

# Programmeren Gevorderd: Gentse Feesten

## Inhoudstafel

1. Deadlines .....	1
1.1. Indienen .....	1
2. Setting .....	2
3. Architectuur en puntenverdeling .....	3
3.1. Domeinlaag (20%) .....	3
3.2. Unit testen (10%) .....	3
3.3. Persistentielaag (30%) .....	4
3.4. Gegevensinitialisatie (10%) .....	4
3.5. Presentatielaag (30 %) .....	4
4. Bonus! .....	4
5. Te gebruiken technologieën .....	4

## 1. Deadlines

- Deadline: 26 mei 2023 om 23:59
- Mondelinge verdediging: in de week van 29 mei



Wie de opdracht niet indient voor de deadline wordt niet uitgenodigd voor de mondelinge verdediging. De tijdsindeling voor mondelinge verdediging wordt kort na de deadline bekendgemaakt.



Dien gerust meerdere keren in op Chamilo, enkel de laatste versie wordt bekeken en verbeterd maar zo ben je wel bestand tegen calamiteiten op de dag van de deadline.

### 1.1. Indienen

Bij het indienen worden er meerdere zaken verwacht

1. Een zip file van de volledige solution: Je Visual Studio project folder, en dit zonder de subfolders `bin` en `obj`.
2. Een export van je schema's en data uit je database in een `.sql` script:
  - Hoe je dit doet vind je terug in deze YouTube tutorial: [www.youtube.com/watch?v=vckr6vt-FHM](https://www.youtube.com/watch?v=vckr6vt-FHM) vanaf 4'39".
  - Exporteer al je tabellen: schema en data

## 2. Setting

Voor de Gentse Feesten van 2023 wil het stadsbestuur een applicatie ontwikkelen voor VIP personen. De applicatie staat de gebruiker toe om evenementen per dag te plannen, en ze krijgen als promotie ook een vast dagbedrag.



Een overzicht van de beschikbare evenementen vind je in een CSV-bestand in bijlage.

Om een dagplan te maken gelden volgende regels (dit zijn **domeinregels!**):

1. Hetzelfde evenement kan op één dag niet tweemaal worden toegevoegd aan een dagplan.
2. Een evenement in een dagplan mag niet overlappen met een ander evenement.
3. Een evenement moet natuurlijk plaatsvinden op dezelfde datum als de datum waarvoor het dagplan gemaakt wordt.
4. Een gebruiker mag voor een bepaalde dag slechts één dagplan aanmaken.
5. Tussen alle evenementen op een dagplan moet minstens een periode van 30 minuten zitten, zodat men tijd heeft zich naar het volgende evenement te begeven.
6. De kostprijs van alle evenementen samen per dag mag het beschikbare dagbedrag van de gebruiker niet overschrijden.



Indien één van deze situaties optreedt wordt de gebruiker verhinderd om het evenement toe te voegen aan een dagplan en wordt in een boodschap aan de gebruiker verteld waarom een evenement niet toegevoegd kan worden.

Voor een **dagplan** kan worden opgevraagd:

- Datum
- Bezoeker met info
- Evenement(en) met tijd en kostprijs
- Totale kostprijs
- Andere nuttige info...

Voor een **gebruiker** kan worden opgevraagd:

- Zijn gegevens (naam en voornaam)
- Een overzicht van zijn dagplannen
- Zijn resterend dagbudget voor die dag
- Andere nuttige info...

Voor een **evenement** kan worden opgevraagd:

- Titel
- Beschrijving
- Rooster: Tijdstippen waarop het evenement plaatsvindt
- Tijdsduur
- Prijs
- Andere nuttige info...

## 3. Architectuur en puntenverdeling

Je maakt gebruik van een **3-lagen architectuur**: presentatie - domein - persistentie eventueel aangevuld met een startup.

### 3.1. Domeinlaag (20%)



Alle domeinregels moeten worden afgedwongen in de domeinlaag en niet in de presentatie- of de persistentielaag. Ontwikkel een domeinmodel met klassen en gebruik geen DataTable op je domeinlaag.

- Let op de relaties tussen de klassen: maak voor jezelf eventueel een klassendiagram.
- Maak waar nodig gebruik van DTO objecten om informatie door te geven aan de presentatielaag.

### 3.2. Unit testen (10%)

- Voorzie unit testen in een afzonderlijke assembly voor deze domein klassen met het xUnit framework.
- Belangrijk is minstens de nodige testen uit te voeren voor de domeinregels.

### 3.3. Persistentielaag (30%)

- Indien primary key id's aangemaakt moeten worden in je oplossing, dan dient dit te gebeuren door de databank (maak gebruik van IDENTITY(1,1)).
- Wanneer je de inhoud van de databank aanpast, gebruik dan transacties waar nodig; dit is zeker van belang wanneer je bij een bepaalde operatie in meer tabellen tegelijk moet schrijven en alle schrijfacties succesvol moeten uitgevoerd zijn alvorens de operatie als doorgevoerd beschouwd mag worden.
- Let erop dat je parameters gebruikt waar dit kan (om SQL Injection te voorkomen).

### 3.4. Gegevensinitialisatie (10%)

- Maak een afzonderlijke console-applicatie om gegevens op te laden naar je databanktabellen (= CVS file parsen en hiermee de databank opvullen)
- Zoals reeds aangehaald vind je de evenementen in bijgevoegd bestand.
- Laad daarenboven minstens 50 VIP gebruikers op uit een .csv bestand dat je zelf aanmaakt. ("Let ChatGPT be your assistant for this one!")
- Maak de tabellen van je databank eerst leeg alvorens je het opladen start.

### 3.5. Presentatielaag (30 %)

- Maak de presentatielaag met WPF en bouw minstens de volgende schermen:
  1. Een overzicht van alle gebruikers.
  2. Opzoeken en selecteren van gebruiker op naam (→ implementeer een filter op naam)
  3. Maken van een nieuw dagplan voor de geselecteerde gebruiker met de selectie van evenementen (→ toon enkel evenementen die op de datum van het dagplan plaatsvinden).
    - a. Bij de selectie van evenementen kan je opzoeken op titel (→ implementeer een filter op titel)
  4. Selecteren van een bestaand dagplan van de geselecteerde klant. Toon alle gegevens van dit dagplan.

## 4. Bonus!

Maak het mogelijk om een geselecteerde dagplan te exporteren naar een document in AsciiDoc formaat (waarvan de syntax werd aangeleerd in Analyse). Dit document bevat de nodige gegevens van de gebruiker en zijn geplande evenementen.

## 5. Te gebruiken technologieën

- Visual Studio
- C#

- .NET 6.0 of hoger
- SQL server Express 2022 en ADO.NET
- WPF
- Excepties om fouten (o.a. bij de domeinregels) op te vangen
- Drie lagen architectuur