LIVELY LOCAL MARKET

UCS503 Software Engineering Project Report

End-Semester Evaluation

Submitted by:

101803062 Muskan Goel

101803670 Nikhil Bansal

101803671 Kartik Sharma

101803672 Kushagar Sharma

Submitted to:

Sawinder Kaur

Computer Science and Engineering Department TIET, Patiala

November 2020

TABLE OF CONTENTS

Contents Page No

1. Project Overview

The central concept of the application is to allow the customers to view the nearby shops and area which would help them to get familiar with the locality they are living in. The user can view any shop, its products and also the location of the shop.

Depending on the services provided by the shop, the user can also order any item needed if the home delivery system is available.

The application is designed into two modules. First for the customers who wish to purchase and view the articles and second for the shopkeepers who update the information about their shops and availability of their products and services.

The end user of this application is the user who uses this application to find the nearby departmental stores, restaurants, and other places and the shopkeepers who update their information.

The application deployed at the customer database contains the details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction.

Software Requirements Specification

for

Lively Local Market

Version 1.0 approved

Prepared by
Kartik Sharma
Kushagar Sharma
Muskan Goel
Nikhil Bansal

Submitted To: Sawinder Kaur
Group 5 (TIET)
September 2020

Table of Contents

Revision History

Name	Date	Reason for Changes	Version

1. Introduction

This project will give the users an idea about their locality that would help them to meet their basic necessities without much effort.

1.1 Purpose

The major purpose of this project is to

- Create a review system for the local market of the user's area.
- Contain the information of the different kinds of shops or vendors in that locality.
- The user can give ratings to a particular shop/vendor or can even add any new shop that does not currently exist in the database.

1.2 Document Conventions

The followings are the list of conventions and acronyms used in this document and project a well are:

- Main headings – **Bold** (user can easily distinguish it from other subheadings)
- Font size Heading 1(16), Heading 2(14), Heading 3(12)
- Bullet points (use to identify special information about the software)
- *Italicized* text is used to label and recognize diagrams
- The font of document is Times New Roman
- **User:** A general login is assigned to users
- **Admin:** A login id representing a user with user administration privileges to the software

This document was organized in accordance with the Software Requirements Specification template of the IEEE.

1.3 Intended Audience and Reading Suggestions

This software Requirement document is intended for the project managers, developers who can review the project's capabilities and more easily understand where their efforts should be targeted to improve or add features to it. Project testers can use this as a base for their testing strategy. End users of this application who wish to read about this project can do. We could read it in a front to end manner but part by part information has been given where each reader could read the part related to their interest which have been explained in the next few lines. To get an overview about the document and the project, see Overall Description and to read about the features of the product, check System Features. The readers who are keen to know about interface and navigation between different front ends could go through External Interface Requirements. Other aspects which have been used to make the project could be seen in Other Nonfunctional requirements.

1.4 Project Scope

Using this application, the user can adjust in a new place without facing much trouble as it will help the user to get familiarized with the local market.

1.4.1 Business case and goal

It is difficult to find commodities these days, specifically when someone moves to a new place. The goal of this implementation is to provide an interface for end users as well as the business dealers which makes dealing and searching for goods easier. The deadline of this project is 15 November 2020.

1.4.2 Project description and deliverables

This project is a web application with searching, routing and map functionality. Final delivered project should be able to

- Register new users
- Register new business dealer
- Allow user access goods he/she desire
- Allow dealers to update their locations, description, average pricing etc
- Allow users to review a particular dealer
- Sort dealers based on their average ratings
- Provide users with map location and route to the dealer he/she desires

1.4.3 Limitations

- Application is limited to web browsers only
- Since OSRM is used for routing, the routing may not be that accurate
- Depending on server hosting services, there may be a limit to the number of users that can be online at a given time

1.4.4 Assumptions

- The location provided by dealer is accurate
- Reviews are honest
- Registered user is authentic

1.5 References

https://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html

https://www.researchgate.net/publication/304703920_A_Review_Paper_on_E-Commerce

https://github.com/jpeisenbarth/SRS-Tex

https://www.bmc.com/blogs/software-requirements-specification-how-to-write-srs-with-examples/

2. Overall Description

This part of the document will give you an overview of the requirements and other subsections. All the factors that impact the requirements of the project are being considered in the sections below.

2.1 Project Perspective

The product is supposed to be open source. It is a web-based system implementing a client-server model.

- User account: The system allows the user to create their accounts in the system and provide features of searching, rating, and adding any shop.
- Number of users being supported by the system: Though the number is precisely not mentioned since it depends on the server hosting services but the system is able to support a large number of online users at a time.
- Search: search is simply a local search engine based on keywords.

2.2 Product Features

- Using this product, the user will have a clear idea of the area where he/she is living in.
- The user will be able to meet the necessities without going through much trouble of finding the shop.
- The user can also get the product delivered if the shop has that facility.
- The reviews and ratings given to any shop will help other users to decide among different shops.

2.3 User Classes and Characteristics

Users should be familiar with items like login, register, order system etc.

2.3.1 Customer

Through a web browser the customer can search for commodities whichever they require which could be either searching for fruit vendors or cobblers or for grocery shops even for the barbers. The user can login using his/her account details or new customers can set up an account very quickly. The users just need to provide basic details about them.

2.4 Operating Environment

The system operates with following software components and applications. A full internet connection is needed for OSS and it could be operated on any operating system and a browser required like Google Chrome or Microsoft edge or any other browser could be used which are supported by various operating systems.

2.5 Design and Implementation Constraints

Design constraints are those **constraints** that are imposed on the **design** solution,**Design constraints** can have a significant impact on the **design** and should be validated prior to imposing them on the solution.

2.5.1 Hardware

Since the number of users that can use the web application at a time directly depends on the hosting server hardware, Hence Hardware is limited by server's memory and server gateway.

2.5.2 Software

Languages used for this application are HTML, CSS, JavaScript. The backend of the application is written using Nodejs library along with Express, Passport, Flash messages. The database is implemented using SQL. Map is implemented using leaflet.js and routing is done by OSRM (open source routing machine).

2.6 User Documentation

Following Documentation will be provided along with the website:

- User manual for dealers
- User manual for end user

2.7 Assumptions and Dependencies

These following assumptions will be used in the software:

- Client/User has an active internet connection or has access to view the website or software.
- Client/User runs an operating system which supports the Internet Browsing.
- Client/User should know the proper details for the login. No false details should be accepted.
- Client/User has an operating system which supports or has built-in GPS system.
- Our website will not be violating any internet ethnic or cultural rules and won't be blocked by the Telecom companies.
- During the rainy season, the delivery charges may increase.
- The supplier will deliver the consumables on time with proper conditions.
- There will be no expiry products going to be delivered
- Time between two locations is estimated with moderate traffic.

3. System Features

This section demonstrates most prominent features and explains how they can be used and what results will they give back to the user i.e. a functional requirement defines a system and its components further specifying "What should the software system do?"

The table below (3.1) shows a template that will be used to describe functional requirements for different types of users: students, different shopkeepers as one can easily deduce the functional requirements for other user types with this template.

Purpose	A description of the functional requirements and its reasons
Input	What are the inputs; in what form will they arrive; from what sources can the inputs come; what are the legal domains of each input.
Processing	Describes the outcome rather than the implementation; includes any validity checks on the data, exact timing of operation (if needed), how to handle unexpected or abnormal situations
Output	The form, shape, destination and volume of output; output timing; range of parameters in the output; unit of measure of the output; process by which output is stored or destroyed; process for handling error messages produced as output.

- **Transaction correctness and cancellations**: The system will provide a button to return to the home page to restart the transaction and a cancel button to cancel the transaction.
- **Authorization Levels**: This system has two levels of authorization:
  1. Administrator level
  2. User level

The users are of different types: students, different shopkeepers

- **Authentication:** The system will verify identity by asking login id and password.
- **Historical Data**: Use of stored data for checking orders and login details.
- **End-User**: The system is open to all users.

3.1 Functional Requirement for User

While making a product it is highly important to know about the requirements which a user will be requiring while using the product which are discussed in this part.

3.1.1 Description and Priority

This is one of the most prioritized requirements as it enables the user to access the software and place requests to the shopkeepers.

3.1.2 Stimulus/Response Sequences

The user can view the outlet details on the map. He/She can also track the request made by him/her and get a response from the shopkeeper to which He/She has placed the order.

3.1.3 Functional Requirements

Purpose	It provides information specific to each user like Contact Number, Name, Roll Number, Email ID, Hostel Details etc.
Inputs	Any user can view a page of information by choosing from one of the options given on the screen. Selecting and placing requests is performed with a simple touch.
Processing	The menu responds to selections by displaying a page containing the pre-defined text requested information.
Output	Output consists of a screen of information specific to each user. For example, each user will be able to: 1. Access and edit his/her login details. 2. Track and edit his/her requests. 3. View Outlet Details, Delivery Details. 4. View Map.

3.2 Functional Requirements for Admin

The requirements which are necessary for an admin to run the project are given in this subsection.

3.2.1 Description and Priority

Admin holds the highest priority regarding the software. They can generate or edit any data backup/recovery of data at any time.

3.2.2 Stimulus/Response Sequences

The admin can view the details of all the requests that have been placed by the users. This provides the admin with the exact specification of the number of products that have been donated/accepted by users along with user details to whom it has been provided.

3.2.3 Functional Requirements

The admin will be provided with all the data of the requests made previously. He/She can also edit the map details, outlet details and transport details.

Purpose	Admin holds the highest priority and can view all requests. He/She can also edit the map, outlet and transport details.
Inputs	The Admin can view and edit pages of information by choosing from one of the options given on the screen. All the necessary changes and operations can be performed by a simple touch.
Processing	The software responds to selections by displaying the required information.
Output	Output consists of a screen of information specific to the administrator. For example, admin will be able to: 1. Access all the user details. 2. Track and process all the user requests. 3. Modify Outlet Details. 4. Modify Map Data.

4. External Interface Requirements

Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

4.1 User Interfaces

The user interface for the software shall be compatible with any browser such as Internet Explorer, Mozilla etc. by which users can access the system.

The user interface shall be implemented using any tool or software package.

4.2 Hardware Interfaces

Lively Local Market is a web-based application and hence it is not hardware extensive. It requires a device with a basic processing unit, a working internet connection and an operating system which support a browser.

4.3 Software Interfaces

Following are the software used for our system application:

1. Front-end software: HTML, CSS, and JavaScript
2. Back-end software: NodeJS

Software Used	**Description**
Operating System	We have chosen windows operating system for it's best support and user-friendliness
Database	To save the item and user records, we have chosen SQL database
UI Layout	For visuals we are using Bootstrap and jQuery which are libraries of HTML, CSS, and JavaScript respectively.

4.4 Communications Interfaces

Lively Local Market uses a web browser, HTTP/HTTPS as network server communication protocol, electronic forms will be used for login and user registration which will be end-to-end encrypted.

5. Other Nonfunctional Requirements

In this section we discuss the other important requirements of the application concerning its performance, security and safety. These constraints are typically imposed by the customer, by the development organization, or by external regulations. The constraints may be imposed on the hardware, software, data, operational procedures, interfaces, or any other part of the system.

5.1 Performance Requirements

- **Lively Local Market** requires a system with at least 500 MHz CPU and 128 MBs of RAM. Performance entirely depends on the internet connectivity. Amount of information to be handled can vary from user to user.
- **RESPONSE TIME:** Average response time shall be less than 5 seconds.
- **THROUGHPUT:** The system shall accommodate at least 1000 booked per minute. For normal conditions, 95% of the transactions should be processed in less than 5 seconds.
- **RECOVERY TIME:** In case of a system failure, redundant systems shall resume operations within 1 minute. Average repair time shall be less than 1 hour.
- **START-UP/SHUTDOWN TIME:** The system shall be operational within 1 minute of starting up.
- **CAPACITY:** The system accommodates 2000 concurrent users. Multiple users can login to the system from different machines. The software will support simultaneous user access only if there are multiple terminals.
- **UTILISATION OF RESOURCES:** The system shall store in the database no more than one million transactions. If the database grows over this limit, old transactions shall be backed up and deleted from the operational database.

5.2 Safety Requirements

- The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.
- To ensure that no one of the users loses any data while using software (due to a crash or a bug) the developer team updates it regularly. There is a feedback section where the user can write to us regarding any issues (Complaint/Feedback).

5.3 Security Requirements

Lively Local Market does not have any security requirements and thus any type of user can use it without any additional privileges, but some basic security will be provided like:

- Proper security regarding the access of data will be provided.
- External security could be provided by giving the login authentication.
- The data stored in the database must be private plus the admin can lock the private data so that no user could access it.

5.4 Software Quality Attributes

The product is targeted towards users such as Students and vendors. Our

software displays various quality attributes that are given below:

Availability- The software is easily available for everyone.

Correctness- The results of the program will be accurate.

Maintainability- After the deployment of the project if any error occurs then it can be easily

rectified by the software developer.

Portability- The software can be deployed at any device.

Reusability- The data and record that are saved in the database can be reused if needed.

Robustness-If there is any error in any window or module then it does not affect the remaining part of the software.

Usability-To performs any operations and to understand the functioning of software is very easy.

Productivity-This software will produce every desired result with accuracy.

Timelines-The time limit is very important. It will save much time and provide fast accessing.

3. Structured Analysis

The analysis model allows you to drill down into the specification of certain requirements. Using graphical notation to communicate more clearly than natural language (imprecise) and code (too detailed). Models help us organize, visualize, understand, and create complex things.

3.1 Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself.

3.1.1 Level 0

It's a basic overview of the whole system or process being analysed or modelled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

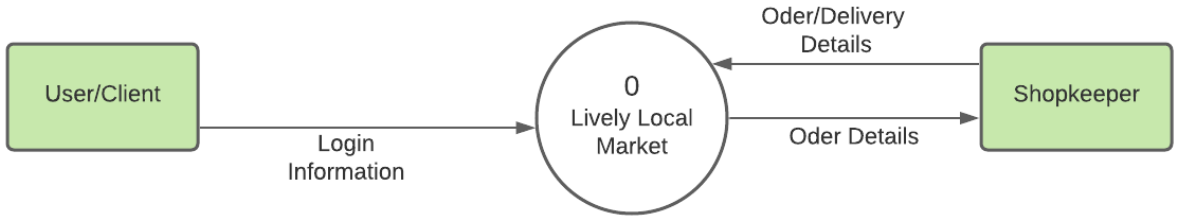


Figure 1 DFD level-0

3.1.2 Level 1

A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into subprocesses that can then be analyzed and improved on a more intimate level.

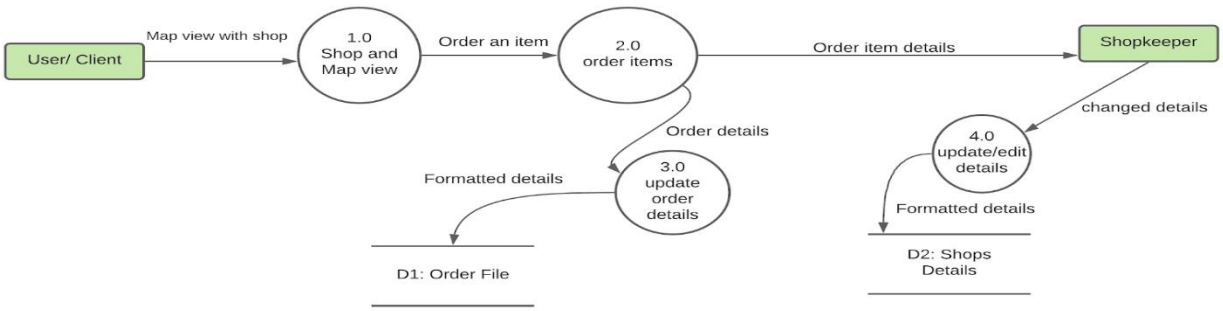


Figure 2 DFD level-1

3.1.3 Level 2

A level 2 data flow diagram (DFD) offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system.

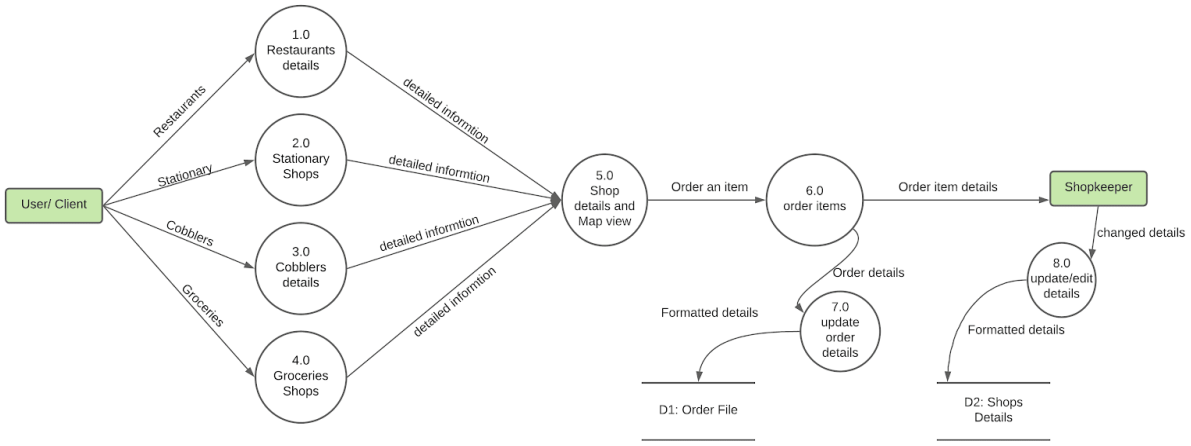


Figure 3 DFD level-2

3.2 ER Diagram

They help to identify different system elements and their relationships with each other. It is often used as the basis for data flow diagrams or DFD's as they are commonly known.

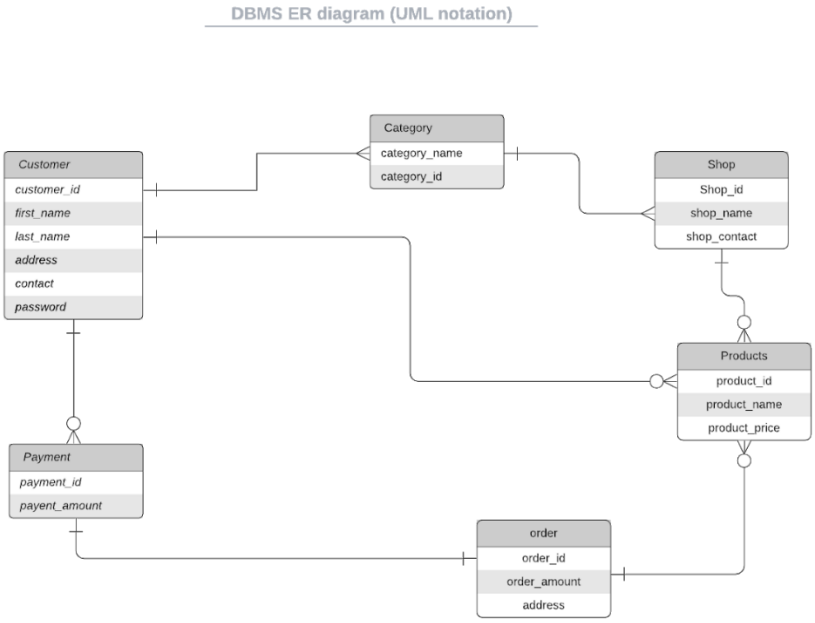


Figure 4 ER Diagram

3.3 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

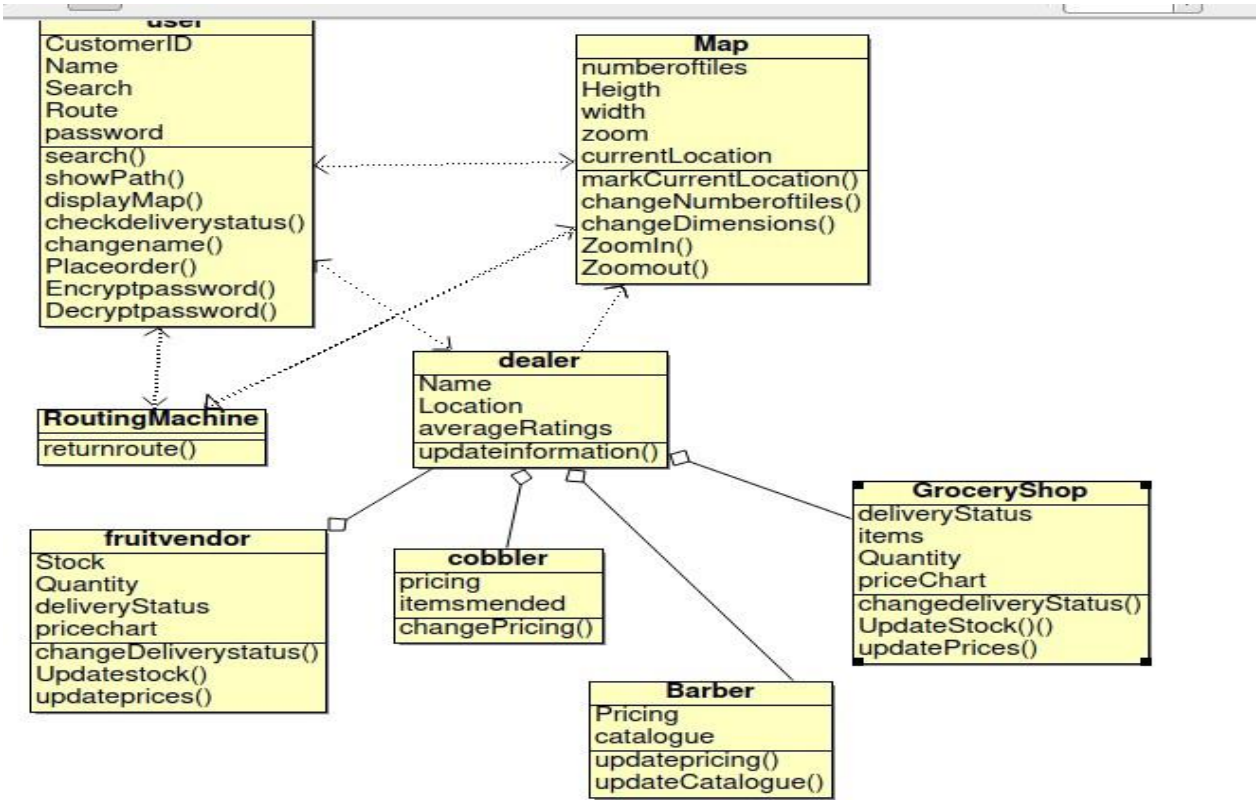


Figure 5 Class Diagram

4. Object Oriented Analysis

Object Oriented Analysis (OOA) is the first technical activity performed as part of object-oriented software engineering. OOA introduces new concepts to investigate a problem. It is based in a set of basic principles, which are as follows-

1. The information domain is modeled.
2. Behavior is represented.
3. Function is described.
4. Data, functional, and behavioral models are divided to uncover greater detail.
5. Early models represent the essence of the problem, while later ones provide implementation details.

4.1 Use Case Diagram

UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation.

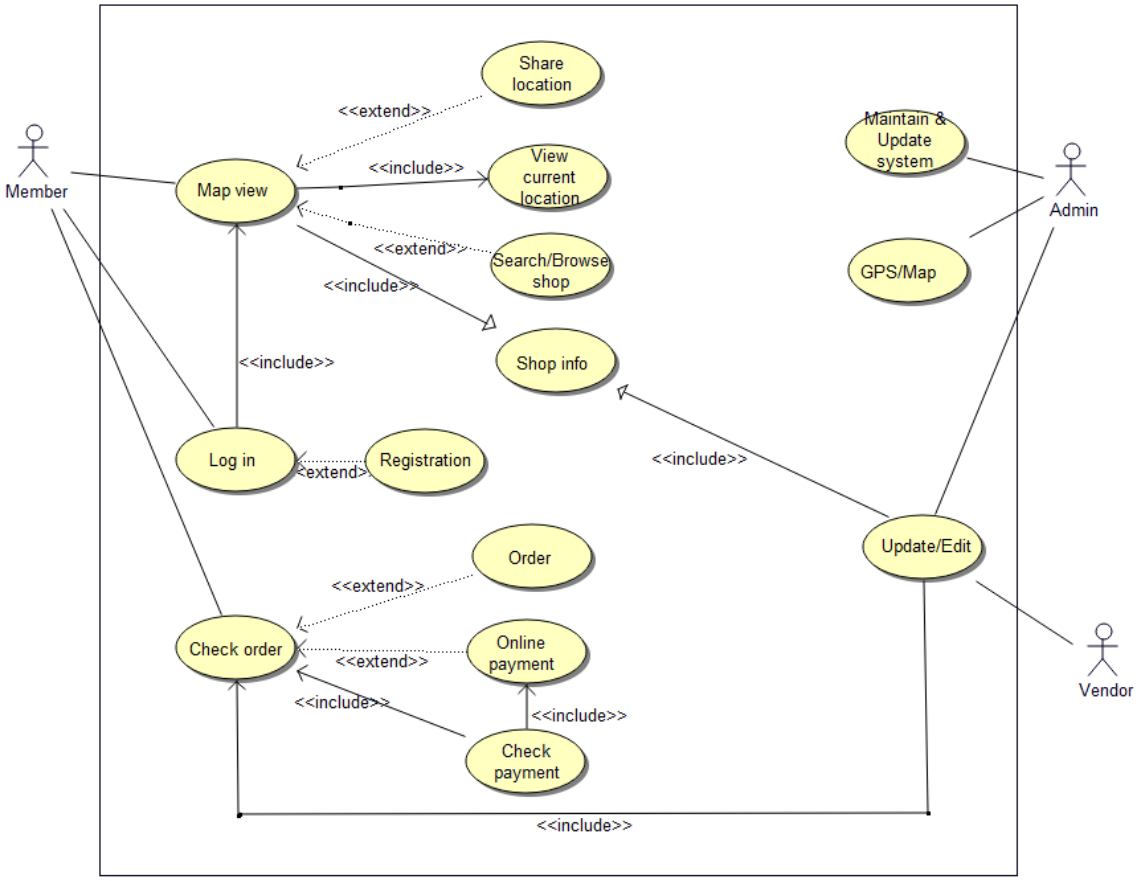


Figure 6 Use Case Diagram

4.2 Use Case Templates

Use case Name	Register Member
Type	Primary
Actor(s)	Member
Brief descriptions	To enroll in order to use the application
Preconditions	User should be a new student
Post-conditions	After successful registration, the member can use all facilities available

Use case Name	Search shop/vendor
Type	Primary
Actor(s)	Member
Brief descriptions	Member checks if vendor is available or not, or if the shop is opened
Preconditions	User should be a verified and registered member
Post-conditions	If a dealer is available, a member can contact them if they want

Use case Name	Order
Type	Primary
Actor(s)	Member,Vendor
Brief descriptions	Member could order for delivery of an item
Preconditions	The item selected for order must be there with the dealer
Post-conditions	Member can now request to complete order

Use case Name	Maintain and Update system
Type	Secondary
Actor(s)	Admin
Brief descriptions	The admin would keep updating the application from there end

Preconditions	When the admin has to add some new vendor to the application or remove any
Post-conditions	It would include access to some new vendors or decrease access of some other vendors

Use case Name	Update/Edit
Type	Secondary
Actor(s)	Admin,Vendor
Brief descriptions	The vendor could add or remove items and admin could help to update info about vendors
Preconditions	The vendor must be added by the admin
Post-conditions	In this way, vendor could keep the page updated

Use case Name	Map View
Type	Primary
Actor(s)	Member
Brief descriptions	To get a map view of the vendor or even to get the view around oneself
Preconditions	The user must have an active internet connection and a valid account details.
Post-conditions	The user will be able to see locality in map

4.3 Activity Diagram

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram.

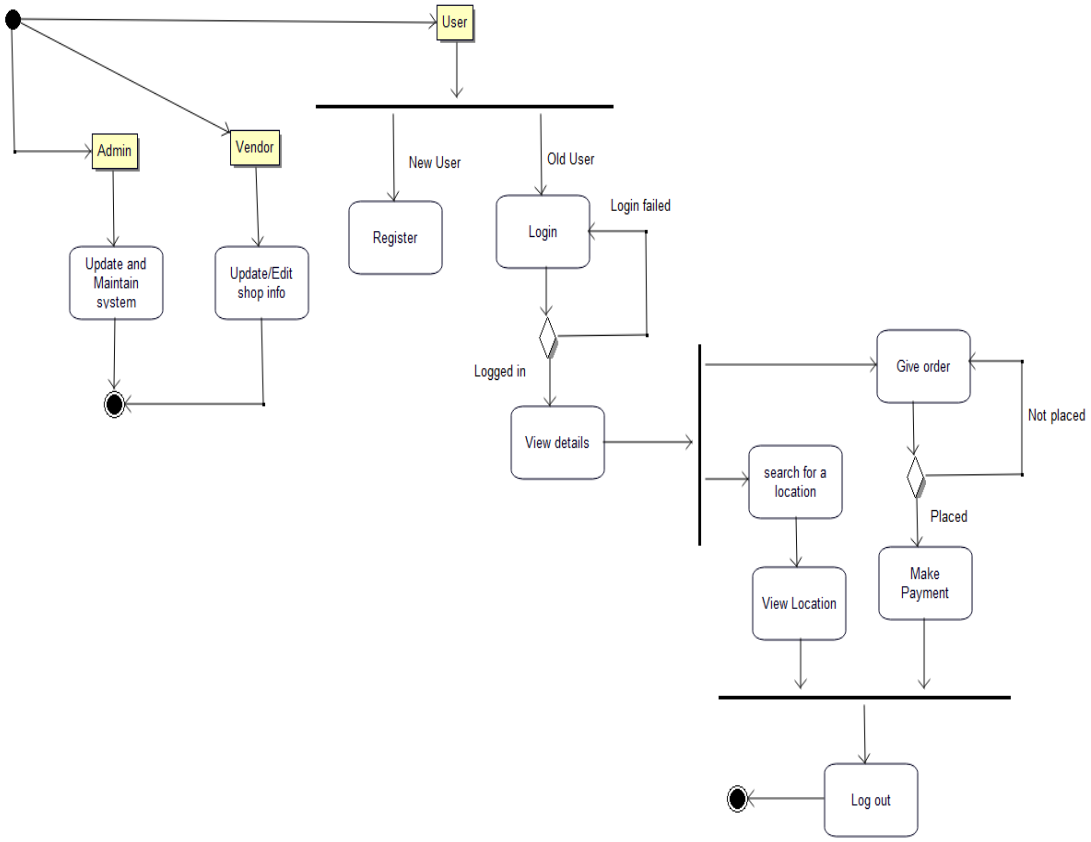


Figure 7 Activity Diagram

4.4 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object

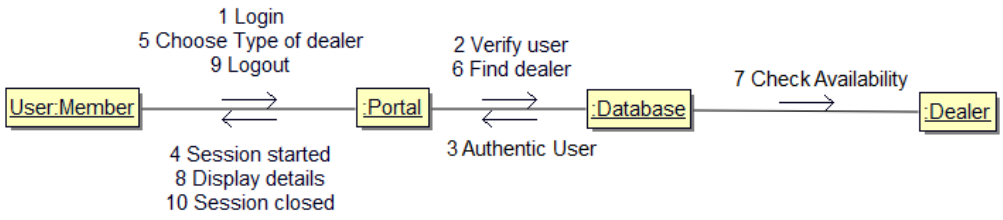


Figure 8 Collaboration Diagram

4.5 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

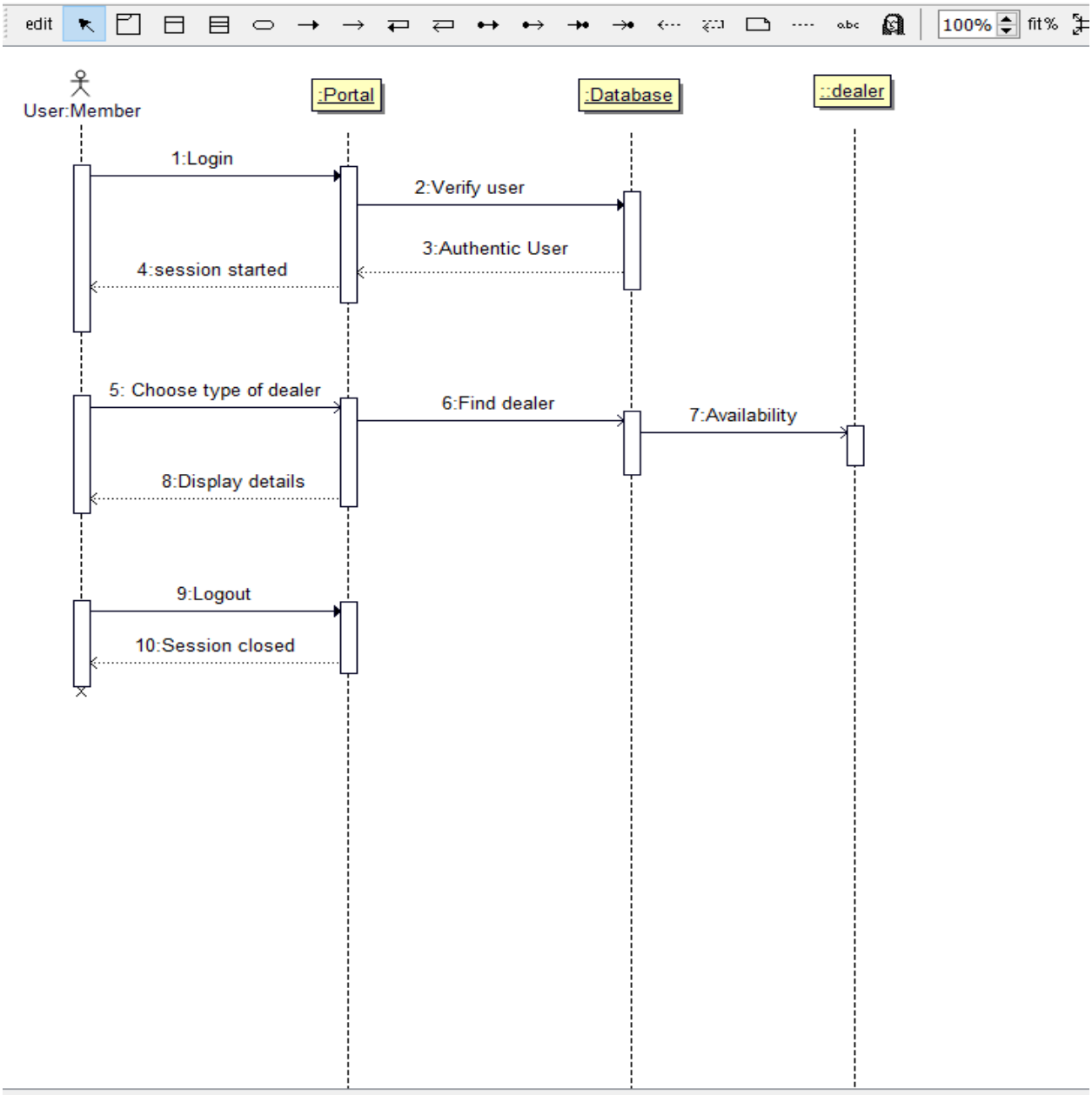


Figure 9 Sequence Diagram

4.6 State Chart Diagram

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems

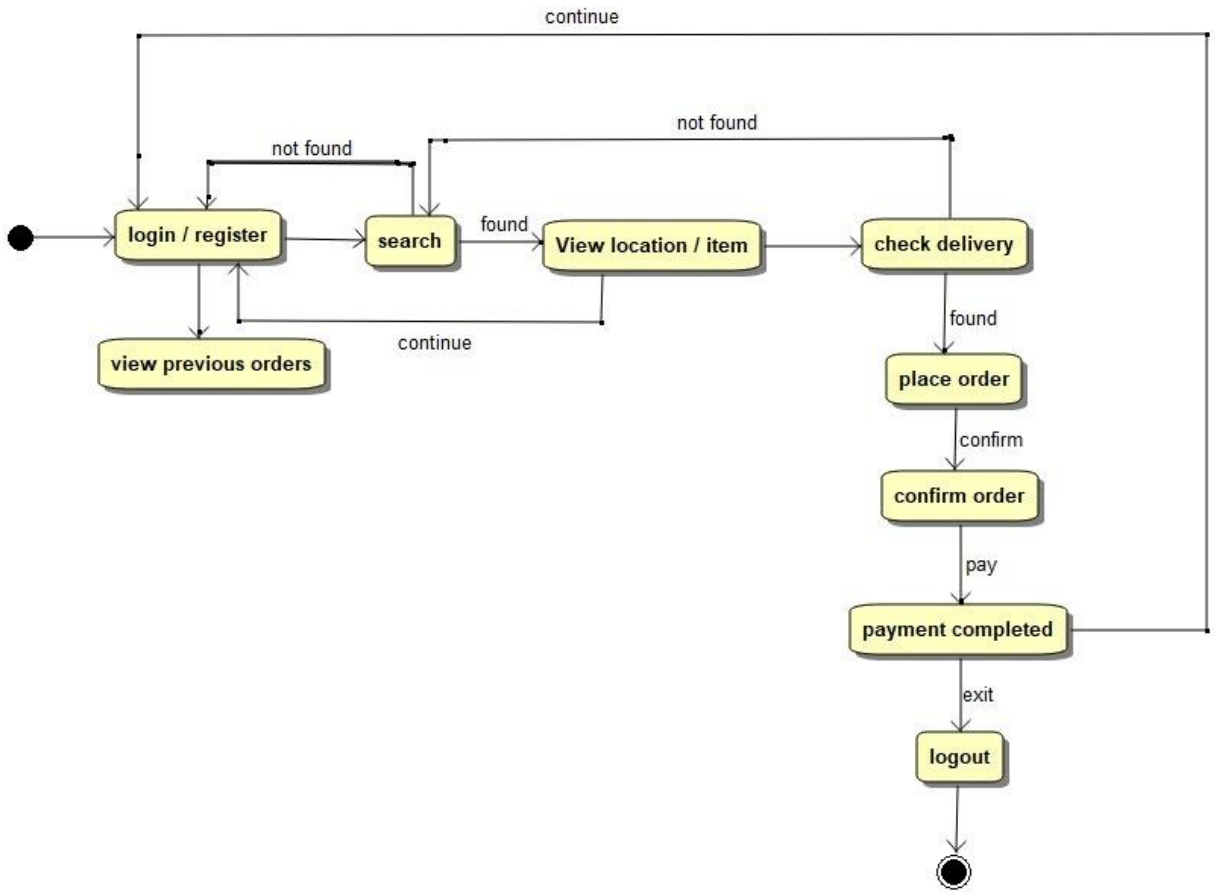
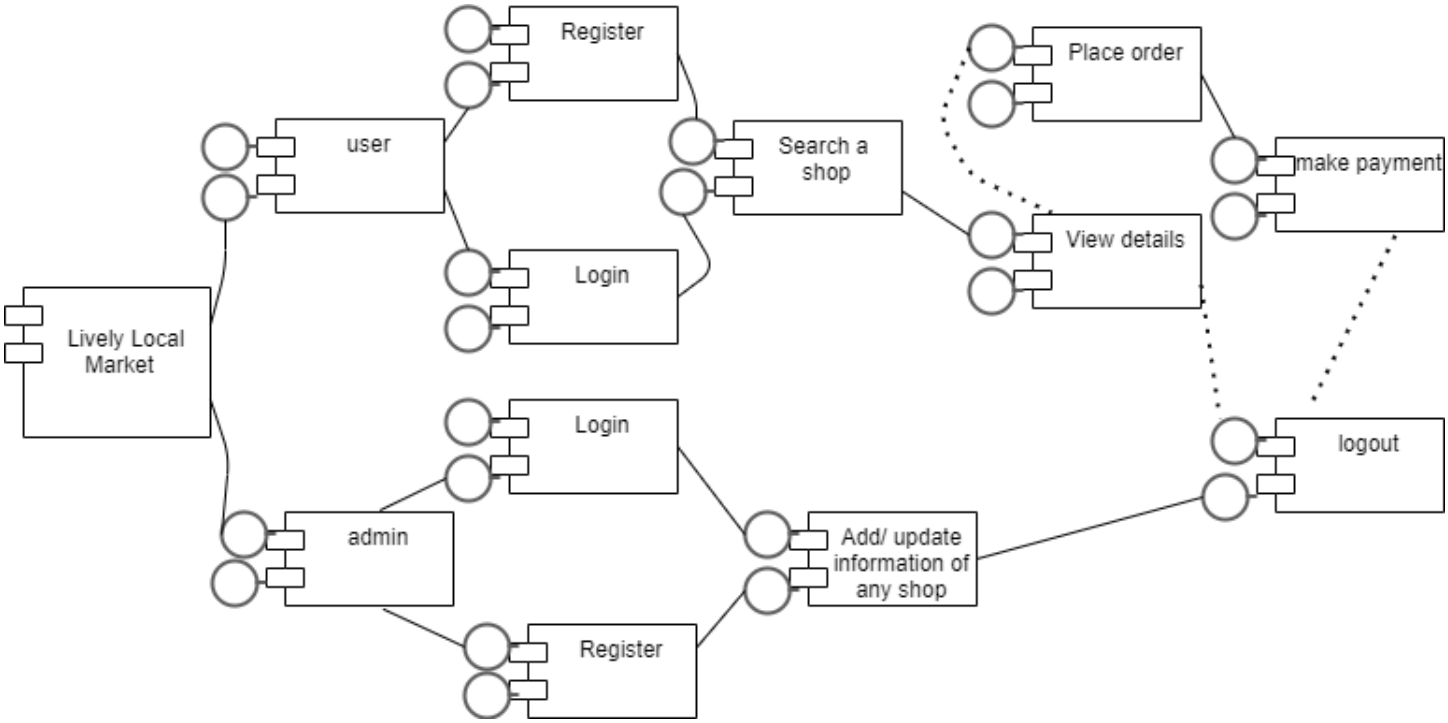


Figure 10 State Chart Diagram

4.7 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



4.8 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system

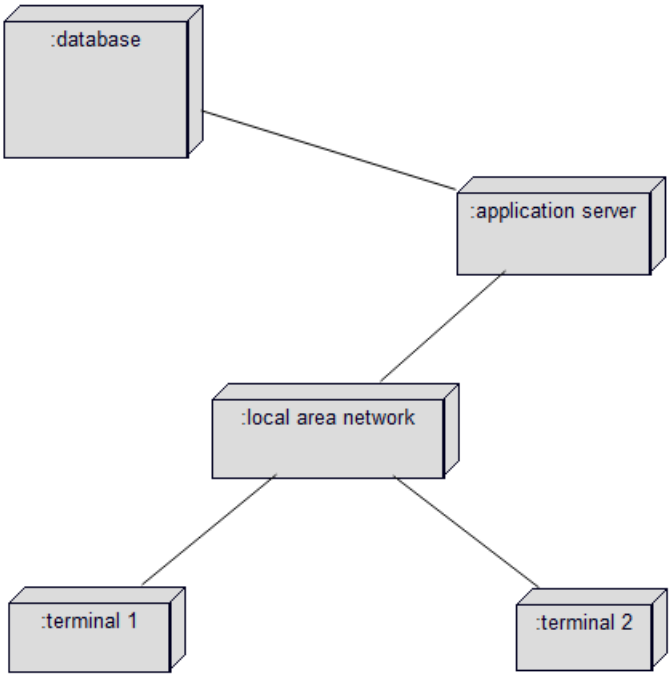


Figure 11 Deployment Diagram

5. Testing

5.1 Test Plan

Test case no: 1
Test case name: login into system
Components tested: login button, check validation of username
Environment requirements: web browser, database, internet connectivity
Testing strategy: On clicking the login button, a window to enter username and password is displayed. The username name and password is entered and login button is clicked. The database is searched for the username, if an account with that username does not exist, an error message is displayed and a prompt to enter new username is displayed. If the username is valid, the user is successfully logged in.
Post conditions: If the user is successfully logged in, the home page is displayed with successful login message. Otherwise an error message reporting login failed is displayed.

Test case no: 2
Test case name: signup into system
Components tested: signup button, module store to details in DB
Environment requirements: web browser, database, internet connectivity
Testing strategy: On clicking the signup button, a window to enter details is displayed. The username, password, phone no and category are filled and signup button is clicked. If all required details are filled and valid, the user is successfully registered otherwise an error message is displayed.
Post conditions: Record of user is stored in database and homepage is displayed with successful registration message.

Test case no: 3
Test case name: search vendor
Components tested: vendor button, get list of particular type of vendor, display vendors result, display detailed profile of account
Environment requirements: web browser, database, internet connectivity
Testing strategy: The type of vendor user wants to visit, user can find it on the homepage and see vendors according to ratings they have got from people and could find if they want to see any particular shop over there and they would get all the required information of the particular vendor.
Post conditions: Search results according to the selected filters are displayed or details of a particular result are displayed if it is clicked.

5.2 Test Case Report

5.2.1 Test case 1

Test case #:1.1

Test Case Name: login into system

System: Lively Local Market

Subsystem: Profile

Designed By: Kushagar, Kartik, Muskan, Nikhil

Design Date: 02/12/2020

Execution Date: 02/12/2020

Executed By: Kushagar, Muskan, Nikhil, Kartik

Short Description: Testing the login functionality in system

Pre-conditions:
The user has a valid user account and password
The system displays main page

Step	Action	Expected System Response	Pass/Fail	Comment
1	Navigate on login Button	Able to see login page	pass	
2	Enter username	The username entered into username field	pass	
3	Enter password	The password entered into password field	pass	
4	Click login button	The user is logged into system and form will be displayed	pass	
5	Enter invalid password	The message “invalid username or invalid password” is shown		

Post condition:
User is validated with Database and successfully log in to account. The details of user will be displayed if it is available.

5.2.2 Test case 2

Test case #:2.1 **Test Case Name:** Signup into system

System: Lively Local Market **Subsystem:** Profile
Designed By: Kushagar, Muskan, Nikhil, Kartik **Design Date:0**2/12/2020
Executed By: Kushagar, Kartik, Muskan, Nikhil **Execution Date:** 02/12/2020

Short Description: Testing the Signup functionality in system

Pre-conditions:
The user has a unique uid and valid phone number and valid email address
The system displays main page

Step	Action	Expected System Response	Pass/Fail	Comment
1	Navigate on Signup Button	Able to see login page	pass	
2	Enter Name	The Name entered into Your name field	pass	
3	Enter Email address	The email address entered Your email field	pass	
4	Select the occupation	The student is selected into occupation field	pass	
5	Enter password	The password entered into Password field	pass	
6	Enter confirm password	The confirm password entered into field	pass	
7	Select the Home delivery option	The option is selected according to user	pass	
8	Click signup button	Move to create description page	pass	

Post condition:
The record of user is stored in database after successful signup.

5.2.3 Test case 3

Test case #:3.1 **Test Case Name:** Get details of a shop

System: Lively Local Market **Subsystem:** Profile

Designed By: Kushagar, Muskan, Nikhil, Kartik **Design Date:**02/12/2020

Executed By: Kushagar, Muskan, Nikhil, Kartik **Execution Date:** 02/12/2020

Short Description: Testing the Searching functionality in system

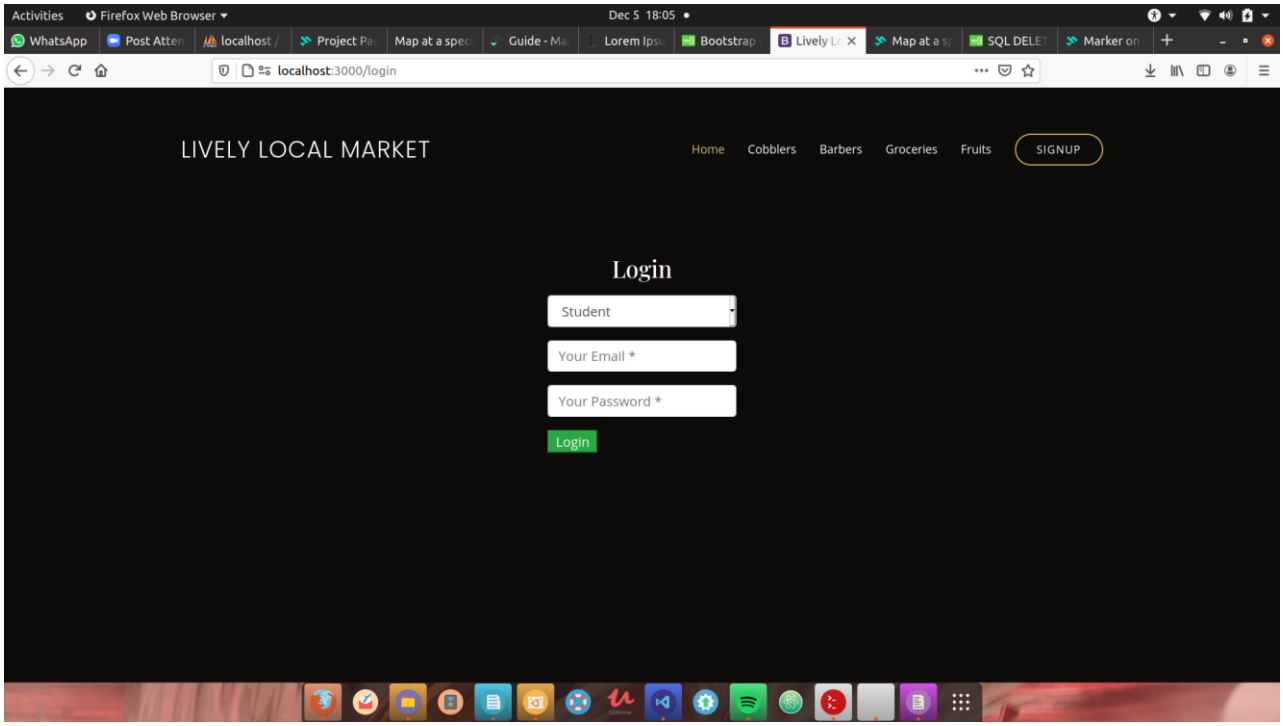
Pre-conditions:
The system displays main page

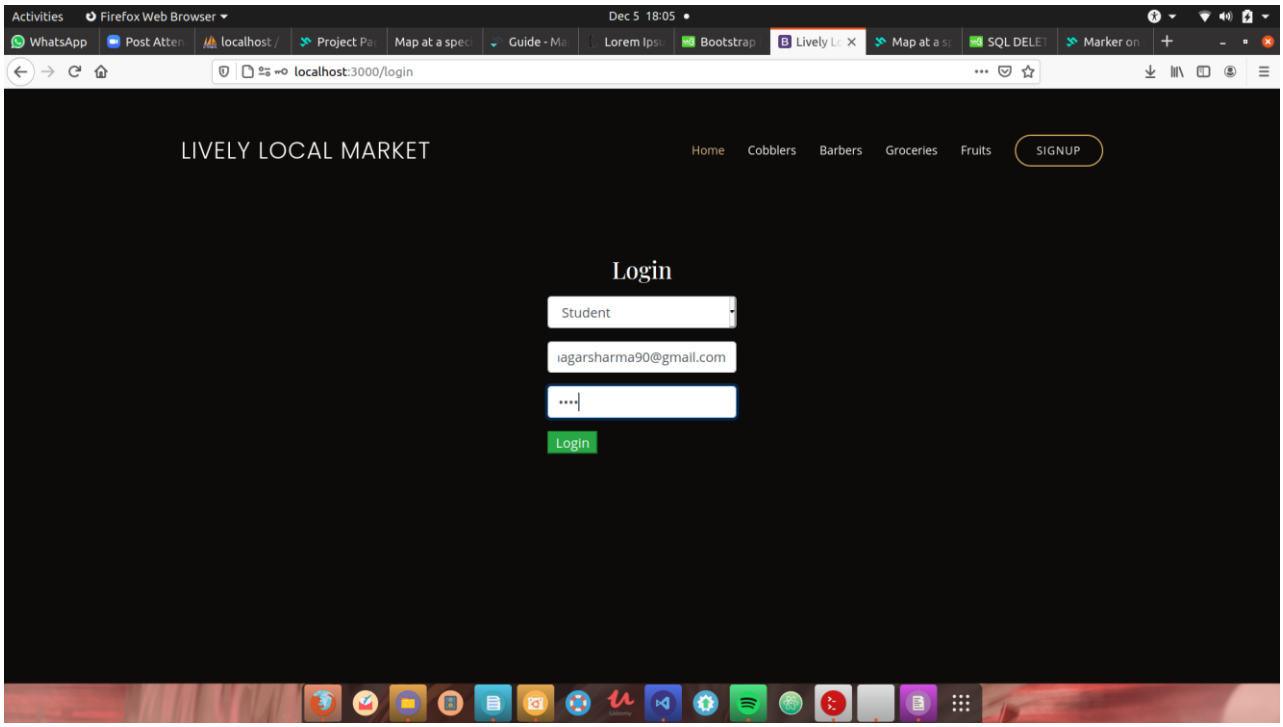
Step	Action	Expected System Response	Pass/Fail	Comment
1	Select requirement	show dealers	pass	
2	Select between dealers	List of store/ vendors is displayed.	pass	
3	Select the desired store/ vendor	The desired store/ vendor is selected.	pass	
4	Click on detail button	Details displayed	pass	

Post conditions:
Search results according to selected filter are displayed on screen

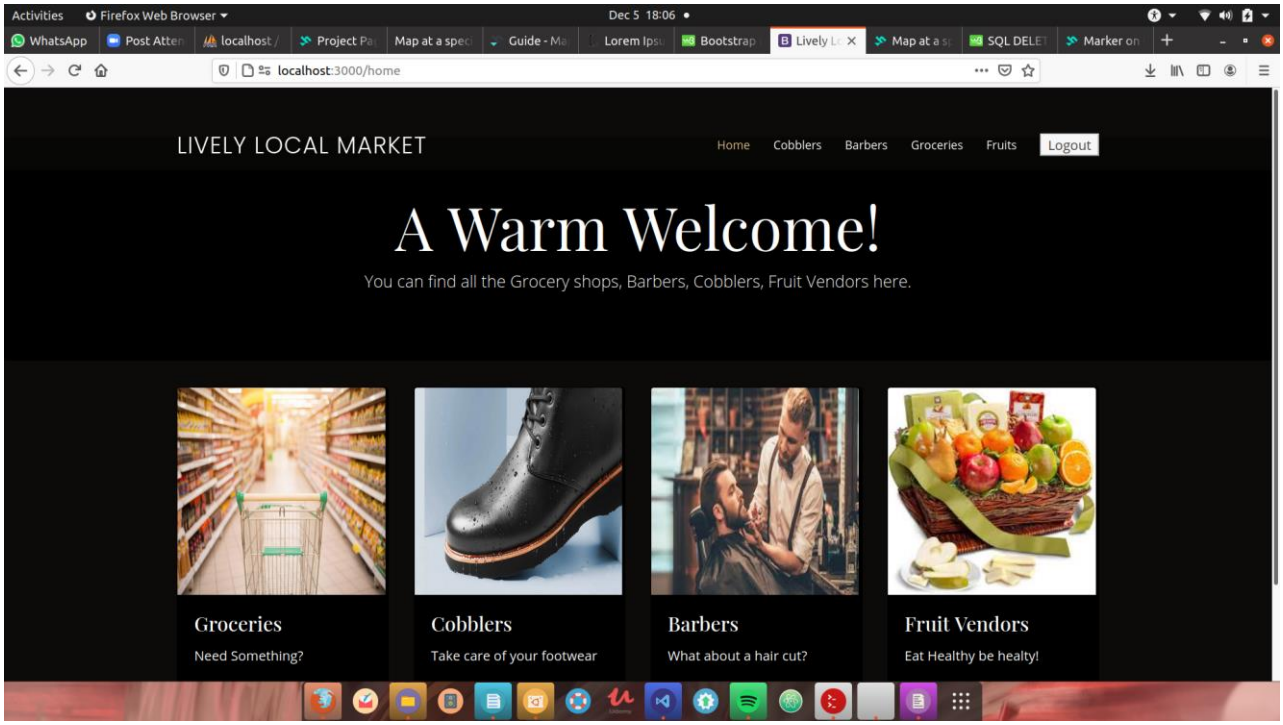
5.3 Screenshots

Login Page

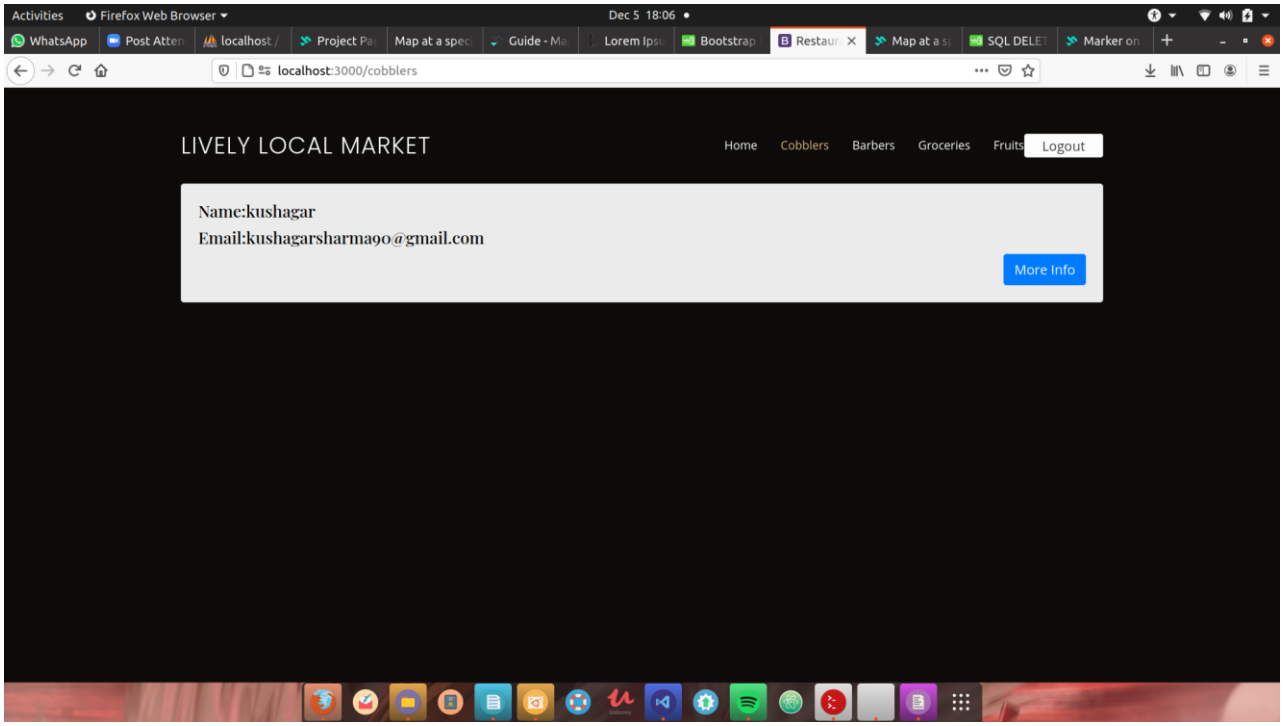




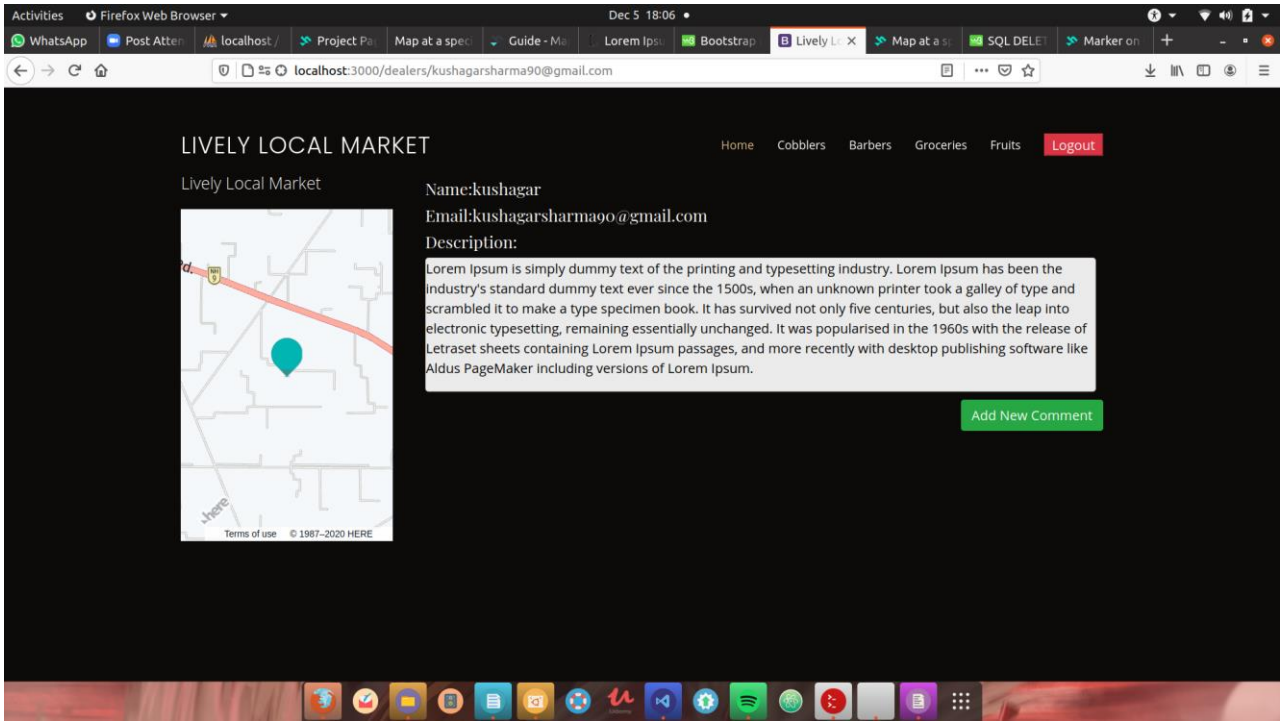
User Homepage



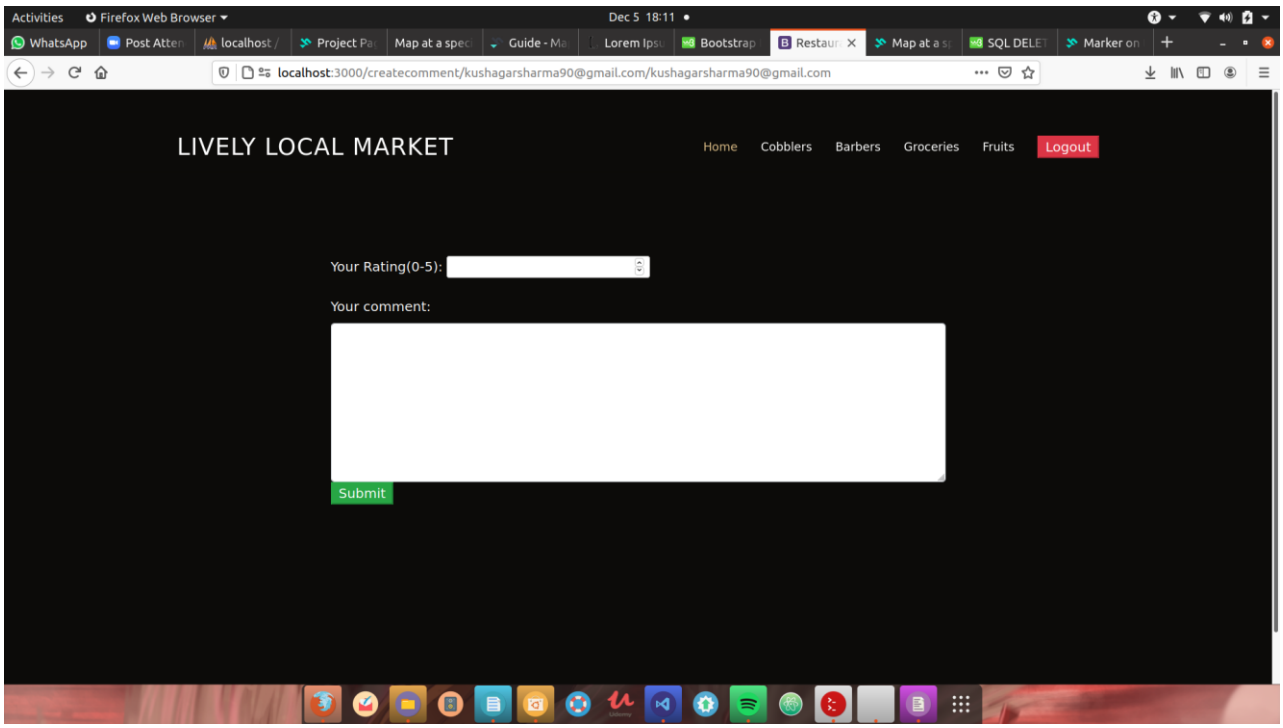
Vendor Page

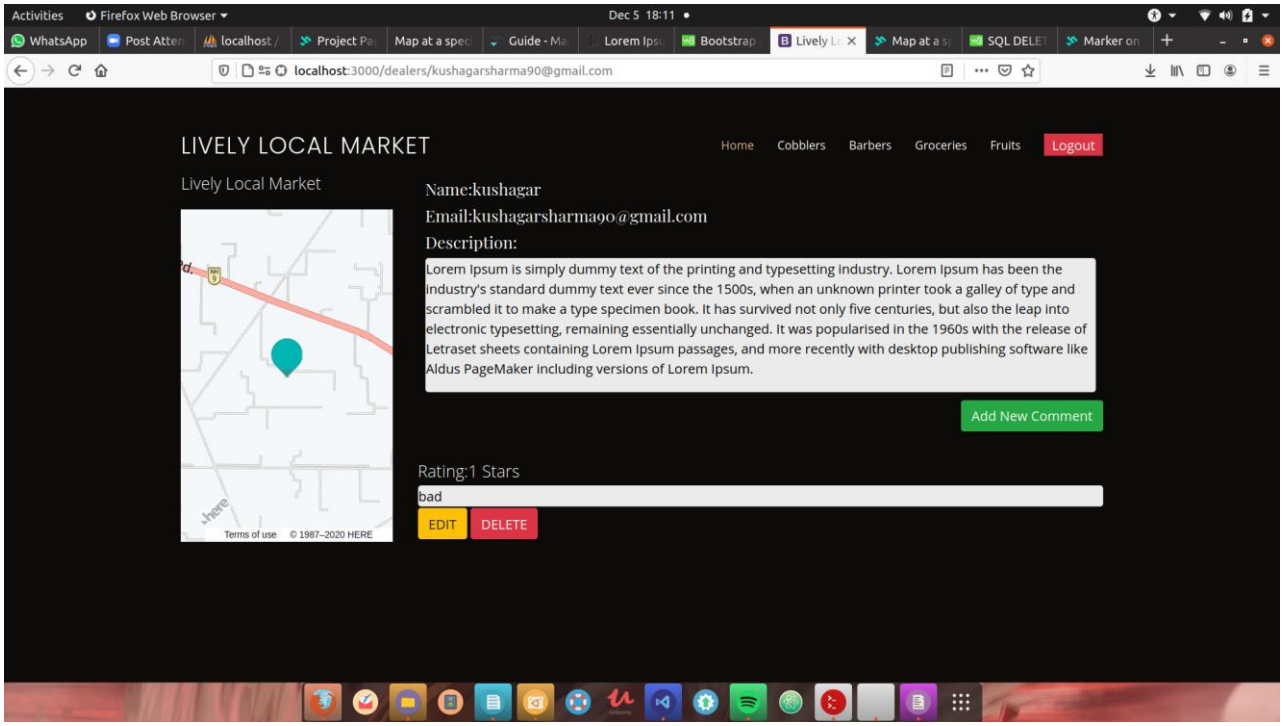


Location



Rating/Comment Page




Activities
Firefox Web Browser
Dec 5 18:11
localhost:3000/dealers/kushagarsharma90@gmail.com
LIVELY LOCAL MARKET
Home
Cobblers
Barbers
Groceries
Fruits
Logout
Lively Local Market
Terms of use
© 1987–2020 HERE
Name:kushagar
Email:kushagarsharma90@gmail.com
Description:
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
Add New Comment
Rating:1 Stars
bad
EDIT
DELETE