RESTAURANT ONLINE RESERVATION SYSTEM

AMENUDDIN BIN ALIAS

BACHELOR OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING)

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL : **RESTAURANT ONLINE RESERVATION SYSTEM**

SESI PENGAJIAN: 2014/2015

Saya: AMENUDDIN BIN ALIAS

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (X)

☐	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☒	TIDAK TERHAD	

Disahkan oleh

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

NO 47 LRG PRMTG LIMAU
KG. SG SOI
25150 KUANTAN,
PAHANG DARUL MAKMUR

Nama penyelia:

EN. ABDUL SAHLI BIN FAKHARUDIN

Tarikh: 5/1/2015

Tarikh: 5/1/2015

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasiberkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

*** Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan

DECLARATION

I declare that this technical report entitled "Restaurant Online Reservation System is the result of my own research except as record in the references. The report has not been accepted for any bachelor and is not concurrently submitted in candidature of any other bachelor

Signature :.....

Name : Amenuddin Bin Alias

Date : 25 December 2014

ACKNOWLEDGEMENT

First of all, I would like to express my sincere thanks to my project supervisor, Mister Ssahli Bin Fakharudin for his insightful comments, outstanding advice, and exceptional guidance. I would also like to express my heartiest appreciation for his patience in spending a lot of time to guide me in my project and provide a lot of valuable and practical suggestions during this period.

Secondly, I would like to thank all my lecturers and instructors that have been guiding me in completing this documentation. My appreciation goes to Madam Azma Abdullah and all my friends a undergraduate student of FSKKP, who helped me during the process of development and documentation in order to assist myself to succeed the project. Also to my parent and family, Mr Alias Bin Embong and Mrs. Aminah binti Husain for being understanding and supportive not to mention financial support to me for the accomplishment of this undergraduate project.

As for all personnel involved direct and indirectly in Restaurant Online Reservation System, I am very thankful to have been guided and helped by them. For their helps and guidance, I would like to appreciate them. Thank You.

ABSTRACT

Restaurant Online Reservation System is developed for customer, staff and manager of Horizon Garden Restaurant. This system helps customer with the reservation and staff with the reservation management. In the existing system, the process is conducted manually that are wasting time and energy. Therefore, propose system is to change the current process to computerized system. The methodology that has been applied during project development is the prototype model. The advantages of the prototype model are it can be an early act or representation of the final product and this method can reduce risk and limits expenses and costs. In the development of the propose system Microsoft Visual Studio 2010 has been used as the programming tool with the SQL Server 2010 as the database.

ABSTRAK

Sistem Tempahan Online Restoran dibangunkan untuk pelanggan, kakitangan dan pihak pengurusan restoran New Horizon Garden. Sistem ini membantu pelanggan dan kakitangan untuk menguruskan tempahan meja secara atas talian. Di dalam sistem yang sedia ada, proses tersebut dijalankan secara manual yang hanya akan membuang masa dan tenaga. Oleh itu, sistem ini dibangunkan untuk mengubah cara tempahan meja dari manual kepada sistem atas talian. Metodologi yang telah digunakan semasa pembangunan projek ialah model prototaip. Kelebihan model prototaip adalah ia boleh menjadi satu tindakan awal atau perwakilan produk akhir dan kaedah ini boleh mengurangkan risiko dan had perbelanjaan dan kos. Dalam pembangunan sistem dan tugas Microsoft Visual Studio 2010 telah digunakan sebagai alatan pengaturcaraan dengan SQL Server 2010 sebagai pangkalan data untuk menyimpan maklumat yang berkaitan dengan tempahan.

TABLE OF CONTENTS

Tables of Figures

List of Tables

LIST OF ACRONYMS / ABBREVIATIONS / GLOSSARY

ABBREVIATIONS	DEFINITIONS
RORS	Restaurant Online Reservation System
SDD	Software Design Document
SRS	Software Requirement Specification
STR	Software Test Result

CHAPTER 1

INTRODUCTION

1.1. Introduction

Restaurant Online Reservation System (RORS) for Horizon Garden Restaurant is the reservation system to replace the manual reservation system that are currently use. This system can record all the customer information for the reservation such as customer's name, reservation date and time, number of people, and contact number.

For the customer, they can book the table on their own without need to go to the restaurant to make the reservation. In this system the user can also view the menu and check for the table available during their reservation time either lunch or dinner once the user input the time and date for the reservation.

The system also is developing for the use with the all type of web browser. As we all know there are many type of web browser with the access to the internet. With this, the customer can access to this restaurant website and make an online reservation at anytime and anywhere.

1.1.1. Problem Statement

Client of Horizon Garden Restaurant can book their table right on time before the days they are going to have a lunch or supper there. The issue is this restaurant is placed a long way from the main street which implies that the client need to drive around a couple of buildings in the wake of confronting a street jam amid a crest hours with an extremely restricted parking space before they enter the restaurant just to make a reservation. The client data from the reservation like name, contact number, number of individual, demand and requests is recorded in the "Reservation Book".

Another issue is, if one of the important data is not being recorded the restaurant can't contact the client for the affirmation as we know in some cases client simply scratch off their reservation without telling the restaurant.

Accordingly, the RORS is created to make it simple for the client to make and deal with their reservation. This system will empower clients to reserve the table in whenever and anyplace as long as they have an internet connection. This system additionally will empower client to pick their time and date by their own particular and they likewise can see the rundown of menu before they can put the reservation.

1.1.2. Objective

In order to develop the RORS, the overall objectives of this system are:

i. To computerize the reservation system from manual to computer system.
ii. To develop an online reservation system for user convenient.
iii. To easily manage customer information for the reservation

1.1.3. Scope

This project is limited to the following clause:

i. Design and develop the system using a ASP.net, Visual Basic and SQL server.

ii. The users of the Online Table Reservation System are the manager, Staff and customer.

iii. The study area is at the New Horizon Garden restaurant.

iv. The system can support Windows 7 and above.

v. The system is web based system and has a client side and server side.

vi. The system is develops using the computer with CPU Intel(R) Core™ i3-3220 3.30 GHz and Windows 7 Ultimate as the operating system.

1.2. Review Previous Work

1.2.1. Open Table

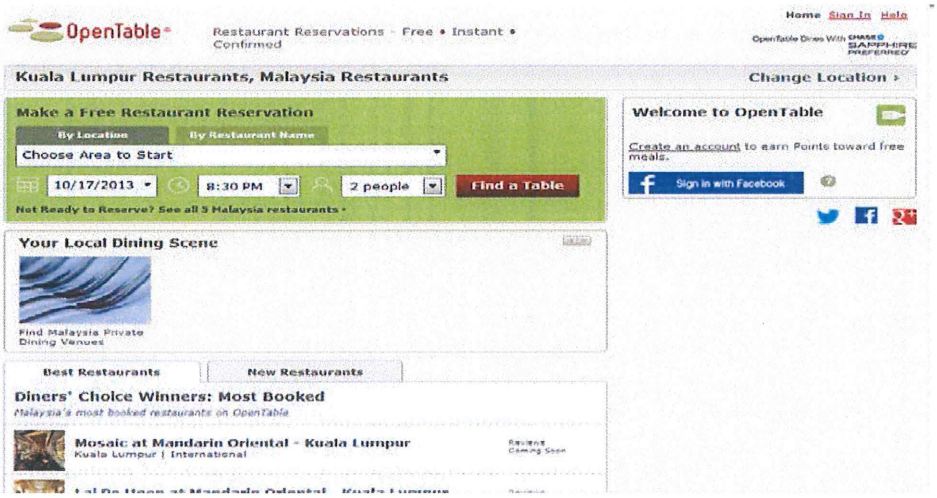


Figure 1. 1: Open Table Website

Open Table is a website that is connecting to restaurants and searching their computer reservation system to find a table for the customers that are trying to reserve a table for the restaurant on their choice.

Figure 1.1 shows the open table website. This website provides the choice of the restaurant for entire Malaysia. Customer also can choose their date, time and number of people to make the reservation.[1].

The strength of the Open Table is the customer can choose their location and the system will display the list of the restaurant near their location for the customer to make their choice. Beside that they also show the available time to make the reservation. This website also is not limited in the Malaysia country only but it provides the list of restaurant from the several countries around the world. In short this website is very suitable for the tourist if they want to look for a restaurant for having a meal. Unfortunately, this website is hardly to find because it is a website that can be used by all restaurant it is not directly connect to a specific restaurant

when the customer enter the name of the restaurant for a booking purpose. Besides, this website also only has been use by the high class restaurant.

RORS is implementing the function for the date, time, and number of people based on this website. These concepts of scroll down menu are making the customer easy to place the reservation.

1.2.2. Saloma Theatre Restaurant

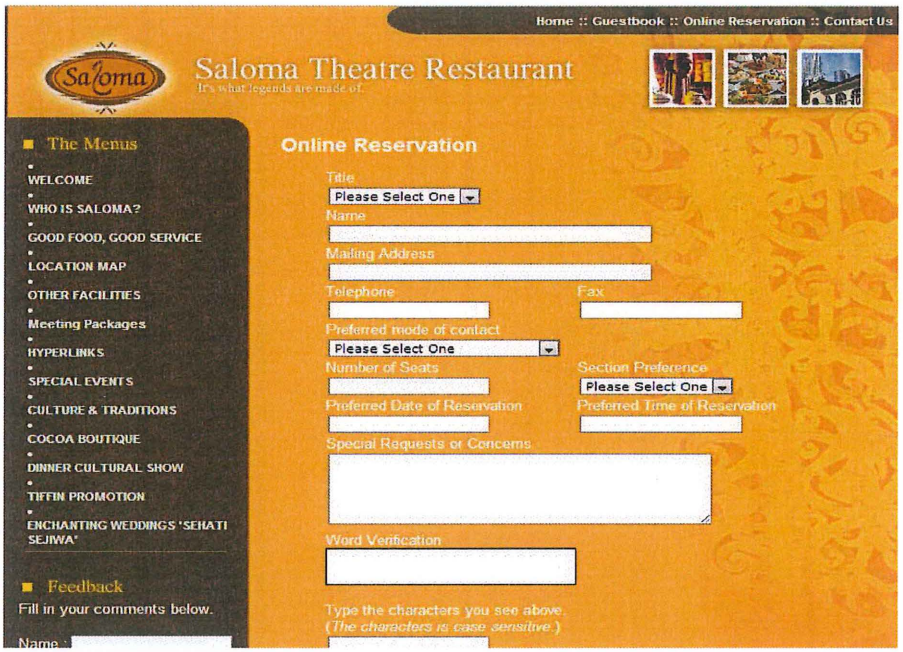


Figure 1. 2: Saloma Theatre Restaurant website

Saloma Theatre Restaurant website is the website that shares a lot of information about the restaurant to their customers. They share the information on about their services, facilities, special events, culture and tradition, and promotion. Among all of that information this website also provides the online reservation for their customer. Figure 1.2 is the example of the online reservation form for their restaurant.[2]

The strength about their system is their let the customer to choose where they want their table either at the lounge or at the theatre. Beside that they also leave a space for their customer to put their special request if any. The most interesting part at their online reservation system is they ask their customer to put a verification word to verify if they are really the one who reserves the table.

Unfortunately, this system are required their customer to put the date and time of reservation by themselves. They do not provide the calendar and the available time for reservation.

The online reservation system for Horizon Garden is implementing the function for the special request from this website. This is for the customer to request something to the restaurant for an example if they want the table to be prepared as the VIP table they can put them under the special request.

1.2.3. DJU Catering

Figure 1. 3: DJU catering online reservation

DJU catering is an experienced company in having help by several skilled, and energy. DJU catering ready to receive food booking cooked and dry for councils such as wedding, charity night banquet, festivals, meeting an others function. DJU catering also prepare for customer to make their ordering via online. Figure 1.3 shows the form for customer orders.[3]

To make an online ordering at DJU catering restaurant, customers should insert all the ordering information such as menu, quantity, date and other.
They are also required to insert their personal information such as name, contact number, email, and address. All the information will be saved in the DJU catering restaurant database for the record and the DJU catering will call the customer for the confirmation.

The RORS will be implemented the function for the calendar that are been used by this website. This is for the customer to looking for the suitable date to make the reservation.

1.3. Current System and Limitations

The current system in overall was managed manually. Whenever the customers want to make the reservation for the restaurant thy need to go there and meet with the staff to place the reservation or they can make a phone call. Then the staff will record their name date of reservation, number of people, function, and others information regarding the reservation in the logbook.

To check for the available table and time for the reservation, the staffs need to check on the logbook and go the page for the specific date and time. If the request date and time that are request for the customer for reservation are not available the staff will offer for the other date and time if the customer still want place the reservation. Otherwise the customer will go back empty handed or they will go to other restaurant.

There are many limitation and problems that arise by using this method. If the first time customers want to make the reservation at the restaurant and they are coming at the time the restaurant having a break it will be a burden for the customer especially if they come from far away.

Beside that if they are want to make the reservation by making a phone call during a work hours, the chances for their call to be answered by the staff are small because the he staff will not answer the phone call during busy time. This happen because, at this restaurant there are no person in charge to answer the phone call because all the staff includes the supervisor will need to served customer during peak hours.

1.4. Methodology

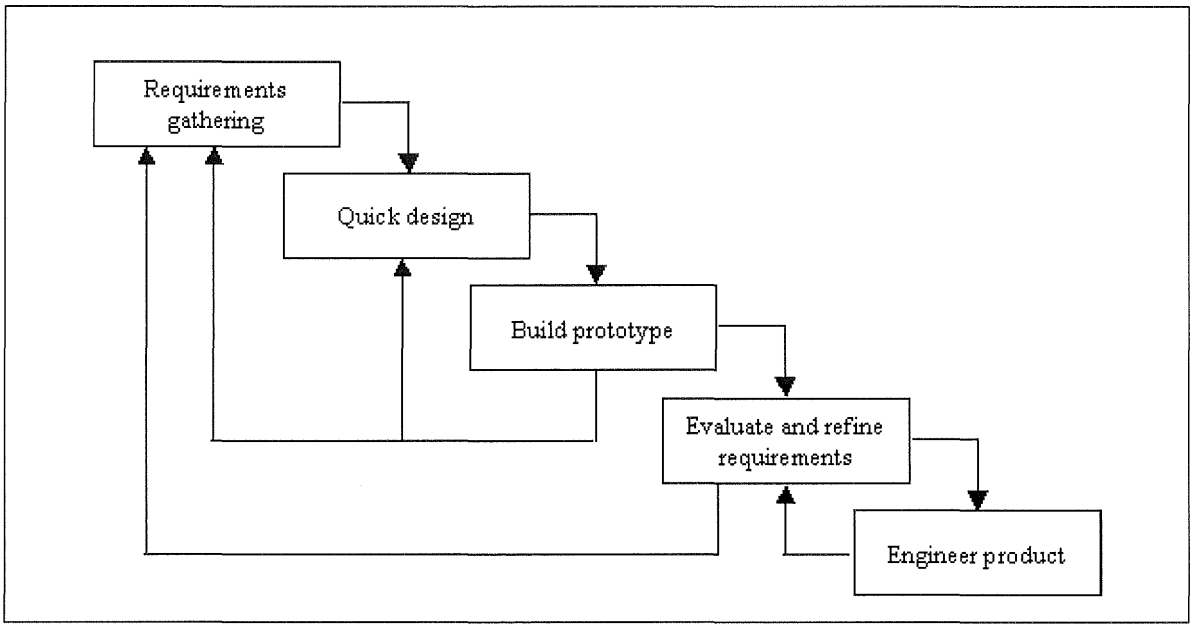


Figure 1. 4: Prototyping Model

The methodology has been implied in the project development is the prototyping model. Figure 1.10 shows the prototyping model. Prototyping is one of the software development lifecycle beside waterfall, agile and others. This method can reduce risk and limit expenses costs.

1.4.1. Requirement Gathering

Requirements are gathered during the meeting between the developer and the client. The important element, input and output are also identified. The client of this system is the manager of the Horizon Garden Restaurant. The information regarding the system is being gathered so that the system that will be developed will meet the client's requirement. The Software Requirement Specification (SRS) document will be produced in this phase. The SRS include product description, interface requirements, software product features and requirement traceability.

Table 1.2: Information need to be gathered

Source	Information gathering
Client	• Flow of the system (How the systems work?) • Data information of the system (What data should be included In the system?)
System	• Users of the system (How many privileges in the database?
Environment	• System environment (Which environment is compatible with the system?) E.g. web-based, Stand alone

1.4.2. Quick Design

This process come after the requirement gathering which is the developer will design initial prototype which include user interfaces. They focus on a representation of those aspects of the system that are visible to the client which is input and output approaches. This is where the Software Design Document (SDD) will be produced. He design description and details are included in SDD.

1.4.3. Build Prototype

The quick design phase leads to the build of a prototype phase. In this phase, developer starts to construct and develop the prototype to be shown to the customer. The prototype is build base on the requirement and the initial design. This is where the developer start codes the prototype.

1.4.4. Evaluate and Refine

This process comes after the prototype building process. The prototype then will be showed to the client to be evaluated and examined by the client. Client or end-user will provide the feedback on addition or changes in their requirement base on the prototype. In this phase, Software Test Result (STR) has been produced.

1.4.5. Engineer Product

These whole processes are repeated until there no more change or addition requested from the client during the evaluate and examine phase. After all process the system becomes the end product that will be sent to the client as the final product. If the client has agreed to the end product, the user manual will be produced.

1.5. Report Organization

This report consists of four parts as below:

Chapter 1 – Introduction

- In this part contain introduction, problem statement, objective, scope, previous work review, current system, methodology, and report organization.

Chapter 2 - Software Requirements Specification (SRS)

- In this part contain product description, interfaces requirements, software product features, requirement traceability.

Chapter 3 - Software Design Document (SDD)

- In this part contain system overview, system states and modes, system design description, and database design.

Chapters 4 – Implementation and Testing

- In this is the Implementation and Testing phase. This phase test case data collected base on the each use case of the system.

Chapters 5 – Result and Analysis

- This chapter is the list of the result and analysis of the system proposed.

Chapters 6 – Conclusion

- Concludes all the chapters and the recommendations for future researchers.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1. Product Description

This part explains the functions and requirement for the RORS. In this part, it describes the base of the developers to acknowledge what are expected from the system and the description on the process of the components cooperates with each other in the environment.

2.1.1. Product Perspective

The RORS is a web-based system which is can be accessed anywhere and anytime using as long as there is an internet connection. The customer can make their online reservation through this system. The customer also can view the available table and menu before making the reservation. The reservation then goes into the database that can be view by the staff and manager. Staff can manage the data which is they can add, delete and cancel reservation if needed. Staff also can view the available table for booking at certain date and time for the walk in reservation.

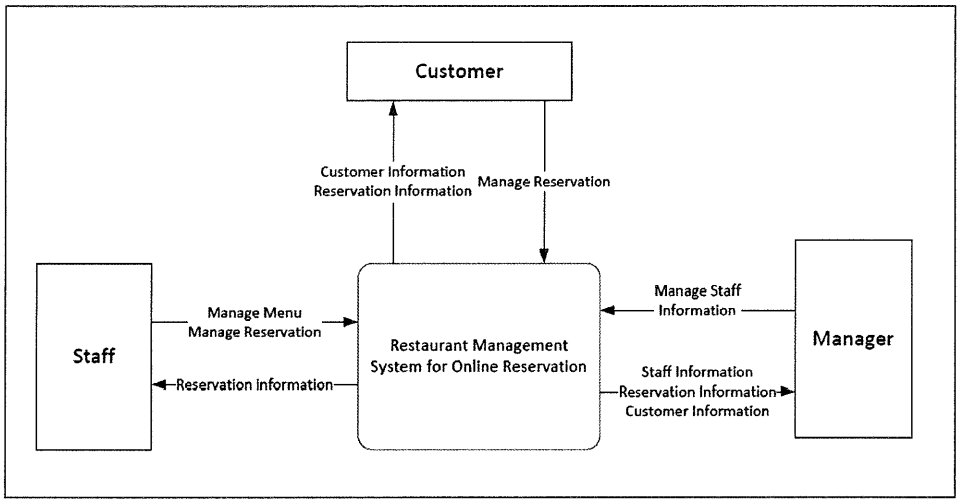


Figure 2. 1 : Context Diagram for RORS

2.1.2. Product Function

RORS start when customer login into the system to make the reservation. The user will be prompt to their perspective page if the password and username is entering correctly. There are different privileges in the system for the staff, manager and customer. For customer they can only place reservation and cancel their reservation only while, staff and manager are able to update data such as menu, date and time beside cancel the reservation.

The process of manage data starts when there data need to be add, delete, and update in the system. When there are new menu or special promotion the staff will update their menu list for the customer to view. The data will be deleted if the time period of the promotion is ended. For the managers they will add or update the data regarding the staff information. Especially, if there are new staff. This data need to be update so that the staff will be able to use this system with their own id and password.

The process of search data starts when the staffs want to find the available table at the certain date and time. When there are walk-in customer come to make the reservation the staff will search the available table from the system to look for the available table at the particular date and time as requested by the customer. Figure 2.2 shows the function of the RORS system.

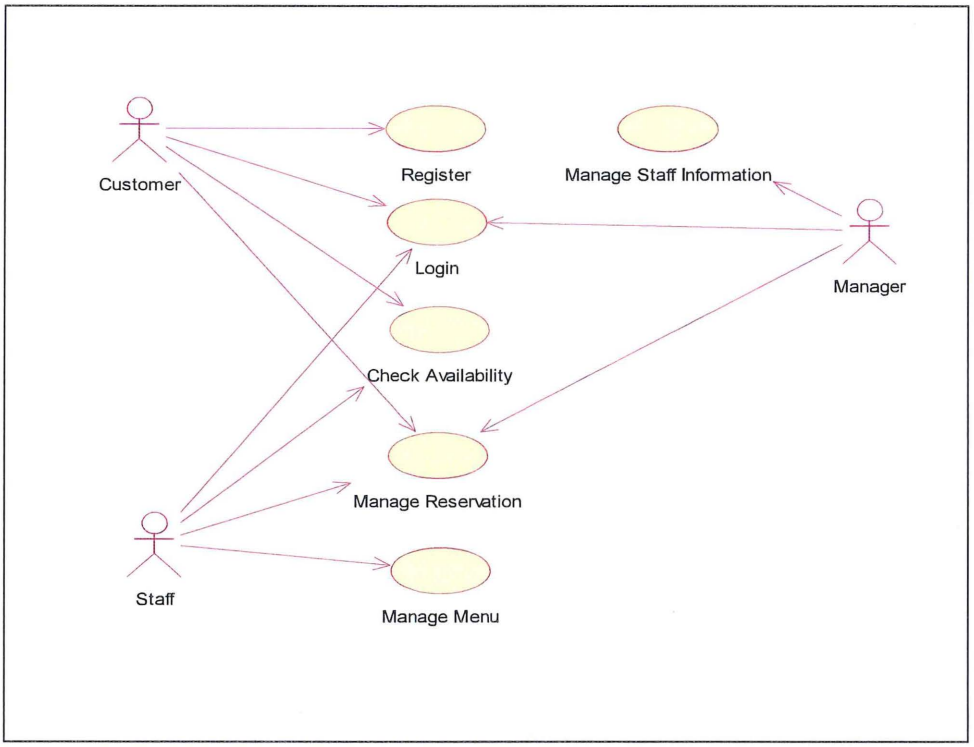


Figure 2. 2: Use case diagram for RORS system

2.1.3. User characteristics

Table 2. 1: User Characteristic

User	Education Level	Background Experience
Customer	At least SPM and above.	• Familiar with the web-based system. • Able to use the system without supervision
Staff	• At least SPM and above • Food and beverage (F&B) staff	• Fast learning
Manager	At least Diploma in Business Administration and above	• Has experience with the administration system. • Familiar with the web-based system.

2.1.4. Constraints

There are some several constraints during developing this project. The constraints are as the following:

- The user cannot access the website when the server is down.
- If the number of the concurrent user are more than the limited number the system cannot be accessed
- The system cannot be accessed when the internet connection is low.

2.2. Interface Requirement

2.2.1. User interfaces

2.2.1.1. Customer interfaces

Customer is ask to login before they can enter the system. If the customer do not have an account yet they will be ask to register for their account. Once the customer have login they will be displayed the main page of the system. At here customer can check the availability of the place for the reservation. Once the customer has confirmed there are available place for the reservation, they can proceed to the place reservation page. At here they are require to fill the information for the reservation. Then they will give the choice either to place an order or not. Once the customer has completed their reservation the system will display back to the user about their reservation information.

2.2.1.2. Staff interfaces

Once the staff login into the system, the system will prompt the staff to the reservation information page. At here they view all the reservation on that day. Beside that they also can search the reservation by entering the date and time. Other than that they also can cancel or delete the reservation. Other than date staff also is responsible to manage the menu data. They can add, delete, or update the menu if necessary.

2.2.1.3. Manager interfaces

After the manager login into the system, the main page for the manager will be displayed. At here they will be a list of button that will redirect the manager into the specific interface. The interface lists are staff information, customer information, reservation list, and menu list.

2.2.2. Hardware interface

The minimum hardware requirement to develop the RORS system:

Table 2. 2: Hardware Requirement

Hardware	Specification	Purpose
PC	Intel Core I3	Development process and documentation
Android Device	Sony Xperia Tipo Dual	For the test on the web browser in android platform
Pen Drive	16gb	Backup data

2.2.3. Software interface

Table 2. 3: Software Requirement

Software	Function
Windows 7 professional	Current operating system for the development process.
Adobe Dreamweaver	Development of RORS system.
MySQL	Database management of the system.
Rational Rose	Diagram and documentation
Microsoft Project	Gantt chart development
Java Netbeans	Development tools

2.3. Software Product Features

2.3.1. Use Case Register *[SRS-RORS-100-01]*

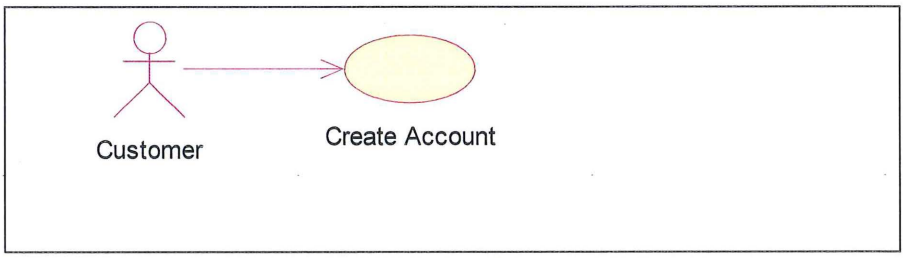


Figure 2. 3: Create Account Use Case Diagram

Table 2. 4: Use Case Create Account

Use Case ID	SRS-RORS-100-01
Brief Description	This use case initiated by the customer only. It enable user to register their account in order to use the online reservation system.
Actor	Customer
Pre-Condition	User want to register their account
Basic Flow	• Use case begin when the user click on the "Create Account" link. *[SRS-RORS-100-01-01]* • The user required to fill the registration form to create their account in the database. • The system validate the required information whether it valid or not.(**E1: Invalid Information)** • The system display login interface
Alternative Flow	None
Exception Flow	**E1: Invalid *Information[****SRS-RORS-100-01-02]*** • The system prompt the "Invalid Information" message • The system waits for the valid information. • The use case continues.
Post-Condition	Login interface
Constraints	None
Sequence Diagram	Refer Appendix: B-1.1 : Basic Flow B-1.2 : Exception Flow

	02] • The system display the error message says "Invalid username or password". • User re-enter their username and password • The process continued until the valid username and password are entered. • Use case continues.
Post-Condition	The user will be prompted to their interface: • Customer: Customer reservation interface. • Staff: Reservation Information interface. • Manager: Management system interface.
Constraints	None
Sequence Diagram	Refer appendix B-2.1: Basic Flow B-2.2: Exception Flow

2.3.3. Use Case Check Availability *[SRS-RORS-100-03]*

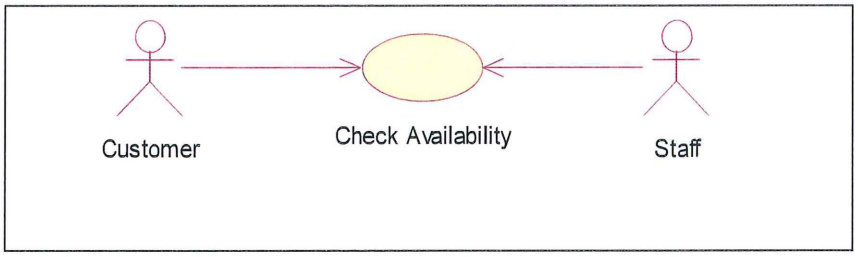


Figure 2. 5 : Check Availability Use Case diagram

Table 2. 6: Use Case Check Availability

Use Case ID	SRS-RORS-100-03
Brief Description	The use case is initiated by the customer and staff. It provides capability to view the availability table for reservation at the certain time and date.
Actor	Customer and staff
Pre-Condition	Check Availability button clicked
Basic Flow	• Use case start when the "Check Availability" button is click. *[SRS-RORS-100-03-01]* • Users enter date and time for reservation • The system displays the available and unavailable table for the reservation. *[SRS-RORS-100-03-02]* • The use case ends.
Alternative Flow	None
Exception Flow	None
Post-Condition	**Staff:** Reservation Information **Customer:** Place Reservation interface.
Constraints	None
Sequence Diagram	Refer appendix

	B-3.1: Basic Flow

2.3.4. Use Case Manage Reservation *[SRS-RORS-100-04]*

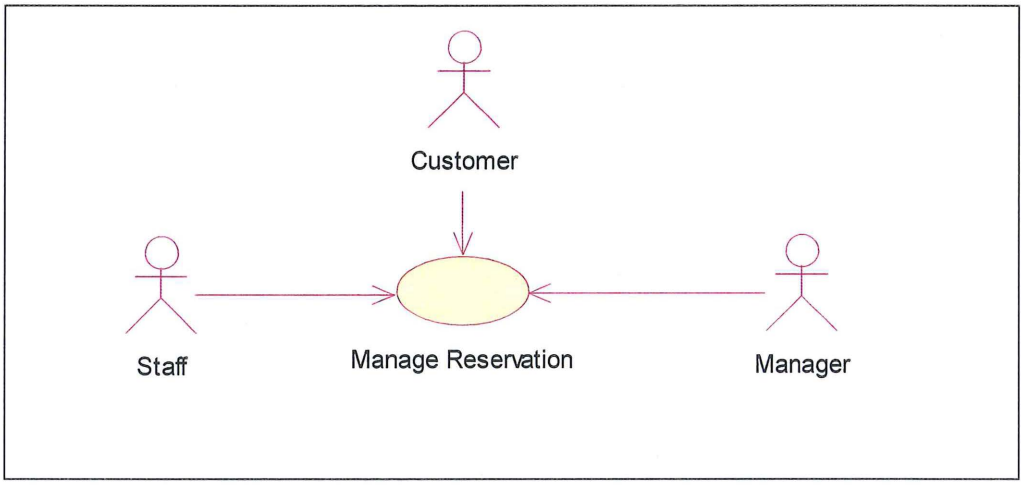


Figure 2. 6 Manage Reservation Use Case diagram

Table 2. 7: Use Case Manage Reservation

Use Case ID	SRS-RORS-100-04
Brief Description	This use case initiated by customer, staff and manager. Its enable the user to view the reservation, add new reservation, delete or cancel reservation, update and search reservation.
Actor	Customer, staff and manager
Pre-Condition	Users have login into the system
Basic Flow	• The use case start when the user click on the "Reservation Button" • Users then are prompted to the Reservation page. • At the reservation page users can do the following function: 1. Add new reservation

	(A1 : Add new reservation) 2. Delete or cancel the reservation **(A2 ; Delete or cancel reservation)** 3. Update the reservation information **(A3: Update reservation information)** 4. View reservation **(A4 : View reservation)** • The use case end.
Alternative Flow	**A1 : Add new reservation [*SRS-RORS-100-04-01]*** • This function is triggered by customer. • Customer click on "Add Reservation" button. • The system displays the "Add Reservation" page with a form. • Customer are requires to enter all the information regarding the reservation. • User saves the data. • The system sends the data into the database. • The system display the saved data • Use case continues. **A2 ; Delete or cancel reservation [*SRS-RORS-100-04-02]*** • This function can be triggered by customer, staff and manager. • The users click on "Cancel Reservation" button. • The system displays the delete reservation page. • Users select which data need to be deleted. • The systems ask for the confirmation. • User can choose to either to proceed or not. • The system deletes the data after confirmation.

	• Use case continues. **A3: Update reservation information [*SRS-RORS-100-04-03]*** • This function can only be triggered by the customer. • Customer click on the "Update Reservation" button. • The system displays the Update Reservation page. • Customer change the data that they want to change. • Customer saves the data. • System asks for the confirmation of the change made by customer. • System save the data after confirmation • Use case continues. **A4 : View reservation [*SRS-RORS-100-04-04]*** • This function can be triggered by the customer, staff, and manager. • Users click on the "View Reservation" button. • System displays all the reservation that has been made for customer interface and list of all the reservation information for the staff and manager interface. • Use case continues. **A5: Place Order [*SRS-RORS-100-04-05]*** • This function can be triggered by customer only. • Customer click on "Place Order" button. • System displays the Place Order page. • System will ask the customer to fill in the order form. • Users fill in the order form.

	• System saves the data. • Use case continues.
Exception Flow	None
Post-Condition	Main page
Constraints	None
Sequence Diagram	B-4.1: Basic Flow B-4.2:Alternative Flow B-4.3: Exception Flow

2.3.5. Use Case Manage Menu *[SRS-RORS-100-05]*

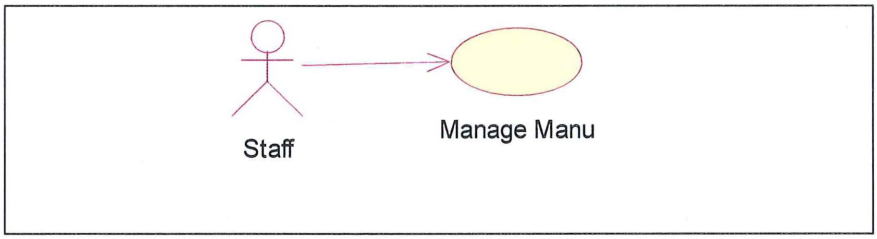


Figure 2. 7: Manage Menu use Case diagram

Table 2. 8: Use Case Manage Menu

Use Case ID	SRS-RORS-100-05
Brief Description	This use case initiated by the staff to manage their menu information. It provides the capability to add menu, delete menu and update menu.
Actor	Staff
Pre-Condition	Manage Menu button clicked.
Basic Flow	• The use case starts when the staffs click on the "Manage Menu" button. ***[SRS-RORS-100-05-01]*** • The staffs are able to do the following functions: a. Add menu **(A1: Add menu)**

	b. Delete menu **(A2: Delete menu)** c. Update menu **(A3: Update menu)** • Use case ends.
Alternative Flow	**A1: Add menu** ***[SRS-RORS-100-05-02]*** • The staffs click on the "Add menu" button. The system displays the Add Menu form. • The staff enter the new menu information • The staff clicks on the "Save" button". • The system displays a message says "Menu has been added". • The use case continues. **A2: Delete menu.** ***[SRS-RORS-100-05-03]*** • The staffs click on the "Delete Menu" button. The system the display the Delete Menu page. • The staff select the menu to be deleted from the list • The system asks for the confirmation. • The staff were asks to confirm or not • The system shows "Menu has been deleted". • The use case continues. **A3: Update menu.** ***[SRS-RORS-100-05-04]*** • The staffs click on the "Update Menu" button. The system shows Update Menu page. • The staff selected which menu need to be update

	from the list. • Display the selected data information. • The staffs change the data that is needed to be change. • The staffs save the update. • The system asks for the confirmation. • Staffs confirm the update. • The system displays the menu that has been update. • Use case continues.
Exception Flow	None
Post-Condition	Menu page
Constraints	None
Sequence Diagram	Refer appendix B-5.1: Basic Flow B-5.2: Alternative Flow

2.3.6. Use Case Manage Staff Information *[SRS-RORS-100-06]*

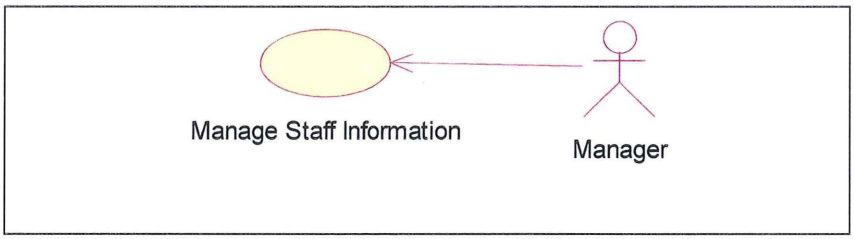


Figure 2. 8 : Manage Staff information use case

Table 2. 9: Use Case Manage Staff Information

Use Case ID	SRS-RORS-100-06
Brief Description	This use case can only initiated by the manager. Its enable the manager to view, add, delete, and update staffs information.
Actor	Manager
Pre-Condition	Main page for manager is display.
Basic Flow	• The use case starts when the manager clicks on the "Manage Staff information" button. ***[SRS-RORS-100-06-01]*** • The manager are able to do the following functions: a. Add **(A1:Add new staff)** b. Delete **(A2:Delete staff Information)** c. Update (**A3:Update staff information**) d. View **(A4: View staff information)** e. • The use case ends
Alternative Flow	**A1: Add new staff *[SRS-RORS-100-06-02]*** • The manager clicks on the "Add Staff" button. The system shows the Add New Staff page with a form • The manager enters the information about the new

	staff. • The manager saves the information. • The information is sent to the database by the system. • Use case continues. **A2: Delete staff Information.** ***[SRS-RORS-100-06-03]*** • The manager clicks on the "Delete Staff" button the system display the Delete Staff page. • The manager selects the staffs that are needed to be deleted. • The system asks for the confirmation. • The manager confirms the action. • The use case continues. **A3:Update staff information** ***[SRS-RORS-100-06-04]*** • The manager clicks on the "Update Staff" information. The system display Update Staff page. • The manager selects the staff to be updated. • The system shows the selected staff information. • The manager change the information that's need to be updated • The manager save the update • The system asks for the confirmation. • The manager confirms the action. • The system saves the information to the database and displays the updated information.

	• The use case continues **A4: View staff information *[SRS-RORS-100-06-05]*** • The manager clicks on the "View Staff" button. The system display all the list of staff • The manager clicks on the name of the staff • The system will display the information about the selected staff. • Use case continues.
Exception Flow	None
Post-Condition	Main page
Constraints	None
Sequence Diagram	Refer appendix B-6.1: Basic Flow B-6.2 : Alternative Flow

2.4. Requirement Traceability

Requirement	Description
1. ***SRS-RORS-100-01-01***	User click on the "Create Account" button.
2. ***SRS-RORS-100-01-02***	System check the information filled by customer either it is valid or not.
3. ***SRS-RORS-100-02-01***	User click on the "Login" button
4. ***SRS-RORS-100-02-02***	System check for valid username and password
5. ***SRS-RORS-100-03-01***	User click on "Check Availability"
6. ***SRS-RORS-100-03-02***	User enter date and time to search for availability
7. ***SRS-RORS-100-04-01***	User click on "Add Reservation" button
8. ***SRS-RORS-100-04-02***	User click on "Delete Reservation" button
9. ***SRS-RORS-100-04-03***	User clicks on "Update Reservation"
10. ***SRS-RORS-100-04-04***	User click on "View Reservation" button
11. ***SRS-RORS-100-04-05***	Customer click on "Place Order" button
12. ***SRS-RORS-100-05-01***	Staff click on "Manage Menu" button
13. ***SRS-RORS-100-05-02***	Staff clicks on "Add Menu"
14. ***SRS-RORS-100-05-03***	Staff click on "Delete Menu" button
15. ***SRS-RORS-100-05-04***	Staff click on the "Update Menu" button
16. ***SRS-RORS-100-06-01***	Manager click on "Manage Staff Information" button
17. ***SRS-RORS-100-06-02***	Manager click on "Add Staff" button
18. ***SRS-RORS-100-06-03***	Manager click on "Delete Staff" button

19. ***SRS-RORS-100-06-04***	Manager click on "Update Staff" button
20. ***SRS-RORS-100-06-05***	Manager click on "View Staff" button

CHAPTER 3

SOFTWARE DESIGN DOCUMENT (SDD)

3.1. System Overview

This part of the document explains the design for the RORS system based on the requirement. All the design in this part is present in SDD which is referred from SDD document. [6]

The user which is customer, staff and manager are interacting with the system. The context diagram of RORS below show the interaction between users and the system. The users are requires to login before they can access the system. For customer, they are able to manage reservation, register account and check availability. For staff, they are able to manage menu, check availability and manage menu. For manager, they are able to manage reservation and staff information.

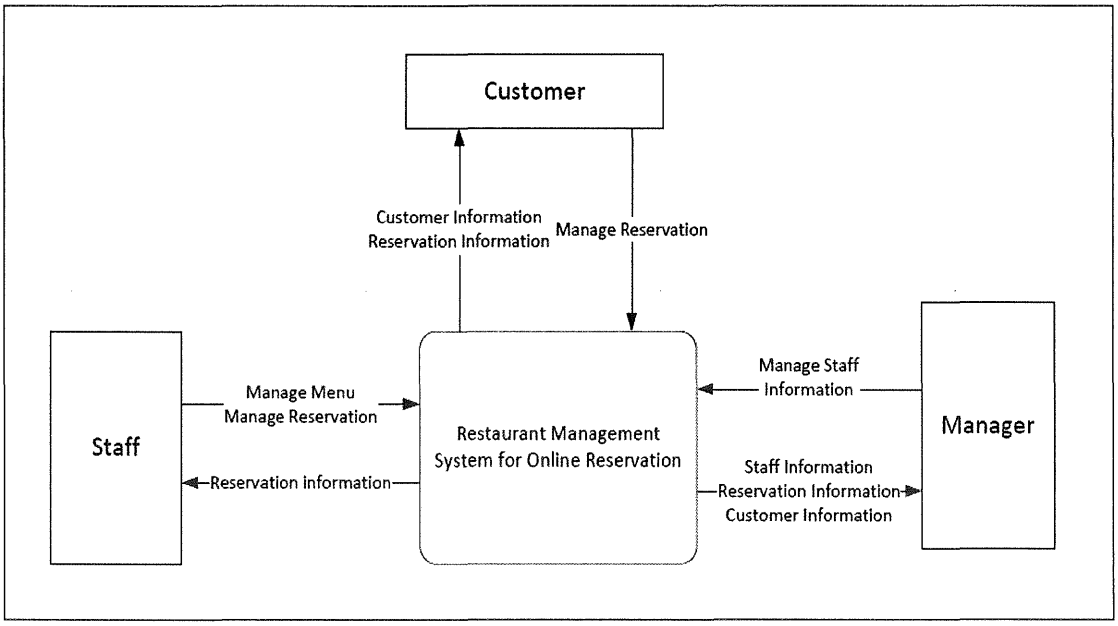


Figure 3. 1 : Context Diagram for RORS

3.1.1. System Architecture

This part identifies the internal organizational structures which is including the details description of the relationship between subsystems.

3.1.1.1. Static Organization

Static organization consists of the list of the package of the system which is registration, ordering, information, and reservation.

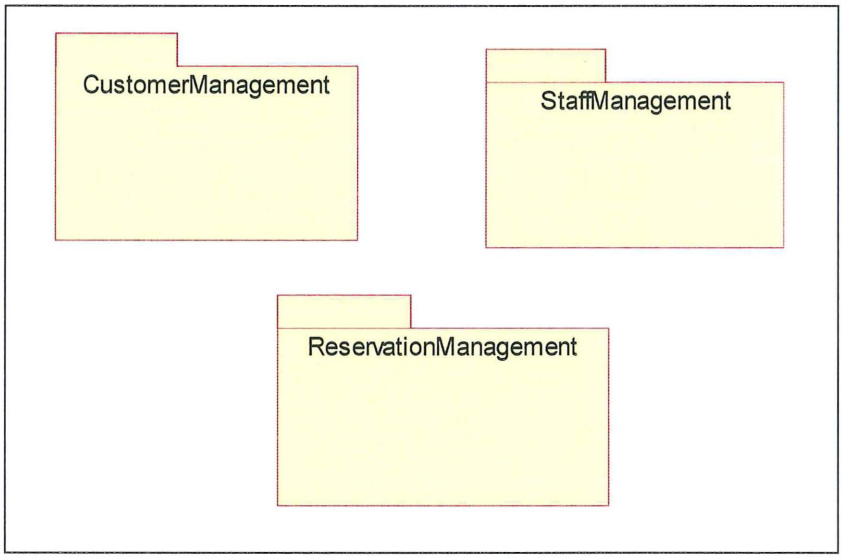


Figure 3. 2: Static Organization for RORS

This section describes the detail for each package which is registration, ordering, information, reservation.

1. **Customer Management**

   This package responsible for manage customer information from the customer who want to use this system.

   a. Class Customer Info
   b. Class Customer Controller
   c. Class Customer View

2. **Staff Management**

   This package responsible to manage the staff information which is use by manager

   a. Class Staff Info
   b. Class Staff Controller
   c. Class Staff View

3. **Reservation Management**

   This package is responsible to manage the reservation data

   a. Class Reservation Info
   b. Class Reservation Controller
   c. Class Reservation View

3.1.1.2. Dynamic Organization

Component diagram which is exist in the system. Figure 3.3 shows the component diagram.

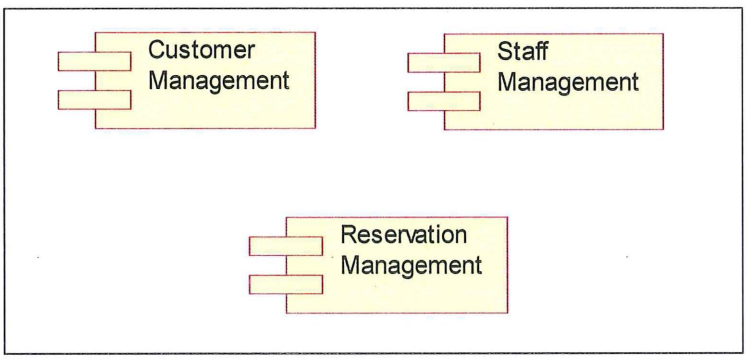


Figure 3. 3: Component Diagram for RORS

3.1.1.3. Subsystem Interface

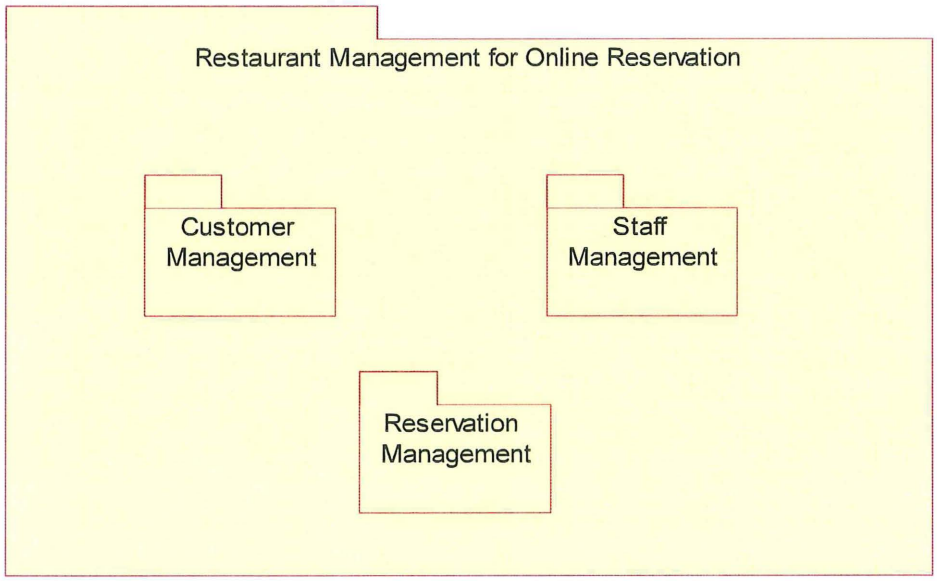


Figure 3. 4: Subsystem Interface

3.2. SYTEM STATES AND MODES

This section describes the state diagram for RORS. Figure 3.6 shows the state diagram for RORS. This shows the summary process from login activity until logout.

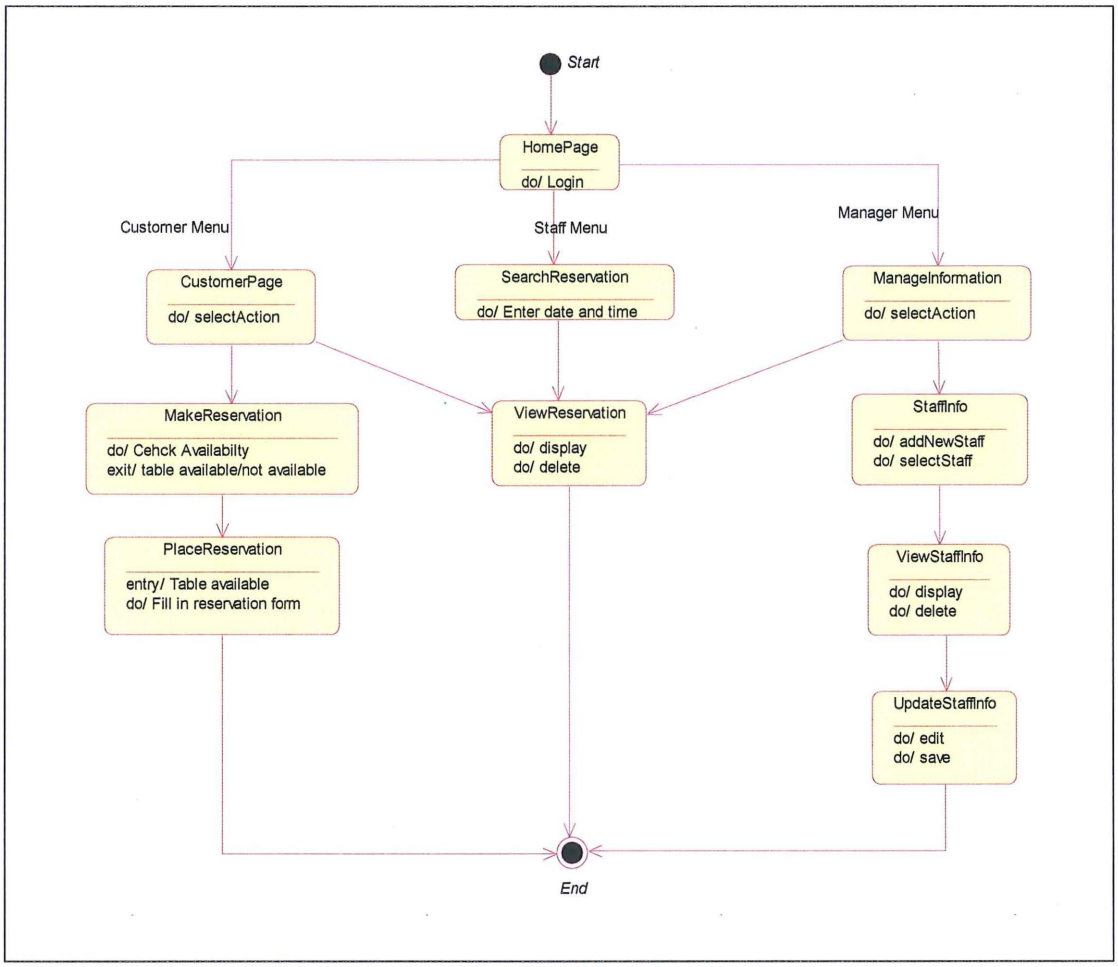


Figure 3. 5: State Diagram for RORS

The user needs to login to access his system. For customer menu they are able to search for availability, place a new reservation and view their reservation. While for staff they are only able to search for the reservation by entering the date and time, they are also able to view the reservation. For manager, they are able to manage the staff information such as add new staff, update, and delete the staff information. Manager also able to view the reservation information.

3.3. System Design Description

3.3.1. System Design

RORS consists of four packages which are divided into Registration, Ordering, Information, and Reservation.

3.3.1.1. Customer Management Subsystem

The internal part of the Registration subsystem include the package diagram of Registration subsystem with the classes diagram hide inside the package is shown in Figure 3.6.

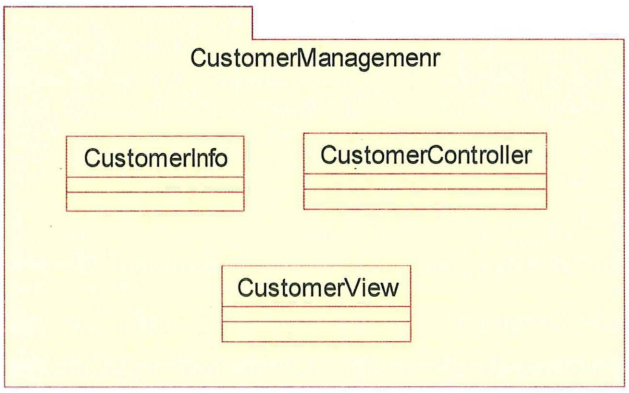


Figure 3. 6: Visibility of Customer Management

3.3.1.2. Staff Management Subsystem

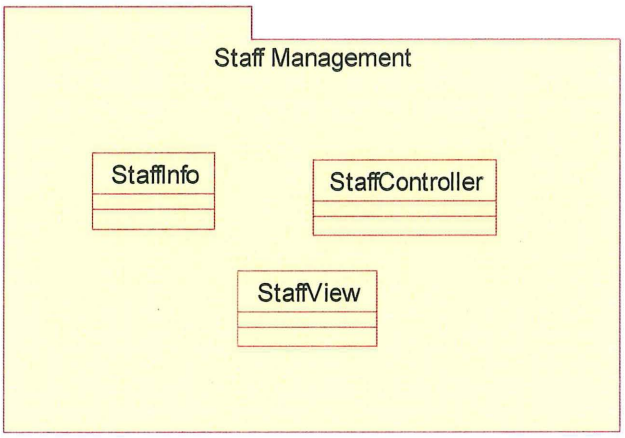


Figure 3. 7: Visibility of Staff Management

3.3.1.3. Reservation Management Subsystem

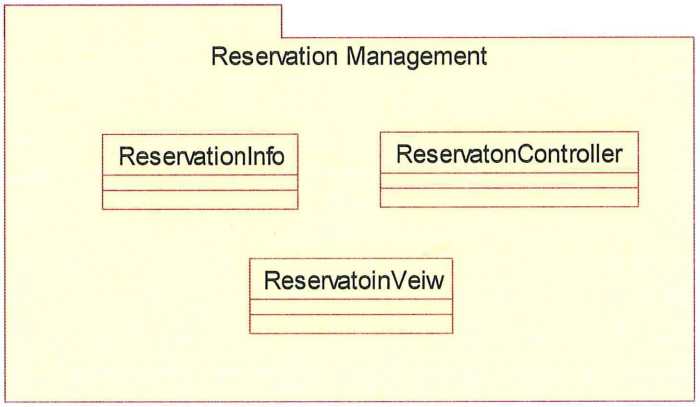


Figure 3. 8: Visibility of Reservation Management

3.3.2. Detailed design

This part divided into the following paragraph and subparagraph to describe the detailed design.

3.3.2.1. Customer Management Subsystem

Registration subsystem is shown in Figure 3.11.This subsystem need to describe the relationship among the other classes that are exists in this subsystem. Registration subsystem consists of Registration Info, Registration Controller, and Registration Interface.

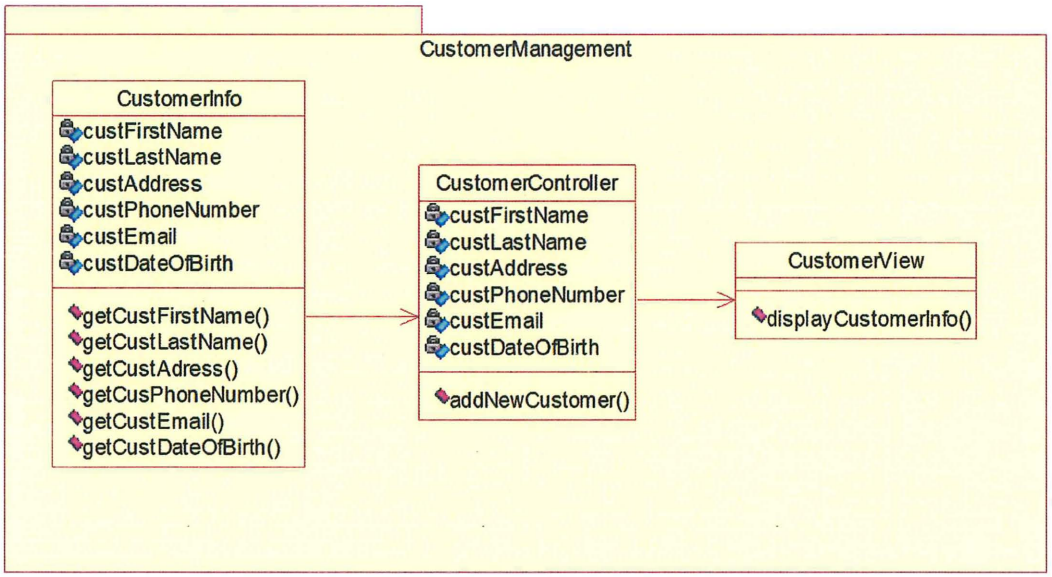


Figure 3. 9: Customer Management detail design

1. **Customer Info class**

The purpose of this class is to insert the new information of the new customer to create an account for them all the information will be saving into the customer database. With an account the customer can access the system to make the reservation.

i. Class Registration Info Design

a. Input/ Output data elements

List of input and output data elements:

Input: custFirstName, cusLastName, custDateOfBirth, custAddress, custPhoneNumber, custEmai

Output: displayCustomerInfo ()

b. Local data elements

Table 3. 1: Local data elements for custFirstName

Name	custFirstName
Description	Customer's first name
Data Type	string
Precision/resolution	-

Table 3. 2: Local data elements for custLastName

Name	custLastName
Description	Customer's last name
Data Type	string
Precision/resolution	-

Table 3. 3: Local data elements for custDateOfBirth

Name	custDateOfBirth
Description	Customer's birthday

Data Type	date
Precision/resolution	-

Table 3. 4: Local data elements for custAddress

Name	custAddress
Description	Customer's home address
Data Type	string
Precision/resolution	-

Table 3. 5: Local data elements for custPhoneNumber

Name	custPhoneNumber
Description	Customer's contact number
Data Type	integer
Precision/resolution	-

Table 3. 6: Local data elements for custEmail

Name	custEmail
Description	Customer's email address
Data Type	string
Precision/resolution	-

c. Algorithm

This section states the purpose and describes in detail the algorithms of this class.

Class type : Model class

Responsibility : Add new customer information

Attributes : custFirstName, custLastName, custDateOfBirth, custAddress, custPhoneNumber, custEmail.

Method : getCustFirstName (), getCustLastName (), getCustDateOfBirth (), getCustAddress (), getCustPhoneNumber (), getCustEmail().

1. **getCustFisrtName *()***

   Responsibility : Read customer first name

   Input Parameter : firstName

   Output Parameter : Display customer first name

   Algorithm : BEGIN
   Read customer first name
   Display
   END

2. ***getCustLastName()***

   Responsibility : Read customer last name

   Input Parameter : lastName

   Output Parameter : Display customer last name

   Algorithm : BEGIN
   Read customer last name
   Display
   END

3. *getCustDateOfBirth()*

Responsibility : Read customer date of birth

Input Parameter : birthday

Output Parameter : Display date of birth

Algorithm : BEGIN

Read customer birth of date

Display

END

4. *getCustAddress()*

Responsibility : Read customer home address

Input Parameter : address

Output Parameter : display address

Algorithm : BEGIN

Read customer address

Display

END

5. *getCustPhoneNumber()*

Responsibility : Read phone number

Input Parameter : phoneNumber

Output Parameter : Display phone number

Algorithm : BEGIN

Read new phone number

Display

END

6. *getCustEmail()*

Responsibility : Read customer email address

Input Parameter : Email

Output Parameter : Display email address

Algorithm : BEGIN

Read new email address

Display

END

2. **Customer Controller Class**

This class is created to control the connection between view class and main class

i. **Class Customer Controller Design**

This section specifies the design of the CustomerController.

a. **Input/ Output data elements**

List of input and output elements:

Input : None

Output : None

b. **Local data elements**

None

c. **Algorithm**

This section states the purpose and describes in detail the algorithms of the class:

Class Type : Controller class

Responsibility : Call the object from the customer info to be add into database

Attributes : None

Methods : addNewCustomer ()

1. addNewCustomer()

Responsibility : Add new customer

Input Parameter : All attributes from Customer Info class

Output Parameter : Display Mail Information

Algorithm : BEGIN
Add new customer information
Save
Display
END

3. Customer View Class

This class is created to be the interface based on the Customer Info class. It used to be viewed by the user and display the customer information.

i. Class Customer View Design

This specifies the design of Customer View class.

a. Input/ Output data elements

List of input and output data elements:

Input : None

Output : None

b. Local data elements

None

c. Algorithm

This session describes the detailed algorithms of this class

Class Type : View class

Responsibility : Display customer information on interface

Attributes : None

Methods : display CustomerInfo ()

1. displayCustomerInfo()

Responsibility : Display all customer information on interface

Input parameter : None
Output parameter : None
Algorithm : BEGIN
Display customer information
END

3.3.2.2. Staff Management Subsystem

Staff management subsystem is shown in Figure 3.12. This subsystem need to describe the relationship among other classes.

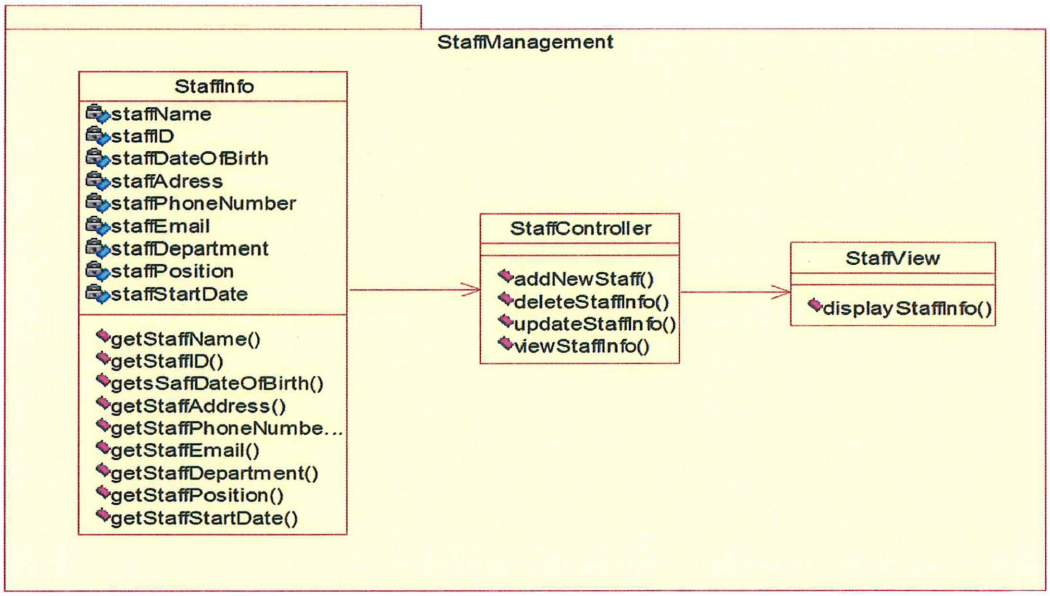


Figure 3. 10: Staff Management subsystem detailed design

1. **Staff Info Class**

The purpose of this class is to add, delete, update and view the staff information.

i. Class Staff Info Design

This session specifies the design of Staff Info.

a. Input/ Output data elements

List of input and output data elements:

Input : staffName, staffID, staffDateOfBirth, staffAddress, staffPhoneNumber, staffEmail, staffDepartment, staffPosition, staffStartDate

Output : geSstaffName (), geStaffID (), geStaffDateOfBirth (), geStaffAddress (), geStaffPhoneNumber (), geStaffEmail (), getSaffDepartment (), staffPosition, staffStartDate

b. Local data elements

Table 3. 7: Local data definition for staffName

Name	staffName
Description	Staff's name
Data Type	String
Precision/resolution	None

Table 3. 8: Local data definition for staffID

Name	staffID
Description	ID of the staff
Data Type	String
Precision/resolution	None

Table 3. 9: Local data definition for staffDateOfBirth

Name	staffDateOfBirth
Description	Staff's date of birth
Data Type	Date
Precision/resolution	None

Table 3. 10: Local data definition for staffAddress

Name	staffAddress
Description	Home address of the staff
Data Type	String
Precision/resolution	None

Table 3. 11:Local data definition for staffPhoneNumber

Name	staffPhoneNumber
Description	Contact number of the staff
Data Type	String
Precision/resolution	None

Table 3. 12: Local data definition for staffEmail

Name	staffEmail
Description	Email address of the staff
Data Type	String
Precision/resolution	None

Table 3. 13: Local data definition for staffEmail

Name	staffDepartment
Description	Department where the staff work
Data Type	String
Precision/resolution	None

Table 3. 14: Local data definition for staffPosition

Name	staffPosition
Description	Staff's position
Data Type	String
Precision/resolution	None

Table 3. 15: Local data definition for staffStartDate

Name	staffStartDate
Description	The first date of the staff start working
Data Type	String
Precision/resolution	None

c. Algorithm

These sections describe the purpose and describe in the detail the algorithm of this class.

Class Type : Model Class

Responsibility : Get the staff information

Attributes : staffName, staffID, staffDateOfBirth, staffAddress, staffPhoneNumber, staffEmail, staffDepartment, staffPosition, staffStartDate

Methods : getStaffName (), getStaffID (), getStaffDateOfBirth (), getStaffAddress (), getStaffPhoneNumber (), getStaffEmail (), getStaffDepartment (), getStaffPosition (), getStaffStartDate ()

1. getStaffName()

Responsibility : Read staff name

Input parameter : Name

Output parameter : Display staff name

Algorithm : BEGIN

Read staff name

Display

END

2. *getStaffID()*

Responsibility : Read staff ID

Input parameter : ID

Output parameter : Display staff ID

Algorithm : BEGIN

Read staff ID

Display

END

3. *getStaffDateOfBirth()*

Responsibility : Read staff date of birth

Input parameter : Birthdate

Output parameter : Display

Algorithm : BEGIN

Read staff date of birth

Save

END

4. *getStaffAddress()*

Responsibility : Read staffs address

Input parameter : Address

Output parameter : None

Algorithm : BEGIN

Read staffs address

Save

END

5. *getStaffPhoneNumber()*

Responsibility : Read staff phone number

Input parameter : Phone number

Output parameter : None

Algorithm : BEGIN

Read staff phone number

Save

END

6. *getStaffemail()*

Responsibility : Read staff email

Input parameter : Email

Output parameter : None

Algorithm : BEGIN

Read staff email

Save

END

7. *getStaffDepartment()*

Responsibility : Read staff department

Input parameter : Department

Output parameter : None

Algorithm : BEGIN

Read staff department

Save

END

8. ***getStaffPosition()***

Responsibility : Read staff position

Input parameter : Position

Output parameter : None

Algorithm : BEGIN

Read staff position

Save

END

9. ***getStaffStartDate()***

Responsibility : Read staffs start date

Input parameter : **Start**

Output parameter : **None**

Algorithm : BEGIN

Read staff start date

Save

END

2. Staff Controller Class

This class is created to connect and control the class between view class and the main class in the subsystem.

i. Class Staff Controller Design

This section specifies the design of Staff Controller.

a. **Input/ Output**

List of input and output:

Input : None

Output : None

b. **Local data elements**

None

c. **Algorithm**

This section states the purpose and describes in detail the algorithm of this class.

Class Type : Controller class

Responsibility : Manage the data that is inserting by the user from Staff info class.

Attributes : None

Methods : addNewStaff (), deleteStaffInfo (), updateStaffInfo (), viewStaffInfo ().

1. addNewStaff()

Responsibility : Add new data from staff info into database

Input parameter : All input parameter from staff info

Output parameter : None

Algorithm : BEGIN

Add new staff information

Save

END

2. *deleteStaffInfo()*

Responsibility : Remove staff information from the system

Input parameter : All input parameter from staff info class

Output parameter : None

Algorithm : BEGIN

Select data to be delete

Delete

END

3. *updateStaffInfo()*

Responsibility : Update staff information from database

Input parameter : All input parameter from staff info class

Output parameter : None

Algorithm : BEGIN

Read staff info data

Change staff info

Save

END

4. *viewStaffInfo()*

Responsibility : Display the staff info on interface

Input parameter : None

Output parameter : None

Algorithm : BEGIN

Display the staff info on interface

END

3. **Staff View Class**

This class is created to connect and control the class between view class and the main class in the subsystem.

i. **Class Staff View Design**

This section specifies the design of Staff View.

a. **Input/ Output**

List of input and output:

Input : None

Output : None

b. **Local data elements**

None

c. **Algorithm**

This section states the purpose and describes in detail the algorithm of this class.

Class Type : View Class

Responsibility : Display all the staff information

Attributes : None

Methods : displayStaffInfo ().

1. displayStaffInfo ()

Responsibility : Display all the staff information

Input parameter : None

Output parameter : None

Algorithm : BEGIN

Display information on interface

END

Output : displayReservationInfo

b. Local data elements

Table 3. 16: Local data definition for custName

Name	custName
Description	Name of the customer who make the reservation
Data Type	String
Precision/resolution	None

Table 3. 17: Local data definition

Name	reserveNumber
Description	Number of reservation
Data Type	Integer
Precision/resolution	None

Table 3. 18: Local data definition reserveNumberOfPerson

Name	reserveNumberOfPerson
Description	Number of person for the reservation
Data Type	Integer
Precision/resolution	None

Table 3. 19: Local data definition for reserveDate

Name	reserveDate
Description	Date of reservation
Data Type	Date
Precision/resolution	None

Table 3. 20: Local data definition for reserveTime

Name	reserveTime
Description	Reservation time
Data Type	Integer
Precision/resolution	None

Table 3. 21: Local data definition for reserveEvent

Name	reserveEvent
Description	Purpose of the reservation
Data Type	String
Precision/resolution	None

Table 3. 22: Local data definition reserveSpecRequest

Name	reserveSpecRequest
Description	Special request for the reservation
Data Type	String
Precision/resolution	None

c. Algorithm

This section states the purpose and describes the algorithm for this class.

Class type : Model class

Responsibility : To get the reservation information from the customer

Attributes : custName, reserveNumber, reserveNumberOfPerson, reserveDate, reserveTime, reserveEvent, reserveSpecRequest

Method : getCustName (), getReserveNumber (), getReserveNumberOfPerson (), getReserveDate (),

getReserveTime (), getReserveEvent (), getReserveSpecRequest ()

1. *getCustName ()*

   Responsibility : Call customer name from customer info

   Input parameter : None

   Output parameter : None

   Algorithm : BEGIN

   Call name from customer info

   Display

   END

2. *getReserveNumber ()*

   Responsibility : To set the reserve number for the reservation.

   Input parameter : reserveNumber

   Output parameter : None

   Algorithm : BEGIN

   Set the number for the reservation

   Save

3. *getReserveNumberOfPerson ()*

   Responsibility : No read the number of person for the new reservation.

   Input parameter : NumOfPerson

   Output parameter : None

   Algorithm : BEGIN

   Read the number of person

   Save

   End

4. *getReserveDate ()*

   Responsibility : Read the reservation date

   Input parameter : reserveDate ()

   Output parameter : None

   Algorithm : BEGIN

   Read the reservation date

   Save

   END

5. *getReserveTime ()*

   Responsibility : Read the reservation time

   Input parameter : reserveTime

   Output parameter : None

   Algorithm : BEGIN

   Read the reservation time

   Save

   END

6. *getReserveEvent ()*

   Responsibility : Read the reservation event

   Input parameter : reserveEvent

   Output parameter : None

   Algorithm : BEGIN

   Read reservation event

   Save

   END

7. *getReserveSpecRequest ()*

Responsibility : To read the reservation special request.
Input parameter : Request
Output parameter : None
Algorithm : BEGIN
Read special request
Save

END

2. Reservation Controller Class

This class created to connect and control the view class with the main class in the subsystem.

a. Input/ Output data elements

The list of input and output data elements:

Input : None

Output : None

b. Local data elements

None

c. Algorithm

This section states the purpose and describes the algorithm for this class.

Class Type : Controller class

Responsibility : To add new reservation, delete reservation, update reservation, and view reservation.

Attributes : None

Methods : addNewReservation (), deleteReservation (), updateReservation, viewReservation.

1. *addNewReservation ()*

   Responsibility : Add new data of reservation

   Input parameter : All the attributes in Reservation Info class

   Output parameter : None

   Algorithm : BEGIN

   Read data from Reservation Info class

   Save

   END

2. *deleteReservation ()*

   Responsibility : Delete the reservation data

   Input parameter : None

   Output parameter : None

   Algorithm : BEGIN

   Remove the reservation data

   Save

   END

3. *updateReservation ()*

   Responsibility : To change the reservation information

   Input parameter : All attributes from Reservation Info data

   Output parameter : None

   Algorithm : BEGIN

   Call the reservation information

   Enter new data

   Save

   END

4. *viewReservation ()*

Responsibility : To call the reservation information

Input parameter : None

Output parameter : None

Algorithm : BEGIN
Call the reservation information
View
END

3. Reservation View Class

This class created for interface based on the Reservation Info class. It used to be viewed by the user.

i. Class Reservation View Design

This session specifies the design of Reservation View

a. Input/ Output data elements:

The list of input and output:

Input : None

Output : None

b. Local data elements

None

c. Algorithm

This section describes the detail of the algorithm for this class.

Class Type : View class

Responsibility : Display the reservation information

Attributes : None

Methods : displayReservationInfo ()

1. ***displayReservationInfo ()***

   Responsibility : Display the reservation list on the interface

   Input parameter : None

   Output parameter : None

   Algorithm : BEGIN

   Call all the data from reservation information

   Display

   END

3.4. Database Design

Database design is a process of developing a database design or data model that meet with the user requirement. In the RORS, there is a several database table which are customer, staff, reservation and menu.

Table 3. 23: CustomerInfo table

Name	Data Type	Primary Key	Foreign Key
Cust_ID	int	Yes	-
Name	varchar	-	-
Phone_number	varchar	-	-
Email	varchar	-	-
Password	varchar	-	-
Question	varchar	-	-
Answer	varchar	-	-
custUserName	varchar	-	-
custPassword	varchar	-	-

Table 3. 24: StaffInfo Table

Name	Data Type	Primary Key	Foreign Key
Staff_ID	integer	Yes	-
Firstname	varchar	-	-
Lastname	varchar	-	-
Age	integer	-	-
Address	varchar	-	-
Phone	varchar	-	-
Email	varchar	-	-
Password	varchar	-	-

Position	varchar	-	-
Image	image		

Table 3. 25: Reservation table

Name	Data Type	Primary Key	Foreign Key
Reserve_ID	integer	Yes	-
Name	varchar	-	-
Number_person	integer	-	-
Date	date	-	-
Time	integer	-	-
Request	varchar	-	-
Number_Table	integer	-	-
Category	varchar	-	-
Cust_ID	int	-	Yes

Table 3. 26: Menu table

Name	Data Type	Primary Key	Foreign Key
Menu_ID	integer	Yes	-
Dishes	varchar	-	-
Category	varchar	-	-
Price	money	-	-
Imagepath	varchar	-	-

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1. Introduction

This section will discuss about the structure of the system and the usage of IDE tools during the development of the system. The implementation activities include the system design and testing. System design is the main activities in the implementation phase. During implementation, the developer has to ensure that the system design is meet the user requirement before implementing into the system.

Implementation phase is an important phase in order to develop an effective system. In this phase, the developer will determine the tools that will be used to develop the system, starting from building interfaces to running the system with the free error until completing all functions that have been state in the previous chapter.

4.2. Tools And Technologies

Restaurant Online Reservation System is develops using Microsoft Visual Studio 2010 ASP.net Web Application. This system implemented ASP.net for the client and server side and VB.net as a back end code. SQL Server database is using to store the database information.

4.3. Database Connection

Through this phase, all the information will be stored in the Horizon database. This system consists of few table which are CustomerInfo, Menu, Reservation, and StaffInfo.

Figure 4.1 and 4.2 show the configuration between the SQL data source and table selection. The table selection will be configured differently depend on the SQL data source.

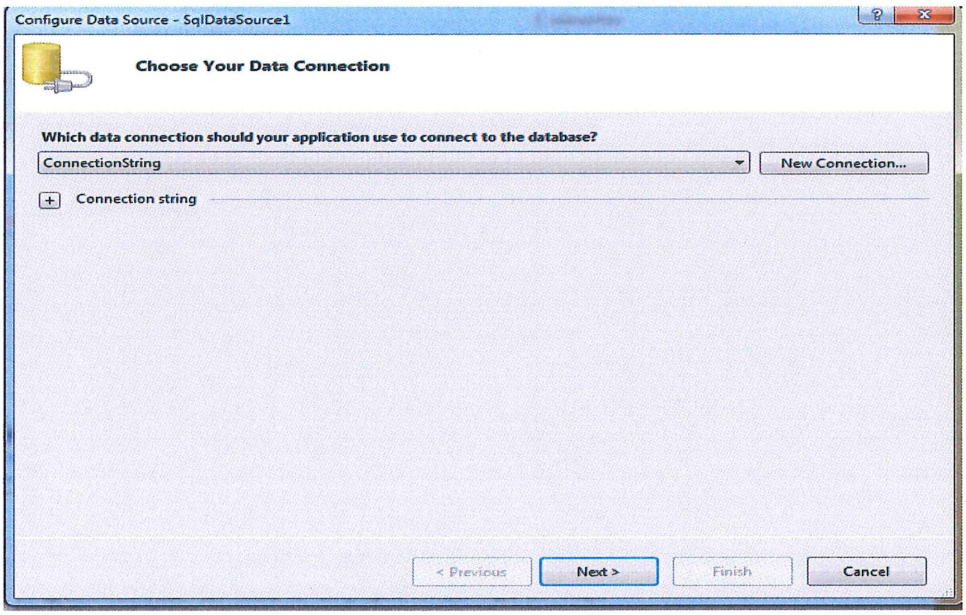


Figure 4. 1 : Configure SQL data source connection

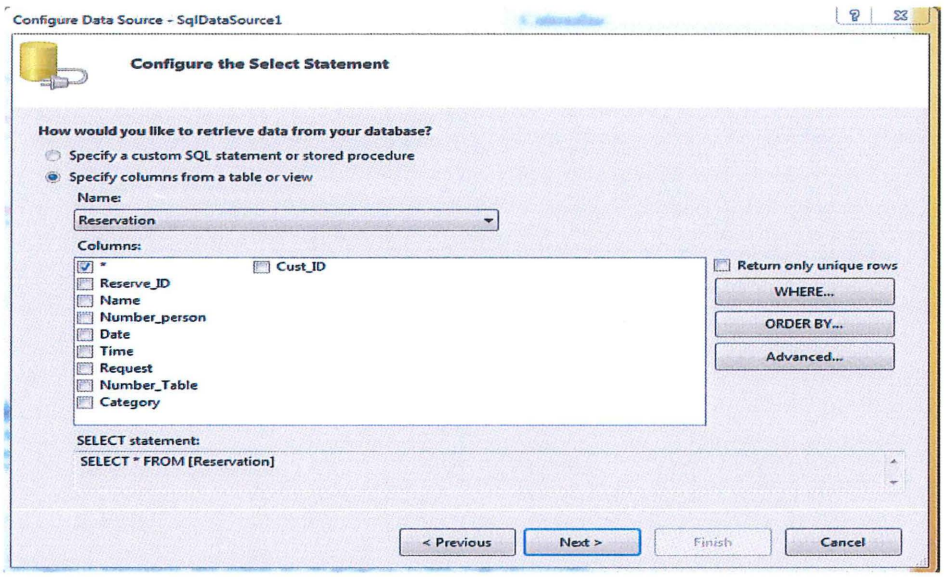


Figure 4. 2: : Select a table from database

4.4. System Function

This section will describe all the functions contain in this system based on the user.

4.4.1. Create Account

Figure 4.3 shows the Create Account for the customer to create the account. Users are required to fill in all the information requested in the form. For the staff they also need to login to the system so that the staff able to record the reservation information for the walk in reservation. The staff account is create by the manager as shown in Figure 4.4.

Figure 4. 3 : Customer create account page

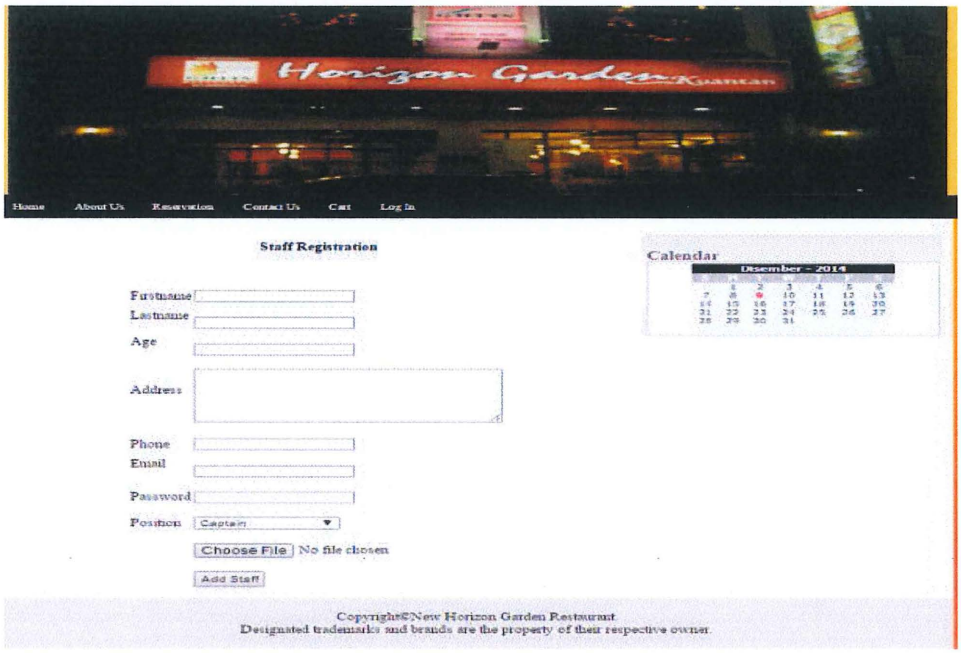


Figure 4. 4: Staff register page

4.4.2 Login

After user and staff already have an account they need to Login into the system using their email as a username and a password that already given to them. Once the user insert the username and password the system will check with the database that have the same username and password in their account. If the information match the user will be successfully login into the system. Once login, user can use the system according to their role which is customer, staff or admin. Figure below shows the login form for the users.

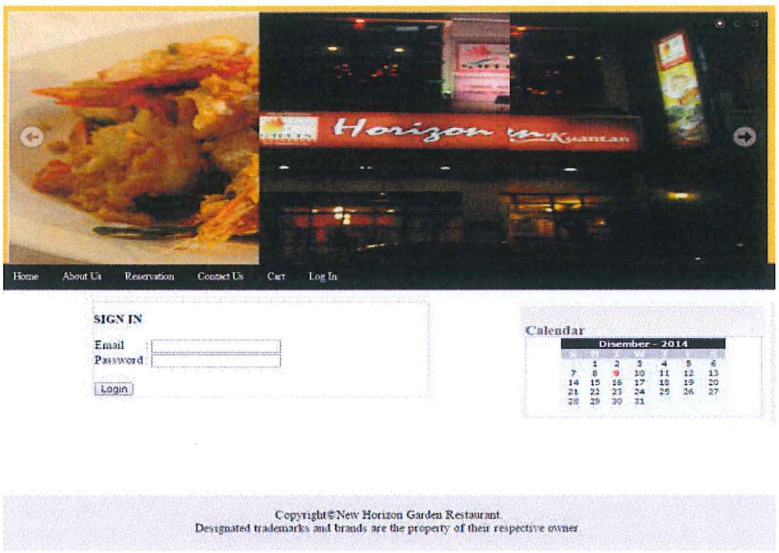


Figure 4. 5:Customer Login form

Figure 4. 6: Staff login form

4.4.3. Check Availability

This function for the customer and staff to be able them to check the availability of the table for the reservation at a certain date and time. Once, user enter the information the system check with database and display the amount of the table available for the certain date and time and user can choose place a reservation or back to home page. If they are not available the system will display the "FULL HOUSE" text. Then, user can choose to see for the next date and time or going back to the home page. Figure 4.7 shows the check availability form.

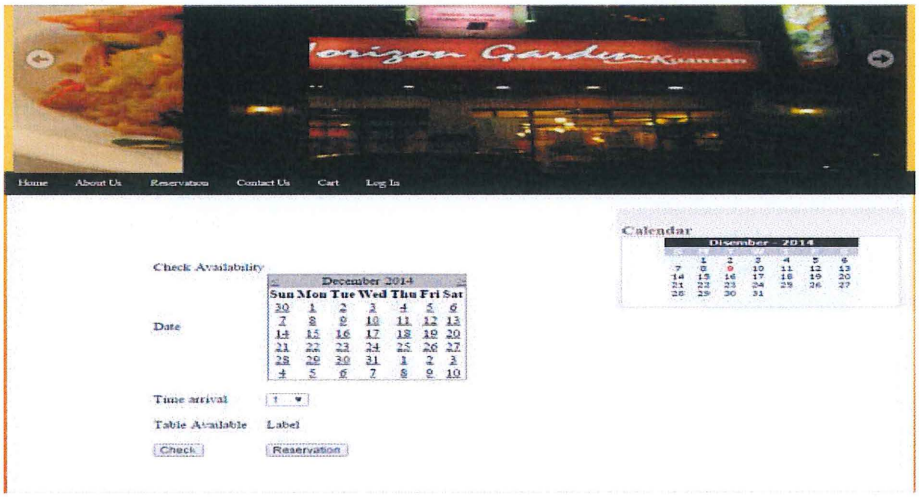


Figure 4. 7: Check availability form

4.4.4 Reservation

This is the main function of this system. After the availability checking customer are prompt to the next page which is reservation form to place the reservation. The reservation information will be stored in the table "reservation". This information then can be view by the customer so that they are able to manage the reservation like update or cancel their reservation. Figure 4.8 and 4.9 show the reservation and manage reservation page.

Figure 4. 8: Reservation form

		Reserve_ID	Cust_ID	Name	Number_person	Date	Time	Request	Number_Table	Category
Delete	Change	28	1005	lolo	23	12/13/2014	3		3	Lunch and Dinner
Delete	Change	14	1005	lala	23	12/2/2014	2	lala	3	Function
Delete	Change	15	1005	apis	24	11/13/2014	2	hoho	3	Function
Delete	Change	16	1005	ben	23	11/13/2014	2	test	3	Lunch and Dinner
Delete	Change	29	1005	lala	23	12/13/2014	3		3	Lunch and Dinner
Delete	Change	30	1005	holah	3	1/2/2015	4		1	
Delete	Change	32	1005	lolo	4	12/25/2014	1		1	

Figure 4. 9: Manage reservation page

4.4.5 Manage Staff Information

This function is for the manager to manage the staff information so that the staff will have an authority to access the system. The manager can add, delete, and update all the staff information. The "Add" function is for the staff to add new staff information into the system so that the staff can login to the system and manage the reservation. Once the staff is registered into the system all their activity in the system will be record for the reference. The "DELETE" function is for the manager to delete the staff information once they are no longer working at the restaurant as they are no longer using the system. This also will clear some of the memory space. Beside that manager also can make a change to the staff information system in case there is a change in their information such as the telephone number, email address and home address.

Figure 4. 10: Staff registration page

4.4.6 Manage reservation

This function is for the staff to manage the reservation list in the database. They enable to update, delete and insert the new reservation for the walk in reservation. Figure 4.12 shows the manage reservation page for staff. This page also have a searching function where user are able to search the data by fill in the form and click the search button.

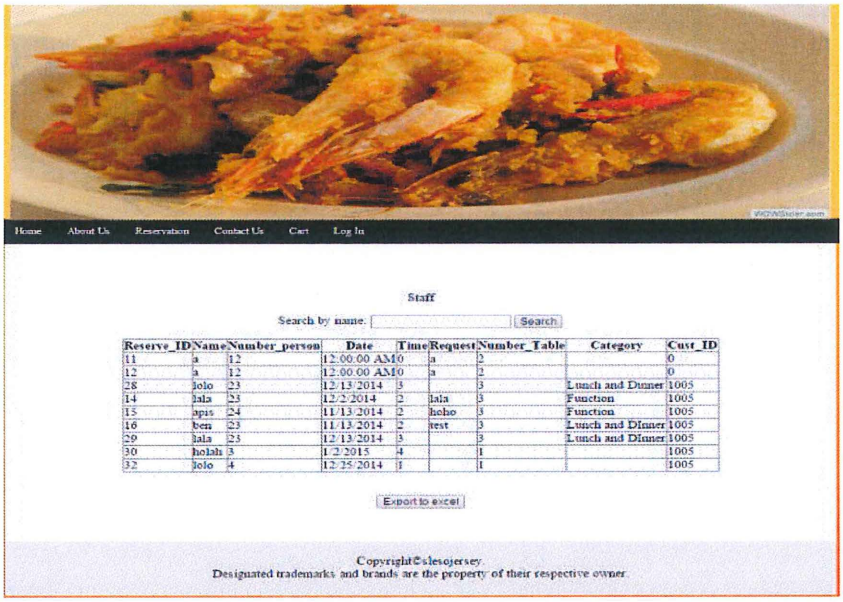


Figure 4. 11: Reservation list page

4.5. Conclusion

This chapter has been discussed about the implementation phase about Restaurant Online Reservation System. The implementation would be on designing database, interface and generating source code of system. The most important part is database design, because data will be stored in database and it is connected to each other. Therefore, database need to be accurate to make sure it will work smoothly and efficiency.

CHAPTER 5

RESULT AND DISCUSSION

5.1. Introduction

In this chapter, all the output for the system will be discussed. The outcome, assumption and further research about this system also discussed in this chapter. Hopefully, the discussion can bring ideas and more benefits to the future developer in order to upgrade and enhance the performance and functionality of the system in future.

5.2. Result Analysis

The propose system, Restaurant Online Reservation System has met all the objective of this project which are:

i. To computerize the reservation system from manual to computer system.
ii. To develop an online reservation system for customer convenient.
iii. To easily manage customer information for the reservation.

5.3. Result Of The System

Restaurant Online Reservation System is a web based application. This system is generally for the restaurant to help them manage the customer reservation. For implementation this system is developed by using a Prototyping Model methodology based on its practical and realistic method. System testing was done by module to sub modules. Results of the testing are as follows.

5.3.1 Register Account

The customer to be able to make the online reservation they need to an account in the system by fill in the information need requested by the form. The information then will be save in the database "Horizon" under the table "CustomerInfo", refer Figure 5.1 and Figure 5.2. The customer will be given the unique Cust_ID in the table.

Figure 5. 1: Registration form

Cust_ID	Name	Phone_number	Email	Password	Question	Answer
1	NULL	123456	as	123	123	123
1002	amen	123	as	123	Who is your fav...	spiderman
1003	am	124	as	123	What is your fa...	crystal blue
1004	amen	123	as@mail	123	What is your fa...	we
1005	amen	123456	amen.alias@g...	123	Who is your fav...	superman
1006	a	a	a	a	What is your fa...	a
1007	amen	0148313732	amen.alias@g...	12345	What is your fa...	nasi lemak
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 5. 2: CustomerInfo table

5.3.2. Login

Once the user customer create their account they are able to login into the system and they can proceed with the customer reservation. After customer insert the username and password and click login button the system will match the data with the one in the database and is the data match the system will fetch the name of the user and store in the session named as "User" and their name will stay until the they logout from the system.

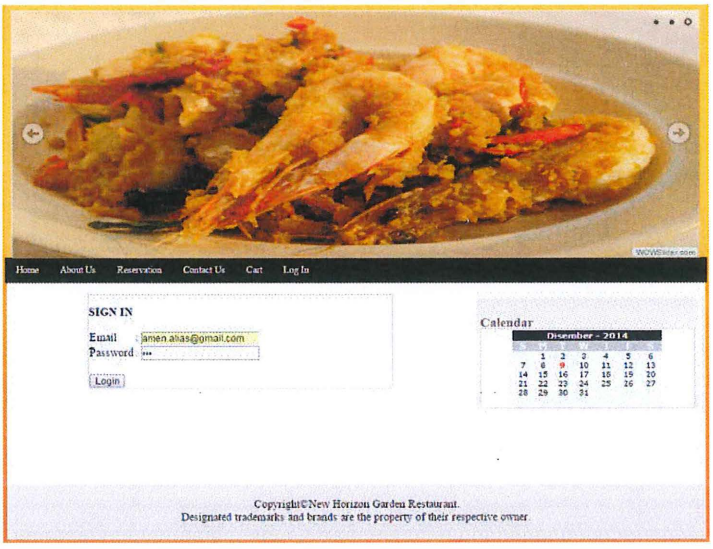


Figure 5. 3: Login form

5.3.3. Check Availability

This function is for staff and customer to check the available table for reservation for a certain date and time. User will select a reservation date and time and the system will show the number of table available for that particular date and time, refer figure 5.4.

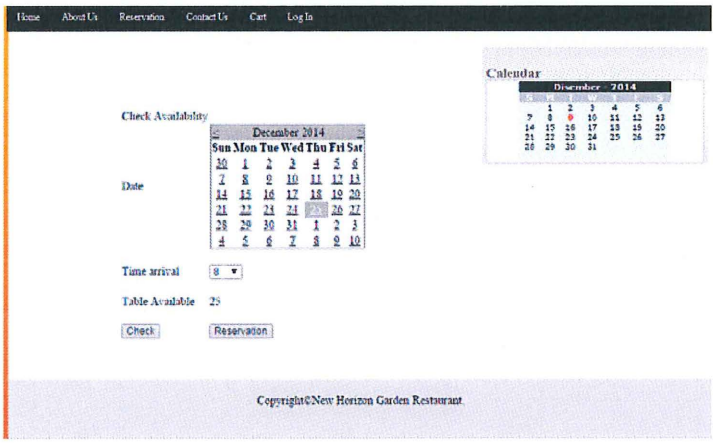


Figure 5. 4: Check table availability

5.3.4 Reservation

The reservation form will request the information from the user to make the reservation, refer Figure 5.5. After the user finish fill in the form and click the reserve button their reservation information will be save in the "Horizon" database under the table named "Reservation", refer figure 5.6. This information can be display by staff and the customer to manage the reservation information.

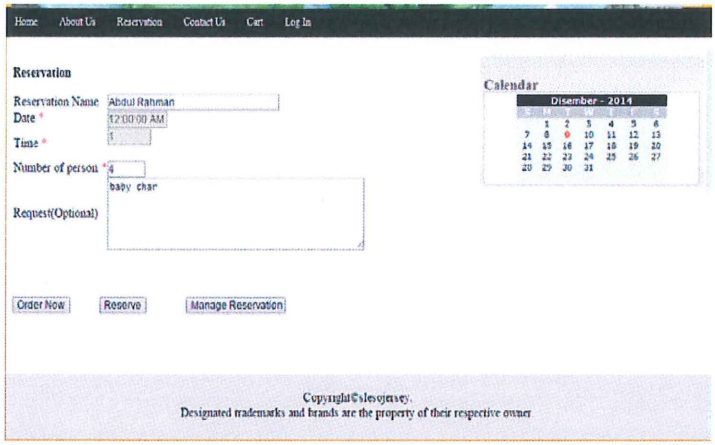


Figure 5. 5: Reservation form

Reserve_ID	Name	Number_person	Date	Time	Request	Number_Table	Category	Cust_ID
11	a	12	12:00:00 AM	0	a	2		0
12	a	12	12:00:00 AM	0	a	2		0
28	lolo	23	12/13/2014	3		3	Lunch and Din...	1005
14	lala	23	12/2/2014	2	lala	3	Function	1005
15	apis	24	11/13/2014	2	hoho	3	Function	1005
16	ben	23	11/13/2014	2	test	3	Lunch and Din...	1005
29	lala	23	12/13/2014	3		3	Lunch and Din...	1005
30	holah	3	1/2/2015	4		1		1005
32	lolo	4	12/25/2014	1		1		1005
33	Abdul Rahman	4	12:00:00 AM	1	baby char	1		1007

Figure 5. 6: Reservation table

5.4. Advantage And Disadvantage

The advantages of this system are;

i. Customer can manage their own reservation.

ii. User convenient.

iii. It help staffs to manage the reservation information.

iv. Cost efficient.

The disadvantages of this system are;

i. Internet connection problem. Since this system is a web based application it really depends on the internet connection. If facing with poor internet connection then the system cannot be access.

5.5 Future Work

Although this system is met the requirement, objective and scope successfully, the system still have the limitation. There are some constraints of the system.

i. There is no notification when the new reservations come in.

ii. The user cannot choose the table.

iii.

5.6 Conclusion

This chapter has been discussed about the output and detail about the outcome, assumption and future research about this system. Basically this system has meet all the objective which are (1) to computerize the reservation system from manual to computer system. (2) to develop an online reservation system for customer convenient. (3) to manage the record of customer information for reservation. User acceptance testing has been conducted by the developer to ensure the system is free from the bugs and error. The advantage of this system are cost effectiveness because it replace the manual reservation system and also increase the efficiency and accuracy and avoid any misunderstanding between the restaurant and the customer. The constraint also has been discussed in this chapter to enhance the functionality of the system.

REFERENCES

[1] Open Table
http://www.opentable.com/malaysia-restaurants [Retrieve on 24th October 2013]

[2] Saloma Theatre Restaurant
http://www.saloma.com.my/reservation.cfm [Retrieve on 24th October 2013]

[3]DJU Catering
http://www.djucatering.com/borangtempahan.php [Retrieve on 17th October 2013]

[4] Software Requirement Specification
http://techwhirl.com/writing-software-requirements-specifications/ [Retrieve on 25th October 2013]

[5] How to write SRS
http://www.microtoolsinc.com/Howsrs.php [Retrieve on 25th October 2103]

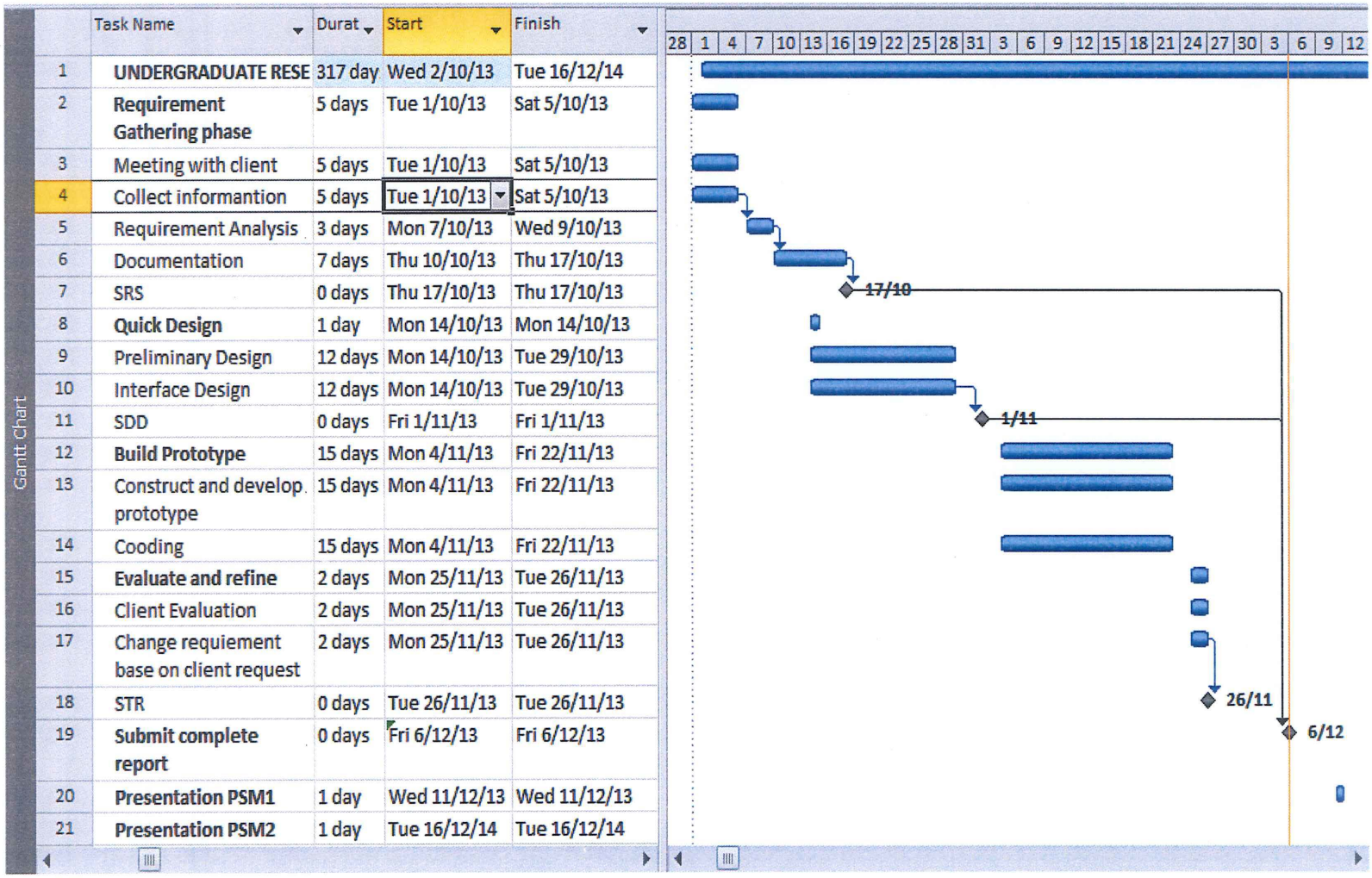
[6] Software Design Document
http://www.toptal.com/freelance/why-design-documents-matter [Retrieve on 1st November 2013]

[7] Class Diagram
http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/ [Retrieve on 1st November 2013]

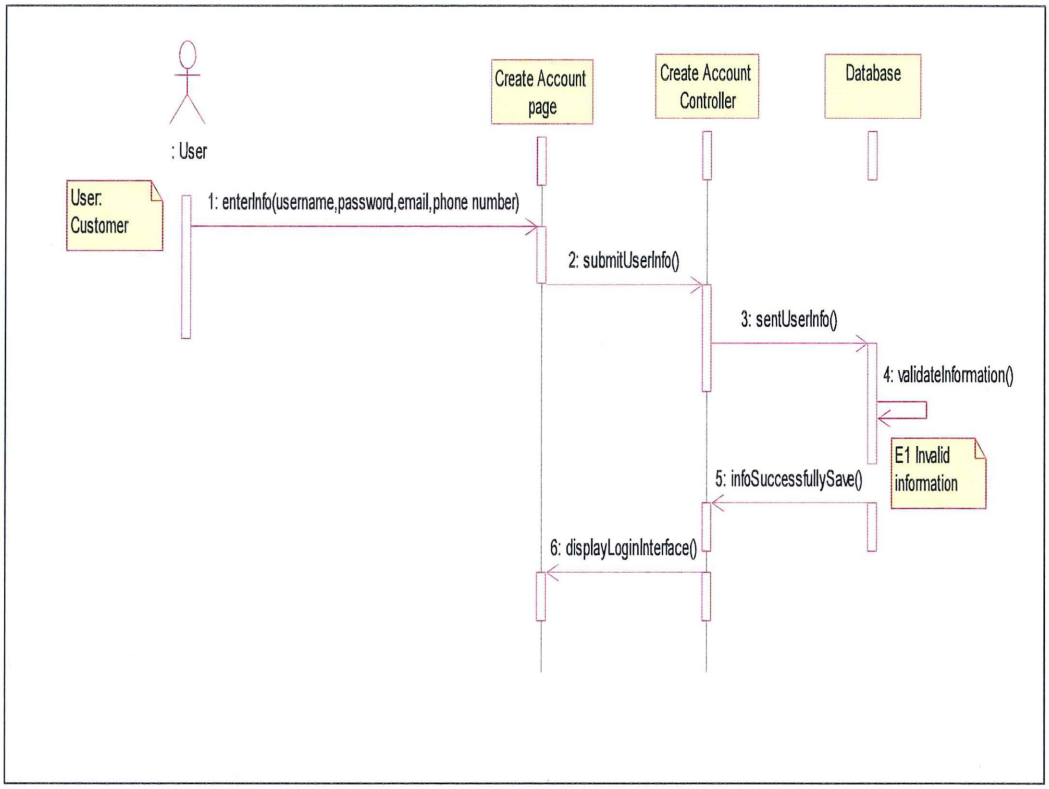
[8] Model View Controller
http://msdn.microsoft.com/en-us/library/ff649643.aspx [Retrieve on 15th November 2013]

APPENDIX A

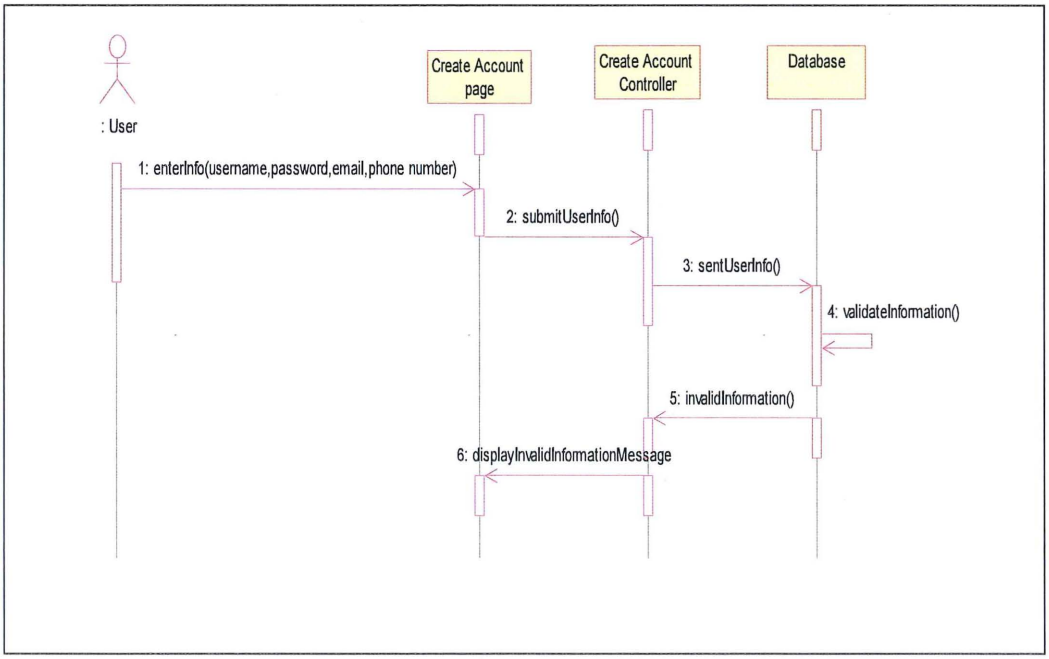
GANNT CHART


Gantt Chart
Task Name
Durat
Start
Finish
1 UNDERGRADUATE RESE 317 day Wed 2/10/13 Tue 16/12/14
2 Requirement Gathering phase 5 days Tue 1/10/13 Sat 5/10/13
3 Meeting with client 5 days Tue 1/10/13 Sat 5/10/13
4 Collect informantion 5 days Tue 1/10/13 Sat 5/10/13
5 Requirement Analysis 3 days Mon 7/10/13 Wed 9/10/13
6 Documentation 7 days Thu 10/10/13 Thu 17/10/13
7 SRS 0 days Thu 17/10/13 Thu 17/10/13
8 Quick Design 1 day Mon 14/10/13 Mon 14/10/13
9 Preliminary Design 12 days Mon 14/10/13 Tue 29/10/13
10 Interface Design 12 days Mon 14/10/13 Tue 29/10/13
11 SDD 0 days Fri 1/11/13 Fri 1/11/13
12 Build Prototype 15 days Mon 4/11/13 Fri 22/11/13
13 Construct and develop prototype 15 days Mon 4/11/13 Fri 22/11/13
14 Cooding 15 days Mon 4/11/13 Fri 22/11/13
15 Evaluate and refine 2 days Mon 25/11/13 Tue 26/11/13
16 Client Evaluation 2 days Mon 25/11/13 Tue 26/11/13
17 Change requiement base on client request 2 days Mon 25/11/13 Tue 26/11/13
18 STR 0 days Tue 26/11/13 Tue 26/11/13
19 Submit complete report 0 days Fri 6/12/13 Fri 6/12/13
20 Presentation PSM1 1 day Wed 11/12/13 Wed 11/12/13
21 Presentation PSM2 1 day Tue 16/12/14 Tue 16/12/14
28 1 4 7 10 13 16 19 22 25 28 31 3 6 9 12 15 18 21 24 27 30 3 6 9 12
17/10
1/11
26/11
6/12

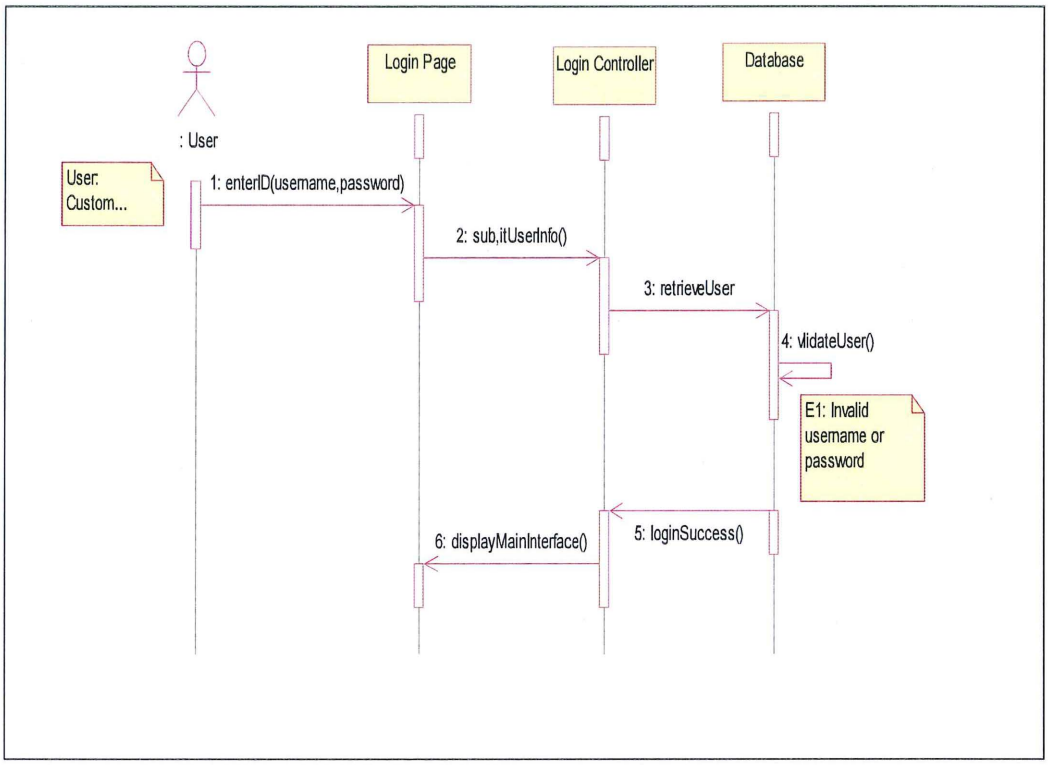
APPENDIX B
SEQUENCE DIAGRAM



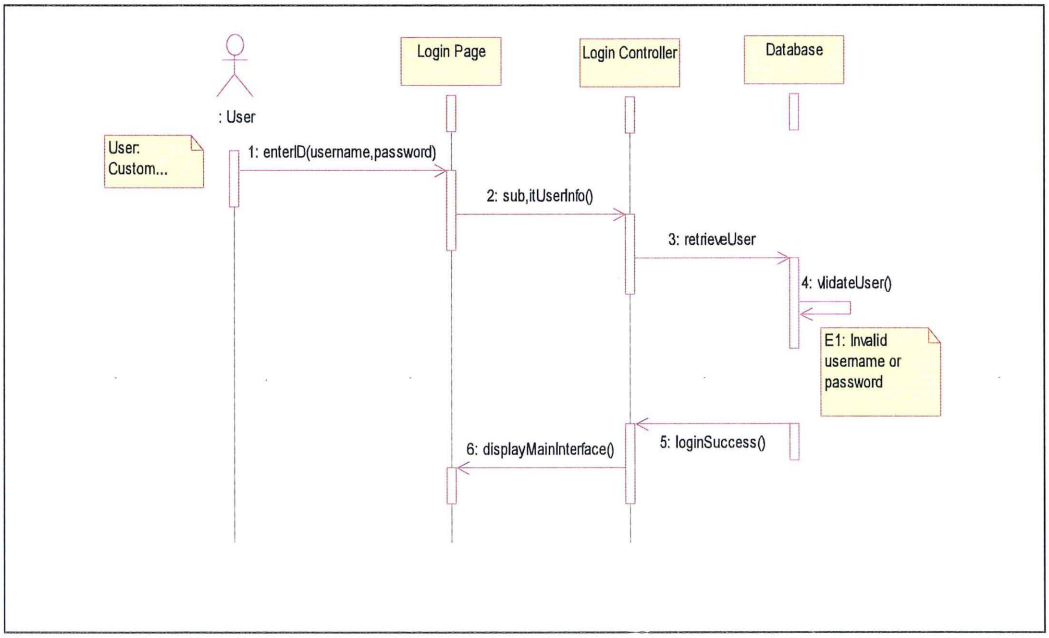
Appendix B-1.1: basic Flow- create account sequence diagram



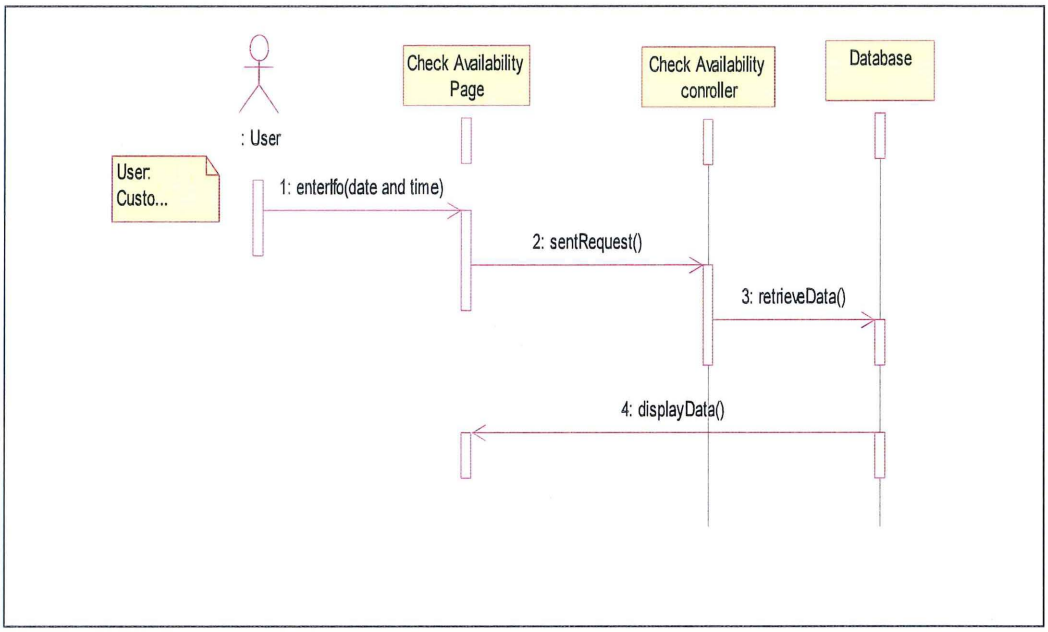
Appendix B-1.2: E1- invalid information sequence diagram



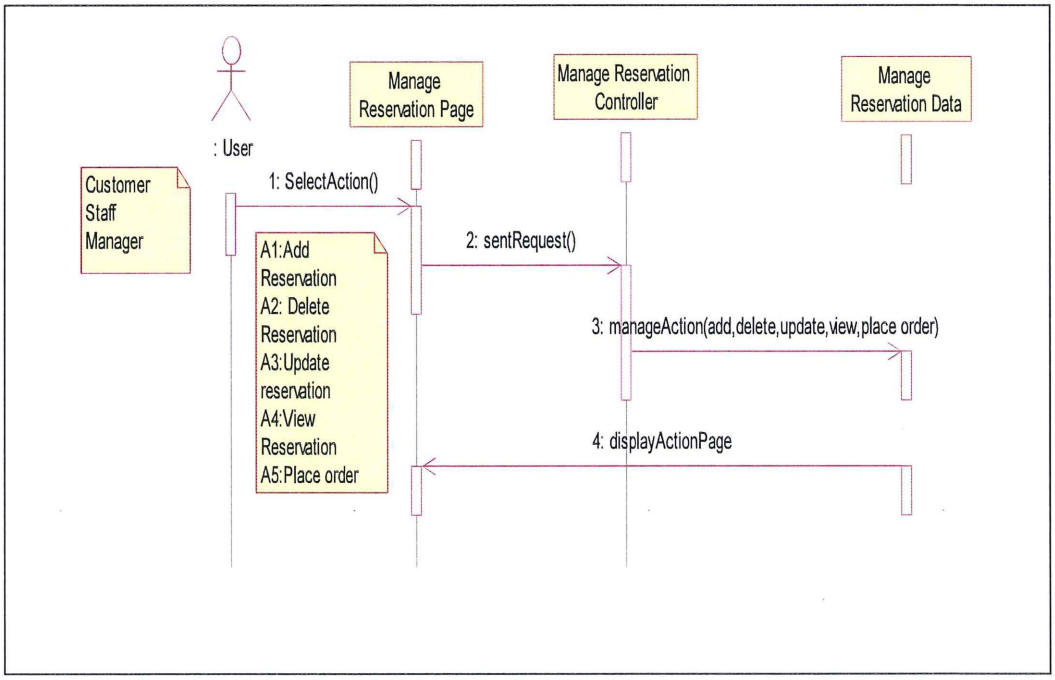
Appendix B-2.1: Basic Flow- login sequence diagram



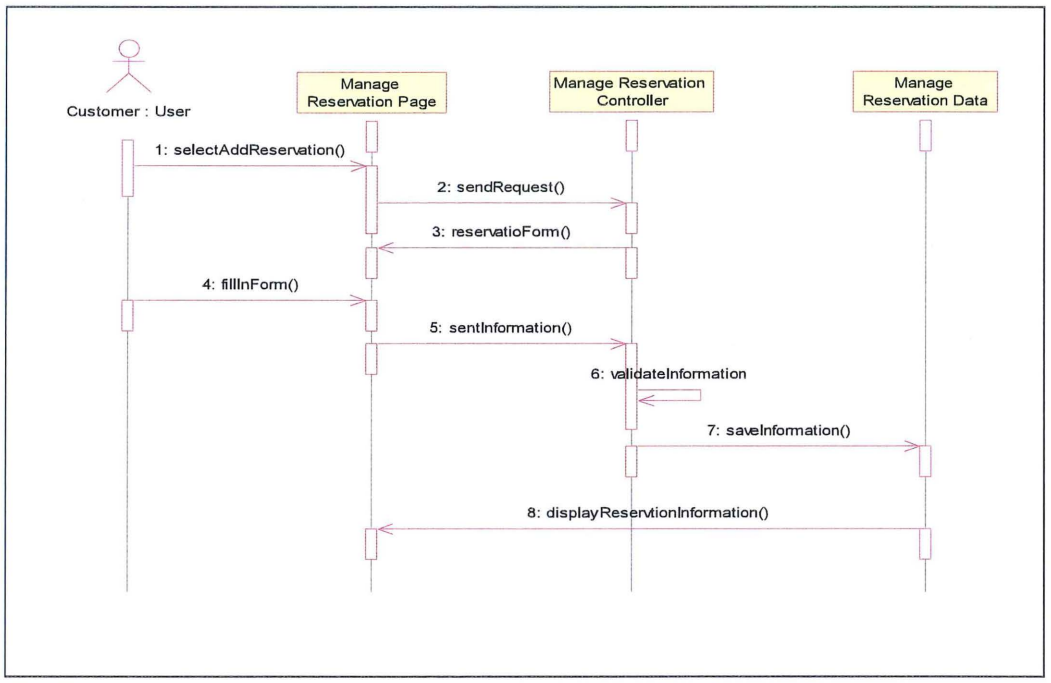
Appendix B-2.2: E1-invalid username or password sequence diagram



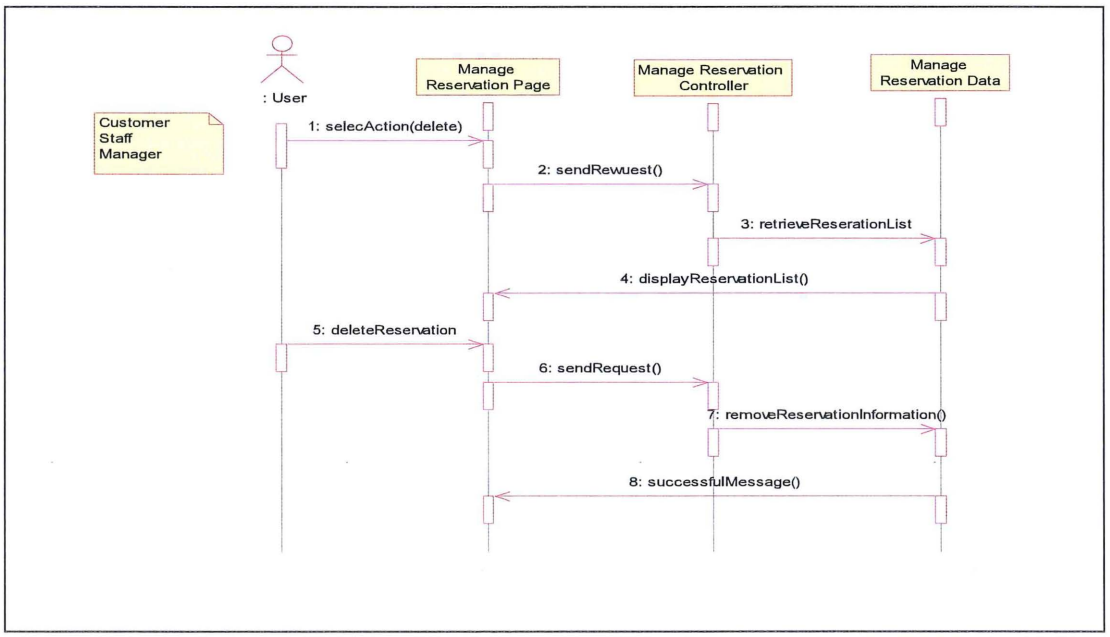
Appendix B-3.1: Basic Flow- Check Availability sequence diagram



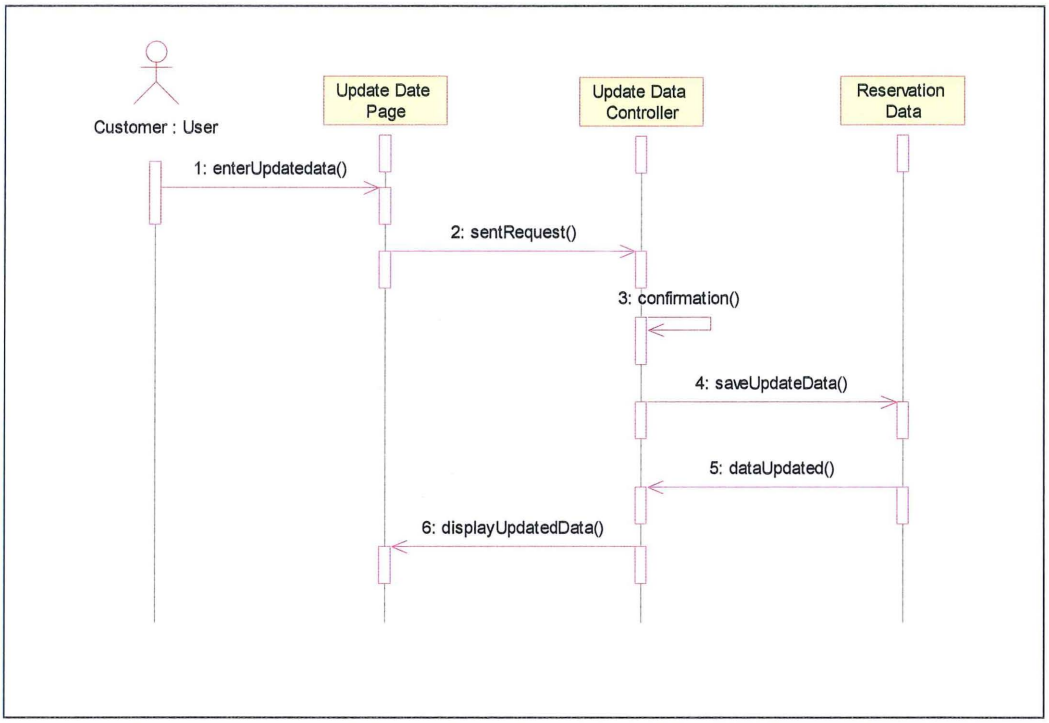
Appendix B-4.1: Basic Flow-Manage Reservation sequence diagra



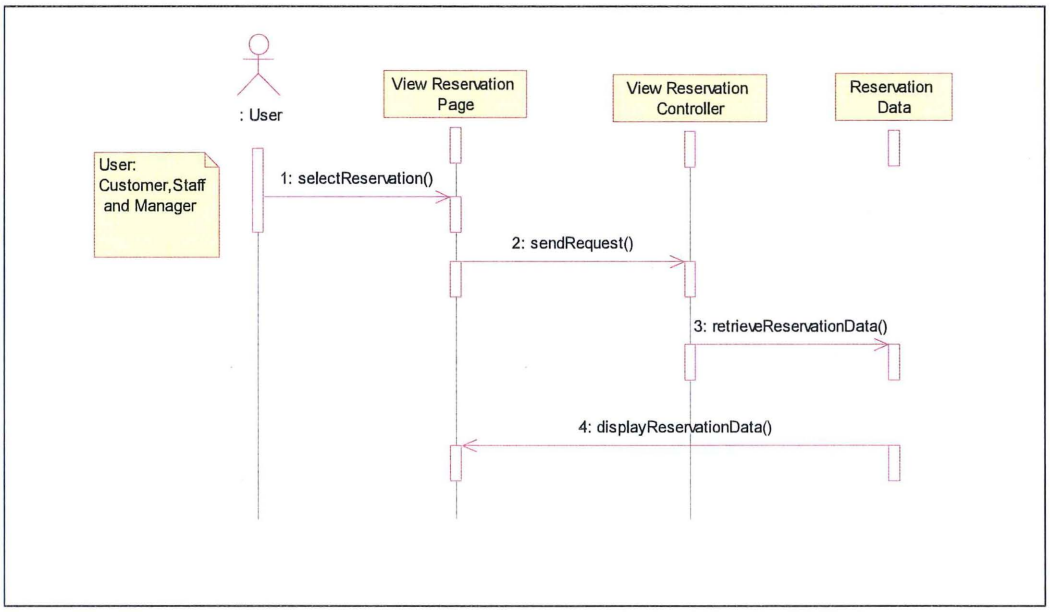
Appendix B-4.2: Alternative Flow 1-Add reservation sequence diagram



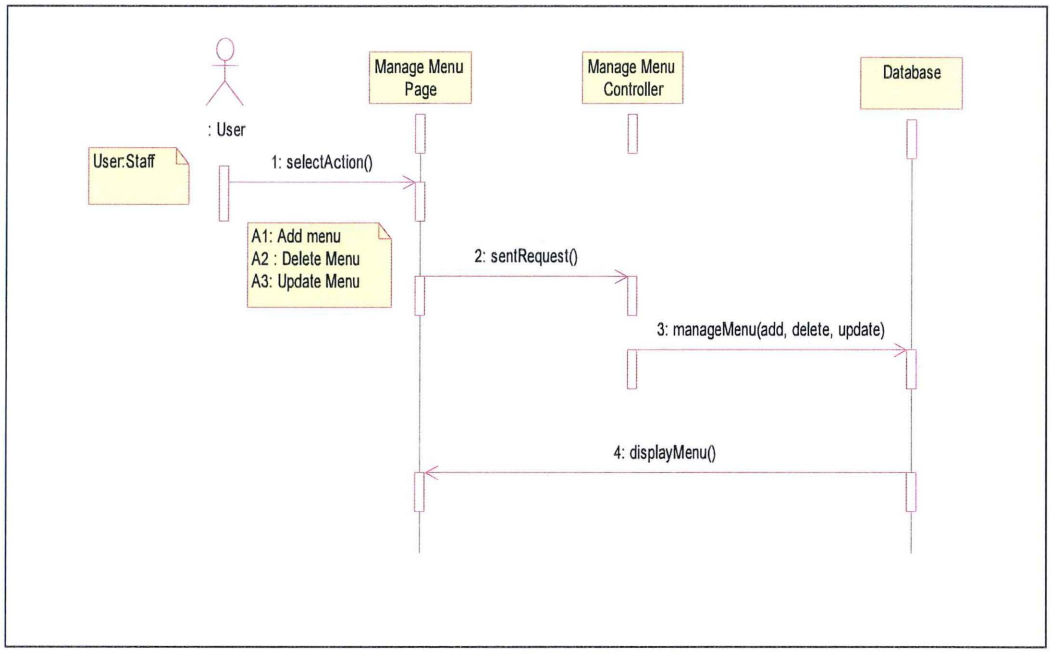
Appendix B-4.2: Alternative Flow 2- Delete Reservation sequence diagram



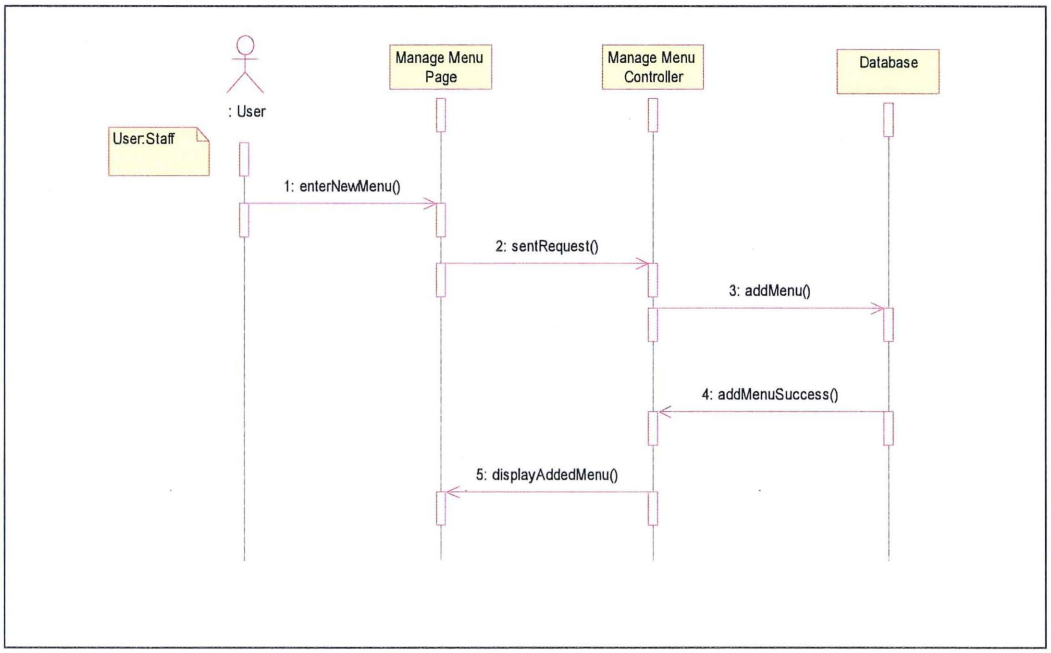
Appendix B-4.2: Alternative Flow 3-Update Reservation Information Sequence Diagram



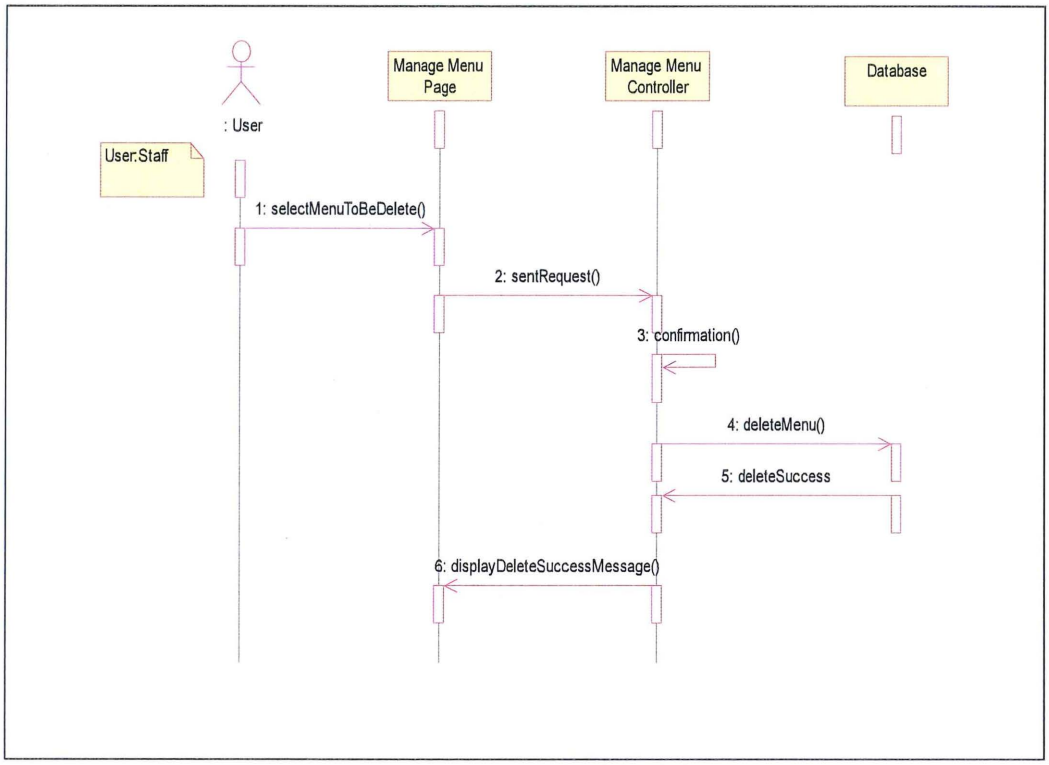
Appendix B-4.2: Alternative Flow 4- View Reservation sequence diagram



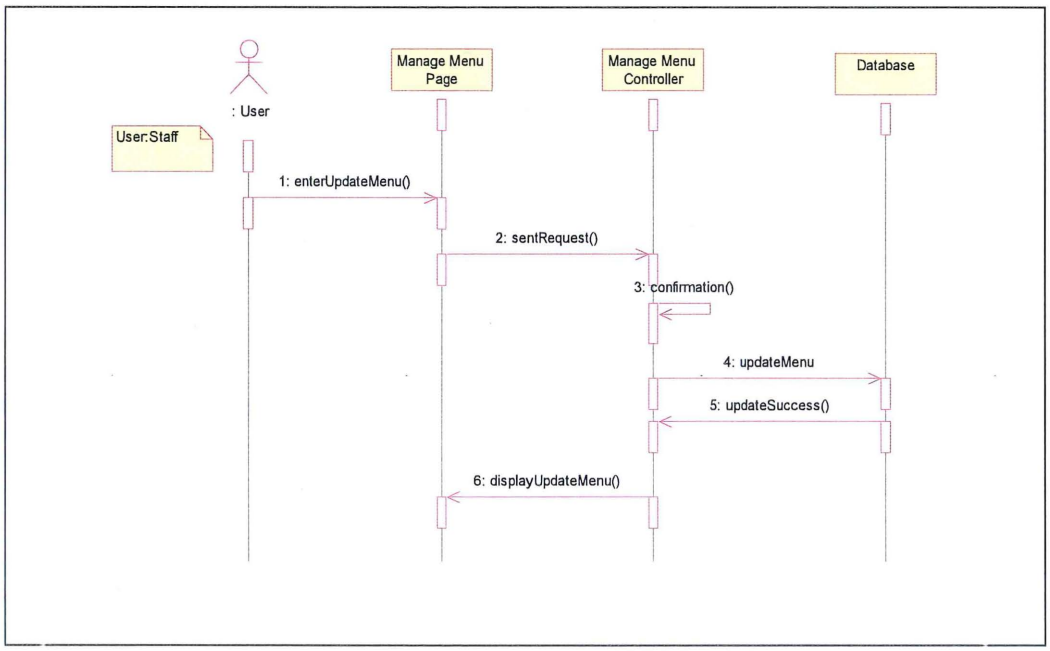
Appendix B-5.1: Basic Flow- Manage Menu sequence diagram



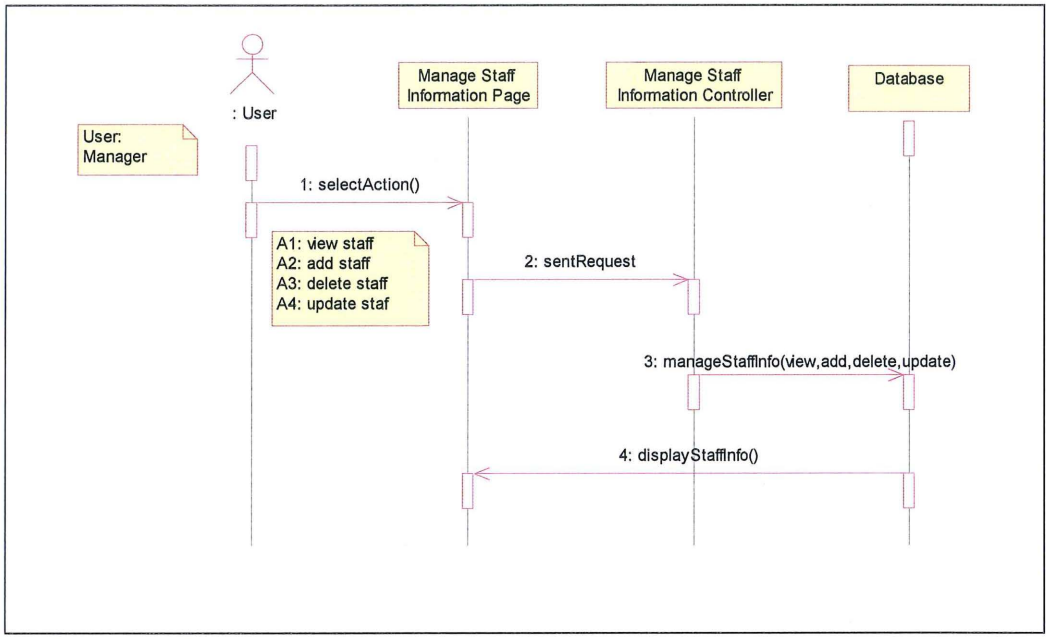
Appendix B-5.2: A1-Add Menu Sequence diagram



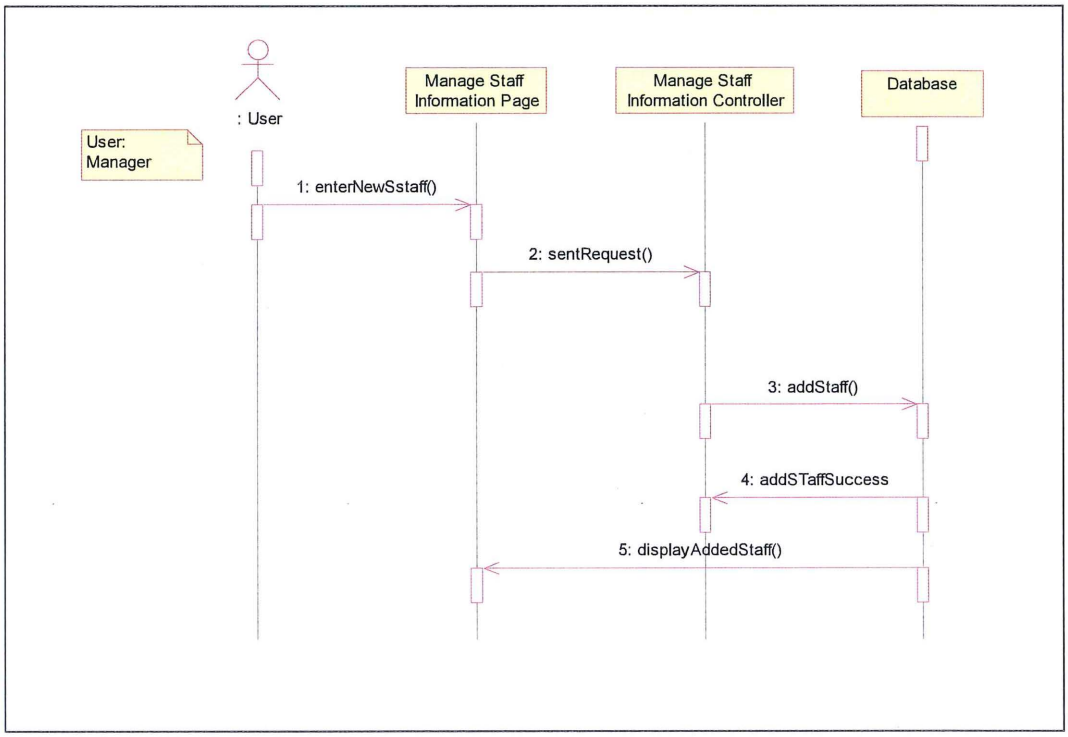
Appendix B-5.2: A2- Delete Menu sequence diagram



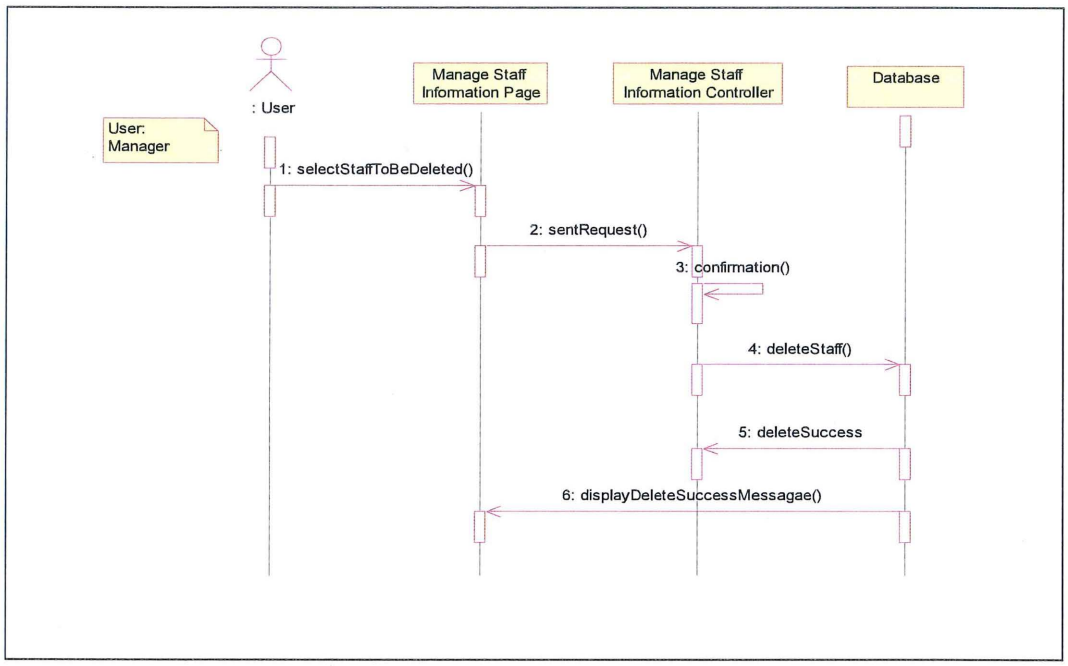
Appendix B-5.2: A3-Update Menu sequence diagram



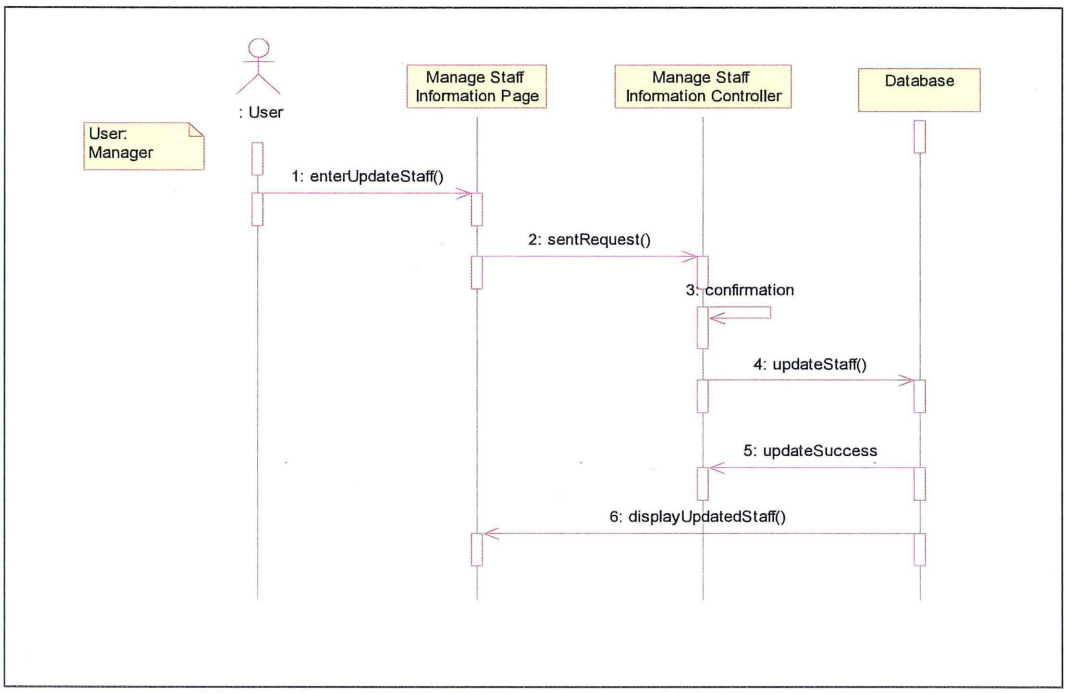
Appendix B-6.1: Basic Flow- Manage Staff Information sequence diagram



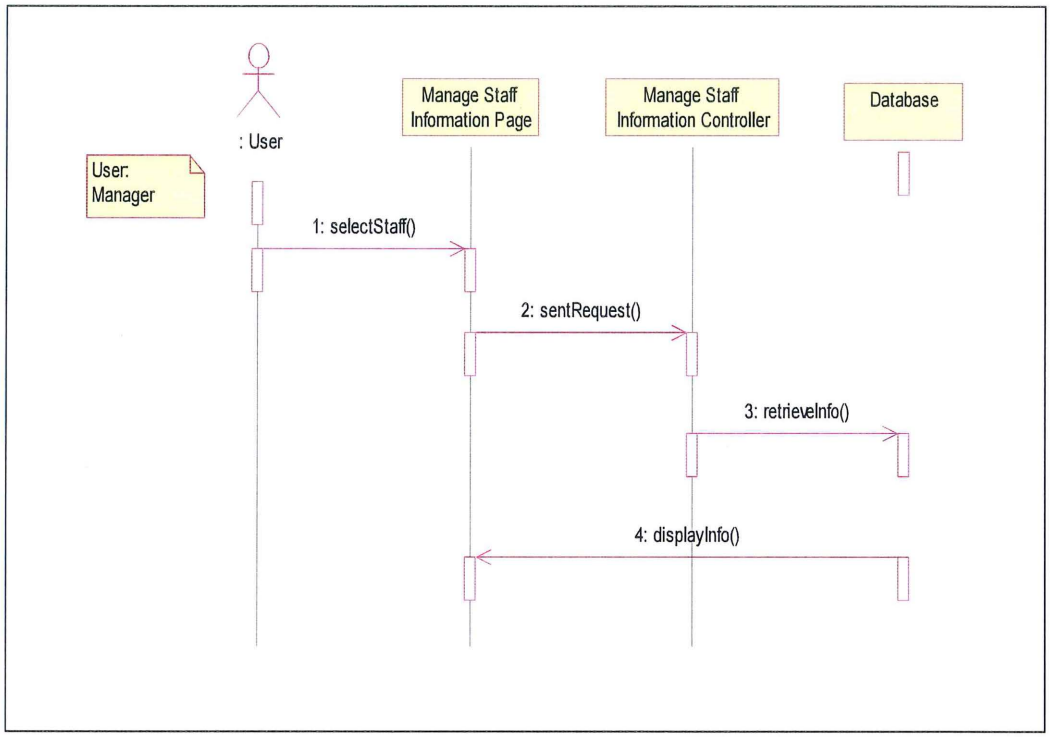
Appendix B-6.2: A1-Add Staff sequence diagram



Appendix B-6.2: A2-Delete Staff sequence diagram



Appendix B-6.2: A3-Update Staff sequence diagram



Appendix B-6.2: A4-View Staff sequence diagram