RateMyFood

\

Table of Contents

List of Tables

List of Figures

Project Synopsis

1. Title

RateMyFood

2. Statement about the problem

To find hygienic local food stalls and small-sized restaurants which provide quality food and services when visiting other localities or area is a common problem people often face due to lack of knowledge about that area. A lot of time is invested to find such places and even though we do find such places the results are quite disappointing.

3. Why this topic?

People often suffer from the problem of finding local food stalls which provide quality food and services. With the help of system, good local food stalls and small restaurants will get recognition and it will help them to generate revenue. The users will save quite the time alongside with good quality food.

4. Objective and Scope

4.1. Objective

- To make accessible platform with good user experience to users which will help them to find quality local stalls and restaurants.
- Restaurants will be able to register on the platform.
- To make information of restaurants, local food stalls available to users for particular locality.
- Users can find quality restaurants and can review the restaurants.

4.2. Scope

- The website will be available in Mumbai, later it can be expanded.
- The website will be available to local food stalls and small restaurants.

5. Methodology

To develop this system, I am going to use the Incremental model.

6. Proposed Architecture

This system will work on three-tier architecture, the web-based application.

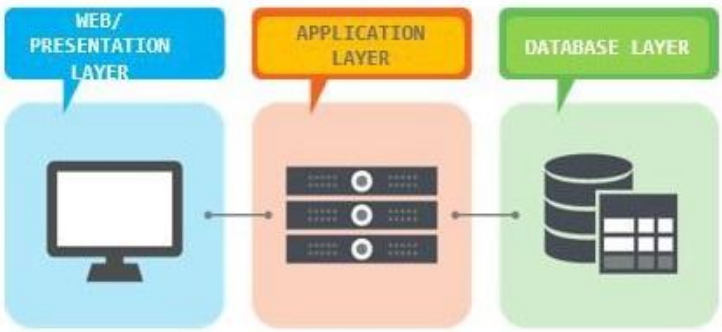


Figure 1.1 Proposed Architecture

7. Requirements

7.1. Software Requirements

- **Frontend -** HTML, CSS and ReactJS
- **Backend -** ExpressJS
- **Database** – MongoDB
- **Operating System -** Windows 7 or higher

7.2. Hardware Requirements

- **Processor -** Intel Pentinum or higher, AMD Ryzen
- **RAM -** 1 GB or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

8. Platform

Visual Studio Code

9. Contribution

RateMyFood website helps in saving time and efforts to find hygienic and quality food. It also provides a marketing platform for local food stalls and small-sized restaurants. Both the customers and the shop keepers can use the website to shorten the distance between them and get feedbacks for their improvements.

Chapter 1:
Introduction

1.1. Background

The inspiration is taken from yelp. Yelp is a review website for various services. User can see basic information about particular service provider. They can see various photos regarding that service provider's shop. User can see reviews given by other users for a specific service. They can also add review from their experience about that shop.

1.2. Objectives

- Users can create an account and can see various local food stalls and restaurants of particular area.
- Restaurants can create an account and add their basic information.
- Users can review particular restaurants by giving them ratings.
- Users can also add text, photos based reviews about restaurants.
- Restaurants with maximum good reviews will top the list.
- Based on the reviews given by users, website will provide some points to users which will get converted into coupons.

1.3. Purpose, Scope and Applicability

1.3.1. Purpose

Purpose of this system is to solve the problem of finding local food stalls and restaurants which provide quality food and services. This system will save time of users and they can enjoy quality food without too much work. To give recognition to restaurants which provide quality food and services is also one of the purposes of this system.

1.3.2. Scope

- The website will be available to local food stalls and small restaurants.
- The website will be available in Mumbai, later it can be expanded.
- This website will be compatible to all digital devices like phone, laptop, tablet, etc.

1.3.3. Applicability

RateMyFood website would make finding for street food easier. Anyone with devices like smartphone, tablets, laptops and decent internet connection can use this website. This website is more applicable in cities and populated places around center of attractions.

Chapter 2:
Survey Of Technologies

1. AngularJS

AngularJS is a free and open source JavaScript -based web framework for developing single page applications. It was maintained mainly by Google and a community of individuals and corporations. It aimed to simplify both the development and the testing of such applications by providing a framework for client-side model-view-controller (MVC) and model-view-viewmodel (MVVM) architectures, along with components commonly used in web applications and Progressive web applications.

2. PHP

PHP is a general-purpose scripting language geared toward web development. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

3. PostgreSQL

PostgreSQL is an advanced, enterprise class open source relational database that supports both SQL (relational) and JSON (non-relational) querying. It is a highly stable database management system, backed by more than 20 years of community development which has contributed to its high levels of resilience, integrity, and correctness. PostgreSQL is used as the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications. PostgreSQL has a rich history for support of advanced data types, and supports a level of performance optimization that is common across its commercial database counterparts, like Oracle and SQL Server.

4. Svelte

Svelte is a free and open source front-end compiler. Svelte is not a monolithic JavaScript library imported by applications instead, Svelte compiles HTML templates to specialized code that manipulates the DOM directly, which may reduce the size of transferred files and give better client performance, application code is also processed by the compiler, inserting calls to automatically recomputed data and re-render UI elements when the data they depend on is modified. This also avoids the overhead associated with *runtime* intermediate representations, such as virtual DOM unlike traditional frameworks (such as React and Vue) which carry out the bulk of their work at runtime, *i.e.* in the browser. The compiler itself is written in TypeScript.

5. Node.js

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine (i.e. V8 Engine) and executes JavaScript code outside a web-browser, which was designed to build scalable network applications. Node.js lets developers use JavaScript to write command line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Why these technologies?

- **Frontend – ReactJS (JavaScript Framework)**

  ReactJS is flexible as we can use it on a vast variety of platforms to build quality user interfaces. ReactJS has a great developer experience. RaectJS has META's support as well as broader community support which will help this system solve many problems easily. ReactJS has great performance and it is easy to test.

- **Backend – ExpressJS (Node.js Framework)**

  ExpressJS acts only as a thin layer of core web application features. ExpressJS has no out of the box object relational mapping or templating engine. It strives to put control in the developer's hands and make web application development for Node.js easier. This freedom, coupled with lighting fast setup and the pure JavaScript environment of Node.js makes ExpressJS a strong candidate for development.

Chapter 3:
Requirement And Analysis

3.1. Problem Definition

The common problem people face is to find hygienic food stalls and restaurants which provide quality food and services when visiting other localities. RateMyFood will make finding for street food easier. User will be able to see basic information about local stalls. They can also add review from their experience.

Sub Problems –

1. People face problem in finding good food stalls.

It is a common problem to find hygienic, small-sized restaurants and local food stalls providing quality food and services. While visiting new places people often face this problem. This problem costs lot of user's quality time and efforts. The system will solve this problem by suggesting user a good local food stall according to their location and cuisine, based on other people's reviews.

2. Good local food stalls does not get recognition.

There are lot of good local food stalls and small-sized restaurants available in each and every locality. But these stalls often miss out on opportunities due to lack of recognition. The system will solve this problem by providing recognition to such stalls and restaurants. Local stalls with maximum good reviews from customers will top the list. This will help good food stalls get recognition among people.

3.2. Requirement Specification

3.2.1. Requirement Gathering

- **One-on-one interview**

This is one of the most effective techniques for requirement gathering. In this method, the system creator talks to the users who are very close to the problem of the project system creator working on. It is the responsibility of the interviewer to extract relevant information. Below are the tips for conducting productive interviews –

- Ask open-ended questions

- Interview the right person
- Share the question ahead of time

For this system to gather requirements, one-on-one interviews have been done.

- **Brainstorming**

This is a common technique used early in a project, because it acts as a starting point of a project. With brainstorming, we gather as many ideas as possible to identify, categorize and assign tasks, opportunities and potential solutions quickly. In brainstorming sessions, it is important to take notes on generated problems and ideas.

To gather requirements for RateMyFood website brainstorming sessions have been conducted.

- **Studying similar system (yelp.com, travelinsider.com, etc.)**

This technique also helps to gather requirements for the system. We can save time by analyzing existing similar systems which provide very valuable information. This technique gives what we need in our system by analyzing existing similar system's pros and cons.

This technique has been used to gather information for RateMyFood.

Some questions asked during interviews and brainstorming sessions –

- Where do you prefer to eat while you are on a trip?
- Do you find any difficulties in searching for good food? If yes, then how you tackle this problem?
- What do you consider a good food service in this industry?
- Which format will you prefer for posting a review?
- Do you prefer to register yourself on this type of promotion platform?

3.2.2. Requirement Analysis

3.2.2.1. Functional Requirements

i. Two types of accounts, one for common user and another for restaurants.

ii. Ratings and Reviews

Users can add reviews in form of plain text, images, etc. and can give ratings to the various restaurants.

iii. Form for Restaurants

A basic form for restaurants where they can add their basic information such as name, photos, address, contact info, restaurant menu, etc.

iv. Like and Comments

Users can like and comment other users' reviews.

v. Mobile Friendliness

Most of the users are mobile users so website should be mobile friendly.

3.2.2.2. Non-Functional Requirements

i. Security

Restaurant's and user's data should be safe. There must be authentication system for website to access.

ii. Usability

Website's interface should be clean, good-looking and most importantly easy to access even for non-technical users.

iii. Scalability

Website should work as expected even when it scales.

iv. Performance

Focus should be on loading every page of a site as fast as possible regardless of the number of integrations and traffic on the site.

3.2.2.3. System Requirements

i. Login

Input: Username and Password

Source: User

Output: Logged in successfully

Destination: Database

Action: After validation of the information provided by user, user will get access to their account on the website

Pre-Condition: User must provide username and password

Post-Condition: Redirected to home page if credentials are valid

ii. Register

Input: Name, Email, Phone Number, Password, Confirm Password, etc.

Source: User

Output: registered successfully

Destination: Database

Action: After successful registration, separate account for user will be created

Pre-Condition: User must provide all the fields in the form for registration

Post-Condition: Redirected to Login if details are valid and correct

iii. Location Permission

Input: Dialog box for GPS permission (enable/disable)

Source: User

Output: Location permission granted successfully

Destination: Database

Action: Restaurants in that particular location will be displayed

Pre-Condition: Internet connectivity, GPS enabled device

Post-Condition: Location permission granted successfully

iv. File Permission

Input: Dialog box for file access permission (enable/disable)

Source: User

Output: File permission granted successfully

Destination: Database

Action: List of files in the device will be displayed

Pre-Condition: File format should be correct, depending on tasks

Post-Condition: File size

v. Post a review

Input: Text, images and ratings

Source: User

Output: Review posted successfully

Destination: Database

Action: Review posted by user will be displayed under that particular restaurant's section

Pre-Condition: Review must be written in correct format

Post-Condition: Displaying the review under restaurant's section

3.3. Planning and Scheduling

3.3.1. Activity Table

Task No.	Task Name	Start Date	End Date
T1	Project Synopsis	15/06/2022	18/06/2022
T2	Survey of Technologies	20/06/2022	25/06/2022
T3	Problem Definition and Sub-Problems	27/06/2022	02/07/2022
T4	Requirement Gathering	04/07/2022	09/07/2022
	Requirement Analysis		
T5	System Requirements	11/07/2022	16/07/2022
T6	Planning and Scheduling	18/07/2022	29/07/2022
	Software and Hardware Requirements		
	Data Model		
T7	Data Flow Diagram	01/08/2022	20/08/2022
T8	Class Diagram	22/08/2022	27/08/2022
T9	Use Case Diagram	29/08/2022	10/09/2022
T10	Sequence Diagram	12/09/2022	14/09/2022
T11	Activity Diagram	14/09/2022	17/09/2022
	State Diagram		
T12	User Interface Design	15/09/2022	24/09/2022
	Test Cases		

Table 3.1 Activity Table

3.3.2. Gantt Chart

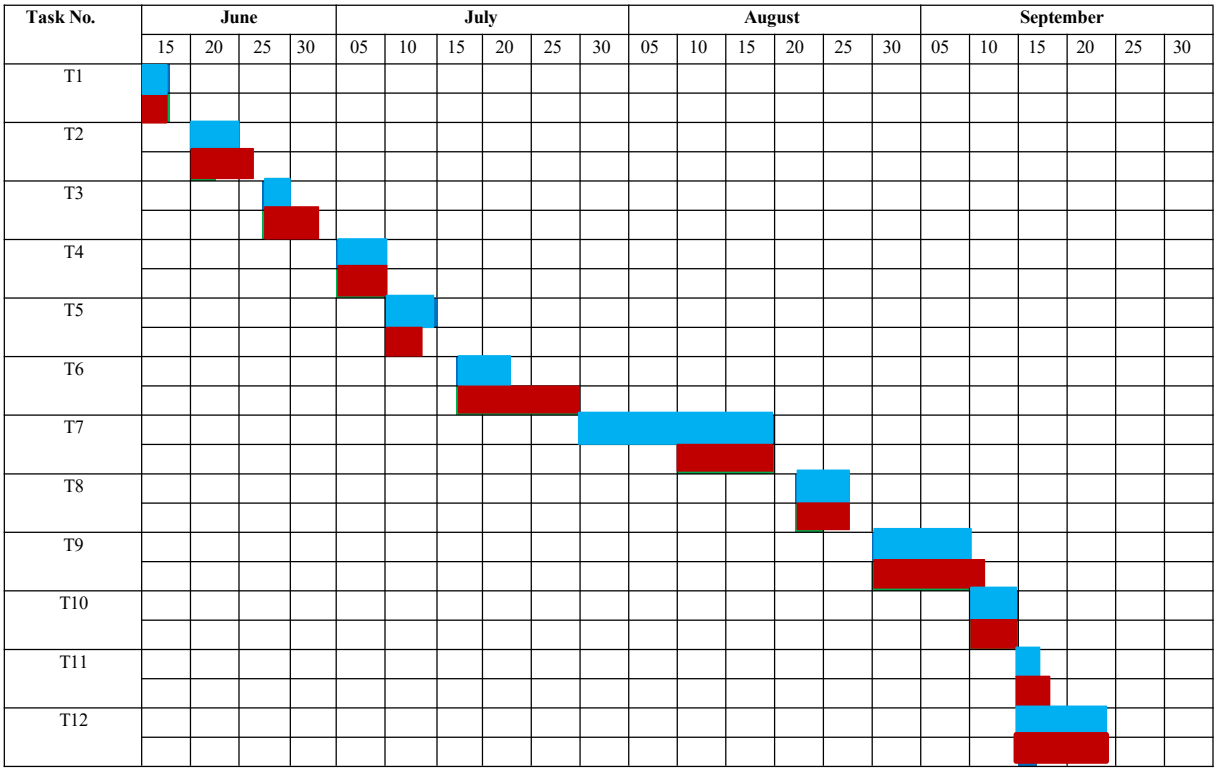


Figure 3.1 Gantt Chart

3.4. Software and Hardware Requirements

3.4.1. Software Requirements

- **Frontend -** HTML, CSS, ReactJS
- **Backend -** ExpressJS
- **Database -** MongoDB
- **Operating System** - Windows 7 or higher, Linux or Mac OS

3.4.2. Hardware Requirements

- **Processor -** Intel Pentinum or higher
- **RAM -** 1GB or more
- **Monitor -** 17 CRT or LCD, Plasma etc.
- **Hard-Disk -** 256 or more (SSD preferable)

3.5. Conceptual Models

3.5.1. Data Model

Data in MongoDB has a flexible schema. Documents in the same collection do not need to have the same set of fields or structure Common fields in a collection's documents may hold different types of data. MongoDB provides two types of data models - Embedded data model and Normalized data model. Based on the requirement, you can use either of the models while preparing your document.

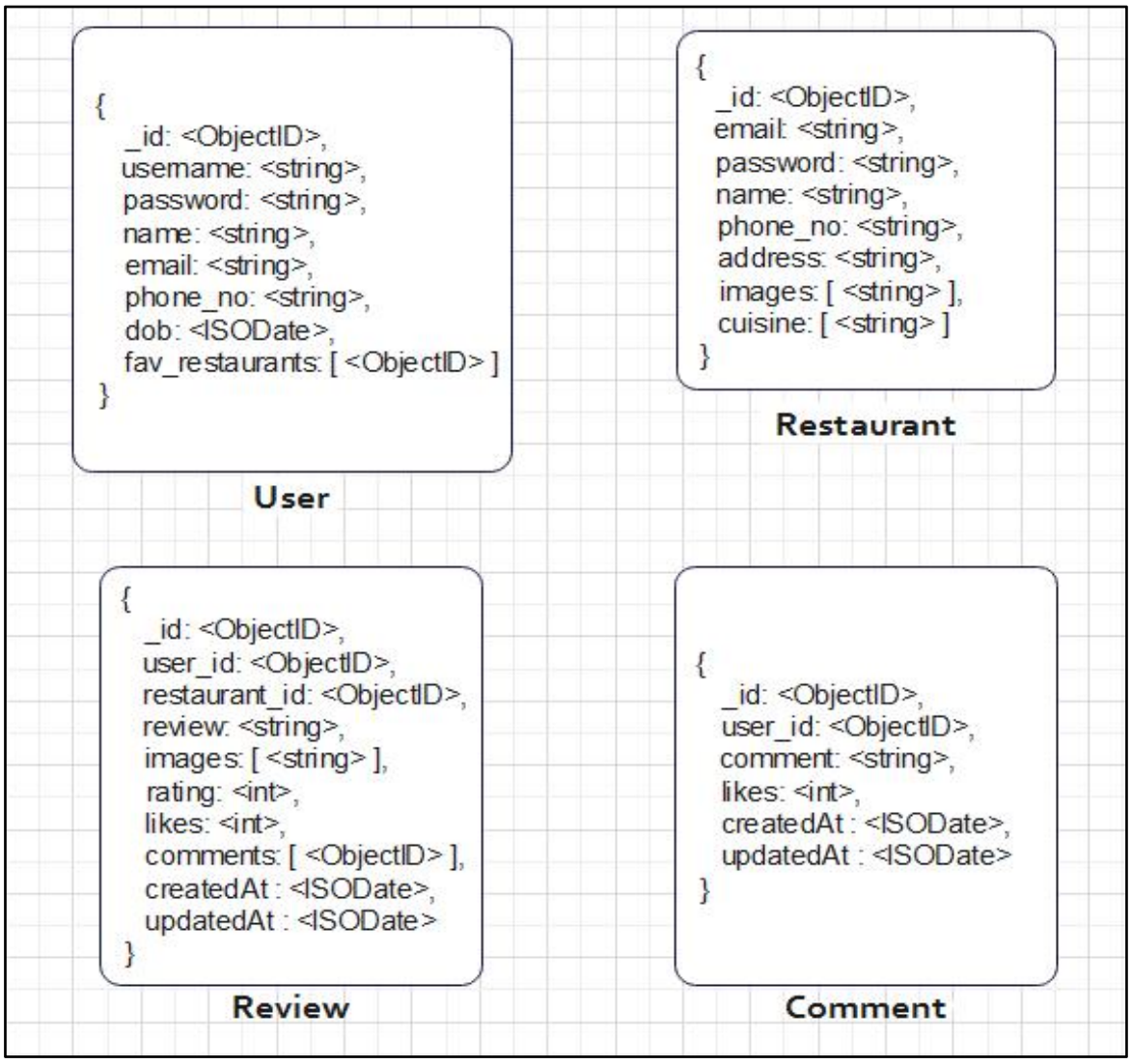


Figure 3.2 Data Model

3.5.2. Data-Flow Diagram

A data flow diagram (DFD) is a visual representation of the information flow through a process or system. DFDs help you better understand process or system operation to discover potential problems, improve efficiency, and develop better processes. They range from simple overviews to complex, granular displays of a process or system.

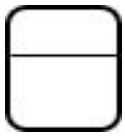
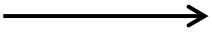
Name	Symbol	Description
Process		A process transforms incoming data flow into outgoing data flow.
Data Store		Data stores are repositories of data in the system.
Data Flow		Data flows are pipelines through which packets of information flow.
External Entity		External entities are objects outside the system, with which the system communicates.

Table 3.2 Data Flow Diagram Symbols

Reference – Lucid Chart (https://www.lucidchart.com/pages/data-flow-diagram)

Diagram:

Context Level

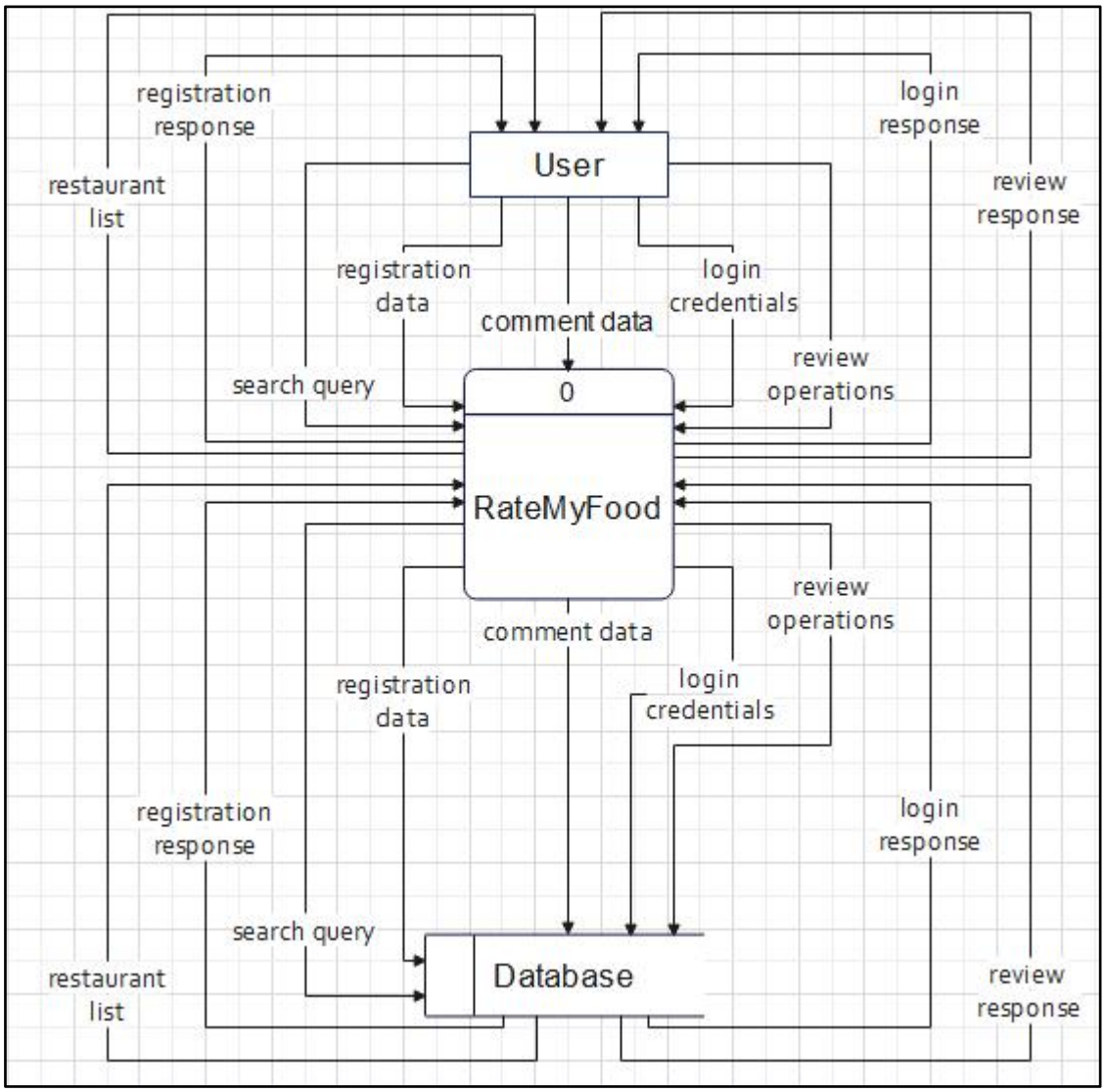


Fig 3.3 Level 0 Data Flow Diagram

Level 1

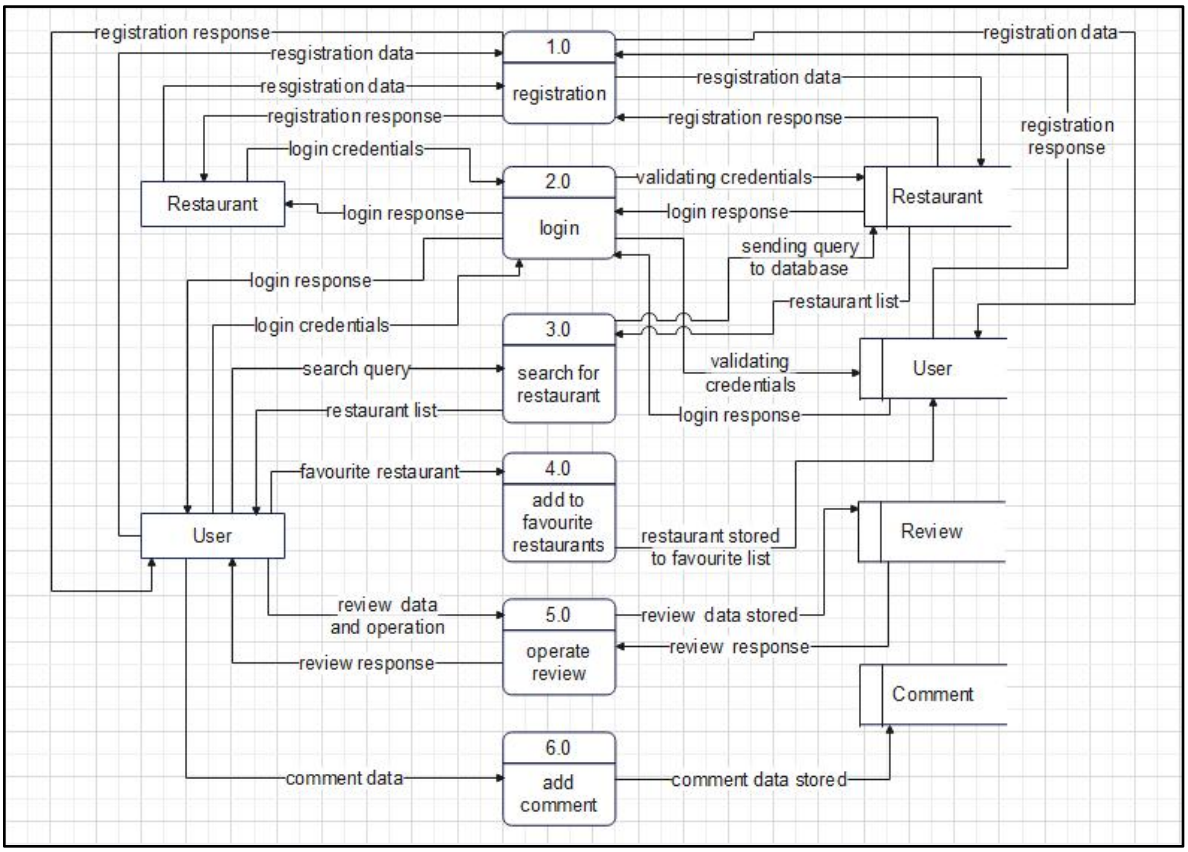


Fig 3.4 Level 1 Data Flow Diagram

Level 2

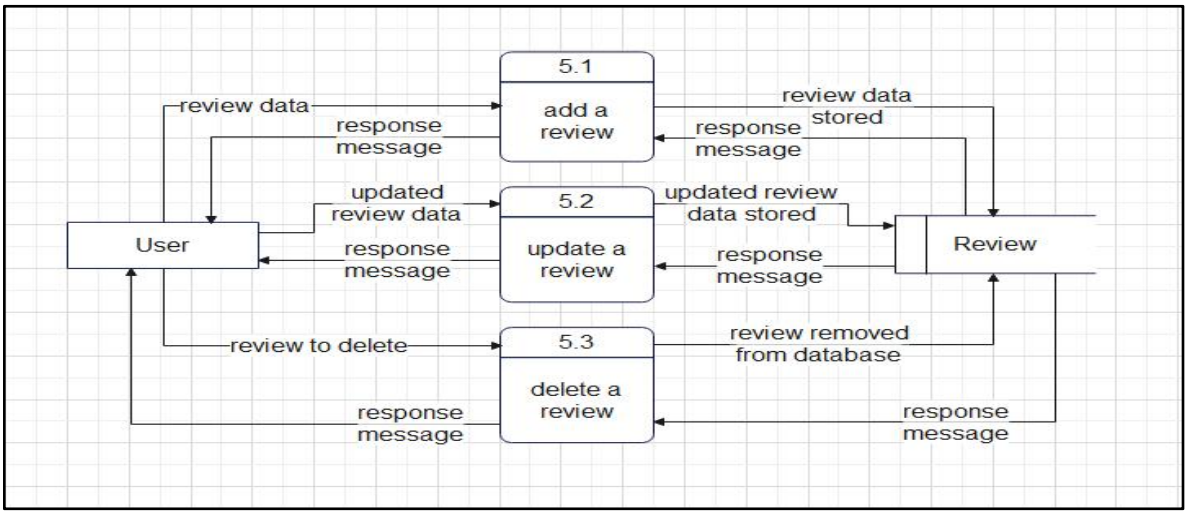


Fig 3.5 Level 2 Data Flow Diagram

3.5.3. Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

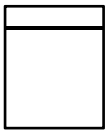
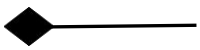
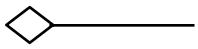
Name	Symbol	Description
Class		This box is used to define a class and the attributes and operation are listed in it along with access modifiers
Public	+	The plus sign states that attribute can be accessed in any other class
Private	-	The dash sign state that attribute cannot be accessed by any other class
Composition		A composition states the one class is totally depended on the other class
Aggregation		An aggregation states that one class is partially associated with the other class
One	1	This describes only one entity participation in the association
Many	1..*	This describes one or more entity participation in the association

Table 3.3 Class Diagram Symbols

Reference – Lucid Chart (https://www.lucidchart.com/pages/uml-class-diagram)

Diagram:

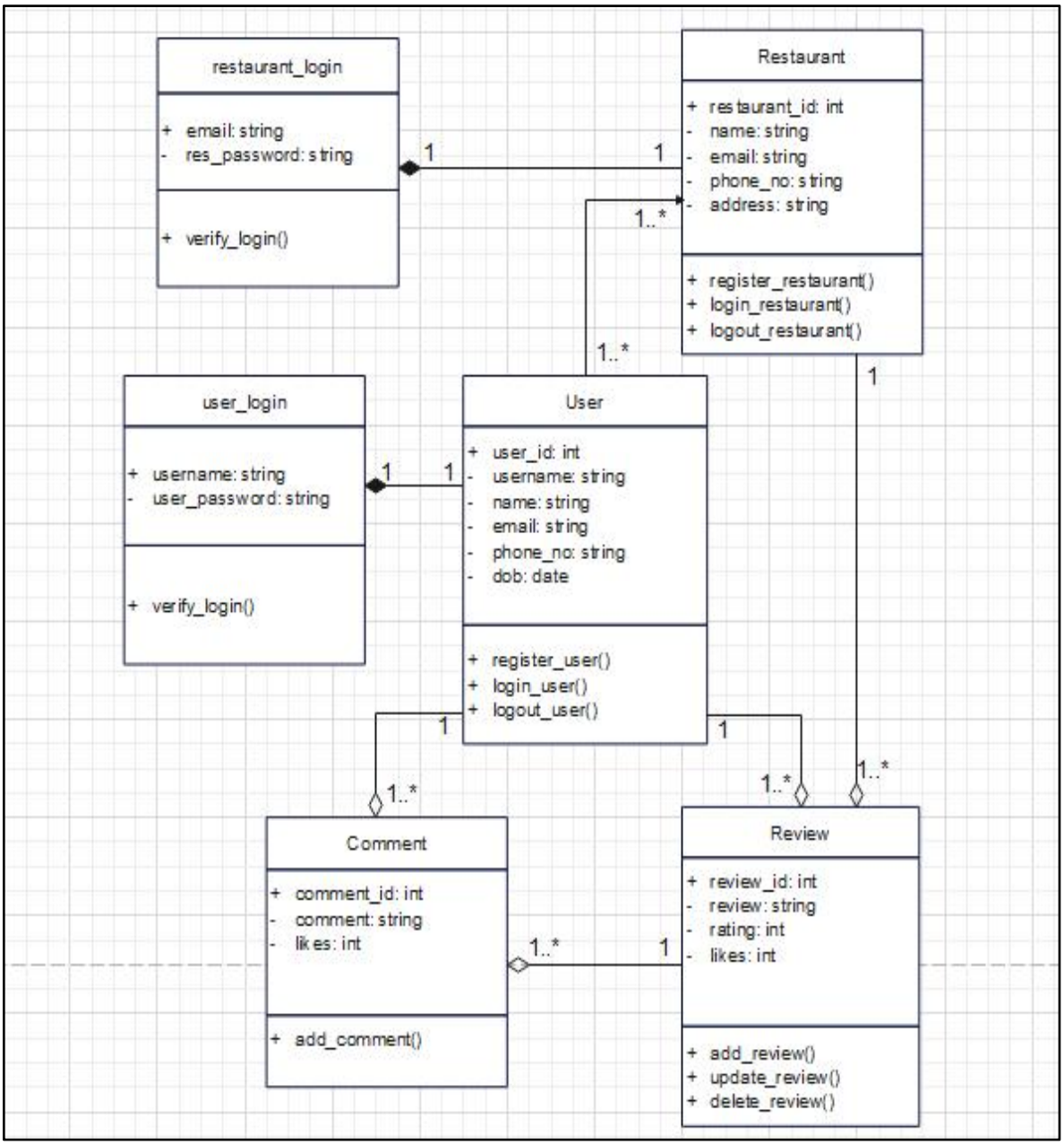


Fig 3.6 Class Diagram

3.5.4. Use Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

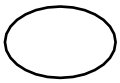
Name	Symbol	Description
Actor		The actors are used to define the environment interacting with the system
Use Case		A Use Case describes the functionality of the system
Association		An association is used to show interaction of actors with use cases
Include	<<include>>	This association states that the base use case is executed with the help of include use case
Extend	<<extend >>	The extend states that the extend use case will be executed after the execution of base use case but it will not always execute

Table 3.4 Use Case Diagram Symbols

Reference – Software Engineering, “Ian Somerville”, 9th Edition, Pearson Education

Diagram:

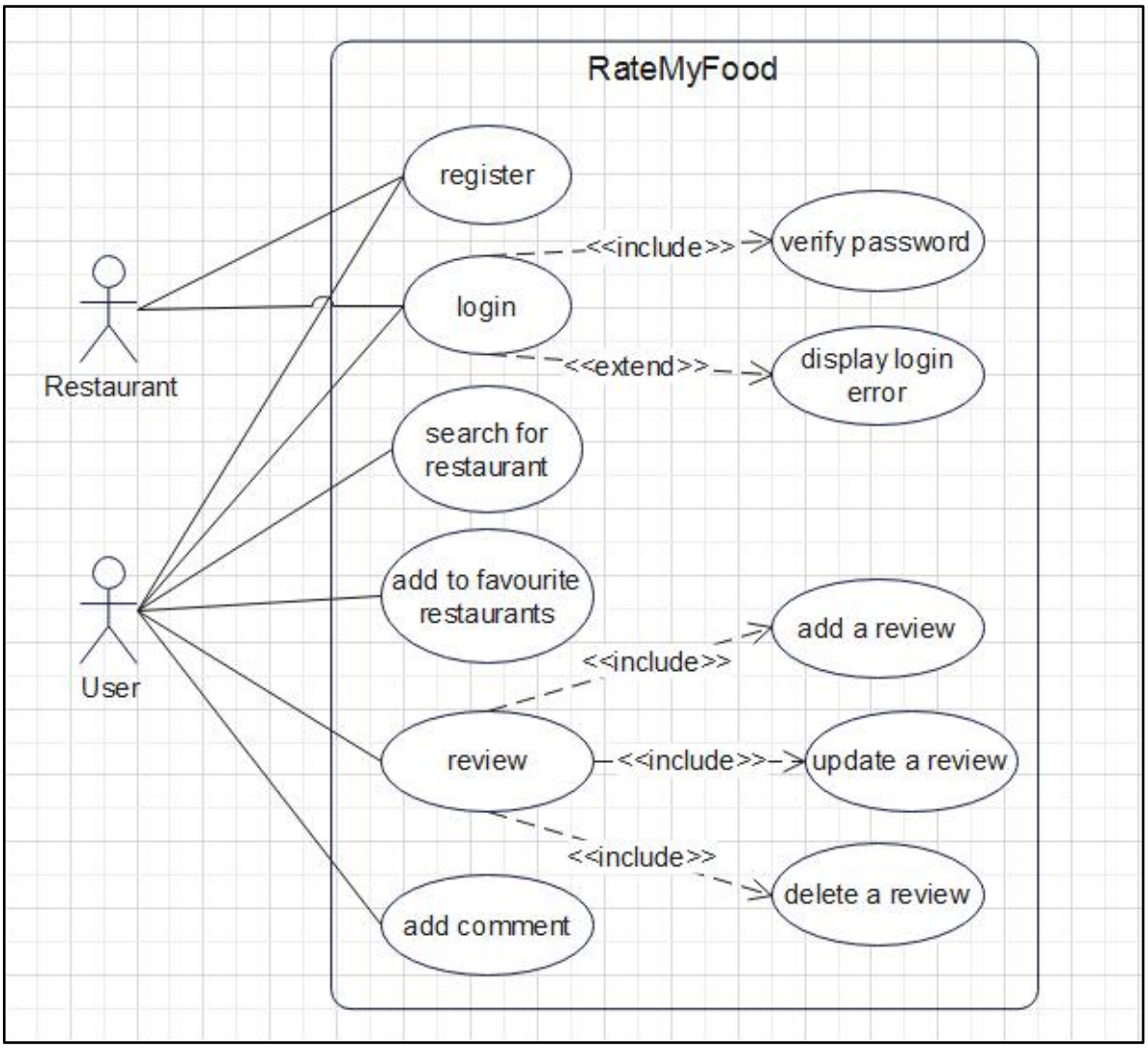


Fig 3.7 Use Case Diagram

Description for Use Cases

1. **Register**
   - **Description –** This will help users and restaurants to create a new account
   - **Actors –** User and restaurant admin
   - **Data –** Information such as name, address, email, etc.
   - **Stimulus** – Request for new account creation
   - **Response** – If all the fields are entered correctly, user will be registered successfully

2. **Login**
    - **Description –** User can enter the login credentials to enter website through their account
    - **Actors –** User and restaurant admin
    - **Data –** Username, email, phone no. and password
    - **Stimulus** – User's and restaurant's request for login and use the website
    - **Response** – If login credentials are correct, user will be directed to their account

3. **Search for restaurant**
    - **Description –** Users can enter some keywords and can find the list of restaurants according to the keywords entered and location of users
    - **Actor –** User
    - **Data –** Name of the restaurant, cuisine, dish name, etc.
    - **Stimulus** – Request for list of restaurant
    - **Response** – The list of restaurants based on the keywords

4. **Add to favourite restaurants**
    - **Description –** This is a feature for users to save profiles of restaurant they like. Users can make a list in their account for their favourite restaurants
    - **Actor –** User
    - **Data –** Selected checkbox
    - **Stimulus** – Request to add a restaurant to favourites list
    - **Response** – Restaurant added to the favourites list

5. **Review**
    - **Description –** This will help users to operate reviews. The user can add, update or delete the review for the choice of their restaurant
    - **Actor –** User
    - **Data –** Review in text and image format, ratings, etc.
    - **Stimulus** – Request to perform operations on review
    - **Response** – If all the operation request are proper, review operations will be carried out successfully or error message will be generated

6. **Add comment**
    - **Description –** This will help users add comment to a particular review
    - **Actor –** User
    - **Data –** comment data
    - **Stimulus** – Request to add a comment on review
    - **Response** – Comment will be added successfully under the review

3.5.5. Sequence Diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

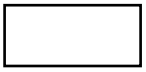
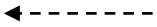
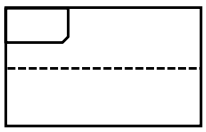
Name	Symbols	Description
Objects		This symbol is used to define the data objects
Actor		The actors are used to define the environment interacting with the system
Lifeline of the Object		The amount of time for which the object instance is involved in the computation.
Message		It is used to show the interactions between actor and objects.
Return Message		It returns a message for a request
Alternative Frame		This frame is used to show two different conditions and their working.

Table 3.5 Sequence Diagram Symbols

Reference – Software Engineering, "Ian Somerville", 9th Edition, Pearson Education

Diagram:

Register

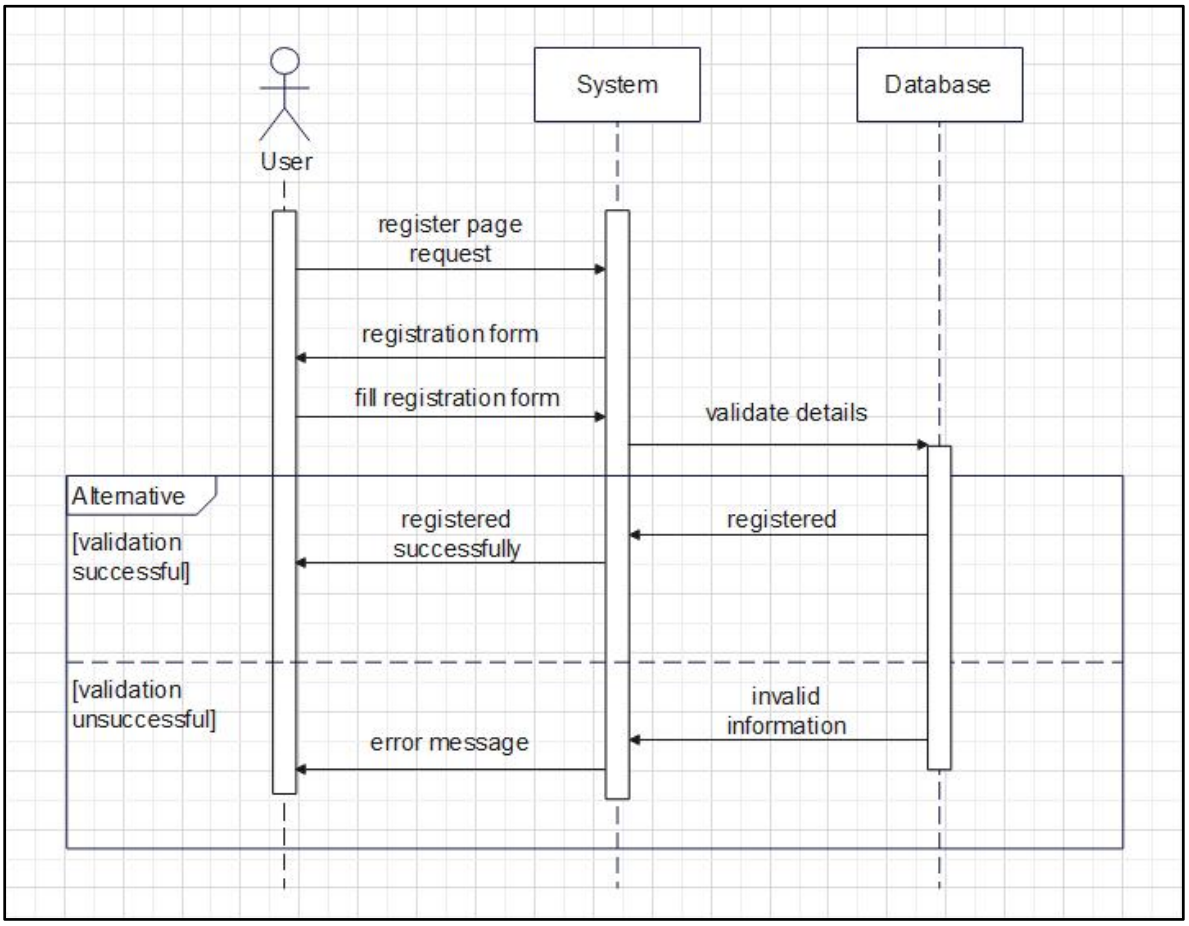


Fig 3.8 Register Sequence Diagram

Login

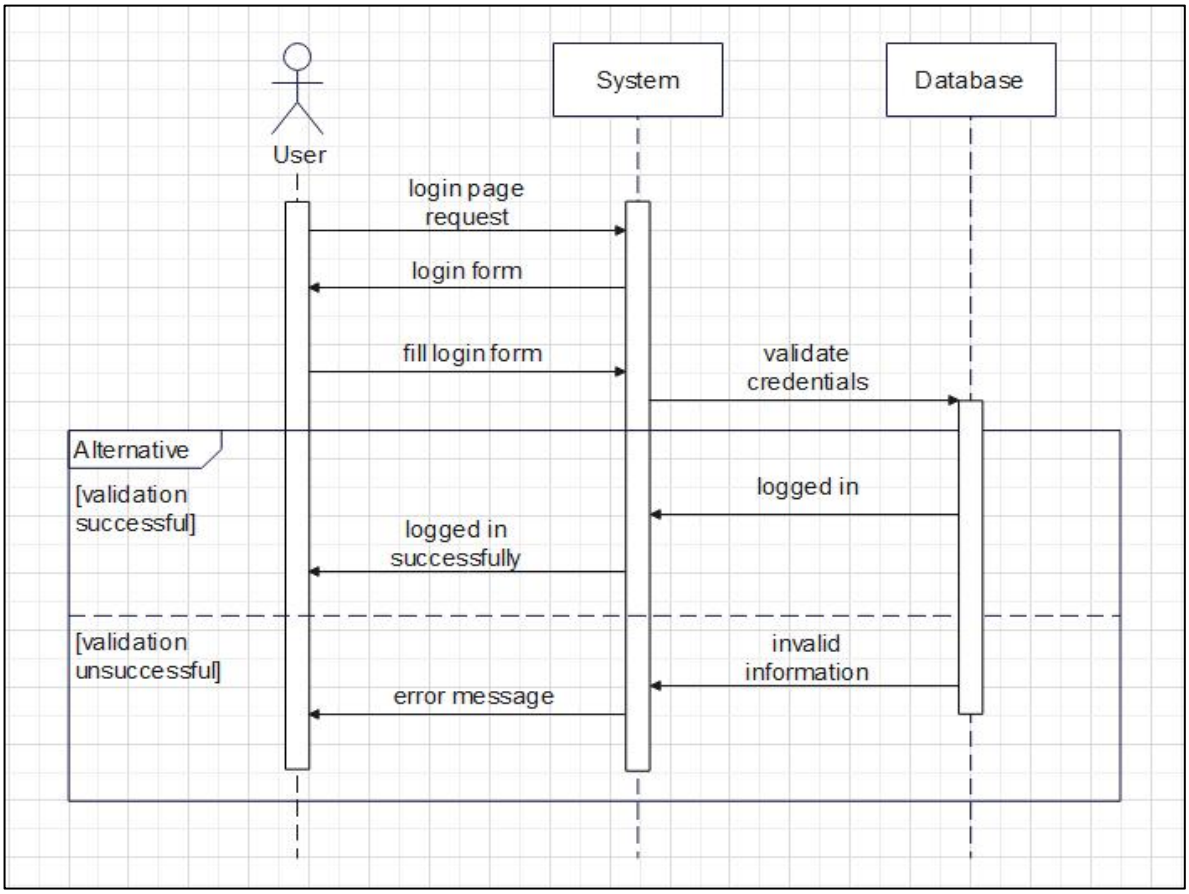


Fig 3.9 Login Sequence Diagram

Search for restaurant

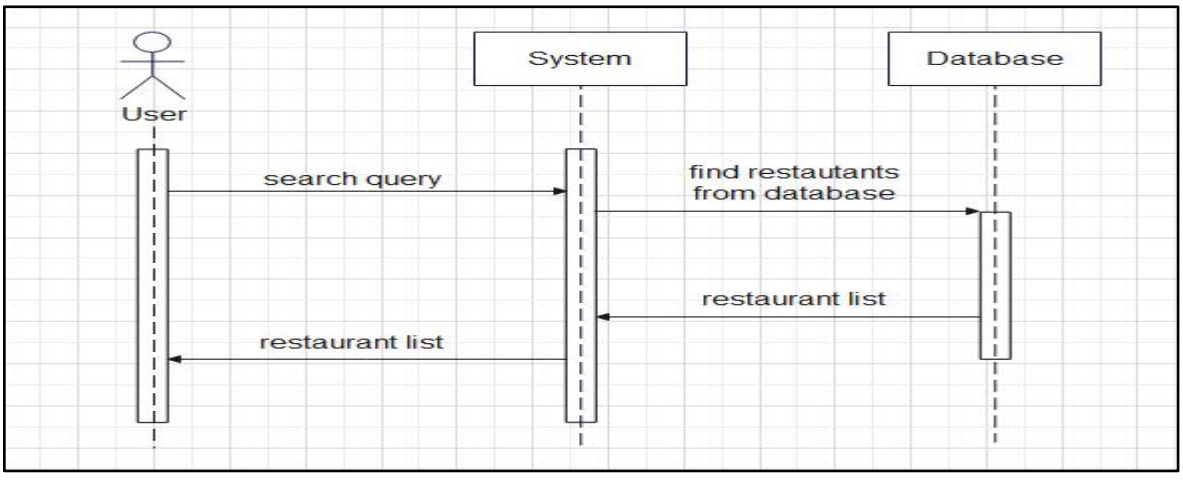


Fig 3.10 Search For Restaurant Sequence Diagram

Add to favourite restaurants

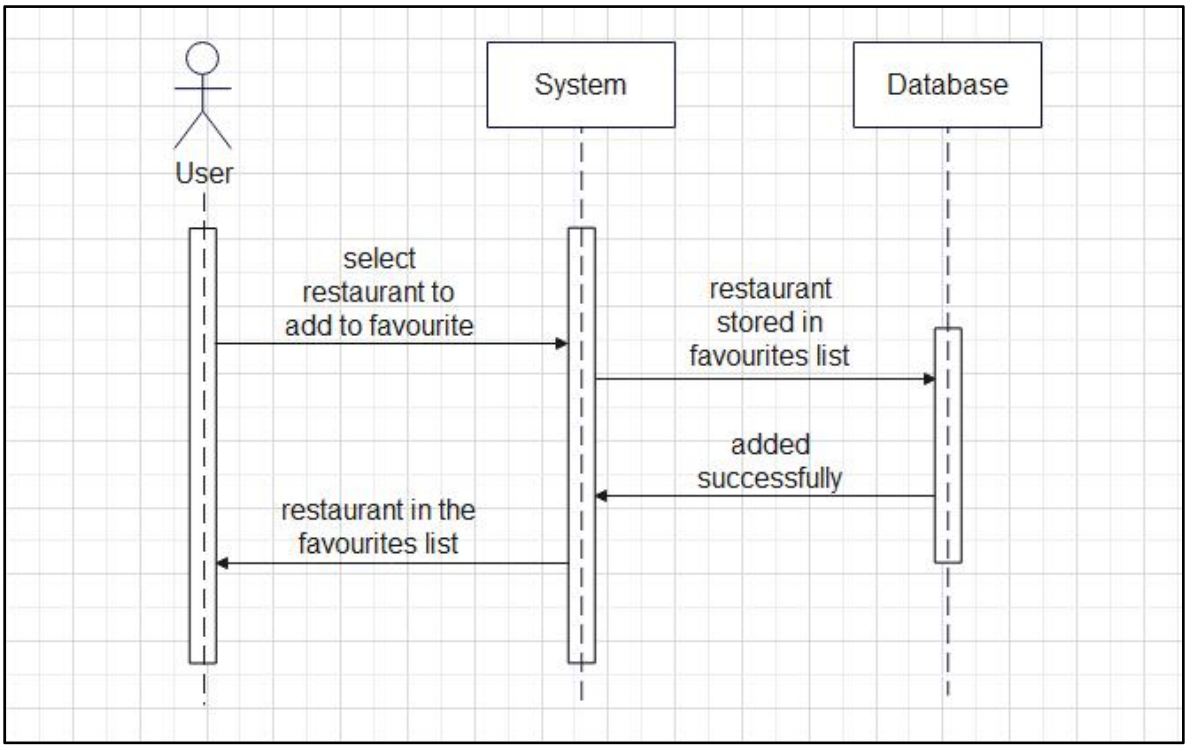


Fig 3.11 Add To Favourite Restaurants Sequence Diagram

Add comment

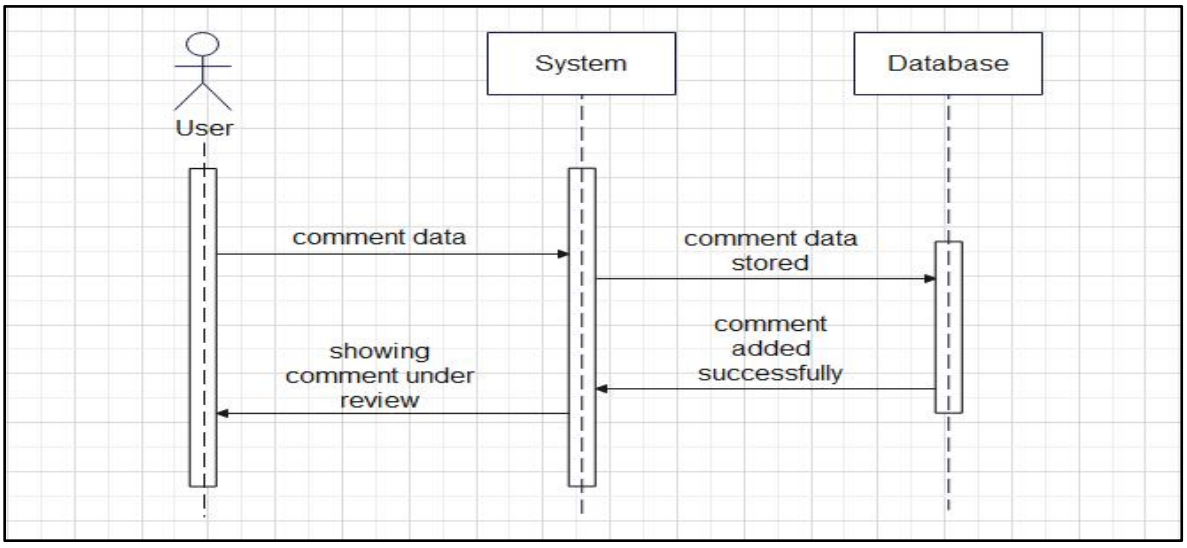


Fig 3.12 Add comment Sequence Diagram

Review

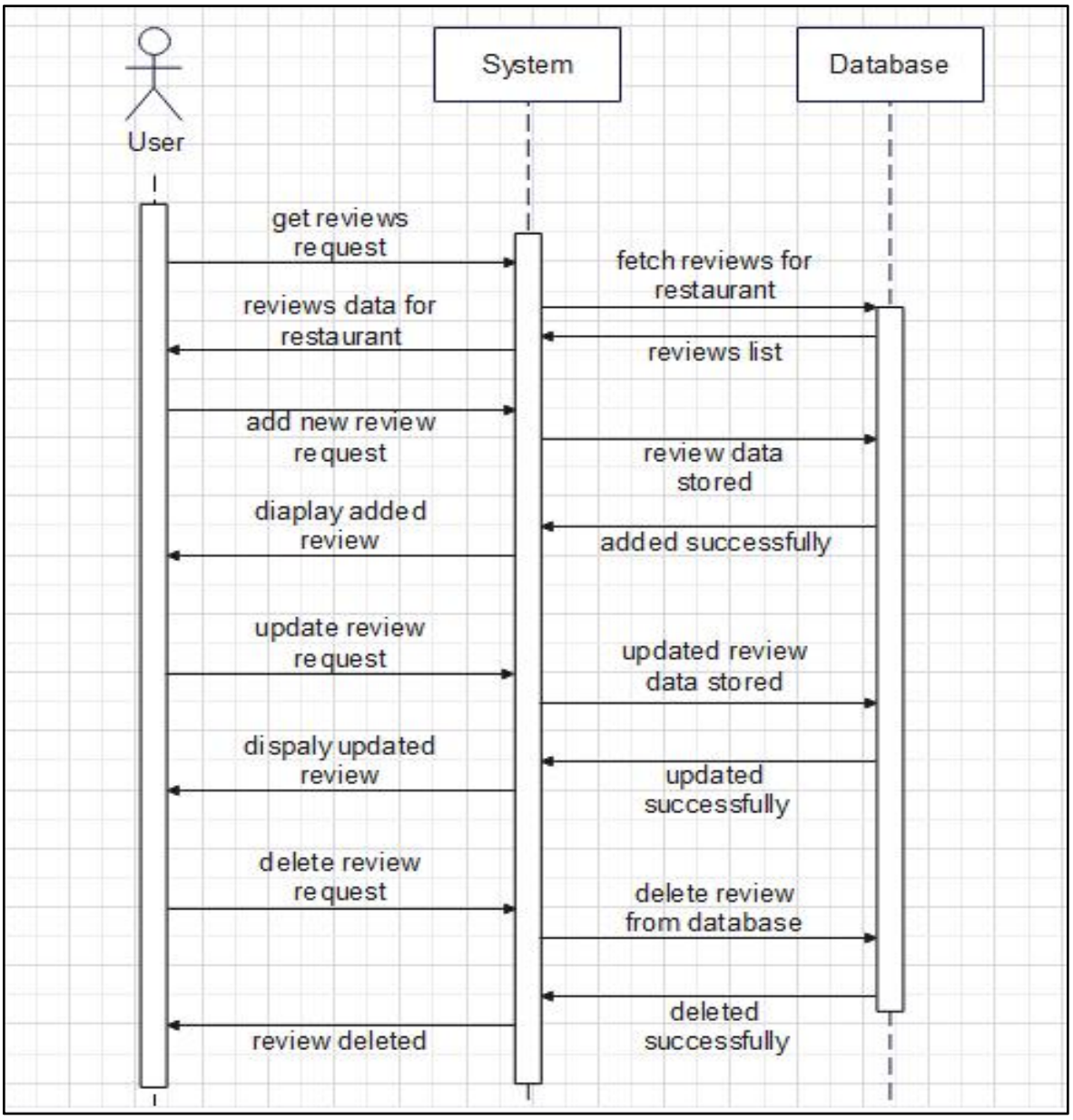


Fig 3.13 Review Sequence Diagram

3.5.6. Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

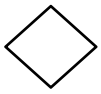
Name	Symbol	Description
Start		This symbol indicates start of an activity
End		This symbol indicates end of an activity
Activity		This symbol indicated the activities of a particular system.
Flow of Work		This symbol is used to show the flow of work from activity to activity
Decision		This symbol represents the decision parameter and used where a decision is need to be taken

Table 3.6 Activity Diagram Symbols

Reference – Software Engineering, "Ian Somerville", 9th Edition, Pearson Education

Diagram:

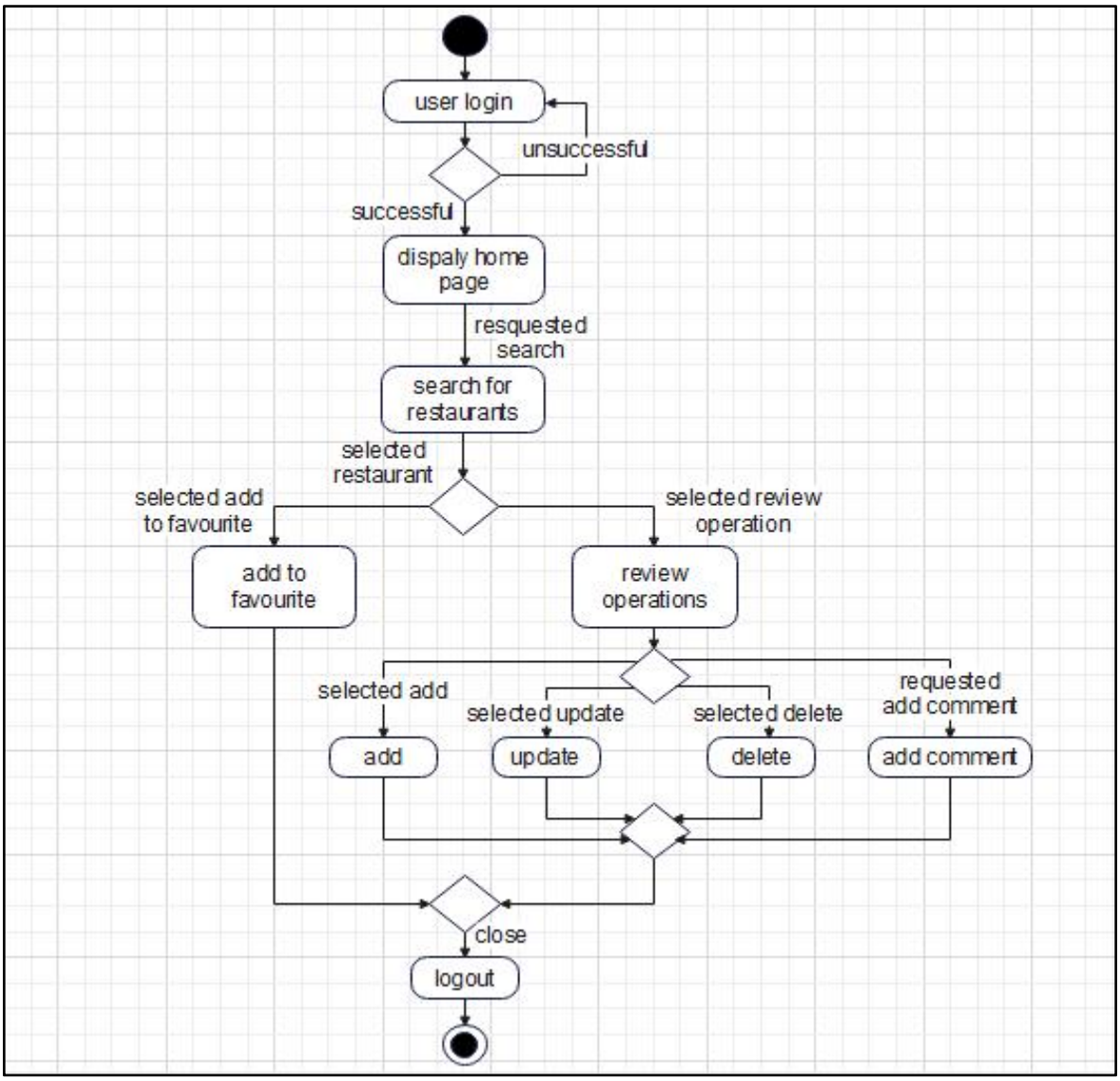


Fig 3.14 Activity Diagram

3.5.7. State-Chart Diagram

State-Chart diagram is one of the 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. Reactive systems can be defined as a system that responds to external or internal events. State-Chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State-Chart diagram is to model lifetime of an object from creation to termination.

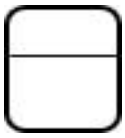
Name	Symbol	Description
Start		This symbol indicates start of an activity
End		This symbol indicates end of an activity
Activity		This symbol indicated the activities of a particular system.
Flow of Work		This symbol is used to show the flow of work from activity to activity

Table 3.7 State-Chart Diagram Symbols

Reference – Software Engineering, "Ian Somerville", 9th Edition, Pearson Education

Diagram:

Register

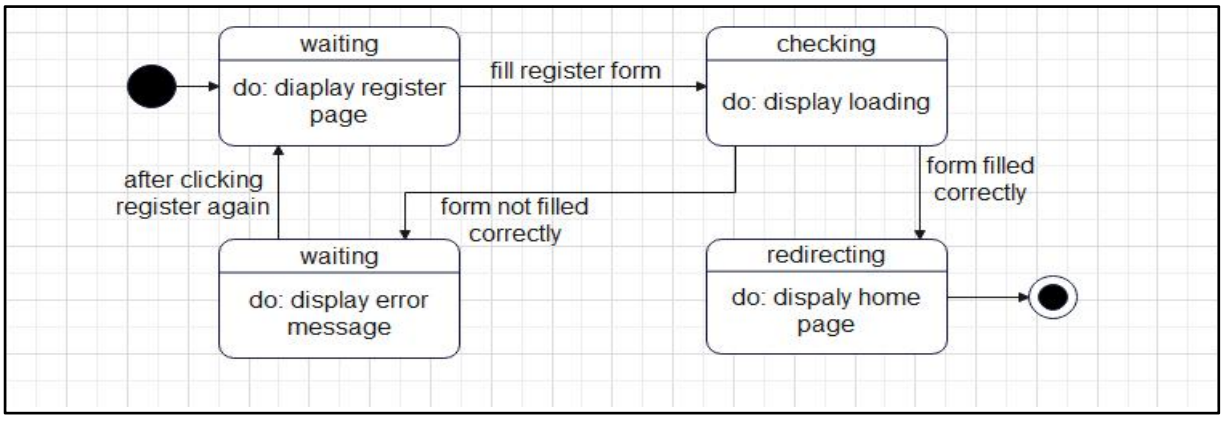


Fig 3.15 Register State-Chart Diagram

Login

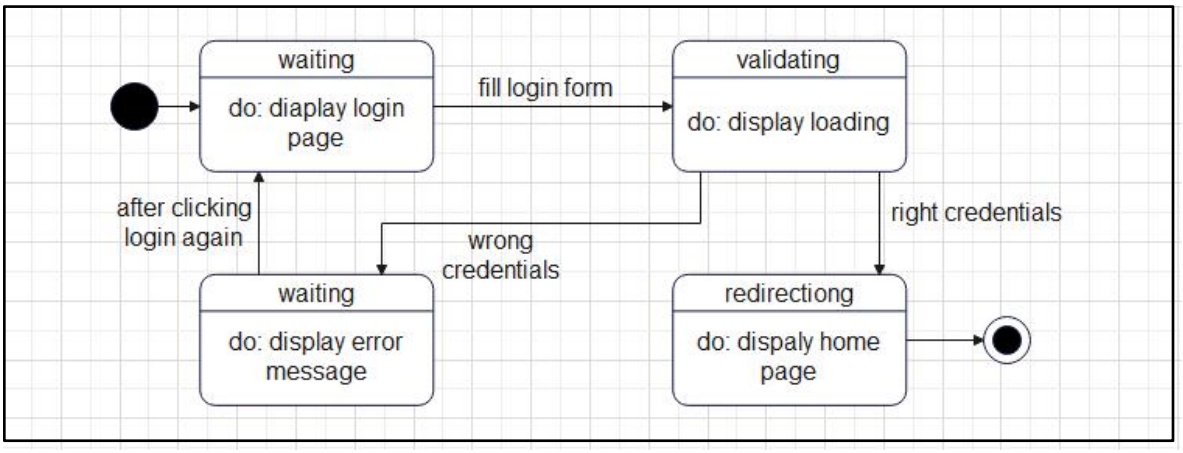


Fig 3.16 Login State-Chart Diagram

Search for restaurants

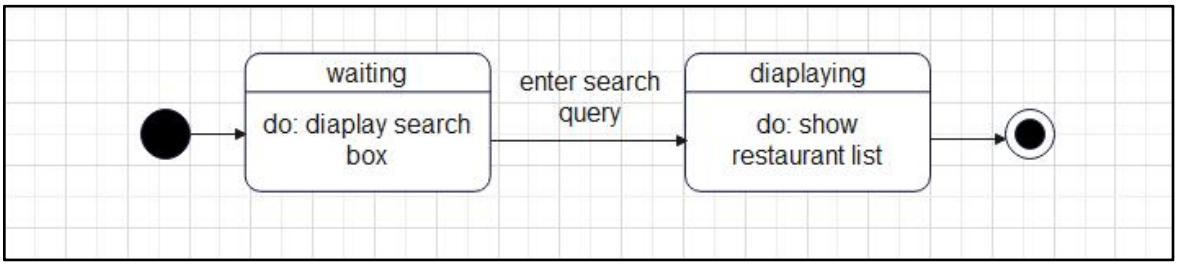


Fig 3.17 Search For Restaurant State-Chart Diagram

Add to favourite restaurants

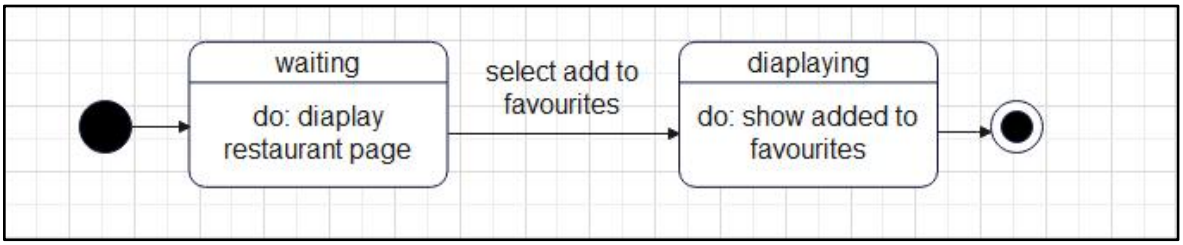


Fig 3.18 Add To Favourite Restaurants State-Chart Diagram

Review

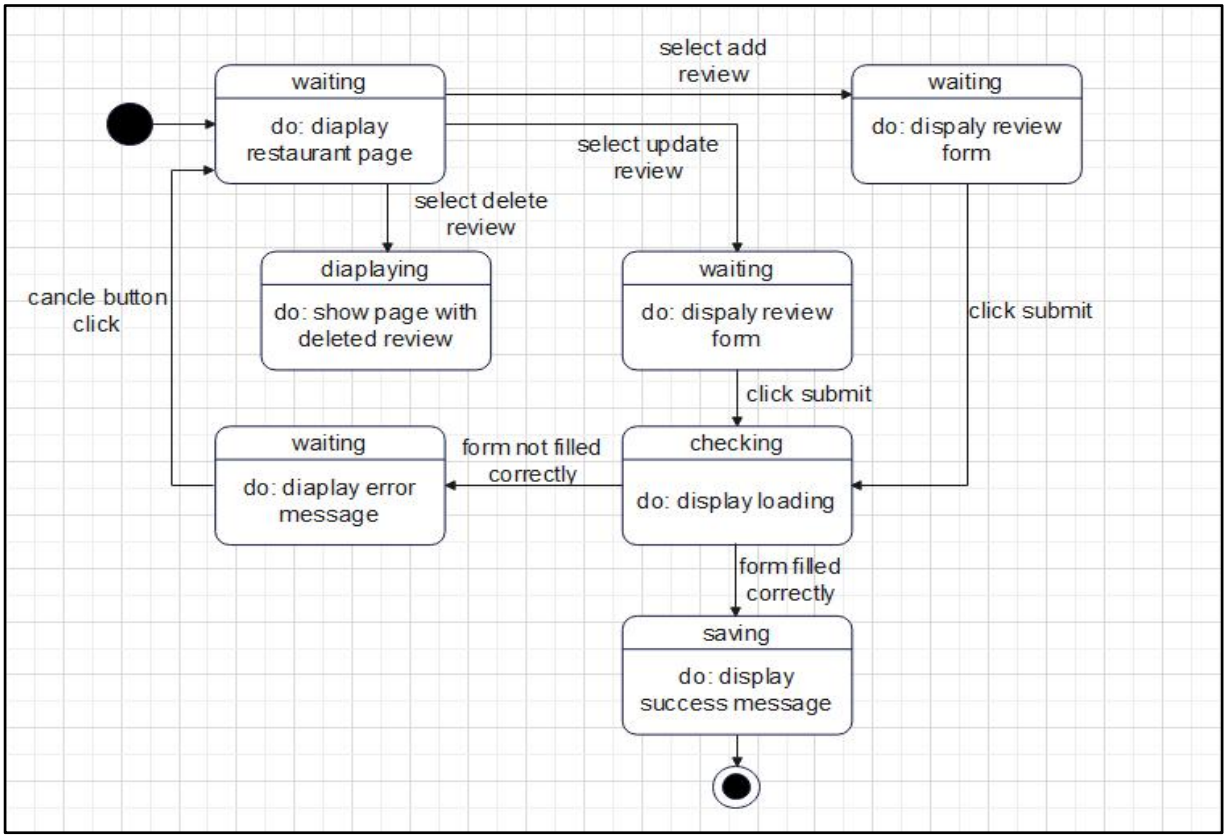


Fig 3.19 Review State-Chart Diagram

Add Comment

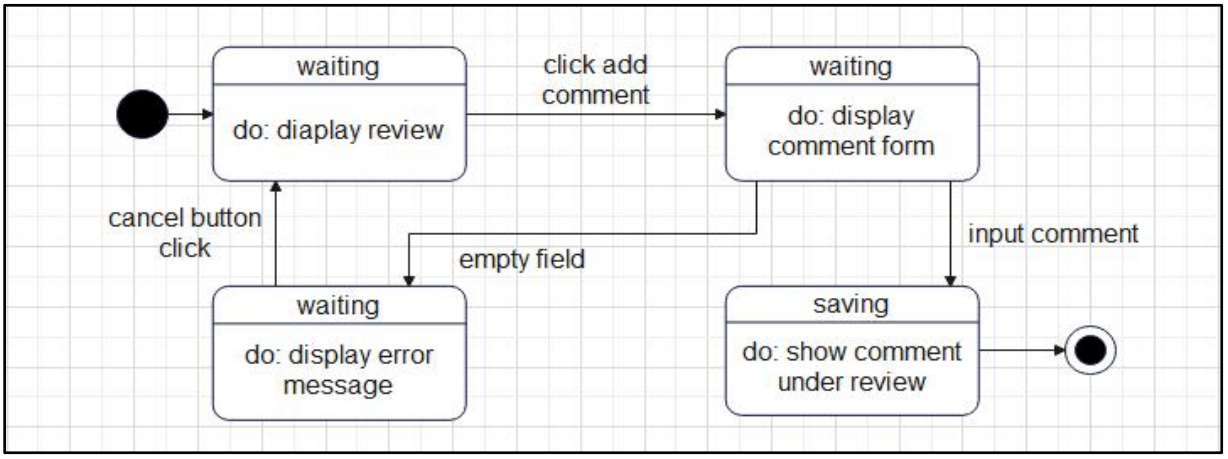


Fig 3.20 Review State-Chart Diagram

Chapter 4:
System Design

4.1. User Interface

Home page before login

Fig 4.1 User Interface Of Home Page Before Login

Login page

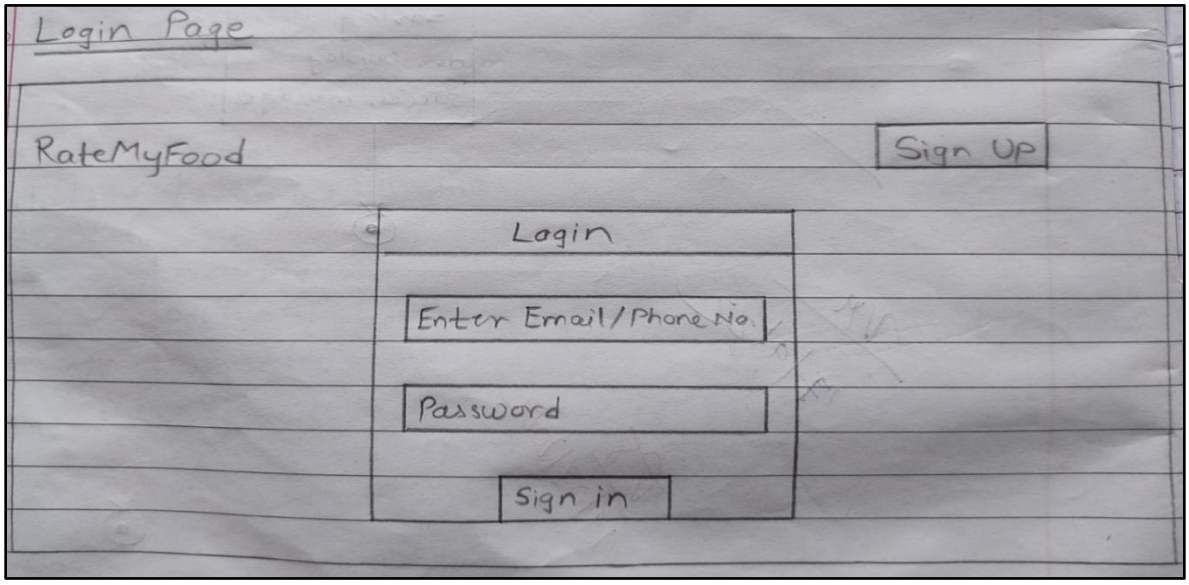


Fig 4.2 User Interface Of Login Page

Register page for restaurant

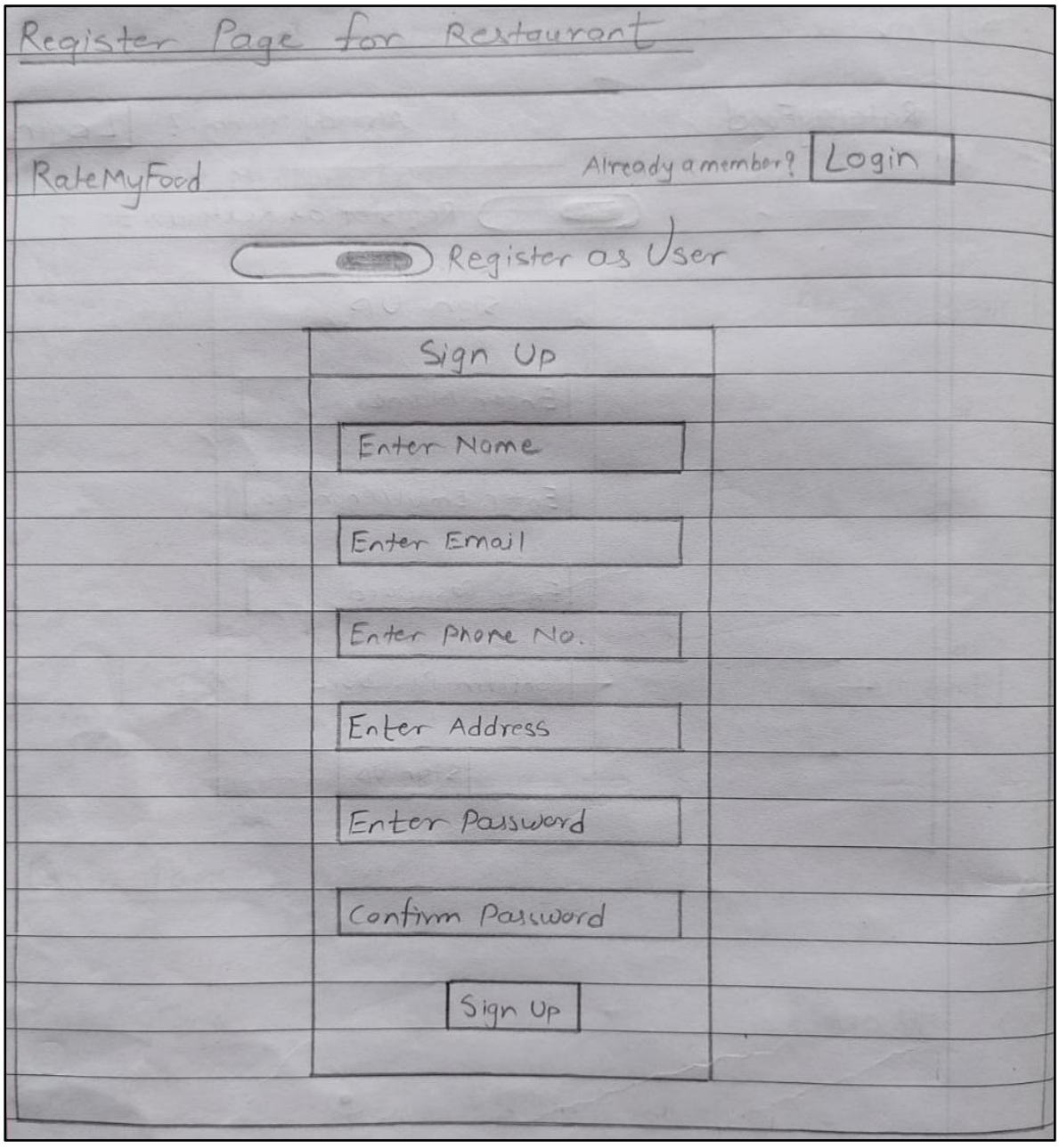


Fig 4.3 User Interface Of Register Page For Restaurant

Register page for user

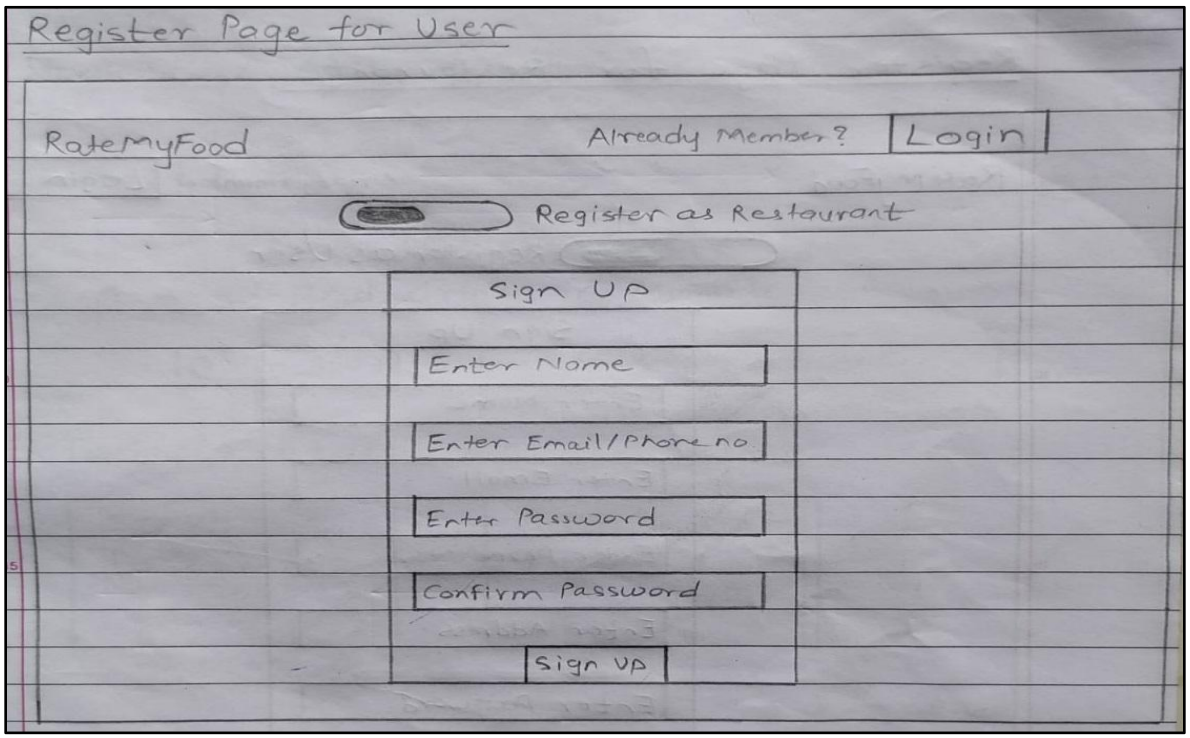


Fig 4.4 User Interface Of Register Page For User

Home page after login

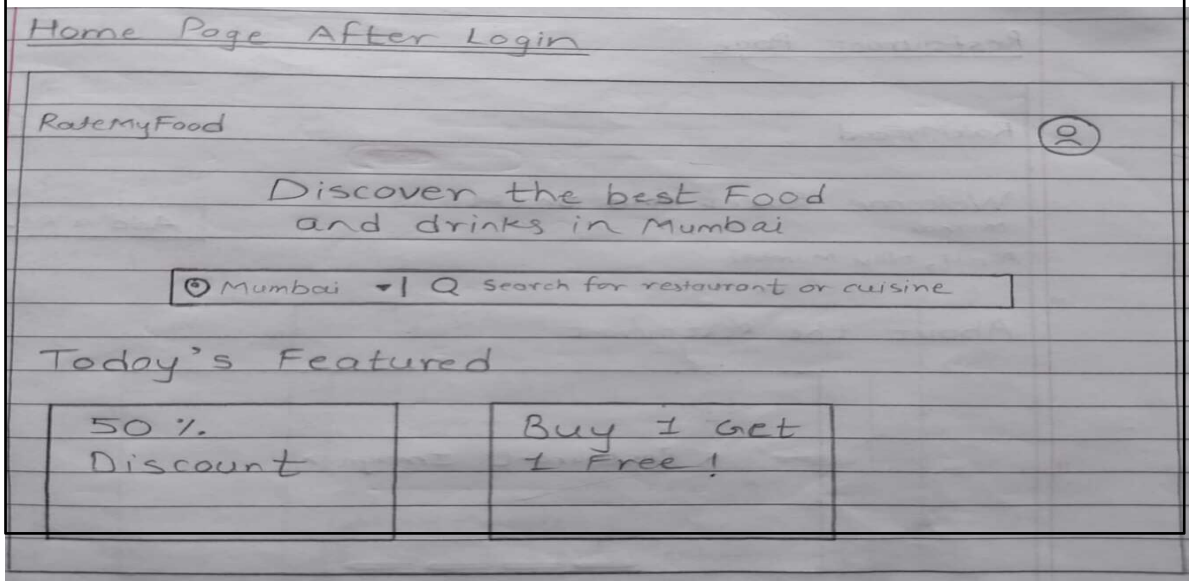


Fig 4.5 User Interface Of Home Page After Login

Home page after searching

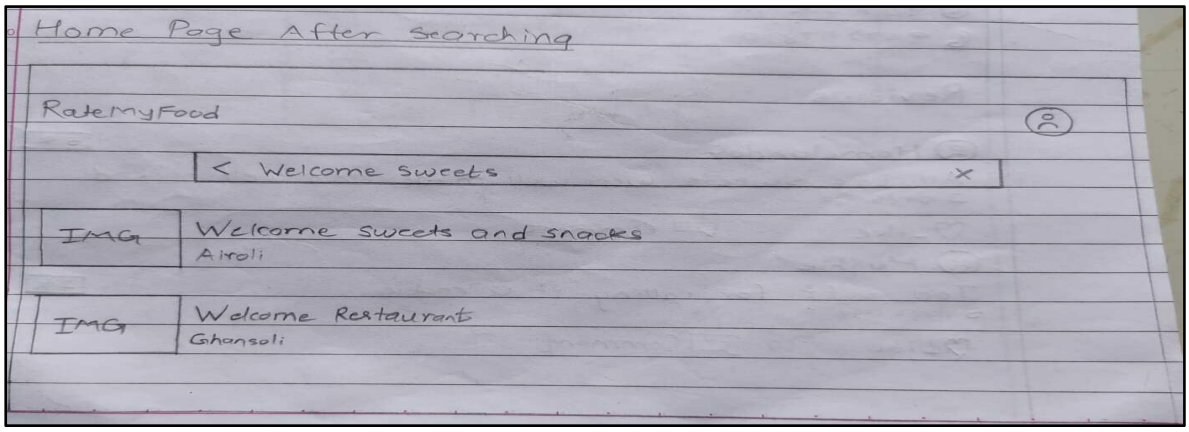


Fig 4.6 User Interface Of Home Page After Searching

Review page

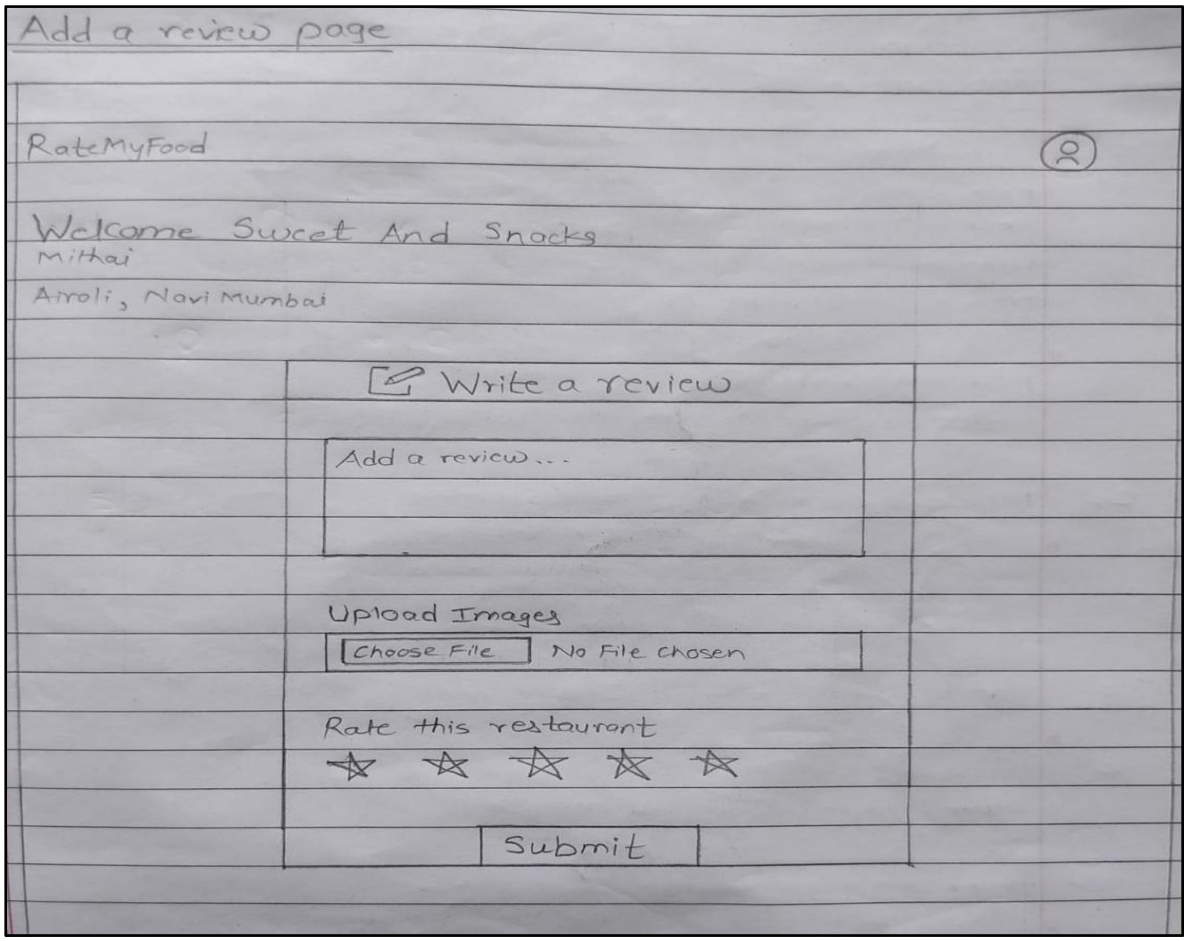


Fig 4.7 User Interface Of Review Page

Restaurant page

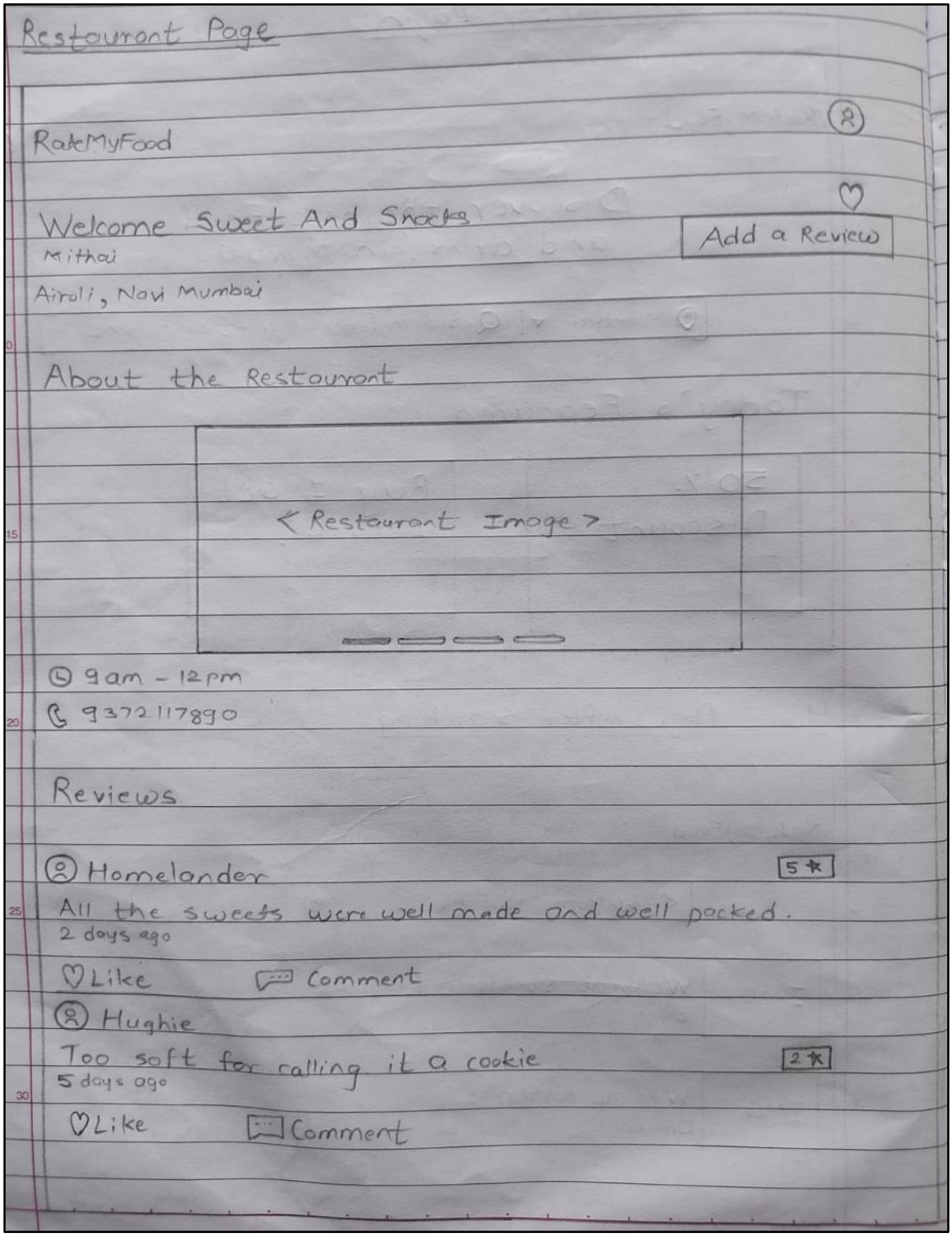


Fig 4.8 User Interface Of Restaurant Page

4.2. Test Cases

1. Registration for users

No.	Test Case	Expected Output	Actual Output	Remark
1	Name: popclaw Email: pop2@gmail.com Username: ATrain Phone No: 7585746379 DOB: 15 July 2007 Password: ATrain@123 Confirm Password : ATrain@123	Registered successfully.		
2	Email: popclow12	Invalid email.		
3	Password: 123456	Password must have 8-10 characters and should contain special character, uppercase and number.		
4	Password: ATrain@123 Confirm Password: 123456	Password doesn't match.		
5	Email: pop2@gmail.com Username: ATrain	User already exist.		
6	Phone No: 155236	Invalid phone no.		
7	Name: same Email: same@gmail.com Username: game	Please fill all the mandatory fields.		

Table 4.1 Test Cases For User Registration

2. Registration for restaurant

No.	Test Case	Expected Output	Actual Output	Remark
1	Name: Welcome Sweets Email: welcom@gmail.com Phone No: 9867564361 Address: Airoli Cuisine: Indian Password: Welcom@123 Confirm Password : Welcom@123	Registered successfully.		
2	Email: welcome12	Invalid email.		
3	Password: 123456	Password must have 8-10 characters and should contain special character, uppercase and number.		
4	Password: Welcom@123 Confirm Password: 123456	Password doesn't match.		
5	Email: welcom@gmail.com	User already exist.		
6	Phone No: 155236	Invalid phone no.		
7	Name: Om Food Mart Email: om@gmail.com	Please fill all the mandatory fields.		

Table 4.2 Test Cases For Restaurant Registration

3. Login

No.	Test Case	Expected Output	Actual Output	Remark
1	Email: welcom@gmail.com Password: Welcom@123	Logged in successfully.		
2	Email: welcom@gmail.com Password: 1232fgQ#	Email or password is wrong.		
3	Email: welcom@gmail.com	Please fill all the fields.		

Table 4.3 Test Cases For Login

4. Search for restaurant

No.	Test Case	Expected Output	Actual Output	Remark
1	welcome	Restaurants named welcome are shown.		
2	vadapav	Restaurants selling vadapav are shown.		
3	null	Please enter restaurant name or cuisine or dish to search.		

Table 4.4 Test Cases For Restaurant search

5. Review

No.	Test Case	Expected Output	Actual Output	Remark
1	Review: taste is bad Image: dish.jpeg Rating: 2	Review added successfully.		
2	Review: good dish Image: dish.jpeg	Please fill all the mandatory fields.		
3	Review: very good food Rating: 4	Review added successfully.		
4	null	Please fill all the mandatory fields.		

Table 4.5 Test Cases For Review

6. Comment

No.	Test Case	Expected Output	Actual Output	Remark
1	Comment: helpful review	Comment added successfully		
2	null	Please add comment.		

Table 4.6 Test Cases For Comment

Bibliography

- Software Engineering, “Ian Somerville”, 9th Edition, Pearson Education.

  (Referred from 29th July 2022 to 23rd September 2022)

- Lucid Chart (https://www.lucidchart.com)

  (Referred from 1st August 2022 to 17th September 2022)

- w3schools (https://www.w3schools.com)

  (Referred from 1st August 2022 to 17th September 2022)

- GeeksforGeeks (https://www.geeksforgeeks.org)

  (Referred from 1st August 2022 to 17th September 2022)

- Tutorialspoint (https://www.tutorialspoint.com)

  (Referred from 1st August 2022 to 17th September 2022)

- Stack Overflow (https://stackoverflow.com)

  (Referred from 1st August 2022 to 17th September 2022)