

ПРОЕКТ ПО ПРЕДМЕТОТ ЕЛЕКТРОНСКА И МОБИЛНА ТРГОВИЈА

Веб апликација за онлјан нарачка на храна

Изворен код достапен [тука](#)

Изработено од:

Симона Стојановска 181036

Октомври, 2021

Содржина

Вовед	3
Сценарио кое се обработува	3
ЕА дијаграм	4
Атрибути на ентитети	5
Ограничени контексти	6
Агрегати	6
Aggregate root	6
Бизнис правила	7
Вредносни објекти	7
Настани	7

Вовед

„My Restaurant“ е веб апликација која овозможува преглед и нарачка на јадења од мени на некој ресторан. Апликацијата овозможува регистрација и најава на корисници. Корисниците може да го прегледуваат менито. Освен што ги прегледува јадењата корисникот може да одбере некое јадење и истото да го додаде во својата нарачка. Откако ќе заврши се селектирањето јадења кои сака да ги порача, корисникот плаќа за својата нарачка. Веб апликацијата овозможува преглед на сите нарачки кои се подготвени за достава. Корисниците се во улога клиенти, customer, а еден корисник е во улога на администратор. Backend-от на апликацијата има микросервисна архитектура, составена од 4 микросервиси, Customer, Delivery, Menu и Order микросервисот.

Customer микросервисот работи на порта 8082 и овозможува најава и регистрација на корисници. За имплементација на автентикација и авторизација се користат JWT токени.

Delivery микросервисот работи на порта 8083 и овозможува креирање на достава од соодветна нарачка и преглед на сите доставки. Листата од сите доставки може да ја прегледаат корисници со улога на администратори.

Menu микросервисот работи на порта 8084. Тој овозможува преглед на сите достапни јадење, преглед на деталите за едно јадење, додавање на некое јадење. Додавањето на јадења е достапно само за администратори додека пак другите функционалности се достапни за сите корисници без разлика на тоа дали се авторизирани или не.

Order микросервисот работи на порта 8085. Секој авторизиран корисник може да креира нова нарачка при клик на копче „Create new order“. Откако ќе креира нова нарачка, корисникот може да додава ставки во нарачката. Секоја ставка е составена од јадење и количина. Корисникот може да ја зголеми или намали количината на јадења од ставката. Исто така може да избрише ставка од нарачката. Откако ќе заврши со додавање/бришење/изменување на ставки од нарачката, корисникот продолжува кон преглед на адресата за достава и плаќање на нарачката. Доколку сака во тој момент може и да ја откаже нарачката. Откако корисникот ќе заврши со плаќање на нарачката, се креира соодветна достава.

Редоследот на извршување на микросервисите е Customer, Menu, Order, Delivery

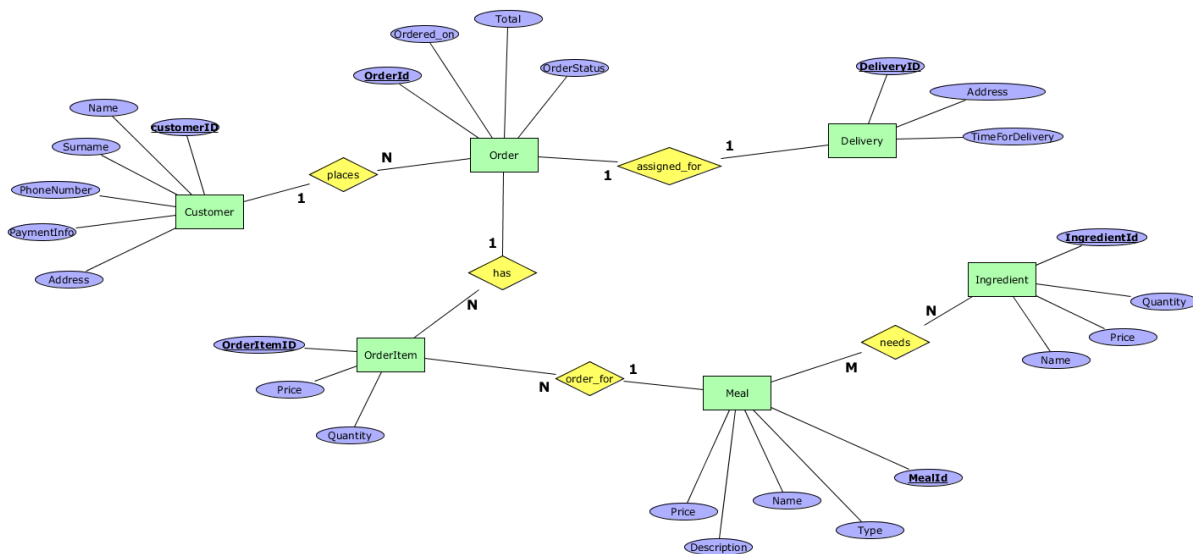
Сценарио кое се обработува

Сценариото кое ќе го обработувам е систем за менаџирање на некој ресторан. Конкретно ќе се осврнам на делот за креирање на нарачки и нивна достава. Корисниците ќе го користат системот online. Тие би можеле да го разгледуваат менито и да селектираат некое јадење од него. За секое јадење се чува информација за тоа од кои состојки е приготвено. Корисникот ќе може да креира нарачка само доколку е најавен на системот. Откако корисникот ќе се најави, во делот „My order“ му се овозможува креирање на нова нарачка. Во една нарачка може да се додадат повеќе јадења. Откако ќе заврши со нарачката, корисникот ја прегледува адресата на која сака да се достави нарачката и врши плаќање на истата. Информациите за нарачката и адресата за достава се предаваат на одделот за достава. Значи системот овозможува управување со ресторанот, конкретно со јадењата, со нарачките, со корисниците и со доставата на нарачките.

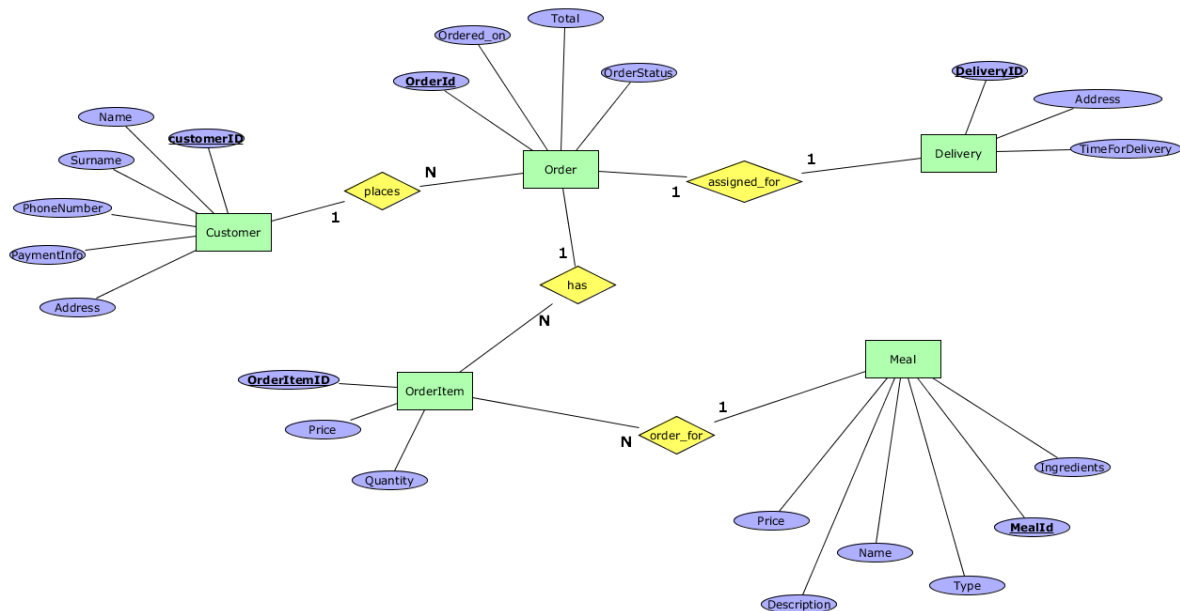
ЕА дијаграм

Направена е мала измена во ЕА дијаграмот во однос на оној дијаграм кој е предаден во Лабораториска вежба 03. Конкретно ентитетот Состојки,Ingredients е моделиран како вредносен објект, наместо како ентитет. При дефинирање на ограничените контексти, дефинирам еден ограничен контекст кој се однесува на самиот ресторан. Во тој ограничен контекст беа ентитетите Meal и Ingredient, а како Aggregate Root го одбрав Meal агрегатот. Со тоа сите операции поврзани со состојките ќе треба да поминуваат преку овој агрегат. Значи секогаш кога додаваме некој состојка, треба да дефинираме јадење од кое е дел, а потоа информации за самата состојка. За да го одбегнам тој проблем имам две можности, да дефинирам посебен ограничен контекст за состојките или да ги моделирам како вредносен објект за секој јадење. Бидејќи апликацијата се фокусира на нарачка на храна, се одлучив состојките да ги моделирам како вредносен објект.

На Слика1 е прикажан ЕА дијаграмот прикачен како дел од Лаб 03, а на Слика2 е прикажан корегираниот ЕА дијаграм.



Слика 1 ЕА дијаграм од Лаб 03



Слика 2 Коригиран EA дијаграм

Ентитетите кои ги имам дефинирано се: Customer, Order, Orderitem, Meal, Delivery. Како што кажува и самото име на ентитетот, Customer ги означува корисниците, Order е нарачката, Orderitem е ставка во нарачката, Meal го репрезентира јадењето, а Delivery ентитетот се однесува на доставата, односно на информациите поврзани со доставата.

Релациите помеѓу ентитетите се: Customer креира Order. Еден клиент може да креира N нарачки, но една нарачка му припаѓа само на еден клиент.

Секоја нарачка се состои од ставки (Orderitem). Една нарачка може да има N ставки, но една ставка припаѓа само на една нарачка.

Секоја ставка се состои од јадење (Meal). Едно јадење може да се јави во N ставки, но во една ставка има само едно јадење.

Секоја нарачка се назначува за достава (Delivery). Една нарачка може да се достави само еднаш. Една достава се однесува само на една нарачка (бидејќи во Delivery ентитетот се чува адреса на која треба да се достави нарачката и време на достава).

Атрибути на ентитети

Еден од ентитетите е Customer. Тој ги има атрибутите CustomerId (корисничко име, емаил адреса, кој се зема за примарен клуч). Name (име на корисникот, од тип String). Surname (презиме на корисникот, од тип String). PhoneNumber (телефонски на корисникот, од тип String). Paymentinfo (кредитна картичка на корисникот, од тип String), Address (адреса на живеење на корисникот, од тип String).

Друг ентитет е Order. Неговите атрибути се orderId (id на ентитетот од тип string, кој се зема за примарен клуч), Ordered on (датум на кој е креирана нарачката, од тип date). Total (вкупната сума на нарачката). Order_status (статус на нарачката, енумерација).

Друг ентитет е orderItem. Неговите атрибути се orderItemId (id на ентитетот од тип string, кој се зема за примарен клуч), Price (ја означува цената на ставката од тип double), Quantity ја означува количината на ставката, од тип int).

Следен ентитет е Meal. Тој ги има атрибутите MealId (id на ентитетот од тип string, кој се зема за примарен клуч). Name (име на јадењето од тип String), Description (опис на јадењето од тип String). Price (цена на јадењето од тип double). Type (тип на јадење, се чува како енумерација).

И на крај имаме ентитет Delivery. Тој ги има атрибутите DeliveryId (id на ентитетот од тип string, кој се зема за примарен клуч), Address (адресата на која треба да се достави нарачката од тип String), Timefor Delivery (време на достава на нарачката, од тип date).

Ограничени контексти

За системот за нарачка на храна во еден ресторан имаме неколку ограничени контексти. Првиот ограничен контекст се однесува на менаџирање со нарачки. Во овој контекст се ентитетите Order и OrderItem. Друг ограничен контекст се однесува на самиот ресторан. Во него имаме само еден ентитет, Meal. Следниот ограничен контекст се однесува на доставата. Во него го имаме само ентитетот Delivery. И последниот ограничен контекст се однесува на корисниците. Тука го имаме само ентитетот Customer.

Агрегати

Претходно дефинирав 4 ограничени контексти. Едниот се однесуваше на нарачки, другиот на ресторанот, третиот на корисниците и четвртиот на достава. Во ограничениот контекст на нарачки имаме 2 ентитета, Order и OrderItem. Тие два ентитета претставуваат агрегати. Во контекстот кој се однесува на ресторанот имаме го имаме ентитетот Meal кој претставува агрегат. Во ограничениот контекст за корисници имаме еден ентитет, Customer, кој претставува агрегат. Слично и во ограничениот контекст за достава имаме само еден ентитет Delivery, кој претставува агрегат.

Aggregate root

Претходно дефинирав 4 ограничени контекст, за нарачки, за ресторан, за корисници и за достава. Во ограничениот контекст на нарачки имаме 2 агрегати, Order и OrderItem. Како aggregate root се зема Order агрегатот и за целата комуникација со контекстот за управување со нарачки ќе се користи Order агрегатот. Во контекстот кој се однесува на ресторанот имаме само еден агрегат, Meal, кој се зема за aggregate root. Па така секогаш кога додаваме некое јадење во нарачката имаме комуникација помеѓу aggregate root на контекстот на нарачки и на ресторанот. Во ограничениот контекст за корисници имаме еден агрегат, Customer, кој претставува aggregate root. На крај во ограничениот контекст за достава имаме еден агрегат, Delivery кој се зема за aggregate root.

Бизнис правила

Customer ентитетот ќе има имплементација на регистрација и најава на корисници.

Meal ентитетот овозможува преглед на достапните јадења и додавање на ново јадење во менито. Исто така, корисникот селектира јадење и количина и тоа се додава како ставка во нарачката.

Order ентитетот ќе овозможи креирање на нова нарачка, додавање на ставки (OrderItems) во нарачката. Откако ќе заврши со нарачката, истата се затвара (OrderStatus=closed). По ова корисникот може да ја внесе адресата на која треба да се изврши доставата. Информациите за нарачката и адресата на достава се користат за креирање на Delivery објект.

Вредносни објекти

Во ова сценарио имаме неколку вредносни објекти. Едниот од нив, ќе го именувам како Money. Овој вредносен објект би го користеле ентитети Order (атрибутот Total), OrderItem (атрибутот Price), Meal (атрибутот Price). Ќе ги имплементираат основните операции за работа со пари, додавање, одземање, множење.

Другиот вредносен објект кој може да го имплементирам е PaymentInfo за еден корисник. Него би го користел ентитетот Customer (атрибутот PaymentInfo). Би се чувале број на кредитна картичка, име и презиме на корисник на картичка, датум на важење на картичката и кодот.

Следниот вредносен објект е Address. Овој вредносен објект би го имале ентитети Customer (атрибутот Address), Delivery (атрибутот Address). Во овој вредносен објект би чувале улица, број, град, држава.

Друг вредносен објект кој може да се дефинира, а би го користел ентитетот Customer е FullName кој го означува името и презимето на корисникот на системот.

Вредносен објект кој го користи Meal ентитетот е Ingredient. Тој ги означува состојките од кое е направено некое јадење. Се чува низа од карактери, која администраторот ја внесува со додавање/изменување на некое јадење.

На крај дефиниран е Quantity вредносен објект кој ја означува количината на порачано јадење во некоја ставка од нарачката. Овој вредносен објект овозможува зголемување/намалување на количината на некое јадење од ставка во нарачката на корисникот.

Настани

Кога корисникот ќе додаде некое јадење во својата нарачка се публикува настанот MealAddedInOrder, со што бројот на нарачки за соодветното јадење се зголемува. Инаку за секое јадење се дефинира број на нарачки кој би можел да го користат вработените за приготвување на нарачката. Се секое додавање на ставка во нарачката овој број се зголемува. Доколку корисникот избрише некоја ставка од нарачката се публикува настанот MealRemovedFromOrder, со што бројот на нарачки за јадењето се намалува. Овие два настани ги публикува Order микросервисот кон Meal микросервисот. Откако корисникот ќе ја плати нарачката се публикува настанот SuccessfullyPaidForOrder со што треба да се креира Delivery објект. Овој настан го публикува Order

микросервисот кон Delivery микросервисот. Откако нарачката ќе биде доставена, ќе се заврши со достава, се публикува настанот OrderSuccessfullyDelivered со што се менува статусот на нарачката. Овој настан го публикува Delivery микросервисот, кон Order микросервисот.