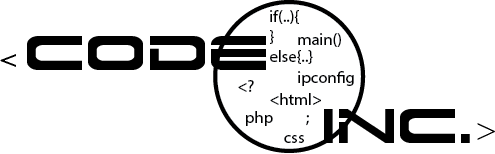
Software Requirements Specification

Version 1.0

April 27, 2017

Web Application System



**Team Members:**

Stephanus Ismael

Jeremaya Fekayamala

Loide M Kephas

Samuel James

Manuel Yolanda

Client: Dr. Stanley. N. Kanyemba

Lecture: Pro. Nic Bidwell

Submitted in partial fulfillment

Of the requirements of

CMP3731 Software Engineering

**TABLE OF CONTENTS**

Table of contents 1

1. Introduction 2

1.1. Purpose 2

1.2. Scope of Project 3

1.3. Definitions 3

1.4. Overview 3

1.5. Who is doing what? 4

1.6 Gantt chart 4

1.7. Process methodology 5

2. Requirement Analysis 5

2.1. Method to gather requirements 5

2.2. Functional Requirements 5

2.3. Non-Functional Requirements 6

2.4. Risk assessment 7

3. Modeling Requirements 7

3.1. UML Design 7

3.1.1 Use case Diagram 7

3.1.1.1 Use case diagram Admin 8

3.1.1.1 Use case diagram User 8

3.1.2. Activity diagrams 9

3.1.2.1. Activity diagram for admin 9

3.1.2.2. Activity diagram for user 19

3.1.3. Sequence diagram 10

3.1.3.1 Sequence diagram for admin 10

3.1.3.2 Sequence diagram for user 10

3.1.4. Class Diagram 11

4. Architecture of our System 11

4.1. Client-Server architecture 11

5. Physical Design 12

6 Test Plan 18

7. Implementation 20

7.1. Software Requirement 20

7.2. Hardware Requirement 20

8. Further development 20

9. Conclusion 20

10. Bibliography 21

**1. INTRODUCTION**

This document goes over the requirements that must be fulfilled before the software can be delivered to the client, (Dr S.N. Kanyemba) a medical doctor is an orthopaedic surgery at #Khon Institute practice. For the purpose of this document “client” shall explicitly refer to Dr Kanyemba. Section 1 discusses the scope of the application, any definitions or acronyms used, along with the organization of the SRS document. The requirements, physical design and test cases of the system (web application) will be discussed in subsequent sections of this document. Code Inc. planned to regularly meet with the client every fortnight and other form of communication is via telephone. At the end of the semester, will prepare a system, which is a Doctor’s appointment booking system. The application will include all features mentioned in ‘Purpose, Scope and Objective’ part of this document. Code Inc. will deliver a draft SRS report at the end of the semester. As a part of the deliverables, the Code Inc. team will also give an overall system presentation to the client and other audience.

**About client**

#Khon Institute is a general orthopaedic practice (i.e. bone, joint and muscle diseases). The practice derives its name “#Khon” from an indigenous Damara word translated to "Bones" in English.

Sub-disciplines of #Khon Institute are:

* Surgery
* Pediatric orthopedics
* Hand surgery
* Foot surgery
* Reconstruction
* Trauma

The #Khon Institute is wholly owned by a Namibian orthopaedic surgeon Dr. Stanley Ndakoro Kanyemba. Dr. Kanyemba was born and raised in Tsumeb and attended school at Nomtsoub primary before moving to Concordia College for his secondary education. Dr. Kanyemba went to the University of Namibia before embarking his journey into medical school. He graduated from University of KwaZulu Natal’s Medical School, and obtained his postgraduate from University of Wits and his fellowship from COMSA.

* 1. **Purpose**

This document is to describe all the software requirement specification (SRS) for the Doctor’s appointment booking system (Dabs) which we obtained by interviewing the client one-on-one to get the requirements. The system aims to help the patients to make appointment online via the internet. #Khon Institute is based and operates in Windhoek.

**1.2. Scope of Project**

This Doctor’s appointment booking system is centralized online calendar which displays doctor’s availability and available slots at all time. This system enables doctor’s secretary to give appointments to patients according to the doctor’s availability status. The system provides placing a new appointment, modifying and deleting an existing appointment, showing weekly schedule, questions and answers, addressed topics on medical subjects and advertisements. This system should facilitate adding, deleting and updating schedule seamlessly. Moreover, it should be easy to use and understandable. The aim of the project is to prepare a Doctor’s appointment booking system, which arranges the most appropriate time for patients.

**1.3. Definitions, acronyms, and abbreviations**

Define all terms, acronyms, and abbreviations need to understand the SRS

SRS - Software Requirements Specification

Dabs – Doctor’s appointment booking system

User - The person who uses the software

Patient - The person who uses the software

Doctor – A medical practitioner

Secretary – A medical practitioner’s assistant

Receptionist - a person who deals with clients and visitors to a surgery

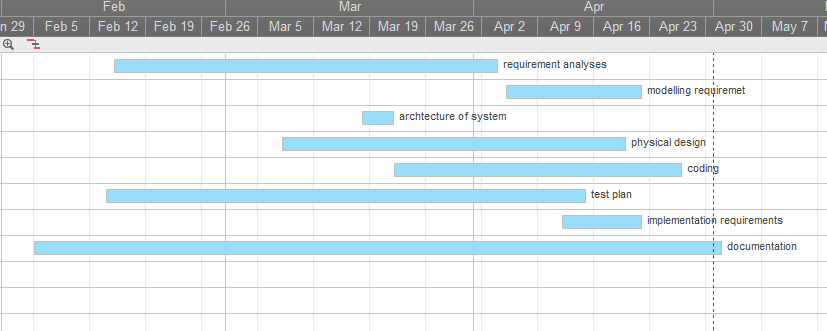
Administrator – A person who manages the system.

**1.4. Overview**

**Description of problem**

Its current system is a paper-based system which required to be physically present at the practice to make a booking, but with the increase in the number of patients visiting, it has become difficult to manage the appointment system manually. The purpose of this project is to solve these complications by creating custom-built database software to manage the appointment system. For the receptionist (Admin) to make it easy to set date and time for the appointment of the patients to the doctor. This document is intended for both the client (Dr Stanley .N. Kanyemba) and the project development team (Code Inc.).

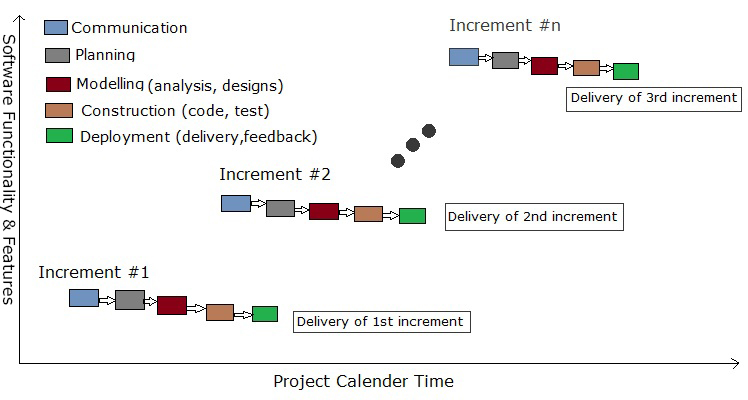
**1.5. GANTT CHART**



**1.6. Who is doing what?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stephan | Loide | James | Jeremy | Yolanda |
| meet with the client and Writing the SRS, | meet with the client and Writing the SRS, | meet with the client | meet with the client and Writing the SRS, | meet with the client and Writing the SRS, |
| User query, index, contact page | login and registration page | work on further development | work on further development | login, registration and contact page |
| Booking and admin page | Booking and admin page | Booking and page | Booking and page | Booking and page |
| diagrams | diagrams |  | diagrams | diagrams |
| Test case |  |  |  | Test case |

**1.7. Process methodology**



* The methodology that will be used to implement the project is iterative and incremental development, which is an agile methodology. We are using the agile methods as the project requirements is most likely to change with time thus; we need a flexible way to alter the project when necessary.
* Another reason why the agile method is preferred is because Code Inc. is small group and the product being develop is of a small scale. The system should have a high usability level with its simplicity in usage and success in practice.

1. **REQUIREMENTS ANALYSIS**

**2.1. Method to gather requirements**

One-on-one interviews: Code Inc. team members meet with the clients when client was available for requirement gathering as well as for further clarity

Brainstorming: based on the knowledge some team members have on what could possibly happen in a general practice some of the requirements were brained stormed but later were presented to the client, some requirements were approved others rejected.

**2.2. Functional Requirements**

* It must allow a user to register.
* It must allow a user to login using the credentials they submitted at the registration point.
* It must allow a user to logout.
* Store all basic information in the database. (Basic information are as follow:

Patient names, passwords, patient No, patient file, appointment time and date, number of appointments)

* It must allow users to select the appointment time according to his or her preference.
* It should allow setting of new appointments
* Modifying and deleting an existing appointment
* Allows appointment cancellations
* Show weekly and Monthly schedules
* Questions and answers platform
* Address health topics
* Show adverts

**Server Requirements**

* A correct combination of username and password is required for a user to login.
* A user must be logged on to be able to view or edit any appointment information.
* Calendar system connected to a database

**User Accounts**

* + Each account must securely store all of its data.
  + Each account must have a separate identifying account number.
  + Data must always be related to an account, and a user cannot view data he/she does not own.

**Basic Information That Must be Stored**

* + 1. Username
    2. Password
    3. Email
    4. Nationality
    5. Age
    6. Medical aid number

**2.3. Non-Functional Requirements**

* The system will be usable (Usability) meaning all users will find it easy to understand and use
* The system will be readily available (Availability), the system has to be available at all times.
* Reliability, the system shall be designed so that it can be completely reliable. Reliability comes in

two forms: reliability in terms of the system always being up and reliability in perfect atomic transactions.

* Performance of the system will be efficient enough to provide service in real time for a large

number of people.

* The system will be secured in order to prevent unauthorized access of data, as well as protecting

the data from data loss by doing daily backups, and keeping the data in secured places.

* The system will be maintained regularly to ensure that the information kept in the system is up

to date. Information could range from; patient names, passwords, health history, appointments availability, doctor's absence.

* Recoverability, the system will ensure that information stored is able to be recovered during a

disaster event or upon the request of users

**2.4. Risk assessment**

* A Doctor’s appointment booking system will possibly be handling sensitive medical/personal data, so security is the utmost concern.
* The system has to provide secure internet connections for the transfer of private data.
* Flooding of accounts
* Fake bookings/ appointments

**Solutions**

* Generating a unique patient ID that is required by every patient in order to set up appointments. The patient ID is only updated into the patient’s account once a patient file is opened by the client in the presence of the patient and registration fee is paid.
* Cancellations of bookings should be done until 2 hours before a scheduled appointment. Failure to do so will result in a penalty charge for a missed appointment.
* Users will have to login to the system.
* Logging in will determine what level of access the user has with the system and the views with which they are provided.
* Security also means that user’s personal information shall be guarded very closely and all passwords shall be encrypted in the database.
* Being able to prove to customers that the system is very secure will be essential for the future success of an online appointment booking system.
* The web hosting service should provide security services so that third parties cannot modify the information hosted on the web site.

1. **MODELING REQUIREMENTS**

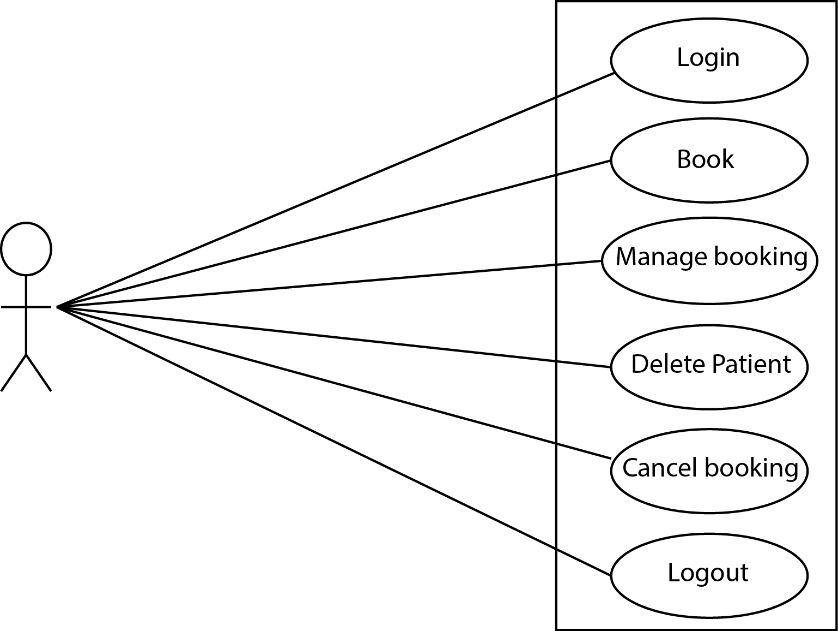
**3.1. UML Design**

UML design is the shortest form of “Unified Modelling Language”. The purpose of this modelling language is to visualize the design of the system. In this document, the following diagrams will be presented:

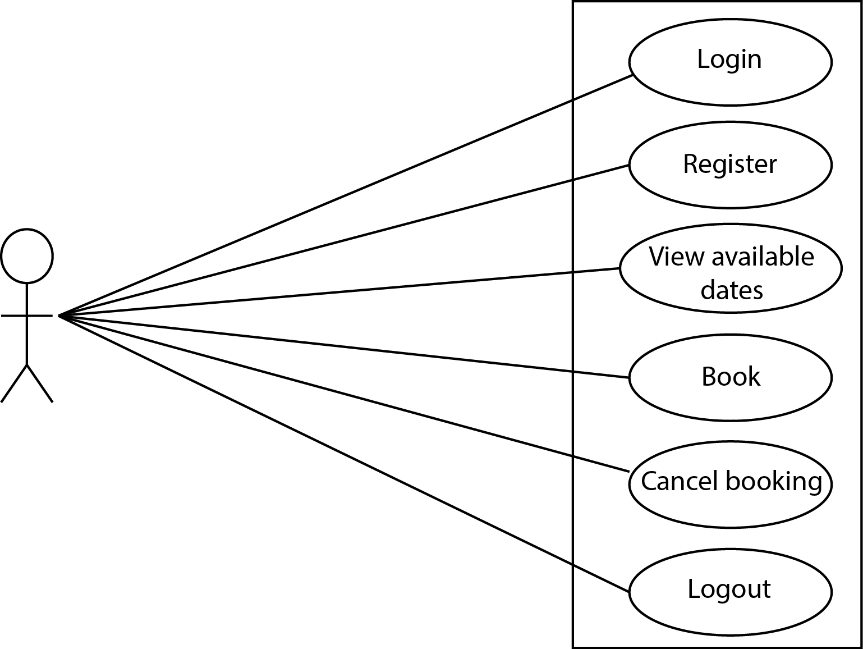
* Use case Diagram
* Activity Diagram
* Sequence Diagram
* Class Diagram
  + 1. **Use case Diagram**

It is also called behavioral UML diagram. It gives a graphic over-view of the actors involved in a system directly. It shows the different functions needed by the actors.

**3.1.1.1. Use case Diagram Admin**



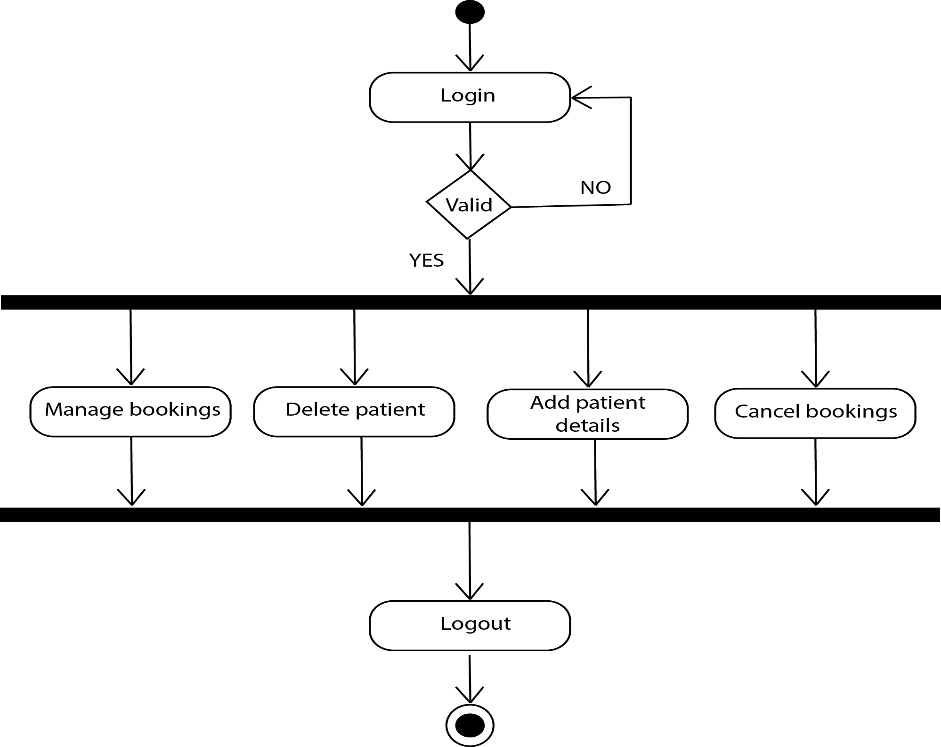
**3.1.1.2. Use case Diagram User**

****

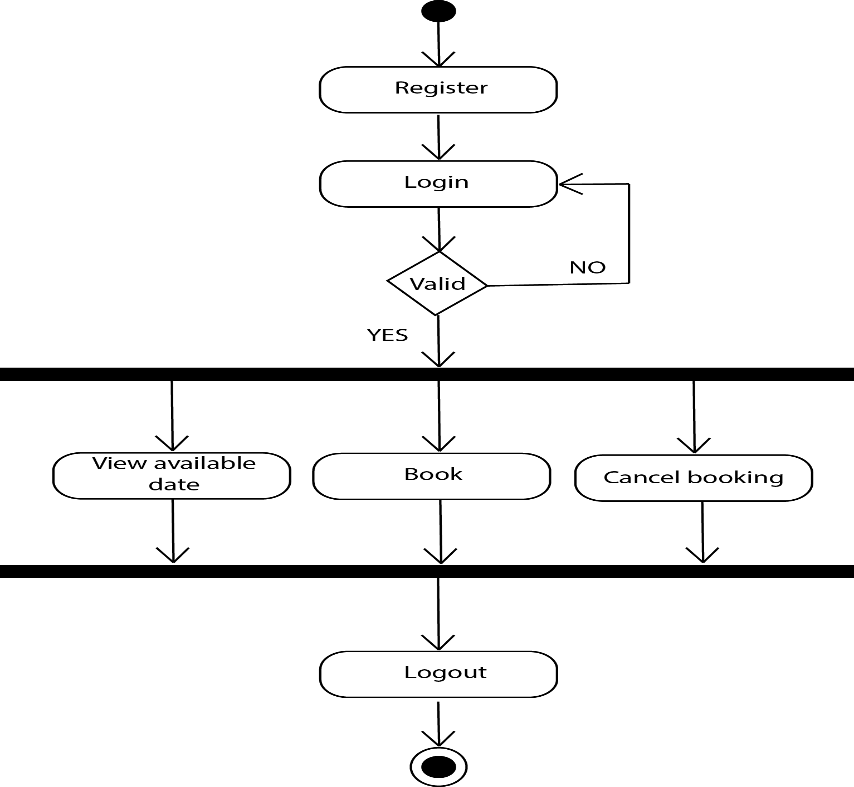
* + 1. **Activity Diagram**

This diagram visually/graphically represents the series of actions or workflow of stepwise activities and actions with support for choice, iteration and concurrency.

**3.1.2.1. Activity Diagram for Admin**



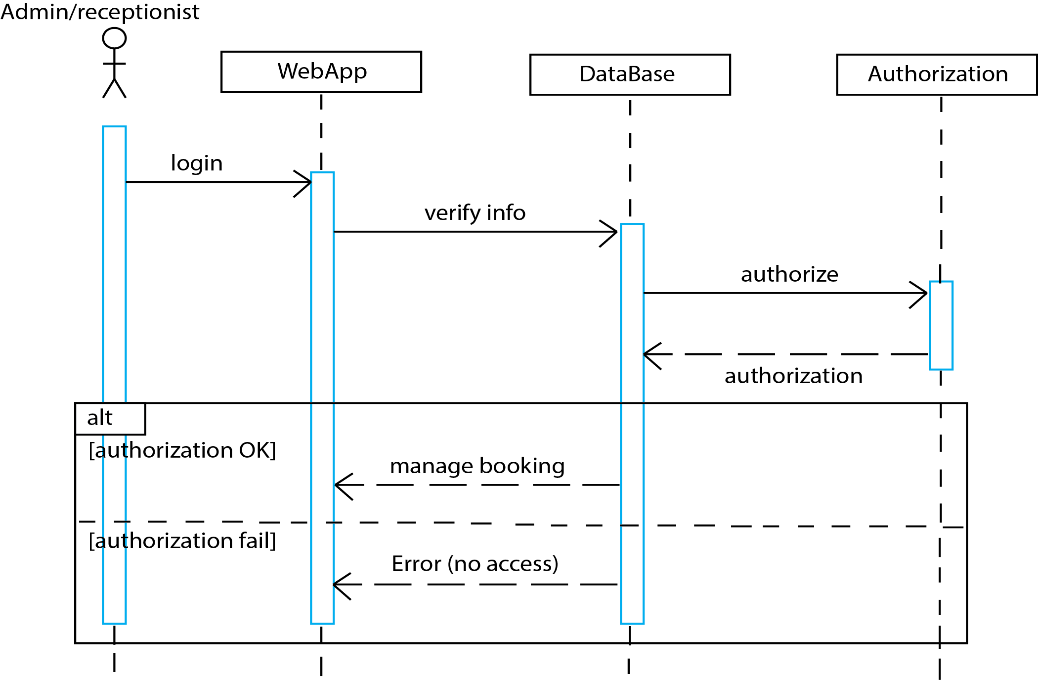
**3.1.2.2 Activity Diagram for User**



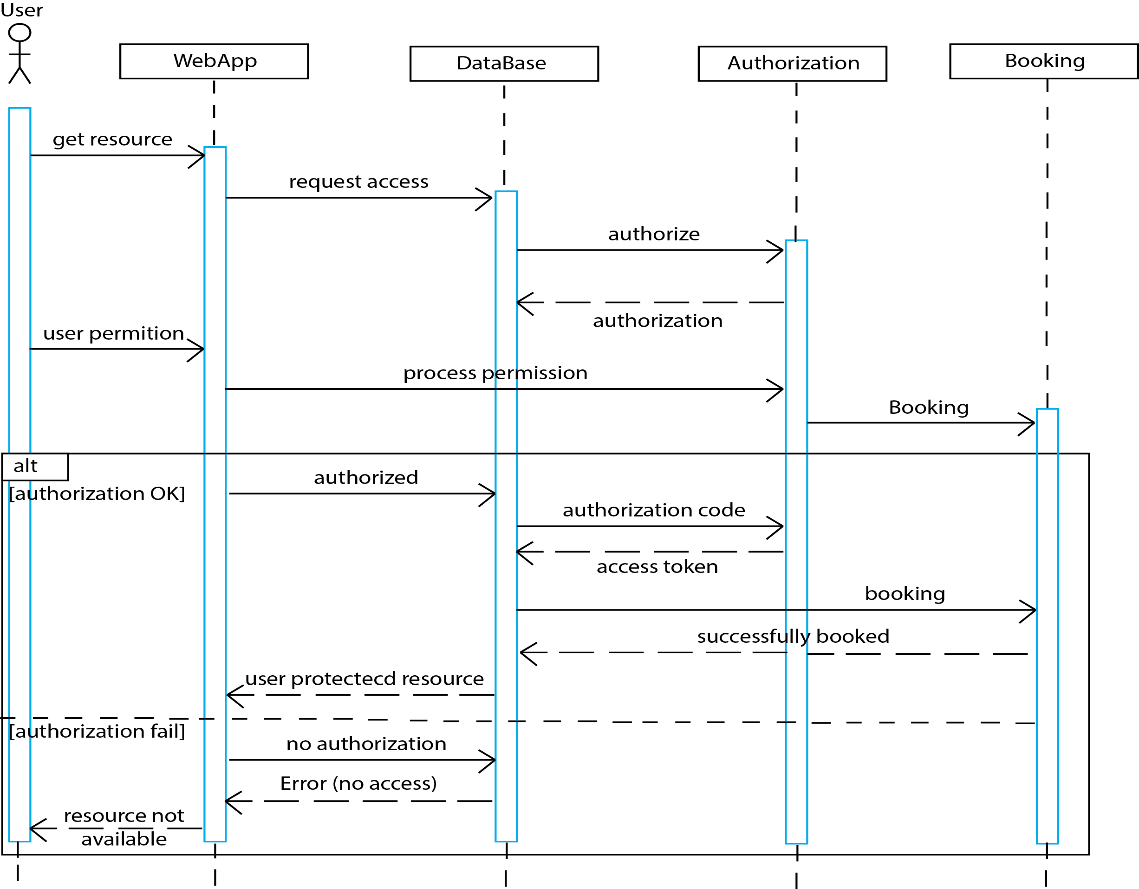
* + 1. **Sequence diagram**

This diagram models the flow of within the system in a visual manner, also how objects operate with one another and in what manner.

**3.1.3.1 Sequence diagram for admin**

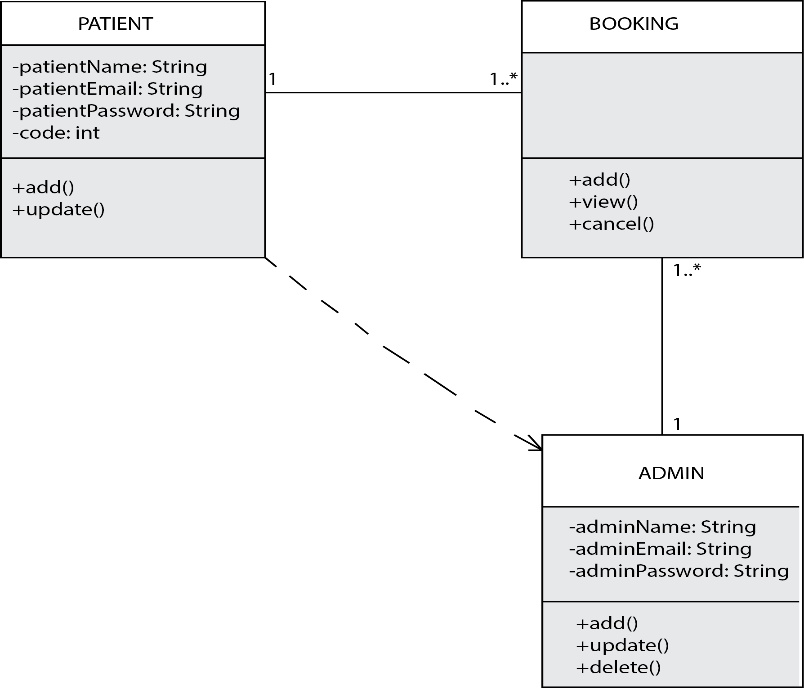


**3.1.3.2. Sequence diagram for user**



**3.1.4. Class Diagram**

This is the most used UML diagram in the field of software engineering design. It is main building block of any object oriented solution. Usually it illustrates the classes in a system, attributes and operations of each class and also the relationship between each class.

****

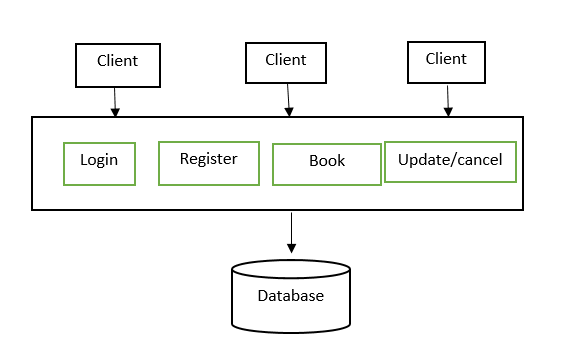
**4. ARCHITECTURE OF OUR SYSTEM**

**4.1. Client-Server architecture**

A client-server architecture is a distributed system that shows how data and processing is distributed across a range of components.

Why Client-Server architecture?

Because our system is a web application which is based on the client-server model. The client being users/patients that send requests and the Server being our database that offers service to the requests send by the clients. Our system has a client side and server side. Hence it is a suitable match for the Client-Server architecture.



**Advantages** of using a Client-Server architecture:

* Distribution of data is straightforward
* Makes efficient use of networked systems
* Easy to add new servers or updating existing servers

**Disadvantages** of using a Client-Server architecture

* No shared data system, so sub-system use different data organization. Data interchange may be inefficient
* Redundant management in each server
* No central register of names and services. It may be difficult to find out which servers and services is available.

**5. PHYSICAL DESIGN**

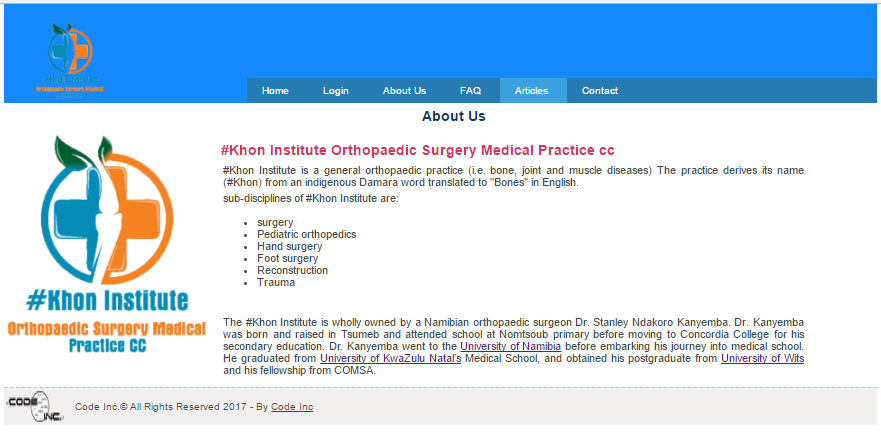
**5.1. Home Page**

The Home page interface welcome users to the webApp. It contains the navigation bar that links to the different pages such as login, about etc. This page contains a slide show of the client at work and a section where users can leave comments and also have a live discussion going which was what the client requested. Also on the home page are the medical articles and frequently asked questions sections which are abstracts that link a user to their main pages for further reading.



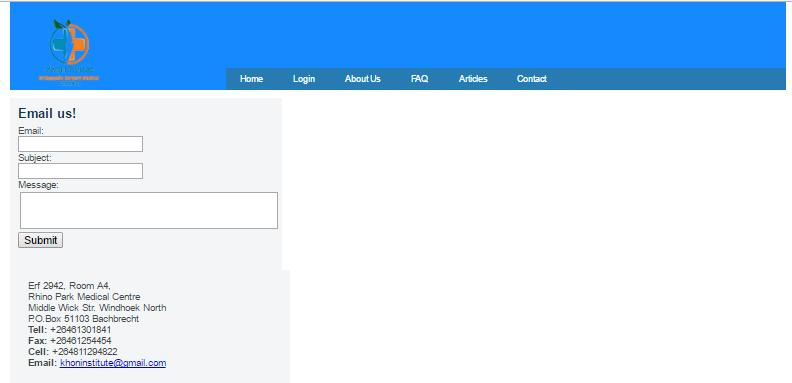
**5.2. About Us Page**

The About Us page gives a brief description of #Khon Institute as well as brief information about the client. Hovering over and clicking on the clients name will take you to the clients Facebook page. One of the request from the client was also that hyperlinks be included, so when you hover and click on the institutions the client attended will direct you to the official websites of the respective institutions.



**5.3. Contact Page**

From here users can get the contact number as well as address of #Khon Institute. They can also send enquiries through the system directly to the client. They have to make sure they enter their details correctly like email address, name. If users wish to visit the office they can simply follow the information from it.

****

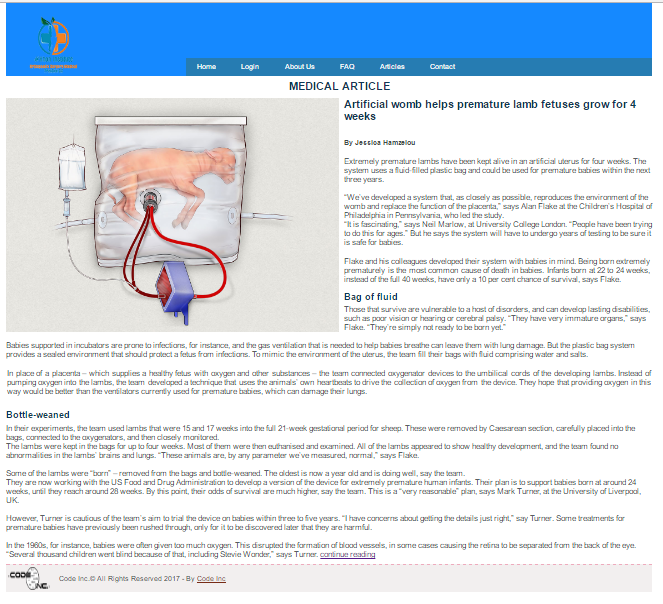
**5.4. Frequently Asked Questions (FAQ) Page**

This page features medical related questions that are regularly asked which the client will try answer in as simple terms as possible for any ordinary person to understand. It was the clients expressed wish to also include this page to help people that might have similar problems and require solutions.



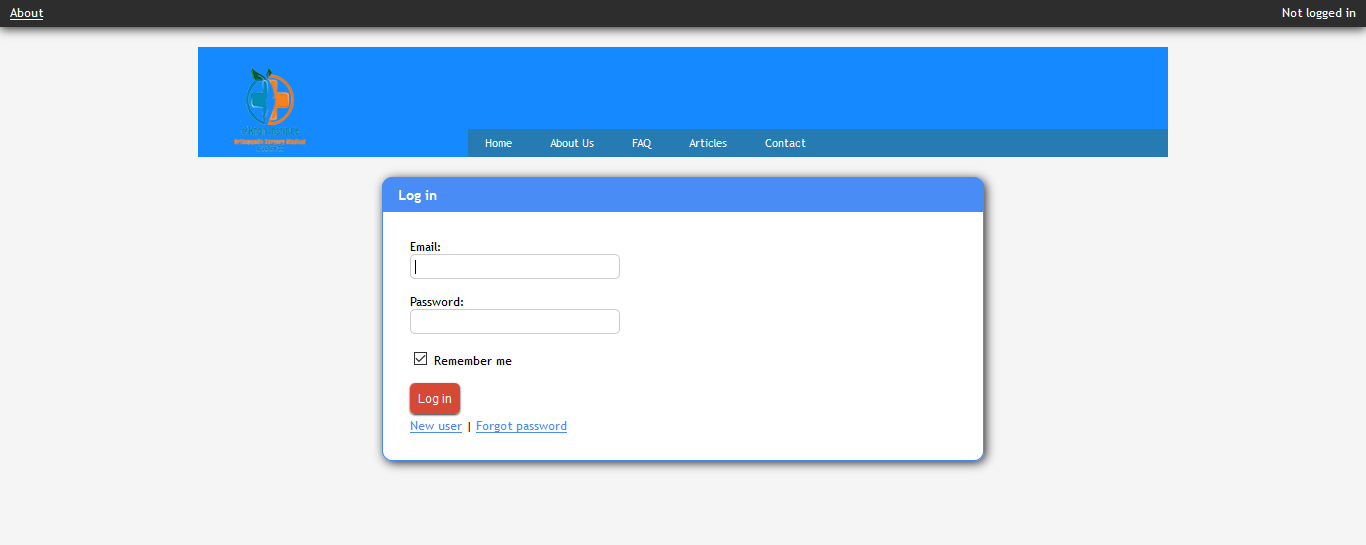
**5.5. Medical Articles Page**

This page feature medical articles or medical topics of interest. This include new breakthrough in the medical world in terms of technological advancement or a cure to an illness, or just some medical tips and health awareness.



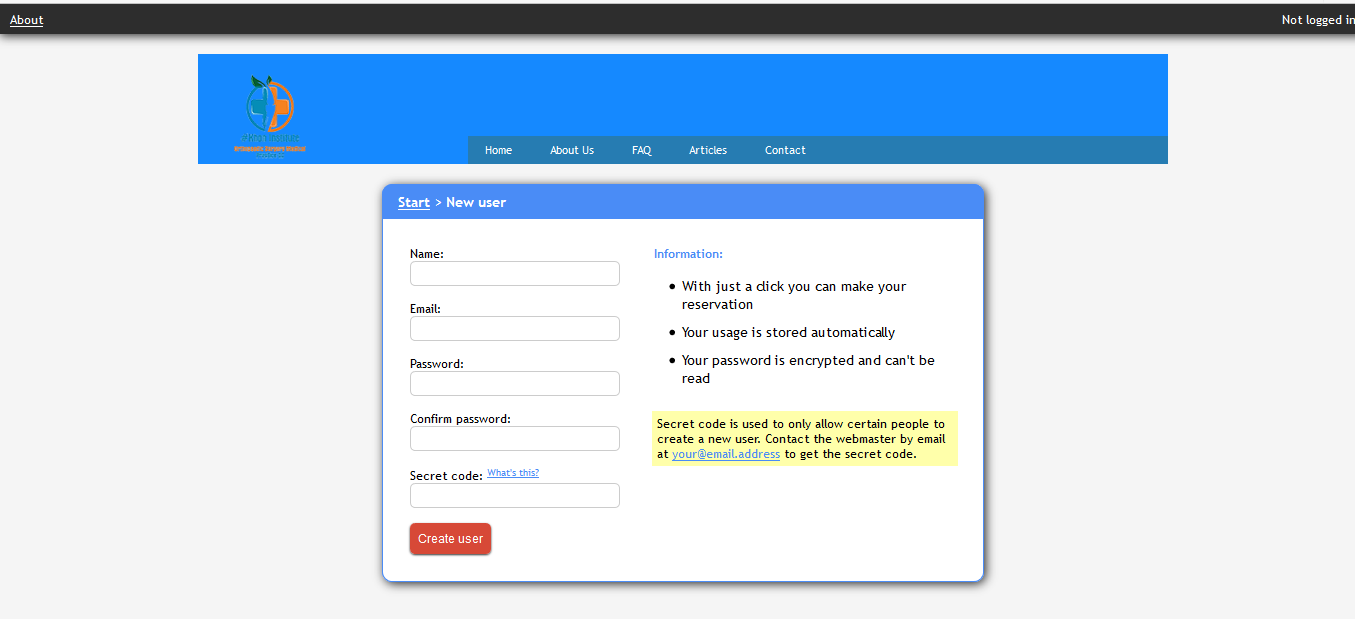
**5.6. Login Page**

This page allows a registered user to access the booking page in order to reserve an appoint with the client. None registered users are given an option to register



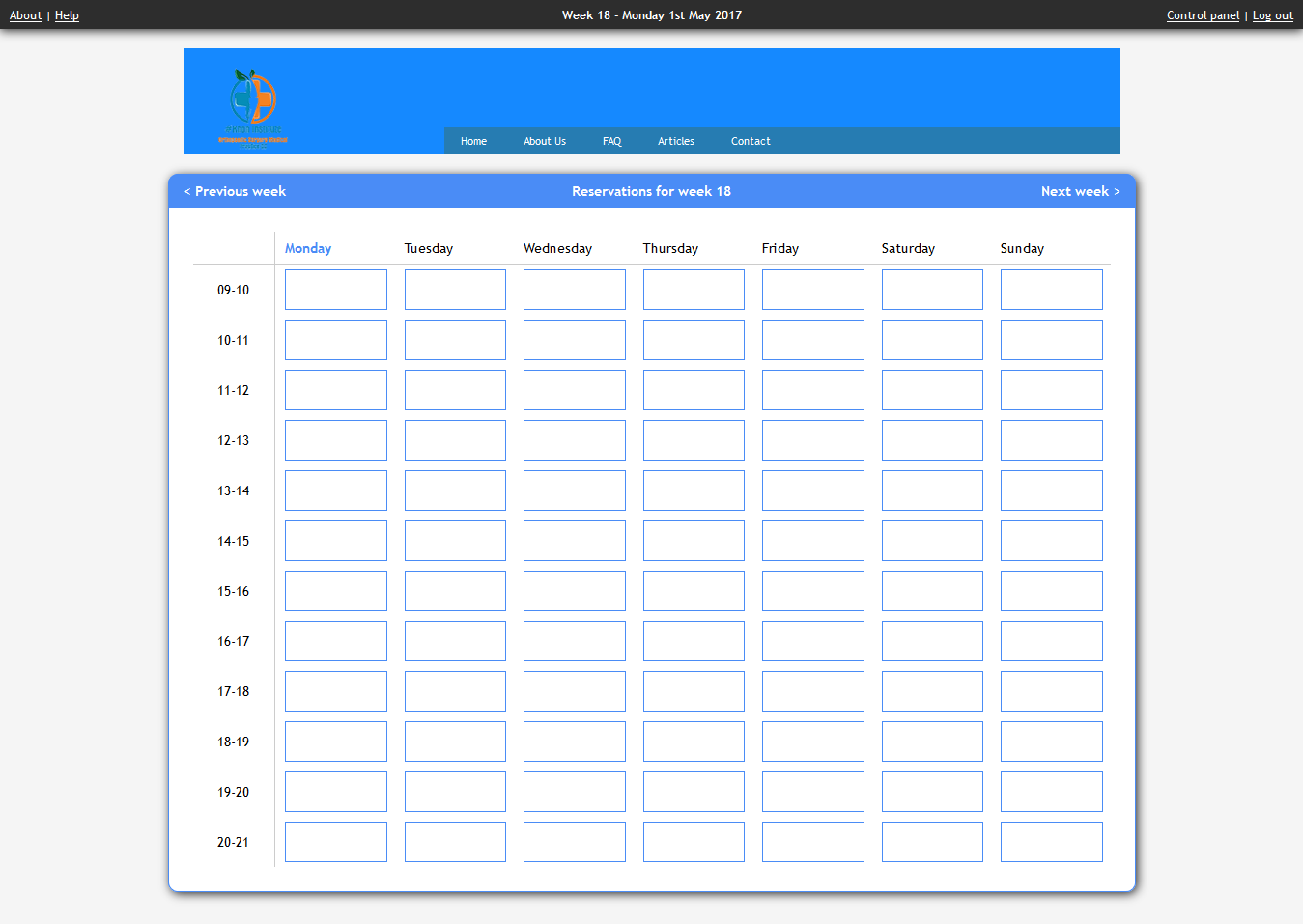
**5.7. Register Page**

None registered users can register here.



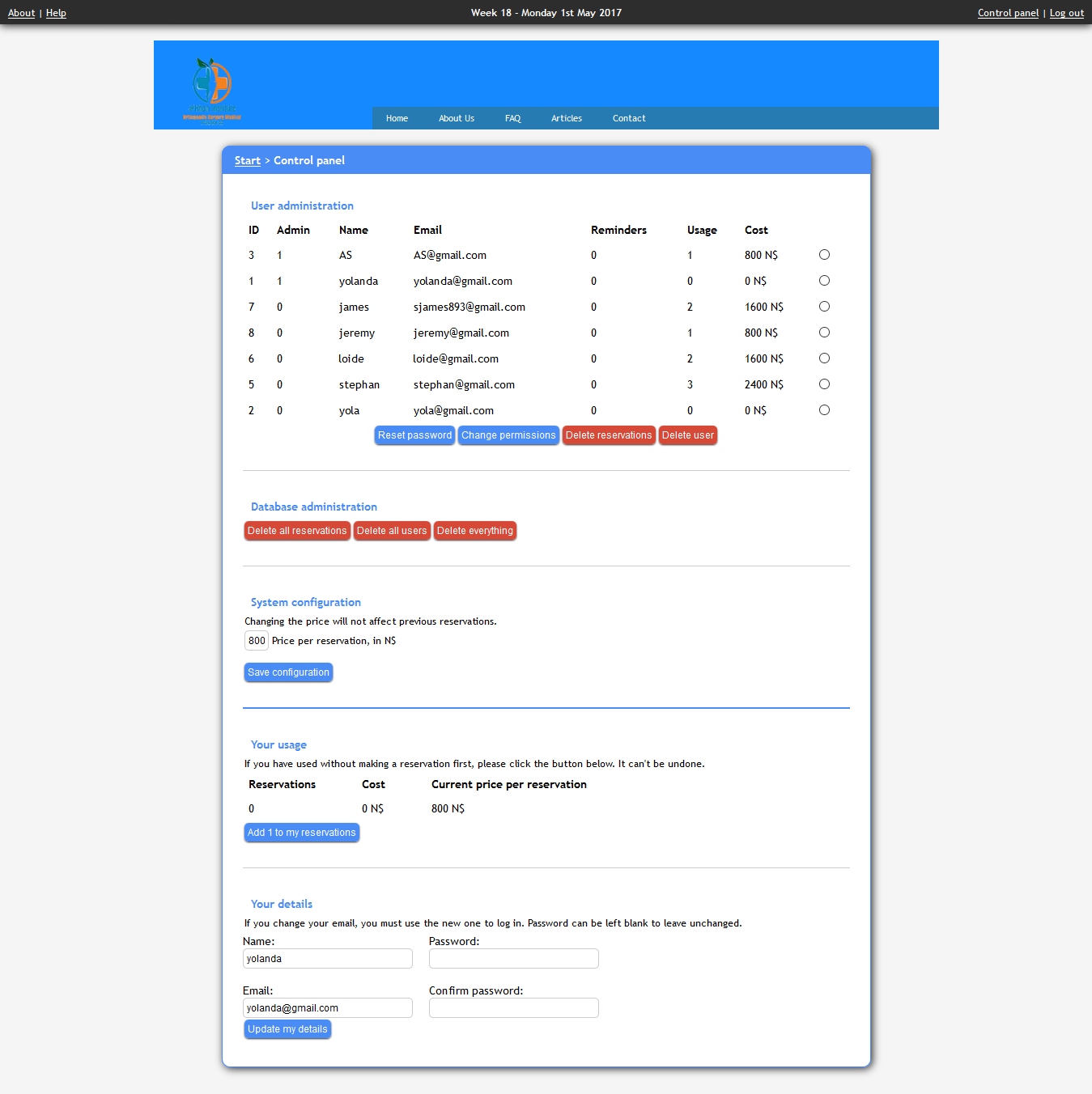
**5.8. User Booking Page**

Users are directed to this page to make bookings, edit or cancel bookings



**5.9. Admin Page**

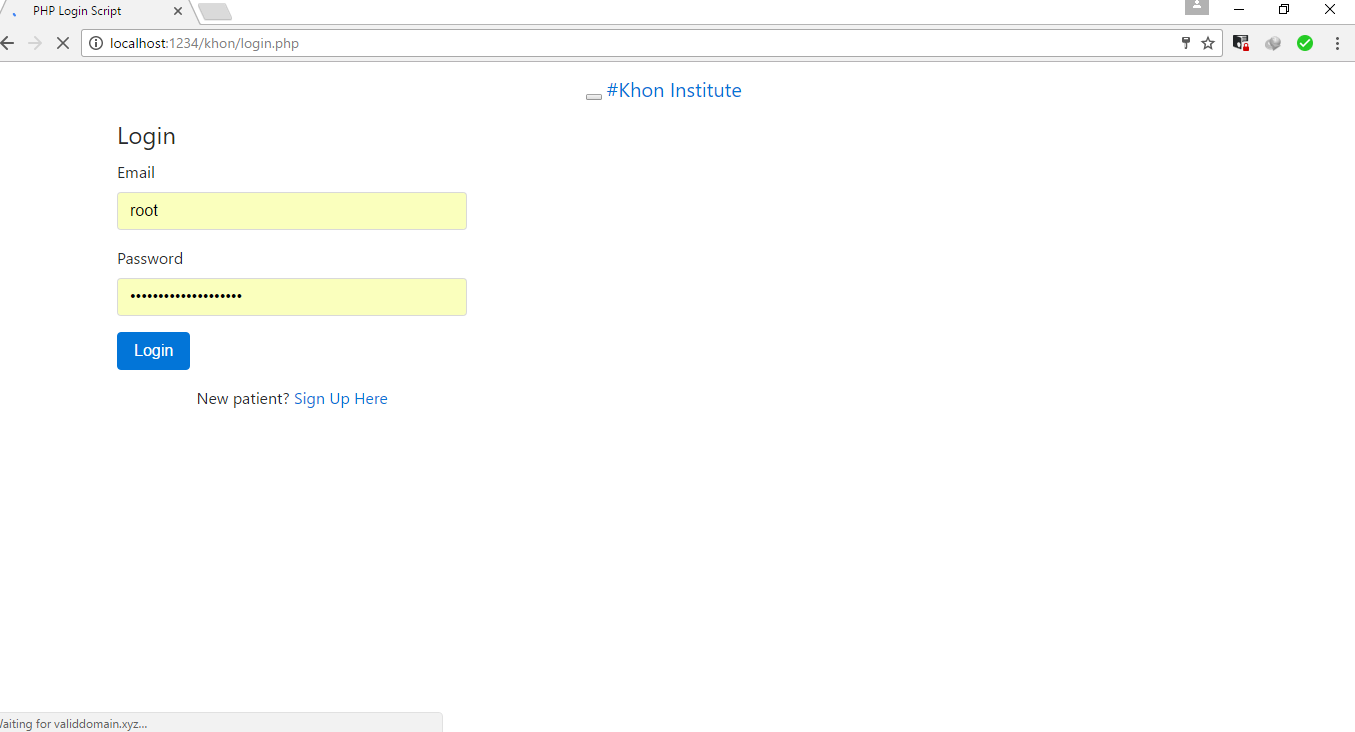
This is the page where the admin can administer the bookings.



**6. TEST PLAN**

**Test Case 1**

* Test Title: Integrating all pages from all designers
* Test Procedures: include hyperlinks on the navigation menu
* Test Data: click on any menu icon
* Expected Result: It will redirect to the selected page.
* Errors encountered were mostly dealing with database connection and with CSSs as shown in the screen shot below.



**Test Case 2**

* Test Title: menu Booking navigation icon
* Test Procedures: Click on the icon.
* Test Data: Users need to click on the icon bar.
* Expected Result: It will redirect to the system home page.

**Test Case 3**

* Test Title: Patient/User & Admin Login.
* Test Procedures: Type email & password.
* Test Data: email & password must be in alphanumeric. Otherwise system will show error (ex. Please enter valid alphanumeric data).
* Expected Result: It will redirect to login page.

**Test Case 4**

* Test Title: Sign Up
* Test Procedures: Click sign up page.
* Test Data: Input customer’s information and click on the button “SIGN UP”.
* Expected Result: It will register new patient.

**Test Case 5**

* Test Title: Search.
* Test Procedures: Type destinations name with selecting check-in and check-out.
* Test Data: Valid destinations name with date.
* Expected Result: System will search according to customer’s choice.

**Test Case 6**

* Test Title: Manage Booking.
* Test Procedures: Click on the link “Manage Booking”.
* Test Data: Add, Edit, Del & Save button.
* Expected Result: Staffs can edit, add and Del individual customers booking records.

**Test Case 7**

* Test Title: Username & Password
* Test Procedures: Enter customers or staffs valid username (Upper Case and Lower Case) and password (Alphanumeric).
* Test Data: Invalid password will show the warning message (Please Enter Valid Password).
* Expected Result: The system will follow the validation pattern.

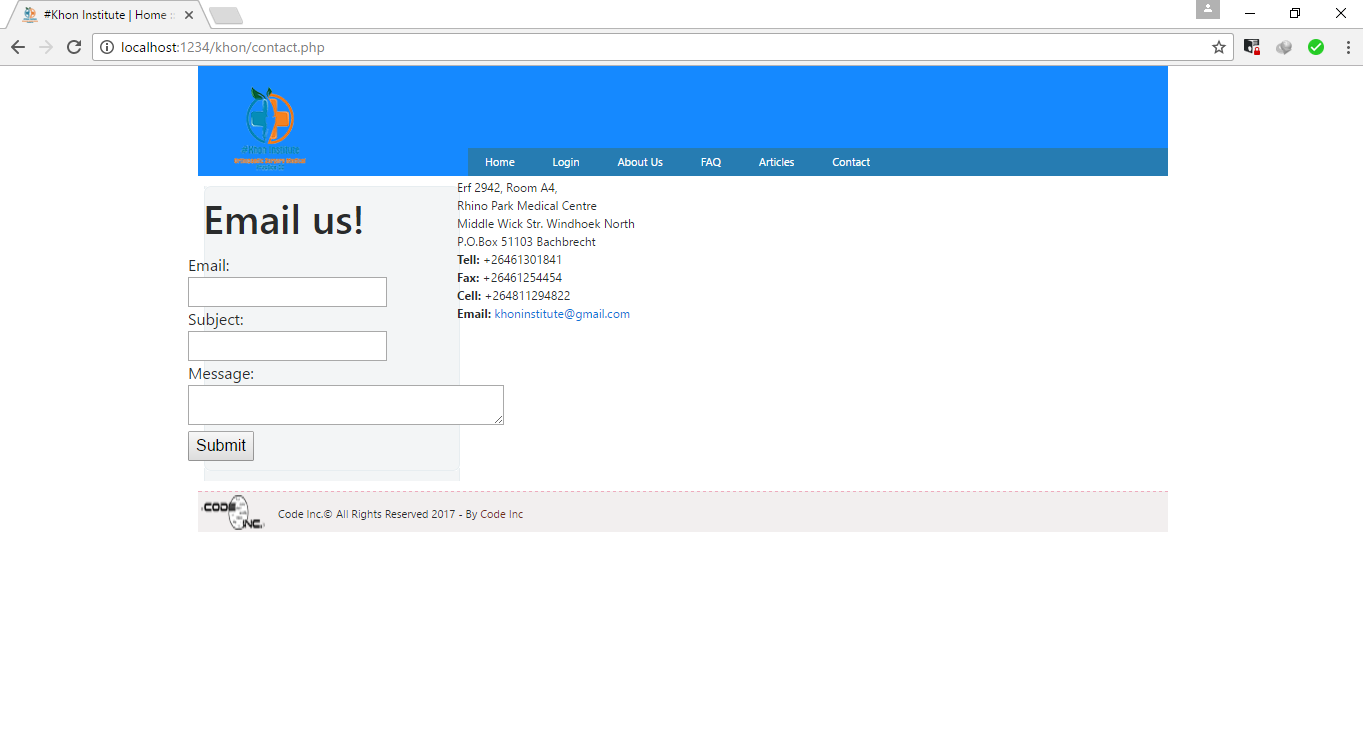
**Test Case 8**

* Test Title: View Booking Cancel Button
* Test Procedures: Customers need to sign in and can cancel the booking when they press cancel button.
* Test Data: Cancel their particular booking record.
* Expected Result: Will cancel their booking records from database.

**Test Case 9**

* Test Title: Contact page
* Customer needs to view the information on how to get hold of the institution
* Test Data: View the page
* Expected Results: The page should display proper and fully fledged page with CSS integrated

THE ERROR:

****

**7. IMPLEMENTATION**

System implementation is the most important steps in case of finalizing the approved web system. We need to justify some basic requirement (software & hardware) so that the system will work without having obligation and customers dissatisfactions.

**7.1. Software Requirement:**

* Operating System: Windows (XP, 7, 8, 8.1) or Mac OSX (Tiger, Leopard, Snow Leopard, Lion, Yosemite).
* Web Browser: Google Chrome, Internet Explorer (ver. 8 or later), Mozilla Firefox, Safari (Mac).
* Database Management System: MySQL, SQL Server, Microsoft Access, Oracle.
* Web Development System: Visual Studio 2010 or later, Adobe Dreamweaver, Atom, Notepad, and Notepad++.
* Others: .NET FRAMEWORK.

**7.2. Hardware Requirement**

* RAM: Minimum 1GB or higher.
* HDD: Minimum 50 GB.
* Processor: Intel Pentium 4 or AMD.
* LAN: Version 1.6.6.406(For fixing up client disconnection).
* Computer

**8. FURTHER DEVELOPMENT**

Given that the project was developed incrementally, there are still requirements emerging and some still to be completed such as:

Validate email: to avoid users to login using fake emails, to register a link must be sent to their email.

Allow only the patients that have visited the practice already and have a patient number register.

Edit the registration page, as there are data that the admin requires from users e.g. country of origin.

**9. CONCLUSION**

The #Khon project was incrementally successful. Despite the challenges faced such as meeting the client due to a busy schedule of the client, Code Inc. managed to meet the main requirements in time, users can register, login, book, in their own convenience and most of all admin can manage all bookings system at any time and in a very simplified manner. The project though performs the main functionalities is still to be further developed as it uses an agile model thus there is enough room for further development.

**10. BIBLIOGRAPHY**

W3School.COM, 2017. W3School *.* [Online]

Available at: https://www.w3schools.com

Booking.com, 2017. *Booking.com.* [Online]

Available at: http://www.booking.com/index.html?aid=378266;label=booking-name-

sONOsHFUDs3qkZY8eHVLqgS35420572471:pl:ta:p1:p2544,000:ac:ap1t1:neg;ws=&gclid=Cj0KEQjwq52iB RDEvrC12Jnz6coBEiQA2otXAp8sdf2xl13y5p6ad6X7mKjuGFwi8-0hZO3yvMK1P9QaAvtt8P8HAQ

EASYTOBOOK.COM, 2017. *EASY TO BOOK.* [Online]

Available at: http://www.easytobook.com/?amu=2599051302&utm\_nooverride=1 Expedia AU, 2017. *Expedia AU.* [Online] Available at: http://www.expedia.com.au/Hotels