

Software Requirements Specification

A.T.M. Application

Group 9.

Date	Changes	Version
2/1/18	Initial Document	1.0
2/9/18	Second draft	2.0

Table of Contents

1. Introduction.....	2
1.1 Purpose.....	2
1.2 Scope	2
1.3 Definitions, acronyms, and abbreviations	2
1.4 Overview.....	2
2. Overall Description.....	2
2.1 Product perspective.....	2
2.2 Product functions.....	3
2.3 User characteristics.....	4
2.4 Constraints.....	4
2.5 Assumptions and dependencies.....	4
3. Specific requirements.....	4
3.1 External interface requirements.....	4
3.2 Functions.....	4
3.3 Performance requirements.....	6
3.4 Logical database requirements.....	6
3.5 Design constraints.....	6
3.6 Software system attributes.....	6

1. Introduction

1.1 Purpose

The purpose of this document is to explain the requirements and features of the Java based A.T.M. system.

1.2 Scope

The A.T.M. application that we are proposing shall be a banking application that can be used on devices that can run Java applications. This program will allow the user to withdraw or deposit money into their account, transfer money between accounts of the same user, and check the current balance of the user's account. In addition to these features we will be implementing a feature that allows the user to withdraw from an international account and update their information accordingly. All of these functions are only able to be accessed if a valid debit card is used with the correct personal identification number. This will be made using java eclipse and a MySQL database.

1.3 Definitions, acronyms, and abbreviations

A.T.M.- automated teller machine

Foreign Currency- any currency other than the United States Dollar.

JDBC- Java Database Connection

PIN- personal identification number

Routing number- the identification number for a specific bank

1.4 Overview

The rest of this document will focus on the description, specific requirements, and an appendix for the ATM application. The description will focus on the use and purpose of this program. The specific requirements details all of the functionality of the program and when certain functions will be implemented. The appendix for the time being is empty.

2. Overall Description

2.1 Product perspective

2.1.1 System Interfaces

This program will host systems to be able to run java applications and have a connection to the MySQL database. This connection will be using the JDBC driver.

2.1.2 User Interfaces

This user interface will include both a virtual and physical number pad and will accept input from each, as well as a mouse for the purpose of testing the software. More input and output devices shall be added in the future.

2.1.3 Hardware Interfaces

In later iterations full hardware support in the form of a card reader, printer, and a scanner for bills and checks to be deposited is to be achieved.

2.1.4 Software Interfaces

This program will be made using Java Eclipse Neon. In addition to this there will be a Maria mySQL database.

2.1.5 Communications Interfaces

This program must be connected to the database in order to allow transactions. If a device does not have a connection to the database then transactions of any kind cannot take place.

2.1.6 Memory

This will consist of a simple graphical user interface and should not take much space on any device. For the purposes of testing the database will be much smaller in comparison to a world wide banking database. Many of the entries in the database will be designer made so that the application will have the full functionality of an A.T.M. but will not have access to actual bank accounts within the scope of this course.

2.1.7 Operations

There are several operations that are performed by the program. First there is the initial set up which allows the bank that owns the atm to set their location and set the transaction fees for users that do not have an account at the same bank. Then from there we have the security measure of needing the matching PIN to the debit card. Once that has been passed it should have all the functionality of an ATM allowing for deposits, withdrawals, transfers between a user's accounts, and checking account balance. In addition to these features this program can withdraw from an account that is a foreign currency.

2.2 Product functions

This product's function is to serve as a standardized ATM program that can run on a machine and does not have to worry about that machine's operating system.

2.3 User characteristics

The user must have either a checking or savings account with a bank and have a valid debit/credit card to be able to use the program.

2.4 Constraints

2.4.1 This application will require a connection to the internet in order to function properly, this is due to the required connection to a database.

2.4.2 The main concern is making sure that the connection to the database is secure and that the user's personal information is protected at all times during the connection to the database and the use of the A.T.M. Application.

2.5 Assumptions and dependencies

This program needs a host machine that can run a java program. The only devices that require any additional software will be Linux systems, these systems will need to download the Java Runtime Environment from the command shell for root users. In addition to this the host machine requires an internet connection. This may be done wirelessly or through a networking cable.

3. Specific requirements

3.1 External interface requirements

This program will require a card reader in future iterations, however, during testing of the initial product these will not be necessary. As not all hardware is standard there will be small differences between manufactures and thus cannot be stated at this time.

3.2 Functions

3.2.1 Phase 1

3.2.1.1 Start up

3.2.1.1.1 During phase one the atm information will be provided in the test cases. Upon start though there is a simple gui for users to enter data.

3.2.1.2 Checking credentials

3.2.1.2.1 When prompted with a valid card number the program will ask for the pin that is associated with that account.

3.2.1.2.1.1 If the pin does not match the transaction will end.

3.2.1.3 Withdraw

3.2.1.3.1 After a valid card number and pin have been entered then the user may withdraw from their account, the system will remove the amount specified by the user from the account number provided, the balance of the account will then be updated.

3.2.1.3.1.1 A user can not take out more than is in their account, if less than 20 USD is in the account then they shall not be able to

continue the transaction.

3.2.1.3.1.2 If the machine does not have enough money to allow for a withdraw an error message will display and the transaction will end.

3.2.1.4 Transfer

3.2.1.4 If a user has more than one account that is linked to the card used at the ATM then they will be able to transfer money from one of their accounts to another one of their accounts.

3.2.1.4.1 The amount that can be transferred between accounts must be less than the total of the giving account. If the amount is greater or equal to the value of the giving account then an error message will be shown and the transaction will end.

3.2.1.5 Deposit

3.2.1.5.1 A user may say that they wish to deposit into either their checking or savings account.

3.2.1.5.1.1 The issuer of the check must have enough funds to transfer into the account or the transaction will not continue.

3.2.1.5.1.2 If the account is valued at a different currency then any amount of money placed into the account must be from the same country as the location of the ATM from phase two onward.

3.2.2 Phase 2

3.2.2.1 Graphical user interface

3.2.2.1.1 In phase two a GUI shall be implemented so that the product is user friendly, in addition to this the product will be able to be used on touch screen devices. This allows for users to either press physical buttons on the machine to select functions or to use the touch screen.

3.2.2.1.2 When the program is first started it will ask for the information of the bank that owns the machine to know when to charge a transaction fee. This information will be taken from a GUI in phase two, but for phase one this shall be hard coded.

3.2.2.2 Allowing foreign currencies

3.2.2.2.1 Not every user will be in the United States and there are many currencies used internationally. To account for this a conversion rate system will be implemented to allow for withdraws into the currency of the ATMs location.

3.2.2.2.1.1 This will apply to both the withdraw and deposit and will update the account's value in the currency native to the user.

3.2.3 Future iterations

3.2.3.1 Hardware

3.2.3.1.1 The future iterations will be dedicated to gaining hardware support.

3.2.3.1.1.1 The future iterations will be able to recognize card readers, a sensor for depositing checks or paper money, and a receipt printer.

3.3 Performance requirements

We aim for 90% of transactions to be completed within one minute from start to finish. The time it takes to send a query and receive the information in question shall be less than 3 seconds provided that the account is valid. There must be many hosts and for the purpose of testing we will have 2 host machines. These two machines will allow for many users and the only way that problems will occur is if the user of both machines is the same person.

3.4 Logical database requirements

A MySQL database will be implemented to hold the bank and accounting information. Banks will have a unique routing number and the account numbers will be unique as well. The user may use several different banks and the database must be able to assign multiple accounts to a person. This also means that any card that specifies a user can only have accounts from the bank that issued the card. People will be identified by their social security number and any personal or numerical information that is used during a transaction will be cleared from the software at the end of the transaction.

3.5 Design constraints

As this software aims to replace the software that is used on current A.T.M.s there may be problems due to the changes in hardware that a specific unit uses. This is because of the changes in technology that occur with time and recognising the input and output devices as well as the connections to these devices is the main concern with the design.

3.6 Software system attributes

3.6.1 Reliability

This program must be able to maintain a secure connection to the database and remove any personal information at the end of each transaction to maintain the reliability and security for a banking application.

3.6.2 Availability

This program must be able to run indefinitely, as we expect there to be long idle periods followed by short periods of use. This means that the program must be ready to receive input anytime that the device is not in use

3.6.3 Security

Due to this being a banking application security is a primary concern. Due to a lack of experience in this area we are currently planning on putting the input into a hashmap as well as the data that is received from the database in order to establish a positive confirmation without a direct comparison.

3.6.4 Maintainability

This project shall be fairly simple to maintain from a functional standpoint. Security measures will be the most difficult to change or maintain as this may require a different style of connection to the database and would require a new function to support the changes. The other function that would require regular maintenance would be the conversion rates of foreign currencies, this would

require daily updated values to be entered into the database.

3.6.5 Portability

The application is made to be portable due to Java not being a primary operating system so it can be used on many different devices and is not dependent on any specific manufacturer or host software. The problems with portability will stem from the input and output devices.