**Project Log Book**

**Group Members:**

|  |  |
| --- | --- |
| **Entry Date** | **Work Done** |
| January 7th, 2022 | Discussed the basic plan to build the prototype for RM in class, noting down all constraints to be taken care of. Furthermore, we decided our next group meeting would be on January 15th, 2022 (Friday) at 5:30, meeting place: Faisal Malik' House. |
| January 15th, 2022 | Meeting at Faisal Place: We discussed about the project objective. Using the Software Management Plan template printed from the web site, we stepped through each section and discussed what was required and what resources were available to us. We also discussed how this prototype should be flexible for other countries. There was constant reference to the "Spice Fusion" and other related articles. |
| January 16th, 2022 | Finished a rough draft prototype and set it up on the online account. |
| January 19th, 2022 | Faisal checked the document of the Software Project Management Plan, and she made some correction marking the corrections in red. |
| January 20th, 2022 | The mistakes were corrected on the web site, and email was sent to Faisal to check the document for any more mistakes |
| January 20th, 2022 | The document was checked by Faisal and few more mistakes were found. These mistakes were corrected and put on the web. |
| January 22th, 2022 | Meeting at Faisal Place: We discussed the Spice Fusion System in more detail and added more information to the SPMP document. |
| January 25th, 2022 | Faisal checked the document of the Software Project Management Plan, and she made some corrections. |
| January 27th, 2022 | The mistakes were corrected on the web site, and email was sent to Faisal to check the document for any more mistakes. |
| January 29th, 2022 | Meeting at Faisal Place: We discussed parts 4 and 5 of the Software Project Management Plan in more detail and decided to update some information in the SPMP document. The different parts of the document were divided between the team for updates. |
| February 3th, 2022 | Finished updating the rough draft prototype and set it up on the online account. Sent all team members email with link to latest copy of the document. |
| February 4th, 2022 | Faisal checked the document of the Software Project Management Plan. The mistakes were corrected on the web site. The latest version of the document is available online. |

***Last Updated on Marach 14th, 2022***

**Software Requirements Specification**

*for*

***Spice Fusion System***

******

***Muhammad Faisal Malik***

***Syed Salman Khursheed***

***M. Maaz Khan***

***Shaghil Arshad***

***Rayyan Mirza***

***March 14, 2022***

|  |  |  |
| --- | --- | --- |
| Version | Changes Made | Date |
| 1.0 | [First Pass for Review](http://www.geocities.com/cs5391/SRS1.htm) | 1/24/2022 |
| 1.2 | [Second Pass for Review](http://www.geocities.com/cs5391/SRS2.htm) | 2/07/2022 |
| 1.3 | [Third Pass for Review](http://www.geocities.com/cs5391/SRS3.htm) | 2/28/2022 |
| 1.4 | RM Review Version | 3/04/2022 |
|  | | |

**Table of Contents**

1. Introduction

2. The General Description

3. Specific Requirements

4. Supporting Information

**1. Introduction**

**1.1 Purpose**

This document describes the software requirements for the Automated Restaurant Order System built for the Restaurant Manger (RM).

**1.2  Scope In**

The RM is requesting proposals to build a prototype of an spice fusion (sf) for their current system. This new This project is fully automated and aims at providing effective management services in Restaurants by controlling information such as Customer details, staff Records, Kitchen Management schedules, Products availability, and Customer invoices.

The system will be designed to provide an electronic version of the Restaurant order management system. The system will have a user-friendly graphical interface and will be more cost effective compared to the current non-electronic version of the Restaurant order management system.

The objectives of this development effort are:

1. To provide existing clerks with a new environment in which to make order for restaurant.
2. To provide an avenue for customers to get their food in a more convenient way.
3. The existing system is manual – largely paper based and lacks standards.
4. To implement a prototype of a scaled down version of the final system to test the solution and further develop requirements.
5. To collect statistics in a more efficient manner for future restaurant.
6. To increase efficiency of food ordering in restaurant.

**1.3  Scope Out**

The following features will not be the part of this Project:

**1.3 Definitions, Acronyms, and Abbreviations.**

APPM – AsiaPac Marketing Manager

CASE – Computer Aided Software Engineering

PP - Project Plan

SFS – Spice Fusion System

SDD - Software Design Description

SRS - Software Requirement Specification

SDS – Software Design Specification

SPMP - Software Project Management Plan

GUI – Graphical User Interface

QAM – Quality Assurance Manager

PDM – Project Development Manager

PMP – Project Management Professional

TBD – To be determined

UML – Unified Modeling Language

**1.4 References**

       Situation Update – Spice Fusion Web App

       <https://spicefusion.org/>

       Pressman, Roger S., *Software Engineering: A Practitioner’s Approach*, McGraw-Hill Companies, Inc., 1997.

**1.5 Overview**

Chapter 2 of the SRS is a brief description of the characteristics of the software to be built, its functions, its users, its constraints and its dependencies.

Chapter 3 is about specific requirements, such as functional requirements, external interface requirements, performance requirements, and also design constraints and quality characteristics.

Finally, chapter 4 includes all the supporting information, such as the Table of Contents, the Appendices, and the Index.

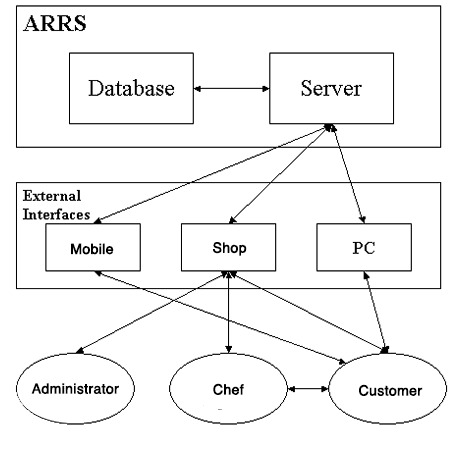
**2. The General Description**

This section describes the general factors that affect the product and its requirements. This section consists of five subsections that follow. This section does not state specific requirements. Each of the subsections makes those requirements easier to understand, it does not specify design or express specific requirements. Such detail is provided in section 3.

**2.1 Product Perspective**

The Automated Spice Fusion System diagram showing the overview of the system’s modules and the relationship of the system to external interfaces is presented in Figure 2.1.

## Figure 2.1 Overview/Architecture Diagram of the SFS



SFS

**Functions of System Components:**

Database:

       Stores data

       Creates reports

       Provides access to data

       Updates information

Server:

       Provides access to the database

       Authenticates users

       Processes orders

       Performs backups

       Produces reports

**External Interfaces:**

Personal Computers

       Users (customers, chefs, front desk, and administration) may use personal computers to obtain a remote access to the server and the reservation database via the Internet.

Cell Phones

       Serve as a medium of accessing the server and the order database.

       Customers may use cell phones and the latest telecommunication technologies to access the web app and the web database via Internet, or they may use cell phones to call admin to inquire about food and price information.

Computer Hardware and Peripheral Equipment to be used:

       30 workstations, which include CPUs, monitors, keyboards, and mice

       Printers

       Network

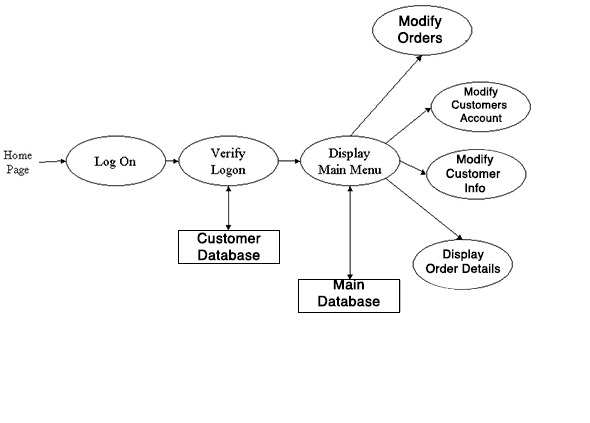
       Cell phones to test connection to the server via remote access

**2.2 Product Functions**

This section provides a summary of the functions that the software will perform.

**2.2.1 Function Relationships**

Figure 2.2 to 2.6 depict the relationships among the functions to be implemented by the system.  
Figure 2.2 SFS General Function Relationship/**Higher Level Usecase** Diagram

****

**2.2.2 Module 1: Make a Menu Item Function (Faisal Malik)**

***Description:***

This function allows the user to

* Creating Food Categories and Sub Categories
* Updating Food Categories and Sub Categories
* Reviewing Food Categories and Sub Categories
* Deleting Food Categories and Sub Categories
* Searching Food Categories and Sub Categories
* Creating MenuItem
* Updating MenuItem
* Reviewing MenuItem
* Deleting MenuItem
* Searching MenuItem

**2.2.3 Module 2: Make a Coupon Function (shaghil)**

***Description:*** This function allows the user to [

* Creating Coupon
* Updating Coupon
* Reviewing Coupon
* Deleting Coupon
* Searching Coupon

**2.2.4 Module 3: Make a order managing Function(Rayyan Mirza)**

***Description:*** This function allows the user to

* Creating Order
* Reviewing Order Confirmation Email
* Searching Order History
* Paying Through Stripe Payment
* Social Login
* Order Confirmation

**2.2.5 Module 4: Make a Payment Function(Maaz)**

***Description:*** This function allows the user to

* Confirming Orders Payments
* Updating Orders Payments
* Reviewing Orders Payments
* Deleting Orders Payments

Searching Orders Payments **2.2.6 Module 5: Add To Cart Function(salman)**

1. ***Description:*** This function allows the user to

* Updating Shopping Cart
* Searching Cart Items

**2.3 User Characteristics**

The main users of the system will be the Customers ordering our product, the chef who are working with us, and the CRM administration that access the reports generated by the system. The users are not required to have knowledge in the computer field. The graphical interface provides an easy way of using the ARRS system with minimum of training.

**2.4 General Constraints**

The constraints for the project are:

* The functional prototype should be available after 30 days upon the arrival of the management team to China. This may prove to be a serious time constraint on the development of a successful prototype.
* Communication with the Chinese team members may prove to be difficult since some Chinese developers do not speak English and the management team does not speak Chinese. Even with the presence of a translator, communication may be difficult. Absence of the translator may severely affect project development.
* Team members are restricted from bringing their own equipment, and insufficient equipment supply may hinder project development.
* Team members are restricted to bringing only the analysts of the team to China. This might affect the project development if more people are needed or the required skills are not available.
* The majority of the Chinese population does not have or have a limited access to the Internet.

**2.5 Assumptions and Dependencies** **or Business Logic**

The assumptions for the project are:

* Ten Orders I first half of the day from Karachi. These order then will be dispatched according to their turns.

       There are types of food as listed below

* + Continental
  + Chinese
  + Italian
  + Desi
* Ordering are assigned according to their turns.
* Ordering involves payment to be paid through stripe payment method or Cod.
* Order List will be provided to riders for each customers. .
* Chinese Ministry will provide us with information about Food and Health measurements and its requirement used in China, so that it can be applied to the our system.
* Network connection will always remain established.

## <ADD OOAD REPORT DIAGRAMS HERE>

**3. Specific Requirements**

This section of the SRS contains design requirements for the Automated Railway Reservation System.

**3.1 Functional Requirements**

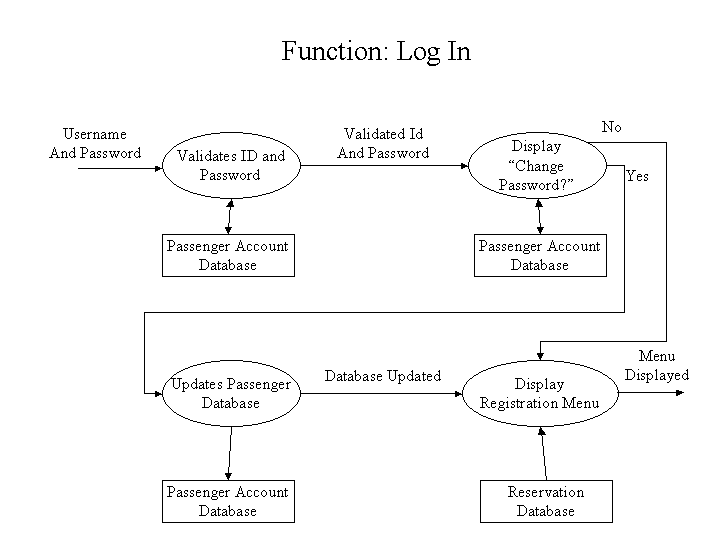
**3.1.1 Log In Function**

1. ***Description:*** This function ensures that only authorized users gain access to the Reservation databases. An authorized user is a user who has an account on the system. Users include passengers, train officials, and CRM ministry officials. The user must type a valid username and password to gain access.

1. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | Allows access to online SFS |
| Inputs | Username, password |
| Source | 1. User inputs username and password 2. Press Login Button |
| Alternate case |  |
| Outputs | Successful login; unsuccessful login |
| Destination | None |
| Precondition | Authorized User |
| Post Condition | No change to Passenger Accounts Database |
| Side Effects | Failures and successful logins are sent to Reservation Database |

1. ***Detailed Use case Diagram for Login: optional***
2. ***Use case Realization for Login: optional***
3. ***Flow of Event or Data Flow Diagram for Login: optional***



1. ***Sequence Diagram for Login: optional***
2. ***Collaboration Diagram for Login: optional***
3. ***Activity Diagram for Login: optional***
4. ***Class Diagram for Login: optional***
5. ***State Chart Diagram for Login: optional***
   * 1. **Module 1 complete CRUD Make a Menu Item Function (Faisal Malik)**

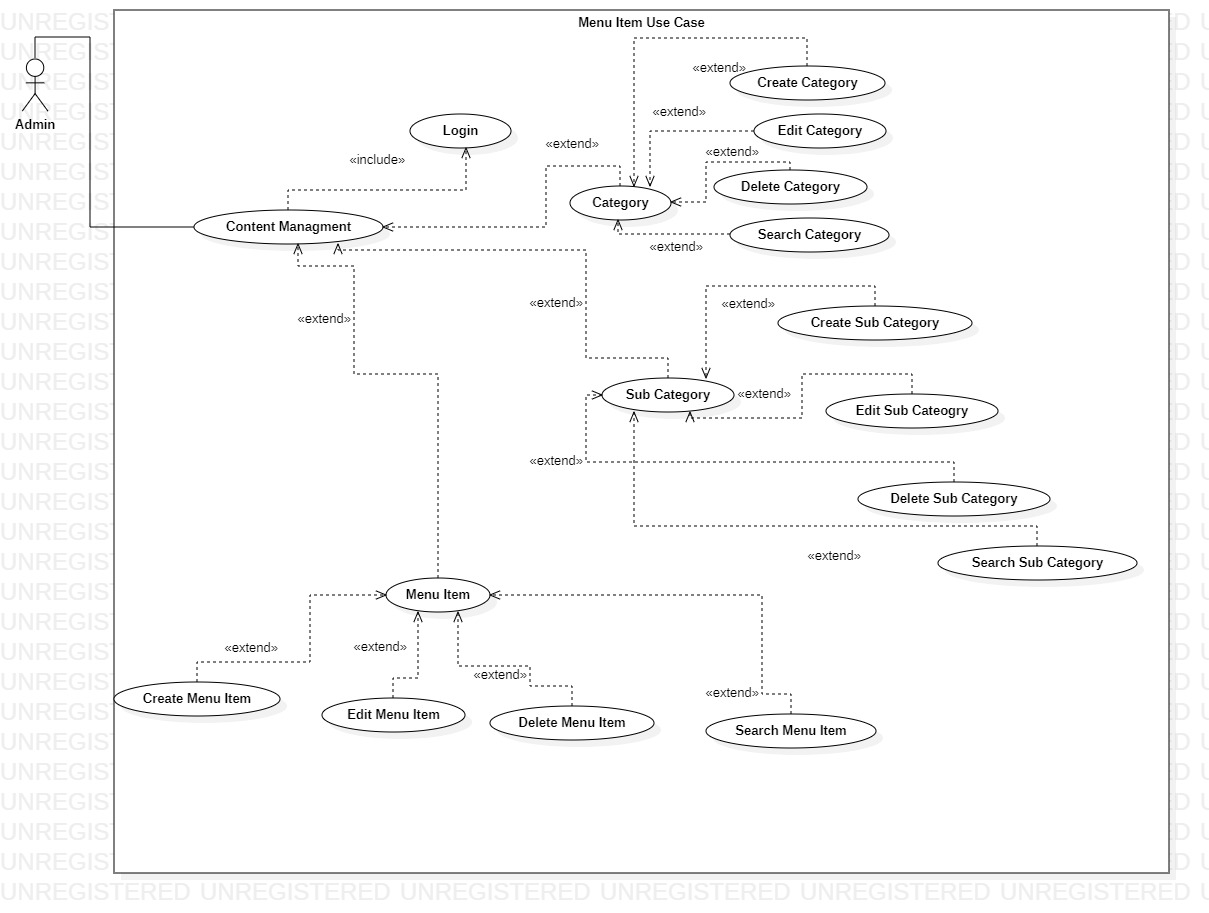
1. ***Description:*** This function allows the user to

* Creating Food Categories and Sub Categories
* Updating Food Categories and Sub Categories
* Reviewing Food Categories and Sub Categories
* Deleting Food Categories and Sub Categories
* Searching Food Categories and Sub Categories
* Creating MenuItem
* Updating MenuItem
* Reviewing MenuItem
* Deleting MenuItem
* Searching MenuItem

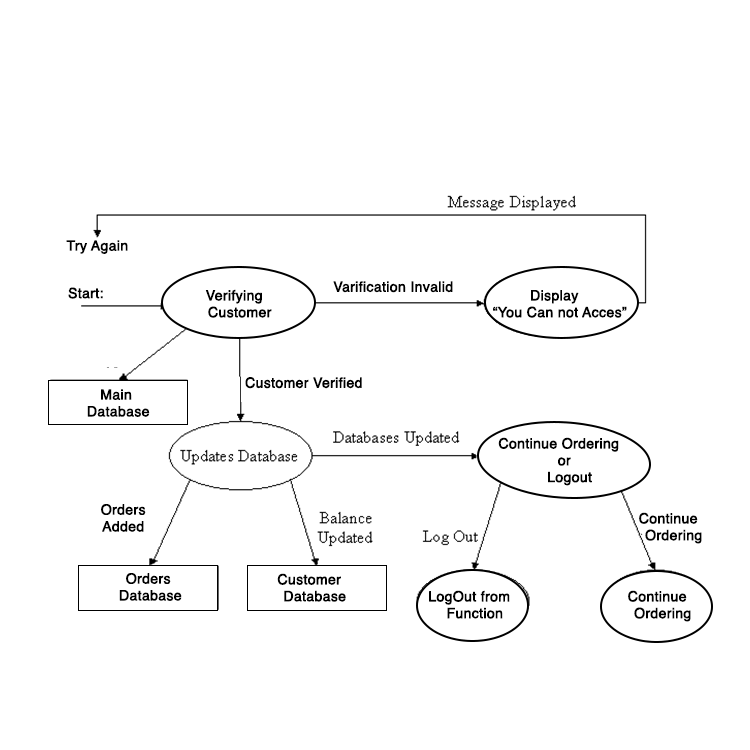
1. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | [ make | drop | view | update] a Menu Item to the Category |
| Inputs | From Menu Item to cart |
| Source | 1. User inputs Menu Item to cart 2. Press Button … |
| Alternate Case |  |
| Outputs | Added | Deleted | Viewed | Modified coupons |
| Destination | Computer screen  main database  customer database |
| Precondition | Valid information: Menu Item no, can not provide Menu Item Without Category. |
| Post Condition | Menu Items added to Categories |
| Side Effects | current Menu Items adjusted  Balance due adjusted |

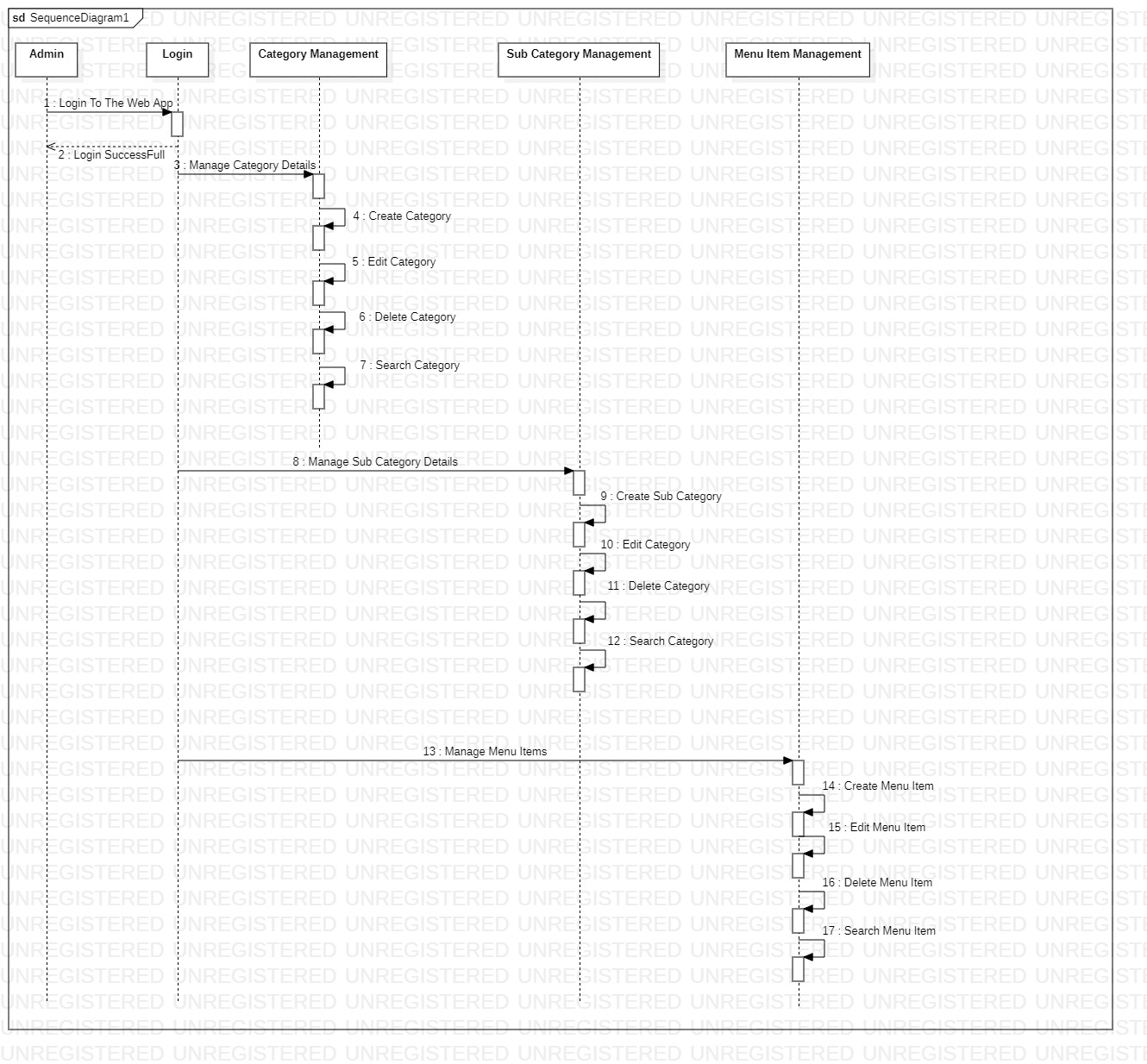
1. ***Use case Diagram:***

******

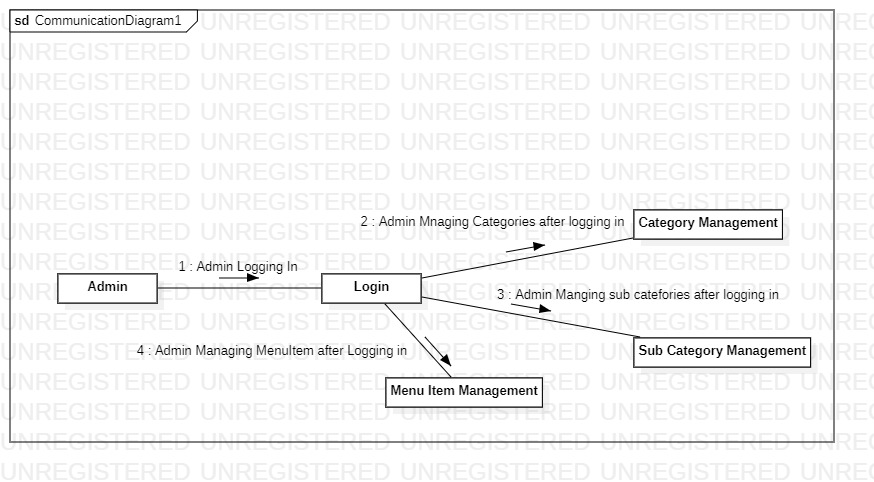
1. ***Flow of Event or Data Flow Diagram:***



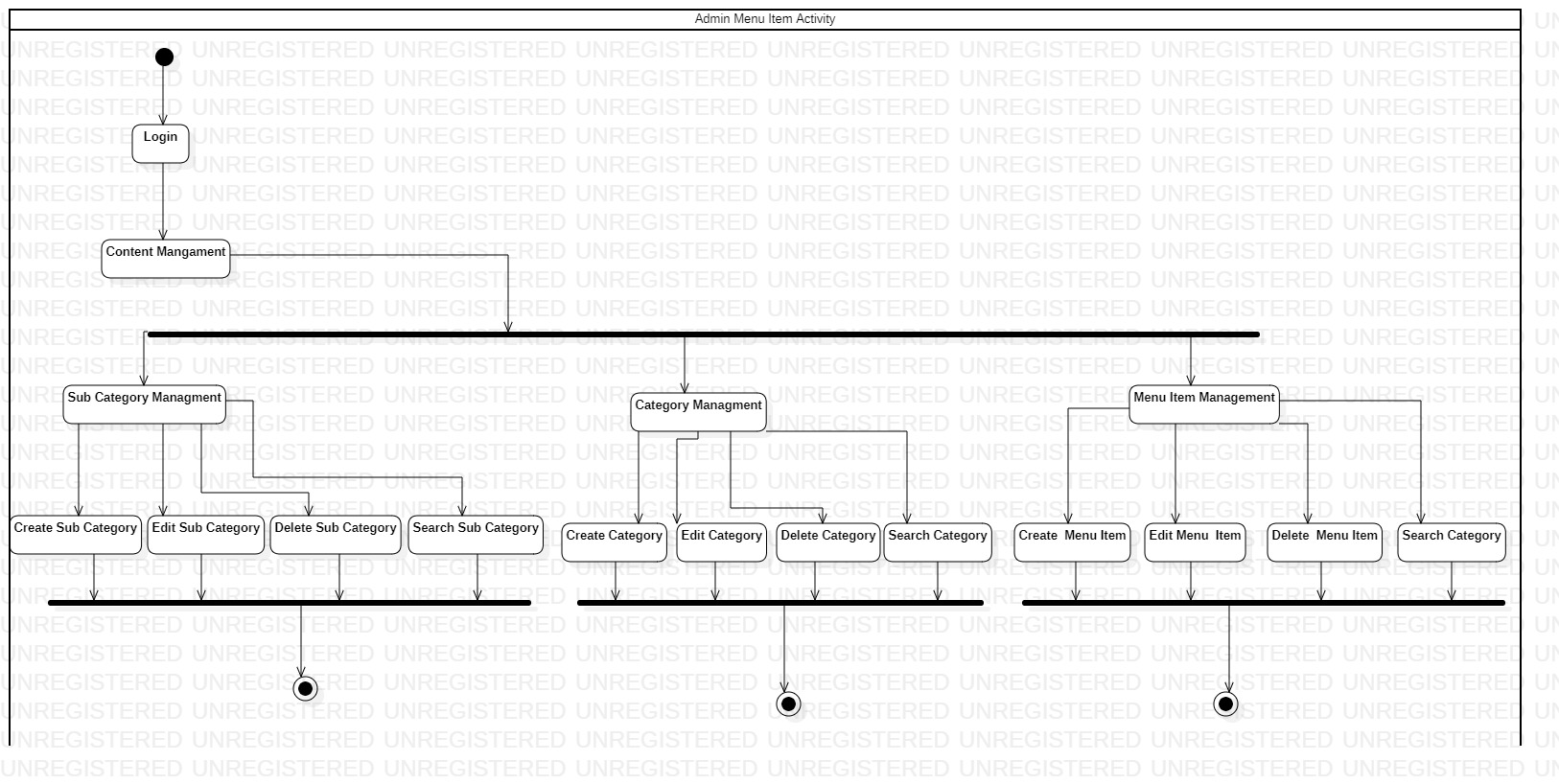
1. ***Sequence Diagram:***



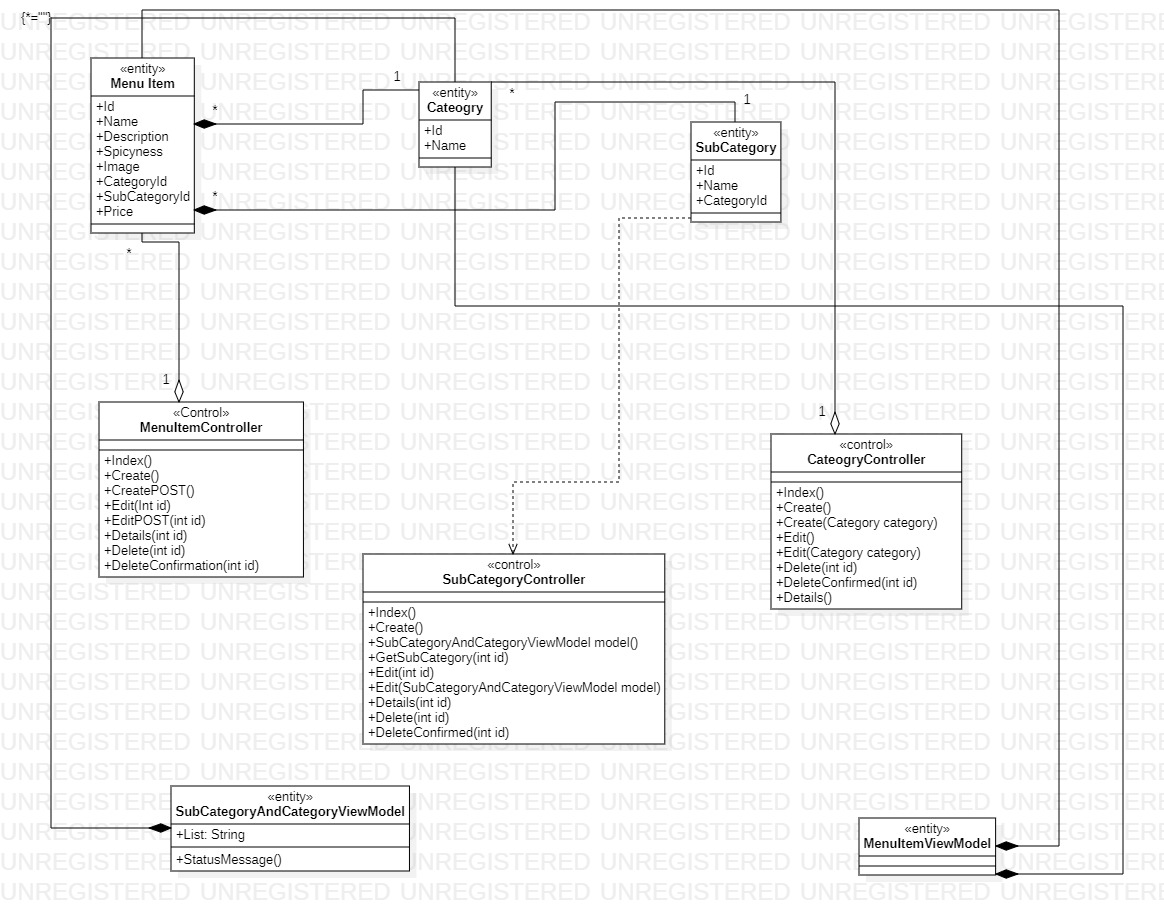
1. ***Collaboration Diagram:***

******

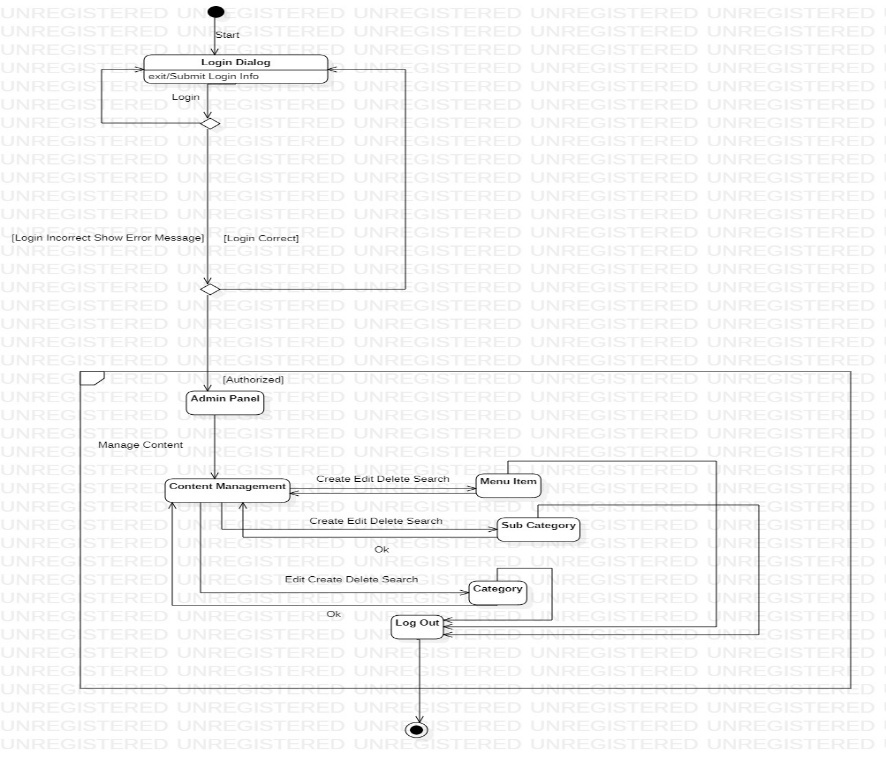
1. ***Activity Diagram:***

******

1. ***Class Diagram:***

******

1. ***State Chart Diagram:***

******

* + 1. **Module 2 complete CRUD Make a Coupon Function (shaghil)**

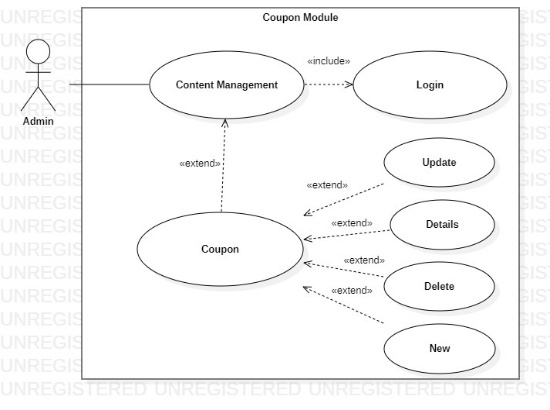
1. ***Description:*** This function allows the user to [

* Creating Coupon
* Updating Coupon
* Reviewing Coupon
* Deleting Coupon
* Searching Coupon

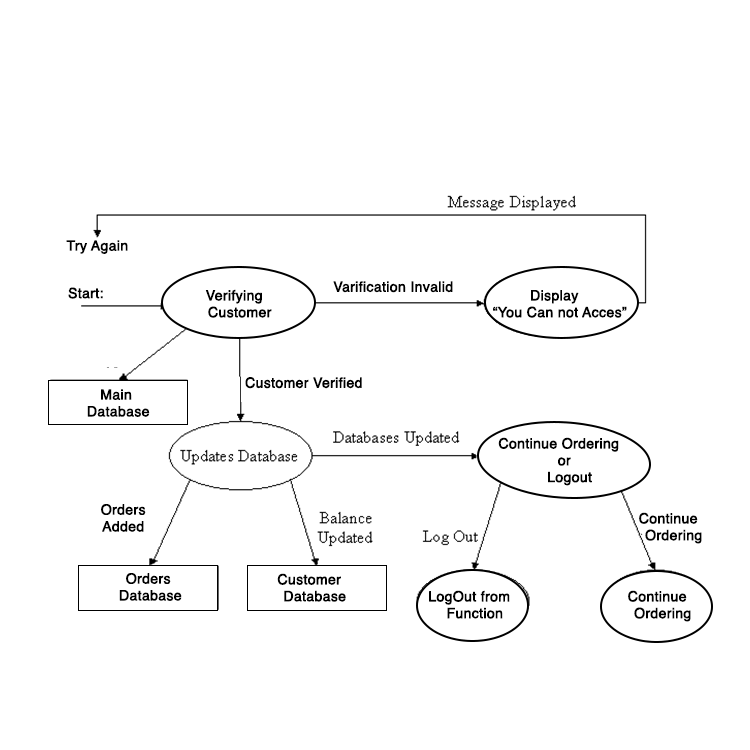
1. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | [ make | drop | view | update] a Coupon to the user’s account |
| Inputs | Coupons Managment |
| Source | 1. User inputs Menu Item to cart 2. Press Button … |
| Alternate Case |  |
| Outputs | Added | Deleted | Viewed | Modified coupons |
| Destination | Computer screen  main database  customer database |
| Precondition | Valid information: coupon no, can not provide expired one. |
| Post Condition | Coupons added to users account |
| Side Effects | User’s current coupons adjusted  Balance due adjusted |

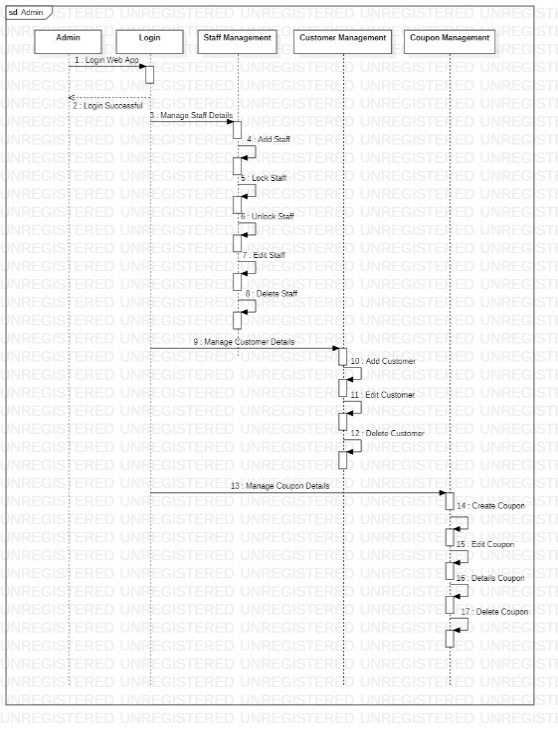
1. ***Use case Diagram:***

******

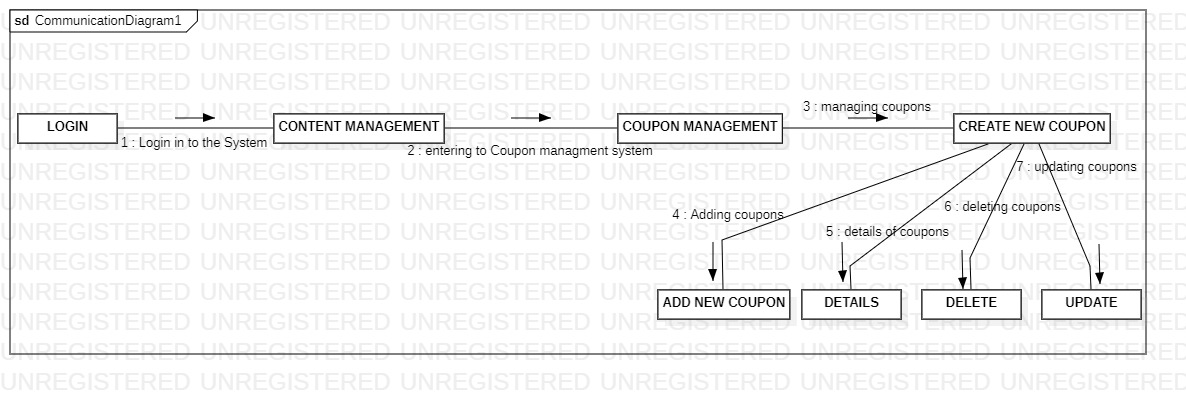
1. ***Flow of Event or Data Flow Diagram:***



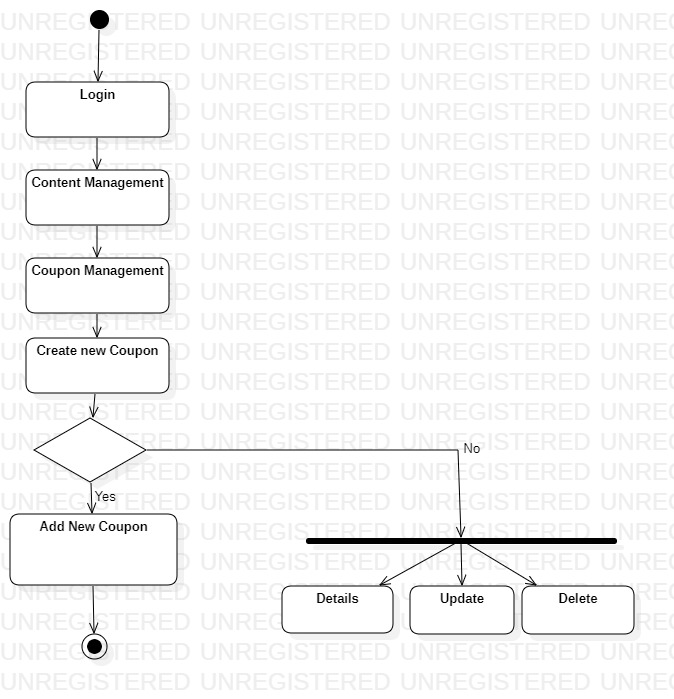
1. ***Sequence Diagram:***



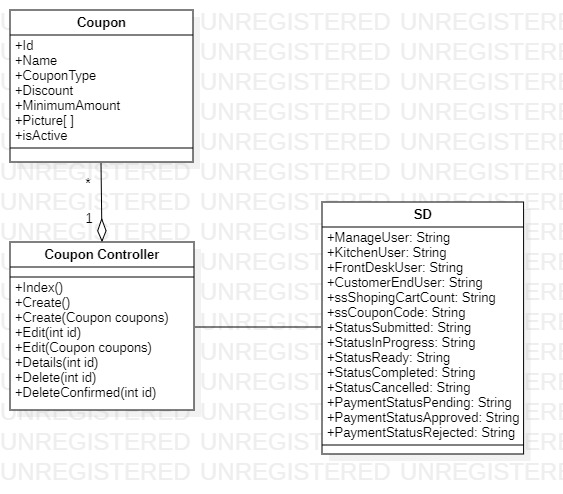
1. ***Collaboration Diagram:***

******

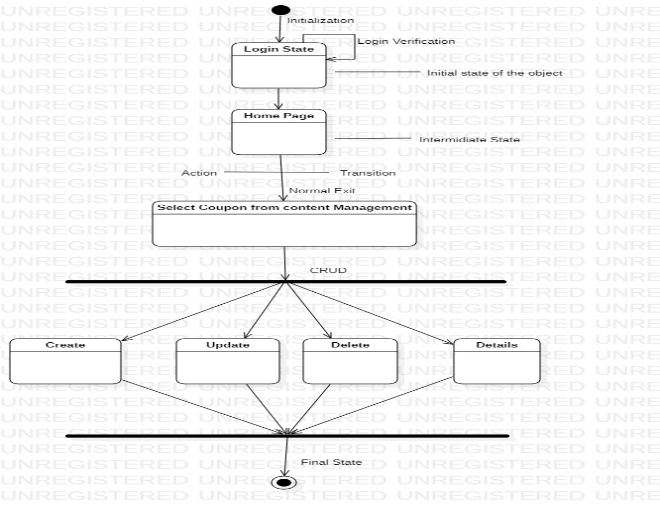
1. ***Activity Diagram:***

******

1. ***Class Diagram:***

******

1. ***State Chart Diagram:***

****

* + 1. **Module 3 complete CRUD Make a Order Managing Function(Rayyan Mirza)**

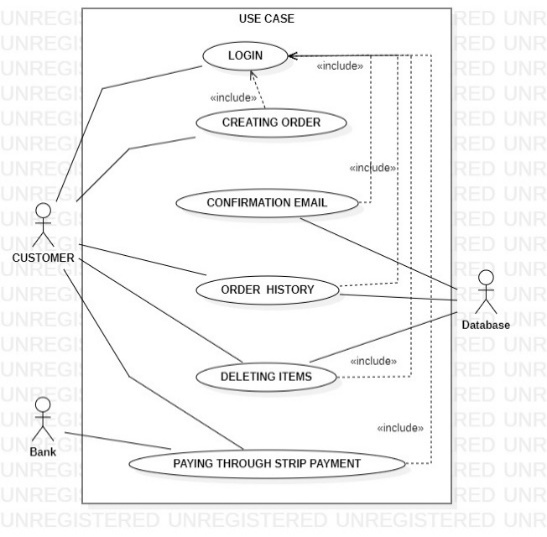
1. ***Description:*** This function allows the user to

* Creating Order
* Reviewing Order Confirmation Email
* Searching Order History
* Paying Through Stripe Payment
* Social Login
* Order Confirmation

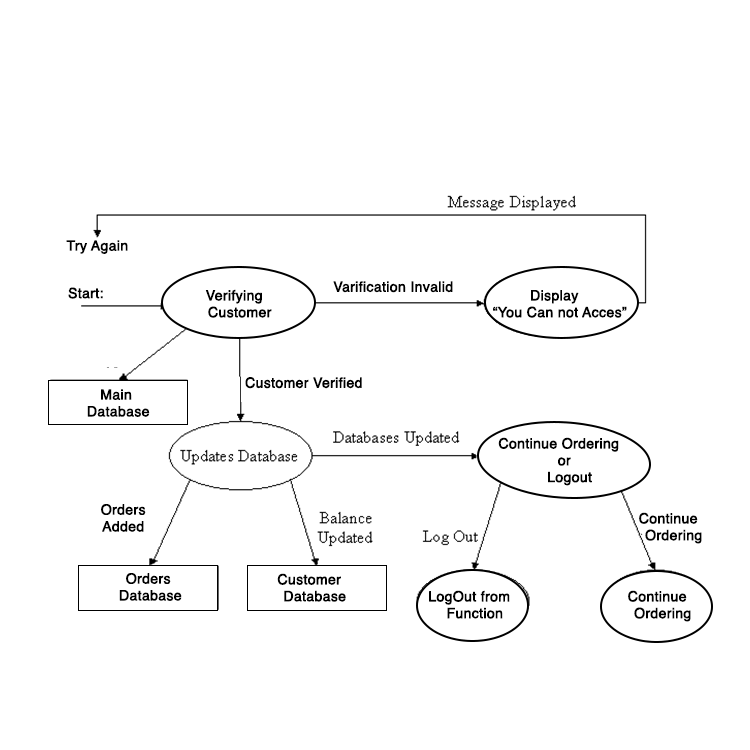
1. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | [ make | drop | view | update] a Orders to the user’s account |
| Inputs | From Menu Item to cart |
| Source | 1. User inputs Cart To Payment 2. Press Button … |
| Alternate Case |  |
| Outputs | Added | Deleted | Viewed | Modified ORDERS |
| Destination | Computer screen  Order database  customer database |
| Precondition | Valid information: order no, can not provide already existing one. |
| Post Condition | Orders added to users account |
| Side Effects | User’s current orders adjusted  Balance due adjusted |

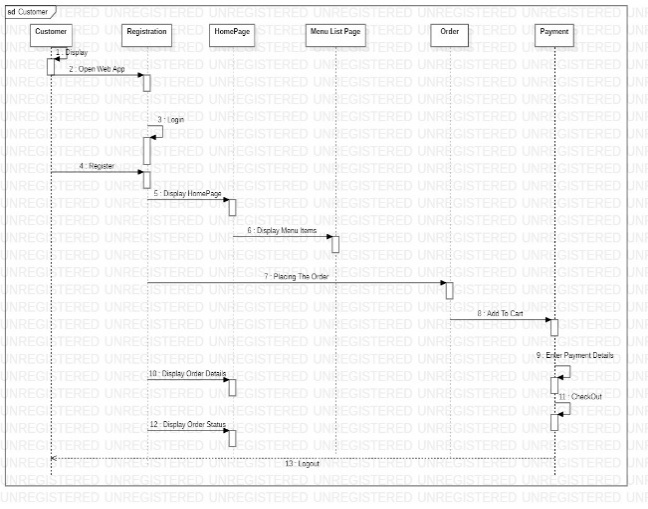
1. ***Use case Diagram:***

******

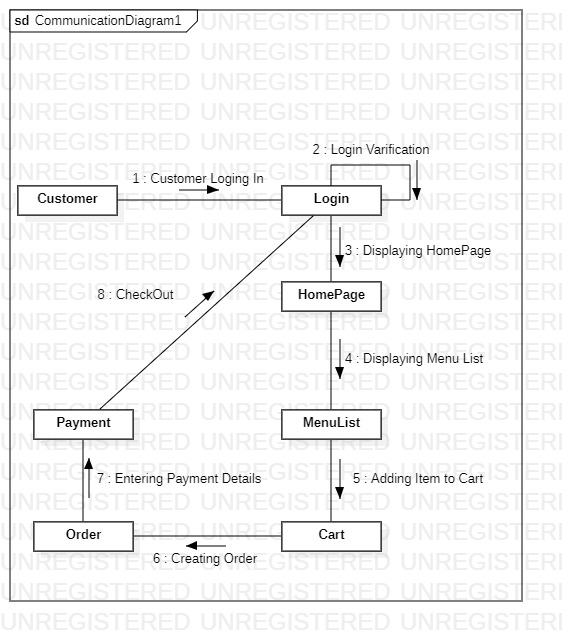
1. ***Flow of Event or Data Flow Diagram:***



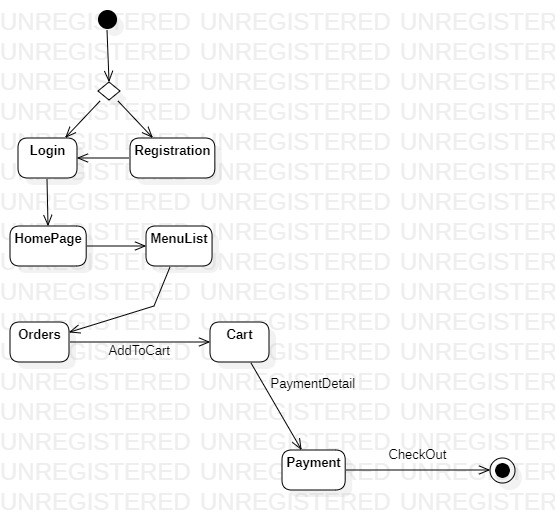
1. ***Sequence Diagram:***



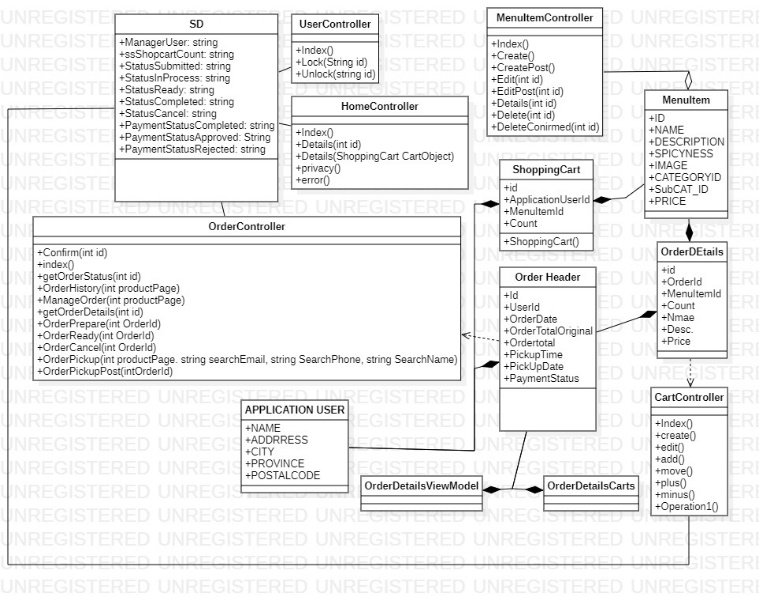
1. ***Collaboration Diagram:***

******

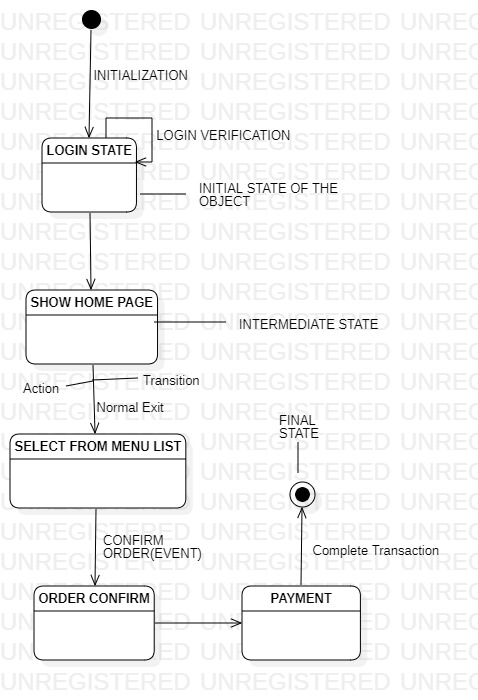
1. ***Activity Diagram:***

******

1. ***Class Diagram:***

******

1. ***State Chart Diagram:***

****

* + 1. **Module 4 complete CRUD Make a Payment Function(Maaz)**

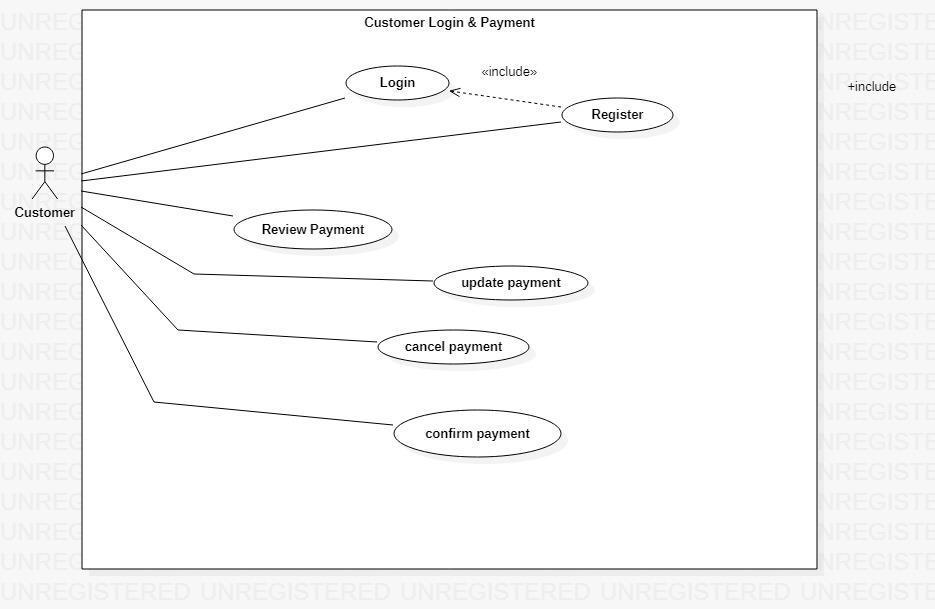
1. ***Description:*** This function allows the user to

* Confirming Orders Payments
* Updating Orders Payments
* Reviewing Orders Payments
* Deleting Orders Payments
* Searching Orders Payments

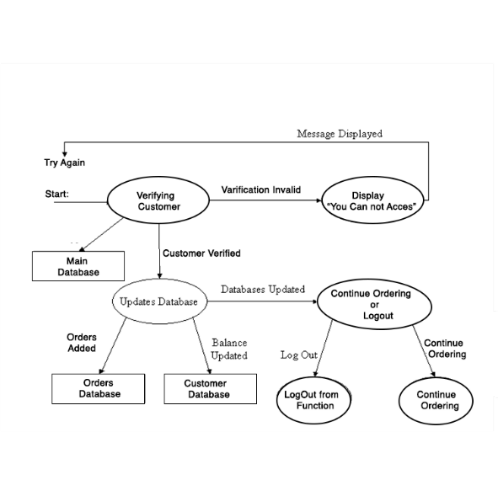
1. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | Customer order profile to the user’s account |
| Inputs | From Cart To Payment Method |
| Source | 1. Stripe Payment Gateway 2. Press Button … |
| Alternate Case |  |
| Outputs | Added | Deleted | Viewed | Modified payment |
| Destination | Computer screen  main database  customer database |
| Precondition | Valid information: user login verified |
| Post Condition | Can not |
| Side Effects | User’s current Payment adjusted  Balance due adjusted |

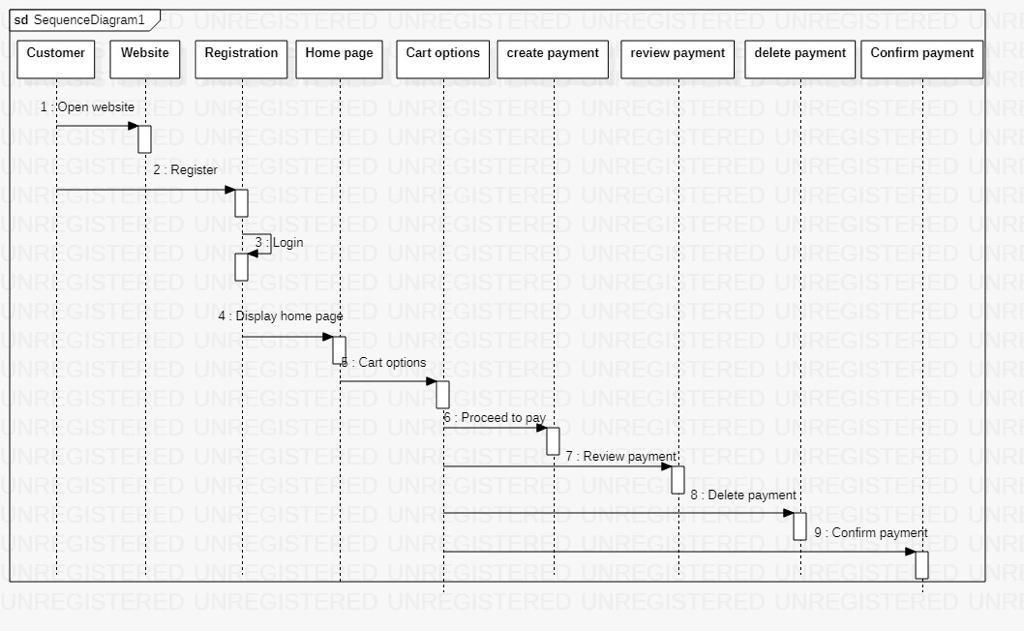
1. ***Use case Diagram:***

******

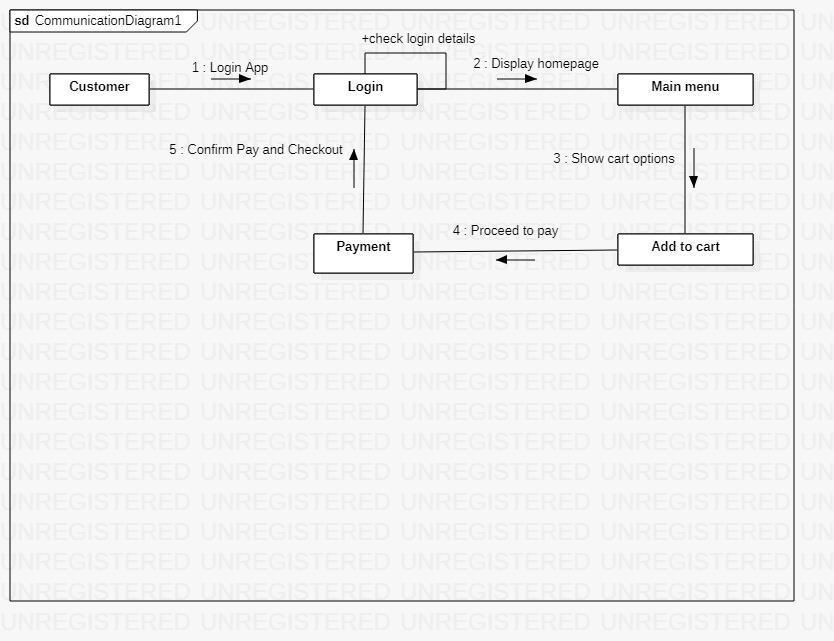
1. ***Flow of Event or Data Flow Diagram:***



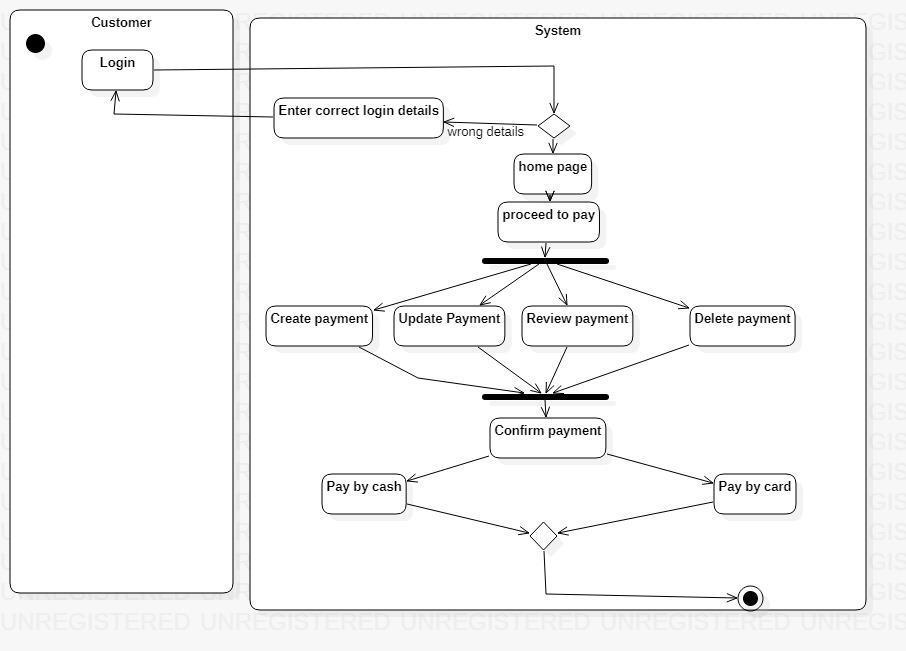
1. ***Sequence Diagram:***



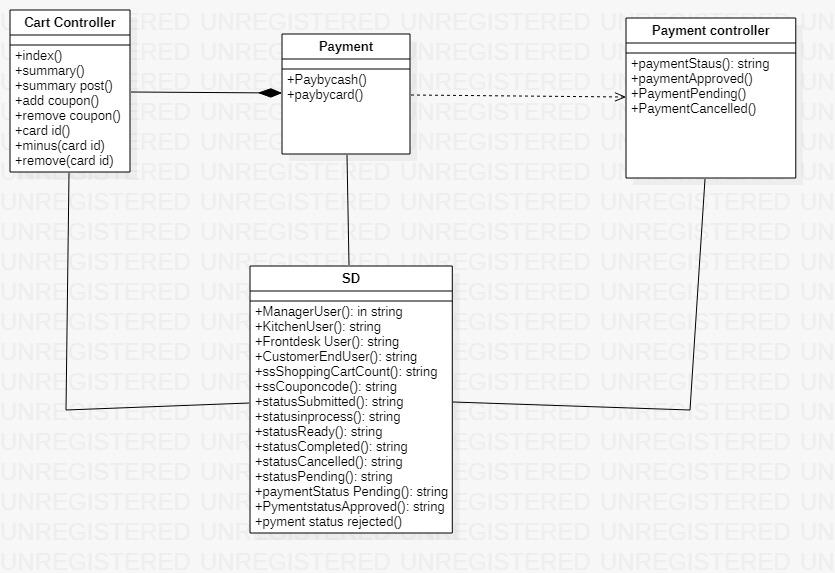
1. ***Collaboration Diagram:***

******

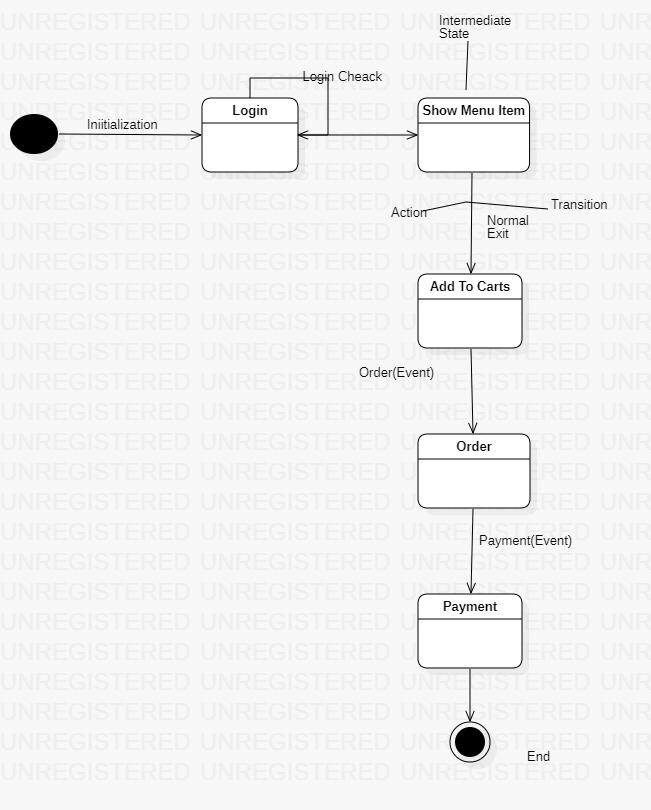
1. ***Activity Diagram:***

******

1. ***Class Diagram:***

******

1. ***State Chart Diagram:***

****

* + 1. **Module 5 complete CRUD Make a Add To Cart Function(Salman)**

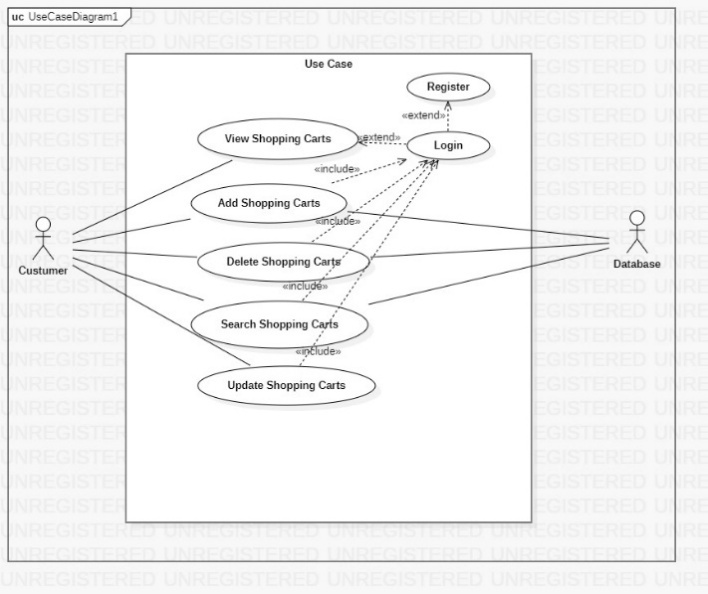
1. ***Description:*** This function allows the user to

* Updating Shopping Cart
* Searching Cart Items

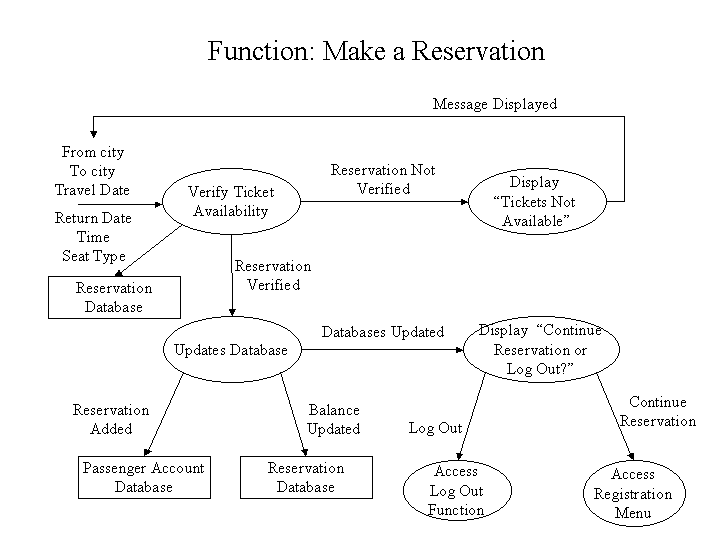
1. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | [ make | drop | view | update] Items From Shopping Cart |
| Inputs | From Confirmation screen |
| Source | 1. Confirmation screen 2. Press Button … |
| Alternate Case |  |
| Outputs | Confirmation showed |
| Destination | Computer screen  main database  customer database |
| Precondition | Valid information |
| Post Condition | Confirmation send to user |
| Side Effects | none |

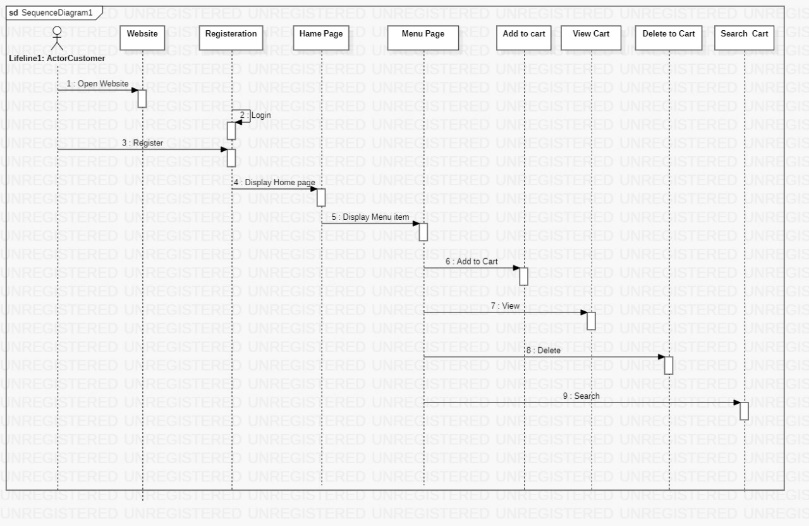
1. ***Use case Diagram:***

******

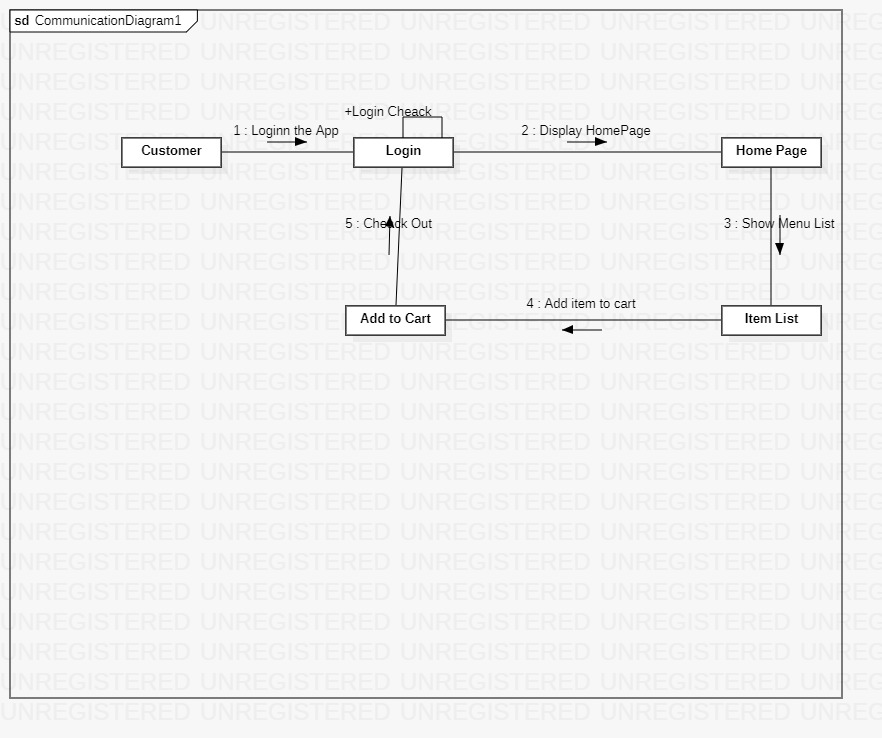
1. ***Flow of Event or Data Flow Diagram:***



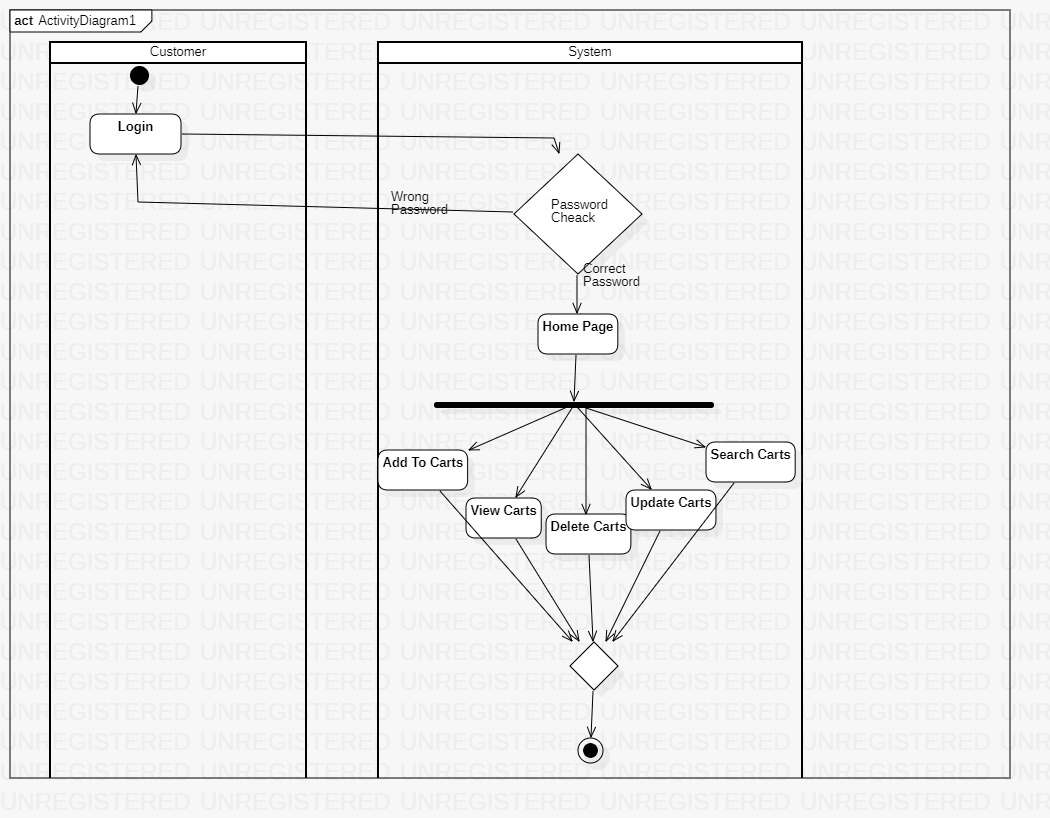
1. ***Sequence Diagram:***



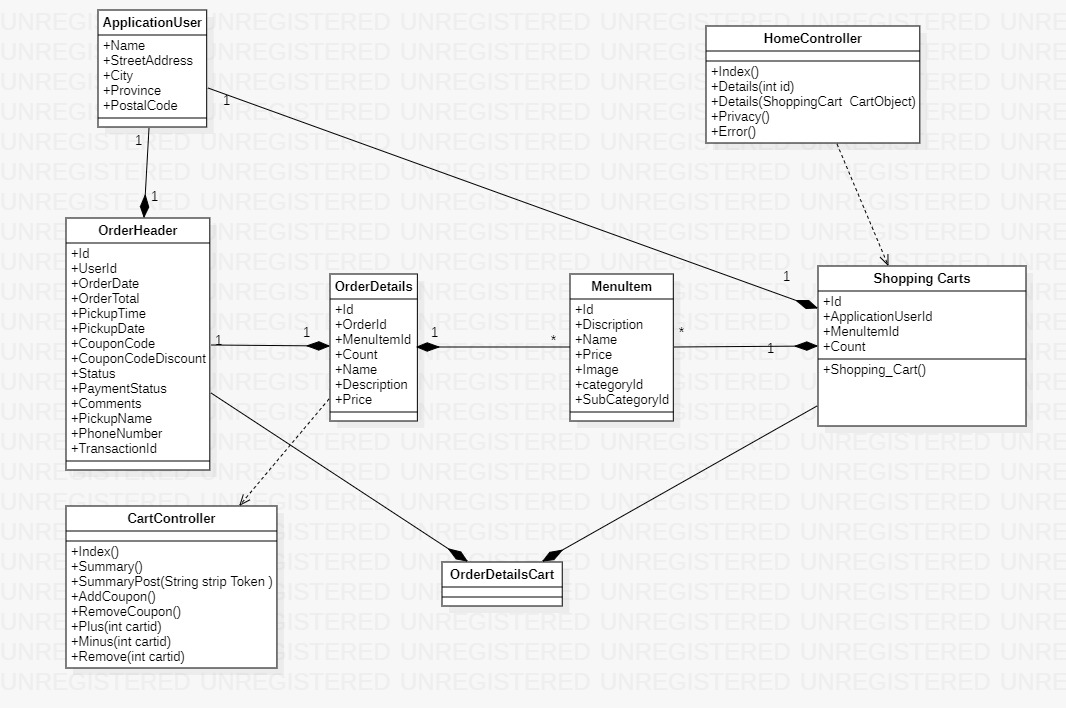
1. ***Collaboration Diagram:***

******

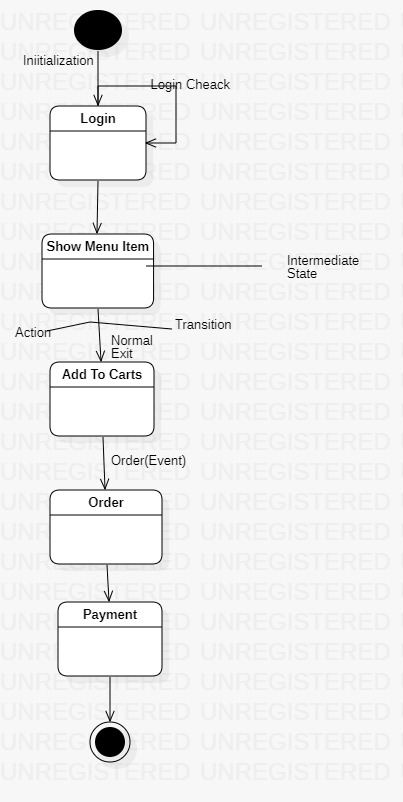
1. ***Activity Diagram:***

******

1. ***Class Diagram:***

******

1. ***State Chart Diagram:***

****

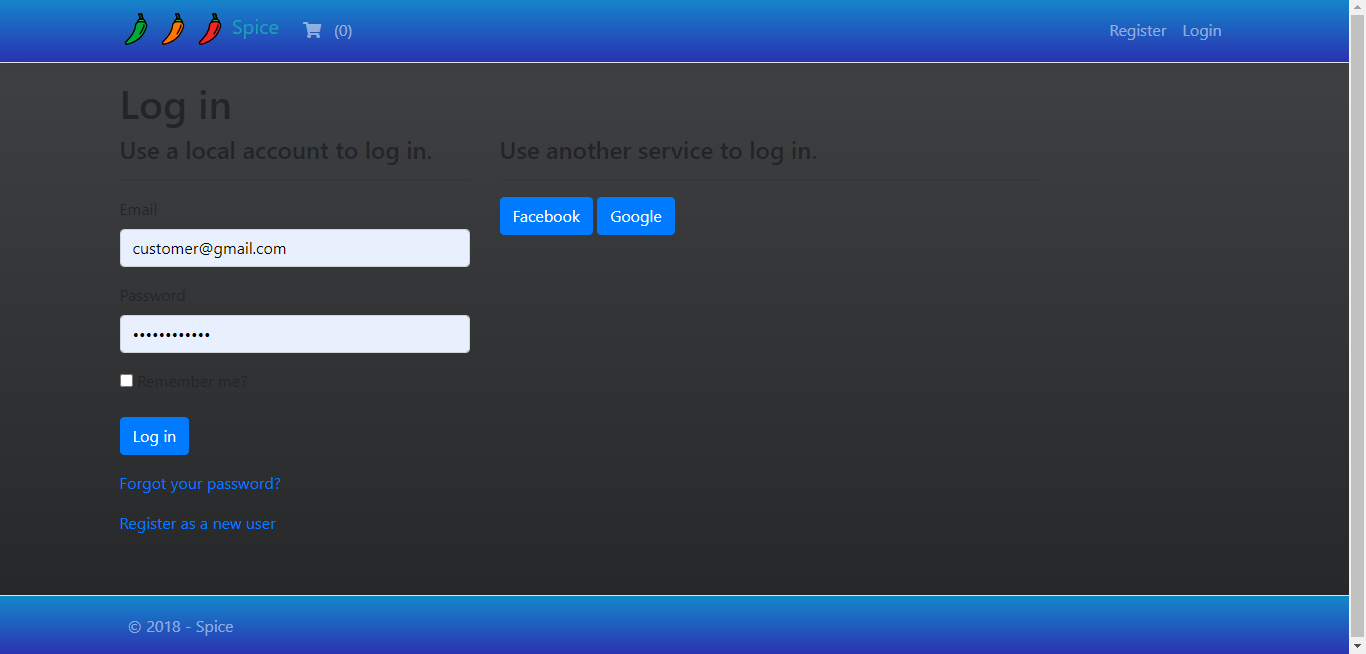
**3.2. External Interface Requirements**

**3.2.1 User Interfaces**

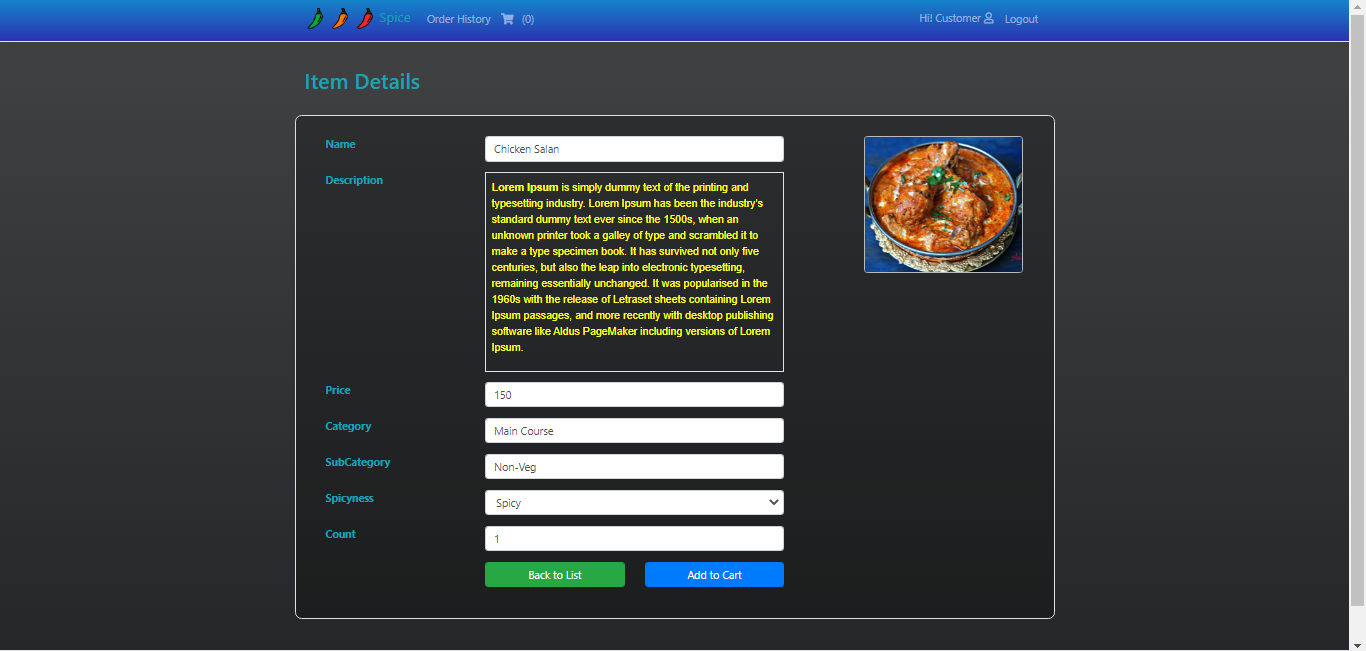
The user interfaces are divided into two major components. One part includes the user accessing the system using a cell phone. The other portion involves accessing the system through a remote site or at a particular location specifically designed to access the system. For instance, the clerks and the CRM access the reservation system from the reservation or CRM office.

The diagrams and explanations below demonstrate the major transition from one user interface to another. This is a brief description. However, a more detailed demonstration is done in the prototype. The purpose of this interaction is to illustrate the overall view of the SRS.

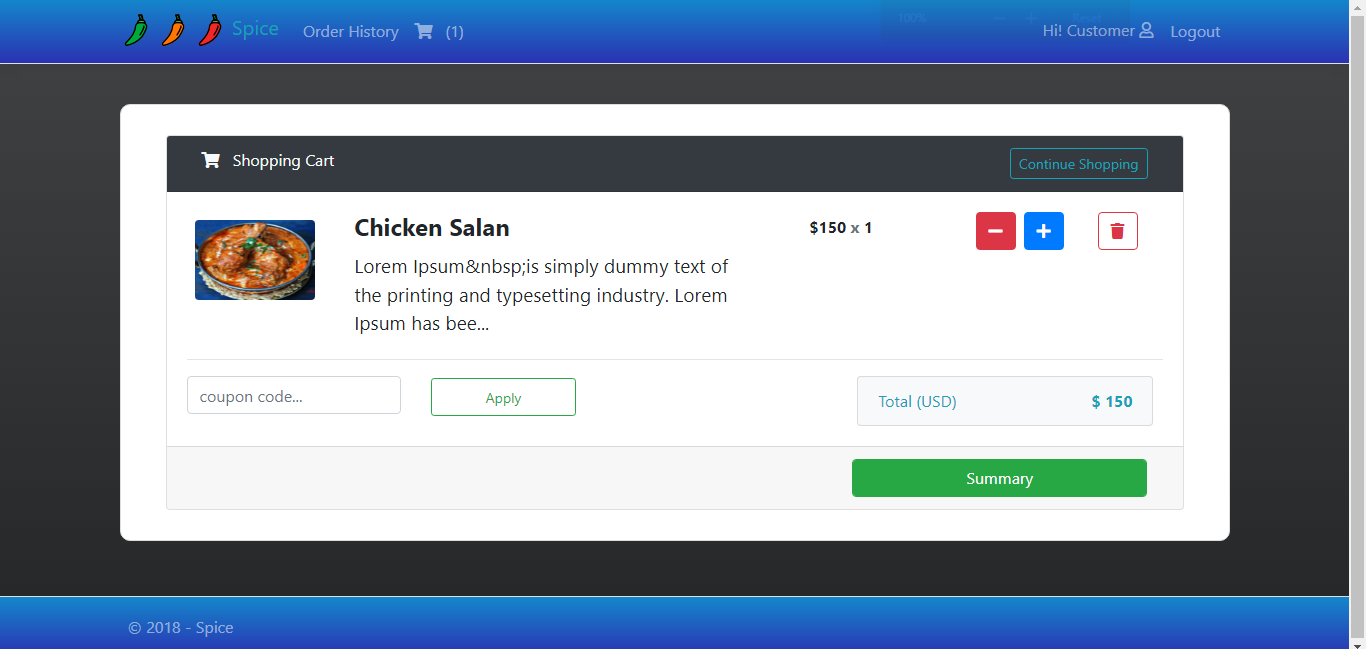
The diagram below illustrates the **major functionalities or modules** . These functionalities will be displayed depending on the user. For instance, we will see all our functionalities while the normal user and the clerks will only see the functions that we are wanting them to see.



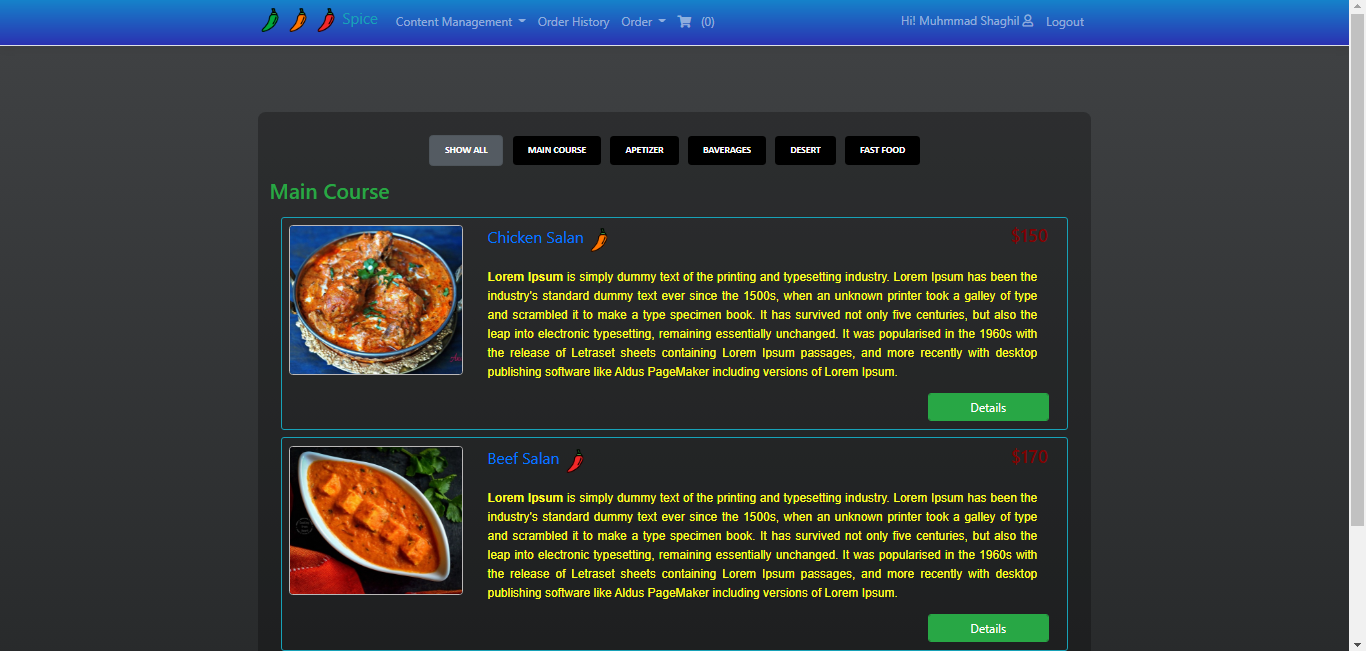
Selecting one of these functions will take the user to a different user interface. For instance, choosing this will display the all the content for individual users. The title of this page is consistent with the function selected, and since the LoginRole was selected, the title displays LoginRoleName. The purpose of this is to allow the user know what part of the system they are accessing. Furthermore, the user can select any of the functions.



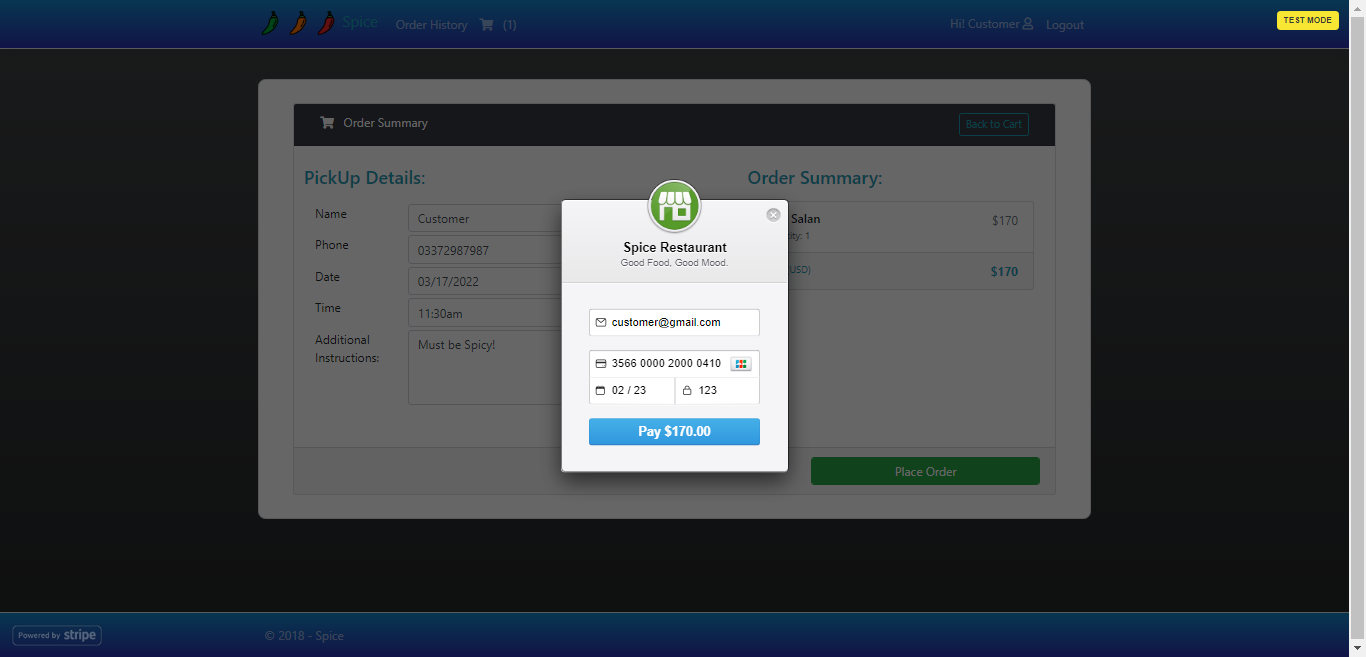
The user can select any of our functionalities. For the sake of this demonstration, if the user clicks on the Add to cart function the diagram below is displayed. Once again the title is the same as the main function and a subtitle indicates the second function selected. The buttons on screen allows the user to navigate through the interfaces. For instance, the back button will take the user to the above page.



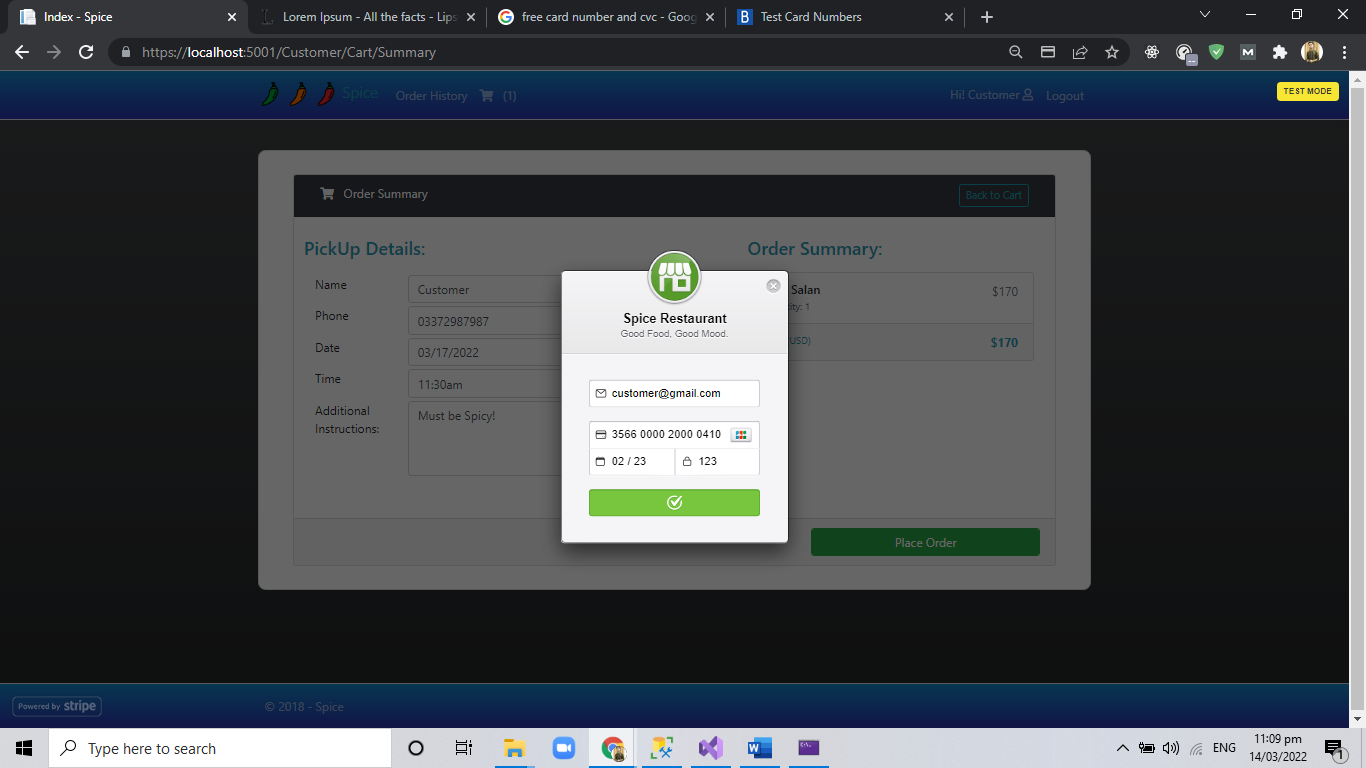
The Display Available function displays all the orders that are available and are ready to dispatch from our point to your point. Furthermore, the order details are also mentioned. The detail button will take the user to more details and the confirm button takes the person to the payment page.



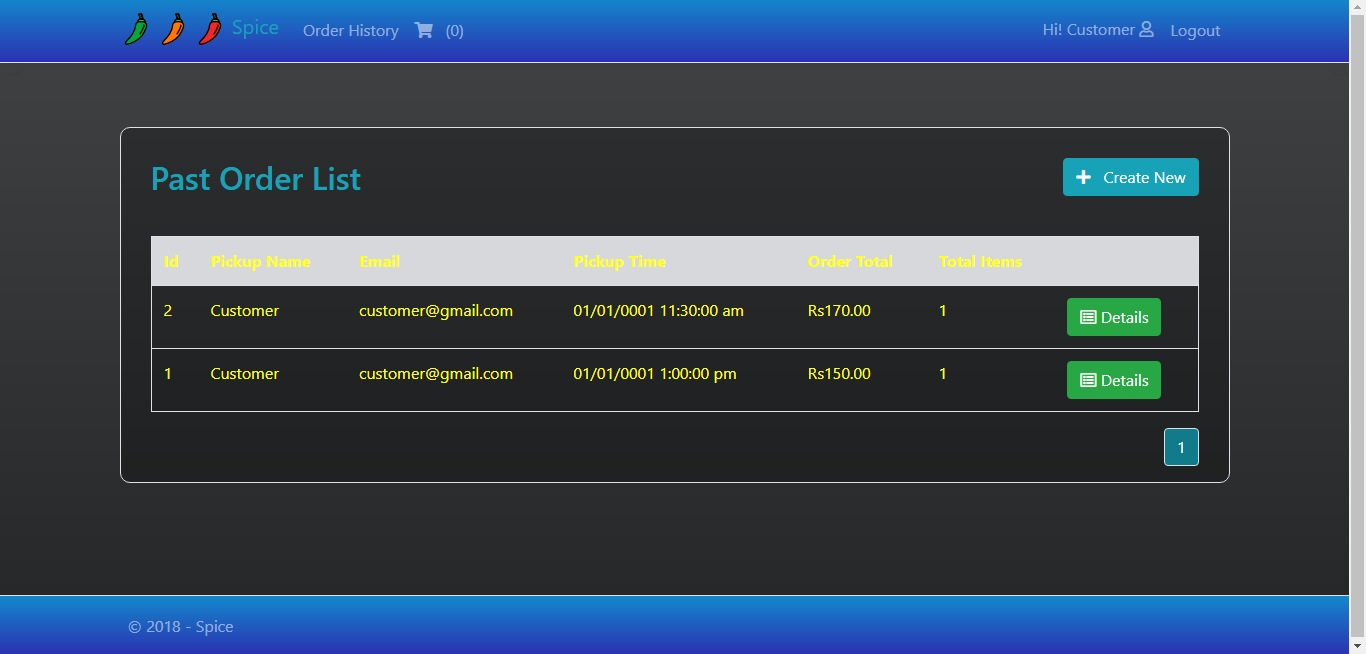
The following page allows the user to pay for the order as appropriate. Now, this page is part of the Payment Stripe function, and it is used here to make payment for the orders selected. This makes it easier for the user since they do not have to go back to the main menu and to access their account.



Finally, the submit button displays the appreciation page as shown below with a button to go to the main menu.



The above illustration has shown a brief overview of the user interfaces involved for the normal and clerk users. However, the CRM have specifically requested a number of reports, Therefore, the CRM interface is able to access all our functionalities as shown in the main menu (first diagram). Once the CRM selects the Orders function, a list of two reports is displayed as linked list. This is shown in the diagram below:



The orders selected here shows the number of orders that are being dispatched. This report indicates the major traffic flow, The diagram below shows the report format to be displayed.



As mentioned earlier, the system can also be accessed through the wireless phones. In that case, the overall system will be the same as the above presentation except that the format will be simplified, since the phones do not have graphic support. The phones will have access to the Spice Fusion Accounts, however it is difficult to display the reports

**3.2.2 Hardware Interfaces**

The SRS includes two major hardware components: cellular phones and regular PC's. The cell phones require WAP (wireless application protocol) network protocol, which is already programmed in the latest phones.

The second component involves the regular PC’s, which communicate with the server. The server then communicates with the database. The protocol involved between the PC's and the server is the HTTP protocol, which allows communication between the PC's and the Server. The remote PC's, such as someone accessing from home using the Internet, are able access the information through the CGI. The requests come in through the HTTP protocol,

**3.2.3 Software Interfaces**

An SQL DBMS will be used to manage the database and any changes made to it. Furthermore, the DBMS will make regular backups of the database and generate reports regularly so that they can be accessed by the CRM. The Apache server between the client and the database will handle all communication, and the server will run on a Linux operating system. Furthermore, the HTML pages must be implemented such that they can be displayed on two common browsers: Netscape and Internet Explorer.

Information about the products used for the SFS:

(1) Name: Oracle

(2) Mnemonic: Oracle

(3) Version Number: ?

(4) Source: Oracle

(1) Name: Linux

(2) Mnemonic: Linux

(3) Version Number: 6.2

(4) Source: Unix

(1) Name: Internet Explorer

(2) Mnemonic: IE

(3) Version Number: 5.00

(4) Source: Microsoft

(1) Name: Apache

(2) Mnemonic: Apache

(3) Version Number: 1.3.14

(4) Source: Apache Software Foundation

**3.3 Performance Requirements**

The following sections list the performance requirements for the system.

**3.3.1 User Requirements**

|  |  |
| --- | --- |
| User Requirements | **Description of Requirement For Design Environment** |
| Location(s) and Number(s) of Users | Karachi,Lahore,Islamabad |
| Expected Growth in Number of Users |  |
| After 1 Year | 50% |
| After 2 Years | TBD |
| After 3 Years | TBD |
| User Expectation |  |
| Interactivity | User expect that it provides a very easy to use graphical user interface |
| Reliability | For some applications, reliability must be 100% during the application session |
| Adaptability | Network must adapt to user additions, deletions and changes |
| Security | Encryption software would be used for Credit Card transactions |
| Cost / Funding | Less than $250K |

**3.3.2 Application Requirements**

Since no specified service is indicated, then we have listed the applications as best – efforts. This may change as we learn more about the application.

The communication package is determined to be burst in nature, with small data sizes and frequent transmissions. We can consider this application to be interactive-burst, while the database transaction-processing application is described by the CRM as transferring large amounts of data (initial estimates are 1 MB/transaction), we have listed this application as interactive-bulk.

|  |  |  |
| --- | --- | --- |
| Categorizing  Applications | Best-Efforts | Application  Locations |
| Communication | 100 Kb/s | Karachi and LAHORE |
| Database Access | 400 Kb/s | All Locations |
| Database Transaction processing | 1.5 Mb/s | All Locations |

**3.3.3 Host Requirements**

|  |  |  |
| --- | --- | --- |
|  | Type of Host or  Equipment | Numbers and  Locations |
| Host A | PC | Karachi (10), Lahore(7), Islamabad(10) |
| Host B | Database Server | Karachi |
| Host C | Application Server | Karachi |

**3.4.1 Standards Compliance**

There are no design constraints that can be imposed by other standards limitations.

**3.4.2 Software Limitations**

·        must be able to run Internet Explorer or Netscape Communicator web browsers to access the system.

**3.4.3 Hardware Limitations**

·        Input/Output: One or two-button mouse, keyboard, cell-phone, or touch screen required.

·        Network card required at thin-client terminals to make communication with server possible.

**3.5 Quality Characteristics**

There are a number of quality characteristics that apply to the SRS software system.

**3.5.1 Portability**

The SFS system will be developed using MVC5 AND .net core so that it can be accessed from any type of system using just a regular web browser.

**3.5.2 Reliability**

The system should be capable of processing a given number of orders within a give time frame with no errors and the system should be available and operational all the time. During the development of the prototype for the 3 cities, the system will be tested in its actual environment to ensure that it can handle the load of orders that occur during a regular workday.

**3.5.3 Usability**

The SFS system will be developed so that it is an easy to use system that requires the least amount of user input possible. Every input will be validated. The user should only have general computer use knowledge. Error messages will be displayed if the user enters an invalid value or tries to access a function without the required permissions. An easy and well-structured user manual will be provided to the CRM and the system will include descriptive help for all operations allowed.

**3.5.4 Correctness**

The SFS system will be considered correct when the CRM approves the prototype presented and agrees that all the functions they require are implemented as stated in the Software Requirements Specification.

**3.5.5 Flexibility**

The SFS system should be developed in such a way that it is easily customizable. If new functions are required by CRM, there will be little effort required to update the system to support new cities or new transactions.

**3.5.6 Security**

The SFS system should not compromise the customer information at any time. The user information will never be sold to other parties and will be kept secure at all times. Users will be authenticated to ensure that no unauthorized users gain access to private information.

**3.5.7 Maintainability**

The SFS source code will be kept well structure and documented so that it is easier to maintain and extend the system. All changes to the system shall be documented.

**3.6 Other Requirements**

Certain requirements may, due to the nature of the software, the user organization, etc., be placed in separate categories such as those below.

**3.6.1 Data Base**

The Spice fusion System will have one main database .The database will be created on SQL server. The following are the requirements for the database that is to be developed as part of the product. They include:

**Reservation Database**

|  |  |
| --- | --- |
| Types of information | Schedule information for the trains, including date, time, departure city, destination city, ticket cost and ticket availability for a particular train |
| Frequency of use | Depends on the Customer demand, which may reach 5,000 per day during peak periods |
| Accessing capabilities | The database should allow access to at least 1,000 people at once; the users will have a general access to the information about the orders, and a secure access to the content (available only to CRM officials) using a username and a password |
| Data element and file descriptions | To be determined |
| Relationship of data elements, records and files | To be determined |
| Static and dynamic organization | To be determined |
| Retention requirements for data | Previous order information will be available as long as the order for a particular person is in use and at least one year after the order has been cancelled. The reports information will be available at least for 5 years |

**Customer Account Database**

|  |  |
| --- | --- |
| Types of information | Passenger account information including their name, address, phone numbers, last reservations, balance owed, credit card number (if they paid by a credit card) |
| Frequency of use | Depends on the Customer demand, which may reach 5,000 per day during peak periods |
| Accessing capabilities | The database should allow access to at least 500 people at once; the users will have a secure access to the database using a username and a password |

|  |  |
| --- | --- |
| Data element and file descriptions | To be determined |
| Relationship of data elements, records and files | To be determined |
| Static and dynamic organization | To be determined |
| Retention requirements for data | Customer account will be available for as long as a Customer is using the account, and at least for 6 month since the Customer logged on last time. |

**3.6.2 Operations**

The normal operations required by the user can be viewed as the following:

User-initiated Operations:

These operations include the login operation, which is initiated by the users. Also, the process of becoming a new user is in this category. Building, changing, and viewing itineraries, as well as paying for the itinerary are all initiated by the users. The user initiates the report generation activity.

Interactive Operations and Unattended Operations:

The users initiate all the operations mentioned above, and almost all of them are somehow interactive. Displaying the Food schedule is non-interactive. The report display is a non-interactive operation, although selecting the desired reports will require user input.

Data Processing Support Functions:

The user account data is used to create new accounts, as well as to validate user id's during login functions. For building itineraries, user input, user account data, and food schedule data are used, and processed. User data along with final results of user interaction (whether the user purchased a food, number of items bought, etc.) are collected, and used for report generation purposes. Administrative users' inputs are collected in order to modify and present schedules.

Backup and Recovery Operations: database used is production database. The main operation used for the backup and recovery is SQL server built-in cold backup, which is also known as the "archive mode". Depending on the customer's needs and budget, additional redundancy can be added using systems like RAID 5 and tape backup.   **3.6.3 Site Adaptation Requirements**

There are no site adaptation requirements for this project.

**4. Supporting Information.**

There is no supporting information required for this project