**Software Requirements Specification**

Food ordering and Payment App for PES

**0.1**

**27-09-2019**

Team No

|  |  |  |
| --- | --- | --- |
| Sl.No | Name | USN# |
| 1 | Gundam Satyabhama Reddy | 01FB16ECS124 |
| 2 | Ishita Bhandari | 01FB16ECS143 |
| 3 | Harshith Arun Kumar | 01FB16ECS131 |
| 4 | Hari Charan U | 01FB16ECS128 |
| 5 | Iresh Shankarayya Hiremath | 01FB16ECS141 |
| 6 | Kunal Paliwal | 01FB16ECS172 |
| 7 | Hrishikesh S | 01FB16ECS139 |
| 8 | Kevin Paul Kodiyan | 01FB16ECS166 |
| 9 | Harshith H.K. | 01FB17ECS713 |
| 10 | Kusumanjali R | 01FB16ECS173 |

# **Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Approver** | **Comments** |
| 27-09-2019 | 0.1 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

[**Revision History**](#_wqbezb7q76yr) **2**

[**1. Introduction**](#_ox9mkpbt7pkf) **4**

[1.1 Purpose and Intended Audience](#_63kduzhetm9o) 4

[1.2 Scope](#_hoc91rrx6a0k) 4

[1.3 Definitions, Acronyms, and Abbreviations](#_vc2chdooqnhi) 4

[**2. General Description**](#_p6ywhvsfrdiv) **4**

[2.1 Product Perspective](#_ysf78ldl3q94) 4

[2.2 Product Features.](#_nwc829wyrd9b) 5

[2.3 User Characteristics](#_3ufyd8gp0ia6) 6

[2.4 Operating Environment](#_ry5jpjhxw4cc) 7

[2.5 Constraints](#_oqcpm4w2j8xi) 8

[2.6 Assumptions and Dependencies](#_k0qg4jp8zd8h) 8

[2.7 Documentation](#_2h0sltvams7a) 8

[**3. Specific Requirements**](#_tk2i2ckraama) **9**

[3.1 Functional Requirements](#_tnsmo33af80q) 9

[Administrators:](#_73wqvrau4w2x) 10

[3.2 Non-Functional Requirements](#_yinkvltvxjgt) 10

[**4. External Interface Requirements**](#_40ovs734yxua) **12**

[4.1 User Interfaces](#_3j2qqm3) 12

[4.2 Hardware and Software in terms of how they would interact or how they would be executed](#_4i7ojhp) 12

[4.2.1 Software Interfaces](#_2xcytpi) 12

[4.2.3 Communication Interfaces](#_svagpm22lp7f) 12

[**5 Appendix A: Glossary**](#_ucop9e6eqj5) **12**

# **1. Introduction**

## **1.1 Purpose and Intended Audience**

*The purpose of this document is to provide a brief overview of the software requirement specifications for an application system to order food and to make payment within an institution.*

*The intended audience of the app would be the organizations who wish to reduce/manage the crowd near food stalls.*

## **1.2 Scope**

*Food ordering app is a mobile application which will help the members within an institution order food from the canteens of that institution, enabling smooth transactions and lesser crowd near food stalls during break times. This app also allows people to order home food from caterers. The vendor-view of the app also helps them to serve food better and to analyse their sales. The goal of this application is to help students/employees order food in a more systematic and easy way.*

## **1.3 Definitions, Acronyms, and Abbreviations**

*SRS - Software Requirement Specifications.*

*FE - Feature*

*FR - Functional Requirement*

*NF - Non Functional Requirement*

*OE - Operating Environment*

*AS - Assumptions*

*CO - Constraints*

*DE - Dependencies*

*UI - User Interface*

*SI - Software Interface*

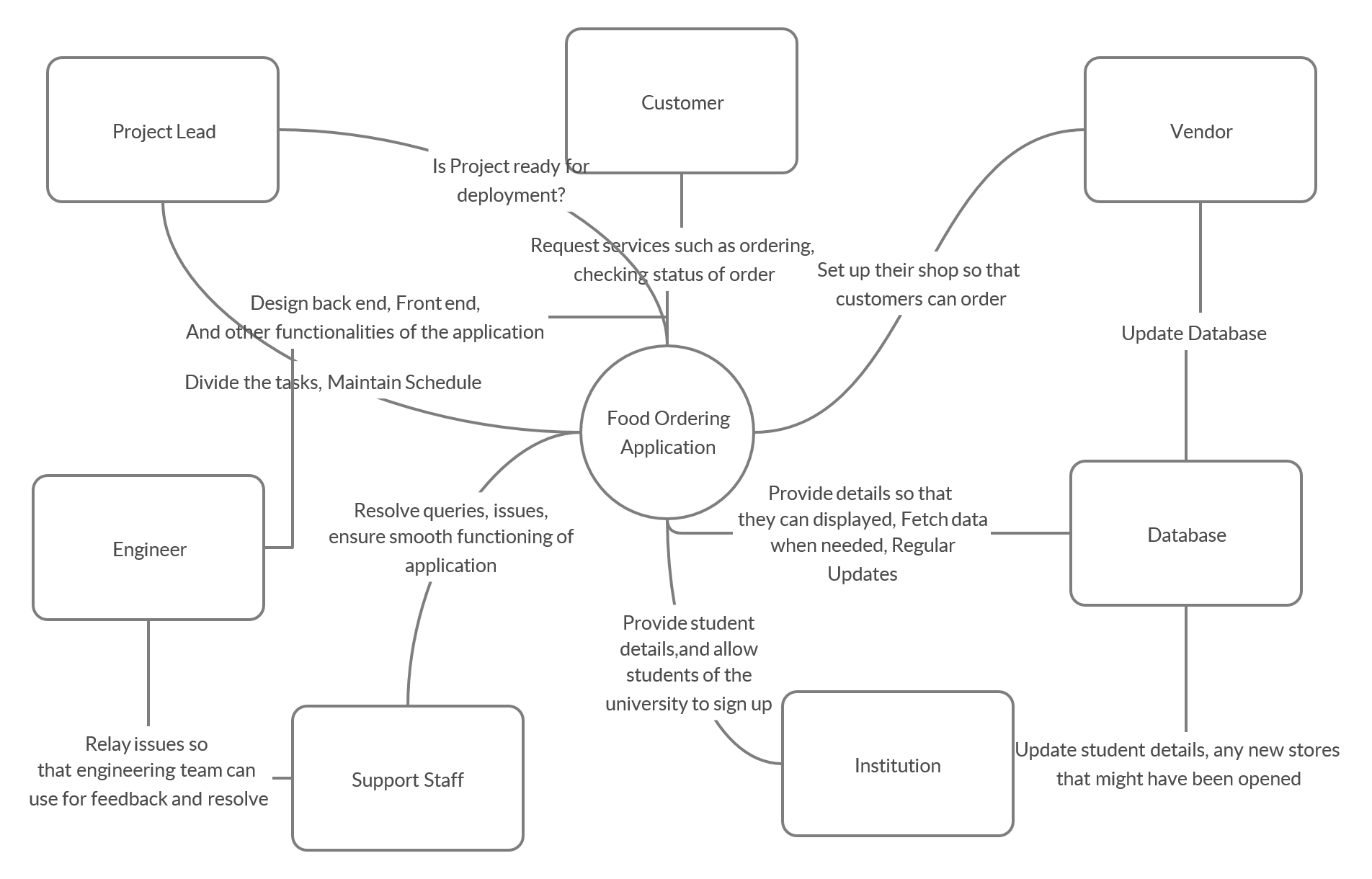
*CI - Communication Interface*

# **2. General Description**

## **2.1 Product Perspective**

*The food ordering app suggested in this document talks about use within an institution. If we were to look at the current market, the existing players would be : Swiggy, Zomato and UberEats. Our app is going to be customised for an institution, and also allow users to make home deliveries. Vendors are given a view as well, thus allowing both sides of the transaction (Customer and Vendor) to use this well thought out application. As suggested earlier, this app could be used to reduce overcrowding at stalls allowing food to be prepared earlier.*

*The food ordering app , on a high level analysis consists of a number of players and interfaces involved such as: The customer, the vendor, the payment mechanism, the institution, database, and the people involved in the development and support of the application: Project Lead, Engineer, Support Staff.*

**

**Figure 1**

*Context diagram for the food ordering application*

*Note: Project Lead, Engineer and Support Staff are all specialized forms of Employee and have access to all Employee actions in addition to their own specialized actions. Database is constantly queried at regular intervals for updates of information.*

## 

## 2**.2 Product Feature**s.

FE-1: Data and application hosting on AWS

FE-2: Register and log on to the application as a customer or a vendor

FE-3: Secure Payments

FE-4: Generate, print and export sale summary

FE-5: Order status with notifications

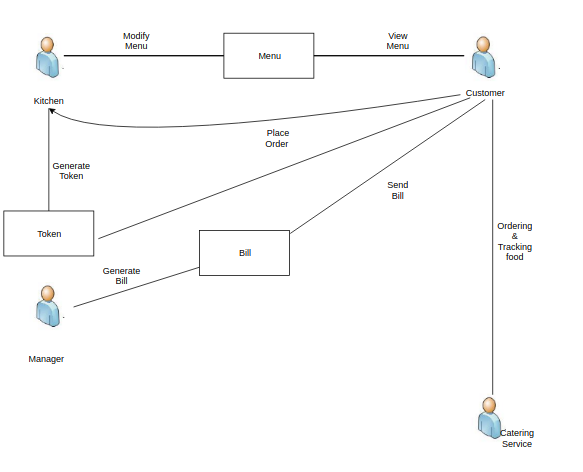
FE-6: Delivery Tracking

FE-7: Customization of menus

FE-8: Brand customization

FE-9: Recommendations for ordering food

FE-10: Provide review for services



**Figure 2**  
*Major features and how they relate.*

## **2.3 User Characteristics**

|  |  |
| --- | --- |
| Customer | The customer is the person or people who will primarily use the app to order food. They use the app to create and maintain a customer profile which stores various details, necessary for ordering food. They will choose from the list of available vendors and order food from their custom menu. The payment for the order can also be made by the customer on the app. |
| Vendor | A vendor is a person or group of people that use the app to host their menu on the app. They will receive orders from customers which they will cater to. |
| Administrator | An administrator manages the day to day operations of the app, making sure customers and vendors have a seamless experience with the app |

## **2.4 Operating Environment**

|  |  |
| --- | --- |
| OE-1: | Since we are using centralized database, System is not dependent on geographical areas. |
| OE-2: | System shall operate in newest versions of all web browsers. And it also operate in Android and Ios operating system |
| OE-3: | There are limited items available, users won’t be allowed to book an item if it is out of stock. |
| OE-4: | Every request is sent to the database , where all the computation takes place and order is placed and the same is sent. |
| OE-5: | User should be able to access all the services. |
| OE-6: | Personal data will be stored in the database, so the database must be secure. |
| OE-7: | Javascript,ReactNative can be used for developing front end. |

## **2.5 Constraints**

|  |  |
| --- | --- |
| CO-1: | All data and the application backend will be hosted on AWS. |
| CO-2: | User interface shall be composed using React native framework and Javascript. |

## **2.6 Assumptions and Dependencies**

|  |  |
| --- | --- |
| AS-1: | Number of orders per day will not exceed the strength of the University the service is initially deployed in. |
| AS-2: | Every student in the University has a phone with an operating system (Android, iOS, Windows) capable of running applications built on React Native . |
| AS-3: | If the users choose to use the web app on the browser instead of the application, they have enabled javascript in the browser. |
| AS-4: | Number of requests/day to the application server do not exceed 1500 approximately (initially). |
| DE-1: | AWS Elastic Compute 2 for the application server. |
| DE-2: | React Native Libraries for the front-end. |

## **2.7 Documentation**

No user documentation information at this time.

# 

# **3. Specific Requirements**

## **3.1 Functional Requirements**

**Vendors:**

|  |  |  |
| --- | --- | --- |
| FR-01 | Login | All users should be able to log on to the system before they are provided access to any functionality. |
| FR-02 | Signup | Any new food stall should be able to create an admin account on our application. |
| FR-03 | View Records | View the records of Products, Orders, Payments, Customers and Sales. |
| FR-04 | Generate Reports | Generate reports of Stores, Customers, Purchasing, Sales, Payment,Products and Orders. |
| FR-05 | Generate Bill | The application should allow the vendors to generate bills for particular orders |

**Customers**

|  |  |  |
| --- | --- | --- |
| FR-06 | Login | All users should be able to log on to the system before they are provided access to any functionality. |
| FR-07 | Signup | Any student/teacher should be able to create an account on the application |
| FR-08 | View Menu | The users should be able to a consolidated menu of all the available canteens and delivery services. |
| FR-09 | View offers | The customer should be able view applicable offers on the available items |
| FR-10 | Add to cart | The customer should be able to add the selected food items in their cart without exception |
| FR-11 | Make Payment | The customer should be able to pay using online methods or cash |
| FR-12 | Track order | The customer should be able to track the order |
| FR-13 | Cancel Order | The customer can cancel the order anytime before delivery/collection. |

## Administrators:

|  |  |  |
| --- | --- | --- |
| FR-14 | Add user | Handle creation of new user |
| FR-15 | Add vendor | Handle creation of new vendor |
| FR-16 | Delete user | Delete an existing user |
| FR-17 | Delete vendor | Delete an existing vendor |
| FR-18 | Manage Institutions | Manage the data for different institutions |

## 

## **3.2 Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Non-Functional Requirement** | **Description** |
| **NF-01** | **Usability** | a) UI made easy for the user to learn on how to use the system and easy to remember.  b) On an average, A normal order from user who knows or is aware of the UI would take 50-60 seconds |
| **NF-02** | **Performance** | a) Performance in terms of response time speed and latency if any.  b) The system is bound to achieve 99% up time and running time of 24\*7.  c) The app UI or the website will have minimalistic response time of less than 3 seconds. |
| **NF-03** | **Capacity** | a) On a busy day upto 80-100 RPM (Requests Per Minute) can be handled by the system real time. |
| **NF-04** | **Reliability and Availability** | a) Over Flow of request will be avoided at all time and be handled well and truly(if any)  b) MTBF – Mean Time Between Failure - 1/168 time units  c) MTTR – Mean Time To Repair  1/0.168 time units  d)Adaptability - The system is to made sure to reach every user across platforms and browsers with varying measure in terms of performance. |
| **NF-05** | **Security** | a) The security standards of Open Web Application Security Project *(*OWASP) are taken into consideration for the system and be followed up.  b) User accepts the information policy compiling to the security measure and user data is to be protected all times. |
| **NF-06** | **Portability** | a) As specified in Availability the system is made to reach every user across different platforms and browsers and there by provides Portability and inter-Operability across platforms too |
| **NF-07** | **Maintainability and Support** | a) System can be Deployed and setup and can be used on the college server within 4-5 working days.  b) Support team can be accessed and is made available with contact mail-id’s and info on the application /website all the time. Customer can reach out any time trouble |

# 4. External Interface Requirements

### **4.1 User Interfaces**

UI-1: Mobile phone app (Android and iOS) shall permit complete user actions using the phone screen. User actions all user interactions required for above feature implementations

### **4.2 Hardware and Software in terms of how they would interact or how they would be executed**

### **4.2.1 Software Interfaces**

SI-1: Database - The system shall communicate with a database through a programmatic interface for the following operations:

SI-1.1: To manage customer, vendor and administrator data.

SI-1.2: To allow a customer to register, login and maintain a customer profile the stores various customer details like name, phone number, address details, payment details, etc

SI-1.3: To allow an administrator to manage customer and vendor information

SI-1.4: To allow a vendor to customise vendor profile and menu and upload various menu items, offers, etc

SI-1.5: To store system metadata

**4.2.2 Hardware Interfaces**

None.

### **4.2.3 Communication Interfaces**

CI-1: The system shall send the customer necessary notifications during various stages of the delivery (order initialisation, confirmation, delivering, delivered, etc).

CI-2: The system shall send an email message to confirm registration with the system for all users.

CI-3: The system shall send a notification to vendors when a new order has been placed and various stages of delivery (see above).

CI-4: The system shall send a notification to the customer when payment is due and after payment is completed. It shall also send an email after payment confirming the payment along with the receipt.

CI-5: The system shall send a notification to the vendors when an issue in an order has been raised by the customer.

# **5 Appendix A: Glossary**

No glossary terms available at this time.