

Food24

Software Requirement Specification by

**Software Junkies**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **Id** |
| Imran Hossain | 191-15-12722 |
| Md. Ibrahim Khalil Niloy | 191-15-12765 |

TABLE OF CONTENT

1. INTRODUCTION

1.1 PURPOSE-----------------------------------------------------------------------------------------------------03

1.2 DOCUMENT CONVENTIONS-----------------------------------------------------------------------------03

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS----------------------------------------------03

1.4 PROJECT SCOPE---------------------------------------------------------------------------------------------03

1.5 REFERENCES------------------------------------------------------------------------------------------------ 03

2. OVERALL DESCRIPTION-----------------------------------------------------------------------------------------------03

2.1 PRODUCT PERSPECTIVE-----------------------------------------------------------------------------------03

2.2 PRODUCT FEATURES---------------------------------------------------------------------------------------04

2.3 USER CLASS and CHARACTERISTICS--------------------------------------------------------------------05

2.4 USE CASE DIAGRAM---------------------------------------------------------------------------------------06

2.5 OPERATING ENVIRONMENT-----------------------------------------------------------------------------07

2.6 DESIGN and IMPLEMENTATION CONSTRAINTS-----------------------------------------------------07

2.7 ASSUMPTION DEPENDENCIES---------------------------------------------------------------------------07

3. SYSTEM FEATURES

3.1 STIMULUS/RESPONSE SEQUENCES

3.2 FUNCTIONAL REQUIREMENTS

3.3 CLIENT/SERVER SYSTEM

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

4.2 HARDWARE INTERFACES

4.3 SOFTWARE INTERFACES

4.4 COMMUNICATION INTERFACES

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

5.2 SAFETY REQUIREMENTS

5.3 SECURITY REQUIREMENTS

5.4 SOFTWARE QUALITY ATTRIBUTES

1. Introduction

1.1 Purpose

The purpose of this document is to build an online home-made food delivery system in which people can enjoy food as they eat at their own home.

1.2 Document Conventions

This document uses the following conventions:

|  |  |
| --- | --- |
| ER | Entity Relationship |
| Info. | Information |
| No. | Number |

1.3 Intended Audience and Suggestions

This document is a prototype of Online Home-made Food Order and Delivery System. This project will be implemented by team Software Junkies under the guidance of Software project Sir, Md. Sohid Ullah. This project will be very useful for busy hungry people as well as freelance cooker.

1.4 Project Scope

The purpose of this online home-made food order and delivery system is to solve meal problem of busy people. It's a great solution for bachelors who are facing meal problem. Our customers can enjoy home-made healthy food as they cook at their own house. This will be a great source for income money for our freelance cookers and our delivery mans. Our freelance cooker can easily earn money. They just need to cook some extra food while making for their own house.

1.5 Reference

2. Overall Description

2.1 Product Perspective

Our product will provide the following information:

* Food Details:

It includes every food item serial number, item name, price, items quantity and a good picture of that food item.

* Customer description:

Our product will store every customer detail in its database. Every customer must have a unique user name and password. First of all, he has to sign up with his email or phone number. We would also record his address, profession and other necessary information.

* Cooker description:

Every cooker would have a unique user name. He/she must have to register with his/her phone number or email and any identifiable document like NID and passport. He\she also has to provide his name, address, profession (if any) etc. His/her location will be hidden from customers.

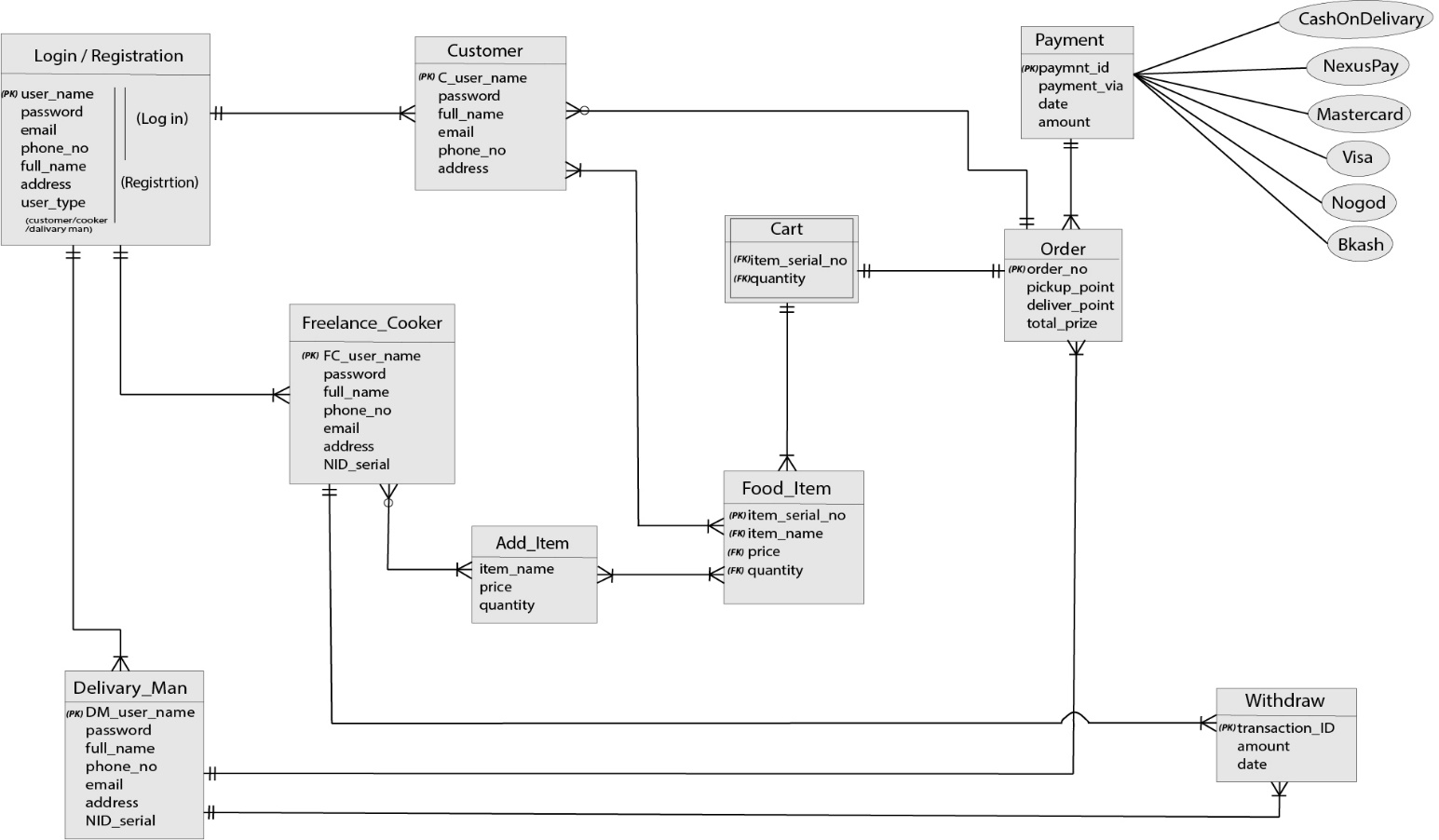
* Delivery man description:

Like cookers delivery mans has to register with his identification document (NID, passport), driving license (if bike) and phone number or email address. We will record his previous experience and other basic information like name, address, date of birth, other profession (if any).

* Availability Nearby:

At the opening of this app it will show a map. The map will provide all the foods available nearby and the user's own location. Of course, the user won't see the exact location of the foods (basically cooker) for our business policy.

2.2 Product Features



2.3 User Class and Characteristics

A cooker will cook his\her food and post it in this app with clear picture. When a customer will open the app, he will see all the foods available near him. He won’t see the exact location. When he will order any item, a pickup request would be sent to delivery man nearby the food location. Delivery man will receive the food and deliver to the customer. The delivery man will take the payment if customer will choose cash on delivery option as payment. All the process will be monitored by admins (automatically or manually).

So, this system has four users (Customer, Cooker, Delivery man and Admin). Customers has the access to customer functions, Cookers has the access to cooker functions, delivery man has access to deliver functions and admin has all those functions including System management functions.

Customers should be able to do the following functions:

* View item
* Order item
* Review item or cooker
* Search
* Customize own information

Cookers should be able to do the following functions:

* Add item
* View profile rate\reviews by others
* Customize own profile
* Withdraw money

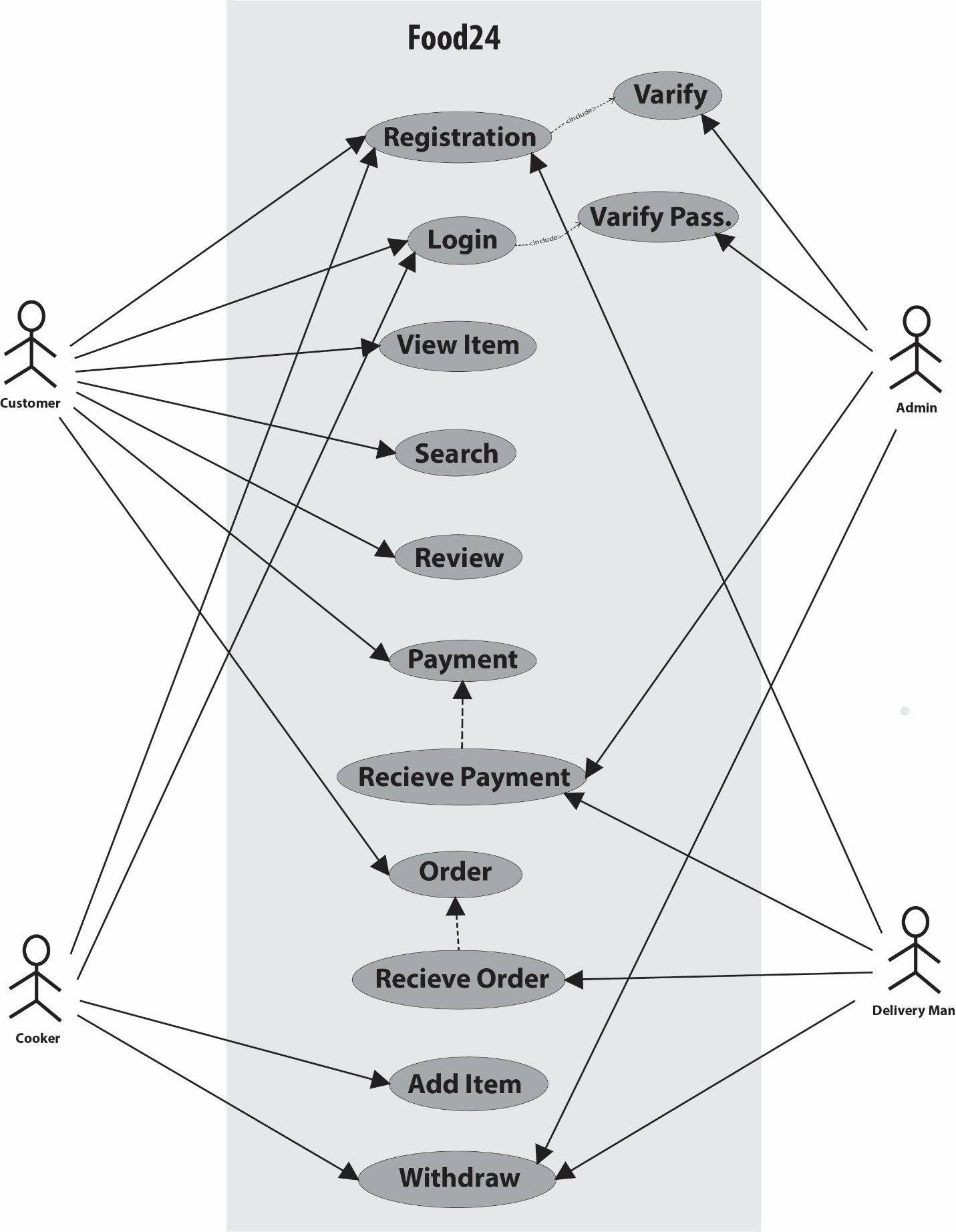
Delivery man should be able to do the following functions:

* Receive delivery request
* Confirm after delivery
* Customize own information

Admin should be able to do the following functions:

* Approve registrations of Cookers and Delivery mans
* Identify information given by users (NID, passport, payment info, driving license)
* Categorize users
* Receive and process online payment
* Monitor delivery system
* Check food (cookers post) quality

Use case diagram



Each item will be available for a limited time to serve fresh foods. Delivery time will be as minimum as possible.

2.4 OPERATING ENVIRONMENT

Operating environment for the airline management system is as listed below. Include

the details as per your application

* Database
* Client/server system
* Operating system: Windows.
* Database: SQL+ database
* Platform: vb.net/Java/PHP

2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

* SQL commands for queries/applications
* How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
* Implement the database at least using a centralized database management system

3. SYSTEM FEATURES

3.1 STIMULUS/RESPONSE SEQUENCES

* Search for Airline Flights for two Travel cities
* Displays a detailed list of available flights and make a “Reservation” or Book a ticket on a particular flight.
* Cancel an existing Reservation.

3.2 FUNCTIONAL REQUIREMENTS

Other system features include:

* DATABASE
* CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of

responsibilities, the client is the application (also known as the front-end), and

the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites

3.3 CLIENT/SERVER SYSTEM

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

* Front-end software: Vb.net version
* Back-end software: SQL+

4.2 HARDWARE INTERFACES

* Windows
* A browser which supports CGI, HTML &amp; JavaScript

4.3 SOFTWARE INTERFACES

Following are the software used for the flight management online application

|  |  |
| --- | --- |
| Software Used | Description |
| Operating System | We have chosen Windows operating  system for its best support and user-  friendliness. |
| Database | To save the flight records, passengers  records we have chosen SQL+ database. |
| VB .Net | To implement the project, we have  chosen Vb.Net language for its more  interactive support. |

4.4 COMMUNICATION INTERFACES

This project supports all types of web browsers. We are using simple electronic forms

for the reservation forms, ticket booking etc.

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of airline database are as listed

below.

* ER DIAGRAM:

The E-R Diagram constitutes a technique for representing the logical structure of

a database in a pictorial manner. This analysis is then used to organize data as a

relation, normalizing relation and finally obtaining a relation database.

* NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that

information is to be stored only once. Storing information several times leads to

wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a

database table. Similarly, in traditional databases as well as improperly designed

relational databases, data redundancy can be a problem. These can be

eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So

that each table deals with a single theme. There are three different kinds of

modifications of anomalies and formulated the first, second and third normal

forms (3NF) is considered sufficient for most practical purposes. It should be

considered only after a thorough analysis and complete understanding of its

implications.

5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic

failure, such as a disk crash, the recovery method restores a past copy of the database

that was backed up to archival storage (typically tape) and reconstructs a more current

state by reapplying or redoing the operations of committed transactions from the

backed up log, up to the time of failure.

5.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the

special requirements of the security market mean that vendors must choose their

database partner carefully.

5.4 SOFTWARE QUALITY ATTRIBUTES

* AVAILABILITY: The flight should be available on the specified date and specified

time as many customers are doing advance reservations.

* CORRECTNESS: The flight should reach start from correct start terminal and

should reach the correct destination.

* MAINTAINABILITY: The administrators and flight in chargers should maintain

correct schedules of flights.

* USABILITY: The flight schedules should satisfy a maximum number of customers’

needs.