Easy meal order application

Goal

The goal is to create ASP.NET Core MVC 3.1 web applications with a good application architecture using unit tests and at least a CI-build in Azure DevOps.

Case

Easy Meal is a meal ordering service where you can order your meals for the entire week. Our motto is: “Your freedom of choice is our first priority”.

Actors

There are 2 actors in the system: cooks and customers.

The following information about the actors is available in the system:

Cook:

- Name
- Email address
- Phone number

Customer:

- Customer number
- Address information
- Diet restrictions
- Birthdate

User stories

UC 1: As a cook I want to plan the meals, so that the customers can order a variety of meals every week.

A meal consists of three parts: appetizer, main dish and dessert. A different meal is available for every day of the week. Each day should have at least a different main dish. There is no rule that the appetizer and dessert should be different every day. Our customers have diet restrictions. We must take the following restrictions into account:

- Salt free
- Diabetes
- Gluten free

Every customer with a diet restriction, should be able to order a complete meal every day. The cook must take this into account when planning the meals for the week. The system must check this requirement.

The cook must complete the planning for week n before Sunday 23:59 in week n – 1. The cook must mark the planning complete before this time. If the planning for that week is not complete or valid, then the cook cannot complete the planning for the week. After the cook has marked the planning complete, the planning cannot be changed any more.

Every dish has a name, a price, diet restriction information, a short description and a picture.

The system must allow existing dishes to be added to a menu plan.

UC 2: As a customer I want to use my email address as a username to access the system so that I don't have to remember my username

A customer must be able to login to the system using his email address. The information to administer for a user is included in the actor section. A user must be able to change his information, except for the email address.

UC 3: As a customer I want to be able to order my meals for the coming week (n + 1), taking my diet restrictions into account, so that I have a good meal every day

A complete meal consists of the 3 courses. A customer must be able to order a meal consisting of at least two courses, so the appetizer or the desert can be left out. The customer must order for the entire week, so a single order for just one day is not allowed. For the system, the week starts on Monday and ends on Sunday. Skipping a single day in the week is allowed. Every week order should have at least 4 meals for the working days. The weekend is optional. A meal can be ordered with the sizing option: small (price – 20%), regular (price as specified) and large (price + 20%).

At the top of the screen, the system must show the summary information for the order being made. So, kind of a shopping card. The customer must complete the order on Thursday before 23:59 in week n. The allows the cook to order the ingredients for the next week.

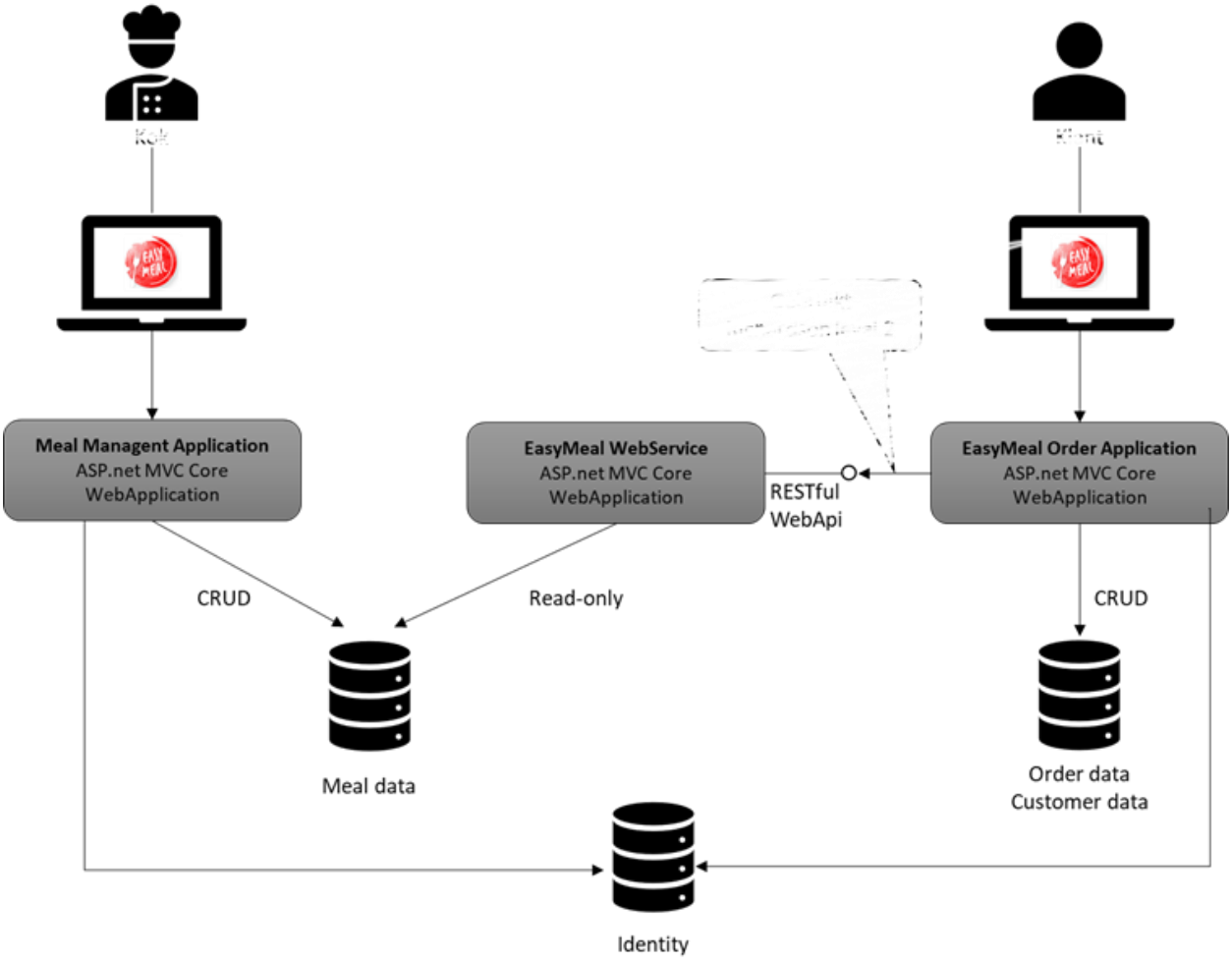
UC 4: As a customer I want to generate an invoice per month, so that I can keep track of my spending for the meals.

When ordering the meals for the week, the customer gets an indication of the costs. Per month discounts can apply. The customer receives a free meal on his birthday. If the customer orders more than 15 meals per month, then a 10% discount is applied. The free meal for the birthday is excluded from the 15 meals boundary. The discounts must be visible on the invoice. The invoice does not have to be generated into a pdf-document.

The following periods are important for the system.

Planning week	Order week	Delivery week
N - 1	N	N + 1

Architecture



There are three applications in this assignment:

- The meal management application for the cook.
- The meal information must be made available for other systems by means of an ASP.NET Core WebAPI service.
- The customer must be able to order the meals using the order application. The information about the meals must be retrieved from the WebAPI. The meal information may not be duplicated in the order application.