What is my app and who is it for?

I'm creating a web domain for a small home-based cake production. It will be a webpage with customer zone for registered users. The goal is for present the product to people, spread awareness about the product and enable them to reserve the product of their wishes for a particular date and time, since that is the only way the business can satisfy the potential demand in current conditions.

Obviously the more people the app attracts the better. However, realistically the target group will be ladies, mostly mothers, who would want our product for their family as a birthday cake, a desert for a festive lunch or dinner, or just as an average day sweet to have with coffee.

How will it work?

As mentioned, the app will have two purposes. Firstly, it will serve as the face of the business, where it will provide information to all **visitors** of the webpage. This will include a little intro about the business, description of our product(s), and of all available variants such as fillings, icings, and a lactose free version. To be transparent, prices will also be available up front with the product(s), together with the minimum amount of days the product(s) need to be prepared.

Secondly, there will be a **customer zone** available upon registration. In there, users will have the option to:

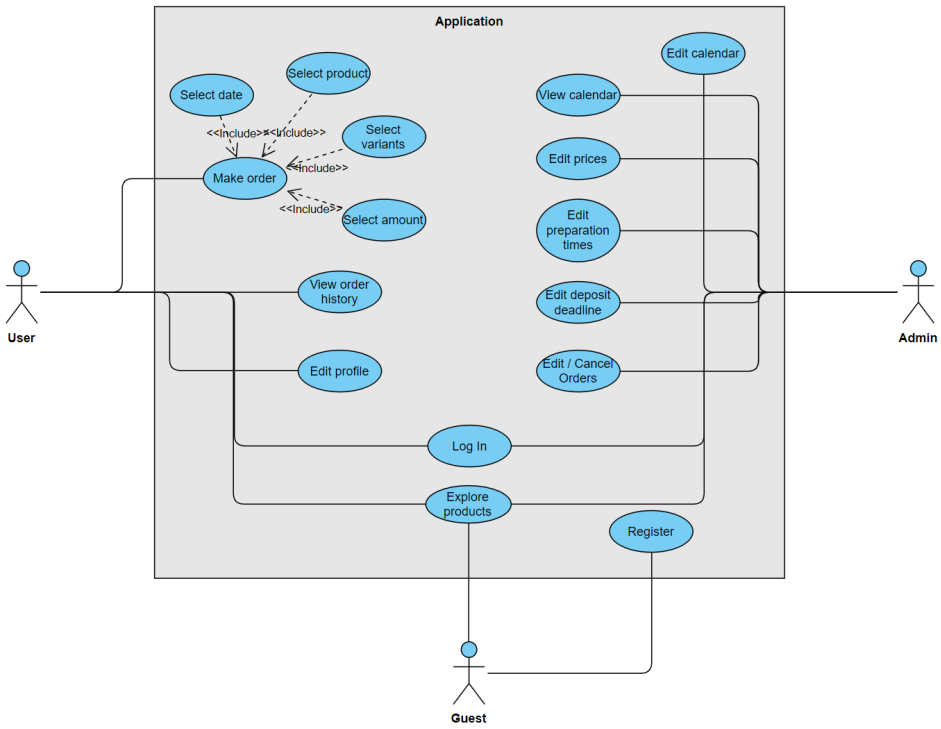
- `Make an order`
    - In this section, there will be a calendar with available dates. The customer will choose the type and amount of product(s) of their wishes and based on that, available dates will be calculated and displayed in the calendar. A color scheme to distinct unavailable and available dates will be used.
    - After choosing the date, either the user will be redirected to a different page or a pop-up window will appear, asking the customer to provide the delivery address and other required information before placing the order. After executing the order, payment info will be displayed prompting the customer to pay the down payment, or the whole amount.
- `Check their active and past orders`
    - Here users will see all their orders. Each of them will have status: active (pending, processed), finished.
    - Attached to pending orders will be the payment info should the user after placing the order decide to leave the down payment for later. However, if the down payment will not be realized enough in advance (this duration will be something like `argmin[TBD, preparation time]`), the order will be canceled - this might have to be checked manually for now.
- `Edit their account information`
    - The users will be able to define delivery address that will be automatically filled to the order placing form.
    - Also they will be able to edit their username, name, email, password etc.

Mobile version

The functionality of the mobile version of the app will be the same as the desktop version, however the design will be more minimalistic and tailored for the smaller screen. There will probably not be enough time during the semester to implement this, so we will leave it as an extra option, but most likely it will be done somewhen after.

User roles

- `Guest`
  - Only has access to the general website, where they can see the information about the business as well as about the product(s)
  - Can register as a user
- `User`
  - Can log in
  - Can explore the general page
  - Has access to customer zone
  - Can create orders
  - Can see order history
  - Can modify profile info
- `Admin`
  - Can log in
  - Can explore the general page
  - Can modify and cancel orders
  - Can modify the down payment delay individually for each order
  - Can modify preparation times for each variant and prices for each variant
  - Can view calendar with all due orders for each day
  - Can modify calendar, i.e. set available workload for each day, which will also automatically affect calculations for other days



Data model

`Users`

user id	role id	name	second name	email	phone	password hashed	street	house number	city	post code

`Roles`

role id	role name
	user / admin

`Orders metadata`

order id	user id	status	order time	order deadline	preparation time	deposit deadline	price	paid
		pending / processed / finished / modified						

`Orders details`

- For every product in each order, there will be an entry in this table with applied variants

detail id	order id	product id	variant ids	price	preparation time
			list of variant ids		

`Products`

product id	name	base price	preparation time

`Variants`

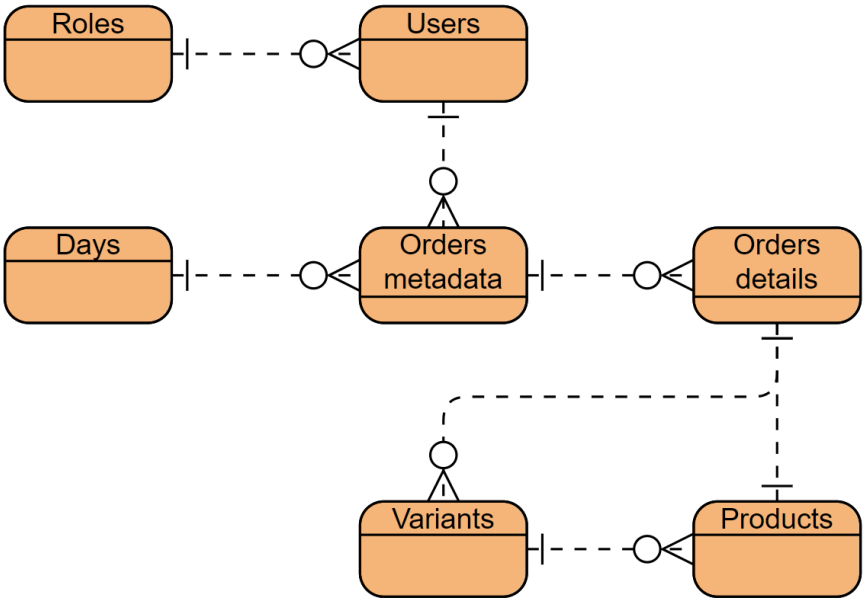
- Each product has it's own column determining the price when the variant is applied to that product (-1 if the variant is not applicable)

variant id	name	preparation time	product id	product id	...
		in case this variant adds time to default version	price when applied to product with this id	price when applied to product with this id	

`Days`

- For each day in the calendar there will be a set amount of cakes that can be produced and when a new order will be placed, it will be added to the "queue" of all concerned days which will be calculated based on deadline and preparation time.
- It would be a good idea for each cake to have some kind of a strategy so they don't add equal workload to all days in the preparation time period, since not all tasks require equal work. For example, after the cake is done it needs one day to "sit" to get the right taste. In the meantime another cake can be baked.

date	total workload	available workload	orders ("queue")	note
	set by admin, the maximum amount of cakes that can be produced	= total workload - sum(orders that need to be processed on this day)	list of orders that need to be processed on this day	



Software requirements

- `Client`
  - JS, HTML, CSS
  - React, jQuery
  - Next.js
- `Server`
  - JS, Typescript
  - Node.js, Express.js
  - Postgres
- `Communication`
  - Rest API
- `Hosting`
  - GoDaddy
- `Supported browsers`
  - Chrome
  - Firefox

Timeline

Week	Goal
5	frontend - guest pages and login/register page
6	frontend - order system, user zone; database setup
7	solve creating orders, changing profile info, bind with databases
8	admin interface
9	deploy to hosting service
10 - 12	fixes

Future goals

- Mobile version
- Compatibility with Safari
- Live rendering while choosing variants