Software Requirement Specification(SRS)
FOR
Food Delivery System (FDS)

Revision History

Version No.	Date	Prepared by / Modified by	Significant Changes
	26.03.2022	SNEHA JOSHI ARPITA GUPTA AKASH SHARMA NIKHIL KAUSHIK MEHAK KAPOOR PALADUGULA SRI LAKSHMI PRAVALLIKA	

Glossary

Abbreviation	Description
OFDS	Online Food Delivering System

Table of Contents

1 Introduction

ONLINE FOOD DELIVERY SYSTEM is a website designated primarily for use in the food delivery industry. This system allows to quickly and easily manage an online menu which customers can browse and use to place orders with just few clicks.
The Online Food Delivery System that we are proposing here, greatly simplifies the ordering and delivering process for both the customer and the restaurant. The system presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Users also get the facility of tracing their order whenever they want.

2 Scope of Change

- This system would be hassle free for users as they can select the food item they want and make payment for it.
- Reduce the purchasing time for customers.
- Improved and efficient services are provided to the customers by the inclusion of internet.

2.1 Functional:

Registration: If user is a new user, then he/she can register on system. For registration user needs to fill registration form, after this system send OTP to register Mobile number or Verification Mail to registered Email ID to activate the account.

Login: After successfully registration user can login into the system by registered mobile number and password or registered Email ID and password.

Home: Home is the interactive or dynamic page that shows Food, Restaurant, Special Offers and Categories. User can click on any function of the home page of his/her choice or they can also search it.

Membership: If user get access of Membership, then He/she get the free delivery or additional discounts.

Logout: User can remove from the system by clicking Logout.

Cart: User can add products in the cart and purchase it. They edit, remove and add product in the cart. User can also add Food to wish list if user want to buy food later.

Placed order: By clicking on the placed order user can buy food at that time.

Notification: A notification sends to user for reminder, Order confirmation, as an alert and offers.

Payment: After buying of the product user need to pay the amount online they have different options like Credit card, Debit card, Wallets, Net Banking or they can pay via Cash on Delivery option.

Bills: There will auto bill generator after the buying of food and it will send to User on their registered mobile number or Email ID.

Your Order: Your order is the page where user can get all the food details current as well previous orders.

Order Tracking: This function is use by User to track their order.

Feedback or Review/Rating: User will able to give their valuable feedback or their suggestions. They can review and rating the restaurant and dishes.

Customer Care: User can clarify their doubts with executives to get in touch with them via customer care number or Email ID.

2.2 Non-Functional:

Language Issue: There are many other language options based on different region.

Operational requirements: The system should operate properly in browsers.

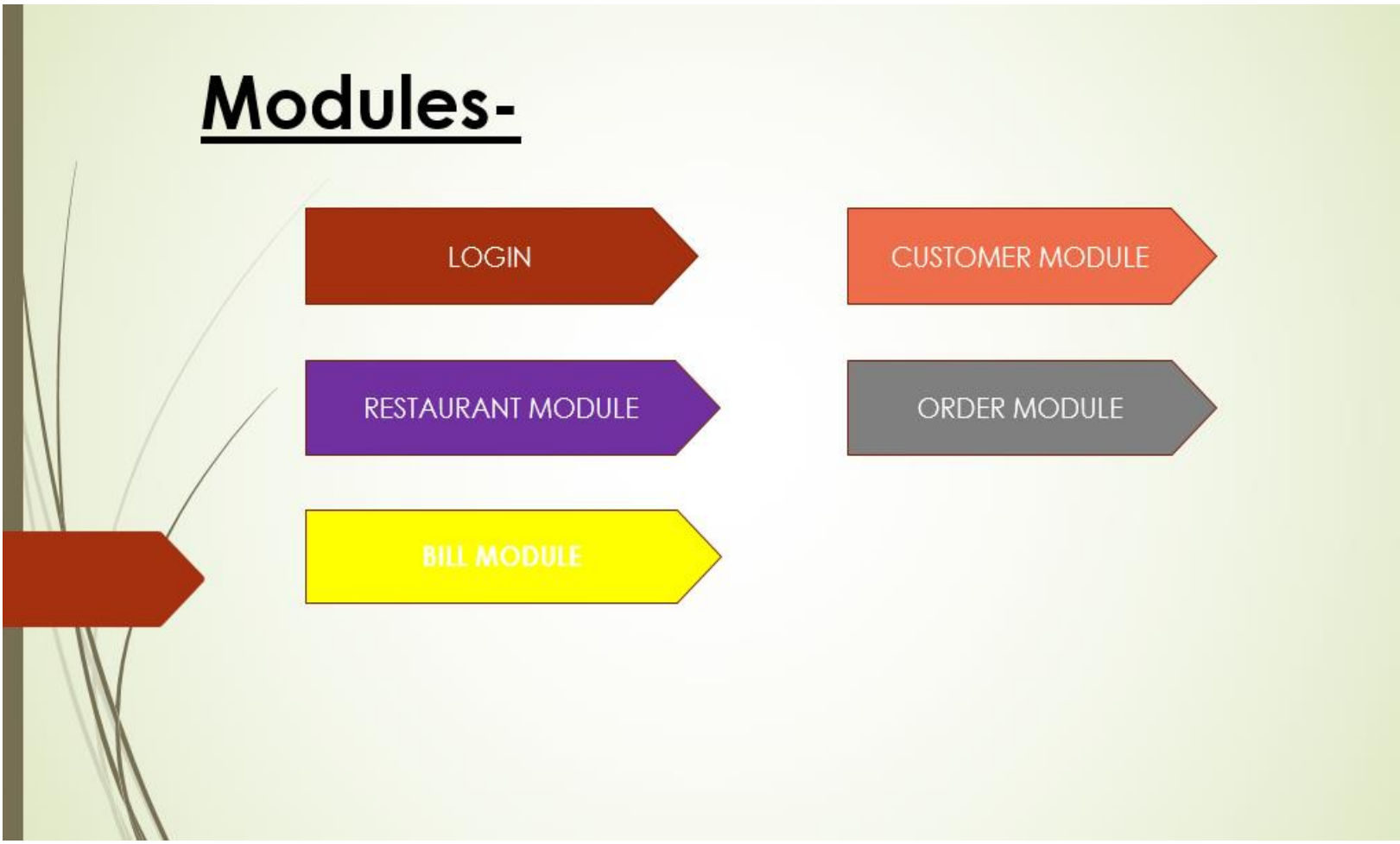
Performance requirements: The system should let user place an order in a short period of time.

Security requirements: The system shall validate the username and password in order to login and make changes to the system.

Passwords should be encrypted.

Usability requirements: The system should have a simple and easy-to-learn graphic user interfaces.

3 List of impacted modules



4 Design and Detailed technical updates

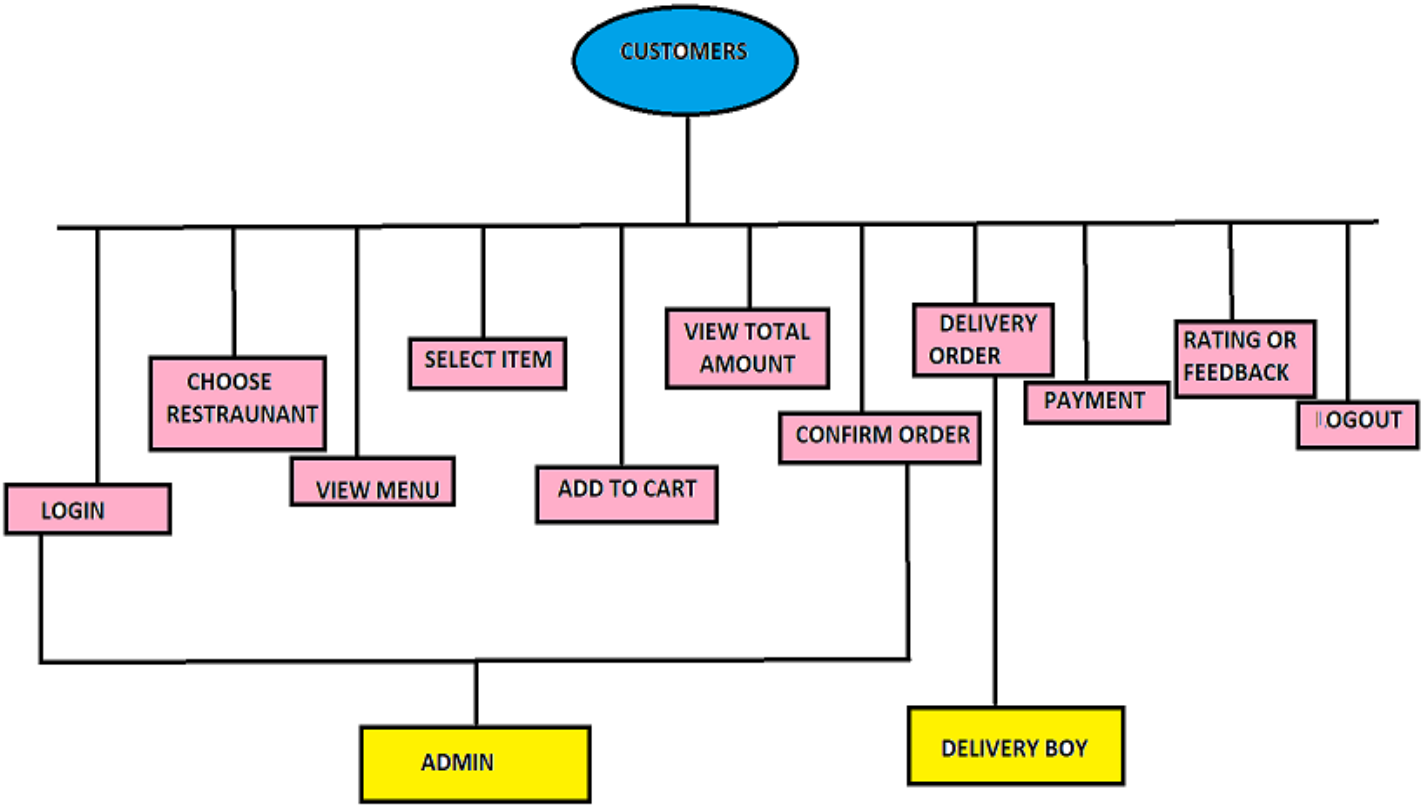
4.1 Process model

This subsection extends upon the functional requirements given through the presentation of detailed use cases. To facilitate requirements given through the presentation of detailed use cases. To Facilitate an unambiguous and clear view of how the end-users interact with the system, the end-users involved in the use cases, a use case diagram and detailed use case descriptions and provided.

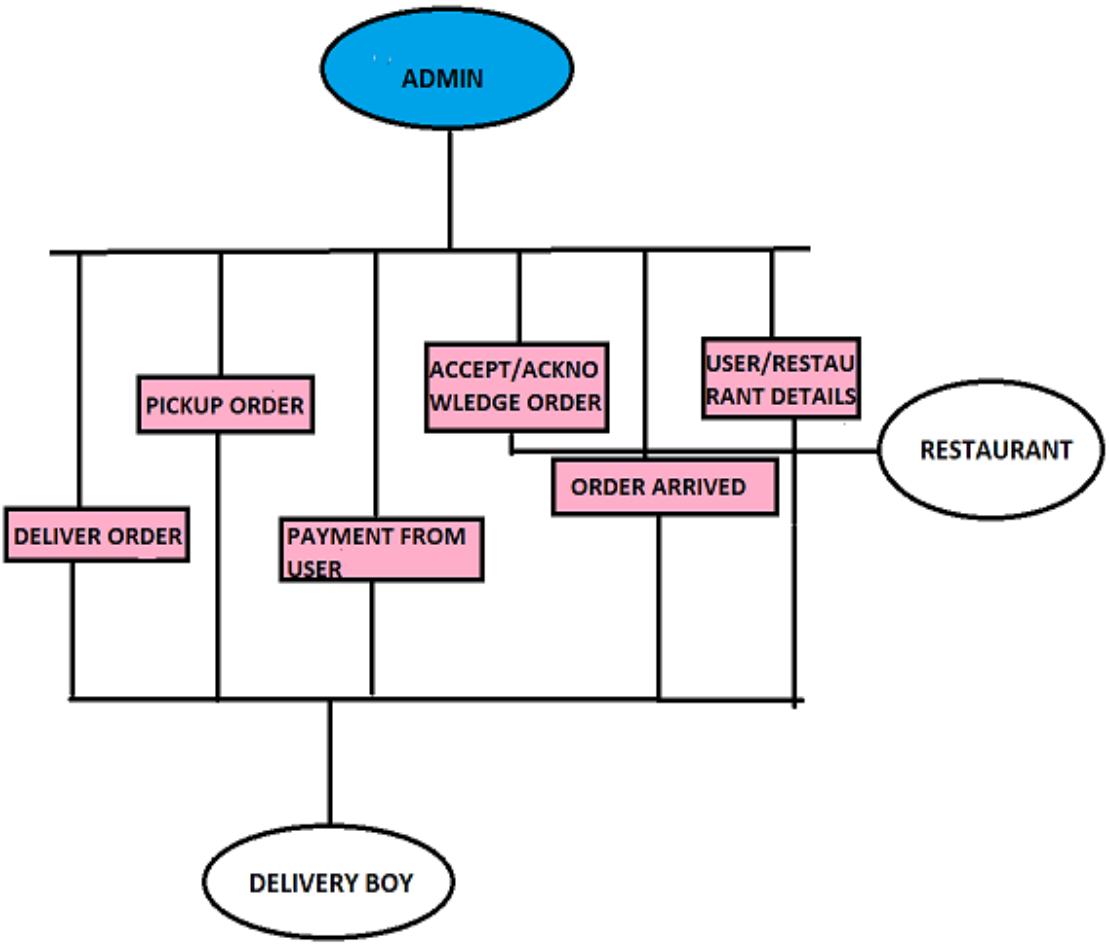
4.1.1 Use case Model

There are four end users in the system- Customer, Restaurant, Admin, Delivery boy.
Base specializations: Customer, Restaurant and Admin
Delivery boy interacts between both the customer and Restaurant as a generalization.

Use case for Customer

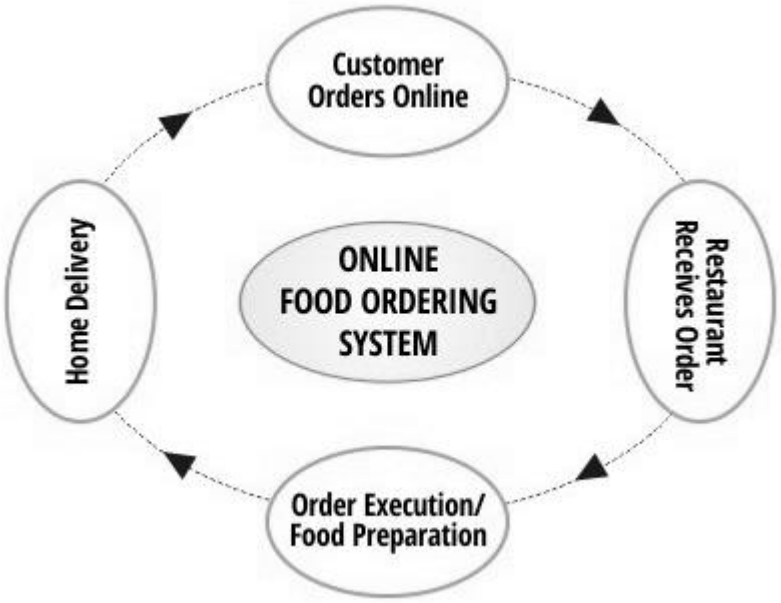


Use case for Restaurant:

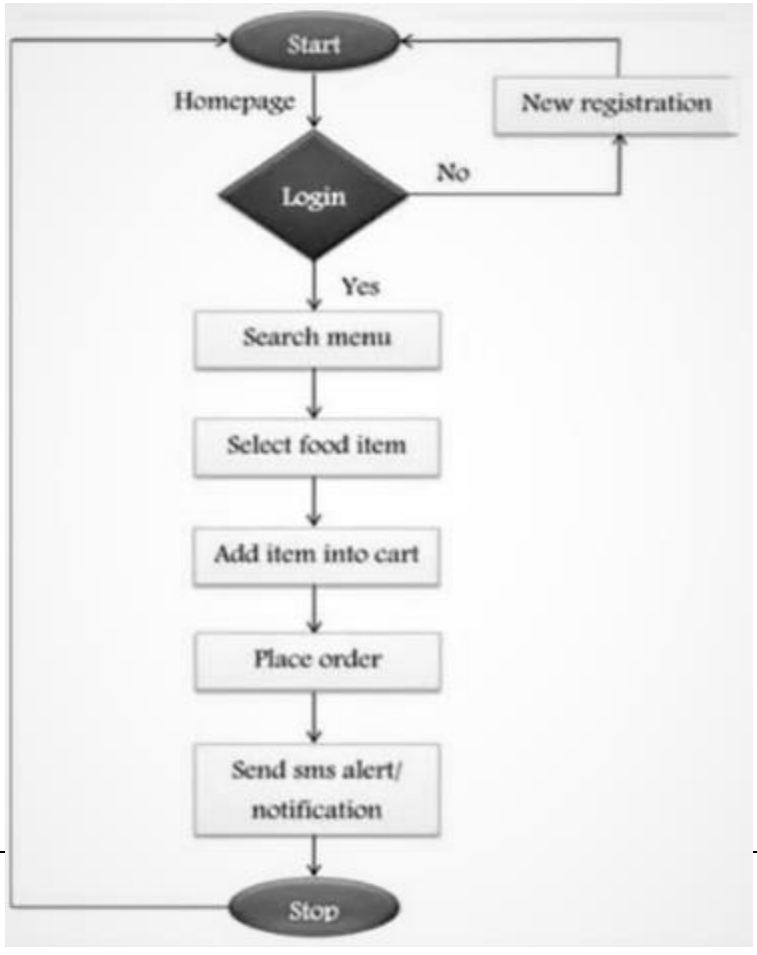


4.1.2 Workflow diagram

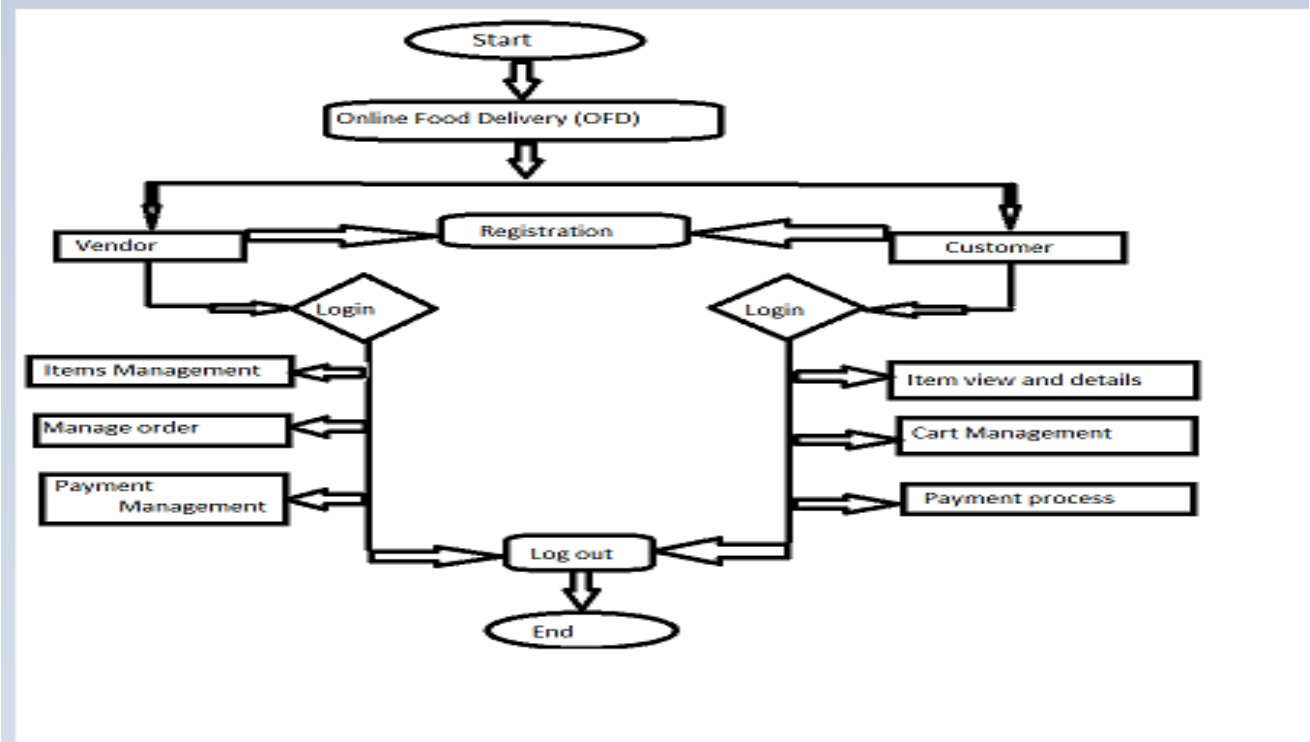
This diagram shows about the main process of the system.



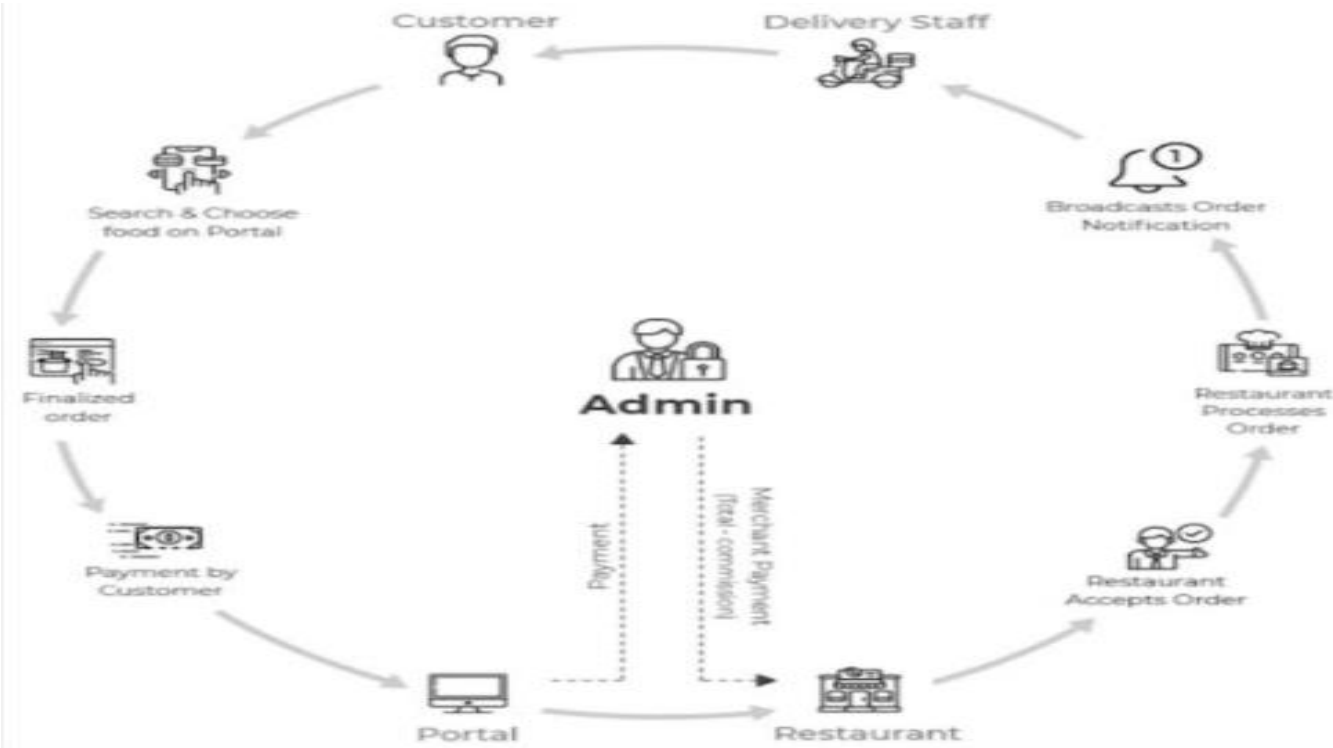
This diagram shows the process in customer's point of view.



In flow chart shows about the features and the facilities of our system. It can also capable to provide a special feature for the vendor's/food center owners to create an account and provide the best services.

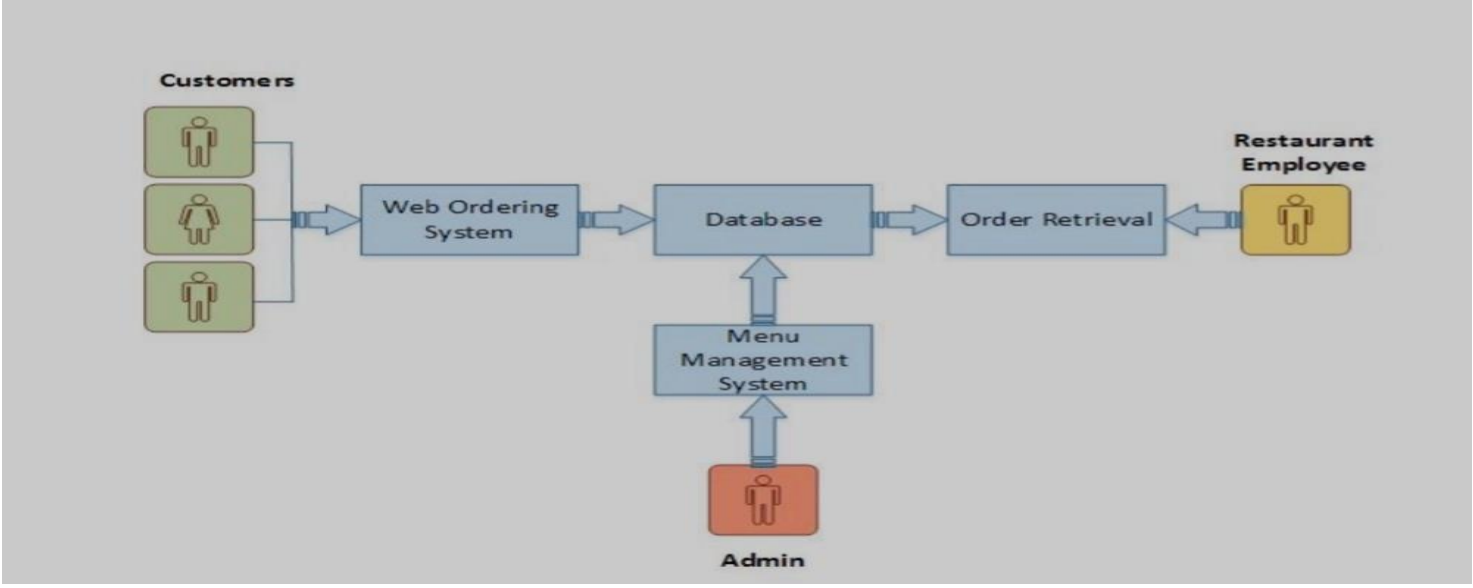


The flow chart below shows about the pictorial representation of the system process clearly.



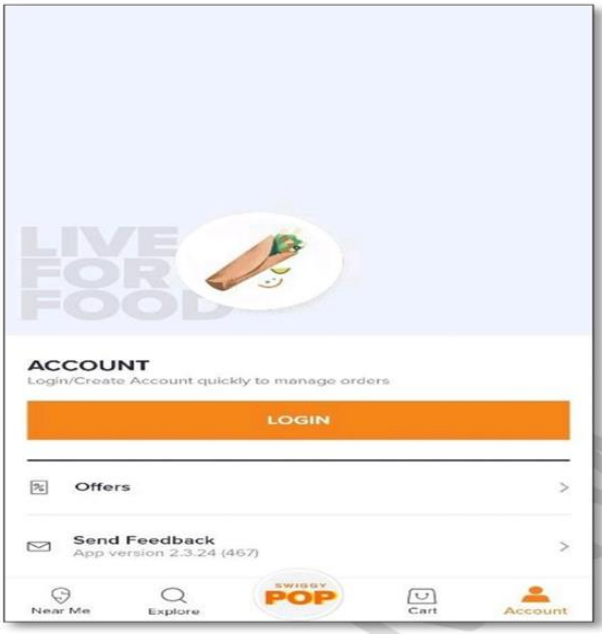
4.2 Product Perspective

The Online food delivering system application would have the following basic function:

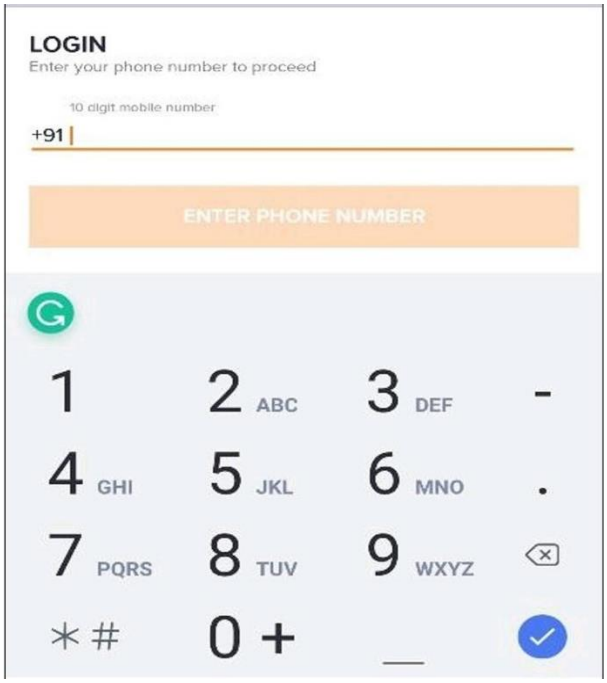


5 Details of Alternative Design Specification

USER LOGIN/CREATE

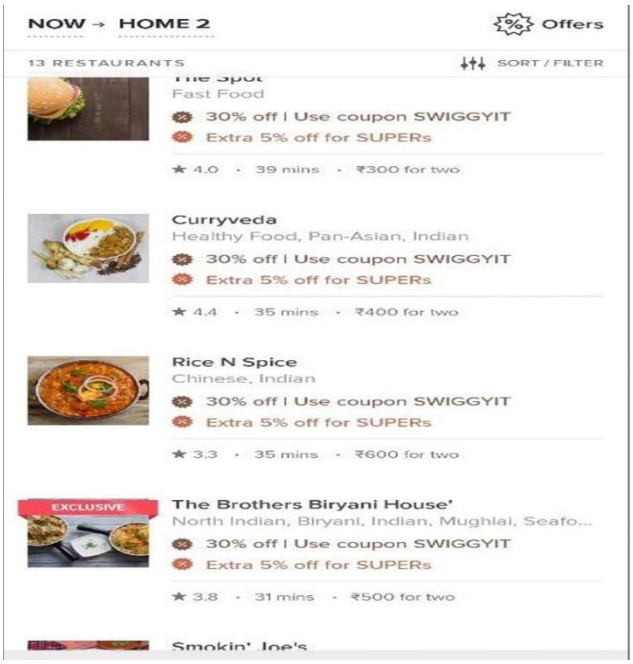


ENTER REGISTERED MOBILE NUMBER

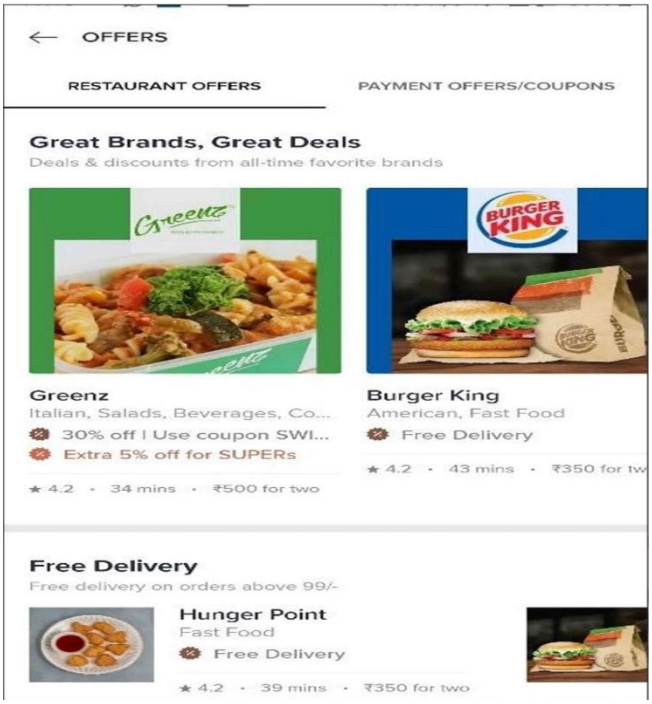


OTP VERIFICATION

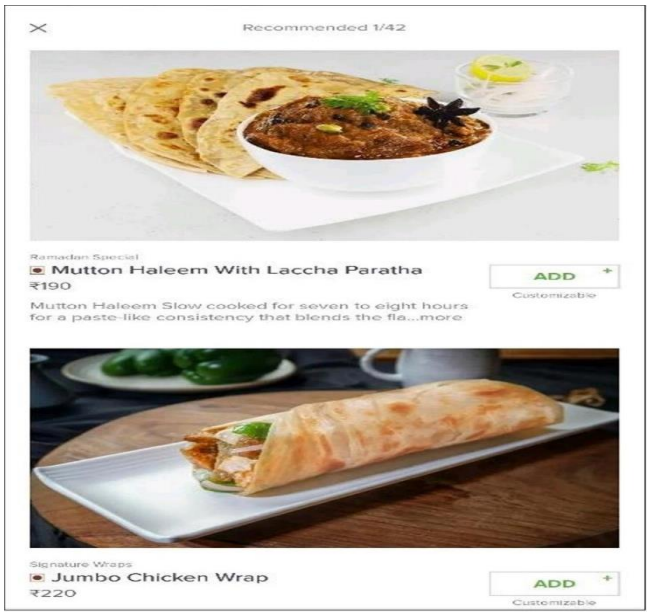
MAIN MENU



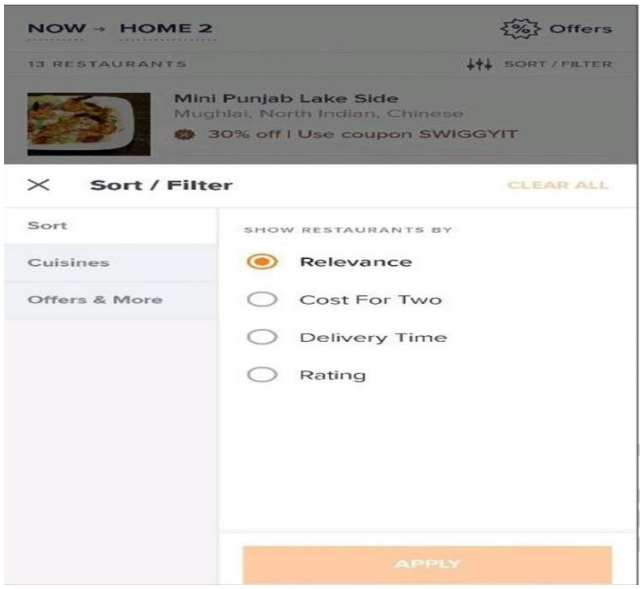
OFFER DISPLAYED



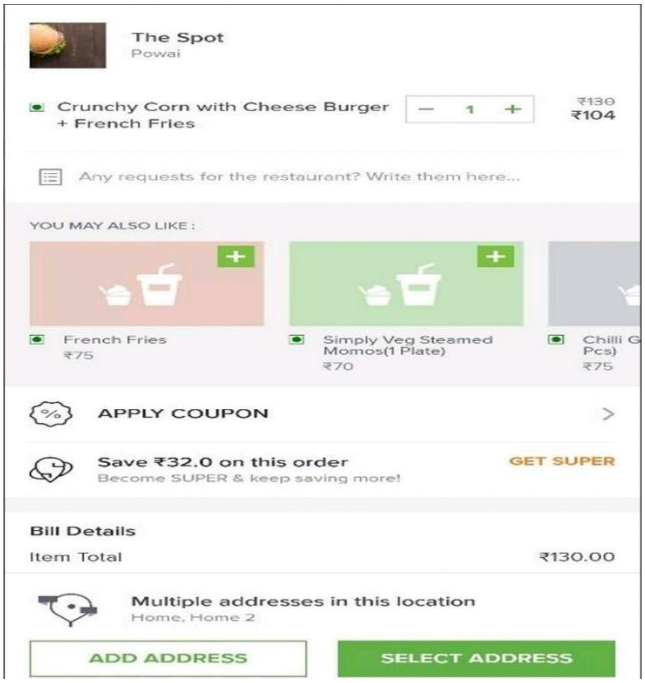
ITEM DESCRIPTION



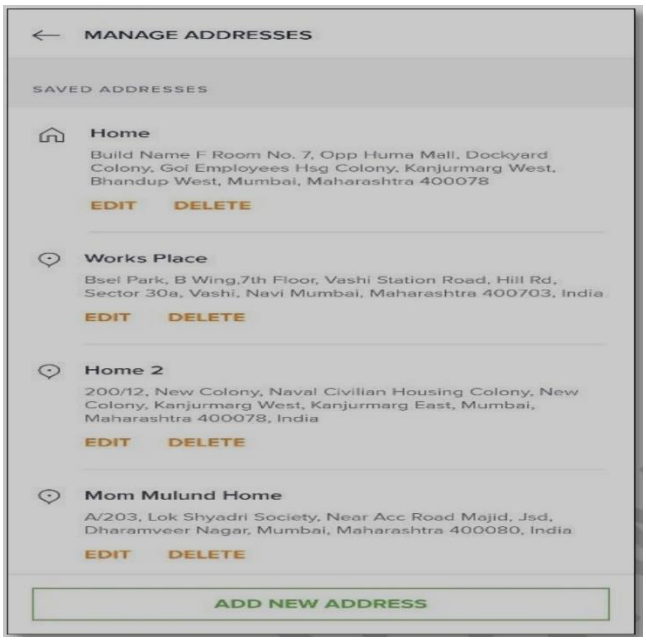
FILTER



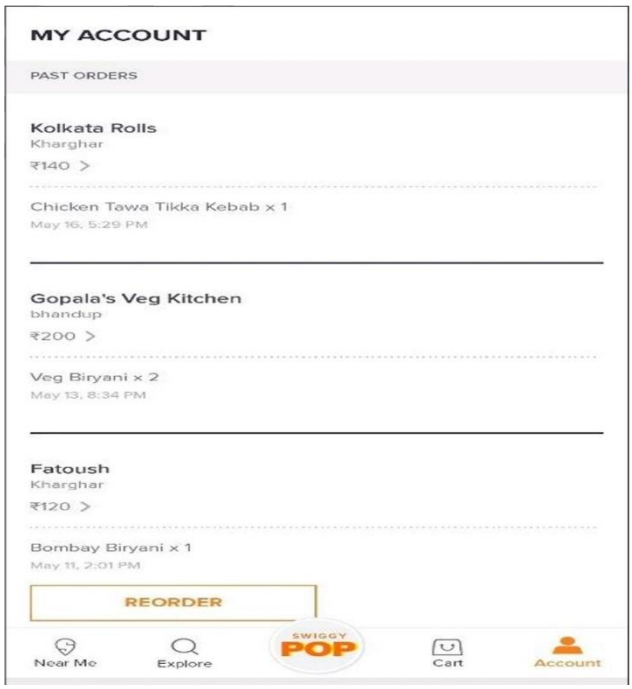
ADD ITEMS



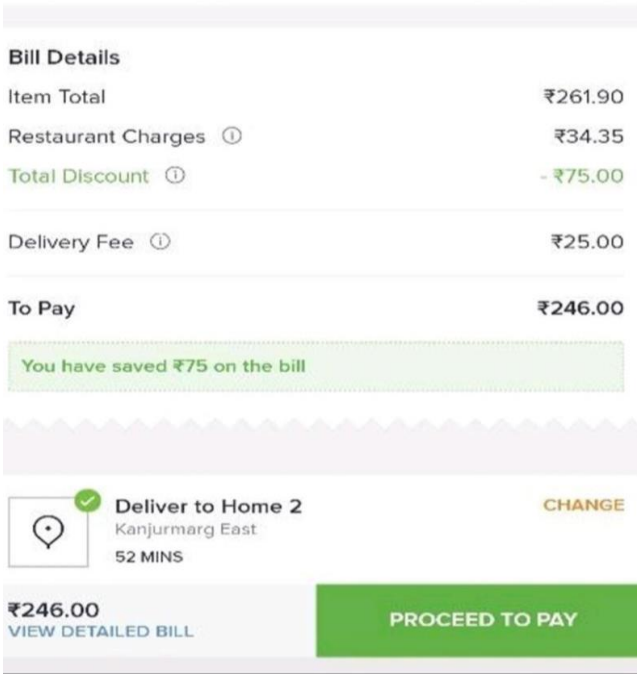
MANAGE ADDRESS



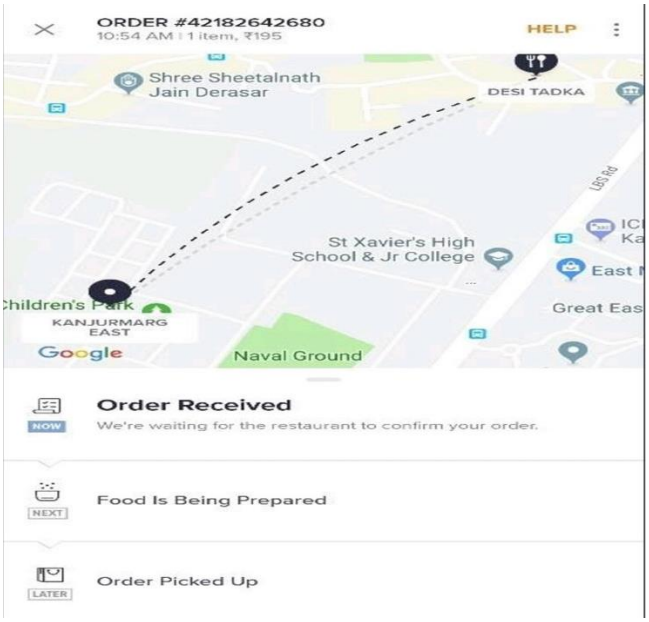
ORDER DISPLAY



PROCEED TO PAY



ORDER STATUS



FEEDBACK

6 Specific Requirements

6.1 Functional Requirements:

These type of requirements can be thought of as an 'objective' of the software product to be built. The functional requirements form the core of the entire software development process. When a software project is undertaken, the functionalities it must serve, are clearly outlined in the requirements specification document.
It is very important to understand what the client's needs are, the need for a software product, output expected from the system. In short, the key functionalities a software system must deliver.

Functionality can be understood in the context of a real world scenario. For instance, one needs an application to calculate the price for selected products and generate an invoice for the same. The application should also be able to integrate components such as comparison of one product with another similar one of a different brand, and list comparisons to enable the user to make a wise choice. Such facilities can be termed as functional requirements.

6.1.1 Functional requirement is supposed to adhere to some basic ground rules:

1-Functional requirements to be met should be aligned with the business requirements.

2-Timely review of the functionalities to ensure that the project is moving in the right direction.

3-Adhering to rules and policies, and not leaking information to any external sources.

6.2 Non-Functional Requirements:

These set of requirements are an extension to the functional requirements. Non-functional requirements are not the ones that a user demands, but is implicitly expected from the application developed so far. It can be thought

of as the “efficiency” of the application in terms of its performance. This term can be further elaborated in the light of a simple fact.

When we order food online, the food has to be the one that is selected, but beyond that many companies offer delivery within 30 minutes, else, free delivery. This free delivery can be thought of as a non- functional requirement which extends beyond providing the basic functionality, that is food delivery.

Similarly following factors form the major areas which constitutes a set of non -functional requirements.

Efficiency - It is the parameter to measure the performance of the software application.

Quality - Quality is the overall factor contributing to non- functional requirement. A quality end product is what everyone demands.

Response Time - The time taken to perform an action triggered by the user.

Availability - The extent to which a system is available for use initially. If more technically defined, then it can be stated as the proportion of time a system is in functioning condition.

Interoperability - Easy to use interface that should enable the user to work with any product with ease.

Security- Application should not be vulnerable to any exploitation. That is, strict authentication mechanisms to be applied to the system.

Fault Tolerance- A major aspect of any software product. It portrays the capability of a system to continue performing in the event of failure of some other components.

Documentation - Provides an overview about the system, clearly lays down the specifications in detail.

7 Additional details

7.1 Assumptions

The SRS assumes that none of the constituent system components will be implemented as embedded applications.

- List of Restaurant and dishes will be provided by the Client.
- Images of the Restaurant and dishes will be provided by the vendors.
- Images of the Restaurant and dishes will be in High Definition.
- Restaurant and dishes description will be provided by vendors.
- Terms and Conditions will be provided by legal team.
- Return Policy will be provided by legal team.
- Discount & Offers will be provided by the merchants.
- Newsletters and Emails will be provided by the marketing team.
- Company Address will be provided by the client.
- Company Contact Number will be provided by the Client.
- All the Content of the Company will be provided by the Client.
- Requirement will be Sign off by the Client.
- Finally, it is further assumed that the deployment environment is capable of supporting an IEEE 802.11 wireless network for system communication.

7.2 Constraints

- Return Policy is subject to approval by legal team.
- Design is subject to approval by UI team.
- There must be a strong audit chain with all system actions logged.
- The system has criticality insofar as it is a live system.
- The RMOS should be written in an object-oriented language with strong GUI links and a simple, accessible network API.
- The primary candidate tool chains are Java/Swing, C++/Qt and Python/Qt.
- The system must be reliable enough to run crash and glitch free more or less indefinitely, or facilitate error recovery strong enough such that glitches are never revealed to its end-users.

8 References

1. **http://getbootstrap.com/**
2. **https://javabrains.io/**
3. **https://docs.oracle.com/cd/E2468_01/server.121/e41484.pdf**
4. **https://www.youtube.com/watch?v=ZBg**