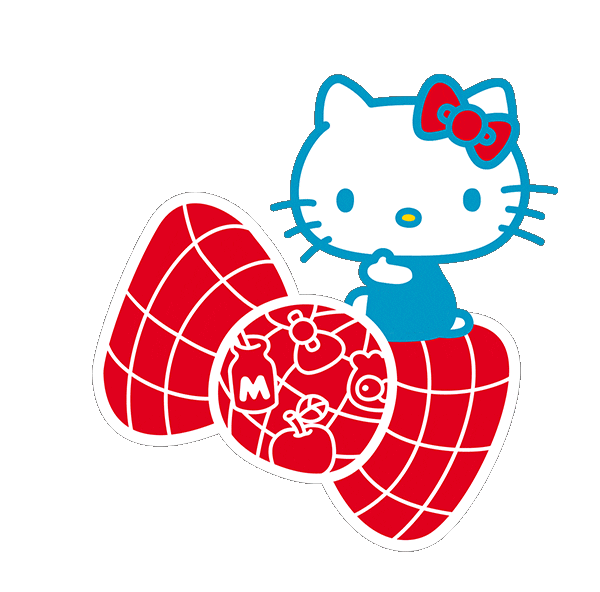
Sanrio Airlines



CIS 3270 Project Documentation

By: Sunny Hoang & Kimmy Wang

# Table of Contents

[**Table of Contents 2**](#_bdxyjocnjw4m)

[**Requirements 3**](#_do2vo7uiteef)

[Coding Requirements 3](#_ywofehx8ds3)

[Specifications of the Project 3](#_lzqc8ogujsi1)

[**UML Class Diagrams 4**](#_db7s53ev7jy3)

[**Data Model 6**](#_2titf0cj94wn)

[**Functionality 6**](#_2th1puhn0s54)

[Main Functionality of the Application 6](#_kodnzdrfcuwz)

[Splash Screen 6](#_nvk1nhcadwwm)

[Main Menu 7](#_eripte9qrwea)

[Before A User Is Logged In 7](#_o7jla02csk54)

[After A User Is Logged In 7](#_apv8gd787tn4)

[An option to go back to the main menu in any case 7](#_yl9ntcv7np7x)

[Login Page 7](#_q6rlwy4h51og)

[Forgot Password Page 7](#_jsuqs4yq81is)

[User Registration 7](#_plzunf1fsepe)

[User Searches For Flights Based on Specified Criteria 8](#_y7ts3etf7nlh)

[User Books a Flight After 3 Criteria is Checked 9](#_cnz2d3z41ejf)

[My Trips Page 10](#_z8fykryvz25i)

[Upon User Logging Out, Previously Booked Flights Should Be In Their Account 10](#_2uq68m5zy0e6)

[Initializing the Airport Database 10](#_sbiz03pt9ro)

# Requirements

## **Coding Requirements**

To demonstrate our comprehensive knowledge of Java, the team has successfully coded the application within these guidelines:

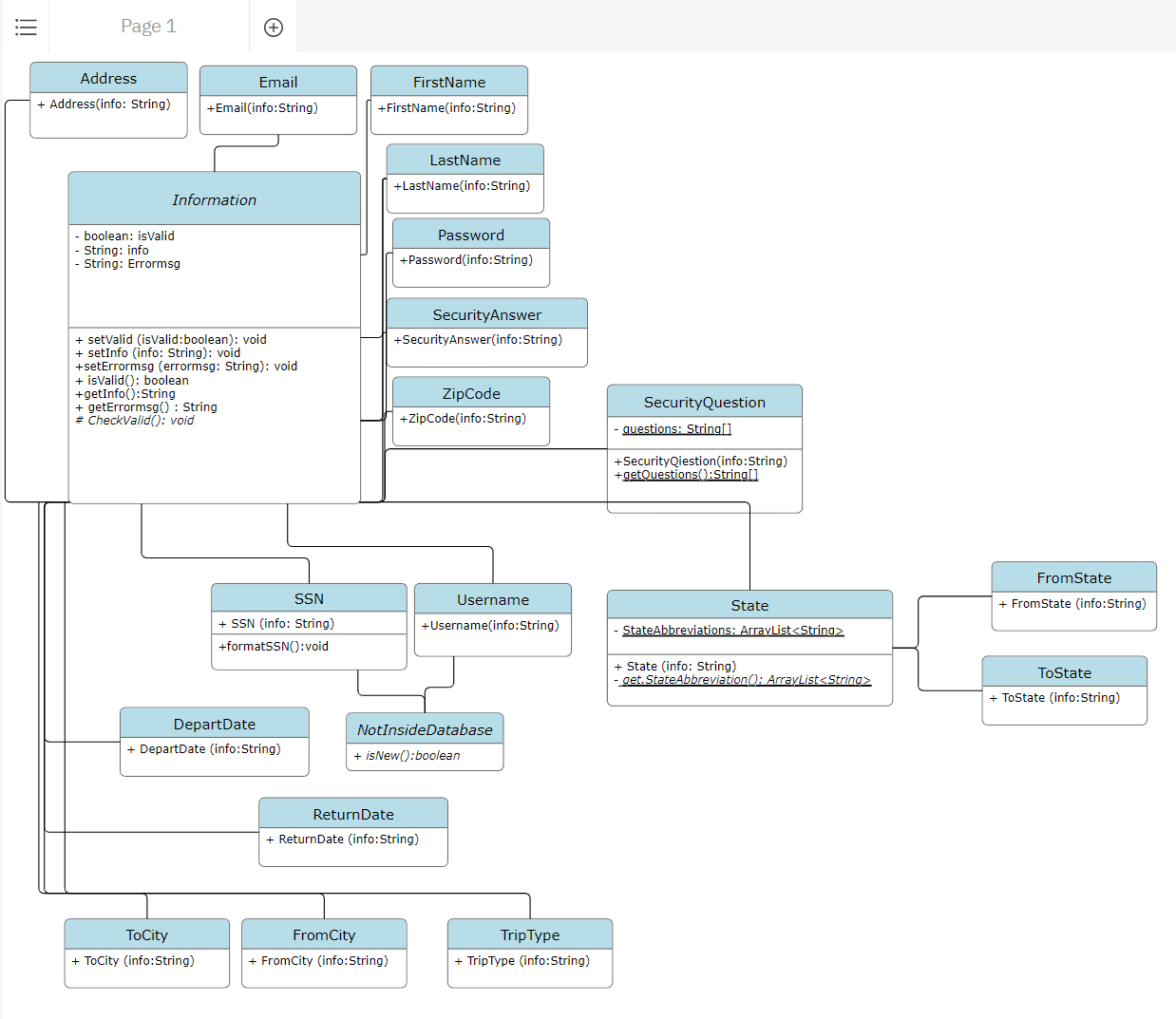
* Program was coded with at least 4 different classes
* At least 3 concepts of Encapsulation, Inheritance, Polymorphism, or Composition was utilized
* At least 1 Abstract class and 1 Interface was created and utilized
* Code has proper comments
* Exception handling was implemented in every section of the code to prevent the application from crashing
* Database programming part was done using JDBC
* Azure was used as RDBMS

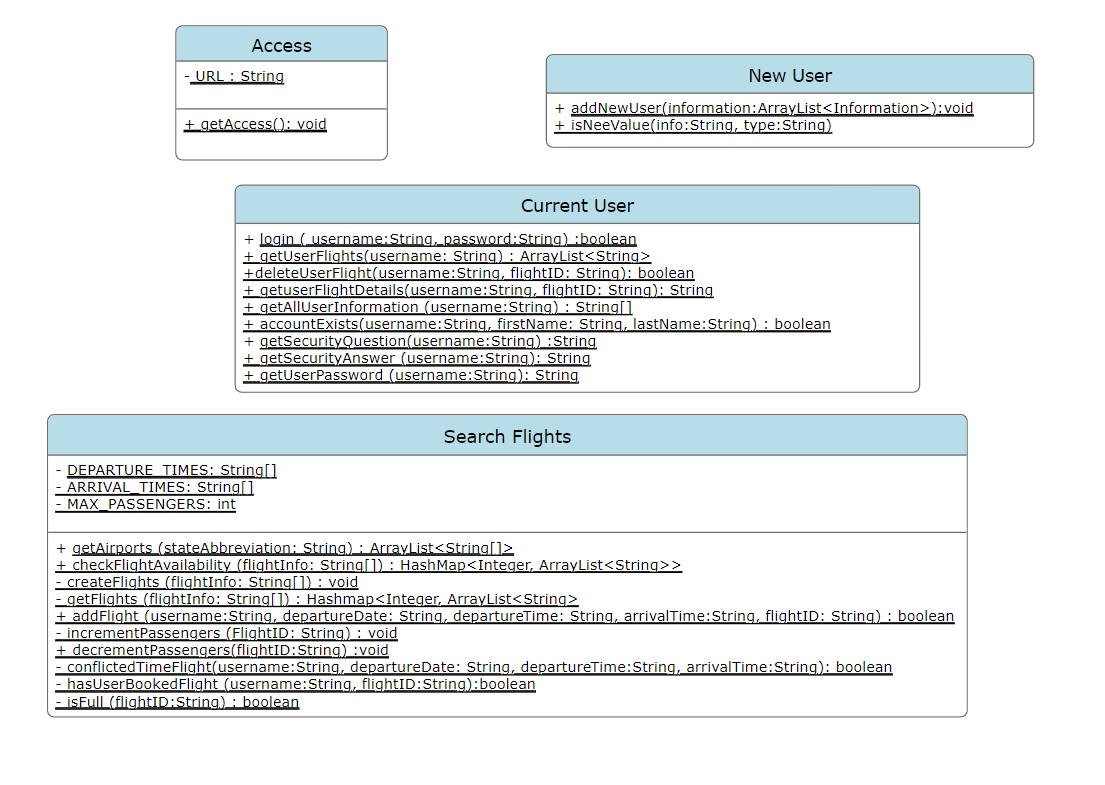
## **Specifications of the Project**

Per the functionality requirements as defined by Dr. Aria, our application has these features:

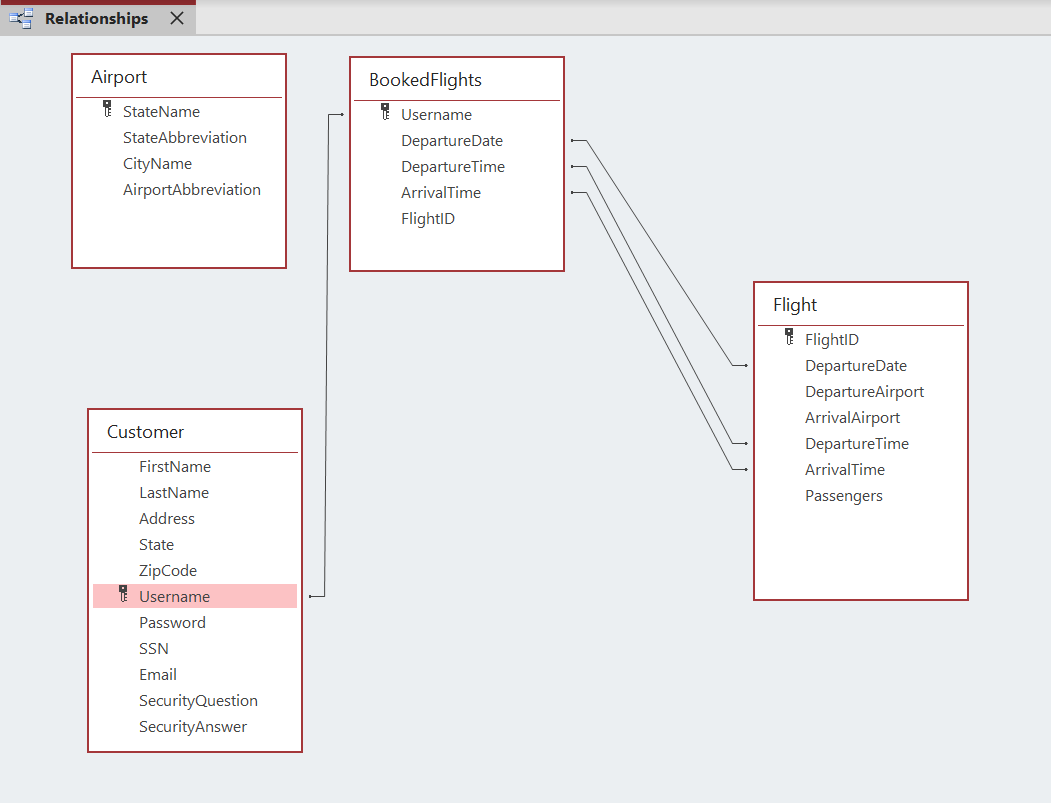
1. Upon start of the application, user should see a splash screen
2. Main menu gives the option to the user to register or login
3. Login requires username and password
4. Registration requires information such as first name, last name, address, zip code, state, username, password, email, SSN, and a security question for password recovery
5. User can login using username and password (should register first if not a member)
6. Enable user to retrieve his password knowing the username with the security question
7. After login into application, a customer should be able to:
   1. Search flight database for flights based on different criteria such a from/to city, date, and time of flight
   2. Be able to book a flight and add that to his account
   3. Be able to delete a flight from his account
   4. Customer should not be able to book same flight more than once
   5. If there is a conflict about date and time of a flight, application should warn the customer about this and should not let user book the conflicting flight
   6. Application should keep track of number of passengers booked at each flight and should not let a user book the flight if it is full
8. User should be able to logout and upon login, previously booked reservations should be there in his/her account
9. You should always provide an option in any case to go back to the Main Menu

# UML Class Diagrams





# Data Model



* The one with the key label is the primary key
* The others with lines means foreign key

# Functionality

## **Main Functionality of the Application**

### Splash Screen

* Upon starting the application, a splash screen is momentarily displayed to the user for approximately 5 seconds before it switches to the main menu.

### Main Menu

#### Before A User Is Logged In

* After the splash screen is displayed or if the user logs out of their account, a scene that prompts the user to either login or register is displayed.
* The user of the application is not allowed to book a flight prior to logging in / registering for an account.
* This ensures that booked flights will always correspond to a specific user.

#### After A User Is Logged In

* After logging in / registering for an account, the main menu contains additional buttons on the top right consisting of:
  + Book Flight
  + My Trips
  + Logout

#### An option to go back to the main menu in any case

* In every scene the user is able to access, a button/hyperlink for them to go back to the main menu is provided.

### Login Page

* Upon switching to the login page, the user is prompted to enter their username and password credentials. To hide the password in the event people around them are looking at their screen, their password is hidden from them. If the credentials does not match any credentials in the database, a message is shown to the user indicating the information is either incorrect or does not exist in the database
* In any case, the user is able to go to the registration page and the main menu page.

### Forgot Password Page

* In the event the user does not remember their password, there is an option for them to go to the Forgot Password page. It will prompt the user to enter some account information before displaying their security question they selected during registration. Lastly, they must enter the correct answer to their security question before their password is displayed to them.

### User Registration

For a user to successfully create an account, these requirements must be met for each of the entry fields

* First Name
  + Must consist of at least 3 alphabet letters
* Last Name
  + Must consist of at least 3 alphabet letters
* Address
  + Must contain at least 1 digit to represent a building number
  + Must contain at least 3 alphabet letters to represent a street name
* State
  + Must select a state represented by abbreviations from the list of states available
* Zip Code
  + Must contain only 5 digits to represent a valid zip code
* Username
  + Must contain at least 3 letters and may contain zero or more numbers
  + Username must not already exist in the database
* Password
  + Must contain at least 5 letters and 1 number
* SSN
  + Must contain only 9 numbers
  + User is able to enter their SSN with or without dashes (-)
  + SSN must not already exist in the database
* Email

[ 1 ]@[ 2 ].com

* + Section 1 must contain at least 5 letters or numbers
  + Section 2 must contain at least 4 letters or numbers
  + Overall, the entry must also contain @ and .com to represent a valid email
* Security Question
  + User is able to select 1 out of the 5 different security questions
* Security Answer
  + User must enter enter an answer of any length and characters to be a valid response

In the event that any entry field is not valid or left blank, the user will be informed of such. They will be required to fix their entry(s) before successfully registering for an account.

After the user is registered, it will automatically sign them in and will go back to the main menu.

### User Searches For Flights Based on Specified Criteria

Upon selecting the Book Flights hyperlink on the main menu, the user will be directed to a different page where they will be able to select and choose these criterias. Similar to the registration page, all of the requirements must be met before the user is able to search for a flight:

* From
  + A state must be selected
  + Upon choosing a specific state, the user will have access to choosing a specific city they are traveling from
* To
  + After a user selects the state they are traveling from, they will be able to select the state they are going to
  + Depending on the state they are traveling from, the user will not be able to book that same state as their final destination
  + Upon choosing a specific state, the use will have access to choosing a specific city they wish to travel to
* Type of Trip
  + User must either select a Round or One-Way trip
  + By default, it automatically selects the Round trip
* Departure Date
  + Regardless of whether the user selected a Round or One-Way trip, they must select a departure date
  + By default, today’s date is selected as the departure date
  + The user will not be able to select dates that are before today’s date
* Return Date
  + IF THE USER SELECTS ROUND TRIP, the ability to choose a return date is made available to the user. Otherwise, it will be hidden to the user
  + Depending on the departure date that was previously selected, the user will not be able to select a date that is before the departure date

In any event, if the user tries to search for flights before all of the necessary fields are completed, they will be warned and prevented from looking at the available flights.

Once the user has the necessary fields completed and they press on the Search Flights button, they will be redirected to a page containing the available flights.

### User Books a Flight After 3 Criteria is Checked

After successfully entering criteria from the search flights page, they will be redirected to a page containing 4 flights that differ in their departure and arrival time.

If the user selects the add button for any of the 4 flights, 3 different criteria will be checked before it is successfully booked:

1. Flight must not already be booked by the same user
2. Flight must not have conflicting time and date based on their currently booked flights
3. Flight must not be full

If the previously mentioned criteria is fulfilled, the booked flight will be successfully added to the user’s account. An alert box will be displayed to inform the user the booking was successful.

If any of the previously mentioned criteria is not fulfilled, the application will warn the user and the flight will not be added to the user’s account.

Lastly, once a flight is successfully booked, the screen will switch depending on the trip type they selected

Round

* User will be prompted to book a return flight
* After successfully booking a return flight, the user will be redirected to their My Trips page

One-Way

* User will be redirected to their My Trips page

### My Trips Page

From the main menu, a logged in user will be able to access all of their current trips from the My Trip link. A list containing all of the flights the user booked will be displayed. A user will be able to select one and either

* Delete the Flight
* Request Additional Information About the Flight

### Upon User Logging Out, Previously Booked Flights Should Be In Their Account

In the event the user wishes to logout or have another user login to their respective account, the ability to logout will be on the main menu. Since we are utilizing Azure as our RDBMS, the booked flights will remain in the database even after the user logs out or closes the application.

## **Initializing the Airport Database**

If anyone is curious, a separate package with a class/file named “AirportTable” is included in the source folder. This class is not used when executing the main application, but it is run just once to initialize the information in the Airport table regarding all of the different airports in each state.

After connecting to the database

1. A CSV file containing all of the airports and their state, city, and airport abbreviation is imported using the Scanner class
2. A while loop is used to iterate through the CSV file as long as it continues to have information
3. In the while loop, it will grab each comma separated value and insert it into the database

NOTE: This file only has to run once and does not need to be run again UNLESS the airport table is deleted, truncated, or has missing values. In the event it has missing values, it is advised to truncate the entire table before running the file again to avoid errors regarding duplicate entries.