Team 4 (Budget Manager - Pokket)

1. **Marvin Huynh**

* Student ID: 028075981
* Email: marvin.huynh01@student.csulb.edu

1. **Y Nguyen**

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

1. **Nathanael Wolski**

* Student ID: 025908868
* Email: nathanael.wolski@student.csulb.edu

1. **Tom Yorick Pol Sarrazin**

* Student ID:
* Email:

1. **Victor Perez**

* Student ID: 030571552
* Email: victor.perez01@student.csulb.edu

Preface

| Version | Date | Changes |
| --- | --- | --- |
| 1.0 | 3/14/2024 | Initial Version |
| 1.1 | 3/15/2024 | Added Exceptional Course of Action |
|  |  |  |

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

**User Stories**

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.
7. As a user, I want to allocate money towards specific products
8. As a user, I want to be able to calculate spending or scan products to reduce total money.
9. As a user, I want to implement a budget period that displays the total days left of a budget period.
10. As a user, I want to be able to pay with a third party platform (Paypal or Venmo)

**Use Case: Setting monthly budgets**

| Identifier | UC-1 Setting monthly budgets |
| --- | --- |
| Purpose | Set monthly budgets to control spending |
| Requirements | User Story #2 |
| Development Risks | None |
| Pre-conditions | User is logged in and on expenses page |
| Post-conditions | Values for expenses are updated to user’s desired value |

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

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 | User clicks “edit budgets” button |  |
| 2 | User enters desired budget for each category |  |
| 3 | User clicks “save” button | Runs plausibility checks on submitted information |
| 4 |  | Checks are successful |
| 5 |  | Saves submitted information in expenses database |
|  |  |  |

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

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 | User selects “edit budgets” |  |
| 2 | User selects “back” button |  |
| 3 |  | Changes are not saved and discarded |
| 4 |  | User is returned to default expenses page view |
| 5 |  |  |
| 6 |  |  |

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

| Seq# | Actor’s Action | System’s Response |
| --- | --- | --- |
| 1 | User selects “edit budgets” |  |
| 2 | User enters budget edits |  |
| 3 | User clicks “save” button |  |
| 4 |  | System displays an alert saying the edits were not successfully saved |
| 5 | User closes the alert |  |
|  |  | System displays default expenses view |

System Architecture

Components

1. **Frontend**: Web-based user interface (built with React).
2. **Backend**: RESTful API (built with Node.js and Express).
3. **Database**: MongoDB for data storage.
4. **Authentication**: OAuth 2.0 for user authentication.

Deployment Diagram