SE 3XA3: Software Requirements Specification
PocketSaver

Team 12,
Mevin Mathew , mathem1 , 400012057
Shalmi Patel , patels19 , 400023762
Diya Mathews , mathewsd , 400014156

October 6, 2017

Contents

List of Tables

List of Figures

N/A

Table 1: **Revision History**

Date	Version	Notes
10/6/2017	1.0	Initial document
Date 2	1.1	Notes

1 Project Drivers

1.1 The Purpose of the Project

The purpose of this project is to help everyday consumers to keep an accurate record on their personal finances and expenses. This in turn, allows for effective money and budget management by the user to save money, and spend smarter.

1.2 The Stakeholders

The stakeholders of this project are anyone who has interest in this project. This includes Professor Bokhari who is the client, the developers, Mevin Mathew, Diya Mathews, and Shalmi Patel, as well as the teaching assistants Jean Ferreira and Christopher McDonald. Other stakeholders include consumers who use the application.

1.2.1 The Client

The client for PocketSaver is Dr. Bokhari as he is the one who requested this project to be completed in the first place for the course of Software Engineering 3XA3. This personal finance application is being created for this course and could be distributed to customers if the developers choose.

1.2.2 The Customers

The target customer for the PocketSaver is a consumer who has either an Android or an iOS device that has the space capacity to download the application onto the device. The personal finance application has a target audience of working class people who earn their own income and spend as average consumers however, it can be used by consumers of all ages.

1.2.3 Other Stakeholders

Christopher McDonald and Jean Ferreira who are both teaching assistants for Dr. Bokhari in the course Software Engineering 3XA3. They help both Dr. Bokhari as well as the developers during the software development process.

1.3 Mandated Constraints

1. An Android Device with APK 23 and iOS version 8.0 or higher

2. Enough memory space on the device to download and install

1.4 Naming Conventions and Terminology

There are no naming conventions or terminology used in this project other than common day to day terminology for spending money.

1.5 Relevant Facts and Assumptions

We are assuming that the user has an Android APK 23 package or iOS 8.0 or greater as well as enough memory on their phone to download and install the app. We also assume that the user is familiar with a menu page navigation style for the application to navigate from page to page.

2 Functional Requirements

2.1 The Scope of the Work and the Product

PocketSaver is based on the open source project CoCoin found on GitLab. With our project, we would like to improve on CoCoin's UI and make it more user-friendly. Much like CoCoin, this application will have an interactive interface allowing the user to enter and edit the amount of money spent. As the expenses accumulate, the application will allow the user to view the amount of money spent that day, month, year, or even longer. Users will be able to see the patterns in their spending habits by comparing their expenses day to day or month to month.

2.1.1 The Context of the Work

The main objective of the application is to keep track of the personal finances of the user. A database is required to keep record of transactions entered by the user. When the user wants to input a new purchase the application has to prompt the user and then take that information and store it in the database. Similarly when requested, the application can extract data from

the database and present the user with detailed information of their previous finances in the form of a graph.

2.1.2 Work Partitioning

This application will be developed by all three members of the team. Since Mevin has the most experience is application development, he will be taking the lead and ensuring that all the backend code is implemented properly. Shalmi will be the leader for the user interface portion of the project. With the help of the rest of the team, she will be paying attention to software qualities such as usability, reliability, and correctness. Diya will ensure that all documentation is kept up-to-date and completed thoroughly and on time. Although there is one leader for each aspect of this project, all members of the team will contribute to the backend, user interface and documentation equally.

2.1.3 Individual Product Use Cases

There are a couple of use case scenarios that we have to consider when developing the application. The bare minimal use cases are as follows.

- Add Transactions

  - Allows the user to enter the amount spent on a transaction
  - The entered amount must be retained for future reference

- Display Trend of Past Transactions (Graphically)

  - Allows the user to see all past transactions visually in the form or a graph
  - The graph should be adjustable to various time periods and show trends in spending habits.

There are a couple of use case scenarios that would make the application user-friendly. These include the following.

- Add transaction on a previous day

  - Allows the user to enter any transactions forgotten from the previous day

- The transactions are able to be entered for any specific date, past or present (not future)

- Edit past transactions

  - Allows user to edit past entries of transactions in the application

- Subtract/delete transactions

  - Allows user to delete past transactions
  - Also allows user to log refunds from previous transactions

2.2   Functional Requirements

1. The application will allow the user to input and delete transactions on a past or current day

2. The application will allow the user to view previous transactions and view the trend of their previous transactions

3. The application will allow the user to edit previous transactions

4. The application will record all inputs by the user on a database

3   Non-functional Requirements

3.1   Look and Feel Requirements

1. The application should have a clean user interface

   - This will make the navigation in the application for users to understand. With an appealing colour scheme, etc.

2. The application should run smoothly

   - The application should run smoothly, with minimal wait/loading times.

3.2 Usability and Humanity Requirements

1. The application must be easy to use and easy to navigate

   - 90% of users should be able to find all functions of the application without help. The layout will be easy to understand, which will allow the user to navigate through the application.

3.3 Performance Requirements

All recent transactions or edits of previous transactions should be reflected on the graph with no delay

3.4 Operational and Environmental Requirements

1. The application should not take up too much storage space on a phone

2. The application works on both Android and iOS

3.5 Maintainability and Support Requirements

If the application crashes, none of the data should be lost

3.6 Security Requirements

1. The application will only store all information collected from the user on the users device

2. The application will not require the user to enter any sensitive information (like credit card information etc.)

3.7 Cultural Requirements

N/A

3.8 Legal Requirements

The application is suitable for all ages, but targets those who are old enough to go shopping on their own

3.9 Health and Safety Requirements

N/A

4 Project Issues

4.1 Open Issues

Issues that could occur with this application, would include UI adaptability. Items such as screen sizes, which change per device therefore causing page scaling issues. This may lead to having current parts of the page to be cut out.

4.2 Off-the-Shelf Solutions

Here are the following off-the-shelf solutions were are going to be using:

1. Android device/emulator for hosting the application

2. GitLab for issue tracking and version control

3. Microsoft Unit Test Framework for unit testing

4. Latex for documentation generation

5. Visual Studio IDE, to develop application

4.3 New Problems

Current Problem we are facing is with the master detail page and creating a navigation system for the application.

4.4 Tasks

The development cycle we are following is the waterfall cycle. The different tasks that need to complete to make this application successful would include, code, documentations, and unit testing. These categories are now broken up into different times, of when each is due, to stay on track. The application will be tested through several testcases, testing all features. Please refer to the Gantt Chart or details on the deadlines, etc.

4.5 Migration to the New Product

The migration to the new product, is mostly on the UI. At the moment, CoCoin has many flaws within the UI. The improvements would include better flow, layout, outputs, and input changes.

4.6 Risks

Risks involving that application, would involve the database. The database can crash, and this could lead to a possibility of losing data. Another issue would be not having access to the database, which leads to losing data that was entered through the application.

4.7 Costs

The Project will have a budget of $0. All the resources that will be used during the project are free of charge.

4.8 User Documentation and Training

This application will not have a form of user documentation. The teams goal is to make this application easy to use, therefore the user will pick up the application flow quickly.

4.9 Waiting Room

After implementing change to CoCoin, other features that we can add is user accounts, where the data can be saved into an account. The accounts would allow to log in and out from varies platforms. Another feature that we can add, is access to the accounts through a web browsers, which will allow the user to see data on a computer.

4.10 Ideas for Solutions

The team should start off with making a page to page layout of how the application is going to flow, thus removing all the clutter and confusion with the navigation system. Then continue to implement it. This will help organize the teams thoughts.

5 Appendix

N/A

5.1 Symbolic Parameters

N/A