**Software Engineering**

**Software Requirements Specification**

**(SRS) Document**

**QuickScan**

**1/30/2020**

**Version 0.1**

**Rollins College**

|  |
| --- |
| **Revisions** |

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 0.1 | Raul Gomez, Sandro Giannini, Lauren Hancock,  May Phan | Begin initial release | 4/23/2020 |

|  |
| --- |
| **Review & Approval** |

Requirements Document Approval History

| Approving Party | Version Approved | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Requirements Document Review History

| Reviewer | Version Reviewed | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Table of Contents** |

[1. Introduction 3](#_Toc244519333)

1.1 Purpose 3

1.2 Document Conventions…………………………………………………………………….3

1.3 Intended Audience…………………………………………………………………………3

1.4 Scope……………………………………………………………………………………….3

1.5 References………………………………………………………………………………….3

[2. General Description 3](#_Toc244519334)

2.1 Product Perspective………………………………………………………………………...3

2.2 Product Features 4

2.3 User Class Characteristics 5

2.4 Operating Environment 5

2.5 Constraints 6

2.6 Assumptions and Dependencies 6

[3. System Requirements](#_Toc244519335) 7

[4. Non-Functional Requirements](#_Toc244519336) 7

[4.1 User Interfaces](#_Toc244519337) 7

[4.2 Hardware Interfaces](#_Toc244519338) 7

[4.3 Communications Interfaces](#_Toc244519339) 7

[4.4 Software Interfaces](#_Toc244519340) 7

4.5 Performance Requirements 8

4.6 Security Requirements 8

4.7 Software quality attributes 8

1. Introduction

* 1. **Purpose:**

This SRS describes the application functional and non-functional for QuickScan 0.1. QuickScan 0.1 will allow users to scan groceries items in store via barcode and visual recognition

* 1. **Document conventions:**

No documents conventions are being used at this time.

* 1. **Intended audience:**

This document is intended to be used by members of the project team that will implement and verify the correct functioning of the system

* 1. **Project scope**

QuickScan 0.1 is an Android and iOS application that allows users to easily scan items in their shopping cart while shopping, using their mobile phones camera at any groceries store via either barcode or products visual recognition. The application produces a master barcode that will be used to easily check-out.

* 1. **References:** 
     1. <https://www.appcoda.com/barcode-reader-swift/>
     2. <https://apps.apple.com/us/app/xcode/id497799835?mt=12>
     3. <https://www.free-barcode-generator.net/itf-14/>
     4. [How to make your own barcode](https://www.inflowinventory.com/blog/how-to-make-your-own-barcodes/)
     5. [Image Training](https://www.analyticsvidhya.com/blog/2019/01/build-image-classification-model-10-minutes/)
     6. [Python Deep Learning](https://www.youtube.com/watch?v=mB7fdy67eFw) (for predictive features)
     7. [TensorFlowjs](https://github.com/IBM/tfjs-web-app)
     8. [Walmart OpenAPI](https://developer.walmartlabs.com/docs)

## 2. General Description

**2.1 Product perspective:** This project was originated by the idea that scanning items in groceries stores is time consuming and most of the time users will not know how much they spend until they check out. This application will manage their shopping list during the trip, and allows users to check out faster, and see how much they will be paying before going to check out. The system is expected to evolve at least three releases, ultimately includes a predictive feature to recommend users about products they might need based on their shopping trend, support more stores, and push notifications to remind users which items they might need soon based on their inputs.

The data flow diagram (DFD) in Figure 1 illustrates the overall schema and plan for the project.

![A picture containing text

Description automatically generated]()

**Figure 1**

*Data Flow Diagram QuickScan 0.1*

**2.2 Product features:** A high level summary of the functions the software would perform and the features to be included.

FE-1: Data hosting on Amazon Web Services (AWS)

FE-2: Scan products via barcode or visual recognition

FE-3: Print total cost of all items in cart

FE-4: Produce a master barcode on all items in cart

FE-5: Shopping trend learning and storing

FE-6: Integration with email accounts (Gmail)

FE-7: Recommending products

FE-8: Shopping reminder

FE-9: Support Android and iOS

Figure 2 shows the relationship between features of the product and the users

![A close up of a whiteboard

Description automatically generated]()

**Figure 2**

*Major features and how they relate to each other and the users*

* 1. **User class and characteristics:**

|  |  |
| --- | --- |
| Client (favored) | The clients are the person or people who are using the application for their shopping trips. They will use the application on their phones to scan in products to view the master barcode for check out. |
| Business | The business is the entity who will have access to client’s database on shopping trends and products. They can use this information to personalize their advertisement to the client |

* 1. **Operating environment:**

|  |  |
| --- | --- |
| OE-1: | System is not dependent on geographical areas |
| OE-2: | System shall operate in newest versions of both Android (10.0) and iOS (13) |
| OE-3: | Product’s information is provided by Customized Product Database (CPD) |
| OE-4: | Personal data will be stored in the database (AWS), so the database must be secure |
| OE-5: | Android Studio and XCode will be used for User Interface (UI) |
| OE-6: | MySQL will be used to look up item |
| OE-7: | TensorFlow.js and Python will be used for training shopping trend models |

* 1. **Design and Implementation Constraints:**

|  |  |
| --- | --- |
| CO-1: | All data will be stored on AWS |
| CO-2: | Version 0.1 will only be utilizing Customized Product Database |

Figure 3 shows a brief outline of design interactions and actions:

![A close up of a map

Description automatically generated]()

**Figure 3**

*A simple flowchart of how the application shall perform*

* 1. **Assumptions and dependencies:**

|  |  |
| --- | --- |
| AS-1: | No more than 10GB of data stored on AWS |
| AS-2: | User has an active email with which they can log in |
| AS-3: | User has a smartphone with camera |
| AS-4: | Users have at least Android Oreo or iOS 12 on their smartphones |
| DE-1: | Free AWS Account |

## 3. System Requirements

**3.1 Functional requirements**

|  |  |
| --- | --- |
| FR-1: | Operating systems Android and iOS |
| FR-2: | Windows or Macintosh computers |
| FR-3: | Android Studio, XCode and any code editor for TensorFlow and Python |
| FR-4: | Team members all have GitHub account |
| FR-5: | Team members all have either Android phones or iPhone (for testing) |

## 4. Non-Functional Requirements

**4.1 External Interface Requirements**

4.1.1 User Interfaces

|  |  |
| --- | --- |
| UI-1: | Phone application shall permit complete navigation and selection using haptic responses alone, in addition to using stylus or other touch-screen friendly pens |
| UI-2: | System shall ask to allow application permission to use camera for scanning |
| UI-3: | Phone application shall support hardware as well as virtual keyboard input |

*Note: Sample screen layout will be discussed among team members*

4.1.2 Hardware Interfaces

No hardware interfaces have been identified.

**4.2 Communications Interfaces**

|  |  |
| --- | --- |
| CI-1: | The application shall send a notification to the user to remind them to go shopping for products that have run out |
| CI-2: | The application shall send an email message to confirm registration with the system |
| CI-3 | The application shall send a notification to recommend the user about a product that they might need |

**4.3 Software Interfaces**

|  |  |  |
| --- | --- | --- |
| SI-1: | Android Studio | To develop in Android environment |
| SI-2: | XCode | To develop in iOS environment |
| SI-3: | Visual Studio/PyCharm | To training deep learning models |
| SI-4: | AWS | To store shopping trends and users infomation |

**4.4 Performance requirements**

|  |  |
| --- | --- |
| PE-1: | Scanning barcode should take no longer than 4 seconds to find an item over a 40KBps modem connection. |
| PE-2: | Scanning master barcode should take no longer than 4 seconds to register in a register |
| PE-3 | Recommending a product should take user no more than 10 trips to establish a trend |

**4.5 Safety requirements**

No safety requirements have been identified.

**4.6 Security requirements**

|  |  |
| --- | --- |
| SE-1: | Users shall be required to log in to the system for all operations. |
| SE-2: | The application shall permit users who are not owner of the account or developers to view only their own information, not the information of other users. |
| SE-3 | All shopping trends shall be kept secure with AWS and no one but developers and business entity can view |

**4.6 Software quality attributes**

Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.

*Note: Will be discussed among team members*

**4.7 Other requirements**

These may include the legal requirements, resource utilizations, future updates etc.

*Note: Will be discussed among team members*