CREDIT CARD MANAGEMENT

A COURSE PROJECT REPORT

By

Aryan Singhal (RA2111003010970)

Under the guidance of **Dr. Karthikeyan M**

In partial fulfillment for the Course

18CSC303J-Database Management Systems

In

School of Computing

FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OFSCIENCE

AND TECHNOLOGY

Kattankulathur , Chengalpattu District

APRIL 2024.

Acknowledgement

We would like to express our gratitude to our Professor, **Dr. Karthikeyan M** who gave us the golden opportunity to do this wonderful project on the topic **"Credit Card Management System"** which also helped us in doing a lot of research and we came to know about so many new things we are
really thankful to him.

We are also thankful to all the other faculty, teaching and non-teaching staff members of our department for their kind co-operation and help.

Lastly, we would also like to thank our friends who helped us a lot in finishing this project within the limited time. We are making this project not only for marks but to also increase our knowledge.

Index

Introduction

Our Credit Card Management System is a Python-based application developed using Tkinter for the graphical user interface and SQL for database management. This system aims to provide users with a convenient platform for managing their credit card information, transactions, and applications.

Advantages of SQL

- Data Integrity:Ensures accuracy and consistency of stored data.
- Data Manipulation: Powerful tools for efficient data management.
- Scalability: Capable of handling large volumes of data and users.
- Transaction Support: Ensures data consistency by grouping operations.
- Security: Robust features for protecting sensitive data from unauthorized access.

Advantages of Tkinter

- Cross-Platform Compatibility: Provides a consistent interface across different platforms.
- Easy to Learn and Use: Simple API for quick development of GUI applications.
- Extensive Widget Library: Rich set of built-in widgets for creating interactive interfaces.
- Integration with Python: Seamless integration with Python for implementing complex logic.
- Active Community and Documentation: Large community and resources for learning and support.

Implementation in the Project

- SQL: Used for creating and interacting with the database.
- Tkinter: Used for designing and implementing the graphical user interface.
- Data Management: SQL ensures efficient data management and manipulation.
- User Interface: Tkinter provides a user-friendly interface for seamless interaction.

Project Features and Objectives

Functionality

- Welcome Page: Users are greeted with a welcome message and options to log in or create a new account.
- Login: Users can log in using their username and password.
- Create New Account: Users can create a new account if they don't have one already.
- Main Menu: Upon successful login, users can access the main application window, which provides access to various functionalities such as managing credit cards, setting limits, applying for new credit cards, etc.
- Additional Windows: Depending on the selected functionality from the main menu, additional windows may open for specific tasks such as creating a new credit card, setting limits, etc.

Application Features

- Insert Data: Add new records to the database.
- Update Data: Modify existing records in the database.
- Delete Data: Remove records from the database.
- Retrieve Data: Fetch specific records from the database.
- View All Data: Display all records stored in the database.

Objectives

- To provide users with a seamless experience for managing their credit cards and financial transactions.
- To ensure data integrity by allowing users to insert, update, delete, and retrieve records from the database.
- To enhance user convenience by offering features like viewing all data in the database and navigating through different windows for specific tasks.
- To create a user-friendly interface that simplifies complex operations such as creating new accounts, logging in, and accessing main application functionalities.

Back End Design , Front End Design and Connectivity

Back End Design:

The back-end design of the project involves the implementation of the database schema and the logic to interact with the database. In this project, SQLite is used as the database management system. The back-end design includes:

1. Database Schema: Defines the structure of the database tables and the relationships between them. In this project, tables such as USER1, CREDIT_CARD, LIMITS, APPLICATIONS, etc., are defined with appropriate columns and constraints.

2. SQL Queries: The back end handles SQL queries to perform CRUD operations (Create, Read, Update, Delete) on the database. For example, INSERT, SELECT, UPDATE, DELETE queries are used to manage user accounts, credit card information, limits, applications, etc.

Front End Design:

The front-end design focuses on the graphical user interface (GUI) that users interact with. In this project, Tkinter, a Python library for creating GUI applications, is used. The front-end design includes:

1. Window Layout: Defines the layout and appearance of windows and widgets such as labels, buttons, entry fields, etc. Different windows are created for tasks like login, creating a new account, main menu, etc.

2. User Input Handling: Captures user input from widgets and validates it before sending requests to the back end. For example, when a user enters their login credentials, the front end validates the input before sending a request to the back end for authentication.

Connectivity:

Connectivity refers to how the front end and back end communicate and interact with each other. In this project:

1. Front-End Actions Trigger Back-End Processes: User actions such as clicking a button or entering data trigger corresponding processes in the back end. For example, when a user clicks the "Create Account" button, it triggers the back end to insert the new account information into the database.

2. Back-End Responses Update Front-End Display: After processing requests, the back end sends responses to update the front-end display. For example, if a user's login credentials are valid, the back end sends a response to the front end to open the main menu window.

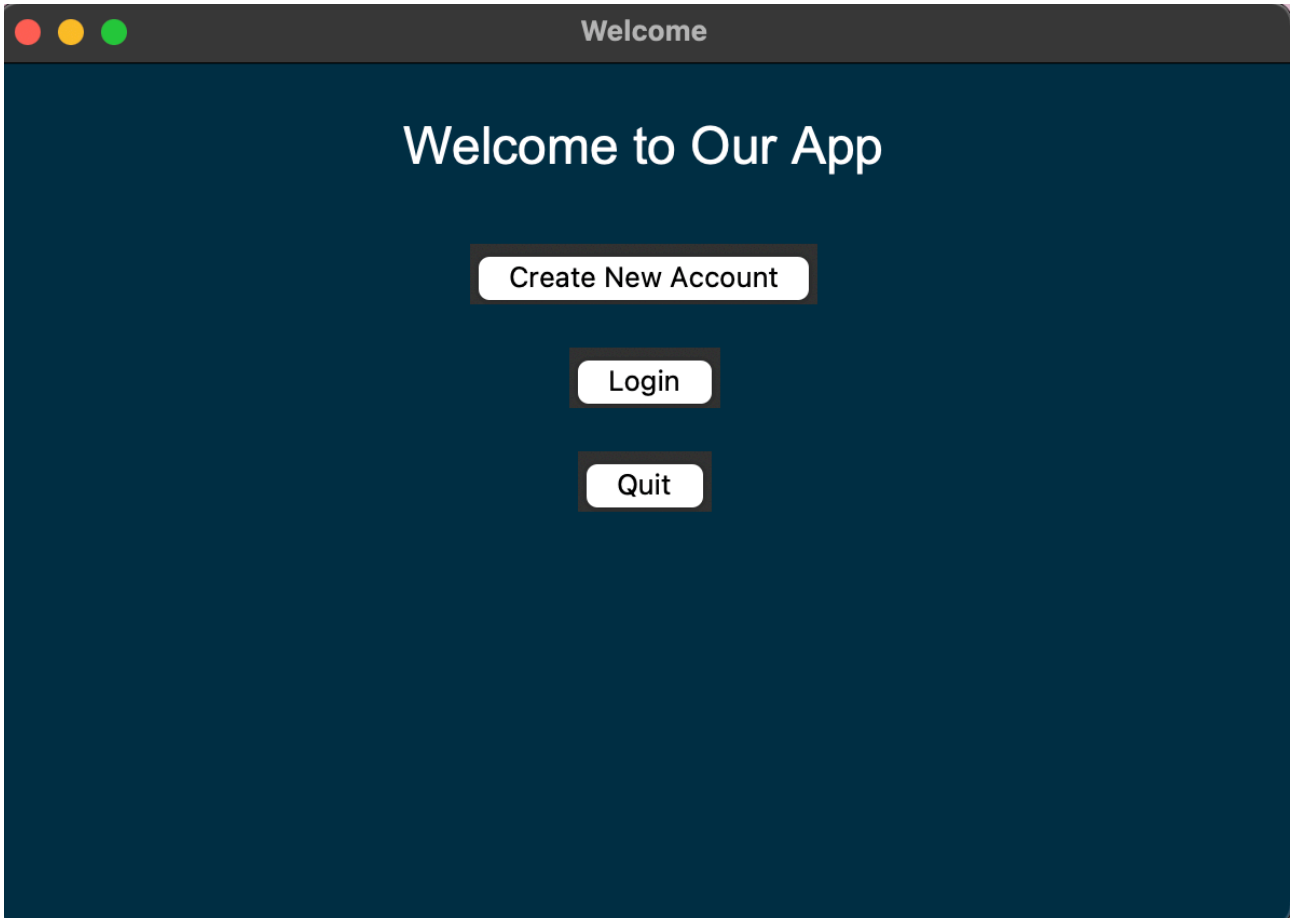
Overall, the project achieves seamless integration between the front end and back end to provide users with a smooth and intuitive experience for managing their credit cards and financial transactions.

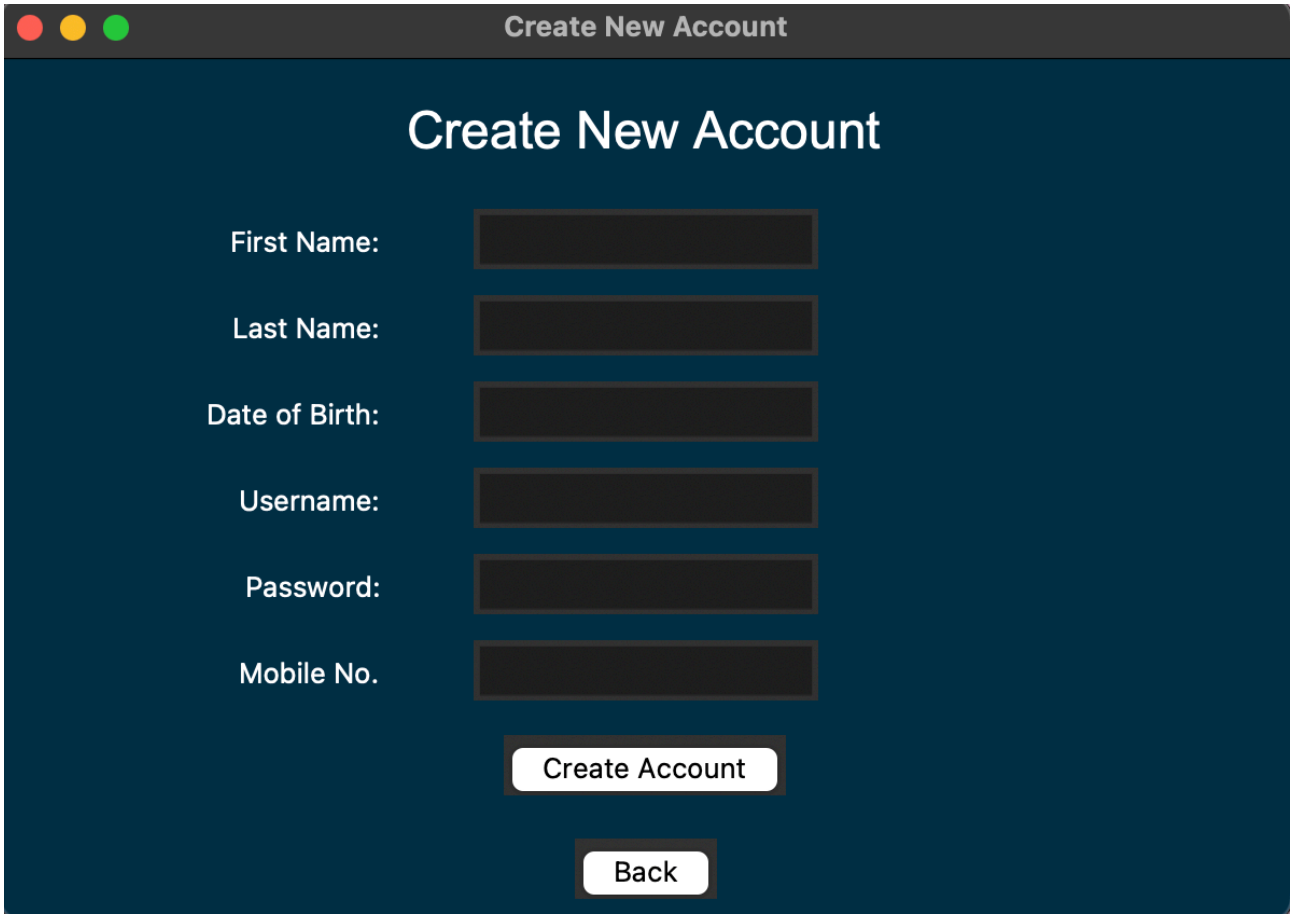
Output

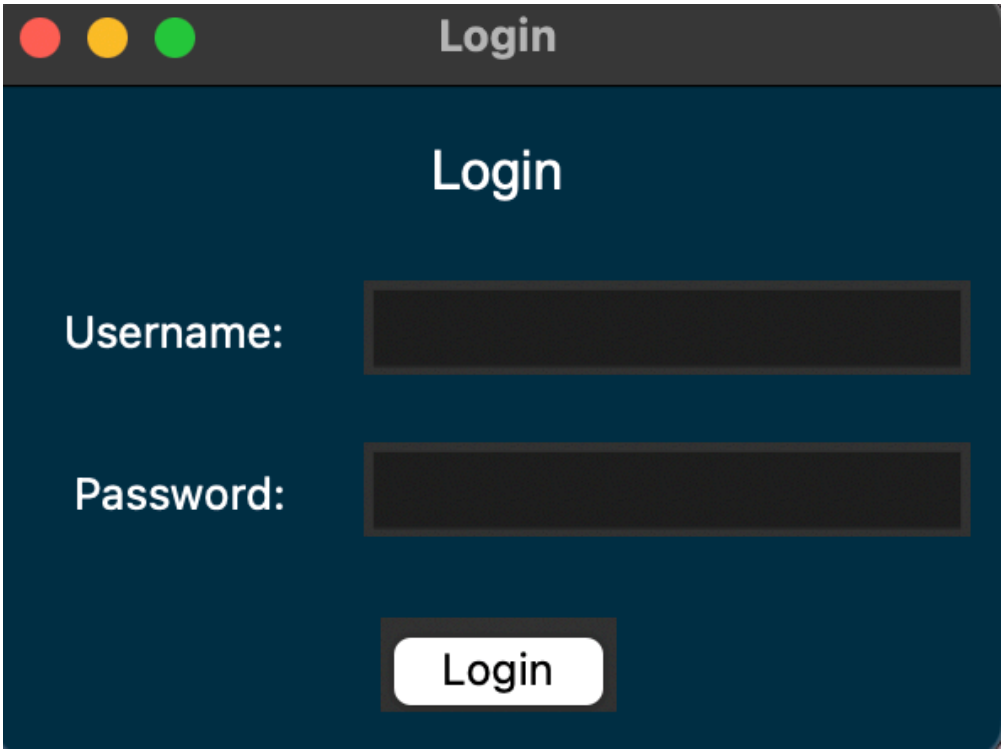
The output of the project is a user-friendly credit card management system with the following features:

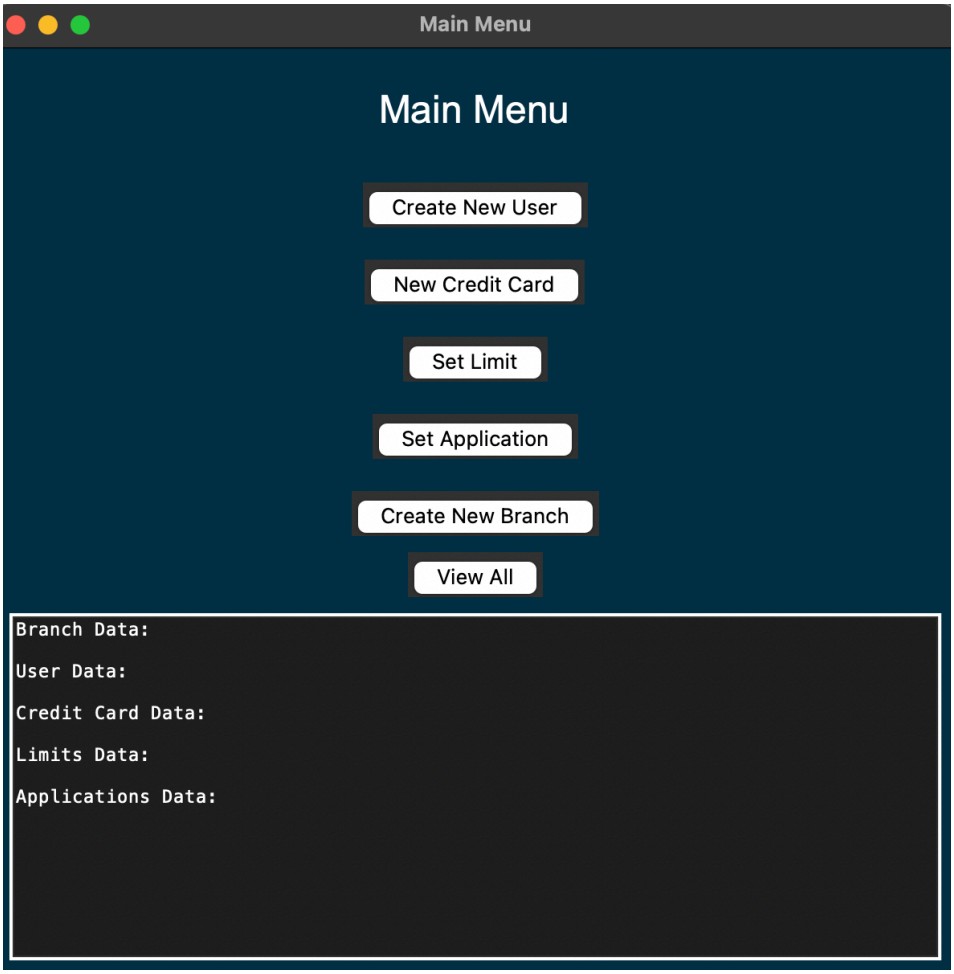
1. Welcome Page: Users are greeted and can log in or create a new account.
2. Login Page: Users log in with their credentials.
3. Create New Account: New users can create an account.
4. Main Menu: Provides access to various functionalities.
5. Additional Windows: Open for specific tasks.
6. Database Interactions: CRUD operations on user data.
7. Data Display: View all stored information.

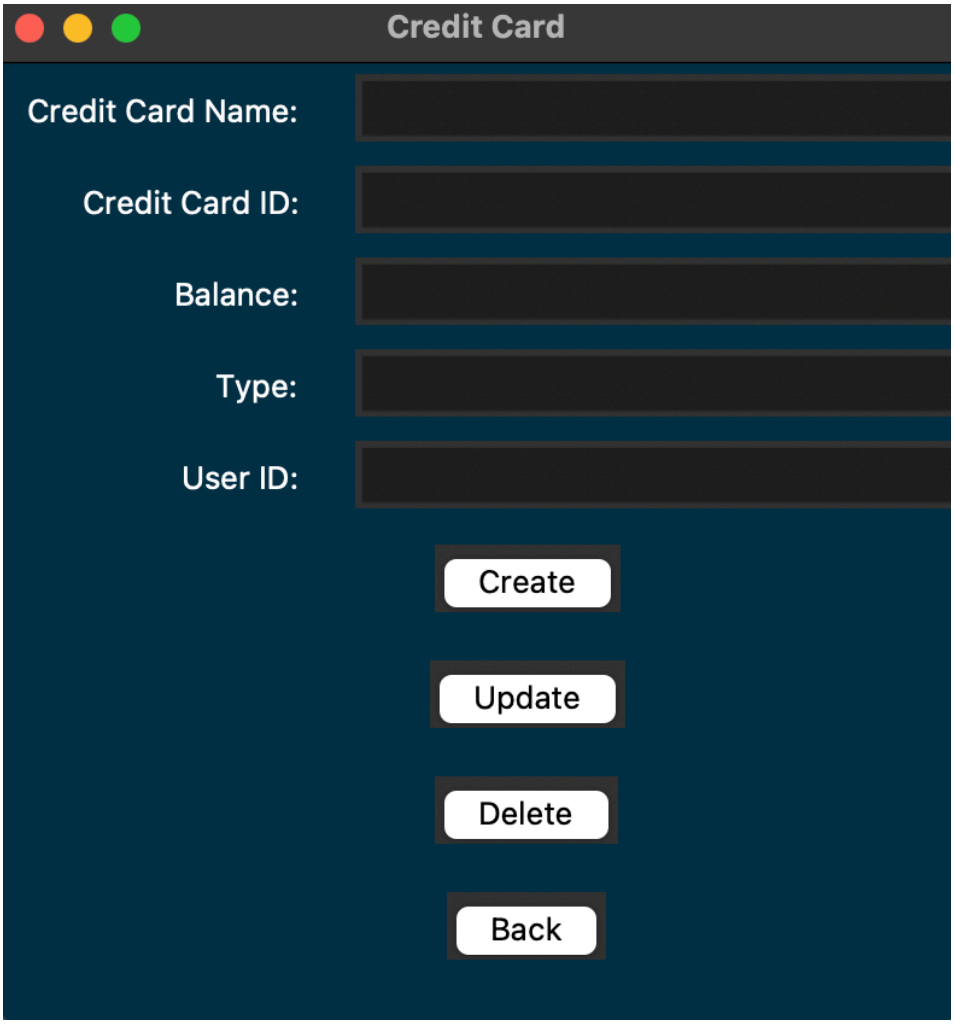
Overall, the output is a streamlined system for efficient credit card management.

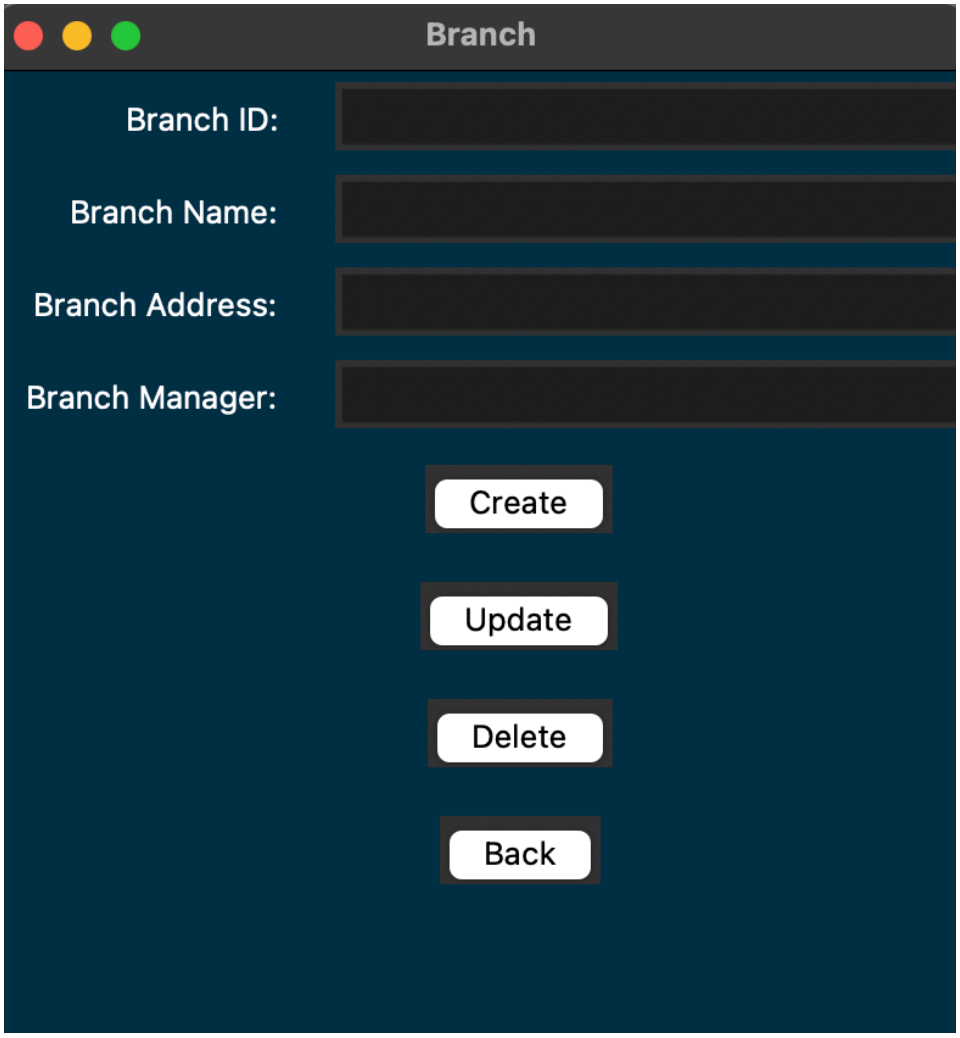


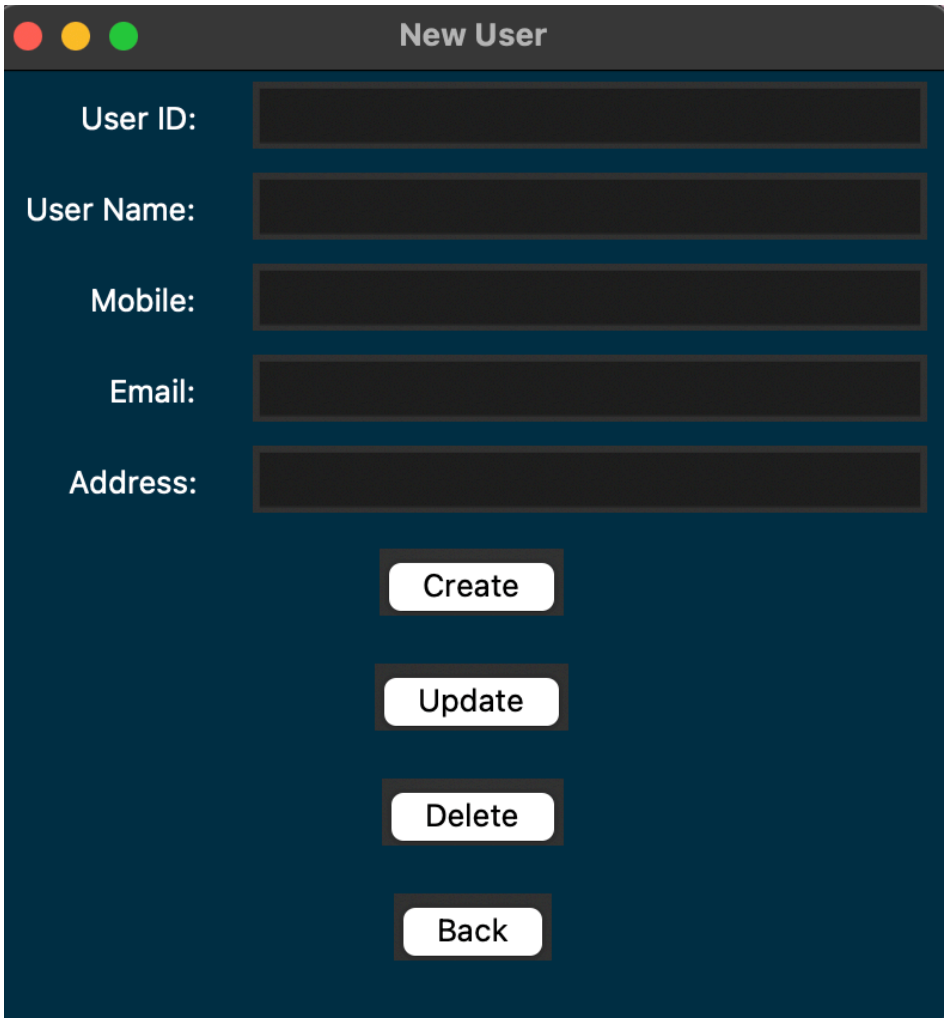

Create New Account
Create New Account
First Name:
Last Name:
Date of Birth:
Username:
Password:
Mobile No.
Create Account
Back

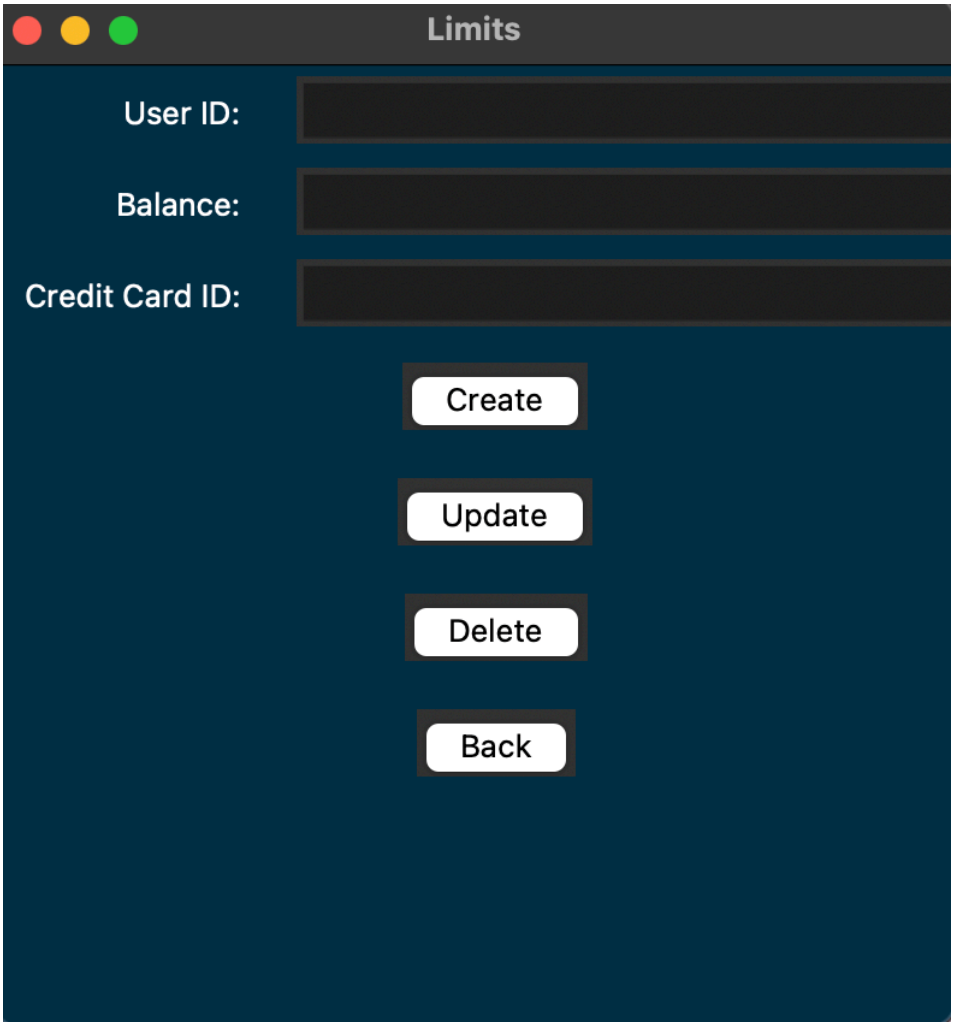

Login
Login
Username:
Password:
Login

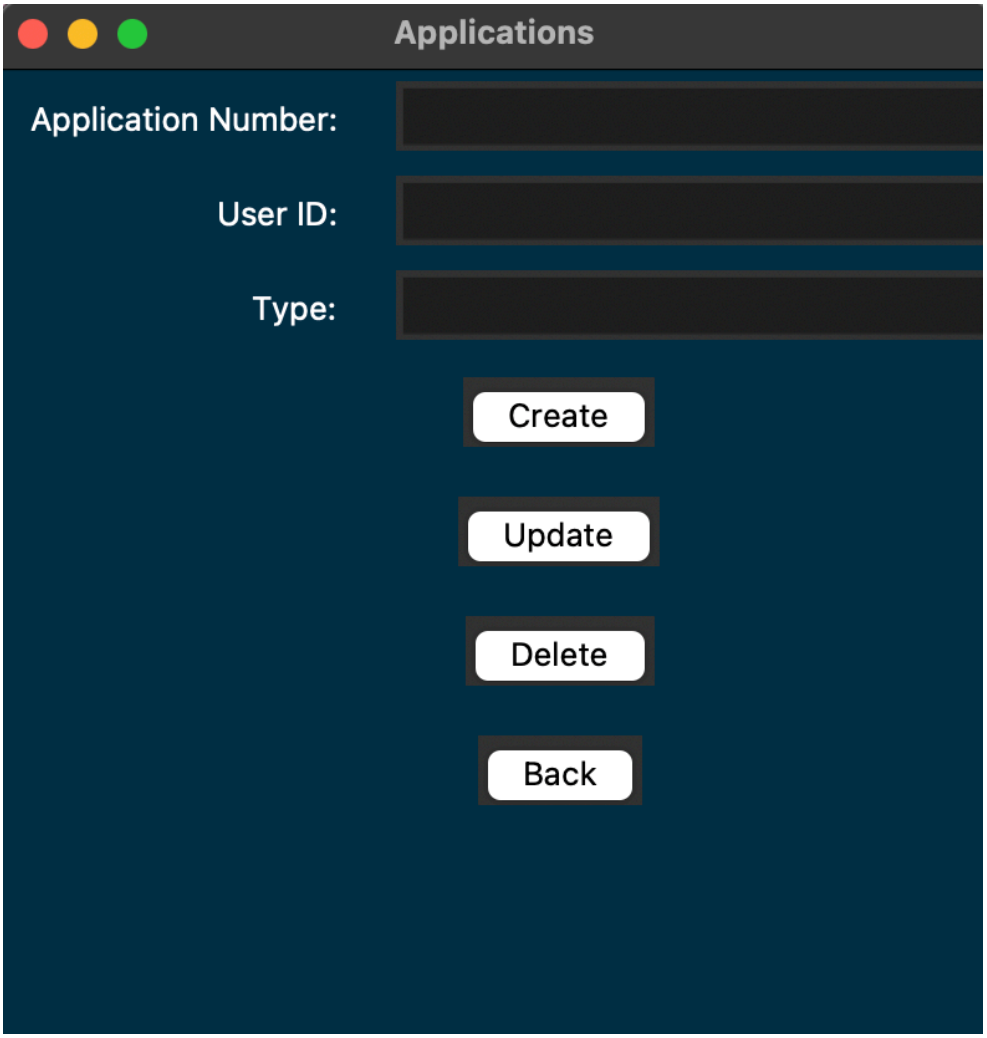

Main Menu
Main Menu
Create New User
New Credit Card
Set Limit
Set Application
Create New Branch
View All
Branch Data:
User Data:
Credit Card Data:
Limits Data:
Applications Data:


Credit Card
Credit Card Name:
Credit Card ID:
Balance:
Type:
User ID:
Create
Update
Delete
Back

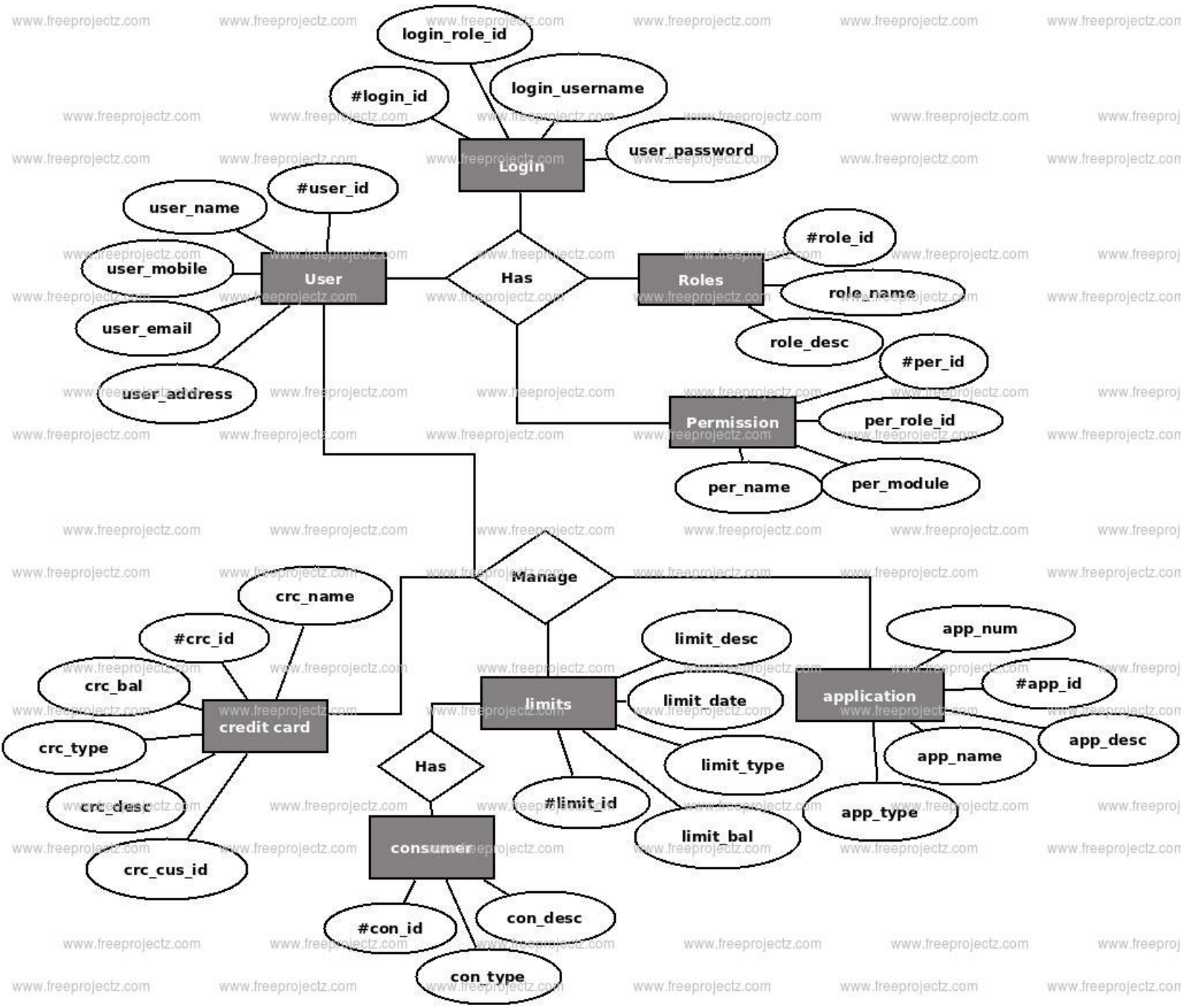

Branch
Branch ID:
Branch Name:
Branch Address:
Branch Manager:
Create
Update
Delete
Back


New User
User ID:
User Name:
Mobile:
Email:
Address:
Create
Update
Delete
Back


Limits
User ID:
Balance:
Credit Card ID:
Create
Update
Delete
Back


Applications
Application Number:
User ID:
Type:
Create
Update
Delete
Back

ER diagram



ER Diagram For Credit Card Approval System

Applications

The credit card management system developed in this project has various applications, including:

1. User Convenience: Provides users with a centralized platform to manage their credit card information, enhancing convenience and accessibility.
2. Data Integrity: Ensures the integrity of user data by securely storing and managing it in a centralized database, minimizing the risk of data loss or unauthorized access.
3. Efficient Management: Streamlines the process of credit card management by offering features such as creating new cards, setting limits, and viewing applications, thereby enhancing efficiency and productivity.
4. Error Reduction: Incorporates error handling mechanisms to minimize the occurrence of errors and ensure a smooth user experience.
5. User Authentication: Implements robust user authentication mechanisms to safeguard user accounts and prevent unauthorized access.
6. User-Friendly Interface: Offers a user-friendly interface designed using Tkinter, making it intuitive and easy to navigate for users of all levels of expertise.

These applications collectively contribute to the effectiveness and utility of the credit card management system, catering to the needs of both users and administrators.

Conclusion

Incorporating Tkinter and Python into the project provided us with a powerful toolkit for developing graphical user interfaces (GUIs) and implementing functionality seamlessly. Tkinter, being a standard GUI library for Python, offered a wide range of widgets and tools for building interactive applications.

By leveraging Python's simplicity and versatility along with Tkinter's robust features, we were able to create an intuitive and user-friendly interface for our credit card management system. Python's extensive standard library and Tkinter's ease of use allowed us to rapidly prototype and iterate on various components of the application.

Furthermore, Python's flexibility enabled us to integrate backend logic seamlessly with the frontend GUI, ensuring smooth communication and data flow between different parts of the system. This integration facilitated the development of a cohesive and efficient application that met our project objectives.

Overall, the combination of Tkinter and Python proved to be a powerful and efficient choice for developing our credit card management system, enabling us to create a functional and user-friendly application that met the requirements of our project.

Project link : https://github.com/aryan-0543/tk-cc-app

Bibliography

1. Python Documentation: Official documentation for the Python programming language. Available online at https://docs.python.org/.
2. Tkinter Documentation: Official documentation for the Tkinter GUI toolkit in Python. Available online at https://docs.python.org/3/library/tkinter.html.
3. SQLite Documentation: Official documentation for SQLite, the database engine used in this project. Available online at https://www.sqlite.org/docs.html.