

BASIC BANKING APPLICATION

PROBLEM STATEMENT:

A Banking concern wants to computerize its Banking Services. Bank Management is generally a very complicated task. To reduce the complexity, it wants to computerize its Banking services and for this purpose it wants application, which will handle all its banking activities.

As the bank for which the software is to be created provides loan and, the various activities handled by the bank is provision of loan, and handling the regular banking services. The software created must allow the user to handle each of the service provided by the bank.

BANK MANAGEMENT

SOFTWARE REQUIREMENTS SPECIFICATION:

1. INTRODUCTION:

This document is to provide the software requirement specification of the computerized solutions for certain banking services. It keeps track of the account holder's details.

1.1. Purpose:

The main purpose of this software is to carry out all banking tasks like transactions and maintaining accounts.

1.2. Scope:

The software has the ability to manage large number of accounts and keep a track of the banking activities.

1.2.1. Benefits:

- Maintains all banking activities with no error.
- Graphical user interface.
- Customer friendly.

1.2.2. Objectives:

- Provides fully automated banking services.
- Efficient functioning of the bank.

1.3. Reference:

The books and reference materials used during this stages of the project include:

- www.cplusplus.com : For all coding syntax and their uses.
- GTKMM tutorials : To learn graphics in c++.
- Sumita arora : Introduction to c++.

1.4. Overview:

The rest of this document gives an overview of the feature and functions of the bank management application.

2. OVERALL DESCRIPTION:

2.1. PRODUCT PERSPECTIVE:

The “Bank Management” is an application that remains to be independent. The user interfaces and the hardware interfaces related with this software is defined below:

2.1.1. User Interfaces:

The screen formats and the menu structure should be in such a way that the user should find it easy to use the application. The application should be user-friendly. It should have high security.

2.1.2. Software Interfaces:

<i>OS</i>	<i>NAME</i>	<i>SOURCE</i>
LINUX	Codeblocks with integrated GTKMM libraries	Codeblocks.org
WINDOWS	Codeblocks with integrated GTKMM libraries	Codeblocks.org

2.2. PRODUCT FUNCTIONS:

The main functions are:

- Create an account.
- Carries out transactions
- *Delete an account.*

- Display the account holders' details.
- Modify the account holder's details.

2.3. USER CHARACTERISTICS:

There is no need for any experience or technical knowledge. Everything is graphics based and easy to use.

3. SPECIFIC REQUIREMENTS:

3.1. EXTERNAL INTERFACE REQUIREMENTS:

3.1.1. User Interfaces:

The interface used in GUI must be easy to understand. This interface serves as a bridge between the user and the application. It also makes the user interaction with the system easy.

The user interface includes:

- Screen formats :

The introductory screen will be the first to be displayed which allows the user to log in using their id and password or to create a new account.

- Windows formats:

When the user chooses a particular topic then the information pertaining to that topic will be displayed in a new window, which will allow multiple windows to be available on the screen, and the user can switch between them.

- Data Format:

The data entered by the user will be alphanumeric and numeric for mobile numbers.

3.2. SOFTWARE PRODUCT FEATURES:

3.2.1. FEATURE 1: CREATING A NEW ACCOUNT:

The purpose of this feature is to create a new account.

3.2.1.1. Purpose:

The purpose of this feature is to enable the customer to enter his personal details .

3.2.1.2. Response sequence:

Once the option is chosen, a window will appear asking for all personal details of the account holder as to begin a new account.

3.2.1.3. Associated functionality requirement:

3.2.1.3.1. Introduction:

The user must be provided with the text and entry box.

3.2.1.3.2 Inputs:

The user has to enter the, name of the account holder, address, phone number, date of birth, and the opening balance.

3.2.1.3.3 Outputs:

A Success message box if everything is correct or re-entering the details if some data entered is invalid.

3.2.2.FEATURE 2:CREDIT/DEBIT (Making transactions):

This feature is to keep a track on the transaction made by the account holder.

3.2.2.1. Purpose:

The purpose of this feature is to keep on the transactions made by the account holder. If the A/C holder credits some amount, the amount gets added up to his account. If he withdraws some amount from his account, then that amount is subtracted from his account. It's compulsory that the account holder must maintain a minimum balance of Rs. 1000 in his account.

3.2.2.2. Response Sequence:

Once the user chooses that option, the page corresponding to that is to be displayed.

3.2.2.3. Associated Functionality Requirements:

3.2.2.3.1. Functionality Requirement: Transactions:

3.2.2.3.1.1. Introduction:

A page concerned with transaction details has to be displayed.

3.2.2.3.1.2. Inputs:

The user must enter the nature of transaction made (whether credit or debit) and the amount.

3.2.2.3.1.3. Processing:

Considering the given input to be correct if the nature of transaction is credit then the amount is added to the account else it taken from the account.

3.2.2.3.1.4. Outputs:

The required change is made in the account and a success message box or else an error message if given amount to be withdrawn is more than in his account balance.

3.2.3.FEATURE 3:FIXED DEPOSIT:

This feature is to get a Fixed Deposit to your account

3.2.2.1. Purpose:

The purpose of this feature is get a Fixed Deposit on your account, which can be redeemed 3 years later at an interest rate of 8% per annum.

3.2.2.2. Response Sequence:

Once the user chooses that option, the page corresponding to that is to be displayed asking for fixed deposit details.

3.2.2.3. Associated Functionality Requirements:

3.2.2.3.1. Functionality Requirement:

3.2.2.3.1.1. Introduction:

A page concerned fixed deposit details has to be displayed.

3.2.2.3.1.2. Inputs:

The user must enter the amount of money that has to be Fixed Deposited.

3.2.2.3.1.3. Processing:

Considering the given input to be correct, the amount is added to the Fixed Deposit balance.

3.2.2.3.1.4. Outputs:

The required change is made in the account and a success message box.

3.2.4. FEATURE 4:-DISPLAYING AND MODIFYING A/c HOLDERS' DETAILS:

3.2.4.1. Purpose:

The purpose is to enable the user to view the details of the his /her accounts and modify them if required.

3.2.4.2. Response:

When the customer requests to view the details of a his account , a screen prompts viewing his account details and option to edit them. Name and Account number cant be modified

3.2.5. FEATURE 5:-DELETING ACCOUNTS:

3.2.5.1. Purpose:

The purpose is to enable the user to close his account from the bank.

3.2.5.2. Response:

When the customer requests to close his account , a screen prompts asking for confirmation.

3.2.6.FEATURE 6:LOAN facilities:

This feature is to take loan from the bank in need for different purposes(home/personal).

3.2.2.1. Purpose:

The purpose of this feature is to provide users with loan at an interest of 10% per annum and keep a track of the date the user took loan for a period of 5 years. It calculates the EMI to be paid by the user afterwards.

3.2.2.2. Response Sequence:

Once the user chooses that option, the page corresponding to that is to be displayed asking for loan details whether he wants a loan or pay his EMI .

3.2.2.3. Associated Functionality Requirements:

3.2.2.3.1. Functionality Requirement: loan:

3.2.2.3.1.1. Introduction:

A page concerned loan details and its terms and conditions has to be displayed.

3.2.2.3.1.2. Inputs:

The user must enter the amount of loan and other details mentioned.

3.2.2.3.1.3. Outputs:

The required change is made in the account and a success message box for loan sanction appears on the screen.

3.3. PERFORMANCE REQUIREMENTS:

A large database should be present as large data will be stored.

3.4. DESIGN CONSTRAINTS:

- Based completely on windows and linux functionality performance with codeblocks integrated with GTKMM graphics library.
- The software must be valid and accessible only by authorized users.

DATA FLOW DIAGRAM:



