

DeliverUS Exam - Model A - Orders

Please note that DeliverUS is described in: <https://github.com/IISSE2-IS-2025>

Exam Statement

You are required to implement the graphical interface for some functional requirements of the owners, specifically:

RF.01 – Viewing Orders by Restaurant

As

an owner,

I want

to view a list of all orders made to one of my restaurants. For each order, I need to know its creation date, status, delivery address, and price,

So that

I can track and manage them.

RF.02 – Editing Some Properties of Orders

As

an owner,

I want

to be able to edit the delivery address and price of orders made to my restaurants,

So that

I can correct errors or update relevant information.

Acceptance Tests

- The owner can access order editing from the restaurant's order list.
- Only the delivery address and total price fields are shown and must be presented with the current values of the order to be edited.
- Both fields (delivery address and total price) are mandatory.
- If an order is saved without filling in the delivery address, an error message is displayed indicating that the field is required.
- If an order is saved without setting the total price, an error message is displayed indicating that the field is required.
- If a price of 0 or lower is entered, an error message is displayed indicating that the value must be greater than 0.
- When changes are successfully saved, the order information is updated, and a confirmation message is displayed.

- After the data is successfully updated, the system should automatically navigate to the order list showing the updated information.

RF.03 – Viewing Restaurant Order Analytics

As

an owner,

I want

to view analytics about the orders of each of my restaurants. For each restaurant, I want to know: the amount invoiced today, the number of orders in **pending** status, the number of orders today in **delivered** status, and the number of orders placed yesterday.

So that

I can make informed decisions about the management of my restaurant.

RF.04 – Changing an Order's Status

As

an owner,

I want

to be able to change the current status of orders placed at my restaurants. From the **pending** status, I can change it to **in process**; from **in process** to **sent**; and from **sent** to **delivered**,

So that

I can keep the status of my orders up-to-date and inform the customer accordingly.

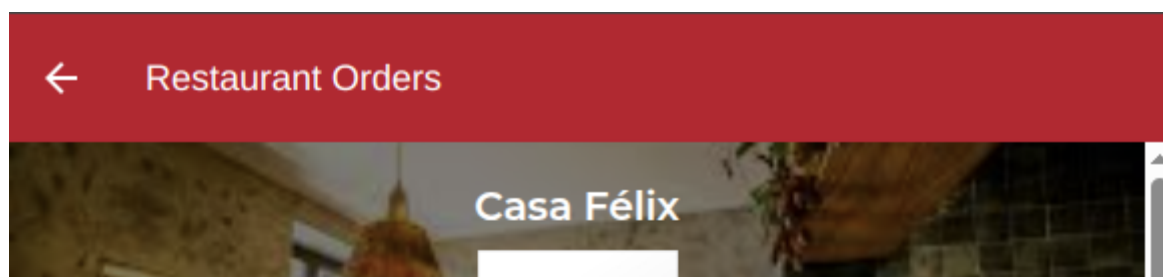
Acceptance Tests


- When the status of an order is successfully changed, it must be reflected in the list and the analytics section must be updated. Additionally, a success message should be displayed.
- If the status change fails, an error message must be displayed.
- Orders in the **delivered** status cannot have their status changed, nor should the option to change it be presented.

Exercises

1. Order List. 3 points

Work on RF.01 in the file `./DeliverUS-Frontend-Owner/src/screens/orders/OrdersScreen.js`, making all necessary changes to display an interface as shown in Figure 1.






Cocina Tradicional
Spanish

Invoiced today
65.00€

#Pending orders
1

#Delivered today
1

#Yesterday orders
2




PENDING


Order created at 2025-04-21T08:23:24.000Z


Status: pending

Address: Pending order address

15.00€

 Edit

 Advance




IN PROCESS


Order created at 2025-04-20T08:23:24.000Z


Status: in process

Address: Yesterday order 1

17.00€

 Edit

 Advance




IN PROCESS


Order created at 2025-04-21T08:23:24.000Z


Status: in process

Address: In process order address

18.00€

 Edit

 Advance




SENT


Order created at 2025-04-21T08:23:24.000Z

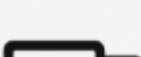
Status: sent

Address: Sent order address

12.50€

 Edit

 Advance



Order created at 2025-04-20T08:23:24.000Z

Status: delivered

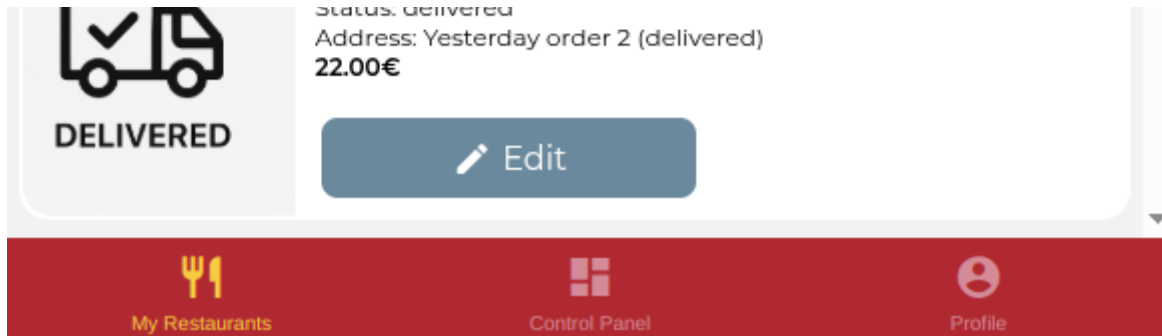


Figure 1: Order list and analytics

Key aspects to consider:

- The order list screen is accessible from the **Check orders** button, which has been added to the restaurant detail screen **RestaurantDetailScreen**.
- The image representing the order's status can be obtained using the function **getOrderImage(status)**, which is provided in the same file.
- The order list can be obtained using the function **getRestaurantOrders(restaurantId)**, which is already provided in **./DeliverUS-Frontend-Owner/src/api/RestaurantEndpoints.js** and makes the necessary request to the backend.
- The order edit button will be worked on in Exercise 2.
- The analytics section will be handled in Exercise 3.
- The status change button will be handled in Exercise 4.

2. Order Editing. 3 points

Work on RF.02 in the file **./DeliverUS-Frontend-Owner/src/screens/orders/EditOrderScreen.js** as shown in Figure 2.

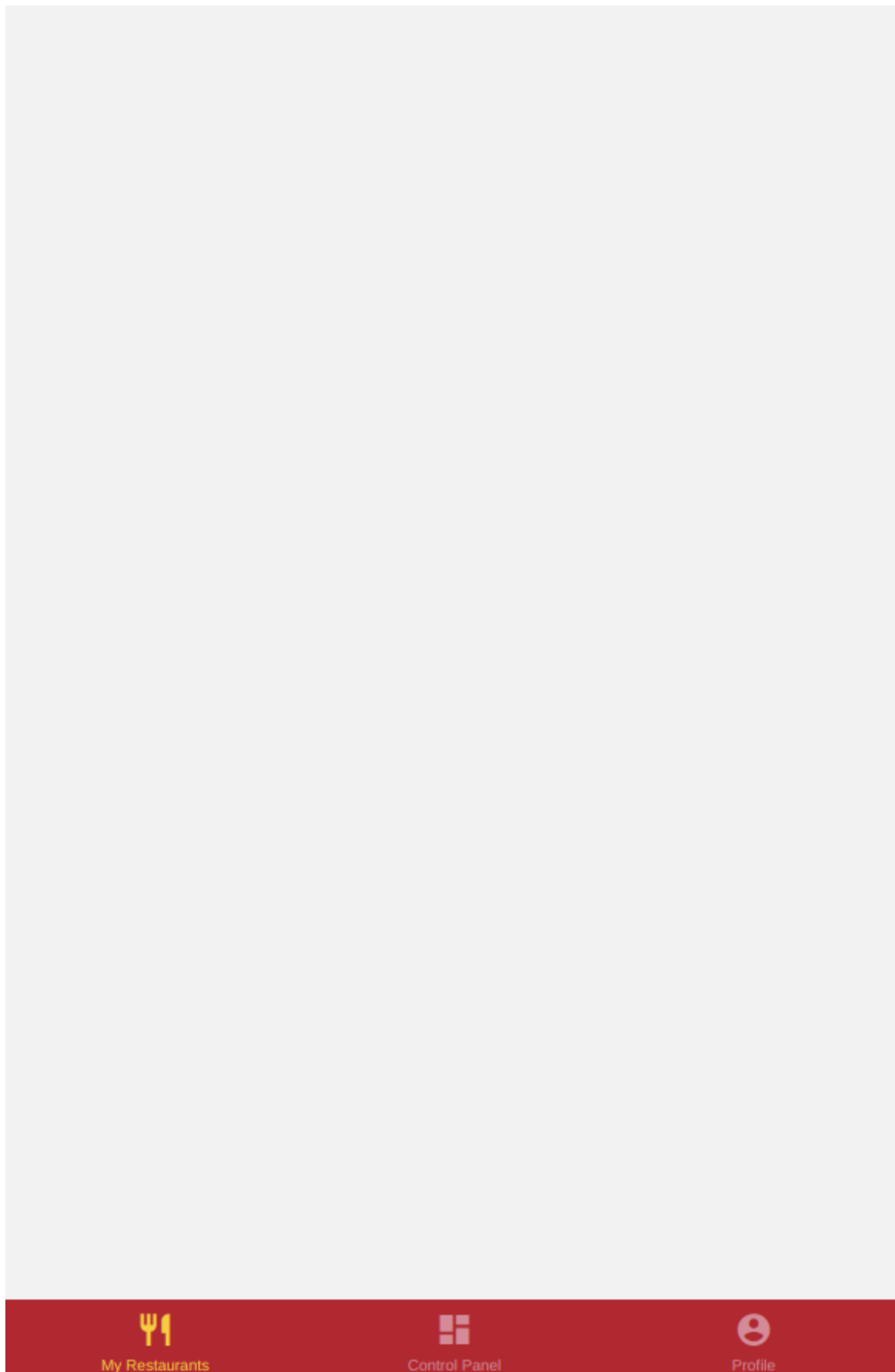


Figure 2: Order editing

Key aspects to consider:

- You must add a button to navigate to the order editing screen from the order list in the file `./DeliverUS-Frontend-Owner/src/screens/orders/OrdersScreen.js`.

- You must implement the necessary elements to enable navigation to the order editing screen.

3. Restaurant Analytics. 2 points

Work on RF.03 in the file `./DeliverUS-Frontend-Owner/src/screens/orders/OrdersScreen.js`, making all necessary changes to display the analytics section as shown in Figure 1.

Key aspects to consider:

- The styles `analyticsContainer`, `analyticsRow`, and `analyticsCell` are provided to display the analytics section. You must complete the properties related to Flex to adjust it accurately to the style presented in Figure 1.
- The analytics object returned by the backend has the following structure (example values):

```
{
  restaurantId: 1,
  numYesterdayOrders: 2,
  numPendingOrders: 1,
  numDeliveredTodayOrders: 1,
  invoicedToday: 65.0
}
```

4. Status Change. 2 points

Work on RF.04 in the file `./DeliverUS-Frontend-Owner/src/screens/orders/OrdersScreen.js`, making all necessary changes to allow advancing the status of each order as shown in Figure 1.

Key aspects to consider:

- You are provided with the function `nextStatus(order)` in the file `./DeliverUS-Frontend-Owner/src/api/OrderEndpoints.js`, which makes the request to the backend to advance the status.
- The `MaterialCommunityIcon` name for the `Advance` button is `skip-next`.

Submission Procedure

1. Delete the **DeliverUS-Backend/node_modules**, **DeliverUS-Frontend-Owner/node_modules**, and **DeliverUS-Frontend-Owner/.expo** folders.
2. Create a ZIP file that includes the entire project. **Important: Ensure the ZIP is not the same as the one you downloaded, and includes your solution.**
3. Notify the professor before submission.
4. When the professor approves, upload the ZIP to the Virtual Learning Platform. **It is crucial to wait until the platform shows a link to the ZIP before clicking the submit button.** It is recommended to download the ZIP to check what has been uploaded. Once verified, you may submit the exam.

Environment Setup

a) Windows

- Open a terminal and run the following command:

```
npm run install:all:win
```

b) Linux/macOS

- Open a terminal and run the following command:

```
npm run install:all:linux
```

Execution

Backend

- To **recreate the migrations and seeders**, open a terminal and run the following command:

```
npm run migrate:backend
```

- To **start the backend**, open a terminal and run the following command:

```
npm run start:backend
```

Frontend

- To **run the frontend application for the owner**, open a new terminal and run the following command:

```
npm run start:frontend
```

Debugging

- To **debug the frontend**, ensure that an instance of the frontend you wish to debug is running, and use the browser's debugging tools.