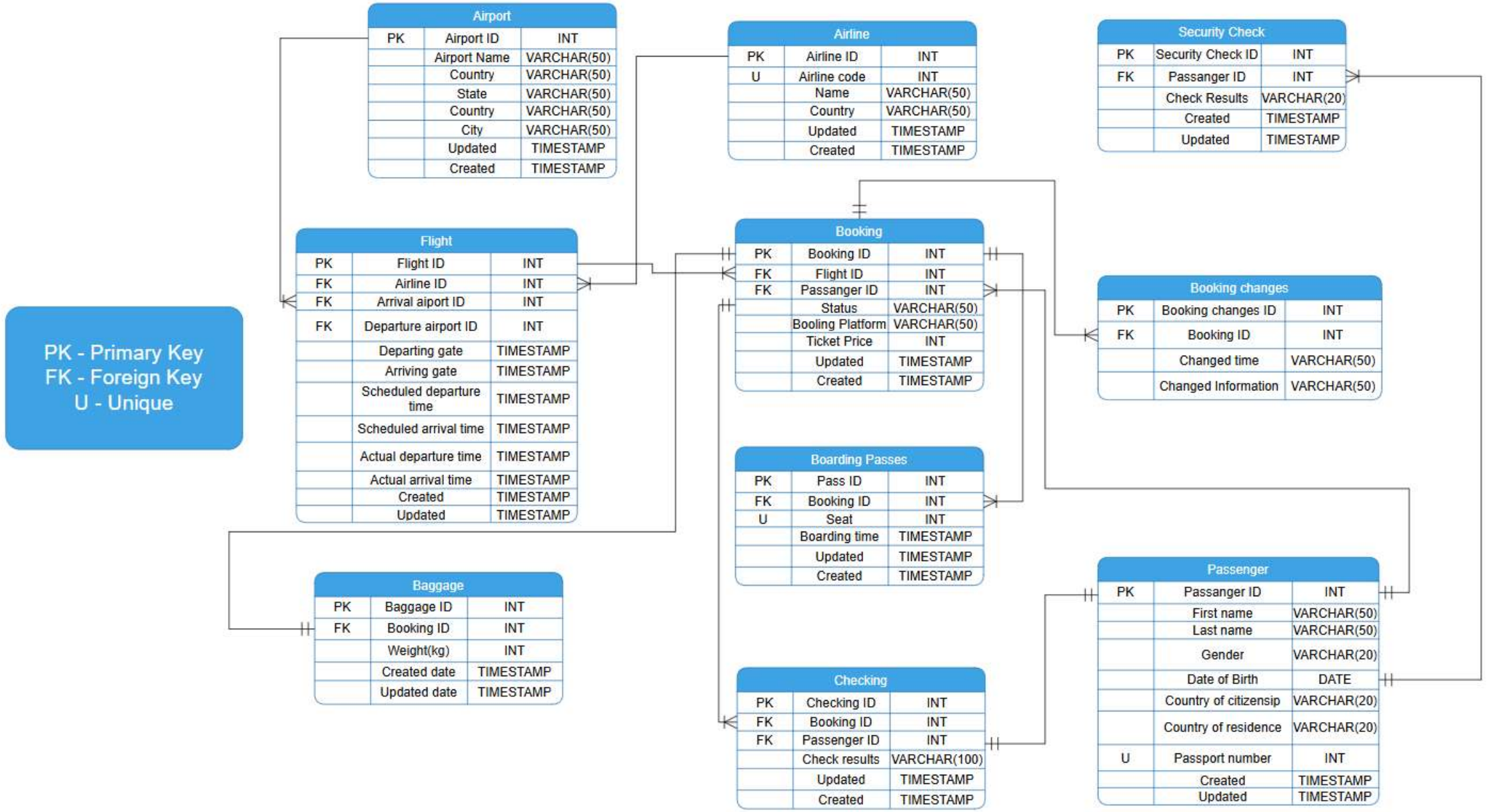
PK - Primary Key
FK - Foreign Key
U - Unique
Airport
PK Airport ID INT
Airport Name VARCHAR(50)
Country VARCHAR(50)
State VARCHAR(50)
Country VARCHAR(50)
City VARCHAR(50)
Updated TIMESTAMP
Created TIMESTAMP
Airline
PK Airline ID INT
U Airline code INT
Name VARCHAR(50)
Country VARCHAR(50)
Updated TIMESTAMP
Created TIMESTAMP
Security Check
PK Security Check ID INT
FK Passanger ID INT
Check Results VARCHAR(20)
Created TIMESTAMP
Updated TIMESTAMP
Flight
PK Flight ID INT
FK Airline ID INT
FK Arrival aiport ID INT
FK Departure airport ID INT
Departing gate TIMESTAMP
Arriving gate TIMESTAMP
Scheduled departure time TIMESTAMP
Scheduled arrival time TIMESTAMP
Actual departure time TIMESTAMP
Actual arrival time TIMESTAMP
Created TIMESTAMP
Updated TIMESTAMP
Booking
PK Booking ID INT
FK Flight ID INT
FK Passanger ID INT
Status VARCHAR(50)
Booling Platform VARCHAR(50)
Ticket Price INT
Updated TIMESTAMP
Created TIMESTAMP
Booking changes
PK Booking changes ID INT
FK Booking ID INT
Changed time VARCHAR(50)
Changed Information VARCHAR(50)
Boarding Passes
PK Pass ID INT
FK Booking ID INT
U Seat INT
Boarding time TIMESTAMP
Updated TIMESTAMP
Created TIMESTAMP
Baggage
PK Baggage ID INT
FK Booking ID INT
Weight(kg) INT
Created date TIMESTAMP
Updated date TIMESTAMP
Checking
PK Checking ID INT
FK Booking ID INT
FK Passenger ID INT
Check results VARCHAR(100)
Updated TIMESTAMP
Created TIMESTAMP
Passenger
PK Passanger ID INT
First name VARCHAR(50)
Last name VARCHAR(50)
Gender VARCHAR(20)
Date of Birth DATE
Country of citizensip VARCHAR(20)
Country of residence VARCHAR(20)
U Passport number INT
Created TIMESTAMP
Updated TIMESTAMP

Laboratory Work 1: ERD Diagram

Introduction
First, I read the system description and found the main entities: Airport, Airline, Flight, Passenger, Booking, BoardingPass, Baggage, BaggageCheck, and SecurityCheck. After that, I wrote down the attributes for each entity and marked primary keys, foreign keys, and unique fields.

Then, I checked the relationships between entities. For example, one airline can have many flights, and one passenger can have many bookings. I also noticed that the relation between flights and passengers is many-to-many, so I solved it with the Booking table.

Next, I normalized the database to the third normal form. I made sure that all attributes are atomic (1NF), that they depend on the whole primary key (2NF), and that there are no transitive dependencies (3NF).

Finally, I created the ER diagram in draw.io. I added entities, attributes, and connected them with the correct relationships and cardinalities. After finishing the diagram, I wrote a short description of the design and exported everything to PDF.

Entities and Attributes
- **Airport**: airport_id (PK), airport_name, country, state, city, created_at, updated_at.
- **Flights**: flight_id (PK), airline_id (FK), departure_airport_id (FK), arrival_airport_id (FK), departure_gate, arrival_gate, scheduled_departure_time, scheduled_arrival_time, actual_departure_time, actual_arrival_time, created_at, updated_at.
- **Airlines**: airline_id (PK), airline_code, name, country, created_at, updated_at.
- **Passengers**: passenger_id (PK), first_name, last_name, gender, date_of_birth, citizenship_country, residence_country, passport_number, created_at, updated_at.
- **Bookings**: booking_id (PK), flight_id (FK), passenger_id (FK), status, booking_platform, ticket_price, created_at, updated_at.
- **Booking Change**: change_id (PK), booking_id (FK), change_details, created_at, updated_at.
- **Boarding Pass**: boarding_pass_id (PK), booking_id (FK), seat, boarding_time, created_at, updated_at.
- **Baggage**: baggage_id (PK), booking_id (FK), weight_kg, created_at, updated_at.
- **Security Check**: check_id (PK), passenger_id (FK), check_results, created_at, updated_at.
- **Checking**: checking_id (PK), booking_id (FK), passenger_id (FK), check_results, created_at, updated_at.

Legend
- | = exactly one
- O = zero (optional)
- > = many

Combinations:
- |—| = one to one (1 : 1)
- |—> = one to many (1 : N)
- O—| = zero or one (0 : 1)
- O—> = zero or many (0 : N)

Conclusion
The ERD for the airport management system has been designed according to the requirements. The diagram is normalized to 3NF, attributes and relationships are clearly defined, and the cardinalities correctly represent the business rules of the system.