CSI 2132 Course Project Report

Our e-hotels project aims to streamline the management of hotel bookings and rental services of the five most well-known hotels chains in North America through a comprehensive web-based application. Leveraging a combination of front-end and back-end technologies, our solution provides a user-friendly interface for both customers and hotel staff to interact with the booking system efficiently. Customers can browse available rooms, hotel staff can oversee bookings and renting, and the administration can apply changes to hotel information stored in the database.

- Technology Stack
    - IDE: intelliJ
    - DBMS: PostgreSQL, PgAdmin as client
    - Web Server: Apache Tomcat
    - Programming Languages:
        - Frontend: HTML, CSS, JavaScript, Bootstrap framework
        - Backend: Java

- Installation Guide
    - Download the application package and extract it on your local machine
    - Open PostgreSQL and execute the DDL scripts on a client such as pgAdmin to create the necessary tables
    - Connect your PostgreSQL to the project through the ConnectionDB file
    - Build the application using Maven and run it (either through the JAR file or the IDE such as intellij)
    - Once the application is running, open a web browser and access the application using the given URL (http://localhost:8080/e-hotels)

- List of DDLs for Database Creation

    -
      ```
      Create Table HotelChain(
      hotelChainID int NOT NULL CHECK (hotelChainID BETWEEN 1 AND 5),
      num_hotels int CHECK (num_hotels BETWEEN 1 AND 1000),
      email VARCHAR(255) CHECK (email like '%@%'),
      street VARCHAR(100) NOT NULL,
      city VARCHAR(100) NOT NULL,
      province VARCHAR(100) NOT NULL,
      PostalCode VARCHAR(6) NOT NULL CHECK (LENGTH(PostalCode)
      BETWEEN 5 AND 6),
      phoneNumber NUMERIC(10,0) Not Null,
      PRIMARY KEY (hotelChainID)
      );
      ```

-
  ```
  CREATE TABLE Hotel (
  hotelID int NOT NULL CHECK (hotelID >= 1 AND hotelID <= 1000),
  hotelChainID int NOT NULL CHECK (hotelChainID BETWEEN 1 AND 5),
  num_rooms int CHECK (num_rooms BETWEEN 1 AND 100),
  email VARCHAR(255) CHECK (email like '%@%'),
  star_rating int CHECK (star_rating >= 1 AND star_rating <= 5),
  street VARCHAR(100) NOT NULL,
  city VARCHAR(100) NOT NULL,
  province VARCHAR(100) NOT NULL,
  PostalCode VARCHAR(6) NOT NULL CHECK (LENGTH(PostalCode)
  BETWEEN 5 AND 6),
  phoneNumber NUMERIC(10,0) Not Null,
  PRIMARY KEY (hotelID, hotelChainID),
  FOREIGN KEY (hotelChainID)
  REFERENCES HotelChain
  );
  ```

-
  ```
  CREATE TABLE Employee (
  SSN_SIN NUMERIC(9,0) PRIMARY KEY CHECK (SSN_SIN > 0),
  hotelID INT NOT NULL CHECK (hotelID BETWEEN 1 AND 1000),
  hotelChainID INT NOT NULL CHECK (hotelChainID BETWEEN 1 AND
  5),
  first_name VARCHAR(50) NOT NULL,
  middle_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
   street VARCHAR(100) NOT NULL,
  city VARCHAR(100) NOT NULL,
  province_state VARCHAR(100) NOT NULL,
  postal_code_zip_code VARCHAR(20) NOT NULL,
  Roles_chef BOOLEAN,
  Roles_maintenanceTechnician BOOLEAN,
  Roles_receptionist BOOLEAN,
  Roles_hotelKeeper BOOLEAN,
  FOREIGN KEY (hotelID, hotelChainID) REFERENCES Hotel(hotelID,
  hotelChainID),
  FOREIGN KEY (hotelChainID) REFERENCES HotelChain(hotelChainID)
  );
  ```

-
  ```
  CREATE TABLE Customer (
  customer_ID NUMERIC(9,0) PRIMARY KEY CHECK (customer_ID > 0),
  street VARCHAR(100) NOT NULL,
  city VARCHAR(100) NOT NULL,
  ```

```
province_state VARCHAR(100) NOT NULL CHECK (province_state IN
('North America')),
postal_code_zip_code VARCHAR(20) NOT NULL CHECK
(LENGTH(postal_code_zip_code) BETWEEN 5 AND 6),
first_name VARCHAR(50) NOT NULL,
middle_name VARCHAR(50),
last_name VARCHAR(50) NOT NULL,
DateofRegistration DATE CHECK (DateofRegistration IS NOT NULL),
CONSTRAINT valid_dateofregistration CHECK (DateofRegistration::TEXT
~ '^\d{4}-\d{2}-\d{2}$')
);
```

-
  ```
  CREATE TABLE BookingArchive (
  booking_ID NUMERIC(9,0) PRIMARY KEY CHECK (booking_ID > 0),
  bookingDate DATE NOT NULL,
  Customer_first_name VARCHAR(50) NOT NULL,
  Customer_middle_name VARCHAR(50),
  Customer_last_name VARCHAR(50) NOT NULL,
  Room_number NUMERIC(5,0) CHECK (Room_number > 0),
  hotelID INT CHECK (hotelID > 0),
  hotelChainID INT CHECK (hotelChainID BETWEEN 1 AND 5)
  );
  ```

-
  ```
  CREATE TABLE RentingArchive (
  renting_ID NUMERIC(9,0) PRIMARY KEY CHECK (renting_ID > 0),
  Date DATE NOT NULL,
  Customer_first_name VARCHAR(50) NOT NULL,
  Customer_middle_name VARCHAR(50),
  Customer_last_name VARCHAR(50) NOT NULL,
  Room_number NUMERIC(5,0) CHECK (Room_number > 0),
  hotelID INT CHECK (hotelID > 0),
  hotelChainID INT CHECK (hotelChainID BETWEEN 1 AND 5)
  );
  ```

-
  ```
  CREATE TABLE Room (
  room_number NUMERIC(5,0) PRIMARY KEY CHECK (room_number > 0),
  HotelID INT CHECK (HotelID BETWEEN 1 AND 1000),
  HotelChainID INT CHECK (HotelChainID BETWEEN 1 AND 5),
  problems_water BOOLEAN,
  problems_electrical BOOLEAN,
  problems_furniture BOOLEAN,
  problems_other VARCHAR(255),
  price DECIMAL(6,2) CHECK (price BETWEEN 100.00 AND 10000.00),
  amenities_tv BOOLEAN,
  ```

```
amenities_wifi BOOLEAN,
amenities_air_con BOOLEAN,
amenities_fridge BOOLEAN,
amenities_toiletries BOOLEAN,
capacities_single BOOLEAN,
capacities_double BOOLEAN,
guest_capacity INT CHECK (guest_capacity BETWEEN 1 AND 5),
sea_view BOOLEAN,
mountain_view BOOLEAN,
extendable BOOLEAN,
FOREIGN KEY (hotelID, hotelChainID) REFERENCES Hotel(hotelID,
hotelChainID),
FOREIGN KEY (hotelChainID) REFERENCES HotelChain(hotelChainID)
);
```

-
  ```
  CREATE TABLE Renting (
  Renting_ID SERIAL PRIMARY KEY,
  rentingDate DATE NOT NULL,
  customer_ID NUMERIC(9,0) NOT NULL,
  room_number NUMERIC(5,0) NOT NULL,
  hotelID NUMERIC(5,0) NOT NULL,
  hotelChainID NUMERIC(5,0) NOT NULL,
  Payment NUMERIC(16,0) NOT NULL,
  FOREIGN KEY (customer_ID) REFERENCES Customer(customer_ID)
  );
  ```

-
  ```
  CREATE TABLE Booking(
  booking_ID NUMERIC(9,0) PRIMARY KEY CHECK (booking_ID > 0),
  bookingDate DATE NOT NULL,
  customer_ID NUMERIC(9,0) not null,
  room_number NUMERIC(5,0) CHECK (room_number > 0),
  hotelID INT CHECK (hotelID > 0),
  hotelChainID INT CHECK (hotelChainID BETWEEN 1 AND 5),
  FOREIGN KEY (customer_ID) REFERENCES Customer(customer_ID)
  );
  ```