Technisch ontwerp

14-03-2022

Piotr Tadrala

AP2O-S201

Versie: 1.7

2022

Inhoud

[**Versiebeheer** 2](#_Toc104904882)

[**Class Diagram** 2](#_Toc104904883)

[**ERD** 3](#_Toc104904884)

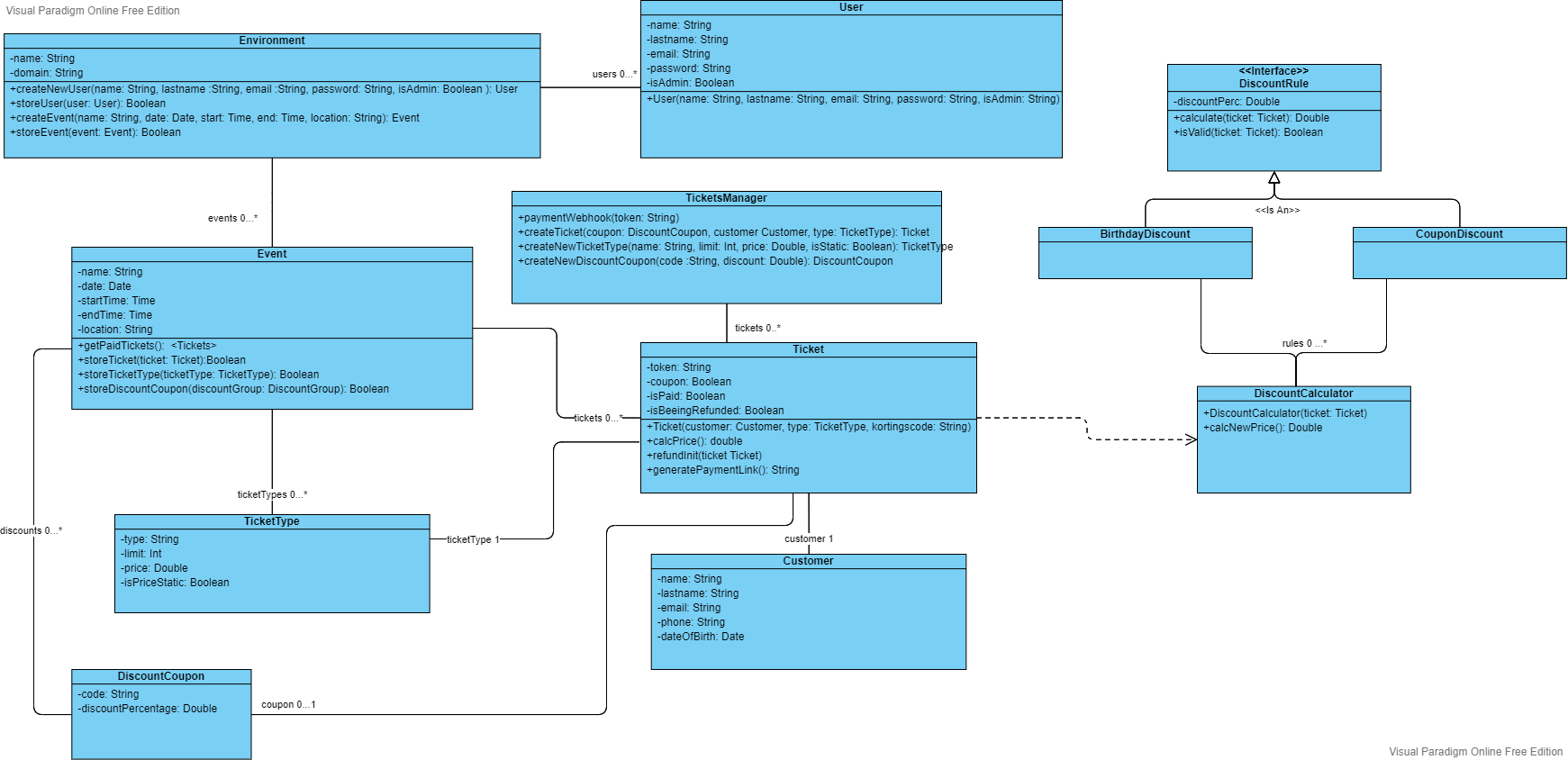
[**Algoritme** 4](#_Toc104904885)

[**Architectuur** 5](#_Toc104904886)

# **Versiebeheer**

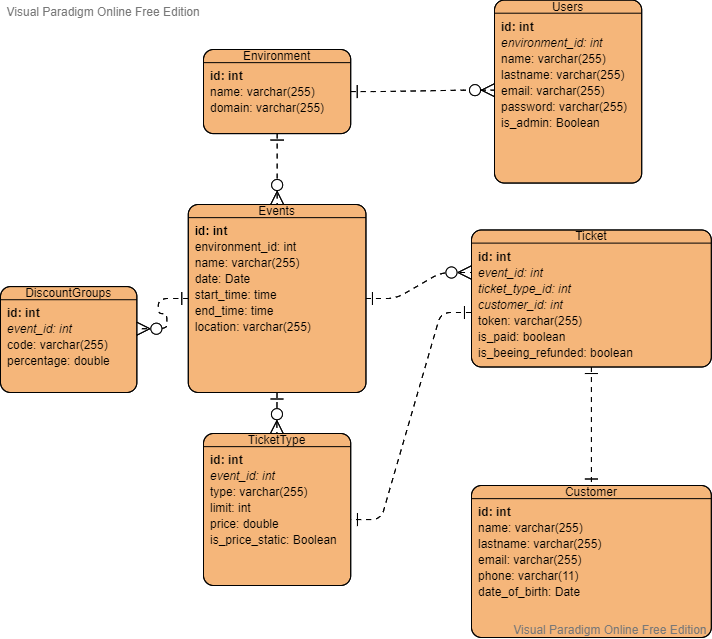
|  |  |  |
| --- | --- | --- |
| Versie | Aanpassing | Datum |
| 1.0 | Class Diagram | 14-03-2022 |
| 1.1 | Class Diagram relaties aangepast en Use Cases toegevoegd. | 28-03-2022 |
| 1.2 | Class Diagram relaties aangepast | 05-04-2022 |
| 1.3 | Class ‘DiscountGroup’ toegevoegd | 10-04-2022 |
| 1.3 | Nieuwe methodes toegevoegd aan Environment, Event en Ticket | 10-04-2022 |
| 1.4 | Algoritme methodes toegevoegd | 19-04-2022 |
| 1.5 | ERD en Algoritme toegevoegd | 28-04-2022 |
| 1.6 | Architectuur toegevoegd | 24-05-2022 |
| 1.6 | Class Diagram aangepast | 24-05-2022 |
| 1.7 | Class diagram:   * EventContainer weggehaald * Associatie tussen Environment en Events aangemaakt * Event methodes verplaatst naar Envrionment en TicketManager | 15-06-2022 |
| 1.7 | Aannames en overwegingen toegevoegd | 15-06-2022 |
| 1.7 | Architectuur aangepast | 15-06-2022 |

# **Class Diagram**

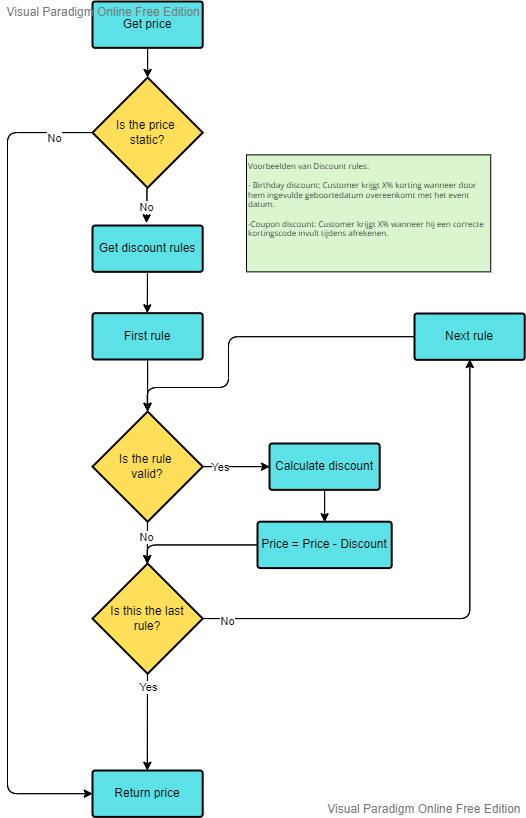


[**Source**](https://imgur.com/a/1vY8qTd)

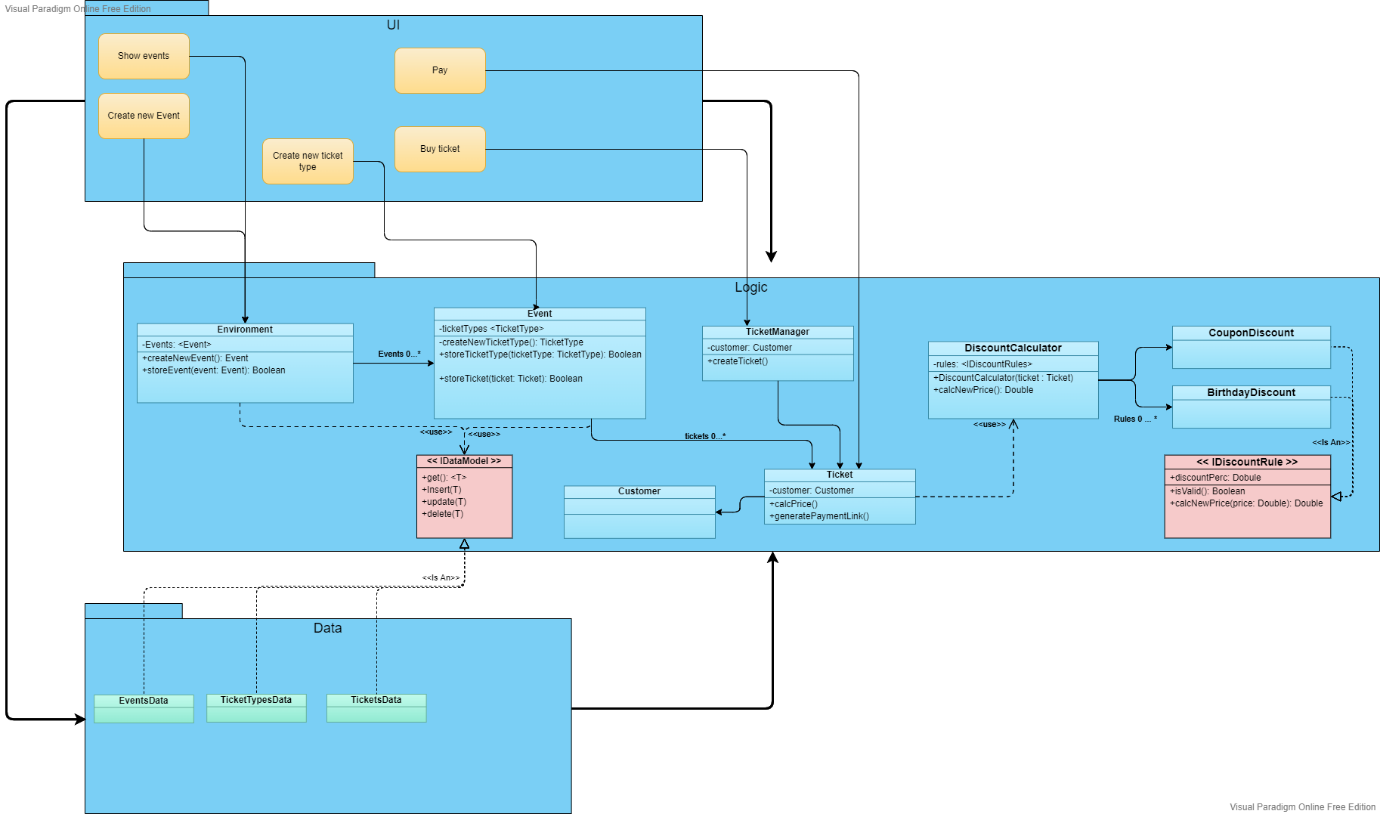
# **ERD**



# **Algoritme**



# **Architectuur**



[**Source**](https://imgur.com/a/kwMwEAA)

# **Aannames en overwegingen**

Tijdens het ontwerpen van mijn technisch ontwerp kwam ik een paar dilemma's tegen. een daarvan was hoe kan ik mijn applicatie aan 'Single responsibility' kan laten voldoen. In eerste instantie had ik onnodig veel methodes in een class zitten, namelijk 'Environment'. Na overleg met René heb ik een oplossing kunnen bedenken. Ik heb een nieuwe container class geïntroduceerd ‘TicketsManager' die ticket specifieke methodes overneemt.

Zodra ik klaar was met mijn technisch ontwerp ben ik aan de slag gegaan met de architectuur van mijn applicatie. Ik heb gekozen voor een architectuur die richting het logica layer wijst om aan 'Dependency inversion principle' te voldoen. Hierdoor kan ik mijn logica layer afhankelijk maken van abstractie in plaats van harde koppelingen tussen de layers.  Applicatie is hierdoor ook veel meel modulair aangezien de layers binnen de applicatie 'loosely coupled' zijn.