

BOĞAZİÇİ UNIVERSITY

CMPE 321

PROJECT 3

SUMMER 2017

Airline Reservation System

Mustafa Enes ÇAKIR

August 4, 2017

Contents

1	Introduction	2
2	Interface	2
3	Database	6
3.1	ER Diagram	6
3.2	Procedures	6
3.3	Triggers	8
4	Conclusions	8

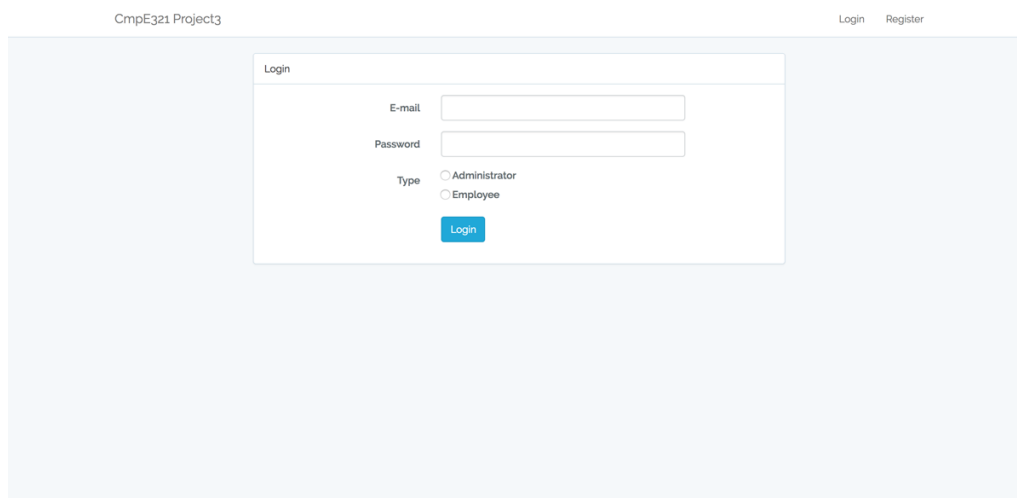
1 Introduction

It's an in-house web-based airlines reservation application. Administrators can generate general instances and employees make reservations for customers. They can select seat on plane easily.

I was planning to implemented it with Laravel Framework. But Laravel is handling almost everything, such as *SQL inception*. So, I decided to developed my own little PHP framework. I heavily inspired from Laravel.

Entry point of my application is *index.php*. After that point, my *Router* handles routing. I have 3 important directory. *App* directory contains project specific files, such as *Models*, *Controller*, *Routes*, *Views*. *Core* directory contains my framework classes, such as *QueryBuilder*, *Auth*, *Router*, *Request*. *Public* directory contains stylesheets, scripts etc.

2 Interface



CmpE321 Project3

Login Register

Login

E-mail

Password

Type ☐ Administrator ☐ Employee

Login

Figure 1: Login Page for Administrators and Employees

CmpE321 Project3 Administrators Employees Airports Airplanes Pilots Flights Enes Çakır

Airports + Create









#	Name	City	
1	Atatürk Airport	İstanbul	 
2	Sabiha Gökçen Airport	İstanbul	 
3	Adnan Menderes	İzmir	 
4	Esenboğa Airport	Ankara	 

Figure 2: Airport Listing Page

CmpE321 Project3 Administrators Employees Airports Airplanes Pilots Flights Enes Çakır

Pilots + Create









#	Name	Age	
1	Murat Bakan	45	 
2	Barış Savaş	30	 
3	Mehmet Ahmet	38	 
4	Ayşe Fatma	28	 

Figure 3: Pilot Listing Page

CmpE321 Project3

AdministratorsEmployeesAirportsAirplanesPilotsFlights

Enes Çakır

Flights

+ Create









#	Number	Departure	Destination	Pilot	Airplane	Full Seat #	Total Seat #	Departure Time	
4	TK905	Esenboğa Airport (Ankara)	Atatürk Airport (İstanbul)	Mehmet Ahmet	Airbus 123	4	96	19-08-2017 13:00	 
5	TK403	Esenboğa Airport (Ankara)	Adnan Menderes (İzmir)	Barış Savaş	Boing 303	0	80	01-08-2017 05:00	 
6	TK202	Sabiha Gökçen Airport (İstanbul)	Esenboğa Airport (Ankara)	Barış Savaş	Airbus 123	0	96	01-08-2017 09:30	 
7	TK101	Atatürk Airport (İstanbul)	Adnan Menderes (İzmir)	Ayşe Fatma	Boing 405	0	120	02-08-2017 06:05	 

Figure 4: Flight Listing Page

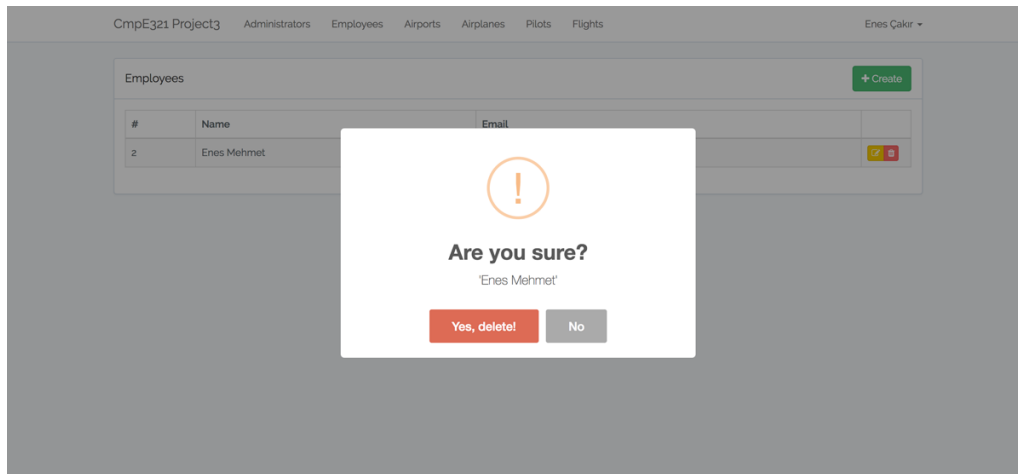


Figure 5: Delete Confirmation Popup

CmpE321 Project3 Administrators Employees Airports Airplanes Pilots Flights Enes Çakır ▾

Create an Flight

Flight Number

Departure Time

Departure Airport

Atatürk Airport ▾

Destination Airport

Sabiha Gökçen Airport ▾

Pilot

Murat Bakan ▾

Airplane

Boeing 303 ▾

Save

Figure 6: Creating a Flight Page

CmpE321 Project3 Customers Reservations Enes Mehmet ▾

Make a Reservation

Customer Name

Osman Mehmet

Seat:

A2

Reserve It

	A	B	C	D
1	Red	Red	Red	Red
2	Green	Grey	Red	Red
3	Grey	Grey	Grey	Grey
4	Grey	Grey	Red	Grey
5	Grey	Grey	Grey	Grey
6	Grey	Grey	Grey	Grey
7	Grey	Grey	Grey	Grey
8	Grey	Grey	Grey	Grey
9	Grey	Grey	Grey	Grey
10	Grey	Grey	Grey	Grey
11	Grey	Grey	Grey	Grey

Figure 7: Making Reservation Page

3 Database

3.1 ER Diagram

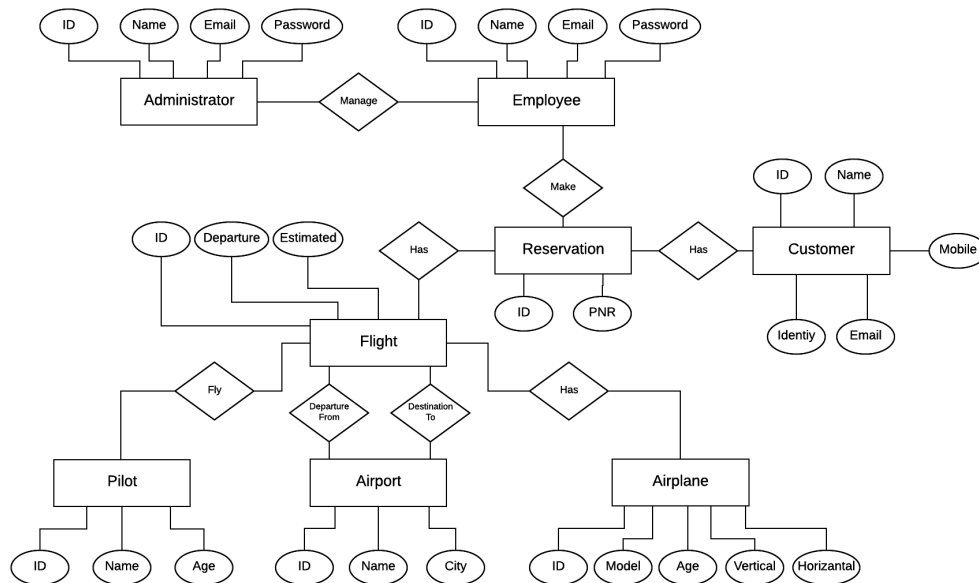


Figure 8: ER Diagram of Database

3.2 Procedures

```
1 CREATE PROCEDURE `past_flights`(IN pid VARCHAR(10))
2 BEGIN
3     IF pid = "ALL" THEN
4         SELECT f.*, des.name as des_name, des.city as des_city,
5             dep.name as dep_name, dep.city as dep_city,
6             p.name as p_name, ap.model as ap_model
7         FROM flights as f
8         JOIN airports as des ON des.id = f.destination_id
9         JOIN airports as dep ON dep.id = f.departure_id
10        JOIN pilots as p ON p.id = f.pilot_id
11        JOIN airplanes as ap ON ap.id = f.airplane_id
12        WHERE f.departured_at < NOW();
13    ELSE
14        SELECT f.*, des.name as des_name, des.city as des_city,
```

```

15         dep.name as dep_name, dep.city as dep_city,
16         p.name as p_name, ap.model as ap_model
17     FROM flights as f
18     JOIN airports as des ON des.id = f.destination_id
19     JOIN airports as dep ON dep.id = f.departure_id
20     JOIN pilots as p ON p.id = f.pilot_id
21     JOIN airplanes as ap ON ap.id = f.airplane_id
22     WHERE f.departured_at < NOW() AND f.pilot_id = pid;
23 END IF;
24 END;

```

Code 1: Past Flight Procedure

```

1 CREATE PROCEDURE `future_flights`(IN pid VARCHAR(10))
2 BEGIN
3     IF pid = "ALL" THEN
4         SELECT f.*, des.name as des_name, des.city as des_city,
5             dep.name as dep_name, dep.city as dep_city,
6             p.name as p_name, ap.model as ap_model
7         FROM flights as f
8         JOIN airports as des ON des.id = f.destination_id
9         JOIN airports as dep ON dep.id = f.departure_id
10        JOIN pilots as p ON p.id = f.pilot_id
11        JOIN airplanes as ap ON ap.id = f.airplane_id
12        WHERE f.departured_at > NOW();
13    ELSE
14        SELECT f.*, des.name as des_name, des.city as des_city,
15            dep.name as dep_name, dep.city as dep_city,
16            p.name as p_name, ap.model as ap_model
17        FROM flights as f
18        JOIN airports as des ON des.id = f.destination_id
19        JOIN airports as dep ON dep.id = f.departure_id
20        JOIN pilots as p ON p.id = f.pilot_id
21        JOIN airplanes as ap ON ap.id = f.airplane_id
22        WHERE f.departured_at > NOW() AND f.pilot_id = pid;
23    END IF;
24 END;

```

Code 2: Future Flight Procedure

```
1 CREATE PROCEDURE `full_seats` ( IN f_id VARCHAR(10) )
2 BEGIN
3     SELECT seat FROM reservations WHERE flight_id = f_id;
4 END;
```

Code 3: Full Seat Procedure

3.3 Triggers

```
1 CREATE TRIGGER `check_customer` BEFORE INSERT ON reservations
2 FOR EACH ROW
3 BEGIN
4     IF EXISTS (SELECT * FROM customers WHERE name LIKE NEW.customer_name)
5     THEN
6         SET NEW.customer_id = (SELECT id FROM customers
7                                 WHERE name LIKE NEW.customer_name
8                                 LIMIT 1);
9         SET NEW.customer_name = NULL;
10    ELSE
11        INSERT INTO customers(name) VALUES (NEW.customer_name);
12        SET NEW.customer_id = (SELECT LAST_INSERT_ID());
13        SET NEW.customer_name = NULL;
14    END
```

Code 4: Customer Checking Trigger

4 Conclusions

It was a great chance for me to understand foundation of web frameworks. I encountered with lots of problems, and I tried to find best practices every time. In this process, I investigated Laravel's core files.

I used *PDO* class instead of *mysqli*. For resolving SQL inception issue, I *PDO* parametrized queries.