High Level Document (HLD)

Banking Application in Java(Console Based Application)

Document Version Control

Date Issued	Version	Description	Author
18/07/2023	1	Initial HLD - V1.0	Rashi Jain

Contents

Abstract

In the era of internet, most of the people all over the world completed their transaction on internet.

Though the user of electronic transaction or E-money transaction system increase rapidly.

The growth in online transactions has resulted in a greater demand .

Banking application allow you to access your banking accounts and conduct financial transactions using a mobile device, anytime and from anywhere.

1. Introduction

1.1. Why this High Level Design Document?

The purpose of this High Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding.

The HLD will:

- present all of the design aspects and define them in detail
- describe the user interface being implemented
- describe the hardware and software interfaces
- describe the performance requirements
- include design features and the architecture of the project
- list and describe the non-functional attributes like:
  - security
  - reliability
  - maintainability
  - portability
  - reusability
  - application compatibility
  - resource utilization
  - serviceability

1.2. Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2. General Description

2.1 Product perspective

Banking Application allows customer to have direct access to their financial information and undertake financial transactions without the hassle of going to the bank.

2.2 Problem Statement

Take input from the user using Scanner class, basics about string ,how to print in java, variables, if/else statements, methods, loops, etc. In simple banking application will be coding the simple bank operations like check balance, deposit, withdraw, and exit.

2.3 Proposed Solution

The solution proposed here is a banking application which is console based. It stores the information of various Account holders and manages all the transactions efficiently. It is easy to understand.

2.4 Technical Requirement

- Computer System

2.5 Tools required

IntelliJ IDEA

- IDE is one of the most important tools for Java programmers. There are several IDEs (Integrated Development Environments) that are widely used by Java developers such as NetBeans or Eclipse.

- IntelliJ IDEA is also an IDE which has many good features that could help Java developers create robust Java application. Powered by deep intelligence, it indexes your source code and provides relevant suggestions to the users in every context.

- Besides, it has all crucial tools such as integrated version control systems, frameworks, and supported programming languages that boost the productivity of Java developers.

2.6 Constraints

The banking application is user-friendly. Users should not be required to know the internal working of the application.

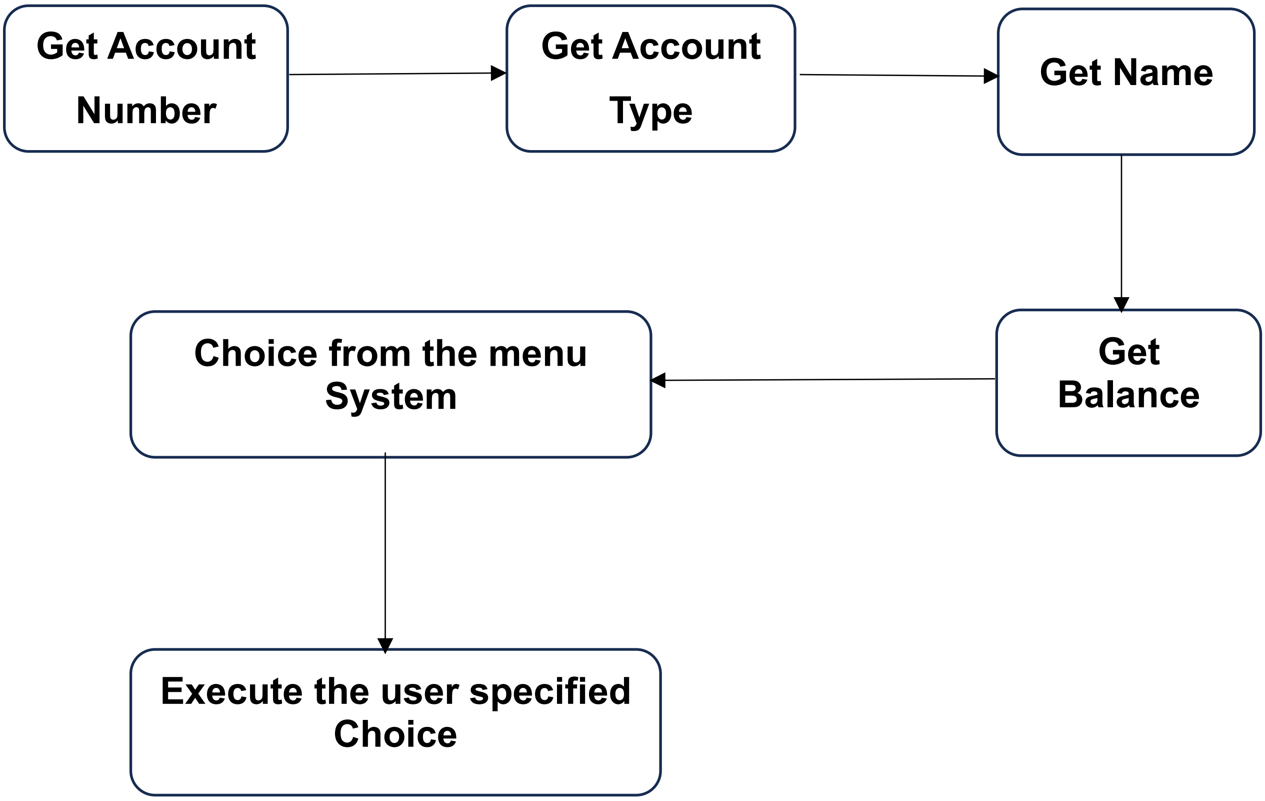
2.7 Assumptions

The main objective of the project is to implement the use cases as previously mentioned(problem statement).It is assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

3. Design Details

3.1 Process Flow

A process flow is a series of steps you and your team go through to achieve your goal.


Get Account Number
Get Account Type
Get Name
Get Balance
Choice from the menu System
Execute the user specified Choice

3.2 Error Handling

Error handling is the process of responding to and recovering from error conditions in your program. Swift provides first-class support for throwing, catching, propagating, and manipulating recoverable errors at runtime. Some operations aren't guaranteed to always complete execution or produce a useful output.

3.3 Performance

Even a single account in the banking application holds several records are at a time. Since reliability and scalability are a major requirement, banking applications need to go through extensive performance testing. It must take necessary actions. Also, model retraining is very important to improve the performance.

3.4 Reusability

Code reusability is the capacity to repurpose pre-existing code when developing new software applications.

4. Conclusion

With passing time, banking activities have become widespread, and today, banks propose different other services. The importance of bank can't be denied at any point. Though banks do several things, their chief job is taking funds that are known as deposits. A core banking system is recognized as the back-end system which processes regular banking transactions. Additionally, it posts updates to different financial records and accounts.

In Conclusion, banking holds a crucial role in our day-to-day life. We must adhere to the banking system as responsible citizens. The banking system acts as a crucial base for the financial system as well as the entire economic system of the country. It provides a base to the market and the companies.