

Eindopdracht Web 4 Semester 2 2022-2023

FINAAL 25/04/2023

Inleiding

Het horeca bestelsysteem **Swagger** is ontworpen voor gebruik in de voedselbezorgingsindustrie.

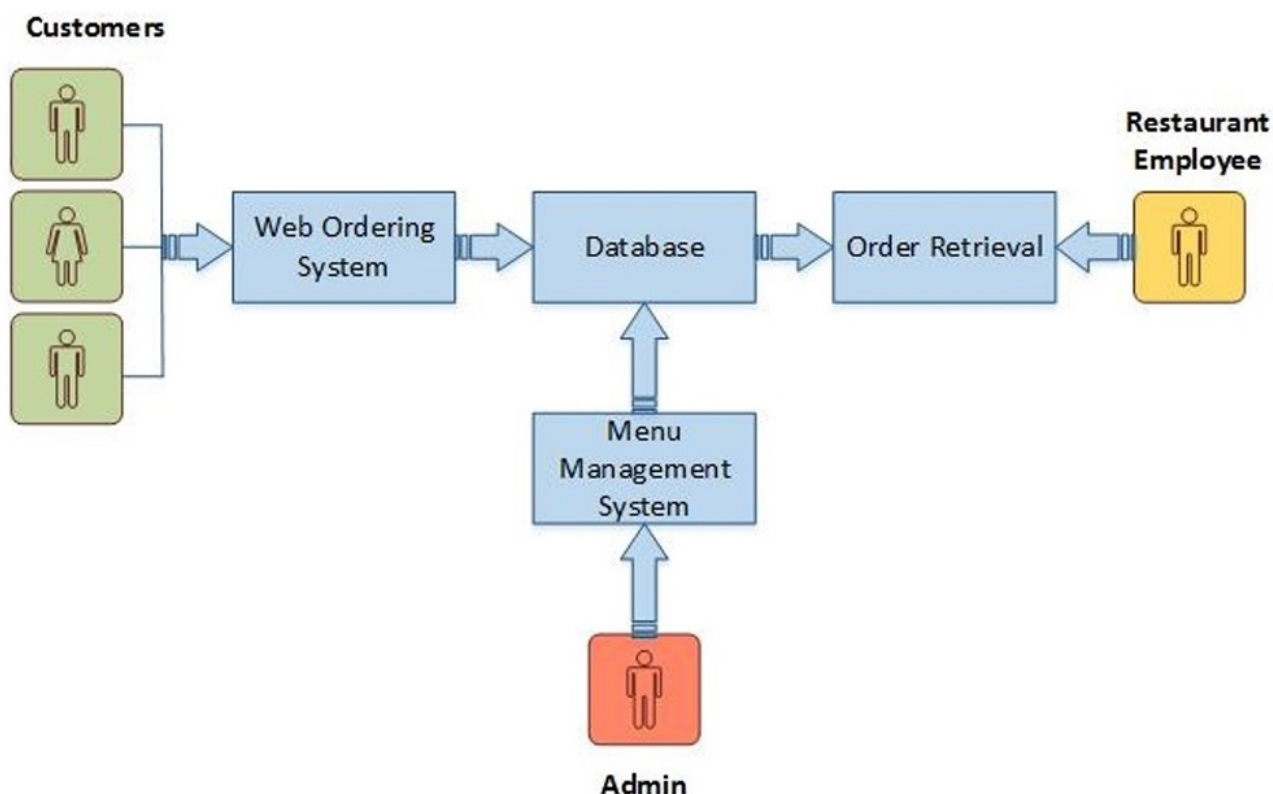
Het systeem maakt het mogelijk om snel en gemakkelijk een online menu te beheren dat klanten kunnen doorbladeren en gebruiken om bestellingen te plaatsen met slechts een paar klikken. Restaurantmedewerkers gebruiken deze bestellingen vervolgens via een gemakkelijk te navigeren grafische interface voor een efficiënte verwerking.



Het is algemeen bekend dat het in de huidige markt uiterst moeilijk is om een nieuw kleinschalig bedrijf te beginnen en de concurrentie van gevestigde en gevestigde eigenaars te doorstaan. In de

snelle tijd van vandaag, waarin iedereen in tijdnood zit, zijn de meeste mensen kieskeurig als het gaat om het plaatsen van een voedselbestelling. De klanten van vandaag voelen zich niet alleen aangetrokken tot deze manier van bestellen omdat online bestellen heel gemakkelijk is, maar ook omdat zij zicht hebben op de aangeboden artikelen, de prijs en een uiterst vereenvoudigde navigatie voor de bestelling.

Swagger vereenvoudigt het bestelproces aanzienlijk voor zowel de klant als het restaurant. Het systeem presenteert op een gebruiksvriendelijke manier een interactief en actueel menu met alle beschikbare opties. De klant kan één of meer artikelen kiezen om een bestelling te plaatsen. De klant kan alle bestelgegevens in de winkelwagen bekijken voordat hij afrekent. Aan het einde krijgt de klant een orderbevestiging. Zodra de bestelling is geplaatst, wordt deze ingevoerd in de database.



Opdracht

Je implementeert de volledige REST service achter Swagger en een kleine WPF of Razor applicatie die toelaat een beperkt deel van de REST service te gebruiken, namelijk enkel het menubeheer.

Opgelet: **gegevens worden nooit echt uit de databank verwijderd** en enkel gemarkeerd als niet meer geldig.

De REST service voorziet de volgende functionaliteit.

Klanten kunnen:

1. Een persoonlijk account aanmaken. Een account wordt bewaard in de databank en omvat minstens volgende gegevens:
 - naam

- voornaam.
- straat.
- huisnummer.
- busnummer.
- stad.
- postcode.
- land.
- telefoonnummer.
- emailadres.
- paswoord (hashed opgeslagen in de databank).
- een unieke gebruikerscode (wordt automatisch voorzien: UUID). Indien zinvol, controleer de geldigheid van deze gegevens en geef duidelijke foutmeldingen terug indien deze niet conform zijn.

2. Hun account beheren:

- verwijderen.
- alle eigenschappen aanpassen met uitzondering van de unieke gebruikerscode.

3. Het menu van het restaurant opvragen. Een menu bestaat uit een lijst van gerechten. Gerechten worden bewaard in de databank en omvatten minstens volgende gegevens:

- unieke referentiecode (wordt automatisch voorzien: UUID).
- naam.
- beschrijving.
- prijs in EUR.
- aantal verkrijgbaar.

4. Een bestelling aanmaken. Een bestelling wordt bewaard in de databank en omvat minstens volgende gegevens:

- unieke referentiecode (wordt automatisch voorzien: UUID)
- datum en tijdsmoment van aanmaak.
- datum en tijdsmoment van betaling.

5. Een gerecht toevoegen aan een bestelling zolang deze niet betaald en bijgevolg afgerond is - toch indien het gerecht nog steeds beschikbaar is.

6. Een gerecht verwijderen uit een bestelling zolang deze nog niet betaald is en bijgevolg niet afgerond.

7. Alle gerechten uit een nog niet betaalde bestelling verwijderen.

8. De bestelling plaatsen (dit wil zeggen: betalen).

9. Bevestiging van een geplaatste bestelling ontvangen via email.

10. De persoonlijk geplaatste bestellingen opvragen:

- alle.
- een specifieke aan de hand van de unieke referentiecode.

Gebruik de juiste REST sleutelwoorden voor de verschillende operaties (GET/POST/PUT/DELETE/...).

Als gebruikersinterface wordt Swagger gebruikt.

Een specifieke bijkomende set operaties is enkel beschikbaar voor de **administrator** van het systeem:

1. Een gerecht toevoegen.
2. Een gerecht verwijderen.
3. De eigenschappen van een gerecht aanpassen.

De set van operaties voor de administrator moet niet beveiligd worden.

Met behulp van de kleine en eenvoudige WPF of Razor applicatie die je vervolgens implementeert en die de REST API gebruikt, kan de administrator de rechten van het menu dat wordt getoond aan de klanten van het bestelsysteem, volledig beheren: - een gerecht toevoegen. - een gerecht verwijderen. - een gerecht aanpassen (uitgezonderd de unieke referentiecode).

Het is niet nodig om foto's van de rechten te voorzien (niet als bestand, niet in de databank, niet in je WPF- of Razor-client).

Voorzie minimale beveiliging:

1. Heartbeat, inclusief beschikbaarheid van je EF Core context
2. Rate limiting

Tools

- Visual Studio 2022 (.NET 7.0)
- git

Technisch

- SQLite of SQLServer of MySQL
- Entity Framework Core
- ASP.NET Core
- WPF of Razor

Lagen

1. Implementeer je REST API in meer lagen met een assembly per laag:
 - Repository.

- Domain.
 - REST API.
2. Voorzie een IRepository-laag waarachter Entity Framework Core weggeborgen zit zodat deze implementatie op termijn snel vervangen kan worden door bijvoorbeeld ADO.NET.
 3. Zorg ervoor door onder andere gebruik te maken van connection strings in appsettings.json dat SQLite op termijn snel vervangen kan worden door bijvoorbeeld SQLServer of MySQL.

Bonuspunten

Optioneel kan je je resultaat ophalen door voor een of meer van volgende aspecten te kiezen:

1. Overzichtelijke logging in client en server via SeriLog.
2. xUnit test suite voor je domein.
3. Razor in plaats van WPF.
4. Foto's van je gerechten.

Evaluatie

Je krijgt **tot twee kansen** om je werkende oplossing voor te stellen.

Voor de verdediging zullen tijdsloten op Chamilo aangemaakt worden: kies er eentje uit, individueel. Je kan de herkansing in principe enkel opnemen als je voor je eerste kans iets indiende en nog niet slaagde of als je gewettigde afwezigheid kan aantonen. Als je niets indiende bij de eerste kans, dan kan je met andere woorden het recht op je laatste kans kwijtspelen.

Indien een herkansing nodig is, zal deze ingepland worden **ongeveer 10 dagen na je eerste kans**.

Je kan telkens voor beide kansen **doorwerken aan je source code tot bijna aan het ogenblik waarop je je presentatie start**: net dat moment, laad je je source code op als **zip-bestand onder de juiste opdracht in Chamilo**.

De deadline ligt vast: dit betekent dat je niet zal kunnen indienen nadien! Organiseer je goed zodat je niet in tijdsnood komt.

Vragen?

Stel je vragen via een emailtje aan luc.vervoort@hogent.be.