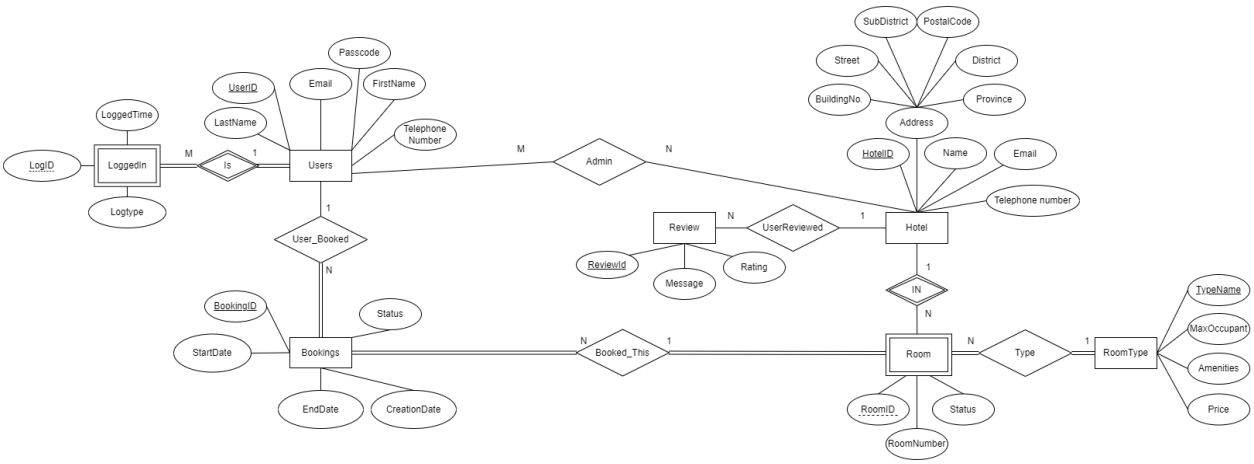
# Final Report

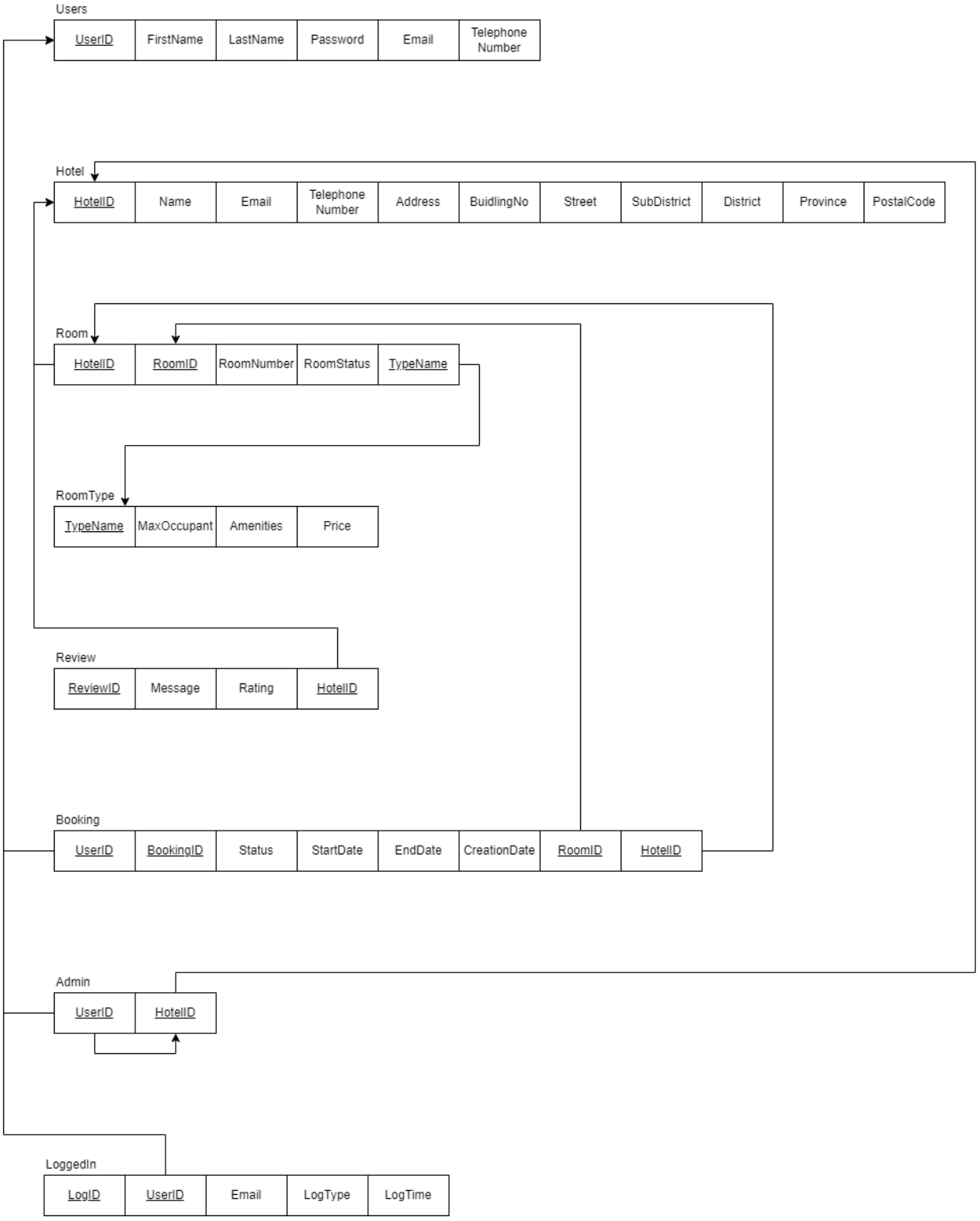
1. Project Name

   Book Local, Sleep Easy

2. ER Diagram



3. Schema Diagram



4. SQL Commands

    1. The system shall allow a user to register by specifying the name, telephone number, email, and password.

```
CREATE OR REPLACE FUNCTION add_user (
    firstname VARCHAR(255),
    lastname VARCHAR(255),
    emaill VARCHAR(255),
    passcode VARCHAR(255),
    telephoneNumber VARCHAR(10))
    RETURNS BOOLEAN
    LANGUAGE plpgsql
    AS
$$
DECLARE
    checkemail VARCHAR(255);
    rn INTEGER;
BEGIN
    SELECT users.email
    INTO checkemail
    FROM users
    WHERE emaill = users.email;

    IF FOUND THEN
        RETURN FALSE;
    END IF;

    INSERT INTO users VALUES
    (DEFAULT, firstname, lastname, emaill, passcode, telephoneNumber);

    RETURN TRUE;

END;
$$
```

| "userid" | "firstname" | "lastname" | "email" | "passcode" | "telephonenumber" |
|---|---|---|---|---|---|
| 1 | "John" | "Doe" | "john.doe@example.com" | "password123" | "1234567890" |
| … | | | | | |
| 6 | "John" | "Doe" | "alice.j@example.com" | "password123" | "1234567890" |

2. After registration, the user becomes a registered user, and the system shall allow the user to log in to use the system by specifying the email and password. The system shall allow a registered user to log out.

```
CREATE OR REPLACE FUNCTION log_in(
    emaill VARCHAR,
    passcodee VARCHAR)
    RETURNS BOOLEAN
    LANGUAGE plpgsql
    AS
$$
DECLARE
    realpasscode VARCHAR(255);
    realuserID INTEGER;
BEGIN
    SELECT u.passcode, u.userID
    INTO realpasscode,realuserID
    FROM users u
    WHERE u.email = emaill;

    IF NOT FOUND THEN
        RETURN FALSE;
    ELSE
        IF realpasscode = passcodee THEN
            INSERT INTO loggedin VALUES (DEFAULT, realuserID,
emaill,'login', now());
            RETURN TRUE;
        ELSE
            RETURN FALSE;
        END IF;
    END IF;
END;
$$


CREATE OR REPLACE PROCEDURE log_out(
    emaill VARCHAR
    )
    LANGUAGE plpgsql
    AS
$$
DECLARE
    realuserID INTEGER;
BEGIN
```

```
    SELECT u.userID
    INTO realuserID
    FROM users u
    WHERE u.email = emaill;
    INSERT INTO loggedin VALUES (DEFAULT, realuserID, emaill,'logout',
now());
    COMMIT;
END;
$$
```

| "logid" | "userid" | "email" | "logtype" | "loggedtime" |
|---|---|---|---|---|
| 1 | 1 | "john.doe@example.com" | "login" | "2024-02-07 07:07:50.973113" |
| 2 | 1 | "john.doe@example.com" | "logout" | "2024-02-07 07:08:16.829738" |

3. After login, the system shall allow the registered user to book up to 3 nights by specifying the date and the preferred hotel. The hotel list is also provided to the user. A hotel information includes the hotel name, address, and telephone number.

```
--Checking if the date specified is <= 3 nights.
CREATE OR REPLACE FUNCTION hotel_list(
    in_startDate DATE,
    in_endDate DATE)
RETURNS TABLE(
    HotelName VARCHAR,
    RoomType VARCHAR,
    buildingNo VARCHAR,
    street VARCHAR,
    subDistrictName VARCHAR,
    districtName VARCHAR,
    province VARCHAR,
    postalCode VARCHAR,
    email VARCHAR
)
LANGUAGE plpgsql
AS $$
BEGIN
    IF (in_endDate - in_startDate) <= 4 THEN
        RETURN QUERY
        SELECT h.name,
               rt.typeName,
               h.buildingNo,
               h.street,
               h.subDistrictName,
               h.districtName,
               h.province,
               h.postalCode,
               h.email
        FROM room R
        JOIN Hotel h ON R.hotelId = h.hotelId
        JOIN RoomType rt ON R.typeName = rt.typeName
        WHERE NOT EXISTS (
            SELECT 1
            FROM Booking b
            WHERE R.roomId = b.roomId
              AND h.hotelId = b.hotelId
              AND ((in_startDate BETWEEN b.startDate AND b.endDate) OR
(in_endDate BETWEEN b.startDate AND b.endDate))
        );
    ELSE
```

```
        RETURN NEXT;
    END IF;
END;
$$;

-- Book Hotel
CREATE OR REPLACE FUNCTION booking_hotel(
    in_startDate DATE,
    in_enddate DATE,
    hotel_id INTEGER,
    room_id INTEGER,
    user_Id INTEGER
)
RETURNS BOOLEAN
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO booking VALUES(user_Id, DEFAULT, 1,in_startdate,
in_enddate, NOW(), room_id, hotel_id);
    RETURN TRUE;
END $$
```

| "user id" | "booking id" | "bookingstatus" | "startdate" | "enddate" | "creation_date" | "room id" | "hotel id" |
|---|---|---|---|---|---|---|---|
| 1 | 1 | "1" | "2024-02-01" | "2024-02-05" | "2024-01-15 10:00:00" | 1 | 1 |
| … | | | | | | | |
| 1 | 6 | "1" | "2023-02-01" | "2023-02-02" | "2024-02-07 07:10:27.561318" | 1 | 1 |

4. The system shall allow the registered user to view his hotel bookings.

```
CREATE OR REPLACE FUNCTION view_hotel_bookings(
    Inuserid INTEGER
)
RETURNS TABLE(
    bookingID INTEGER,
    bookingStatus VARCHAR,
    startDate DATE,
    endDate DATE,
    RoomType VARCHAR,
    Hotelname VARCHAR
    )
LANGUAGE plpgsql
AS $$

BEGIN

RETURN QUERY
SELECT DISTINCT B.bookingID, B.bookingStatus, B.startDate, B.endDate,
R.typeName, H.name
FROM Booking B, users U, Room R, Hotel H
WHERE B.userID = Inuserid and  B.roomID = R.roomID and H.hotelID =
B.hotelID;


END;
$$
```

| "bookingid" | "bookingstatus" | "startdate" | "enddate" | "roomtype" | "hotelname" |
|---|---|---|---|---|---|
| 6 | "1" | "2023-02-01" | "2023-02-02" | "Standard" | "Luxury Hotel" |
| 1 | "1" | "2024-02-01" | "2024-02-05" | "Standard" | "Luxury Hotel" |

5. The system shall allow the registered user to edit his hotel bookings.

```
CREATE OR REPLACE PROCEDURE edit_hotel_bookings(
    booking_ID INTEGER,
    booking_status VARCHAR,
    start_date DATE,
    end_date DATE,
    creationdate TIMESTAMP,
    room_id INTEGER,
    hotel_id INTEGER
)
    LANGUAGE plpgsql
    AS
$$
DECLARE
    realhotelID INTEGER;
BEGIN
    UPDATE booking SET bookingStatus = booking_status,
                       startdate = start_date,
                       enddate = end_date,
                       creation_date = NOW(),
                       roomid = room_id,
                       hotelid = hotel_id

        WHERE bookingID = booking_ID;
        RETURN;
END;
$$
```

| "userid" | "bookingid" | "bookingstatus" | "startdate" | "enddate" | "creation_date" | "roomid" | "hotelid" |
|---|---|---|---|---|---|---|---|
| 2 | 2 | "1" | "2024-03-01" | "2024-03-10" | "2024-02-01 12:30:00" | 3 | 2 |
| … | | | | | | | |
| 1 | 1 | "2" | "2024-03-01" | "2024-03-10" | "2024-02-07 07:13:42.264409" | 2 | 1 |

6. The system shall allow the registered user to delete his hotel bookings.

```
CREATE OR REPLACE PROCEDURE delete_hotel_bookings(
    userID INTEGER, booking_ID INTEGER
)
LANGUAGE plpgsql
AS $$
BEGIN
    DELETE FROM booking B where booking_ID = B.bookingID;
END;
$$
```

| "userid" | "bookingid" | "bookingstatus" | "startdate" | "enddate" | "creation_date" | "roomid" | "hotelid" |
|---|---|---|---|---|---|---|---|
| 2 | 2 | "1" | "2024-03-01" | "2024-03-10" | "2024-02-01 12:30:00" | 3 | 2 |
| … | | | | | | | |
| 1 | 6 | "1" | "2023-02-01" | "2023-02-02" | "2024-02-07 07:10:27.561318" | 1 | 1 |

7. The system shall allow the admin to view any hotel bookings.

```
CREATE OR REPLACE FUNCTION admin_view_booking(
    Inuserid INTEGER
)
RETURNS TABLE(
    bookingID INTEGER,
    firstname VARCHAR,
    lastname VARCHAR,
    bookingStatus VARCHAR,
    startDate DATE,
    endDate DATE,
    RoomType VARCHAR
    )
LANGUAGE plpgsql
AS $$
DECLARE
    userhotelID INTEGER;
BEGIN
SELECT A.hotelID
INTO userhotelID
FROM Admin A
WHERE InuserID = A.UserID;

IF NOT FOUND THEN
    RETURN NEXT;
END IF;

RETURN QUERY
SELECT DISTINCT B.bookingID, U.firstname, U.lastname, B.bookingStatus,
B.startDate, B.endDate, R.typeName
FROM Booking B, users U, Room R
WHERE B.userID = U.userID and B.hotelID = userhotelID and B.roomID =
R.roomID;


END;
$$
```

| "bookingid" | "firstname" | "lastname" | "bookingstatus" | "startdate" | "enddate" | "roomtype" |
|---|---|---|---|---|---|---|
| 2 | "Jane" | "Smith" | "1" | "2024-03-01" | "2024-03-10" | "Deluxe" |

8. The system shall allow the admin to edit any hotel bookings.

```
CREATE OR REPLACE PROCEDURE admin_edit_booking(
    booking_ID INTEGER,
    booking_Status VARCHAR,
    start_date DATE,
    end_date DATE,
    creationdate TIMESTAMP,
    room_id INTEGER,
    edituser INTEGER
)
    LANGUAGE plpgsql
    AS
$$
DECLARE
    realhotelID INTEGER;
BEGIN
    SELECT Admin.hotelID
    INTO realhotelID
    FROM Admin
    WHERE Admin.userID = edituser;
    IF NOT FOUND THEN
        RETURN;
    ELSE
        UPDATE booking set bookingStatus = booking_Status,
                           startdate = start_date,
                           enddate = end_date,
                           creation_date = NOW(),
                           roomid = room_id

        where bookingID = booking_ID;
        RETURN;
    END IF;
END;
$$
```

| "userid" | "bookingid" | "bookingstatus" | "startdate" | "enddate" | "creation_date" | "roomid" | "hotelid" |
|---|---|---|---|---|---|---|---|
| 3 | 3 | "1" | "2024-04-15" | "2024-04-20" | "2024-03-01 08:45:00" | 5 | 3 |
| … | | | | | | | |
| 2 | 2 | "2" | "2024-03-01" | "2024-03-10" | "2024-02-07 07:17:28.3894" | 4 | 2 |

9. The system shall allow the admin to delete any hotel bookings.

```
CREATE OR REPLACE PROCEDURE admin_delete_booking(
    adminuserID INTEGER,
    deletebookingID INTEGER
)
    LANGUAGE plpgsql
    AS
$$
DECLARE
    adminId INTEGER;
BEGIN
    SELECT admin.userId
    INTO adminId
    FROM admin
    WHERE admin.userId = adminuserId;

    IF NOT FOUND THEN
        RETURN;
    ELSE
        DELETE FROM booking WHERE booking.bookingId = deletebookingID;
        RETURN;
    END IF;
END;
$$
```

| "userid" | "bookingid" | "bookingstatus" | "startdate" | "enddate" | "creation_date" | "roomid" | "hotelid" |
|---|---|---|---|---|---|---|---|
| 3 | 3 | "1" | "2024-04-15" | "2024-04-20" | "2024-03-01 08:45:00" | 5 | 3 |
| … | | | | | | | |
| 1 | 6 | "1" | "2023-02-01" | "2023-02-02" | "2024-02-07 07:10:27.561318" | 1 | 1 |

5. Complex Query

แสดง ราคาเฉลี่ยของ hotel ที่อยู่ในจังหวัด กรุงเทพ ที่มี rating มากกว่า 4.0 มา 3 อันดับแรก โดยเรียงจากมากไปน้อย โดยนำข้อมูล rating จาก Review ที่มี hotelId ตรงกับ hotelId ที่ต้องการหา

```sql
SELECT h.name AS HotelName, ROUND(AVG(rt.price),2) AS AveragePrice
FROM Hotel h
JOIN Room ro ON h.hotelId = ro.hotelId
JOIN Review re ON h.hotelId = re.hotelId
JOIN RoomType rt ON ro.typeName = rt.typeName
WHERE h.province = 'Bangkok' AND re.rating > 4.0
GROUP BY h.hotelId, h.name
ORDER BY AVG(rt.price) DESC
LIMIT 3;
```

| "hotelname" | "averageprice" |
|---|---|
| "Cozy Inn" | "200.00" |
| "Luxury Hotel" | "150.00" |
| "Mountain Lodge" | "125.00" |

6. Document-based Design Schema

```
{
  "User": {
    "properties": {
      "_id": { "bsonType": "objectId" },
      "firstName": { "bsonType": "string" },
      "lastName": { "bsonType": "string" },
      "email": { "bsonType": "string", "unique": true },
      "passcode": { "bsonType": "string" },
      "telephoneNumber": { "bsonType": "string", "pattern": "/^[0-9]{10}$/" },
      "isAdmin": { "bsonType": "bool" }
    },
    "required": ["_id", "firstName", "lastName", "email", "passcode",
"telephoneNumber", "isAdmin"]
  },

"Admin": {
    "properties": {
      "_id": { "bsonType": "objectId" },
      "userId": { "bsonType": "objectId" },
      "hotelId": { "bsonType": "objectId" }
    },
    "required": ["_id", "userId", "hotelId"],
  },

  "Hotel": {
    "properties": {
      "_id": { "bsonType": "objectId" },
      "name": { "bsonType": "string" },
      "telephoneNumber": { "bsonType": "string", "pattern": "/^[0-9]{10}$/" },
      "email": { "bsonType": "string" },
      "buildingNo": { "bsonType": "string" },
      "street": { "bsonType": "string" },
      "subDistrictName": { "bsonType": "string" },
      "districtName": { "bsonType": "string" },
      "province": { "bsonType": "string" },
      "postalCode": { "bsonType": "string", "pattern": "/^[0-9]{5}$/" },
      "rooms": {
        "bsonType": "array",
        "items": {
          "bsonType": "object",
          "properties": {
            "_id": { "bsonType": "objectId" },
            "roomNumber": { "bsonType": "int" },
```

```
            "status": { "bsonType": "string" },
            "type": { "bsonType": "string" }
          },
          "required": ["_id", "roomNumber", "status", "type"]
        }
      }
    },
    "required": ["_id", "name", "telephoneNumber", "buildingNo",
"subDistrictName", "districtName", "province", "postalCode"]
  },

  "RoomType": {
    "properties": {
      "_id": { "bsonType": "objectId" },
      "typeName": { "bsonType": "string" },
      "maxOccupant": { "bsonType": "int" },
      "amenities": { "bsonType": "string" },
      "price": { "bsonType": "int" }
    },
    "required": ["_id", "typeName", "maxOccupant", "amenities", "price"]
  },

  "Booking": {
    "properties": {
      "_id": { "bsonType": "objectId" },
      "bookingStatus": { "bsonType": "string", "enum": ["pending", "confirmed",
"cancelled"] },
      "startDate": { "bsonType": "date" },
      "endDate": { "bsonType": "date" },
      "creation_Date": { "bsonType": "date" },
      "user": { "bsonType": "objectId" },
      "hotel": { "bsonType": "objectId" },
      "room": { "bsonType": "objectId" }
    },
    "required": ["_id", "bookingStatus", "startDate", "endDate", "creation_Date",
"user", "hotel", "room"]
  },

  "loggedin": {
    "properties": {
      "_id": { "bsonType": "objectId" },
      "userId": { "bsonType": "objectId" },
      "email": { "bsonType": "string" },
      "logType": { "bsonType": "string", "enum": ["login", "logout"] },
      "loggedTime": { "bsonType": "date" }
```

```
    },
    "required": ["_id", "userId", "email", "logType", "loggedTime"]
  }
}
```