



SCHOOL OF COMPUTING
UNIVERSITI UTARA MALAYSIA

STIA1123 PROGRAMMING 2

PROJECT (30%)

SECOND SEMESTER SESSION 2019/2020 (A192)

| | NAME | MATRIC NUMBER |
|----|----------------------------------|----------------------|
| 1. | MUHAMMAD DANIAL FITRI B. GHAZALI | 272868 |
| 2. | MUHAMMAD AFIRUDIN BIN JAMILAN | 273829 |
| 3. | LING CHING SEN | 271738 |
| 4. | TIANG CHANG YAO GALVIN | 271872 |
| 5. | GOKUL NATH VINA A/L GANESAN | 272930 |

Date of Submission : 16 JULY 2020

TABLE OF CONTENTS

| INDEX | TOPIC | PAGE NUMBER |
|-------|---------------------------|-------------|
| 0 | BACKGROUND OF THE PROJECT | 3 |
| 1 | MAIN LAUNCHER | 4 – 5 |
| 2 | AIRLINE | 6 – 10 |
| 3 | AIRPORT | 11 - 19 |
| 4 | AIRPLANE | 20 – 32 |
| 5 | RESTAURANT | 33 - 39 |
| 6 | APPENDIX | 40 |

0.0 - BACKGROUND OF THE PROJECT

The project aimed to aid the tourism industry during the PKPP phase by compiling the Standard Operating Procedures (SOP) for various businesses related to the industry in a single accessible program. Consumers will be able to check SOPs for various businesses using the program while the businesses will be able to update the SOPs appropriately.

There are several sections included in the program, which are airlines, airports, airplanes, restaurants, and hotels. Users will be able to check or update the SOP based on the business on the appropriate section. All of the data was also stored on the server, allowing every copies of the program to access the same data source.

To aid the development, a different member of our team handled each of the different sections. We also used a special frontend and backend packages to abstract some elements of the program such as GUIs and server-client connection, allowing the programmer to focus on their section.

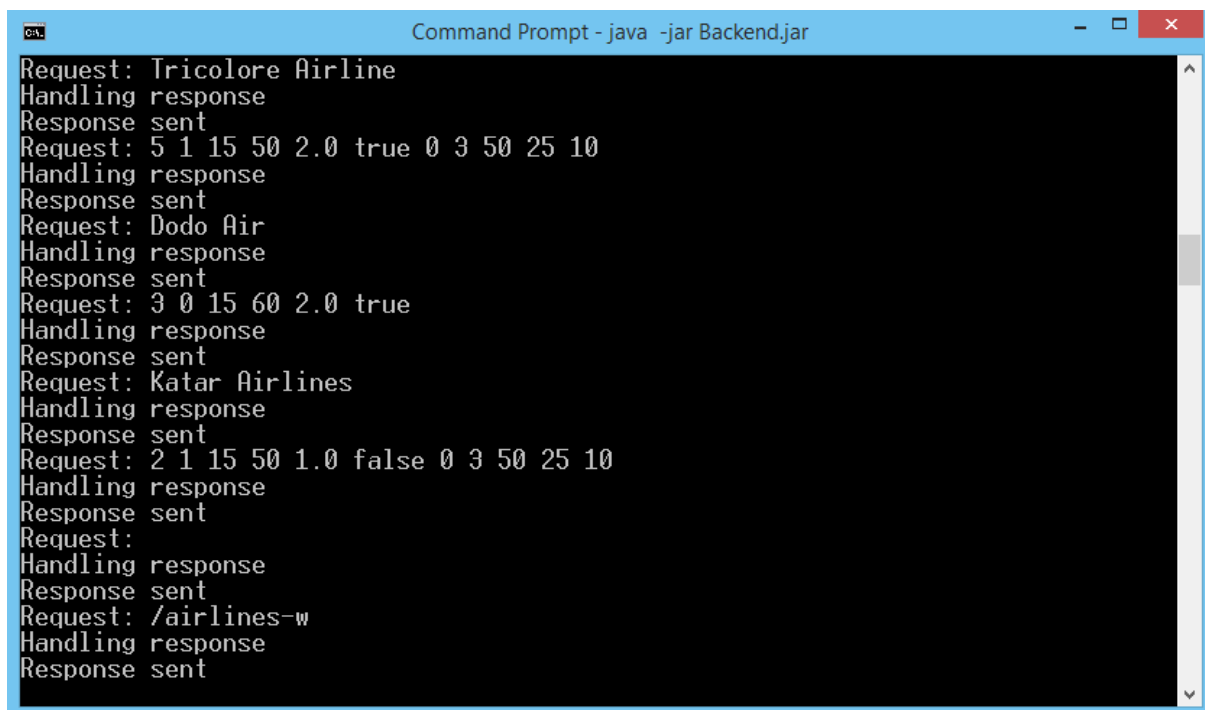
1.0 – MAIN LAUNCHER

1.1 – PROGRAM DESCRIPTION

The main launcher is the main entry point of the program. When the user launches the program, the main launcher will initialize all of the contents and create a connection to the server. The main launcher depends on several packages, Frontend.jar and Backend.jar.

1.1.1 - BACKEND

The Backend defines the server-client logic of the program. The launcher uses the Client class to make use of the methods that allows it to communicate with the server. Therefore, the program must be run when the machine is connected to the internet.



```
Command Prompt - java -jar Backend.jar
Request: Tricolore Airline
Handling response
Response sent
Request: 5 1 15 50 2.0 true 0 3 50 25 10
Handling response
Response sent
Request: Dodo Air
Handling response
Response sent
Request: 3 0 15 60 2.0 true
Handling response
Response sent
Request: Katar Airlines
Handling response
Response sent
Request: 2 1 15 50 1.0 false 0 3 50 25 10
Handling response
Response sent
Request:
Handling response
Response sent
Request: /airlines-w
Handling response
Response sent
```

Figure 1 - The Server interface inside the Backend package

1.1.2 – FRONTEND

The Frontend defines all of the launcher GUI and provide an abstract Content class to be used by all topic. The navigation buttons on the side will call the appropriate Content to be displayed. The user then can interact with all Content freely. If they switch to a different Content, the original Content won't be reinitialized, allowing any user to continue using a Content in the state that they left it in during the session.

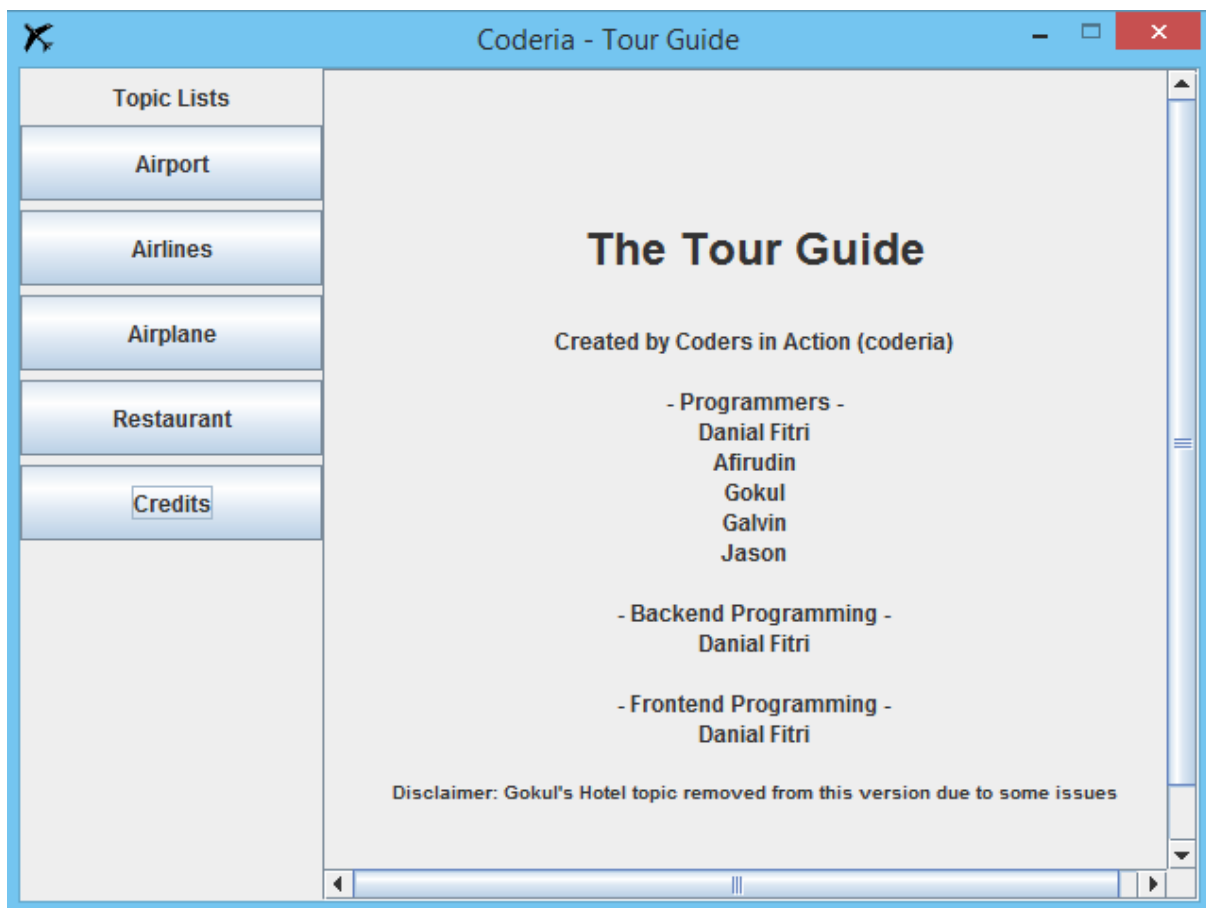


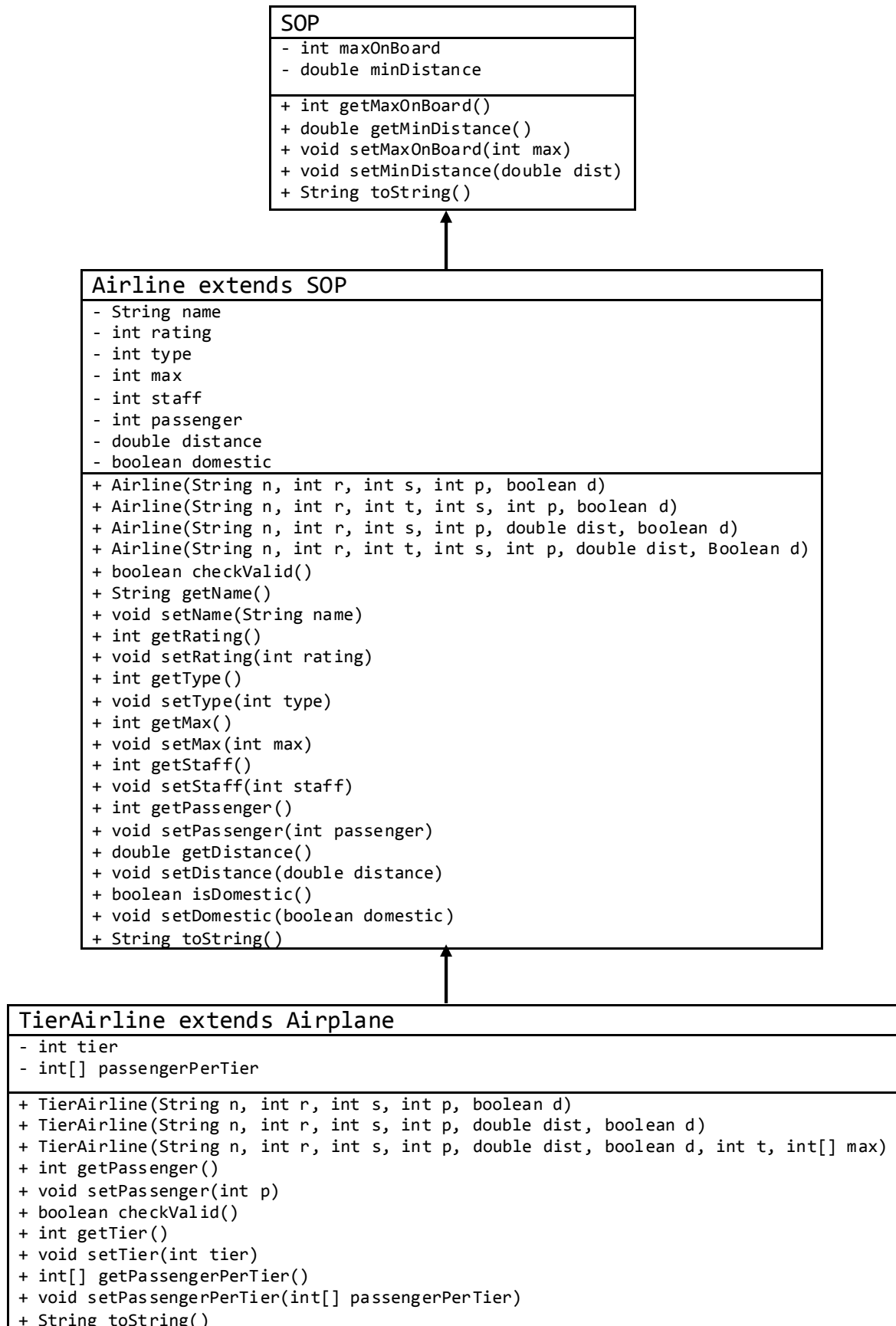
Figure 2 - The final interface of the program

1.2 – THE CODE

The code of the MainLauncher, Backend, and Frontend can be found in the source folder in the Github repository.

2.0 – AIRLINE

2.1 – UML DIAGRAM



2.2 – AIRLINE PROGRAM DESCRIPTION

The airline part of the program can be accessed by clicking the appropriate button on the sidebar. The content area will be updated to show the AirlineContent UI. In the middle of the UI is a table that will list the airlines with their respective SOPs. It will also show whether the airline's SOP adheres to the SOP set by the government. The user can click the SOP button to see the SOP set by the government. Each airlines must follow this SOP. Then, a new record can be added by clicking the ADD button. This will bring up a separate window where the user can enter the details of the airline. Then, the record can be modified or deleted by clicking the EDIT or DEL button respectively, which brings up a separate window where the user can select a specific airline. Finally, the SEARCH button allows the user to search the details of a particular airline by its name. Every actions that modifies data such as ADD, EDIT, and DEL will cause a similar change to occur on the server.

As for the logic of the program, the SOP class defines the standard set by the government, such as the maximum number of person on board and minimum distance per person. Airline class is a direct subclass of it, and defines a single normal airline. TierAirline on the other hand is a subclass of the Airline class, and defines an airline where its passengers are divided into tiers. Both Airline and TierAirline class contains a method checkValid() that allows us to check if the SOP defined adheres to the standard.

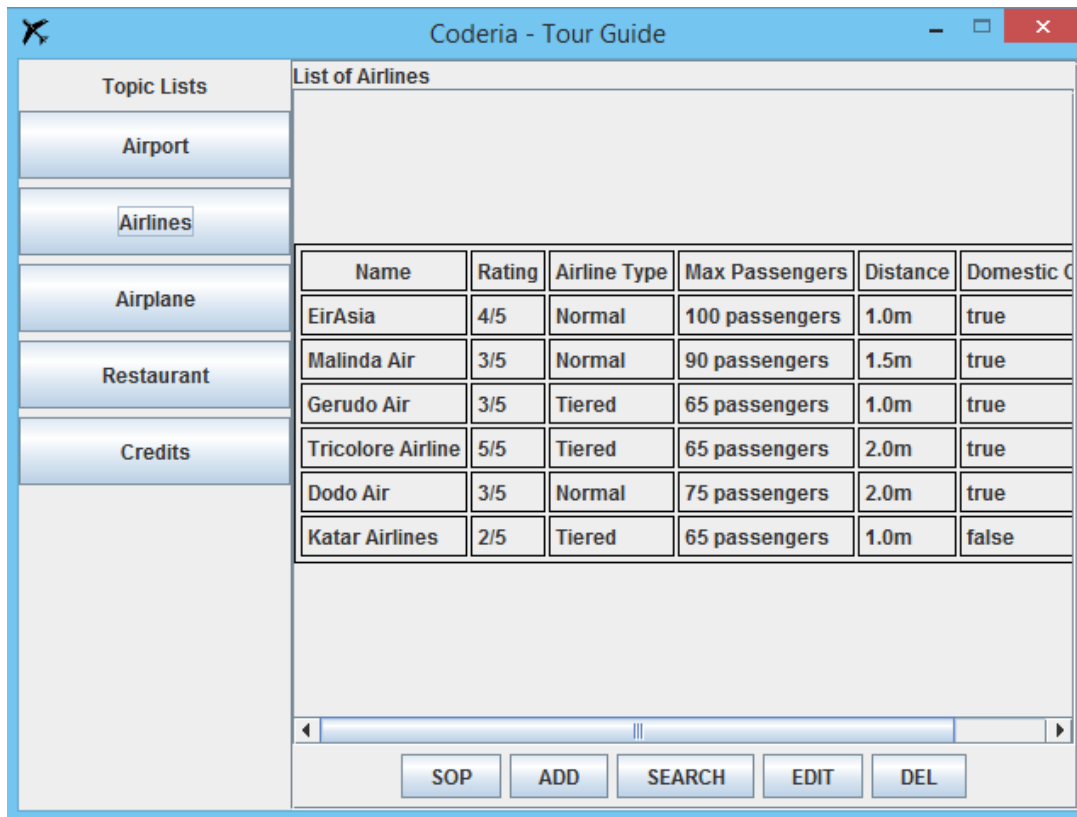


Figure 3 - The main interface of Airline

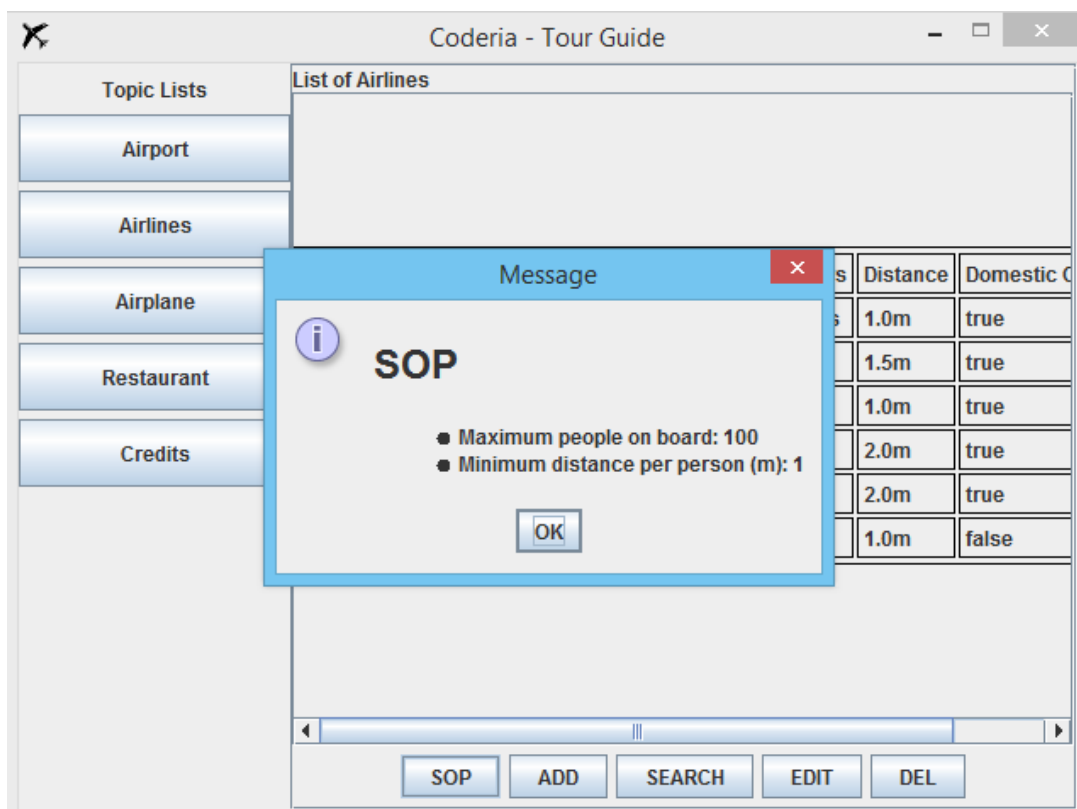


Figure 4 - The details of SOP set by the government

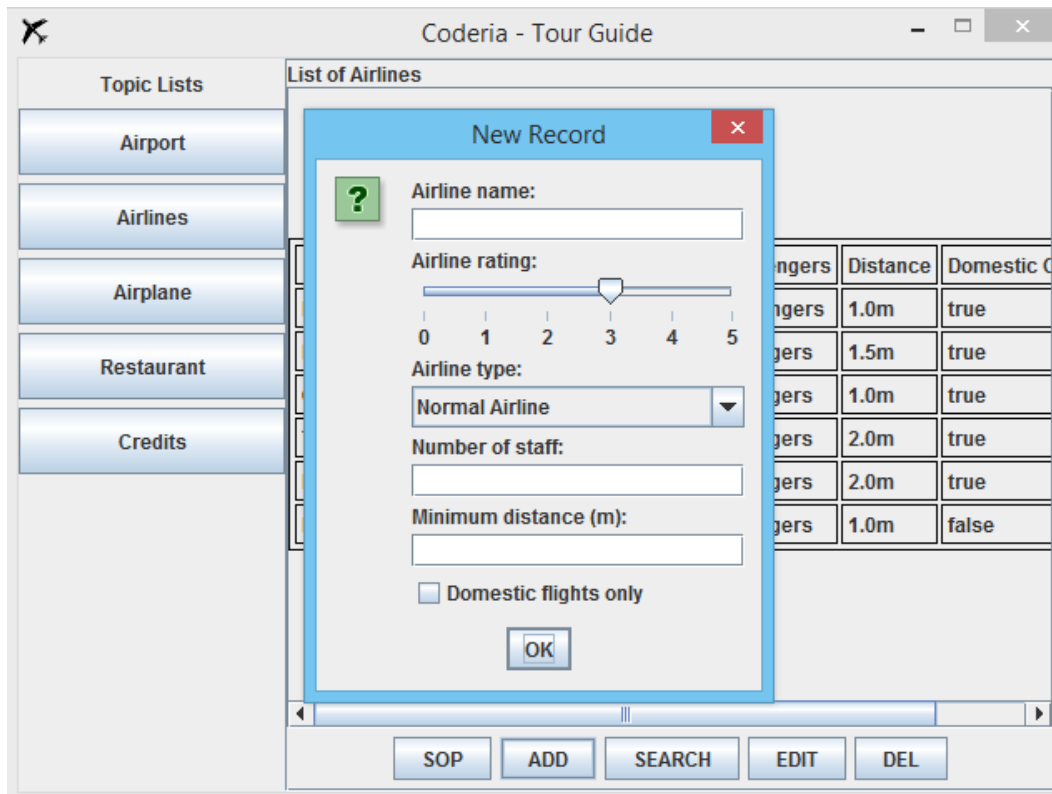


Figure 5 - The interface to add new airlines. Incorrect input will be rejected

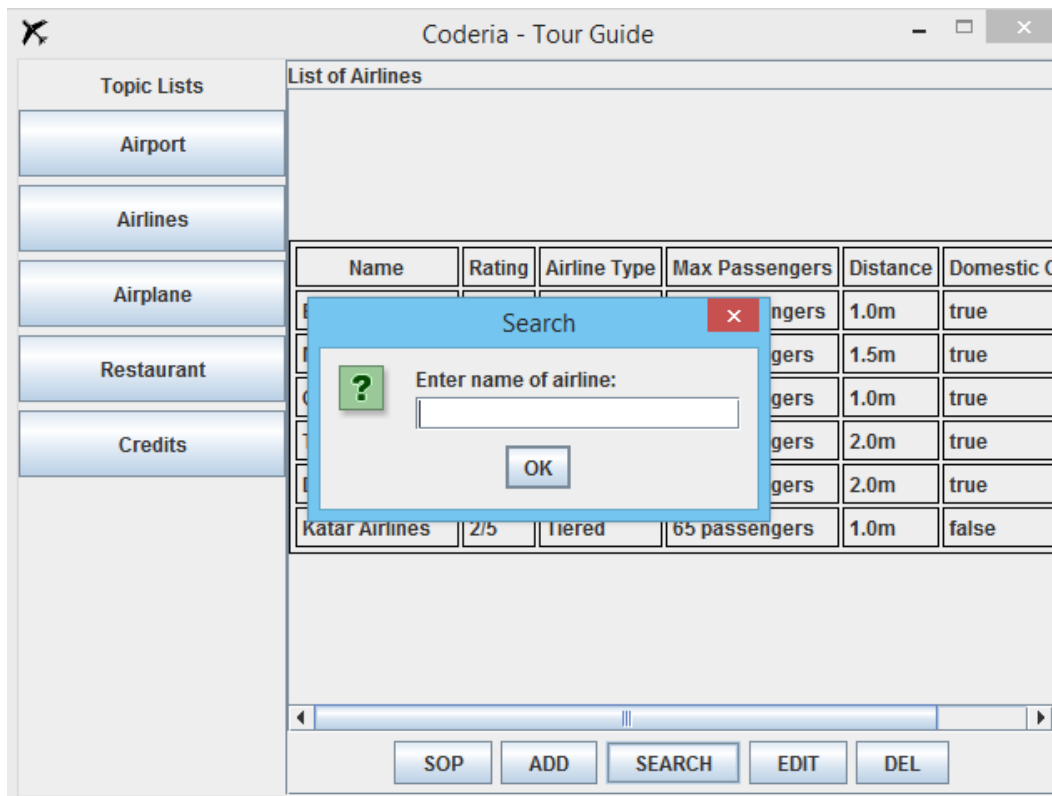


Figure 6 - Searching for a record. Linear search algorithm was used

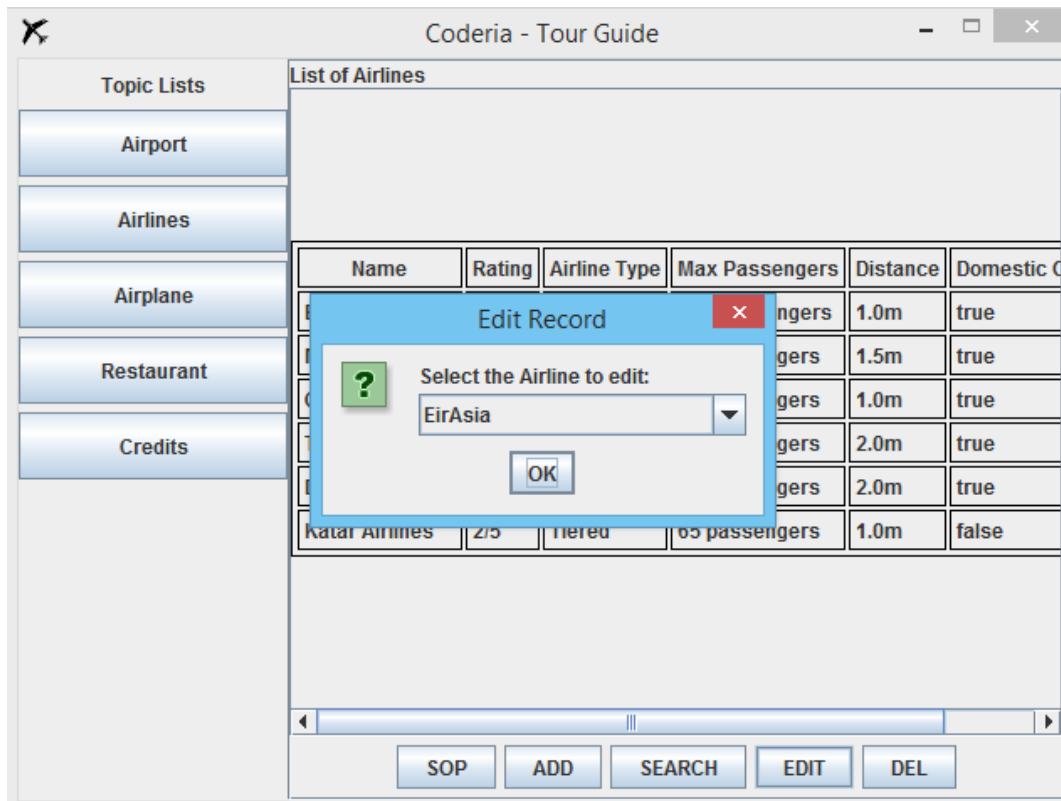


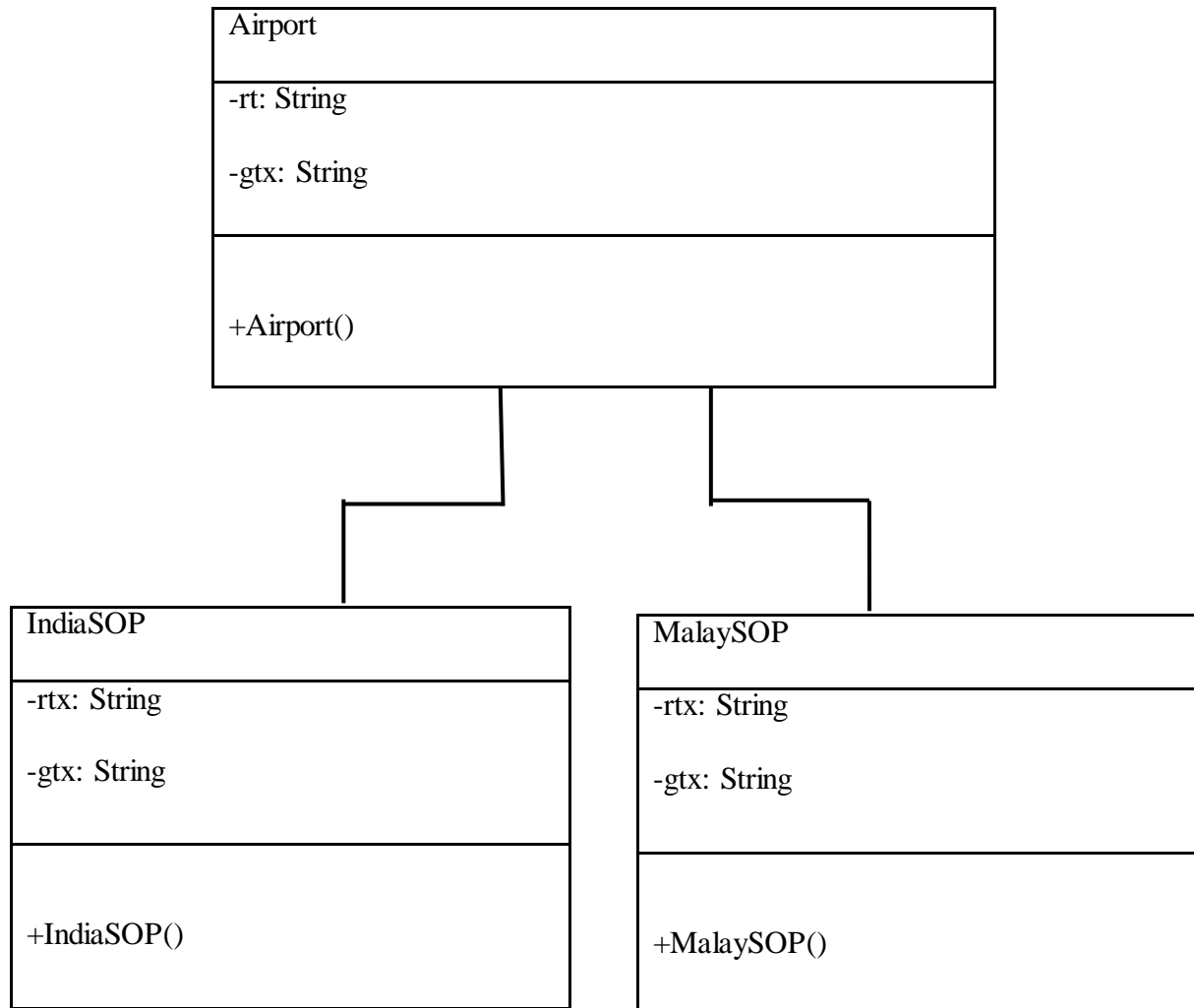
Figure 7 - EDIT and DEL requires the user to select the airlines from a dropdown

2.3 – THE CODE

The code for Airlines can be found in the Source folder in the Github repository.

3.0 – AIRPORT

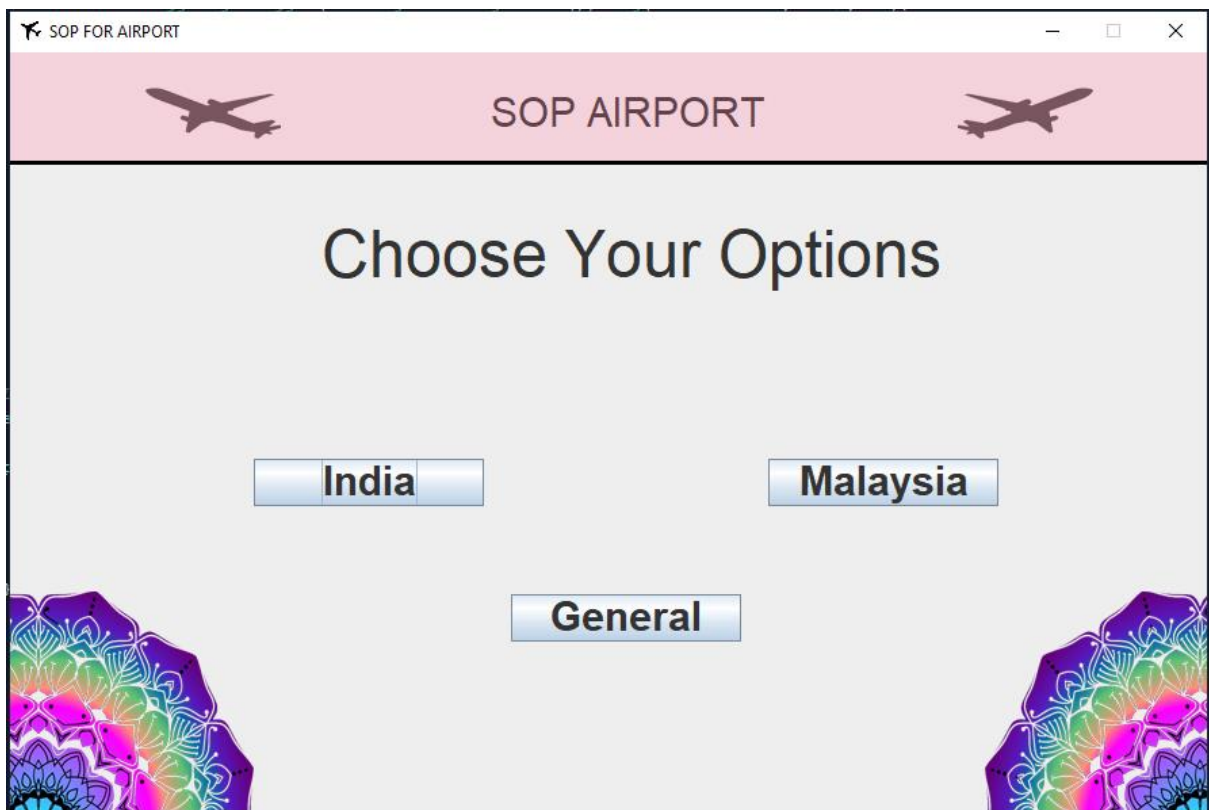
3.1 – UML DIAGRAM



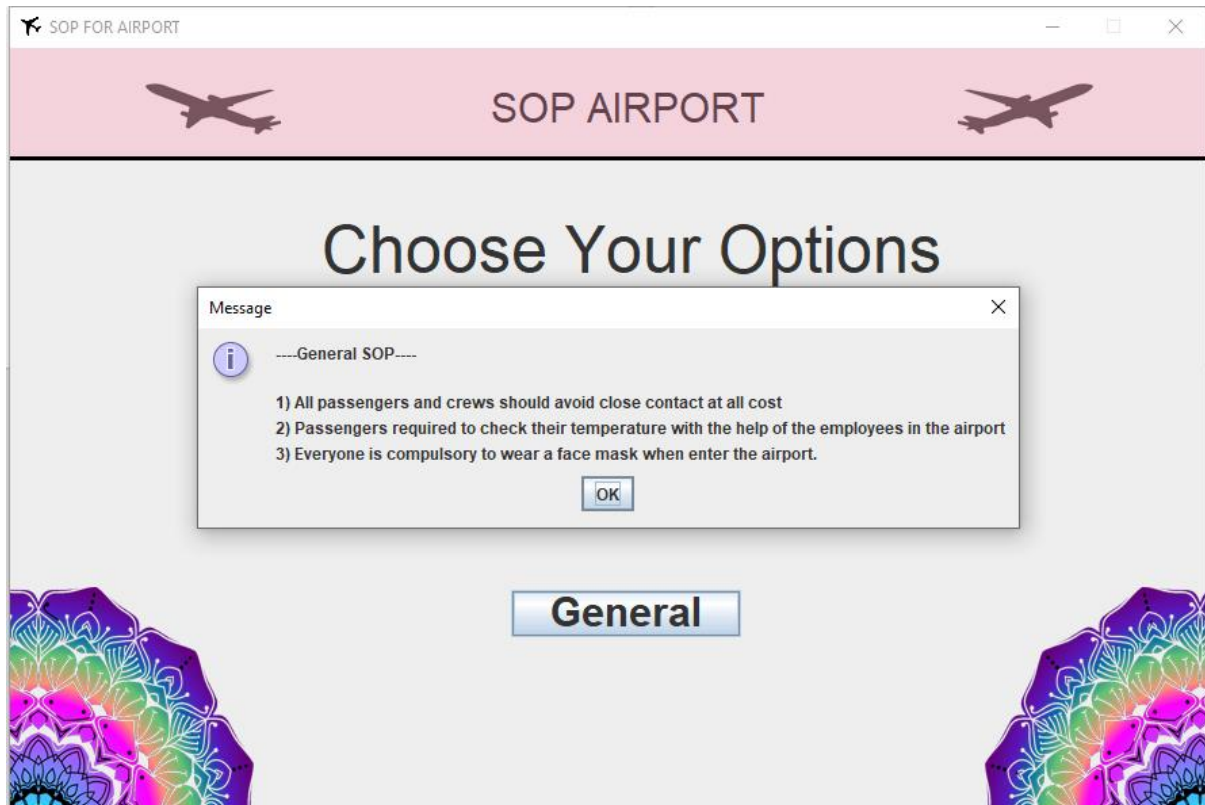
3.2 – PROGRAM DESCRIPTION

Airport Class

This class have the main method. When we run the program, it will show the main page of the GUI. On top of it called SOP Airport. The main page has only 3 buttons to choose from, which are India, Malaysia and General.



When you clicked the General button, it will pop out a JOptionPane that contains the data for general SOP for airport around the world. The data had been taken from a text file called GeneralSOP.txt. I used the BufferedReader function to make sure the JOptionPane will read the data from the text file.



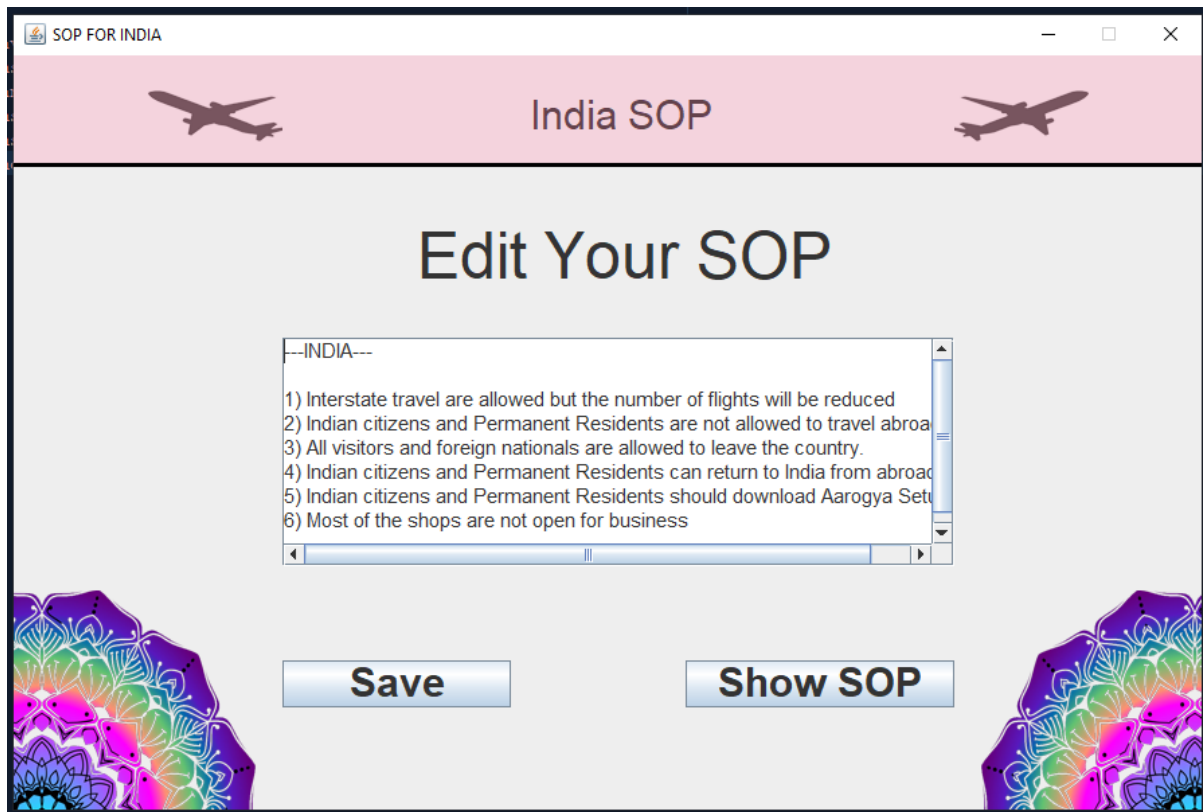
```
1  ----General SOP----
2
3  1) All passengers and crews should avoid close contact at all cost
4  2) Passengers required to check their temperature with the help of the employees in the airport
5  3) Everyone is compulsory to wear a face mask when enter the airport.
```

IndiaSOP Class:

In the main page, when you clicked the India button, it will pop out new GUI. In this section, it will have three components, one JTextArea and 2 Jbutton, which are Save and Show SOP. This is where you can edit your SOP.



JTextArea will show data about the SOP for the country India. To make the data visible, you have to click the Show SOP button for the text area to show the data. The data had been taken from a text file that is called SopIndia.txt. The text area had been implemented with a JScrollPane option.



```
1  ---INDIA---
2
3  1) Interstate travel are allowed but the number of flights will be reduced
4  2) Indian citizens and Permanent Residents are not allowed to travel abroad.
5  3) All visitors and foreign nationals are allowed to leave the country.
6  4) Indian citizens and Permanent Residents can return to India from abroad.
7  5) Indian citizens and Permanent Residents should download Aarogya Setu app
8  6) Most of the shops are not open for business
9
```

In the text area, you can edit, add and also delete your data. The text area is editable. After you finish manipulating the data inside the text area, you can keep your updated data by clicking the Save button. What it does is it will read and write the data from the text area into the SopIndia.txt. This can be done by implementing the PrintWriter function.



```
1  ---INDIA---
2
3  1) Interstate travel are allowed but the number of flights will be reduced
4  2) Indian citizens and Permanent Residents are not allowed to travel abroad.
5  3) All visitors and foreign nationals are allowed to leave the country.
6  4) Indian citizens and Permanent Residents can return to India from abroad.
7  5) Indian citizens and Permanent Residents should download Aarogya Setu app
8  6) Most of the shops are not open for business
9  7) Indian citizens coming back from outside the country will not need to do thermal screening.
10
```


MalaysiaSOP Class:

In the main page, when you clicked the Malaysia button, it will pop out new GUI. In this section, it will have three components, one JTextArea and 2 JButton, which are Save and Show SOP. This is where you can edit your SOP



JTextArea will show data about the SOP for the country Malaysia. To make the data visible, you have to click the Show SOP button for the text area to show the data. The data had been taken from a text file that is called SopMalaysia.txt. The text area had been implemented with a JScrollPane option.



```
1  ----KLIA----
2
3  1) Interstate travel are allowed.
4  2) Malaysian citizens and Permanent Residents are not allowed to travel abroad.
5  3) All visitors and foreign nationals are allowed to leave the country.
6  4) Malaysian citizens and Permanent Residents can return to Malaysia from abroad.
7  5) Malaysian citizens and Permanent Residents should download MySejahtera app.
8  6) Most of the shops are open for business.
9
```

In the text area, you can edit, add and also delete your data. The text area is editable. After you finish manipulating the data inside the text area, you can keep your updated data by clicking the Save button. What it does is it will read and write the data from the text area into the SopMalaysia.txt. This can be done by implementing the PrintWriter function.



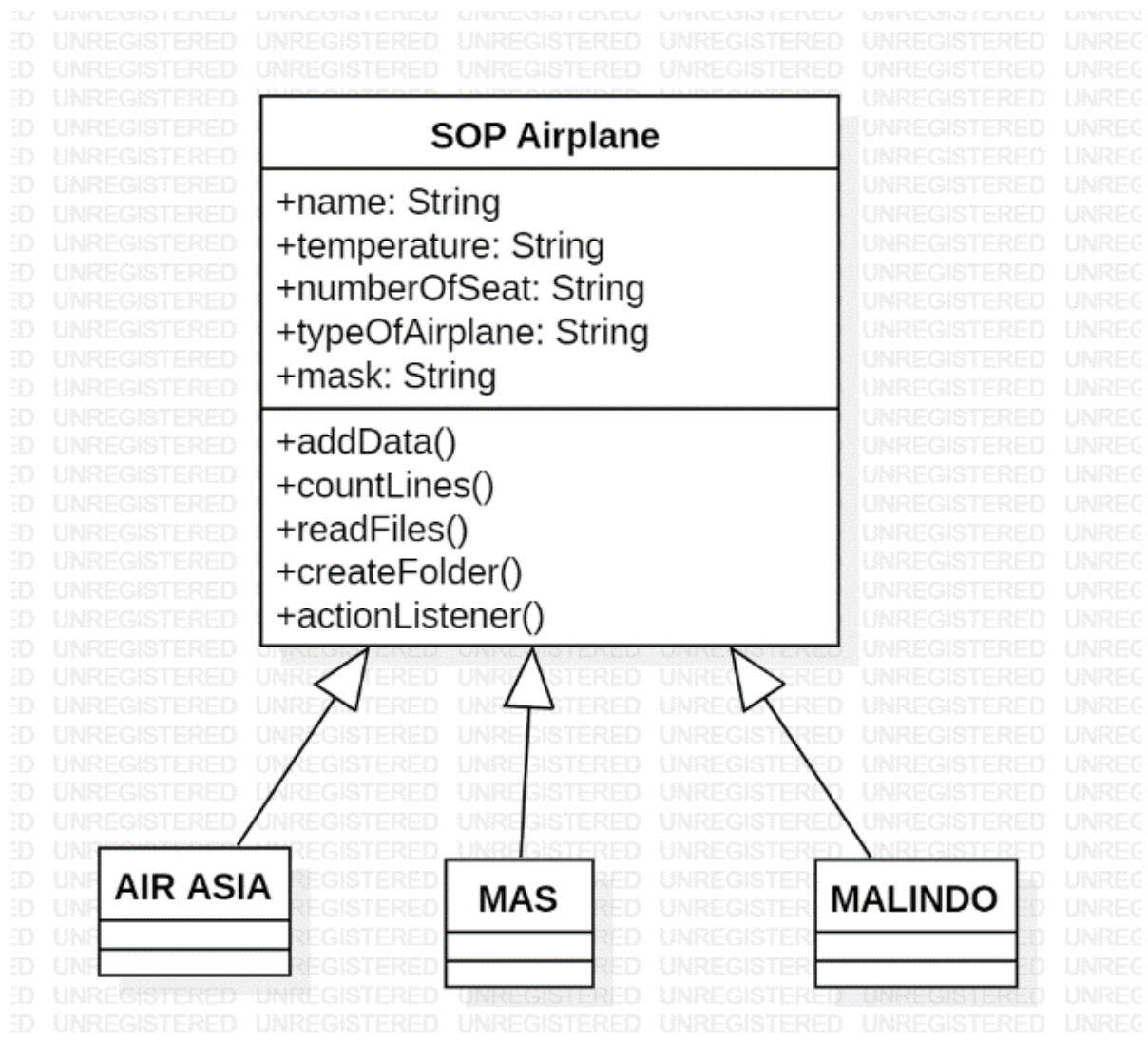
```
1  ----KLIA----
2
3  1) Interstate travel are allowed.
4  2) Malaysian citizens and Permanent Residents are not allowed to travel abroad.
5  3) All visitors and foreign nationals are allowed to leave the country.
6  4) Malaysian citizens and Permanent Residents can return to Malaysia from abroad.
7  5) Malaysian citizens and Permanent Residents should download MySejahtera app.
8  6) Most of the shops are open for business.
9  7) Malaysian coming back from other country have to pass the thermal screening test.
10
```

3.3 – THE CODE

The code can be found in the Source folder in the Github repository.

4.0 – AIRPLANE

4.1 – UML DIAGRAM



4.2 – PROGRAM DESCRIPTION

First, the program will ask for the information of the passenger, MT just have to key in all of the information of the passenger, name, temperature, number of seat, type of airplane and status, mask.

MT can click on the SOP for each airplane to refer before they adding the information.

If the MT didn't fill any one of the information, the system will be pop out an error message after he or she clicking the add button. Here is the important things MT have to concern, MT has require to select the row of information and click on update button after adding each of the information to make sure the format is correct.

After filling all information, MT can click on the add button to check whether all of the information is accurate with the SOP or not.

If one of the information didn't comply with the selected airplane's SOP, the program will ask the MT to check for the SOP again and amend the information. If all of the information are correct, the program will add all the information to table. After add please selected that information and click on update to update format.

The information that have been added can be delete also by clicking the delete button. After clicking the delete button, a confirm message will pop out to make the confirmation.

Furthermore, MT can also edit the data by clicking on the data that he or she wish to edit.

After editing, MT can just click on update and the information of the person will be updated.

Next, MT can also click on the search button to search for the name of passenger. This may help MT to confirm that whether that particular passenger has been register or not.

MT also have an option to reset all of the data by clicking the reset button.

After all, MT can click on the save all button to save all the information into a file. When clicking save all, an option pane will pop out to make sure MT has already update each of the information. After clicking yes, the information will be save in a file that will be on your desktop.

If clicking no the program will exit the option pane and pop out a message ask MT to update now.

The interface of airplane program:

The screenshot displays a software window titled "untitled tourism project". On the left is a vertical sidebar with buttons for "Placeholder text", "Airport", "Airlines", "Airplane" (which is currently selected), "Restaurant", "Hotel", and "Credits". The main area of the window is divided into two sections. The top section, titled "<<SOP AIRPLANE MALAYSIA>>", contains a form with the following fields: "NAME:" (text input), "Temperature:" (dropdown menu), "Number of Seat:" (dropdown menu), "Type of Airplane:" (dropdown menu), and "Mask:" (dropdown menu). The bottom section contains five buttons: "Add", "Reset", "Edit", "Update", and "Search". A horizontal scrollbar is visible at the bottom of the main content area.

Insert information and click on add:

The screenshot shows a web application window titled "untitled tourism project". On the left is a sidebar with a "Placeholder text" label and a list of categories: Airport, Airlines, Airplane, Restaurant, Hotel, and Credits. The "Airplane" category is selected. The main area displays a form titled "<<SOP AIRPLANE MALAYSIA>>". The form contains the following fields and values: "NAME:" with the value "jason", "Temperature:" with a dropdown menu showing "Below 37", "Number of Seat:" with a dropdown menu showing "30-40", "Type of Airplane:" with a dropdown menu showing "MAS", and "Mask:" with a dropdown menu showing "true". At the bottom of the form are five buttons: "Add", "Reset", "Edit", "Update", and "Search".

If one of the information are wrong, empty or doesn't comply with SOP:

This screenshot shows the same application window as before, but with a validation error. The "Temperature:" dropdown menu now shows "Above 37". A "Message" dialog box is overlaid on the form, containing an information icon and the text "Please check and follow SOP of the selected Airplane". The dialog box has an "OK" button. The form fields and buttons remain the same as in the previous screenshot.

The program when scroll down:

The screenshot shows a window titled "untitled tourism project". On the left is a sidebar with buttons for "Placeholder text", "Airport", "Airlines", "Airplane", "Restaurant", "Hotel", and "Credits". The main area contains a form with the following fields and controls:

- Number of Seat:** A dropdown menu currently showing "Below 37".
- Type of Airplane:** A dropdown menu currently showing "MAS".
- Mask:** A dropdown menu currently showing "true".
- A vertical stack of buttons: "Add", "Reset", "Edit", "Update", "Search", and "Delete".
- At the bottom, there are three buttons: "SOP AIR ASIA", "SOP MAS", and "SOP MALINDO", followed by a "Save all" button.

The data after update. In a correct format:

The screenshot shows the same window after an update. The main area now displays a table with the following data:

| <<SOP AIRPLANE MALAYSIA>> | | | | | | |
|---------------------------|----------|-------|-----|------|--|--|
| NAME: | Below 37 | 30-40 | MAS | true | | |
| Jason | Below 37 | 30-40 | MAS | true | | |

Below the table, the same vertical stack of buttons ("Add", "Reset", "Edit", "Update", "Search") is visible. The sidebar remains the same.

The program after clicking reset:

The screenshot shows a Java Swing window titled "untitled tourism project". The window contains a form for "SOP AIRPLANE MALAYSIA". On the left is a vertical menu with buttons for "Placeholder text", "Airport", "Airlines", "Airplane", "Restaurant", "Hotel", and "Credits". The main area of the form has the following fields and controls:

- NAME:** A text field containing "jason".
- Temperature:** A dropdown menu.
- Number of Seat:** A dropdown menu.
- Type of Airplane:** A dropdown menu.
- Mask:** A dropdown menu.
- Buttons:** "Add", "Reset", "Edit", "Update", and "Search" are arranged vertically.
- Table:** A table with 6 columns. The first row contains the values: "jason", "Below 37", "30-40", "MAS", and "true".

| <<SOP AIRPLANE MALAYSIA>> | | | | | |
|---------------------------|-------|----------|-------|-----|------|
| NAME: | jason | Below 37 | 30-40 | MAS | true |

The program after select the information and click on edit button:

untitled tourism project

Placeholder text

NAME: jason

Temperature: Below 37

Number of Seat: 30-40

Type of Airplane: MAS

Mask: true

Buttons: Add, Reset, Edit, Update, Search

The program after select the information and insert update information, click on update button:

untitled tourism project

Placeholder text

NAME: jason ling

Temperature: Below 37

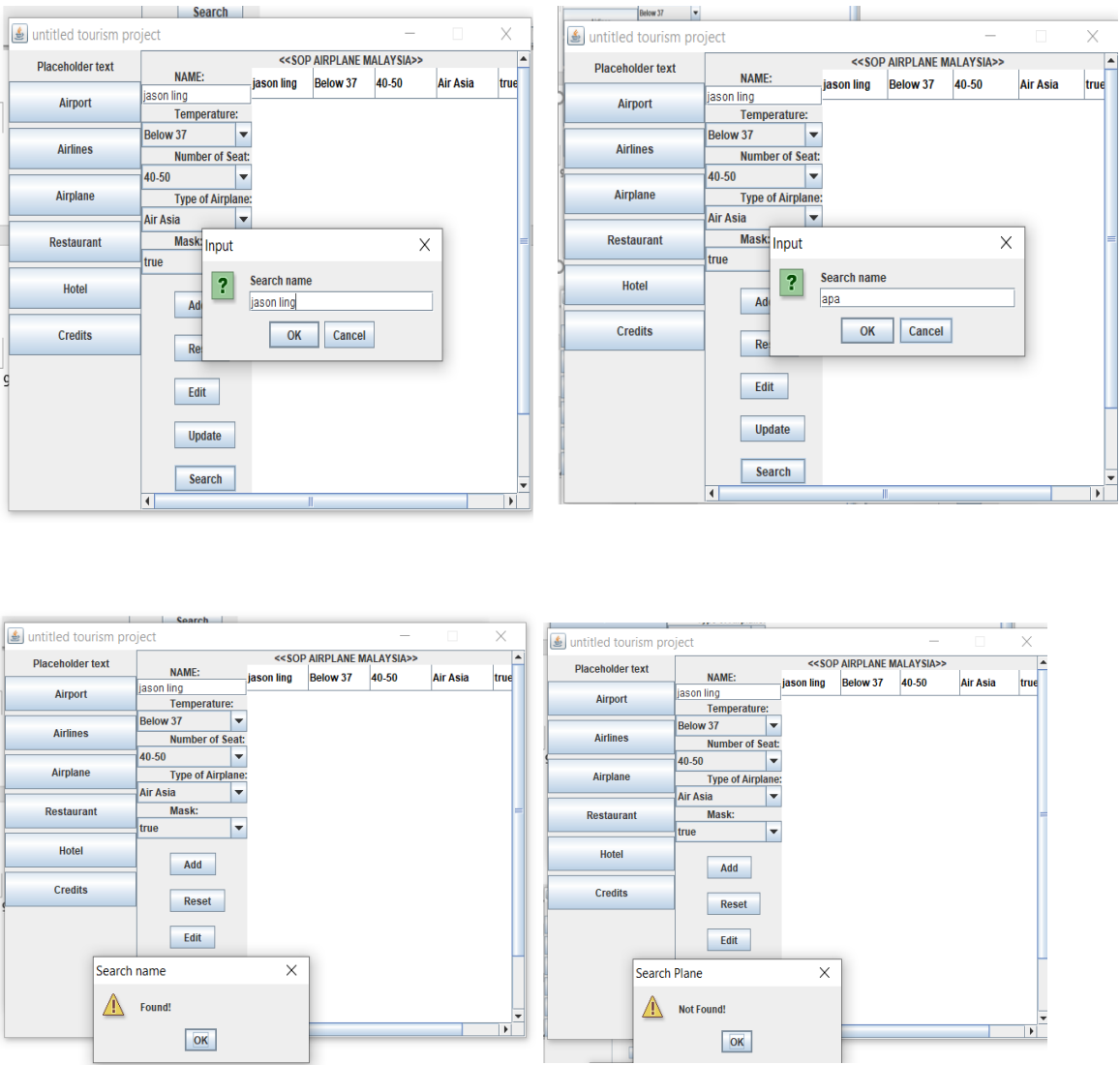
Number of Seat: 40-50

Type of Airplane: Air Asia

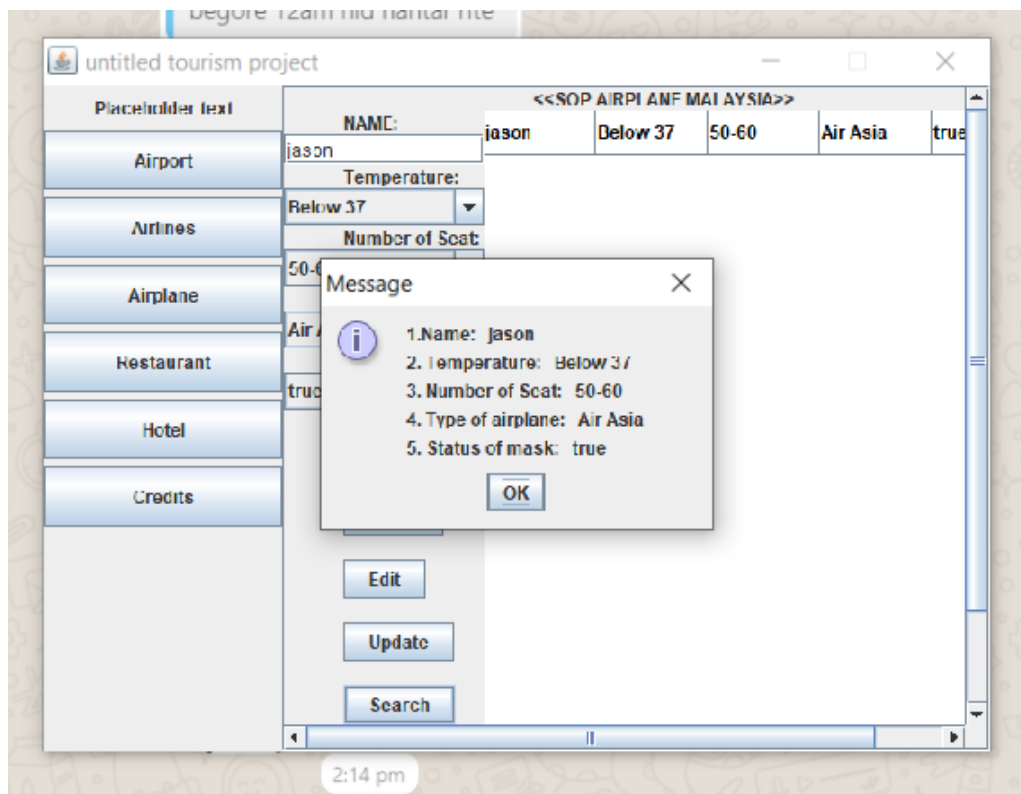
Mask: true

Buttons: Add, Reset, Edit, Update, Search

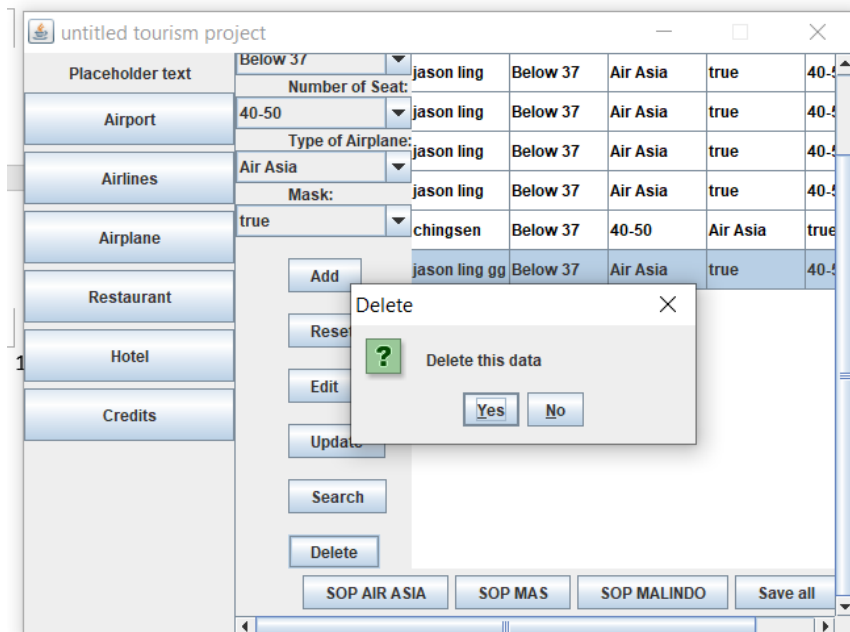
The program search for the name of passenger if found and if not found:



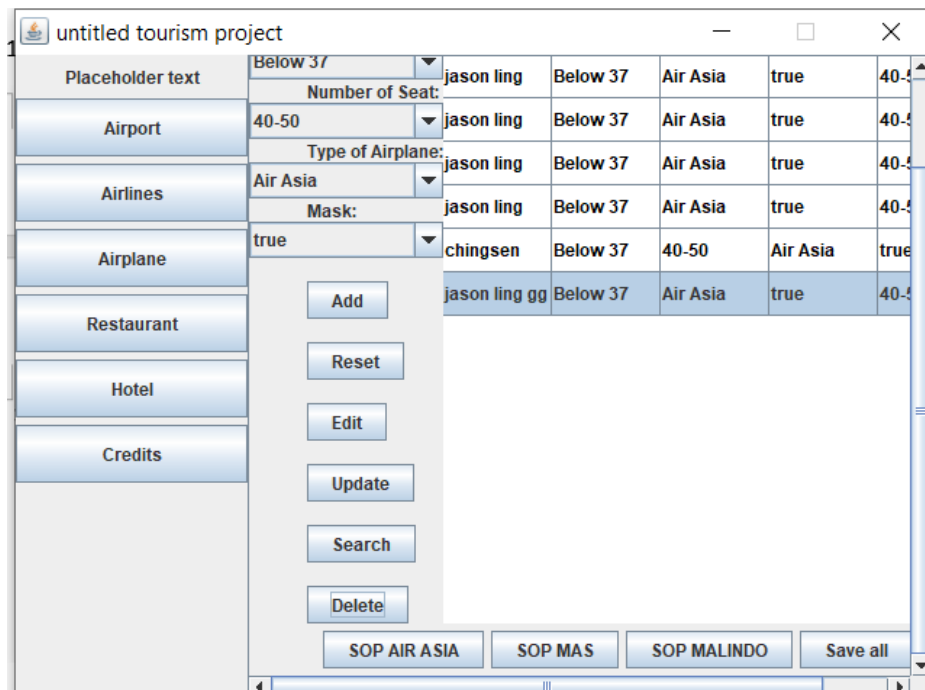
If the search is found



The program when select the information and clicking on delete button:



If click no:



If click yes and as you can see only the number for information format has been updated:

| Name | Number of Seat | Type of Airplane | Mask | Checkbox |
|---------------|----------------|------------------|----------|----------|
| jason ling | Below 37 | Air Asia | true | 40-50 |
| jason ling | Below 37 | Air Asia | true | 40-50 |
| jason ling | Below 37 | Air Asia | true | 40-50 |
| chingsen | Below 37 | 40-50 | Air Asia | true |
| jason ling gg | Below 37 | Air Asia | true | 40-50 |

When click on SOP SIR ASIA:

Message

1. Temperature cannot above 37 degree celcius
2. Number of passenger cannot more than 110
3. All passenger must wearing a mask
4. Must fill in all of the data

OK

When click on SOP MAS:

untitled tourism project

| | | | | | | |
|------------------|----------|---------------|----------|----------|------|------|
| Placeholder text | Below 37 | jason ling | Below 37 | Air Asia | true | 40-5 |
| Airport | 40-50 | jason ling | Below 37 | Air Asia | true | 40-5 |
| Airlines | Air Asia | jason ling | Below 37 | Air Asia | true | 40-5 |
| Mask: | chingsen | Below 37 | 40-50 | Air Asia | true | |
| Airplane | true | jason ling gg | Below 37 | Air Asia | true | 40-5 |

Message

1. Temperature cannot above 37 degree celcius
2. Number of passenger cannot more than 120
3. All passenger must wearing a mask
4. Must fill in all of the data

OK

Search

Delete

SOP AIR ASIA SOP MAS SOP MALINDO Save all

When click on SOP MALINDO:

untitled tourism project

| | | | | | | |
|------------------|----------|---------------|----------|----------|------|------|
| Placeholder text | Below 37 | jason ling | Below 37 | Air Asia | true | 40-5 |
| Airport | 40-50 | jason ling | Below 37 | Air Asia | true | 40-5 |
| Airlines | Air Asia | jason ling | Below 37 | Air Asia | true | 40-5 |
| Mask: | chingsen | Below 37 | 40-50 | Air Asia | true | |
| Airplane | true | jason ling gg | Below 37 | Air Asia | true | 40-5 |

Message

1. Temperature cannot above 37 degree celcius
2. Number of passenger cannot more than 140
3. All passenger must wearing a mask
4. Must fill in all of the data

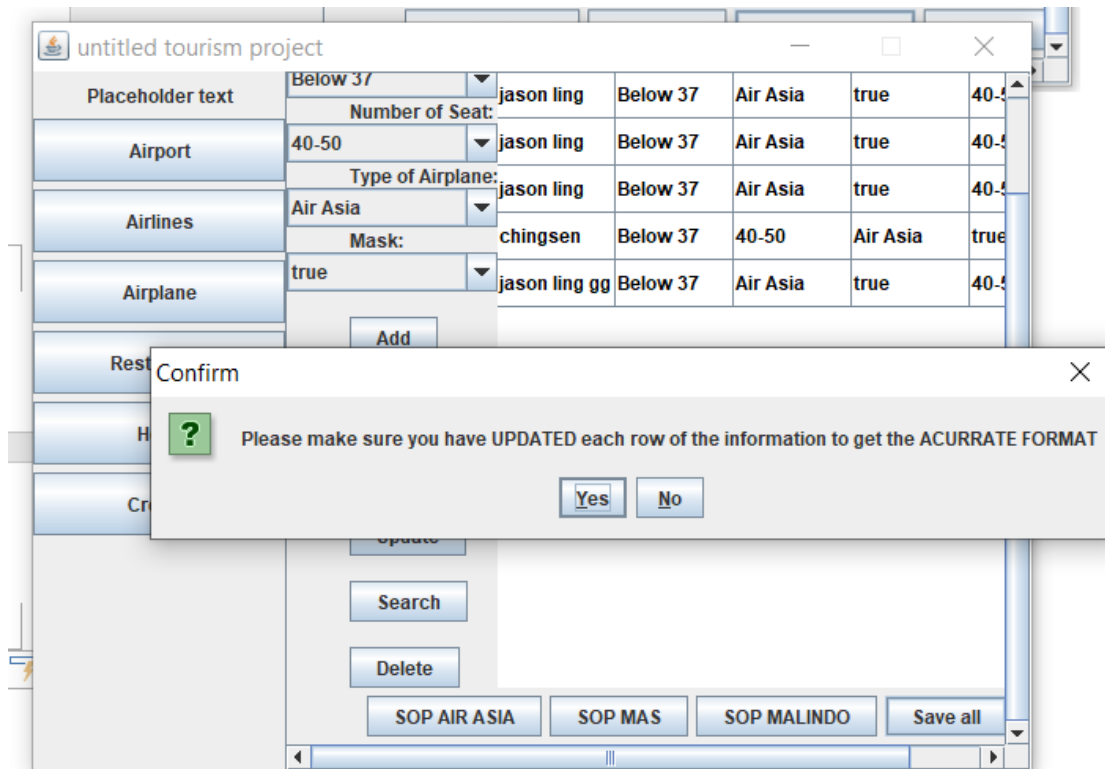
OK

Search

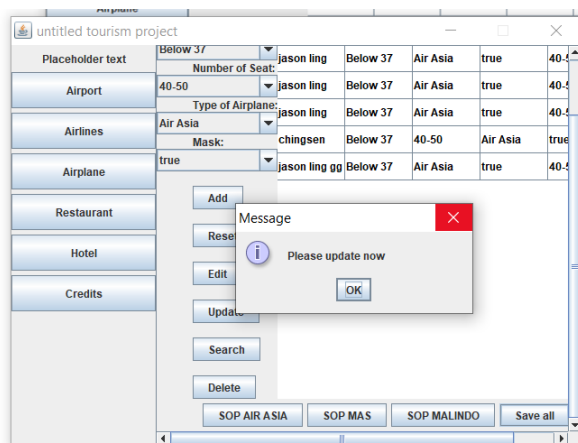
Delete

SOP AIR ASIA SOP MAS SOP MALINDO Save all

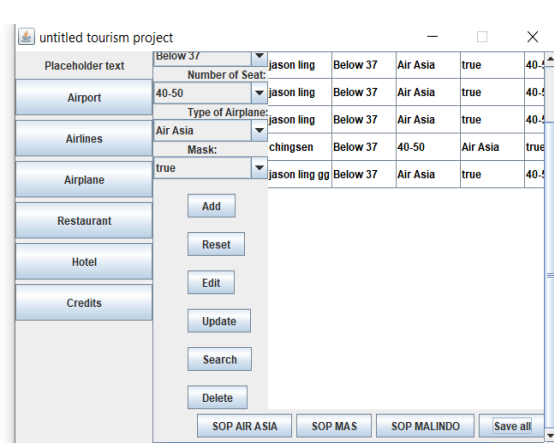
When click on save all button:



If clicking no



If clicking yes

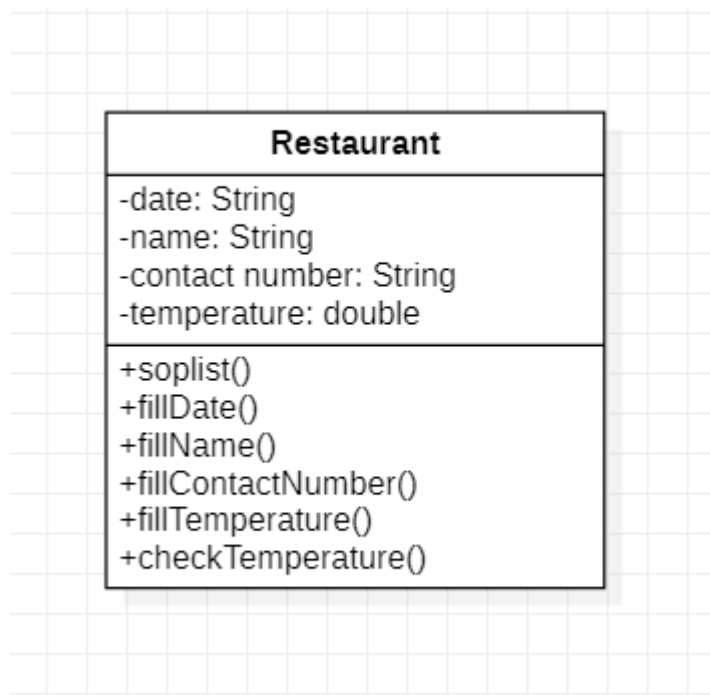


4.3 – THE CODE

The code is available in the Source folder in the Github repository.

5.0 – RESTAURANT

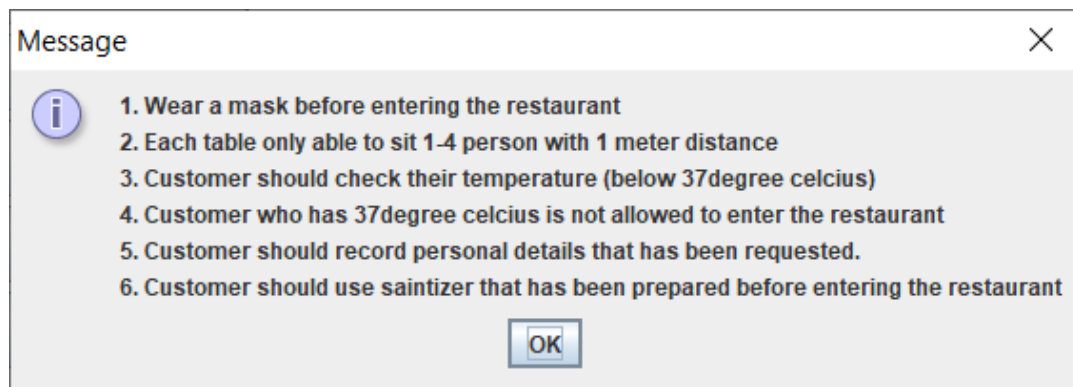
5.1 – UML DIAGRAM



5.2 – PROGRAM DESCRIPTION

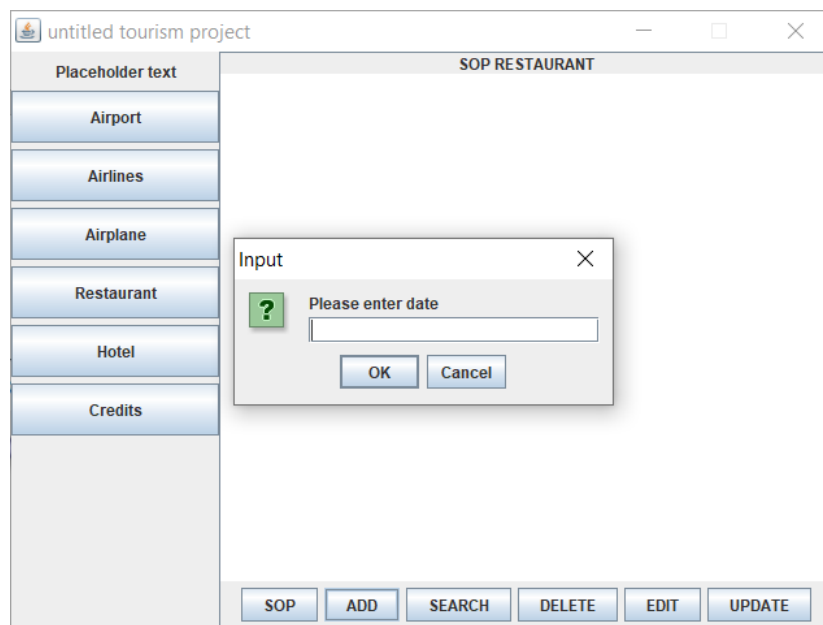
Basically it's very easy to be used. User able to check on Standard of Procedure (SOP) that has been commonly used in all restaurant while clicked on SOP button. User able to repeatedly to check on SOP in case they forget it. Besides this system is created to record all the details of customer who are going to enter the restaurant so the government able to find out the group of people once someone is affected with COVID-19. User is required to enter date, name, contact number and temperature. Once user's temperature is higher than 37 Celsius Degree, user will be informed that he/she is not allowed to enter the restaurant. User able to use the "Search" button to search the details that they wish to such as name or date. If data is found, customer's details will be pop out with date, name, contact number and temperature. This is to let user save their time instead of checking on the data one by one. On the "Delete" button, it's in case for those customers who might be change their mind after they entered their details. User able to select the details and delete the data by clicking on delete button. Next, for the edit button, this button is prepared to let customer to edit their details in case they entered the wrong details, such as wrong date, wrong name, invalid contact number or wrong temperature. The last button, Update, is prepared for user to back up their data on a text file in case they face some problem on the system. So they could check back the data while they need.

SOP button output:

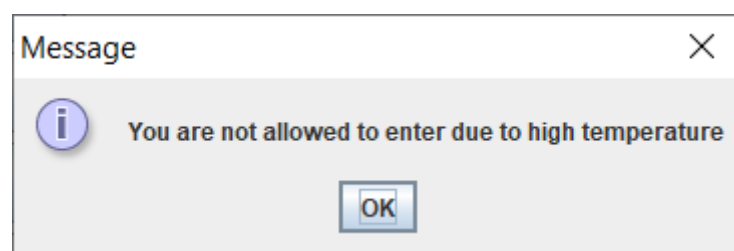


Add Button:

This will be one of the sample input which is using optionpane. The data of name, contact number and temperature will be required to input in the same way:



Sample dialog message while user enter the temperature which higher than 37:



Sample Data while User entered the data:

The screenshot shows a window titled 'untitled tourism project'. On the left is a sidebar with buttons: Placeholder text, Airport, Airlines, Airplane, Restaurant, Hotel, and Credits. The main area contains a table titled 'SOP RESTAURANT' with the following data:

| Placeholder text | | | | |
|------------------|-----------|--------|-----------|------|
| | 28/5/2019 | Tiang | 01523232 | 37.0 |
| Airport | 30/5/2019 | Jeri | 015454545 | 36.8 |
| Airlines | 31/5/2019 | Daniel | 018654327 | 38.0 |
| Airplane | | | | |
| Restaurant | | | | |
| Hotel | | | | |
| Credits | | | | |

At the bottom of the window are buttons: SOP, ADD, SEARCH, DELETE, EDIT, and UPDATE.

Optionpane for search data while click on search button:

The screenshot shows an 'Input' dialog box with a green question mark icon and the text 'Please enter the details you wish to search'. Below the text is a text input field. At the bottom are 'OK' and 'Cancel' buttons.

Sample output after search data:

The screenshot shows the 'untitled tourism project' window with the same sidebar and table as before. A 'Search Data' dialog box is open, displaying the following search results:

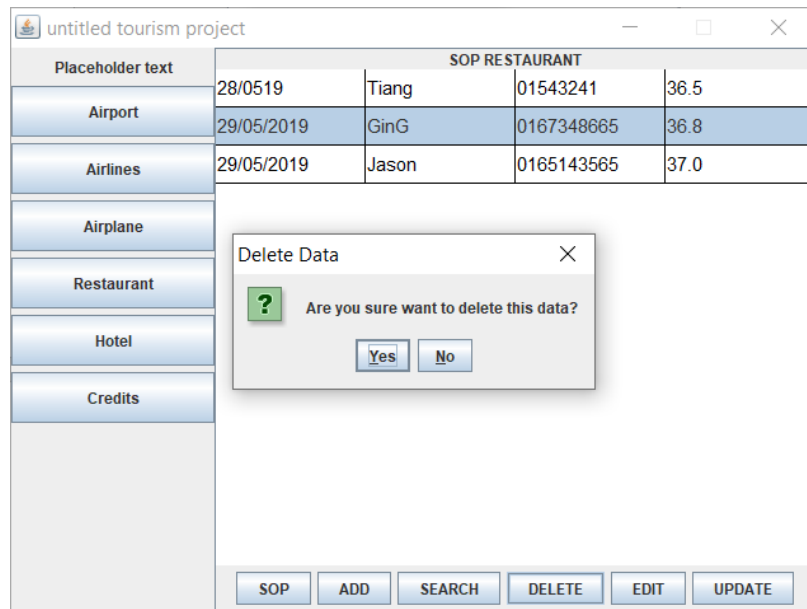
29/05/2019
GinG
0167348665
36.8Celcius Degree

At the bottom of the dialog box is an 'OK' button. The main window's table now shows updated data:

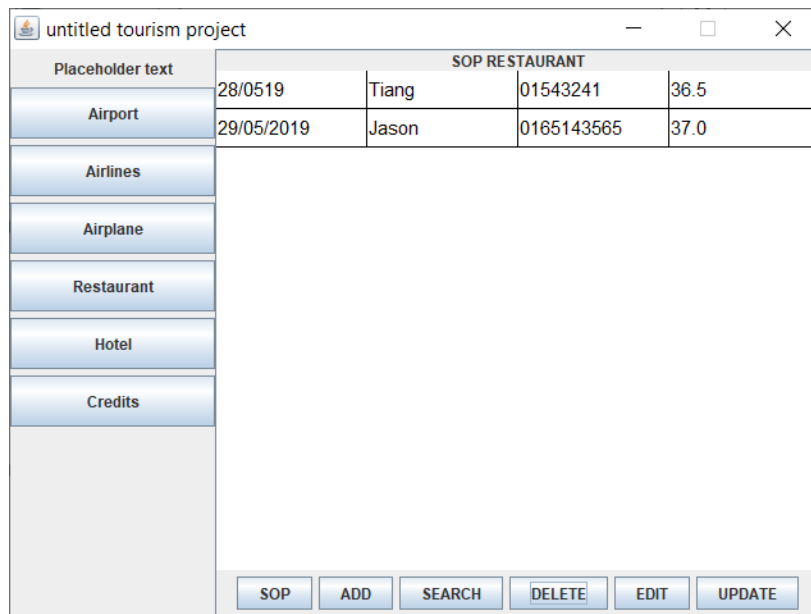
| Placeholder text | | | | |
|------------------|------------|-------|------------|------|
| | 28/0519 | Tiang | 01543241 | 36.5 |
| Airport | 29/05/2019 | GinG | 0167348665 | 36.8 |
| Airlines | 29/05/2019 | Jason | 0165143565 | 37.0 |
| Airplane | | | | |
| Restaurant | | | | |
| Hotel | | | | |
| Credits | | | | |

At the bottom of the window are buttons: SOP, ADD, SEARCH, DELETE, EDIT, and UPDATE.

Confirmation dialog while select row and click on delete button:



Data has been deleted:



Sample while user click on edit data:

The screenshot shows a web application window titled "untitled tourism project". On the left is a sidebar with a list of categories: Placeholder text, Airport, Airlines, Airplane, Restaurant, Hotel, and Credits. The "Restaurant" category is selected. The main area displays a table titled "SOP RESTAURANT" with two rows of data. An "Input" dialog box is open in the center, prompting the user to "Please enter date" with a text input field and "OK" and "Cancel" buttons. At the bottom of the main area are buttons for "SOP", "ADD", "SEARCH", "DELETE", "EDIT", and "UPDATE".

| SOP RESTAURANT | | | | |
|----------------|-------|------------|------|--|
| 28/0519 | Tiang | 01543241 | 36.5 | |
| 29/05/2019 | Jason | 0165143565 | 37.0 | |

Input

?

Please enter date

OK Cancel

SOP ADD SEARCH DELETE EDIT UPDATE

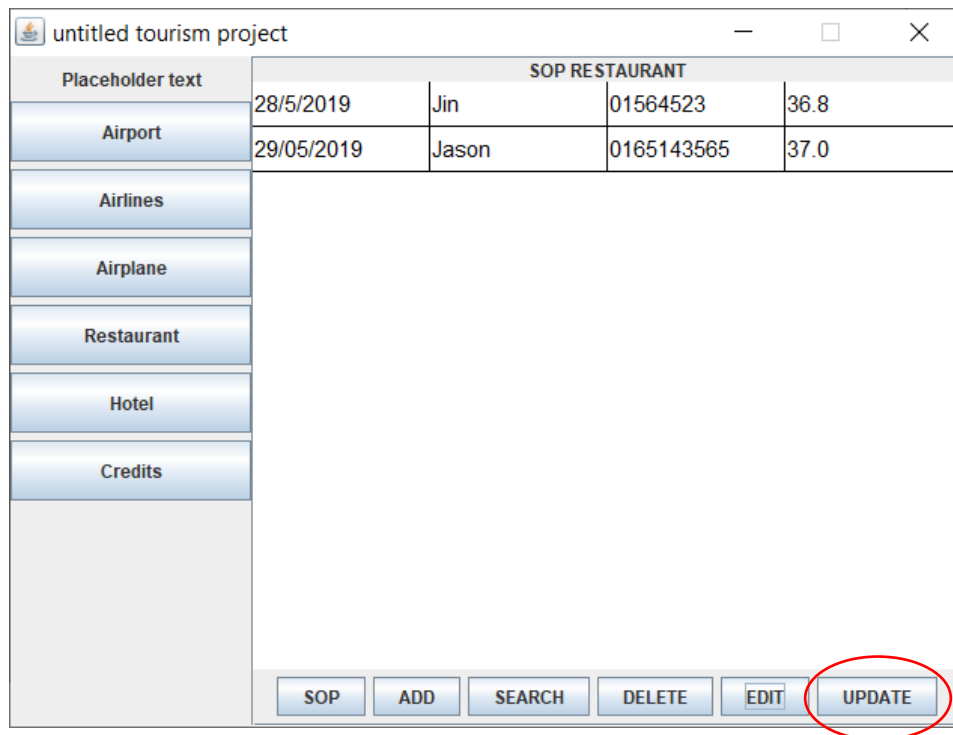
Sample data updated:

The screenshot shows the same web application window, but the data in the "SOP RESTAURANT" table has been updated. The first row now shows a date of "28/5/2019", a name of "Jin", and a value of "36.8". The "EDIT" button at the bottom is highlighted with a blue border, indicating it was the last action performed.

| SOP RESTAURANT | | | | |
|----------------|-------|------------|------|--|
| 28/5/2019 | Jin | 01564523 | 36.8 | |
| 29/05/2019 | Jason | 0165143565 | 37.0 | |

SOP ADD SEARCH DELETE EDIT UPDATE

While the button Update is clicked, all the data in will be save in text file:



| SOP RESTAURANT | | | | |
|------------------|------------|-------|------------|------|
| Placeholder text | 28/5/2019 | Jin | 01564523 | 36.8 |
| Airport | 29/05/2019 | Jason | 0165143565 | 37.0 |

Buttons: SOP, ADD, SEARCH, DELETE, EDIT, UPDATE (circled in red)

5.3 – THE CODE

The code is available in the Source folder in the Github repository.

APPENDIX

Project Github repository: <https://github.com/dfx81/Tourism-Project/>