AST6 Airline Online Service

**Team 6**

Kunda Wu

Thong Le

John McGinley

**Instructor**

Dr. Mike Wu

## Project Overview

Our objective is to develop a web-based application for stakeholders who have recently founded an airline named Airline Services Team-6 (or AST6) and are in need of a website that can provide a booking service to their customers. Because they are new in the market, they want their website to be released as fast as possible, and they are giving us 3 months to accomplish their expectations. Needless to say, they consider efficiency, security, reliability and scalability of the utmost importance. A small list of the features that they find absolutely essential:

* Client:

1. UI is responsive, simple but modern.
2. Console interface for users to manage their flight, special offers and flight history
3. Search functionality that allows users to sort their options by price/layovers/etc

* Server:

1. Fast and accurate response and handling logic operation based on user’s input.
2. Connect client-server workflow to database management system

* Database:

1. Scalable: not specific
2. Secure: user’s information
3. Reliable: data should be available all time
4. Consists of a collection of data related to flights (costs, number of seats, etc)

## 

## System Environment

* 3-tier Architecture diagram:

Client

(ReactJS)

HTTP request

query

AWS (Cloud)

return data

HTTP response

Business layer

(express.js)

Data access layer (mysql

-connector)

Database

(MySQL)

Server (2-layer)

* Hardware & Software:

1. Apache2
2. Visual Studio Code (latest)
3. OpenVPN

* RDBMS: MySQL community server (v8.0.17)
* Host: Home server
* Frameworks & its core dependencies

1. Node.js (v11.10.1), npm (v6.7.0)
2. Front-end: React (v3.0.1), Redux (latest)
3. Back-end: express.js (v4.16.1) and mysql (connector, v2.17.1)

## Functional Requirements

1. **Users and how users access your system**

There is only one type of user which is customer/ client to our airline services. All the flight information and processes will be pre-set and simulated.

\*\* Users must sign up to start using our services.

* 1. Sign-Up

Users will be prompted to enter some information:

* + - First name, middle initial and last name
    - Gender
    - Date of Birth (DOB)
    - Email. This is important because users will have to use email to sign in our services, to get email confirmation (see 2-factor authentication, page 5), or to reset their password
    - Password (8 characters, 1 special character)
  1. Sign-In
  + Users can sign in using registered email and password
  + If users set 2-factor authentication, they will have to enter a confirmation code sent to their registered email
  1. Reset Password
* The function invokes when users click the ‘Forgot password’ button.
* It will then ask the user to provide the email to receive reset-password link.

1. **Functionality/features, functional processes and I/O(s)**

User Dashboard: When users sign in, they will have access to dashboard where they can search for flights, reserve flights, change their profile and as well as access to flight management

* + - 1. Search for flights:
* Users can pick departure and destination, pick depart time and arrival time
* Users can refine search options such as special offers applied, day/ night time, small/ big plane, extra services
* Users can prioritize flight fares - sort the given list of potential flights by the following criteria:

i. Based on price range (Allow users to set min and max price)

ii. Based on price sort (lowest to highest or vice versa)

ii. Based on duration/number of stops

iii. Based on flight classes (First Class, Economy Plus, or Economy)

* + - 1. Reserve flight: After users complete flight search and ready to pick up flight line
* Users can start to reserve it by entering more personal information such as passport and credit card information. Then, users will be prompted to confirm or cancel the reservation

I. Confirm: Users will receive reservation receipt via registered email

II. Cancel: Reservation will be cancelled

* Finally, users will be redirected back to state of searching for a flight
  + - 1. Profile edit:

Users can update their password, set their preferences:

* + - Saved personal information (used to book flight): passport
    - 2-factor authentication: more secure for users’ account, using email confirmation whenever they sign in
    - Preferred Flight Class: next time they search, their flight class will be set as their default, they can change it anytime
    - Services: preferred services users want to be provided along their flights
    - Alert: alert users about the status of their next flight
      1. In flight management: users can view information about their flights booked with our airline.
* The information can be time, status (delay, ontime, postpone).
* Users can use this feature to cancel (Cannot be done before 24 hours prior to departure)
* Users can add more services to their flight (upgrade their class)
  + - 1. Emergency check in:
* Users will be directed to page displayed with the most early next flights
* Users can pick and fill up passport and credit card information to get in that flight

## Non-functional Issues

1. **Graphical User Interface (GUI)**
   1. Web browser
   2. Mobile device’s browser
   3. GUI is built with React, HTML5 and CSS3
2. **Security and miscellaneous**
   1. Authorization:

The security of users will be protected by email address and password. Enrolled user​ ​account​ ​information​ ​will​ ​be​ ​securely​ ​stored​ ​in​ the database.​ ​​​In​ ​order​ ​to​ ​login​ ​to the​ ​system​, the ​user​ ​must​ ​provide​ ​an​ ​existing​ ​email​ ​account​ ​along​ ​with​ ​the associated​ ​password.​ Meanwhile, if a user happens to forget password, the user have to reset his password before signing in to the system. ​The​ database system ​will​ ​be​ ​implemented​ ​using​ ​local host ​providing​ connection​ ​to​ ​server.

* 1. Secure sign-in: two-factor authentication (a confirmation code will be sent to user’s registered email)
  2. User’s information is encrypted (simple encryption) before persisting into the database system
  3. Simulated flight information

Each flight will have a maximum number of seats and each person can not double-book the same flight. When the user enters a search query, a series of desired flights will be displayed, with the number of seats left. If the number of seats is zero, that flight will not be shown to the user, even if it meets their other criteria.

* 1. Maintainability and scalability

System will be designed so it will be easy to maintain and scale up when the business grows

Also, it depends on further requirements from the stakeholders

* 1. Reliability of booking/flight information

All the reservation/ flight information will be persisted and secured

* 1. Performance of this booking service (Times, overall experience, and etc)

In this AST6 project, we implement the concurrently popular technology Node.js and React to build the client side and server side.