THE UNIVERSITY OF ZAMBIA

SCHOOL OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER

SCIENCE



**Name:** TONY MULENGA CHIBESAKUNDA

**Computer Number:** 13031104

**Lecturer:** Mr. MARTIN PHIRI

**Course:** CSC 4630 Assignment 6

**ATM Requirements Analysis Document**

Table of Contents

[**1** **Introduction** 3](#_Toc526417335)

[**1.1** **Purpose of the system** 3](#_Toc526417336)

[**1.2** **Scope of the system** 3](#_Toc526417337)

[**1.3** **Objectives and success criteria of the project** 3](#_Toc526417338)

[**1.4** **Definitions, acronyms and abbreviations** 3](#_Toc526417339)

[**1.5** **References** 4](#_Toc526417340)

[**1.6** **Overview** 4](#_Toc526417341)

[**2** **Current System** 4](#_Toc526417342)

[**3** **Proposed System** 5](#_Toc526417343)

[**3.1** **Overview** 5](#_Toc526417344)

[**3.2** **Functional requirements** 5](#_Toc526417345)

[**3.3** **Nonfunctional requirements** 6](#_Toc526417346)

[**3.3.1** **Usability** 6](#_Toc526417347)

[**3.3.2** **Reliability** 7](#_Toc526417348)

[**3.3.3** **Performance** 7](#_Toc526417349)

[**3.3.4** **Supportability** 7](#_Toc526417350)

[**3.3.5** **Implementation** 7](#_Toc526417351)

[**3.3.6** **Interface** 7](#_Toc526417352)

[**3.3.7** **Packaging** 7](#_Toc526417353)

[**3.3.8** **Legal** 7](#_Toc526417354)

[**3.4** **System models** 7](#_Toc526417355)

[**3.4.1** **Scenarios** 7](#_Toc526417356)

[**3.4.2** **Use case Model** 8](#_Toc526417357)

[**3.4.3** **Object Model** 9](#_Toc526417358)

[**3.4.4** **Dynamic Model** 11](#_Toc526417359)

[**3.4.5** **User interface – navigational paths and screen mock-ups** 15](#_Toc526417360)

[**4** **Glossary** 16](#_Toc526417361)

# **Introduction**

## **Purpose of the system**

A local bank intends to install a new automated teller machine (ATM) to allow users (i.e., bank customers) to perform basic financial transactions. Each user can have only one account at the bank. ATM users should be able to view their account balance, withdraw cash and deposit funds.

## **Scope of the system**

The ATM software system will support a computerized banking system for a local bank.

## **Objectives and success criteria of the project**

To simulate the operations of a basic ATM system and to design and implement an object-oriented ATM software system which will be later integrated with the hardware. Since the ATM software is depicting all the basic operations of a simple and basic real life ATM system, it should therefore enable users to:

* View their account balance
* Withdraw cash
* Deposit Funds

## **Definitions, acronyms and abbreviations**

* Account

A single account in a bank against which financial transactions can be applied. We assume that a user can only have one account at the bank.

* Automated Teller Machine (ATM)

A station that allows users (bank customers) to perform their own basic transactions by using their account number and password to do so. The ATM interacts with the customer to gather transaction information, and sends this information to the bank’s database for validation, processing and thus prompt the user to perform basic financial transactions like; view balance, withdraw and deposit cash. We assume that the ATM doesn’t use ATM cards but only requires users to provide their account number and password.

* Bank Database

A computer information storage facility where the bank’s information about users and their accounts is stored for retrieval when a user wants to perform financial transactions.

* Bank Customer (User)

The holder of one account in a bank.

* Transaction

A single integral request for basic operations on the account of a single customer. Such operations include:

* + View account balance
  + Withdraw cash
  + Deposit funds

## **References**

[1] P.Deitel and H.Deitel, "ATM Case Study" in Java How to Program, 9th ed, Ed. Boston: Prentice Hall, 2012, pp. 470-509.

[2] B.Bruegge and A.H.Dutoit, "Requirements Elicitation" in Object-Oriented Software Engineering using UML, Patterns, and JAVA, 3rd ed, Ed. Boston: Prentice Hall, 2010, pp.152.

## **Overview**

This document is organized as follows: we have the first section, known as the introduction which is divided into a number of sub-sections, beginning with the purpose of system, then followed by the scope of the system, objectives and success criteria of the project, definitions, acronyms and abbreviations, references and then the overview arranged in that order respectively. Section 2 describes the current system. Section 3, is mentions about the proposed system, which is subdivided into a number of sub-sections. This includes; the overview, functional requirements, the nonfunctional requirements which are further divided into smaller sub-sections, which talk about usability, reliability, performance, supportability, implementation, interface, packaging and legal. System models fall under the sub-section of section 3, and these are also divided into smaller sub-sections just like those of the nonfunctional requirements. The sub-sections of the system models include; scenarios, use case model, object model, dynamic model and user interface – navigational paths and screen mock-ups in that order, respectively. And lastly section 4 includes the glossary.

# **Current System**

The local bank does not have an ATM software system, and currently all financial transactions are done over the counter inside the bank. Now this way of doing things does not offer users a 24-hour service to perform financial transactions. Therefore users have to perform basic financial transactions within the confines of the banks working hours. In order to provide users with a 24-hour service of performing basic financial transactions, the bank wishes to develop and implement an ATM software system which will be later integrated with the hardware in order to achieve this.

# **Proposed System**

## **Overview**

This section describes what the ATM software system will do, the factors that will influence the ATM’s performance as well as the models used in developing the system.

## **Functional requirements**

The functional requirements are organized into two general sections, the authorization and the general functional requirements

* Authorization

The authorization of the client begins after a customer has entered a valid account number and password.

Functional requirement 1

* + **Description**: if account number or password is invalid
  + **Input:** customer enters account number and password
  + **Processing:** check if account number or password exist in bank database
  + **Output:** Display error message and prompt user to try again, with only two attempts remaining

Functional requirement 2

* + **Description:** if account number and password is valid
  + **Input:** customer enters account number and password
  + **Processing:** check if account number and password exist in database
  + **Output:** redirect user to the main menu where they can select options to perform
* General

These are the transactions the ATM software system offers.

Functional requirement 3

* **Description:** check account balance
* **Input:** customer selects the check account balance option in the main menu
* **Processing:** system checks user’s account balance from the bank database
* **Output:** Display the user’s account balance as reflected in the database

Functional requirement 4

* **Description:** withdraw cash
* **Input:** customer selects the withdraw option and can then select which amount to withdraw according to the predefined amounts
* **Processing:** system checks if customer’s account has sufficient funds in account
* **Output:** if account has sufficient funds, dispense the amount selected, and prompt user to collect their money from cash dispenser. If account has insufficient funds, display insufficient funds message to user and prompt user to enter another amount

Functional requirement 5

* **Description:** deposit cash
* **Input:** customer selects the deposit option in the main menu, then prompt customer to place the money in the deposit slot
* **Processing:** collect money from the customer and add the amount to customers account
* **Output:** display message that amount was successfully added

## **Nonfunctional requirements**

### **Usability**

The ATM system has a basic user interface, which prompts users to enter their account number and password to login. After which, when the user is authorized, the main menu offers options which are numbered, and the user simply enters the number of the option they wish to select.

### **Reliability**

The ATM system will perform the transaction a user selects, and nothing more than that. For instance, if user selects to view account balance, the ATM will only show the user’s account balance and not withdraw the user’s money from their account. For security purposes, users are only allowed to make 3 attempts when login into their accounts.

### **Performance**

All selected transactions shall be completed by the ATM software system. Transactions that are in progress will be completed and not left undone.

### **Supportability**

Since the ATM software system will offer basic financial transactions, maintaining the system will be affordable. The software system will easily integrate with the ATM hardware system.

### **Implementation**

This software system will be implemented in Java.

### **Interface**

The software system will interface with the hardware system, as well as the bank’s database.

### **Packaging**

Not applicable.

### **Legal**

This ATM software system will only be used by this bank. And it will not leak any user information to the public.

## **System models**

### **Scenarios**

The scenarios are divided into two main classes;

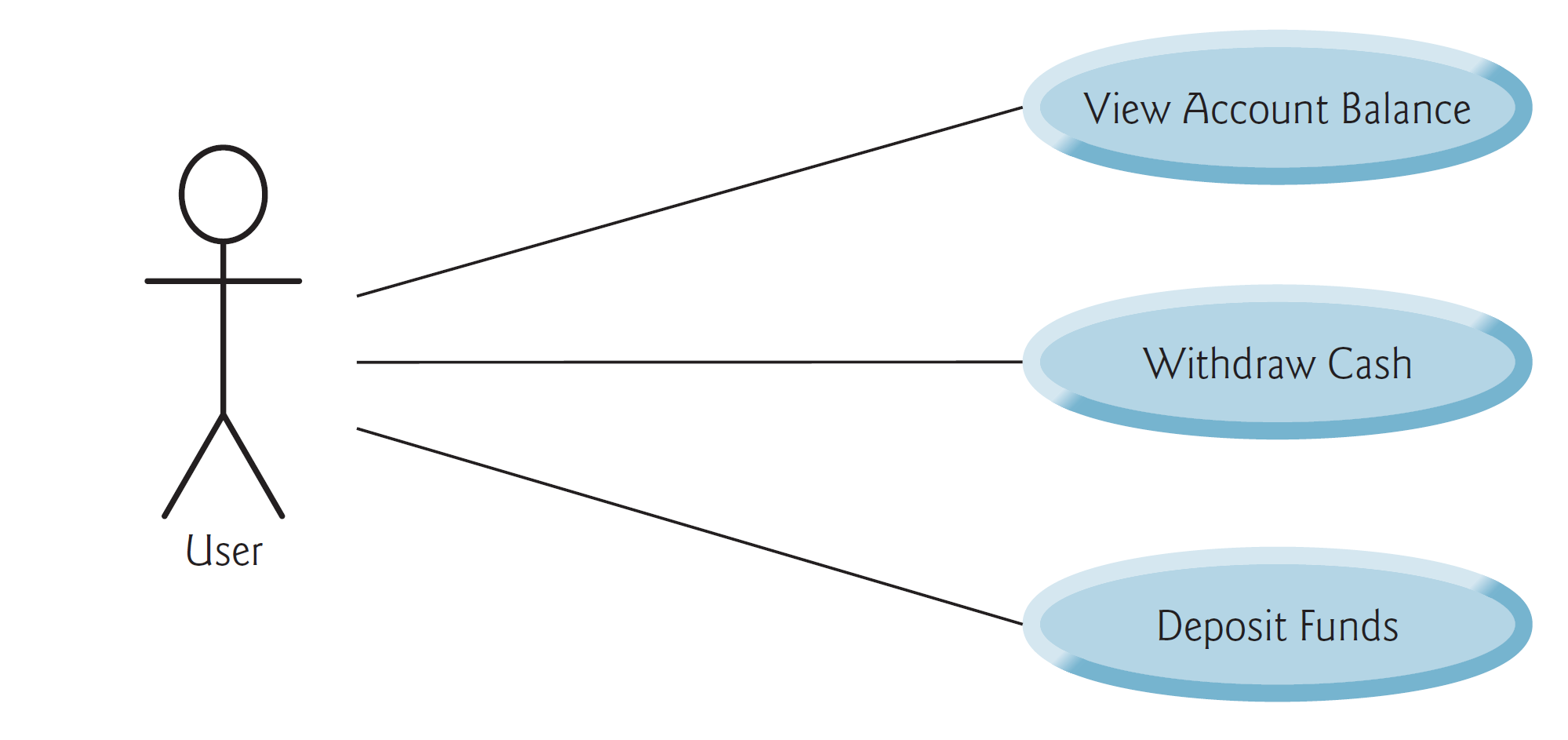
Success Scenarios

* Withdraw cash – Basic Flow
* Withdraw cash of a non-standard amount – Basic Flow, Alternative Flow: Handle the Withdraw of a Non-Standard Amount
* Withdraw cash with withdrawal receipt printed – Basic Flow, Alternative Flow: Handle the Printing of Receipts

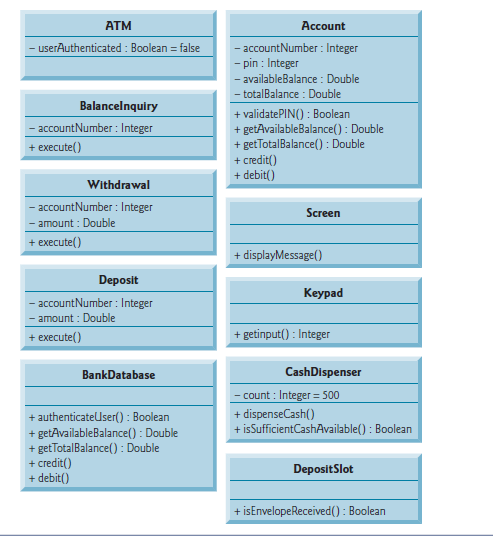
Failure Scenarios.

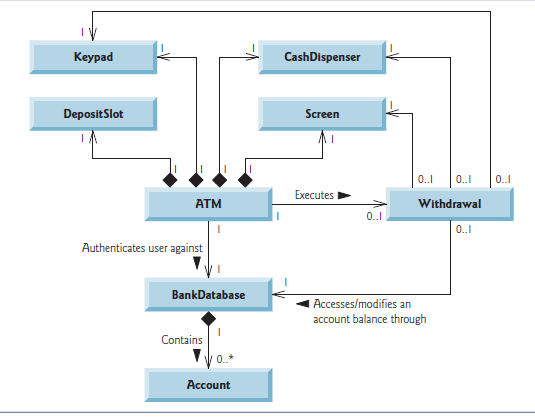
* Fail due to authentication failure – Basic Flow, Alternative Flow: Handle Authentication Failures
* Fail due to invalid account number or password – Basic Flow, Alternative Flow: Handle Invalid Card
* Fail due to insufficient funds in the account – Basic Flow, Alternative Flow: Handle the Bank Not Approving the Withdrawal
* Fail due to cash dispensing errors – Basic Flow, Alternative Flow: Handle Cash Dispensing Errors
* Quit – Basic Flow, Alternative Flow: Handle the Customer Quitting the Session

### **Use case Model**



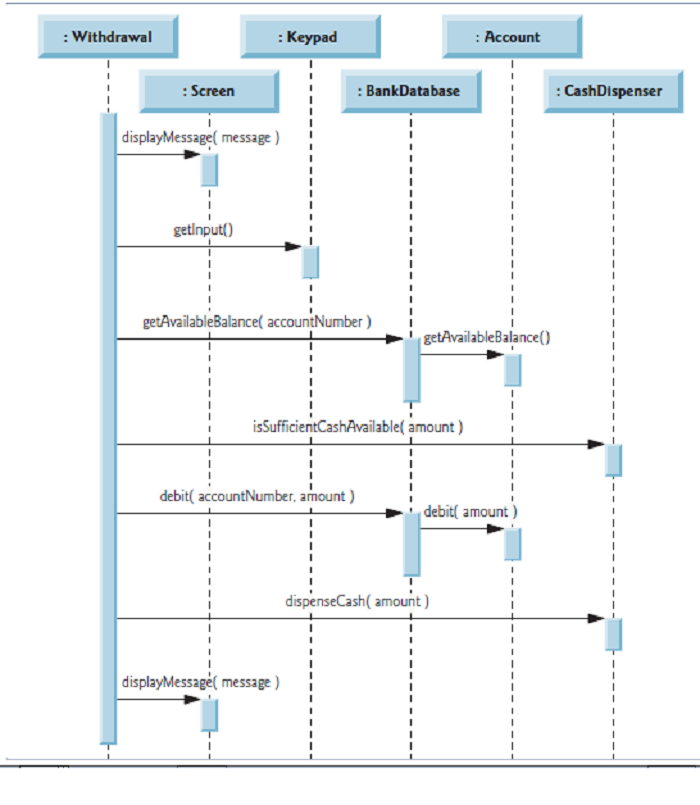
### **Object Model**



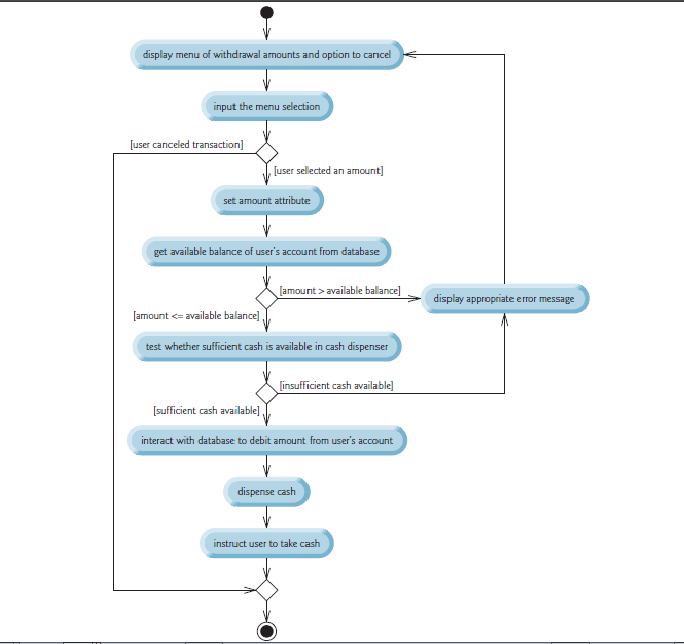


Association diagram for the ATM

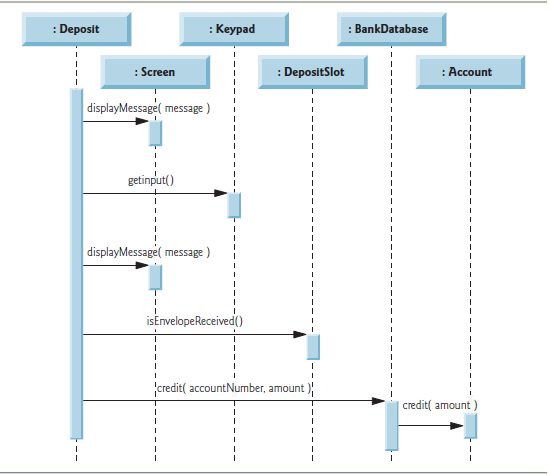
### **Dynamic Model**



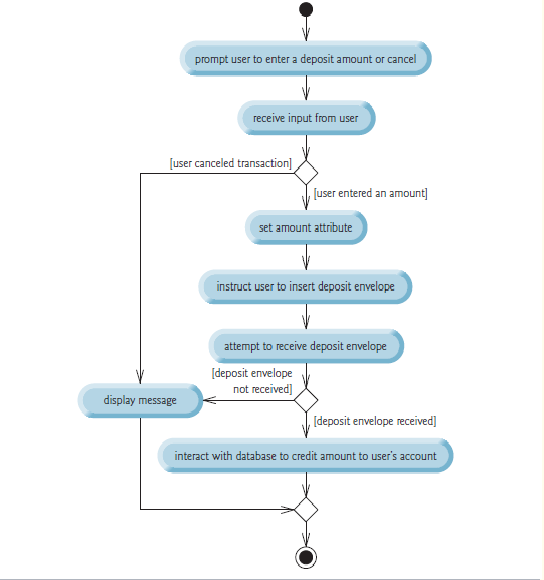
A sequence diagram for a withdraw procedure.



Activity diagram for a withdraw procedure.

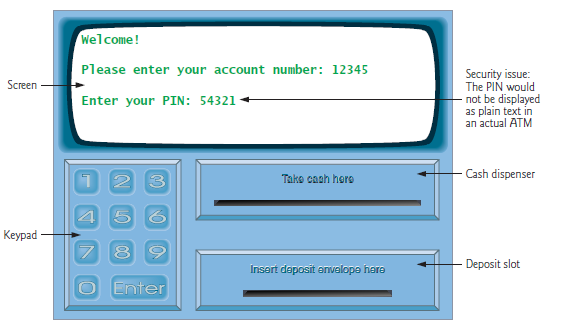


Sequence diagram for deposits.

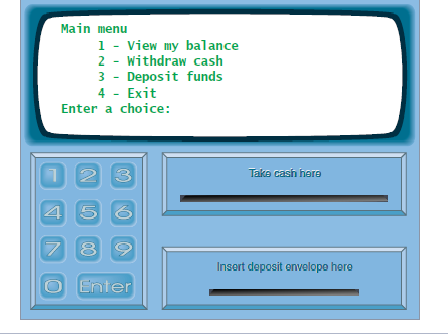


Activity diagram for deposits.

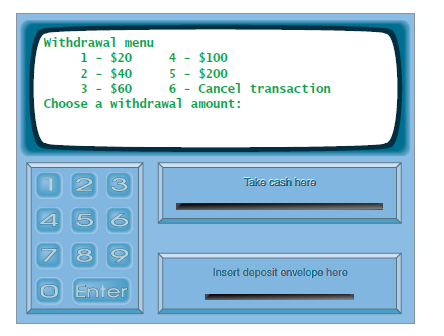
### **User interface – navigational paths and screen mock-ups**



User interface



Main menu



Withdraw menu

# **Glossary**

**Account number**

A unique number which identifies a user’s or bank customer’s account.

**Authorization**

When someone is given official permission to do something.

**Automated Teller Machine (ATM)**

ATMs can be referred to as cash machines or cash points. They enable bank customers or users to check their balance, withdraw money and deposit money into their account using an account number and password in the case pf this ATM software system.

**Balance**

A balance refers to the amount of money that you have in your bank account.

**Bank**

An organization that offers a range of financial services (e.g. savings, loans, current accounts and mortgages).

**Deposit**

To put money into a bank account.

**Transaction**

A payment made from one person to another; deposit or withdrawal from an account.

**Withdrawal**

Money taken out of an account.