CS 353: Database Systems

Final Report

Travel Agency Management System

Ahmet Salman, 21901004

Muhammed Can Küçükaslan, 21901779

Mustafa Yasir Altunhan, 21802530

Güven Gergerli, 21803393

4/1/2022

Course Instructor: Özgür Ulusoy

1.0 Description of the System

The name of our system is Levart, it is a travel agency management system that will be used by customers, employees, and tour guides. It aims to simplify the process of planning a holiday or vacation for the users.

Levart allows the users to create accounts so they can access the other features of the system.

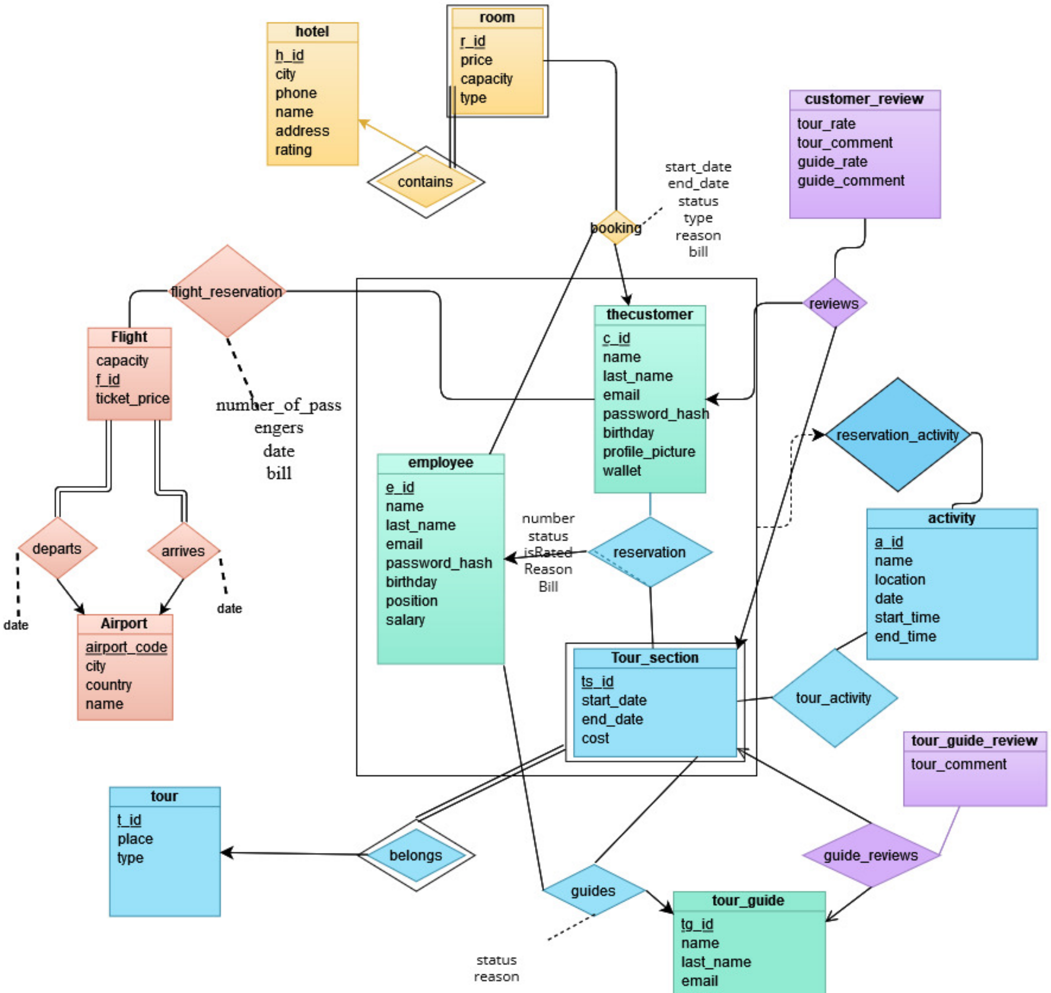
After being authenticated, users will be able to view all currently available tours and filter based on the date they are available, the user can also see the tour details such as the tour guide responsible for this tour and the basic activities, as well as possible extra activities that could be individually registered. After choosing a tour they like, they can send a reserve request, which will be viewed by an employee and either rejected or approved. Assuming it was accepted, the tour will include all the basic activities by default, then the user has the option to add extra activities. The system also allows the users to reserve hotel rooms, which go through the same approval process as tour reservations. Finally, the user can also view all available flights and filter by city departure and destination.

From the employees' point of view, he/she can approve/reject hotel and tour reservations of users, if the request is rejected they must provide a reason, they also can create new tours and send assignment requests to tour guides.

Finally, the tour guide can view currently available assigned tours and accept/reject them.

2.0 Final E/R Model

- Activity type (basic/extra) is moved to tour_activity table
- The customer has a wallet(money/budget) attribute.
- Removed type from activity
- It seems "end_time" and "start_time" attributes are added to the "activity" table.
- Removing tgr_id from tour_guide_review table
- Added bill attributes to reservation, and booking tables
- Added "reason" for rejections to reservation,booking, and guides table
- Aggregate “employee, thecustomer, tour_sesction and reservation” relation to establish a relation with the reservation_activity.



3.0 Final List of Tables

3.1 hotel

Relational Model:
hotel(h_id, phone, name, city, address, rating)
Keys:

- PK: {h_id}
- FK: none

3.2 room

Relational Model:
room(r_id, h_id, price, capacity, type,)
Keys:

- PK: {r_id, h_id}
- FK: {(h_id)}

3.3 thecustomer

Relational Model:
thecustomer(c_id, name, last_name, email, password_hash, wallet, birthday, profile_picture)
Keys:

- PK: {c_id}
- FK: none

3.4 employee

Relational Model:
employee(e_id, name, last_name, email, password_hash, birthday, salary, position)
Keys:

- PK: {e_id}
- FK: none

3.5 tour_guide

Relational Model:

tour_guide(tg_id, name, last_name, email, password_hash, birthday, registration, profile_picture)
Keys:

- PK: {tg_id}
- FK: none

3.6 tour

Relational Model:
tour(t_id, place, type)
Keys:

- PK: {t_id}
- FK: none

3.7 activity

Relational Model:
activity(a_id, name, location, date, start_time, end_time)
Keys:

- PK: {a_id}
- FK: none

3.8 airport

Relational Model:
airport(airport_code, name, city, country)
Keys:

- PK: {airport_code}
- FK: none

3.9 tour_section

Relational Model:
tour_section(ts_id, t_id, start_date, end_date, cost)
Keys:

- PK: {(ts_id, t_id)}
- FK: {(t_id)}

3.10 customer_review

Relational Model:

customer_review(c_id, ts_id, t_id, tour_rate, tour_comment, guide_rate, guide_comment)

Keys:

- PK: {c_id, ts_id, t_id}
- FK: {(c_id), (ts_id, t_id)}

3.11 tour_activity

Relational Model:

tour_activity(ts_id, a_id,t_id, type, cost)

Keys:

- PK: {(ts_id, t_id, a_id)}
- FK: {(ts_id, t_id), (a_id)}

3.12 booking

Relational Model:

booking(c_id, r_id, h_id, e_id, start_date, end_date, type, status, reason, bill)

Keys:

- PK: {(c_id, r_id, h_id, e_id)}
- FK: {(c_id), (r_id, h_id), (e_id)}

3.13 reservation

Relational Model:

reservation(c_id, ts_id, t_id, e_id, number, status, isRated, reason, bill)

Keys:

- PK: {(c_id, ts_id, t_id, e_id)}
- FK: {(c_id), (ts_id, t_id), (e_id)}

3.14 reservation_activity

Relational Model:

reservaction_activity(c_id, ts_id, t_id, e_id, a_id)

Keys:

- PK: {(c_id, ts_id, t_id, e_id, a_id)}
- FK: {(c_id, ts_id, t_id, e_id),(a_id)}

3.15 tour_guide_review

Relational Model:

tour_guide_review(t_id, ts_id, tg_id, tour_comment)

Keys:

- PK: {(t_id, ts_id, tg_id)}
- FK: {(tg_id), (ts_id, t_id)}

3.16 guides

Relational Model:

guides(tg_id, ts_id, e_id, status, reason)

Keys:

- PK: {(tg_id, ts_id, e_id)}
- FK: {(tg_id), (ts_id, t_id), (e_id)}

3.17 flight

Relational Model:

flight(f_id, capacity, ticket_price, dept_airport, dest_airport, dept_date, arrive_date)

Keys:

- PK: {f_id}
- FK: {(dept_airport), (dest_airport)}

3.18. flight_reservation

Relational Model:

flight_reservation(f_id, c_id, number_of_passengers, date, bill)

Keys:

- PK: {f_id, c_id}
- FK: {(f_id), (c_id)}

4.0 Implementation Details

In the project, we used the Apache web server and MariaDB 10.4.* as Database Management System.

We developed the web application using PHP 8.0.* and we have chosen the PHP interpreter respectively.

We used both the mysqli[1] and PHP Data Objects (PDO)[2] to connect the database. To avoid repetitive code for database connection we have written a configuration file where the database connection is established and returned to where that file is included (or *required*).

For user interface design we've used the Html5, CSS and JavaScript. We also used *bootstrap* and *jquery* libraries for CSS and JavaScript.

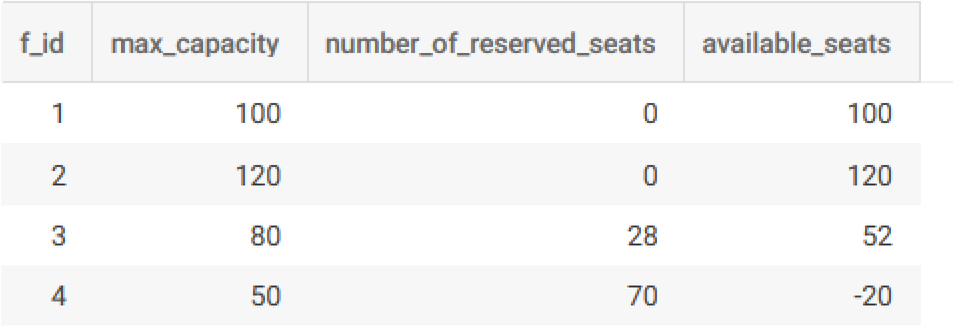
5.0 Advanced DB Components

The original queries used for these components are given in the appendix.

5.1 View

flight_capacity

We created a *flight_capacity* view that has the columns *f_id, max_capacity, number_of_reserved_seats, available_seats*. This view enables us to easily obtain the number of available seats of a flight.



f_id	max_capacity	number_of_reserved_seats	available_seats
1	100	0	100
2	120	0	120
3	80	28	52
4	50	70	-20

[1] https://www.php.net/manual/en/book.mysqli.php

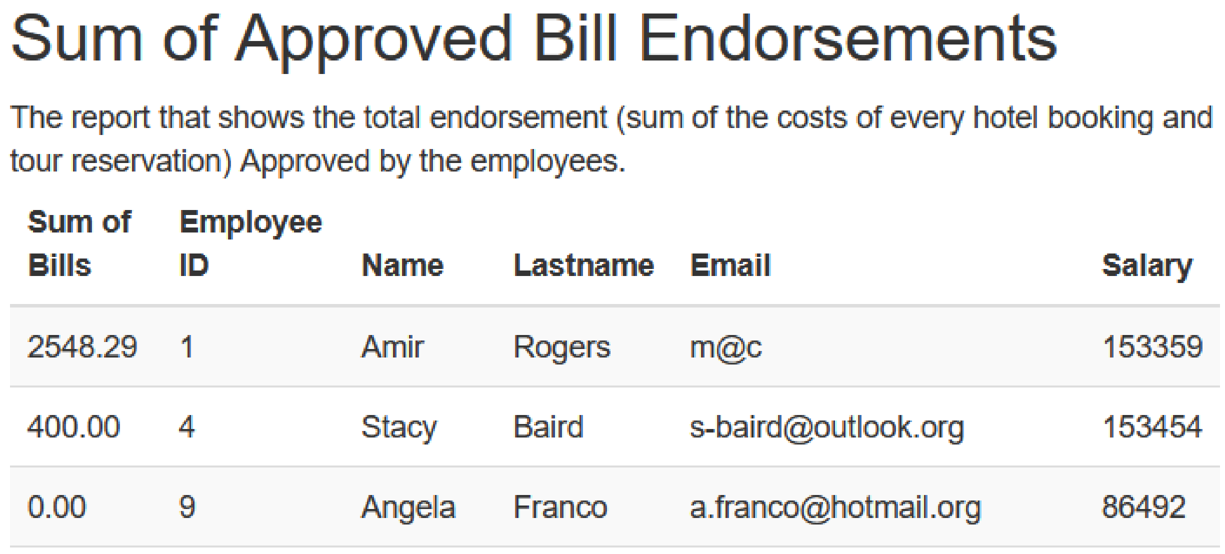
[2] https://www.php.net/manual/en/book.pdo.php

5.1 Stored Procedures

sum_of_bill_endorsement

This stored procedure is used to generate the report that shows the total endorsement (sum of the costs of every hotel booking and tour reservation) approved or rejected by the employees. The definition of the stored procedure is (including create statement):

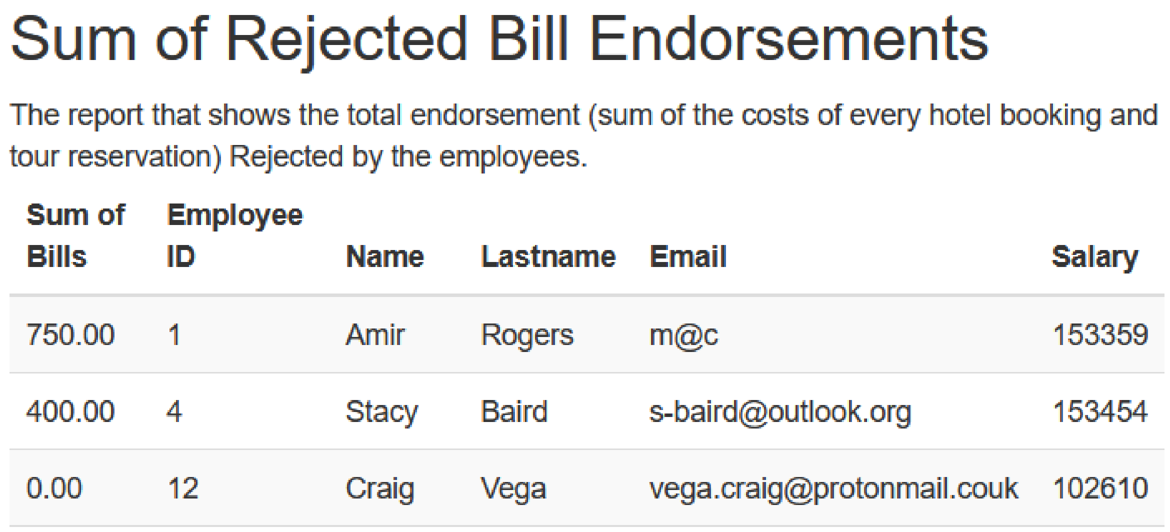
Sample Output:



Sum of Approved Bill Endorsements

The report that shows the total endorsement (sum of the costs of every hotel booking and tour reservation) Approved by the employees.

Sum of Bills	Employee ID	Name	Lastname	Email	Salary
2548.29	1	Amir	Rogers	m@c	153359
400.00	4	Stacy	Baird	s-baird@outlook.org	153454
0.00	9	Angela	Franco	a.franco@hotmail.org	86492



Sum of Rejected Bill Endorsements

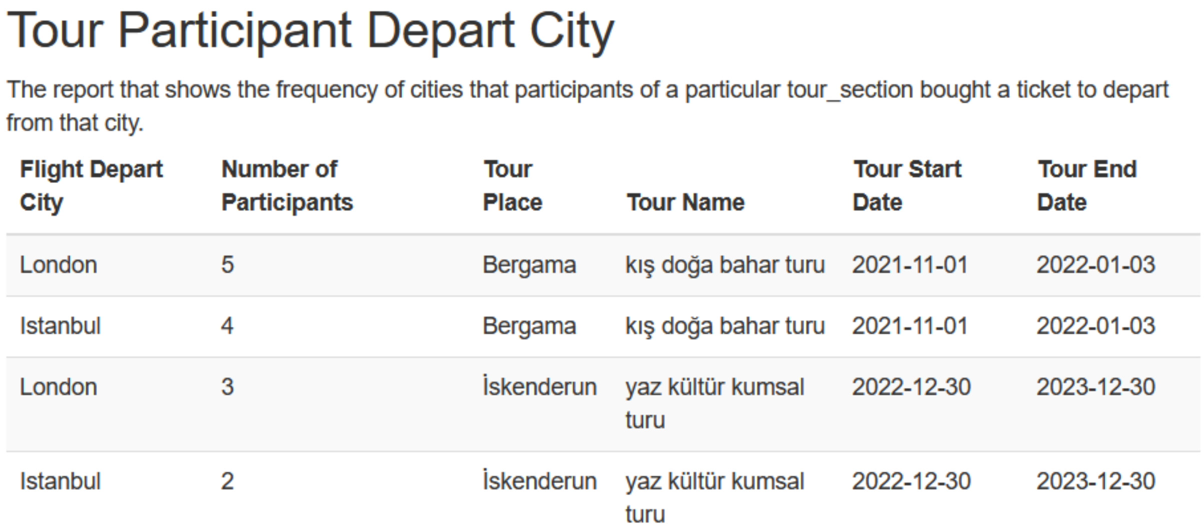
The report that shows the total endorsement (sum of the costs of every hotel booking and tour reservation) Rejected by the employees.

Sum of Bills	Employee ID	Name	Lastname	Email	Salary
750.00	1	Amir	Rogers	m@c	153359
400.00	4	Stacy	Baird	s-baird@outlook.org	153454
0.00	12	Craig	Vega	vega.craig@protonmail.couk	102610

tour_participant_depart_city

This stored procedure is used to generate a report that shows the frequency of cities that participants of a particular tour_section bought a ticket to depart from that city.

Sample Output



Tour Participant Depart City

The report that shows the frequency of cities that participants of a particular tour_section bought a ticket to depart from that city.

Flight Depart City	Number of Participants	Tour Place	Tour Name	Tour Start Date	Tour End Date
London	5	Bergama	kış doğa bahar turu	2021-11-01	2022-01-03
Istanbul	4	Bergama	kış doğa bahar turu	2021-11-01	2022-01-03
London	3	İskenderun	yaz kültür kumsal turu	2022-12-30	2023-12-30
Istanbul	2	İskenderun	yaz kültür kumsal turu	2022-12-30	2023-12-30

6.0 User Manual

Registration

When logging into the Levart for the first time as a user, the user must create an account, if the user does not have an existing account. Before creating an account, the user must specify the type of user account. The types of accounts can be selected with the drop-down menu. There exists three: Customer, Employee, and Tour Guide. Registration section can be found by clicking the “Not a user? Sign up” hypertext.

The Customer type is created for the purpose of using the traveling system as a customer. The customer type of user, in brief, can book or cancel tours, hotels, and flights. If the owner of the account has the purpose of using the application with the purpose of benefiting from the service, the user must create a customer type of account by filling in the required data. These data are; Name, Surname, E-mail, Password, and Birthday.

The Employee type is created for the purpose of serving the traveling system as an employee. The employee type of user, in brief, can create tours and confirm tour, hotel, and flight reservations. If the owner of the account has the purpose of using the application with the purpose of contributing to the service, the user must create an employee type of account by filling in the required data. These data are; Name, Surname, E-mail, Password, and Birthday.

The Tour Guide type is created with the purpose of serving the traveling system as a tour guide. The tour guide type of user, in brief, can sign up to guide existing tours. If the owner of the account has the purpose of using the application with the purpose of contributing to the service as a guide, the user must create a tour guide type of account by filling in the required data. These data are; Name, Surname, E-mail, Password, and Birthday.

Login

In order to access the website do operations as either customer, employee, or tour guide; the user must log in. First, the user needs to specify the type of the account from the drop-down menu. After that user must enter the email and password of the account and after that click on the login button. If the user does not have an existing account, the user must first do the registration process to create an account before logging in.

Dashboard

When the user logs in, the website redirects the user to the dashboard page. The dashboard page consists of general information specified and organized to the user which is some kind of a home page for users. On the dashboard page, users can see general information about tours, hotels, and flights. Specifically, customer types of users can see reserved tours, reserved hotels, and buy tickets. In addition, customers can see the details of tours and hotels as well as cancel them from the dashboard. Employee types of users can see Incoming Tours and Available Hotels. In addition, employees can see the details of tours and hotels as well as cancel them if they find it necessary. Tour Guide types of users can see Completed Tours in the dashboard. In addition, tour guides can see the details of the completed tours as well as give feedback on tours.

Tours

Tours are the foundation of the system and the website. Every type of user can interact with the tours in many different ways, which will be explained in detail.

Customers use the service of tours as a reservation system. First of all, in order to book or reserve a tour, the customer needs to go to the “Book Tour” page. On this page, customers can see the tours that are available to book. On that page, the customer also can see the details of the tour, visit the tour guide’s profile, reserve the tours and add extra activities. When clicked into the reserve button near the tour, the customer reserves the tour which can be seen in the dashboard afterward. When the tours are over and finished, the customer can see the tours on the “Past Tours” page. On that page, customers can see the details of the tour and can go to the tour guide profile, and can rate the tour. When the Rate tour button is clicked, the customer is redirected to the rating page. The page has point and open-ended questions for users to rate both the tour and the tour guide. Note that the tour can only be rated once.

Tours are created by the employees from the “Create New Tour” page. The employee enters the name, start-end dates, location, and other details about the tour in order to create a tour. Employees also need to assign a tour guide to an existing tour in order for the tour to be reserved by the customers. If data needs to be changed from a tour, employees can edit the tours by the Edit button next to tours. At last, employees need to accept tours from the “Pending Tours” page.

Tour guides can assign themselves to tours from the “Available Tours” page if they find it to be suitable for them. Also after completing the tours tour guides can give feedback to tours by clicking the “Feedback” button.

Tour Activities

Tour Activities are selectable objects that are assigned to tours in order to give more depth, explanation, and user choice into tours. There are two types of activities which are “basic” and “extra” activities. Basic activities come from the tour itself and cannot be removed by the customer. However, extra activities can be selected or removed by the customer. The Activities of a tour can be found on the details page which can be found by the details button near the tour. Customers can do operations on the tours that have not started yet and cannot do operations on past tours. Addition and removal of extra activities can also be done on the details page via buttons, namely “Add Extra Activity” and “Remove Extra Activity”. Both “basic” and “extra” activities are created by the employees. While creating an activity, an employee must specify whether the activity is a fix or extra activity.

Hotels

Hotels are establishments providing accommodation and other services for travelers and tourists. A customer without the dependency of a tour can reserve hotels for themselves or more people. The reservation of hotels can be done on the “Reserve a Hotel” page. On this page, customers can find all hotels that are available to reserve. The hotels are listed with essential details such as hotel name, hotel location, the available number of rooms, and similar details. Moreover, the customer can also filter the hotels that are listed in the system by his/her wish. In order for a customer to reserve a hotel, the customer must specify the type of room they want from the drop-down list and the start and end dates of the hotel accommodation. The prices of rooms can also be found in the drop-down menu. Once the hotel is reserved, the customer can delete the reservation from the “Reserved Hotels” section in the dashboard. Also, the details of the hotel reservations can be found on the “Hotel Reservation Details” page which can be found by the button next to the hotel reservation in the “Hotel Reservations” section on the Dashboard.

The hotels and the rooms of the hotel are registered to the system by the employees. An employee can register a new hotel and its rooms from the “Register Hotel” page. On the register hotel page, employees must fill in all the data about the hotel in the page. In addition, Rooms of the hotel must be added to the hotel on the same page by specifying the name of the room and price of the room. When a user makes a reservation for a hotel, employees need to accept or deny the reservation by the “Accept” and “Deny” buttons near the hotel reservations which can be found on the “Pending Hotel Reservations” page. On this page, the data about the hotel as well as the reservation, such as the room selected, the start and end dates of the hotel, and the customer who made the reservation. Additional details can be found by clicking the details button which will redirect the employee to the “Hotel Reservation Details” page.

Flights

Customers in addition to reserving tours and hotels in the traveling system can also reserve flights in the system. In order for a customer to reserve a flight, the customer must go to the “Reserve a Flight” page. On the page, the customer can see all available flights one after another. Moreover, the customer can also filter the flights that are listed in the system by his/her wish. The flights are listed with essential details for the customers such as arrival and departure location, arrival and departure date, arrival and departure airport name, price per person, and flight number. In order to reserve a flight, the customer must specify the number of passengers. After the flight reservation is complete, the flight reservation can be found in the dashboard under the section “Your Flights”. Here the details about the flight can be found as well as cancellation of flight reservation can be done by clicking the “Cancel Reservation” button near the flight reservation.

Profiles

Every type of user, namely customers, employees, and tour guides has a profile that consists of details of the user. A user can visit his/her profile by clicking the “Profile” button. On the “Profile” page, a user can update his/her email as well as password by entering the new password, and confirming it. After an action of changing, the user needs to click on the “Update Email” or “Update Password” button.

For customers, the details of a tour guide are essential. That is because a customer can find details of a tour guide from the button named “Tour Guide Profile” near the tour reservation. When clicked, the user will be redirected into the profile of the tour guide which consists of details of a tour guide. In addition, the points of ratings the tour guide gained from tour feedbacks can also be seen on the tour guide’s profile page.

Employees also need to visit both customers’ and tour guide profiles since they have to decide on whether the customer or the tour guide is suitable for the tour or not in the “Pending Tours” section. In order to gain additional detail, the employee can visit profiles to assess to approve or refuse the tour reservation.

Payment

Payment of any action, namely tour reservation, hotel reservation, and flight reservation can be done through a virtual wallet called “Wallet”. A customer who wants to pay for a reservation of any type can spend money from their wallet. In order to load money onto the Wallet, the user must go to his/her profile and fill in the amount they wish to add and click on the button named “Add Money” which deposits the specified amount into the customer’s wallet. After the transaction is completed, the user can spend the money in the wallet on reservation payments.

7.0 Work Contributions

7.1 Ahmet Salman

- Project Proposal
  - Introduction
  - Project Description

- Design Report
  - All “customer” UI mockups (Figures 3.1 - 3.17)
  - Their respective queries

- Final Report
  - Description of The System
  - Final List of Tables

- Implementation: (Customer’s POV)
  - Customer’s Dashboard (Tour and Hotel Reservation)
  - Past Tours
  - Tour Details
  - Reserving/Cancelling Extra Activities
  - Book a Tour
  - Customer Profile
  - Tour Guide Profile (Comments and Ratings)
  - Filtering in Hotel Reservation
  - Filtering in Reserve Flight

7.2 Muhammed Can Küçükaslan

- Created Project Web Page and GitHub repository.
- Project Proposal
  - Written the functional requirements
  - Drew some parts of E/R Diagram

- Design Report
  - Revised E-R model and updated accordingly
  - Prepared the table schemas and corresponding create statements

- Final Report
  - Revised E-R model and updated accordingly
  - Advanced Database Components and Sample outputs
- Implementation

    - Initial implementation of the all database tables and relationships
    - Implementing the advanced database features such as View, Stored procedures etc.
    - Implement the tour guide pages: see future and previous tours, their details, and accept or reject the offers etc
    - Implement the “offer guideship” feature for the employee pages.

7.3 Mustafa Yasir Altunhan

- Project Proposal
    - Drew some parts of the E/R Diagram.
- Design Report
    - Revised E-R model.
- Final Report
    - Updated E-R design according to the feedback given by our TA.
    - Made revision in the final list of tables which were written by Ahmet Salman
- Implementation
  I implemented most of the classes and interface related to the employee part of the implementation. This includes all the bullet points listed below but not limited to that.
    - Profile page for the employee
    - Make a tour reservation page that makes a tour reservation for a customer
    - Extra activity for a reservation page which adds an extra activity for the specified reservation
    - Sql statements of the Tours and Guides page which include the sql statements of list of tours without guides
    - Create a new hotel page
    - Create new tour page
    - Create a room for a hotel in the system
    - Pending hotel reservations page which lists the pending hotel reservations made by the customers. They can either be accepted or rejected by specifying the reason.
    - Pending tour reservations page which lists the pending tour reservations made by the customers. They can either be accepted or rejected by specifying the reason.
    - Make a hotel reservation for a customer
    - Adding an extra activity for a specified tour.

7.4 Güven Gergerli

- Project Proposal
    - Non-Functional Requirements
    - Limitations

- Design Report
    - All “employee” and “tour guide” UI mockups (Figures 3.18 - 3.31)
    - Their respective queries

- Final Report
    - User Manual

- Implementation: (Customer’s POV)
    - Customer’s Dashboard (Flight Reservation)
    - Past Tours
    - Book a Tour
    - Tour Rating
    - Reserve Hotel
    - Reserve Flight
    - Past Tour Details
    - Helped on filtering

8.0 Appendix

8.1 flight_capacity

```
CREATE VIEW flight_capacity
AS SELECT
        `f`.`f_id` AS `f_id`,
        `f`.`capacity` AS `max_capacity`,
        IFNULL(SUM(`fr`.`number_of_passengers`),
        0) AS `number_of_reserved_seats`,
        IFNULL(
        (
        `f`.`capacity` - SUM(`fr`.`number_of_passengers`)
        ),
        `f`.`capacity`
        ) AS `available_seats`
FROM
        (
        `flight` `f`
        LEFT JOIN `flight_reservation` `fr` ON
        ((`f`.`f_id` = `fr`.`f_id`))
        )
GROUP BY
        `f`.`f_id`,
        `f`.`capacity`;
```

8.2 sum_of_bill_endorsement

```
CREATE  PROCEDURE `sum_of_bill_endorsement`(IN `reservation_status`
ENUM('approved','pending','rejected'))
SELECT
     sum_of_bills,
     e.e_id AS employee_id,
     e.name,
     e.lastname,
     e.email,
     e.salary,
     e.position
FROM
     (
     SELECT
     e_id,
     IFNULL(SUM(sum_of_bills),
     0) AS sum_of_bills
     FROM
     (
     SELECT
          e.e_id,
          IFNULL(SUM(`r`.`bill`),
          0) AS sum_of_bills,
          'reservation' AS TYPE
     FROM
          employee AS e
     LEFT JOIN reservation r ON
          r.e_id = e.e_id AND r.status = reservation_status
     GROUP BY
          e.e_id
     UNION
     SELECT
     e.e_id,
     IFNULL(SUM(`b`.`bill`),
     0) AS sum_of_bills,
     'bill' AS TYPE
     FROM
     employee AS e
     LEFT JOIN booking b ON
     b.e_id = e.e_id AND b.status = reservation_status
     GROUP BY
     e.e_id
     ) AS union_table
GROUP BY
     union_table.e_id
) the_list
LEFT JOIN employee e ON
     the_list.e_id = e.e_id
ORDER BY
     sum_of_bills
DESC
```

8.3 flight_capacity

```
SELECT
```

```
      stats.city AS `Flight Depart City`,
number_of_participants_from_that_city, stats.ts_id, t.place AS `Tour
Place`, t.type `Tour Name`, ts.start_date AS `Tour Start Date`,
ts.end_date AS `Tour End Date`
FROM
      (
      SELECT
      a.city,
      r.ts_id,
      COUNT(*) AS number_of_participants_from_that_city
      FROM
      flight_reservation fr
      NATURAL JOIN flight f LEFT JOIN airport a ON
      f.dept_airport = a.airport_code
      LEFT JOIN reservation r ON
      r.c_id = fr.c_id
      GROUP BY
      a.city,
      r.ts_id
) stats
LEFT JOIN tour_section ts ON
      stats.ts_id = ts.ts_id
LEFT JOIN tour t ON
      ts.ts_id = t.t_id
ORDER BY
      number_of_participants_from_that_city
DESC
```