# SOFTWARE REQUIREMENT SPECIFICATION

# Yumzy

# INTRODUCTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out **functional** and **non-functional** requirements, and may include a set of use cases that describe user interactions that the software must provide.

# Purpose

# This document presents a detailed explanation of the objectives, features, user interface and application of Yumzy in real life. It will also describe how the system will perform and under which it must operate. In this document it will also be shown the user interface. Both the stakeholders, users and the developers of the system can benefit from this document.

# Intended Audience and Document Overview

# This document is intended for different types of readers such as customers, stakeholder, system designer, system developer and tester. By reading this document a reader can learn about what the project is implemented for and how it will present its basic ideas. This document has a sequential overview of the whole project so if a reader reads the document from top to bottom, he will get a clear idea about the project.

# Process Involved:

The following use cases of the “Yumzy” conceptualization are in scope.

* Unregistered Users also Can view (You can click skip option and view the restaurant)
* Register to Application
* Food item Available- various types of indian cuisines, american cuisines and coffee
* Search for food items easily
* Category of food item- Rajasthani, Gujarati, Maharashtrian, Punjabi, South Indian, Bengali, American, Coffee
* Cart feature
* Date and time of food item delivery will be notified by the system
* The admin can add/delete Suppliers and delivery boys.
* Allows the customers to maintain a cart.
  1. **Existing System**

The current system for shopping is to visit the Restaurant manually and from the available list of food items, choose the item the customer wants and buying the item by making a payment of the price of the item.

* + - It is less user-friendly.
    - User must go to Restaurant and select food item.
    - It is difficult to identify the required food items.
    - Description of the food item is limited.
    - It is a time-consuming process
    - Not in reach of distant users.
  1. **New System**

In the proposed system customer need not go to the Restaurant for buying the food item. He can order the food item he wish to buy through the website or portal. The Restaurant owner will be admin of the system. Restaurant owner can appoint suppliers who will help owner in managing the customers and food item orders. The system also recommends a home delivery system for the purchased food item.

1. **OVERALL DESCRIPTION**
   1. **Product Perspective:**

This software will help the user to connect to the restaurant, through this system a shop owner can reach other areas, helps to make an order for those who are unable to go to the restaurant, it keeps record and payment details so, owner can make good decisions, and provides a great variety to the user.

* 1. **Product Scope:**

This system can be implemented to any Restaurant in the locality or to multinational branded Restaurants having retail outlet chains. The system recommends a facility to accept the orders 24X7 and a home delivery system which can make customers happy. If Restaurants are providing an online portal where their customers can enjoy easy Restaurant from anywhere, the Restaurants wont be losing any more customers to the trending online Restaurants such as Flipcart or ebay. Since the application is available and always available.

* 1. **User Classes and characteristics:**

There are four types of actors and one cooperating system Characteristics. There are several users of this system:

**ADMIN/ADMINISTRATOR:**

Who can make changes in interface of application. He is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the Restaurant. The administrator has all the information about the users and about all food item.

**CUSTOMER:**

With no any special training to operate the software

**SUPPLIER/RESTAURANT:**

Will get order from user and prepare food

**DELIVERY BOY:**

Gets information and takes food from the restaurant and delivers to user door

* 1. **Use Case Diagrams**

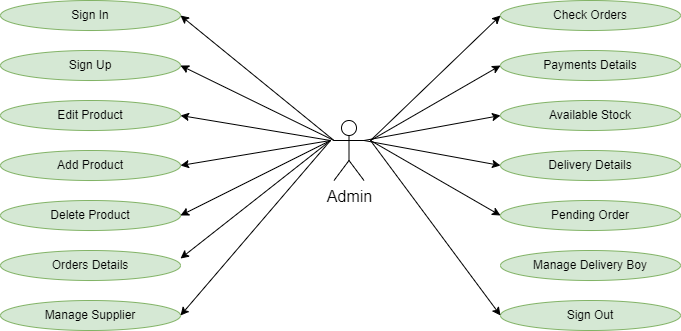
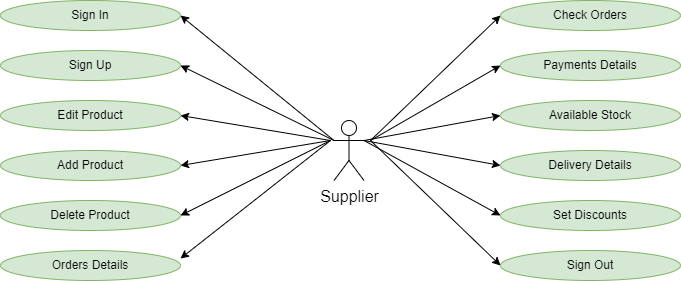
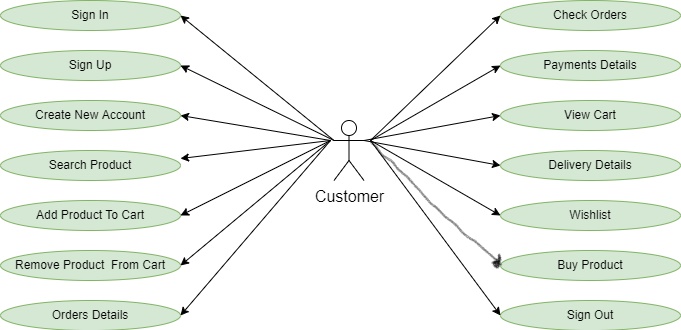


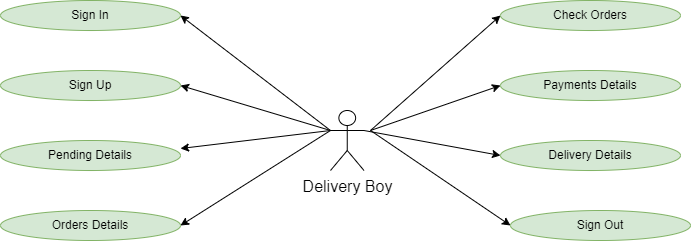
Figure 1: Admin Use Case Diagram



**Figure 2: Supplier Use Case Diagram**



**Figure 3: Customer Use Case Diagram**



**Figure 4: Customer Use Case Diagram**

# SPECIFIC REQUIREMENTS

# Functional Requirements

This subsection presents the identified functional requirements for the subject Yumzy. Initially, general requirements that pertain to the whole system are given. Where possible, subsequent requirements have been demarcated based on their relevance to the users of the system, that is, Customers, Restaurants, Admins and Delivery boys.

**3.1.1 General:**

The following are the identified functional **general** requirements that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| **Requirement** | Description |
| G1 | A server shall host the Yumzy App and provide system data processing and storage capability. |
| G2 | A surface app page shall provide a customer with all customer system functionality. |
| G3 | The App shall provide a User/Restaurant with all user/restaurant system functionality (according to access control). |
| G4 | A display shall provide a Delivery Boy with all Delivery boy system functionality. |
| G5 | The App shall be capable of interfacing with a register to facilitate the accurate processing of a payment |

**3.1.2 Administrator:**

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the Restaurant. The administrator has all the information about the users and about all food item.

This module is divided into different sub modules.

* + - 1. Manage Suppliers
      2. Manage Food item
      3. Manage Users
      4. Manage Orders
      5. Manage Delivery Boy

The following are the identified functional **Admin** requirements that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| **Requirement** | Description |
| A1 | Admin shall able to Manage users |
| A2 | Admin shall able to Manage providers/suppliers |
| A3 | Admin shall able to Mange accounts |
| A4 | Admin shall able to CRUD food items for restaurant |
| A5 | Admin shall to manage restaurant details |
| A6 | Admin shall to manage restaurant ratings |
| A7 | Admin shall able to manage Delivery boy details |
| A8 | Admin shall able to display top dishes with highest ratings |

**3.1.3 Suppliers/Restaurant:**

The following are the identified functional **Restaurant** requirements that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| **Requirement** | Description |
| R1 | Restaurant shall able to CRUD food items from menu |
| R2 | Restaurant shall be able to receive orders from customers |
| R3 | Restaurant shall be able to view the orders which has been ordered by customers |
| R4 | Restaurant shall be able to receive acknowledgement from Delivery Boy |
| R5 | Restaurant shall be able to view the payment |
| R6 | Restaurant shall able to receive acknowledgements on delivered orders |
| R7 | Restaurant shall able to give offers on food items |

**3.1.4 Customer:**

The following are the identified functional **Customer** requirements that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| **Requirement** | Description |
| C01 | Customer Shall be able to login or skip from registration to enter the menu dashboard |
| C02 | Customer can search the food item as per his wish in specific category. |
| C03 | Customer shall be able to order foods and add to cart |
| C04 | Customer shall be able to remove orders from cart |
| C05 | Customer shall be able to navigate between menu and can add items to cart |
| C06 | Customer has a privilege to his order and can see his order details. |
| C07 | Customer shall be able to see his order history |
| C08 | Customer shall be able to cancel the order |
| C09 | Customers can buy food items from his cart by making payment. |
| C10 | Customer can have a wish list for future buying food item he can add food item in the wish to list. |

**3.1.5 Delivery Boy:**

The following are the identified functional **Delivery boy** requirements that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| **Requirement** | Description |
| D1 | Delivery Boy shall be able to sign in, sign out and the delivery boy is added by the admin. |
| D2 | Delivery boy shall able to see the pending order details after signing into his account. |
| D3 | Delivery boy shall able to see the delivered order details after signing into his account. |
| D4 | Delivery boy shall able to see the payment history of his delivery food item. |
| D5 | Delivery boy shall able to reach the restaurant and check with order details |
| D6 | Delivery boy shall able to receive customer details from restaurant |
| D7 | Delivery boy shall able to pick up and deliver order to customer |

* 1. **NON-FUNCTIONAL REQUIREMENTS**

This subsection presents the identified non-functional requirements for the Yumzy App. The subcategories of non-functional requirements given are safety, security, interface, human engineering, qualification, operational and maintenance.

**3.2.1 Safety**

The following are the identified non-functional **safety** requirements that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| S1 | The system shall log every state and state change of action, tablet and display to provision recovery from system failure. |
| S2 | The system shall be capable of restoring itself to its previous state in the event of failure (e.g. a system crash or power loss). |
| S3 | The system shall be able to display a menu at all times to facilitate manual order taking should the need arise. |
| S4 | The system shall utilize periodic 120-second keep-alive messages between mobile and the server to monitor app operational status |

* + 1. **Human Engineering Requirements**

The following are the identified non-functional **Human engineering requirements** that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| H01 | Any element of the system will take no longer than 10-seconds to restart. |
| H02 | Admin must not dismiss an engaged menu unless the customer requests it. |

* + 1. **Efficiency Requirement**

The following are the identified non-functional **Efficiency Requirement** that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| E1 | When an online food delivery portal is implemented, the customers can purchase food item in an efficient manner. |

* + 1. **Reliability Requirement**

The following are the identified non-functional **Reliability Requirement** that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| R1 | The system should provide a reliable environment to both customers and owner. |
| R2 | All orders should be reaching at the admin without any errors. |

* + 1. **Usability Requirement**

The following are the identified non-functional **Usability Requirement** that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| U1 | The Web application is designed for user friendly environment and should be ease of use. |

* + 1. **Implementation Requirement**

The following are the identified non-functional **Implementation Requirement** that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| I1 | The System is Implemented using React in frontend with Spring Boot as backend and it will be used for database connectivity. |
| I2 | The Systems Database part is developed using MySQL. |
| I3 | Responsive web designing is used for making the website compatible for any type of screen. |

* + 1. **Delivery Requirement**

The following are the identified non-functional **Delivery Requirement** that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| D1 | The whole system is expected to be delivered in four months of time with a weekly Evaluation by the project guide. |

* 1. **PERFORMANCE REQUIREMNET**

The following are the identified non-functional **Performance requirements** that directly relate to the entire Yumzy System.

|  |  |
| --- | --- |
| Requirements | Description |
| P1 | The server shall be capable of supporting no less than 200 concurrent connections from any combination of computers, tablets and displays. |
| P2 | The server shall be capable of supporting an arbitrary number of active orders, that is, no orders shall be lost under any circumstances. |

* 1. **HARDWARE & SOFTWARE REQUIREMNETS**
     1. **Hardware Requirements**

Hardware requirements for insurance on internet will be same throughout which are as follows:

|  |  |
| --- | --- |
| **RAM** | 2 GB |
| **Hard disk** | 320 GB |
| **Processor** | Dual Core |

* + 1. **Software Requirements**

**Client side:**

|  |  |
| --- | --- |
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

**Server side:**

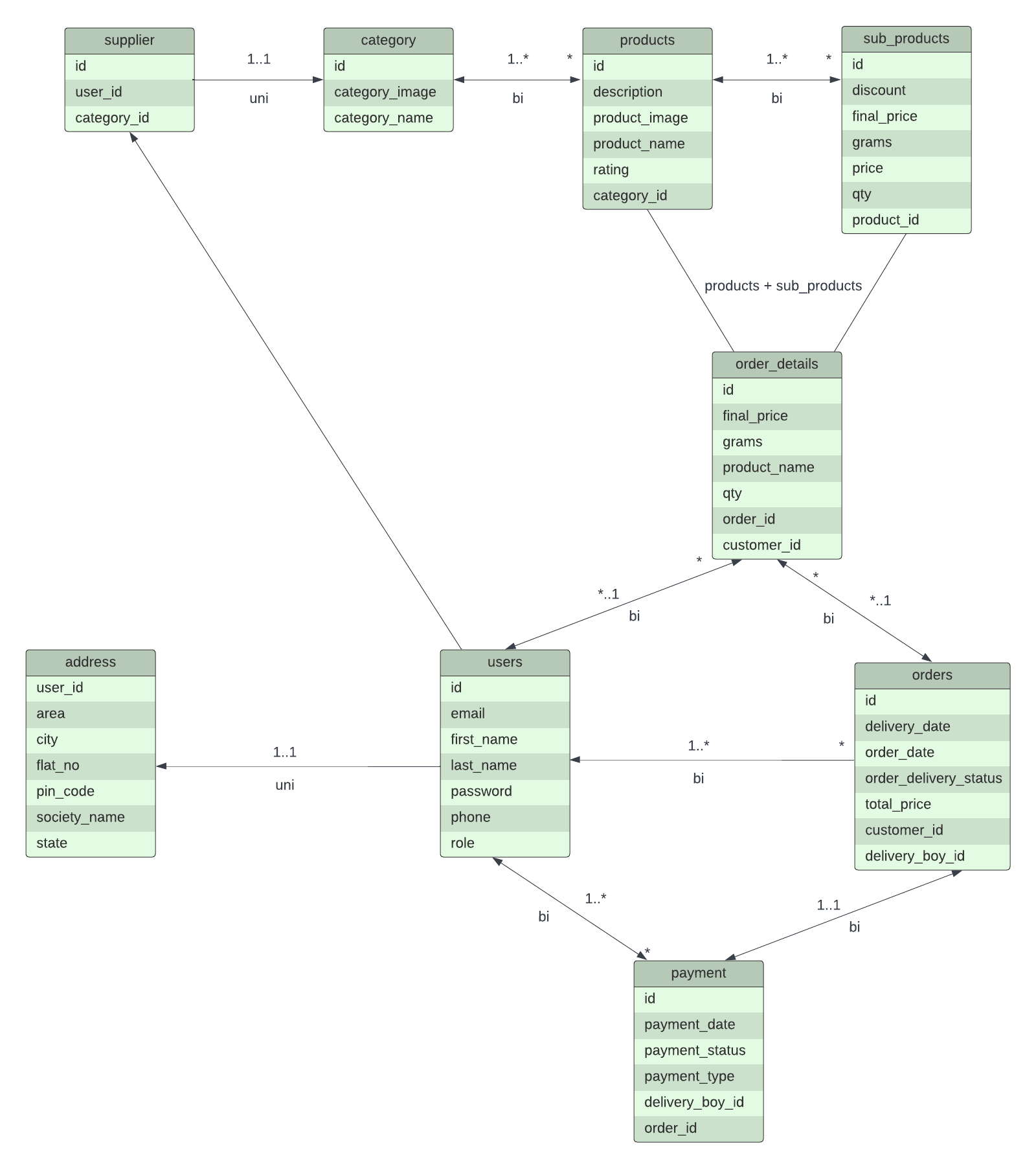
|  |  |
| --- | --- |
| **Web Server** | TOMCAT |
| **Server side Language** | Spring Boot |
| **Database Server** | MYSQL |
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

1. **SYSTEM DESIGN SPECIFICATION**

**4.1** **ER Diagram**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

* It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

****

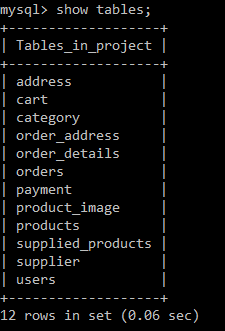
**5.2 Database Design**

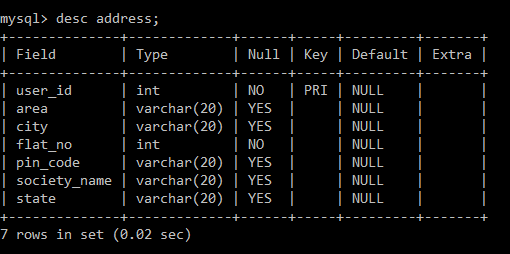
The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant database.

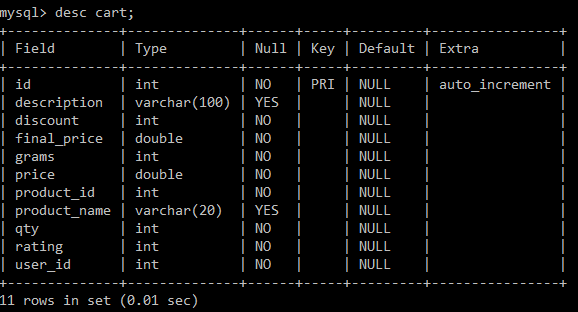
**5.3 Table Structure**

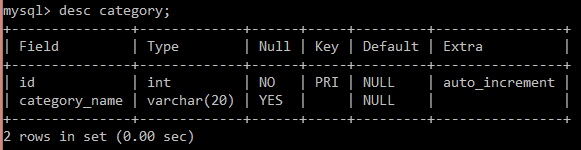
**5.3.1 Tables**

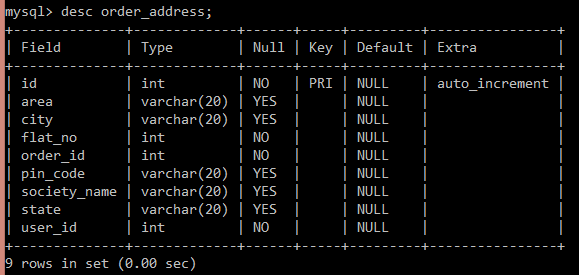


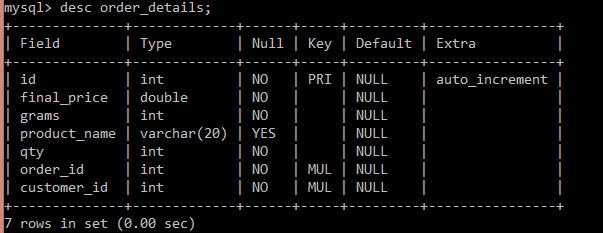
**5.3.2 Address**

**5.3.3 Cart**

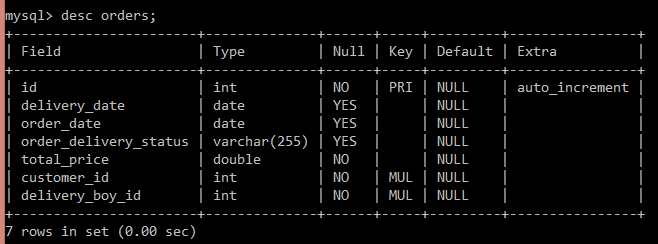


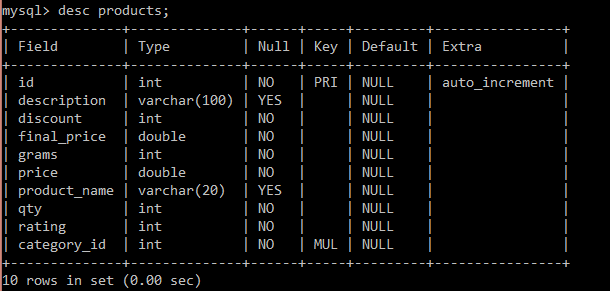
**5.3.4 Category**

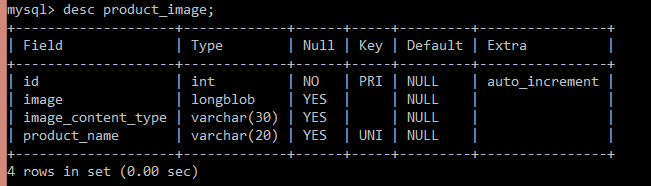
**5.3.5 Order Address**

**5.3.6 Order Details**

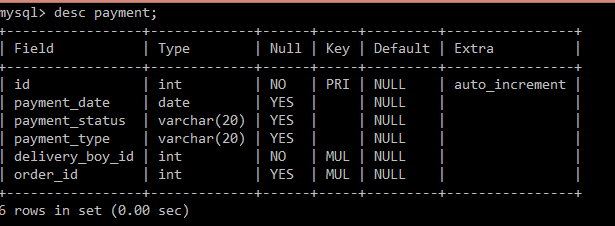
**5.3.7 Orders**

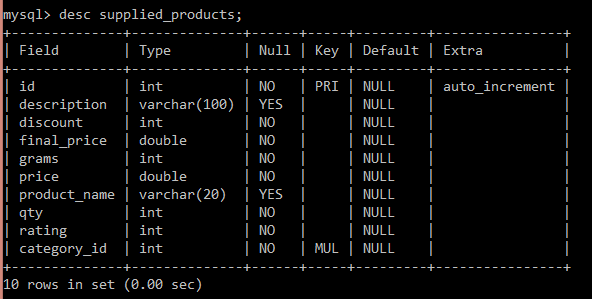


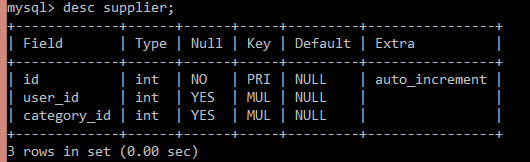
**5.3.8 Products**

**5.3.9 Product Image**

**5.3.10 Payment**



**5.3.11 Supplied Product**

**5.3.12 Supplier**

**5.3.13 Users**

