

K L UNIVERSITY
FRESHMAN ENGINEERING DEPARTMENT

Project Based Lab Report

On
Banking Management System

SUBMITTED BY:

ID NUMBER **NAME**

2300031039 Keerti Krishna Sreenivas S

UNDER THE ESTEEMED GUIDANCE OF

Mrs. I Aswani
Assistant Professor, CSE



KL UNIVERSITY
Green fields, Vaddeswaram – 522 502
Guntur Dt., AP, India.

DEPARTMENT OF BASIC ENGINEERING SCIENCES



CERTIFICATE

This is to certify that the project based laboratory report entitled “Banking Management System” submitted by **Mr. Keerti Krishna Sreenivas S** bearing Regd. No. 2300031039 to the **Department of Basic Engineering Sciences, KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in ”**DATA STRUCTURES-23SC1202**” course in I B Tech II Semester, is a Bonafede record of the work carried out by him/her under my supervision during the academic year 2023– 2024.

PROJECT SUPERVISOR

Mrs. I.ASWANI

HEAD OF THE DEPARTMENT

Dr. D. HARITHA

ACKNOWLEDGEMENTS

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project-based laboratory report.

I express sincere gratitude to our Director **Dr. A. Jagdeesh** for his administration towards our academic growth.

I express sincere gratitude to our Coordinator **Dr. MSG Prasad** for his leadership and constant motivation towards our academic growth.

I express sincere gratitude to HOD-BES **Dr. D. Haritha** for her leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank you for providing us with the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **Mrs. I. Aswani** for her novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, we are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to making this project a success.

Name:

Keerti Krishna Sreenivas S

ID Number:

2300031039

ABSTRACT

Banking system

The presented C code outlines the functionality of a basic banking system, offering a range of services including account creation, viewing account details, depositing and withdrawing funds, and assessing loan eligibility. Upon account creation, users are prompted to input their name, initial balance, and monthly income, with the system automatically generating a unique account number. Transactions, such as deposit and withdrawal, are facilitated by users inputting their account number, along with a corresponding password, and specifying the desired amount. Moreover, the system provides a feature for users to determine their eligibility for a loan, where they are prompted to enter their account number and monthly income. Notably, eligibility for the loan is contingent upon a monthly income exceeding 30,000 Rs. If met, the system congratulates the user on their eligibility; otherwise, it notifies them of their ineligibility. Overall, this banking system serves as a fundamental tool for account management, while also incorporating a loan eligibility check to assist users in their financial planning and decision-making processes.

INDEX

S.NO	TITLE	PAGE NO
1.	Introduction	6
2.	Aim of the Project	7
2.1	Advantages & Disadvantages	7
2.2	Future Implementation	8
3.	Software & Hardware Details	9
4.	Data Flow Diagram	10
5.	Algorithm for each Module	11
6.	Implementation	12
7.	Integration and System Testing	19
8.	Conclusion	22

INTRODUCTION

The banking sector plays a pivotal role in modern economies by facilitating financial transactions and providing various financial services to individuals and businesses. With the advancement of technology, the traditional banking system has evolved into digital platforms, offering customers more convenient ways to manage their finances. In this context, the development of a comprehensive banking system becomes essential to meet the diverse needs of customers and ensure efficient banking operations.

The purpose of this project is to design and implement a comprehensive banking system with transactional features and loan eligibility assessment. The system aims to provide users with essential banking functionalities, including account creation, balance inquiries, deposits, withdrawals, and loan eligibility checks. By integrating these features into a single platform, users can conveniently perform banking tasks and access financial services from anywhere with internet connectivity.

This project will involve the development of a user-friendly interface for interacting with the banking system, ensuring ease of navigation and accessibility for users of varying technical proficiency. Additionally, robust security measures will be implemented to safeguard user data and financial transactions against potential threats and unauthorized access.

Through this project, users will benefit from a convenient and efficient banking experience, allowing them to manage their finances effectively and access essential banking services without the constraints of traditional banking hours or physical branch locations. Moreover, the loan eligibility assessment feature will enable users to determine their eligibility for loans based on predefined criteria, facilitating informed financial decision-making.

Overall, the implementation of this comprehensive banking system will contribute to enhancing financial inclusion, promoting digital banking adoption, and improving overall banking efficiency and customer satisfaction.

AIM

The aim is to create a user-friendly digital banking system with account management, fund transfers, loan applications, and eligibility assessment, prioritizing convenience, security, and financial inclusion.

Advantages: -

Convenience: Customers can perform various banking tasks, such as account management, fund transfers, and loan applications, from the comfort of their homes or on the go using digital devices connected to the internet. This eliminates the need to visit physical bank branches, saving time and effort.

Accessibility: The system provides accessibility to banking services to a broader population, including those residing in remote areas or with limited mobility. By offering an intuitive user interface and support for multiple languages, the system ensures inclusivity and accommodates users from diverse backgrounds.

Efficiency: Automation of banking processes streamlines operations, reduces manual errors, and accelerates transaction processing. This leads to faster fund transfers, quicker loan approvals, and prompt resolution of customer inquiries, enhancing overall operational efficiency.

Security: Robust security measures, such as data encryption, multi-factor authentication, and real-time fraud detection, safeguard customer information and financial transactions against unauthorized access and fraudulent activities. This instills trust and confidence among customers regarding the safety of their funds and personal data.

Financial Management: The system offers features for monitoring account balances, tracking transaction history, setting up automated bill payments, and generating financial reports. This empowers customers to manage their finances effectively, budget their expenses, and make informed financial decisions.

Disadvantages: -

The disadvantages include susceptibility to security breaches due to inherent vulnerabilities in digital systems, dependence on technology that is prone to malfunctions or cyber-attacks, and limited accessibility for individuals without internet connectivity or digital literacy, thereby potentially hindering financial inclusion efforts.

Future enhancements: -

Future enhancements may encompass integration of advanced security measures such as biometric authentication and blockchain technology to bolster data protection and prevent unauthorized access, implementation of artificial intelligence algorithms for personalized financial recommendations and fraud detection, and expansion of digital banking services to include innovative features such as peer-to-peer lending, automated budgeting tools, and virtual financial advisors. Additionally, further optimization of user interfaces and mobile banking applications to enhance user experience and accessibility across diverse devices and platforms could be pursued.

SYSTEM REQUIREMENTS

➤ SOFTWARE REQUIREMENTS:

The major software requirements of the project are as follows:

Language : Turbo-C

Operating system: Windows 10 or more.

Technical requirements: Monitor, CPU, Keyboard, Mouse, etc...

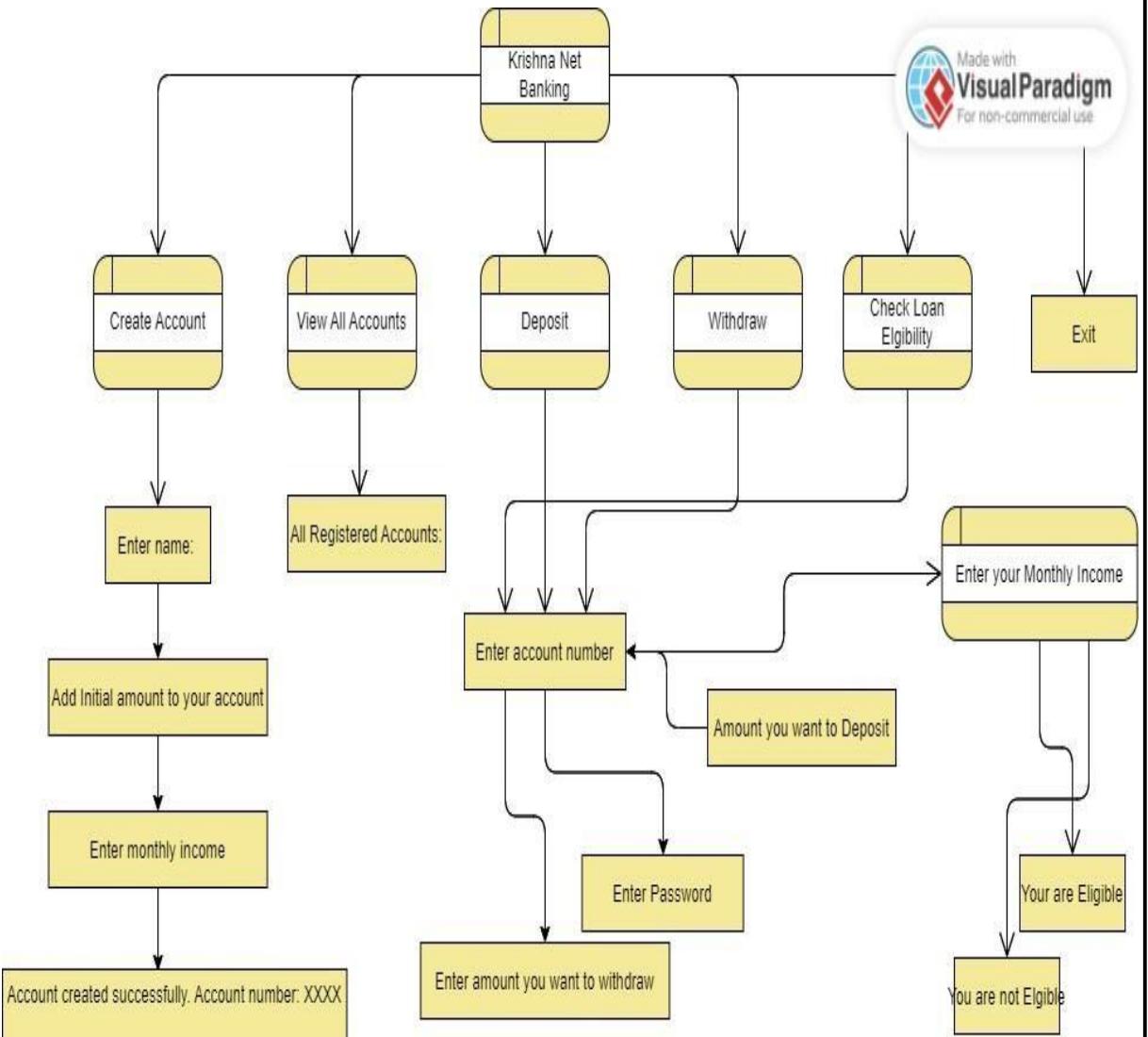
➤ HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are as follows:

RAM : 8 GB RAM

Processor : RYZEN 5

DATA FLOW DIAGRAM



ALGORITHM

- Initialize variables and data structures.
- Set up initial accounts with account numbers, names, and balances.
- Start user interaction loop.
- Display menu options (create account, view accounts, deposit, withdraw, exit).
- Prompt user for choice.
- If choice is create account:
 - Prompt for account details (name, initial balance).
 - Add new account.
- If choice is view accounts:
 - Display list of all accounts.
- If choice is deposit:
 - Prompt for account number and amount.
 - Update account balance with deposit amount.
- If choice is withdraw:
 - Prompt for account number and amount.
 - Update account balance with withdrawal amount.
- If choice is exit:
 - Terminate program.
- Implement account operations (create, view, deposit, withdraw) with error handling.
- Optionally, implement data persistence and security measures.
- Test thoroughly and debug any issues.
- Document algorithm and functionalities.
- Continuously optimize and enhance the system based on user feedback.

IMPLEMENTATION

```
#include <stdio.h>
#include <string.h>

#define MAX_ACCOUNTS 10000
#define MAX_PASSWORD_LENGTH 20

struct BankAccount {
    int accNumber;
    char name[50];
    char password[MAX_PASSWORD_LENGTH];
    float balance;
    float monthlyIncome;
};

struct BankAccount accounts[MAX_ACCOUNTS];
int numAccounts = 0;

void createAccount();
void viewAllAccounts();
void deposit();
void withdraw();
void checkLoanEligibility();

int main() {
    int choice;
```

```
while (1) {  
    printf("\n\tWelcome to Krishna Net Banking\n");  
    printf("1. Create Account\n");  
    printf("2. View All Accounts\n");  
    printf("3. Deposit\n");  
    printf("4. Withdraw\n");  
    printf("5. Check Loan Eligibility\n");  
    printf("6. Exit\n");  
    printf("Enter your choice: ");  
    scanf("%d", &choice);  
  
    switch (choice) {  
        case 1:  
            createAccount();  
            break;  
        case 2:  
            viewAllAccounts();  
            break;  
        case 3:  
            deposit();  
            break;  
        case 4:  
            withdraw();  
            break;  
        case 5:  
            checkLoanEligibility();  
    }  
}
```

```

        break;

    case 6:
        printf("Exiting...\n");
        return 0;

    default:
        printf("Invalid choice. Please try again.\n");
    }

}

return 0;
}

void createAccount() {
    if (numAccounts < MAX_ACCOUNTS) {
        struct BankAccount newAccount;
        newAccount.accNumber = numAccounts + 1;
        printf("Enter name: ");
        scanf("%s", newAccount.name);
        printf("Set password: ");
        scanf("%s", newAccount.password);
        printf("Add Initial amount to your account: ");
        scanf("%f", &newAccount.balance);
        printf("Enter monthly income: ");
        scanf("%f", &newAccount.monthlyIncome); // Store monthly income information
        accounts[numAccounts++] = newAccount;
        printf("Account created successfully. Account number: %04d\n",
               newAccount.accNumber);
    } else {

```

```

printf("Maximum accounts reached. Cannot create more accounts.\n");

}

}

void viewAllAccounts() {

if (numAccounts == 0) {

printf("No accounts found.\n");

} else {

printf("All Registered Accounts:\n");

printf("Acc. No.\tName\tBalance\tMonthly Income\n");

for (int i = 0; i < numAccounts; i++) {

printf("%04d\t%s\t%.2f\t%.2f\n", accounts[i].accNumber, accounts[i].name,
accounts[i].balance, accounts[i].monthlyIncome);

}

}

}

void deposit() {

int accNumber;

float amount;

char password[MAX_PASSWORD_LENGTH];

printf("Enter account number: ");

scanf("%d", &accNumber);

for (int i = 0; i < numAccounts; i++) {

if (accounts[i].accNumber == accNumber) {

printf("Enter password: ");

scanf("%s", password);

if (strcmp(password, accounts[i].password) == 0) {

printf("Enter amount to deposit: ");

scanf("%f", &amount);

```

```

accounts[i].balance += amount;

printf("Deposit successful. Current balance: %.2f\n", accounts[i].balance);

} else {

    printf("Incorrect password. Access denied.\n");

}

return;
}

printf("Account not found.\n");

}

void withdraw() {

    int accNumber;

    float amount;

    char password[MAX_PASSWORD_LENGTH];

    printf("Enter account number: ");

    scanf("%d", &accNumber);

    for (int i = 0; i < numAccounts; i++) {

        if (accounts[i].accNumber == accNumber) {

            printf("Enter password: ");

            scanf("%s", password);

            if (strcmp(password, accounts[i].password) == 0) {

                printf("Enter amount to withdraw: ");

                scanf("%f", &amount);

                if (accounts[i].balance >= amount) {

                    accounts[i].balance -= amount;

                    printf("Withdrawal successful. Current balance: %.2f\n", accounts[i].balance);

                } else {

```

```

        printf("Insufficient balance\n");

    }

} else {

    printf("Incorrect password. Access denied.\n");

}

return;

}

printf("Account not found.\n");

}

void checkLoanEligibility() {

int accNumber;

printf("Enter account number: ");

scanf("%d", &accNumber);

for (int i = 0; i < numAccounts; i++) {

if (accounts[i].accNumber == accNumber) {

    if (accounts[i].monthlyIncome >= 30000) { // Use stored monthly income for
eligibility check

        printf("Congratulations! You are eligible for a loan.\n");

    } else {

        printf("Sorry, you are not eligible for a loan.\n");

    }

return;

}

printf("Account not found.\n");

}

```

INTEGRATION AND SYSTEM TESTING

OUTPUTS

Screen Shots:

```
🎯 Welcome to Krishna Net Banking
1. Create Account
2. View All Accounts
3. Deposit
4. Withdraw
5. Check Loan Eligibility
6. Exit
Enter your choice: 5
Enter account number: 0021
Enter monthly income: 2000
Sorry, you are not eligible for a loan.
```

```
🎯 Welcome to Krishna Net Banking
1. Create Account
2. View All Accounts
3. Deposit
4. Withdraw
5. Check Loan Eligibility
6. Exit
Enter your choice: 4
Enter account number: 0021
Enter password: pass11
Enter amount to withdraw: 102000
Withdrawal successful. Current balance: 0.00
```

🎯 Welcome to Krishna Net Banking

1. Create Account
2. View All Accounts
3. Deposit
4. Withdraw
5. Check Loan Eligibility
6. Exit

Enter your choice: 2

All Registered Accounts:

Acc. No.	Name	Balance	Monthly Income
0001	Customer1	0.00	0.00
0002	Customer2	0.00	0.00
0003	Customer3	0.00	0.00
0004	Customer4	0.00	0.00
0005	Customer5	0.00	0.00
0006	Customer6	0.00	0.00
0007	Customer7	0.00	0.00
0008	Customer8	0.00	0.00
0009	Customer9	0.00	0.00
0010	Customer10	0.00	0.00
0021	Krishna	52000.00	522000.00

🎯 Welcome to Krishna Net Banking

1. Create Account
2. View All Accounts
3. Deposit
4. Withdraw
5. Check Loan Eligibility
6. Exit

Enter your choice: 3

Enter account number: 0021

Enter password: pass11

Enter amount to deposit: 50000

Deposit successful. Current balance: 102000.00

⌚ Welcome to Krishna Net Banking

1. Create Account
2. View All Accounts
3. Deposit
4. Withdraw
5. Check Loan Eligibility
6. Exit

Enter your choice: 1

Enter name: Krishna

Add Initial amount to your account: 52000

Enter monthly income: 522000

Account created successfully. Account number: 0021

⌚ Welcome to Krishna Net Banking

1. Create Account
2. View All Accounts
3. Deposit
4. Withdraw
5. Check Loan Eligibility
6. Exit

Enter your choice: 6

Exiting...

CONCLUSION

The developed C-based banking application represents a significant step forward in digital banking solutions, offering users a reliable platform for managing their finances efficiently. By providing essential features such as account creation, balance inquiries, deposit and withdrawal capabilities, and loan eligibility assessments, the application empowers users to take control of their financial activities with ease.

Looking ahead, there exists vast potential for further enhancements and refinements to enrich the user experience and expand the application's capabilities. These could include the implementation of advanced security measures to ensure the safety of sensitive information, the integration of additional banking services such as bill payments and fund transfers, and the optimization of performance to handle larger user bases and transaction volumes seamlessly.

Moreover, continuous feedback from users and stakeholders will be instrumental in guiding future developments, enabling the application to adapt to evolving user needs and industry trends effectively. By remaining agile and responsive to feedback, the banking application can continue to evolve and thrive in an ever-changing digital landscape, delivering greater value and convenience to users worldwide.