Kassasysteem semester 6

Door Sem Kivits

Link to GitHub <https://github.com/solstice257/kassasysteem.git>

# Context

Het kassasysteem dat ik ga ontwerpen is bedoeld voor grootschalig gebruik bij grote events/festivals waarbij voornamelijk drank wordt verkocht. Het systeem is dus niet bedoeld om in horecazaken gebruikt te worden zoals: restaurants, hotels, ijscokarren, etc.

Om het makkelijker te maken om user stories, requirements, wireframes te maken en bedenken zullen wij het gebruik van het systeem beperken tot het populaire feest in omstreken Noord-Brabant, Carnaval.

# User stories

|  |  |
| --- | --- |
| User story | Omschrijving |
| US-01 | Als een gebruiker  Wil ik mijn bestellingen kunnen afrekenen  Zodat de afgerekende bestelling wordt opgeslagen |
| US-02 | Als een gebruiker Wil ik de prijs en het product kunnen controleren voordat het product wordt doorgeslagen Zodat ik zeker weet dat ik geen fouten maak met de prijs of het product |
| US-03 | Als een gebruiker  Will ik makkelijk toegang hebben tot de meest bestelde producten  Zodat ik sneller en efficiënter kan werken |
| US-04 | Als een kroegeneigenaar  Wil ik de totale omzet van een dag kunnen zien  Zodat ik kan zien hoe veel de winst op een dag is |
| US-05 | Als een kroegeneigenaar  Wil ik met meerdere kassa’s tegelijk kunnen werken  Zodat er op meerdere plekken kan worden besteld |

# Tools en gebruikte technieken

## Front-end

Voor dit project ga ik Svelte gebruiken om een front-end te maken. Toen ik onderzoek aan het doen was naar verschillende front-end frameworks kwam ik Svelte tegen als een hoog beoordeeld framework. Ik was geïnteresseerd in het framework dus besloot het te gebruiken.

## Backend

Voor de backend wil ik gaan werken met het .NET Core framework en met de taal C#. C# is de taal waar ik het bekendst mee ben. Hetzelfde geldt voor het .NET Core framework, hiermee ben ik ook al bekend. Net als mijn redenering om React te gebruiken voor het front-end, gebruik ik ook C# en .NET Core om meer mijn focus te kunnen leggen om dingen die ik nog niet weet.

## CI/CD

Om de geschreven code automatisch te testen en deployen zal ik gebruik maken van de CI/CD tools van AWS. Dit is voor mij de beste keuze aangezien ik voor een opdracht vanuit het stagebedrijf van S4 ook met AWS werk en zo mijn kennis kan worden overgedragen tussen de twee verschillende projecten.

Aangezien het gebruik van AWS niet gratis is zal ik eerst een pipeline opzetten met GitHub om kosten te besparen.

## Cloud deployment

Vanuit de AWS Pipeline kun je ook deployen naar een cloud service van AWS. Tegen het eind van het semester zal ik gaan kijken hoe ik mijn project in een container kan zetten in een AWS cloud omgeving. i.v.m. de kosten van een cloud service zal ik het eerst lokaal containers deployen.

# Requirements

## Functional requirements

|  |  |  |
| --- | --- | --- |
| **FR-01** | Gebruikers kunnen producten selecteren via een knop | |
| ***K01-1*** | | Op een knop drukken laat de prijs van het geselecteerde product zien |
| ***B01-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **FR-02** | Gebruikers kunnen de bestelling afronden door op een “bestel” knop te drukken | |
| ***K02-1*** | | Nadat je op de “bestel” knop drukt kun je niet meer bij de bestelling. Fouten zijn niet op te lossen door een algemene gebruiker |
| ***k02-2*** | | Voordat je de bestelling doorslaat krijg je nog een notificatie: “weet je zeker dat je dit product wil doorslaan?” |
| ***b02-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **FR-03** | Het systeem berekend automatisch de btw op de doorgeslagen producten | |
| ***K03-1*** | | Het programma houdt rekening met de verschillende btw-tarieven op verschillende producten |
| ***B03-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **FR-04** | De dagomzet wordt opgeslagen in een database | |
| ***K04-1*** | |  |
| ***B04-1*** | | De omzet van verschillende dagen is makkelijke te overzien |

|  |  |  |
| --- | --- | --- |
| **FR-05** | Gebruikers kunnen aangeslagen producten verwijderen voordat ze deze doorslaan | |
| ***K05-1*** | | Als een gebruiker de verkeerde knop indrukt dan moet je deze kunnen verwijderen uit de lijst met aangeslagen producten. |
| ***B05-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **FR-06** | Gebruikers hebben een account om toegang te krijgen tot de interface van het systeem | |
| ***K06-1*** | | Elke gebruiker krijgt een code van 2 cijfers om mee in te loggen in het systeem |
| ***B06-1*** | |  |

## Non-functional requirements

|  |  |  |
| --- | --- | --- |
| **nFR-01** | Bij het mergen met de dev-general branch, moet de code automatisch getest worden. | |
| ***K01-1*** | | Dit wordt gedaan d.m.v. een CI/CD pipeline en verschillende pipeline tools |
| ***B01-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **nFR-02** | Bij het ontwikkelen van de applicatie wordt veel aan de schaalbaarheid hiervan gedacht. | |
| ***K02-1*** | | De applicatie moet een groot aantal gebruikers kunnen verwerken. |
| ***B02-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **nFR-03** | De applicatie moet alleenstaand op een cloud omgeving komen te staan | |
| ***K03-1*** | | Hiervoor wordt Azure of AWS gebruikt |
| ***B03-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **nFR-04** | Bij het ontwikkelen van de applicatie wordt gedacht een de privacy van de gebruiker | |
| ***K04-1*** | |  |
| ***B04-1*** | | Zijn er privacy gerelateerde zaken waarmee rekening gehouden moet worden? |

|  |  |  |
| --- | --- | --- |
| **nFR-05** | Bij het ontwikkelen van de applicatie wordt de owasp top 10 security risks in acht genomen. | |
| ***K05-1*** | | Om dit te testen wordt “Owasp zap” gebruikt. |
| ***B05-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **nFR-06** | Data die bij het gebruiken van de applicatie wordt verwerkt, moet niet toegankelijk zijn voor mensen die er geen baat bij hebben | |
| ***K06-1*** | | Gegevens zoals de omzet moet alleen toegankelijk zijn voor mensen die daar geautoriseerd voor zijn. |
| ***B06-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **nFR-07** | De performance van de applicatie moet altijd op een gewenst niveau blijven. | |
| ***K07-1*** | | Een kassasysteem kan niet langzaam werken. Als het 2 seconde langer duurt om een bestelling te verwerken dan gewenst zal er veel omzet verloren gaan over een langere periode. |
| ***B07-1*** | |  |

|  |  |  |
| --- | --- | --- |
| **nFR-08** | De geschreven code moet van hoge kwaliteit zijn. | |
| ***K08-1*** | | Betere code zorgt ervoor dat de performance van een applicatie ook beter is. |
| ***B08-1*** | | Het schrijven van hoge kwaliteit code zorgt voor een betere schaalbaarheid van de applicatie. |

# MoSCoW

## Must

- FR01 Gebruikers kunnen producten selecteren via een knop

- FR02 Gebruikers kunnen de bestelling afronden door op een “bestel” knop te drukken

- FR03 Gebruikers kunnen aangeslagen producten verwijderen voordat ze deze doorslaan

## Should

- FR06 Gebruikers hebben een account om toegang te krijgen tot de interface van het systeem

- FR04 De dagomzet wordt opgeslagen in een database

## Could

- FR03 Het systeem berekend automatisch de btw op de doorgeslagen producten

# Entity relationship diagram

