

Restaurant Recommendation System by Using Data Mining Techniques

Uditha Punchihewa

149226G

Faculty of Information Technology

University of Moratuwa

November 2018

Restaurant Recommendation System by Using Data Mining Techniques

Uditha Punchihewa

149226G

Dissertation submitted by Faculty of Information Technology,
University of Moratuwa, Sri Lanka for the fulfillment of the requirement of the
Honors Degree
of MSc in Information Technology

November 2018

Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student

U.V. Punchihewa

Signature of Student

Date:

Supervised by:

Name of Supervisor

Mr. Saminda Premaratne

Senior Lecture,

Faculty of Information Technology,

University of Moratuwa.

Signature of supervisor

Date:

Dedication

This dissertation is dedicated to my beloved mother, father, wife, family and friends who gave me endless courage and support to achieve my tasks whenever I discouraged.

Acknowledgment

My heartiest thanks should go to my supervisor Mr. Saminda Premaratne for the guidance, assistance, encouragement and providing this opportunity to do the research on this field. Also, sincerely thanks to all my teachers and demonstrators who taught subjects in my MSc IT degree and the things that I learned from many subjects helped me to fulfill the hard task to be a manageable one. In addition to those, I'm taking this opportunity thank my loving parents, my wife and family who helped and encouraged to make this research successful. Last but not least, my sincere thank should go to all my friends who helped me in numerous ways to make this work possible

Abstract

Nowadays we can see many restaurants in different category in different locations. But do you know what the best restaurants there are? Is there a way to identify the best restaurants? With this proposed solution, we can identify the best restaurants around you that is called “Restaurant recommendation system”. It provides a good experience to the end user to find the best restaurant for their preferences.

Based on previous user experience or feelings about the restaurant we can get some idea about the restaurant. If someone recommends a system, we can think of it as a good restaurant. In this system takes three user inputs on restaurant menu items. The first one is recommending or not recommending as direct input on a restaurant. The second input is rating on a restaurant menu item scaling one to five. The third input is analyzing users’ reviews about the restaurant menu item. So, we can apply data mining techniques like opinion mining and Apriori algorithm on the user inputs and get the output as recommended menu item along with the restaurant.

Based on those three user inputs above mentioned we could decide what the best restaurant meals are by calculating average ratings and ranking them. Not only that. This system will also provide suggestions about restaurants or meals of a restaurant based on users’ preferences by using data mining techniques like collaboration filtering. The collaboration filtering is focused on users’ preferences. And, it uses opinion mining techniques to analyze reviews. This system is a web application with the REST service. For the front-end development used Angular to make it rich application and for back end uses Java with REST web service. In addition to that uses spring framework and hibernate for object relational mapping.

Table of Contents

Introduction.....	1
1.1 Prolegomena.....	1
1.2 Background and Motivation.....	1
1.3 Problem Statement.....	2
1.4 Hypothesis.....	2
1.5 Aim and Objectives	2
1.6 Approach of Recommending Restaurant	3
1.7 Structure of the Thesis	4
1.8 Summary.....	4
Achievements and Challenges in Recommendation of Restaurants.....	5
2.1 Introduction.....	5
2.2 Early Development	5
2.3 Collaboration Filtering.....	7
2.4 Opinion Mining Techniques.....	7
2.5 Modern Trends in Data Mining	8
2.6 Future Challenges of Data Mining.....	8
2.7 Problem Definition	8
2.8 Summary.....	8
Technologies	9
3.1 Introduction.....	9
3.2 Java.....	9
3.3 Angular	9
3.4 REST	9
3.5 Spring	10
3.6 Hibernate.....	10
3.7 MySQL	10
3.8 Weka.....	10
3.9 Summary.....	11
Novel Approach to Restaurant Recommendation.....	12
4.1 Introduction.....	12
4.2 Hypothesis.....	12
4.3 Recommendation Engine	12
4.4 Apriori Algorithm for Associations	13
4.4.1 Create Relationship Table Food Item Against User	13
4.4.2 Calculate Support for Food Item	14
4.4.3 Filter High Support Food Items.....	14
4.4.4 Filter High Support Food Item and Make Association	14
4.5 Sentimental Analysis	15
4.5.1 Apply Bag of Words.....	16
4.5.2 Recursive Neural Network (RNN)	16
4.6 Users	17
4.7 Input.....	17
4.8 Output	18
4.9 Process	18
4.9.1 Login Process	18
4.9.2 Recommendation Process	18
4.9.3 Rating Process.....	18
4.9.4 Review Process	19
4.9.5 Final Rate Calculation Process	19
4.10 Features.....	19
4.11 Data Collection.....	19
4.11.1 Business JSON.....	20

4.11.2 Review JSON	22
4.11.3 User JSON	22
4.12 Storing Data	23
4.13 Summary	23
Design of Restaurant Recommendation	24
5.1 Introduction	24
5.2 Top Level Design	24
5.3 Web Application	24
5.4 Web Service	25
5.5 Database	25
5.6 Web Container	25
5.7 Java Tool	25
5.8 Detailed Design	26
5.9 Main Modules	26
5.9.1 Login Module	27
5.9.2 User Module	27
5.9.3 Restaurant Module	28
5.9.4 Dashboard	28
5.9.5 Search Module	28
5.9.6 Recommendation Engine	29
5.10 Summary	29
Implementation	30
6.1 Introduction	30
6.2 Overall Solution	30
6.3 Collecting, Reading and Storing Data	30
6.4 Sentimental Analysis	34
6.5 Apriori Algorithm	35
6.5.1 Definition	36
6.5.2 Prepare Data Set to Apply Apriori Algorithm	36
6.5.3 Generate ARFF File for Association Rules	37
6.5.4 ARFF File	37
6.5.5 Generate Association Rules Using ARFF	38
6.6 The Ways of Recommending Restaurant Meals	40
6.6.1 Recommend Restaurant Meals Based on User Recommendations	41
6.6.2 Recommend Best Restaurant Meals	41
6.7 Summary	42
Evaluation	43
7.1 Introduction	43
7.2 Evaluation of Web Application	43
7.3 Evaluation of Final Rating	43
7.4 Evaluation of Opinion Mining	45
7.5 Evaluation of Restaurant Menu Item Final Rating	46
7.6 Evaluation of Restaurant Menu Item on User Recommendations	47
7.7 Evaluation of Restaurant Menu Item on User Ratings	48
7.8 Evaluation of Restaurant Menu Item on User Reviews	49
7.9 Summary	49
Conclusion	50
8.1 Introduction	50
8.2 Conclusion	50
8.3 Achievement of Objectives	50
8.4 Further Works	51
8.4.1 Content Based Filtering	52
8.5 Summary	52
Appendix A	54
1. User Login Page	54
2. User Registration Page	55
3. Restaurant Creation Page	56

4. Restaurant View Page.....	57
5. Restaurant List Page.....	58
6. Menu Item List Page.....	59
7. Menu Item Page.....	60
8. Menu Item Creation Page.....	61
Appendix B.....	62
1. User Table.....	62
2. Restaurant Table.....	62
3. Menu Table.....	63
4. Menu Item Table.....	63
5. Restaurant Controller.....	64
6. User Controller.....	64
7. AprioriUtil.....	65
8. SentimentUtil.....	66

List of Figures

Figure 2.3.1 – Collaboration Filtering.....	7
Figure 4.4 – Apriori Algorithm.....	13
Figure 4.5 – Sentimental Analysis.....	15
Figure 4.5.2 – Recursive Neural Network (RNN).....	16
Figure 5.2.1 – Top Level Design.....	24
Figure 5.8.1 – Detailed Design.....	26
Figure 5.9.1.1 – Login Module.....	27
Figure 5.9.5.1 – Search Module.....	28
Figure 6.3.1 – Business JSON.....	30
Figure 6.3.2 – Restaurant Table.....	32
Figure 6.5.2.1 – Associations As CSV.....	36
Figure 6.5.3.1 – Weka Tool for Associations.....	37
Figure 6.5.4.1 – ARFF File.....	38
Figure 6.5.5.1 – Association Rules.....	39
Figure 7.3 – Restaurant Average Rating on Specific Time Period.....	43
Figure 7.5.1 – Fried Rice of Messing Restaurant.....	46
Figure 7.6.1 – Restaurant Menu Item Recommend Data.....	47
Figure 7.7.1 – Restaurant Menu Item Rate Data.....	48
Figure 7.8.1 – Restaurant Menu Item Review Data.....	49
Figure 8.4.1.1 – Content Based Filtering.....	52
Figure Appendix A – User Login Page.....	54
Figure Appendix A – User Registration Page.....	55
Figure Appendix A – Restaurant Create Page.....	56
Figure Appendix A – Restaurant View Page.....	57
Figure Appendix A – Restaurant List Page.....	58
Figure Appendix A – Menu Item List Page.....	59
Figure Appendix A – Menu Item Page.....	60
Figure Appendix A – Menu Item Creation Page.....	61
Figure Appendix B – User Table.....	62
Figure Appendix B – Restaurant Table.....	62
Figure Appendix B – Menu Table.....	63
Figure Appendix B – Menu Item Table.....	63

Introduction

1.1 Prolegomena

In this modern world, there are many restaurants nearby. But it is difficult to decide which restaurants are the best. Which meal is the best? So, the end user can use the restaurant recommendation system to get some idea about the restaurant based on previous users' experiences in that restaurant. This system gets some inputs about the restaurant such as recommend / not recommend, rating scale one to five and reviews. After getting those inputs, the system will analyze them and rate the restaurant. Based on the final rating the other users can get an idea which restaurant is the best. Not only that, the same approach we can apply to the meals of a restaurant and rank them according to the users' inputs. Then the end user can decide which restaurant which meal they should buy. Though it is very useful to have a restaurant recommendation system to the end user there are few systems are available.

1.2 Background and Motivation

When we have family gatherings, birthday parties, get together or any other celebrations will end up with delicious food. What if the restaurant is not good? What if the meal is not delicious? Your celebration will be ruined. Right? Is there are a way or a system to find out the best restaurant? If yes that will help you to find the best restaurant. How if it is listed up all the necessary information such as menu, location, facilities, etc.? Your life is easier, right? Then this Restaurant recommendation system will give the solution for you.

In this system, the end user can see what the recommended restaurants in your area are. What are the recommended meals they have? What are the facilities such as dine-in, take away, wireless Internet, car parking and so on. Not only that, but it also

provides the prices of meals, the location of the restaurant, a comparison of two meals for user convenience.

1.3 Problem Statement

Today we can see many restaurants around you. Do you know what is the best restaurant based on your preferences near you? You may ask for information about a restaurant from your friend, from your neighbor or from your relatives. Based on their recommendation you can visit the restaurant. But that restaurant may not suit you. Because different people have different preferences or taste. So, recommending a restaurant is the challenge topic nowadays. Even though significant numbers of researches have been conducted around this area it won't satisfy user needs all the times. So, the needs of the restaurant recommendation system is very high in people thoughts. Since the Internet facility spread all over the world and if we can provide recommendations to the users' door step it will help to the end user.

1.4 Hypothesis

Recommend a restaurant or a meal can be done combining collaborative filtering and opinion mining techniques.

1.5 Aim and Objectives

The aim of this proposed system is to provide necessary recommendation about a restaurant or a meal based on users' preferences. Following are the main objectives.

- To critically review previous recommendation systems
- To develop new algorithm combining data mining techniques
- To collect information about restaurants
- To collect information about restaurant from users
- To rate restaurant meals and restaurants based on users' inputs
- To suggest restaurants or restaurant meals based on the algorithm according to users' preferences
- To give direction to the restaurant

1.6 Approach of Recommending Restaurant

In order to recommend a restaurant or a meal the system needs to take user inputs from many ways. These inputs apply for both restaurants and meals. This will explain how we can use those user inputs to recommend a restaurant and the same can be applied to a meal. User recommends / not recommends, rating on a restaurant and finally reviews on a restaurant are the user inputs. In this system provides facility to rate a restaurant scale one to five. Based on the user experience can rate the restaurant. Collect the information about the restaurant from all the user ratings and calculate the final rating. Based on the final rating user can decide which restaurant is the best. And, it provides facility to recommend or not recommend on a restaurant. So, collecting that information also user can decide which restaurant is recommended by others. This system also provides facility to add review on a restaurant. Based on the reviews, user can express their feelings about the restaurant. So other users can read the reviews and conclude which restaurants the best. User recommends / not recommends, and user rating are called direct inputs which can give direct impression about the restaurant. But user reviews are not direct inputs. It has both good and bad impression and user should carefully choose which restaurant is the best. This system will analyze the reviews and give a rating on the review. By collective all the ratings of the reviews on a restaurant system will give final rating on the restaurant through reviews.

Not only that based on users directs inputs above mentioned system could predict some meal to a user by using collaboration filtering. Collaboration filtering is about user preference-based filtering. This will be explained more in Chapter 02. In order to mining the reviews, the system is used opinion mining techniques which also be explained in detail in Chapter 02.

1.7 Structure of the Thesis

This thesis has been organized as follows. Chapter 1 will give the introduction of the thesis and the reviewer can get some idea about the thesis by reading it. Chapter 2 critically reviews the literature on restaurant recommendation techniques and identifies the research problems. Chapter 3 is about the technologies for restaurant recommendation system. Chapter 4 presents a new approach to restaurant recommendation. Chapter 5 describe about the design of the proposed solution. Chapter 6 describes the implementation details more descriptively. Finally, Chapter 7 and Chapter 8 describe evaluation and conclusion respectively.

1.8 Summary

This chapter gave an overall picture of the entire project presented in this thesis. As such we described the background/motivation, problem definition, hypothesis, objectives, and a brief overview of the solution. Next presents a critical review of restaurant recommendation.

Achievements and Challenges in Recommendation of Restaurants

2.1 Introduction

Chapter 1 gave a comprehensive description of the overall project, structure of the thesis and the solution. In this chapter, it will provide critical reviews of the literature that relates to developments and challenges of the restaurant recommendation.

2.2 Early Development

P. Dwivedi and N. Chheda[1] thinks the use of collaborative filtering[2] and knowledge based approach[3] separately has certain drawbacks. So, they a proposed hybrid restaurant recommendation system by combining advantages of both these techniques and increase the accuracy.

Y. Gao, W. Yu, P. Chao, R. Zhang, A. Zhou, and X. Yang[4] proposed a novel approach, which capture the relationship between ratings and user reviews. Based on the user reviews, user preferences of the restaurant will be affected. So, they collect the hidden aspects from reviews and create a profile for the user against restaurants. Based on this model it will recommend restaurants.

T. N. Trang Tran, M. Atas, A. Felfernig, and M. Stettinger[7] mainly focused on recommending healthy food to the end user based on the users' health problems. This system keeps nutritional consumptions records and control the behavior of the end user in a positive manner. Also, it has recommendation ability recommend group of foods for the group of people.

I. TATLI[8] proposed a model to recommend similar kind of food for the end user based on content-based recommendation. First, it will keep users' preferences and calculate the similarities on food items based on the users' interest and recommend.

M. Phanich, P. Pholkul, and S. Phimoltares[9] proposed recommendation system for the diabetic users. In this system, it will keep the record of the food items which are lead to diabetic and recommend similar nutrition level food to the end user to substitute them.

C. Trattner and D. Elswailer[10] thinks recommending a food is not enough to the end user. It is better to give a summary of the nutrition levels to the end user so that the end user can get much more idea about the food items they have eaten. And, the user can get a better idea to balance what kind of nutrition they should take by looking at the nutrition summary. Also, it provides recommendations to the group of people which gives the idea to eat healthy food.

D. C. MANOLI and D. G. BROWN[11] proposed a mobile-based solution to recommend a restaurant based on collaborative filtering. It uses a group of knowledge to recommend the restaurant for users who are new to the system and users who already in the system which has interesting on the restaurant.

A. Gupta and K. Singh[12] proposed a location-based restaurant recommendation mobile system. That system will study user visiting patterns of a restaurant and suggest restaurants based on machine learning algorithms. It also uses web application the 3G services.

J. Almeida[13] proposed a recommendation system by analyzing users' preferences. It uses content-based algorithms for this recommendation engine. It uses Rocchio's algorithm and variations to recommend restaurants.

2.3 Collaboration Filtering

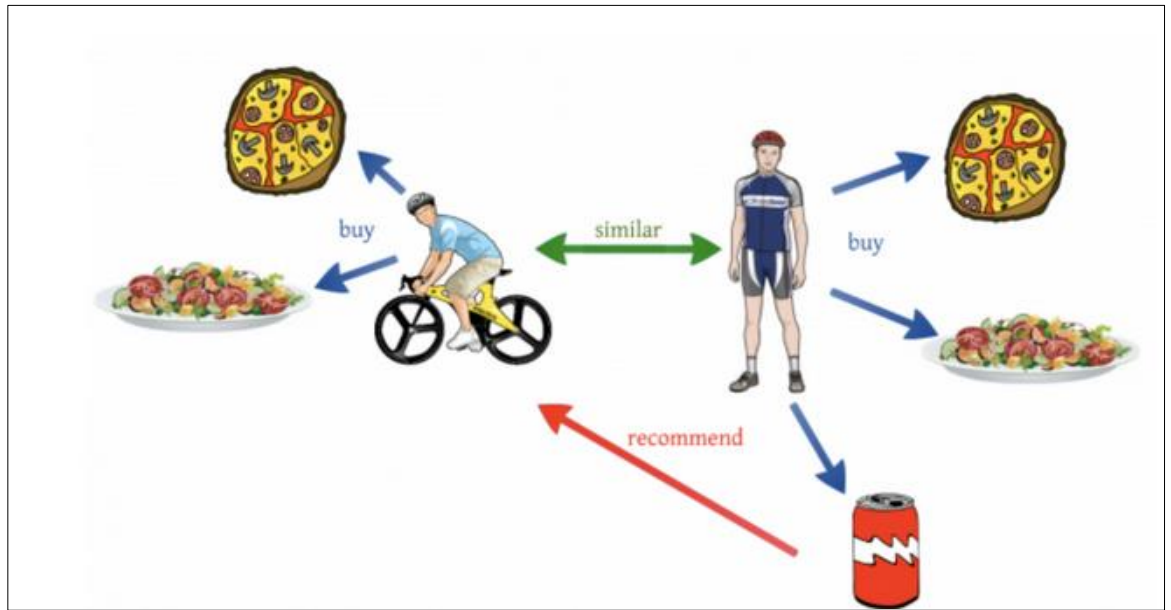


Figure 2.3.1 – Collaboration Filtering

Collaboration filtering [2] is to collect previous users' preferences and do the recommendation to other users based on their preferences. With this approach, it will identify similar kind of users and suggest similar items which they preferred. But for the new user, it won't give a proper recommendation because the system doesn't know the preferences about the new user. So, it will fail to give a recommendation for a new user. Not only that it will fail for the new items also.

2.4 Opinion Mining Techniques

One of natural language processing, called, opinion mining [15] collect user feelings, attitude and appraisals of a review. Then it will classify the reviews into positive, neutral and negative classifications. Next one is comparison opinion [6]. That is do a comparison over another topic or object. So, gathering opinion mining data on a topic will rate the topic or product based on the comments scale 1 to 5.

2.5 Modern Trends in Data Mining

Most of the researches use collaborative filtering, content-based filtering and opinion mining techniques to recommend restaurants. But now a days most the researches use combinations of recommendation techniques for the recommendation. So, it is called hybrid recommendation systems.

2.6 Future Challenges of Data Mining

Few researchers are conducted researches recommending restaurants based on users' reviews on restaurant. That has a big impact on recommending a restaurant to other users. So, analyzing reviews we should use opinion mining technique as future challenge of data mining. And, people may add reviews, over rate purposely to make a restaurant popular among other users. That should be identified by using some data mining techniques. That will be a big challenge in future.

2.7 Problem Definition

Even though significant numbers of researches have been conducted around restaurant recommendation it is not accurate every time. So, need to use some techniques or combining many techniques to analyze the users' inputs and provide necessary restaurant recommendations to the end user is the challenge.

2.8 Summary

This chapter presented a comprehensive literature review on restaurant recommendation and it clearly mentioned about what is the problem definition while presenting the newly trend of recommendation and future challenges. The next chapter will give detail description about the technologies implement proposed solution.

Technologies

3.1 Introduction

This chapter will give detail description about the technologies that used to recommend restaurants. The main programming language is Java and it used to implement the business logic and for user interface implementation uses Angular which is a front-end framework. And, Weka is used analyze user inputs and provide necessary recommendations.

3.2 Java

One of the high-level programming languages owned by Oracle Corporation. It is platform independent and uses object-oriented concepts. It has the concept “write once, run any ware” because it is running Java Virtual Machine and it only needs compiled code. Because of that feature it is not going to depend on the computer architecture.

3.3 Angular

Angular is very popular front-end development framework owned by Google. It has, its own platform to build web application on top of it. It provides very advanced features like dependency injection, declarative templates, two-way binding and tools. Angular application will run on both web and mobile applications.

3.4 REST

REST is stand for Representational State Transfer and it is an architectural pattern which defines a set of constraints and developer should adhere to develop web services. Web services uses REST, called as RESTful web services. Now a days, all kind of developments end with mobile applications. So, developers use REST because it gives JSON response which can easily use for mobile applications.

3.5 Spring

Spring framework is a Java light weight framework that facilitates module driven programming. It provides many modules and developer can easily integrate them on the demand. Spring gives features like dependency injection infrastructure to develop Java application.

3.6 Hibernate

Hibernate is a Java framework and object relational mapping tool. Developer can easily map with Java POJOs with the database tables. It will give easy environment to the developer so that developer should do less code and rest of the things will be handled by Hibernate framework.

3.7 MySQL

MySQL is an open source database and it will give ease of use, reliability and performance to the developer for the implementation. It also supports triggers, stored procedures to give proper solution from developer side. Now a days MySQL becomes a famous database for the all kind of web applications, mobile applications, cloud systems and embedded systems.

3.8 Weka

WEKA is stand for Waikato Environment for Knowledge Analysis and also it is popular machine learning software. It has been written in Java and developer can easily integrate to the application. It was developed by University of Waikato in New Zealand. It is open source and has many patterns and all kind of data mining algorithms.

3.9 Summary

In this chapter presented about the technology used to recommend restaurants. Mainly it presented about Java, Angular, REST and Weka. Next chapter will present about the approach that has been taken to resolve recommending restaurants.

Novel Approach to Restaurant Recommendation

4.1 Introduction

Here we present our new approach to restaurant recommendation and it will be a novel solution to our research problem. The approach is described under the headings of hypothesis, users of the system, input to the system, output of the system, process, convert input to the output and features of the system.

4.2 Hypothesis

We hypothesis that restaurant meal recommendation can be done with collaboration filtering and opinion mining techniques.

4.3 Recommendation Engine

This is the heart of the system. This will suggest menu items based on previous users' recommendations. To learn those patterns this engine uses one of data mining algorithm called Apriori algorithm. By using Apriori algorithm it will generate some association rules based on previously generated ARFF file buying patterns. And, it uses opinion mining techniques like sentimental analysis to classify the reviews.

4.4 Apriori Algorithm for Associations

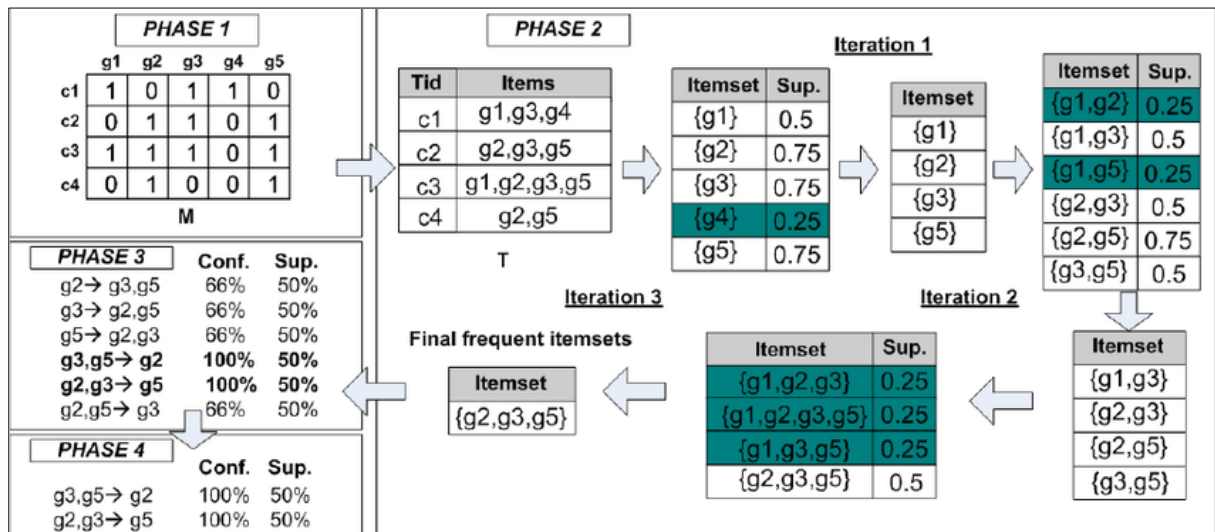


Figure 4.4 – Apriori Algorithm

Let's consider phase 1 in the diagram, rows are the users of the system and columns are restaurant menu items. So, arrange the table so that putting 1 (or "t") if the user has recommended the restaurant menu item. If it is not put 0 (or "?"). Then create the phase 2 table as in the diagram After creating that table apply Apriori algorithm to find associations.

4.4.1 Create Relationship Table Food Item Against User

As explained above create following table food item against user based on users' preferences

	Rice	Noodles	Chicken	Fish	Vegetable
John	1	0	1	1	0
David	0	1	1	0	1
Miller	1	1	1	0	1
Steve	0	1	0	0	1

4.4.2 Calculate Support for Food Item

Calculate support by using following formula,

$$\text{Support} = \frac{\text{Num of food item occurrence}}{\text{Num of transactions}}$$

Food Item	Support
Rice (2 occurrences)	0.5
Noodles (3 occurrences)	0.75
Chicken (3 occurrences)	0.75
Fish (1 occurrence)	0.25
Vegetable (3 occurrences)	0.75

4.4.3 Filter High Support Food Items

Filter high support food items where the support is more than 0.5 and calculate the support for combinations of food items as follows,

Food Item	Support
Noodles, Chicken (2 Occurrences)	0.5
Noodles, Vegetable (3 Occurrences)	0.75
Chicken, Vegetable (2 Occurrences)	0.5

4.4.4 Filter High Support Food Item and Make Association

According to the above table Noodles and Vegetable had the high support. Then we can consider following associations,

Noodles → Vegetables

Vegetables → Noodles

If a user sees Noodles the system will suggest Vegetables. And also, if a user sees vegetable the system will suggest Noodles.

4.5 Sentimental Analysis

Sentimental analysis is one of natural language processing also known as opinion mining technique. This analysis we can perform on restaurant menu item's comments that have been added by the end user. Mining the comments and categorized them into positive, negative and neutral buckets. Based on Bayes algorithm it will calculate the rating on the comment and people can get better understanding about the comment.

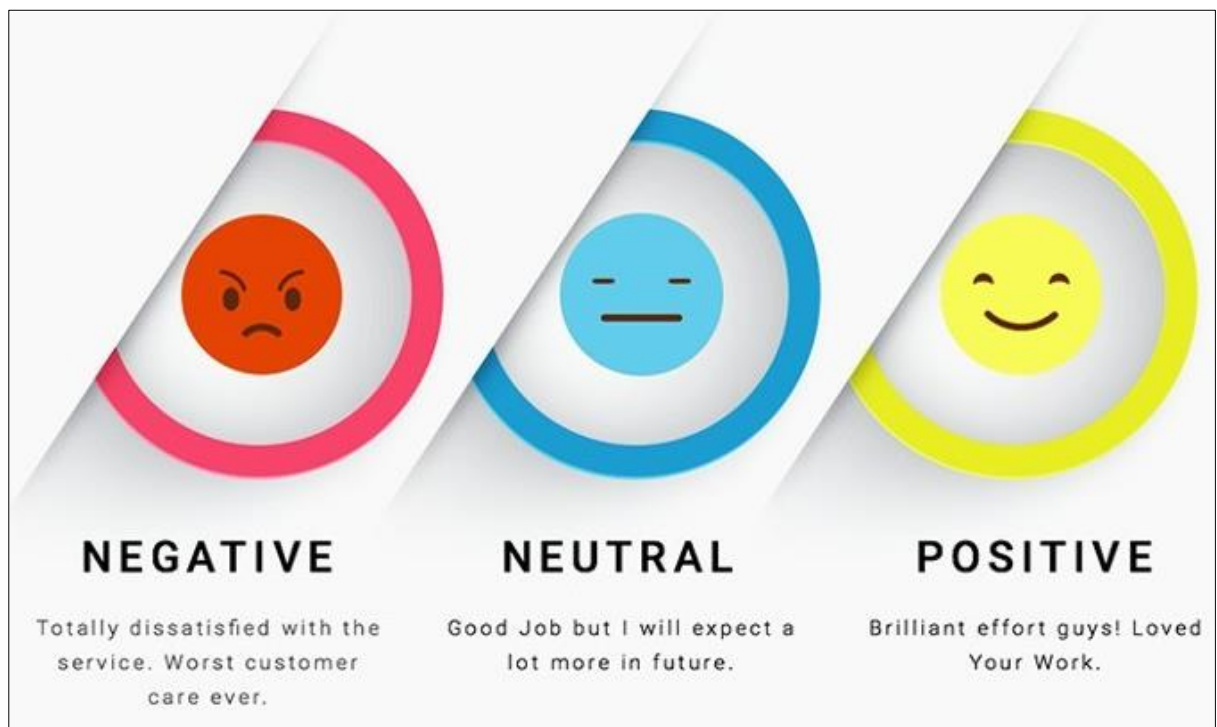


Figure 4.5 – Sentimental Analysis

Let's consider following comments on a Food item and will check the algorithm weather the comment is positive or negative.

"I recommend this meal to everyone"

For this comment apply some data set which is called bag of words. That has been classifying as Positive, Negative and Neutral.

4.5.1 Apply Bag of Words

I	Neutral
Recommend	Positive
This	Neutral
Meal	Neutral
To	Neutral
Everyone	Neutral

After applying bag of words, we can clearly say that this comment is positive. But sometimes applying a bag of words not accurate. So, another technique called RNN (Recursive Neural Network) can apply to increase the accuracy.

4.5.2 Recursive Neural Network (RNN)

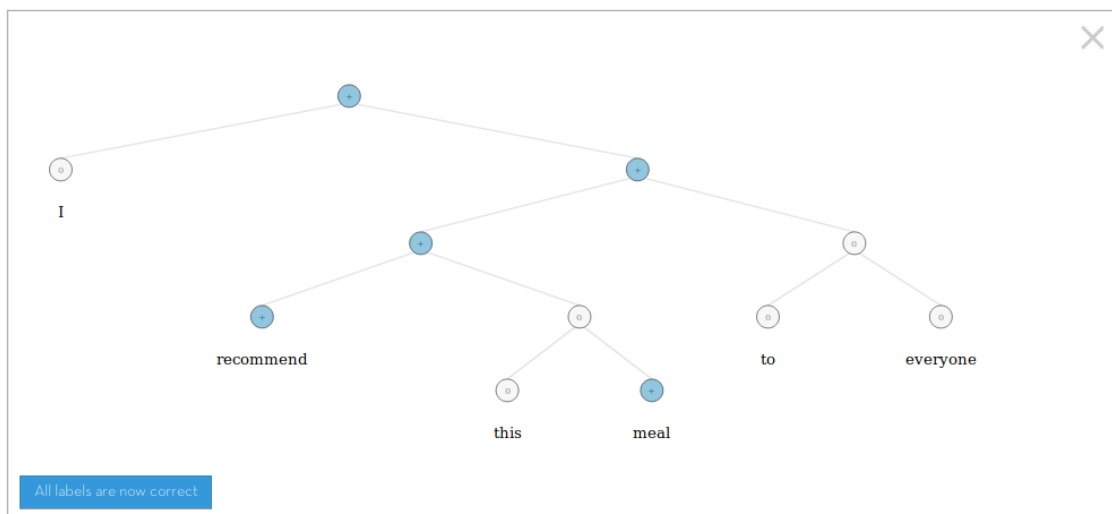


Figure 4.5.2 – Recursive Neural Network (RNN)

Here we take group of words and apply some trained data to determine that is positive, negative or neural. Here, we are going to use Stanford-nlp library and it is keeping the Stanford Sentiment Treebank which is already trained with movie reviews.

This Meal	Neutral
To Everyone	Neutral
Recommend This Meal	Positive
Recommend This Meal To Everyone	Positive
I Recommend This Meal To Everyone	Positive

After applying RNN to the comment we can get 85% accuracy review classifications. Stanford-nlp library classify a comment into five classifications. Very positive (5), Positive (4), Neutral (3), Negative (2) and Very negative (1). Based on the classification give a rating to the comment for final rating calculations.

4.6 Users

Many users are benefited by the restaurant recommendation directly or indirectly. Customer is the main user of this system. Customer can find out what is the best restaurant around him and based on customers' preferences. If a customer wants to share their experience as a review and it will be easy to decide which restaurant is the best for other customers. Based on the good ratings, recommendations and reviews many customers will be approached the restaurant and it will increase their revenue day by day. So not recommendations, bad ratings and bad reviews also give indication to the owner or manager about the restaurant and they can make some decision to keep the restaurant better way so that customers will happy and they can increase their revenue. Finally, the admin is the user who can manage this application. Not only them people who like to study data mining also can use this system for learning purpose.

4.7 Input

The major three inputs are recommendation (which is a direct input for the restaurant from customer who will recommend this restaurant or not recommend to the other customers), ratings and reviews about the restaurant that for restaurant recommendation purpose. And, user can input restaurant profile, restaurant menu items, location of the restaurant, user profile.

4.8 Output

The main output is the recommended meals with restaurants. In addition to that user can refer restaurant profile, user profile, location of the restaurant and reviews of restaurants. User can search restaurant meals by location or restaurant meals by meal name or both location and meal name.

4.9 Process

Restaurant recommendation process can be divided into several sections. Following are some major process.

4.9.1 Login Process

The system executes a user authentication process before allowing a user to use the facilities of the system. This process is done by verifying of user name and the password. After the authentication process, it will allow user to access authorized pages and add necessary inputs to the system.

4.9.2 Recommendation Process

As in the inputs section one of main inputs are Recommend and Not Recommend features on a restaurant meal. Once a user has logged into the system user can do the recommendation and then calculate all the recommendations and not recommendations separately on a restaurant meal and take recommendation percentage.

4.9.3 Rating Process

Next major input is rating on a restaurant meal. Based on each rating calculate average rating on a restaurant meal.

4.9.4 Review Process

The final major input is reviews on a restaurant meal. Based on opinion mining techniques rate the review scale 1 to 5. And calculate average of ratings of all the reviews on a restaurant meal.

4.9.5 Final Rate Calculation Process

After calculating recommendation percentage, average rating and average review rating calculate average rating on a restaurant meal by using those inputs and it will be the final rating on the restaurant meal. Based on that final rating it will rank the restaurant meal.

4.10 Features

Connection with the input, output, users and process, the over features of the system include the following characteristics.

- Online solutions
- Multiple input forms
- User friendly
- Responsiveness

4.11 Data Collection

In order to suggest restaurant meals, the recommendation engine should be trained. So, got some data from Yelp (<https://www.yelp.com/dataset/download>) which is a restaurant recommendation system. Following are some information about the data set.

4.11.1 Business JSON

```
{
// string, 22 character unique string business id
"business_id": "tnhfDv5I18EaGSXZGiuQGg",

// string, the business's name
"name": "Garaje",

// string, the neighborhood's name
"neighborhood": "SoMa",

// string, the full address of the business
"address": "475 3rd St",

// string, the city
"city": "San Francisco",

// string, 2 character state code, if applicable
"state": "CA",

// string, the postal code
"postal code": "94107",

// float, latitude
"latitude": 37.7817529521,

// float, longitude
"longitude": -122.39612197,

// float, star rating, rounded to half-stars
"stars": 4.5,

// interger, number of reviews
"review_count": 1198,

// integer, 0 or 1 for closed or open, respectively
"is_open": 1,
```

```
// object, business attributes to values. note: some attribute values
might be objects
```

```
"attributes": {
  "RestaurantsTakeOut": true,
  "BusinessParking": {
    "garage": false,
    "street": true,
    "validated": false,
    "lot": false,
    "valet": false
  },
},
```

```
// an array of strings of business categories
```

```
"categories": [
  "Mexican",
  "Burgers",
  "Gastropubs"
],
```

```
// an object of key day to value hours, hours are using a 24hr clock
```

```
"hours": {
  "Monday": "10:00-21:00",
  "Tuesday": "10:00-21:00",
  "Friday": "10:00-21:00",
  "Wednesday": "10:00-21:00",
  "Thursday": "10:00-21:00",
  "Sunday": "11:00-18:00",
  "Saturday": "10:00-21:00"
}
}
```

4.11.2 Review JSON

```
{
  // string, 22 character unique review id
  "review_id": "zdSx_SD6obEhz9VrW9uAWA",

  // string, 22 character unique user id, maps to the user in
  user.json
  "user_id": "Ha3iJu77CxlRfm-vQRs_8g",

  // string, 22 character business id, maps to business in
  business.json
  "business_id": "tnhfDv5I18EaGSXZGiuQGg",

  // integer, star rating
  "stars": 4,

  // string, date formatted YYYY-MM-DD
  "date": "2016-03-09",

  // string, the review itself
  "text": "Great place to hang out after work: the prices are decent,
and the ambience is fun. It's a bit loud, but very lively.
The staff is friendly, and the food is good. They have a good
selection of drinks.",

  // integer, number of useful votes received
  "useful": 0,

  // integer, number of funny votes received
  "funny": 0,

  // integer, number of cool votes received
  "cool": 0
}
```

4.11.3 User JSON

```
{
  // string, 22 character unique user id, maps to the user in
  user.json
  "user_id": "Ha3iJu77CxlRfm-vQRs_8g",

  // string, the user's first name
  "name": "Sebastien",

  // integer, the number of reviews they've written
  "review_count": 56,

  // string, when the user joined Yelp, formatted like YYYY-MM-DD
  "yelping_since": "2011-01-01",

  // integer, number of useful votes sent by the user
  "useful": 21,

  // float, average rating of all reviews
  "average_stars": 4.31
}
```

4.12 Storing Data

Read above data set by using Java file reading techniques convert JSON data into Java class by using Java JSON to object conversion techniques. And use that class and use Hibernate ORM to store them into MySQL database.

4.13 Summary

This chapter presented about the approach of recommending restaurant based on three major inputs recommend / not recommend, rate and user reviews. Next chapter will present about the design.

Design of Restaurant Recommendation

5.1 Introduction

The previous approach chapter gave full overview about the entire solution. This chapter describes the design of the solution for the process presented in the approach. Here it supports for both desktop and mobile devices with the page responsiveness.

5.2 Top Level Design

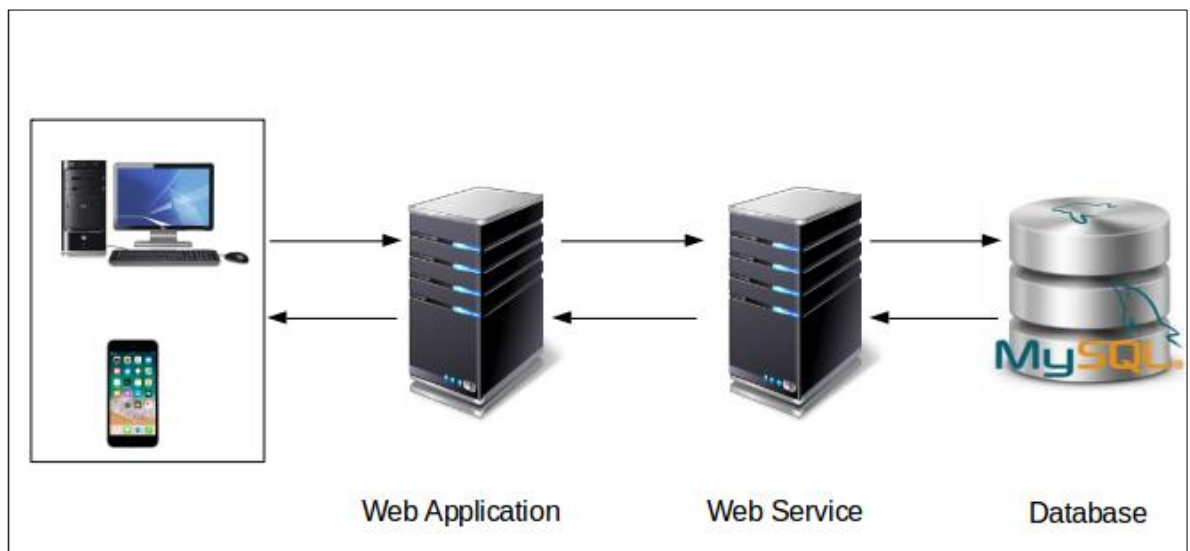


Figure 5.2.1 – Top Level Design

5.3 Web Application

Web application basically provides user interfaces such as HTML form enter restaurant details, user details, give recommend/ not recommend, give ratings, add reviews on restaurant meal and so on. For the web application development, here we use Angular which provides rich Internet applications along with the bootstrap CSS framework for user attraction and page responsiveness. Web application directly communicates with web service to submit data, retrieve data and data manipulation.

5.4 Web Service

Web service provides the business logic like adding restaurant information, user information, calculating restaurant meal ratings and so on. Here we use REST web service along with Spring and Hibernate Java frameworks. And, here we use Weka Java API to train the data set and using that predict the recommendation along with Java algorithms. Here handles the HTTP request that coming from the web application and based on the request do necessary business logic execution with the data that receiving from the database and send manipulated data to the web application as the response.

5.5 Database

Here we use MySQL as the database and it keeps restaurant details, user details, recommend / not recommend details, ratings, user reviews and so on. MySQL is light weight open source database and it is easy to use and easy to understand other users or developers. Web service is the one accessing database to perform CRUD operations.

5.6 Web Container

Here we use Apache Tomcat as the servlet container and it is open source. It is light weight and embedded in the application with spring boot.

5.7 Java Tool

We can't directly use the data set as it is. So, need to do some pre-processing on the data set. So, we can use this application tools to do pre-processing. And also for the restaurant meals recommendation it is using Weka tool to generate ARFF (Attribute Relation File Format) for Apriori algorithm to give association rules. Based on the association rules system can suggest a restaurant meal when user buy some other restaurant meal.

5.8 Detailed Design

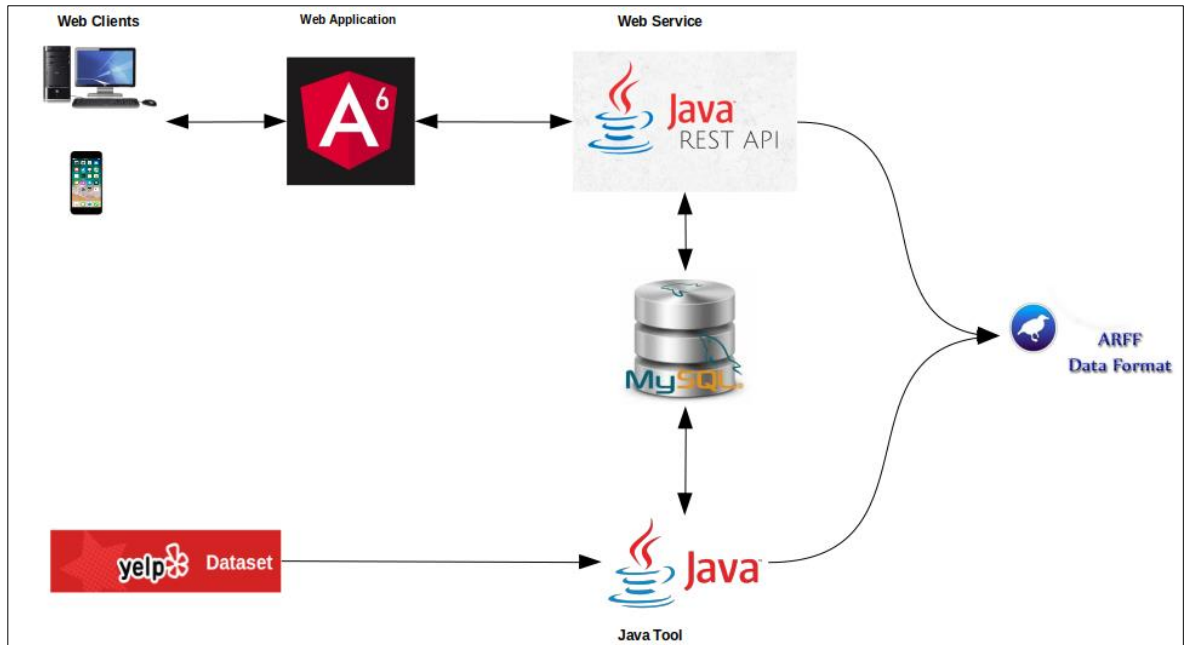


Figure 5.8.1 – Detailed Design

5.9 Main Modules

Here we have several major modules as follows,

- Login Module
- User Module
- Restaurant Module
- Dashboard
- Search Module
- Recommendation Engine

5.9.1 Login Module

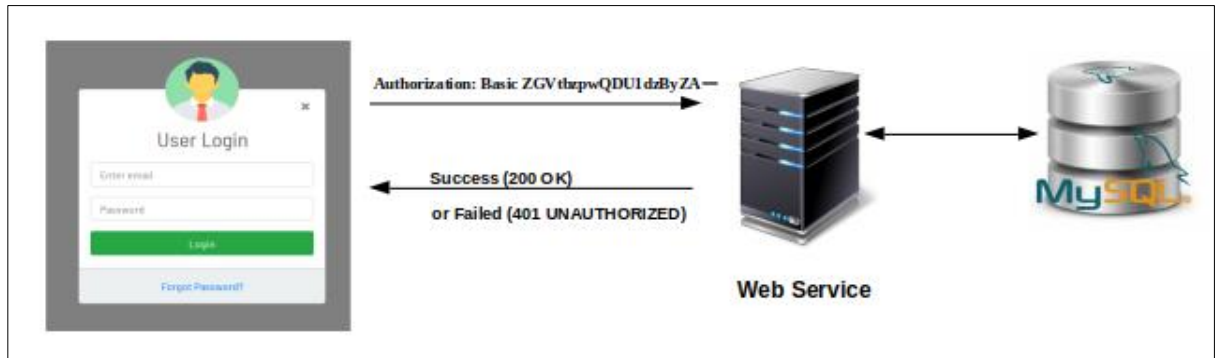


Figure 5.9.1.1 – Login Module

Login module is one of main module of this system. User can log in to the system by giving user name and password. It will use basic authentication to send user credentials from web application to the web service encoding it with base64 algorithm in HTTP header. First take the user name and password combine them with colon (username:password). Then apply the base64 algorithm to that username:password string and take the encoded string. Finally send the request to the web service in following format for the login process.

Authorization: Basic ZGVtbzpwQDU1dzByZA==

After the sending basic authentication token service side break the token again into user name and password and validate it with database user name and password. If it is valid send 200 OK response with the basic authentication token. If it is not valid send 401 UNAUTHORIZED response to the web application.

5.9.2 User Module

Main web form of the user module is user registration form. User should enter necessary details and need to submit it for the registration. This is the web form user should enter their user name and password to log into the system after the registration is done.

5.9.3 Restaurant Module

In this module user will be able to create restaurant profile by entering name, address, telephone, location and so on. After entering those information users should submit the form to register a restaurant. And, this module provides facility to create restaurant menu categories and menu items.

5.9.4 Dashboard

When user log into the system they will redirect to the dashboard. There user can see recommended restaurant meals and best restaurant meals so that user can get better idea where they should go based on the results.

5.9.5 Search Module

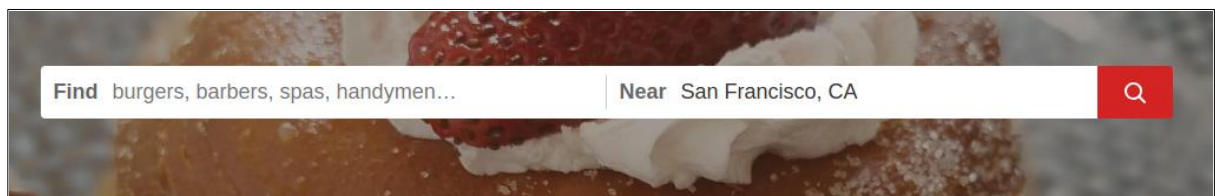


Figure 5.9.5.1 – Search Module

In this system user will be given search facilities so that user can search recommended restaurant meals by selecting restaurant location or restaurant meal name. If the user selects one of restaurant location the system will provide the entire available restaurants around that location and suggest restaurant meals based on recommendations. And, user can perform a search selecting restaurant menu item name and the system will provide the entire available restaurant with the menu item based on the recommendation.

5.9.6 Recommendation Engine

This is the heart of the system. This will suggest menu items based on previous users' recommendations. To learn those patterns this engine uses one of data mining algorithm called Apriori algorithm. By using Apriori algorithm it will generate some association rules based on previously generated ARFF file buying patterns as described in the Chapter 04. And, it uses opinion mining technique like sentimental analysis together with associations.

5.10 Summary

In this chapter, we discussed about design for the solution in detail. Here we use client server application which is hosted in Tomcat web container. Next chapter will discuss about the implementation.

Implementation

6.1 Introduction

As in the previous chapter design, this chapter will be discussed about the implementation of each modules to reach the solution to the problem.

6.2 Overall Solution

This has been implemented as a web application with the responsiveness. So, this application can be viewed from smart phones also. This is primarily client-server architecture with extension for Google API.

6.3 Collecting, Reading and Storing Data

First downloaded the data set from Yelp and read them in a Java program. Since it is a JSON formatted file here used spring with Jackson which is a JSON to Java object convert library along with Java to do the program. Following is the business.json sample that contains one of the restaurant information like name, place, lat, long .etc.

```
{"business_id": "l09JfMeQ6ynYs5MCJtrcmQ", "name": "Alize Catering", "neighborhood": "Yonge and Eglinton", "address": "2459 Yonge St", "city": "Toronto", "state": "ON", "postal_code": "M4P 2H6", "latitude": 43.7113993, "longitude": -79.3993388, "stars": 3.0, "review_count": 12, "is_open": 0, "attributes": {"GoodForMeal": {"dessert": false, "latenight": false, "lunch": false, "dinner": true, "breakfast": false, "brunch": false}, "Alcohol": "full_bar", "Caters": false, "HasTV": false, "RestaurantsGoodForGroups": false, "NoiseLevel": "average", "WiFi": "no", "RestaurantsAttire": "casual", "RestaurantsReservations": true, "OutdoorSeating": true, "BusinessAcceptsCreditCards": true, "RestaurantsPriceRange2": 3, "RestaurantsTableService": true, "RestaurantsDelivery": false, "Ambience": {"romantic": true, "intimate": false, "classy": false, "hipster": false, "touristy": false, "trendy": false, "upscale": false, "casual": false}, "RestaurantsTakeOut": true, "GoodForKids": false, "WheelchairAccessible": false, "BusinessParking": {"garage": false, "street": true, "validated": false, "lot": false, "valet": false}}, "categories": ["Italian", "French", "Restaurants"], "hours": {"Monday": "9:00-22:00", "Tuesday": "9:00-22:00", "Friday": "9:00-22:00", "Wednesday": "9:00-22:00", "Thursday": "9:00-22:00", "Sunday": "9:00-22:00", "Saturday": "9:00-22:00"}}
```

Figure 6.3.1 – Business JSON

Following is the Java POJO class to map above restaurant information indicated in JSON format in order to store it into the database;

```
@Entity
@Table(name = "Restaurant")
@Getter
@Setter
@JsonIgnoreProperties(ignoreUnknown = true)
public class Restaurant {
    @Id
    @Column
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Transient
    @Column(name = "business_id")
    @JsonProperty("business_id")
    private String businessId;
    @Column
    @JsonProperty("name")
    private String name;
    @Column
    private String description;
    @Column
    @JsonProperty("address")
    private String address;
    @Column
    @JsonProperty("city")
    private String city;
    @Transient
    @Column
    @JsonProperty("state")
    private String state;
    @Column
    @JsonProperty("latitude")
    private Float lat;
    @Column
    @JsonProperty("longitude")
    private Float lng;
    @Column
    @JsonProperty("stars")
    private Float rate;
    @Transient
    @JsonProperty("categories")
    private List<String> categories;
}
```

By using above Java POJO class we can map the business.json data and it is ready to save into the database. Following is the code snippet to read map with POJO and save business data into the database.


```

private void saveRestaurants() {
try (BufferedReader br = new BufferedReader(new FileReader(BUSINESS_JSON)) {
String line;
ObjectMapper objectMapper = new ObjectMapper();
restaurantIds = new ArrayList<>();
int count = 0;
while ((line = br.readLine()) != null) {
Restaurant restaurant = objectMapper.readValue(line, Restaurant.class);
if (restaurant.getCategories().contains("Restaurants")) {
restaurant.setDescription(restaurant.getName());
System.out.println(restaurant);
restaurantDb.save(restaurant);
restaurantIds.add(restaurant.getId());
count++;
if (count >= MAX_RESTAURANT_COUNT) {
break;
}
}
}
} catch (IOException e) {
e.printStackTrace();
}
}
}
System.out.println("Restaurant count is : " + count);
} catch (IOException e) {
e.printStackTrace();
}
}
}

```

Following is the database Restaurant table to store restaurant details that was read from Yelp data set.

```
mysql> desc Restaurant;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
address	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
description	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
lat	float	YES		NULL	
lng	float	YES		NULL	
name	varchar(255)	YES		NULL	
rate	float	YES		NULL	
opening_hours	varchar(255)	YES		NULL	
phone	varchar(255)	YES		NULL	
price_range	varchar(255)	YES		NULL	
website	varchar(255)	YES		NULL	

13 rows in set (0.00 sec)

Figure 6.3.2 – Restaurant Table

Like we did for restaurant details we can apply similar coding for the rest of the data like review and users. Since Yelp is not providing menu categories and menu item information so had to generate them dynamically. And, people can recommend menu item and can give rating on a menu item. That data set also generated as follows.

Program for generating menu item ratings.

```
private void saveMenuItemRates() {
    List<RestaurantMenuItem> restaurantMenuItems = restaurantMenuItemDb.findAll();
    for (RestaurantMenuItem restaurantMenuItem : restaurantMenuItems) {
        int reviewCount = getRandomNumber(25);
        List<Date> reviewDates = getSortedRandomDates(reviewCount + 3);
        for (int i = 0; i <= reviewCount; i++) {
            double rateCount = getRandomNumber(50);
            double rate = rateCount / 10;
            int userId = getRandomNumber(MAX_USER_COUNT);
            RestaurantMenuItemRate restaurantMenuItemRate = new
                RestaurantMenuItemRate();
            restaurantMenuItemRate.setDate(reviewDates.get(i));
            restaurantMenuItemRate.setMenuItemId(restaurantMenuItem.getId());
            restaurantMenuItemRate.setRate(rate);
            restaurantMenuItemRate.setUserId(userId);
            restaurantMenuItemRateDb.save(restaurantMenuItemRate);
            System.out.println(restaurantMenuItemRate);
        }
    }
}
```

Program for generating menu item recommends.

```
private void saveMenuItemRecommends() {
    List<RestaurantMenuItem> restaurantMenuItems = restaurantMenuItemDb.findAll();
    for (RestaurantMenuItem restaurantMenuItem : restaurantMenuItems) {
        int reviewCount = getRandomNumber(50);
        List<Date> reviewDates = getSortedRandomDates(reviewCount + 3);
        for (int i = 0; i <= reviewCount; i++) {
            int flag = getRandomNumber(2);
            int userId = getRandomNumber(MAX_USER_COUNT);
            RestaurantMenuItemRecommend restaurantMenuItemRecommend = new
                RestaurantMenuItemRecommend();
            restaurantMenuItemRecommend.setDate(reviewDates.get(i));
            restaurantMenuItemRecommend.setMenuItemId(restaurantMenuItem.getId());
            restaurantMenuItemRecommend.setUserId(userId);
            restaurantMenuItemRecommend.setRecommend(flag == 1);
            restaurantMenuItemRecommendDb.save(restaurantMenuItemRecommend);
            System.out.println(restaurantMenuItemRecommend);
        }
    }
}
```

More details about the coding and about the database please refer Appendix B

6.4 Sentimental Analysis

For the sentimental analysis we are using Stanford CoreNLP[16] Java library and it can be used to do the analysis on piece of text. This library can easily use to analyze on the text with few coding. It provides many tooling supports like sentiment analysis, part-of-speech (POS) tagger, bootstrapped pattern learning, the named entity recognizer (NER), the open information extraction tools, the parser, and co-reference resolution system. Moreover, an annotator pipeline can include additional custom or third-party annotators. CoreNLP's analyses provide the foundational building blocks for higher-level and domain-specific text understanding applications.

Let's consider following three sentences,

- I liked this meal.
- It was a fantastic experience in this restaurant.
- It gives lot of troubles.

After applying sentimental analysis on those sentences will generate results as follows.

- `<sentence id="1" sentimentValue="3" sentiment="Positive">`
- `<sentence id="2" sentimentValue="4" sentiment="Very Positive">`
- `<sentence id="3" sentimentValue="1" sentiment="Negative">`

As you can see it generate some sentiment value for the Negative, Positive and Neutral sentences as below,

- 0 – Very Negative
- 1 – Negative
- 2 – Neutral
- 3 – Positive
- 4 – Very Positive

Since we are giving rating from 1 to 5 just give the rating as `sentimentValue + 1`. Then it will indicate the sentimental analysis rating as 1 to 5. So, use this library to

generate ratings on reviews for menu items or restaurants. After that ratings could be as below.

- 1 – Very Negative
- 2 – Negative
- 3 – Neutral
- 4 – Positive
- 5 – Very Positive

Following is the Java code snippet used to generate sentiment analysis rating for all restaurant reviews and menu item reviews.

```
public static float getSentimentRate(String line) {
    Properties props = new Properties();
    props.setProperty("annotators", "tokenize, ssplit, parse, sentiment");
    StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
    float mainSentiment = 0;
    if (line != null && line.length() > 0) {
        int longest = 0;
        Annotation annotation = pipeline.process(line);
        for (CoreMap sentence :
            annotation.get(CoreAnnotations.SentencesAnnotation.class)) {
            Tree tree =
                sentence.get(SentimentCoreAnnotations.SentimentAnnotatedTree.class);
            int sentiment = RNNCoreAnnotations.getPredictedClass(tree);
            String partText = sentence.toString();
            if (partText.length() > longest) {
                mainSentiment = sentiment;
                longest = partText.length();
            }
        }
    }
    return mainSentiment + 1;
}
```

6.5 Apriori Algorithm

Apriori algorithm [17] is frequent association mining algorithm and it will useful to generate associations to recommend other menu items to the end user.

6.5.1 Definition

Association rule mining is defined as [18]:

Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of n binary attributes called items.

Let $D = \{t_1, t_2, \dots, t_m\}$ be a set of m transactions.

Each transaction in D has a unique transaction ID and contains a subset of the items in I . A rule is defined as an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The sets of items X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively.

6.5.2 Prepare Data Set to Apply Apriori Algorithm

In this system database, we keep user recommended data on menu items. In order to make associations we just create following table from the database. In this table for columns are menu items' ids, rows are users' ids. Actually, this is a CSV formatted data set which will be the input for the Weka tool.

Let's consider following scenarios,

- If the user has recommended a menu item, we marked it as "t"
- If the user has not recommended a menu item we marked it as "?"

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	t	?	?	?	t	?	?	?	?	?	t	?	?	?	?	t	?	?	?	?	t	t	?	?	?
2	?	?	?	?	?	?	?	t	?	?	?	?	t	?	t	?	?	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?	t	?	?	?	t	?	t	?	?	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?	?	t	?	?	t	?	?	?	?	t	t	?	?	?	?	?	?
5	?	?	t	t	?	?	?	t	?	?	?	?	?	?	t	?	t	t	t	t	?	t	t	?	t
6	?	?	t	?	?	?	?	?	t	?	?	?	t	?	?	?	t	?	t	?	?	?	?	t	t
7	?	?	t	?	t	?	?	?	?	?	t	?	?	?	t	?	?	t	?	?	t	?	t	?	?
8	t	?	?	?	?	t	t	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
9	?	?	?	t	?	?	?	t	?	?	t	?	?	?	?	?	?	?	t	?	?	?	?	t	?
10	?	?	t	?	?	?	t	?	?	t	t	?	?	?	?	?	t	?	t	?	?	?	?	t	?
11	?	?	?	?	t	?	?	?	?	?	?	?	?	?	?	?	t	?	?	?	?	t	t	?	?
12	?	?	?	?	?	?	t	t	?	?	?	t	t	?	?	?	t	?	?	t	?	?	?	?	?
13	?	?	?	t	t	?	?	t	t	t	?	?	?	t	?	?	?	t	t	?	?	?	?	?	?
14	?	?	?	t	t	?	?	t	?	?	?	?	?	?	?	?	?	t	t	?	?	?	?	t	?
15	?	?	?	?	?	?	?	?	t	?	?	?	?	?	t	?	t	t	t	?	?	?	?	t	?
16	?	?	t	?	?	t	t	?	?	?	?	?	t	?	?	t	?	?	?	?	?	t	t	t	?
17	?	?	?	?	t	?	t	t	?	?	t	?	?	?	t	?	?	?	?	?	?	?	t	?	?
18	?	?	?	?	t	t	?	?	?	?	?	?	t	?	?	?	?	?	?	?	?	?	?	t	?
19	?	?	?	t	?	?	?	?	?	t	t	?	t	?	?	?	?	?	t	?	?	?	?	t	?
20	?	?	t	t	?	t	?	?	t	t	?	?	t	?	?	?	?	t	?	?	?	t	?	?	?

Figure 6.5.2.1 – Associations As CSV

6.5.3 Generate ARFF File for Association Rules

Open the weka tool and open above generated CSV file into ARFF viewer as follows. Then weka will generate ARFF file which has the association information which can be used to predict menu item to the end user through the program.

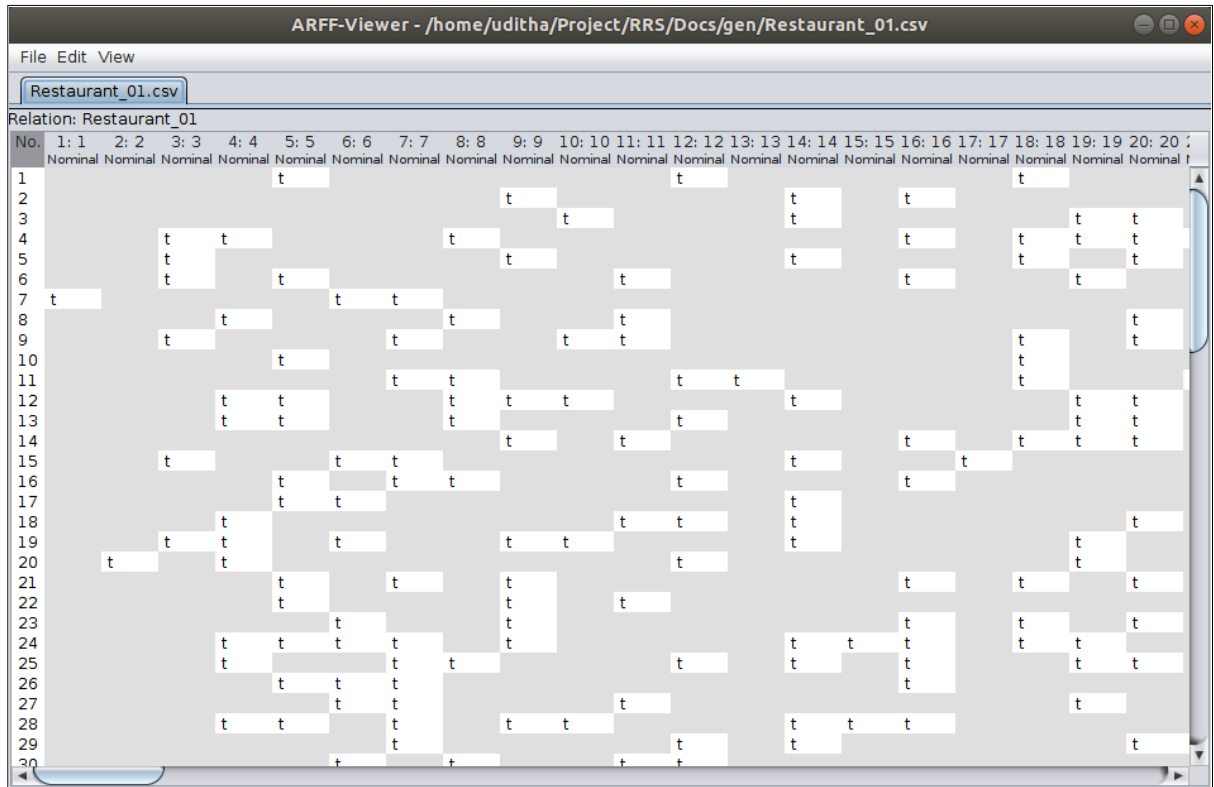


Figure 6.5.3.1 – Weka Tool for Associations

6.5.4 ARFF File

Attribute-Relation File Format or ARFF file is an input to the Apriori algorithm. It is a text file and contains list of attributes and attribute values for a relation of two items. ARFF files invented by Waikato University to develop machine learning software. Following is the ARFF file.

```

1 @relation Restaurant_01
2
3 @attribute 1 {t}
4 @attribute 2 {t}
5 @attribute 3 {t}
6 @attribute 4 {t}
7 @attribute 5 {t}
8 @attribute 6 {t}
9 @attribute 7 {t}
10 @attribute 8 {t}
11 @attribute 9 {t}
12 @attribute 10 {t}
13
14 @data
15 ?,?,?,?,t,?,?,?,?
16 ?,?,?,?,?,?,?,t,?
17 ?,?,?,?,?,?,?,?,t
18 ?,?,t,t,?,?,?,t,?
19 ?,?,t,?,?,?,?,t,?
20 ?,?,t,?,t,?,?,?,?
21 t,?,?,?,t,t,?,?,?
22 ?,?,?,t,?,?,?,t,?
23 ?,?,t,?,?,?,t,?,t
24 ?,?,?,?,t,?,?,?,?
25

```

Figure 6.5.4.1 – ARFF File

6.5.5 Generate Association Rules Using ARFF

By using above created ARFF file we can generate association rules via a Java program. Following is some output of the association rules.

Here 124 → 44 means, when a user buy menu item id 124 (Let's say Fried Rice) suggest menu item id 44 (Let's say Chicken Devil). Likewise using following association rules system can suggest other menu items accordingly. Please note that these menu items are belongs to a restaurant. So, suggestion will happen among the restaurant menu items.

124,	----->	44		Confidence (0.667)
136,	----->	106		Confidence (0.655)
39,	----->	7		Confidence (0.636)
60,	----->	16		Confidence (0.633)
69,	----->	66		Confidence (0.633)
9,	----->	150		Confidence (0.615)
12,	----->	22		Confidence (0.615)
116,	----->	150		Confidence (0.615)
26,	----->	14		Confidence (0.607)
143,	----->	150		Confidence (0.607)
93,	----->	38		Confidence (0.6)
35,	----->	106		Confidence (0.588)
119,	----->	14		Confidence (0.586)
65,	----->	16		Confidence (0.586)
154,	----->	41		Confidence (0.586)
151,	----->	106		Confidence (0.586)
38,	----->	150		Confidence (0.579)
116,	----->	14		Confidence (0.577)
141,	----->	44		Confidence (0.577)
82,	----->	128		Confidence (0.577)
138,	----->	99		Confidence (0.577)
39,	----->	16		Confidence (0.576)
135,	----->	150		Confidence (0.576)
128,	----->	64		Confidence (0.571)
3,	----->	24		Confidence (0.571)

Figure 6.5.5.1 – Association Rules

Following is the Java program that used to generate association rules by using above mentioned ARFF file.

```
public static void getAssociations(String arffFilePath, int numberOfRules) throws
    Exception {
    AssociationRules associationRules = getAssociationRules(arffFilePath, numberOfRules);
    List<AssociationRule> associationRuleList = associationRules.getRules();
    for (AssociationRule associationRule : associationRuleList) {
    Collection<Item> premiseCollection = associationRule.getPremise();
    List<String> items = new ArrayList<>();
    for (Item item : premiseCollection) {
    Attribute attribute = item.getAttribute();
    String met = attribute.name();
    items.add(met);
    }
    Collection<Item> consequence = associationRule.getConsequence();
    List<String> predictItems = new ArrayList<>();
    for (Item item : consequence) {
    Attribute attribute = item.getAttribute();
    String met = attribute.name();
    predictItems.add(met);
    }
    int premiseSupport = associationRule.getPremiseSupport();
    int totaSupport = associationRule.getTotalSupport();
    double confidenceNonFormatted = (float) totaSupport / premiseSupport;
    DecimalFormat twoDForm = new DecimalFormat("#.###");
    String formattedNumberString = twoDForm.format(confidenceNonFormatted);
    Double formattedConfidence = Double.valueOf(formattedNumberString);
    System.out.println(items + "-->" + predictItems + " | " +
        formattedConfidence);
    }
}
```

6.6 The Ways of Recommending Restaurant Meals

In this system recommendations will give in two ways. First one is recommending a restaurant meal based on previous users' recommendations. But here it considers only the user recommendation. It is a direct input for the recommendation and used Apriori algorithm to do suggestion on restaurant meals. But some users may give a rating (from 1 to 5) and some user can add a review on restaurant meals. So, the second recommendation of the system is calculating user reviews, user ratings and user recommendations suggest best recommended meal to the end user. Please refer following sections for more details.

6.6.1 Recommend Restaurant Meals Based on User Recommendations

Here it is considering only the user recommendations which can be given through the system on restaurant meal. That is a direct input for the user recommendation and for this approach used Apriori algorithm to suggest restaurant meals as explained above sections.

6.6.2 Recommend Best Restaurant Meals

In order to recommend best restaurant, we have to take all the user recommendations, user reviews and user ratings. Based on those three inputs calculate average rating on a restaurant meal. Then suggest best rated or maximum rated restaurant meals to the end user. Following are the formulas to calculate average rating on restaurant meal.

Calculate average user rating on a restaurant meal. Let's called the results as X.

$$X = \frac{\sum_{i=1}^n x_i}{n};$$

Where x is rating on a restaurant meal and n is number of ratings on a restaurant meal

Calculate average user review rating on a restaurant meal. Let's called the results as Y.

$$Y = \frac{\sum_{i=1}^n y_i}{n};$$

Where y is review rating on restaurant meal and n is number of reviews on a restaurant meal

Calculate user recommendation rating on a restaurant meal. Let's called the results as Z.

$$Z = \frac{\text{numberofrecoemendation}}{\text{numberofrecomendations+numberofnotrecomendations}} \times 5;$$

So, let's calculate the average rating on restaurant meal. Let's called the result as R.

$$R = \frac{X + Y + Z}{3}$$

So, based on maximum R suggests restaurant meals to the end user.

6.7 Summary

This chapter provides implementation details of each module of proposed solution. Moreover, it mentioned software, algorithm and resulting models of each module in the design. Next chapter evaluates all the modules implemented in the solution.

Evaluation

7.1 Introduction

Previous chapter described the implementation for the design of the restaurant recommendation system. In this chapter will describe how the software solution can be tested functionality, reliability and efficiency. Testing is one of critical process to check whether the system is working as expected or not. In other words evaluating a system and its components to check it satisfies the given requirements.

7.2 Evaluation of Web Application

Evaluation of the web application such as create restaurant, register user, list recommended restaurants and so on. Done by manually. Here basically what we did is validation of web forms tested alignments and spelling mistakes and grammar.

7.3 Evaluation of Final Rating

We will create two data sets as one data set for specific date range and other data set for another date range as follows,

```
mysql> select avg(rate) from RestaurantMenuItemRate where menu_item_id = 1 and date between "2018-01-01" and "2018-08-31";
+-----+
| avg(rate) |
+-----+
| 2.2363636363636363 |
+-----+
1 row in set (0.02 sec)

mysql> select avg(rate) from RestaurantMenuItemRate where menu_item_id = 1 and date between "2018-09-01" and "2018-12-31";
+-----+
| avg(rate) |
+-----+
|      3.7 |
+-----+
1 row in set (0.02 sec)
```

Figure 7.3 – Restaurant Average Rating on Specific Time Period

According to above queries ratings are,

From 2018-01-01 To 2018-08-31 = 2.24

From 2018-09-01 To 2018-12-31 = 3.7

So, calculation for similarity is as follows,

$$\text{Similarity} = \frac{\min(\text{Before Rate}, \text{After Rate})}{\max(\text{Before Rate}, \text{After Rate})} * 100$$

For the above example similarity = 60.54%

So according to this example accuracy of the rating is 60.54%.

So, let's consider following table 10 restaurant menu items, average ratings and the accuracy,

Menu Item	From 2018-01-01 To 2018-08-31	From 2018-09-01 To 2018-12-31	Accuracy (%)
1	2.24	3.7	60.54
2	2.90	2.66	91.72
3	2.00	1.99	99.5
4	1.74	3.22	54.03
5	2.61	2.63	99.23
6	2.84	4.13	68.76
7	2.54	2.24	88.18
8	1.55	4.7	32.97
9	3.25	3.96	82.07
10	2.27	2.44	93.03
Average Accuracy			77.00

According to the above table we have 77% accuracy to do the similar rating on same menu item.

7.4 Evaluation of Opinion Mining

To evaluate the reviews and its ratings should be done manually. Then it is easy to evaluate reviews looking at the reviews to achieve high accuracy. This can be Positive, Negative and Neutral. Then evaluate the results with system.

Let's consider following sentences and check it has given correct ratings.

- I love this meal so much (Positive review)
- Nice meal (Hard to say positive or negative, so it is neutral review)
- Very bad meal ever (Negative review)

After executing following main program to execute opinion mining algorithm as stated in Appendix section gives following results.

```
public static void main(String[] args) {  
    String review = "I love this meal so much";  
    float rate = SentimentUtil.getSentimentRate(review);  
    System.out.println(review + " : " + rate);  
    review = "Nice meal";  
    rate = SentimentUtil.getSentimentRate(review);  
    System.out.println(review + " : " + rate);  
    review = "Very bad meal ever";  
    rate = SentimentUtil.getSentimentRate(review);  
    System.out.println(review + " : " + rate);  
}
```

Output:

- I love this meal so much (Positive): 4.0
- Nice meal (Neutral): 3.0
- Very bad meal ever (Negative): 2.0

Based on the following reference given by the Stanford sentiment analysis tool we can verify that the output is matched and accurate.

- 1 – Very Negative
- 2 – Negative
- 3 – Neutral
- 4 – Positive
- 5 – Very Positive

7.5 Evaluation of Restaurant Menu Item Final Rating

Let's take one of restaurant menu item. I'm going to take Fried Rice menu item of Messina restaurant. Following is the image of that menu item.

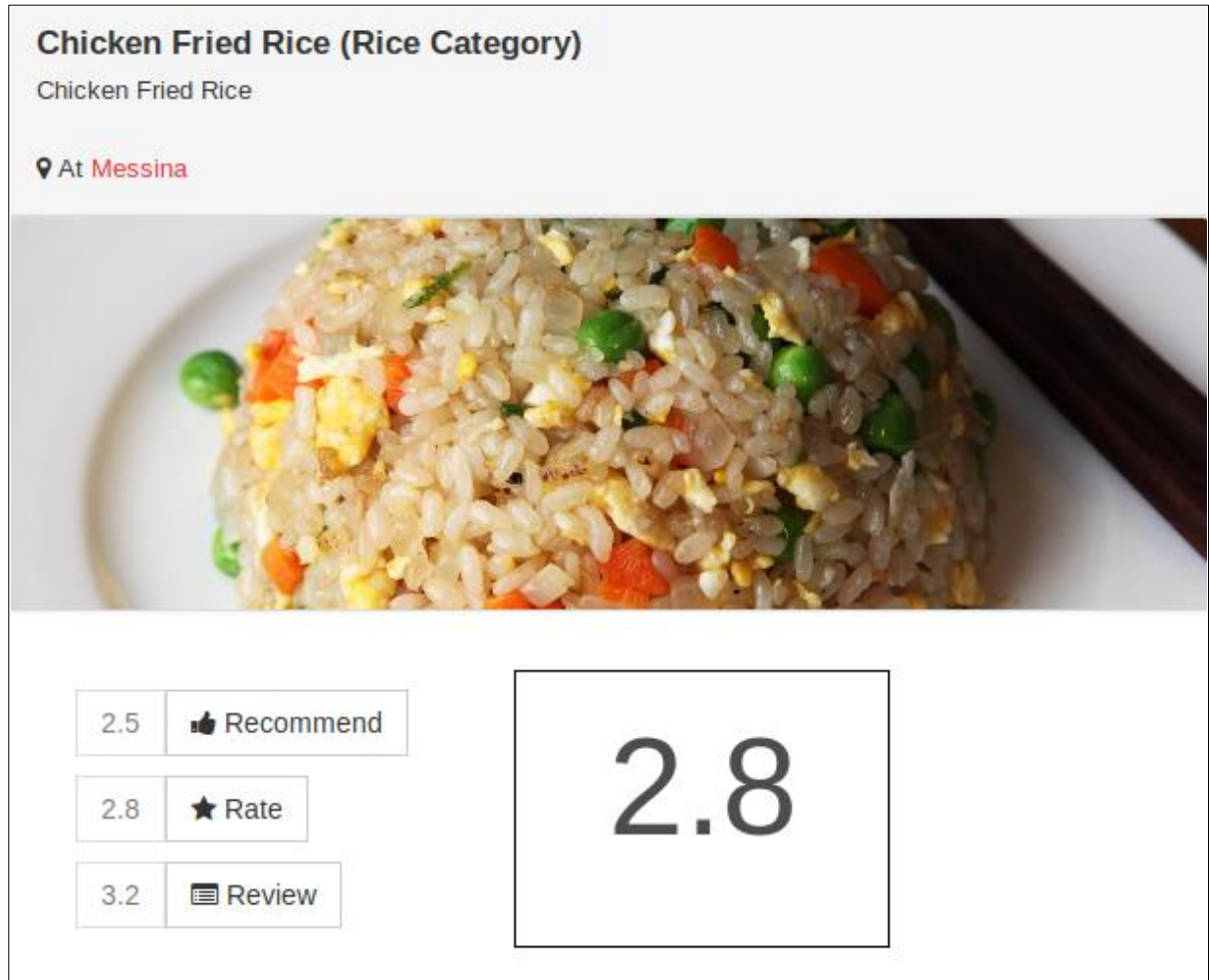


Figure 7.5.1 – Fried Rice of Messina Restaurant

As you can see here,

- Average recommend rating: 2.5
- Average user rating: 2.8
- Average user review rating: 3.2

So, the final rating calculate taking average number of each ratings as follows,

$$\text{Final rating} = \frac{2.5+2.8+3.2}{3} = 2.8$$

Seems this is not a good rating on this menu item. Based on this rating user can get some idea about the restaurant menu item. Let's calculate each and every rating in other sections

7.6 Evaluation of Restaurant Menu Item on User Recommendations

Following is the data for user recommendations on above menu item. In the recommend column "." (dot) is consider as true otherwise false.

```
mysql> select * from RestaurantMenuItemRecommend where menu_item_id = 297;
+-----+-----+-----+-----+-----+
| id    | date                | menu_item_id | recommend | user_id |
+-----+-----+-----+-----+-----+
| 8245  | 2018-01-07 22:54:32 | 297          | .         | 84      |
| 8246  | 2018-01-09 22:54:32 | 297          |           | 26      |
| 8247  | 2018-02-06 22:54:32 | 297          | .         | 73      |
| 8248  | 2018-02-10 22:54:32 | 297          |           | 9        |
| 8249  | 2018-02-14 22:54:32 | 297          |           | 53      |
| 8250  | 2018-04-03 22:54:32 | 297          |           | 98      |
| 8251  | 2018-08-02 22:54:32 | 297          |           | 77      |
| 8252  | 2018-08-03 22:54:32 | 297          | .         | 67      |
| 8253  | 2018-09-28 22:54:32 | 297          | .         | 10      |
| 8254  | 2018-10-12 22:54:32 | 297          | .         | 99      |
+-----+-----+-----+-----+-----+
10 rows in set (0.09 sec)
```

Figure 7.6.1 – Restaurant Menu Item Recommend Data

Let's calculate followings,

- Number of recommendations = 5
- Number of not recommendations = 5

$$\begin{aligned} \text{Average recommendation} &= \frac{\text{numofrecomendations}}{\text{numofrecomendations} + \text{numofnotrecomendations}} \times 5 \\ &= \frac{5}{5+5} \times 5 = 2.5 \end{aligned}$$

We can compare the average user recommendation with Figure 5.2.1

7.7 Evaluation of Restaurant Menu Item on User Ratings

Following is the data for user ratings on above menu item.

```
mysql> select * from RestaurantMenuItemRate where menu_item_id = 297;
+-----+-----+-----+-----+-----+
| id | date | menu_item_id | rate | user_id |
+-----+-----+-----+-----+-----+
| 4340 | 2018-03-01 21:33:09 | 297 | 2.8 | 55 |
| 4341 | 2018-03-12 21:33:09 | 297 | 4.7 | 64 |
| 4342 | 2018-03-27 21:33:09 | 297 | 4.9 | 35 |
| 4343 | 2018-04-14 21:33:09 | 297 | 4.5 | 97 |
| 4344 | 2018-04-18 21:33:09 | 297 | 4 | 41 |
| 4345 | 2018-04-20 21:33:09 | 297 | 4.1 | 79 |
| 4346 | 2018-05-26 21:33:09 | 297 | 0.6 | 21 |
| 4347 | 2018-06-13 21:33:09 | 297 | 0.8 | 74 |
| 4348 | 2018-07-28 21:33:09 | 297 | 2.8 | 20 |
| 4349 | 2018-08-13 21:33:09 | 297 | 0.7 | 80 |
| 4350 | 2018-08-28 21:33:09 | 297 | 1.1 | 24 |
| 4351 | 2018-09-04 21:33:09 | 297 | 1.8 | 36 |
| 4352 | 2018-09-04 21:33:09 | 297 | 0.3 | 65 |
| 4353 | 2018-09-04 21:33:09 | 297 | 3.7 | 73 |
| 4354 | 2018-09-09 21:33:09 | 297 | 4.3 | 69 |
| 4355 | 2018-10-03 21:33:09 | 297 | 4.1 | 48 |
| 4356 | 2018-10-10 21:33:09 | 297 | 3.5 | 1 |
| 4357 | 2018-10-17 21:33:09 | 297 | 1.3 | 27 |
| 4358 | 2018-10-19 21:33:09 | 297 | 3.7 | 48 |
+-----+-----+-----+-----+-----+
19 rows in set (0.01 sec)
```

Figure 7.7.1 – Restaurant Menu Item Rate Data

Here the average rating of this menu item is 2.8 and in the menu item page also it is giving the rating as 2.8 as follows. Please check Figure 5.2.1.

7.8 Evaluation of Restaurant Menu Item on User Reviews

Following is the data for review ratings on above menu item.

```
mysql> select menu_item_id, rate, text from RestaurantMenuItemReview where menu_item_id = 297;
+-----+-----+-----+
| menu_item_id | rate | text                                     |
+-----+-----+-----+
|          297 |    2 | Really a bad meal                       |
|          297 |    4 | Worth trying for the first time         |
|          297 |    2 | It was so terrible                     |
|          297 |    2 | Food quality is not good               |
|          297 |    3 | I won't go there again                 |
|          297 |    5 | Wow what a lovely and tasty meal      |
|          297 |    4 | Not brilliant                          |
|          297 |    4 | I would just like to thank you for an  |
|          297 |    4 | excellent meal we had at your restau  |
|          297 |    4 | rant last night                        |
|          297 |    4 | This meal made my perfect day even    |
|          297 |    2 | better                                  |
|          297 |    2 | The food is not tasty                  |
|          297 |    4 | Very enjoyable evening with this meal |
|          297 |    4 | Very enjoyable evening with this meal |
|          297 |    3 | The food was not fresh                 |
|          297 |    4 | This meal impressed me a lot           |
|          297 |    2 | Food quality is not good               |
|          297 |    2 | The food is not tasty                  |
|          297 |    2 | Very bad food                          |
|          297 |    4 | I love this                            |
|          297 |    2 | Something wrong with this meal         |
|          297 |    4 | Awesome                                |
|          297 |    3 | I won't go there again                 |
|          297 |    4 | This meal impressed me a lot           |
|          297 |    5 | Really nice meal as always, well done |
|          297 |    2 | Really a bad meal                       |
+-----+-----+-----+
24 rows in set (0.04 sec)
```

Figure 7.8.1 – Restaurant Menu Item Review Data

After calculating average rating on reviews, it is giving it as 3.2. We can confirm that checking Figure 5.2.1.

7.9 Summary

This chapter evaluated inputs and output of the used mythologies that discussed in the implementation chapter. Next chapter will describe the conclusion of the research project and further work that can be implemented.

Conclusion

8.1 Introduction

Previous chapter provided the evaluation comparing the test data and this chapter contains a discussion on the result obtained and how the objectives are met. And also here discuss the future works can be done to continue this research to the improve this restaurant recommendation system.

8.2 Conclusion

Our experiment shows that the new approach to recommending restaurant has given high accuracy. Overall conclusion supports the hypothesis that novel restaurant recommendation algorithm can offer high accuracy and efficiency in the recommendation of restaurants. Objective-wise achievements are almost covered and following are some of the achievements.

8.3 Achievement of Objectives

Objective of critically reviewing previous recommendations are evident from the comprehensive literature reviews in the Literature Review chapter. The next objective was developing new algorithm combining data mining techniques also achieved because we used Apriori algorithm and opinion mining techniques to recommend restaurant. The next objective was collection data about the restaurants and the users and getting restaurant information from users. This also achieved getting data set from Yelp and provided the web form to submit restaurant information. The next objective was rate restaurants based on user inputs. Here we rate the restaurant menu

items and based on those rating recommend the restaurants. Also recommend restaurant menu items and restaurants based on the algorithm and user inputs. The next objectives are maintaining profiles of restaurants and users and it also provided in the system. So, we have achieved almost all the objectives as stated in Introduction chapter.

8.4 Further Works

Here we used two main data mining techniques opinion mining and Apriori algorithm for collaborative filtering. In this system, there may be spam reviews. If we can identify them and filter it out, we can use real reviews to recommend a restaurant. Also, there may be spam users registered to the system and do the necessary action to improve their ratings. So, if we can avoid them or if we can identify them this system will be giving 100% accuracy results to the end user and it will really help for them. In order to improve user experience if can use technique like content-based filtering then system can recommend restaurant menu item based on the ingredients of the menu items they prefer. Finally, this system used ratings, reviews and recommendations based on restaurant menu item only. If we can use user ratings, reviews and recommendations on restaurant also then we can give more accurate results to the end user.

8.4.1 Content Based Filtering

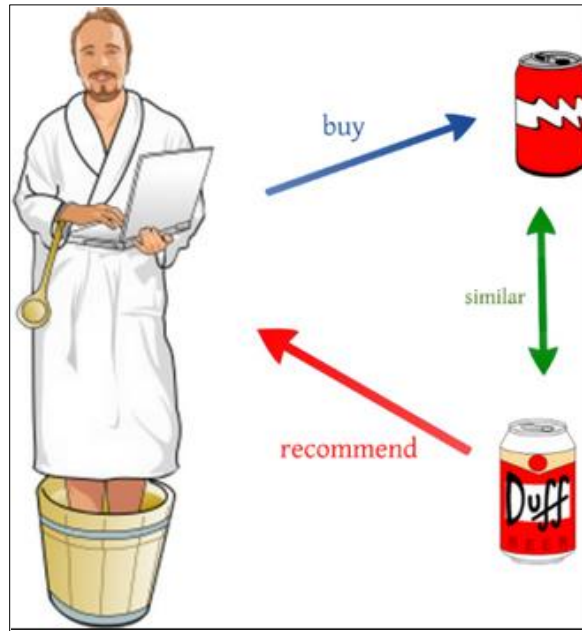


Figure 8.4.1.1 – Content Based Filtering

Content-based filtering [18] is recommend similar kind of items to the user. It will break the items into several chunks and based on that it will calculate the similarity to predict other similar items.

8.5 Summary

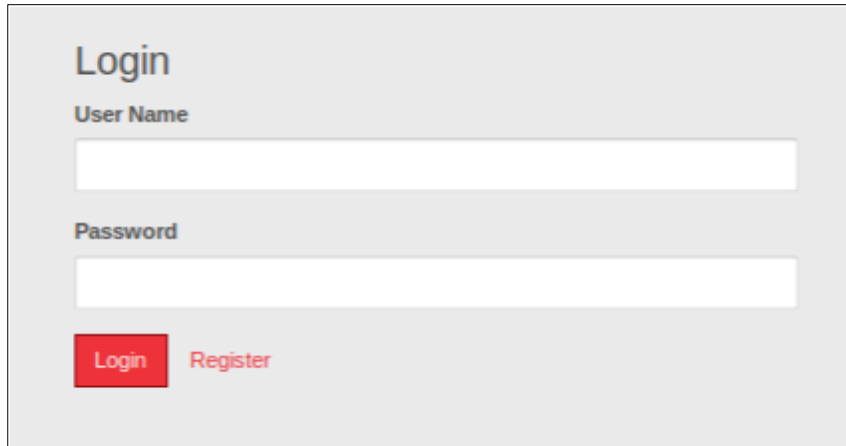
This chapter provided conclusion and all the objective achievements. Here we achieved almost all the objectives as mentioned above. Also, this chapter provided further work of this research to improve this system and give good experience to the end user.

References

- [1] P. Dwivedi and N. Chheda, "A Hybrid Restaurant Recommender," *International Journal of Computer Applications*, vol. 55, no. 16, pp. 20–25, Oct. 2012.
- [2] J. Lee, M. Sun, and G. Lebanon, "A Comparative Study of Collaborative Filtering Algorithms," *arXiv:1205.3193 [cs, stat]*, May 2012.
- [3] T. Tran, "Combining Collaborative Filtering and Knowledge-Based Approaches for Better Recommendation Systems," p. 8.
- [4] A. Submission, "A Preference-Based Restaurant Recommendation System for Individuals and Groups. Team Size: 3," p. 7.
- [5] Y. Gao, W. Yu, P. Chao, R. Zhang, A. Zhou, and X. Yang, "A Restaurant Recommendation System by Analyzing Ratings and Aspects in Reviews," 2015, pp. 526–530.
- [6] T. N. Trang Tran, M. Atas, A. Felfernig, and M. Stettinger, "An overview of recommender systems in the healthy food domain," *Journal of Intelligent Information Systems*, vol. 50, no. 3, pp. 501–526, Jun. 2018.
- [7] "A Survey on Food Recommendation Systems," *International Journal of Advanced Research in Computer Science and Software Engineering*, p. 3, 2016.
- [8] I. TATLI, "Food Recommendation System Project Report," p. 7.
- [9] M. Phanich, P. Pholkul, and S. Phimoltares, "Food Recommendation System Using Clustering Analysis for Diabetic Patients," 2010, pp. 1–8.
- [10] C. Trattner and D. Elswailer, "Food Recommender Systems: Important Contributions, Challenges and Future Research Directions," *arXiv:1711.02760 [cs]*, Nov. 2017.
- [11] D. C. MANOLI and D. G. BROWN, "FoodR Recommender System for restaurants," p. 56.
- [12] A. Gupta and K. Singh, "Location Based Personalized Restaurant Recommendation System for Mobile Environments," in *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013*, 2013.
- [13] J. Almeida, "Personalized Food Recommendations," p. 10.
- [14] N. Mishra and C. K. Jha, "Classification of Opinion Mining Techniques," *International Journal of Computer Applications*, vol. 56, no. 13, pp. 1–6, Oct. 2012.
- [15] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, Maryland, 2014, pp. 55–60.
- [16] C. Kaur, "Association Rule Mining using Apriori Algorithm: A Survey," vol. 2, no. 6, p. 4.
- [17] S. Maitrey and C. K. Jha, "Association Rule Mining: A Technique for Revolution in Requirement Analysis," vol. 4, no. 8, p. 6, 2014.
- [18] M. J. Pazzani and D. Billsus, "Content-based Recommendation Systems," p. 18.

Appendix A

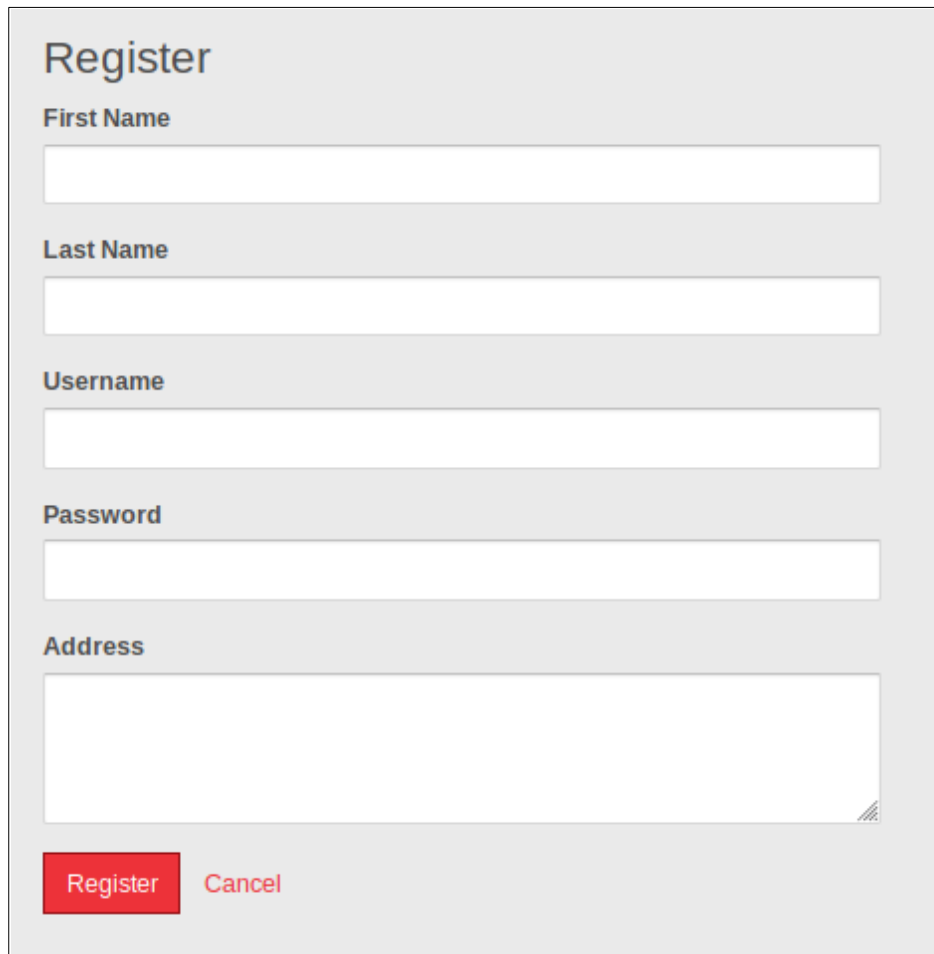
1. User Login Page



The image shows a user login page with a light gray background. At the top, the word "Login" is displayed in a large, dark font. Below it, the text "User Name" is followed by a white rectangular input field. Underneath that, the text "Password" is followed by another white rectangular input field. At the bottom of the form, there are two buttons: a red button with the text "Login" in white, and a gray button with the text "Register" in gray.

Figure Appendix A – User Login Page

2. User Registration Page



The image shows a user registration form titled "Register". It contains five input fields: "First Name", "Last Name", "Username", "Password", and "Address". At the bottom, there are two buttons: "Register" (highlighted in red) and "Cancel".

Register

First Name

Last Name

Username

Password

Address

Figure Appendix A – User Registration Page

3. Restaurant Creation Page

Name

Description

Address

Cuisine

Facilities

Price Range (Rs) for 2 people


Operating Hours (eg: Mon-Fri 11AM - 6PM)

Contact Number

Email

Website URL (http://www.example.com)

Google Map (Click on the map to select a location)




A Google Map of Sri Lanka is displayed, showing major cities and roads. A red pin is placed on the map, indicating a location in Colombo. The map includes labels for Negombo, Kandy, Nuwara E, and Bentota. The Google logo and copyright information are visible at the bottom of the map.

Figure Appendix A – Restaurant Create Page

4. Restaurant View Page

Messina
📍 Messina



Restaurant Information ✎

Serving Cuisines:

Price Range Approx Rs. for two people.


[✉ Email](#) [🌐 Website](#) [☰ Menu](#)

1 [👍 Add Favorite](#)

Facilities

Google Map

Your Review ?



Describe your experience at messina...

☆☆☆☆☆ Rate this

Number of words: 0

[✎ Post Review](#)

Figure Appendix A – Restaurant View Page

5. Restaurant List Page

Restaurants







 <p>Brick House Tavern + Tap Brick House Tavern + Tap 0 reviews</p>	 <p>Messina Messina 0 reviews</p>	 <p>East Coast Coffee East Coast Coffee 0 reviews</p>
 <p>Showmars Government Center Showmars Government Center 0 reviews</p>	 <p>Alize Catering Alize Catering 0 reviews</p>	 <p>Maxim Bakery & Restaurant Maxim Bakery & Restaurant 0 reviews</p>

Figure Appendix A – Restaurant List Page

6. Menu Item List Page

Prawns	Fresh Juices
<ul style="list-style-type: none">• Devilled Prawns• Prawns With Kan Kun• Sweet & Sour Prawns• Garlic Prawns• Prawn In Dry Red Chillies• Battered Fried Prawns• Ginger Prawns• Prawn In Hot Garlic Sauce• Fried Prawns• Prawn With Brown Sauce• Prawn Curry• Mixed Sea Food Devilled• Prawn Toasts• Hot Thai Prawns	<ul style="list-style-type: none">• Orange• Mango• WaterMelon• Pineapple• Papaya• Ambarella
Mutton	Crab
<ul style="list-style-type: none">• Devilled Mutton• Fried Mutton With Onion Salad• Fried Mutton In Spicy Sauce• Mutton Curry• Mutton Stew	<ul style="list-style-type: none">• Singapore Chillie Crab• Fried Crab With Hot Butter Sauce


Figure Appendix A – Menu Item List Page

7. Menu Item Page

Chicken Fried Rice (Rice Category)

Chicken Fried Rice

📍 At **Messina**




2.5 Recommend

2.8 Rate

3.2 Review

2.8

Your Review ?



Describe your experience at chicken fried rice...

☆☆☆☆☆ Rate this

Number of words: 0

Post Review

Figure Appendix A – Menu Item Page

8. Menu Item Creation Page

Menu Item Name

Menu Item Description

Menu Category

Small Price

Medium Price

Large Price

Create

Figure Appendix A – Menu Item Creation Page

Appendix B

1. User Table

```
mysql> desc User;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
email	varchar(255)	YES		NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
user_name	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
address	varchar(255)	YES		NULL	
role	varchar(255)	YES		NULL	

8 rows in set (0.00 sec)

Figure Appendix B – User Table

2. Restaurant Table

```
mysql> desc Restaurant;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
address	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
description	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
lat	float	YES		NULL	
lng	float	YES		NULL	
name	varchar(255)	YES		NULL	
rate	float	YES		NULL	
opening_hours	varchar(255)	YES		NULL	
phone	varchar(255)	YES		NULL	
price_range	varchar(255)	YES		NULL	
website	varchar(255)	YES		NULL	

13 rows in set (0.08 sec)

Figure Appendix B – Restaurant Table

3. Menu Table

```
mysql> desc Menu;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(5)        | NO   | PRI | NULL     | auto_increment |
| name      | varchar(255)  | NO   |     | NULL     |                |
| description | text          | YES  |     | NULL     |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

Figure Appendix B – Menu Table

4. Menu Item Table

```
mysql> desc MenuItem;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(5)        | NO   | PRI | NULL     | auto_increment |
| menu_id   | int(5)        | NO   | MUL | NULL     |                |
| name      | varchar(255)  | NO   |     | NULL     |                |
| description | text          | YES  |     | NULL     |                |
| small_price | decimal(10,2) | YES  |     | NULL     |                |
| medium_price | decimal(10,2) | YES  |     | NULL     |                |
| large_price | decimal(10,2) | YES  |     | NULL     |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Figure Appendix B – Menu Item Table

5. Restaurant Controller

```
@CrossOrigin(origins = "http://localhost:4200")
@RestController
@RequestMapping("/restaurants")
public class RestaurantController {
    @Autowired
    private RestaurantService restaurantService;
    @GetMapping
    public List<Restaurant> getRestaurants() {
        return restaurantService.getAllRestaurants();
    }
    @GetMapping(value =("/{id}")
    public Restaurant getRestaurantById(@PathVariable("id") Integer id) {
        return restaurantService.getRestaurantById(id);
    }
    @PostMapping(value = "/add")
    public Restaurant addRestaurants(@RequestBody Restaurant restaurant) {
        return restaurantService.saveRestaurant(restaurant);
    }
}
```

6. User Controller

```
@CrossOrigin(origins = "http://localhost:4200")
@RestController
@RequestMapping("/users")
public class UserController {
    @Autowired
    private UserService userService;
    @PostMapping(value = "/create")
    public User createUser(@RequestBody User user) {
        return userService.saveUser(user);
    }
    @GetMapping(value =("/{id}")
    public User getUserById(@PathVariable("id") Integer id) {
        return userService.getUserById(id);
    }
}
```

7. AprioriUtil

```
public final class AprioriUtil {  
  
    public static void getAssociations(String arffFilePath, int numberOfRules) throws  
    Exception {  
        AssociationRules associationRules = getAssociationRules(arffFilePath,  
            numberOfRules);  
        List<AssociationRule> associationRuleList =  
            associationRules.getRules();  
        for (AssociationRule associationRule : associationRuleList) {  
            Collection<Item> premiseCollection = associationRule.getPremise();  
            List<String> items = new ArrayList<>();  
            for (Item item : premiseCollection) {  
                Attribute attribute = item.getAttribute();  
                String met = attribute.name();  
                items.add(met);  
            }  
            Collection<Item> consequence = associationRule.getConsequence();  
            List<String> predictItems = new ArrayList<>();  
            for (Item item : consequence) {  
                Attribute attribute = item.getAttribute();  
                String met = attribute.name();  
                predictItems.add(met);  
            }  
            int premiseSupport = associationRule.getPremiseSupport();  
            int totaSupport = associationRule.getTotalSupport();  
            double confidenceNonFormatted = (float) totaSupport /  
                premiseSupport;  
            DecimalFormat twoDForm = new DecimalFormat("#.###");  
            String formattedNumberString =  
                twoDForm.format(confidenceNonFormatted);  
            Double formattedConfidence = Double.valueOf(formattedNumberString);  
            System.out.println(items + " --> " + predictItems + " | " +  
                formattedConfidence);  
        }  
    }  
}
```

8. SentimentUtil

```
public final class SentimentUtil {  
    public static float getSentimentRate(String line) {  
        Properties props = new Properties();  
        props.setProperty("annotators", "tokenize, ssplit, parse, sentiment");  
        StanfordCoreNLP pipeline = new StanfordCoreNLP(props);  
        float mainSentiment = 0;  
        if (line != null && line.length() > 0) {  
            int longest = 0;  
            Annotation annotation = pipeline.process(line);  
            for (CoreMap sentence :  
                annotation.get(CoreAnnotations.SentencesAnnotation.class)) {  
                Tree tree = sentence.get(SentimentCoreAnnotations.SentimentAnnotatedTree.class);  
                int sentiment = RNNCoreAnnotations.getPredictedClass(tree);  
                String partText = sentence.toString();  
                if (partText.length() > longest) {  
                    mainSentiment = sentiment;  
                    longest = partText.length();  
                }  
            }  
        }  
        return mainSentiment + 1;  
    }  
}
```