uOttawa

L'Université canadienne
Canada's university

CSI 2132 - Database I

Prof: Verena Kantere

Project 1

Group 100

Team member:

Shota Mochizuki: 300333137

Yihan Liao: 300166750

Hanqi Wang: 300252427

Presented to:

Prof: Verena Kantere

Due Date: Mar 31th, 2025

A. The DBMS and the programming languages that we have used in our implementation of the application.

Our application utilizes PostgreSQL as its database management system, with pgAdmin 4 used for managing and operating the database. The backend logic is implemented in Java, and Apache Tomcat is employed as the web server to efficiently handle server-side processing and request management. For the frontend, HTML is primarily used, with some dynamic content handled through JSP (JavaServer Pages) to deliver an interactive user interface. The entire development and integration of the system were carried out using IntelliJ IDEA, which served as our primary integrated development environment (IDE). By combining these technologies, our system provides an intuitive and user-friendly web-based hotel management platform that enables users to search for hotels, check room availability, make reservations, and efficiently manage related data.

B. Specific steps to guide someone to install your applications

Install PostgreSQL and pgAdmin 4

Download the PostgreSQL installer from the official website:
https://www.postgresql.org/download/

Follow the installation instructions and ensure that **pgAdmin 4** is included or install it separately from: https://www.pgadmin.org/download/

After installation, open pgAdmin 4 and connect to your PostgreSQL server.

Install Apache Tomcat

Download Apache Tomcat (version 9 or later) from the official website:
https://tomcat.apache.org/

Extract the downloaded files and configure Tomcat by placing your application files (or the compiled .war file) in the webapps directory.

Install IntelliJ IDEA

Download IntelliJ IDEA (Community or Ultimate Edition) from:
https://www.jetbrains.com/idea/download/

Run the installer and follow the setup instructions.

Launch IntelliJ IDEA and open your Java web application project.

Make sure to configure your project with the appropriate **JDK** and set up **Tomcat** as an application server (via IntelliJ's Run/Debug Configurations).

You can build, run, and test your project directly within IntelliJ.

Install Java Development Kit (JDK)

Download the latest **Java SE Development Kit (JDK)** (version 17 or compatible) from the official Oracle website:
https://www.oracle.com/java/technologies/javase-downloads.html

Follow the installation instructions for your operating system (Windows, macOS, or Linux).

After installation, confirm that Java is properly installed by running the following command in your terminal or command prompt:

bash

```
java -version
```

You should see the installed version printed out.

C. DDLs that create our database

```
DROP TABLE IF EXISTS Renting_archive CASCADE;
DROP TABLE IF EXISTS Booking_archive CASCADE;
DROP TABLE IF EXISTS Renting CASCADE;
DROP TABLE IF EXISTS Booking CASCADE;
DROP TABLE IF EXISTS Customer CASCADE;
DROP TABLE IF EXISTS Employee CASCADE;
DROP TABLE IF EXISTS Room CASCADE;
DROP TABLE IF EXISTS Hotel CASCADE;
DROP TABLE IF EXISTS Hotel_Chain CASCADE;
CREATE TABLE Hotel_Chain (
    chain_name VARCHAR(100) PRIMARY KEY,
    country VARCHAR(50),
    city VARCHAR(50),
    street_number VARCHAR(10),
    unit_number VARCHAR(10),
    zip_code VARCHAR(10),
    number_of_hotels INT CHECK (number_of_hotels >= 0),
    contact_email VARCHAR(100),
    phone_number VARCHAR(20)
);
-- Hotel table
CREATE TABLE Hotel (
    hotel_name VARCHAR(100) PRIMARY KEY,
    chain_name VARCHAR(100) NOT NULL,
    country VARCHAR(50),
    city VARCHAR(50),
    street_number VARCHAR(10),
    unit_number VARCHAR(10),
    zip_code VARCHAR(10),
    star_rating INT CHECK (star_rating BETWEEN 1 AND 5),
    number_of_rooms INT CHECK (number_of_rooms >= 0),
    contact_email VARCHAR(100),
    phone_number VARCHAR(20),
    FOREIGN KEY (chain_name) REFERENCES Hotel_Chain(chain_name) ON DELETE CASCADE
);


-- Room table
CREATE TABLE Room (
    room_number INT,
    hotel_name VARCHAR(100),
    price DECIMAL(10, 2) CHECK (price >= 0),
    amenities TEXT, -- example: "TV, air condition, fridge"
    capacity VARCHAR(20) CHECK (capacity IN ('single', 'double', 'triple', 'family')),
    view_type VARCHAR(20) CHECK (view_type IN ('sea', 'mountain', 'none')),
    expandable BOOLEAN DEFAULT FALSE,
```

```sql
    issues TEXT, -- example: "broken window"
    status VARCHAR(20) CHECK (status IN ('available', 'booked', 'rented', 'maintenance')) DEFAULT 'available',
    PRIMARY KEY (room_number, hotel_name),
    FOREIGN KEY (hotel_name) REFERENCES Hotel(hotel_name) ON DELETE CASCADE
);
-- Employee table
CREATE TABLE Employee (
    SSN VARCHAR(20) PRIMARY KEY,
    first_name VARCHAR(50),
    mid_name VARCHAR(50),
    last_name VARCHAR(50),
    country VARCHAR(50),
    city VARCHAR(50),
    street_number VARCHAR(10),
    unit_number VARCHAR(10),
    zip_code VARCHAR(10),
    role VARCHAR(50),
    hotel_name VARCHAR(100),
    manager_SSN VARCHAR(20),
    FOREIGN KEY (hotel_name) REFERENCES Hotel(hotel_name) ON DELETE SET NULL,
    FOREIGN KEY (manager_SSN) REFERENCES Employee(SSN) ON DELETE SET NULL
);
```

```
-- Customer table
CREATE TABLE Customer (
    customer_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    mid_name VARCHAR(50),
    last_name VARCHAR(50),
    country VARCHAR(50),
    city VARCHAR(50),
    street_number VARCHAR(10),
    unit_number VARCHAR(10),
    zip_code VARCHAR(10),
    ID_type VARCHAR(20) CHECK (ID_type IN ('SSN', 'SIN', 'driving_license')),
    registration_date DATE DEFAULT CURRENT_DATE
);

-- Booking table
CREATE TABLE Booking (
    booking_number SERIAL PRIMARY KEY,
    BookingDate DATE DEFAULT CURRENT_DATE,
    CheckInDate DATE NOT NULL,
    CheckOutDate DATE NOT NULL,
    customer_id INT,
    room_number INT,
    hotel_name VARCHAR(100),
    chain_name VARCHAR(100),
    CONSTRAINT valid_dates CHECK (CheckInDate < CheckOutDate),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE SET NULL,
    FOREIGN KEY (room_number, hotel_name) REFERENCES Room(room_number, hotel_name) ON DELETE SET NULL,
    FOREIGN KEY (chain_name) REFERENCES Hotel_Chain(chain_name) ON DELETE SET NULL
);


CREATE TABLE Renting (
    renting_number SERIAL PRIMARY KEY,
    rent_date DATE DEFAULT CURRENT_DATE,
    checkin_date DATE NOT NULL,
    checkout_date DATE NOT NULL,
    customer_id INT,
    room_number INT,
    hotel_name VARCHAR(100),
    chain_name VARCHAR(100),
    employee_SSN VARCHAR(20),
    CONSTRAINT valid_renting_dates CHECK (checkin_date < checkout_date),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE SET NULL,
    FOREIGN KEY (room_number, hotel_name) REFERENCES Room(room_number, hotel_name) ON DELETE SET NULL,

    FOREIGN KEY (chain_name) REFERENCES Hotel_Chain(chain_name) ON DELETE SET NULL,

    FOREIGN KEY (employee_SSN) REFERENCES Employee(SSN) ON DELETE SET NULL

);


-- Booking_archive table

CREATE TABLE Booking_archive (

    booking_number INT PRIMARY KEY,

    BookingDate DATE,

    CheckInDate DATE,

    CheckOutDate DATE,

    customer_id INT,

    room_number INT,

    hotel_name VARCHAR(100),

    chain_name VARCHAR(100)

);


-- Renting_archive table

CREATE TABLE Renting_archive (

    renting_number INT PRIMARY KEY,

    rent_date DATE,

    checkin_date DATE,
    checkout_date DATE,

    customer_id INT,

    room_number INT,

    hotel_name VARCHAR(100),

    chain_name VARCHAR(100),

    employee_SSN VARCHAR(20),

    CONSTRAINT valid_renting_archive_dates CHECK (checkin_date < checkout_date)

);
```

Table 1 Contents of the video

Requirement	Start timestamp
1	0min00-1min34
2	1min35-2min50
3	2min51-4min16
4	4min17-5min33
5	5min34-6min25
6	6min26-7min01
7	7min02-7min33
8	7min34-8min07
9	8min08-15min55