Rams Bank International

Automated Teller Machine (ATM) Software

Architecture Document Specifications

Version 1.0.0
March 12, 2016

TEAM
Otto Barbalho
Tolaz Hewa
Markus Jeeger
Baheer Kamal
Mikhail Korchevskiy
Alessandro Profenna

Table of Contents

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to specify the architectural specifications and model of RBI's Automated Teller Machine (ATM) system.

1.2 Overview

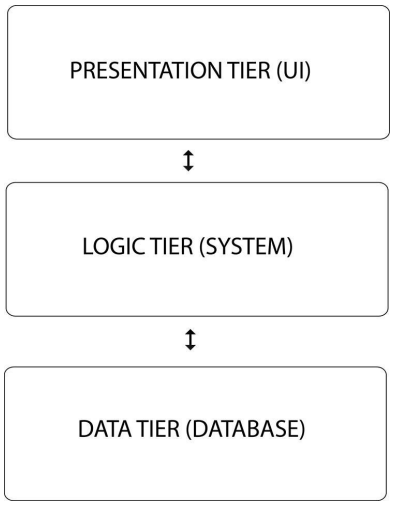
The remainder of this document is structured in three parts. The first part, Section 2, depicts the architectural style chosen for this software. Section 3 outlines the components that are to be built for this software. This will be shown through the use of a class diagram and package diagram, with details of the interconnectedness of the classes and interfaces. Lastly, Section 4 will describe how the components of this architecture relate to the functional requirements outlined in the Requirements Document (Feb. 26, 2016).

2. ARCHITECTURAL STYLE

2.1 Architecture Overview

This ATM software will follow the structure of a 3-tier layered, object-oriented architectural style. This is the best way to represent the required components. The user-interface components are what the user interacts with, so they are part of the top layer. These components interact with the next layer, which involves all account functionality. This second layer communicates with our database, which is the bottom layer of this architecture.

2.2 Architecture Diagram



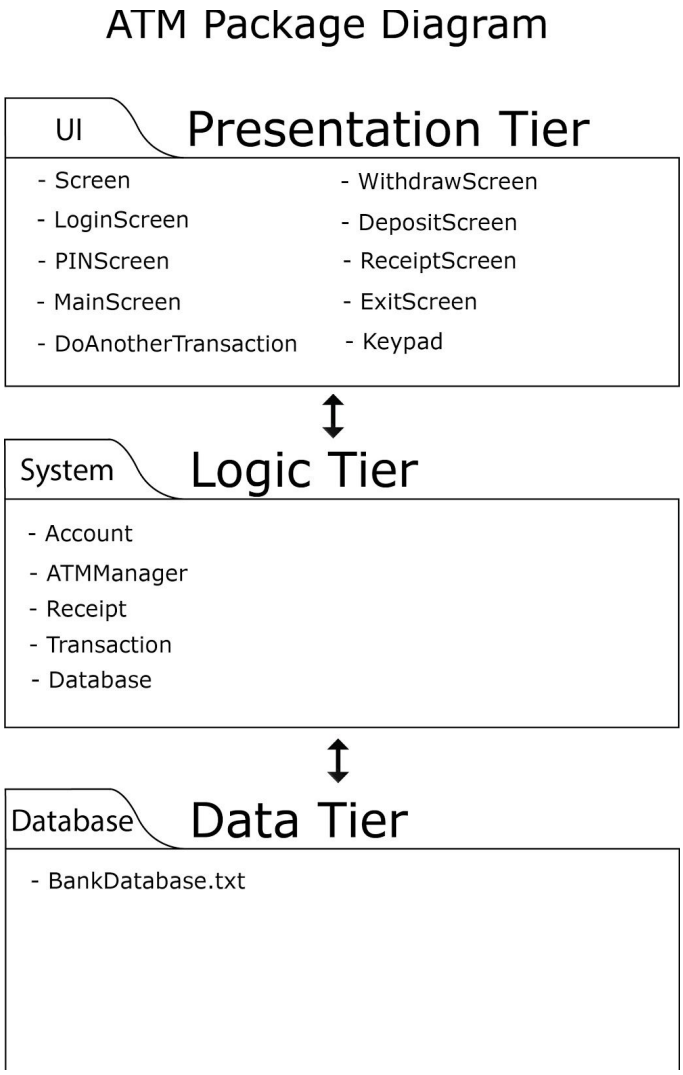
3. DESIGN COMPONENTS

3.1 Components Overview

The components that make up this software involve the design of three packages, two interfaces, and fourteen classes. The components are interconnected in a way that best represents the functional need of the software.

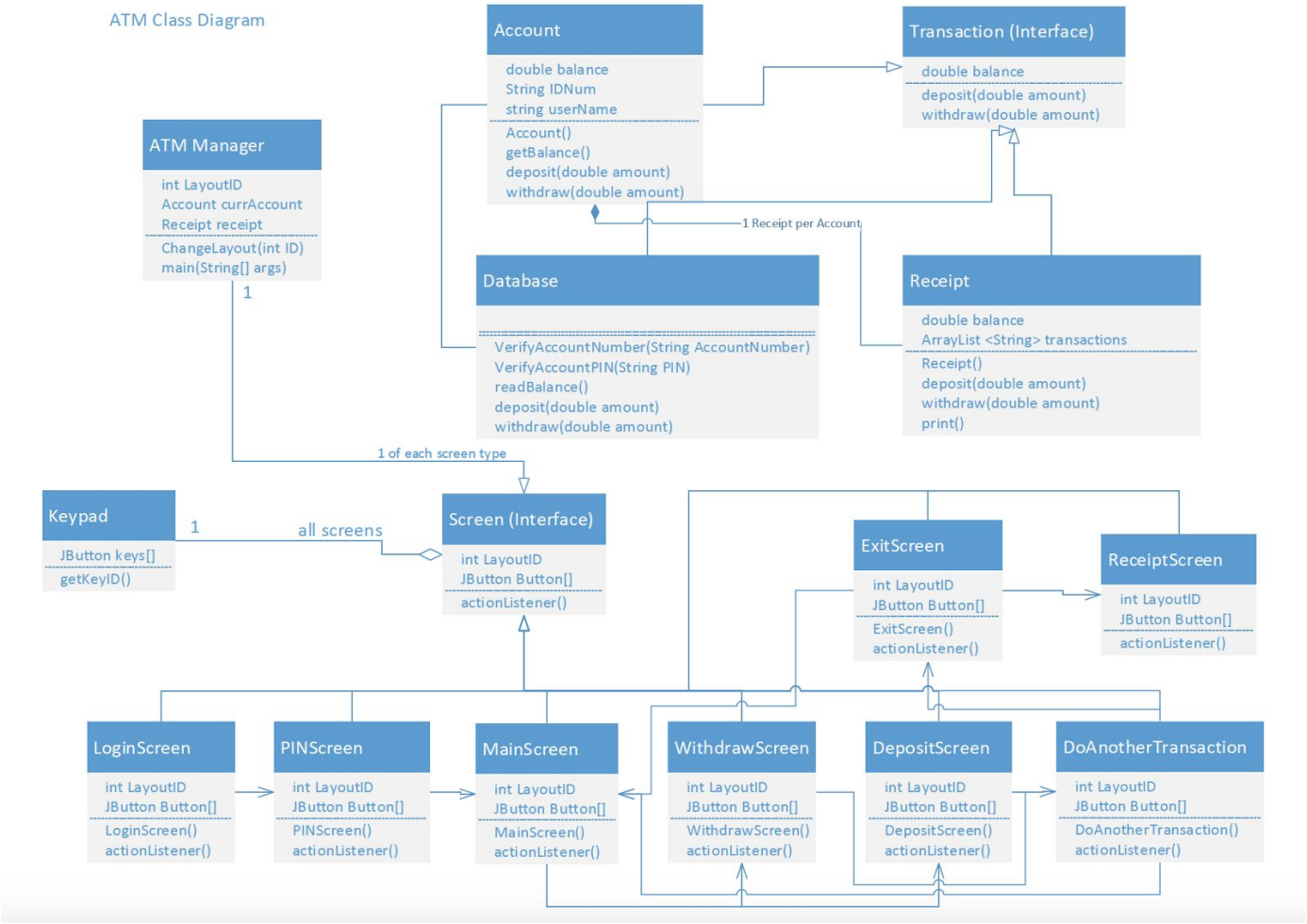
3.2 Package Diagram

This diagram provides a brief depiction of the packages that this software consists of. The UI package contains all of the components related to the UI of the program. The System package contains the related functionality components of an ATM system, and the Database package contains the necessary database components.



3.3 Class Diagram

This diagram shows the classes involved in this design and the relationships between them.



3.4 Class Descriptions

3.4.1 Transaction (Interface)

- has methods and attributes that are common to the Account, Database, and Receipt classes

3.4.2 Screen (Interface)

- includes methods and attributes common to all of the UI classes

3.4.3 ATMManager

- The main class that starts the ATM
- Manages the current working screen

3.4.4 Database

- Takes care of all operations utilizing the database

3.4.5 Receipt

- Creates a log of activities performed on the current session
- Prints out all activity on to the ReceiptScreen

3.4.6 Keypad

- Contains buttons that enter inputs into active text fields

3.4.7 ExitScreen

- Prompts the user to continue with the option for more transactions or to exit the session

3.4.8 ReceiptScreen

- Asks user if they wish to print receipt
- Displays the receipt (a collection of transactions performed during the current session)

3.4.9 LoginScreen

- Asks user for account ID and sends it to the database to verify

3.4.10 PINScreen

- Asks user for account PIN and sends it to the database to verify

3.4.11 MainScreen

- Gives user options to Deposit, Withdraw, or View Balance
- Shows balance if View Balance is selected

3.4.12 WithdrawScreen

- Gives user options to determine how much to withdraw

3.4.13 DepositScreen

- Gives user options to determine how much to deposit

3.4.14 DoAnotherTransaction

- Asks user if they wish to perform another transaction and acts based option selected

4. REQUIREMENTS RELATIONS

4.1 Functional Requirements

Each of the components used in this system architecture satisfy multiple aspects of all the functional use case requirements.

4.1.1 Account

- Use Case #1 Login
- Use Case #4 Exit
- Use Case #7 ViewBalance
- Use Case #8 Withdraw
- Use Case #9 CheckAvailableFunds
- Use Case #14 Deposit

4.1.2 Database

- Use Case #1 Login
- Use Case #2 ValidateAccountNumber
- Use Case #3 ValidatePin
- Use Case #4 Exit
- Use Case #7 ViewBalance
- Use Case #8 Withdraw
- Use Case #9 CheckAvailableFunds
- Use Case #12 PrintReceipt
- Use Case #14 Deposit

4.1.3 ATMManager

- Use Case #1 Login
- Use Case #4 Exit
- Use Case #5 Return
- Use Case #6 ViewMainMenu
- Use Case #7 ViewBalance
- Use Case #8 Withdraw
- Use Case #10 DispenseMoney
- Use Case #12 PrintReceipt
- Use Case #14 Deposit

4.1.4 Receipt

- 4.1) Use Case #4 Exit

- 4.2) Use Case #11 CheckReceipt
- 4.3) Use Case #12 PrintReceipt

4.1.5 Transaction

- Use Case #4 Exit
- Use Case #8 Withdraw
- Use Case #9 CheckAvailableFunds
- Use Case #10 DispenseMoney
- Use Case #12 PrintReceipt
- Use Case #14 Deposit

4.2 Interface Requirements

The specified User Interface Requirements are satisfied by the functions performed by all of the components in the UI package of this software architecture. The desired screens have their own classes with the proper buttons allocated for each screen.