Complete Requirements:

Users:

1. Users and the owner.
2. Authenticate owner.

Users

3. Creating a Booking for table :

Users can reserve a table by providing date, time, party size, contact details.
If Booking is in Waiting status, ask customer to ConfirmationCode.
Provide a unique Confirmation Code and status back to the customer.

Users l Edit/Cancel a Booking :
Users can edit an existing Booking using Confirmation Code.
Users can edit date, time, and party size.
Use same Confirmation Code and return new status.
Users can cancel an existing Booking using Confirmation Code.

Owner

4. Login:

Owner can login using email and password.
Registration form is optional.

5. View Booking :

Owner can view list of Booking s and can select Booking item for more details.

6. View Seating Area

Owner can view seating area (tables) in a list form.
Each item can have Confirmation Code, Size, Status, Since fields.
On clicking ConfirmationCode, open Booking detail screen.

7. Create and Edit a Booking :

Same as Customer Create and Edit Booking flows.

8. Profile & Settings

Owner can view/edit restaurant profile details like Name, Contact, Email, Address etc.
Owner can update settings like Auto Assign, Restaurant Open/Closing days and times etc.

9. Owner l Assign Table:

Open Booking detail screen from list of Booking s.
On clicking Assign Table, open seating map and select table.

10. Owner l Auto Assign Table:

If Auto Assign is enabled, system should assign the table to a new Booking automatically.

11. Owner l Change Table:

Owner should be able to change the table for a Booking .

12. Owner | View Contact List:
Owner can view contact list of all the Users and their past Booking s.

Minimum needs for development:

User Interface Design:
Responsive. Test on desktop, mobile, and tablet. You can use a CSS framework like Bootstrap.
Use new HTML5 and CSS3 features like semantics, fonts, transitions, transformations.

JavaScript:
Single Page Application using AngularJS
Multiple Views and templates.

Back-end (for JAVA):
Data exchange: JSON
J2EE Servlets using JSON transformation library like Google Gson, JSON.org etc.
Jersey + Jackson (preferred)
MySQL, Tomcat (or any other J2EE server)
Build, Optimization, and Testing
grunt
Protractor for e2e testing.