Group Name: TiTechnics

Software Construction and Development

Deliverable 01

2nd May 2024

Khadija Bibi SP-21424

Fatima Rehmanullah SP-21282

Muhammad Alyan SP-21319

Konain Raza SP-21303

Aazeen Iftikhar SP-21329

Instructor: Miss Fatima Gillani

Software Requirement Specification Document

Personal Finance Tracker

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the development of a Personal Finance Tracker application. This application aims to help users track their expenses, income, budgets, and view transaction history in a simple and intuitive manner. This application is to be developed using JAVA and MySQL.

1.2 Scope

Our Personal Finance Tracker will include the following functionalities:

- User Login, registration, and authentication
- Income Tracking and Budgeting
- Expense Tracking and Transaction History
- Loan Management
- Bill Payments

2 Functional Requirements

2.1 User Authentication

2.1.1 The application shall allow new users to register by providing their username, email and a password for their account.

2.1.2 The application shall ensure that email is in the right format before inserting the information to database.

2.1.3 The application shall search for the availability of username among the existing usernames.

2.1.4 The application shall salt and hash the password by using SHA-256 cryptographic technique before registering the user.

2.1.5 Existing users shall be able to log in by entering their username and password.

2.1.6 The application shall authenticate users' credentials before allowing the access to account.

2.2 Expense Tracking and Budgeting

2.2.1 Users shall be able to add, edit, and delete expense entries.

2.2.2 Each expense entry shall include the date, amount, category, and description.

2.2.3 Users shall be notified when the expense limit exceeds their specified budget. Allow users to manually input expenses, including details such as date, amount, category, description, and payment method.

2.2.4 The application shall allow users to categorize expenses into predefined categories (e.g., groceries, utilities, transportation, and entertainment).

2.2.5 The application shall provide the option for users to customize or create their own expense categories.
2.2.6 The application shall allow users to edit or delete previously recorded expenses.
2.2.7 The application shall allow users to specify the payment method used for each expense entry.
2.2.8 The application shall allow users to view expenses through search and filtering.
2.2.9 The application shall allow users to export expense data in common formats such as CSV or PDF for backup purposes or further analysis.
2.2.10 Users should be able to set budget limits for different categories (e.g., groceries, utilities, entertainment).
2.2.11 The system should allow users to specify the duration of the budget (e.g., monthly, weekly).
2.2.12 The system should track expenses in each category and compare them against the set budget limits.
2.2.13 Users should be able to view a summary of their spending against the budget for each category.
2.2.14 Users should receive notifications or alerts when they are close to or have exceeded their budget limits in any category.
2.2.15 The system should provide options for users to customize the threshold for budget alerts.

2.3 Income Tracking and Transaction History

2.3.1 Users can add individual income entries by providing the required information: date, amount, and description.
2.3.2 Each entry includes an "Edit" button and a "Delete" button for users to modify or remove the entry if needed.
2.3.3 Users can edit existing income entries by clicking the "Edit" button associated with each entry.
2.3.4 Editing allows users to update the date, amount, or description of the income entry.
2.3.5 Users can delete existing income entries by clicking the "Delete" button associated with each entry.
2.3.6 Deleting removes the income entry from the list.
2.3.7 Once users have added, edited, or deleted individual income entries, they can click the "Submit Income" button to finalize and save all changes
2.3.8 Clicking this button adds all income entries to the user's income records.
2.3.9 The system should perform validation checks to ensure that required fields (date, amount) are filled in before allowing the user to add or edit an income entry.
2.3.10 Proper error messages should be displayed if validation fails.

2.3.11 Upon successful submission of income entries, the system should provide a confirmation message to the user, indicating that the entries have been saved successfully.

2.4 Loan Management

2.4.1 Users should be able to input loan details such as principal amount, interest rate, and term.
2.4.2 The system should validate the input data to ensure accuracy and completeness.
2.4.3 Users should be able to compare different loans based on interest rates, terms, and monthly payments.
2.4.4 The system should provide a comparison tool with parameters for comparison.
2.4.5 Users should be able to delete loans with a confirmation prompt.
2.4.6 The system should prompt the user for confirmation before deleting the loan.
2.4.7 Users should have access to a detailed payment history including dates, amounts, and remaining balances for each payment.
2.4.8 The system should store payment history data associated with each loan.
2.4.9 Users should be able to view a comprehensive list of existing loans with essential details for each loan entry.
2.4.10 The system should provide a clear and organized display of existing loans.

2.5 Bill Payments

2.5.1 Users should be able to input bill dates in the format "DD-MM-YYYY" via the command line.
2.5.2 The system should schedule reminders for each bill date entered by the user.
2.5.3 Reminders for bill payments should be displayed when their scheduled time arrives.
2.5.4 The system should handle cases where users input invalid date formats gracefully and prompt them to enter the date in the correct format.
2.5.5 The system should terminate gracefully when the user inputs "done" to signify the end of bill date entries.

3 Non-functional Requirements

3.1 The application shall be developed using Java and MySQL as the primary technologies.
3.2 The user interface shall be intuitive and easy to navigate i.e., should align with UX principles.
3.3 The application shall provide clear and concise error messages in case of invalid input.
3.4 The application shall ensure the integration of salting with hashing to maximize the security.
3.5 The system should be able to handle a large volume of expense entries efficiently, ensuring fast response times even during peak usage periods.
3.6 The system should maintain accurate records of expenses and ensure data integrity, with minimal risk of data loss or corruption.

3.7 The user interface should be intuitive and easy to navigate, with clear instructions and prompts for entering and managing expenses.

3.8 The budgeting module should efficiently calculate and update spending summaries and alerts in real-time, ensuring a responsive user experience.

3.9 The budgeting module should accurately track expenses against budget limits, with minimal risk of calculation errors or discrepancies.

3.10 The dashboard shall have an intuitive and visually appealing user interface design using Swing components.

3.11 The dashboard shall implement basic security measures to protect user data and prevent unauthorized access.

3.12 The income tracking module shall implement security measures to protect sensitive financial data.

3.13 The income tracking module shall be optimized for performance, ensuring fast loading times and smooth user experience.

3.14 Reminders should be scheduled accurately, ensuring that users receive timely notifications for their bill payments.

3.15 The system should efficiently handle multiple bill dates and reminders, ensuring that it remains responsive even with a large number of scheduled reminders.