PERSONAL FINANCE MANAGEMENT SYSTEM

Submitted by

KARTHIKEYA DEVARLA [RA2211003010192]
SIDDARTH JAVVADI [RA2211003010154]
GOWTHAM REDDY KOMMEPALLI [RA2211003010169]

Under the Guidance of

Mr. ARULALAN V

Assistant Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELORS OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

NOVEMBER 2023

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled **“PERSONAL FINANCE MANAGEMENT SYSTEM”** is the bonafide work done by **KARTHIKEYA DEVARLA [RA2211003010192],SIDDARTH JAVVADI [RA2211003010154] and GOWTHAM REDDY KOMMEPALLI [RA2211003010169]** who carried out under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE
Faculty In-Charge
Mr. Arulalan V
Assistant Professor
Department of Computing Technologies
SRM Institute of Science and Technology

HEAD OF THE DEPARTMENT
Dr. M. Pushpalatha
Professor and Head
Department of Computing Technologies
SRM Institute of Science and Technology

ABSTRACT

Money management with intuitive planning and budgeting helps to reduce inessential expenditures. Such expenditures do not add value to an individual’s living standards. They can be saved or invested for better use in the future. Money management also lowers the risk of running out of money. It helps individuals to achieve their financial goals in the long term.

Financial advisors in private banks, insurance firms, and other financial institutes provide personal money management services. Individuals can also process their money management needs through personal finance applications.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	INTRODUCTION	5
2	MODULES	6
3	ARCHITECTURE & DESIGN	7
4	IMPLEMENTATION - CODE SNIPPET	8
5	OUTPUT	14
6	CONCLUSION	

1.INTRODUCTION

A personal finance management system is a comprehensive framework that individuals use to effectively manage their financial resources. It encompasses a set of practices, tools, and strategies aimed at achieving financial stability, achieving financial goals, and making informed financial decisions. Such a system is crucial for anyone seeking to take control of their finances, make the most of their income, and secure their financial future.
Key components of a personal finance management system typically include:

Budgeting: Creating a budget is the foundation of any personal finance management system. It involves tracking income and expenses, setting financial goals, and allocating funds for various categories, such as housing, food, transportation, savings, and entertainment.
Savings and Investments: This part of the system focuses on building an emergency fund, saving for specific goals (like buying a home or retirement), and investing wisely to grow your wealth over time.
Debt Management: Managing debt effectively is crucial. This involves strategies to pay down high-interest debts, such as credit card balances, student loans, or mortgages, and minimizing the financial burden associated with interest payments.
Financial Goals and Planning: Identifying short-term and long-term financial goals is essential. These goals might include saving for a vacation, buying a car, or planning for retirement. A personal finance management system helps you create a roadmap to achieve these objectives.
Expense Tracking: Keeping a close eye on your expenses allows you to identify areas where you can cut back and save more. There are various tools and apps available to help with expense tracking.
Income Maximization: This component focuses on strategies to increase your income, such as career development, side hustles, or passive income sources like investments and rental properties.
Insurance and Risk Management: Ensuring that you have adequate insurance coverage for health, life, property, and other risks is an essential part of protecting your financial well-being.
Tax Planning: Understanding the tax implications of your financial decisions can help you minimize tax liability and keep more of your income.
Estate Planning: Preparing for the distribution of your assets and protecting your family's financial future through wills, trusts, and other estate planning tools.
Regular Review and Adjustment: A personal finance management system isn't static. It requires regular review and adjustment as your financial situation changes, and as you progress towards your goals.

By implementing a personal finance management system, individuals can gain a sense of financial security, reduce stress, and work towards achieving their dreams and aspirations. It provides a structured framework to make informed financial decisions, ultimately helping individuals take control of their financial destiny.

2.MODULES

Random: The Python Random module is a built-in module for generating random integers in Python. These numbers occur randomly and does not follow any rules or instructions. We can therefore use this module to generate random numbers, display a random item for a list or string, and so on.

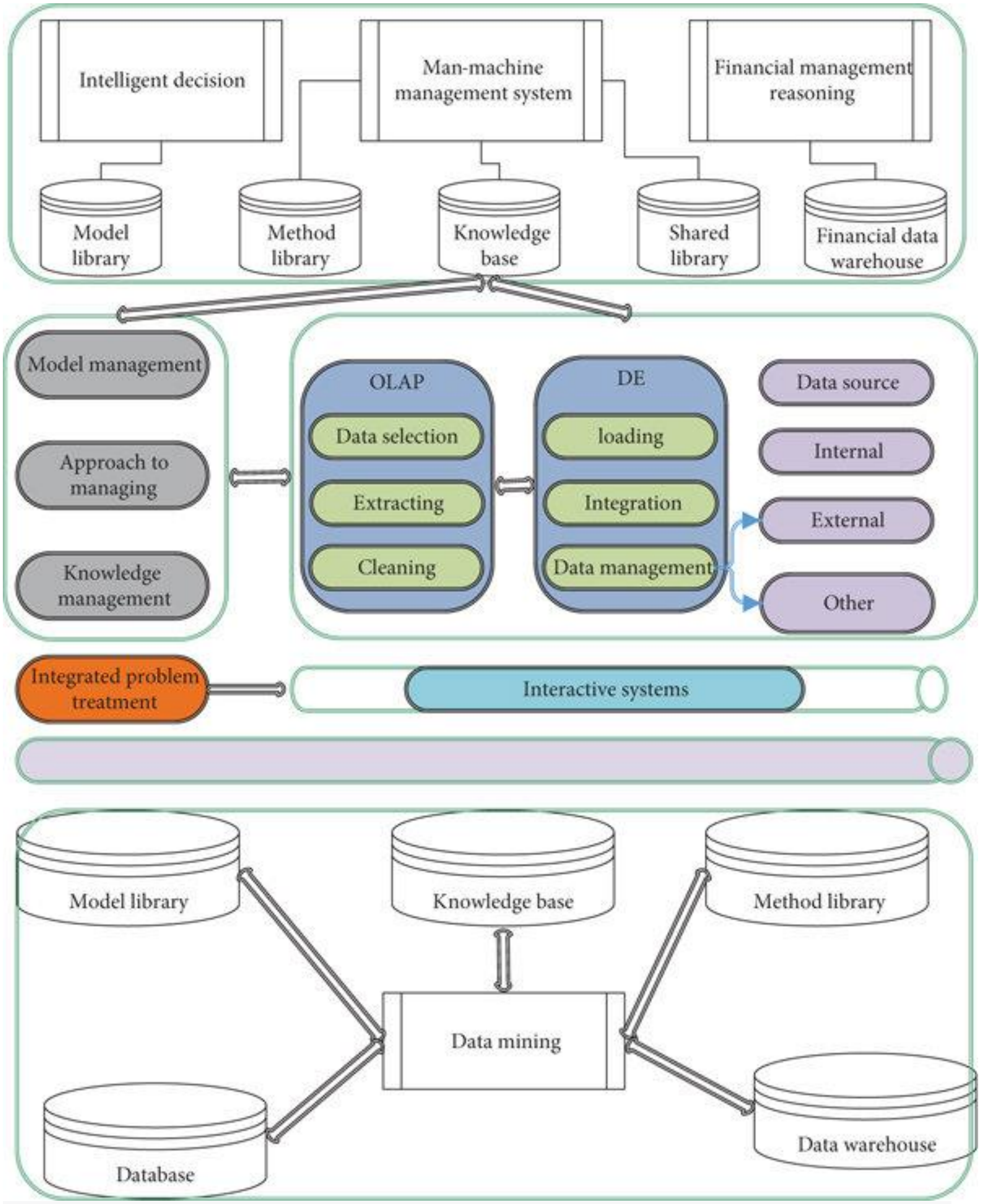
Sqlite3: Python SQLite3 module is used to integrate the SQLite database with Python. It is a standardized Python DBI API 2.0 and provides a straightforward and simple-to-use interface for interacting with SQLite databases. There is no need to install this module separately as it comes along with Python after the 2.5x version.

Libraries:

Operating System: The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality.

Sys: The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

3. ARCHITECTURE AND DESIGN



When designing the architecture, consider a modular and scalable approach that allows for easy expansion and maintenance. Building a personal finance management system is a complex endeavor, and it should prioritize security, user experience, and data accuracy. Additionally, it's important to stay up to date with evolving financial regulations and adapt the system to meet compliance requirements.

4. IMPLEMENTATION CODE SNIPPET

```
import random

import sqlite3



def check_luhn_algo(checkme):
    converting_to_int_list = []
    for item in list(checkme):
        converting_to_int_list.append(int(item))
    luhn_card_no = converting_to_int_list[:-1]
    tmp_list = luhn_card_no.copy()
    for i in range(0, len(tmp_list), 2):
        tmp_list[i] *= 2
        if tmp_list[i] > 9:
            tmp_list[i] -= 9
    checksum = list(str(10 - sum(tmp_list) % 10))
    if len(checksum) != 1:
        checksum = [0]
    luhn_card_no.extend(checksum)
    del tmp_list
    card_no_for_db = ''.join(map(str, luhn_card_no))
    if card_no_for_db == checkme:
        return True
    else:
        return False



def create_card():
    def get_pin():
        pin = ""
        for each in random.sample(range(9), k=4):
            pin += str(each)
        return pin
    conn = sqlite3.connect("card.s3db")
    curr = conn.cursor()
    iin = [4, 0, 0, 0, 0, 0]
    random_acc_no = random.sample(range(9), 9)
    luhn_card_no = []
    luhn_card_no.extend(iin)
    luhn_card_no.extend(random_acc_no)
    tmp_list = luhn_card_no.copy()
    for i in range(0, len(tmp_list), 2):
        tmp_list[i] *= 2
        if tmp_list[i] > 9:
            tmp_list[i] -= 9
    checksum = list(str(10 - sum(tmp_list) % 10))
    if len(checksum) != 1:
        checksum = [0]
    luhn_card_no.extend(checksum)
    del tmp_list
    card_no_for_db = ''.join(map(str, luhn_card_no))
    card_pin_for_db = get_pin()
    print("\nYour card has been created")
    print("Your card number:\n{}\nYour card PIN:\n{}\n".format(card_no_for_db, card_pin_for_db))
    curr.execute('SELECT id from card;')
    db_return = curr.fetchall()
    try:
        listofrows = (lambda l: [item for sublist in l for item in sublist])(db_return)
        myid = max(listofrows)
    except ValueError:
        myid = 0
    dontsqlinjectme = (myid, card_no_for_db, card_pin_for_db)
    curr.execute('INSERT INTO card (id, number, pin) VALUES (?, ?, ?);', dontsqlinjectme)
    conn.commit()



def retrieve_from_db(user_enters_card_no, user_enters_pin):
    conn = sqlite3.connect("card.s3db")
```

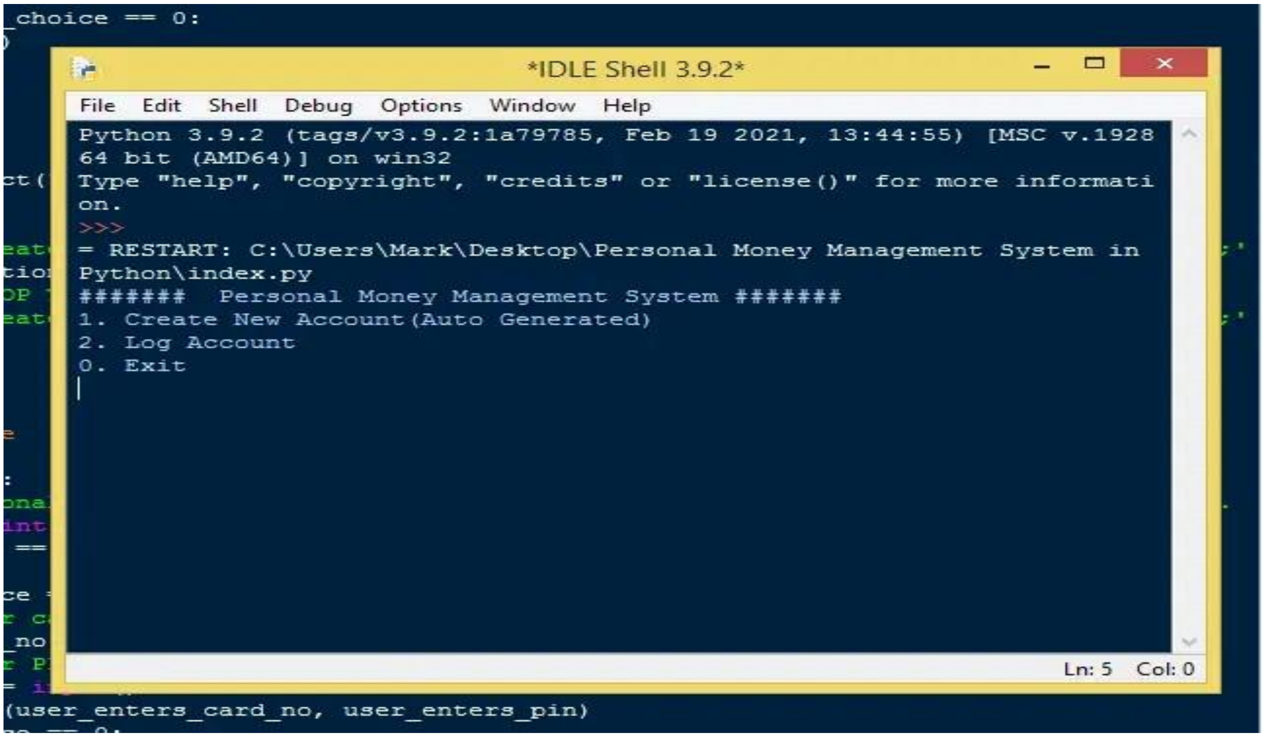
```
curr = conn.cursor()
card_number = user_enters_card_no
pin = user_enters_pin
dontsqlinjectme = (card_number, pin)
curr.execute('SELECT number, pin FROM card WHERE number = ? and pin = ?;', dontsqlinjectme)
db_return = curr.fetchone()
match = False
try:
    if card_number in db_return and pin in db_return:
        match = True
        print("You have successfully logged in!")
except sqlite3.OperationalError:
    print("\nWrong card number or PIN!\n")
except TypeError:
    print("\nWrong card number or PIN!\n")
while match:
    print("Personal Money Management System\n1. Balance\n2. Add Money\n3. Transfer money\n4. Close
                account\n5.Log out\n0.Exit")
    second_menu_choice = int(input())
    if second_menu_choice == 1:
        curr.execute('SELECT balance FROM card WHERE number = ? and pin = ?;', (card_number, pin))
        db_return = curr.fetchone()
        print("\nBalance: {}\n".format(db_return[0]))
    elif second_menu_choice == 2:
        print("\nEnter Money:")
        dontsqlinjectme = (int(input()), card_number, pin)
        curr.execute('UPDATE card SET balance = balance + ? WHERE number = ? and pin = ?;',
                dontsqlinjectme)
        conn.commit()
        print("Money was added!")
    elif second_menu_choice == 3:
        global transfer_destination
        transfer_destination = []
        print("Enter card number:")
        user_enters_transferdest = input()
```

```
if len(user_enters_transferdest) != 16:
    print("\nProbably you made a mistake in the card number.\nPlease try again!\n")
    continue
elif len(user_enters_transferdest) == 16:
    if user_enters_transferdest == card_number:
        print("\nYou can't transfer money to the same account!\n")
        continue
    elif not check_luhn_algo(user_enters_transferdest):
        '# IF CHECK LUHN ALGO RETURNS FALSE. NOT FALSE = TRUE AND THEN WE
                CONTINUE'
        print("\nLUHN CHECK:Probably you made a mistake in the card number.\nPlease try again!\n")
        continue
    else:
        transfer_destination = (int(user_enters_transferdest),)
curr.execute('SELECT number FROM card WHERE number = ?;', transfer_destination)
db_return = curr.fetchone()
try:
    len(db_return)
    print("\nEnter how much money you want to transfer:\n")
    user_enters_transfermoney = int(input())
    curr.execute('SELECT balance FROM card WHERE number = ? and pin = ?', (card_number, pin))
    db_return = curr.fetchone()
    if user_enters_transfermoney > db_return[0]:
        print("\nNot enough money!\n")
        continue
    else:
        curr.execute('UPDATE card SET balance = balance + ? WHERE number = ?;', (
            user_enters_transfermoney, int(user_enters_transferdest)))
        curr.execute('UPDATE card SET balance = balance - ? WHERE number = ?;', (
            user_enters_transfermoney, card_number))
        conn.commit()
        print("\nSuccess!\n")
        continue
except TypeError:
    print("\nSuch a card does not exist.\n")
```

```
                continue

        elif second_menu_choice == 4:
            dontsqlinjectme = (card_number, pin)
            curr.execute('DELETE FROM card WHERE number = ? and pin = ?;', dontsqlinjectme)
            conn.commit()
            print("\nThe account has been closed!\n")
            break
        elif second_menu_choice == 5:
            print("You have successfully logged out!")
            match = False
        elif second_menu_choice == 0:
            print("Bye!")
            conn.close()
            exit()



def create_db():
    conn = sqlite3.connect("card.s3db")
    curr = conn.cursor()
    try:
        curr.execute('create table card (id INTEGER, number TEXT, pin TEXT, balance INTEGER default 0);')
    except sqlite3.OperationalError:
        curr.execute('DROP TABLE card;')
        curr.execute('create table card (id INTEGER, number TEXT, pin TEXT, balance INTEGER default 0);')
    finally:
        conn.commit()



program_is_running = True
create_db()
while program_is_running:
    print("#######  Personal Money Management System #######\n1. Create New Account(Auto
                        Generated)\n2. Log Account\n0. Exit")
    first_menu_choice = int(input())
```

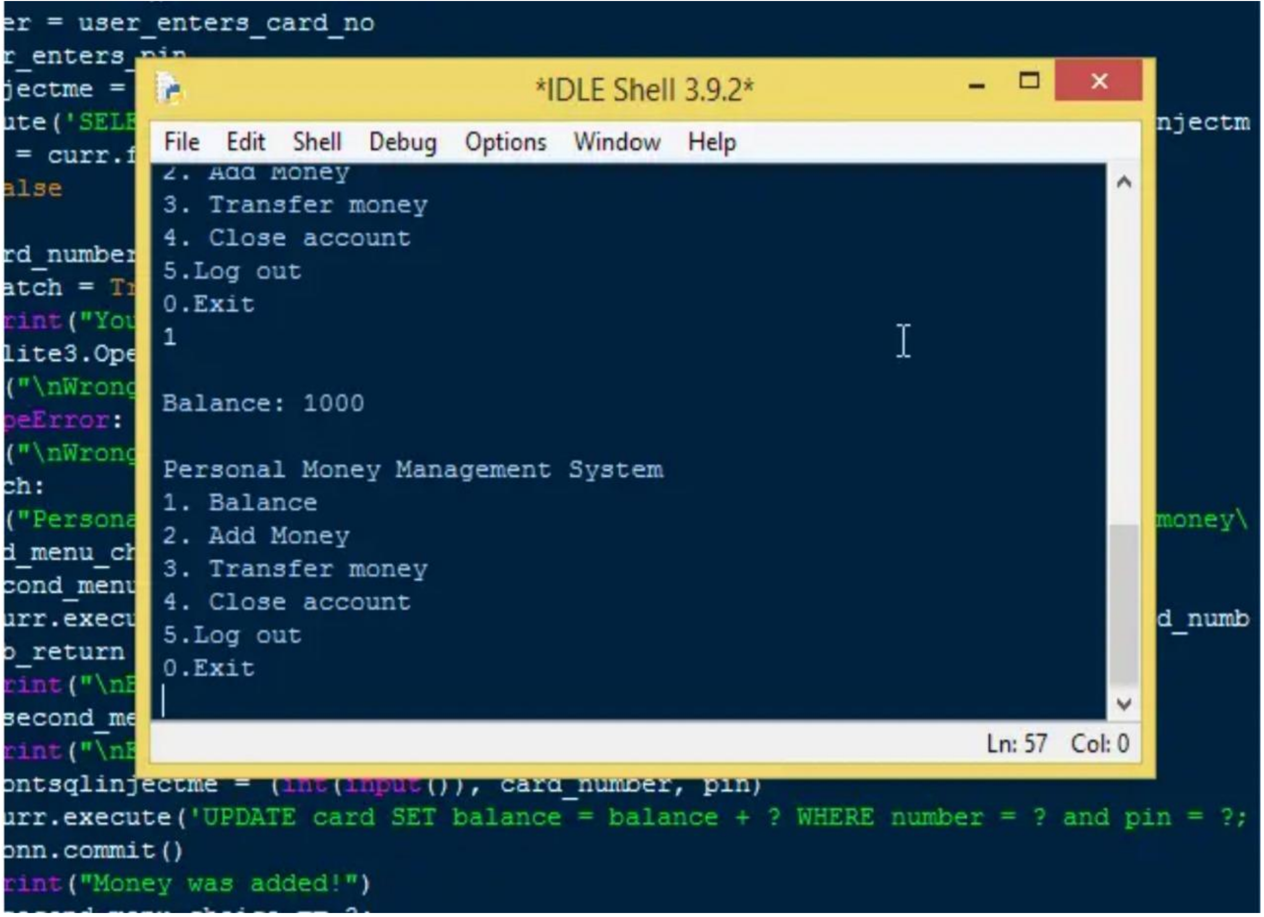
```
if first_menu_choice == 1:
    create_card()
elif first_menu_choice == 2:
    print("Enter your card number:")
    user_enters_card_no = input()
    print("Enter your PIN:")
    user_enters_pin = input()
    retrieve_from_db(user_enters_card_no, user_enters_pin)
elif first_menu_choice == 0:
    print("Thank you! Bye!")
                                program_is_running = False
```

5. OUTPUT SCREENSHOTS



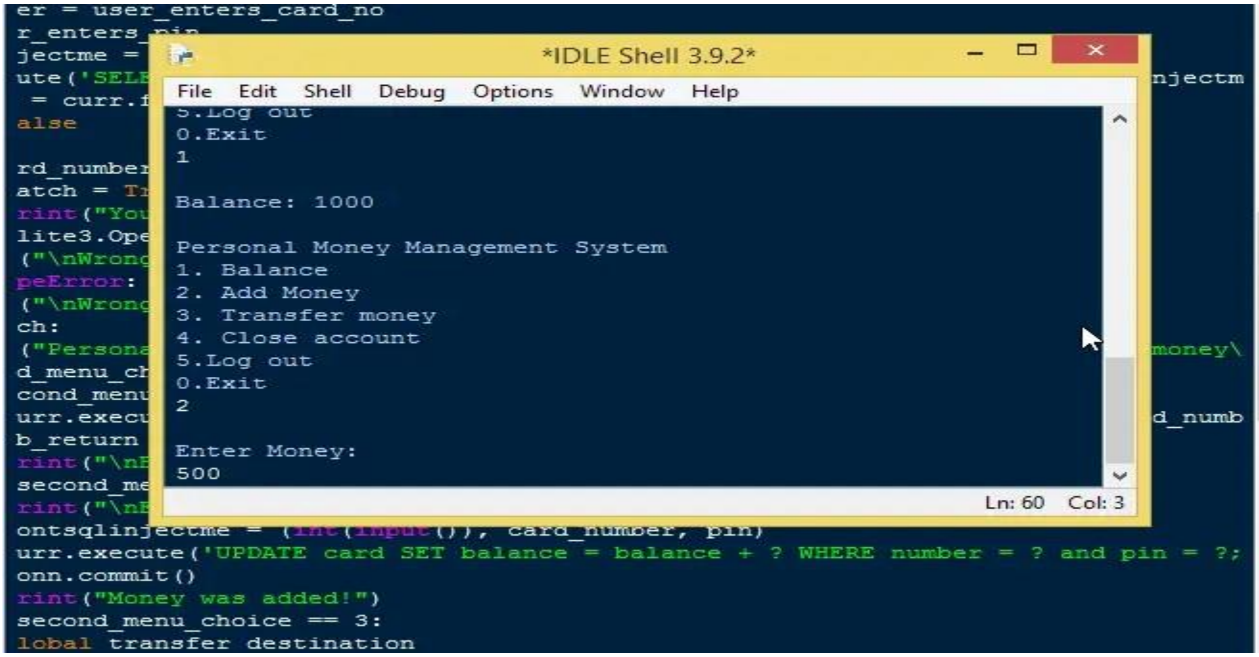
```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informati
on.
>>>
= RESTART: C:\Users\Mark\Desktop\Personal Money Management System in
Python\index.py
#######  Personal Money Management System #######
1. Create New Account(Auto Generated)
2. Log Account
0. Exit
```

```
(user_enters_card_no, user_enters_pin)
```



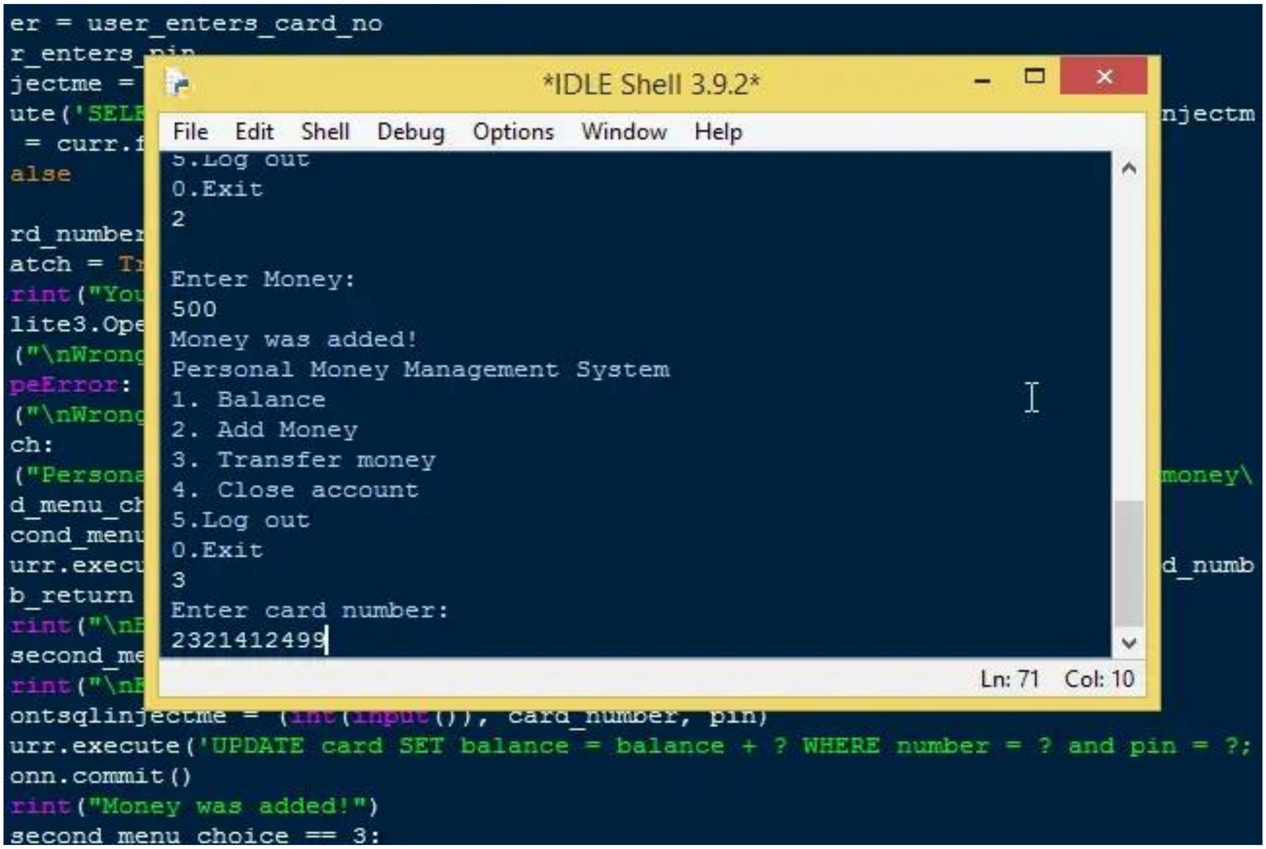
```
2. Add Money
3. Transfer money
4. Close account
5.Log out
0.Exit
1

Balance: 1000

Personal Money Management System
1. Balance
2. Add Money
3. Transfer money
4. Close account
5.Log out
0.Exit
```

```
urr.execute('UPDATE card SET balance = balance + ? WHERE number = ? and pin = ?;
onn.commit()
rint("Money was added!")
```



```
5.Log out
0.Exit
1

Balance: 1000

Personal Money Management System
1. Balance
2. Add Money
3. Transfer money
4. Close account
5.Log out
0.Exit
2

Enter Money:
500
```

```
er = user_enters_card_no
r_enters_pin
jectme =
ute('SELE                                                                   njectm
 = curr.f
alse

rd_number
atch = Tr
rint("You
lite3.Ope
("\nWrong
peError:
("\nWrong
ch:
("Persona                                                                   money\
d_menu_ch
cond_menu
urr.execu                                                                   d_numb
b_return
rint("\nE
second_me
rint("\nE
ontsqlinjectme = (int(input()), card_number, pin)
urr.execute('UPDATE card SET balance = balance + ? WHERE number = ? and pin = ?;
onn.commit()
rint("Money was added!")
second_menu_choice == 3:
lobal transfer_destination
```



```
5.Log out
0.Exit
2

Enter Money:
500
Money was added!
Personal Money Management System
1. Balance
2. Add Money
3. Transfer money
4. Close account
5.Log out
0.Exit
3
Enter card number:
2321412499
```

```
er = user_enters_card_no
r_enters_pin
jectme =
ute('SELE                                                                   njectm
 = curr.f
alse

rd_number
atch = Tr
rint("You
lite3.Ope
("\nWrong
peError:
("\nWrong
ch:
("Persona                                                                   money\
d_menu_ch
cond_menu
urr.execu                                                                   d_numb
b_return
rint("\nE
second_me
rint("\nE
ontsqlinjectme = (int(input()), card_number, pin)
urr.execute('UPDATE card SET balance = balance + ? WHERE number = ? and pin = ?;
onn.commit()
rint("Money was added!")
second_menu_choice == 3:
```

6. CONCLUSION

In conclusion, a personal finance management system is an indispensable tool for individuals seeking to take control of their financial well-being. It serves as a structured framework that empowers users to efficiently manage their finances, make informed decisions, and work toward their financial goals. The system's architecture and design should prioritize user-friendly interfaces, data security, and insightful financial insights, all while offering the flexibility to adapt to evolving financial needs.

With such a system in place, individuals can achieve greater financial stability, reduce stress, and ultimately secure their financial future. By embracing the principles of budgeting, savings, debt management, and smart financial planning, anyone can navigate the complexities of personal finance with confidence. Regular review, updates, and adherence to privacy and compliance standards are essential to ensuring the system's ongoing effectiveness. Ultimately, a well-designed personal finance management system equips individuals with the tools and knowledge to make their money work for them, helping them achieve their financial aspirations and objectives.

In essence, a personal finance management system is not just a tool; it is a philosophy of financial responsibility and empowerment. It encourages individuals to be proactive, informed, and strategic in their financial decisions, enabling them to take charge of their financial destinies and work towards a more secure and prosperous future.