

Airport Operation Management System

Final Report

Team Members: Azlan Amjad, Saud Agha, Faiz Marsad

Submission Date: April 15th, 2022

I. ABSTRACT

Currently, during the flight booking process, online travel agencies act as the middleman between customers and airlines, this results in customer dissatisfaction due to lack of communication and due to the overall difficulty in the flight booking process. Our goal for this project was to remove this middleman, and provide a way for airports and airlines to directly interact with potential passengers. To meet this goal we used React.js, React Redux, and Materials UI for developing the frontend. For the backend we used Django for the API and PostgresQL for the database. The primary users for the website are the passengers who can browse and purchase both fares for flights, and stay at hotels, and file complaints against the airlines and airport. The airport administrator is in charge of adding companies, hotels, and stays for the passenger to purchase, and resolving any complaints posted by passengers. While the airline admin is in charge of adding flights and fares, and resolving complaints made by passengers against the employing airline. With these features, we hope to satisfy the need of a self-contained website in which passengers, airlines, and the airport can conduct business without an online travel agency acting as mediator.

Index Terms:

Term	Definitions
Online Travel Agency	Web marketplace for travel related services
React.js	Javascript library used for building UI in frontend
React Redux	Javascript library used for central state management
Materials UI	React UI library for faster and easier web design
Django	Python web framework that allows us to create dynamic web app for hosting API endpoints in the backend
API	APIs allow communication between different software components
PostgresQL	Object Relational Database Management System

II. TABLE OF CONTENTS

ABSTRACT	1
TABLE OF CONTENTS	2
INTRODUCTION	3
DESCRIPTION OF THE SYSTEM	4
PROJECT DESIGN	5
Describing End-Users	5
EER Diagram	7
IMPLEMENTATION	8
Relational Model	8
DataBase Management System	9
PROFESSIONAL API DOCUMENTATION	29
USER GUIDE	30

III. INTRODUCTION

We set out to create a self-contained system in which passengers, airline administrators, and airport administrators can conduct business in a fair, respectable manner. To accomplish this objective, we required a database which can store the various user types information, as well as required information to host a platform which allows these users to interact. The purchasing of fares, which is the business between passenger and the airline, and indirectly the airline administrator, required our system to store data regarding destinations that the airline provides flights for, airplanes and the amount of seats that each airplane has, and the pricing for the fares. An additional feature we added was the ability for the passenger to file complaints, and to accomplish this we needed to store the complaint itself, the passenger filing it, and when the time comes, the airline administrator who resolves the said complaint. Between the passenger and the airport, and indirectly the airport administrator, the system required the storing of a stay at a hotel. Thus, our database would need not only store the stay and the hotel, but the company which runs the hotel. Additionally, the passenger has the option to file a complaint against the airport, which also needs to be stored in a similar manner as the complaint against an airline. Finally, after a successful transaction between the passenger and the airline and/or airport, the system must hold the transaction data as well. As such we designed a database to hold these various information to ensure that all three user types can accomplish their required goals with ease.

IV. DESCRIPTION OF THE SYSTEM

Since there are currently no OTAs that allow direct communication between the airline's personnel and the customers, we created a web app that fulfills all the necessary requirements of a travel website as well as allow communication between passengers and airline and airport administrators. We have achieved that by having three different types of users that can interact with the website. Users no longer have to go through different processes and experience longer waiting times to have their problems solved. They can write any complaints explaining their particular issue. The airport and the airline administrators are able to view the complaints from the website and solve their issues quickly. In addition to resolving complaints, both administrators have the power to add items (fares and stays) for their respective employer.

V. PROJECT DESIGN

A. Describing End-Users

There are three different types of users that can interact with the website. Like any traditional OTAs, passengers are able to create their own account and book for flights or reserve a hotel of their liking. In addition to this, they are also able to file complaints and they can convey their dissatisfaction towards both the airport and the airline using the complaints dashboard. Airport admins can take a closer look into any complaints launched by the user regarding the airport and solve it swiftly without any bureaucracy. They are also able to add hotels for the passengers using the website, and stays that the passenger can purchase for the added hotel. The airline admin is responsible for resolving any complaints made by the users related to the airline and thus help to improve the consumer services. They are also responsible for adding different flights from Calgary to other parts of the world that the customers can book. In addition to the three end users of the website, the included Django Superuser acts as a “central” administrator, with the role of maintaining the entire database. The Superuser, which has access through Django's administrator dashboard, can add both airline and airport administrators, as well as manage airplanes and destinations that are approved by the airport. In addition, the super user has the power to remove any information, in order to maintain the database, outside of the website's sphere of influence. Due to the nature of this power, we decided to use Django's already available Superuser to ensure a safe and secure method was made to handle the sensitive data. Password fields are not viewable by the superuser as the raw passwords are not stored.

The detailed accessibility of the users are explained below :

Passengers are able to :

- Create a user account by filling up the fields in the sign up page
- Log into the website via the Login page by using the exact username and the password that was used to create an account
- View and add flight fares on their desired day and month to the cart.
- Remove flights from the cart.
- Are able to file complaints regarding the airport and the airline from their profile page.
- Can check if the filed complaints are resolved.
- Are able to view and add a hotel stay of their liking to the cart
- Checkout any items added to the cart

Airport Admins can :

- Add Companies that own hotels.
- Add hotels that have stays.
- Add stays by specifying stay name, description, and price to an added hotel.
- Can resolve any complaints made by the consumer directed towards the airport through the profile page.
- They can login via the specific admin username and password and don't have to sign up for an account. Accounts are added by a central administrator.

Airline Admins can :

- Add flights from the flights page
- Can resolve any complaints made by the consumer directed towards the airline through the profile page.
- They can login via the specific admin username and password and don't have to sign up for an account

Django Superuser can :

- Add, modify, and remove any and all data types.
- The username and password is created via Django's CLI, thus the airport will have authority over the superuser.
- Login into via the Django admin dashboard.
- Maintain the database to ensure day-to-day functions run smoothly.

Note: in further section, we will include a more in-depth rundown of different users.

B. EER Diagram

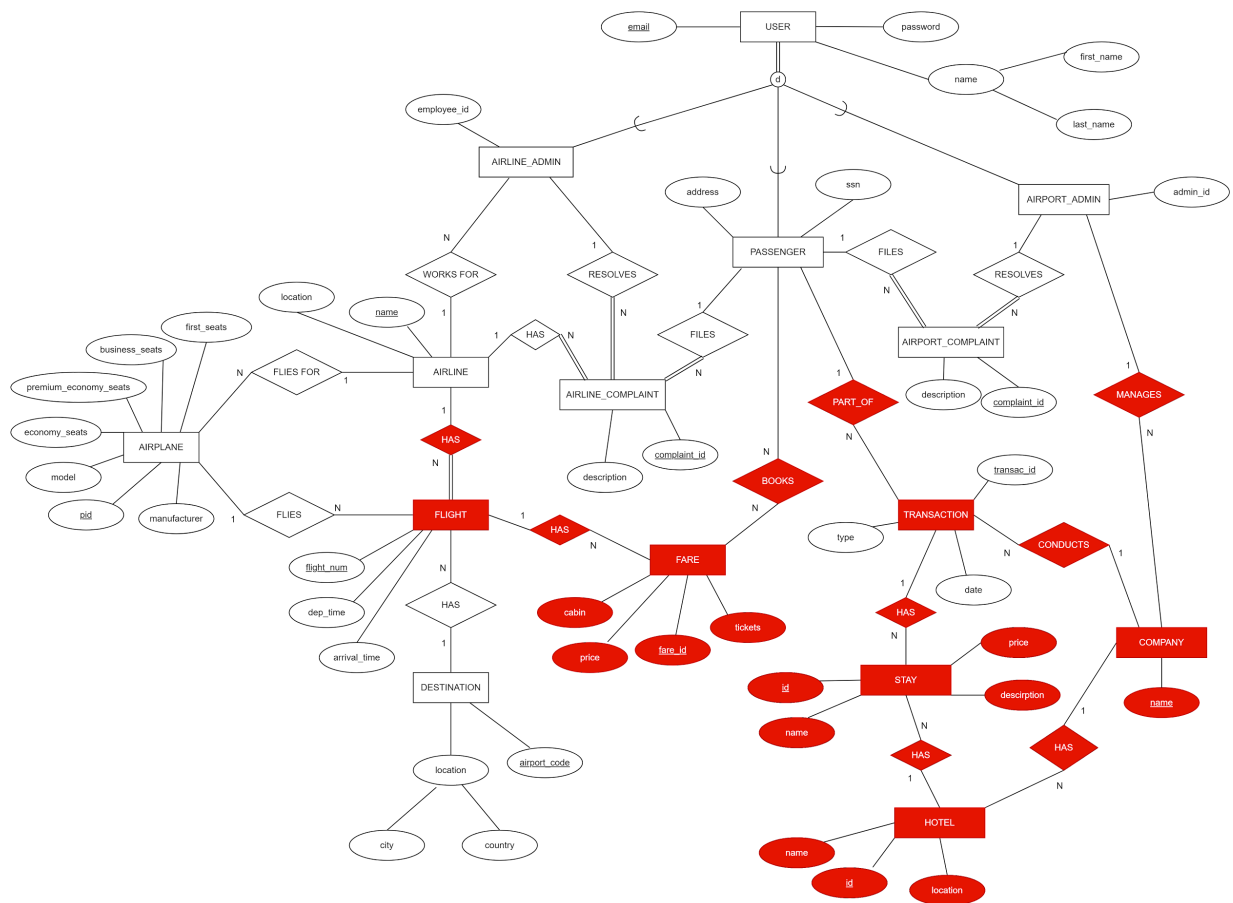


Figure 2.0, Extended Entity Relationship Diagram (Updated)

The changes in our EER diagram are indicated by the red. Compared to the initially submitted one, we decided to make Flight into a strong entity, due to the nature of Django's API setup which required each entity to have an individual primary key. In addition, we removed the "ticket" entity completely, and replaced it with the "fare" entity, as the fare can have multiple tickets which the passenger can book. This was preferable than having a single ticket which the passenger can book. Lastly, we removed the "store" and "item" entities from the initial diagram. This is due to us realizing that a "store" does not make sense in practical terms of a feature an Airport should offer potential passengers. We replaced these entities with "stay", "hotel", and "company" entities. These entities represent passengers having the option to "purchase" a stay at a hotel, which is run by a company. For maintenance, the airport administrator is in charge of adding companies, hotels, and stays which the passenger can purchase.

VI. IMPLEMENTATION

A. Relational Model

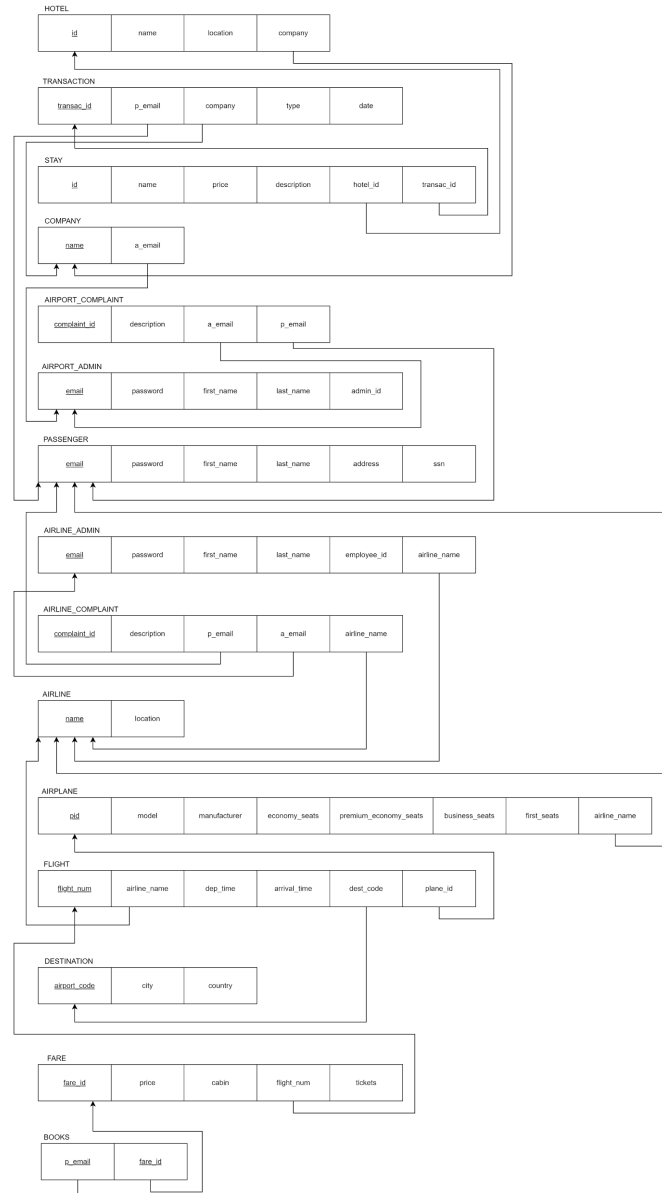


Figure 5.0, Relational Model

This relational model was generated using the above mentioned EER. One change we made was adding a separate “books” entity, which would hold the information between a passenger and a fare they purchase. We deemed it necessary to add this entity as it will help the database better track which tickets are bought by which passengers.

B. DataBase Management System

For the DataBase Management System we selected PostgreSQL. Before selecting the DBMS, we first wanted to find a suitable API framework with extended support and documentation. We decided upon Django; a python based web framework, which is both free and open source. Further researching Django, we came to find that SQLite and PostgreSQL work best with Django. However, the downside of SQLite is that it is more aimed towards local data storage for individual applications and devices [1]. Due to the nature of our web application, we needed a DBMS that emphasized scalability and centralization. Thus, we decided upon PostgreSQL, which has good support from the Django community.

Implemented SQL Statements:

(Note: these SQL statements are generated using logging via Django)

To simplify this section, we will list the SQL Query statements in the web pages they occur.

Furthermore, we will exclude the Django Superuser query statements and focus on three primary end-users (Passenger, Airline Administrator, Airport Administrator).

The following pages will generally be used by the Passenger end-user:

Sign Up Page:

```
SELECT (1) AS "a" FROM "user" WHERE "user"."email" = 'superman@man.com' LIMIT 1;
args=('superman@man.com,'); alias=default
```

```
SELECT (1) AS "a" FROM "passenger" WHERE "passenger"."ssn" = '403123123' LIMIT 1;
args=('403123123,'); alias=default
```

```
INSERT INTO "user" ("password", "last_login", "is_superuser", "first_name", "last_name",
"is_staff", "is_active", "date_joined", "email", "a_passenger", "an_airport_admin",
"an_airline_admin") VALUES
('pbkdf2_sha256$320000$tWk9ZKmeE8fv42t8bOOzsj$NZwc8/I9E0JRQ7PEMdm7Dgi4nr+xfy1ql2/2h3rRV14=', NULL, false, 'Saud', 'Agha', false, true,
'2022-04-16T03:36:55.661109+00:00':timestamp, 'superman@man.com', false, false, false)
RETURNING "user"."id";
args=('pbkdf2_sha256$320000$tWk9ZKmeE8fv42t8bOOzsj$NZwc8/I9E0JRQ7PEMdm7Dgi4nr+xfy1ql2/2h3rRV14=', None, False, 'Saud', 'Agha', False, True, datetime.datetime(2022, 4, 16, 3, 36, 55, 661109, tzinfo=datetime.timezone.utc), 'superman@man.com', False, False, False);
alias=default
```

```
UPDATE "user" SET "password" =
'pbkdf2_sha256$320000$tWk9ZKmeE8fv42t8bOOzsj$NZwc8/I9E0JRQ7PEMdm7Dgi4nr+xfy1
```

```
ql2/2h3rRVl4=', "last_login" = NULL, "is_superuser" = false, "first_name" = 'Saud', "last_name"
= 'Agha', "is_staff" = false, "is_active" = true, "date_joined" =
'2022-04-16T03:36:55.661109+00:00':timestampz, "email" = 'superman@man.com',
"a_passenger" = true, "an_airport_admin" = false, "an_airline_admin" = false WHERE
"user"."id" = 6;
args=('pbkdf2_sha256$320000$tWk9ZKmeE8fv42t8bOOzsj$NZwc8/I9E0JRQ7PEMdm7Dgi4n
r+xfy1ql2/2h3rRVl4=', False, 'Saud', 'Agha', False, True, datetime.datetime(2022, 4, 16, 3, 36,
55, 661109, tzinfo=datetime.timezone.utc), 'superman@man.com', True, False, False, 6);
alias=default
```

```
INSERT INTO "passenger" ("user_id", "ssn", "address") VALUES (6, '403123123', '123 Street')
RETURNING "passenger"."id"; args=(6, '403123123', '123 Street'); alias=default
```

Login Page:

```
INSERT INTO "token_blacklist_outstandingtoken" ("user_id", "jti", "token", "created_at",
"expires_at") VALUES (6, '5d06dd4fcf4e4dd4b737263d95684a26',
'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t8b90eXBIIjoicmVmcmVzaCI6MTY1MDE2Njc3NSwiaWF0IjoxNjUwMDgwMzc1LCJqdGkiOiI1ZDA2ZGQ0ZmNmNGU0ZGQ0YjczNzI2M2Q5NTY4NGEyNiIsInVzZXJfaWQiOiJZ9.ZDjCdGFXFX3dIxa2S6SsWxYoLb2nOTy1BSpuInZPCUrM', '2022-04-16T03:39:35.217484+00:00':timestampz,
'2022-04-17T03:39:35+00:00':timestampz) RETURNING
"token_blacklist_outstandingtoken"."id"; args=(6, '5d06dd4fcf4e4dd4b737263d95684a26',
'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t8b90eXBIIjoicmVmcmVzaCI6MTY1MDE2Njc3NSwiaWF0IjoxNjUwMDgwMzc1LCJqdGkiOiI1ZDA2ZGQ0ZmNmNGU0ZGQ0YjczNzI2M2Q5NTY4NGEyNiIsInVzZXJfaWQiOiJZ9.ZDjCdGFXFX3dIxa2S6SsWxYoLb2nOTy1BSpuInZPCUrM', datetime.datetime(2022, 4, 16, 3, 39, 35, 217484,
tzinfo=datetime.timezone.utc), datetime.datetime(2022, 4, 17, 3, 39, 35,
tzinfo=datetime.timezone.utc)); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "auth_group"."id", "auth_group"."name" FROM "auth_group" INNER JOIN
"user_groups" ON ("auth_group"."id" = "user_groups"."group_id") WHERE
"user_groups"."user_id" = 6; args=(6,); alias=default
```

```
SELECT "auth_permission"."id", "auth_permission"."name",
"auth_permission"."content_type_id", "auth_permission"."codename" FROM "auth_permission"

INNER JOIN "user_user_permissions" ON ("auth_permission"."id" =
"user_user_permissions"."permission_id") INNER JOIN "django_content_type" ON
("auth_permission"."content_type_id" = "django_content_type"."id") WHERE
"user_user_permissions"."user_id" = 6 ORDER BY "django_content_type"."app_label" ASC,
"django_content_type"."model" ASC, "auth_permission"."codename" ASC; args=(6,);
alias=default
```

Home page:

```
SELECT "destination"."airport_code", "destination"."city", "destination"."country" FROM
"destination"; args=(); alias=default
```

```
SELECT "destination"."airport_code", "destination"."city", "destination"."country" FROM
"destination"; args=(); alias=default
```

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE ("flight"."destination_id" =
'YYZ' AND ("flight"."dep_time" AT TIME ZONE 'UTC')::date = '2022-04-16'::date);
args=('YYZ', datetime.date(2022, 4, 16)); alias=default
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "destination"."airport_code", "destination"."city", "destination"."country" FROM
"destination" WHERE "destination"."airport_code" = 'YYZ' LIMIT 21; args=('YYZ,');
alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "destination"."airport_code", "destination"."city", "destination"."country" FROM
"destination" WHERE "destination"."airport_code" = 'YYZ' LIMIT 21; args=('YYZ,');
alias=default
```

Flight Search Results page:

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE "flight"."id" = 1 LIMIT
21; args=(1,); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "destination"."airport_code", "destination"."city", "destination"."country" FROM
"destination" WHERE "destination"."airport_code" = 'YYZ' LIMIT 21; args=('YYZ,');
alias=default
```

```
SELECT "fare"."id", "fare"."price", "fare"."cabin", "fare"."flight_id", "fare"."tickets_quantity"
FROM "fare" WHERE ("fare"."flight_id" = 1 AND NOT ("fare"."tickets_quantity" = 0));
args=(1, 0); alias=default
```

Reservation Companies page:

```
SELECT "company"."id", "company"."name", "company"."admin_id" FROM "company";
args=(); alias=default
```

Reservation Hotels page:

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."company_id" = 1;
args=(1,); alias=default
```

```
SELECT "company"."id", "company"."name", "company"."admin_id" FROM "company"
WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "company"."id", "company"."name", "company"."admin_id" FROM "company"
WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default
```

Reservation Stays page:

```
SELECT "stay"."id", "stay"."name", "stay"."price", "stay"."description", "stay"."hotel_id",
"stay"."transaction_id", "stay"."image", "stay"."thumbnail" FROM "stay" WHERE
("stay"."hotel_id" = 1 AND "stay"."transaction_id" IS NULL); args=(1,); alias=default
```

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."id" = 1 LIMIT 21;
args=(1,); alias=default
```

```
SELECT "company"."id", "company"."name", "company"."admin_id" FROM "company"
WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."id" = 1 LIMIT 21;
args=(1,); alias=default
(0.004) SELECT "company"."id", "company"."name", "company"."admin_id" FROM
"company" WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default
```

Passenger Account Page:

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "auth_group"."id", "auth_group"."name" FROM "auth_group" INNER JOIN
"user_groups" ON ("auth_group"."id" = "user_groups"."group_id") WHERE
"user_groups"."user_id" = 6; args=(6,); alias=default
```

```
SELECT "auth_permission"."id", "auth_permission"."name",
"auth_permission"."content_type_id", "auth_permission"."codename" FROM "auth_permission"
INNER JOIN "user_user_permissions" ON ("auth_permission"."id" =
"user_user_permissions"."permission_id") INNER JOIN "django_content_type" ON
("auth_permission"."content_type_id" = "django_content_type"."id") WHERE
```

```
"user_user_permissions"."user_id" = 6 ORDER BY "django_content_type"."app_label" ASC,
"django_content_type"."model" ASC, "auth_permission"."codename" ASC; args=(6,);
alias=default
```

Complaints page:

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline"; args=();
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
SELECT "airport_complaint"."id", "airport_complaint"."description",
"airport_complaint"."admin_id", "airport_complaint"."passenger_id" FROM "airport_complaint"
WHERE "airport_complaint"."passenger_id" = 3; args=(3,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
(0.005) SELECT "airline_complaint"."id", "airline_complaint"."description",
"airline_complaint"."admin_id", "airline_complaint"."passenger_id",
"airline_complaint"."airline_id" FROM "airline_complaint" WHERE
"airline_complaint"."passenger_id" = 3; args=(3,); alias=default
```

Adding a Complaint and the page refreshing with added complaint:

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
INSERT INTO "airport_complaint" ("description", "admin_id", "passenger_id") VALUES
('washrooms stink.', NULL, 3) RETURNING "airport_complaint"."id"; args=('washrooms stink.',
None, 3); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline"; args=();
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
SELECT "airport_complaint"."id", "airport_complaint"."description",
"airport_complaint"."admin_id", "airport_complaint"."passenger_id" FROM "airport_complaint"
WHERE "airport_complaint"."passenger_id" = 3; args=(3,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline"; args=();
alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "airport_complaint"."id", "airport_complaint"."description",
"airport_complaint"."admin_id", "airport_complaint"."passenger_id" FROM "airport_complaint"
WHERE "airport_complaint"."passenger_id" = 3; args=(3,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
```



```
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
SELECT "airline_complaint"."id", "airline_complaint"."description",
"airline_complaint"."admin_id", "airline_complaint"."passenger_id",
"airline_complaint"."airline_id" FROM "airline_complaint" WHERE
"airline_complaint"."passenger_id" = 3; args=(3,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 6 LIMIT 21; args=(6,); alias=default
```

```
SELECT "airline_complaint"."id", "airline_complaint"."description",
"airline_complaint"."admin_id", "airline_complaint"."passenger_id",
"airline_complaint"."airline_id" FROM "airline_complaint" WHERE
"airline_complaint"."passenger_id" = 3; args=(3,); alias=default
```

Checking out items:

```
SELECT (1) AS "a" FROM "books" WHERE ("books"."fare_id" = 9 AND
"books"."passenger_id" = 1) LIMIT 1; args=(9, 1); alias=default
```

```
SELECT "fare"."id", "fare"."price", "fare"."cabin", "fare"."flight_id", "fare"."tickets_quantity"
FROM "fare" WHERE "fare"."id" = 9 LIMIT 21; args=(9,); alias=default
```

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE "flight"."id" = 1 LIMIT
21; args=(1,); alias=default
(0.002) UPDATE "fare" SET "price" = 20.00, "cabin" = 'Economy', "flight_id" = 1,
"tickets_quantity" = 18 WHERE "fare"."id" = 9; args=(Decimal('20.00'), 'Economy', 1, 18, 9);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "fare"."id", "fare"."price", "fare"."cabin", "fare"."flight_id", "fare"."tickets_quantity"
FROM "fare" WHERE "fare"."id" = 9 LIMIT 21; args=(9,); alias=default
```

```
INSERT INTO "books" ("passenger_id", "fare_id") VALUES (1, 9) RETURNING "books"."id";
args=(1, 9); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
(0.001) SELECT "company"."id", "company"."name", "company"."admin_id" FROM
"company" WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
INSERT INTO "transaction" ("passenger_id", "company_id", "type", "date") VALUES (1, 1,
'reservation', '2022-04-16'::date) RETURNING "transaction"."id"; args=(1, 1, 'reservation',
datetime.date(2022, 4, 16)); alias=default
```

```
[16/Apr/2022 04:35:57] "POST /api/v1/transaction/ HTTP/1.1" 201 75
id list created
```

```
SELECT "stay"."id", "stay"."name", "stay"."price", "stay"."description", "stay"."hotel_id",
"stay"."transaction_id", "stay"."image", "stay"."thumbnail" FROM "stay" WHERE
"stay"."name" IN ('3 Bed 1 Bath') LIMIT 21; args=('3 Bed 1 Bath,'); alias=default
```

```
SELECT "stay"."id", "stay"."name", "stay"."price", "stay"."description", "stay"."hotel_id",
"stay"."transaction_id", "stay"."image", "stay"."thumbnail" FROM "stay" WHERE
"stay"."name" IN ('3 Bed 1 Bath'); args=('3 Bed 1 Bath,'); alias=default
```

```
SELECT "stay"."id", "stay"."name", "stay"."price", "stay"."description", "stay"."hotel_id",
"stay"."transaction_id", "stay"."image", "stay"."thumbnail" FROM "stay" WHERE
("stay"."name" IN ('3 Bed 1 Bath') AND "stay"."id" = 3) LIMIT 21; args=('3 Bed 1 Bath', 3);
alias=default
```

```
SELECT (1) AS "a" FROM "stay" WHERE ("stay"."name" = '3 Bed 1 Bath' AND NOT
("stay"."id" = 3)) LIMIT 1; args=('3 Bed 1 Bath', 3); alias=default
```

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."id" = 1 LIMIT 21;
args=(1,); alias=default
```

```
SELECT "transaction"."id", "transaction"."passenger_id", "transaction"."company_id",
"transaction"."type", "transaction"."date" FROM "transaction" WHERE "transaction"."id" = 5
LIMIT 21; args=(5,); alias=default
```

```
UPDATE "stay" SET "name" = (CASE WHEN ("stay"."id" = 3) THEN '3 Bed 1 Bath' ELSE
NULL END)::varchar(255), "price" = (CASE WHEN ("stay"."id" = 3) THEN 50.00 ELSE
NULL END)::numeric(6, 2), "description" = (CASE WHEN ("stay"."id" = 3) THEN 'Nice
Views!' ELSE NULL END)::text, "hotel_id" = (CASE WHEN ("stay"."id" = 3) THEN 1 ELSE
NULL END)::bigint, "transaction_id" = (CASE WHEN ("stay"."id" = 3) THEN 5 ELSE NULL
END)::bigint WHERE "stay"."id" IN (3); args=(3, '3 Bed 1 Bath', 3, Decimal('50.00'), 3, 'Nice
Views!', 3, 1, 3, 5, 3); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."user_id" = 4 LIMIT 21; args=(4,); alias=default
```

Logging out (Universal for end-users):

```
SELECT (1) AS "a" FROM "token_blacklist_blacklistedtoken" INNER JOIN
"token_blacklist_outstandingtoken" ON ("token_blacklist_blacklistedtoken"."token_id" =
"token_blacklist_outstandingtoken"."id") WHERE "token_blacklist_outstandingtoken"."jti" =
'317939607e50455c91213eb5f6afe237' LIMIT 1;
args=('317939607e50455c91213eb5f6afe237'); alias=default
```

```
SELECT "token_blacklist_outstandingtoken"."id",
"token_blacklist_outstandingtoken"."user_id", "token_blacklist_outstandingtoken"."jti",
"token_blacklist_outstandingtoken"."token", "token_blacklist_outstandingtoken"."created_at",
"token_blacklist_outstandingtoken"."expires_at" FROM "token_blacklist_outstandingtoken"
WHERE "token_blacklist_outstandingtoken"."jti" = '317939607e50455c91213eb5f6afe237'
LIMIT 21; args=('317939607e50455c91213eb5f6afe237'); alias=default
```

```
SELECT "token_blacklist_blacklistedtoken"."id",
"token_blacklist_blacklistedtoken"."token_id",
"token_blacklist_blacklistedtoken"."blacklisted_at" FROM "token_blacklist_blacklistedtoken"
WHERE "token_blacklist_blacklistedtoken"."token_id" = 11 LIMIT 21; args=(11); alias=default
```

```
INSERT INTO "token_blacklist_blacklistedtoken" ("token_id", "blacklisted_at") VALUES (11,
'2022-04-16T04:05:43.981432+00:00':timestampz) RETURNING
"token_blacklist_blacklistedtoken"."id"; args=(11, datetime.datetime(2022, 4, 16, 4, 5, 43,
981432, tzinfo=datetime.timezone.utc)); alias=default
```

The following pages will generally be used by the airport administrator:

Airport admin dashboard:

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 3 LIMIT 21; args=(3);
alias=default
```

```
(0.002) SELECT "auth_group"."id", "auth_group"."name" FROM "auth_group" INNER JOIN
"user_groups" ON ("auth_group"."id" = "user_groups"."group_id") WHERE
"user_groups"."user_id" = 3; args=(3,); alias=default
```

```
SELECT "auth_permission"."id", "auth_permission"."name",
"auth_permission"."content_type_id", "auth_permission"."codename" FROM "auth_permission"
INNER JOIN "user_user_permissions" ON ("auth_permission"."id" =
"user_user_permissions"."permission_id") INNER JOIN "django_content_type" ON
("auth_permission"."content_type_id" = "django_content_type"."id") WHERE
"user_user_permissions"."user_id" = 3 ORDER BY "django_content_type"."app_label" ASC,
"django_content_type"."model" ASC, "auth_permission"."codename" ASC; args=(3,);
alias=default
```

To reduce repetitiveness, as the airport-administrator shares pages with the passenger, we have excluded the same SQL queries and have added the queries pertaining to the administrator itself.

Airport Admin adding a Company:

```
SELECT (1) AS "a" FROM "company" WHERE "company"."name" = 'Marriott International'
LIMIT 1; args=('Marriott International,'); alias=default
```

```
INSERT INTO "company" ("name", "admin_id") VALUES ('Marriott International', NULL)
RETURNING "company"."id"; args=('Marriott International', None); alias=default
```

```
SELECT "company"."id", "company"."name", "company"."admin_id" FROM "company";
args=(); alias=default (This last query is to load the page with the added company)
```

Airport Admin adding a Hotel:

```
SELECT (1) AS "a" FROM "hotel" WHERE "hotel"."name" = 'Best Western Central' LIMIT 1;
args=('Best Western Central,'); alias=default
```

```
SELECT "company"."id", "company"."name", "company"."admin_id" FROM "company"
WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
INSERT INTO "hotel" ("name", "location", "company_id", "image", "thumbnail") VALUES
('Best Western Central', 'Toronto', 1, "", "") RETURNING "hotel"."id"; args=('Best Western
Central', 'Toronto', 1, "", ""); alias=default
```

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."company_id" = 1;
args=(1,); alias=default
```

```
SELECT "company"."id", "company"."name", "company"."admin_id" FROM "company"
WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default (This query is repeated N times,
where N is the number of hotels for the company)
```

Airport Admin adding a Stay:

```
SELECT "stay"."id", "stay"."name", "stay"."price", "stay"."description", "stay"."hotel_id",
"stay"."transaction_id", "stay"."image", "stay"."thumbnail" FROM "stay" WHERE
("stay"."hotel_id" = 1 AND "stay"."transaction_id" IS NULL); args=(1,); alias=default
```

```
SELECT (1) AS "a" FROM "stay" WHERE "stay"."name" = '2 Bed 4 Baths' LIMIT 1; args=('2
Bed 4 Baths'); alias=default
```

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."id" = 1 LIMIT 21;
args=(1,); alias=default
```

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."id" = 1 LIMIT 21;
args=(1,); alias=default
```

```
INSERT INTO "stay" ("name", "price", "description", "hotel_id", "transaction_id", "image",
"thumbnail") VALUES ('2 Bed 4 Baths', 250.00, 'Good place to stay!!!', 1, NULL, "", "")
RETURNING "stay"."id"; args=('2 Bed 4 Baths', Decimal('250.00'), 'Good place to stay!!!', 1,
None, "", ""); alias=default
```

```
SELECT "stay"."id", "stay"."name", "stay"."price", "stay"."description", "stay"."hotel_id",
"stay"."transaction_id", "stay"."image", "stay"."thumbnail" FROM "stay" WHERE
("stay"."hotel_id" = 1 AND "stay"."transaction_id" IS NULL); args=(1,); alias=default
```

```
SELECT "hotel"."id", "hotel"."name", "hotel"."location", "hotel"."company_id",
"hotel"."image", "hotel"."thumbnail" FROM "hotel" WHERE "hotel"."id" = 1 LIMIT 21;
args=(1,); alias=default
(0.001) SELECT "company"."id", "company"."name", "company"."admin_id" FROM
"company" WHERE "company"."id" = 1 LIMIT 21; args=(1,); alias=default
```

Airport Admin Resolving a Complaint: :

```
SELECT "airport_complaint"."id", "airport_complaint"."description",
"airport_complaint"."admin_id", "airport_complaint"."passenger_id" FROM
"airport_complaint"; args=(); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 4 LIMIT 21; args=(4,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 4 LIMIT 21; args=(4,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "airport_admin"."id", "airport_admin"."user_id", "airport_admin"."admin_id" FROM
"airport_admin" WHERE "airport_admin"."user_id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "airport_complaint"."id", "airport_complaint"."description",
"airport_complaint"."admin_id", "airport_complaint"."passenger_id" FROM "airport_complaint"
WHERE "airport_complaint"."id" = 2 LIMIT 21; args=(2,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "airport_admin"."id", "airport_admin"."user_id", "airport_admin"."admin_id" FROM
"airport_admin" WHERE "airport_admin"."id" = 1 LIMIT 21; args=(1,); alias=default
UPDATE "airport_complaint" SET "description" = 'Food issue', "admin_id" = 1, "passenger_id"
= 1 WHERE "airport_complaint"."id" = 2; args=('Food issue', 1, 1, 2); alias=default
```

```
SELECT "airport_complaint"."id", "airport_complaint"."description",
"airport_complaint"."admin_id", "airport_complaint"."passenger_id" FROM
"airport_complaint"; args=(); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 4 LIMIT 21; args=(4,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```



```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 4 LIMIT 21; args=(4,);
alias=default
```

```
SELECT "airport_admin"."id", "airport_admin"."user_id", "airport_admin"."admin_id" FROM
"airport_admin" WHERE "airport_admin"."user_id" = 3 LIMIT 21; args=(3,); alias=default
```

Lastly, we will cover the Airline-administrators SQL queries for each functionality. To reduce repetitiveness, login and logout will be excluded. Only the functionalities in the dashboard are presented here.

Adding a flight and fare:

```
SELECT "airline_admin"."id", "airline_admin"."user_id", "airline_admin"."employee_id",
"airline_admin"."airline_id" FROM "airline_admin" WHERE "airline_admin"."user_id" = 2
LIMIT 21; args=(2,); alias=default
```

```
(0.001) SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "destination"."airport_code", "destination"."city", "destination"."country" FROM
"destination"; args=(); alias=default
```

```
SELECT "airplane"."id", "airplane"."model", "airplane"."manufacturer",
"airplane"."economy_seats", "airplane"."premium_economy_seats", "airplane"."business_seats",
"airplane"."first_seats", "airplane"."airline_id" FROM "airplane" WHERE "airplane"."airline_id"
= 1; args=(1,); alias=default
```

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE "flight"."airline_id" = 1;
args=(1,); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "destination"."airport_code", "destination"."city", "destination"."country" FROM
"destination" WHERE "destination"."airport_code" = 'YYZ' LIMIT 21; args=('YYZ,');
alias=default
```

```
SELECT "airplane"."id", "airplane"."model", "airplane"."manufacturer",
"airplane"."economy_seats", "airplane"."premium_economy_seats", "airplane"."business_seats",
"airplane"."first_seats", "airplane"."airline_id" FROM "airplane" WHERE "airplane"."id" = 1
LIMIT 21; args=(1,); alias=default
```

```
INSERT INTO "flight" ("airline_id", "dep_time", "arrival_time", "destination_id", "plane_id")
VALUES (1, '2022-04-16T04:43:27.216000+00:00':timestampz,
'2022-04-16T04:43:27.216000+00:00':timestampz, 'YYZ', 1) RETURNING "flight"."id";
args=(1, datetime.datetime(2022, 4, 16, 4, 43, 27, 216000,
tzinfo=zoneinfo.ZoneInfo(key='UTC')), datetime.datetime(2022, 4, 16, 4, 43, 27, 216000,
tzinfo=zoneinfo.ZoneInfo(key='UTC')), 'YYZ', 1); alias=default
```

```
SELECT "airplane"."id", "airplane"."model", "airplane"."manufacturer",
"airplane"."economy_seats", "airplane"."premium_economy_seats", "airplane"."business_seats",
"airplane"."first_seats", "airplane"."airline_id" FROM "airplane" WHERE "airplane"."id" = 1;
args=(1,); alias=default
```

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE "flight"."id" = 4 LIMIT
21; args=(4,); alias=default
```

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE "flight"."id" = 4 LIMIT
21; args=(4,); alias=default
```

```
INSERT INTO "fare" ("price", "cabin", "flight_id", "tickets_quantity") VALUES (250.00,
'Economy', 4, 30) RETURNING "fare"."id"; args=(Decimal('250.00'), 'Economy', 4, 30);
alias=default
```

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE "flight"."id" = 4 LIMIT
21; args=(4,); alias=default
```

```
INSERT INTO "fare" ("price", "cabin", "flight_id", "tickets_quantity") VALUES (400.00,
'Premium Economy', 4, 30) RETURNING "fare"."id"; args=(Decimal('400.00'), 'Premium
Economy', 4, 30); alias=default
```

```
SELECT "flight"."id", "flight"."airline_id", "flight"."dep_time", "flight"."arrival_time",
"flight"."destination_id", "flight"."plane_id" FROM "flight" WHERE "flight"."id" = 4 LIMIT
21; args=(4,); alias=default
```

```
INSERT INTO "fare" ("price", "cabin", "flight_id", "tickets_quantity") VALUES (600.00, 'First',
4, 30) RETURNING "fare"."id"; args=(Decimal('600.00'), 'First', 4, 30); alias=default
```

```
INSERT INTO "fare" ("price", "cabin", "flight_id", "tickets_quantity") VALUES (500.00,
'Business', 4, 30) RETURNING "fare"."id"; args=(Decimal('500.00'), 'Business', 4, 30);
alias=default
```

```
SELECT "airline_admin"."id", "airline_admin"."user_id", "airline_admin"."employee_id",
"airline_admin"."airline_id" FROM "airline_admin" WHERE "airline_admin"."user_id" = 2
LIMIT 21; args=(2,); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

Airline administrator resolving a complaint:

```
SELECT "airline_admin"."id", "airline_admin"."user_id", "airline_admin"."employee_id",
"airline_admin"."airline_id" FROM "airline_admin" WHERE "airline_admin"."user_id" = 2
LIMIT 21; args=(2,); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "airline_complaint"."id", "airline_complaint"."description",
"airline_complaint"."admin_id", "airline_complaint"."passenger_id",
"airline_complaint"."airline_id" FROM "airline_complaint" WHERE
"airline_complaint"."airline_id" = 1; args=(1,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 4 LIMIT 21; args=(4,);
alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
(0.006) SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "airline_complaint"."id", "airline_complaint"."description",
"airline_complaint"."admin_id", "airline_complaint"."passenger_id",
"airline_complaint"."airline_id" FROM "airline_complaint" WHERE "airline_complaint"."id" =
2 LIMIT 21; args=(2,); alias=default
```

```
SELECT "airline_admin"."id", "airline_admin"."user_id", "airline_admin"."employee_id",
"airline_admin"."airline_id" FROM "airline_admin" WHERE "airline_admin"."id" = 1 LIMIT
21; args=(1,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
UPDATE "airline_complaint" SET "description" = 'Bathroom issue', "admin_id" = 1,
"passenger_id" = 1, "airline_id" = 1 WHERE "airline_complaint"."id" = 2; args=('Bathroom
issue', 1, 1, 1, 2); alias=default
```

```
SELECT "airline_complaint"."id", "airline_complaint"."description",
"airline_complaint"."admin_id", "airline_complaint"."passenger_id",
"airline_complaint"."airline_id" FROM "airline_complaint" WHERE
"airline_complaint"."airline_id" = 1; args=(1,); alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 3 LIMIT 21; args=(3,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 6 LIMIT 21; args=(6,);
alias=default
```

```
SELECT "passenger"."id", "passenger"."user_id", "passenger"."ssn", "passenger"."address"
FROM "passenger" WHERE "passenger"."id" = 1 LIMIT 21; args=(1,); alias=default
```

```
SELECT "user"."id", "user"."password", "user"."last_login", "user"."is_superuser",
"user"."first_name", "user"."last_name", "user"."is_staff", "user"."is_active",
"user"."date_joined", "user"."email", "user"."a_passenger", "user"."an_airport_admin",
"user"."an_airline_admin" FROM "user" WHERE "user"."id" = 4 LIMIT 21; args=(4,);
alias=default
```

```
(0.001) SELECT "airline"."id", "airline"."name", "airline"."location" FROM "airline" WHERE
"airline"."id" = 1 LIMIT 21; args=(1,); alias=default
```

VII. PROFESSIONAL API DOCUMENTATION

The professional API documentation is available on Postman:

<https://documenter.getpostman.com/view/20119315/Uyr5nK6a>

Due to the nature of this web application, further iterations might result in addition or removal of API endpoints. We attempted to get the most commonly used endpoints documented. This documentation is subject to change.

VIII. USER GUIDE

In this section, we will go over the basics of setting up and using the Airport Operations and Management System for YYC International Airport.

1) Necessary Setup:

- For the full functionality of the Airport Operations and Management System, it is assumed that you have Python 3.9 with Pipenv configured. As well as possess adequate knowledge of Command Line Interfaces.
- The Django backend will be running in a Python Virtual Environment (Pipenv)
- The database will be running using PostgreSQL.
- Prior to starting up the backend, a database instance must be set up with PostgreSQL.
- A sample database has been provided in the Github repository (/backend directory) to help quick start.
- To load the sample run “./manage.py loaddata db.json”.
- The password for all users is “airport123”.

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'airportdb',  
        'USER': 'airport-system-api',  
        'PASSWORD': 'cp5c471',  
        'HOST': '127.0.0.1',  
        'PORT': 5432,  
        'TEST': {},  
    }  
}
```

Please configure the database instance based on this information. This can be modified in settings.py found in the backend folder.

1) Creating the Superuser and starting up Django

```

saud@DESKTOP-A2TLT89:~/School/github/Airport-Operation-Management-Application/backend$ pipenv install
Installing dependencies from Pipfile.lock (202f7f)...
  0/0 - 00:00:00
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.

```

Start by running the “`pipenv install`” command to download any dependencies and create a virtual environment for the backend. Once done, run “`pipenv shell`” to activate the virtual environment.

```

(backend) saud@DESKTOP-A2TLT89:~/School/github/Airport-Operation-Management-Application/backend$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, api, auth, contenttypes, sessions, token_blacklist
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0001_initial... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK

```

Run “`python manage.py migrate`” to set up the database.

```

(backend) saud@DESKTOP-A2TLT89:~/School/github/Airport-Operation-Management-Application/backend$ python manage.py createsuperuser
Email: admin@mail.com
First name: Admin
Last name: Admin
Password:
Password (again):
Superuser created successfully.

```

Run “`python manage.py createsuperuser`” to create the superuser. Fill in the information as requested.

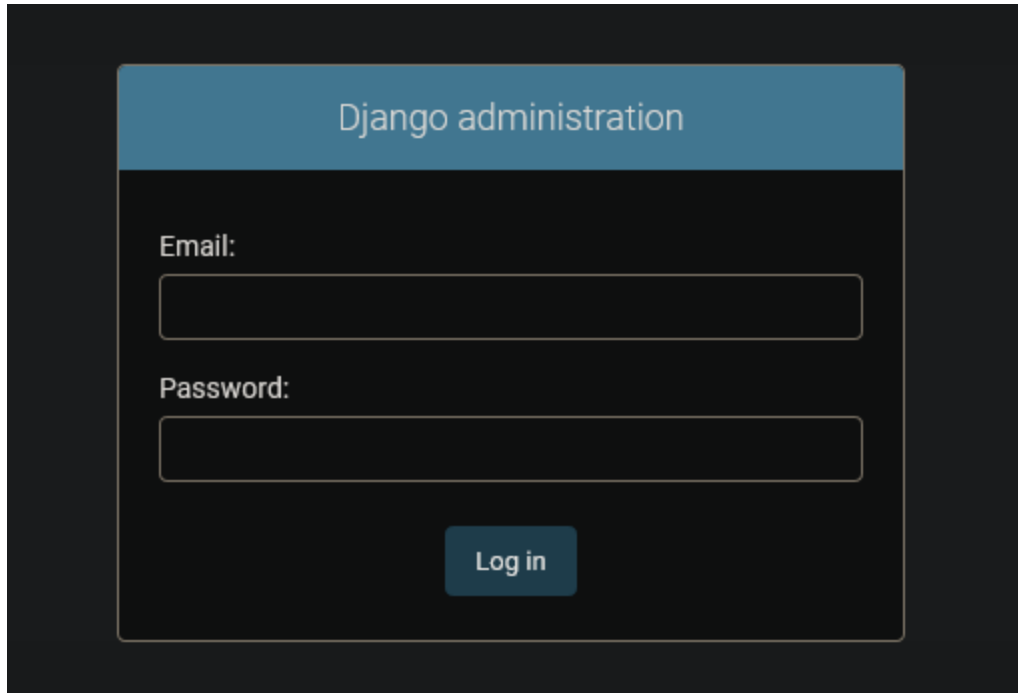
```

(backend) saud@DESKTOP-A2TLT89:~/School/github/Airport-Operation-Management-Application/backend$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 16, 2022 - 00:02:34
Django version 4.0.3, using settings 'backend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

```

Run “`python manage.py runserver`” to start the Django API Server.

A screenshot of the Django administration login interface. It features a dark blue header with the text "Django administration". Below the header, there are two input fields: "Email:" and "Password:". The "Email:" field is a simple text box, while the "Password:" field is a password box with a small eye icon to toggle visibility. Below these fields is a blue "Log in" button. The entire form is set against a dark background.

Django administration

Email:

Password:

Log in

Navigate to “<http://127.0.0.1:8000/admin/>” on web browser of choice. This will prompt a login field. Use the above created Superuser to login.

Site administration

API

Airline admins [+ Add](#) [✎ Change](#)

Airline complaints [+ Add](#) [✎ Change](#)

Airlines [+ Add](#) [✎ Change](#)

Airplanes [+ Add](#) [✎ Change](#)

Airport admins [+ Add](#) [✎ Change](#)

Airport complaints [+ Add](#) [✎ Change](#)

Bookss [+ Add](#) [✎ Change](#)

Companyys [+ Add](#) [✎ Change](#)

Destinations [+ Add](#) [✎ Change](#)

Fares [+ Add](#) [✎ Change](#)

Flights [+ Add](#) [✎ Change](#)

Hotels [+ Add](#) [✎ Change](#)

Passengers [+ Add](#) [✎ Change](#)

Stays [+ Add](#) [✎ Change](#)

Transactions [+ Add](#) [✎ Change](#)

Users [+ Add](#) [✎ Change](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [✎ Change](#)

TOKEN BLACKLIST

Blacklisted tokens [+ Add](#) [✎ Change](#)

Outstanding tokens [✎ Change](#)

Once logged in, the Superuser has access to all the entities of the database.

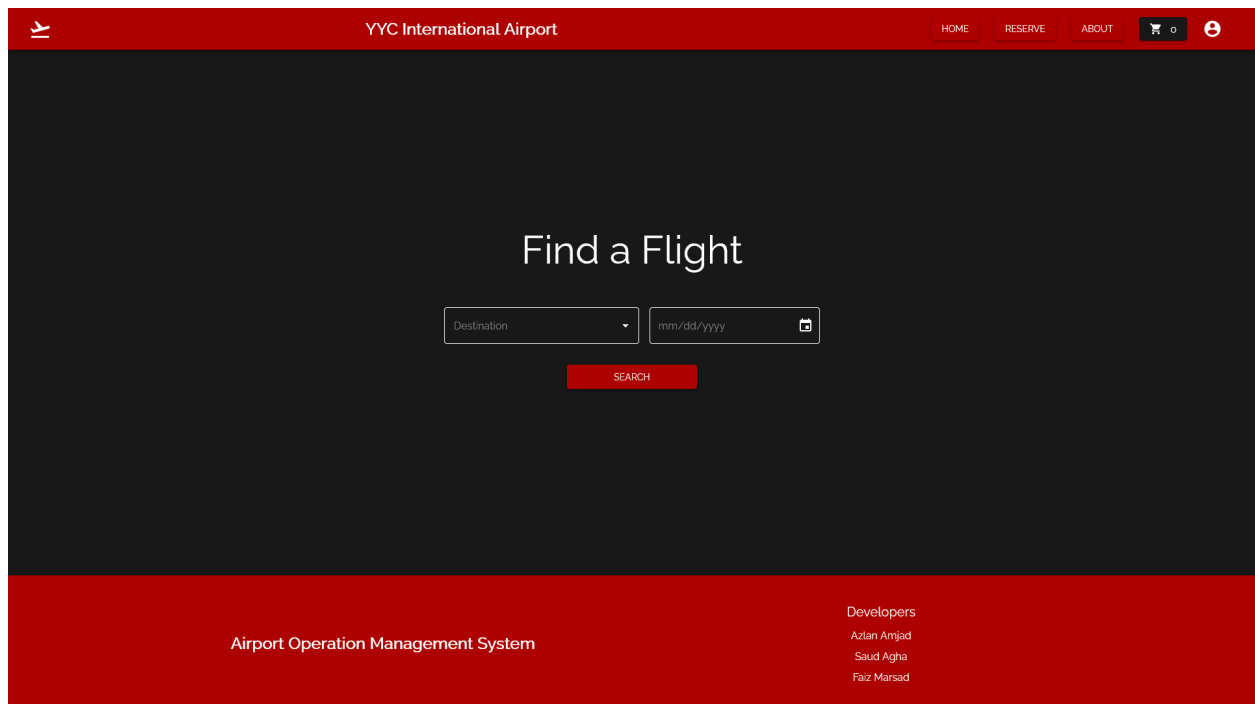
API		
<u>Airline admins</u>	+ Add	Change
Airline complaints	+ Add	Change
<u>Airlines</u>	+ Add	Change
<u>Airplanes</u>	+ Add	Change
<u>Airport admins</u>	+ Add	Change
<u>Airport complaints</u>	+ Add	Change
Bookss	+ Add	Change
Companys	+ Add	Change
<u>Destinations</u>	+ Add	Change
Fares	+ Add	Change
Flights	+ Add	Change
Hotels	+ Add	Change
Passengers	+ Add	Change
Stays	+ Add	Change
Transactions	+ Add	Change
<u>Users</u>	+ Add	Change
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
TOKEN BLACKLIST		
Blacklisted tokens	+ Add	Change
Outstanding tokens		Change

For successful operation, please add the required information in the underlined entities. These should be discussed with Airport higher ups before adding. After these entities are populated, the front end is ready to be started.

2) Starting the Front-end and browsing the web application

```
saud@DESKTOP-A2TLT89:~/School/github/Airport-Operation-Management-Application/frontend$ npm start  
  
> frontend@0.1.0 start  
> react-scripts start
```

Run “npm start” in the frontend directory to start the front end.



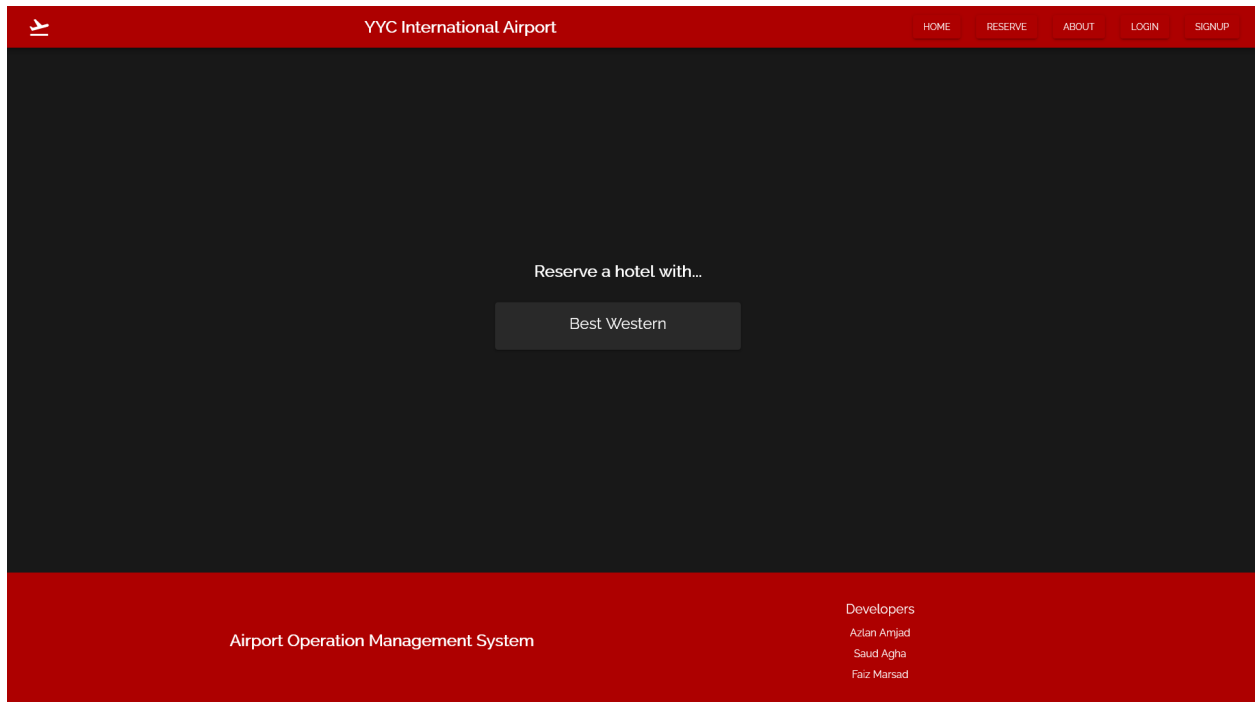
Once started, the homepage will be set up (<http://localhost:3000/>). There is an unofficial 4th user type, which consists of a user that has chosen to not login. Very limited usage of the web application is granted to an unregistered user.

The screenshot shows the YYC International Airport website. The header is red with the airport logo and navigation links: HOME, RESERVE, ABOUT, a shopping cart icon with '0', and a user profile icon. The main content area is dark grey with the heading 'Find a Flight'. Below this is a search form with two input fields: 'Toronto, Canada - YYZ' and '04/16/2022'. A red 'SEARCH' button is positioned below the date field. Under the button, the text 'Search Results' is displayed. A search result card is shown, containing the text: '(April 16, 2022 | 00:24 UTC) - (April 16, 2022 | 00:24 UTC)', 'Calgary (YYC) - Toronto (YYZ)', and 'Air Canada'. A right-pointing arrow is on the right side of the card. The footer is red and contains the text 'Airport Operation Management System' on the left and 'Developers' followed by the names 'Azlan Amjad', 'Saud Agha', and 'Faiz Marsad' on the right.

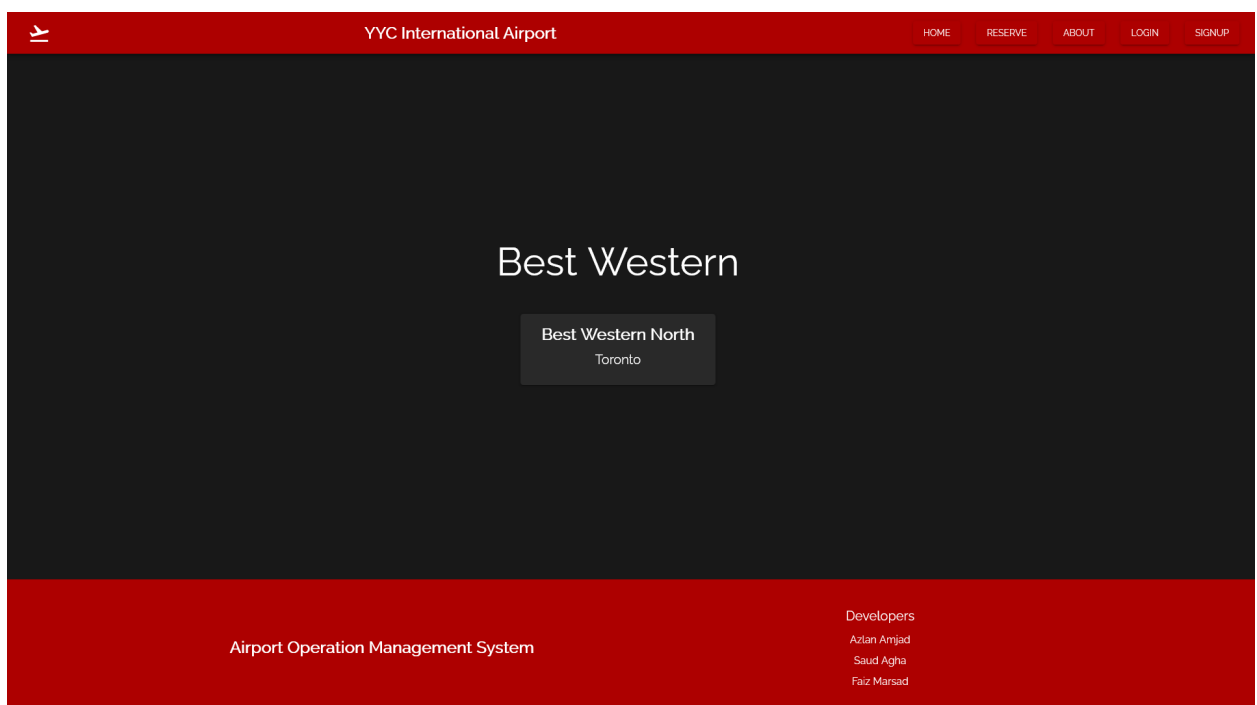
Specifying a search criteria will reveal any flights matching the criteria.

The screenshot shows the YYC International Airport website with the heading 'Calgary to Toronto'. Below the heading, it says 'Air Canada - April 17, 2022' and '00:30 UTC - 00:31 UTC'. The main content area is dark grey with the heading 'Select Fare to Toronto'. Below this heading are four fare options, each in a dark grey box: 'CA \$300.00 Economy Class 30 Tickets Left', 'CA \$1000.00 First Class 30 Tickets Left', 'CA \$700.00 Business Class 30 Tickets Left', and 'CA \$500.00 Premium Economy Class 30 Tickets Left'.

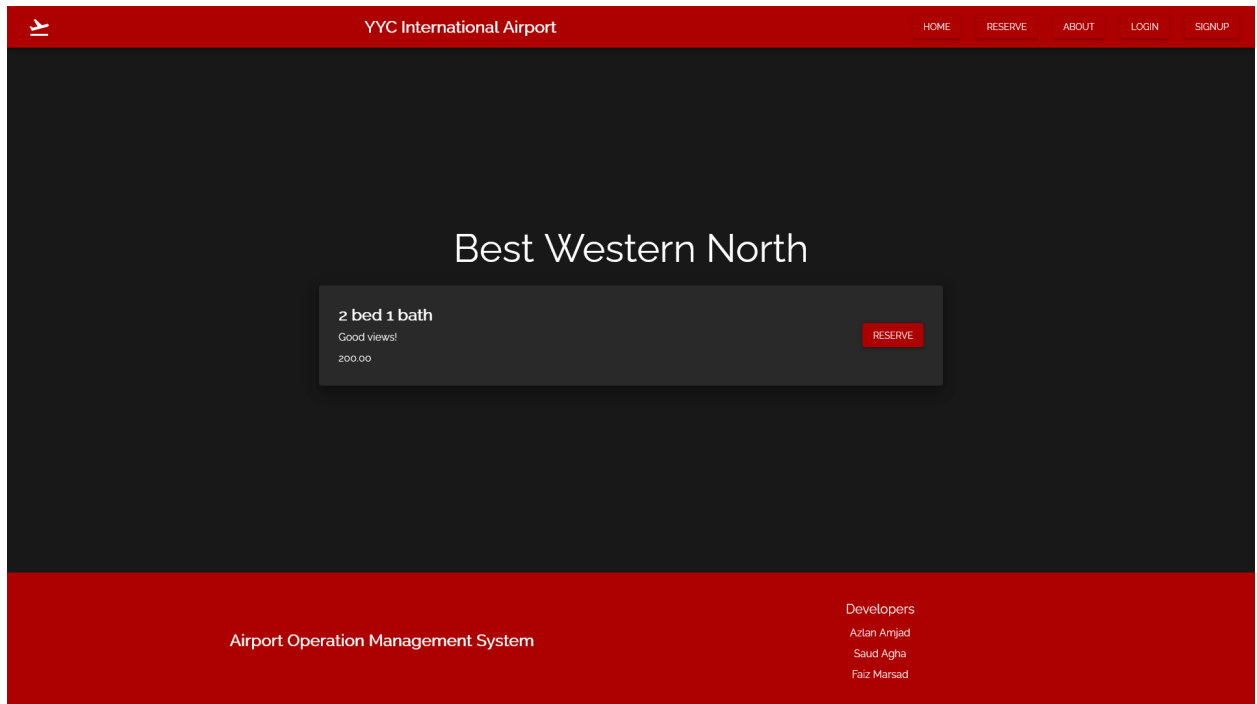
Selecting a flight will display all the fares.



Selecting the reserve button will prompt the user to select a hotel company to reserve with.



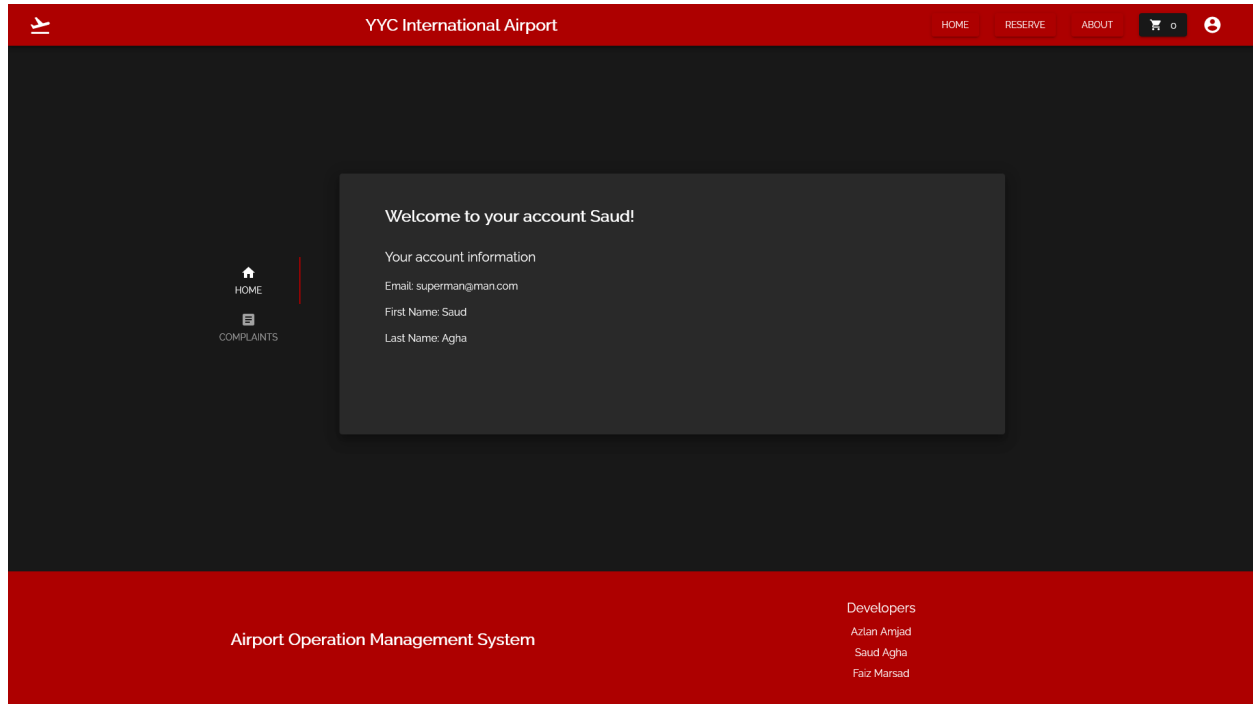
In addition, the user will be asked for the hotel itself.



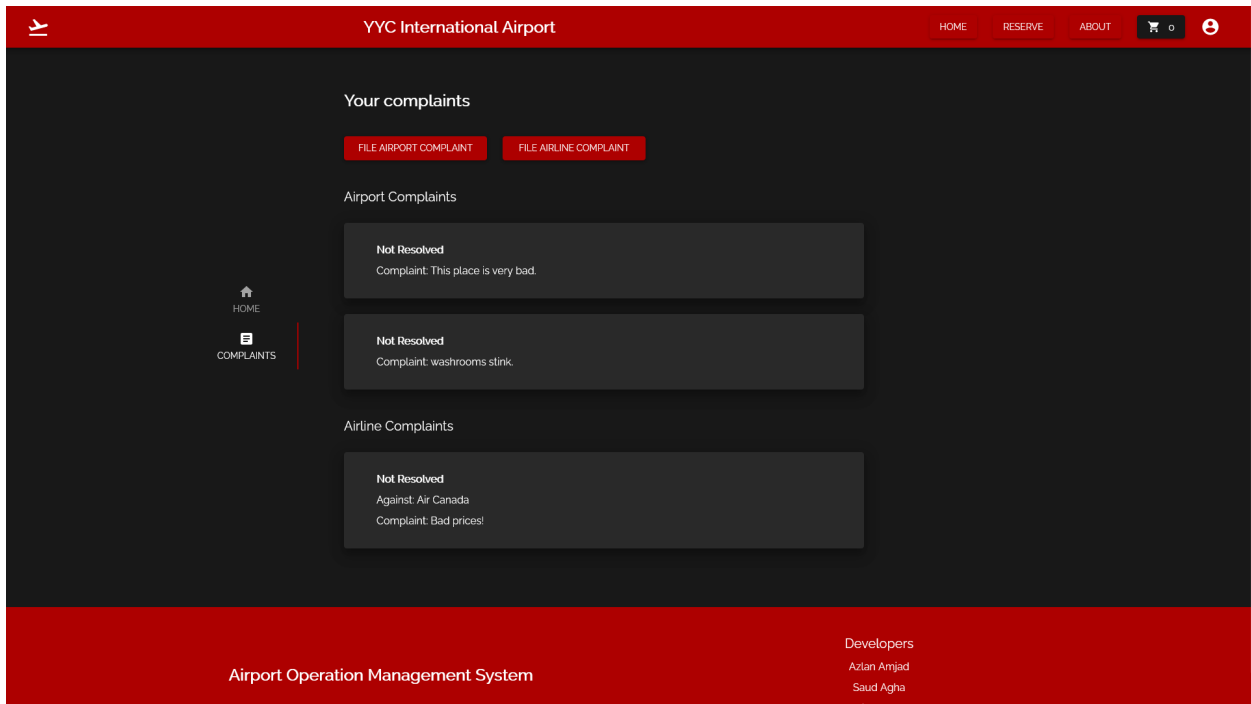
Lastly, any stays that the hotel has will be revealed. Hitting the “Reserve” button will prompt the user to login. Next we will cover the signup and login procedures, as well as the extended functionalities of “Passenger” users.

3) Passenger Functionalities

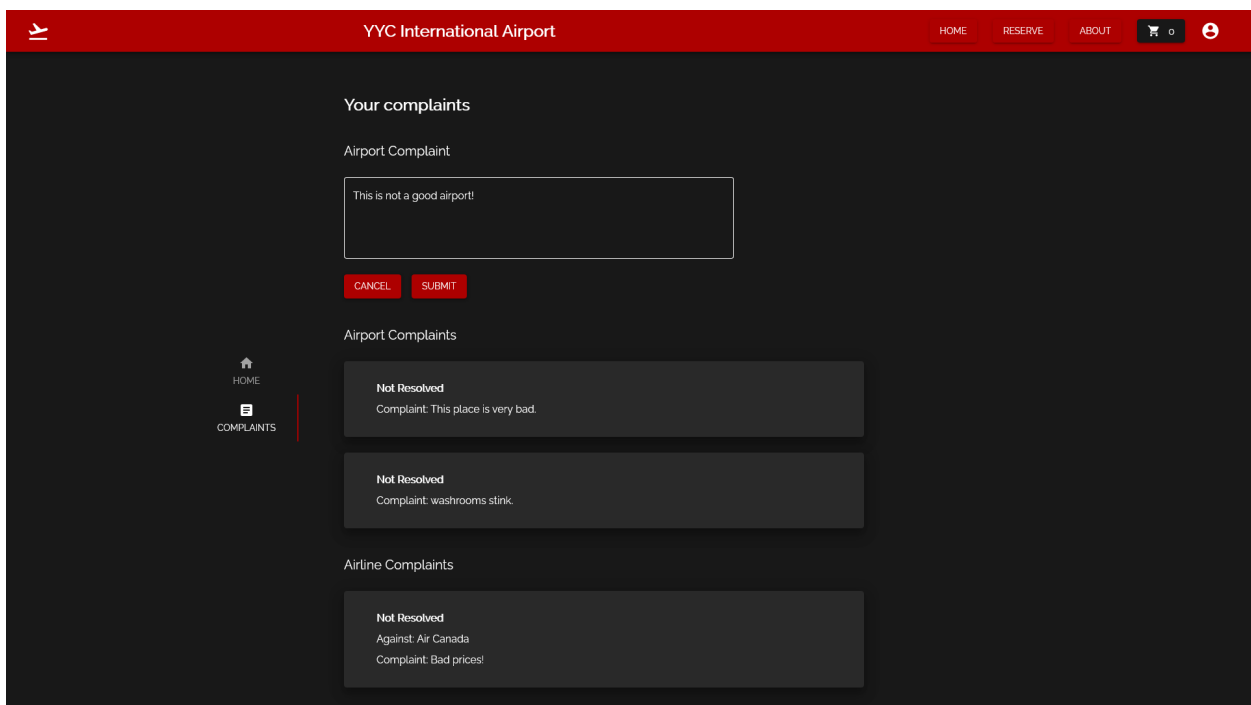
The passenger functionalities are the same as above except some added ones.



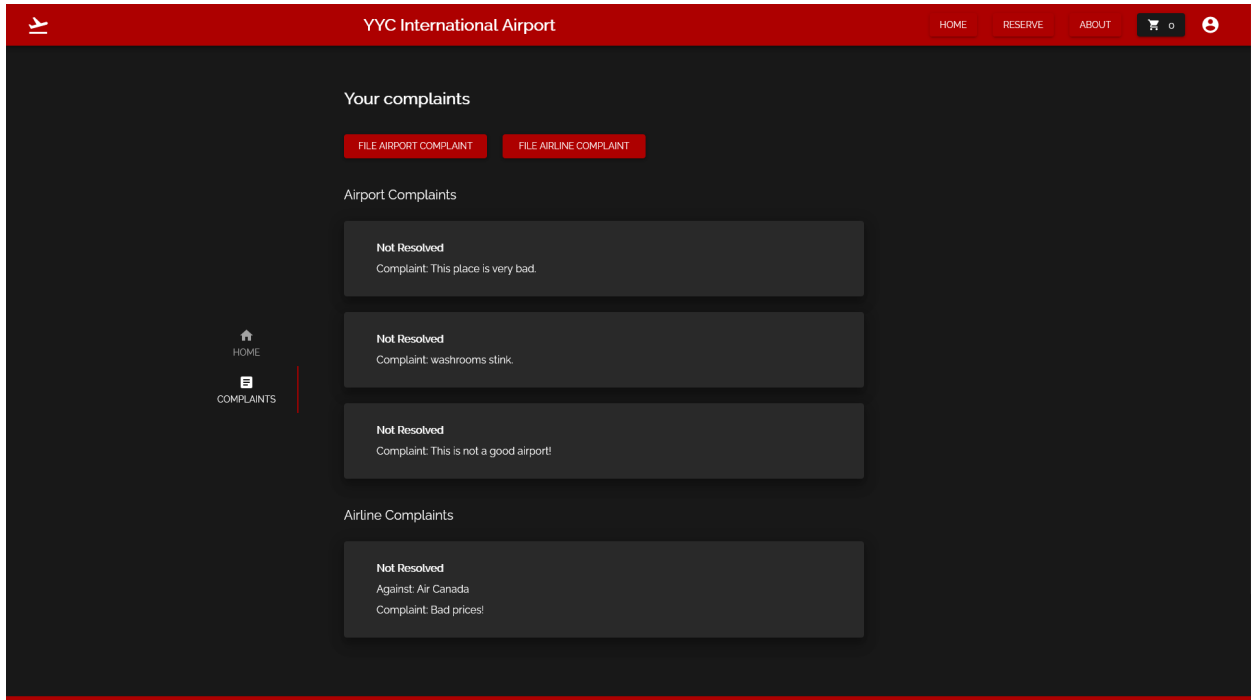
The passenger has an Account page, where they can view account information and file complaints.



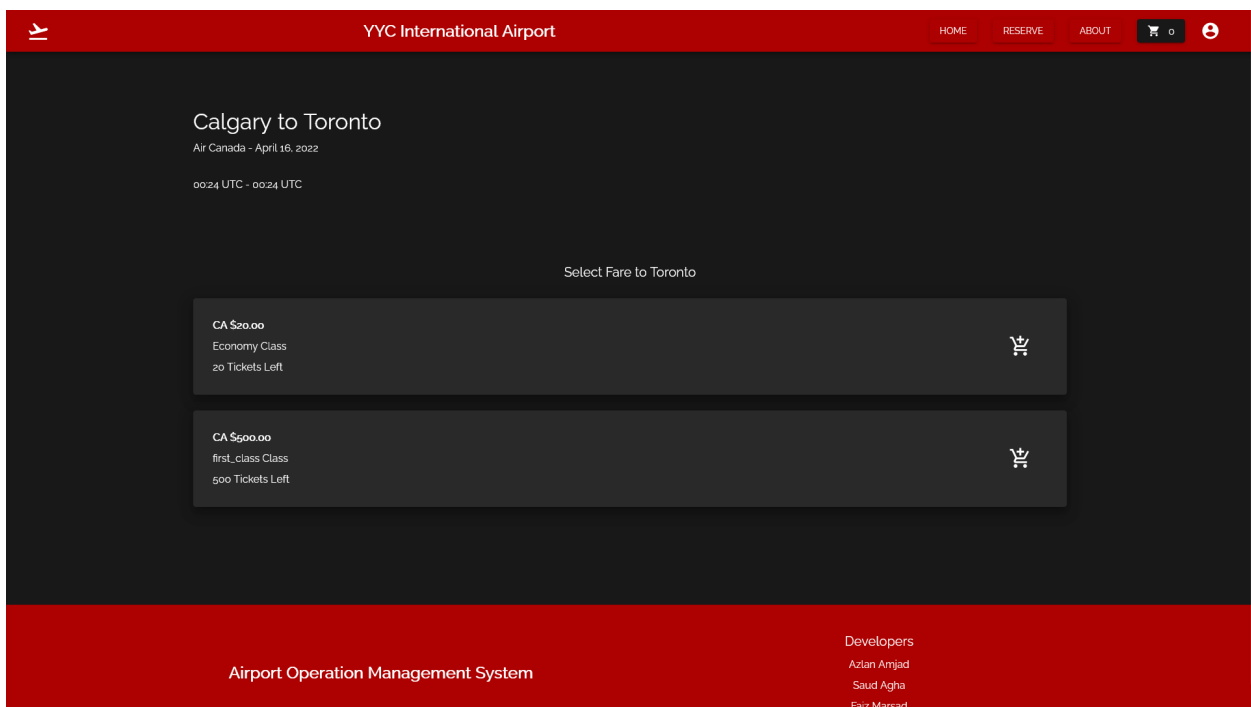
The complaints tab shows all complaints made by the passenger, as well as if they have been resolved.



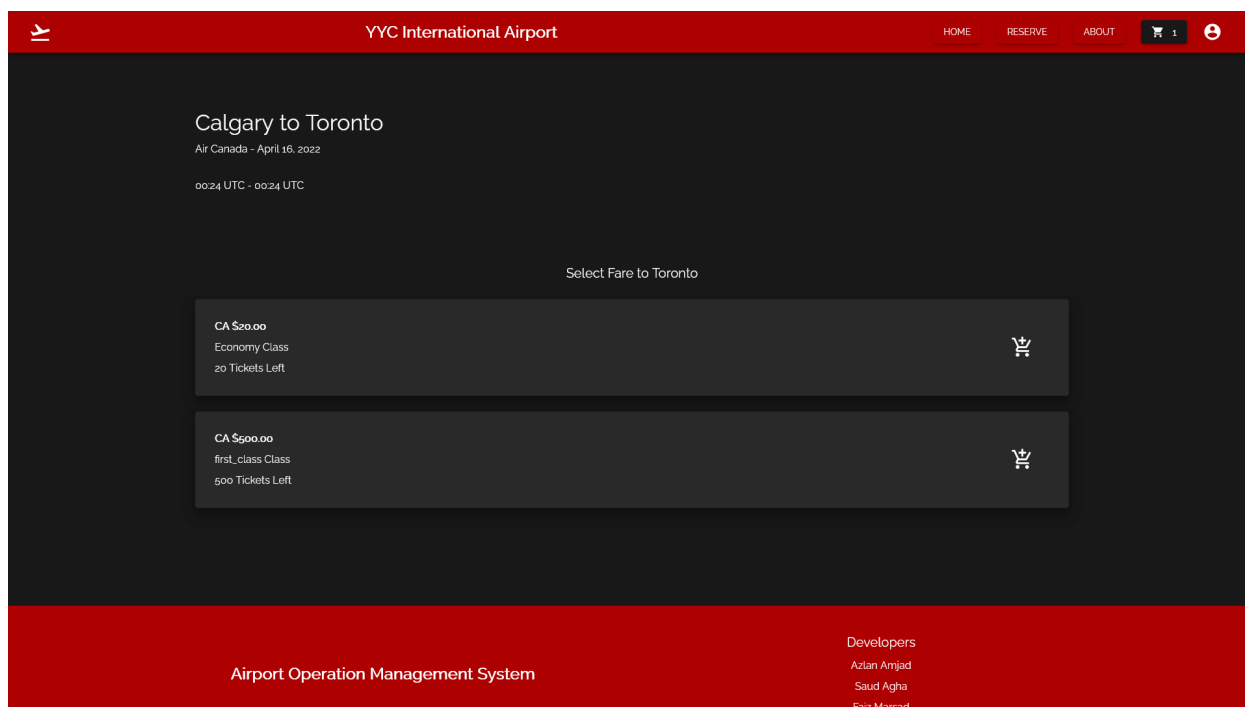
To add a complaint, press one of the options of who to file against. For example we will file against the Airport.



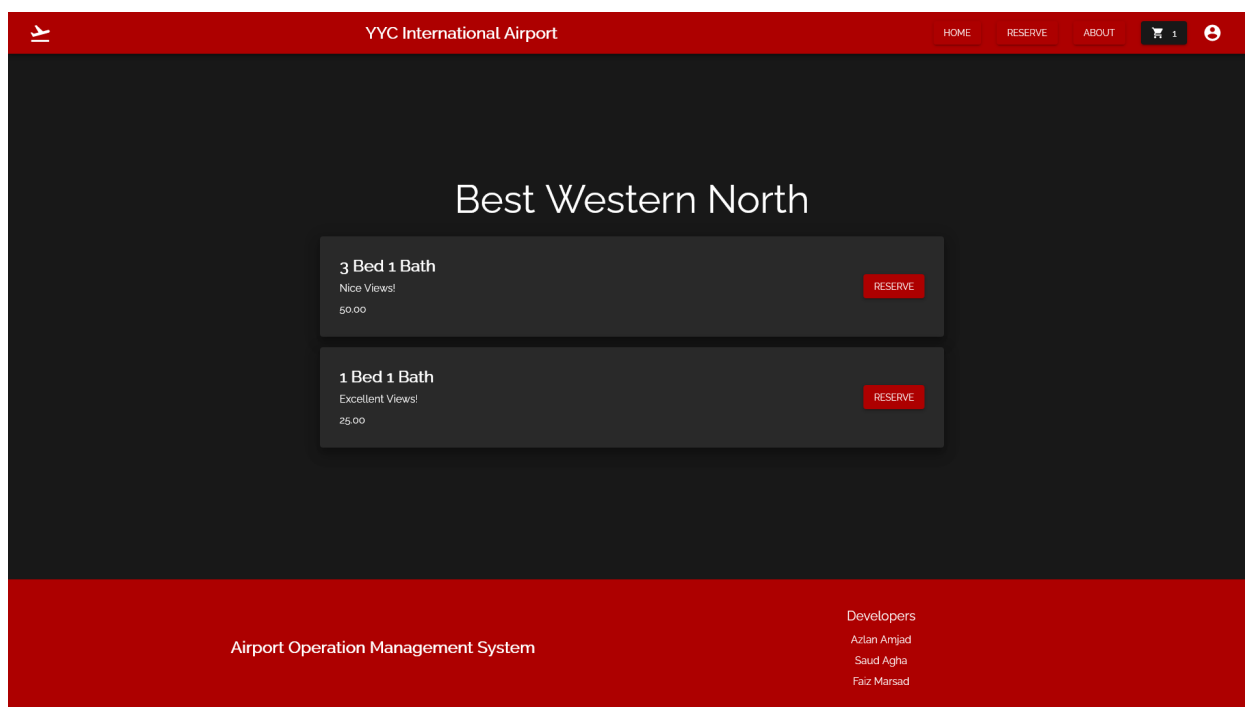
Hitting submit will add the complaint, and update the interface indicating that it has been added.



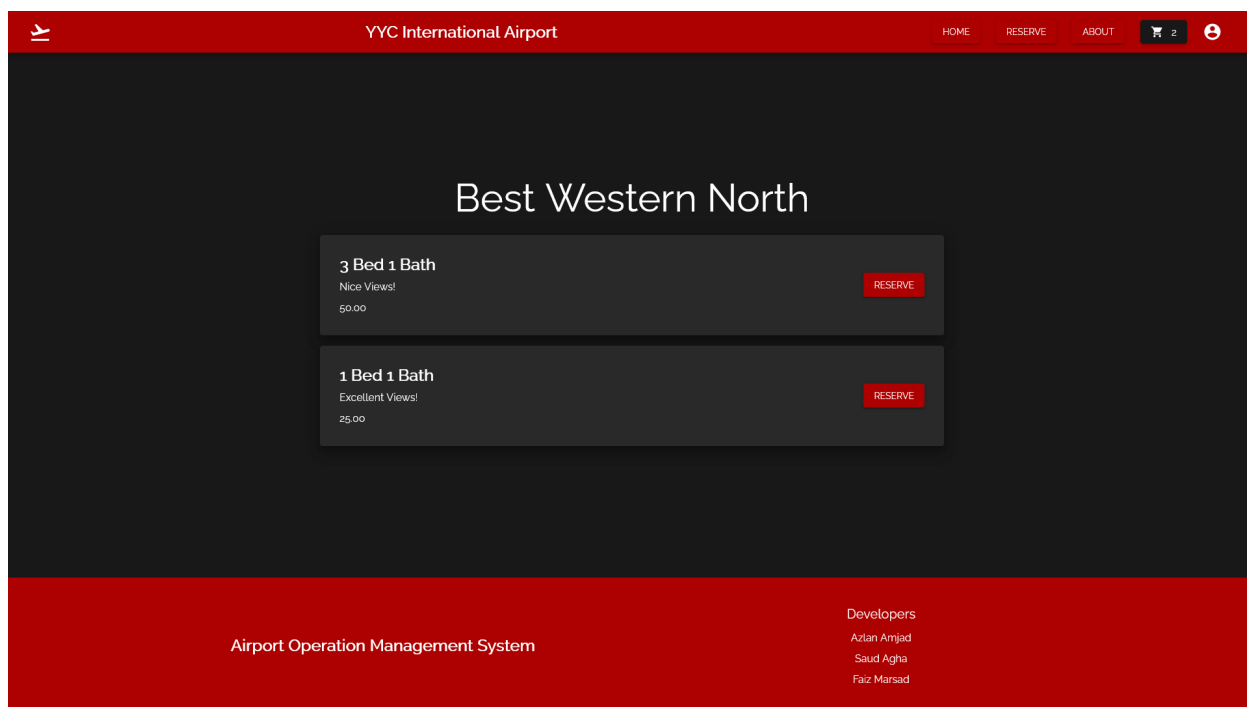
When viewing a flight the passenger has the option to add a fare to the cart. By hitting the “Add to Cart” button.



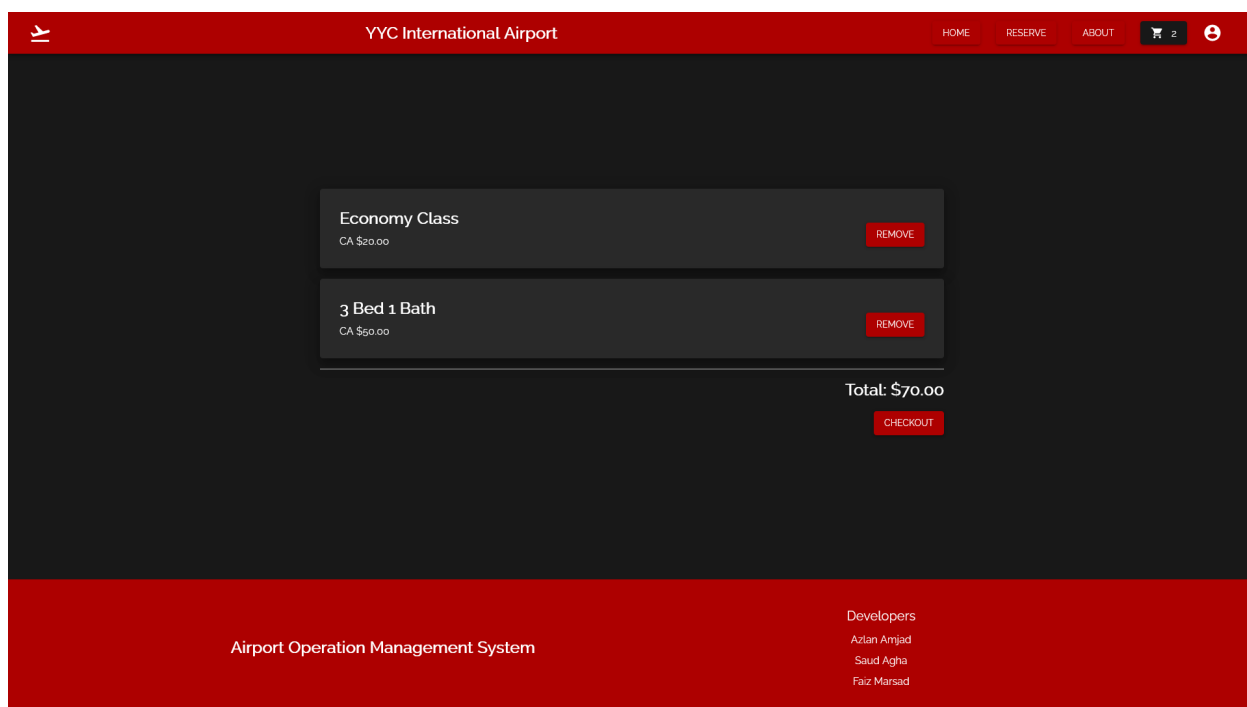
Doing so will increase the cart number, indicating the added item.



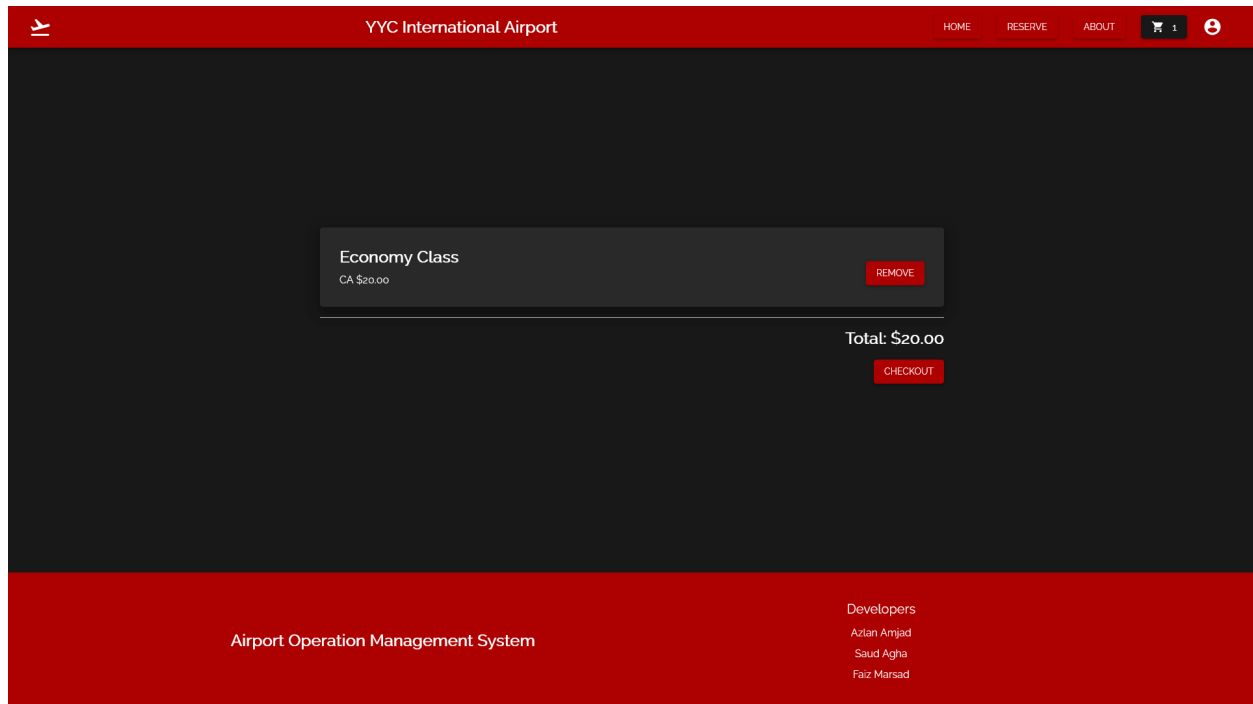
In the Hotel Stay's page, the passenger can press the "RESERVE" button to add a stay to the cart.



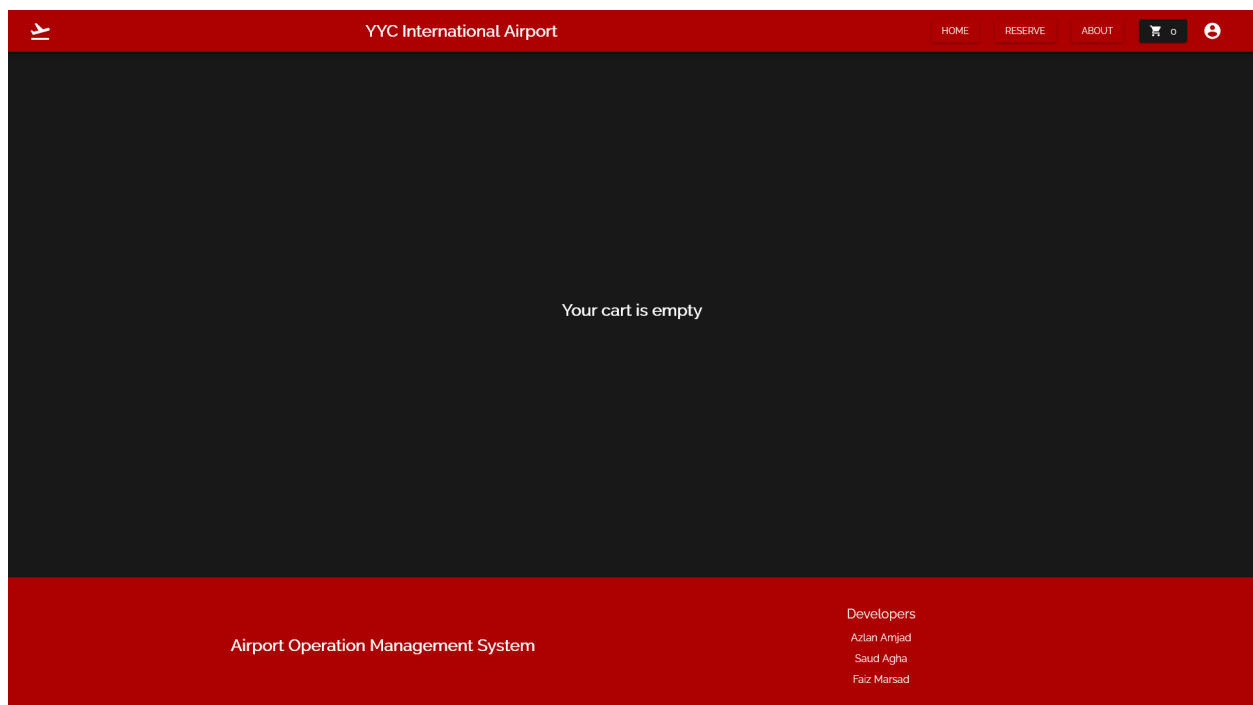
Doing so will also increase the number of items in the cart.



In the cart, passengers can view items previously added. Clicking the “REMOVE” button removes the selected item from the cart. Let's remove the “3 Bed 1 Bath” stay.



The stay has been removed from the cart.

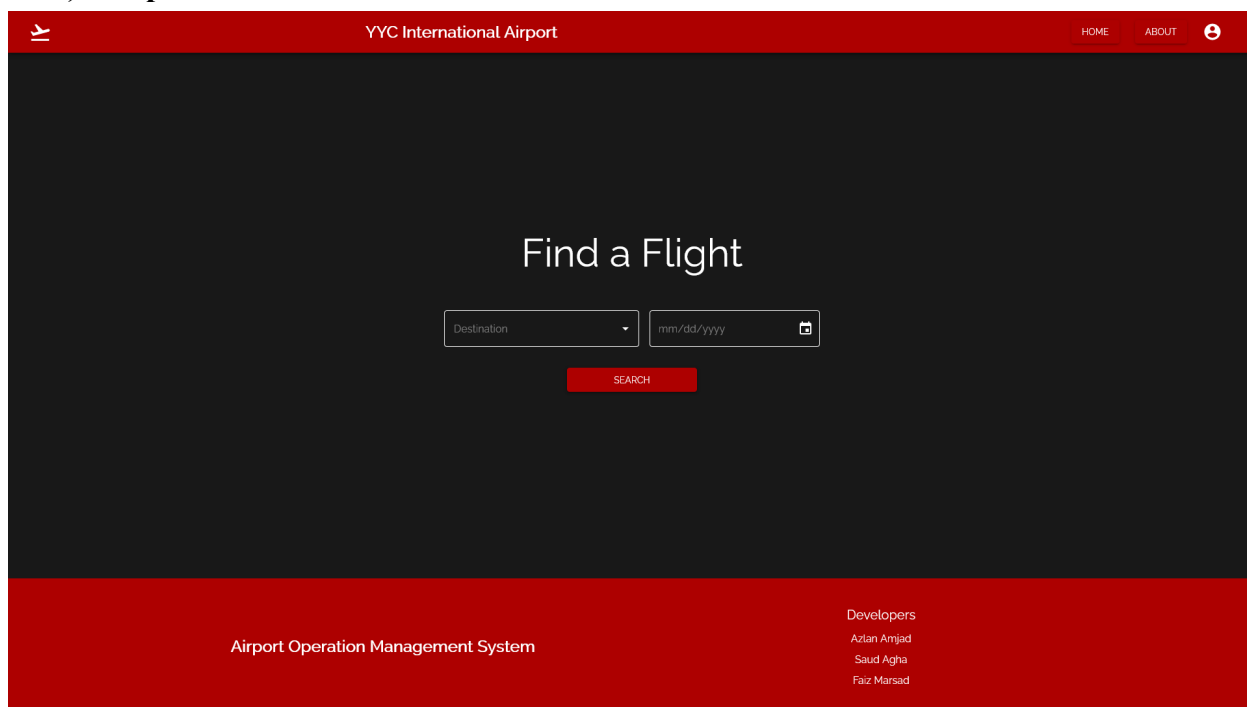


When hitting the checkout button, the passenger purchases the added items and the cart empties.

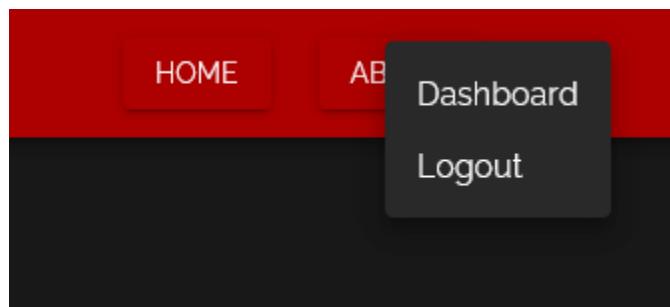
	Id [PK] bigint	fare_id bigint	passenger_id bigint
1	1	1	[null]
2	2	1	[null]
3	3	2	[null]
4	4	1	1
5	5	9	3

The database is updated for the purchased items.

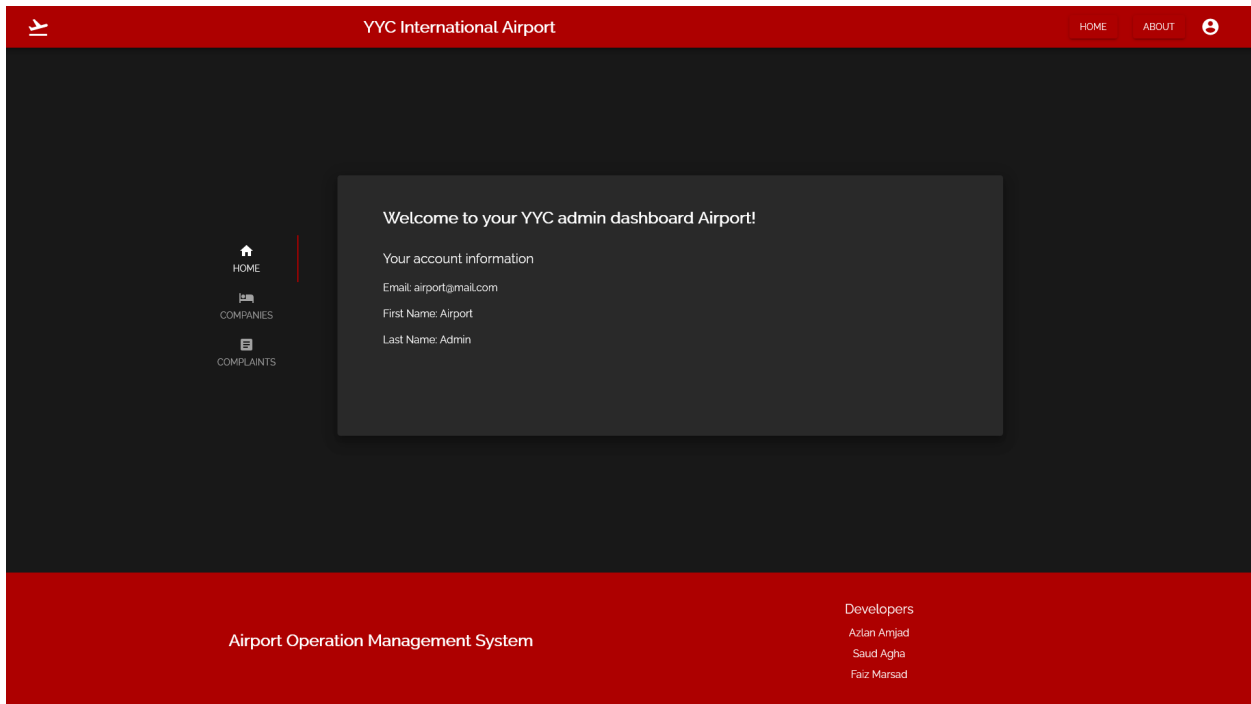
4) Airport Admin Functionalities



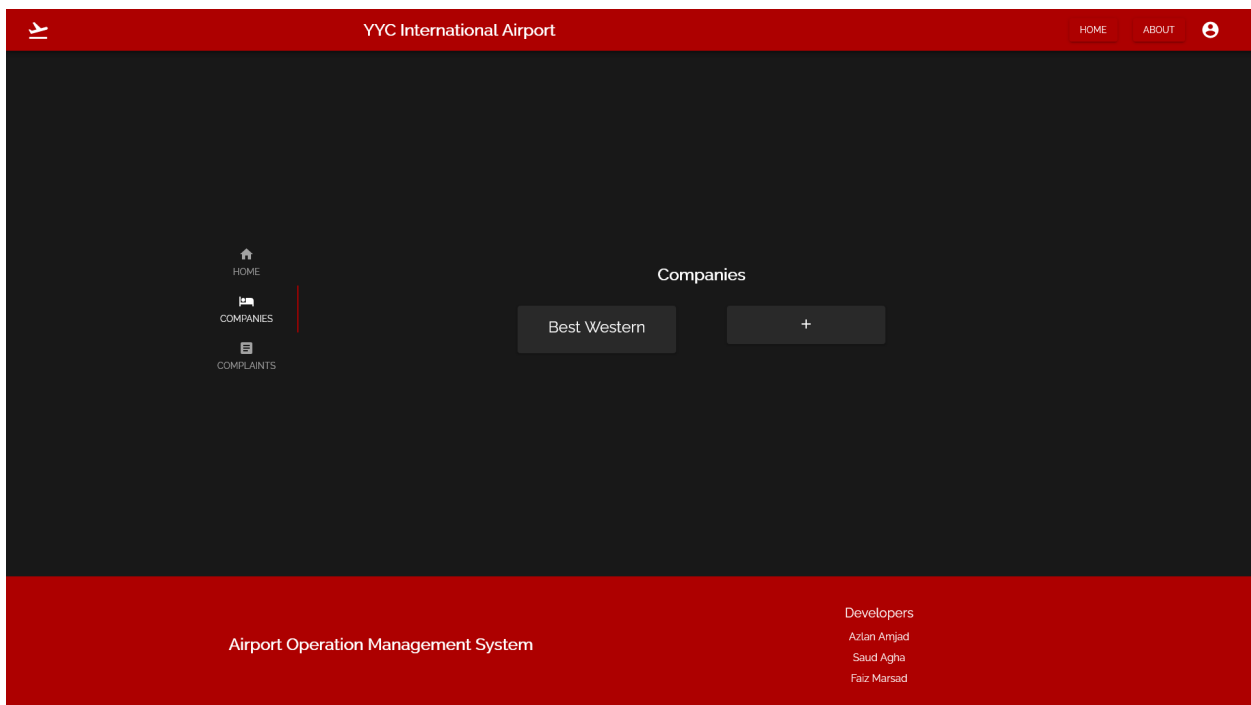
Once logged in as an airport admin, the user will be redirected to the homepage. The navbar has been updated to reflect the airport administrator.



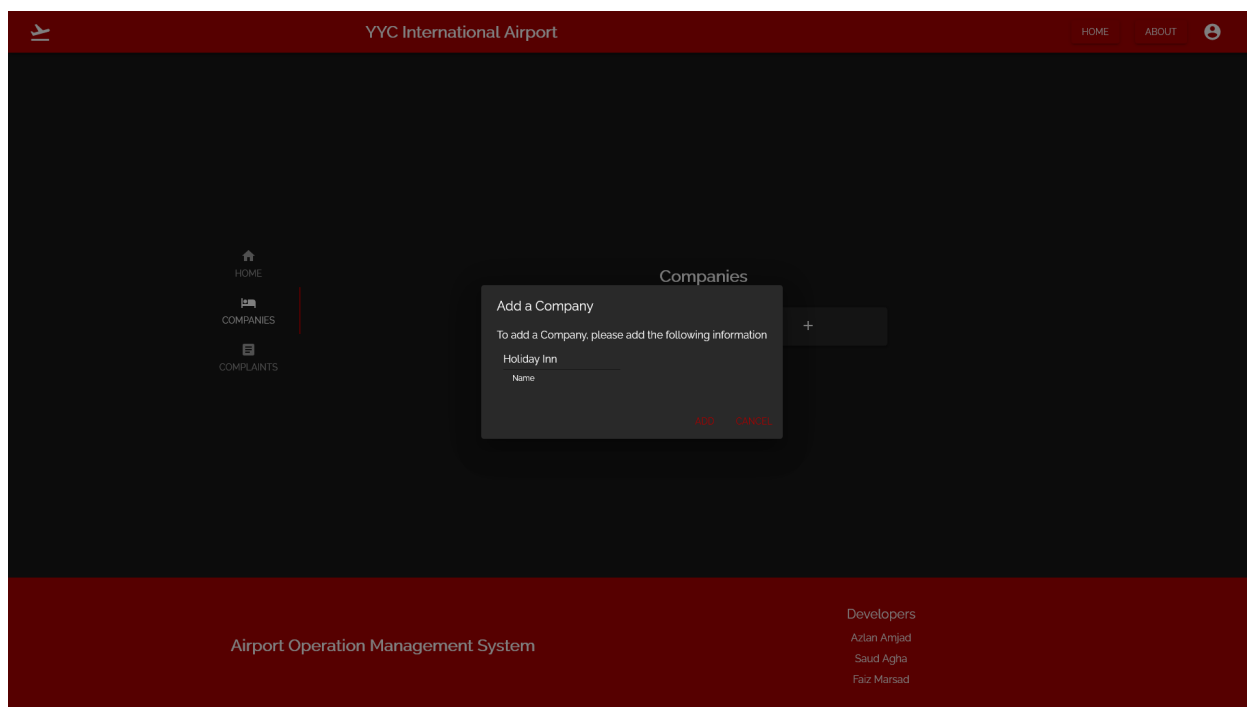
Hitting the avatar on the top right, will reveal a set of options. Let's start with the "Dashboard".



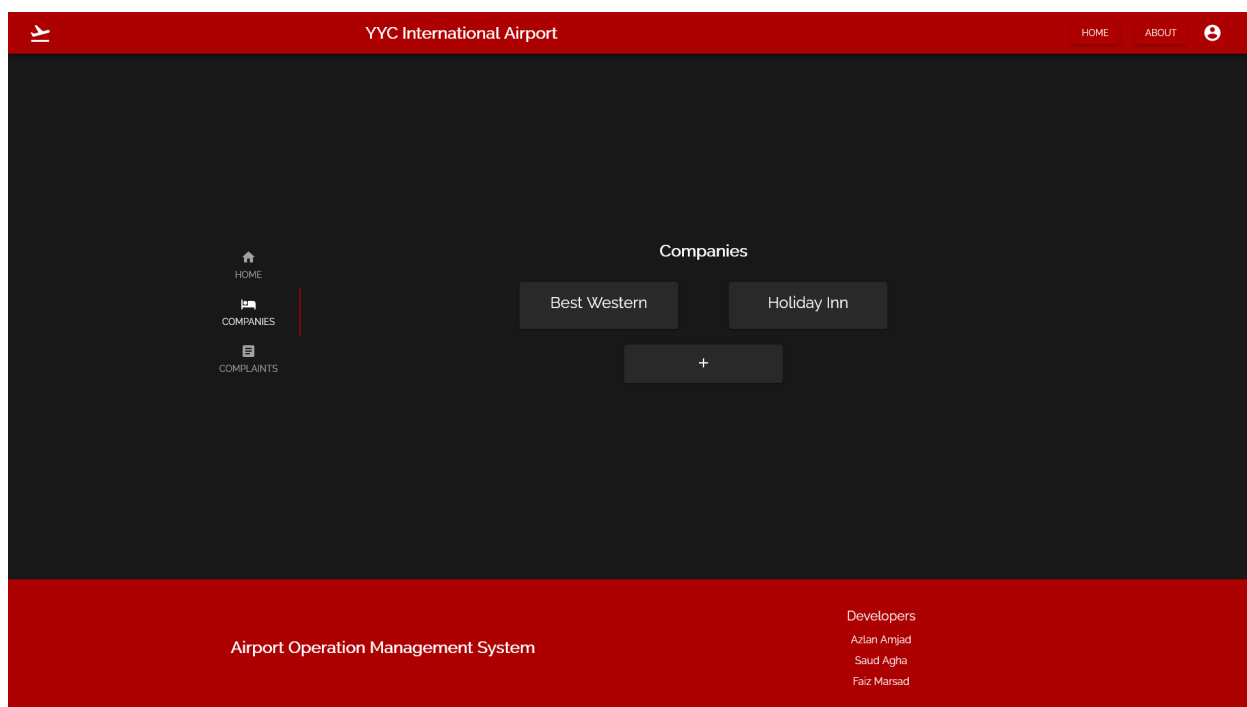
The dashboard acts as the hub for the airport administrator. Here, the account information is displayed and the admin has the option to add stays, or resolve any complaints. Let's start with adding a stay.



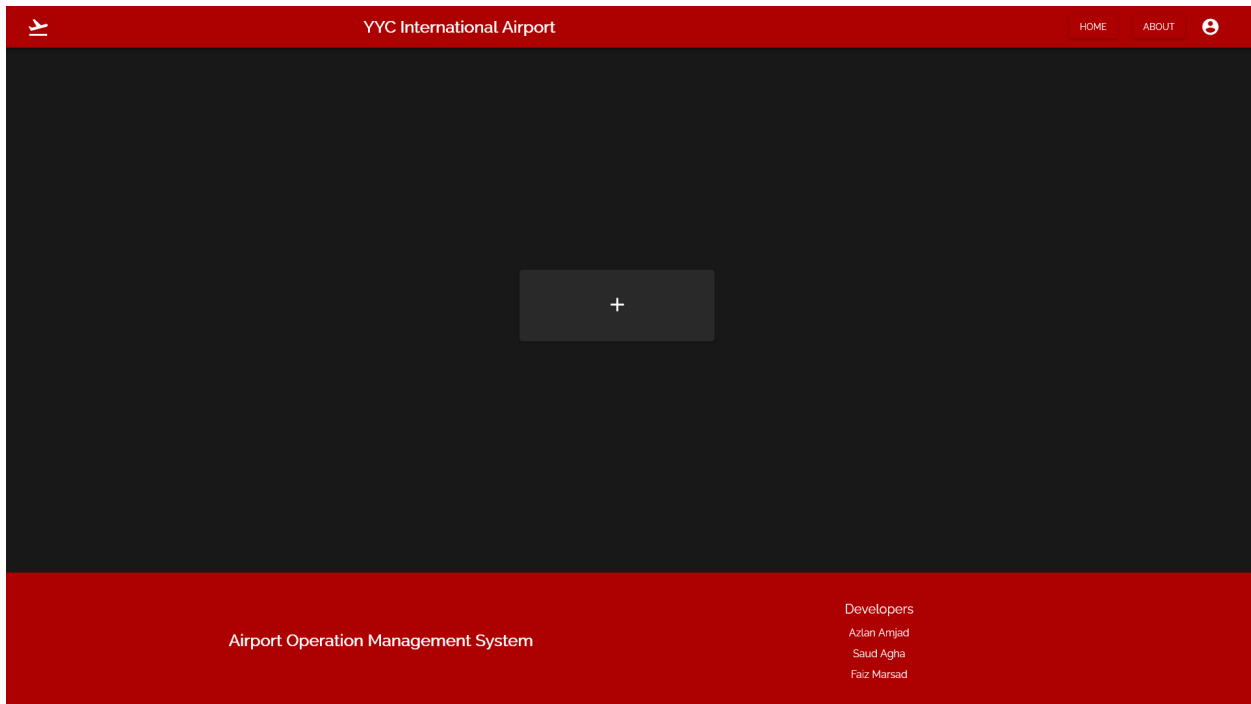
Selecting the "COMPANIES" tab, will reveal the already added companies as well as an option to add an additional company.



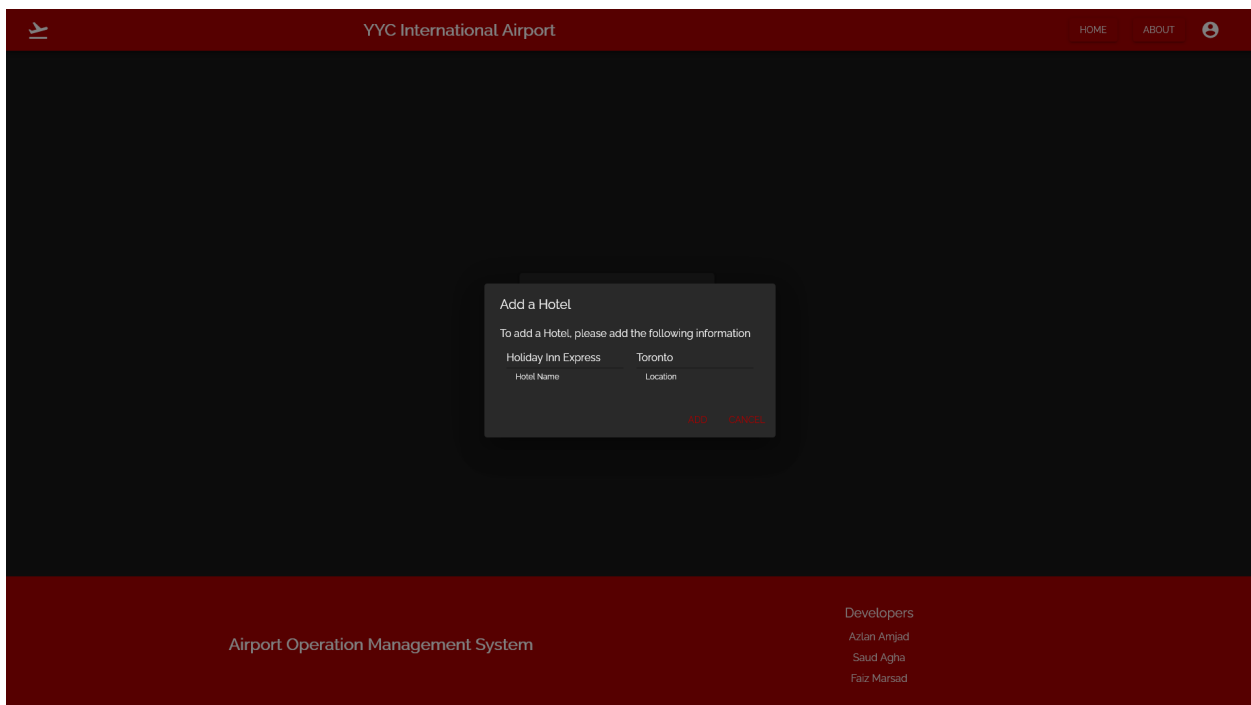
By pressing the “+” button, a menu will pop up asking for the name of the company to be added.



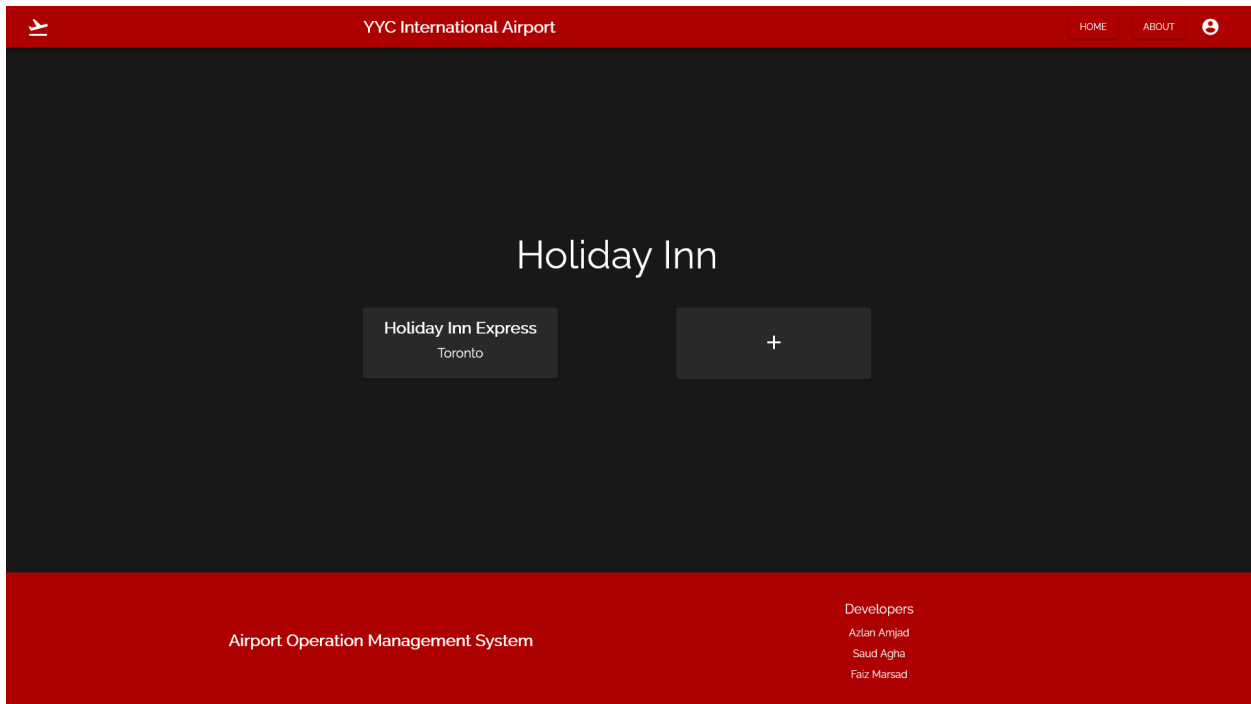
Hitting the “ADD” button will add the company to the database, and update the interface. Lets select the added company.



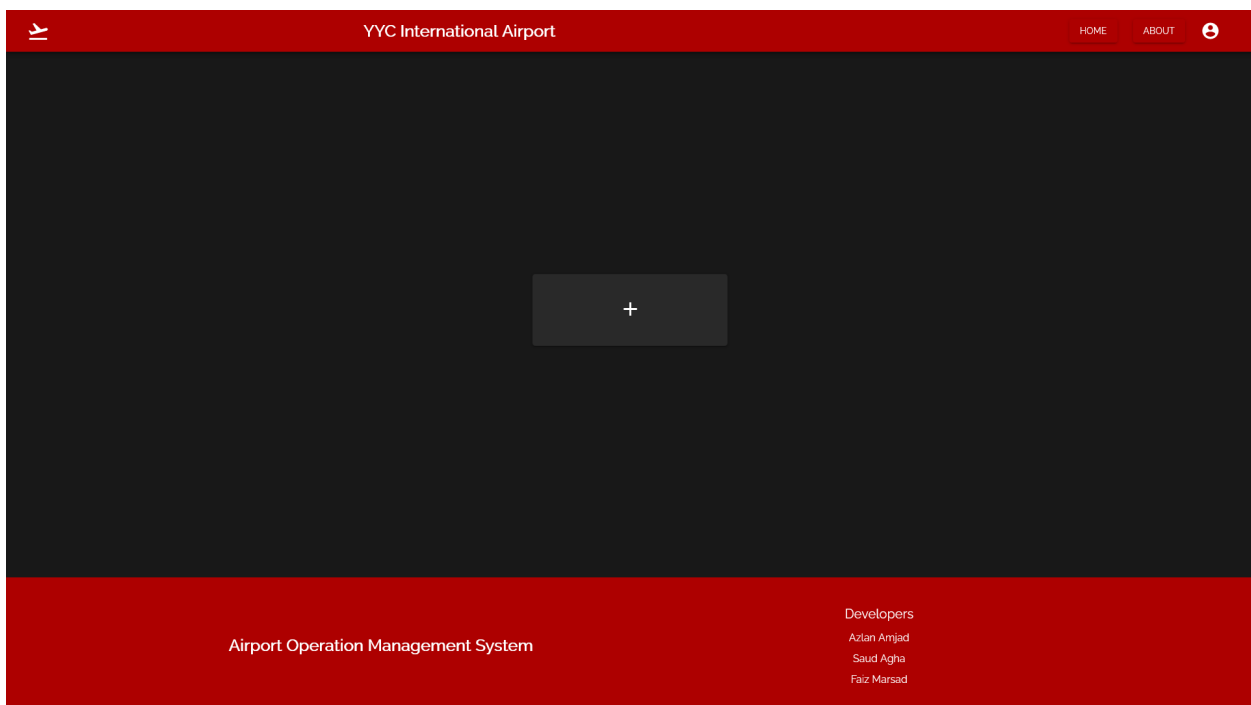
Initially, the added company will have no corresponding hotels. It is the airport administrators job to add hotels for each company. To do so, press the “+” button.



A new menu will pop up. Fill in the required information and hit “ADD”.



Doing so, will populate the interface with the added hotel. This hotel is also added to the database. Lets select the added hotel, to add a stay.



Initially, newly added hotels will have no stays. To add one, press the “+” button.

YYC International Airport

HOME ABOUT

Add a Stay

To add a Stay, please add the following information

3 Beds 2 Bath's 300 ☐

Stay Name Price (\$)

Description

Good Views

ADD CANCEL

Airport Operation Management System

Developers
Azlan Amjad
Saud Agha
Faiz Marsad

Hitting the “+” button will reveal a new menu. Fill in the information and hit “ADD” to add the stay.

YYC International Airport

HOME ABOUT

Holiday Inn Express

3 Beds 2 Bath's

Good Views

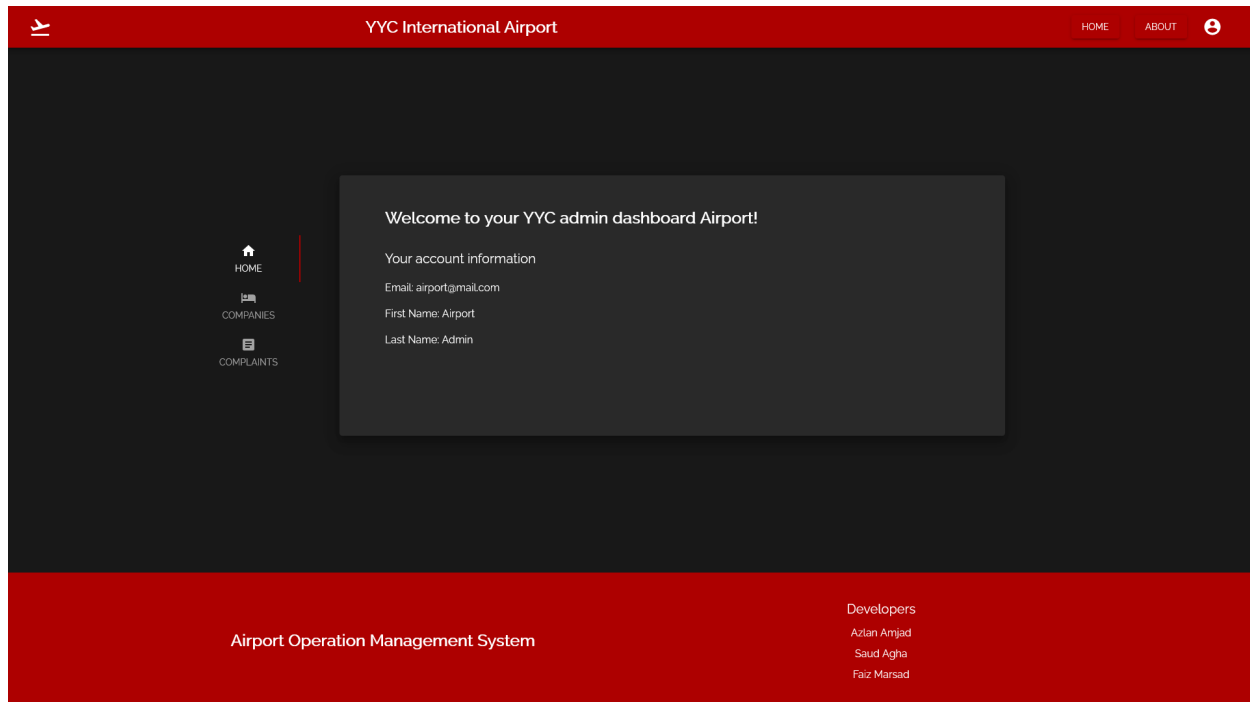
300.00

+

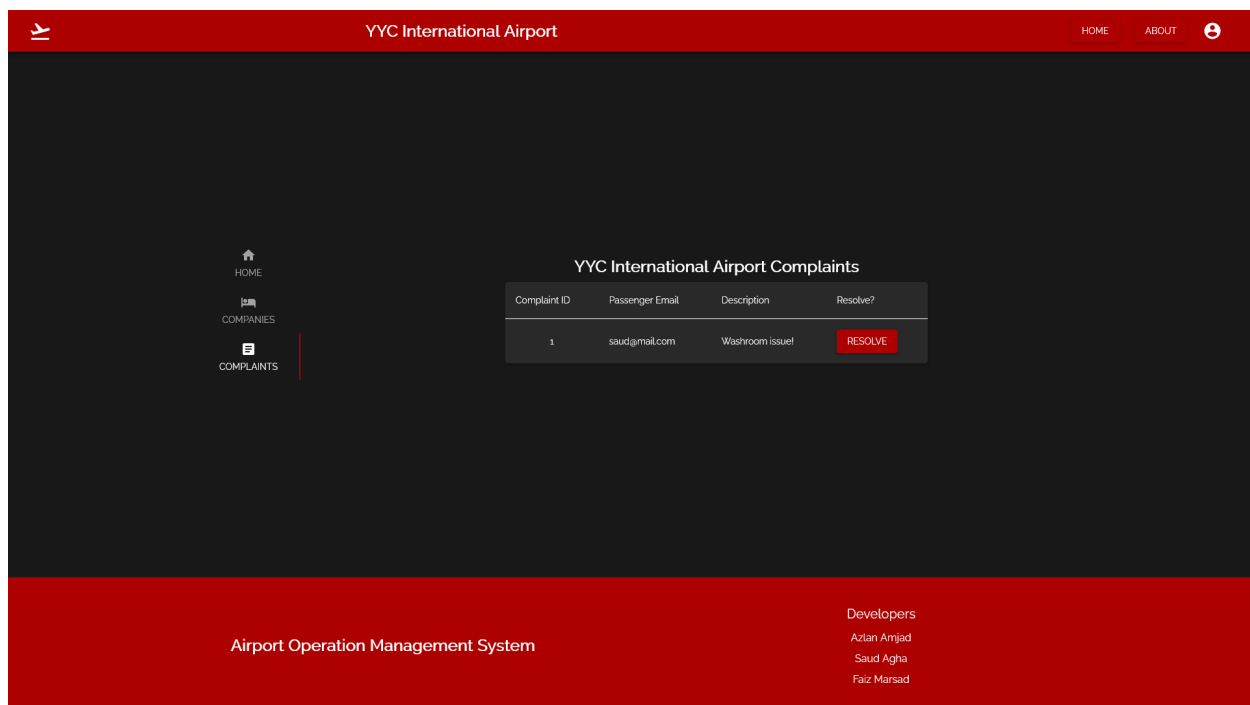
Airport Operation Management System

Developers
Azlan Amjad
Saud Agha
Faiz Marsad

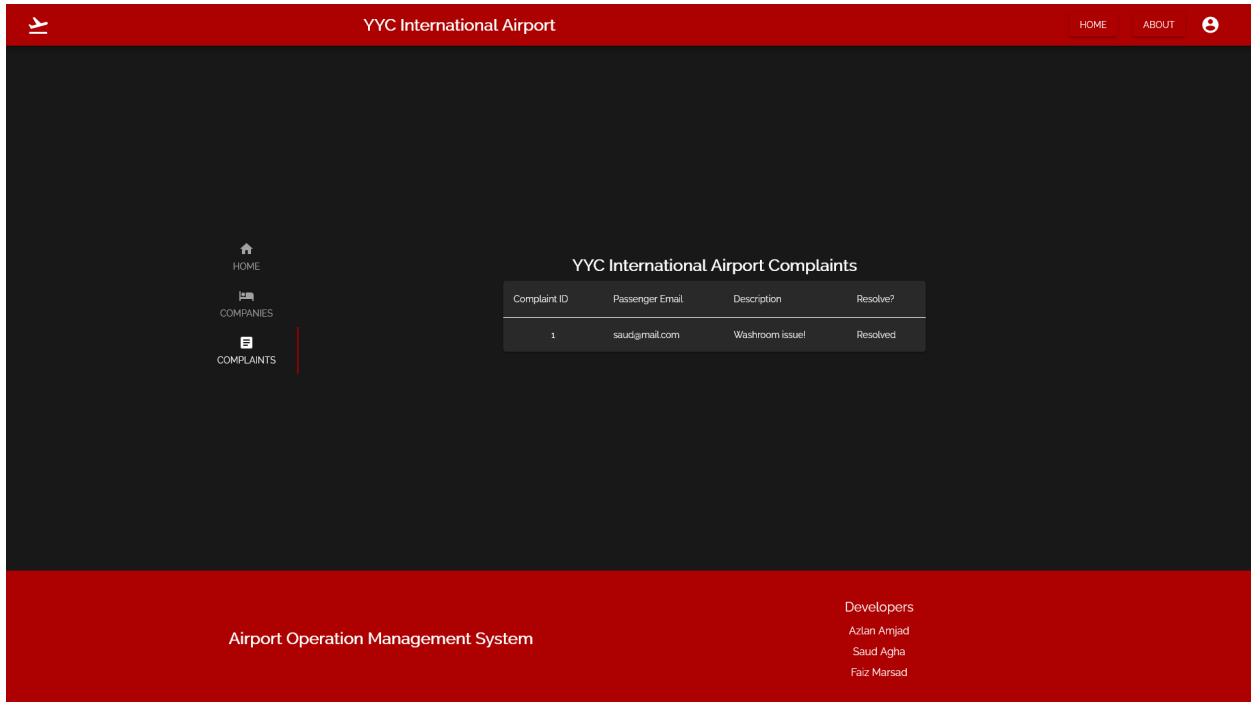
The stay has been added to the database, and the interface will update to reflect that. This stay is now purchasable by potential passengers.



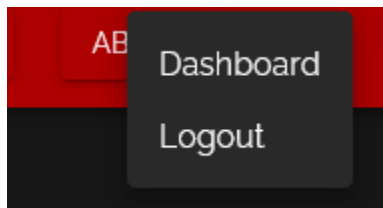
Let's head back to the airport-administrator's dashboard (<http://localhost:3000/airport-dashboard/home>) to look at the "COMPLAINTS" option.



Selecting "COMPLAINTS" will reveal any complaints made against the airport. It is the job of the airport administrator to resolve the complaint, once the complaint has been resolved hit the "RESOLVE" button to update the database.

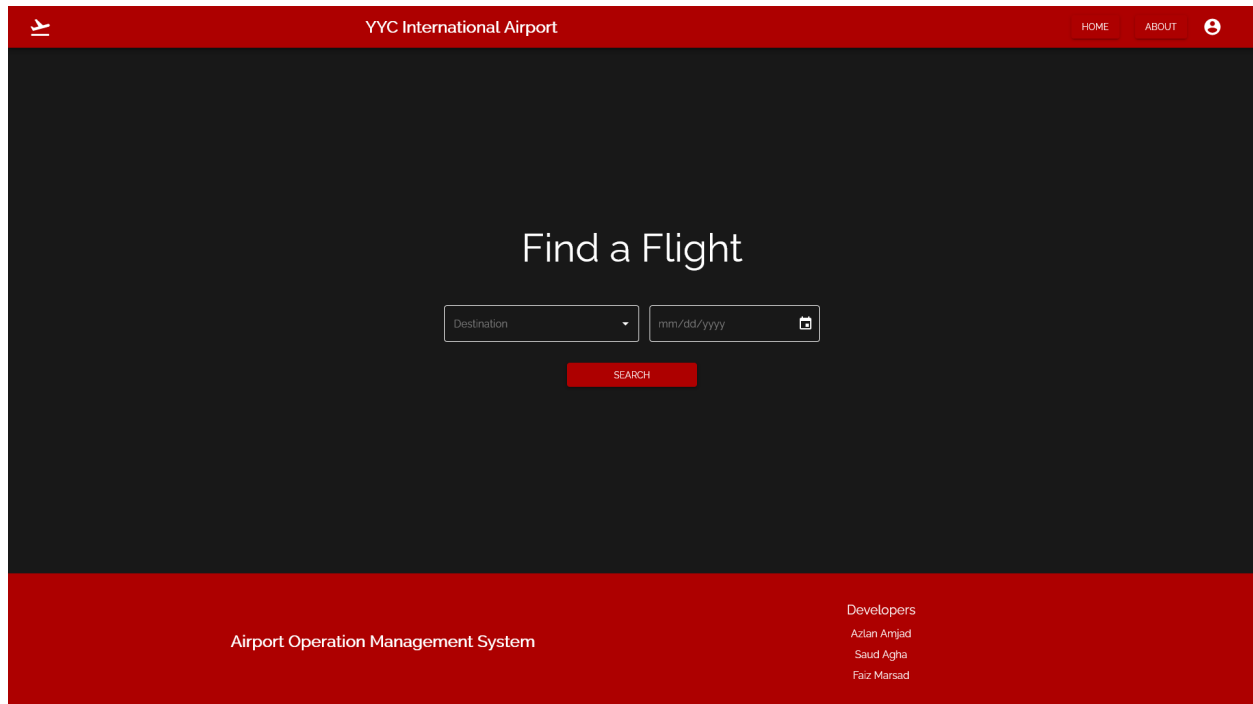


A resolved complaint will show up “Resolved”. With this, we have covered the functionalities of the airport administrator. Click the avatar in the top right, and select logout to logout securely.

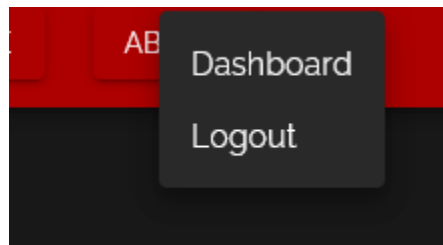


Next, we'll go over the Airline administrator.

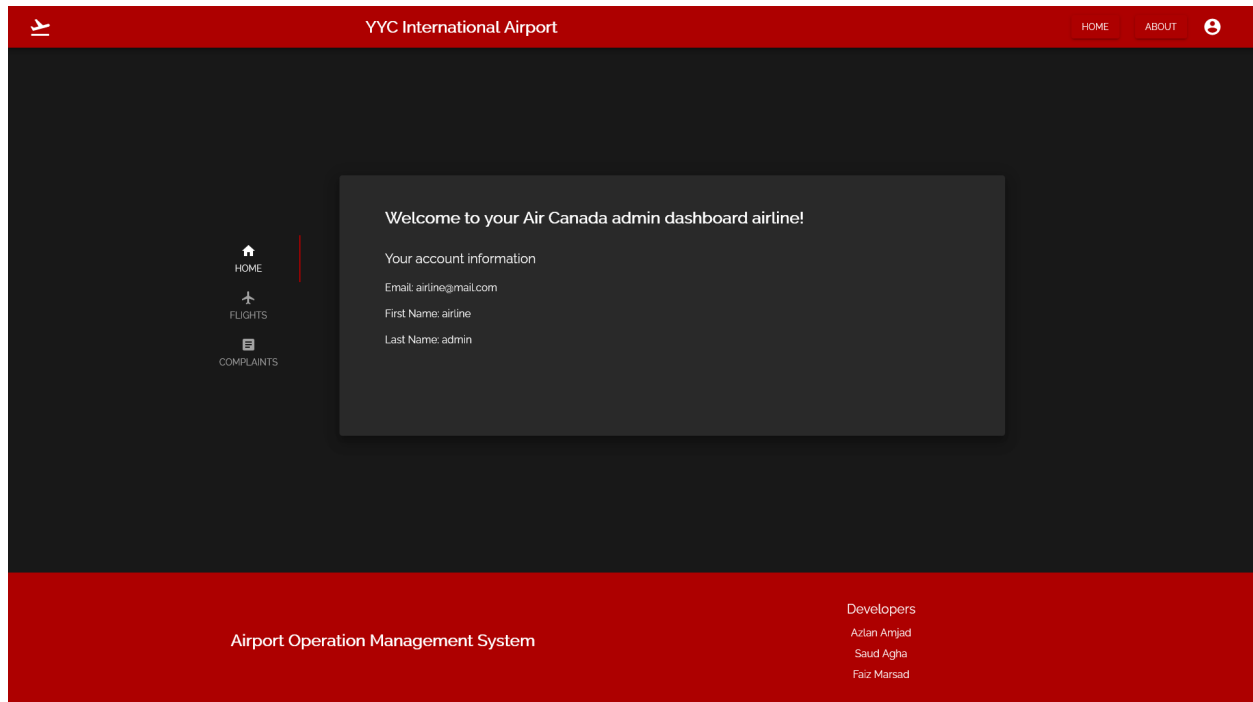
5) Airline Admin Functionalities



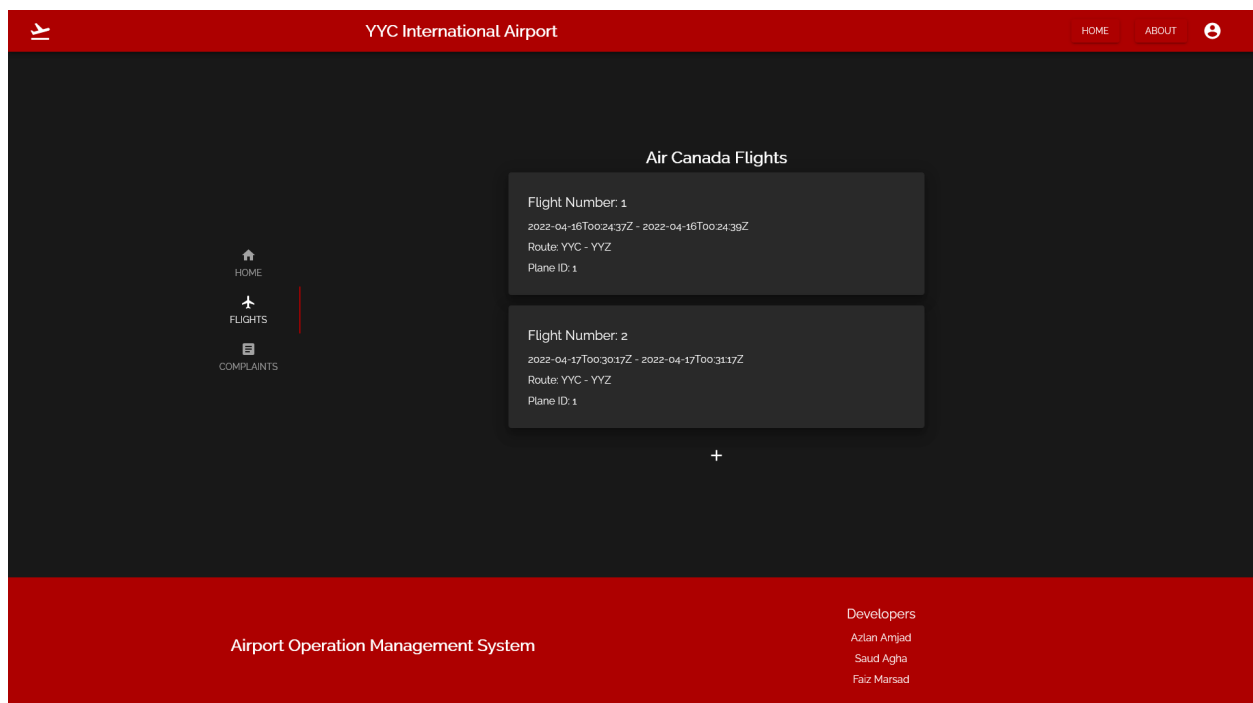
For the Airline-admin, once logged in you will be redirected to the homepage. Similar to the airport administrator, the main functionalities lie in the dashboard.



To access, press the avatar button in the top right, and select “Dashboard”.



Doing so, will redirect to the admin dashboard. Showing the account information, as well as the different functionalities. To start we'll go over the "FLIGHTS" tab.



Here, all the flights for the airline are displayed, and an option to add additional flights. Press the "+" button to add a flight.

Fill in the required information and hit “ADD” to add the flight and fares to the database.

Once added, the interface will reflect the change. The flight + fares will be viewable and purchasable by potential passengers. Lets go to the “COMPLAINTS” section next.

YYC International Airport

HOME ABOUT

HOME
FLIGHTS
COMPLAINTS

Air Canada Complaints

Complaint ID	Passenger Email	Description	Resolve?
1	saud@gmail.com	Long lines? Really?	RESOLVE

Airport Operation Management System

Developers
Azlan Amjad
Saud Agha
Faiz Marsad

In the COMPLAINTS section, any complaint made against the airline will be displayed. Once the airline administrator resolves the complaint, they must hit the “RESOLVE” button to update the database.

YYC International Airport

HOME ABOUT

HOME
FLIGHTS
COMPLAINTS

Air Canada Complaints

Complaint ID	Passenger Email	Description	Resolve?
1	saud@gmail.com	Long lines? Really?	Resolved

Airport Operation Management System

Developers
Azlan Amjad
Saud Agha
Faiz Marsad

Once resolved, the table will update to reflect the change. That summarizes the key functionalities of the Airline-Administrator, and our user manual. For any assistance please contact us and we will help resolve. Thanks

IX. REFERENCES

- 1) *Appropriate uses for SQLite*. [Online]. Available:
<https://www.sqlite.org/whentouse.html>. [Accessed: 15-Apr-2022].