Clinical Management System

For - **Kenway Medicals (Pvt) Ltd.**

Candidate Name: R.S.Gunawardhana

BCS Membership Number: 990636231

Abstract

Kenway Medicals (Pvt) Ltd is a company which is operating in Sri Lanka. It is a medical company that has sub-divisions across the country. Its' main idea is to provide clinical services to patients island-wide. It consists of a channeling department which is responsible for doctor appointments and a medicinal department responsible for the issuing of pharmaceutical drugs.

A notable problem in the Channeling Department is that the core tasks of the company, which are handling doctor appointments, scheduling doctor appointments are all carried on manually. Tracking such tasks on paper, by the staff members in the institution has created a lot of confusion among the people involved in the process, i.e. doctors, patients, staff members. It proved to be inefficient and arduous task.

Failing to maintain these records carefully, will earn the company nothing but even more distressed people. Considering the consequences of the problem on the institution and the people involved in the service process, it was proposed to them to transition into a web based system that would establish the processes in the institution more efficiently.

The main motivation for carrying out this project is to standardize the business process ensuring the efficiency and to fill the much needed communication gap between the doctors, patients, administrators and staff and to streamline the interactions and business processes of Kenway Medicals (Pvt) Ltd. Hence, it was proposed to develop a web based solution for Kenway Medicals, that would fulfill a specific number of functions including Patients' Appointment Handling, Appointments Scheduling for doctors and User management while covering a scope defined in more detail, which will all be discussed extensively in the chapters that follow.

Acknowledgement

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 reality.

First and foremost I owe my deep gratitude to the British Computer Society – The Chartered Institute for IT for offering us this prestigious qualification. I'd also like to thanks all members of the staff at ESOFT Metro Campus who guided us from the beginning.
A very special recognition needs to be given to my project supervisor Ms. Dilshara Weerasinghe for his/her
extensive assistance, without whom the completion of this project would have been extremely complicated.
I must thank, the Managing Director at Kenway Medicals (Pvt) Ltd for their willingness and patience in supporting this project from the beginning to the very end.

And finally, I'd like to thank my every supporting family and friends for their unmatched encouragement and support throughout the BCS HEQ program.

Table of Contents

List of Figures

List of Tables

1. Chapter 1 - Introduction

1.1 Organizational Background

Kenway Medicals (Pvt) Ltd is a company which is operating in Sri Lanka. It is a medical company that has sub-divisions across the country. Its' main idea is to provide clinical services to patients island-wide. It consists of a channeling department which is responsible for doctor appointments and a medicinal department responsible for the issuing of pharmaceutical drugs.

1.2 Motivation for the project

The main motivation for carrying out this project is to fill the much needed communication gap between the doctors, patients, administrators and staff and to streamline the interactions and business processes of Kenway Medicals (Pvt) Ltd. The company has been maintaining a file based system for a number of years which proved to be inefficient and arduous task. In order to overcome the difficulty in maintaining manual records, and all the cascading problems associated with it, it is essential to move the business process into the digital platform. Hence, it was proposed to develop a web based solution for Kenway Medicals that would fulfill a specific number of functions covering a well defined scope, which will all be discussed extensively in the chapters that follow.

1.3 Objectives of the project

The system proposed, is intended as a solution which fulfills the following major objectives.

- Connecting the Patients, Doctors, Staff and Administrators to allow real-time processing of data.
- A central monitoring system that administrators can use to monitor island-wide branches.
- Implementing a user management system to manage all the patients and doctor profiles through an interface provided to staff members.
- Providing a real-time schedule for appointments and sessions.

1.4 Scope of the project

An in-depth analysis of requirements on this system lead to the following improved version of scope. The improvements made from the scope that was proposed for this project is assed extensively in this section. Figure 1 below represents a diagrammatic overview of the improved scope of the Clinical Management System.

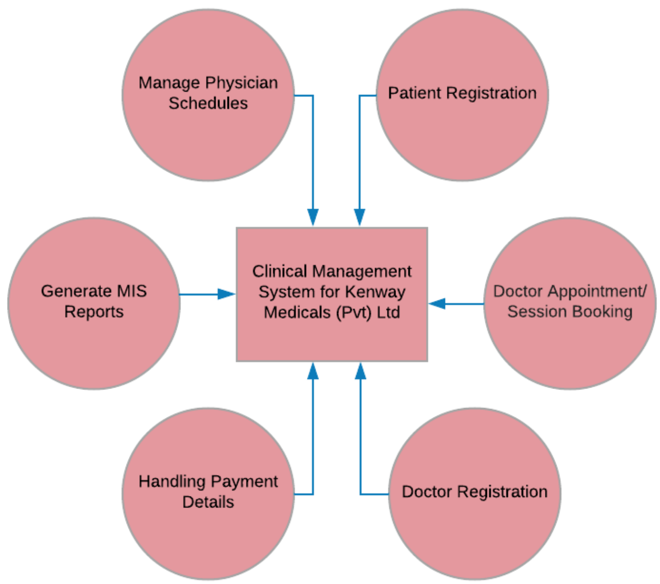


Figure 1 - System Scope

The modules proposed in the project proposal have been rearranged into a more comprehensive structure as shown in Figure 1. The major differences from the modules proposed in the project proposal and the improved scope are as follows.

A new module has been introduced to produce **Management Information System Reports** which is vital for strategic decision making of top level management of the Clinic.

A **Payment Details Handling** module was defined along with a redefinition of the tasks that it performed. As per client's request, it was decided that the payments should be carried out on the clinic premises and not online. (find the detailed assessment of this change discussed in Chapter 7 – Self Assessment.) However, some aspects of payment handling will be carried out through the system (such as tracking the payment received by patients per appointment) and they will act as a part of this this module.

Doctor and Patient Registration and management of their data are defined as modules of their own due to the difference in these users' registration processes from conventional systems.

It was identified that tasks that involve **Booking Sessions or Doctor Appointments** needed a module of their own to pay thorough attention to their development specifications.

Management of Doctor/ Physician Schedules was defined as a separate module as this module consists of functionalities that requires attention of its own.

NOTE: Another important definer of scope is that the new system on its first release would focus on systemizing the service process of main branch of the Clinic. It was agreed that the system could be extended to other branches if it proves to carry out its intended tasks successfully.

1.5 Structure of Dissertation

After the Introduction chapter, the dissertation contains six main chapters. The Analysis chapter describes the existing system, requirements and feasibility study for the proposed system and the process model used to develop the system.
The Design chapter includes some diagrams relating to the system design. It further describes the designing approach used in designing of the database structure as well as the user interfaces.
The main codes, functions used in the system and the database design are described in the System development chapter.
The software and hardware environment used for implementing the system are described in the Implementation chapter
The test plan and test cases of the system are given in the Evaluation chapter. 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, User Documentation, Test Results and Code Listing are provided.

2. Chapter 2 - Requirement Analysis & Specification

2.1 Introduction

The problems that software engineers have to solve are often immensely complex. Understanding the nature of the problems can be very difficult, especially if the system is new. Consequently, it is difficult to establish exactly what the system should do. The descriptions of the services and constraints are the requirements for the system and the process of finding out, analyzing, documenting and checking these services and constraints is called Requirements Engineering.

The proposed system for the Clinical management system is to be developed using PHP and Web based technologies. The purpose is to make the process of manual handling much easier. Before the requirements were finalized, the developer had first discussed with selected users about the scope and requirements.

2.2 Current System

The existing system consists of a file based approach. It uses manual file systems to record the personal details of Doctors and patients/Store medical details and records such as clinic times etc. Following are the functions performed by the existing system.

- Storing personal information of patients and doctors (Every division has a file consisting of patients and doctors of their branch)
- Logging clinic time periods (Doctor schedules) for each doctor
- Registering patient appointments for a doctor (clinical session).
- Payment logging of facility fees at the counter. This has to be manually written down in a file based system for each individual patient and appointment.

2.3 Problems identified in the existing system

- It is quite difficult to centrally monitor all the activity taking place within the facility by an administrator
- Data redundancy can occur due to the traditional file systems. Also the patients and doctors sometimes face quite a lot of inconveniences due to missing or misplaced files and records.
- The patients have no means of checking the availability of doctors before arriving at the facility. They have to either come in and check the available doctors and check their schedules or make a call to the office and check.
- Extensive procedures for cancelling and making appointments have to be made because to make or cancel an appointment the patient has to either come to the facility and do that or they have to call the office.

2.4 Feasibility Study

For project of this type, that is a project that handles sensitive user data from a domain that should be handled with much attention to detail and care, its vital that we assess and address the legal, ethical, social and professional issues that involve with the development of this system. Feasibility study that was outlined during the requirement analysis phase is given below. The assessment of how any issues that was identified through this process will be given in Chapter 7 - Self Assessment section.

Operational and Social

It was taken into consideration whether the current work practices and procedures support the new system and social factors of how the organizational changes will affect the working lives of those affected by the system. The staff and management of Kenway Medicals are a group of people who are very open and willing to adapt to change. However, it is expected that the patients may all not be the same. In order to get the non-technical users to use the developed system, much care will be taken to ensure simpler system design and thorough user training and support.

Legal

A project may face legal issues after completion if this factor is not considered at this stage. It was analyzed whether the proposed system conflicts with legal requirements e.g. as a system handling a lot of sensitive user data, it must comply with the local Data Protection Acts. As per the analysis, it was found that the proposed system does not conflict with any local legalities confirming the legal feasibility of the Clinical Management System.

Technical

The technical aspect explores if the project feasibility is within the limits of current technology and does the technology exist at all, or if it is available within given resource constraints. At Kenway Medicals, technology required for the implementation of the system are already available confirming the feasibility of the project from technical perspective.

Ethical

This is one of the most easily overlooked feasibility criteria. Which is the same reason why a lot of seemingly great systems face shattering consequences as a result of failing to conform to the ethics of the profession, the client company as well as ethics in general. It is vital that that we ensure the security and confidentiality of the data a system requests from users and maintained thereafter. In order to build the trust between the user and the system, meticulous planning was made starting from proposing a friendly look and feel for the system to adopting HTTPS for the system.

2.5 Functional Requirements

These are compulsory features that must be included within the project as identified by the client. We were able to identify several key functional modules required by the client.

Patient registration

A patient wishing to obtain the services of Kenway must first register with the system by providing personal details such as Full name, Date of birth, Address, Telephone number and email. This enables the organization to keep track of their customer base and also to inform patients about doctor arrivals (via SMS) and sudden unexpected changes such as the unplanned visit cancellation of a doctor.

Appointment/Session booking

One of the major requirements of this system is for patients to be able to book an appointment online with a doctor that the patient wishes to consult. All appointments/sessions are assigned a unique code and an appointment number along with date and time. The appointment numbers indicates the order in which the patients would be seen by a given doctor. They're issued in a first come first served basis.

Each doctor has a limit on a number of appointments per day. This fact must be taken into consideration prior to booking an appointment. The number of patients to be seen by a doctor per day is called "slots" and if all the slots are already booked for a given day then the system should allow the patient to make a booking on an alternative day.

Those who book appointments online should remember the appointment code provided at the time of booking the appointment. When they visit the facility for actual consultation, they should produce this code at the reception.

It is also possible for a patient to book an appointment by visiting the facility. His or her requirements are presented to any staff member at the reception and the clerk will access the system to book an appointment. Upon successful booking of an appointment, a staff member will provide the patient with their appointment number. A flat fee of Rs.300 is charged by the facility for providing the appointment. The doctor's fee is charged separately by a nurse at the time of consultation. This system will record the appointment fee only, doctor's fee is outside the scope of this system.

Patients who booked an appointment successfully with a desired doctor should receive an SMS reminder 1 hour before the appointment.

Doctor Registration

The doctors who wish to serve the facility first have to inquire the receptionist about their registration. After a few verifications the respective staff member will get the details of the doctor and create a separate account for them via the Staff interface. Every registered doctor is given the current system's schedule for all the appointments within the month. This ensures that the doctors are aware of the current workings and helps them to adjust their sessions for the month.

Manage the doctor/physician schedules

Ability for each individual doctor/physician to manage their own schedule times and number of patients per session. Allows each physician to see the patients registered for their each individual session.

Payment Handling

When a patient has appointed a doctor, the databases are updated to show their due fee to the facility. The payment is currently handled at the office. A patient who has appointed a doctor is required to give their appointment number, appointed doctors name and the time of appointment at the office to confirm their appointment and to pay the facility fee. A staff member is able to handle this by taking the required amount of cash from the patient and update the database via the interface provided to each staff member at the office. After the payment has been made, an SMS is sent out to the patient using the mobile number they have provided at the registration saying that the payment has been registered within the system.

Production of Management Information reports

The system should produce necessary reports to monitor the performance of the channeling facility. It should produce an income report that shows the number of appointments and the income generated by them. This should be grouped into doctor order. The system should also be able to produce a list of registered patients for a given time period. It should also provide a report regarding the current clinical sessions and the number of registered patients within the system.

2.6 Non-Functional Requirements

Accuracy and Consistency

These are very important non functional requirements that should be considered when storing data belonging to all users of the system. For this, the databases should be properly designed and normalized.

Security

There should be superior security mechanisms since the system stores very confidential information such as personal information of patients and doctors.

Usability

Usability should be very much important in this project since majority of the users of this system are non-technical people. i.e. patients, physicians as well as the other users of the system.

Reliability

This is a non-functional requirement of the system users. There should be trustworthiness between the users and the system.

Reusability and Maintainability

In a case where the system needs any changes in the future, it should not be a tricky task. Proper documentation and using standard methods when developing the system will ensure this non-functional requirement.

2.7 Similar systems

In this section several features of similar web applications are compared to differentiate and select better non-functional requirements for the new system.

The comparison is carried out between 2 different companies offering similar services:

1. Nawaloka Hospitals – E Channeling
   http://www.nawaloka.com/channeling (Nawaloka Hospitals PLC | Sri Lanka, 2018)

2. Hemas Hospitals – E Channeling
   https://doctors.hemashospitals.com/channel (Hemas Hospitals | Hemas Channel Online , 2018)

Feature	Nawaloka Hospitals	Hemas Hospitals	Kenway Medicals (New system)
User Interface	A simple user interface is provided to appoint a doctor. It doesn't provide any channeling sessions for the particular doctor, the time has to be manually arranged	More desirable interface is provided which has a doctor search facility to either select a doctor by name or specialty. The time period of each appointment for each doctor is also provided.	A simple interface featuring most of the bullets in Hemas Hospitals UI is to be implemented including doctor name, appointment times and adding additional points such as Number of patients taken per session and number of currently registered patients.
User friendliness	This has a quite simple interface but the number of provided information regarding booking a session is quite low. Therefore manual handling like calling customer support has to be done to get inquiries regarding the sessions	The UI provides sufficient information regarding the sessions and doctors but navigating the interface is quite complex. It is quite tedious for the regular person	UI provides all the necessary information regarding the Sessions, doctors and date/time. And navigating it is quite easily accomplished since there's no complex routines used when managing, booking sessions.
Automation	The Automation feature is quite low in this since it requires several manual handling when appointing, booking and paying for the session. This all has to be done via customer support through an agent.	Automation is adequate since only the payment has to be managed by the customer support.	Has the same automation level as Hemas Hospitals but is planned to improve to provide payment automation through a payment gateway such as PayPal, Skrill or Bank Account in later parts of system development iterations
UI Responsiveness (UI Build Time)	The responsiveness is quite low considering there're not many features available in the interface. The time taken to build the interface is quite high relative to lesser	Responsiveness is considerable since the UI has many features. The search feature however consumes much time to output a result. A proper	Build time is relatively high compared to other systems plus this consists of same level of features as the Hemas Hospitals management system.

	components available	assumption for that reason might be poorly coded search algorithm	

Table 1 - Similar Systems

2.8 Requirement Analysis

Within the UML, one of the early steps involves building a Use Case Model. The essence of this model is to capture user requirements of a new system, whether it is being developed from scratch or based on an existing system, by detailing all the scenarios that users will be performing. (Aisel.aisnet.org , 2018)

The overall use case for the system is provided below in Figure 2. An extensive collection of Use case descriptions can be found in Appendix C – Design Documentation.

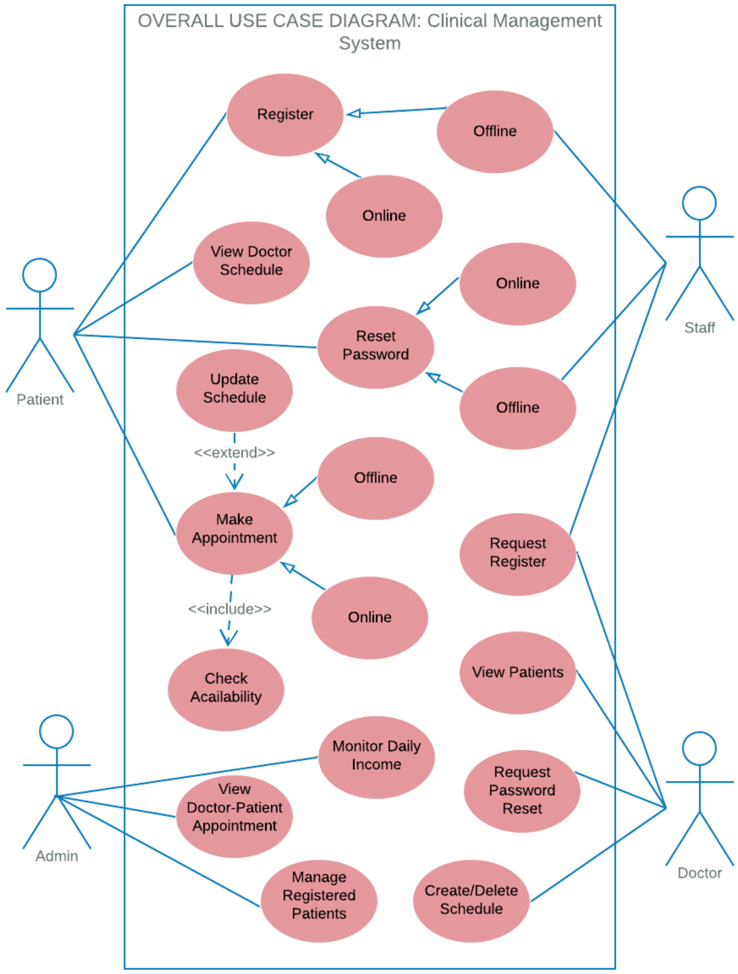


Figure 2 - Overall Use Case Diagram for the system

Summary

The functional requirements of the system is identified while the developer was inquiring the in depth level details with the client. These are the must-have features of the new system.

A feature comparison is done with several other similar service providers to determine and adapt the proper user interface and non-functional requirements. The good features of each comparison are taken into consideration while developing the user interface and most of the non-functional requirements.

3. Chapter 3 - Design

3.1 Introduction

This chapter mainly concentrates on how the system will be designed & developed. Many design concepts are inherited, such as Modularity, Data structures, Encapsulation etc. After close examination with the system the design model accepted by the developer was the Incremental Design Model.

3.2 Process Model

In an Iterative Incremental model, initially, a partial implementation of a total system is constructed so that it will be in a deliverable state. Increased functionality is added. Defects, if any, from the prior delivery are fixed and the working product is delivered. The process is repeated until the entire product development is completed. The repetitions of these processes are called iterations. At the end of every iteration, a product increment is delivered. (SDLC - Iterative incremental model)

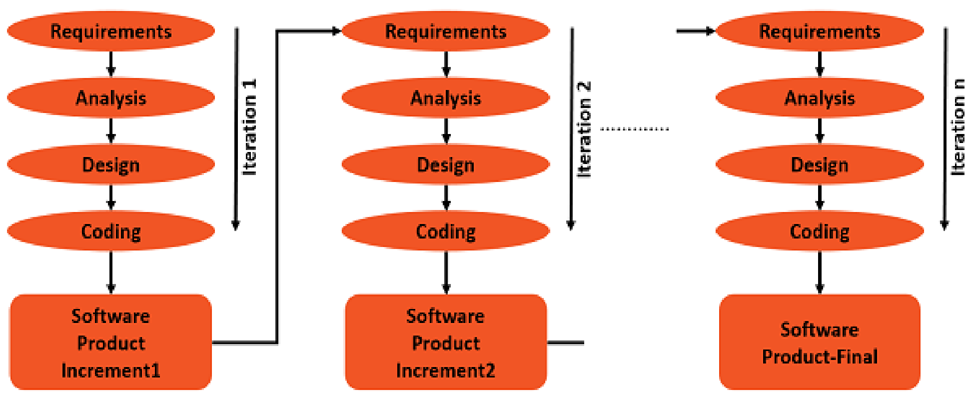


Table 2 - Iterative Incremental model

Why Iterative and Incremental Model?

- Nature of this methodology facilitates documentation of the development of the system.
- Stakeholders can be given concrete evidence of project status throughout the life cycle due to the incremental delivery.
- Gradual implementation provides the ability to monitor the effect of incremental changes, isolate issues and make adjustments before the organization is negatively impacted.

3.3 Database Design

A well-designed database gives users access to essential information. By the use of diagrammatic tools such as an entity relationship diagram (ERD), also known as an entity relationship model, you can design a database that performs well and adapts to future needs. (Xu, 2016)

An ERD is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. (What is entity relationship diagram (ERD)? - Definition from WhatIs.com)

The ERD for the Clinical Management System is given below.

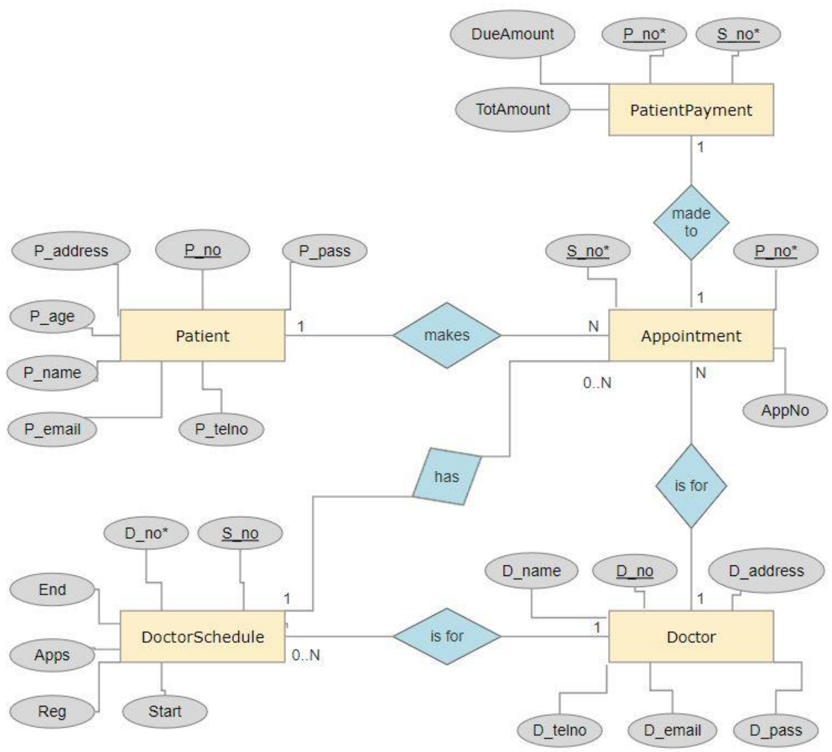


Figure 3 - Entity Relationship Diagram

3.4 Class Diagram

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. UML Class diagram is one of the popular methods among software engineers to document software architecture. Class diagrams are a type of structure diagram because they describe what must be present in the system being modeled. ("UML Class Diagram Tutorial", 2018)

The various components in a class diagram can represent the classes that will actually be programmed, the main objects, or the interactions between classes and objects. They enable the system designers, and developers to better understand the general overview of the schematics of the system. Also, it visually expresses any specific needs of a system and disseminates that information throughout the development process.

Figure 4 given below depicts the class diagram for the Clinical Management System of Kenway Medicals (Pvt) Ltd.

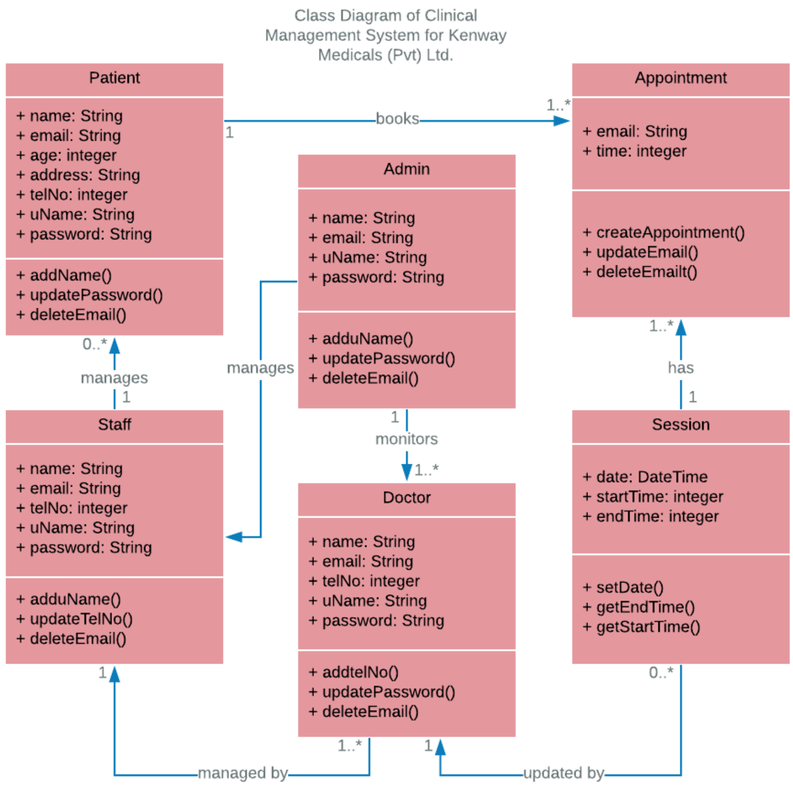


Figure 4 - Class Diagram for the System

3.5 Design Diagrams

This section focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions.

User interface design is a craft that involves building an essential part of the user experience; users are very swift to judge designs on usability and likeability. Therefore, its important to focus on building interfaces users will find *highly usable* and *efficient*. Thus, a thorough understanding of the contexts users will find themselves in when making those judgments is crucial. ("What is User Interface (UI) Design?", 2018)

The interfaces that has been created should allow users to attain goals *directly* and as effortlessly as possible. Following user-interfaces illustrates how the user interacts with the system. User friendliness and system flexibility are greatly exaggerated by the fine use of web base interface and appearance.

Copy-left images from sites like pixabay.com and unsplash.com were downloaded and edited. (Stunning Free Images · Pixabay , 2018) (Beautiful Free Images & Pictures | Unsplash, 2018)

Some of the main interfaces of the system with design definitions are given below. Find further User interfaces in Appendix B – User Documentation.

User Login

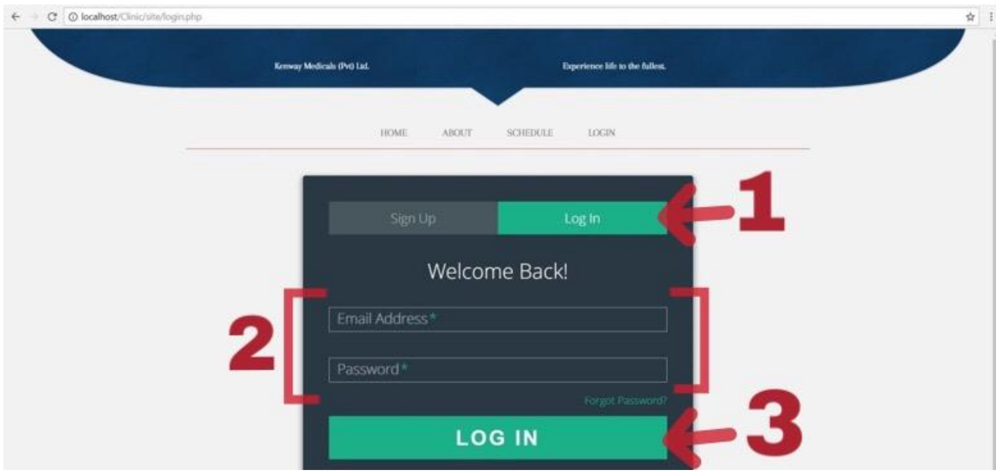


Figure 5 – Login

1. In order to log in to the system, a user can first click on Login from the navigation bar. Then the user will be directed to the login page as shown in the above figure.
2. This section consists of fields that requires the user login information. Here, the user must provide the email as the username and the respective password that's assigned for the email at registration.
   Note: * represent the mandatory fields. Mandatory fields must be provided correctly to login into the system. Incorrect information inputs will yield error message.
3. Finally, click Log In to log in to the system.

Staff – Register Doctors

A logged in 'staff' user would be able to handle a Doctor registration using this interface.

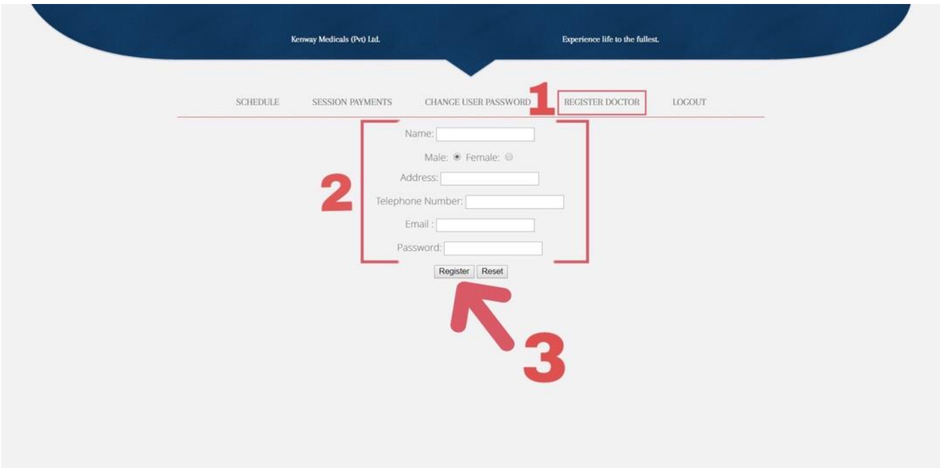


Figure 6 - Register Doctors

1. By selecting "Register Doctor" as shown by number 1 in the figure above, the staff member will be directed to the doctor registration view.
2. Next, in order to register with the system, the information that requested by the form must be filled in.

   Note: * represent the mandatory fields. Mandatory fields must be provided to register into the system.

3. Finally, Selecting the Register button or pressing Enter key on the keyboard will complete the Doctor registration into the system.

Patient – Make / Cancel appointment

A user logged in as a 'patient' would be able to see this page and book/cancel an appointment with a desired doctor in a given session.

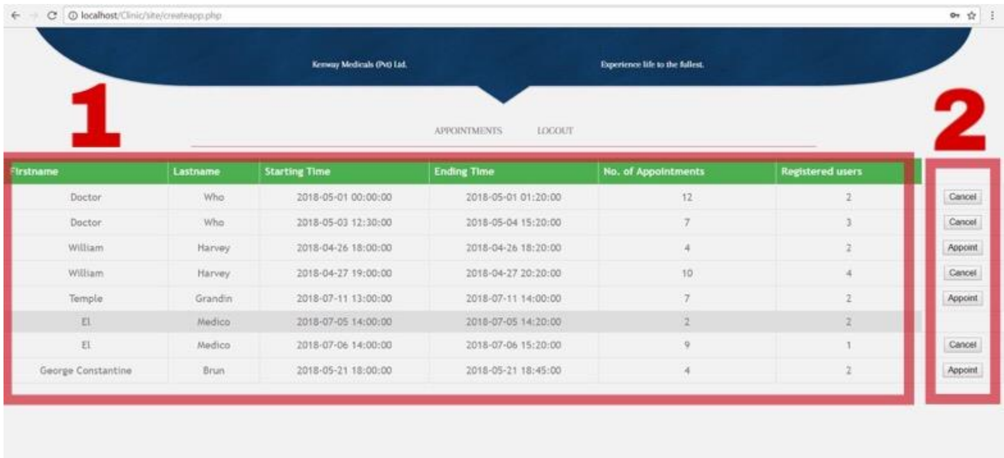


Figure 7 -Doctor Appointments

1. Once a patient is logged in to the system, the user can select Appointments tab from the menu bar. This will display a list of Doctor Appointments as shown by number 1 in the above interface.

2. From the doctor appointment details listed as shown in figure 8 above, the user can select a doctor who has slots that are still available for an appointment. The patient can place an appointment with the doctor by clicking Appoint button on the respective doctor appointment record. Any appointments that are placed for the wrong doctor can be removed by clicking the Cancel button.

(For extensive User Interface design refer Appendix A – User Documentation)

3. Chapter 4 - System Development

3.1 Database Architecture

The system uses a MySQL Database server. It has been chosen because of the high stability, compatibility with many platforms and the usage easiness. And it is also freely available, easy to learn and integrate into a PHP based web application.

Database of this system consists of five major relations. The relations and the respective data that can be held by these relations are as given below.

- Appointments (S_no, P_no*, Appno)
- Patient (P_no, P_name, P_address, P_age, P_telno, P_email, P_pass)
- Doctor (D_no, D_name, D_address, D_telno, D_email, D_pass)
- Doctor Schedule (S_no, D_no*, start, end, apps, reg)
- PatientPayment (P_no, S_no*, Totamount, Dueamount)

1. Appointments

This table contains all the appointment details of each individual customer with a doctor's session. S_no field in the table are uniquely identified.

Appointments (S_no, P_no*, Appno)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	S_no	int(255)			No	None			Change
2	P_no	int(255)			No	None			Change
3	Appno	int(255)			No	None			Change

Figure 8 - Appointments Table

2. Patient

This table stores all the user credentials relevant to the Account registration of a patient. Here the field 'cno' is incremental and is uniquely identified.

Customer (P_no, P_name, P_address, P_age, P_telno, P_email, P_pass)

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
	1	**P_no**	int(255)			No	*None*		AUTO_INCREMENT
	2	**P_name**	varchar(50)	latin1_swedish_ci		No	*None*		
	3	**P_address**	varchar(50)	latin1_swedish_ci		No	*None*		
	4	**P_age**	varchar(15)	latin1_swedish_ci		No	*None*		
	5	**P_telno**	varchar(15)	latin1_swedish_ci		No	*None*		
	6	**P_email**	varchar(30)	latin1_swedish_ci		No	*None*		
	7	**P_pass**	varchar(15)	latin1_swedish_ci		No	*None*		

Figure 9 - Patient Table

3. Doctor

This relation maintains the Doctor Credentials that relate to the registration process. Here the field 'D_no' is incremental and is uniquely identified.

Schedule (D_no, D_name, D_address, D_telno, D_email, D_pass)

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
	1	**D_no**	int(255)			No	*None*		AUTO_INCREMENT
	2	**D_name**	varchar(50)	latin1_swedish_ci		No	*None*		
	3	**D_address**	varchar(50)	latin1_swedish_ci		No	*None*		
	4	**D_telno**	varchar(15)	latin1_swedish_ci		No	*None*		
	5	**D_email**	varchar(25)	latin1_swedish_ci		No	*None*		
	6	**D_pass**	varchar(15)	latin1_swedish_ci		No	*None*		

Figure 10 - Doctor Table

4. Doctor Schedule

This table contains all the doctor sessions initiated by individual doctors. Here the field 'sno' is incremental and uniquely identified.

Schedule (S_no, D_no*, start, end, apps, reg)

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
	1	**S_no**	int(11)			No	*None*		AUTO_INCREMENT
	2	**D_no**	int(11)			No	*None*		
	3	**Start**	date			No	*None*		
	4	**end**	date			No	*None*		
	5	**apps**	int(11)			No	*None*		
	6	**reg**	int(11)			No	*None*		

Figure 11 – Doctor Schedule Table

5. Patient Payment

This relation holds all information on patient payment. Here the field 'P_no' is incremental and is uniquely identified. Here the Totamount field is used to store the total amount of money the patient has to pay the facility. Dueamount field is used to store the amount of money that is due to be paid by the patient.

PatientPayment (P_no, S_no*, Totamount, Dueamount)

	#	Name	Type	Collation	Attributes	Null	Default	Comments
	1	**P_no**	int(255)			No	*None*	
	2	**S_no**	int(255)			No	*None*	
	3	**Totamount**	int(255)			No	*None*	
	4	**Dueamount**	int(255)			No	*None*	

Figure 12 - Patient Payment

3.2 System Code Modules

Following are several code modules used while developing components of the system. All of this was developed and tested on a controlled sandboxed environment.

Database Connection

A database connection lets you access database data sources. With a database connection, you can read data from database tables, perform SQL queries or insert records into database tables. These actions are taken by the components using a database connection. ("Help - CloverETL", 2018)

Following are code snippets used for making database connections in the Clinical Management System.

```php
<?php

    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "users";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

?>
```

Figure 13 - Database connection

Display Schedule

A doctor's schedule can consist of zero or more appointments placed by several patients. In order to keep track these appointments, a new functionality (more specifically a relation in DB) named schedule is created. Following blocks of code display the current doctor sessions. This code will list the all the appointments relevant to a doctor in a tabular, easy to understand format.

```php
<?php include "php/header.php"; ?>
</br>
<?php
    include "php/dbcon.php";

    $result = mysqli_query($conn,"SELECT d_name,start,end,apps,reg FROM doctor d, doctorschedule s WHERE d.d_no = s.d_no;");

    echo "<center><table border='1' id='customers'>
    <tr>
    <th>Name</th>
    <th>Starting Time</th>
    <th>Ending Time</th>
    <th>No. of Appointments</th>
    <th>Registered users</th>
    </tr>";

    while($row = mysqli_fetch_array($result)){
        echo "<tr>";
        echo "<td>" . $row['d_name'] . "</td>";
        echo "<td>" . $row['start'] . "</td>";
        echo "<td>" . $row['end'] . "</td>";
        echo "<td>" . $row['apps'] . "</td>";
        echo "<td>" . $row['reg'] . "</td>";
        echo "</tr>";
    }
    echo "</table></center>";

    mysqli_close($conn);
?>
```

Figure 14 - Display Schedule

Defining User Privileges

In a system such as this which contains several sensitive user information, it is important to have control of what level of information a user has access to. In order to do that, we can create suitable user roles and define respective user privileges. Assigning appropriate user role and privileges can prevent the wrong person from accessing the wrong information.

As a way of implementing the above, users in this system are given different levels of users with different privileges when stored in the database. Each user’s email, first name, last name and privilege is stored in un-encrypted form while the password is MD5 hashed and stored in the database.

```php
<?php
    include "dbcon.php";

    $fname = $_POST['fname'];
    $lname = $_POST['lname'];
    $email = $_POST['email'];
    $pass = $_POST['pass'];
    $priv = 0;


    switch($_POST['type']){
        case 'patient':
            $priv = 0;
            break;
        case 'doctor':
            $priv = 1;
            break;
        case 'admin':
            $priv = 2;
            break;
        case 'staff':
            $priv = 3;
            break;
        default:
            $priv = 0;
            break;
    }
```

Figure 15 – Define user privileges

Password Recovery by Email

This function is a must have in any system in order to facilitate the users with an option to regain access to their account, if they forget their username or password. The regular patients of the institution are provided the option to either recover their passwords through customer service or emails. This function works mainly through email.

```php
include "php/dbcon.php";

// Was the form submitted?
if (isset($_POST["ForgotPassword"])) {

    // Harvest submitted e-mail address
    if (filter_var($_POST["email"], FILTER_VALIDATE_EMAIL)) {
        $email = $_POST["email"];

    }else{
        echo "Email is not valid";
        exit;
    }
    echo $email;
    // Check to see if a user exists with this e-mail
    $query = $conn->prepare('SELECT email FROM customer WHERE email = :email');
    $query->bindParam(':email', $email);
    $query->execute();
    $userExists = $query->fetch(PDO::FETCH_ASSOC);
    $conn = null;

    if ($userExists("email"))
    {
        // Create a unique salt. This will never leave PHP unencrypted.
        $salt = "498#2D83B631%3800EBD!801600D*7E3CC13";

        // Create the unique user password reset key
        $password = hash('sha512', $salt.$userExists["email"]);

        // Create a url which we will direct them to reset their password
        $pwrurl = "localhost/reset_password.php?q=".$password;

        // Mail them their key
        $mailbody = "Dear user,\n\nIf this e-mail does not apply to you please ignore it. It appears that you have requested
        mail($email, "Kenway Medicals - Password Reset", $mailbody);
        echo "Your password recovery key has been sent to your e-mail address.";

    }
    else
        echo "No user with that e-mail address exists.";
}
?>
```

Figure 16 - Password recovery

3.3 Development Environment

Hardware environment

- Intel Pentium Dual Core 2.03GHz processor CPU
- Computer memory 1024MB
- 200GB Hard Disk Drive
- Stable Internet connection / Network

Software environment

- Microsoft Windows 7 Ultimate / Ubuntu
- MySQL Server
- MySQL Control Centre

3.4 Operational Environment

A dedicated server running 24/7 for the MySQL database and the web application is needed. Minimum system specifications for the server is as stated above.

Summary

All of the functional requirements as stated in the Requirement Analysis stage are achieved in this along with a few other non-functional requirements to improve user friendliness. The system code is modularized so that any further improvement is easily supported in the coming iterations of the system life cycle.

4. Chapter 5 - Testing

4.1 Introduction

Software testing can be stated as the process of validating and verifying that a software program/application/product,

1. Meet the requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics.

Testing process is carried out with the intent of finding errors. Software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. Verification and Validation is the name given to these checking and analysis process. Verification involves checking that the software conforms to its specification, such as functional and non-functional requirements. Validation is to ensure that the software system meets the customer's expectations. (Types of Software Testing: Different Testing Types with Details — Software Testing Help , 2018)

4.2 Testing Strategies

In the developed Clinical Management System, Developer has used 3 main testing strategies. Black Box testing, White Box testing and System testing.

- **Black Box Testing** – This strategy is mostly applied on Unit testing and is not carried out by the developer, since the component is too large and less critical. Black Box is carried out by studying system's inputs and outputs and deciding on the necessary outcome.
- **White Box Testing** – This is also not carried out by the developer but instead a specialized person/team is used. The method is the same as the black box

testing but the team is provided with the source code to analyze vulnerabilities within the code.

- **System Testing** – This includes Recovery Testing and Security Testing. In recovery testing, Developer forced the MySQL database to crash and used the restore function to recover the database. Security testing is used to prevent unauthorized entry to the system. That is to check if the basic security features are met.

4.3 Test Cases and Test Results

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.

The process of developing test cases can also help find problems in the requirements or design of an application. (Test Case - Software Testing Fundamentals, 2018)

Test cases for few of the module are given below. Test cases and test results for the rest of the system modules are available in Appendix E – Test Cases and Test Results.

Test Cases

1.0 Login Module

#	Procedure	Expected Result
1.1	Provide empty email & password	Display error message “Please enter your email ID”
1.2	Provide valid email & invalid password	Display error message “Incorrect email ID or password”
1.3	Provide valid password & invalid email	Display error message “Incorrect email ID or password”

1.4	Provide valid email & password	Display user landing page
1.5	Provide invalid email format & valid password	Display error message "Incorrect email ID"
1.6	Provide password with characters exceeding max limit	Display error message "Password cannot contain more than 8 digits. Please try again!"
1.7	Pressing 'back' after successfully logging in	Move back to user landing page
1.8	Copying password to clipboard after entered	Password doesn't get copied to the clipboard
1.9	Use forged SQL queries as email & password	Display error message "Incorrect email ID or password"

Table 3 – Test Cases for Login Module

2.0 Signup Module

#	Procedure	Expected Result
2.1	Provide invalid email format	Display error message "Please enter a valid Email ID"
2.2	Provide password exceeding maximum character limit	Display error message "Password cannot contain more than 8 digits. Please try again!"
2.3	Provide invalid/incomplete password format	Display message "Your password must be at least • 8 characters long • 1 uppercase & 1 lowercase character • 1 number"
2.4	Leave a required field empty and submit form	Display message "Please fill all required fields"
2.5	Use forged SQL queries in form fields	Display error message "Incorrect email ID or password"

Table 4 – Test Cases for Signup Module

3.0 Patient - Appointment Module

#	Procedure	Expected Result
3.1	Appoint an available doctor	Log appointment on database and update webpage
3.2	Appoint an unavailable doctor	Do not display appoint button in UI
3.3	Cancel appointment from a previously appointed doctor	Cancel the appointment, log the database and update webpage
3.4	Appoint maximum registered session	Do not display appoint button in UI
3.5	Display previously placed appointments by clicking on View appointments.	The page with list of appointments is displayed.
3.6	Edit appointment information.	Relevant appointment record is displayed for editing.
3.7	View only Doctors available for appointment at a given time.	All appointment data are filtered to show only the available appointment slots in a given time.
3.8	Submit an appointment.	This should display the success or failure to the submission. If successful, should display the appointment number.

Table 5 – Test Cases for Appointment Module

4.4 User Acceptance

The feedback at the implementation stage is covered in this part. Firstly the client feedback for the system is taken into consideration before making the system go live in a specific region. The system is made to go live only when the client approved of the product.

The client's feedback at different iterations of the product design is discussed in this module. All of this was taken into consideration before releasing the final system.

The system was mainly broken down into 4 main parts as,

i. General Website Layout
ii. Schedule display
iii. Account layout design

iv. Integrating the backend

The client feedback that was received based on the above given areas of the system is as given below.

General Website Layout

Client accepted the UI we provided. Although some design properties on the display was changed.

Schedule Display

The table design was changed from the initial display.

Account Layout Design

This was accepted the first iteration and the client was very happy about it.

Final product

Was accepted after a few minor adjustments in the user interface.

Customers' feedback after successfully implementing the system is stated down in this module. All of this feedback was taken after the final system has been successfully implemented in a region. Numerous feedback gathering techniques were used in this

i. **Customer surveys after successfully appointing a doctor**

   Customers were often required to complete a survey after appointing one or more of the doctors using the website. This was a very effective way of gathering customer feedback since 90% of the people provided us with a feedback.

ii. **Email and Customer contact forms**

   Email is one of the most valuable ways to gather candid customer feedback. Periodic gathering of feedback using email gave us a good insight into how users explore the features of the system. This was a relatively slower way of gathering feedback from customers but it provided more in-depth details.

iii. **Exploratory customer interviews**

Several picked up face to face individual interviews were carried out between a few of the patients that used the system.

iv. **Social media**

Another type of feedback we gathered is through social media like Facebook, Twitter and Reddit.

v. **On-site analytics**

We were able to monitor few of the customer behaviors and navigations on the site to determine customer interaction within the system. This was also taken into consideration.

From these customer feedbacks we were able to identify some features that majority of the people liked and disliked and some additional features that were missing.

- Majority of the customers liked the User interface, the simplicity of navigation and the ease of understanding in the system
- A few feedbacks stated that the load time on some pages were too long.
- Additional feedbacks stated that the presence of a search function and an online payment facility would be beneficial.

The final feedback analysis was heavily considered when providing updates to the system.

5. Chapter 6 - Implementation

5.1 Introduction

Implementation is the systematically structured approach to effectively integrate software based service or component into the workflow of an organizational structure.
This chapter focuses on how the deliverables of the project will be implemented in the live environment, which is a completely functioning Clinical Management System. Furthermore this chapter emphasizes on the implementation process, how it was done and the implementation requirements.

5.2 Implementation Process

The new system is to be implemented in a conversion method known as ‘Phased-In Method’. The working version of the new system is implemented in one branch of the organization and based on the feedback it is installed throughout the country branch by branch. This method is more suitable in this scenario because the staff and doctors needs training to use the system and the public should be made aware of such a system if we are to successfully migrate from the old system.
Defining the implementation process includes how the developer plans to integrate newly developed system into current system so that it will support Phased-in adoption method without any collisions. If the user requirements are met and are happy with the new system, conversion process takes place, where the old system is closed and proceeding with the new system.
In order to reduce risk, firstly only one branch is taken for implementation. If the criteria for the new system are met, the old system is removed. The process requires careful planning and control and a significant investment in labor hours.

5.3 Hardware & Software Installation

Hardware and Software configuration has to be done according to specifications defined in the Requirements Analysis stage before moving on with the implementation process. Before implementing the newly developed system, it was pre-tested in a sand-boxed developer environment. This test was carried out to identify proper hardware and software requirements for smooth and efficient performance of the developed system. For extensive software installation and database setup refer Appendix A – System Documentation.

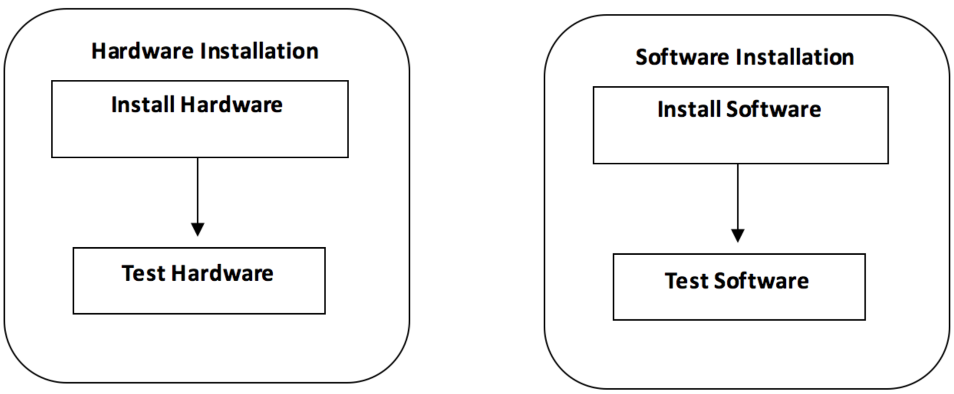


Figure 17 - Hardware and software installation

6. Chapter 7 - Self-Assessment

6.1 Introduction

This chapter concerns mainly on the problems and hardships that were faced during each stage of the Software Development Life Cycle. It emphasizes all the solutions as well as possible techniques for evasion in such situations in the future.

6.2 Problems & Solutions at different stages

Many meetings were organized with the client and the company to determine the exact nature of system that they require. A main problem that arose within the requirement gathering phase is the inability to round up the proper functional and non-functional requirements that are needed by the client. This problem mainly arose due to the inability of the client to express himself clearly. And hence it lead to a delayed start on the project.

At the beginning of the project, a security audit wasn't available. Due to that, several misconfigured security vulnerabilities could be seen at the end of the development phase. A questioning was done by the developer covering different areas to round down the functional and the non-functional requirements. Critical areas of the system were identified by the developer and left for the design and development phase to patch them accordingly.

The time period assigned for this stage was only 2 weeks. So it was a tight schedule for the developer to work on considering the lack of expression and cooperation from the client. The next thing that had to be worried about was the selection of a development model since this system could be approached in several different ways and finding the most appropriate one was quite a difficult task.

Another problem to be considered was the development of a user friendly and a satisfactory interface for the system since this system was to go live in the internet. This was one of the most considerable challenges because without a proper friendly interface no one might use the system. After thorough examination it was decided that it is best to use the Iterative development model since this will provide maximum flexibility when developing the system. Choosing of the iterative development model also helped in developing a user friendly and satisfactory interface because the system has always been checked with the client for his consent. Also this provided a way for the developer to provide the client with future updates to the system in a convenient and efficient way. If the development model was something other than this, it will be hard for the developer to provide future updates to the system as the client requests.

Most of the problems that was within the system was mitigated or completely fixed using the iterative method. Because of that it was easy to carry out the system development without many problems. But I had to decide between using commercial or free databases and decided to go with the free open-source systems such as MySQL because of its efficiency and the freely available online support. After that several obstacles were met when implementing the system and it was decided to use the Phased-In conversion method to mitigate those issues. The system was enabled one region at a time for both implementing and testing purposes. Enabling the system on a single region gave problems such as staff and doctor training for the new user interface to be carried out in each individual region. This had to be carried out very carefully. While implementing the system region by region we were getting feedback about the system from both the client and the customers. Getting this feedback and analyzing such a vast quantity of feedbacks provided a major problem since there was only a single developer available. Responding according to both client and customer needs was quite a tedious task for a single developer. To deal with system handling problems, staff training was decided to be done prior to the system implementation on a controlled sandboxed environment. This ensured that the doctors and the staff knew how to handle the system prior to it going live in the region. When dealing with the feedbacks from customers, firstly feedback from a single type of method such as social media was done. After a time period that method was closed and a different method was used such as feedback forms on the website. This ensured less overhead on the developer as well as it reached out to almost all the customers that used the system. Majority of the improvements came through this feedback system. Since an iterative model was used, client feedbacks were handled accordingly since each new iteration of the system was verified by the client before moving onto the next phase. All the test cases were pre-laid on the design & development stage, so it was possible to free run through each test case scenario to secure and verify the integrity of the system.

Looking back on how well the system conforms to the study of feasibility of the system as well as external factors that determine the success of the system, it was realized that one of the main social issues we faced was to encourage the patients to adopt the new system. Not all patients are literate in using a web based system. Therefore, it was a challenge to make this system with a simplified design and simple flow of process that would overcome the barrier caused by lack of transparency and encourage patients to use to the new system.

On the other hand, any ethical issues that may have risen was mitigated by ensuring highest safety and security of the valuable data this system holds. Legally and ethically, developing a clinical site such as this in Sri Lanka has no barriers. Any patient interested in making online interaction with the Clinic can use the system without a problem. Finally, professional issues that was faced is studying a system from a domain that's from the spectrum of medicine. That was a great challenge for me as a system analyst and as a developer. However, with great amount of research, and using suitable requirements gathering methods and much professional help, it was possible to meet the client's requirements.

I can say that the system in its current state is achieving its end-goal but more and more improvements could be added to the system in its lifetime. With the coming technologies and advancements, this system can be upgraded into a more suitable and convenient application. For example, the same principles used in the web based system can be ported into a mobile client making it easy to access for any smart phone user. Also the payment handling and confirmation done by the staff at the office could also be automated by providing an online gateway for the money transactions.

The current report generation system is on-demand but further improvement could make it deliver a daily, weekly or even a monthly report of the sales to the administrators automatically via email or by any other mean deemed necessary. These improvements could be slowly pushed in over the time without damaging or interrupting the workflow of the existing system. With that note I'm happy to say that this project has been a success.

6.3 Time period – Gantt Chart

Task / Week	1 - 2	3 - 4	5-6	7-8	9-10	11 - 12	13 - 14	15 – 16	17 - 18	19 - 20	21 - 22	23 - 24
Investigation & Analysis												
Design												
Development												
Testing												
Implementation												
Documentation												

Figure 18 - Project Schedule

References

Aisel.aisnet.org . (2018). Retrieved 2018, from Aisel.aisnet.org : http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1690&context=amcis2001

Beautiful Free Images & Pictures | Unsplash. (2018). Retrieved 2018, from Unsplash.com: https://unsplash.com/

Help - CloverETL. (2018). Retrieved from http://doc.cloveretl.com/documentation/UserGuide/index.jsp?topic=/com.cloveretl.gui.docs/docs/database-connections.html

Hemas Hospitals | Hemas Channel Online . (2018). Retrieved 2018, from Doctors.hemashospitals.com : https://doctors.hemashospitals.com/channel

Nawaloka Hospitals PLC | Sri Lanka. (2018). Retrieved 2018, from Nawaloka.com: http://www.nawaloka.com

Stunning Free Images · Pixabay . (2018). Retrieved 2018, from Pixabay.com : https://pixabay.com/

Test Case - Software Testing Fundamentals. (2018). Retrieved 2018, from Software Testing Fundamentals: http://softwaretestingfundamentals.com/test-case/

Types of Software Testing: Different Testing Types with Details — Software Testing Help . (2018). Retrieved 2018, from Softwaretestinghelp.com : https://www.softwaretestinghelp.com/types-of-software-testing/

UML Class Diagram Tutorial. (2018). Retrieved from https://www.lucidchart.com/pages/uml-class-diagram

Visual-paradigm.com . (n.d.). Retrieved from Visual-paradigm.com : https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/

What is entity relationship diagram (ERD)? - Definition from WhatIs.com. (n.d.). Retrieved 01 13, 2018, from SearchDataManagement: https://searchdatamanagement.techtarget.com/definition/entity-relationship-diagram-ERD

What is User Interface (UI) Design?. (2018). Retrieved from https://www.interaction-design.org/literature/topics/ui-design

Xu, H. (2016). *Design and Implementation for Ontology Modeling of Design Knowledge Based on UML Class Diagram*.

Appendix A - System Documentation

Hardware Requirements

Hardware Component	Minimum Requirement
Processor	Inter Core i5 or equal processor
Memory	8GB RAM or more
Hard Disk	Min 1GB free disk space or higher
Display	Viewable in any resolution
Printer	Dot matrix, Ink jet printer or Laser printer
Internet	Minimum 512kbps ADSL connection

Table 6 - Hardware Requirements of the System

Software Requirements

Software Component	Minimum Requirement
Operating System	Any Linux Distribution
XAMPP	XAMPP 5.6.11 or above
Code Editor	Notepad++ /suitable editor
Image Editor	Adobe Photoshop CC 2015 or a higher version
Web Browser	Internet Explorer/Firefox/Google Chrome/Opera (best viewed in Google Chrome)

Table 7 - Software Requirements of the System

How to Setup

1. Copy the clinic folder and paste it inside the htdocs folder in the following path C:\xampp\htdocs

2. Install the other required software according to their user manuals.

Database Setup

Figure A-3 shows few screen prints of few of the below steps.

Step 1: Type the following URL in the browser's address bar, which will open phpMyAdmin

http://localhost/phpmyadmin/

Step 2: Login by giving the username and password

Step 3: Create a blank database as "clinic".

Step 4: Click the Import tab and select the file clinic.sql (The path would be .../Database/clinic.sql) and click on the go button in order to import the database.

Appendix B - User Documentation

This section consists of the Diagram design of User Registration page and System Interfaces that are developed for other functionalities of the system. They are accompanied with a brief description to help the user understand the flow of the system.

User Registration (Diagram Design)

This is the login page for all patients, doctors, staff, admins and the registration/signup for patients.

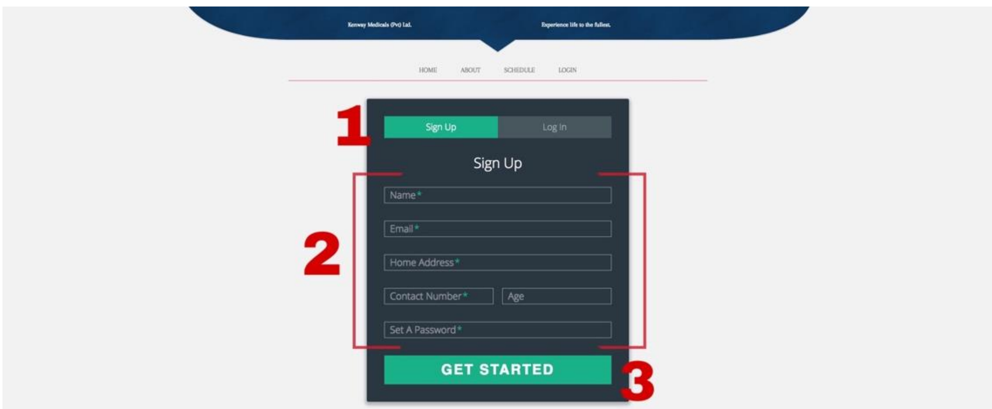


Figure 19 - Sign Up

4. If the user lacks a user account, the user will be directed to the Sign Up view (shown above) by clicking the "Don't have an account yet?" found below the Log In button shown in figure above.

5. Next, in order to register with the system, the user must fill in information that is requested by the form.

   Note: * represent the mandatory fields. Mandatory fields must be provided to register into the system.

6. Clicking on Get started button or pressing Enter key on the keyboard will complete the registration process.

Home Page

Registered or unregistered, users of all types can view the following page as the home page. the most commonly used functions of the system are displayed right on the top of the home page navigation bar, whereas more general information on company and its services are displayed right below the slider module.

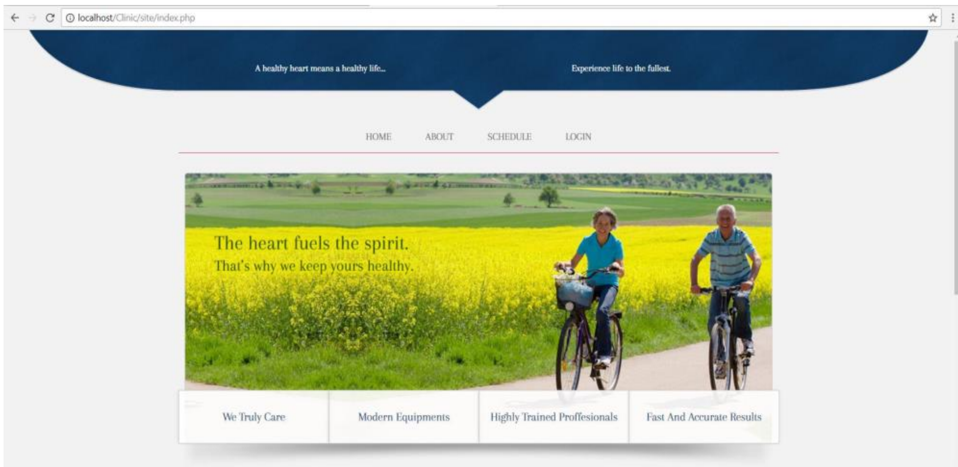


Figure 5 - Home

Staff – User Password reset

A 'staff' user while logged in is able to change a password of any Patient/Doctor using this interface.

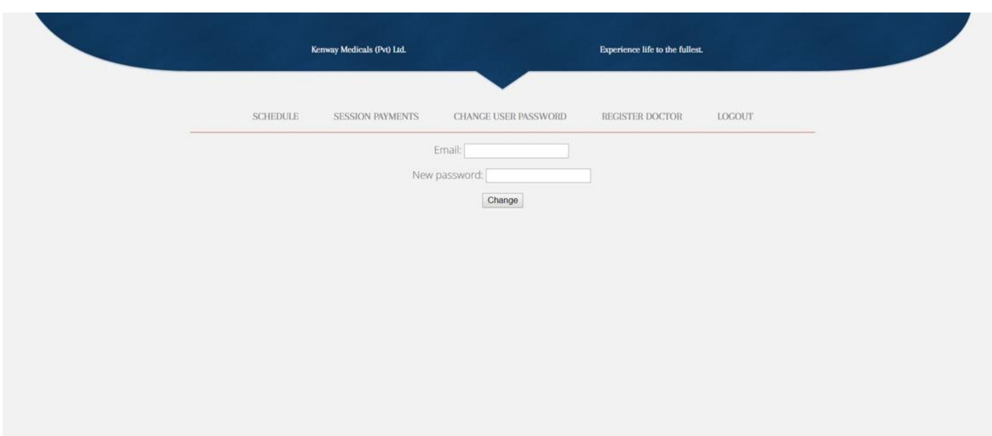


Figure 20 - Password reset

Staff – Update Payment Details

This view will allow the Staff to update the system with the offline payment made by Patients at the time of attending an appointment.

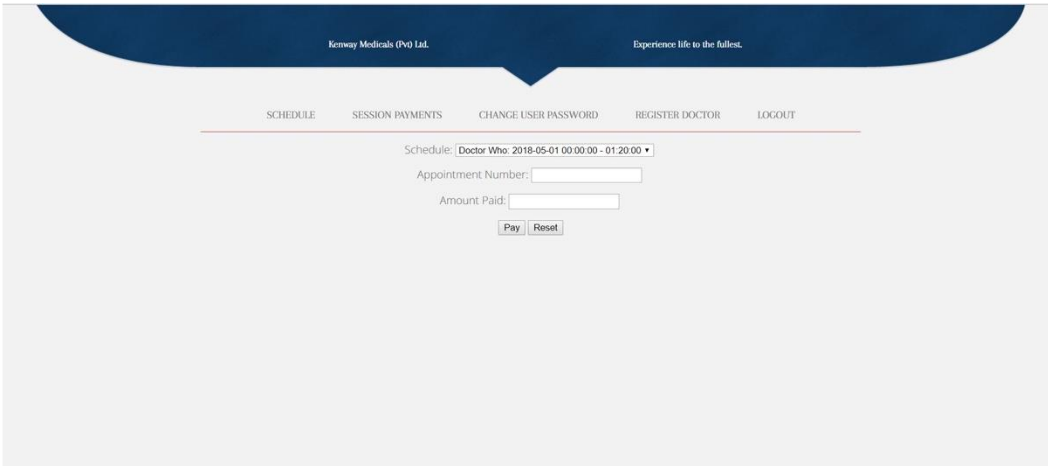


Figure 21 - Updating Payment Details

Patient - Schedule

This page displays the schedule to a normal viewer of the website. It lists all the sessions initiated by each doctor with the number of appointments available for each session and the number of registered users

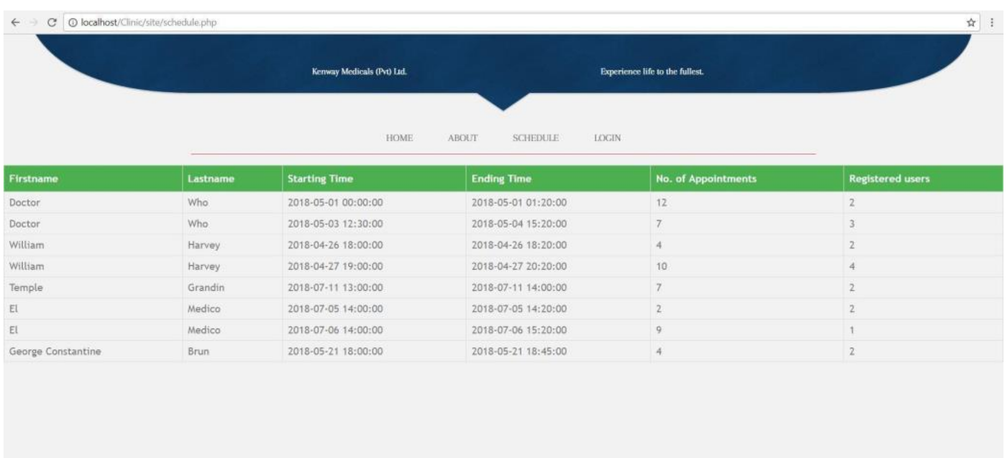


Figure 22 - Schedule view

Doctor – View Patients

A logged in ‘doctor’ is able to view his/her patients for each individual session initiated.

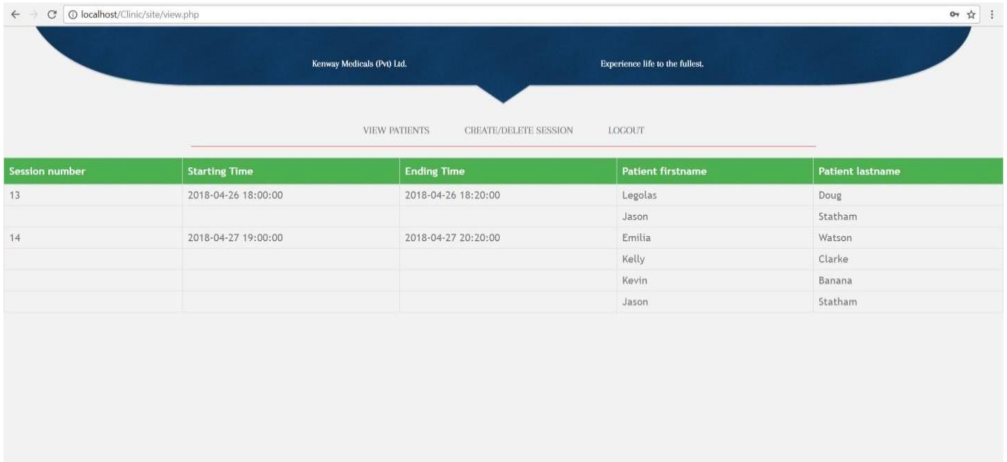


Figure 23 - View patients

Doctor – Create/Delete Session

A logged in doctor would be able to start a new clinical session or delete an existing session.

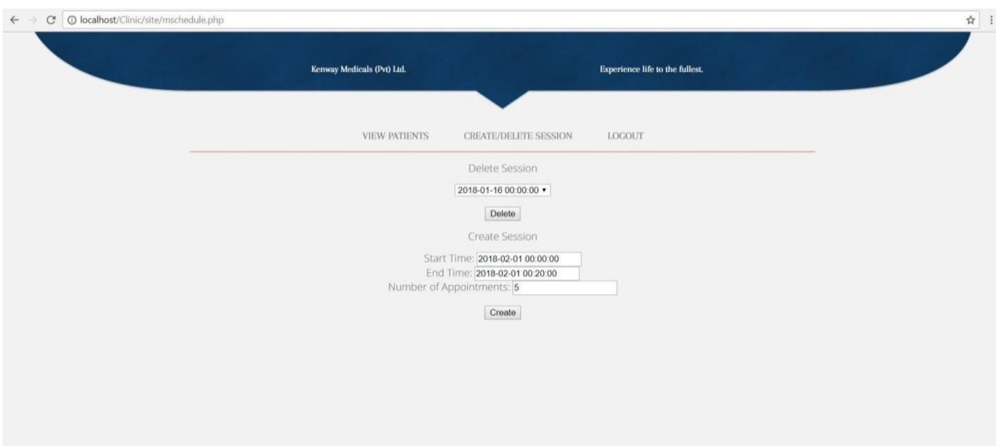


Figure 24 - Create/Delete session

Appendix C - Design Documentation

Sequence Diagram

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the facts of a Web base software-intensive system under development. Developer used UML to design the proposed system. UML combines techniques from data modeling (entity relationship diagrams), business modeling, object modeling etc.

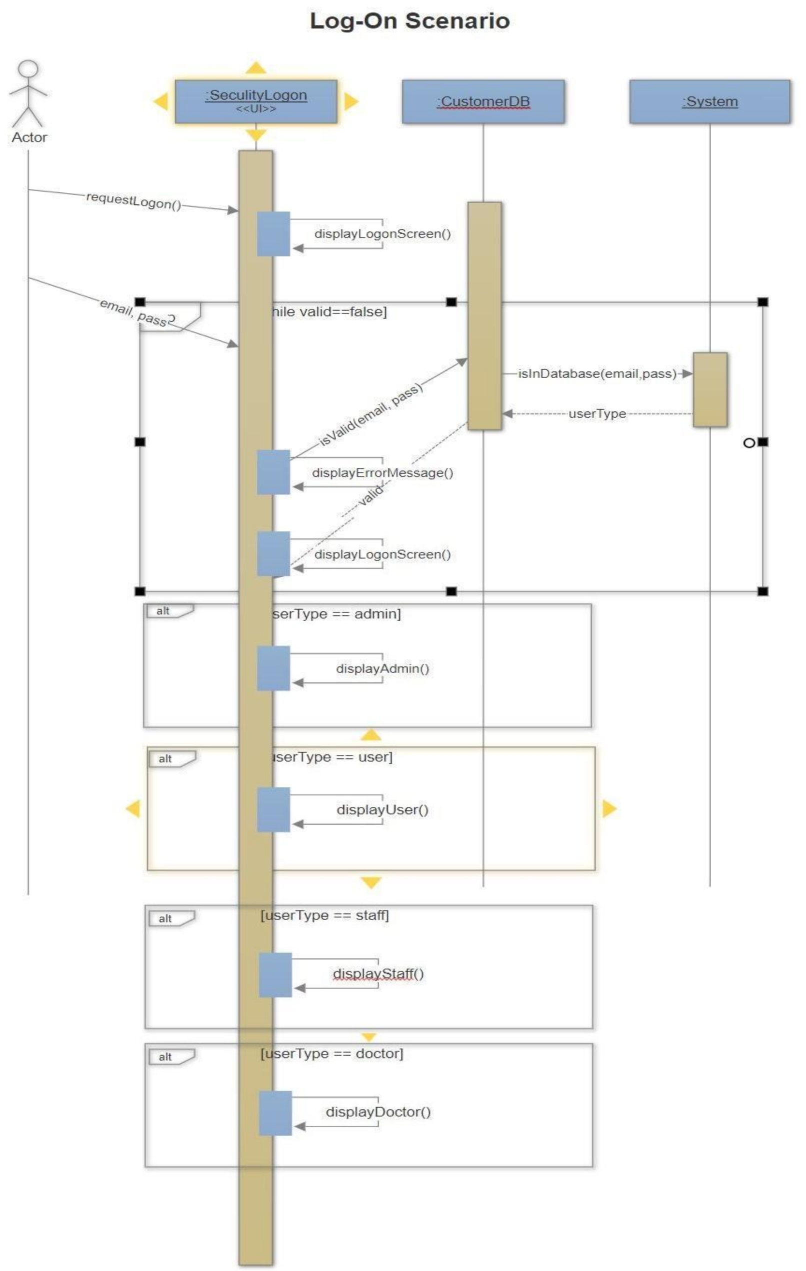


Figure 25 - Sequence Diagram

Use Case Descriptions

Use Case Name	Create/Delete schedule
Actors	Doctor
Overview	
System will add a new schedule or remove an existing doctor schedule.	

Pre-conditions
The Doctor should be logged in to the system.
Flow of Events
1. The user clicks on “Manage Schedule” from the top menu bar. 2. Choose the schedule that should be deleted, from the list of schedules displayed. 3. Click on “Delete” button relevant to the schedule selected.
Post-conditions
The schedule should be removed from the Doctor’s Schedule List.

Table 8 - Use Case Description for create and delete doctor schedule

Use Case Name	Login
Actors	Doctor, Patient, Staff
Overview	
The authorized personnel logs into his/her account.	
Pre-conditions	
The user should have registered with the system.	
Flow of Events	
1. The user enters user name and password. 2. If entry is invalid, the user is redirected to the login page with an error message. 3. If entry is valid, redirects the user is directed to the appropriate home page based the account privileges of the user.	
Post-conditions	
Only the authorized personnel logs into his/her account, others have no access.	

Table 9 - Use Case Description for system login

Use Case Name	Logout
Actors	Doctor, Patient, Staff
Overview	
The authorized personnel exit from his/her account.	
Pre-conditions	

The user should have already logged into the system using a valid username and password.
Flow of Events
1. The user selects "Logout" from "My Account" section. 2. User is logged out of the system and redirected to home page. **Post-conditions**
User is logged out of the system and redirected to home page.

Table 10 - Use Case Description for system logout

Use Case Name	Register
Actors	Doctor, Patient, Staff, Admin
Overview	
A new user is added into the system.	
Pre-conditions	
The user should be eligible to be registered with the system. Must be either a Doctor, Patient or a staff member.	
Flow of Events	
1. The user enters registration details. 2. The details are directed to the Admin. 3. If the details are valid, the registration is approved by the Admin. Else, rejected.	
Post-conditions	
If a valid user, a new user account is created in the system.	

Table 11 - Use Case Description registering with the system

Use Case Name	Manage registered patients
Actors	Admin
Overview	
A list of registered users of the system is displayed, enabling the Admin to take appropriate actions on the user accounts.	
Pre-conditions	

The Admin must have been logged in to the system with a valid email and password.
Flow of Events
1. Admin clicks on “System Reports” from the menu bar. 2. Select “Registered patients” from the Reports options displayed on screen. 3. Select the time period of the data the report should consist of.
4. Finally, click “View”.
Post-conditions
The admin will be able to view the list of registered patients for the selected time period in the form of a report.

Table 12 - Use Case Description for Managing Registered Patients

Use Case Name	Make online appointments
Actors	Patient
Overview	
A new doctor appointment will be placed by the patient through the system.	
Pre-conditions	
The patient must be a registered user with the system.	
Flow of Events	
1. Patient clicks on “Place Doctor Appointment” from the menu bar. 2. From the Doctor schedules that’s displayed on the screen, the user selects the available slots for appointment. 3. Then click on “Place an appointment” button that’s next to the relevant appointment slot.	
Post-conditions	
The appointment will be placed along with the patient details. An SMS notification will be delivered to the patient prior to the event, along with the appointment details.	

Table 13 - Use Case Description for Making online appointments

Use Case Name	Reset Password
Actors	Doctor, Patient, Staff, Admin

Overview
A new password will be created as a result of the given series of actions.
Pre-conditions
The user must have a registered user account with a valid email.
Flow of Events
1. User clicks on “Forgot your password?” which is right above the Log In button in the Log In screen. 2. Upon selecting “Forgot your password” an email with a password recovery link will be sent to the user’s registered email. 3. User can then follow the instructions as provided in the screen accordingly until a new password is created.
4. User will be then prompted to sign in to the system with the new Log In Details.
Post-conditions
The user will have access to the system with a new password.

Table 14 - Use Case Description for Resetting Password

Use Case Name	Request Register
Actors	Doctor, Staff
Overview	
A new Doctor Account will be created in the system.	
Pre-conditions	
The doctor must request a new account from a staff member who has the relevant system privileges to create new account. The staff member should be logged into the system.	
Flow of Events	
1. Select “Register Doctors” from the user menu displayed on top of the page. 2. Fill in the User details in the User Registration form that’s displayed on screen. 3. Finally, select “Register User” button.	
Post-conditions	
A new Doctor account will be added to the system.	

Table 15 - Use Case Description for Doctor Registration

Appendix D - System Generated Reports

Generating reports are cornerstones of any establishment's daily operations. It is an integral activity to making strategic decisions. Report generation module in CMS – For Kenway Medicals increases the efficiency of report generation and reduces the possibility of errors which were usually made during manual report generation process prior to the developed system.

The following section discusses the core reports generated by the Clinical Management System.

Patient Appointments Report

Upon request, list of patient appointments will be generated by the system. The administrator or the staff can log in to the system and select "System reports" section. Here the user can select Patient Appointments List in order to produce this report. It is possible to filter the appointments list by selecting the time duration. Clicking on Generate Report button will generate this report list. Such filtered or unfiltered data can be printed out by clicking the print button.

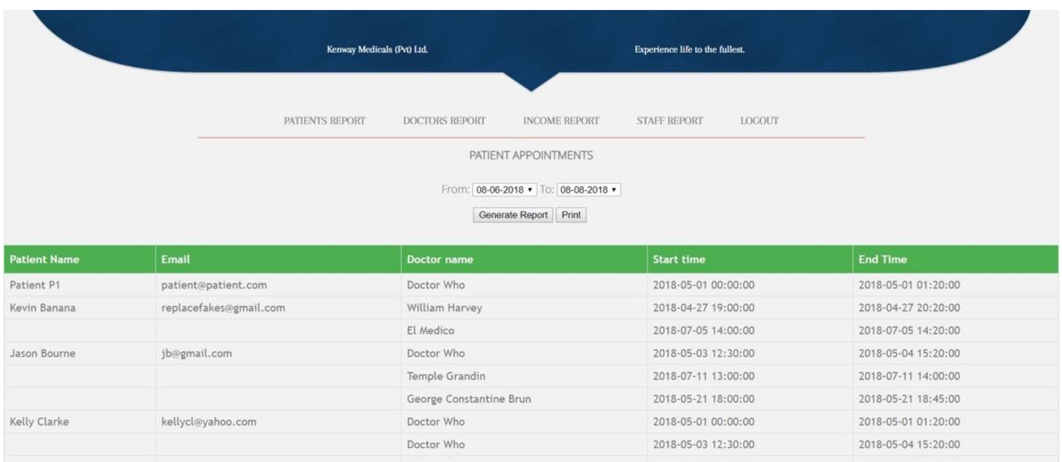
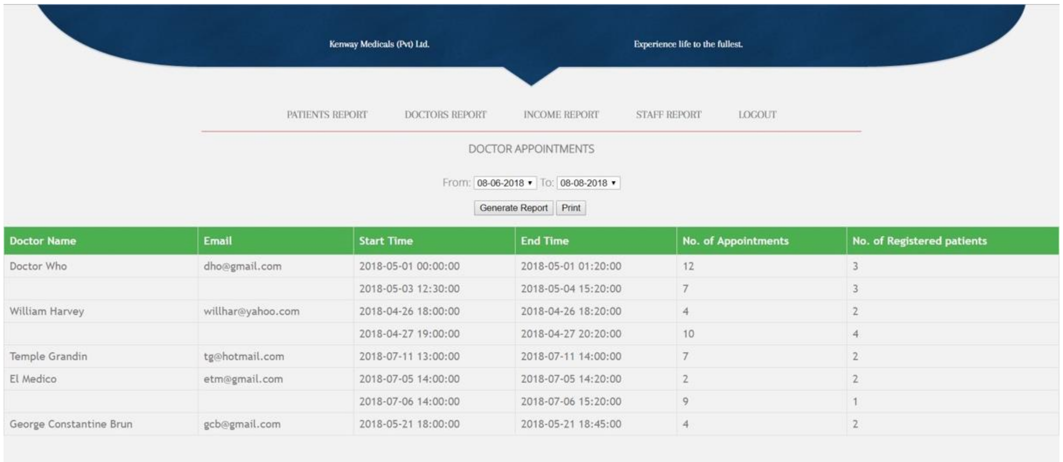


Figure 26 - Generate Patient Appointments Report

Doctor Appointments Report

The list of Doctor appointments is also generated upon request. This report can be produced by selecting the Doctor Appointments List from "System reports" section found in the menu bar. The Doctor Appointments List can be filtered by specifying a time duration. The report is generated by clicking Generate Report button. A print of this report can be taken by clicking on Print button.



Doctor Name	Email	Start Time	End Time	No. of Appointments	No. of Registered patients
Doctor Who	dho@gmail.com	2018-05-01 00:00:00	2018-05-01 01:20:00	12	3
		2018-05-03 12:30:00	2018-05-04 15:20:00	7	3
William Harvey	willhar@yahoo.com	2018-04-26 18:00:00	2018-04-26 18:20:00	4	2
		2018-04-27 19:00:00	2018-04-27 20:20:00	10	4
Temple Grandin	tg@hotmail.com	2018-07-11 13:00:00	2018-07-11 14:00:00	7	2
El Medico	etm@gmail.com	2018-07-05 14:00:00	2018-07-05 14:20:00	2	2
		2018-07-06 14:00:00	2018-07-06 15:20:00	9	1
George Constantine Brun	gcb@gmail.com	2018-05-21 18:00:00	2018-05-21 18:45:00	4	2

Figure 27 - Generate Doctor Appointments Report

Income Report

This is one of the most often produced report produced by the system to support top management decision making. With that thought in mind, the interface of the report was opted for a graphical representation of the data. This reports allows comparison between existing data. One can print this repot by selecting the "Print" button.

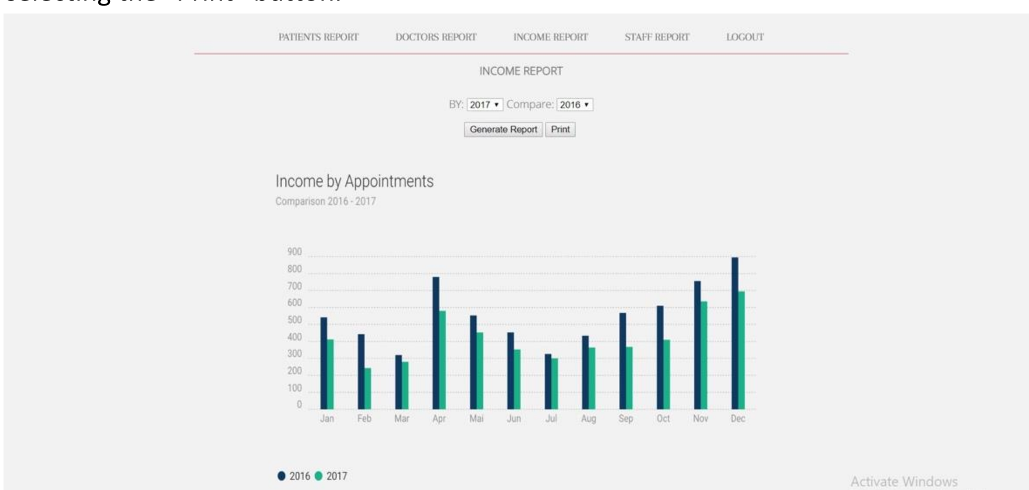


Figure 28 - Income Report

Staff Report

This report consists of the list of staff members in the system. Staff list can be categorized by User registration date. Once the time duration is selected, selecting the Generate Report button would produce the list of registered staff members. A print of this report can be obtained by clicking the Print button.

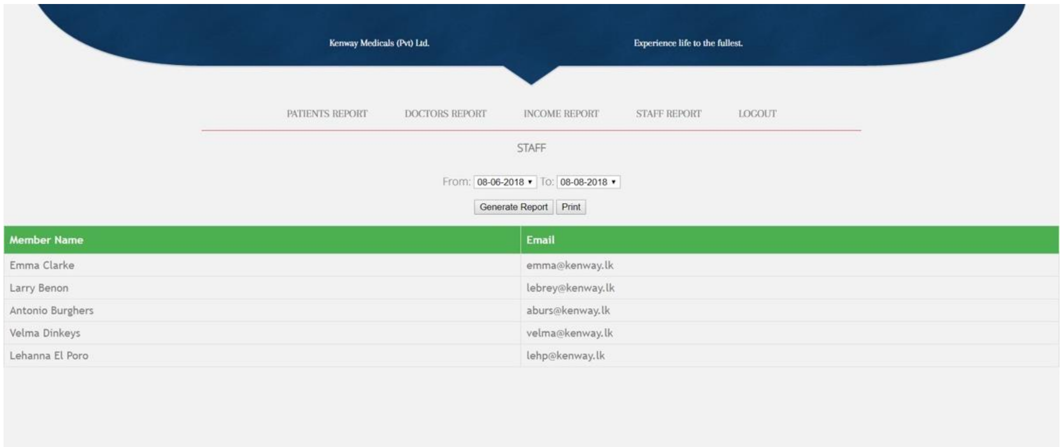


Figure 29 - Generate Staff Report

Appendix E – Test Cases and Test Results

4.0 Login Module

#	Description	Result	Result
1.1	When empty email & password	Display error message “Please enter your email ID” Please enter your email ID!	Pass
1.2	Provide valid email & invalid password	Display error message “Incorrect email ID or password” **Oops!** Incorrect email ID or password	Pass
1.3	Provide valid password & invalid email	Display error message “Incorrect email ID or password” **Oops!** Incorrect email ID or password	Pass
1.5	Provide invalid email format & valid password	Display error message “Incorrect email ID” **Oops!** Incorrect email ID	Pass
1.6	Provide password with characters exceeding max limit	Display error message “Password cannot contain more than 8 digits. Please try again!” Your password cannot contain more than 8 digits. Please try again!	Pass

1.9	Use forged SQL queries as email & password	Display error message “Incorrect email ID or password” **Oops!** Incorrect email ID or password	Pass

Table 16 – Test Cases for Login Module

5.0 Signup Module

#	Procedure	Expected Result	Result
2.1	Provide invalid email format	Display error message “Please enter a valid Email ID” Please a valid email ID!	Pass
2.2	Provide password exceeding maximum character limit	Display error message “Password cannot contain more than 8 digits. Please try again!” Your password cannot contain more than 8 digits. Please try again!	Pass
2.3	Provide invalid/incomplete password format	Display message “Your password must be at least • 8 characters long • 1 uppercase & 1 lowercase character • 1 number” Your password must be at least • 8 characters long • 1 uppercase & 1 lowercase character • 1 number	Pass
2.4	Leave a required field empty and submit form	Display message “Please fill all required fields” Please fill all required fields!	Pass
2.5	Use forged SQL queries in form fields	Display error message “Incorrect email ID or password”	Pass

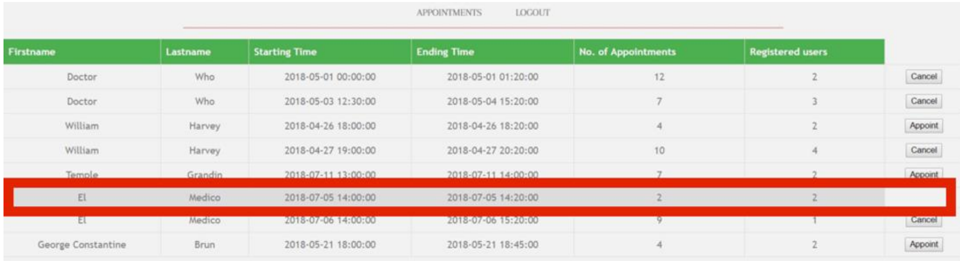
		Oops! Incorrect email ID or password	

Table 17 – Test Cases for Signup Module

6.0 Patient - Appointment Module

#	Procedure	Expected Result	Result
3.2	Appoint an unavailable doctor	Do not display appoint button in UI Refer Figure 30	Pass
3.3	Cancel appointment from a previously appointed doctor	Cancel the appointment Refer Figure 31	Pass
3.4	Appoint maximum registered session	Do not display appoint button in UI Refer Figure 30	Pass
3.5	Display previously placed appointments by clicking on View appointments.	The page with list of appointments is displayed. Refer Figure 32	Pass
3.8	Submit an appointment.	Should display the following message upon successful submission. “Your appointment is successfully added! Your appointment number is xyz.” **Appointment is successfully added!** Your appointment number is 32. Should display the following message upon unsuccessful submission. “Sorry! Something went wrong. Please add your appointment again!” **Sorry! Something went wrong.** Please add your appointment again!	Pass

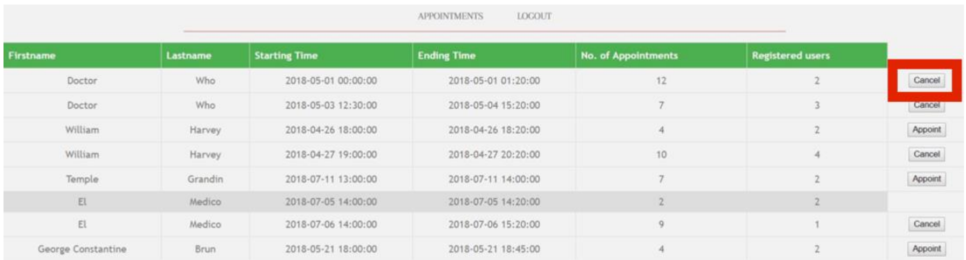
Table 18 – Test Cases for Appointment Module



APPOINTMENTS LOGOUT

Firstname	Lastname	Starting Time	Ending Time	No. of Appointments	Registered users	
Doctor	Who	2018-05-01 00:00:00	2018-05-01 01:20:00	12	2	Cancel
Doctor	Who	2018-05-03 12:30:00	2018-05-04 15:20:00	7	3	Cancel
William	Harvey	2018-04-26 18:00:00	2018-04-26 18:20:00	4	2	Appoint
William	Harvey	2018-04-27 19:00:00	2018-04-27 20:20:00	10	4	Cancel
Temple	Grandin	2018-07-11 13:00:00	2018-07-11 14:00:00	7	2	Appoint
El	Medico	2018-07-05 14:00:00	2018-07-05 14:20:00	2	2	
El	Medico	2018-07-06 14:00:00	2018-07-06 15:20:00	9	1	Cancel
George Constantine	Brun	2018-05-21 18:00:00	2018-05-21 18:45:00	4	2	Appoint

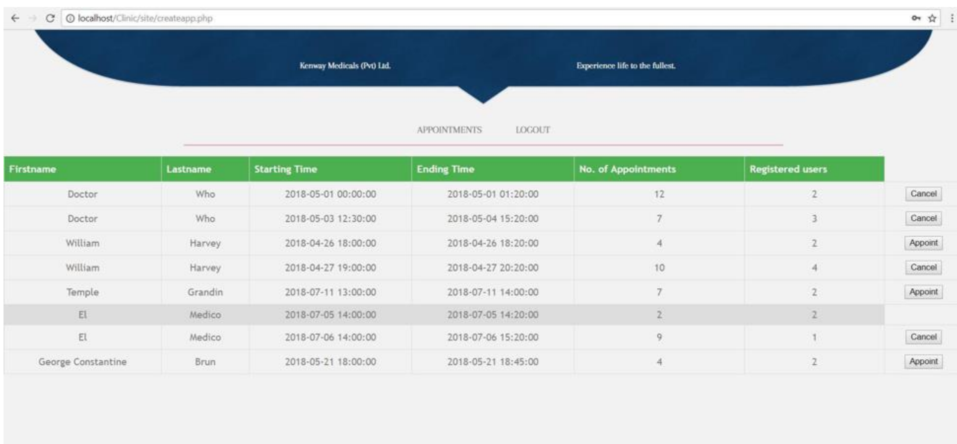
Figure 30 - Hide Appoint Button



APPOINTMENTS LOGOUT

Firstname	Lastname	Starting Time	Ending Time	No. of Appointments	Registered users	
Doctor	Who	2018-05-01 00:00:00	2018-05-01 01:20:00	12	2	Cancel
Doctor	Who	2018-05-03 12:30:00	2018-05-04 15:20:00	7	3	Cancel
William	Harvey	2018-04-26 18:00:00	2018-04-26 18:20:00	4	2	Appoint
William	Harvey	2018-04-27 19:00:00	2018-04-27 20:20:00	10	4	Cancel
Temple	Grandin	2018-07-11 13:00:00	2018-07-11 14:00:00	7	2	Appoint
El	Medico	2018-07-05 14:00:00	2018-07-05 14:20:00	2	2	
El	Medico	2018-07-06 14:00:00	2018-07-06 15:20:00	9	1	Cancel
George Constantine	Brun	2018-05-21 18:00:00	2018-05-21 18:45:00	4	2	Appoint

Figure 31 - Cancel Appointment



localhost/Clinic/site/createapp.php

Kenway Medicals (Pvt) Ltd. Experience life to the fullest.

APPOINTMENTS LOGOUT

Firstname	Lastname	Starting Time	Ending Time	No. of Appointments	Registered users	
Doctor	Who	2018-05-01 00:00:00	2018-05-01 01:20:00	12	2	Cancel
Doctor	Who	2018-05-03 12:30:00	2018-05-04 15:20:00	7	3	Cancel
William	Harvey	2018-04-26 18:00:00	2018-04-26 18:20:00	4	2	Appoint
William	Harvey	2018-04-27 19:00:00	2018-04-27 20:20:00	10	4	Cancel
Temple	Grandin	2018-07-11 13:00:00	2018-07-11 14:00:00	7	2	Appoint
El	Medico	2018-07-05 14:00:00	2018-07-05 14:20:00	2	2	
El	Medico	2018-07-06 14:00:00	2018-07-06 15:20:00	9	1	Cancel
George Constantine	Brun	2018-05-21 18:00:00	2018-05-21 18:45:00	4	2	Appoint

Figure 32 - Display List of appointments

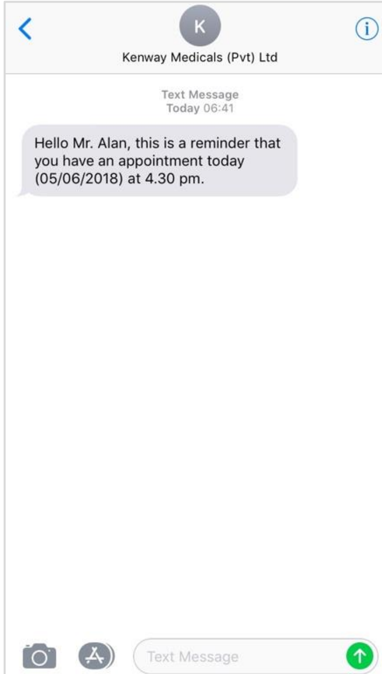


Figure 33 - SMS notification for doctor appointments

Appendix F - List of Code

Figure 30 to figure 33 given below depict the important code segments that make up the system.

login.php

The following code snippet represents the layout of the login form. This will collect the login information from the user and validate it with the existing user information on the back end. User will be able to proceed with the Log In process if the provided information are valid. Else, the user is prevented from Logging in by displaying an error message.

```
<body>
<?php include "php/header.php"; ?>
  <div class="form">

      <ul class="tab-group" id="tabGroup">
        <li class="tab active"><a href="#signup">Sign Up</a></li>
        <li class="tab"><a href="#login">Log In</a></li>
      </ul>

      <div class="tab-content">
        <div id="signup">
          <h1>Sign Up</h1>

          <form action="php/register.php" method="post">
          <input type="hidden" name="type" value="patient">
          <div class="top-row">
            <div class="field-wrap">
              <label>
                First Name<span class="req">*</span>
              </label>
              <input type="text" required autocomplete="off" name="fname"/>
            </div>

            <div class="field-wrap">
              <label>
                Last Name<span class="req">*</span>
              </label>
              <input type="text"required autocomplete="off" name="lname"/>
            </div>
          </div>

          <div class="field-wrap">
            <label>
              Email Address<span class="req">*</span>
            </label>
            <input type="email"required autocomplete="off" name="email"/>
          </div>

          <div class="field-wrap">
            <label>
              Set A Password<span class="req">*</span>
            </label>
            <input type="password"required autocomplete="off" name="pass"/>
          </div>
```

Figure 34 - Code list for system login

sms_gate.php

The Sms is sent using the Twilio API and that the company has an active subscription. A cron-job is setup to check the database every minute to determine which schedule starts in 1 hour. And when it finds one that starts in 1 hour, the sms script is executed.

```php
<?php

namespace App\AppointmentReminders;

use Illuminate\Log;
use Carbon\Carbon;
use Twilio\Rest\Client;

class AppointmentReminder
{

    public function sendReminders()
    {
        $this->appointments->each(
            function ($appointment) {
                $this->_remindAbout($appointment);
            }
        );
    }

    private function _remindAbout($appointment)
    {
        $name = $appointment->name;
        $time = Carbon::parse($appointment->when, 'UTC')
            ->subMinutes($appointment->timezoneOffset)
            ->format('g:i a');

        $message = "Hello $name, this is a reminder that you have an appointment at $time!";
        $this->_sendMessage($appointment->phoneNumber, $message);
    }

    private function _sendMessage($number, $content)
    {
        $this->twilioClient->messages->create(
            $number,
            array(
                "from" => $this->sendingNumber,
                "body" => $content
            )
        );
    }
}
```

Figure 35 – code list for notification by SMS

reset_password.php

To reset a password, the user must have a valid email account to which the password reset link could be forwarded. The following code snippet shows the structure of the password reset module and it respective functionality.

```
<body>
<?php include "php/header.php"; ?>
  <div class="form">

      <ul class="tab-group" id="tabGroup">
        <li class="tab active"><a href="#reset">Reset</a></li>
      </ul>

      <div class="tab-content">


        <div id="reset">
          <h1 id="tName">Reset your password</h1>

          <form action="php/Change.php" method="post" id="loginForm">

            <div class="field-wrap">
            <label>
              Email Address<span class="req">*</span>
            </label>
            <input type="email"required autocomplete="off"/ name="email">
          </div>


          <button class="button button-block" id="tButton" type="submit" name="ForgotPassword">Request Reset</button>

          </form>

        </div>

      </div><!-- tab-content -->

</div> <!-- /form -->
  <script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

    <script  src="js/index.js"></script>
<?php include "php/footer.php"; ?>
</body>
```

Figure 36 - Code list for password reset

schedule.php

The following code is intended to list the doctor schedule in a tabular format. The data are loaded from the doctorSchedule relation in database. The data from the database then fills in the table on the end user screen record wise.

```
<html>
<head>
    <meta charset="UTF-8">
    <title>Cardiology Website Template</title>
    <link rel="stylesheet" href="css/style.css" type="text/css">
    <!--[if IE 7]>
        <link rel="stylesheet" href="css/ie7.css" type="text/css">
    <![endif]-->

</head>
<body>
    <?php include "php/header.php"; ?>
    </br>
    <?php
        include "php/dbcon.php";
        echo "<center>Delete Session</center></br>";

        $sql = "SELECT cno FROM customer WHERE email ='". $_SESSION['doctor'] ."';";
        $result = mysqli_query($conn,$sql);
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
        $cno = $row['cno'];

        echo "<form action='php/cschedule.php' align='center' method='post'>
            <select name='sno'>";

        $sql2 = "SELECT * FROM schedule WHERE cno ='". $cno ."';";
        $result2 = mysqli_query($conn,$sql2);
        while($row = mysqli_fetch_array($result2)){
            $start = $row['start'];
            $sno = $row['sno'];
            echo "<option value='" . $sno . "'>". $start ."</option>";
        }

        echo "</select><br><br><input type='submit' value='Delete'/></form><br>";

        echo "<center>Create Session</center><br/>";
        echo "<form action='php/cschedule.php' align='center' method='post'>";
        echo "Start Time: <input type='text' name='start' value='2018-02-01 00:00:00'></input><br/>";
        echo "End Time: <input type='text' name='end' value='2018-02-01 00:20:00'></input><br/>";
        echo "Number of Appointments: <input type='text' name='apps' value='5'></input><br/>";

        echo "<br><input type='submit' value='Create'/>";
        echo "</form>"
    ?>
```

Figure 37 - Code list for Scheduling appointments

Content Styling

The following code shows a brief section of the system styling. This code belongs to the style.css file that consists of all the styling elements that's used throughout the system.

```
$body-bg: #c1bdba;
$form-bg: #13232f;
$white: #ffffff;

$main: #1ab188;
$main-light: lighten($main,5%);
$main-dark: darken($main,5%);

$gray-light: #a0b3b0;
$gray: #ddd;

$thin: 300;
$normal: 400;
$bold: 600;
$br: 4px;

*, *:before, *:after {
  box-sizing: border-box;
}

html {
    overflow-y: scroll;
}

body {
  background:$body-bg;
  font-family: 'Titillium Web', sans-serif;
}

a {
  text-decoration:none;
  color:$main;
  transition:.5s ease;
  &:hover {
    color:$main-dark;
  }
}

.form {
  background:rgba($form-bg,.9);
  padding: 40px;
  max-width:600px;
  margin:40px auto;
  border-radius:$br;
  box-shadow:0 4px 10px 4px rgba($form-bg,.3);
}

.tab-group {
  list-style:none;
  padding:0;
  margin:0 0 40px 0;
  &:after {
    content: "";
    display: table;
    clear: both;
  }
```

Figure 38 - Code list for content styling

END OF APPENDIX