Software Requirements Specification

for

Paprika Finance

Version 1.4

Prepared by Andrew Feltham

Team Paprika

April 1, 2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Andrew Feltham	01-04-2018	Added list of unfinished features	1.4
Andrew Feltham	21-02-2018	Added diagrams	1.3
Andrew Feltham	19-02-2018	Additional functional requirements were added	1.2
Andrew Feltham	15-02-2018	Initial Draft	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to build a mobile app that allows users to track, categorize and monitor their financial transactions. This document covers the entire scope of the application.

1.2 Document Conventions

Every requirement shall inherit the priority of the functional requirement unless otherwise state. Bold text shall be used only to designate essential or high priority information.

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers and the client of this application.

The client is the person or company who requested this app and this document will serve as an overview of the application.

The developers will refer to this document in the process of developing it for validation and verification of the project.

1.4 Product Scope

The purpose of this mobile application is to allow users to easily add their financial transactions either manually or by import data from their bank. The mobile application will allow users to track their finances and view statistics with an easy to use interface. The app will also provide recurring bill reminders to the user.

2. Overall Description

2.1 Product Perspective

The mobile application is self contained and has no server side requirements. The app will consist of:

- An encrypted database for storing the financial data
- A simple graphical user interface for accessing and entering the data

- An import and export system to import data from a bank and to export data to other systems.
- A notification system for reminding the user about upcoming bills.

2.2 Product Functions

The mobile app will contain the following features

- Financial transactions and accounts can be entered manually
- The user will be able to categorizing transactions
- Importing and exporting financial information
- The user will be able to set bill reminders
- Show graphs and statistics of the transactions

2.3 User Classes and Characteristics

Users of this product will be non-technical people of any age who want to simply and easily organize and track their financial transactions.

2.4 Operating Environment

The software will run on both iOS and Android operating systems. The application will be able to support many different different phone and tablet configurations.

2.5 Design and Implementation Constraints

As time is limited, the application will be developed with a common code base for both iOS and Android platforms. Ionic has been chosen as a common environment to develop with. In addition, it must have an encrypted database employed in the application for security of the user's information.

2.6 User Documentation

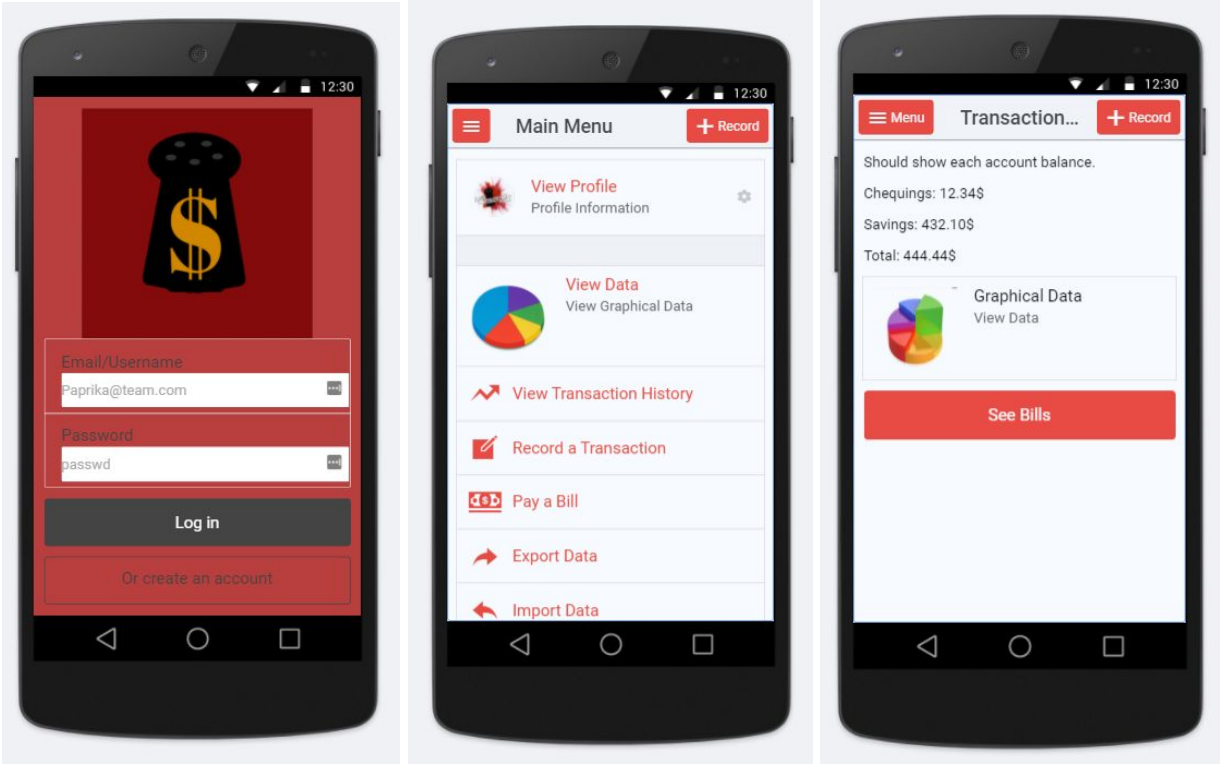
A user manual document will be written and delivered with the application to the client.

3. External Interface Requirements

3.1 User Interfaces

As this is a mobile application the primary interface will be touch screen based GUI following the product family style guides for iOS and Android. Google Material design will be the primary GUI

standards followed during the design. Samples of the protype login screen, main menu, and home screen are shown below as example of the projected design.



3.2 Hardware Interfaces

Supported devices are limited by the Cordova supported operating systems (http://cordova.apache.org):

- Android based phones and tablets running Android version 4.1+
- iOS 8.0+ phones and tablets.

3.3 Software Interfaces

The application will be written using the Ionic framework and will therefore utilize it's software interface to interact with the host operating system. This include the methods to access the local databases and notification systems on the host device.

4. System Features

4.1 Login and Registration

4.1.1 Description and Priority

The user will be able to create an account with a username and password that will allow them access to their encrypted financial information. This is a **HIGH** priority as it is a core function of the app to have an encrypted database.

4.1.2 Stimulus/Response Sequences

- When the user opens the app they are prompted to login or register
- If registering then a screen to create a name and password combination is shown
- Once the user is registered it redirects to login
- The user enters their name and password to open the att.

4.1.3 Functional Requirements

REQ-1: Login screen is shown with name and password input boxes and a link to login or register
REQ-2: Register Screen with a name, password and verify password boxes
REQ-3: Registering a user will create an encrypted database and save the user's name and password hash.

4.2 Transaction List and Statistics

4.2.1 Description and Priority

The user will be able to view their financial transactions with graphs and statistics. This is a **HIGH** priority as it is a core feature.

4.2.2 Stimulus/Response Sequences

- The user can navigate to the summary or statistics part of the app to view the transactions and statistics of their financial data.

4.2.3 Functional Requirements

REQ-1: A screen with summary and statistics exists
REQ-2: The user can navigate to this screen with a menu option
REQ-3: Statistic and graphs are calculated using the information in the transaction database
REQ-4: Types of graphs and content are TBD

4.3 Import and Export

4.3.1 Description and Priority

The user will be able to import CSV files of financial transactions into the app and export their transactions in the app to a CSV file. Priority **MEDIUM**

4.3.2 Stimulus/Response Sequences

- The user can select the import option to browse to a csv file that is imported
- The user can select the export option to export their financial transactions to a csv file that will be saved or shared.

4.3.3 Functional Requirements

REQ-1: A menu option to import csv data
REQ-2: The user can select a csv file that is imported
REQ-3: A menu option to export csv data
REQ-4: The user can export their transaction information to a csv file that they can save or share to their file system or other server.

4.4 Add a transaction

4.4.1 Description and Priority

The user can manually add a transaction with an amount to their list of transactions. Priority is **HIGH**

4.4.2 Stimulus/Response Sequences

- The user has a button to add a new transaction
- A form is displayed that lets the user enter the transaction information including: amount, name, type and category
- The transaction data is saved to the database and added to the list of transactions

4.4.3 Functional Requirements

REQ-1: A menu exists to add a transaction
REQ-2: The user can enter the relevant transaction information
REQ-3: The form is validated so that no incomplete transaction is entered.
REQ-4: The new transaction is displayed in the transaction list.

4.5 Categorize a transaction

4.5.1 Description and Priority

The user can manually categorize a existing or imported transaction. Priority **HIGH** as this is a core feature.

4.5.2 Stimulus/Response Sequences

- The user has a button to edit a transaction in the transaction list.
- A form is displayed that lets the user enter category of the transaction
- The transaction data is saved to the database and updated in the list of transactions

4.5.3 Functional Requirements

REQ-1: A menu exists to edit a transaction
REQ-2: The user can enter the category of the transaction
REQ-3: The form is validated so that no incomplete transaction is entered.
REQ-4: The updated transaction is displayed in the transaction list.

4.6 Set bill reminders

4.6.1 Description and Priority

The user can set one time or recurring bill reminders that notify them to pay bills. This is a MEDIUM priority.

4.6.2 Stimulus/Response Sequences

- The user can select the bill reminders section to view current reminders and change or add now ones.
- A form is displayed that lets the user enter the bill reminder information including name, amount, date and recurring type.
- The bill reminder is saved to the database
- When the bill is coming up to being due a notification is displayed to the user.

4.6.3 Functional Requirements

REQ-1: A user interface exists to list the current bill reminders.
REQ-2: The user can edit or add a new bill reminder.
REQ-3: The bill reminder is saved into the database
REQ-4: Notifications are shown on the mobile device when the bill is coming due.

5. Other Nonfunctional Requirements

5.1 Safety Requirements

As the app is designed to contain the user's financial information the app must be secure and encrypted to prevent this information from being stolen.

5.2 Security Requirements

The app will be required to have an encrypted database to ensure that the user's financial information is secure. This will require a password based or secure alternative authentication method to allow access into the application. The application must automatically log the user out and clear the decrypted date when the user leaves the app.

5.3 Software Quality Attributes

The app is required to be easily used by the desired users. This is the key feature that should be keep in mind throughout the development. As this is designed as a mobile app it must be portable on the major mobile operating systems.

6 Unfinished features

Due to time constraints, health and some poor planning, some required features were not completed.

6.1 List of unfinished features as of the end of the agile sprints

- Import/Export was unable to be completed.
- The profile page was not implemented
- Usability and hint text was not implemented on some screens
- The bill reminders feature was dropped
- Database encryption was not fully implemented due to monetary licensing for the required module

Appendix A: Glossary

app	An application designed to run on mobile devices
csv	A file containing a list of records in a comma separated value format

Appendix B: Analysis Models

Class Diagram