Tehnologii folosite:

* Backend using IntelliJ:
  + Spring Boot
  + Spring Data (JPA Repository)
  + Spring MVC
  + Spring Security
  + MySQL DB
* Testing Backend functionality:
  + Postaman
* Frontend using IntelliJ or Visual Code:
  + Angular

Database Model:

* Users
  + Admin
    - Id
    - Account name
    - Password (passwordEncoder -> BCryptPasswordEncoder)
  + User
    - Id
    - First name
    - Last name
    - Email(used for login)
    - Password (passwordEncoder -> BCryptPasswordEncoder)
* Tables
  + - Id
    - Size (total available seats)
    - No of seats reserved (>= ½ size) (people attending)
    - Time frame for reservation (no overlapping)
* **Restaurants Locations(optional)**
  + Tables (One to Many -> Seats)
    - Id
    - Size (total available seats)
    - No of seats booked (>= ½ size)
    - Date
    - Time frame for reservation (no overlapping)
    - Interval for reservation (cum ar fi intre 12-23)
    - **Status (for selected date) Available, Fully Booked, etc (optional)**
  + **Seats (Many to One -> Tables)**
    - Id
    - Select / Unselect
  + **Time for reservation (Many to One -> Tables)**
    - Id
    - Value (00-23)
* **Reviews (optional)**
  + - **Id**
    - **Content**
    - **Username**
    - **Date posted (instant time)**

User entities functionality:

* User
  + Create account
  + Register
    - See all tables and available seats
    - See the reservation interval and choose time frame for reservation
      * **No time gaps larger than 30 minutes (optional)**
    - Make table reservation
      * Set no of people attending
    - Choose time frame for reservation
    - **Cancel (Delete) reservation with 1 hour or more before start (optional)**
    - Modify reservation (change table. reserved seats, time frame -> see time gaps)
  + **Leave a review for services (optional)**
  + Logout
    - **Ask for confirmation if there are unsaved changes (optional)**
    - **Ask for logout (optional)**
* Admin
  + Create account (how ?? maybe un admin initial care sa valideze un alt admin?)
  + **Validate user accounts before first login (optional)**
  + Create, Read, Update, Delete
    - Tables, seats
    - **Restaurants / locations -> tables, seats etc (optional)**
    - Users
  + Logout
  + Access to different pages then users:
    - Registered users
    - Table management
    - **Restaurants / Locations management (optional)**

MVC functionality:

1. Landing page:
   1. With was the app is about
   2. Max no of tables available etc
   3. **No. of restaurants available for booking (optional)**
   4. **Clients testimonial (face, text, name) -> should be a registered user (happy about service, quick reservation, and cancellation etc) -> (optional)**
2. Menu (landing page):
   * 1. Login button - Input fields cu Email(used for login), Password
     2. Signup button - Input fields cu First name, Last name, Email(used for login), Password, **Reenter password (optional)**
     3. **Available locations/restaurants with a short individual presentation page (optional)**
     4. **See customer reviews (optional)**
        1. **By location (optional)**
3. **Restaurants / Locations (optional)**
   1. Click and select restaurant
4. Reserve table
   1. Click and select table
   2. Enter no of seats to reserve
   3. Show reserved time frame (in the selected day)
   4. Enter date to reserve
   5. Enter time frame to reserve
   6. Validate reservation
   7. UI: ar fi misto avem o reprezentare grafica cu amplasarea in locatie a meselor. Dam click -> un calendar unde selectam data in care am dori sa rezervam (putem selecta o alta data), apoi ar trebuie sa ne arate orele intre care este rezervata masa in data respectiv si sa ne apara un input field in care ca completam, nr de scaune (validator if no of seats este mai mic de ½ din capacitate), ora de inceput si ora de sfarsit dorita si buton de confirmare;
5. My Reservation (should check if the authenticated user has reservation sau mergem pe if null show a message -> first make a reservation ?)
   1. Summary of the reservation
   2. Edit / Delete reservation
      1. Gen butoane Update – Delete
6. Leave a review
   1. Plain text editor
   2. **Possibility to add photos to review (optional)**
   3. **Show review in the landing page (optional)**
7. CRUD pentru Users, Admins:

* Add/update/delete users/admin
* Basic UI interface for login/logout/register (after button press)

1. CRUD for Tables:

* ShowAll/ShowById -> User
* ShowAll/ShowById/Add/remove/delete/Update -> Admin
* **See all reservations for a specific table (date->time, user, seats reserved) -> Admin (optional)**
* **See all reservations for a specific table (date->time) -> User (optional)**

1. CRUD for seats

* Select/Unselect reserved seats

1. Workflow BackEnd:
   1. Client ->
   2. Controller (Users, Tables, **Reviews,** **Locations (optional)**) ->
      1. Security
         1. UserDetailsImpl
   3. Service (DB CRUD methods) -> MySQL DB ->
   4. Repository (Users, Tables **Reviews,** **Locations (optional)**) ->
   5. Model (Users, Tables **Reviews,** **Locations (optional)**)
2. UI template?! Ca nu cred ca folosim CSS, cum facem designul sa arate cat de cat omenesc? Plain Html sau cautam ceva?