**Thyroid-SPOT**

**Handover Document**

**Table of Contents**

[1. General Information 2](#_Toc51867255)

[a) Project Overview 2](#_Toc51867256)

[b) Technologies Used 2](#_Toc51867257)

[2. Technical Details 3](#_Toc51867258)

[a) Types of Users 3](#_Toc51867259)

[b) Pages and Components 4](#_Toc51867260)

[c) Database Schema 7](#_Toc51867261)

[3. Miscellaneous 8](#_Toc51867262)

[a) Participants Involved 8](#_Toc51867263)

[b) Project Credentials 8](#_Toc51867264)

# 1. General Information

## a) Project Overview

**Written in both ASP.NET C# and jQuery/JavaScript, Thyroid-SPOT is a one-stop website that serves as a bridge between patients dealing with thyroid-related illnesses and endocrinologists (clinicians).. This web application features a mobile responsive design that enables the website to be conveniently used on iOS and Android devices.**

**As a medical app dedicated to patients having thyroid disease and clinicians who specialise in handling such illnesses, this app keeps track of patients by their thyroid readings using FT4 and TSH hormone values. Using these values and a patented and clinically validated algorithm, the app calculates the best optimal dosage chart which clinicians can use to prescribe better medication for their patients. Clinicians can further manage the patient info such as editing incorrect inputted FT4/TSH readings by the patient, and assisting patients in identifying correct medication through images. In essence, this app aids in analysis and allows both groups of users to stay organised and be on track with the treatment progress**

**In addition, in the web app, besides patients and client users, there are admin users. The admins can manage the content and users of the website directly on the website itself. With this feature, the management of the website can be done without having to alternate to an external site.**

## b) Technologies Used

* **Microsoft Azure SQL database**
* **MSSQL Server Management Studio 18**
* **ASP.NET web API (v5.2.7)**
* **ASP.NET entity framework (v6.2.0)**
* **Google Firebase**
* **jQuery (v3.5.1)**
* **HTML**
* **CSS**
* **Bootstrap 4 (v4.3.1)**
* **Moment.js (v2.27.0)**
* **Chart.js (v2.9.3)**
* **Print.js (v1.3.1)**

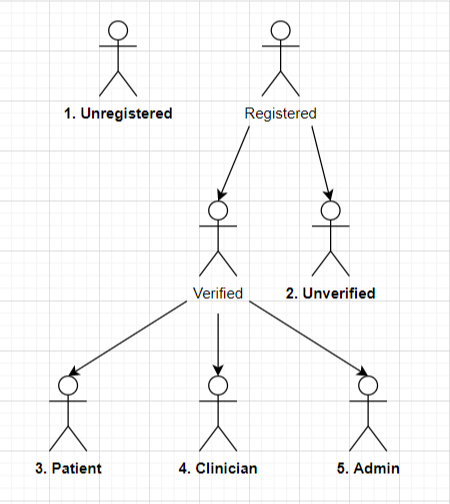
# 2. Technical Details

## a) Types of Users

Thyroid-SPOT is developed to be used by multiple different types of users, all on a single website. This is done using extensive access controls, which checks a user’s current “state” before determining what modules and components can be seen/used by him/her.

In total, there are five different possible states. The first state is the state of unregistered users. Users that are not logged in are still able to access the site and browse some of its features, albeit not much (e.g., unregistered user can read threads and comments, but will not be able to post any). The second state is when users are registered and logged in but have yet to verify the email used for the creation of the account.

The third, fourth, and fifth states are Patients, Clinicians, and Admins, respectively. In these three states, the user is logged in, and the email used for the account is verified. For patient users, they can simply create an account in the account.html page, and by default, the account created will be tagged with a “patient” tag. As for clinicians and admins, only existing admins are able to create new clinician and admin accounts (the emails used for the creation of the accounts still have to be verified).



## b) Pages and Components

|  |  |
| --- | --- |
| **Pages** | |
| index.html | **Access: All types of users.**  This is the home page of the website. |
| about.html | **Access: All types of users.**  This page is where all users can read more information about A\*STAR. For example, A\*STAR’s mission, values, and vision. |
| account.html | **Access: Users that are not logged in.**  This page is where users can either log in to an existing account or register an account (as a patient). |
| disease.html | **Access: All types of users.**  This page shows information about all thyroid-related diseases, such as a disease’s symptoms, causes, and treatments. In this page, Admin users can create, update, and delete diseases. |
| disease-forum.html | **Access: All types of users.**  This page shows up when a user clicks on the forum of a specific disease. Although there is a forum system for each disease, the forums (thread & comments) are not hardcoded. This means that if a new disease is added, it will automatically have its own forum system.  Although all types of users can access this, some have more possible functionalities than others. Users that are logged in can create new threads. Upon creation, they can also edit or delete their own threads. Aside from the above functions, Admins can delete anybody’s thread. |
| disease-comment.html | **Access: All types of users.**  For each thread, there will be a unique comment section (not hardcoded; it is added automatically and dynamically). This means that if a new thread is added, it will automatically have its own comment system.  The access controls for comments is the same as threads. |
| verify-email.html | **Access: Users that have an account that is not verified.**  Upon logging into an account, if the email used for the creation of the account is not verified, the user will be redirected to this page. Until the user either verify his/her email or logs out, he/she will be stuck on this page (the user can try changing the URL directly, but it will not work). |
| profile.html | **Access: All types of logged-in users with verified emails.**  This is the profile page for Patients, Clinicians, and Admins. These three users can change their emails and password or delete the account from this page.  Patients will be able to see an additional tab that contains their patient information. They can edit their patient information. |
| patientinfo.html | **Access: Patients only (logged in & verified).**  This page is only accessible by Patients that are logged in and have their email verified. On this page, Patients can see a more detailed picture of their current health, including a graph. Patients can create reports to store their latest TSH and FT4 values. |
| all-patient.html | **Access: Clinicians only (logged in & verified).**  This page is only accessible by Clinicians that are logged in and have their email verified. On this page, Clinicians can see the list of the patients that are attached to them. They can also add new patients to the list. |
| patient-info-doctor.html | **Access: Clinicians only (logged in & verified).**  This page is only accessible by Clinicians that are logged in and have their email verified. This page is similar to the patientinfo.html, but the view is of the Clinician’s perspective. Clinicians can also add or update their patients’ reports. |
| manage.html | **Access: Admins only (logged in & verified).**  This page is only accessible by Admins that are logged in and have their email verified. On this page, Admins can remove Patients and Clinicians. They can also create new accounts for Clinicians and Admins (cannot create for Patients). |
| **Components** | |
| navbar.html | This is a navbar component which is imported and used in every page (except for verify-email.html). This component is filled with access controls that deal with the routes. For example, a user that is not logged in will not be able to enter profile.html even if he/she enters “profile.html” on the URL directly.  The elements in the navbar change, depending on the state of the current users. For example, the tabs an admin can see are “Home”, “Profile”, “Disease” and “Manage”. On the other hand, the tabs that a user that is not logged in can see are “Home”, “Account” (where user can log in or signup), and “Disease”. |
| footer.html | This is a footer component that is imported and used on every single page of the website. |

|  |  |
| --- | --- |
| **JavaScript files that are shared across multiple pages** | |
| firebase-init.js | This JavaScript file is imported to navbar.html, which then gets imported to every page (except for verify-email.html). This file is used to store the configuration and the initialization of Firebase so that Firebase Auth can be utilized. |
| firebase-log-state.js | Similar to firebase-init.js, this file is imported into the navbar. This file contains codes that revolve around the setting (initialization), storing, and removing of states (done in every page on initialization). Firstly, it checks if the user is logged in. If yes, then it checks if the email is verified. If yes, it attaches the user’s role/permission (e.g., admin or patient) to the user’s state. |

## c) Database Schema

# 3. Miscellaneous

## a) Participants Involved

Under the supervision of Mrs. Fang Ying Huey, the Thyroid-SPOT web application is developed by:

* Chua Chee Hean ([1806562C@student.tp.edu.sg](mailto:1806562C@student.tp.edu.sg))

Chee Hean developed the website’s patient-side functions, thyroid disease page and forum features (e.g, threads, comments).

* Ong Jia Chun ([1802338B@student.tp.edu.sg](mailto:1802338B@student.tp.edu.sg))

Jia Chun developed the website’s access controls, authentication, the admin module, and parts shared across all types of users (e.g., navbar, footer, profile, index, and about).

* Yu Kai Long ([1801188B@student.tp.edu.sg](mailto:1801188B@student.tp.edu.sg))

Kai Long developed the clinician modules. FEEL FREE TO ADD MORE

Feel free to reach us via the emails stated above should there be any concerns or queries.

## b) Project Credentials

Gmail account where the Firebase Auth resides

* Email: **mp.thyroid.spot@gmail.com**
* Password: **A321123z**

MSSQL Server Management Studio

* Server Name: **mp-thyroid-spot-server.database.windows.net**
* Login/Username: **user**
* Password: **A123456z**

The server and database are hosted on Azure (so that the three developers can share the same database) using Jia Chun’s school email address. Do let him know if the exportation of the database is required (basically if the existing data in the current database and server is needed).

**(End of Document)**