NOTTINGHAM FOOD FINDER

Introduction

The food finder system 'Foodie' is the search system for a user to access the list of restaurants against food as an input string. The system acts as a personal assistant to the user which will not only provide the list to the restaurants but also guide to the user on how to reach at the preferred location using maps. Anyone with an internet connection can access the website to locate the restaurant and the map operates in real time which updates the location as the user changes his location. The website offers language settings for different users in the region which can be easily extended for different locales and languages.

The development of the website is divided into four incremental phases and the prototype will be implemented as per phase 1 specification (see Appendix C), which will serve as proof of concept. For further releases, the search can be performed using restaurant's data such as: - the phone, cuisine, home delivery, WiFi, Restaurant type, opening hours and closing hours to provide as much information as possible for the user's final decision. Also, the website will have a feature to rate the restaurants rated by the users for the users. The online community will help other people by sharing their review of the food or restaurant based on their interest.

Analysis

In e-commerce, the user interface of any website is the first impression of the business and it should be developed as the marketing tool. The site should be approachable, simple to use and provide solutions to the customer's needs. For this reason, the food finder Foodie's home page must be simple and its focus should be on the search box with filters which gives clear idea that it is search system for the food.

The search box should appear in the center of the home page and the website should navigate to the next page to retrieve the list of restaurants. The home page act as a separate entity and results will be displayed on next page. The links to other functionalities such as: - user login, change language, find restaurant will appear on the top-left of the page. The website must offer more functionality as advance search with the filters to restrict the list because the user might not be interested in the retrieved list of restaurants. For example: - user might be interested in the restaurant which is 5 miles from his location instead of the ones nearby user's location. The following functionalities will discuss the requirement as per user's needs.

1. Advance Search & Filters

The 'Advance Search' will have following options 'Home Delivery', 'Cuisine', 'Restaurant Type', 'Opening hours' and 'Closing Hours' for the user's convenience. For example: - restaurants that deliver food at home or serve vegetarian food or restaurant with bar or the user might be interested in specific cuisine such as: - Italian, Mediterranean or Indian, and if it's late night or early morning, the user might want to know opening or closing hours. These filters will be available on the searched results page to offer user a chance to refine the search.

2. Calculate Distance

The distance and travel time with the restaurant will provide the quick solution to the next problem of the user after searching for the restaurants i.e. which restaurant is closer to the user's location. For this reason, the website must detect the location coordinates of the user to get the exact location to provide better suggestions. If not, then the location must be taken as an input from the user. This can be implemented developing an algorithm or for quick solution, use 'Google Places API' for easy retrieval of the location based on user input. The API offers a type-ahead feature which provides a suggestion as the user type each character in the search field and autocomplete will fill the rest (Google, 2017).

After getting an input from the user it is must to the coordinates to proceed with the calculations. The feature can be implemented using HTML5 Geolocation API provides the solution to this problem by providing the latitude and longitude to the user (W3C, 2013). Also, this API updates the location when the device moves from one location to another which can benefit the users who are accessing website on the handheld device (w3Schools, 2017). Google Maps are famous for its distance calculation, real-time traffic update and alternate route suggestions that it provides every time it sees a shorter route to reach the destination. This functionality can be implemented by embedding a map into the code using iframe tag. Also, the developer can adjust the size of the map by providing the height and the width on the map (Google, 2017).

3. Languages & Locales

The website should be customizable to regional preferences which will require locale and language configurations. In locale settings, the website should control how the numbers, dates and times must be displayed for the selected region. In language settings, the user must be offered options for the languages in the selected region. For example: - In Canada, the user must

have the option for English and French since these are official languages of the country (Stewart, 2005). The locale & language implementation must be modular and loosely coupled to easily integrate new locales or languages for another region into the system without changing the code from page to page.

4. Session Management

The website must allow separate sessions for each user and it should store the repetitive values in the session, cookie, hidden fields for session management as HTTP is a stateless protocol which maintains no state for user's subsequent request (Kolšek & Security, 2002) . For example: - language or locale data must be maintained throughout the user's subsequent requests. For this purpose, a flag or a string input must be preserved to notify the server every time a user navigates from page to page until the user decides to leave the system.

5. Pagination

The website must have the functionality to divide the list into pages to enhance simplicity for easy navigation. Using this approach, the user will not have to scroll through the list of restaurants. For example: - if results are not filtered, default search will populate all the restaurants from the database which will create a long list of restaurants on the page. The page should display 10 restaurants near to the user's location in ascending order as the user might not be interested in all the data.

The website must follow software architectural design patterns to maintain the code for example: - dividing the logic into Presentation layer, Business logic layer and Persistence layer which will help encapsulating the functionality and enhance component design (IBM, 2017). The development process of the website is divided into four incremental phases separated as per functionalities

Phase 1: Proof of Concept enabling basic search system with directions
Phase 2: Advance search based on place & search filters functionality
Phase 3: Admin Functionality to assign role & permission to add/delete restaurants
Phase 4: Forum for Ratings and Community

Note: - The business requirement specifications are discussed in depth in Appendix C.

Database

The database is implemented in 3NF for search and administrator functionality to avoid duplication of the data. The first entity in the search implementation is a Restaurant which has location, timing, address and phone as its attributes. The second entity is 'Menu' which has type, cuisine, and price as its important attributes. The search will be performed using 'rest' and 'menu' for fetching the list of restaurants. All administrator functionality tables are created with prefix 'admin_'. The relationships for tables 'timing', 'rating' and 'menu' is separated from 'rest' table to remove the duplication of the data. Menu to rest is many to one relationship as one Restaurant can have multiple rows in Menu Items. Day and restId in timing table should be composite primary key because there are 7 days in the week. So, there should exist only seven entries per restaurant. This would restrict the entries. The relationships are identifying relationships as shown in Figure 1 as there must be an entry into parent table i.e. rest for entries in child table. The Search tables and Administrator table are not related. Figure 1 shows the enhanced (or extended) entity relationship model of the current implementation.

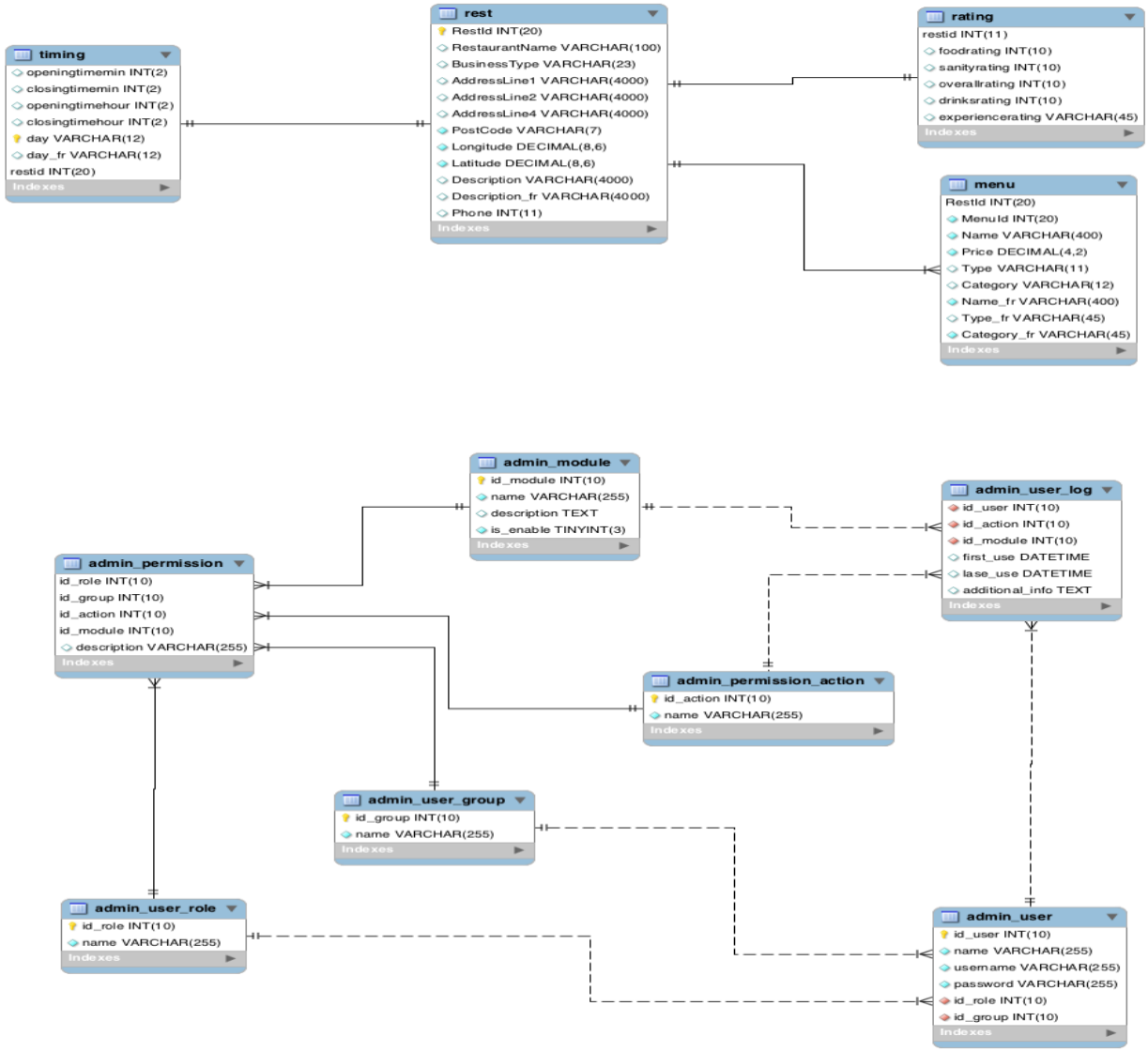

Figure 1: EER diagram for phase1, 2 and 3 implementations in 'foodfinder' schema

The tables starting with prefix 'admin_' will be used by the administrator and staff. The prefix is used from the business perspective; it shouldn't be confused with administrator. The roles setup will differentiate between the administrator and staff. For example: - a role which has full functionality will be called as an administrator but one who cannot delete will be any staff member.

Further, the database must be implemented for the forum and users in the system. The users will be administrator, staff, and customers, where administrator and staff should be developed separately as they are employees in the organization and users table should focus only on the forum

functionality. Every post should be maintained against username and replies will be tracked.

```
CREATE TABLE `user` (
  `firstname` varchar(400) CHARACTER SET latin1 DEFAULT NULL,
  `lastname` varchar(400) CHARACTER SET latin1 DEFAULT NULL,
  `username` varchar(400) CHARACTER SET latin1 NOT NULL,
  `phone` int(11) DEFAULT NULL,
  `email` varchar(200) CHARACTER SET latin1 DEFAULT NULL,
  `password` varchar(400) CHARACTER SET latin1 NOT NULL,
  `addressline1` varchar(400) CHARACTER SET latin1 DEFAULT NULL,
  `addressline2` varchar(400) CHARACTER SET latin1 DEFAULT NULL,
  `addressline3` varchar(400) CHARACTER SET latin1 DEFAULT NULL,
  `addressline4` varchar(400) CHARACTER SET latin1 DEFAULT NULL,
  `zipcode` varchar(11) CHARACTER SET latin1 DEFAULT NULL,
  `userid` varchar(45) CHARACTER SET latin1 NOT NULL,
  PRIMARY KEY (`userid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Figure 2: User DDL

Note: - Please refer to 'Appendix E' for restaurant's DDLs.

Design

The website uses PHP for the development as it's server side scripting language, the code is executed on the server rather than client's machine. It can be easily embedded with HTML tags and scripts so the crucial role can be written in PHP (Harwani, 2015). The database is developed in relational database system MySQL since it fast and provide faster transactions when accessed by simultaneous users (Axmark & Widenius, n.d.).

From the design perspective, the front end is developed using HTML 5, CSS, Bootstrap and JQuery, server side is implemented using PHP and for easy integration, XAMPP server is used as it provides full setup for apache and MySQL server. The user will send a request to the server and data which satisfies the search criteria will be fetched from the database (as shown in Figure 3 & 4).

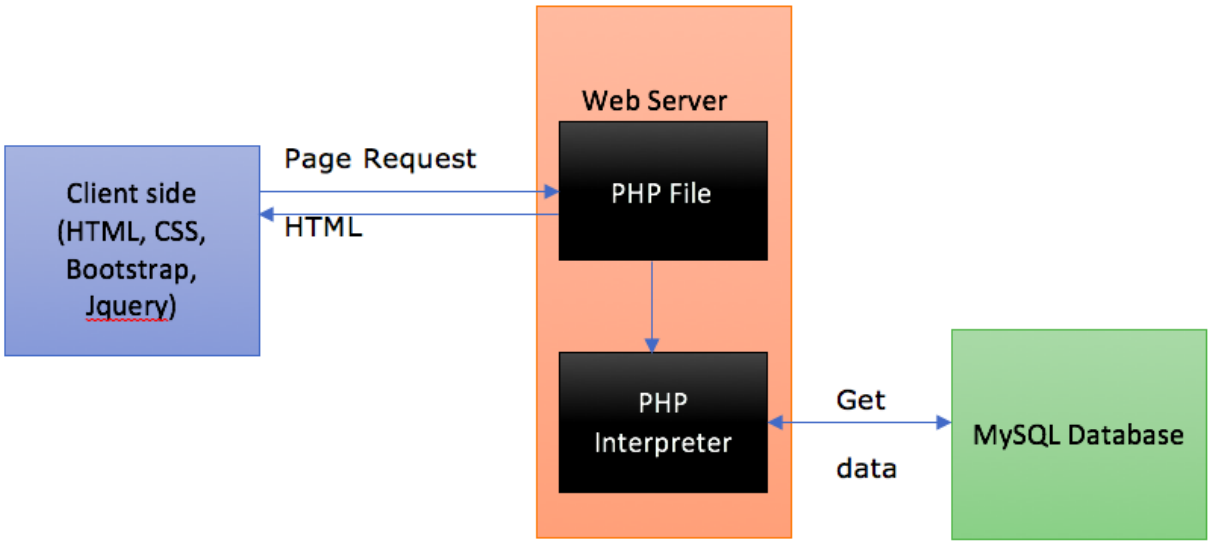

Figure 3:Client side–Server side communication diagram. Client Side -> HTML 5, CSS, Bootstrap 3, JQuery. Server side -> PHP. Database -> MySQL, Server -> XAMPP.

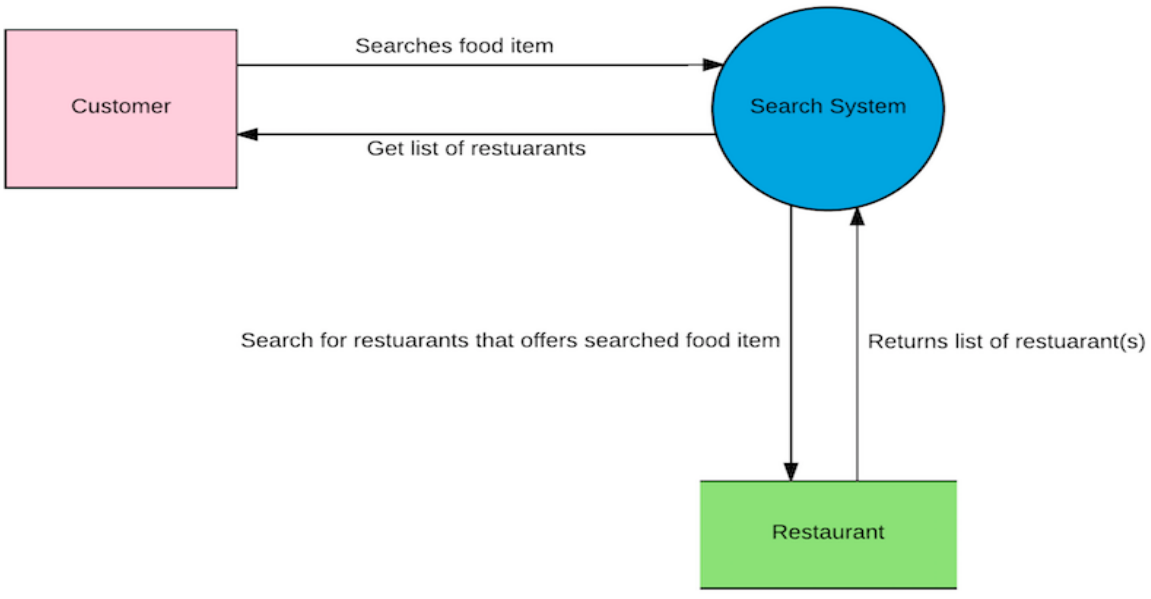

Figure 4: Data Flow Diagram (Phase 1)

The Foodie search system is divided into two parts: Administrator functionality and Front End. Administrator acts as a backend process to manage the data on the Front End. The front end is further divided into the search engine for finding food, the list of restaurants for bulk access and forum for Foodie community. The relationship among the modules and their access (illustrated in figure 5) where the administrator is the one who will add the data into the system for a user to access information on restaurants and community.

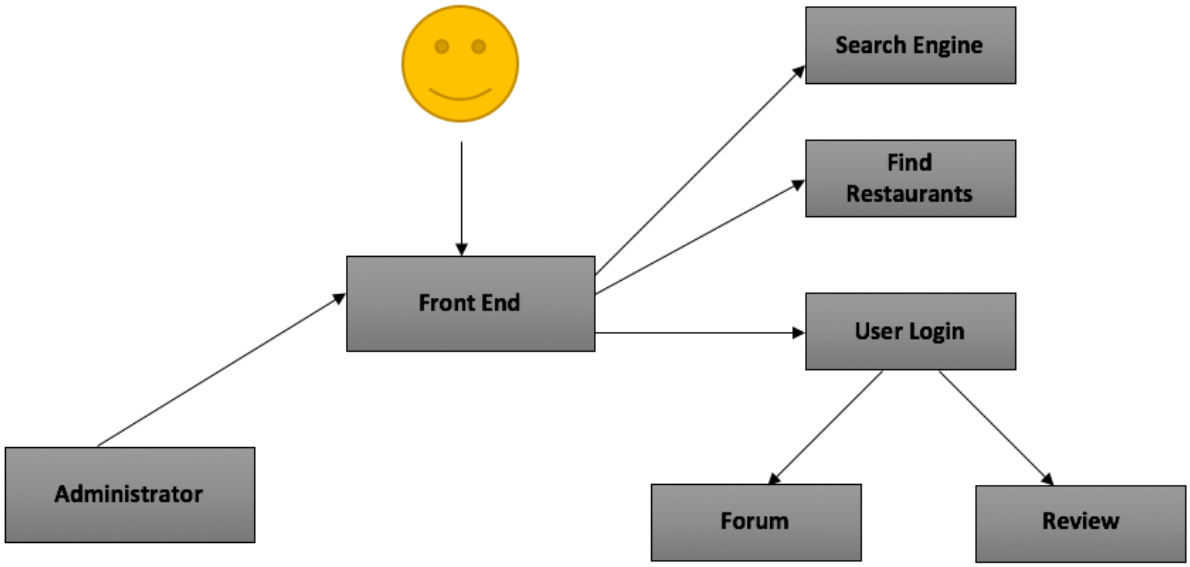

Figure 5: High Level Diagram of the Foodie

The different users of the system are Administrator, Staff and Customer. The Administrator can create role, permissions, restaurants and assign the roles to Staff members to create/add/delete data. The Customer can search data for food or restaurant, join online Foodie community and review a restaurant.

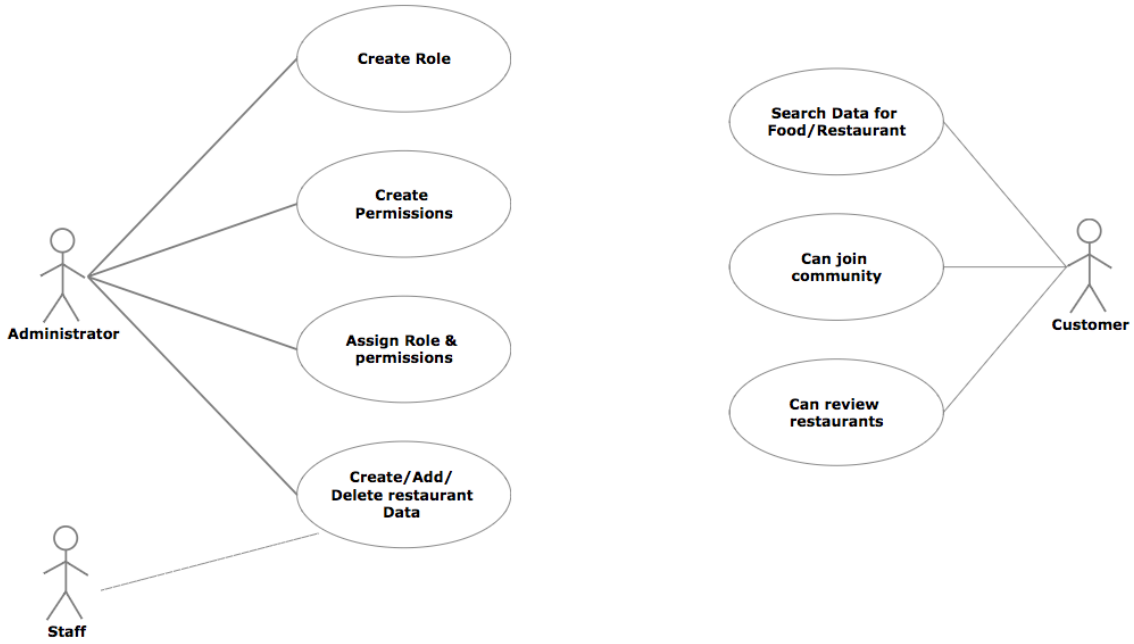

Figure 6: Use case scenario

For easy configuration, pagination & language variables are maintained in a separate file to add/update/remove configurable files for future enhancement. The other reason is to have a centrally configured file for the functionality so that when change requirement comes, the developer doesn't have to look for references all over in the file for one variable. For example: - for language, the user may not want to see French on one page and English on another. The software architecture must be divided into layers based on their functions to enhance the maintainability of the code. For example: - the front-end layer is UI layer which is visible to the user, business layer in code manages the main logic of the action event and data layer manages the communication to the database (Brüstel, et al., 2012). Further, the data connection established must be closed once the transaction is completed so that there are no unwanted connections in the memory.

Since there would multiple languages in the website, the 'UTF-8' data char set must be maintained for both database and HTML to provide letters other than English alphabets (w3schools, 2017). The website should also take care of the SQL injections to protect against the unexpected input data by users, alteration to the database structure, corruption of data or revelation of private and confidential information (Boyd & Keromytis, 2004).

Description of prototype

The prototype is the proof of concept for the phase 1 where the user searches for the food and get the list of restaurants with the directions sorted by the distance as the searched results. By default, distance is set to 5 miles, and the empty search returns the list of all the restaurants in the database. The results' list is further divided into pages with the help of pagination to restrict the list on the page. By default, the pagination limit is set to 1 by using a PHP variable '**$setLimit = 5**' which can be configured in '**variables.php**' page. The database is designed to save latitude and longitude for the destination into the tables to save time to fetch coordinates every time for search per user. For testing, 10 restaurants with their details are stored in 'rest' table and their 5 dishes are stored in 'menu' table. The rest of the section talks about the various cases that can be performed in the prototype.

Home Page: On the home page, search bar and distance slider give you quick option to search the food. The design is kept simple with the content aligned in the center to focus only on the search box. The language and user login are provided on the top left. The 'Find Restaurant', 'Forum', and 'User' are not action event in this prototype.

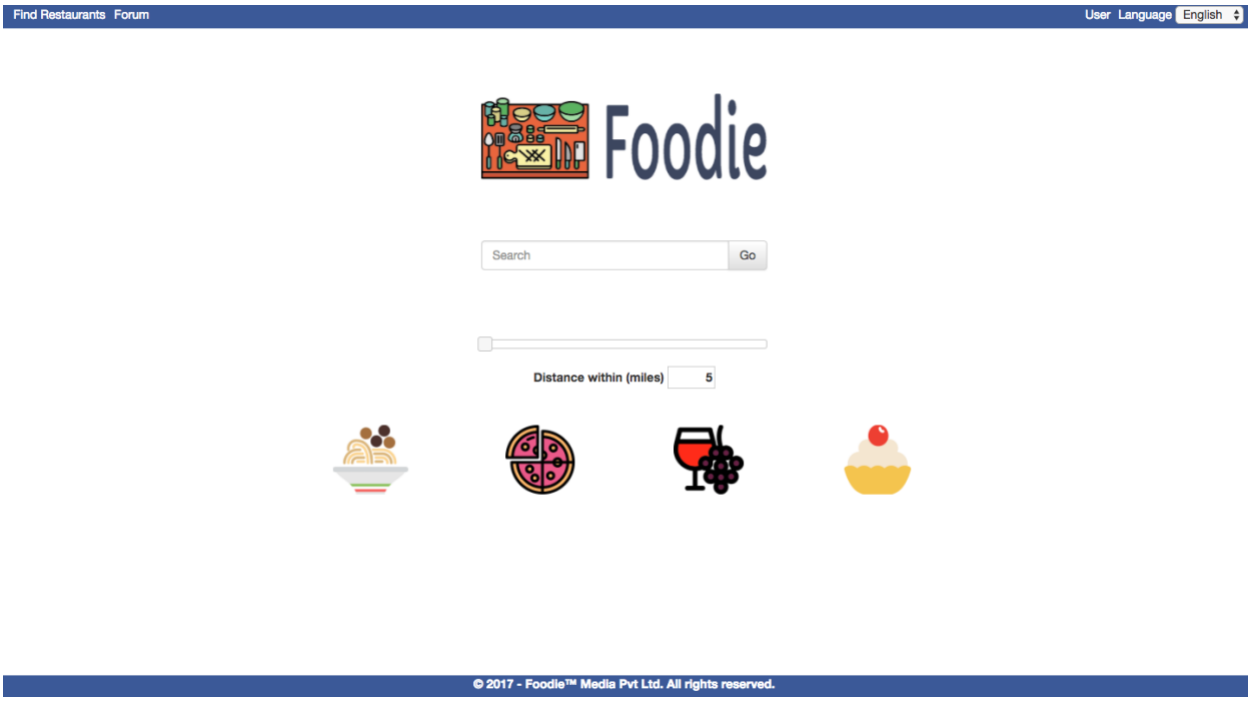

Figure 7: Home Page

Language Selector: Further, dropdown on top-right is provided to switch between languages. Once the language is changed, the content will be loaded again in the selected language. The language preference throughout the

session and will not be unloaded until the user deletes the session history from the page. This setting is kept on purpose because the users might not want to select the language option every time they visit the page.

Figure 8: Language Selector

Default Search: The second searched results page returns the list of the restaurants based on empty search and shows you Google map for distance and travel time. Further, the map height and width can be adjusted by changing the '<iFrame>' settings in the code. Currently, the map takes the location coordinates from the GPS enabled device with the help of HTML5 Geolocation code and then embed the Google Map using iFrame in the HTML code. The result of the map interprets differently for each destination i.e. restaurant since their coordinates are stored in the database. But source location updates as the device moves. If device blocks the location or is not GPS enabled, then Jubilee Campus is taken as the default location.

Note: - The pagination has changed for this picture. The final settings will show five results per page.

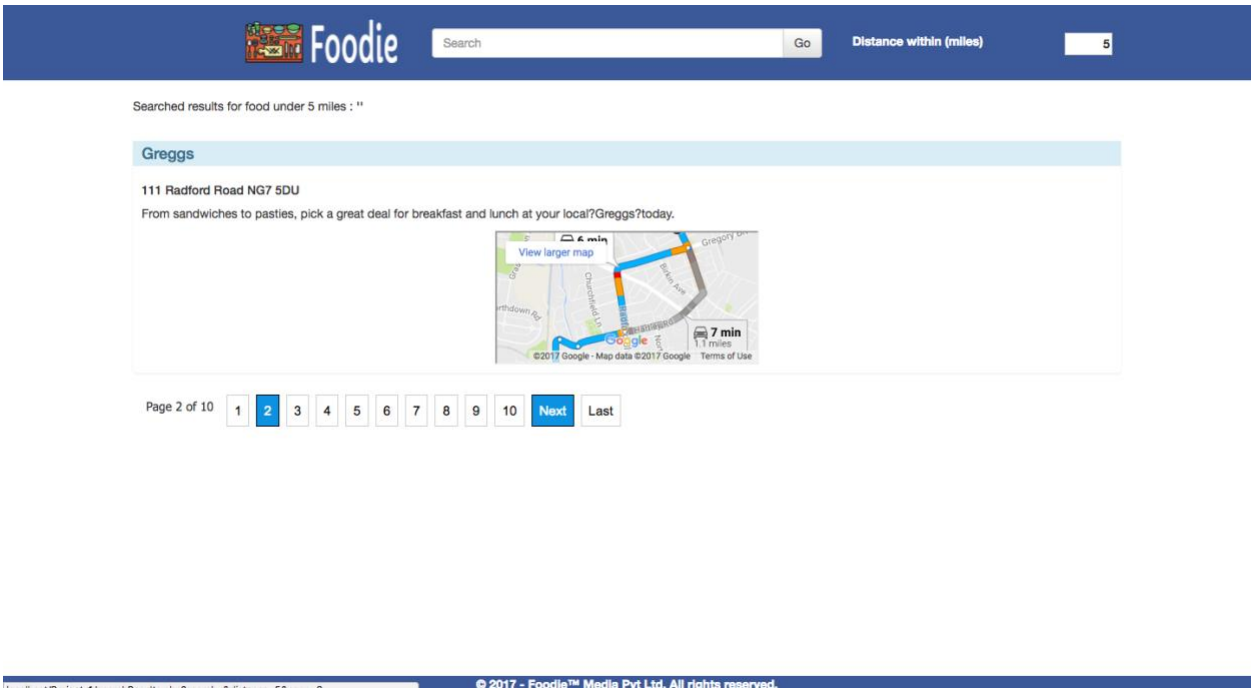

Figure 9: Result with the direction displayed in pages

<u>Search for 'crème' within 0.6 mile</u>: This search returns the list of restaurants which satisfies the search criteria. In this case, there's only one restaurant satisfied the search criteria. In this case, pagination is not visible since there was one result on the list. But if it was one than one, the pagination should be visible.

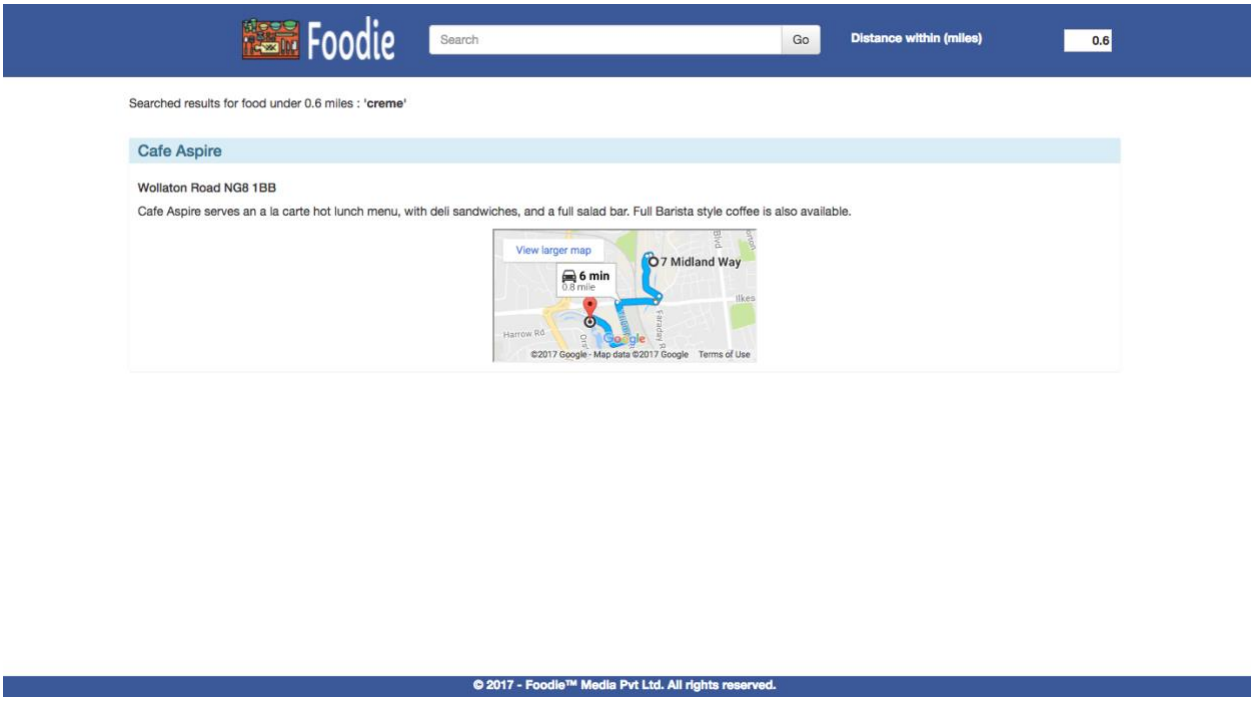

Figure 10: Search for 'crème' with 0.6 mile

Search for pasta within 5 miles: This search should also return the list of the restaurants that satisfies the search criteria. In this case, pasta is served only in Bella Italia.

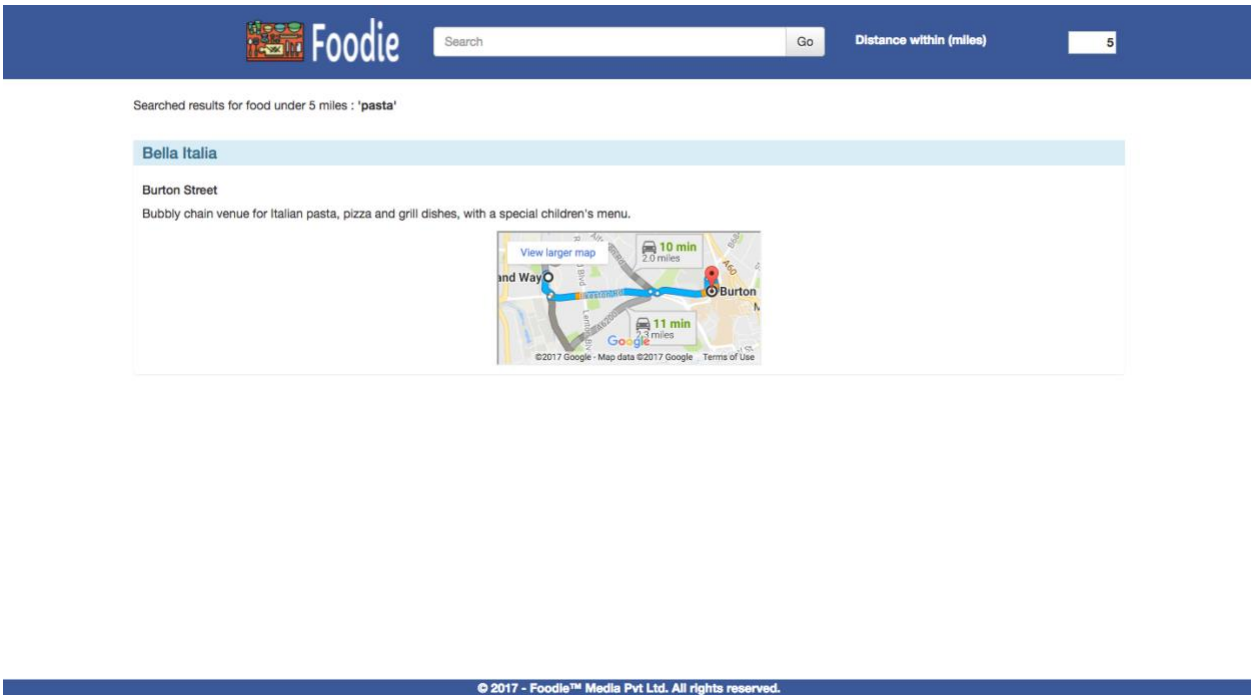

Figure 11: Search for pasta

Responsive Design: The implementation of the responsive design is achieved using Bootstrap and CSS. The device media configurations are maintained in 'tablets.css' and 'phones.css' files under style.

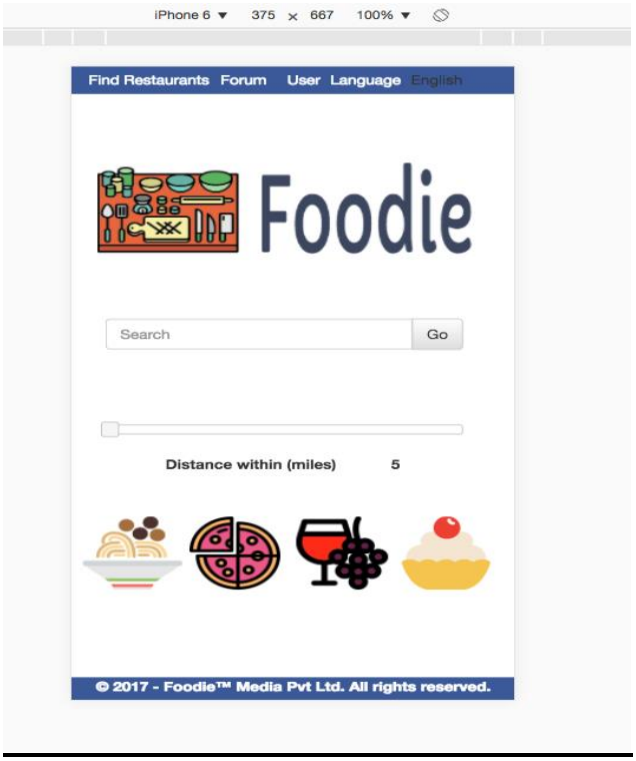

Figure 12: Mobile Search

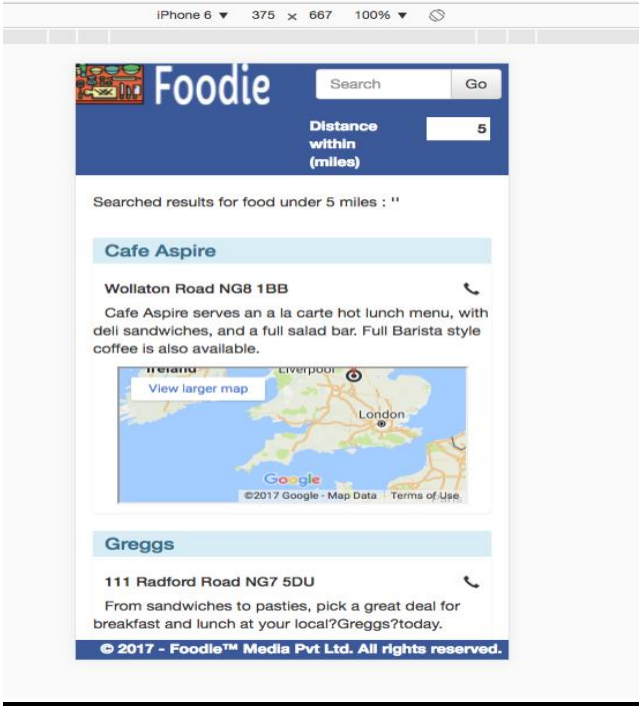

Figure 13: Responsive Design Results

Language Configuration: Language configuration for English and French languages in the Foodie is easy to add. The language files are defined in the array variable and then set in session variable based on user preference. The session variable for language is preserved until the user deletes the session data or change the preference. In figure 14 & figure 15, screen shots of the PHP code for language configurations are listed in English and French languages respectively.

```php
<?php
/**
 * Created by PhpStorm.
 * User: shobhikabharti
 * Date: 4/22/17
 * Time: 18:30
 */
/*
-------------------
Language: English
-------------------
*/
$language = array();
$language['PAGE_TITLE'] = 'Nottingham Food Finder';
$language['LANG'] = 'Language';
$language['Go'] = 'Go';
$language['DISTANCE'] = 'Distance within (miles)';
$language['SEARCH'] = 'Search';
$language['SEARCH2'] = 'Searched results for food ';
$language['UNDER'] = 'under ';
$language['MILES'] = 'miles';
$language['EXCEPTION1'] = 'We\'re sorry to have no results at present moment.';
$language['EXCEPTION2'] = 'Nothing seems searched. Aren\'t you hungry?';
$language['RESTAURANTS'] = 'Find Restaurants';
$language['FORUM'] = 'Forum';
$language['USER'] = 'User';
?>
```

Figure 14: list of variables in english.php

```php
<?php
/**
 * Created by PhpStorm.
 * User: shobhikabharti
 * Date: 4/22/17
 * Time: 18:30
 */
/*
-------------------
Language: French
-------------------
*/
$language = array();
$language['PAGE_TITLE'] = 'Nottingham Ailments Chercheur';
$language['LANG'] = 'La langue';
$language['Go'] = 'Aller';
$language['DISTANCE'] = 'Distance à l\'intérieur (miles)';
$language['SEARCH'] = 'Rechercher';
$language['SEARCH2'] = 'Résultats recherchés pour la nourriture ';
$language['UNDER'] = 'en dessous de ';
$language['MILES'] = 'miles';
$language['EXCEPTION1'] = 'Nous sommes désolés de ne pas avoir de résultats à l\'heure actuelle..';
$language['EXCEPTION2'] = 'Rien ne semble être recherché. N\'as-tu pas faim?';
$language['RESTAURANTS'] = 'Trouver des Restaurants';
$language['FORUM'] = 'Forum';
$language['USER'] = 'Utilisateur';
?>
```

Figure 15: list of variables in french.php

Database configuration: The tables are stored in MySQL database with username as 'root' and password as 'password' which is provided in the connection variables as shown in Figure 1. With the help of 'mysqli', the connection is established between the application and SQL server. The connection code is maintained in the separate file 'connection.php'. In case the dump for schema is executed under the different name or the name of username, password, servername and database schema is changed, the strings can be modified for new configurations.

```php
<?php
/**
 * Created by PhpStorm.
 * User: shobhikabharti
 * Date: 4/22/17
 * Time: 10:50
 */


$servername = "127.0.0.1";
$username = "root";
$password = "password";
$database ="foodfinder";

// Create connection
$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

?>
```

Figure 16: Connection Configuration

Note: - If back button from the browser is pressed, the page refreshes and shows you default search options with the preferred language since language variables are maintained in session.

Conclusion

The requirement for website listed in Phase 1 is complete. Unit test cases executed are attached in 'Appendix D'. The mobile website can be enhanced for the distance. From the design perspective, the functionality should be implemented in the layers for maintenance. The components must be loosely coupled and encapsulated for further implementation of phases.

Deliverables (Pen Drive)

1. Project Code - Project_1
2. Database Dump File - Dump20170505.sql
3. Requirements for the website – XAMPP and MSSQL should be configured on the server

Further Suggestions

Forum functionality implementation in MongoDB as an alternative to MSSQL to decrease the load on the MSSQL server for faster execution of queries. "MongoDB is a non-relational database which provides high flexibility at addition or deletion of an attribute from the database because they do not have a fixed database schema. MongoDB provided lower execution times than MySQL in all four basic operations, which is essential when an application should provide support to thousands of users simultaneously" (Gyrödi, et al., 2015). These features would help in forum implementation as users comments on the threads simultaneously.

Appendix

A. Glossary

Abbreviation	Term	Description
-	JQuery	jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers
-	Bootstrap	Bootstrap, a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development.
CSS	Cascading Style Sheets	Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation

		of a document written in a markup language.
HTML	Hyper Text Markup Language	HTML describes the structure of Web pages using markup
MySQL		Open source RDBMS
PHP	Hypertext Preprocessor	PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML
SQL	Structured Query Language	SQL is a standard language for storing, manipulating and retrieving data in databases.

B. Importing Data Dump

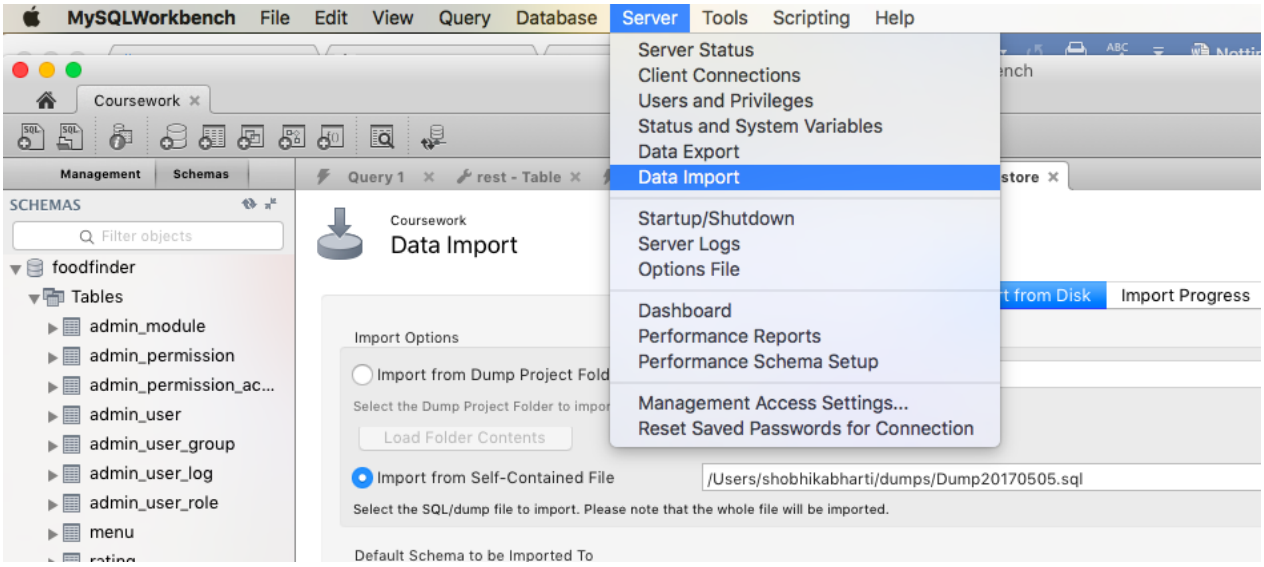

Figure 17: MySQL Workbench data import option

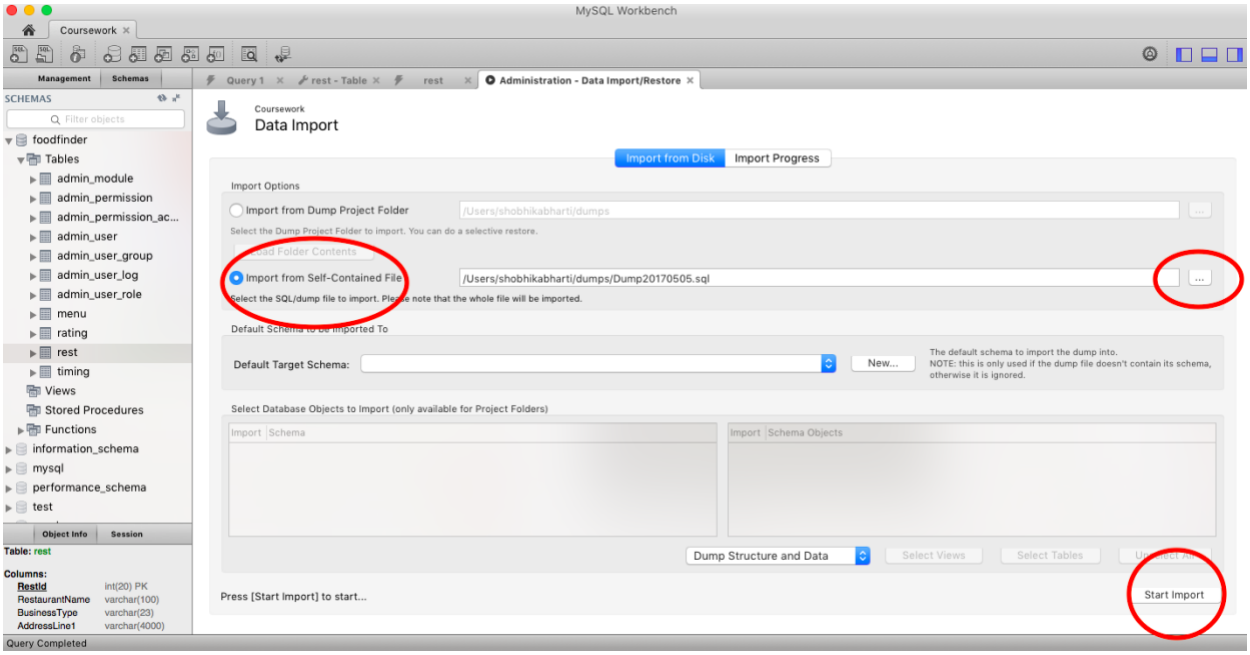

Figure 18: Select second option to import data dump into the database

C. Requirement Specification

Phase 1: Proof of Concept enabling basic search system with directions

Business Rules for phase 1

1. The system in Phase 1 has two pages: Home page and Searched results page.
2. The home page has the search bar, the language selector and the distance slider.
3. The search bar must accept string input.
4. The language selector displays at the top right of the home page.
5. User has an option to select language from English and French from the language selector.
6. The search can be performed from search bar or distance filter or both
7. Search should be performed on food item.
8. The default distance in distance filter is 5 miles but user has the option to change the value from the slider and the textbox.
9. The distance slider which has a value till 500 but user can define a value in the textbox in decimal format.
10. Searched results displays list of results on next page.
11. Each searched item has a map for navigation with an estimated time to reach the searched destination.

12. On searched results page, user has an option to search again either on search bar or distance or both.
13. The results are sorted in ascending order i.e. the restaurant serving the food item nearby is listed as first item on the list.
14. Pagination is allowed on the search results page if it exceeds the length. The default limit is set to 1.
15. Pagination is not an action event if it reaches last page
16. The searched results page has a Foodie icon next to the search bar, if clicked, it navigates to home page.
17. The website should maintain the session for every user. For example: - User 1(English) and User 2(French) should have their sites in different languages.
18. Responsive Design – Front end should display the content on website as well as handheld devices.

Phase 2: Advance search based on place & search filters functionality

Business Rules for Phase 2

Phase 2 is divided into two sections i.e. 'Advance search' on home page and filters for searched results on second page.

A. Business rules for Advance Search on home page

1. The user can filter search based on 'Home Delivery', 'Cuisine', 'Restaurant Type', 'Opening hours'.
2. In 'Home Delivery', Restaurant must have an option to deliver the food at home.
3. 'Cuisine' must display an option to filter from these cuisines: American, Asian, BBQ, Bar Food, Belgian, Breakfast, British, Burger, Cafe, Caribbean, Chinese, Coffee and Tea, Contemporary, Diner, Finger Food, Fish and Chips, French, Greek, Grill, Healthy Food, Indian, International, Italian, Japanese, Kebab, Korean, Latin American, Malaysian, Mediterranean, Mexican, Middle Eastern, Pakistani, Pizza, Portuguese, Sandwich, Seafood, Spanish, Steak, Sushi, Tapas, Thai, Turkish.
4. Restaurant Type must display an option to filter Dine-Out, Take-Away, Delivery, Bar and Drinks & Nightlife.
5. Opening hours must have opening and closing time of restaurant.

B. Business rules for Filters for searched results

1. The searched results page must have filters for 'Sort-By', 'Home-Delivery', 'Location', 'Cuisine', 'Restaurant Type', and 'Wi-Fi'.
2. The 'Sort-By' filter must have an option to sort ascending or descending order.
3. The 'Sort-By' filter must have option for 'Popularity', 'Rating', 'Cost', and 'Recently added' restaurants.
4. In 'Home Delivery', Restaurant must have an option to deliver the food at home.
5. The 'Location' filter must have an option to choose locations in East-Midlands.
6. 'Cuisine' must display an option to filter from these cuisines: American, Asian, BBQ, Bar Food, Belgian, Breakfast, British, Burger, Cafe, Caribbean, Chinese, Coffee and Tea, Contemporary, Diner, Finger Food, Fish and Chips, French, Greek, Grill, Healthy Food, Indian, International, Italian, Japanese, Kebab, Korean, Latin American, Malaysian, Mediterranean, Mexican, Middle Eastern, Pakistani, Pizza, Portuguese, Sandwich, Seafood, Spanish, Steak, Sushi, Tapas, Thai, Turkish.
7. Restaurant Type must display an option to filter Dine-Out, Take-Away, Delivery, Bar and Drinks & Nightlife.
8. 'Wi-Fi' must have an option of Yes or No.

Phase 3: Admin Functionality to assign role & permission to add/delete restaurants

Business rules for Phase 3

In Phase 3, functionality for the administrator of the website is implemented. It is divided into two parts: Administrator's functionality and Restaurant Configuration.

A. Business rules for Administrator's functionality

1. The administrator functionality must be accessible from 'User' tab.
2. 'User' tab is a link next to 'Language' filter on top-right.
3. 'User' tab must be on left to the 'Language' filter on home page.
4. 'User' tab will be used for both administrator and online community.
5. The administrator functionality will be accessible to Administrator and Staff members only.
6. The administrator functionality page is a separate login page or the login option from the drop down on the home page.
7. The administrator has the rights to add roles and permissions to the staff.

8. The permissions are divided into three categories: ‘Create’, ‘Update’ and ‘Delete’.
9. The roles are divided into two categories: ‘Administrator’ and ‘Staff’.
10. The ‘Administrator’ must have an option to ‘Create’, ‘Modify’ and ‘Delete.
11. The ‘Staff’ must have an option to access permissions or roles assigned by administrator.
12. The ‘Administrator’ and ‘Staff’ must be operate in different sessions.
13. The permissions and roles must be available on the same page for the administrator.
14. The administrator must configure role first then assign the permissions to the role for staff usage.
15. The ‘Administrator’ functionality must have a check to delete or modify other users’ data.
16. The ‘Administrator’ and ‘Staff’ must have access to block or delete users.
17. The ‘Administrator’ and ‘Staff’ must have access to block or delete comments.

B. Business rules for Restaurant Configuration

1. The page must three tabs: ‘Restaurant’ and ‘Menu’.
2. The Restaurant tab must have two action events: ‘Create Restaurant’ or ‘Modify Restaurant’
3. If user selects ‘Create Restaurant’ then the form should be editable with ‘Save’ option at the end of the page.
4. If user selects ‘Modify Restaurant’ then the form should be editable with ‘Save’ option at the end of the page.
5. The ‘Create Restaurant’ form must have following details in the form: 'Restaurant Name', ‘Restaurant Type', 'Address Line 1', 'Address Line 2', ‘Country’, ‘County’, ‘City’, 'Post Code', 'Longitude', 'Latitude', 'Description', 'Phone', ‘Opening Time Hour’, ‘Opening Time Minute’, ‘Closing Time Hour’ and ‘Closing Time minute’.
6. The 'Restaurant Name' must be mandatory alphanumeric input.
7. ‘Restaurant Type' must be dropdown
8. 'Address Line 1' must be mandatory alphanumeric input with space.
9. 'Address Line 2' must be alphanumeric input with space.
10. ‘Country’ must have an option to select one option from list of countries. It is a mandatory input.

11. ‘County’ must be filtered on ‘Country’ input and have an option to select one option from list of counties present in the selected Country. It is a mandatory input.
12. ‘City’ must be filtered by ‘County’ and have an option to select one option from list of cities present in the selected County. It is a mandatory input.
13. ‘Post Code’ must be alphanumeric input and must be validated. It is a mandatory input.
14. ‘Longitude’ and ‘Latitude’ must be calculated on user’s response available from 8-13 points above.
15. ‘Description’ is alphanumeric input with spaces and hyphens. The Description must be translated into other languages.
16. ‘Phone’ is alphanumeric input.
17. ‘Opening Time Hour’ and ‘Closing Time Hour’ must be dropdown having values from 00-23.
18. ‘Opening Time Minute’ and ‘Closing Time minute’ must be dropdown having value from 00-59.
19. ‘Opening Time Hour’ and ‘Opening Time Minute’ must be validated.
20. ‘Closing Time Hour’ and ‘Closing Time minute’ must be validated.
21. The form must be validated for the Restaurant’s location before saving.
22. The save must create an alphanumeric unique identifier for Restaurant.
23. The ‘Modify’ action on ‘Restaurant’ tab must have 'Restaurant Name', ‘Restaurant Type', 'Address Line 1', 'Address Line 2', ‘Country’, ‘County’, ‘City’, 'Post Code', 'Longitude', 'Latitude', 'Description', 'Phone', ‘Opening Time Hour’, ‘Opening Time Minute’, ‘Closing Time Hour’ and ‘Closing Time minute’ as editable option with same rules from 6- 21 above.
24. The ‘Modify’ page must also have an alphanumeric input for ‘Restaurant Unique Id’ which will be generated at the time creating a restaurant.
25. ‘Menu’ tab must have option to add ‘Country’, ‘County’, ‘City’, ‘Restaurant Unique id’, ‘Restaurant Name’, ‘Name’, ‘Price’, ‘Type’, and ‘Category’.
26. The user can filter restaurant on ‘Restaurant Unique id’ or ‘Country’, ‘County’, ‘City’ then selecting from dropdown ‘Restaurant Name’.
27. ‘Name’ must be alphanumeric input for the menu item.
28. ‘Price’ must be decimal input.

29. ‘Type’ must be a dropdown with values ‘Appetizers’, ‘Main-course’, or ‘Dessert’.
30. ‘Category’ must be checkbox with values ‘Breakfast’, ‘Lunch’, ‘Dinner’, ‘Snacks’, ‘Cafe’, ‘Pub’ and ‘Bar’.

Phase 4: Forum for Ratings and Community

Business rules for Phase 4

Phase 4 is an implementation of the Community to login and review/comment the restaurants.

1. The interested user should be registered by using ‘User’ tab available on the top right left to the ‘Language’ selector.
2. The user must be redirected to ‘Sign Up’ page.
3. ‘Sign-Up’ page must have following inputs: ’First Name’, ‘Last Name’, ‘Email’, ‘Password’, ‘Re-Type Password’.
4. ’First Name’ and ‘Last Name’ must alphanumeric textbox.
5. ‘Email’ must be validated.
6. ‘Password’ can have string value.
7. ‘Password’ must be checked for one number and one special character for strong password.
8. User must have access to login from Facebook and Google accounts.
9. In case, user has registered using steps 2-7, the user must be verified by sending an email.
10. The ‘Searched Results’ page must have option to review/ rate each restaurant in the list.
11. The ‘Rating’ must have ten stars.
12. If the user rated the restaurants by selecting the stars, response must be saved without redirecting the page.
13. If the user chose to review the restaurant, the pop-up will appear and response must be saved without redirecting the page.
14. The user must have a separate page.
15. The user must an option to upload the image or inherit the image from Google/Facebook account.
16. If the user removes image, the image must not be inherited from Facebook/Google account.
17. If the user updates the image, the image must not be replaced by Google/Facebook account.
18. The user page must have profile and settings tab.
19. The user profile tab must have ‘Image’, ‘First Name’, ‘Last Name’.
20. The user settings tab must have ‘Email’ and ‘Password’.

21. If the user changes 'Email', validate and re-verify the email.
22. If the user changes password, enable 'Re-Type Password' and validate the Password as in Step 7.
23. The user can follow another user.

D. Unit Test Cases

Test case id	Description	Expected Result	Actual Result	Remarks
1	Search food item	The system should navigate to next page	PASS	
2	Search food item and specify distance. For example: - 0.5 miles	The system should navigate to next page and display the list of restaurants within 0.5 miles	PASS	
3	Change language to French and search food item	The system should navigate to next page and display values in French language	PASS	
4	Change language to French, search food item and specify distance	The system return list of restaurants which has the food item on their menu within specified distance		
5	The slider must be clickable event	The slider should have functionality to move	PASS	
6	The value must be entered in decimal	The textbox should validate the user input	PASS	
7	The slider value must be reflected in textbox	The slider value should be visible in textbox	PASS	
8	Empty search on home page	The searched results page must return list of all restaurants in system	PASS	
9	Search pasta within 5 miles on home page	As per current database, it should return only 'Bella Italia'	PASS	

10	Change language to French and search menu item 'crème'	The system must return 10 restaurants & home page, searched page must be displayed in French language	PASS	
11	The page is on first page, the next page and 2 link must be action event	The page is on first page, the next page and 2 link must be clickable and should display next page	PASS	
12	Start new, search food item on home page	The system must navigate to searched results page	PASS	
13	pagination must not be visible if the list is of size 1	The pagination buttons must not be displayed on bottom of page	PASS	
14	The empty search on searched results page	It should display list of restaurants	PASS	
15	Search pasta within 5 miles on searched results page	As per current database, it should return only 'Bella Italia'	PASS	
16	Click foodie icon	It must navigate to home page	PASS	
17	Repeat steps from 12-16 for French Language	The behavior must be same as in steps 12-16 but the language must be French	PASS	

E. Database Queries

DDLS used for Phase 1

```
CREATE TABLE `menu` (
  `RestId` int(20) NOT NULL,
  `MenuId` int(20) NOT NULL,
  `Name` varchar(400) CHARACTER SET latin1 NOT NULL,
  `Price` decimal(4,2) NOT NULL,
  `Type` varchar(11) CHARACTER SET latin1 DEFAULT NULL,
  `Category` varchar(12) CHARACTER SET latin1 DEFAULT NULL,
  `Name_fr` varchar(400) CHARACTER SET latin1 NOT NULL,
  `Type_fr` varchar(45) CHARACTER SET latin1 DEFAULT NULL,
  `Category_fr` varchar(45) CHARACTER SET latin1 NOT NULL,
  KEY `restId_idx` (`RestId`),
  CONSTRAINT `restId` FOREIGN KEY (`RestId`) REFERENCES `rest`
(`RestId`) ON DELETE NO ACTION ON UPDATE NO ACTION
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `rest` (
  `RestId` int(20) NOT NULL,
  `RestaurantName` varchar(100) CHARACTER SET latin1 DEFAULT NULL,
  `BusinessType` varchar(23) CHARACTER SET latin1 DEFAULT NULL,
  `AddressLine1` varchar(4000) CHARACTER SET latin1 DEFAULT NULL,
  `AddressLine2` varchar(4000) CHARACTER SET latin1 DEFAULT NULL,
  `AddressLine4` varchar(4000) CHARACTER SET latin1 DEFAULT NULL,
  `PostCode` varchar(7) CHARACTER SET latin1 NOT NULL,
  `Longitude` decimal(8,6) NOT NULL,
  `Latitude` decimal(8,6) NOT NULL,
  `Description` varchar(4000) CHARACTER SET latin1 DEFAULT NULL,
  `Description_fr` varchar(4000) CHARACTER SET latin1 DEFAULT NULL,
  `Phone` int(11) DEFAULT NULL,
  PRIMARY KEY (`RestId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

CREATE TABLE `rating` (
  `restid` int(11) NOT NULL,
  `foodrating` int(10) DEFAULT NULL,
  `sanityrating` int(10) DEFAULT NULL,
  `overallrating` int(10) DEFAULT NULL,
  `drinksrating` int(10) DEFAULT NULL,
  `experiencerating` varchar(45) CHARACTER SET latin1 DEFAULT NULL,
  PRIMARY KEY (`restid`),
  CONSTRAINT `restid_2` FOREIGN KEY (`restid`) REFERENCES `rest`
(`RestId`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8

CREATE TABLE `timing` (
  `openingtimemin` int(2) DEFAULT NULL,
  `closingtimemin` int(2) DEFAULT NULL,
  `openingtimehour` int(2) DEFAULT NULL,
  `closingtimehour` int(2) DEFAULT NULL,
  `day` varchar(12) CHARACTER SET latin1 NOT NULL,
  `day_fr` varchar(12) CHARACTER SET latin1 DEFAULT NULL,
  `restid` int(20) NOT NULL,
  PRIMARY KEY (`day`,`restid`),
  KEY `restid_idx` (`restid`),
  CONSTRAINT `restId_1` FOREIGN KEY (`restid`) REFERENCES `rest`
(`RestId`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Bibliography

Axmark, D. & Widenius, M., n.d. *MySQL Introduction.* [Online]
Available at: http://delivery.acm.org/10.1145/330000/328041/a5-axmark.html?ip=31.205.42.149&id=328041&acc=ACTIVE%20SERVICE&key=BF07A2EE685417C5%2E9530128DD756F5CF%2E280D9278F484A59C%2E4D4702B0C3E38B35&CFID=933795310&CFTOKEN=38732455&__acm__=1494444022_e5bbc59290bce6f054e4c4abbc459104#URLTOKEN#

Bootstrap, n.d. *Bootsrap.* [Online]
Available at: http://getbootstrap.com/
[Accessed 2017].

Boyd, S. W. & Keromytis, A. D., 2004. SQLrand: Preventing SQL Injection Attacks. *IEEEXplore.*

Brüstel, J., Preuss, T. & Schwenke, C., 2012. An Architecture for E-Government Social Web Applications. *IEEEXplore,* July.

Data.gov.uk, 2017. *Data.gov.uk.* [Online]
Available at: https://data.gov.uk/

Google, 2017. *Google Maps Embed API.* [Online]
Available at: https://developers.google.com/maps/documentation/embed/guide#directions_mode

Google, 2017. *Places Library.* [Online]
Available at:
https://developers.google.com/maps/documentation/javascript/places#place_searches

Gyrödi, C., Gyrödi, R., Pecherle, G. & Olah, A., 2015. A Comparative Study: MongoDB vs. MySQL. *IEEEXplore.*

Harwani, B., 2015. *Make an E-commerce Site in a Weekend.* s.l.:Apress.

IBM, 2017. *Data service layer.* [Online]
Available at:
https://www.ibm.com/support/knowledgecenter/en/SSZLC2_7.0.0/com.ibm.commerce.developer.soa.doc/concepts/csddsl.htm

Jason, 2009. *How to Paginate Data With PHP.* [Online]
Available at: https://code.tutsplus.com/tutorials/how-to-paginate-data-with-php--net-2928

JQuery, 2017. *JQuery.* [Online]
Available at: https://jquery.com/
[Accessed 2017].

Kolšek, M. & Security, A., 2002. *Session Fixation Vulnerability in Web-based Applications.* [Online]
Available at: http://www.acros.si/papers/session_fixation.pdf

php, 2017. *php.* [Online]
Available at: http://php.net/manual/en/intro-whatis.php

Stewart, H., 2005. *The Differences between Locales and Languages.* [Online]
Available at: https://blogs.msdn.microsoft.com/heaths/2005/02/17/the-differences-between-locales-and-languages/

tutorialspoint, 2017. *Design Pattern - Factory Pattern.* [Online]
Available at: https://www.tutorialspoint.com/design_pattern/factory_pattern.htm

W3C, 2013. *Geolocation API Specification.* [Online]
Available at: https://www.w3.org/TR/2013/REC-geolocation-API-20131024/

w3schools, 2017. *HTML Unicode (UTF-8) Reference.* [Online]
Available at: https://www.w3schools.com/charsets/ref_html_utf8.asp
w3Schools, 2017. *HTML5 Geolocation.* [Online]
Available at: https://www.w3schools.com/html/html5_geolocation.asp
w3schools, 2017. *SQL Tutorial.* [Online]
Available at: https://www.w3schools.com/sql/DEfaULT.asP