

User Class

```
package com.restaurant.pojo;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

import lombok.Data;

@Entity
@Data
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long userId;
    private String name;
    @Column(unique = true)
    private String email;
    private String password;
    @Column(unique = true)
    private String mobileNumber;
    public Long getUserId() {
        return userId;
    }
    public void setUserId(Long userId) {
        this.userId = userId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getMobileNumber() {
        return mobileNumber;
    }
}
```

```

        public void setMobileNumber(String mobileNumber) {
            this.mobileNumber = mobileNumber;
        }
        @Override
        public String toString() {
            return "User [userId=" + userId + ", name=" + name + ", email=" +
email + ", password=" + password
                                + ", mobileNumber=" + mobileNumber + "];"
        }
        public User(Long userId, String name, String email, String password, String
mobileNumber) {
            super();
            this.userId = userId;
            this.name = name;
            this.email = email;
            this.password = password;
            this.mobileNumber = mobileNumber;
        }
        public User() {
            super();
            // TODO Auto-generated constructor stub
        }
    }
}

```

```
User Repository
package com.restaurant.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import com.restaurant.pojo.User;

public interface UserRepo extends JpaRepository<User, Long> {

    User findByEmail(String email);

    User findByMobileNumber(String mobileNumber);

    User findByEmailAndPassword(String email, String password);
}
```

User Service

```
package com.restaurant.service;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
import com.restaurant.pojo.OrderDetails;
```

```
import com.restaurant.pojo.User;
```

```
import com.restaurant.repo.UserRepo;
```

```
@Service
```

```
public class UserService {
```

```
    @Autowired
```

```
    UserRepo userRepo;
```

```
    public User getUserByEmail(String email) {  
        return userRepo.findByEmail(email);  
    }
```

```
    public User getUserByMobileNumber(String mobileNumber) {  
        return userRepo.findByMobileNumber(mobileNumber);  
    }
```

```
    public void registerUser(User user) {  
        userRepo.save(user);  
    }
```

```
    public String checkAdminCredentials(String email, String password) {  
        User user = userRepo.findByEmailAndPassword(email, password);  
        if (user == null)  
            return "invalid";  
        else  
            return user.getEmail();  
    }
```

```
    public User updateUser(User user) {  
        return userRepo.save(user);  
    }
```

```
    public List<User> getAllUsers(){  
        return userRepo.findAll();  
    }
```

```
}
```

User Controller

```
package com.restaurant.controller;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.http.MediaType;
```

```
import org.springframework.web.bind.annotation.CrossOrigin;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.PutMapping;
```

```
import org.springframework.web.bind.annotation.RequestBody;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
import com.restaurant.pojo.OrderDetails;
```

```
import com.restaurant.pojo.User;
```

```
import com.restaurant.service.UserService;
```

```
@RestController
```

```
@CrossOrigin(origins = "*")
```

```
@RequestMapping("/user")
```

```
public class UserController {
```

```
    @Autowired
```

```
    UserService userService;
```

```
    @PostMapping("/register")
```

```
    public String registerUser(@RequestBody User user) {
```

```
        System.out.println(user.getEmail()+" "+user.getMobileNumber()+"
```

```
        "+user.getName()+" "+user.getPassword());
```

```
        User testUser1 = userService.getUserByEmail(user.getEmail());
```

```
        if (testUser1 != null)
```

```
            return "Email already exists!";
```

```
        User testUser2 =
```

```
        userService.getUserByMobileNumber(user.getMobileNumber());
```

```
        if (testUser2 != null)
```

```
            return "Mobile Number already exists!";
```

```
        userService.registerUser(user);
```

```
        return "Registered Successfully! Login to continue";
```

```
    }
```

```
    @PostMapping("/login")
```

```
    public String loginUserCheck(@RequestBody User user) {
```

```
        return
```

```
        userService.checkAdminCredentials(user.getEmail(),user.getPassword());
```

```
    }
```

```
    @GetMapping("/email/{userEmail}")
```

```
    public User getUserByEmail(@PathVariable String userEmail) {
```

```

        System.out.println(userEmail);
        userEmail= userEmail.substring(1, userEmail.length()-1);
        return userService.getUserByEmail(userEmail);
    }
    // http://localhost:9002/user
    @GetMapping(value="", produces=MediaType.APPLICATION_JSON_VALUE)
    public List<User> getAllUsers(){
        return userService.getAllUsers();
    }

    @PutMapping("/update")
    public String updateUser(@RequestBody User user) {
        if(userService.updateUser(user)!=null)
            return "done";
        return "0";
    }
}

```