MULTIPLE RESTAURANT FOOD ORDERING & DELIVERY MANAGEMENT SYSTEM FOR DRAGOON DIGITAL ART LAIR

R. P. L. Fernando

BIT Registration Number: R142003

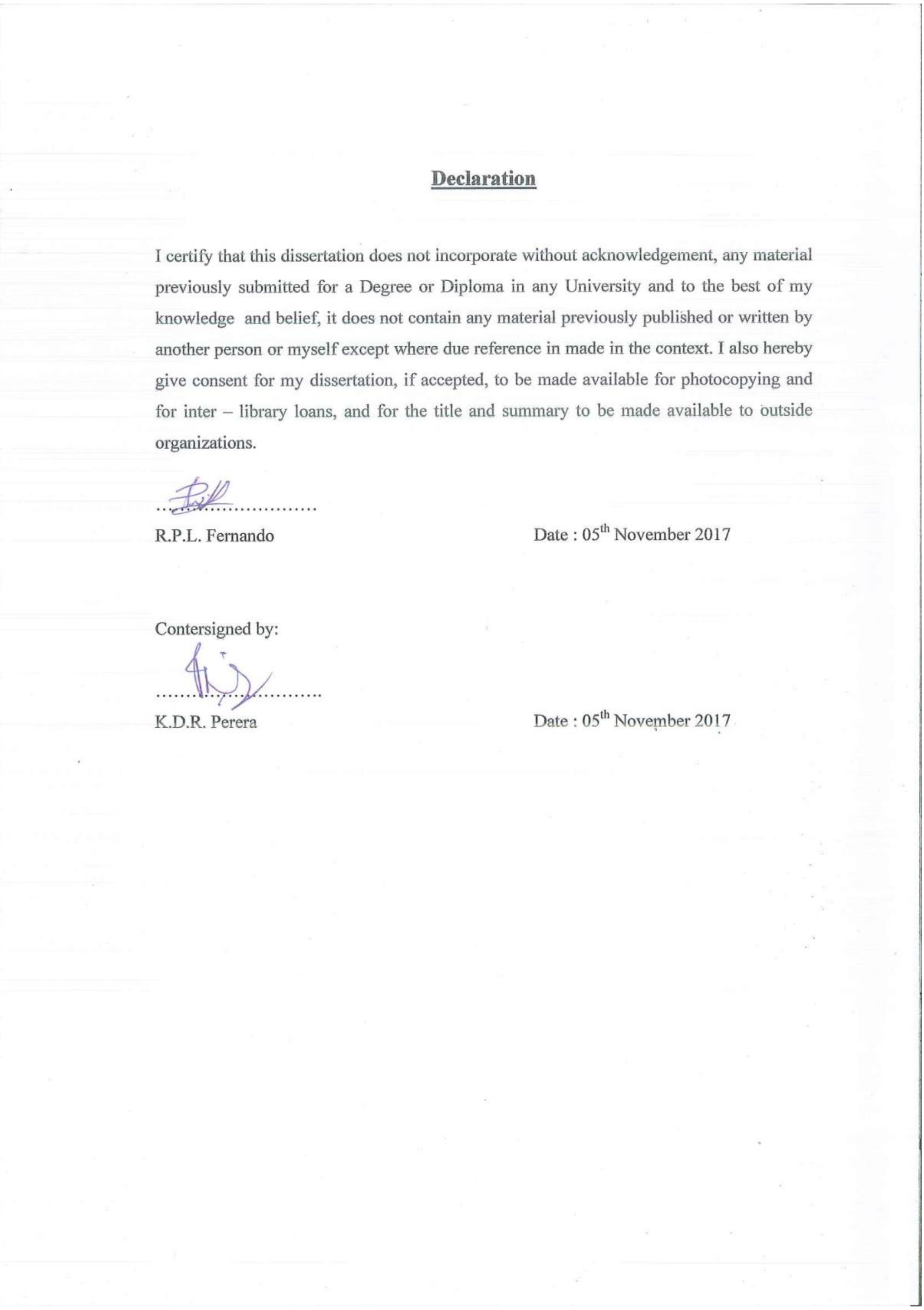
Index Number: 1420038

Client Name: Dragoon Digital Art Lair

Supervisor: Mr. K.D.R. Perera

November 2017

This dissertation is submitted in partial fulfilment of the requirement of the Degree of Bachelor of Information Technology (external) of the University of Colombo School of Computing



Declaration

I certify that this dissertation does not incorporate without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference in made in the context. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for inter – library loans, and for the title and summary to be made available to outside organizations.

……………………………

R.P.L. Fernando Date : 05th November 2017

Contersigned by:

……………………………

K.D.R. Perera Date : 05th November 2017

Abstract

Dragoon Digital Art Lair is a branding company that supports growth of small and medium sized businesses. And they have identified that there is a competitive advantage to restaurants with home delivery service compared to those without. These small and medium sized restaurants cannot afford a delivery fleet because of the large capital involved. Therefore the client of this project is developing a delivery service infrastructure to support those restaurants.

As the current available delivery management systems are targeted at single restaurants, they wanted to develop a custom software system. The system is planned to assist in the complete process from managing the infrastructure to the ordering and delivery process to the end customer.

In order to develop this ordering and delivery management system, Rational Unified Process (RUP) was chosen as the process model as it supports the unclear requirements present in the system as well as Object Oriented Development (OOD). As this system is an online web based system, PHP was chosen as the language along with a MySQL database system. The system is planned to be run on Linux server with Apache web server software. Development tools such as Sublime Text, NetBeans IDE, WAMP software and MySQLWorkbench was chosen for the project implementation.

The final objective of the system is to provide an interface between the restaurants and end customers while supporting the management and administration of the whole infrastructure.

Acknowledgement

It has been a wonderful experience to complete this project and many people supported me in making this project a success. First of all, I would like to express my heartfelt gratitude to Mr. K.D.R. Perara, who provided valuable insights and support as the supervisor, which helped greatly in successfully completing this project.

I would also like to thank the staff and the management of OpenArc School of Business & Technology, for their help and encouragements. Also, I express my appreciation for the BIT coordinator and all the staff at the USCS BIT programme which in many ways provided us with the means and the support to complete this project.

Furthermore, I would like to express my heartfelt gratitude to my parents and my brother, who supported me all through my education and paved the way to complete this project with lot of encouragement and support throughout the project development.

Finally, I would like to express my gratitude for everyone including my friends, who helped and encouraged me on the way.

Table of Contents

List of Figures

List of Tables

List of Acronyms

OOD - Object Oriented Design

UML – Universal Modeling Language

IDE – Integrated Development Environment

SQL – Structured Query Language

HTML – Hyper Text Markup Language

WAMP – Windows Apache MySQL PHP

RUP – Rational Unified Process

PHP – PHP Hypertext Preprocessor

SDLC – Software Development Life Cycle

FDO – Front Desk Officer

CHAPTER 01 – INTRODUCTION

1.1. Introduction

The client of this project is trying to help the small and medium sized enterprises to break through to the next level of their businesses. They currently does this through supporting the businesses with marketing and promotional campaigns.

They Identified that most restaurant owners suffer from lack of resources to manage their own fleet of delivery personnel to do home deliveries. This has led to the proposal to create a delivery infrastructure that restaurant owners can make use of by paying an affordable fee. Thus, removing the barrier of high capital required to start home delivery from the restaurant owners side.

1.2. Motivation of the Project

As this proposed infrastructure spans across restaurants, customers, delivery personnel and the administrative staff of the client; manual system is not an ideal solution. With the current level of accessibility to internet among the people of Sri Lanka, an online web based system can help the client as described below.

- A web based system can reach more customers.
- When the infrastructure grows, a manual system will not be efficient.
- Most customers prefer to order via an online web site because they can clearly see the products and take their time to choose what they want.
- The system will make it easy to produce reports and analyze trends.
- The restaurant owners will find it easier to manage their details in the system through an online web page rather than communicating back and forth through a client support officer.
- The client can manage the infrastructure with minimum number of staff using the proposed system as the most of the work will be handled by the system.

The above points encouraged the decision to develop a web based system to manage the ordering and delivery management infrastructure as it will help the client to minimize their costs while increasing their efficiency and customer satisfaction of both restaurant owners and of customers who make orders through the system.

1.3. Scope of the Project

Identifying the scope of the system was important to prevent scope creep and to ensure that the finished system does satisfy all the needed functions properly. Given below is the identified scope of the system.

- Customer facing website with restaurant and their menu browsing, searching, user accounts, ordering and reviews section
- Restaurant facing web portal with order management, restaurant data management and menu updating.
- Portal for delivery staff to manage assigned orders, view pickup and drop-off points.
- Admin panel for the staff to manage all orders, dispatch orders, manage system data, manage restaurants.
- Accounts handling to manage customer payments and payments to restaurants.
- Email notification system will be implemented to message customers and restaurant owners
- Reports will be generated to assist managers and restaurant owners
- Securing the system with authorization and authentication.

1.4. Objectives of the System

The primary objective of the system is to provide an interface between restaurants and customers while providing support for the management of the whole infrastructure. Below are the objectives of the system.

- Provide a web based portal (website) for customers to easily discover restaurants and browse their menus and place orders
- Provide a web based portal for registered restaurants to access their administration portal to modify data and view reports and to view pending orders
- Provide a web portal for delivery staff to view assigned orders, pickups and drop off points

- Implement an Admin panel to manage system related functions, data and to make manual orders received via offline channels and to maintain the website data
- Maintain a database to store related data
- Provide an easy to way for customers to place orders online or offline (via calls)
- Provide ability to support multiple restaurants
- Enable registered restaurants to manage their data such as menus easily and view information about orders
- Improve customer engagement through user-friendly web interface
- Provide user account creation facility for customers to easily access previous orders and addresses, etc. for quick ordering.
- Provide an enhanced user experience through email notifications on status updates on customer orders

1.5. Structure of the Dissertation

The dissertation discusses the analysis, design and development of the Food Ordering and Delivery Management System and consists of five chapters as given in the below list.

1.5.1. Chapter 02 – Analysis

This chapter discusses the requirement gathering methods that were used and the functional and non-functional requirements of the system. A review of the existing systems is also carried out in this chapter.

1.5.2. Chapter 03 – Design

This chapter is focused on the design phases of the system including the appropriate diagrams of the system and of the database structure. The user interfaces design is also discussed in this chapter.

1.5.3. Chapter 04 – Implementation

This chapter discusses the development process of the system including the development environment and tools. A review of the codes used in the system is also carried out in the chapter.

1.5.4. Chapter 05 - Evaluation

The testing methods used to test the system as well as the test plans, test cases and tools along with their results are discussed in this chapter.

1.5.5. Chapter 06 - Conclusion

This chapter discusses the future enhancements of the system as well as lessons learnt from the project.

References

Includes sources such as books, documents, websites that were used to gather knowledge to develop the system as well as to write the dissertation.

CHAPTER 02 – ANALYSIS

2.1. Introduction

System analysis is a very important part of developing a software system. The primary goals of the analysis is to identify the requirements of the client clearly and to study and understand any existing systems. In this chapter, the requirement gathering techniques, analysis of the gathered requirements along with functional and non-functional requirements, and comparison of alternative solutions are discussed.

2.2. Fact Gathering Techniques

The first part of the analysis is to gather the requirements and facts about the software system. There are multiple ways to gather facts, but in this project, the following techniques were primarily used.

- Interviews
- Analysis of existing similar systems
- Documentation

Interviews were conducted with the client of the project to get a clear picture of the requirements from their end. Feedbacks and questions helped to verify the gathered facts as well uncover new facts. Some of the existing systems that follow a similar model was analyzed to identify possible uncovered requirements and facts along with important parts of the systems and user interaction patterns.

Documenting the gathered facts and requirements also helped to identify and clarify the requirements further. Design diagrams and documents were helpful in this method.

2.4. Current System

There is no current system in place as the whole infrastructure depends on the proposed software system.

2.3. Proposed System

After gathering the requirements and identifying the main points of importance, analysis of the requirements was started. Diagrams and documentation was used to analyze the requirements and to present the requirements in an understandable manner.

The following high level Use Case Diagram from *Figure 2.1* shows the main use cases and the following actors.

1. Administrator
2. Manager
3. Front Desk Officer
4. Restaurant Owner
5. Customer
6. Delivery Personnel (Couriers)

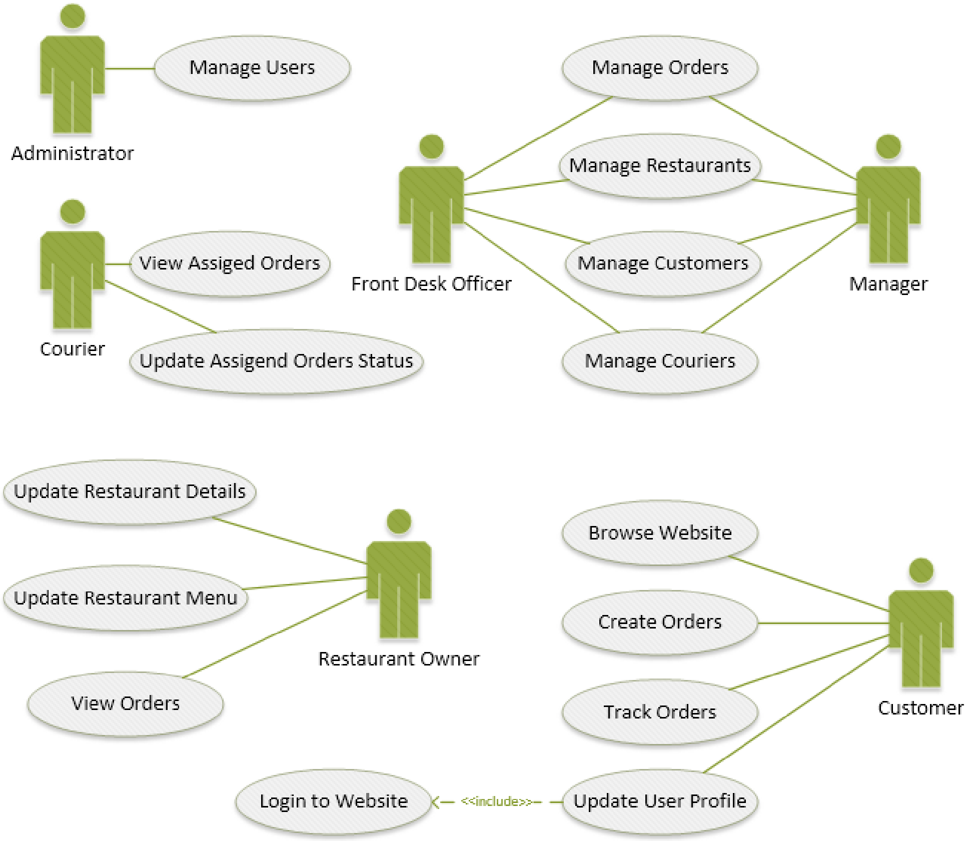


Figure 2.1 - High Level Use Case Diagram for the system

The system is modularized into several modules. The modules are as follows,

1. User Management Module
2. Order Management Module
3. Restaurant Management Module
4. Customer Management Module
5. Web Site Content Management Module
6. Website module
7. Courier Management Module
8. Report Generation Module
9. Reference Data Management Module

2.4. Functional Requirements

Functional Requirements "are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations." [1] Functional requirements for each module is given below.

2.4.1. User Management Module

- The Administrator should be able to create, update and remove users and assign user roles.
- Also, he should be able to create and update User Roles and their permissions.
- Only users with Administrator role should get access to this module.

2.4.2. Order Management Module

- Front Desk Officer and Manager should be able to access this module.
- User should be able create, update and remove orders and view orders
- User should be able to assign a courier to an order
- User should be able to view the related Customer and Restaurant's details.
- User should be able to filter and search orders

2.4.3. Restaurant Management Module

- Front Desk Officer and Manager should be able to access this module.
- User should be able view, create, update and remove restaurants and menus.
- User should be able to view and update the restaurant's menu
- User should be able to filter and search restaurants.

2.4.4. Customer Management Module

- Front Desk Officer and Manager should be able to access this module.
- User should be able to create, update and remove customers
- User should be able to view each customer's order history

2.4.5. Web Site Content Management Module

- Front Desk Officer and Manager should be able to access this module.
- Updatable content in the website includes offers, banners, top categories, etc.

2.4.6. Website Module

- Visitors can browse the website to find their favorite restaurants and their menu
- Visitors should be able to add items to the shopping cart and complete order
- Visitors can create an account by signing up in the website
- Signed in visitors can track orders, save orders and view their order history
- Visitors without an account has to provide all necessary details before each order.

2.4.7. Courier Management Module

- Front Desk Officer and Manager should be able to access this module.
- Courier details should be able to be created, updated and removed.
- User should be able to view the delivery history of the couriers.

2.4.8. Report Generation Module

- Only the manager should be able to access this module
- Manager should be able to view and customize the period of the reports

- Reports should provide insights with relevant charts and tables.

2.4.9. Reference Data Management Module

- All the reference data used in other modules are managed by this module
- Reference data include restaurant categories, cities, etc.
- Front Desk Officer and Manager should be able to view, create, update and remove reference data items.

2.5. Non-Functional Requirements

Non-functional requirements “are requirements that are not directly concerned with the specific services delivered by the system to its users.” These represent the characteristics that the system should have such as performance, security, or availability. [1]

2.5.1. Security

Users should only be able to access the modules that are permitted by their user roles. An Authentication and authorizations system should be included to enforce this requirement.

2.5.2. Availability

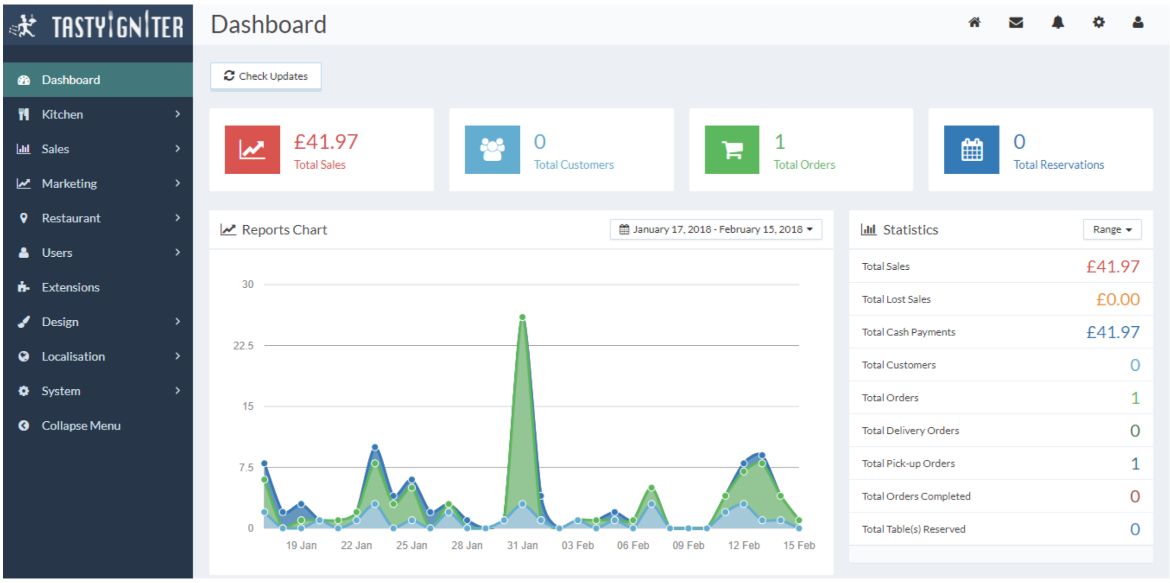
The system should be robust to handle exceptions and continue operating normally. Specially the availability of the website to the customers is very important.

2.5.3. Usability

The system should be easy to use and actions should not be confusing. Every action should reflect the changes that would happen and provide proper feedback.

2.6. Existing Similar Systems

There are existing systems such as TastyIgniter [2] (figure 2.2) and Orderlord [3] (figure 2.3) available. But they are focused on managing a single restaurant’s delivery process. As this system has to handle multiple restaurant and provide a common portal for them all, such a product is not suitable for the client. As a result this system was decided to develop entirely targeted at the requirements mentioned above, to support multiple restaurants.



TASTYIGNITER

Dashboard

- Dashboard
- Kitchen
- Sales
- Marketing
- Restaurant
- Users
- Extensions
- Design
- Localisation
- System
- Collapse Menu

Check Updates

£41.97 Total Sales

0 Total Customers

1 Total Orders

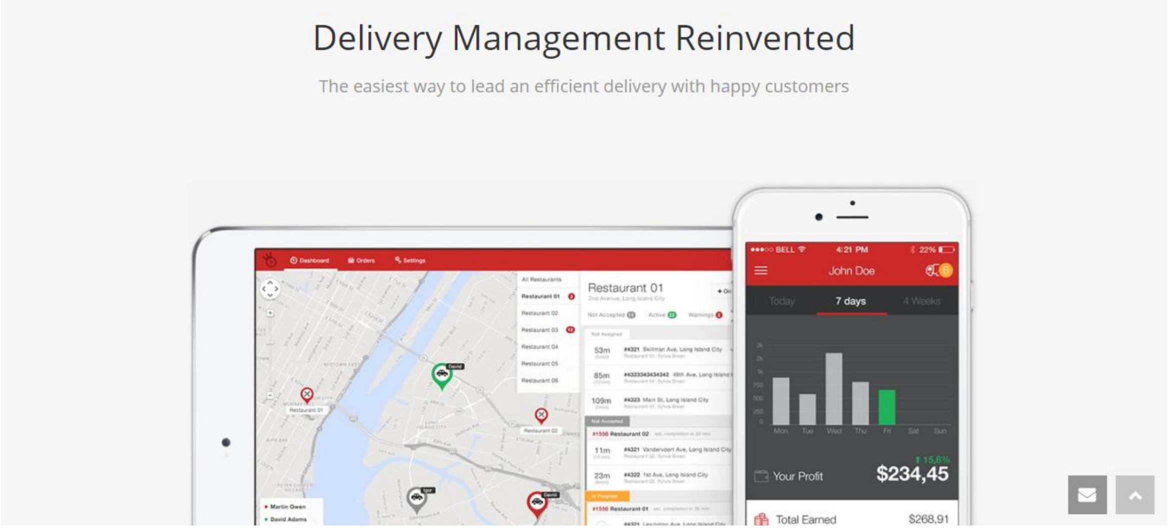
0 Total Reservations

Reports Chart — January 17, 2018 - February 15, 2018

Statistics — Range

Total Sales	£41.97
Total Lost Sales	£0.00
Total Cash Payments	£41.97
Total Customers	0
Total Orders	1
Total Delivery Orders	0
Total Pick-up Orders	1
Total Orders Completed	0
Total Table(s) Reserved	0

Figure 2.2 – TastyIgniter



Delivery Management Reinvented
The easiest way to lead an efficient delivery with happy customers
Restaurant 01
John Doe
Today
7 days
4 Weeks
Your Profit
$234,45
Total Earned
$268,91

Figure 2.3 – OrderLord

CHAPTER 03 – DESIGN

3.1. Introduction

It is very important to have a good design before starting the implementation of any software project. It helps the developers and also the designers to understand how the system should be organized, how it should be modularized and how the data should be stored. There are many system design practices such as Object Oriented Design used by software engineers. And it is important to select the right method for the design.

3.2. Software Development Process Models

A software process model is simplified representation of a software development approach. And these process models integrate different processes, tools and techniques depending on the model. The selection of a process model depends largely on the requirements of the system and the environment of the system. [1]

There are many process models that have been developed for different use cases. The list below includes some of the process models that were considered for this project.

- **Waterfall model** – This method is used when the requirements are solid and clearly understood. Mostly used for computerizing existing manual systems. This is not a good methodology for this project as the requirements are not as solid and clearly defined.
- **Prototyping** – As the name suggests, prototypes of the system are created and presented to the user, and the next prototype is improved upon this feedback. As it requires a lot of time and effort, this method is not a good fit for this project.
- **Rational Unified Process (RUP)** – This is a iterative and incremental development method. RUP allows the users to provide feedback and then iteratively improve the system. Also, RUP supports Object Oriented Development.

After considering the above mentioned methodologies, RUP was chosen as it supports Object Oriented Development As the requirements are not solid and not clearly defined, this will help to make sure that the final system satisfies all the necessary client requirements sufficiently. Also as the system is divided into modules it is easier to adapt the iterative and incremental development model within the project.

Given below are the four phases of the RUP process. [1][4]

1. **Inception** – The goal of the inception phase is to establish a business case for the system.
2. **Elaboration** – Elaboration phase involves to develop an understanding of the problem domain and establish an architectural framework.
3. **Construction** – The construction phase involves system design, programming, and testing.
4. **Transition** – Concerned with moving the system from the development community to the user community and making it work in a real environment.

3.3. Object Oriented Design (OOD)

From the available design methods, Object oriented design provides a natural way of dealing with the mapping between real world entities into the software system, while promoting reusable software development. Unified Modeling Language (UML) is used to define diagrams used in OOD. There are many OOD UML diagrams that can be used to model the system. Some of them are given below.

- Use Case Diagrams
- Entity Relationship Diagrams
- Sequence Diagrams
- Activity Diagrams

3.4. Food Ordering and Delivery Management System Design

The below sections discuss the architecture of the system using the UML diagrams used in Object Oriented Design.

3.4.1. High Level Use Case Diagrams

The high level overall use case diagram was given in the *figure 2.1* and the below use case diagrams from *figure 3.1* to *figure 3.5* each describe the high level use cases of each actor of the system.

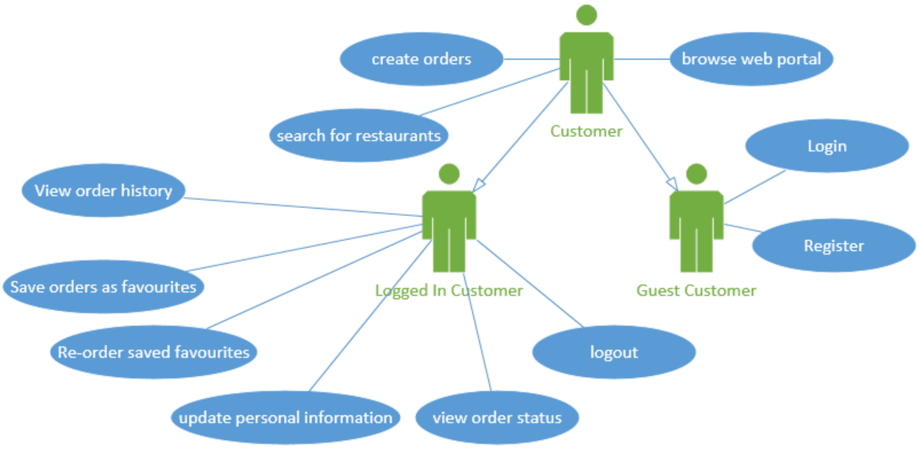
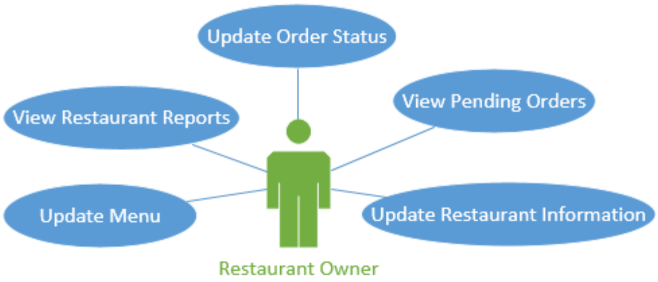


Figure 3.1 – Use case diagram for the Customer



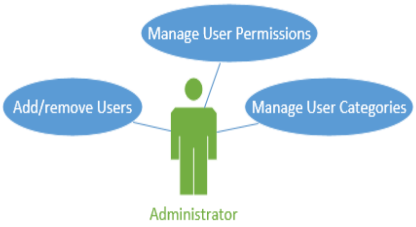


Figure 3.2 – Use case diagram for Restaurant Owner

Figure 3.3 – Use case diagram for the Administrator

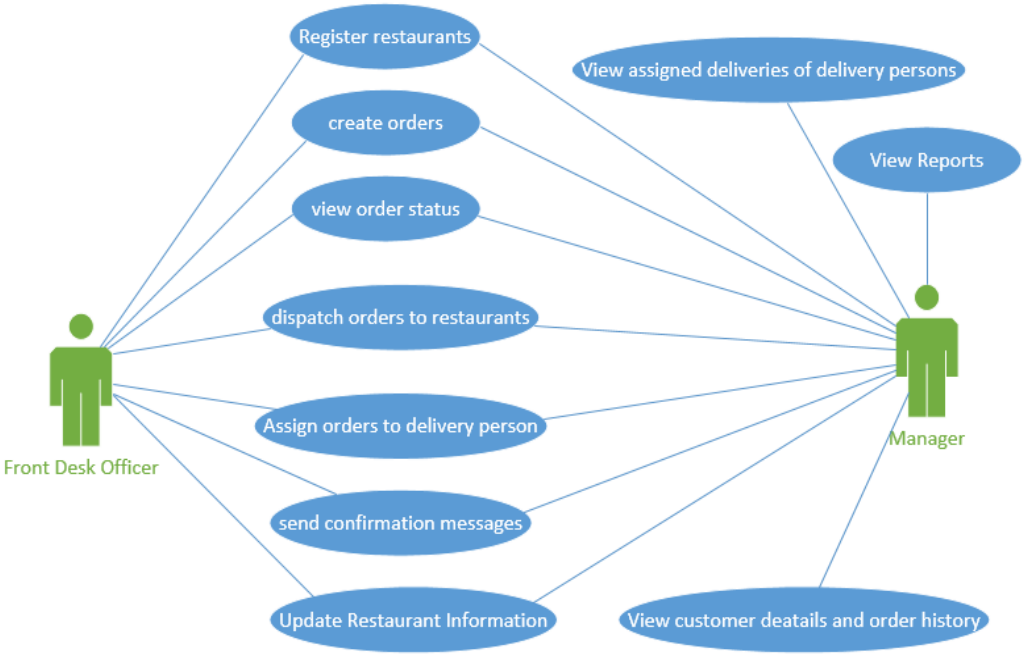

Register restaurants
View assigned deliveries of delivery persons
create orders
View Reports
view order status
dispatch orders to restaurants
Front Desk Officer
Assign orders to delivery person
Manager
send confirmation messages
Update Restaurant Information
View customer deatails and order history

Figure 3.4 – Use case diagram for the Front desk Officer and Manager

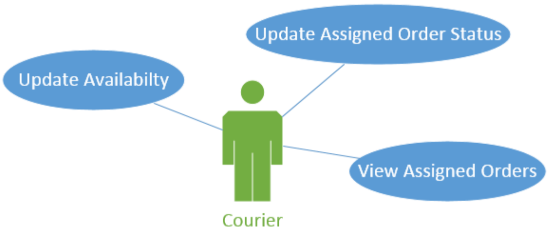


Figure 3.5 – Use case Diagram for Courier

3.4.2. Database Design

Database design deals with creating a data schema for the data to be stored with the system. The database design plays a very significant role in the system design, as a bad data model can make the code complex and error prone because of complex and non-optimal queries. Reducing the redundancy of the database using Normalization is very important part of the database design. Given below is a brief description of the normalization levels.

1. **First Normal Form (1NF)** – Removing of duplicating groups and non-atomic column, and defining a primary key for the table.

2. **Second Normal Form (2NF)** – This level removes partial dependencies by using separate tables and foreign keys.
3. **Third Normal Form (3NF)** – This level is to remove transitive dependencies.

Given below is the Entity Relationship Diagram for the system’s database.

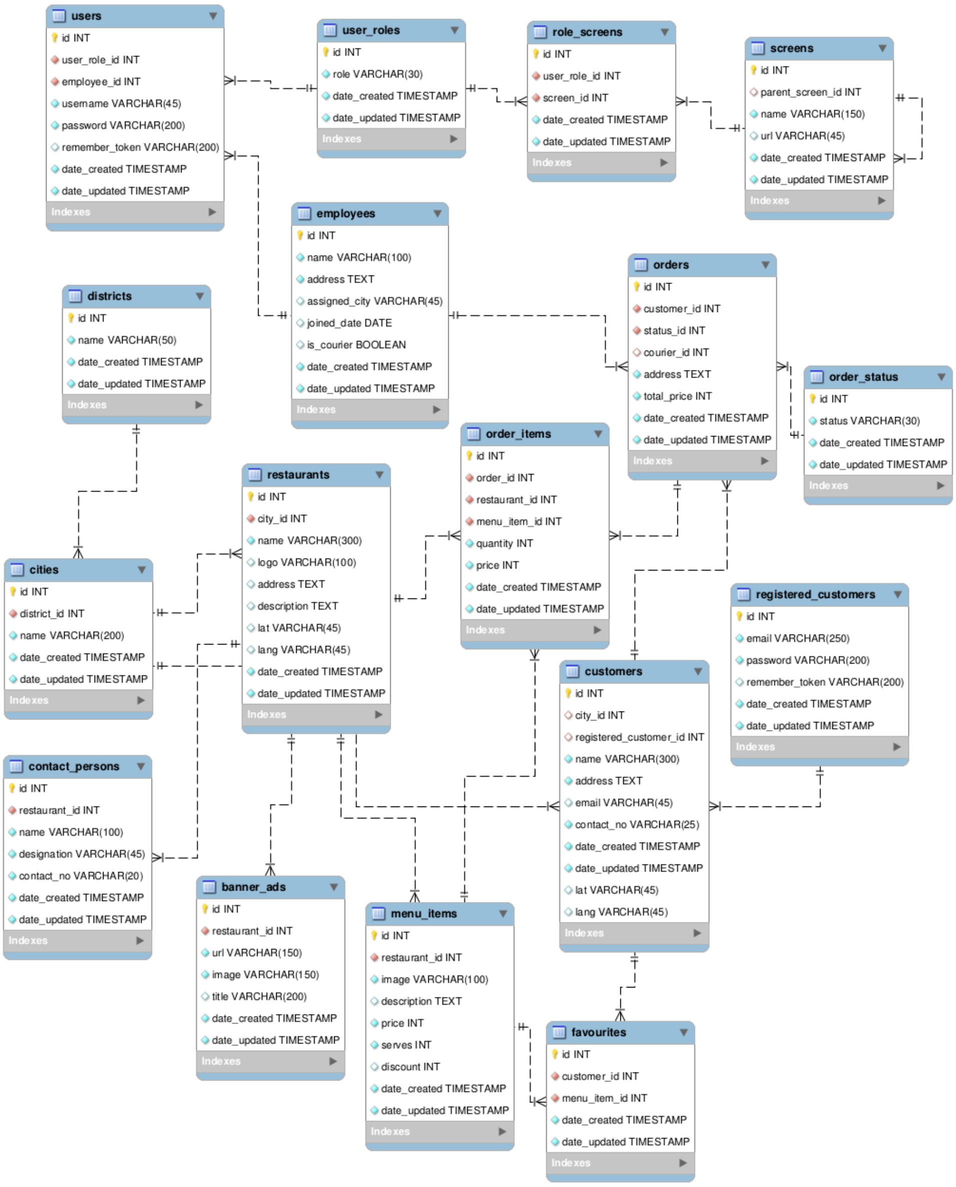


Figure 3.6 – Entity Relationship Diagram

3.4.3. Class Diagram of the System

Below figure 3.7 represents the class diagram of the system.

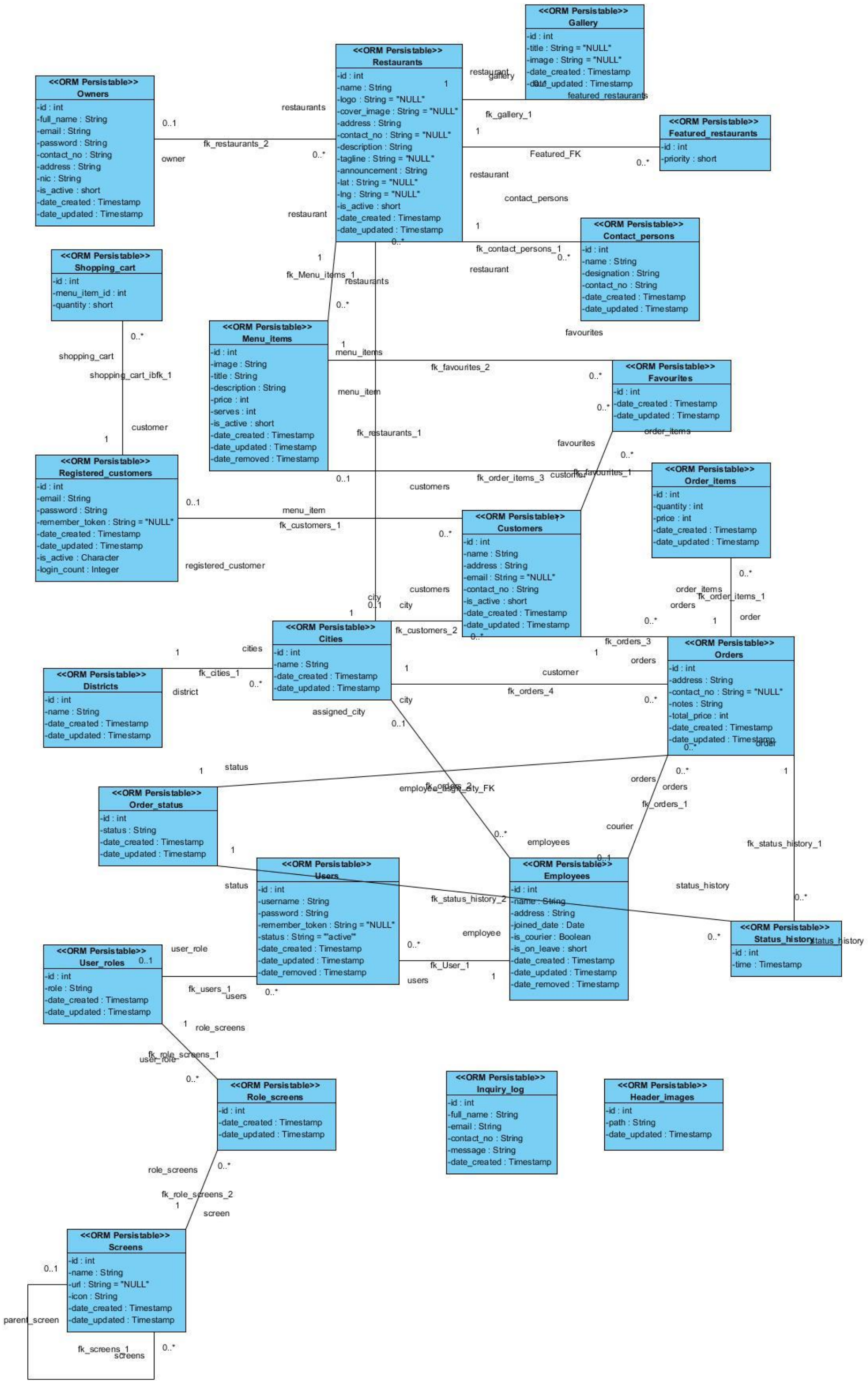


Figure 3.7 – Class Diagram

3.4.4. Activity Diagram for Order Placement Process

The below activity diagram provided in figure 3.7 for the order placing process within the system.

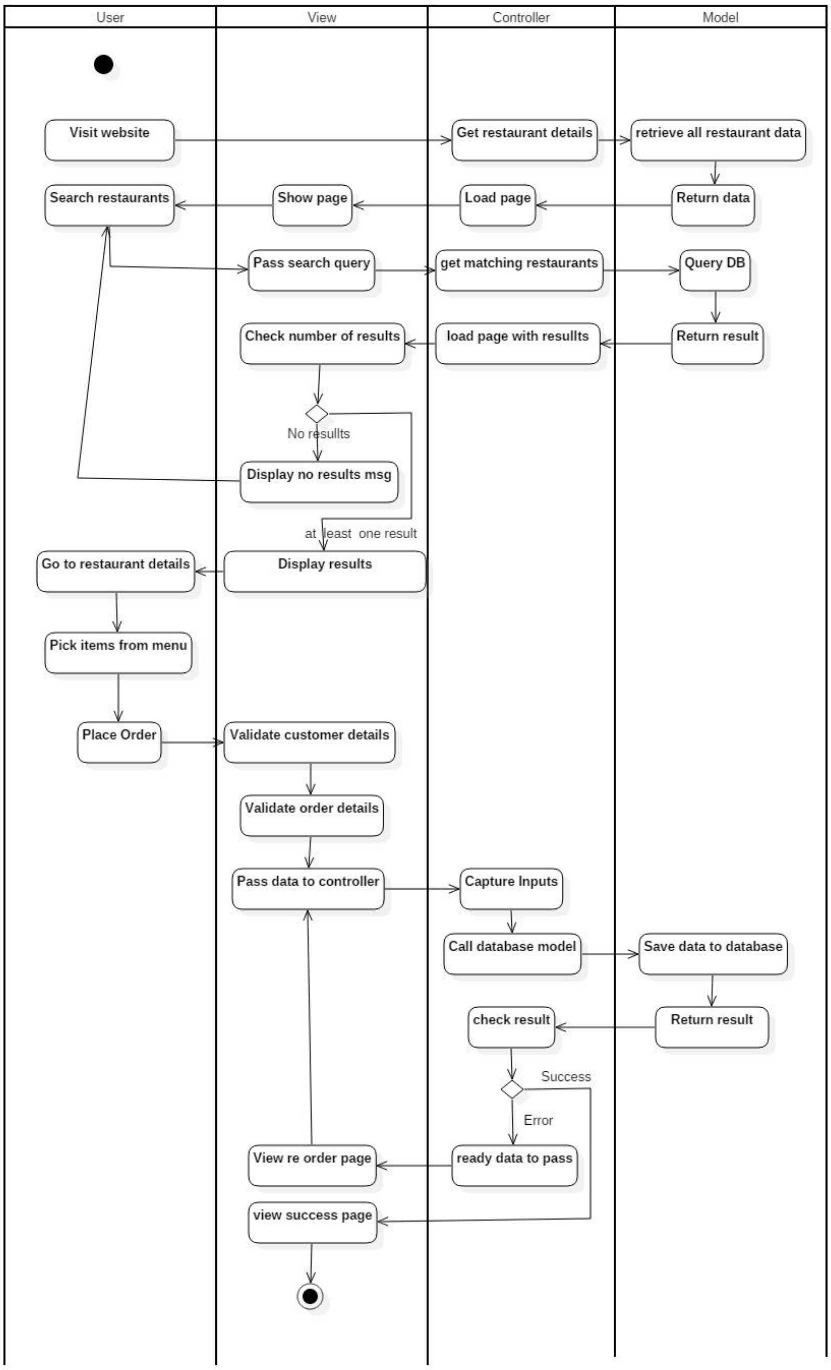


Figure 3.8 – Activity Diagram for Order Placing Process

3.4.5. Sequence Diagram for Order Placement Process

The below sequence diagram provided in figure 3.8 for the order placing process within the system.

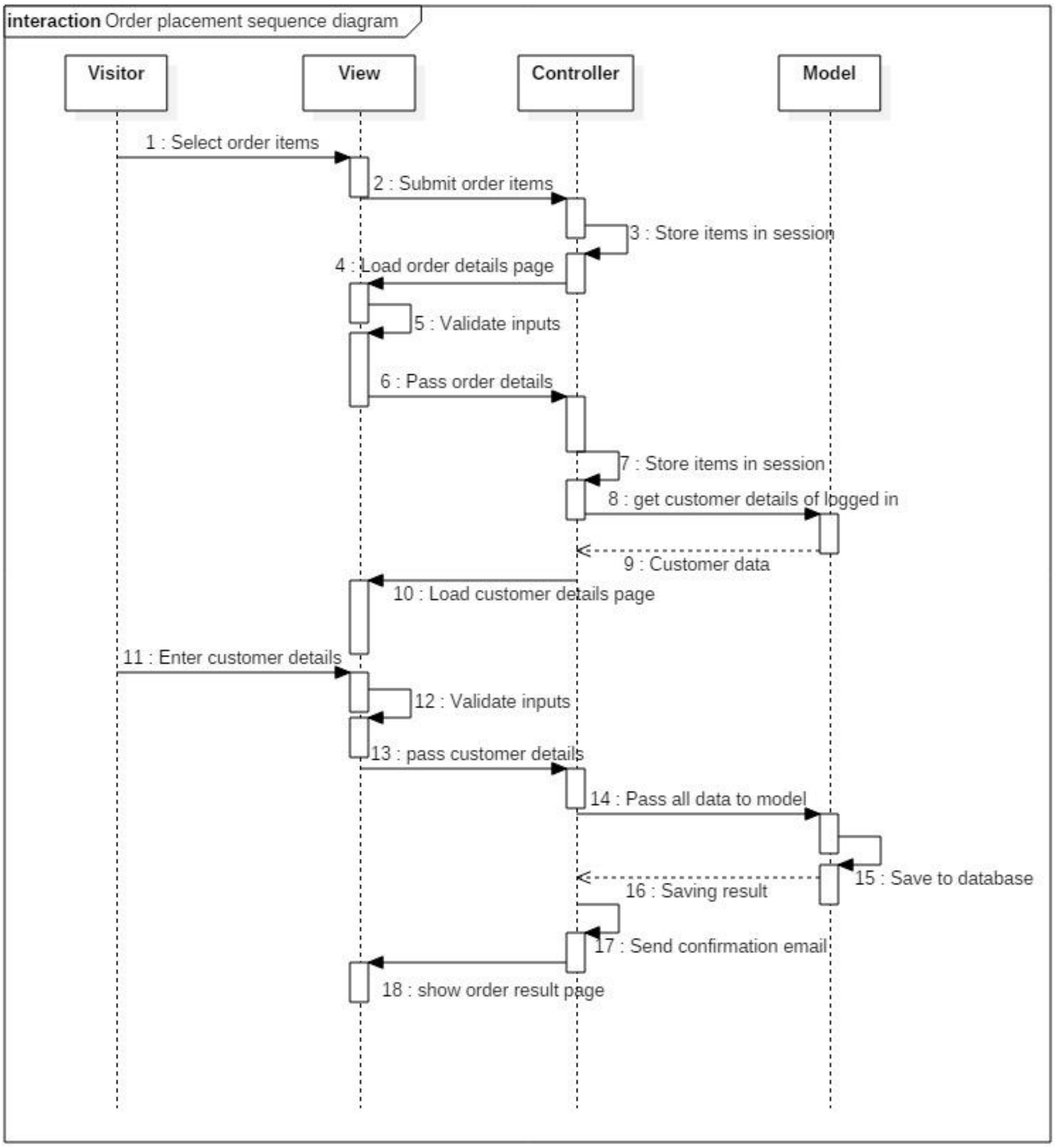

interaction Order placement sequence diagram
Visitor
View
Controller
Model
1 : Select order items
2 : Submit order items
3 : Store items in session
4 : Load order details page
5 : Validate inputs
6 : Pass order details
7 : Store items in session
8 : get customer details of logged in
9 : Customer data
10 : Load customer details page
11 : Enter customer details
12 : Validate inputs
13 : pass customer details
14 : Pass all data to model
15 : Save to database
16 : Saving result
17 : Send confirmation email
18 : show order result page

Figure 3.9 – Sequence Diagram for the Order Placing Process

3.4.6. User Interface Design

As the user interface is what a user interacts with, it is very important to pay close attention when designing the user interface. A bad user interface can cripple a functionally superior system in the point of view of a common user. So, it is very important to design the user interface with the end user in mind along with their mental and physical capabilities and skills. Given below are some of the user interface mockups for the system.

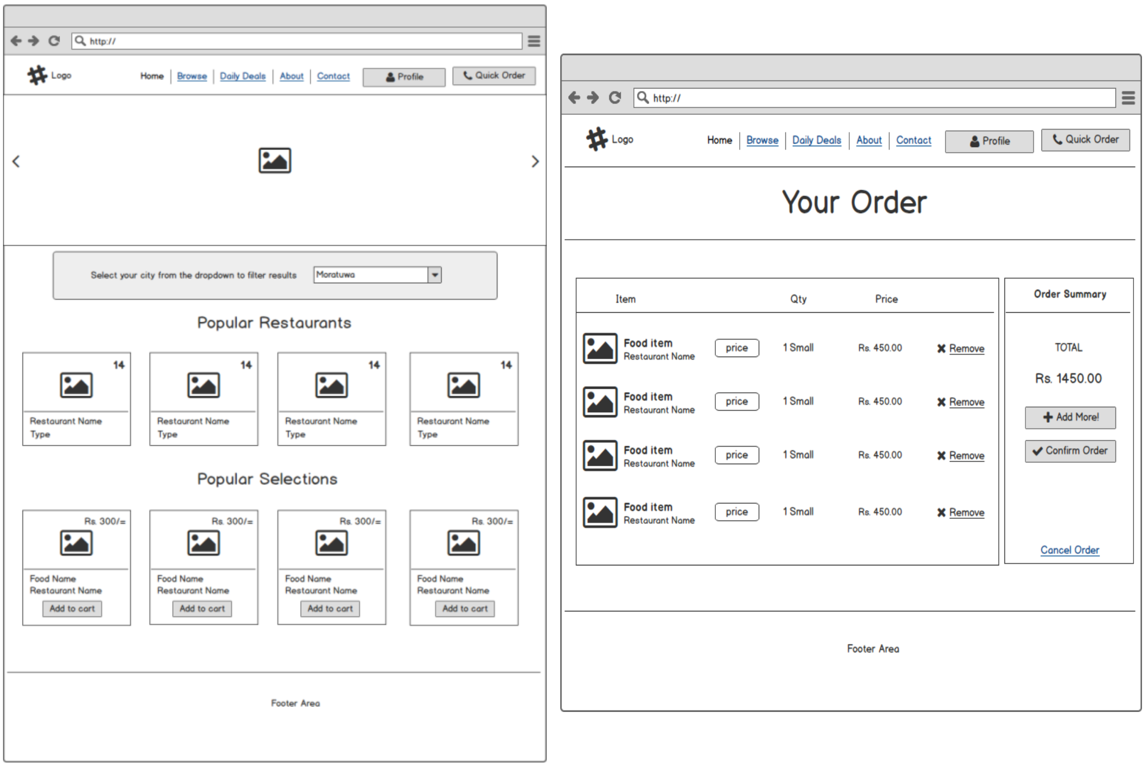


Figure 3.10 – Wireframes of the Food Ordering Website

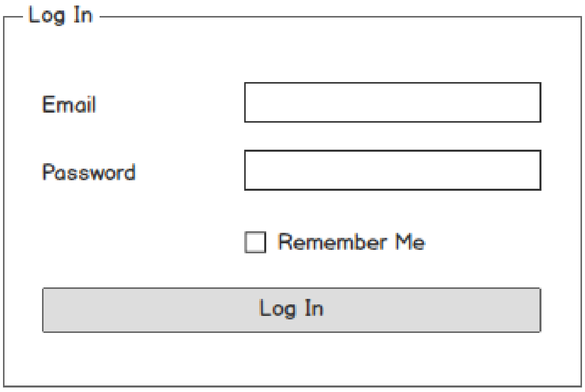


Figure 3.11 – Login Form for the Back End of the System

CHAPTER 04 – IMPLEMENTATION

4.1. Introduction

Implementation phase is where the designed system starts get its final form. Implementation phase is a very important step in the life cycle of the system as this is where the designed solution is transformed into an usable working software system.

Coding is a time consuming task, but with a good design the developer can easily write the software as required without getting into too many issues. Some other important things which has to be considered when coding are readability, maintainability, input validation and commenting.

This chapter discusses the implementation environment, languages and tools used, structure of the codebase and the flow of the system.

4.2. Hardware and Software Environment

4.2.1. Hardware Environment

- Intel Core i3 Processor 1.70Ghz
- 4GB RAM
- 500GB Hard Disk Drive
- Printer
- Modem/Router

4.2.2. Software Environment

- Microsoft Windows 8.1
- Apache Server 2.4.9
- MySQL 5.6.17
- PHP 5.5.12
- Google Chrome Web Browser

4.3. Development Tools and Technologies

The following tools and technologies were used to develop the system.

- HTML

  HTML allows the definition of document structure in web pages.
- CSS

  Web pages are styled using CSS (Cascading Style Sheets)
- JavaScript

  Scripting language used with HTML to provide dynamic behavior to web pages
- PHP

  Server side language used to handle requests, system logic and database access
- CodeIgniter

  A PHP framework that provides a solid architecture to support the system
- MySQL

  A freely available popular database management system. System’s database is managed by MySQL
- JQuery

  A JavaScript library which provides shorter and easier functions for common uses
- Bootstrap Framework

  A CSS framework that provides styling for HTML elements
- Sublime Text Editor

  A text editor that have basic IDE like functionality
- MySQL Workbench

  A graphical client to interact with MySQL database system
- Microsoft Visio

  A diagraming and modeling software that supports UML
- Adobe Photoshop

  Feature rich photo editor for image manipulation needs

4.4. System Architecture

The system was designed with MVC architecture in mind. MVC (Model-View-Controller) architecture is popular in web based systems as it provides clear cut separation between client and server side code. The concept of a Model also separates data access from business logic which is handled by the Controller. View acts as the interface between the system and the user.

Below diagram shows how the data is passed from user to database and vice versa.

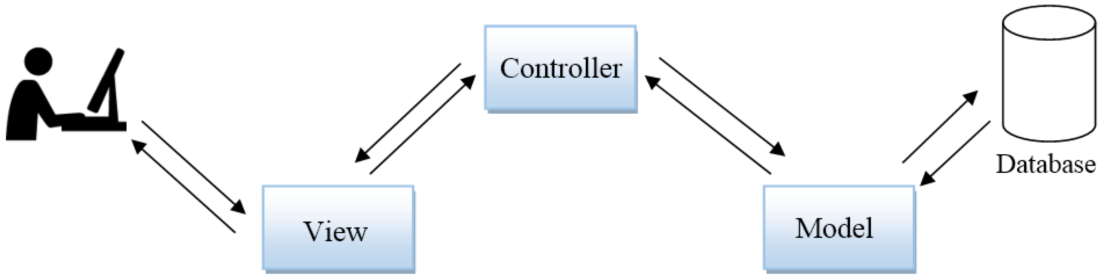


Figure 4.1 – MVC Architecture

4.5. Code Analysis

4.5.1. System login

The below code is taken from Login Controller. This function handles the login request and the database call to validate the input data, and redirects the user accordingly. And if the login is a success, the user’s name and ID will be saved to the session. These variables are later used to validate login again when accessing pages.

```
/**
 * Data submitted from Login form is captured here and
 * validate against the DB records to redirect the user
 * according to the result
 *
 * @return null
 */
public function validate_login()
{
      // capture the POST variables passed from the form
      $username = $this->input->post("username");
      $password = $this->input->post("password");

      // loads the models that deals with users table
      $this->load->model("Users_Model");

      // call validate function in model and store the returned
      // value in $result variable
      // return value is either TRUE or FALSE
      $result = $this->Users_Model->validate_login($username,
                  $password);

      if ($result) {     // if the user's login details are valid
            // set susername and users's ID in session
            $data = array(
                        'username' => $username,
                        'user_id' => $result[0]->id
                    );
            $this->session->set_userdata($data);

            // forward to dashboard
            redirect("Welcome");
      } else { // if login details are invalid
            $msg = 'Invalid username and/or password.';
            // reload the login page with the error message
            $this->index($msg);
      }
}
```

When the model is called in the above function, below function is invoked. It will query the database with the given parameters and check them against the given values to return a TRUE or FALSE result back to the function caller.

```
/**
 * validate the login by querying the database with given parameters
 * and checking the count of the result's rows
 *
 * @param {string} username
 * @param {string} password
 * @return {boolean} 'true' if credentials are valid, if not 'false'
 */
public function validate_login($username, $password)
{
      // retrieve the record matching the parameters from the
database
      // only check active users (1 represents active)
      $result = $this->db->get_where(
                              'users',
                              array(
                                    'username' => $username,
                                    'status' => '1'
                              )
                  );

      // get the results as an object array
      $user = $result->result();

      // if the object array is empty (null), the username is
      // incorrect.
      // Also, even if user exists check the passwords to see
      // if they match
      // if the passwords matches, the user is ok
      if ($user != null && password_verify($password, $user[0]-
      >password)) {
            return $result->result();
      }

      return false;
}
```

4.5.2. Search Box in the website

This code is used to display the search box in the website. The data captured in this form is then passed on to the server. For more details refer to Appendix F.

```
<!-- Start of search form -->
<form class="form-inline" method="GET" action="<?php echo base_url()
?>index.php/Restaurants/search">
      <div class="form-row">
            <div class="col">
                  Search for
            </div>
            <div class="col">
              <!-- this input holds the search query entered by the
                   user -->
              <input class="form-control" type="text" name="q"
placeholder="Restaurant Name"
                                    required="required" value="<?php
echo $this->input->get('q') ?>">
            </div>
            <div class="col">
            in
            </div>
            <div class="col">
              <!-- the available cities are taken from the database
                   and shown as dropdown -->
              <select class="form-control" name="c">
                  <?php foreach ($cities as $city): ?>
                    <!-- when the results are shown the selected
city               will be 'selected' using the passed parameters --
>
                    <option <?php if($this->input->get('c') ==
$city-                        >name) echo 'selected' ?>
                        value="<?php echo $city->name ?>"><?php echo
                        $city->name ?>
                    </option>
                  <?php endforeach ?>
              </select>
            </div>

            <div class="col">
                <!-- submit button -->
                  <button class="btn btn-outline-primary">
                        <i class="fa fa-search fa-fw"></i>
                        Search
                  </button>
            </div>
      </div>
```

4.6. Re-used Codes

- JQuery JavaScript library was used for validation and other JS scripting needs [6]
- Bootstrap frameworks styling components were used to build the website along with its grid feature [7]
- Font-Awesome icon set was used for the system [8]
- CodeIgniter frameworks Query Builder class's functions were used for database manipulation and user interface loading. [9]
- Google Maps JavaScript codes were used to implement the map location functions [10]

CHAPTER 05 – EVALUATION

5.1. Introduction

The purpose of developing a software system is to solve a problem or to increase the efficiency of a process. The developers may have put their best effort to make sure the system satisfy the necessary requirements but unless a good evaluation is conducted, it is difficult to identify any issues that may reside in the system.

Testing the system while in development is a good way to make sure the system is going in the right direction. After the completion of the system, clients should also evaluate the system to make sure the system does what it needs to be done in the right way. The feedback from these testing should be incorporated into the system to improve it.

5.2. Software Testing

There are accepted testing phases when developing software system. Below are the main testing phases that had been adopted when developing the system.

5.2.1. Unit Testing

Unit testing is the process of testing program components, such as methods or object classes. Individual functions or methods are the simplest type of component. [1]

5.2.2. System Testing

System testing during development involves integrating components to create a version of the system and then testing the integrated system. System testing checks that components are compatible, interact correctly and transfer the right data at the right time across their interfaces. [1]

5.2.3. User/Acceptance Testing

User or customer testing is a stage in the testing process in which users or customers provide input and advice on system testing. User testing is essential, even when comprehensive system and release testing have been carried out. The reason for this is that influences from the user’s working environment have a major effect on the reliability, performance, usability, and robustness of a system. [1]

5.3. Test Plan and Test Cases

The test plan is the specification of how and what will be tested at the specified stages of the software development life cycle. It describes the scope, approach, resources and schedule of intended test activities [5].

Unit testing were performed when a system unit was finished to make sure that it performs the expected task without any errors. Since this testing was done parallel to the development process, many errors of the system was uncovered early in the development which helped when developing interconnected units by reducing the errors when integrating.

5.4. System Test Cases

Test cases were identified for each module of the system. Below are the test cases for each module.

5.4.1. User Authentication

Table 1 User Authentication Test Cases

Test No.	Test Case	Inputs / Steps	Expected Result
1	User input validation	Correct username Correct password	Successful login
2	User input validation	Correct username Incorrect password	Error Message
3	User input validation	Incorrect username Correct password	Error Message
4	User input validation	Incorrect username Incorrect password	Error Message
5	User input validation	Both fields empty	Validation error message

5.4.2. Add Employee – User Management Module

Table 2 Add Employee Test Cases

Test No.	Test Case	Inputs / Steps	Expected Result
1	All fields must be filled	Fill all fields	Saving Successful
2	All fields must be filled	One or more fields are not filled	Error Message
3	Joined date is less than today’s date	Set joined date to the day after	Error Message

5.4.3. Add User – User Management Module

Table 3 Add User Test Cases

Test No.	Test Case	Inputs / Steps	Expected Result
1	All fields must be filled	Fill all fields	Saving Successful
2	All fields must be filled	One or more fields are not filled	Error Message
3	Username must be more than 3 characters	Username with 5 characters	No error message
4	Username must be more than 3 characters	Username with 2 characters	Error Message
5	Password must be more than 5 characters long	Password with 10 characters	No error message
6	Password must be more than 5 characters long	Password with 5 characters	Error message
7	Repeat password field must match the password field’s value	Same password on both fields	No error
8	Repeat password field must match the password field’s value	Different values on the two fields	Error message

5.4.4. Order Management Module

Table 4 Order Management Test Cases

Test No.	Test Case	Inputs / Steps	Expected Result
1	New orders should have pending status assigned	Create a new order through website	Order status value is set to ‘pending’
2	When a courier is assigned to order, status should change to dispatched	Assign a courier to an order	After saving order status is set to ‘Dispatched’

5.4.5. Restaurant Management Module

Table 5 Restaurant Management Test Cases

Test No.	Test Case	Inputs / Steps	Expected Result
1	Add new restaurant	Fill all fields	Saving Successful
2	Add new restaurant	One or more fields are not filled	Error Message
3	Add new restaurant	Logo/cover image must be less than 3 MB in size	Error Message
4	Add new restaurant	Logo/cover image is not an image file	Error Message
5	Add new restaurant	Tagline is more than 150 characters long	Error message
6	Add new restaurant	Latitude & Longitude are integer values	Error message

5.4.6. Website Module

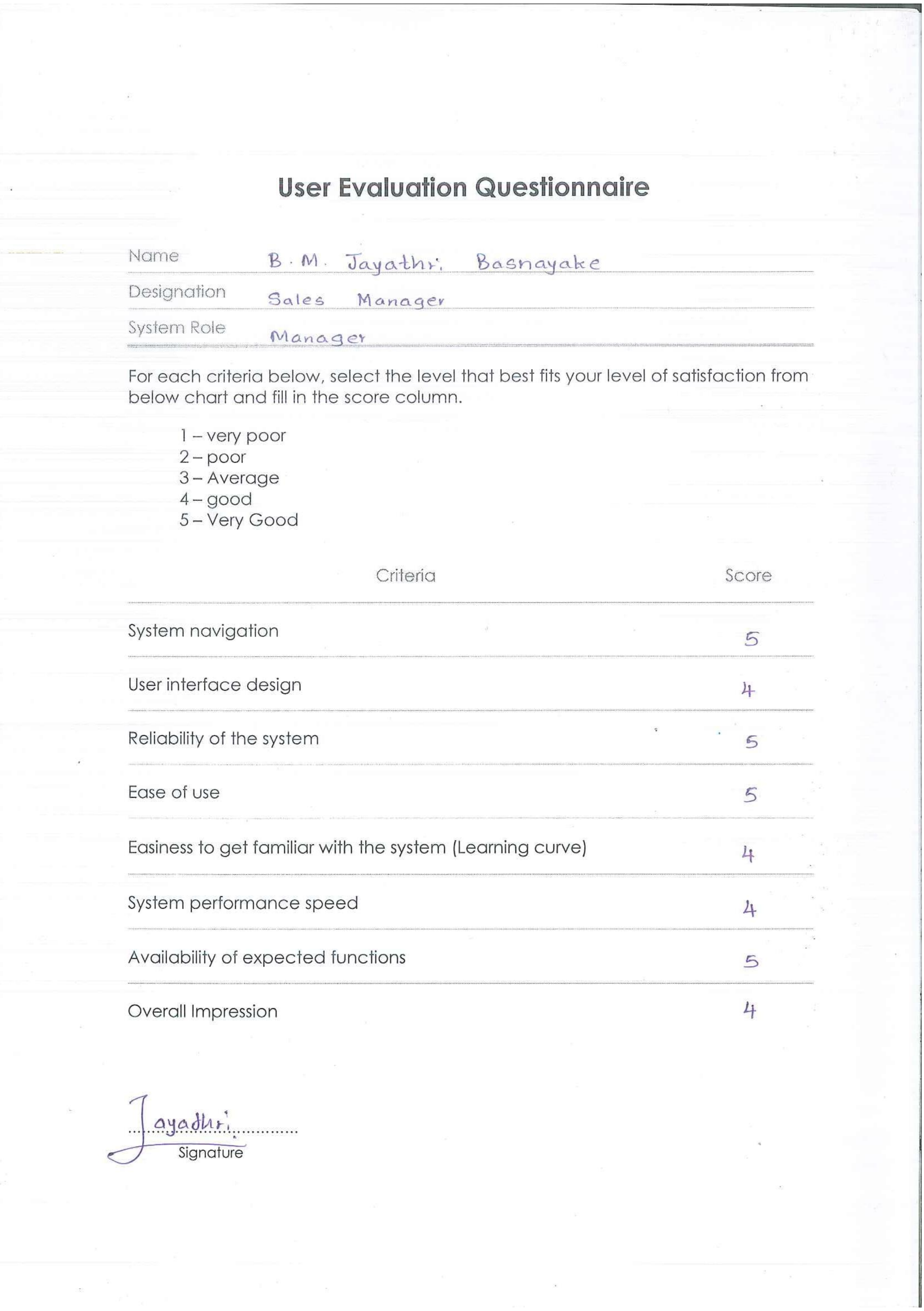
Table 6 Website Test Cases

Test No.	Test Case	Inputs / Steps	Expected Result
1	Search restaurants	Non existing restaurant name	No results message
2	Search restaurants	Empty search box No city selected	Error message
3	Search restaurants	Empty search box City selected	All restaurants in that city
4	Access user profile page	User is not logged in	Redirect to login page with error message
5	Access user profile page	User is logged in	Load page successfully
6	Sign up	All fields area filled	Signup user
7	Sign up	Email address is invalid	Show error message
8	Sign up	Contact no is invalid	Show error message
9	Sign up	Email already registered	Show error message
10	Sign up	Password is less than 8 characters long	Show error message
11	Sign Up	Repeat password value is different to password field value	Show error message

5.6. User Evaluation

Few employees from the client company was assigned to test the system as an available user roles to evaluate the whole system. They tested accompanying website and provided helpful feedback on some of the functionality. The website received positive feedback from the client and then they proceeded to evaluate the system. The administrator has access to the complete system and thus the client manager was able to evaluate each area of the system and provide feedback where things needed modifications or improvements. Overall the manager was satisfied with the system. The client certificate that was received has been appended to Appendix G.

The user evaluation questionnaires along with the feedback from the client company's employees are given in figure 5.1, figure 5.2 and figure 5.3.



User Evaluation Questionnaire

Name B. M. Jayathri. Basnayake

Designation Sales Manager

System Role Manager

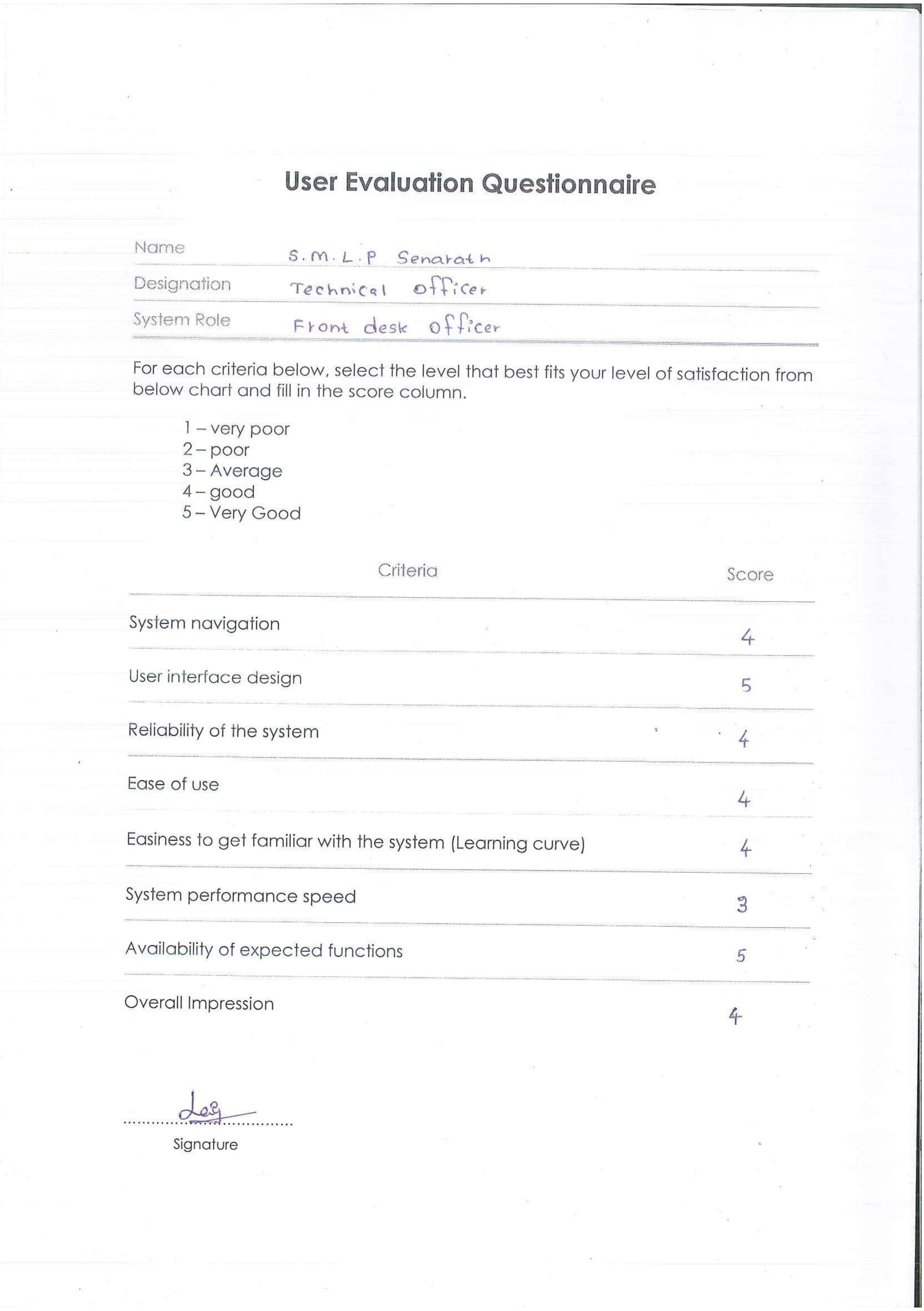
For each criteria below, select the level that best fits your level of satisfaction from below chart and fill in the score column.

1 – very poor
2 – poor
3 – Average
4 – good
5 – Very Good

Criteria	Score
System navigation	5
User interface design	4
Reliability of the system	5
Ease of use	5
Easiness to get familiar with the system (Learning curve)	4
System performance speed	4
Availability of expected functions	5
Overall Impression	4

Jayadhri.
Signature

Figure 5.1 User Evaluation Questionnaire 1



User Evaluation Questionnaire

Name S.M.L.P Senarath

Designation Technical officer

System Role Front desk officer

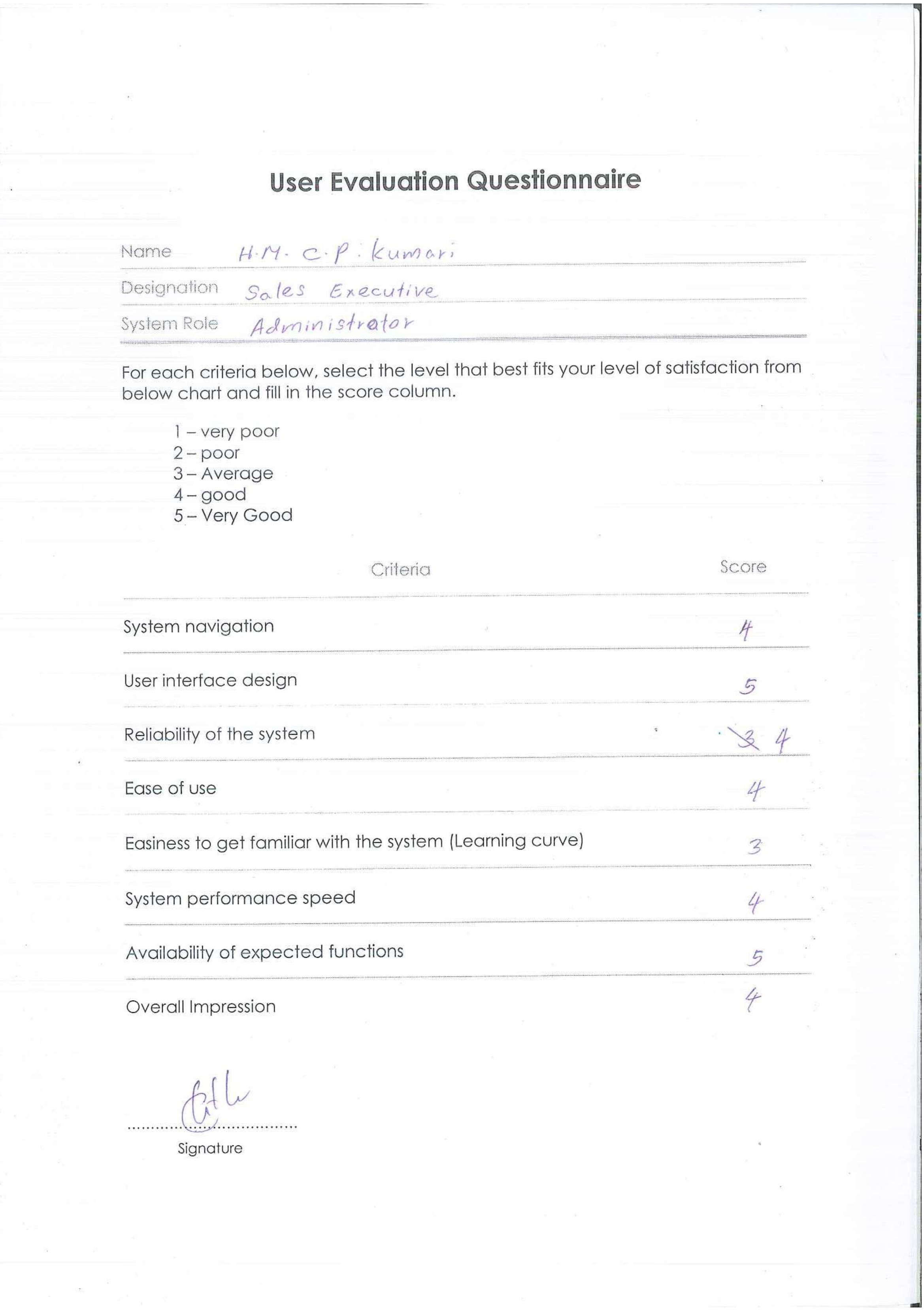
For each criteria below, select the level that best fits your level of satisfaction from below chart and fill in the score column.

1 – very poor
2 – poor
3 – Average
4 – good
5 – Very Good

Criteria	Score
System navigation	4
User interface design	5
Reliability of the system	4
Ease of use	4
Easiness to get familiar with the system (Learning curve)	4
System performance speed	3
Availability of expected functions	5
Overall Impression	4

....
Signature

Figure 5.2 User Evaluation Questionnaire 2



User Evaluation Questionnaire

Name H.M. C.P. kumari

Designation Sales Executive

System Role Administrator

For each criteria below, select the level that best fits your level of satisfaction from below chart and fill in the score column.

1 – very poor
2 – poor
3 – Average
4 – good
5 – Very Good

Criteria	Score
System navigation	4
User interface design	5
Reliability of the system	~~3~~ 4
Ease of use	4
Easiness to get familiar with the system (Learning curve)	3
System performance speed	4
Availability of expected functions	5
Overall Impression	4

....
Signature

Figure 5.3 User Evaluation Questionnaire 3

CHAPTER 06 – CONCLUSION

6.1. introduction

The current restaurant market is a very competitive one as more and more restaurants are coming into the field. Many of the giants in the restaurant industry provides convention features such as home delivery which attracts lots of customers because of the ease of ordering and travelling. While bigger players in the industry can manage to have their own fleet of delivery vehicles, small and medium sized restaurants cannot afford such a fleet.

After seeing that these restaurants can benefit from having home delivery facility, the client company was inspired to introduce a solution to help these restaurant owners. They came up with the idea to provide delivery as a service to these restaurants as a solution. This solves the need for the restaurants to manage their own delivery fleets and hiring additional staff to manage ordering and also the cost of developing their own online ordering platforms from scratch. This system was built to assist the whole process from the ordering to the delivery and also to help manage the content of the website.

The clients requirements were analyzed with the help of similar existing business that provide delivery options. The system’s goal was to automate and streamline the process from the beginning to the end and provide easy to use functions to the users to help them reduce human error.

The system was developed with those things in mind and the client testing provided positive feedback on the system and the suggested modification were incorporated as needed.

It is difficult to identify and develop a system that can perform perfectly and provide all the required functions that a user of that system expects, but based on the client feedback it can be said that the system has managed to achieve clients expected objectives to a satisfactory level.

6.2. Lessons Learnt

As this project was a web based solution, it provided lot of opportunities to learn about the web technologies. From the value of commenting the code to the importance of good design diagrams was learnt when developing the system. And as time constrained project, naturally it is became important to learn to manage the time to balance the project work with other work and studies.

Some difficulties were faced when developing the system. Mainly when developing the website. It became very clear that designing good user interfaces is not an easy task and making thing simple is as simple as it seems to be. To solve this many design article were referred to get the inspiration to make a appealing interface to the best of my abilities.

Database design was another area that provided many learning opportunities as it became clear that having a good database design is crucial to the success of the system more than any other area. Some of the concept leant from the database subjects were used when creating the database and the knowledge gathered while designing the database was very helpful.

6.3. Critical Assessment of project

The system in the current completed state can support the complete process from browsing and placing orders to ultimately managing the delivery of the order. The user interface design of the website module is not the most creative or easy to use interface at the moment and can be and has to be improved to match the competition in the market. A lot of thought was given to security and scalability as much as the experience and knowledge of the developer allowed.

The main differentiating factor of the developed system from other solutions mentioned in section 2.6, is the ability to handle many restaurants using a single system rather than one restaurant at a time. As this feature is the main point of the system, it was given high priority. It can be said with very much certainty that the system does perform as expected in regard to the functionality.

There are few area such as order location tracking, online payments and ability to use social accounts to login to system that have not been implemented, largely due to the novelty of these technologies to the developer rather than any technical limitation. These features will be implemented in future increments of the system.

As a final word, the system provides all needed functions to manage multiple restaurant delivery infrastructure with satisfactory usability and performance.

6.4. Future Improvements

The current system was developed in a limited time frame and as a result some of the functions that may improve the system will have to incorporated in a future version. Below are some of the future improvements that can be added to the system.

- Providing a mobile application to ease the ordering process
- Provide online payment facility by integrating a payment gateway into the system
- Provide a mobile application to the couriers with GPS tracking to enable customers to track their orders in real time using the above mentioned mobile app
- Provide more features to restaurants when managing their own listing like including more details and such.
- Introduce a restaurant reviewing and rating program through the website to enable visitors to become interact more with the system.
- Add more reports to assist the managers

References

[1] Sommerville, I. (2011). *Software engineering*. Boston: Addison-Wesley.

[2] "Restaurant Online Ordering & Management Software", *TastyIgniter*, 2017. [Online]. Available: https://tastyigniter.com/. [Accessed: 03- Aug- 2017].

[3] "Restaurant Delivery Management Software", *Landing.orderlord.com*, 2017. [Online]. Available: http://landing.orderlord.com/. [Accessed: 03- Aug- 2017].

[4] *Rational Unified Process: Best Practices for Software development Teams*. Rational Software, 2003.

[5] "Test Plan - Software Testing Fundamentals", Software Testing Fundamentals, 2017. [Online]. Available: http://softwaretestingfundamentals.com/test-plan/. [Accessed: 06- Nov- 2017].

[6] jquery.org, "jQuery", Jquery.com, 2017. [Online]. Available: https://jquery.com/. [Accessed: 05- Nov- 2017].

[7] "Bootstrap", Getbootstrap.com, 2017. [Online]. Available: http://getbootstrap.com/. [Accessed: 05- Nov- 2017].

[8] Fontawesome.io, 2017. [Online]. Available: http://fontawesome.io/. [Accessed: 05- Nov- 2017].

[9] "CodeIgniter Web Framework", Codeigniter.com, 2017. [Online]. Available: https://codeigniter.com/. [Accessed: 05- Nov- 2017].

[5] "Google Maps JavaScript API | Google Developers", Google Developers, 2017. [Online]. Available: https://developers.google.com/maps/documentation/javascript/. [Accessed: 05- Nov- 2017].

APPENDIX A – SYSTEM DOCUMENTATION

This document lists the steps needed to set up the system and the software and hardware requirements of the system.

Minimum Hardware Requirements

- Intel Core 2 Duo processor
- 1GB of RAM
- 40GB Hard Disk Capacity
- 1024 * 768 Monitor
- Internet Router or Modem

Software Requirements

- Windows 7 or higher Operating System
- WAMP v2.5 or separate installations of
    - PHP 5.5.12
    - MySQL 5.6.17
    - Apache Server 2.4.9
    - phpMyAdmin 4.1.14
- Google Chrome or Mozilla Firefox web browser
- A PDF reader software (Adobe Acrobat Reader)

System Installation

1. Install WAMP Bundle using the installer from http://www.wampserver.com/en/
2. Install Google Chrome or Mozilla Firefox if not already installed
3. Start WAMP application and verify all components are running using the indicator on the notification panel. WAMP icon should turn green if all components are started.
4. Open Google Chrome or Mozilla Firefox
5. Go to http://localhost/phpmyadmin
6. Click on 'New' icons on the top of the left sidebar to create a new database
7. Provide 'deliverit' as the database name and create the database
8. Go to the created database by selecting it from the left sidebar

9. Click on ‘Import’ tab on the top tab row
10. Click on ‘Choose File’ button and browse to the CD and open the ‘deliverit.sql’ file in the Database folder
11. After successful import, copy the project folder from the CD to C:\wamp\www folder
12. Go to C:\wamp\www\deliverit\application\config\database.php and update the database connection information of the website
13. Go to C:\wamp\www\deliverit\admin\application\config\database.php and update the database connection information for the system
14. If the project folder was copied to a subdirectory of the www folder, please go to C:\wamp\www\deliverit\application\config\config.php and update the base URL in line 26 accordingly.
15. Also update the base URL if needed in the system configuration file located in C:\wamp\www\deliverit\admin\application\config\config.php in line 26.
16. Now you can go to http://localhost/deliverit to access the system website
17. You can access the system using http://localhost/deliverit/admin
18. Use the credentials below to login to the system as an administrator
    - Username: Admin
    - Password: aaaaaa
19. Please refer to Appendix C on how to navigate and use the system.

APPENDIX B – DESIGN DOCUMENTATION

Use Case Diagrams with Detailed Descriptions

Below are the detailed use cases of the high level use case given in chapter 3.

User management Module

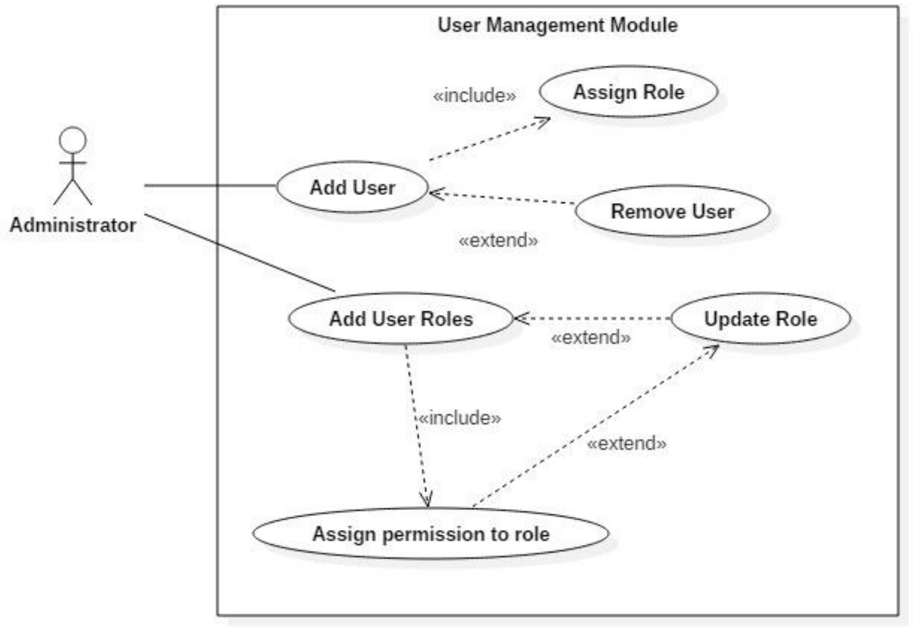


Figure B1 User Management Module Use Case

Table 7 User Management Use Case Description

Use case	User Management
Actor	Administrator
Description	Add, update and remove users
Pre-Condition	
The logged in user must be an administrator	
Flow of Events	
1. Add users 2. Assign a role to user 3. Update user details	
Post-Condition	
The added user should be saved with the given permissions	
Use case	Privileges Management
Actor	Administrator
Description	Add, update and remove roles and their privileges
Pre-Condition	

The logged in user must be an administrator
Flow of Events
1. Add roles 2. Assign privileges to roles 3. Update role
Post-Condition
The added role should be saved with the given privileges

Order management Module

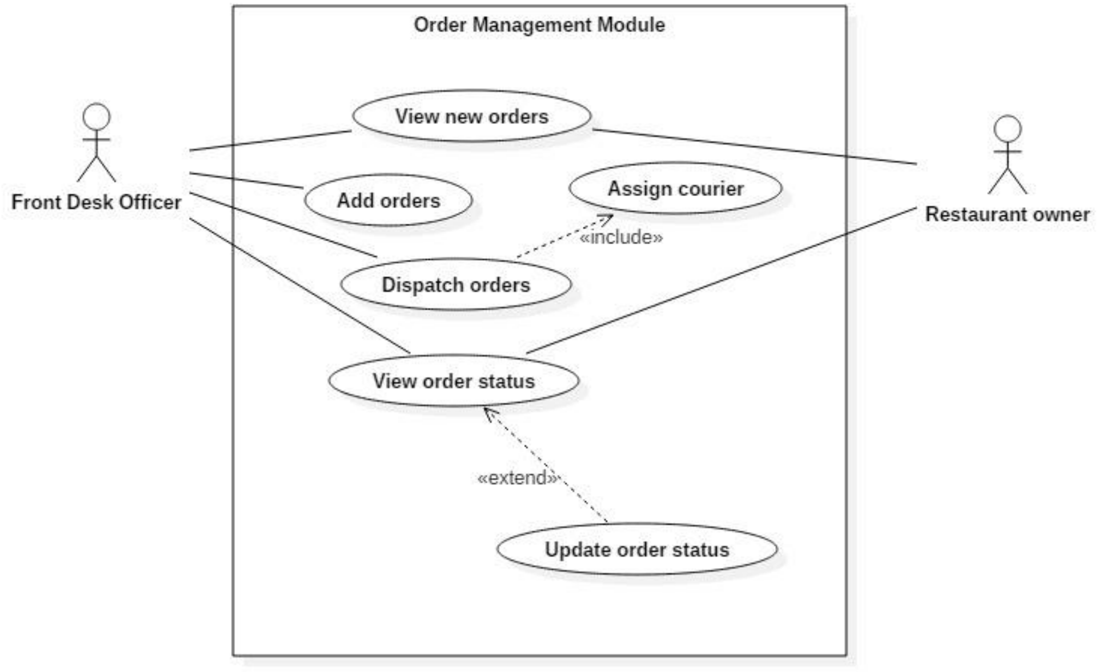


Figure B2 Order Management Module Use Case

Table 8 User Management Use Case Description

Use case	Order Management
Actor	Front Desk Officer, Restaurant Owner
Description	View pending orders and dispatch them by assigning couriers. Restaurant owners can check for dispatched orders and ready them while couriers arrive.
Pre-Condition	
There must be pending orders and both users must be logged in to the system	
Flow of Events	

1. View orders 2. Assign couriers 3. Restaurant owner readies orders 4. Order status are updated when they are ready 5. Orders are handed over to the courier 6. Order status is updated to ‘delivered’ after successful delivery
Post-Condition
The status of the order should be marked as delivered

Restaurant management Module

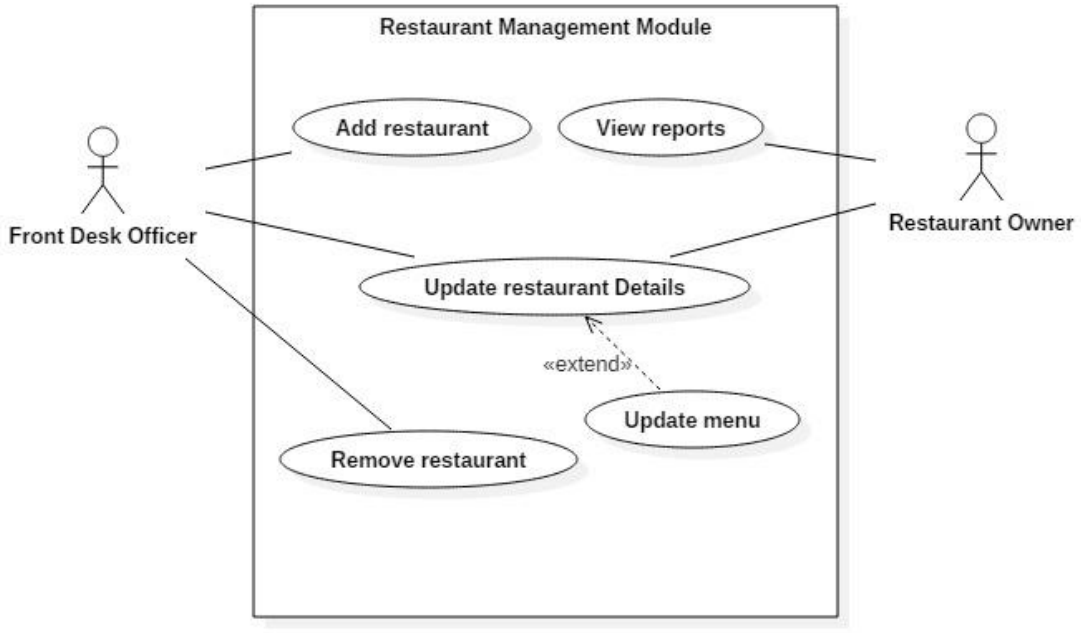


Figure B3 Restaurant Management Module Use Case

Table 9 Add Restaurant Use Case Description

Use case	Add restaurants
Actor	Front Desk Officer
Description	Add new restaurants
Pre-Condition	
The user must be logged in to the system and the restaurants has been subscribed to DeliverIt service.	
Flow of Events	
1. Add Restaurant Details 2. Save details 3. View added restaurant	
Post-Condition	
The added restaurant must show up in the website correctly	

Customer management Module

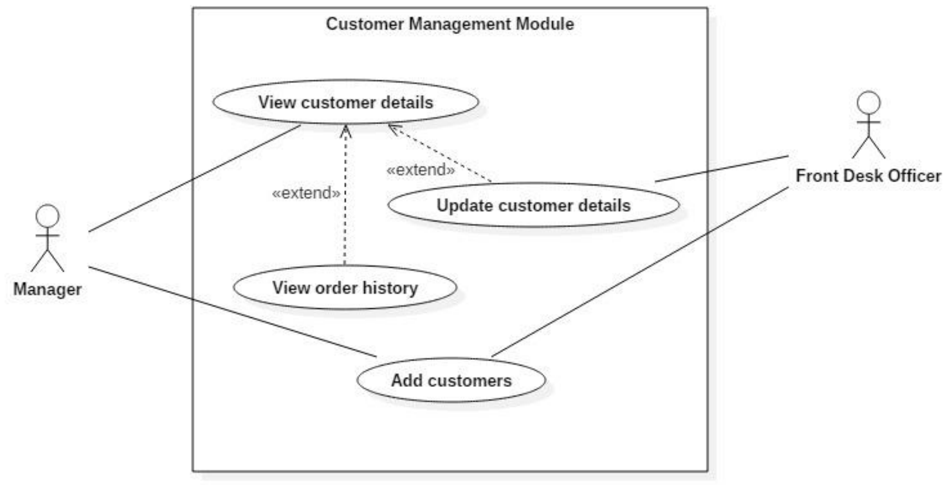


Figure B4 Customer Management Module Use Case

Table 10 Customer Management Use Case Description

Use case	Customer Management
Actor	Front Desk Officer, Manager
Description	Add, View and update customer details and view their order history
Pre-Condition	
The user must be logged in to the system and must have the correct privileges. There must be at least one placed order to see a customer's order history	
Flow of Events	
1. View customer 2. View order history 3. Update details	
Post-Condition	
Any updates must be saved in the database correctly	

Website Module

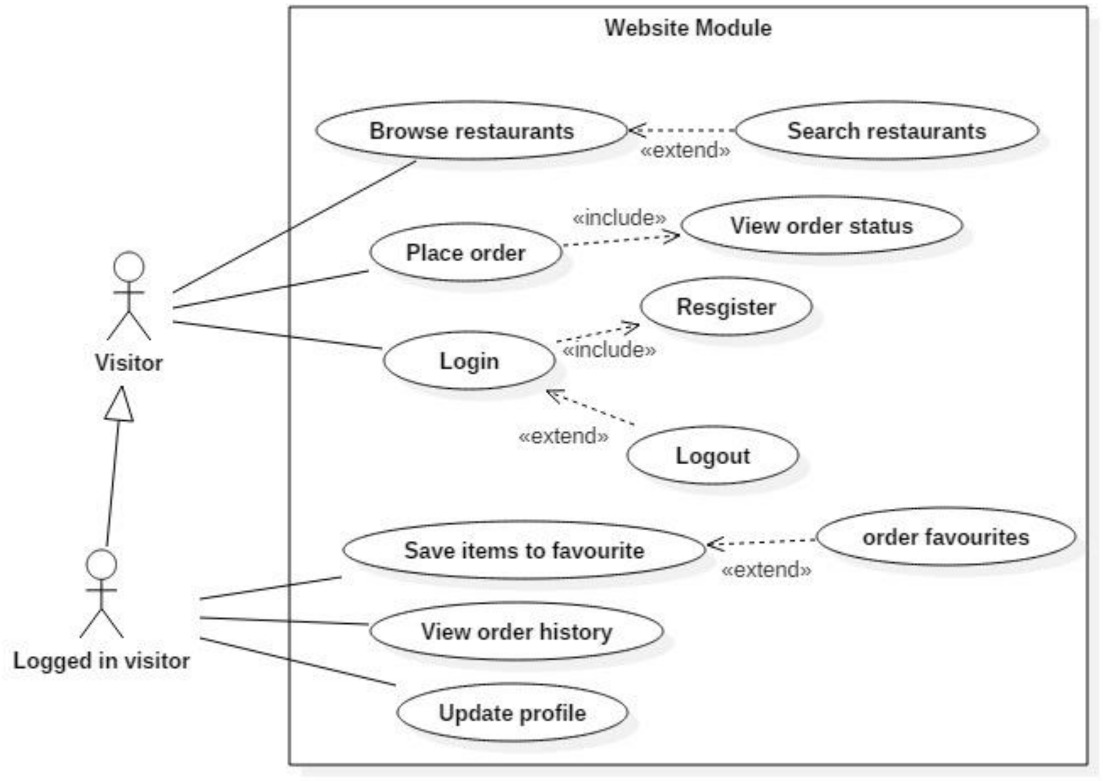


Figure B5 Website Module Use Case

Table 11 Browse Web Site Use Case Description

Use case	Browse Web Site
Actor	Visitor
Description	Browse the website to find restaurants and menus
Pre-Condition	
User must visit the website	
Flow of Events	
1. Visit website 2. Search restaurants 3. Select items	
Post-Condition	
Selected items are added to the customer’s cart	

Table 12 Place Orders Use Case Description

Use case	Place orders
Actor	Visitor
Description	Place orders on the selected items
Pre-Condition	
Visitor must select at least one item to order	
Flow of Events	
1. Select items 2. Provide information 3. Place order 4. Receive confirmation	
Post-Condition	
A confirmation email is sent after placing an order. And a new record is added to the database	

Website management Module

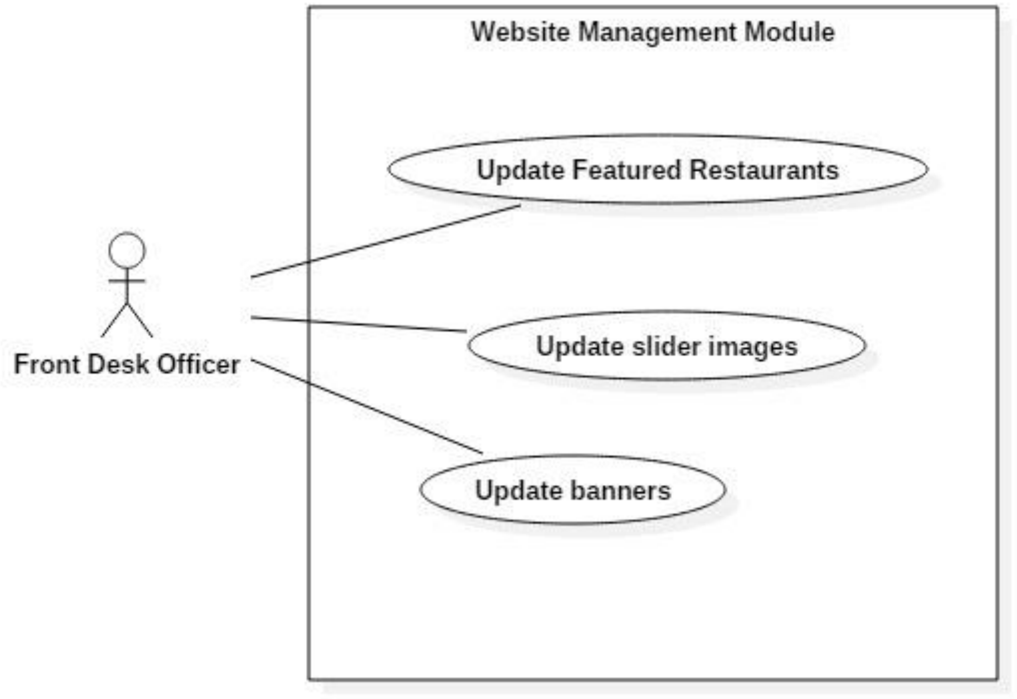


Figure B6 Website Management Module Use Case

Table 13 Manage Website Use Case Description

Use case	Manage Website
Actor	Front Desk Officer
Description	Update the information on the website
Pre-Condition	
The user must be logged in to the system and has the correct permissions	
Flow of Events	
1. View items 2. Edit items	
Post-Condition	
The modifications have to show up in the website correctly	

Courier Module

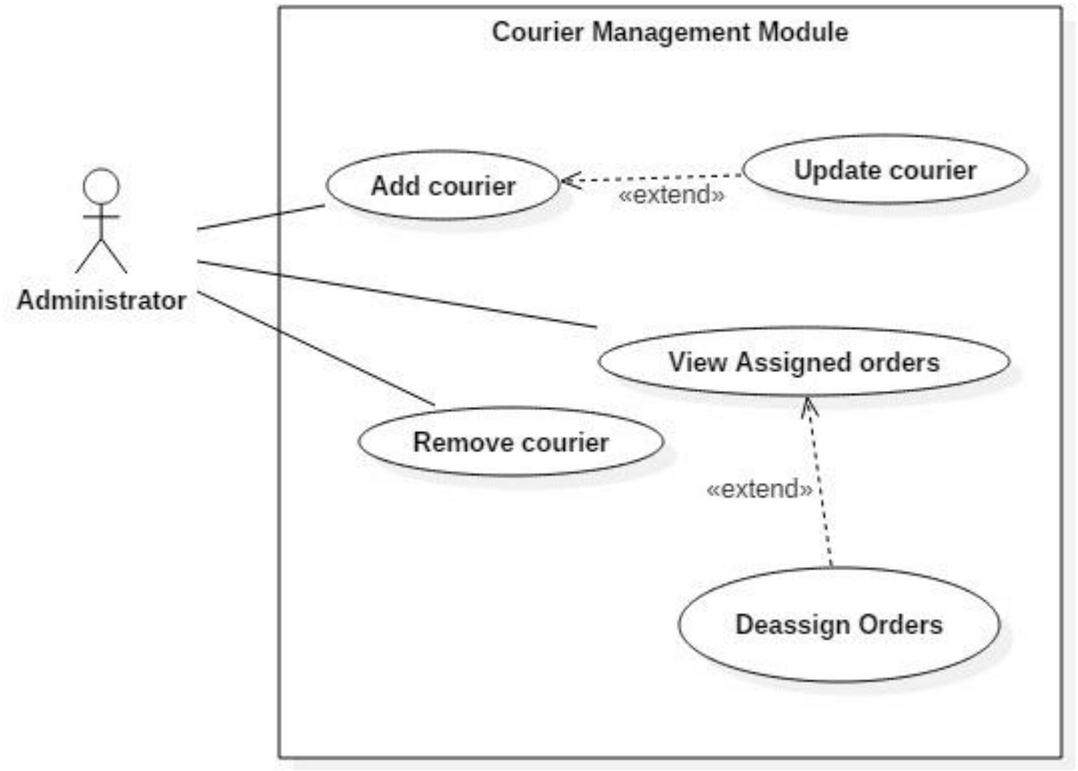


Figure B7 Courier Management Module Use Case

Table 14 Courier Management Use Case Description

Use case	Courier Management
Actor	Administrator
Description	Manage the couriers and view assigned orders and remove them from the courier if needed
Pre-Condition	
The logged in user must be an administrator.	
Flow of Events	

1. View courier 2. Update courier 3. View assigned orders 4. Remove assigned orders
Post-Condition
Modifications are saved in the database and de-assigned orders\ status changes to de-assigned

APPENDIX C – USER DOCUMENTATION

- ## Logging in to the system

Go to the admin panel address using your web browser and a login form will appear. Use the given username and password to login to the system. See figure C1.

If you do not remember the username or your password, Please contact your administrator.

- ## Dashboard

After logging in you will directed to your dashboard. The look of the dashboard depends on the user level and for an administrator

The dashboard contains an at-a-glance view of the system (figure C2). With the number of pending orders, number of registered customers in the system and the total number of restaurants in the system along with the number of owners.

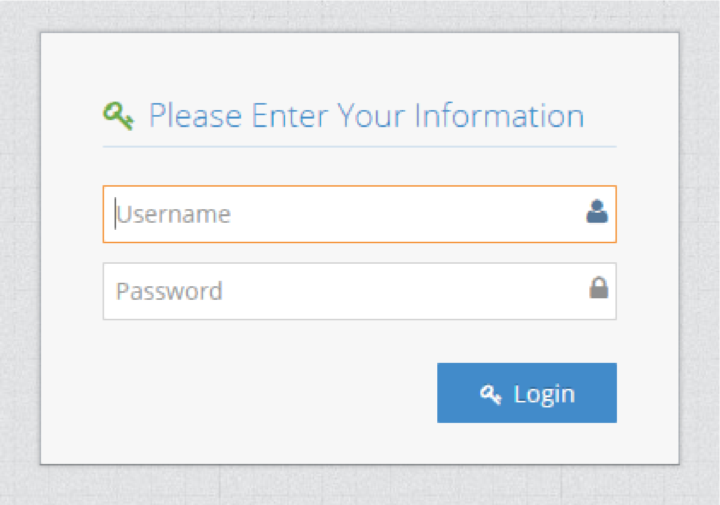
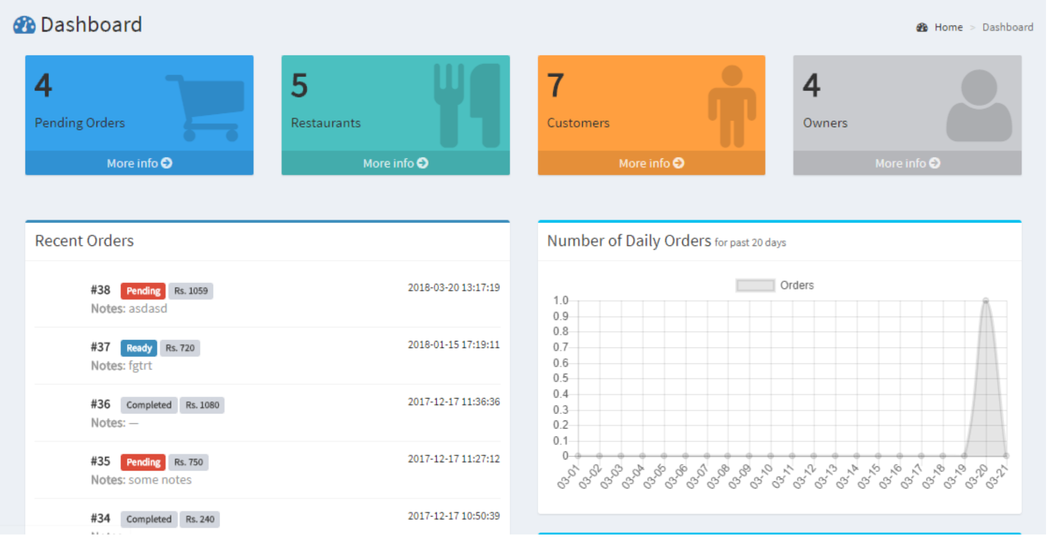


Figure C1 Login Form



Recent Orders

#38 Pending Rs. 1059 2018-03-20 13:17:19
Notes: asdasd

#37 Ready Rs. 720 2018-01-15 17:19:11
Notes: fgtrt

#36 Completed Rs. 1080 2017-12-17 11:36:36
Notes: —

#35 Pending Rs. 750 2017-12-17 11:27:12
Notes: some notes

#34 Completed Rs. 240 2017-12-17 10:50:39

Figure C2 Dashboard

After that a pie chart shows the number of orders that are in each of the states.

- ## Side Bar

Side bar houses the main navigation area of the system. From this you can switch to any part of the system, given you have the necessary permissions. Figure C3 shows the navigation menu as it would be displayed to an administrator.

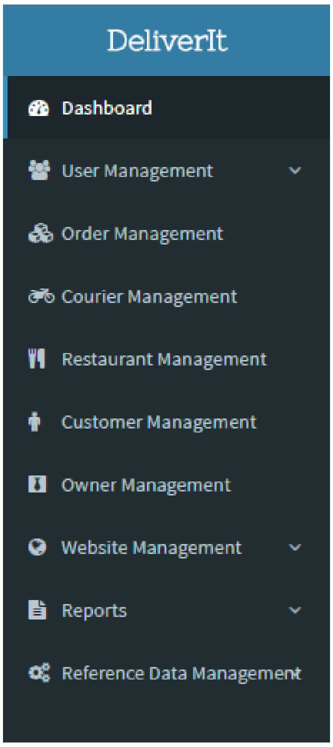
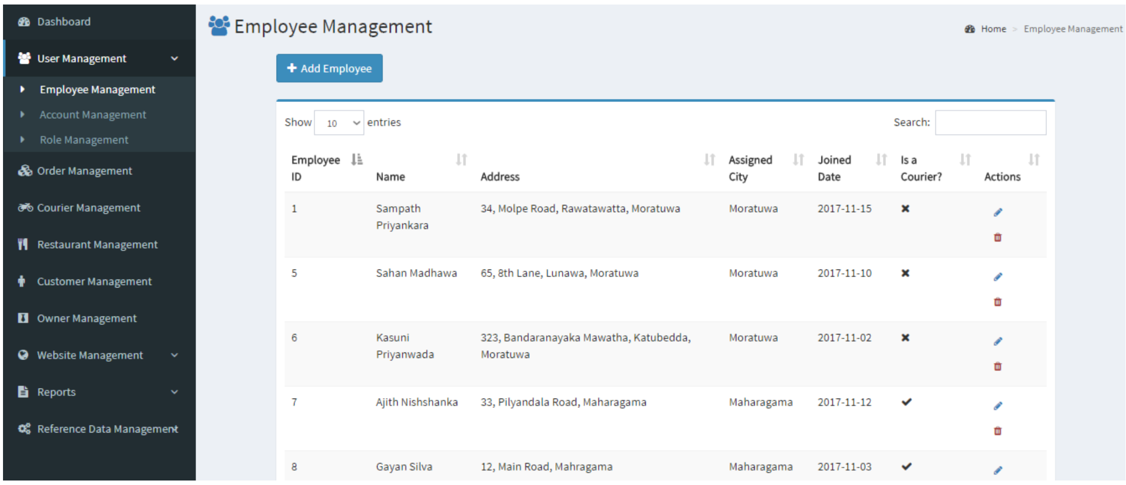


Figure C3 Sidebar Navigation

- ## User Management

This is module responsible for managing the user accounts. You can create and assign permissions to user account from this section. Only an administrator can access this section. See figure C4



Employee Management

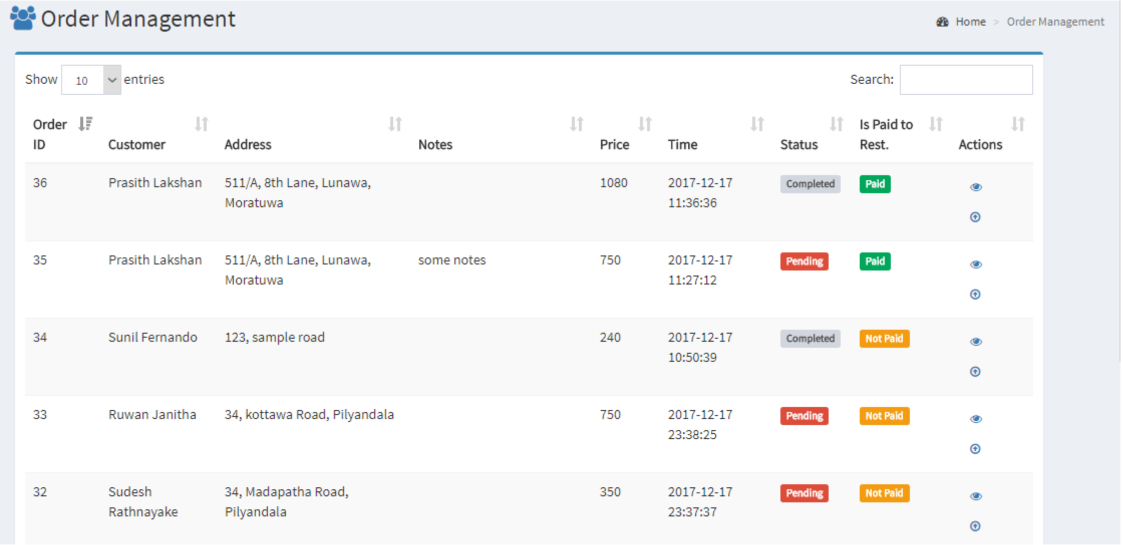
Show 10 entries Search:

Employee ID	Name	Address	Assigned City	Joined Date	Is a Courier?	Actions
1	Sampath Priyankara	34, Molpe Road, Rawatawatta, Moratuwa	Moratuwa	2017-11-15	✗	
5	Sahan Madhawa	65, 8th Lane, Lunawa, Moratuwa	Moratuwa	2017-11-10	✗	
6	Kasuni Priyanwada	323, Bandaranayaka Mawatha, Katubedda, Moratuwa	Moratuwa	2017-11-02	✗	
7	Ajith Nishshanka	33, Pilyandala Road, Maharagama	Maharagama	2017-11-12	✓	
8	Gayan Silva	12, Main Road, Mahragama	Maharagama	2017-11-03	✓	

Figure C4 Users list

- ## Order Management

This is where the orders placed by the customers can be viewed in. All the orders are shown here (figure C5) with the latest orders at the top.



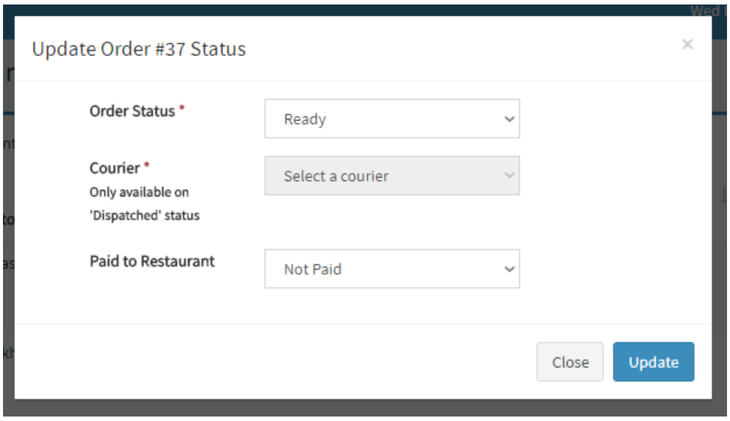
Order Management

Show 10 entries Search:

Order ID	Customer	Address	Notes	Price	Time	Status	Is Paid to Rest.	Actions
36	Prasith Lakshan	511/A, 8th Lane, Lunawa, Moratuwa		1080	2017-12-17 11:36:36	Completed	Paid	
35	Prasith Lakshan	511/A, 8th Lane, Lunawa, Moratuwa	some notes	750	2017-12-17 11:27:12	Pending	Paid	
34	Sunil Fernando	123, sample road		240	2017-12-17 10:50:39	Completed	Not Paid	
33	Ruwan Janitha	34, kottawa Road, Pilyandala		750	2017-12-17 23:38:25	Pending	Not Paid	
32	Sudesh Rathnayake	34, Madapatha Road, Pilyandala		350	2017-12-17 23:37:37	Pending	Not Paid	

Figure C5 Order list

Use the view button to view the order and to update the order status, use the update button next to the view button. A popup will show up with option to update order status and assign a courier if needed as in figure C6.



Update Order #37 Status

Order Status *: Ready

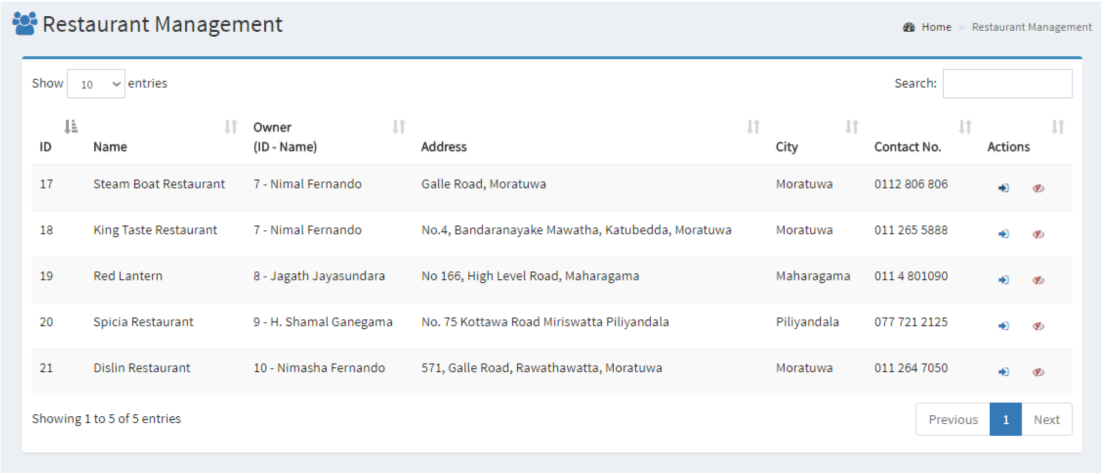
Courier *
Only available on 'Dispatched' status: Select a courier

Paid to Restaurant: Not Paid

Figure C6 Order Status Update popup

- ## Restaurant Management

This section helps to view and manage all the registered restaurants in the system. This section provides the ability to view and edit restaurants as well as to disable them if necessary. The admin can login to the owner's account to view and edit restaurant directly from this page. See figure C7.



Restaurant Management

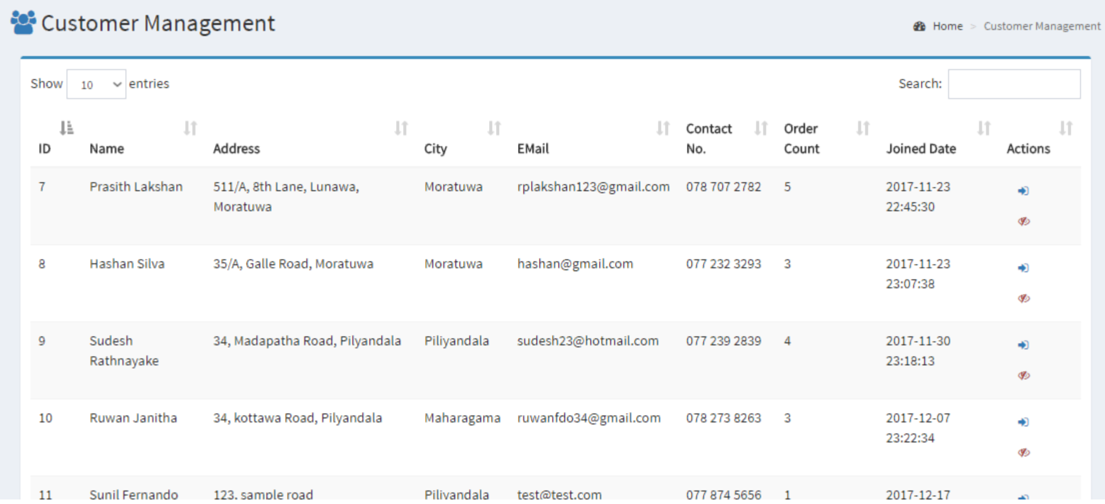
ID	Name	Owner (ID - Name)	Address	City	Contact No.	Actions
17	Steam Boat Restaurant	7 - Nimal Fernando	Galle Road, Moratuwa	Moratuwa	0112 806 806	
18	King Taste Restaurant	7 - Nimal Fernando	No.4, Bandaranayake Mawatha, Katubedda, Moratuwa	Moratuwa	011 265 5888	
19	Red Lantern	8 - Jagath Jayasundara	No 166, High Level Road, Maharagama	Maharagama	011 4 801090	
20	Spicia Restaurant	9 - H. Shamal Ganegama	No. 75 Kottawa Road Miriswatta Piliyandala	Piliyandala	077 721 2125	
21	Dislin Restaurant	10 - Nimasha Fernando	571, Galle Road, Rawathawatta, Moratuwa	Moratuwa	011 264 7050	

Showing 1 to 5 of 5 entries

Figure C7 Restaurant management list

- ## Customer & Owner Management Sections

These two pages provide ability to view, disable (block login and suspend account) and to login to the front end website as the selected customer or owner and edit their information. See figure C8.



Customer Management

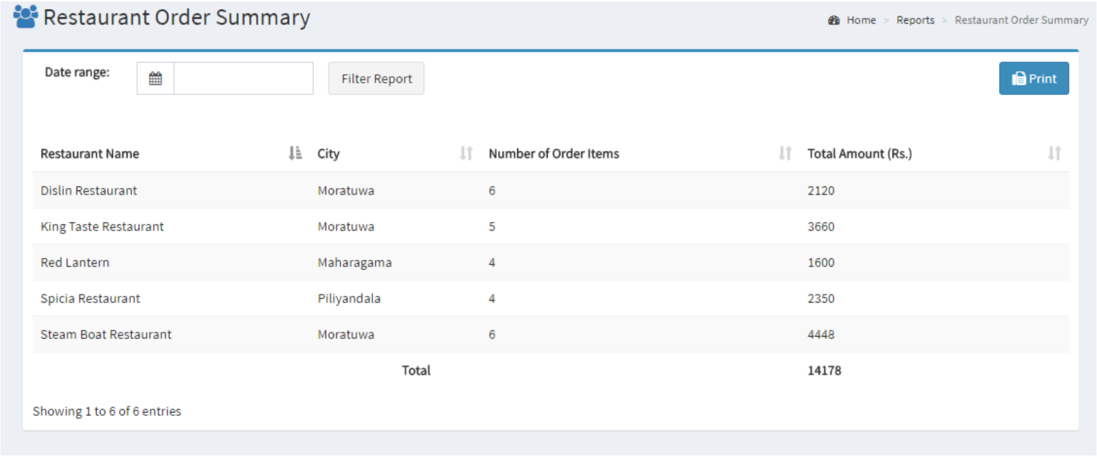
ID	Name	Address	City	EMail	Contact No.	Order Count	Joined Date	Actions
7	Prasith Lakshan	511/A, 8th Lane, Lunawa, Moratuwa	Moratuwa	rplakshan123@gmail.com	078 707 2782	5	2017-11-23 22:45:30	
8	Hashan Silva	35/A, Galle Road, Moratuwa	Moratuwa	hashan@gmail.com	077 232 3293	3	2017-11-23 23:07:38	
9	Sudesh Rathnayake	34, Madapatha Road, Pilyandala	Piliyandala	sudesh23@hotmail.com	077 239 2839	4	2017-11-30 23:18:13	
10	Ruwan Janitha	34, kottawa Road, Pilyandala	Maharagama	ruwanfdo34@gmail.com	078 273 8263	3	2017-12-07 23:22:34	
11	Sunil Fernando	123. sample road	Piliyandala	test@test.com	077 874 5656	1	2017-12-17	

Figure C8 Customer Management Section

Other modules behave in a similar pattern and the user interface is designed to be consistent between each module. So, It is easier to use the system without depending on guidance.

- ## Report Generation

Go to the reports section using the sidebar menu and select the report type you want to view. It will display the report and provide options to customize the report further. See figure C9.



Restaurant Order Summary

Home > Reports > Restaurant Order Summary

Date range: Filter Report Print

Restaurant Name	City	Number of Order Items	Total Amount (Rs.)
Dislin Restaurant	Moratuwa	6	2120
King Taste Restaurant	Moratuwa	5	3660
Red Lantern	Maharagama	4	1600
Spicia Restaurant	Piliyandala	4	2350
Steam Boat Restaurant	Moratuwa	6	4448
	Total		14178

Showing 1 to 6 of 6 entries

Figure C9 Restaurant summary Report

Below figure C10 shows another graphical report available in the system.

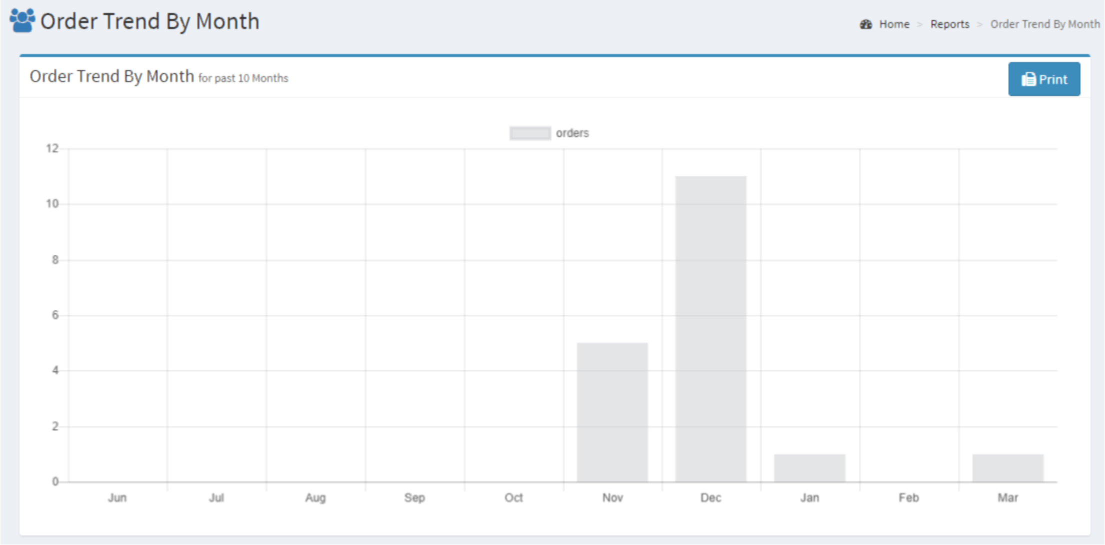
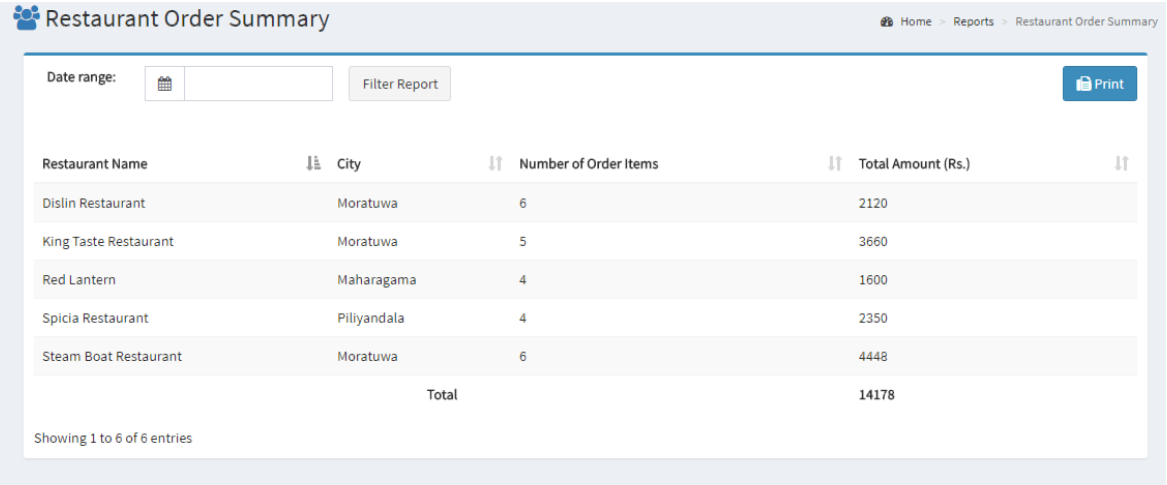


Figure C10 Order Trend by Month Report

APPENDIX D – MANAGEMENT REPORTS

Restaurant Order Summary Report

This report (figure D1) is to identify what restaurants are performing well and providing the most amount of orders. This can be customized to summarize the data between the given dates.



Date range:

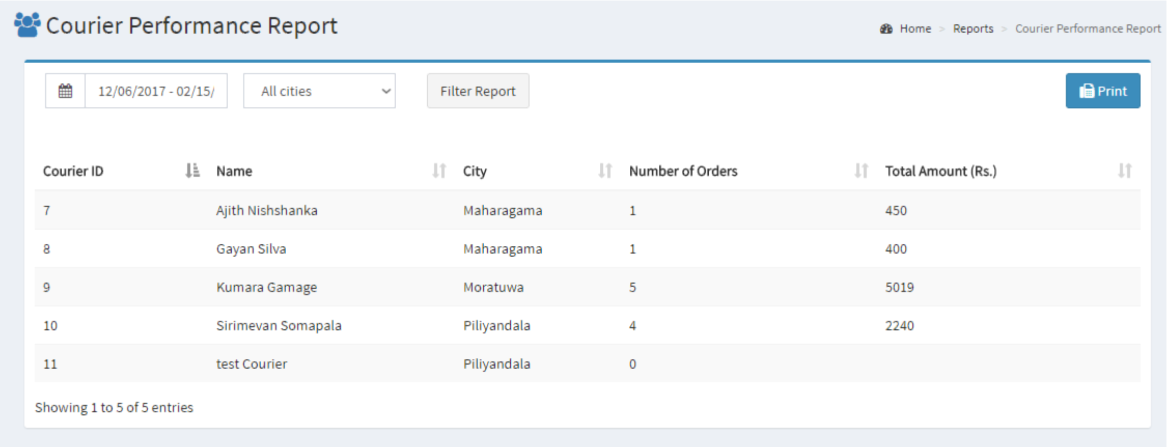
Restaurant Name	City	Number of Order Items	Total Amount (Rs.)
Dislin Restaurant	Moratuwa	6	2120
King Taste Restaurant	Moratuwa	5	3660
Red Lantern	Maharagama	4	1600
Spicia Restaurant	Piliyandala	4	2350
Steam Boat Restaurant	Moratuwa	6	4448
		Total	**14178**

Showing 1 to 6 of 6 entries

Figure D1 Restaurant Order Summary Report

Courier Performance Report

This reports (figure D3) analyses how couriers have performed over a given time period. In addition to providing a time range, the city of the couriers can also be specified.


Courier Performance Report
Home > Reports > Courier Performance Report
Filter Report
Print

12/06/2017 - 02/15/

All cities

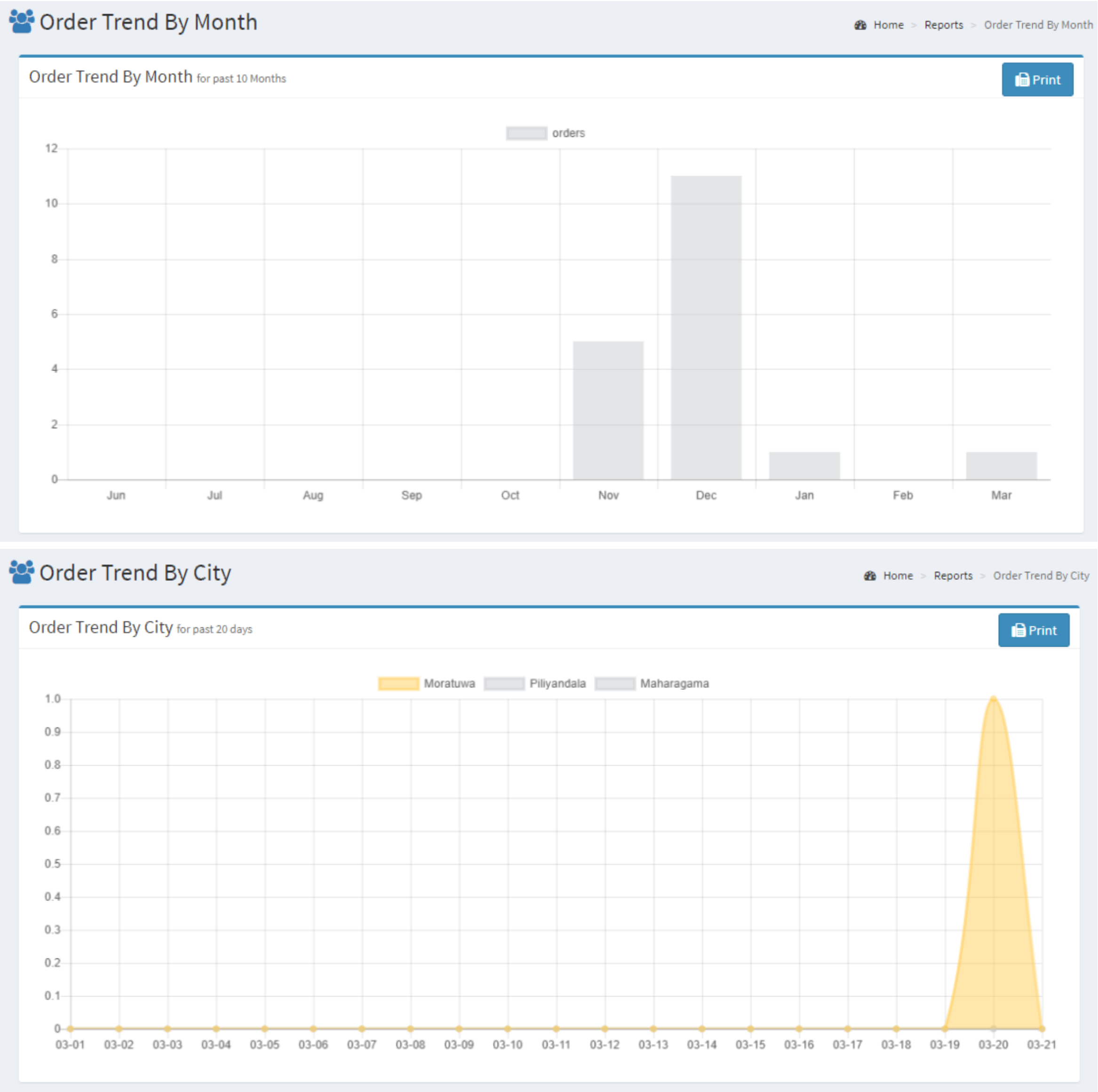
Courier ID	Name	City	Number of Orders	Total Amount (Rs.)
7	Ajith Nishshanka	Maharagama	1	450
8	Gayan Silva	Maharagama	1	400
9	Kumara Gamage	Moratuwa	5	5019
10	Sirimevan Somapala	Piliyandala	4	2240
11	test Courier	Piliyandala	0	

Showing 1 to 5 of 5 entries

Figure D2 Courier Performance Report

Order Trend Reports

Below charts (figure D3) shows the pattern of number of orders by month and by city. respectively.



Order Trend By Month

Order Trend By Month for past 10 Months

Order Trend By City

Order Trend By City for past 20 days

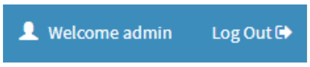
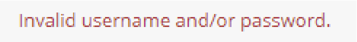
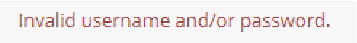
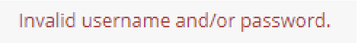
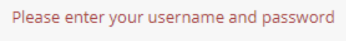
Figure D3 Order Trend Reports

APPENDIX E – TEST RESULTS

The results of the tests cases that were identified are given below.

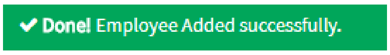
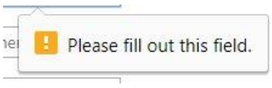
Test Results for User Authentication

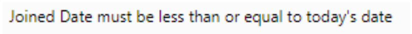
Table 15 User Authentication Test Results

Test No.	Test Case	Inputs / Steps	Expected Result
1	User input validation	Correct username Correct password	Successful login  Welcome admin Log Out
2	User input validation	Correct username Incorrect password	Error Message  Invalid username and/or password.
3	User input validation	Incorrect username Correct password	Error Message  Invalid username and/or password.
4	User input validation	Incorrect username Incorrect password	Error Message  Invalid username and/or password.
5	User input validation	Both fields empty	Validation error message  Please enter your username and password

Add Employee – User Management Module

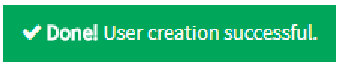
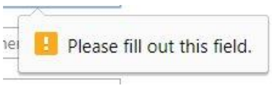
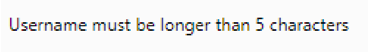
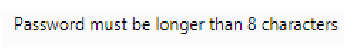
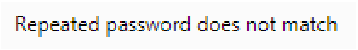
Table 16 Add Employee Test Results

Test No.	Test Case	Inputs / Steps	Expected Result
1	All fields must be filled	Fill all fields	Saving Successful  ✓ Done! Employee Added successfully.
2	All fields must be filled	One or more fields are not filled	Error Message  ! Please fill out this field.

3	Joined date is less than today's date	Set joined date to the day after	Error Message  Joined Date must be less than or equal to today's date

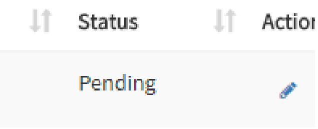
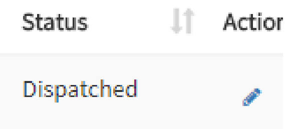
Add User – User Management Module

Table 17 Add User Test Results

Test No.	Test Case	Inputs / Steps	Expected Result
1	All fields must be filled	Fill all fields	Saving Successful  ✓ Done! User creation successful.
2	All fields must be filled	One or more fields are not filled	Error Message  Please fill out this field.
3	Username must be more than 5 characters	Username with 5 characters	No error message
4	Username must be more than 5 characters	Username with 2 characters	Error Message  Username must be longer than 5 characters
5	Password must be more than 5 characters long	Password with 10 characters	No error message
6	Password must be more than 5 characters long	Password with 5 characters	Error message  Password must be longer than 8 characters
7	Repeat password field must match the password field's value	Same password on both fields	No error
8	Repeat password field must match the password field's value	Different values on the two fields	Error message  Repeated password does not match

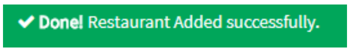
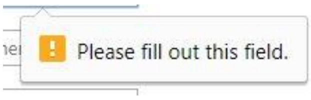
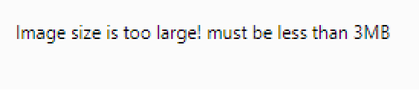
Order Management Module

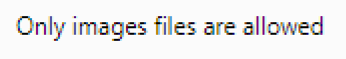
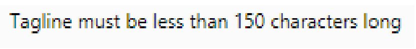
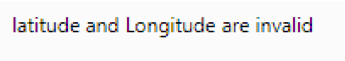
Table 18 Order Management Test Results

Test No.	Test Case	Inputs / Steps	Expected Result
1	New orders should have pending status assigned	Create a new order through website	Order status value is set to 'pending'  Status Actior Pending
2	When a courier is assigned to order, status should change to dispatched	Assign a courier to an order	After saving order status is set to 'Dispatched'  Status Actior Dispatched

Restaurant Management Module

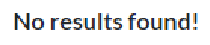
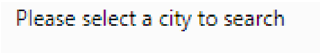
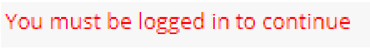
Table 19 Restaurant Management

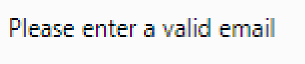
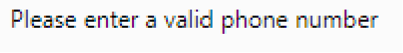
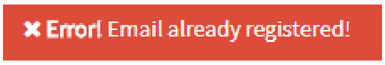
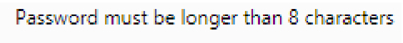
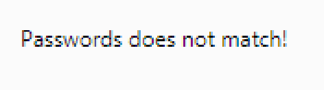
Test No.	Test Case	Inputs / Steps	Expected Result
1	Add new restaurant	Fill all fields	Saving Successful  ✓ Done! Restaurant Added successfully.
2	Add new restaurant	One or more fields are not filled	Error Message  Please fill out this field.
3	Add new restaurant	Logo/cover image must be less than 3 MB in size	Error Message  Image size is too large! must be less than 3MB
4	Add new restaurant	Logo/cover image is not an image file	Error Message

			 Only images files are allowed
5	Add new restaurant	Tagline is more than 150 characters long	Error message  Tagline must be less than 150 characters long
6	Add new restaurant	Latitude & Longitude are integer values	Error message  latitude and Longitude are invalid

Website Module

Table 20 Website Module Test Results

Test No.	Test Case	Inputs / Steps	Expected Result
1	Search restaurants	Non existing restaurant name	No results message  No results found!
2	Search restaurants	Empty search box No city selected	Error message  Please select a city to search
3	Search restaurants	Empty search box City selected	All restaurants in that city
4	Access user profile page	User is not logged in	Redirect to login page with error message  You must be logged in to continue
5	Access user profile page	User is logged in	Load page successfully
6	Sign up	All fields area filled	Signup user

7	Sign up	Email address is invalid	Show error message  Please enter a valid email
8	Sign up	Contact no is invalid	Show error message  Please enter a valid phone number
9	Sign up	Email already registered	Show error message  ✖ Error! Email already registered!
10	Sign up	Password is less than 8 characters long	Show error message  Password must be longer than 8 characters
11	Sign Up	Repeat password value is different to password field value	Show error message  Passwords does not match!

APPENDIX F – CODE LISTING

Below are some of the code samples that would be of interest to anyone trying to understand the system.

Code Related to Restaurant Search Function

Search box HTML code in ‘restaurant_list.php’ VIEW file

```
<!-- Start of search form -->
<form class="form-inline" method="GET" action="<?php echo base_url()
?>index.php/Restaurants/search">
      <div class="form-row">
            <div class="col">
                  Search for
            </div>
            <div class="col">
              <!-- this input holds the search query entered by the
                   user -->
              <input class="form-control" type="text" name="q"
placeholder="Restaurant Name"
                                    required="required" value="<?php
echo $this->input->get('q') ?>">
            </div>
            <div class="col">
            in
            </div>
            <div class="col">
              <!-- the available cities are taken from the database
                   and shown as dropdown -->
              <select class="form-control" name="c">
                  <?php foreach ($cities as $city): ?>
                    <!-- when the results are shown the selected city
                   will be 'selected' using the passed parameters -->
                    <option <?php if($this->input->get('c') == $city-
                        >name) echo 'selected' ?>
                        value="<?php echo $city->name ?>"><?php echo
                        $city->name ?>
                    </option>
                  <?php endforeach ?>
              </select>
            </div>

            <div class="col">
                <!-- submit button -->
                  <button class="btn btn-outline-primary">
                        <i class="fa fa-search fa-fw"></i>
                        Search
                  </button>
            </div>
      </div>
</form>
```

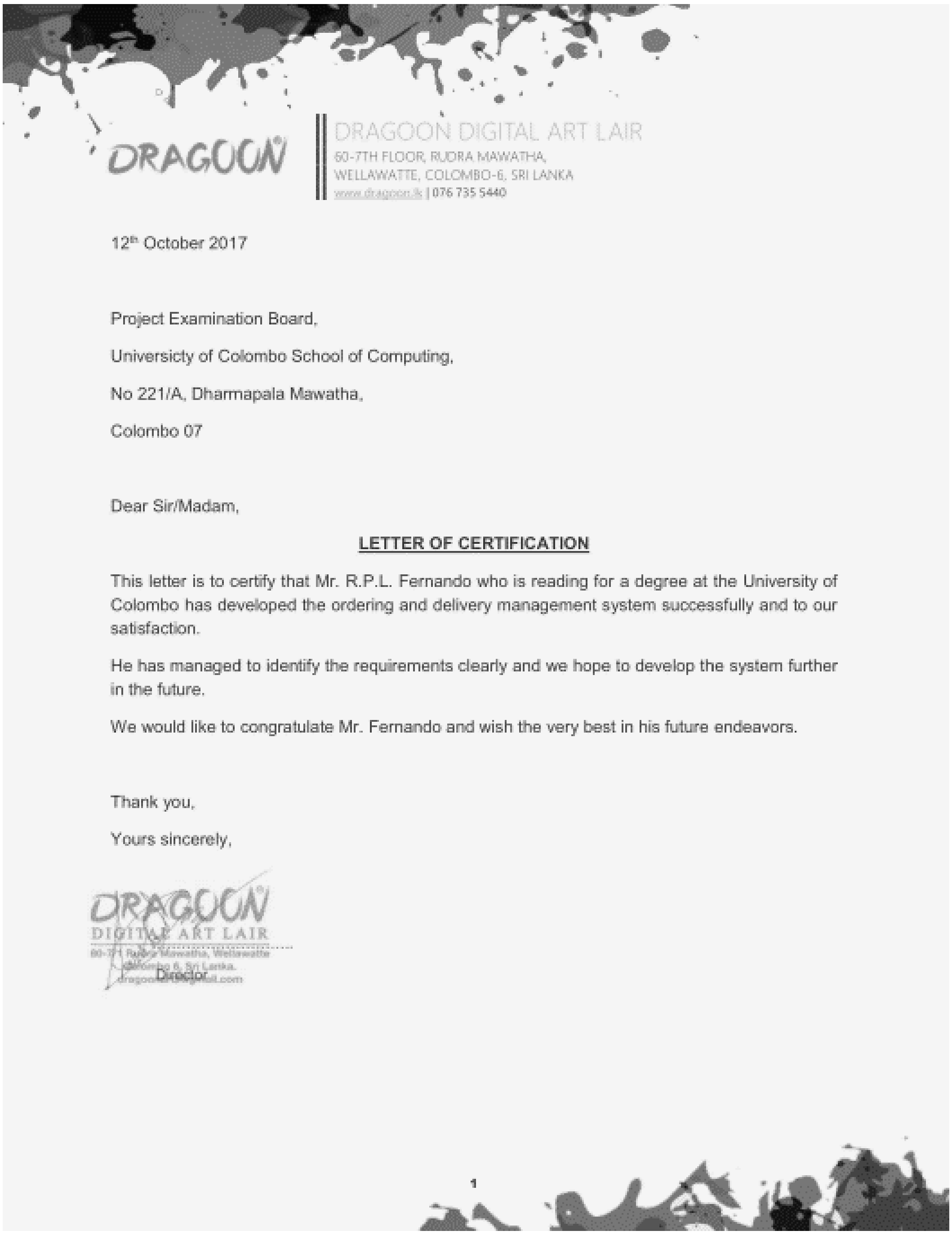
Search related function in the ‘Restaurants.php’ CONTROLLER file

```
public function search()
{
        // load the needed models
        $this->load->model('Cities_Modal');
        $this->load->model('Restaurants_Modal');

        // Get all GET data and pass them through XSS filtering
        $inputs = $this->input->get(NULL, TRUE);

        // get all cities to populate the dropdown
        $data['cities'] = $this->Cities_Modal->get_all_cities();
        // get the result of the query
        $data['restaurants'] = $this->Restaurants_Modal-
>search_restaurants($inputs);

        // pass the data to the views and load them
        $this->load->view('header');
        $this->load->view('restaurants_list', $data);
        $this->load->view('footer');
}
```

Search related function in the ‘Restaurants_Model.php’ MODEL file

```
public function search_restaurants($inputs)
{
        // we only need the restaurant's ID, name, logo path, cover
        image path, tagline and the city name
        $this->db->select('restaurants.id, restaurants.name, logo,
                        cover_image, tagline, cities.name as city');
        $this->db->from('restaurants');
        // we have to join with cities to get the name of the city
        $this->db->join('cities', 'restaurants.city_id = cities.id');
        // pass the input to LIKE keyword with % wildcard at the both
        sides
        $this->db->like('restaurants.name', $inputs['q'], 'both');
        // AND also select records in the selected city only
        $this->db->where('cities.name', $inputs['c']);
        // run the query and get the results
        $query = $this->db->get();

        // convert the results into an array of objects and return it
to      the caller
        return $query->result();
}
```

APPENDIX G – CLIENT CERTIFICATE



DRAGOON

DRAGOON DIGITAL ART LAIR
60-7TH FLOOR, RUDRA MAWATHA,
WELLAWATTE, COLOMBO-6, SRI LANKA
www.dragoon.lk | 076 735 5440

12th October 2017

Project Examination Board,

Universicty of Colombo School of Computing,

No 221/A, Dharmapala Mawatha,

Colombo 07

Dear Sir/Madam,

LETTER OF CERTIFICATION

This letter is to certify that Mr. R.P.L. Fernando who is reading for a degree at the University of Colombo has developed the ordering and delivery management system successfully and to our satisfaction.

He has managed to identify the requirements clearly and we hope to develop the system further in the future.

We would like to congratulate Mr. Fernando and wish the very best in his future endeavors.

Thank you,

Yours sincerely,

DRAGOON
DIGITAL ART LAIR
60-7/1 Rudra Mawatha, Wellawatte
Colombo 6, Sri Lanka.
dragoon[illegible].com

Director

GLOSSARY

SDLC – Software Development Life Cycle is the set of steps that a software development project will pass throughout its development

PHP – It is a scripting language designed to handle server side operations. PHP stands for PHP Hypertext Preprocessor

WAMP – WAMP is a software bundle that packs several software needed for web development

CSS – Stands for Cascading Style Sheets. CSS is used to apply styling to HTML documents

HTML – Hypertext Markup Language is a language used to structure documents in the world wide web

SQL – Structured Query Language is a language used to interact with database management systems.

MySQL – A popular open source database management system

JavaScript – A scripting language used to make hypertext documents more interactive. Runs on the web browsers.

UML – Unified Modeling Language is set of rules and symbols that is used to draw models of when analyzing a software system.

JQuery – It is a collection of commonly used JavaScript functions that enable quick development.

RUP – Rational Unified Process is a software development methodology

OOD – Object Oriented Design is a software designing technique that takes advantage of the Object Oriented Approach.

INDEX