# Software Requirements Specification

**for**

# Digital Food Court Management System

**Prepared by**

Md. Abirul Islam Khan

Arlena Awal

Lameya Rahman

Sadia Ferdous

A.R. Golam Morshed

**American International University- Bangladesh**

**12th December, 2022**

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| 1st draft | 26-10-22 | Initial version draft sent out for comments | 1.0 |
| 2nd draft | 12-12-22 | Added the rest of the document, and fixed typos and other issues in the initial draft | 1.1 |

# 1. Introduction

## 1.1 Purpose

Our project's primary goal is to create a digital food court. Customers can swiftly and conveniently place their food orders. Orders are also received by employers, who then bring them to the chef so that the meal can be prepared fast. Home delivery is an option for customers. Managers are capable of handling their workforce and salaries with tact. One piece of software houses the entire system. People will come to taste it because this kind of system is uncommon in our nation. In terms of our business, it will be better. Last but not least, digitizing this system frees up a ton of time so that individuals can use it.

## 1.2 Document Convention

This document was created based on the IEEE template foe System Requirement Specification Documents. Priority is assigned to requirements to show how they will be implemented. Bold type is used to highlight defined keywords. The normal text font size that has been used in this document is 12pt with 1.5 line spacing, and for all the texts “Times New Roman” font is used, and justified aligned. Defined terms are highlighted with **bolding**.

## 1.3 Intended Audience & Reading Suggestions

This document’s intended audience is mainly the QA engineers, testers and developers of the system. This paper explains the purpose, requirements and appearance of the software. Below is given the list of individuals who contributed to this report. This mailing group will be sent drafts for feedback, changes and input.

| | |
|---|---|
| Lameya Raman | lameyarahman@gmail.com |
| Abirul Islam Khan | abirulislam75@gmail.com |

| | |
|---|---|
| Arlena Awal | arlenaawal13@gmail.com |
| A.R. Golam Morshed | golamorshed001@gmail.com |
| Sadia Ferdous | sadiaferdous@gmail.com |

### 1.4 Scope

Project scope how we are going to get our goal. For our system project scope is below.

- **Search restaurant:** In a food court customers can get many restaurants in one place. Using this application, they can search their preferable restaurant at first.
- **Show menu:** After getting preferable restaurants, customers can see the menu in this application.
- **Order Food:** After choosing the menu, customer can order their preferable food.
- **Payment:** Customer can also make their payment or call an employee to give cash payment in this application.
- **Receiving order:** Employees can also get the order in one application.

## 2. Overall Description

### 2.1 Product Perspective

Digital food court management system is for customers who want an easier food delivery system. Also, the deliveryman and chefs associated to specific restaurants will be benefitted from this software. This software mainly targets food courts that would like to have an integrated system of their own. With this software, food ordering, delivering, meal preparation etc. can be managed and controlled in an effective manner.

It is an online based project. In the near future the application version of it also be introduced. A very active developer team is working on it, to support and sent feedback to users.

## 2.2 Product Functions

**Customer**

- Register
- Login
- Profile
- Settings
- Food delivery
- Dine in
- Cart
- Add to cart
- Remove from cart
- Make payment
- Select restaurant
- Cancel order
- Logout
- Help center

**Deliveryman**

- Receive order
- Deliver order
- Cancel order
- Finish delivery

**Chef**

- Accept order
- Finish meal
- Cancel order

**Admin**

- Add restaurant

- Add employee
- Manage menu
- Remove employee
- Remove restaurant

### 2.3 User Classes and Characteristics

- Typical users: customers who wants to order food for delivery or dine in
- Service users:  delivery man and chefs who are in charge of service
- Internal user: Admin/manager who are able to control and maintain the internal issues of the food court
- Programmers who are working on the project by further developing it and fix existing bugs

## 2.4 Operating Environment

- Windows 7
- Windows 8
- Windows 10
- Mac OS X
- Linux

## 2.5 Design and Implementation Constraints

Digital Food Court Management System will be developed with PHP. As it is an online based project, it will use JavaScript and react for the designing purpose. Angular framework and MySQL database will be used. As for the application version, decisions are still under further discussion.

## 2.6 Assumptions and Dependencies

Digital Food Court Management System is an online based project. There is no extra requirement needed for the user to use this program. A system with internet connection is the minimal requirement. For the application version, things are still under discussion.

# 3. External Interface Requirements

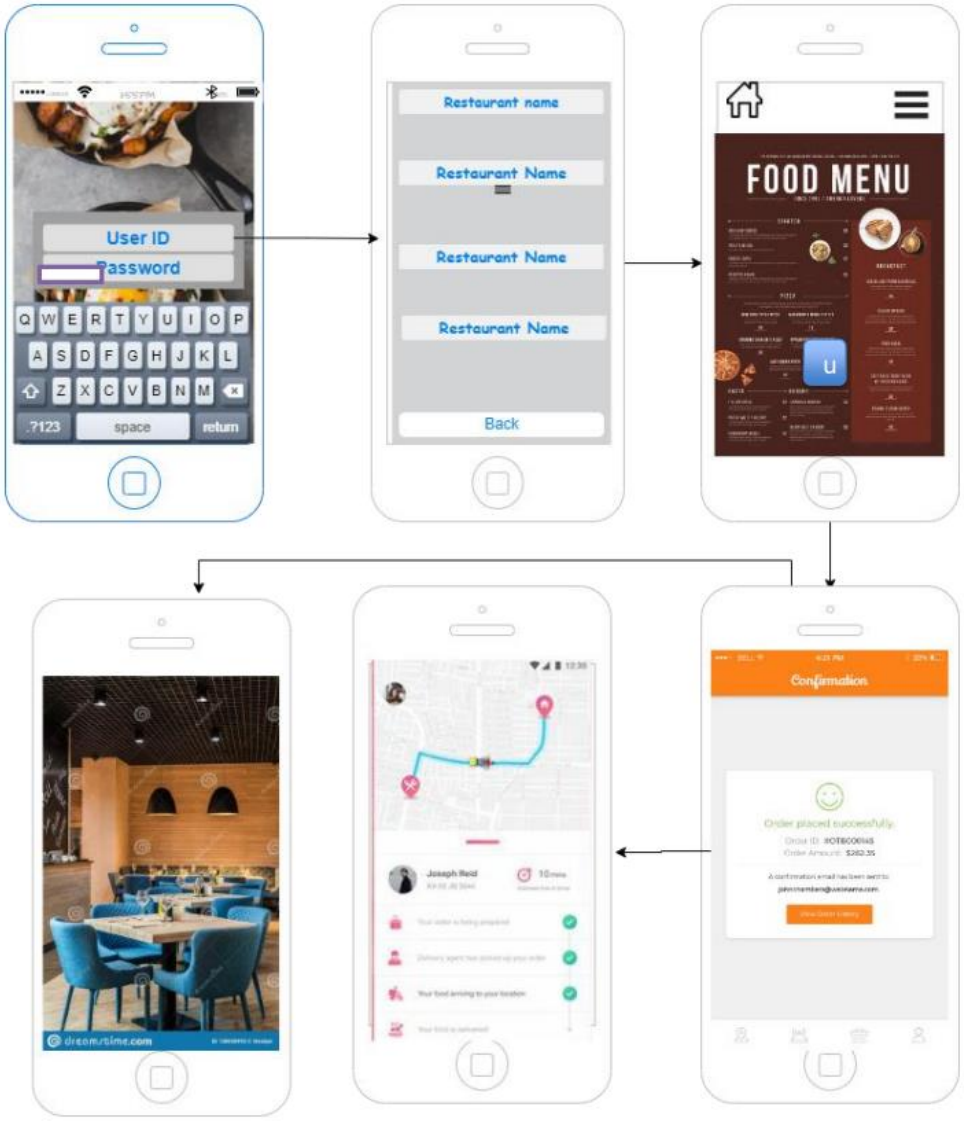
## 3.1 User Interfaces

There are two separate apps for food-shops and customers

However, the app will maintain consistency and follow some standards:

- Every user has the registration page and also have the button to go to login page.
- In login page there are input box for username, password and login button.
- After login there shall be a fixed menu bar at the top with following buttons (contains All, Budget, Deals/discounts/Rating button).
- In homepage it contains restaurants name with list of top-to-bottom format.
- There are fixed drop-down menu pointer at top left with following options (contains Profile, Help, Settings, Logout option for easier access to information).
- On clicking the logo on the top left side, the system return to Home Page.
- There are a “Contact us” and “Logout” buttons at bottom.
- On clicking the “Logout” buttons system session will end and return to “Login” page.

The layout is simple and easy to use for every user. Also, the layout will provide maximum user experience to the user. In case of any errors, it will provide user the proper guidelines to resolve the issue.




## 3.2 Hardware and Software Interfaces

We need the following technology for development of the app:

Camera: for QR scanning

GPS receiver: indicates user location

Moreover, following technology stack we need to used: MySQL database, JavaScript and React for front end, PHP for backend, any framework such as Angular and server such as Apache, Tomcat. A cache such as Memcached can be required to store data. Cloud is used for backup and retrieval of data and security.

### 3.3 Communication Interfaces

The system will use following communication interfaces:

- Emails (Confirmation, cancelation)
- Social media (Promotion)
- Text communication (Feedback, on-demand SMS service)
- Phone number (Emergency communication)

Protocols is required for secure communication and message encryption.

## 4. System Features

### 4.1 Registration and login

#### 4.1.1 Description and Priority

This is the most prioritized feature. This feature allows the customers, deliverymen, chefs and other users to register and then log into the system to order food and receive orders. As there are multiple type of users, there will be separate registration login UIs for each and every user will have access constrains.

#### 4.1.2 Stimulus/Response Sequences

Users will need to provide correct email and a strong password for Registration.

After putting the required information, a confirmation mail will be sent to the user.

After registration, a Login screen will appear.

User will enter the email and correct password to login.

After logging in customer can edit and setup their profile.

### 4.1.3 Functional Requirements

REQ-1: The system must send the user a confirmation mail to complete the registration process.

REQ-2: There must be error messages displayed whenever a wrong email, password or username is given input.

REQ-3: ‘Guest Mode’ feature should be available for temporary access to the system.

REQ-4: Users must be able to edit their profile only after registering to the system.

## 4.2 Search Restaurants

### 4.2.1 Description and Priority

This feature also has a high priority. Customers should be able to search for there desired restaurants in the search bar. After clicking the search button there must be multiple results of the search. Customers are required to keep their GPS on for better search results. There are some suggested keywords given for searching in the search bar.

### 4.2.2 Stimulus/Response Sequences

Customers need to click the search bar to write text or the restaurant name.

After writing their desired restaurant customers need to click the search button.

After clicking there will be a list of results.

Users can choose and click on any of the restaurants and start ordering food.

### 4.2.3 Functional Requirement

REQ-1: Customer needs to keep their GPS on.

REQ-2: The system must record the data to give the customer suggestions in their next search.

REQ-3: The system must have a search history.

## 4.3 Order Food

### 4.3.1 Description and Priority

This feature allows customers to order food from their desired restaurants. After searching and selecting the restaurant customer will be able to view their menu. They will be able to select and add food items in their virtual cart. They can also increase or decrease the quantity of the food items with the options remove from cart and add to cart. After adding customer can process the payment feature.

### 4.3.2 Stimulus/Response Sequences

First the customer searches and selects a restaurant.

Then they can view the menu.

From the menu customers can add food items into their virtual carts.

Customer can also remove food items from cart if needed.

Then customer can click the confirm order button and go to the payment process.

### 4.3.3 Functional Requirement

REQ-1: This system shall have a User Interface of the virtual cart where the customer can add, remove and edit their food items.

REQ-2: User Interface shall have data fields for the following information, like quantity.

REQ-3: User Interface shall have a cancel button to cancel the order.

REQ-4: User Interface shall have a confirm order button to confirm the order and move to payment procedure.

## 4.4 Payment

### 4.4.1 Description and Priority

This feature allows the customer to pay for their order. Customer can pay through three different categories such as COD, mobile banking or Debit/Credit cards., rescue and etc. Customer needs to fill up a form with information like address, phone number etc. and finally make payment.

### 4.4.2 Stimulus/Response Sequences

Customers need to add food items in their cart and then confirm order to proceed to payment.

After confirming order there will be a form to fillup with fullname address and contact for the delivery.

After filling the form up there will be three payment options to choose from.

After choosing the payment procedure customer can make payment and the order process is completed.

### 4.4.3 Functional Requirement

REQ-1: The system must give the live update of the customer's order.

REQ-2: There must be minimum 1 item in the cart to continue the payment process.

# 5.Other Nonfunctional Requirements

A non-functional requirement is for my product what we need and what we do not. Its focus is on the quality of that product. So, it is also known as a quality attribute. For our system, non-functional

requirements in below.

## 5.1 Performance Requirements

Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it. Poor performance leads to negative user experience. It also jeopardizes

system safety when it’s overloaded. For example, the front-page load time must be no more than 2 seconds for users that access the website using an LTE mobile connection. The system should be able to handle 20 million users without performance deterioration.

## 5.2 Safety Requirements

Safety defines how likely it is for the software to work without failure for a given period of time. Safety decreases because of bugs in the code, hardware failures, or problems with other system components. To measure software safety, you can count the percentage of operations that are completed correctly or track the average period of time the system runs before failing. For example, the database update process must roll back all related updates when any update fails.

## 5.3 Security Requirements

Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different users’ roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information. Security also includes protection against viruses and malware attacks. For example, access permissions for the particular system information may only be changed by the system’s data administrator.

## 5.4 Software Quality Attributes:

Software quality attributes can be defined as, the system shall respond to any single customer ordering operation within five seconds. "Respond" means that the appropriate operation output will be displayed to the user within the five second limit. When the order selection is made and confirmed by the customer, the receipt shall be produced within ten seconds and the customer order shall be transmitted to the cook within five seconds. The overall performance of the inventory control system shall be no worse than the average performance of comparable systems currently available in the market place.

## 5.5 Business Rules

Business rules for our food court management are:

• The customer must supply a local telephone number for web orders.

• A valid credit/debit card must be available for web order confirmation.

• The web order must be confirmed by the head waitress via return phone call.

• Preparation of the web order is not started until after the confirmation call.

• The credit/debit card number is held for web orders if the customer chooses to pay with cash or personal check.

• Payment for a web order is not processed until customer pick-up

# 6 Other Requirements

**Database requirements:**

For database of our system a customer must provide name, unique phone number, food order number from the menu. Each customer will have unique refer code from the system after placing order. Chef will have that refer code and handle the order.

**Legal requirements:**

Customer must give information like name, national institute of design number or passport number, mobile number, address. Customer should place order through a valid customer account.

**Data tolerance:**

System can access multiple requests. System can handle with more data or access request at the same moment. No server issues will be occurring.

**Logging and monitoring operations:**

System will check the valid access of any account. System would be able to monitor the given data is relevant or not. System would be able to restrict that particular user for not giving valid request or data input for such short period of time for security purpose.

**Shutdown & recovery:**

Customer can close their account for any valid purpose through system and give feedback for it.

On the other hand, any customer can recover their old accounts with giving the valid information via the system.

# Appendix A: Glossary

**Abbreviations:**

**GPS:** Global Positioning System

**QR Code:** Quick Response Code

**MYSQL:** MY Structured Query Language

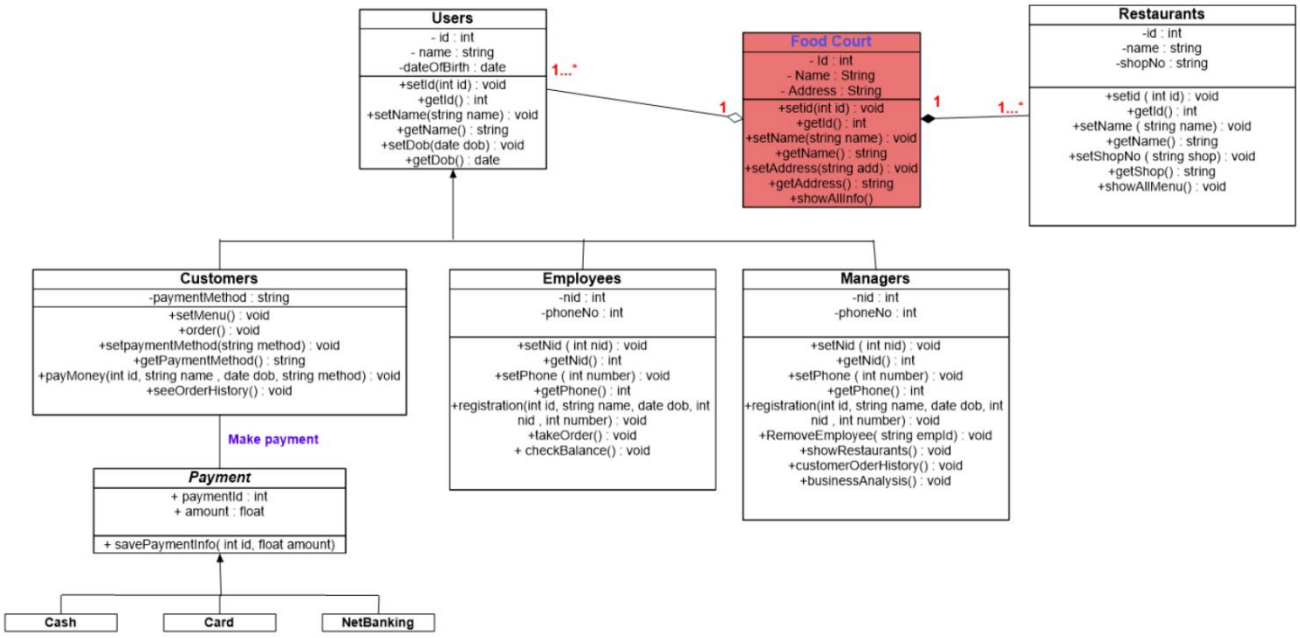
**COD:** Cash on Delivery

**Acronyms:**

**PHP:** Hypertext Preprocessor

# Appendix B: Analysis Models

## ER Diagram:


Users
- id : int
- name : string
-dateOfBirth : date
+setId(int id) : void
+getId() : int
+setName(string name) : void
+getName() : string
+setDob(date dob) : void
+getDob() : date
1...*
1
Food Court
- Id : int
- Name : String
- Address : String
+setid(int id) : void
+getId() : int
+setName(string name) : void
+getName() : string
+setAddress(string add) : void
+getAddress() : string
+showAllInfo()
1
1...*
Restaurants
-id : int
-name : string
-shopNo : string
+setid ( int id) : void
+getId() : int
+setName ( string name) : void
+getName() : string
+setShopNo ( string shop) : void
+getShop() : string
+showAllMenu() : void
Customers
-paymentMethod : string
+setMenu() : void
+order() : void
+setpaymentMethod(string method) : void
+getPaymentMethod() : string
+payMoney(int id, string name , date dob, string method) : void
+seeOrderHistory() : void
Make payment
Employees
-nid : int
-phoneNo : int
+setNid ( int nid) : void
+getNid() : int
+setPhone ( int number) : void
+getPhone() : int
+registration(int id, string name, date dob, int nid , int number) : void
+takeOrder() : void
+ checkBalance() : void
Managers
-nid : int
-phoneNo : int
+setNid ( int nid) : void
+getNid() : int
+setPhone ( int number) : void
+getPhone() : int
+registration(int id, string name, date dob, int nid , int number) : void
+RemoveEmployee( string empId) : void
+showRestaurants() : void
+customerOderHistory() : void
+businessAnalysis() : void
Payment
+ paymentId : int
+ amount : float
+ savePaymentInfo( int id, float amount)
Cash
Card
NetBanking