Team 4 (Budget Manager - Pokket)

1. **Marvin Huynh**

* Student ID:
* Email:

1. **Y Nguyen**

* Student ID: 028328532
* Email: y.nguyen-zintgraff01@student.csulb.edu

1. **Nathanael Wolski**

* Student ID:
* Email:

1. **Tom Yorick Pol Sarrazin**

* Student ID:
* Email:

1. **Victor Perez**

* Student ID:
* Email:

Preface

(Describe the purpose of this document, its expected readership and its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.)

| Version | Date | Changes |
| --- | --- | --- |
| 1.0 | 3/14/2024 | Initial Version |
| 1.1 | 15th of Augtember, YYYY | Added Glossary |
| 2.0 | 3/11/24 | Fixed version date notation, … |

Purpose

This document serves as a comprehensive guide for the development and understanding of the software project titled "Pokket"

Audience

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

Introduction

The Pokket Budget Management App is a digital tool designed to help users effectively manage their finances by providing features for budget tracking, expense monitoring, and financial goal setting.

Project Overview

The Pokket Budget Management App aims to empower users to take control of their finances by offering intuitive budgeting tools and insightful financial analytics. It will provide functionalities such as expense categorization, budget planning, goal setting, and transaction tracking.

Project Goals

1. Simplify budget management for users.
2. Enable users to track their expenses effortlessly.
3. Assist users in setting and achieving their financial goals.

Glossary

1. **Budget**: A plan for managing income and expenses over a specific period.
2. **Expense**: Money spent on goods or services.
3. **Transaction**: A record of financial activity, such as purchases or transfers.
4. **Goal**: A desired outcome or achievement that users strive to attain through financial planning.

User Requirements and Use Cases

(Outline what the system must do from the user's perspective. User stories need to use the format discussed in class and on our slides. Use cases provide detailed scenarios of system interactions.)

User Stories

(A collection of user stories that apply to the project.)

1. As a new user, I want to sign up easily so that I can start managing my budget.
2. As an existing user, I want to log in securely to access my budget information.
3. As a user, I want to categorize my expenses accurately to understand my spending habits better.
4. As a user, I want to set monthly budgets for different expense categories to control my spending.
5. As a user, I want to receive notifications when I exceed my budget limits to avoid overspending.
6. As a user, I want to track my progress towards my financial goals to stay motivated.

Use Case: Adding a New Product

| Identifier | UC-2 Add new Product to Inventory |
| --- | --- |
| Purpose | Update inventory with new product name and quantity |
| Requirements | User Story #2 |
| Development Risks | None |
| Pre-conditions | Inventory manager is logged in and on inventory management page |
| Post-conditions | New product is added to inventory |

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

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 | Inventory manager selects “Add new product” |  |
| 2 | Enters product details (name, SKU, quantity) |  |
| 3 | Clicks “save” | Runs plausibility checks on submitted information |
|  |  | Checks are successful |
|  |  | Saves submitted information in inventory database |
|  |  |  |

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

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

System Architecture

(Describe the high-level design of the software.)

Components

1. **Frontend**: Web-based user interface (built with React).
2. **Backend**: RESTful API (built with Node.js and Express).
3. **Database**: MySQL for data storage.
4. **Authentication**:

Deployment Diagram