## CS 306- PROJECT PHASE TWO REPORT

#### Introduction:

In this report, the newly added components to the SQL system and backhand codes will be explained. They are inspected by their functionality, how they affected the system and importance for the process. Furthermore, each component will be explained by their changes on the database system. The changes will be supported and visualized by the screenshots taken before and after of the executions. Lastly the css, php and html codes will be shown as well as trigger and procedure sql files.

## **Trigger Implementation and Testing:**

In this phase of the project we have added two trigger models; check\_seat\_availabilty and check\_flight\_time\_overlap.

The aim of the "check\_seat\_availabilty" trigger is allowing or preventing one from operating one already booked seat. The process follows the steps:

- 1- Creates a boolean to store whether the seat is already in Belongs to table
- 2- Traverse the table to detect any matching between the input variables and the table.
- 3- If there is a match, the boolean turns to true and an error is raised to alert the person that this seat is already taken.

```
CREATE TRIGGER check_seat_availability

BEFORE INSERT ON Belongs_to

FOR EACH ROW

BEGIN

DECLARE seat_already_booked BOOLEAN;

SELECT

CASE

WHEN EXISTS (SELECT 1 FROM Seats WHERE seat_number = NEW.seat_number AND f_id = NEW.f_id AND is_booked = TRUE) THEN TRUE

ELSE FALSE

END

INTO seat_already_booked

FROM dual;

If seat_already_booked THEN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Seat is already booked.';

END IF;

END //
```

It is observed that the example command resulted in an error since 12B is entered to Belongs\_to table already.

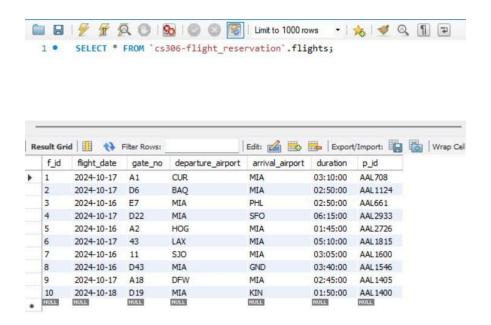
The trigger "check\_flight\_time\_overlap" ensures that, when a new flight is added, the plane does not have another overlapping flight given the certain time interval of the plane's flight duration. This trigger secures the integrity of the flight schedule, and automates conflict detection while scheduling.

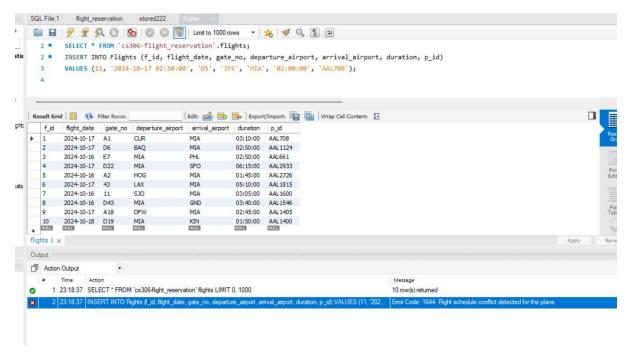
The trigger calculates the end time by using flight time and duration, then checks for conflicts from the Flights table by matching the p\_id and raises an error if the conflict is found.

Below are the SQL scripts and the implementation of the check flight time overlap trigger:

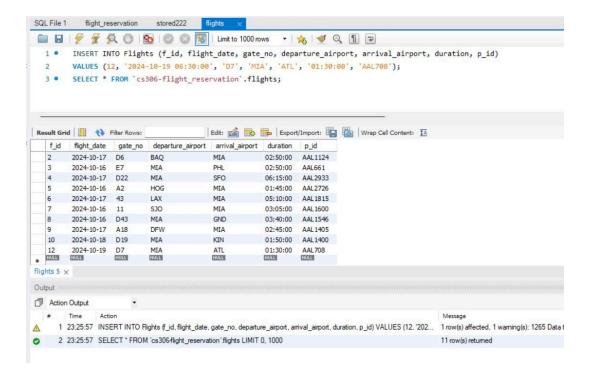
```
- | 🏡 | 🥩 🔍 🗻 🖃
215
       DELIMITER //
217 • CREATE TRIGGER check_flight_time_overlap
218
       BEFORE INSERT ON Flights
219
      FOR EACH ROW
220
       BEGIN
221
         DECLARE conflict_count INT;
222
          DECLARE new_end_time DATETIME;
          -- Calculate the end time of the new flight
223
         SET new_end_time = NEW.flight_date + INTERVAL TIME_TO_SEC(NEW.duration) SECOND;
224
          -- Check for overlapping flights with the same plane
225
          SELECT COUNT(*)
226
227
          INTO conflict count
           FROM Flights
228
229
           WHERE p id = NEW.p id
230
231
                (NEW.flight_date < (flight_date + INTERVAL TIME_TO_SEC(duration) SECOND))
                (flight_date < new_end_time)
234
235
236
           IF conflict_count > 0 THEN
237
             SIGNAL SQLSTATE '45000'
              SET MESSAGE_TEXT = 'Flight schedule conflict detected for the plane.';
238
239
           END IF:
240
       END //
241
       DELIMITER ;
242
```

Flights table before and after the trigger implementation:





The example command resulted in an error since there is another flight for the plane with p\_id=11 for the inserted duration of flying time.



## **Stored Procedure Implementation:**

We have implemented two stored procedure implementations; SearchFlights and book\_seat.

The aim of the "book\_seat" procedure is to successfully book an available seat from the Seats table and update the required table status. The procedure happens as following:

- 1- The seat existence and availability is checked
- 2- If the seat is bookable, the status of is\_booked converted to true for specified seat in Seats table
- 3- Then the new booking is added to Belongs\_to table
- 4- The transactions are committed sequentially with the end of procedures.

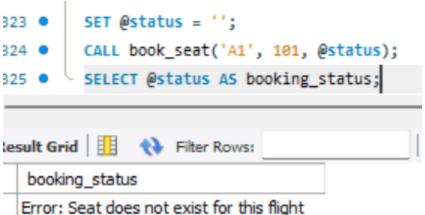
```
    CREATE PROCEDURE book_seat(

        IN p_seat_number VARCHAR(10),
       IN p_flight_id INT,
        OUT p_status VARCHAR(100)
   - )

    BEGIN

        -- Declare variables for checking
       DECLARE seat_exists INT;
       DECLARE is_booked BOOLEAN;
        -- Check if seat exists for the flight
        SELECT COUNT(*) INTO seat_exists
        FROM Seats
        WHERE seat_number = p_seat_number AND f_id = p_flight_id;
    IF seat_exists = 0 THEN
        SET p_status = 'Error: Seat does not exist for this flight';
    ELSE
        -- Check if seat is already booked
        SELECT is_booked INTO is_booked
        FROM Seats
        WHERE seat_number = p_seat_number AND f_id = p_flight_id;
        IF is_booked = TRUE THEN
            SET p_status = 'Error: Seat is already booked';
        ELSE
```

```
IF is_booked = TRUE THEN
        SET p_status = 'Error: Seat is already booked';
ELSE
        START TRANSACTION;
        UPDATE Seats
        SET is_booked = TRUE
        WHERE seat_number = p_seat_number AND f_id = p_flight_id;
        INSERT INTO Belongs_to (seat_number, f_id)
        VALUES (p_seat_number, p_flight_id);
        -- If we get here, both operations succeeded
        COMMIT;
        SET p_status = 'Success: Seat booked successfully';
        END IF;
        END IF;
```



As it can be seen from the procedure piece that the flight doesn't own an A1 seat so that the variables raised an error to alert the person.

The "SearchFlights" stored procedure retrieves flights based on the departure airport, arrival airport, and flight date. This allows users to efficiently search for available flights matching their criteria.

Below are the SQL scripts of the stored procedures and their retrieval examples:

```
DELIMITER $$
   DROP PROCEDURE IF EXISTS SearchFlights $$

    ○ CREATE PROCEDURE SearchFlights(
       IN dep_airport VARCHAR(50), -- Input: Departure airport
       IN arr_airport VARCHAR(50), -- Input: Arrival airport
       IN flight date DATE
                                     -- Input: Flight date
   BEGIN
       -- Select flights based on the input parameters
       SELECT f_id, flight_date, gate_no, departure_airport, arrival_airport, duration
       FROM Flights
       WHERE departure_airport = dep_airport
       AND arrival airport = arr airport
       AND flight date = flight date;
   END $$
   DELIMITER ;
        33
               CALL SearchFlights('CUR', 'MIA', '2024-10-17');
        35
        36
       Result Grid Filter Rows:
                                            Export: Wrap Cell Content: IA
                flight_date
                           gate_no departure_airport arrival_airport duration
         1
                2024-10-17 A1
                                                               03:10:00
```

#### Web Access Module:

There are two web access modules for users to interact with the database, one for checking whether a flight is available by entering departure and arrival airports with the flight date. Users then may check whether there are available unbooked seats on the flight that they have found.

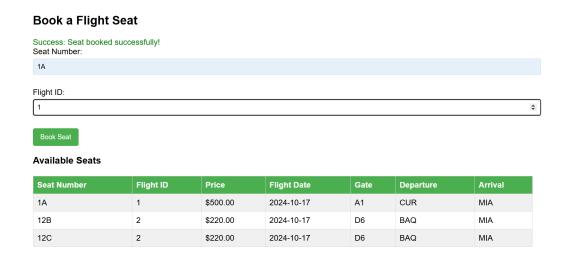
### For booking:

This html file is creating the web interference for the php file to be seen with the combination of css file. It creates two interactive input blocks for the user to enter their desired seat and flight id. This information is available under the available seats table. When the booking is successful,

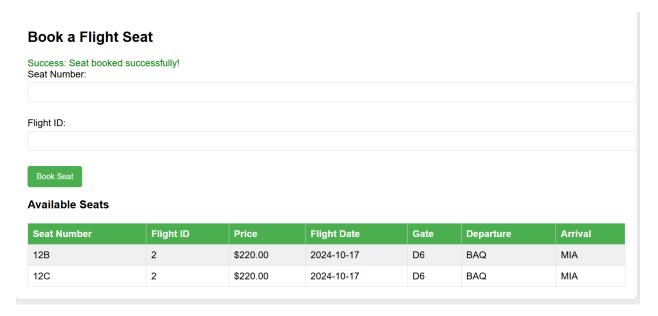
the specified seat is erased from the table. Depending on the choice, system can raise the following messages:

- 1- Successful booking
- 2- Not available seat
- 3- There is no such seat for that flight

Before selecting the seat:



After booking:



For seat table:

Before booking:

| Seat Number | Flight ID | ls Booked | Price    | Near Emergency Exit |
|-------------|-----------|-----------|----------|---------------------|
| 12A         | 2         | Yes       | \$220.00 | No                  |
| 12B         | 2         | No        | \$220.00 | No                  |
| 12C         | 2         | No        | \$220.00 | No                  |
| 13A         | 3         | Yes       | \$180.00 | Yes                 |
| 13B         | 3         | Yes       | \$180.00 | Yes                 |
| 13C         | 3         | Yes       | \$180.00 | Yes                 |
| 14A         | 4         | Yes       | \$100.00 | No                  |
| 15A         | 5         | Yes       | \$300.00 | Yes                 |
| 1A          | 1         | No        | \$500.00 | Yes                 |
| 1B          | 1         | Yes       | \$500.00 | Yes                 |

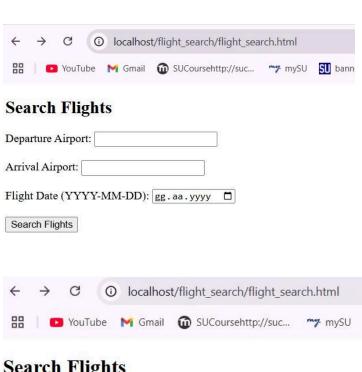
## After booking:

| Seat Number | Flight ID | ls Booked | Price    | Near Emergency Exit |
|-------------|-----------|-----------|----------|---------------------|
| 12A         | 2         | Yes       | \$220.00 | No                  |
| 12B         | 2         | No        | \$220.00 | No                  |
| 12C         | 2         | No        | \$220.00 | No                  |
| 13A         | 3         | Yes       | \$180.00 | Yes                 |
| 13B         | 3         | Yes       | \$180.00 | Yes                 |
| 13C         | 3         | Yes       | \$180.00 | Yes                 |
| 14A         | 4         | Yes       | \$100.00 | No                  |
| 15A         | 5         | Yes       | \$300.00 | Yes                 |
| 1A          | 1         | Yes       | \$500.00 | Yes                 |
| 1B          | 1         | Yes       | \$500.00 | Yes                 |
|             |           |           |          |                     |

## For flight checking:

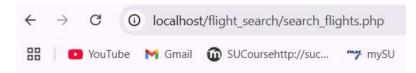
The web interface allows users to search for flights by entering departure and arrival airports with the flight date. The results based on the form submission are gathered from the database and displayed on the result page.

Below are the screenshots of the input form, insertion confirmation and data view.



# Search Flights

Departure Airport: CUR Arrival Airport: MIA Flight Date (YYYY-MM-DD): 19.10.2024 Search Flights



Input Date: 2024-10-19

Departure Airport: CUR

Arrival Airport: MIA

No flights found for the given parameters.



### Back-end HTML and PHP codes:

```
<!DOCTYPE html>
    <title>Flight Seat Booking System</title>
   <link rel="stylesheet" href="seat.css">
    <div class="container">
       <h2>Book a Flight Seat</h2>
       <?php if ($message) echo $message; ?>
       <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
           <div class="form-group">
                <label for="seat number">Seat Number:</label>
                <input type="text" id="seat_number" name="seat_number" required>
           <div class="form-group">
               <label for="flight_id">Flight ID:</label>
               <input type="number" id="flight_id" name="flight_id" required>
            <input type="submit" value="Book Seat">
       </form>
       <h3>Available Seats</h3>
```

```
<h3>Available Seats</h3>
      Seat Number
      Flight ID
      Price
      Flight Date
      Gate
      Departure
      Arrival
   if ($available_seats) {
      while($row = $ Convert special characters to HTML entities
   ctr class="clickab" htmlspecialchars( string $string [, int $flags = ENT_COMPAT | ENT_HTML401 [, string $encoding =
      <\td><?php echo ini_get("default_charset") [, bool $double_encode ]]]): string
      <?php echo htmlspecialchars($row['f_id']); ?>
      $<?php echo htmlspecialchars($row['price']); ?>
      <?php echo htmlspecialchars($row['flight_date']); ?>
      <?php echo htmlspecialchars($row['gate_no']); ?>
<?php echo htmlspecialchars($row['departure_airport']); ?>
      <?php echo htmlspecialchars($row['arrival_airport']); ?>
```

#### For seat table:

This html file shows the seat table variables in a table format.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Flight Seats Information</title>
   <link rel="stylesheet" href="table.css">
   <div class="container">
       <h1>Flight Seats Information</h1>
       <div class="table-container">
          <h2>Seats Table</h2>
              <thead>
                      Seat Number
                      Flight ID
                      Is Booked
                      Price
                      Near Emergency Exit
              </thead>
                  if ($seats result) {
                      while($row = mysqli_fetch_assoc($seats_result)) {
```

```
</thead>

<?php
if ($seats_result) {
    while($row = mysqli_fetch_assoc($seats_result)) {
        echo "<tr>        "echo "        ""        "        "        "
        "
        echo ""
        "
        echo "
```

The php files give the dynamic procedure to the database systems. There are two php files for the seat checking and booking process which obtain data from the system and put it on the web server. One for seat availability check and one for the current conditions of the seat table itself.

There is another php file for the flight checking which checks whether there exists a flight based on given conditions.

For displaying seat table:

```
<?php
// Include the configuration file.
require_once 'config.php';

// Query to get seats data
$seats_query = "SELECT * FROM Seats";
$seats_result = mysqli_query($db, $seats_query);

?>
```

For evaluating desired seat from available seat table:

```
include 'config.php';
$message = '';
$status = '';
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $seat_number = $_POST['seat_number'];
   $flight_id = $_POST['flight_id'];
    // First check if the seat exists and is available
    $check_query = "SELECT is_booked FROM Seats WHERE seat_number = ? AND f_id = ?";
   $check_stmt = $db->prepare($check_query);
   $check_stmt->bind_param("si", $seat_number, $flight_id);
   $check_stmt->execute();
   $result = $check_stmt->get_result();
   if ($result->num_rows === 0) {
       $message = '<div style="color: red;">Error: Seat does not exist for this flight.</div>';
        $seat_data = $result->fetch_assoc();
        if ($seat_data['is_booked']) {
           $message = '<div style="color: red;">Error: This seat is already booked.</div>';
           $db->begin_transaction();
```

```
try {
    // Update the seat status
    $update_query = "UPDATE Seats SET is_booked = TRUE WHERE seat_number = ? AND f_id = ?";
    $update_stmt = $db->prepare($update_query);
    $update_stmt->bind_param("si", $seat_number, $flight_id);
    $update_stmt->execute();
    $belongs_query = "INSERT INTO Belongs_to (seat_number, f_id) VALUES (?, ?)";
    $belongs_stmt = $db->prepare($belongs_query);
    $belongs_stmt->bind_param("si", $seat_number, $flight_id);
    $belongs_stmt->execute();
    // If we got here, commit the transaction
    $db->commit();
    $message = '<div style="color: green;">Success: Seat booked successfully!</div>';
} catch (Exception $e) {
    // If there was an error, roll back the transaction
    $db->rollback();
    $message = '<div style="color: red;">Error: ' . $e->getMessage() . '</div>';
if (isset($update_stmt)) $update_stmt->close();
if (isset($belongs_stmt)) $belongs_stmt->close();
```

```
$check_stmt->close();
// Query to get all available seats with additional information
$available_seats_query = "
   SELECT
       s.seat_number,
       s.f_id,
       s.price,
       s.is_booked,
       f.departure_airport,
       f.arrival_airport,
       f.flight_date,
       f.gate_no
   FROM Seats s
   JOIN Flights f ON s.f_id = f.f_id
   WHERE s.is_booked = FALSE
   ORDER BY f.flight_date, s.f_id, s.seat_number";
$available_seats = $db->query($available_seats_query);
if (!$available_seats) {
    $message = '<div style="color: red;">Error retrieving available seats: ' . $db->error . '</div>';
```

## For flight checking process:

```
e config.php
C: > xampp > htdocs > flight_search > ◆ flight_search.html > ...
  1 <!DOCTYPE html>
      <html lang="en">
          <meta charset="UTF-8">
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
          <title>Flight Search</title>
          <h2>Search Flights</h2>
          <form method="POST" action="search_flights.php">
              <label for="departure_airport">Departure Airport:</label>
              <input type="text" id="departure_airport" name="departure_airport" required>
              <label for="arrival_airport">Arrival Airport:</label>
              <input type="text" id="arrival_airport" name="arrival_airport" required>
              <label for="flight_date">Flight Date (YYYY-MM-DD):</label>
              <input type="date" id="flight date" name="flight date" required>
              <button type="submit">Search Flights</button>
          </form>
```

```
C: > xampp > htdocs > flight_search > ** search_flights.php
      include "config.php";
      $form submitted = false;
      if ($_SERVER["REQUEST_METHOD"] == "POST") {
           $departure_airport = $_POST['departure_airport']; // departure airport
          $arrival_airport = $_POST['arrival_airport']; // arrival airport
$input_date = $_POST['flight_date']; // flight date in YYYY-MM-DD format
          $form submitted = true;
          echo "Input Date: " . $input date . "";
          echo "Departure Airport: " . $departure_airport . "";
echo "Arrival Airport: " . $arrival_airport . "";
          $sql = "SELECT * FROM Flights WHERE flight_date = '$input_date'
                   AND departure_airport = '$departure_airport'
                  AND arrival_airport = '$arrival_airport'";
          $result = mysqli_query($db, $sql);
           if (!$result) {
              die("Query failed: " . mysqli error($db));
           if (mysqli_num_rows($result) > 0) {
               echo "<h3>Flights Found:</h3>";
               echo "
                           Flight ID
                           Departure Airport
                           Flight Date
                           Duration
                           Gate Number
```

```
C: > xampp > htdocs > flight_search > fli
```

```
echo "
                            Flight Date
                            Duration
                            Gate Number
                        ";
               while ($row = mysqli fetch assoc($result)) {
                   echo "
                            " . $row['f_id'] . "
                            . $row[ f_1d ] . 

. $row['departure_airport'] . "

. $row['arrival_airport'] . "

. $row['flight_date'] . "

. $row['duration'] . "

. $row['gate_no'] . "

49
              echo "";
               echo "No flights found for the given parameters.";
     <!DOCTYPE html>
     <html lang="en">
          <meta charset="UTF-8">
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
          <title>Flight Search</title>
```

```
C:) xampp > htdocs > flight_search > flights.php

// CIDUCITYE TUBL:

// CHARD | Character | Content | Con
```