**PERSONAL FINANCIAL MANAGEMENT SYSTEM**

****

**A Project Documentation Submitted to the**

**College of Information Technology**

**Cagayan de Oro College**

**PHINMA Education**

**In Partial Fulfilment of the requirements**

**For Computer Programming I**

**Submitted By:**

**Aguirre, Louwie Jie C.**

**Bansing, Paul Markuss S.**

**Bansing, Rastel Cyrus R.**

**Cabual, Keshtian Mark**

**Elison, Cedric E.**

**Galeon, Jay Lourenz C.**

**Perater, Royen July**

**Sismundo, Mark Reniel L.**

**March 2025**

**Introduction**

Present day living demands secure effective management of personal finances. The software application was created to help users achieve better financial control because it allows users to track their money alongside expenses and savings functions with protected data storage features.

The Personal Finance Management System (PFMS) tracks basic income and expenses, but it lacks key aspects for effective financial management. The lack of budgeting tools, goal setting, and data visualization makes it difficult for consumers to plan and monitor their finances effectively. Furthermore, the single-user functionality restricts its usability for multiple users. Kozhevnikov et al. (2019) argue that modern finance systems should go beyond transaction tracking to include elements that improve financial planning. Without these features, the system is less successful at facilitating informed financial decisions.

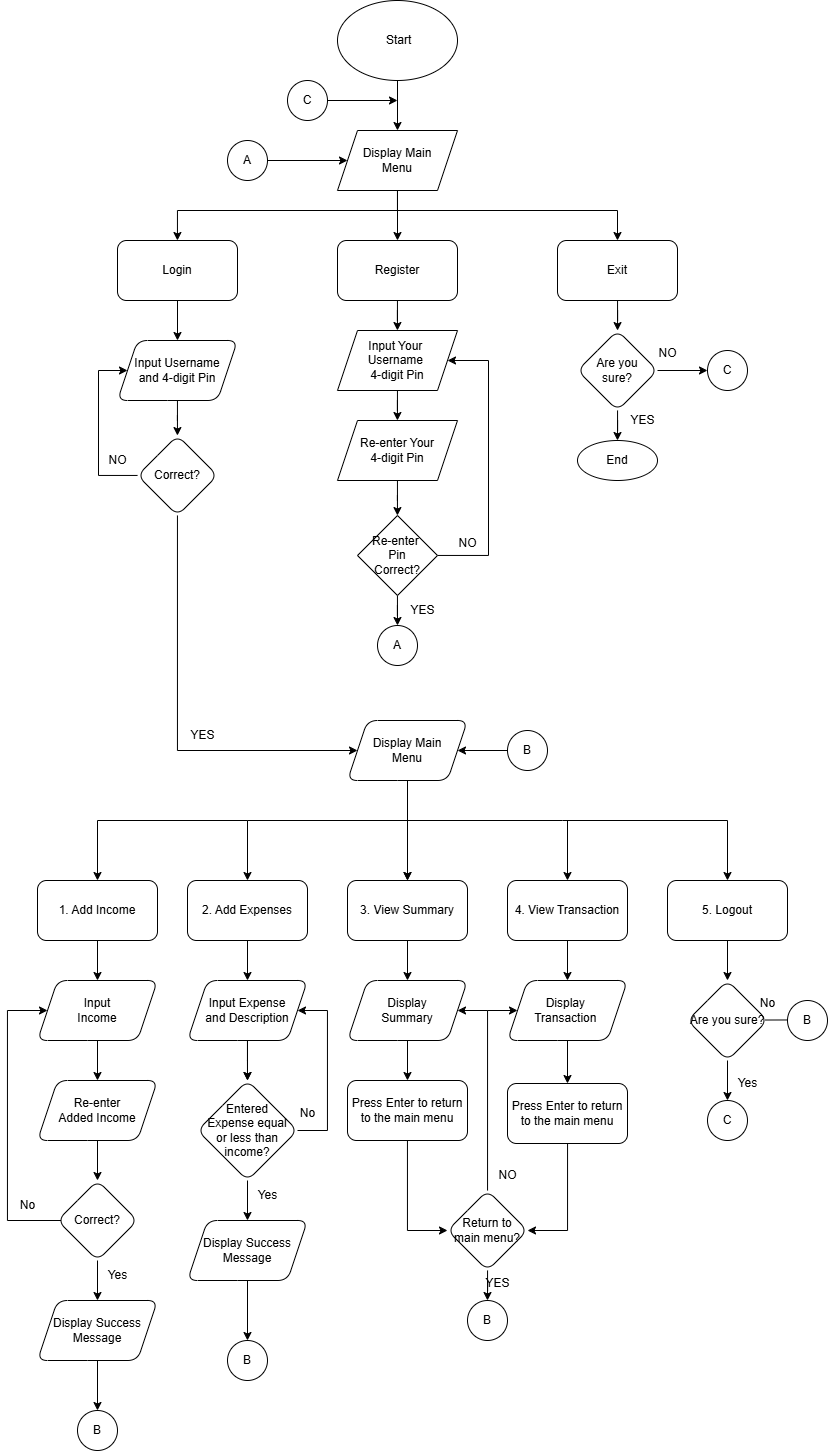
Users may easily manage their accounts by using this personal finance management system to effectively track their income and expenses. People can log in to enter earnings, classify expenses, and view financial summary with secure user verification. By ensuring precise financial tracking, the system assists users in making wise selections and forming responsible spending habits. Users may successfully budget, prevent overspending, and strive toward long-term financial stability by keeping an accurate record of all transactions.

The objectives of this personal finance management system are to ensure long-term financial stability and success by providing users with an organized approach to tracking income, recording spending, and monitoring their overall financial health. This system aims to ensure safe user authentication, track income and expenses, provide financial summaries, and keep a transaction history. By offering these capabilities through an interactive and user-friendly interface, the system assists users in making informed financial decisions and developing responsible spending habits.

The Personal Finance Management System (PFMS) in the given code possesses multiple essential characteristics similar to those outlined in recent publications. Both systems include user authentication, guaranteeing safe access via a username and PIN login, which is a highly suggested method for financial security. Moreover, the system offers income and expense tracking, enabling users to log transactions with details, a basic functionality present in nearly all financial management apps. As indicated by the research conducted by Kozhevnikov et al. (2019), the PFMS preserves a transaction history, allowing users to examine previous financial activities, and offers a financial summary, showcasing total income, expenses, and balance to aid in improved financial decision-making. Nonetheless, the literature also points out various characteristics that could improve the system even more. Kozhevnikov et al. (2019) highlight the significance of budgeting and goal setting, allowing users to organize monthly costs and monitor financial objectives, which is presently lacking in the system.

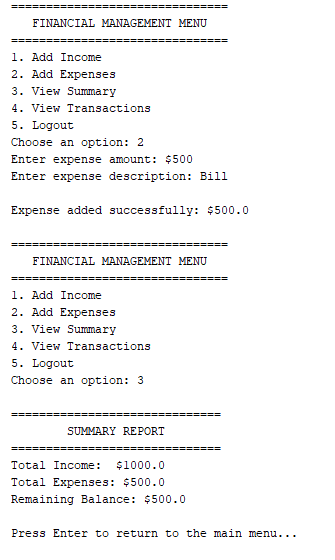
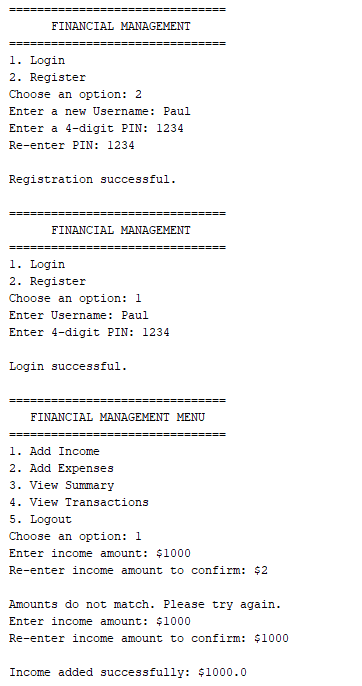
Moreover, data visualization tools like charts and graphs can assist users in analysing spending patterns more efficiently. Additionally, multi-account management is another characteristic present in sophisticated PFMS applications, enabling users to monitor finances across various accounts instead of limiting themselves to a single user session. Although the current code addresses key elements of personal finance tracking, adding budgeting, financial objectives, and data visualization may enhance the system’s effectiveness and user experience.

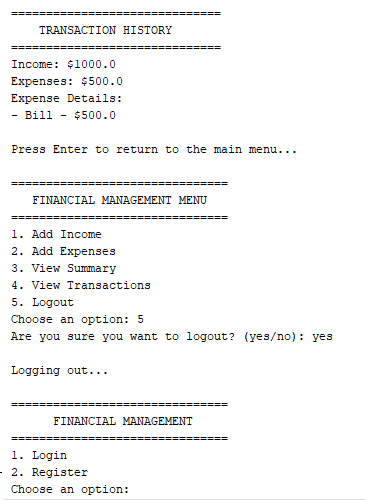
**Flowchart**

****

**Figure 1.0 System Flow Chart**

**Output Screenshot**

****

****

**Figure 2.0 System Output Screenshot**

**Program Code**

import java.util.HashMap;

import java.util.ArrayList;

import java.util.Scanner;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class FinancialM {

private static HashMap<String, String> users = new HashMap<>();

private static double income = 0, expenses = 0;

private static ArrayList<String> expenseDescriptions = new ArrayList<>();

private static ArrayList<String> incomeDescriptions = new ArrayList<>();

private static Scanner scanner = new Scanner(System.in);

private static String currentUser = "";

private static DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");

public static void main(String[] args) {

while (true) {

System.out.println("\n===============================");

System.out.println(" FINANCIAL MANAGEMENT ");

System.out.println("===============================");

System.out.println("1. Login");

System.out.println("2. Register");

System.out.println("3. Exit");

System.out.print("Choose an option: ");

int choice = getValidIntegerInput();

if (choice == 1) {

login();

} else if (choice == 2) {

register();

} else if (choice == 3) {

if (confirmExit()) {

System.out.println("\nExiting application...");

break;

}

} else {

System.out.println("Invalid choice.");

}

}

}

private static int getValidIntegerInput() {

while (true) {

if (scanner.hasNextInt()) {

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

return choice;

} else {

System.out.println("Invalid input. Please enter a number.");

scanner.next(); // Consume invalid input

}

}

}

private static void login() {

System.out.print("Enter Username: ");

String username = scanner.nextLine();

System.out.print("Enter 4-digit PIN: ");

String pin = scanner.nextLine();

if (users.containsKey(username) && users.get(username).equals(pin)) {

currentUser = username;

System.out.println("\nLogin successful.");

manageFinances();

} else {

System.out.println("\nIncorrect username or PIN.");

}

}

private static void register() {

String username;

while (true) {

System.out.print("Enter a new Username: ");

username = scanner.nextLine();

if (users.containsKey(username)) {

System.out.println("Username already exists. Please choose a different one.");

} else {

break;

}

}

String pin;

while (true) {

System.out.print("Enter a 4-digit PIN: ");

pin = scanner.nextLine();

if (pin.matches("\\d{4}")) {

break;

} else {

System.out.println("Invalid PIN. Please enter exactly 4 digits.");

}

}

System.out.print("Re-enter PIN: ");

String confirmPin = scanner.nextLine();

if (pin.equals(confirmPin)) {

users.put(username, pin);

System.out.println("\nRegistration successful.");

} else {

System.out.println("\nPINs do not match.");

}

}

private static void manageFinances() {

while (true) {

System.out.println("\n===============================");

System.out.println(" FINANCIAL MANAGEMENT MENU ");

System.out.println("===============================");

System.out.println("1. Add Income");

System.out.println("2. Add Expenses");

System.out.println("3. View Summary");

System.out.println("4. View Transactions");

System.out.println("5. Logout");

System.out.print("Choose an option: ");

int choice = getValidIntegerInput();

switch (choice) {

case 1:

addIncome();

break;

case 2:

addExpenses();

break;

case 3:

viewSummary();

break;

case 4:

viewTransactions();

break;

case 5:

if (confirmLogout()) {

System.out.println("\nLogging out...");

currentUser = "";

return;

}

break;

default:

System.out.println("\nInvalid choice. Try again.");

}

}

}

private static void addIncome() {

double incomeAmount = 0;

boolean validIncome = false;

while (!validIncome) {

System.out.print("Enter income amount: $");

double amount = getValidDoubleInput();

System.out.print("Re-enter income amount to confirm: $");

double reEnteredAmount = getValidDoubleInput();

if (amount == reEnteredAmount) {

incomeAmount = amount;

validIncome = true;

} else {

System.out.println("The amounts do not match. Please enter the income again.");

}

}

String timestamp = LocalDateTime.now().format(formatter);

income += incomeAmount;

incomeDescriptions.add("Income added: $" + incomeAmount + " on " + timestamp);

System.out.println("\nIncome added successfully: $" + incomeAmount);

}

private static void addExpenses() {

System.out.print("Enter expense amount: $");

double amount = getValidDoubleInput();

scanner.nextLine(); // Consume the newline left over

System.out.print("Enter expense description: ");

String description = scanner.nextLine();

if (amount > 0 && amount <= income - expenses) {

expenses += amount;

String timestamp = LocalDateTime.now().format(formatter);

expenseDescriptions.add(description + " - $" + amount + " on " + timestamp);

System.out.println("\nExpense added successfully: $" + amount);

} else {

System.out.println("\nExpense exceeds available balance.");

}

}

private static double getValidDoubleInput() {

while (true) {

if (scanner.hasNextDouble()) {

double amount = scanner.nextDouble();

scanner.nextLine(); // Consume the newline character

return amount;

} else {

System.out.println("Invalid input. Please enter a valid numeric value.");

scanner.next(); // Consume invalid input

}

}

}

private static void viewSummary() {

System.out.println("\n==============================");

System.out.println(" SUMMARY REPORT ");

System.out.println("==============================");

System.out.println("Total Income: $" + income);

System.out.println("Total Expenses: $" + expenses);

System.out.println("Remaining Balance: $" + (income - expenses));

System.out.println("Last Updated: " + LocalDateTime.now().format(formatter));

System.out.print("\nPress 'B' to go back to the menu: ");

String input = scanner.nextLine().trim().toLowerCase();

if (input.equals("b")) {

if (confirmBack()) {

return; // Will return to the previous menu (manageFinances())

} else {

viewSummary(); // Display summary again

}

} else {

System.out.println("Invalid input. Returning to the main menu.");

return;

}

}

private static void viewTransactions() {

System.out.println("\n==============================");

System.out.println(" TRANSACTION HISTORY ");

System.out.println("==============================");

for (String record : incomeDescriptions) {

System.out.println(record);

}

for (String record : expenseDescriptions) {

System.out.println(record);

}

System.out.print("\nPress 'B' to go back to the menu: ");

String input = scanner.nextLine().trim().toLowerCase();

if (input.equals("b")) {

if (confirmBack()) {

return; // Will return to the previous menu (manageFinances())

} else {

viewTransactions(); // Display transactions again

}

} else {

System.out.println("Invalid input. Returning to the main menu.");

return;

}

}

private static boolean confirmBack() {

System.out.print("Are you sure you want to go back? (yes/no): ");

String response = scanner.nextLine().trim().toLowerCase();

return response.equals("yes");

}

private static boolean confirmLogout() {

System.out.print("Are you sure you want to logout? (yes/no): ");

String response = scanner.nextLine().trim().toLowerCase();

return response.equals("yes");

}

private static void promptContinue() {

System.out.print("\nPress Enter to return to the main menu...");

scanner.nextLine(); // Wait for the user to press Enter

}

private static boolean confirmExit() {

System.out.print("Are you sure you want to exit? (yes/no): ");

String response = scanner.nextLine().trim().toLowerCase();

return response.equals("yes");

}

}

**Documentation**

****