**A**

**Project Report**

**On**

**“Simple Banking System ”**

Submitted to



**Punyashlok Ahilyadevi Holkar Solapur University, Solapur.**

**In the Partial fulfilment of**

**“M.sc(CS)”**

**Submitted By**

**Mr. Gaikwad Amol Sharad.**

**Under the Guidance of**

**Prof. Mr.Salunkhe sir**

**Greenfingers College Of Computer and Technology, Akluj Year 2023-2024**



Greenfingers College of Computer and Technology, Akluj

**Department of Computer Science**



This is to certify that the project entitled **“Simple Banking System**” has been carried out by the following students in partial fulfilment of the degree of **Msc(CS)** at **Greenfingers College of Computer and Technology, Akluj** submitted to **Punyashlok Ahilyadevi Holkar** **Solapur University, Solapur** during the academic year **2023-2024**.

## Mr. Gaikwad Amol Sharad.

## Date: 08/05/2024

## Place: Akluj

Project Guide EXTERNAL EXAMINAR ( H.O.D)

Acknowledgment

Acknowledgment

“ When we start the journey towards something worthwhile it’s never a simple trail nor an easy mile ,but we often move on without back. At all the peoples who helped put us on track , so today when we’ve reached the end of our journey .we’d like to thank of all those who walked with us .”

I sincerely thank to all who’s blessing & good wishes have enabled me to complete the task of **“Simple Banking System**”.

It is a moment of great satisfaction pleasure gratitude for me to give heartily thanks those who help me to complete this project.

I am also helpful to Head, Department of Computer Science am also thankful to **Prof. Salunkhe S.S. Sir**

**Department of Computer science**.

Also, I would like to take this opportunity to thanks our lecturers as wellas guide **Prof. Salunkhe S.S. Sir** for their valuable guidance & suggestions to complete this project work in time.

I must also thank all my friends & well-wishers whose kind helps co-operation encouraged me immensely to complete the project.

Yours Faithfully,

**Mr. Gaikwad Amol Sharad.**

INDEX

**Index Page**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Contents** | **Page No.** |
| 1 | Introduction and Objective | 8 -13 |
| 2 | Feasibility Study | 14 -17 |
| 3 | System requirement specification | 18 – 22 |
| 4 | Need of Project | 23 – 24 |
| 5 | * 1. Scope of Project | 25 – 27 |
| 6 | * 1. Fact Finding Techniques | 28-29 |
| 7 | * 1. SDLC | 30 – 34 |
| 8 | * 1. Introduction to Front-end application | 35 – 40 |
| 9 | * 1. Future Enhancements | 41 - 42 |
| 10 | E-R Diagram | 43 – 46 |
| 11 | Testing and Debugging | 47 – 53 |
| 12 | Home Page  Login  Create new account  Quite | 54 – 55 |
| 13 | After Login  Check balance  Credit Amount in account  Debit Amount from account  View transition history  Log out | 56 - 58 |
| 14 | Input /Output Screenshots | 59 – 76 |
| 15 | Limitations | 77 – 79 |
| 16 | Bibliography | 80 - 81 |

Introduction

# Introduction

During the past several decades personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function There are many factors that have influenced this transformation like technological advances, professionalism and general recognition of human beings as most important resources

A simple banking system in Python is a software application that allows users to perform basic banking transactions such as Login account creation, deposit, withdrawal, balance enquiry, transation history and Log out . The system is designed to provide a user-friendly interface and ensure the security of user data and transactions.

This project intends to introduce more user friendliness in the various activities such as Login account creation, deposit, withdrawal, balance enquiry, transation history and Log out. The Login of Account holder has been made quite simple as all the details of the customer can be obtained by simply keying in the identification of account number of that customer. Similarly, Transition history can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date

The entire information has maintained in the Files and whoever wants to retrieve can't retrieve, only authorization user can retrieve the necessary information which can be accessible from the file.

Objective of project

OBJECTIVE OF THE PROJECT

A simple banking system in Python is a software application that allows users to perform basic banking transactions such as Login account creation, deposit, withdrawal, balance enquiry, transation history and Log out . The system is designed to provide a user-friendly interface and ensure the security of user data and transactions.

This project intends to introduce more user friendliness in the various activities such as Login account creation, deposit, withdrawal, balance enquiry, transation history and Log out. The Login of Account holder has been made quite simple as all the details of the customer can be obtained by simply keying in the identification of account number of that customer. Similarly, Transition history can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The entire information has maintained in the Files and whoever wants to retrieve can't retrieve, only authorization user can retrieve the necessary information which can be accessible from the file

The main objective of our project is providing the different typed of customers facility the main objective of this system is to find out the actual customer service Etc

It should fulfill almost all the process requirements of any Bank.

It should increase the productivity of bank by utilizing the working hours more and more, with minimum manpower.

This project includes the entire upgraded feature required for the computerization banking system. This system is very easy to use, so that any user can use without getting pre-knowledge about this Its very much user friendly and meet almost all daily working process requirements. This system is completely GUI based and can be use by mouse and as well as keyboard. This system is melded in such a way that has got all features to upgrade without making much change in existing components

Feasibility analysis

Feasibility Analysis

Depending on the results of the nunal investigation, the survey is expanded to a more detailed feanbitny study A feasibility study us a test of a system proposal According to its workalnity, impact on the organization, ability to meet user needs and effective of the resources its main task done during the feasibility study are

1. Evaluation of existing system and procedure Our group went to various Banking Professionals to gather information about the software system. They are using and evaluating those system and the procedures invoked in it during the period of feasibility study

2. Analysis of alternative candidate systems after studying the various systems we derived various alternatives through which we develop our project and evaluated the alternatite The most appropriate is selected

Feasibility

study

FEASIBILITY STUDY

The only tangible benefit provided by the proposed system is that the paper work is reduced to the minimum and hence the reduction in cost incurred on Stationary and its stage. The system provides many benefits that can't be measured in terms of Money for user's friendliness, more user response being more efficient.

TECHNICAL FEASIBILITY

The proposed system is technically feasible as it can be developed easily with the help of available technology The proposed system requires IDLE using Python as a Interface for Programming & back-end as .txt file for storing maintaining records.

OPERATIONAL FEASIBILITY

Automation makes our life easy. The proposed system is highly user friendly and is much easily able to interact with the system. Therefore the users will readily accept the system as data entry and making queries can be easily done.

SYSTEm requirements

Hardware

Requirements

Hardware Requirements

Hardware is a set of physical components, which performs the functions of applying appropriate, predefined instructions. In other words, one can say that electronic and mechanical parts of computer constitute hardware.

This package is designed on a powerful programming language Python. It is a powerful Graphical User Interface. The backend is Notepad, which is used to maintain database. It can run on almost all the popular microcomputers. The following are the minimum hardware specifications to run this package:

Personal Computer: -

It minimum contains i3 Processor with 2 GB RAM or More .

software

Requirements

Software Requirments

The software is a set of procedures of coded information or a program which when fed into the computer hardware, enables the computer to perform the various tasks. Software is like a current inside the wire, which cannot be seen but its effect can be felt.

1. Operating System:- Windows 10 or more

2. Application Software:-

Application software uses front end Python 3.11.1 and Notepad etc.

Editor:- Pycharm / Visual Studio Code.

Need of

Project

Need of Project

A simple banking system in Python can be useful for individuals or small businesses who want to manage their finances without having to rely on a traditional bank. It can also be a great learning tool for students who want to gain practical experience in programming and financial management. With a simple banking system in Python, users can easily track their transactions, monitor their account balances, and perform basic banking functions from the comfort of their own computer. Additionally, by using Python, the program can be easily customized and expanded to include additional features as needed.

scope of

Project

Scope of Project

A simple banking system in Python can be useful for individuals or small businesses who want to manage their finances without having to rely on a traditional bank. It can also be a great learning tool for students who want to gain practical experience in programming and financial management. With a simple banking system in Python, users can easily track their transactions, monitor their account balances, and perform basic banking functions from the comfort of their own computer. Additionally, by using Python, the program can be easily customized and expanded to include additional features as needed.

The scope of a simple banking system in Python can include various functionalities such as:

1. **Account creation and management**: The system should allow users to create and manage their accounts by providing basic information such as name, address, email, and phone number.

2. **Credit and Debit:** The system should allow users to deposit and withdraw money from their accounts.

3. **Balance Enquiry:** Users should be able to check their account balance at any time.

4. **Transaction History:** The system should maintain a record of all the transactions made by the user, including deposits, withdrawals, and transfers.

5. **Security**: The system should ensure the security of user data and transactions by implementing appropriate security measures such as encryption and authentication.

6. **User Interface**: The system should have a user-friendly interface that allows users to easily navigate through the different functionalities.

Overall, the scope of a simple banking system in Python would be to provide basic banking services to users while ensuring their security and convenience

Fact finding techniques

**Fact Finding Techniquies**

1. Interviews: Conducting interviews with potential users of the banking system can help identify their needs and preferences.

2. Surveys: Surveys can be used to gather feedback on potential features and functionality of the banking system.

3. Focus groups: Focus groups can be used to gather feedback from a group of potential users on their experience with the banking system.

4. Industry reports and publications: Reviewing industry reports and publications on banking systems and technology can provide insights into current market trends and demands for banking services.

5. Case studies: Analyzing case studies of successful banking systems and their implementation processes can provide valuable insights into the development of a simple banking system.

6. Prototyping and testing: Using prototyping and testing can help gather feedback and refine the design of the banking system.

7. User personas and scenarios: Creating user personas and scenarios can help understand how users will interact with the system and identify areas for improvement.

Software development life cycle

SOFTWARE LIFE DEVELOPMENT CYCLE

A system development life cycle is a logical process by which system analysts, software engineers, programmers, and end users build information systems and computer applications to solve business problems and needs.

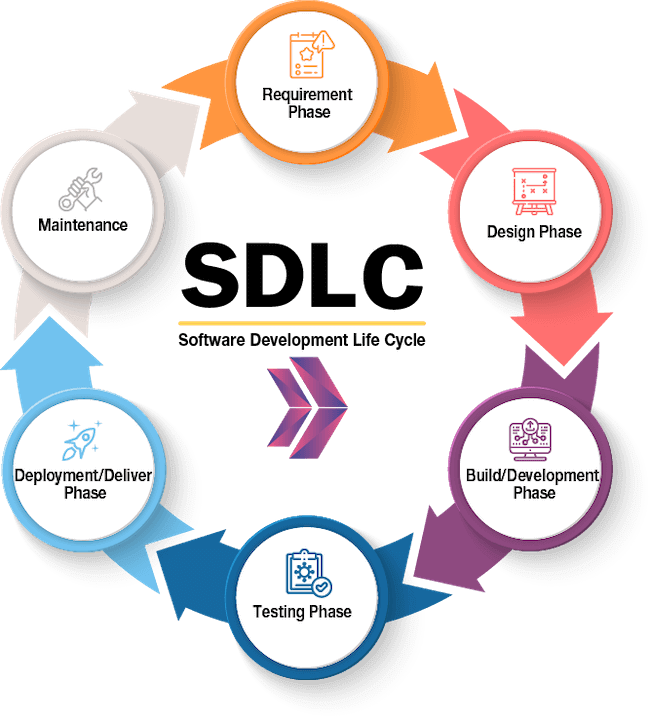
The major phases involved in the MIS development process are referred to as system development life cycle. Each phase of the development process must have well defined

objectives and at the end of each phase progress towards meeting the objectives must be evaluated.

The development process should not continue until the objectives of all prior phases have been met.

System development life cycle is a phased approach to analysis and design to ensure that systems are best developed.

The system development life cycle can be divided into seven phases as shown in fig.



The software development life cycle (SDLC) is a process used by software developers to design, develop, test, and deploy software. The SDLC consists of several phases that are typically followed in a sequential order:

1. **Planning:** In this phase, the project goals and requirements are defined, and a plan is created for how the project will be executed.

2. **Analysis**: In this phase, the requirements are analyzed in detail to determine the scope of the project and identify any potential issues or risks.

3. **Design:** In this phase, the software architecture is designed, including the user interface, database schema, and other technical details.

4. **Implementation:** In this phase, the actual coding of the software is done, based on the design created in the previous phase.

5. **Testing:** In this phase, the software is tested to ensure that it meets the requirements and functions correctly.

6. **Deployment**: In this phase, the software is released to users, either through a public release or an internal deployment.

7. **Maintenance:** In this phase, the software is monitored and updated as needed to fix any bugs or add new features.

Following the SDLC helps ensure that software is developed in a structured and organized manner, with each phase building on the work done in the previous phase. This can help reduce errors and improve the overall quality of the software.

Introduction to

front-end

tool

Introduction to Front-end tool

Python is a high-level, interpreted programming language that is widely used for various applications, including web development, data analysis, and artificial intelligence. While Python is typically associated with back-end development, it can also be used as a front-end language to create dynamic and engaging user interfaces. In this report, we will explore the benefits of using Python for front-end development, discuss popular frameworks and libraries, provide practical examples, and highlight potential limitations. By the end of this report, readers will have a better understanding of how Python can be used to create compelling user experiences.

Need of Python Programming

Python programming is in high demand due to its versatility and ease of use. Python is a popular language for web development, data analysis, machine learning, artificial intelligence, and scientific computing.

Its simple syntax and large community of developers make it an accessible language for beginners and experts a like. Python's popularity in web development is due to its ability to create dynamic and engaging user interfaces.

Python frameworks such as Django and Flask allow developers to create robust web applications quickly and efficiently. Additionally, Python's ability to integrate with other languages and technologies makes it a valuable tool in creating complex web applications.

In data analysis and machine learning, Python has become the language of choice due to its extensive libraries and tools such as NumPy, Pandas, and Scikit-learn. These libraries allow developers to analyze large datasets, build predictive models, and create visualizations. Python's versatility extends beyond web development and data analysis.

It can be used for scientific computing, game development, desktop applications, and more. Its flexibility and ease of use make it a valuable tool for developers across industries.

In conclusion, the need for Python programming is evident in its versatility, ease of use, and extensive community of developers. As technology continues to evolve, Python will remain a valuable tool for creating compelling user experiences.

Advantages of Python Programming

#### 1. Easy to Read, Learn and Write

Python is a**high-level programming language** that has English-like syntax. This makes it easier to read and understand the code.

Python is really easy to**pick up**and **learn**, that is why a lot of people recommend Python to beginners. You need less lines of code to perform the same task as compared to other major languages like **C/C++** and **Java**.

#### 2. Improved Productivity

Python is a very **productive language**. Due to the simplicity of Python, developers can focus on solving the problem. They don’t need to spend too much time in understanding the **syntax** or **behavior** of the programming language. You write less code and get more things done.

#### 3. Interpreted Language

Python is an interpreted language which means that Python directly**executes the code** line by line. In case of any error, it stops further execution and reports back the error which has occurred.

Python shows only one error even if the program has multiple errors. This makes **debugging** easier.

#### 4. Dynamically Typed

Python doesn’t know the type of variable until we run the code. It automatically assigns the data type during **execution**. The programmer doesn’t need to worry about declaring variables and their data types.

#### 5. Free and Open-Source

Python comes under the **OSI approved** open-source license. This makes it **free**to **use** and **distribute**. You can download the source code, modify it and even distribute your version of Python. This is useful for organizations that want to modify some specific behavior and use their version for development.

#### 6. Vast Libraries Support

The standard library of Python is huge, you can find almost all the functions needed for your task. So, you don’t have to depend on external libraries.

But even if you do, a **Python package manager (pip)**makes things easier to import other great packages from the **Python package index (PyPi)**. It consists of over 200,000 packages.

#### 7. Portability

In many languages like C/C++, you need to change your **code** to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere.

Future enhancement

Future Enhancement

The scope of a simple banking system project in Python would include creating a program that allows users to perform basic banking functions such as creating accounts, depositing and withdrawing money, and checking account balances. The program would also need to have a secure login system to protect user information and transactions.

Additional features that could be included in the project may include:

1. Transfer funds between accounts

2. View transaction history

3. Generate account statements

4. Set up automatic payments or bill pay

5. Send and receive money through online banking services.

The project could be expanded to include more advanced features such as fraud detection, loan management, and investment tracking. However, the scope of the project would depend on the level of complexity desired and the intended audience for the program.

E – r diagram

E - R Diagram

ER-modeling is a data modeling technique used in software engineering to produce a conceptual data model of a information system. Diagrams created using this ER- modeling technique are called Entity-Relationship Diagrams, or ER diagrams or ERDS. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases.

Dr. Peter Chen is the originator of the Entity-Relationship Model His original paper about ER-modeling is one of the most cited papers in the computer software field Currently the ER model serves as the foundation of many system analyses and design methodologies, computer-aided software engineering (CASE) tools, and repository systems

The onginal notation for ER-Diagrams uses rectangles to represent entities, and diamonds to represent relationships

There are three basic elements in ER-Diagrams

Entities are the "things" for which we want to store information. An entity is a person, place, thing or event

Attributes are the data we want to collect for an entity

Relationships describe the relations between the entities

ERDS show entities in a database and relationships between tables within that database It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works

Entity (Instance)

An instance of a physical object in the real world

Entity Class

Group of objects of the same type

Eg. Entity Class Student". Entities "John". "Trish" etc

Attributes

Properties of Entities that describe their characteristics.

Types:

Simple

Attribute that is not discible, gags

Composite

Auribute composed of several simple attributes. e.g.address (house number, street, dotricte

Multiple

Attribute with a set of possible values for the same

entity, eg. Phone (home, mobile etc y or email

Key

Uniquely Ids the Entity eg PPSN, Chassis No.

Each simple attribute associated with a VS that may be assigned to that attribute for each individual entity.

e.g. age=integer

testing & debugging

Testing and Debugging

INTRODUCTION:-

The implementation phase of software development is concerned with translating design specification into source code. The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. Thus goal can be achieved by making the source code as clear and straightforward as possible Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness and complexity are indications of inadequate design and misdirected thinking,

Source code clarity is enhanced by structured coding techniques, by good coding style. by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages

The implementation team should be provided with a well-defined set of software requirement, an architectural design specification, and a detailed design description. Each team member must understand the objectives of implementation

TERMS IN TESTING FUNDAMENTAL

**1. Error**

The term error is used in two ways It refers to the difference between the actual output of software and the correct output in this interpretation, error is essential a measure of the difference between actual and ideal Error is also to used to refer to human action that result in software containing a defect or fault

**2. Fault**

Fault a condition that causes to fail in performing its required function. A fault is a basic reason for software malfunction and is synonymous with the commonly used term Bug

**3. Failure**

Failure is the inability of a system or component to perform a required function according to its specifications A software failure occurs if the behavior of the software is the different from the specified behavior Failure may be caused due to functional or performance reason

**a. Unit Testing**

The term unit testing comprises the sets of tests performed by an individual programmer prior to integration of the unit into a larger system.

A program unit is usually small enough that the programmer who developed it can test it in great detail, and certainly in greater detail than will be possible when the unit is integrated into an evolving software product. In the unit testing the programs are tested.

separately, independent of each other. Since the check is done at the programs level, it is also called program teasing

**b. Module Testing**

A module and encapsulates related component. So can be tested without other system module

**c. Subsystem Testing**

Subsystem testing may be independently design and implemented common problems are sub-system interface mistake in this checking we concentrate on it

There are four categories of tests that a programmer will typically perform on a program unit.

1) Functional test

2) Performance test

3) Stress test

4) Structure test

**1) Functional Test**

Functional test cases involve exercising the code with Nominal input values for which expected results are known, as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values

**2) Performance Test**

Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device unlization by the program unit. A certain amount of avoid expending too much effort un fine-tuning of a program unit that contributes little to the overall performance of the entire system. Performance testing is most productive at the subsystem and system levels

**3) Stress Test**

Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks

4) **Structure Test**

Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths Some authors refer collectively to functional performance and stress testing as "black box" testing While structure testing is referred to as "white box" or "glass box" testing. The major activities in structural testing are deciding which path to exercise, deriving test date to exercise those paths, determining the test coverage criterion to be used, executing the test, and measuring the test coverage achieved when the test cases are exercised.

DEBUGGING

Defect testing is intended to find areas where the program does not confirm to its specifications. are designed to reveal the presence of defect in the system. When defect have been found in the program. There must be discovered and removed. This is called "Debugging

Home

page

**Home Page :**

The Home Page of “Simple Banking System “ Application consists of following :

**Login:**

The user can Login to his account by entering valid name,account number and password which is given at the creation of account .

If the name ,account number and password are valid then login is successfull. And user goes to next step.

If the entered information is not valid then its show error or a dialogue box that shows message like “Invalid credentials ,Try Again”

**Create new account :**

The new user wants to create new bank account so, he has to enter the information like name ,opening credit and password.

After clicking on submit button it will gives a new unique bank account number which can be used to access your bank account .

**Quit :**

The whole process is completed then if you want close the application you use Quit . on clicking on quit it will close the application.

after

login

**After Login :**

After filling the valid information you are comes to main working page of the “Simple Banking System”

It will the display the name of account holder to be logged in .

Which can be perform following :

**Check Balance:**

Which is shows account holders current bank balance in ruppes.

**Credit amount in your amount :**

If account holder want to deposit the money in his account then he/she can easily credit it .

After credit you can check account balance is updated as credited.

And also shows a message like “Amount Credited Successfully “

**Debit amount from account :**

If account holder want to debit money from his account then he / she can easily do it .

After debit you can check account balance is updated as debited .

And also shows a message like “Amount Debited Successfully ”

**View Transition History:**

The view Transition History consists of the date and time of credited or debited amount also displays the amount to be credited or debited from the account with current balance .

**Logout :**

The overall process is complete then you can logout .

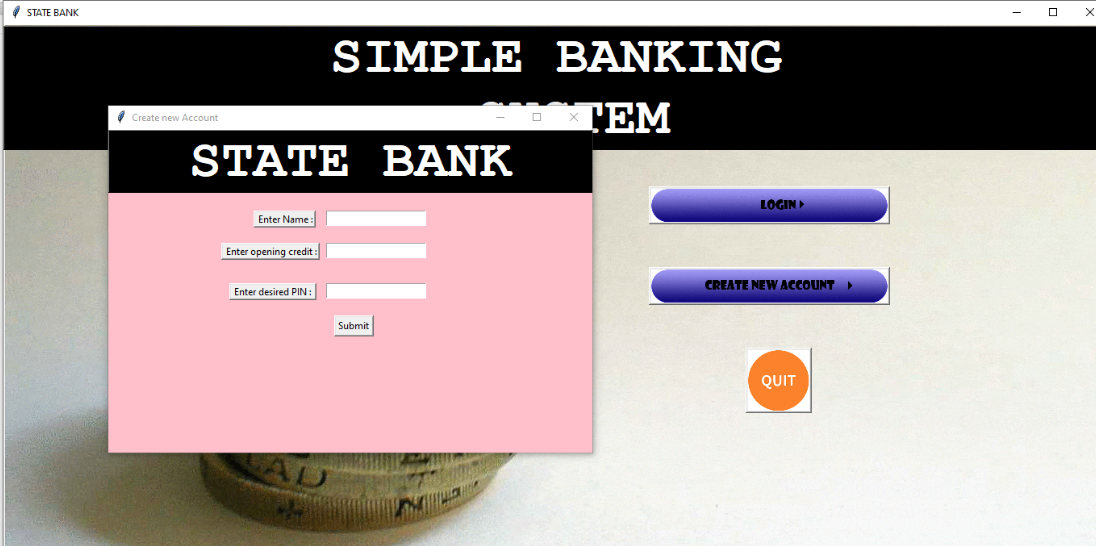
After the logout it goes to home page .

screenshots

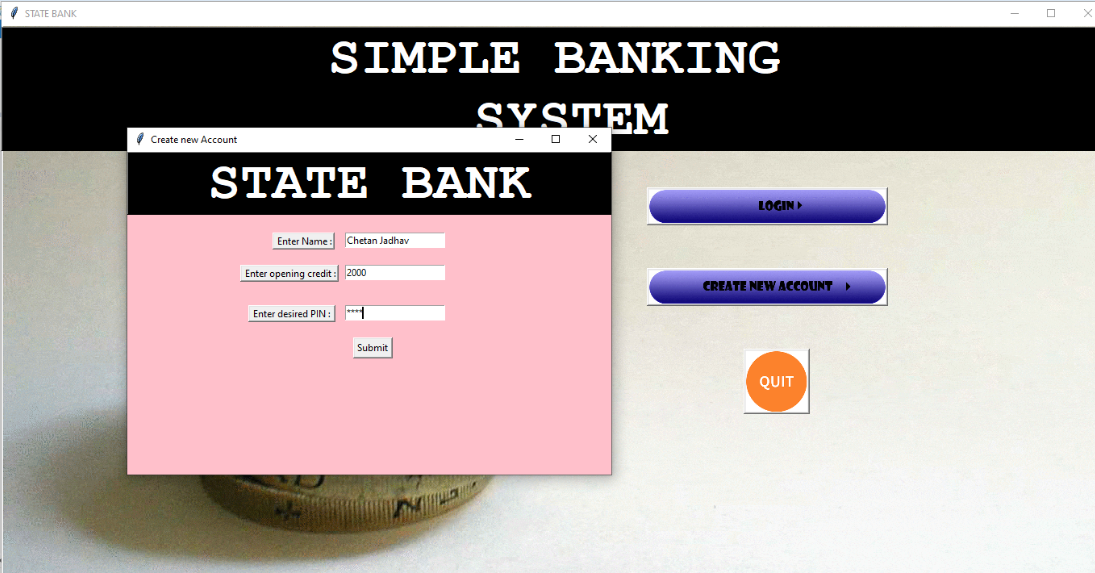
**Home Page :**

****

**Create New Account :**

****

**Entering the valid information:**

****

**Account Number :**

****

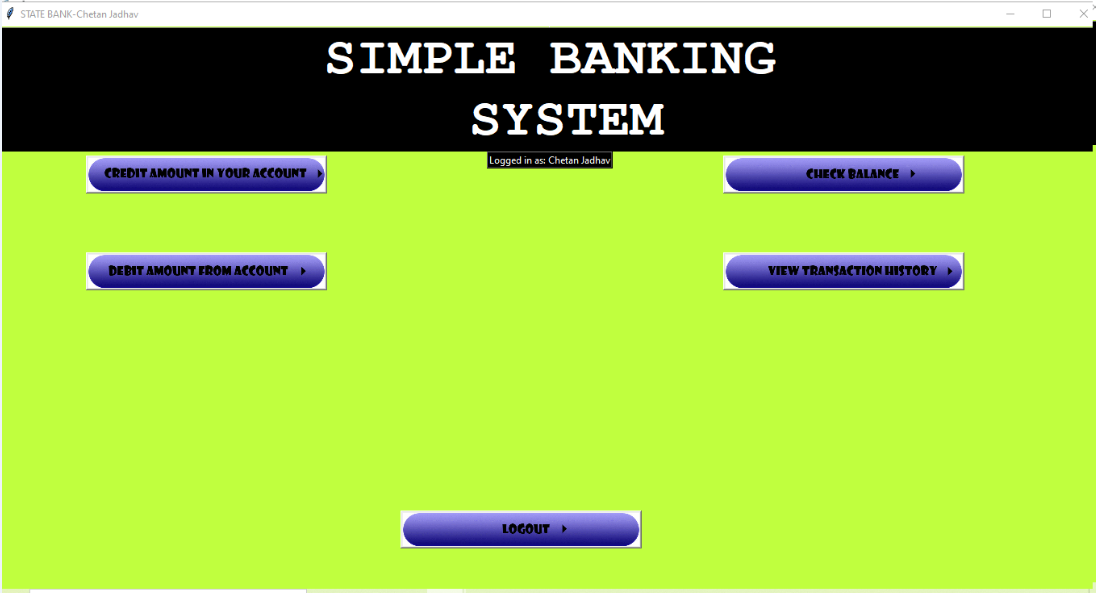
**Login page :**

****

**Entering valid Information :**

****

**After Login Main Menu :**



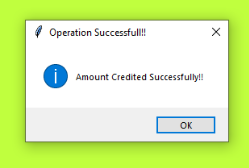
**Credit Amount in your Account :**



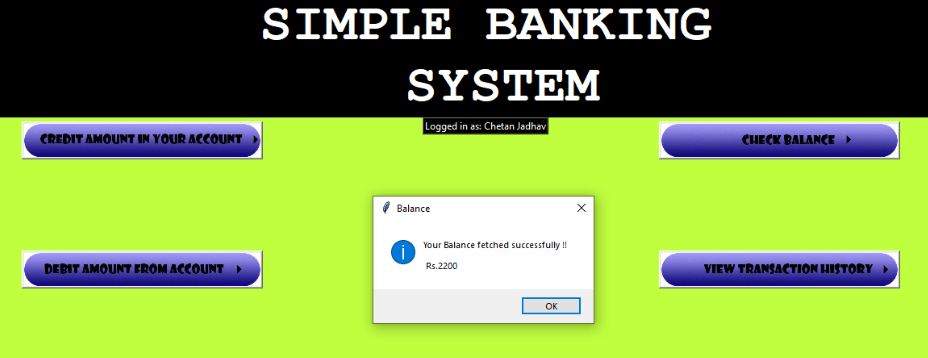
**Entering Amount:**



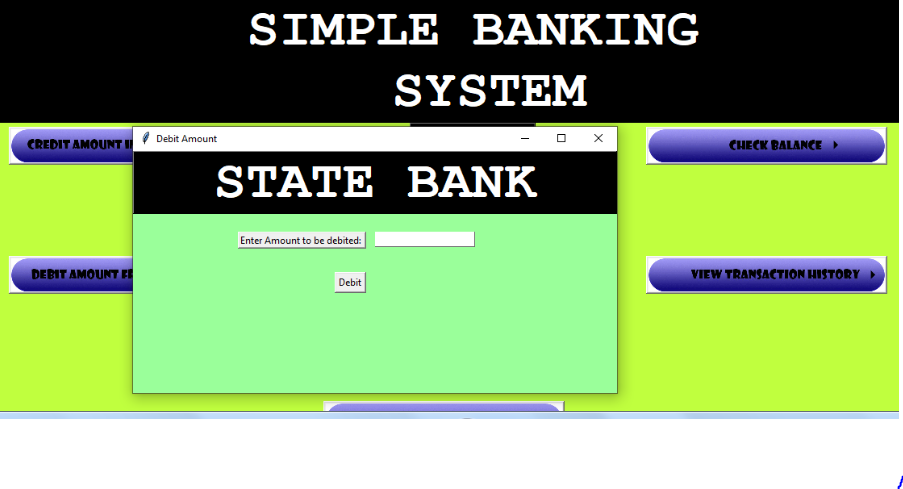
**Amount credited :**



**Check Balance after Credit :**



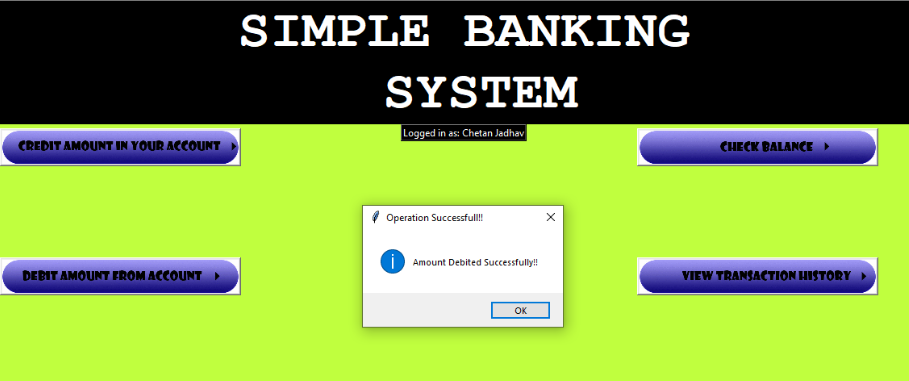
**Debit Amount From your account :**



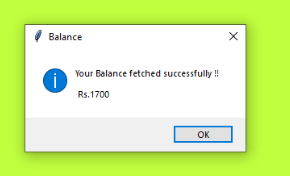
**Entering the amount to be debited:**



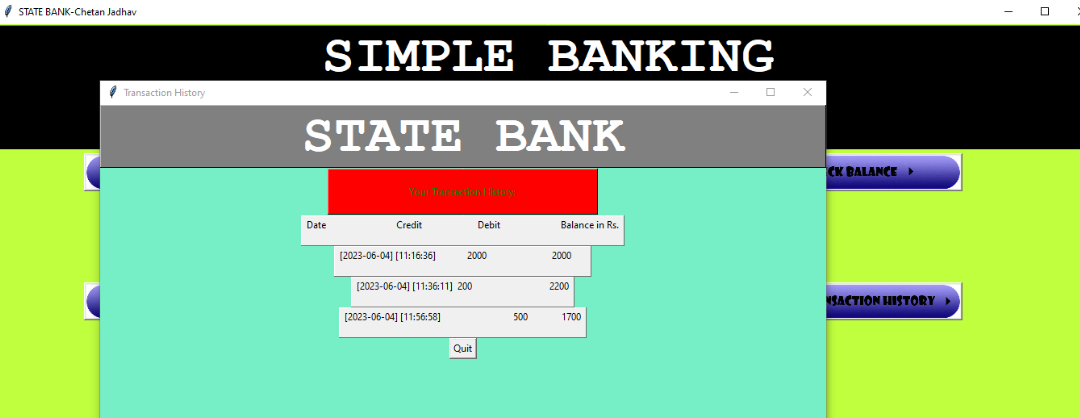
**After Amount Debited :**



**Check Balance After Amount Debited:**



**View Transition History:**



**Log Out :**



Future Scope of project

Future Scope Of Project

The future scope of a simple banking system could include:

**1. Integration with new technologies:** As technology continues to evolve, simple banking systems may need to integrate with new technologies such as blockchain, artificial intelligence, and machine learning to stay competitive.

**2. Enhanced security features**: With the increasing threat of cyber attacks, banking systems will need to continue to enhance their security features to protect customer data and prevent fraud.

**3. Mobile banking:** With the rise of mobile devices, banking systems will need to focus on providing a seamless mobile banking experience for customers.

**4. Personalization:** Simple banking systems may also focus on providing a more personalized experience for customers, tailoring products and services to their specific needs and preferences.

**5. Open banking:** The concept of open banking, where banks share customer data with third-party providers, is gaining traction in many countries. Simple banking systems may need to adapt to this trend to remain competitive.

Overall, the future scope of a simple banking system will likely involve a continued focus on innovation and customer-centricity, as well as a commitment to staying up-to-date with the latest technologies and trends in the industry

bibliography

Bibliography and Reference

BOOK:

### Head First Python – A Brain-Friendly Guide (2nd Edition)

WEBSITES-

https://stackify.com/learn-python-tutorials/

www.PYTHONTUTORIALS.COM

www.geeksforgeeks.com

SEARCH ENGINES –

YAHOO, GOOGLE etc.