



**Requirement Specification for ATM System**

**College： Maths、Physics and Information**

**Major： Software Engineering 151**

**Author： 陈鹏鹏（201532120101）**

**王 琪（201532120115）**

**11/12/2017**

**Content**

[1. System Overview 3](#_Toc500872182)

[1.1 Functional Introduction 3](#_Toc500872183)

[1.2 User Role 3](#_Toc500872184)

[2. Reason 4](#_Toc500872185)

[2.1 For the bank 4](#_Toc500872186)

[2.2 For the students 4](#_Toc500872187)

[3. System Boundary 4](#_Toc500872188)

[3.1 Design Scope 4](#_Toc500872189)

[4. User scenarios 5](#_Toc500872190)

[5. Use Case 6](#_Toc500872191)

[5.1 User use case steps 6](#_Toc500872192)

[5.2 Use Case Diagrams 7](#_Toc500872193)

[6. Requirement 8](#_Toc500872194)

[6.1 Functional Requirement 8](#_Toc500872195)

[6.2 Non-Functional Requirement 8](#_Toc500872196)

[7. Technical requirement 9](#_Toc500872197)

[Appendix 10](#_Toc500872198)

## 1. System Overview

ATM System is close to our life. It provides many businesses, including withdrawing、depositing、transferring and querying. These businesses occur at any time in our daily life. The person who has applied a bank account is called depositor. One depositor can apply for several accounts in the bank. The depositor can deposit money into an account, or withdraw it from his account, or transfer the money from one account to another, and query their balance and transaction records at any time. In addition, depositors can also change their password. Every time when the amount of money or password changes, the database of the bank which store the depositors and accounts information will also change.

### Functional Introduction

The function can be concluded to 5 parts : user validation、withdraw、deposit、transfer and query balance. Just shown as the Figure 1.1.

ATM System

Exit

Operation

Login

Enter Password

Enter Account

Withdraw

Exit

Query

Transfer

Deposit

Validation

Figure 1.1: the function module diagram for the ATM system

### 1.2 User Role

Table 1.1 : The User role and responsibility

|  |  |
| --- | --- |
| **System User** | **Scope of duty** |
| Customer | The user who has an account can withdraw、deposit、transfer and query. |
| Administrator | The administrator mainly processes some fortuitous event for the ATM system.  For example: When the user enters wrong password over 3 times |
| Mechanic | The mechanic is to repair the ATM system when it is out of control and maintain the system regularly. |
| The bank computer | The ATM system shares the same database with the bank computers.  They will update the account information synchronously if users do transaction on one of the two ways. |

## 2. Reason

### 2.1 For the bank

The traditional counter service of bank exposes some problems with the improvement of social life. The biggest problem is low efficiency, because the space and employees are limited. Besides, there are some inconvenient processes, including taking number paper and signature. Therefore, an ATM system greatly saved the time for customers and employees. It improves the word efficiency for the bank also is very convenient for customers.

### 2.2 For the students

For us, a design of ATM system is really a challenge. How to apply the knowledge learnt in the class and notebooks to the practice, that help us to improve our ability, including cooperation 、analysis and coding ability. Besides, the ATM system consists of so many functionalities and actors, how they interact with each other, which has to be familiar with object oriented. Analyze the whole system by dividing several classes will make the system modular and have clear logic. After the practice, we can make a progress in the study and have a deep understanding to this course.

## 3. System Boundary

### 3.1 Design Scope

Consider the difficulties、workload and time in the development, the system will focus on the analysis and design but pay less energy in the detailed realization.

1. Requirement Specification. Before finishing the part, firstly we need to analysis deeply, given the project overview、functional and non-functional requirements、user role and user scenarios. As well as functional module diagram and flow chart to help us better understand the requirements.
2. UML Diagrams. Class diagram helps to describe the relationship between different classes. Use case diagram helps to analyze the functions of the system and clarify the interaction between the system and the external user roles.
3. Database Design. By UML diagram, we can list the tables needed in the system quickly.
4. According to the UML diagrams, we can set about coding.

## 4. User scenarios

Table 4.1 : Scene 1 for customer

|  |  |
| --- | --- |
| User Role | Customer |
| Business Requirements | The customer wants to withdraw/deposit/transfer/query in the ATM system. |
| Scene description | 1. Jack comes to a ATM machine, he wants to withdraw 2000 yuan for his monthly living expense. Firstly, he inserts his bank card, there is a login interface. He enters correct password, then the screen shows four operations, including withdrawing、depositing、transferring and querying. He chooses withdrawing, then enters the amount he wants to take. The ATM system processes his request, he can withdraw the money in the slot. Then the system asks him whether wants to continue to do transaction. He chooses query to check his balance, absolutely his balance lacks of 2000 yuan. He don’t want to do transaction any more, so he choose exit to take out his bank card. 2. Mary is the mother of Jack. It is a new month, she comes to a ATM machine to transfer money to his son. Firstly she inserts her bank card and enters correct password. Then she choose transferring, it asks her to enter the account she wants to transfer to. She enters the account of Jack and the amount she transfers to. The ATM processes her request and show successfully transferred interface. Mary finishes her transaction and choose exit to take her card. 3. Lily gets her salary with cash and she wants to deposit to her bank card. She comes to a ATM machine, after inserting card and enters correct password, she chooses depositing operation. The deposit slot opens and asks her to put money on, she puts 1000 yuan into the slot and confirms. After the system checks the money, she finishes her depositing and directly chooses exit. |

Table 4.2 : Scene 2 for administrator

|  |  |
| --- | --- |
| User Role | Administrator |
| Business Requirements | The administrator mainly processes some fortuitous event for the ATM system. |
| Scene description | 1. Angela comes to an ATM machine to check her balance, but she forgot her password. After she has tried the third time with still wrong password, the system prompts Angela that her account was locked up and can’t continue to do transaction. She has to contact the administrator for help to her further using for the transaction. 2. When the balance of ATM is less than 1000 yuan, the system feedbacks to the administrator automatically. |

## 5. Use Case

### 5.1 User use case steps

Table5.1 Use case steps for login

|  |  |
| --- | --- |
| Use case name | Login |
| Behavior role | Customer who wants to do transaction |
| Brief explanation | The customer needs to enter correct account and password to realize login. |
| Precondition | (1)The customer needs to have at least an account and correct password.  (2)The account should consist of 19 digits and password with 6 digits. |
| Postcondition | ATM system will come to the operation choosing interface. |

Table5.2 Use case steps for querying

|  |  |
| --- | --- |
| Use case name | Query |
| Behavior role | Customer who wants to do transaction |
| Brief explanation | 1. Choose query in the operation choosing interface. 2. ATM shows the balance, click inform to return operation choosing interface |
| Precondition | The customer must login successfully. |
| Postcondition | ATM returns to the operation choosing interface. |

Table5.3 Use case steps for transferring

|  |  |
| --- | --- |
| Use case name | Transfer |
| Behavior role | Customer who wants to do transaction |
| Brief explanation | 1. Choose transfer in the operation choosing interface. 2. Enter the account and amount of money transfer to. 3. Confirm the transferred information and continue. 4. Transferred successfully and show the balance. |
| Precondition | (1)The customer must login successfully.  (2)The account for receiver should be able to correct.  (3)The amount of transferred money should be larger than 0 and no more than the balance of his own account. |
| Postcondition | ATM shows the balance and database updates the balance of the two accounts. |

Table5.4 Use case steps for withdrawing

|  |  |
| --- | --- |
| Use case name | Withdraw |
| Behavior role | Customer who wants to do transaction |
| Brief explanation | 1. Choose withdraw in the operation choosing interface. 2. Enter the amount of money to withdraw. 3. ATM opens the withdraw slot, customers take out the money. |
| Precondition | (1)The customer must login successfully.  (2)The amount of money should be larger than 100 and be the multiple of 100.  (3)The amount of money should be no more than 2500yuan, as well as the balance. |
| Postcondition | ATM shows the balance and database updates the balance. |

Table5.5 Use case steps for depositing

|  |  |
| --- | --- |
| Use case name | Deposit |
| Behavior role | Customer who wants to do transaction |
| Brief explanation | 1. Choose deposit in the operation choosing interface. 2. ATM opens the deposit slot, customers put into money. 3. ATM checks the money, if no forged money, shows successfully deposited. |
| Precondition | (1)The customer must login successfully.  (2)The amount of money should be larger than 100 and be the multiple of 100. |
| Postcondition | ATM shows the balance and database updates the balance. |

### 5.2 Use Case Diagrams

See the appendix

## 6. Requirement

### 6.1 Functional Requirement

Table6.1 Functional requirements

|  |  |
| --- | --- |
| **Functional Requirements** | |
| **Content** | **Description** |
| Login | Customers who enter correct account and password can use the system and do transactions. |
| Withdraw | After the system confirm withdrawal request, the system interface asks to enter amount of money, which should be no more than 2500 yuan. Customer enters a number and clicks confirm, the system will send the request to currency counting machine, then the currency counting machine will pay out money. After customer takes the money, the system can automatically update balance. When customer chooses finish service, the system will exit. |
| Deposit | After the system confirms the deposit request, system come to the deposit interface, the customer will put money into the deposit slot, the system check the money, then record and update the balance. The system shows that deposit is completed, the customer clicks exit, then the system logs out. |
| Transfer | Customers choose to query the balance service, the system confirms the request, according to the customer information to find the balance, and displayed on the interface. |
| Query | Customers choose to query the balance service, the system prompts the customer to enter the account and transferred amount. The amount should not be larger than balance. After the customer confirms the information, the system processes the request and the database will update the balance. |
| Exit | When customers choose exit, the system will log out. The customer must login again if he wants to do transaction. |

### 6.2 Non-Functional Requirement

Table6.2 Non-Functional requirements

|  |  |
| --- | --- |
| **Non-Functional Requirements** | |
| **Content** | **Description** |
| Performance requirement | If the system doesn’t receive any response over 20s, it will show that “Error, request timeout” and return to the main interface. |
| Reliability requirement | According to the statement in the requirement specification, the function of the system should be executable. |
| Usability requirement | The system design should have a good usability and easy to operate. Besides, it should satisfy the common user usages under the normal Windows operating systems. |
| Security requirement | The system should protect the users’ personal information, which shouldn’t be easy to be accessed by others. Even if the system is out of control, it still needs to ensure the safety of the data relevant to the user. |
| Supportability requirement | The system has good maintainability, when the system  broke down, to make sure it can be reverted to the  normal state in two days. |

## 7. Technical requirement

Hardware equipment: Two computers

Development Platform：Windows

Development Language：Java

Development Tool: Eclipse

The system adopts the popular B/S architecture

Application Server：Tomcat7.0

Database Server：MySql、SQL Server Or Oracle

Considering the pertinence, safety, ease of use, the system has good measurability.

## Appendix

Figure 5.2 Use Case Diagram for ATM system