Reception Hall and Catering Management System

For

New Sampath Caterers and Reception Hall

K.S.N.Wijethunga

BIT Registration Number: R141819

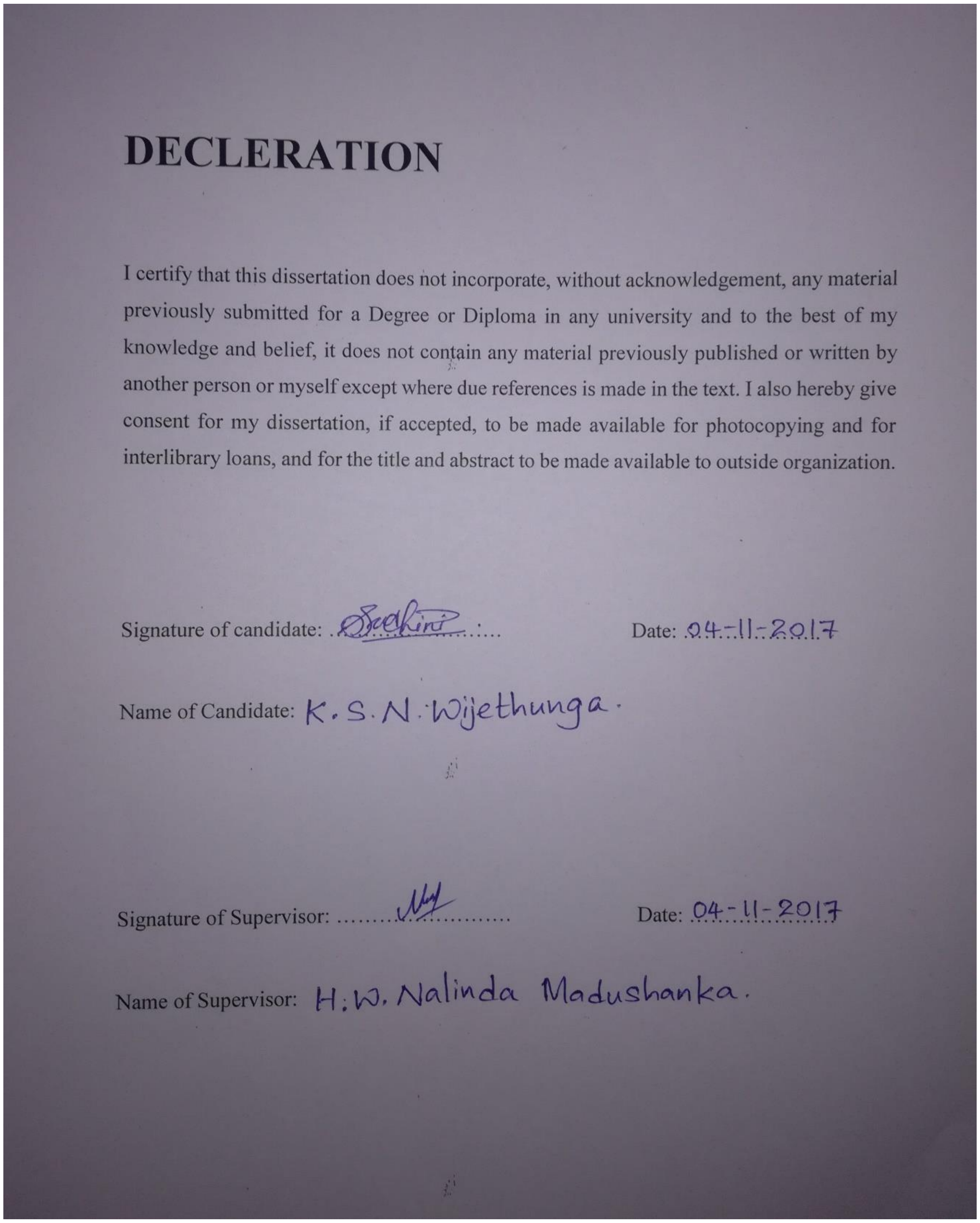
Index No: 1418191

Names of Supervisor: H.W. Nalinda Madushanka

2017

This dissertation is submitted in partial 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: 04-11-2017

Name of Candidate: K.S.N. Wijethunga.

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

Name of Supervisor: H.W. Nalinda Madushanka.

ABSTRACT

The New Sampath Caterers and Reception Hall is a well-known Reception Hall which is situated in near the Delgoda Town. There are 2 Function Halls and about 25 employees are working in this Reception Hall. Currently they are handling their day to day functions (Mostly Wedding Functions) by using a manual system. Due to this manual system, currently this company is unable to provide a better service for their clients and cannot manage day to day functions properly.

Due to weaknesses of the manual system cause to develop an automated standalone Reception Hall and catering Management System. In this system, it is supposed to have Booking Function and Menu Management primarily. Employee, User Accounts, Kitchen and Service Management are covered along with the above primary functions. Report generating and notification will support a highly valued system

NetBeans IDE (Integrated Development Environment) and JavaFX Scene Builder was used to implement the proposed system in java, javaFX, MySQL, Hibernate technologies, MVC architecture and Object Oriented approach. Unified Modelling Language was used for analysis and design the system. Jasper reports used to report generation. The proposed system is windows based solution

The proposed system will achieve the client’s functional and non-functional requirements and provide an efficient and user friendly working environment. The system has been providing excellent solution for, without using additional human effort to handle day to day business functions

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to BIT coordinators of University of Colombo School of Computing for providing us a good guidance to do the project successfully.

I am sincerely grateful to my supervisor H.W.Nalinda Mdaushanka guiding and supervising me throughout the development.

I also wish to express my gratitude for officials and other staff members of the New Sampath caterers and Reception Hall who rendered their help during the period of my project work.

Finally, my thanks go to all my friends, family and everyone who gave their valuable support in various ways and encourage me to successfully complete my project and most importantly, I would like to thank all my teachers and lecturers who help me to come this far.

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 1 – INTRODUCTION

1.1 THE CLIENT

New Sampath Caterers and Reception Hall is situated in 1km ahead from Delgoda to Colombo road .The atmosphere at Sampath Reception Hall is quiet, peaceful and serene.

This all started as small reception hall with few employees in about 6 years ago. At present, the reception hall has achieved very good name for the functions and caterings. At present, New Sampath Reception Hall and Caterers has achieved large customer loyalty attaining Delgoda area. With the popularization of New Sampath Caterers and Reception Hall, visit lot of customers to book wedding hall and their needs. Therefore, New Sampath reception hall’s vision is to give a great service to their customers. Their management welcomes new ideas to keep their path in a successful way.

They are committed to offer quality accommodation and services to their guests and fulfill guest expectations each and every time. This company is going on through a manual system which has many weaknesses and the developing system will support to growth the capability and performance of their work, by shrinking the massive manual work. Further they can continue and achieve their day to day business functions successfully.

1.2 PROBLEM DOMAIN

Currently New Sampath Caterers and Reception Hall is using a manual system to handle hotel processes. Although this is a standardized quality Reception hall it is not realizing its maximum potential due to delay of the activities of the current redundant manual system.

When a guest books a function, all the booking details are recorded in files. Calculation of bills and booked menu items 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.

1.3 MOTIVATION

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 business industry. It will help them to manage Guest details, Function Booking details, Menu Customization details, Hall Booking Details etc... And help to generating various reports more efficiently and assist the management in decision making process.

1.4 OBJECTIVES AND SCOPE OF THE PROJECT

1.4.1 Objectives

- Reduce time and operational cost required to perform functions.

  This system will be manage the document cost and human effort for Function Booking, Hall Management, Catering Management, Kitchen item Management process.

- Ensure efficient and reliable communication within the Reception Hall

  This system will manage processes between departments of the hotel efficiently.

- 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 hotel.

- Enable automated data entry methods.

Instead of error prone human effort, the system will ease of entering, 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.

1.4.2 Scope

- Manage function booking details of the company.

  The systems keep records of function halls, function types, menus and other facilities suitable for the function booking and manage them properly.

- Catering item management

  The proposed system can manage function menu and 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.

- Kitchen Item Management

  The system will handle the kitchen Items for food menus and maintain it properly.

- Accommodation and service management

  The system handles accommodations and services available for the function booking process, kitchen section and other services.

- Manage user details.

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

- Manage staff details.

  This system is able to insert, update or delete staff details.

- Report generating

  The system will generate reports when necessary.

1.5 STRUCTURE OF THE DISSERTATION

This dissertation shows the works that carried out during the each phases of the project when creating the Reception Hall Management and catering management system for New Sampath Reception Hall & caterers.

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

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 system.

2.1 FACT GATHERING TECHNIQUES

Gathering of requirements should be properly accomplished before the start of the analysis stage. Inadequate information would cause waste of time for modifications and lose the opportunity and would cause additional unnecessary overhead. There are several fact finding techniques which can be used to collect the clear and accurate information. In this project facts were gathered by using following techniques.

1. Interviews
2. Observations
3. Sampling & documentation

Management activities and other hotel activities are closely observed as the guests are served at the hotel. In this observation, it is realized so many problems with the current manual system, including, hall booking and menu management process were not well managed and guests records were not protected well from unauthorized access.

2.1.1 Interview

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.

2.2 CURRENT MANUAL SYSTEM

Staff at the New Sampath Reception Hall & caterers is recorded function bookings, kitchen, guest 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

- Difficulty in 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.

The following figure2.1 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.

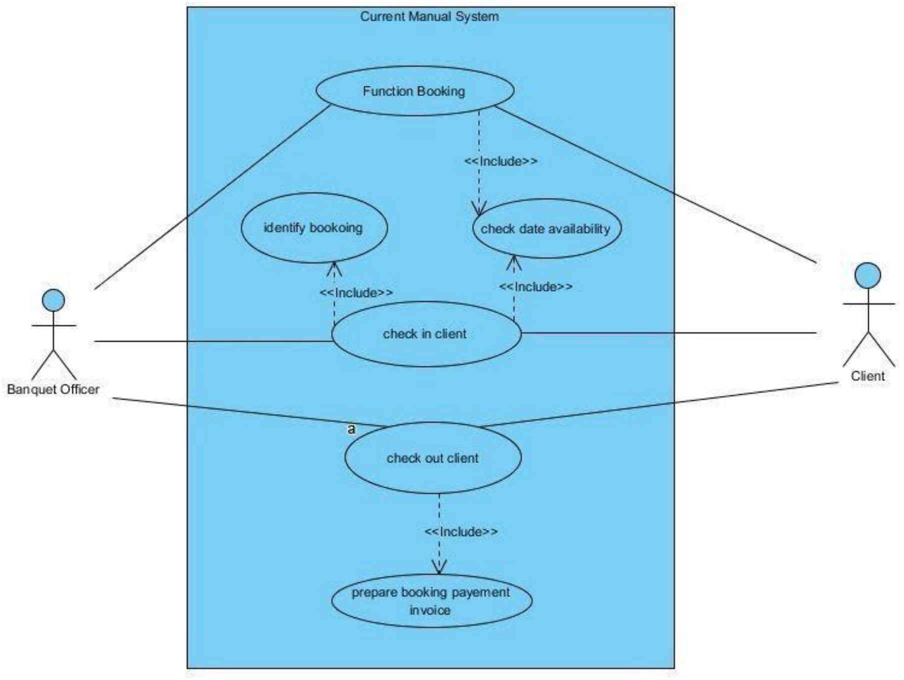


Figure 2. 1 Use case diagram for existing system

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.

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.

Check in customer: When customer check-in to the hotel, the current manual system keeps a note of customer and booking. The manual system is unable to provide a check-in card to customer. The check-in card acts as an agreement between the hall booking and customer.

Check out customer: When customer 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.

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

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 customer.

2.3 Similar Systems

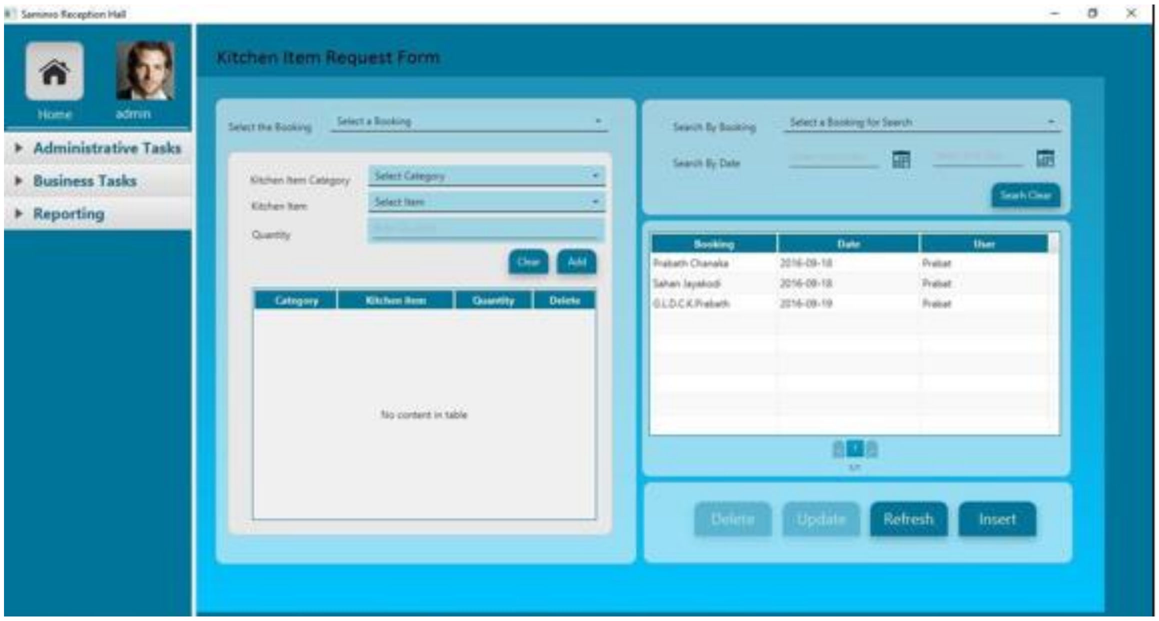
2.3.1 Reception Hall Management System for Saminro reception hall

Reception Hall Management systems for hotel industry, the system have introduced easy-to-use hotel Software. A perfect software solution for hotel industry, this comprehensive software suite comes integrated with modules for many aspects of hotel management.

Features:-

- Easy to Use
- Manage Reservations
- Reservation, Check-In and Check-Out
- Track Customers and avoid Double Bookings
- Back Office Features
- Reports

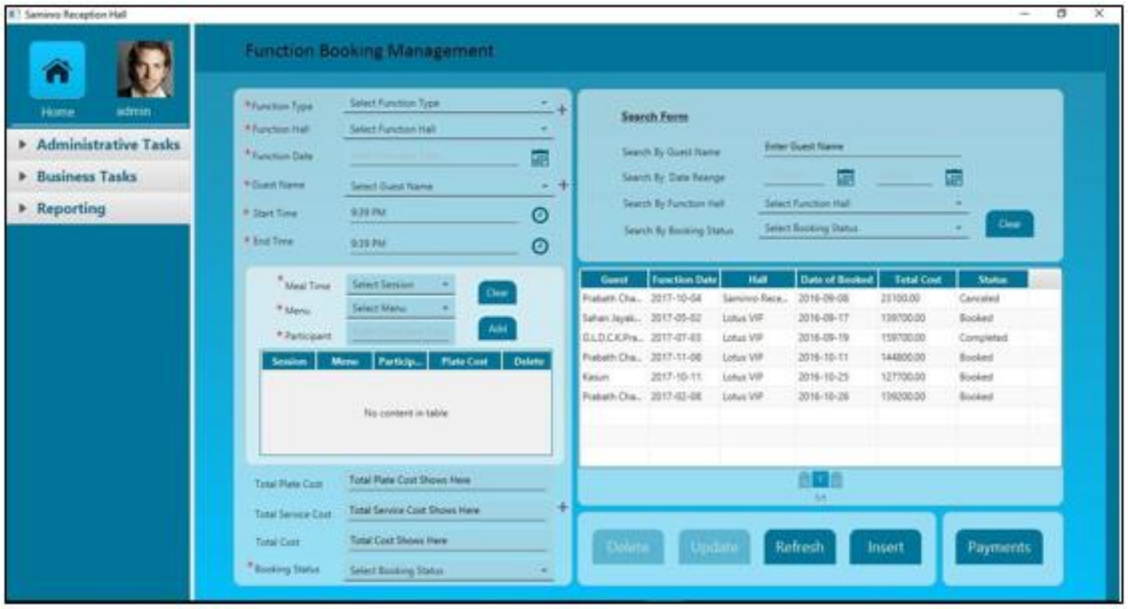
Figure 2.2 shows the sample screenshots of the Reception hall management system



Category	Kitchen Item	Quantity	Delete

Booking	Date	User
Prabath Chanaka	2016-09-18	Prabat
Sahan Jayakodi	2016-09-18	Prabat
G.L.D.C.K.Prabath	2016-09-19	Prabat

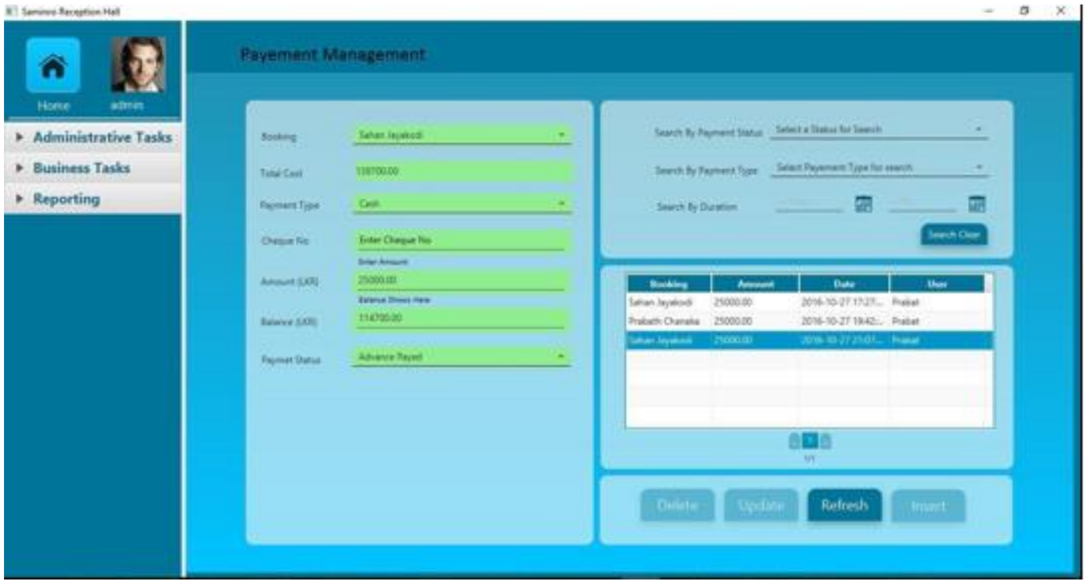
Figure 2. 2 Screenshots of Reception hall management system for Saminro hotel



Session	Menu	Particip...	Plate Cost	Delete

Guest	Function Date	Hall	Date of Booked	Total Cost	Status
Prabath Cha...	2017-10-04	Saminro Rece...	2016-09-08	23100.00	Canceled
Sahan Jayak...	2017-05-02	Lotus VIP	2016-09-17	139700.00	Booked
G.L.D.C.K.Pra...	2017-07-03	Lotus VIP	2016-09-19	159700.00	Completed
Prabath Cha...	2017-11-06	Lotus VIP	2016-10-11	144800.00	Booked
Kasun	2017-10-11	Lotus VIP	2016-10-25	127700.00	Booked
Prabath Cha...	2017-02-08	Lotus VIP	2016-10-26	139200.00	Booked

Figure 2. 3 Screenshots of Reception hall management system for Saminro hotel



Booking	Sahan Jayakodi
Total Cost	118700.00
Payment Type	Cash
Cheque No	Enter Cheque No
Amount (LKR)	25000.00
Balance (LKR)	114700.00
Paymet Status	Advance Payed

Search By Payment Status	Select a Status for Search
Search By Payment Type	Select Payement Type for search
Search By Duration	

Booking	Amount	Date	User
Sahan Jayakodi	25000.00	2016-10-27 17:27:...	Prabat
Prabath Chanaka	25000.00	2016-10-27 19:42:...	Prabat
Sahan Jayakodi	25000.00	2016-10-27 21:07:...	Prabat

Figure 2. 4 Screenshots of Reception hall management system for Saminro hotel

2.3.2 Bistonesoft

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.

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 2.5 shows the sample screenshot of Bistonesoft.

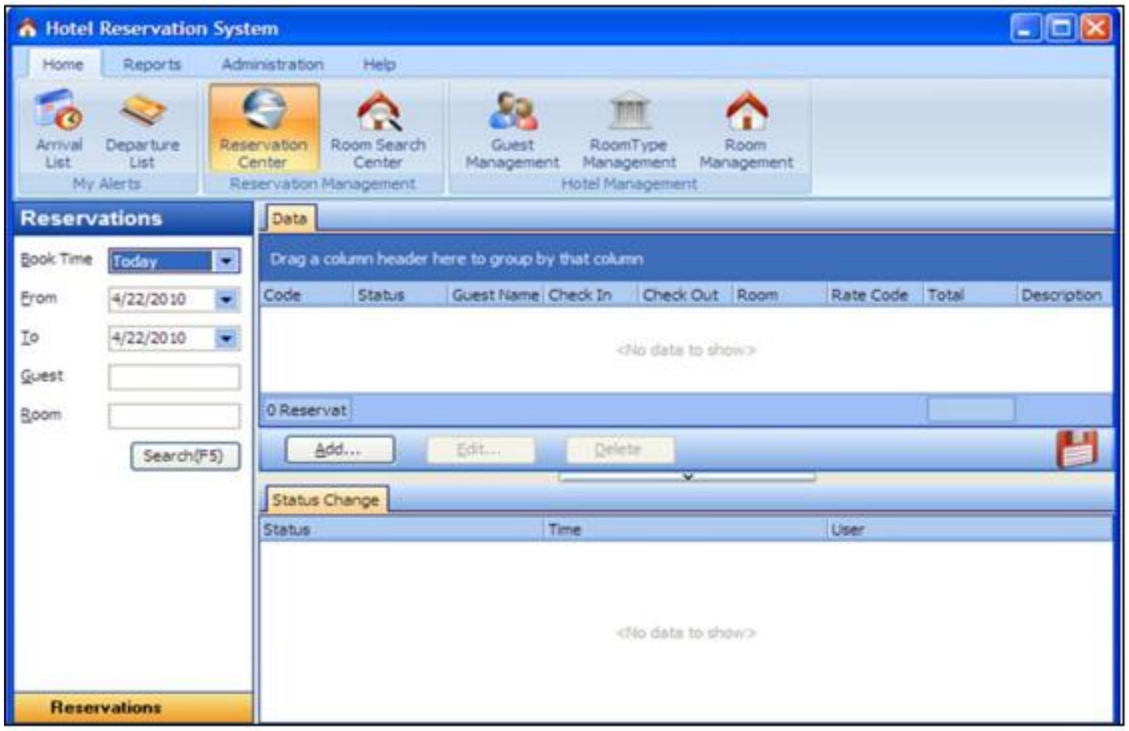


Figure 2. 5 Screenshot of Bistonesoft

2.3.3 Anand systems Inc

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.

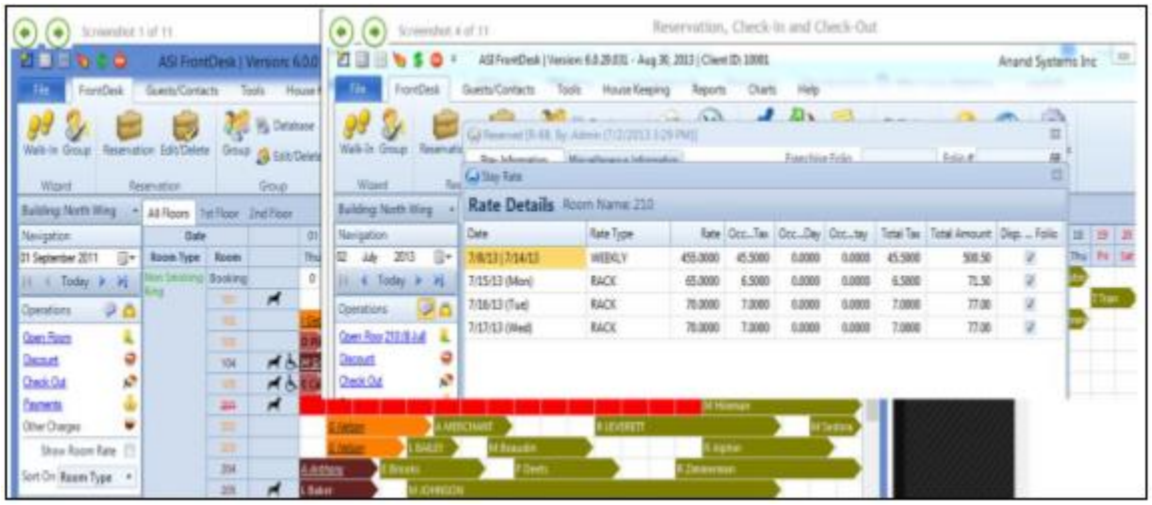


Figure 2. 6 Screenshot of Anand Systems Inc.

2.4 FUNCTIONAL REQUIREMENT

- User login management

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

- Manage booking details

  Hall booking, Catering booking and service bookings are included in booking. Each and every booking has a unique identity in the system.

- Manage customer details

  The system records customer details properly and secure manner. When user needed the system can find any customer detail.

- Manage payment details

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

- Manage menu and menu customization

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

- Manage kitchen item & requests

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

  The system can manage employee details properly. Each employee has a unique record in system.
- Manage service details

  This facility is given service booking to a service and relevant provider and solution that chosen by the customer.
- Manage hall details

  This details are given to the hall booking, this includes hall name, hall facility.
- Retrieve reports

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

2.5 NON-FUNCTIONAL REQUIREMENTS

- Usability

  Overall system should be simple as much as possible with more user friendly environment.
- Maintainability

  The system should be re-build for changing needs.
- Correctness

  The information provided by the system should be correct.
- Reliability

  The system can be managed correct information.
- Robustness

  System should be executed properly when there is an error.
- Availability

  The system should be available at any time.

2.6 PROCESS SOFTWARE DEVELOPMENT MODELS

A software development process is a framework imposed on the development of a software product. Synonyms include software life cycle and software process. There are several models for such processes, each describing approaches to a variety of software development implement process methodologies.

Software development process model used as follows,

2.6.1 Waterfall

The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards through the phases of conception, initiation, analysis, design, construction, testing, production, implementation and maintenance.

2.6.2 Prototyping

Software prototyping is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed. It is an activity that can occur in software development and is comparable to prototyping as known from other fields, such as mechanical engineering or manufacturing.

Prototyping has several benefits: The software designer and implementer can get valuable feedback from the users early in the project. The client and the contractor can compare if the software made matches the software specification, according to which the software program is built. It also allows the software engineer some insight into the accuracy of initial project estimates and whether the deadlines and milestones proposed can be successfully met. [1]

2.6.3 Iterative and incremental development

Iterative and Incremental development is any combination of both iterative design or iterative method and incremental build model for software development. The combination is of long standing and has been widely suggested for large development efforts. Iterative and incremental development are essential parts of the Modified waterfall models, Rational Unified Process, Extreme Programming and generally the various agile software development frameworks. [2]

2.6.4 Process Model for the System- RUP Model

RUP is a software development process from Rational, a division of IBM. It divides the development process into four distinct phases that each involves business modeling, analysis and design, implementation, testing, and deployment. The four phases are:

1. **Inception** - The idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.
2. **Elaboration** - The project's architecture and required resources are further evaluated. Developers consider possible applications of the software and costs associated with the development.
3. **Construction** - The project is developed and completed. The software is designed, written, and tested.
4. **Transition** - The software is released to the public. Final adjustments or updates are made based on feedback from end users.

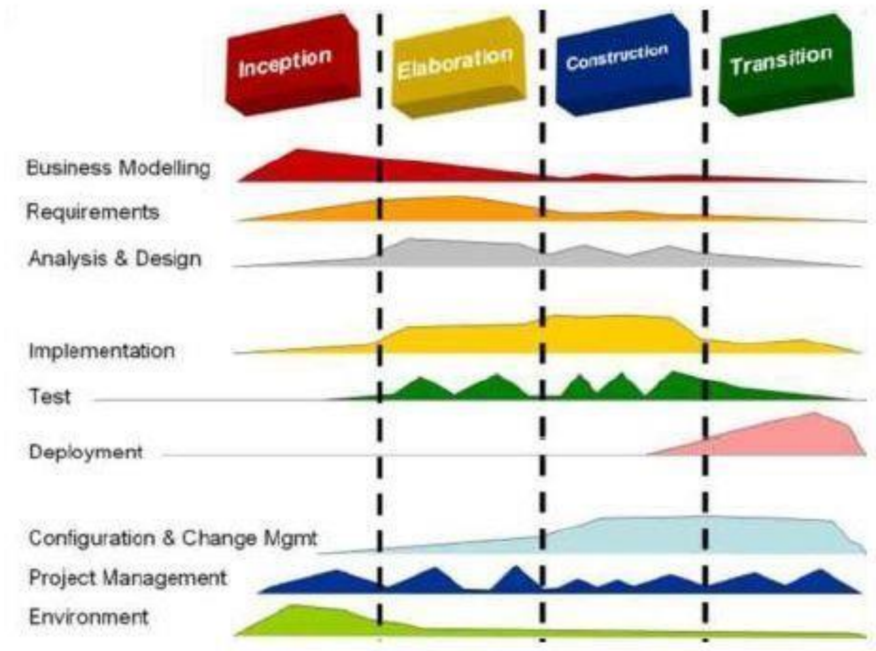


Figure 2. 7 Rational Unified process Model

CHAPTER 03- DESIGN

This is an important part of any software development project. Design of a system provide a technical solutions for the gathered functional and non-functional requirements in analysis phase. UML diagrams are used for this purpose.

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 [3].

3.1 ALTERNATE SOLUTION

When considering the scope of the New Sampath caterers and Reception hall, there are several departments and lots of processes to handle. Also the proposed system needs to available all the time. Therefore a standalone system is selected because New Sampath caterers and Reception hall have not branch network yet and they do not manage online function booking facility.

3.1.1 Reasons to choose a standalone system

- Client mostly preferred to a standalone system.
- Standalone systems not needed to pay additional cost for web hosting.
- Can have a total control over standalone applications and protect it from various vulnerabilities.
- Can use it without internet facility
- Easy to develop and easy to maintain.
- Standalone systems not needed to pay additional cost for web hosting.

3.2 OBJECT ORIENTED DESIGN

Object-oriented analysis and design (OOAD) is a software engineering approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its class, its state (data elements), and its behavior. Various models can be created to show the static structure, dynamic behavior, and run-time deployment of these collaborating objects. There are a number of different notations for representing these models. Unified Modeling Language (UML) plays an important role in Object Orientation. UML allows us to build easy to use and easy to understand models of objects so that programmers can easily write software.

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
- Accountant
- Billing clerk
- Receptionist
- Front-desk clerk

Figure 3.1 shows use cases of user management module which is related to manager and cashier. Login process is a major use case which is related to many actors (users). And use case description of user management is listed in table 3.1

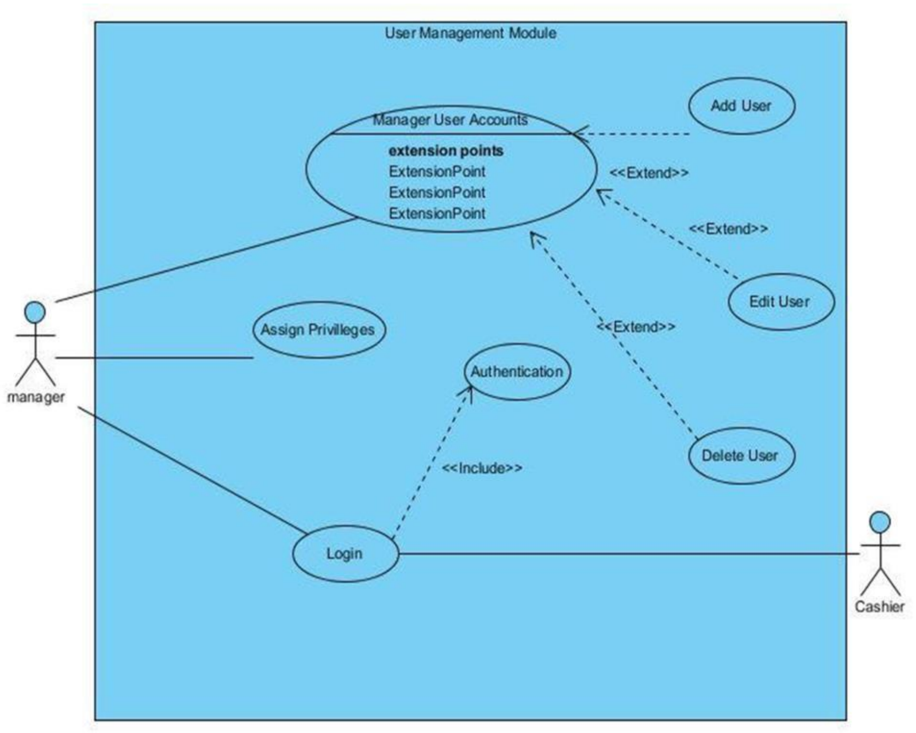


Figure 3. 1 Use case diagram for user management

Use case	Login
Actor	All Users(Manager, Administrator, Receptionist, Front desk clerk, Accountant, Billing clerk)
Over view	
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 event	
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

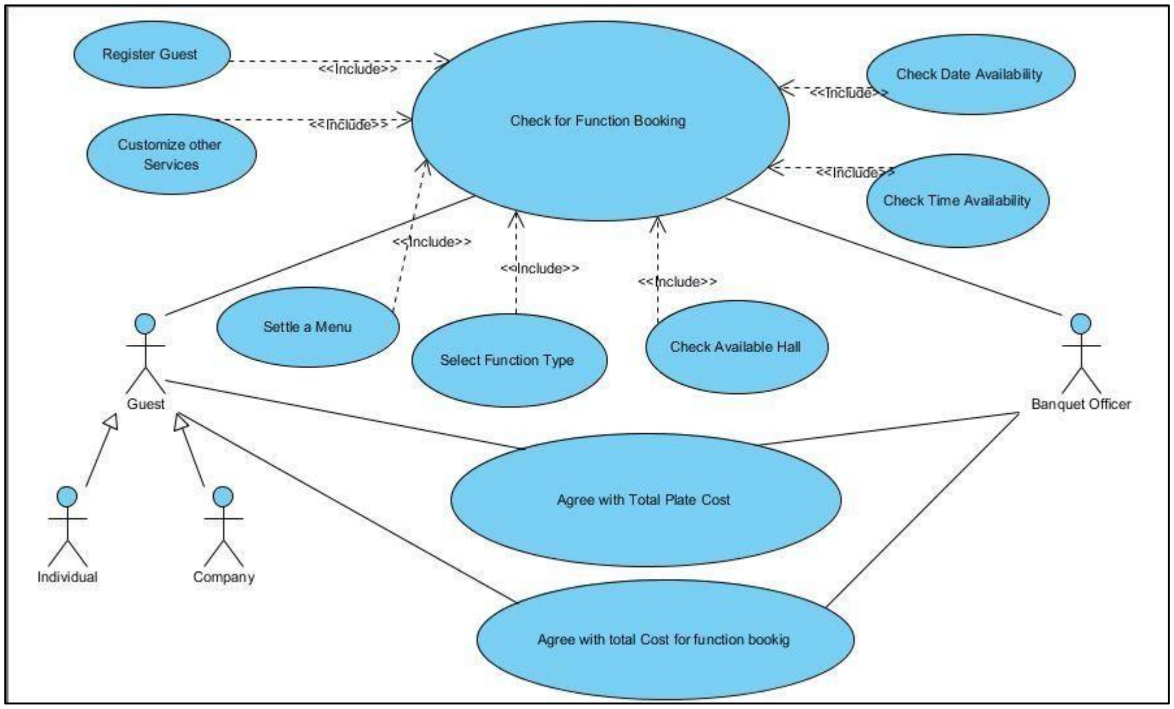


Figure 3. 2 Use case diagram for Function Booking Management

Use case	Function Booking Management
Actor	Banquet Officer, customer
Over view	
Record Function Booking Details	
Precondition(s)	
Customer should give relevant details about him / herself, Function Date & Time, Function Type, Selected Menu and Services etc…	
Flow of event	

Banquet officer fills the forms generated by the system, then auto 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

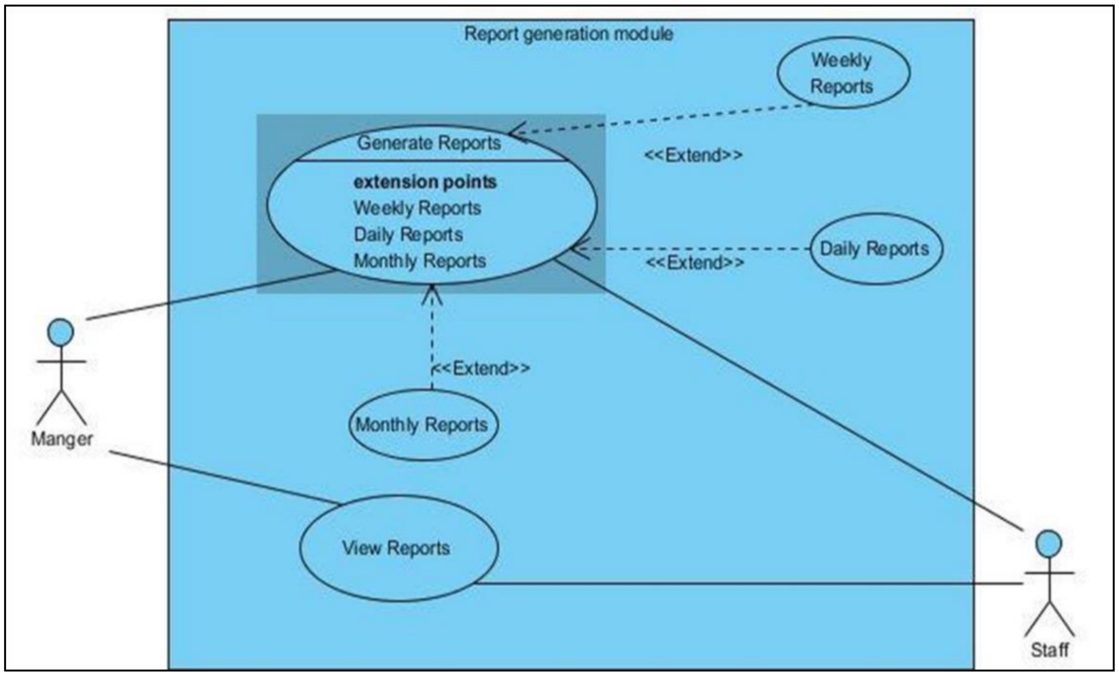


Figure 3. 3 Use case diagram for report generation

Use case	View reports
Actor	Manager and other users (staff)
Over view	
Adding details about report criteria.	

Precondition(s)
Manager should enter relevant information.
Flow of event
Manager selects the relevant criteria to generate reports.
Post Conditions
Create the relevant report.

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.

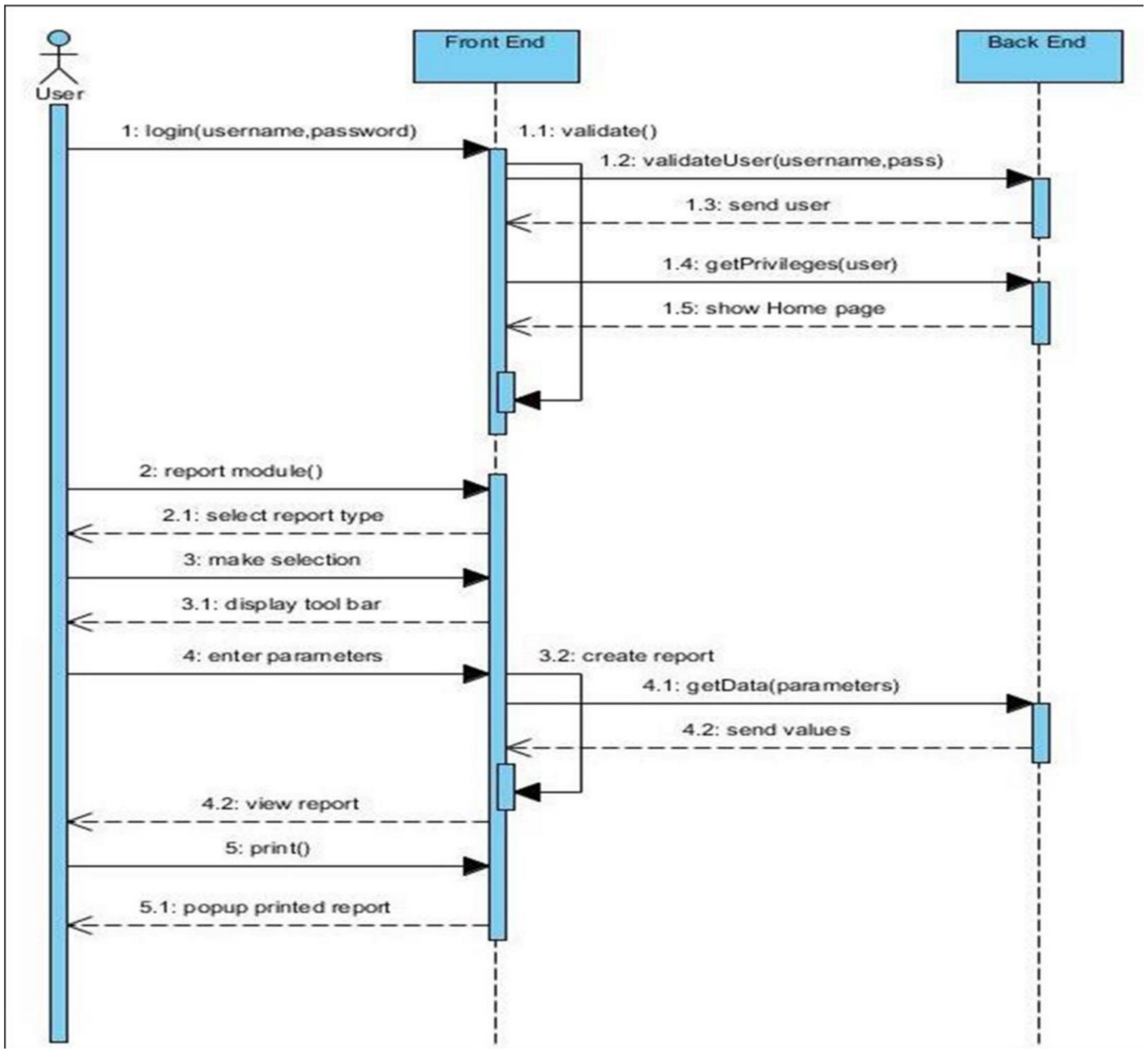


Figure 3. 4 Sequence diagram for Report Generation

3.2.3 Activity diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

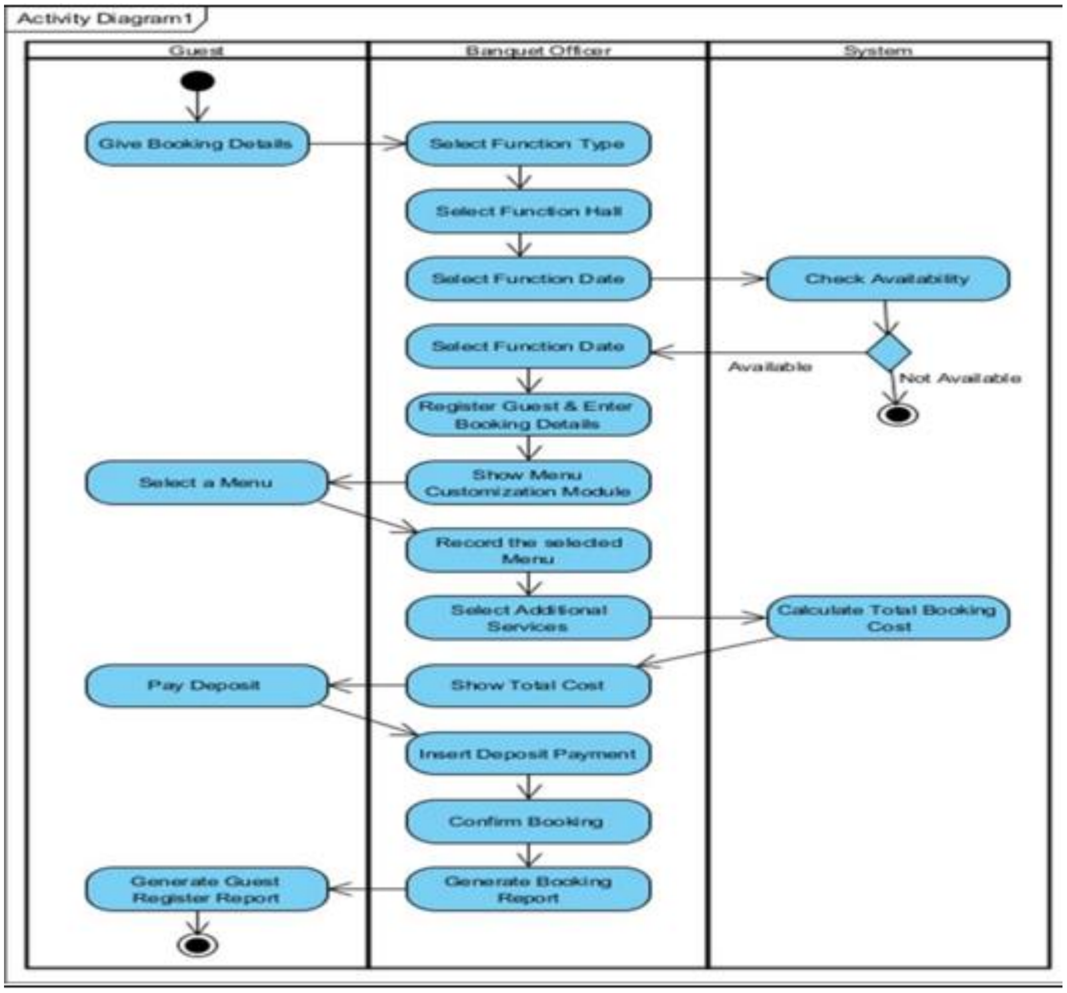


Figure 3. 5 Activity Diagram for Function Booking Management

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).

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

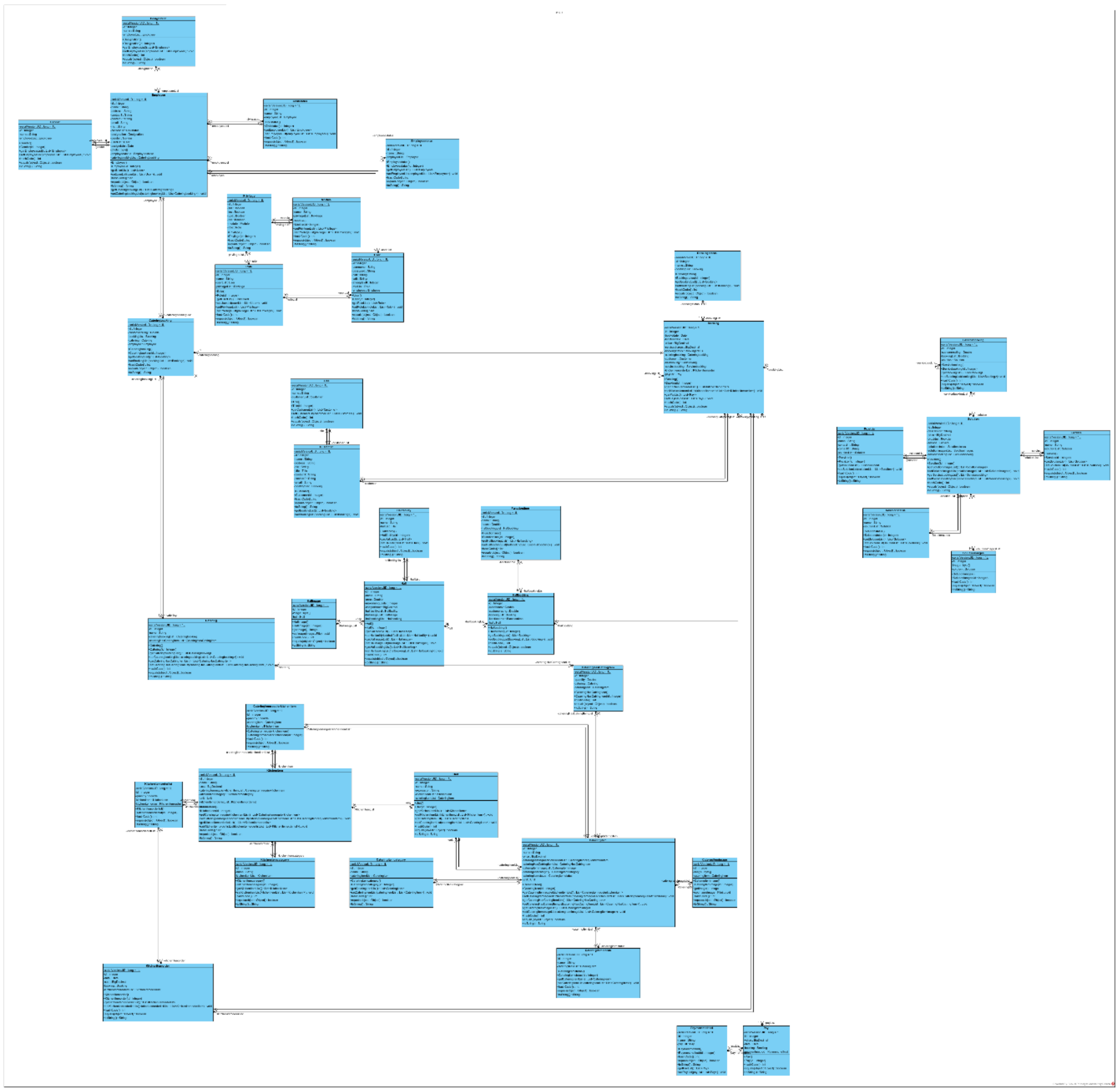


Figure 3. 6 Class diagram for the system

3.3 DATABASE DESIGNING FOR THE SYSTEM

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 [4].

When designing the database model data anomalies can be occurred. To avoid this problem database is normalized up to 3rd Normal Form. Following is a brief description of Normal Forms.

- First Normal Form (1NF):

  Every attribute value in the relation must be single, atomic value

- Second Normal Form (2NF)

  Main objective of the 2NF is eliminating the partial dependencies and creating separate tables and relate tables with a foreign key.

- Third Normal Form (3NF)

  Relation should be in second normal form and it contains no transitive dependencies.

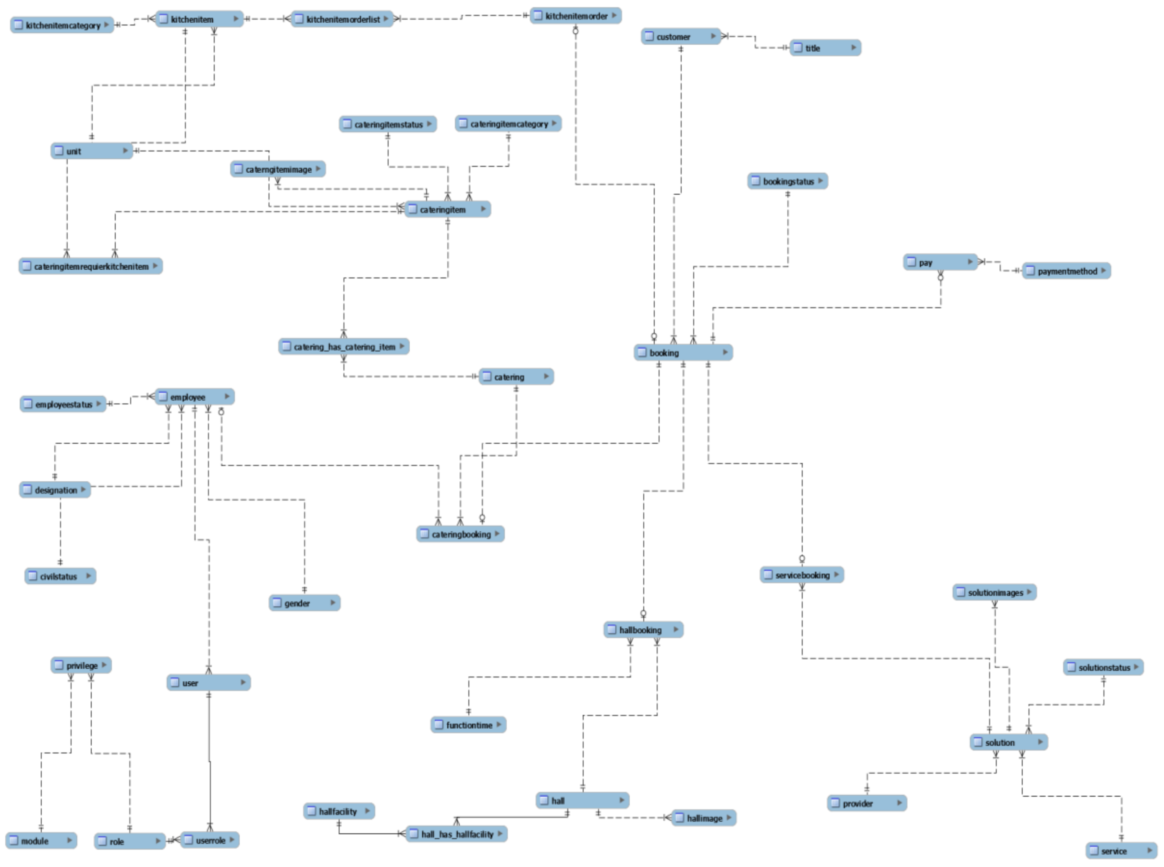


Figure 3. 7 Database design for the system

3.4 USER INTERFACE DESIGN

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

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

3.4.1 Login Interface

The figure 3.8 shows the user interface which will be used for login to the system for enter the system any user have to give valid user name & a password. If user satisfies the authentication test requirements he/she will allow log into the system. Although, if user could not provide the correct user name & password in three times the login form will stuck for a time.

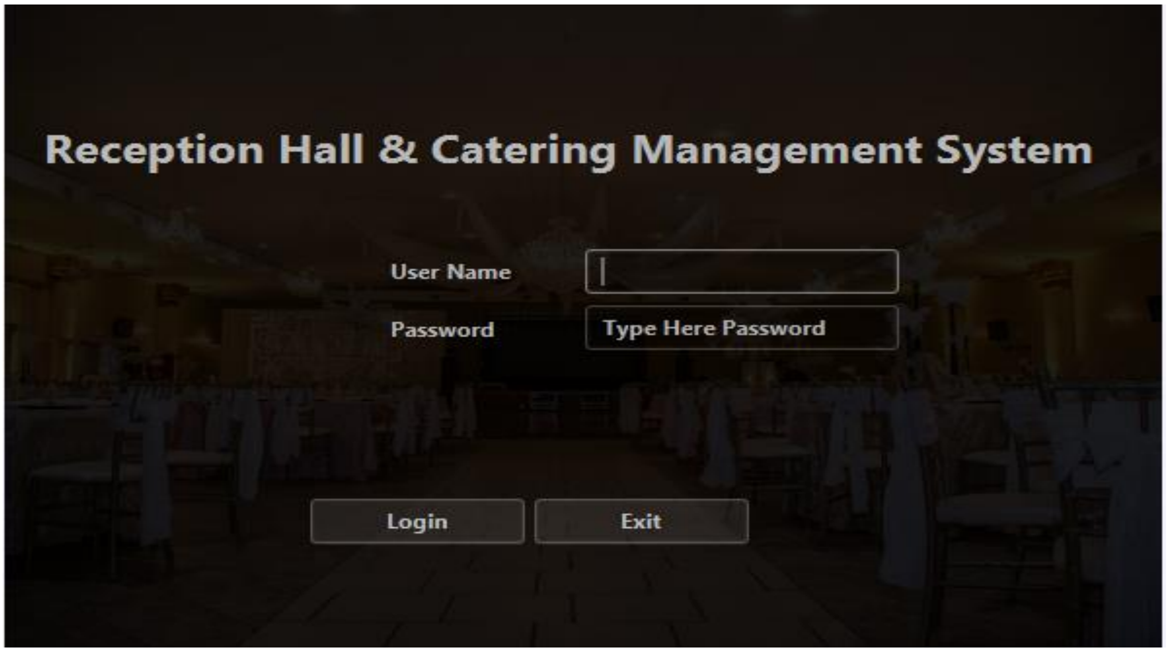
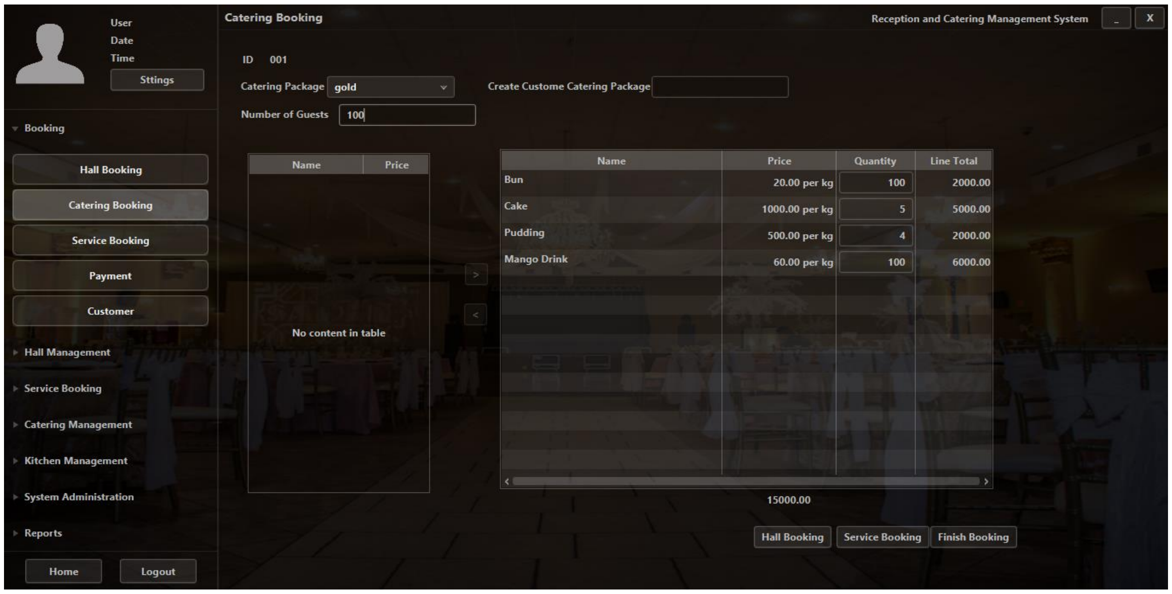


Figure 3. 8 Login User Interface

3.4.2 Catering Booking Interface

Figure 3.9 represent the Catering Booking Management Module which can make bookings. This interface provide facility to catering packages, number of guests and also customer can create custom catering package.



Catering Booking

ID 001

Catering Package: gold

Create Custome Catering Package:

Number of Guests: 100

Name	Price
No content in table	

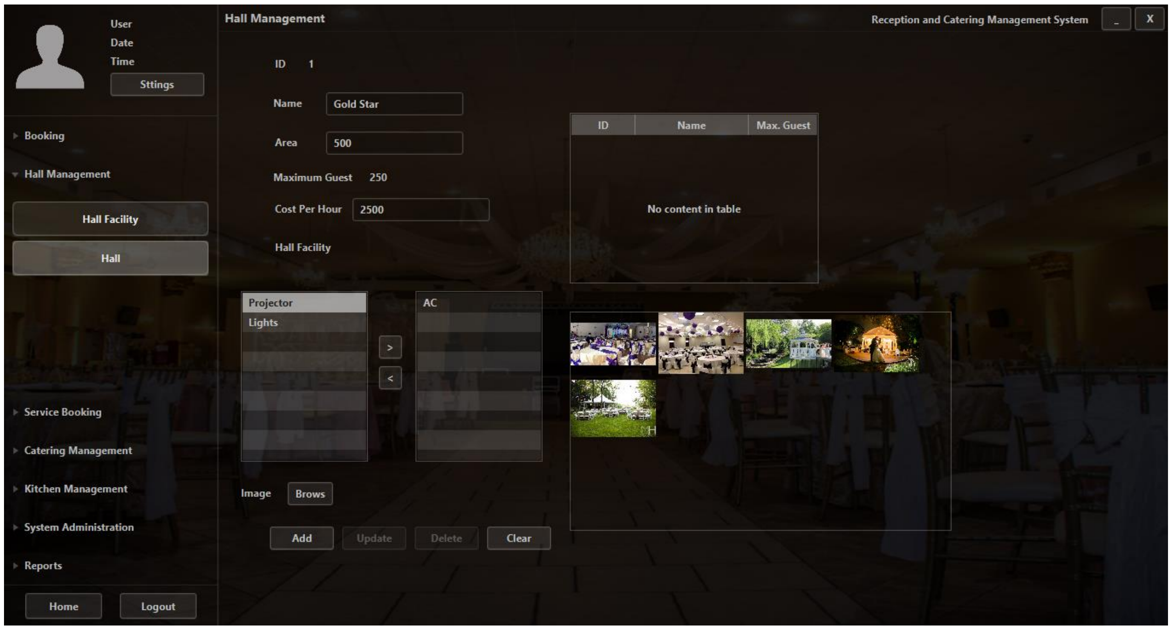
Name	Price	Quantity	Line Total
Bun	20.00 per kg	100	2000.00
Cake	1000.00 per kg	5	5000.00
Pudding	500.00 per kg	4	2000.00
Mango Drink	60.00 per kg	100	6000.00

15000.00

Figure 3. 9 Catering Booking interface

3.4.3 Hall Management Interface

Figure 3.10 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 hour and hall images.



Hall Management

ID 1

Name: Gold Star

Area: 500

Maximum Guest 250

Cost Per Hour: 2500

Hall Facility

Projector	AC
Lights	

Image

ID	Name	Max. Guest
No content in table		

Figure 3. 10 Hall Management interface

3.4.4 Photo Enlarger interface

Figure 3.11 represent the photo enlarger interface which can enlarge the image what customer want to see. When there are many photos, this interface helps to watch photos one by one.

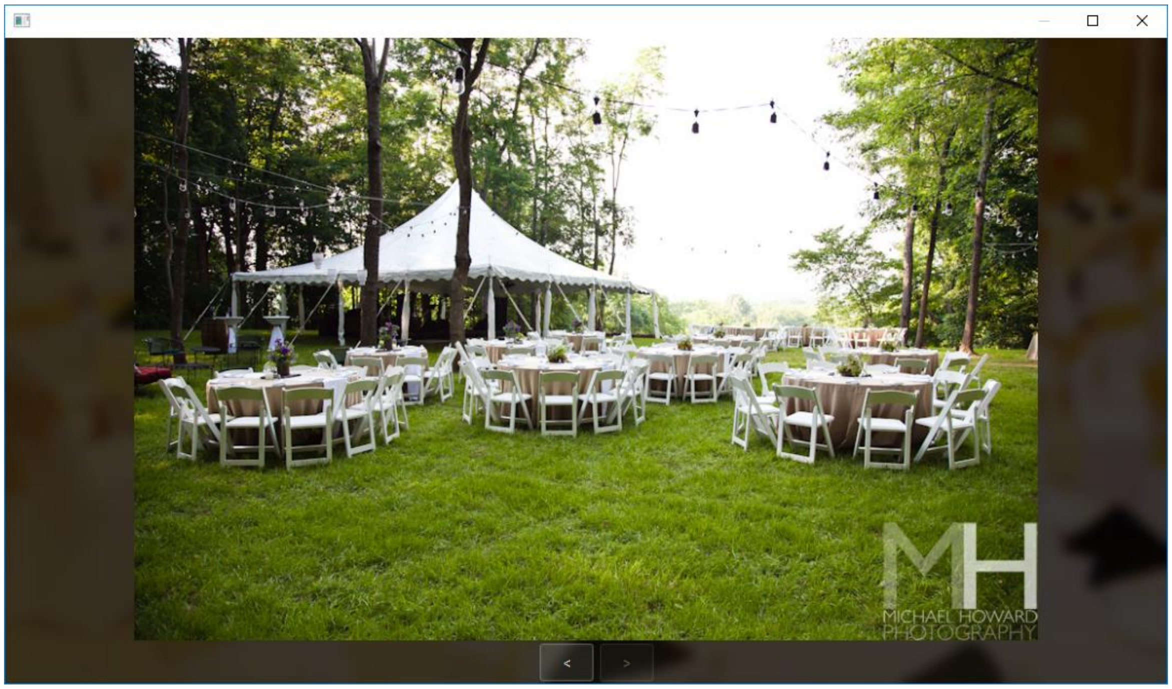
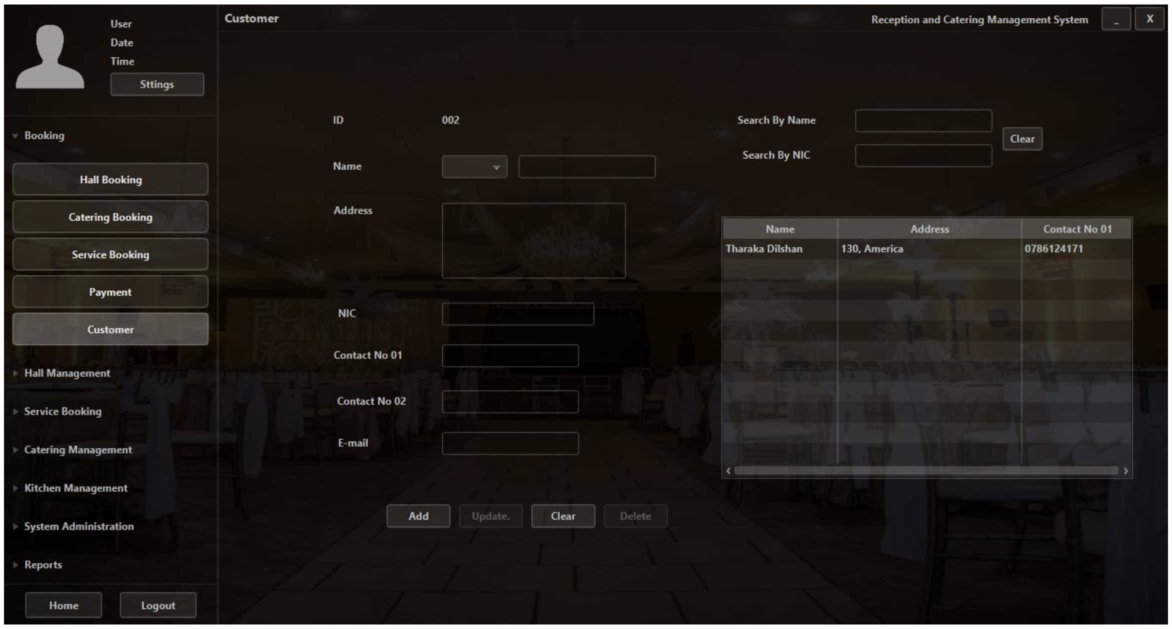


Figure 3. 11 Photo Enlarger interface

3.4.5 Customer Register Interface

Figure 3.12 represent the customer Register Form which can save customer information and find information. One can find information by name or NIC using search criteria.



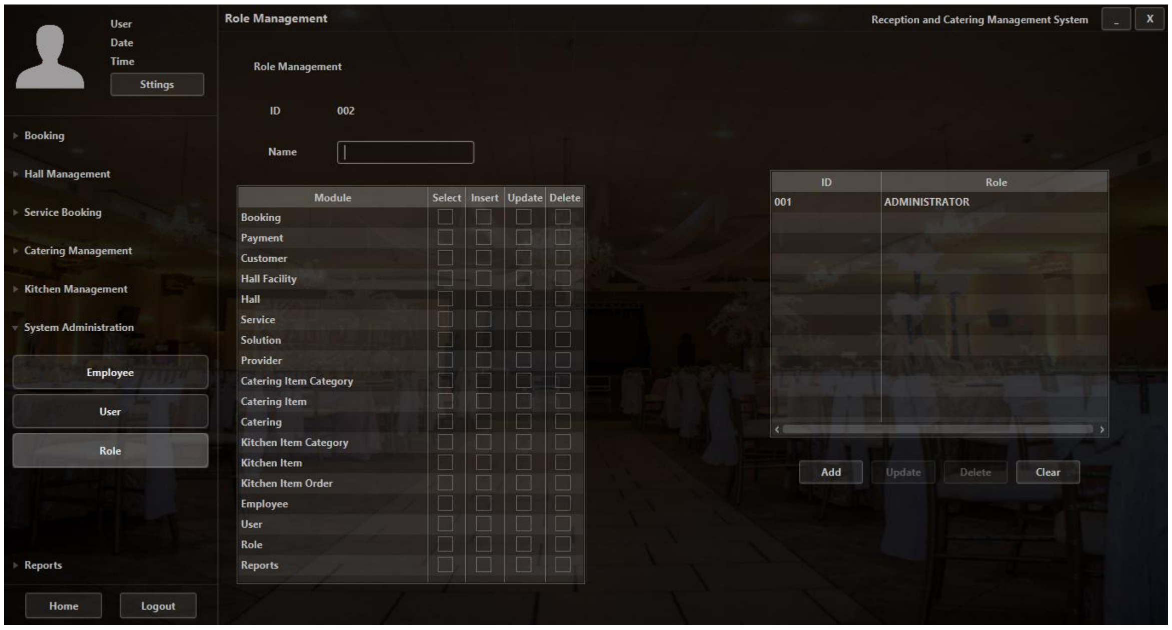
ID 002

Name	Address	Contact No 01
Tharaka Dilshan	130, America	0786124171

Figure 3. 12 Customer Management interface

3.4.6 Role Management Interface

Figure 3.13 the Role management interface which manager/owner can give the privileges to employees, to handle the modules which they can access.



ID 002

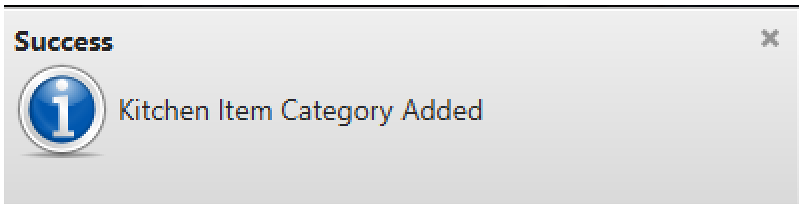
Module	Select	Insert	Update	Delete
Booking	☐	☐	☐	☐
Payment	☐	☐	☐	☐
Customer	☐	☐	☐	☐
Hall Facility	☐	☐	☐	☐
Hall	☐	☐	☐	☐
Service	☐	☐	☐	☐
Solution	☐	☐	☐	☐
Provider	☐	☐	☐	☐
Catering Item Category	☐	☐	☐	☐
Catering Item	☐	☐	☐	☐
Catering	☐	☐	☐	☐
Kitchen Item Category	☐	☐	☐	☐
Kitchen Item	☐	☐	☐	☐
Kitchen Item Order	☐	☐	☐	☐
Employee	☐	☐	☐	☐
User	☐	☐	☐	☐
Role	☐	☐	☐	☐
Reports	☐	☐	☐	☐

ID	Role
001	ADMINISTRATOR

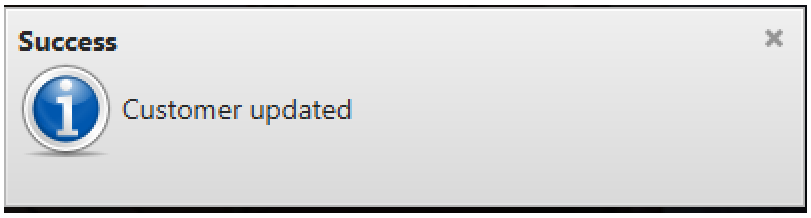
Figure 3. 13 Role Management interface

3.4.7 System Notifications

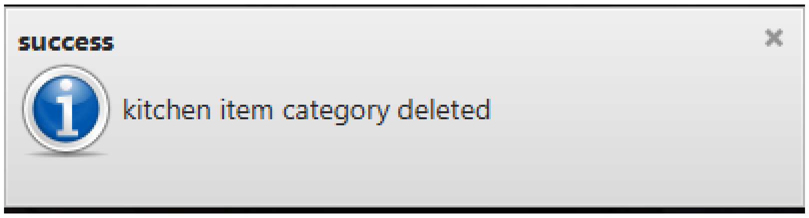
After saving, updating and deleting record, every form should show the successful message. Figure 3.14 shows the messages using to inform saving, updating, deleting of record


Success

Kitchen Item Category Added


Success

Customer updated


success

kitchen item category deleted

Figure 3. 14 Notifications

CHAPTER 04 – IMPLEMENTATION

4.1 INTRODUCTION

After completion 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.

4.2 IMPLEMENTATION ENVIRONMENT

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

4.2.1 Hardware Requirements

- Intel Core i3 Processor 2.50GHz
- RAM – 4GB or above
- Hard disk – 160GB or above

4.2.2 Software requirements

- MySQL Workbench 6.3 CE
- MySQL Server 5.5
- NetBeans IDE 8.2
- JavaFX Scene Builder 8.0
- Visual Paradigm 14.2

- Java JDK 1.8.0_102
- MySQL Query Browser 1.2.17

4.3 DEVELOPMENT TOOLS

4.3.1 NetBeans 8.2

Netbeans 8.2 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.

There are many features in NetBeans 8.2 that facilitate to easy and speedy development in software. Design GUIs for applications quickly and smoothly by using editors and drag-and-drop tools in the IDE. The NetBeans Editor indents lines, matches words and brackets, and highlights source code syntactically and semantically. It also provides code templates, coding tips, and refactoring tools. [5]

4.3.2 Java Language

Java is a computer programming language which was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. It was initially called “Oak” but was renamed “Java” in 1995. It is easy to use, reliable, secure and platform independent. Java is used in a wide variety of computing platforms spanning from embedded devices and mobile phones on the low end to enterprise servers and super computers on the high end.

Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another.

Java uses an automatic garbage collector to manage memory in the object lifecycle. The programmer determines when objects are created, and the Java runtime is responsible for recovering the memory once objects are no longer in use. Once no references to an object remain, the unreachable memory becomes eligible to be freed automatically by the garbage collector.

4.3.3 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 [6].

4.3.4 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.

4.3.5 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 [7].”

4.3.6 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 [8].

Architecture of hibernate framework is given in Figure 4.1

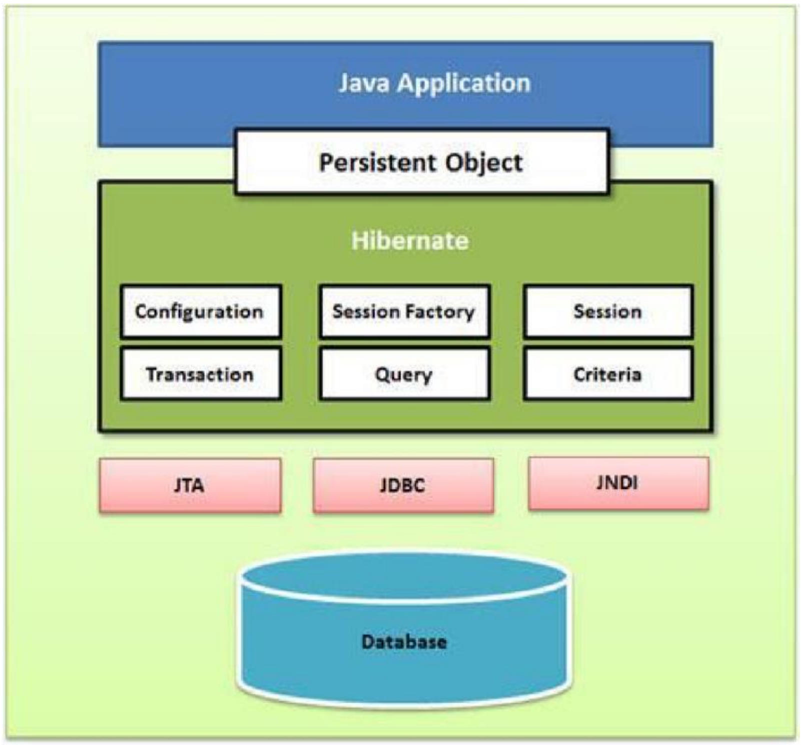


Figure 4. 1 Architecture of Hibernate Framework

4.3.7 Jasper Report

Jasper Report is an open source reporting library that can be embedded into any Java application. It provides the necessary features to generate dynamic reports, including data retrieval using JDBC (Java Database Connectivity), as well as support for parameters, expressions, variables, and groups.

It also includes advanced features, such as custom data sources, script lets, and sub reports. It has a flexible report layout and can present data textually or graphically. It is capable of exporting reports to a variety of formats and developers can supply data in multiple ways.

4.3.8 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 [9].

4.4 SYSTEM IMPLEMENTATION

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.

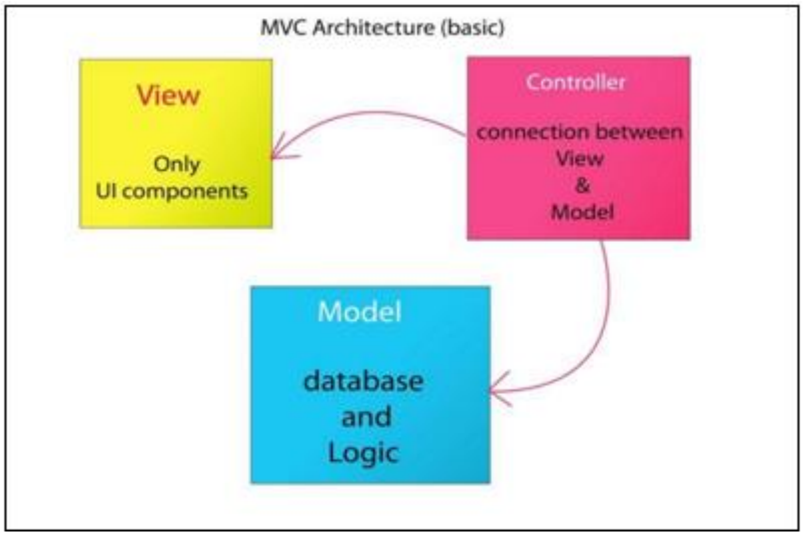


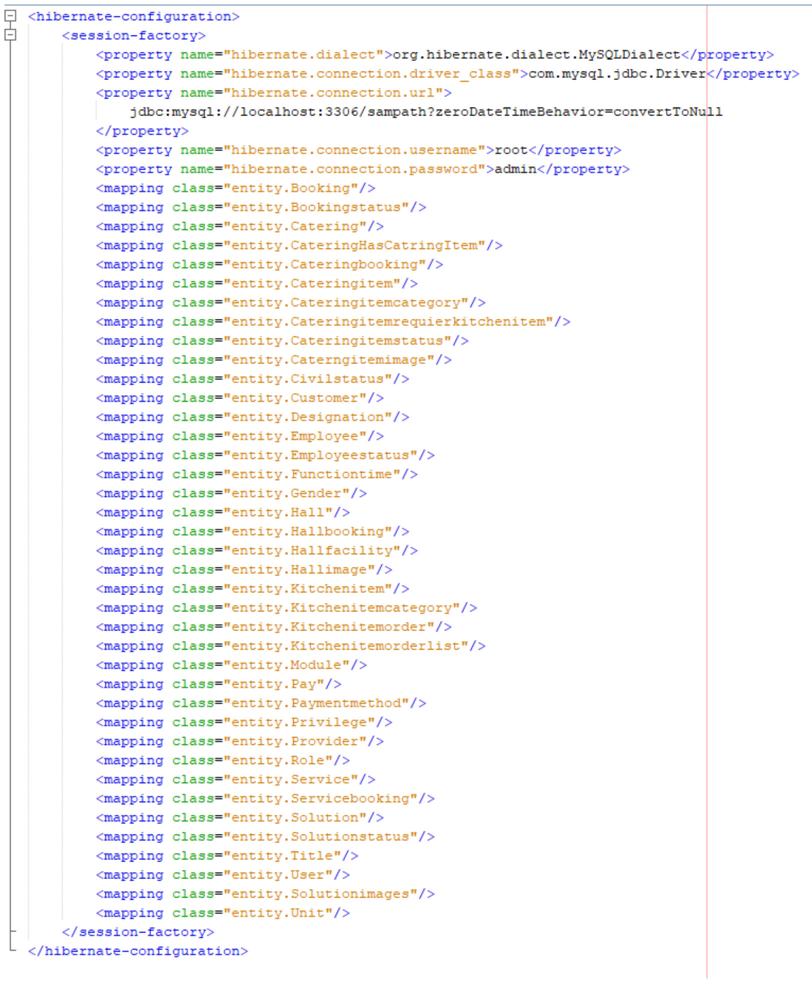
Figure 4. 2 MVC Architecture

4.5 DATA (MODEL) 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.

4.5.1 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.

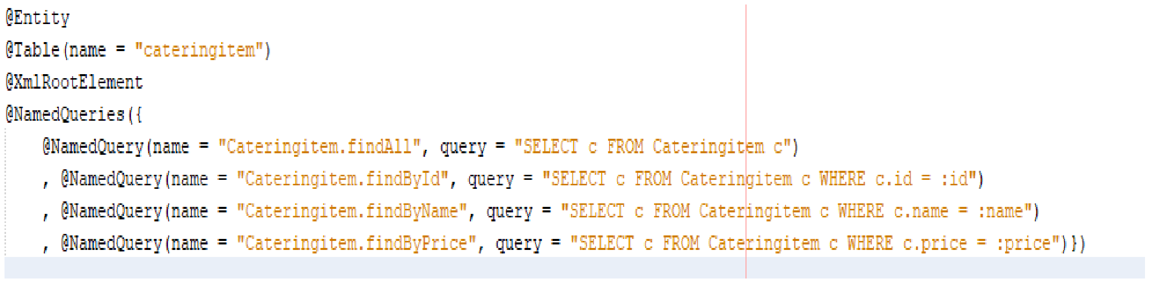


```xml
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.url">
            jdbc:mysql://localhost:3306/sampath?zeroDateTimeBehavior=convertToNull
        </property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">admin</property>
        <mapping class="entity.Booking"/>
        <mapping class="entity.Bookingstatus"/>
        <mapping class="entity.Catering"/>
        <mapping class="entity.CateringHasCatringItem"/>
        <mapping class="entity.Cateringbooking"/>
        <mapping class="entity.Cateringitem"/>
        <mapping class="entity.Cateringitemcategory"/>
        <mapping class="entity.Cateringitemrequierkitchenitem"/>
        <mapping class="entity.Cateringitemstatus"/>
        <mapping class="entity.Caterngitemimage"/>
        <mapping class="entity.Civilstatus"/>
        <mapping class="entity.Customer"/>
        <mapping class="entity.Designation"/>
        <mapping class="entity.Employee"/>
        <mapping class="entity.Employeestatus"/>
        <mapping class="entity.Functiontime"/>
        <mapping class="entity.Gender"/>
        <mapping class="entity.Hall"/>
        <mapping class="entity.Hallbooking"/>
        <mapping class="entity.Hallfacility"/>
        <mapping class="entity.Hallimage"/>
        <mapping class="entity.Kitchenitem"/>
        <mapping class="entity.Kitchenitemcategory"/>
        <mapping class="entity.Kitchenitemorder"/>
        <mapping class="entity.Kitchenitemorderlist"/>
        <mapping class="entity.Module"/>
        <mapping class="entity.Pay"/>
        <mapping class="entity.Paymentmethod"/>
        <mapping class="entity.Privilege"/>
        <mapping class="entity.Provider"/>
        <mapping class="entity.Role"/>
        <mapping class="entity.Service"/>
        <mapping class="entity.Servicebooking"/>
        <mapping class="entity.Solution"/>
        <mapping class="entity.Solutionstatus"/>
        <mapping class="entity.Title"/>
        <mapping class="entity.User"/>
        <mapping class="entity.Solutionimages"/>
        <mapping class="entity.Unit"/>
    </session-factory>
</hibernate-configuration>
```

Code 4.1 hibernate.cfg.xml

4.5.2 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.

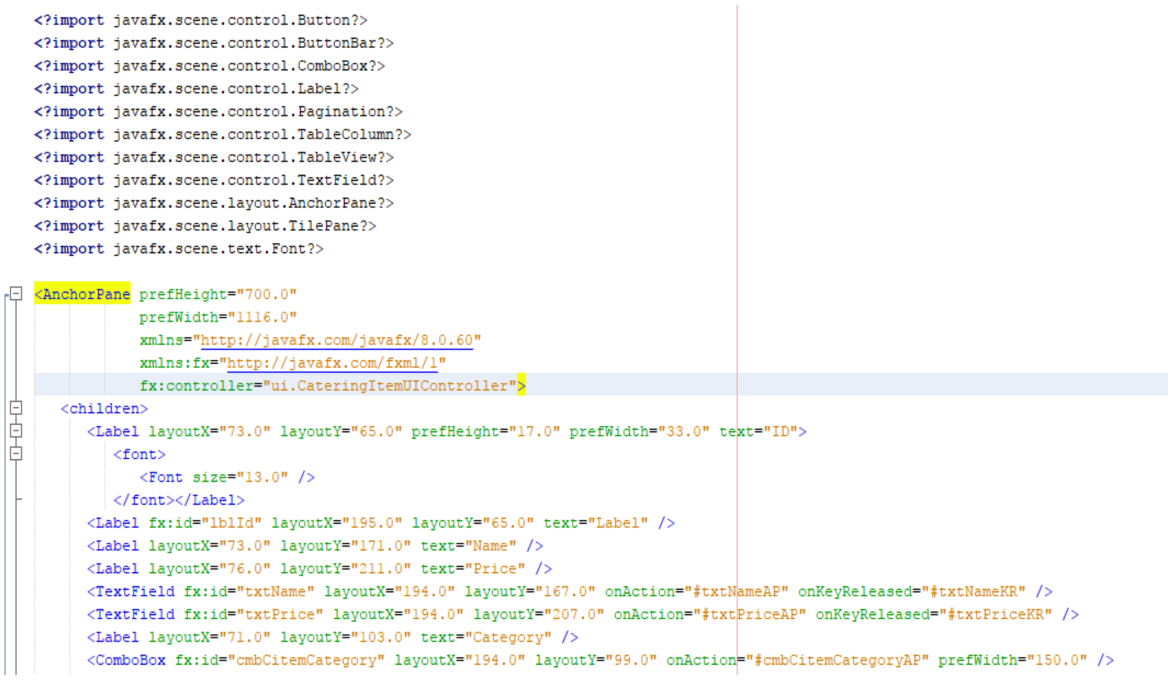


```java
@Entity
@Table(name = "cateringitem")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Cateringitem.findAll", query = "SELECT c FROM Cateringitem c")
    , @NamedQuery(name = "Cateringitem.findById", query = "SELECT c FROM Cateringitem c WHERE c.id = :id")
    , @NamedQuery(name = "Cateringitem.findByName", query = "SELECT c FROM Cateringitem c WHERE c.name = :name")
    , @NamedQuery(name = "Cateringitem.findByPrice", query = "SELECT c FROM Cateringitem c WHERE c.price = :price")})
```

Code 4.2 Cateringitem.java

4.6 VIEW (INTERFACE) LAYER IMPLEMENTATION

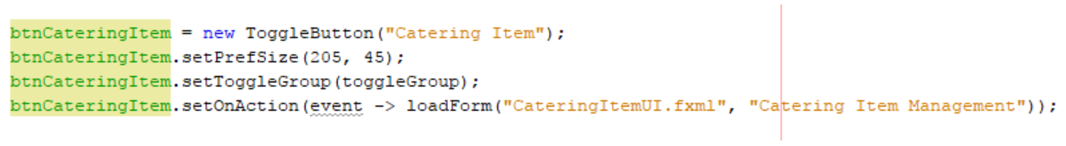
Interface is the layer that interrelates with the user & it is a representation of the logic behind in the database. Interfaces simplify users to do required operations using forms as well as loading views. JavaFX Scene Builder designer view was used for the construction of the user interfaces and to 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.



```xml
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ButtonBar?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Pagination?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.TilePane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="700.0"
            prefWidth="1116.0"
            xmlns="http://javafx.com/javafx/8.0.60"
            xmlns:fx="http://javafx.com/fxml/1"
            fx:controller="ui.CateringItemUIController">
   <children>
      <Label layoutX="73.0" layoutY="65.0" prefHeight="17.0" prefWidth="33.0" text="ID">
         <font>
            <Font size="13.0" />
         </font></Label>
      <Label fx:id="lblId" layoutX="195.0" layoutY="65.0" text="Label" />
      <Label layoutX="73.0" layoutY="171.0" text="Name" />
      <Label layoutX="76.0" layoutY="211.0" text="Price" />
      <TextField fx:id="txtName" layoutX="194.0" layoutY="167.0" onAction="#txtNameAP" onKeyReleased="#txtNameKR" />
      <TextField fx:id="txtPrice" layoutX="194.0" layoutY="207.0" onAction="#txtPriceAP" onKeyReleased="#txtPriceKR" />
      <Label layoutX="71.0" layoutY="103.0" text="Category" />
      <ComboBox fx:id="cmbCitemCategory" layoutX="194.0" layoutY="99.0" onAction="#cmbCitemCategoryAP" prefWidth="150.0" />
```

Code 4.3 CateringItem.fxml

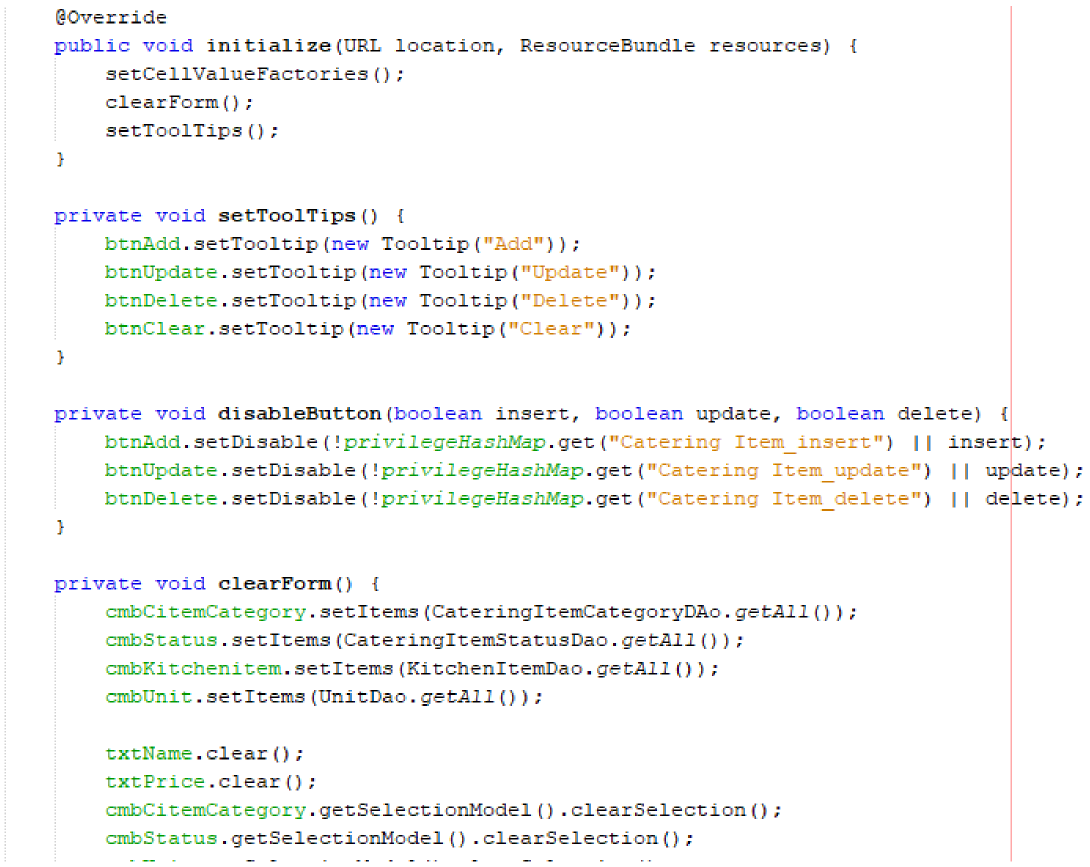
Following shows the java codes use for load the above CateringItemUI.fxml into the Java Application class.



```
btnCateringItem = new ToggleButton("Catering Item");
btnCateringItem.setPrefSize(205, 45);
btnCateringItem.setToggleGroup(toggleGroup);
btnCateringItem.setOnAction(event -> loadForm("CateringItemUI.fxml", "Catering Item Management"));
```

Code 4.4 Cateringitem.java

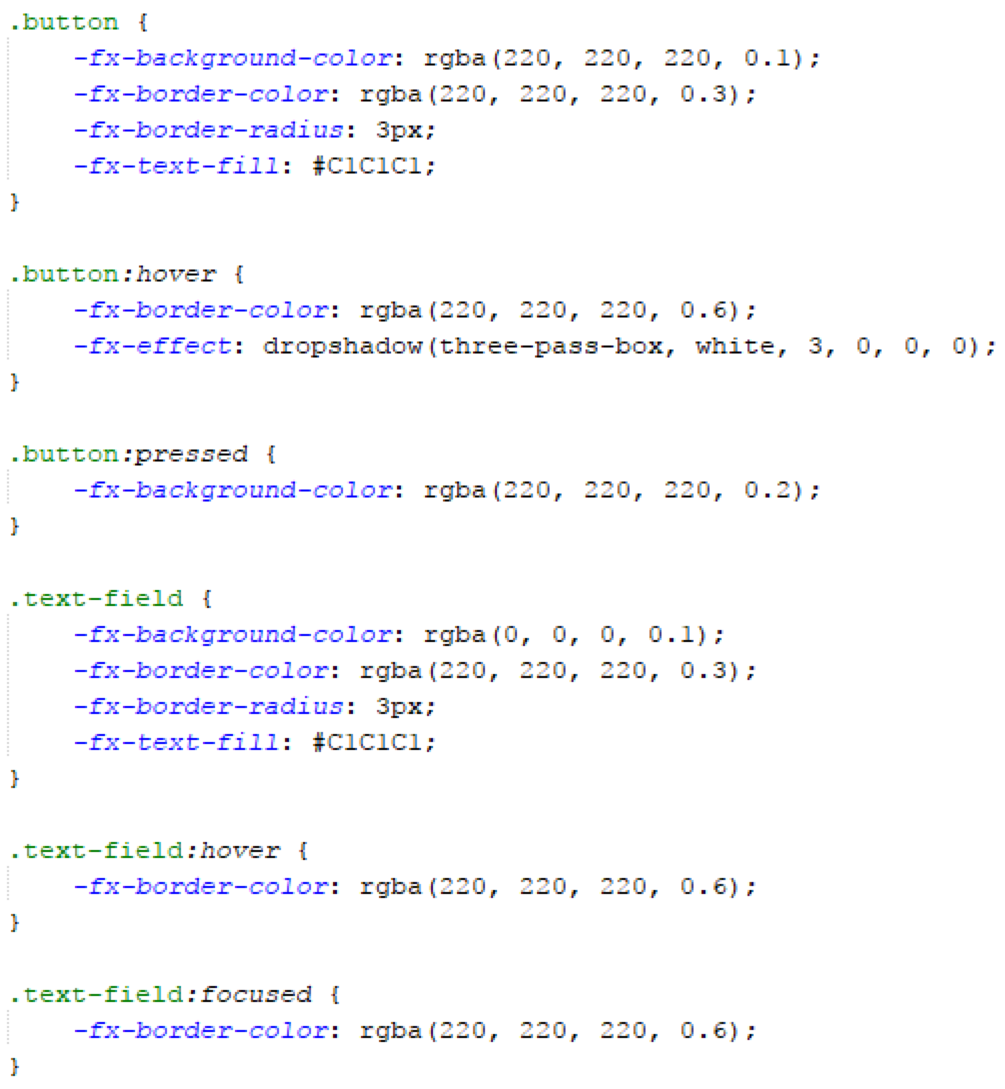
Following represents the java code segment which use for the controller class (Interface Controller) for above interface.



```
@Override
public void initialize(URL location, ResourceBundle resources) {
    setCellValueFactories();
    clearForm();
    setToolTips();
}

private void setToolTips() {
    btnAdd.setTooltip(new Tooltip("Add"));
    btnUpdate.setTooltip(new Tooltip("Update"));
    btnDelete.setTooltip(new Tooltip("Delete"));
    btnClear.setTooltip(new Tooltip("Clear"));
}

private void disableButton(boolean insert, boolean update, boolean delete) {
    btnAdd.setDisable(!privilegeHashMap.get("Catering Item_insert") || insert);
    btnUpdate.setDisable(!privilegeHashMap.get("Catering Item_update") || update);
    btnDelete.setDisable(!privilegeHashMap.get("Catering Item_delete") || delete);
}

private void clearForm() {
    cmbCitemCategory.setItems(CateringItemCategoryDAo.getAll());
    cmbStatus.setItems(CateringItemStatusDao.getAll());
    cmbKitchenitem.setItems(KitchenItemDao.getAll());
    cmbUnit.setItems(UnitDao.getAll());

    txtName.clear();
    txtPrice.clear();
    cmbCitemCategory.getSelectionModel().clearSelection();
    cmbStatus.getSelectionModel().clearSelection();
```

Code 4.5 CateringitemController.java

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



```
.button {
    -fx-background-color: rgba(220, 220, 220, 0.1);
    -fx-border-color: rgba(220, 220, 220, 0.3);
    -fx-border-radius: 3px;
    -fx-text-fill: #C1C1C1;
}

.button:hover {
    -fx-border-color: rgba(220, 220, 220, 0.6);
    -fx-effect: dropshadow(three-pass-box, white, 3, 0, 0, 0);
}

.button:pressed {
    -fx-background-color: rgba(220, 220, 220, 0.2);
}

.text-field {
    -fx-background-color: rgba(0, 0, 0, 0.1);
    -fx-border-color: rgba(220, 220, 220, 0.3);
    -fx-border-radius: 3px;
    -fx-text-fill: #C1C1C1;
}

.text-field:hover {
    -fx-border-color: rgba(220, 220, 220, 0.6);
}

.text-field:focused {
    -fx-border-color: rgba(220, 220, 220, 0.6);
}
```

Code 4.6 Style.css

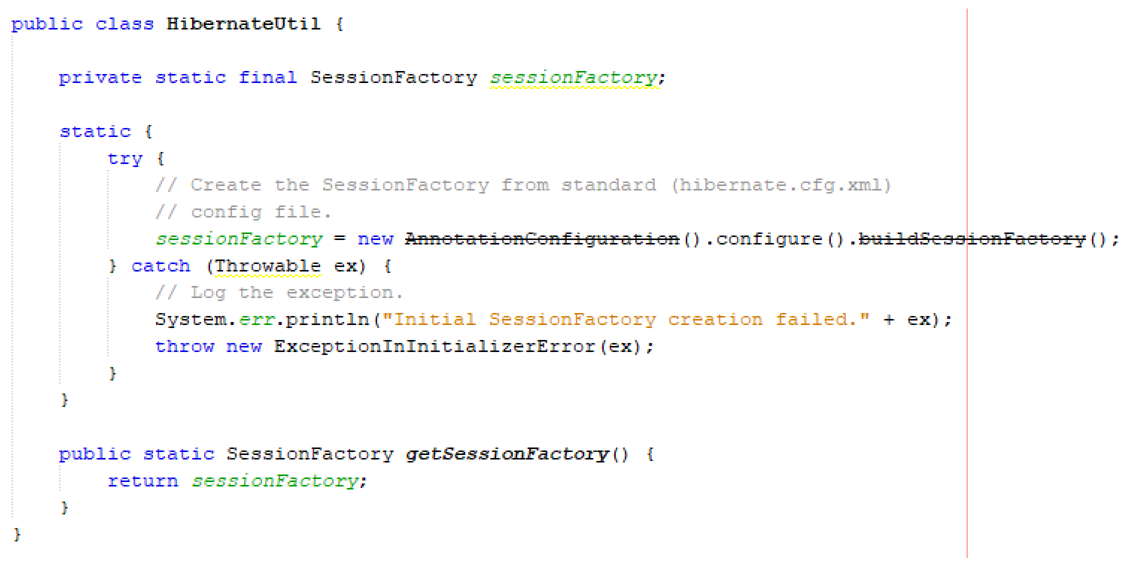
4.7 CONTROLLER (DAO) LAYER IMPLEMENTATION

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.

4.7.1 Hibernate Sessions

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.

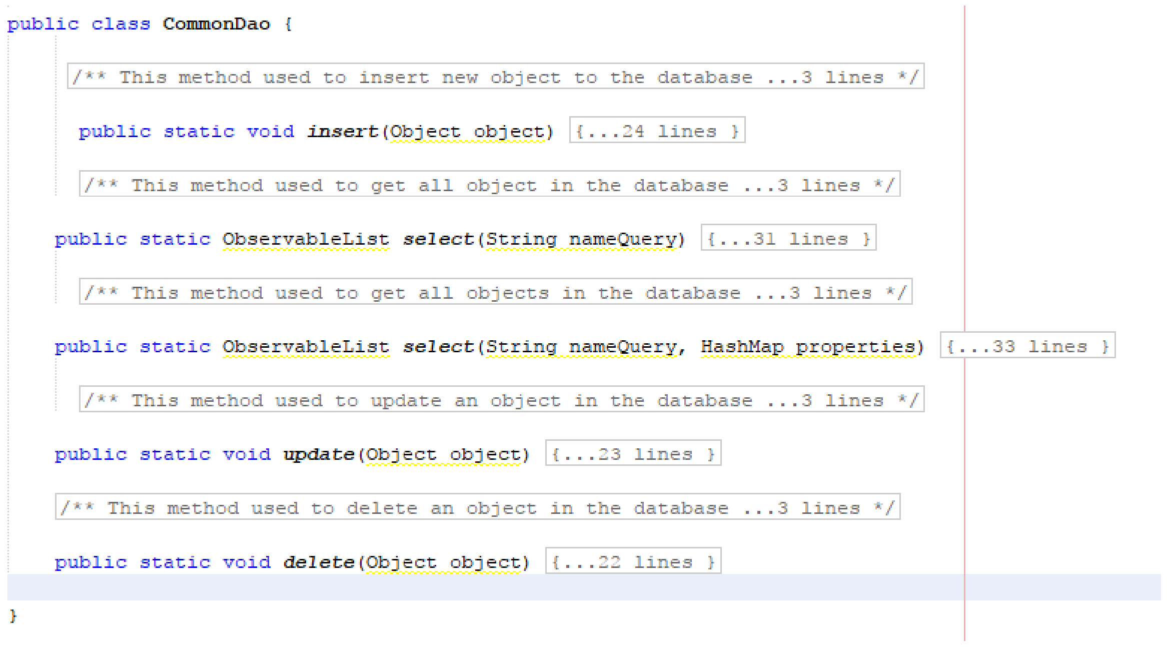


```
public class HibernateUtil {

    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

4.7.2 DAO (Data Access Objects)

A data access object (DAO) is an object that provides an abstract interface to some type of database or persistence mechanism and providing some specific operations without exposing details of the database. It provides a mapping from application to the persistence layer. This isolation separates the concerns of what data accesses the application needs, in terms of domain-specific objects and data types (the public interface of the DAO), and how these needs can be satisfied with a specific DBMS, database schema, etc. (the implementation of the DAO).



```
public class CommonDao {

      /** This method used to insert new object to the database ...3 lines */

       public static void insert(Object object) {...24 lines }

       /** This method used to get all object in the database ...3 lines */

    public static ObservableList select(String nameQuery) {...31 lines }

       /** This method used to get all objects in the database ...3 lines */

    public static ObservableList select(String nameQuery, HashMap properties) {...33 lines }

       /** This method used to update an object in the database ...3 lines */

    public static void update(Object object) {...23 lines }

    /** This method used to delete an object in the database ...3 lines */

    public static void delete(Object object) {...22 lines }

}
```

Code 4.8: CommnDao.java

4.8 ACKNOWLEDGEMENT OF REUSED CODE MODULES

- Jasper Reports

  "Jasper Reports" 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-separated values or XML files. It can be used in Java-enabled applications, including JavaEE or web applications, to generate dynamic content.

- 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.

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.

In software testing verification and validation is the process of checking that a software system meets specifications and that it fulfills its intended purpose. Verification is process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Validation checks that the product design satisfies or fits the intended use or the software meets 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 Reception Hall Management System for the New Sampath Caterers 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 [10].

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 [11].”

5.2.1 Test cases & test results for Login Module

Test cases includes low level details about testing. All the test cases and results are included in Appendix E. Table 5.1 shows login related test cases. Login is very critical functionality when it comes to security. Login function should be tested thoroughly.

Test No	Description	Expected result	Pass/Fail
1	Login into the system with valid username & password	User should get main window of the system	Pass
2	Login into the system with invalid username & password	Should show the remaining attempts	Pass
3	Login into the system with empty username/password	Should show an error message at the login page	Pass
4	Changing password with valid passwords.	Password should get changed next login new password should work.	Pass
5	Login out from the system	After confirmation system should logout and show the login screen.	Pass

Table 5. 1 Test case and results for Login Module

5.2.2 Test cases & test results for Customer Management Module

Test cases and test results for Customer Management Module are listed in table 5.2 the table has test name, guest type, NIC no, mobile no, land no, email, address. Expected result and the actual result generated by each test case.

Test No	Description	Expected result	Pass/Fail
1	Enter digits and symbols as guest name	Text color of the text field change to red.	Pass
2	Select Combo Box	List all the Guest types to combo box	Pass
3	Enter letters and symbols as contact numbers.	Text color of the text field change to red	Pass

4	Enter all information correctly and click Add	Notification box will be shown as “Customer added succeed” and added data will be shown at the table.	Pass
5	Without entering data to some fields and click Add	Employee should not get created. Database record should not get added. Show a red star before the relevant text field that did not enter the data.	Pass
6	Select a row from the table	Selected Customer details will be load into the relevant Text Fields. Add button will be disabled and Update, Delete buttons will be enabled	Pass
7	Enter Text value in the “Search by Name” field	Customer table will refresh according to typed text values	Pass
8	Enter Text value in the “Search by NIC” field	Customer table will refresh according to typed text values	Pass
9	Click Clear button	Clear text Fields data and clear selection of the table	Pass

Table 5. 2 Test case and results for Customer Management Module

5.2.3 Test cases & test results for User module

Test cases and test results for User module are listed in table 5.3 the table has test no, description, expected result and the actual result generated by each test case.

Test No	Description	Expected result	Pass/Fail
1	Select Combo Box “Employee”	List all the employees to combo box	Pass
2	Confirm Password should equal to Password	Error Message box will be shown as “Passwords don’t match”	Pass

3	Select “>>” button	All the roles view in available list will be removed and it will be added into assigned list	Pass
4	Select “<<” button	All the roles view in assigned list will be removed and it will be added into available list	Pass
5	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

Table 5. 3 Test cases and results for User module

5.2.4 Test cases & test results for Employee module

Employee entity is one of the most important entities in this system. Create, retrieve, update and delete functions were tested extensively. Table 5.4 describes employee related test cases.

Test No	Description	Expected result	Pass/Fail
1	Create employee with valid information.	Notification box will be shown as “Employee added succeed” and added data will be shown at the table.	Pass
2	Create employee with invalid information.	Employee should not get created. Database record should not get added. Text of text filed of invalid information, should change to red.	Pass
3	Update employee with proper information	Employee details should get changed. System should return a success notification.	Pass
4	Employee search and filtering	Employee should be filtered out.	Pass

5	Deleting of an employee	Delete confirmation should return. Once confirmed employee should get deleted.	Pass

Table 5. 4 Test cases and results for Employee module

5.2.5 Test cases & test results for Catering Item Management module

Test cases and test results for catering Item Management Module are listed in table 5.5 the table has test Catering Item category, Units, Inner Table for kitchen item, expected result and the actual result generated by each test case.

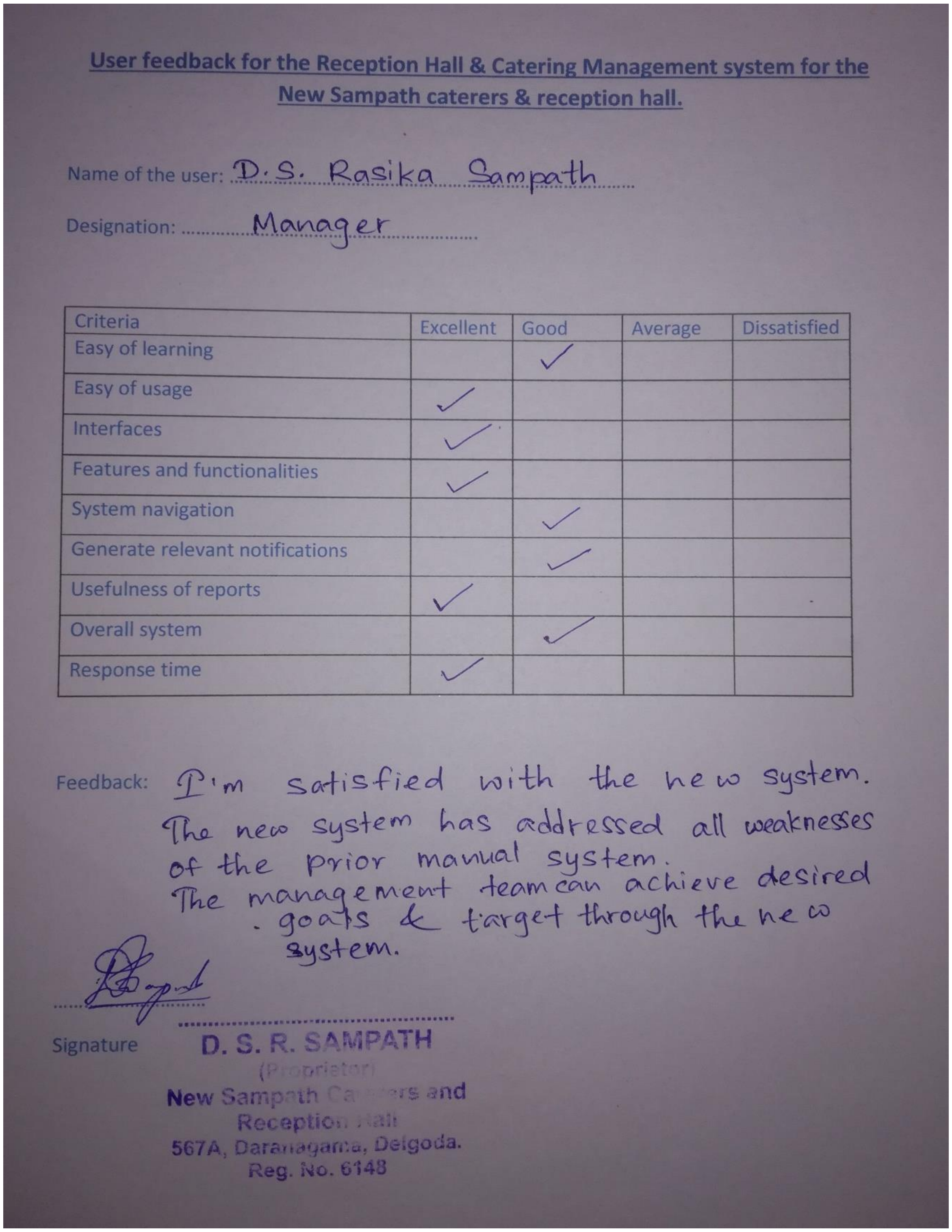
Test No	**Description**	**Expected result**	**Pass/Fail**
1	Select Combo Box	List all the Catering Item Category to Combo Box	Pass
2	Select Combo Box	List all the Units to Combo Box	Pass
3	Select Kitchen Item and enter quantity then click add button in inner table	Fill the inner table with quantity and selected Kitchen Items	Pass
4	Select a row of the table	Fill the relevant text fields with inner table	Pass

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

5.3 USER EVALUATION

Normally user evaluation is done by selecting different users of the system. In this Reception Hall & Catering Management system, Manager has been taken as an administrator of the system and other users has taken as normal users with different privileges. User evaluation questionnaire was given to target population and results has summarized.

Following figure 5.1 shows user evaluation questionnaire.



User feedback for the Reception Hall & Catering Management system for the New Sampath caterers & reception hall.

Name of the user: D. S. Rasika Sampath

Designation: Manager

Criteria	Excellent	Good	Average	Dissatisfied
Easy of learning		✓		
Easy of usage	✓			
Interfaces	✓			
Features and functionalities	✓			
System navigation		✓		
Generate relevant notifications		✓		
Usefulness of reports	✓			
Overall system		✓		
Response time	✓			

Feedback: I'm satisfied with the new system.
The new system has addressed all weaknesses of the prior manual system.
The management team can achieve desired
. goals & target through the new system.

[illegible]

Signature

D. S. R. SAMPATH
(Proprietor)
New Sampath Caterers and
Reception Hall
567A, Daranagama, Delgoda.
Reg. No. 6148

Figure 5. 1 User evaluation results

CHAPTER 06 – CONCLUSION

6.1 CRITICAL ASSESSMENT OF PROJECT

New Sampath caterers and Reception hall is a well-known reception hall in Delgoda area. They need to enhance their business with new technology. Earlier they faced many problems with manual system and heavy paper works. But currently they are carrying out their function reservation and day to day activities effectively.

Earlier they faced many difficulties of bookings of hall booking, catering booking, service booking. They have to spend lot of time to handle this process. Currently because of this reception hall & catering management system, they can carry out their activities effectively and efficiently. Since all the activities have automated through this system, now they can enhance their business easily.

Since including the report generation process of this system, they can analyze their business status and forecasting about future business. Further decision making ability is increased.

During the analysis phase functional and non-functional requirements were identified in each module. After that all functional and non-functional requirements of each and every module were successfully completed in implement phase. End of the project additional functionalities and features were added which client requested. Further client was satisfied about my system finally.

6.2 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.

6.3 FUTUER IMPROVEMENTS

This proposed system was developed within a time constraint and functionalities which were agreed by the client. Following are the some of the future enhancement of the system.

- Web based 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.
- Expand the Reception Hall & Catering Management System.
- Introduced system has been developed with the possibility of any expansions such as adding more functions to the system much easily.

REFERANCES

[1] Wikipedia. Prototyping [Online], Available: https://en.wikipedia.org/wiki/Software_prototyping , [Accessed on: 10th August 2017]

[2] Wikipedia. Iterative and Incremental development [Online], Available: https://en.wikipedia.org/wiki/Iterative_and_incremental_development , [Accessed on: 10th August 2017]

[3] Wikipedia, "Systems design", [Online]. Available:

http://en.wikipedia.org/wiki/Systems_design . [Accessed: 10th August 2017].

[4] Wikipedia, "Database design", [Online]. Available:

https://en.wikipedia.org/wiki/Database_design . [Accessed: 10th August 2017].

[5] Wikipedia, "NetBeans", [Online]. Available:

https://en.wikipedia.org/wiki/NetBeans . [Accessed: 12th September 2017].

[6] Wikipedia, "MySQL", [Online]. Available:

https://en.wikipedia.org/wiki/MySQL . [Accessed: 12th September 2017].

[7] Wikipedia, "Visual Paradigm for UML", [Online]. Available:

https://en.wikipedia.org/wiki/Visual_Paradigm_for_UML . [Accessed: 12th September 2017].

[8] Wikipedia, "Hibernate (framework)", [Online]. Available:

https://en.wikipedia.org/wiki/Hibernate_(framework) . [Accessed: 12th September 2017].

[9] Wikipedia, "Java Persistence API", [Online]. Available:

https://en.wikipedia.org/wiki/Java_Persistence_API . [Accessed: 12th September 2017].

[10] Wikipedia, "System testing", [Online]. Available:

http://en.wikipedia.org/wiki/System_testing . [Accessed: 12th September 2017].

[11] Wikipedia, "Test case", [Online]. Available:

http://en.wikipedia.org/wiki/Test_case . [Accessed: 12th September 2017].

[12] A. Systems, "Hotel Management Software, Hotel PMS | Anand Systems Inc.", Anandsystems.com, [Online]. Available:

https://www.anandsystems.com/ . [Accessed: 12th September 2017].

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.

Hardware & software configuration requirements,

Hardware Requirements	Software requirements
4GB RAM	MySQL Server 5.5
160GB Hard Disk	NetBeans IDE
Intel(R) Core(TM)i3 – 4010U @ CPU 1.70GHz	MySQL Workbench MySQL Query Browser JavaFX Scene Builder Visual Paradigm

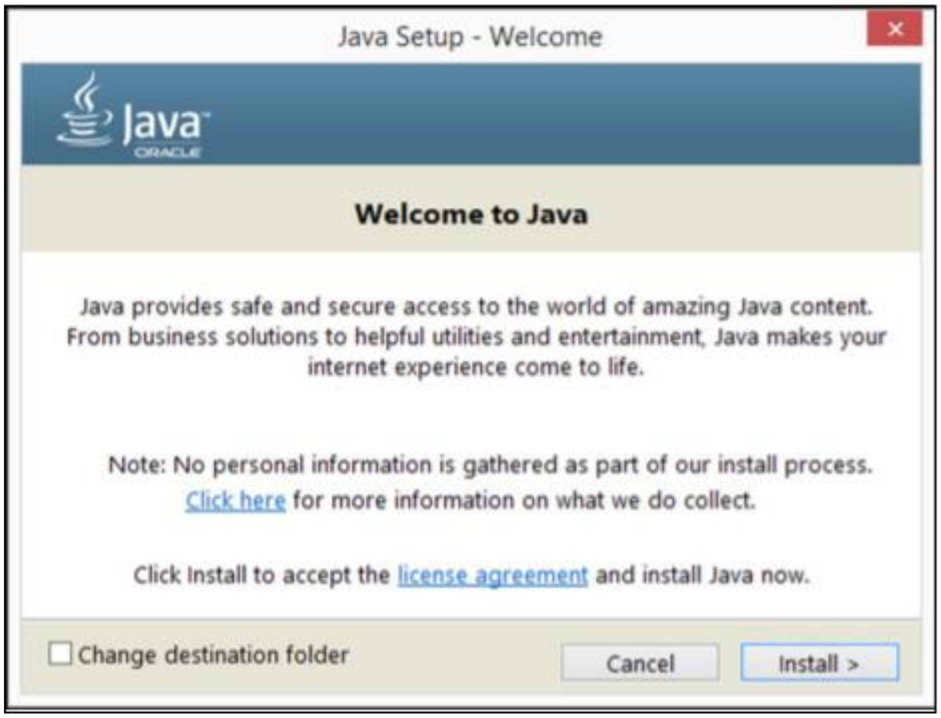
Table A. 1 Implementation Environment

A.1 SOFTWARE INSTALLATION AND CONFIGERATION

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.

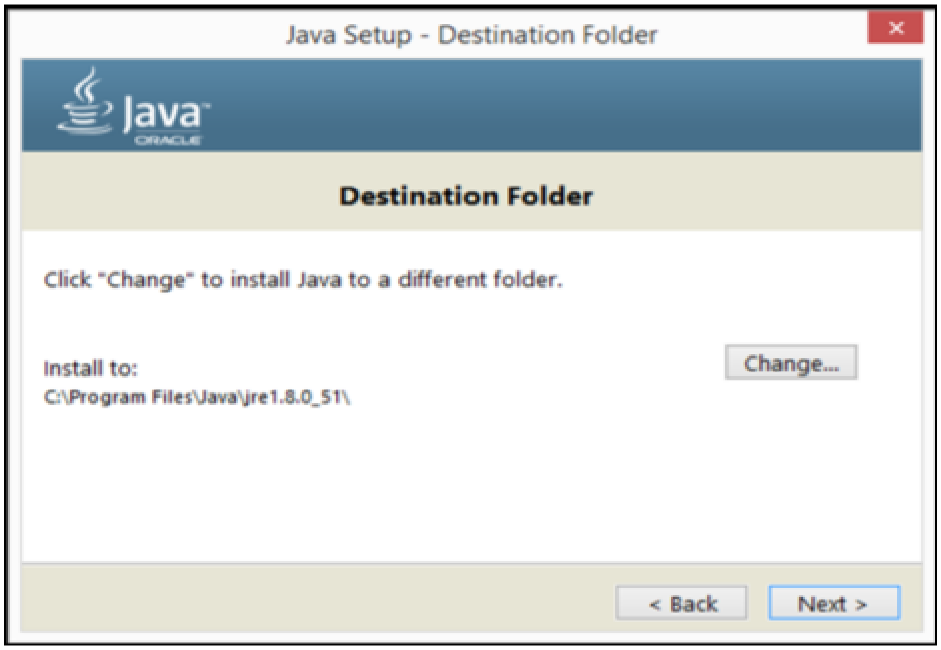
Note: No personal information is gathered as part of our install process. Click here for more information on what we do collect.

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

☐ Change destination folder

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

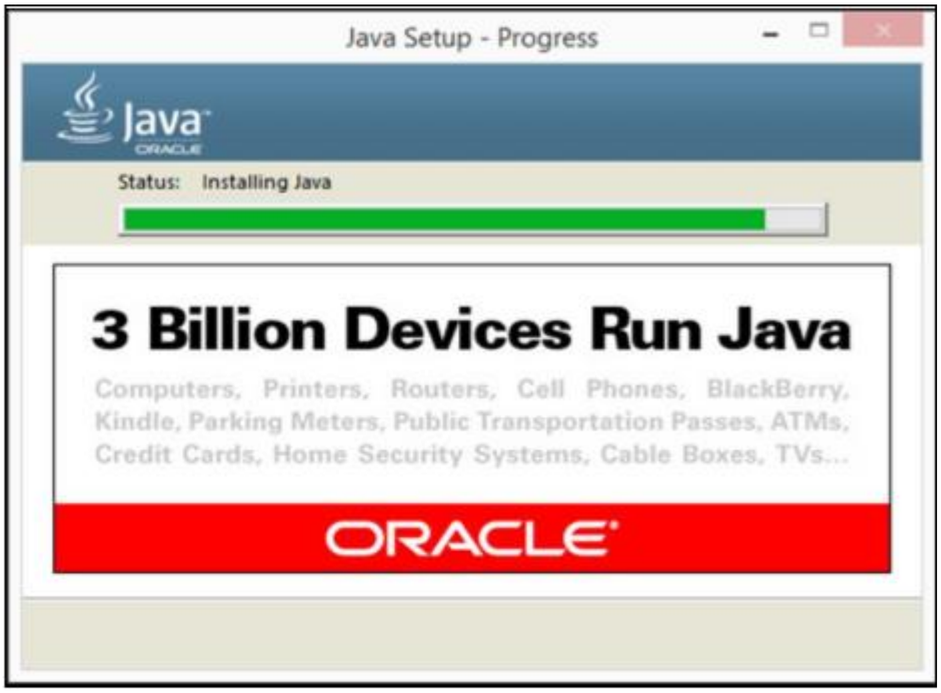
Destination Folder

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)



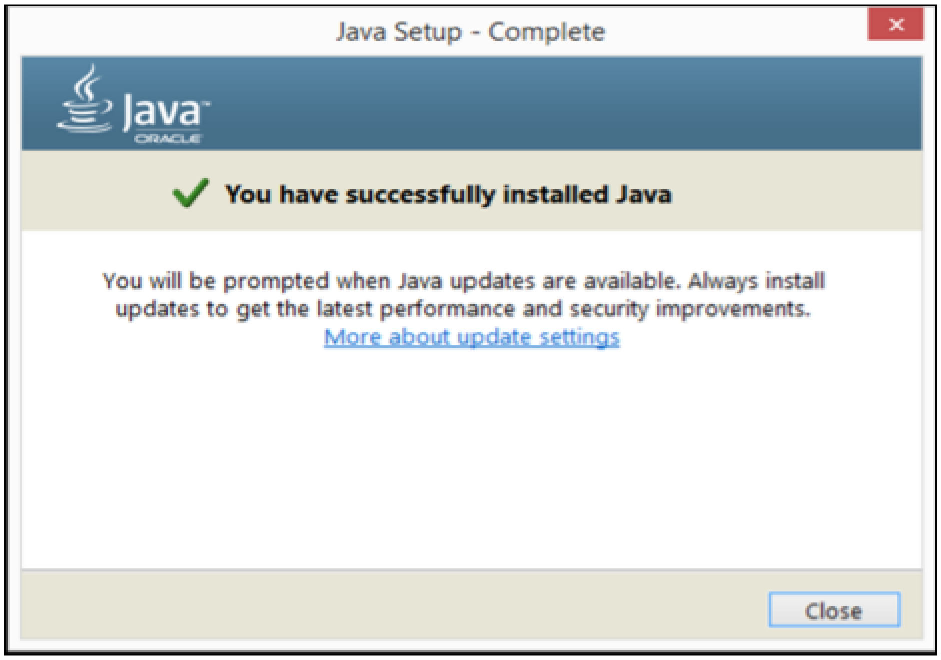
Status: Installing Java

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)



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

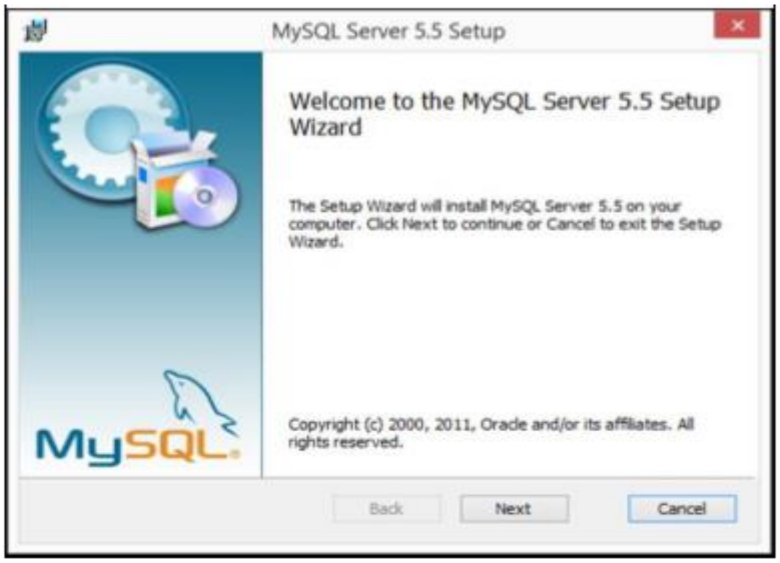
Figure A. 4 Installation Wizard of JRE (Step 4)

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)



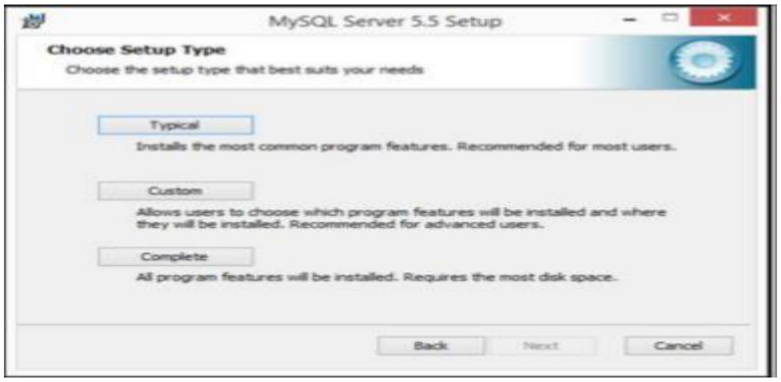
Welcome to the MySQL Server 5.5 Setup Wizard

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)



Choose Setup Type

Choose the setup type that best suits your needs

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

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

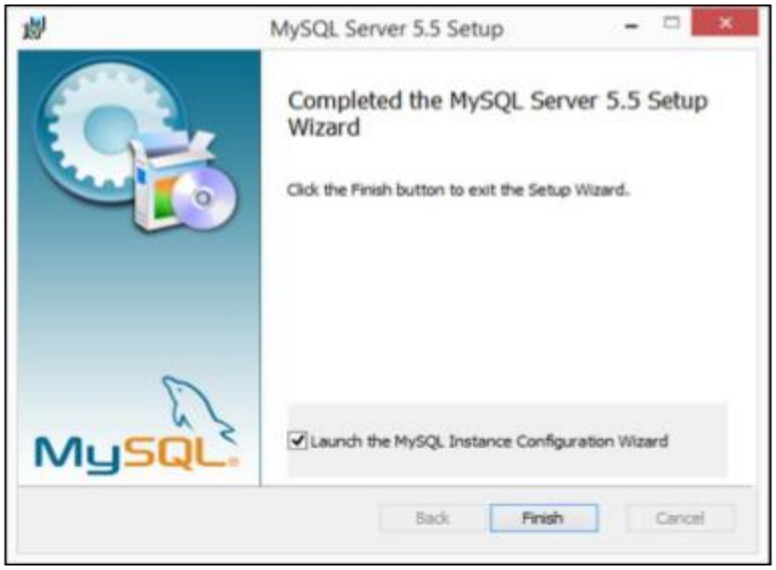
All program features will be installed. Requires the most disk space.

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".

Following figure A.7 shows completion screen in the installation wizard of MySQL Server as (Step 3)


MySQL Server 5.5 Setup

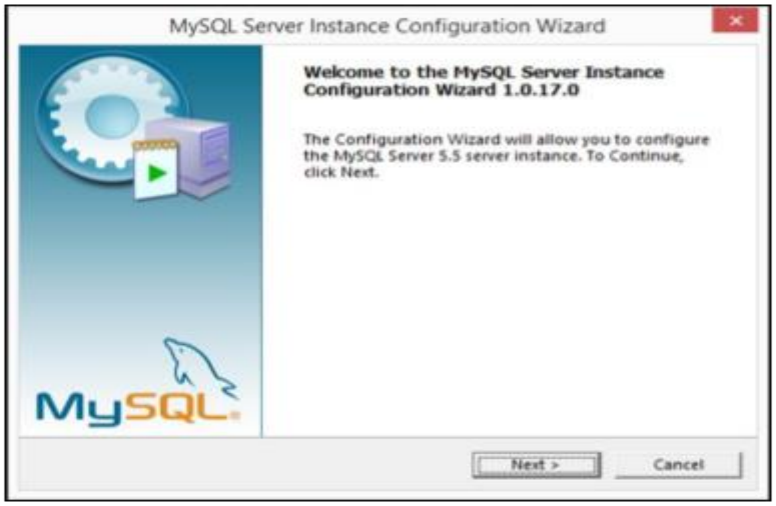
Completed the MySQL Server 5.5 Setup Wizard

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

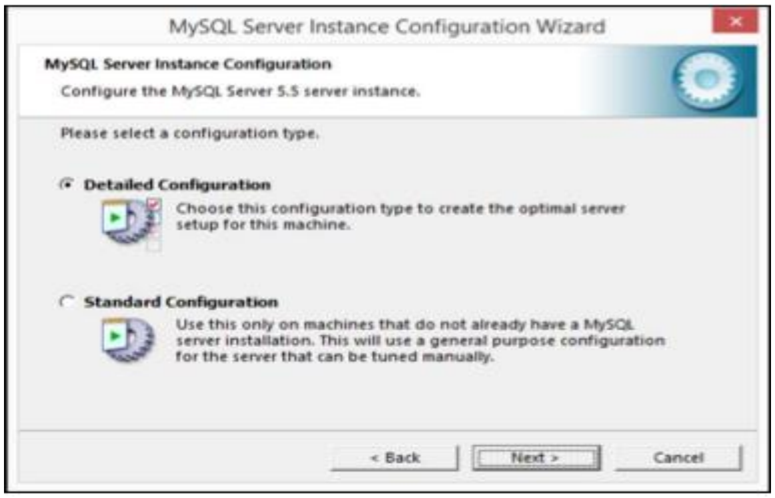
The Configuration Wizard will allow you to configure the MySQL Server 5.5 server instance. To Continue, click Next.

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**

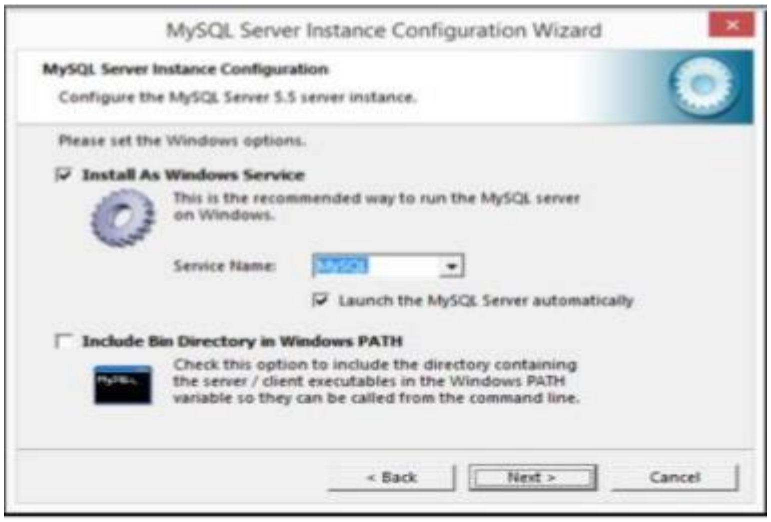
Choose this configuration type to create the optimal server setup for this machine.

( ) **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.

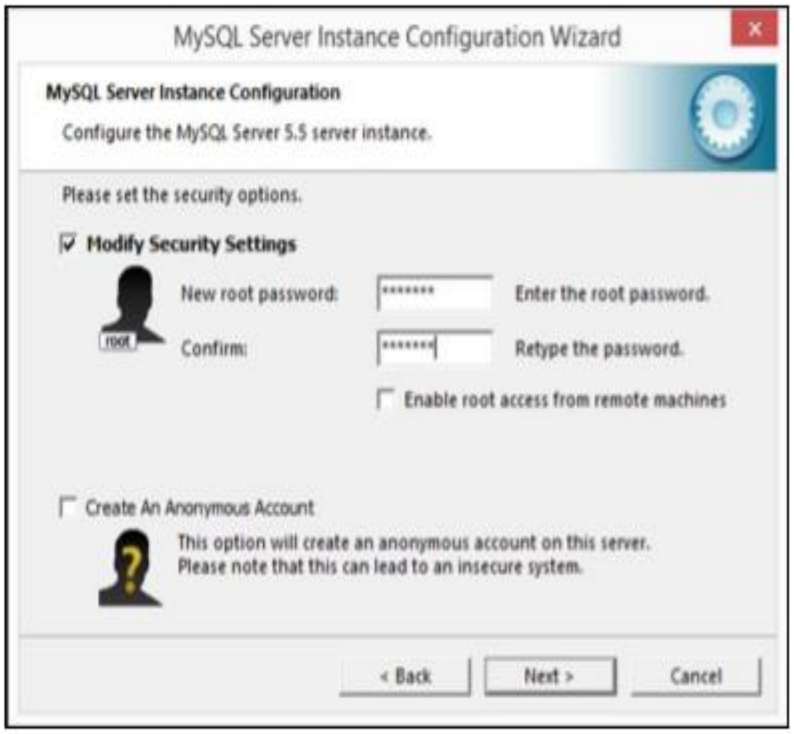
Figure A. 10 Installation Progress MySQL Server (Step 6)

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.

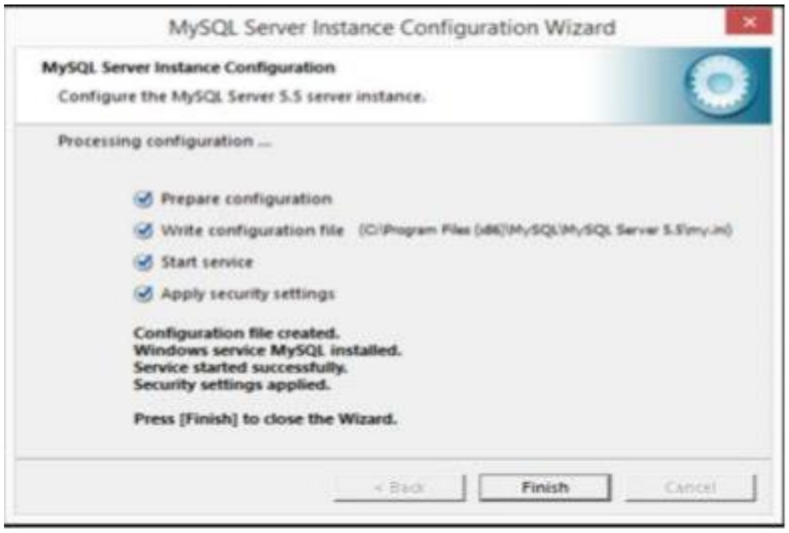
[ ] Enable root access from remote machines

[ ] 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.

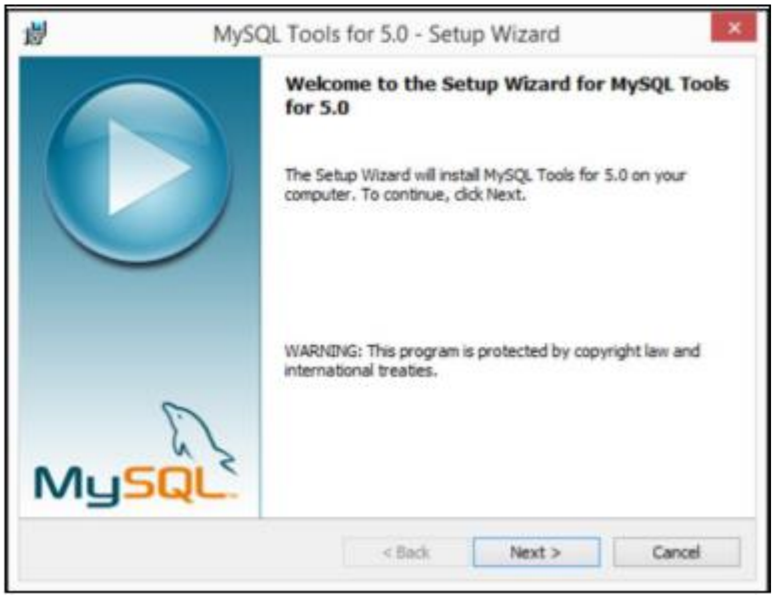
Press [Finish] to close the Wizard.

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

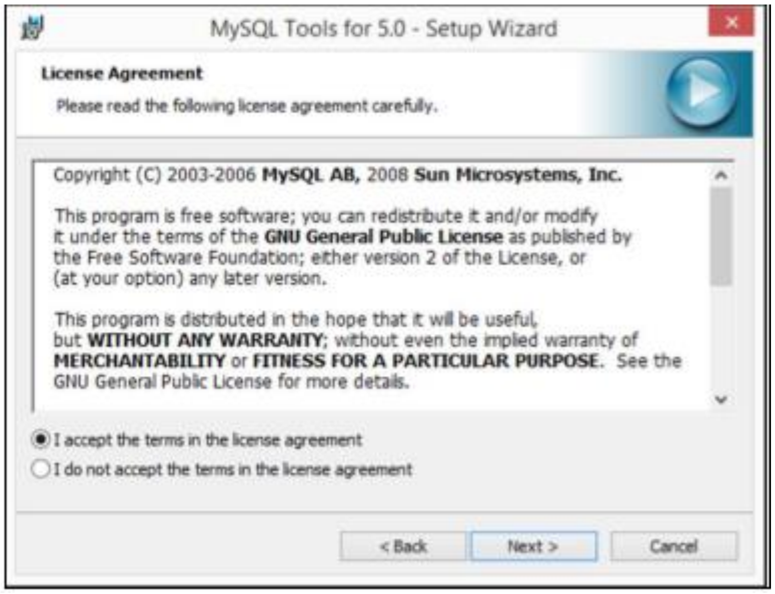
Welcome to the Setup Wizard for MySQL Tools for 5.0

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.**

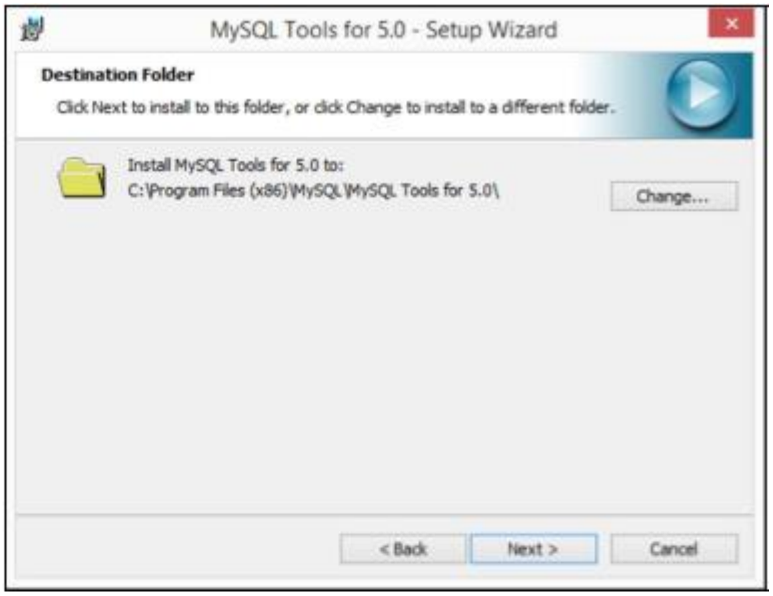
This program is free software; you can redistribute it and/or modify
it under the terms of the **GNU General Public License** as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

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)

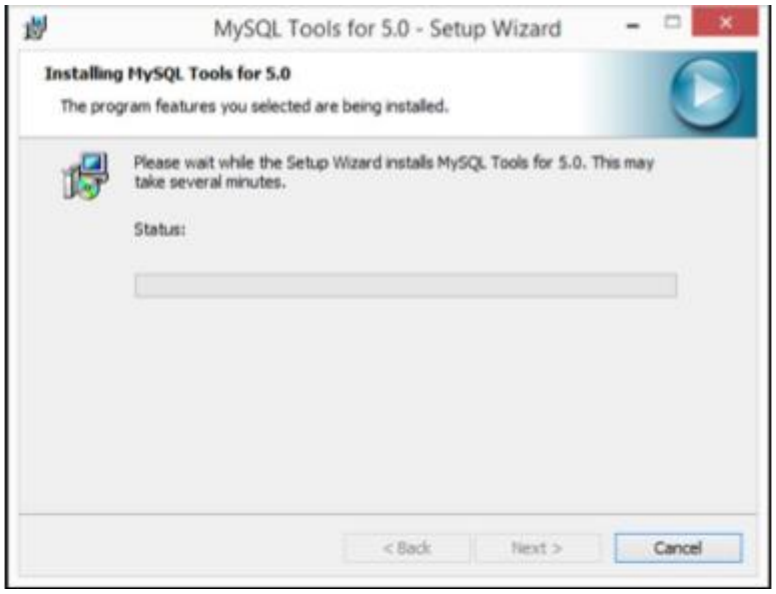

MySQL Tools for 5.0 - Setup Wizard
Destination Folder
Click Next to install to this folder, or click Change to install to a different folder.

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

Change...
< Back
Next >
Cancel

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)



Installing MySQL Tools for 5.0

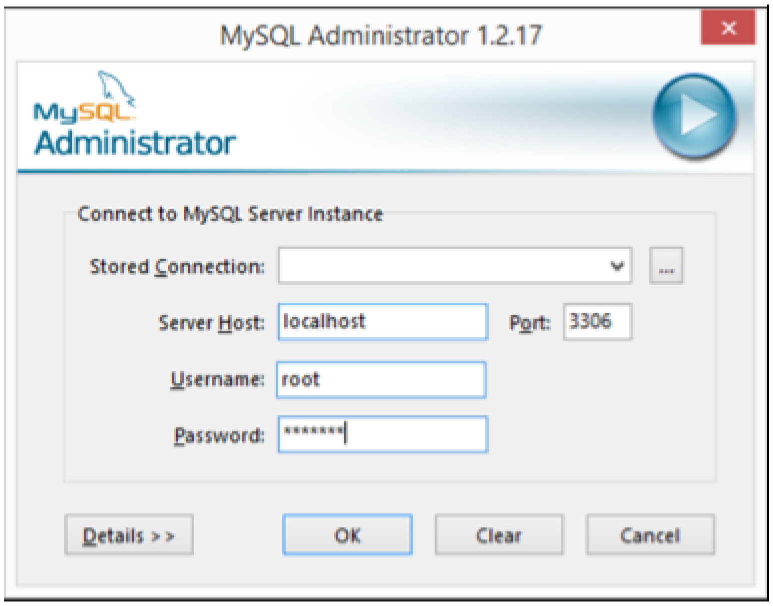
The program features you selected are being installed.

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:

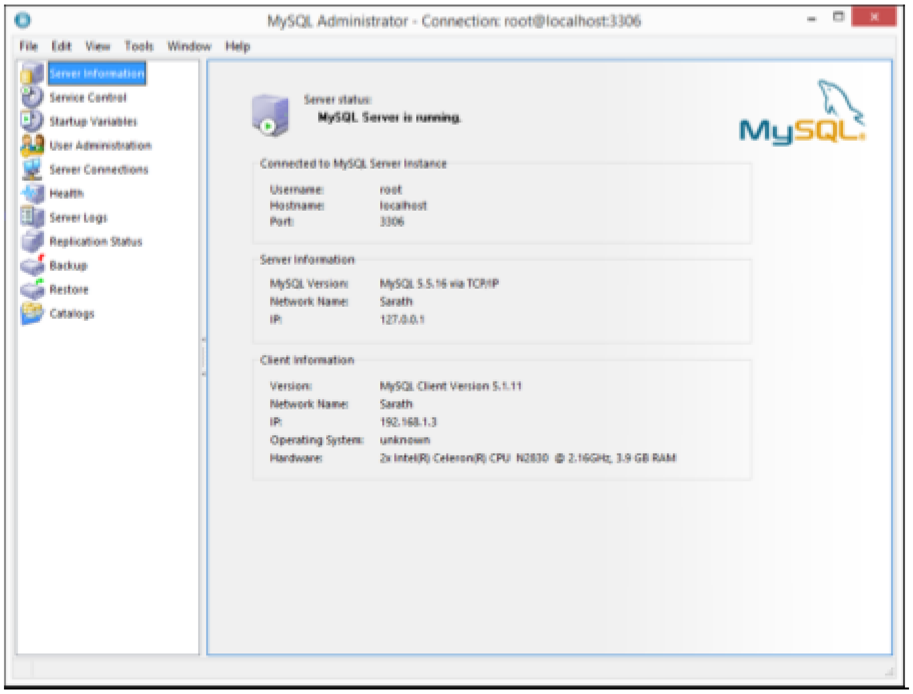
Server Host: localhost Port: 3306

Username: root

Password: *******

Figure A. 17 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 Reception Hall Management System

After setting up the database,

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

APPENDIX B –DESIGN DOCUMENTATION

B.1 Activity Diagram

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

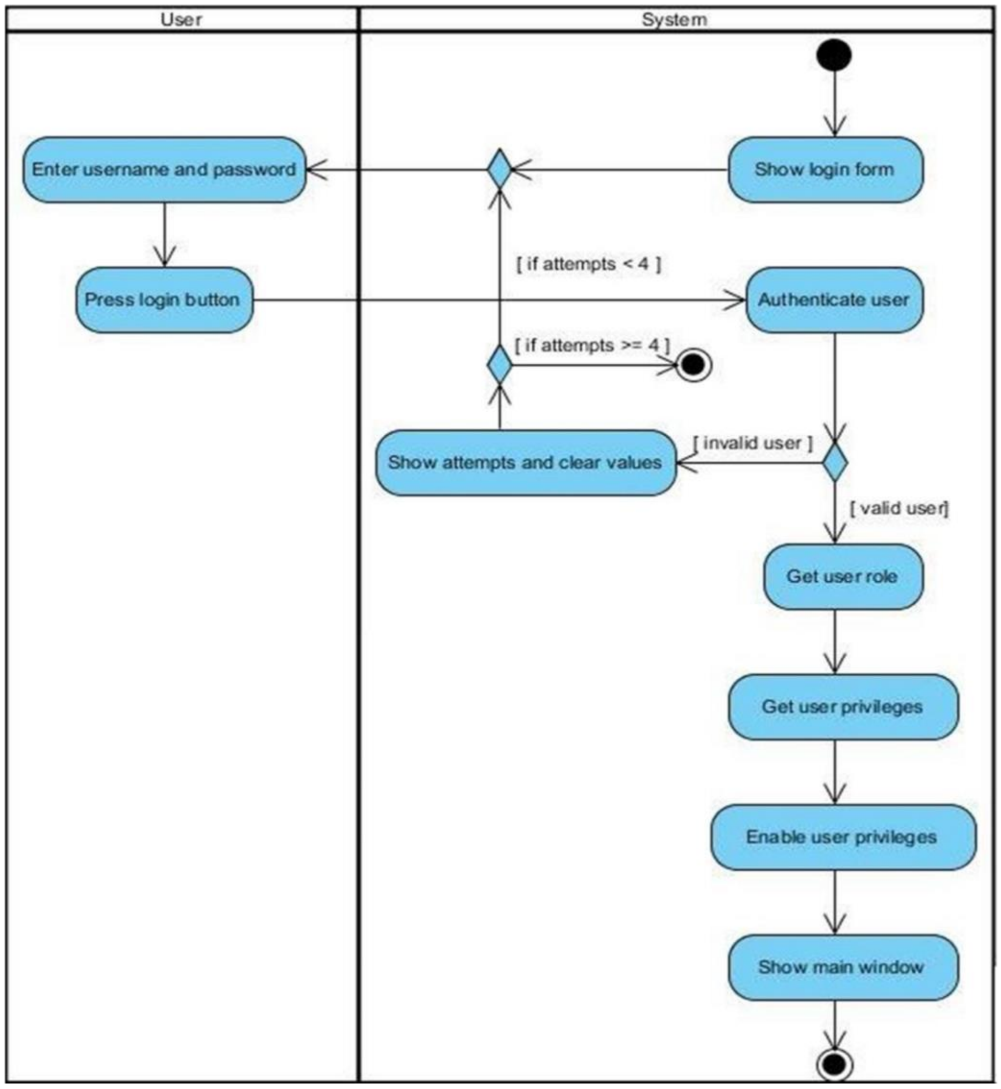


Figure B. 1 Activity Diagram for insert new User

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 booking details are some use cases relevant to Administrator. Use case description for Administrator is listed in table B.1, B.2, B.3

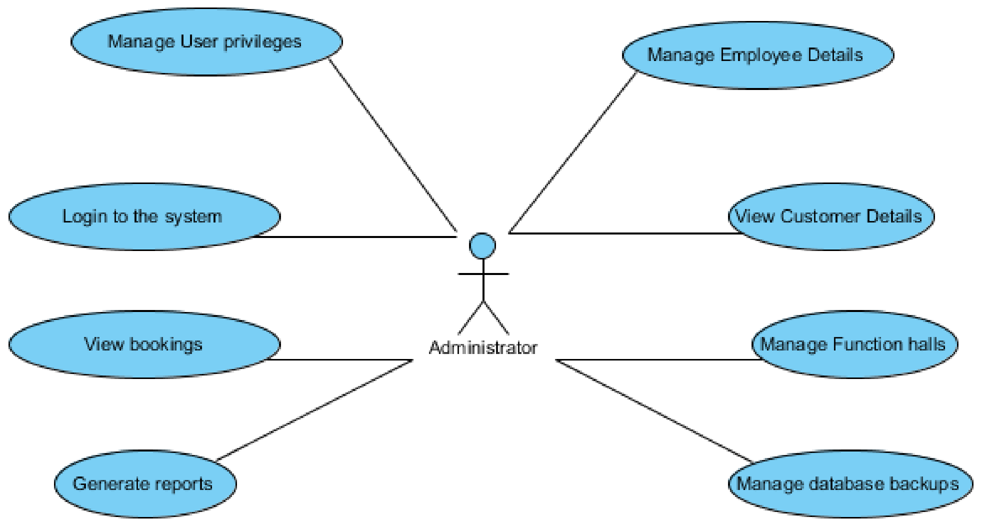


Figure B. 2 Use case diagram for Administrator

Use case	Manage Hall Details (Add new)
Actor	Administrator
Over view	
Add new hall into the system	
Precondition(s)	
User must login to the system under authorized user privilege	
Flow of event	
Go to main window and select Hall Management Click Hall button Enter Relevant Information and click Add	

Post Conditions
New hall will be available to the customer

Table B. 1 Use case description for Add Hall

Use case	Manage Booking Details (Update existing)
Actor	Administrator
Over view	
Update booking details	
Precondition(s)	
User must login to the system under authorized user privilege	
Flow of event	
Go to main window and Select Service Booking Click Provider 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
Over view	
Remove booking details	
Precondition(s)	
User must login to the system under authorized user privilege	
Flow of event	
Go to main window and Select Service booking Click Provider button Select a row from table The system will load selected data into the form	

Then click delete button
Post Conditions
Booking details will delete

Table B. 3 Use case description for Delete Booking

Following figure B.3 shows use cases relevant to Booking Manager. To perform such use cases first of all Booking Manager has to login to the system. Make catering booking, service booking, hall booking and finish the bookings are use cases relevant to Booking Manager. And use case description for Booking Manager is listed in table B.4

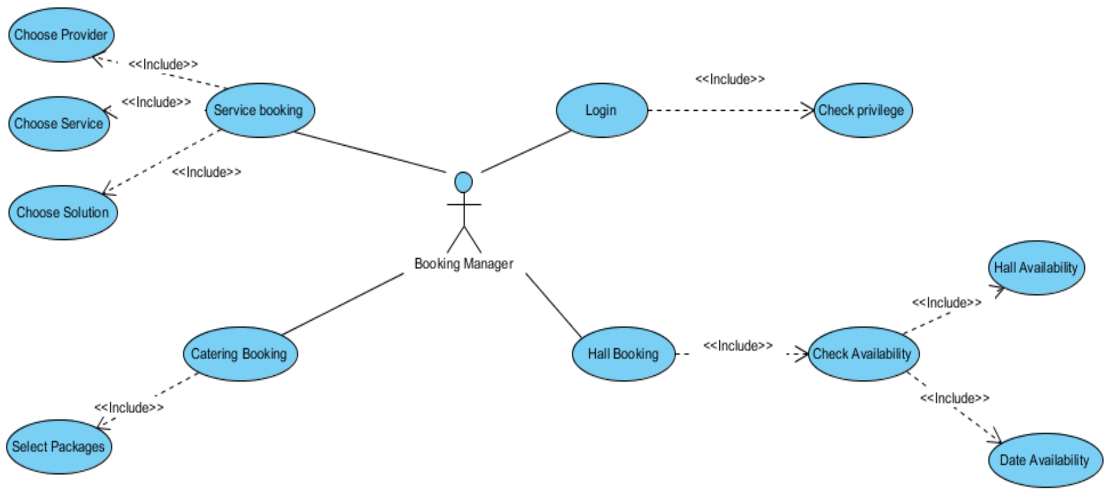


Figure B. 3 Use case diagram for Booking Manager

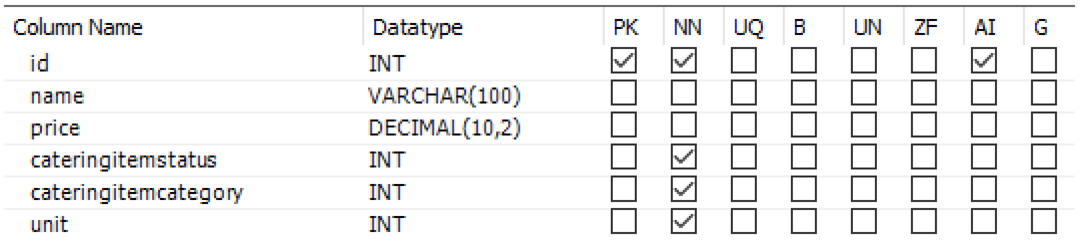
Use case	Manage Booking Details
Actor	Booking Manager
Over view	
Add new booking	
Precondition(s)	

User must login to the system under authorized user privilege
Flow of event
Go to main window and Select Booking Click Hall Booking button Enter Relevant Information and click Catering or Service Booking Button (If Catering)Enter Relevant Information and click Service Booking Enter Relevant Information and click Finish Booking
Post Conditions
New Booking will be available to the relevant customer

Table B. 4 Use case description for Booking

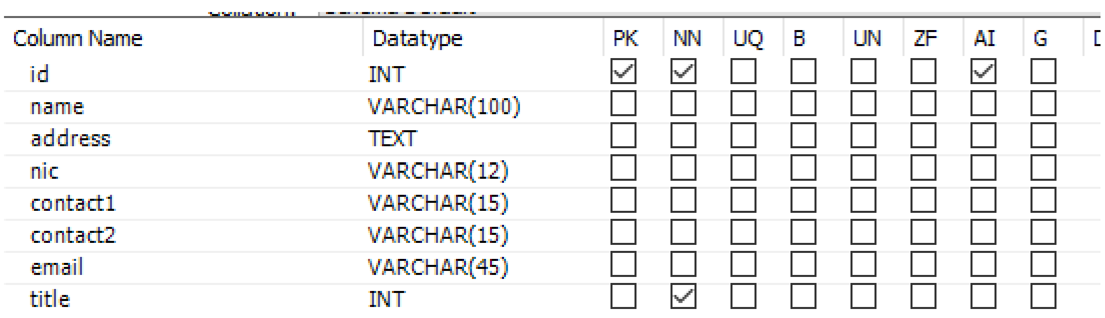
B.3 Database Table Design

Followings show the table design in the database of the system.



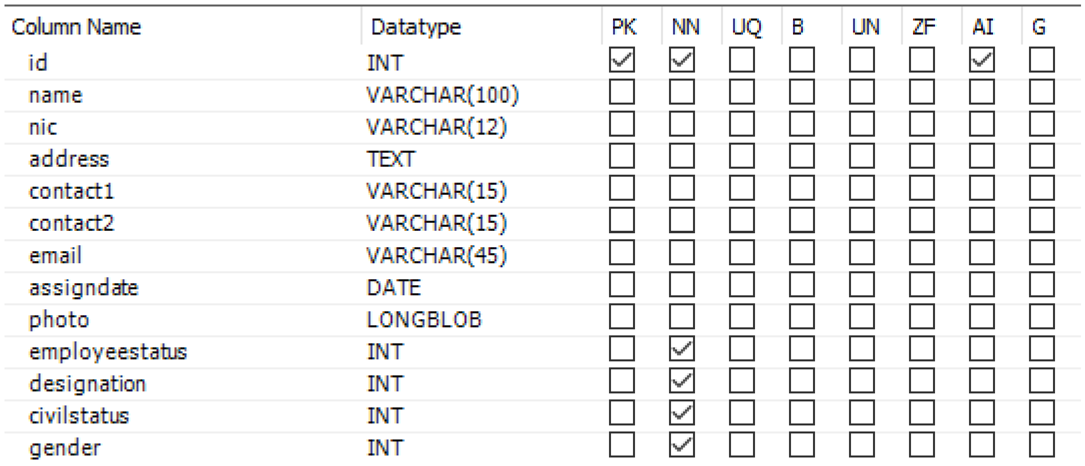
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(100)								
price	DECIMAL(10,2)								
cateringitemstatus	INT		✓						
cateringitemcategory	INT		✓						
unit	INT		✓						

Figure B. 4 Catering item Table



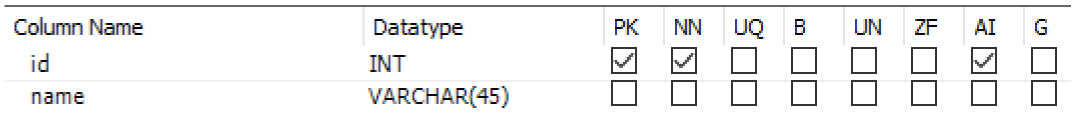
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(100)								
address	TEXT								
nic	VARCHAR(12)								
contact1	VARCHAR(15)								
contact2	VARCHAR(15)								
email	VARCHAR(45)								
title	INT		✓						

Figure B. 5 Customer Table



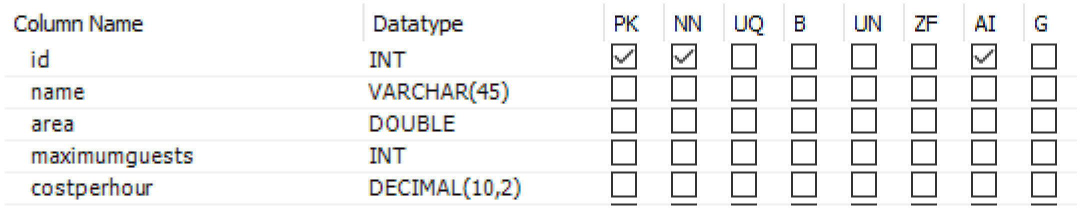
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(100)								
nic	VARCHAR(12)								
address	TEXT								
contact1	VARCHAR(15)								
contact2	VARCHAR(15)								
email	VARCHAR(45)								
assigndate	DATE								
photo	LONGBLOB								
employeestatus	INT		✓						
designation	INT		✓						
civilstatus	INT		✓						
gender	INT		✓						

Figure B. 6Employee Table



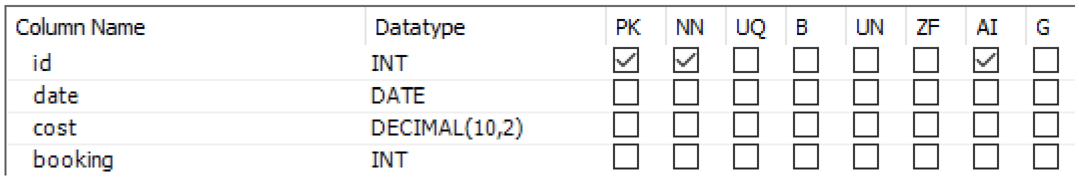
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(45)								

Figure B. 7 Hall Facility Table



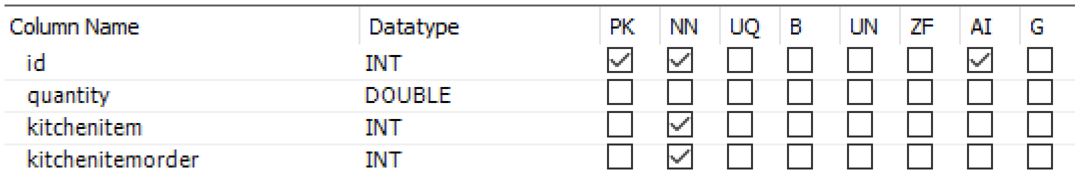
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(45)								
area	DOUBLE								
maximumguests	INT								
costperhour	DECIMAL(10,2)								

Figure B. 8 Hall Table



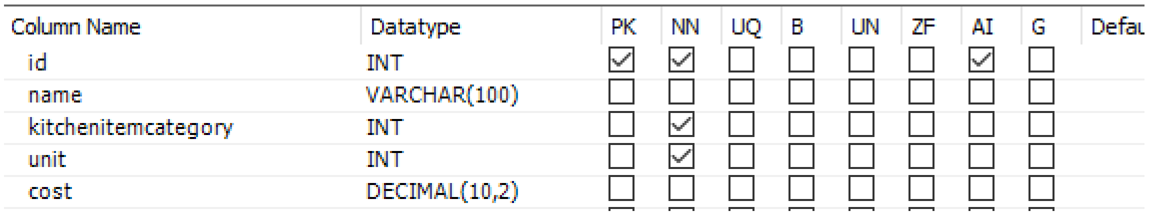
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
date	DATE								
cost	DECIMAL(10,2)								
booking	INT								

Figure B. 9 Kitchen Item Order Table



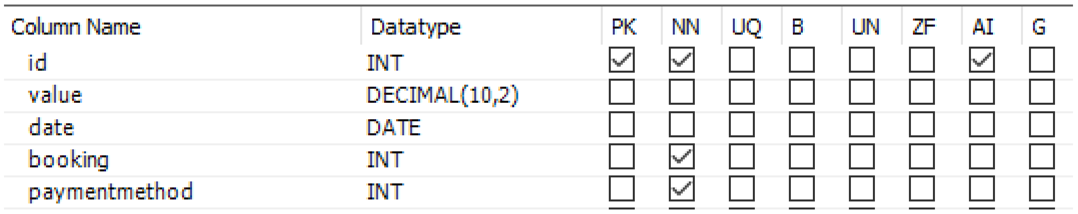
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
quantity	DOUBLE								
kitchenitem	INT		✓						
kitchenitemorder	INT		✓						

Figure B. 10 Kitchen Item Order list Table



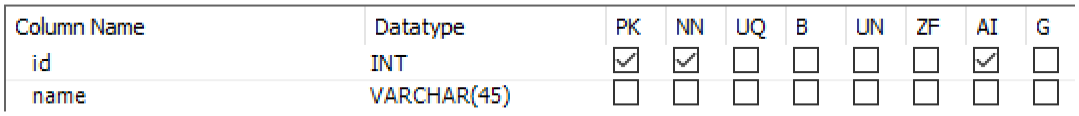
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Defau
id	INT	✓	✓					✓		
name	VARCHAR(100)									
kitchenitemcategory	INT		✓							
unit	INT		✓							
cost	DECIMAL(10,2)									

Figure B. 11 Kitchen Item Table



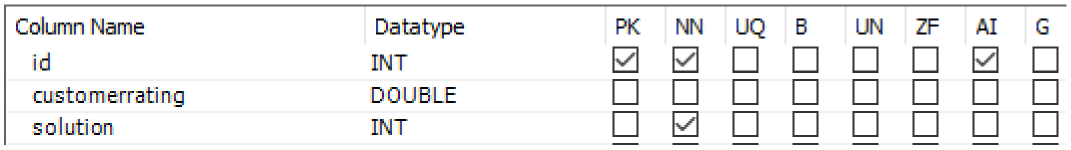
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
value	DECIMAL(10,2)								
date	DATE								
booking	INT		✓						
paymentmethod	INT		✓						

Figure B. 12 Payment Table



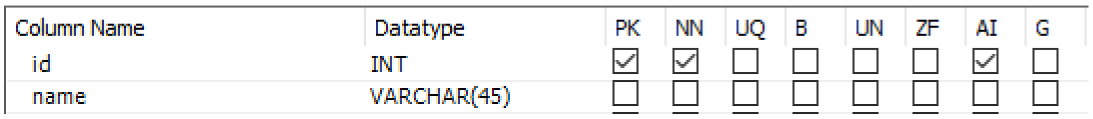
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(45)								

Figure B. 13 Service Table



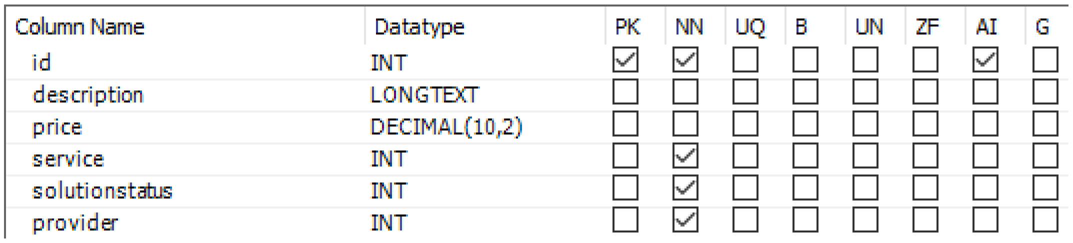
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
customerrating	DOUBLE								
solution	INT		✓						

Figure B. 14 Service Booking Table



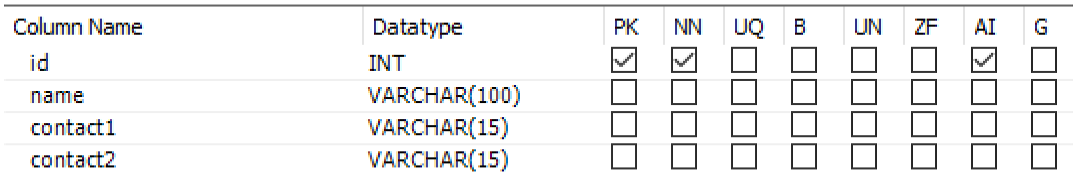
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(45)								

Figure B. 15 Role Table



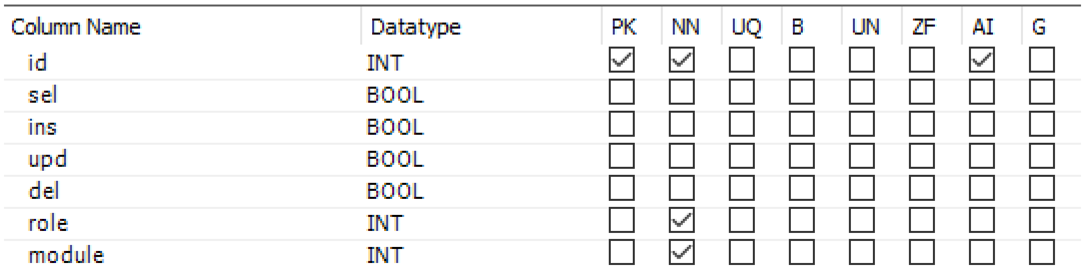
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
description	LONGTEXT								
price	DECIMAL(10,2)								
service	INT		✓						
solutionstatus	INT		✓						
provider	INT		✓						

Figure B. 16 Solution Table



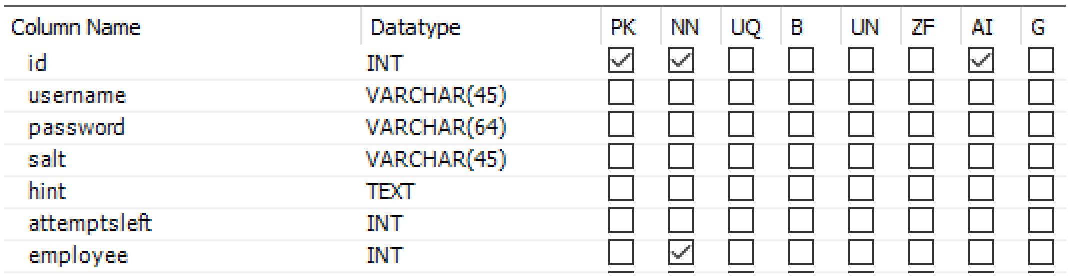
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(100)								
contact1	VARCHAR(15)								
contact2	VARCHAR(15)								

Figure B. 17 Provider Table



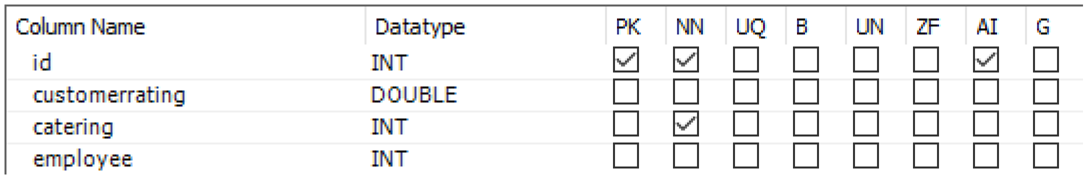
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
sel	BOOL								
ins	BOOL								
upd	BOOL								
del	BOOL								
role	INT		✓						
module	INT		✓						

Figure B. 18 Privilege Table



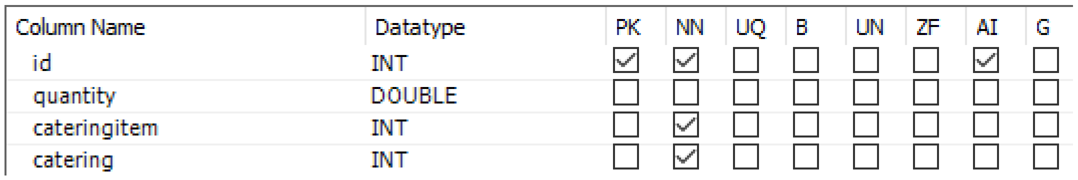
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
username	VARCHAR(45)								
password	VARCHAR(64)								
salt	VARCHAR(45)								
hint	TEXT								
attemptsleft	INT								
employee	INT		✓						

Figure B. 19 User Table



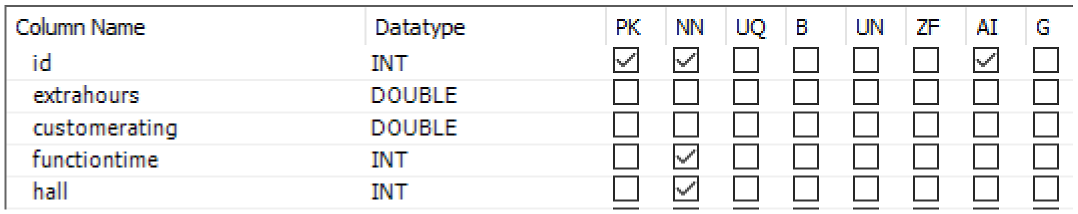
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
customerrating	DOUBLE								
catering	INT		✓						
employee	INT								

Figure B. 20 Catering Booking Table



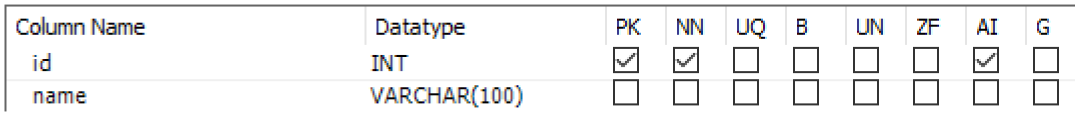
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
quantity	DOUBLE								
cateringitem	INT		✓						
catering	INT		✓						

Figure B. 21 Catering has catering item Table



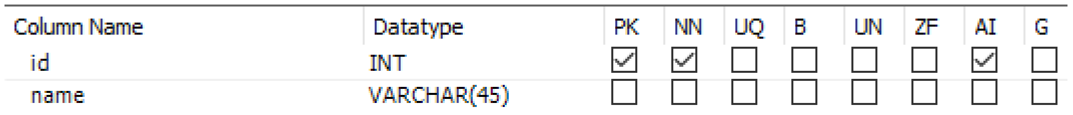
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
extrahours	DOUBLE								
customerating	DOUBLE								
functiontime	INT		✓						
hall	INT		✓						

Figure B. 22 Hall Booking Table



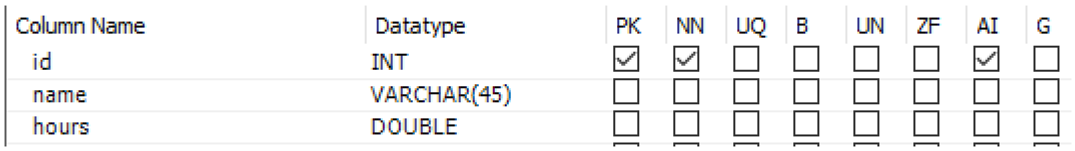
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(100)								

Figure B. 23 Kitchen Item Category Table



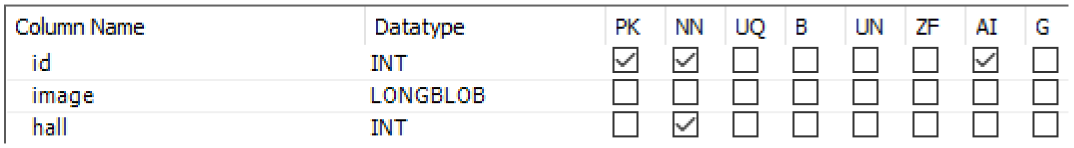
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(45)								

Figure B. 24 Catering Item Category Table



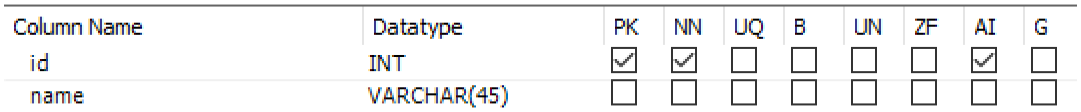
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓					✓	
name	VARCHAR(45)								
hours	DOUBLE								

Figure B. 25 Function Time Table



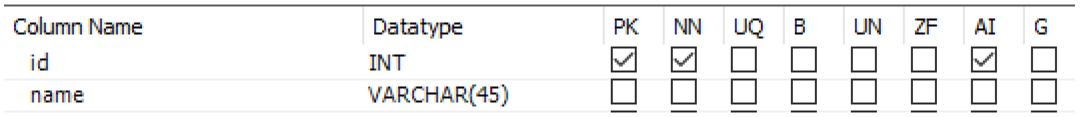
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓	☐	☐	☐	☐	✓	☐
image	LONGBLOB	☐	☐	☐	☐	☐	☐	☐	☐
hall	INT	☐	✓	☐	☐	☐	☐	☐	☐

Figure B. 26 Hall Image Table



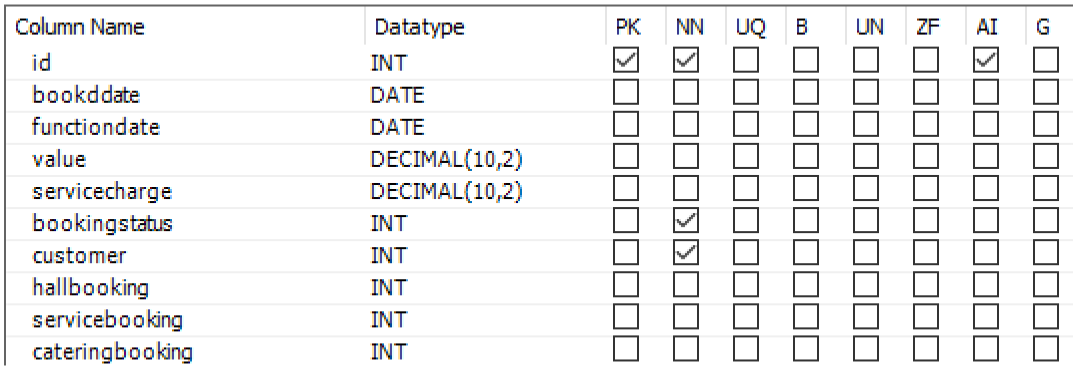
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓	☐	☐	☐	☐	✓	☐
name	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐

Figure B. 27 Designation Table



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓	☐	☐	☐	☐	✓	☐
name	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐

Figure B. 28 Module Table



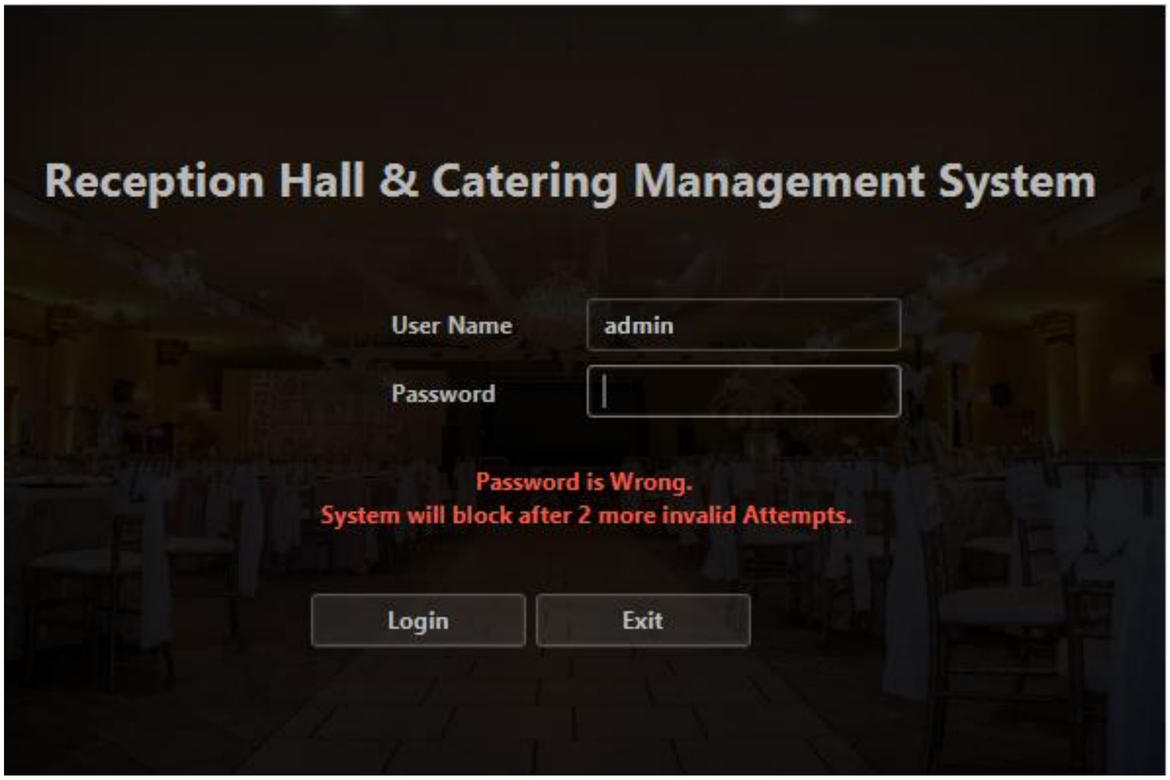
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
id	INT	✓	✓	☐	☐	☐	☐	✓	☐
bookddate	DATE	☐	☐	☐	☐	☐	☐	☐	☐
functiondate	DATE	☐	☐	☐	☐	☐	☐	☐	☐
value	DECIMAL(10,2)	☐	☐	☐	☐	☐	☐	☐	☐
servicecharge	DECIMAL(10,2)	☐	☐	☐	☐	☐	☐	☐	☐
bookingstatus	INT	☐	✓	☐	☐	☐	☐	☐	☐
customer	INT	☐	✓	☐	☐	☐	☐	☐	☐
hallbooking	INT	☐	☐	☐	☐	☐	☐	☐	☐
servicebooking	INT	☐	☐	☐	☐	☐	☐	☐	☐
cateringbooking	INT	☐	☐	☐	☐	☐	☐	☐	☐

Figure B. 29 Booking Table

APPENDIX C –USER DOCUMENTATION

- Login form

Following Figure C.1 shows user login form which allows users to log into the system. All levels of users can log into the system in one form. When user try to log into the system check whether this user is valid user or not otherwise system display error message.

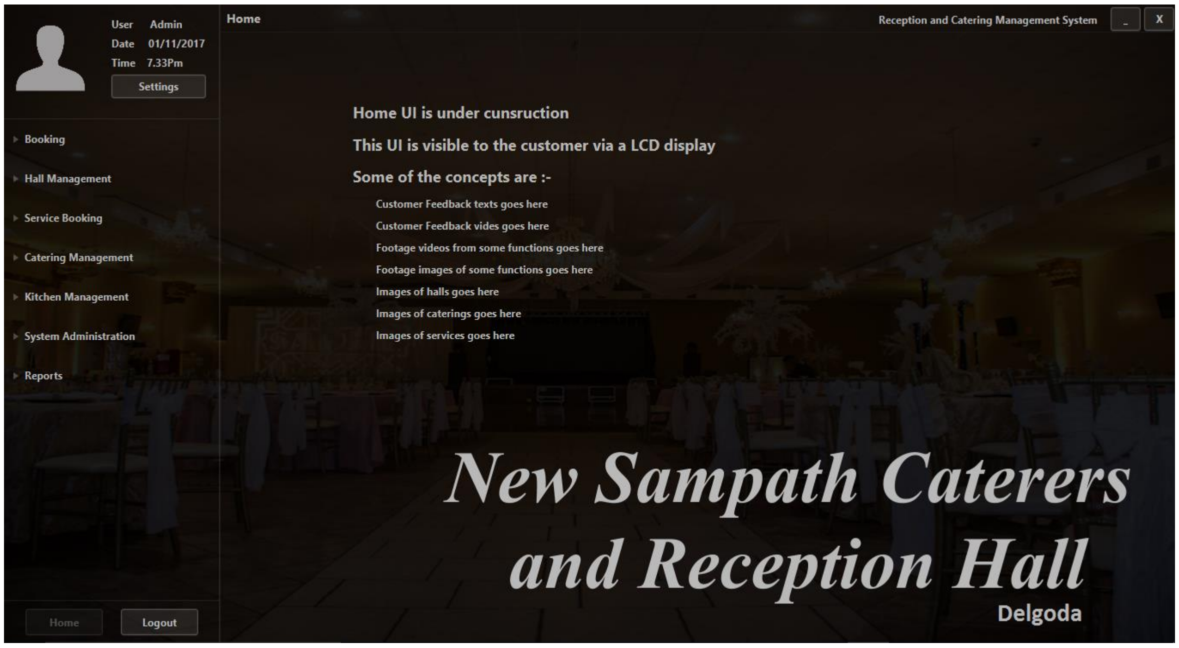


Password is Wrong.
System will block after 2 more invalid Attempts.

Figure C. 1 Login Form

- Main Form

All the functions of each and every module can be accessible by users through this main form. Following figure C.2 shows how main menu looks like.



Home UI is under cunsruction

This UI is visible to the customer via a LCD display

Some of the concepts are :-

- Customer Feedback texts goes here
- Customer Feedback vides goes here
- Footage videos from some functions goes here
- Footage images of some functions goes here
- Images of halls goes here
- Images of caterings goes here
- Images of services goes here

Figure C. 2 Main Form

- Main menu items

Figure C.3 shows the main menu item of the main form.

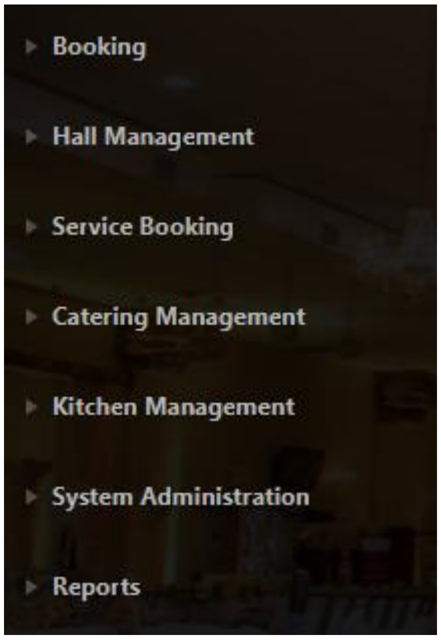
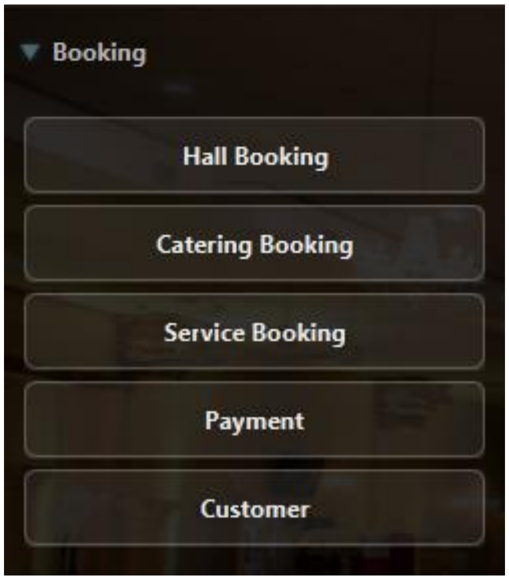
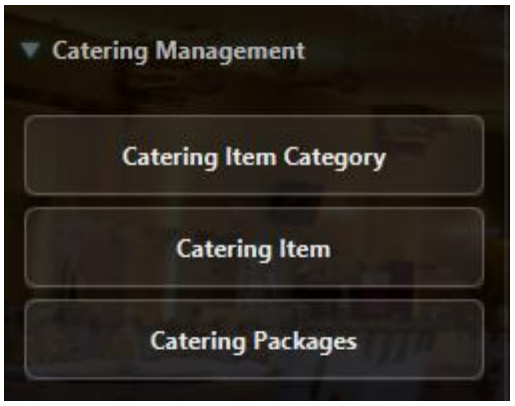


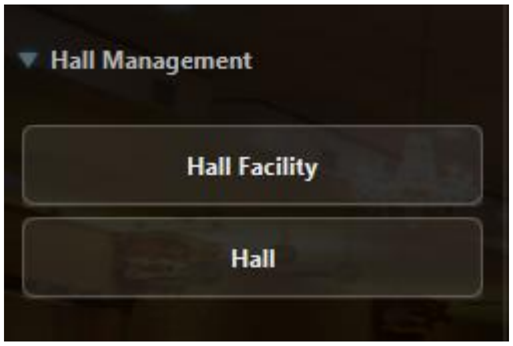
Figure C. 3 Main menu items

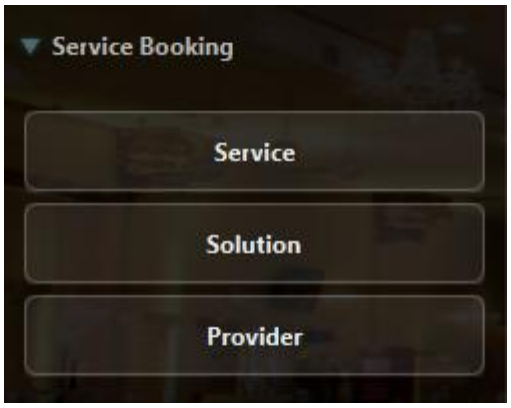
- Submenu items

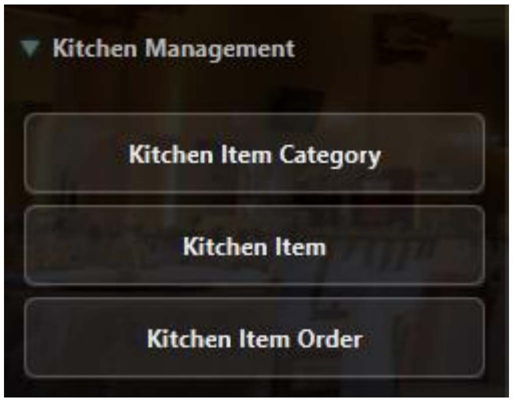
Figure C.4 shows the sub menu items of the main form











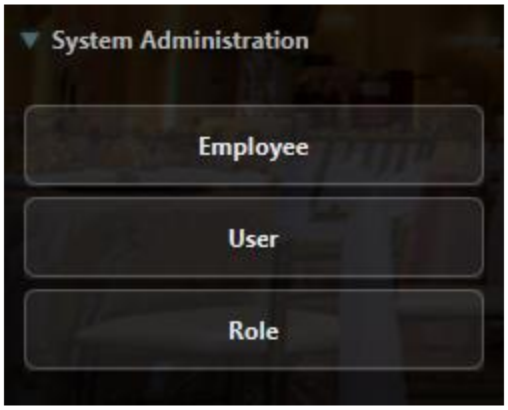
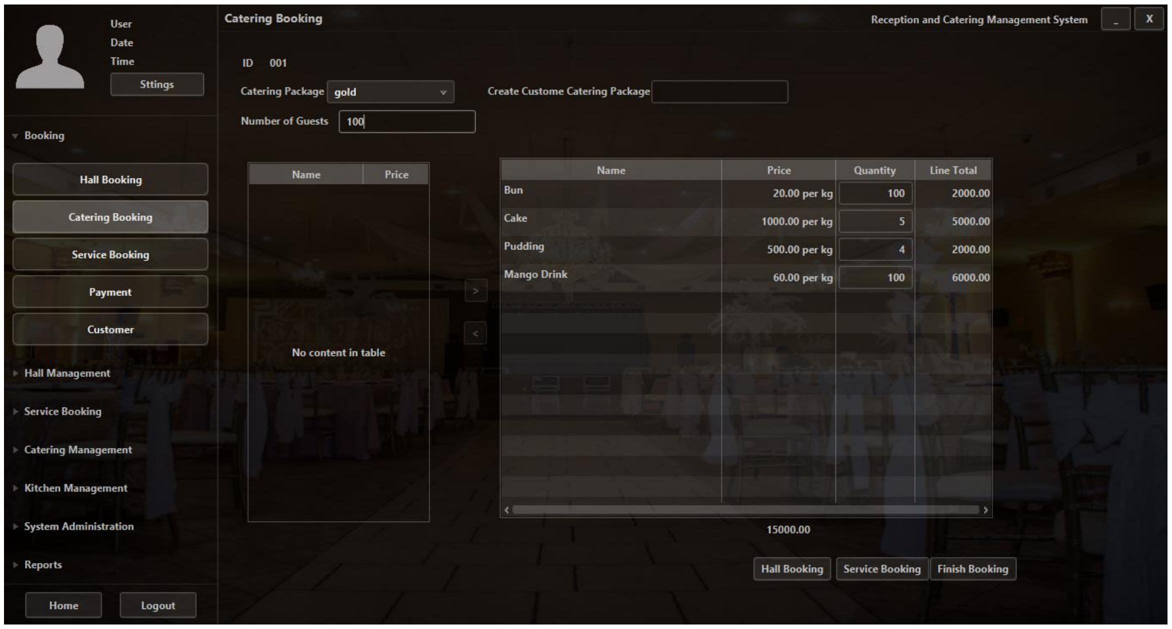


Figure C. 4 Sub menu items

- Catering Booking Form

By giving the number of guests and select catering package customer can book a catering booking. If customer not satisfy with the package, he can customize a catering packages his wish from create custom catering package field.

Catering Booking Form will display as follows. Figure C.5 represents the Catering Booking form.



Catering Booking

Reception and Catering Management System

ID 001

Catering Package: gold

Create Custome Catering Package:

Number of Guests: 100

Name	Price
No content in table	

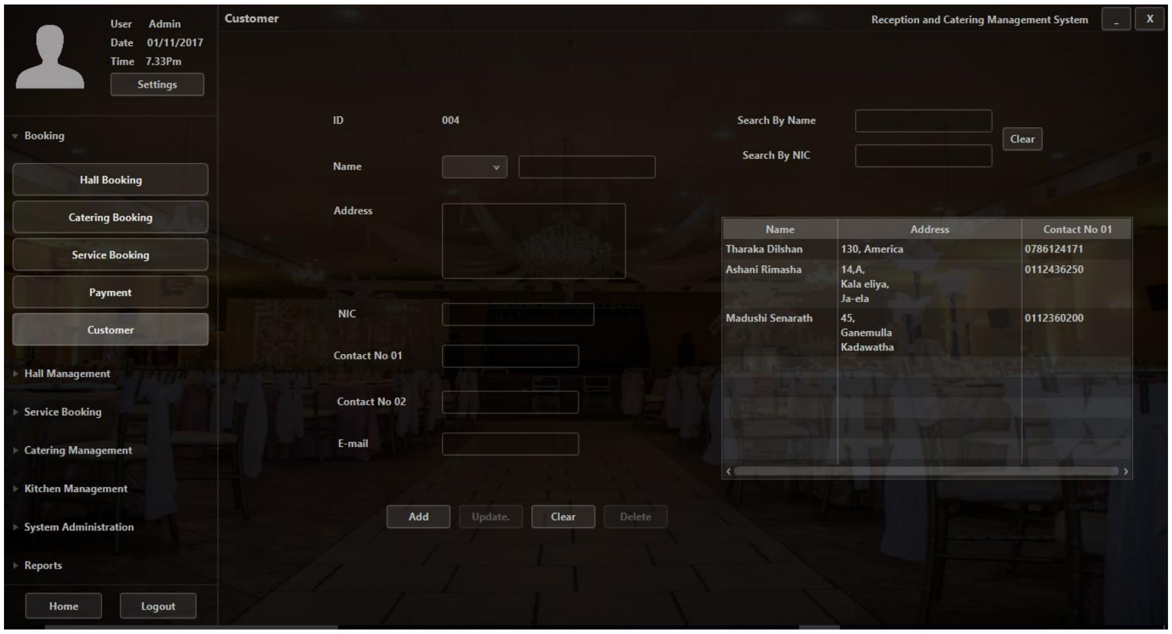
Name	Price	Quantity	Line Total
Bun	20.00 per kg	100	2000.00
Cake	1000.00 per kg	5	5000.00
Pudding	500.00 per kg	4	2000.00
Mango Drink	60.00 per kg	100	6000.00

15000.00

Figure C. 5 Catering Booking form

- Customer Registration form

When a guest come to a booking, he or she must fill the data for a reservation.



ID 004
Name
Address
NIC
Contact No 01
Contact No 02
E-mail

Search By Name
Search By NIC
Clear

Name	Address	Contact No 01
Tharaka Dilshan	130, America	0786124171
Ashani Rimasha	14,A, Kala eliya, Ja-ela	0112436250
Madushi Senarath	45, Ganemulla Kadawatha	0112360200

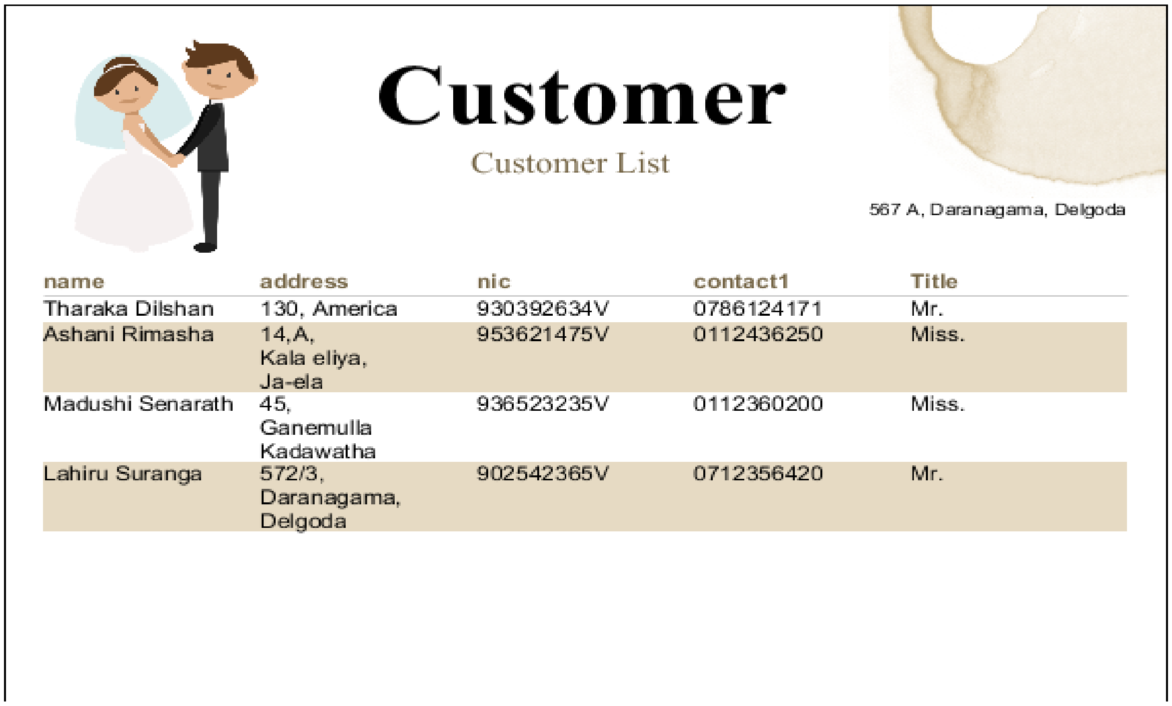
Figure C. 6 Customer registration form

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.

- Customer report

A report for every customer can be produced for the use of both hotel & guest with an explanation of further information. In here relevant customer details are printed in report. User can customize report information fields depending on their needs. Figure D.1 shows the customer details report generated by system.



Customer

Customer List

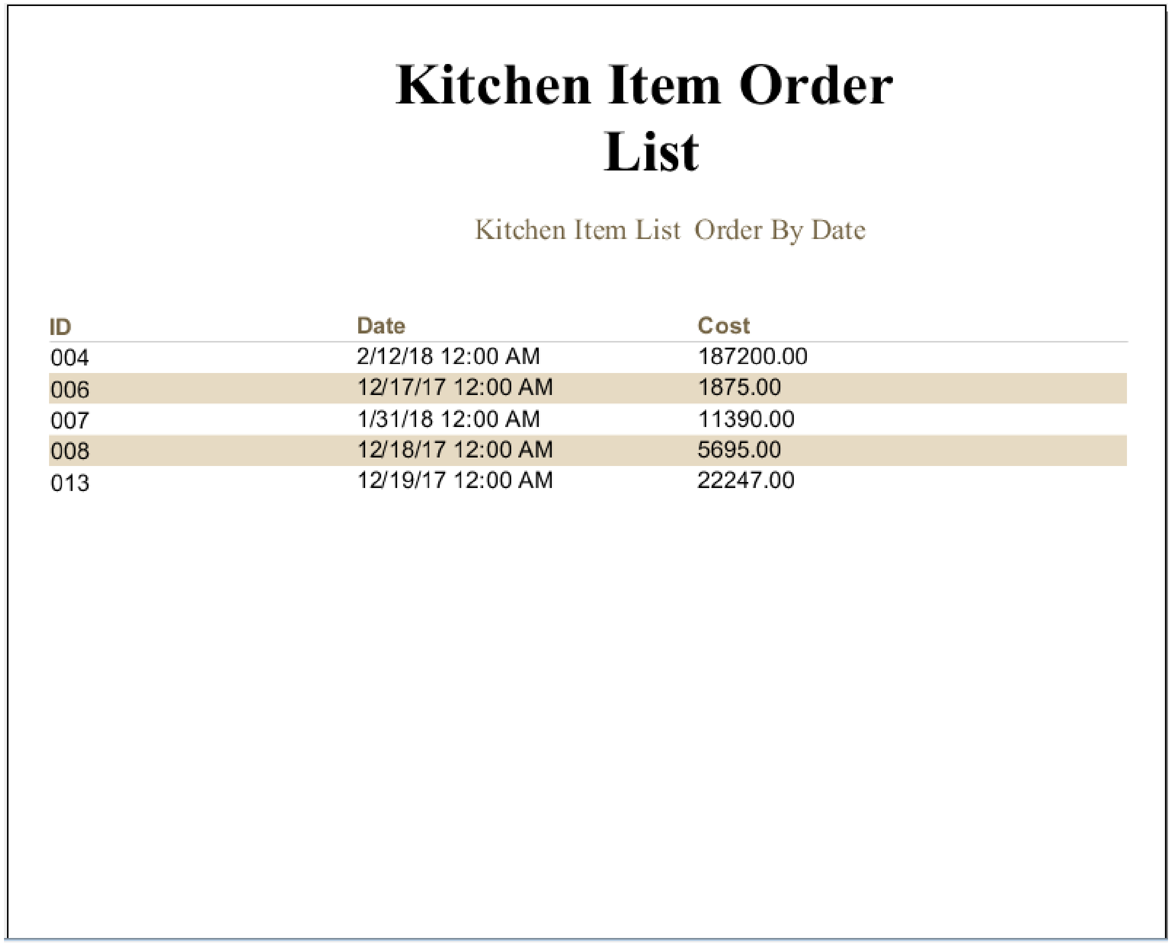
567 A, Daranagama, Delgoda

name	address	nic	contact1	Title
Tharaka Dilshan	130, America	930392634V	0786124171	Mr.
Ashani Rimasha	14,A, Kala eliya, Ja-ela	953621475V	0112436250	Miss.
Madushi Senarath	45, Ganemulla Kadawatha	936523235V	0112360200	Miss.
Lahiru Suranga	572/3, Daranagama, Delgoda	902542365V	0712356420	Mr.

Figure D. 1 Customer Details Report

- Kitchen Item order by date range report

A report for kitchen item orders can be produced for the use creation of kitchen item request from by manager. Following figure D.2 shows the kitchen item order details report generated by system. User can customize report depending on their needs



Kitchen Item Order List

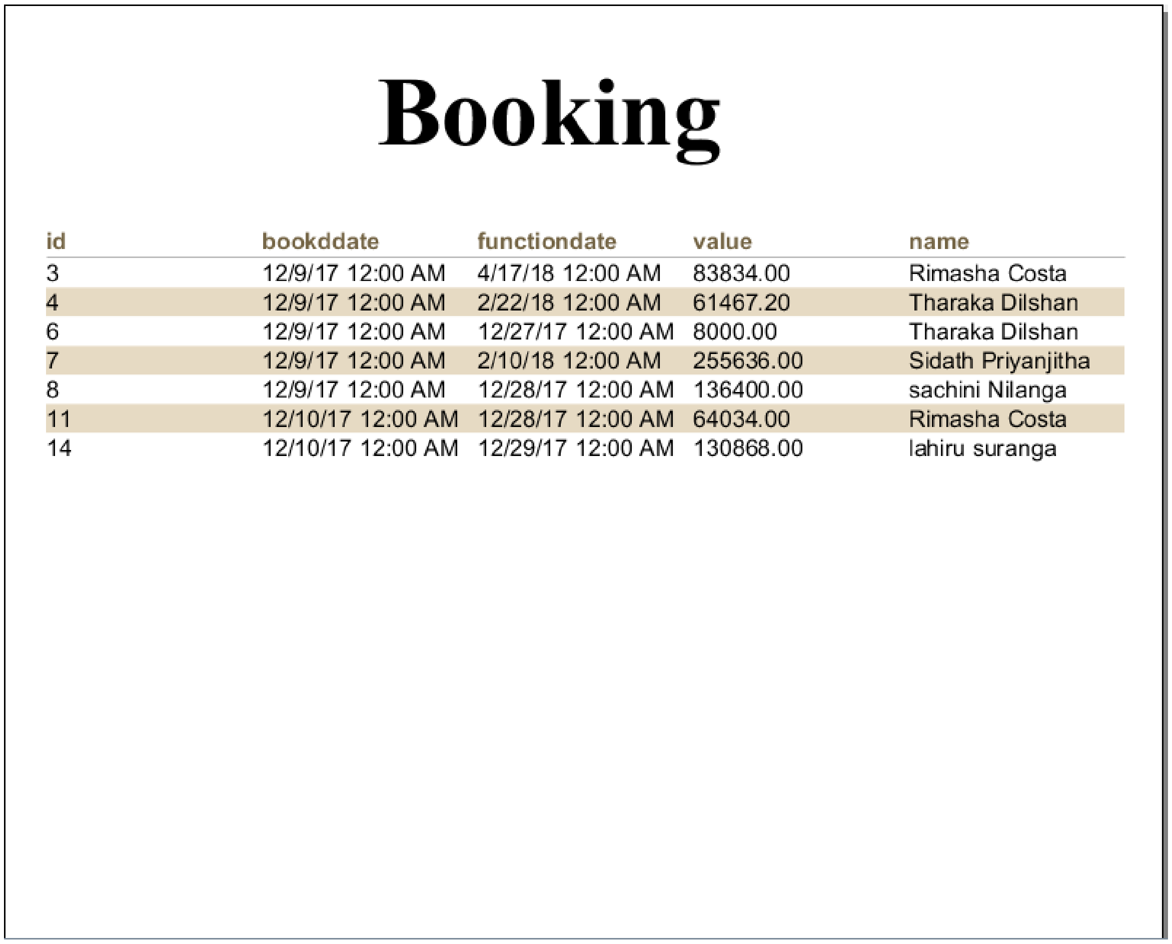
Kitchen Item List Order By Date

ID	Date	Cost
004	2/12/18 12:00 AM	187200.00
006	12/17/17 12:00 AM	1875.00
007	1/31/18 12:00 AM	11390.00
008	12/18/17 12:00 AM	5695.00
013	12/19/17 12:00 AM	22247.00

Figure D. 2 Kitchen Item list order by date range Details Report

- Bookings by date range detail report

A report for bookings can be produced for the use of hotel with an explanation of further information. Figure D.3 shows the bookings details report generated by system.



Booking

id	bookddate	functiondate	value	name
3	12/9/17 12:00 AM	4/17/18 12:00 AM	83834.00	Rimasha Costa
4	12/9/17 12:00 AM	2/22/18 12:00 AM	61467.20	Tharaka Dilshan
6	12/9/17 12:00 AM	12/27/17 12:00 AM	8000.00	Tharaka Dilshan
7	12/9/17 12:00 AM	2/10/18 12:00 AM	255636.00	Sidath Priyanjitha
8	12/9/17 12:00 AM	12/28/17 12:00 AM	136400.00	sachini Nilanga
11	12/10/17 12:00 AM	12/28/17 12:00 AM	64034.00	Rimasha Costa
14	12/10/17 12:00 AM	12/29/17 12:00 AM	130868.00	lahiru suranga

Figure D. 3 Bookings by date range Details Report

APPENDIX E –TEST RESULTS

Login related test results are mentioned in Table E.1. All the test cases were passed. Login is one of the critical function for the security.

Test No	Description	Expected result	Pass/Fail
1	Login into the system with valid username & password	User should get main window of the system	Pass
2	Login into the system with invalid username & password	Should show the remaining attempts	Pass
3	Login into the system with empty username/password	Should show an error message at the login page	Pass
4	Changing password with valid passwords.	Password should get changed next login new password should work.	Pass
5	Login out from the system	After confirmation system should logout and show the login screen.	Pass

Table E. 1 Login Related Test cases

Employee entity related test cases are mentioned in Table E.2. Add, update and delete functions were tested.

Test No	Description	Expected result	Pass/Fail
1	Create employee with valid information.	Notification box will be shown as “Employee added succeed” and added data will be shown at the table.	Pass
2	Create employee with invalid information.	Employee should not get created. Database record should not get added. Text of text filed of invalid information, should change to red.	Pass

3	Update employee with proper information	Employee details should get changed. System should return a success notification.	Pass
4	Employee search and filtering	Employee should be filtered out.	Pass
5	Deleting of an employee	Delete confirmation should return. Once confirmed employee should get deleted.	Pass

Table E. 2 Employee related test cases

User entity related test cases are mention in Table E.3 add, update, delete functions are tested.

Test No	Description	Expected result	Pass/Fail
1	Select Combo Box "Employee"	List all the employees to combo box	Pass
2	Confirm Password should equal to Password	Error Message box will be shown as "Passwords don't match"	Pass
3	Select ">>" button	All the roles view in available list will be removed and it will be added into assigned list	Pass
4	Select "<<" button	All the roles view in assigned list will be removed and it will be added into available list	Pass
5	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

Table E. 3 User related test cases

Customer entity related test cases are mentioned in Table E.4 add, update, delete functions are tested.

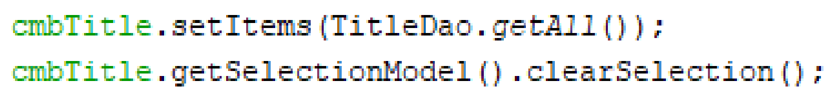
Test No	Description	Expected result	Pass/Fail
1	Enter digits and symbols as guest name	Text color of the text field change to red.	Pass

2	Select Combo Box	List all the Guest types to combo box	Pass
3	Enter letters and symbols as contact numbers.	Text color of the text field change to red	Pass
4	Enter all information correctly and click Add	Notification box will be shown as “Customer added succeed” and added data will be shown at the table.	Pass
5	Without entering data to some fields and click Add	Employee should not get created. Database record should not get added. Show a red star before the relevant text field that did not enter the data.	Pass
6	Select a row from the table	Selected Customer details will be load into the relevant Text Fields. Add button will be disabled and Update, Delete buttons will be enabled	Pass
7	Enter Text value in the “Search by Name” field	Customer table will refresh according to typed text values	Pass
8	Enter Text value in the “Search by NIC” field	Customer table will refresh according to typed text values	Pass
9	Click Clear button	Clear text Fields data and clear selection of the table	Pass

Table E. 4 Customer related test cases

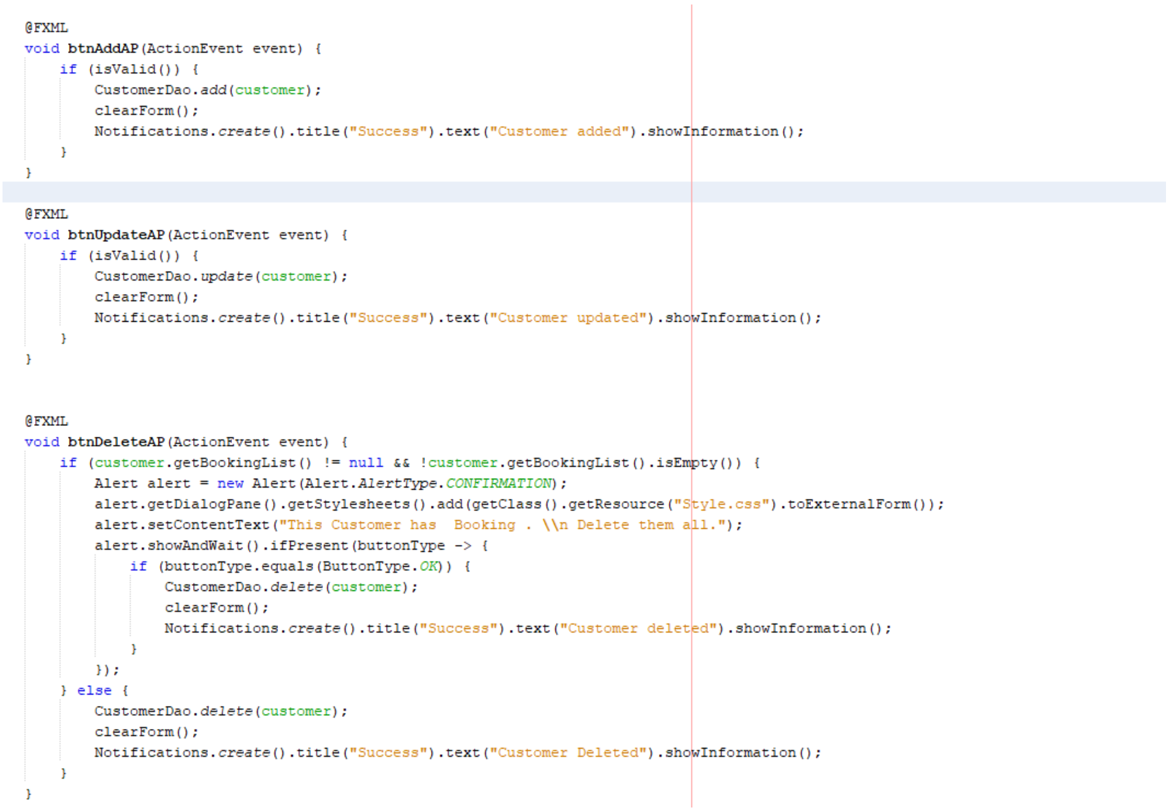
APPENDIX F –CODE LISTING

F.1 Code for Load Title



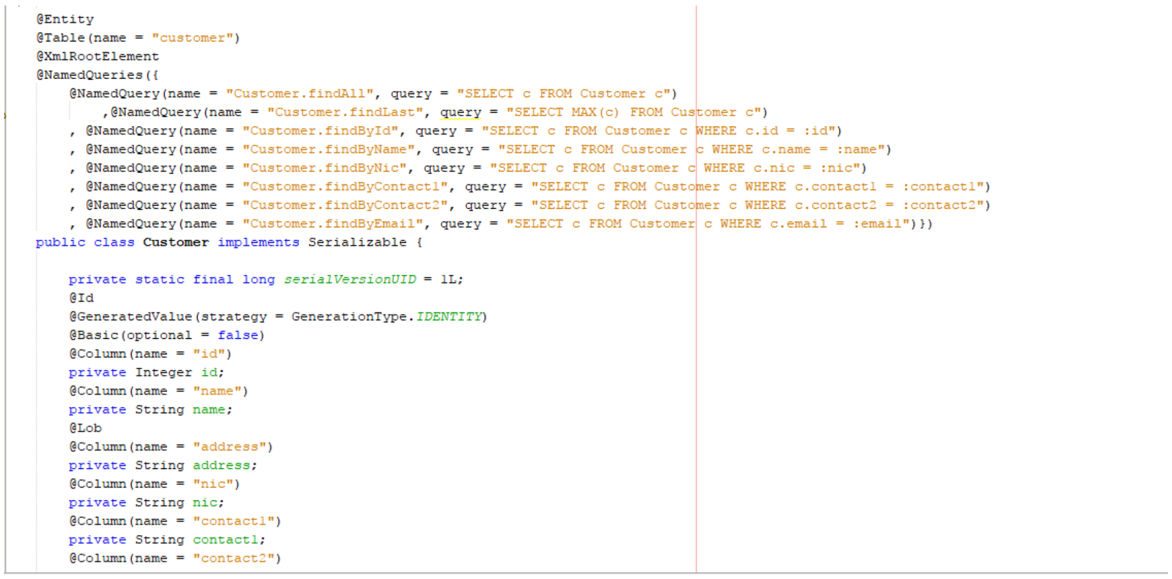
```
cmbTitle.setItems(TitleDao.getAll());
cmbTitle.getSelectionModel().clearSelection();
```

F.2 Code for Add, Update, Delete operations in Customer module



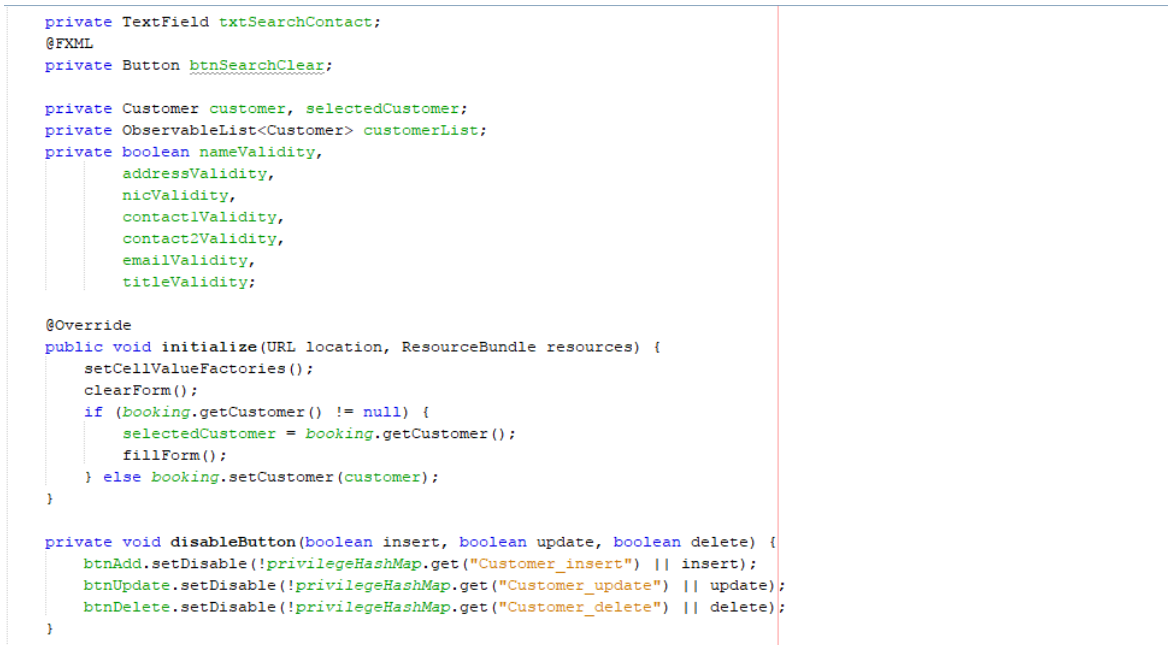
```
@FXML
void btnAddAP(ActionEvent event) {
    if (isValid()) {
        CustomerDao.add(customer);
        clearForm();
        Notifications.create().title("Success").text("Customer added").showInformation();
    }
}

@FXML
void btnUpdateAP(ActionEvent event) {
    if (isValid()) {
        CustomerDao.update(customer);
        clearForm();
        Notifications.create().title("Success").text("Customer updated").showInformation();
    }
}


@FXML
void btnDeleteAP(ActionEvent event) {
    if (customer.getBookingList() != null && !customer.getBookingList().isEmpty()) {
        Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
        alert.getDialogPane().getStylesheets().add(getClass().getResource("Style.css").toExternalForm());
        alert.setContentText("This Customer has  Booking . \\n Delete them all.");
        alert.showAndWait().ifPresent(buttonType -> {
            if (buttonType.equals(ButtonType.OK)) {
                CustomerDao.delete(customer);
                clearForm();
                Notifications.create().title("Success").text("Customer deleted").showInformation();
            }
        });
    } else {
        CustomerDao.delete(customer);
        clearForm();
        Notifications.create().title("Success").text("Customer Deleted").showInformation();
    }
}
```

F.3 Entity class for Customer

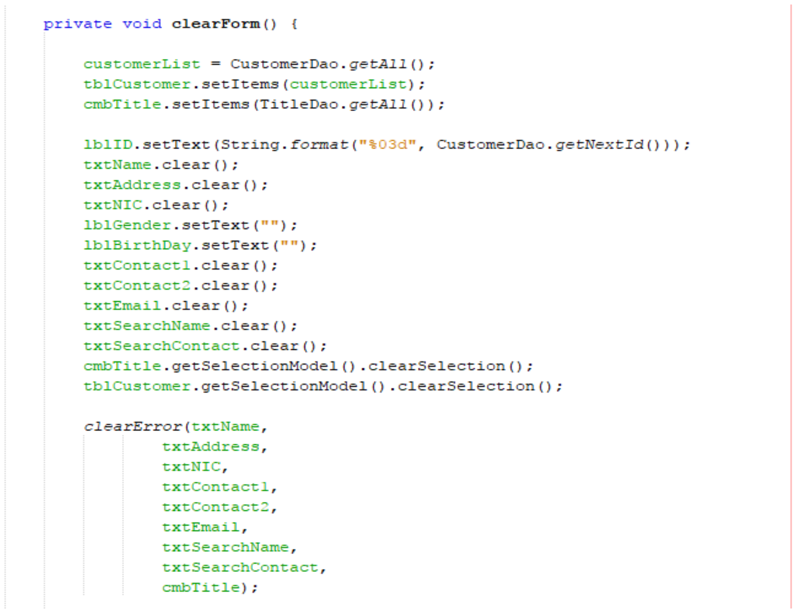


```
@Entity
@Table(name = "customer")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Customer.findAll", query = "SELECT c FROM Customer c")
        ,@NamedQuery(name = "Customer.findLast", query = "SELECT MAX(c) FROM Customer c")
    , @NamedQuery(name = "Customer.findById", query = "SELECT c FROM Customer c WHERE c.id = :id")
    , @NamedQuery(name = "Customer.findByName", query = "SELECT c FROM Customer c WHERE c.name = :name")
    , @NamedQuery(name = "Customer.findByNic", query = "SELECT c FROM Customer c WHERE c.nic = :nic")
    , @NamedQuery(name = "Customer.findByContact1", query = "SELECT c FROM Customer c WHERE c.contact1 = :contact1")
    , @NamedQuery(name = "Customer.findByContact2", query = "SELECT c FROM Customer c WHERE c.contact2 = :contact2")
    , @NamedQuery(name = "Customer.findByEmail", query = "SELECT c FROM Customer c WHERE c.email = :email")})
public class Customer implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id")
    private Integer id;
    @Column(name = "name")
    private String name;
    @Lob
    @Column(name = "address")
    private String address;
    @Column(name = "nic")
    private String nic;
    @Column(name = "contact1")
    private String contact1;
    @Column(name = "contact2")
```

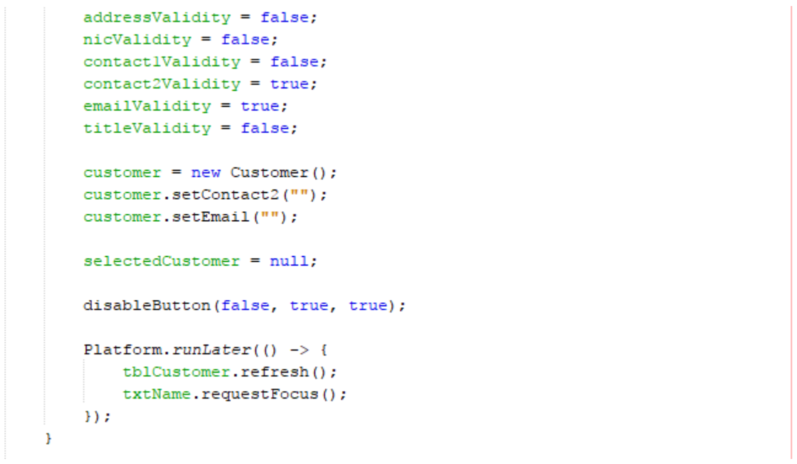
F.4 Controller class for Customer



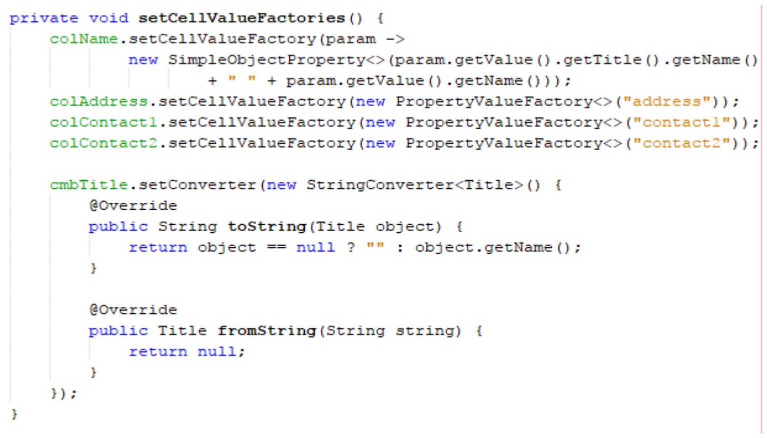
```
private TextField txtSearchContact;
@FXML
private Button btnSearchClear;

private Customer customer, selectedCustomer;
private ObservableList<Customer> customerList;
private boolean nameValidity,
        addressValidity,
        nicValidity,
        contact1Validity,
        contact2Validity,
        emailValidity,
        titleValidity;

@Override
public void initialize(URL location, ResourceBundle resources) {
    setCellValueFactories();
    clearForm();
    if (booking.getCustomer() != null) {
        selectedCustomer = booking.getCustomer();
        fillForm();
    } else booking.setCustomer(customer);
}

private void disableButton(boolean insert, boolean update, boolean delete) {
    btnAdd.setDisable(!privilegeHashMap.get("Customer_insert") || insert);
    btnUpdate.setDisable(!privilegeHashMap.get("Customer_update") || update);
    btnDelete.setDisable(!privilegeHashMap.get("Customer_delete") || delete);
}
```



```
private void clearForm() {

    customerList = CustomerDao.getAll();
    tblCustomer.setItems(customerList);
    cmbTitle.setItems(TitleDao.getAll());

    lblID.setText(String.format("%03d", CustomerDao.getNextId()));
    txtName.clear();
    txtAddress.clear();
    txtNIC.clear();
    lblGender.setText("");
    lblBirthDay.setText("");
    txtContact1.clear();
    txtContact2.clear();
    txtEmail.clear();
    txtSearchName.clear();
    txtSearchContact.clear();
    cmbTitle.getSelectionModel().clearSelection();
    tblCustomer.getSelectionModel().clearSelection();

    clearError(txtName,
            txtAddress,
            txtNIC,
            txtContact1,
            txtContact2,
            txtEmail,
            txtSearchName,
            txtSearchContact,
            cmbTitle);
```

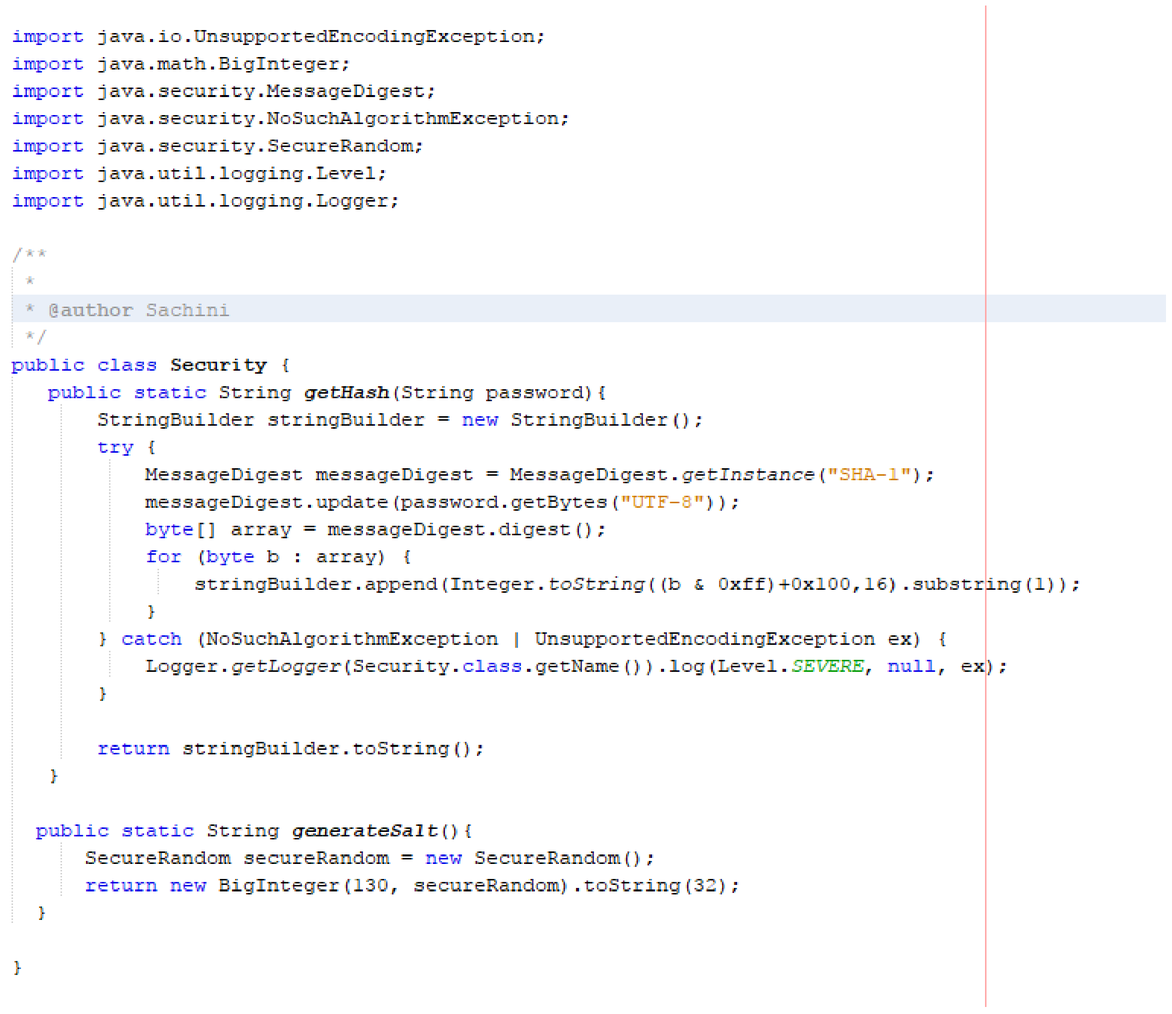


```
    addressValidity = false;
    nicValidity = false;
    contact1Validity = false;
    contact2Validity = true;
    emailValidity = true;
    titleValidity = false;

    customer = new Customer();
    customer.setContact2("");
    customer.setEmail("");

    selectedCustomer = null;

    disableButton(false, true, true);

    Platform.runLater(() -> {
        tblCustomer.refresh();
        txtName.requestFocus();
    });
}
```



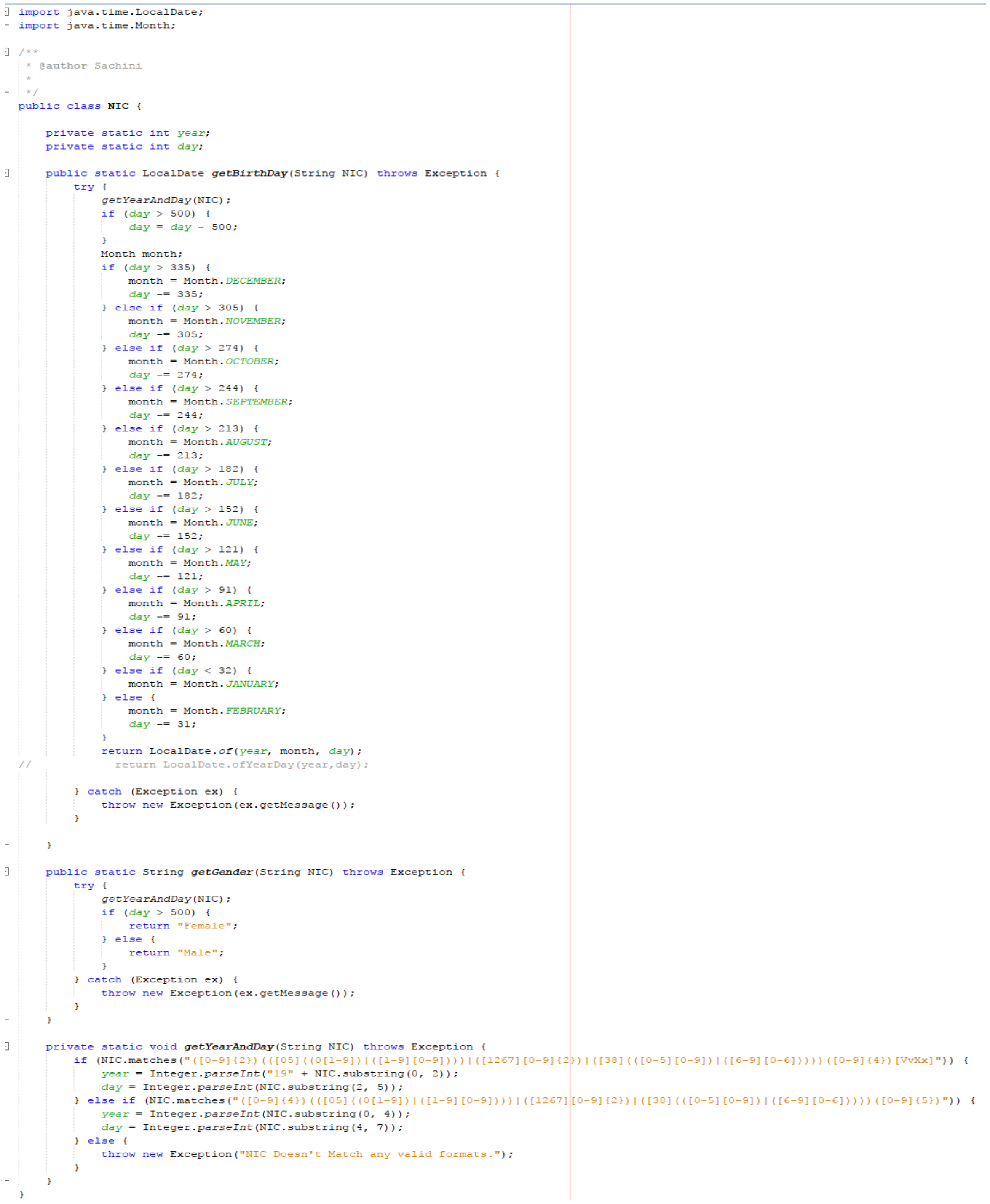
```
private void setCellValueFactories() {
    colName.setCellValueFactory(param ->
            new SimpleObjectProperty<>(param.getValue().getTitle().getName()
                    + " " + param.getValue().getName()));
    colAddress.setCellValueFactory(new PropertyValueFactory<>("address"));
    colContact1.setCellValueFactory(new PropertyValueFactory<>("contact1"));
    colContact2.setCellValueFactory(new PropertyValueFactory<>("contact2"));

    cmbTitle.setConverter(new StringConverter<Title>() {
        @Override
        public String toString(Title object) {
            return object == null ? "" : object.getName();
        }

        @Override
        public Title fromString(String string) {
            return null;
        }
    });
}
```

F.4 Security Class



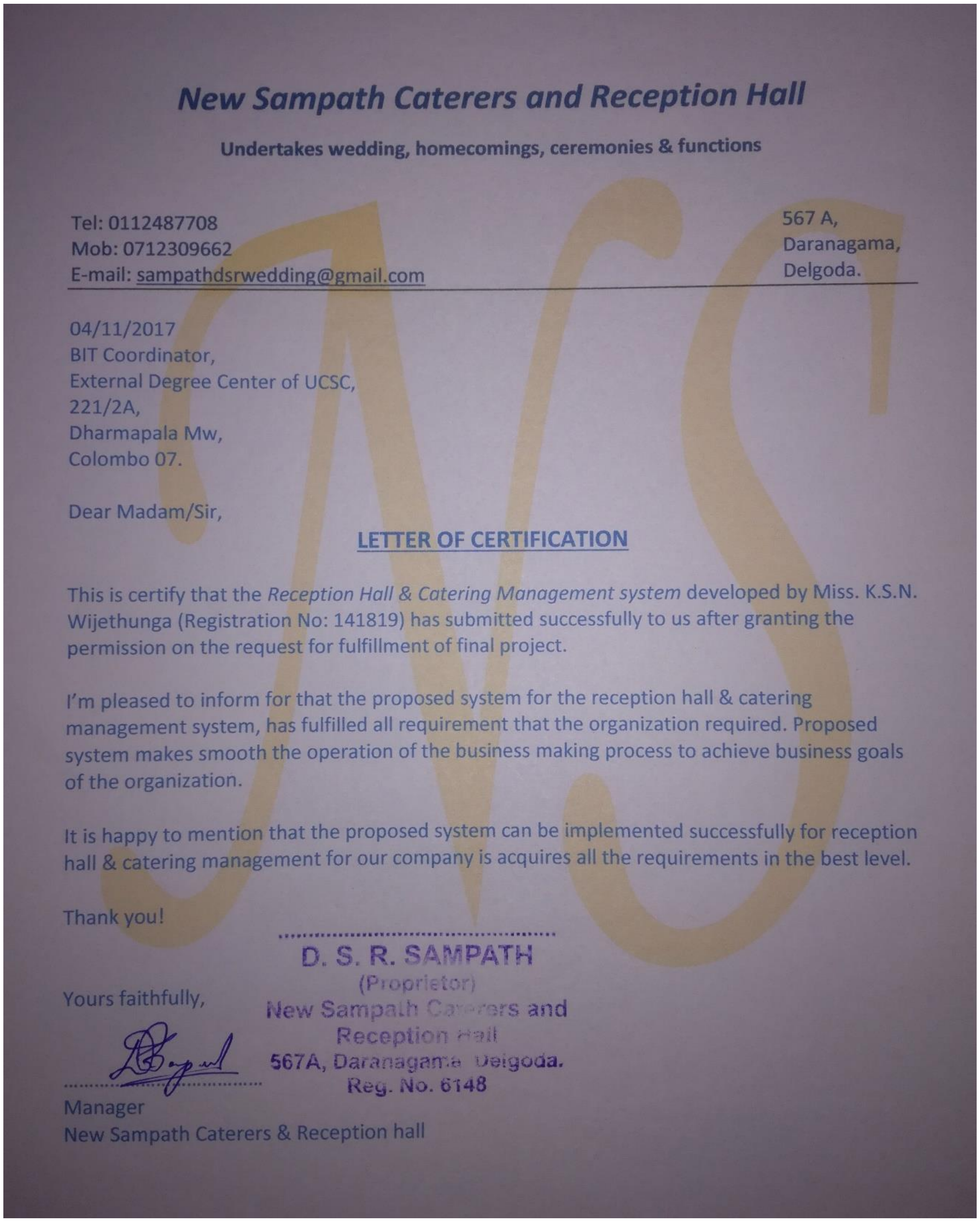
```
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Sachini
 */
public class Security {
   public static String getHash(String password){
       StringBuilder stringBuilder = new StringBuilder();
       try {
           MessageDigest messageDigest = MessageDigest.getInstance("SHA-1");
           messageDigest.update(password.getBytes("UTF-8"));
           byte[] array = messageDigest.digest();
           for (byte b : array) {
               stringBuilder.append(Integer.toString((b & 0xff)+0x100,16).substring(1));
           }
       } catch (NoSuchAlgorithmException | UnsupportedEncodingException ex) {
           Logger.getLogger(Security.class.getName()).log(Level.SEVERE, null, ex);
       }

       return stringBuilder.toString();
   }

  public static String generateSalt(){
      SecureRandom secureRandom = new SecureRandom();
      return new BigInteger(130, secureRandom).toString(32);
  }

}
```

F.5 NIC class for get gender and birthdate by NIC number



```java
import java.time.LocalDate;
import java.time.Month;

/**
 * @author Sachini
 *
 */
public class NIC {

    private static int year;
    private static int day;

    public static LocalDate getBirthDay(String NIC) throws Exception {
        try {
            getYearAndDay(NIC);
            if (day > 500) {
                day = day - 500;
            }
            Month month;
            if (day > 335) {
                month = Month.DECEMBER;
                day -= 335;
            } else if (day > 305) {
                month = Month.NOVEMBER;
                day -= 305;
            } else if (day > 274) {
                month = Month.OCTOBER;
                day -= 274;
            } else if (day > 244) {
                month = Month.SEPTEMBER;
                day -= 244;
            } else if (day > 213) {
                month = Month.AUGUST;
                day -= 213;
            } else if (day > 182) {
                month = Month.JULY;
                day -= 182;
            } else if (day > 152) {
                month = Month.JUNE;
                day -= 152;
            } else if (day > 121) {
                month = Month.MAY;
                day -= 121;
            } else if (day > 91) {
                month = Month.APRIL;
                day -= 91;
            } else if (day > 60) {
                month = Month.MARCH;
                day -= 60;
            } else if (day < 32) {
                month = Month.JANUARY;
            } else {
                month = Month.FEBRUARY;
                day -= 31;
            }
            return LocalDate.of(year, month, day);
//             return LocalDate.ofYearDay(year,day);

        } catch (Exception ex) {
            throw new Exception(ex.getMessage());
        }

    }

    public static String getGender(String NIC) throws Exception {
        try {
            getYearAndDay(NIC);
            if (day > 500) {
                return "Female";
            } else {
                return "Male";
            }
        } catch (Exception ex) {
            throw new Exception(ex.getMessage());
        }
    }

    private static void getYearAndDay(String NIC) throws Exception {
        if (NIC.matches("([0-9]{2})(([05]((0[1-9])|([1-9][0-9])))|([1267][0-9]{2})|([38](([0-5][0-9])|([6-9][0-6]))))([0-9]{4})[VvXx]")) {
            year = Integer.parseInt("19" + NIC.substring(0, 2));
            day = Integer.parseInt(NIC.substring(2, 5));
        } else if (NIC.matches("([0-9]{4})(([05]((0[1-9])|([1-9][0-9])))|([1267][0-9]{2})|([38](([0-5][0-9])|([6-9][0-6]))))([0-9]{5})")) {
            year = Integer.parseInt(NIC.substring(0, 4));
            day = Integer.parseInt(NIC.substring(4, 7));
        } else {
            throw new Exception("NIC Doesn't Match any valid formats.");
        }
    }
}
```

APPENDIX G –CLIENT CERTIFICATE



New Sampath Caterers and Reception Hall

Undertakes wedding, homecomings, ceremonies & functions

Tel: 0112487708
Mob: 0712309662
E-mail: sampathdsrwedding@gmail.com

567 A,
Daranagama,
Delgoda.

04/11/2017
BIT Coordinator,
External Degree Center of UCSC,
221/2A,
Dharmapala Mw,
Colombo 07.

Dear Madam/Sir,

LETTER OF CERTIFICATION

This is certify that the *Reception Hall & Catering Management system* developed by Miss. K.S.N. Wijethunga (Registration No: 141819) has submitted successfully to us after granting the permission on the request for fulfillment of final project.

I'm pleased to inform for that the proposed system for the reception hall & catering management system, has fulfilled all requirement that the organization required. Proposed system makes smooth the operation of the business making process to achieve business goals of the organization.

It is happy to mention that the proposed system can be implemented successfully for reception hall & catering management for our company is acquires all the requirements in the best level.

Thank you!

Yours faithfully,

[illegible]

Manager
New Sampath Caterers & Reception hall

D. S. R. SAMPATH
(Proprietor)
New Sampath Caterers and
Reception Hall
567A, Daranagama Delgoda.
Reg. No. 6148

INDEX

GLOSSARY

- **Use case diagram**-use case diagram represent the how users interact with the system
- **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 vi ews.
- **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.
- **JavaFX**- is a software platform for making and providing rich internet applications (RIAs are) that can run across a wide variety of devices.
- **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.
- **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.

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