2. SOFTWARE REQUIREMENT SPECIFICATION

2.1 Introduction:

This Software Requirement Specification document provides a complete description of all the functionalities and the specifications of the **“Online food mess”** system. The following section provides an overview of the derived Software Requirements Specification (SRS) for the application. To begin with, the purpose of the document is presented**,** and its intended audience is outlined.

Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant software will do and the relevant benefits associated with it. The nomenclature used throughout the SRS is also offered. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

2.1.1 Purpose:

Apeato is a website designed primarily to get the users better food on daily basis. This system will allow customers to order food at very affordable cost. The system also allows to quickly and easily manage an online menu which customers can browse and use to place orders with just few clicks. Restaurant employees then use these orders through an easy to navigate graphical interface for efficient processing. The purpose of this document is to describe the external requirement of the project.

The main purpose is to translate the ideas in the mind of the client into a formal document. A software requirements specification (SRS) minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs, and human users in a wide variety of real-world situations. The SRS provides critical information about development operations, quality assurance, and maintenance to the developers.

2.1.2 Scope:

- The main objective of Apeato is to provide the food to customers through online, with the help of internet.

- It helps the customer to order the food by sitting at home or any other places.
- With the help of these apps, the manual work is reduced.
- The order is taken through online, and the food is delivered to the told place.

2.1.4 Definition, Acronyms, Abbreviations:

GUI	: Graphical User Interface
DBMS	: Database Management System
RDBMS	: Relational Database Management System
SRS	: Software Requirement Specification
ADMIN	: The Administrator.
CPU	: Central processing unit
PHP	: Hypertext Preprocessor.
SQL	: Structured Query Language.
HTML	: Hyper Text Markup Language.
CSS	: Cascading style sheet

2.1.4 References:

www.w3schools.com

www.geeksforgeeks.com

2.1.5 Overview:

APEATO is a website designed primarily for use in the food delivery industry. This system will allow customers to order food at very affordable cost on daily basis. Apeato is a web-based application which enables customers to order their food online for home delivery or pick up from the restaurant. This SRS is made to briefly describe the requirements of the project of online food ordering using speaking agents. It will tell us about all the requirements for making this project.

2.2 Overall Description:

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce its basic functionality of it. It will also describe what type of users will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.2.1 Product Perspective:

The **"APEATO"** software is a fully independent product and not a part of any other system. The users of the system are categorized as admin and customers. The system gives online access to each separate user. The software described in this SRS is the software for a complete Restaurant food ordering system. It relies on several external interfaces for persistence and unhandled tasks, as well as physically interfacing with humans. The system gives online access to each separate user. This system provides an easy way for customers to order food online while sitting at home. It provides a simple database rather than complex ones for high requirements and it provides a good and easy graphical user interface (GUI) to both new as well as experienced users of the system. This project mainly focuses on the solution regarding the existing system where the users must go to restaurant and order food.

2.2.2 Product Functions:

- Web Ordering System- provides the functionality for customers to place their order online.
- Menu Management-allows the restaurant to control what can be ordered by the customers.
- Order Retrieval System-This is a final logical component. Allows restaurant to keep track of all orders placed.
- It decreases the load of a person involve in existing manual system.
- It enlarges the flexibilities in the existing system with a web-based user-interactive interface.
- The customer can give the feedback for every item ordered.
- It manages all the activities online.

2.2.3 User Classes and Characteristics:

The system has two user levels.

Admin:

The admin can log in to the system using his/her unique credential. Here admin can manage the details of customers and the restaurant.

- Login.
- Manage profile.
- Manage users.
- Update menu.
- Update package
- Order details.
- Manage orders.
- Payment.
- view Employee information
- manage payment.
- View feedback.
- Report.

Customer:

Customers can register through email-id and password. They can view the menu, place an order, track order and add them to the cart. They can also do the payment.

- Registration
- Login
- Manage profile
- View menu
- View package
- Order
- Payments
- Write feedback

General Constraints:

General constraints include the following:

- This application requires an internet connection.

- Customers must register and get confirmation from the admin to access the website for the first time.
- Only restaurant can manage the menu.
- No one has the right to change the information of someone else account on this website.
- The end system should also allow for seamless recovery, without data loss, from individual device failure.

2.2.4 <u>Assumptions and Dependencies:</u>

- It is assumed that this system has two types of users, i.e., admin and customer.
- The restaurant should be careful while updating menu items.
- All the data entered will be correct and up to date.
- It is assumed that the needed changes, to collect and store the data, will be made within the current application and database.

2.3 SPECIFIC REQUIREMENTS:

2.3.1 External Interface Requirements:

2.3.1.1 User Interface:

A user interface is a point of human interaction and communication with the system.

We have taken the following requirements during design,

- Textboxes to enter details.
- Buttons to add, delete, update and search.
- Labels to display the information.
- Checkboxes.
- Combo boxes and list boxes.
- Grid box to display the information.

2.3.1.2 Software Requirements:

- Operating system: Windows.

- Text editor: Sublime Text 3
- Language: PHP
- Server: Apache
- User interface: HTML, CSS, JavaScript
- Database: MySQL
- Browser: Chrome, Mozilla Firefox, or any other browsing application.

2.3.1.3 Hardware Requirements:

- Processor: Intel Pentium dual-core or above
- Processor Speed: 2GHz
- RAM – 1GB
- Hard Disk – Minimum 40 GB

2.3.1.4 Communication Interface:

The communications function required by this product is HTTP protocol, and internet communication is through TCP/IP protocol.

2.4 Functional Requirements:

a) Admin:

- **Login**: Admin will enter the website using username and password.
- **Manage Restaurant**: In this module, the admin can manage all the restaurants details who have registered to the system i.e he can approve/reject, view and delete the restaurants details.
- **View Customers**: In this module, the admin can view all the customers details who have registered to the system.
- **Manage order Report**: In this module, the admin can view the order report of all the restaurants in the particular food court.
- **Manage table bookings**: In this module, the admin can add the tables that are available with image.
- **View payments**: In this module, the admin can view payments made by the customers.
- **View Feedback**: The admin can view the feedback given by the customer.

b) Customer:

- **Registration:** The user can give his/her basic details.
- **Login:** The user can log in using his/her username and password.
- **Manage Profile:** The user can manage personal details.
- **View Restaurant:** The user can view the details of the restaurant in the system.
- **View Menu:** The user can view different menu provided by the restaurants.
- **Place an order:** In this module, the user can place an order.
- **View order status:** In this module, the user can keep track of the order status.
- **Payments:** The user can make payments for the restaurant.
- **Send Feedback:** The user can send feedback to the restaurant.

2.5 Performance Requirements:

- The server shall be capable of supporting an arbitrary number of active customer payments, that is, no payments shall be lost under any circumstances.

- Page load time should be less than 40 sec.
- Should have a good memory space.
- The server shall be capable of supporting an arbitrary number of active orders, that is, no orders shall be lost under any circumstances.
- Should be error-free.
- 1MB file should get uploaded in 60 sec.

2.6 Design Constraints:

- While customers register to the system, mandatory fields must be checked for validation whether the customer has filled appropriate data in these mandatory fields. If not, a proper error message should be displayed, or else the data is to be stored in a database for later retrieval.
- All mandatory fields should be filled by the customers while ordering the food from the restaurants
- All mandatory fields should be filled by the customer while adding the profile details.
- The system must be designed in such a way that will be easy to use and visible on most browsers**.**

2.7 System Attributes:

The Quality of the database is maintained in such a way that it can be very user-friendly to users of the database.

- **Performance**: It defines how well software system accomplishes certain functions under specific conditions.
- **Reliability:** The system that performs correctly during a specific time.
- **User-friendly Interfaces:** An interface that is created for specific group of people or without a clear target audience but aimed at maximum user interface.
- **Maintainability:** The ease with which you can repair, improve, and understand software code.
- **Portability:** This system shall be portable, and we can switch the server very easily.

- **Flexibility:** The ability of a system to respond to potential internal or external changes effecting its value delivery, in a timely and cost affecting manner.
- **Timeliness:** The system shall carry out all the operations with consumptions of very less time.

2.8 Other Requirements:

2.8.1 Safety Requirements:

- There are two user levels in online food ordering system, Access to the various subsystems will be protected by a user log-in screen that requires a username and password. This gives different views and accessible functions of user levels through the system.
- Email ID once registered to the system cannot be changed to make every user unique and easily identifiable
- Maintaining backups ensures the system database security. The system can be restored in any case of an emergency.

2.8.2 Security Requirements:

- The server on which the '**Apeato**' resides will have its security to prevent unauthorized write/delete access. There is no restriction on reading access.
- The proposed website will be secure. There are different categories of users they are admin and customer.
- Depending upon the category of using the access rights are decided.
- Admin has the maximum privilege to all subsystems.