/2024 | 1.0 | Initial Version |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

TABLE OF CONTENTS

1. Introduction 1

1.1. Purpose of this Document 1

1.2. Scope of this Document 1

1.3. Overview 1

2. GenERAL Description 1

3. FUNCTIONAL REQUIREMENTS 2

3.1. ACCOUNT MANAGEMENT 2

3.1.1. CREATING AN ACCOUNT……………………………………………………………………………………………2

3.1.2. BALANCE SEARCH 2

3.1.3. TRANSFER FUNDS…………………………………………………………………………………………..2

4. INTERFACE REQUIREMENTS 3

4.1. UI REQUIREMENTS 3

5. PERFORMANCE REQUIREMENTS 3

5.1. ReSPONSE TIME 3

5.2. SYSTEM RESOURCE UTILIZATION 3

5.2.1. SYSTEM UPTIME 4

6. DESIGN CONSTRAINTS 4

7. NON-FUNCTIONAL ATTRIBUTES 4

7.1. SECURITY 4

7.2. RELAIBILITY 4

7.3. USABILITY 4

7.4. MAINTAINABILITY…………………………………………………………………………………………5

7.4. SCALABILITY…………………………………………………………………………………………………….5

APPENDIX A. Overview of Requirements Traceability A-1

APPENDIX B. Example Requirements Tagging, Attributes, and Relationships B-1

APPENDIX C. Traceability Metrics and Reporting C-1

APPENDIX D. COTS, MOTS, AND CUSTOM CONSIDERATIONS D-1

# Introduction

## Purpose of this Document

The purpose of this Requirements Specification document is to clearly define and outline the functional and non-functional requirements for the development of the "Banking Management System" software application. This document serves as a foundation for both the development and the validation processes, ensuring that all stakeholders, including developers, testers, and end-users, have a shared understanding of the project's objectives. By detailing the specific requirements, expectations, and constraints of the system, this document aims to minimize any ambiguity and set a clear roadmap for the design, development, and implementation of the application. It will also act as a reference point throughout the project's lifecycle to ensure that the delivered product aligns with the defined goals.

## Scope of this Document

This document covers all essential details needed to develop the Bank Management System application, including functionalities such as account creation, balance inquiry, fund transfer, account closure, and user login. The scope includes system architecture, integration, performance, security requirements, data management, and user accessibility. Additionally, it outlines the development timeline, resource allocation, and costs. This document ensures that the project stays within budget, meets deadlines, and satisfies end-user needs. It provides a clear framework for understanding the project's goals, deliverables, and expected outcomes.

## Overview

The Bank Management System is a secure platform that enables bank employees to manage customer accounts efficiently. With an intuitive user interface, employees can create accounts, view balances, transfer funds, and close accounts. The application is designed for quick and secure operations, aiming to streamline the bank’s day-to-day operations while ensuring data security and integrity.

# GenERAL Description

The Bank Management System application provides a centralized platform where bank employees can perform essential banking functions such as account creation, balance checks, money transfers, and account closures. The application focuses on user-friendliness, security, and efficiency, ensuring employees can manage accounts easily while maintaining customer data protection and system reliability.

Key features of the **Bank Management system** include:

* **Account Management**: Users can easily create, view, update, and can close the account.
* **Fund Transfers**: Securely transfer funds between accounts.
* **Balance Inquiry**: Check account balances.
* **Performance**: Quick responses to ensure a smooth experience even under heavy use.

# FUNCTional Requirements

## account Management

### creating an account

* Employees can create a new account by entering details such as:
* Customer Full Name
* Account Type
* Deposit
* The system should verify required fields and ensure data formats are correct (e.g., name characters, positive deposit amount).
* The validated data should be securely stored in the database.

### balance Search

* Employees can check the balance of an existing account.
* The system must retrieve the account balance in real-time from the database.

### transfer funds

* Employees can transfer money between accounts by entering:
* Source Account Number
* Destination Account Number
* Amount
* The system should validate sufficient balance in the source account before initiating a transfer.
* Both accounts should be updated immediately, and a transaction record should be stored in the database.

### close account

* Employees can close a customer account.
* The system should require confirmation for account closure to avoid accidental deletion.
* Account closure must ensure all pending transactions are completed, and account data is archived for record-keeping.

# Interface Requirements

## User Interface (UI) Requirements

### General UI Layout

* The application should have a clear and intuitive layout with options for creating accounts, viewing balances, performing fund transfers, and closing accounts.
* The main screen should display recent transactions and a quick-access menu for common tasks.
* The design should prioritize ease of use and ensure minimal steps for common tasks.

### system interfaces

* The application should integrate with the bank’s backend database for all CRUD (Create, Read, Update, Delete) operations.
* All transactions should be logged and stored for auditing and record-keeping.

# Performance Requirements

## Response time

* **Login**: The system must authenticate users within 2 seconds.
* Account Creation, Fund Transfer, Balance Check, and Account Closure: Each of these actions should complete within 1 second.

## System Resource Utilization

* CPU usage should not exceed 25% during normal operations.
* Memory usage must be maintained within 100 MB for handling up to 10,000 accounts.

## SYSTEM UPTIME

* The application should have a 99.9% uptime over a rolling 30-day period, ensuring consistent availability for employees.

# Design Constraints

## Platform and Device Compatibility

* The application is intended for desktop use and is compatible only with or higher operating systems.
* The software should operate on both Intel and AMD processors.

## Data storage and management

* The system will use a relational database for data storage, supporting CRUD operations and ensuring data integrity.

## security requirements

* All user data, including account details and transactions, must be encrypted.
* The application must adhere to data privacy regulations such as GDPR and CCPA.

# Non-functional attributes

## Security

* **Data Protection**: Ensure data encryption for all stored and transmitted data.
* **Access Control**: Only authorized personnel can access the system.
* **Incident Response**: The system should detect unauthorized access attempts and notify administrators.

## Reliability

* **Data Integrity**: The system must ensure data consistency, especially during concurrent transactions.

## Usability

* The application should be intuitive, requiring minimal training for employees to use.
* All major tasks should require no more than three steps.

## Maintainability

* **Modularity**: The system should be built in a modular fashion to allow easy updates or feature additions.
* **Code Quality**: Ensure clean, well-documented code following coding standards.

## Scalability

* System should support horizontal scalability to handle an increasing number of accounts and transactions.