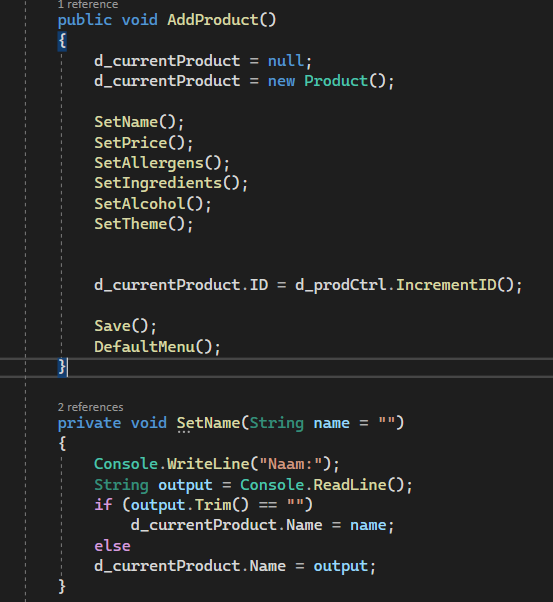
Project-B Jakes Restaurant

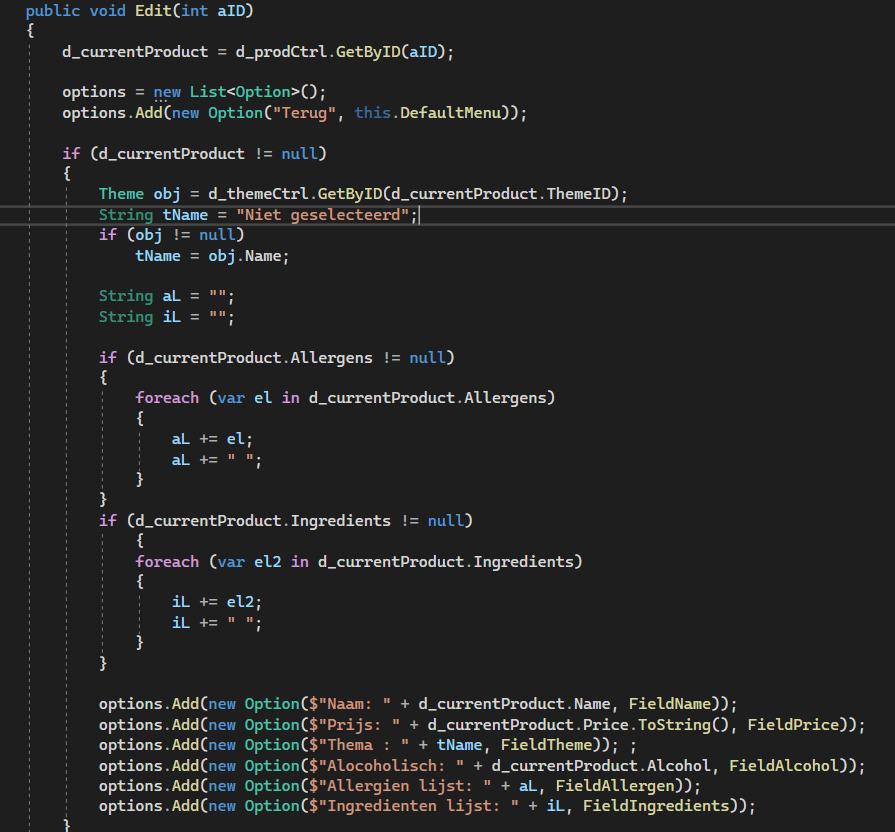
# Front-end:

**Product management:**

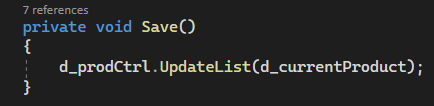
Het toevoegen van een product heb ik doormiddel van de functie AddProduct gemaakt. Het toevoegen van een product heb ik voor makkelijk onderhoud in meerdere functies gezet. Deze functies vragen aan de gebruiker voor het invoeren van data. De functie SetName is hier een voorbeeld van.

****

Om een product te wijzigen heeft de gebruiker vanaf een lijst een product geselecteerd. In de Edit functie wordt op een visuele manier het geselecteerde product weergegeven. Om een item te wijzigen zal de gebruiker het gewenste item moeten selecteren.

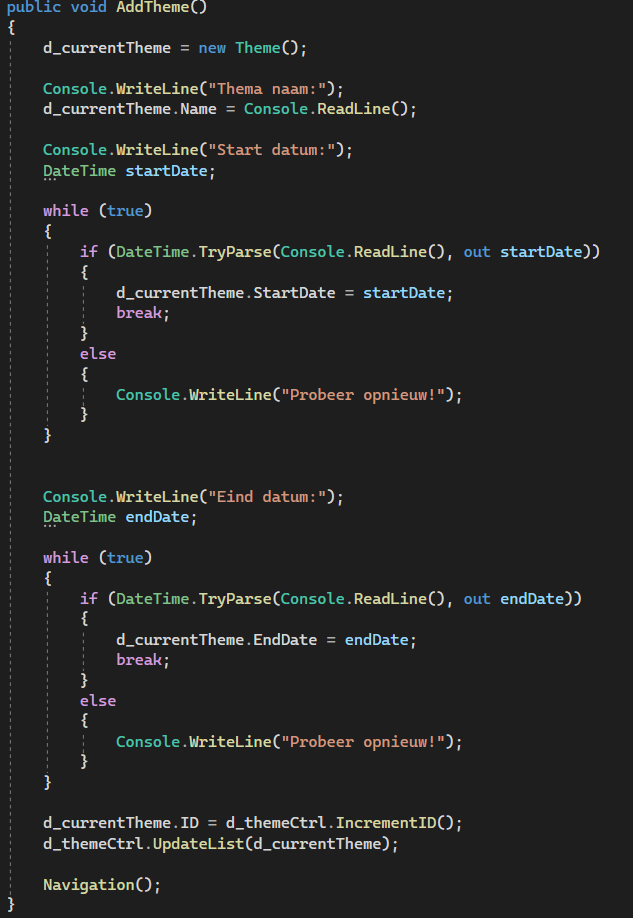


De save functie zorgt ervoor dat het nieuwe of aangepaste product opgeslagen wordt naar een json file.



**Thema:**

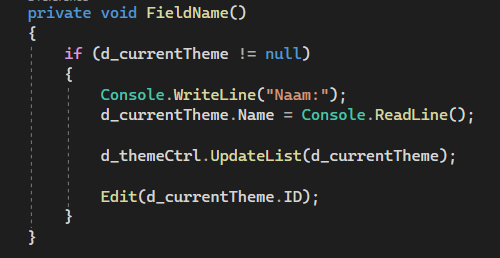
Het toevoegen van een thema gebeurd door de functie AddTheme. In deze functie is het erg belangrijk dat correcte gegevens worden ingevoerd zoals een start en eind datum. Deze functie controlleerd alleen of het een geldige datum is.



Om een thema te wijzigen heeft de gebruiker vanaf een lijst een thema geselecteerd. Via het menu kan de gebruiker het gewenste item wijzigen zoals de naam, start datum etc.



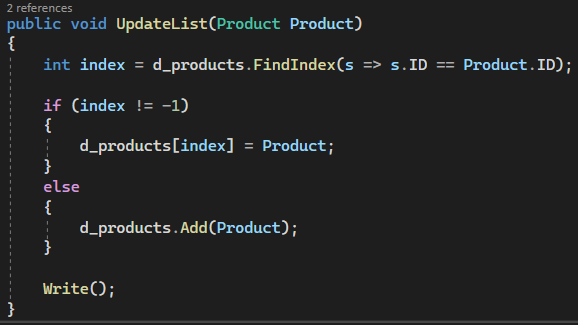
De functie FieldName wordt vanaf het menu aangeroepen om de gebruiker een nieuwe naam in te laten voeren. Deze functie zal het aangepaste thema opslaan en terug navigeren naar het geselecteerde thema.



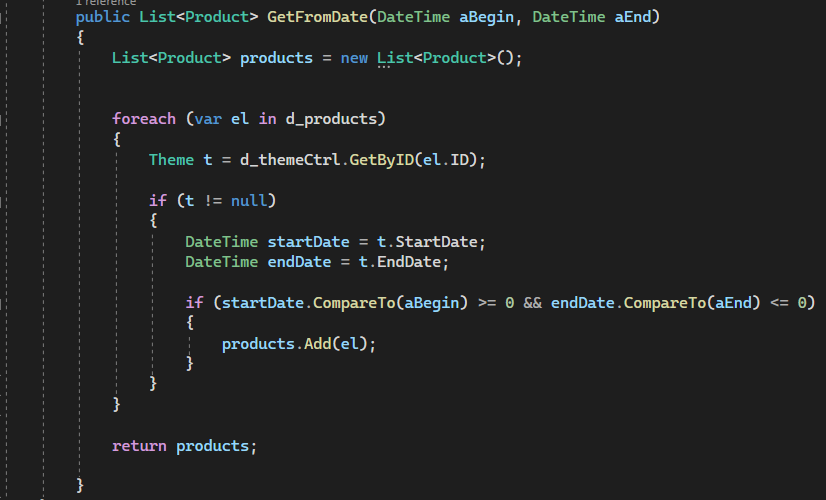
# Back-end:

**Product management:**

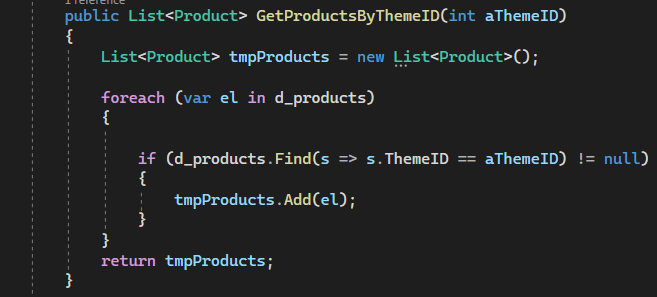
De functie UpdateList wordt gebruikt om een product dat nog niet bestaat op te slaan en een product dat al bestaat te overschrijven met nieuwe informatie.

****

De functie GetFromDate haalt producten op die seizoen gebonden zijn. Oftewel gekoppeld via een thema. Wanneer de ingestelde begin en eind datum gevonden zijn zal deze toegevoegd worden in een lijst van producten.

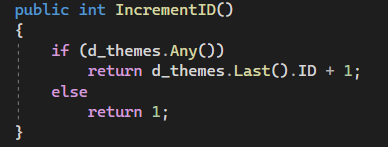


De functie GetProductsByThemeID haalt alle producten op die gekoppeld zijn aan een specifieke thema ID.

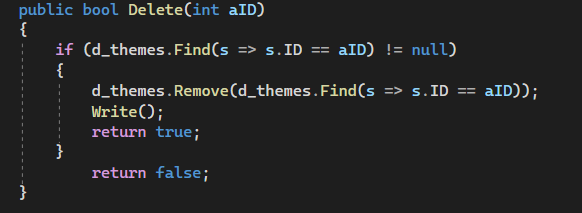


**Thema management:**

De functie IncrementID wordt bij het aanmaken van een object aangeroepen. Deze functie zal het laatste item in de lijst opzoeken en het ID + 1 terug geven. Als er geen items gevonden zijn in de lijst zal het eerste object een ID van 1 krijgen.



Het verwijderen van een Thema is doormiddel van de functie Delete gerealiseerd. Deze functie zoekt in een lijst naar een thema die met het opgegeven ID verwijderd moet worden. Als dit thema gevonden is zal deze uit de lijst verwijderd worden en opgeslagen worden naar een json file.

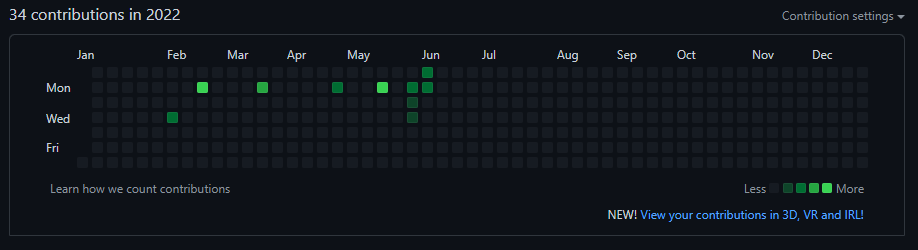


# Git Contribution:

In deze periode ben ik bekend gewordem met het gebruik van GIT. Ik heb geleerd dat het belangrijk is om tijdig te commiten om conflicten tegen te gaan.

<https://github.com/MichaelHeerkens01>





# Completed tasks:

|  |  |
| --- | --- |
| Ticket | Sub-tickets |
| Front-end | |
| Products view | * Add * Edit * Delete * View based on theme ID * View based on theme date |
| Themes view | * Add * Edit * Delete |
| Test case (Products) |  |
| Test case (Themes) |  |
| Back-end | |
| Product management | * Add * Edit * Delete * GetProductsByThemeID * GetFromDate * GetByID * IncrementID |
| Theme management | * Add * Edit * Delete * GetByID * IncrementID |
| Remodel products |  |
| Rename components product management |  |