

# Technisch ontwerp

Piotr Tadralla  
AP2O-S201  
VERSIE: 1.7

14-03-2022

## Inhoud

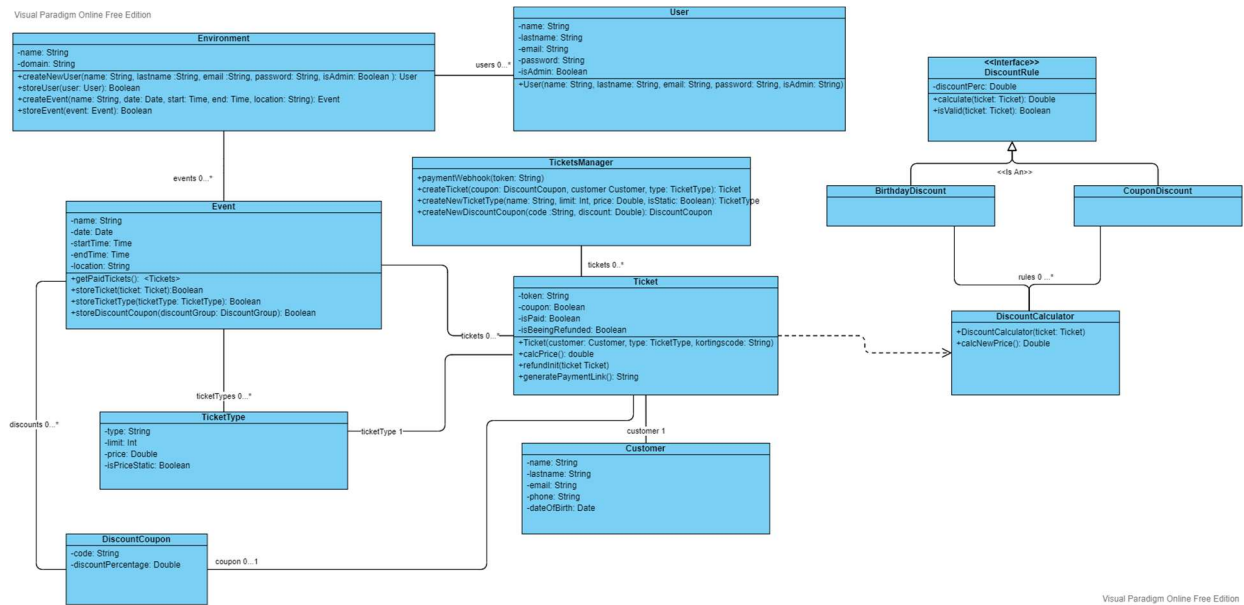
<b>Versiebeheer .....</b>	<b>2</b>
<b>Class Diagram .....</b>	<b>3</b>
<b>ERD.....</b>	<b>4</b>
<b>Algoritme .....</b>	<b>5</b>
<b>Architectuur.....</b>	<b>6</b>

## 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: <ul style="list-style-type: none"><li>• EventContainer weggehaald</li><li>• Associatie tussen Environment en Events aangemaakt</li><li>• Event methodes verplaatst naar Environment en TicketManager</li></ul>	15-06-2022
1.7	Aannames en overwegingen toegevoegd	15-06-2022
1.7	Architectuur aangepast	15-06-2022

# Class Diagram

Visual Paradigm Online Free Edition

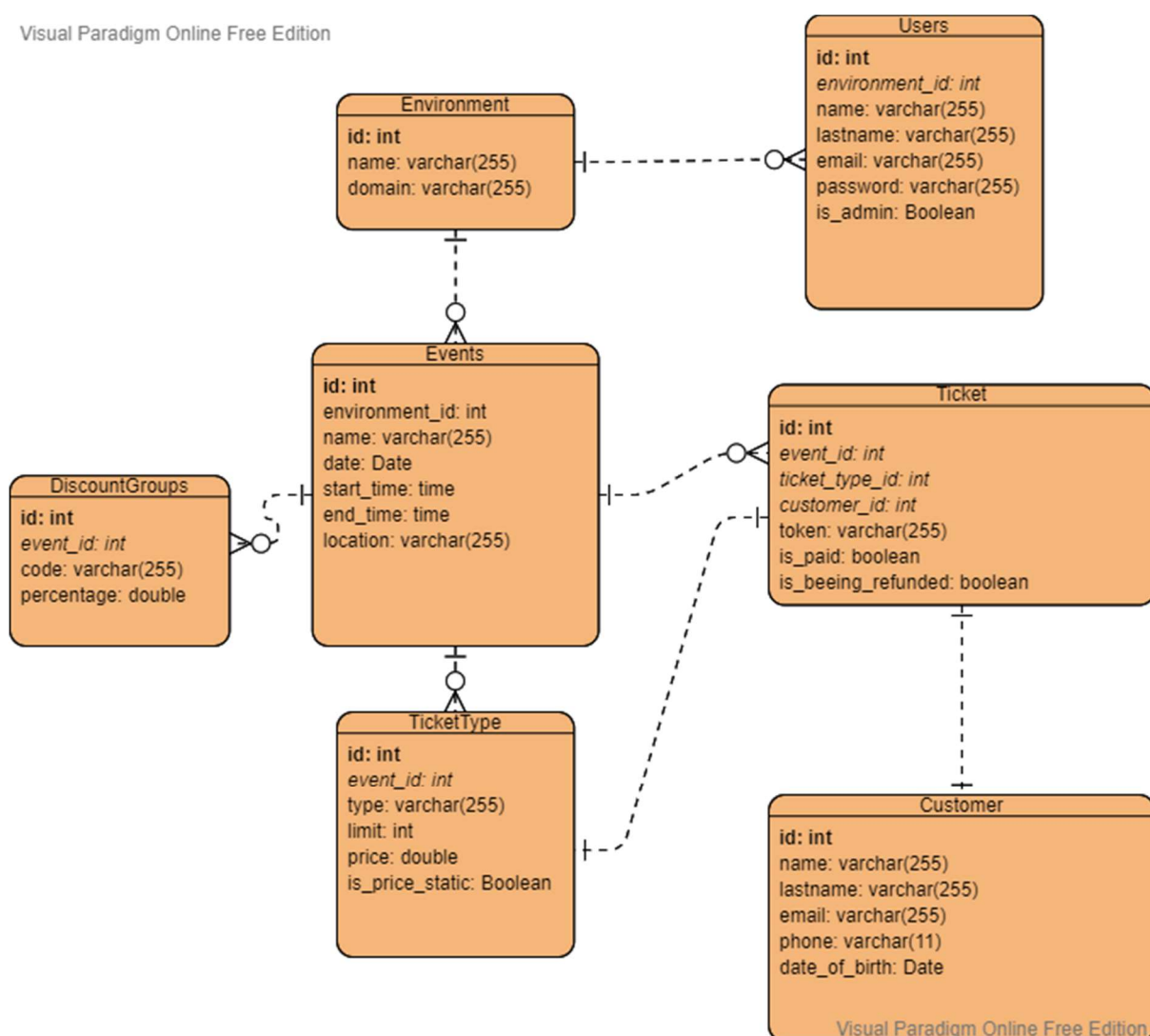


Visual Paradigm Online Free Edition

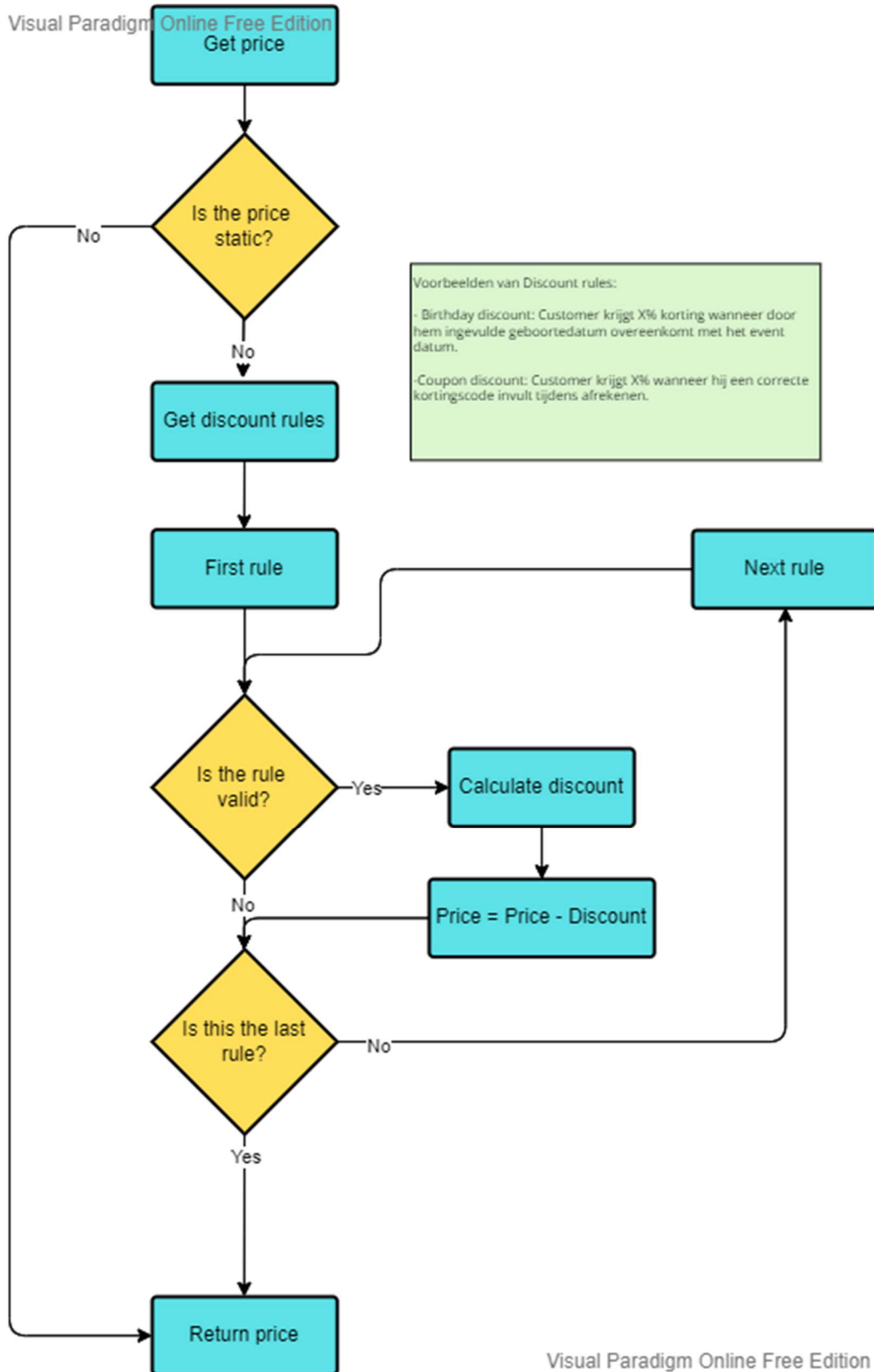
[Source](#)

# ERD

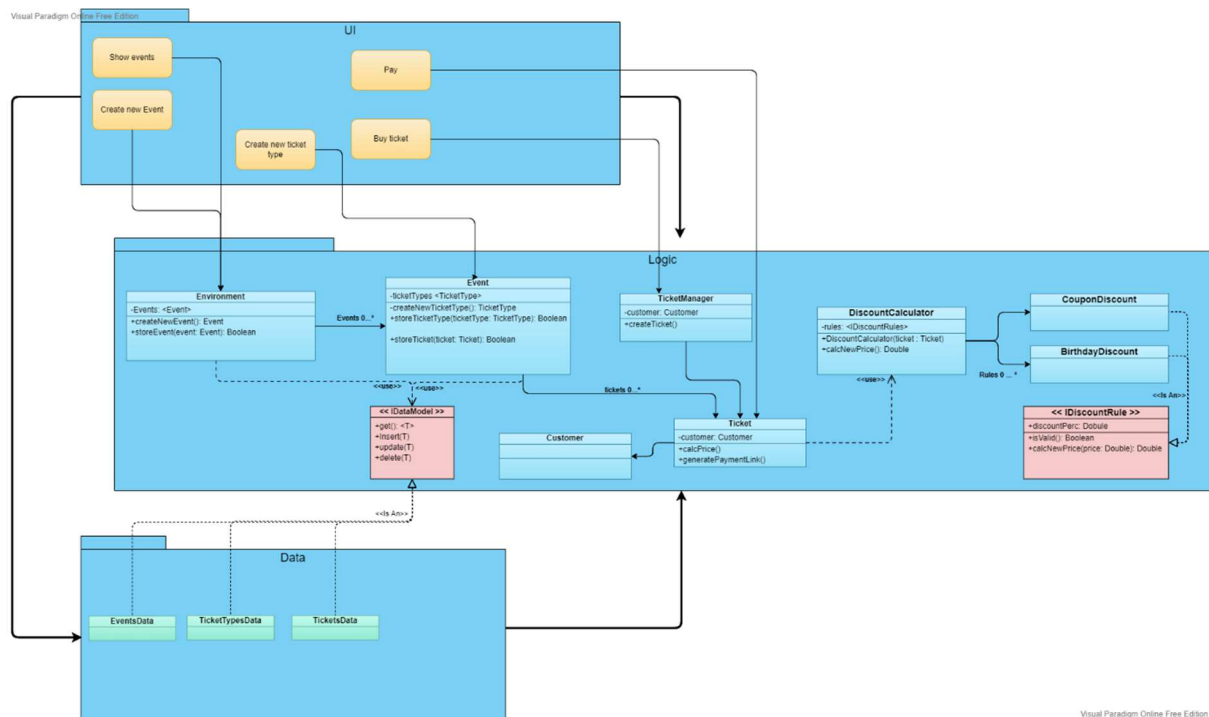
Visual Paradigm Online Free Edition



# Algoritme



# Architectuur



[Source](#)

## 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 meer modulair aangezien de layers binnen de applicatie 'loosely coupled' zijn.