THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE, TRINIDAD & TOBAGO, WEST INDIES
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
MASc. in Electrical & Computer Engineering

ECNG 6710

Research Methods

CW2

MATLAB Project

Tariq Mohammed

816010063

Assignment F: Health Diagnostics Center

Course Lecturer: Dr. Sanjay Bahadoorsingh

Date Assigned: February 2nd 2022

Date Submitted: March 28th 2022

THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE, TRINIDAD & TOBAGO, WEST INDIES
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
B. Sc. in Electrical & Computer Engineering

CHEATING, PLAGIARISM AND COLLUSION DECLARATION FORM

According to Rules 3.31 and 3.32 of The UWI Faculty of Engineering Undergraduate Regulations and Syllabuses 2018/2019:

3.31 "**Cheating, Plagiarism and Collusion are serious offences under University Regulations**.

(a) Cheating is any attempt to benefit one's self or another by deceit or fraud.

(b) Plagiarism is the unauthorised and/or unacknowledged use of another person's intellectual efforts and creations howsoever recorded, including whether formally published or in manuscript or in typescript or other printed or electronically presented form and includes taking passages, ideas or structures from another work or author without proper and unequivocal attribution of such source(s), using the conventions for attributions or citing used in this University. Plagiarism is a form of cheating.

(c) For the purposes of these Regulations, 'collusion' shall mean the unauthorised or unlawful collaboration or agreement between two or more students in the preparation, writing or production of a course assignment for examination and assessment, to the extent that they have produced the same or substantially the same paper, project report, as the case may be, as if it were their separate and individual efforts, in circumstances where they knew or had reason to know that the assignment or a part thereof was not intended to be a group project, but was rather to be the product of each student's individual efforts.

3.32 Cheating, plagiarism and collusion shall be reported to the Campus Committee on Examinations and the penalties would be in accordance with the University Examination Regulations."

I, ……Tariq Mohammed……, have read and understood Rules 3.31 and 3.32 of The UWI Faculty of Engineering Undergraduate Regulations and Syllabuses 2018/2019 on Cheating, Plagiarism and Collusion.

I understand that my submission is subject to the electronic plagiarism checker, Turnitin.

I declare that this assignment is my own work and does not involve cheating, plagiarism or collusion.

Signature: …… Date: ……23/03/22……

Abstract

The Neval Memorial Medical Laboratory wishes to monitor the effectiveness of their existing shift system using logs recorded hourly. The diagnostic centre employs several technicians and testing equipment, performing a variety of medical tests to patients each day. This project sets out to develop a software solution to assist the centre's owner in analyzing the shift system's effectiveness, so that he can prepare duty schedules for the lab technicians. MATLAB's, through its App Designer feature, is used to develop a UI which solves this problem. A two-windowed application is created through this project, allowing users to import and validate their data, and then perform a variety of 2D and 3D customizable plots. Apart from the software's ability to complete all plots, the system met all other functional requirements.

Table of Contents

1 Background

1.1 Problem Description (adapted from myelearning):

Dr. Peterson, is the owner of "*The Neval Memorial Medical Laboratory", a* health diagnostic testing center which employs several technicians, testing equipment and handles a number of medical tests on patients every day. The technicians operate on a 24 hour shift system for which Dr. Peterson wishes to monitor the effectiveness of. He logs information on an hourly basis.

This project sets out to develop a software solution to assist Dr. Peterson in analyzing the effectiveness of his shift system so that he can prepare duty schedules for the lab technicians.

1.2 Approach to the Problem

One can use MATLAB, a popular programming and numeric computing platform among engineers to develop a software solution, to solve *The Neval Memorial Medical Laboratory's* problem. MATLAB's App Designer will therefore be chosen for implementing this project.

1.3 Scope of the Solution

The scope of this solution is captured through the following requirements:

Data Upload App:
REQ1-1: The data upload app will display the application's name, company's logo and the words "*Data Upload*".
REQ1-2: The data upload app will allow users to import Excel files from their local storage.
REQ1-3: The data upload app will allow user to open the plot generator app importing data.
REQ1-4: The data upload app will display error messages if non-numeric data is imported into the application, or if the path to import data is invalid.
REQ1-5: The data upload app will prompt user confirmation on attempt to close the app.

Plot Generator App:
REQ2-1: The plot generator app will display the application's name, company's logo and the words "*Plot Generator*".
REQ2-2: The plot generator app will automatically generate a 3D plot on opening.
REQ2-3: The plot generator app will allow users to toggle between 2D and 3D plots.
REQ2-4: The plot generator app will allow users to select the data which they wish to plot.
REQ2-5: The plot generator app will allow users to toggle the grid on the generated plots.
REQ2-6: The plot generator app will allow users to select a colour scheme from a list of options.
REQ2-7: The plot generator app will display error messages if the same data is chosen for 2 axes.
REQ2-8: The plot generator app will prompt user confirmation on attempt to close the app.

2 Design

In following with the requirement presented in Section 1.3 of this report, as well as the project specifications, two designs were created:

1. *Data Upload Window* design
2. *Plot Generator Window* design

2.1 GUI Layout and Description – Data Upload Window

The *Data Upload Window* allows users to select a local Excel file containing the headers from the project. The file's path is displayed once selected. This path can only be changed by selecting another file. After choosing a file, data can be imported into the table in Figure 1 by clicking on the "Load Data" button. If erroneous data is detected, an error popup will be displayed to the user. This is shown in Section 3. If the imported data is all numeric or datetime, the user can click on the "Plot Data" button to launch the plots, shown in the Plot Generator App. Should the user still have erroneous (non-numeric or datetime) data and attempts to plot, an error message will be displayed. A dark theme was chosen to draw the users attention.

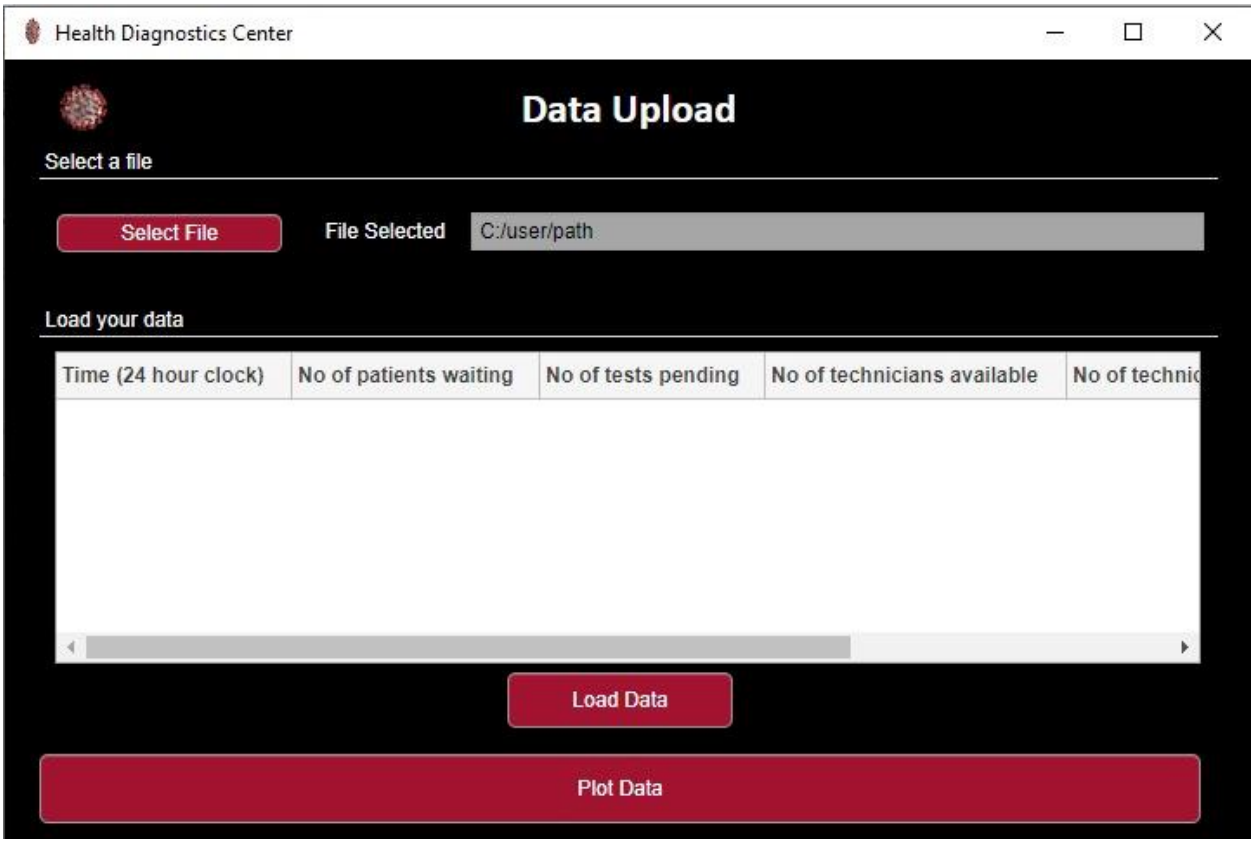

Figure 1 Data Upload Window Layout

2.2 GUI Layout and Description – Plot Generator Window

The *Plot Generator Window* will automatically open and show a 3D plot after the "Plot Data" button is pressed on the *Data Upload Window*. The default configuration and plot is shown in

Figure 2. Here, users can switch between 2D and 3D plots, change the axes, according to the data in the imported file, toggle the grid and change the colour schemes. An error message is displayed to users if they try to plot the same data on more than one axis, for both 2D and 3D plots.
The light-theme used here was intentional, as the design team wished to explicitly differentiate between both the *Data Upload Window* as well as the *Plot Generator Window*.

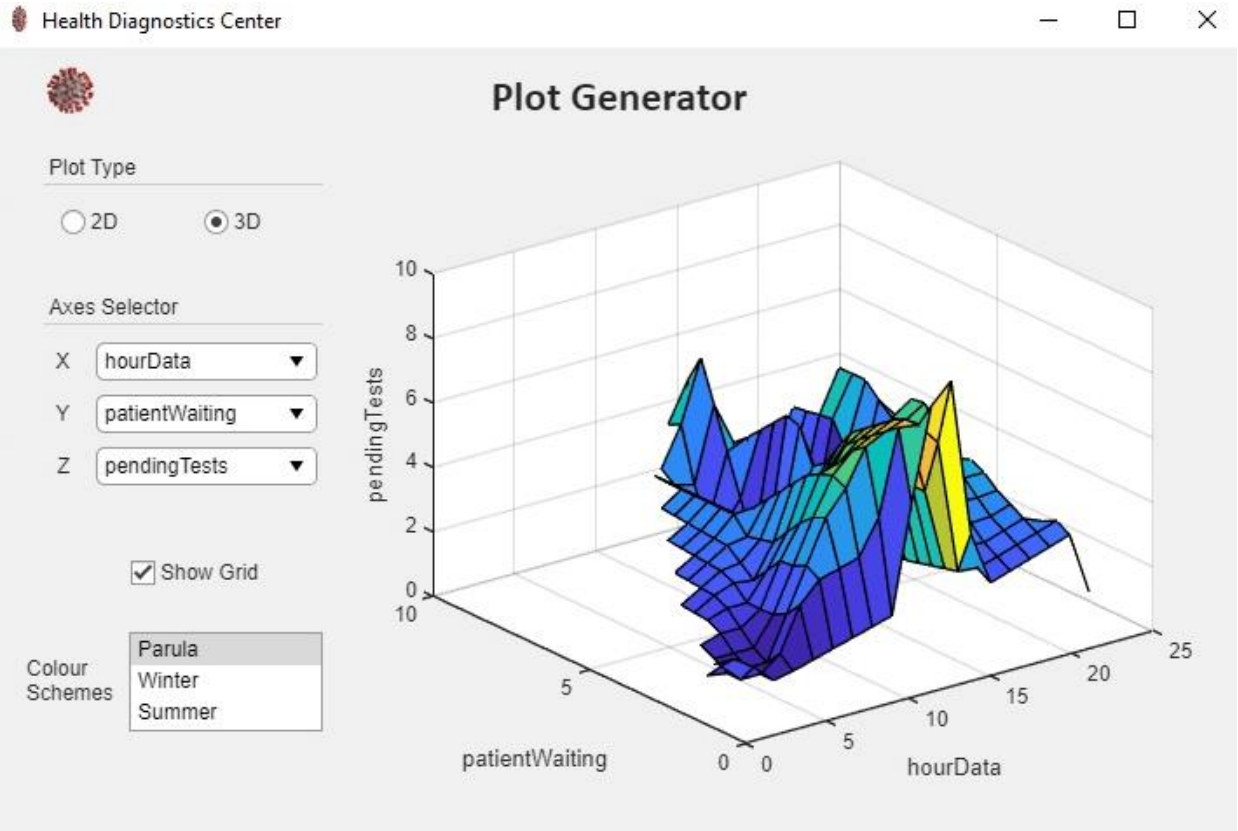

Figure 2 Plot Generator Window Layout

2.3 Dataset Identification

The dataset chosen represents one day's worth of reports, from 12:00AM (00:00) to 11:00PM (23:00). Assumption: *The Neval Memorial Medical Laboratory* is a 24/7 medical laboratory. Two datasets, comprising 24 rows each (one for each hour of the day), are used in this project:

- **Nominal data**: using numeric and datetime data (see Appendix A), which is the nominal data types to be collected
- **Erroneous data**: using non-numeric and non-datetime data (see Appendix B) to test the resilience of the designed system

3 Results

This section presents a summary of the results from testing performed. While all of the system tests met the requirements, one outlier existed in the 3D plotting of patientData. Due to the time constraints on delivery, project engineers have opted to replace the plotter feature for that data grouping with an error message (see Figure 13), with hopes of patching the identified bugs in time for future releases.

3.1 Loading of Data into GUI

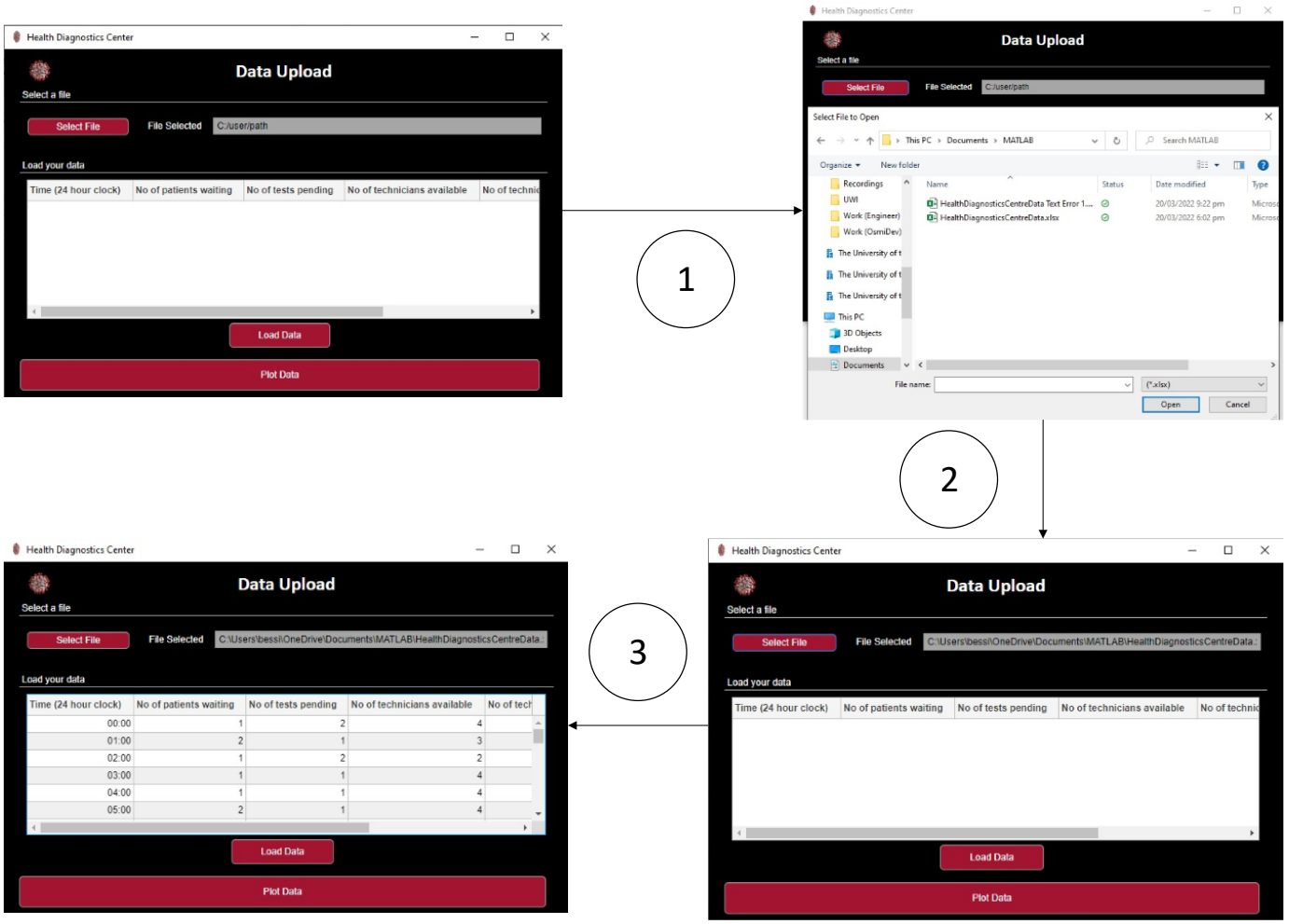

Figure 3 Data Upload Results

3.2 2D Plot – Axis Variable Change Feature

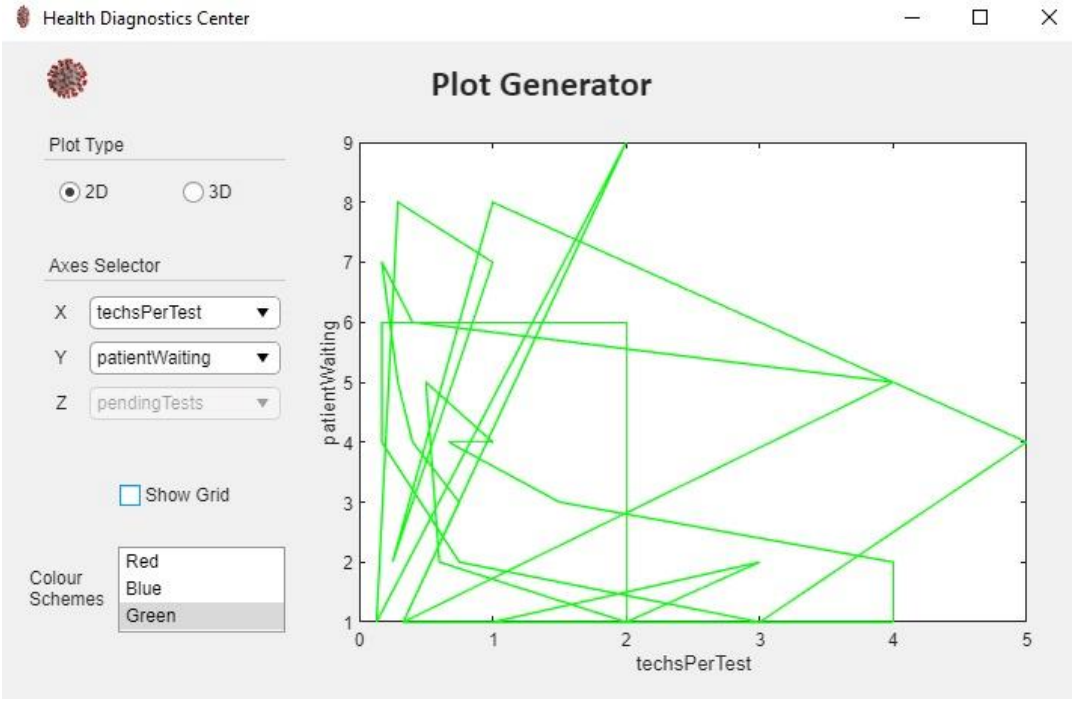

Figure 4 Variable Axis and Colour Change

3.3 2D Plot – Colour Change Feature

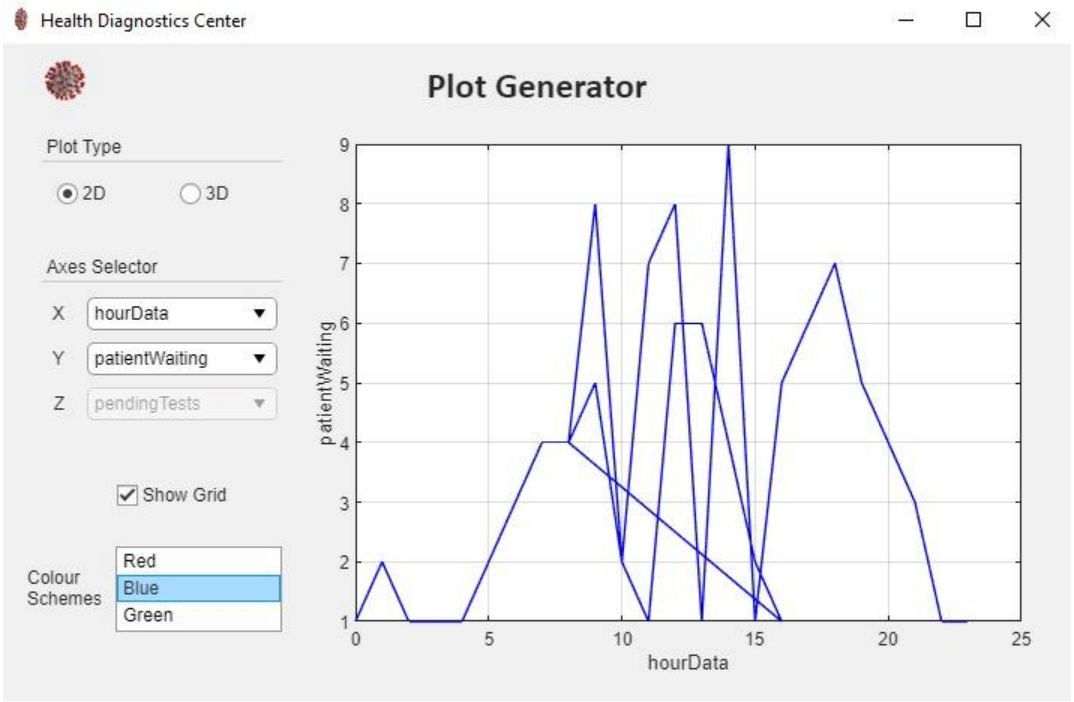

Figure 5 2D Colour Change (Blue Selected)

3.4 3D Plot – Axis Variable Change Feature

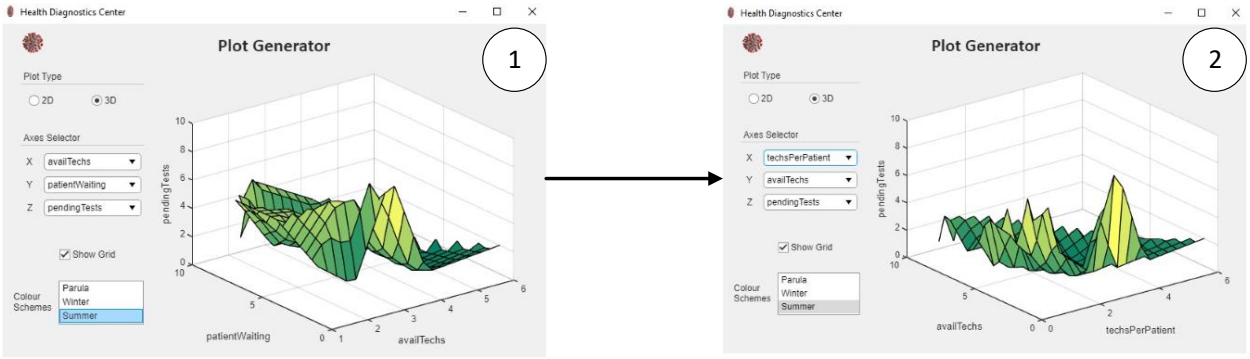

Figure 6 3D Plot Axes Change

3.5 3D Plot – Colour Change Feature

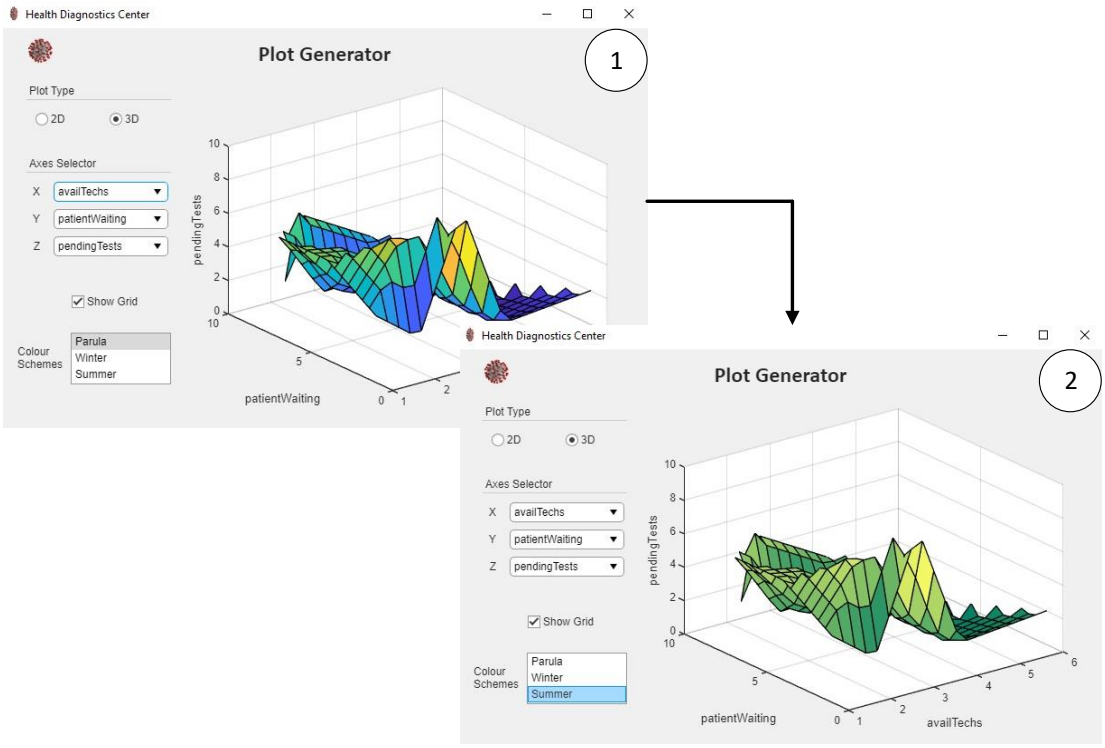

Figure 7 Colour Change from Parula to Summer

3.6 Error Handling and Exit Dialog Boxes

Data Upload App: Invalid file path

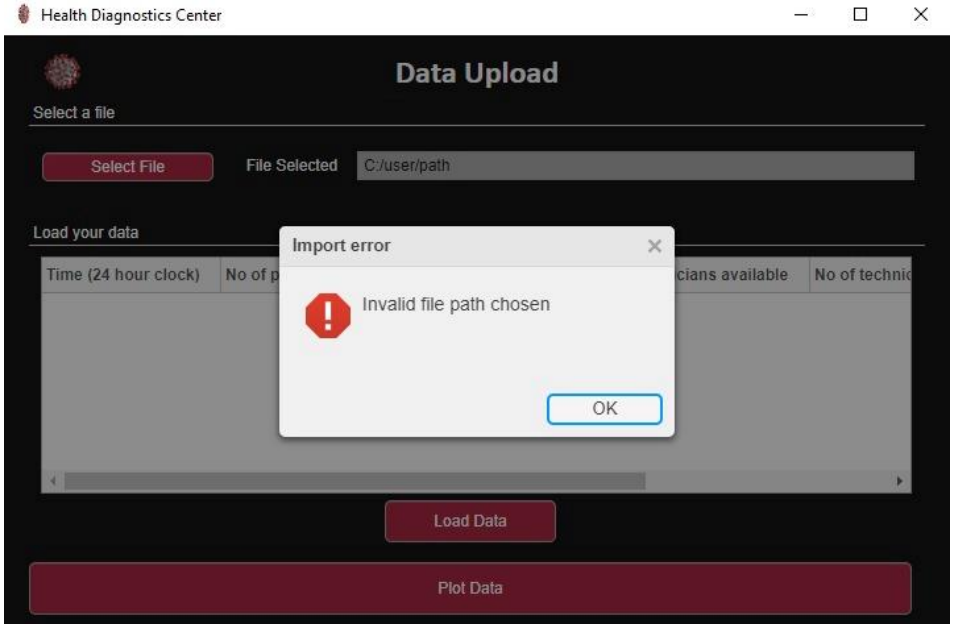

Figure 8 Error: Invalid file Path

Data Upload App: Non-numeric data

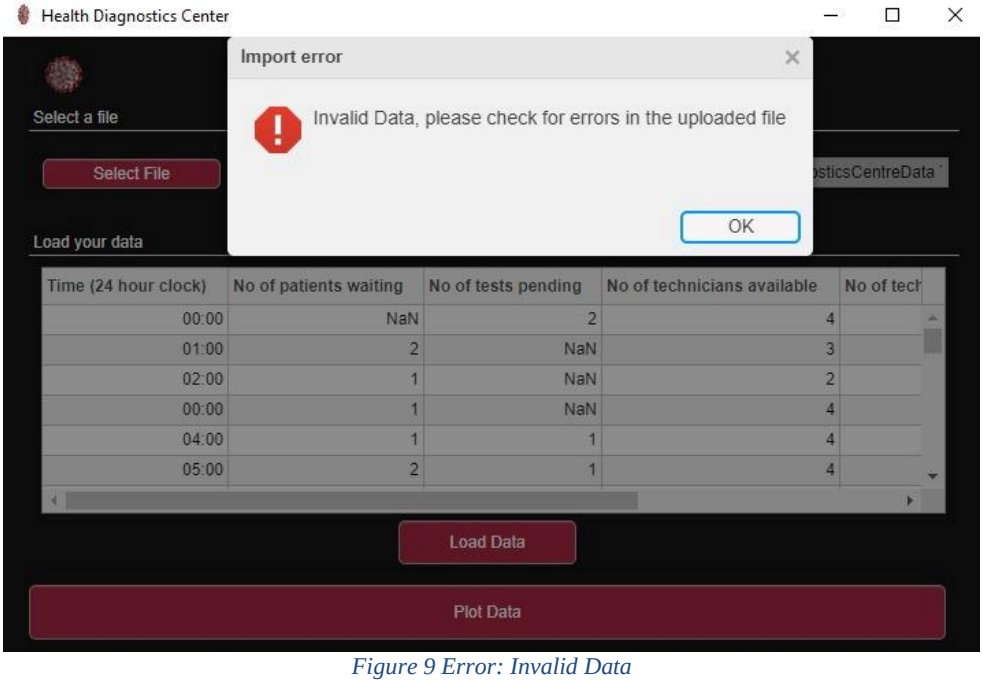

Figure 9 Error: Invalid Data

Data Upload App: Exit dialog

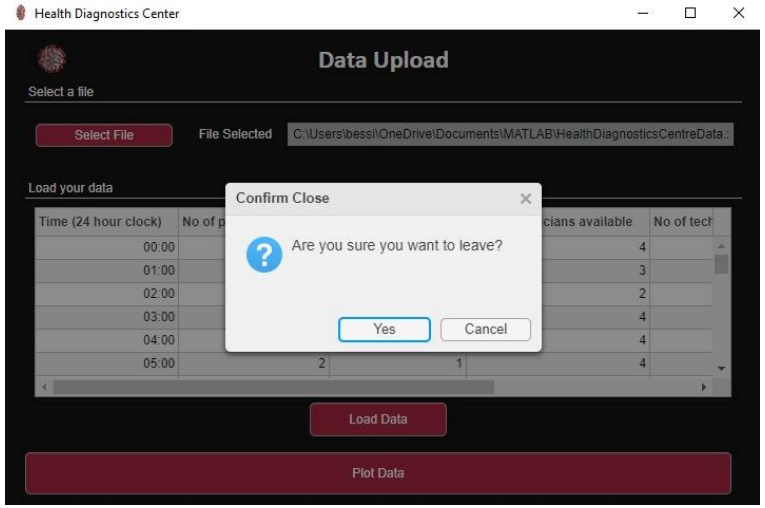

Figure 10 Exit Dialog Box for Data Upload App

Plot Generator App: Exit Dialog

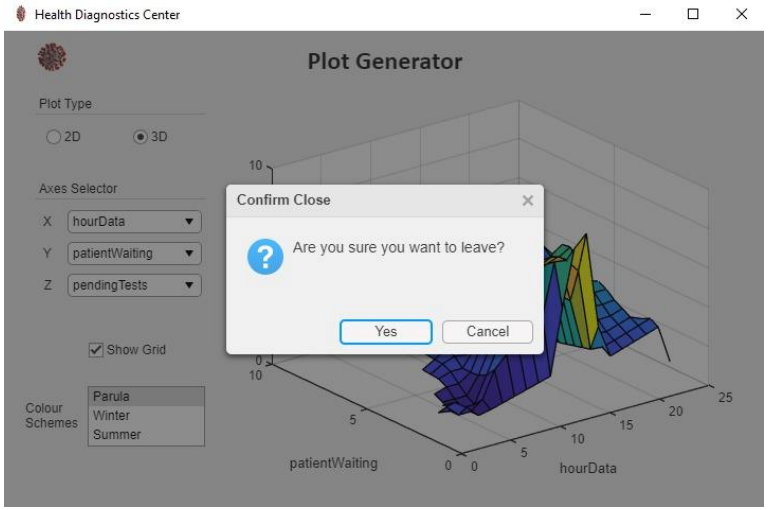

Figure 11 Exit Dialog Box for Plot Generator App

Plot Generator App: Redundant Axes

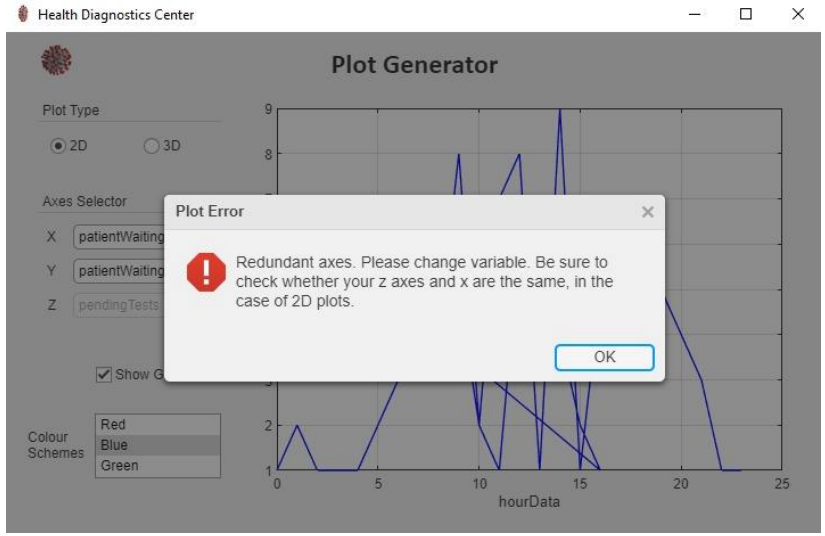

Figure 12 Error Message for Redundant Axes

Additional Error: When plotting PatientData (Unresolved)

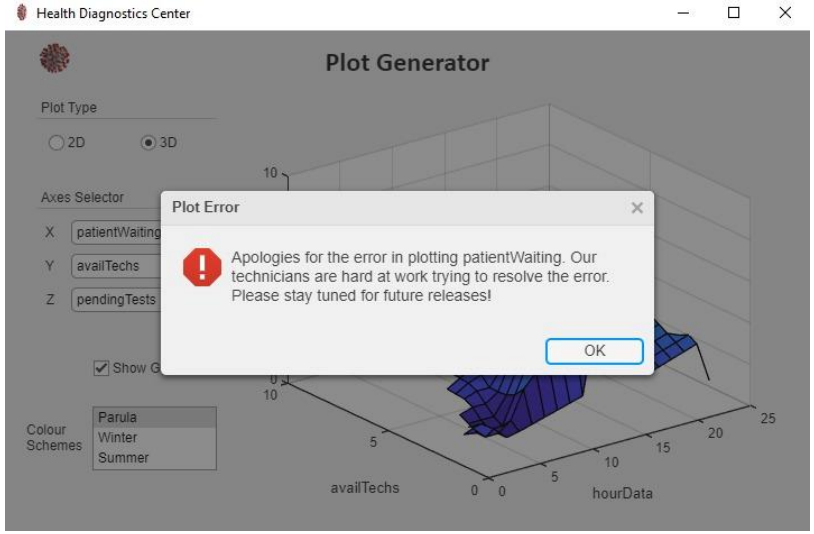

Figure 13 Error Message for Inability to Plot patientData in 3D

4 Summary of Achievements

Overall, the project's deployment was considered to be a success, as shown in Table 1 Summary of Project Achievements, with only one of the requirements being partially completed.

Table 1 Summary of Project Achievements

App	Requirement	Achievement Status
Data Upload App	**REQ1-1**: The data upload app will display the application's name, company's logo and the words "*Data Upload*".	Complete
	REQ1-2: The data upload app will allow users to import Excel files from their local storage.	Complete
	REQ1-3: The data upload app will allow user to open the plot generator app importing data.	Complete
	REQ1-4: The data upload app will display error messages if non-numeric data is imported into the application, or if the path to import data is invalid.	Complete
	REQ1-5: The data upload app will prompt user confirmation on attempt to close the app.	Complete
Plot Generator App	**REQ2-1**: The plot generator app will display the application's name, company's logo and the words "*Plot Generator*".	Complete
	REQ2-2: The plot generator app will automatically generate a 3D plot on opening.	Complete
	REQ2-3: The plot generator app will allow users to toggle between 2D and 3D plots.	Partially complete[1]
	REQ2-4: The plot generator app will allow users to select the data which they wish to plot.	Complete
	REQ2-5: The plot generator app will allow users to toggle the grid on the generated plots.	Complete
	REQ2-6: The plot generator app will allow users to select a colour scheme from a list of options.	Complete
	REQ2-7: The plot generator app will display error messages if the same data is chosen for 2 axes.	Complete
	REQ2-8: The plot generator app will prompt user confirmation on attempt to close the app.	Complete

[1] Error when placing "patientWaiting" data on x or z axes; all other data works fine on any axes.

Appendix A: Valid Dataset

Time (24 hour clock)	No of patients waiting	No of tests pending	No of technicians available	No of technicians per patient	No of technicians per test
0:00	1	2	4	4	2
1:00	2	1	3	1.5	3
2:00	1	2	2	2	1
3:00	1	1	4	4	4
4:00	1	1	4	4	4
5:00	2	1	4	2	4
6:00	3	2	3	1	1.5
7:00	4	3	2	0.5	0.67
8:00	4	4	4	1	1
9:00	5	4	2	0.4	0.5
10:00	2	5	3	1.5	0.6
11:00	1	2	4	4	2
12:00	6	1	2	0.33	2
13:00	6	6	1	0.17	0.17
14:00	4	6	1	0.25	0.17
15:00	2	4	3	1.5	0.75
16:00	1	2	6	6	3
8:00	4	1	5	1.25	5
9:00	8	4	4	0.5	1
10:00	2	8	2	1	0.25
11:00	7	2	2	0.29	1
12:00	8	7	2	0.25	0.29
13:00	1	8	1	1	0.13
14:00	9	1	2	0.22	2
15:00	1	9	3	3	0.33
16:00	5	1	4	0.8	4
17:00	6	5	2	0.33	0.4
18:00	7	6	1	0.14	0.17
19:00	5	7	2	0.4	0.29
20:00	4	5	2	0.5	0.4
21:00	3	4	3	1	0.75
22:00	1	3	1	1	0.33
23:00	1	1	1	1	1

Appendix B: Erroneous Dataset (Errors in Red)

Time (24 hour clock)	No of patients waiting	No of tests pending	No of technicians available	No of technicians per patient	No of technicians per test
0:00	a	2	4	#VALUE!	2
1:00	2	a	3	1.5	#VALUE!
2:00	1	-	2	2	s
0:00	1	@	4	4	#VALUE!
4:00	1	1	4	4	4
5:00	2	1	4	2	4
6:00	3	2	3	1	1.5
7:00	4	3	2	0.5	0.67
8:00	4	4	4	1	1
9:00	5	4	2	0.4	0.5
10:00	2	5	3	1.5	0.6
11:00	1	2	4	4	2
12:00	6	1	2	0.33	2
13:00	6	6	1	0.17	0.17
14:00	4	6	1	0.25	0.17
15:00	2	4	3	1.5	0.75
16:00	1	2	6	6	3
8:00	4	1	5	1.25	5
9:00	8	4	4	0.5	1
10:00	2	8	2	1	0.25
11:00	7	2	2	0.29	1
12:00	8	7	2	0.25	0.29
13:00	1	8	1	1	0.13
14:00	9	1	2	0.22	2
15:00	1	9	3	3	0.33
16:00	5	1	4	0.8	4
17:00	6	5	2	0.33	0.4
18:00	7	6	1	0.14	0.17
19:00	5	7	2	0.4	0.29
20:00	4	5	2	0.5	0.4
21:00	3	4	3	1	0.75
22:00	1	3	1	1	0.33
23:00	1	1	1	1	1