



# **Generalized Web Based System for Dental Services Management Institutes**

**A dissertation submitted for the Degree of  
Master of Information Technology**

**S.M.M.Dayananda**

**University of Colombo School of Computing**

**2020**



# **Generalized Web Based System For Dental Services Management Institutes**

**S.M.M.Dayananda**

**2020**

## Declaration

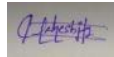
The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: S.M.M.Dayananda

Registration Number: 2017/MIT/012

Index Number: 17550129



---

Signature:

Date:20/06/2020

This is to certify that this thesis is based on the work of

Mr/Ms: S.M.M.Dayananda

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Prof:G.K.A.Dias



---

Signature:

Date: 21/06/2020

## **Abstract**

Oral healthcare industry mainly focuses on providing goods and services to patients that need treatments such as curative treatments, preventive treatments and palliative treatments. Then oral health care industry is one of the major component of healthcare industry in the world.

Government hospitals and private hospitals of Sri Lanka are providing oral healthcare Services. Some private dental service institutes also are providing such services. With the fast-paced life of the busy modern world, the lack of information technology is making a significant impact on the efficiency of the daily functioning of the hospitals and other institutes, customer satisfaction and many other factors. In lots of institutes, all the treatments, assets/equipment maintenance related tasks are handled using a manual file-based system, which makes the effectiveness of treatments lower and produces a high maintenance cost.

Developed system is a well-organized and generalized system which can be mapped for all types of dental services institutes and hospitals. Mainly, system will facilitate patient and staff registration, making appointments for treatments, tracking treatment details for each patient and keeping stock details. System also comprises of modules for report generating, handling payments and managing notifications through system users.

From all other process models, the incremental development process was chosen as the process model for the system by considering all the advantages it has over the other models. The system is being developed in MVC(Model-View-Controller) architectural design using Code Igniter framework based on PHP (PHP Hypertext pre-processor) for server-side scripting with, HTML and CSS are used to create interfaces and to make it more appeal, AJAX is being used to make the system more user-friendly, and MySQL is being used to develop the database of the system. Apache webserver has been chosen to host the system. Sublime Text is being used as the code editor for the development. Those tools were helped to achieve the anticipated benefits.

## **Acknowledgements**

I would like to express my deepest appreciation and gratitude to all those who provided me the possibility to complete my project successfully. At very first time, I would like to thank MIT course coordinator and academic staff of UCSC for giving this valuable opportunity to follow this internationally recognized master degree program and provided me the guidelines to a successful completion.

I extend my sincere gratitude to my supervisor, Professor of University of Colombo School of Computing, Prof: G. K. A. Dias for constant guiding, motivating and supervising me throughout the development. And also I would like to thank my advisor Mrs Uthpala Kumanayake for advising me all the time when I needed.

Furthermore, I extend my heartfelt thanks to Dentist of District Hospital, Godakawela Mrs Damayanthi Palika and Nurse in charge of orthodontic Dental unit of Teaching Hospital, Ratnapura Mrs Uththara for helping me during the project by providing me with the data needed to help make this project success.

Finally, I thank my friends and family for helping me to make this project a reality.

# Contents

Declaration .....	ii
Abstract .....	iii
Acknowledgements .....	iv
Table of figures .....	viii
List of tables .....	ix
List of acronyms .....	x
Chapter 1 – Introduction .....	1
1.1 Introduction .....	1
1.2 Motivation for the project .....	1
1.3 Objectives .....	2
1.4 Scope.....	3
1.5 Outline of the dissertation.....	4
Chapter 2 – Background .....	6
2.1 Requirement gathering techniques .....	6
2.2 Functional requirements .....	6
2.3 Non-functional requirements .....	7
2.4 Review of similar systems .....	7
2.5 Comparison between similar systems and the proposed system .....	10
2.6 Comparison of alternative design strategies .....	10
2.7 Justification of choice of the design .....	11
Chapter 3- Methodology .....	12
3.1 - Introduction .....	12
3.2 Architectural design of the system .....	12
3.3 Design methodology used.....	13
3.3.1 ER diagram of the system .....	14
3.3.2 Use case Diagram of the system.....	14

3.3.3 Use case narratives for login to the system .....	16
3.3.4 Use case narratives for making appointments .....	16
3.3.5 Use case narratives for generating reports .....	17
3.3.6 Use case narratives for manage treatments .....	17
3.3.7 Class diagram of the system .....	18
3.3.8 Sequence diagram for the system .....	19
3.3.9 Activity Diagram for the user login module .....	20
3.3.10 Activity diagram for the making appointments .....	21
3.4 Modules of the system.....	22
3.5 User interface design .....	23
3.5.1 Home page.....	23
3.5.2 Login interface.....	23
3.5.3 Admin Dashboard.....	24
3.5.4 Patients appointments details view .....	25
3.6 – Implementation environment .....	25
3.6.1 Hardware Environment .....	25
3.6.2 Software and Technologies .....	26
3.6.3 Major coding parts .....	27
Chapter 4 – Evaluation .....	30
4.1 Introduction.....	30
4.2 Test strategy.....	30
4.2.1. Unit Testing .....	30
4.2.2. Integrated Testing .....	30
4.2.3. System Testing .....	31
4.3 Test cases .....	31
4.3.1 Test case for user login and authentication .....	31
4.3.2 Test case for making appointments form .....	31

4.3.3 Test case for add new treatment .....	32
Chapter 5 - Conclusion .....	33
5.1 Critical evaluation of the project.....	33
5.2 Lessons Learnt .....	33
5.3 Future work.....	34



## Table of figures

Figure 2.1 Home page of Nawaloka hospital.....	8
Figure 2.2 View of Dental unit of Nawaloka hospital.....	8
Figure 2.3 Home page of Srilanka Dental Association .....	9
Figure 2.4 Home page of Durdans hospital .....	9
Figure 3.1 Model View Controller architecture pattern .....	13
Figure 3.2 ER diagram for the system .....	14
Figure 3.3 Use case diagram for required system .....	15
Figure 3.4 Class diagram for the system.....	18
Figure 3.5 Sequence diagram for the system .....	19
Figure 3.6 Activity diagram for the user login module .....	20
Figure 3.7Activity diagram for the making appointments.....	21
Figure 3.8 Modules of the system .....	22
Figure 3.9Login view.....	24
Figure 3.10 Admin dashboard view .....	25
Figure 3.11Patients'appointments details view.....	25
Figure 3.12 Implementation of database connection.....	28
Figure 3.13 Controller class for the login module.....	29
Figure 3.14 Model class for the login module .....	30
Figure 3.15 View class for the login module .....	30

## List of tables

Table 2.1 Comparison of similar systems and proposed system.....	10
Table 3.1 Use case naratives for login to the system.....	16
Table 3.2 Use case naratives for making appoitments .....	16
Table 3.3 Use case naratives for generating reports.....	17
Table 3.4 Use case naratives formanage treatments.....	17
Table 4.1 Test case for login and authentication.....	32
Table 4.2 Test case for making appointments.....	33
Table 4.3 Test case foradd new treatment .....	33

## **List of acronyms**

MVC - Model View Controller

ERD - Entity Relationship Diagram

UML - Unified Modelling Language

OOP - Object Oriented Programming

SQL - Structured Query Language

PHP - Hypertext Pre Processor

HTML-Hypertext mark-up Language

# **Chapter 1 – Introduction**

## **1.1 Introduction**

Usually, government hospitals and private hospitals of Sri Lanka are providing dental clinical services. And also some private dental service institutes are providing such services. Work loads of institutes may differ from one to another according to the level of services they are providing. Servicing may vary from general oral treatments to orthodontic treatments. Hard to find generalized solution for all type of institutes or hospitals. This generalized dental service management system can be used for any dental service institution. Another problem is most of these institutes or hospitals are managing such operations using manual book keeping process. In this existing scenario all records relevant to the patients, staff details, treatments' details and stock details are maintained using manual file-based process. From patient registration to closing treatment plan has to maintain several kinds of record books. Sometimes, for a one activity, 2 or 3 books have to be updated. Stock management is also maintain using books. This whole process is very time consuming task. And other thing is both staff and patients are facing lots of difficulties due to lack of interactions between them. Dental service management web based system is a good solution to avoid these difficulties in the existing management process.

## **1.2 Motivation for the project**

Nowadays the concern of oral healthcare have increased among both younger and elder generations due to vital enhancement of technology in oral healthcare industry. In Srilankan context most of the current dental services management systems are developing for a specific client. Rather than following same procedure, aim is to

develop a system which can be generally used by all types of dental services institutions.

With manual operational environment, institutes may face lacks the efficiency and quality of the service. Main function of Dental services institutes is provide good service to the patients. This system may strongly supports to the main function by reducing delays due to manual operational environment.

### 1.3 Objectives

The system is mainly aims at giving a well-organized and generalized system which can be mapped for all types of dental services institutions, while minimizing operating costs and maximizing effectiveness of service.

The following are the objectives identified for the system.

- Provide a generalized solution

Since this system should be used by any types of dental service management institutions, functions should be mapped with requirements from all types. For a one particular institute, relevant services and operations of that institute should be included in the system.

- Reducing paper works

Normally, patient registration to making payments tasks are done by using lots of paper works. And also it is very hard to find historical details related to patients or treatments. By replacing this system, paper works can be reduced.

- Enhance interaction among staff and patients

When considering manual operational environment, patient come then meet doctors and take treatments and finally leave. There is no any further communication stuff

among patients and staff in the institute. This web based system will bridge this gap by notifying current updates to all.

- Handling payments easily

Patients should be able to make their payments in an easy way. Staff also should be capable to handle payments within the institute using a reliable process.

- Update stock details automatically

Existing procedures that can be found in a dental services providing institutions, used a book based inventory to keep stock details. It is very time consuming task which wants to be managed by a trained person. Unlike that procedure, system should facilitate to update stock details automatically when some equipment has added or has used.

- Provides data analysis/reports

System should be able to provide some information related to a patient, a treatment or a specific time period. Report generating module will do analysis task through reports.

## 1.4 Scope

Following is the scope of the system according to the objectives.

- Configuration

Since this is a generalized solution, when an institute logged to the system at first time, there are some configurations which wants to be set up by them relevant to their institute.

- User Management

This consists user registration, keeping users' details, user authentication and update details.

- Manage Appointments

This module consists of making appointments by patients and managing that made appointments by staff.

- Manage Treatments

This consists keeping track of treatment details of patients. When considering one patient, treatments may be launched in planned order. Then treatments should be tracked.

- Manage stock

Equipment which needs to treatments are managing in a stock. When ordering new items from market stock should be updated manually. When using any item for a treatment that record should be reduced from stock automatically. And also system should notify when an item reached the minimum count.

- Manage notifications

Managing with tight scheduling and multiple tasks can sometimes lead to losing focus towards priority tasks. It lets you prioritize your task by setting reminders for important tasks so you can manage them accordingly.

- Report generating

System generates reports as the requirement of system users with respect to a specific time period or relevant to a patient or a treatment.

- Manage payments

This consists a module which helps to do transactions between service provider and client.

## 1.5 Outline of the dissertation

## Chapter 2- Background

This chapter comprises the requirements gathering techniques, similar systems studied to do the feature analysis, UML diagram for the ongoing process and a comparison of alternative design strategies. Moreover, functional and non-functional requirements are comprised.

## Chapter 3 –Methodology

The design of the system has given in this chapter. The system should design according to the identified requirements. This chapter includes UML design diagrams such as use case, EER, Class and state diagrams, furthermore some of the user interfaces and tools which were used in this implementation environment.

## Chapter 4 – Evaluation

This chapter gives a critical evaluation of the system. It will discuss whether the project objectives were satisfied and if not, the reasons for them and lessons learnt during the course of the project expanded upon.

## Chapter 5 - Conclusion

This chapter consists of the work indicating a summary of the results of the project and highlight how improvements could be made through future work.



## Chapter 2 – Background

### 2.1 Requirement gathering techniques

There are numerous techniques available when it comes to requirements gathering, like interviews, observations, focus groups, and document analysis, etc. On this project, requirements had been acquired via the usage of the following techniques.

- Interviews
- Document analysis
- Observation

Initially, had few meetings with the head of a private dental practise centre and gathered all the necessary information about the requirement. All the treatments details, inventory maintenance details were given by her. Subsequently had a meeting with the nurse in charge of orthodontic dental unit of teaching hospital, Ratnapura. By distinguishing the requirements, a separate collection of meetings were conducted to obtain the existing documents of the manual system and other essential information. Based on the collected data and current materials, functional and non-functional requirements were identified.

### 2.2 Functional requirements

The following are the functional requirements identified for the system.

- Facilitating User registration and user login
- User management
- Handling appointments
- Managing and tracking treatments
- Generating reports
- Making payments
- Managing stock
- Handling notifications

## 2.3 Non-functional requirements

The following are the non-functional requirements identified for the system.

- Automated Database Backups

Database must take backups of the existing data within a specified time period. This will prevent the system from data loss up to a certain extent and promote the reliability of the system.

- Mobile and cross browser compatibility

Since the system may be accessed through every possible device with internet connection, one should be able to adopt in to those views accordingly. Many users use a wide range of browsers to access the internet. Dental service management system must be able to provide a similar view in each of those browsers.

- Ease of Use

System must be user friendly in every way. Human eye friendly colours should be used in the system. Interfaces should not be complicated to its user. Some modern UI and UX concepts such as material design should be adopted in this system in order to give a good impression to the user.

## 2.4 Review of similar systems

- Nawaloka Hospital Dental unit[1]

Nawaloka hospital is one of the leading private hospitals in Srilanka. Figure 1 shows the Home page of the Nawaloka hospital. They are not maintaining the separate website for dental services management. Dental unit is sub part of the main website (Figure 2). Online sub system of Dental unit of Nawaloka hospital providing only channelling facility since dental unit is managing under Nawaloka hospital website.

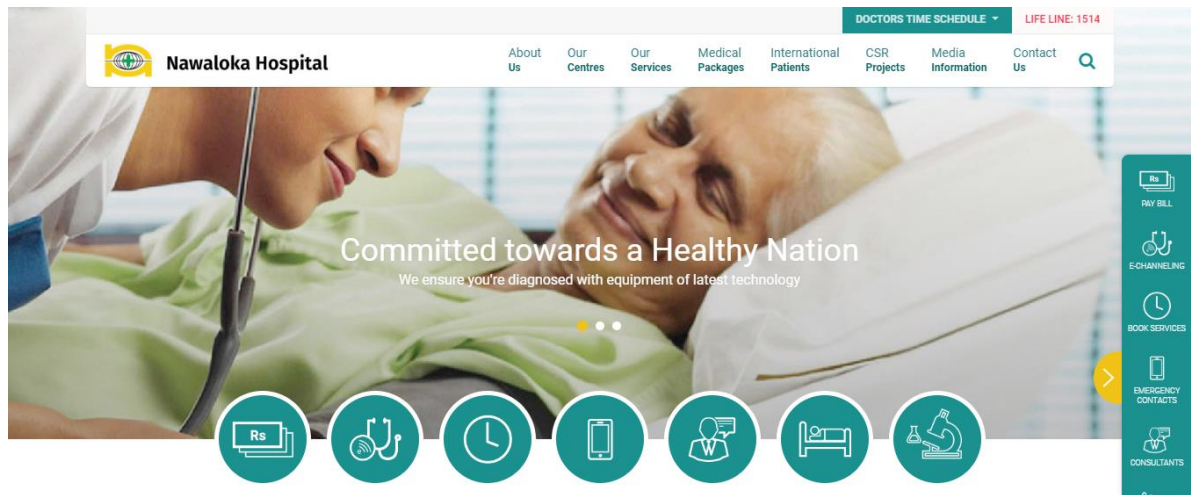


Figure 2.1 Home page of Nawaloka hospital

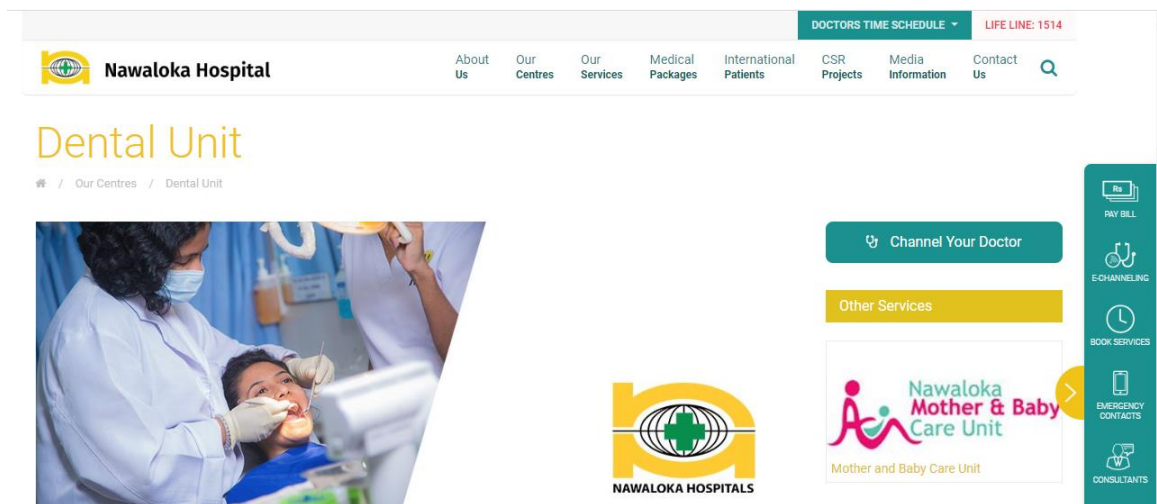


Figure 2.2 View of Dental unit of Nawaloka hospital

- Srilanka Dental association[2]

Srilanka Dental Association managing a website to help maintaining oral health of Srilankan nation. Figure 2.3 shows home page of Srilanka Dental Association's website.

This is basically used to providing information to the patients and staff. This is an informative website. They are using eight number of main menus in their website.



Figure 2.3 Home page of Srilanka Dental Association

- Durdans Hospital Dental Unit[3]

Durdans hospital also one of the leading hospital in private sector in Sri Lanka. It provides several treatments related to the oral health. They maintaining dental unit as a sub part of the hospital’s website.

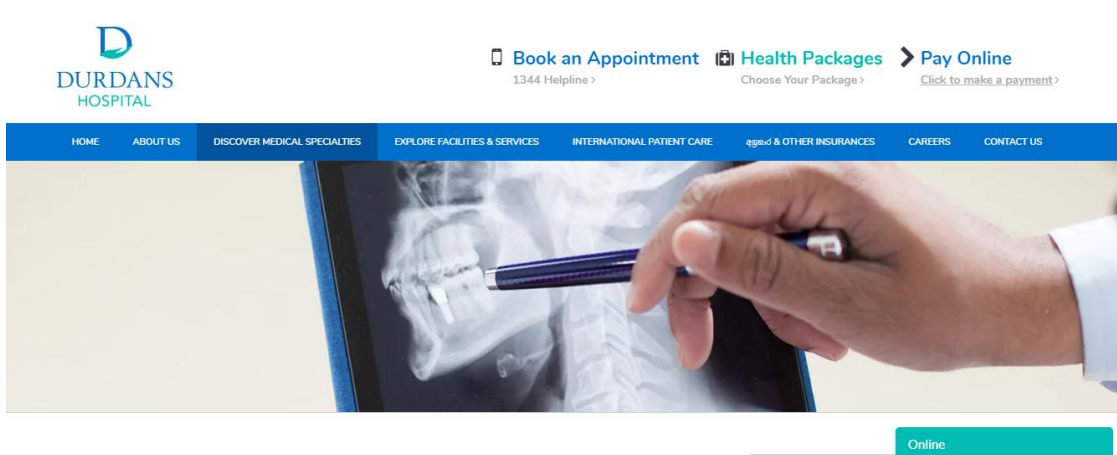


Figure 2.4 Home page of Durdans hospital

## 2.5 Comparison between similar systems and the proposed system

There are three examples for dental management systems as explained earlier. Nawaloka hospital dental unit and Durdans hospital dental unit are managing as sub part of hospital website. Srilanka dental association website is an informative website that gives information about oral health to public. Nowadays, an online system should be complete with functions. So, it is convenient for the user of the system. The proposed system will fill the gap of these online sites.

Similar systems	Proposed system
Dental services are managing as a sub part of the hospital website.	Whole system have dedicated to managing dental services.
Srilanka dental association website is an informative website.	Includes information related to the oral health. But not the informative website.
Basically unavailable the treatment plans of the patients and treatment history.	Comprise a separate treatment module to manage treatment plans and treatment history of the patients.
Unable to generate reports	Facilitate generating reports.

*Table 2.1 Comparison of similar systems and proposed system*

## 2.6 Comparison of alternative design strategies

It can be divided three sectors which is used for design primarily software system as follows, stand-alone system, mobile application and web based system. According to information which is getting from analysis phase, it is decided to design web based system.

- Standalone software

Standalone software can be developed and installed on to the client's computer. This process will require to be installed a server software to facilitate database functionality.

- Mobile application

A mobile application can be developed and integrated in to a downloadable store. This will give the ability to access system from anywhere.

- Web based software

Web based software will require a web browser to be installed on to the computer with an internet connection.

## 2.7 Justification of choice of the design

All of the above approaches have their own advantages and disadvantages. A web based software approach has been selected to this process, since it is most suitable for the background of the project. Standalone software need to work with a server and the server will require some additional resources to run smoothly. A user may find installing a server and starting it up on his or her machine would be an extra piece of work. Additionally, the database has to be update synchronized to a centralized server with time to time. This will take some additional resource allocation compared to other methodologies. New versions of the system also will be also needed to be installed on each and every user's machine.

A mobile application will require the user to have a web brows able phone such as a smart phone which in turn could be expensive and every user may not be able to afford a smart phone. A web based software solution will only require an internet connection with a browser. Versions or updates to the software can be easily deployed without breaking the system. Mobile version of the software can also be easily created with the help of responsive design frameworks for the web. No synchronization of data or hardware upgrades will be required within this approach. From the developer's perspective PHP was selected as the development language for this project since it is platform independency, robustness and wide range of available plugins and documentation.

## **Chapter 3- Methodology**

### **3.1 - Introduction**

Based on the required general solution and the detailed analysis of a new system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the development of a system.

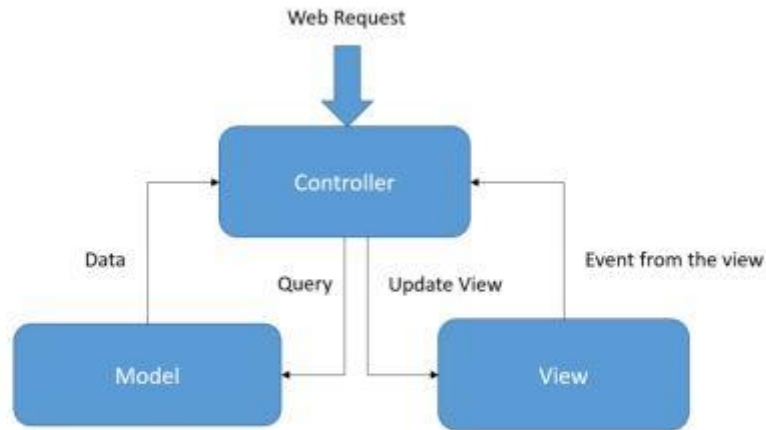
The logical system design arrived at as a result of system analysis and is converted into physical system design. In the design phase the SDLC process continues to move from the questions of the analysis phase to the how.

In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. Data structure, control process, equipment source, workload and limitation of the system, Interface, documentation, training, procedures of using the system, taking backups and staffing requirement are decided at this stage.

This chapter focuses on Object-oriented designing. UML (Unified modelling language) diagrams that have been used to design the system are incorporated. Moreover, some of the interfaces have been comprised.

### **3.2 – Architectural design of the system**

A better solution is given to develop the object-oriented systems for the general problems when developing the software by the design patterns. MVC design patten is used for the proper system. It reduced the coupling between database and UI model by this architecture. Bad effects which affects for the whole system are reduced by the changes which are occurring in the future time



*Figure 3.1 Model View Controller architecture pattern*

### 3.3 – Design methodology used

Among the object-oriented design, functional oriented design, structural design, OOD is used to design this system and it can be provided the client requirement efficiently and effectively which compatible for its complementation.

OOD design approach was selected for this project over structured approach. OOD concepts help to align the system components with the real word scenarios with flexibility and clarity. Reusable components can also be generated easy with OOD concepts. Unlike in structural approach changes in the requirements can be implemented in a cost and time effective manner. UML has been selected to model the system since it provides easy to understand symbols and notations which are easy to analyze according to the design.



### 3.3.1 ER diagram of the system

Figure 3.2 depicts the Entity Relationship(ER) for the system.

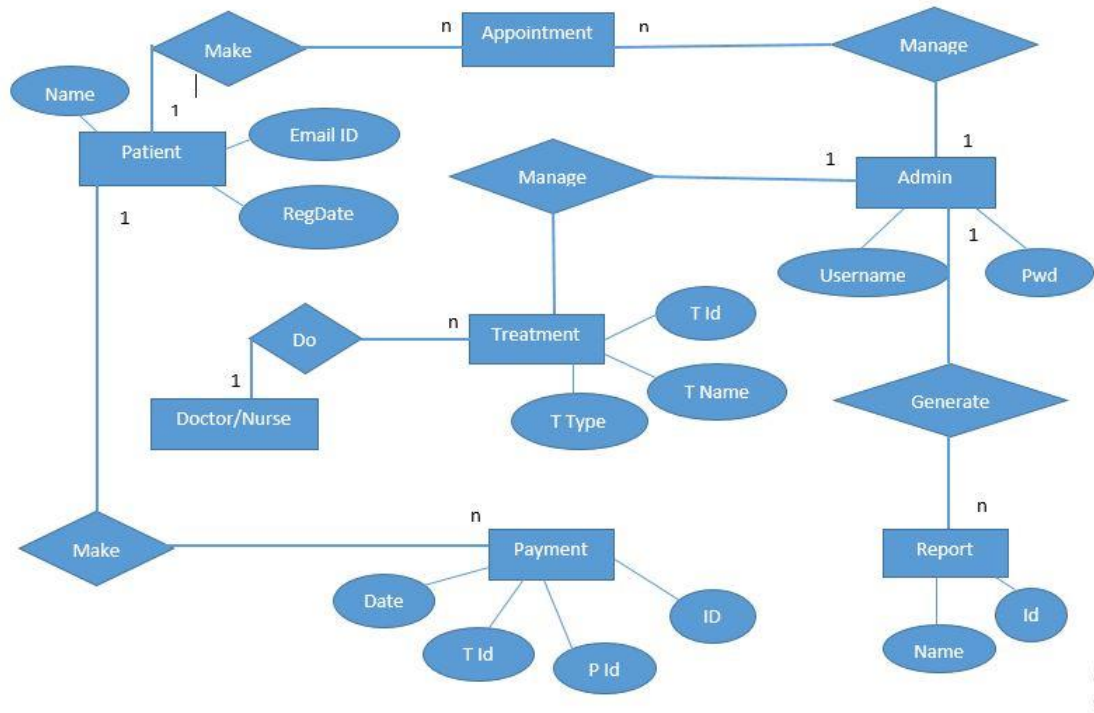


Figure 3.2 ER diagram for the system

### 3.3.2 Use case Diagram of the system

The way that aspectual and simple system user's interaction can be shown by the Use Case Diagram. It is the best diagram to record functional requirements.

Actors (stakeholders) and their functions which are identified in analysis phase, are further identified in this diagram. Use Case diagram which is relevant to the Overall system are represented firstly and after the use Case diagram which is relevant to the individual stakeholders are presented.

Identifying the stakeholders in the system is more critical. Following are the identified stakeholders.

- ❖ Administrator
- ❖ Doctor
- ❖ Nurse
- ❖ Patient

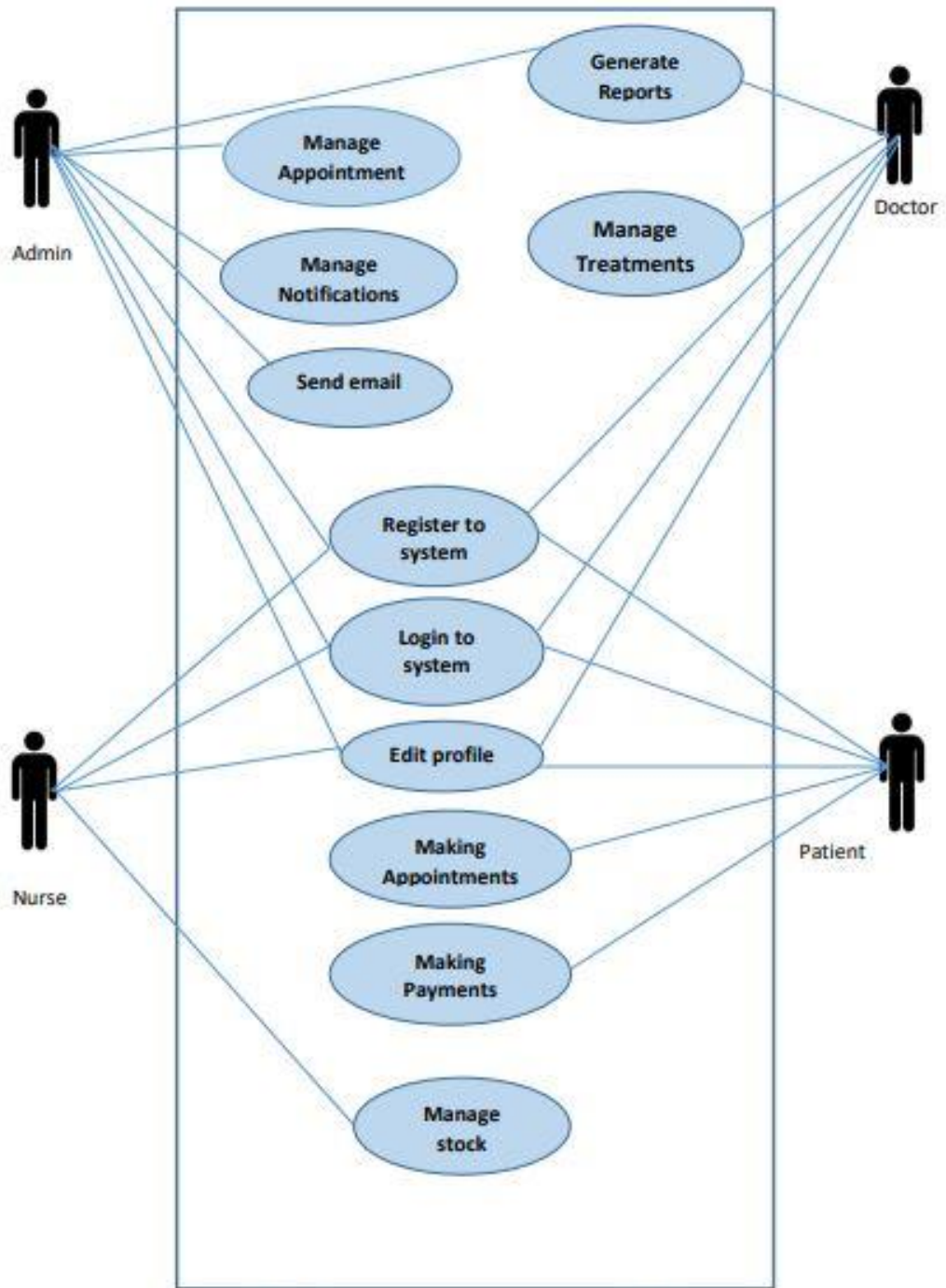


Figure 3.3 Use case diagram for the required system

### 3.3.3 Use case narratives for login to the system

All the users who use this system are required to login to the system. Use Case narrative which relating to these as follows.

Use case	Login to the System
Actors	Administrator, Doctor, Nurse, Patient
Overview	
Registered users will login to the system	
Preconditions	
Users must register with the system	
User should enter correct username and password	
Typical course of events	
<ol style="list-style-type: none"> <li>1. User enters the username and password.</li> <li>2. If an entry is incorrect system will notify error message.</li> <li>3. If an entry is correct, the system redirects the user to the main window.</li> </ol>	
Post conditions	
User will redirect to login page or to main window	

*Table 3.1 Use case narratives for login to the system*

### 3.3.4 Use case narratives for making appointments

All patients should make an appointment before getting treatments. Since they have to use making appointments module in the system. Use Case narrative which relating to these as follows.

Use case	Making appointments
Actors	Patient
Overview	
Patients will make an appointment before getting treatment	
Preconditions	
Patient must login to the system	
Patient should enter required details to the making appointments form	
Typical course of events	
<ol style="list-style-type: none"> <li>1. Patient enters the required details.</li> <li>2. If an entry is incorrect/invalid system will notify error message.</li> <li>3. If all entries are correct, the system make the appointment successfully.</li> </ol>	
Post conditions	
System will notify the appointment has made successfully.	

*Table 3.2 Use case narratives for making appointments*

### 3.3.5 Use case narratives for generating reports

Reports can generate with respect to the patients' treatments, institute services for a particular period, payment calculations for a particular period and staff or patient descriptions.

Use case	Generating reports
Actors	Administrator, Doctor
Overview	
Administrator and Doctor can generate reports	
Preconditions	
Administrator or Doctor should login to the system	
Typical course of events	
<ol style="list-style-type: none"> <li>1. User should select the report which needs to generate</li> <li>2. User should add details for report</li> </ol>	
Post conditions	
System will generate the required report.	

*Table 3.3 Use case narratives for generating reports*

### 3.3.6 Use case narratives for manage treatments

After treating to the patient, doctor should add those details to the system under relevant patient's profile.

Use case	Manage treatments
Actors	Doctor
Overview	
Doctor will enter the treatment details for each treatment	
Preconditions	
Doctor should login to the system	
Typical course of events	
<ol style="list-style-type: none"> <li>1. Select relevant patient from the dropdown in the treatment view</li> <li>2. Add treatment details to the system</li> <li>3. Save treatment details</li> </ol>	
Post conditions	
System will record the new treatment's details for the relevant patient	

*Table 3.4 Use case narratives for manage treatments*

### 3.3.7 Class diagram of the system

Class diagram of the system will describe the structure of the system by indicating classes, attributes, objects and their relationships.

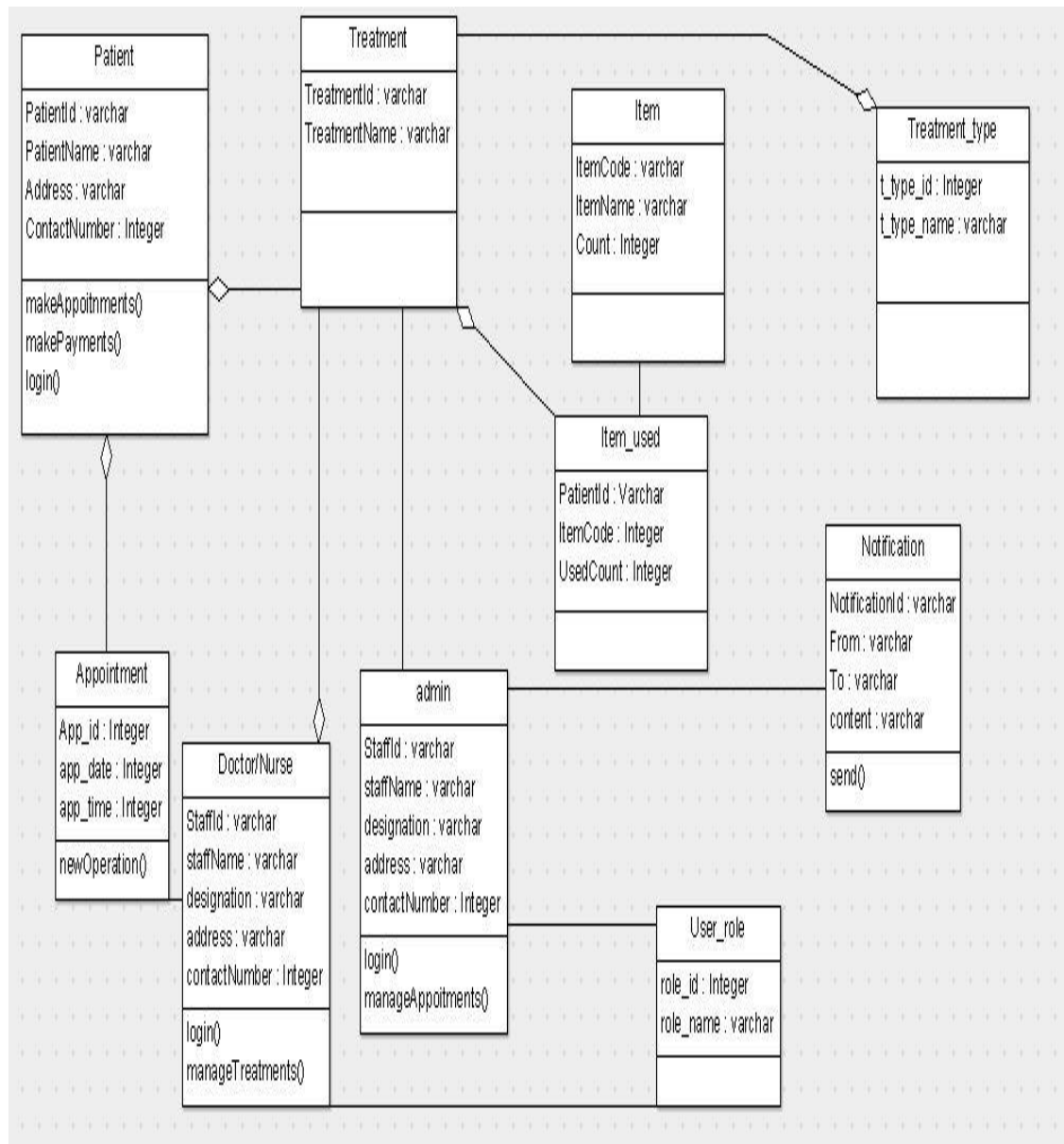


Figure 3.4 class diagram of the system

### 3.3.8 Sequence diagram for the system

Figure 3.5 shows the sequence diagram of the system.

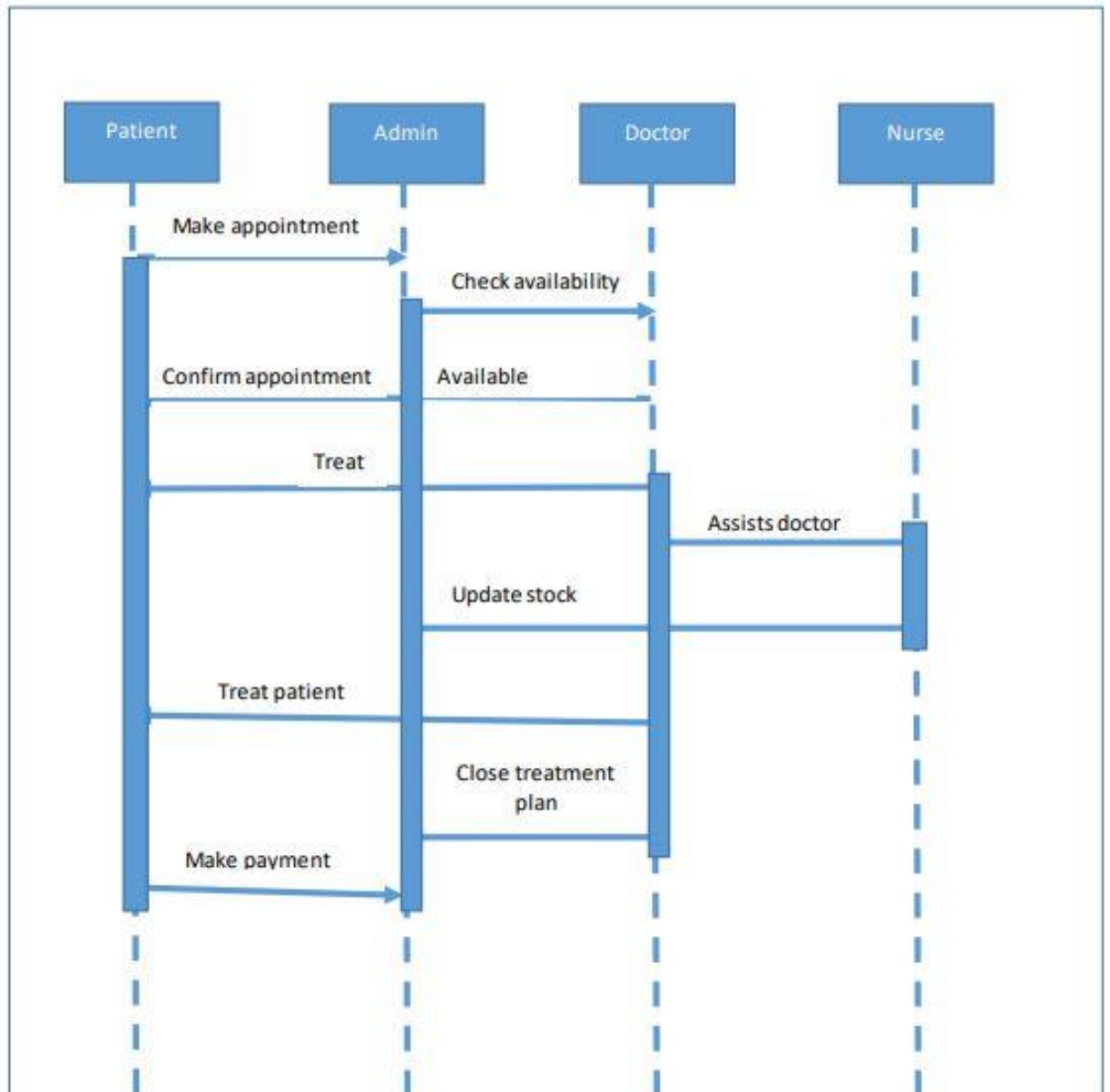


Figure 3.5 Sequence diagram

### 3.3.9 Activity diagram for the user login module

Figure 3.5 shows the sequence diagram of the system

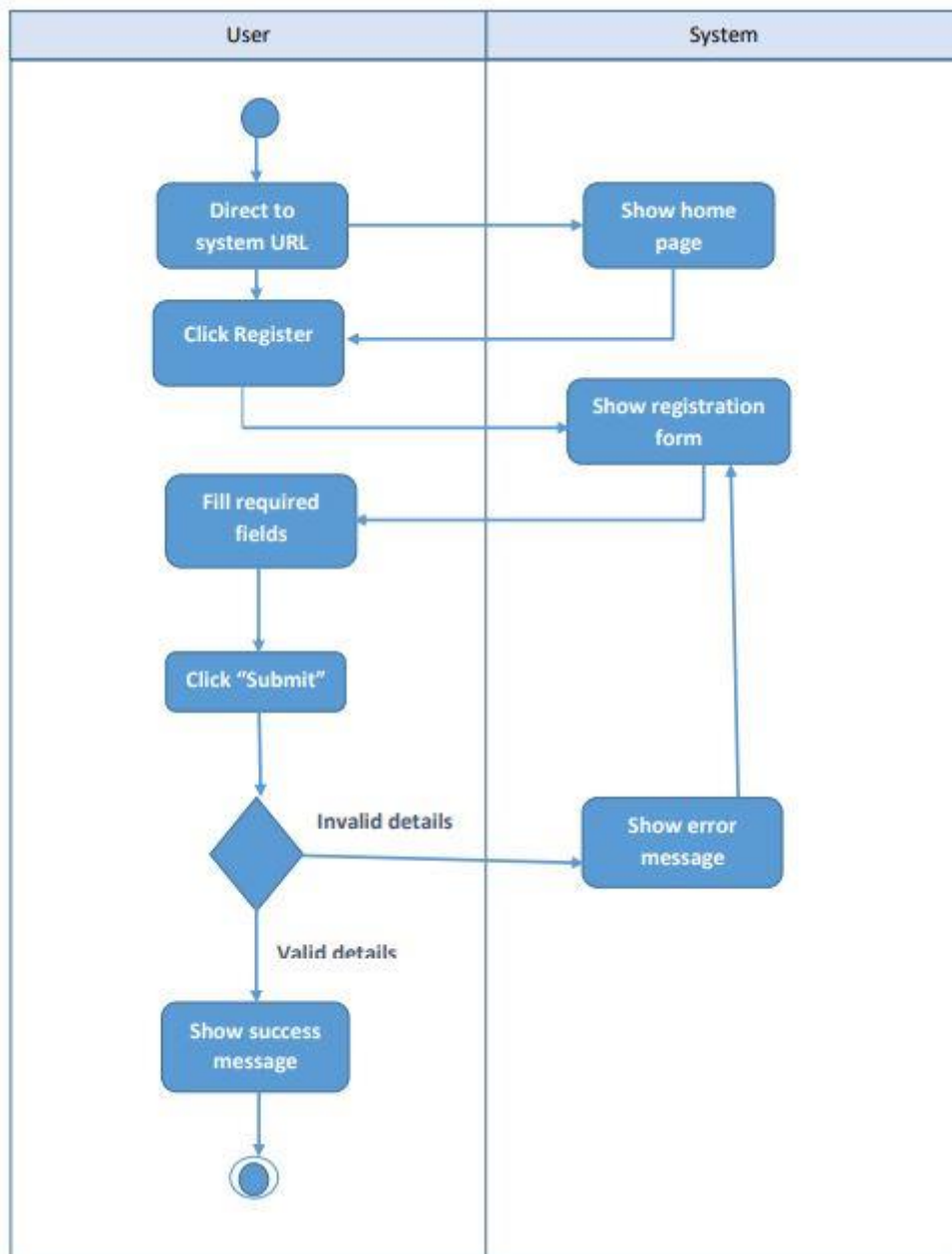


Figure 3.6 Activity diagram for the user login module

Figure 3.7 shows the activity diagram for the making appointments.

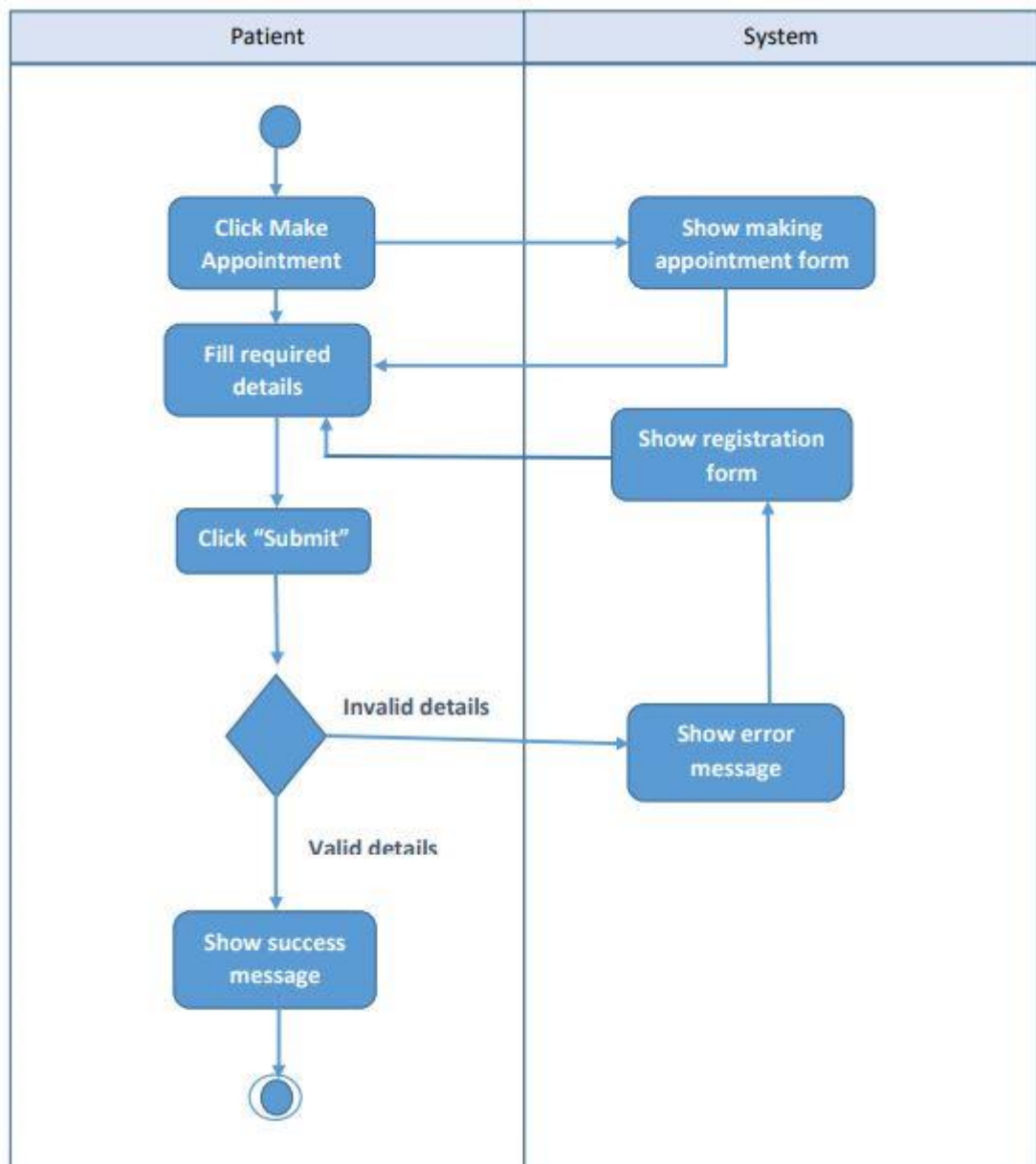


Figure 3.7 Activity diagram for the making appointments

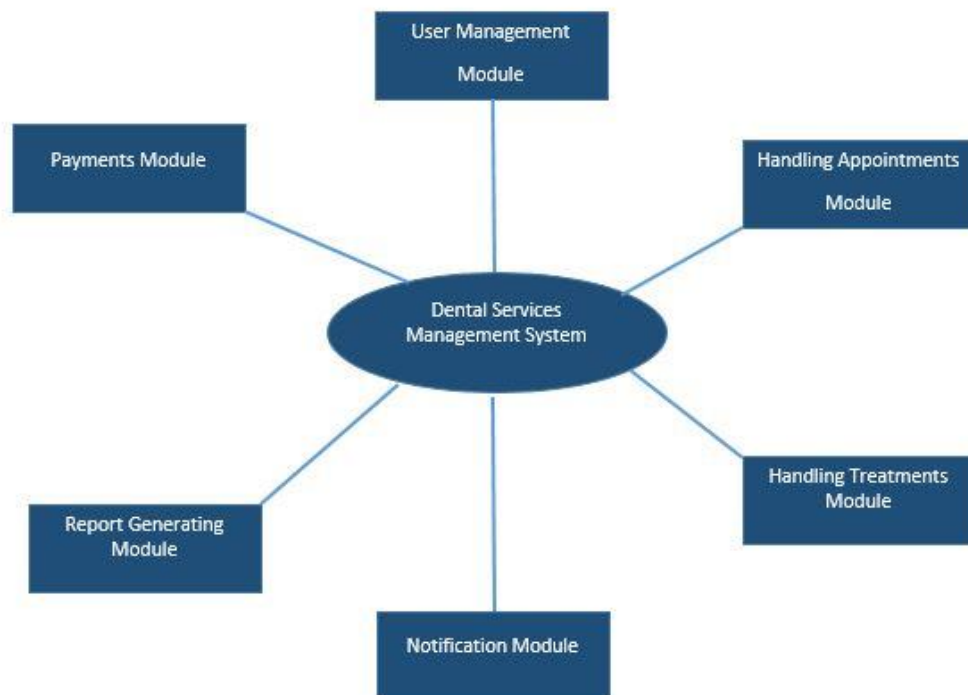


### 3.4 Modules of the system

Dental services management system mainly comprises of six modules. These modules are as follows.

- User management module
- Handling appointments module
- Handling treatments module
- Notification module
- Report generating module
- Payment module

In analysis, identified main requirements are covered by these six modules. Interoperability of these modules is essential when the system accomplishes the required functionality.



*Figure 3.8 Modules of the system*

### 3.5 User interface design

The main source of interaction between the user and the system is the user interface. A good user interface should be well structured and user friendly. User interface design is very user friendly and that means, the user activities can be done easily and simply. Many things have to be considered before designing the user interface. The method they are accustomed to, their job description, number of activities and their levels of intelligence are some major things. The key to this organization is student management and the user interface is designed to make it easier and quicker. [4]

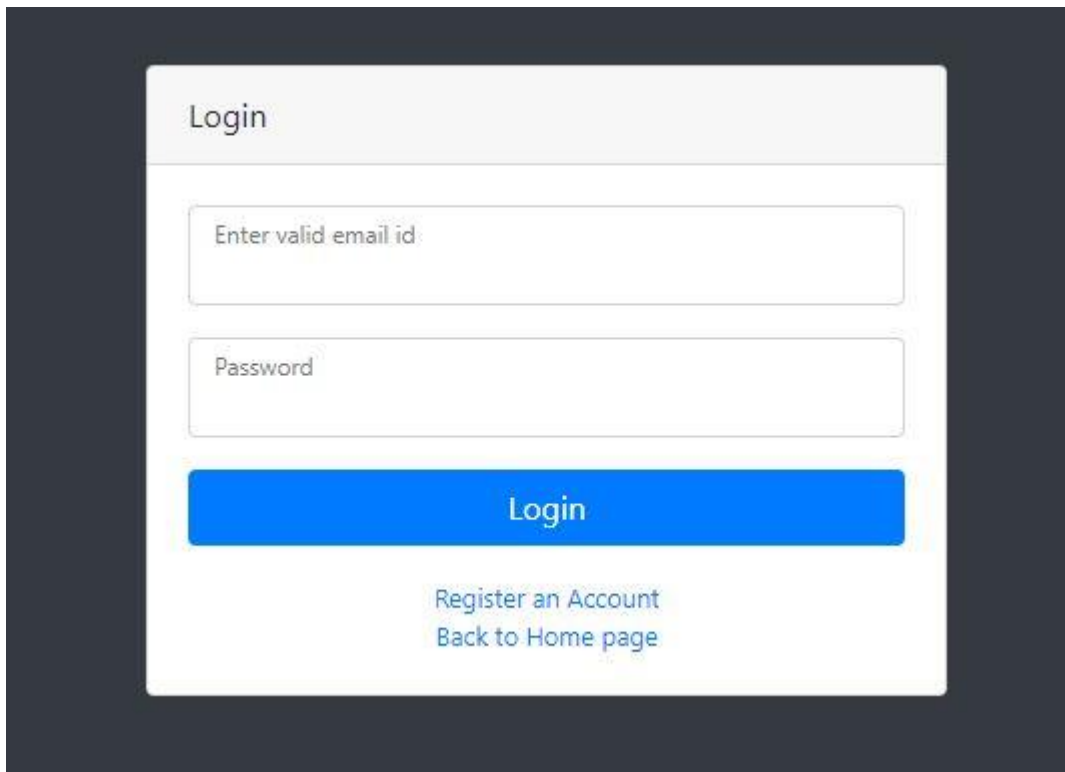
The icons that the user can easily identify when designing the user interface, were used by the system. A consistency theme was maintained by this system. The button behaviour is maintained as consistency.

#### 3.5.1 Home page

Figure 3.4 shows main Home page of the system. It is the main web page which directs users through the website.

#### 3.5.2 Login interface

Figure 3.5 shows the system login interface. All users will login to the system using login interface.



*Figure 3.9 Login view*

### 3.5.3 Admin Dashboard

Figure 3.6 shows the administrator dashboard. It has comprises of header, sidebar, middle container and footer.

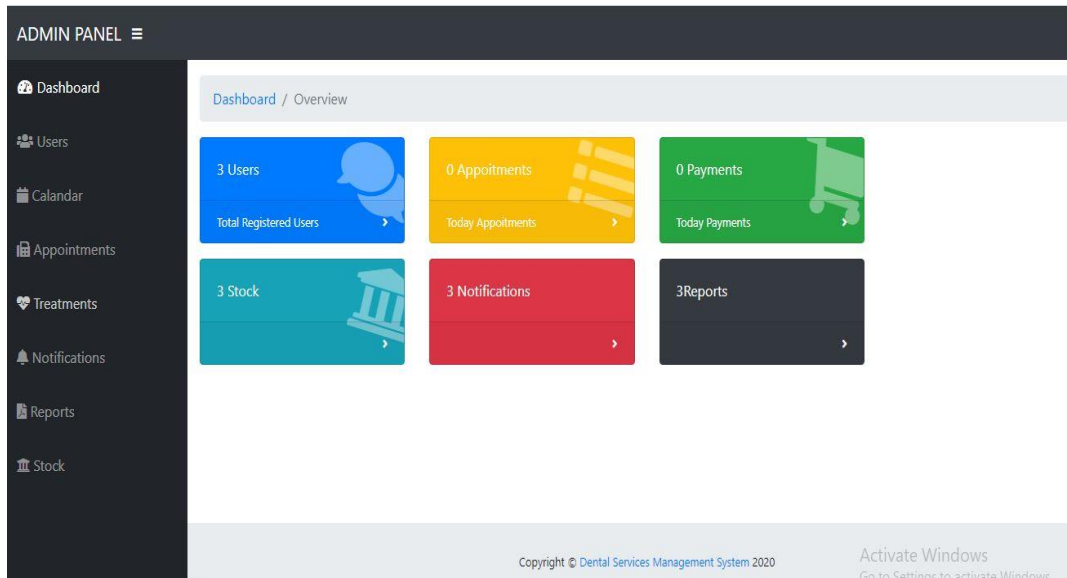


Figure 3.10 Admin dashboard view

### 3.5.4 Patients appointments details view

Dashboard / Patients' Appointments

Appointments Details

Show 10 entries Search:

#	Appointment date	Appointment time	issue	date	action
1	2020-08-03	00:00:10.300000	cravity	2020-05-13 11:10:26	<a href="#">✎</a>
2	2020-09-16	00:00:10.300000	Filling	2020-05-13 11:12:32	<a href="#">✎</a>
3	2020-09-20	00:00:10.300000	Filling	2020-05-13 11:13:15	<a href="#">✎</a>
#	Appointment date	Appointment time	issue	date	action

Showing 1 to 3 of 3 entries

Previous 1 Next

Figure 3.11 Patients appointments details view

## 3.6 – Implementation environment

### 3.6.1 Hardware Environment

- ❖ Operating System – Windows 10

- ❖ Processor - Intel® Core™ i7 2.40GHz
- ❖ Installed Memory – 8GB
- ❖ Hard Disk – 1TB

### 3.6.2 Software and Technologies

The development phase of a system involves a diverse set of tools and software. The tools and software listed below were used in the development phase of this system.

- ❖ XAMPP 3.2.2

This is an open-source cross-platform web server solution package, and this contains the Apache HTTP Server, MySQL database, and interpreters for the PHP and Perl programming languages. This system was developed using the following technologies, PHP 7.3.1 Apache 2.4.37 MySQL 5.0.12

- ❖ PHP (Hypertext Pre-Processor)

It is the most broadly used server-side scripting language that is freely obtainable yet rich with functions. PHP was used to handle the backend processes.

- ❖ MySQL (Structured Query Language)

It is a popularly used open-source relational database management system (RDBMS), and this was used for database related development of this system.

- ❖ HTML (Hyper Text Markup Language)

This was used to create interfaces of the system.

- ❖ CSS (Cascading Style Sheet)

This is a mark-up language used to make the system more attractive to the user. And this was used to make the system more attractive and more user-friendly.

- ❖ JavaScript

This is a widely used scripting language in web-based systems which was used for client-side scripting in this system.

- ❖ JQuery

This is a fast, small, and feature-rich JavaScript library. This was used to make HTML manipulation, event handling and animation of this system.

- ❖ AJAX (Asynchronous JavaScript and XML)

This was used to dynamically retrieve data from the database, avoiding the need of refreshing the page constantly.

- ❖ Debugging

Google Chrome and Mozilla Firefox were used for debugging.

### 3.6.3 Major coding parts

- ❖ Database connection

In Code Igniter framework, there is a separate file called “database.php” for database settings. It has to be completed by including relevant database of this project.

```
$db[ 'default' ] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '123',
    'database' => 'ciproject',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Figure 3.12 Implementation of Database connection

### ❖ User login module

User login module is a core requirement of the system. Since this project is developed using MVC architectural design, login module comprises of three separate files under the Model, View and Controller classes.

Figure 3.13 shows the controller class of the user login module.

```
|<?php
defined('BASEPATH') OR exit('No direct script access allowed');
Class Login extends CI_Controller {

    public function index(){
        $this->form_validation->set_rules('username','Username','required');
        $this->form_validation->set_rules('password','Password','required');
        if($this->form_validation->run()){
            $username=$this->input->post('username');
            $password=$this->input->post('password');
            $this->load->model('Admin_Login_Model');
            $validate=$this->Admin_Login_Model->validatelogin($username,$password);
            if($validate)
            {
                $this->session->set_userdata('adid',$validate);
                return redirect('admin/dashboard');
            } else{
                $this->session->set_flashdata('error', 'Invalid details. Please try again with valid details');
                redirect('admin/login');
            }

        } else {
            $this->load->view('admin/login');
        }
    }

    //function for logout
    public function logout(){
        $this->session->unset_userdata('adid');
        $this->session->sess_destroy();
        return redirect('admin/login');
    }
}
```

Figure 3.13 Controller class of the login module

Figure 3.14 shows the model class of the user login module.

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
Class User_Login_Model extends CI_Model {

public function validatelogin($emailid,$password,$status){

$query=$this->db->where(['emailId'=>$emailid,'userPassword'=>$password]);
$account=$this->db->get('users')->row();
if($account!=NULL){

$dbstatus=$account->isActive;
//$user_role=$account->user_role;

//verifying status
if( $dbstatus==$status){
return $account->id;
} else {
return NULL;
$this->session->set_flashdata('error', 'Your account is not active contact admin');
redirect('user/login');
}
}
return NULL;
}
public function GetRole($emailid,$password){
$query=$this->db->where(['emailId'=>$emailid,'userPassword'=>$password]);
$account=$this->db->get('users')->row();
$role=$account->user_role;
return $role;
}
}
}

```

Figure 3.14 Model class of the user login module

Figure 3.15 shows the view class of the user login module.

```

<?php if ($this->session->flashdata('error')) { ??
<p style="color:red; font-size:18px;" align="center"><?php echo $this->session->flashdata('error');?></p>

<?php } ??
<div class="card-body">
<?php echo form_open('user/login');?>
<div class="form-group">
<div class="form-label-group">
<?php echo form_input(['name'=>'emailid','id'=>'emailid','class'=>'form-control','autofocus'=>'autofocus','value'
=>set_value('emailid')]);?>
<?php echo form_label('Enter valid email id', 'emailid'); ??
<?php echo form_error('emailid',"<div style='color:red'>","</div>");?>
</div>
</div>
<div class="form-group">
<div class="form-label-group">
<?php echo form_password(['name'=>'password','id'=>'password','class'=>'form-control','autofocus'=>'autofocus','
value'=>set_value('password')]);?>
<?php echo form_label('Password', 'password'); ??
<?php echo form_error('password',"<div style='color:red'>","</div>");?>
</div>
</div>

<?php echo form_submit(['name'=>'login','value'=>'Login','class'=>'btn btn-primary btn-block']); ?>
<?php echo form_close(); ??
<div class="text-center">
<a class="d-block small mt-3" href="<?php echo site_url('user/Signup'); ?>">Register an Account</a>
<a class="d-block small" href="<?php echo site_url('Home'); ?>">Back to Home page</a>

```

Figure 3.15 View class of the user login module



## **Chapter 4 – Evaluation**

### 4.1 Introduction

The testing process which used in software development are described by this chapter. The gaps, errors or missing requirements between the client requirement and the actual system are checked by this phase. Testing is the mandatory process in software development. A better system and a client expected system will be given by this. A defects free quality product also will give by this system.

### 4.2 Test strategy

Various testing techniques are used in system development. In developing this system, unit testing, integrated testing and system testing were used. Unit testing and integrated testing are used from the beginning, when developing the system and system testing is the technique used after the system development. Many test cases and test data were used during the testing phase and a detailed description of it is given below. [5]

#### 4.2.1. Unit Testing

Unit testing is one of the simplest and first testing technique found in testing. Unit testing is the software testing technique that is performed on individual unit of source code. The unit tests are included in associated control data, usage procedures and operating procedures tests to verify that the unit is correct. Each unit of the source code perform as designed is the main purpose of the unit testing.

One or more outputs are used in unit testing and out of them one output can be received mostly. Unit testing is used differently in different programming methodologies (e.g. procedural programming). The smallest unit in Object Oriented Programming is a method. It could be a supper class, abstract class or child class.

#### 4.2.2. Integrated Testing

Often integrated testing is done after the unit testing in software development. In unit testing, the individual smallest unit is tested. But here, the smallest units are combined

and bulk of unit are tested. Bugs and errors that many occurs during integrate of units are identified in integrated testing.

#### 4.2.3. System Testing

A test is done in the fully completed system in the process of system testing. Client expected requirement is checked whether it is working properly by the system. This is done after the unit testing and integrated testing. Errors that were not detected in previous phase are also exposed by this testing.

### 4.3 Test cases

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly as it is. [6]

#### 4.3.1 Test case for user login and authentication

Test number	Description	Expected output	Pass/Fail
1	Test using valid username and valid password.	User will login to system	pass
2	Test using valid username and invalid password	Display error message saying "Invalid username or password"	pass
3	Test using invalid username and invalid password	Display error message saying "Invalid username or password"	pass
4	Test using invalid username and valid password	Display error message saying "Invalid username or password"	pass
5	Test using valid username and empty password	Display error message saying "Invalid username or password"	pass

*Table 4.1 Test case for user login and authentication*

#### 4.3.2 Test case for making appointments form

Test number	Description	Expected output	Pass/Fail
1	Test using date ,time and issue	Direct to the view appointments window.	Pass
2	Test using date, time and empty issue	Display error message “Required issue field”	pass
3	Test using date, empty time and empty issue.	Display error messages “Required time field” , “Required issue field”	pass

*Table 4.2 Test case for making appointments form*

#### 4.3.3 Test case for add new treatment

Test number	Description	Expected output	Pass/Fail
1	Test by clicking add new treatment button	Add treatment form will appear	Pass
2	Fill required fields with valid inputs and click “Submit” button	Message will appears that says “Treatment is added successfully”	Pass
3	Fill required fields with invalid inputs and click “Submit” button	Message will appears that says “Invalid input values”	Pass

*Table 4.3 Test case for add new treatment*

## **Chapter 5 - Conclusion**

### **5.1 Critical evaluation of the project**

Dental Management system has come to the successful point with facing different kind of challenges and solving all of them. New skills and knowledge gained throughout the period assisted me to face the challenges and solve all the problems.

Since this project conducted to fulfil the requirements of all types of dental services institutes, system requirements have gathered from several institutes and some instructions has taken from the expert of the domain. By considering their similarities and variety system functionalities have designed and implemented. Configuration module handle the generalized nature of the system. In the end, all of major requirements have accomplished through the system. User management, Handling appointments, handling treatments, generating reports, sending notifications, managing stock are the other main functions of the system. Drawbacks in manual operational environments have addressed the system.

Critical requirements provided by the institutions were transformed into the functionality of the Maintenance management system. As per the feedback of them, the system can be distinguished as a successful system.

### **5.2 Lessons Learnt**

As a postgraduate student, this was a great opportunity for me to apply the previously learned lessons into a practical environment. The domain area was familiar for me, therefore it helped to gather business requirements easily and it broadened my horizons into understanding, how to map those related business processes into a computerized system.

As a postgraduate the knowledge gained throughout the project was really valuable. In addition, this gave me an exceptional experience of being in a complete software development life cycle, starting from feasibility studies to conclusion of the project. This project gave an opportunity to get extensive knowledge on Code igniter, PHP, MVC, MySQL, and many more languages, tools and technologies. And also, it helped to test and implement most important theories and technologies learnt throughout the MIT degree program.

### 5.3 Future work

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

- Develop a mobile application which can be used by patients and doctors directly.
- Enhance the making appointment facility to unregistered users.
- Develop the system that can be used for oral surgical treatments.

\*\*\*\*\*

## References

- [1] Nawaloka Hospital website [Online]. Available: <https://www.nawaloka.com>.
- [2] Srilanka Dental Association [Online]. Available: <http://www.slda.lk/>
- [3] Dental unit of Durdans hospital [Online]. Available:  
<https://www.durdans.com/dental-care/>
- [4] MVC architecture pattern [Online]. Available: [5] User interface design  
[Online]. Available:  
[https://www.tutorialspoint.com/software\\_engineering/software\\_user\\_interface\\_design.htm](https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm)
- [5] Test strategy [Online]. Available:  
[https://www.tutorialspoint.com/software\\_testing\\_dictionary/test\\_strategy.htm](https://www.tutorialspoint.com/software_testing_dictionary/test_strategy.htm)
- [6] Test cases [Online]. Available:  
[https://www.tutorialspoint.com/software\\_testing\\_dictionary/test\\_case.htm](https://www.tutorialspoint.com/software_testing_dictionary/test_case.htm)