Bilkent University

Department of Computer Engineering

CS 353 - Database Systems

Tour Reservation Management System

Final Report

- Group 4 -

Mahmud Sami Aydın
Halil Şahiner
Abdullah Talayhan
Emin Bahadır Tülüce

Table of Contents

1. Description of the System

In this project, an application system for tour reservation management is designed and implemented . The application provides a functional interface for a basic tour reservation business and its customers. The database system is used to store the data of customers, and the representative employees of the travel agency. Moreover it is used for storing the tours and the information associated with them. It helps us in accessing and manipulating the data through necessary queries.

The system has the following functionalities are supported for a customer:

- A customer can create a customer account..
- A customer can edit their profile data.
- A customer can see the list of past tours that they have attended.
- A customer can see the list of available tours.
- A customer can filter the list of available tours according to some tags and filtering parameters.
- A customer can cancel a tour reservation.
- A customer can see the details of their current tours.
    - A tour can have multiple days that are in the system as well.
    - A tour can list all the accommodation places, travelling routes and trip events in chronological order. It should also display properties about these listed elements (date, time, place etc.)
- A customer can have associated dependent travelers to include them in them reservations.

The system has the following functionalities are supported for a travel agency staff:

- All staff accounts are added to the system by administrator and there is no registration process for them.
- A travel agency employee can login to the system from the exclusive staff login page.
- A travel agency employee can add a tour to the system,
    - They can choose tour days that the tour have.
    - They can set the quota for the tour.
    - They can add accommodation places, travelling routes and trip events to the tour by specifying their properties (date, time, place etc.).
- A travel agency employee can cancel a whole tour.
- A travel agency employee can see the list of the customers that have made reservation to a particular tour and their details.

2. Final E/R

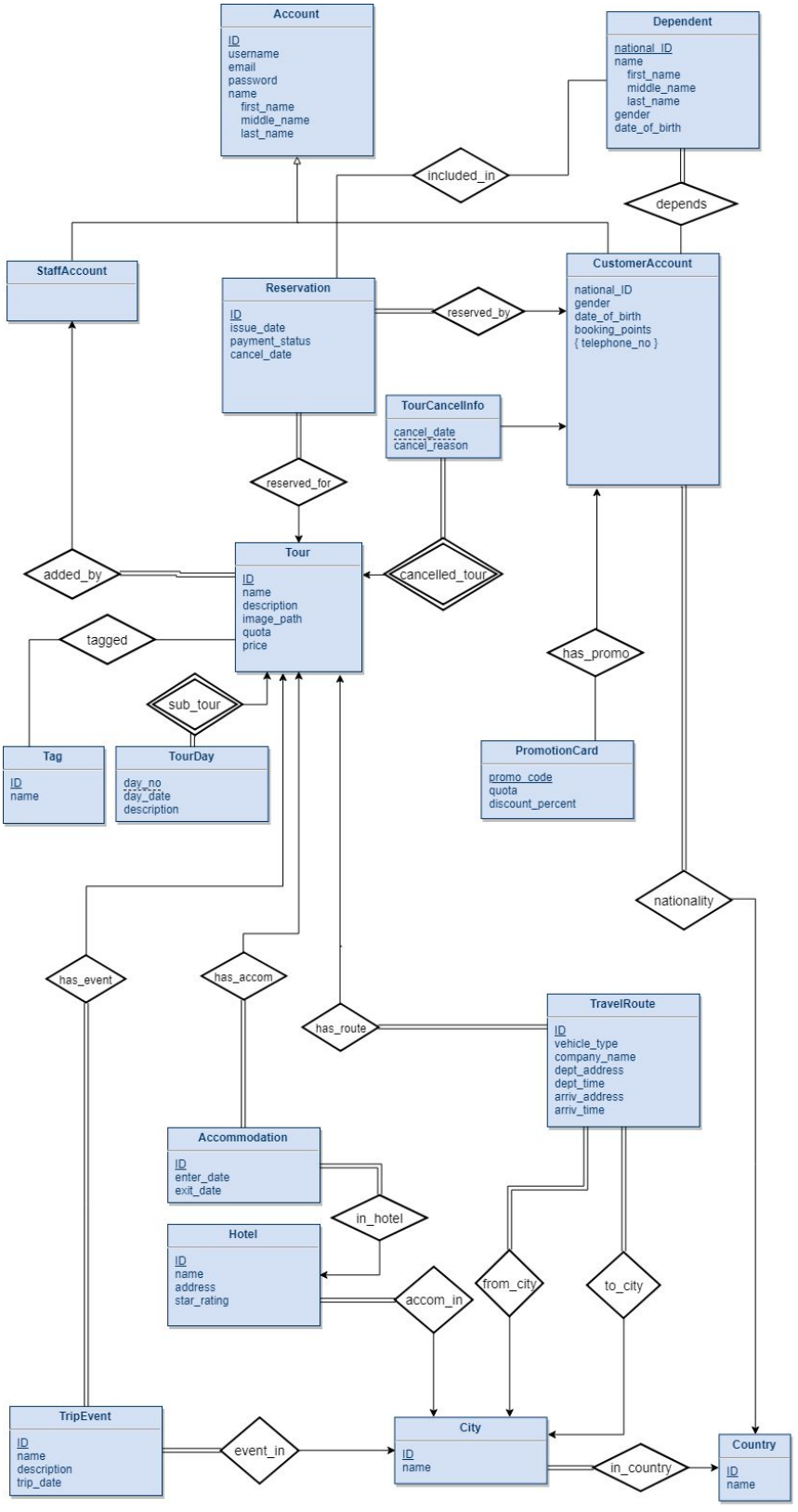


Figure 1

3. List of Tables

Account (ID, username, email, passwd, first_name, middle_name, last_name)

StaffAccount (ID)
Foreign Key (ID) references Account(ID)

CustomerAccount (ID, national_ID, nationality, gender, date_of_birth, booking_points)
Foreign Key (ID) references Account(ID)
Foreign Key (nationality) references Country(ID)

Dependent (national_ID, customer_ID, gender, date_of_birth DATE, first_name, middle_name, last_name)
Foreign Key (customer_ID) references CustomerAccount(ID)

PromotionCard (promo_code, discount_percent))

CustomerPromotionCards (promo_code, customer_ID)
Foreign Key (customer_ID) references CustomerAccount(ID)
Foreign Key (promo_code) references PromotionCard(promo_code)

CustomerTelephones (customer_ID, telephone_no)
Foreign Key (customer_ID) references CustomerAccount(ID)

Tour (ID, name, description, image_path, quota, price, creator_ID)
Foreign Key (creator_ID) references StaffAccount(ID)

Reservation (ID, customer_ID, tour_ID, issue_date, payment_status, cancel_date)
Foreign Key (customer_ID) references CustomerAccount(ID)
Foreign Key (tour_ID) references Tour(ID)

IncludedDependents (reservation_ID, dependent_ID)
Foreign Key (reservation_ID) references Reservation(ID)
Foreign Key (dependent_ID) references Dependent(national_ID)

TourCancel (tour_ID, cancel_date, cancel_reason)
Foreign Key (tour_ID) references Tour(ID)

Tag (ID, name)

TourTags (tour_ID, tag_ID)
Foreign Key (tour_ID) references Tour(ID)
Foreign Key (tag_ID) references Tag(ID)

TourDay (tour_ID, day_no day_date, description)
Foreign Key (tour_ID) references Tour(ID)

City (ID, name, country_ID)
Foreign Key (country_ID) references Country(ID)

Hotel (ID, city_ID, name, address, star_rating)
Foreign Key (city_ID) references City(ID)

Accommodation (ID, tour_ID, place_ID, enter_date, exit_date)
Foreign Key (tour_ID) references Tour(ID)
Foreign Key (place_ID) references Hotel(ID)

TripEvent (ID, tour_ID, city_ID, name, description, trip_date)
Foreign Key (city_ID) references City(ID)
Foreign Key (tour_ID) references Tour(ID)

TravelRoute (ID, vehicle_type, company_name, tour_ID, from_city_ID, to_city_ID, dept_address, dept_time, arriv_address, arriv_time)
Foreign Key (tour_ID) references Tour(ID)
Foreign Key (from_city_ID) references City(ID)
Foreign Key (to_city_ID) references City(ID)

Country (ID, name)

4. Implementation Details

4.1. Environment, Framework and Languages

In this project we have used PHP for generating the web pages in HTML and we have used PHP extensively for database queries and building the core logic of most of the functionalities. Javascript and JQuery was also used for input management and some dynamic elements. For our database, of course, we have used SQL (the MySQL version) and we have written them in separate files for a better testing, sharing and more steady development practices (See Section 4.2).

The system works on a remote server that runs an Apache web server and a MySQL database. The server runs all the PHP codes and communicates with the database within this server.

4.2. Problems and Solutions

4.2.1. Working with SQL Queries

```
file_names = ["../drop_tables.sql",
              "../create_tables.sql",
              "../create_views.sql",
              "../populate_tables.sql"]

input_files = []
for file_name in file_names:
  input_files.append(open(file_name, "r"))

output_str = ""
for input_file in input_files:
  output_str += input_file.read()

output_file = open("../RESET_ALL.sql", "w")
output_file.write(output_str)

for input_file in input_files:
  input_file.close()
output_file.close()
```

Figure 2

We needed to work on sample data for our database during our development. And we also needed to be on the same page with each other, in terms of table structures. For this purpose, we have written our SQL queries in .sql files and included them in our version control system. Since we needed to run each time they are updated, we have came up with a solution for fastening this process. A Python script (See Figure 1) that automatically merges them in the essential order and outputs a single file called “RESET_ALL.sql”. Thus, each time we change an SQL query in an .sql file, all we need to do is just running the script and using the output .sql file.

4.2.2. Generating Sample Data

We have used a dummy-data generator tool (available on http://filldb.info) to populate our database with rich data. We have imported our table structures to the system. Then we have chosen which type of dummy data will be generated (valid ranges phone numbers, valid dummy street addresses and dummy company names). Then we have exported this this generated data from the tool and imported into our database.

5. Advanced DB Features

5.1. Secondary Indices

The secondary indices we created for our system is for commonly used tables such as Account table which covers all the users of IBITUR -staff and customers- by their usernames and Tour table which indexed the tours by their prices.

```
CREATE INDEX username_index USING BTREE ON Account(username);
CREATE INDEX price_index USING BTREE ON Tour(price);
```

5.2. Advanced Features

A tour has many components in it (accommodations, travel routes, trip events, day descriptions). It's not practical to list all of these components in a short preview of the tour, thus we decided to create a summary view of the tour, which includes only the basic attributes of it.

```
CREATE VIEW TourPreview AS
    (SELECT Tour.ID AS tour_ID, name, description,image_path,
price, start_date, end_date, (quota - used_quota) AS
remaining_quota
    FROM Tour, TourUsedQuotas, TourInterval
    WHERE Tour.ID = TourUsedQuotas.tour_ID AND Tour.ID =
TourInterval.tour_ID);
```

5.3. Reports

5.3.1. Most visited cities of the month

This report will generate a table of cities and their total reservation number, made in a month. If a tour has more than one city in it, then both of the cities will be counted as a reservation made to them.

```
CREATE VIEW TempTourAssociations AS (
    SELECT tour_ID, city_name
    FROM TourAssociations NATURAL JOIN TourPreview
    WHERE (SELECT CURRENT_DATE + INTERVAL - 1 MONTH) <= start_date
        AND start_date <= (NOW()) );

CREATE VIEW CityPopularity AS (
   SELECT city_name, SUM(used_quota) AS popularity
   FROM TempTourAssociations NATURAL JOIN TourUsedQuotas
```

```
    GROUP BY city_name ORDER BY popularity DESC );
```

5.3.2. Top revenue-making countries of the year

This report will generate a table of countries and the total amount of spending that is done on that tours which include that country.

```
CREATE VIEW TempTourAssociations AS (
     SELECT tour_ID, country_name
     FROM TourAssociations NATURAL JOIN TourPreview
     WHERE (SELECT CURRENT_DATE + INTERVAL - 1 YEAR)
            <= start_date AND start_date <= (NOW())   );

CREATE VIEW CountryRevenues AS (
    SELECT country_name, SUM(price) AS revenue
    FROM TempTourAssociations NATURAL JOIN TourPreview
    GROUP BY country_name ORDER BY revenue DESC);
```

5.3.3. Report Outputs

Top revenue-making countries of the year

1. Bulgaria [~6539.00 TL]
2. Costa Rica [~4481.00 TL]
3. Bouvet Island (Bouvetoya) [~4454.00 TL]
4. Somalia [~4120.00 TL]
5. India [~3933.00 TL]
6. Hungary [~3888.00 TL]
7. Micronesia [~3395.00 TL]
8. Ghana [~3145.00 TL]
9. Pitcairn Islands [~3056.00 TL]
10. Iraq [~2933.00 TL]

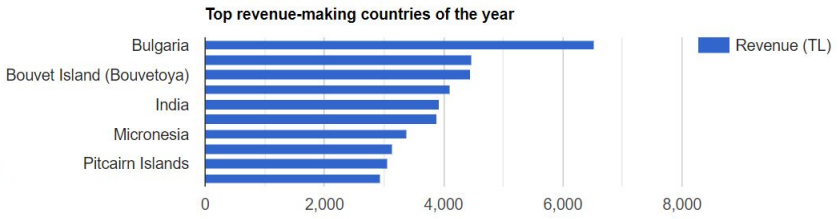
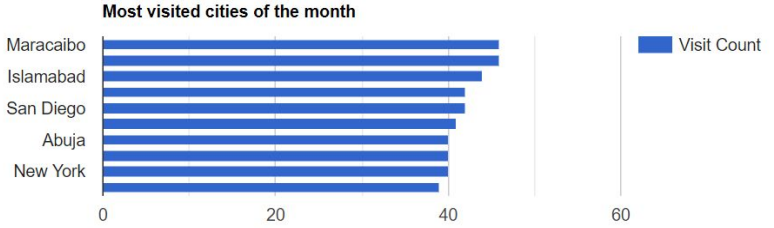


Figure 3

Most visited cities of the month

1. Maracaibo [~46 visits]
2. Baghdad [~46 visits]
3. Islamabad [~44 visits]
4. Kampala [~42 visits]
5. San Diego [~42 visits]
6. Dhaka [~41 visits]
7. Abuja [~40 visits]
8. Cali [~40 visits]
9. New York [~40 visits]
10. Kyoto [~39 visits]



6.Users Manual

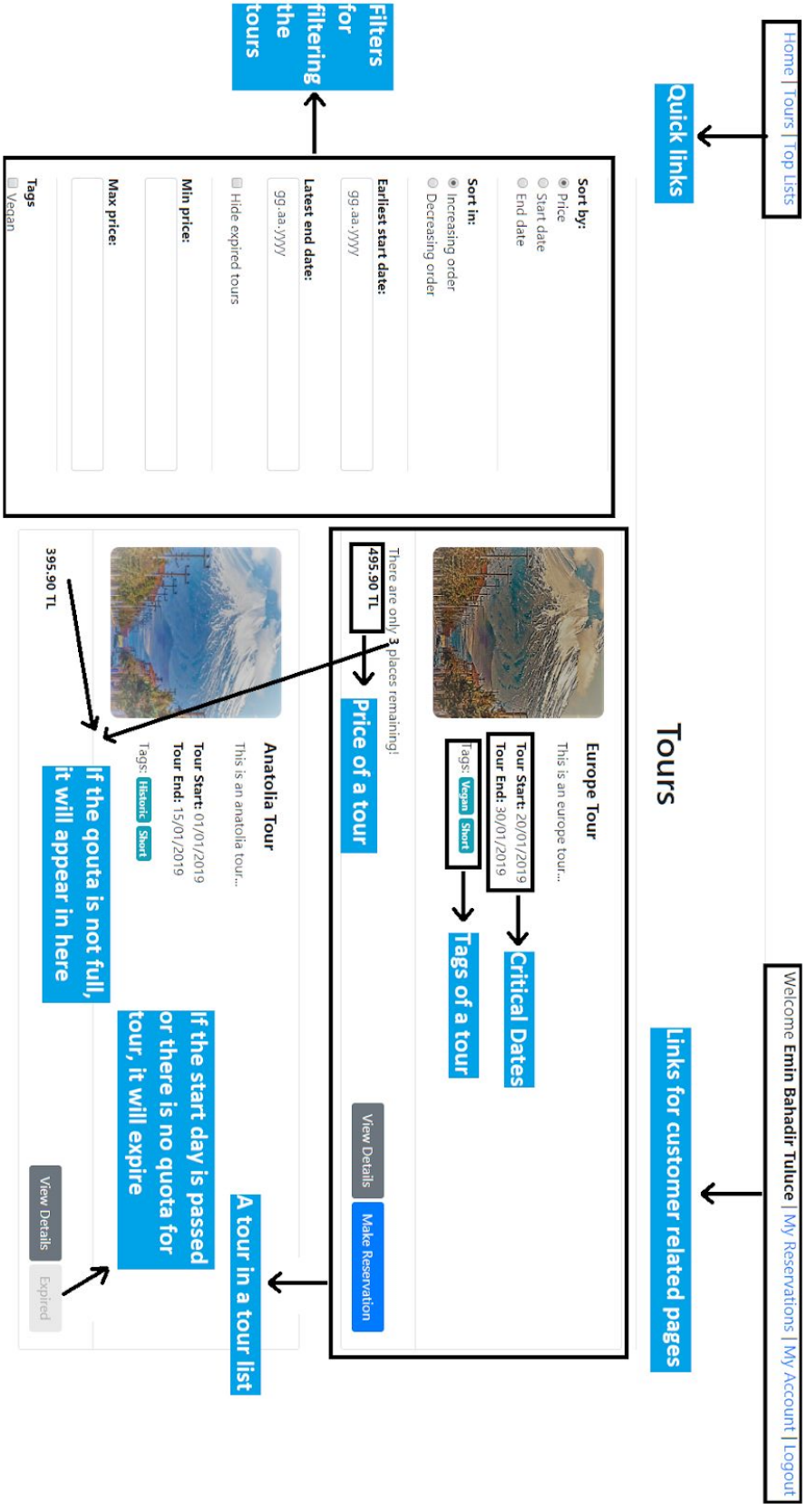


Figure 5

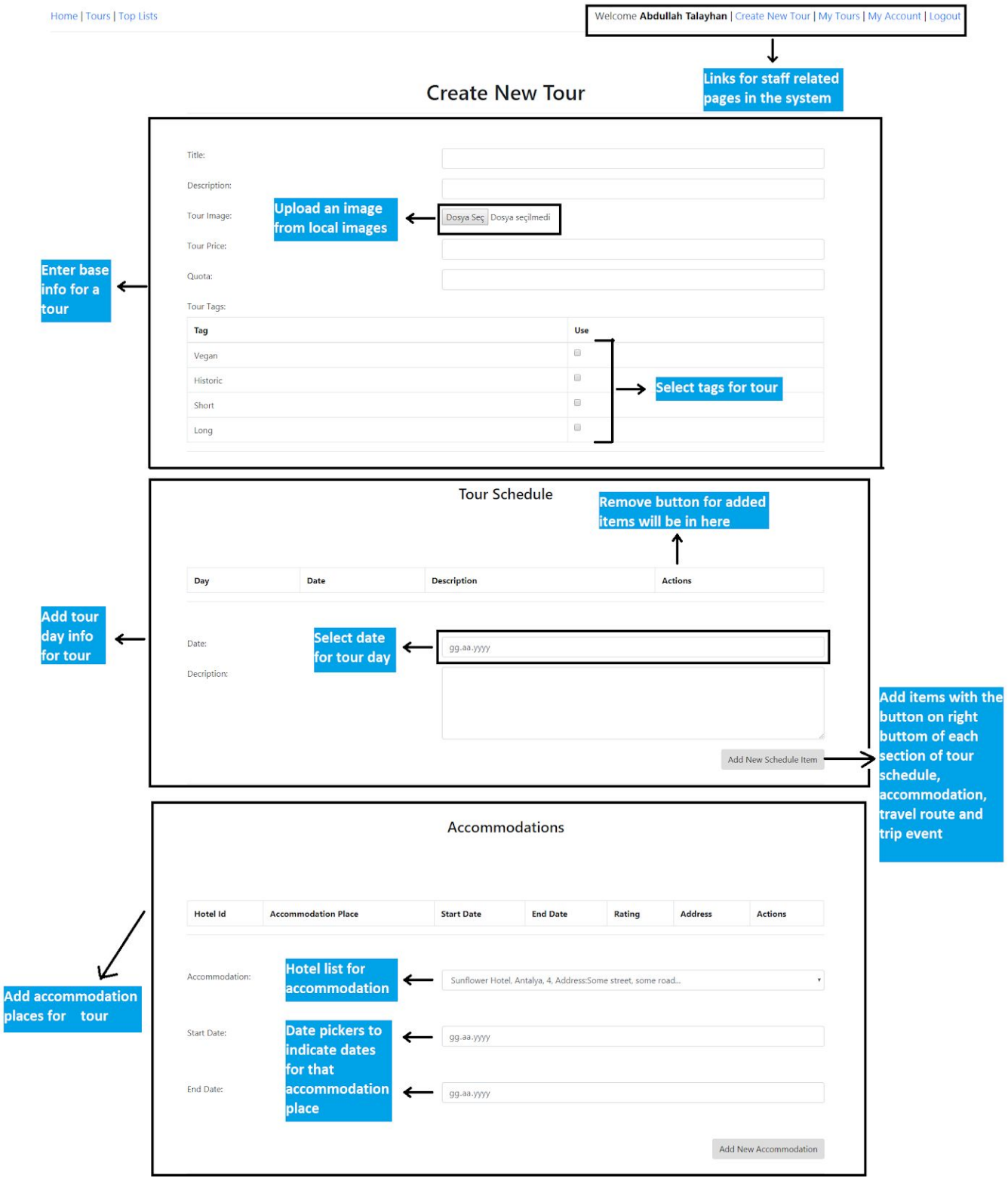

Home | Tours | Top Lists
Welcome Abdullah Talayhan | Create New Tour | My Tours | My Account | Logout
Links for staff related pages in the system
Create New Tour
Title:
Description:
Tour Image:
Upload an image from local images
Dosya Seç
Dosya seçilmedi
Tour Price:
Quota:
Tour Tags:
Enter base info for a tour
Tag
Use
Vegan
Historic
Short
Long
Select tags for tour
Tour Schedule
Remove button for added items will be in here
Day
Date
Description
Actions
Add tour day info for tour
Date:
Select date for tour day
gg.aa.yyyy
Decription:
Add New Schedule Item
Add items with the button on right buttom of each section of tour schedule, accommodation, travel route and trip event
Accommodations
Hotel Id
Accommodation Place
Start Date
End Date
Rating
Address
Actions
Add accommodation places for tour
Accommodation:
Hotel list for accommodation
Sunflower Hotel, Antalya, 4, Address:Some street, some road...
Start Date:
Date pickers to indicate dates for that accommodation place
gg.aa.yyyy
End Date:
gg.aa.yyyy
Add New Accommodation

Figure 6

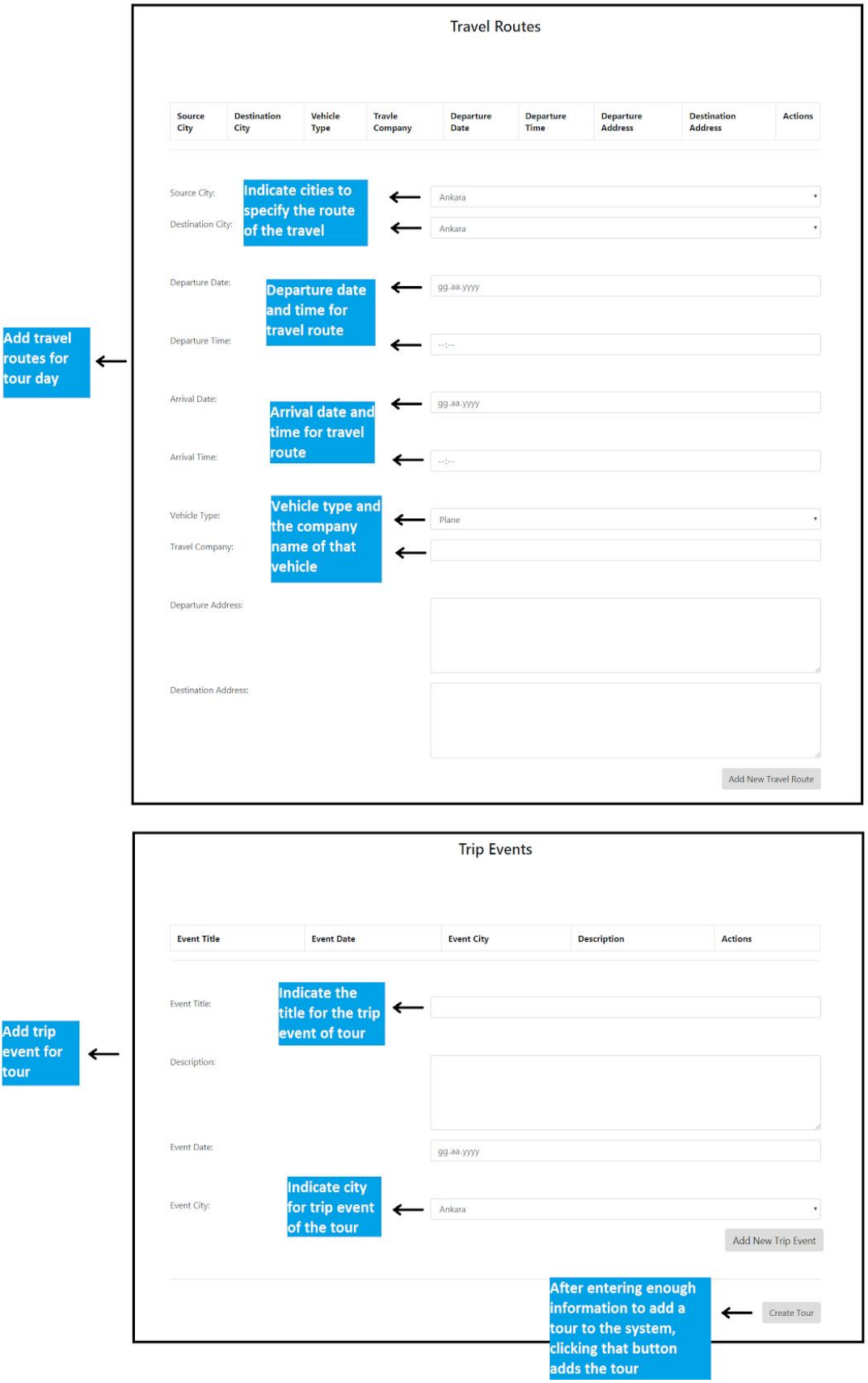

Travel Routes
Source City
Destination City
Vehicle Type
Travle Company
Departure Date
Departure Time
Departure Address
Destination Address
Actions
Source City:
Destination City:
Ankara
Ankara
Indicate cities to specify the route of the travel
Departure Date:
Departure Time:
gg.aa.yyyy
--:--
Departure date and time for travel route
Add travel routes for tour day
Arrival Date:
Arrival Time:
gg.aa.yyyy
--:--
Arrival date and time for travel route
Vehicle Type:
Travel Company:
Plane
Vehicle type and the company name of that vehicle
Departure Address:
Destination Address:
Add New Travel Route
Trip Events
Event Title
Event Date
Event City
Description
Actions
Event Title:
Indicate the title for the trip event of tour
Add trip event for tour
Description:
Event Date:
gg.aa.yyyy
Event City:
Ankara
Indicate city for trip event of the tour
Add New Trip Event
After entering enough information to add a tour to the system, clicking that button adds the tour
Create Tour

Figure 7

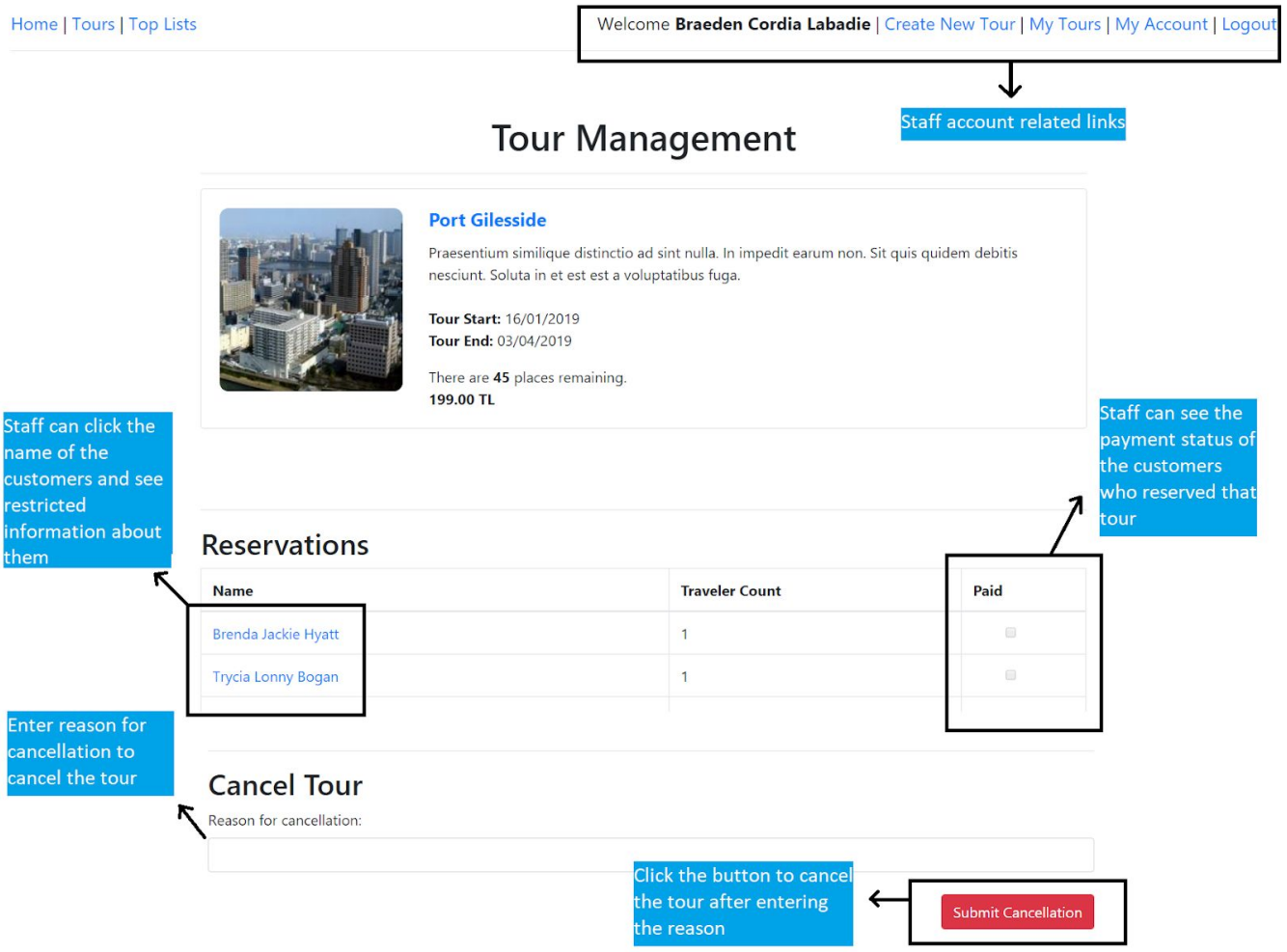


Figure 8

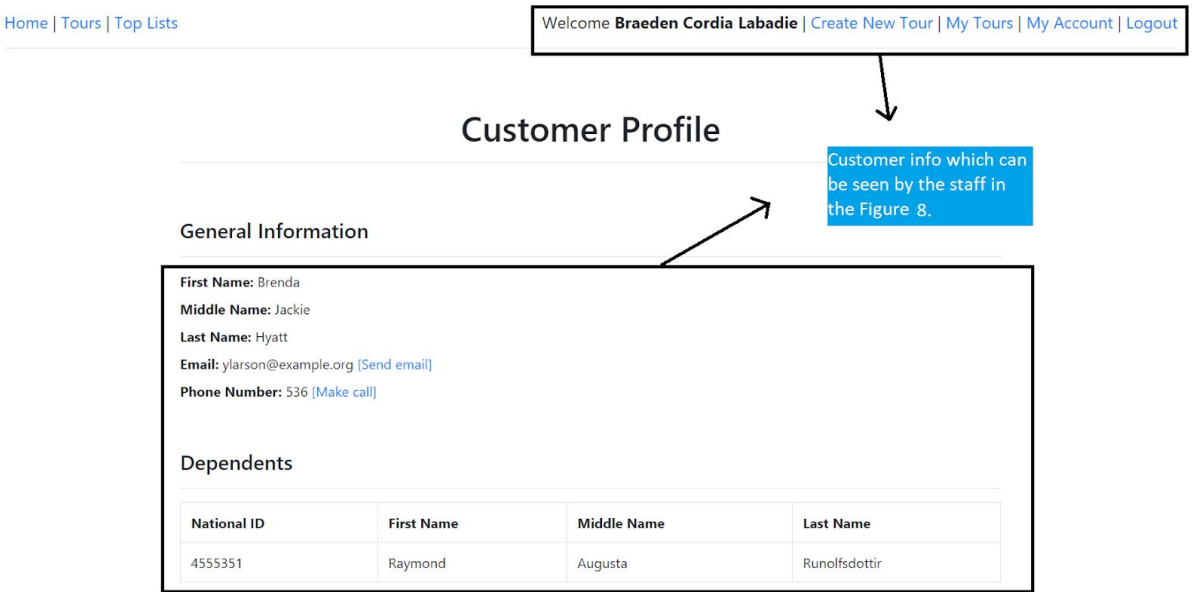


Figure 9