



MAINTENANCE DOCUMENT

TP11 – FiveS

VicEnerG

Version 4.0

Chai Lam Loi 28136179
Hui Tao 30296439
Lang Sun 30662826
RongXuan Li 31089429
Zhe Zhang 29960762

Table of Contents

1. Introduction	3
2. System Description	3
2.1 System Application	3
2.2 System Organization	4
2.2.1 Internet	4
2.2.2 Firewall	5
2.2.3 Secure Virtual Machine	5
2.2.4 MYSQL database server	5
2.2.5 Web application server	5
2.3 System Security	5
2.3.1 Security Implementation	5
2.3.1.1 AntiForgery Token	5
2.3.1.2 ViewModel Validations	5
2.3.1.3 Firewall Implementation	6
2.3.1.4 Error Page	6
2.3.2 Privacy	6
2.3.3 Communication Encryption	6
3. Project Description	7
3.1 Equipment Environment	7
3.1.1 Target Environment	7
3.1.2 Development Environment	8
3.1.3 Hosting Environment	8
3.1.2.1 Web Application	8
3.1.2.2 Database	9
3.2 Support Software	10
3.3 Database Characteristics	11
3.3.1 Open data details	12
3.3.2 Initial Data Setup:	14
3.3.3 Data Increment:	14
4. Error Conditions	15
5. Testing	15
5.1 Testing Information	15
5.2 Documentation Testing Carried Out	16
5.2.1 Backup and Recovery Testing:	16
5.2.2 Usability Testing:	16
5.2.3 Unit Testing:	16

5.2.4 Pressure test	17
6. Source Code and DB Script	18

1. Introduction

This report will be the reference to technical staff to maintain the VicEnerG system for further improvement. The maintenance document will introduce the necessary information and define the tasks and activities for staff. It will provide the most detailed description of the system, including system application, organization, and security. The project will also contain the Equipment Environment, computer hardware, support software, and relevant database. The last thing in the document that needed to be informed by the maintenance staff is testing; detailed test plans and results will be provided.

2. System Description

2.1 System Application

The purpose of the system is to help users save energy in their daily life through knowing carbon footprint knowledge and using solar, water and carbon calculators.

- Solar calculator, as one of the main functions of our website, can provide users with solar energy output, comparison bagplot, and application usage graph.
- Water calculator is the other main function, which can provide monthly water saving to users through inputting their address or drawing maps.

The website supports customer service, like we will receive customer complaints/issues through contact emails and resolve the issues which they propose.

The system contains some information pages, which have texts and graphs. The data used for visualization is downloaded from CLEAN ENERGY REGULATOR. The file contains two types of data: string and integer. The data is saved in a CSV file and is stored in the google drive.

Data Source: <http://www.cleanenergyregulator.gov.au/RET/Forms-and-resources/Postcode-data-for-small-scale-installations>

2.2 System Organization

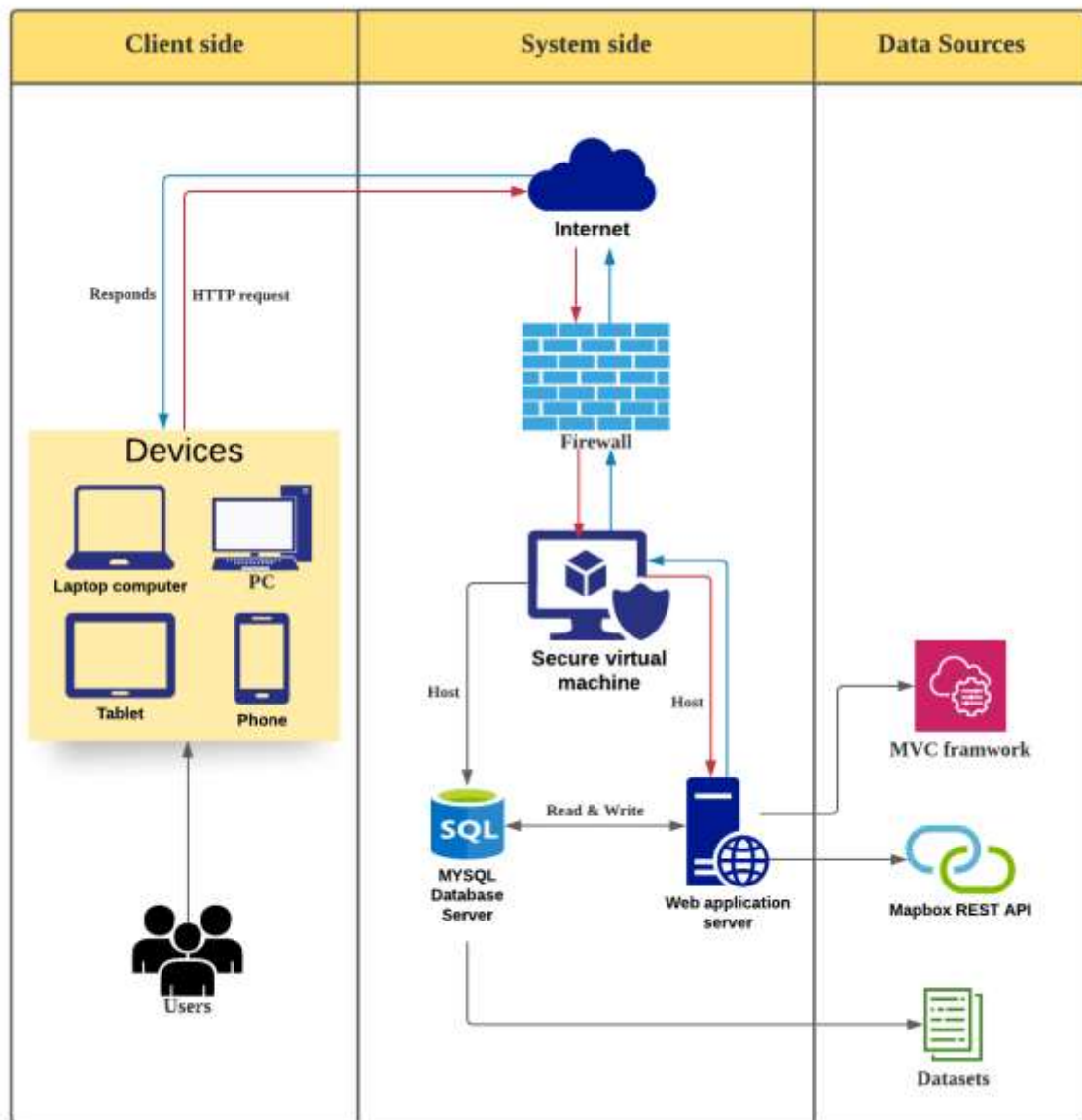


Figure. 4th Version of System Diagram of VicEnerG

The figure above is the system diagram of VicEnerG which contains three layers: Client-side, System side and Data Sources. The users from the client side will access the system through the web browser of their devices. The system side consists of **5 major components**. Each component will be described in the following:

2.2.1 Internet

The internet is the place that transmits the requests and responses between the client side and the system side. Both the client and system sides will use IP addresses to communicate and exchange data. The request and response for the client-side will be handled by the web browser of their devices.

2.2.2 Firewall

This firewall is provided by an Azure virtual machine. The firewall would prevent the users from accessing the backend of the web application which they are not supposed to be. The firewall is initialised with malicious IP detection and DDOS prevention.

2.2.3 Secure Virtual Machine

The secure Virtual Machine is hosted by the Azure platform. The virtual machine is used to host both the Database server and Web application server. Please refer to the 3.1.3 hosting requirement, to check out the minimum requirement of the virtual machine.

2.2.4 MYSQL database server

The system has used an MYSQL database server to store the website dataset. The website dataset is collected from the Open dataset and wrangled by the data scientist. Please refer to the Support documentation located at the 'Handover Document Folder' for data collection, wrangling.

2.2.5 Web application server

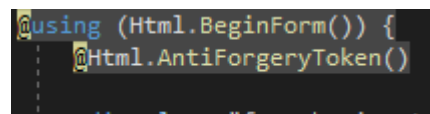
The web application server is used to host the web application. The web application is developed using the ASP.NET MVC framework and has used Mapbox REST API to implement map drawing features. Please refer to 3.2 support software to check out what software has been used in the web application.

2.3 System Security

2.3.1 Security Implementation

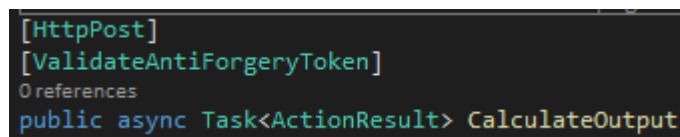
2.3.1.1 AntiForgery Token

The inputs required by the client are the postcode, roof size and number of panels required. Hence, input validation has been performed using the ASP.NET AntiForgeryToken feature to prevent the input injection such as SQL and XSS injection. The system has placed an AntiForgeryToken at every form of the web application. The purpose for this token is to prevent input injection such as SQL and XSS injection. The Figure2 has shown where the system initialises the AntiForgeryToken in the form. The Figure3 has shown how the controller of the system validates the AntiForgeryToken. If a form that doesn't have a token or has an invalid token has been posted to the system, the controller will reject that form to prevent any injection attacks.



```
@using (Html.BeginForm()) {  
    @Html.AntiForgeryToken()  
    ...  
}
```

Figure. The AntiForgeryToken in html form



```
[HttpPost]  
[ValidateAntiForgeryToken]  
public async Task<ActionResult> CalculateOutput()  
{  
    ...  
}
```

Figure. Controller validating token

2.3.1.2 ViewModel Validations

Moreover, the ViewModel has been utilized with the use of model annotation validation to limit the input of components, hence strengthening the security of the web application by limiting

the input of users. Screenshots below show the validation of the number of panels and postcode number for the calculators:

```
[Range(3000, 3999, ErrorMessage = "Victoria postcode is in between 3000-3999")]
[Required(ErrorMessage = "Postcode is required")]
1 reference
public int Postcode { get; set; }

[Required(ErrorMessage = "Number of panels is required")]
[Range(1, 10, ErrorMessage = "The range of the estimation for solar panels is 1-10")]
2 references
```

Figure. Annotation Validation in ViewModel

```
<div>
    using (Html.BeginForm("CalculateCarbon", "Carbon")) {
        Html.AntiforgeryToken()
        <input type="hidden" id="Postcode" name="Postcode" value=@Model.Postcode />
        <input type="hidden" id="NumberPanels" name="NumberPanels" value=@Model.NumberPanels />
        <input type="submit" class="btn btn-primary btn-lg btn-block btn-carbon-redirect" value="Find out more at Carbon Calcula
    }
</div>
```

Figure. Annotation Validation in Calculator View Page

2.3.1.3 Firewall Implementation

Firewall has been implemented in the hosted VM with default settings.

2.3.1.4 Error Page

The error page debugging information has been turned off. When the user enters invalid URL in our domain, the custom error page will be shown to redirect them to the home page.

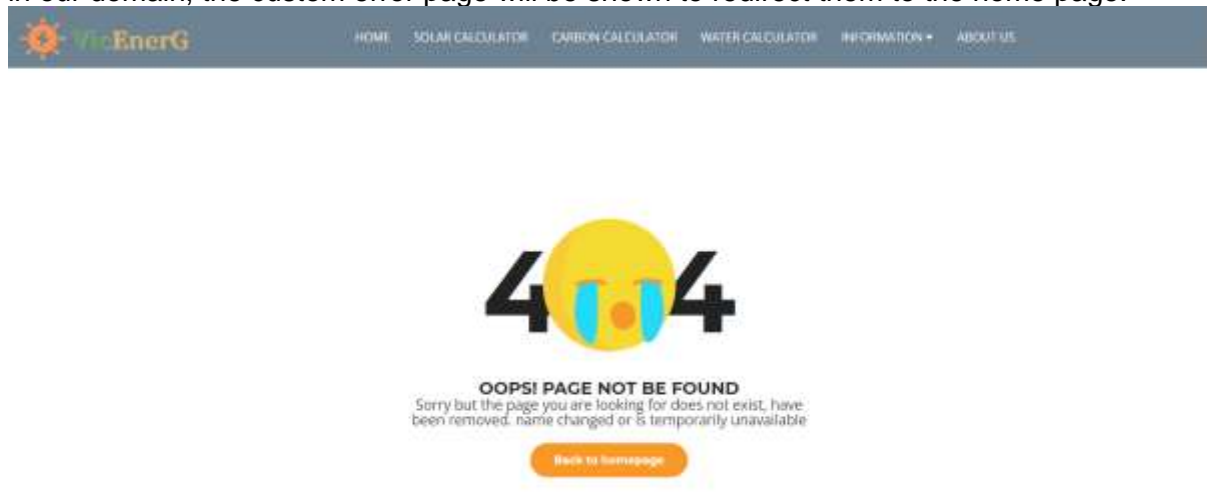


Figure. Error page

2.3.2 Privacy

In addition, the web application has no login session in the system, no cookies and no sensitive information saved in the system, hence no privacy issue for the web application.

The username and password for the VM as well as SQL database must be kept safely to prevent the database or VM breaching. It is recommended not to write the username and password in plaintext form in the code, such as in the Web.config, to increase the security of the web application.

2.3.3 Communication Encryption

For security reasons, the web application must be protected using SSL/TLS communication. This is very important for the security of web applications, and thus, you must implement the

SSL/TLS certificate for the web application if you want to reimplement the web application for the company.

The following link is the steps to setup the SSL/TLS certificate in the VM using KeyVault:
Tutorial: Secure a web server on a Windows virtual machine in Azure with TLS/SSL certificates stored in Key Vault

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/tutorial-secure-web-server>

If you already had a certificate, you can refer to this link to understand how to import the certificates into KeyVault:

Import-AzKeyVaultCertificate

<https://docs.microsoft.com/en-us/powershell/module/az.keyvault/import-azkeyvaultcertificate?view=azps-6.4.0&viewFallbackFrom=azps-4.8.0>

If you don't want to use KeyVault to store the certificates, you can refer to these links to manually setup the certificate:

How to install an SSL certificate on IIS10

<https://www.ssllabs.com/knowledgebase/how-to-install-an-ssl-certificate-on-iis-10/>

How to Install an SSL/TLS Certificate on Microsoft IIS

<https://www.youtube.com/watch?v=nB3TFDkDkcU>

Install an SSL/TLS Certificate in Windows IIS 10

<https://www.ssl.com/how-to/install-an-ssl-tls-certificate-in-iis-10/>

3. Project Description

3.1 Equipment Environment

3.1.1 Target Environment

For desktop, the web application can support the latest versions of most desktop browsers.

	Chrome	Firefox	Internet Explorer	Microsoft Edge	Opera	Safari
Mac	Supported	Supported	Not Applicable	Not Applicable	Supported	Supported
Windows	Supported	Supported	Supported , IE10+	Supported	Supported	Not Applicable

For mobile, the web application supports the latest versions of each major platform's default browsers. Note that proxy browsers (such as Opera Mini, Opera Mobile's Turbo mode, UC Browser Mini, Amazon Silk) are not supported.

	Chrome	Firefox	Safari	Android Browser & WebView	Microsoft Edge
Android	Supported	Supported	Not Applicable	Android supported v5.0+	Supported
iOS	Supported	Supported	Supported	Not Applicable	Supported

3.1.2 Development Environment

This application is being developed and maintained using Microsoft Visual Studio Communication 2019 with ASP.NET Framework 4.8. Due to the specification of framework, only Windows OS can be used to develop and maintain this web application. The following link is the minimum and recommended specification for Microsoft Visual Studio 2019.

<https://docs.microsoft.com/en-us/visualstudio/releases/2019/system-requirements>

Details of required computer hardware list:

	Recommended Requirement	Minimum Requirement
OS	Windows 7 and above	Windows 7 and above
Processor	Quad-core or better	1.8 GHz or faster processor
RAM	8 GB	4 GB
Hard Disk	100 GB	50 GB
Video Card	WXGA (1366 by 768) or higher	Support minimum display resolution of 720p (1280 by 720)

3.1.3 Hosting Environment

3.1.2.1 Web Application

For hosting, this web application is hosted on Azure Virtual Machine using image Windows Server 2019 Datacenter. Since this web application is developed using ASP.NET Framework 4.8, Windows OS must be utilised for hosting the web application. Any VM that supports the Windows Server 2019 Datacenter image is recommended. This is a small scale web application, hence the 1vcpus and 2GiB memory are enough for the web application. The table below shows the recommended and minimum requirement of the VM:

	Recommended Requirement	Minimum Requirement
Image	Windows Server 2019 Datacenter	Windows Server 2019 Datacenter

Size	2vcpus and 2GiB memory	1vcpus and 1GiB memory
Services Installed	Web Management Service, Windows IIS, ASP.NET 4.8, Web Deploy v3.6	Web Management Service, Windows IIS, ASP.NET 4.8, Web Deploy v3.6

The following link is the steps to create a VM on the Azure platform.

Quickstart: Create a Windows server VM with the Azure Stack Hub portal

<https://docs.microsoft.com/en-us/azure-stack/user/azure-stack-quick-windows-portal?view=azs-2102>

The following link includes the step of setting up the VM to host ASP.NET web applications and deploying the web application from Visual Studio to the VM. Note that the framework used is ASP.NET 4.8, hence you should download ASP.NET Framework 4.8 components in your VM, instead of .NET 5.0 shown in the link:

Deploy a C# ASP.NET web app to a VM in Azure Stack Hub

<https://docs.microsoft.com/en-us/azure-stack/user/azure-stack-dev-start-howto-vm-dotnet?view=azs-2102>

3.1.2.2 Database

After setting up the VM, it must communicate to the database so that the web application's features can be run successfully. The Database is hosted on Azure SQL Database. The structure of the database will be discussed in the following section "Database Characteristics". Since the size of the data stored in the database is 32MB, the recommended database size is around 1GB.

The table below shows the minimum and recommended requirements of the database:

	Recommended Requirement	Minimum Requirement
Database Type	SQL Relational Database	SQL Relational Database
Size	2GB	1GB

To connect the web application to the database, the connection string is used. The connection string of the database can be found on the information of that database. The steps to connect the database in Windows IIS manually is provided in the following link with section "3.1. Data Source Connection Strings":

Configuring Step 3: Configure Data Source Settings

<https://docs.microsoft.com/en-us/iis/application-frameworks/scenario-build-an-aspnet-website-on-iis/configuring-step-3-configure-data-source-settings>

Other than that, you can set up the connection string in the Visual Studio publish profile when deploying the web application to the VM. The screenshot of the database publish profile is shown below:

Publish

Connection

Settings

vicenergVM *

Configuration: Release

File Publish Options

Databases

VicEnerG_ModelContainer

Server=tcp:vicenerg.database.windows.net,1433;Initial Catalog=VicEnerG;Persi

☒ Use this connection string at runtime (update destination web.config)

☐ Execute Code First Migrations (runs on application start)

In order to publish a Code First model, Code First Migrations should be used. [Learn more about this](#)

< Prev Next > Save Cancel

You can also follow this link to create a publish profile and enter the connection string:
[How to: Deploy a Web Project Using One-Click Publish in Visual Studio](https://docs.microsoft.com/en-us/previous-versions/aspnet/dd465337(v=vs.110))
[https://docs.microsoft.com/en-us/previous-versions/aspnet/dd465337\(v=vs.110\)](https://docs.microsoft.com/en-us/previous-versions/aspnet/dd465337(v=vs.110))

Custom Domain

If you want to set up a custom domain for the web application, you can refer to this link:
[Add Custom Domain to Azure VM or resource](https://docs.microsoft.com/en-us/azure/virtual-machines/custom-domain)
<https://docs.microsoft.com/en-us/azure/virtual-machines/custom-domain>

3.2 Support Software

The software has been adopted for the web application are below:

No.	Software Name	Purpose	No.	Software Name	Purpose
1	Visual Studio Community 2019, version 16.11	ASP.NET web application development, maintaining and testing environment.	9	Chartjs	Javascript library displaying interactive diagrams

2	Azure SQL Database	Cloud-Hosted Database	10	modernizr 2.8.3	Javascript library for utilizing the browser and the website
3	Azure Virtual Machine	Web application hosting	11	Bootstrap 3.4.1	Javascript library for website UI components
4	Axure	UI/UX Design	12	turfjs library	Javascript library for geospatial analysis
5	Lucid Chart	Architecture and Diagram Design Tool	13	mapbox-gl	Javascript library for display a map on the website
6	Pentest_Tools	Web application Security Testing tool	14	mapbox-gl-draw library	Javascript library for drawing polygons on the map
7	Jquery 3.4.1	Basic Javascript library for HTML	15	jquery.validate	Javascript library for Client-side validation
8	ASP.NET MVC framework 4.8	The framework of the website system	16	C#	Programming language for development

3.3 Database Characteristics

The database used in the system is Microsoft SQL database and its implementation is based on the relational diagram.

Here's ER diagram:

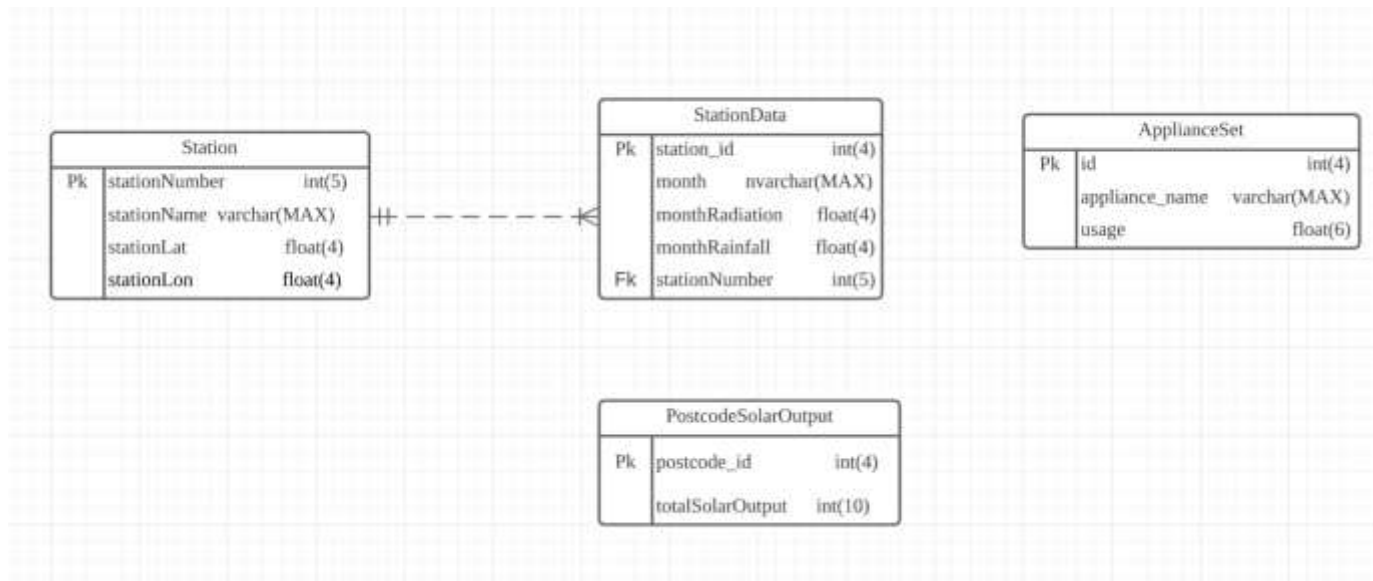


Figure. The ER diagram of the database.

In the database model, there are four tables, which are Station, StationData, ApplianceSet, and PostcodeSolarOutput.

Station table: It contains 150 rows of station data. The primary key is stationNumber, which is the unique id of the climate station. The stationName is the name of the climate station. The stationLat and stationLon provide the location of the climate stations.

StationData: This primary key of the entity is station_id, ranging from 0-1800. The foreign key of the table is stationNumber, which offers a link between Station and StationData. The monthRadiation means monthly average radiation per station in Victoria. And the monthRainfall means monthly average rainfall per station in Victoria.

ApplianceSet: In this table, each application has a unique and special identity number, which is limited to four numbers. The table also has the application name and usage columns. The usage column shows how much the electricity will be used per hour for each application.

PostcodeSolarOutput: The primary key of the table is postcode_id. The totalSolarOutput column offers the total solar radiation of one postcode in a year.

1. [Table - Station] to [Table - StationData] relationship: one to many
2. [Table - Station] to [Table - ApplicationSet]: no relationship
3. [Table - StationData] to [Table - ApplicationSet]: no relationship
4. [Table - Station] to [Table - PostcodeSolarOutput]: no relationship
5. [Table - StationData] to [Table - PostcodeSolarOutput]: no relationship

3.3.1 Open data details

Open datasets

Names	Physical access used	Frequency of source updates	Frequency of ITERATION System updates	Granularity	Copyright details
Request point climate data from 1889 to yesterday https://www.longpaddock.qld.gov.au/silo/point-data/#responseTab2	CSV API	Data daily updated	Daily	The sun radiation in each station by day	https://www.qld.gov.au/legal/copyright Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made .
Postcode data for small-scale installations-SGU-Solar http://www.cleanenergyregulator.gov.au/DocumentAssets/Pages/Postcode-data-for-small-scale-installations---SGU-Solar.aspx	CSV	Data yearly updated	Yearly	The solar installation in Victoria each month	http://www.cleanenergyregulator.gov.au/About/Policies-and-publications/Copyright#1 Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made .

Australia Postcode https://www.matthewproctor.com/australian-postcodes	XLSX	Data yearly updated	Yearly	The postcode in Victoria	https://www.matthewproctor.com/ Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made .
---	------	---------------------	--------	--------------------------	---

3.3.2 Initial Data Setup:

The data was cleaned and wrangled by using python. The script should combine 150 files at first. Each file is the solar radiation data of each station in a year. Then, the data scientist needs to add the latitude and longitude columns for each station. In this stage, he won't use any model to do data prediction. The only thing he needs to do is to calculate the average value of monthly solar radiation for each station.

The data scientists also use R studio to do data cleaning and wrangling for the second data. He needs to select the postcode number in two ranges: 3000-3999 & 8000-8999. Then, he needs to select the installation month from January to December, and sum all of the data frames, to get the installation number in each month in Victoria.

The data scientists use R studio to do data cleaning and wrangling for the second and the third data. They need to select the postcode number in two ranges: 3000-3999 & 8000-8999 from the second data. Then, they need to sum solar output for each postcode in Victoria in 2020. If the postcode number is missing, they should add the postcode to the data and set the default value according to the postcode number in the third data.

3.3.3 Data Increment:

A new table called rainfallData, which contains attributes of daily rainfall, month, stationNumber, should be created. So, the members need to use python to do data cleaning and wrangling for the climate data. They need to combine 150 CSV files again. Each file is the rainfall data of each station in a year. In this stage, they don't need to use any model to do data prediction. The only thing they need to do is to calculate the average value of monthly rainfall for each station.

In the future, more station data may be gathered to expand the system database. People will download the data from the QLD website, do data wrangling and update the station table. At that time, they will add the new data to the database. They will update the climate data and postcode solar_installation data annually.

Generate table from database model (SQL):

3.0 Code and Database Folder > Website Database Modeling > Database modeling.sql

Import data to database (SQL):

4. Error Conditions

Our system is a web-based system that can be only accessed by a web browser with HTTP URL requests of any devices. The most common types of HTTP errors are:

- HTTP Error 500/503 (Internal Server Error)
- HTTP Error 403 (Forbidden)
- HTTP Error 404 (Not Found)
- HTTP Error 400 (Bad Request)
- HTTP Error 401 (Unauthorized)

However, our system does not have an authentication feature (login / logout). Therefore, Error 403 and Error 401 will not happen in our system.

The following table describe the system wide error:

Error type	Root cause	Solution
HTTP Error 400 (Bad request)	The user is submitting scripts or danger information through the information input section of the calculation pages.	Default error page: Ask the user the check their input
HTTP Error 500/503 (Server Error)	Internal system error. The possible factor that causes this error will be: lost connection to the database or users passed invalid data into the system or authentication token of API is expired.	Default error page: Ask the user to report the bug when the errors keep happening
HTTP Error 404(Not Found)	The user is entering the url manually and there is a typo in the url with our domain.	404 error page: Ask the user to check the URL and try again.

The following link is an instruction on how to handle HTTP errors by using ASP.NET MVC:
[Exception Handling in asp.net MVC](#)

5. Testing

5.1 Testing Information

For the postal code in the solar calculator page, the input should be a four-digit value. In order to ensure the reasonableness of the data input, I checked the relevant literature and found

that the postal code range of Victoria, Australia is 3000 to 3999. Therefore, the valid value of the test is a four-digit integer in the interval, and the illegal value is a thinking integer outside the interval, as well as a negative number.

For page jumps, Epic 6 Carbon Footprint General Information and other functions will undergo usability UX/UX tests. Mainly uses black and white box testing technology. Among them, the black box only checks whether the program functions are used normally in accordance with the requirements specification, and whether the program can properly receive input data and generate correct output information. At the same time, the white box test tests all logical paths.

For website security, there are many free evaluation websites on the Internet. This web application uses Pentest_tools as a tool to test the web security. After using it, a website security assessment report will be generated.

5.2 Documentation Testing Carried Out

Integrity (Acceptance Testing):

All input content, including zip code input, selection of electrical appliances, and carbon emission output indicators, have all gone through the following steps.

1. Develop test plans, test items, test strategies and acceptance criteria, and pass the plan review with customer participation.
2. Establish a test environment, design test cases, and review them.
3. Prepare test data, execute test cases, and record test results.
4. Analyze the test results, analyze the test results according to the acceptance criteria, and make the acceptance and test evaluation. The test project passed; the test project failed, and there is no workaround, which requires great modification; the test project did not pass, but there are workarounds, which will be improved in the later maintenance period or the next version; the test project cannot be evaluated or cannot be given a complete Evaluate. The reason must be given at this time. If it is because the test item is not clearly stated, the test plan should be modified.
5. Submit the test report.

5.2.1 Backup and Recovery Testing:

Due to unpredictable system abnormalities and failures, which may result in the loss of data and source code, backup and data recovery tests are performed to measure the web app's ability to recover files and data sets when the above-mentioned abnormalities occur again.

5.2.2 Usability Testing:

Usability testing to ensure that users can have a friendly and usable system. In each function, we designed usability testing, including map tool, page jump assistant, scroll progress bar, navigation bar to highlight the current page and button high-contrast colors.

Testing details please follow the path blow to find:

- FIT5120 IE Professional TP11
 - 5.0 Product Testing & Feedback Folder
 - Usability Testing Video
 - Iteration 1
 - Iteration 2
 - Iteration 3

5.2.3 Unit Testing:

Unit testing is to write test code to detect specific, clear, and fine-grained functions. Unit testing does not necessarily guarantee that the program functions are correct, nor does it guarantee that the overall business is prepared. Unit testing is not only used to ensure the correctness

of the current code, but more importantly, it is used to ensure the correctness after the code is repaired, improved or refactored.

Testing details please follow the path blow to find:

- FIT5120 IE Professional TP11
 - 2.0 Iterations Folder
 - Iteration 1
 - Testing
 - Iteration 2
 - Testing
 - Iteration 3
 - Testing

5.2.4 Pressure test

Pressure testing is used to determine the reliability and stability of all web resources (e.g. websites, applications and APIs). Pressure testing aims to find the breaking point where a website/application is subjected to extremely high loads over a period of time. This is also known as endurance testing. This removes the root cause of any breakpoints and helps to identify areas where resources can be scaled up so that the website or application does not crash under peak conditions. The below images show the result of pressure testing.

It can be seen that the site realises very well when visited by a small volume of users with a low concurrency profile.



Testing details please follow the path blow to find:

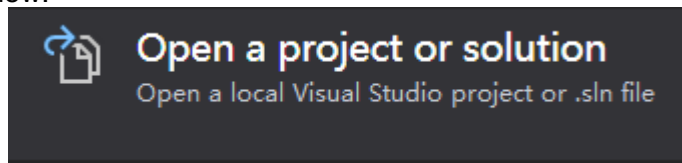
- FIT5120 IE Professional TP11
 - 2.0 Iterations Folder
 - Iteration 1
 - Testing
 - Iteration 2

- Testing
- Iteration 3
 - Testing

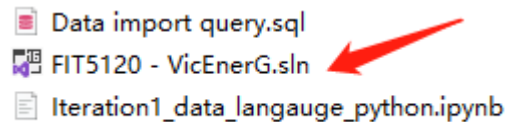
6. Source Code and DB Script

The source code of this project can be found in “../3.0 Code and Database Folder/VicEnerG Project Source Code.zip”. To run this project on your computer, please following the steps blow:

1. Download the zip called “VicEnerG Project Source Code.zip”.
2. Unzip the file.
3. Launch your visual studio any patch after 2017.
4. Open the *.sln* file which means choose the option “open a project or solution” on the right of the window.



5. Choose the *FIT5120- VicEnerG.sln*.



6. To import the initial database, please copy the *App_Data* folder into the *FIT5120 - VicEnerG* folder and choose to change all option.

