Financial Management System with Machine Learning Support

Implementation of Expense Management System

Caleigh Susan Anak Jeffry
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
Sarawak, Malaysia
82476@siswa.unimas.my

***Abstract*—Managing personal and shared expenses is challenging for university students. This project introduces Spendette, a web-based expense management system that supports budget tracking, share expense management and spending analysis. The system integrates with Machine Learning Model to analyze spending patterns that promote better financial awareness among students.**

Keywords component; Expense tracking, shared expenses, Machine Learning, financial management.

I. INTRODUCTION

Managing personal finances is an essential skill for university students, particularly in a cashless society where digital transactions are increasingly common. While cashless payment offers convenience, they also make it difficult for students to track daily spending and control expenses effectively. Small and frequent transactions lead to overspending and difficulty adhering to a personal budget. These challenges further intensified when managing shared expenses with peers, which often rely on manual tracking methods. These manual tracking methods such as spreadsheets, messaging apps, or verbal reminders are prone to errors and delays.

To address these issues, this project introduces Spendette, a web-based financial management system designed to support both personal and shared expense management. The system enables students to record, organize, and monitor their spending and income in a structured manner. In addition, Spendette incorporates Machine Learning, which evaluates historical spending patterns, detects unusual expenditures and offers personalized budgeting recommendations. By combining automated expense and income tracking, group finance management, and predictive insights, Spendette aims to enhance financial awareness and assist students in achieving long-term financial stability.

A. Problem Statement

University students often lack the knowledge and experience needed to manage their finances, leading to poor financial decisions and financial stress that can affect their mental health and wellbeing [1]. This issue frequently happens among students that face difficulties in managing their financial situation efficiently, especially tracking both of their personal and shared expenses. In addition, frequent small purchases can accumulate unexpectedly and result in budget overruns, therefore, systematically tracking expenses through receipt collection and categorized spending records enables individuals to identify and manage areas of excessive expenditure more effectively [2]. Manually tracking their spending and budget can lead to a range of problems such as:

- Difficulty in expense tracking.
- Limited budget control and financial awareness
- Challenges in managing shared financial responsibilities.

B. Aims and Objectives

The main objective of this project is to develop Spendette, a web-based financial management system that enables university students to manage personal and shared expenses effectively while improving financial awareness and spending behavior. The objectives of the project are:

- To develop a responsive web application that enables users to record, manage and monitor both personal and shared financial transactions to ensure accuracy and transparency.
- To generate clear financial summaries and balance sheet that assist users in monitoring their spending effectively.
- To implement a machine learning model to analyze spending habits and provide personalized budgeting recommendations for improved financial planning.

II. BACKGROUND STUDY

A background study of three similar existing systems which are Pennywise, Spendee and Goodbudget is conducted and compared to provide a comparative foundation for the development of the proposed system. The three existing

systems are being studied in terms of their design and characteristics to define each of their strengths and limitations.

A. Comparison Between Existing System and Proposed System

While all aim to enhance financial management efficiency, each system emphasizes distinct features ranging from data interpretability to user engagement. Table 1 illustrates the differences and similarities of existing systems and the comparison of existing systems and the proposed system.

Table 1 Comparison Between Systems

Feature	Existing Systems			Proposed System
	Penny wise	Spende e	Goodb udget	Spendett e
Personal Expense Management	Yes	Yes	Yes	Yes
Group Expense Management	No	No	No	Yes
Income Management	No	Yes	Yes	Yes
Fixed Bills Management	Yes	Yes	No	Yes
Expense Category Management	Yes	Yes	Yes	Yes
Dashboard	Yes	Yes	Yes	Yes
Financial Report	Yes	Yes	Yes	Yes
Reminders & Notifications	Yes	Yes	No	Yes
Machine Learning Model	Yes	Yes	No	Yes

B. Technology Review

- Frontend Development: Built with HTML, CSS and JavaScript to provide a responsive and interactive interface suitable for expense tracking.
- Backend & Database: PHP and SQL for managing authentication, data processing and CRUD operations in financial management systems.
- Machine Learning: Random Forest model will be trained within the system to generate personalized budget recommendations for users and to detect potential overspending.

III. METHODOLOGY

This section outlines the methodological framework adopted in the development of Spendette, a financial management system designed to support both personal and shared expenses. The development of Spendette, a financial management system with machine learning support, will be completed using the Agile methodology, specifically Scrum framework. Scrum is an agile framework that combines iterative and incremental models, enabling successive builds with expanding functionality for object-oriented software development [3].

A. Agile Methodology

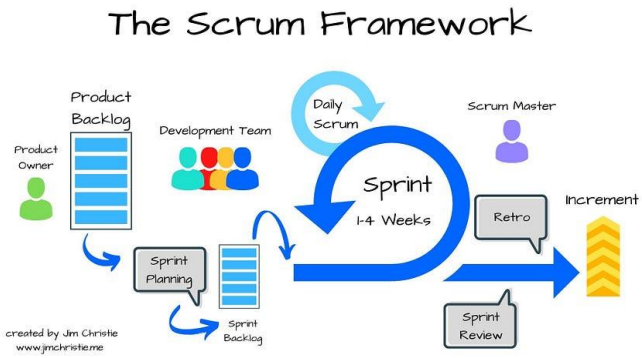

Figure 1 Scrum Framework

Source: https://share.google/images/Vv92JcPmkuQuWGzCL

Agile support continuous learning, progressive refinement and manageable task segmentation as it breaks the development work into smaller iterations called sprints. Each sprint focuses on completing a specific set of features that enables the review of progress frequently, identify issues early and adjust development plan as needed. These advantages of sprint are important for the development of Spendette as this system combines financial tracking, web development and machine learning.

B. Scrum Framework

- Product Backlog: Ordered list of prioritized system features and requirements.
- Sprint Planning: Select backlog items, defines sprint objectives, breaks task into manageable activities and estimate effort.
- Sprint Execution: Design, implement and test selected features including personal and shared expense modules and machine learning integration.
- Sprint Review: Evaluate completed functionalities against sprint goals, demonstrate system performance and gather feedback for backlog updates.
- Sprint Retrospective: Reflect on development process to identify strengths, challenges and area for improvement.
- Product Increment: Each sprint delivers a usable system increment with new functionalities, ensuring continuous progress toward project objectives.

IV. PROPOSED SYSTEM

The requirement analysis and design features of the proposed system are discussed in further detail in this chapter. The system design illustrates how users interact with the system and how system components are structured and connected. The design is represented by using use case diagram, activity diagram, sequence diagram, system architecture and user interface designs to ensure efficient system operation

A. Use Case Diagram

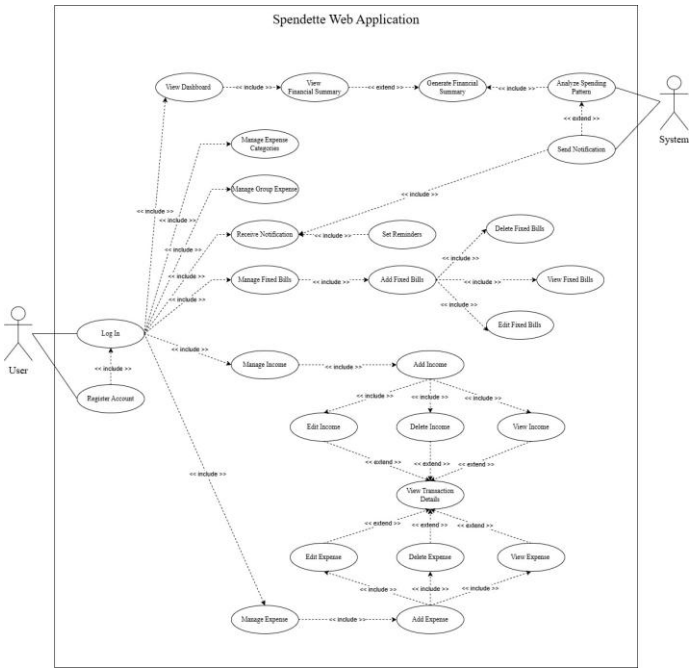

Figure 2 Use Case Diagram

As shown in Figure 2, the use case diagram for the Spendette Web Application provides a high-level overview of the system's functional requirements and illustrates interactions between external actors and the system. It serves as a key tool for understanding system boundaries and feature dependencies.

The primary actors include the User, representing the end-user interacting with the system and the System, which performs automated internal processes. The use cases cover key functionalities such as authentication (Register Account and Log In), financial management (Manage Expense, Manage Income, Manage Fixed Bills, Manage Group Expenses, Manage Expense Categories and View Transaction Details), data visualization (View Dashboard, View Financial Summary and Generate Financial Summary), notifications, and machine learning-based analysis of spending patterns (Analyse Spending Pattern).

The diagram also defines relationships between use cases: «include» indicates mandatory sub-processes executed as part of a base use case, while «extend» represents optional or conditional behaviour that enhances the base use case under specific circumstances.

B. Activity Diagram

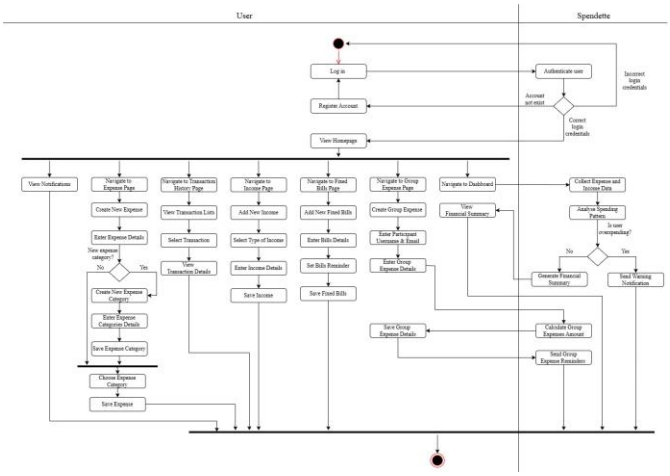

Figure 3 Activity Diagram

As shown in Figure 3, the activity diagram for the Spendette Web Application represents the flow of user interactions and corresponding system responses. It captures both sequential activities and decision-based branching, providing a clear view of the system's operational logic.

From the user perspective, activities begin with logging in or registering, followed by navigating the homepage and viewing notifications. Users can then manage financial data, including expenses, income, fixed bills and group expenses, with tasks such as entering information, selecting categories and saving records. Additionally, users can access the dashboard to view financial summaries and gain insights into their overall financial status.

From the system perspective, Spendette handles authorization, profile management and data collection, while also performing automated analysis of spending patterns. The activity diagram serves as a valuable tool for understanding the workflow, system behaviour and the interaction between user actions and automated processes.

C. Sequence Diagram

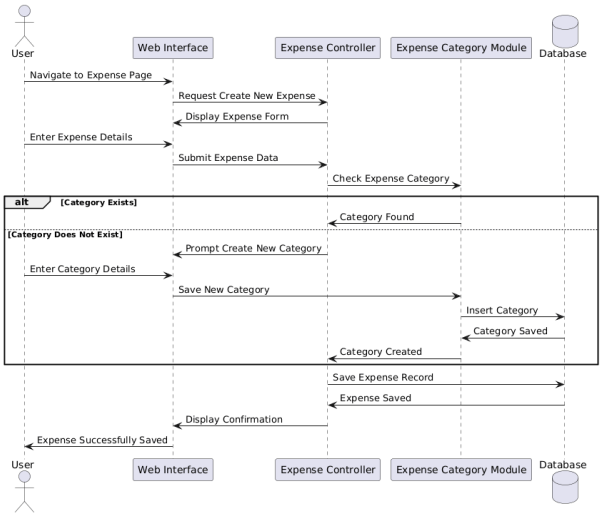

Figure 4 Sequence diagram

As shown in Figure 4, the sequence diagram illustrates the interaction flow for the core feature of the Spendette system, which is managing expenses. The process begins when the user navigates to the expense page and inputs expense details. The system then validates whether the selected expense category exists. If the category does not exist, the user is prompted to create a new category before proceeding. After validation, the expense record is saved in the database and confirmation is displayed to the user. This diagram effectively models the step-by-step interaction between the user and system components for expense management.

D. System Architecture

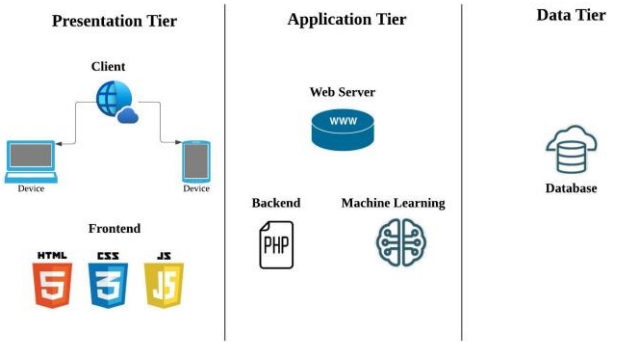

Figure 5 System Architecture

As shown in Figure 5, the system architecture depicts the overall structure of Spendette and the interaction between its components. The proposed system adopts a three-tier architecture consisting of the presentation tier, application tier and data tier. The presentation tier includes client-side components such as the web browser and frontend interface, developed using HTML, CSS and JavaScript. The application tier manages server-side processing through PHP, the web server and the integrated machine learning model. The data tier comprises the database, which stores user information, transaction records and historical datasets. This layered architecture ensures modularity, scalability and efficient interaction among system components.

E. User Interface Design

This section describes the user interface for the core features in Spendette Web Application with an emphasis on usability, consistency and visual clarity. The interface is designed to be user-friendly and intuitive and allow users to navigate the system easily with minimal learning effort. Key design principles such as clear layout structure, appropriate color usage, readable typography and logical placement of components are applied to enhance overall user interaction.

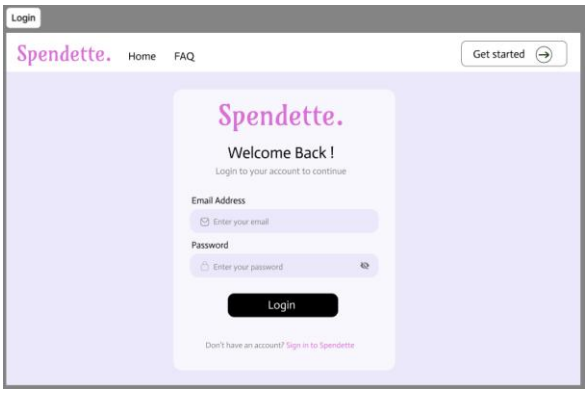

Figure 6 Login Page

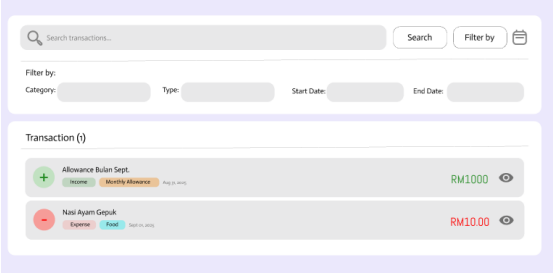

Figure 7 Transaction List

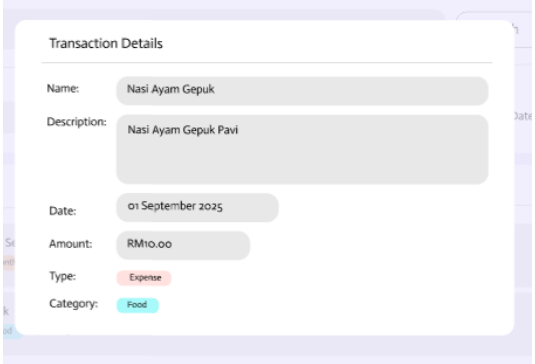

Figure 8 Transaction Details

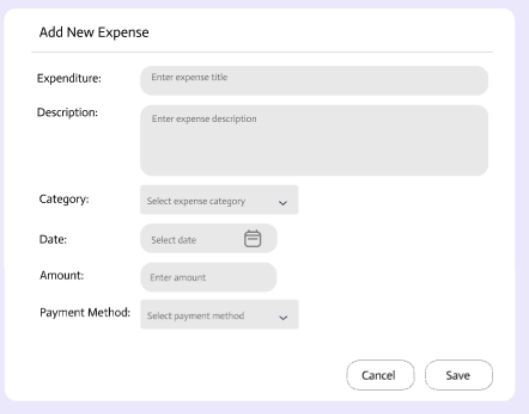

Figure 9 Add New Expense

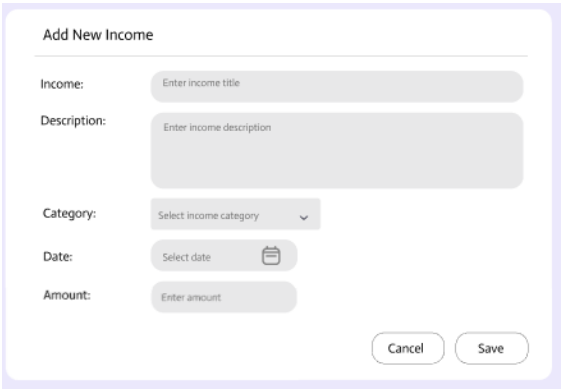

Figure 10 Add New Income

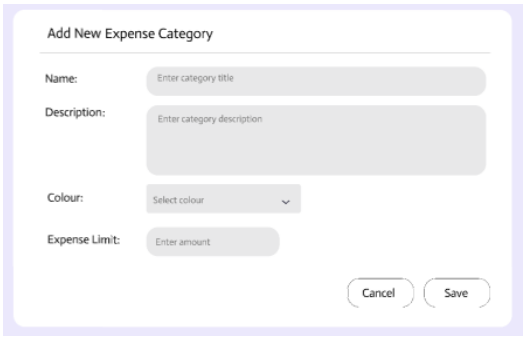

Figure 11 Create Expense Category

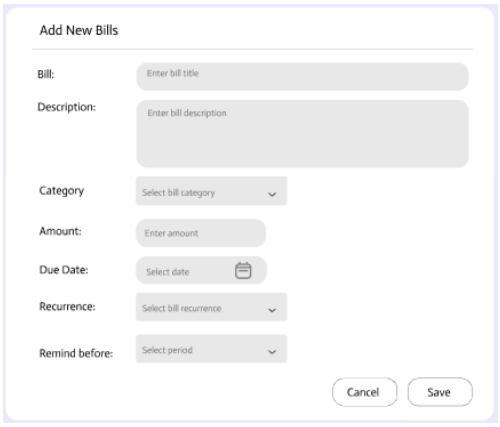

Figure 12 Fixed Bills Feature

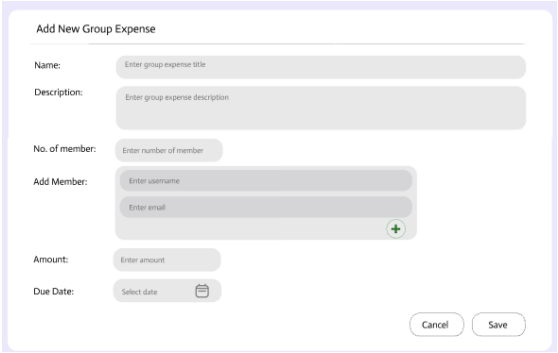

Figure 13 Create Group Expense

F. Functional Requirements

Functional requirements define the specific features and operations that the system must support, including user registration and login, management of personal and group expenses, income, fixed bills, transaction history, financial summaries, spending analysis and notifications or reminders. These requirements ensure that the system meets the core needs of students for managing both personal and shared finances. Table 2 below describes the functional requirement of Spendette.

Table 2 Functional Requirement List

Functional Requirements	Description
User Registration	The system shall allow new users to create an account by providing valid registration details.
User Login	The system shall authenticate users using valid credentials before granting access.
Manage Expense Category	The system shall allow users to create, view, update, and delete expense categories for organizing expenses.
Manage Expense	The system shall allow users to create, view, update, and delete expense records with relevant details and categories.
Manage Income	The system shall allow users to create, view, update, and delete income records.
Manage Fixed Bills	The system shall allow users to create, view, update, and delete fixed bill records and manage bill reminders.
Manage Group Expense	The system shall allow users to create, view, update, and delete group expense records and participant information.
View Transaction History	The system shall allow users to view a consolidated list of income and expense transactions.
View Financial Summary	The system shall generate and display a financial summary based on recorded transactions.
Spending Analysis	The system shall analyze user spending patterns to identify spending behavior.
Notification & Reminder	The system shall send notifications for bill reminders and overspending alerts.

G. *Non-Functional Requirements*

Non-functional requirements define the quality attributes of the system, including usability, security, reliability, scalability, performance and data accuracy. These requirements ensure that Spendette provides secure, user-friendly and efficient experience and maintains data integrity. Table 3 below shows the non-functional requirements for Spendette.

Table 3 Non-Functional Requirement List

Non-Functional Requirements	Description
Usability	The system shall provide a user-friendly and intuitive interface suitable for university students.
Security	The system shall ensure secure authentication and protect user financial data from unauthorized access.
Reliability	The system shall ensure consistent operation and data integrity during normal usage.
Scalability	The system shall be able to handle an increasing number of users and financial records.
Performance	The system shall process user requests and display results within an acceptable response time.
Data Accuracy	The system shall ensure accurate calculation of expenses, income, and financial summaries.

V. CONCLUSION

This project introduced Spendette, a web-based financial management system that helps university students manage personal and shared expenses. The system addresses challenges related to cashless spending and manual expense tracking by providing automated tracking, data visualization, and intelligent analysis. By integrating Machine Learning for budgeting recommendations and overspending detection, Spendette enhances financial awareness and supports responsible financial decision-making.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project supervisor for invaluable guidance, constructive feedback and continuous support throughout the development of this Final Year Project I. I am also thankful to the lecturers and faculty members for their academic guidance and valuable insights. In addition, I would like to thank my friends and peers for their encouragement, support, and helpful suggestions during this project. Lastly, I would like to acknowledge the resources and tools that contributed to the successful completion of this work.

REFERENCES

[1] Cappelli, T., Banks, A. P., & Gardner, B. (2024). Understanding money-management behaviour and its potential determinants among undergraduate students: A scoping review. *PLoS ONE*, *19*(8), e0307137. https://doi.org/10.1371/journal.pone.0307137

[2] Irby, L. (2024). *10 simple ways to manage your money better*. The Balance. https://www.thebalancemoney.com/ways-to-be-better-with-money-960664

[3] Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). SCRUM model for agile methodology. 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 864-869. https://ieeexplore.ieee.org/document/8229928