Programming Methodology

Group Name: "AIRROR"

Members: 1) Chotiwat Silarak (#6731312821) 2) Yanaphat Chatphokhinkun (#6731313421)
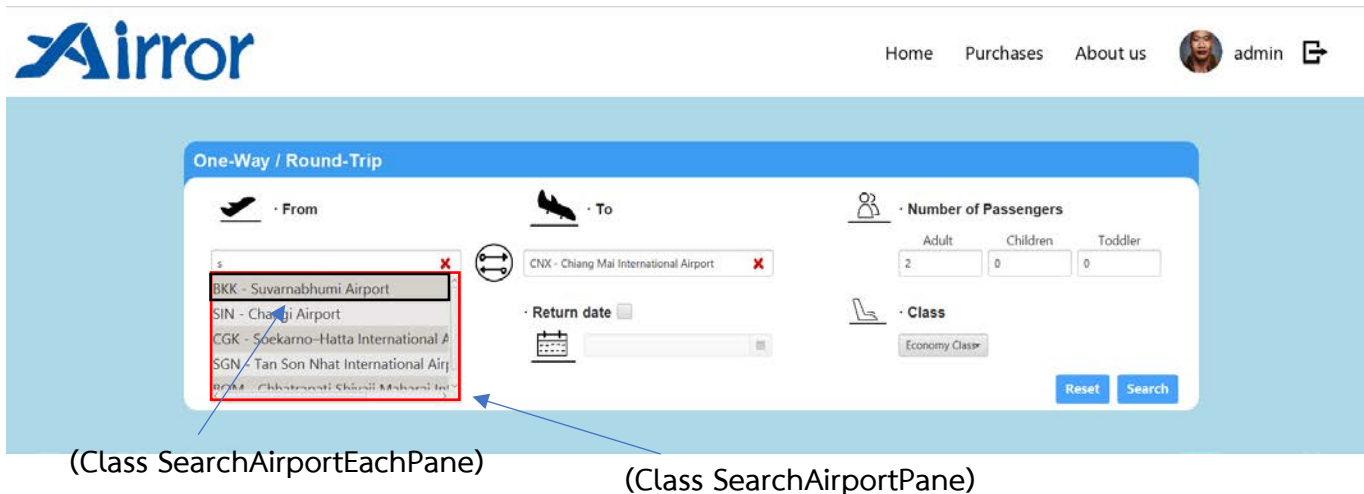
UML Diagram: Click Here!

What is our project about?

We are making application; specifically, "Airplane Booking" Application. Our application provides basic functions such as booking, purchases list and sorting features. In addition, it contains user sign in and sign up function.

- **Login Page (Class LoginPage)**

- **Main Page / Home Page (Class MainPage)**

(Class SearchAirportEachPane)

(Class SearchAirportPane)

- Available Filghts Page (Class FlightAvailablePage)

BKK(Bangkok) ➡ **CNX(Chiang Mai)**
Suvarnabhumi Airport   Chiang Mai International Airport

07 May 2025 / Adult : 2 Children : 0 Toddler : 0 / Economy Class

Back

Choose Airlines
- ☐ Goat airline
- ☐ Penguin airline
- ☐ Shiba airline
- ☐ Giraffe airline
- ☐ Chicken airline
- ☐ Banana airline

Price ▲   Duration ▲   Arrival Time ▲

Penguin airline   05:00 BKK   1 hr 15 mins   06:15 CNX   ฿2000.0 Per Person   **Select**

Goat airline   11:15 BKK   9 hrs 58 mins   21:13 CNX   ฿5536.0 Per Person   **Select**

(Class SearchFlightAvailablePane)

(Class SearchFlightAvailableEachPane)

- Purchases Page (Class PurchasePage)

🛒 **Purchases**

Up Coming   Past

date	depart	origin	arrive	destination	booking number
07/05/2025	18:00	Suvarnabhumi Airport(BKK)	19:15	Chiang Mai International Airport(CNX)	1
07/05/2025	21:00	Chiang Mai International Airport(CNX)	04:53	Suvarnabhumi Airport(BKK)	2

(Class SearchPurchaseEachPane)

(Class SearchPurchasePane)

- **About us Page (Class AboutUsPage)**
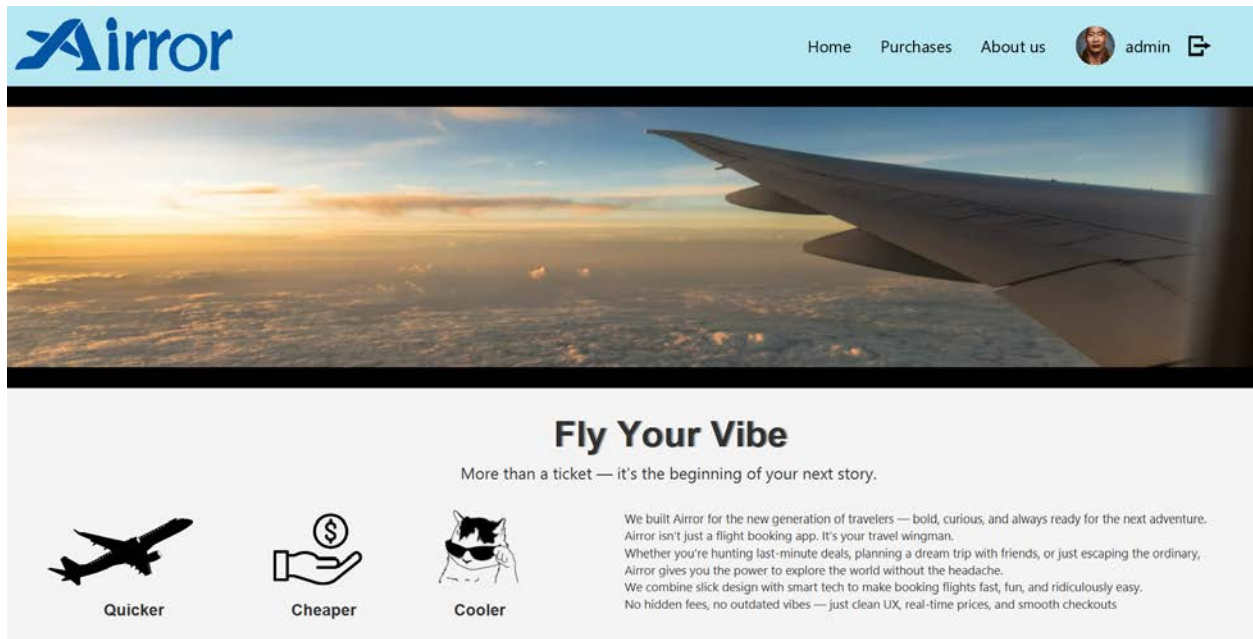
Implementation Details

Please note that : + (public), - (private), # (protected), <u>underlined</u> (static), *italic* (abstract), All_CAPS (final)

1. Package application

- This package contains our application inside.

1.1 Class MyApplication

Fields

- <u>Scene scene</u>	Scene of the Application
- <u>Stage stage</u>	Stage of the Application
- <u>ArrayList<Parent> listOfRoots</u>	List of all roots/parents node in the scene graph

Methods

+ void main(String[] args)	Main Function
+ void start(Stage stage)	Start the Application
+ void loadData()	Loading the data such as flights, purchases, airline name and etc. from Text File (Via Utils.IOReaderWriter.java)
+ void keepData()	Putting the data such as flights, purchases, username and password, and etc. to Text File (Via Utils.IOReaderWriter.java)
+ void setStageStyle(Stage stage)	Setting the stage's style and add Event Handler
+ Getter & Setter	-

2. Package Exceptions

- This package contains all the exceptions we used in our Application.

2.1 Class IncorrectAndPartialFillFormException

Method

+ IncorrectAndPartialFillFormException()	Initiate incorrect and partial filling of the form Exception

2.2 Class IncorrectFillFormException

Method

+ IncorrectFillFormException()	Initiate incorrect filling of the form Exception

2.3 Class PartialFillFormException

Method

+ PartialFillFormException()	Initiate partial filling of the form Exception

3. Package interfaces

- This package contains the interface we use.

3.1 Interface Searchable -> Used by any classes which can search or filter the data.

Method

void addSearchEachPane(Object obj)	**Different among pages**

4. Package logics

- This package contains all the information we used e.g. flight data, purchase data, request data.

4.1 Class FlightData -> represent each flight

Fields

- String departAbbr	Abbreviation of a departure (such as BKK, CNX)
- String destinyAbbr	Abbreviation of a destination (such as BKK, CNX)
- String airlineName	Name of an airline
- String departAirportName	Name of departure's airport
- String destinyAirportName	Name of destination's airport
- ImageView airlineImage	Image of an airline's logo

- String departTime	Time of departure (such as 19:14)
- String arrivalTime	Time of arrival (such as 19:14)
- double price	Price of a flight
- LocalDate departDate	Date of departure

Methods

+ FlightData(String departAbbr, String departAirportName, String destinyAbbr, String destinyAirportName, String airlineName, ImageView airlineImage, String departTime, String arrivalTime, double price, LocalDate departDate)	Initialize all fields
+ String toString()	Return departAbbr + "," + departAirportName + "," + destinyAbbr + "," + destinyAirportName + "," + airlineName + "," + departTime + ","+ arrivalTime + "," + price + "," + departDate ("dd/MM/yyyy")
+ Getter & Setter	-

4.2 Class PurchaseData -> represent each purchase

Fields

- String classes	Class (Such as Economy Class)
- int adultField	The Number of adults
- int childrenField	The Number of children
- int toddlerField	The Number of toddlers
- FlightData flightData	The purchased flight

methods

+ PurchaseData(FlightData flightData, String classes, int adultField, int childrenField, int toddlerField)	Initialize all fields
+ String toString()	Return flightData.toString() + "," + classes + "," + adultField + "," + childrenField + "," + toddlerField
+ Getter & Setter	-

4.3 Class RequestData -> represent request data such as departure, destination, classes, etc.

Fields

- String departField	A depature (such as "CNX – Chiang Mai International Airport")
- String destinyField	A destination (such as "BKK – Don Mueng International Airport")
- LocalDate departDate	Date of depature
- LocalDate destinyDate	Date of destination
- int adultField	The number of adults
- int childrenField	The number of children
- int toddlerField	The number of toddlers
- boolean isReturn	True if it is two-way flights. Otherwise, it is false.
- String classes	Class (Such as Economy Class)

Methods

+ RequestData(String departField, String destinyField, LocalDate departDate, LocalDate destinyDate, int adultField, int childrenField, int toddlerField, Boolean isReturn, String classes)	Initialize all fields
+ RequestData(RequestData requestData)	Copy constructor
+ Getter & Setter	-

4.4 UserData (No Implementation, just for the future user features)

5. Package pages

- This package contains all the pages appearing in Application.

5.1 *Abstract Class Page* –> Smallest unit of page (Every page has to extend this)

Fields

# Canvas canvas	A canvas of the instance

Methods

+ *void setHeader(GraphicsContext gc)*	Set header of the page
+ *void setStyle()*	Style the page
+ void setTopLeftAnchor(Node node, double left, double top)	AnchorPane.setTop(node, top) AnchorPane.setLeft(node, left)
+ Getter	

5.2 Class LoginPage extends Page –> login / sign up page

Fields

- Map<String, String> passwords	A map which has a username as a key and a password as a value
+ String loginUsername	A Username String

Methods

+ LoginPage()	Initialize Login page
+ void setHeader(GraphicsContext gc)	No implementation needed
+ void setStyle()	Set style of the login page
+ Getter & Setter	-

5.3 Class MainPage extends Page -> Main page of the Application

Fields

+ MainPage mainPageInstance	An instance of MainPage
+ ScrollPane departSearchBar	A searchbar for departure
+ ScrollPane destinySearchBar	A searchbar for destination
+ TextField departField	An input TextField for depature
+ TextField destinyField	An input TextField for destination
+ DatePicker departDatePicker	A DatePicker for picking depart date
+ DatePicker destinyDatePicker	A DatePicker for picking returning date
+ TextField adultField	An input TextField for the number of adults

+ TextField childrenField	An input TextField for the number of children
+ TextField toddlerField	An input TextField for the number of toddler
+ CheckBox checkbox	A CheckBox for checking whether the flight will be two-way or not
+ ChoiceBox<String> choiceBox	A ChoiceBox for selecting class (Such as Economy Class)

Methods

- MainPage()	Initialize home page
+ void *setHeader(GraphicsContext gc)*	Create header section using BorderPane
+ void *setStyle()*	Set width of MainPage to UIComponent.USER_MAX_SCREEN_WIDTH
+ void setMiddle()	Create and style content section (A booking system)
+ void reset()	Reset the content section (A booking system)
+ MainPage getInstance(Boolean bool)	Create MainPage's instance Bool specifies whether or not we are going to get new mainPageInstance (True) or not (False)

5.4 Class FlightAvailablePage extends Page -> Page that show available flights to user

Fields

+ <u>ArrayList<FlightData> flightsList</u>	All available flights
+ ArrayList<String> airlinesCondition	Conditions for filtering
+ <u>Map<String, String> townName</u>	A map which has an abbreviated city name as key and full name as value (Such as ("BKK", "Bangkok"))
- RequestData requestData	An RequestData for finding flights

Methods

+ FlightAvailablePage(RequestData requestData)	Initialize available flights page
+ *void setStyle()*	No implementation needed
+ *void setHeader(GraphicsContext gc)*	Create header section of flight available page
+ void setMIddle(GraphicsContext gc, RequestData requestData)	Create content section of flight available page
+ <u>void setSortingEventListener(SearchFlightAvailablePane searchPane, ImageView imageView, Comparator comparator)</u>	Set event handler to ImageView, which when it is clicked, the available flights will sort determined by comparator
+ void getByConditions(SearchFlightAvailablePane searchPane, RequestData requestData)	Get the flights determined by the condition strings (Airline names)
+ void setAirlineEventListener(SearchFlightAvailablePane	Add event handler to checkbox for filtering by the

searchPane, RequestData requestData, CheckBox checkbox, String condition)	condition strings (Airline names)
+ Getter & Setter / Getter & Setter	-

5.5 Class PurchasePage extends Page -> Page showing all the purchased flights by user

Fields

+ ArrayList<PurchaseData> pendingList	An ArrayList which contains the pending flights which are not confirmed yet.
+ ArrayList<PurchaseData> purchasesList	An ArrayList which contains all of PurchaseData

Methods

+ PurchasePage()	Initialize purchase page
+ *void setHeader(GraphicsContext gc)*	Create header section of purchase page
+ void setMiddle(GraphicsContext gc)	Create content section of purchase page (Purchases List)
+ *void setStyle()*	No implementation needed
+ Getter & Setter	-

5.6 Class AboutUsPage extends Page -> About us page

Field

+ <u>AboutUsPage aboutUsPageInstance</u>	Instance of AboutUsPage

Methods

- AboutUs()	Initialize about us page
+ *void setHeader(GraphicsContext gc)*	Create header section of about us page
+ void setMiddle(GraphicsContext gc)	Create content section of about us page
+ void setStyle()	No implementation needed
+ <u>Getter</u>	-

6. Package panes

- The package contains all the important component / pane of the page

6.1 Class SearchAirportPane extends VBox implements Searchable

-> A pane that shows search result in the MainPage

Field

+ <u>ArrayList<String> airports</u>	List of all airports

Methods

+ SearchAirportPane(String condition)	Initialize SearchAirportPane with the given string condition
+ *void addSearchEachPane(Object obj)*	Add SearchAirportEachPane to SearchAirportPane
+ <u>Getter</u>	-

6.2 Class SearchFlightAvailablePane extends VBox implements Searchable

-> A pane that shows flights result in FlightAvailablePage

Field

+ ArrayList<FlightData> flightsSelected	Contain flights which satisfies the requestData

Methods

+ SearchFlightAvailablePane(RequestData requestData)	Initialize SearchFlightAvailablePane with the given RequestData
+ *void addSearchEachPane(Object obj)*	Add SearchFlightAvailableEachPane to SearchFlightAvailablePane
+ Getter	-

6.3 Class SearchPurchasePane extends VBox implements Searchable

-> A pane that shows all the purchases by the user in the PurchasePage

Methods

+ SearchPurchasePane()	Initialize SearchPurchasePane (Purchase List)
+ *void addSearchEachPane(Object obj)*	Add SearchPurchaseEachPane to SearchPurchasePane

6.4 Class SearchAirportEachPane extends BorderPane -> A pane representing each result which is in SearchAirportPane

Field

- String name	A name of an airport

Methods

+ SearchAirportEachPane(String name, double prefWidth)	Initialize SearchAirportEachPane with the given string name and double prefWidth
+ Getter	-

6.5 Class SearchFlightAvailableEachPane extends AnchorPane -> A plane representing each flight which is in SearchFlightAvailablePane

Field

- FlightData flightData	FlightData of a flight

Methods

+ SearchFlightAvailableEachPane(FlightData flightData, RequestData requestData, double prefWidth, double prefHeight)	Initialize SearchFlightAvailableEachPane with the given FlightData RequestData and double prefWidth and prefHeight
+ setTopLeftAnchor(Node node, double left, double top)	AnchorPane.setTopAnchor(node, top); AnchorPane.setLeftAnchor(node, left);
+ Getter	-

6.6 Class SearchPurchaseEachPane extends AnchorPane -> A pane representing each purchase in SearchPurchasePane

Field

- PurchaseData purchaseData	PurchaseData of a purchase

Methods

+ SearchPurchaseEachPane(PurchaseData purchaseData, int number, SearchPurchasePane purchasePane)	Initialize SearchPurchaseEachPane with the given PurchaseData, the number(Booking number) and SearchPurchasePane
+ setTopLeftAnchor(Node node, double left, double top)	AnchorPane.setTopAnchor(node, top); AnchorPane.setLeftAnchor(node, left);
+ Getter	-

7. Package utils

- A package contains all the important functions apart from all of the above (Switching between page, Reading and Writing file, GUI)

7.1 Class Goto -> A class used for switching between page

Methods

+ void goToMainPage(Boolean bool)	Go to main page
+ void goToFlightAvailablePage(RequestData)	Go to available flights page

+ void goToPurchasePage()	Go to purchase page
+ void goToAboutUsPage()	Go to about us page
+ void goToLoginPage()	Go to login page
+ void back()	Go back to previous page

7.2 Class IOReaderWriter -> A class which used for reading and writing file both externally and internally. (External -> outside jar / Internal -> inside jar)

Field

+ Scanner sc	A scanner for reading file

Methods

+ ArrayList<String> getStringsFromTextFile(String path)	Read text file to ArrayList<String> internally (Inside a jar file)
+ ArrayList<String> getStringsFromTextFileExternally(String fileName) throws Exception	Read text file to ArrayList<String> externally (Outside a jar file)
+ ArrayList<FlightData> getListOfFlightData(String path)	Read text file to ArrayList<FlightData> internally
+ ArrayList<FlightData> getListOfFlightDataExternally(String fileName) throws Exception	Read text file to ArrayList<FlightData> externally
+ ArrayList<PurchaseData> getListOfPurchaseDataExternally(String fileName) throws Exception	Read text file to ArrayList<PurchaseData> externally

+ <u>Map<String, String> getMap(String path)</u>	Read text file to Map<String, String> internally
+ <u>Map<String, String> getMapExternally(String fileName)</u>	Read text file to Map<String, String> externally
+ <u>void writeStringsExternally(ArrayList<Object> objectsList, String filename)</u>	Write text file externally from ArrayList<Object>

7.3 Class UIComponent -> A class which contains useful GUI functions and nodes

Fields

+ <u>double USER_MAX_SCREEN_WIDTH</u>	Maximum screen's width
+ <u>double USER_MAX_SCREEN_HEIGHT</u>	Maximum screen's Height

Methods

+ <u>label getLabel(String name, int fontSize)</u>	Return Label with the "name" text and font size set to "fontSize"
+ <u>Text getText(String name, int fontSize)</u>	Return text with the "name" text and font size set to "fontSize"
+ <u>Text getText(String name, Font font)</u>	Return text with the "name" text and set font to "font"
+ <u>ImageView getImageView(String url_string, double fitHeight, Boolean isPreserveRatio)</u>	Return ImageView located at "url_string" and set it fitHeight to "fitHeight" and set PreserveRatio to "isPreserveRatio"
+ <u>Image getImage(String url_string)</u>	Return Image located at "url_string"
+ <u>Image getImage(String url_string, double requestedWidth, double</u>	Return Image located at "url_string" and set it width to "requestedWidth",

requestedHeight, boolean preserveRatio, boolean smooth)	set it height to "requestedHeight", set preserveRatio to "preserveRatio" and set smooth to "smooth"
+ TextField getTextField(String text, int prefWidth, int prefHeight)	Return TextField with the "text" text and set it prefWidth to "prefWidth" and set it prefHeight to "prefHeight"
+ void drawLine(double x1, double y1, double x2, double y2, GraphicsContext gc)	Draw line in canvas between (x1, y1) and (x2, y2)
+ int strHourToIntMin(String time)	Return the minutes required to go from 0:00 to time (e.g. time = 18:00 -> return 18 * 60 + 0 * 0
+ String intMinToStrHour(int time)	Reverse of strHourToIntMin(String time)
+ int getDuration(String departTime, String arrivalTime)	Return the difference in min between two points of time (e.g. departTime = 18:00, arrivalTime = 18:50 -> return 50)
+ boolean isBefore(String time1, String time2)	Return true if time1 is before time 2. Otherwise, return false.
+ void setFont(Label[] arr, Font font)	Set every label's font in arr to "font"
+ DatePicker getDatePicker()	Return new DatePicker()
+ CheckBox getCheckBox()	Return new CheckBox()
+ ChoiceBox<String> getChoiceBox()	Return new ChoiceBox<String>()
+ Button getButton(String name)	Return new Button(name)