



# WED-16.30-6 PRESENTATION

VERDOUW, Midori (s3575912)

PHAM, Van (s3788106)

TJIONG, Julian (s3786866)

CHEONG, Hon Khuin Jonathan (s3642842)

SONG, Jason (s3744335)



# 1. Making a Booking (Front End)

React App

localhost:1337/business/1/schedule

Home About Contact Us Login Sign Up

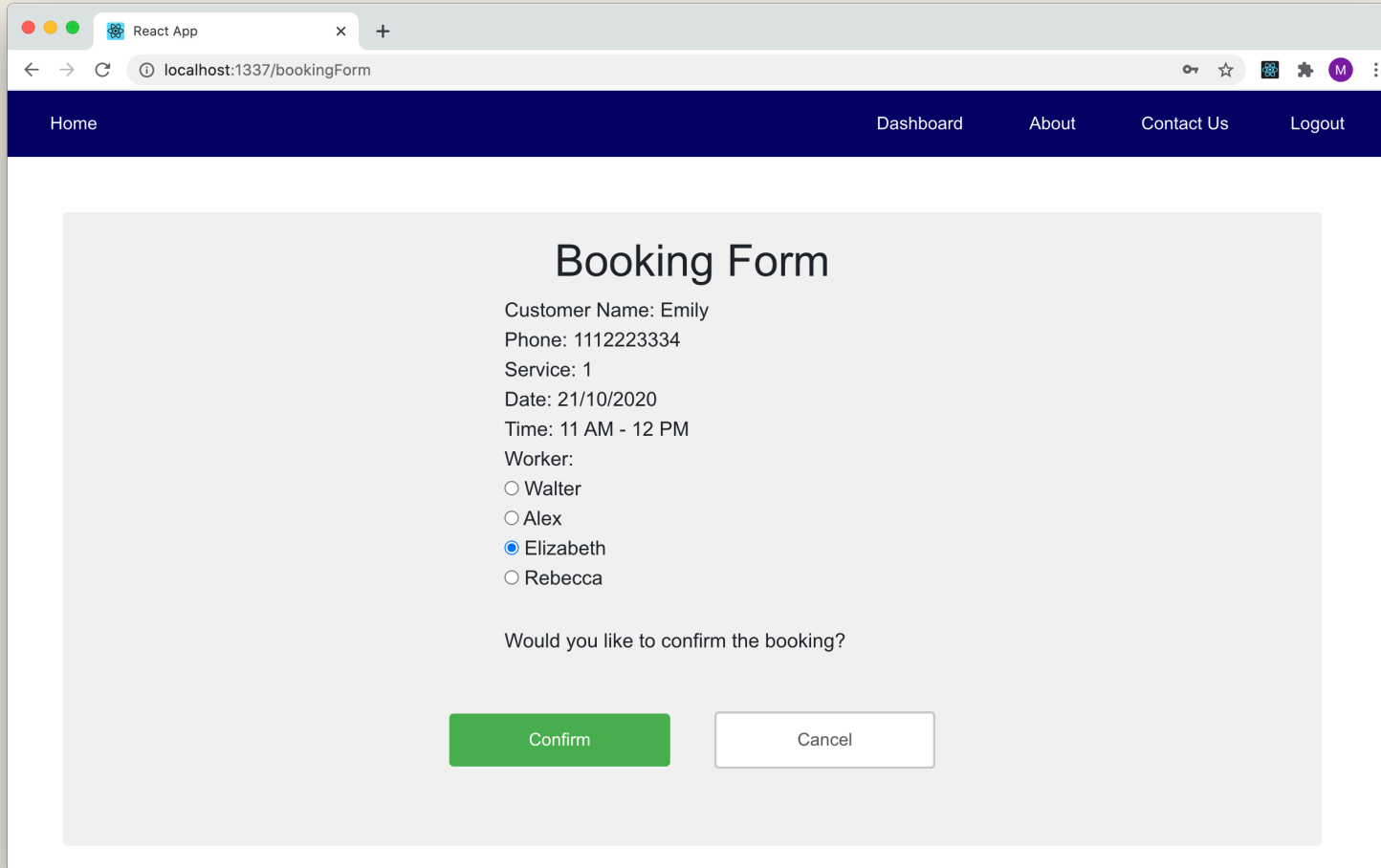
## Business Name 1 Booking

October 2020

	18 SUN	19 MON	20 TUE	21 WED	22 THU	23 FRI	24 SAT
9:00 AM	9 AM - 10 AM	9 AM	9 AM	9 AM - 10 AM	9 AM	9 AM	9 AM
10:00 AM	10 AM - 11 AM	10 AM	10 AM	10 AM - 11 AM	10 AM	10 AM	10 AM
11:00 AM	11 AM - 12 PM	11 AM	11 AM	11 AM - 12 PM	11 AM - 12 PM	11 AM	11 AM
12:00 PM	12 PM - 1 PM	12 PM	12 PM	12 PM - 1 PM	12 PM - 1 PM	12 PM	12 PM
1:00 PM	1 PM - 2 PM	1 PM	1 PM	1 PM - 2 PM	1 PM - 2 PM	1 PM	1 PM

- Business schedule page
- Each timeslot must take the availability of all workers into account
- Available & Unavailable timeslots

# 1. Making a Booking (Front End)



React App

localhost:1337/bookingForm

Home Dashboard About Contact Us Logout

## Booking Form

Customer Name: Emily  
Phone: 1112223334  
Service: 1  
Date: 21/10/2020  
Time: 11 AM - 12 PM  
Worker:  
☐ Walter  
☐ Alex  
☒ Elizabeth  
☐ Rebecca

Would you like to confirm the booking?

Confirm Cancel

- Booking form
- Redux to keep user information
- Use action to communicate with backend

## 2. Making API for Make Booking (Back End)

```
public class BookingForm {  
    private Long id;  
    private Long worker_id;  
    private Long cust_id;
```

- BookingForm class for @RequestBody to receive Json from FrontEnd
- Call the editScheduleService

```
@PostMapping("")  
@ResponseBody  
public void editSchedule(@RequestBody BookingForm bookingForm) {  
    Long scheduleId = bookingForm.getId();  
    Long workerId = bookingForm.getWorker_id();  
    Long custId = bookingForm.getCust_id();  
  
    editScheduleService.editSchedule(scheduleId, custId);  
}
```

## 2. Making API for Make Booking (Back End)

```
public void editSchedule(Long id, Long custId) { scheduleRepository.updateSchedule(custId, id); }
```

```
@Transactional
@Modifying
@Query("UPDATE WorkerSchedule s SET s.booked = TRUE, s.custId = :custId WHERE s.id = :id")
public void updateSchedule(@Param("custId") Long custId, @Param("id") Long id);
```

POST ▼ http://localhost:8080/api/booking

Params Authorization Headers (8) **Body** ●

☐ none ☐ form-data ☐ x-www-form-urlencoded

```
1 {
2   "id" : "5",
3   "cust_id" : "19191919",
4   "worker_id" : "2"
5 }
```

■ Learnt things on @Query

### 3. Login Functionality (Back End)

```
public class LoginForm {  
    private String username;  
    private String password;  
}
```

- LoginForm for @RequestBody
- Checks if credentials are correct

```
@PostMapping("")  
@ResponseBody  
public ResponseEntity<?> login(@RequestBody LoginForm loginForm) {  
    String personId = loginForm.getUsername();  
    String password = loginForm.getPassword();  
  
    Person person = personService.findByPersonIdentifier(personId);  
    String passwordTru = person.getPassword();  
  
    if(passwordTru.equals(password)){  
        return new ResponseEntity<Person>(person, HttpStatus.OK);  
    }else{  
        return new ResponseEntity<String>(body: "Wrong credentials mate", HttpStatus.OK);  
    }  
}
```

### 3. Login Functionality (Back End)

```
1 {
2   "username" : "EMILY",
3   "password" : "123"
4 }
5 }
```

dy Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ▼

```
1 {
2   "id": 11,
3   "name": "Emily",
4   "username": "EMILY",
5   "password": "123",
6   "desc": "customer desc",
7   "mobileNum": "1112223334",
8   "address": "225-245 Plenty Rd, Bundoora",
9   "start_date": "2020-00-29",
10  "end_date": "2021-00-30",
11  "created_At": "2019-00-31",
12  "updated_At": null,
13  "role": "c"
14 }
```

- Retrieve user information if credentials are valid

# Retrospective

## ■ What worked well?

- Implementation of core functionalities
- Sprints based development process
- Independent coding task

## ■ What could be improved?

- More involvement in CI/CD
- Unit testing
- Higher code quality
- Implementation discussion

## ■ What “surprised” you?

- Some members are inactive/missing
- The amount to learn

## ■ What did you learn?

- New platforms and tools
- Spring Boot annotations
- Scrum
- AWS