Software Requirement Specification for Banking System

Table of Contents

Introduction

This document, Software Requirements Specification (SRS), is created to document the software requirements for the Banking System. A bank has several automated teller machines (ATMs), which are geographically distributed and connected via a wide area network to a central server. Each ATM machine has a card reader, a cash dispenser, a keyboard/display, and a receipt printer. By using the ATM machine, a customer can withdraw cash from either checking or savings account, query the balance of an account, or transfer funds from one account to another. A transaction is initiated when a customer inserts an ATM card into the card reader. Encoded on the magnetic strip on the back of the ATM card is the card number, the start date, and the expiration date. Assuming the card is recognized, the system validates the ATM card to determine that the expiration date has not passed, that the userentered PIN (personal identification number) matches the PIN maintained by the system, and that the card is not lost or stolen. The customer is allowed three attempts to enter the correct PIN; the card is confiscated if the third attempt fails. Cards that have been reported lost or stolen are also confiscated. If the PIN is validated satisfactorily, the customer is prompted for a withdrawal, query, or transfer transaction. Before a transfer transaction can be approved, the system determines that the customer has at least two accounts and that there are sufficient funds in the account to be debited. For approved query and transfer requests, a receipt is printed and card ejected. A customer may cancel a transaction at any time; the transaction is terminated and the card is ejected. Customer records, account records, and debit card records are all maintained at the server. An ATM operator may start up and close down the ATM to replenish the ATM cash dispenser and for routine maintenance.

Literature Survey

As competition has intensified and customer needs have also increased, so too have the challenges faced by banks. Customers demand access to their financial information regardless of their location or the time of day, and if their current financial institution can't provide it they can always go to someone else who can. Often installed decades ago, legacy core banking systems just can't cope – it may be impossible to support the latest products and when it is, the process is complex, time consuming and expensive. Just keeping these systems running can often consume more than 70% of the IT budget leaving little money to gain advantage over competitors. And by the time the data is collected it is often too late – the customers' needs have moved on. We can see long queues of customers in a bank every now and then. This queue is the final result of the slow processing speed of the Bank. So, a highly interactive and user-friendly solution should be developed. With the implementation of Banking system, the customers' status has been changed from 'Branch Customers' to " Bank Customers". It is immaterial with which branch of the Bank the customer deals with. For the smooth working of the bank, the bank needs to be designed in such a way that, all the operations that were previously performed with difficulties are performed easily in this system. For the customers an internet solution is the most appropriate one as almost all customers have access to it.A well interfaced GUI would be used for connecting to the main database server for updating and retrieving the data of the customers. It would also deal with the Employees of the Bank, their registration, removal, manager allotment, etc.

Requirements

Functional Requirements

- **Purpose**
  To register a new customer
- **Inputs**
  The required data for registration of a new customer in the bank (Like Name, Address, Designation etc).
- **Outputs**
  A Success Message be displayed on successful registration or else an error message will be displayed.

Non-Functional Requirements

Non-functional requirements are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. They may specify system performance, security, availability, and other emergent properties.

Safety Requirements

a) Backup, recovery & business continuity Banks should ensure adequate back up of data as may be required by their operations. Banks should also have, well documented and tested business continuity plans that address all aspects of the bank's business
b) Both data and software should be backed up periodically.
c) An off-site back up is necessary for recovery from major failures / disasters to ensure business continuity.

Security Requirements

a) Account ID and Password (PIN) Protection
b) Auto Timeout Screen Blanking
c) Sign-off Button
d) Failed Log-on Attempts
e) Encryption

Software Quality Attributes

a. Reliability
Measure if product is reliable enough to sustain in any condition. It should give consistently correct results. Product reliability is measured in terms of working of project under different working environment and different conditions.

b. Maintainability
Different versions of the product should be easy to maintain. For development, it should be easy to add code to existing system, should be easy to upgrade for new features and new technologies time to time. Maintenance should be cost effective and easy. System be easy to maintain and correcting defects or making a change in the software.

c. Usability
This can be measured in terms of ease of use. Application should be user friendly. It should be easy to learn. Navigation should be simple.

d. Portability

This can be measured in terms of Costing issues related to porting, Technical issues related to porting, Behavioral issues related to porting.

e. Correctness

Application should be correct in terms of its functionality, calculations used internally and the navigation should be correct. This means application should adhere to functional requirements.

f. Efficiency

To Major system quality attribute. Measured in terms of time required to complete any task given to the system. For example system should utilize processor capacity, disk space and memory efficiently. If system is using all the available resources then user will get degraded performance failing the system for efficiency. If system is not efficient then it can not be used in real time applications.

g. Flexibility

Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.

Hardware Requirements

- Standard PC
- Internet connection with good enough speed
- ATM
- 128MB of more RAM (256 recommend)
- Smart Mobile phone

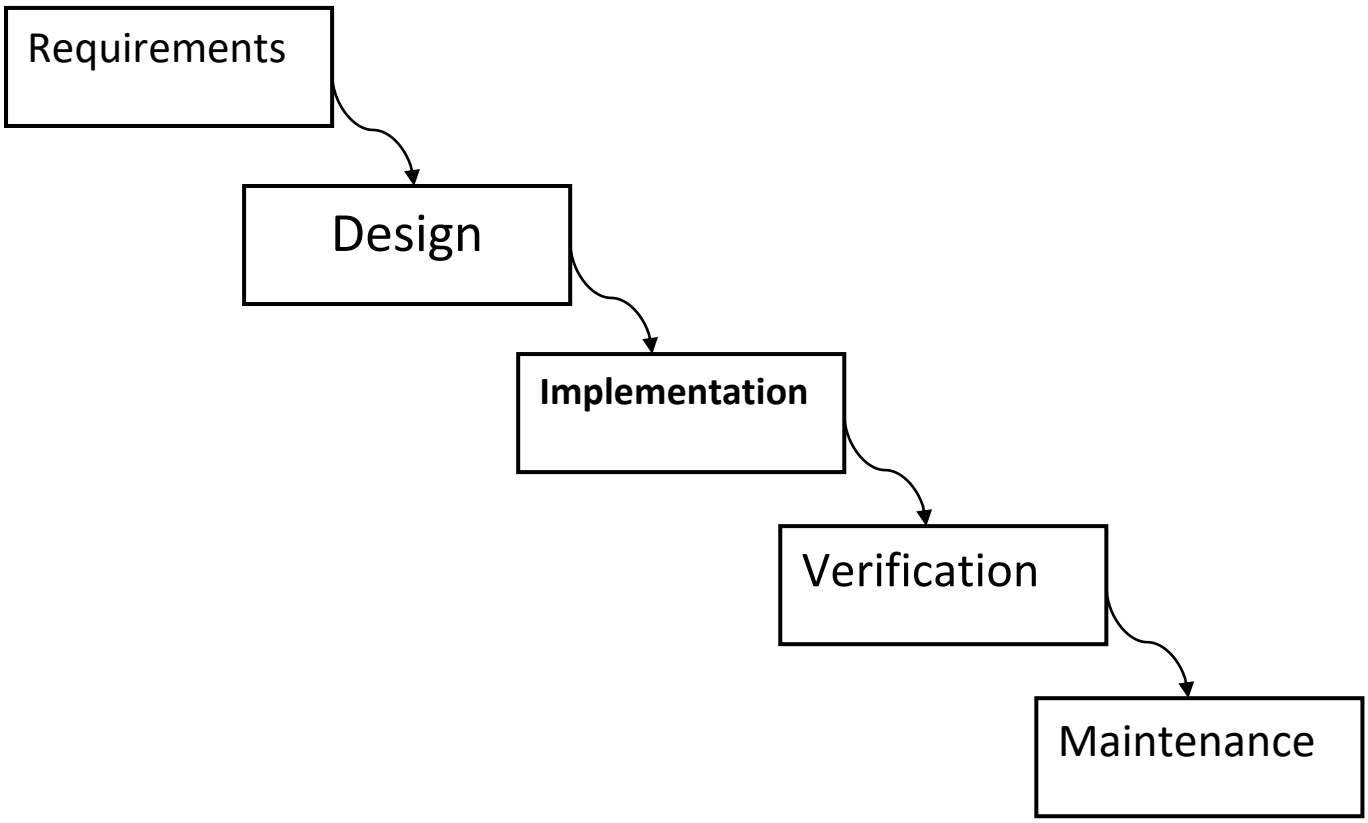
Software Requirements

This product is developed mainly using open source technologies like apa che, php, gtk+ etc. So,we can use any operating system for developing this product. Frontend: GTK+ 2.8.20 , GCC 4.0.0, PHP 5.20 , Glade 2.10.1 (For CBS) Backend: MySql 4.17 Web Server: Apache 2.2 Platform used: Fedora Core 4 Linux, Windows XP / Windows7/ Windows Vista Web Browser: Microsoft Internet Explorer 4.0,Mozilla ,Google Chrome or later.

Water Fall Model

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Analysis, Requirement Specification, Design, Implementation, Testing and Integration and Operation and Maintenance. If in the beginning of the project failures are detected, it takes less effort (and therefore time and money) for this error. In the waterfall model phases to be properly sealed first before proceeding to the next stage. It is believed that the phases are correct before proceeding to the next phase. In the waterfall model lay the emphasis on documentation. It is a straightforward method. The way of working ensures that there are specific phases. This tells you what stage it is. One can use this method of milestones. Milestones can be used to monitor the progress of the project to estimate.

In our Project, all the requirements are clear and well known and the project is large. All the activities in our project are carried out in above mentioned phases of waterfall model.



Feasibility Study

The prime focus of the feasibility is evaluating the practicality of the proposed system keeping in mind a number of factors. The following factors are taken into account before deciding in favor of the new system.

- **Economic Feasibility**

  The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

  - The costs conduct a full system investigation
  - The cost of the hardware and software
  - The benefits in the form of reduced costs or fewer costly errors.
  - Since the system is developed as part of project work, there are already available, it given and indication of the system is economically possible for developed.

- **Technical Feasibility**

  The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical issues raised during the investigation are: Does the existing technology sufficient for the suggested one? Can the system expand if developed? The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

- **Operational Feasibility**

  This includes the following questions:
  Is there sufficient support for the users?
  Will the proposed system cause harm?
  The project would be beneficial because if satisfies the objectives when developed and installed. All the behavioral aspects are considered carefully and conclude that project is behaviorally feasible.

Data Dictionary

Data Dictionary Actor Description Instructions

Name - Name of the actor EXACTLY as it appears on the use case diagram. It must be a noun or noun phrase with the first letter of the name capitalized.

Alternate Name(s) - Alternative names this actor may be referred to in the application domain. Providing these names helps the readers to understand this document.

Input Data - List of the inputs to the system that this actor provides. This section must contain a list of the use cases with which this actor interacts (has a line on the use case diagram) and provides input. For each use case, list the inputs this actor provides.

Output Data - List of the outputs from the system that this actor receives. This section must contain a list of the use cases with which this actor interacts (has a line on the use case diagram) and receives output. For each use case, list the outputs this actor receives.

Description - Brief description of the general purpose or role of this actor.

Comments - Any additional information that aid in the understanding of this actor.

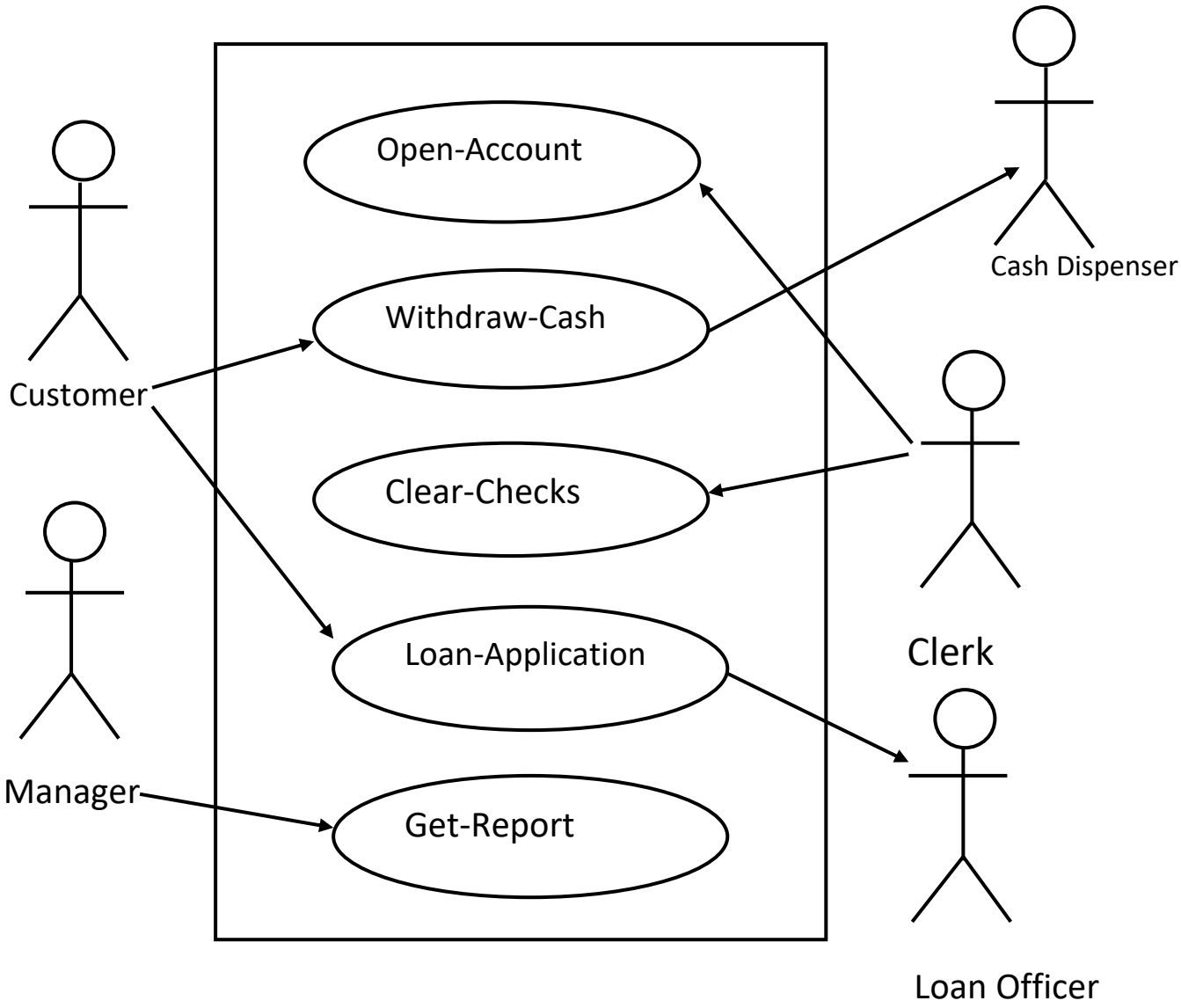
Design & Implementation

- The product is completely data oriented.
- Here, users would input the various details of the transactions customers, employees etc. for storing, updating, processing or retrieval of data from the database as per the instructions given and display an acknowledging message to the user.
- Login and password is used for identification of customer's account and there is no facility for non users to login.
- This system works only on a single server.
- GUI is only in English.
- Limited to HTTP/HTTPS protocols.
- When we consider the banking in this we provide the details of how to access the bank account without going to the bank through internet.
- When we consider the priority of this project it is mainly of medium cost, efficient to user access data, provides the required data, safe and secure one .we can know the details of our account whether it may be a transaction or deposit or balance enquiry etc.
- Overall view of the banking system: The overall view (design and implementation) of the banking System is as shown below:

1) Class Diagram
2) Use-case Diagram

Use-Case Diagram

Banking System Use Case Diagram



Data Dictionary Actor Description Template

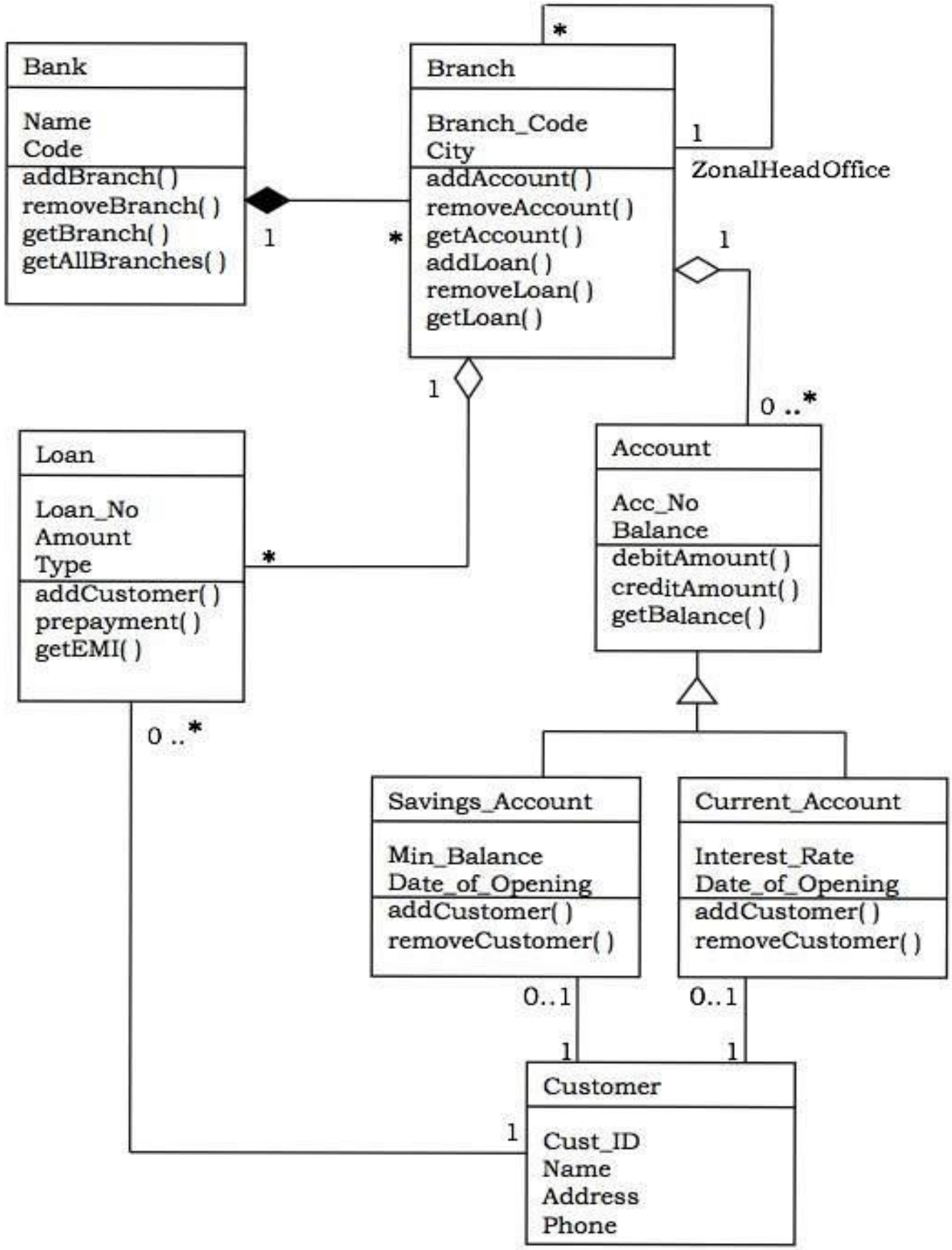
Input Data

Name of Use Case	Inputs to the System
Withdraw cash	Enters ATM card, Enters pin
Transfer funds	Enters ATM card, Enters pin
Check Balance	Enters ATM card, Enters pin
Deposit cash	Cash amount, Enters ATM card, Enters pin

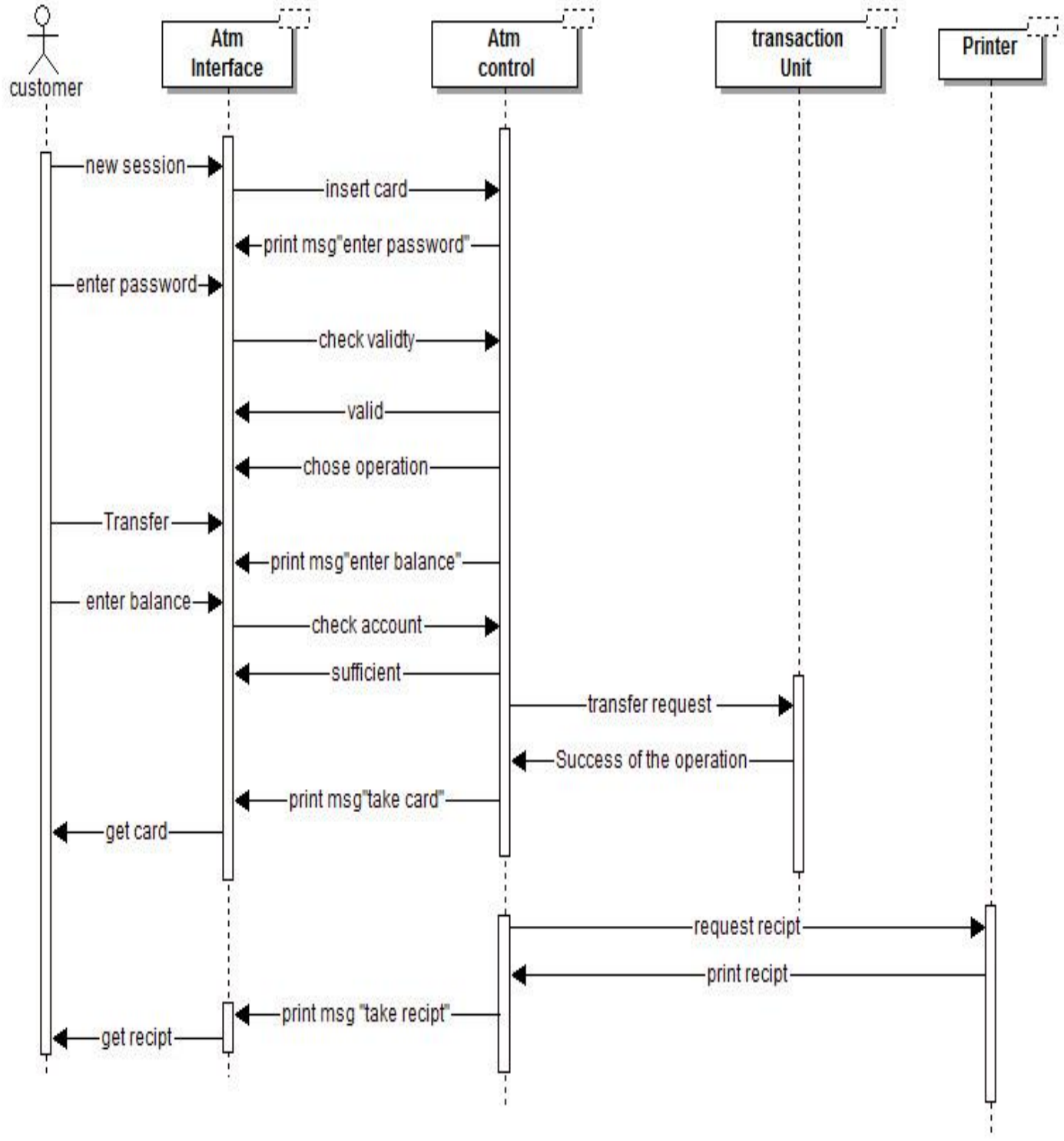
Output Data

Name of the Use Case	Outputs from the system
Withdraw cash	Get cash amount, Get receipt
Ceck Balance	Get status receipt
Transfer funds	Get receipt
Deposit cash	Get receipt

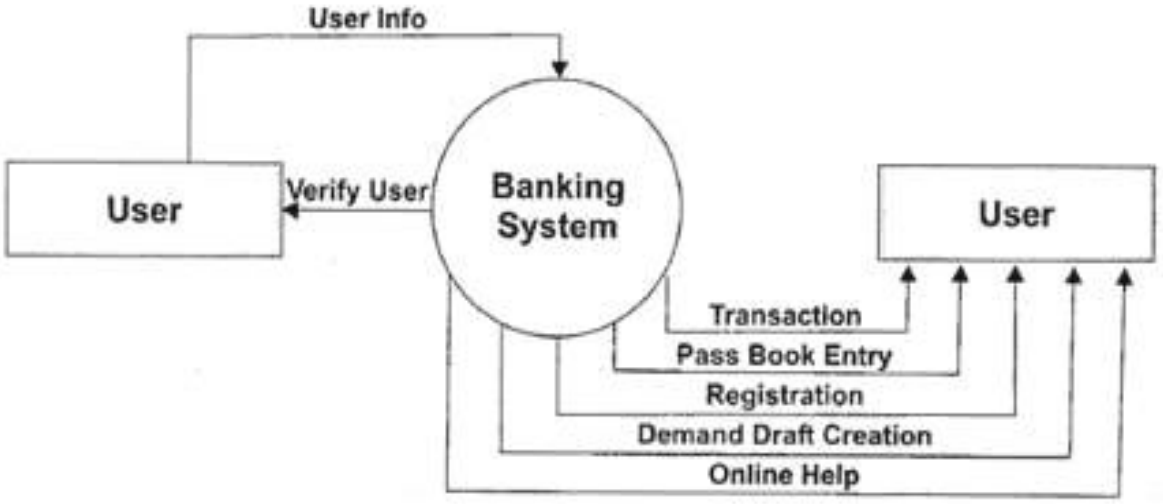
Class Diagram



Sequence Diagram of Banking System

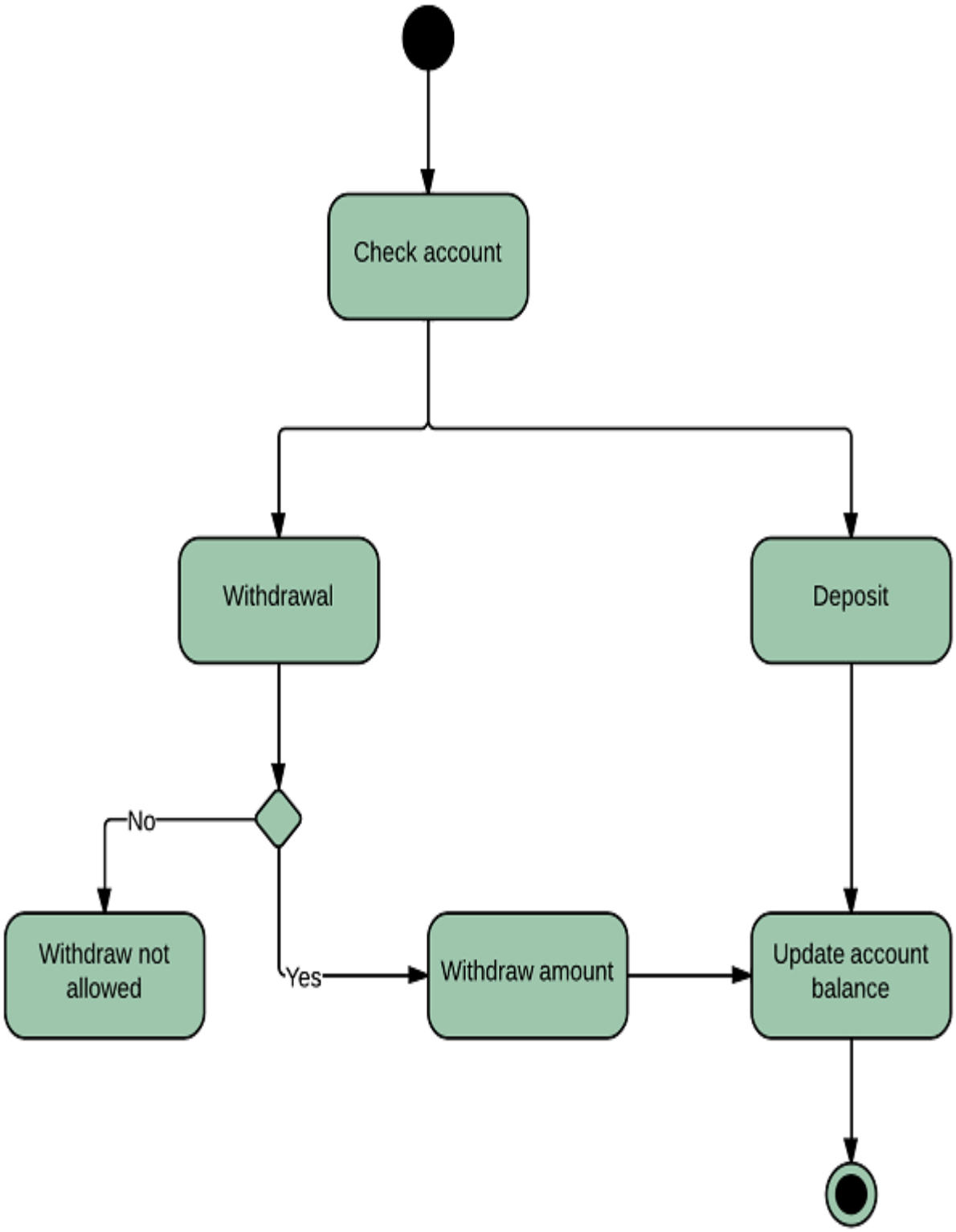


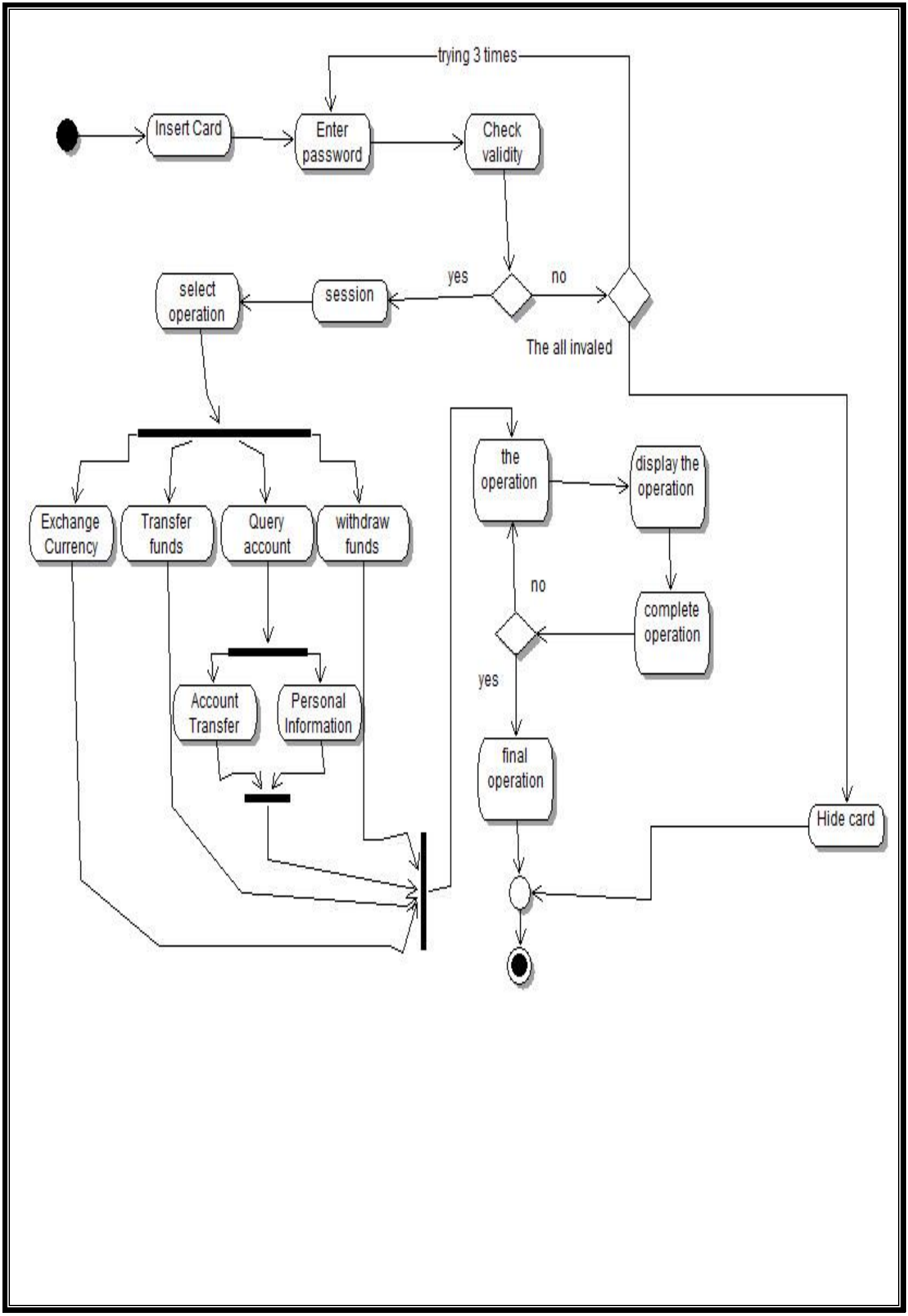
Data Flow Diagram of Banking System

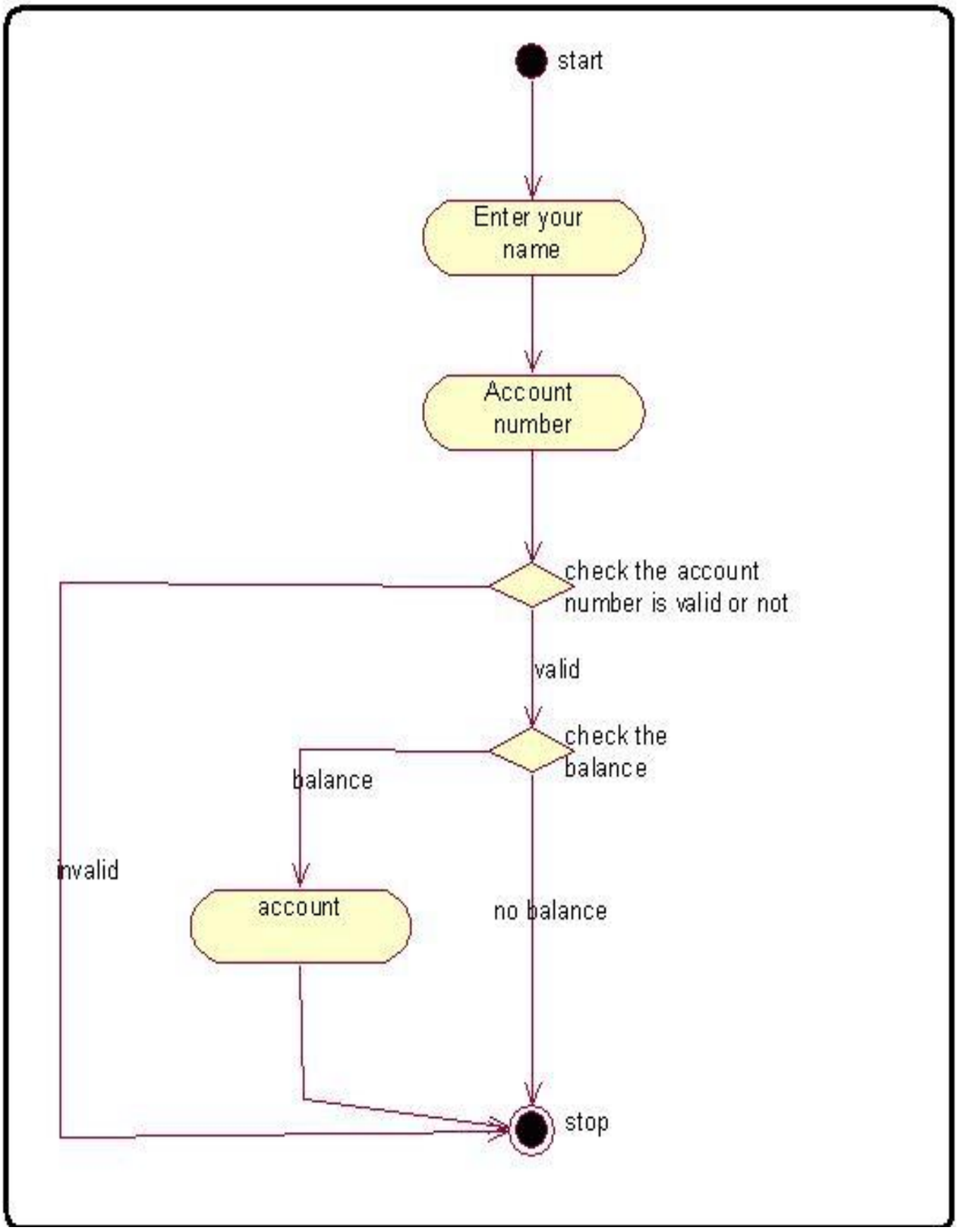


Level 0 DFD of a Banking System

Activity Diagrams of Banking System




trying 3 times
Insert Card
Enter password
Check validity
yes
no
session
select operation
The all invaled
Exchange Currency
Transfer funds
Query account
withdraw funds
Account Transfer
Personal Information
the operation
display the operation
complete operation
no
yes
final operation
Hide card


start
Enter your name
Account number
check the account number is valid or not
valid
check the balance
balance
invalid
account
no balance
stop

Testing and Results

The reason behind testing is to find errors. Every program or software has errors in it, against the common view that there are no errors in it if the program or software is working. Executing the programs with the intention of finding the errors in it is therefore testing; hence a successful test is one which finds errors. Testing is an activity; however it is restricted to being performed after the development phase is complete, but is carried parallel with all stages of system development, starting with requirement specification. A test case is a set of the data that a system will process as normal input. The software units developed in the system are modules and routines that are assembled and integrated to perform the required function of the system. Test results once gathered and evaluated, provide a qualitative indication of the software quality and reliability and serve as basis for design modification if required. The testing phase of the implementations works accurately and efficiently before live operation commences.

Unit Testing

The unit testing was done after the coding phase was done. The purpose of the unit testing was to locate errors on the module, independent of the other modules. Some changes in the coding were done during the testing. Finally all the modules were individually tested form bottom up starting with smallest and lowest modules and proceeding one at a time.

Black Box Testing

This method of software testing tests the functionality of an application as opposed to its internal structures or working. Specific knowledge of the internal structure and programming knowledge in general is not required. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. The test designer selects valid and invalid inputs and determines the correct output.

White Box Testing

This method of software testing tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are required and used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

Integration Testing

Once the unit was over, all the modules were integrated for integration testing. External and internal interfaces are implemented and work as per design, the performance of the module is not degraded.

Validation Testing

At the culmination of integration testing, software is said to be completely assembled as a package; interfacing errors have been uncovered and corrected. Then as a final series of software test, validation tests were carried out.

Acceptance Testing

This is the final stage in the testing process before the system is accepted for operational use. Any requirement problem or requirement definition problem revealed from acceptance testing are considered and made error free.

Conclusion

This Software Requirements Specification (SRS) specifies the requirements needed for the Banking System, which will be used in the Banks. This document will be used by the customer to ensure all specifications are correct and verified by the software Engineer to design the system. It deals with the internal banking functions like new account registration, withdrawal, deposit, account closure,& exclusively for the customers, who could access it from anywhere having an internet connection. The banking system uses a well interfaced GUI and well designed Web Forms for specific actions required by the users. It will need to be connected to a main database server for storing and retrieving the data of the customers.

This SRS would be used by the following people:

Bank Employee: They would be using the Core Banking Solution to perform the various banking functionalities.

Bank Customer: They would be using the E - Banking Solution to view their account details.

Research Students: Research students are advised to read all the section of this document to get an overall idea of the workflow and technicalities of the software.

Testers: It can be used as a documentation to know the interfaces.