**Software Requirements Specification for Banking Application System**

**Banking Management System**

**(SRS Report)**

****

**POORNIMA COLLEGE OF ENGINEERING**

SESSION (2023 - 2024) THIRD SEMESTER

**Submitted To:- Date:- 2nd December 2023 DATE:- 29TH NOVEMBER 2023**

**Ms. CHITRA THINGER**

Lecturer, Computer Science and Engineering department

**Submitted By:- SECTION:- A**

**AMAN SHIVDASANI** (PCE22CS012) **BATCH:- A-1**

**ABHEET GARG** (PCE22CS004) **BRANCH:- CSE**

**ANIRUDH JAIMAN** (PCE22CS190)

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Topic Name** | **Page No.** |
| **1.** | **Introduction** | **1.** |
| **2.** | **Literature survey** | **2.** |
| **3.** | **Requirements** | **3.** |
|  | **3.1 Functional Requirements** | **3.** |
|  | **3.2 Non- Functional Requirements** | **3.** |
|  | **3.2.1 Safety Requirements** | **3.** |
|  | **3.2.2 Security Requirements** | **4.** |
|  | **3.2.3 Software Quality Attributes** | **4.** |
|  | **3.3 Hardware Requirements** | **5.** |
|  | **3.4 Software Requirements** | **6.** |
|  | **3.5 WaterFall Model** | **7.** |
|  | **3.6 Feasibility Study** | **8.** |
|  | **3.6.1 Economic Feasibility** | **8.** |
|  | **3.6.2 Technical Feasibility** | **8.** |
|  | **3.6.3 Operational Feasibility** | **9.** |
| **4.** | **Data Dictionary** | **10.** |
|  | **4.1 Data Flow Diagram** | **11.** |
| **5.** | **Design and Implementation** | **12.** |
|  | **5.1 Class diagram design** | **13.** |
|  | **5.2 Use case diagram** | **13.** |
|  | **5.3 Sequence diagram** | **14.** |
|  | **5.4 E-R Diagram** | **15.** |
| **6.** | **State diagram** | **16.** |
| **7.** | **Activity Diagram** | **17.** |
|  | **Deployment Diagram** | **17** |
| **8.** | **Testing And Results** | **18.** |
|  | **8.1 Unit Testing** | **18.** |
|  | **8.2 Black Box Testing** | **18.** |
|  | **8.3 White Box Testing** | **19.** |
|  | **8.4 Integration Testing** | **19.** |
|  | **8.5 Validation Testing** | **19.** |
|  | **8.6 Acceptance Testing** | **19.** |
| **9.** | **Conclusion** | **20.** |
| **10.** | **Bibliography** | **21.** |

# CHAPTER-1 INTRODUCTION

=========================================================================

This document, Software Requirements Specification (SRS), is created to document the software requirements for the Banking Application System. A bank has several accounts, which are registered for customers and connected via a wide area network to a central server or Database. The Banking Application System can be used by customers to login on the server and to perform various operations through their account. Each Bank account can perform operations like, Check Balance, Withdraw Money, Deposit Fund, Transfer Fund, Payments Done, Payments Received and Loan. By using the Banking Application System, 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. The customer needs to login on the system if they have an existing account and if not then they need to sign- up. While signing up they need to provide the system with Mobile No. , Name, Balance, Age and Password. Thus their account is created and they are sent to Home page. The System then asks for account number and password. If the ACCOUNT NUMBER and PIN is validated satisfactorily, the customer is prompted for a check balance, withdrawal, query, loan, 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 the customer is notified regarding the transfer and lastly logged out. A customer may cancel a transaction at any time; the transaction is terminated and they are sent back to the home page. Customer records, account records, and records of past loans and dues are all maintained at the server.

=========================================================================

# CHAPTER 2

## 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. 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.

=======================================================================

# Chapter -3 Requirements

============================================================================

# Functional Requirements

* **Purpose**

To register a new customer

To login on an existing account

* **Inputs**

The required data for registration of a new customer in the bank (Like Name, Age, Balance, Mobile NO., Password etc.)

The required data for SIGN IN is your account number and password respectively.

* **Output**

A Success Message will be displayed on successful registration with your account number.

The customer would be successfully logged in on his account if the Account number and Password are correct and if not an error message would 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

* + - 1. Backup, recovery & business continuity Banks should ensure adequate back up of data as may be required by their operations.
      2. Both data and software should be backed up periodically.

#### Security Requirements

1. Account ID and Password (PIN) Protection
2. Password recovery
3. Sign-off feature
4. Failed Log-in Attempts
5. Encryption

#### Software Quality Attributes

1. **Reliability**

Measures if System is reliable enough to sustain in any condition. Should give consistently correct results. System reliability is measured in terms of working of project under different working environment and different conditions.

#### Maintainability

Different versions of the product should be easy to maintain. For development, its 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.

#### Usability

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

#### Portability

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

#### 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.

#### 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.

#### 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
* Better processor
* 2GB or more RAM (6 recommended)
* At least 500 MB Hardisk space.
* Smart mobile phone

# Software Requirements

This product is developed mainly using the concepts of Java, Sql, Git, Swing etc..

Backend: My Sql , Java Web Server: Apache 2.2

Platform used: Fedora Core 4 Linux, Windows XP / Windows7/ Windows Vista/Windows 11

Web Browser: Microsoft Internet Explorer 4.0, Mozilla ,Google Chrome or later

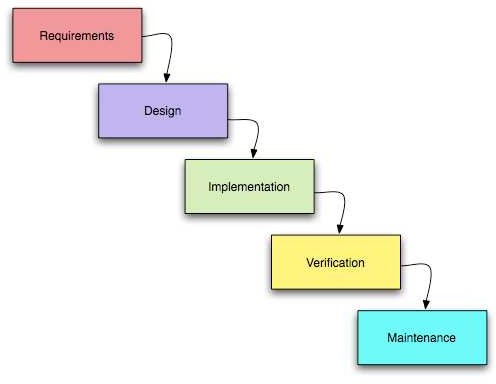
# WaterFall Model

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.



**Figure: 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 software.

-The benefits in the form of reduced costs or fewer costly errors.

-Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.*.*

#### 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.

Though 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.

#### OPERATATIONAL 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 it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

==========================================================================

# Chapter -4 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 pro 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.

#### -------------------------------------------------------------------------------------------------------------

**Data Dictionary Actor Description TEMPLATE**

|  |  |
| --- | --- |
| **Name** | Banking Application User, Admin |
| **Alternate Names** | AMAN SHIVDASANI |

**Input Data**

|  |  |
| --- | --- |
| **Name of Use Case** | **Inputs to the System** |
| Withdraw cash | Login on the system, Enters pin |
| Transfer funds | Login on the system, Enters pin |
| Check Balance | Login on the system, Enters pin |
| Deposit cash | Cash amount, Login on the system, Enters pin |

**Output Data**

|  |  |
| --- | --- |
| **Name of Use Case** | **Outputs from the System** |
| Withdraw cash | Get cash amount, Get receipt |
| Check Balance | Get status receipt |
| Transfer funds | Get receipt |
| Deposit cash | Get receipt |

**Description**

The customer operates his bank account (for Withdrawing cash, Check Balance, Transferring funds, Deposit cash) by logging in on the system.

After logging in, he performs necessary task like entering valid pin and selecting required option.

He can also use the banks Toll Free No. for any query related to his bank account.

**Comments**

The customers are the common people who have accounts in bank and so they are issued a valid Account No. and Password from bank so that they can operate there account from remote places.

# Data Flow Diagrams:

=======================================================================

# Chapter 5

**DESIGN & IMPLEMENTATION**

============================================================================

* + - The System 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 application system:

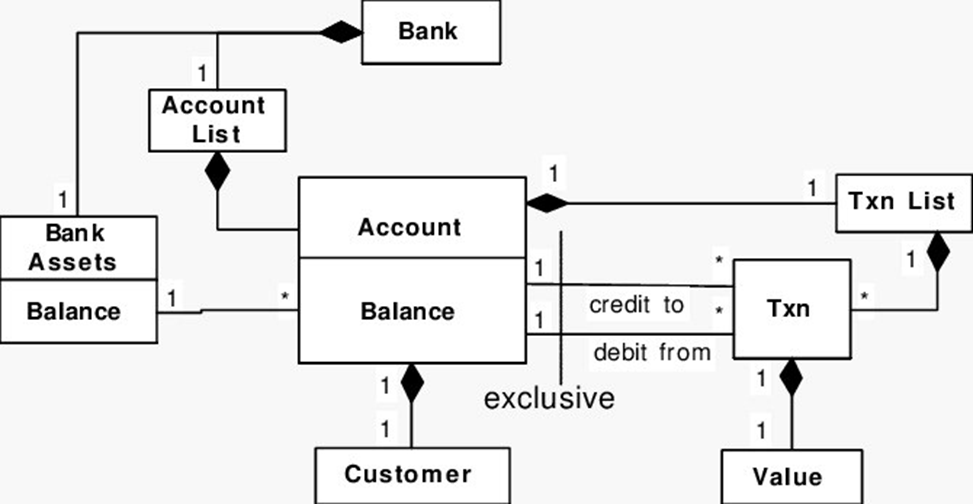
The overall view (design and implementation) of the banking application System is as shown below:

### Class Diagram

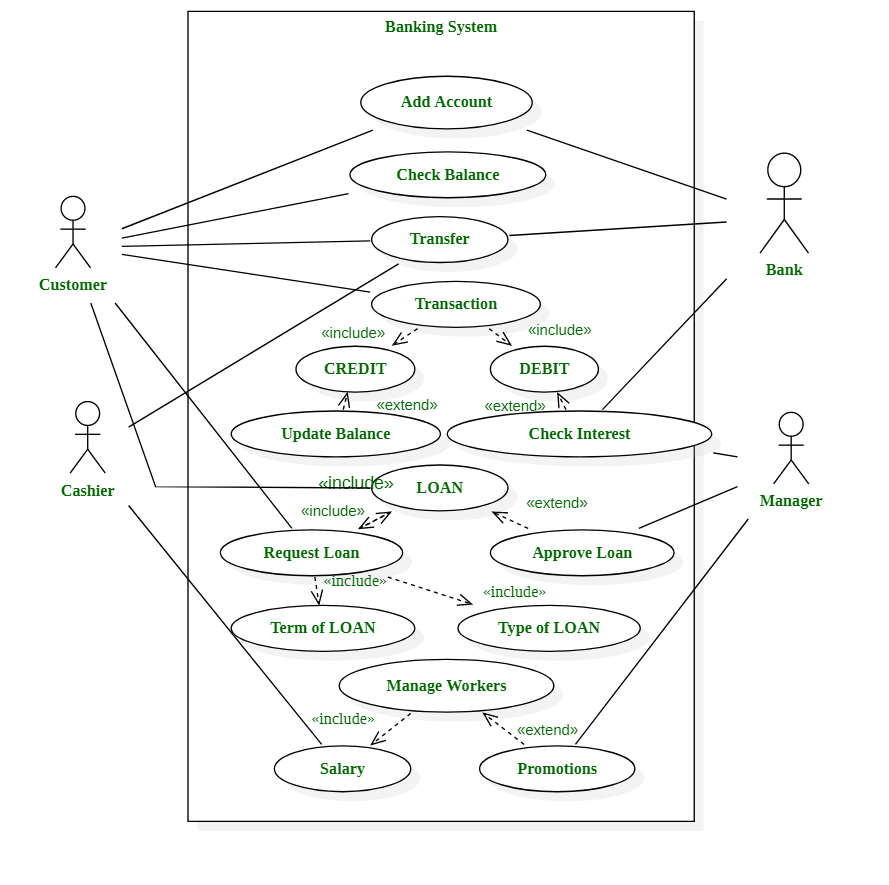
1. Use-case Diagram

### Sequence Diagram

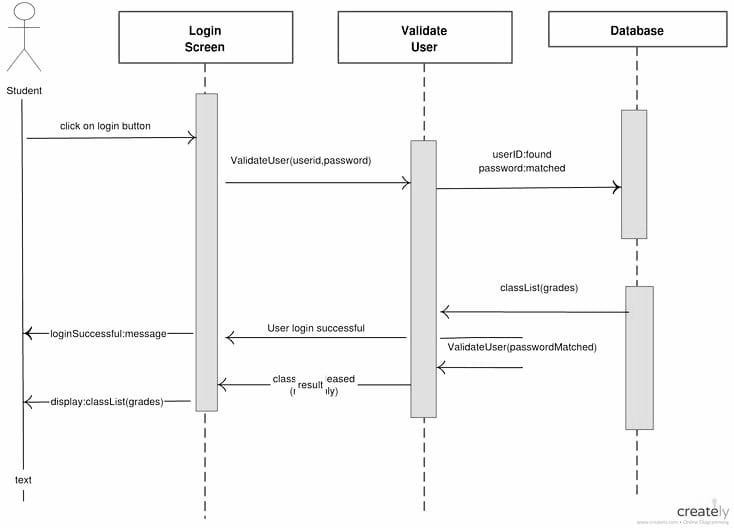
1. E-R Diagram
   1. **Class Diagram:**

****

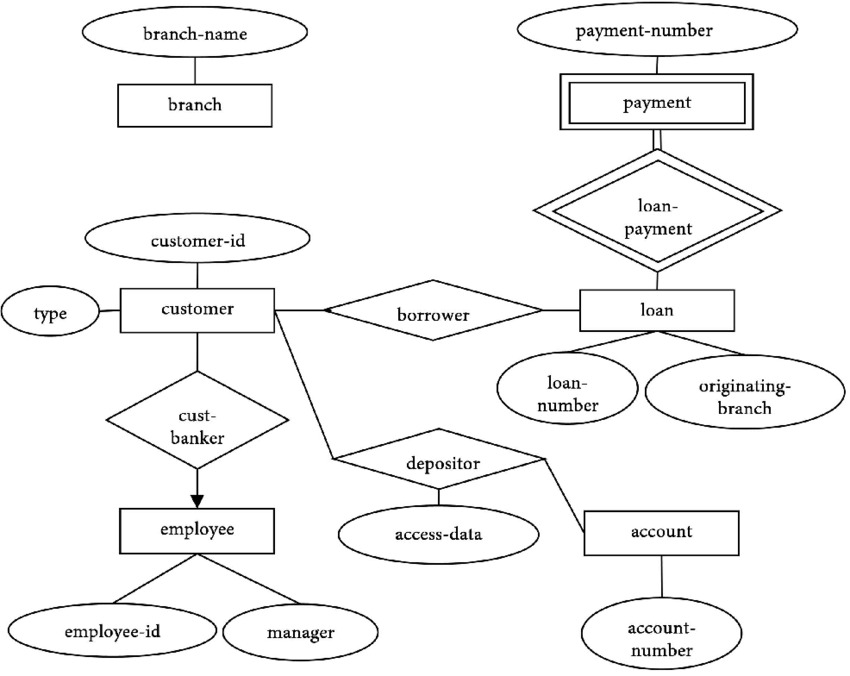
* 1. **Use-Case Diagram:**



## Sequence Diagram :



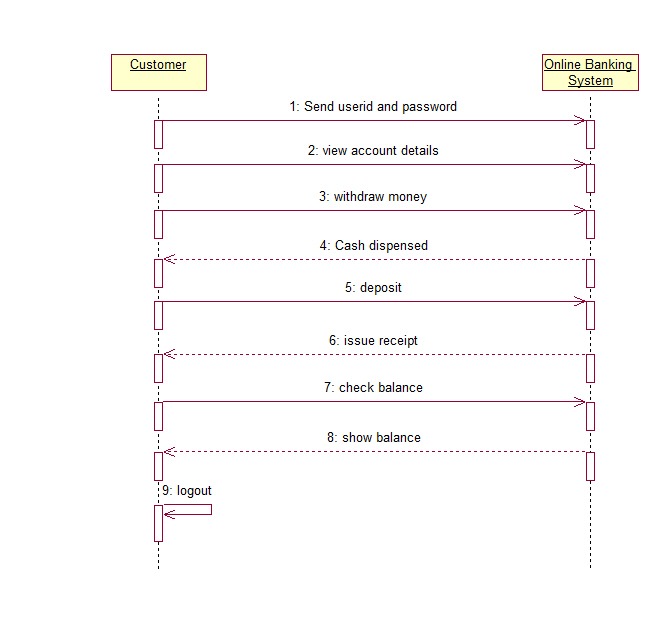
* 1. **E-R [Entity-Relationship] Diagram :**

****

==========================================================================

# Chapter -6 State Diagrams

### The State Diagram for the Banking System can show as:

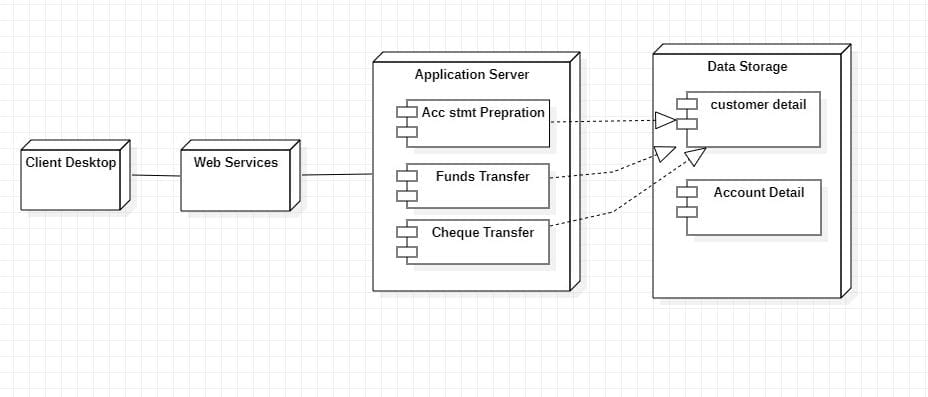


# Chapter -7 Activity Diagram

### The Activity Diagram for the Banking System can show as:

Page 17

**Deployment Diagram**

****

**CHAPTER-8 TESTING AND RESULT**

==========================================================================

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 from 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.

## 6 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.

=========================================================================

# Chapter -9

## CONCLUSION

==========================================================================

This Software Requirements Specification (SRS) specifies the requirements needed for the Banking Application 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 Customers:** They would be using the e­Banking 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.

========================================================================

SRS For Banking System

# CHAPTER-10 BIBLIOGRAPHY

=========================================================================

## Websites:

1. [www.w3schools.com](http://www.w3schools.com/)
2. [www.goodphptutorials.com](http://www.goodphptutorials.com/)
3. [www.homeandlearn.co.uk](http://www.homeandlearn.co.uk/)
4. [www.freewebmasterhelp.com](http://www.freewebmasterhelp.com/)
5. [www.stackoverflow.com](http://www.stackoverflow.com/)
6. [www.developerfusion.co.uk](http://www.developerfusion.co.uk/)

========================================================================

Page 21