**Software Requirements Specification**

**for**

**FODO (Food Ordering App.)**

**Version 1.0 approved**

**Prepared by Sayabsi**

*(Sagnik Mitra, Yash Rastogi, Simran Singh, Abhijit Singh)*

**NIIT University**

**30th August, 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** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

*The ‘Food Ordering App’ is an android application designed for restaurants to ease the communication with their customers. The purpose is to remove any ‘middlemen charges’ that are applied by other food applications. Individual restaurants will have their own personalized application for food delivery and/or ordering in-dining.*

## Document Conventions

There are no special conventions in this document.

## Intended Audience and Reading Suggestions

*This document is intended for faculty of Software Engineering Course and the members involved in developing this product.* ***The reader may start reading from section 2.1****.*

## Product Scope

The purpose of this project is for each restaurant to have its own personalized app with multiple uses like: Personalized Loyalty Programs, Delivery management and tracking, Inventory management, comprehensive menu display.

## References

1. Material Design

Google

<https://material.io/design/>

2. Firebase

Google

https://firebase.google.com/

# Overall Description

## Product Perspective

A new, self-contained product leveraging the features of Firebase real-time DBMS and the Android app eco-system, Firebase provides database management for features such as Order History, Menu, Inventory, etc.

## Product Functions

* Menu Display
* Delivery Ordering
* Delivery Tracking
* Loyalty Programs
* Order History
* Bill Payment

## User Classes and Characteristics

Users of the system should be able to view the menu and order for delivery once logged in. The user can also pay the bill in-app when dining in or during delivery. The user can redeem their loyalty points for various benefits.The app will support two types of user privileges staff and customer.

Staff user can (in addition to customer user):

* Cancel order
* Edit Menu
* Change delivery status
* Edit offers
* View all orders

## Operating Environment

* Client/server system
* Operating System: Android
* Database: Firebase NoSQL
* Platform: Java.

## Design and Implementation Constraints

Implementation of loyalty programs and offers, delivery feature is up to the restaurant requirement.

## User Documentation

A basic user documentation will be made available to the project customer containing details about how to use the app.

## Assumptions and Dependencies

*TBD*

# External Interface Requirements

## User Interfaces

*Front-end: Material Design Based Android front-end*

*Back-end: Firebase Database*

## Hardware Interfaces

*Android 4.4 and above.*

## Software Interfaces

|  |  |
| --- | --- |
| *Software used* | *Android Studio 3.2* |
| *Operating system* | *Android 4.4 and above* |
| *Database* | *Firebase by Google* |
| *Java* | *To implement the project we have chosen Java language as its the primary language for development in Android* |

## Communications Interfaces

*The app can run on any Android Phone having Google Apps as it requires Google SSO to authenticate. All database connections are secured and encrypted by HTTPS .*

# System Features

## Sign-In

4.1.1 Description and Priority

*The user will sign in to verify whether it’s a human or a bot. It has a higher priority since every action in this app will depend on this.*

4.1.2 Stimulus/Response Sequences

*Based on the user’s sign-in Info (Name, Phone Number) the order will be placed.*

4.1.3 Functional Requirements

*TBD*

## Ordering Food through the Menu provided

4.1.1 Description and Priority

*The user will select their desired items and place it in order phase.*

4.1.2 Stimulus/Response Sequences

*TBD*

4.1.3 Functional Requirements

*TBD*

## Check availability

4.1.1 Description and Priority

*The user will provide address for delivery to check whether it can be reached by the specific restaurant or not.*

4.1.2 Stimulus/Response Sequences

*TBD*

4.1.3 Functional Requirements

*TBD*

# Other Nonfunctional Requirements

## Performance Requirements

*No such requirements needed since the application is expected to be of smaller size and run on android ver. 4.4 and above.*

## Safety Requirements

*The app is in development process. Currently, there are no safety issues.*

## Security Requirements

*The users may need to verify themselves (google sign-in or any other authentication) if we implement payment option in our product. Otherwise, there is no such requirement in the current phase of our product.*

## Software Quality Attributes

*TBD*

## Business Rules

*TBD*

# Other Requirements

*TBD*

**Appendix A: Glossary**

*ver. - version*

**Appendix B: Analysis Models**

It will be determined based on the development of our product.

**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.>*