**Student (s) Number as per your student card:**

**\_\_\_\_\_**

**Course Title:** MSc Information Systems with Computing (January 2019)

**Lecturer Name:** Paul Laird

**Module/Subject Title:** Enterprise Information Systems

**Assignment Title:** Green Health Doctor Appointment System

1. **Introduction**

Reviewing the problem of having to appear physically at a given clinic or health centre in order to book appointments before knowing when the doctor would be available and what time and day you would meet them is extremely strenuous to be even be thought of. This has led to the creation of different online systems and hence the creation of Green health.

There is basically a thousand and one features that could be implemented to solve a large range of problem in this field but for the sake of simplicity and time constraint which would be discussed later on this report, only a fraction of Information systems has been implemented. Nevertheless this is a very reasonable amount as it targets the latest technology as well.

Green Health is a system built to enable anyone to be able to find a doctor near them and not just that but also book appointments with ease. When you think about the green health system, think of a world of medicine where doctors are registered into the system irrespective of their affiliated hospital

1. **Aims and Objectives**

The aim of the project undertaken is to develop a basic project/system where patients can register and view doctors that are available and then book an appointment with them, depending on whether the time slot is free or not. The data and information is to be stored and easily retrieved at any giving point in time.

At least one of the technologies stated in the requirement is to be used in addition to either the creation or functionality of the system.

1. **Methodology**

The green health system was built adopting different technologies which would be expatiated in this chapter. This technologies include the Asp.Net Core MVC, Bootstrap, Cascading Style Sheet, Entity Framework Core, SQL Database, OAuth and Azure Cloud Server

* 1. **Asp.Net Core MVC**

The green health system was designed using visual studio 2019 and the asp.net core mvc framework was used. The programming language that was used with asp.net core is the c-sharp programming language. Asp.Net has been a radical shift for major developers using the Microsoft platform to create technologies or apps as the case may be. The word MVC stands for Model-View-Controller.

ASP.NET Core MVC is a web application development framework from Microsoft that combines the effectiveness and tidiness of model-view-controller (MVC) architecture, ideas and techniques from agile development, and the best parts of the .NET platform (Freeman, 2016)**.**

Before the emergence of the Asp.net Core, there were others like Asp.Net webforms. Asp.Net and .Net Frameworks which had their own limitations even though we wouldn’t go into detail in this chapter.

Asp.Net Core has assemblies that supports it in functioning well. For instance Asp.Net Core Identity was used to handle user session, login and registration.

* 1. **Entity Framework Core (For Accessing Database)**

Entity framework is an object-relational database that helps transport data from the Asp.Net application to an sql database or whichever database one is using. Entity Framework Core has one key task: storing.NET objects in a database and retrieving them again later. Put another way, Entity Framework Core acts as the bridge between an ASP.NET Core MVC application and a database (Freeman, Pro EntityFramework Core 2for ASP.NETCore MVC, 2018)**.** Storing objects in database can be exhausting and really complex but the Entity Framework makes this easy either with the database first approach where you create the database first and then it creates the database tables for you or the code first approach where the database and the tables are created with ease from code. So no need to write a dot net code and an sql code separately. In information system, data entry, management and retrieval is very important, so therefore entity framework was adopted in this project as the best model for this.

* 1. **Azure Cloud Host**

Based on requirement for this project the Microsoft Azure has been chosen for the cloud service in order to allow ease of information transfer and delivery. This cloud service makes the application to be accessible at any given point in time. Microsoft Azure is Microsoft's cloud computing platform, providing a wide variety of services you can use without purchasing and provisioning your own hardware. Azure enables the rapid development of solutions and provides the resources to accomplish tasks that may not be feasible in an on-premises environment (Michael Collier, 2016).

* 1. **Authentication: OAuth**

The OAuth is an authentication framework that enables third-party applications to acquire limited access to a web service. As an open protocol, the OAuth allows secure authentication in a simple and standard method from mobile, web or even desktop applications

1. The *application* requests authorization to access service resources from the *user*
2. If the *user* authorized the request, the *application* receives an authorization grant
3. The *application* requests an access token from the *authorization server* (API) by presenting authentication of its own identity, and the authorization grant.
4. If the application identity is authenticated and the authorization grant is valid, the *authorization server* (API) issues an access token to the application. Authorization is complete.
5. The *application* requests the resource from the *resource server* (API) and presents the access token for authentication.
6. If the access token is valid, the *resource server* (API) serves the resource to the *application*



**Figure 1.0 – The API Flow Diagram**

1. **System Design and Analysis**

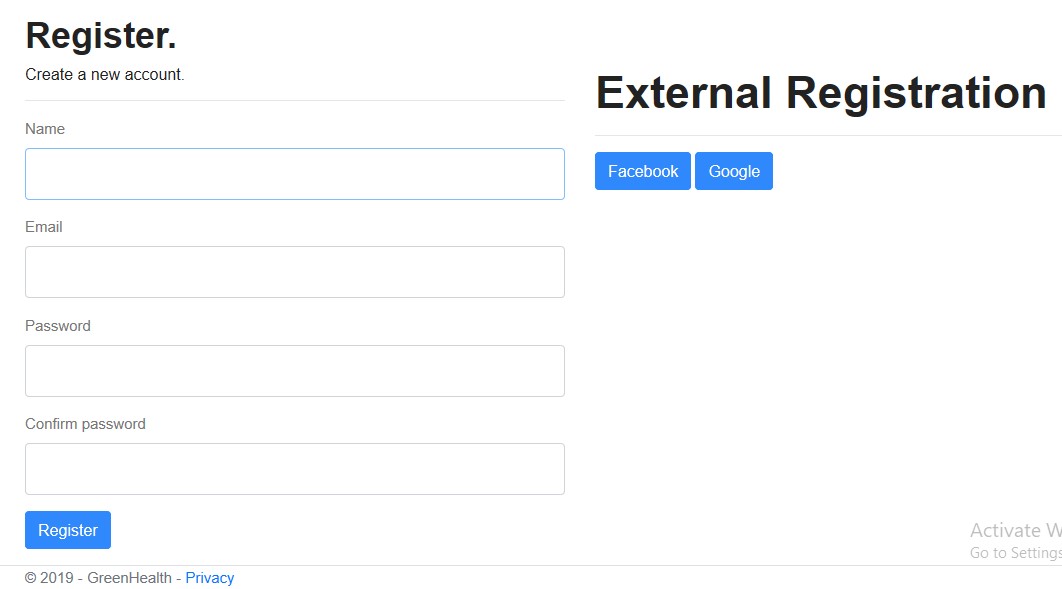
The system design and analysis is been discussed in this chapter. Only the subjects relevant to the course are talked about in depth.

* 1. **Registration**

For every website that require users to provide their data and also have at least a little privilege of managing their data, an account registration is required. Users who wish to be part of the system would provide details requested by the system’s administration and these inputs verified from checking if the fields are empty down to checking if an account like that exists in the system’s database. The green health system has adopted the account registration approach to allow users register and verify themselves.

Only Patients are advised to register on the normal registration page, while the admin is in charge of registering and verifying the doctor. Patients are sent confirmation emails upon registration which they have to verify before allowed to log into the system and make use of the system.

Upon patient registration, they are given a role known as “Patient”. Admin role is seeded into the database by default and once a doctor is registered, the get the “Doctor” role. These various role gives better room for authorization.



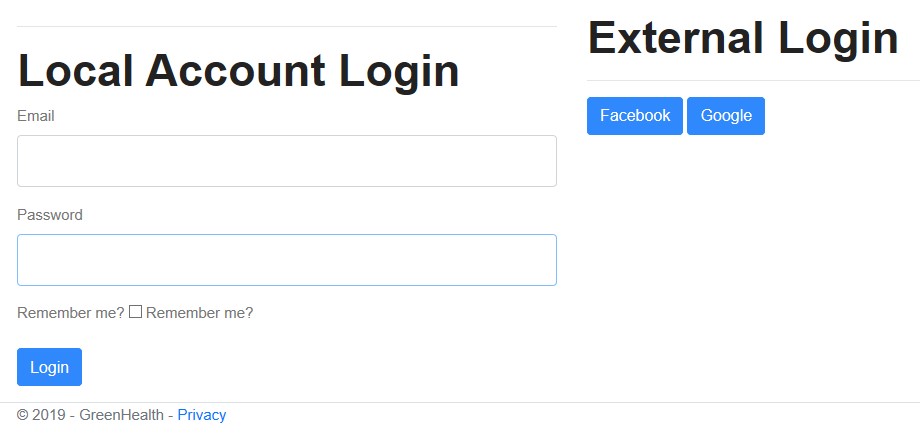
**Figure 1.1 – Account Registration**

* 1. **Login**

The Login system as adopted by green health, allows users who have been registered in the system to enter their credentials and then having gone through authentication are allowed to access their profile page, down to booking an appointment with an available doctor.

If email and password combination are wrong, the error is displayed to the user and if email is not confirmed, the user would be notified and asked to confirm their email address before logging into the site.

Upon successful login, the roles “Admin”, “Doctor”, and “Patient” are redirected to different pages. This is called page redirect based on roles.



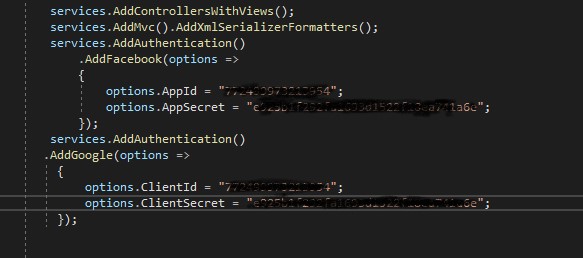
**Figure 1.2 – Account Login**

* 1. **External Registration and Login**

Two external social logins have been added at the time of compiling this report. The code was written to be able to take as many external logins as possible. I have added Facebook and Google buttons to both the login and registration page. These as been made possible through Facebook and Google’s Application Programming Interface (API) which are open source.

On the registration page, once any of the button is clicked, the API tries to fetch information

from its site and compares it with the information in the database, if the account exists and the email has been verified, the user is logged in, else the user information is inserted into the database. To login without the externa logins, they have to update their password when signed in.

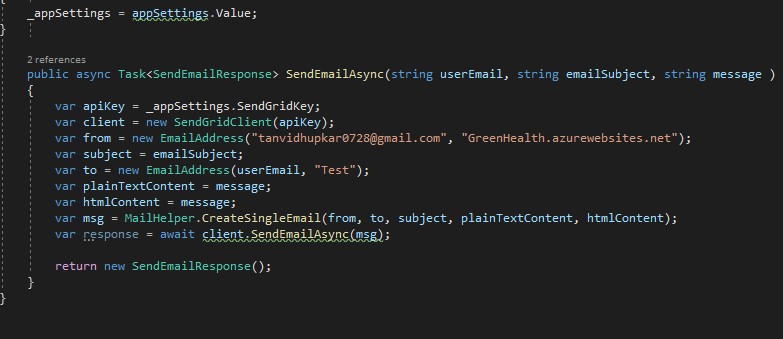


**Figure 1.3 – Adding External Logins**

* 1. **Email Confirmation**

The email confirmation was adopted to make sure user’s email addresses are valid. Send Grid was used as the SMTP provider. A free account was registered and then a User ID and a secret key was given to be able to access the API that allows the green health system to be able to send mails to valid addresses.

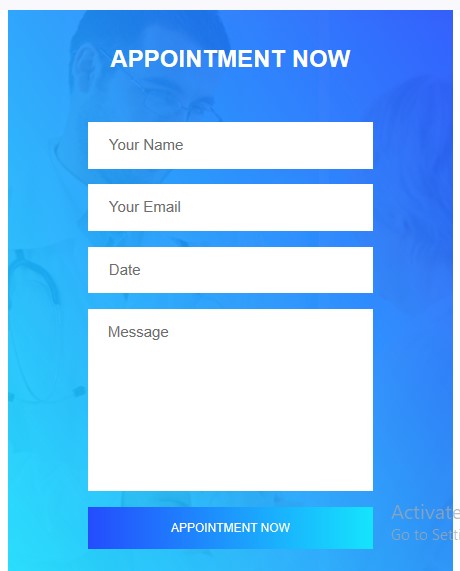
The user receives the email and clicks on the provided link and then a message stating the email has been verified is displayed.



**Figure 1.4 – Email Confirmation Query**

* 1. **Email Appointment Booking**

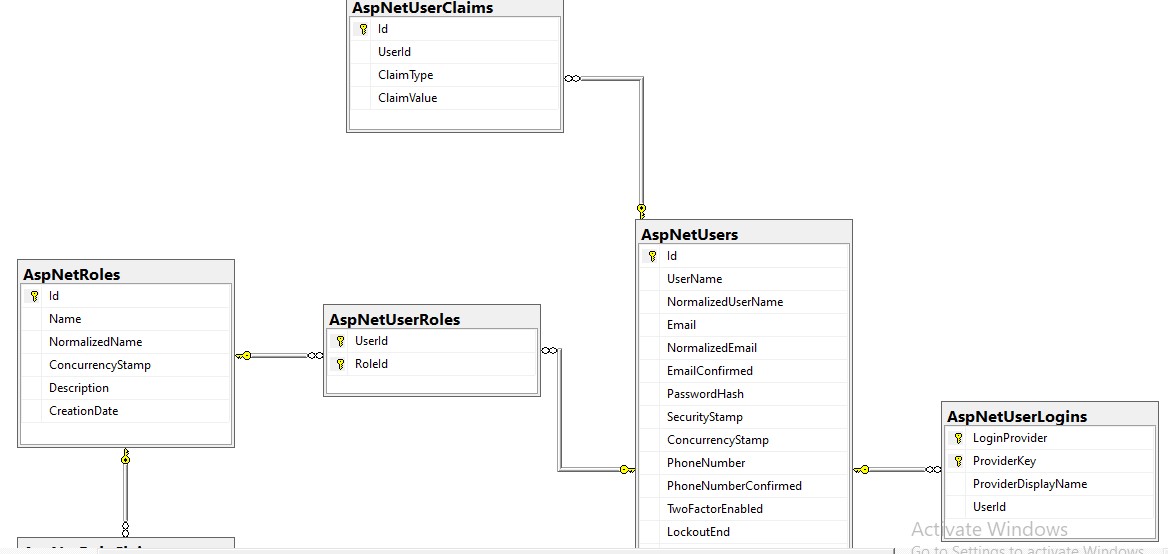
Users who are registered or not can book appointment via email, the admin receives the mail and sets up an appointment for them. A normal SMTP is been used for this approach.

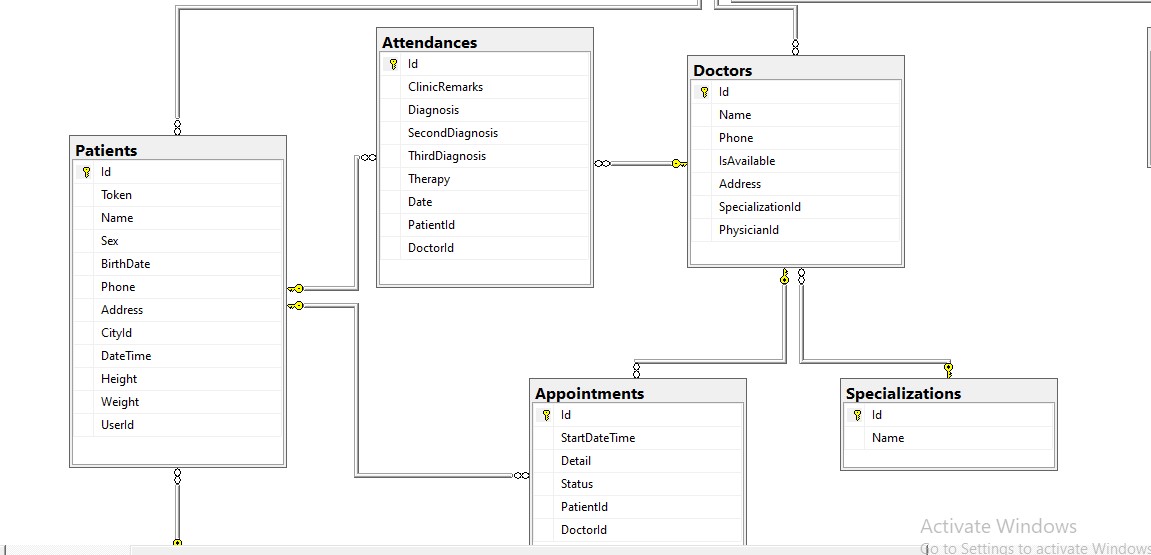


**Figure 1.5 – Book Appointment By Email**

* 1. **The Database**

As discussed in chapter 3.0, a database is a very essential part of a given system that requires storing and retrieving of information. The SQL database was used in the SQL Server Management Studio environment to achieve basic operations regarding insert, delete, update and retrieve. A diagram is shown below to give a glimpse of how the database tables are connected.





**Figure 1.6 – The Database Structure**

1. **Limitation / Conclusion**

In a proper scenario of building a system like green health, a lot of time is required to put the system in good shape and function. Not just time but also enough knowledge of the frameworks and programming languages used.

A few challenges were faced while building the system and these challenges include: my level of expertise in the above stated programming language and framework environment. However, there were lots of online researches done to achieve what was achieved.

Although the system wasn’t built completely as per proposal due to the above stated challenges but the technologies like OAuth, SMTP and Azure Cloud as per requirements were adopted in this project.

# References

Freeman, A. (2016). *Pro ASP.NET Core MVC.* New York: APress.

Freeman, A. (2018). *Pro EntityFramework Core 2for ASP.NETCore MVC.* London: APress.

Michael Collier, R. S. (2016). *Fundamentals of Azure.* Microsoft Press.

1. **RESOURCES**

<https://www.webopedia.com/TERM/W/Web_Services.html>

<https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>

<https://stackify.com/iis-web-server/>

<https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx>

**7.0 APPENDIX**

