



Online Salon Management System
For
Samudra Bridal Palace

J.A.J.N. Jayaweera

BIT Registration No: R142099

Index No: 1420992

2017/2018

Supervisors: P.D. Dilshan Navinda

Dimuthu Lakshan Amithasekare




**This dissertation 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 dose 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 herby give consent for my dissertation, if accepted, to be made available for photocopy in and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

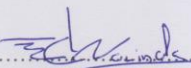
Signature of Candidate : 

Name if the Candidate : J. A. J. N. Jayaweera Date: 2017.08.15

Countersigned by:

Signature of Supervisor 01 : 

Name if the Supervisor 01 : Dimuthu Lakshani Date: 2017.08.15

Signature of Supervisor 02 : 

Name if the Supervisor 02 : Dilshan Navinda Date: 2017.08.15

Figure: Declaration

ABSTRACT

Samudra Bridal Palace is a well-recognized and reputed salons among peoples and modelers around the Maharagama area. They have three salons around Maharagama, Baralasangamuwa and Nugegoda areas. Day by day their customer base is growing up and current manual process not efficient enough to provide quick and reliable service to them. Also hard to maintain salon internal process. Issuing reliable report is much needed, Because of these salon owner willing to manage their salons from one place and also to keep their well reputation among salons and peoples much needed.

They have no automation system for their process. Even if they use computers and tablet they do not have proper system to full fill their needs. They are using record book to tracking their day by day process in salons. This is very inefficient and unreliable. And keeping records not accurate with relevant salon data when calculate reports. And also salary calculation of employees not easy.

This system will provide the functionalities of manage salons' details, manage day by day bookings, manage issuing many reports (employee's salary report, product report, booking report etc.), manage service, billing, manage employee details, manage customer details and manage notification system for day to day transactions. This system will help to improve the higher efficiency in the processes in the salons. It will enable salon staff to add data to the system without bothering about creating report formats that will increase the efficiency of salon staff. This system will used employee's ids and passwords for employees to access the system. In this case salon's confidential data will not be available to the unauthorized access.

The project was implemented using Visual Studio 2012 as the IDE and SQL server used as the DBMS. Coding was done using the C# programming language and angular 2(JavaScript framework). In addition, Visual Paradigm for UML was used to draw the UML diagrams shown in this interim report.

The system will provide user friendly simple interface which will help them to make reliable, quick and effective service to their customers.

ACKNOWLEDGEMENTS

I take this space to acknowledge and extend my heartfelt gratitude to those who have helped me in various ways throughout the project work to make this project a success.

First, I wish to express my gratitude to the BIT Coordinator of University of Colombo School of Computing (UCSC) and project examination board of Bachelors of Information Technology (BIT) for giving me the opportunity to apply the knowledge gained through the BIT degree program.

My sincere appreciation goes to the Samudra Bridal Palace owner Mr. Samudra Siriwardana and staff, who enlightened me about the processes and functionalities of the salon and co-operated in the course of development of the project.

In the mean time I would like to thanks my supervisors Mr. Dimuthu Lakshan Amithasekara and Mr. Dilshan Navinda for counseling and supervising me throughout the development of the project.

I would especially thanks to Mr. Sumedha Milroy for guiding me through all phases of the project and helping for the dissertation document.

I am also grateful to my family and friends especially to my parents for support given in every way possible throughout the process of this degree program of three years.

TABLE OF CONTENTS

DECLARATION	i
ABSTRACT.....	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES.....	vii
LIST OF TABLE.....	ix
LIST OF ACRONYMS.....	x
CHAPTER 01-INTRODUCTION.....	1
1.1. Background	1
1.2. Motivations For The Project.....	1
1.3. Scope Of The Project.....	2
1.4. User Requirements	3
1.5. System Requirements	4
1.5.1. Requirements for Web Hosting	4
1.5.2. End User Requirements	4
1.5.3. Requirements For Software	4
1.6. Outline Of The Remaining Chapters	4
CHAPTER 02-ANALYSIS.....	6
2.1. Current Manual System	6
2.2. Weaknesses Of Current Manual System.....	7
2.3. Outline of Existing Similar Solution.....	7
2.4. Requirement Gathering	9
2.5. Requirement Analysis	10
2.6. Functional Requirements.....	10
CHAPTER 03-DESIGN	11
3.1. Alternative Solutions.....	11
3.2. Selected Solution	11
3.3. Software Development Process Model	12
3.4. Relevant Design Diagrams.....	12
3.4.1. Use Case Diagram	12
3.4.2. Class diagram for the system.....	16
3.4.3. Activity Diagram	18
3.4.4. Sequence Diagram	19
3.4.5. Database Design And Entity Relationship Diagrams.....	21

3.4.6. User Interface Design.....	22
CHAPTER 04: IMPLEMENTATION	24
4.1. Hardware and Software Requirements	24
4.2. Development Tools	24
4.3. Code Feature.....	27
1. Design Architecture.....	27
Major Code Segments.....	29
1. Database Connection	29
2. Service Page Add Model	29
3. Service Component Save Function.....	32
4. Service Layer for add new salon service	33
5. Web API Layer (Application programming interface)	34
6. Service API.....	34
7. Service Business Layer	34
8. Data Access layer	34
9. Stored Procedure	36
4.4. Re-usable Components	38
CHAPTER 05: EVALUATION	39
5.1. Requirement for A Good System Evaluation	39
5.2. Test Plan.....	39
5.3. Test Strategies.....	40
5.3.1. Unit testing.....	40
5.3.2. Black box testing	40
5.3.3. White box testing.....	40
5.3.4. Integration testing.....	41
5.3.5. System testing	41
5.4. Test Case Generation	42
5.4.1. Test Case and Test Result for Service Category	42
5.1.1. Test Cases and Test Result for Service	44
5.5. User Evaluation	47
CHAPTER 06: CONCLUSION.....	51
6.1. Overview	51
6.2. Lessons Learnt.....	51
6.3. Critical Assessment of Project.....	51
6.4. Future Enhancements	52
REFERENCE.....	53

APPENDIX A – SYSTEM DOCUMENTATION	55
APPENDIX B – DESIGN DOCUMENTATION	62
APPENDIX C – USER MANUAL	64
System Login	64
Home Page	64
Handel Booking	65
Create Bookings	65
View Booking.....	69
Cancel Booking.....	69
Modify Booking	70
Complete Booking.....	70
APPENDIX D – MANAGEMENT REPORTS	72
APPENDIX E – TEST RESULTS	75
APPENDIX F – CODE LISTING	78
Delete Service Form System	78
Service Layer for delete service	79
Web API Layer	79
Service API.....	79
Service Business Layer	80
Data Access Layer.....	80
Stored Procedure	81
APPENDIX G – CLIENT CERTIFICATE	82
INDEX	83

LIST OF FIGURES

Figure: Declaration	i
Figure 2.1: Use Case Diagram For Current Manual System	6
Figure 2.2: Main Window of Salon Management System	7
Figure 2.3: Booking View of the JBS System	8
Figure 2.4: Salon Report Creating View	8
Figure 3.1: Use Case Diagram for Owner	13
Figure 3.2: Use Case Diagram for Salon Admin	14
Figure 3.3: Use Case Diagram for Employee	15
Figure 3.4: Class Diagram for New System	17
Figure 3.5: Activity Diagram for System Login	18
Figure 3.6: Activity Diagram for Add New Service	19
Figure 3.7: Sequence Diagram For Add Booking	20
Figure 3.8: Sequence Diagram For Add Service	20
Figure 3.9: Entity Relationship Diagram for the System	21
Figure 3.10: Login Window of the System	22
Figure 3.11: Main Window of the System	23
Figure 3.12: Employee Manage Window	23
Figure 4.1: Solution Structure for Projects in Visual Studio	28
Figure 4.2: Service Add Model	30
Figure 4.3: Angular 2 Date Picker View	38
Figure 4.4: Day Pilot Pro Angular view	38
Figure 5.1: Service Category View	43
Figure 5.2: Validation for Required Fields	43
Figure 5.3: Delete Confirmation Message	44
Figure 5.4: Status Change to “True” for Deleted Records	44
Figure 5.5: Service View	45
Figure 5.6: Validation for required fields in Add Service view	46
Figure 5.8: In DB service table status changed to 1	46
Figure 5.9: User Feedback from Salon Owner	48
Figure 5.10: User Feedback from Beautician	49

Figure 5.11: Summary Of Product Satisfaction Survey	50
Figure A.1: Install SQL Server 2012 step 1	55
Figure A.2: Install SQL Server 2012 step 2	56
Figure A.3: Install SQL Server 2012 step 3	56
Figure A.4: Install SQL Server 2012 step 4	57
Figure A.5: Install SQL Server 2012 step 5	57
Figure A.6: Install SQL Server 2012 step 6	58
Figure A.7: Install SQL Server 2012 step 7	58
Figure A.8: Install SQL Server 2012 step 8	59
Figure A.9: Install SQL Server 2012 step 9	59
Figure A.10: Install SQL Server 2012 step 10	60
Figure B.1: Activity diagram for add new branch salon	62
Figure B.2: Activity diagram for deleting a services	63
Figure C.1: Login view	64
Figure C.2: Home Page View	65
Figure C.3: Branch Selecting View	65
Figure C.4: Booking Dashboard	66
Figure C.5: Booking Panel – Service	66
Figure C.6: Booking Panel - Customer	67
Figure C.7: Booking Panel - Confirmation	68
Figure C.8: Available Bookings Shown On the Booking Dashboard	68
Figure C.9: Booking Details View	69
Figure C.10: Cancel Booking Confirmation Message	69
Figure C.11: Modify Booking View	70
Figure C.12: Invoice Generates When Complete Booking	71
Figure D.1: Salon service bookings report	72
Figure D.2: Employee attendance report	73
Figure D.3: Employee booking count report	73
Figure D.4: Daily income from completed appointments	74
Figure G 1: Client Certificate	85

LIST OF TABLE

Table 3.1: Use case description for System Login	15
Table 3.2: Use case description for Generate Salon's Reports	16
Table 3.3: Use case description for Create Bookings	16
Table 3.4: Use case description for complete booking	16
Table 5.1: Test Cases for Service category	43
Table 5.2: Test Cases for Service	45
Table D 1: Test result for online booking	75
Table D 2: Test result for add booking from dashboard	76
Table D 3: Test result for cancel booking from dashboard	76
Table D 4: Test result for complete booking from dashboard	77

LIST OF ACRONYMS

OSMS	- Online Salon Management System
SBP	- Samudra Bridal Palace
RAM	- Random Access Memory
HDD	- Hard disk drive
Gb	- Gigabyte
RAD	- Rapid Application Development
OOAD	- Object Oriented Analysis and Designing
ORM	- Object Relational Mapper
DBMS	- Database Management System
UML	- Unified Modeling Language
ER	- Entity-Relationship
IIS	- Internet Information Services
IDE	- Integrated Development Environment
API	- Application Programming Interface
CSS	- Cascading Style Sheets
ISO	- International Organization for Standardization
IEC	- International Electrotechnical Commission
FCL	- Framework Class Library
CLR	- Common Language Runtime
EF	- Entity Framework
MVC	- Model–View–Controller pattern
XAML	- Extensible Application Markup Language
HTML	- HyperText Markup Language
WCF	- Windows Communication Foundation
SOA	- Service Oriented Architecture
WSDL	- Web Services Description Language
SQL	- Structured Query Language
ACID	- Atomicity, Consistency, Isolation, Durability

CHAPTER 01-INTRODUCTION

1.1. Background

Day by day modern beauty culture is getting rapidly popular among the peoples, especially among young men and woman. Also they are having a habit of preferring a usual one beautician. So the salon owners are eager to provide their services at its best to the customers for make sure they are remain with salon as long as. In this case they are looking for best way to manage their salons in more efficiencies and keep customers attractive to salons.

Samudra Bridal Palace is a well-recognized and reputed salons around the Boralasgamuwa area. The Salon started about five years ago with one employees in Boralasgamuwa town. By now they have three salon around Maharagama, Nugegoda, Boralasgamuwa areas. Day by day their customer base is growing and current manual process not efficient enough to provide quick and reliable service to them. Managing reliable booking is much needed also generate reports, Because of these bookings directly involve in day by day salons service.

1.2. Motivations For The Project

- Samudra Bridal Palace is well-recognized and reputed group of salons in the area.
- They have no automated salon management system for manage their salon process.
- Most of the employees are using smart phones and tablets, so eventually they got the basic idea about how to use web browsers and access internet and work on it. And they can easily adept to this online web base salon management system with simple system introduction of how to use it.
- The system will provide user friendly simple interface which will help them to make reliable, quick and effective service.

1.3. Scope Of The Project

The proposed system will provide manage day to day salon process easily.

- **Manage branch details :-**
Salon owner can manage his salons in one location. Each branches have separate login and separate branch account. Salon owner and SBP admin can generate reports with all possible areas according branches and as all in one branches. Also can manage branch properties add employee / add service etc.
- **Manage employee details :-**
At the beginning there was only few staff members with Samudra salon. Now, with the expansion of their service to many region around the area they had to recruit new staff members. With the increase of staff It hard to keep records without proper system. To keep employees details there should have module for manage employee. It should include manage employee attendance too.
- **Manage service details :-**
Every salon has offer veracity of services related to beauty culture. Salon has to maintain a service list with the price and service details. It's also help full for salon employees to deal with customers. Especially when creating bill. To full fill this manage service module will be added to the system.
- **Manage bookings by salon :-**
Salon's customers can directly call to salon and salon admin should create booking for that customer with specific service and time. Additionally booking can assign specific employee.
- **Manage invoice :-**
Customers had to pay full billed amount when the service received. In this case the salon front desk should be able to generate bill for customers. And the payment details for the bill should be recorded in to the system.
- **Generate reports :-**

System reports are most important for owner to get valuable decisions and prediction in management. Many types of reports are allowed in system. Daily summary of payment details, summary of booking, monthly summary of them, and most requested reports types by salon owner, and so on.

- Manage customer details :-
Their customers are mostly regular basis. So they are planning to keep historical data to evaluate who got service and product from their salons. For this they are need to keep customers details against the issued reports historical data.
- Manage product details :-
Each day salon expends lot of products and equipment in their activities. So it should be manage to prevent run out stocks. General reorder points for those items set into system and system will manage message to indicate before stock ran out.

1.4. User Requirements

This system will have four potential users as given below,

- Owner
- Salon Admin
- Employees
- Customers

Main requirement of this system is manage salon process in efficient way. This system will provide for it,

- Booking Module
- Product Module
- Service Module
- HR Module
- OSMS Settings Module
- Report Module
- Billing Module

1.5. System Requirements

At the end of project system should works on customer's environment. There are requirements in Hardware and Software to run the new system.

1.5.1. Requirements for Web Hosting

Web server requirements for host the application.

- Cloud base IIS v.10 server
- 2Gb RAM
- Unmetered bandwidth
- Unmetered storage

1.5.2. End User Requirements

Hardware requirements for this system as given below,

- Computer or any mobile device able to run web browser and stable internet connection.

1.5.3. Requirements For Software

- Software requirements for this system is any web browser runs on any operating system.

1.6. Outline Of The Remaining Chapters

After the Introduction chapter, the dissertation contains five main chapters.

Chapter 02: ANALYSIS - The Analysis chapter describes the existing system, requirements and feasibility study for the proposed system and the process model used to develop the system.

Chapter 03: DESIGN - The Design chapter includes some diagrams relating to the system design. It further describes the designing approach used and designing of the data base as well as the user interfaces.

Chapter 04: IMPLEMENTATION - The main codes and the software and hardware environment used for implementing the system are described in the Implementation chapter.

Chapter 05: EVALUTION - The test plan and test cases of the system are given in the Evaluation chapter.

Chapter 06: CONCLUSION - The final chapter, the Conclusion includes lessons learnt and the information about further development of the system.

After the main chapters there is a Reference section where all the materials referred to write the dissertation are given. Furthermore in the appendices, System Documentation, Design Documentation, Management Reports, Test Results, Code Listing and the. Finally, a Client Certificate is provided.

CHAPTER 02-ANALYSIS

This section will cover the process of discovering, analyzing, defining, and documenting the requirements that are related to the business.

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements [1].

2.1. Current Manual System

They have been using Microsoft Word and Excel files to keep their day to day record. They keep records of day to day use items and equipment in excel files which maintain separate files in each salon. And also each day they are creating separate word file as summary for daily income. Once owner visit the salon he collect the files and do calculation to get rough idea about the salon income and expenses. This method is very in reliable, insufficient and insecure. Even they use computers and printers they do not have proper system to full fill their needs.

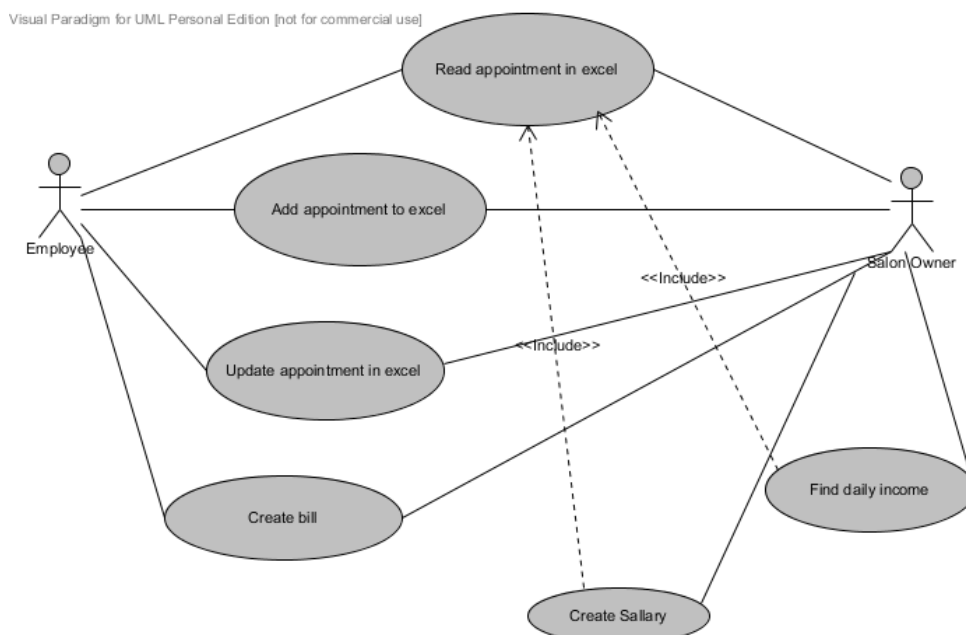


Figure 2.1: Use Case Diagram For Current Manual System

2.2. Weaknesses Of Current Manual System

When carried out further analysis of current manual system some weaknesses were identified as follows,

- No proper way to manage customers' appointments.
- No proper plan for restock the salon items and equipment.
- Cannot analysis there historical data to make predictions.
- No data backup option.
- Very often these files get attack by computer virus and lost information.
- Anyone can access these files and salon's confidential details not protect from unauthorized access.
- Do not have employee details management system.
- Took more time to preparing bill.

2.3. Outline of Existing Similar Solution

2.3.1. JustBookSalon

JustBookSalon, a popular salon directory software is being used by more than 1500 salons across the world. It is highly user friendly, reliable and easily configurable according to needs of salons. Figure 2.2 describes the main window of the JustBookSalon system [2].

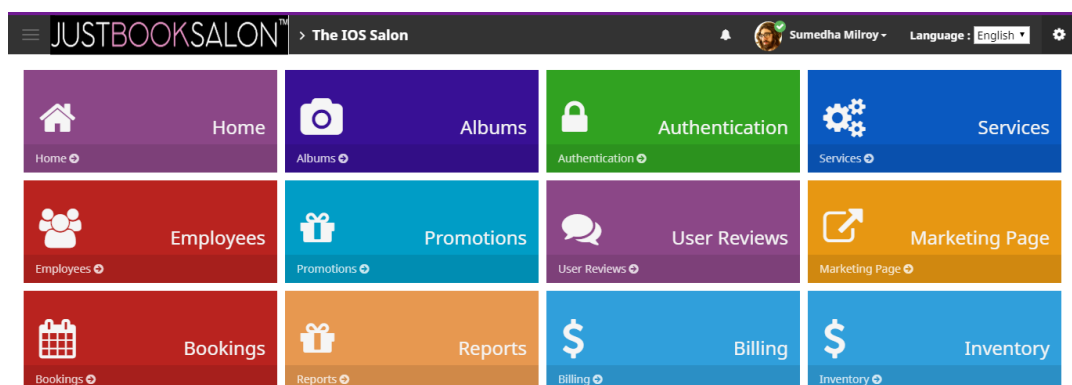


Figure 2.2: Main Window of Salon Management System

The system has features like Billing, Booking, Services management, Employee management, Product management, Maintain salon details, Manage authentication and Manage promotions according services and generate various salon reports.

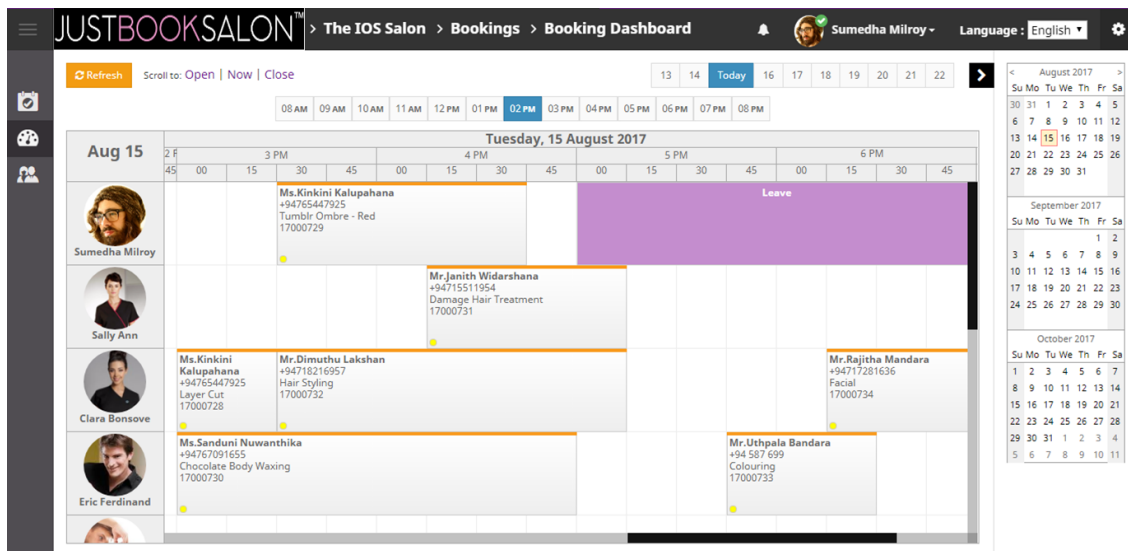


Figure 2.3: Booking View of the JBS System

Salon report format creating view shown on Figure 2.4 describes the details about customers.

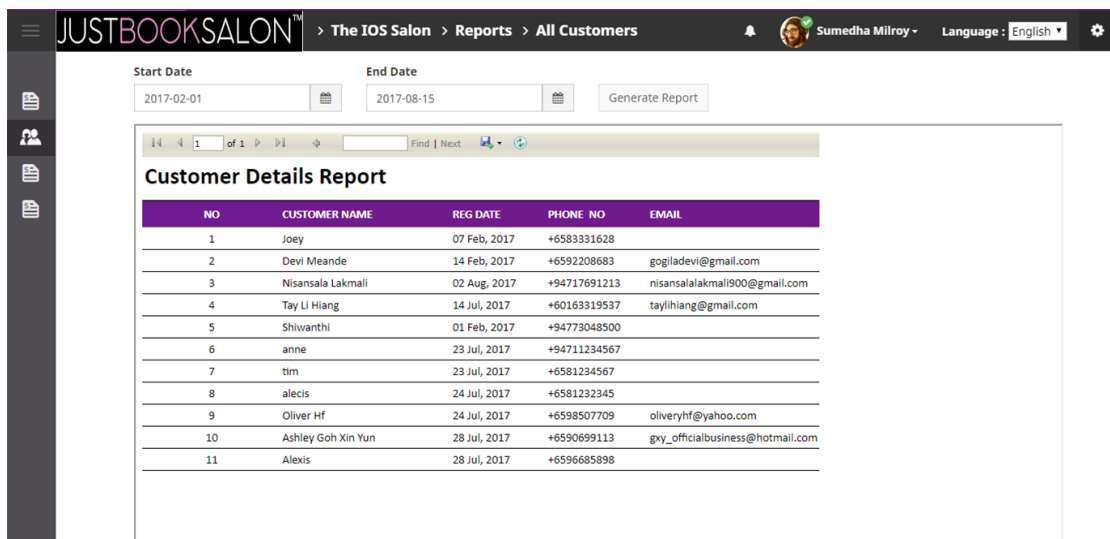


Figure 2.4: Salon Report Creating View.

2.4. Requirement Gathering

Requirement gathering is the most crucial part of the software development life cycle. The system is built by using user requirements. Otherwise it is difficult to develop a good product. In this project interviewing and discussion are the main requirement gathering techniques [3].

Following requirement gathering techniques were used gather their requirements.

- **Interview user :-**

By using interview it can get clear idea about currently on going manual process. Users of different levels can be interviewed and that will help to facilitate users with their different needs.

- **Observation :-**

The study of users in their natural habitats is what observation is about. By observing users, an analyst can identify a process flow, awkward steps, pain points and opportunities for improvement. Observation can be passive or active (asking questions while observing). Passive observation is better for getting feedback on a prototype (to refine requirements), where active observation is more effective at getting an understanding of an existing business process. Either approach can be used to uncover implicit requirements that otherwise might go overlooked

- **Document Analysis :-**

Before the system is designed it has to analysis books, bills and other documents of their manual system. This will help to get the domain knowledge of the business process.

- **Prototyping :-**

Prototyping is a final requirement gathering technique used in this business process to analyze the requirement. Prototyping was a simple working system which has been built and it was demonstrated to users and then requested to submit their requirements, and changes to be made system. Prototyping is a good method when the user does not have a clear idea about the system.

2.5. Requirement Analysis

After gathering requirements from system owner and system users, requirements are well analyzed for identify the each user's role in the proposed system and witch are the functions should have each user role. In this case Use case diagrams were used for simplify the analysis. Using a use case diagram can clearly identify external internal user roles and use case for each user roles.

Identified Critical User roles of the system

- Owner
- Salon Admin
- Employees
- Customers

2.6. Functional Requirements

- Booking Dashboard– Manage booking very easily without having mistaken on booking of the already schedule in system.
- Online Bookings – Allow customers to schedule their appointments online.
- Manage Branches (Salons) – Allow salon owner to manage all the salons' information within one place.
- Manage stock of products and equipment – allow owner to maintain stock without runout the item or products.
- Manage Billing – Avoiding calculation of billing by human hand make it easy user and reliable to customer.
- Manage Employee - It is very easy to access employee details because all the data of the employees who are working in the salons are stored in the database. But only authorized persons can view employee details.
- Manage Authentication – Improve reliability of the system and the protect confidentiality of patients' by manage access to system
- Manage Services– Authorized person can manage services according salons.

CHAPTER 03-DESIGN

Design is also a very important part in the project. Design data base and design user interfaces done in this phase. According to the IEEE definition, design is both “the process of defining the architecture, components, interfaces, and other characteristics of a system or component” and “the result of [that] process”

3.1. Alternative Solutions

There are two different alternates identify for the OSMS for SBP during the requirement analysis. Those suggested alternate is

- Stand Alone System

Standalone systems are able to function without getting help from the internet and all the path of the system are hosted on single host. This type of system not suitable for the Sanudra Bridal Palace as they planning to connect their other salons to manage within system.

- Web Base System

3.2. Selected Solution

- Web Base System

For the current process of the salons, they are using computers to save daily records. Also they have internet facility in all the branches as they provide free Wi-Fi facility to the customers. And owner has android tab. Also he is hoping to use this system from his tab. So he needed the system to be compatible with his mobile devices, laptop and also with desktop computer in salon.

I have suggested and explained owner, to serve all these needs we should developed web base system. Then the owner of SBP was agreed to develop a web base system that this running on the any device connected to internet.

After discus with my supervisor it was suggested web service integrate to system for facilitate future enhancement like mobile app development.

Finally the system was developed with combination of C# and Angular2 as Web base system [4].

3.3. Software Development Process Model

There are several types of software development process models such as

- Waterfall Model
- Prototype Model
- Rapid Application Development (RAD)
- Spiral Development
- Incremental Development

3.4. Relevant Design Diagrams

Incremental Development were chosen from the above mentioned models.

Following Models were used to develop the process of the system,

- Use Case Diagram
- Sequence Diagram
- Class Diagram
- Entity Relationship Diagram
- Activity Diagram

C# is the language that uses to develop the system which is fully support to Object Oriented Concepts, and also Entity Framework use as the Object Relational Mapper (ORM). The proposed system were built on the Layered architecture.

3.4.1. Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well [5].

Following figure 3.1 illustrates the identified use cases for salon owner in the new system. It's simply describes the functionalities for owner in the system. In addition to the other users of system owner has additional use cases for system reports. Using system reports owner can get better decisions on the salon management.



Figure 3.1: Use Case Diagram for Owner

Following Figure 3.2 is for identified use cases for salon admin in salon. Its express the deferent use cases that individual branch (salon) on the system.

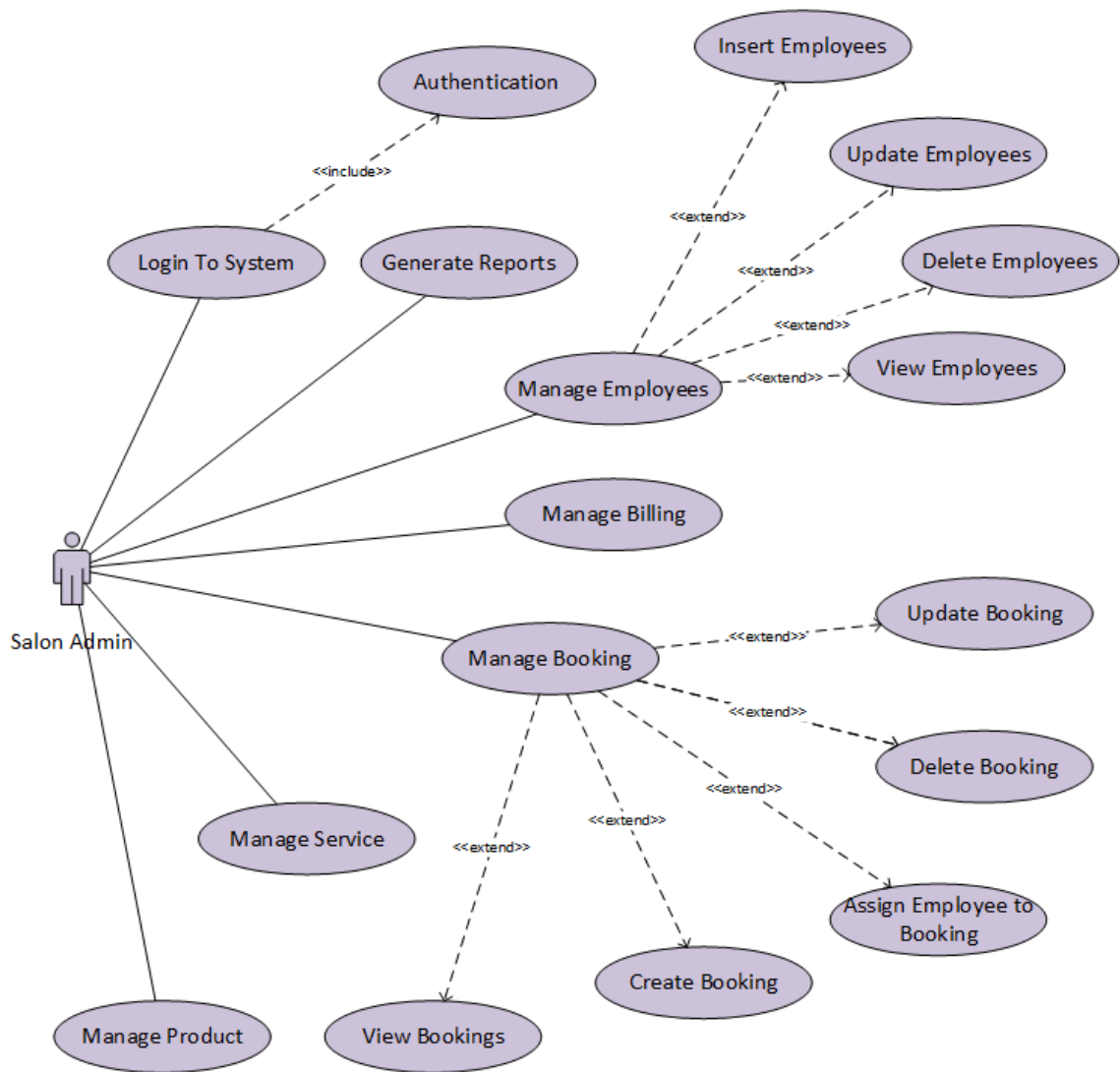


Figure 3.2: Use Case Diagram for Salon Admin

Following Figure 3.3 is for identified use cases for employee in salon. Its express the deferent use cases that employee work on the system.

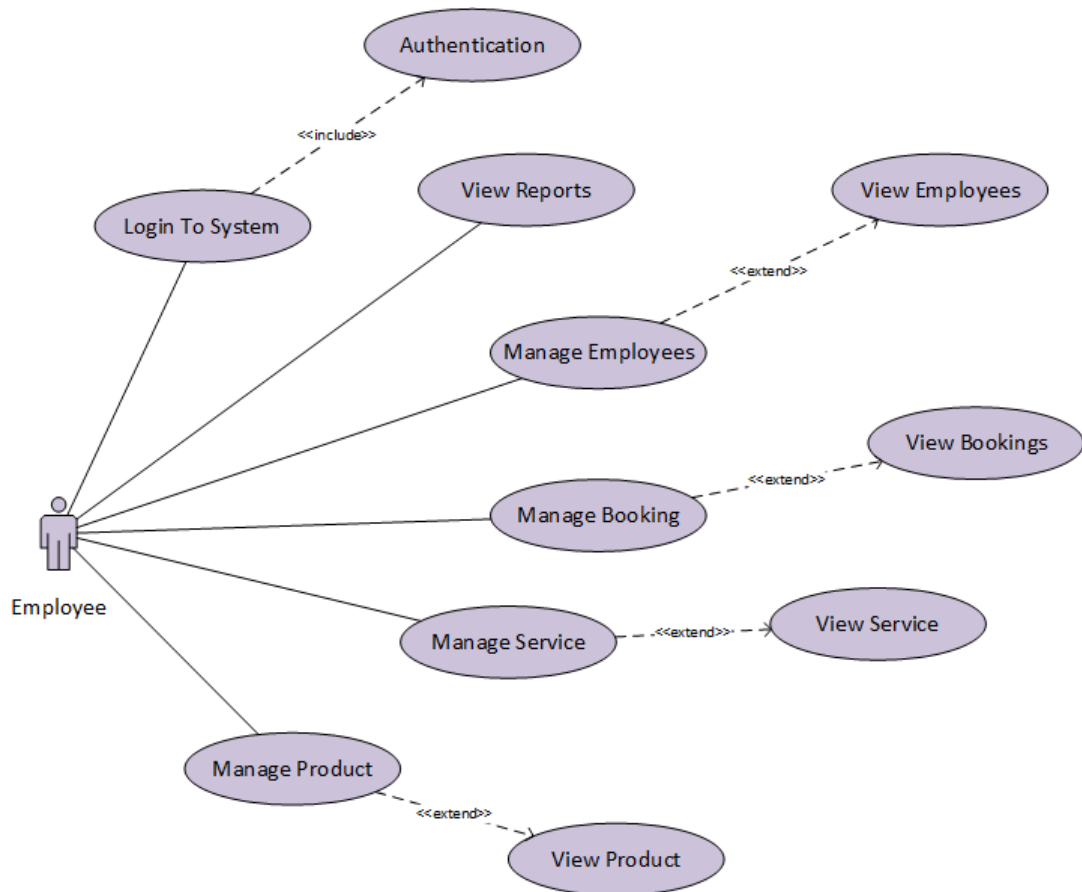


Figure 3.3: Use Case Diagram for Employee

More details about use cases can be described using use case description. From these use case description can get a clear understand about use case diagrams.

A few use case descriptions were presented here to better explain about the use cases of owner use case diagram. These descriptions were contain overview, precondition, flow of events and post conditions.

Bellow table 3.1 is Use case description for the login use case.

Use Case	Login
Actor	All Employees
Overview	Register employees can login to the system
Precondition	Employee must register with the system, Employee should have User Name and Password
Flow of Events	Enter employee Details, Validate employee details, Login to the system
Post Condition	Invalid employee will get the error message and reject login. Valid employee will get the main window of the system

Table 3.1: Use case description for System Login

Table 3.2 is the use case description for the Generate Salon’s Reports use case.

Use Case	Generate Salon’s Reports
Actor	SBP/ Owner/ Salon Admin
Overview	
Generate Salon’s Reports	
Precondition	
Employee should login to the system under authorized user type	
Flow of Events	
Generate salon report using veiled parameters	
Post Condition	
Relevant Report should be show.	

Table 3.2: Use case description for Generate Salon’s Reports

Table 3.3 shows the Use case description for the Create Bookings use case.

Use Case	Create Bookings
Actor	SBP Owner / Salon Admin
Overview	
Create Bookings	
Precondition	
Employee should login to the system under authorized user type	
Flow of Events	
Create booking, open booking module and create booking under specific customer.	
Post Condition	

Table 3.3: Use case description for Create Bookings

Table 3.4 shows the use case description for complete bookings

Use Case	Generate Salon’s Reports
Actor	SBP/ Owner/ Salon Admin
Overview	
Generate Salon’s Reports	
Precondition	
Employee should login to the system under authorized user type	
Flow of Events	
Open booking module. Find booking Complete specific booking	
Post Condition	
Success message should be show. On dashboard complete booking should be identify separately	

Table 3.4: Use case description for complete booking

3.4.2. Class diagram for the system.

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing

the system's classes, their attributes, operations (or methods), and the relationships among objects [6].

Detailed Class diagram for the Samudra Bridal Palace as follows in figure 3.4 will give an idea about the system structure using relationship between system's classes.



Figure 3.4: Class Diagram for New System

3.4.3. Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control [7].

Basically Activity diagram is a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Following Figure 3.4 illustrates the activity diagram for “Login” in the system. It describe the activities of system user and system during the login.

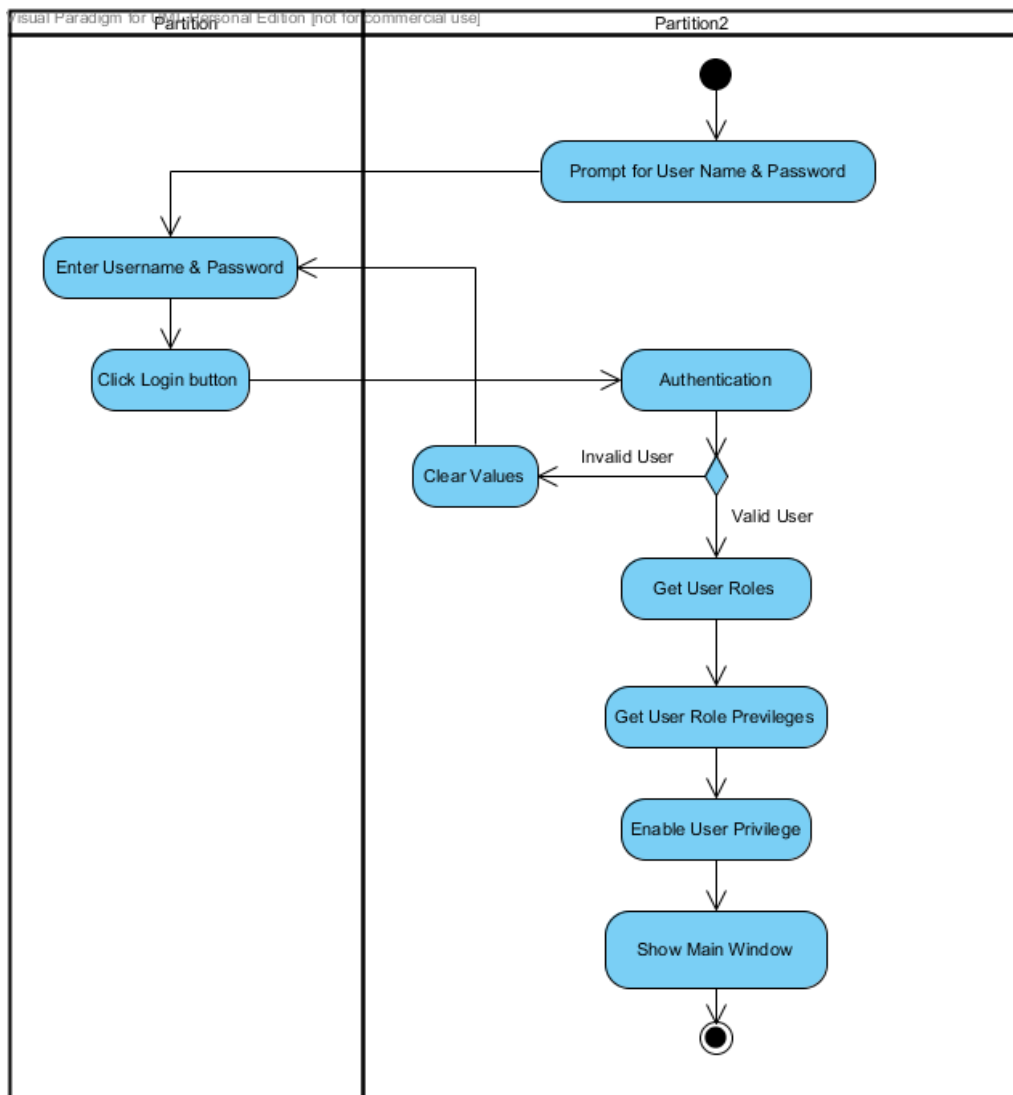


Figure 3.5: Activity Diagram for System Login

Following activity diagram Figure 3.5 shows the “Add New Service” in the system. It describes the activities between user and the system during the insert salon service to the system. In this activity diagram describes the how SBP staff add new service to the system.

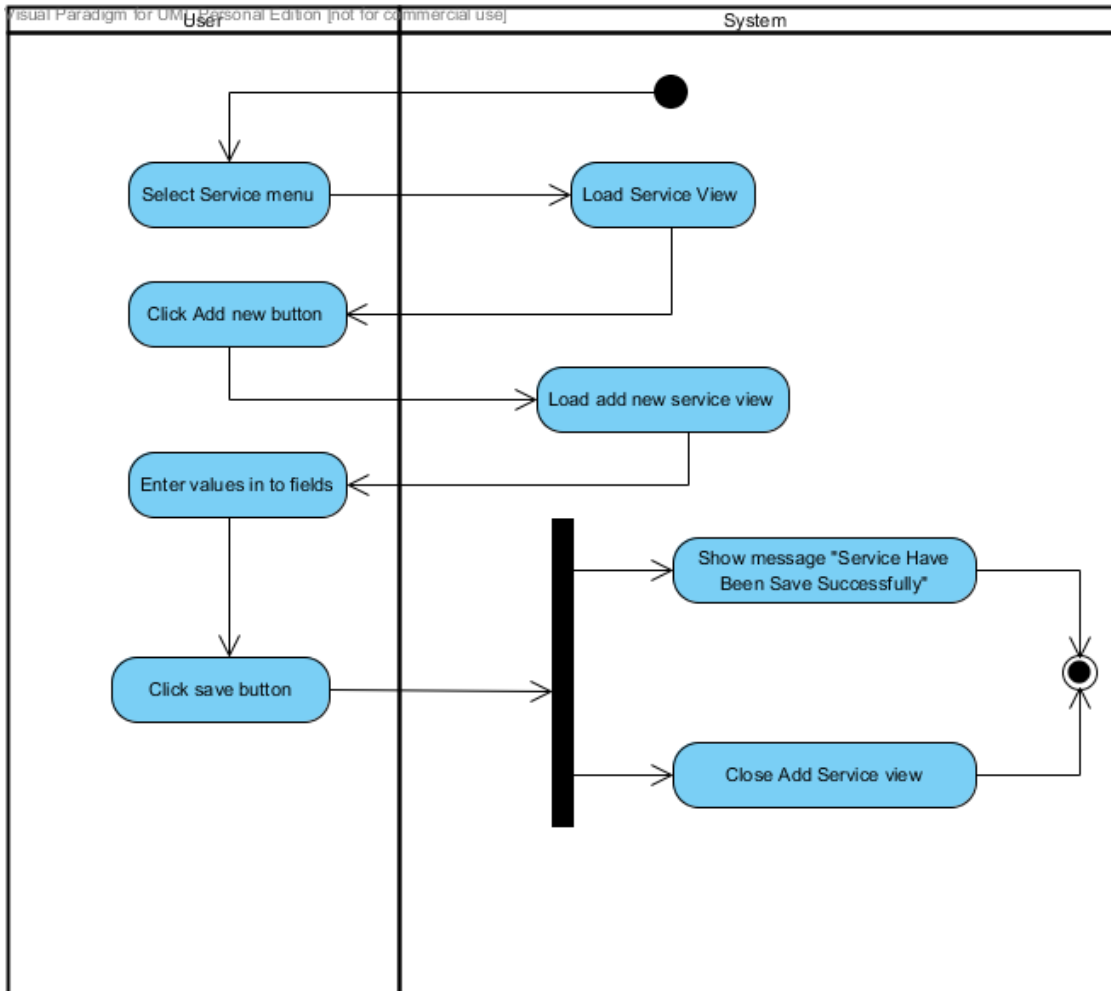


Figure 3.6: Activity Diagram for Add New Service

3.4.4. Sequence Diagram

A sequence diagram is a kind of interaction diagram. It describes the time ordering of the messages between objects in a specific requirement. A sequence diagram shows a set of objects and the messages sent and received by the instance of the objects. We can use a sequence diagram to illustrate the dynamic view of a system.

Following sequence diagram Figure 3.7 illustrate the add booking for customer from booking dashboard.

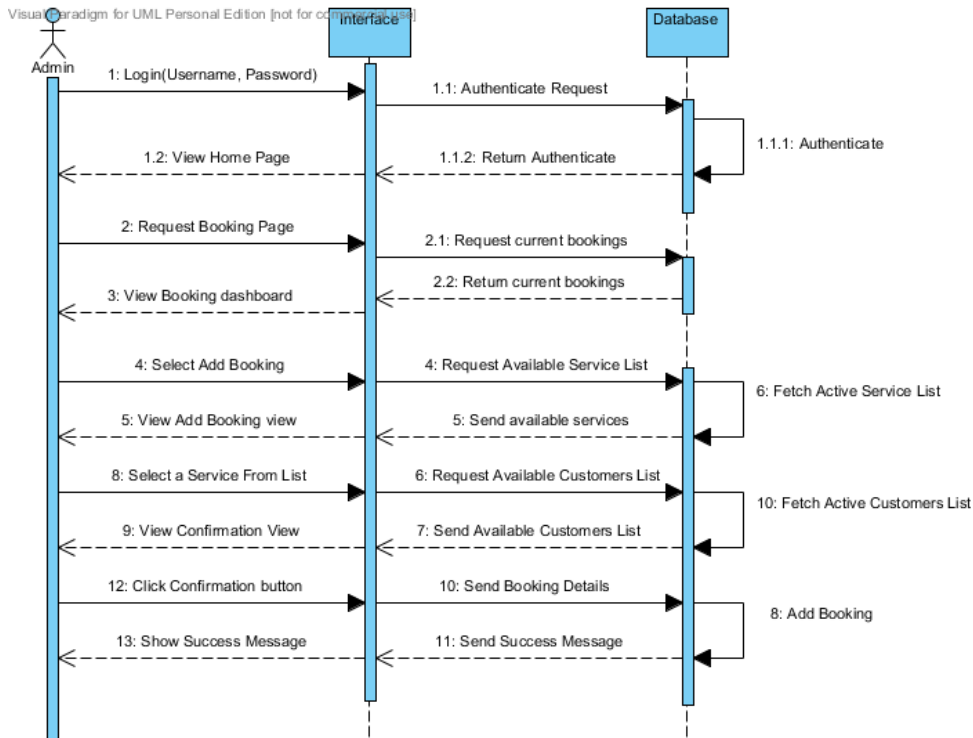


Figure 3.7: Sequence Diagram For Add Booking

Following sequence diagram Figure 3.8 illustrate add services in to system by system administrator.

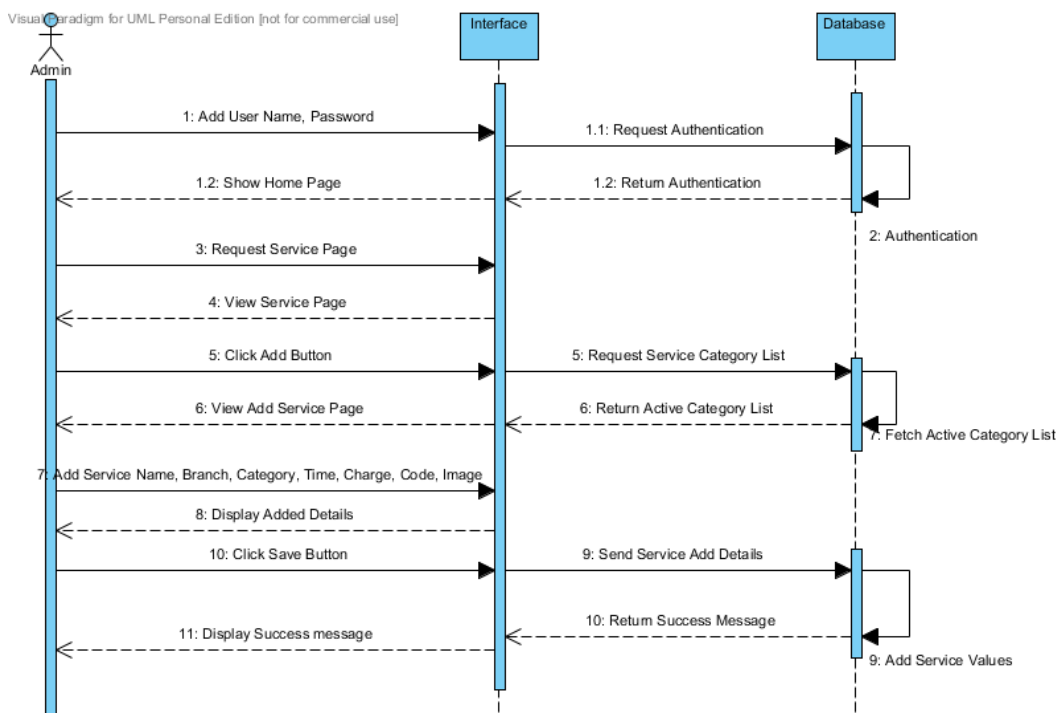


Figure 3.8: Sequence Diagram For Add Service

3.4.5. Database Design And Entity Relationship Diagrams

In software engineering, an entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them [8]. A database is more and more important part of the information system. Database design is achieved through data modelling. This is a technique used for defining business requirements for a database. Following Figure 3.7 is the Entity relationship diagram for the new system.



Figure 3.9: Entity Relationship Diagram for the System

3.4.6. User Interface Design

Among all the major components of the system, a major role of the system is played by user interfaces. Interactivity in between system and the user is managed by the interface. User friendliness, integrated color combination and the well-organized components are dependent on it.

Without having a user friendly interface, interaction with the system becomes hard. Privileges must be set by the System Administrator to users in different ways. Management of those privileges and presenting them effectively to users is helped by the interfaces.

Log in Window

Login form uses to login to the system for authorized users using a valid username and password. If username and password correct user allow to login to the system. Else alert to user that enter the correct password. Figure 3.8 is the preview of login window.



Figure 3.10: Login Window of the System

Main window of the system

Main window of the system provide the tile view for access the modules in system and see overview of the salon day by day process. For each modules it's provides separate tile. User can access these modules by clicking on tiles. Figure 3.9 shows the appearance of main window.

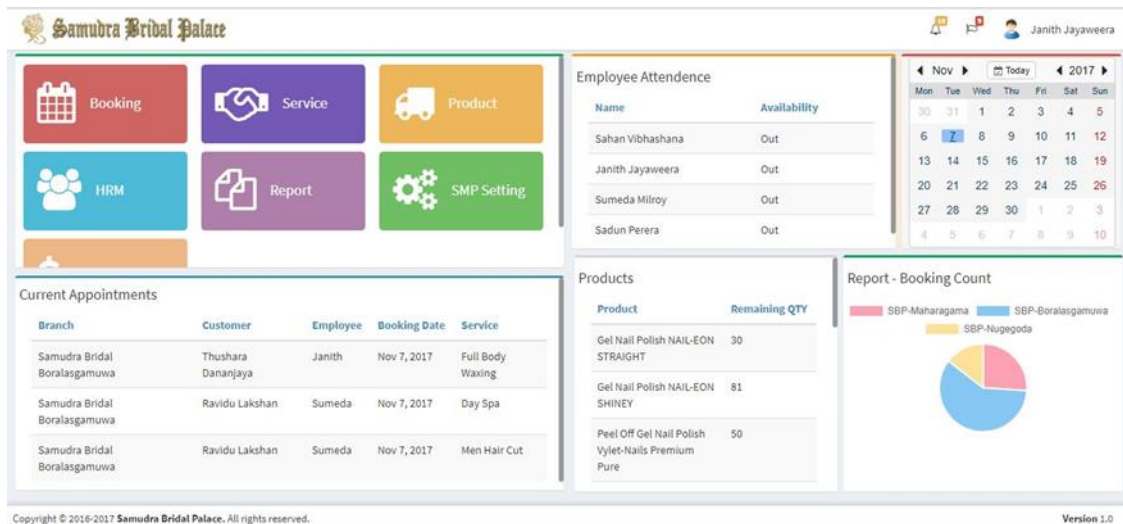


Figure 3.11: Main Window of the System

Service Manage View

Manage service details is one of the client requirement. In system Figure 3.12 preview the service management view in system. User can add, sort by, edit and delete service in system.

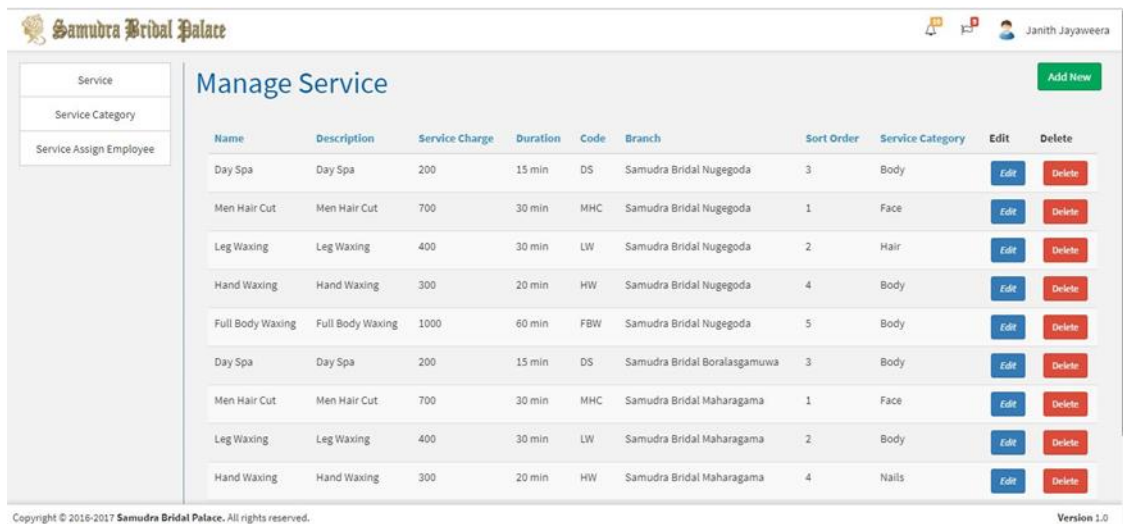


Figure 3.12: Employee Manage Window

CHAPTER 04: IMPLEMENTATION

This chapter is a description of the activities that were carried out during the development of the system. During this phase the results of the design phase were taken and the system was built accordingly. Code modules, interfaces and database tables were created and successfully integrated.

4.1. Hardware and Software Requirements

Hardware and software configuration for the implementation environment is as follows,

- Software Requirements
 - Microsoft Windows 7 or above operating system
 - Microsoft .NET framework 4.5
 - MS SQL 2012
 - IIS (Internet Information Services) 8.5
- Hardware Requirements
 - Intel Core i3 or above Processor
 - 4Gb above Physical Memory for better perform
 - 250GB above HDD

4.2. Development Tools

- Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows superfamily of operating systems, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language,

provided a language-specific service exists. Built-in languages include C, C++ and C++/CLI (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, Java Script and CSS.

Microsoft provides "Express" editions of its Visual Studio at no cost. Commercial versions of Visual Studio along with select past versions are available for free to students via Microsoft's DreamSpark program.

- Visual Code

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open-source, although the official download is under a proprietary license.

Visual Studio Code is based on Electron, a framework which is used to deploy Node.js applications for the desktop running on the Blink layout engine. Although it uses the Electron framework, the software does not use Atom and instead employs the same editor component (code-named "Monaco") used in Visual Studio Team Services (formerly called Visual Studio Online). https://en.wikipedia.org/wiki/Visual_Studio_Code

- C# Language

C#(pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure. C# is intended to be a simple, modern, general-purpose, object-oriented programming

language. Its development team is led by Anders Hejlsberg. The most recent version is C# 7.0, which was released in 2017 along with Visual Studio 2017.

- Angular 2

Angular JS is an open source framework built over JavaScript. It was built by the developers at Google. This framework was used to overcome obstacles encountered while working with Single Page applications. Also, testing was considered as a key aspect while building the framework. It was ensured that the framework could be easily tested. The initial release of the framework was in October 2010. Angular 2.0 was officially announced at the ng-conference in October, 2014.

Following are the key features of Angular 2 –

- Components – The earlier version of Angular had a focus of Controllers but now has changed the focus to having components over controllers. Components help to build the applications into many modules. This helps in better maintaining the application over a period of time.
- TypeScript – The newer version of Angular is based on TypeScript. This is a superset of JavaScript and is maintained by Microsoft.
- Services – Services are a set of code that can be shared by different components of an application. So for example if you had a data component that picked data from a database, you could have it as a shared service that could be used across multiple applications.

In addition, Angular 2 has better event-handling capabilities, powerful templates, and better support for mobile devices [10].

- SQL Server 2012

Microsoft SQL Server 2012 is a relational database management system (RDBMS) designed for the enterprise environment. Like its predecessors, SQL Server 2012 comprises a set of programming extensions to enhance the Structured Query Language (SQL), a standard interactive and programming language for getting information from and updating a database [12].

Microsoft SQL Server 2012, which supplants SQL Server 2008 R2, offers new capabilities, notable among them the following.

- Column Store indexes: Read-only indexes that group data, streamlining the processing of large data warehouse queries.
- Support for Windows Server Core: This is a stripped-down version that places a far lower demand on computer resources than a full install does.
- Power View: Makes it possible to generate mash-ups of business intelligence (BI) reports.
- Enhanced Auditing: Users can customize their audit logs to accommodate a wider range of events with greater flexibility than was previously possible.
- Always On: Users can fail over multiple databases and read secondary copies, enhancing disaster recovery (DR) operations.
- Distributed Replay: A workload can be taken from a production server and played on another server to test it under realistic conditions before deploying it.

4.3. Code Feature

The main code modules developed in the system have been mentioned below by briefly describing their functionality. Code modules consist with comments to identify the specific reason of a particular code line.

1. Design Architecture

Project structure built according to the N-Layered architecture. Figure 4.1 is illustrating the Solution structure for projects in Visual Studio. In the project list,

- Presentation Layer for front end (angular 2 project).
- WebAPI Layer for handle API calls
- Application Layer
- Business Layer for handle business related functions
- Domain Layer for handle domain objects
- Infrastructure Layer for handle and calls stored procedure
- SP_DataAccess Layer for handle common db configuration

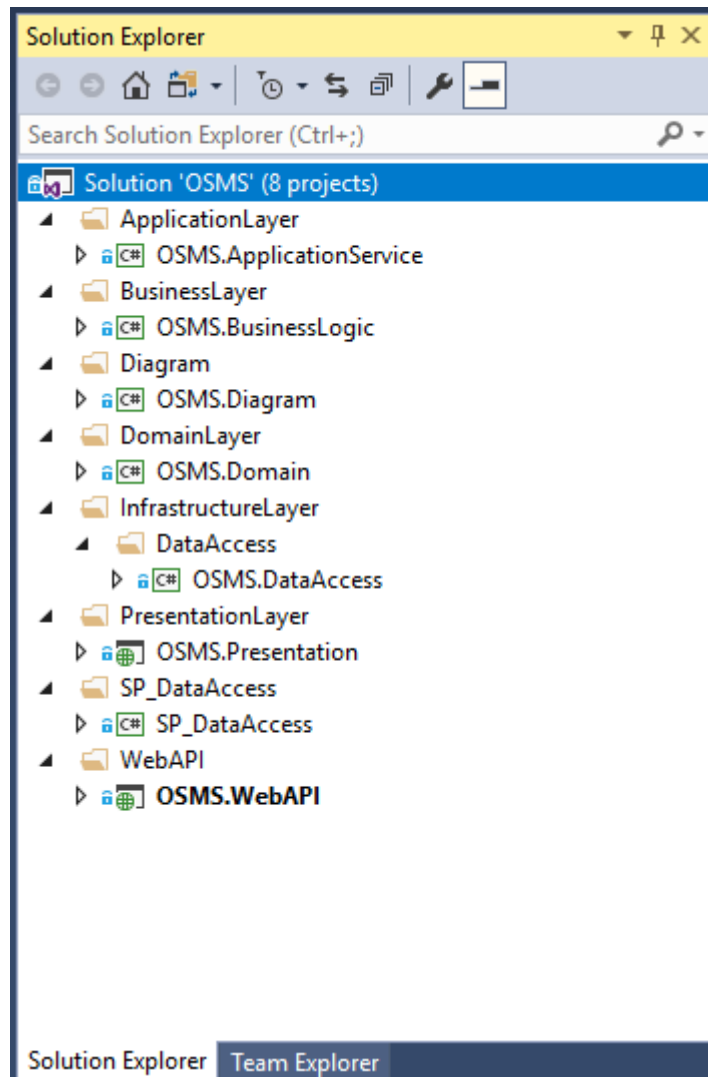


Figure 4.1: Solution Structure for Projects in Visual Studio

Major Code Segments

1. Database Connection

```
<configuration>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data
Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=&quot;|DataDirectory|\aspnet-OTEM
WebAPI2-20170530075704.mdf&quot;;Initial Catalog=&quot;aspnet-OTEM WebAPI2-
20170530075704&quot;;Integrated Security=True"
providerName="System.Data.SqlClient"/>
    <!-- Connection string for connect to local database -->
    <add name="OSMSContext" connectionString="Data Source=localhost;Initial
Catalog=OSMS;Integrated Security=True;" providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <appSettings>
    <!-- Path to save images -->
    <add key="ResourceFileLocation" value="E:\BIT\Project\OSMS.WebAPI\Images\"/>
    <!-- Web api url -->
    <add key="ResourceServerUrl" value="http://localhost:51502"/>

    <!-- Email configurations -->
    <add key="MailServer" value="smtp.gmail.com" />
    <add key="MailServerPort" value="25" />
    <add key="MailAddress" value="giheshaadithya@gmail.com" />
    <add key="MailPwd" value="1qaz2wsx@" />
    <add key="EnableSsl" value="true" />

  </appSettings>
  <system.web>
    <authentication mode="None"/>
    <compilation debug="true" targetFramework="4.5"/>
    <httpRuntime targetFramework="4.5"/>
    <httpModules>
      <add name="ApplicationInsightsWebTracking"
type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule,
Microsoft.AI.Web"/>
    </httpModules>
  </system.web>
</configuration>
```

2. Service Page Add Model

Service add view

Figure 4.2: Service add model

Code Level

```

<div class="row">
  <modal #addServiceModal (onClose)="resetServiceModal()">r
    <modal-header>
      <h4 class="modal-title">Add Service</h4>
    </modal-header>
    <modal-content>
      <form [formGroup]="serviceForm" (ngSubmit)="submitService($event)">
        <div class="row">
          <div class="col-md-4">
            <div class="form-group">
              <label>Service Name</label>
              <input type="text" class="form-control" formControlName="Name">
              <!-- Validation -->
              <div *ngIf="serviceForm.controls['Name'].hasError('required') &&
serviceForm.controls['Name'].touched" class="input-error">Please enter Service Name</div>
            </div>
          </div>
          <div class="col-md-4">
            <div class="form-group">
              <label>Select Branch</label>

```

```

        <select class="form-control"
(change)="onChangeBranch($event.target.value)" formControlName="BranchId">
            <option *ngFor="let bl of branchList" [ngValue]="bl.Id"> {{bl.Name}}
</option>
        </select>
    </div>
</div>
<div class="col-md-4">
    <label>Service Category</label>
    <select class="form-control" formControlName="ServiceCategory">
        <option *ngFor="let sc of serciveCategoryList" [ngValue]="sc.Id">
{{sc.Name}} </option>
    </select>
    <!-- Validation -->
    <div *ngIf="serviceForm.controls['ServiceCategory'].hasError('required') &&
serviceForm.controls['ServiceCategory'].touched"
        class="input-error">Please select Service Sub Category</div>
    </div>
</div>
<div class="row">
    <div class="col-md-4">
        <div class="form-group">
            <label>Service Time(Min)</label>
            <input type="text" class="form-control" formControlName="Duration">
            <!-- Validation -->
            <div *ngIf="serviceForm.controls['Duration'].hasError('required') &&
serviceForm.controls['Duration'].touched" class="input-error">Please enter Service
Duration</div>
        </div>
    </div>
    <div class="col-md-4">
        <div class="form-group">
            <label>Service Code</label>
            <input type="text" class="form-control" formControlName="Code">
        </div>
    </div>
    <div class="col-md-4">
        <div class="form-group">
            <label>Sort Order</label>
            <input type="number" class="form-control"
formControlName="SortOrder">
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-4">
        <div class="form-group">
            <label>Service Charge</label>
            <input type="number" class="form-control"
formControlName="ServiceCharge">
            <!-- Validation -->
            <div *ngIf="serviceForm.controls['ServiceCharge'].hasError('required') &&
serviceForm.controls['ServiceCharge'].touched"
                class="input-error">Please enter Service Charge</div>
        </div>
    </div>

```

```

</div>
<!-- Image Upload Section -->
<div class="col-md-4">
  <div class="form-group">
    <label>Service Image</label>
    <div class="input-group file-upload">
      <input type="text" class="form-control" id="service-img"
placeholder="Select a file...">
      <div class="input-group-addon">
        <span>Browse</span>
        <input class="" type="file" (change)="onChange($event)" size="60"/>
      </div>
    </div>
  </div>
</div>
</div>
<!-- End Image Upload Section -->
</div>
<div class="">
  <div class="form-group">
    <label>Description</label>
    <textarea class="form-control" rows="5" id="comment"
formControlName="Description"></textarea>
  </div>
</div>
</form>
</modal-content>
<modal-footer>
  <button type="button" class="btn btn-primary pull-right" data-dismiss="modal"
#closeServiceModal (click)="addServiceModal.close()">close</button>
  <button type="submit" [disabled]="!serviceForm.valid" class="btn btn-primary pull-
right">Save</button>
</modal-footer>
</modal>
</div>

```

3. Service Component Save Function

```

/**
 * Save Service data to the database after form validation
 * @param {object} service model
 * @return {int} service id
 */
submitServiceData() {
  // Check service form validation
  if (this.serviceForm.dirty && this.serviceForm.valid) {
    // Variable to hold a reference of addComment/updateComment
    let serviceOperation: any;
    // Check service add or edit
    if (this.serviceModel.Id == undefined) {
      this.editing = false;

```

```

}
serviceOperation = this._salonServiceService.insertOrUpdateService(this.serviceModel)
// Subscribe to observable
serviceOperation.subscribe(
  (service: any) => {
    this.uploadImage(service.Id);
    if (this.editing) {
      this.footerNotification.success('Update', 'Successfully saved');
    } else {
      this.footerNotification.success('Save', 'Successfully saved');
    }
    this.closeServiceModal.nativeElement.click();
    this.loadServiceList();
    if (this.editing) this.editing = !this.editing;
  },
  (err: any) => {
    // Log errors if any
    console.log(err);
  });
} else if (!this.serviceForm.dirty) {
  this.footerNotification.info('Info', 'No changes detected ');
}
}
}

```

4. Service Layer for add new salon service

```

/**
 * save service details
 * @param {object} service object
 */
insertOrUpdateService(obj: Object) {
  let url = this.commonService._serviceUrl + '/' + 'api/Service/InsertOrUpdateService';
  // Convert json object to C# object
  let params = this.commonService.convertObject(obj);
  return this._http.post(url, params.toString(), this.options)
    .map((res: Response) => {
      // Success on post request.
      return res;
    })
    .catch((error: any) => {
      // Error on post request.
      return Observable.throw(error);
    });
}
}

```

5. Web API Layer (Application programming interface)

```
#region Service API Commands

/// <summary>
/// Pass Service data to Service API
/// </summary>
/// <param name="service"></param>
/// <returns>Service Id</returns>
[HttpPost]
public int InsertOrUpdateService(Service service)
{
    return ServiceAPI.InsertOrUpdateService(service);
}
```

6. Service API

```
/// <summary>
/// Pass Service data to Service Business Layer
/// </summary>
/// <param name="service"></param>
public static int InsertOrUpdateService(Service service)
{
    return ServiceBL.InsertOrUpdateService(service);
}
```

7. Service Business Layer

```
/// <summary>
/// Pass Service data to Service Action
/// </summary>
/// <param name="service"></param>
public static int InsertOrUpdateService(Service service)
{
    return new InsertOrUpdateServiceAction(service).Execute(EnumDatabase.OSMS);
}
```

8. Data Access layer

```
using OSMS.Domain.DomainObject;
using SP_DataAccess.DbFramework;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Common;
using System.Data.SqlClient;
```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OSMS.DataAccess.SpDbCommands.ServiceSpCommands
{
    public class InsertOrUpdateServiceAction : OSMSDbActionBase<int>
    {
        private readonly Service _service;

        public InsertOrUpdateServiceAction(Service service)
        {
            _service = service;
        }

        /// <summary>
        /// Get Database connection and execute stored procedure
        /// </summary>
        /// <param name="db connection"></param>
        /// <returns>Service id</returns>
        protected override int Body(DbConnection connection)
        {
            int serviceCatId;
            try
            {
                const string storedProcedureName = "dbo.OSMS_SVC_InsertOrUpdateService";
                var cmd = CreateCommand(CommandType.StoredProcedure,
storedProcedureName);

                // Add Service data to stored procedure
                cmd.Parameters.Add(new SqlParameter("@id", _service.Id));
                cmd.Parameters.Add(new SqlParameter("@name", _service.Name));
                cmd.Parameters.Add(new SqlParameter("@description", _service.Description));
                cmd.Parameters.Add(new SqlParameter("@imagePath", _service.ImagePath));
                cmd.Parameters.Add(new SqlParameter("@duration", _service.Duration));
                cmd.Parameters.Add(new SqlParameter("@serviceCharge",
_service.ServiceCharge));
                cmd.Parameters.Add(new SqlParameter("@branchId", _service.BranchId));
                cmd.Parameters.Add(new SqlParameter("@code", _service.Code));
                cmd.Parameters.Add(new SqlParameter("@sortOrder", _service.SortOrder));
                cmd.Parameters.Add(new SqlParameter("@serviceCategory",
_service.ServiceCategory));
                cmd.Parameters.Add(new SqlParameter("@isDelete", _service.IsDelete));

                //Set out put Parameter
                DbParameter outputParam = new SqlParameter();
                outputParam.DbType = DbType.Int32;
                outputParam.ParameterName = "@serviceId";
                outputParam.Direction = ParameterDirection.Output;
                cmd.Parameters.Add(outputParam);
                cmd.ExecuteNonQuery();
                cmd.Dispose();

                serviceCatId = outputParam.Value == null ? -1 :
Convert.ToInt32(outputParam.Value);
            }
            catch { }
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        throw ex;
    }
    return serviceCatId;
}
}
}

```

9. Stored Procedure

```

USE [OSMS]
GO
/***** Object: StoredProcedure [dbo].[OSMS_SVC_InsertOrUpdateService]  Script Date:
2017-11-07 9:00:16 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Janith Jayaweera
-- Create date: 2017/07/03
-- Description:  Insert or update service details
-- =====
ALTER PROCEDURE [dbo].[OSMS_SVC_InsertOrUpdateService]
    -- Add the parameters for the stored procedure here
    @id int,
    @name varchar(200),
    @description varchar(1000) = null,
    @isDelete bit,
    @imagePath varchar(500) = null,
    @duration int = null,
    @serviceCharge varchar(500) = null,
    @code varchar(500) = null,
    @sortOrder int = null,
    @serviceCategory int = null,
    @branchId int = null,
    @serviceId int output
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here

    IF @id = 0 OR @id IS NULL
        BEGIN
            INSERT INTO
                [dbo].[Service]

```

```

        (Name,
        Description ,
        IsDelete,
        ImagePath,
        Duration,
        ServiceCharge,
        Code,
        SortOrder,
        ServiceCategory,
        BranchId
        )
VALUES
    (@name,
    @description,
    @isDelete,
    @imagePath,
    @duration,
    @serviceCharge,
    @code,
    @sortOrder,
    @serviceCategory,
    @branchId
    )

SET @serviceId = @@IDENTITY
END
ELSE
-- Update statements for procedure here
BEGIN
    UPDATE [dbo].[Service]
    SET
        Name = @name,
        Description = @description,
        IsDelete = @isDelete,
        ImagePath = @imagePath,
        Duration = @duration,
        ServiceCharge = @serviceCharge,
        Code = @code,
        SortOrder = @sortOrder,
        ServiceCategory = @serviceCategory,
        BranchId = @branchId
    WHERE
        Id = @id

    SET @serviceId = @id
END
END

```


4.4. Re-usable Components

The following re-usable components have been used when implementing the system to add more attractiveness and to maximize the efficiency of the system.

- **Angular 2 Date Picker - mydatepicker**

In the system Angular 2 Date Picker – mydatepicker was used as date pickers in forms for insert dates. [16]

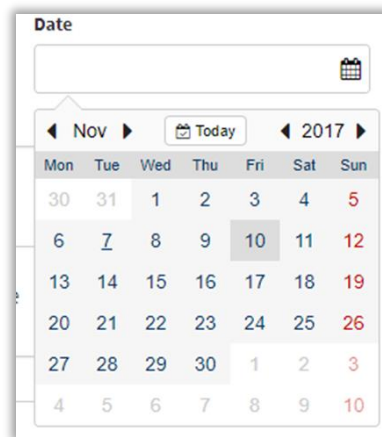


Figure 4.3: Angular 2 Date Picker View

- **JS PDF & JS PDF Auto table**

For the reporting module JS PDF & JS PDF Auto table library was used to generate report and print reports. [17]

- **Day Pilot Pro Angular**

Day Pilot Pro Angular was use for booking dashboard [18]

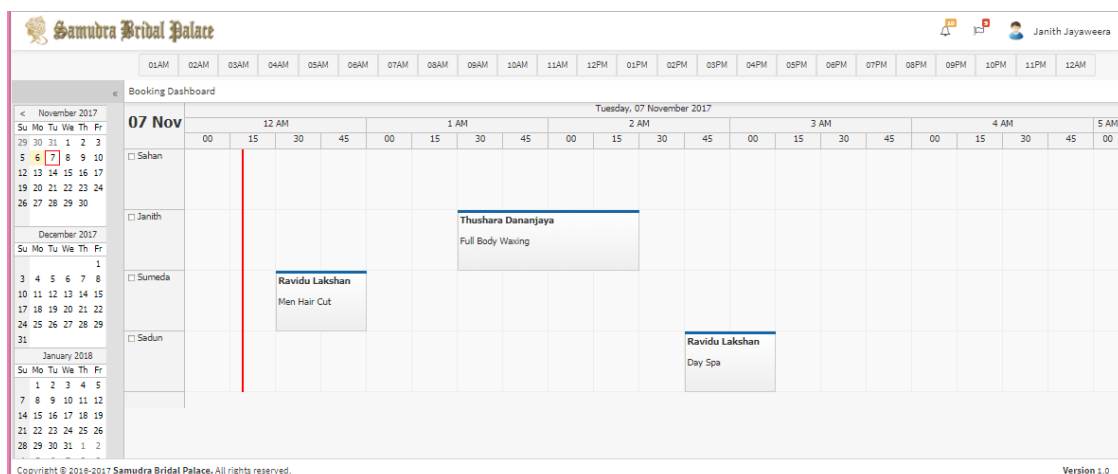


Figure 4.4: Day Pilot Pro Angular view

- **ng 2 Model**

On the system ng2 model used for popups.

CHAPTER 05: EVALUATION

After the successful implementation of a system for its accuracy the system is needed to be verified and validated. Software testing is performed to verify, that the completed software functions according to the expectations defined by the requirements. In this chapter describes about the test plan used in the development of the system, and sample of a test result is also provided as evidence that the specifications have been met.

5.1. Requirement for A Good System Evaluation

Before the system developed in to business environment it must test in various ways to ensure system not gain any type of error. System testing is performed on the entire system in the context of a Functional Requirement Specification(s) and/or a System Requirement Specification. System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

These tests were helped to check,

- The functions developed are according to the specification.
- The identification and reporting of error that occur in the system and correcting them as needed.
- Improving the performance and efficiency of the system.

5.2. Test Plan

The test plan is describes the testing strategies and the approaches to testing quality assurance will be used to validate the quality of the system. While the coding system, it was carried out the unit testing to ensure that functionalities are working as expected. This was done by testing the code unit wise.

An integration testing has been done for the test modules in system. It was integration of few units of codes. And all the modules together tested as system testing to confirm all the modules work together. And also as the system users are

able to access via any browsers need to be test at least most common web browsers like Chrome, Firefox and Edge etc.

5.3. Test Strategies

5.3.1. Unit testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended [19].

5.3.2. Black box testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher-level testing, but can also dominate unit testing as well [20].

5.3.3. White box testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as

programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT) [21].

5.3.4. Integration testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing [22].

5.3.5. System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole [23].

5.4. Test Case Generation

In test plan most, important part is test cases writing. Most commonly test cases include the test case title, inputs, expected results, test case status (Pass/Fail). To minimized complexity of testing total solution tested by module wise. Test Cases for the “Test” Module in OSMS

5.4.1. Test Case and Test Result for Service Category

In the table 5.1 it's showing the test cases and test result for service category. Mainly adding service category, editing service category, deleting service category and validation tested on here.

ID	Description/steps to test	Expected Result/s	Actual Result/s	Status
1	Click Service Category on left navigation in Service Module	Service category view should be loaded in to service window	See Figure 5.1	Pass
2	Add service category Values: Category Name- Face Description - Facial treatments Branch - Boralasgamuwa Click save	1-Should be save to database table 2-Text fields should be cleared. 3-Add Category view should be get closed. 4-New row with the entered values should be added in to table on view.	1-Record added to the database table 2-Text fields get cleared 3- View get closed 4-New record added to the table on view	Pass
3	Add new service category- Click save with empty text fields	1-Red color border should be appear around the required fields & shown error massage.	See Figure 5.2	Pass
4	Add new category Values: “Name, Branch” field and click save	1-Should be save to database table 2-text fields should be clear. 3-Table in view should be added new row with entered values.	1-Record added to the database table 2-Text fields got cleared 3-record added to the db	Pass
5	Add new test category- Values: only to “Description” text field and click save	1-Fields validations should be fired 2-Data should not be insert in to database table	Red color border appear around text box validation messages shown	Pass
6	Delete service category- Click Delete button in table	1-Delete confirmation message should be shown to the users 2- If click yes - selected category should be removed from view 3- If click no – selected category should not be removed from view 4- deleted record should not be deleted from database table, status value should set 1 in database	1- See Figure 5.3 2-Deleted row removed from table in view 3- on data base record update to 1 in status column. Figure 5.4	Pass
7	Edit service category- Click Edit button in table	1-selected service category's values should be fills into service category form.	Selected service category loaded to the form.	Pass

8	Save changes made on service category 1-Click Edit Change values in fields 2- Click save	1-database table should be updated as changes done in fields. 2- Edit view should be closed 3-Table in view should be refresh with new values.	On the DB its changes applied. Edit view get closed. Changes reflected on view	Pass
---	--	--	--	------

Table 5.1: Test Cases for Service category

Bellow Figure 5.1, 5.2, 5.3, 5.4 images are refers to “Actual Result” column in table 5.1 Test Case for Service Category.

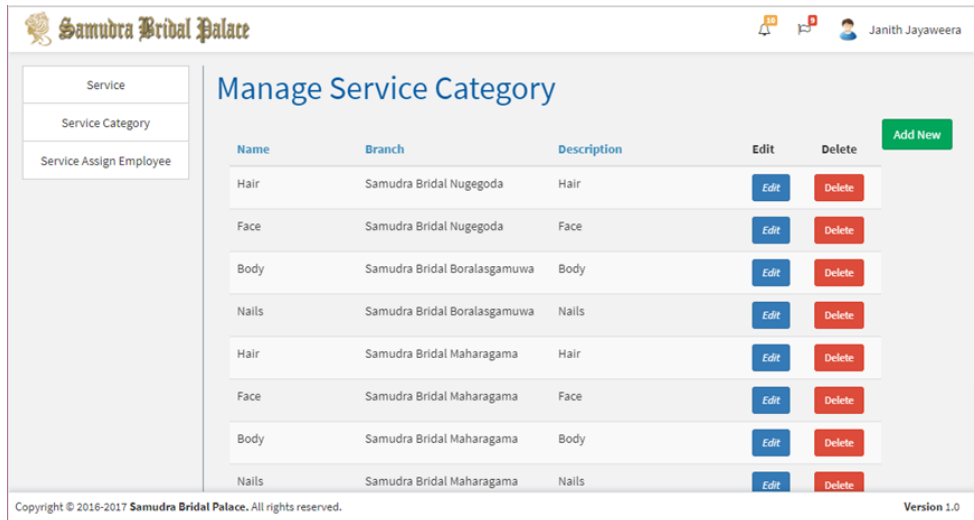


Figure 5.1: Service Category View

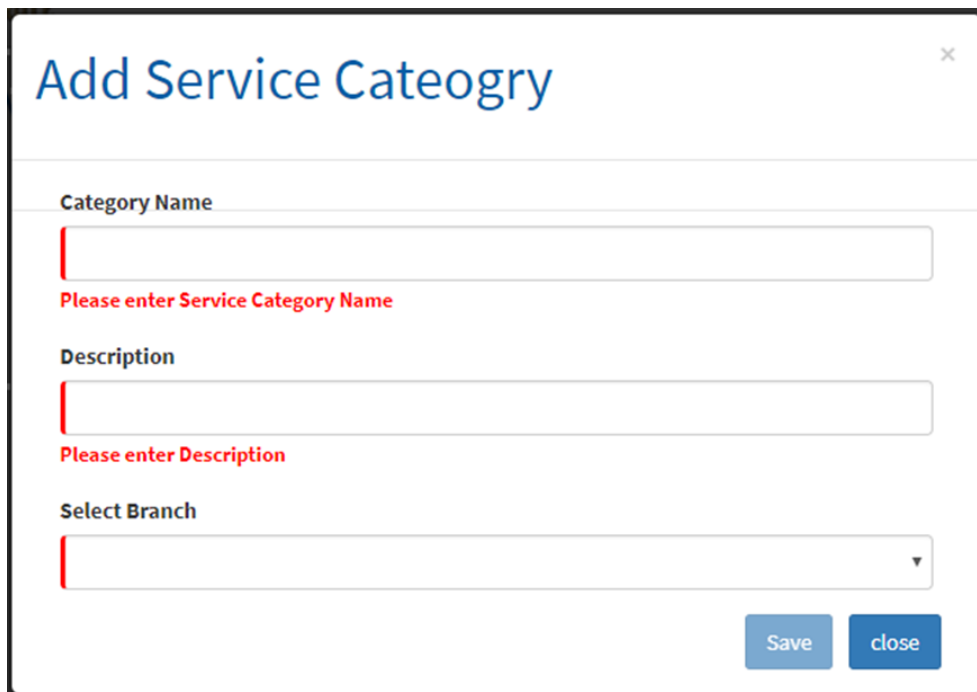


Figure 5.2: Validation for Required Fields

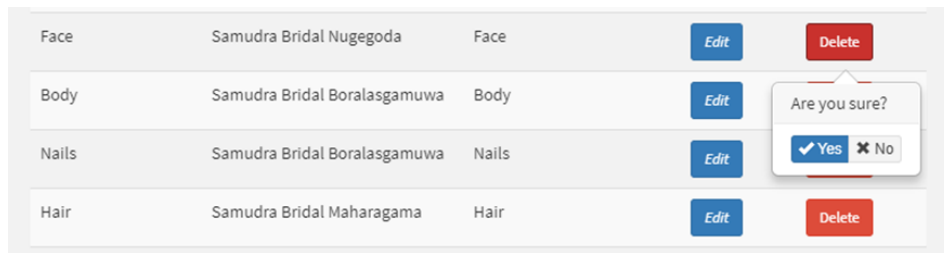


Figure 5.3: Delete Confirmation Message

	Id	Name	Description	IsDelete	BranchId
	32	Hair	Hair	False	6
▶	33	Face	Face	True	6
	35	Body	Body	False	4
	55	Nails	Nails	False	4
	57	Hair	Hair	False	2
	58	Face	Face	False	2
	59	Body	Body	False	2
	60	Nails	Nails	False	2
	61	Hair	Hair	False	4
	62	Face	Face	False	4
	63	Body	Body	False	6
	64	Nails	Nails	False	6
*	NULL	NULL	NULL	NULL	NULL

Figure 5.4: Status Change to “True” for Deleted Records

5.1.1. Test Cases and Test Result for Service

In the table 5.2 it’s showing the test cases and test result for the adding service, editing service, deleting service and validation tested on here.

ID	Description	Expected result	Actual Result/s	Status
1	Click Service on left navigation	Service View should be load into main window	See figure 5.5	Pass
2	Add new service - Service Name: Eye Brow Service Cat: Face Description: Eye Brow Duration: 15 minutes Price: 200.00	1-Should be save in database table 2- Add Service view should be get closed. 3-Table in view should added new row with entered values.	1-Record added to the database table 2-Add view get closed. 3-New record shown on the service view table	Pass
3	Add new service Click save with empty text fields for service name	1-Fields validations should be shown “Please Enter Service Name” for Service name	See Figure 5.6	Pass
4	Add new service Values: Only for Service Name	1-Fields validations should be shown required for Service Category, duration and price	1-Vlaidation tool tip shown when click on the text box.	Pass

		2-Should not save to database table	2.data not insert to the database table	
5	Add new service Values: Service Name and Service Category	1-Fields validations should be shown for Price and Duration 2-Should not save to database table	1-Shown the tool tip message for price text box. 2- Not save to the DB	Pass
6	Add new service Values: Service Name, Service Category, Duration, Description	1-Fields validations should be shown "Is required" Price 2-Should not save to database table	1-Shown the tool tip message for price text box. 2- Not save to the DB	Pass
7	Click Delete in data grid	1-Should prompt for user to confirm	See Figure 5.7	Pass
8	Click Delete in data grid Confirm as yes	1-Selected Service's isDelete attribute value should update as "1" in database 2-Table in view should refresh and deleted row should be remove	1-See Figure 5.8 2- deleted record removed from the table in view	Pass
9	Click Delete in data grid Confirm as No	1-Confirmation message should be closed 2-No changes in data base or table in view	1-Confirmation message closed 2-record not removed from table in view	Pass

Table 5.2: Test Cases for Service

Bellow Figure 5.5, 5.6, 5.7, 5.8 images are refers to "Actual Result" column in Table 5.2: Test Case for Service.

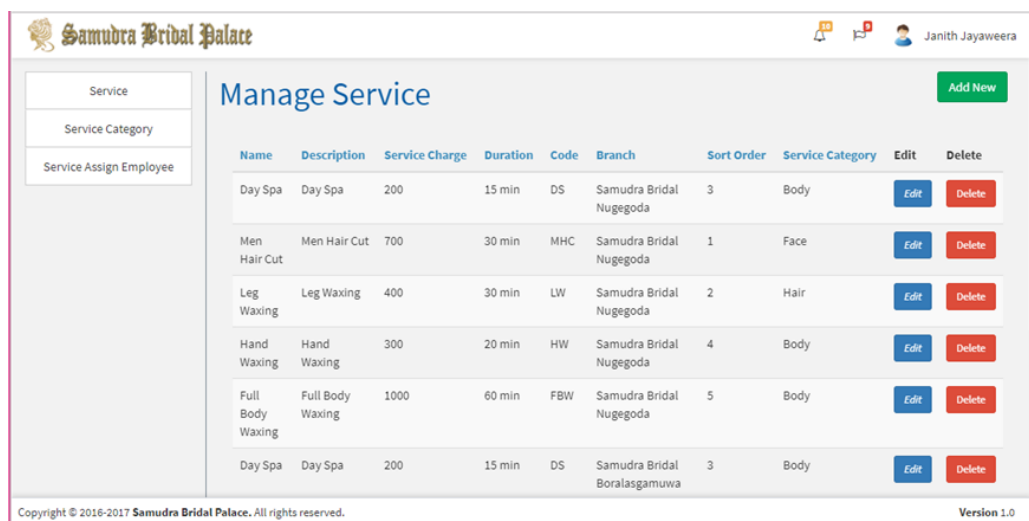


Figure 5.5: Service View

ADD SERVICE ✕

Service Name

Please enter Service Name

Select Branch

▼

Service Category

▼

Service Time(Min)

Please enter Service Duration

Service Code

Sort Order

Service Charge

Please enter Service Charge

Service Image

Select a file..
Browse

Description

Save
close

Figure 5.6: Validation for required fields in Add Service view

Name	Description	Service Charge	Duration	Code	Branch	Sort Order	Service Category	Edit	Delete
Day Spa	Day Spa	200	15 min	DS	Samudra Bridal Nugegoda	3	Body	Edit	Delete
Men Hair Cut	Men Hair Cut	700	30 min	MHC	Samudra Bridal Nugegoda	1	Face	Edit	<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> Are you sure? <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No </div>
Leg Waxing	Leg Waxing	400	30 min	LW	Samudra Bridal Nugegoda	2	Hair	Edit	Delete

Figure 5.7: Service delete confirmation message

Id	Name	Description	ImagePath	Duration	ServiceCharge	IsDelete	BranchId	Code	SortOrder	ServiceCategory
1047	Day Spa	Day Spa	NULL	15	200	False	6	DS	3	59
1055	Men Hair Cut	Men Hair Cut	NULL	30	700	True	6	MHC	1	33
1056	Leg Waxing	Leg Waxing	NULL	30	400	False	6	LW	2	32
1057	Hand Waxing	Hand Waxing	NULL	20	300	False	6	HW	4	35
1058	Full Body Waxing	Full Body Waxing	NULL	60	1000	False	6	FBW	5	35
1059	Day Spa	Day Spa	NULL	15	200	False	4	DS	3	35
1060	Men Hair Cut	Men Hair Cut	NULL	30	700	False	2	MHC	1	58
1061	Leg Waxing	Leg Waxing	NULL	30	400	False	2	LW	2	59
1062	Hand Waxing	Hand Waxing	NULL	20	300	False	2	HW	4	60
1063	Full Body Waxing	Full Body Waxing	NULL	60	1000	False	2	FBW	5	60
1064	Day Spa 3	Day Spa	NULL	15	200	False	4	DS	3	61
1065	Men Hair Cut	Men Hair Cut	NULL	30	700	False	4	MHC	1	61
1066	Leg Waxing	Leg Waxing	NULL	30	400	False	4	LW	2	62
1067	Hand Waxing	Hand Waxing	NULL	20	300	False	4	HW	4	62
1068	Full Body Waxing	Full Body Waxing	NULL	60	1000	False	4	FBW	5	61
1069	dsfds	33	NULL	45	400	False	2	DD	3	59
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 5.8: In DB service table status changed to 1

5.5. User Evaluation

After implemented the system successfully it was tested by the client side. The user acceptance testing was carried out by implementing the system at the real working environment along with the real test data & available conditions in the actual background.

For the evaluation, system users were selected and they were given an evaluation form to answer some questions about the system. All the users were asked to perform their appropriate functionalities and test the system. Finally, their feedback was collected and analyzed. The final overall results were indicate that system work properly and they were satisfied about the system. According to the test results it can manage easily and supply user-friendly environment for the users. Some bugs were identified during the accepting testing and they were corrected and also some small features were changed according to the user requirements for better performance. Finally, overall feedback was indicated that the system was implemented successfully with all the objectives and customer satisfaction.

The questionnaire used to assess the customer feedback is given below with test results. All the users were given this evaluation form and data were collected and analyze

Figure 5.9 illustrates the product satisfaction survey form filled by salon owner. Along with other salon employees feedback for the survey analysed for final summry product satisfaction.

PRODUCT SATISFACTION SURVEY

Online Salon Management System for Samudra Bridal Palace

To complete this web base system in correct manner need your feedback in every way.

Please provide correct information regarding the system.

Put a "✓" (tick) in relevant column.

#	Description	Rating			
		Excellent	Good	Normal	Bad
1	User Friendliness	✓			
2	Usefulness	✓			
3	Understandability		✓		
4	Speed		✓		
5	Report format	✓			
6	Authentication	✓			
7	Security feature		✓		
8	Searching Techniques		✓		
9	User Tracking		✓		
10	Backup Facility			✓	
11	Ease of installation			✓	
12	Ease of learn		✓		
13	Compatibility with hardware		✓		

Comment :.....

Post/Title : *Salon owner*.....

Thank you for your valuable time spent on this.

Figure 5.9: User Feedback from Salon Owner

Figure 5.10 illustrates the product satisfaction survey form filled by salon manager (Samudra Bridal Palace Maharagama Branch). Along with other salon employees feedback for the survey analysed for final summry satisfaction.

PRODUCT SATISFACTION SURVEY

Online Salon Management System for Samudra Bridal Palace

To complete this web base system in correct manner need your feedback in every way.
Please provide correct information regarding the system.
Put a "✓" (tick) in relevant column.

#	Description	Rating			
		Excellent	Good	Normal	Bad
1	User Friendliness	✓			
2	Usefulness	✓			
3	Understandability	✓			
4	Speed	✓			
5	Report format	✓			
6	Authentication		✓		
7	Security feature				
8	Searching Techniques	✓			
9	User Tracking				
10	Backup Facility		✓		
11	Ease of installation				
12	Ease of learn	✓			
13	Compatibility with hardware				

Comment :

Post/Title : *Beautician*

Thank you for your valuable time spent on this.

Figure 5.10: User Feedback from Beautician

Overall satisfaction of SBP staff about the new system is good and Figure 5.11 illustrate the chart for the summary of all the staff answered for the satisfaction survey form.

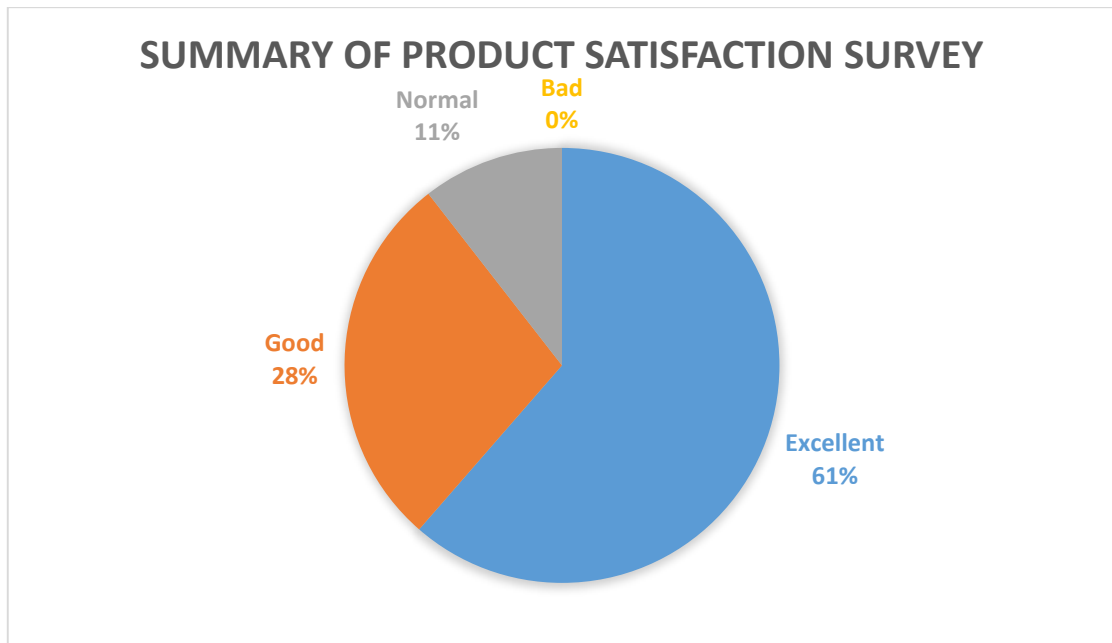


Figure 5.11: Summary of Product Satisfaction survey

CHAPTER 06: CONCLUSION

6.1. Overview

From start to end of this project, was prototyped to the client to ensure that his requirement has been fulfilled by the implementation of system in each phase. This constant checking with the client is assured that the developed system met the requirements of business that were identified in requirement analysis phase and client requested.

By user reviewing the functional and non-functional requirements that discovered during the analysis phase and checking back with the functionalities implemented in the developed system, it can be said that all the requirements of the user have been satisfied.

The simple user interfaces that was designed and developed are easy to learn and use proved to be satisfactory for the user.

6.2. Lessons Learnt

I was not familiar with the C# language or Visual Studio development environment. So, I had to learn C# and JavaScript frameworks including Angular 2 to fulfill this project. In addition to these Angular base UI designing was learned.

I found some grate video tutorials and some example, and view them and follow examples to grab lessons.

Mostly inexperience of project planning and scheduling identify and have grate advices from my supervisors to improve proper time management and planning.

6.3. Critical Assessment of Project

With the customer satisfaction survey summary, it indicates the system is achieved the most of their original requirements. Their functional requirements for the new system are gathered in system analysis part (CHAPTER 02: ANALYSIS) given bellow as summary.

- Manage Salon Report
- Manage Authentication
- Manage Branches (Salons)

- Manage Booking
- Manage Employee
- Manage Billing
- Manage Services and Product

And the nonfunctional requirements of the new system is given bellow as summary.

- User Friendliness
- Improved Reliability
- Simple Interfaces

All the above functional and nonfunctional requirements are fulfill by the new system and during the system development some more future enhancements ware identified.

6.4. Future Enhancements

Adding following future enhancement in the system are as follows.

1. Create attractive separate web sites for each branches using data collect in current version.
2. Improve appointment booking using mobile app
3. Enhance the features of the system

REFERENCE

- [1] Wikipedia. “Requirements analysis” Internet: http://en.wikipedia.org/wiki/Requirements_analysis [May 14 2017]
- [2] Accurate iOneSoft (Pvt.) Ltd. “JustBookSalon”. Internet: <http://justbooksalon.com> [May 10 2017]
- [3] Tyner Blain. “Ten Requirements Gathering Techniques”. Internet: <http://tynerblain.com/blog/2006/11/21/ten-requirements-gathering-techniques/> [May 21 2017]
- [4] PCMag Digital Group. “Definition of: Web application”. Internet: <http://www.pcmag.com/encyclopedia/term/54272/web-application> [May 28 2017]
- [5] Wikipedia. “Use Case Diagram”. Internet: http://en.wikipedia.org/wiki/Use_Case_Diagram [June 06 2017]
- [6] Wikipedia. “Class Diagram”. Internet: http://en.wikipedia.org/wiki/Class_diagram [June 10 2017]
- [7] Wikipedia. “Activity Diagram”. Internet: http://en.wikipedia.org/wiki/Activity_diagram [June 10 2017]
- [8] Wikipedia. “Entity Relationship Diagram”. Internet: http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model [June 16 2017]
- [10] Tutorials Point. “Angular 2”. Internet: https://www.tutorialspoint.com/angular2/angular2_overview.htm [June 03 2017]
- [11] MSDN. “Model View Viewmodel”, Internet: <https://code.msdn.microsoft.com/silverlight/How-to-implement-MVVM-71a65441> [June 12 2017]
- [12] Tec Target. “SQL Server 2012”. Internet: <http://whatis.techtarget.com/definition/SQL-Server-2012> [June 02 2017]
- [14] Angular.io. “Form Validation”. Internet: <https://angular.io/guide/form-validation> [August 26 2017]
- [16] GitHub. “mydatepicker”. Internet: <https://github.com/kekeh/mydatepicker> [August 17 2017]
- [17] NPM Package. “JS PDF”. Internet: <https://www.npmjs.com/package/pdf-viewer> [August 17 2017]
- [18] Daypilot. “Angular 2 Scheduler”. Internet: <https://doc.daypilot.org/scheduler/angular> [August 17 2017]

- [19] Wikipedia. “Unit Testing”. Internet: http://en.wikipedia.org/wiki/Unit_testing [August 17 2017]
- [20] Wikipedia. “Black Box Testing”. Internet: http://en.wikipedia.org/wiki/Black-box_testing [August 15 2017]
- [21] Wikipedia. “White Box Testing”. Internet: http://en.wikipedia.org/wiki/White-box_testing [August 15 2017]
- [22] Wikipedia. “Integration Testing”. Internet: http://en.wikipedia.org/wiki/Integration_testing [August 15 2017]
- [23] Wikipedia. “System Testing”. Internet: http://en.wikipedia.org/wiki/System_testing [August 15 2017]
- [24] MSDN. “How to Install and Configure Internet Information Services”. Internet: [http://msdn.microsoft.com/en-us/library/hh167503\(v=nav.71\).aspx](http://msdn.microsoft.com/en-us/library/hh167503(v=nav.71).aspx) [August 06 2017]

APPENDIX A – SYSTEM DOCUMENTATION

The system documentation provides guidance for system administrator to setup system on server environment to use it. This section contains importance information about system configurations, steps how to setup and additional details about hardware and software configuration needed.

Hardware Requirements

- Personal Computer
- Web Server
- Printer

Software Requirements

- Web Browser
- Web Server
- .NET 4.5
- SQL Server 2012
- IIS

How to setup system

A.1.1. Install SQL Server 2012

Step1: To start the SQL server installation, click the first option after selecting installation tab from the installation center as shown below. Since this is going to be new SQL server installation select the first option.



Figure A.1: Install SQL Server 2012 step 1

Step2: The installation will then check all the pre-requisites that are required to install the SQL server. If everything is fine, you will see green checks as shown below. Click OK.

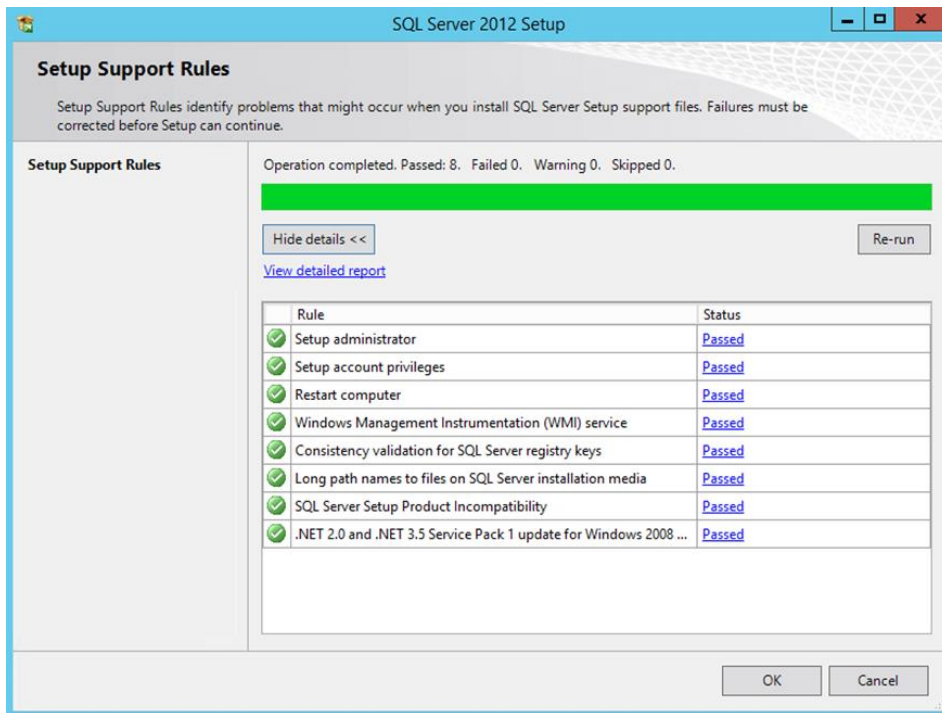


Figure A.2: Install SQL Server 2012 step 2

Step3: Accept the license terms and click Next.

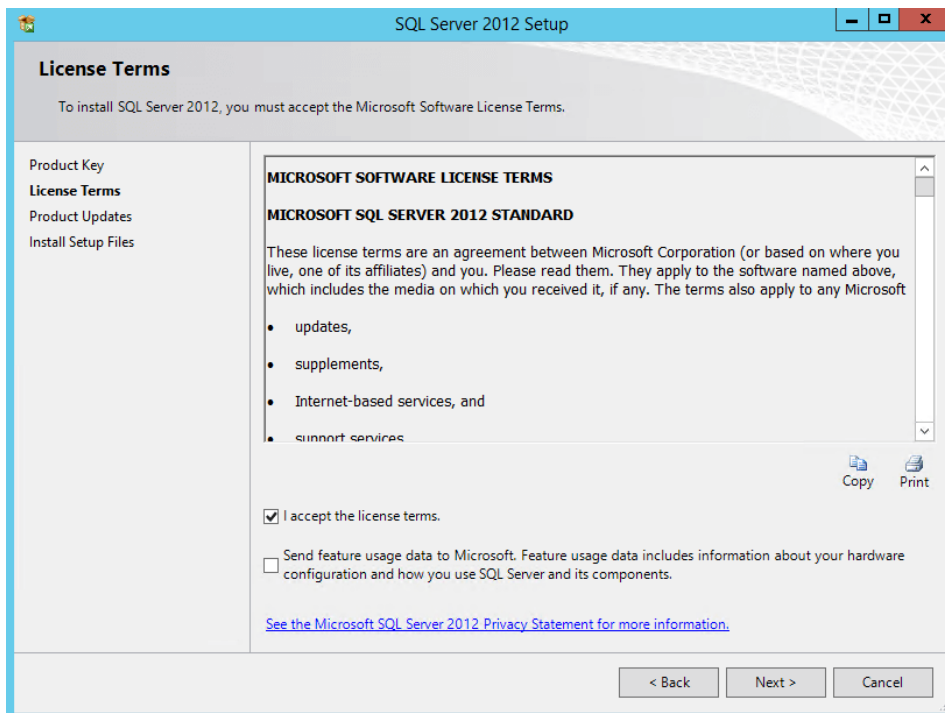


Figure A.3: Install SQL Server 2012 step 3

Step4: The installation does the pre-requisite checks once again as you can see below.

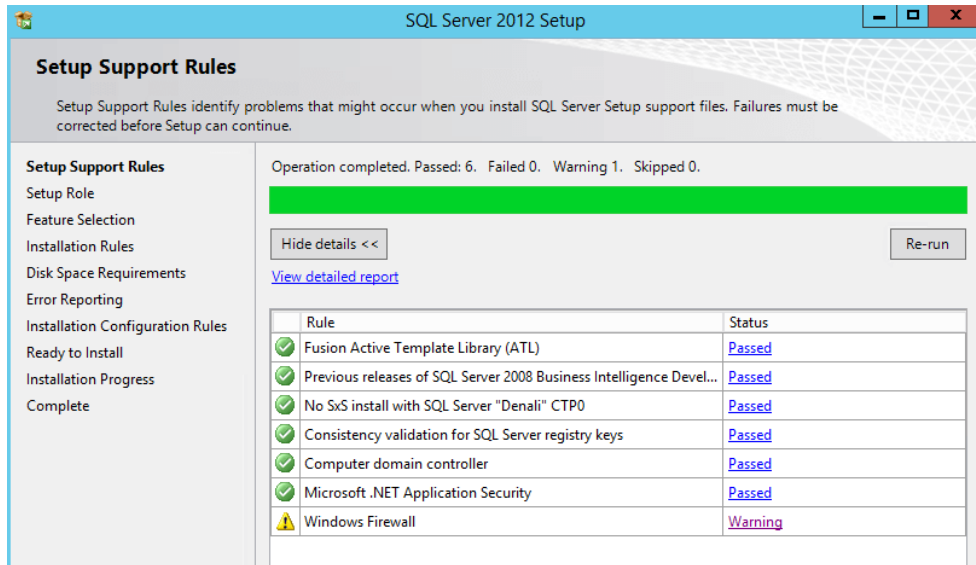


Figure A.4: Install SQL Server 2012 step 4

Step5: Select the required SQL features and click **Next**.

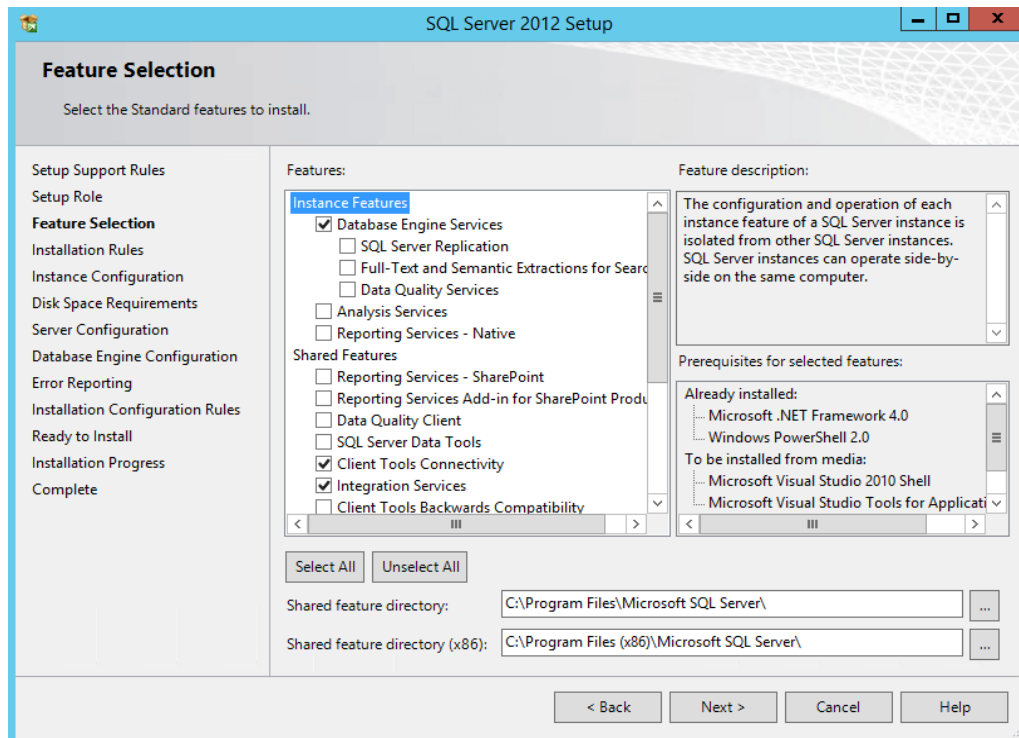


Figure A.5: Install SQL Server 2012 step 5

Step6: Under Instance Configuration, choose Default instance. You can also define custom instance name. Similarly, you have the option to choose the location where the instance will be stored. I will leave the defaults and click Next button.

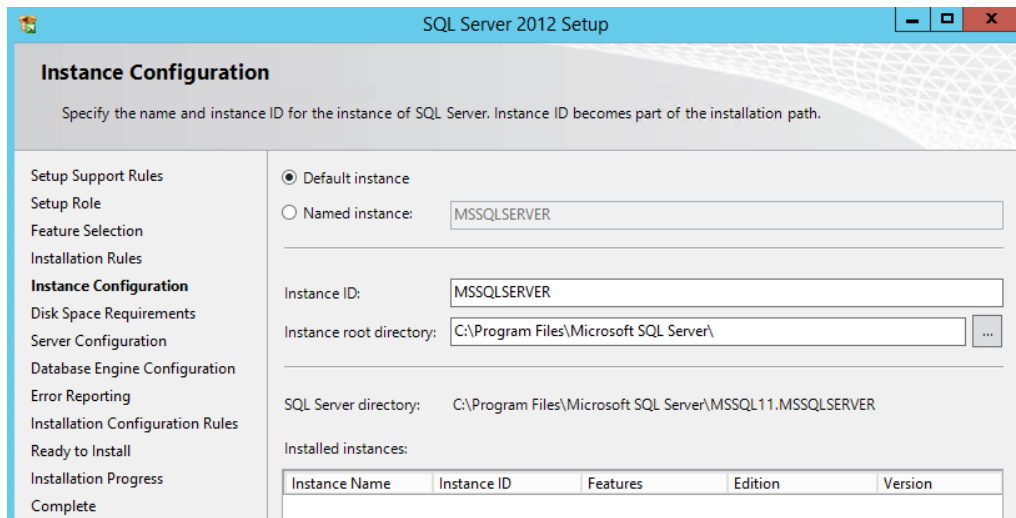


Figure A.6: Install SQL Server 2012 step 6

Step7: Under Server Configuration, review the Service Accounts, I will leave the defaults. Here, you can also use Active Directory user for various SQL services.

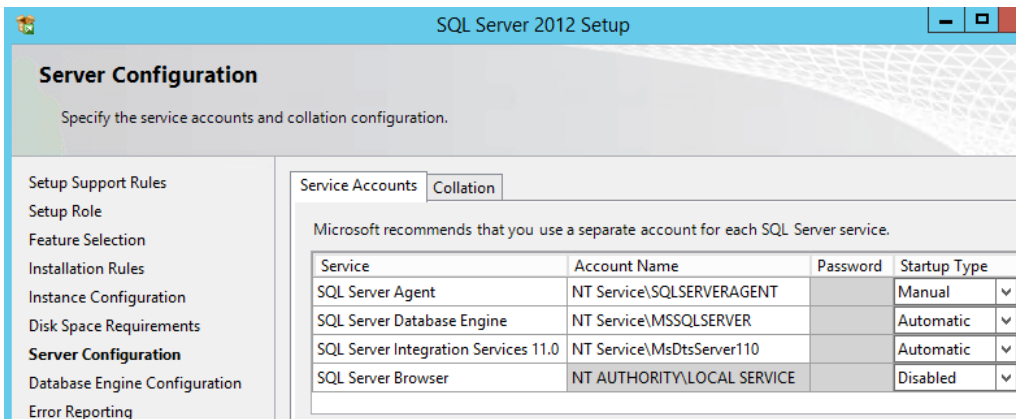


Figure A.7: Install SQL Server 2012 step 7

Step8: Under Database Engine Configuration, choose Windows authentication mode. You also have option to choose both windows authentication and SQL server authentication. Then, add the users who will be managing this SQL server and click Next.

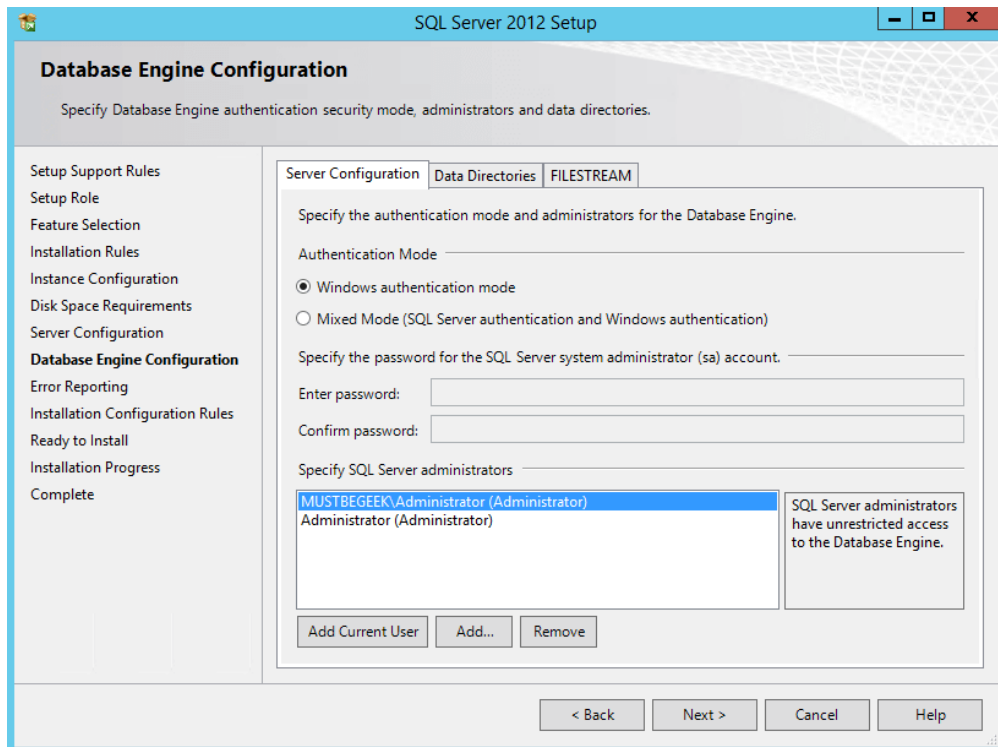


Figure A.8: Install SQL Server 2012 step 8

Step9: Under Ready to Install, review the installation information and click the Install button.

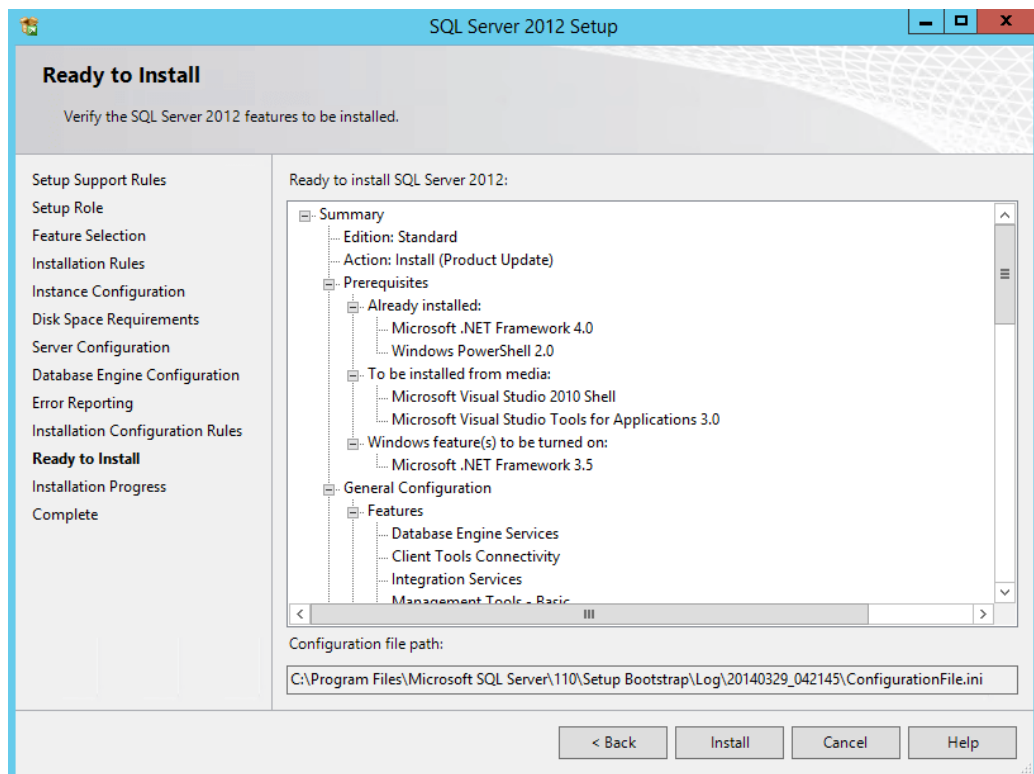


Figure A.9: Install SQL Server 2012 step 9

Step10: After the installation is complete, click Close. You can now start creating databases.

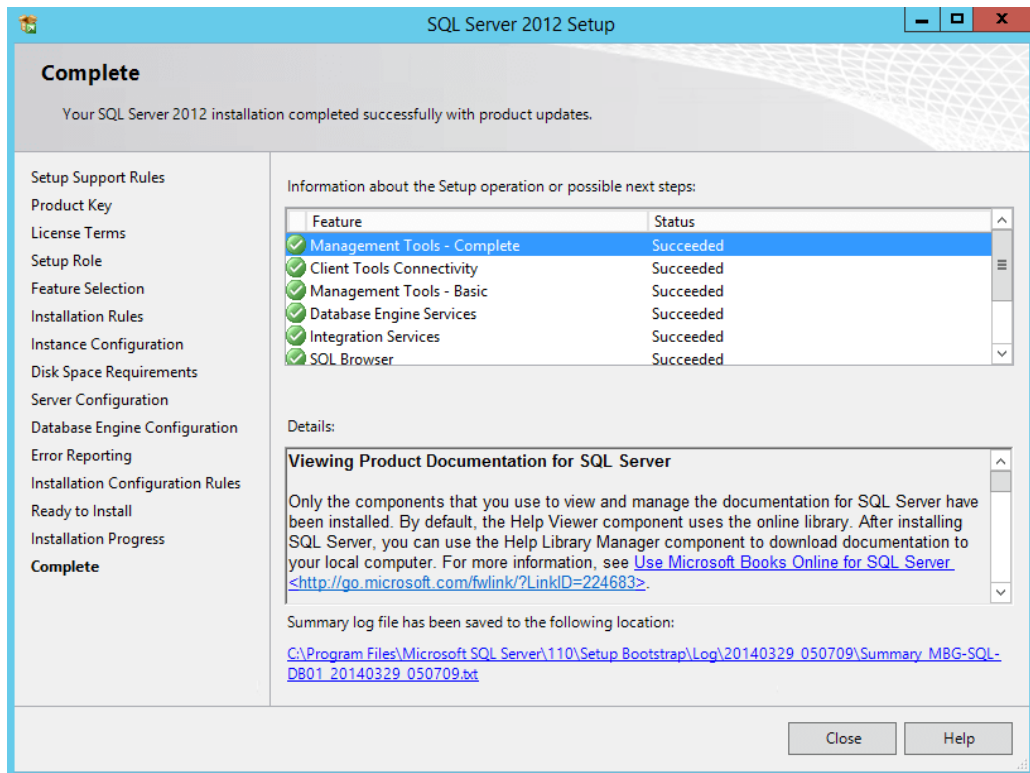


Figure A.10: Install SQL Server 2012 step 10

A.1.3 How to install IIS on Windows 8

1. On the **Start** page, choose **Control Panel**, and then choose **Programs**.
2. Under **Programs and Features**, choose **Turn Windows features on or off**.
3. In the **Windows features** dialog box, expand **.NET Framework 4.5 Advanced Services**, and then select **ASP.NET 4.5**.
4. Expand **WCF Services**, and then select **HTTP Activation**.
5. If the .NET Framework 3.50 is installed, then expand **Microsoft .NET Framework 3.5 (including .NET 2.0 and 3.0)**, and then select **Windows Communication Foundation HTTP Activation**.
6. In the **Windows features** list, expand **Internet Information Services**, and then expand **World Wide Web Services**.
7. Expand **Application Development Features**, and select the following features:
 - .NET Extensibility 3.5 (if .NET Framework 3.5 is installed)
 - .NET Extensibility 4.5

- ASP.NET 3.5 (if .NET Framework 3.5 is installed)
 - ASP.NET 4.5
 - ISAPI Extensions
 - ISAPI Filters
8. Expand **Common HTTP Features**, and then select the **Static Content** feature.
 9. Expand **Security**, and then select the following features:
 - **Request Filtering**
 - **Windows Authentication**
 10. To install Internet Information Service (IIS) Manager, under **Internet Information Services**, expand **Web Management Tools**, and then select **IIS Management Console**.
This step is optional
 11. Choose the **OK** button to complete the installation.
 12. To verify that the web server has been installed correctly, start your browser, and then type **http://localhost** in the address. The default web site opens and should display an IIS 8 image. For more information and to install on Windows 7 [24]

APPENDIX B – DESIGN DOCUMENTATION

Following Figure B.1 is activity diagram for add new branch salon to the system. In this scenario when user login system will check the credential and load the authorized module for user. Then user select “SMP Settings” module form view system will load the “Branch Salons” view. Then user able to add details for new branch and click “Save”. Then the system add new branch salon to data base and refresh the view for show the newly added branch salon.

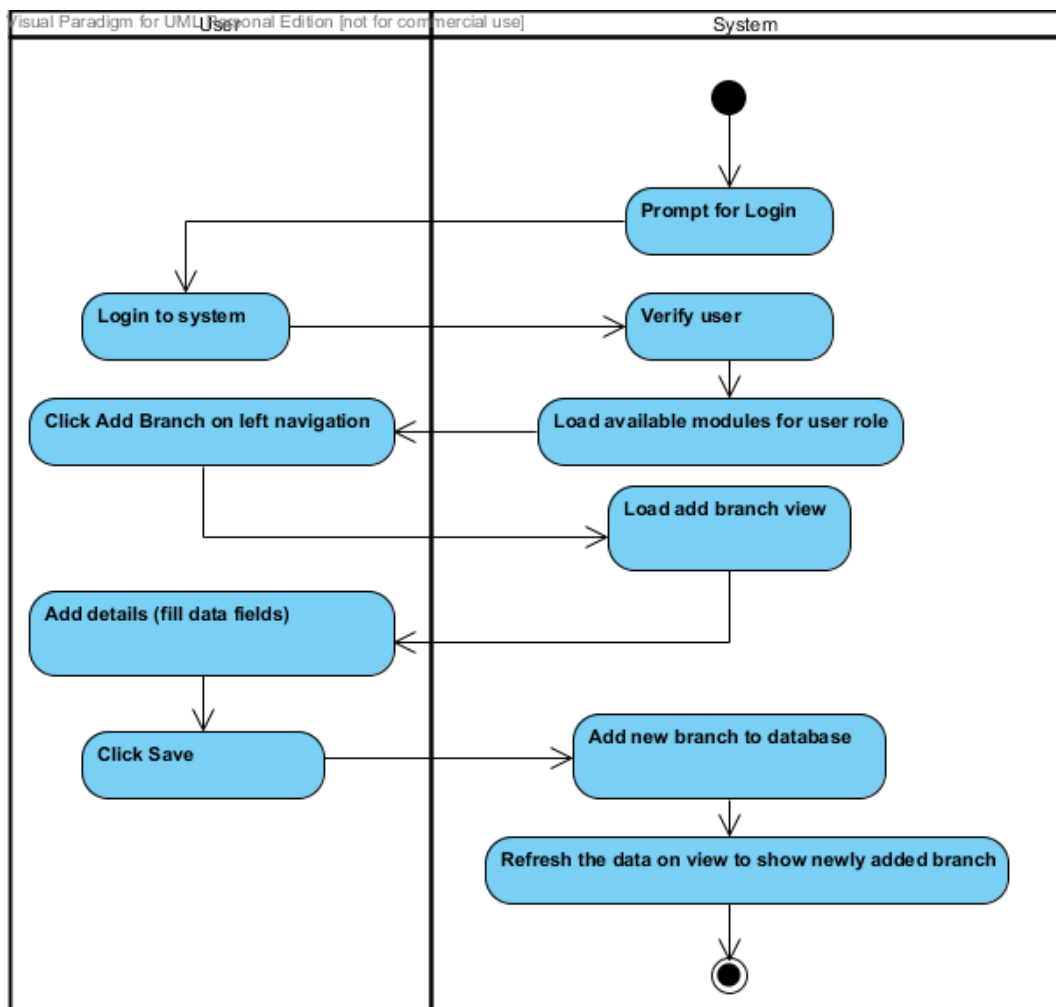


Figure B.1: Activity diagram for add new branch salon

Following Figure B.2 is activity diagram for deleting a services from system. In this scenario system shows the branch salons in database on the view. User select the delete

for selected collecting center. System will check the selected collecting center is in used or not. If it is in use show message can't delete. User can click ok to close message or after 20 second system will automatically closed the message. If it is not in use, record will delete (change the status) and refresh the view and removed the deleted collection center from view.

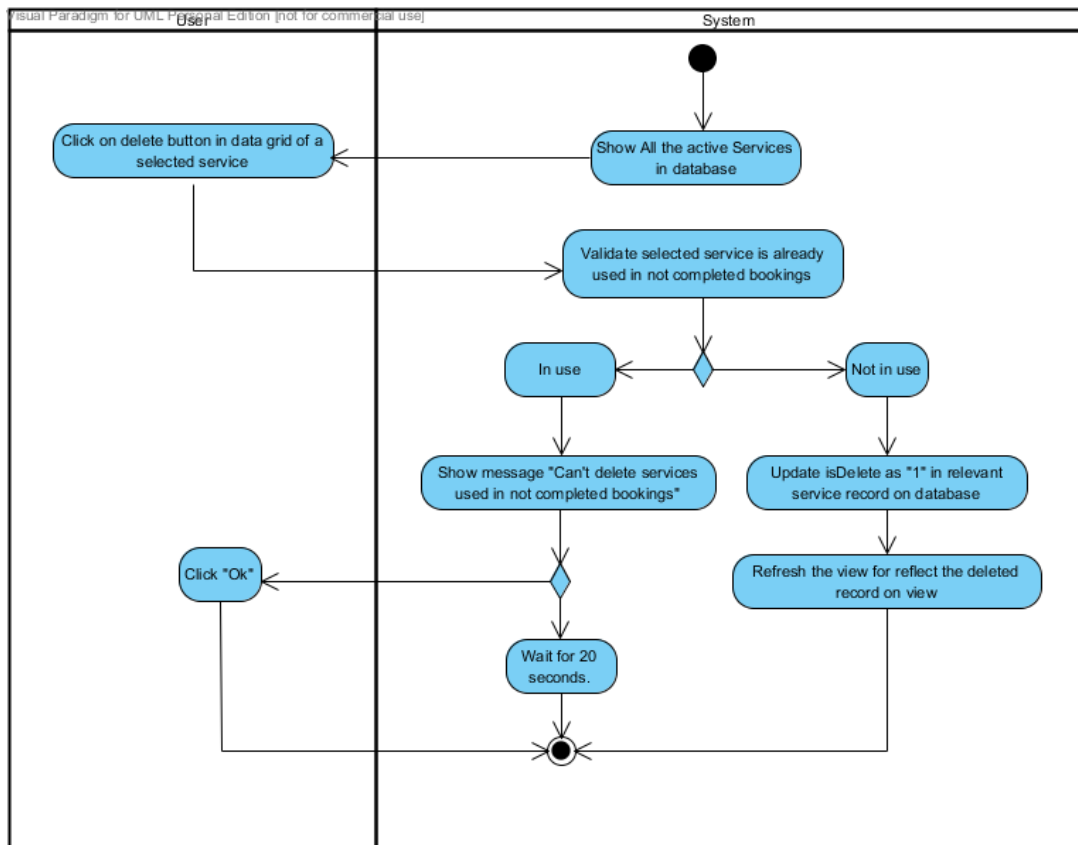


Figure B.2: Activity diagram for deleting a services

APPENDIX C – USER MANUAL

System Login

Employee must be login to the system with correct credentials. In this case system administrator password will be provided by developer.

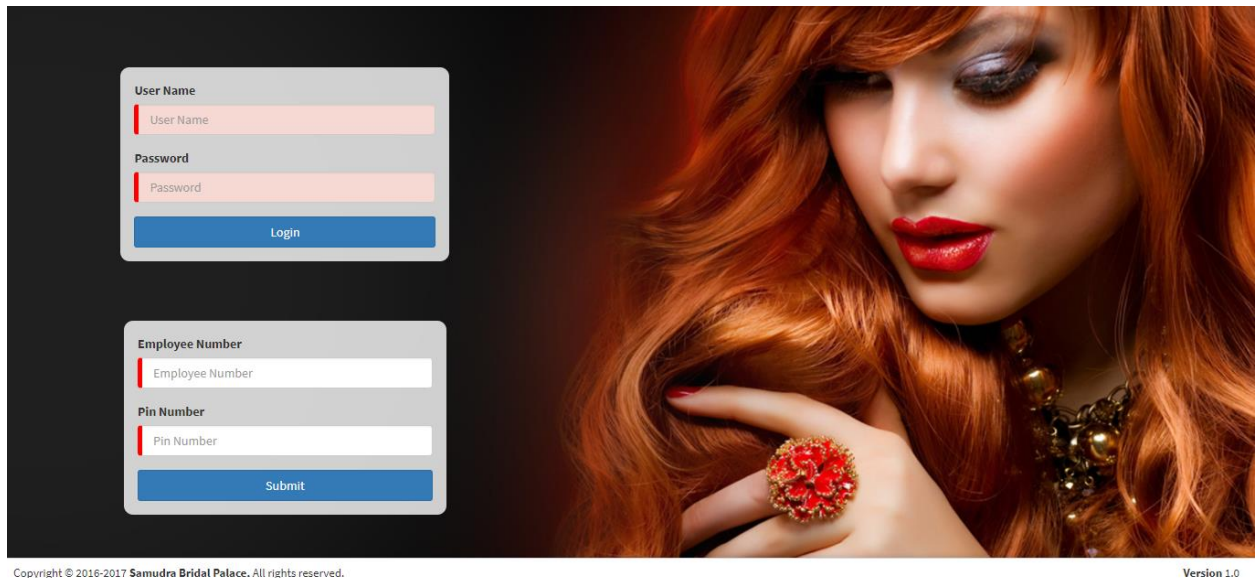


Figure C.1: Login view

Home Page

When the system administrator login to the system successfully it will appear the system home page (Figure C.1). On the home page user will see all the modules and some important details in categories view.

It's included,

- Modules
 - Booking
 - Services
 - Products
 - HRM
 - Reports
 - Invoice
- Brunch wise current ongoing booking details
- Current day employee attendance
- Stock details of products use in in house.
- Booking summery in chart view.
- Calendar view to direct access to booking view

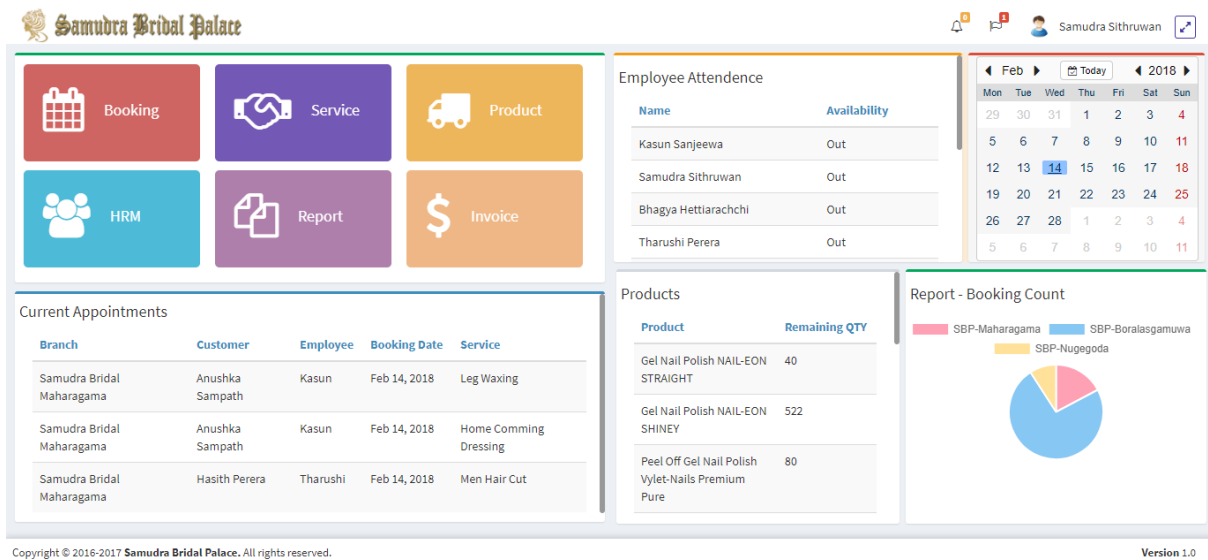


Figure C.2: Home Page View

Handel Booking

Create Bookings

Select a Branch Salon

When user select booking module, it will show the branch selecting view as shown in Figure C.2. Main process for create booking start from here.

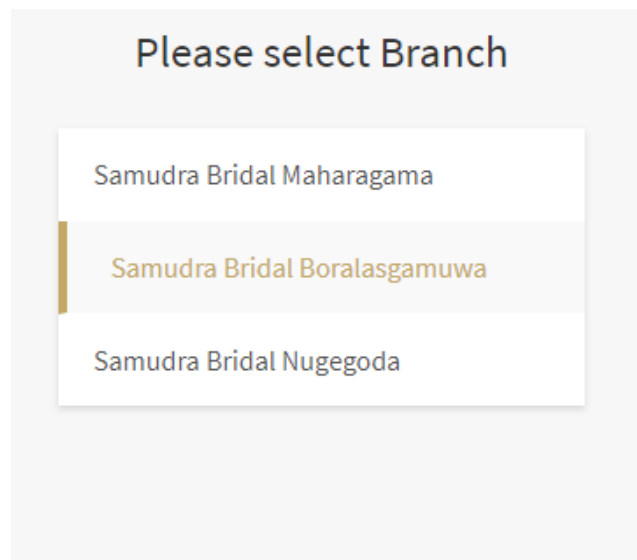


Figure C.3: Branch Selecting View

Booking Dashboard

From booking dashboard (Figure C.4) user can choose,

- A date for booking
- An employee for booking
- A time for booking

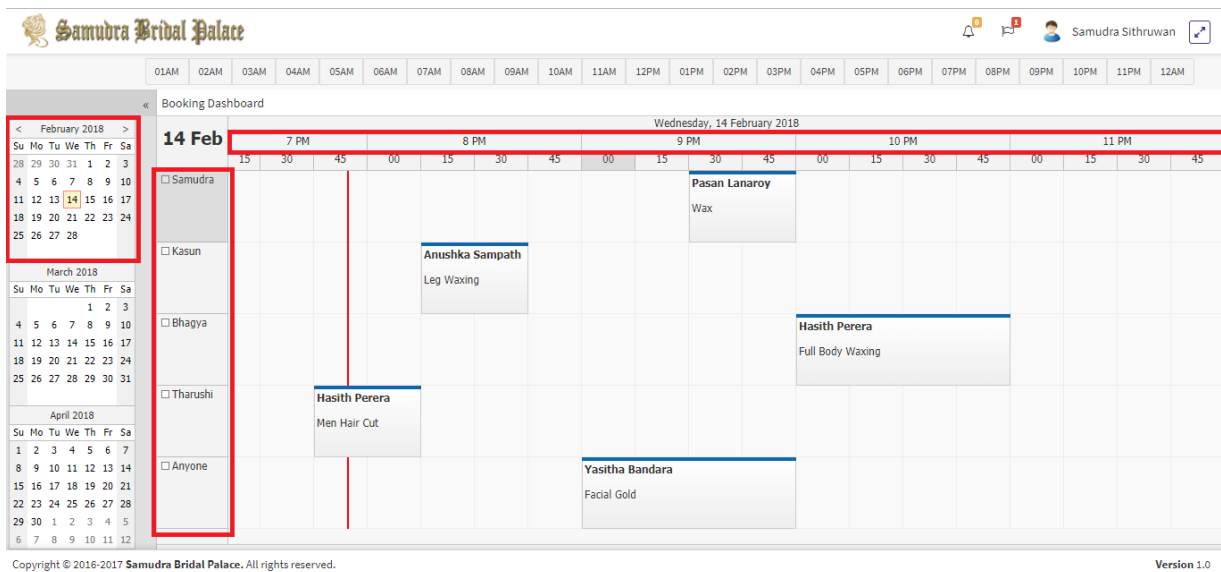


Figure C.4: Booking Dashboard

After decide date and employee, then user can click on time slot for as booking start time. Then the Booking Panel will appear shown as on Figure C.5.

Booking Panel

From the booking panel employee can select,

- Service
- Customer

To complete the booking.

In first step, employee can search and select relevant service (Figure C.5)

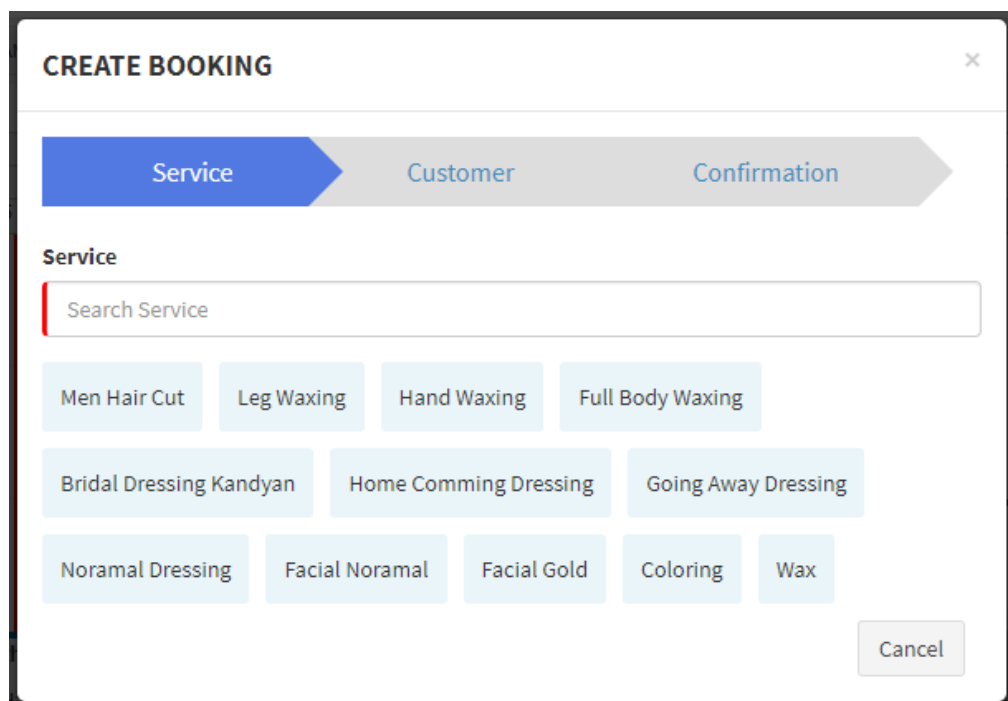


Figure C.5: Booking Panel – Service

When select a service it will navigate to the next step. In this step (Figure C.6) user can search and select relevant customer or add new customer

Figure C.6: Booking Panel - Customer

After select or add customer it will navigate to confirmation view as last step (Figure C.7). In this step it will show summary of booking and employee can add additional comment for booking.

When the booking completed with confirmation, it will appear on the dash board as shown in Figure C.8.

User can perform four main operations on the bookings

- View Booking
- Edit Booking
- Cancel Booking
- Complete Booking

CREATE BOOKING ✕

Service
Customer
Confirmation

Service
Hand Waxing

Service Duration
20 min

Service Price
LKR 300

Booking Date
Feb 14, 2018

Booking Start Time
22:35:00

Booking End Time
22:35:00

Employee Name
Tharushi

»

Customer Name
Tharidu Dilshan

Comment

Create Booking
Cancel

Figure C.7: Booking Panel - Confirmation

Booking Dashboard

Wednesday, 14 February 2018

	7 PM				8 PM				9 PM				10 PM				11 PM			
	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00
<input type="checkbox"/> Samudra																				
<input type="checkbox"/> Kasun																				
<input type="checkbox"/> Bhagya																				
<input type="checkbox"/> Tharushi																				
<input type="checkbox"/> Anyone																				

Bookings:

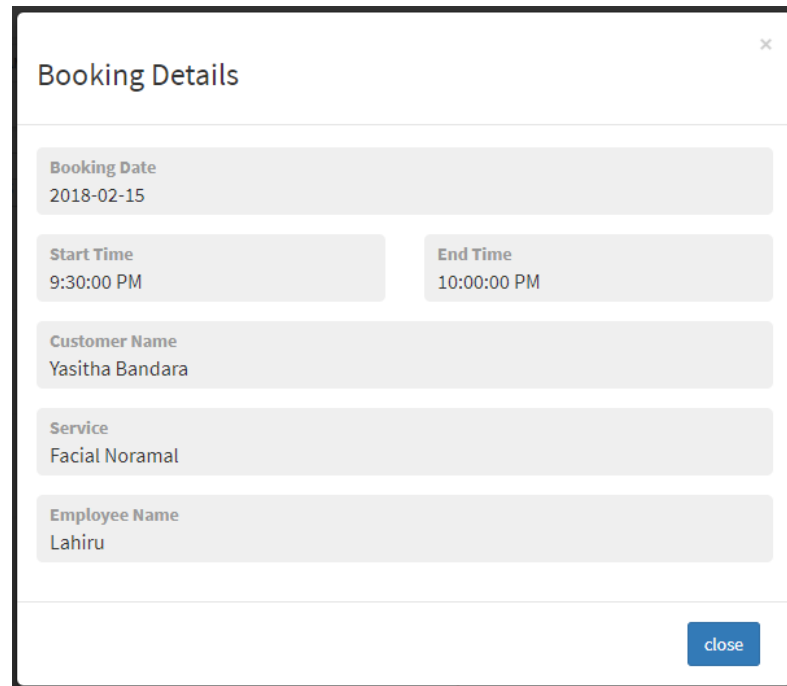
- Pasan Lanaroy** (Wax) - 9:15 PM - 9:45 PM
- Anushka Sampath** (Leg Waxing) - 8:15 PM - 8:45 PM
- Hasith Perera** (Full Body Waxing) - 10:15 PM - 10:45 PM
- Hasith Perera** (Men Hair Cut) - 7:45 PM - 8:15 PM
- Yasitha Bandara** (Facial Gold) - 9:15 PM - 9:45 PM

Figure C.8: Available Bookings Shown On the Booking Dashboard

View Booking

Click on a booking showing on dashboard, it will appear menu with all four options. Select view booking option (Figure C.8).

It will appear the detail view of the relevant booking as shown in Figure C.9.



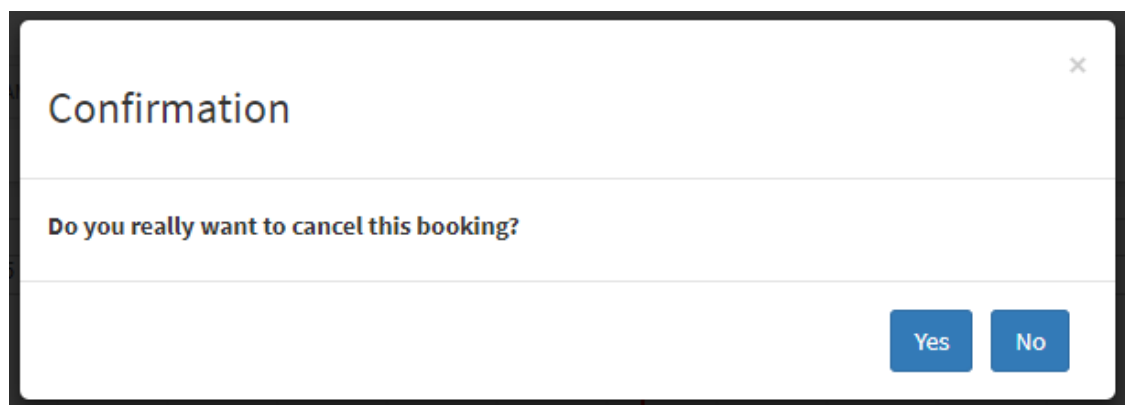
Booking Date	
2018-02-15	
Start Time	End Time
9:30:00 PM	10:00:00 PM
Customer Name	
Yasitha Bandara	
Service	
Facial Noramal	
Employee Name	
Lahiru	

close

Figure C.9: Booking Details View

Cancel Booking

Select cancel option from the menu appear when click on booking on dashboard. It will show the confirmation message to the user as shown in Figure C.10.



Confirmation

Do you really want to cancel this booking?

Yes No

Figure C.10: Cancel Booking Confirmation Message

Click on “Yes”. It will cancel the booking and it removed from the dashboard. Then user can add new booking to the same time slots.

Modify Booking

Select modify booking from the menu appear when click on booking on dashboard. It will show the modify booking view. As same as add booking view user can change the details of the booking from modify booking view (Figure C.11.)

MODIFY BOOKING [Close]

Service [Active] Customer Confirmation

Service

Facial Noramal

Day Spa Day Spa Men Hair Cut Leg Waxing Hand Waxing

Full Body Waxing Bridal Dressing Kandyan Home Comming Dressing

Going Away Dressing Normal Dressing Facial Noramal Facial Gold

Coloring Wax

Cancel

Figure C.11: Modify Booking View

Complete Booking

Select complete booking from the menu appear when click on booking on dashboard. It will appear the invoice view with the relevant service charge details as shown on Figure C.9.

Invoice Number: 68 Date: 14.02.2018 Customer: Anushka Sampath Employee: Kasun

Description:

Add Product Or Service

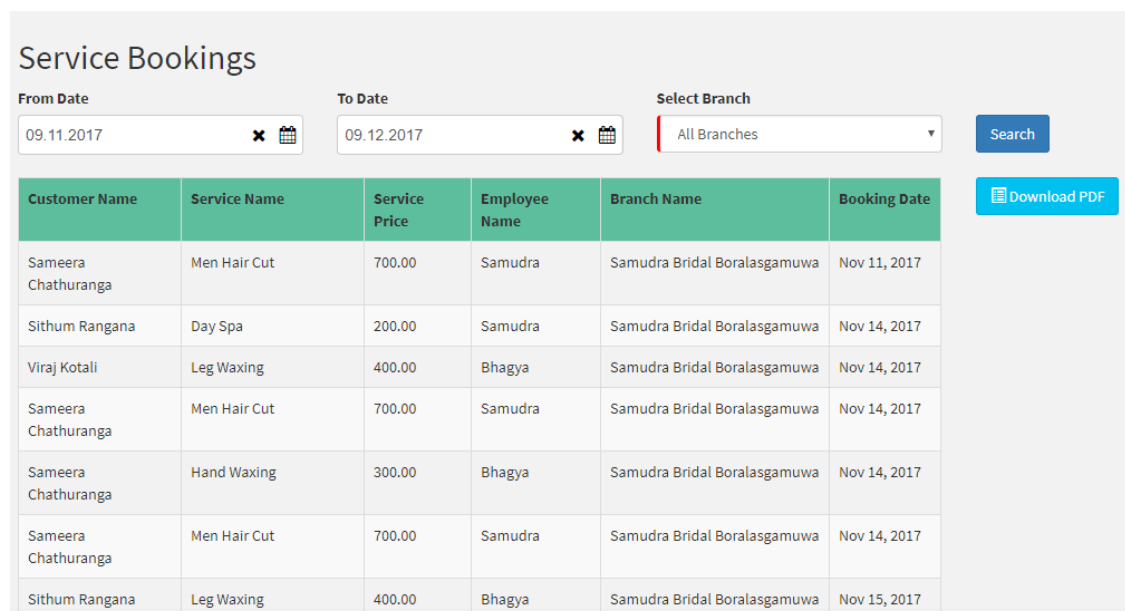
Product Or Service	Quantity	Unit Rate	Amount	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>
Product Or Service	Quantity	Unit Rate	Amount	Edit Delete
Leg Waxing	1	400	400	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
			Total - 400	<input type="button" value="Save"/>

Figure C.12: Invoice Generates When Complete Booking

APPENDIX D – MANAGEMENT REPORTS

OSMS provide various types of management reports for the owner. And even more reports can be added to the system with help of developers. These reports can help salon owners to make management decisions. Owner can generate results for reports by changing report criteria. Even it can be download as PDF files and it can use to print hard copy of report.

Following Figure D.1 is report for salon service bookings. For the selected date range owner can generate a report of bookings for all salon or selected salon.



Customer Name	Service Name	Service Price	Employee Name	Branch Name	Booking Date
Sameera Chathuranga	Men Hair Cut	700.00	Samudra	Samudra Bridal Boralasgamuwa	Nov 11, 2017
Sithum Rangana	Day Spa	200.00	Samudra	Samudra Bridal Boralasgamuwa	Nov 14, 2017
Viraj Kotali	Leg Waxing	400.00	Bhagya	Samudra Bridal Boralasgamuwa	Nov 14, 2017
Sameera Chathuranga	Men Hair Cut	700.00	Samudra	Samudra Bridal Boralasgamuwa	Nov 14, 2017
Sameera Chathuranga	Hand Waxing	300.00	Bhagya	Samudra Bridal Boralasgamuwa	Nov 14, 2017
Sameera Chathuranga	Men Hair Cut	700.00	Samudra	Samudra Bridal Boralasgamuwa	Nov 14, 2017
Sithum Rangana	Leg Waxing	400.00	Bhagya	Samudra Bridal Boralasgamuwa	Nov 15, 2017

Figure D.1: Salon service bookings report

Following Figure D.2 is report for employee attendance. Salon owner can be generate employee work record from this report. Report generate information about employee working hours and overtime. Mainly for employee salary and commotion.

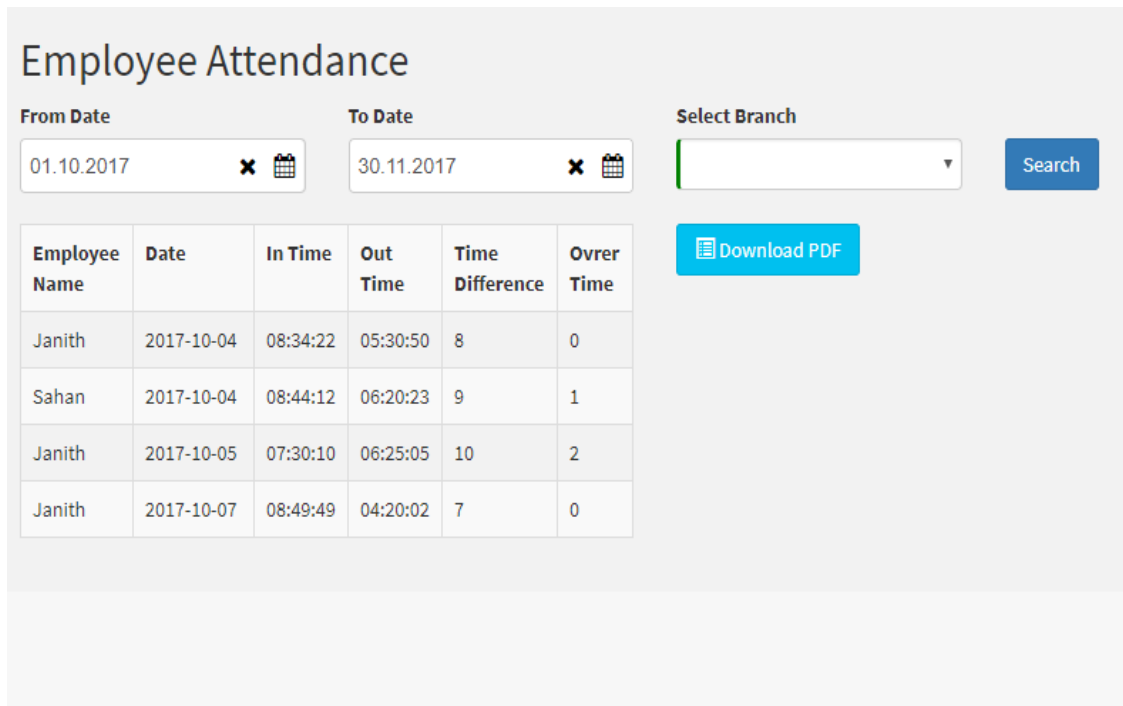


Figure D.2: Employee attendance report

Following figure D.3 is report of employee wise booking count for selected date range. This report also can be used to measure employee productivity.

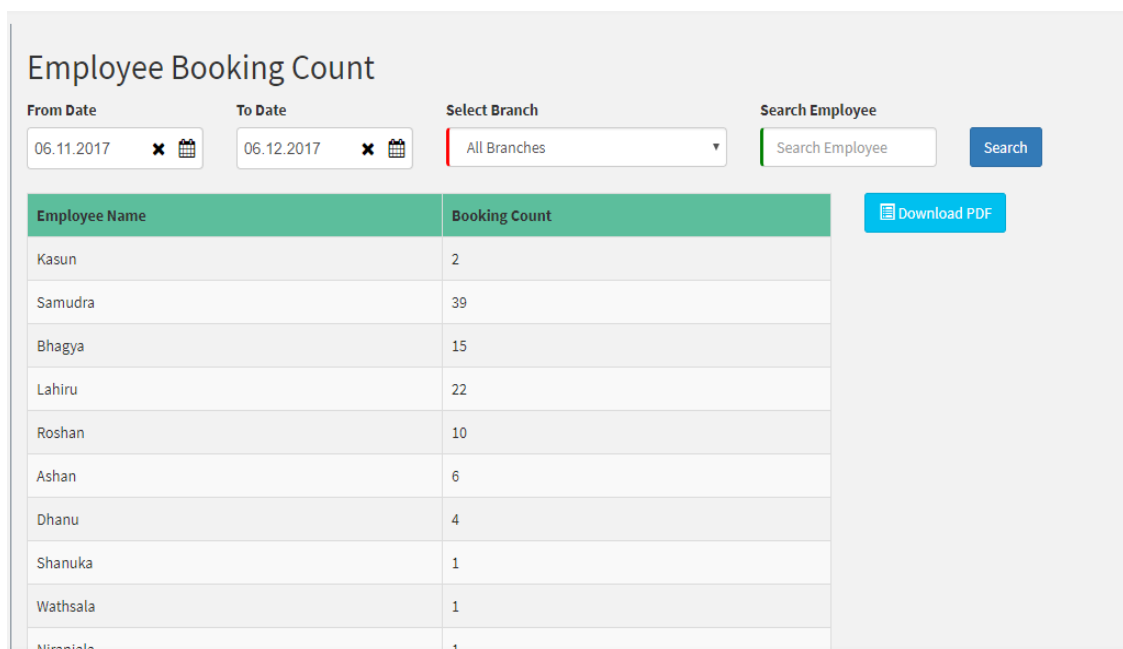


Figure D.3: Employee booking count report

Following figure D.4 is report of daily income from completed appointment booking. It's help owner to have an idea about daily income of each branch.

Daily Income

Date: 09.12.2017 ✕ 📅 Select Branch: Samudra Bridal Maharagama Search

Invoice No	Date	Service Name	Customer Name	Employee Name	Price
42	Dec 9, 2017	Home Comming Dressing	Pasan Lanaroy	Samudra	25000.00
43	Dec 9, 2017	Bridal Dressing Kandyan	Nipuna Niwanthika	Lahiru	20000.00
44	Dec 9, 2017	Leg Waxing	Nipuna Niwanthika	Lahiru	400.00
45	Dec 9, 2017	Wax	Yasitha Bandara	Ashan	300.00
46	Dec 9, 2017	Hand Waxing	Pasan Lanaroy	Roshan	300.00
47	Dec 9, 2017	Men Hair Cut	Yasitha Bandara	Ashan	700.00
48	Dec 9, 2017	Wax	Yasitha Bandara	Anyone	300.00
49	Dec 9, 2017	Leg Waxing	Chathura Saranga	Ashan	400.00
50	Dec 9, 2017	Normal Dressing	Sahan Bandara	Ashan	4000.00
51	Dec 9, 2017	Coloring	Yasitha Bandara	Ashan	1000.00

[Download PDF](#)

Figure D.4: Daily income from completed appointments

APPENDIX E – TEST RESULTS

Test cases for customer create booking online

ID	Description / Steps to Test	Expected result	Actual Result	P/F
	Go to marketing site	Marketing site should be loaded	Marketing site loaded	P
	Select a salon	Should be direct to the salon page	Directed to the salon page	P
	Select book now button	Services list should be loaded	Services list loaded	P
	Click on book button of a selected service	Booking view step 1 should be loaded	Booking view loaded	P
	Select a date form view	Should be direct to the step 2 time slot selecting view	Directed to the step 2.	P
	Select time slot	Should be direct to the step 3 confirm view	Directed to the step 3	P
	Click confirm button	Should be get closed create booking view	Create booking view get closed	P
	Go to adminconsole booking panel and select the above booking added date	On the time line it should be shown the relevant time slots covering for booking	On the time line it shown by covering the time slot of relevant booking	P

Table D 1: Test result for online booking

Test cases for admin create booking on dashboard

ID	Description / Steps to Test	Expected result	Actual Result	P/F
	Go to booking module	<ol style="list-style-type: none"> Features belong to the booking module should be loaded in to the left navigation. Booking dash board should be loaded as default selected feature 	Booking dashboard loaded	P
	Click on the future time slot between working hours.	Create booking button should be shown.	Create booking button appeared.	P
	Click on "Create booking" button	Create booking view should be appear	Create booking view appeared	P
	Select service from list	Should be move to next step 2 – select customer view	Select customer view loaded	P
	Search customer by name	Customers list should be loaded according to the entered name.	Customers list appeared according to the entered name	P

	Select a customer from list	Should be move to step 3 – confirm booking view	Confirm booking view appeared.	P
		Selected time, service and customer should be shown on the step 3 view	Selected time services and customer shown on the view	P
	Click Confirm button	Create booking view should be closed	Create booking view get closed.	P
		Create booking success message should be appear	Success message appeared.	P
		On the dash board, a time slot size of a service time should be covered.	Booking for the service indicating on the time line by covering time slots size of service time.	P

Table D 2: Test result for add booking from dashboard

Test cases for cancel booking.

ID	Description / Steps to Test	Expected result	Actual Result	P/F
	Go to booking dashboard	Already added booking should be shown on the time line by covering time slots.	Previous added bookings indicate on the time line.	P
	Click on the selected booking(time slot)	Menu should be appear with, <ul style="list-style-type: none"> • Cancel booking • Edit booking • Complete booking 	Menu shown with, <ul style="list-style-type: none"> • Cancel booking • Edit booking • Complete booking 	P
	Click on Cancel booking	Confirmation message should be shown to the user	Confirmation message appeared	P
	Click “Yes”	Confirmation message should be get closed	Confirmation message get closed	P
		Booking cancel success message should be shown to the user	Booking cancel success message shown.	P
		Time slot cover should be removed for relevant booking.	Time slot cover of the relevant booking has removed.	P

Table D 3: Test result for cancel booking from dashboard

Test case for complete booking with generate invoice

ID	Description / Steps to Test	Expected result	Actual Result	P/F
	Go to booking dashboard	Already added booking should be shown on the time line by covering time slots	Previous added bookings indicate on the time line.	P
	Click on selected booking	Menu should be appear with,	Menu shown with, <ul style="list-style-type: none"> • Cancel booking 	P

		<ul style="list-style-type: none"> • Cancel booking • Edit booking • Complete booking 	<ul style="list-style-type: none"> • Edit booking Complete booking 	
	Click on Complete booking	Invoice view should be generated with the service details included in booking.	Invoice view loaded Service price include in the invoice	P
		Invoice generated for the customer of booking.	Invoice created for the customer of the relevant booking.	P
	Click Finish button	Print view should be loaded in to the view	Print view of the invoice loaded	P
	Click print button	Print view should be closed	Print view closed	P
		Invoice generate view should be get closed	Invoice generate view get closed	P
		Booking complete success message should be shown to the user	Booking complete success message shown	P
		On the dash board, relevant booking should be change its color to Green	On the dash board relevant booking changed its color to green	P

Table D 4: Test result for complete booking from dashboard

APPENDIX F – CODE LISTING

Delete Service Form System

Html Code level

```

        <td class="text-center">
            <div class="popover confirmation fade bottom in hidden" id="{{item.Id}}"
style="display: block;">
                <div class="arrow"></div>
                <h3 class="popover-title">Are you sure?</h3>
                <div class="popover-content">
                    <p class="confirmation-content" style="display: none;"></p>
                    <div class="confirmation-buttons text-center">
                        <div class="btn-group">
                            <button (click)="confirmDelete(item)" type="button" class="btn
btn-xs btn-primary" data-apply="confirmation">
                                <i class="glyphicon glyphicon-ok"></i> Yes
                            </button>
                            <button (click)="cancelDelete(item.Id)" type="button" class="btn
btn-xs btn-default" data-dismiss="confirmation">
                                <i class="glyphicon glyphicon-remove"></i> No
                            </button>
                        </div>
                    </div>
                </div>
            </div>
            <div class="confirmation-buttons text-center">
                <div class="btn-group">
                    <button class="btn btn-sm btn-danger {{item.Id}}"
(click)="deleteServiceClick(item.Id)">Delete</button>
                </div>
            </td>

```

Delete Function

```

/**
 * Delete confirmation
 * @param {object} service
 */
confirmDelete(service: any) {
    this.deleteStatus = true;
    this._salonServiceService.deleteService(service.Id)
        .subscribe(reuslt => {
            // Reload service page
            this.loadServiceList();
        }, errorMessage => {
            // Log errors if any
            console.log(errorMessage);
        });
}

```

Service Layer for delete service

```

/**
 * delete service cateogry
 * @param {any} service category id
 */
deleteServiceCategory(id: any) {
  let url = this.commonService._serviceUrl + '/' + 'api/Service/DeleteServiceCategory';
  // let body: string = obj;
  let headers = new Headers({ 'Content-Type': 'application/x-www-form-urlencoded' });
  let options = new RequestOptions({ headers: headers });
  var x = "Id=" + id;

  return this._http.post(url, x, options)
    .map((res: Response) => {
      return res;
    })
    .catch((error: any) => {
      // Error on post request.
      return Observable.throw(error);
    });
}

```

Web API Layer

```

/// <summary>
/// Pass service id to Service API
/// </summary>
/// <param name="service"></param>
/// <returns>true of false</returns>
[HttpPost]
public bool DeleteService(Service service)
{
  return ServiceAPI.DeleteService(service.Id);
}

```

Service API

```

/// <summary>
/// Pass Service Id to Business layer
/// </summary>
/// <param name="serviceId"></param>
public static bool DeleteService(int serviceId)
{
  return ServiceBL.DeleteService(serviceId);
}

```

Service Business Layer

```

/// <summary>
/// Pass Service if to data access layer
/// </summary>
/// <param name="serviceCategoryId"></param>
public static bool DeleteService(int serviceCategoryId)
{
    return new DeleteServiceAction(serviceCategoryId).Execute(EnumDatabase.OSMS);
}

```

Data Access Layer

```

using OSMS.Domain.DomainObject;
using SP_DataAccess.DbFramework;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Common;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OSMS.DataAccess.SpDbCommands.ServiceSpCommands
{
    public class DeleteServiceAction : OSMSDbActionBase<bool>
    {
        private readonly int _serviceId;

        public DeleteServiceAction(int serviceId)
        {
            _serviceId = serviceId;
        }

        protected override bool Body(DbConnection connection)
        {
            try
            {
                // Common update for service delete
                const string storedProcedureName = "dbo.OSMS_Common_Update";
                var cmd = CreateCommand(CommandType.StoredProcedure,
storedProcedureName);

                // Get table name form service object
                cmd.Parameters.Add(new SqlParameter("@tableName", new
Service().TableName));
                cmd.Parameters.Add(new SqlParameter("@fields", "IsDelete=" + true + ""));
                cmd.Parameters.Add(new SqlParameter("@values", "Id=" + _serviceId + ""));
                cmd.ExecuteNonQuery();
                cmd.Dispose();
            }
            catch { }
        }
    }
}

```

```

        return true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}
}

```

Stored Procedure

```

USE [OSMS]
GO
/***** Object: StoredProcedure [dbo].[OSMS_Common_Update]  Script Date: 2017-11-08 1:02:38
AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:                Janith Jayaweera
-- Create date: 2017-07-08
-- Description:   Common Update
-- =====
ALTER PROCEDURE [dbo].[OSMS_Common_Update]
    @tableName varchar(50),
    @fields varchar(max),
    @values varchar(max)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    DECLARE @sql nvarchar(max)
    -- Insert statements for procedure here

    SET @sql = 'UPDATE '+@tableName+' SET '+@fields+' WHERE '+@values;

    EXEC sp_sqlexec @sql
END

```

APPENDIX G – CLIENT CERTIFICATE



Figure G 1: Client Certificate

INDEX

.

.Net24,25
.NET Framework24,60

A

ACTIVITY DIAGRAMviii,18
Analysis51,53
ANALYSIS51
appreciationiii
automationii

B

BITi,iii

C

Clientviii,5,23
CONCLUSION51
CURRENT MANUAL SYSTEMiv,6,7
customer baseii,1
customers1,2,3,7,10

D

Database21,24,29
Database Management Systemx,24
Delete35,36,42,44,62,76
DESIGNiv,x,4,9,11,12,21,22
Designingx,4,51
Development Toolsv,24
Document Analysis9

E

Effectiveii,9,22
ENTITY RELATIONSHIP DIAGRAMS21
Entity-Relationshipx
EVALUTION5

F

full fillii,2,6

H

Hardwarev,4,24
HARDWARE4

I

Implementation4,24
IMPLEMENTATION4
Interfaceii,v,1
Interview user9

L

Language12,16,24

M

Manager61
Microsoft6,24,53
Modelii,iv,vii,x,12
SQL24,53
SQL Serverii,viii,24

O

Object Orientedx,12
Observation9
Ownerii,vii,10

P

PROPOSED SYSTEM2,4,10,12
Prototyping9

R

Reliablei,1,6,10
Reports13,38,72
reputationii
REQUIREMENTiv,3,4,6,9,11,39

S

SCOPEiv,2,41

Software39,40,41
supervisors..... iii, 51

USE CASE DIAGRAM 12,53

U

unauthorized access ii, 7
Use Case..... iv,vii,ix,6,12

V

View x, 10
Visual Studio ii, 24, 27, 51