Muhammad Irfan Jamil

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

[Document title]

Table of Contents

[1. Picker – App 2](#_Toc212265177)

[1.1. UI Flow 2](#_Toc212265178)

[2. Database 5](#_Toc212265179)

[2.1. Schema Overview 5](#_Toc212265180)

[2.2. Table 1 – User 5](#_Toc212265181)

[2.3. Table 2 – Session 6](#_Toc212265182)

[2.4. Table 3 – Session Restaurant 7](#_Toc212265183)

[2.5. Folder Structure 8](#_Toc212265184)

# Picker – App

# UI Flow

The login page is for users to login using their username and password.

Only logged in users are able to “View Sessions” that are exclusively tagged to the individual users.

A screenshot of a computer

AI-generated content may be incorrect.

Note that the insert script has 2 default users:

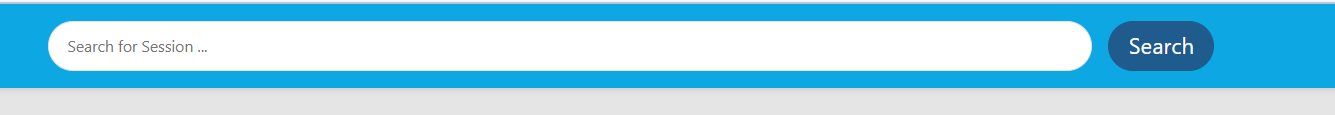
|  |  |
| --- | --- |
| Username | Password |
| john | 3M1j9xRGzu |
| susan | tWtEurziU2 |

Once the user has logged in, they are able to see the paginated sessions.

Here there are a few functionalities available for the user:

1. **Search for a Session**

A User can search for an existing Session using its containing session name using the search bar at the top of the page. This will result in a filtered list of session.



1. **Create a Session**

By clicking on the “Create Session” button, the user will be redirected to an input form that will help to create a new session. However, the user is able to cancel the session creation by clicking on the “Back to Session List” button. A green rectangular sign with white text

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

1. **View an Active Session**

The user can view an active session by clicking on the “View” button, this will redirect the user to a view that list the associated session restaurants. Here, the user can invite anonymous guests to enter the session by copying and sending the url address over to the guests. Both the host user and guests can add in a restaurant option by clicking on the “Create Restaurant” button. However, non-hosting users will not be able to join this session activity.

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

1. **Adding a restaurant**

Below is the input form for the host user or guests to add in a restaurant of their choice. The user/guest can confirm this by clicking the “Add” button. Or choose to cancel by clicking the “Back to Session” button.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Close & Generate a Session**

At the end of the session activity, the host user can close the session and allow the application to pick a random restaurant as the result by clicking on the “Close & Generate” button.

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

1. **Paginate through the list of Sessions or list of Session Restaurants**

In both view pages, the user / guests are able to paginate through the list according to the page size chosen.

A white rectangular object with blue arrows

AI-generated content may be incorrect.

# Database

# Schema Overview

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Purpose** | **Primary key** | **Foreign Keys** |
| user | Stores user information such as login credentials, status and role | id | - |
| session | Tracks user session | id | user\_id -> user.id |
| session\_restuarant | Maps sessions to restaurants | id | session\_id -> session.id |

# Table 1 – User

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints / Notes** |
| id | BIGINT(20) | Unique identifier for each user | Primary Key, Not Null, Auto Increment |
| user\_id | VARCHAR(100) |  | Default: Null |
| first\_name | VARCHAR(100) | User’s first name | Default: Null |
| last\_name | VARCHAR(100) | User’s last name | Default: Null |
| username | VARCHAR(100) | Unique username for login | Not Null, Unique |
| password | VARCHAR(255) | User’s password | Not Null |
| email | VARCHAR(150) | User’s email | Default: Null |
| profile\_image\_url | VARCHAR(255) | User’s profile image URL | Default: Null |
| last\_login\_date | DATETIME | Last login timestamp | Default: Null |
| last\_login\_date\_display | DATETIME | Last login timestamp to display | Default: Null |
| join\_date | DATETIME | Date when user joined | Default: Null |
| role | VARCHAR(50) | User role | Default: Null, Comment ‘e.g. ROLE\_USER OR ROLE\_ADMIN’ |
| authorities | BLOB | User’s permissions | Default: Null, Comment ‘Serialized Java String[] of authorities’ |
| is\_active | BOOLEAN | Whether the user is active | Default: TRUE |
| is\_not\_locked | BOOLEAN | Whether the account is locked | Default: TRUE |

# Table 2 – Session

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints / Notes** |
| id | BIGINT(20) | Unique identifier for each session | Primary Key, Not Null, Auto Increment |
| link | VARCHAR(255) | Session URL | Default: Null |
| name | VARCHAR(255) | Session’s name | Default: Null |
| description | VARCHAR(255) | Session’s description | Default: Null |
| image\_url | VARCHAR(255) | Session’s image URL | Default: Null |
| active | BIT | Whether the session is active | Default: 1 |
| date\_created | DATETIME(6) | Timestamp when the session was created | Default: Null |
| last\_updated | DATETIME(6) | Timestamp when the session was last updated | Default: Null |
| result | VARCHAR(255) | Session’s result | Default: Null |
| user\_id | BIGINT(20) | User’s id | Foreign Key -> user.id, Not Null |

# Table 3 – Session Restaurant

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints / Notes** |
| id | BIGINT(20) | Unique identifier for each record | Primary Key, Not Null, Auto Increment |
| name | VARCHAR(255) |  | Default: Null |
| description | VARCHAR(255) |  | Default: Null |
| active | BIT |  | Default: 1 |
| date\_created | DATETIME(6) | Timestamp when the record was created | Default: Null |
| last\_updated | DATETIME(6) | Timestamp when the record was last updated | Default: Null |
| session\_id | BIGINT(20) | Session’s id | Foreign Key -> session.id, Not Null |

# Folder Structure

There are namely 4 main folders:

1. Starter Files and Documentation
2. Angular Source code
3. Spring Boot Source code
4. Kafka Spring Boot Source code

Please see folder “\Restaurant Picker App\Starter Files and Documentation\” to find the Table Creation and Sample Data.sql script.

A screenshot of a computer

AI-generated content may be incorrect.