Technisch Ontwerp

Use cases:

UC1 t/m UC3

Niveau 2  
Versie: 0.1  
Datum: 9-6-2025

Team SE  
2025

Versiebeheer

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Wijzigingen |
| 0.1 | 9-6-2025 | Initiële setup |
|  |  |  |

Distributie

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Ontvangers |
| 0.1 | 9-6-2025 |  |
|  |  |  |

Inhoud

[1 Inleiding 1](#_Toc201148258)

[2 Setup 2](#_Toc201148259)

[2.1 Repositories 2](#_Toc201148260)

[2.2 Runnen project 2](#_Toc201148261)

[3 Technieken 3](#_Toc201148262)

[3.1 Tools 3](#_Toc201148263)

[3.2 Programmeertalen 3](#_Toc201148264)

[3.3 Frameworks 3](#_Toc201148265)

[4 Definition of Done 4](#_Toc201148266)

[5 Systeem Context (C4 niveau 1) 5](#_Toc201148267)

[6 Container Context (C4 niveau 2) 6](#_Toc201148268)

[7 Componenten Context (C4 niveau 3) 7](#_Toc201148269)

[7.1 Desktop applicatie 7](#_Toc201148270)

[7.2 Security Maatregelen 8](#_Toc201148271)

[8 Code Context (C4 niveau 4) 9](#_Toc201148272)

[8.1 Packages 10](#_Toc201148273)

[8.2 Configuraties 10](#_Toc201148274)

[Figuren 12](#_Toc201148275)

[Bibliografie 13](#_Toc201148276)

# Inleiding

In dit technisch ontwerp wordt een systeem beschreven dat zich richt op het ontwerp en de implementatie van de use case die is beschreven in het Functioneel Ontwerp. Dit systeem stelt geïnteresseerden in staat om een beeld te vormen van de skills van een student-webdeveloper. Het ontwerp omvat een gedetailleerde analyse van de systeemcontext, container- en componentdiagrammen.

Veel plezier met ontwerpen!

Team SE

# Setup

In dit hoofdstuk een beschrijving hoe het systeem lokaal te runnen is.

## Repositories

Om het systeem te kunnen runnen is het noodzakelijk om de bijbehorende repositories te clonen. Deze zijn te vinden in de studentenhandleiding.

## Runnen project

Open de solution in Visual Studio 2022 of een soortgelijke tool. Voor het runnen van het project heb je .NET 8 nodig en apart daarvan MAUI. Vervolgens is het project direct te starten, zonder verdere configuratie.

# Technieken

In dit hoofdstuk wordt een overzicht gegeven van de gebruikte tools en standaarden.

## Tools

In deze paragraaf een overzicht van de gebruikte tools.

**Visual Studio**

Visual Studio is een IDE die de ontwikkelaar ondersteunt bij het schrijven van code, onder andere door middel van auto-completion

**GitHub**GitHub wordt gebruikt voor de opslag en het beheer van de Git repository. Ook kan gebruikgemaakt worden van de CI/CD om code te deployen naar Cloudflare en Skylab.

## Programmeertalen

Bij het ontwikkelen van het systeem is een programmeer taal gebruikt. Hieronder volgen de talen die gebruikt zijn.

**C#**De .NET applicatie is geschreven in C#.

## Frameworks

Bij het ontwikkelen van het systeem is gebruik gemaakt van een framework. Deze zijn hieronder beschreven.

**.NET** (Microsoft, 2023)

Voor het ontwikkelen van de back-end is gebruik gemaakt van .NET Maui.

# Definition of Done

In dit hoofdstuk is de definition of done uitgewerkt. Dit zijn de eisen, waar nieuwe functionaliteit (technisch) aan moet voldoen, voordat deze kan worden afgerond. Zoals vastgelegd in het projectplan zijn deze eisen voorgelegd aan de opdrachtgever. Deze DOD zou in plaats van in het TO opgenomen kunnen worden in andere projectdocumentatie.

|  |  |
| --- | --- |
| Design | |
| 1. | Het FO en TO weerspiegelen de gerealiseerde functionaliteit |
| 2. | De documentatie heeft een consistente opmaak en duidelijk gestructureerde hoofdstukken. |
| 3. | De documentatie is opgesteld volgens de Windesheim standaarden. |
| Development and Testing | |
| 4. | De gerealiseerde functionaliteit voldoet aan alle acceptatiecriteria (vastgelegd in het functioneel ontwerp) |
| 5. | De gerealiseerde functionaliteit voldoet aan de eisen gesteld in het issue ([zie issue templates](https://github.com/Windesheim-HBO-ICT/webdev-showcase/tree/master/2324%20semester%201/DOD/issue-templates)) |
| 6. | De testen van de gerealiseerde functionaliteit slagen allen |
| Deployment | |
| 7. | De pipeline van de feature branch slaagt |
| 8. | Alle secure parameters zijn opgenomen als environment variabelen |

# Systeem Context (C4 niveau 1)

Dit hoofdstuk geeft een overzicht van het systeem. De klant is iemand die boodschappenlijsten wilt maken en beheren. De klant is een persoon in de context van een bezoeker van een supermarkt.

A diagram of a company

AI-generated content may be incorrect.

Figuur 1 C4 level 1 van de Boodschappen App

# Container Context (C4 niveau 2)

Dit hoofdstuk beschrijft de containers waaruit het systeem bestaat. Voor de Boodschappen App is er één container, de desktop applicatie. Dit is een desktop applicatie gemaakt in het framework van MAUI. Dit is te zien in Figuur 2.

A diagram of a computer application

AI-generated content may be incorrect.

Figuur 2 C4 level 2 van de Boodschappen App

# Componenten Context (C4 niveau 3)

In dit hoofdstuk is voor de componenten van de containers van de Boodschappen app getoond hoe de architectuur is vormgegeven.

## Desktop applicatie

Voor de desktop applicatie is gebruik gemaakt van de architectuur MVVM (Model-View-ViewModel). Zoals te zien is in Figuur 3 heeft de klant alleen interactie met de *views* van de applicatie. De *views* zijn verantwoordelijk voor de opmaak en weergave van de gebruikersinterface. Aan de hand van databinding wordt een *view* gekoppeld aan een *viewmodel*. Hiermee heeft de *view* geen logica nodig en kan dat op andere plekken worden afgehandeld. De *viewmodel* fungeert als de tussenlaag tussen de *view* en de *model*. In de *viewmodel* zit alle logica die nodig is voor de *view,* zoals het ophalen van gegevens en afhandelen van gebruikersacties. De *models* bevatten de dataclasses (entiteiten), waar geen logica in is opgenomen, maar wel de data in wordt opgeslagen.

A diagram of a company

AI-generated content may be incorrect.

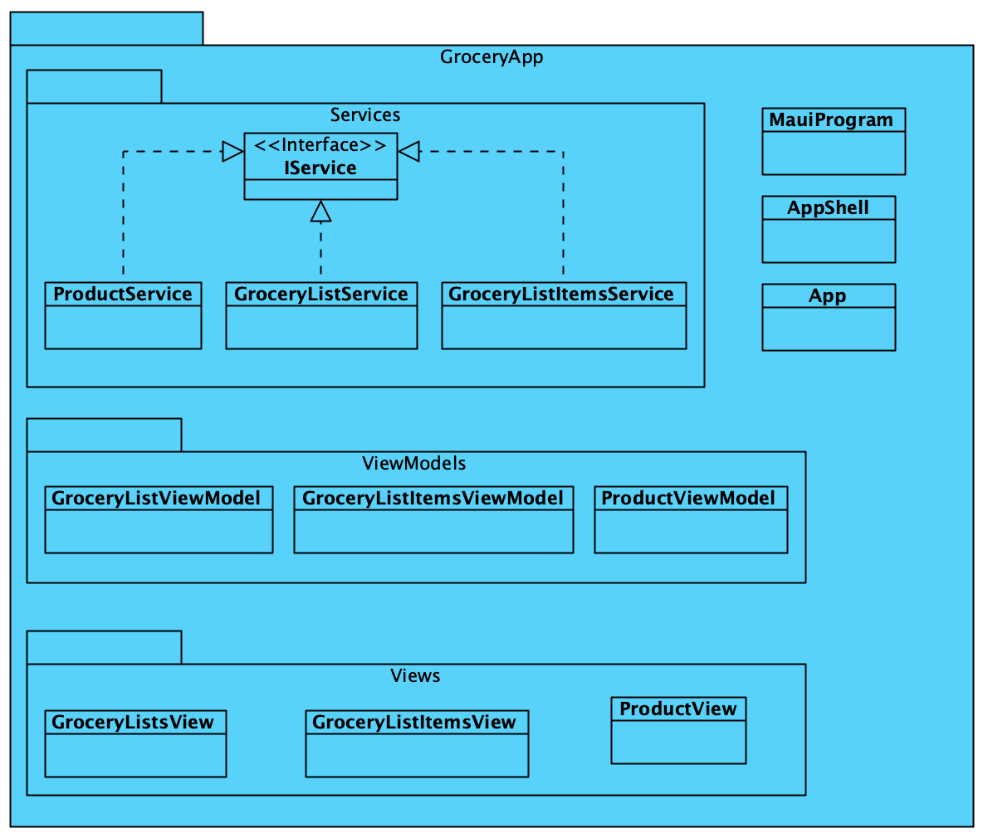
Figuur 3 C4 level 3 van de Boodschappen App

## Security Maatregelen

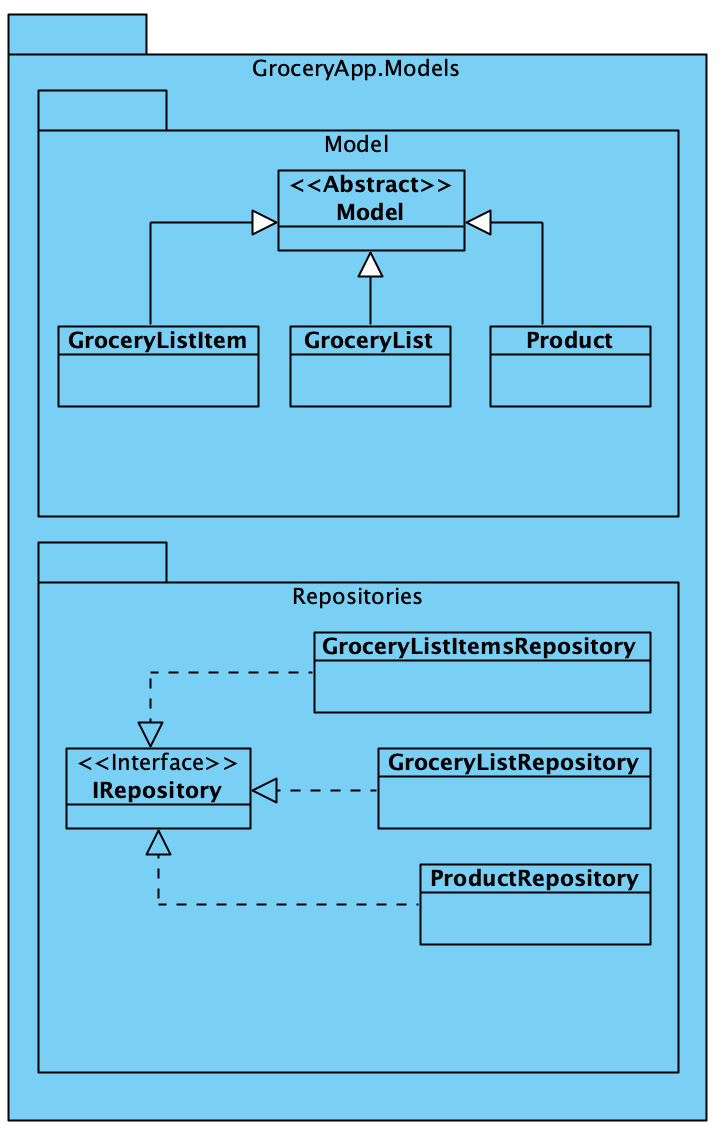
Omdat geen Threat Model is uitgevoerd zijn ook geen aanvullende security maatregelen genomen. Geen nadere actie nodig.

# Code Context (C4 niveau 4)

Voor de desktop applicatie is een klassendiagram gemaakt, te zien in Figuur 4 en Figuur 5. De GroceryApp bestaat uit drie verschillende projecten: GroceryApp, GroceryApp.Models en GroceryApp.Data. Deze projecten zijn zo opgesteld zodat de juiste scheiding van verantwoordelijkheden wordt aangehouden volgens de three-tiers architectuur. De GroceryApp.Data heeft nog geen klassendiagram, omdat deze nog geen implementaties bevat.



Figuur 4 Klassendiagram van GroceryApp



Figuur 5 Klassendiagram van GroceryApp.Models

## Packages

|  |  |  |
| --- | --- | --- |
| Naam | Versie | Beschrijving |
| MAUI.Controls | 8.0.100 | .NET Multi-platform App UI (.NET MAUI) is een cross-platform framework om native mobiele en desktop applicaties met C# en XAML te maken. Door .NET MAUI te gebruiken, kan je apps developen die op Android, iOS, macOS, Tizen en Windows kunnen runnen op één codebase. |
| CommunityToolKit.MVVM | 8.4.0 | Helper package om te werken met MVVM. |

## Configuraties

Om een nieuwe repository toe te voegen aan de applicatie, moet er in de MauiProgram.cs een line toegevoegd worden, waarbij de *Model* en *Repository* vervangen moeten worden met de correct klasse namen.

builder.Services.AddSingleton<IRepository<Model>, Repository>();

# Figuren

[Figuur 1 C4 level 1 van de Boodschappen App 5](#_Toc201148277)

[Figuur 2 C4 level 2 van de Boodschappen App 6](#_Toc201148278)

[Figuur 3 C4 level 3 van de Boodschappen App 7](#_Toc201148279)

[Figuur 4 Klassendiagram van GroceryApp 9](#_Toc201148280)

[Figuur 5 Klassendiagram van GroceryApp.Models 10](#_Toc201148281)

# Bibliografie

Cypress.io. (2023). *Why Cypress.* Opgehaald van Cypress: https://docs.cypress.io/guides/overview/why-cypress

ESLint. (2023). *Documentation.* Opgehaald van ESLint: https://eslint.org/docs/latest/

Microsoft. (2023). *.NET documentation.* Opgehaald van Microsoft: https://learn.microsoft.com/en-us/dotnet/

NUnit. (2023). *NUnit.* Opgehaald van NUnit Documentation Site: https://docs.nunit.org/