Catering Service & Reception Hall Management System

For

Mayadunne Catering Service.

A.P.N.K. Dharmarathne.

BIT Registration Number: R141836

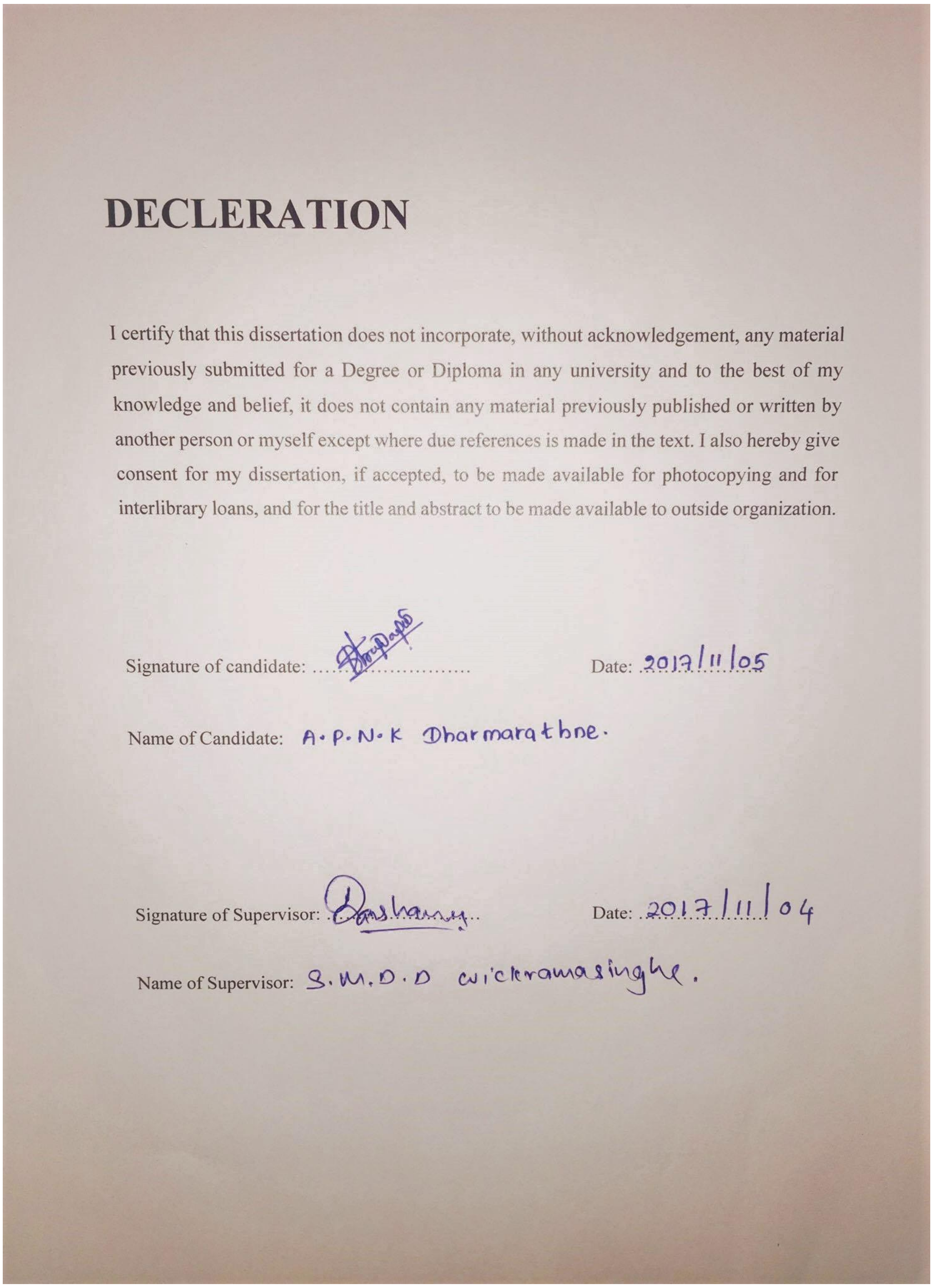
Index Number: 1418361

Name of the Supervisor: W.D. Dinushal Wickramasinghe

December 2017

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

DECLARATION



DECLERATION

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due references is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organization.

Signature of candidate: [illegible] Date: 2017/11/05

Name of Candidate: A.P.N.K Dharmarathne.

Signature of Supervisor: [illegible] Date: 2017/11/04

Name of Supervisor: S.M.D.D Wickramasinghe.

ABSTRACT

Mayadunne Catering Service is one of the famous catering service in Mabima area. Mayadunne Catering Service is providing services such as catering for any kind of functions and maintain two reception halls called Mayadunne Reception hall and Mayadunne banquet hall. There are about 30 employees are working in Mayadunne Catering Service. Currently they are handling their day to day functions by using of a manual system. Due to this manual system, currently this company is unable to provide a better service for their clients and can’t manage day to day functions properly.

As solutions and weaknesses in the manual system is considered to develop an automated standalone Catering Service & Reception Hall Management System. In this system, it is supposed to have Menu Management and Reception Hall Management primarily. In addition this proposed system manage functional supportive services such as photography, videography and decorations. Employee, User Accounts, Kitchen item Management are covered along with the above primary functions. Report generating and notification will manage to support for a highly valued system.

This system was developed using the object oriented concepts, Java language as the programming language on Netbeans as the IDE, Hibernate framework and MySQL database on Windows operating system. Jasper Reports were used to generate reports. In addition, Scene Builder was chosen as the interface designing tool and Visual Paradigm was used to draw the UML diagrams shown in this Report.

At the end of this development process, the stand alone system which was tested by many users of the company, was successfully handed over to the client to enhance their company business process as well as to help the company to become a leading catering service in Sri Lanka.

ACKNOWLEDGMENT

There are many honorable persons that I wish to express my gratitude and take this opportunity to remind some of them specially.

- First and foremost I owe my deep gratitude to the University Of Colombo School Of Computing for offering us this precious degree program and all its staff who guided me from the beginning.

- Another special word of thanks and gratitude goes out to my learned Supervisor Mr. S.M. Darshana Dinushal Wickramasinghe for his able guidance and support rendered to me throughout the completion of my Project and dissertation writing.

- I will be failing in my duty if I do not avail of this opportunity of expressing a word of gratitude to Mr. R.D.D Suranga, Managing Director and Mr. Susith Sanasuma, Administrators of Earth University College who extended their unstinted support and encouraged me to launch on this Project successfully.

- I am also much obliged to Mr. S.M.M. Sarath Keerthi, Proprietor of Mayadunne Catering Service who enlightened me with his words of wisdom relating to the process, functionalities and other necessary information of the Organization and Co-operated immensely in the course of the development of the project.

- I also acknowledge with a deep sense of reverence, my gratitude towards my parents and member of family, who has always supported me morally as well as economically.

- At last not least gratitude goes to all of my friends who directly or indirectly helped me to complete this project.

- Any omission in this brief acknowledgement does not mean lack of gratitude.

TABLE OF CONTENT

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CSS	–	Cascading Style Sheets
DAO	–	Data Access Objects
GUI	–	Graphical User Interface
JDBC	–	Java Database Connectivity
GUI	–	Graphical User Interface
JDBC	–	Java Database Connectivity
JRE	–	Java Runtime Environment
OOAD	–	Object Oriented Analysis and Designing
OOP	–	Object Oriented Programming
OS	–	Operating System
POJO	–	Plain Old Java Objects
RAD	–	Rapid Application Development
RDBMS	–	Relational Database Management System
SQL	–	Structured Query Language
UML	–	Unified Modeling Language
HQL	–	Hibernate Query Language
ORM	–	Object Relational Mapping
OMG	–	Object Managing Group
WWW	–	World Wide Web

CHAPTER 01 – INTRODUCTION

1.1 INTRODUCTION

Mayadunne Catering Service is one of the famous business organization that provide foods and reception hall facilities in Mabima area. It is located at 394/1/A, Mabima, Heiyanthuduwa. Mayadunne Catering Service is providing services such as delicious and quality meals for any kind of functions and providing reception hall facilities for various reception needs.

Mayadunne Catering Service is started as a small catering service in about 15 years ago with small group of employee. At present the Organization has achieved large customer loyalty attaining Mabima area and also they maintain two reception halls. They have been able to provide excellent services to the customers who come to them.So Mayadunne Catering Service's vision is to be a recognized as the top caterer and serves quality meals and luxury reception hall facility. Mayadunne Catering Service management welcomes new ideas to keep their path in a successful way.

Currently they are handling their day to day functions by using of a manual system. They need to expand their organization and now they need to increase the quality of their services by adding information technology to their procedures and overcome the drawback of their manual system.

1.2 MOTIVATION FOR THE PROJECT

Currently the Mayadunne Catering Service is using a manual system to handle catering and reception hall processes. Although the hotel is a standardized quality catering service and reception hall it is not realizing its maximum potential due to delay of the activities of the current redundant manual system.

When a guest makes an order, all the order details are recorded in files. Calculation of bills and order management are done manually too. Due to the difficulty in location of guests files because its need a large storage space, it is extremely difficult to staff to manage them efficiently. Also difficulty in data analysis, human errors, poor communication between departments and unauthorized modification of data due to low security levels are main problems that are identified in the existing system. Because of that it is difficult to handle day to day functions of the hotel manually.

The proposed system will help to staff members to steer the hotel functionality and transactions. Also, it will realize its maximum potential in addition to its competence, of the hotel business industry.

1.3 OBJECTIVES AND SCOPE OF PROPOSED SYSTEM

1.3.1 OBJECTIVES

- Reduce human effort

  Order acceptance, payment process, menu selection process, reception hall booking process are automated by system.
- Provide user friendly graphical user interface and reduce the human errors.
- Improve the productivity of system users.
- Improve data accuracy
- Improve efficiency and simplicity.
- Reduce the extensive paperwork & Improve reporting process

  By reducing paper works and spaces which need to store large files, this system improves efficiency of the organization.
- Enable automated data entry methods

  Instead of error prone human effort, the system will ease the, deleting, and updating information related to the hotel.
- Improve security levels

  This system prevents unauthorized access of data by giving necessary privileges to suitable users.
- Improve the quality of organization and elaborate company reputation.

1.3.2 SCOPE

- Catering management

  The proposed system can manage change menu items according to the client's expectations and keep track of price lists of different kind of dishes available in the menu forms.
- Reception hall booking management

  The system will handle booking details according to customer's requirements.
- Service management

  The system handles services available for the function booking process, kitchen section and other services.
- Manage staff and guests details

  This system is able to insert, update or delete staff details efficiently.
- Manage Customer details

  This system is able to insert, update or delete Customer details efficiently.
- Manage user details

  The system keeps details about users of the system and also it can grant access privileges to users.
- Manage menu customization

  The system is able to customize the menus.
- Payment management

  The system is able to calculate the payments.
- Report generating and notifications

  The system will generate reports when necessary.

1.4 STRUCTURE OF THE DISSERTATION

This dissertation shows the works that carried out during the each phases of the project when creating the Catering and reception hall management system for Mayadunne Catering Service.

CHAPTER 02 – ANALYSIS

This chapter provides an analysis of the functional and non-functional requirements. It will also describe about fact gathering techniques and existing manual system.

CHAPTER 03 – DESIGN

This chapter will provide class diagrams, activity diagrams, Entity Relationship diagrams and other related diagrams regarding to the proposed system. System requirements, system architecture, sub systems, database structure and user interfaces are other topics describe in this chapter.

CHAPTER 04 – IMPLEMENTATION

This chapter will describe about all major codes, module structures, the implementation environment, development tools and platform used in the project.

CHAPTER 05 – EVALUATION

Evaluation chapter will provide details about testing methodologies used in the test cases and all the test cases in the project.

CHAPTER 06 – CONCLUSION

This chapter will conclude the dissertation with a critical evaluation of the system and suggestions for any future enhancements.

CHAPTER 02 – ANALYSIS

This chapter explains the functional and non-functional requirements of the problem domain and requirement gathering techniques.

Systems analysis is the methodical investigation of a problem and the identification and ranking of alternative solutions to the problem. Systems analysis is often called structured systems analysis when certain “structured” tools and techniques, such as Data Flow Diagrams, are used in conducting the analysis. This chapter will give an overview of the existing system, fact gathering techniques and functional and non-functional requirements of the hotel system.

2.1 CURRENT MANUAL SYSTEM

Staff at the Mayadunne Catering Service is recorded function bookings, kitchen, Customer details and employee details by manually in books and files. Bookings are done by only visiting the hotel. Clients’ personal details are recorded during booking process. Menu selection and paying advance booking payments also done with this booking process.

This process is so hard to handle in this way at the hotel with this manual process.

2.1.1 WEAKNESSES OF CURRENT MANUAL SYSTEM

- Difficult to find location of guest files.
- Need a large storage space to store files.
- Guest files can easily get lost or mix-up with other guest files and Documents
- Retrieval of guest details is difficult
- Data analysis is difficult.
- Data backups are not available.

2.2 USE CASE DIAGRAM FOR THE CURRENT MANUAL SYSTEM

The following figure 2.1 shows the major processes of current manual system. Client and front banquet officer play major roles who are the actors in the use case diagram.

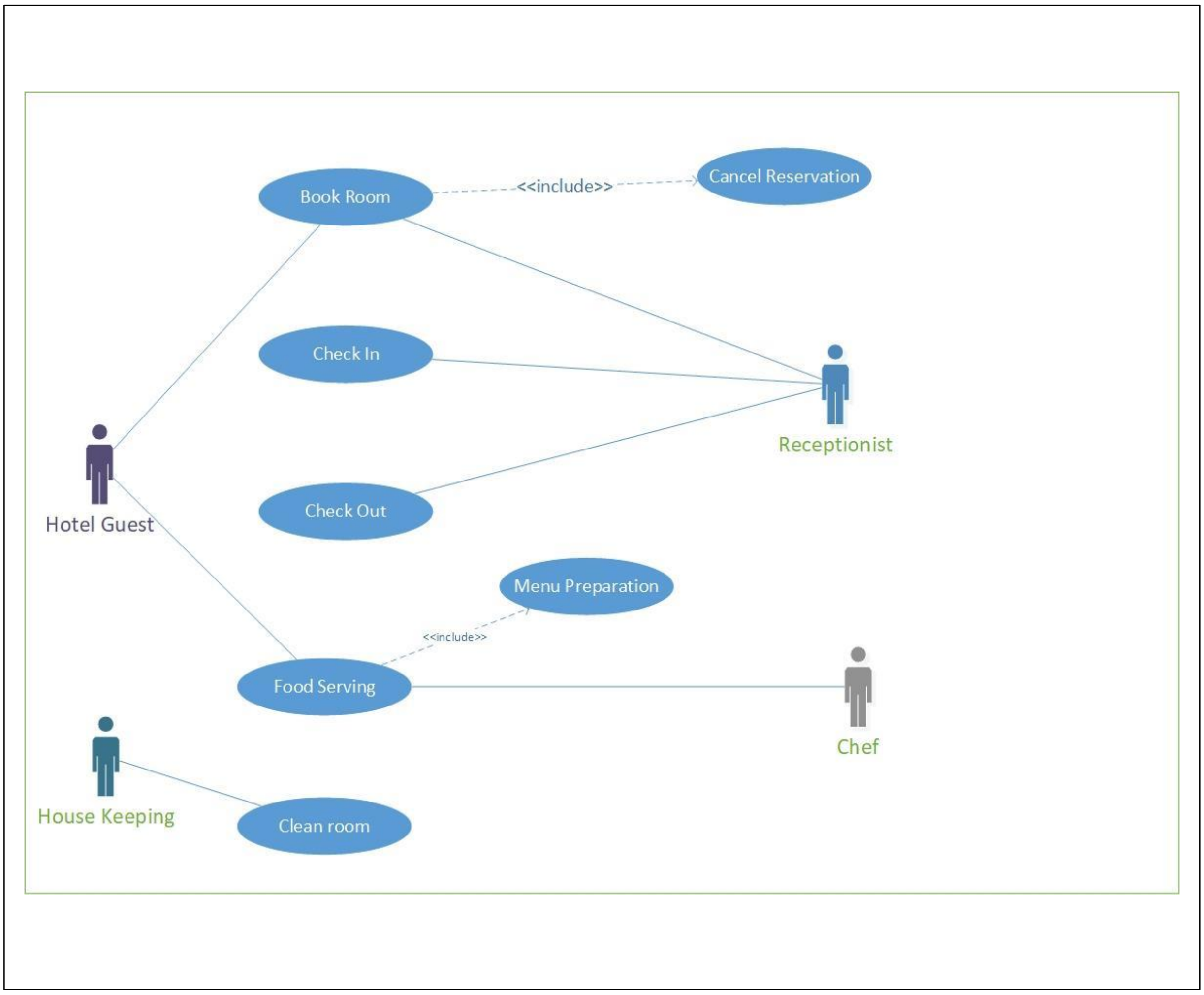


Figure 2. 1Use case diagram for existing system

2.2.1 FUNCTION BOOKING

This is a main process of hotel and a key requirement of the proposed automated system. The manual system does not have facility to keep function booking information properly. Currently information keeps in papers.

2.2.2 CHECK AVAILABLE DATES

Currently Reception Hall does not have facility to check whether what the current statuses of function halls are. There is a manual ledger to keep track of function hall statues.

2.2.3 CHECK-IN GUEST

When guest check-in to the hotel, the current manual system keeps a note of guest and booking. The manual system is unable to provide a check-in card to guest.

2.2.4 CHECK-OUT GUEST

When guest come to pay total bill for the booking there is no mechanism to calculate total bill properly. Because, for calculate total bill, banquet officer should find the file of the relevant booking in this manual system.

2.2.5 PREPARE REPORT

A detail report of the booking is the payment document of guest. The current manual system is unable to provide that kind of report of the reservation properly.

2.2.6 IDENTIFY BOOKING

Identify or search a booked functions is a main process of hotel and a key requirement of the proposed automated system. The manual system does not have a facility to uniquely identify a booking according to relevant guest.

2.3 OUTLINE OF EXISTING SIMILAR SYSTEMS

There are so many hotel management systems worldwide. Some of them are listed below.

2.3.1 ANAND SYSTEMS INC

For hotel industry, they have introduced easy-to-use **hotel Software**, known as ASI Front Desk. A perfect software solution for hospitality industry, this comprehensive software suite comes integrated with modules for many aspects of hotel management. Often referred as Property Management

System in the hospitality industry, this special class of software is ideally suited for use at hotels, military guest houses, motels, resorts, inns, lodges, hostel, suites, ranch, apartments, medical centers and bed and breakfast operations [13]

Features:-

- Easy to Use
- Reservation, Check-In and Check-Out
- Group Management
- Back Office Features
- House Keeping & Maintenance
- Tax
- Reports

Figure 2.2 shows the sample screenshot of the Anand Systems Inc.

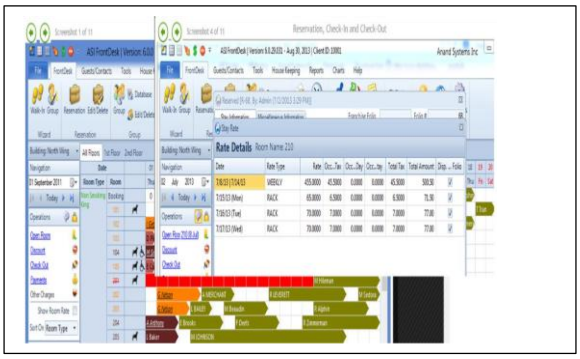


figure 2. 2 screenshot of the Anand Systems

2.3.2 BISTONESOFT

BISTONESOFT
Hotel Management Expert

If every room/bed of the hotel has a fixed rate, and the hotel offers a lot of services for their guests, they can download and use Bistone

Hotel Reservation System. With its help, clients can manage their rooms/beds easily, and avoid double bookings. [12]

Features:-

- Manage Reservations, Rooms/Beds, Services
- Highly Configurable and Easy To Use
- Affordable - A One Time Price - No Hidden Costs
- Track Customers and avoid Double Bookings
- Analyze Reservation Data
- Automatic Calculations
- Provide Valuable Reports

Figure 2.2 shows the sample screenshot of Bistonesoft.

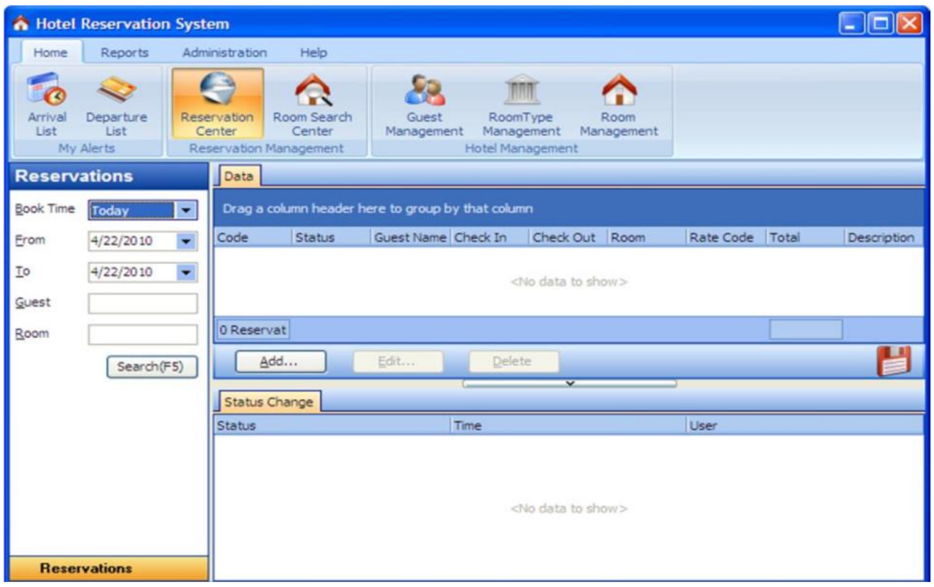


Figure 2. 2 Screenshot of Bistonesoft.

2.3.3 EZEE

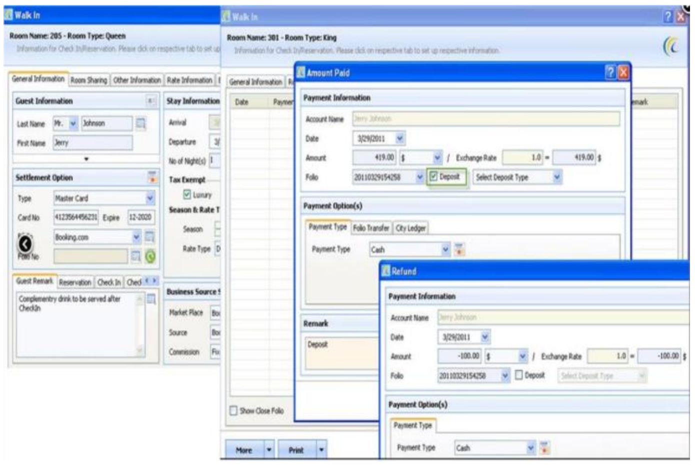
eZee Front Desk, Hotel Software is the modern solution which has whole range of integrated modules to cover every aspect of property management. The generalized version of eZee front desk hotel reservation software was accepted in worldwide with due to its state-of-art technology and extremely easy to use in nature. The major modules of the latest eZee Front-

Desk constitutes of easy check-in / check-out, 2 click guest reservation [12].

Features:-

- Day Use Feature
- Multi-currency settlement & print folio and bills
- Reservation & Group Booking Management
- Back Office Module
- Laundry Management
- Housekeeping Module Management

Figure 2.3 shows the sample screenshot of Ezee Solution



Room Name: 205 - Room Type: Queen
Information for Check In/Reservation. Please click on respective tab to set up

Guest Information
Last Name: Mr. Johnson
First Name: Jerry

Settlement Option
Type: Master Card
Card No: 4123564456231 Expire: 12-2020
Booking.com

Complementry drink to be served after CheckIn

Stay Information
No of Night(s): 1
Tax Exempt: ☑ Luxury

Room Name: 301 - Room Type: King
Information for Check In/Reservation. Please click on respective tab to set up respective information.

Amount Paid
Payment Information
Account Name: Jerry Johnson
Date: 3/29/2011
Amount: 419.00 $ / Exchange Rate 1.0 = 419.00 $
Folio: 20110329154258 ☑ Deposit Select Deposit Type
Payment Option(s)
Payment Type: Cash
Remark: Deposit

Refund
Payment Information
Account Name: Jerry Johnson
Date: 3/29/2011
Amount: -100.00 $ / Exchange Rate 1.0 = -100.00 $
Folio: 20110329154258 ☐ Deposit Select Deposit Type
Payment Option(s)
Payment Type: Cash

☐ Show Close Folio

Figure 2. 3 screenshot of Ezee Solution

2.4 FACT GATHERING TECHNIQUES

Combination of different fact finding techniques are used to capture requirements of the Connection manage domain. Following are techniques used to gather requirements in Mayadunne Catering Service.

- Observing company process.

To get an idea about the work load of the current manual file base system in the Water Board, direct observation was carried out. All critical functions were identified using this method.

- Interviews and Discussions.

Face-to-face contact with users through individual interviewing is the primary source of requirements and an important way you gather and validate their requirements. Remember that it is not the only possible technique, and that you can conduct interviews many different ways. Develop a repertoire of styles to fit different situations. Unless you use the system yourself, you will need to make an effort to understand and experience the user's problem to describe it clearly and correctly.

.

- Review company documents.

In the current manual file base system many documents are created and handled by the employees. This is the best method to gather information in a short time period. Many documents are handled by the manual system such as,

Consumer Registration Form

Paid estimated and Connection files

Billing Details files

These documents were carefully analyzing to gather requirements.

2.5 IDENTIFIED REQUIREMENTS FOR THE PROPOSED SYSTEM

2.5.1 FUNCTIONAL REQUIREMENTS

- Manage staff details

  The system can manage staff details properly. Each employee of staff has a unique record in system.

- Manage guest details

  The system records guest details properly and secure manner.

  When user needed the system can find any guest detail.

- Manage function booking details

  Function booking details are more important to the company.

  Each and every booking has a unique identity in the system.

- Manage menu and menu customization

  Menu and menu customization details are considered as master data in system. They can be managed properly.

- User login management

  The system can handle different user accounts. Each and every user has a user account that has different access privileges.

- Manage payment details.

  Booking payments and transactions are happening through cashier in the hotel. The system can manage payments properly.

- Manage kitchen Requests

  Kitchen management is important aspect as function booking. System provides Stock Management Module for that.

- Retrieve reports

  By defining different criteria, the system can retrieve management reports.

2.5.2 NON – FUNCTIONAL REQUIREMENTS

- Availability

  The system should be available at any time.

- Correctness

  The information provided by the system should be correct.

- Efficiency

  The system should use minimum resources to operate.

- Flexibility

  The system can be response to different situations when needed.

- Integrity

  Can be integrated and communicate with other systems properly

- Maintainability

  The system should be re-build for changing needs.

- Reliability

  The system can be managed correct information.

- Run in low configuration machines

  System should be run in low powered platforms with resources.

- Easiness

  The system should be easy to understand to users.

- Robustness

  System should be executed properly when there is an error.

2.6 METHODOLOGY FOR THE SELECTED SYSTEM

There are several methodologies one can use to develop a system. Some of them are describe below.

2.6.1 Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

2.6.3 AGILE MODEL

Ask people to define Agile development and it’s likely you will receive a range of definitions emphasizing different aspects of the process. To us, Agile is about collaborating to deliver the highest value product increment, with high quality, as quickly and as frequently as possible, and continuously improving the delivery process.

2.6.4 RAD MODEL

Rapid Application Development (RAD) is used as the methodology for the proposed system. Rapid application development is a model based on the concept that higher quality products can be developed faster through more expedient processes, such as early prototyping, reusing software components and less formality in team. Below figure 2.5 illustrates the work flow of RAD model.

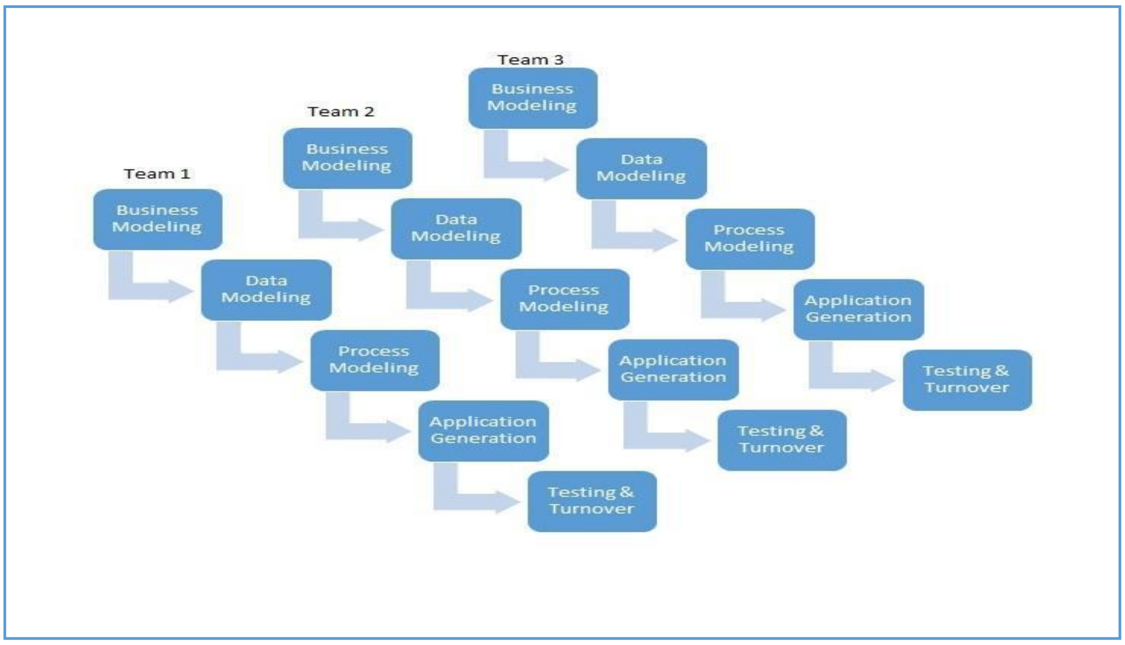


Figure 2. 5 Rad Model

CHAPTER 03 - DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of system theory to product development. There is some overlap with the disciplines of systems analysis, system architecture and system engineering [1].

This chapter will describe about system requirements, system architecture, subsystems, database structure and user interfaces. And also it will provide class diagrams, Entity Relationship diagrams and other related diagrams.

3.1 ALTERNATIVE SOLUTIONS

The alternate solutions for the Inventory, Connection Management and Inquiry processing System are developing a web base system or using collection of software to perform these tasks.

3.1.1 WEB BASED SYSTEM

Web based system provides access to a single software system over the internet for users wherever they are using a web browser. To implement it additional resources such as web server, database server and hardware (switches, routers and network cables) components are needed. The deployment and maintenance is difficult and in the case of network failure system is unavailable.

3.1.2 REASONS FOR CHOOSING STANDALONE SYSTEM

- Client requested a standalone system
- Needless configuration.
- Easy to maintain and deploy.
- Web application development relies significantly on Internet connectivity
- It can perform much faster than web based system.
- Can use it without Internet facility.

3.2 OBJECT ORIENTED DESIGN

It’s a structured method for analyzing, designing a system by applying the object-orientated concepts, and develop a set of graphical system models during the development life cycle of the software. The analysis phase identifies the objects, their relationship, and behavior using the conceptual model (an abstract definition for the objects). While in design phase, we describe these objects (by creating class diagram from conceptual diagram—usually mapping conceptual model to class diagram), their attributes, behavior, and interactions. The input for object-oriented design is provided by the output of object-oriented analysis. But, analysis and design may occur in parallel, and the results of one activity can be used by the other.

The following object models were used for the designing process of the system.

3.2.1 USE CASE DIAGRAMS

Critical users in the system

- Manager
- Administrator
- Owner
- Clerk
- Receptionist

A UML high level essential use case is a brief description of the main processes used to accomplish the system function. A high level use case each of the major processes in the system and therefore there is one high level use case for each of the major processes. Figure 3.1 shows high level use cases of user management module which is related to proposed system. And use case description of user management is listed in table 3.1

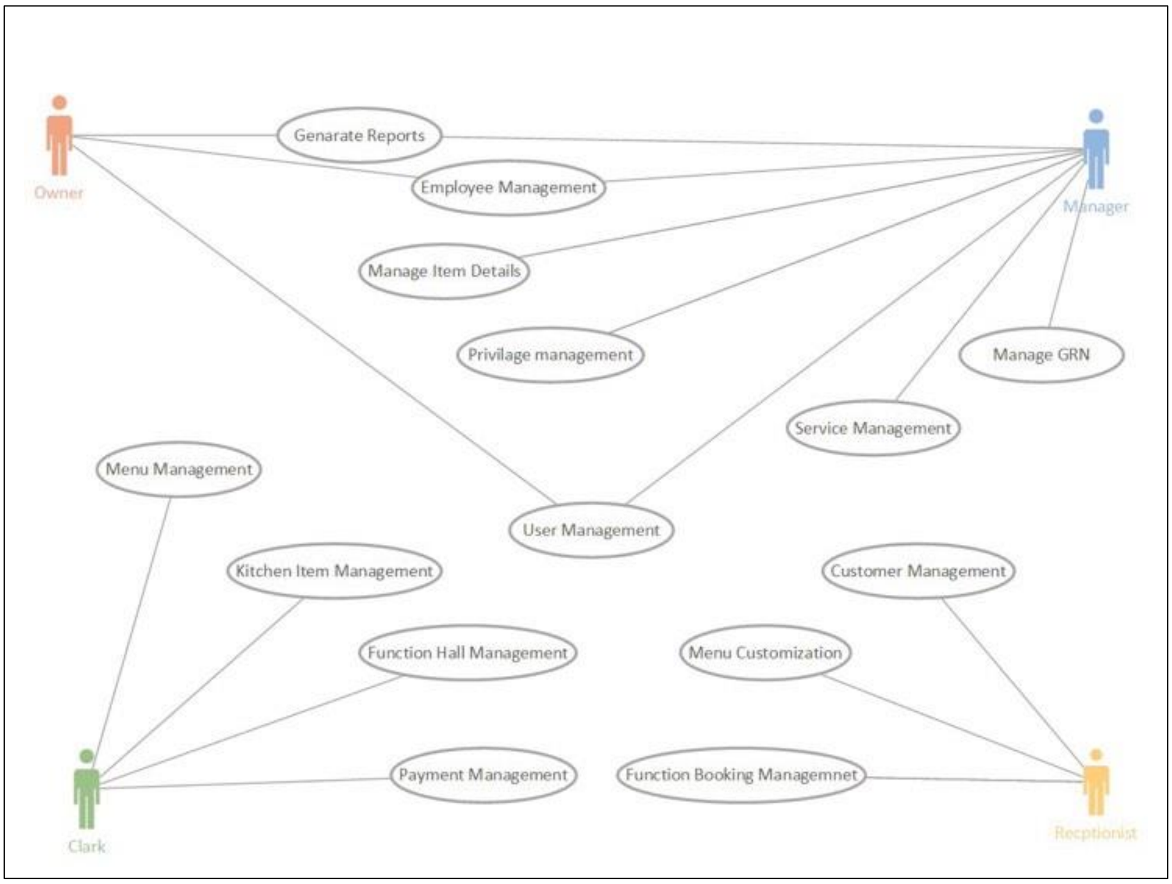


Figure 3. 1 Highlevel Usecase diagram

Use Case	Login
Actor	All Users(Manager, Administrator, Receptionist, Front desk clerk, Accountant, Billing clerk)
Overview	
Registered users can login to the system	
Precondition(s)	
Users must be registered with the system and must have a valid user name and password.	
Flow of Events	
Enter a valid user name and password.	
Post Conditions	
Show error message, Reject login to the system.	

Table 3. 1 Use case description for Login

The following figure 3.2 shows use cases of Function Booking Management module which is related to front desk clerk and administrator. In here the guest can be an individual or a company. And use case description of booking management has listed in table 3.2

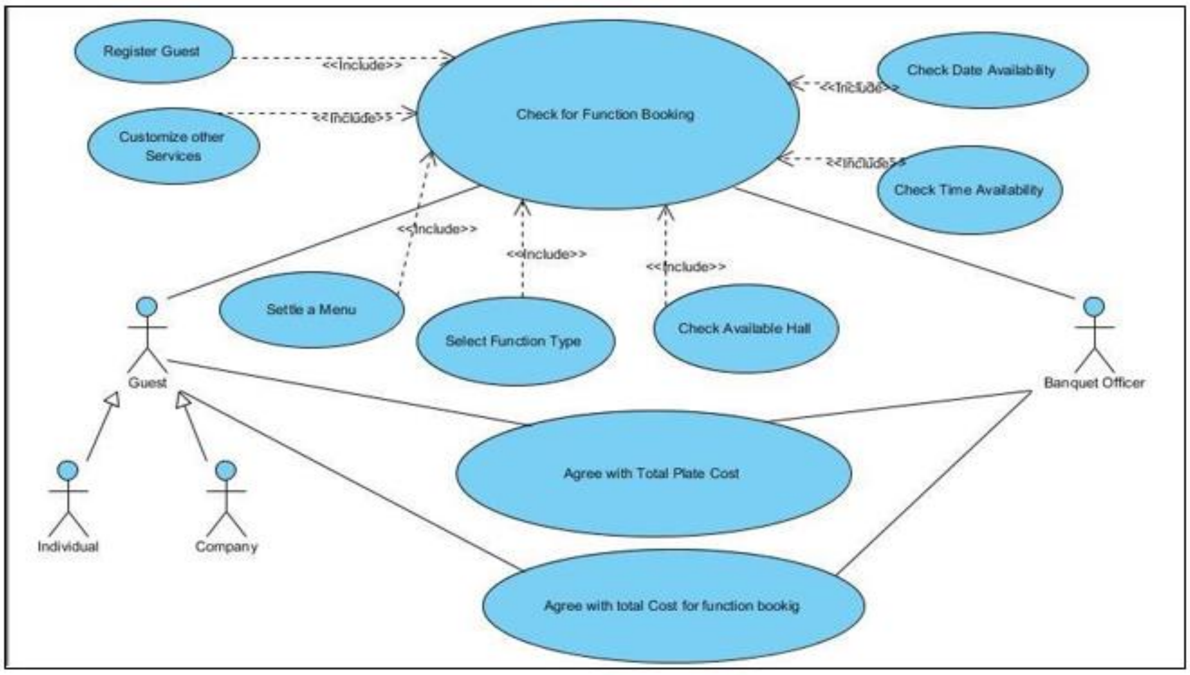
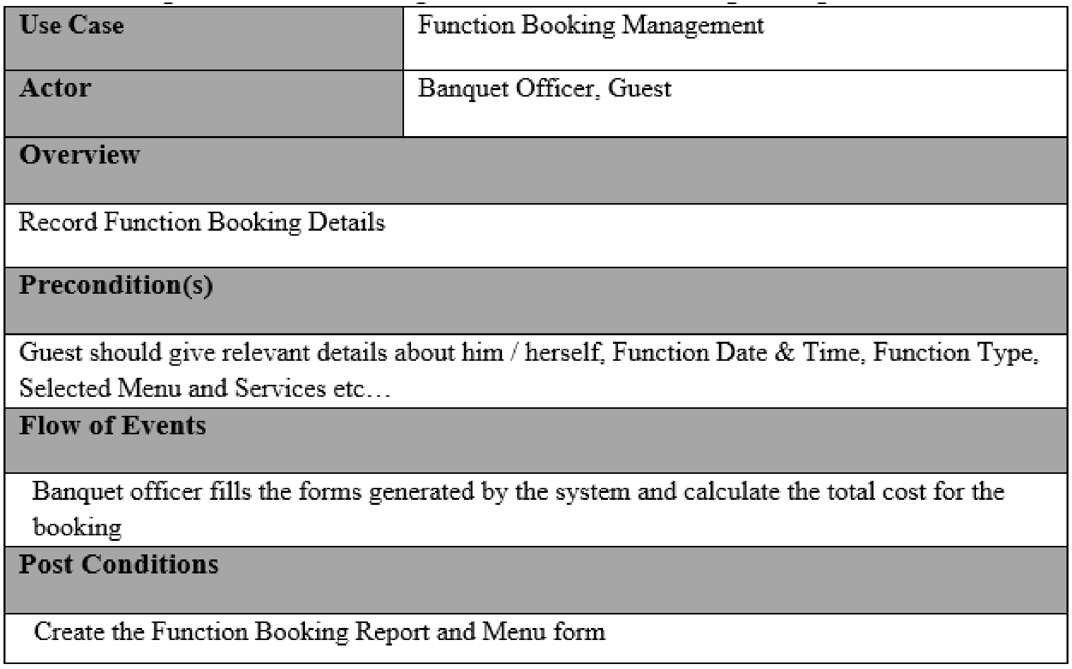


Figure 3. 2 Use case diagram for Function Booking Management



Use Case	Function Booking Management
Actor	Banquet Officer, Guest
Overview	
Record Function Booking Details	
Precondition(s)	
Guest should give relevant details about him / herself, Function Date & Time, Function Type, Selected Menu and Services etc…	
Flow of Events	
Banquet officer fills the forms generated by the system and calculate the total cost for the booking	
Post Conditions	
Create the Function Booking Report and Menu form	

Table 3. 2 Use case description for Function Booking Module

The following figure 3.3 shows use cases of report generation module which is related to manager and users of staff. In here the reports can be categorized as yearly, monthly, weekly and daily. User can generate reports providing relevant criteria. And use case description of report generation module is listed in table 3.3

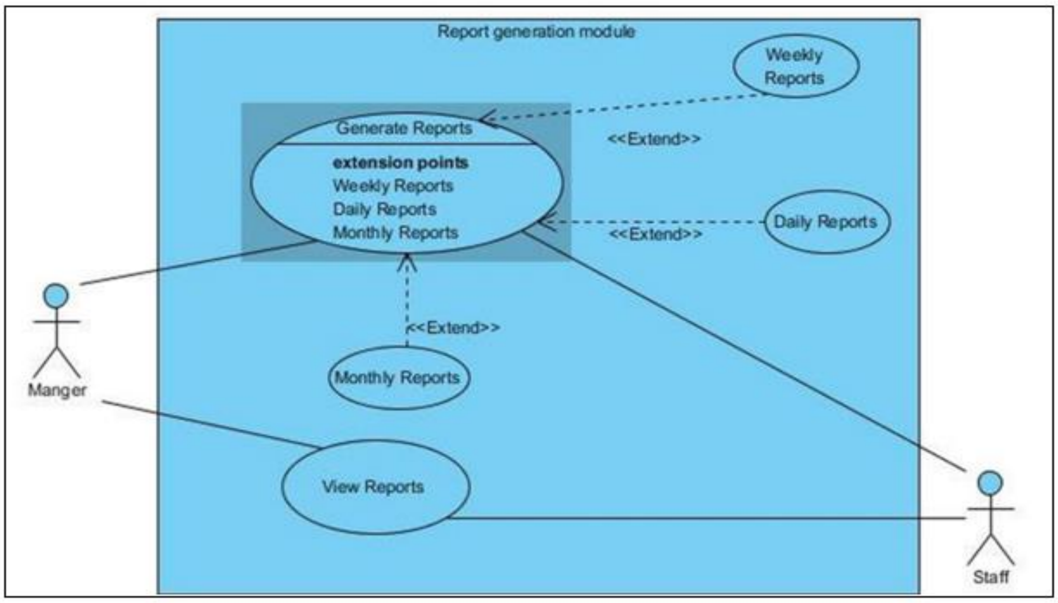


Figure 3. 3 Use case diagram for report generation

Use Case	View reports
Actor	Manager and other users (staff)
Overview	
Adding details about report criteria.	
Precondition(s)	
Manager should enter relevant information.	
Flow of Events	
Manager selects the relevant criteria to generate reports.	
Post Conditions	
Create the relevant report.	

Table 3. 3 Table 3.3 Use case description for View reports

3.2.2 SEQUENCE DIAGRAMS

A sequence diagram is a kind of interaction diagram that represents how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

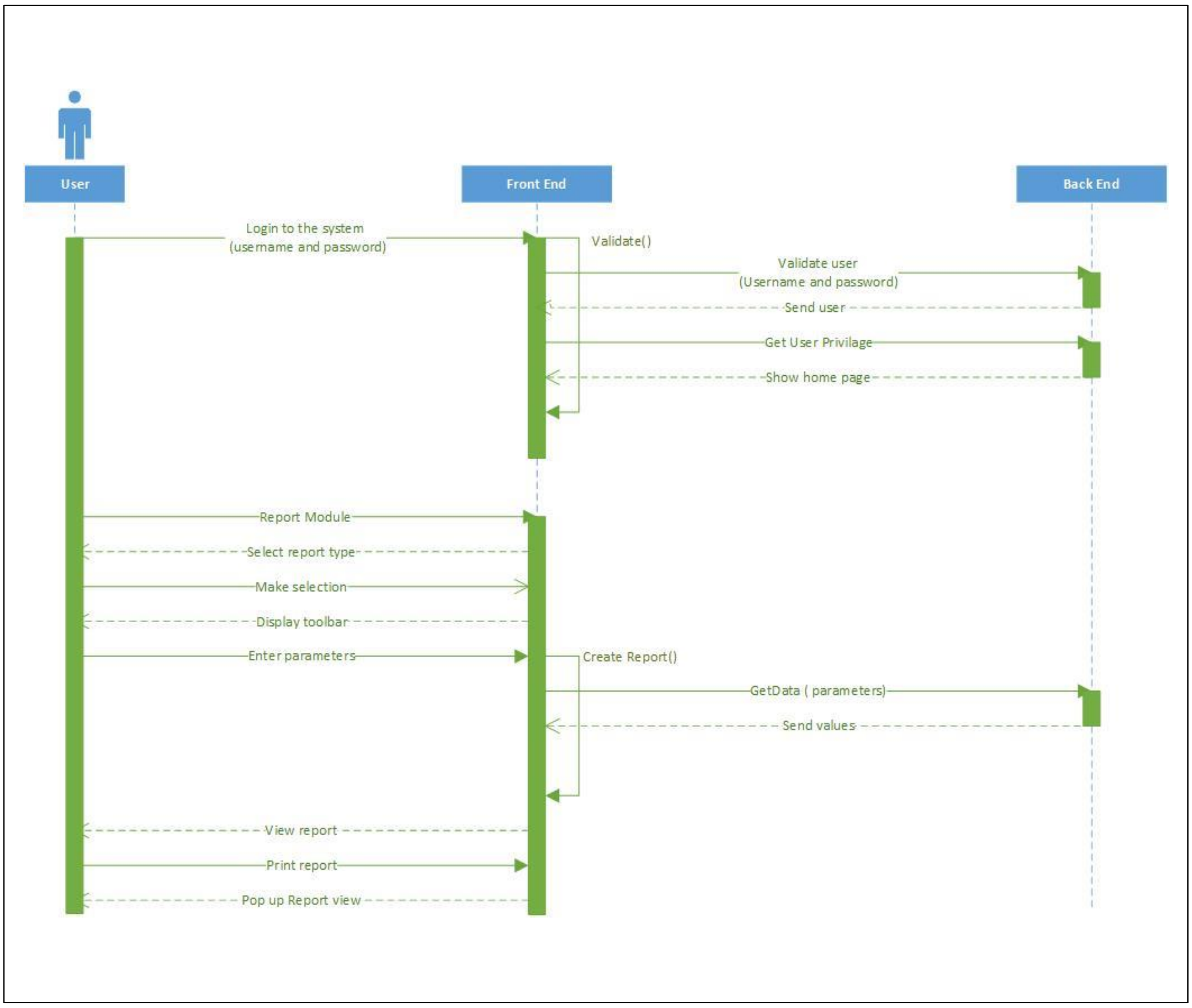


Figure 3. 4 Sequence diagram for Report Generation

3.2.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes.

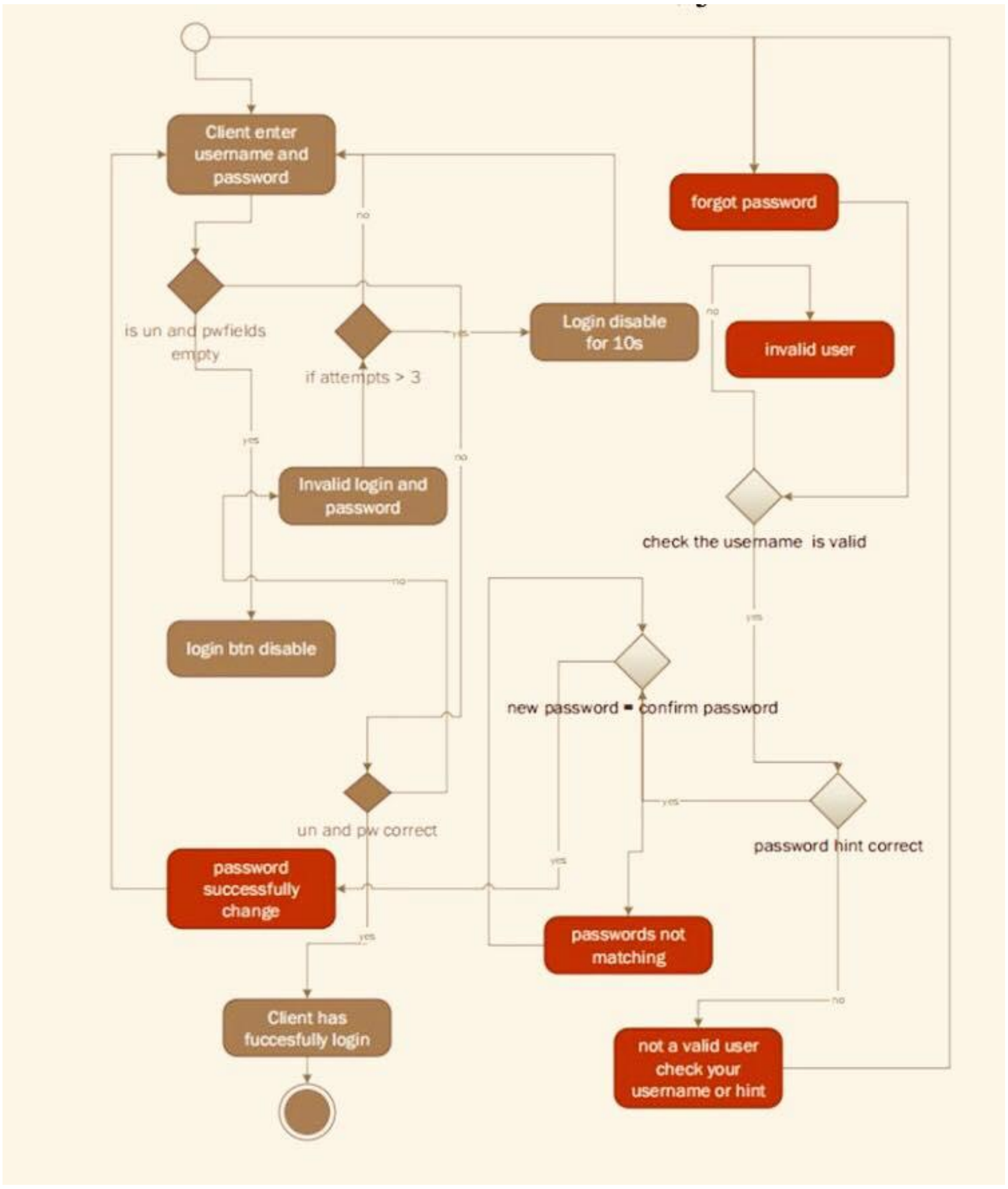

Client enter username and password
forgot password
no
is un and pwfields empty
yes
if attempts > 3
Login disable for 10s
no
invalid user
yes
no
Invalid login and password
check the username is valid
no
login btn disable
yes
new password = confirm password
un and pw correct
yes
password hint correct
password successfully change
yes
passwords not matching
yes
no
Client has fuccesfully login
not a valid user check your username or hint

Figure 3. 5 Activity diagram for User Login

3.2.4 CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

The following class diagram in Figure 3.6 depicts the overall class diagram of the system.

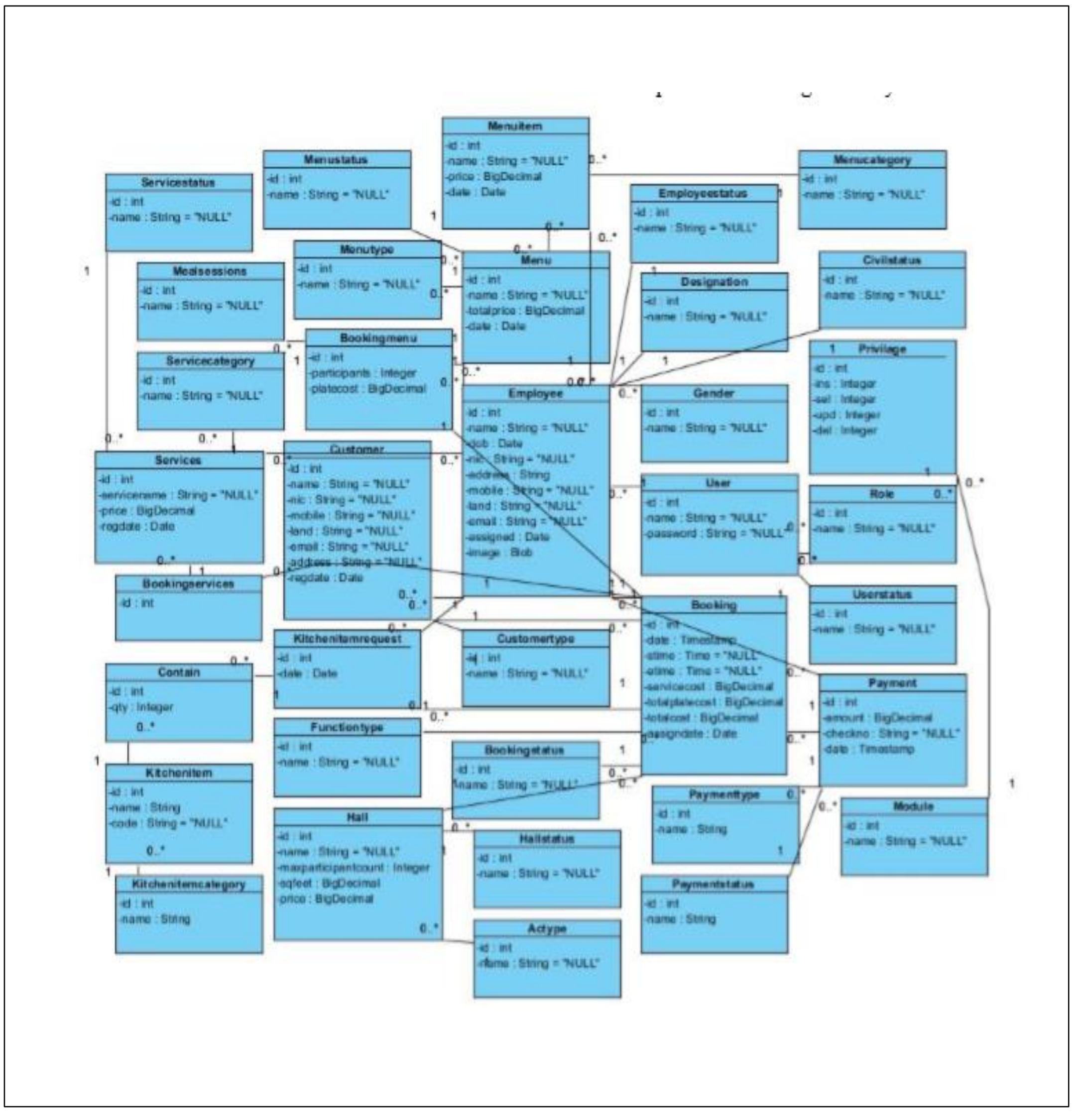


Figure 3. 6 Class diagram for the system

3.3 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity [2].

There are few steps in database design process. The proposed system was created according to these steps.

- Determine the purpose of database
- Determine the tables need
- Determine the fields need
- Determine the relationships
- Refine design

To avoid loss of data and data redundancy the database was normalized up to 3rd

Normal Form

3.3.1. DATABASE NORMALIZATION

1st Normal Form - each attribute value should be atomic [3].

2nd Normal Form - A relation is in 2NF if no non key attribute is functionally depending on just a part of the key [3].

3rd Normal Form - A non-key attribute should not be functionally depended on another Non-key attribute [3].

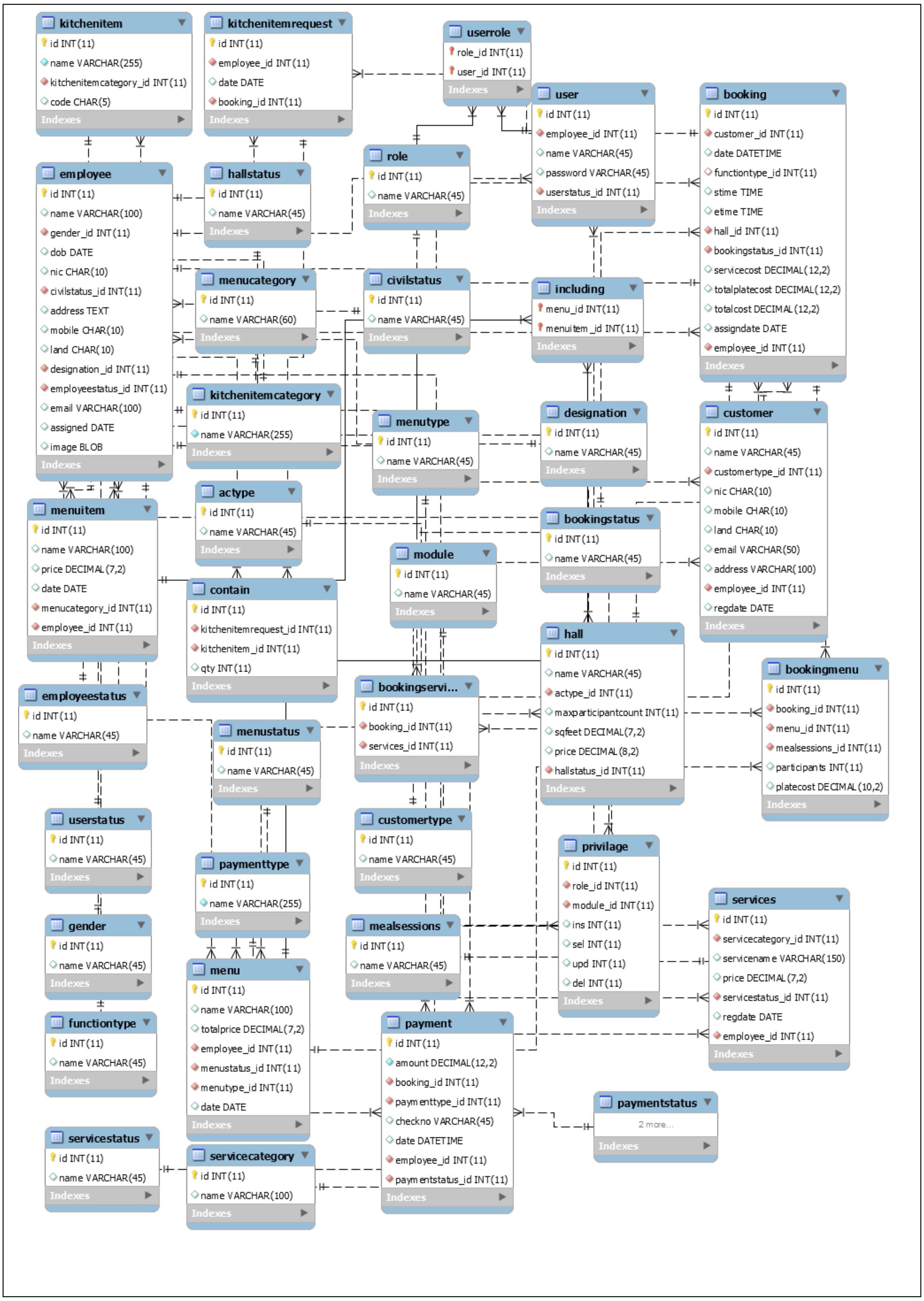


Figure 3. 7 Database design for the system

3.4 USER INTERFACE DESIGN

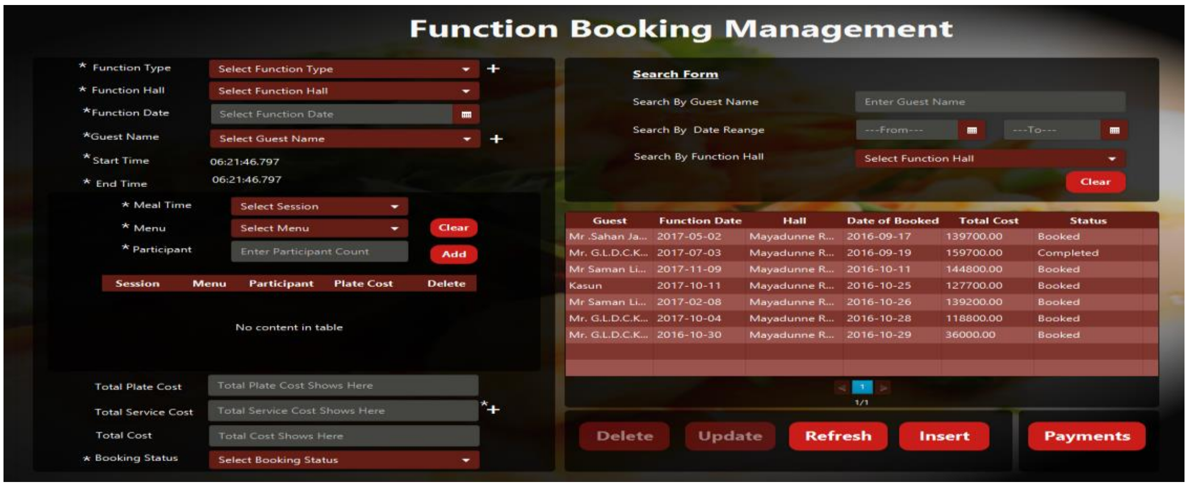
The 10 most general principles for interaction design. They are called 'heuristics'.

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

When creating the proposed system above principles were achieved as possible. Some of the user interfaces are shown below.

3.4.1 FUNCTION BOOKING INTERFACE

Figure 3.8 represent the Function Booking Management Module which can make bookings. This interface provide facility to select function type, function date, Select food menu and additional services and shows total booking cost and provide pagination to data table to view bookings page by page.



Function Booking Management

* Function Type	Select Function Type
* Function Hall	Select Function Hall
*Function Date	Select Function Date
*Guest Name	Select Guest Name
* Start Time	06:21:46.797
* End Time	06:21:46.797
* Meal Time	Select Session
* Menu	Select Menu
* Participant	Enter Participant Count

Session	Menu	Participant	Plate Cost	Delete

No content in table

Total Plate Cost	Total Plate Cost Shows Here
Total Service Cost	Total Service Cost Shows Here
Total Cost	Total Cost Shows Here
* Booking Status	Select Booking Status

Search Form

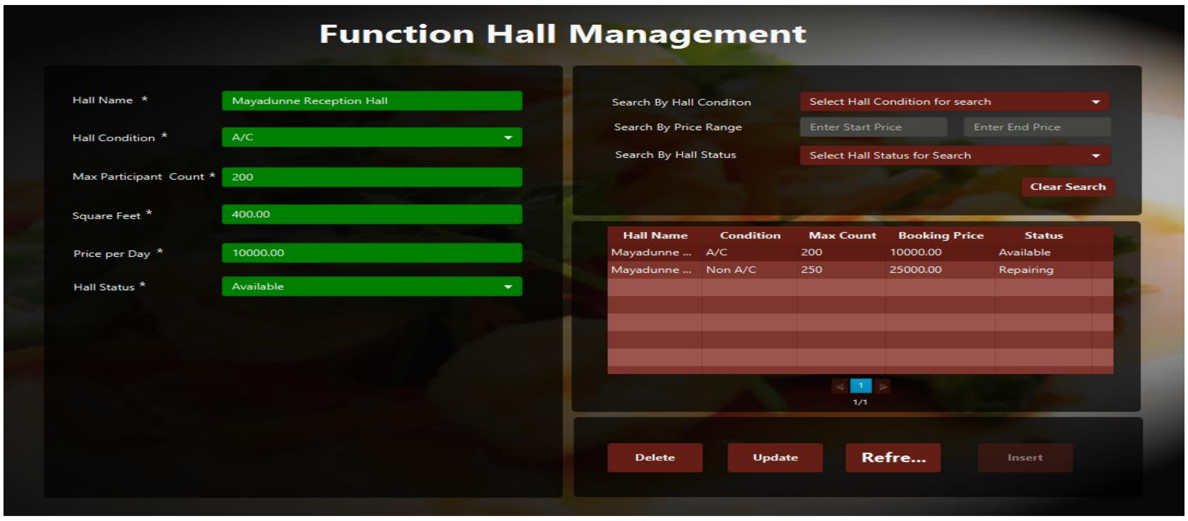
Search By Guest Name	Enter Guest Name
Search By Date Reange	---From--- ---To---
Search By Function Hall	Select Function Hall

Guest	Function Date	Hall	Date of Booked	Total Cost	Status
Mr .Sahan Ja...	2017-05-02	Mayadunne R...	2016-09-17	139700.00	Booked
Mr. G.L.D.C.K...	2017-07-03	Mayadunne R...	2016-09-19	159700.00	Completed
Mr Saman Li...	2017-11-09	Mayadunne R...	2016-10-11	144800.00	Booked
Kasun	2017-10-11	Mayadunne R...	2016-10-25	127700.00	Booked
Mr Saman Li...	2017-02-08	Mayadunne R...	2016-10-26	139200.00	Booked
Mr. G.L.D.C.K...	2017-10-04	Mayadunne R...	2016-10-28	118800.00	Booked
Mr. G.L.D.C.K...	2016-10-30	Mayadunne R...	2016-10-29	36000.00	Booked

Figure 3. 8 Interface design for Function Booking management

3.4.2 HALL MANAGEMENT INTERFACE

Figure 3.9 represent the hall registration which can add new hall into the system. This interface provide facility to select hall condition, max count, square feet, price per day and status. Also it provides pagination to data table to view added hall page by page.



Function Hall Management

Hall Name *	Mayadunne Reception Hall
Hall Condition *	A/C
Max Participant Count *	200
Square Feet *	400.00
Price per Day *	10000.00
Hall Status *	Available

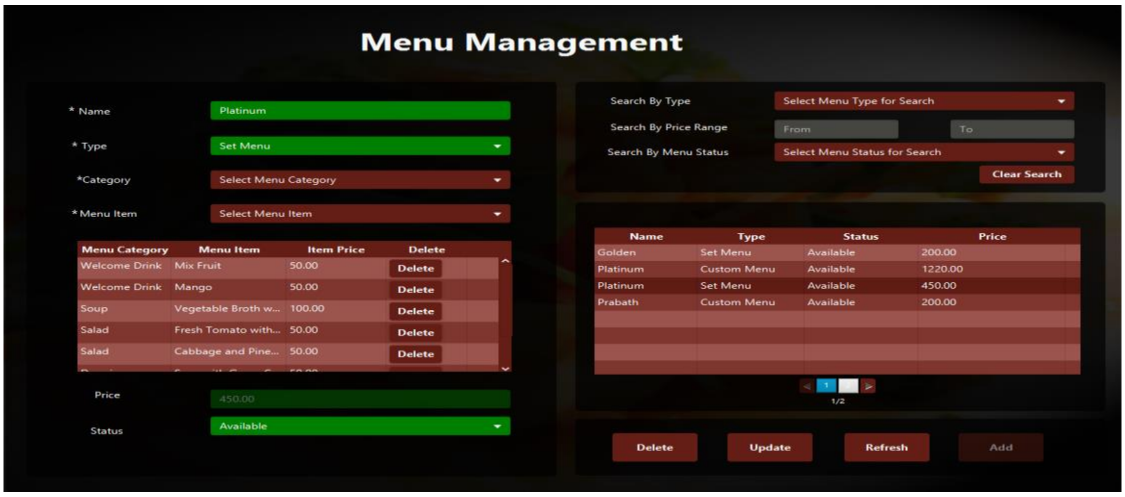
Search By Hall Conditon	Select Hall Condition for search
Search By Price Range	Enter Start Price Enter End Price
Search By Hall Status	Select Hall Status for Search

Hall Name	Condition	Max Count	Booking Price	Status
Mayadunne ...	A/C	200	10000.00	Available
Mayadunne ...	Non A/C	250	25000.00	Repairing

Figure 3. 9 Interface design Function Hall Management

3.4.3 MENU MANAGEMENT INTERFACE

Figure 3.10 represent the customize food menus as which can save for the function. This interface facilitate to select menu type, menu category, menu item and status. Also shows menu price and provides pagination to data table to view added food menus page by page.



* Name	Platinum
* Type	Set Menu
*Category	Select Menu Category
* Menu Item	Select Menu Item

Menu Category	Menu Item	Item Price	Delete
Welcome Drink	Mix Fruit	50.00	Delete
Welcome Drink	Mango	50.00	Delete
Soup	Vegetable Broth w...	100.00	Delete
Salad	Fresh Tomato with...	50.00	Delete
Salad	Cabbage and Pine...	50.00	Delete

Price	450.00
Status	Available
Search By Type	Select Menu Type for Search
Search By Price Range	From / To
Search By Menu Status	Select Menu Status for Search

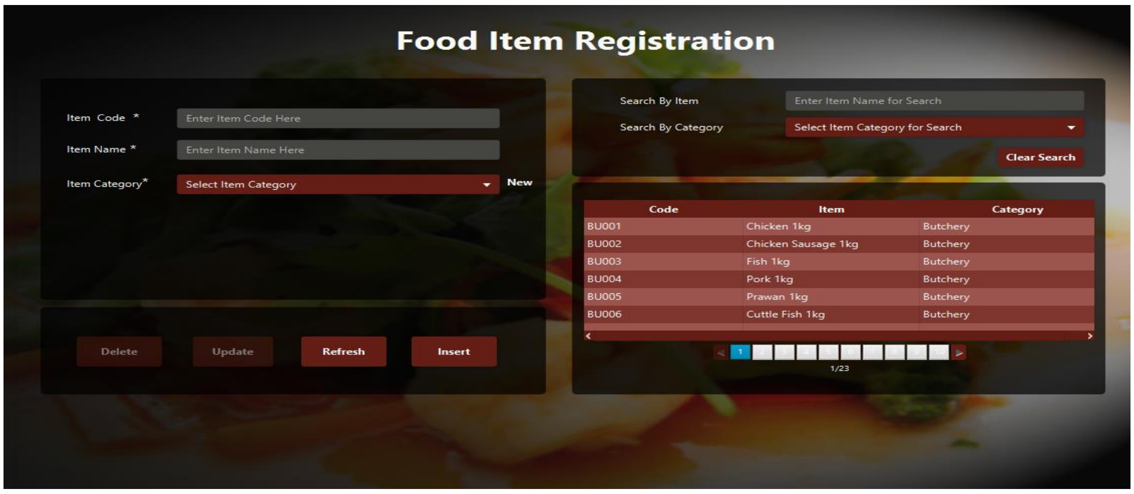
Name	Type	Status	Price
Golden	Set Menu	Available	200.00
Platinum	Custom Menu	Available	1220.00
Platinum	Set Menu	Available	450.00
Prabath	Custom Menu	Available	200.00

1/2

Figure 3. 10 Interface design for Menu Management

3.4.4 KITCHEN ITEM REGISTRATION INTERFACE

Figure 3.12 represent the Kitchen Item Registration which can save kitchen items for create Kitchen Item Request Form. It provides pagination to data table to view added food menus page by page.



Item Code *	Enter Item Code Here
Item Name *	Enter Item Name Here
Item Category*	Select Item Category
Search By Item	Enter Item Name for Search
Search By Category	Select Item Category for Search

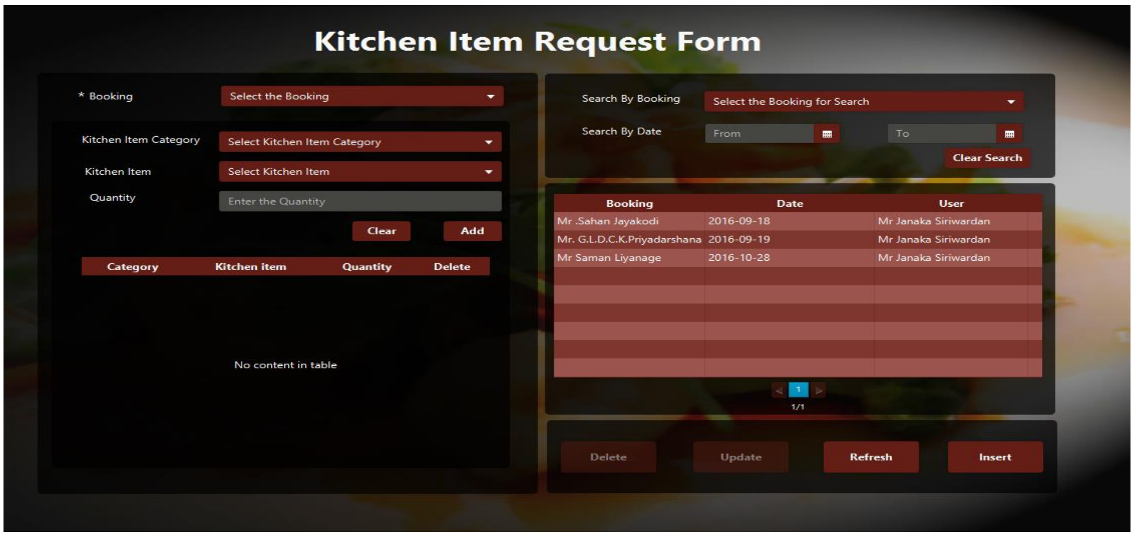
Code	Item	Category
BU001	Chicken 1kg	Butchery
BU002	Chicken Sausage 1kg	Butchery
BU003	Fish 1kg	Butchery
BU004	Pork 1kg	Butchery
BU005	Prawan 1kg	Butchery
BU006	Cuttle Fish 1kg	Butchery

1/23

Figure 3. 11 Interface design for Kitchen Item Registration

3.4.5 KITCHEN ITEM REQUEST FORM INTERFACE

Figure 3.13 represent the Kitchen Item Request Form which can create Kitchen Item Request Form. This interface facilitate to create request form for relevant food menu that guest have selected.



Kitchen Item Request Form

* Booking	Select the Booking
Kitchen Item Category	Select Kitchen Item Category
Kitchen Item	Select Kitchen Item
Quantity	Enter the Quantity

Category	Kitchen item	Quantity	Delete

No content in table

Search By Booking	Select the Booking for Search
Search By Date	From / To

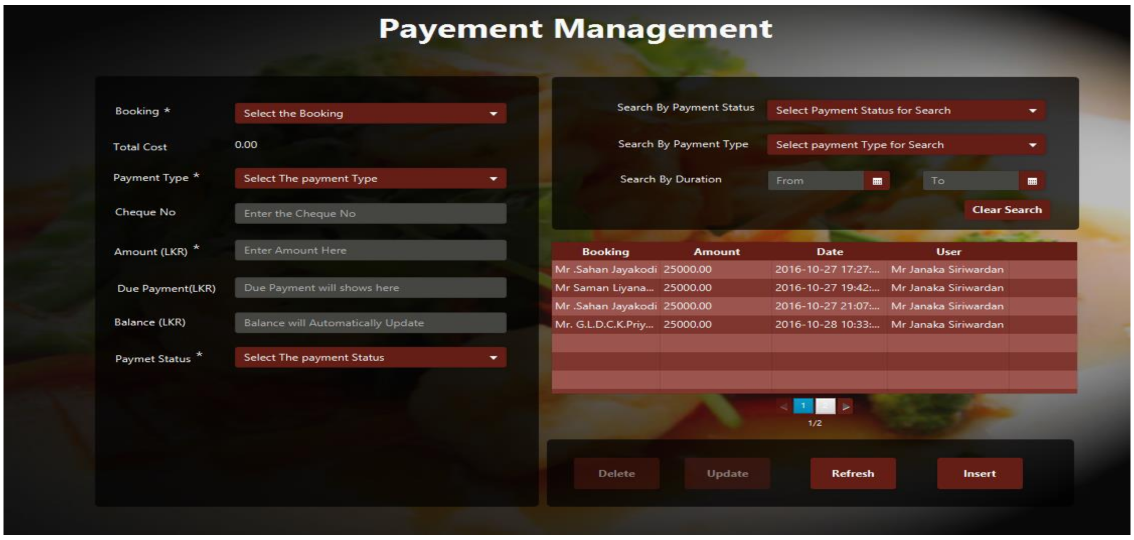
Booking	Date	User
Mr .Sahan Jayakodi	2016-09-18	Mr Janaka Siriwardan
Mr. G.L.D.C.K.Priyadarshana	2016-09-19	Mr Janaka Siriwardan
Mr Saman Liyanage	2016-10-28	Mr Janaka Siriwardan

1/1

Figure 3. 12 Interface design for Kitchen Item Request Form

3.4.6 PAYMENT MANAGEMENT INTERFACE

Figure 3.14 represent the Payment Management Form. One can pay several payments such as Deposit Payment and Full Balance or Complete Full Payment.



Payement Management

Booking *	Select the Booking
Total Cost	0.00
Payment Type *	Select The payment Type
Cheque No	Enter the Cheque No
Amount (LKR) *	Enter Amount Here
Due Payment(LKR)	Due Payment will shows here
Balance (LKR)	Balance will Automatically Update
Paymet Status *	Select The payment Status

Search By Payment Status	Select Payment Status for Search
Search By Payment Type	Select payment Type for Search
Search By Duration	From / To

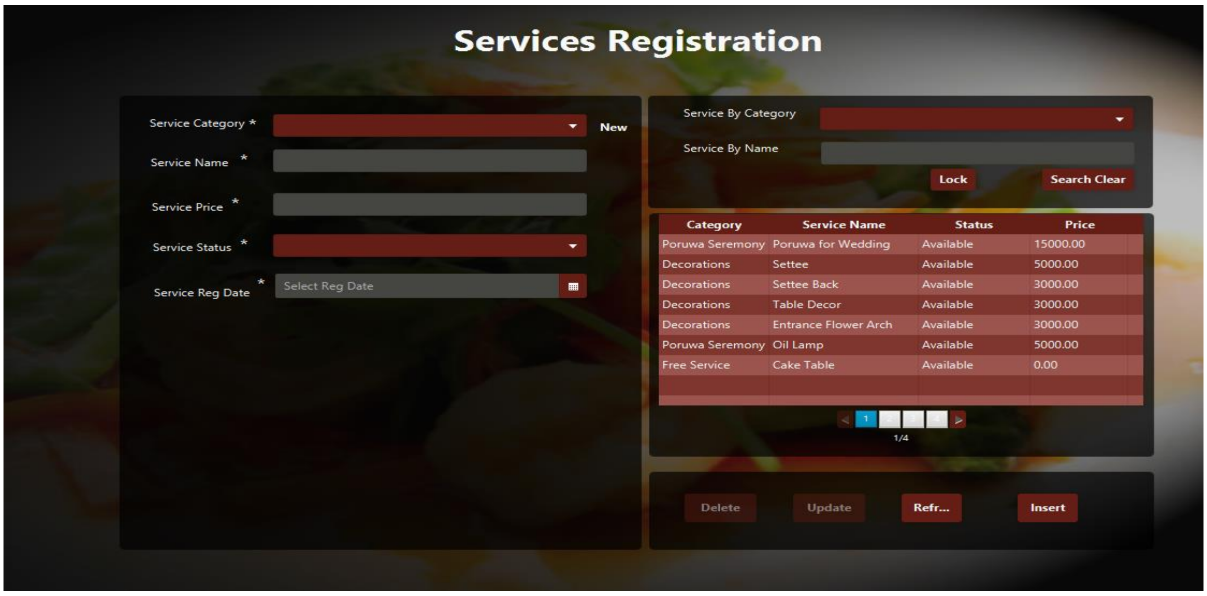
Booking	Amount	Date	User
Mr .Sahan Jayakodi	25000.00	2016-10-27 17:27:...	Mr Janaka Siriwardan
Mr Saman Liyana...	25000.00	2016-10-27 19:42:...	Mr Janaka Siriwardan
Mr .Sahan Jayakodi	25000.00	2016-10-27 21:07:...	Mr Janaka Siriwardan
Mr. G.L.D.C.K.Priy...	25000.00	2016-10-28 10:33:...	Mr Janaka Siriwardan

1/2

Figure 3. 13 Interface design for Payment Management

3.4.7 SERVICES REGISTRATION INTERFACE

Figure 3.14 represent the Services Registration which can save Services for provide Services.



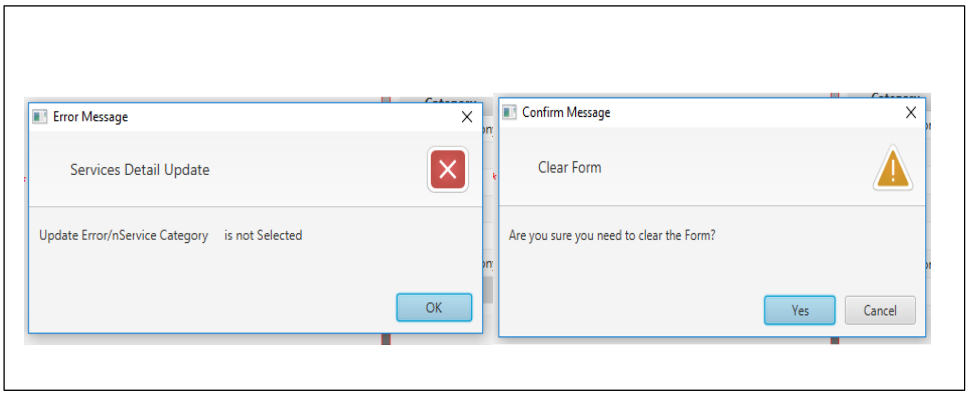
Category	Service Name	Status	Price
Poruwa Seremony	Poruwa for Wedding	Available	15000.00
Decorations	Settee	Available	5000.00
Decorations	Settee Back	Available	3000.00
Decorations	Table Decor	Available	3000.00
Decorations	Entrance Flower Arch	Available	3000.00
Poruwa Seremony	Oil Lamp	Available	5000.00
Free Service	Cake Table	Available	0.00

Figure 3. 14 Interface design for services Registration

3.4.8 SYSTEM NOTIFICATIONS

Before saving, updating, deleting record, system checks authorization from the user. Following 3.15, 3.16 figures shows the confirmation message and warning message which is displaying in relevant authorization statues respectively.

Error Messages



Services Detail Update

Update Error/nService Category is not Selected

Clear Form

Are you sure you need to clear the Form?

Figure 3. 15 Interface design for Error Messages

CHAPTER 04 – IMPLEMENTATION

After conclusion of the design phase, implementation phase of the system initiates allowing to what the design phase was scheduled by using suitable techniques, strategies and tools. The business strategies were made in the design phase transforms to technical constraints which verify the client requirements at this implementation phase.

The major purpose of this chapter is to discover about the implementation environment, techniques, tools and also the modularized components used to develop the system. The modularized code fragments have been comprised to support the development functionalities of the system in this chapter. Code fragmentations which were comprised in the system with a remark are meant to be used for yet to come enhancements.

4.1IMPLEMENTATION ENVIRONMENT

The environment of the business organization and the functional & non-functional requirements of the intended system were taken into consideration when the selection of the set of software tools & other resources. Ensure of the high performance, technology feasibility, maintainability & user friendliness was the important aspects of the selection process.

The hardware and software environment is listed in table 4.1

Hardware Environment	**Software Environment**
Intel(R) Core(TM) i5 - 2410M @ CPU 2.30 GHz	MySQL Server 5.5
4GB RAM	MySQL Workbench 6.3 CE
160GB Hard Disk	MySQL Query Browser 1.2.17 Netbeans IDE 8.1 JavaFX Scene Builder 8.0 Visual Paradigm 10.0 Java JDK 1.8.0_102

Table 4. 1 Implementation Environment

4.2 DEVELOPMENT TOOLS

4.2.1 NETBEANS IDE 8.1

Netbeans 8.1 version was selected and used as the IDE for the improvement of the system. Netbeans is to deliver an extensible IDE that delivers all the tools required to develop desktop, web enterprise, and mobile applications. The adeptness to install plug-ins permits developers to tailor the IDE to their distinct development perceptions. It is an emerging tool principally with Java, but also with more languages, in certain PHP, C/C++, and HTML5. The NetBeans IDE is developed in Java and can run on Windows, OS X, Linux, Solaris and other platforms associate an attuned JVM.

It simplifies collection for Java SE comprises what is desired to begin developing NetBeans plugins and NetBeans Platform centered applications; offering user interface management, windows management & various utilities. Netbeans encloses tinted features as; Swing GUI Builder which offers drags and drops Swing controllers to develop an application's GUI. Profiler is a mechanism that supports to track the speed and memory usage of an application to support to identify drawbacks and memory escapes. Working with collaboration on open source projects are offered by the Developer Alliance**.** Besides the Netbeans Platform is envisioned to be used for APIs, are offered to support for develop desktop applications quickly by managing some of the more mutual tasks (e.g., menus, window management, file access) [15].

4.2.2 JAVA LANGUAGE

Java is a programming language initially developed by James Gosling at Sun

Microsystems (which has later combined into Oracle Corporation) and released in 1995 as a fundamental component of Sun Microsystems' Java platform.

Java applications are obviously compiled to byte code (class file) that can run on several Java Virtual Machine (JVM) irrespective of computer architecture. Java is a concurrent, general purpose, object-oriented language and class-based, that is precisely designed to have as few implementation dependencies as conceivable. It is envisioned to allow application developers "write once, run anywhere" (WORA), meaning that code that works on one platform does not want to be recompiled to run on other. Java is as of 2012 one of the best standard programming languages in use, chiefly for client server web applications.

Java uses a programmed garbage collector to accomplish memory management in the object lifecycle. The programmer defines when objects are created, and the Java runtime is answerable for recovering the memory once objects are no longer in use. Once no situations to an object continue, the isolated memory becomes suitable to be freed automatically by the garbage collector.

The imaginative and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in submission with the terms of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed substitute implementations of those Sun technologies, such as the GNU Compiler for Java and GNU Class path [11].

4.2.3 MYSQL

MySQL is a relational database management system (RDBMS), and liners with no GUI tools to administer MySQL databases or manipulate data enclosed inside the databases. Users may use the encompassed command line tools, or practice MySQL "front-ends", desktop software and web applications that build and manage MySQL databases, formulate database structures, back up data, review status, and work with data records.

The official MySQL Workbench is a free incorporated environment developed by MySQL AB, which allows users to graphically manage MySQL databases and visually formulate database structures. MySQL Workbench switches the prior package of software, MySQL GUI Tools.

The MySQL Query Browser is a graphical tool for constructing, implementing, and enhancing queries in a graphical environment. The MySQL Query Browser is developed to execute query and explore [4].

4.2.4 JAVAFX SCENE BUILDER

Scene Builder is a User Interface outline tool for JavaFX and Visual Framework for FXML. It allows designers and developers to construct JavaFX-centered UIs & Scene Builder are entirely developed with JavaFX 8.0 APIs to discover and study about JavaFX stuffs.

4.2.5 VISUAL PARADIGM

"Visual Paradigm for UML is a CASE tool with various selections for modeling with UML2 diagrams as well as supports SysML requirements diagrams and ER diagrams. The tool has a virtuous working environment, which enables inspecting and influence of the modeling project. It is a professional tool and also helps precise changes to source code of several programming languages such as C++ and Java [5]."

4.2.6 HIBERNATE

Hibernate is an object-relational mapping (ORM) library for the Java language, offering a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate answers object-relational impedance incompatible problems by exchanging direct persistence-related database entrees with high-level object management functions. Hibernates key feature is mapping from Java classes to database tables (and from Java data types to SQL data types). Hibernate also offers data query and retrieval services. It also produces the SQL calls and attempts to release the developer from manual result set management and object adaptation and preserve the application portable to all reinforced SQL databases with slight performance overhead. HQL is contraction of Hibernate Query Language. HQL is SQL encouraged language offered by hibernate. Developer can execute SQL like queries to work with data objects. The top layer is the application layer that comprises application program, which has transient object (POJO/Plain Old Java Object). This layer obtains the amenity of data persistence. The convenience layer contains interfaces and classes that are integral part of the hibernate architecture. These interfaces and classes offer valuable services, such as making of Session Factory, Session interfaces, and transaction processing [6].

Bellow figure 4.1 illustrates the architecture of Hibernate framework. There are different APIs have integrated to Hibernate framework, such as JTA, JDBC and JNDI. It acts as an interface to connect Java Application and Database.

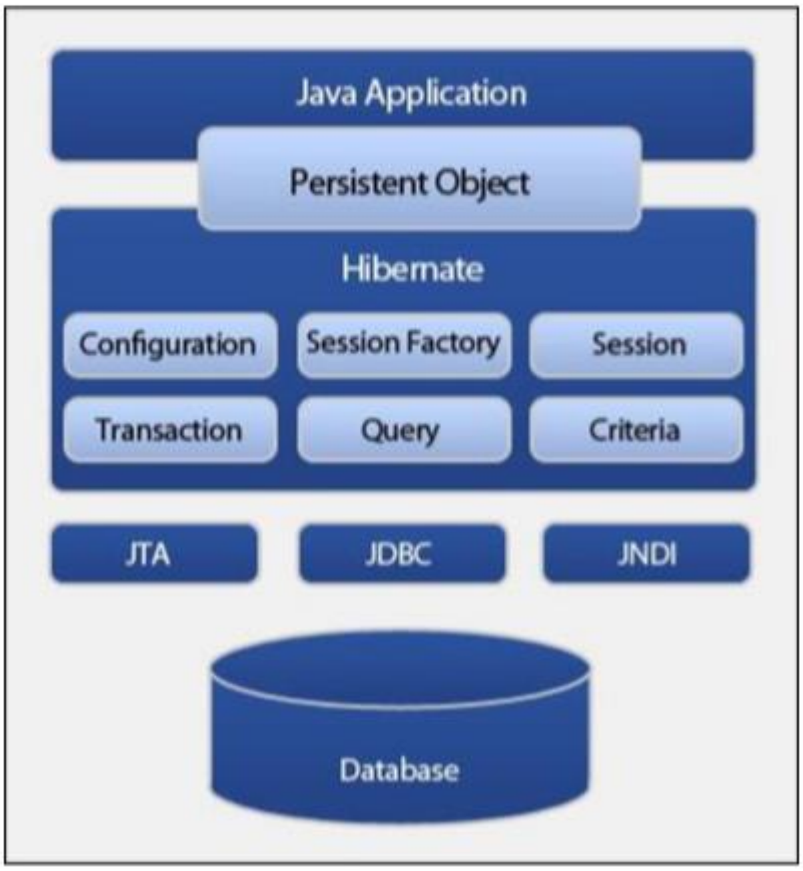


Figure 4. 1 Hibernate Framework

4.2.7 JASPER REPORT

Jasper Reports offers essential features to produce dynamic reports, containing data retrieval using JDBC (Java Database Connectivity), as well as help for parameters, expressions, variables, and groups. Jasper Reports also contains advanced features, such as custom data sources, script lets, and sub reports.

4.2.8 JPA - JAVA PERSISTENCE API

The Java Persistence API is a Java specification for accessing, persisting, and managing data between Java objects / classes and a relational database. JPA is now considered the standard industry approach for Object to Relational Mapping (ORM) in the Java Industry [14].

4.3 IMPLEMENTATION OF THE PROPOSED SYSTEM

The Architecture used to implement the system was MVC model. It states to Model View-Controller. MVC is a most applying design pattern since of its reliability & other key usages. It is reusable & communicative that lets more readable & portable. Model– view–controller (MVC) is a software design pattern for constructing user interfaces. It splits a given software application into three interrelated parts, so as to distinct inner representations of information from the ways that information is offered to or accepted from the user.

Model - This is the layer which switches data in the system. It realizes all facts about data which required being presented. It also controls the rules to entree the data objects and complete any kind of operation on them. This layer is liberated from other system layers such as, View and Controller. Model denotes an object or JAVA POJO carrying data. It can also have logic to modify controller if its data modifications.

View - This is the layer which routines Model’s data querying methods to acquire the data for the purpose of representing. This layer is liberated from application logic. A view must guarantee that its presence replicates the state of the model.

Controller - Controller acts on both model and view. It controls the data stream into model object and updates the view whenever data changes. It keeps view and model separate.

Figure 4.2 shows how MVC architecture works.

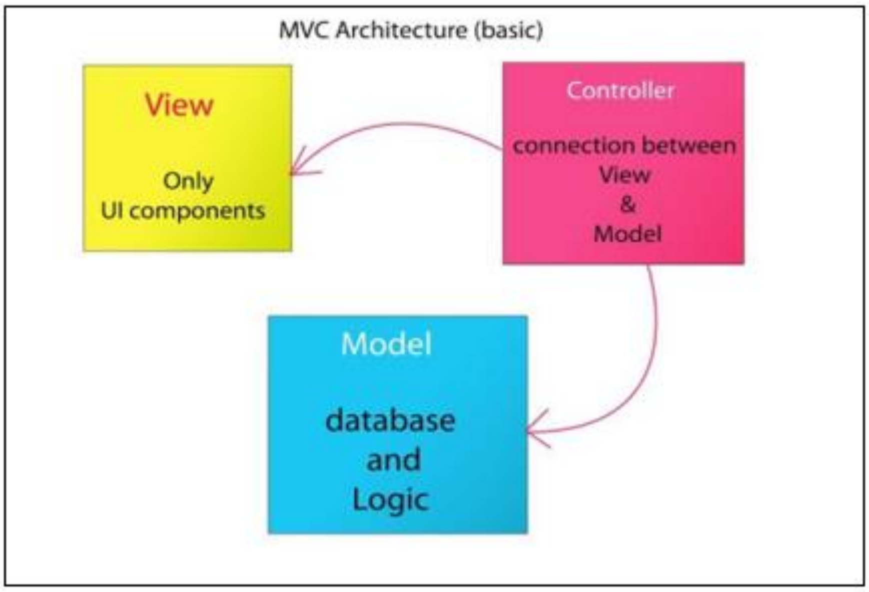


Figure 4. 2 MVC Architecture

4.4 MODEL (DATA) LAYER IMPLEMENTATION

Hibernate is considered with data persistence as it transmits to relational databases

(RDBMS). In Object-Oriented applications, there is consuming an object database (ODBMS) as opposite to a RDBMS. In this layer there are tables and relations between them - one-to-one, one-to-many, many-to-many, etc. It helps to perform CRUD operations in order to insert data for the database, read data from them as rows, update the table data, & also delete the needless data.

4.4.1 HIBERNATE CONFIGURATION

Hibernate requires to identify in advance where to discover the mapping information that determines how your Java classes transmit to the database tables. Hibernate also needs a set of configuration settings linked to database and other related parameters. All such information is typically supplied as usual Java properties file termed hibernate. Properties, or as an XML file named hibernate.cfg.xml. An occurrence of org.hibernate.cfg.Configuration embodies an entire set of mappings of an application's Java types to an SQL database. The org.hibernate.cfg.Configuration is used to figure an absolute org.hibernate.SessionFactory.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernateconfiguration-3.0.dtd">
<hibernate-configuration>
<session-factory> <property
 name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</propey>
 <property
 name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</proty>
 <property
 name="hibernate.connection.url">jdbc:mysql://localhost:3306/earthunive
 rsity?zeroDateTimeBehavior=convertToNull</property>
 <property name="hibernate.connection.username">root</property>
 <property name="hibernate.connection.password">bit</property>
 <property name="hibernate.show_sql">false</property>
 <mapping class="entity.Civilstatus"/>
 <mapping class="entity.Designation"/>
 <mapping class="entity.Employee"/>
 <mapping class="entity.Employeestatus"/>
 <mapping class="entity.Gender"/>
 <mapping class="entity.Module"/>
 <mapping class="entity.Privilage"/>
 <mapping class="entity.Role"/>
 <mapping class="entity.User"/>
 <mapping class="entity.Userstatus"/>
 <mapping class="entity.Actype"/>
 <mapping class="entity.Hall"/>


  <mapping class="entity.Hallstatus"/>
 <mapping class="entity.Menucategory"/>
 <mapping class="entity.Menuitem"/>
 <mapping class="entity.Servicecategory"/>
 <mapping class="entity.Services"/>
 <mapping class="entity.Servicestatus"/>
```

```
<mapping class="entity.Menu"/>
<mapping class="entity.Menustatus"/>
<mapping class="entity.Menutype"/>
<mapping class="entity.Booking"/>
<mapping class="entity.Bookingmenu"/>
<mapping class="entity.Bookingstatus"/>
<mapping class="entity.Functiontype"/>
<mapping class="entity.Mealsessions"/>
<mapping class="entity.Customer"/>
<mapping class="entity.Customertype"/>
<mapping class="entity.Bookingservices"/>
<mapping class="entity.Contain"/>
<mapping class="entity.Kitchenitem"/>
<mapping class="entity.Kitchenitemcategory"/>
<mapping class="entity.Payment"/>
<mapping class="entity.Paymentstatus"/>
<mapping class="entity.Paymenttype"/>
<mapping class="entity.Kitchenitemrequest"/>
</session-factory>
</hibernate-configuration>
```

Code 4.1 hibernate.cfg.xml

4.4.2 JAVA ENTITIES WITH ANNOTATIONS AND NAMED QUERIES

Java entities whose objects or instances will be warehoused in database tables are called persistent classes in Hibernate. Hibernate works best if these entities follow some simple rules, also known as the Plain Old Java Object (POJO) programming model. An annotation, in the Java computer programming language, is a form of syntactic metadata that can be added to Java source code. Classes, methods, variables, parameters and packages may be annotated.

```
@Entity
@Table(name = "menuitem")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Menuitem.findAll", query = "SELECT m FROM Menuitem m"),
    @NamedQuery(name = "Menuitem.findById", query = "SELECT m FROM Menuitem m WHERE m.id = :id"),
```

@NamedQuery(name = "Menuitem.findByName", query = "SELECT m FROM Menuitem m WHERE m.name = :name"),

@NamedQuery(name = "Menuitem.findByDate", query = "SELECT m FROM Menuitem m WHERE m.date = :date"),

@NamedQuery(name = "Menuitem.getAllByCategory", query = "SELECT m FROM Menuitem m WHERE m.menucategoryId = :category"),

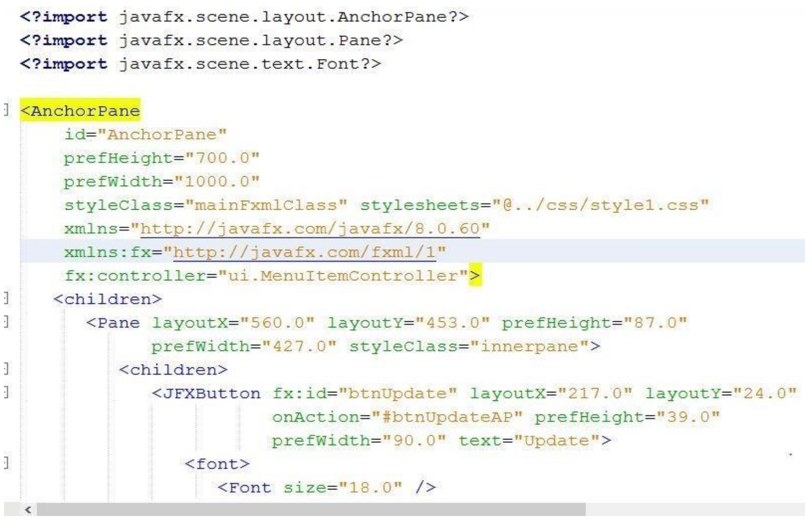
@NamedQuery(name = "Menuitem.findMenuItemByCategory", query = "SELECT m FROM Menuitem m WHERE m.menucategoryId = :category")

})

Code 4.2 Menuitem.java

4.5 VIEW (INTERFACE) LAYER IMPLEMENTATION

Interface is the layer that interrelates with the user & it is a representation of the logic behind in the database. Interfaces simplify users to do required operations using forms as well as loading views. JavaFX Scene Builder designer view was used for the construction of the user interfaces and to add controls. Adobe Photoshop was used to make and modify necessary images. To enhance controllers to the user interface designer view offers with the drag and drop functionality and then the component can be setup as the developer wish. When developer adds a new controller to the user interface JavaFX Scene Builder will automatically produces the necessary xml code for that, which reduces the developing time significantly.



```
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>

<AnchorPane
    id="AnchorPane"
    prefHeight="700.0"
    prefWidth="1000.0"
    styleClass="mainFxmlClass" stylesheets="@../css/style1.css"
    xmlns="http://javafx.com/javafx/8.0.60"
    xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="ui.MenuItemController">
   <children>
      <Pane layoutX="560.0" layoutY="453.0" prefHeight="87.0"
            prefWidth="427.0" styleClass="innerpane">
         <children>
            <JFXButton fx:id="btnUpdate" layoutX="217.0" layoutY="24.0"
                       onAction="#btnUpdateAP" prefHeight="39.0"
                       prefWidth="90.0" text="Update">
               <font>
                  <Font size="18.0" />
```

Code 4.3 MenuItem.fxml

Following represents the java code segment which use for the controller class (Interface Controller) for above interface.

```
package ui; public class menuItemUIController implements
Initializable { private void loadForm(){  menuitem = new
Menuitem(); oldMenuitem = null; txtBuilding.setText("");
txtName.setText(""); txtDescription.setText("");
txtRemark.setText(""); txtNameSearch.setText("");
fillTableBuilding(BuildingDao.getAll());
setStyle(FXStyle.initial); disableButtons(false, true,
true);
    }
Privat  void  setStyle(String  color){  txtBuilding.setStyle(color);
    txtName.setStyle(color);                                          if
    (!txtDescription.getChildrenUnmodifiable().isEmpty()) {

((ScrollPane)txtDescription.getChildrenUnmodifiable().get(0)).getCont
ent().setStyle(color);
        }
        if (!txtRemark.getChildrenUnmodifiable().isEmpty()) {

((ScrollPane)txtRemark.getChildrenUnmodifiable().get(0)).getContent()
.setStyle(color);
        }
    }
private void disableButtons(boolean add, boolean update, boolean
delete){ btnAdd.setDisable(add);
        btnUpdate.setDisable(update);
        btnDelete.setDisable(delete);
    }
    }
```

Code 4.5 MenuitemController.java

Following represents the css code segment which use to apply styling options to above interface.

```
#rich-blue {
    -fx-background-color: #000000, linear-
        gradient(#7ebcea, #2f4b8f),
        linear-gradient(#426ab7, #263e75),
        linear-gradient(#395cab, #223768);
    -fx-background-insets: 0,1,2,3;
    -fx-background-radius: 3,2,2,2;
    -fx-text-fill: white;
    -fx-font-size: 12px;
    -fx-width:60px;
    -fx-height:10px;
}

#rich-blue:hover{
    -fx-background-color:        #000000,
        linear-gradient(aqua,   #2f4b8f),
        linear-gradient(aqua,   #263e75),
        linear-gradient(aqua, #223768);
}
```

Code 4.6 Style1.css

4.6 CONTROLLER (DAO) LAYER IMPLEMENTATION

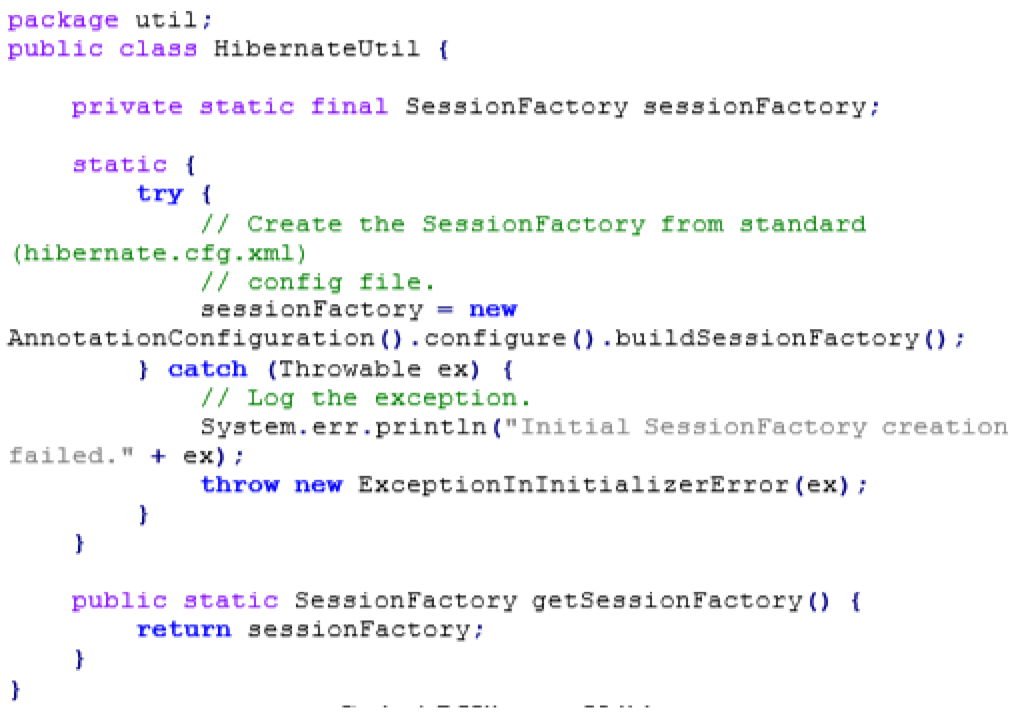
Controller (DAO) layer is the association between data layer & the interface layer. Here logical perception is, Parse a user request (i.e., "read" it), and ensure the user request

(i.e., verify it on forms to application's requirements), define what the user is trying to do (based on form elements), acquire data from the Model (if required) to compromise in answer to user, select the next View the client should see.

The arrangement of calls to the Model (business-logic layer), and/or the arrangement of views and an input from the user expresses the application's workflow. Workflow is accordingly defined in the Controller layer of the application.

4.6.1 HIBERNAT SESSION

The Session Factory is the perception that is a single data collection and thread safe. Since of this feature, many threads can access this simultaneously and the sessions are requested, and also the cache that is immutable of compiled mappings for a specific database. A Session Factory will be constructed only at the time of its startup. In order to access it in the application code, it should be enclosed in singleton. This covering creates the easy approachability to it in an application code, it should be enclosed in singleton. This covering creates the easy approachability to it in an application code.



```
package util;
public class HibernateUtil {

    private static final SessionFactory sessionFactory;

    static {
        try {
            // Create the SessionFactory from standard
(hibernate.cfg.xml)
            // config file.
            sessionFactory = new
AnnotationConfiguration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            // Log the exception.
            System.err.println("Initial SessionFactory creation
failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

Code 4.7 HibernateUtil.java

4.6.2 DAO (DATA ACCESS OBJECTS)

A **data access object** (DAO) is an object that offers an abstract interface to some kind of database or persistence technique, offering some specific operations without revealing details of the database. It offers a mapping from application calls to the persistence layer. This isolation splits the involvements of what data accesses the application requires, in standings of domain-specific objects and data types (the public interface of the DAO), and how these requires can be fulfilled with a specific DBMS, database schema, etc. (the implementation of the DAO).

```java
public class MenuitemDao {
public static void add(Menuitem menuitem){
        CommonDao.add(menuitem);
    }

    /**
     * This method use to get all menuitem list
     * @return menuitem list
     */
    public static ObservableList<Menuitem> getAll(){
        return
(ObservableList<Menuitem>)CommonDao.select("Menuitem.findAll");


    /**
     * This method use to get menuitem to given id
     * @param id id of the menuitem
     * @return menuitem object
     */
    public static Menuitem getById(Integer id){
        HashMap hashMap = new HashMap();
        hashMap.put("id", id);
        return (Menuitem)CommonDao.select("Menuitem.findById",
hashMap).get(0);
    }

     * This method use to delete menuitem
     * @param building menuitem object
     */
    public static void delete(Menuitem menuitem){
        CommonDao.delete(menuitem);
    }
}
```

Code 4.8 MenuitemDao.java

4.7 ACKNOWLEDGEMENT OF REUSED CODE MODULES

I take this place to acknowledge that following code modules were reused for the development of Reception Hall Management System.

- ControlsFx Library

"ControlsFX" is an open source project for JavaFX that aims to provide really high quality UI controls and other tools to complement the core JavaFX distribution. It has been developed for JavaFX 8.0 and beyond.

- JFoenix Library

"JFoenix" is an open source material design project for JavaFX that aims to provide really high quality user friendly UI controls and other tools to complement the core JavaFX distribution. This library has been developed by Google Material design using Java components.

- JasperReports

"JasperReports" is an open source Java reporting tool write to a variety of targets, such as: screen, a printer, into PDF, HTML, and Microsoft Excel. Comma-seperated values or XML files. It can be used in Java-enabled applications, including JavaEE or web applications, to generate dynamic content.

CHAPTER 05 – EVALUATION

Software testing is executed to certify, that the finalized software package tasks according to the expectations described by the requirements/specifications. The whole objective is not to find every software bug that occurs, but to expose situations that could harmfully affect the customer, usability and/or maintainability. In software verification and validation is the procedure of inspection that a software system satisfies specifications and that it achieves its intended purpose.

Verification is procedure of maintaining software to govern whether the products of a given development phase fulfills the situations forced at the beginning of that phase. Validation reviews that the product design fulfills or fits the intended use or the software achieves the user requirements.

Testing can be performed at any phase of the implementation process, but it differs on the test method used. Software testing should be performed specification, design, code generation as well as the execution of business environment using test conditions & test data by a unique test panel discrete from the developer.

5.1 TEST STRATEGIES

The Catering Service and Reception Hall Management System for the Mayadunne Catering Service was tested following strict test plan in order to certify the ultimate product is more reliable.

Testing was performed while development process is going on. Inspection of test cases & test data was carried out following Prototyping.

5.1.1 UNIT TESTING

Unit testing is a testing framework by which individual units of source code, sets of one or more computer program modules together with related control data, usage processes, and operating processes, are tested to describe if they are ready for use. Most of the cases, the smallest testable portion of an application is considered as a unit. In OOP it can be an interface such as a class or an individual service or method. Unit testing was performed while developing the system to verify whether the source codes are accurate and working well.

5.1.2 INTEGRATION TESTING

Alliance of the modules in the system requires the integration testing process to certify that the integrated units also functioning well & gives the expected results. Integration testing is carried out after unit testing done & before the system validation stage. Integration test plan performs as its test units are pre-tested by unit testing.

5.1.3 SYSTEM TESTING

System testing is carried on the whole system in the domain of a Functional requirement Specifications and/or a System Requirement Specification. System testing tests not only the design, but also the performance and even the trusted expectations of the customer. It is also envisioned to check up to and beyond the bounds defined in the software/hardware requirements specifications.

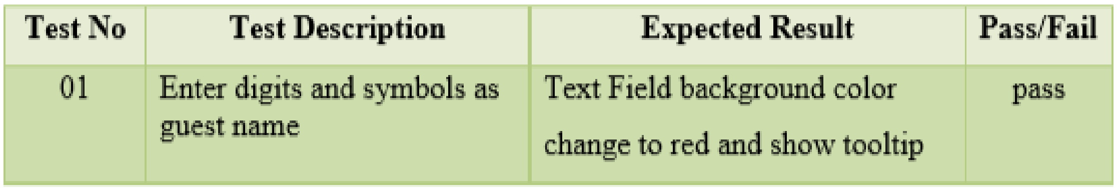
As a rule, system testing tracks, as its input, all of the "integrated" software units that have approved integration testing and also the software system itself integrated with any suitable hardware systems. The purpose of integration testing is to notify any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more restricted type of testing; it seems to notify defects both within the "inter- assemblages" and also within the system as a whole [9].

5.2 TEST CASES

“A test case is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it were originally established for it to do [10]. ”

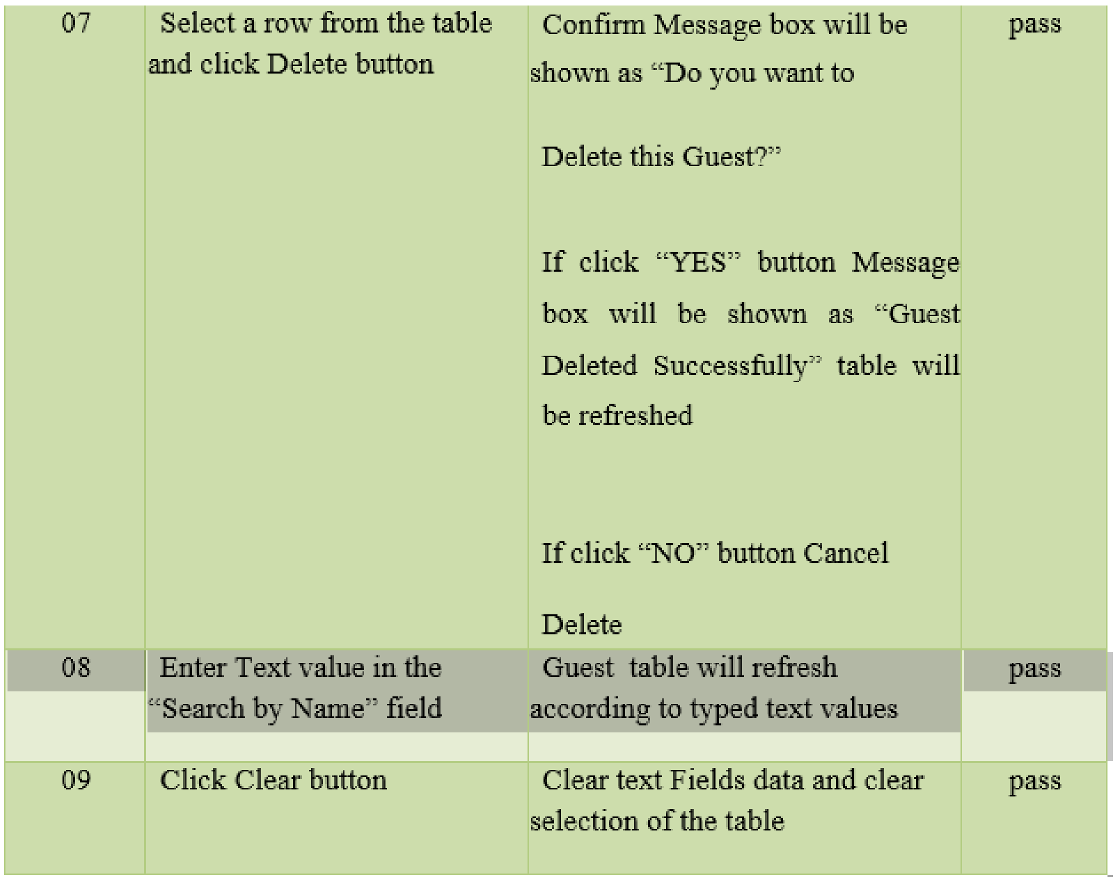
5.2.1 TEST CASES & TEST RESULTS FOR GUEST MANAGEMENT MODULE

Test cases and test results for Guest Management Module are listed in table 5.1 the table has test name, guest type, NIC no ,mobile no , land no, email, address, register date, expected result and the actual result generated by each test case.



Test No	Test Description	Expected Result	Pass/Fail
01	Enter digits and symbols as guest name	Text Field background color change to red and show tooltip	pass
02	Select Combo Box	List all the Guest types to combo box	pass
03	Enter less than 10 characters	Text Field background color change to red Tooltip will be shown as "NIC no is Invalid"	pass

04	Enter all information correctly and click Save	Confirm Message box will be Shown as “Do you want to add this Guest with following details?” If click “YES” button Message box will be shown as “Guest Added Successfully” and added data will be shown below in the table If click “NO” button Cancel Save and Text Fields Values will be Clear	pass
05	Select a row from the table	Selected Guest details will be load into the relevant Text Fields Add button will be disabled and Update, Delete buttons will be enabled	Pass
06	Select a row from the table and change the Building data and click Update button	Confirm Message box will be shown as “Do you want to update this Guest with	pass



07	Select a row from the table and click Delete button	Confirm Message box will be shown as "Do you want to Delete this Guest?" If click "YES" button Message box will be shown as "Guest Deleted Successfully" table will be refreshed If click "NO" button Cancel Delete	pass
08	Enter Text value in the "Search by Name" field	Guest table will refresh according to typed text values	pass
09	Click Clear button	Clear text Fields data and clear selection of the table	pass

Table 5. 1 Test case and results for Guest Management Module

5.2.2 TEST CASES & TEST RESULTS FOR USER MODULE

Test cases and test results for User module are listed in table 5.2 the table has test no, description, expected result and the actual result generated by each test case.

Test No	**Test Description**	**Expected Result**	**Pass/Fail**
01	Select Combo Box	List all the employees to combo	pass

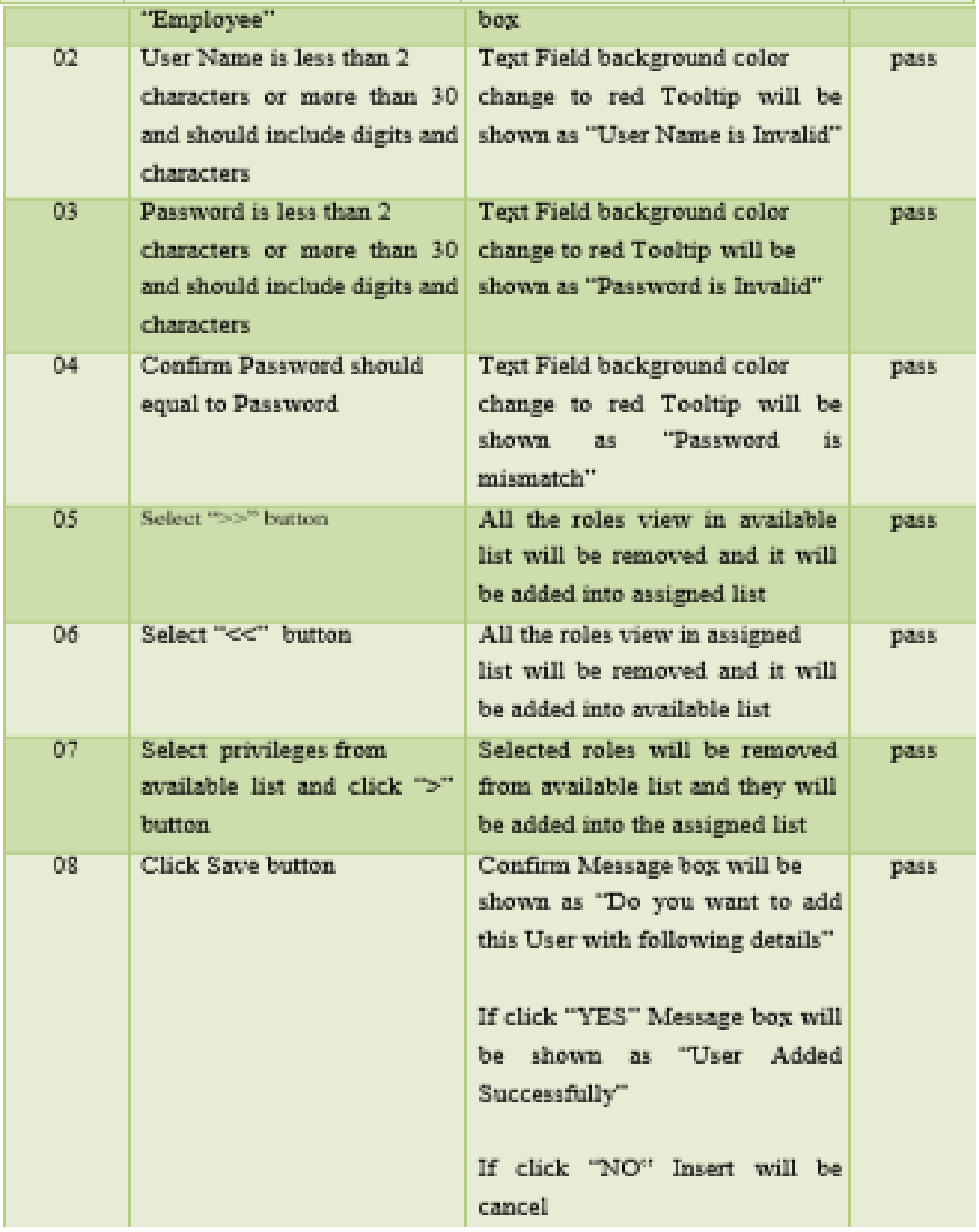
		following details?" If click "YES" button Message box will be shown as "Guest Updated Successfully" and updated data will be shown below in the table If click "NO" button Cancel Update.	
07	Select a row from the table and click Delete button	Confirm Message box will be shown as "Do you want to Delete this Guest?" If click "YES" button Message box will be shown as "Guest Deleted Successfully" table will be refreshed If click "NO" button Cancel Delete	pass
08	Enter Text value in the "Search by Name" field	Guest table will refresh according to typed text values	pass
09	Click Clear button	Clear text Fields data and clear selection of the table	pass

09	Select a row from the table and change the User data and click Update button	Confirm Message box will be shown as "Do you want to update this User with following details?" If click "YES" button Message box will be shown as "User Updated Successfully" and updated data will be shown below in the table If click "NO" button Cancel Update.	pass
10	Select a row from the table and click Delete button	Confirm Message box will be shown as "Do you want to delete this User?" If click "YES" button Message box will be shown as "User Deleted Successfully" table will be refreshed If click "NO" button Cancel Delete	pass

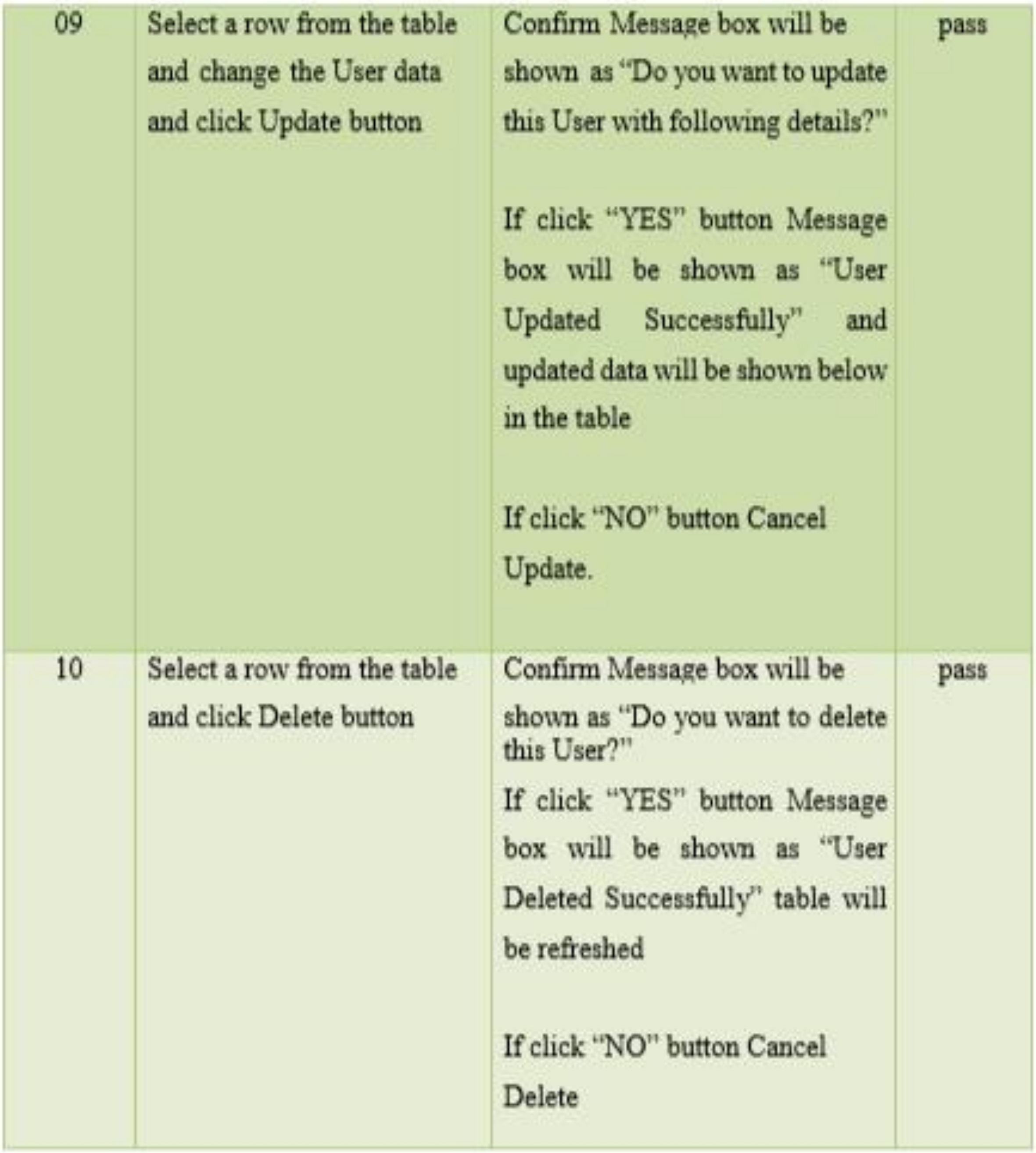
Table 5. 2 Test cases and results for User module

5.2.3 TEST CASES & TEST RESULTS FOR FUNCTION BOOKING MODULE

Test cases and test results for Function Booking Management Module are listed in table 5.3 the table has test Function Type, Function Date, Guest, Inner Table for food menu, services, expected result and the actual result generated by each test case.



Test No	Test Description	Expected Result	Pass/Fail
	"Employee"	box	
02	User Name is less than 2 characters or more than 30 and should include digits and characters	Text Field background color change to red Tooltip will be shown as "User Name is Invalid"	pass
03	Password is less than 2 characters or more than 30 and should include digits and characters	Text Field background color change to red Tooltip will be shown as "Password is Invalid"	pass
04	Confirm Password should equal to Password	Text Field background color change to red Tooltip will be shown as "Password is mismatch"	pass
05	Select ">>" button	All the roles view in available list will be removed and it will be added into assigned list	pass
06	Select "<<" button	All the roles view in assigned list will be removed and it will be added into available list	pass
07	Select privileges from available list and click ">" button	Selected roles will be removed from available list and they will be added into the assigned list	pass
08	Click Save button	Confirm Message box will be shown as "Do you want to add this User with following details" If click "YES" Message box will be shown as "User Added Successfully" If click "NO" Insert will be cancel	pass



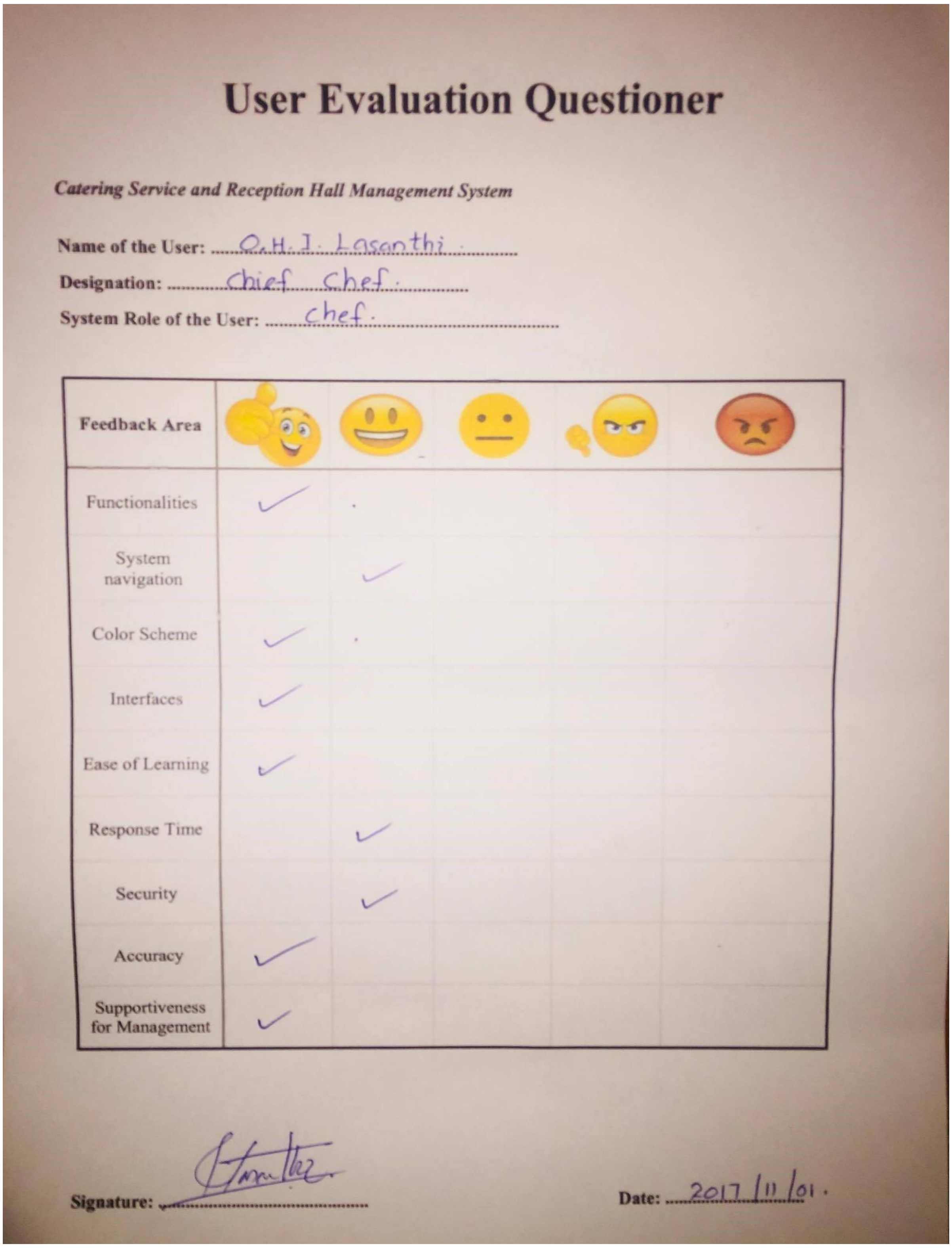
09	Select a row from the table and change the User data and click Update button	Confirm Message box will be shown as "Do you want to update this User with following details?" If click "YES" button Message box will be shown as "User Updated Successfully" and updated data will be shown below in the table If click "NO" button Cancel Update.	pass
10	Select a row from the table and click Delete button	Confirm Message box will be shown as "Do you want to delete this User?" If click "YES" button Message box will be shown as "User Deleted Successfully" table will be refreshed If click "NO" button Cancel Delete	pass

Table 5. 3 Test cases and results for Room type module

5.3 USER EVALUATION

As the definitive product acquired, envisioned Catering Service and Reception Hall Management System for Mayadunne Catering Service was verified by the client. The user acceptance testing was passed out by implementing the system at the actual working environment along with the actual test data & accessible settings in the actual background. For the assessment, ad-hoc system users were chosen determining their privileges they given & they were enquired to accomplish their suitable functionalities. The performance & weaknesses were observed as well as the test results & the user friendliness of the system. Interview was passed out with the real users, about the functionalities and user friendliness of the system, in order to acquire an understanding the level of fulfillment of the users. Over all survey of user comment was done as the final stage.

Figure 5.1 and 5.2 represents the user evaluation results and user feedback results during the user acceptance testing.



User Evaluation Questioner

Catering Service and Reception Hall Management System

Name of the User: O. H. I. Lasanthi .

Designation: Chief Chef .

System Role of the User: chef .

Feedback Area	👍	😃	😐	😒	😠
Functionalities	✓				
System navigation		✓			
Color Scheme	✓				
Interfaces	✓				
Ease of Learning	✓				
Response Time		✓			
Security		✓			
Accuracy	✓				
Supportiveness for Management	✓				

Signature: [illegible]

Date: 2017 / 11 / 01 .

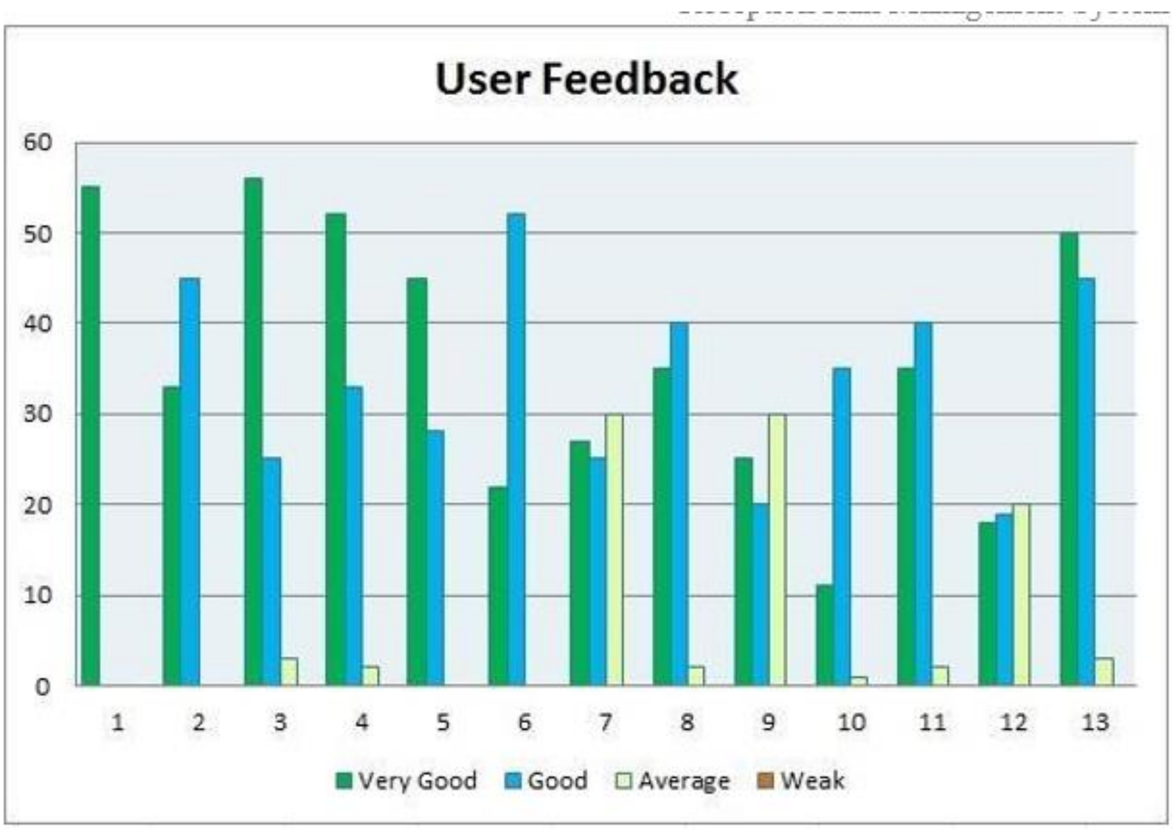


Figure 5. 1 User feedback results

CHAPTER 06 – CONCLUSION

The “Catering Service and Reception Hall Management System for Mayadunne Catering Service” has been eventually stretched to the goal hospitable contests and learning new abilities that accepted master the tasks.

There were key intentions to cover that the client wanted from the established system. One is to increase the adeptness of the business process. To enhance the efficiency of the business, it is essential to get advanced adeptness from the employees. As I advanced a PBX system, presented call accounting system has been highlighted. It facilitated the PBX activities & ensures the reliability of updates in the system just in time.

The system track Bookings wanted updates & create statements of status changes. Other than that it accomplished guest and function that request to the booking process allowing to the booking levels. While dealing with the guests system was helped to interlink with them over a mailing system.

And also the report producing progression has been helped to the administration for better determination making to develop new business approaches. Security improvements of the new system also grew the high consideration of the client side.

The procedural & the logical expansion have reasoned to large functional & nonfunctional improvements of the office environment as well as the performance of the employees by getting easier to the distinct tasks & motivating them. The whole procedure of the business organization also involved absolutely with improving productivity.

That was exposed by the feedback of client’s as well as the employee’s & the new performance of the business environment.

6.1 LESSONS LEARNT

It was an opportunity to get extensive knowledge on JavaFX, XML combined with the pre-gained knowledge in MVC, Hibernate, SQL management & Netbeans. And vast ranges of technologies also have been used to make this high quality product. From the primary stage of the system development life cycle there were more skills to develop like investigating facts in feasibility studies, effective communicating with people, time management, project planning, report writing, training people to new system environment & many valuable capabilities.

6.2 FUTUER IMPROVEMENTS

In future following features are planned to add to the newly built system as further improvements.

Expand the Catering Service and Reception Hall Management System.

Web based Catering Service and Reception Hall Management System will be introduced for ease of connecting branches.

- Allow online guests to book functions that they need through the system and deliver relevant information.
- Allow online guests to do payments online.
- System will be expanded in the future to input attendance and payment records using external reading devices making the system fully automated.

Introduced system has been developed with the possibility of any expansions such as adding more functions to the system much easily.

REFERENCES

[1] Wikipedia, "Systems design", 2016. [Online]. Available: https://en.wikipedia.org/wiki/Systems_design.
[Accessed: 16- Jun- 2017].

[2] Wikipedia, "Database design", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/Database_design.
[Accessed: 16- Jun- 2017].

[3] M. Chapple, "The Basics of Normalizing a Database", About.com Tech, 2016. [Online]. Available:
http://databases.about.com/od/specificproducts/a/normalization.htm.
[Accessed: 16- Jun- 2017].

[4] Wikipedia, "MySQL", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/MySQL
[Accessed: 16- Jun- 2017].

[5] Wikipedia, "Visual Paradigm for UML", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/Visual_Paradigm_for_UML
[Accessed: 16- Jun- 2017].

[6] Wikipedia, "Hibernate (framework)", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/Hibernate
[Accessed: 16- Jun- 2017].

[7] Wikipedia, "Software testing", 2016. [Online]. Available:
http://en.wikipedia.org/wiki/Software_testing.
[Accessed: 20- Jul- 2017].

[8] Wikipedia, "Software verification and validation", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/Software_verification_and_validation.
[Accessed: 20- Jul- 2017].

[9] Wikipedia, "System testing", 2016. [Online]. Available:
http://en.wikipedia.org/wiki/System_testing.
[Accessed: 20- Jul- 2017].

[10] Wikipedia, "Test case", 2016. [Online]. Available:

http://en.wikipedia.org/wiki/Test_case
[Accessed: 20- Jul- 2017].

[11] Wikipedia, "Java (programming language)", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/Java_(programming_language).
[Accessed: 20- Jul- 2017].

[12] V. Patel, "eZee PMS Hotel Software system for hotel management, hotels reservation software", Ezeefrontdesk.com, 2016. [Online]. Available:

http://www.ezeefrontdesk.com/.

[Accessed: 02- Aug- 2017].

[13] A. Systems, "Hotel Management Software, Hotel PMS | Anand Systems Inc.", Anandsystems.com, 2016. [Online]. Available:
https://www.anandsystems.com/.
[Accessed: 02- Aug- 2017].

[14] Wikipedia, "Java Persistence API", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/Java_Persistence_API.
[Accessed: 02- Aug- 2017].

[15] Wikipedia, "NetBeans", 2016. [Online]. Available:
https://en.wikipedia.org/wiki/NetBeans.
[Accessed: 02- Aug- 2017].

APPENDIX

APPENDIX-A

SYSTEM DOCUMENTATION

This segment includes significance information for administrators, users and anyone who desires to carry on this project. This includes information and steps about installation, configuration.

The hardware and software environment which is needed to implement the system in

Client's site is listed in table A.1

Hardware Environment	**Software Environment**
Intel(R) Core(TM) i5 - 2410M @ CPU 2.30 GHz	MySQL Server 5.5
4GB RAM	MySQL Workbench
160GB Hard Disk	MySQL Query Browser NetBeans IDE JavaFX Scene Builder Visual Paradigm

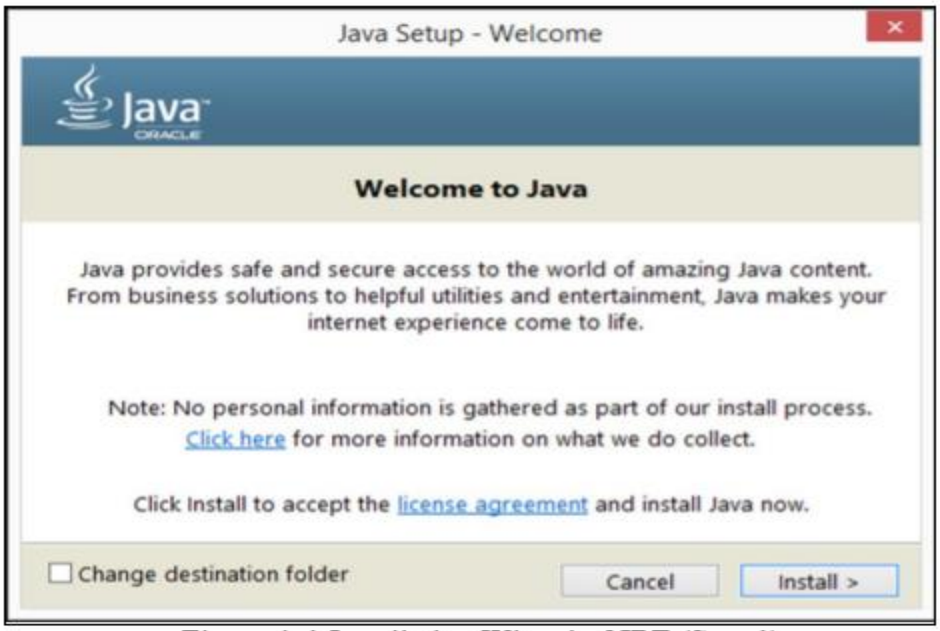
Table A. 1 Implementation Environment

A.1 HOW TO SETUP

A.1.1 INSTALL JAVA RUN TIME ON CLIENT MACHINE

Following are some significant screenshots of installing JAVA runtime on client machine. In here the installation order is showed as step by step.

Below figure A.1 shows welcome screen in the installation wizard of JRE as (Step 1)



Java Setup - Welcome

Welcome to Java

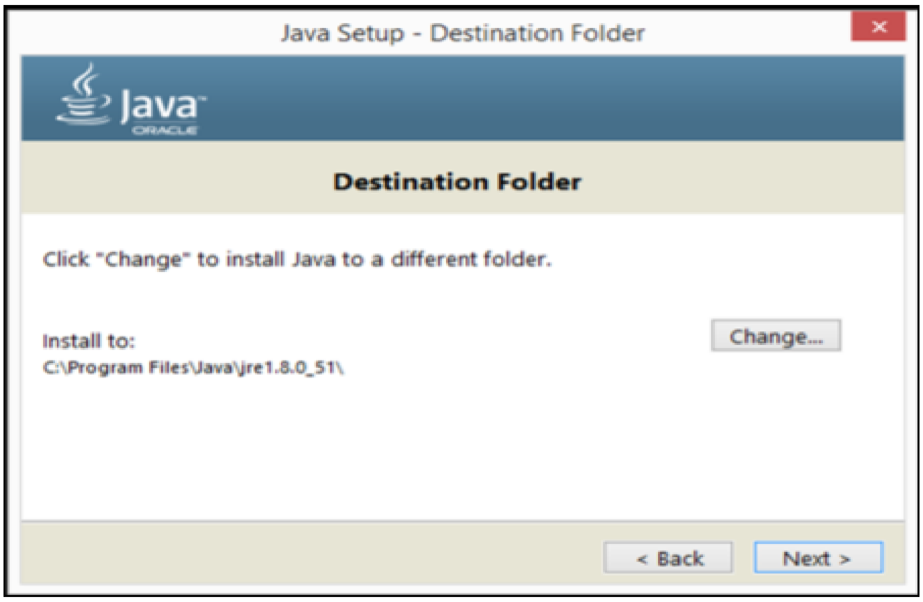
Java provides safe and secure access to the world of amazing Java content. From business solutions to helpful utilities and entertainment, Java makes your internet experience come to life.

Note: No personal information is gathered as part of our install process. Click here for more information on what we do collect.

Click Install to accept the license agreement and install Java now.

Figure A. 1 Installation Wizard of JRE (Step 1)

Below figure A.2 shows destination folder changing screen in the installation wizard of JRE as (Step 2)



Java Setup - Destination Folder

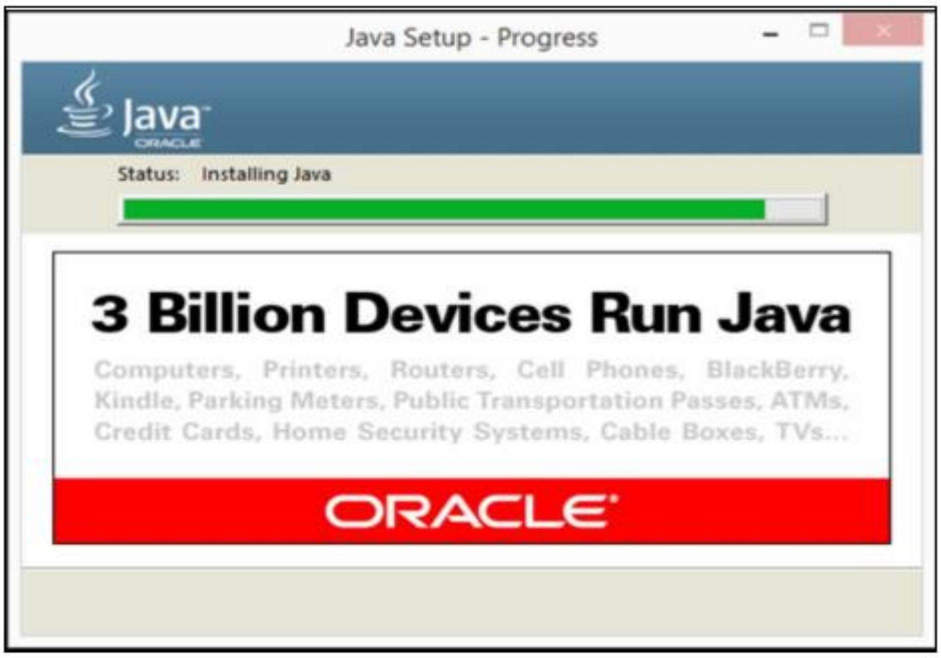
Destination Folder

Click "Change" to install Java to a different folder.

Install to:
C:\Program Files\Java\jre1.8.0_51\

Figure A. 2 Installation Wizard of JRE (Step 2)

Below figure A.3 shows installation progress in the installation wizard of JRE as (Step 3)



Java Setup - Progress

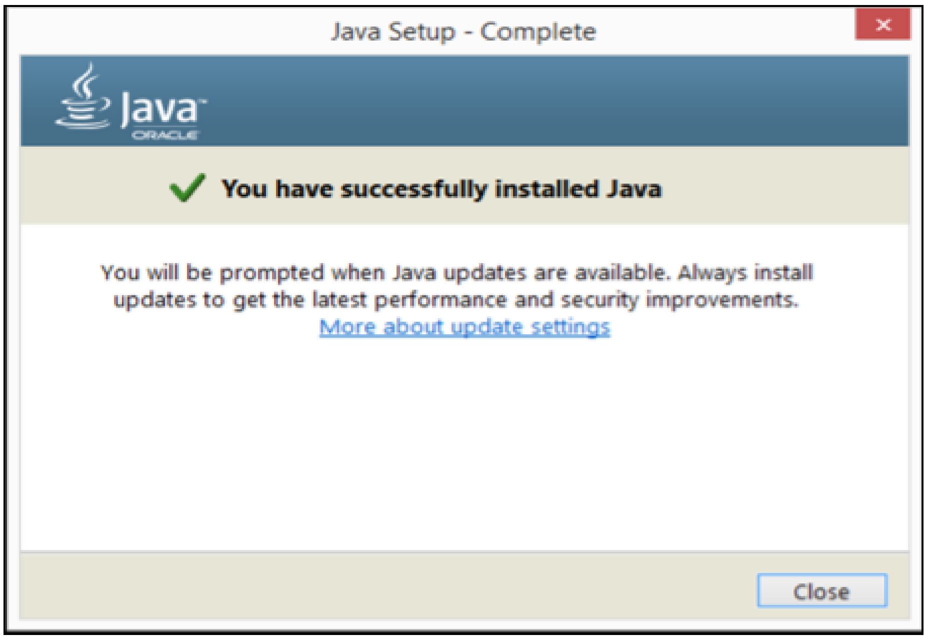
Status: Installing Java

3 Billion Devices Run Java

Computers, Printers, Routers, Cell Phones, BlackBerry, Kindle, Parking Meters, Public Transportation Passes, ATMs, Credit Cards, Home Security Systems, Cable Boxes, TVs...

Figure A. 3 Installation Wizard of JRE (Step 3)

Below figure A.4 shows completion screen in the installation wizard of JRE as (Step 4)



Java Setup - Complete

You have successfully installed Java

You will be prompted when Java updates are available. Always install updates to get the latest performance and security improvements.
More about update settings

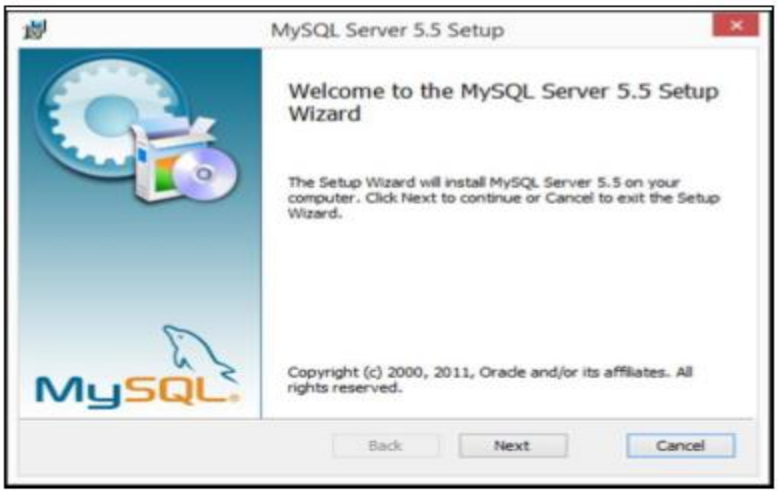
Figure A. 4 Installation Wizard of JRE (Step 4)

A.1.2 INSTALLING MYSQL SERVER 5.5

Step 1: Download MySQL Community Server 5.5 installation file appropriate for the platform. Open installation file for MySQL Community Server and press “Next”.

Step 2: Choose “Typical” setup type, click “Next” and “Install”.

Bellow figure A.5 shows the welcome screen in the installation wizard of MySQL Server as (Step 1)


MySQL Server 5.5 Setup

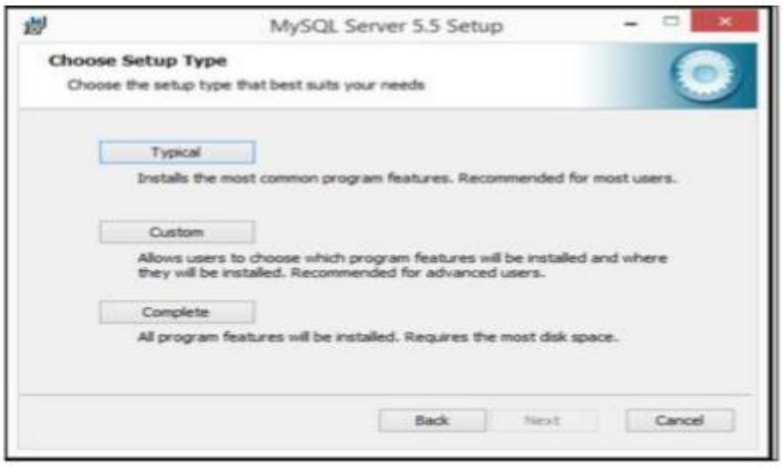
Welcome to the MySQL Server 5.5 Setup Wizard

The Setup Wizard will install MySQL Server 5.5 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Figure A. 5 Installation Wizard of MySQL Server (Step 1)

Bellow figure A.6 shows the setup type screen in the installation wizard of MySQL Server as (Step 2)


MySQL Server 5.5 Setup

Choose Setup Type

Choose the setup type that best suits your needs

Typical

Installs the most common program features. Recommended for most users.

Custom

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

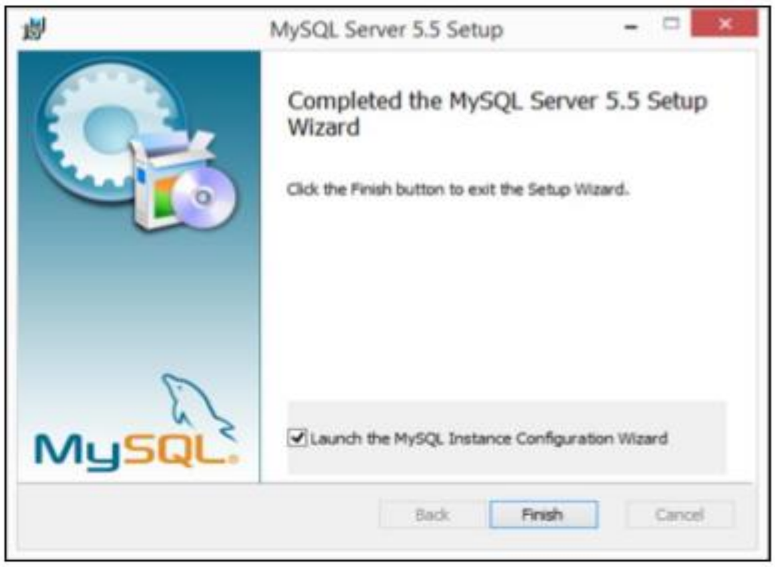
Complete

All program features will be installed. Requires the most disk space.

Figure A. 6 Installation Wizard of MySQL Server (Step 2)

Step 3: After installation process is completed, check "Launch the MySQL Instance Configuration Wizard" and click "Finish".

Step 4: In "Configuration Wizard" click "Next".


MySQL Server 5.5 Setup

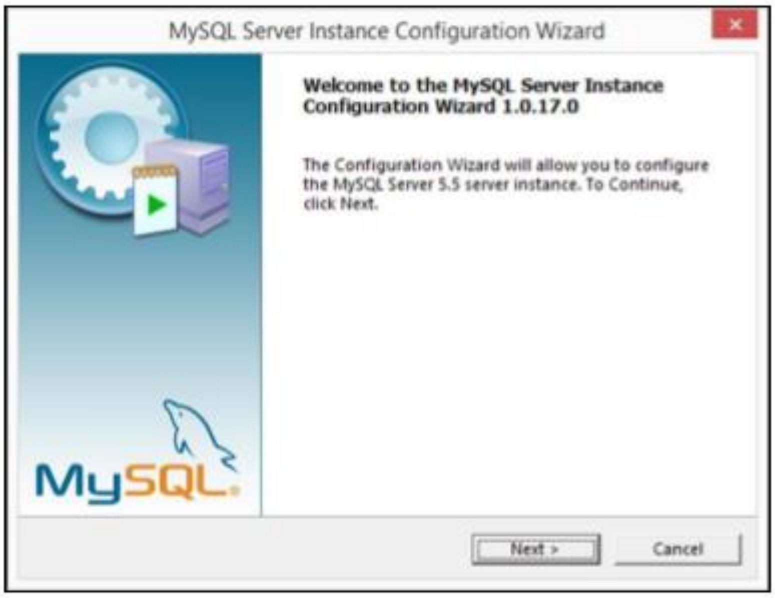
Completed the MySQL Server 5.5 Setup Wizard

Click the Finish button to exit the Setup Wizard.

☑ Launch the MySQL Instance Configuration Wizard

Figure A. 7 Installation Wizard of MySQL Server (Step 3)

Following figure A.8 shows welcome screen in the installation wizard of MySQL Server Instance Configuration as (Step 4)


MySQL Server Instance Configuration Wizard

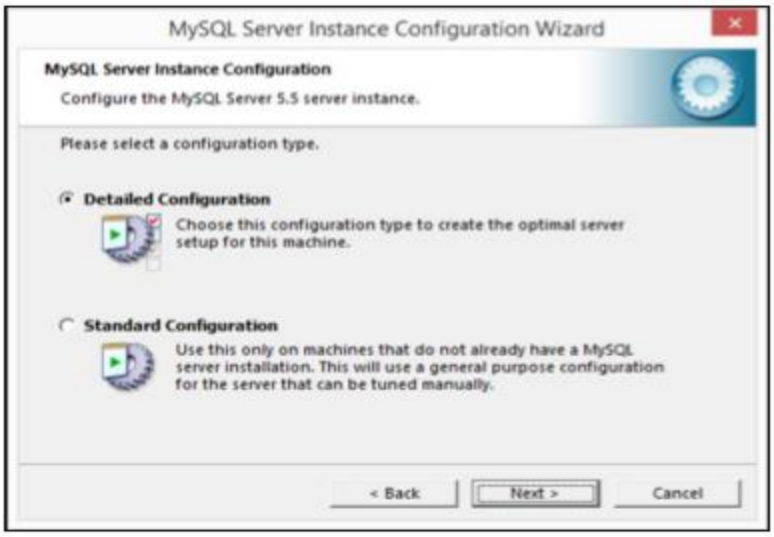
Welcome to the MySQL Server Instance Configuration Wizard 1.0.17.0

The Configuration Wizard will allow you to configure the MySQL Server 5.5 server instance. To Continue, click Next.

Figure A. 8 Installation Wizard of MySQL Server (Step 4)

Step 5: Choose "Detailed Configuration" and click "Next".

Step 6: Check "Install as Windows Service", select service name "MySQL". Check "Launch the MySQL Server automatically" (this feature will run service automatically after installation), check to "Include Bin Directory in Windows PATH" and click "Next". Following figure A.9 shows configuration type screen in the installation wizard of MySQL Server Instance Configuration as (Step 5)



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

Please select a configuration type.

(●) Detailed Configuration

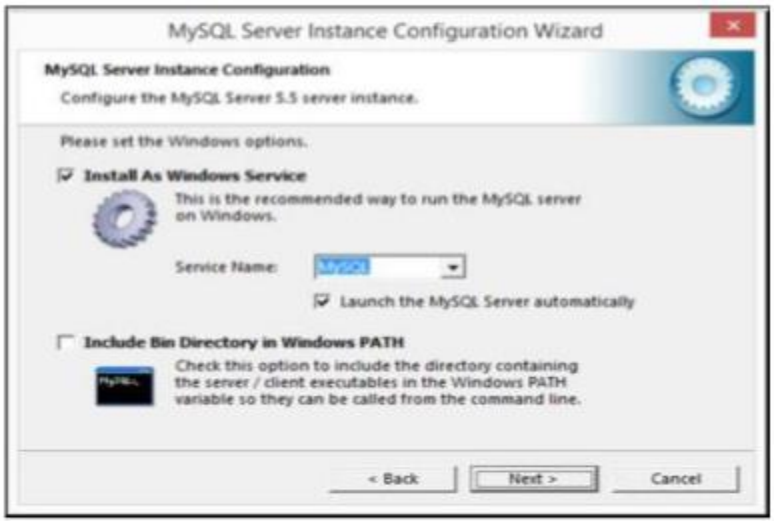
Choose this configuration type to create the optimal server setup for this machine.

( ) Standard Configuration

Use this only on machines that do not already have a MySQL server installation. This will use a general purpose configuration for the server that can be tuned manually.

Figure A. 9 Installation Wizard of MySQL Server (Step 5)

Following figure A.10 shows windows options screen in the installation wizard of MySQL Server Instance Configuration as (Step 6)



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

Please set the Windows options.

☑ Install As Windows Service

This is the recommended way to run the MySQL server on Windows.

Service Name: MySQL

☑ Launch the MySQL Server automatically

☐ Include Bin Directory in Windows PATH

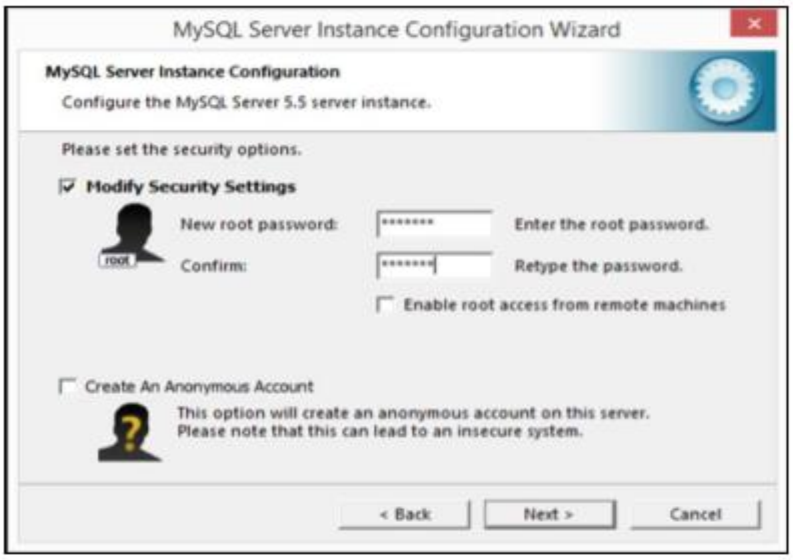
Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line.

Figure A. 10 Installation Progress MySQL Server (Step 6)

Step 7: Set a long password for the “root” user, check “Enable root access from remote machines” and do not create an Anonymous account. Click “Next” and then “Execute”. Step 8: After configuration process is completed click “Finish”.

Following figure A.11 shows security options screen in the installation wizard of

MySQL Server Instance Configuration as (Step 7)



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

Please set the security options.

☑ **Modify Security Settings**

New root password:	*******	Enter the root password.
Confirm:	*******	Retype the password.

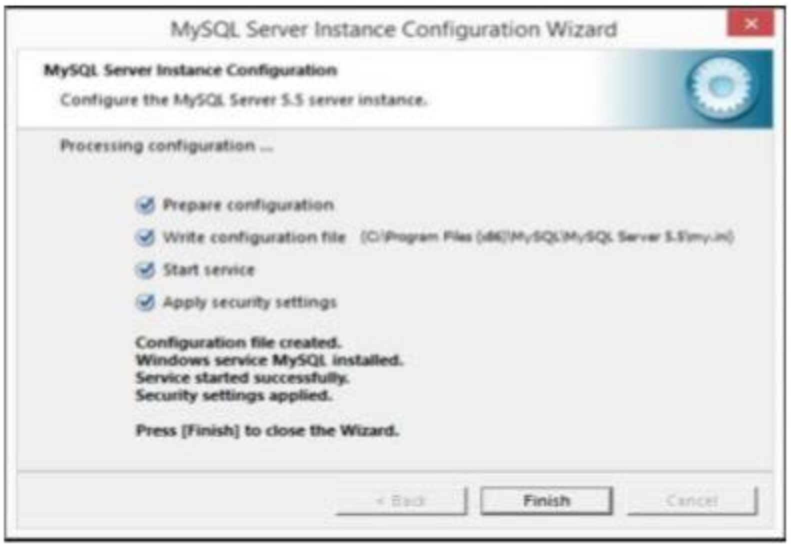
☐ Enable root access from remote machines

☐ Create An Anonymous Account

This option will create an anonymous account on this server.
Please note that this can lead to an insecure system.

Figure A. 11 Installation Wizard of MySQL Server (Step 7)

Following figure A.12 shows processing configuration screen in the installation wizard of MySQL Server Instance Configuration as (Step 8)



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.5 server instance.

Processing configuration ...

✓ Prepare configuration
✓ Write configuration file (C:\Program Files (x86)\MySQL\MySQL Server 5.5\my.ini)
✓ Start service
✓ Apply security settings

Configuration file created.
Windows service MySQL installed.
Service started successfully.
Security settings applied.

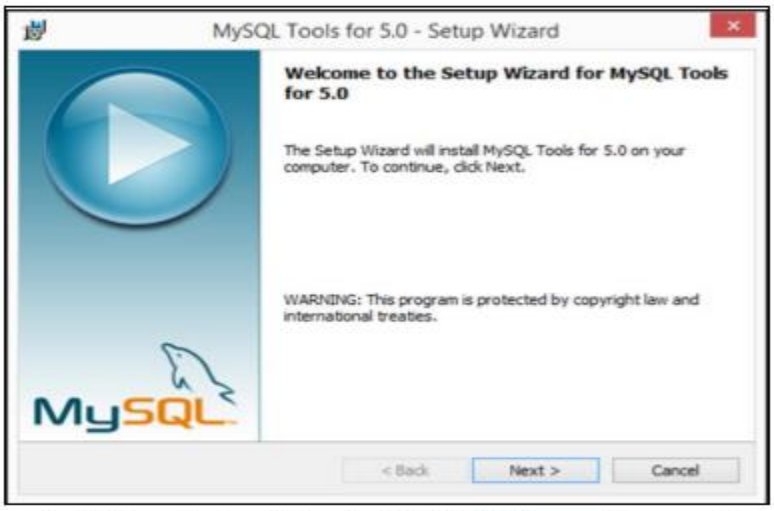
Press [Finish] to close the Wizard.

Figure A. 12 Installation Wizard of MySQL Server (Step 8)

A.1.3 INSTALLING MYSQL QUERY BROWSER

Following are some significant screenshots of installing MySQL Query Browser on client machine. It is a GUI based query manipulation application. In here the installation order is showed as step by step.

Below figure A.13 shows welcome screen in the installation wizard of MySQL Query Browser as (Step 1)


MySQL Tools for 5.0 - Setup Wizard

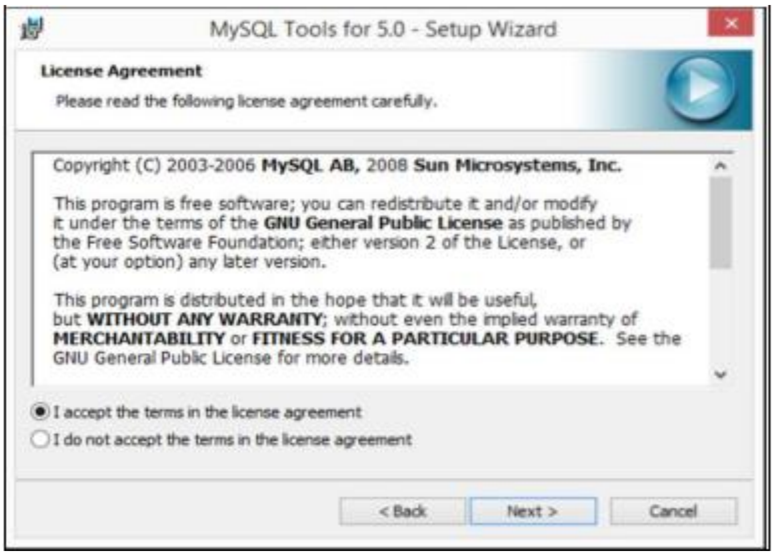
Welcome to the Setup Wizard for MySQL Tools for 5.0

The Setup Wizard will install MySQL Tools for 5.0 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Figure A. 13 Installation Wizard of MySQL Query Browser (Step 1)

Below figure A.14 shows license agreement screen in the installation wizard of MySQL Query Browser as (Step 2)


MySQL Tools for 5.0 - Setup Wizard

License Agreement

Please read the following license agreement carefully.

Copyright (C) 2003-2006 **MySQL AB, 2008 Sun Microsystems, Inc.**

This program is free software; you can redistribute it and/or modify it under the terms of the **GNU General Public License** as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

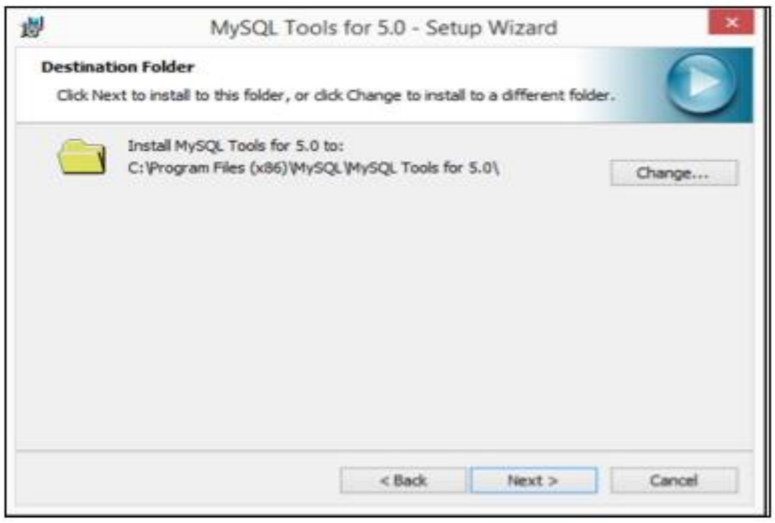
This program is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU General Public License for more details.

[✓ I accept the terms in the license agreement]

I do not accept the terms in the license agreement

Figure A. 14 Installation Wizard of MySQL Query Browser (Step 2)

Below figure A.15 shows destination folder changing screen in the installation wizard of MySQL Query Browser as (Step 4)



MySQL Tools for 5.0 - Setup Wizard

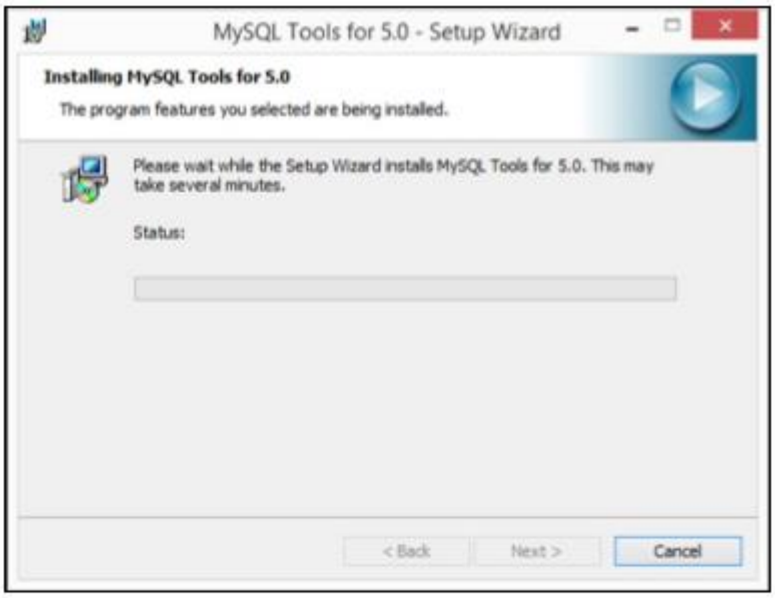
Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install MySQL Tools for 5.0 to:
C:\Program Files (x86)\MySQL\MySQL Tools for 5.0\

Figure A. 15 Installation Wizard of MySQL Query Browser (Step 3)

Below figure A.16 shows installation progress in the installation wizard of MySQL Query Browser as (Step 5)



MySQL Tools for 5.0 - Setup Wizard

Installing MySQL Tools for 5.0

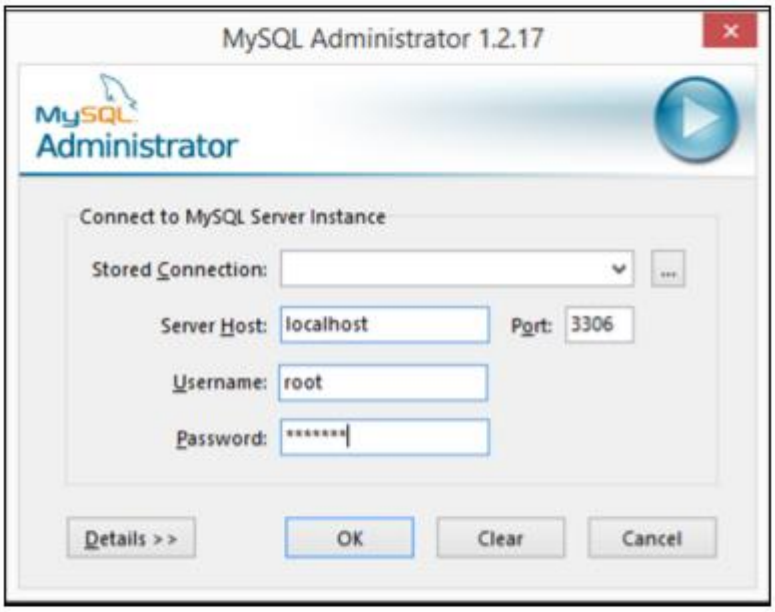
The program features you selected are being installed.

Please wait while the Setup Wizard installs MySQL Tools for 5.0. This may take several minutes.

Status:

Figure A. 16 Installation Wizard of MySQL Query Browser (Step 5)

Below figure A.17 shows login screen in the MySQL Query Browser as (Step 4). In here relevant host, username and password must provide to log to the MySQL Server.



Connect to MySQL Server Instance

Stored Connection:

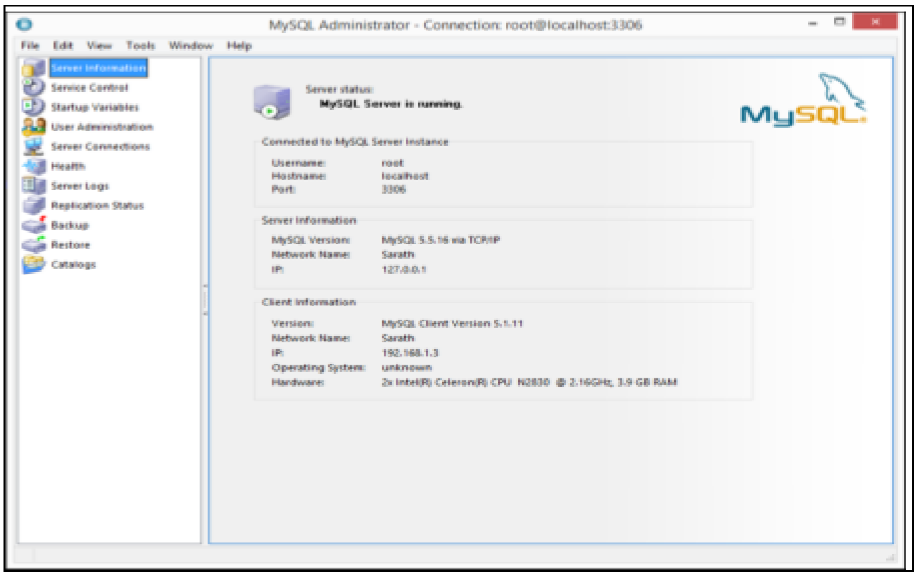
Server Host: localhost Port: 3306

Username: root

Password: *******

Figure A. 17 Figure A.17 Login Screen of MySQL Query Browser (Step 5)

Below figure A.18 shows server information screen in the MySQL Query Browser as (Step 7)



```
Server status:
  MySQL Server is running.

Connected to MySQL Server Instance
  Username:          root
  Hostname:          localhost
  Port:              3306

Server Information
  MySQL Version:     MySQL 5.5.16 via TCP/IP
  Network Name:      Sarath
  IP:                127.0.0.1

Client Information
  Version:           MySQL Client Version 5.1.11
  Network Name:      Sarath
  IP:                192.168.1.3
  Operating System:  unknown
  Hardware:          2x Intel(R) Celeron(R) CPU  N2830  @ 2.16GHz, 3.9 GB RAM
```

Figure A. 18 Server Information Screen of MySQL Query Browser (Step 6)

A.1.4 INSTALLING CATERING SERVICE AND RECEPTION HALL MANAGEMENT SYSTEM

After setting up the database,

1. Select relevant device.
2. Click Add &.
3. Browse the Hotel Management System file from CD.
4. Run the setup.exe file located in CD.

APPENDIX-B

B.1 ACTIVITY DIAGRAM

Following figure B.1 represents the activity diagram for user login. For successful login, user has to complete defined activities.

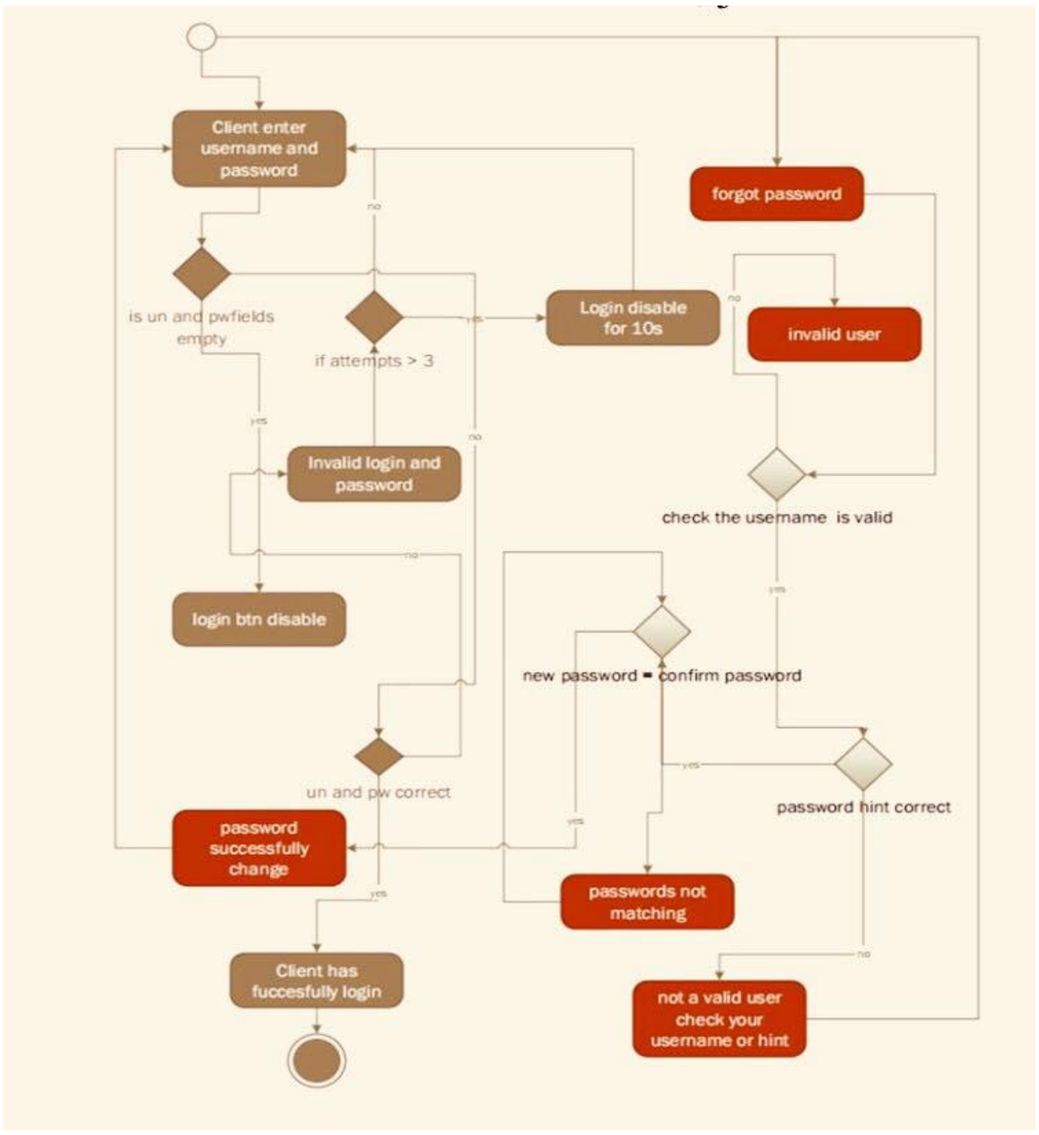

Client enter username and password
forgot password
is un and pwfields empty
if attempts > 3
Login disable for 10s
invalid user
Invalid login and password
check the username is valid
login btn disable
new password = confirm password
un and pw correct
password hint correct
password successfully change
passwords not matching
Client has fuccesfully login
not a valid user check your username or hint

Figure B. 1 Activity diagram for user login

B.2 USE CASE DIAGRAM

Following figure B.2 shows use cases relevant to Administrator. To perform such use cases first of all Administrator has to login to the system. Manage employee details, manage user privileges, manage room details are some use cases relevant to Administrator. Use case description for Administrator is listed in table B.1, B.2, B.3

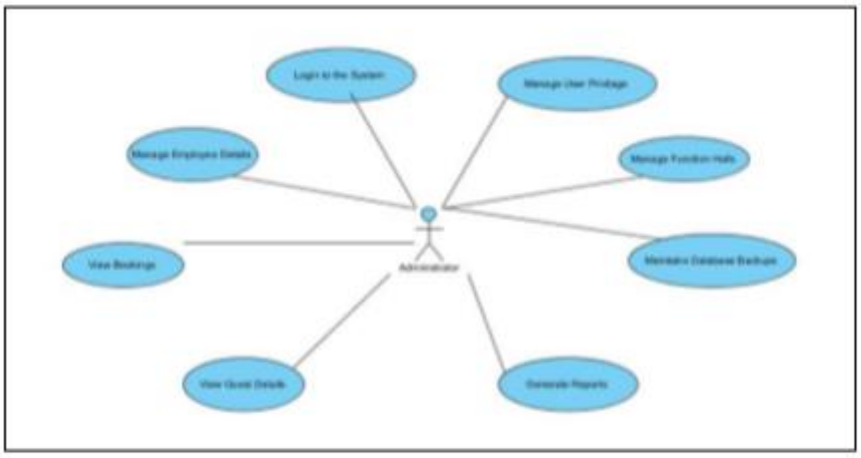
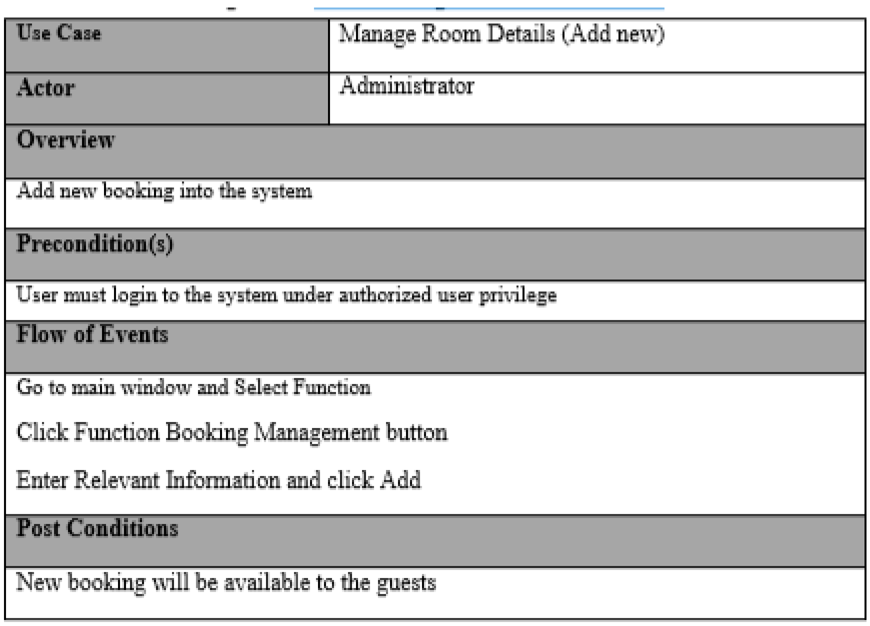
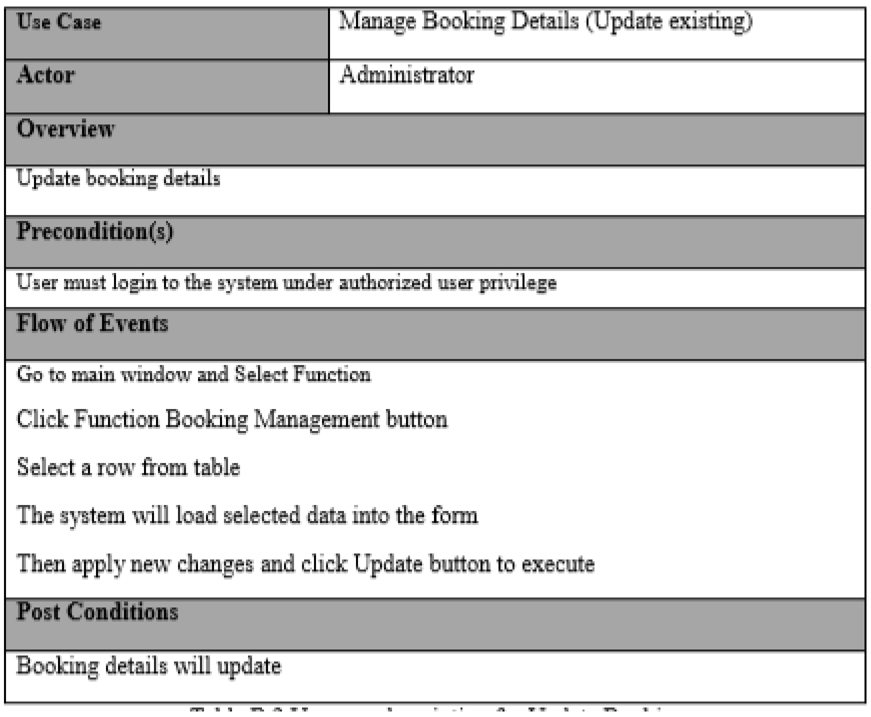


Figure B. 2 Use case diagram for Administrator



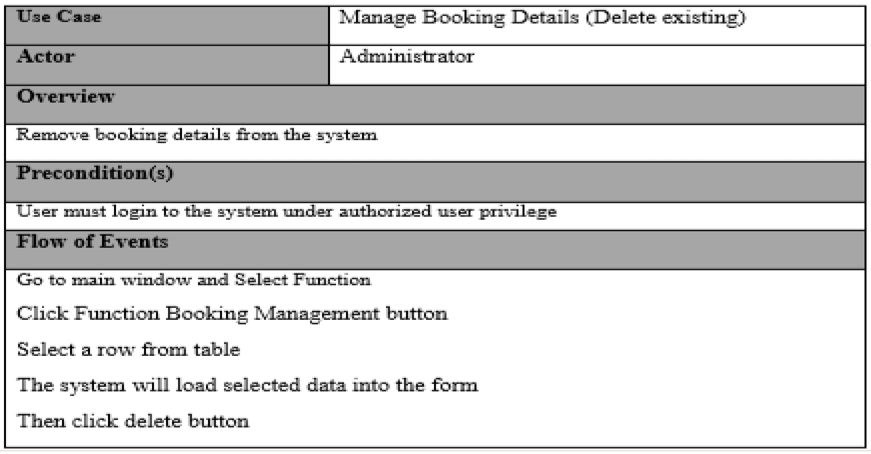
Use Case	Manage Room Details (Add new)
Actor	Administrator
Overview	
Add new booking into the system	
Precondition(s)	
User must login to the system under authorized user privilege	
Flow of Events	
Go to main window and Select Function Click Function Booking Management button Enter Relevant Information and click Add	
Post Conditions	
New booking will be available to the guests	

Table B. 1 Use case description for Add Booking



Use Case	Manage Booking Details (Update existing)
Actor	Administrator
Overview	
Update booking details	
Precondition(s)	
User must login to the system under authorized user privilege	
Flow of Events	
Go to main window and Select Function Click Function Booking Management button Select a row from table The system will load selected data into the form Then apply new changes and click Update button to execute	
Post Conditions	
Booking details will update	

Table B. 2 Use case description for Update Booking



Use Case	Manage Booking Details (Delete existing)
Actor	Administrator
Overview	
Remove booking details from the system	
Precondition(s)	
User must login to the system under authorized user privilege	
Flow of Events	
Go to main window and Select Function Click Function Booking Management button Select a row from table The system will load selected data into the form Then click delete button	

Table B. 3 Use case description for Delete Booking

Following figure B.3 shows use cases relevant to Executive Chef. To perform such use cases first of all Executive Chef has to login to the system. Make Menu form report by Booking, manage Kitchen Items, Create Kitchen Item Request Form are some use cases relevant to Executive Chef. And use case description for Chef is listed in table B.4

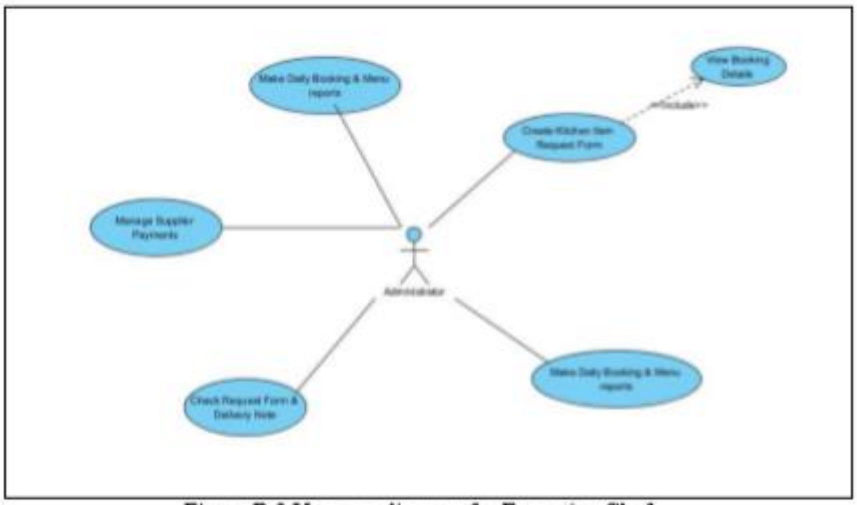
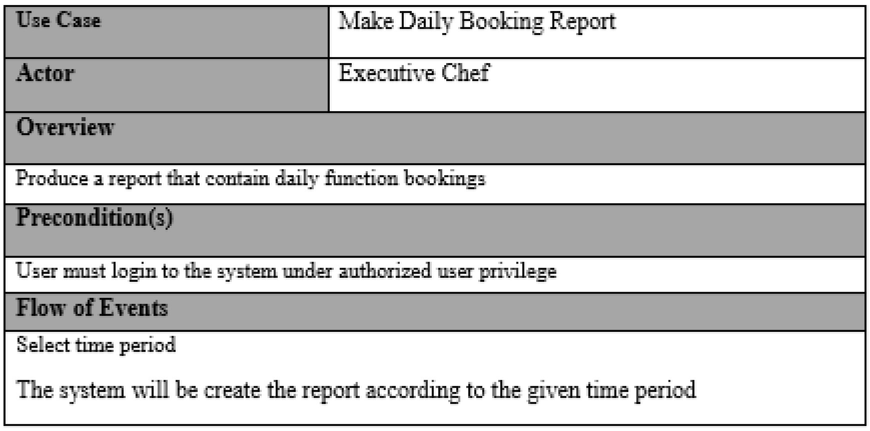

Make Daily Booking & Menu reports
View Booking Details
<<Include>>
Create Kitchen Item Request Form
Manage Supplier Payments
Administrator
Check Request Form & Delivery Note
Make Daily Booking & Menu reports

Figure B. 3 Use case diagram for Executive Chef

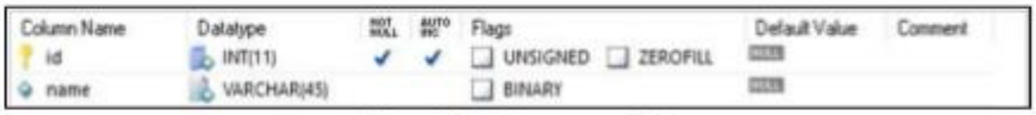


Use Case	Make Daily Booking Report
Actor	Executive Chef
Overview	
Produce a report that contain daily function bookings	
Precondition(s)	
User must login to the system under authorized user privilege	
Flow of Events	
Select time period The system will be create the report according to the given time period	

Table B. 4 Use case description for Make Day End Report

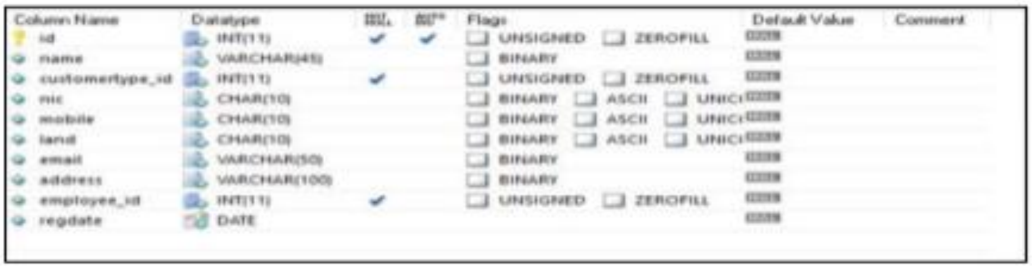
B.3 DATABASE TABLE DESIGN

Figure B.8-B.33 show the table design in the database of the system.



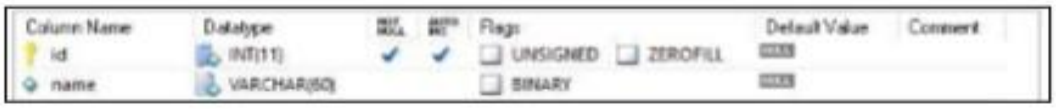
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	[illegible]	
name	VARCHAR(45)			☐ BINARY	[illegible]	

Figure B. 4 Guest type Table



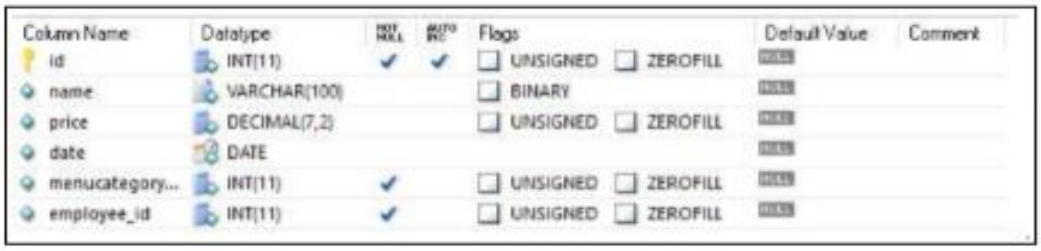
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	[illegible]	
name	VARCHAR(45)			☐ BINARY	[illegible]	
customertype_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	[illegible]	
nic	CHAR(10)			☐ BINARY ☐ ASCII ☐ UNICI	[illegible]	
mobile	CHAR(10)			☐ BINARY ☐ ASCII ☐ UNICI	[illegible]	
land	CHAR(10)			☐ BINARY ☐ ASCII ☐ UNICI	[illegible]	
email	VARCHAR(50)			☐ BINARY	[illegible]	
address	VARCHAR(100)			☐ BINARY	[illegible]	
employee_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	[illegible]	
regdate	DATE				[illegible]	

Figure B. 5 Guest Table



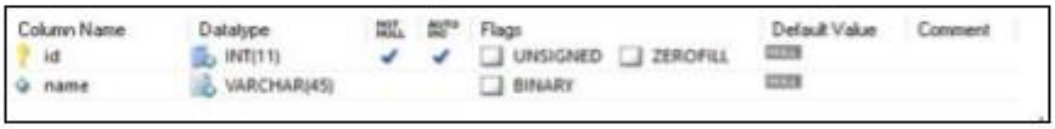
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	[illegible]	
name	VARCHAR(60)			☐ BINARY	[illegible]	

Figure B. 6 Menu Category Table



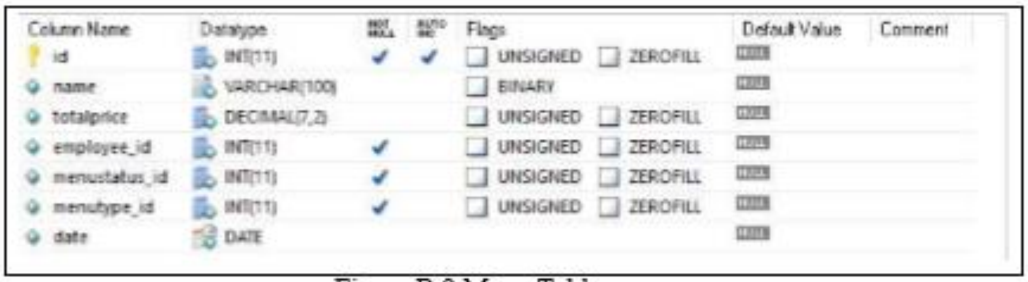
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
name	VARCHAR(100)			☐ BINARY	NULL	
price	DECIMAL(7,2)			☐ UNSIGNED ☐ ZEROFILL	NULL	
date	DATE				NULL	
menucategory...	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
employee_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	

Figure B. 7 Menu Item Table



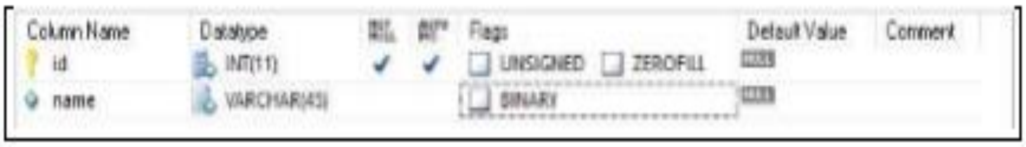
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
name	VARCHAR(45)			☐ BINARY	NULL	

Figure B. 8 Menu Type Table



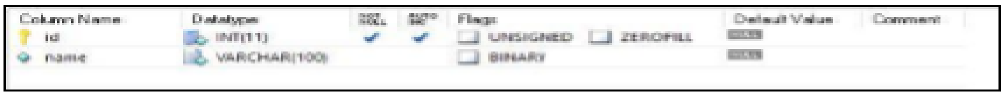
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
name	VARCHAR(100)			☐ BINARY	NULL	
totalprice	DECIMAL(7,2)			☐ UNSIGNED ☐ ZEROFILL	NULL	
employee_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
menustatus_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
menutype_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
date	DATE				NULL	

Figure B. 9 Menu Table



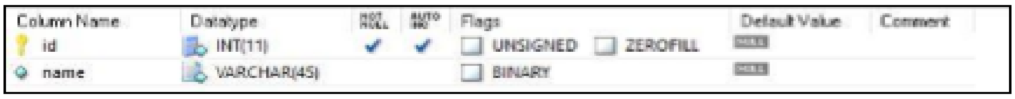
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
name	VARCHAR(45)			☐ BINARY	NULL	

Figure B. 10 Menu Status



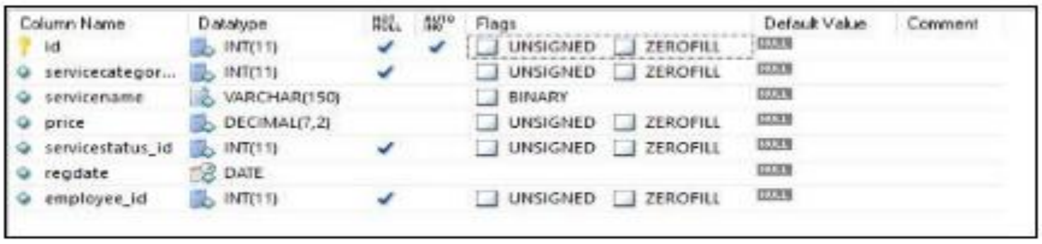
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
name	VARCHAR(100)			☐ BINARY	NULL	

Figure B. 11 Service Category



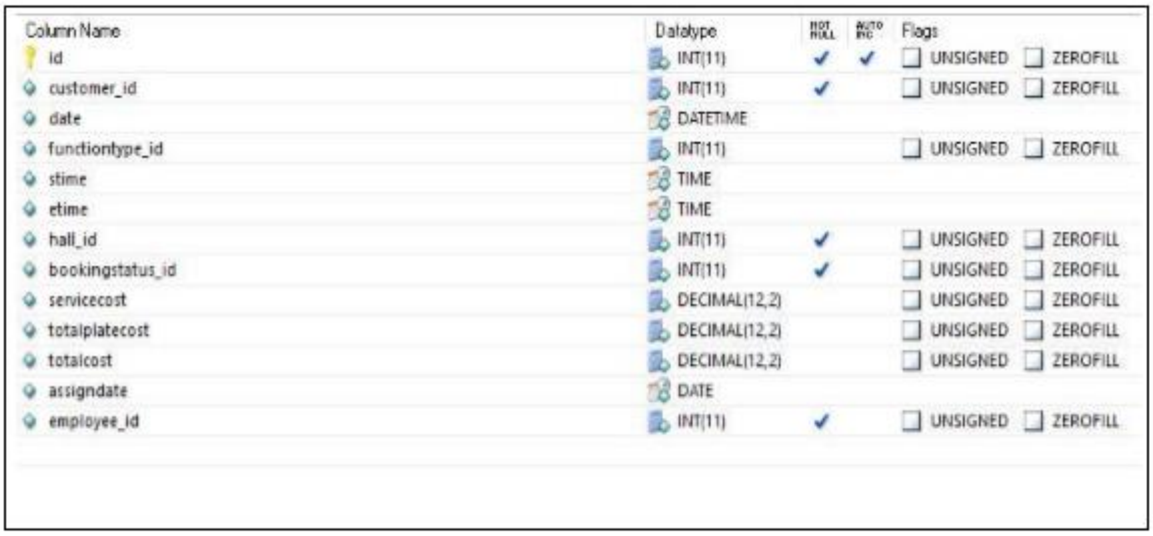
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
name	VARCHAR(45)			☐ BINARY	NULL	

Figure B. 12 Service Status



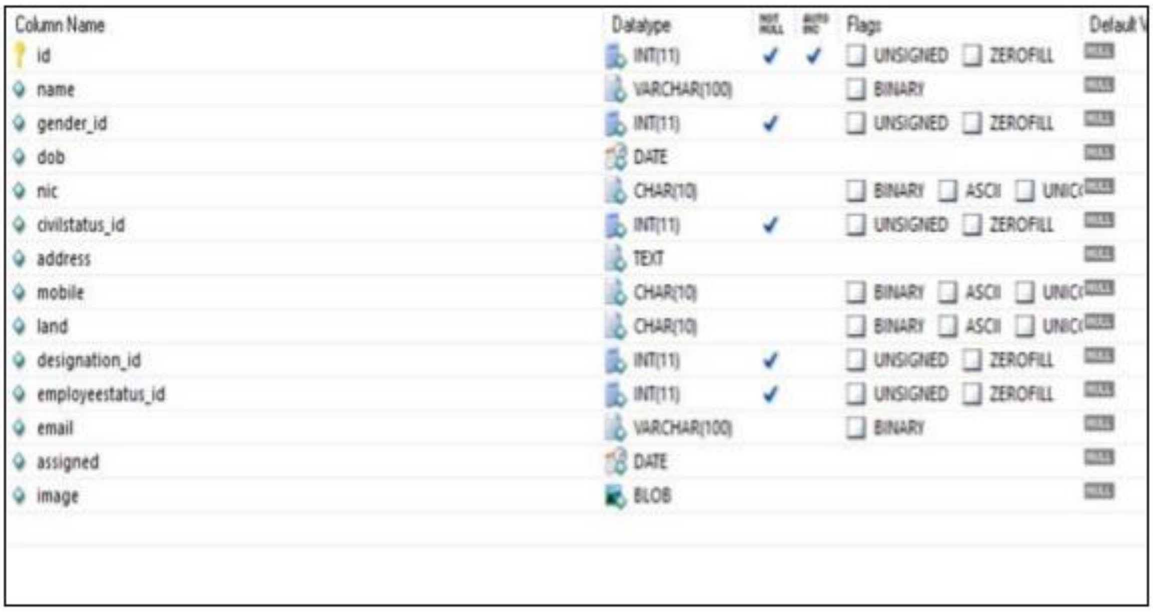
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
servicecategor...	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
servicename	VARCHAR(150)			☐ BINARY	NULL	
price	DECIMAL(7,2)			☐ UNSIGNED ☐ ZEROFILL	NULL	
servicestatus_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
regdate	DATE				NULL	
employee_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	

Figure B. 13 Service Tables



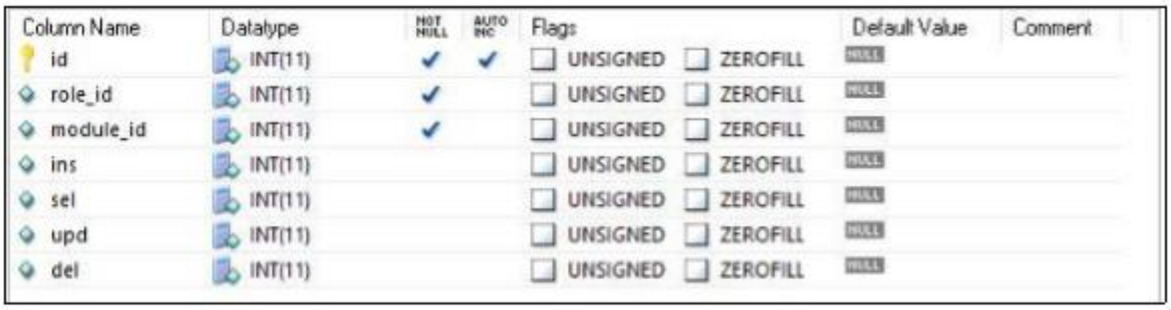
Column Name	Datatype	NOT NULL	AUTO INC	Flags
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL
customer_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL
date	DATETIME			
functiontype_id	INT(11)			☐ UNSIGNED ☐ ZEROFILL
stime	TIME			
etime	TIME			
hall_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL
bookingstatus_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL
servicecost	DECIMAL(12,2)			☐ UNSIGNED ☐ ZEROFILL
totalplatecost	DECIMAL(12,2)			☐ UNSIGNED ☐ ZEROFILL
totalcost	DECIMAL(12,2)			☐ UNSIGNED ☐ ZEROFILL
assigndate	DATE			
employee_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL

Figure B. 14 Booking Table



Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default V
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL
name	VARCHAR(100)			☐ BINARY	NULL
gender_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL
dob	DATE				NULL
nic	CHAR(10)			☐ BINARY ☐ ASCII ☐ UNIC	NULL
civilstatus_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL
address	TEXT				NULL
mobile	CHAR(10)			☐ BINARY ☐ ASCII ☐ UNIC	NULL
land	CHAR(10)			☐ BINARY ☐ ASCII ☐ UNIC	NULL
designation_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL
employeestatus_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL
email	VARCHAR(100)			☐ BINARY	NULL
assigned	DATE				NULL
image	BLOB				NULL

Figure B. 15 Employee Table



Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
role_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
module_id	INT(11)	✓		☐ UNSIGNED ☐ ZEROFILL	NULL	
ins	INT(11)			☐ UNSIGNED ☐ ZEROFILL	NULL	
sel	INT(11)			☐ UNSIGNED ☐ ZEROFILL	NULL	
upd	INT(11)			☐ UNSIGNED ☐ ZEROFILL	NULL	
del	INT(11)			☐ UNSIGNED ☐ ZEROFILL	NULL	

Figure B. 16 Privilege Table

APPENDIX-C

USER DOCUMENTATION

Catering Service and Reception Hall Management System for Mayadunne Catering Service has been developed with lots of processes and characters in order to perform operations efficiently. In order to acquire the maximum from the implemented system, it is very significant for a user to discover all the characters of the system and how to use these processes and characters efficiently. User documentation provides initial overview knowledge on using the Catering Service and Reception Hall Management System step by step.

Add a new booking to the system.

Following figure C.1 represents the Main Window of the system. To view main window the user should enter correct username and password to obtain authorized entry to the system. Once logged in, the user is focused to the main window.

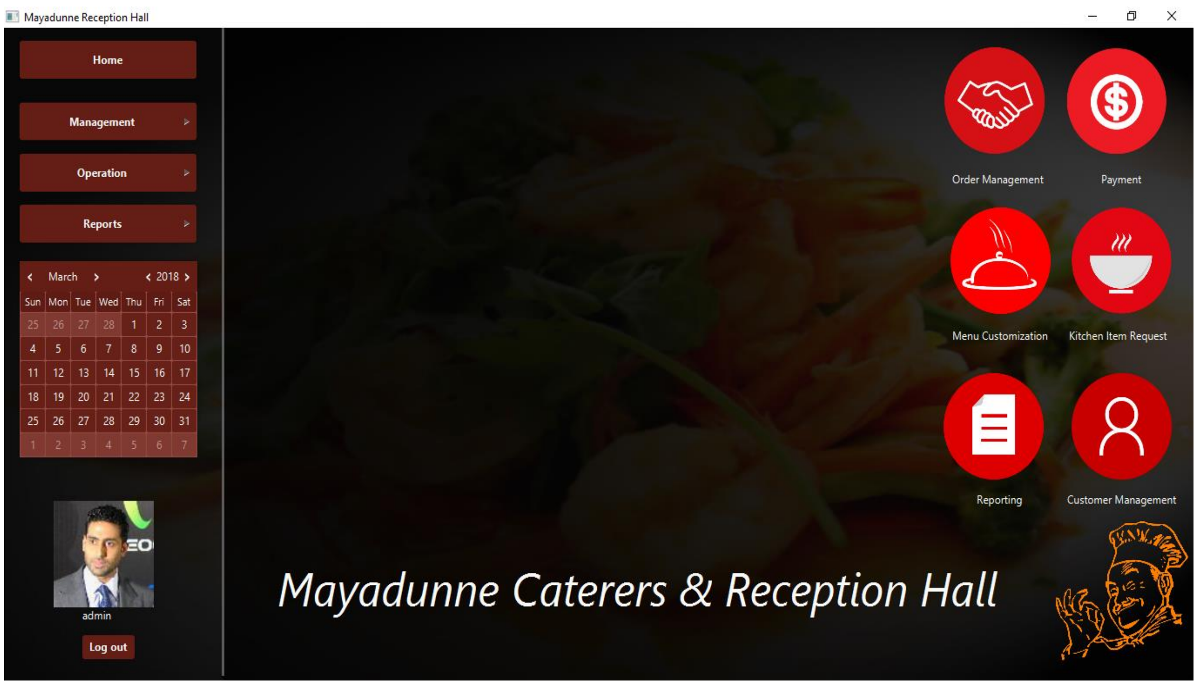


Figure C. 1 Main Window

Following figure C.2 shows the booking sub tab under Function Booking tab. In here click Function Booking Management button.

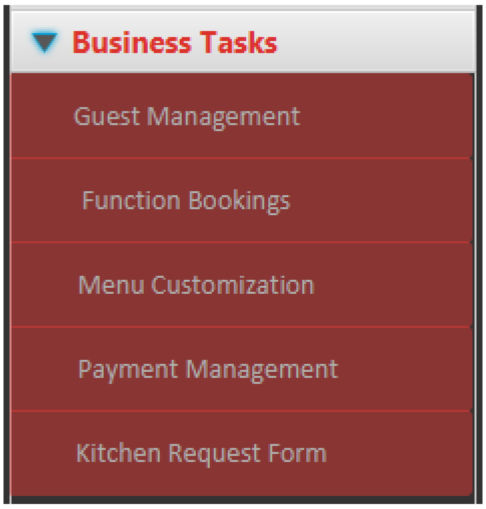
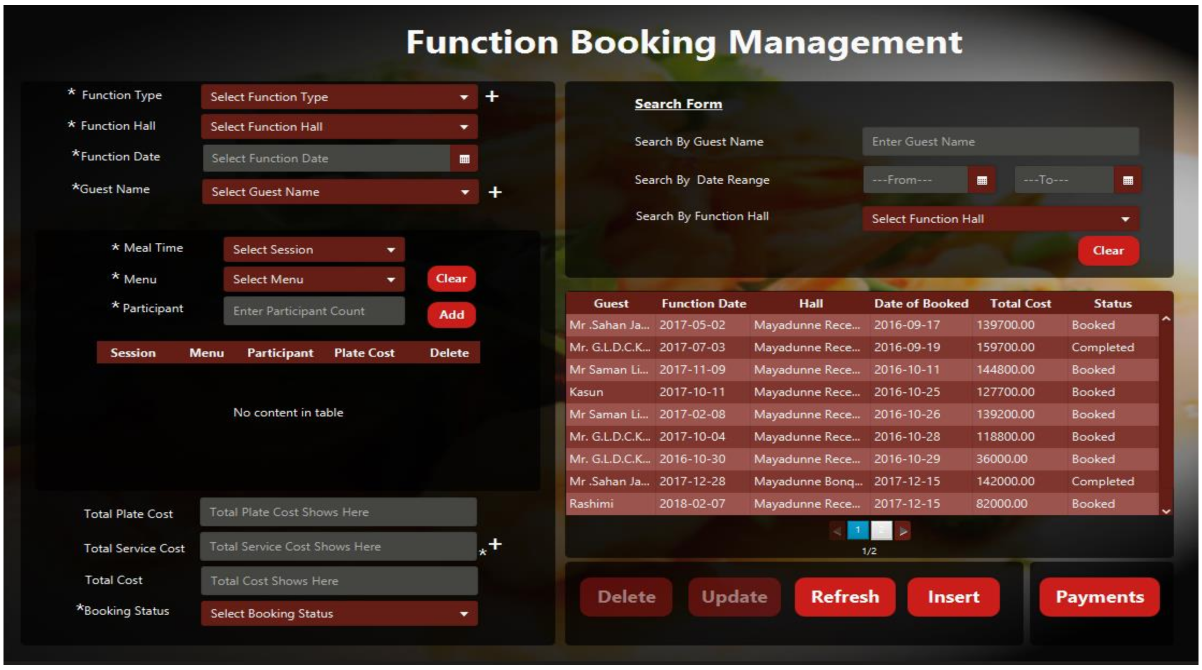


Figure C. 2 Function Booking sub tab

Then Function Booking Form will display as follows. Figure C.3 represents the Function Booking form.



Session	Menu	Participant	Plate Cost	Delete

No content in table

Guest	Function Date	Hall	Date of Booked	Total Cost	Status
Mr .Sahan Ja...	2017-05-02	Mayadunne Rece...	2016-09-17	139700.00	Booked
Mr. G.L.D.C.K...	2017-07-03	Mayadunne Rece...	2016-09-19	159700.00	Completed
Mr Saman Li...	2017-11-09	Mayadunne Rece...	2016-10-11	144800.00	Booked
Kasun	2017-10-11	Mayadunne Rece...	2016-10-25	127700.00	Booked
Mr Saman Li...	2017-02-08	Mayadunne Rece...	2016-10-26	139200.00	Booked
Mr. G.L.D.C.K...	2017-10-04	Mayadunne Rece...	2016-10-28	118800.00	Booked
Mr. G.L.D.C.K...	2016-10-30	Mayadunne Rece...	2016-10-29	36000.00	Booked
Mr .Sahan Ja...	2017-12-28	Mayadunne Bonq...	2017-12-15	142000.00	Completed
Rashimi	2018-02-07	Mayadunne Rece...	2017-12-15	82000.00	Booked

Figure C. 3 Function Booking form

Then select the Function type combo box. Figure C.4 represents a form control of Function Booking form.

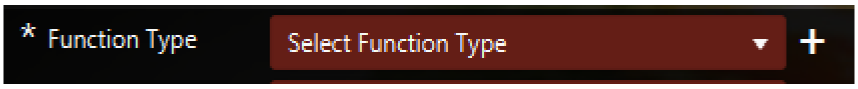


Figure C. 4 Figure C.4 Function Type combo box

Next click on add button to add details of the food menu into the inner table. Figure C.7 shows inner table

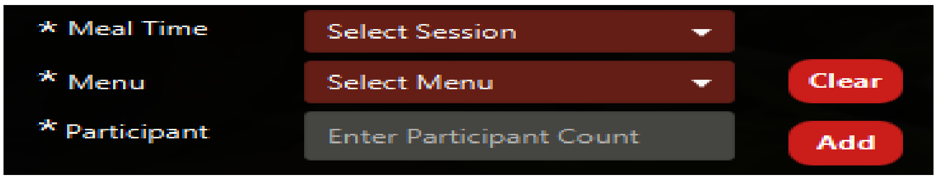
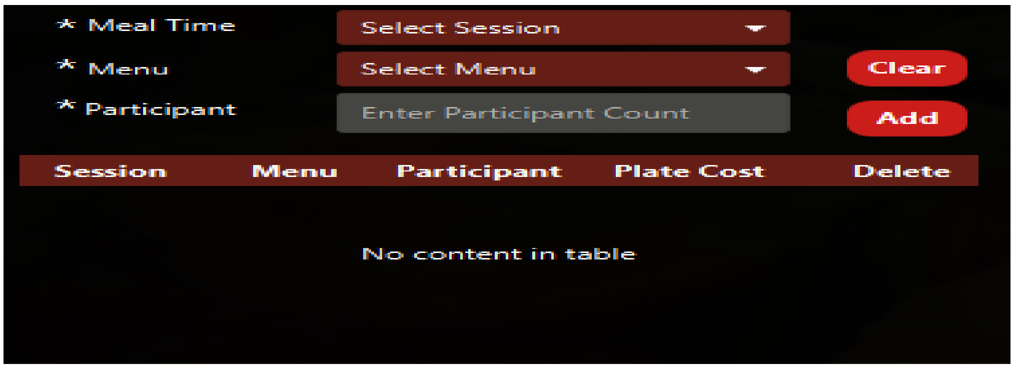


Figure C. 5 Menu Selection Combo Boxes

Next click on add button to add details of the food menu into the inner table. Figure C.7 shows inner table.



Session	Menu	Participant	Plate Cost	Delete

No content in table

Figure C. 6 Insert a Food Menu into Inner Table

Then click "Add" button to save. Figure C.8 shows the actions bar in booking form. You can select "Update" and "Delete" button to change item and delete item details, when you click on the row of table.

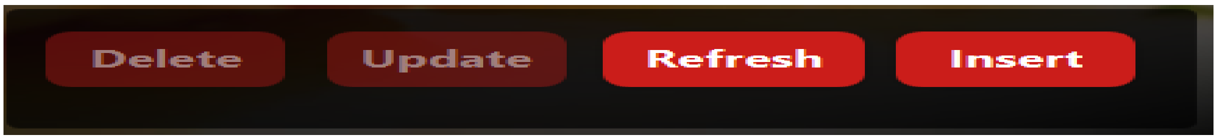


Figure C. 7 Actions Bar

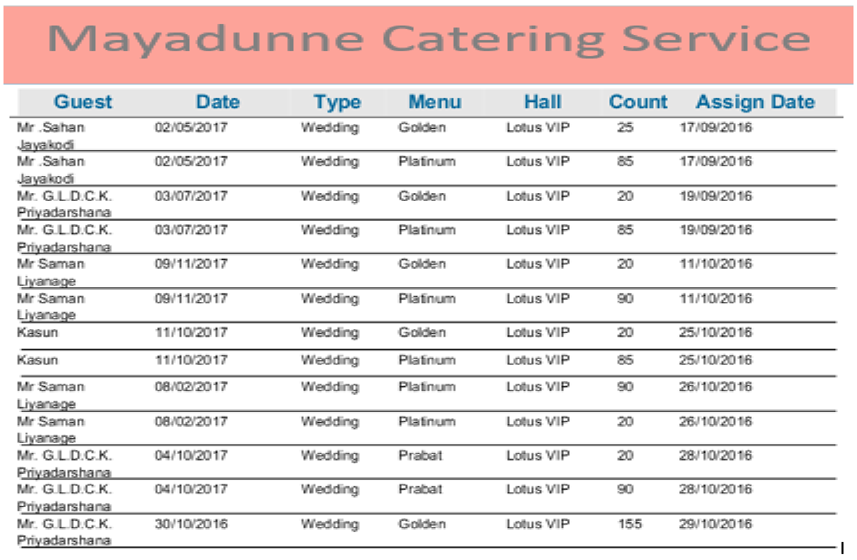
APPENDIX-D

MANAGEMENT REPORTS

System agrees users such as managers, administrators to produce several types of reports in order to use for the decision making procedure of the business. As system generates daily, monthly & yearly reports after deploying them management can use them to explore & classify the movements, arrangements & periodic variations of the business & prediction of future business situations.

D.1 FUNCTION BOOKING REPORT

A report for every booking can be produced for the use of create menu form and send it to the kitchen. In here guest name, booking date, hall, selected menu and relevant details are printed in report. Figure D.1 shows the function booking report.



Mayadunne Catering Service

Guest	Date	Type	Menu	Hall	Count	Assign Date
Mr .Sahan Jayakodi	02/05/2017	Wedding	Golden	Lotus VIP	25	17/09/2016
Mr .Sahan Jayakodi	02/05/2017	Wedding	Platinum	Lotus VIP	85	17/09/2016
Mr. G.L.D.C.K. Priyadarshana	03/07/2017	Wedding	Golden	Lotus VIP	20	19/09/2016
Mr. G.L.D.C.K. Priyadarshana	03/07/2017	Wedding	Platinum	Lotus VIP	85	19/09/2016
Mr Saman Liyanage	09/11/2017	Wedding	Golden	Lotus VIP	20	11/10/2016
Mr Saman Liyanage	09/11/2017	Wedding	Platinum	Lotus VIP	90	11/10/2016
Kasun	11/10/2017	Wedding	Golden	Lotus VIP	20	25/10/2016
Kasun	11/10/2017	Wedding	Platinum	Lotus VIP	85	25/10/2016
Mr Saman Liyanage	08/02/2017	Wedding	Platinum	Lotus VIP	90	26/10/2016
Mr Saman Liyanage	08/02/2017	Wedding	Platinum	Lotus VIP	20	26/10/2016
Mr. G.L.D.C.K. Priyadarshana	04/10/2017	Wedding	Prabat	Lotus VIP	20	28/10/2016
Mr. G.L.D.C.K. Priyadarshana	04/10/2017	Wedding	Prabat	Lotus VIP	90	28/10/2016
Mr. G.L.D.C.K. Priyadarshana	30/10/2016	Wedding	Golden	Lotus VIP	155	29/10/2016

Figure D. 1 Function Booking Report

D.4 GUEST ARRIVAL BY MONTH

Guest arrival has been recognized under each month to understand how arrival variation by guest. Then management team can get decisions to improve guest arrival. Following figure D.4 shows the guest arrival by month chart. User can customize chart depending on their needs. User can select relevant criteria to generate chart.

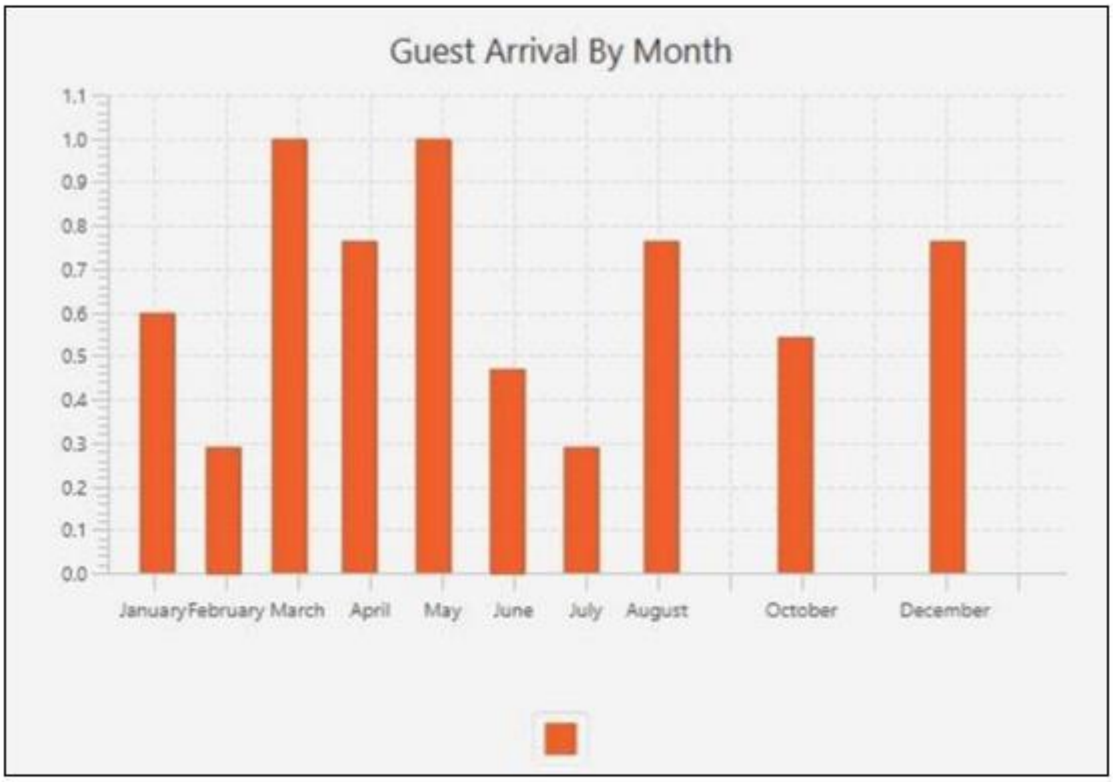


Figure D. 2 Guest Arrival by Month Chart

APPENDIX-E

TEST RESULTS

E.1 TEST CASE FOR MENU ITEM MANAGEMENT MODULE

Test cases and test results for Menu Item Management Module are listed in table E.1 the table has test menu category, name, price, expected result and the actual result generated by each test case.

Test No	Test Description	Expected Result	Pass/Fail
01	Select Combo Box	List all the Item Category to the Combo Box	pass
02	Menu Name is less than 2 characters or more than 30	Text Field background color change to red Tooltip will be shown as “Menu Name is Invalid”	pass
03	Menu Item price with 2 decimal places	Text Field background color change to red Tooltip will be shown as “Price is Invalid”	pass

04	Enter all information correctly and click Insert	Confirm Message box will be Shown as “Do you want to add this Menu Item with following details?” If click “YES” button Message box will be shown as “Menu Item Added Successfully” and added data will be shown below in the table If click “NO” button Cancel Save and Text Fields Values will be Clear	pass
05	Select a row from the table	Selected Floor details will be load into the relevant Text Fields Add button will be disabled and Update, Delete buttons will be enabled	Pass

06	Select a row from the table and change the menu item data and click Update button	Confirm Message box will be Shown as "Do you want to update this menu Item with following details?" If click "YES" button Message box will be shown as "Menu Item Updated Successfully" and updated data will be shown below in the table If click "NO" button Cancel Update.	pass
07	Select a row from the table and click Delete button	Confirm Message box will be shown as "Do you want to delete This menu item?" If click "YES" button Message box will be hown as "Menu Item Deleted Successfully" table will be refreshed If click "NO" button Cancel Delete	pass
08	Select one from the Search by Category Combo box	Menu Item table will refresh according to typed text values	pass
09	Click Clear button	Clear text Fields data and clear selection of the table	pass

Table E. 1 Test cases and results for Menu Item Management Module

E.2 TEST CASE FOR MENU MANAGEMENT MODULE

Test cases and test results for Menu Management Module are listed in table E.2 the table has test no, description, expected result and the actual result generated by each test case.

Test No	Test Description	Expected Result	Pass/Fail
01	Menu Name is less than 2 characters or more than 30	Text Field background color change to red Tooltip will be shown as "Hotel Food Name is Invalid"	pass
02	Select Combo Box	List all the Menu Types to the Combo Box	pass
03	Select Menu Item Category Combo Box and Select menu Item then click "Add" Button	Selected Menu Category and Item are added into Inner Table and create delete button in last column of the line and price of the customized menu appear in below text-field	pass
04	Enter all information correctly and click Save	Confirm Message box will be Shown as "Do you want to add this Menu with following details?" If click "YES" button Message box will be shown as "Menu Added Successfully" and added data will be shown below in the table If click "NO" button Cancel Save and Text Fields Values will be Clear	pass
05	Select a row from the table	Selected Menu details will be load into the relevant Text Fields Add button will be disabled and Update, Delete buttons will be enabled	Pass

06	Select a row from the table and change the Menu data and click Update button	Confirm Message box will be Shown as "Do you want to update this menu with following details?" If click "YES" button Message box will be shown as "Menu Updated Successfully" and updated data will be shown below in the table If click "NO" button Cancel Update.	pass
07	Select a row from the table and click Delete button	Confirm Message box will be Shown as "Do you want to delete this Menu?" If click "YES" button Message box will be shown as "Menu Deleted Successfully" table will be refreshed If click "NO" button Cancel Delete	pass
08	Select one from Search by Type Combo Box and Search by Menu Status Combo Box	Menu table will refresh according to selected values	pass
09	Click Clear button	Clear text Fields data and clear selection of the table	pass

Table E. 2 Test cases and results for Menu Management Module

APPENDIX-F

CODE TESTING

F.1 CODE FOR GET ALL OBJECTS

Following method can be used to get all table records as java objects (POJO) to given query using Hibernate Session Factory. Hibernate maps the relations to relevant java objects.

```
    public static ObservableList select(String nameQuery) {
        ObservableList list = null;
        Session session =
 HibernateUtil.getSessionFactory().openSession();
        Transaction transaction = null;

        try {
            transaction = session.beginTransaction();
            Query query = session.getNamedQuery(nameQuery);
            list = FXCollections.observableList(query.list());
        } catch (HibernateException e) {
            System.out.println(e.getMessage());
            if (transaction != null) {
                transaction.rollback();
            }
            System.out.println(e.getMessage());
            System.out.println("Can't select");
        } finally {
            session.close();
        }
        return list;
    }
```

Following method can be used to get all table records as java objects (POJO) to given query and parameters using Hibernate Session Factory. Hibernate maps the relations to relevant java objects.

```
    public static ObservableList select(String nameQuery, HashMap
properties) {
        ObservableList list = null;
        Session session =
HibernateUtil.getSessionFactory().openSession();
        Transaction transaction = null;

        try {

            transaction = session.beginTransaction();
            Query query =
session.getNamedQuery(nameQuery).setProperties(properties);
            list = FXCollections.observableList(query.list());
        } catch (HibernateException e) {
            System.out.println(e.getMessage());

            if (transaction != null) {


                 transaction.rollback();
             }
             System.out.println(e.getMessage());
             System.out.println("Can't select list");
         } finally {
             session.close();
         }
         return list;
     }
```

F.2 CODE FOR FILL COMBO BOX

Following method can be used to fill gender combo box which is a form control. It gets all gender types using control layer (Dao) access and sets to the gender combo box.

```
    @FXML
    private ComboBox<Gender> cmbGender;

      private void fillComboGender() {
        ObservableList<Gender> genders = GenderDao.getAll();
        cmbGender.setItems(genders);
        cmbGender.getSelectionModel().select(-1);
    }
```

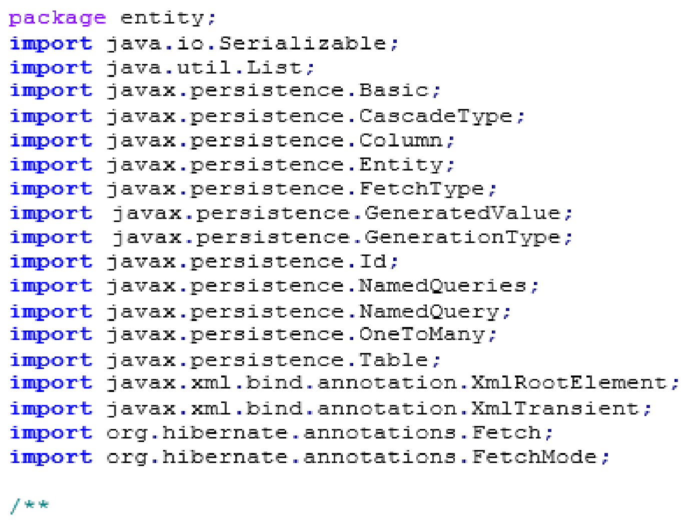
F.3 CODES FOR FILL TABLE VIEW

Following method can be used to fill guest table. (In JavaFX a table is known as table view) which is a form control. It gets guest records using control layer (Dao) access and sets to each table column with given data types.

```
@FXML
private TableColumn<Guest, String> tbcGuestCode;
@FXML
private TableColumn<Guest, Guesttitle> tbcGuestTitle;
@FXML
private TableColumn<Guest, String> tbcGuestIdentifier;
@FXML
private TableColumn<Guest, String> tbcGuestName;
@FXML
private TableColumn<Guest, Gender> tbcGuestGender;
@FXML
private TableColumn<Guest, String> tbcGuestAddress;
@FXML
private TableColumn<Guest, Country> tbcGuestCountry;
@FXML
private TableColumn<Guest, Guesttype> tbcGuestType;
@FXML
private TableColumn<Guest, Company> tbcGuestCompany;
@FXML
private TableColumn<Guest, String> tbcGuestPhone;
@FXML
private void fillGuestTableView(ObservableList<Guest> guests){



        tbcGuestCode.setCellValueFactory(new
PropertyValueFactory<>("guestid"));
        tbcGuestTitle.setCellValueFactory(new
PropertyValueFactory<>("guesttitleId"));
        tbcGuestIdentifier.setCellValueFactory(new
PropertyValueFactory<>("identifier"));
        tbcGuestName.setCellValueFactory(new
PropertyValueFactory<>("name"));
        tbcGuestGender.setCellValueFactory(new
PropertyValueFactory<>("genderId"));
        tbcGuestAddress.setCellValueFactory(new
PropertyValueFactory<>("address"));
        tbcGuestCountry.setCellValueFactory(new
PropertyValueFactory<>("countryId"));
        tbcGuestPassportNo.setCellValueFactory(new
PropertyValueFactory<>("passportno"));
        tbcGuestType.setCellValueFactory(new
PropertyValueFactory<>("guesttypeId"));
        tbcGuestCompany.setCellValueFactory(new
PropertyValueFactory<>("companyId"));
        tbcGuestPhone.setCellValueFactory(new
PropertyValueFactory<>("phone1"));
        tblGuest.getItems().clear();
        tblGuest.setItems(guests);
        labGuestRecCount.setText("[ "+String.valueOf(guests.size())+"
Record(s) Found ]");
    }
```

F.4 ENTITY CLASS FOR GUEST TYPE

Following Entity class is known as a POJO. Entity encapsulates attributes and methods, where attributes acquire private access and method acquire public access. Additionally in here, there are JPA annotations which are runtime syntactical metadata.



```
package entity;
import java.io.Serializable;
import java.util.List;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import  javax.persistence.GeneratedValue;
import  javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;
import org.hibernate.annotations.Fetch;
import org.hibernate.annotations.FetchMode;

/**
```

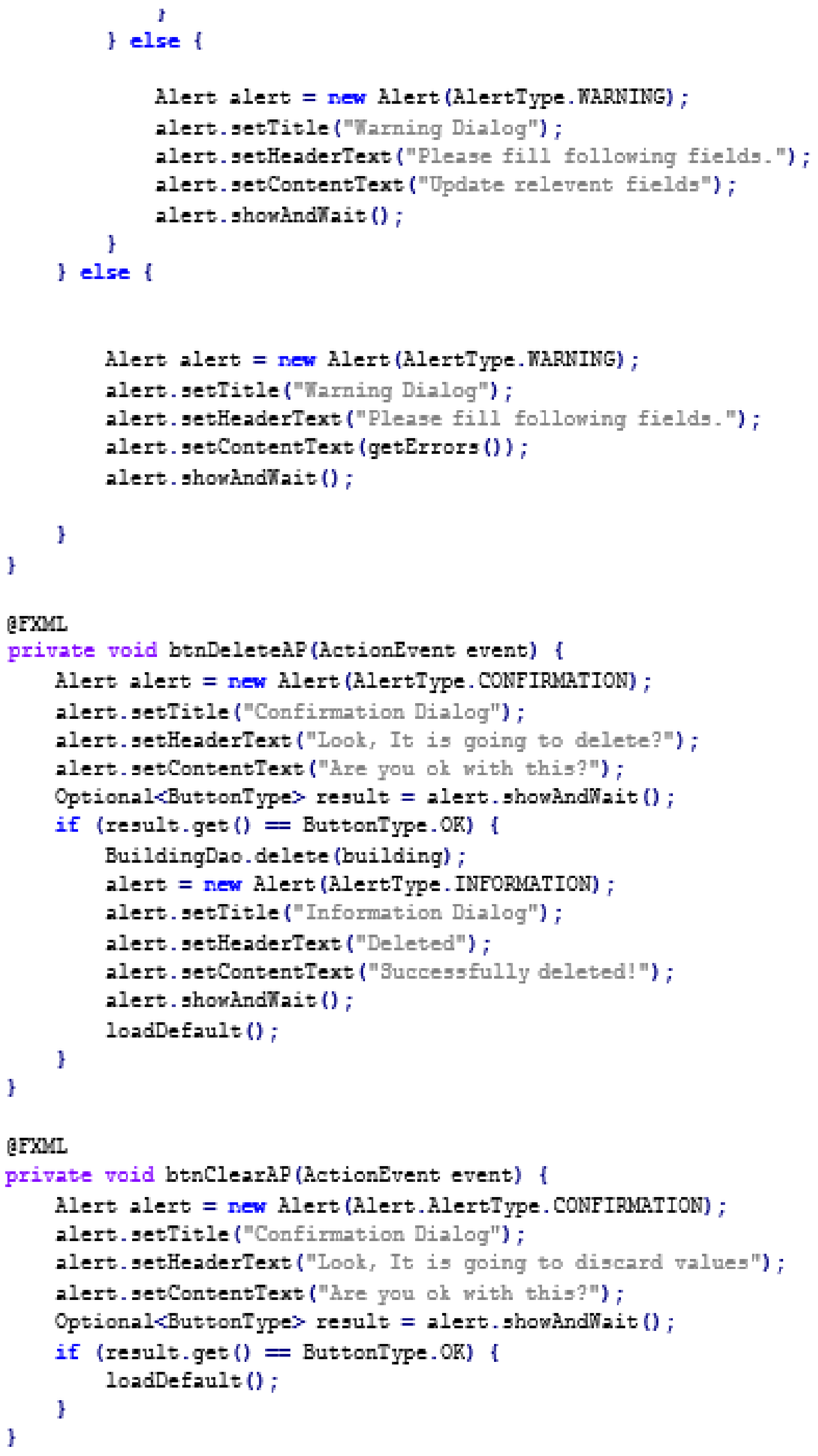
```
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Guesttype.findAll", query = "SELECT g FROM
Guesttype g"),
    @NamedQuery(name = " Guesttype.findById", query = "SELECT g FROM
Guesttype g WHERE g.id = :id"),
    @NamedQuery(name = " Guesttype.findByType", query = "SELECT g
FROM Guesttype g WHERE g.type = :type")})
public class Guesttype implements Serializable {
private static final long serialVersionUID = 1L;
 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 @Basic(optional = false)
 @Column(name = "id")
 private Integer id;
 @Basic(optional = false)
 @Column(name = "name")
 private String name;
 @OneToMany(cascade = CascadeType.ALL, mappedBy = "customertypeId",
etch = FetchType.EAGER)
 private List<Customer> customerList;
    public Guesttype() {
    }

    public Bedtype(Integer id) {
        this.id = id;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    @XmlTransient
    public List<Guest> getGuestList() {
        return guestList;
    }

    public void setGuestList(List<Guest> guestList) {
        this. guestList = guestList;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id != null ? id.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
fields are not set
        if (!(object instanceof Guesttype)) {
            return false;
        }
        Guesttype other = (Guesttype) object;
        if ((this.id == null && other.id != null) || (this.id != null
&& !this.id.equals(other.id))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return name;
    }

}
```

F.5 CONTROLLER CLASS FOR FUNCTIONBOOKING.FXML

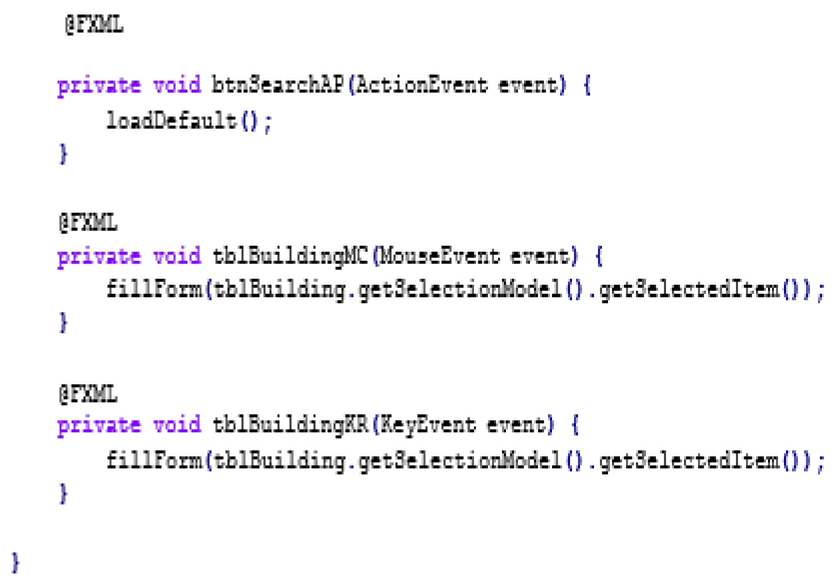
Following java class is the controller of FunctionBooking.fxml. In here all controllers, event, event listeners and styles are applied using JavaFX API. package ui;

```
import  dao.BuildingDao;
import  entity.Building;
import java.net.URL;
import java.util.Optional;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.ButtonType;
import javafx.scene.control.Label;
import javafx.scene.control.Pagination;
import javafx.scene.control.ScrollPane;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.image.ImageView;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
```

```
@FXML
private TextArea txtDescription;
@FXML
private Label labRecordCount;
@FXML
private TextField txtNameSearch;
@FXML
private Button btnSearch;
@FXML
private Button btnAdd;
@FXML
private Button btnUpdate;
@FXML
private Button btnDelete;
@FXML
private Button btnClear;
@FXML
private TextField txtName;
@FXML
private Pagination paginate;
@FXML
private TableView<Booking> tblBooking;
@FXML
private TableColumn< Booking, Guest> colGuestName;
@FXML
private TableColumn< Booking, Date> colFunctionDate;
@FXML
private TableColumn< Booking, Hall> colHall;
@FXML
private TableColumn< Booking, Date> colBookingDate;
@FXML
private TableColumn< Booking, BigDecimal> colTotalCost;
@FXML
private TableColumn< Booking, Bookingstatus> colStatus;
Building building;
Building oldBuilding;

@Override
public void initialize(URL url, ResourceBundle rb) {
    loadDefault();
}

private void loadDefault(){
    building = new Building();
    oldBuilding = null;
    txtBuilding.setText("");
    txtName.setText("");
    txtDescription.setText("");
    txtRemark.setText("");
    txtNameSearch.setText("");
    fillTableBuilding(BuildingDao.getAll());
    setStyle(FXStyle.initial);
    disableButtons(false, true, true);}
    private void setStyle(String color){
    txtBuilding.setStyle(color);
    txtName.setStyle(color);
        if (!txtDescription.getChildrenUnmodifiable().isEmpty()) {

((ScrollPane)txtDescription.getChildrenUnmodifiable().get(0)).getCont
ent().setStyle(color);
        }
        if (!txtRemark.getChildrenUnmodifiable().isEmpty()) {

((ScrollPane)txtRemark.getChildrenUnmodifiable().get(0)).getContent()
.setStyle(color);
        }
    }

    private void disableButtons(boolean add, boolean update, boolean
delete){
        btnAdd.setDisable(add);
        btnUpdate.setDisable(update);
        btnDelete.setDisable(delete);
    }

    private void fillTableBuilding(ObservableList<Building>
buildings){
        tbcCode.setCellValueFactory(new
PropertyValueFactory<>("buildingid"));
        tbcName.setCellValueFactory(new
PropertyValueFactory<>("name"));
        tbcDescription.setCellValueFactory(new
PropertyValueFactory<>("description"));
        tbcRemark.setCellValueFactory(new
PropertyValueFactory<>("remark"));
        tblBuilding.setItems(buildings);
        labRecordCount.setText("[
"+String.valueOf(buildings.size())+" Record(s) Found ]");
        tblBuilding.getSelectionModel().clearSelection();
    }

    private void fillForm(Building building){
        disableButtons(true, false, false);
        setStyle(FXStyle.valid);
        this.building = BuildingDao.getById(building.getId());
        this.oldBuilding = BuildingDao.getById(building.getId());
        txtBuilding.setText(building.getBuildingid());
        txtName.setText(building.getName());
        txtDescription.setText(building.getDescription());
        txtRemark.setText(building.getRemark());
    }

    @FXML
    private void txtBuildingKR(KeyEvent event) {
        if (building.setBuildingid(txtBuilding.getText().trim())) {
            if (oldBuilding != null &&
!building.getBuildingid().equals(oldBuilding.getBuildingid())) {
    @FXML
    private void txtNameKR(KeyEvent event) {
        if (building.setName(txtName.getText().trim())) {
            if (oldBuilding != null &&
!building.getName().equals(oldBuilding.getName())) {
                txtName.setStyle(FXStyle.updated);
            } else {
                txtName.setStyle(FXStyle.valid);
            }
        } else {
            txtName.setStyle(FXStyle.invalid);
        }
    }

    @FXML
    private void txtDescriptionKR(KeyEvent event) {
        if (building.setDescription(txtDescription.getText().trim()))
{
            if (oldBuilding != null &&
!building.getDescription().equals(oldBuilding.getDescription())) {
                txtDescription.setStyle(FXStyle.updated);

((ScrollPane)txtDescription.getChildrenUnmodifiable().get(0)).getCont
ent().setStyle(FXStyle.updated);
            } else {
                txtDescription.setStyle(FXStyle.valid);

((ScrollPane)txtDescription.getChildrenUnmodifiable().get(0)).getCont
ent().setStyle(FXStyle.valid);
            }
        } else {
            txtDescription.setStyle(FXStyle.invalid);

((ScrollPane)txtDescription.getChildrenUnmodifiable().get(0)).getCont
ent().setStyle(FXStyle.invalid);
        }
    }

    @FXML
    private void txtRemarkKR(KeyEvent event) {
        if (building.setRemark(txtRemark.getText().trim())) {
            if (oldBuilding != null &&
!building.getRemark().equals(oldBuilding.getRemark())) {
                txtRemark.setStyle(FXStyle.updated);

((ScrollPane)txtRemark.getChildrenUnmodifiable().get(0)).getContent()
.setStyle(FXStyle.updated);
            } else {
                txtRemark.setStyle(FXStyle.valid);

((ScrollPane)txtRemark.getChildrenUnmodifiable().get(0)).getContent()
.setStyle(FXStyle.valid);
            }
    private String getErrors(){

        String confirmation = "";

        if (building.getBuildingid() == null) {
            confirmation = confirmation + "Building Id is Invalid\n";
        }
        if (building.getName() == null) {
            confirmation = confirmation + "Name is Invalid\n";
        }

        if (building.getDescription() == null) {
            confirmation = confirmation + "Description is Invalid\n";
        }

        if (building.getRemark() == null) {
            confirmation = confirmation + "Remark is Invalid\n";
        }

        return confirmation;
    }

    public String getUpdates() {

        String confermation = "";

        if
(!building.getBuildingid().equals(oldBuilding.getBuildingid())) {
            confermation = confermation +
oldBuilding.getBuildingid()+ " chnaged to " +
building.getBuildingid()+ "\n";
        }

        if (!building.getName().equals(oldBuilding.getName())) {
            confermation = confermation + oldBuilding.getName()+ "
chnaged to " + building.getName()+ "\n";
        }

        if
(!building.getDescription().equals(oldBuilding.getDescription())) {
            confermation = confermation +
oldBuilding.getDescription()+ " chnaged to " +
building.getDescription()+ "\n";
        }

        if (!building.getRemark().equals(oldBuilding.getRemark())) {
            confermation = confermation + oldBuilding.getRemark()+ "
chnaged to " + building.getRemark()+ "\n";
        }

        return confermation;}
    @FXML
    private void btnAddAP(ActionEvent event) {
        boolean validity = building.validity();
        if (validity) {
            String confermation = "Ara you sure you need to add this 
Building with following details\n "
                    + "Building Id : " + building.getBuildingid()
                    + "\nName : " + building.getName()
                    + "\nDescription : " + building.getDescription()
                    + "\nRemark : " + building.getRemark();

            Alert alert = new Alert(AlertType.CONFIRMATION);
            alert.setTitle("Confirmation Dialog");
            alert.setHeaderText("Do you want add values ?");
            alert.setContentText(confermation);
            Optional<ButtonType> result = alert.showAndWait();

            if (result.get() == ButtonType.OK) {
                BuildingDao.add(building);
                alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText(null);
                alert.setContentText("Successfully added!");
                alert.showAndWait();
                loadDefault();
            }

        } else {

            Alert alert = new Alert(AlertType.WARNING);
            alert.setTitle("Warning Dialog");
            alert.setHeaderText("Please fill following fields.");
            alert.setContentText(getErrors());
            alert.showAndWait();

        }
    }

    @FXML
    private void btnUpdateAP(ActionEvent event) {
        boolean validity = building.validity();

        if (validity) {
            String updates = getUpdates();

            if (!updates.isEmpty()) {

                Alert alert = new Alert(AlertType.CONFIRMATION);
                alert.setTitle("Confirmation Dialog");
                alert.setHeaderText("Do you want update Building with 
following modifications?\n");
                alert.setContentText(updates);
                Optional<ButtonType> result = alert.showAndWait();

                if (result.get() == ButtonType.OK) {
```



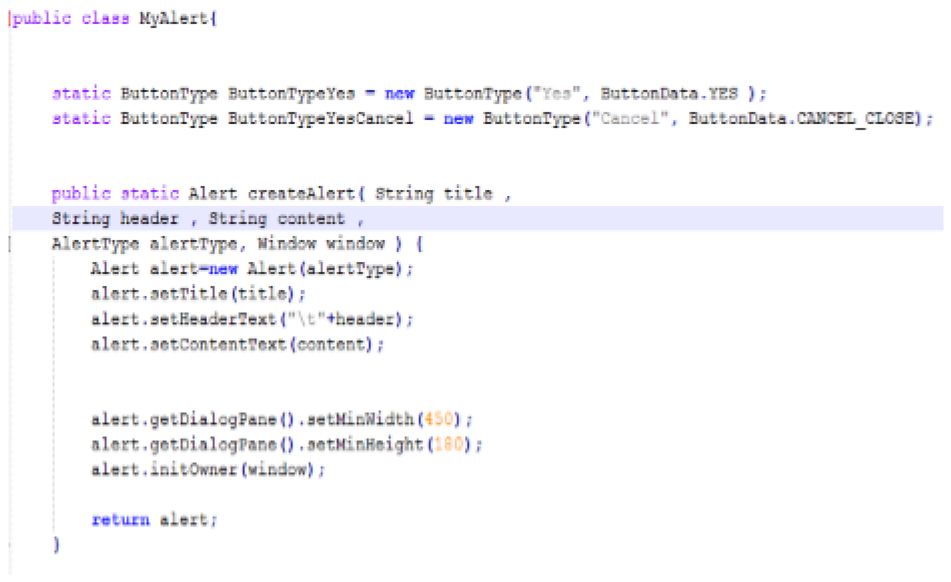
```
                }
            } else {

                Alert alert = new Alert(AlertType.WARNING);
                alert.setTitle("Warning Dialog");
                alert.setHeaderText("Please fill following fields.");
                alert.setContentText("Update relevent fields");
                alert.showAndWait();
            }
        } else {


            Alert alert = new Alert(AlertType.WARNING);
            alert.setTitle("Warning Dialog");
            alert.setHeaderText("Please fill following fields.");
            alert.setContentText(getErrors());
            alert.showAndWait();

        }
    }

    @FXML
    private void btnDeleteAP(ActionEvent event) {
        Alert alert = new Alert(AlertType.CONFIRMATION);
        alert.setTitle("Confirmation Dialog");
        alert.setHeaderText("Look, It is going to delete?");
        alert.setContentText("Are you ok with this?");
        Optional<ButtonType> result = alert.showAndWait();
        if (result.get() == ButtonType.OK) {
            BuildingDao.delete(building);
            alert = new Alert(AlertType.INFORMATION);
            alert.setTitle("Information Dialog");
            alert.setHeaderText("Deleted");
            alert.setContentText("Successfully deleted!");
            alert.showAndWait();
            loadDefault();
        }
    }

    @FXML
    private void btnClearAP(ActionEvent event) {
        Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
        alert.setTitle("Confirmation Dialog");
        alert.setHeaderText("Look, It is going to discard values");
        alert.setContentText("Are you ok with this?");
        Optional<ButtonType> result = alert.showAndWait();
        if (result.get() == ButtonType.OK) {
            loadDefault();
        }
    }
```



```
 @FXML

    private void btnSearchAP(ActionEvent event) {
        loadDefault();
    }

    @FXML
    private void tblBuildingMC(MouseEvent event) {
        fillForm(tblBuilding.getSelectionModel().getSelectedItem());
    }

    @FXML
    private void tblBuildingKR(KeyEvent event) {
        fillForm(tblBuilding.getSelectionModel().getSelectedItem());
    }

}
```

F.6 DIALOG BOXES AND CONFIRMATION MESSAGE BOXES (MYALERT.JAVA)

Following java class was used for create dialog boxes such as error messages, confirmation messages.



```
public class MyAlert{


    static ButtonType ButtonTypeYes = new ButtonType("Yes", ButtonData.YES );
    static ButtonType ButtonTypeYesCancel = new ButtonType("Cancel", ButtonData.CANCEL_CLOSE);


    public static Alert createAlert( String title ,
    String header , String content ,
    AlertType alertType, Window window ) {
        Alert alert=new Alert(alertType);
        alert.setTitle(title);
        alert.setHeaderText("\t"+header);
        alert.setContentText(content);


        alert.getDialogPane().setMinWidth(450);
        alert.getDialogPane().setMinHeight(180);
        alert.initOwner(window);

        return alert;
    }
```

APPENDIX-G

CLIENT CERTIFICATION

INDEX

GLOSSARY

- **Organizational chart** – demonstrations the structure of the organization together with all perceptions top to bottom as a flow.

- **Database** - is an organized group of data for one or more reasons, usually in digital form.

- **Backup** – means preserving a copy of information as a peripheral storage where usually track in another central computer or in peripheral disks.

- **Spiral Development** -is a software development process relating portions of both design and prototyping-in-stages, in an effort to relate advantages of top-down and bottom-up views.

- **Incremental Development** - is a cyclic methodology to software development in which conducts are repeated in a structured manner and the software is developed in increments.

- **UML** – mentions to Unified Modeling Language is a standardized general-purpose modeling language in the field of object-oriented engineering. This compromises a set of graphic element techniques to make visual models of object-oriented software intensive systems.

- **Data Definition Language** –DDL is syntax alike to a computer programming language for describing data structures (data structure is a precise way of storing and organizing data), especially database schemas.

- **Graphical User Interface** - is a type of user interface that requires users to interact with electronic strategies with images rather than text commands.

- **PHP**- Hyper-text Pre-processor is a server-side programming language. C/C++ - are two programming languages.

- **HTML**–Hyper Text Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser.

- **OSX**– previously Mac OS X, is a series of UNIX (multitasking, multi-user computer operating system)-based graphical interface operating systems (a collection of software that manages computer hardware resources and provides common services for computer programs.) Developed, marketed, and sold by Apple Inc.

- **JVM**. - A Java virtual machine is a virtual machine that can run Java byte code. It is the code execution component of the Java platform.

- **Relational database management system** - database management system (DBMS) that is based on the relational model (that all data is represented in standings of tuples, assembled into relations.)

- **JavaFX**- is a software platform for making and providing rich internet applications (RIAs are) that can run across a wide variety of devices.