Epics

Login customers

1. customers can register.
2. customers can login.
3. customers can reset their password.

Customers

1. A list of products must be shown to the customers when choosing what they want to eat.
2. The option to tip the waiter/waitress must be given when you've chosen to pay after your whole diner.

Login admin

1. Admin can register Admin accounts.
2. Admin can login.
3. Admin can password reset.

Login employees

1. Employees can register.
2. Employees can login.
3. Employees can ask for a password reset.

Employees

1. Employees (when logged into an administrator account) must be able to add items to, edit or remove from the menu.
2. An employee can see a (schematic) map with the tables in the restaurant.
    a. On this map the locations and routes of the robots are displayed.
3. It needs to be easy for the user to switch between map and order overview mode.
    a. Both views need all the possibilities for checking (and changing) orders.
4. An employee must be able to summon the robot(s) when food needs to be delivered to a table.
5. Employees need to be able to see if a customer wants service (at a specific table).

Robot properties en stop

1. Robot must show the following properties:
    a. Battery percentage
    b. Location in the restaurant
    c. Robot status (busy, available, charging, etc.)
    d. Should be able to show multiple robots.
2. Robot must have an emergency stop button.

Orders

1. Guests can add food and drinks to their order.
2. Guests can view their order.
3. Guests can place their order.
4. There must be an easy overview of all the orders that are placed.
    a. Oldest and uncompleted first (can be changed with filters)
5. Payment status for every order (paid or unpaid)

6. Orders can be split up into multiple part-orders when the amount of food does not fit onto the in one go.
7. Create an order, dynamically in some scenarios?
    a. When every course is chosen upfront, use a button to advance to the next meal?
    b. When choosing courses on the fly, use a button to advance to your next meal choice?

QR-code

1. Admins can generate QR-codes for tables.
2. QR-code on the table will register the table as active.
3. Set a cookie with a JWT (JSON web token) upon scanning the QR-code. Set the life cycle to a few hours to prevent them from ordering anything after they visited the restaurant.

Payment methods

1. Pay the whole bill before you eat (more like a store basket, choose everything you want to eat and immediately pay after you have made your choices)
2. Pay after you eat (Traditional way of visiting a restaurant)
3. Pay with your (credit) card.
4. Pay with cash (Contactless, let the robot pick it up?)

Menu

1. Guests can view the menu with all available items.
2. Admins can view the menu.
3. Admins can change the items in a menu.
4. Admins can add items to a menu.
5. Admins can delete items from the menu.

House style change

1. House style (Logo, colors, and information of the restaurant) can be configured at any time.

Export

1. Needs to be able to export all orders and transactions from any period.

Requirements Horeca-Robot

Employee app:

- Employees (when logged into an administrator account) must be able to add items to, edit or remove from the menu.
- An employee can see a (schematic) map with the tables in the restaurant.
    - On this map the locations and routes of the robots are displayed.
- An employee must be able to summon the robot(s) when food needs to be delivered to a table.
- The robot must have an emergency stop
- The app must show certain information about the robot:
    - Battery percentage.
    - Location in the restaurant.
    - Robot Status (Busy, available, charging, etc.)
    - Should be able to show multiple robots
- There are separate roles for accounts, regular employee, or administrator.
- There must be an easy overview of all the orders that are placed.
    - Oldest and uncompleted first (can be changed with filters)
- Payment status for every order (paid or unpaid)
- Employees need to be able to see if a customer wants service (at a specific table).
- Needs to be able to export all orders and transactions from any period.
- It needs to be easy for the user to switch between map and order overview mode.
    - Both views need all the possibilities for checking (and changing) orders.
- Orders can be split up into multiple part-orders when the amount of food does not fit onto the in one go.
- House style (Logo, colors, and information of the restaurant) can be configured at any time.

Customer app:

- Using a QR-code to be able to start eating
    - QR-code on the table will register the table as active.
    - Set a cookie with a JWT (JSON web token) upon scanning the QR-code. Set the life cycle to a few hours to prevent them from ordering anything after they visited the restaurant.
- Make a choice between different payment options
    - Pay the whole bill before you eat (more like a store basket, choose everything you want to eat and immediately pay after you have made your choices)
    - Pay after you eat (Traditional way of visiting a restaurant)
    - Pay with your (credit) card.
    - Pay with cash (Contactless, let the robot pick it up?)
- A list of products must be shown to the customers when choosing what they want to eat.
- The option to tip the waiter/waitress must be given when you've chosen to pay after your whole diner.
- Create an order, dynamically in some scenarios?
    - When every course is chosen upfront, use a button to advance to the next meal?
    - When choosing courses on the fly, use a button to advance to your next meal choice?