



**SALON MANAGEMENT SYSTEM
FOR
“SALON NIROSHA”**

G.J.V.P.S.O. JAYAWARDENA

Registration Number: R141179

Index Number: 1411799

Name of the Supervisor: Ms. Kasuni Gayanthika Sugathadasa

November 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

Declaration


I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference 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 organizations.



G.J.V.P.S.O. Jayawardena

Date: 02/11/2017

Countersigned by:



Ms. K.G. Sugathadasa

Date: 02/11/2017

Abstract

'Salon Nirosha' which is a hair and beauty parlor currently use a manual procedure to deal with its management processes. Its customers do not have a proper way to make an appointment; other than making a call or visit the Salon premise. Salon owner, employees and customers need to keep reminders on their mobiles over appointments. Salon owner and her employees maintain a diary to note down the appointment details. Service details of the salon are written in papers; which always leading for misplacements. Owner need to write all the service details at a new paper once it get misplaced or updated. Once a payment has done, customer will receive a hand written receipt (from the manual receipt book), and the cashier is keeping a copy of the same receipt. There is a higher risk of misplacing the receipt copies. Owner has no proper way to manage her employees and clients.

This project was done to solve the abovementioned problems prevails at 'Salon Nirosha'. It was expected to provide a Salon Management System Software to manage her employees, clients, client appointments, reminders, services, resources and payments while providing a dashboard to view information and generating crucial reports to support the higher managerial decisions.

Project development was based on OO (Object Oriented) concepts. Java was selected as the programming language; basically jsp and servlet frameworks were used. STS (Spring Tool Suit) was the IDE (Integrated Development Environment) used. Apache Tomcat was used as the server, and pgAdmin as the database. Improved version of windows starts from version 7 was expected to be used as the OS (Operating System). Several design tools like: MS Visio, Visual Paradigm etc., used at design drawing. Entire project was developed under Iterative Waterfall SDLC Methodology.

Salon Management System which was fully tested using a variety of testing methods (such as unit, integration and system testing) was implemented at the client site successfully. The user manual handover to the salon employees was seemed to be very useful for them to work with the system. The highest satisfactory feedback of the Client was able to gain after the user acceptance testing.

Acknowledgements

Initially I would like to honor my sincerest gratitude for the Coordinator and the project examination board of UCSC (University of School of Computing) of Bachelor of Information Technology (BIT) degree program; for providing me the opportunity to do a final year project as to enhance our skills and to augment our carrier path.

Next I would declare my acknowledgement to the supervisor Ms. Kasuni Gayanthika Sugathadasa; as she directed me for the successful completion of the project while giving me the technical guidance; by devoting her valuable time to measure the progress of my project.

Then I would pay my thankfulness to the client Ms. Nirosha Wanasekara, who is the owner of 'Salon Nirosha', for backing as the client of my final year project and for helping me to gather the business requirement successfully by spending her valuable time on me.

I would be really thankful to the lecture panel at OpenArc School of Business and Technology for giving instructions at most stages of the project when needed.

For my parents I pay my heartiest gratitude; as they always lift me up in success by standing either side of each step in my life.

Finally I would like to award my honest gratitude for every person who helped me a lot, directly as well as indirectly, by giving me their utmost support, time and instructions in many ways to make my final year project a success.

Table of Contents

Declaration.....	ii
Abstract.....	iii
Acknowledgements.....	iv
Table of Contents.....	v
List of Figures.....	vii
List of Tables.....	ix
List of Acronyms.....	x
Chapter 1 - Introduction.....	1
1.1 Motivation for the Project:.....	1
1.2 Objectives of the Project.....	2
1.3 Scope of the Project.....	2
1.4 Outline on Other Chapters.....	3
Chapter 2 - Analysis.....	4
2.1 Failures of Current System.....	4
2.2 Use Case for the Current System.....	5
2.3 Existing Similar Systems.....	7
2.4 Requirements.....	10
2.4.1 Functional Requirements.....	10
2.4.2 Non-Functional Requirements.....	11
2.5 System Users and their involvement at the system.....	12
2.5.1 System Administrator.....	12
2.5.2 Stylist/ Other Employee.....	12
2.5.3 Cashier.....	12
2.6 Selected Methodology.....	13
Chapter 3 - Design.....	14
3.1 Alternate Solutions.....	14
3.2 Design Methodology.....	15
3.2.1 ER Diagram.....	15
3.2.2 Use case Diagrams.....	15
3.2.3 Activity Diagrams.....	18
3.2.4 Class Diagram.....	20

3.3	Table Structures	21
3.4	User Interface Designs.....	23
Chapter 4 - Implementation		25
4.1	Implementation Environment.....	25
4.2	Development Tools.....	26
4.3	Code and Module Structure.....	27
4.4	Codes Reused	31
4.5	Interaction between the System Modules.....	33
Chapter 5 - Evaluation		35
5.1	Test Plan	36
5.2	User Evaluation	39
5.3	Summary Chart Review.....	40
Chapter 6 – Conclusion.....		41
6.1	Lessons Learnt	41
6.2	Problems Encountered	42
6.3	Future Developments.....	42
References.....		43
Appendix A – System Documentation		45
Appendix B – Design Documentation		51
Appendix C – User Documentation		61
Appendix D – Management Reports.....		66
Appendix E – Test Results		69
Appendix F – Code Listing		76
Appendix G – Client Certificate.....		82
Glossary		83
Index.....		84

List of Figures

Figure 2.1: Use case for the Current System	6
Figure 2.2: Appointment Screen of belliatia	7
Figure 2.3: Add New Client Screen of belliatia	8
Figure 2.4: New Client Screen of schedule	8
Figure 2.5: Phases of Iterative Waterfall methodology	13
Figure 3.1: ER Diagram for the Entire System	15
Figure 3.2: Use case diagram for the cases of System Administrator Perspective	16
Figure 3.3: Use case diagram for System User Perspective	17
Figure 3.4: Use case diagram for Cashier Perspective	17
Figure 3.5: Use case diagram for Customer Perspective	17
Figure 3.6: Activity Diagram for Appointment Handling	18
Figure 3.7: Activity Diagram for Payment Handling	19
Figure 3.8: Class Diagram of the System	20
Figure 3.9: Login Screen of the System	23
Figure 3.10: Services Management Screen of the System	24
Figure 3.11: Event Calendar of the system.....	24
Figure 4.1: Tools set used at system development.....	27
Figure 4.2: Package structure of the System	28
Figure 4.3: Database Connection Code	29
Figure 4.4: Date Formatter Code.....	30
Figure 4.5: Sample web.xml Code	30
Figure 4.6: Code use to retrieve the Town for the selected Postal Code	31
Figure 4.7: Code use to carry Service Ids to Staff-Service Mapping Table	31
Figure 4.8: Code use to retrieve the NIC Number using the Date of Birth	32
Figure 4.9: Diagram on MVC Architecture.....	34
Figure 5.1: User Evaluation Form.....	39
Figure 5.2: Chart on User Evaluation Results	40
Figure A.1: JDK	45
Figure A.2: Advance System Setup Screen	46
Figure A.3: System Properties screen	46
Figure A.4: JAVA_HOME Setup Screen	47
Figure A.5: PostgreSQL Installation Screen	48
Figure A.6: Password entering Screen	48
Figure A.7: Port entering Screen.....	49
Figure A.8: Setup Installing Progress Screen	49
Figure A.9: Connect to the SQL Database Server	50
Figure B.1: Use Case for Staff Registration	51
Figure B.2: Use Case for Client Registration	52
Figure B.3: Use Case for Staff Profile Management	53
Figure B.4: Use Case for Client Profile Management	54
Figure B.5: Use Case for Staff Leave Management	55

Figure B.6: Use Case for Appointment Creation	56
Figure B.7: Use Case for Appointment Reschedule	57
Figure B.8: Use Case for Appointment Status Management	58
Figure B.9: Use Case for Payment Application Process	59
Figure C.1: System Login Screen	61
Figure C.2: System Home Page	62
Figure C.3: System Main Menu	62
Figure C.4: Client Registration Screen	63
Figure C.5: Appointment Creation Screen	63
Figure C.6: Appointment Status Handling Screen	64
Figure C.7: Payment Application Module	64
Figure C.8: Sample Invoice Printed after the Payment completion.....	65
Figure C.9: Pie Chart generate using the Appointment statuses	65
Figure D.1: Service wise Sales by Staff Report	66
Figure D.2: Appointment wise Sales by Staff Report.....	66
Figure D.3: Service wise Income by Client Report	67
Figure D.4: Appointment wise Income by Client Report	67
Figure D.5: Appointment Status Report	68
Figure E.1: User Evaluation Form filled by Owner/ System administrator	73
Figure E.2: User Evaluation Form filled by Stylist.....	74
Figure E.3: User Evaluation Form filled by Cashier	75
Figure F.1: Sample Select Query Code	76
Figure F.2: Sample Create JSP Code	77
Figure F.3: Sample Create Servlet Code	77
Figure F.4: Create related DAO	78
Figure F.5: Modify related JSP	78
Figure F.6: Modify related Servlet.....	79
Figure F.7: Modify related DAO.....	79
Figure F.8: Delete related JSP	79
Figure F.9: Delete related Servlet	80
Figure F.10: Delete related DAO	80
Figure F.11: Query to select data from ‘Appointment Status’ table	81
Figure F.12: Javascript required to run the pie chart at the system.....	81

List of Tables

Table 2.1: Criticism on Existing Similar Systems Table.....	9
Table 2.2: Comparison with similar systems and developed system.....	9
Table 3.1: Table Structure of Service Table.....	21
Table 3.2: Table Structure of Appointment Table.....	21
Table 3.3: Table Structure of Client Registration Table.....	22
Table 4.1: Descriptive Table on Tools set used for the Development.....	27
Table 5.1: Test Case for the Entire System.....	38
Table B.1: Use Case Description for Staff Registration.....	51
Table B.2: Use Case Description for Client Registration.....	52
Table B.3: Use Case Description for Staff Profile Management.....	53
Table B.4: Use Case Description for Client Profile Management.....	54
Table B.5: Use Case Description for Client Leave Management.....	55
Table B.6: Use Case Description for Appointment Creation.....	56
Table B.7: Use Case Description for Appointment Reschedule.....	57
Table B.8: Use Case Description for Appointment Status Management.....	58
Table B.9: Use Case Description for Payment Application.....	59
Table E.1: Actual Test Results Table.....	72

List of Acronyms

CRUD	-	Create Read Update Delete
ERD	-	Entity-Relationship Diagram
IDE	-	Integrated Development Environment
JSP	-	Java Server Pages
MS	-	Microsoft
MVC	-	Model View Controller
OO	-	Object Orient
ORDBMS	-	Object-Relational Database Management System
OS	-	Operating System
SDLC	-	Software Development Life Cycle
UI	-	User Interfaces

Chapter 1 - Introduction

A beauty salon or beauty parlor is an establishment dealing with cosmetic treatments for men and women. Other variations of this type of business include hair salons and spas [1].

1.1 Motivation for the Project:

Nowadays it is easy to find Salons, Parlors and Spas at any nook and corner in this world, as beauty and fashion has become a great trend disregarding the age limit and gender. All the Salon Owners are relying on their Customers; who visits Salons to fulfill their own beauty needs. Therefore it is important to value the loyalty and the time of those Customers. From the other side, to make the Salon services better and to enhance the good names of the salons, their Stylist's/Employee's satisfaction also needs to be fulfilled. Not only that the income earns by selling services to the clients should be able to monitor by the salon owners.

Using manual procedures may pave the way for a variety of obstacles when satisfying Customers and Employees of the respective Salons. Valuable time and money of the Salon Owners, Employees and Customers get waste unnecessarily due to these manual dealings. These barriers make direct harms for the incomes and the good names of the Salons and for their Owners.

1.2 Objectives of the Project

Objectives of doing this project are to;

- Eliminate the paper based work use at the Salon premise such as, usage of diaries to note down appointment details, writing manual invoices for the payments done by the Customers etc.
- Eliminate the data redundancy; keeping appointment details at several places (Diary, mobile etc.) by several people (Owner, Employees, Customers etc.).
- Abolish the wastage of time, resources, efforts and money of the Employer, Employees and Customers (Stakeholders).
- Improve the efficient and effectiveness of the Salon management activities, services and processes like maintaining Customers, Employees, Appointments and Payments etc.
- Ease the management and decision making while improving Salons' good name.
- Improve the Client satisfaction and Employee satisfaction.
- Enhance the Stakeholder integration.

1.3 Scope of the Project

- Providing the facility to registering Salon staff and maintaining their details.
- Providing the facility to registering regular Customers and maintaining their details.
- Facilitate appointment handling.
- View appointments leaves and holidays through an event calendar.
- Handling Salon Services along with their respective prices, hours etc.
- Providing Customer Payment handling option.
- Generating invoices through the system.
- Generating reports to support the higher managerial decisions.
- Maintaining an information center (dashboard).
- Reminder generating facility through emails

1.4 Outline on Other Chapters

The rest of this document is organized as follows.

Chapter 2: Analysis

Provide a top level use case diagram on the existing system; a review on two similar software of the proposed system; analysis on the requirement; and a brief description on the selected Software Development Life Cycle (SDLC) methodology.

Chapter 3: Design

How the project requirements are satisfied through the used alternatives was pointed out. Several User Interfaces (UIs) were included along with the main designed diagrams as to provide a clear picture on the system structure.

Chapter 4: Implementation

The major code and module structures were explained with a diagram to further clarify the interaction between modules of the system. In advance an explanation on implementation environment (hardware and software), reused code, development tools, and platform dependence were given.

Chapter 5: Evaluation

Through this chapter a comprehensive test plan that was used to verify and validate the system was provided along with the test results. Tools used for feedback collection also mentioned additionally.

Chapter 6: Conclusion

Critical evaluation of the system and suggestions for the future work were described at this chapter.

Chapter 2 - Analysis

Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose [2].

2.1 Failures of Current System

The manual procedure used by 'Salon Nirosha' stakeholders before implementing the project, faced a lot of troubles at their day today work (at the salon). Salon owner was not used a software to manage her salon work load. So all the stakeholders met with a lot of difficulties at following instances all the time.

Salon Owner was faced the following problems:

- Difficult to identify regular customers.
- Difficult to maintain her-own staff members and their leaves.
- Had to note down, cancel and change appointments along with their dates, times, services and payments regularly.
- Had to keep daily reminders on her mobile over clients' appointment details.
- Had to allocate prices for services and update them from time to time in papers.

Employees were faced the following problems:

- They always had to turn the diary pages to search for appointment details.
- Had to always contact/ message the customers via a mobile to announce the appointment statuses.
- Mixed up with the shifts, leaves and holidays, as there was no proper way of managing them.

Cashier was faced the following problems

- Met with difficulties when finding prices respective to each service.
- Manual bill issue. All the service details, prices etc. were written in it by hand.
- Totaling the bills manually with/ without using calculators was difficult to handle.

Customers also met with certain difficulties as below;

- Mostly had to visit the salon premises to make appointments (Specially the new clients).
- Had to keep daily reminders on mobiles over appointment.
- Had to contact the salon via a mobile to know the appointment statuses.
- Had to wait a considerable time, till issuing the manual payment receipt.

Hence it was essential to automate the activities managed by the salon to improve Salon Nirosha' service quality and save their time while improving their Customers' satisfaction.

2.2 Use Case for the Current System

Use case diagram designed to display the existing manual system at 'Salon Nirosha', is shown by Figure 2.1. This displays how the manual process was went on at 'Salon Nirosha'.

Salon Management System for 'Salon Nirosha'

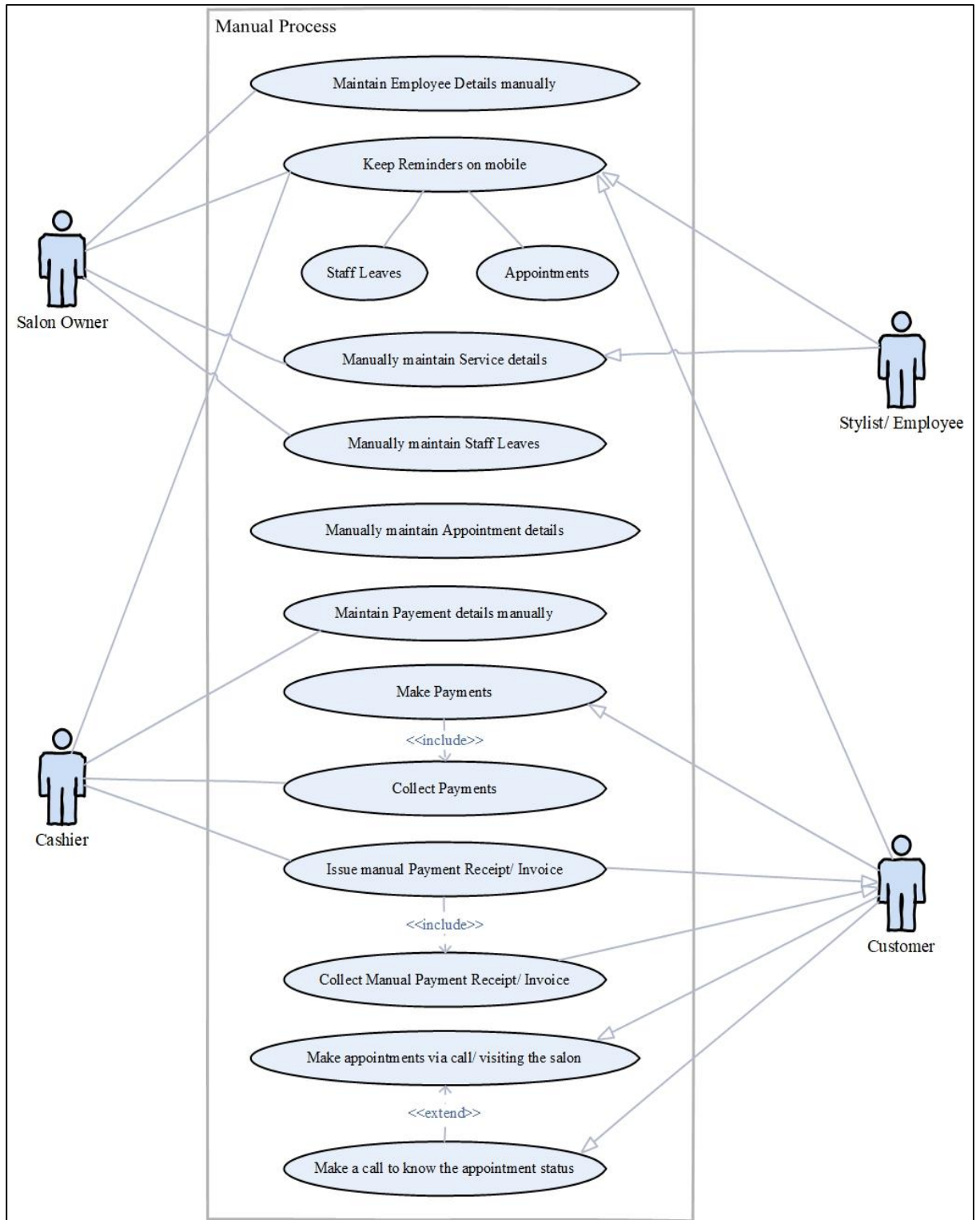


Figure 2.1: Use case for the Current System

2.3 Existing Similar Systems

It was very difficult to find out existing similar systems, as most of the salons use only websites to promote their services. Similar salon management systems found after a huge search were thoroughly studied as to identify functional and non-functional requirements; which have been needed to include to the developed system.

belliata.com:

A software which provide features to the client (owners of salons) to carry out their salon management functions efficiently. For advance features clients should pay for them. Figure 2.2 displays the appointment creation screen at 'belliata.com' site.

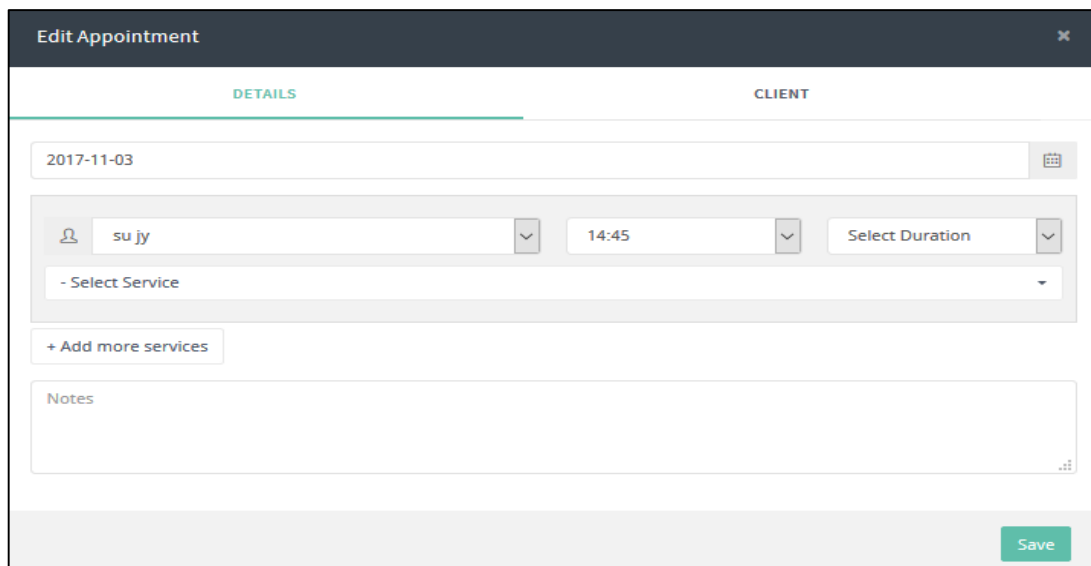


Figure 2.2: Appointment Screen of belliata

Figure 2.3 displays the appointment creation screen use by ‘belliata.com’ site.

Figure 2.3: Add New Client Screen of belliata

shedul.com:

This is also an existing salon management software which consists of very useful features to the client (owners of salons). Figure 2.4 displays the client creation screen at ‘schedule.com’ site.

Figure 2.4: New Client Screen of schedule

Criticism:

Following table at Table 2.1 displays the criticism done on the similar systems with the developed system. It will prove how profitable using the developed salon management system than using the similar systems.

Belliata	Schedule	Salon Nirosha
Need internet to use		No need internet to use
Additional Payment should need to done for the enhancement of features.		Required features are provided and future enhancements can be done if needed.
The systems saved data at a shared database; therefore the privacy of data cannot be assured.		Database is not a shared one; therefore no security issue.

Table 2.1: Criticism on Existing Similar Systems Table

But it is required to say that belliota and schedule systems consist of advanced features like online appointment creation and credit card payments etc.

Comparison:

It was decided to perform a feature comparison also in between the three systems as to identify the weaknesses with the ‘Salon Nirosha’ salon management system compared to the other two; for its further improvement. The comparison is shown by the Table 2.2.

Feature	Belliata	Schedule	Salon Nirosha
User can register Clients.	Yes	Yes	Yes
User can make offline appointments.	Yes	Yes	Yes
User can make online appointments.	Yes	Yes	Implement in the future.
User can do the payment and print invoice.	Yes	Yes	Yes

Table 2.2: Comparison with similar systems and developed system

2.4 Requirements

Basically, Non-functional requirements describe 'how the system works', while functional requirements describe 'what the system should do' [3]. Observations and Interviews were conducted as facts gathering methods at the requirement gathering phase in order to gather requirements.

2.4.1 Functional Requirements

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail [4].

- Create salon staff and maintaining (Update/ Inactive) their details.
- Create regular customers and maintaining (Update/ Inactive) their details.
- Create salon services and maintaining (Update/ Delete) their details such as prices, hours etc.
- Maintain resources (Create/Update/ Delete) at the salon premise.
- Reminder generating facility provide through the system and send via mails for all the respective stakeholders.
- Maintain holidays and staff leaves.
- Facilitate appointment handling through an event calendar by the system.
- Providing customer payment handling option.
 - Enter payment details for the system.
 - View customers' payment balance details.
 - View customers' gross payment details.
- Generating invoices through the system.
- Generating reports to support the higher managerial decisions.
- Maintaining an information center to display crucial data charts.

2.4.2 Non-Functional Requirements

Basically, non-functional requirements relate to qualities of the system that cut across user facing features, such as security, reliability, and performance [5].

- Accessibility – The system is able to be access anywhere at any time by the authorized users.
- Accuracy – The correctness of data inputs to the system was ensured.
- Availability – System is available within working hours. But can be used at special occasions also. E.g.: - At a bridal dressing
- Efficiency – Users were given the facility to perform the salon management processes correctly through the salon sales record management system.
- Effectiveness – Users were given the facility to perform correct salon management processes via the suggesting system.
- Maintainability – This is a considerable factor especially for a non-technical user. Maintainability of the system is not more complex.
- Privacy – The confidentiality of the data inputs to the system has been assured.
- Reliability - Ability of the suggested system to function under stated conditions for a specified period of time has been assured.
- Robustness – When handling payments this function was considered.
- Security – The data feeds to the system has been protected by controlling the user access privileges.

2.5 System Users and their involvement at the system

Mainly there are three system users who need to access the system. They are as follows:

1. System administrator
2. Stylist/ Employee
3. Cashier

2.5.1 System Administrator

- Allow creating Employees/ System Users.
- Allow creating regular Clients.
- Allow maintaining reference data.
- Allow allocating leaves for the staff members.
- Allow generating crucial Reports.
- Allow viewing crucial Reports.
- Allow viewing Information Centre.
- Allow viewing payment information.
- Allow viewing appointments, leaves and holidays calendar.

2.5.2 Stylist/ Other Employee

- Allow creating appointments.
- Allow maintain appointment details.
- Allow maintaining appointment statuses.
- Allow viewing appointments, leaves and holidays calendar.

2.5.3 Cashier

- Allow doing payments.
- Allow printing the invoices.
- Allow viewing appointments, leaves and holidays calendar.

2.6 Selected Methodology

Iterative Waterfall methodology is selected as the SDLC methodology of the project; as the major requirements of the whole system were gathered at the beginning and were able to define clearly.

Iterative waterfall methodology consists of the steps given at the Figure 2.5:

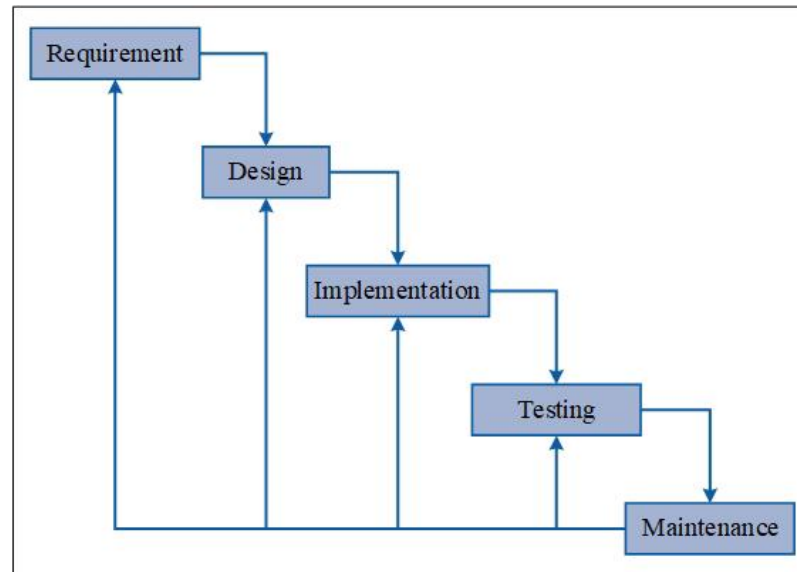


Figure 2.5: Phases of Iterative Waterfall methodology

Iterative waterfall methodology best fits for a project where “Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time” [6], and it consists with the following advantages;

- Some working functionalities were able to be developed quickly and early at the life cycle.
- Results were obtained early and periodically.
- Parallel development was able to be planned.
- Progress was able to be measured.
- It was less costly to change the scope/requirements.
- Testing and debugging during smaller iteration was easy to attempt.

Chapter 3 - Design

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system [7]. This chapter devotes factors on the designing phase of the developed project.

3.1 Alternate Solutions

There are free software systems designed specifically for salon management purposes can be found in the internet. They are able to download freely and use at the salon premises as an alternative solution. But those software are consists with common and limited features. Customizations are not allows by most of those software. Therefore salon owner need to be adapted according to the processes follows by the specific software neglecting their-own process of working.

As to find the most suitable solution for the system to be developed, an evaluation was done in between several identified solutions.

Web based Solutions:

These are compatible with cross platforms, easily manageable and easy to update. It is possible to achieve a far greater level of interoperability between web applications than it is with isolated desktop systems. This type of solutions are able to access easily at anywhere by any time using a web browser [8].

Standalone Solutions:

This type of computer setup requires installing applications on individual machines. They need a considerable time to load the system application at the relevant machine. But standalone has no bandwidth problems as they need no internet usage.

By analyzing these solutions it was decided to provide a 'Web based system Solution' for the client.

3.2 Design Methodology

Object Orientation concept has been used as the design methodology since objects and classes based design diagrams like Use cases, Activity diagrams, Class diagrams and State diagrams are easy to understand.

3.2.1 ER Diagram

An entity-relationship (ER) diagram represents entities and their relationships graphically using the data within databases or information systems. Figure 3.1 represents the relationship between the entities at the database, which use to store all the relevant system data respectively.

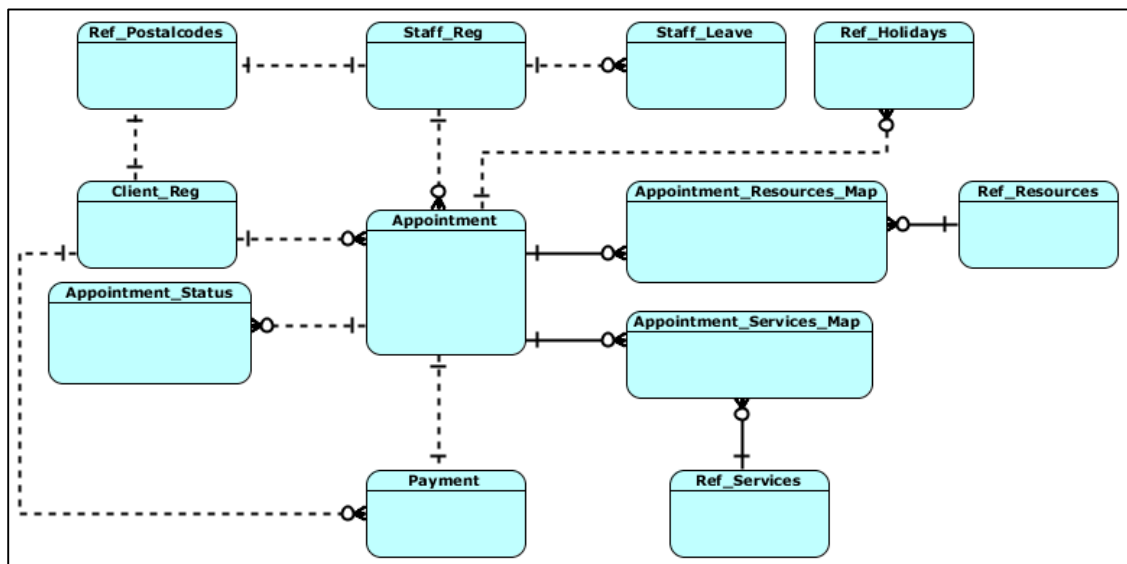


Figure 3.1: ER Diagram for the Entire System

3.2.2 Use case Diagrams

Use Case diagrams use to describe activities (use cases) performed by the system (subject) in collaborate with one or more external users of the system (actors).

Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5 displays the use case diagrams designed for the System Administrator, shows the use case designed for the System User perspective, use case designed for the Cashier perspective and use case designed for the Customer perspective accordingly.

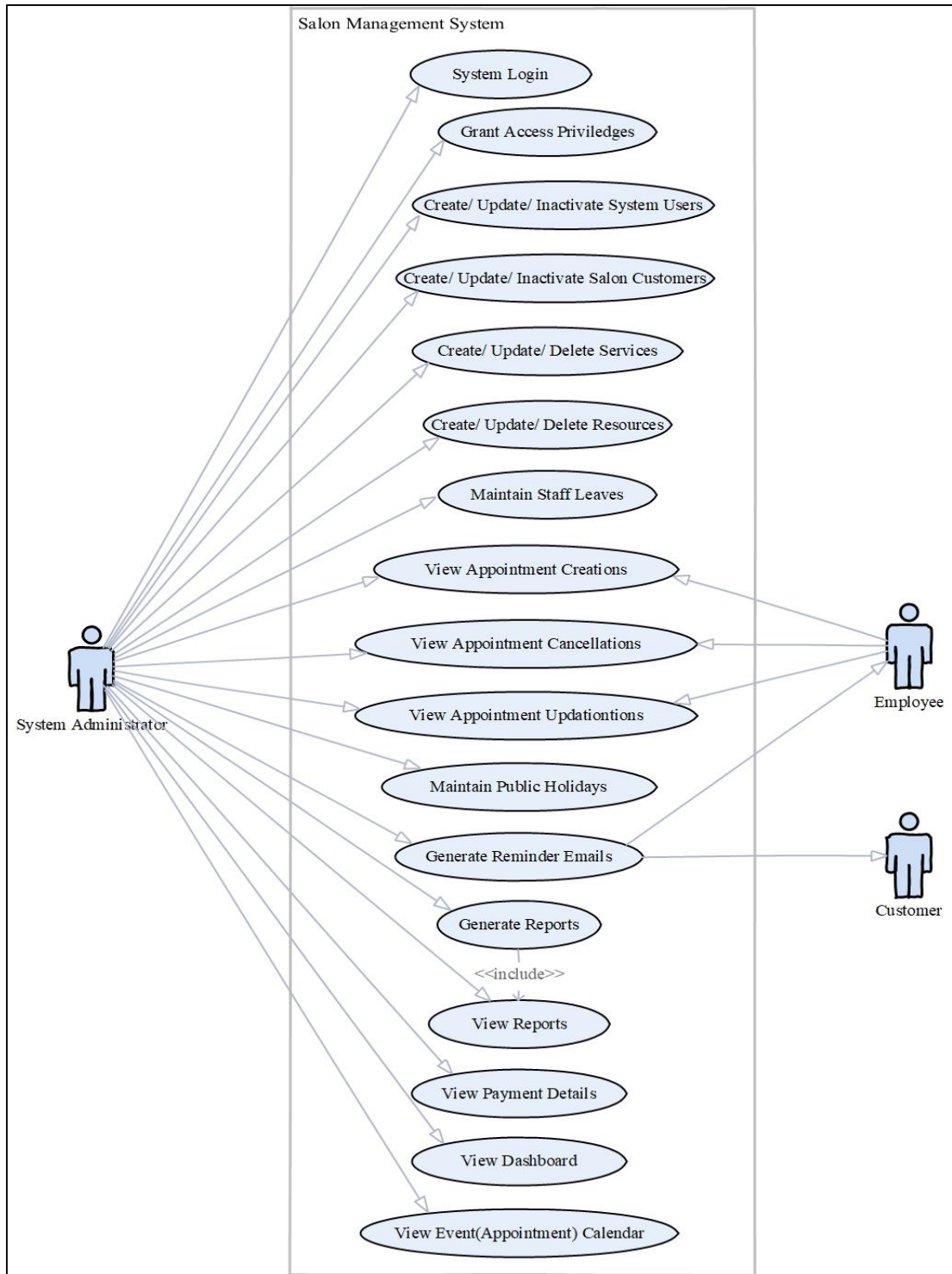


Figure 3.2: Use case diagram for the cases of System Administrator Perspective

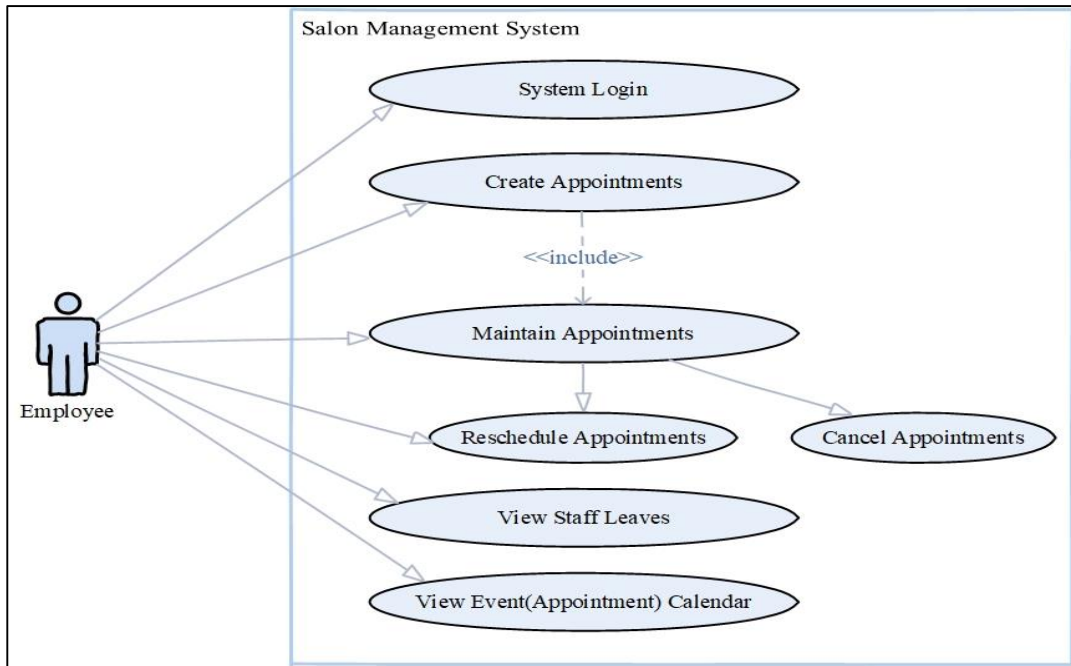


Figure 3.3: Use case diagram for System User Perspective

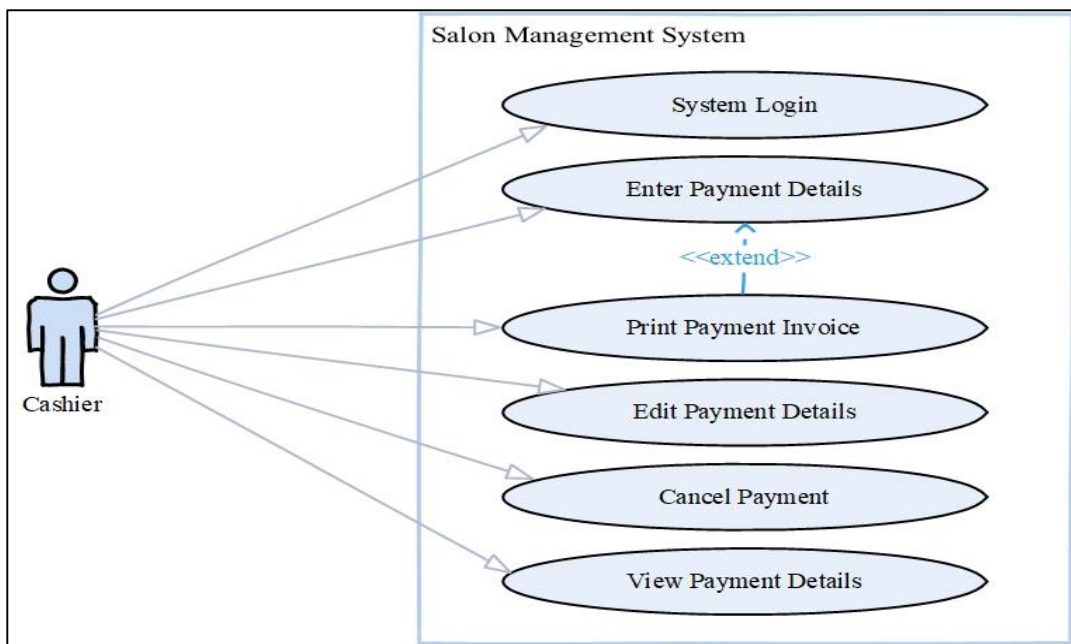


Figure 3.4: Use case diagram for Cashier Perspective

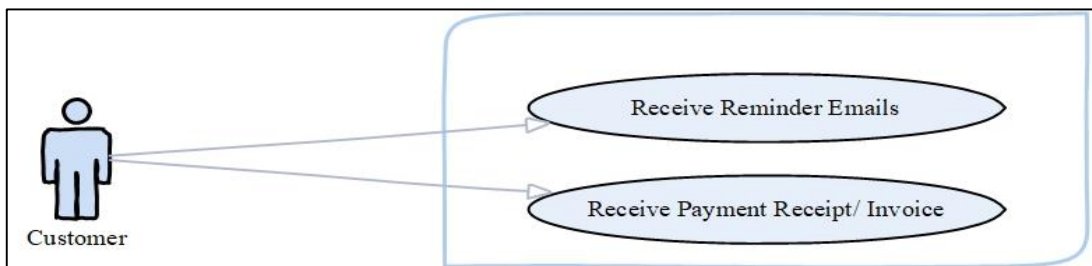


Figure 3.5: Use case diagram for Customer Perspective

3.2.3 Activity Diagrams

Activity Diagrams used to display workflows of stepwise activities followed by the system at special occasions. Demonstrate the logic of an algorithm, describe the steps performed in a UML use case or illustrate a business process or workflow between users and the system via an Activity /diagram.

Figure 3.6 activity diagram designed to give a clear representation on Appointment Handling process related activities.

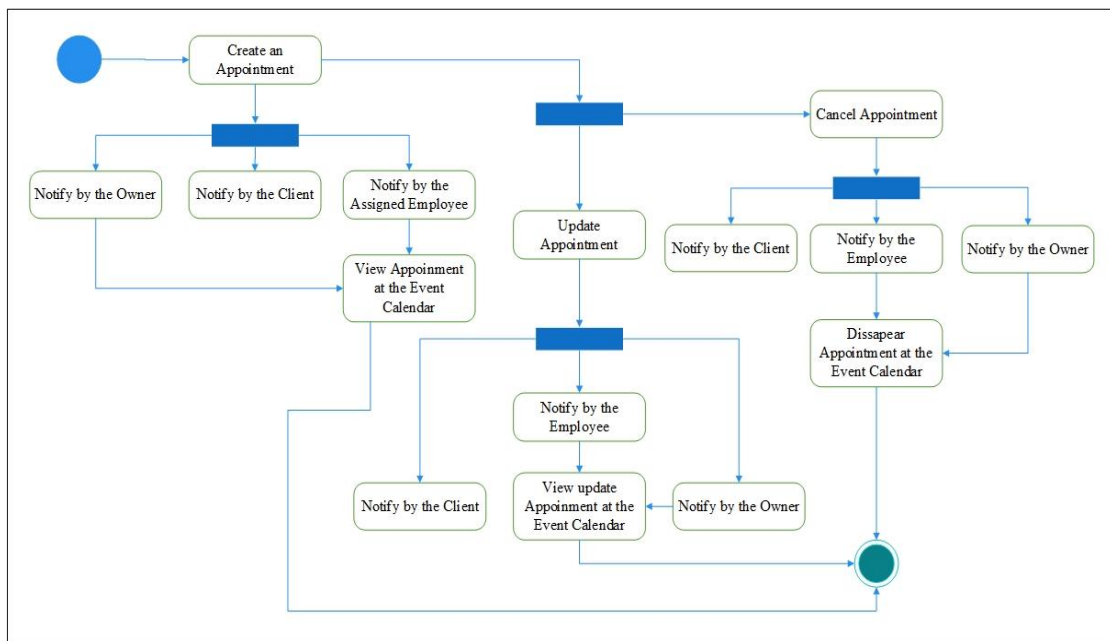


Figure 3.6: Activity Diagram for Appointment Handling

Figure 3.7 displays activity diagram designed to give a clear representation on Payment Handling process related activities; used relevant actions, connectors and decision points etc.

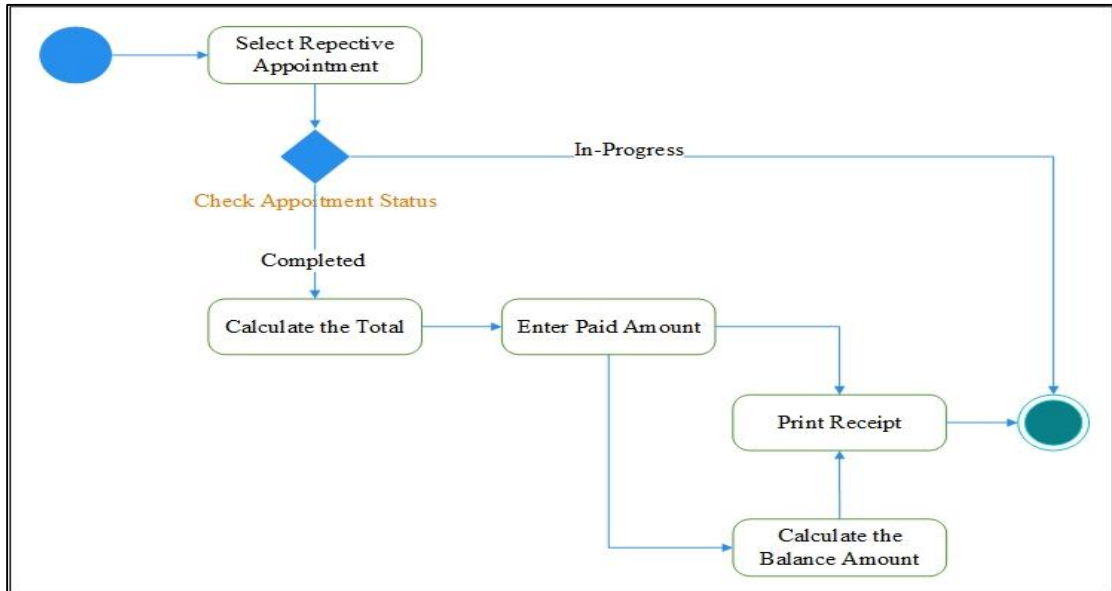


Figure 3.7: Activity Diagram for Payment Handling

3.2.4 Class Diagram

Class diagram describes the attributes, operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages [9].

Figure 3.8 shown below represents the class diagram designed based on the ER diagram for the entire system.

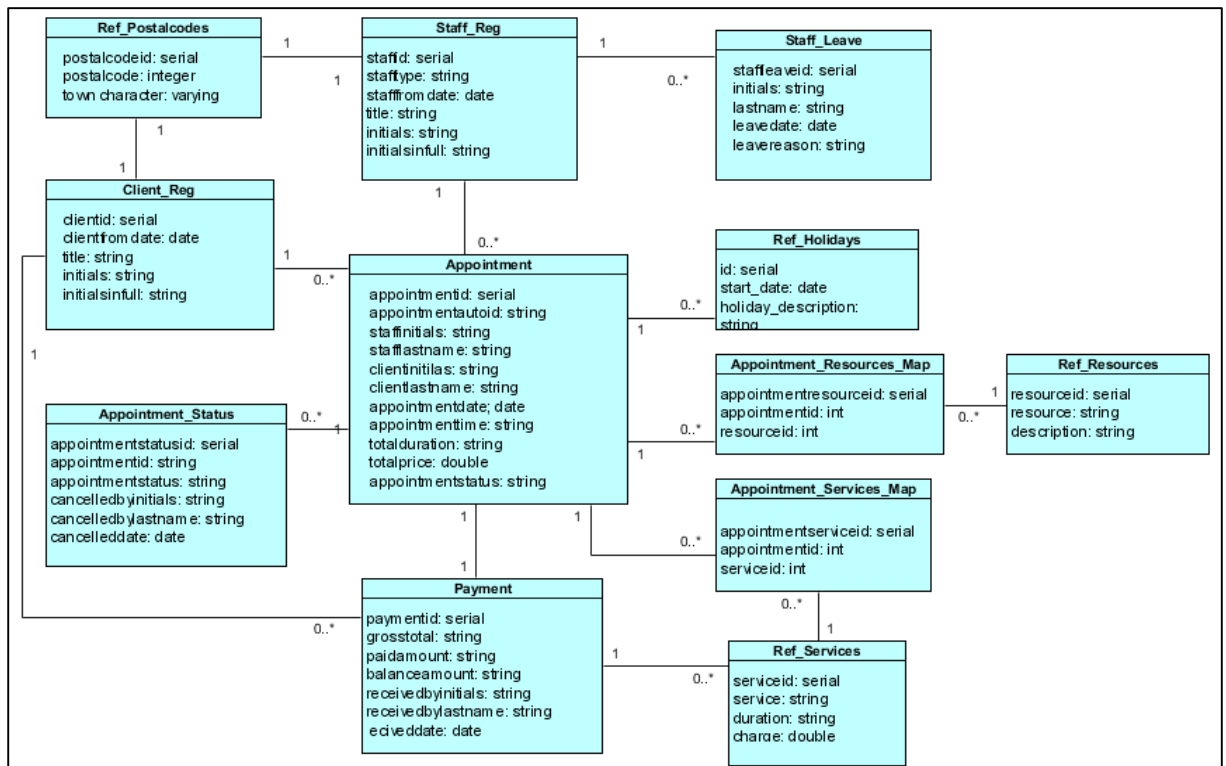


Figure 3.8: Class Diagram of the System

3.3 Table Structures

Designed of table structures were included at this chapter to provide a knowledge on the tables used to store data at the database.

Table structure at Table 3.1 is consist on service data based for the reference module of the system.

Attribute	Description	Data Type	Length	Key
serviceId	Service id	SERIAL	-	PK
serviceName	Service name	VARCHAR	50	-
serviceDuration	Service duration	VARCHAR	50	-
serviceCharge	Service charge	DOUBLE PRECISION	-	-

Table 3.1: Table Structure of Service Table

Table structure at Table 3.2 is consist on appointment data based for the appointment management module of the system.

Attribute	Description	Data Type	Length	Key
appointmentId	Appointment id	SERIAL	-	PK
appointmentautoId	Appointment auto id	CHAR	7	-
staffInitials	Staff initials	VARCHAR	10	-
staffLastname	Staff last name	VARCHAR	60	-
clientInitials	Client initials	VARCHAR	10	-
clientLastname	Client last name	VARCHAR	60	-
appointmentDate	Appointment date	DATE	-	-
appointmentTime	Appointment time	TEXT	-	-
totalDuration	Total duration	TEXT	-	-
totalPrice	Total price	DOUBLE PRECISION	-	-
appointmentStatus	Appointment status	VARCHAR	10	-

Table 3.2: Table Structure of Appointment Table

Table structure at Table 3.3 is consist on client data based for the client management module of the system.

Attribute	Description	Data Type	Length	Key
clientid	Client id	SERIAL	-	PK
clientfromdate	Client from date	DATE	7	-
title	Client title	VARCHAR	6	-
initials	Client initials	VARCHAR	50	-
initialsinfull	Client initials in full	VARCHAR	255	-
lastname	Client last name	VARCHAR	60	-
nic	NIC number	VARCHAR	12	-
dob	Date of birth	DATE	-	-
maritalstatus	Marital status	TEXT	10	-
addressline1	Address line 1	VARCHAR	150	-
addressline2	Address line 2	VARCHAR	150	-
district	District	VARCHAR	50	-
province	Province	VARCHAR	50	-
clientstatus	Client status	VARCHAR	10	-
landnumber	Land number	CHARACTER	12	-
mobilenumber	Mobile number	CHARACTER	12	-
faxnumber	Fax number	CHARACTER	12	-
email	Email address	VARCHAR	150	-
autoclientid	Auto client id	VARCHAR	10	-
postalcodeid	Postal code id	INTEGER	10	FK

Table 3.3: Table Structure of Client Registration Table

3.4 User Interface Designs

When designing the user interfaces, it was focused on what users might need to do and ensured that the interface has elements that are easy to access, understand, and use to facilitate those actions. The designed UIs were a collaboration of interaction design, visual design, and information architecture concepts.

Best practices on UI designing like;

- keeping the interface simple, using common UI elements and icons,
- maintaining color, light, contrast, and texture,
- using typography to create hierarchy and clarity were handled within the system up to the maximum [10].

Shown below are some of the screens designed for the system. Figure 3.9 is the login screen which use to login to the developed system and Figure 3.10 is a sample of an input form designed to manage salon services.

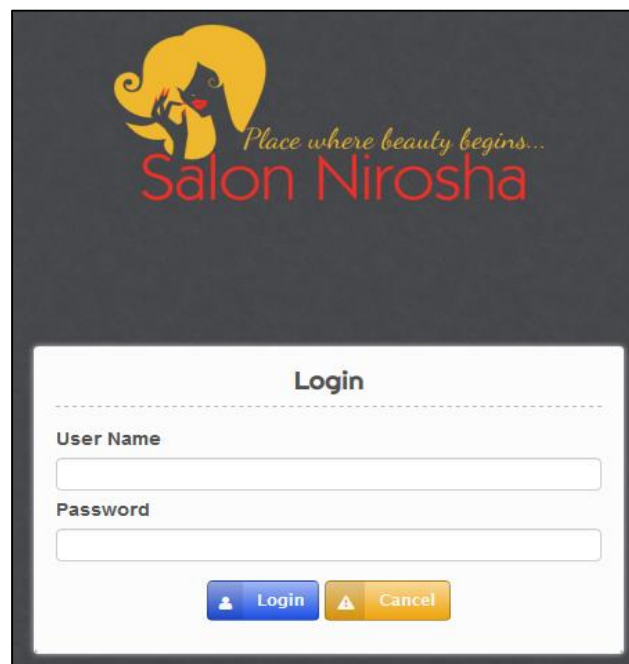


Figure 3.9: Login Screen of the System

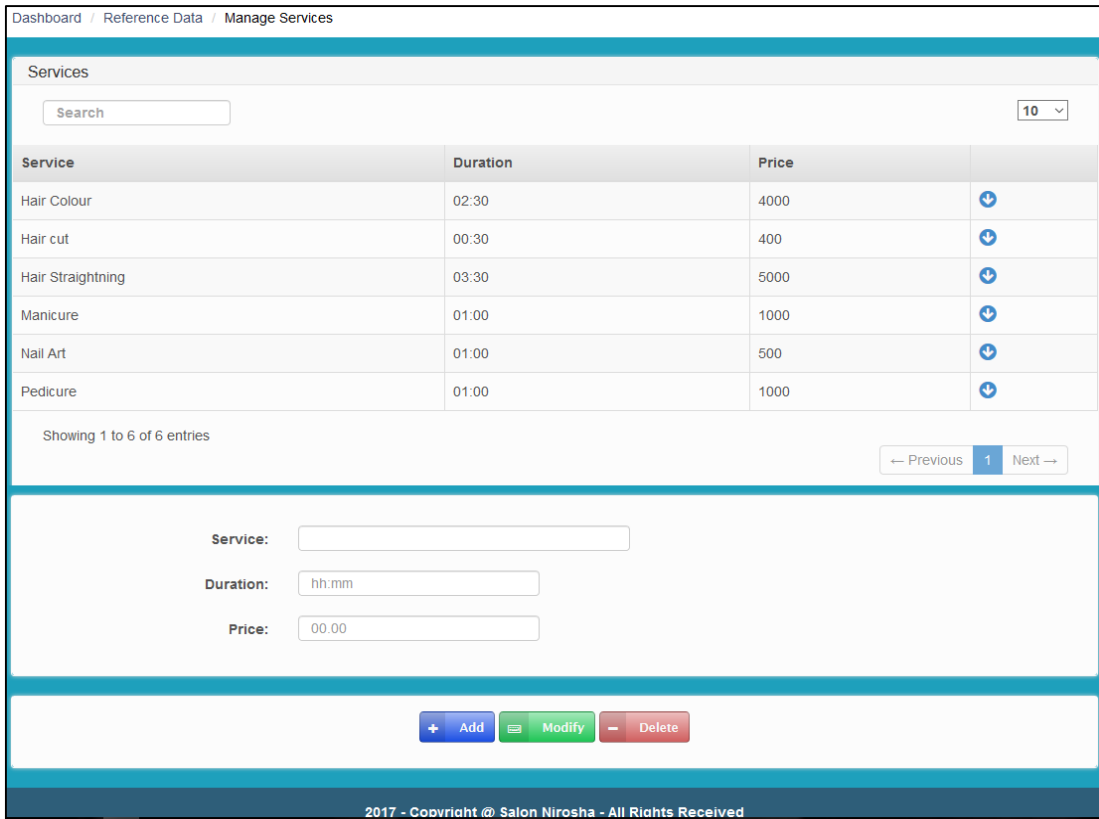


Figure 3.10: Services Management Screen of the System

Following shown Figure 3.11 is the event calendar use by the system to view appointments, leaves and holidays respectively.

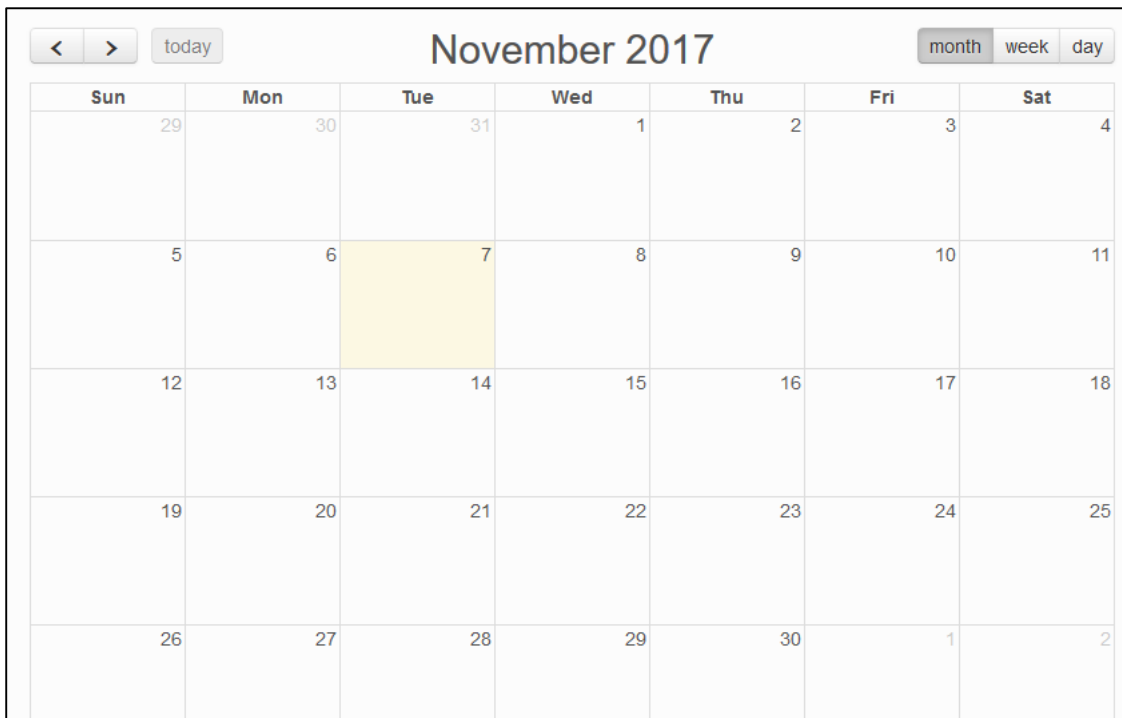


Figure 3.11: Event Calendar of the system

Chapter 4 - Implementation

Each model (which was a collection of features) designed at the designed phase transfer into system by using code, this is the main task of the software developers. Separated modules were turned into a code at this phase of the SDLC. Actually implementation phase considered as the lengthiest phase at SDLC.

This chapter proves how the implementation was done regard to the project.

4.1 Implementation Environment

The system implementation environment is of two separate parts as:

1. Hardware Environment
2. Software Environment

Hardware Requirements:

- Personal Computer or laptop with Core 2Duo CPU or above
- RAM: At least 2 GB
- Hard Disk Space: 30 GB or above
- Processor Speed: From 1.2 GHz
- Desktop or Laptop, Keyboard, Mouse and System Unit (only for a PC)

Software Requirements:

- OS: Windows improved version from 7
- Java development kit (JDK 7).
- IDE: Spring Tool Suite – version 3.5.1
- Apache Tomcat Application Server - version -7.0.42
- Database PostgreSQL Server - version 9.4.1
- Web browser – Google Chrome/ Firefox
- Visual Paradigm for UML - 8.0 Enterprise Edition

4.2 Development Tools

At the moment of developing the system, a variety of tools were used in order to improve the quality of the work and to speed up the work. Table 4.1 shown below provide an explanation on the used tools.

Tool Used	Description
Spring Tool Suit	Provide a customized all-in-one Eclipse based distribution that makes application development easy. The tool suites provide ready-to-use combinations of language support, framework support, and runtime support, and combine them with the existing Java, Web and Java EE tooling from Eclipse [11].
Database PostgreSQL Server	<p>PostgreSQL is an ORDBMS based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department.</p> <p>PostgreSQL is an open-source descendant of this original Berkeley code. It supports a large part of the SQL standard and offers many modern features like:</p> <ul style="list-style-type: none"> • complex queries • foreign keys • triggers • updatable views • transactional integrity • multi-version concurrency control
Apache Tomcat Application Server	The Apache Tomcat® software is an open source implementation of the Java Servlet, Java Server Pages, Java Expression Language and Java Web Socket technologies [12].
Nitro pro 8	Used for the testing purpose of Invoice printing and to refer the project guide during the development phase.

Visual Paradigm for UML - 8.0 Enterprise Edition and MS Visio	Designed diagrams referred at the development phase were designed using these tools.
--	--

Table 4.1: Descriptive Table on Tools set used for the Development

Figure 4.1 displays the tools set use at the system development and implementation.



Figure 4.1: Tools set used at system development

Other than these bootstrap and jquery plugins also were used for the system to run smoothly.

4.3 Code and Module Structure

This section of Chapter 4 discuss on the code and module structure of the developed system. Using the tools mentioned at the above section development phase was completed.

Shown below by Figure 4.2 is the, which displays an overview of the package structure.

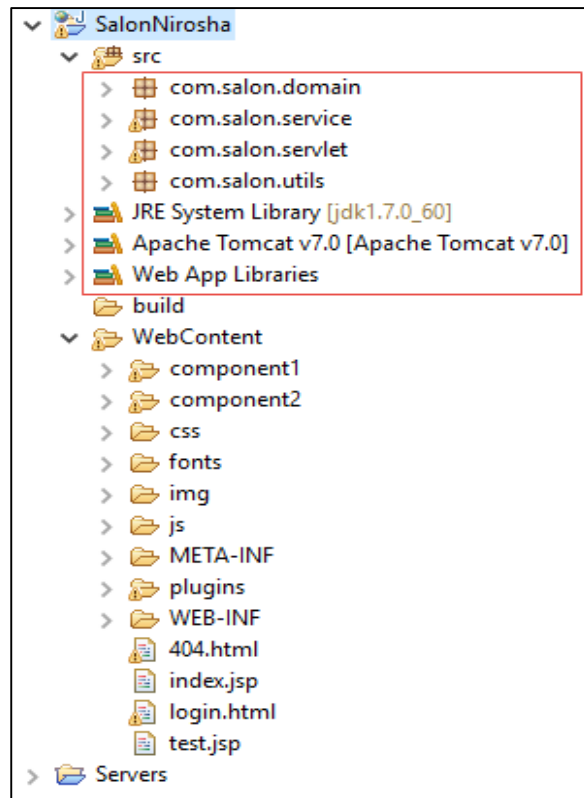


Figure 4.2: Package structure of the System

Domain Classes:

This java package class include the domains which represents the tables at the database.

Services:

This java package contains all the DAO (Data Access Objects) classes of respective interfaces along with the abstarct methods of the database.

Servlets:

An HTTP servlet is a special type of servlet that handles an HTTP request and provides an HTTP response, usually in the form of an HTML page.

Utility Classes:

Utility classes provide common functionalities which can be used at everywhere inside the project code.

JSP:

JSP technology is used to create web applications. It consists of HTML tags and JSP tags inside a JSP page.

Other than the aforementioned modules, JRE system Library and Web App Library has been used at the system as display at the Figure 4.2.

Figure 4.3 displays the Util class developed for the database connection which should be used always when connecting to the database to perform CRUD operations.

```

package com.salon.utils;

import java.sql.Connection;

public class JDBCUtil{
    private static final String DRIVER = "org.postgresql.Driver" ;
    private static final String URL = "jdbc:postgresql://localhost:5432/sms";
    private static final String USERNAME = "postgres";
    private static final String PASSWORD = "123" ;

    public static Connection getConnection() {

        System.out.println("PostgreSQL connectivity test start!");
        try {
            Class.forName(DRIVER);
        } catch (ClassNotFoundException e) {
            System.out.println("No JDBC Driver");
            System.out.println(e.getMessage());
            //return;
        }
        System.out.println("Driver registered successfully!!!");

        Connection con = null ;
        try{
            con = DriverManager.getConnection(URL , USERNAME , PASSWORD);
        } catch (SQLException e){
            System.out.println("Connection failed");
            e.printStackTrace();
            //return;
        }
        if (con != null) {
            System.out.println("Successfully Connected!!!");
        } else{
            System.out.println("Failed");
        }
        return con;
    }
}

```

Figure 4.3: Database Connection Code

Figure 4.4 display the Date Formatter util class developed for the usage of formatting the date captured at most of the screens of the system.

```

public static Date StringtoDate(String d) {

    SimpleDateFormat formatter=new SimpleDateFormat("yyyy-MM-dd");
    Date date=null;
    try {
        date = formatter.parse(d);
    } catch (Exception e) {
        e.getMessage();
    }
    return date;
}

public static Timestamp DateToTimestamp(Date date) {
    Timestamp time=new Timestamp(date.getTime());
    return time;
}

```

Figure 4.4: Date Formatter Code

Figure 4.5 is the sample code on the web.xml; which defines the servlet and its connection with the application layer.

```

<display-name>SalonNirosha</display-name>

<welcome-file-list>
    <welcome-file>login.html</welcome-file>
</welcome-file-list>

<servlet>
    <servlet-name>ServletSignIn</servlet-name>
    <servlet-class>com.salon.servlet.ServletSignIn</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>ServletSignIn</servlet-name>
    <url-pattern>/ServletSignIn</url-pattern>
</servlet-mapping>

```

Figure 4.5: Sample web.xml Code

4.4 Codes Reused

When developing the system, some codes were reused at several locations. This section devotes some of them along with their samples.

The code displays at Figure 4.6 Reused at both Staff Registration and client Registration forms; to get the value of the 'Town' for the selected 'Postal Code'.

```
function setTown() {
    var e = document.getElementById("postalCode");
    town = e.options[e.selectedIndex].text;
    document.getElementById("town").value = town.split("-")[1];
}
```

Figure 4.6: Code use to retrieve the Town for the selected Postal Code

The code displays at Figure 4.7 Reused at both Staff Registration and client Registration forms; to carry the service ids to the staff-service mapping table at the database.

```
<script type="text/javascript">
    //get the service ids to an array
    var serviceIds = new Array();
    function saveServices(id) {
        var option = true;
        var len = serviceIds.length;
        for (var i = 0; i < len; i++) {
            if(serviceIds[i]==id) {
                var index = serviceIds.indexOf(id);
                serviceIds.splice(index,1);
                option = false;
                break;
            }
        }
        if(option){
            serviceIds.push(id);
        }
        document.getElementById("serviceIdsHidden").value = serviceIds.join("@");
    }
}
```

Figure 4.7: Code use to carry Service Ids to Staff-Service Mapping Table

The code displays at Figure 4.8 Reused at both Staff Registration and client Registration forms; to get the value for the 'NIC Number' field for the entered 'Birth Date'.

```
function setDOBviaNic1() {
    var nic=document.getElementById("nicNumber").value;
    var nicYear="19".concat(nic.substring(0,2));
    var nicDays=nic.substring(2,5);
    var month="";
    var feb;
    if(nicDays>=501 && nicDays<= 866){
        nicDays=nicDays-500;
    }
    if ((nicYear % 4) == 0){
        feb = 31 + 29;
    } else{
        feb = 31 + 28;
        if (nicDays>feb){
            nicDays=nicDays-1;
        }
    }
    var dayId;
    nicDays=Number(nicDays);
    if (nicDays <= 31){
        month="01 ";
    }else if (nicDays <= feb){
        month="02";
        nicDays =Number(nicDays) - 31;
    }else if (nicDays <= (feb + 31)){
        month="03";
        nicDays = nicDays - feb;
    }else if (nicDays <= (feb + 61)){
        month="04";
        nicDays = nicDays - (feb + 31);
    }else if (nicDays <= (feb + 92)){
        month="05";
        nicDays = nicDays - (feb + 61);
    }else if (nicDays <= (feb + 122)){
        month="06";
        nicDays = nicDays - (feb + 92);
    }else if (nicDays <= (feb + 153)){
        month="07";
        nicDays = nicDays - (feb + 122);
    }else if (nicDays <= (feb + 184)){
        month="08";
        nicDays = nicDays - (feb + 153);
    }else if (nicDays <= (feb + 214)){
        month="09";
        nicDays = nicDays - (feb + 184);
    }else if (nicDays <= (feb + 245)) {
        month="10";
        nicDays = nicDays - (feb + 214);
    }else if (nicDays <= (feb + 275)){
        month="11";
        nicDays = nicDays - (feb + 245);
    }else if (nicDays <= (feb + 306)){
        month="12";
        nicDays = nicDays - (feb + 275);
    }

    if(nicDays.length==2){
        nicDays="0"+nicDays;
    }
    document.getElementById("dateOfBirth").value = nicYear+"-"+month+"-"+nicDays ;
}
```

Figure 4.8: Code use to retrieve the NIC Number using the Date of Birth

4.5 Interaction between the System Modules

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts [13]:

Model:

The lowest level of the pattern which is responsible for maintaining data. Objects of the model retrieve and store model state in a database.

E.g.: - DAO was used as the 'Model' at the development phase.

View:

This is responsible for displaying all or a portion of the data to the user while enabling them to modify the data.

E.g.: - JSP was used as the 'View' at the development phase.

Controller:

Software Code that controls the interactions between the Model and View. The controller renders the appropriate view with the model data as a response to the URL request.

E.g.: - Servlet was the 'controller' decided to use.

The diagram shown by Figure 4.9 simply explains the MVC architecture.

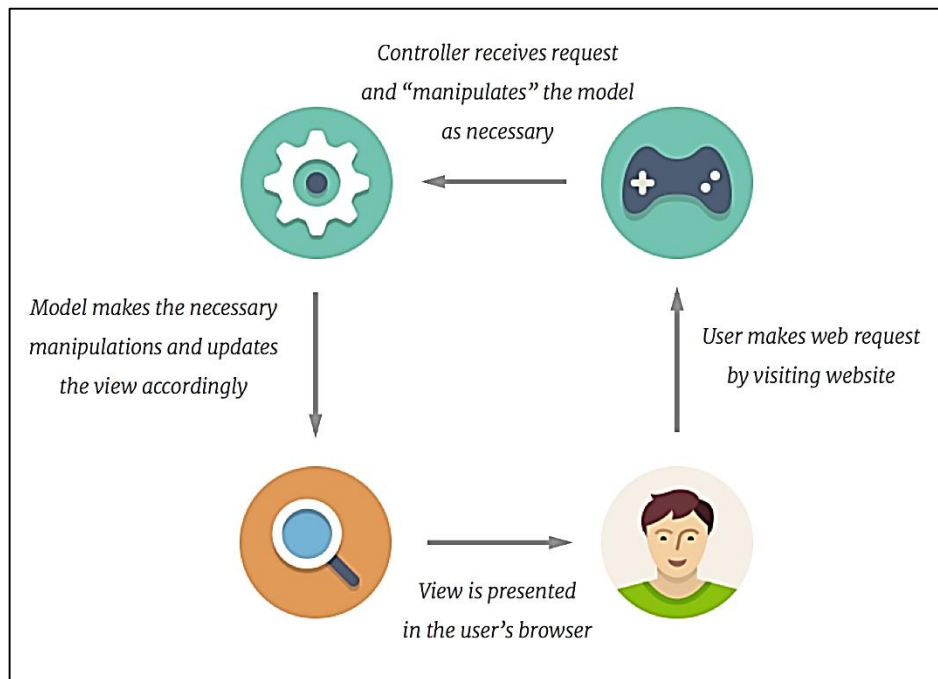


Figure 4.9: Diagram on MVC Architecture

Chapter 5 - Evaluation

Evaluation chapter is based on the system testing plan and its results gained after the system implementation.

Unit Testing:

Testing of individual software modules or components that make up an application or system. These tests are usually done by the developers of the module and in a test-driven-development methodology [14].

Unit Testing was helped to find out errors and weaknesses in each individual software module.

Integration Testing:

Individual module which were subjected to unit testing were tested by integrating with each other. A top down approach was followed. Errors that can be occur at the integration were able to find at the end of this test.

System Testing:

The entire system was tested by interfacing hardware and software components of the entire system as to find out errors with the whole system.

User Acceptance Testing:

All the system users at the salon premise were helped for the acceptance testing, to make it a success. Hardware and software at the salon were used as to test the real status of the system.

Along with these functional testing methods, it was able to provide an error free system for the client, to prevent the obstacle which could occur latterly.

5.1 Test Plan

Planning for test is a very important aspect for a developed system as well as for a system in progress level. The overall functionality of the system should be checked via test plan. Test cases were written for each module to be tested before testing the system. The implemented system was tested using different test cases.

Following Table 5.1 is on the test case which contains all the details on the test plan for the whole system.

Test No.	Module Name	Test Description	Priority	Expected Result
1	System Login	Correct Username and incorrect Password.	HIGH	“Incorrect Password”
2		Correct Password and incorrect Username.	HIGH	“Incorrect Username”
3		Incorrect Username and incorrect Password.	HIGH	“Incorrect Username and Password”
4		Blank Username.	HIGH	“Username Required”
5		Blank Password.	HIGH	“Password Required”
6		Correct both Username and Password.	HIGH	“Login Success”
7	Staff Registration	Validate Staff Information adding.	HIGH	“Staff Registered Successfully”
8		If all the fields empty.	HIGH	“Mandatory fields Required”
9		Blank Employee Type	HIGH	“Employee Type Required”
10		Blank Employee From Date	HIGH	“Employee From Date Required”
11		Blank Initials	MEDIUM	“Initials Required”
12		Blank Last name	MEDIUM	“Last name Required”
13		Blank Mobile Number	HIGH	“Mobile Number Required”
14		Blank Password	HIGH	“Password Required”
15		Blank Confirm Password	HIGH	“Confirm Password Required”

Salon Management System for ‘Salon Nirosha’

16	Staff Registration (Continue)	Blank NIC Number	HIGH	“NIC Required”
17		If Confirm Password differ from Password	HIGH	“Both should be the Same”
18		Incorrect NIC Number Format	HIGH	“Incorrect NIC Number”
19		Incorrect Mobile Number Format	HIGH	“Incorrect Mobile Number”
20		Incorrect Land Number Format	MEDIUM	“Incorrect Land Number”
21		Incorrect Fax Number Format	MEDIUM	“Incorrect Fax Number”
22	Client Registration	Validate Client Information adding.	HIGH	“Client Registered Successfully”
23		If all the fields empty.	HIGH	“Mandatory fields Required”
24		Blank Client From Date	HIGH	“Client From Date Required”
25		Blank Initials	MEDIUM	“Initials Required”
26		Blank Last name	MEDIUM	“Last name Required”
27		Blank Mobile Number	HIGH	“Mobile Number Required”
28		Blank NIC Number	HIGH	“NIC Required”
29		Incorrect NIC Number Format	HIGH	“Incorrect NIC Number”
30		Incorrect Mobile Number Format	HIGH	“Incorrect Mobile Number”
31		Incorrect Land Number Format	MEDIUM	“Incorrect Land Number”
32		Incorrect Fax Number Format	MEDIUM	“Incorrect Fax Number”
33	Staff Profile	Validate Update Profile Details	HIGH	“Staff Profile Updated Successfully”

34	Client Profile	Validate Update Profile Details	HIGH	“Client Profile Updated Successfully”
35	Appointment Creation	Validate Appointment Creation.	HIGH	“Appointment Created Successfully”
36		If all the fields empty.	HIGH	“Mandatory fields Required filling”
37		Staff Not Selected.	MEDIUM	“Staff Required”
38		Blank Date.	MEDIUM	“Date Required”
39		Blank Time.	MEDIUM	“Time Required”
40		Client Not Selected.	MEDIUM	“Client Required”
41		Appointment Reschedule	Validate Appointment Reschedule.	HIGH
42	Appointment Cancellation	Validate Appointment Cancellation.	HIGH	“Appointment Cancelled Successfully”
43	Appointment Status Handling	Status Changed as Started	MEDIUM	“Appointment Started Successfully”
44		Status Changed as No Show	MEDIUM	“Appointment No Show”
45		Status Changed as Complete	MEDIUM	“Appointment Cancelled”
46	Payment Handling	Validate Payment	HIGH	“Payment Done Successfully”
47		Blank Paid Amount	HIGH	“Paid Amount Required”

Table 5.1: Test Case for the Entire System

5.2 User Evaluation

The following feedback or the evaluation form at Figure 5.1 was used to gain the feedback from the client.

User Evaluation Form for Salon Management System for 'Salon Nirosha':

Employee Name :

Designation :

Use the following ratings to mark the results:

Excellent	Very Good	Good	Average	Poor
5	4	3	2	1

#	Assessment Description	Results
1	System Performance	
2	Usability	
3	Security	
4	User Interface	
5	System Quality	

Are you satisfied with the overall system, and are the system features accomplish your business requirement?

.....

.....

Any Other Suggestions/ Comments?

.....

.....

Employee Signature :

Date :

Figure 5.1: User Evaluation Form

Three evaluation forms were distributed among three employees at the salon premise along with the salon owner; after implementing the system at the client site.

Selected employees were under the three user types maintained by the system. Main purpose of this was to get an overall feedback on each area module of the system. Therefore System administrator commented on her experience, System Officer fed back on her experience and the cashier on hers.

5.3 Summary Chart Review

Based on the feedback forms collected, a chart was designed to properly give an idea on the clients’ feedback over the salon management system. The chart at Figure 5.2 is based on the form results attached with Appendix E of this dissertation.



Figure 5.2: Chart on User Evaluation Results

Chapter 6 – Conclusion

Salon Management is still an emerging business area in Sri Lanka; though it is widely spread in the foreign developed countries. When turned into the local market, visits hair and beauty salons and spas in need of fulfilling the beauty needs has become a trend among the people around the country. Therefore providing a better service for the customers at their first and foremost arrival to the salon premise, will aid to make a good impression on their mind over the salon. It can be considered as a fine business trick and most of the Salon Owners follows these artifices.

However at rush hours, it is very difficult to satisfy the customers, especially if the particular salon is fully depend on a manual process; and the staff also become fed up when they were burdened with disgusting tasks. Un-satisfaction of both customers and employees may pave the way for the decline of the specific salons’ businesses. Therefore the concept which turned into a salon management software at the end of this project will be a great assistance for the salon owners to speed up their work by eliminating the unwanted barriers by improving their quality of business.

The project started from requirement gathering phase was successfully completed after passing the user acceptance test next to the implementation phase. However it is not practical to fulfill all the clients’ requirements for the maximum. But the system was up to a satisfactory level and client accepted it with a gratitude.

6.1 Lessons Learnt

By doing the final project, lot of lessons were able to learn at each every phase of the SDLC. Practical understanding on software engineering disciplines were able to capture. Following is the lessons list learned during the project cycle:

- Able to acquire knowledge on a different business area.
- Able to gain a practical training on requirement gathering.
- Able to enhance interpersonal skills, at frequent client meetups.
- Able to learn on new technological tools and frameworks.
- Able to improve documenting skills.
- Able to learn on facing for challenges.

6.2 Problems Encountered

As well as the lessons, it was able to meet up with problems during the project time such as:

- Had to work within a fixed time period by planning the work.
- Had to do tasks like analyzing and designing which were difficult to grab within a limited period.
- Had to learn on Java which is an advance programming language while coding.

Anyway it was happy to say that all the problems encountered were able to solve and move forward to make this project a success one.

6.3 Future Developments

The software handover is the first product used by the client. Hence she may take her time to tell the change requests that she needs to do for the salon management system. But some suggestions were there to enhance the performance of the system as follows:

Reminder generating facility through SMS:

Mobile is an obligatory technological equipment every one keep on their hand at anytime, anywhere. Therefore if the system facilitate an SMS other than an email, it may ease the clients as well as the employees; because refer to an email also take a considerable time.

Providing online appointment creation facility:

Clients at far places will be able to make appointments by themselves if this facility is provided via the system as a future facility.

Giving staff rating facility:

Regular clients may become friendly with their favorite salon staff members (stylists) by the time passes on. Therefore if providing a rating feature by the system; then the salon owner will be able to know the most attractive staff members works at the salon. Therefore she can increase allowances of those employees.

References

- [1] "English Oxford Living Dictionaries," Wikipedia, 2017. [Online]. Available: https://en.oxforddictionaries.com/definition/beauty_salon. [Accessed July 2017].
- [2] "www.tutorialspoint.com," [Online]. Available: https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm. [Accessed 25 04 2017].
- [3] U. Eriksson, "ReQtest," 2017. [Online]. Available: <http://reqtest.com/requirements-blog/what-are-non-functional-requirements/>. [Accessed August 2017].
- [4] "www.tutorialspoint.com," 2017. [Online]. Available: https://www.tutorialspoint.com/software_testing_dictionary/functional_requirements.htm. [Accessed 27 05 2017].
- [5] Martinig & Associates, "www.methodsandtools.com," 1995-2017. [Online]. Available: <http://www.methodsandtools.com/archive/archive.php?id=113>. [Accessed 28 05 2017].
- [6] "Tutorialspoint," 2017. [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm. [Accessed 2 August 2017].
- [7] "Techopedia," 2017. [Online]. Available: <https://www.techopedia.com/definition/29998/system-design>. [Accessed 5 August 2017].
- [8] "/www.reference.com," IAC Publishing, LLC, 2017. [Online]. Available: <https://www.reference.com/technology/advantages-standalone-computer-c30ab51db5ce2560>. [Accessed 03 06 2017].
- [9] "www.tutorialspoint.com," Tutorialspoint, 2017. [Online]. Available: https://www.tutorialspoint.com/uml/uml_class_diagram.htm. [Accessed 05 06 2017].

2017].

- [10] usability.gov, "www.usability.gov," 2017. [Online]. Available: <https://www.usability.gov/what-and-why/user-interface-design.html>. [Accessed 17 6 2017].
- [11] "www.spring.io," Pivotal Software, 2017. [Online]. Available: <https://spring.io/tools>. [Accessed 09 2017].
- [12] "tomcat.apache.org," The Apache Software Foundation , 1999-2017. [Online]. Available: <http://tomcat.apache.org/>. [Accessed 2017].
- [13] "www.tutorialspoint.com," Tutorials Point, 2017. [Online]. Available: https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm. [Accessed 21 8 2017].
- [14] "http://www.qaquestions.org," 2017. [Online]. Available: <http://www.qaquestions.org/manual-testing/software-testing-methodologies-and-types-of-testing/>. [Accessed 2017].
- [15] www.springsource.com, "www.springsource.com," 2016. [Online]. Available: http://dist.springsource.com/release/STS/doc/STS-installation_instructions.pdf. [Accessed 20117].
- [16] "www.postgresqltutorial.com," PostgreSQL Tutorial, 2017. [Online]. Available: <http://www.postgresqltutorial.com/install-postgresql/>. [Accessed 2017].

Appendix A – System Documentation

According to the chapter 4 in this dissertation, implementation environment should arrange before setup the system for its usage. Therefore,

- Java Development Kit
- Apache Tomcat Server
- PostgreSQL Server should install and execute at the implementation environment accordingly.

Installing Java on Windows Operating Systems

1. Download java from Java download page www.java.com.
2. Find the location where downloaded file has been saved and double click to setup java as display in Figure A.1.

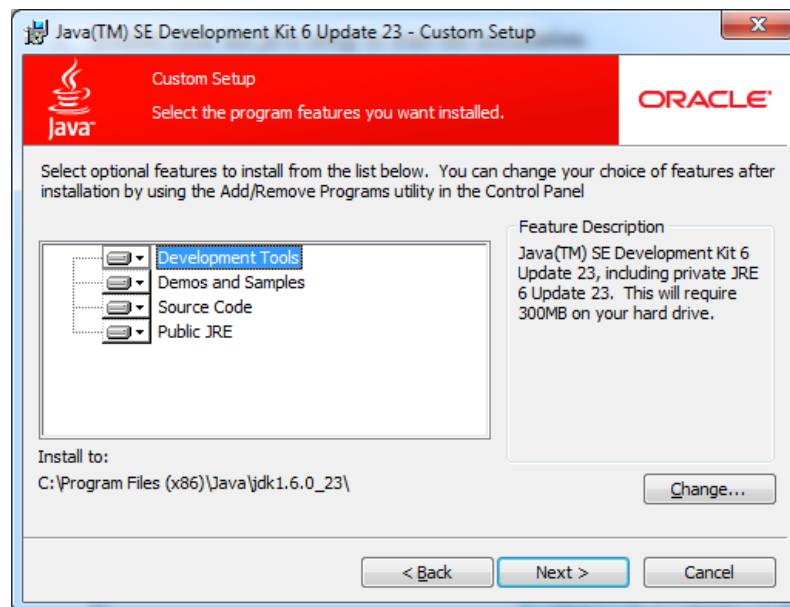


Figure A.1: JDK

3. Then click on the next button at the above screen.
4. Select the location to be installed.
5. Then the installation for JRE will prompt automatically.
6. Select the install location of it as well and click on the ‘Finished’ button.
7. After the installation, configure the JDK path in the environment variable to run Java at any location of the computer.

- For that go to Control Panel System > Security System > Select 'Advance System Settings' using the Figure A.2 screen.

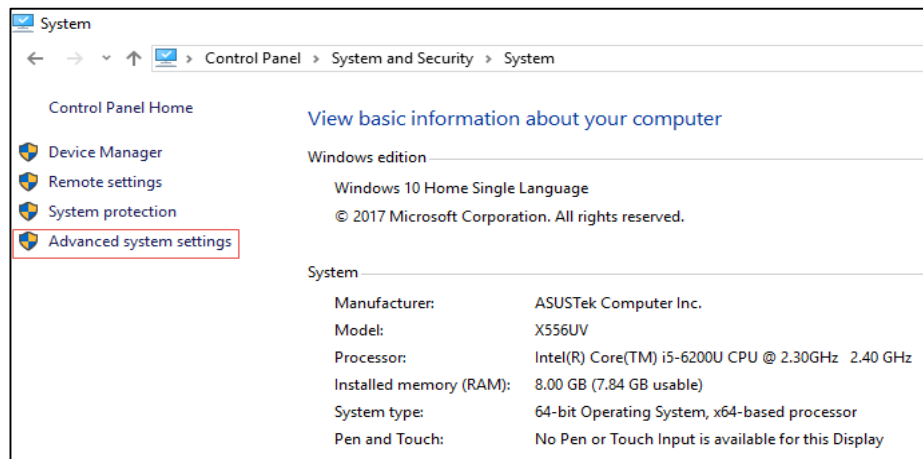


Figure A.2: Advance System Setup Screen

- Next go to the System Properties Screen shown by Figure A.3 and select the 'Environment Variable' button.

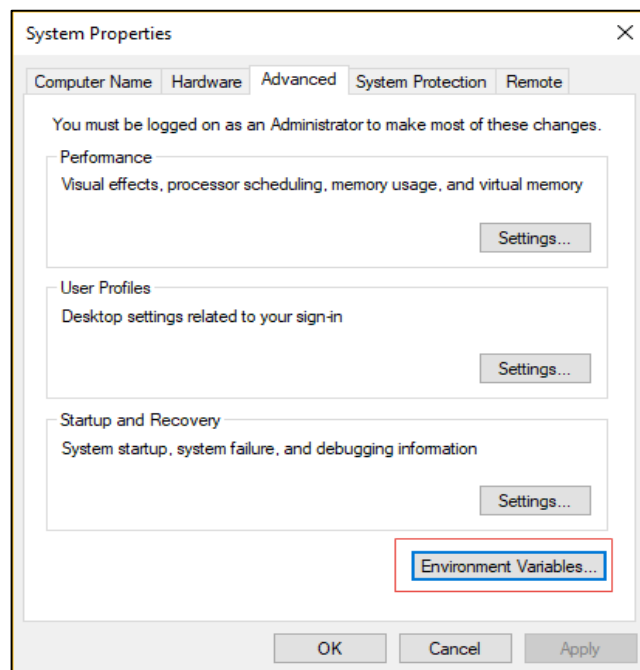


Figure A.3: System Properties screen

10. Create a new system variable named ‘JAVA_HOME’ shown at Figure A.4.

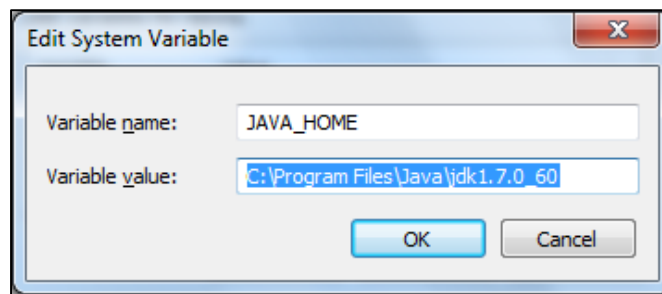


Figure A.4: JAVA_HOME Setup Screen

11. Finally find the relevant Path variable and add the path to the ‘JDK bin folder’ at the end of the variable value.

Installing STS on Windows Operating Systems

Download the STS executable installer (*.exe) from the STS download page. Once downloaded, please double-click the installer to launch the installation process. Once the STS Installer is launched you can finish the installation by going through the following steps:

1. Click “Next” on Welcome page.
2. Review and accept the license on the License Agreement page and click “Next”.
3. Select the installation path on the Target Path page and click “Next” (please don’t install into C:\Program Files, especially on Windows 7, or other multi user directories on Linux or Mac as this will cause issues on updates and extension installs).
4. Select that components that you want to install and click “Next” on the Select Installation Packages.
5. Select the path to your JDK installation (not on Mac OS X) and click “Next” on JDK Path page (this will validate if the selected JDK matches the STS version to install).
6. On the Installation page let the installation proceed and wait until all files are installed; click “Next”.
7. Create start menu shortcuts (on Windows only) and click “Next” on the Setup Shortcuts page.
8. Click “Finish” to end the installation. Check “Launch Spring Source Tool Suite” to launch STS on exit [15].

Installing PostgreSQL Server

1. Download PostgreSQL Installer from PostgreSQL Official website of <http://www.postgresql.org/download/windows/>.
2. Choose the latest version to download. It takes few minutes to complete the download.
3. Then double click on the installer file, an installation wizard will appear and guide you through multiple steps where you can choose different options that you would like to have in PostgreSQL.
4. Start Installing PostgreSQL by clicking the ‘Next button at Figure A.5.

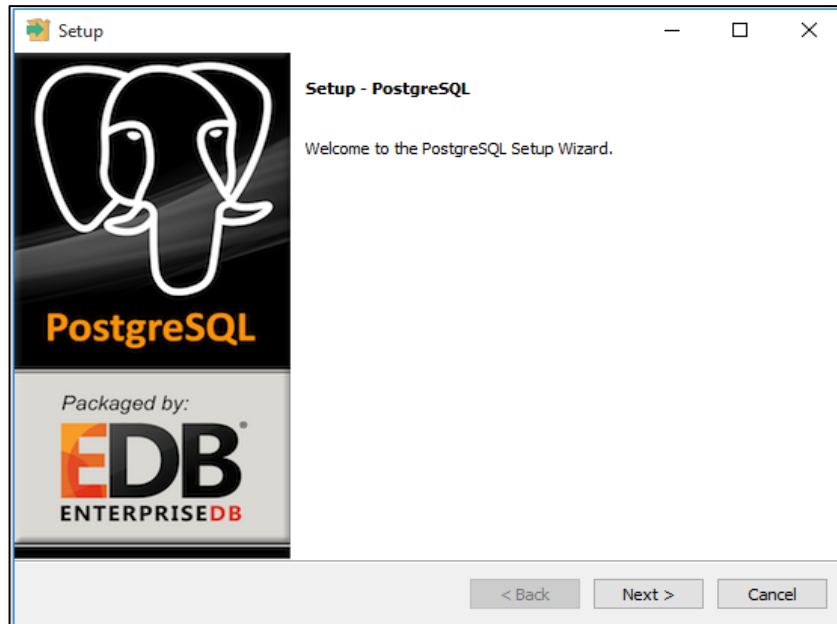


Figure A.5: PostgreSQL Installation Screen

5. Specify installation folder, choose your own or keep the default folder suggested by PostgreSQL installer.
6. Enter the password for the database super user and service account using the Figure A.6.

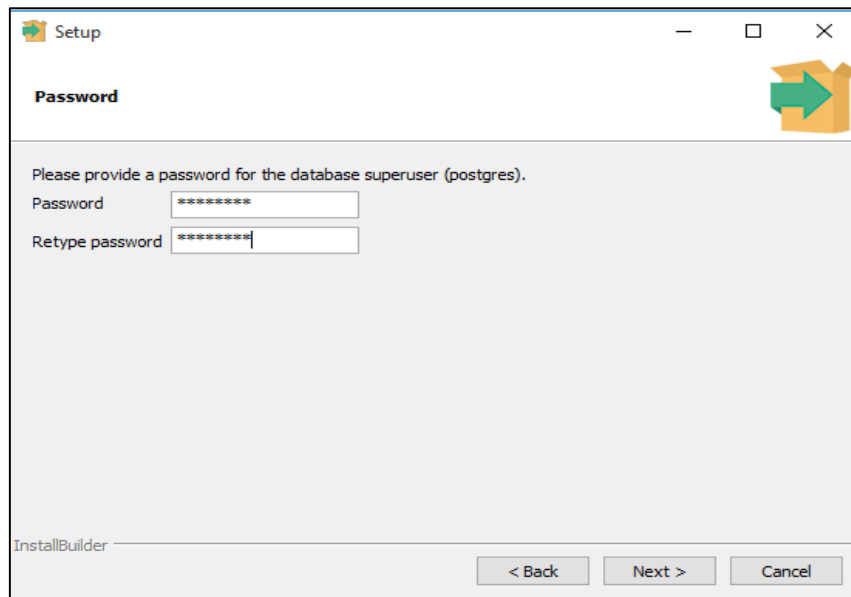


Figure A.6: Password entering Screen

7. Enter the port for PostgreSQL at screen shown by Figure A.7. Make sure that no other applications are using this port. Leave it as default if you are unsure.

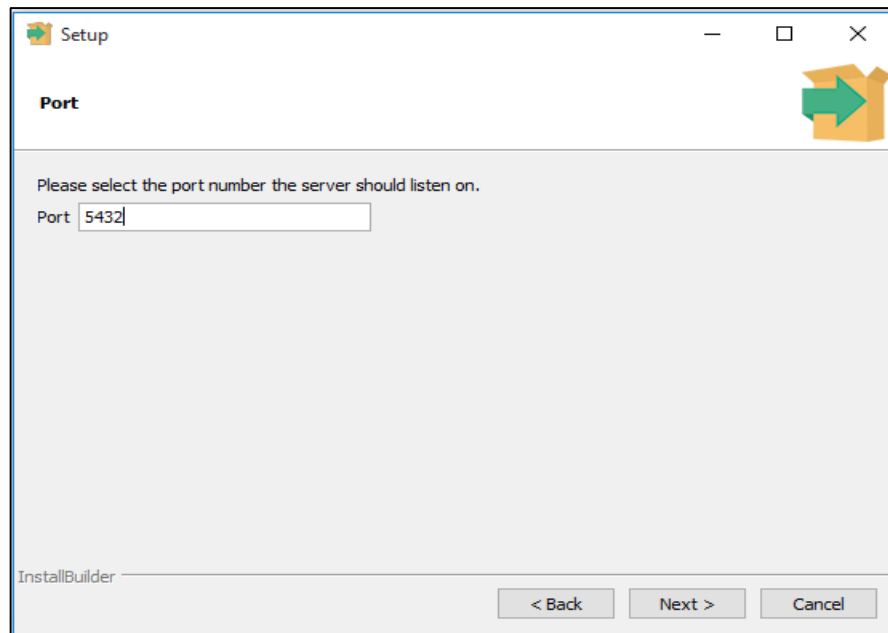


Figure A.7: Port entering Screen

8. Choose the default locale used by the database.
9. Click the Next button to install PostgreSQL as shown at Figure A.8.

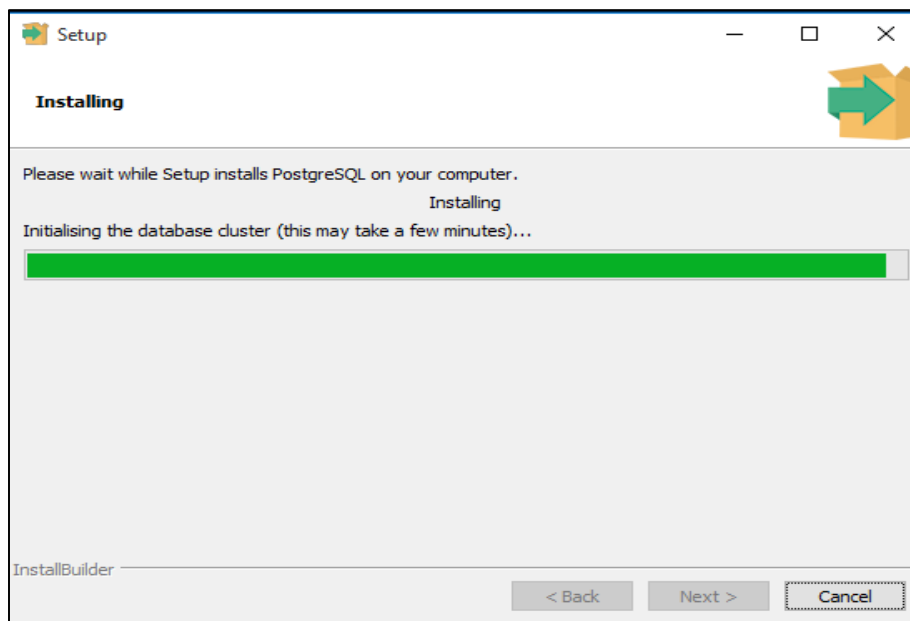


Figure A.8: Setup Installing Progress Screen

10. Next click the Finish button to complete the PostgreSQL installation.
11. Provide admin password once double clicked on the relevant port and provide the admin password used at the installation [16] as shown by Figure A.9.

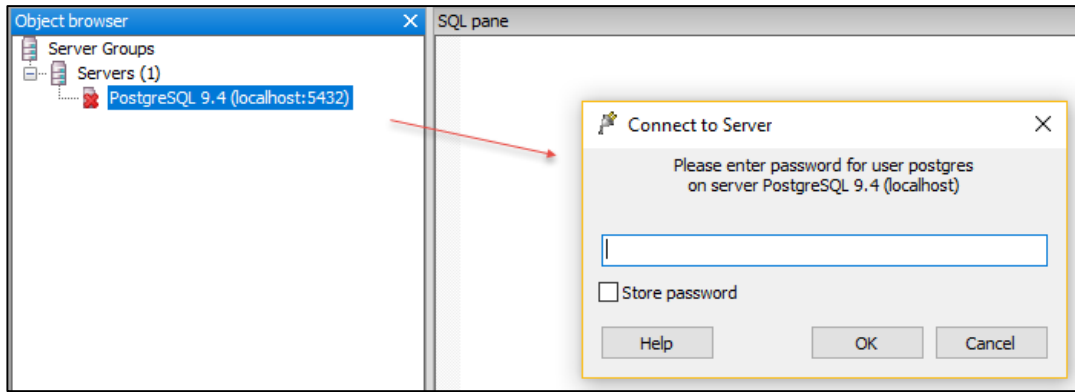


Figure A.9: Connect to the SQL Database Server

Appendix B – Design Documentation

All the relevant designed documents are displayed at this section. Figure B.1 at below displays the use case designed for the staff registration process.

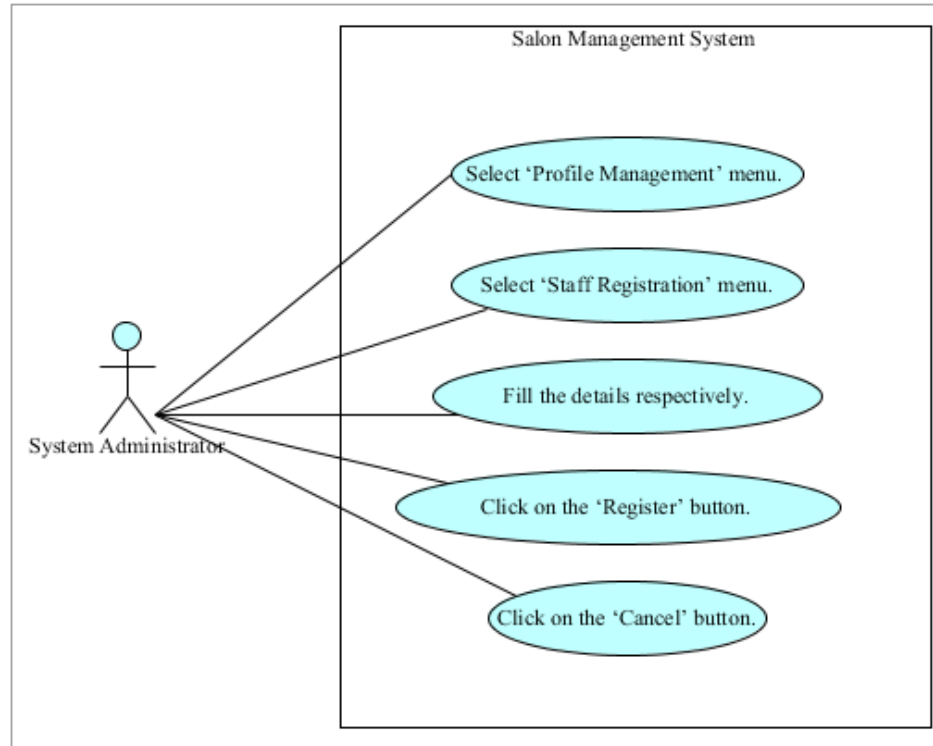


Figure B.1: Use Case for Staff Registration

Table B.1 display the use case description on Staff Registration.

Use Case:	Staff Registration
Actors:	System Administrator
Description:	Register the staff who worked at the salon.
Pre-Conditions:	
	System Administrator should login to the system.
Flow of Events:	
	<ol style="list-style-type: none"> 1. Select 'Profile Management' menu. 2. Select 'Staff Registration' menu. 3. Fill the details respectively. 4. Click on the 'Register' button; Or else click on the 'Cancel' button.

Table B.1: Use Case Description for Staff Registration

Following Figure B.2 displays the use case designed for the client registration process.

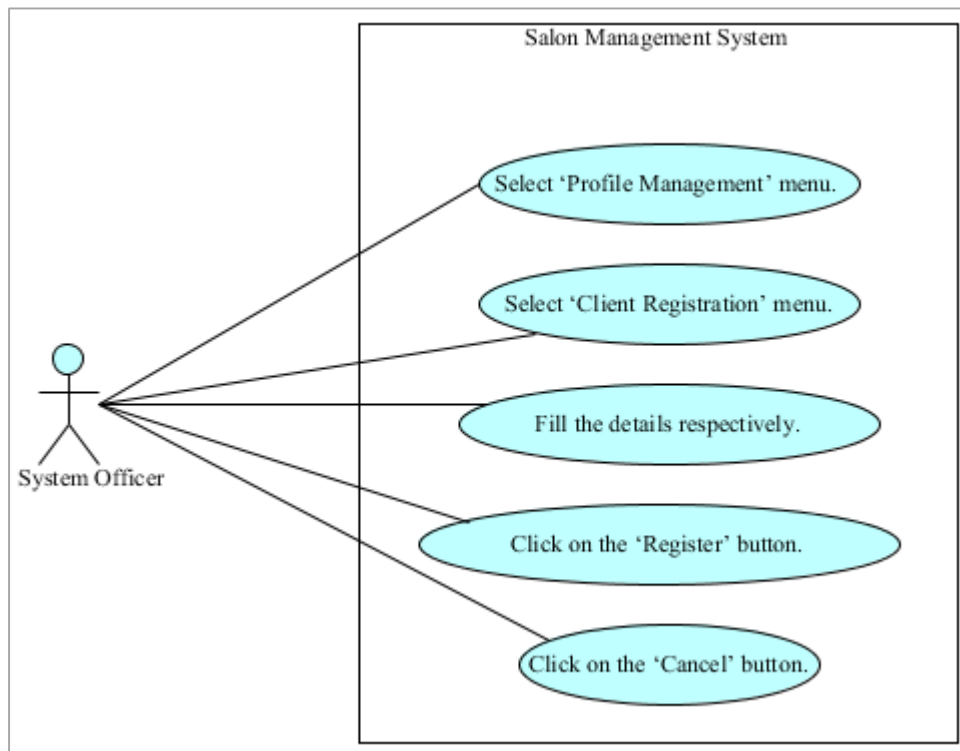


Figure B.2: Use Case for Client Registration

Table B.2 display the use case description on Client Registration.

Use Case:	Client Registration
Actors:	System Officer
Description:	Register the regular clients of the salon.
Pre-Conditions:	
System Officer should login to the system. System Officer should be able to access the Client Registration module.	
Flow of Events:	
<ol style="list-style-type: none"> 1. Select 'Profile Management' menu. 2. Select 'Client Registration' menu. 3. Fill the details respectively. 4. Click on the 'Register' button; Or else click on the 'Cancel' button. 	

Table B.2: Use Case Description for Client Registration

Following Figure B.3 displays the use case designed for the staff profile management process.

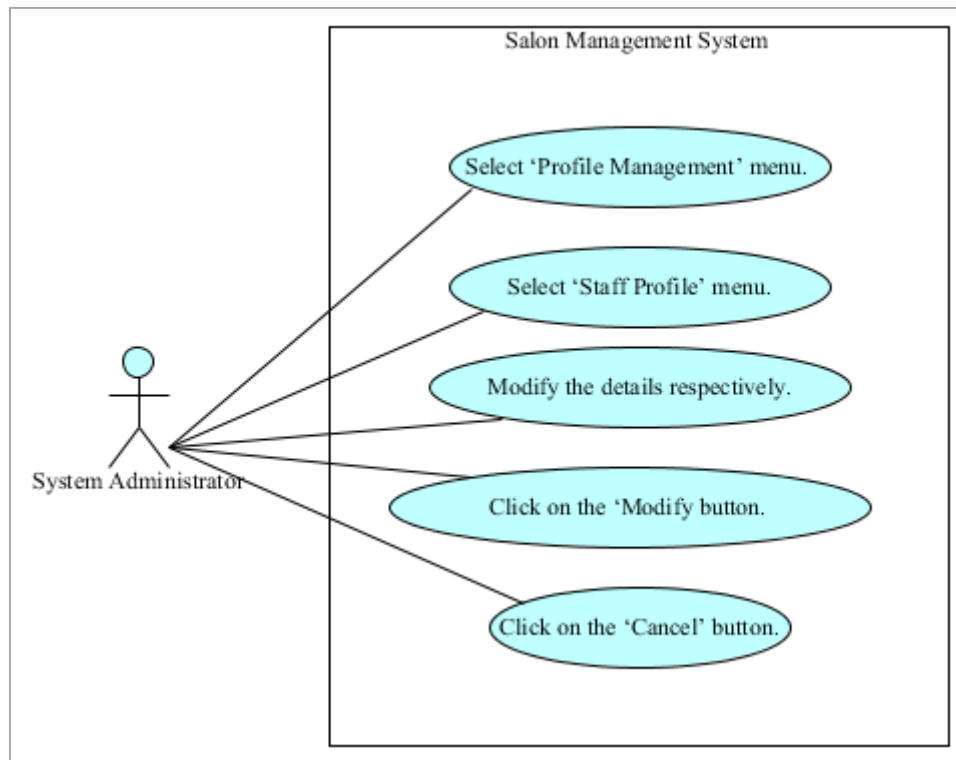


Figure B.3: Use Case for Staff Profile Management

Table B.3 display the use case description on Staff Profile Management.

Use Case:	Staff Profile Management
Actors:	System Administrator
Description:	View and modify the staff who worked at the salon. Inactive the staff members when needed.
Pre-Conditions:	
System Administrator should login to the system.	
Flow of Events:	
<ol style="list-style-type: none"> 1. Select 'Profile Management' menu. 2. Select 'Staff Profile' menu. 3. Modify the details respectively. 4. Click on the 'Modify button; Or else click on the 'Cancel' button. 	

Table B.3: Use Case Description for Staff Profile Management

Following Figure B.4 displays the use case designed for the client profile management process.

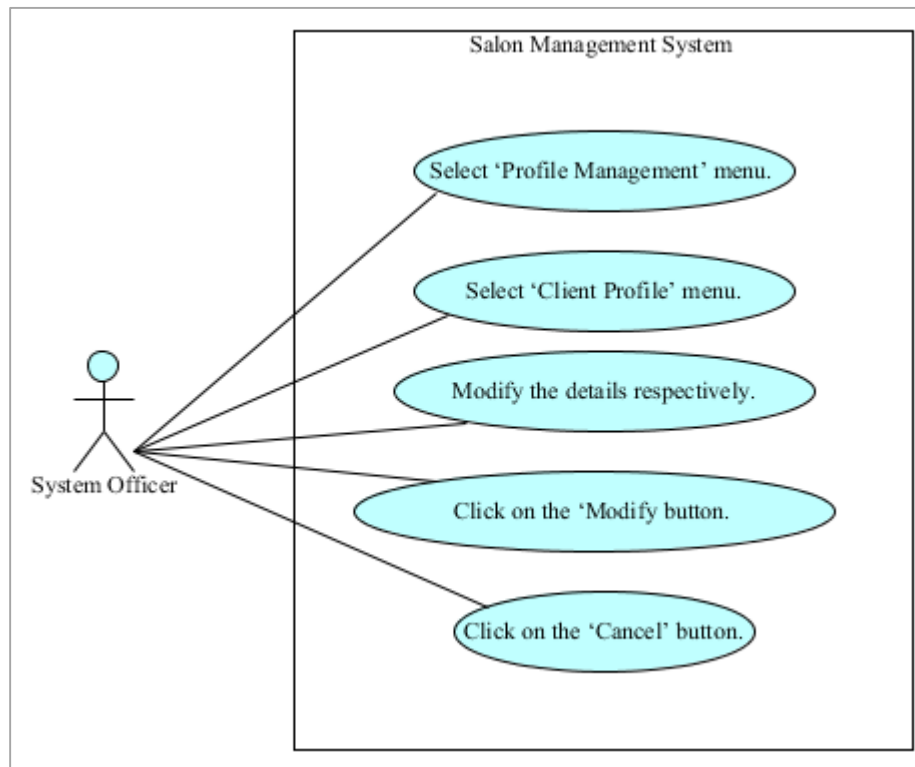


Figure B.4: Use Case for Client Profile Management

Table B.4 display the use case description on Staff Profile Management.

Use Case:	Client Profile Management
Actors:	System Officer
Description:	View and modify the staff who worked at the salon. Inactive the staff members when needed.
Pre-Conditions:	
System Officer should login to the system.	
Flow of Events:	
<ol style="list-style-type: none"> 1. Select ‘Profile Management’ menu. 2. Select ‘Client Profile’ menu. 3. Modify the details respectively. 4. Click on the ‘Modify button; Or else click on the ‘Cancel’ button. 	

Table B.4: Use Case Description for Client Profile Management

Following Figure B.5 displays the use case designed for the staff leave management process.

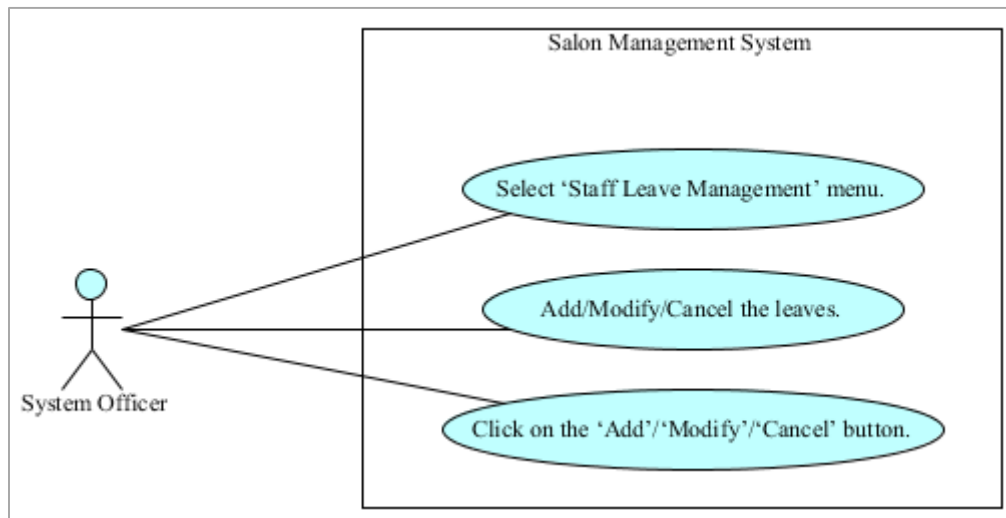


Figure B.5: Use Case for Staff Leave Management

Table B.5 display the use case description on Staff Leave Management.

Use Case:	Staff Leave Management
Actors:	System Administrator
Description:	Add, modify or cancel the leaves applied by staff members.
Pre-Conditions:	
	System Officer should login to the system.
Flow of Events:	
	<ol style="list-style-type: none"> 1. Select ‘Staff Leave Management’ menu. 2. Add/Modify/Cancel the leaves respectively. 3. Click on the ‘Add’/‘Modify’/‘Cancel’ button.

Table B.5: Use Case Description for Client Leave Management

Following Figure B.6 displays the use case designed for the appointment creation process.

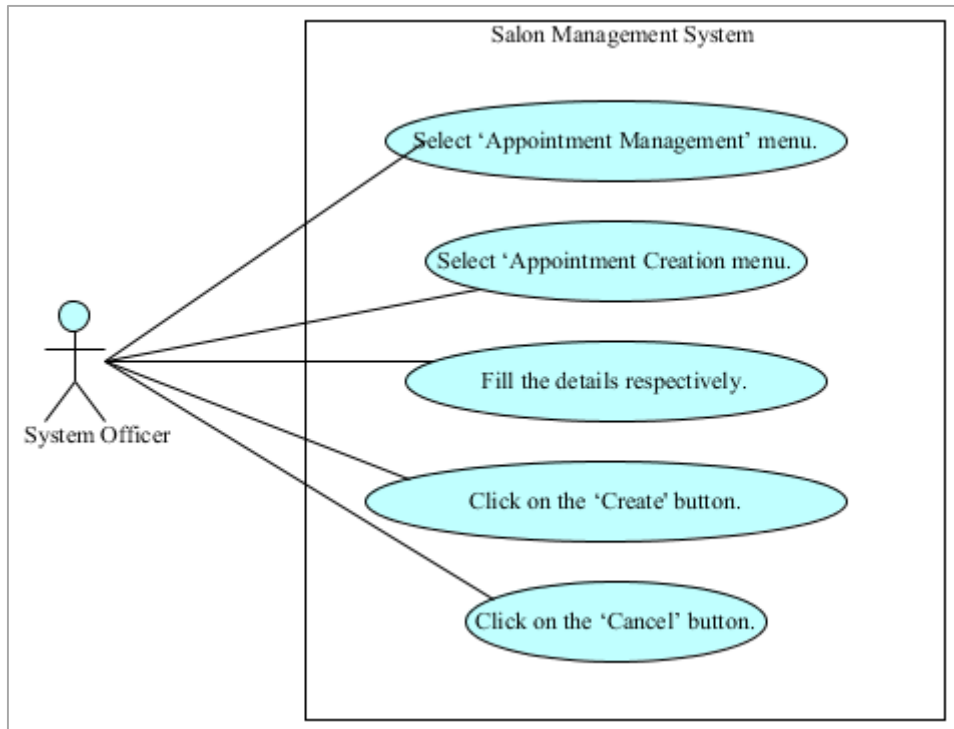


Figure B.6: Use Case for Appointment Creation

Table B.6 display the use case description on Appointment Creation.

Use Case:	Appointment Creation
Actors:	System Officer
Description:	Create Appointments.
Pre-Conditions:	
System Officer should login to the system.	
Flow of Events:	
<ol style="list-style-type: none"> 1. Select 'Appointment Management' menu. 2. Select 'Appointment Creation menu. 3. Fill the details respectively. 4. Click on the 'Create button; Or else click on the 'Clear button. 	

Table B.6: Use Case Description for Appointment Creation

Following Figure B.7 displays the use case designed for the appointment reschedule process.

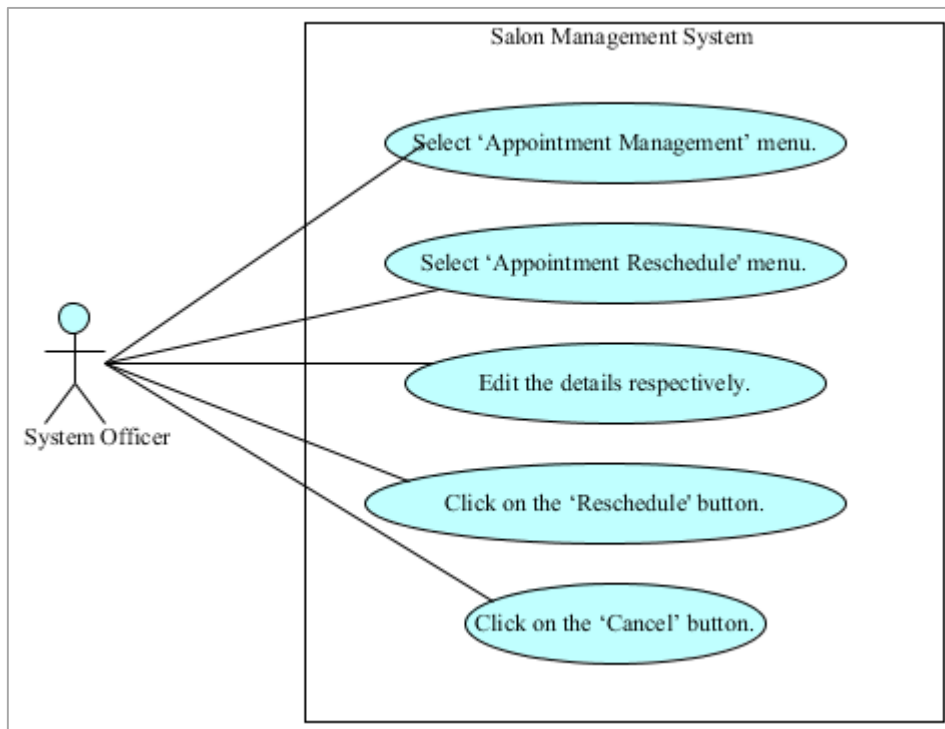


Figure B.7: Use Case for Appointment Reschedule

Table B.7 display the use case description on Appointment Reschedule.

Use Case:	Appointment Reschedule
Actors:	System Officer
Description:	Reschedule Appointments.
Pre-Conditions:	
System Officer should login to the system.	
Flow of Events:	
<ol style="list-style-type: none"> 1. Select 'Appointment Management' menu. 2. Select 'Appointment Reschedule' menu. 3. Edit the details respectively. 4. Click on the 'Reschedule' button; Or else click on the 'Cancel' button. 	

Table B.7: Use Case Description for Appointment Reschedule

Following Figure B.8 displays the use case designed for the appointment status management process.

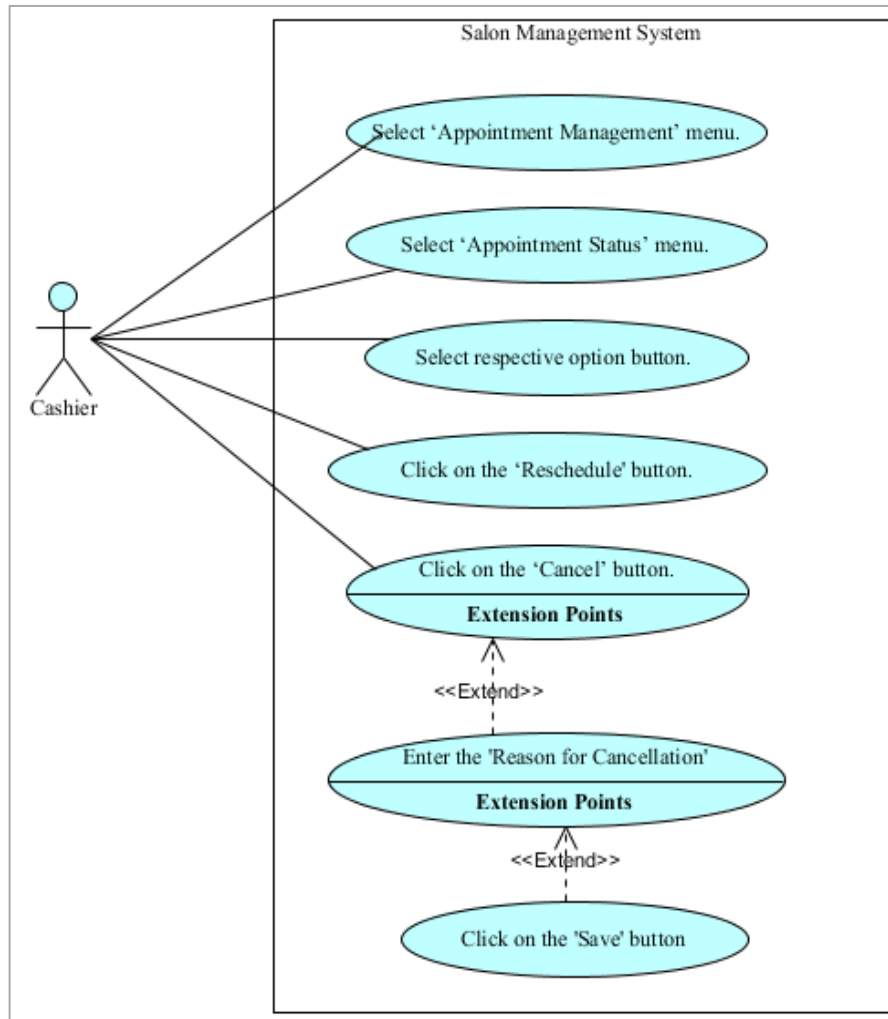


Figure B.8: Use Case for Appointment Status Management

Table B.8 display the use case description on Appointment Status Management.

Use Case:	Appointment Status Management.
Actors:	System Officer
Description:	Change Statuses of the created Appointments time to time and cancelling the appointments.
Pre-Conditions:	
System Officer should login to the system.	
Flow of Events:	
<ol style="list-style-type: none"> 1. Select 'Appointment Management' menu. 2. Select 'Appointment Status' menu. 3. Select respective option button. 4. If selected the 'cancel' button: <ol style="list-style-type: none"> a. Enter the reason for cancellation. b. Click on the 'Save' button. 	

Table B.8: Use Case Description for Appointment Status Management

Following Figure B.9 displays the use case designed for the payment application process.

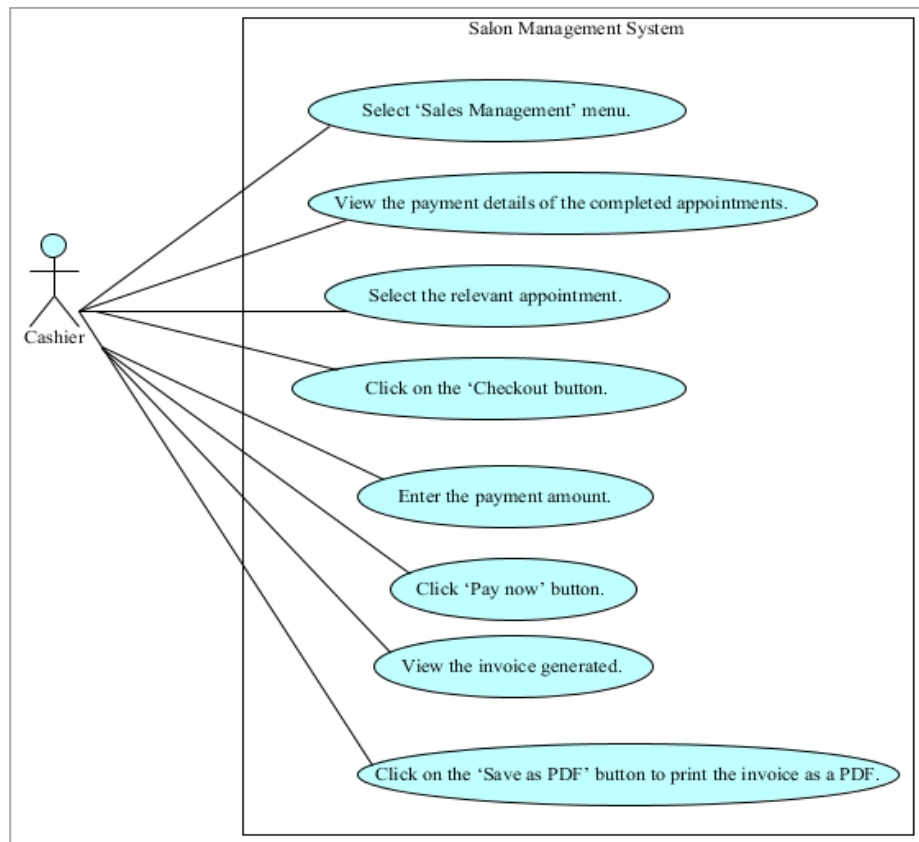


Figure B.9: Use Case for Payment Application Process

Table B.9 displays the use case description on the payment application process.

Use Case:	Apply Payment
Actors:	Cashier
Description:	Enter payment details and print the invoice.
Pre-Conditions:	
Cashier should login to the system.	
Appointment should be in the ‘Completed’ status.	
Flow of Events:	
Select ‘Sales Management’ menu.	
View the payment details of the completed appointments.	
Select the relevant appointment.	
Click on the ‘Checkout button.	
Enter the payment amount.	
Click ‘Pay now’ button.	
View the invoice generated.	
Click on the ‘Save as PDF’ button to print the invoice as a PDF.	
Click on the ‘Reschedule’ button; Or else click on the ‘Cancel button.	

Table B.9: Use Case Description for Payment Application

Other than the specified users for each use case, the System Administrator has been given the access privileges to create/ view/ update/ delete respective data at the total system.

Appendix C – User Documentation

User Document was done for ‘Salon Nirosha’ salon management system, as to aid the system users when using the system for the salon management purposes easily. As the employees who work at the salon are non-technical users, the user documentation may help them at trying the system till they become use to it.

1. Primarily user should log into the system using her own login privileges via the following screen at Figure C.1.

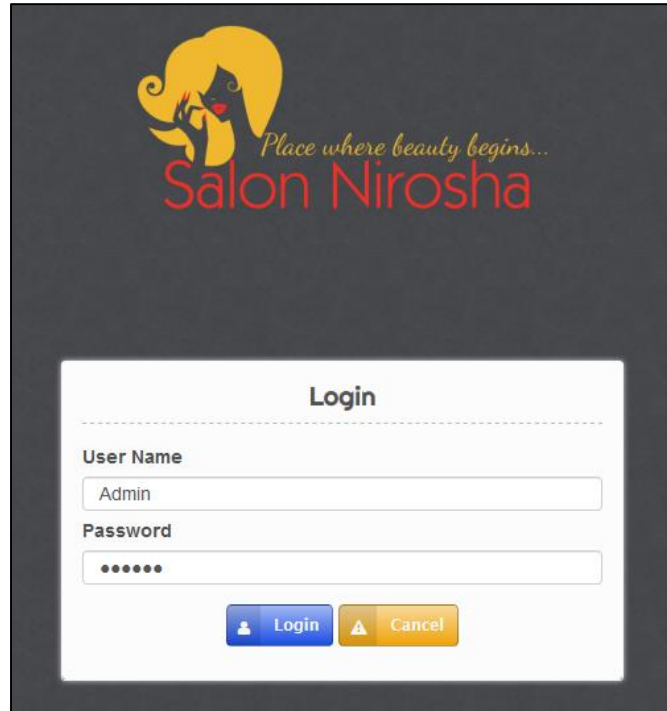


Figure C.1: System Login Screen

2. If the login credentials are correct, she will continue with her home page which illustrate at Figure C.2.

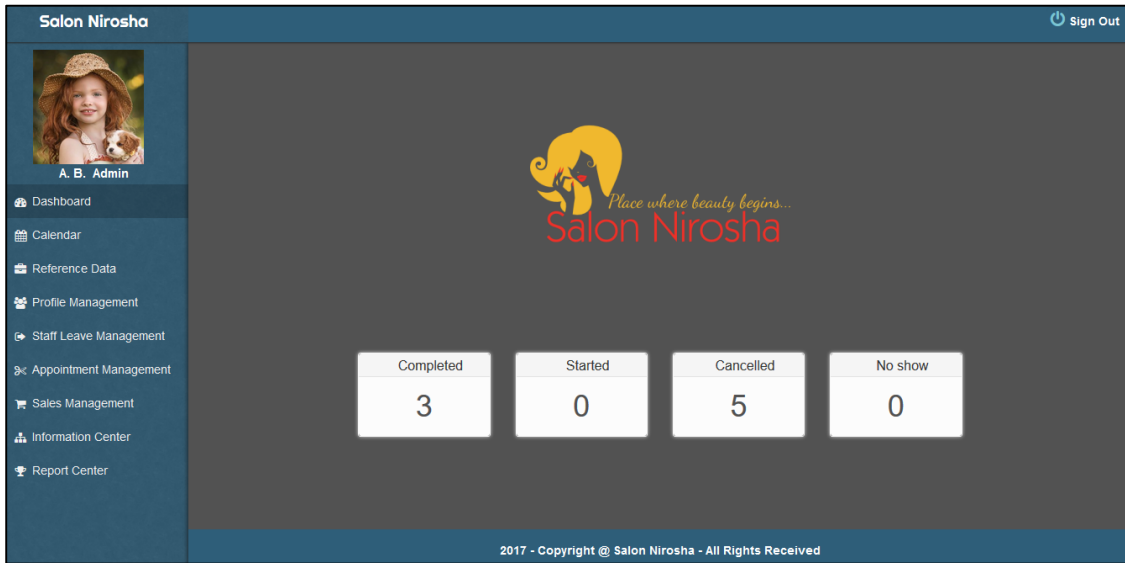


Figure C.2: System Home Page

3. System User should use the main menu given at Figure C.3 to navigate between screens/ modules. Modules display at the menu will changes with the access privileges.

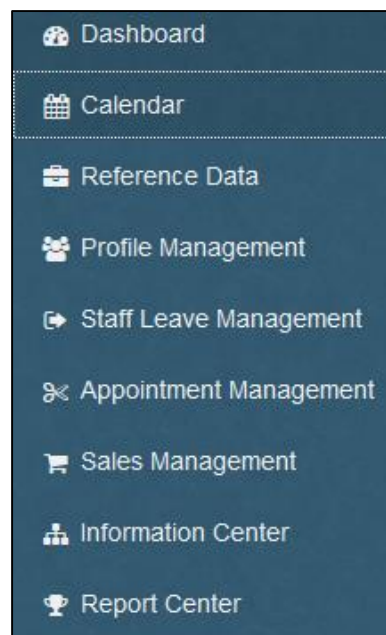


Figure C.3: System Main Menu

- Use the Client Registration screen at Figure C.4 given underneath to register regular clients who visits the salon most often.

Figure C.4: Client Registration Screen

- Appointment Creation screen at Figure C.5 is there for the user to create appointments.

Figure C.5: Appointment Creation Screen

6. Appointment Status can be handled by the aid of the screen at Figure C.6.

Figure C.6: Appointment Status Handling Screen

7. After completing the service respective to the appointment, select on the 'Complete' button at Figure C.6.
8. Then only the 'Check-out' button at the same screen will enable to be clicked.
9. Once clicked the 'Check-out' button following model at Figure C.7 will appear as to enter the payment done by the relevant client.

Figure C.7: Payment Application Module

10. View the invoice to be generate using the system and print the invoice display at Figure C.8.

Salon Nirosha 590, Galle Road, Horethuduwa, Morotuwa Sri Lanka Tel: +94 11 2648303		
# 35778-7363327 Tuesday, 8 Aug 2017 at 12:00am		
1	Wraps on Tuesday, 8 Aug 2017 from 10:30am to 11:40am (1h 10min)	Rs 2,000.00
1	Therapy on Tuesday, 8 Aug 2017 from 11:40am to 12:30pm (50min)	Rs 1,500.00
TOTAL DUE:		Rs 3,500.00
PAID BY CASH:		- Rs 3,500.00
CHANGE:		Rs 0.00

Figure C.8: Sample Invoice Printed after the Payment completion

11. Finally generate charts and reports using the 'Information Center and Report Center at the system. A sample chart on Appointment Statuses is shown at the Figure C.9.

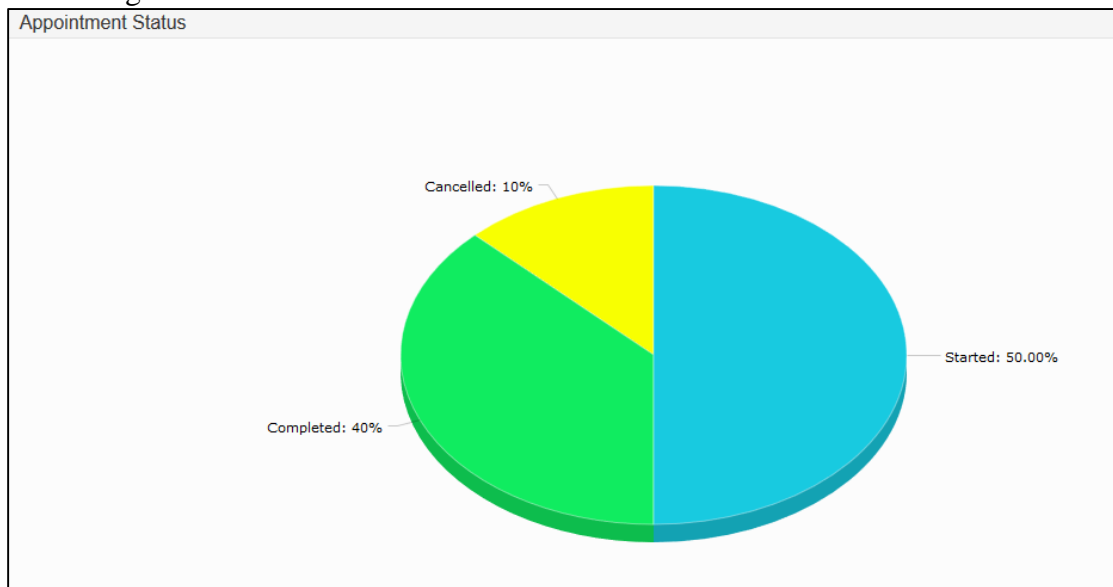


Figure C.9: Pie Chart generate using the Appointment statuses

Appendix D – Management Reports

Management reports generated via the developed system as for the crucial decision making purpose of the owner of the salon. Only the system administrator allow generating and printing the reports by the system.


Select '  Report Center ' at the side menu to continue the report viewing/ generating process.

Figure D.1 is on the report of Service wise Sales; Using this reports owner is able to analyze the skills of the employee as well as the maximum gain by each employee.

Salon Nirosha Wanasekara					
<i>590, Old Galle Road, Horethuduwa, Moratuwa</i>					
From Date: 25/10/2017					
To Date: 30/10/2017					
SALES BY STAFF - SERVICE WISE					
Staff Id	Employee Name	Service	Service Charge	Service Count	Total
ST002	L.K. Wasana	Hair Colour	4000.00	3	12000.00
		Hair cut	400.00	8	3200.00
ST004	U. Janeesha	Hair cut	400.00	2	800.00
		Nail Art	500.00	1	500.00
ST005	O.J. Susani	Manicure	1000.00	5	5000.00
		Hair Colour	4000.00	2	8000.00
Gross Total					29500.00

Figure D.1: Service wise Sales by Staff Report

Figure D.2 is on the report of Appointment wise Sales; using this reports owner is able to find the famous employee among the clients and how each employee had gain an income for the salon via appointments.

Salon Nirosha Wanasekara				
<i>590, Old Galle Road, Horethuduwa, Moratuwa</i>				
From Date: 25/10/2017				
To Date: 30/10/2017				
SALES BY STAFF - APPOINTMENT WISE				
Staff Id	Employee Name	Appointment Count	Appointment Charge	Total
ST002	L.K. Wasana	3	5000.00	15000.00
ST004	U. Janeesha	2	1600.00	3200.00
ST005	O.J. Susani	5	1000.00	5000.00
Gross Total				23200.00

Figure D.2: Appointment wise Sales by Staff Report

Figure D.3 is on the report of Service wise Income; Using this reports owner is able to analyze the most loyal client of the salon as well as the income gain by each client.

Salon Nirosha Wanasekara					
<i>590, Old Galle Road, Horethuduwa, Moratuwa</i>					
From Date: 25/10/2017					
To Date: 30/10/2017					
INCOME BY CLIENT - SERVICE WISE					
Client Id	Client Name	Service	Service Charge	Service Count	Total
CL002	B.S. Perera	Hair Colour	4000.00	1	4000.00
		Hair cut	400.00	1	400.00
CL004	L.K. Warnapura	Hair cut	400.00	1	400.00
		Nail Art	500.00	1	500.00
CL005	P.N. Zoysa	Manicure	1000.00	1	1000.00
		Hair Colour	4000.00	1	4000.00
Gross Total					10300.00

Figure D.3: Service wise Income by Client Report

Figure D.4 is on the report of Appointment wise Income; Using this reports owner is able to analyze how often the clients make appointments along with the income gain.

Salon Nirosha Wanasekara				
<i>590, Old Galle Road, Horethuduwa, Moratuwa</i>				
From Date: 25/10/2017				
To Date: 30/10/2017				
INCOME BY CLIENT - APPOINTMENT WISE				
Client Id	Client Name	Appointment Count	Appointment Charge	Total
CL002	B.S. Perera	2	4400.00	8800.00
CL004	L.K. Warnapura	2	900.00	1800.00
CL005	P.N. Zoysa	1	5000.00	5000.00
Gross Total				15600.00

Figure D.4: Appointment wise Income by Client Report

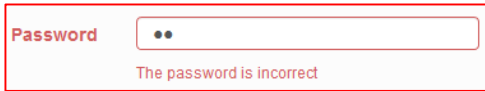
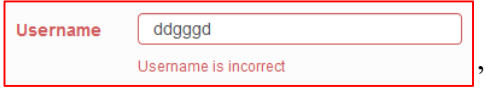
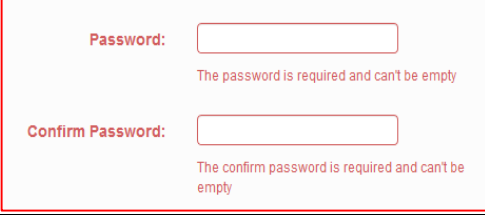
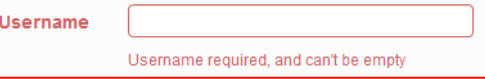
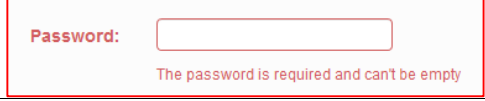
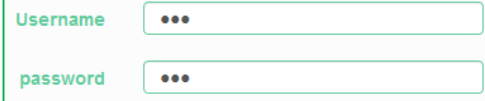
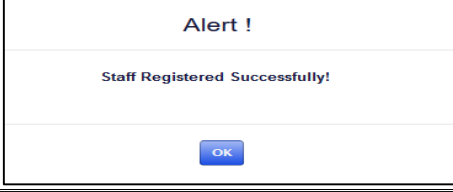

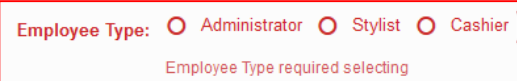

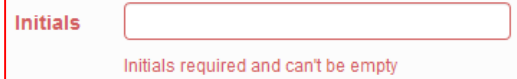
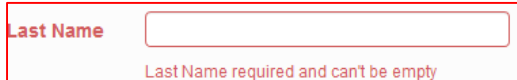

Figure D.5 is on the report of Appointment Statuses; Using this reports owner is able to analyze the statuses of each appointment.

Salon Nirosha Wanasekara					
<i>590, Old Galle Road, Horethuduwa, Moratuwa</i>					
From Date: 25/10/2017					
To Date: 30/10/2017					
APPOINTMENT STATUS					
Staff Id	Employee Name	Client Id	Client Name	Appoinment Id	Status
ST002	L.K. Wasana	CL002	B.S. Perera	AP001	Started
ST004	U. Janeesha	CL004	L.K. Warnapura	AP014	Completed
ST005	O.J. Susani	CL005	P.N. Zoysa	AP016	Cancelled
ST007	K.M. Rathnayake	CL008	K.K. KUMARI	AP017	Created

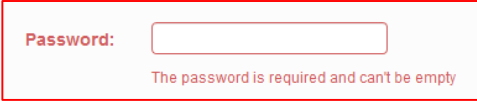
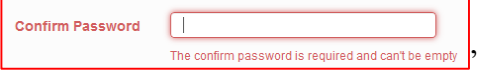
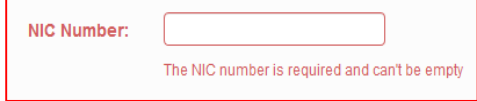
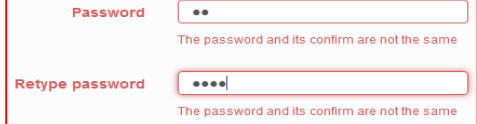
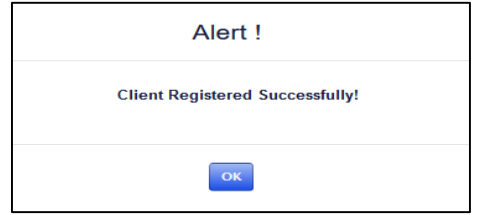


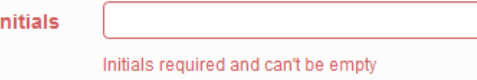
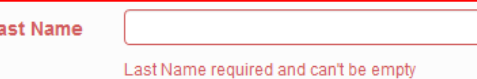
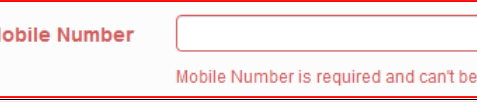
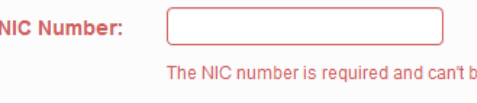
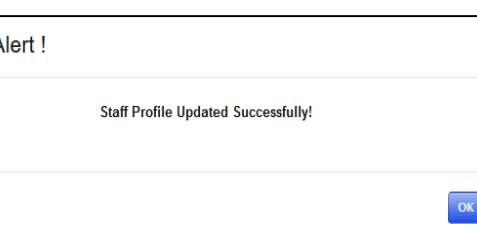
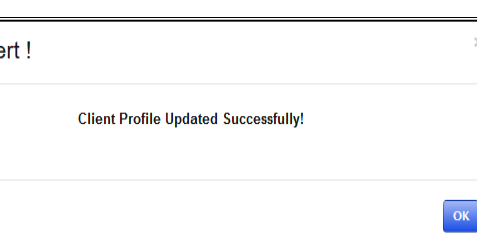
Figure D.5: Appointment Status Report

Appendix E – Test Results

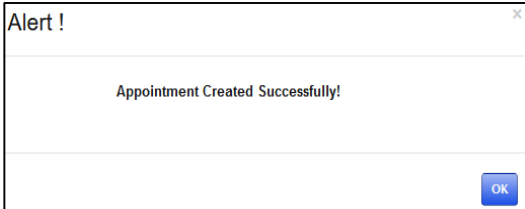


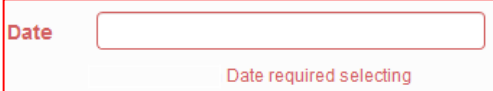
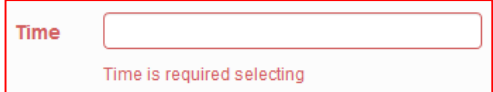
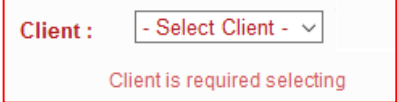
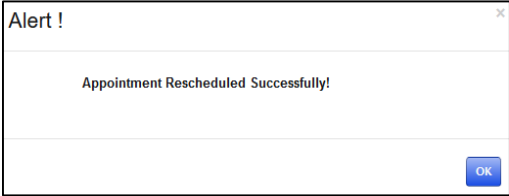
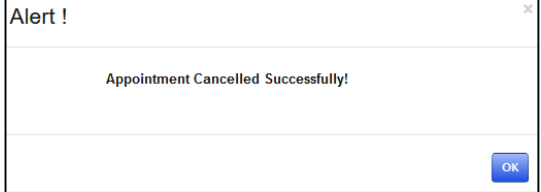
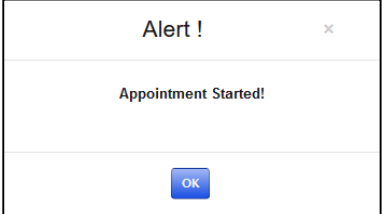
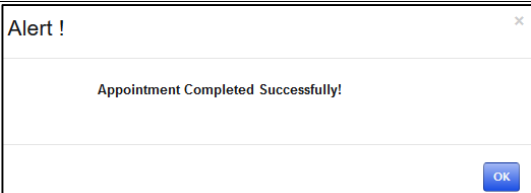
Results with Evidence of Testing were displayed at this appendix. Shown below is the actual test results table named as Table E.1.

Test No.	Module Name	Test Description	Actual Output
1	System Login	Correct Username and incorrect Password.	
2		Correct Password and incorrect Username.	
3		Blank Username and Password.	
4		Blank Username.	
5		Blank Password.	
6		Correct both Username and Password.	
7	Staff Registration	Validate Staff Information adding.	
8		If all the fields empty.	
9		Blank Employee Type	
10		Blank Employee From Date	
11		Blank Initials	
12		Blank Lastname	
13	Blank Mobile Number		

Salon Management System for 'Salon Nirosha'

14		Blank Password	
15		Blank Confirm Password	
16		Blank NIC Number	
17		If Confirm Password differ from Password	
18	Client Registration	Validate Client Information adding.	
19		If all the fields empty.	
20		Blank Client From Date	
21		Blank Initials	
22		Blank Last name	
23		Blank Mobile Number	
24		Blank NIC Number	
25	Staff Profile	Validate Update Profile Details	
26	Client Profile	Validate Update Profile Details	

Salon Management System for 'Salon Nirosha'

27	Appointment Creation	Validate Appointment Creation.	
28		If all the fields empty.	
29		Staff Not Selected.	
30		Blank Date.	
31		Blank Time.	
32		Client Not Selected.	
33	Appointment Reschedule	Validate Appointment Reschedule.	
34	Appointment Cancellation	Validate Appointment Cancellation.	
35	Appointment Status Handling	Status Changed as Started	
36		Status Changed as Complete	

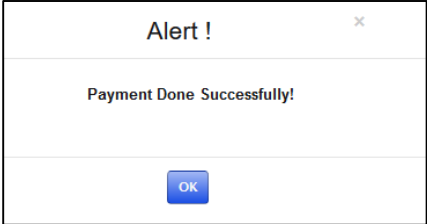
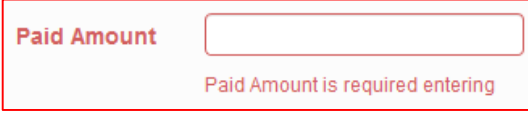
37	Payment Handling	Validate Payment	
38		Blank Paid Amount	

Table E.1: Actual Test Results Table

User Evaluation forms handover by the Owner, Stylist and Cashier are attached at the Figure E.1, Figure E.2 and Figure E.3 respectively.

**User Evaluation Form for Salon Management System for
'Salon Nirosha':**

Employee Name : Ms. Nirosha Wanasekara

Designation : Owner

Use the following ratings to mark the results:

Excellent	Very Good	Good	Average	Poor
5	4	3	2	1

#	Assessment Description	Results
1	System Performance	4
2	Usability	5
3	Security	4
4	User Interface	4
5	System Quality	5

Are you satisfied with the overall system, and are the system features accomplish your business requirement?

Yes. I'm satisfy with this system, designed and developed for my salon.

Any Other Suggestions/Comments?

As with the successful implementation, I thought of enhancing the system, by including the online appointment facility.

Employee Signature : *Nirosha*

Date : 28/11/2017

Figure E.1: User Evaluation Form filled by Owner/ System administrator

**User Evaluation Form for Salon Management System for
'Salon Nirosha':**

Employee Name : Ms. Janeesha Udawatta

Designation : Stylist

Use the following ratings to mark the results:

Excellent	Very Good	Good	Average	Poor
5	4	3	2	1

#	Assessment Description	Results
1	System Performance	5
2	Usability	5
3	Security	5
4	User Interface	4
5	System Quality	4

Are you satisfied with the overall system, and are the system features accomplish your business requirement?

Yes, I satisfy with the system features, that can be accessed by my self. It is up to my requirements

Any Other Suggestions/Comments?

With our business needs, furtherly we can improve the system in the future.

Employee Signature : *Janeesha Udawatta*

Date : 28/11/2017

Figure E.2: User Evaluation Form filled by Stylist

**User Evaluation Form for Salon Management System for
'Salon Nirosha':**

Employee Name : Ms. Harshani Kaushalya

Designation : Cashier

Use the following ratings to mark the results:

Excellent	Very Good	Good	Average	Poor
5	4	3	2	1

#	Assessment Description	Results
1	System Performance	5
2	Usability	4
3	Security	5
4	User Interface	4
5	System Quality	5

Are you satisfied with the overall system, and are the system features accomplish your business requirement?

Yes, the payment part followed at our working place has well automated.

Any Other Suggestions/Comments?

Upto now the feature is ok to deal with. With the usage new requirements may occur

Employee Signature : *Harshani Kaushalya*

Date : 28/11/2017

Figure E.3: User Evaluation Form filled by Cashier

Appendix F – Code Listing

Key codes include in the system were displayed at this appendice as to provide an idea on the overall code structure. Attached sample codes are under following categories accordingly:

- Create/ Update/ Dlete functions performs at jsp, Servlet and DAO
- Data View
- Pie Chart Generation and View

Following Figure F.1 devotes a view on a select query which read data at the database and retrieve them to as to view at the application layer (by running the system on the web browser).

```

<tbody>
  <#
    try {
      PrintWriter writer = response.getWriter();
      Connection con = JDBCUtil.getConnection();
      String sql = "SELECT * FROM ref_services";
      PreparedStatement ps = con.prepareStatement(sql);
      ResultSet rs = ps.executeQuery();

      while (rs.next()) {
        out.print("<tr onclick=javascript:showRow(this);>"
          + "<td>"
            + rs.getString(1)
          + "</td>"
          + "<td>"
            + rs.getString(2)
          + "</td>"
          + "<td>"
            + rs.getString(3)
          + "</td>"
          + "<td>"
            + "<a href='#!'><i style='font-size:20px' class='fa fa-arrow-circle-down'></i></a>"
          + "</td>"
          + "<td style='display: none;'>"
            + rs.getString(4)
          + "</td></tr>"
        );
      }
    } catch (SQLException e) {
      System.out.println(e.getMessage());
    }
  <#
</tbody>

```

Figure F.1: Sample Select Query Code

Following Figure F.2 gives a view on a create model at the jsp page of the Reference Data - Services.

```

<!-- -----Create model start----- -->
<form class="form-horizontal" role="form" action="ServletRefServices?ido=10" method="POST" id="refservicescreate" onkeyup="valid
<!-- Modal -->
  <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span></button><span class=
          <h4 class="modal-title" id="myModalLabel">Add Services</h4>
        </div>
        <div class="modal-body">
          <div class="form-group">
            <label class="col-sm-3 control-label">Service:</label>
            <div class="col-sm-6">
              <input type="text" class="form-control" name="service1" id="service1">
            </div>
          </div>
          <div class="form-group">
            <label class="col-sm-3 control-label">Duration:</label>
            <div class="col-sm-3">
              <input type="text" name="input_time1" id="input_time1" class="form-control" placeholder="Time">
            </div>
          </div>
          <div class="form-group">
            <label class="col-sm-3 control-label">Price:</label>
            <div class="col-sm-3">
              <input type="text" class="form-control" name="charge1" id="charge1">
            </div>
          </div>
        </div>
        <div class="modal-footer">
          <button type="submit" class="btn btn-primary" onclick="validateSecType();">Submit</button>
          <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
        </div>
      </div>
    </div>
  </div>
</form>
<!-- -----Create model end----- -->

```

Figure F.2: Sample Create JSP Code

Following Figure F.3 gives a view on a create servlet code of the Reference Data - Services.

```

public class ServletRefServices extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        int ido = Integer.parseInt(request.getParameter("ido"));
        Services services = new Services();

        if (ido == 10) {
            String service = request.getParameter("service1");
            String duration = request.getParameter("input_time1");
            Double charge = Double.parseDouble(request.getParameter("charge1"));

            services.setService(service);
            services.setDuration(duration);
            services.setCharge(charge);

            int result = ServiceRefServicesDAO.createService(services);
            if (result==1) {
                System.out.println("Data Added");
                response.sendRedirect("ServletSideMenuAccess?id=2");
            }else {
                System.out.println("Error! at creation");
            }
        }
    }
}

```

Figure F.3: Sample Create Servlet Code

Following Figure F.4 gives a view on a insert query at the DAO page of the Reference Data - Services.

```

package com.salon.service;

import java.sql.Connection;

public class ServiceRefServicesDAO {
    public static int createService(Services services) {
        int value = 0;
        Connection con = JDBCUtil.getConnection();
        String sql = "INSERT INTO ref_services(service_name,service_duration,service_charge)VALUES(?, ?, ?)";
        try {
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, services.getService());
            ps.setString(2, services.getDuration());
            ps.setDouble(3, services.getCharge());
            value = ps.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return value;
    }
}

```

Figure F.4: Create related DAO

Following Figure F.5 gives a view on a modify form at the jsp page of the Reference Data - Services.

```

<!-- -----Modify model start----- -->
<form class="form-horizontal" role="form" action="ServletRefServices?ido=11" method="post" id="refservicesmodify" onkeyup="valid
    <div class="row">
        <div class="col-xs-12 col-sm-12">
            <div class="box">
                <div class="box-content">
                    <input type="hidden" name="serviceid" id="serviceid">
                    <div class="form-group">
                        <label class="col-sm-3 control-label">Service:</label>
                        <div class="col-sm-4">
                            <input type="text" class="form-control" name="service" id="service">
                        </div>
                    </div>
                    <div class="form-group">
                        <label class="col-sm-3 control-label">Duration:</label>
                        <div class="col-sm-3">
                            <input type="text" name="input_time" id="input_time" class="form-control" placeholder="Time">
                        </div>
                    </div>
                    <div class="form-group">
                        <label class="col-sm-3 control-label">Price:</label>
                        <div class="col-sm-3">
                            <input type="text" class="form-control" name="charge" id="charge">
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-xs-12 col-sm-12">
            <div class="box">
                <div class="box-content">
                    <center>
                        <button type="button" class="btn btn-primary btn-label-left" data-toggle="modal" data-target="#myModal" dis
                        <button type="submit" onclick="validateLoanTypeModify();" value="modify" id="modify" class="btn btn-success
                        <button type="button" data-toggle="modal" data-target="#myModal" value="delete" id="delete" class="btn btn-
                    </center>
                </div>
            </div>
        </div>
    </div>
</form>
<!-- -----Modify model end----- -->

```

Figure F.5: Modify related JSP

Following Figure F.6 gives a view on a modify servlet code of the Reference Data - Services.

```

}else if (ido == 11) {
    int id = Integer.parseInt(request.getParameter("serviceid"));
    String service = request.getParameter("service");
    String duration = request.getParameter("input_time");
    Double charge = Double.parseDouble(request.getParameter("charge"));
    services.setService(service);
    services.setDuration(duration);
    services.setCharge(charge);
    services.setId(id);
    int result = ServiceRefServicesDAO.modifyService(services);
    if (result==1) {
        System.out.println("Data Modified");
        response.sendRedirect("ServletSideMenuAccess?id=2");
    }else {
        System.out.println("Error! at modification");
    }
}

```

Figure F.6: Modify related Servlet

Following Figure F.7 gives a view on a update query at the DAO page of the Reference Data - Services.

```

public static int modifyService(Services services) {
    int value = 0;
    Connection con = JDBCUtil.getConnection();
    String sql = "UPDATE ref_services SET service_name=?, service_duration=?, service_charge=? WHERE id=?";
    try {
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, services.getService());
        ps.setString(2, services.getDuration());
        ps.setDouble(3, services.getCharge());
        ps.setInt(4, services.getId());
        value = ps.executeUpdate();
    } catch (SQLException e) {
        System.out.println(e.getMessage());
        e.printStackTrace();
    }
    return value;
}

```

Figure F.7: Modify related DAO

Following Figure F.8 gives a view on a delete record at the jsp page of the Reference Data - Services.

```

<!-- -----Delete model start----- -->
<form class="form-horizontal" role="form" method="post" action="ServletRefServices?ido=12" id="refservicesdelete">
  <!-- Modal -->
  <div class="modal fade" id="myModal1" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span><span class="
          <h4 class="modal-title" id="myModalLabel">Are you surely want to delete?</h4>
        </div>
        <div class="modal-body">
          <input type="hidden" id="del_serviceid" name="del_serviceid">
        </div>
        <div class="modal-footer">
          <button type="submit" class="btn btn-danger">Delete</button>
          <button type="button" class="btn btn-default" data-dismiss="modal">Cancel</button>
        </div>
      </div>
    </div>
  </div>
</form>
<!-- -----Delete model end----- -->

```

Figure F.8: Delete related JSP

Following Figure F.9 gives a view on a delete servlet code of the Reference Data - Services.

```

}else if (ido == 12) {
    int id = Integer.parseInt(request.getParameter("del_serviceid"));
    services.setId(id);

    int result = ServiceRefServicesDAO.deleteService(services);
    if (result==1) {
        System.out.println("Data Deleted");
        response.sendRedirect("ServletSideMenuAccess?id=2");
    }else {
        System.out.println("Error! at deletion");
    }
}

```

Figure F.9: Delete related Servlet

Following Figure F.10 gives a view on a delete query at the DAO page of the Reference Data - Services.

```

public static int deleteService(Services services) {
    int value = 0;
    Connection con = JDBCUtil.getConnection();
    String sql = "DELETE FROM ref_services WHERE id=?";
    try {
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setInt(1, services.getId());
        value = ps.executeUpdate();
    } catch (SQLException e) {
        System.out.println(e.getMessage());
        e.printStackTrace();
    }
    return value;
}

```

Figure F.10: Delete related DAO

Figures at Figure F.11 and F.12 describes the pie chart included at the system to view the Appointment Statuses.

```

Connection con = null;
Statement statement = null;
con = JDBCUtil.getConnection();

String query1 = "SELECT count(status) FROM form_appointmentstatus where status='pending'";
statement = con.createStatement();
ResultSet rs1 = statement.executeQuery(query1);
rs1.next();
int pending = rs1.getInt(1);
System.out.println(pending);

String query2 = "SELECT count(status) FROM form_appointmentstatus where status='cancelled'";
statement = con.createStatement();
ResultSet rs2 = statement.executeQuery(query2);
rs2.next();
int rejected = rs2.getInt(1);

String query3 = "SELECT count(status) FROM form_appointmentstatus where status='started'";
statement = con.createStatement();
ResultSet rs3 = statement.executeQuery(query3);
rs3.next();
int appraised = rs3.getInt(1);

String query4 = "SELECT count(status) FROM form_appointmentstatus where status='completed'";
statement = con.createStatement();
ResultSet rs4 = statement.executeQuery(query4);
rs4.next();
int approved = rs4.getInt(1);

```

Figure F.11: Query to select data from 'Appointment Status' table

```

<script type="text/javascript">
var chart = AmCharts
    .makeChart(
        "chartdiv",
        {
            "type" : "pie",
            "theme" : "none",
            "dataProvider" : [ {
                "status" : "Pending",
                "value" : <out.print(pending);%>
            }, {
                "status" : "Cancelled",
                "value" : <out.print(rejected);%>
            }, {
                "status" : "Started",
                "value" : <out.print(appraised);%>
            }, {
                "status" : "Completed",
                "value" : <out.print(approved);%>
            } ],
            "valueField" : "value",
            "titleField" : "status",
            "outlineAlpha" : 0.4,
            "depth3D" : 15,
            "balloonText" : "[[title]]<br><span style='font-size:14px'><b>[[value]]</b> ([[percents]]%)</span>",
            "angle" : 30,
            "exportConfig" : {
                menuItems : [ {
                    icon : '/lib/3/images/export.png',
                    format : 'png'
                } ]
            }
        }
    );
jQuery('.Chart-input').off().on('input change', function() {
    var property = jQuery(this).data('property');
    var target = chart;
    var value = Number(this.value);
    chart.startDuration = 0;

    if (property == 'innerRadius') {
        value += "%";
    }

    target[property] = value;
    chart.validateNow();
});
</script>

```

Figure F.12: Javascript required to run the pie chart at the system

Appendix G – Client Certificate



No. 590, Old Galle Road,
Horethuduwa, Moratuwa,
Sri Lanka

Phone: +94 11 2648303
Fax: +94 77-3035396
Email: wanasekaran@sltnet.lk

26th October 2017

The coordinator,
External Degree center,
University of Colombo School of computing.

Dear Sir,

Certification Letter on the acceptance of Salon Management Software

I hereby certify that 'Salon Nirosha' has accepted the Salon Management Software, designed, developed and implemented at our salon premise by Miss. G.J.V.P.S.O.Jayawardena (R141179) along with the system user manual.

We are happy to announce that the system was upto our original business requirement, and we hope the system will helps us to enrich our salon management processes efficiently and effectively. We are proud of using a local software product developed by a Sri Lankan student and thank her for selecting 'Salon Nirosha' as her client.

Thank you.

Yours sincerely,

Ms. Nirosha Wisanthi Wanasekara

Salon Nirosha Wanasekara
No.590, Old Galle Road, Horethuduwa,
Moratuwa.
TEL : 0112648303 / 0773035396
email : salonniroshaw@yahoo.com
www.salonnirosha.com

Glossary

Actions – A step in the activity wherein the users or software perform a given task.

Apache – This is a free, open-source cross-platform web server software which has been released under the terms of Apache License 2.0.

Bootstrap – This is a front-end web framework which is free and open-source, and use for designing websites and web applications

Beauty Parlor - A beauty salon or beauty parlor is an establishment dealing with cosmetic treatments for men and women.

Connectors - The flow between steps in the diagram.

Decision Node - A conditional branch in the flow which includes a single input and more outputs; this represented with a diamond.

JQuery – This is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.

MVC – This is a software architectural pattern use for implementing user interfaces on computers.

Pie Chart - A pie chart is a circular statistical graphic which is divided into slices to illustrate numerical proportion.

Servlet – This is a Java class that use to extend server capabilities while runs in java enabled server.

Index

A	
acceptance.....	ii, 35
acknowledgement.....	iii
Analysis	3, 4
appointments.....	ii, 2, 4, 5, 9, 12, 24, 58, 59, 63, 66, 67
C	
Class	15, 20
Conclusion	3, 41
Controller	ix, 33
customers	ii, 4, 10
D	
database	ii, 9, 15, 21, 28, 29, 31, 33, 48, 49
Design	3, 14, 15, 51
E	
employees	ii, 61
Evaluation	3, 35
I	
Implementation	3, 25
Information	i, iii, 12, 65
M	
Management	ii, ix, 24, 51, 52, 53, 54, 55, 56, 57, 58, 59, 66
Methodology	ii, 13, 15
Model	ix, 33
O	
Objectives	2
owner.....	ii, iii, 4, 14, 66, 67, 68
R	
reminders	ii, 4, 5
S	
Salon	ii, iii, 1, 2, 4, 5, 9, 61
Scope.....	2
SDLC	ii, ix, 3, 13, 25
services.....	ii, 1, 2, 4, 7, 10, 23
T	
testing	ii, 26, 35
U	
Use Case ...	5, 15, 51, 52, 53, 54, 55, 56, 57, 58, 59
V	
View	ix, 2, 10, 33, 53, 54, 59, 65, 76