05/06/2023

CS 353 - Database Systems

Project Final Report

Room / Flat Rental Application

BNBilkent

Group 18

Project URL: https://github.com/mfkucuk/CS-353
Mehmet Feyyaz Küçük - 22003550
Ender Utlu - 22001983
Ege Ayan - 22002478
Deniz Çelik - 22003271
Parsa Keihan - 22001422

1. Project Description

BNBilkent is a Room / Flat Rental Web Application that comprises a database system. BNBilkent is used to publish and rent accommodations around Turkey. The application consists of different user types and the interaction between them. Homeowners can put their rental dwellings on the website and mention the related features such as available dates, location, etc. Travelers can rent a domicile by searching for a specific date in an area or filter the listed houses based on specific features. Then, they can look for mentioned features and book the place using the balance system. Admins are the usual controllers of the general system. The system provides an in-app balance payment system to book the places before. Also, each advert would be controlled by the system before the final release. The admin is able to create system reports, like finding the most popular rental houses within a region. A list of all possible rooms and flats are shown in the travelers main page. An accommodation can be booked after the payment is done successfully and the balance is reduced. Homeowners should specify some mandatory details about new rentals and extra features if needed before publishing it. Furthermore, travelers can write a review and rate for a place they have stayed in previously, and see others' comments before booking a new rental. Moreover, homeowners and travelers can ask and answer questions related to the rentals if needed. There are different listing, filtering, and searching functionalities in the project. As there is different data to be saved and used in the program BNBilkent uses an efficient, consistent, secure, scalable, and integrated database system to store and publish the data as needed. The database lets us keep all the information the users will interact with in a reliable and scalable way. The database is generated manually and its aim is to help understand how a database system works and how it interacts with frontend-backend codes to provide a user-friendly application.

2. Individual Contributions

Proposal, Design and Final Reports were a group effort and they cannot be divided into parts for any of us to specifically tell where we contributed. We have contributed equally for the reports.

Ege Ayan: Creation of Profile Pages for admin, homeowner and traveler. Creation of Rental pages for traveler and homeowner. Contributed to Main pages for homeowners and travelers. Creation of modal pop-ups such as features, reviews, side-menu, payment method and filter. Contributed to connect backend with frontend using Axios calls by providing appropriate data structures and handling them. Installment of React Router infrastructure for navigation. General debugging and fixing. Contributed to some database queries and backend points. Styling and visual adjustments for React components.

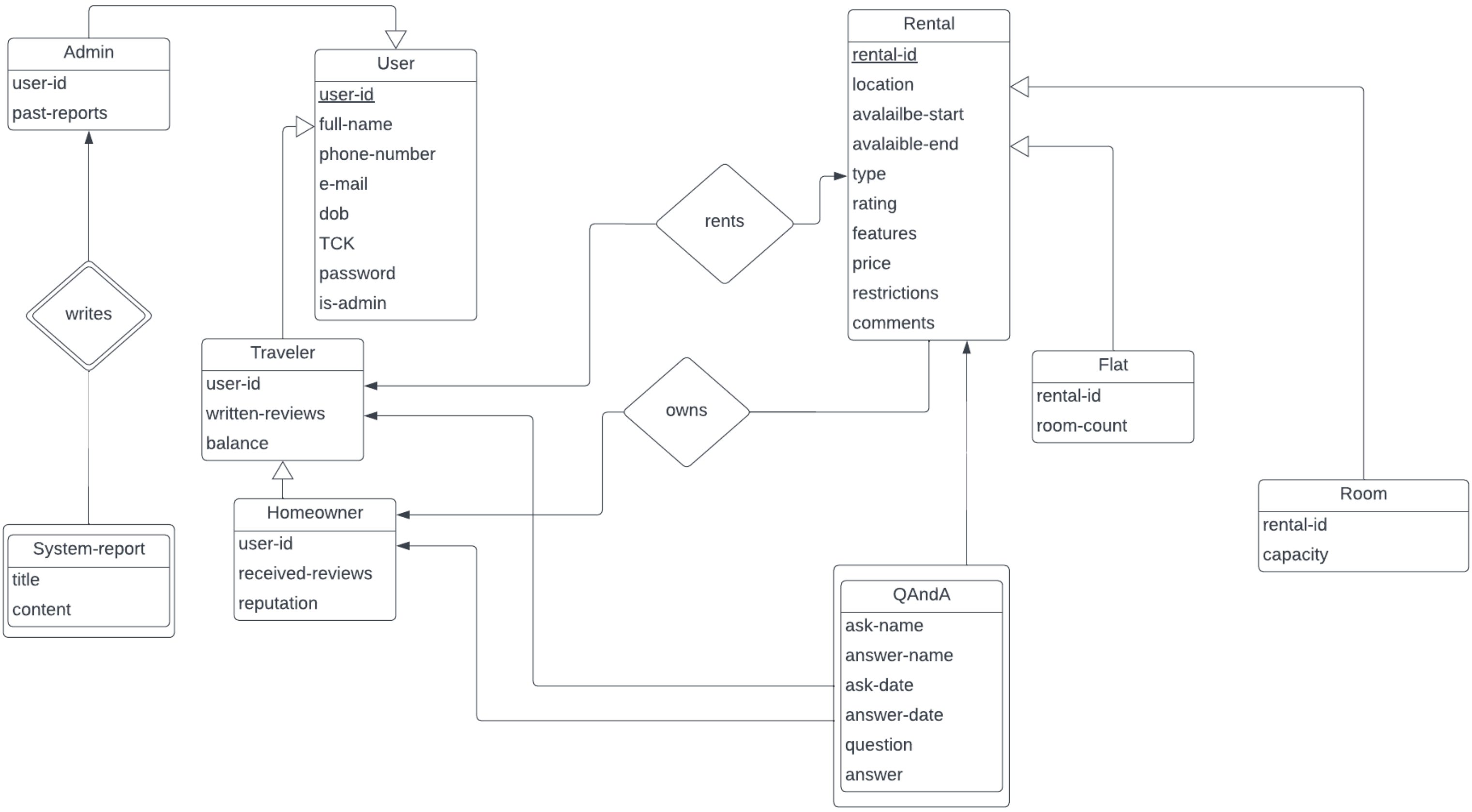
Deniz Çelik: Created the Main pages for homeowners and travelers. Contributed to Profile Pages for admin, homeowner and traveler. Contributed to Rental pages for traveler and homeowner. Contributed to Login, Register and ResetPassword pages. Creation of the logo. Styling improvements, position adjustments, size adjustments, other general updates and bug fixing for all UI pages. Some small functions that are used such as the date formatter. Contributed to connect backend with frontend using Axios calls by providing appropriate data structures and handling them. Contributed to the database for users query and user creation.

Parsa Keihan: Created the common pages: Login, Register, ResetPassword. Contributed to Homeowner and Traveler pages. Used Axios, CSS, HTML, and javascript technologies to provide frontend codes. Creation of modal pop-ups such as features, reviews, side-menu, payment method and filter. Contributed to connect backend with frontend using Axios calls by providing appropriate data structures and handling them. Installment of React Router infrastructure for navigation. General debugging and fixing. Contributed to modifying Database queries, advanced database features, and tables.

Mehmet Feyyaz Küçük: Responsible for backend and some frontend. Creation of rental, room, flat, qanda, system report tables as well as their model, dao, service and controller classes. Set up the appropriate endpoints for most classes. Also, wrote the appropriate SQL statements to access the database by implementing the dao functions. Created the views for traveler, homeowner, admin, room, and flat tables. Wrote the advanced SQL queries in the creation of the system report. Tested backend and database functionalities using pgAdmin4 and Postman. Worked on the frontend side to connect the backend endpoints to the client using Axios.

Ender Utlu: Creation of user, traveler, homeowner and admin tables. Adding the indexing SQL statements. Connection of user, traveler, homeowner, admin and reset password pages with the backend. Used Axios http requests while connecting the backend. Formatting the dates across the whole application, all pages. Creation of backend endpoints of user, traveler, homeowner and admin tables as well as their dao, service, model and views. Wrote the necessary SQL statements in data access classes which are the implementation of dao interfaces. Hosting a local server in testing all application functionality. Fixing some navigation issues in the frontend. Tested a lot of backend functionality using Postman and PGadmin4 since we used postgresql.

3. Final E/R Model



4. Final List of Tables

4.1. Users

Relational Model:
Users (user_id, full_name, e_mail, dob, TCK, password, phone_number, is_admin)
Functional Dependencies:
user_id → full_name, e_mail, dob, TCK, password, phone_number, is_admin
Candidate Key(s): user_id
Primary Key: user_id
Foreign Key(s): None
Normal form: BCNF

4.2. Traveler

Relational Model:
Traveler(user_id, written_reviews, reputation)
Functional Dependencies:
user-id → written-reviews, reputation
Candidate Key(s): user_id
Primary Key: user_id
Foreign Key(s): None
Normal form: BCNF

4.3. Homeowner

Relational Model:
Traveler(user_id, received_reviews, balance)
Functional Dependencies:
user-id → written-reviews, balance
Candidate Key(s): user_id
Primary Key: user_id
Foreign Key(s): None
Normal form: BCNF

4.4. Admin

Relational Model:
Traveler(user_id, past_reports)
Functional Dependencies:
user-id → past_reports
Candidate Key(s): user_id
Primary Key: user_id

Foreign Key(s): None
Normal form: BCNF

4.4. Rental

Relational Model:
Rental(rental_id, location, available_start, available_end, restrictions, type, rating, features, comments, price, traveler_id, homeowner_id)
Functional Dependencies:
rental_id → location, available_start, available_end, restrictions, type, rating, features, comments, price, traveler_id, homeowner_id
Candidate Key(s): rental_id
Primary Key: rental_id
Foreign Key(s):
homeowner_id references Users.user_id
traveler_id references Users.user_id
Normal form: BCNF

4.5. Flat

Relational Model:
Flat(rental_id, room_count)
Functional Dependencies:
rental_id → room_count
Candidate Key(s): rental_id
Primary Key: rental_id
Foreign Key(s):
rental_id references Rental.rental_id
Normal form: BCNF

4.6. Room

Relational Model:
Flat(rental_id, capacity)
Functional Dependencies:
rental_id → capacity
Candidate Key(s): rental_id
Primary Key: rental_id
Foreign Key(s):
rental_id references Rental.rental_id
Normal form: BCNF

4.7. SystemReport

Relational Model:
Flat(admin_id, title, content)
Functional Dependencies:

admin_id→ title, content
Candidate Key(s): admin_id, title, content
Primary Key: admin_id, title, content
Foreign Key(s):
admin_id references Users.user_id
Normal form: BCNF

4.8. QAndA

Relational Model:
QAndA(ask_date, answer_date, question, answer, ask_id, answer_id, rental_id)
Functional Dependencies:
ask_date -> question, ask_id
answer_date -> answer, answer_id
ask_id -> ask_date, question
answer_id -> answer_date, answer
Candidate Key(s): None
Primary Key: None
Foreign Key(s):
rental-id references Rental.rental-id.
ask-id references Users.user-id
answer-id references Users.user-id
Normal form: BCNF

5. Implementation Details

5.1. Database

PostgreSQL is used for database implementation along with PgAdmin 4 for viewing and monitoring the database structure.

5.2. Backend

Java SpringBoot is used for defining endpoints and providing backend. Also, Postman is used for API testing.

5.3. Frontend

The technologies used in frontend code are Javascript, React Library, HTML & CSS, and Axios for connections with backend code. React Modal library is used for creating pop-ups on the screen.

6. Advanced Database Features

6.1 Views

We used views in our database to make queries more convenient. Specifically, since we used vertical mapping in our database (i.e. class B inheriting from class A does not contain the properties of class A in its table) views made it easier to access sub-classes without having to join them everytime.

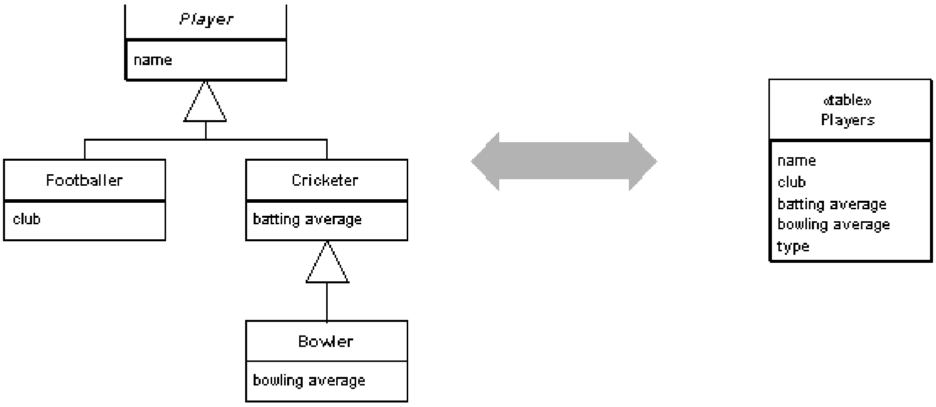


Fig. 1 Example of vertical mapping

Our database design especially had many inheritance relationships between the tables, (i.e. Traveler and Admin inherits User, Homeowner inherits Traveler and Room/Flat inherits Rental). For example, to populate the profile page of a Traveler, instead of separately querying User and Traveler tables, we query the `TravelerView` view once to fetch all the information needed. Here are all the views used in the database:

6.1.1 TravelerView

TravelerView is used to join User and Traveler tables to fetch data faster and easier for traveler related queries.

```
CREATE VIEW TravelerView AS
SELECT U.user_id, U.full_name, U.e_mail, U.dob, U.TCK, U.password,
U.phone_number, U.is_admin, T.written_reviews, T.balance
FROM Users U
JOIN Traveler T ON U.user_id = T.user_id;
```

6.1.2 HomeownerView

HomeownerView is used to join User, Traveler, and Homeowner tables to fetch data faster and easier for homeowner related queries.

```
CREATE VIEW HomeownerView AS
```

```sql
SELECT U.user_id, U.full_name, U.e_mail, U.dob, U.TCK, U.password,
U.phone_number, U.is_admin, T.written_reviews, T.balance, H.received_reviews,
H.reputation
FROM Users U
JOIN Traveler T ON U.user_id = T.user_id
JOIN Homeowner H ON U.user_id = H.user_id;
```

6.1.3 AdminView

AdminView is used to join User and Admin tables to fetch data faster and easier for admin related queries.

```sql
CREATE VIEW AdminView AS
SELECT U.user_id, U.full_name, U.e_mail, U.dob, U.TCK, U.password,
U.phone_number, U.is_admin, A.past_reports
FROM Users U
JOIN Admin A ON U.user_id = A.user_id;
```

6.1.4 RoomView

RoomView is used to join Rental and Room tables to fetch data faster and easier for room related queries.

```sql
CREATE VIEW RoomView AS
SELECT R.rental_id, R.location, R.available_start, R.available_end,
R.restrictions, R.type, R.rating, R.features, R.comments, R.price,
R.traveler_id, R.homeowner_id, Ro.capacity
FROM Rental R
JOIN Room Ro ON R.rental_id = Ro.rental_id;
```

6.1.5 FlatView

FlatView is used to join Rental and Flat tables to fetch data faster and easier for room related queries.

```sql
CREATE VIEW FlatView AS
SELECT R.rental_id, R.location, R.available_start, R.available_end,
R.restrictions, R.type, R.rating, R.features, R.comments, R.price,
R.traveler_id, R.homeowner_id, F.room_count
FROM Rental R
JOIN Flat F ON R.rental_id = F.rental_id;
```

6.2 Unique Indices

```
DROP INDEX IF EXISTS idx_unique_email;
DROP INDEX IF EXISTS idx_unique_tck;
DROP INDEX IF EXISTS idx_unique_phone_number;

CREATE UNIQUE INDEX idx_unique_email ON Users (e_mail);
CREATE UNIQUE INDEX idx_unique_tck ON Users (TCK);
CREATE UNIQUE INDEX idx_unique_phone_number ON Users (phone_number);
```

Unique indices are used to prevent creating multiple users with the same critical info. These are email, phone number and TCK. This indexing prohibits having duplicate of these information.

6.3 SystemReport

To create our SystemReport instance we utilized Advanced SQL queries without the use of any Java logic. Here are some such queries:

6.3.1 COUNT(*) Aggregate Function and GROUP BY keyword

To find rental counts per locations, we grouped the rows by location using the GROUP BY query and counted the grouped rows using the aggregate function count.

```
final String sql = "SELECT location, COUNT(*) as count FROM rental GROUP BY location";
```

6.3.2 ORDER BY and LIMIT keywords

To find lists such as most expensive rentals, least expensive rentals, most reputable homeowners, we sorted them in the ascending or the descending orders using the ORDER BY and ASC/DESC keyword pairs, then limited the number of rows we fetch using the LIMIT keyword.

```
final String sql = "SELECT full_name, reputation FROM HomeownerView ORDER BY reputation DESC LIMIT 10";
```

6.3.3 LIKE keyword

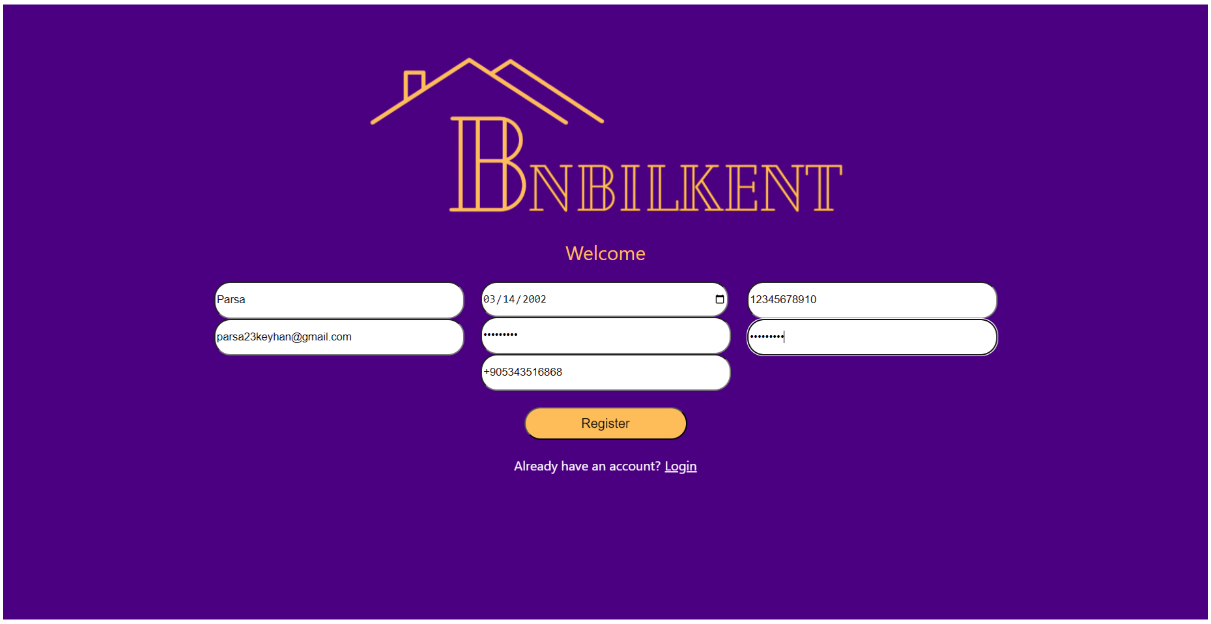
To implement the filtering/search functionality for rentals, we matched the search key using LIKE and wildcard symbol `%`.

```
final String sql = "SELECT * FROM Rental WHERE location LIKE ‘%?%’";
```

7. User Manual

7.1. Register

In this page a new user can register to BNBilkent using his/her full name, date of birth, TCK, Email, Password, and a phone number. Register button navigates to the login page automatically if registration is done successfully. It also gives an error in case there is a problem during the registration process. There is also a hyperlink to navigate to Login under the register button if the user already has an account. A new user is a traveler and a homeowner at the same time but admin user-type is determined by database admin manually.



Parsa

03/14/2002

12345678910

parsa23keyhan@gmail.com

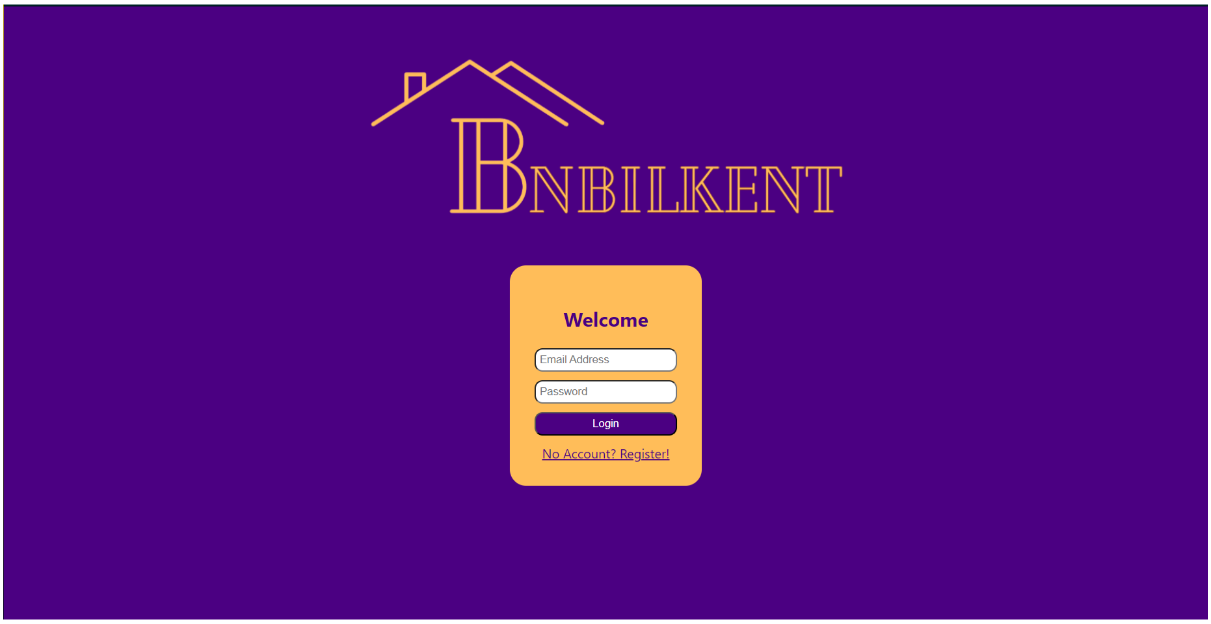
•••••••••

•••••••••

+905343516868

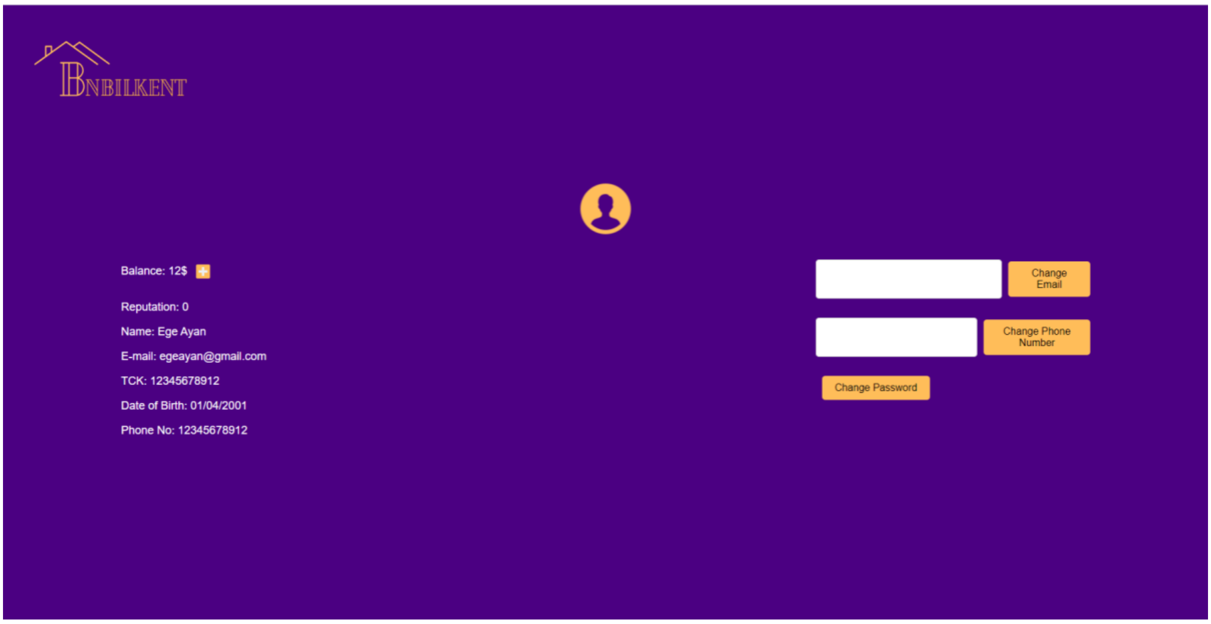
7.2. Login

In this page a user can login to his/her account using Email and the password. There would be an error if there is no match and login occurs when there is a match. There is also a hyperlink under the login button for the user who doesn't have an account. If login is successfully done the navigation is to the traveler main page as a user-type.



7.3. Homeowner Profile Page

In this page Homeowner can change his/her password, phone number, and Email if needed. Also, users can view reputation, which is the average rating from travelers, and his/her personal details. Moreover, the user can add to the account balance with a different window. This page can be accessed via the side menu button named “Profile”.



Balance: 12$

Reputation: 0

Name: Ege Ayan

E-mail: egeayan@gmail.com

TCK: 12345678912

Date of Birth: 01/04/2001

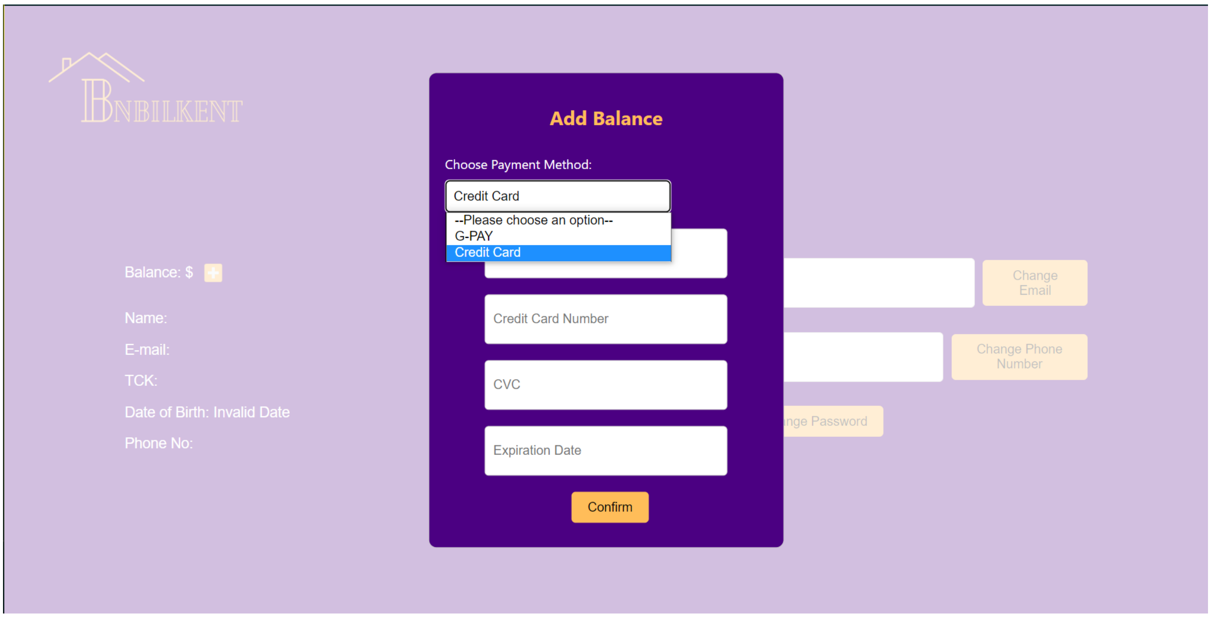
Phone No: 12345678912

7.4. Traveler Profile Page

Same as Homeowner Profile Page but Reputation is omitted.

7.5. Homeowner Add Balance

In this pop-up Homeowner can choose payment option and fill in the necessary information and the amount of balance he/she wants to add. Then, using the confirm button the amount is added to the balance.



Add Balance

Choose Payment Method: Credit Card

Balance: $

Name:

E-mail:

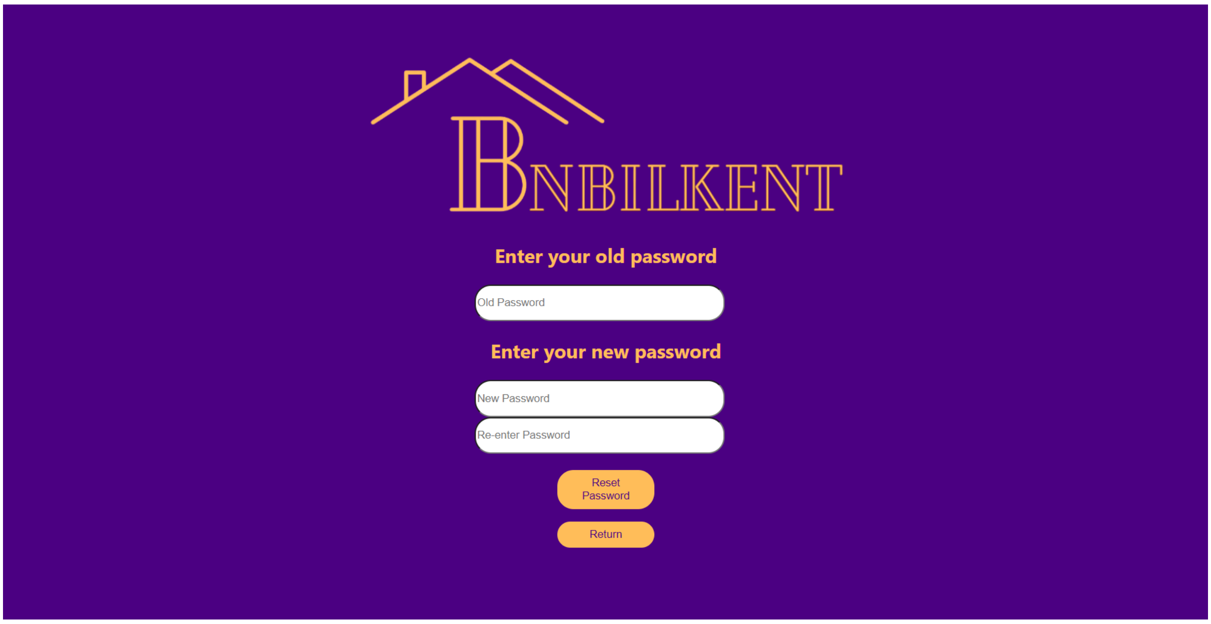
TCK:

Date of Birth: Invalid Date

Phone No:

7.6. Reset Password Page

Users can access this page with a change password button on his/her profile page. The old password should match the already existing password and the new password is written after. The security check for an ideal password is done and the change password button changes the password. The return button is used to go back to the profile page. The new password and re-entered password should match in all pages.

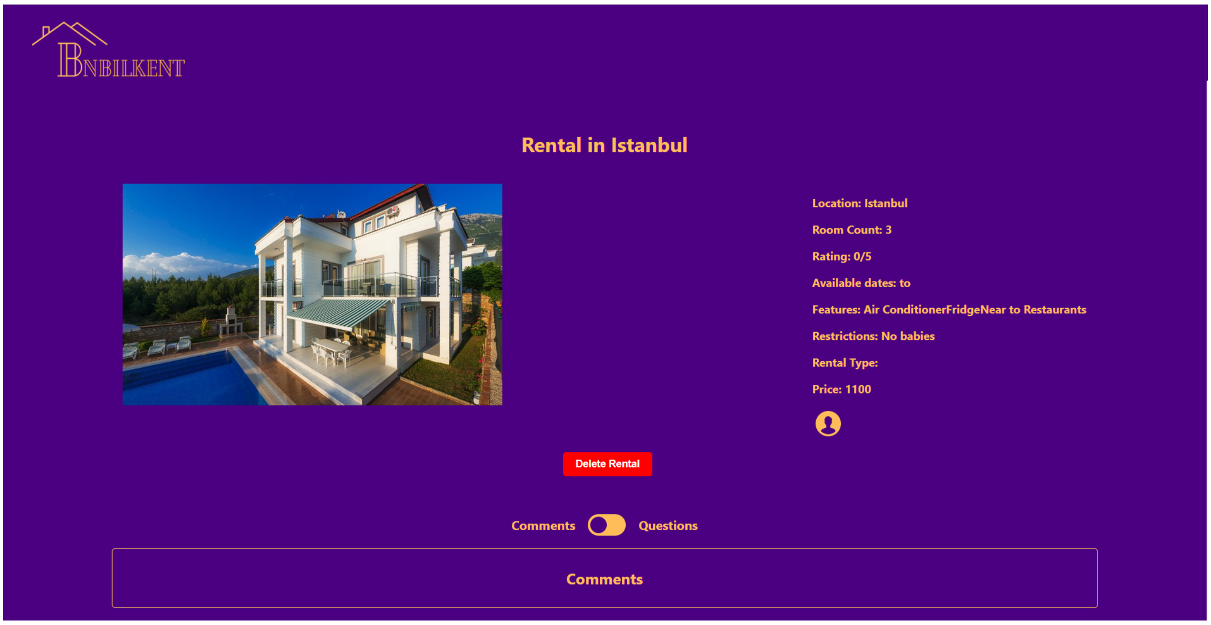


Enter your old password

Enter your new password

7.7. Homeowner Review

In this page, the homeowner can review his/her rental information such as its location, room count, rating, available dates, Features, Restrictions, Rental Type and Price. The homeowner can also view the comments that have written to the rental or see the questions that have been asked about the rental. The homeowner can also reply to the questions through this screen or delete the rental.



Rental in Istanbul

Location: Istanbul

Room Count: 3

Rating: 0/5

Available dates: to

Features: Air ConditionerFridgeNear to Restaurants

Restrictions: No babies

Rental Type:

Price: 1100

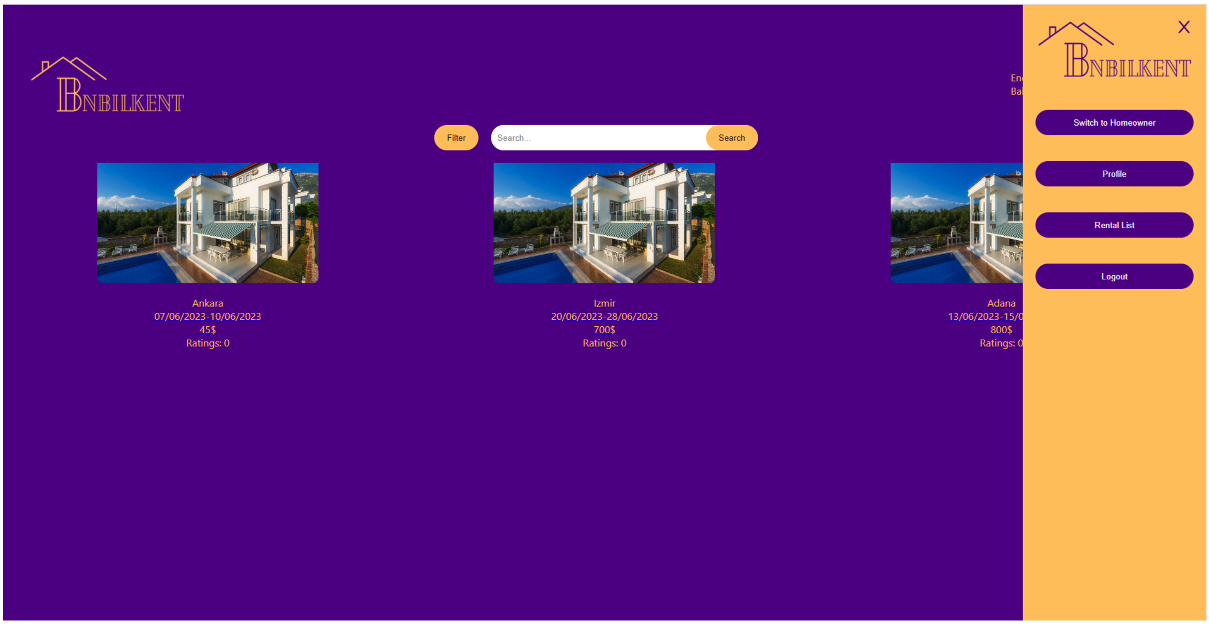
7.8. Traveler Add Balance

It is the same as the homeowner Add-Balance pop-up.

7.9. Traveler Main Page

In this page Traveler can see all visible and available rentals in the system and click on them to rent if needed. Also there are brief details for each rental. There is a filtering and search option for rentals and a side bar at the right of the page. Logout button is to quit the system, Traveler can see current and previous listing using Rental list button, visit profile page with profile button, and Switch to Homeowner user-type using Switch to Homeowner button.

There is also balance and the name of the user shown at the top right of the page.



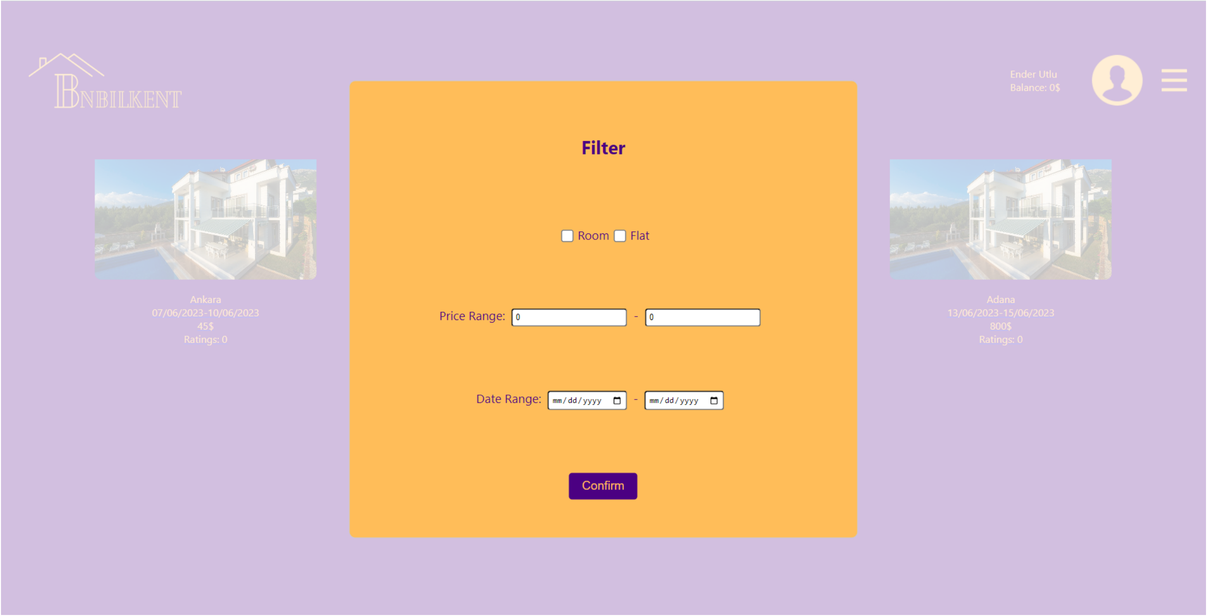
Ankara
07/06/2023-10/06/2023
45$
Ratings: 0

Izmir
20/06/2023-28/06/2023
700$
Ratings: 0

Adana
13/06/2023-15/0
800$
Ratings: 0

7.10. Filter

The filter pop-up can be accessed by the filter button at the traveler main page. Through this screen, the user can specify the type of rentals, price range and date range for the rentals that he/she wants to be listed on the main page by specifying them through check buttons and input fields.



Filter

□ Room □ Flat

Price Range: 0 - 0

Date Range: mm/dd/yyyy - mm/dd/yyyy

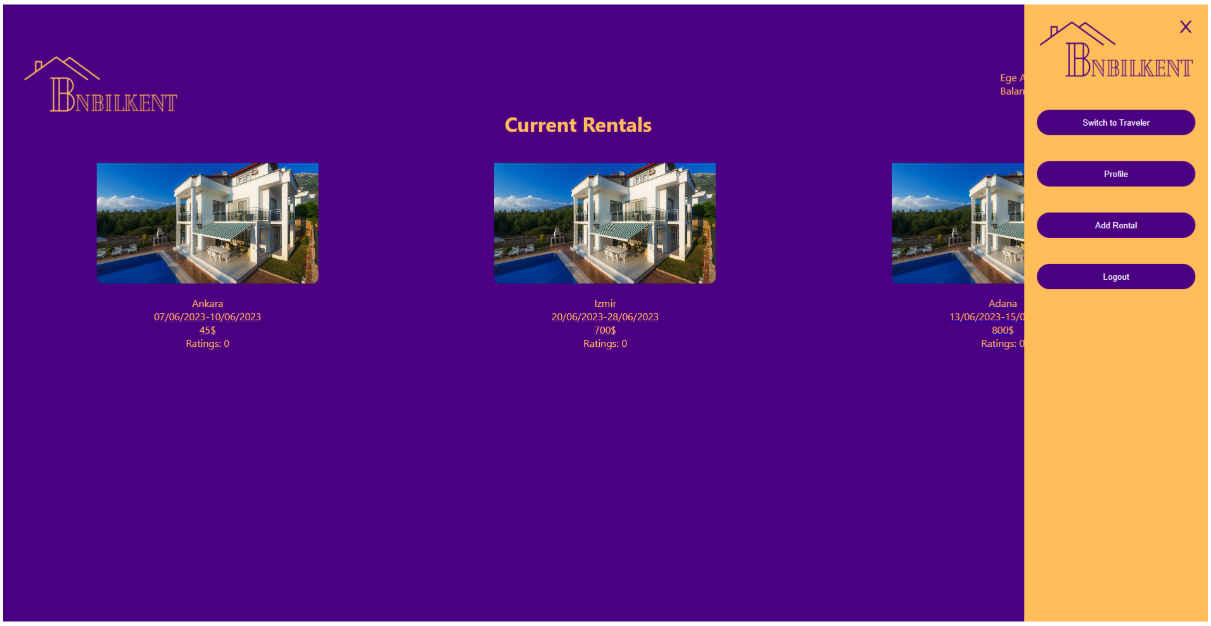
Confirm

Ankara
07/06/2023-10/06/2023
45$
Ratings: 0

Adana
13/06/2023-15/06/2023
800$
Ratings: 0

7.11. Homeowner Main Page

In this page, the homeowner can view all of his/her rentals. When clicked on one of them, the user can navigate to the Homeowner review page with more information. Similar to the traveler main page, there is a side menu that can be accessed through this screen which implements respective functions listed on it.



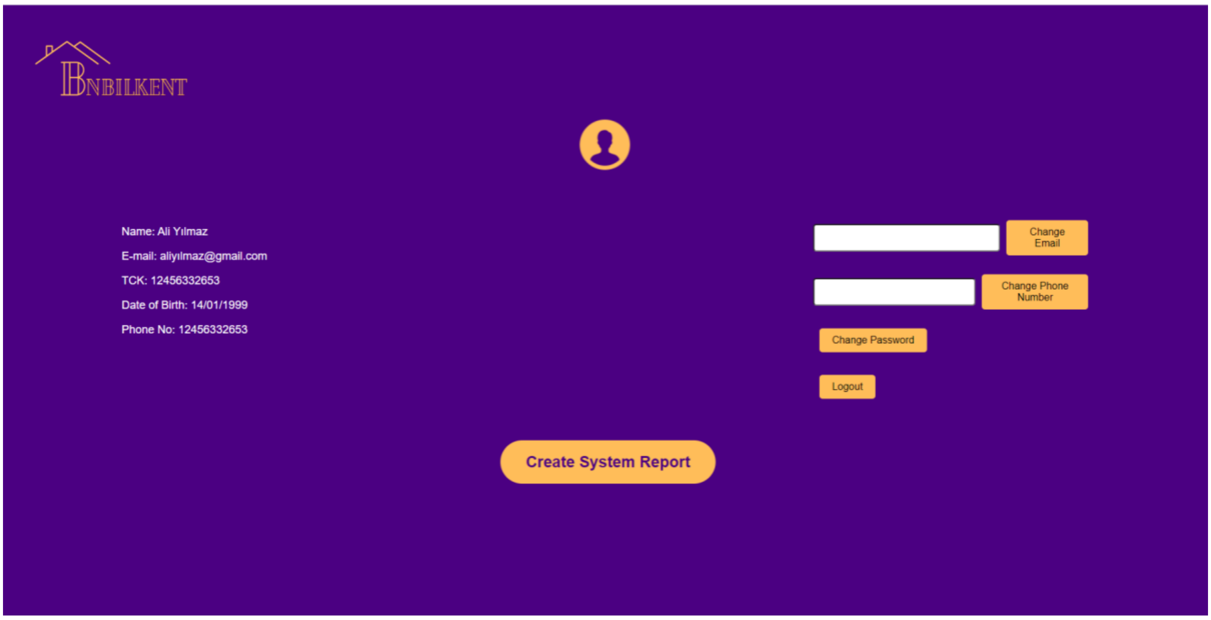
Ankara
07/06/2023-10/06/2023
45$
Ratings: 0

Izmir
20/06/2023-28/06/2023
700$
Ratings: 0

Adana
13/06/2023-15/0
800$
Ratings: 0

7.12. Admin Profile Page

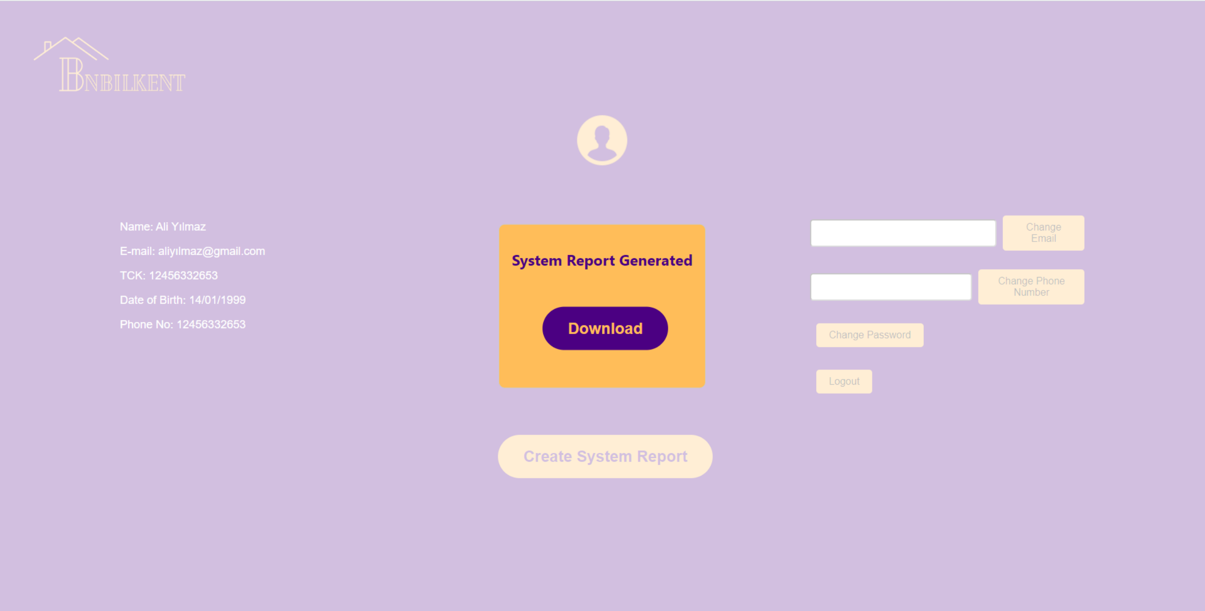
In this page, the admin can see his/her personal information. Also, email and password can be change as needed. Also admin can logout to quit the system. By clicking on creat system report button a new pop-up appears to perform and print some database related queries.



Name: Ali Yılmaz
E-mail: aliyılmaz@gmail.com
TCK: 12456332653
Date of Birth: 14/01/1999
Phone No: 12456332653

7.13. Admin System Report

In this page, the admin can create a system report which can be generated txt.



Name: Ali Yılmaz

E-mail: aliyılmaz@gmail.com

TCK: 12456332653

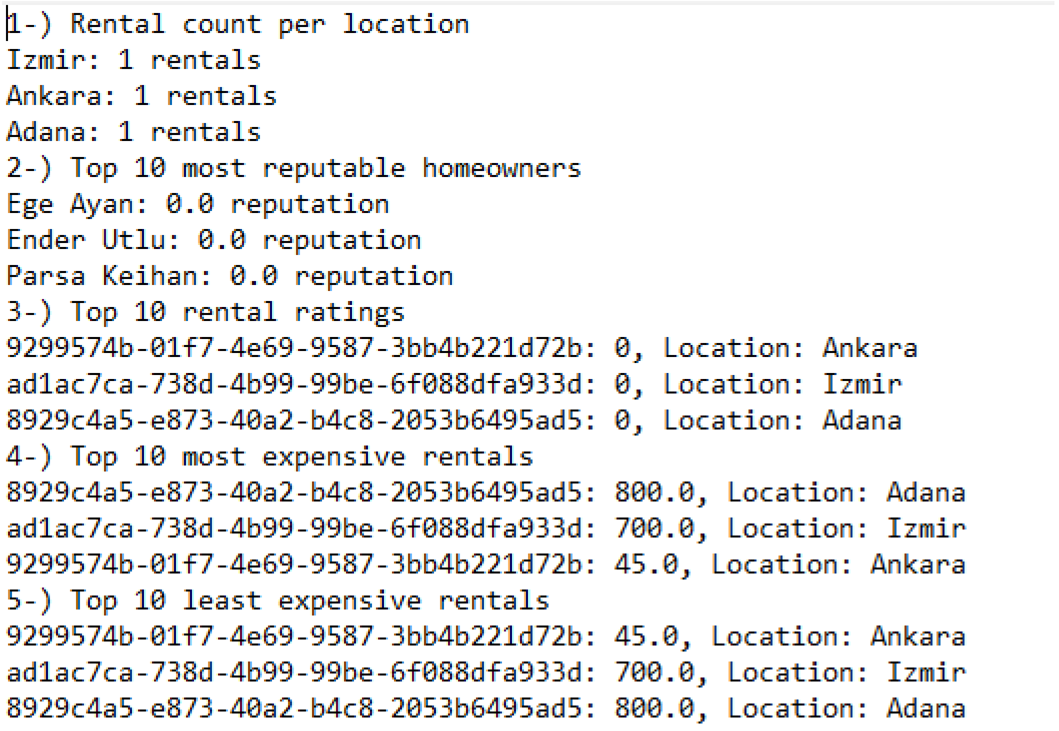
Date of Birth: 14/01/1999

Phone No: 12456332653

System Report Generated

7.14. Admin System Report (.txt file)

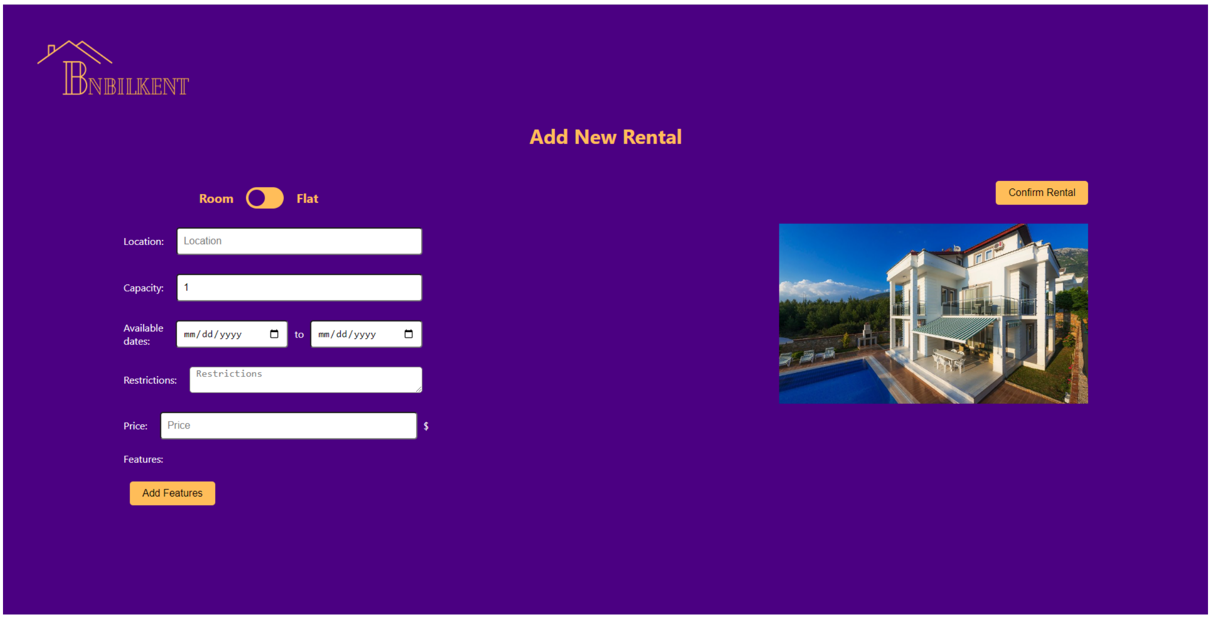
This is the generated report from an admin user.



```
1-) Rental count per location
Izmir: 1 rentals
Ankara: 1 rentals
Adana: 1 rentals
2-) Top 10 most reputable homeowners
Ege Ayan: 0.0 reputation
Ender Utlu: 0.0 reputation
Parsa Keihan: 0.0 reputation
3-) Top 10 rental ratings
9299574b-01f7-4e69-9587-3bb4b221d72b: 0, Location: Ankara
ad1ac7ca-738d-4b99-99be-6f088dfa933d: 0, Location: Izmir
8929c4a5-e873-40a2-b4c8-2053b6495ad5: 0, Location: Adana
4-) Top 10 most expensive rentals
8929c4a5-e873-40a2-b4c8-2053b6495ad5: 800.0, Location: Adana
ad1ac7ca-738d-4b99-99be-6f088dfa933d: 700.0, Location: Izmir
9299574b-01f7-4e69-9587-3bb4b221d72b: 45.0, Location: Ankara
5-) Top 10 least expensive rentals
9299574b-01f7-4e69-9587-3bb4b221d72b: 45.0, Location: Ankara
ad1ac7ca-738d-4b99-99be-6f088dfa933d: 700.0, Location: Izmir
8929c4a5-e873-40a2-b4c8-2053b6495ad5: 800.0, Location: Adana
```

7.15. Homeowner Add New Rental Screen

Through this page, the homeowner can publish a new rental by specifying its type, location capacity, available dates, restrictions and features. This page can be accessed via homeowner main page's side menu button named 'Add Rental'



Add New Rental

Room Flat

Location: Location

Capacity: 1

Available dates: mm/dd/yyyy to mm/dd/yyyy

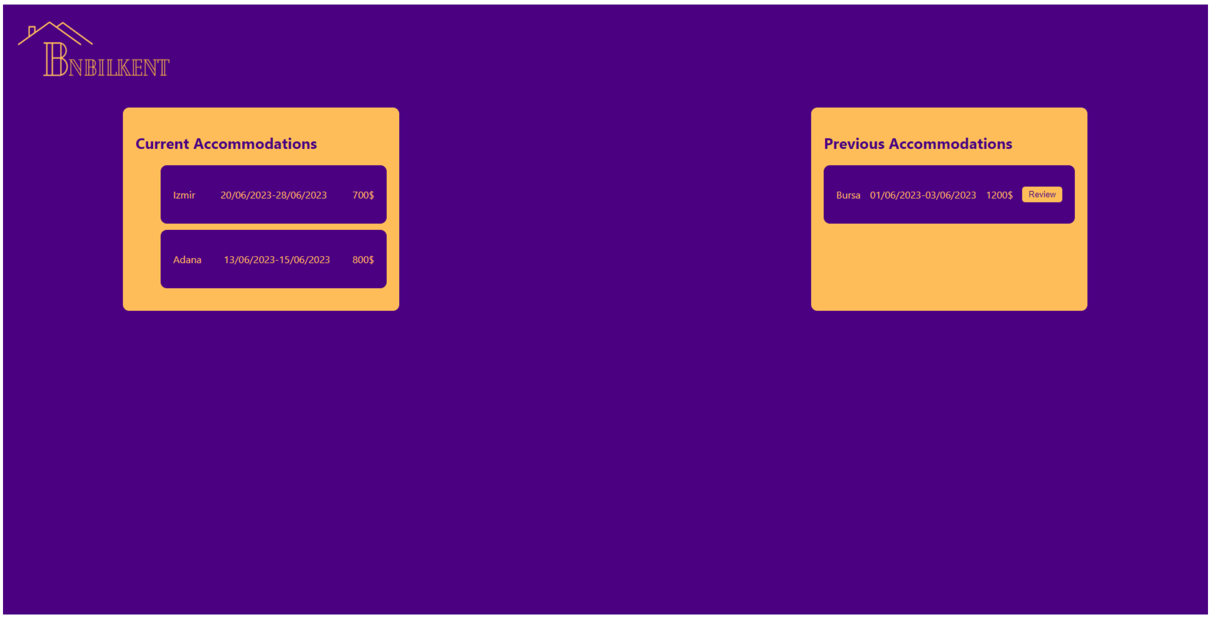
Restrictions: Restrictions

Price: Price $

Features:

7.16. Traveler Listing

In this page, the traveler can list both the current accommodations that he/she is renting and the previous accommodations that they have rented. For previous accommodations there is also a review button which can be used to make comments and give ratings via pop-up.



Current Accommodations

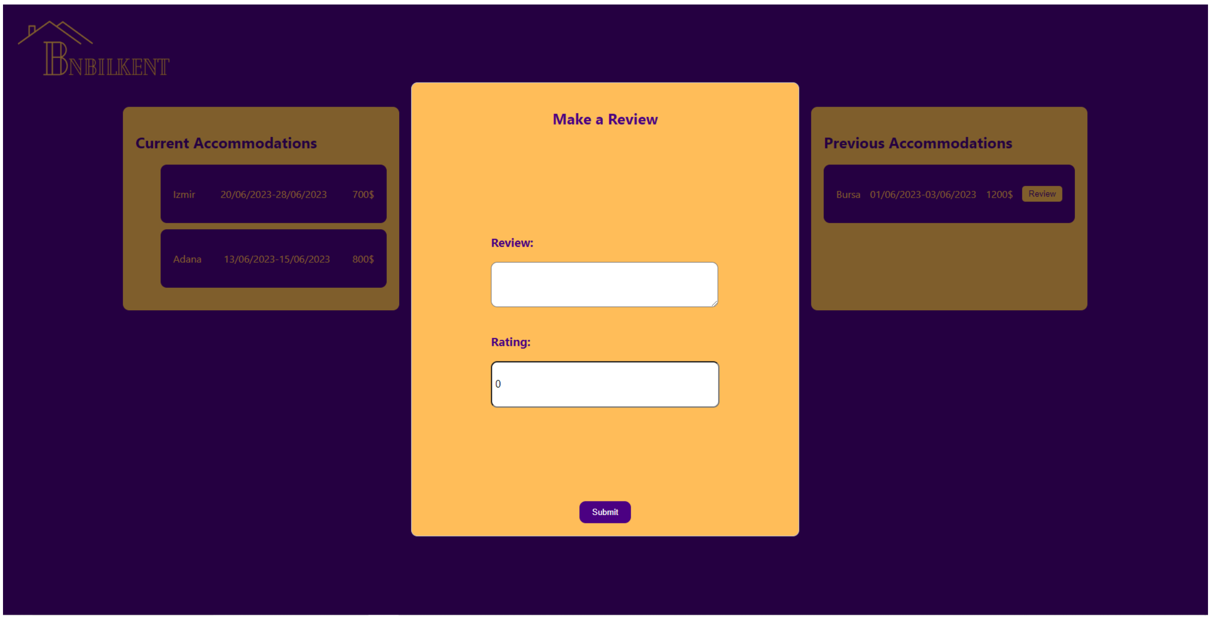
Izmir	20/06/2023-28/06/2023	700$
Adana	13/06/2023-15/06/2023	800$

Previous Accommodations

Bursa	01/06/2023-03/06/2023	1200$	Review

7.17. Traveler Review Pop-up

Through this pop-up a traveler can make a review and give rating to an accommodation that has previously been rented.



Current Accommodations

Izmir 20/06/2023-28/06/2023 700$

Adana 13/06/2023-15/06/2023 800$

Make a Review

Review:

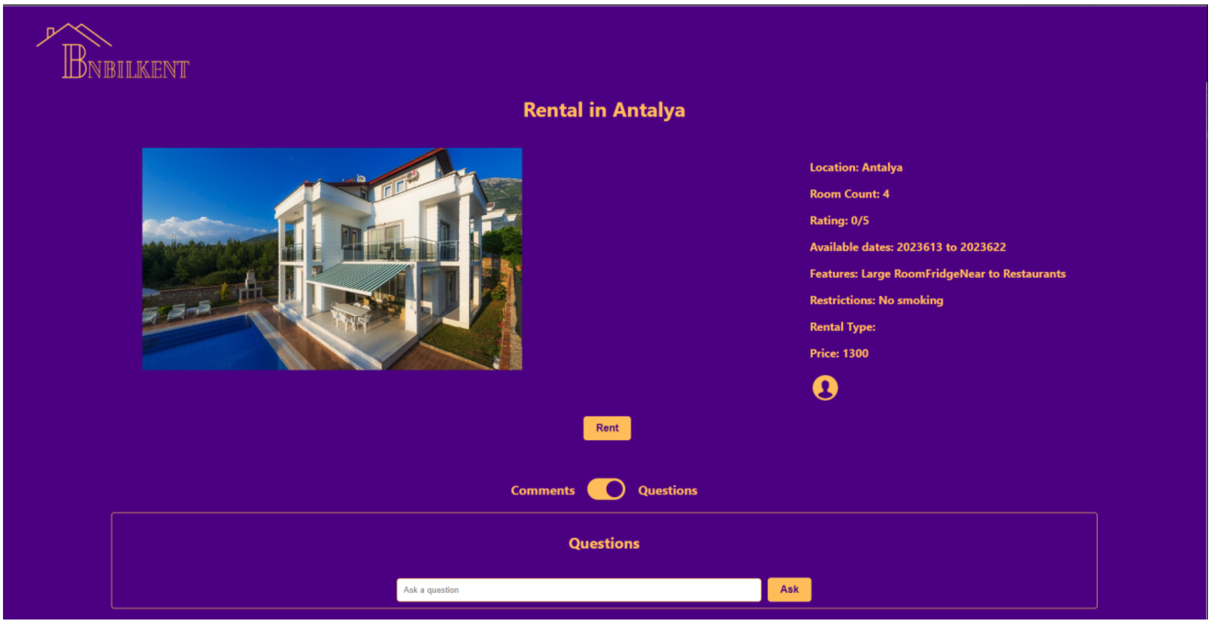
Rating: 0

Previous Accommodations

Bursa 01/06/2023-03/06/2023 1200$

7.18. Traveler Rental Page

In this page, travelers can rent a specified house from the main page by clicking on the Rent button. There is a detailed information section on the right of the page. There is a switch button to see old comments or ask a question to the homeowner.



Rental in Antalya

Location: Antalya

Room Count: 4

Rating: 0/5

Available dates: 2023613 to 2023622

Features: Large RoomFridgeNear to Restaurants

Restrictions: No smoking

Rental Type:

Price: 1300

Questions

Ask a question