

A Prototype Proposal for an Expense Management System

This document contains information relevant to the development of the Expense Management System Prototype proposed by our Software Development Team. It comprises the Project Scope, Project Objectives, Product Backlog, Functional and Non-functional Requirements, and Use Cases.

CMSC 128 - SOFTWARE ENGINEERING
First Semester, A.Y. 2022-2023

Proponents:
Borromeo, Blesse Rachelle
Cimafranca, JV Renz
Lescano, Renmar
Lupac, Coebe Austin

20 October 2022

PROJECT SCOPE

Iteration 1:

- The base system will be deployed as a website application specifically on the Windows operating system. This includes basic features such as adding or editing expenses, calendar, notes, and expenses statistics. Further development will include Linux and Mac operating systems.

Iteration 2:

- Adds a secure login system for users to maintain confidentiality in their personal expenses.

Iteration 3:

- Adds an "ipon" feature in which users are prompted to set the amount for their "ipon" goals in order to save money for future use. The feature includes a puppy, and the software transforms the amount saved into dog food, feeding the dog. The puppy grows into a dog when the user meets the amount objective; otherwise, the dog becomes skinnier and weaker as the user consistently fails to meet the amount target of savings until the dog dies.

Iteration 4:

- Local and online banks will be integrated to keep an account of their transactions.

PROJECT OBJECTIVES




The main objective of this project is to propose a system prototype as a strategy to assist users in budgeting and monitoring their financial expenses or transactions.

Specifically, this project aims:

1. To assist users in keeping records of their current and spent funds.
2. To guide users in making wise budget allocations.
3. To keep track of account transactions on a daily, weekly, monthly, and even yearly basis.
4. To help users in planning their future expenses and avoid being a spendthrift.

PRODUCT BACKLOG

Priority Value: (1 as the Highest Priority)

1.  Must Have
2.  Should Have
3.  Could Have
4.  Won't Have

Story Points Scale:

- | | | |
|--------------------|--------------------|----------------------|
| 1 - takes 1-2 days | 5 - takes 3-5 days | 13 - takes 7-10 days |
| 3 - takes 2-3 days | 8 - takes 5-7 days | |

ITERATION #1

Purchase List

As a User

I want to list out my expenses
So that I can budget them reasonably

Priority:  Must Have
Story Points: 5

Add/Edit Expense Feature

As a User

I want to add/edit my expenses
So that it will be easier to track

Priority:  Must Have
Story Points: 3

Expense Statistics

As a User

I want to see my expense statistics
So that I can see how much I've spent and/or saved

Priority:  Should Have
Story Points: 5

ITERATION #2

Login System

As a User

I want to set up a personal account
So that I can secure the details of my expenses

Priority:  Must Have
Story Points: 3

2-Factor Authentication

As a User

I want to have secured login process
So that I can keep my account secure and private.

Priority:  Must Have
Story Points: 5

ITERATION #3

"IPON" Feature

As a User


I want to have a virtual pet
So that it would be entertaining

Priority:  Could Have
Story Points: 13

Digital Savings Tracker

As a customer

I want to save money
So that I can use it in the future.

Priority:  Should Have
Story Points: 3

ITERATION #4

Digital Banking History Integration

As a Bank Account holder

I want to sync my account to an app
So that I can keep track of my transactions

Priority:  Could Have
Story Points: 8

FUNCTIONAL REQUIREMENTS

The system includes the following functional requirements:

1. Users must be able to browse different website features by clicking a side navigation panel.
2. Users must be able to add, modify, or delete various expense list fields in the system.
3. Users must be able to view expenditure statistics at any time.
4. Users must be able to formulate and set budget goals based on the expenditure statistics.
5. Users must be able to access their accounts by entering their email address and password.

NON-FUNCTIONAL REQUIREMENTS

The system includes the following non-functional requirements:

1. The website must be easy to navigate and functions must be easily-understood.
2. Personal information must be encrypted and can only be accessed by an authorized person.
3. Edit history of expenses can be viewed and the user must get an email confirmation when their bank account is synched.
4. The website must be loaded in not more than three seconds when opened with a fast internet speed.
5. The website must be accessible 24/7 and when the system is down, it must be available for no longer than 24 hours.

USE CASES

Use Case_No.	1
User case	Login
Pre-condition:	User must have an existing active account in the database.
Acceptance Criteria:	User successfully logs in. / User successfully enters the app and accesses the features.
Main Scenario	<ol style="list-style-type: none">1. Open the application2. Proceeds to login page

	<ol style="list-style-type: none"> 3. Enter the username 4. System accepts the username 5. Enter the password 6. System accepts the password 7. Proceeds to log in 8. End.
Extension	<p>If username is wrong:</p> <ol style="list-style-type: none"> 4. a. System detects username is incorrect 3. a. 1. System displays message “Username entered is incorrect” 4. a. 2. Return to Step #3 <p>If password is wrong:</p> <ol style="list-style-type: none"> 6. a. System detects password does not match a username/ incorrect. 5. a. 1. System displays message “Password entered is incorrect” 6. a. 2. Return to step #5 <p>Username not in database / Unregistered user:</p> <ol style="list-style-type: none"> 4. b. System detects that the user is unregistered 4. b. 1. System displays message “Username not found” 4. b. 2. Return to Step #3

Use Case_No.	2
User case	Add/Modify Expense
Pre-condition:	User successfully logs in the system
Acceptance Criteria:	User can edit, modify or delete expense lists in the fields.
Main Scenario	<ol style="list-style-type: none"> 1. Proceed to the “My Expenses” section in the app. 2. System shows the list of expenses 3. System shows “No expenses” if list are empty 4. Adds expenses to the list 5. Edits the desired expenses 6. Deletes the desired expenses 7. System shows the final list 8. End.
Extension	<ol style="list-style-type: none"> 4. a. System detects the user input into the list is blank 4. a. 1. System displays a message “No expenses added” 4. a. 2. Return to #4 5. a. System asks what the user will change (name or the expense amount)

	<p>5. a. 1. System detects user did not pick the value he/she will change</p> <p>5. a. 2. System displays message “Please pick the value you want to change”</p> <p>5. a. 2. Return to #5</p> <p>5. a. 1. System detects user did not pick the expense he/she will delete</p> <p>5. a. 2. System displays message “Please pick the value you want to delete”</p> <p>5. a. 2. Return to #6</p>
--	---

Use Case_No.	3
User case	View Expenditure Statistic
Pre-condition:	User successfully logins in the system
Acceptance Criteria:	User can view different statistics of the expenses
Main Scenario	<ol style="list-style-type: none"> 1. Proceed to the “Track My Expenses” Section 2. System shows the different statistics based on the list 3. End.
Extension	<ol style="list-style-type: none"> 2. a. System displays a sort button if user wants to track expenses ascendingly or descendingly 2. a. 1. System detects user clicking the sort button 2. a. 2. Return to #2

Use Case_No.	4
User case	Access Features
Pre-condition:	User successfully logins in the system
Acceptance Criteria:	User can access all the offered features of the app.
Main Scenario	<ol style="list-style-type: none"> 1. Proceed to “My Expenses” 2. System shows any list of expenses 3. Proceed to “Track My Expenses” 4. System shows statistical data based on the list 5. Proceed to “Settings” 6. Show different options and customizations of the app. 7. Proceed to “My Account” 8. Show different personal account details

	9. End.
Extension	<ul style="list-style-type: none">2. a. System displays a categorize button2. a. 1. System detects user clicking categorize button2. a. 2. Return to #2 4. a. System displays a sort button if user wants to track expenses ascendingly or descendingly4. a. 1. System detects user clicking the sort button4. a. 2. Return to #4 6. a. System detects user clicking “customize the application”6. a. 1. Return to #6