Software Requirements Specification

for

E-money

Version 1.0 approved

Prepared by Vinokkumar, Sushant and Ramya

Humber College

8th February 2018

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Ramya | 8th feb 2018 | Owing to the discussions carried out on the 5th feb 2018, few things to be added to the SRS. |  |
| Vinokkumar | 9th feb 2018 | Added brief description of application structure |  |
| Sushant | 10th feb 2018 | Added description of database structure |  |

# Introduction

## Purpose

The hardware products which are going to be included in our project are fingerprint sensor and QR code scanner. The software products are firebase and android studios. The whole system will be built upon the basis of these products. This system can be further divided in two subsystems such as hardware and software aspects. The purpose for this project is to decrease the use of cash or credit/debit cards and use the money generated via QR code. The reason for using QR code is that we don’t have to worry about losing cash or even if we forget our wallet at home we can use our cell phone to pay with QR code.

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## References

[1] CanaKit Raspberry Pi 3 Complete Starter Kit - 32 GB Edition. (n.d.). Retrieved September 18, 2017, from https://www.amazon.ca/CanaKit-Raspberry-Complete-Starter-Kit/dp/B01C6Q2GSY . Ordered the fingerprint sensor from adafruit "https://www.amazon.com/gp/product/B019TPP1UK/ref=ox\_sc\_act\_image\_3?smid=A14AIPUXD2PDDS&psc=1" and the qr code scanner from eyoyo "https://www.amazon.ca/Barcode-Scanner-Eyoyo-Handheld-Computer/dp/B01I35GSKE "

[2] Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library [Online]. Available: https://ieeexplore.ieee.org/search/advsearch.jsp

[3] LibGuides: Accessible Computer Science Research Guide: Writing & Citing. (n.d.). Retrieved September 18, 2017, from http://dal.ca.libguides.com/c.php?g=257109&p=1716811

# Overall Description

## Product Perspective

This product is basically a replacement of the existing banking applications and apple pay/android pay. This is built keeping security as the primary key. The application is designed in a way that the user can perform or manage transactions as well as pay their bills all in one. The following schematic gives a clear understanding:

**Case 1 & 2:**

**E-money application**

* Transfer money
* Pay using QR scanner

DB

Generate the code and let the merchandiser scan it

Generate QR code and send the code through an email to the recipient

Payment successful ☺

QR scanner scans the code and deposits the money

**Case 3:**

Account Balance: $3250.75

Credit: $200

Enter Acc.no: 1234567890

Scan

Finger Print sensor

## Product Functions

\* The barcode is used to scan the generated code.

\* Mobile application is used to generate the QR code

## User Classes and Characteristics

Almost 90% of world’s population has their day starting with an android smart phone. This project is intended for people having access to their bank account and smartphone. In this era everything is possible via a smartphone for instance, one can order food, taxi, etc. using various android apps. According to e-money the use may vary from an employer to a merchandiser. This application is user friendly for both and is designed in such a way that both has access to all facilities provided by the app itself. People use online banking since one can manage monthly statements and perform transactions with ease. Whereas the merchandiser has to create a default account so that the payment made to them by the customer gets deposited in it.

## Operating Environment

The environment such as weather, location etc., is not an asset for the e-money. The whole hardware is not legitimately big and hence it is easy to carry around in the parts kit itself. The application is intended for android operating system and hence it is not available for other mobile operating systems. The application collides with the fingerprint sensor to give permissions to access the user’s banking account. The barcode scanner plays a part along with the software application to perform transactions. This is how it is been organized to make the hardware and software work together.

## Design and Implementation Constraints

The database incorporated in our product contains bank account info such as the ten-digit account number, account types like chequing account and savings account. The database will also include information about user login which will have the username as the account/card number, the password field as well as fingerprint for enhanced security. If the user does not have an account then they can register and make an account in order to login. For registering the user will be asked to enter their first name, last name, email id, cell phone number, account number/username, fingerprint and password. All this information will the stored in database and will be needed in order to login.

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

The hardware interfaces needed for connecting our components differ from sensor to sensor. QR code scanner is going to be connected using a USB interface with the raspberry pi. The fingerprint scanner is connected through the TTL to USB convertor with the pi.

## Software Interfaces

The database incorporated is firebase and the operating system used is Android studio (lollipop 22). The app itself used tools from the internet. The services needed are an internet connection, access to email protocol, converting image to bitmap and few basic tools.

<yet to add>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

**Case 1: e-mail Transfer:**

In the user interface there is an option called send. Once send is selected it prompt the user to enter the amount to which we want to transfer. This then generated the QR code with the Acc.no and amount. This is packaged into a unique number. This is used only for transferring.

Now the recipient who receives the email with the unique number will be able to deposit the money if he takes the barcode to the bank. The bank who has the barcode scanner scans the code and deposits the money to the recipient’s account.

**Case 2: Payment:**

In this case the user will have access to generate the QR code for the amount he owes the merchandiser. This is packaged into the unique number (varies from the 1st case). This code is scanned by the merchandiser. Now the amount gets deposited directly to the merchandiser’s default account.

**Case 3: Account Detail:**

Now if the user wants to check the Account information he has to enter his Acc.no and scan the fingerprint associated with the bank account. The fingerprint is basically used for security purposes.

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>