Software Requirements Specification

for

myWallet

Version 1.0 approved

Prepared by Parrish Miller

Jovan Grujic

Ronnie Clark

Team: Memory Overflow

February 26, 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.0 Purpose

myWallet is an app that is intended to provide wide functionality regarding mobile purchases. This is an application that can be deployed to digital devices to allow easy electronic payment for services, the ability to send and request money from other myWallet users, or deposit or transfer money to and from a bank account to myWallet.

1.1 Intended Audience and Reading Suggestions

The intended audience for these documents are the users, specifically users that have the ability to download the app, and meet the app requirements for operation. In the next sections of this document, we will discuss requirements of the system, features of the system, and the overall description of the product. It is recommended that this document is read in chronological order.

1.2 Product Scope

By using myWallet, users will have a mobile wallet at their fingertips. Rather than carrying around physical cash or credit cards, oftentimes it can be easier to pay for services using electronic payment. myWallet will have the ability to load and store debit or credit cards to use in purchases, transfer money to and from a bank checking account to store in myWallet(the wallet balance), and provide a way to send or request money to or from other user's myWallet. myWallet strives to be a robust, secure, and successful mobile payment service by implementing common authentication methods such as Touch ID or Face ID for verification for payments or access into your myWallet account.

2. Overall Description

2.1 Product Perspective

This application is designed to send and receive electronic payments. It is a new, self-contained application that mimics the functionality and takes UI credit of some already existing applications such as Venmo and PayPal. myWallet will be available on the App Store for iOS. This app currently is not intended for use on iPad's, OSWatch, MacOS, Android OS, or Windows. The overall perspective of this app is to allow iOS users the ability to use a mobile wallet electronic payment service.

2.2 Product Functions

Major functionality of the product is listed below:

- Store prepaid currency on app
- Send and receive currency from other users
- Transfer money between bank accounts stored on app
- Make electronic payments

2.3 User Classes and Characteristics

This being a mobile app, there will only be a single user in control of the functionality provided with myWallet. Once a user downloads, installs, and then first opens the app, the user will be prompted to create a myWallet account. This account will be able to correspond to all the actions provided and is unique to the myWallet user. There will be no functionality provided for usage regarding multiple accounts.

2.4 Operating Environment

The environments that this software will be intended for are iOS mobile users. For IOS users, we require running version 13.0 or later. We will also require a steady, and secure internet connection for most major features of the product.

2.5 Design and Implementation Constraints

Due to the inherent constraints with myWallet being a third party application, we are limited to any and all constraints made by the different banking corporations we are interacting with. Also, we are confined to the laws and regulations of handling electronic currency. These include both federal, and local state laws. Security concerns also require a robust authentication system, which can affect accessibility to funds.

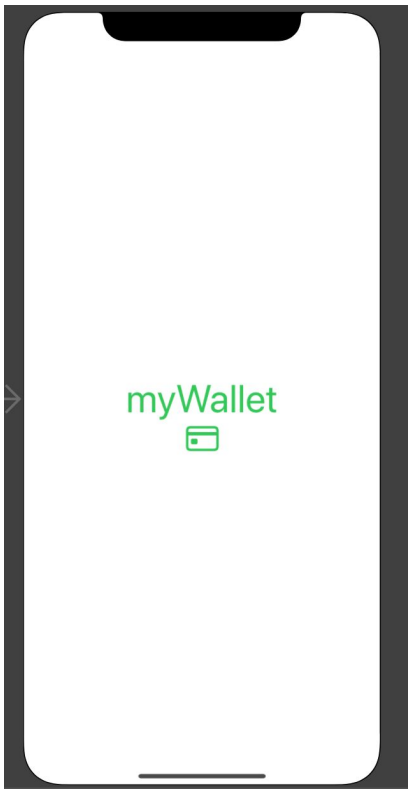
2.6 Assumptions and Dependencies

Assumed factors that could affect the requirements stated in this document include bank compatibility, debit card compatibility, and poor internet connection. Some banks do not allow connections with applications of this kind, and some debit cards do not accept deposits from these apps. If the internet connection is poor, it can seriously impede the application’s capability to communicate with external interfaces. Bank information and card information will need to be verified

3. External Interface Requirements

3.1 User Interfaces

By using MacOS's XCode, we will be following a tabbed style app design including but not limited to: a home screen, pay screen, wallet screen, and settings etc. From there, further UI tab design will be implemented following further design. Shown below is an example mock up launch screen that is first shown when the app is opened. Further UI design is TBA.



3.2 Hardware Interfaces

myWallet will rely on iOS's TouchID and FaceID software to act as a mediary between the phone's ability to authenticate the user thus allowing access into the app and verify purchases. These features can be added using XCode, along with the ability to access the camera to allow QR scanning. which the user will have to allow the app access to do so. All of these rely on the iOS software to allow such permissions.

3.3 Software Interfaces

For this specific project, no external databases are necessary, and the scope of the app is limited to mostly UI design implementation. Therefore, any passwords or database information will have to be stored locally, or may not be used at all. However, the app will reflect the ability that it might simulate a login and authentication process on the front-end and UI design for prototype purposes and this specific ability is TBD.

3.4 Communications Interfaces

myWallet will not communicate with external databases.

4. System Features

Listed below are the features and the requirements that the user shall expect the system to complete. These features are limited to function requirements, as any non-functional requirements are discussed further in the document.

4.1 Creating a myWallet account

4.1.1 Description and Priority

As with any piece of software that requires personal information, this app will require, upon first launch of the app, to either log in or create an account. Without already having an account, the user must create one to begin using the app's features. This is a high priority functionality, considering that the app is useless without this process. Creating an account is only necessary assuming the user doesn't already have one.

4.1.2 Stimulus/Response Sequences

Selecting the "Create an Account" button will allow the user to access a menu style screen to then enter the user's personal information that is unique to the user with provided credentials and authentication. After creating an account, the user will then log into the app using those same unique credentials to that user.

4.1.3 Functional Requirements

REQ-1:User shall be able to create an account.

4.2 Logging into\off the application

4.2.1 Description and Priority

By entering the valid user's credentials, the user will be able to gain access into the app. This is a high priority task considering all further actions desired inside the app will require a login beforehand. Inside the app setting tab, the ability to log out will also be available.

4.2.2 Stimulus/Response Sequences

User types in login credentials including email and password into the given text bars. By pressing "Log In" the user will gain access into the app. This is a common feature for many software applications and the UI design and implementation is minimal here.

4.2.3 Functional Requirements

REQ-2: User shall be able to log into the application.

REQ-3: User shall be able to log out of the application.

4.3 Adding a debit/credit card

4.3.1 Description and Priority

Adding a debit or credit card to the user's myWallet account will allow the user to use it in relevance to completing transactions or sending money to other myWallet user's balances. All of the following payment method processes are considered at medium priority, since the user doesn't necessarily have to complete this task, but the app's functionality mostly operates on these features.

4.3.2 Stimulus/Response Sequences

Specific details regarding how the user will use the UI to add a debit/credit card is TBD.

4.3.3 Functional Requirements

REQ-3: User shall be able to add a debit card to the myWallet account.
REQ-5: User shall be able to add a credit card to the myWallet account.
Note: Adding a debit or credit card will require the basic same process, but some differences between the two e.g. requirements of ZIP code verification might be different for credit cards.

4.4 Adding a bank account

4.4.1 Description and Priority

Along with adding a debit or credit card, myWallet will have the ability to add a bank account for usage inside the app. This is also of medium priority to the user because although you don't have to utilize a bank account for payments or transfers, it is a main functionality of the app's usage.

4.4.2 Stimulus\Response Sequences

Adding a bank account would go hand in hand in a UI perspective regarding adding a debit or credit card. Logically and with ease of use in mind, these two system features are most likely to be combined in the same screen/tab. The specifics on how this UI will operate and look to the user is TBD. All further UI design sequences are TBD until UI design is completed.

4.4.3 Functional Requirements

REQ-6: User shall be able to add a bank account to the user's myWallet account.

4.5 Making online payments

4.5.1 Description and Priority

What's the purpose of an online payment service without the ability to make online payments? This is where the majority of the functionality of the system is flexed, and UI design here will have to reflect the ease of use that will draw user's to using the app. Even though this is the main purpose of the app, this is of medium priority to the user considering they don't have to make online payments to use the app.

4.5.2 Functional Requirements

REQ-7: User shall be able to make online payments using either the myWallet balance, a stored bank account, or a saved debit or credit card.

4.6 Send/Receiving payments

4.6.1 Description and Priority

myWallet will allow users to send payments to other users, and also receive payments from other users. The convenience of this feature is complemented by the intuitive and easy to use UI for sending or receiving payments. While this feature is very convenient, it is not the center of focus for the app, so it will be a medium priority.

4.6.2 Functional Requirements

REQ-8: User shall be able to send and receive payments to other myWallet users.

4.7 Depositing money into bank account

4.7.1 Description and Priority

Users should be able to deposit money from their myWallet account into their own bank accounts. This adds convenience to users that prefer to store their money in their personal bank accounts. It also gives the user the ability to exchange money between multiple bank accounts stored on the app. Because this feature is necessary for the app to be an electronic wallet, it is a high priority.

4.7.2 Functional Requirements

REQ-9: User shall be able to deposit money from the user's myWallet into their storedl bank account.

4.8 Transferring money from a stored bank account

4.8.1 Description and Priority

Along with adding money from the myWallet balance, users will be able to transfer money from a stored bank account to load the account to use in online purchases and sending to other users.

4.8.2 Functional Requirements

REQ-10: User shall be able to transfer money from a stored bank account to load into the user's myWallet balance.

4.9 Storing user balance

4.9.1 Description and Priority

User's myWallet balance will be stored and displayed on the app. We will store the user's balance to ensure no currency is lost or unintentionally added to users. We will also keep track of each users' balance, so that each user shall have the feature of viewing their balance anytime. Because of the importance of keeping track of each user balance, this is a high priority.

4.9.2 Functional Requirements

REQ-11: myWallet shall store all users' balance.

4.10 QR code scanning

4.10.1 Description and Priority

This feature will allow users to send QR codes that link back to the user's account. This QR code can be read by myWallet. This allows users to quickly add other users from their QR code. This feature is convenient but is not as important as other high priority features, because of this it will have a medium priority.

4.10.2 Functional Requirement

REQ-12: Users shall be able to read QR codes that link to other users accounts.

5. Other Nonfunctional Requirements

myWallet has two nonfunctional requirements. All around security, and user friendly UI.

5.1 Security

5.1.1 Description and Priority

Handling currency requires that myWallet is secure or there could be a potential loss of money from security breaches. But currency is not the only thing that needs protection in our system. Sensitive user information must be stored securely as well.

5.1.2 Stimulus/Response Sequences

All of the user information will be encrypted once entered into the system. All transactions and transfers will use SSL to ensure security.

5.1.3 Nonfunctional Requirements

REQ-13: Users data must be encrypted.

5.2 User friendly UI

5.2.1 Description and Priority

For user friendly UI, we expect our system’s UI to be intuitive, and easy to use for users of all technological backgrounds.

5.2.2 Stimulus/Response Sequences

The main screen that will appear upon logging in will give the user an option to either make or request a payment. The user can also use the menu to navigate through the app, and fulfill all of the tasks listed in the functional requirements.

5.2.3 Nonfunctional Requirements

REQ-14: The system UI must be user friendly and consistent throughout.

5.3 User feedback.

5.3.1 Description and Priority

We welcome any questions, comments, or concerns our users might have. We will provide a way for them to voice their thoughts.

5.3.2 Stimulus/Response Sequences

At the bottom of our menu, the user will find a feedback section, where they can choose to either open our application's page on the App Store, or to send us an email directly.

5.3.3 Nonfunctional Requirements

REQ-15: The user must