

# Software Requirements Specification

## My Finance Pal

Group: 20/20 Vision

Travis Nguyen, Parker Tuck, Adam Spinhirne, Allan Lieu

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>1.0 Introduction</b>	<b>2</b>
1.1 Purpose	2
1.2 Scope	2
1.3 Overview	2
<b>2.0 Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Characteristics	3
2.4 Constraints	3
2.5 Assumptions and Dependencies	3
<b>3.0 Specific Requirements</b>	<b>3</b>
3.1 External Interface	3
3.1.1 User Interface	3
3.2 Functional Requirements	4
3.2.1 User	4
3.2.1.1 Login / Registration Page	4
3.2.1.2 User Home Page	4
3.2.1.3 Transactions Page	4
3.2.1.4 Settings Page	4
3.2.2 Administrator	5
3.2.2.1 Manage Users' Accounts	5
3.2.2.2 Answer Users' Questions	5
3.2.2.3 Administer Patches	5
3.3 Performance Requirements	5
3.4 Design Constraints	5

# 1.0 Introduction

## 1.1 Purpose

The goal of this project is to design and provide an Android mobile application for anybody that wants to save money and begin budgeting by keeping track of their expenses.

## 1.2 Scope

The “My Finance Pal” is a financial-based mobile Android application that helps people budget their money based on the user’s spending patterns, monthly/annual income, how much the user wants to save, and what categories the user wants to cut spending from. The user will then choose a custom budget plan provided by the app.

## 1.3 Overview

The remainder of the Software Requirements Specification document details the following sections:

- Overall description and details of the mobile Android application (Sections 2.1 - 2.5)
- Specific requirements for the application’s functionality and performance (Sections 3.1 - 3.34)

# 2.0 Overall Description

## 2.1 Product Perspective

All our mobile application’s functions can work without an Internet or satellite connection, but a connection is required to back up the user’s information to the server’s databases. This will allow any user to access their information on other phones or computers that have the app installed.

## 2.2 Product Functions

- Keep track of user cards and accounts
- Keep running total of spending and earning across all accounts
- Select cards for more detailed descriptions of transactions
- Allows user to keep track of expenses by manually inputted transactions
- Provide budgeting help for users by allowing them to specify how much they would like to save over a given period

- Help with budgeting for everyday costs like food by considering projected earnings and spending to see how much money is left

## 2.3 User Characteristics

- Has an Android phone
- Has access to internet
- Has a valid and accessible email address

## 2.4 Constraints

Users could become tired of manually inputting deposits and expenses. The application will only be available on Android phones because the development team does not have access to Apple software.

## 2.5 Assumptions and Dependencies

- We assume that every user's phone has the appropriate hardware to run the app
- We assume that every user has a bank account

# 3.0 Specific Requirements

## 3.1 External Interface

### 3.1.1 User Interface

- Log-in Screen: If the user already has an account, then he/she will be able to enter text into the username and password textboxes and reach the main screen if the information submitted is authentic. At the initial login screen, the user will also be able to create a new account (using an email address and password), and contact customer support.
- Main Screen: Displays the users' financials (overall and monthly net profit/loss, total amount of money) and a list of active cards being used. The main screen will also have button options such as profile settings and add or remove card.
- Profile Settings Screen: Allows the user to edit their account information, add or remove an active card displayed on the main screen, view frequently asked questions, manage alerts or notifications, view contact information, and provide user feedback.

## 3.2 Functional Requirements

There are two types of users that can interact with our mobile application: customers (users) and administrators. Both have their own requirements because they have different uses of the system.

### 3.2.1 User

#### 3.2.1.1 Login / Registration Page

- The user can access the login page when the app is first opened, or from the 'Settings' options.
- The user will login or register using their personal email and password.
- If the login/registration is not successful, then the server will notify the user with an error message. If successful, user will be navigated to the home page.
- When the user is logged in, the 'Login' option will be replaced with a 'Log Out' option.
- The user will be completely logged out when the 'Log Out' button is selected.

#### 3.2.1.2 User Home Page

- The user will be rerouted to the apps home page following a successful login.
- The server will list the user's running total of current money supply, money spent during that month, goals (including the goal's status), and that month's profits.
- The user will be redirected to one of the options listed in the above bullet if that option is selected.
- The user will also be able to navigate to any of the settings page detailed in Section 3.2.1.4.

#### 3.2.1.3 Transactions Page

- The user will be rerouted to this page when the 'Add Transaction' option is selected from the home page.
- The user will enter information regarding monthly income (which can be set to a recurring entry), individual or monthly expenses (such as rent, gas, food, etc.), and can set budgeting goals that can cover any given period of time.
- Goals, recent expenses and profits, and total balance can be seen on the home page.

#### 3.2.1.4 Settings Page

- The user will be rerouted to the 'Settings' page when it is selected from the home or transactions page.
- The user will be able to update their preferences or personal account information to the application's servers. This includes settings such as: what notifications they receive, email account, account password, user's first and last name, etc.

### 3.2.2 Administrator

#### 3.2.2.1 Manage Users' Accounts

- The administrator can access a user's account upon the user's request to either reset their password, permanently delete their account, or manage their account.
- If the user wants to reset their password, then the administrator will reset the password to something random, then email the user back with a link to reset the password and the temporary password to login to their account.
- If the user wants to permanently delete their account or manage some other setting of their account, then the administrator will do so upon the user's request.

#### 3.2.2.2 Answer Users' Questions

- The administrator will be able to answer any and all of the users' questions through the application. The user's question will be directly answered, and other users will be able to see this question and response by navigating to the 'Frequently Asked Questions' tab.

#### 3.2.2.3 Administer Patches

- The administrator can update the application by sending out patches and updates to the applications servers. Version number, patch information, and bug fixes will be included and posted to inform users. They will be posted in the app store (Google Play) and the version number will be posted in the application itself.

## 3.3 Performance Requirements

Any time that the user enters new information or requests to view an element information, we want the load time to be less than three seconds. This goes for functions such as, but not limited to: updating the user's overall balance, loading new screens, refreshing a page, view card transactions, and updating user profile information.

### 3.4 Design Constraints

Since our application will only be available on mobile Android phones, the display screen will be small (compared to the display screen of a computer). That also means that there will have to be a pop-up, touch keyboard that the user will have to use to enter their information. Because the application will only be for Android phones, it will only be found on the Google Play store. The data will be backed up on a private server that can only be manually accessed by a select few employees, and the servers can only be accessed online by administrators with given access. Of these administrators, each will have to change their password every month.