Team 6 (PennyWise)

1. **Emily Hawkins**
   * **Student ID**: 028199975
   * **Email**: emily.hawkins01@student.csulb.edu
2. **Jonny Olswang**
   * **Student ID**: 030865326
   * **Email**: jonny.olswang01@student.csulb.edu
3. **Kenneth Nguyen**
   * **Student ID**: 027559894
   * **Email**: kenneth.nguyen02@student.csulb.edu
4. **Hong Liang Zhao**
   * **Student ID**: 029524142
   * **Email**: HongLiang.Zhao01@student.csulb.edu
5. **Steven Pham**
   * **Student ID**: 027769779
   * **Email**: Steven.Pham03@student.csulb.edu

## Preface

| Version | Date | Changes |
| --- | --- | --- |
| 1.0 | XX/XX/XXXX | Initial Version |

### Purpose

This document is a comprehensive guide for the development and understanding of the PennyWise personal budget management system software project.

### Audience

The intended audience of this document includes developers, testers, and anyone involved in the project lifecycle.

## Introduction

Effective management of personal finances has become an indispensable part of people's daily lives. Our Expense Management System is designed to help users better grasp their financial situation by simplifying the budgeting and expense tracking process, making personal financial management more accessible and efficient. This system provides a range of features to meet users' different needs for financial management.

First, the system allows users to register an account, which is the basis for managing personal expenses. Once registered, users can create a budget and add new expenses to it, thus accurately tracking their consumption. To help users better understand their spending habits and plan accordingly, the system will send notifications to alert the user when current expenses exceed the previous budget. A key feature of the system is its convenience. It supports integration with other applications (such as Venmo and PayPal) so that expenses on these applications can also be applied to the budget. Overall, PennyWise aims to provide users with a comprehensive and easy-to-use platform for effective financial planning and expense tracking, thereby promoting wiser spending decisions and a healthier financial status.

### Project Overview

PennyWise is a personal budget management system designed for individuals seeking to streamline their financial tracking and planning. It offers tools for budget creation, expense categorization, and budget tracking.

### Project Goals

* Improve the process of tracking expenses.
* Provide consistent updates on changes in spending.
* Provide customization via budget category creation and spending limits.
* Provide connection of accounts with spending applications like Venmo and Paypal.

## Glossary

* **Category**: A label used to group expenditures and incomes.
* **Recurring Expenses**: Expenses that occur regularly, such as monthly bills and subscriptions.
* **Reset Date**: The date the budget cycle ends and a new cycle begins.
* **Priority**: Criteria for determining which budget or expenditures are more important or urgent.
* **Financial Timeline**: specify a time frame for a budget or financial plan.
* **Budget Adjustment**: Changes to the budget based on actual expenditures.
* **API**: Application Programming Interface

## User Requirements and Use Cases

### User Stories

1. As a potential future user, I want to be able to register an account so that I can manage my expenses.
2. As a registered user, I want to be able to add a new expense to a budget that I create so I can accurately track my spending.
3. As a registered user, I want to be able to count the expenditures that I log during a session so that I don’t have to recount to verify if everything has been inputted.
4. As a registered user, I want to be able to receive notifications when my current expenditure rates are different from the previous budget period(s) so I can understand my habits and plan accordingly.
5. As a registered user, I want to be able to set up a monthly budget for different categories so that I can manage my spending and allocate the proper funds to that category.
6. As a registered user, I want to be able to scan a receipt or manually input an expense and have that expense amount be deducted from the budget of that category so that I do not have to calculate the remaining budget.
7. As a registered user, I want to be able to set a budget to be within a specific time period so that I can manage different financial timelines and to prioritize specific budgets and expenses.
8. As a registered user, I want to be able to connect apps such as Venmo and Paypal so that expenses on those apps can be applied to a budget.
9. As a registered user, I want to be able to set a date for when the budget resets so that I do not have to manually reset the budget and I know when I can start spending within that budget.
10. As a registered user, I want to be able to set up recurring expenses such as monthly bills and subscriptions so that I do not have to keep entering them every month.
11. As a registered user, I need to be able to delete or edit budget categories so that I can reflect changes in my spending lifestyle.
12. As a privacy-focused registered user, I need to be able to frequently change my account password so that I can bolster the security of my account.

### Use Case: Adding a New Expense

| Identifier | UC-2 Add new Expense to Budget |
| --- | --- |
| Purpose | Update budget category with new expense and cost |
| Requirements | User Story #2 |
| Development Risks | None |
| Pre-conditions | Registered user is logged in and on add expenses page |
| Post-conditions | New expense is added to the budget category |

***Table 1: Typical Course of Action***

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 | On Add Expenses page, registered user fills in expense information in the form fields. |  |
| 2 | On Add Expenses page, user clicks ‘Add’ on the form. |  |
| 3 |  | Runs checks on information in submitted form |
| 4 |  | Checks are successful |
| 5 |  | New expense is added to the database and budget calculations are updated. |

***Table 2: Alternate Course of Action***

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 | On Add Expenses page, registered user fills in expense information in the form fields. |  |
| 2 | On Add Expenses page, user clicks ‘Add’ on the form. |  |
| 3 |  | Runs checks on information in submitted form |
| 4 |  | Check fails and system alerts the user via an alert box that one or more fields have been left blank. |
| 5 | User clicks the ‘OK’ option on the alert box to close the message. |  |
| 6 | User fills in expense information in the form fields. |  |
| 7 | User clicks ‘Add’. |  |
| 8 |  | Runs checks on information in submitted form |
| 9 |  | Checks are successful |
| 10 |  | New expense is added to the database and budget calculations are updated. |

***Table 3: Exceptional Course of Action***

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 | On Add Expenses page, user selects a category for the expense via a dropdown menu, and inputs a non-numeric value into the amount field. |  |
| 2 | On Add Expenses page, user clicks ‘Add’ on the form. |  |
| 3 |  | Runs checks on information in submitted form |
|  |  | System alerts the user via an alert box that the amount field must only contain numeric values. |
| 4 | User clicks the ‘OK’ option on the alert box to close the message. |  |
| 5 |  | System highlights string in amount field. |
| 6 | User clicks the application home button. |  |
| 7 |  | System displays the application home page. |

## System Architecture

### Components

1. **Frontend**:
2. **Backend**:
3. **Database**:
4. **Authentication**:

### Deployment Diagram