

CHAPTER ONE

INTRODUCTION

1.0 BACKGROUND TO THE STUDY

Efficiency and effectiveness are the main concern of every business including restaurants. Restaurant managers and diners go through a lot of stress in terms of food ordering and table reservation leading to errors in the ordering and reservation processes hence, affecting the whole performance of the restaurant business. We find it necessary to come up with an application that would enable people either order food or reserve a table at a restaurant in an efficient and convenient way. We therefore came up with the idea to design and implement an online restaurant reservation system to enable restaurants manage their food orders and table reservation so that people will not join very long queues before being served their meal at the restaurant.

This chapter will introduce the project concept, outline the general objectives as well as the specific objectives that will eventually lead to the development of the project and the scope of the project. The chapter will also outline the timeline for the development of the project; specifying when each developmental phase will start and end, the limitations of the project and finally describe briefly what the subsequent chapters will entail.

1.1 PROBLEM STATEMENT

The challenges encountered by the existing system serve as a major drawback to the realization of efficiency and customer satisfaction. The experience of ordering in most fast food restaurants is not pleasant for the customers. Restaurant managers often record details of people who call them on telephone for food order and table reservation in ordinary books. This is tedious and leads to errors such as double booking in recording diner information. In other instances too, the restaurant will make arrangements and prepare the food, reserve tables that these diners have called them to do and guess what, some of these diners do not show up. Diners on the other hand often join long queues waiting to be served at restaurants which is frustrating and time consuming. Diners who also wish to make reservation also go through stress of calling such restaurants and this could lead to disappointment when their calls are not answered.

There are no proper reports on the food orders and table reservations and managers cannot make meaning out of these data recorded in books. There is no feedback to diners as to whether their orders were successfully placed either via SMS or email. Such problems are faced by restaurants today and it is our hope that by the end of this project, all these issues would be addressed and resolved.

1.2 MOTIVATION FOR THE PROJECT.

We remember very well when Mr. Frimpong Twum used to ask in Software Engineering class that *“if there is a problem with any system, who has called you to solve it?”* What he essentially meant was that, we cannot just get up one day and say we want to develop a system to solve a problem if there is nothing motivating us to develop that software or system. We are motivated to carry on and build this online reservation system because of the following reasons:

- The frustrations that diners go through in joining long queues in order to be served their meal at restaurants.
- The cost that most restaurants incur as a result of some diners not showing up to take their ordered meals or reserved tables since they have not paid. These diners do not lose anything for not coming for their orders.
- The money that we will get from leasing or selling our software to restaurants who needs the system.

1.3 GENERAL OBJECTIVES

The overall aim of this project is to design and implement an online restaurant reservation system that would enable diners order food and also reserve tables at restaurants to optimize food ordering and table reservation.

1.4 SPECIFIC OBJECTIVES

In order to accomplish the general objective stated above, the following will be our specific tasks to be done. By the end of these specific objectives, the application would be ready to be used by the end users.

- ❖ To research into the restaurant industry as to how food ordering and table reservation is done.
- ❖ To consider the various methodology available for the development of our proposed system and decide on those appropriate for our system.
- ❖ To gather user and system requirements by conducting interviews, observations and the use of questionnaires where necessary.
- ❖ To design the database and system's architecture.
- ❖ To code the system by mapping our database and system's design onto a physical platform.
- ❖ To test the system to ensure it satisfies the specified requirements.
- ❖ To finalize the documentation which will be ongoing as the above mention objectives are taking place.

1.5 PROJECT SCOPE

This project is aimed at targeting the restaurant businesses in Ghana who would like to manage reservations and customer details in an easy manner to enhance their business performance. After its successful implementation, the restaurants in the country are going to benefit as well as customers who will patronize these restaurants web sites.

The project will involve the development of a purely web based online application whose main programming language will be PHP and it will include a server side that enables a restaurant administrator to create an account and customize the software to the needs of the business. The administrator will then be able to perform basic managerial functions such as daily scheduling, approving orders, deleting menus, generating reports, etc.

The client side of this software will enable customers after creating an account with the restaurant to login and order food or reserve a table. The software will include a payment system to enable customers pay before their orders are placed successfully. In this way the restaurant will not run at a loss when a diner fails to show up.

This software application is not going to be a mobile application but then it will be optimized to be responsive on multiple displays such as a mobile devices with internet access. Thus, customers can order or reserve tables on their phones just like using the laptop or personal computer on the web.

1.6 DEFINITION AND EXPLANATION OF TERMS.

RESTAURANT: A place where meals are prepared and served to customers at the premises. People usually pay money to sit and eat their meals at these places.

DINER: This is any person who is eating a meal especially at a restaurant.

RESERVATION: This is the means by which a table is set aside at a restaurant for a diner. In this case the table has been booked in a customer's name within a time frame.

ORDER: This is a request to supply or deliver food booked by diners far in advance before due date and time.

BACK-END: This is a specialized subordinate process or a module that is not directly accessible by the user. This part of the application allows the administrator to interact with the software. This part of the software is where menu details that would be available to the user to place order is entered. Reports generation and basic managerial operations occur at the back-end. We usually refer to this part of the system as the server side of the software application.

FRONT-END: In software development, front-end is that part of the software that the user interacts with in performing his or her functions. It usually consists of Graphical User Interface which makes it easy for users to either login, place food order, reserve a table etc. It is sometimes referred to as the client side of the application

RESPONSIVE DISPLAY: This is an emerging technology whereby online web applications are developed in such a way that it can be displayed on every kind of output screen without changing its appearance and also enhances user friendly navigation to the various links within the system. The team is trying all we can to make this system responsive so that other displays such as tablets, mobile phones, etc. can access the same site and it will be optimized to display just like a mobile application and provide an interactive navigation property for the user.

PHP: It stands for Hypertext Preprocessor and it is a programming language used on the web platform for backend communication to the database.

1.7 PROJECT ACTIVITY PLANING

This project is expected to span a period of six months starting from the 14th October, 2015 to 30th March, 2016 by which we hope to have finished the project and its associated documentation.

This project is expected to go through these development phases before its completion:

❖ Analysis Phase

This phase involves studying the existing systems which will help in getting clearly the requirements for the proposed system. This phase is where the team will define the problem it intends to solve. We as a team will use the various techniques in information gathering to obtain user and system requirements.

❖ Design Phase

After the necessary requirements for the proposed system are gathered from the analysis phase, the design phase then sets in. This phase basically involves detailed specification of the proposed system. This phase is where most of the work will be done and it will include such things as coding and database design of the proposed system.

❖ Implementation and Testing Phase

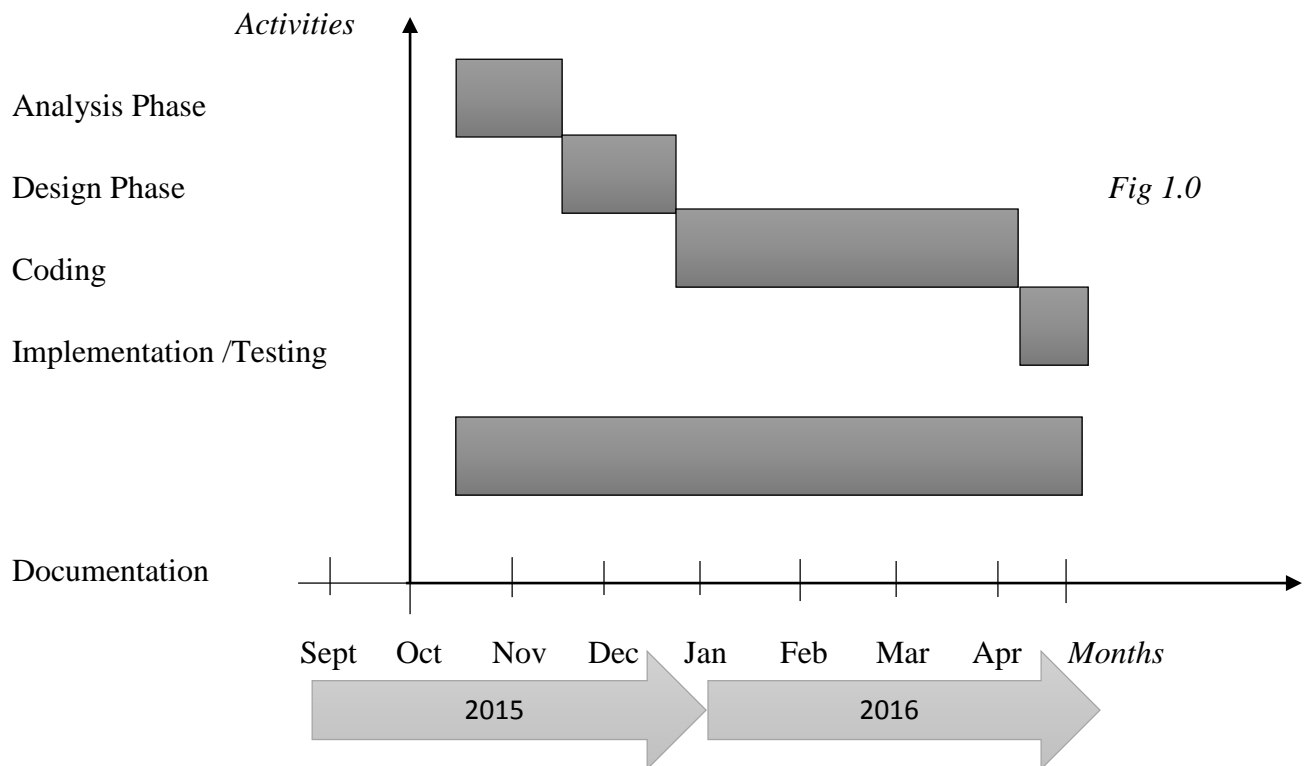
During this phase, the newly developed system is tested and implemented. In implementing this system, the actual system is installed on a selected vender where the various configuration will be set by the administrator. During the system testing, we shall talk about unit testing, Incremental integration testing, integration testing, functional testing and the whole system testing. This phase also includes associated documentation of the new system that would be delivered to our supervisor.

Table 1.0 below indicates the time interval for the development of the system.

Table 1.0

PHASE	TIME INTERVAL
Analysis	14 th October, 2015 – 14 th November, 2015
Design	23 rd November, 2015 – 23 rd December, 2015
Coding	1 st January, 2016 – 15 th April, 2016
Implementation	16 th April, 2016 – 30 th April, 2016
Documentation	14 th October, 2015 – 30 th April, 2016

1.8 GANTT CHART REPRESENTATION OF THE PROJECT ACTIVITY PLANNING



1.9 PERCEIVED PROJECT BENEFITS

At the end of this project, we hope to have acquired some benefits for the restaurants who use this software in their daily operations and it will include the following:

- ❖ To increase customer satisfaction by speeding up food delivery processes.
- ❖ To reduce time wasting by eliminating long queues.
- ❖ To be able to stand out from competitors by automating daily operations which will give food service providers the opportunity to increase sales.
- ❖ To reduce restaurants food wastage and increase efficiency of the restaurants' staff by enabling the restaurants' staff to know what food items the customers want in advance.

As a team developing this software, we also perceive the following benefits after successful completion of the project.

- ❖ To have enhanced our programming skills since this project will expose us to the use of some new programming techniques.
- ❖ To have better insight as to what a software system documentation really entails and how to document properly.
- ❖ To have developed better working ethics including being time conscious as well as developing the right attitude towards team work.

1.10 LIMITATIONS OF THE PROPOSED SYSTEM

- ❖ The application will require internet connection and also the user must be a computer literate. The restaurant has to incur debt in hosting their site online so that people can browse the site and place orders.
- ❖ The set back of the system is that the customers targeted are adults with access to computer systems and mobile phones with internet capabilities. People who cannot afford these resources cannot engage in the food orders and reservations.
- ❖ The other limitation is that the system will only be convenient to people with a small geographical region, basically just around the restaurant i.e. can only help a small area. Even though the restaurant will be on the internet and can be spotted by numerous customers, only those who are close to the restaurant will engage mostly with the restaurant.
- ❖ Time Constraint is also going to be a major challenge to the team developing this system. We will be doing everything possible to finish the project on time so as to submit the final work and documentation to our supervisor. Within this six (6) month of developing the system, there will be things that will happen along the way which will be inevitable. Going for lectures, writing mid-semester examinations, assignments and writing of examination will all interfere with the smooth development of this system and might cause delay in delivery of the final output.
- ❖ Because we want to build the system to be responsive, the team needs to acquire some tablets, android phones, laptops etc. during the system development in order to do the testing to see that it is actually working. All these hardware parts have to be either bought or borrowed from friends because we as a team do not have most of the equipment.

1.11 STRUCTURE OF REPORT

The following describes the structure of the report which will comprise of the contents of the various chapters in the system development.

Chapter one entails introduction to the proposed system, motivation for the proposed system, the general objectives as well as specific objectives to be achieved at the end of the project. It also consist of the project scope as to who the proposed system applies to as well as what the system can do and what it cannot do. This chapter consist of the project activity planning which details out the allocated time scheduled for each development stage. This include a Gantt chart giving a pictorial view of the project activities.

Chapter two introduces literature review where an overview of the subject area will be discussed. Past and present technologies that applies to the proposed system will also be discussed. This chapter will also include the review of existing implementations including highlights from similar vendors. The benefits and challenges of implementing the system will also be discussed. This chapter concludes with a summary of the chapter content.

In chapter three, the development methodologies and development tools that will be used in the design and implementation of the system will be looked at. A brief description of the various methodologies including the advantages and limitations will also be considered. We will also introduce the programming language we shall use in developing the backend as well as the frontend of the system.

Chapter four will explain the system's analysis and its design. This chapter will contain details of the various methods used in requirements capture and specification such as interviews, observations, questionnaires, etc. It will also include details on data driven applications. The proposed system's requirements specification that is, the event driven applications will be looked at in this chapter. This will take a critical look at the functional and non-functional requirements of the proposed system. Use Case modelling including the various diagrams

which will be used where appropriate will be explained. The proposed system's database design and its associated details together with the user interface will be looked at as well.

Chapter five will introduce the implementation of the system. It will talk about mapping logical design onto physical form as well the various construction involved. The results of testing the system will be explained in this chapter. Evaluation of the project will also be detailed out in this chapter.

Chapter six which happens to be the final chapter will consist of findings and conclusion. This will include summary of various problems faced in the development of the system, achievements and challenges, recommendations as well as enhancements that can be made to the system in future.

1.12 CONCLUSION

In conclusion, this chapter basically introduced us to the proposed system we wish to develop for the restaurants industry in Ghana. We have looked at the general objective as well as the specific objectives expected to be accomplished at the end of the project. We have also looked at the scope which basically talked about the inclusions and exclusions of the proposed system as well as who the system applies to. We also looked at the project activity planning where we introduced the various phases in our development. The limitations of the proposed system was also looked at in this chapter as well as using Gantt chart to represent the project activity and the time interval for the completion of the project. The chapter also talked about the structure of this report which details the content of the various chapters.

CHAPTER TWO

LITERATURE REVIEW

2.0 INTRODUCTION

Literature Review is a summary of previous research on a topic and can also be a part of a larger report of a research project. This chapter explains facts or statements which will be used as guidance in developing the system. It will include reviewing the subject area, past and present technologies, research issues that are currently available, general system architecture of existing and the proposed system, highlights of similar implementation from vendors, review of existing implementations, benefits and challenges of implementations and the trends in the industry or future directions for online restaurant reservation system.

2.1 OVERVIEW OF ONLINE FOOD ORDERING AND RESERVATION

Online food ordering is a process of ordering food from a local restaurant or food cooperative through a web page or an app. Much like ordering consumable goods online, many of these allow customers to keep accounts with them in order to make frequent ordering convenient. A customer will search for a favorite restaurant, usually filtered via type of cuisine and choose from available items, and choose delivery or pick-up. Payment can be amongst others either by credit card or cash, with the restaurant returning a percentage to the developers hosting the site. This means that many restaurants offer the technology to their customers to order food by means of their computer via browsers like Mozilla Firefox or via apps developed by individuals to order food.

2.2 CURRENT MARKET SHARES/TRENDS IN THE RESTAURANT INDUSTRY IN USA

For publications of the world's most comprehensive market research on full-service restaurant within the consumer food service market, Euromonitor International is regarded as the best

platform to get such information. They offer information on data and analysis, ranging from market sizes, market shares, industry trends etc.

According to Euromonitor International, growth in the takeaway and home delivery in food ordering in 2014 was strong but not as strong as in previous years in the United States of America (USA). From their executive summary report, 2012 and 2013 saw a current value growth rates of 6% and 4% respectively. From the summary, increased demand post-recession and strong expansion by chains such as Papa John, Domino's Pizza, etc. contributed much of the growth experienced in 2012. From 2012, economic recovery has matured and growth has normalized.

Also, in terms of competition, it is strongly contested by four brands in the food industry namely; Domino's Pizza, Papa John's, Pizza Hut and Little Caesar's Pizza. Domino's Pizza maintain a position at the top with 19% value share and sales of \$3.8 billion, and 15% of outlet numbers, with 4,986 locations in 2014. Papa John's held 13% value share with 10% total outlet during the same period. [<http://www.euromonitor.com/100-home-delivery-takeaway-in-the-us/report>, accessed on: January 18, 2016]

According to TechCrunch, [<https://en.wikipedia.org/wiki/TechCrunch>, Accessed on January 18, 2016] which is an online publisher of technology news primarily covering businesses ranging from startups to established firms, out of \$70 billion [takeout and delivery market], only about \$9 billion [roughly 13%] is online as of March 2015. It therefore means that online food ordering technology is not known to many food vendors and therefore not patronized by most of their clients. As technology advances, it is hoped that online food ordering will take up most of the restaurant markets.

2.3 TECHNOLOGIES AVAILABLE TO THE RESTAURANT INDUSTRY IN TERMS OF FOOD ORDERING AND RESERVATION

Technology touches every facet of our lives as it always has but with the emergence of mobile devices and cloud computing making more of an impact than ever, you'd be hard-pressed to go anywhere and not find a high-tech piece of hardware or software around. These advances have made an impact everywhere, and one of the places we are seeing more and more technology is in the food and restaurant industry. There are numerous technologies that were available in the past

that helped in the online ordering and food reservation. Currently there are also important changes in terms of the technologies that the restaurant industry and many food vendors employ in their operations. Between the arrivals of mobile devices on the table, online reservations, social media and new payment methods, technology has infiltrated the food and restaurant industry like never before. Some of the advances will serve to improve the experience both for the industry and for the patron. Yet with every new advancement comes a new challenge, and with technology moving faster these challenges can seem insurmountable. It is how the industry deals with these advances and the challenges that accompany them that will determine the fate of many restaurants, regardless of what is on the menu.

2.3.1 PAST TECHNOLOGIES THAT WERE EMPLOYED IN FOOD ORDERING AND RESERVATION

Until the advent of the World Wide Web (www) in 1989 by Sir Tim Berners-Lee, several customers used to go directly to restaurants to buy food and reserve tables. There were no known technologies available to have a dedicated websites where customers could order meals or reserve table on the internet. Even after the advent of www in the early 90s, still many vendors were just publishing their menus on the internet with no capability to process order online. Customers just browse through their online menus and may decide to visit the restaurant in person to get the meal.

[https://en.wikipedia.org/wiki/History_of_the_Internet, accessed on: January 18, 2016]

Also, most programmers were using procedural way of coding in the building of their websites. This means that these systems were built with no regard to reuse of codes. They were predominantly relying on imperative programming whereby the instructions written by the programmer tells the computer what to do step-by-step. It was intuitive for them to use in the sense that, the standard was similar to how they wanted the computer to respond to user inputs.

In terms of their database design, most vendors were using MySQL with PHP. The connection to the database, fetching records, and updating menu details were all done using [`mysql_connect()`] which is now obsolete. [<http://codular.com/php-mysqli>, accessed on: January 18, 2016]

In terms of how customers order and reserve food, they were using their personal computers only. The customers use their PCs to visit the vendors' websites and browse the menus. Most mobile phones, tablets and many portable displays were not available for food reservation as we have today.

2.3.2 PRESENT TECHNOLOGIES AVAILABLE IN FOOD ORDERING AND TABLE RESERVATIONS

Technology available today for developing online reservation system is purely Object Oriented Programming (OOP). In this approach, models are organized around objects rather than "actions" and data rather than logic. OOP employs classes which are blueprints from which we instantiate to get an object. A class say 'Order', can be instantiated to form a real world thing called 'Orders' that will represent a food order that customers make on a vendors website.

With regards to database connection with this present technology of developing web applications, the database connections are done with either mySQLi or PDO.

MySQLi (My Sequel Query Language improved) is a new way of connecting with the database. Now developers connect with `mysqli_connect()` with the *i* representing improved. PHP developers are being advised to desist from using just `mysql_connect()`.
[<http://php.net/manual/en/function.mysql-query.php>, accessed on: January 19, 2016]

PDO stands for *PHP Data Objects* and is described as a "lightweight, consistent interface for accessing databases in PHP". Although its name isn't that great, PDO is a lovely way of accessing your database in PHP.

PDO has advantage over mySQLi because the former can work on multiple databases like mysql, pgsql, psql, sqlsrv etc. whereas the latter works with only mysql.

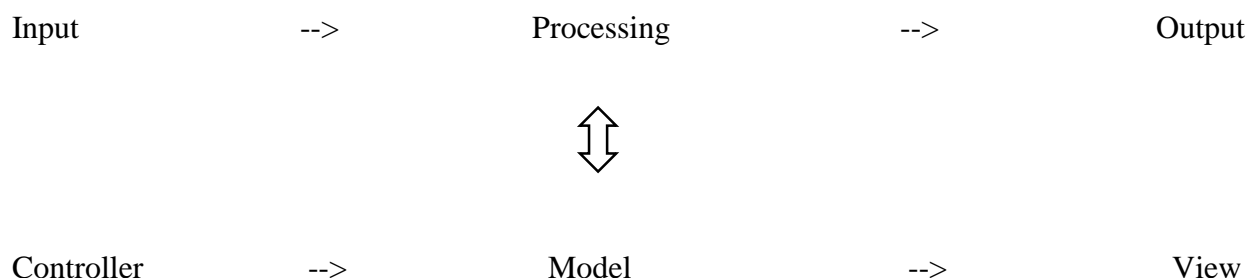
Lastly, mobile devices and tablets are currently available and some are even more powerful than personal computers. In view of this, most developers in the food ordering and table reservation industry today, build their applications to be mobile compatible so that users can browse menus, order meal, and reserve tables from their mobile phones and other handheld devices as though they are using a personal computer.

2.4 GENERAL SYSTEM ARCHITECTURE OF THE ONLINE FOOD ORDERING SYSTEM

Currently, developers of restaurant food ordering and reservations are shifting from procedural programming to Object Oriented Programming. The entire architecture has changed from just writing one's own codes with any particular programming language to what is called Frameworks. Developers now depend on Frameworks to build more robust and powerful applications for food vendors. There are several frameworks available in PHP like CakePHP, Code Ignitor, and Laravel etc.

PHP framework is a library that makes the life of site developers easier by hiding some complexities of HTTP protocol or by adding some useful functions. The most popular framework for web artisans today is Laravel which we will be using to build this particular project.

Laravel MVC paradigm is a way of breaking an application, or even just a piece of an application's interface, into three parts: the Model (M), the View (V), and the Controller (C). MVC was originally developed to map the traditional input, processing, output roles into the Graphical User Interface (GUI) realm:



MODEL

A model is an object representing data or even activity, e.g. a database table. The model manages the behavior and data of the application domain, responds to requests for information about its state and responds to instructions to change state. The model represents enterprise data and the

business rules that govern access to and updates of this data. Often the model serves as a software approximation to a real-world process, so simple real-world modeling techniques apply when defining the model. For instance, since our application will store customer details, there will be a model called User that will represent the table in the database for storing details of customers who visit our site or the manager who will manage the backend of the application.

VIEW

A view is some form of visualization of the state of the model. The view manages the graphical and/or textual output to the portion of the bitmapped display that is allocated to its application. Instead of a bitmapped display, the view may generate HTML or PDF output. The view renders the contents of a model. It accesses enterprise data through the model and specifies how that data should be presented. The view accepts data from the Model and displays it appropriately to the user via a user interface.

CONTROLLER

A controller offers facilities to change the state of the model. The controller interprets the mouse and keyboard inputs from the user, commanding the model and/or the view to change as appropriate. A controller is the means by which the user interacts with the application. A controller accepts input from the user and instructs the model and view to perform actions based on that input. In effect, the controller is responsible for mapping end-user actions to application responses. The controller translates interactions with the view into actions to be performed by the model. Since our application is web based, the controller can be thought of as responding to http get and post requests by the user. So, when a user visits our website to order food, once he/she clicks on a link on the homepage which is an http get request, Laravel will respond and route it to the appropriate controller to handle the request from the user.

[<http://laravelbook.com/laravel-architecture>, accessed on: January 20, 2016]

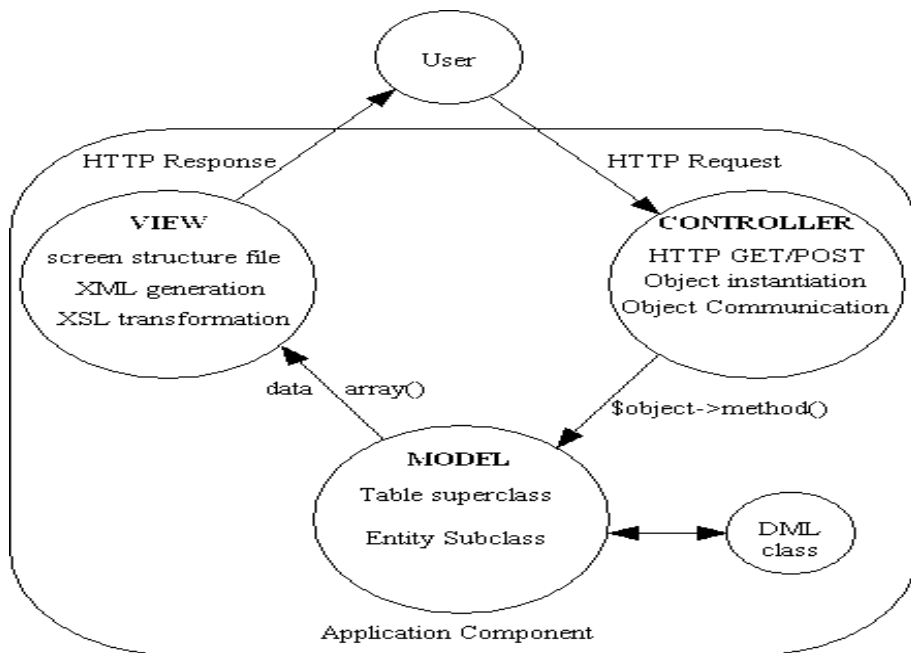


Fig 2.1 General System Architecture of the MVC pattern (1).

[<http://www.tonymarston.net/php-mysql/model-view-controller.html>, accessed on January 20, 2016]

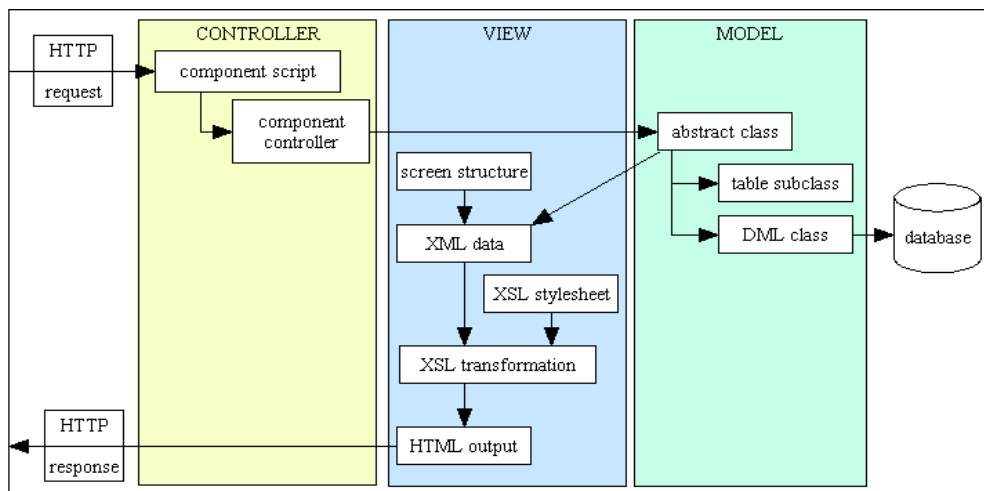


Fig 2.1 General System Architecture of the MVC pattern (2).

[<http://www.tonymarston.net/php-mysql/model-view-controller.html>, accessed on January 20, 2016]

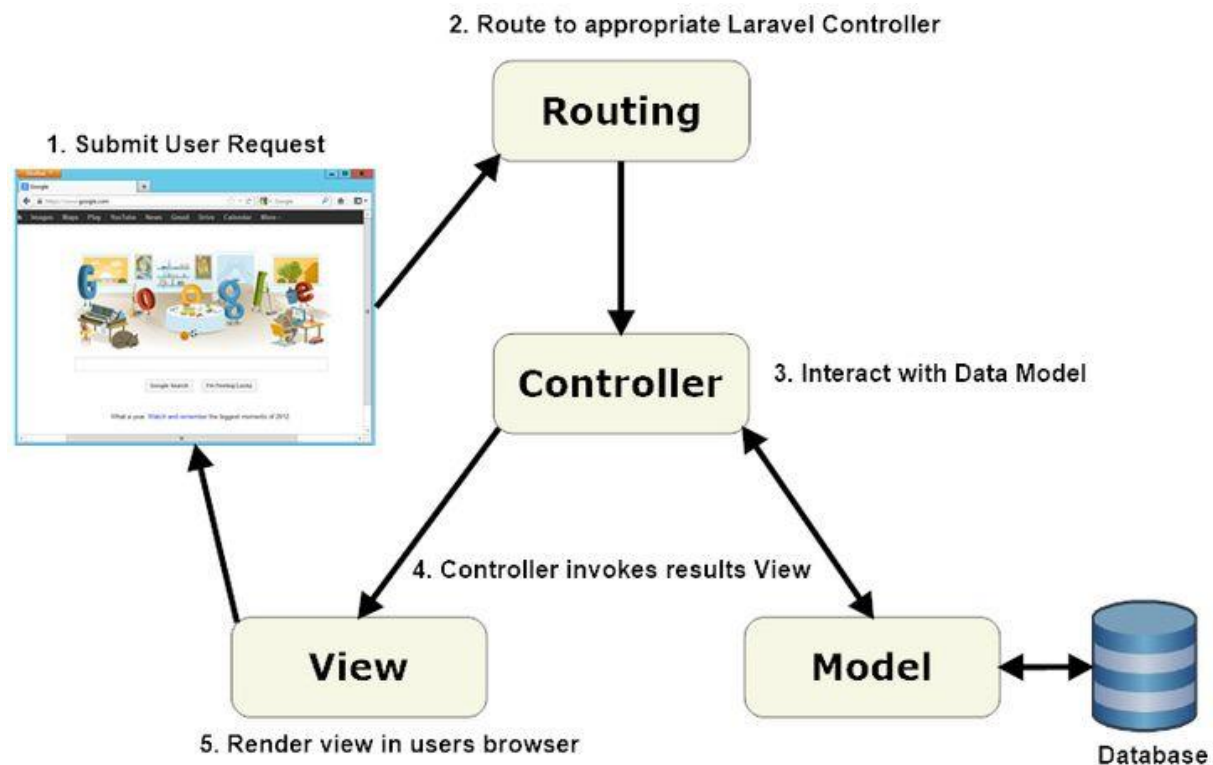


Fig 2.3 Main Components of the Laravel MVC Framework
 [http://laravelbook.com/laravel-architecture, accessed on: January 20, 2016]

2.5 HIGHLIGHTS OF SIMILAR IMPLEMENTATION FROM VENDORS

There are several implementations of online food ordering and table reservations that we would like to highlight from different vendors.

2.5.1 GRUBHUB

Grubhub Inc. is a mobile and online food ordering company that connects diners and corporate businesses with thousands of takeout restaurants in more than 700 U.S cities and London. Grubhub does not charge any fee from customers who visit the site to order food. In 2013, the company sent

more than \$1 billion in gross food sales to local takeout restaurants, processing roughly 174,000 orders on daily basis.

[https://en.wikipedia.org/wiki/GrubHub_Seamless, accessed on: January 22, 2016].

Grubhub works by allowing the customer to enter his/her address or location into a search text box. After searching through the database, it returns with a list of restaurants that are near the locality of the customer so that the user can make order. Customers have the option of searching by restaurant name, cuisine, or menu item. Customers click on a restaurant and proceed with the food order. He/she selects the time and date for the order and then submit the form for further processing. Customers pay for their meal at delivery or when they arrive at the restaurant to pick up their meal. [<https://www.grubhub.com>, accessed on: January 22, 2016]

The following figures depict how Grubhub works when a user initiates food order request.

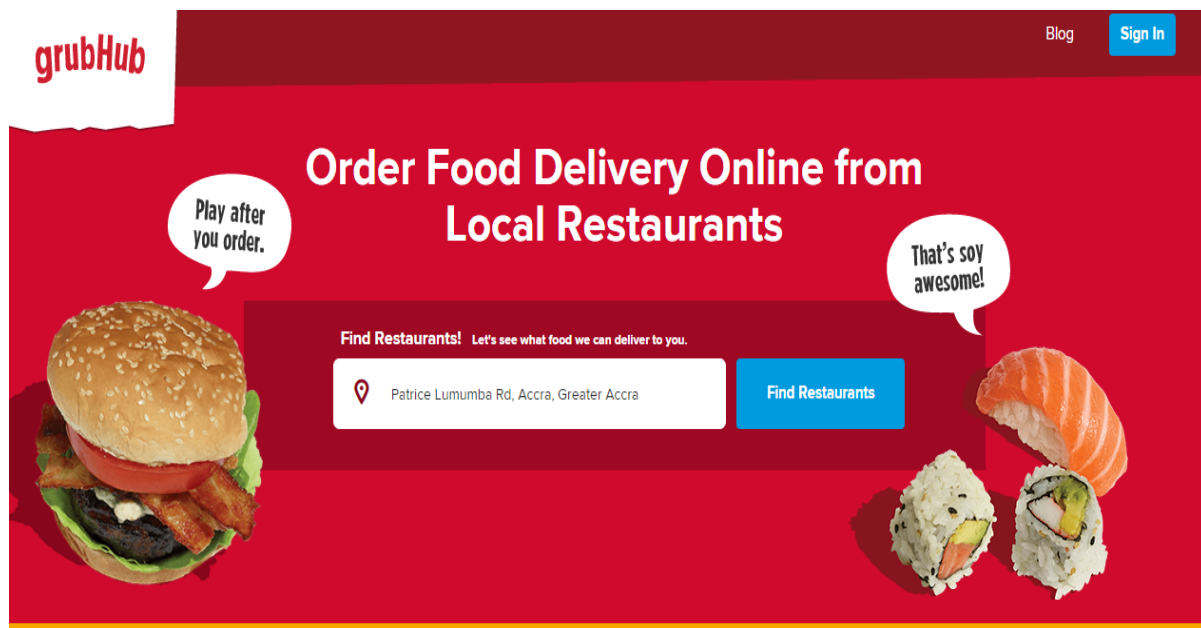


Fig 2.3 Homepage of Grubhub where users can enter their location.

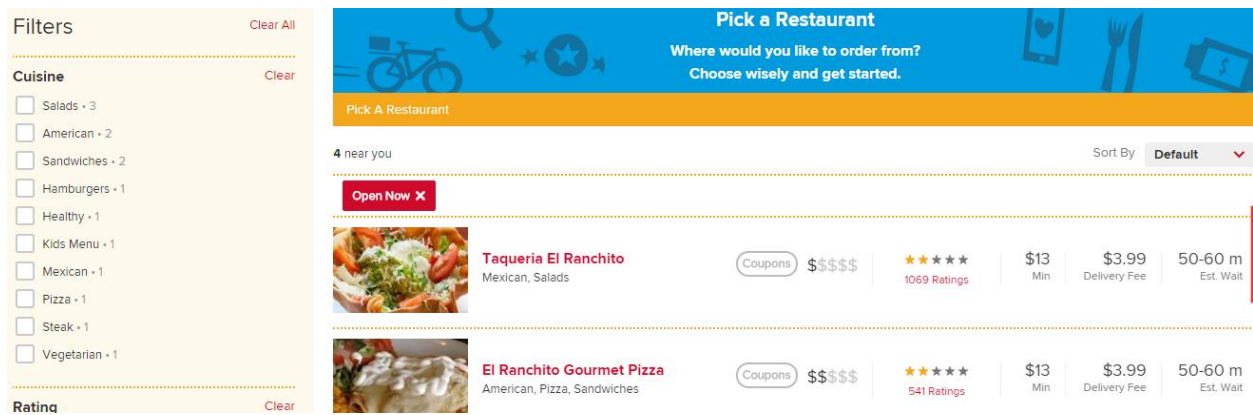


Fig 2.4 Lists of all restaurants available at Grubhub based on user location entered.

Menu Sections	
MOST POPULAR	
<p>Vegetable Omelette \$9.95 +</p> <p>Tomato, green pepper, onion and mushroom. Made with three extra large eggs and served with your choice of two sides.</p>	<p>35th Street Omelette \$10.95 +</p> <p>Avocado, bacon and cheddar cheese. Made with three extra large eggs and served with your choice of two sides.</p>
<p>Skirt Steak and 2 Eggs \$17.95 +</p> <p>Served with your choice of hash browns, O'Brien potatoes, sliced tomatoes or fruit and toast or pancakes.</p>	<p>Ribeye Steak and 2 Eggs \$17.95 +</p> <p>Served with your choice of hash browns, O'Brien potatoes, sliced tomatoes or fruit and toast or pancakes.</p>
<p>Bridgeport Skillet \$10.95 +</p> <p>Ham, bacon, sausage, onion, mushroom and American cheese topped with choice of two eggs and a side.</p>	<p>Full Order Biscuits and Gravy \$6.75 +</p>
<p>"Hay Stack" \$9.95</p> <p>A full order of biscuits and gravy on a plate of hash browns topped with two eggs.</p>	<p>Side of Bacon \$3.50</p>
<p>Milk Shake \$3.75</p>	

Fig 2.5 Bridgeport Restaurant menu lists and their prices

2.5.2 OPENTABLE

OpenTable sells online tools and services for restaurant reservations and reviews. The company went public in 2009. OpenTable basically does two things;

- Sell restaurants tools to manage reservations.
- Operates an online reservation service, both on its site and through partner sites.

OpenTable do not charge consumers who visit their sites but rather have percentage on every reservation that one makes at a particular restaurant. Restaurants pay a **one-time fee for installation and training costs** (which ranges from \$200 to \$700, according to the company), and a **monthly subscription fee** also applies (minimum \$199 per month, with extras costing \$25 to \$89 a month). OpenTable also gets a **fee for each diner** who completes his/her reservation: \$1.00 per diner for reservations made through OpenTable's website or mobile app, and \$0.25 per diner for reservations made through the restaurant's website using OpenTable software.


[<http://www.businessinsider.com/opentable-explainer-2011-10>, accessed on January 20, 2016]


From Fig 2.6, Customers enter the number of persons that need tables, the date, time and the location of the customer in the text fields provided. The system then searches their database for restaurants available.

Fig 2.7 in the next page will bring all the restaurants available and the times that tables can be reserved. A customer clicks on any of the available times and proceeds to the final stage.

Fig 2.8 depicts the final stage that the user enters some personal information if he is a guest on the site so that the order can be placed. Confirmation of successful order placement will be sent to the person's email.

[<http://www.opentable.co.uk/start/home>, accessed on January 20, 2016]




Choose your location


Sign up

The UK's number one restaurant booking website

1 person

24 Jan 2016


09:30


Location or Restaurant

Find a Table

Fig 2.6 *Homepage of OpenTable where users enter details of their order*

11 tables available for 1 person on Sun, 24 January, at 9:30




The Atrium Restaurant at
The Gonville Hotel

Reviews coming soon

£££

12:00



CAU Cambridge

★★★★☆


(1165)

£££

9:15

9:30

9:45



The Chequers of Orwell


★★★★★

(12)

£££

12:00

Offer



The Dining Room at
Paddocks House

★★★★☆

(36)

£££


9:15

9:30

9:45

Fig 2.7 *Restaurants available on OpenTable and the times that reservation can be done.*

Complete your reservation

	GUESTS 1 person	DATE Sun 24 Jan 2016	TIME 9:30	RESTAURANT The Dining Room at Paddocks House
---	--------------------	-------------------------	--------------	---

Already a member? [Sign in](#)

First Name		Last Name	
Phone Number	Mobile <input type="checkbox"/>	United Kingdom +44 <input type="checkbox"/>	
Email			
Add a special request (optional)			

Is this your first visit to this restaurant? ☐ Yes ☐ No

☒ Yes, I want to receive email messages from this restaurant.

☒ Sign me up to receive exclusive dining offers and news from OpenTable.

POINTS
Not a member? Join OpenTable and receive 100 points upon dining.

Fig 2.8 Final stage of OpenTable reservation where few user details are taken

2.5.3 TABLE-BOOKING.COM

Table-Booking.com is the worldwide, multilingual, interactive, real-time, table-reservation system where you can find the best and most amazing restaurants worldwide and make a guaranteed table booking very easily. The guest chooses the desired date and time at their favorite restaurant, clicks on the preferred table for the desired number of guests and simply reserves it. One interesting feature about table-booking.com is the fact that it allows guests to make a lot of their own choices like sitting inside or outside of the restaurant, whether they want to be around where diners smoke or not. The table-reservation system is part of a comprehensive web 2.0 portal which covers all gourmet fields for those who enjoy eating out! Restaurants have the opportunity to place adverts on their portal, which help them to market other products the restaurants may engage in other than table reservation. Table-booking.com also provide for e-card (i.e. a digital

version of a greetings card, typically accessed by the recipient via a hyperlink in an email) and a photo gallery.

[<https://www.table-booking.com/novumibiza.booking>]

Fig 2.9 and Fig 2.10 shows how a guest reserve a table at novumibiza.booking which is a restaurant's website under table-booking.com.

The screenshot displays the reservation interface for Novum Ibiza. The sidebar on the left contains links for 'Table-Booking', 'View Gallery', 'Change Bookings', 'Location Map', 'Show Events', 'Send E-card', 'Join Newsletter', and 'Restaurant Info'. The main reservation area includes a calendar for January 2016, a time selector set to 20:00, a guest count of 7, and checkboxes for 'Inside', 'Outside', 'Non-smoking', and 'Smoking'. A large photograph of the restaurant's outdoor terrace is shown. To the right, a promotional image features a woman with the text 'I like when a restaurant knows my preferences!'. The bottom of the page has a row of international flags and a 'SHARE' button.

Fig 2.9 Homepage of novumibiza.com where users enter their reservation details.

NOVUM
IBIZA ULTIMATE DINING AND ENTERTAINMENT

Personal data ...

Smoking: Smoker
Area: Outside
E-Mail *: kwadwotutu@yahoo
First name *: Joseph
Surname *: Anto
Cellphone *: 0249396348
Comment: Hoping that my reservation will be done on time

* needs to be filled out correctly, since the restaurant might need to contact you prior to the reservation.
To protect the restaurant from "fun-bookings" your IP-address will be stored. (41.66.208.33)

Finalize Booking!

whenever i travel, i save my favorite restaurants.
whenever i come back i find them again.
table-booking.com

Fig 2.10 Final stage of the table reservation from novumibiza.com

2.6 REVIEW OF EXISTING APPLICATIONS FROM VENDORS

Upon careful analysis of the above applications from different vendors, we came out with the following findings;

- We realized that the food ordering and the table reservation are mostly done in the advanced world. The technology is not much grounded in Africa and for that matter in Ghana. The ones we found in Ghana were not purely online food ordering. Customers just call the restaurant and ask for food availability so that the restaurant may bring it to them at an agreed location or better still the customer visiting the venue for the meal.
- We also found that with respect to table-booking.com, customers have a little flexibility of selecting where they would like to have their meal. Those who do not smoke obviously are more concern about their health and then would like to be distanced from those who enjoy their meal and smoke.

- All the vendors we highlighted above either offered table reservation or food ordering but not both.
- We also realized that customers do not make any payment prior to the food ordering or table reservation. This means if a customer does not show up for the meal, the restaurant runs at a loss.
- All the vendors discussed above allowed multiple food vendors to register on their site.

2.7 BENEFITS OF IMPLEMENTATION OF THE ONLINE RESTAURANT RESERVATION SYSTEM

Our main aim is to address some of the shortfalls of the existing systems and to provide a more robust and reliable service for restaurant industry. The system will ensure the following:

- Provide a system that would be used by a single vendor due to the time constraints needed to finish this project. The advantage here is that the restaurant has the flexibility to customize the system to suit the needs of its business.
- Provide a platform for restaurant industry in Ghana to go global so that they can increase their patronage by consumers. This has an added advantage of increasing the revenues of the restaurant industries in Ghana that use our system.
- Provide platform for food vendors to get their monies before successful order is placed. In this part of our world, if people do not pay before the order is placed, some might end up not showing up for their orders and the restaurant may incur debt. We hope to employ a mobile money payment platform to enable customers pay before their orders are placed successfully.
- Provide a platform whereby customers can order food and also reserve tables at the same time which is not available in the ones highlighted above. In this case, with the same vendor, one can reserve a table or order food.

2.8 TRENDS IN THE RESTAURANT INDUSTRY REGARDING ONLINE FOOD ORDERING.

As stated earlier, customers basically order food or reserve tables by means of their PCs, mobile devices as well as other portable devices such as tablets. Customers order food using computers via browsers like Mozilla Firefox or via custom apps. This basically is the main trend with regards to food ordering and table reservation in the restaurant industry.

In spite of consumer demand, restaurants' use of technology remains in its infancy hence, the use of technology in the restaurant industry is expected to gain dominance and recognition in the future.

2.8.1 THE CASE FOR ONLINE ORDERING IN THE RESTAURANT INDUSTRY

Online ordering systems are growing in popularity amongst both restaurants and consumers. They have been associated with increased revenue, improved capacity management, improved productivity, improved transactional marketing, and customer relationship management. While they increase return on investment (ROI) and customer loyalty for restaurants, they also offer consumers an easy, convenient way to order their food. Here are five cases within the usage of these innovative systems as interpreted from the 2011 Cornell Hospitality Report, *The Current State of Online Food Ordering in the U.S. Restaurant Industry*:

- 1. Consumers of pizza restaurants, followed by sandwich places, Mexican restaurants, and Asian restaurants, most frequently use online ordering systems:** Restaurants that have a specific type of food on their menu most frequently implement online ordering systems. Their menus are more defined, not overwhelming customers when they go to order online.

2. **The usage of online ordering by consumers was most prevalent in fast-casual restaurants, followed by casual restaurants and quick-service restaurants:** Restaurants that are known for their quick service as opposed to a leisurely sit down meal tend to use online ordering systems more. The fast service and ease of online ordering matches the service a customer would receive at a more casual restaurant.
3. **The most common reasons customers use online ordering systems are for convenience, speed, order accuracy, ease of use, and credit card acceptance:** Having a quick and easy way for customers to place orders drives them to order more from you, especially when they can do it from the comfort of their home!
4. **Online ordering generates incremental revenue by increasing order frequency for takeout, delivery, and catering and increasing order volume and service:** Restaurants using online ordering systems reported receiving more frequent orders and increases in group and catering orders due to the ease and convenience of placing an order online.
5. **Online promotions associated with online ordering work better than regular promotions:** Online ordering can provide restaurant operations with key customer information that can be used to develop targeted customer promotion strategies. Catering to your costumers makes their experience more personal and keeps them coming back to your restaurant!

2.9 SUMMARY

In this chapter, we started with the overview of the subject area which is online food ordering and reservations. We proceeded with current research issues in the restaurant industry with online food ordering and reservation. We got some interesting statistics from the various restaurants and fast food vendors from United States of America (USA). We then took a look at the technologies available both past and present, the general system architecture of the proposed system and the Laravel framework and later reviewed and highlighted similar implementations from three vendors

namely Grubhub, OpenTable and Table-booking.com. We went further to look at the benefits of our implementation and finally concluded with some cases for online ordering in the restaurant industry from 2011 Cornell Hospitality Report.

CHAPTER THREE

PROJECT METHODOLOGY

3.0 INTRODUCTION

In this chapter we will take a close look at the project methodology as well as the development tools that we will employ in developing this project. We will describe our selected methodology as well as state the advantages and limitations of our selected methodology and also identify which restaurant will be used as our case study.

3.1 DEVELOPMENT METHODOLOGY

A software development methodology in Software Engineering is a framework that is used to structure, plan, and control the process of developing a software product or an Information System. The most popular development methodologies available are the Plan-Driven and the Agile Method. The software process model associated with the Plan-Driven methodology is the Waterfall model while the Incremental model is associated with the Agile Method.

Plan-Driven method is a development approach whereby the requirements of the system are specified far in advance before development start. For applications like safety-critical control systems where a complete analysis of the system is essential, a plan driven approach will be the best approach.

Agile Method is mostly employed when the user requirements are not completely specified due to the changing business environment.

For this project, since we would like to engage with our users because their demands keep on changing, we will use **Agile Method** with **Incremental software process model** to develop this application.

3.1.1 INCREMENTAL DEVELOPMENT

This is a development process model whereby software engineers develop software as a series of versions, exposing it to user criticism and comments and then adjusting the system to suit the needs of the user. In our case, since the online food ordering and table reservation is a web application and hence user requirements may change with time, we will release the software as a series of increments to our supervisor, get his comments and suggestions and adjust the system and release subsequent versions until the final version is accepted by the supervisor (user).

In incremental development, specification, development and validation are interleaved but not separated and also there is rapid feedback across activities. There is no detailed system specification and design documentation is minimized or generated automatically by the programming environment used to implement the system. The user requirements document only defines the most important characteristics of the system. One important aspect of incremental development is that the system's user interfaces are often developed using interactive development system that allows the interfaces to be developed more quickly and may generate web-based interface for web browsers or an interface for a specific platform such as Microsoft Windows.

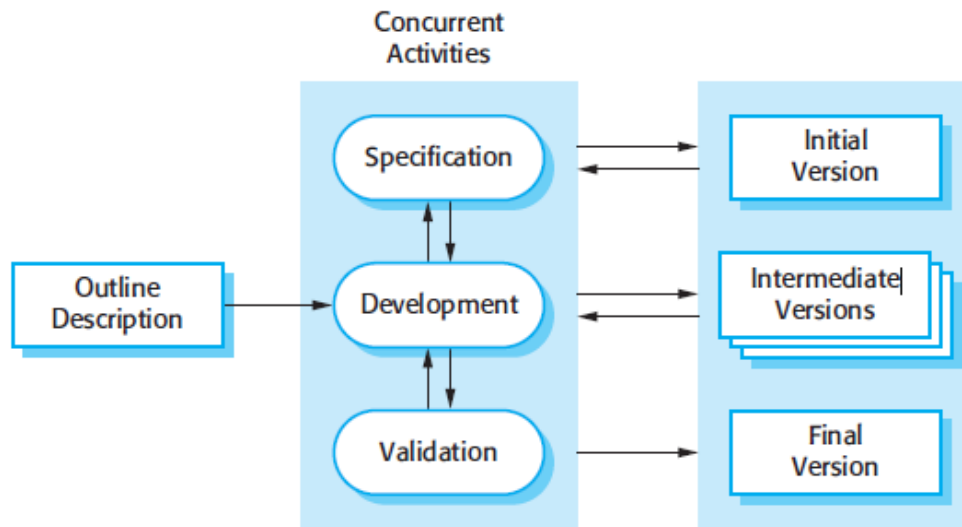


Fig 3.1 Various Activities/Phases of the Incremental Development Process Model

[Lan Sommerville, Software Engineering 9th Edition]

- **Software Specification:** This is the process of understanding and specifying what services are required from the system and identifying the constraints on the system operation and requirements. Software Specification also known as Requirement Engineering is a very vital activity under the incremental development since any errors that occurs at this stage could lead to very severe consequences in the development and implementation stage. Personal experiences, interviews, observations, questionnaires are some of the ways that we will use to get the requirements of the user and to obtain a *requirement document* which will serve as an official document to be used for the rest of the activities. Four main activities are carried out at this stage;
 1. *Feasibility study:* This is where a research is carried out to see whether the proposed system could be carried out under existing budgetary constraints and also whether the system will be accepted by the user. We have carried out the feasibility study and we are certain that the project can be done.

2. *Requirement elicitation and analysis*: This is where we obtain the requirements of the user through observations, interviews and review of similar implementation from vendors.
 3. *Requirement Specification*: This is where we translate the information gathered during the requirement analysis phase into a document (Requirement Document). We shall have **User Requirement** which is an abstract statement of the system requirement of the customer and the end users of the system and **System Requirement** which will be a more detailed description of the system functionalities.
 4. *Requirement validation*: Here, we check the requirement document to ensure realism, consistency and completeness and to ensure that the requirement is free from any ambiguity.
- **System Development**: This stage also known as the implementation stage is where we convert the system specification above into executable system. Here, we shall use a programming language to code the actual system. At this stage too, we may refine any software specification as and when they appear. With incremental development, the process of producing the various increments or versions occurs here. Four main activities will occur at this stage;
 1. *Architectural Design*: The overall architecture of the system (Modules to be implemented) as well as the relationships between modules and how they are distributed are determined here.
 2. *User Interface Design*: Here, we will use integrated development environment to design the various user interfaces for the application. The customer's registration will have a user interface so is the menu selection.
 3. *Component Design*: Here we will take each component of the system and describe how it operates.
 4. *Database Design*: Since we will be using a new database, we will be designing the system's data structures and how they will be represented in a relational databases. The various tables within the system and the relationship between them would be determined here.

- **Software Validation:** This which is generally referred to as Verification and Validation (V &V) is the last stage in the incremental development. This stage ensures that the developed system conforms to the specification document described above and also meets the end user expectations. This stage is where we test the application with dummy data to ensure that it is producing the expected results. We will perform unit testing to ensure each module is working correctly and system testing to ensure that the overall system requirement is met. Any errors that occurs as a results of integrating the various system modules will be addressed before deploying it to the end-users (supervisor).

** The whole process of specification, development and validation will undergo various iterations until the software is finally accepted by the end users (supervisor). Any recommendations or suggestions from the supervisor is factored into the next iteration until the supervisor accepts the final release.

However there are advantages and limitations of the incremental development that we need to take a look at.

3.1.2 ADVANTAGES OF INCREMENTAL DEVELOPMENT OVER WATERFALL MODEL

- The cost of accommodating changing customer demands is reduced. The advantage is that it avoids or reduces rework. Because we engage the end user (supervisor) in the release of the increments or versions, the changing demand of the user (in this case the supervisor) is taken care of until we develop the full system and then the supervisor rejects it for a new work. In the case of Waterfall, a new system will have to be made if customer/users reject the entire application.
- End users can give better feedback on actual work rather than speculations. With incremental development, because the first release is available to the user, they can give

genuine and factual feedback based on the current system rather than depending on the requirement document in the case of the Waterfall model.

- It is possible to deliver applications more rapidly and quickly to the customer. Even if the full system functionality is not implemented in the initial release, users can use the system and enjoy all the benefits before a new version is released to cover all the functionalities. This is not possible in waterfall model since the system can also be given to the user until all the functionalities and requirements are specified far in advance.
- It is flexible and less expensive to change user requirements and scope of the project in the incremental development than in the other process models such as the waterfall model.
- Incremental development is easy to debug and test during smaller iterations. Testing and debugging smaller modules and units within the application is much easier than in the waterfall model where testing is done for the whole application.

3.1.3 LIMITATIONS OF THE INCREMENTAL DEVELOPMENT PROCESS MODEL

- The total cost in the incremental development may exceed the total cost in waterfall model. This is because as new versions are released, the organization may bear additional cost as they suggest new system functionalities to the next release. Until the final release, the cost incurred may be higher than the waterfall model.
- As new increments are added in the various releases, it may tend to degrade and disrupt the system structure. More time and money is needed to bring the structure of the system back to perfection.

3.3 PROJECT CASE STUDY

Restaurants in Kwame Nkrumah University of Science and Technology (KNUST) campus will be our target beneficiary for this project even though the application can be customized by any restaurant in the country to suit their operations. KNUST has a number of restaurants on campus and off campus including Best in Town, Acuzi Eating Palace, Helenus Foods, and McKeon Catering Services etc. Since our system would be for a single vendor, we chose ***Best in Town Restaurant*** to be our case study. This is because it is close to us in terms of proximity and also widely patronized by students on and off campus.

Best in Town Restaurant is located at Attah Mills' junction on the Ayeduae road, opposite Victory Towers hostel. Best in Town deals mainly with taking food orders and delivery although one can eat at their premises as well. They provide customers most of which are students of KNUST, with continental dishes such as fried rice, jollof rice, French fries, among others. It has circulated its telephone numbers across a number of hostels and this serve as the main means by which it receives orders. Customer details often required by the restaurant include hostel name and room number for a successful delivery. Best in Town Restaurant currently provides only continental dishes but have plans to provide local dishes in the course of time.

3.4 DEVELOPMENT TOOLS

In this project, we will use a number of development tools to help us complete this project. Since we are building a web-based application, it is obvious that we have to implement the **client/server architecture**.

Client/Server architecture is a network architecture whereby each computer on the network or process is either a client or a server. **Servers** are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers) or network traffics (network servers). **Clients** are PCs or workstations on which users run applications to access information from the server. In our case, the entire application will be hosted on a server with all cost borne by the restaurant authorities. The customer who is a client will now have to access the information via

the Personal Computers (PCs), tablets, mobile devices etc. The users will either send an HTTP Get/Post request to the host (server) for food order or table reservations and the server will respond accordingly to the calls from the numerous clients scattered all over the world. We have already described the general architecture of the proposed system in chapter 2 so we shall take a very close look at the various tools and Application Programming Interfaces (APIs) that would help us in developing the software product to the end user.

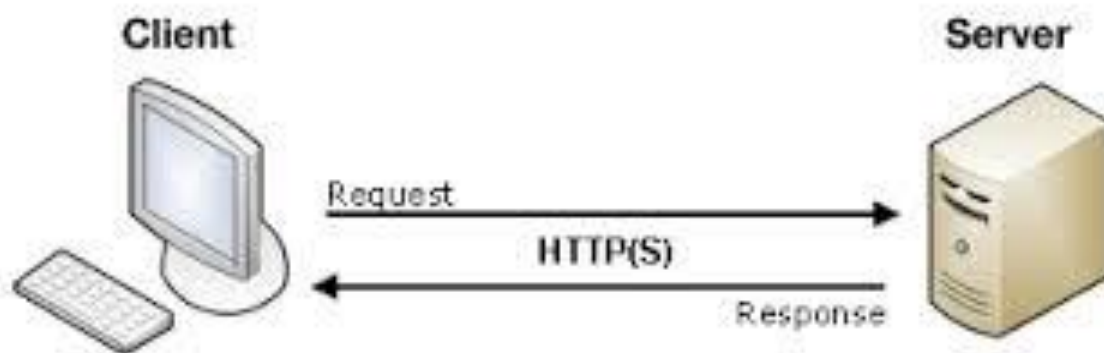


Fig 3.2 Client/Server Architecture.

[https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/Sending_and_retrieving_form_data, accessed on February 20, 2016]

3.4.1 VARIOUS DEVELOPMENT TOOLS FOR BUILDING THE APPLICATION

- **PHP (Hypertext Preprocessor)**

PHP is widely used as open source server-side scripting language developed for web development but also used as general-purpose programming language for other environment. As of January, 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers. [<https://en.wikipedia.org/wiki/PHP>, accessed on: February 20, 2016]

PHP is usually embedded into a Hypertext Markup Language (HTML) source document and interpreted by a web server with a PHP processor module that create the web page document. However, despite its popularity, since it is an open source language managed by different groups of people, there are no written specification or standard for the language. The canonical PHP interpreter serves as the factor standard but work on creating a formal specification has already started in the year 2014. PHP also has powerful command line interface capabilities to support standalone graphical applications. The language has the following advantages;

- ✓ *Simple and easy to learn:* As compared to other scripting languages, it is easier for developers to grasp the syntax of PHP since it is similar to that of C and Java. With very little knowledge of C++ and Java, one can learn the language and embark on any web application project with confidence.
- ✓ *Support:* PHP is very popular in the web development environment. There are various online resources, documentations, blogs, PDFs and social media available to developers to use when they encounter any problem. The fact that it is an open source language deepens the assurance of large community support with better and innovative ways of maintaining the language.
- ✓ *Cross-platform:* PHP is cross-platform and therefore is able to work on various operating systems. It works very well with LINUX. MACOS, LINUX, Microsoft Windows, etc.
- ✓ *Frameworks:* Almost every benefit of PHP seems to go back to the fact that it has a larger community. The number of PHP frameworks available to developers includes; Zend, CodeIgnitor, Symfony, Aiki, Laravel etc. Frameworks help to make development easy since it helps in database access libraries, session management, code reuse etc. For the purpose of this project, we shall use Laravel framework which we will discuss shortly.

Laravel is a PHP and web application *framework* with expressive, robust and elegant syntax providing powerful tools for large applications. It has a superb inversion control container, expressive migration system, and a tightly integrated unit testing support that gives developers

tools to build any web application they are tasked. We are using laravel framework due to the following reasons:

1. **Code and File Organization:** With laravel framework, there is a standard that one needs to follow to keep the codes and various files and folders organized in a particular order. This makes code reading easier as well as putting files in correct folders. There is no need to create a “functions” file and an “include” file in all the pages, the organization and the structure is organized to factor in all these.
2. **Utilities and Libraries:** Using PHP as a scripting language alone can be cumbersome and tedious in some instances. Laravel framework comes with certain libraries and functions straight out of the box to help developers in session management, form validation, database abstraction, sessions and cookies handling etc. This, if used in the recommended way makes lives of web developers much easier.
3. **The MVC Pattern:** As described in chapter 2, laravel framework support the Model, View Controller paradigm and hence makes it more seemingly object oriented. Models will represent the data structures (database), Views will represent the various user interfaces and the Controllers will respond to user requests (Both http get and post).
4. **Security:** Laravel helps to protect data integrity and consistency. There are various techniques that are employed to avoid malicious attacks of data. Forms’ inputs are all guarded against SQL injections. Also there is a property called *protected \$fillable()* which helps to prevent mass assignment vulnerability form data submitted by a user via forms.
5. **Suitable for Teamwork:** Laravel makes it easier for people to work together as a team. This is because the entire project can be shared among a development team with the designers working on the Views since that is where users interacts, the database experts working on the models etc. All these experts do not work in isolations but depend on each other for successful completion of the project.

- **MYSQL**

MySQL is the most popular Open Source Relational SQL (Sequel Query Language) database management system. MySQL is one of the best RDBMS (Relational Database Management System) being used for developing web-based software applications. All the tables and records that would be required in the project would be designed and created with MySQL. Our choice for MySQL is that it is easy to use, support is easily available on the internet, it is open source and hence inexpensive to get and finally it remains one of the most accepted industry standard database for developing web applications.

- **APACHE WEB SERVER**

The Apache Http server is an open source web server application developed by Apache Software Foundation. In our project, we shall use XAMPP [**X** stand for any Operating System, **A** stands for Apache server, **M** stands for MySQL database, **P** stands for PHP and the last **P** stands for Perl programming language] which is an open source cross-platform web server solution stack package developed by Apache Friends. It contains all the essential tools and services for building a web application and since we shall be hosting the application locally on our laptops, this server is critical because without it there would be no way we can develop the application. We chose Apache web server because it is portable, reliable and flexible as compared to the other web servers.

- **HTML (HYPERTEXT MARKUP LANGUAGE)**

HTML is the most basic building block of a webpage used for creating and visually representing a webpage. It is the globally accepted programming language for formatting web pages. It is mostly used by small and medium scale businesses that do not really need advanced functionality on their websites. HTML is free, supports all browsers on the

client's machine, easy to use and understand hence, our choice in building the structure of our web pages.

- **JAVASCRIPT**

It is a scripting language designed to be executed on the client side to add interactivity to web pages. It is used alongside HTML to help client side scripts interact with the user, control the browser and alter the content of the displayed document on a website. We shall also use it in conjunction with laravel to ensure validation rules on our websites. **JQUERY** is a another JavaScript code that we will use to increase our productivity by enabling us achieve critical user interface functionality by writing very small amounts of codes and also producing dynamic web pages with flash-like animations. JQUERY is easy to use and support Ajax functionality. Ajax enables a user interface to perform actions without requiring the whole page to be reloaded. Since JavaScript is an open source language, there are various plugins of JQuery available on the internet to make our work easier.

- **PHOTOSHOP**

Adobe Photoshop is the predominant photo editing and manipulation software on the market. Its uses ranges from full featured editing of large batches of photos to creating intricate digital paintings and drawings that mimic those done by hand. It is a graphic designing tool that enables picture manipulation and editing. We shall use Photoshop to design user interfaces and also the various images and animations that would be required in developing the system.

- **CSS (CASCADING STYLE SHEETS)**

This is a stylesheet language used to describe how a webpage should look. It is used for formatting an HTML document to be rendered in a browser. It's main job is to tell the various web browsers how to display a page to the user and also how certain user agents respond to this pages. It is consistent and also compatible with all known web browser.

- **BOOTSTRAP**

Twitter Bootstrap is the most popular front end framework currently. It is sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and JavaScript. Bootstrap toolset includes pre-developed HTML and Cascading Style Sheets (CSS) templates for forms, buttons, charts and navigation. We shall use Bootstrap to design the styling of our application alongside CSS. Bootstrap is important in our application for the following reasons:

- ✓ **Speed of Development:** Bootstrap makes the work of developers much easier by providing access to mostly used code blocks in our development rather than starting to code from the scratch anytime we need such codes. Moreover since the time frame within which we expected to deploy the application is limited, we should employ any tool that will speed up our work progress.
- ✓ **Responsiveness:** The need to develop responsive websites is becoming very important in this era. As smart phones continue to dominate the market, there is the need to develop applications that could run on mobile phones with ease. In our case for example, we want customers to order food even from their phones, therefore bootstrap will enable us to help build the application in that regard.
- ✓ **Support:** Bootstrap has a large support community that would help and address some of the problems we might face in the course of developing this system.
- ✓ **Excellent Documentation:** Bootstrap also come with self-explanatory documentation that one can refer to in their project. This documentation comes in a PDF format which is easy to read and understand.
- ✓ **Customizable:** Twitter Bootstrap gives the developer the flexibility to pick and choose the features that are needed in their development and discard the rest. This means developers can customize bootstrap to suit what they want.

- **SUBLIME TEXT 3 IDE**

This is a cross-platform and source code editor with Python Application Programming Interface. Sublime text is a sophisticated text editor for code and markup. It is the integrated development environment that we will use to write the executables of the PHP and HTML files. It is a very good editor and we have chosen it because it allows developers to locate files very quickly and easily just a few key strokes, Multiple selections allows developers to interactively make changes to several lines of codes all at once, it allows developers to switch between several projects in the workspace, it is cross-platform and can be used on any operating system, and finally it allows key bindings, menus, snippets, macros and many more.

3.5 SUMMARY

We started this chapter with project methodologies where we identified two main methodologies as plan-driven and agile method. We went further to describe software process model and we chose the incremental development process model under the agile development as the methodology to be used in this project. We identified the main activities of the incremental approach as *specification*, *development* and *validation* and also made it clear that the software would be released as a series of versions to the supervisor until the final product is accepted. Again we took a close look at the benefits of the incremental development over other process models especially the waterfall model.

We further identified our case study as Best in Town and briefly described their operations. Lastly, we analyzed the development tools we will use to build our project. We delved much into the

Laravel framework and discussed five strong reasons why we will use it to develop our application as well as the other tools such as PHP, MySQL, Apache Web Server, HTML, CSS, Adobe Photoshop, JavaScript etc.

CHAPTER FOUR

SYSTEMS ANALYSIS AND DESIGN

4.0 INTRODUCTION

System analysis is a process of collecting factual data, understanding the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. System design is a phase where the new system is designed based on the user requirements and the detailed analysis of the existing system. It is the most crucial phase in the development of a system since the logical system design arrived at as a result of systems analysis is converted into physical system design.

This chapter will outline the requirements specification, functional and non-functional requirements as well as use case modelling which will include some of the unified modelling language (UML) diagrams. The database design E/R diagram and Normalization, and user interface design will also be considered.

4.1 REQUIREMENT SPECIFICATION

A system requirements specification is a comprehensive description of the intended purpose and environment for the system under development. What the software will do and how it will be expected to function is fully described under requirements specification. It outlines functional and non-functional requirements and may include a set of use cases that describe user interactions that the system must provide. Requirements specification allows a thorough analysis of requirements before design can start hence, reducing later redesign.

4.1.1 FUNCTIONAL REQUIREMENTS

Functional requirements describe in detail a system's intended capabilities, appearance and interactions with users. It serves as a kind of guideline and continuing reference point as the system is being developed. The functional requirements for a system usually involves the user interface

and describe each of the possible user input actions and the system's response actions. Functional requirements therefore specify particular results of a system and it drives the application architecture of a system.

Our system will have the following functional requirements:

- ❖ An administrator will be able to login and customize the system to the needs of the restaurant.
- ❖ The administrator can also input menu details on the site.
- ❖ Diners will be able to create an online account with the restaurant so as to login to their portal.
- ❖ Diners will be able to place orders.
- ❖ Diners can reserve tables online using an interactive Graphical User Interface
- ❖ Diners can also pay online before their orders are placed successfully
- ❖ Customers will receive an SMS and Email confirmations.
- ❖ Through the administrator's panel, the system allows the manager to generate reports, see daily schedule, delete menus, and approve menus among others.

4.1.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements essentially specify how a system should behave or work and also are a constraint upon the system's behavior. They specify criteria that judge the operation of a system rather than specific behaviors. They also describe various attributes which affect the functionality's effectiveness. Non-functional requirements do not alter a system's functionality that is, the functional requirements remain the same regardless of the attributes attached to them. Non-functional requirements make up a significant part of the specification in that, users or clients may judge a system based on its non-functional requirements.

Below are the non-functional requirements our system will achieve:

- **Performance**

The system will be an interactive one hence the delays involved will be less in that there are no immediate delays in every action-response of the system.

- **Reliability**

As the system is meant for food ordering and table reservation, the system will be reliable in that it will consistently perform according to its specification so as to enable users make orders and reserve tables as and when they want to without any system failure.

- **Maintainability**

A system should be developed in such a way that it can evolve to meet the changing needs of a customer. Our system will be developed in such a way that it can be customized to meet a particular vendor's needs or preference.

- **Ease of use**

The system will ensure ease of use in that little training time will required to know how to use the system. The user interface will also be friendly so users will have ease when using the system.

- **Scalability**

It is the ability of a system to continue to function well when it is changed in size or volume to meet a user need. The system will be developed to fit onto any device such as a smartphone, laptop, tablet or desktop.

- **Security**

It is a system attribute that reflects the ability of a system to protect itself against external attacks, which may be deliberate or accidental. The system will be developed in such a way that proper login mechanism would be put in place to protect user accounts.

- **Safety**

The system will ensure safety so that information about orders and reservation is securely transmitted to the server without any changes in the information. Also Mpower Payment System is recognized internationally for its secure environment for payments online.

- **Efficiency**

An efficient system does not waste system resources such as memory. The system will therefore ensure fast response time, proper memory and processor cycles utilization and fast processing time.

- **Hardware Requirements**

The hardware requirements are classified into server and client requirements which are detailed below:

Server Requirements

Minimum CPU – Intel® processor Pentium IV minimum

Minimum Disk Space – 40GB

Minimum Memory – 2GB

Client Requirements

Minimum CPU – Intel® processor Pentium IV 1GHz

Minimum Disk Space – 512MB

Minimum Memory – 256MB

- **Software Requirements**

The software requirement for the application runs best on a system with the following capabilities:

- Any operating system such as Windows, Linux, Macintosh, and Mozilla Firefox operating system
- A web browser such as Mozilla Firefox, internet explorer, opera, Google chrome etc.

4.2 USE CASE MODELLING

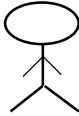

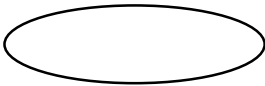

UML stands for Unified Modelling Language. They are diagrams to help aid the creation, visualization and documentation of various aspects of the software engineering process.

A use-case model is a model of how different types of users interact with the system to solve a problem. It therefore describes the goals of the users, the interactions between the users and the system and the required behavior in satisfying these goals.

4.2.1 USE CASE DIAGRAMS

A use-case diagram is used to graphically depict a subset of the model to simplify communications. It shows a subset of the model elements relevant for a particular purpose. Below is a table of use case symbols and their representations.

Table 4.0 Use case symbols and their meanings.

SYMBOL	TERM AND REPRESENTATION
	Actor: An actor represents roles for users of a system, including human users and other systems. An actor is external to a system.
	Association: An association correspond to a sequence of actions between the actors and use case in achieving the use case.
	Use case: A use case represents a user goal that can be achieved by accessing the system.
	System Boundary: The system boundary defines the confines or scope of a system within which the use cases are placed

The actors involved with our system are:

Customer: This actor is one of the main beneficiary of the system. The customer can order food as well as reserve tables using the system.

Manager: This actor is also another beneficiary of the system. Customizing the system to his preference, editing menus, approving reservations are some of the activities of this actor.

System: This actor is responsible for generating reports and sending confirmation messages among others. It however requires the administrator inputs to generate the reports.

Below are the use case diagrams for our system together with their descriptions:

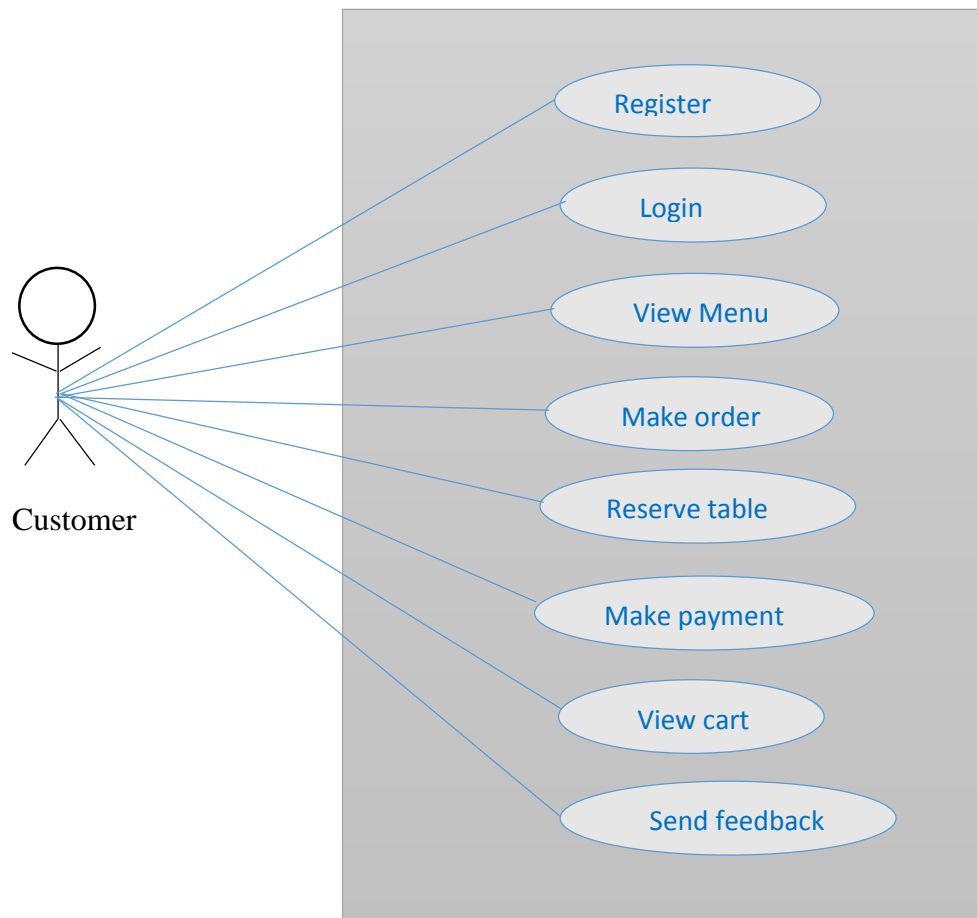


Figure 4.0 Use case diagram for customer

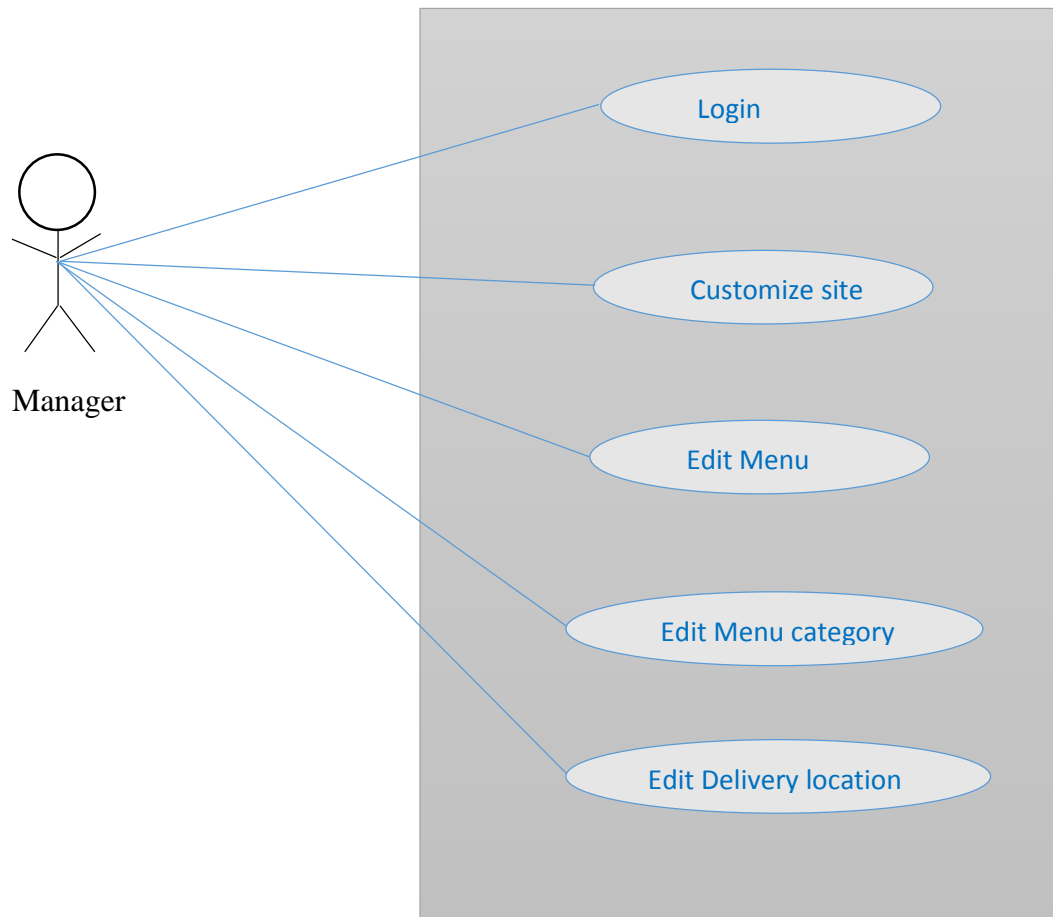


Figure 4.1 Use case diagram for manager

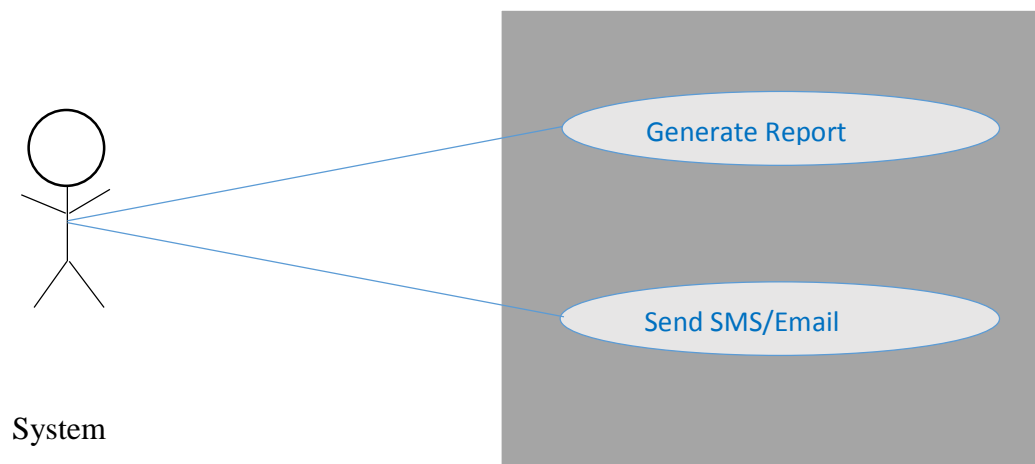


Figure 4.2 Use case diagram for system

Use Case Scenario

Table 4.1 Customer Registration

Use Case Name	Register
Actor	Customer
Description	Enables new customers to create an account by entering their credentials.
Steps Performed	<ol style="list-style-type: none">1. Click on sign up2. Enter personal details and password.3. Click on sign up button.

Table 4.2 Customer login

Use Case Name	Login
Actor	Customer
Precondition	Customer should have already registered.
Description	Enables existing customers to have access to the system's functionality using valid credentials.
Steps Performed	<ol style="list-style-type: none">1. Click on login2. Enter email and password.3. Click on login.

Table 4.3 View Menu

Use Case Name	View Menu
Actor	Customer
Description	Enables a customer to view the menu available together with their prices.
Steps Performed	Go to the website of the vendor using a web browser. You can also view menus after login to the dashboard.

Table 4.4 Make Order

Use Case Name	Make Order
Actor	Customer
Precondition	Customer should login.
Description	Enables customers to order food which is successful after they have proceeded with the payment.
Steps Performed	<ol style="list-style-type: none"> 1. Enter quantity 2. Click on add menu button to add menu to cart. 3. Click on checkout. 4. Enter confirmation and payment details and click on Order now. 5. Proceed with payment using either credit card or mobile money.

Table 4.5 Table Reservation

Use Case Name	Reserve table
Actor	Customer
Precondition	Customer should login
Description	Enables customers to make a table reservation.
Steps Performed	<ol style="list-style-type: none"> 1. Enter number of persons who will need a table. 2. Enter the date and time for which the reservation will be needed. 3. Add comments concerning the reservation if any. 4. Click on reserve table.

Table 4.6 Manager Login

Use Case Name	Login
Actor	Manager
Description	Enables a manager to access the system's functionality available to perform managerial duties.
Steps Performed	<ol style="list-style-type: none"> 1. Click on login. 2. Enter email and password. 3. Click on login.

Table 4.7 Site Customization

Use Case Name	Customize site
Actor	Manager
Description	Enables managers of various vendors to customize the site to suit their preference.
Precondition	Manager should login
Steps Performed	<ol style="list-style-type: none"> 1. Click on customize site. 2. Enter details about the restaurant including its name, address as well as uploading the restaurant's logo. 3. Click on customize site.

Table 4.8 Editing Menu

Use Case Name	Edit Menu
Actor	Manager
Description	Enables a manager to edit available menus such as adding new menu, deleting menu not available as well as upload images for the various menu.
Precondition	Manager should login.
Steps Performed	<ol style="list-style-type: none"> 1. Click on menus. 2. Make changes to the menu by adding or deleting.

Table 4.9 Edit Menu Category

Use Case Name	Edit menu category
Actor	Manager
Description	Enables the manager to group menu based on its category such as continental or local dishes.
Precondition	Manager should login.
Steps Performed	<ol style="list-style-type: none"> 1. Click on categories 2. View all categories. 3. Make the necessary changes by adding or deleting.

Table 4.10 Edit Delivery location

Use Case Name	Edit Delivery location.
Actor	Manager
Description	Enables the manager to view locations to which delivery is possible and make updates where necessary.
Precondition	Manager should login.
Steps Performed	<ol style="list-style-type: none"> 1. Click on delivery locations. 2. Click on view all locations. 3. Make the necessary updates.

Table 4.11 Report Generation

Use Case Name	Generate Report
Actor	System
Description	Generates reports on periodic basis.
Steps Performed	<ol style="list-style-type: none"> 1. Click on report on the manager's interface. 2. Enter the period, that is start and end date desired for report.

Table 4.12 Send SMS/Email

Use Case Name	Send SMS/Email
Actor	System
Description	Sends order confirmation to the customer and order details to the vendor via SMS and Email.
Steps Performed	When an order is successful after payment, the system automatically sends an email and SMS to the customer and vendor.

4.2.2 SEQUENCE DIAGRAM

Sequence diagrams are the most common kind of interaction diagrams that shows how actors and objects interact to realize a use case scenario. It focuses on the message interchange between a number of lifelines. We normally draw a sequence diagram if we have a use case, to describe how the main components of the system interact. Then again sequence diagram helps us identify messages arriving at an interface of a component, to describe how the internal parts of the component interact.


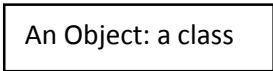




SYMBOL	TERM & MEANING
	An actor: Is an entity or system that derives benefit from and is external to the system. Participates in a sequence by sending and/or receiving messages.
	An object: Participates in a sequence by sending and/or receiving messages.
	A lifeline: Represents the life of an object during a sequence.
	A focus of control: Is a long narrow rectangle placed on or above a lifeline. Denotes when an object is sending or receiving messages.
	A message: Conveys information from one object to another.
	Object destruction: An X is placed at the end of an object's lifeline to show that it is going out of existence

Table 4.13 Sequence diagram symbols and their meanings

Sequence Diagram for Customer Use Case

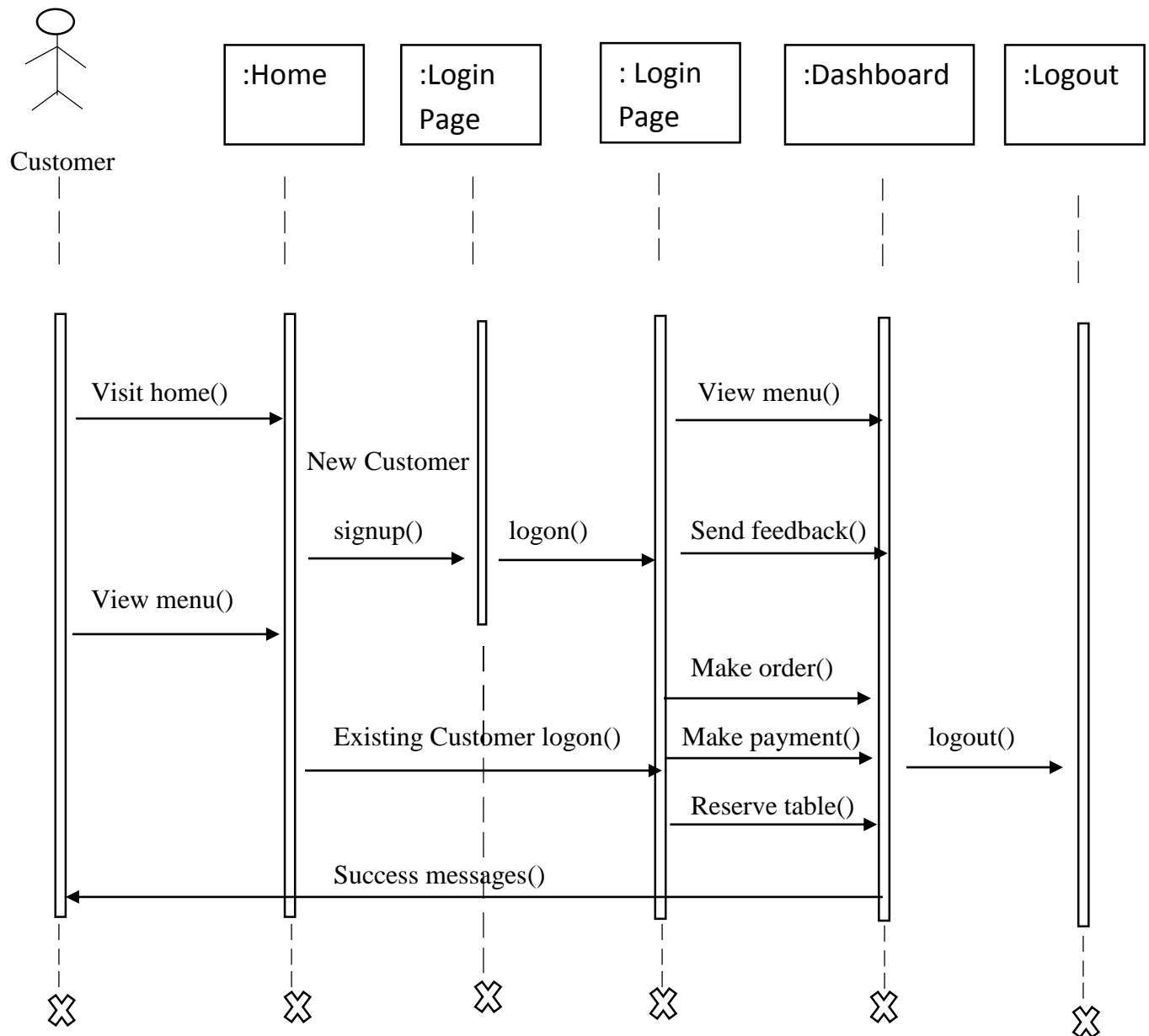


Figure 4.3 shows the sequence diagram for the customer use case.

Sequence Diagram for Manager Use Case

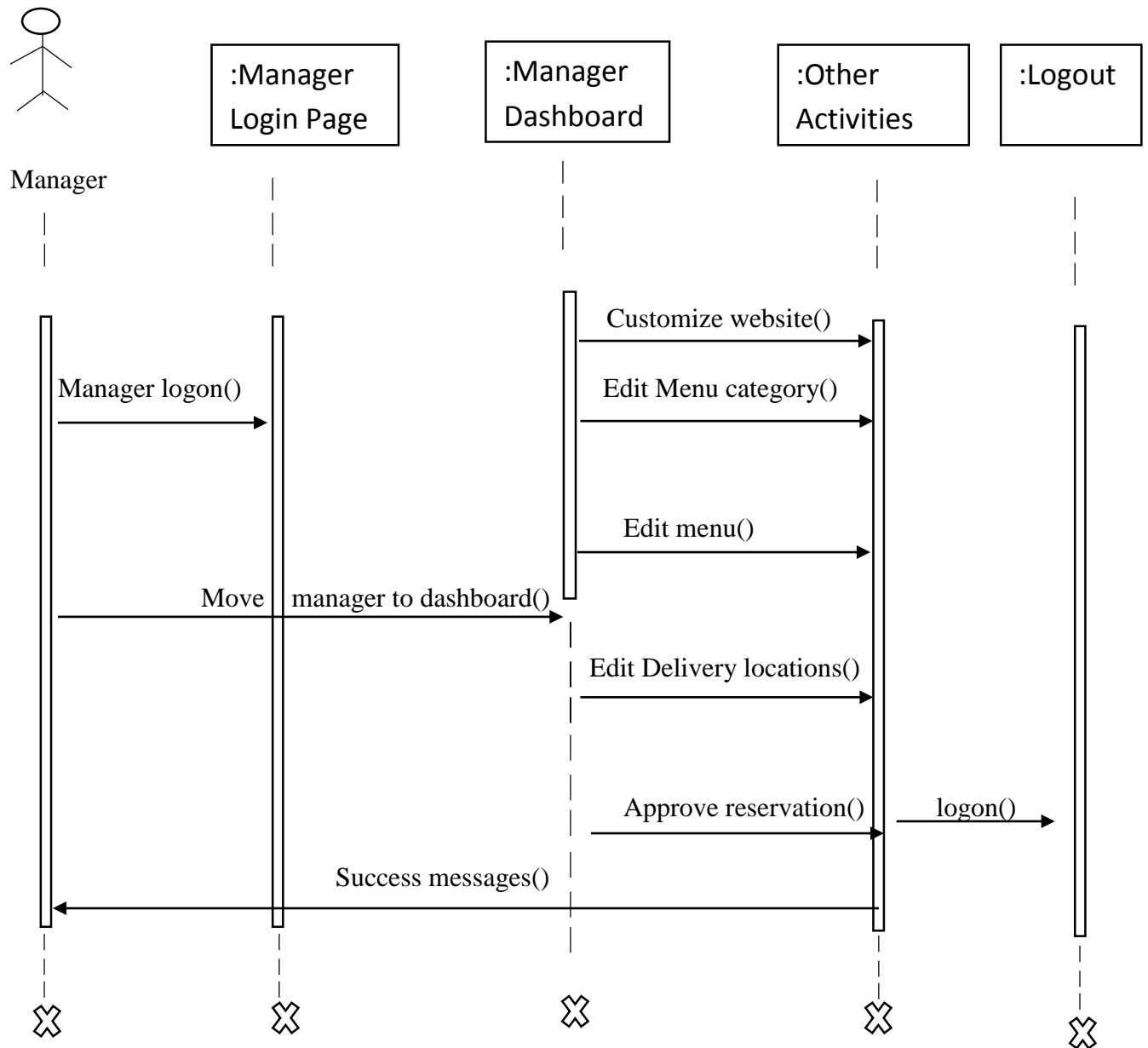


Figure 4.4 shows the sequence diagram for the Manager use case.

4.2.3 ACTIVITY DIAGRAM

An activity diagram is a UML diagram and it's a flow chart used to represent the flow from one activity to another activity. Below are the activity diagrams modelled for each actor.

Figure 4.5 shows the activity diagram for the customer's use case

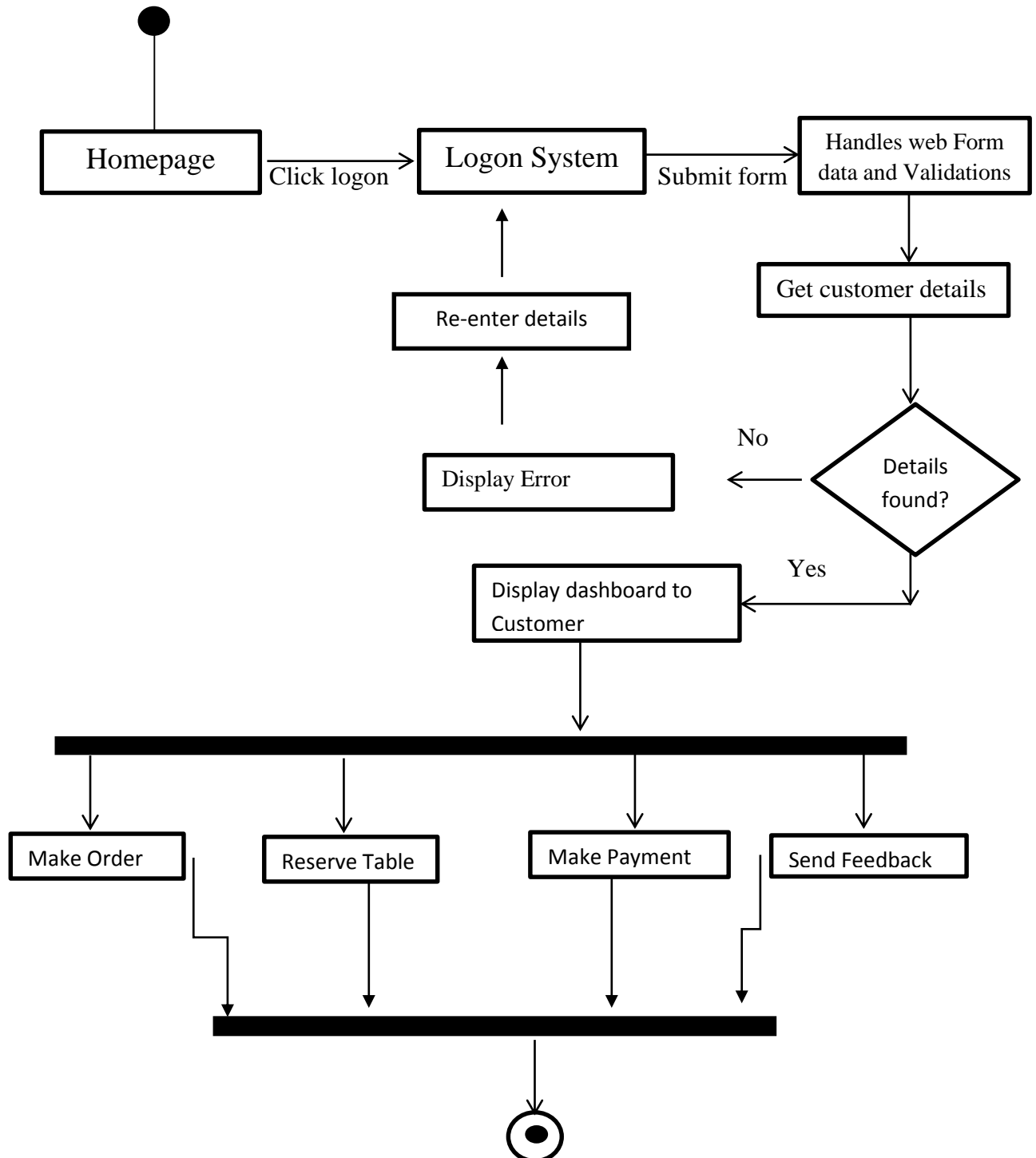
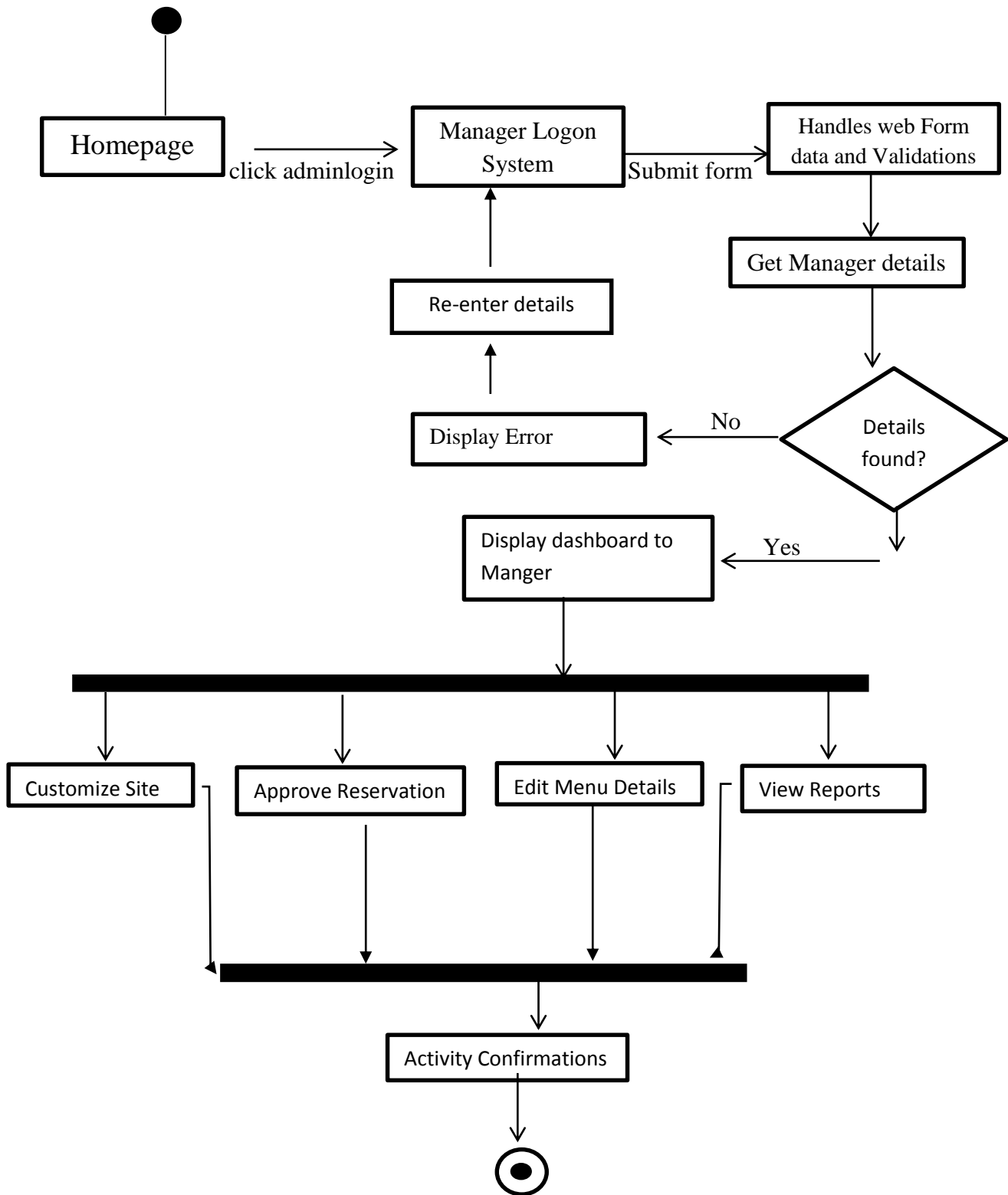


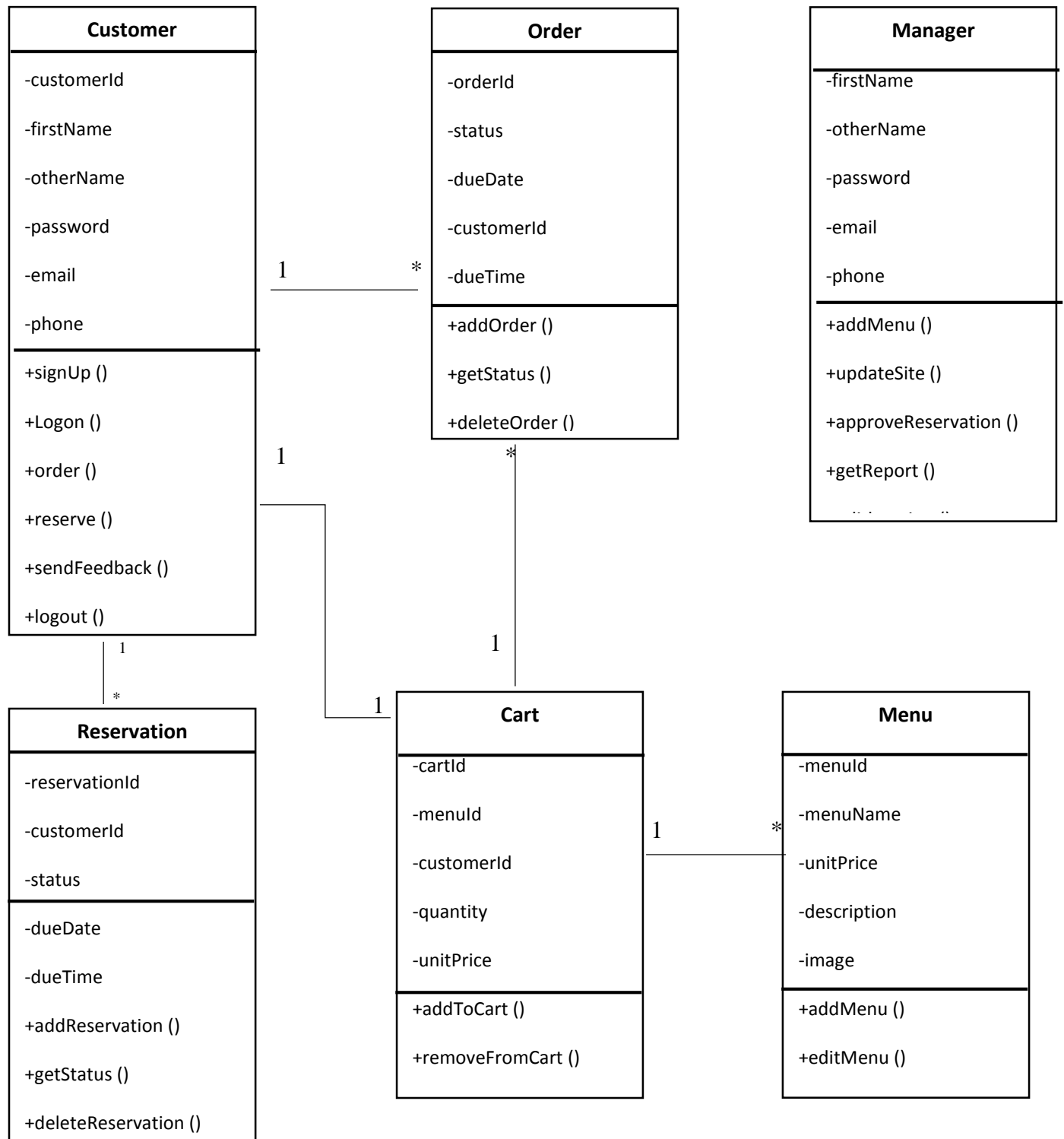
Figure 4.6 shows the activity diagram for the Manager use case



4.2.4 CLASS DIAGRAM

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

Figure 4.7 shows the class diagram for the online food ordering and reservation system



4.3 DATABASE DESIGN

A database is an organized collection of data. It is the collection of schemas, tables, queries, reports, views and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information. In this context, a database is a collection of information organized in such a way that a computer program can quickly rapidly search and retrieve desired pieces of data.

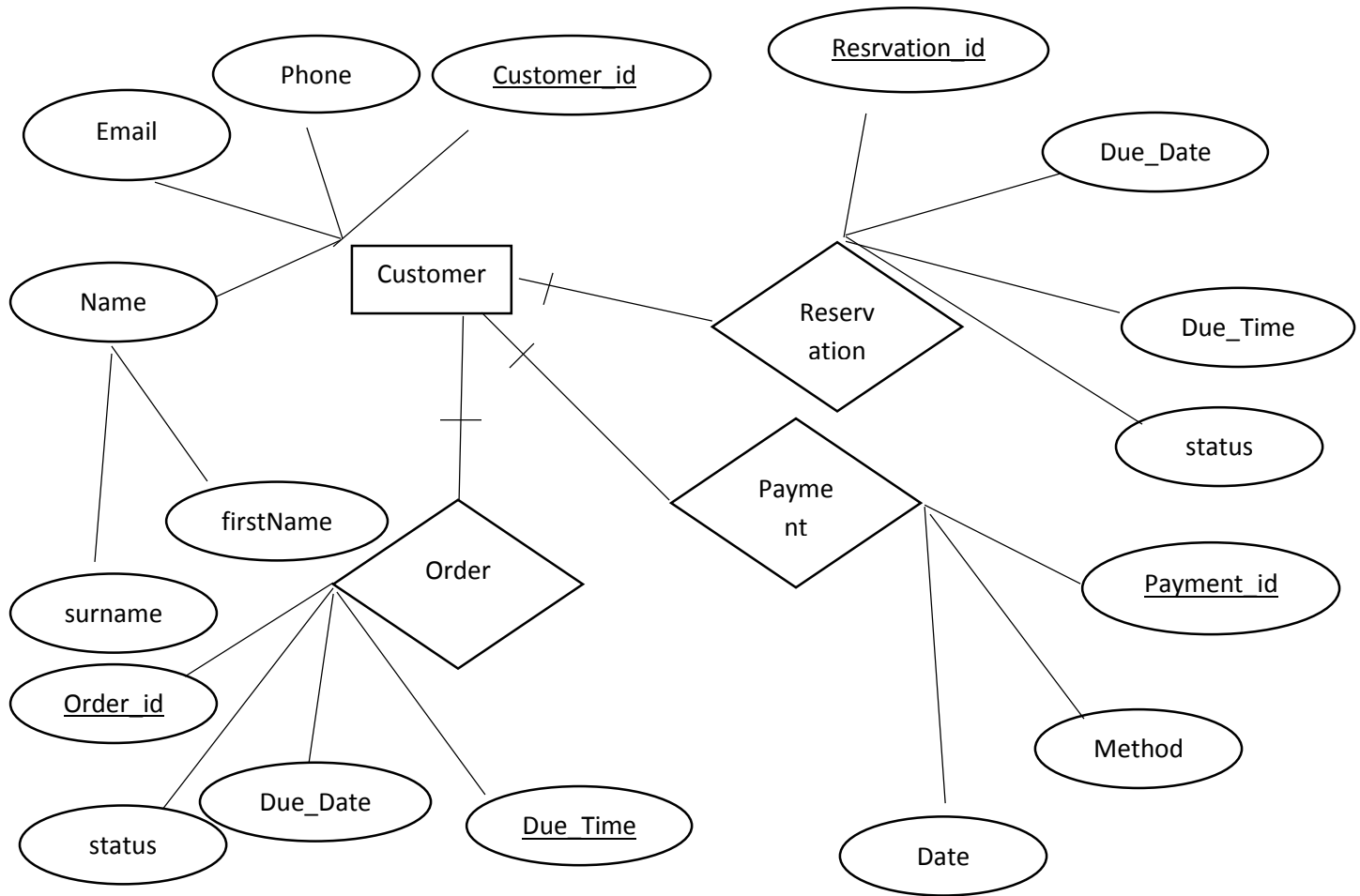
Database design is a process of modeling an enterprise and it involves the logical design of the base data structures used to store the data. A well designed database gives access to up-to-date and accurate information. The Entity-Relationship (ER) model, Unified Modeling Language (UML), Relational Model (RM) among others are some of the models used by database designers. The Entity-Relationship (ER) model was used for our database design.

4.3.1 Entity relationship model

Entity relationship model is a graphical representation of entities and their logical relationships to each other in relation to the organization of data within a database or information systems. The structure for a database in ER modeling is portrayed as a diagram known as entity-relationship diagram (ER diagram). The entity-relationship diagram is an important aid in the design, optimization and debugging of database diagrams.

In an ER diagram the entity represents an object or concept about which data is stored. In our system an example of an entity include a customer and manager. An entity is represented as a rectangle and circles or ovals represent the attributes of an entity in an ER diagram. The relationship refers to how data is stored between entities. They are usually depicted as lines connecting the entities. A relationship could be One-to-One, One-to-Many and Many-to-Many.

Fig 4.8 ER diagram for the customer model



4.3.2 NORMALIZATION

Normalization is the process of organizing data in the database such that it meets certain basic requirements. It includes creating tables and establishing relationships between those tables according to rules designed to protect and to make the database more flexible by eliminating redundancy and inconsistent dependency. A normalized database allows databases to take as little disk space as possible and helps to avoid insertion anomaly update anomaly and deletion anomaly.

- **Insertion Anomaly:** It occurs when certain attributes cannot be inserted or entered into the database without the presence of other attributes.
- **Update Anomaly:** An update anomaly exists when one or more instances of duplicated data is updated but not all.
- **Deletion Anomaly:** This is when certain attributes are lost because of the deletion of other attributes.

Normalization rules are divided into the following normal form:

- **First Normal Form (1NF).** The rule for First Normal Form states that no two rows of data must contain repeating group of information, therefore each set of column must have a unique value. To ensure our tables were in the First Normal Form, we:
 - ✓ Eliminated repeating groups in individual tables.
 - ✓ Created a separate table for each set of related data.
 - ✓ Identified each set of related data with a primary key.
- **Second Normal Form (2NF).** A table is said to be in 2NF if it is in 1NF and there are no partial dependency of any column on the primary key. To ensure our tables were in the Second Normal Form we:
 - ✓ Created separate tables for sets of values that apply to multiple records.
 - ✓ Related these tables with a foreign key.
- **Third normal Form (3NF).** It applies that a table must be in 2NF and every non-prime attribute of the table must be dependent on the primary key and hence cannot be

determined by another non-prime attribute. To ensure our tables were in the Third Normal Form we:

- ✓ Eliminated fields that do not depend on the key.

Our database is normalized in the 3NF to ensure data redundant data is eliminated that is, all data is stored in only one place and also to ensure data dependencies are logical which means all related data items are stored together.

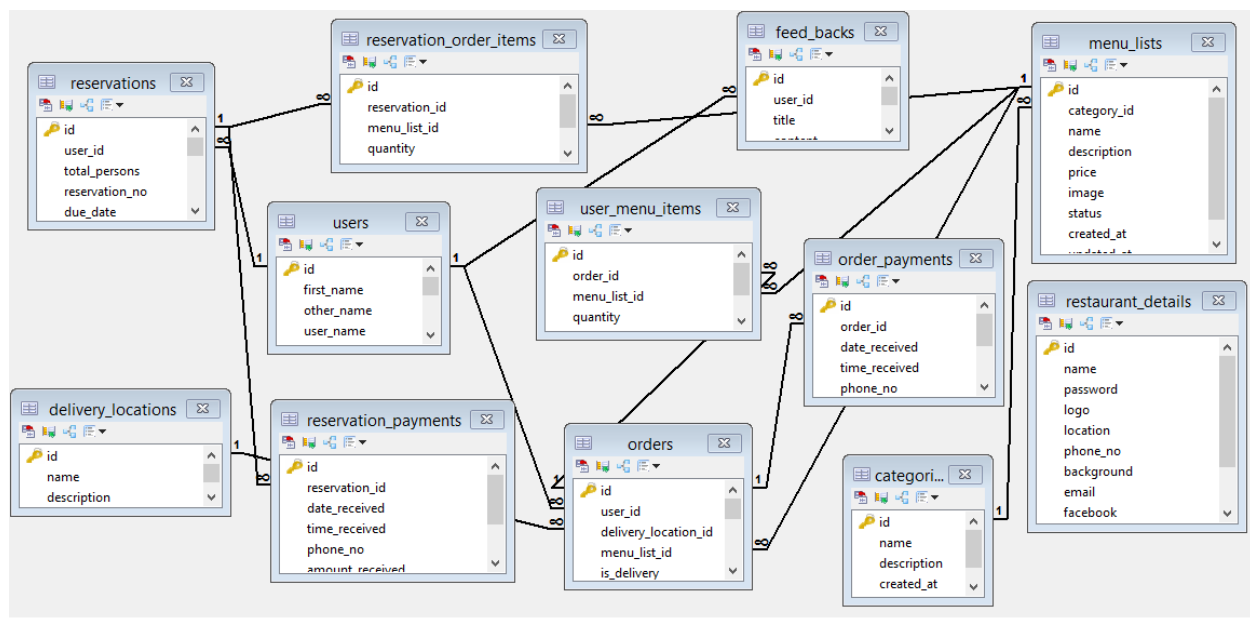


Fig 4.9 Normalized database for all tables

4.4 USER INTERFACE DESIGN

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design and information architecture. Users have become familiar with interface elements acting in a certain way and hence it is very important that we design the interface in a way that will help with task completion, efficiency, and satisfaction.

During our system development phase, we shall use several UI design features like input controls which includes textboxes, radio buttons, checkboxes, date fields, dropdown lists etc. For example when a customer wants to order food, the dates and time for the delivery would be designed with date field and dropdown to help the customer order with ease. We shall also make use of navigation components such as pagination, search field, slider, icons, tags etc. Information components like notifications, messages boxes etc. will also be used to help the user get quick feedbacks from the system.

We intend to use the following best practices recognized by professional programmers in our user interface (UI) design:

- **Keeping the interface simple:** The interface will be designed with the customer who will make the orders and reservations in mind. They need food and probably do not have to spend much time to order food. Our interface will be simple for the customer to navigate easily.
- **Creating Consistence and Common UI elements:** The UI design elements that would be used in the system implementation will be consistent throughout the site. This will help the user to be familiar with certain UI elements so that he/she will not be using different elements for similar operations. Common elements familiar to the user will also be used.
- **We shall be purposeful in page layout.** We shall consider the spatial relationships between items on the page and structure the page based on importance. Careful placement of items can help draw attention to the most important pieces of information and can aid scanning and readability.

- **Strategically use color and texture.** We can direct attention toward or redirect attention away from items using color, light, contrast, etc. We intend to blend colors that will make the website nice and simple.
- **Using typography to create hierarchy and clarity.** Carefully consider how you use typeface. Different sizes, fonts, and arrangement of the text to help increase scan ability, legibility and readability of the user.
- **Making sure that the system communicates what's happening.** It is imperative to inform your users of location, actions, changes in state, or errors. The use of various UI elements to communicate status and, if necessary, next steps can reduce frustration for your user and that is exactly what we will do during the implementation of the system.
- **Thinking about the defaults.** By carefully thinking about and anticipating the goals people bring to our site, we can create defaults that reduce the burden on the user. This becomes particularly important when it comes to form design where you might have an opportunity to have some fields pre-chosen or filled out.

SUMMARY

This chapter focused on the analysis and design of our system. We first outlined the requirement specification of our system which included both functional and non-functional requirements and the hardware requirements. We further explained the various Unified Modelling languages we used in modelling our system. This included use case diagrams, activity diagrams, class diagrams and sequence diagrams. These diagrams provided a pictorial view of the relationships between the various entities or users of our system. We also looked at the design of our system's database including the Entity-Relationship (ER) diagram and the importance of database normalization. We stated that our database was in the third normal form (3NF). We finally talked about how user interface will look at and why it was necessary or the interface to be user friendly.

CHAPTER FIVE

SYSTEM IMPLEMENTATION

5.0 INTRODUCTION

This chapter details out the implementation of the system. This include mapping logical design onto physical platform, the construction which consist of screen shots of forms, databases and reports among others. The chapter will also outline the various testing that were performed on the system, that is unit and system testing and the results obtained.

5.1 MAPPING LOGICAL DESIGN ONTO PHYSICAL PLATFORM

Logical design involves arranging data into a series of logical relationships called entities and attributes. Here, we defined the various fields for the database schema. Customer first names and other names were converted into first_name and other_name respectively. We used the larval framework and the other development tools to develop the various model of the project and linked them together. The next sub-topic gives a graphical view of the various outputs after we transformed the logical design into executable codes

5.2 CONSTRUCTION

This section focuses on the pictorial view of the user interface including the various forms, databases and reports. Below are various screen shots showing how the system was constructed.

The screenshots of the various tables constructed are shown below:

Alter Table 'categories' i							
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	name	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	description	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig 5.0 Database Schema for categories table

Alter Table 'carts' in 'g							
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	user_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	menu_list_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	name	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	quantity	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	unit_price	decimal	5,2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	subtotal_price	decimal	5,2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig 5.1 Database Schema for carts table

Alter Table 'users' in 'c'								
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	first_name	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	other_name	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	phone_no	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	email	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	location	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	user_type	enum	'customer',		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	password	varchar	60		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	remember_token	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Fig 5.2 Database Schema for users table

Alter Table 'user_menu_items'								
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	order_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	menu_list_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	quantity	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	unit_price	decimal	5,2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	sub_total	decimal	5,2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Fig 5.3 Database Schema for user_menu_items table

Alter Table 'reservations'								
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	user_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	total_persons	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	due_date	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	due_time	time			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	is_smoking	enum	'yes', 'no'		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	is_outside	enum	'yes', 'no'		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	flat_rate	decimal	5,2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	comments	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	reservation_status	enum	'pending', '		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Fig 5.4 Database Schema for reservations table

Alter Table 'orders' in								
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	user_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	delivery_location_id	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
	is_delivery	enum	'yes', 'no'		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	due_date	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	due_time	time			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	comments	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	order_status	enum	'pending', '		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	total_amount	decimal	5,2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	delivered	enum	'no', 'yes'		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Fig 5.5 Database Schema for orders table

Alter Table 'menu_lists' in							
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	category_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	name	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	description	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	price	decimal	5,2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	image	blob			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	status	enum	'available'		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig 5.6 Database Schema for menu_lists table

Alter Table 'feed_backs' in							
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	user_id	int	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	title	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	content	text			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig 5.7 Database Schema for feed_backs table

Alter Table 'delivery_location'							
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?
*	id	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	name	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	description	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	price	decimal	5,2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	created_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	updated_at	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig 5.8 Database Schema for delivery_locations table

5.2.1 USER INTERFACES

As we mentioned in the previous chapter about how the user interface would be designed, below are the screenshots of the various user interfaces of our system that we developed.

Login

E-Mail Address

Password

☐ Remember Me

Login

Sign Up

Forgot Your Password?

Fig 5.10 Login Interface that enables dinners to logon and make orders

Create An Account With Us

First Name


Other Name

Phone #

E-Mail Address

Password

Confirm Password

 Sign Up

[Login Instead](#)

Fig 5.11 Registration interface which enables customers to create an online account.

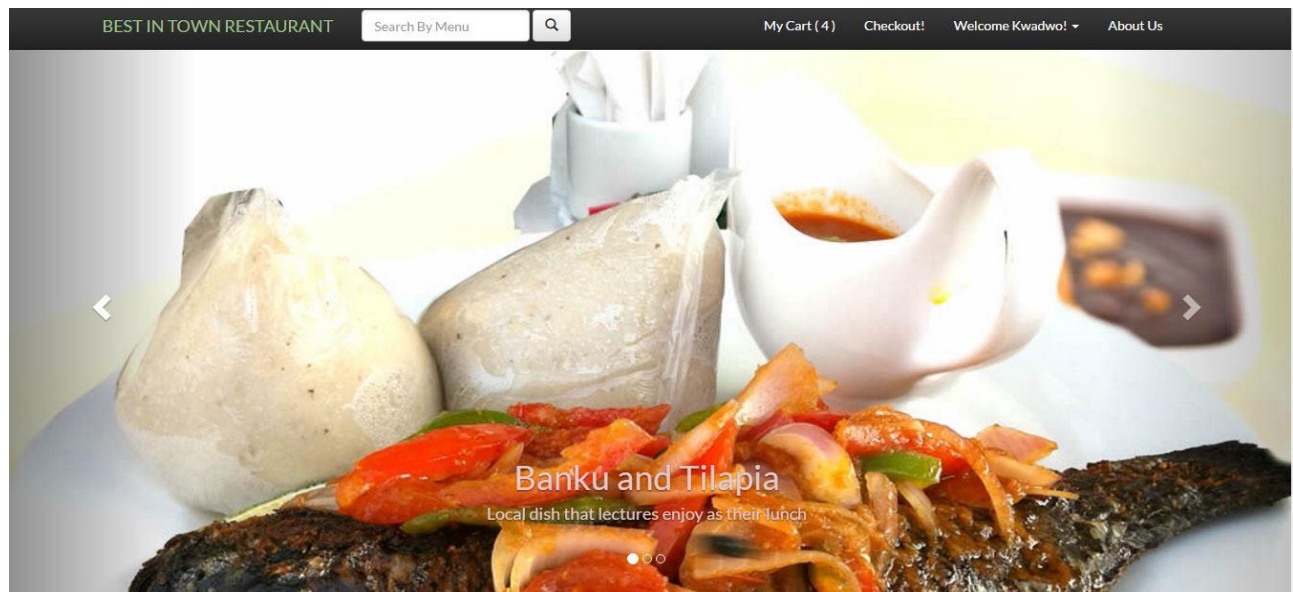



Fig 5.12 Homepage after a Customer has logged in to make order or reserve table.



Best In Town Restaurant » Your Number One Online Food Ordering Vendor on KNUST Campus.

Best In Town Restaurant is the best food ordering and table reservation center where both local and continental dishes are served. We take orders for delivery and pick-up/take away.

We Make Your Table Reservation Very Easy!

Please Note: You Pay a Non-Refundable Amount of GHS 5.00 at Our Payment Portal for a Successful Table Reservation

No. of Persons ▼

05/16/2016

Reservation Time ▼

Any comment about your reservation?

Reserve Now!

☐ Outside Table?
 ☐ Smoking Area?

Fig 5.13 Reservation Interface enables logged in customers to reserved table

FOOD MENUS IN YOUR CART

NO	NAME	QUANTITY	UNIT PRICE	SUB-TOTAL
1	Banku and Tilapia	1	0.10	0.10
2	Fufu and Palmnut soup	2	0.20	0.40
3	Can Coke	1	5.00	5.00
4	Beef Fried rice	1	0.20	0.20

CONFIRMATION AND PAYMENTS

Pick Up or Delivery?

Ayeduae (GHS 0.10)

Maximum Delivery Time : 50 mins

Opening Hours is : 11:00am - 6:00pm

Date & Time For Your Order

05/17/2016 13:00

Give us any Comment on Your Order

Please I don't want too much pepper

Total Amount (GHS):

5.70

Order Now!

Fig 5.14 Cart Information presented to customers before final checkout to make payments.

Best In Town Restaurant

Your Number One Online Food Ordering

Item Description	Quantity	Unit Price	Total Price
Your Pick-Up Location: Ayeduae Near Attah Mills Junction	1	0.10	0.10
Banku and Tilapia	1	0.10	0.10
Fufu and Palmnut soup	2	0.20	0.40
Can Coke	1	5.00	5.00
Beef Fried rice	1	0.20	0.20
VAT (15 Percent) Included			0.87
NHIL (2.5 Percent) Included			0.14
Total Amount:			5.80

X Cancel Checkout

Total Amount Due (GHS)

¢ 5.80

Make Payment

Download Invoice (PDF)

Fig 5.15 Payment Interface at Mpower Payment Portal with order details, amount to be paid as well as Tax information

Whops! it looks like you have cancelled your order at our payment portal

Would You like to start a new order?

Yes No Logout

Fig 5.16 Cancelled Order page where a customer is redirected after cancelling an order.

Confirmed Orders

Get Report

Total Amount Received (GHS): 1845.59

17.5% VAT to GRA (GHS): 322.98

From

To

Go!

Net Amount Received (GHS): 1522.61

ID	CUSTOMER	PHONE	LOCATION	DUE DATE/TIME	ORDER DATE/TIME	STATUS	AMOUNT PAID	DELIVERED?	INFO
1	Kwadwo Tutu	0248396348	Kotei	2016-05-15 13:00:00	2016-05-19 07:13:50	confirmed	95.00	no	
2	Florence Obinim	0200220020	Tech Junction	2016-05-13 11:30:00	2016-05-19 05:57:58	confirmed	70.00	yes	
3	James Owusu	0501382113	Pick-Up	2016-05-15 12:30:00	2016-05-19 05:52:56	confirmed	22.00	no	
4	Osei Owusu	0244909098	Pick-Up	2016-02-15 11:00:00	2016-05-18 07:01:29	confirmed	56.00	yes	
5	Anthony Asare	0501382113	Tech Junction	2016-05-15 11:00:00	2016-05-17 07:08:02	confirmed	40.50	yes	
6	Anthony Asare	0501382113	Africa Hall	2016-05-17 12:30:00	2016-05-16 06:20:05	confirmed	5.00	no	
7	Anthony Asare	0501382113	Indece Hall	2016-05-16 13:00:00	2016-05-15 14:19:46	confirmed	46.50	yes	

Fig 5.17 All Confirmed Orders report from the start of business till date.

CONFIRMED ORDERS FROM 2016-05-01 TO 2016-05-15 FOR MANAGEMENT DECISION MAKING

TOTAL AMOUNT RECEIVED (GHS): 792.5

17.5% VAT AMOUNT TO GRA (GHS): 138.69

NET AMOUNT RECEIVED (GHS): 653.81

ID	CUSTOMER	LOCATION	DUE DATE/TIME	ORDER DATE/TIME	AMOUNT PAID	DELIVERED?
1	Anthony	Pick-Up	2016-04-21 14:00:00	2016-05-10 20:35:52	10.00	yes
2	francis	Pick-Up	2016-05-11 13:30:00	2016-05-11 10:39:13	70.00	yes
3	Belinda	Pick-Up	2016-05-13 11:00:00	2016-05-12 05:39:48	75.50	yes
4	Frank	Republic Hall	2016-05-13 11:00:00	2016-05-12 05:40:42	77.00	yes
5	James	Pick-Up	2016-05-13 13:00:00	2016-05-12 05:45:40	75.50	no
6	Osei	Ayeduse	2016-05-13 12:00:00	2016-05-12 08:21:19	30.00	yes
7	Albert	Tech Junction	2016-05-13 12:30:00	2016-05-13 09:40:00	23.00	no

Fig 5.18 All Confirmed Orders in pdf format for specific date range.

All Orders

















ID	CUSTOMER	PHONE	LOCATION	DUE DATE/TIME	ORDER DATE/TIME	STATUS	AMOUNT PAID	DELIVERED?	INFO
1	Kwadwo Tutu	0248396348	Kotei	2016-05-15 13:00:00	2016-05-19 07:13:50	confirmed	95.00	no	
2	Florence Obinim	0200220020	Tech Junction	2016-05-13 11:30:00	2016-05-19 05:57:58	confirmed	70.00	yes	
3	Osei Owusu	0244909098	Pick-Up	2016-05-15 12:30:00	2016-05-19 05:54:39	pending	23.00	no	
4	James Owusu	0501382113	Pick-Up	2016-05-15 12:30:00	2016-05-19 05:52:56	confirmed	22.00	no	
5	Richard Akyer	0255098899	Katanga Hall	2016-05-15 13:00:00	2016-05-18 07:10:53	pending	94.50	no	
6	Osei Owusu	0244909098	Pick-Up	2016-02-15 11:00:00	2016-05-18 07:01:29	confirmed	56.00	yes	
7	Anthony Asare	0501382113	Tech Junction	2016-05-15 11:00:00	2016-05-17 07:08:02	confirmed	40.50	yes	
8	Kwadwo Tutu	0248396348	Ayeduase	2016-05-17 13:00:00	2016-05-16 14:05:39	pending	5.80	no	
9	James Owusu	0501382113	Pick-Up	2016-03-05 11:00:00	2016-05-16 07:03:37	pending	75.00	no	
10	Anthony Asare	0501382113	Africa Hall	2016-05-17 12:30:00	2016-05-16 06:20:05	confirmed	5.00	no	

Fig 5.19 Pending and Confirmed Orders for manger to approve delivery or otherwise

Orders/Reservations Made

Edit Your Information

Your Order History

NO	DELIVERY?	LOCATION	DUE DATE/TIME	STATUS	AMOUNT	MENUS	CANCEL
1	yes	Kotei	2016-05-15 13:00:00	confirmed	95.00		
2	yes	Ayeduase	2016-05-17 13:00:00	pending	5.80		
3	no	None	2016-05-13 11:30:00	confirmed	14.00		
4	yes	Conti Hall	2016-05-19 11:00:00	confirmed	33.00		
5	no	None	2016-04-22 12:00:00	pending	20.00		
6	no	None	2016-04-18 12:30:00	pending	45.00		
7	yes	Africa Hall	2016-04-18 12:00:00	pending	33.00		
8	yes	Conti Hall	2016-04-18 12:30:00	confirmed	30.00		

Your Reservations History

Fig 5.20 Customer Order and Reservation history that details all orders and reservations made for a period of time.

**CONFIRMED RESERVATIONS FROM 2016-05-01 TO 2016-05-16 FOR
MANAGEMENT DECISION MAKING**

Total Amount Received (GHS): 25

ID	NAME	PERSONS	DATE	TIME	SMOKE	OUTSIDE	AMOUNT
1	Bright	3	2016-05-11	13:00:00	no	no	5.00
2	Belinda	5	2016-05-14	16:00:00	no	no	5.00
3	Frank	6	2016-05-15	15:00:00	no	no	5.00
4	Kwadwo	3	2016-05-13	12:00:00	no	no	5.00
5	Kwadwo	4	2016-05-14	14:30:00	no	yes	5.00

Fig 5.21 Confirmed Reservations Report for a specific date range.

All Food Menus













MENU NAME	MENU DESCRIPTION	PRICE	IMAGE	STATUS	EDIT	DELETE
Shawama	Good Chinnese dish	25.00		available		
Fried Rice with Chicken	Chicken, sallad, onion, etc	12.00		available		
Bottled Malt	Nice taste	3.00		available		
Can Malt	Tastes great	3.50		available		

Fig 5.22 Menu Details Interface for Manager to edit menu information and update details

Where We Deliver To

















NAME	DESCRIPTION	PRICE	EDIT	DELETE
Africa Hall	Basement	1.50		
Indece Hall	Around Car Park	1.50		
Katanga Hall	Around the Foya	1.50		
Conti Hall	Near The Car Park	1.50		
Republic Hall	Opposite the ATM	1.50		
Tech Junction	Around the foot bridge	2.50		
Ayeduse	Near Attah Mills Junction	0.10		
Kotei	Around Lumber Junction	2.00		

Fig 5.22 Delivery Locations Interface for manager to edit delivery locations of the restaurant

Add New Food Menu

Menu Name

Menu Name Here

Menu Description

Enter Description here

Select Category

Please Select Category

Menu Price

Amount in GHS

Menu Status

☒ Available ☐ Unavailable

Fig 5.23 New Menu registration form for manager to add new menu into the database

5.3 TESTING

Testing is the process of evaluating a software or system to detect differences between given input and expected output. Testing is conducted to assess the quality of a system hence can be said to be a validation and verification process. This ensures that system meets the technical requirements that guided its design and development, works as expected and can be implemented with the same characteristics. Testing include both unit and system testing among others.

5.3.1 UNIT TESTING

It is the practice of testing certain functions and areas known as units of a source code. This is important because it helps to verify whether the various units are functioning as expected, hence are returning the proper values, therefore it helps in identifying failures or errors in a source code. With our system we tested the various unit such as the order food unit, customer registration unit, among others to ensure they were returning the expected values. Our system has two main units; one for the customer and one for the manager. We tested the two to ensure that they functioned independently without errors in either of the two units affecting each other. This is relevant because it also helps in ensuring that the various units were error free and working in the most efficient manner.

5.3.2 SYSTEM TESTING

After the unit testing the entire units are integrated to form a complete system. This is to detect inconsistencies between the various units after which system testing is performed on the system as a whole. System testing is usually the final testing conducted to ensure the system to be delivered meets the specifications and purpose. This testing is conducted to evaluate the systems compliance with its specified requirements that is both functional and non- functional requirements.

In conducting system testing we entered some invalid input such as negative order quantity to ensure our system does not accept such values. This was successful because our system responded by alerting us that the input was invalid. We also did same with the date on which the order is to be delivered by entering dates that are past the default date. The system also alerted us that the date was invalid. This enforced the fact that the system is validated and will not accept invalid input.

We also tested the system's payment process by ensuring that an order is only successful when payment is successful. This also was successful, hence the system performs in harmony with the functional requirements.

With the non-functional requirements, our system is very interactive in that it gives feedback to the user for every required function. The system is also easy to use since it has an easy and friendly

looking interface. The system was also tested on other platforms apart from the laptop such as a mobile phone and a tablet, and the system fit onto those devices and functioned well. This proved that the system is also scalable as required.

5.3.3 RESULTS OF TESTING

Below are some screenshots from the various testing we carried out to ensure that our system was functioning in line with the requirements analysis we gathered from users.

As stated above, during our system testing, our application did not accept negative inputs.



Margarita Pizza

GHS 7.00

-3

*Only Positive Integers

Add Menu

Fig 5.24 Testing for a Customer adding negative order details.



Date & Time For Your Order

05/23/2016

Select a Time

Give us an

Value must be 05/21/2016 or earlier.

Fig 5.25 any customer who tries to make order for past dates are not allowed

Phone #

E-Mail Address

The email has already been taken.

Password

Fig 5.26 Testing to ensure that registered customers do not register again.

5.4 EVALUATION OF THE PROJECT

The purpose of evaluation is to assess the system as to whether it does what it is expected to do and if it is working properly. From the testing phase we analyzed that our system is functioning well according to the specified requirements.

One of the objectives for designing and developing the online restaurant reservation system included easing customers of the stress they go through to order food and reserve a table through phone calls. This system provides the solution to that, in that customers in their comfort can log in to the restaurants website and place their orders. They receive both SMS and email confirming their order has been received and being processed which is very assuring.

The system is said to have an interactive interface which is also easy to use because the various food available are displayed right on the home page for customers to view and select their choice. The various food have also been sorted into food categories such as continental dishes, local dishes for easy searching and selection. Customers can therefore use the system as first timers without difficulties. These therefore we believe will increase customer satisfaction as we perceived in chapter one of our documentation.

We were also able to implement the payment system as we stated in our project scope. Customers can therefore only have a successful food order after payment has been done. This will reduce the

cost incurred by most restaurants as a result of customers not turning up for ordered food or reserved table.

On the side of managers, they can perform some managerial duties with the system such as edit menu by adding food available or deleting food unavailable. They can also reply to feedbacks from customers by sending them a direct email. They can generate reports in the form of PDF of the various orders and table reservations received. This will help them access their performance over a specified period of time.

From the above we can confidently conclude that from the evaluation of our project, our project meets its overall aim as stated in chapter one which is to design and implement an online restaurant reservation system to enable diners order food and reserve tables at restaurants to optimize food ordering and table reservation.

• EVALUATION OF SOLUTION

Testing and evaluating the system has helped us realized that our system serves as a solution to the problem statement in chapter one. All that customers have to do is log on to the website of their preferred restaurant and make their orders without joining any long queues. The problem of restaurant having poor records is also solved. This is because managers can use the application to access reports and status of orders and reservations and can also download them in PDF format for record keeping.

From the above we can conclude that our system is a solution to the problem we aimed to address and solve.

• EVALUATION OF METHODOLOGY

We adopted the agile method with the incremental software process model to develop our application as stated in chapter three. This approach best suit our development because we developed the application in increments and based on the suggestions of our supervisor we

made the necessary changes until the final phase was reached. This made the development much easier and helped us to meet the various user and system requirements.

5.5 CONCLUSION

In this chapter, we looked at mapping the logical design onto the physical platform where the logical database schema was transformed into executable codes. We then looked at constructions where we displayed the various user interfaces we discussed in the previous chapter. Screen shots of the various forms, reports, etc. were captured under construction. We proceeded with testing our software. We performed both unit and system testing to ensure the software was working as required. We concluded with evaluation of the project where we highlighted the various efforts we put in place as a team to make this project a success.

CHAPTER SIX

FINDINGS AND CONCLUSION

6.0 INTRODUCTION

Findings and conclusion concerns itself with the facts and figures gathered to satisfy the objectives of a research or the development of a project as well as the inferences drawn from the findings. It also involves the principal results of a research project as well as what the project suggested or indicated and the interpretations of the relevance of the findings of a project.

This chapter will outline the summary of problems, achievements and challenges we faced, recommendations as well as future work.

6.1 SUMMARY OF PROBLEMS

We encountered a number of problems during the development of the project. We both had different programming level and were new to the framework w were using. These therefore made it quite difficult for us to progress at the same rate because the one with better programming ability had to make time and teach the other to be able to progress on the project. We also had different preferences as to how the various interfaces should like and the colors to use.

The above were basically the problems we encountered as a team whiles working on our project.

6.2 ACHIEVEMENTS AND CHALLENGES

ACHIEVEMENTS

There are a number of successes we achieved through effort, skill and hard work and these achievements are stated below:

- Working as a team has helped us to develop good team skills. We have learnt how to communicate better, assign roles and accomplish our given tasks as team members within

a specific time frame and also learnt to cooperate and coordinate our efforts to achieve our aims and objectives.

- The development of the project required the use of a php framework known as Laravel and also MySQLi which we were not familiar with but were able to learn and use it for the development of the project. This have therefore helped us to enhance our programming skills.
- We were able to implement a payment system and report generation for our system. This is an achievement because one of our motivations for undertaking this project included the cost incurred by restaurants. By means of the payment system that problem is tackled.

CHALLENGES

Although the development of the project was a success, we encountered some challenges as well. These include:

- One of the challenges we faced was to complete the project on time. This is as a result of the fact that we had to make time for classes and also for developing the project.
- We also faced the challenge of implementing the payment system. This was very difficult from the onset because we did not know how to go about it and when we did we had problems of integrating the payment system into our system since it was a third party system.
- Generating report was also another challenge we faced especially when we wanted the report to be on periodic bases.

6.3 RECOMMENDATIONS

Recommendations involve suggestions or courses of actions that could be added to the project. Some recommendations for our system include:

- Our project is purely web-based hence we recommend that a mobile version could be developed to make it more convenient for users to use especially now that mobile devices are abundant.
- We also recommend that more payment options should be added for the mobile money section. We have options only for either MTN users or Airtel users hence limiting those who can use the application.

6.4 FUTURE WORK

In future we hope to delve more into the reservation system. This is because that aspect of our system does not offer much services to the user. We hope to expand the choices available to the customer and also develop a separate interface for table reservation.

6.5 CONCLUSION

This chapter highlighted the problems we encountered, achievements and challenges we faced and also provided recommendations for others who will like to work on a similar project. We also talked about changes we will like to make to the system in the future.

REFERENCES

1. <http://www.euromonitor.com/100-home-delivery-takeaway-in-the-us/report>
2. <https://en.wikipedia.org/wiki/TechCrunch>
3. https://en.wikipedia.org/wiki/History_of_the_Internet
4. <http://codular.com/php-mysqli>
5. <http://php.net/manual/en/function.mysql-query.php>
6. <http://laravelbook.com/laravel-architecture>
7. <http://www.tonymarston.net/php-mysql/model-view-controller.html>
8. https://en.wikipedia.org/wiki/GrubHub_Seamless
9. <https://www.grubhub.com>
10. <http://www.businessinsider.com/opentable-explainer-2011-10>
11. <http://www.opentable.co.uk/start/home>
12. <https://www.table-booking.com/novumibiza>
13. <http://aakashtechsupportdocs.readthedocs.io/en/latest/contributors.html>
14. <http://sdc.net.au/>
15. <https://www.table-booking.com/>
16. <https://support.microsoft.com/en-us/kb/283878>
17. Cupps, Bryan, and Tim Glass. "Internet online order method and apparatus." U.S. Patent No. 5,991,739. 23 Nov. 1999.
18. Scifo, Danilo, et al. "Online food ordering system and method." U.S. Patent Application No. 12/069,895.
19. Kimes, Sheryl E. "The current state of online food ordering in the US restaurant industry." Cornell Hospitality Report 11.17 (2011): p6-18.
20. Hong, Jin, and Qihong Nicki Zhang. "Open Mobile Online Reservation and Ordering Systems." U.S. Patent Application No. 12/344,596.
21. http://www.webopedia.com/TERM/C/client_server_architecture.html
22. <http://www.tutorialspoint.com/mysql/>
23. <http://www.phpandstuff.com/articles/top-10-reasons-why-you-should-use-a-php-framework>
24. <http://www.menudrive.com/blog/2013/05/five-online-ordering-trends-in-the-restaurant-industry/>
25. <http://agilemodeling.com/artifacts/classDiagram.htm>
26. <http://www.usability.gov/what-and-why/user-interface-design.html>
27. Jesse James Garrett's The Elements of User Experience: User-Centered Design for the Web and Beyond (2nd Edition).
28. Peter Morville and Louis Rosenfeld's Information Architecture for the World Wide Web: Designing Large-Scale Web Sites
29. <http://www.businessdictionary.com/definition/findings-conclusions-and-recommendations-FCR.html>
30. <http://www.studytonight.com/dbms/database-normalization.php>