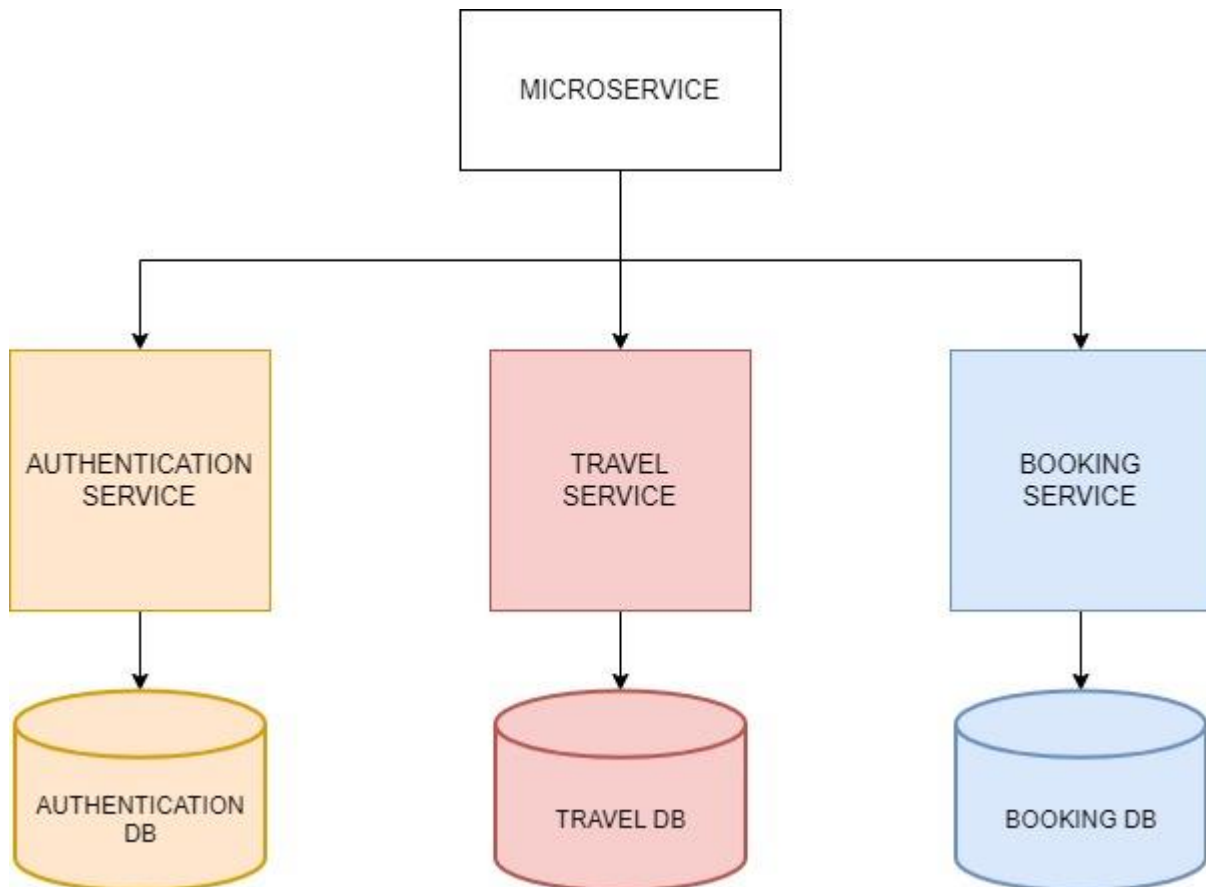
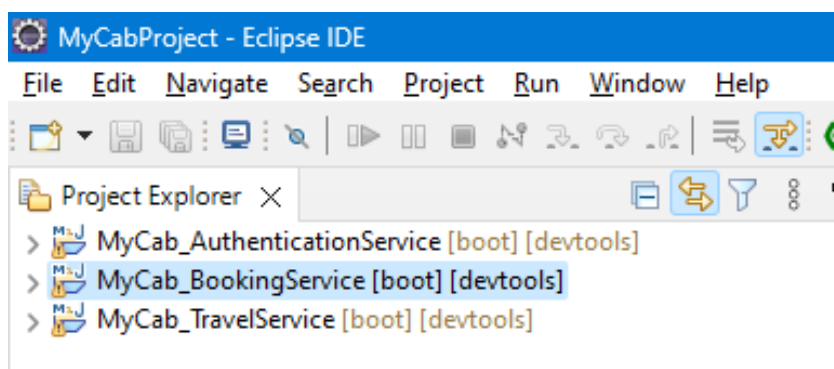


Project : Cab Buddy



Created 3 Services




Created 2 Interface

Admin Panel

- a. Admin can login and include all the information about trave, cab, and Driver

a. Login Panel



Email Address

Password

Login

Don't have an account. Click [here](#) to Register

b. Dashboard – View all travel list

Logo

Travel

Cab

Driver

Welcome, arvend

Log Out

Add New Travel

| No | Travel Id | Travel Source | Travel Destination | Travel Time | Travel Date | Travel Cost | Driver | Cab | Action |
|----|-----------|---------------|--------------------|-------------|-------------|-------------|-----------|----------|-----------------------|
| 1 | 19 | ipoh | KL | 17:37:00 | 02/13/2024 | MYR100.00 | asdasf | test | <div>EditDelete</div> |
| 2 | 20 | Johor | Kedah | 06:09:00 | 02/13/2024 | MYR100.00 | asdasf | test | <div>EditDelete</div> |
| 3 | 21 | Pahang | Johor | 10:00:00 | 02/17/2024 | MYR150.00 | my driver | cab name | <div>EditDelete</div> |
| 4 | 22 | Johor | Kedah | 08:09:00 | 02/17/2024 | MYR200.00 | my driver | cab name | <div>EditDelete</div> |
| 5 | 23 | Johor | Kedah | 10:02:00 | 02/29/2024 | MYR210.00 | my driver | cab name | <div>EditDelete</div> |
| 6 | 24 | Johor | Kedah | 10:30:00 | 02/29/2024 | MYR320.00 | my driver | cab name | <div>EditDelete</div> |

c. Cab Module

a. Create new Cab

Logo

Travel

Cab

Driver

Ceate New Cab

Cab Model

BMW

Cab Name

Ranger

Cab Plate Num

ADK1111

Submit

Back to Cab

b. View all Cab

Add New Cab

| No | Cab Id | Cab Name | Cab Plate Num | Action | |
|----|--------|----------|---------------|--------|--------|
| 1 | test | test | test | Edit | Delete |
| 2 | cab1 | cab name | amh 3211 | Edit | Delete |
| 3 | BMW | Ranger | ADK1111 | Edit | Delete |

c. Edit Cab

Add New Cab

| No | Cab Id | Cab Name | Cab Plate Num | Action |
|----|--------|----------|---------------|-----------------------|
| 1 | test | test | test | <div>EditDelete</div> |
| 2 | cab1 | cab name | amh 3211 | <div>EditDelete</div> |
| 3 | BMW | Ranger | ADK1111 | <div>EditDelete</div> |

Edit Cab

Cab Model

BMW

Cab Name

Ranger

Cab Plate Num

ADK2222

Submit

Back to Cab

d. Delete Cab

Add New Cab

| No | Cab Id | Cab Name | Cab Plate Num | Action |
|----|--------|----------|---------------|-----------------------|
| 1 | test | test | test | <div>EditDelete</div> |
| 2 | cab1 | cab name | amh 3211 | <div>EditDelete</div> |
| 3 | BMW | Ranger | ADK1111 | <div>EditDelete</div> |

Driver Module

- Driver and Cab are 1 to 1 relation
- Cab ID is included as Foreign Key on Driver Table
- One Driver can have only Once Cab Assigned

- Create New Driver
 - Can choose Cab when create new driver

Create New Driver

Driver Name

arvend

Driver Contact

12343545

Driver MyKad

0101101

Select Cab For Driver

Ranger

Submit

Back to Driver

- View all Driver

Add New Driver

| No | Driver Id | Driver Name | Driver Contact | Driver MyKad | Cab Assigned | Action | |
|----|-----------|-------------|----------------|--------------|--------------|--------|--------|
| 1 | 2 | asdasf | asdfsdf | 123123 | test | Edit | Delete |
| 2 | 3 | my driver | 0125185151 | 961006085221 | cab name | Edit | Delete |
| 3 | 4 | arvend | 12343545 | 0101101 | Ranger | Edit | Delete |

c. Edit Driver

Add New Driver

| No | Driver Id | Driver Name | Driver Contact | Driver MyKad |
|----|-----------|-------------|----------------|--------------|
| 1 | 2 | asdasf | asdfadsf | 123123 |
| 2 | 3 | my driver | 0125185151 | 961006085221 |
| 3 | 4 | arvend | 12343545 | 0101101 |

Edit Driver

Driver Name

arvend

Driver MyKad

0101101

Driver Contact

12343545

Select Cab For Driver

Ranger

Submit

Back to Driver

Travel Module

- Driver and Cab are relation to Travel
- One Driver can be in Many Travel
- One Cab can be in many travel
- Driver ID and Cab ID are foriegn key on Travel Table

a. Create new travel

Ceate New Travel

Travel Source

Johor

Travel Destination

Perak

Travel Time

08:00 AM

Travel Date

24/02/2024

Travel Cost

150

Select Driver

arvend

Cab

Ranger

Submit

Back to Travel

b. View all travel

Add New Travel

| No | Travel Id | Travel Source | Travel Destination | Travel Time | Travel Date | Travel Cost | Driver | Cab | Action | |
|----|-----------|---------------|--------------------|-------------|-------------|-------------|-----------|----------|--------|--------|
| 1 | 19 | ipoh | KL | 17:37:00 | 02/13/2024 | MYR100.00 | asdasf | test | Edit | Delete |
| 2 | 20 | Johor | Kedah | 06:09:00 | 02/13/2024 | MYR100.00 | asdasf | test | Edit | Delete |
| 3 | 21 | Pahang | Johor | 10:00:00 | 02/17/2024 | MYR150.00 | my driver | cab name | Edit | Delete |
| 4 | 22 | Johor | Kedah | 08:09:00 | 02/17/2024 | MYR200.00 | my driver | cab name | Edit | Delete |
| 5 | 23 | Johor | Kedah | 10:02:00 | 02/29/2024 | MYR210.00 | my driver | cab name | Edit | Delete |
| 6 | 24 | Johor | Kedah | 10:30:00 | 02/29/2024 | MYR320.00 | my driver | cab name | Edit | Delete |
| 7 | 25 | Johor | Perak | 08:00:00 | 02/24/2024 | MYR150.00 | arvend | Ranger | Edit | Delete |

c. Edit Travel

Edit Travel

Travel Source

Johor

Travel Destination

Perak

Travel Time

08:00 AM

Travel Date

24/02/2024

Travel Cost

150

Select Driver

arvend

Select Cab

Ranger

Submit

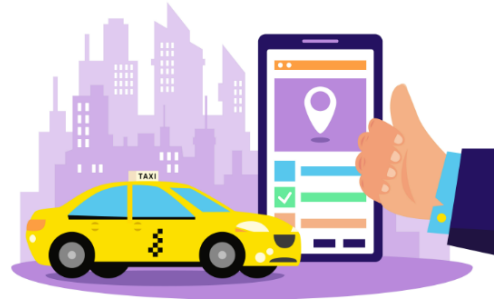
Back to Travel

User Panel

- Home Page

Welcome to MyCab

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam laboriosam dolorem rem aut ipsam nihil eius sequi repellendus dicta itaque ex, nam aliquid excepturi voluptatem commodi qui odio quaerat ut. Sint, veniam soluta inventore dolor reiciendis veritatis quae molestias similique, velit, officiis itaque incidunt cum omnis aut dolores illum eaque!



Book Your Cab

Select Travel Source

Select Travel Destination

Check Availability

Cancel

- User can select source and destination and click submit to view all the travel

Book Your Cab

Select Travel Source

Select Travel Destination

Check Availability

Cancel

| | |
|---|---|
| <div>Johor=>Kedah</div> <div>Travel Time: 06:09:00</div> <div>Travel Date: 2024-02-13</div> <div>Driver: asdasf</div> <div>Cab: test</div> <div>100</div> <div>Book Now</div> | <div>Johor=>Kedah</div> <div>Travel Time: 08:09:00</div> <div>Travel Date: 2024-02-17</div> <div>Driver: my driver</div> <div>Cab: cab name</div> <div>200</div> <div>Book Now</div> |
| <div>Johor=>Kedah</div> <div>Travel Time: 10:02:00</div> <div>Travel Date: 2024-02-29</div> <div>Driver: my driver</div> <div>Cab: cab name</div> <div>210</div> <div>Book Now</div> | <div>Johor=>Kedah</div> <div>Travel Time: 10:30:00</div> <div>Travel Date: 2024-02-29</div> <div>Driver: my driver</div> <div>Cab: cab name</div> <div>320</div> <div>Book Now</div> |

- User click book now button to view booking confirmation
 - o User can select num of days to travel

Booking Form

| | |
|--------------|------------------|
| Your Source | Your Destination |
| Johor | Kedah |
| Travel Date | Departure Time |
| 2024-02-29 | 10:30:00 |
| Your Cab | |
| cab name | |
| Driver Name | |
| my driver | |
| Per Day Cost | |
| 320 | |
| Select Days | |
| 3 | |
| Submit | Cancel |

- Once user click submit, request send to BACKEND and will perform Business Logic and send back response to client
 - o Total Price is calculated
 - o $\text{Total Price} = \text{Total Days} * \text{Per Day Cost}$

Payment Form

| | |
|--------------|------------------|
| Your Source | Your Destination |
| Johor | Kedah |
| Travel Date | Departure Time |
| 2024-02-29 | 10:30:00 |
| Per Day Cost | |
| 320 | |
| Total Days | Total Payment |
| 3 | 960 |
| Make Payment | Cancel |

Database

1. Authentication DB
2. Travel DB
3. Booking DB

Booking DB

[illegible]

Travel DB

- Cab

[illegible]

- Driver

[illegible]

- Travel

[illegible]

Authentication DB

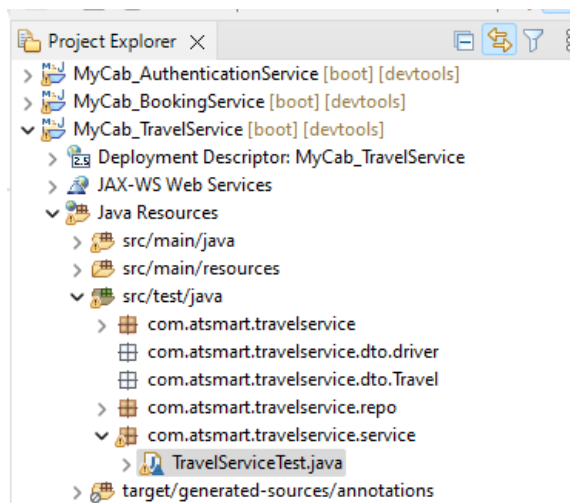
[illegible]

Microservice

- Send request from Booking Service to Travel Service to get travel details by ID

```
BookingService.java X
26 public BookRegisterDto calculateCost(BookOrderDto bookOrderdto) {
27
28     try {
29
30         BookRegisterDto bookRegisterDto = new BookRegisterDto();
31         int travelId = bookOrderdto.getTravelId().intValue();
32
33         RestTemplate temp = new RestTemplate();
34
35         String url = "http://localhost:8082/api/v1/travel/"+travelId;
36
37         ResponseEntity<ResponseDto> responseEntity = temp.getForEntity(url, ResponseDto.class);
38
39         ResponseDto responseBody = responseEntity.getBody();
40
41         // Now you can access the travel details like this:
42         TravelDetailDto travelDetail = responseBody.getData();
43
44         if(travelDetail != null) {
45
46             int perDayPrice = Integer.parseInt(travelDetail.getTravelCost());
47             int totalDays = Integer.parseInt(bookOrderdto.getDays());
48             int calculateTotalCost = perDayPrice * totalDays;
49
50             bookRegisterDto.setTravelDestination(travelDetail.getTravelDestination());
51             bookRegisterDto.setTravelSource(travelDetail.getTravelSource());
52             bookRegisterDto.setTravel_time(travelDetail.getTravel_time());
53             bookRegisterDto.setTravelDate(travelDetail.getTravelDate());
54             bookRegisterDto.setTravelDays(bookOrderdto.getDays());
55             bookRegisterDto.setTravelPerDayCost(travelDetail.getTravelCost());
56             bookRegisterDto.setTravelTotalCost(calculateTotalCost);
57             bookRegisterDto.setStatus(0);
58
59             BookingTravel book = mapper.BookingRegisterDtoToEntity(bookRegisterDto);
60
61             BookingTravel savedBook = repo.save(book);
62
63             BookRegisterDto savedRegister = mapper.EntityToBookRegisterDto(savedBook);
64
65             return savedRegister;
66
67         }else {
68             throw new BookingException("Travel not found");
69         }
70
71     }catch(Exception ex) {
```

Junit



```
TravelServiceTest.java X
1 package com.atsmart.travelservice.service;
2
3 import static org.assertj.core.api.Assertions.assertThat;
35
36 class TravelServiceTest {
37
38     @Mock
39     TravelRepo repo;
40
41     @Mock
42     DriverRepo driverRepo;
43
44     @Mock
45     CabRepo cabRepo;
46
47     @Mock
48     TravelMapper mapper;
49
50     @InjectMocks
51     private TravelService travelService; // Ask Mockito to inject
52
53     AutoCloseable autoCloseable;
54
55     TravelRegisterDto dto;
56     Driver driver;
57     Cab cab;
58     Travel travel;
59     CabDto cabDto;
60     DriverResponseDto driverResponseDto;
61     TravelUpdateDto updateDto;
62     TravelResponseDto expectedResponse;
```

```

@BeforeEach
void setUp() {
    autoCloseable = MockitoAnnotations.openMocks(this);

    // Given
    dto = new TravelRegisterDto("Source", "Destination", "10:30", "100", new Date(System.currentTimeMillis()), 1, 1);

    cab = new Cab(1, "Cab Model", "Cab Name", "Cab PlateNum", null, null, 1, null);
    driver = new Driver(1, "Driver Name", "Driver Contact", "Driver IdentificationNum", null, null, 1, cab, null);

    travel = new Travel(1, "Source", "Destination", LocalTime.of(10, 30), dto.getTravelDate(), "100", null, null, 1, driver, cab);

    cabDto = new CabDto(1, "Cab Model", "Cab Name", "Cab PlateNum");
    driverResponseDto = new DriverResponseDto(1, "Driver Name", "Driver Contact", "Driver IdentificationNum", cabDto);
    updateDto = new TravelUpdateDto(1, "New Source", "New Destination", LocalTime.now(), new Date(System.currentTimeMillis()), "200", 1, 1);
    expectedResponse = new TravelResponseDto(1, "Source", "Destination", "10:30", dto.getTravelDate(), "100", driverResponseDto, cabDto);
}

}

86 @AfterEach
87 void tearDown() throws Exception {
88     autoCloseable.close();
89 }
90
91 @Test
92 void testInsertTravel() {
93
94     mock(TravelRegisterDto.class);
95
96     int driverId = dto.getDriverId().intValue();
97     when(driverRepo.findById(driverId)).thenReturn(Optional.of(driver));
98
99     int cabId = dto.getCabId().intValue();
100    when(cabRepo.findById(cabId)).thenReturn(Optional.of(cab));
101
102    when(mapper.travelRegisterDtoToEntity(dto)).thenReturn(travel);
103
104    when(repo.save(any(Travel.class))).thenReturn(travel);
105
106    when(mapper.entityToTravelResponseDto(travel)).thenReturn(expectedResponse);
107
108    assertThat(travelService.insertTravel(dto)).isEqualTo(expectedResponse);
109 }
110
111 @Test
112 void testGetAllTravel() {
113
114     when(repo.findAll()).thenReturn(List.of(travel));
115     when(mapper.entitiesToTravelRegisterDtos(List.of(travel))).thenReturn(List.of(expectedResponse));
116
117     assertThat(travelService.getAllTravel()).isEqualTo(List.of(expectedResponse));
118
119 }
120

```



```

121 @Test
122 void testGetTravelById() {
123
124     when(repo.findById(1)).thenReturn(Optional.of(travel));
125     when(mapper.entityToTravelResponseDto(travel)).thenReturn(expectedResponse);
126
127     assertThat(travelService.getTravelById(1)).isEqualTo(expectedResponse);
128 }
129
130 @Test
131 void testUpdateTravel() {
132
133     when(repo.findById(1)).thenReturn(Optional.of(travel));
134     when(driverRepo.findById(updateDto.getDriverId().intValue())).thenReturn(Optional.of(driver));
135     when(cabRepo.findById(updateDto.getDriverId().intValue())).thenReturn(Optional.of(cab));
136
137     assertThat(travelService.updateTravel(updateDto, 1)).isTrue();
138     verify(repo).save(travel);
139
140 }
141
142 @Test
143 void testDeleteTravel() {
144     fail("Not yet implemented");
145 }
146
147 @Test
148 void testCheckAvailable() {
149
150     when(repo.findBySourceAndDestination("Source", "Destination")).thenReturn(List.of(travel));
151     when(mapper.entitiesToTravelRegisterDtos(List.of(travel))).thenReturn(List.of(expectedResponse));
152
153     assertThat(travelService.checkAvailable("Source", "Destination")).isEqualTo(List.of(expectedResponse));
154 }
155 }
156 }
157

```

