# Bank Management System

**Declaration**

I hereby declare that the project work entitled “Bank Management System”, submitted to the Department of computer science, Kendriya Vidyalaya Tarakeswar, is prepared by me. All coding is the result of my personal efforts.

**Swagato Bag**

**Class – XII**

**Certificate**

The project report entitled

**“Bank Management System”**

Submitted by **SWAGATO BAG** of class XII for the CBSE Senior Secondary Examination class XII of Computer Science at Kendriya Vidyalaya Tarakeswar, Hooghly has been examined.

---------------------------

Signature of Examiner

**Certificate**

This is to certify that **Swagato Bag** of class XII has prepared the report on the project entitled “Bank Management System”. The report is the result of his efforts and endeavors. The report is found worthy of acceptance as final project report for the subject computer science of class XII. He has prepared the report under my guidance.

-------------------------- --------------------------

Signature of Teacher Signature of Principal

Mr. Ravi Ranjan Mr. Sujit Paul Tirkey

**Acknowledgement**

It is really fine to have such an opportunity to realise the feeling of my gratitude impression in my hearts.

It is difficult to express in words my indebtedness to those intellectual whose co-operation, guidance and suggestions received in presenting this dissertation report in this far, where words fail to express the inner most feeling of gratitude towards my esteemed guide, our chemistry teacher Debasish Dutta Sir, who has been very kind indispensing their valuable suggestion and through guidance. Without him the work could not have been a leaving reality and it would be my no voice to express the deep gratitude to my esteemed guide our chemistry teacher Debasish Dutta Sir.

I am immensely grateful to my friend for their help & Co-operation.

I am also thankful to all staff of our school, Library for their help and co-operation during the completion of project.

No words can adequately express my debt of gratitude to my mother for generating in me a perennial interest in higher studies.

**Thank You**

**TABLE OF CONTENTS**

* Abstract
* Header files and their purpose
* Modules
* Function point analysis
* Requirement point analysis
* Implementation
* Maintenance
* C++ Code
* Requirements
* Bibliography

**ABSTRACT**

Requirements definition and management is recognized as a necessary step in the delivery of successful system s and software projects, discipline is also required by standards, regulations, and quality improvement initiatives. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment.

The “BANK MANAGEMENT SYSTEM” undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for bank management system. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using VB language and. Hence it provides the complete solution for the current management system.

**Header Files Used and Their Purpose**

1. **fstream.h** – for file handling.
2. **conio.h** – for clrscr() and getch() functions.
3. **stdio.h** – for standard i/o operations.
4. **ctype.h** – for character handling.
5. **iostream.h –** for basic I/O operation like cin/cout.
6. **iomanip.h –** for manipulation of lines and outputs.

**MODULES**

1. **Module 1 - New Account** :

A module new account is literally the form for the customer to open a new account. A new account is opened with the following details of the customer, with the account number given by the user.

* Name.
* Type of account – savings or current.
* Initial deposit amount.

1. **Module 2 - Deposit Amount :**

Deposition should be done each time the customer deposits a particular amount for an account. Deposition is done in the account after the following details.

* Account number.
* Amount to deposit.

1. **Module 3 - Withdrawal Amount** :

Using this withdrawal module, the user can withdraw a particular amount for an account. Withdrawal can be done after getting the following details:

* Account number.
* Amount to be withdraw.

1. **Module 4 - Balance Enquiry** :

Using this balance enquiry module, the user can check the balance of a particular account. Balance enquiry can be done after getting the following details:

* Account number.

1. **Module 5 - All account holder list :**

Using this module, the administrator can see all the user accounts’ details.

1. **Module 6 - close an account** :

Using this close an account module, the user can close their account by providing their account number.

1. **Module 7 - Modify an account** :

This module is used to modify the account that we want this done after receiving the account number. Then the corresponding name, type of account and balance are modified.

1. **Module 8 - Exit** :

This module is used to leave the program interface.

**Function Point Analysis**

Function points measure software size by qualifying the functionality provided to the user based solely on logical design and functional specifications. With this in mind, the objectives of FP counting are to:

* Measures functionality that the user requests and receivers.
* Provide a normalized measure across projects and organizations.

A “function point” is one standard unit of software size and complexity. Most software is decomposable into function points, which can then be counted, giving its size and complexity. Standards for sizing software using function points are in the International Function Points Users Group (IFPUG) Function Points Counting Practices Manuals.

**Requirement Analysis**

Requirements are prone to issues of ambiguity, incompleteness, and inconsistency. Techniques such as rigorous inspection have been shown to help deal with these issues. Ambiguities, incompleteness, and inconsistencies that can be resolved in the requirements phase typically cost orders of magnitude less to correct than when these same issues are found in later stages of product development. Requirements analysis strives to address these issues: -

* Take a long time to produce
* Begin to limit the implementation option available
* Are costly to produce requirements for both the system and the software are documented and reviewed with the customer.

**Implementation**

Implementation is the realization, application, or execution of a plan, idea, model, design, specification, standard algorithm, or policy.

The design must be translated in to machine-readable form. The code generation step performs their tasks. If design is performed in a detailed manner, code generation can be accomplished mechanistically.

**Maintenance**

In software, software maintenance is the process of enhancing and optimizing developed software (software release), as well as remedying defects. Software maintenance is one of the phases in the software development process, and follows development of software into the field. The software maintenance phase involves changes to the software in order to correct defects and deficiencies found during field usage as well as the addition of new functionality to improve software’s usability and applicability.

**Requirements**

1. **Hardware Required:**
2. Printer, to print the required documents of the project
3. Compact Drive.
4. Processor: - Pentium III.
5. Ram: - 32 MB.
6. Hard disk: - 20 Gb.
7. **Softwires required:**
8. Operating system: Windows 95 or higher.
9. Dev C++, for execution of program.
10. Download page - https://filehippo.com/download\_dev-c/
11. MS word, for presentation of output.

**C++ Code**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// HEADER FILE USED IN PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<iostream>

#include<fstream>

#include<cctype>

#include<iomanip>

#include<conio.h>

#include<stdio.h>

using namespace std;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// CLASS USED IN PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class account

{

int acno;

char name[50];

int deposit;

char type;

public:

void create\_account(); //function to get data from user

void show\_account() const; //function to show data on screen

void modify(); //function to add new data

void dep(int); //function to accept amount and add to balance amount

void draw(int); //function to accept amount and subtract from balance amount

void report() const; //function to show data in tabular format

int retacno() const; //function to return account number

int retdeposit() const; //function to return balance amount

char rettype() const; //function to return type of account

}; //class ends here

void account::create\_account()

{

cout<<"\nEnter Account No. :";

cin>>acno;

cout<<"\n\nEnter The Name of account Holder : ";

cin.ignore();

cin.getline(name,50);

cout<<"\nEnter Type of The account (C/S) : ";

cin>>type;

type=toupper(type);

cout<<"\nEnter The Initial amount(>=500 for Saving and >=1000 for current ) : ";

cin>>deposit;

cout<<"\n\n\nAccount Created..";

}

void account::show\_account() const

{

cout<<"\nAccount No. : "<<acno;

cout<<"\nAccount Holder Name : ";

cout<<name;

cout<<"\nType of Account : "<<type;

cout<<"\nBalance amount : "<<deposit;

}

void account::modify()

{

cout<<"\nAccount No. : "<<acno;

cout<<"\nModify Account Holder Name : ";

cin.ignore();

cin.getline(name,50);

cout<<"\nModify Type of Account : ";

cin>>type;

type=toupper(type);

cout<<"\nModify Balance amount : ";

cin>>deposit;

}

void account::dep(int x)

{

deposit+=x;

}

void account::draw(int x)

{

deposit-=x;

}

void account::report() const

{

cout<<acno<<setw(10)<<" "<<name<<setw(10)<<" "<<type<<setw(6)<<deposit<<endl;

}

int account::retacno() const

{

return acno;

}

int account::retdeposit() const

{

return deposit;

}

char account::rettype() const

{

return type;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function declaration

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void write\_account(); //function to write record in binary file

void display\_sp(int); //function to display account details given by user

void modify\_account(int); //function to modify record of file

void delete\_account(int); //function to delete record of file

void display\_all(); //function to display all account details

void deposit\_withdraw(int, int); // function to desposit/withdraw amount for given account

void intro(); //introductory screen function

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THE MAIN FUNCTION OF PROGRAM

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main()

{

char ch;

int num;

intro();

do

{

system("cls");

system("color b1");

cout<<"\n\n\n\t\t\t\t MAIN MENU"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\n\n\t01. NEW ACCOUNT";

cout<<"\n\n\t02. DEPOSIT AMOUNT";

cout<<"\n\n\t03. WITHDRAW AMOUNT";

cout<<"\n\n\t04. BALANCE ENQUIRY";

cout<<"\n\n\t05. ALL ACCOUNT HOLDER LIST";

cout<<"\n\n\t06. CLOSE AN ACCOUNT";

cout<<"\n\n\t07. MODIFY AN ACCOUNT";

cout<<"\n\n\t08. EXIT";

cout<<"\n\n\tSelect Your Option (1-8) ";

cin>>ch;

system("cls");

switch(ch)

{

case '1':

system("color f1");

write\_account();

break;

case '2':

system("color f2");

cout<<"\n\n\tEnter The account No. : "; cin>>num;

deposit\_withdraw(num, 1);

break;

case '3':

system("color f3");

cout<<"\n\n\tEnter The account No. : "; cin>>num;

deposit\_withdraw(num, 2);

break;

case '4':

system("color f4");

cout<<"\n\n\tEnter The account No. : "; cin>>num;

display\_sp(num);

break;

case '5':

system("color f5");

display\_all();

break;

case '6':

system("color f6");

cout<<"\n\n\tEnter The account No. : "; cin>>num;

delete\_account(num);

break;

case '7':

system("color f1");

cout<<"\n\n\tEnter The account No. : "; cin>>num;

modify\_account(num);

break;

case '8':

system("color f2");

cout<<"\n\n\tThanks for using bank managemnt system";

break;

default :cout<<"\a";

}

cin.ignore();

cin.get();

}while(ch!='8');

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to write in file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void write\_account()

{

account ac;

ofstream outFile;

outFile.open("account.dat",ios::binary|ios::app);

ac.create\_account();

outFile.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

outFile.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to read specific record from file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void display\_sp(int n)

{

account ac;

bool flag=false;

ifstream inFile;

inFile.open("account.dat",ios::binary);

if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\nBALANCE DETAILS\n";

while(inFile.read(reinterpret\_cast<char \*> (&ac), sizeof(account)))

{

if(ac.retacno()==n)

{

ac.show\_account();

flag=true;

}

}

inFile.close();

if(flag==false)

cout<<"\n\nAccount number does not exist";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to modify record of file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*void modify\_account(int n)

{

bool found=false;

account ac;

fstream File;

File.open("account.dat",ios::binary|ios::in|ios::out);

if(!File)

{

cout<<"File could not be open !! Press any Key...";

return;

}

while(!File.eof() && found==false)

{

File.read(reinterpret\_cast<char \*> (&ac), sizeof(account));

if(ac.retacno()==n)

{

ac.show\_account();

cout<<"\n\nEnter The New Details of account"<<endl;

ac.modify();

int pos=(-1)\*static\_cast<int>(sizeof(account));

File.seekp(pos,ios::cur);

File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

cout<<"\n\n\t Record Updated";

found=true;

}

}

File.close();

if(found==false)

cout<<"\n\n Record Not Found ";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to delete record of file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void delete\_account(int n)

{

account ac;

ifstream inFile;

ofstream outFile;

inFile.open("account.dat",ios::binary);

if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

return;

}

outFile.open("Temp.dat",ios::binary);

inFile.seekg(0,ios::beg);

while(inFile.read(reinterpret\_cast<char \*> (&ac), sizeof(account)))

{

if(ac.retacno()!=n)

{

outFile.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

}

}

inFile.close();

outFile.close();

remove("account.dat");

rename("Temp.dat","account.dat");

cout<<"\n\n\tRecord Deleted ..";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to display all accounts deposit list

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*void display\_all()

{

account ac;

ifstream inFile;

inFile.open("account.dat",ios::binary);

if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\n\n\t\tACCOUNT HOLDER LIST\n\n";

cout<<"================================================================================\n";

cout<<"A/c no. NAME Type Balance\n";

cout<<"================================================================================\n";

while(inFile.read(reinterpret\_cast<char \*> (&ac), sizeof(account)))

{

ac.report();

}

inFile.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to deposit and withdraw amounts

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void deposit\_withdraw(int n, int option)

{

int amt;

bool found=false;

account ac;

fstream File;

File.open("account.dat", ios::binary|ios::in|ios::out);

if(!File)

{

cout<<"File could not be open !! Press any Key...";

return;

}

while(!File.eof() && found==false)

{

File.read(reinterpret\_cast<char \*> (&ac), sizeof(account));

if(ac.retacno()==n)

{

ac.show\_account();

if(option==1)

{

cout<<"\n\n\tTO DEPOSITE AMOUNT ";

cout<<"\n\nEnter The amount to be deposited";

cin>>amt;

ac.dep(amt);

}

if(option==2)

{

cout<<"\n\n\tTO WITHDRAW AMOUNT ";

cout<<"\n\nEnter The amount to be withdraw";

cin>>amt;

int bal=ac.retdeposit()-amt;

if((bal<500 && ac.rettype()=='S') || (bal<1000 && ac.rettype()=='C'))

cout<<"Insufficience balance";

else

ac.draw(amt);

}

int pos=(-1)\*static\_cast<int>(sizeof(ac));

File.seekp(pos,ios::cur);

File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

cout<<"\n\n\t Record Updated";

found=true;

}

}

File.close();

if(found==false)

cout<<"\n\n Record Not Found ";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// INTRODUCTION FUNCTION

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void intro()

{

system("color a1");

cout<<"\n\n\n\t\t\t BANK MANAGEMENT SYSTEM"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\n\n\nMADE BY : Swagato Bag."<<endl;

cout<<"\nSCHOOL : Kendriya Vidyalaya Tarakeswar."<<endl;

cout<<"\n\n\nSystem is paused"<<endl;

cout<<"Press any key to continue...........";

cin.get();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// END OF PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Bibliography**

* 1. **Book:**
     1. Computer science in C++ by: – Sumita Arora.
     2. Let us C++ by: - Yashavant Kanetkar.
  2. **Website:**
     1. https://codescracker.com/cpp/
     2. https://www.hackerrank.com