

Functioneel ontwerp

Piotr Tadralla
AP2O-S201
VERSIE: 1.7

14-02-2022

Inhoud

Versiebeheer	2
Inleiding	2
Flow.....	3
Conceptueel model	3
Environment	3
Users	3
Ticket groepen.....	3
Events en Tickets	4
Customers.....	4
Lijst van eisen	5
Use Case Diagram.....	5
Actors en use cases	5
Diagram	6
Algoritme	9
Acceptatietest	10

Versiebeheer

Versie	Aanpassing	Datum
1.1	Paragraaf 'inleiding' toegevoegd	21-02-2022
1.1	Conceptueel model uitgebreid	21-02-2022
1.1	Conceptueel model diagram aangepast	22-02-2022
1.1	Lijst van eisen uitgebreid	22-02-2022
1.1	Context aan use case diagram toegevoegd	22-02-2022
1.1	Use case diagram aangepast	22-02-2022
1.1	Paragraaf 'Algoritme' toegevoegd	22-02-2022
1.1	Acceptatietest toegevoegd	22-02-2022
1.2	Use Case descipriton	10-03-2022
1.3	Conceptueel model enkelvoud	21-03-2022
1.4	Algoritme aangepast	05-04-2022
1.5	Algoritme test scenario toegevoegd	10-04-2022
1.6	Algoritme aangepast	14-04-2022
1.7	Algoritme voorbeelden toegevoegd	28-04-2022

Inleiding

Bedoeling van de applicatie is om bedrijven die evenementen zoals concerten of wedstrijden organiseren een mogelijkheid geven om al hun events op een plek te beheren.

Binnen de applicatie kunnen medewerkers van het bedrijf:

- Nieuwe events kunnen aanmaken.
- Bij elk event verschillende soorten ticket groepen aanmaken (denk maar aan 'normal' of 'VIP').
- Aan elke ticket groep prijs, hiërarchie level en limiet instellen van hoeveel er maximaal verkocht kunnen worden.
- Refunds accorderen of afwijzen.

Flow

Elk bedrijf zal zijn eigen environment hebben waar die meerdere users kunnen toevoegen. Elk user zal een mogelijkheid hebben om events aan te maken en per event zal die meerdere soorten tickets kunnen aanmaken die vervolgens gekocht kunnen worden door klanten.

Conceptueel model

Environment

Bij het opstarten van de applicatie zullen users hun 'domain' moeten invullen voordat ze bij hun environment kunnen inloggen. Dit zorgt ervoor dat er een globale applicatie kan zijn die verdeeld is in environments.

Users

In het environment zal de eigenaar (die zelf ook een user is) users kunnen aanmaken die vervolgens zaken binnen het environment zullen regelen zoals events aanmaken en beheren, ticket groepen aanmaken, tickets beheren of refunds bevestigen.

Ticket groepen

Alle tickets zullen tot een bepaalde groep behoren die aangemaakt zijn door de users. Elke groep zal een hiërarchie level hebben die ervoor zorgt dat mensen die tickets hebben met een hogere

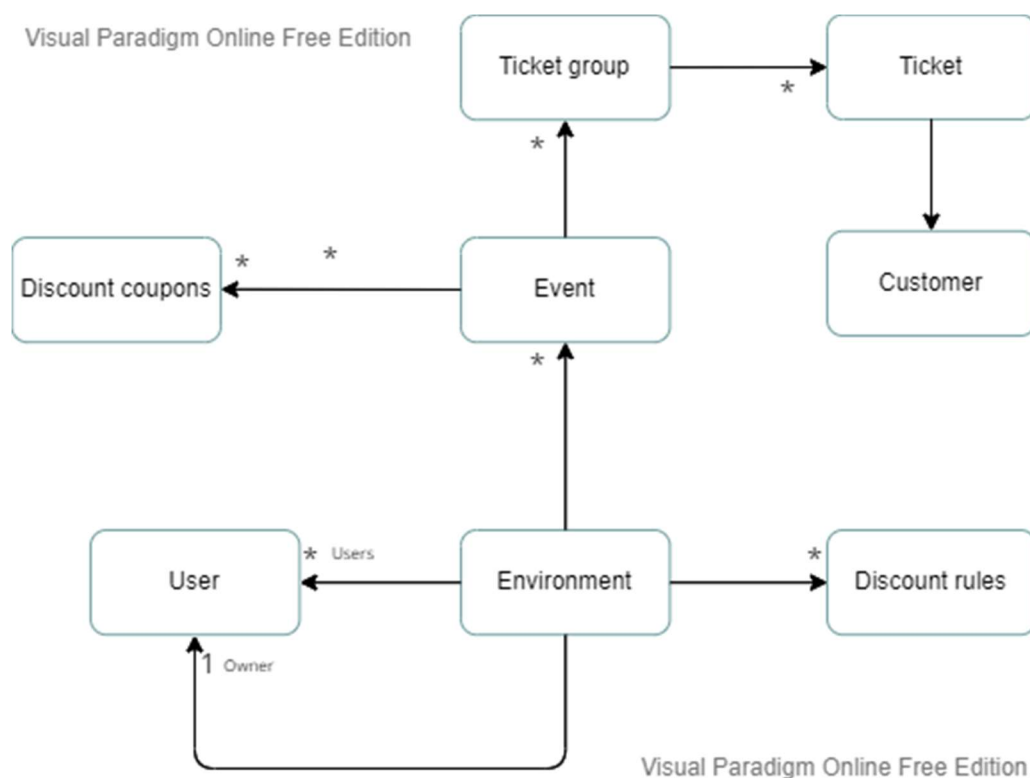
hiërarchie level zich eerder kunnen inchecken en er zal ook meer tijd per persoon gegeven worden waardoor er minder druk zal zijn.

Events en Tickets

Aan elk event zullen er ticket groepen aangemaakt worden. Binnen de groepen zal een hiërarchie systeem plaatsvinden, sommige tickets zullen belangrijker zijn dan andere. Dit concept zal invloed hebben om het tijdstip wanneer een customer zich kan inchecken.

Customers

Customers zullen tickets kunnen kopen via het systeem. Elk ticket zal een uniek token hebben waarmee customers refunds kunnen aanvragen of ticket informatie weergeven zoals tijd, datum locatie en wanneer je zich kan inchecken.



Lijst van eisen

- Elk environment zullen standaard met een user worden geleverd (admin user) die een mogelijkheid zal hebben om nieuwe users aan te maken.
- Users moeten events kunnen aanmaken.
- Users moeten refunds kunnen accorderen of afwijzen.
- Alle events moeten volgende informatie bevatten: begintijd, datum, het tijdstip van inchecken, en locatie.
- Aan elk event moeten er meerdere soorten ticket groepen aangemaakt kunnen worden.
- Bij het aanmaken van een ticket groep moeten users hiërarchie level, prijs en maximale aantal tickets instellen.
- Elk ticket zal een uniek token (id) hebben die door klanten gebruikt kan worden om event info te bekijken.
- Customers moeten een mogelijkheid hebben een refund aan te vragen.

Use Case Diagram

Actors en use cases

- **Customer:** Customer is een klant die via het systeem tickets zal kopen. Customer is gekoppeld aan de volgende use cases:

Buy tickets: kopen van een of meerdere tickets via het systeem.

Show event info: elk ticket zal een uniek token (id) hebben waarmee customers per ticket informatie kunnen weergeven zoals event tijd en datum, locatie of de verwachte inchecktijd.

Refund: Customers zullen ook een mogelijkheid hebben om refund aan te vragen. Deze use case is verder gekoppeld aan users.

- **User:** User is een medewerker die events en tickets in het systeem beheert. User is gekoppeld aan de volgende use cases:

Add event: Toevoegen van een nieuw event.

Delete event: Bestaande events verwijderen. Events zullen niet volledig verwijderd worden, maar op inactief gezet worden waardoor users nog altijd informatie van oude events kunnen ophalen.

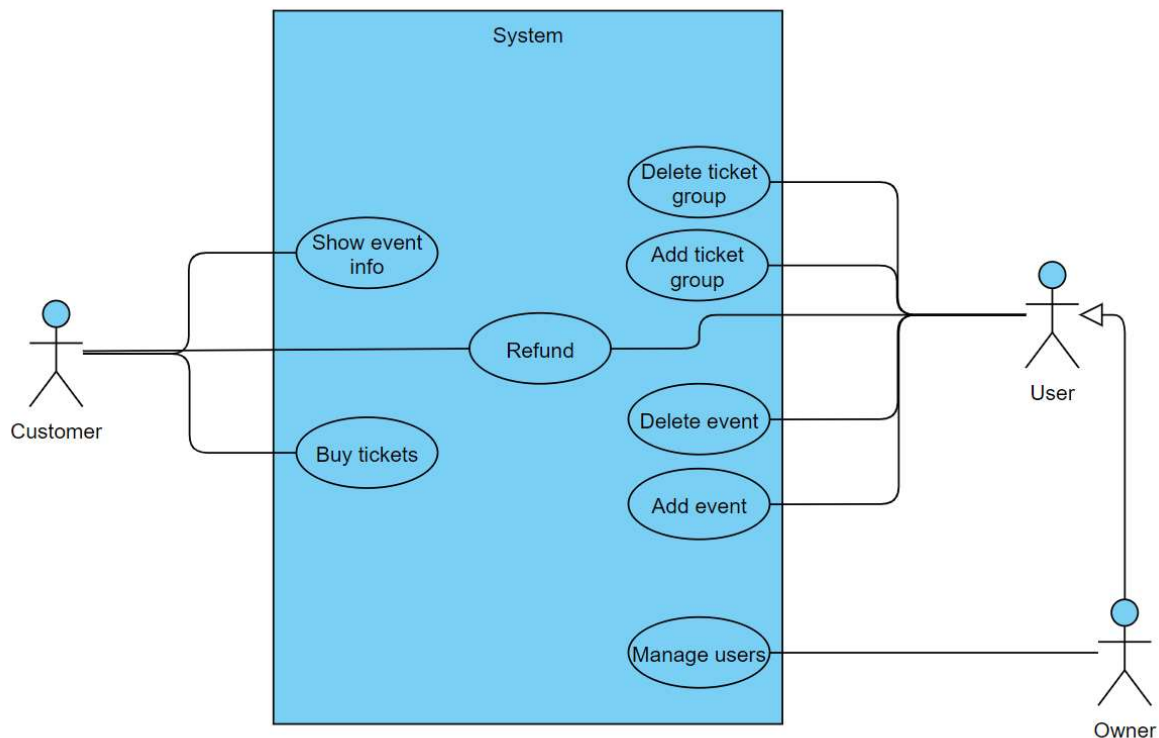
Add ticket group: nieuwe ticket groep toevoegen voor een event. Bijvoorbeeld ticket groep 'Normal' of 'VIP'. Voor elke groep kan een andere prijs of hiërarchie level ingesteld worden.

Delete ticket group: Ticket groepen verwijderen. Hiervoor geldt hetzelfde delete concept als voor events.

Refund: Users zullen customer refunds moeten afhandelen. Wanneer er een refund wordt aangevraagd die niet voldoet aan de voorwaarden zal die door een user afgewezen kunnen worden.

- **Owner:** Actor owner is de eigenaar van het environment en overerft alle use cases die gekoppeld zijn aan de user. Owner heeft ook een extra use case namelijk manage users waarmee hij nieuwe users kan toevoegen of bestaande users verwijderen.

Diagram



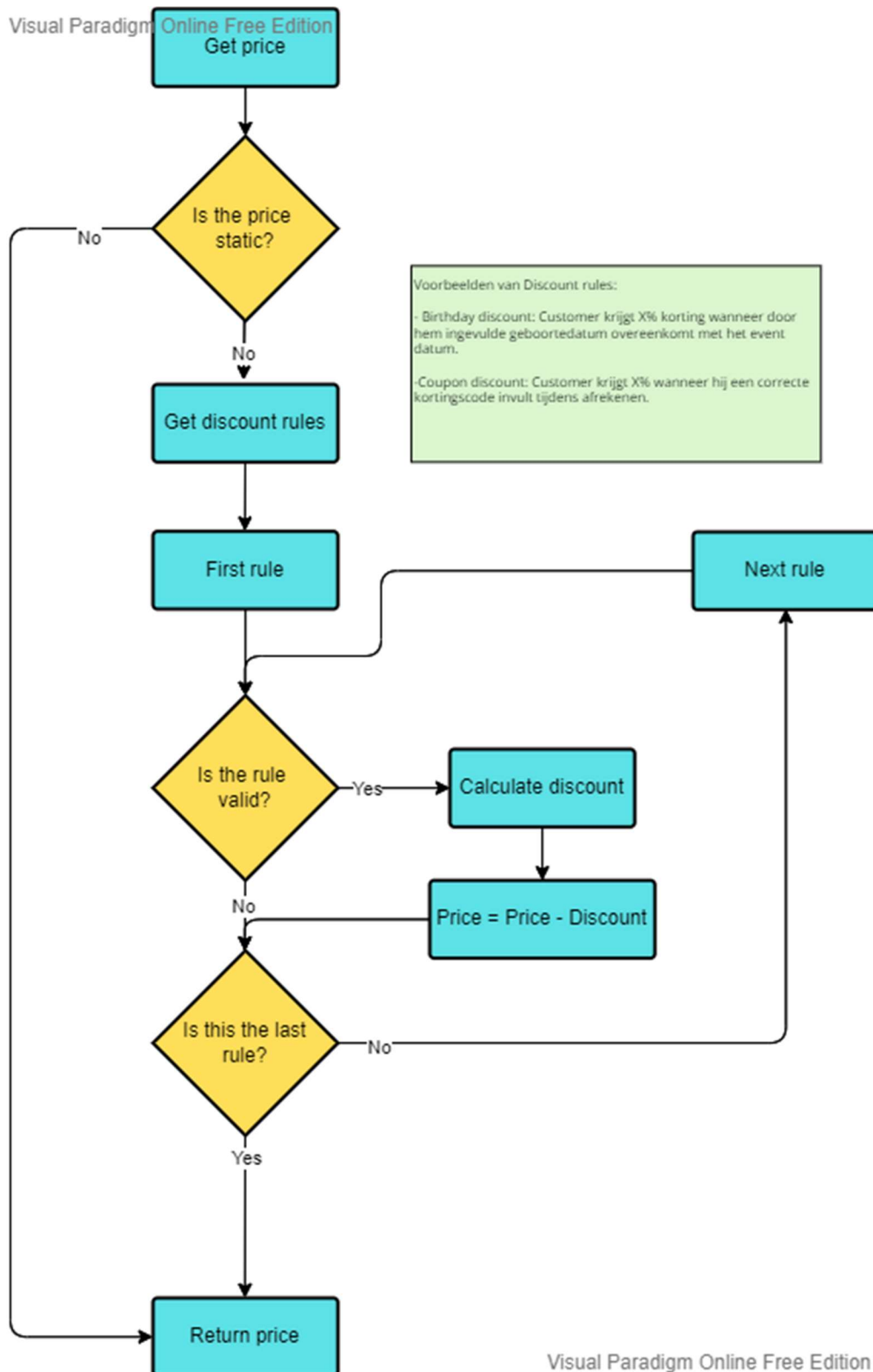
Use Case Name:	Buy tickets	
Scenario:	Customer koopt een ticket of meerdere tickets.	
Description:	Customer selecteert een event en vult alle gegevens in om vervolgens betaling te afronden via een externe payment gateway.	
Actors:	Customer	
Stakeholders:	Customer	
Preconditions:	Customer moet een event en aantal tickets selecteren. Event moet actief zijn. Ticket moeten nog beschikbaar zijn.	
Postconditions:	Aantal beschikbare tickets wordt bijgewerkt. Nieuwe ticket wordt aangemaakt	
Flow of activities:	Actor	System
	1. Customer selecteert een event. 2. Customer selecteert type en aantal tickets. 3. Customer vult alle gegevens in. 4. Customer klikt op afrekenen.	1. Systeem checkt of dat het event nog actief is. 2. Systeem check of dat geselecteerde tickets nog beschikbaar zijn. 3. Systeem valideert klantgegevens 4. Systeem redirect de customer naar een 3rd party payment gateway
Exception condition:	1. Customer selecteert een event die niet meer actief is. 2. Customer kiest voor tickets die niet meer beschikbaar zijn.	

	3. Customer selecteert meer tickets dan dat er beschikbaar zijn. 4. Customer vult onjuiste gegevens in
--	---

Use Case Name:	Show ticket info	
Scenario:	Customer wilt ticket informatie inzien	
Description:	Alle gegenereerde tickets zullen gekoppeld zijn aan een token (id) met behulp of deze token zullen klanten ticket informatie kunnen inzien.	
Actors:	Customer	
Stakeholders:	Customer	
Preconditions:	Customer moet ticket token invoeren in het systeem om event en ticket informatie te weergeven.	
Postconditions:	Customer krijg event en ticket en event informatie te zien en krijgt een mogelijkheid om die te refunden.	
Flow of activities:	Actor	System
	1. Customer opent ticket info pagina. 2. Customer vult de token in	1. Systeem vraagt om ticket token. 2. Systeem laat alle bijhorende gegevens zien
Exception condition:	1. Customer vult onjusite token in.	

Use Case Name:	Refund	
Scenario:	Customer wilt ticket refunden.	
Description:	Customer kan voor een refund kiezen, deze moet bevestigd of afgewezen worden door een user.	
Actors:	Customer User	
Stakeholders:	Customer User	
Preconditions:	Customer open ticket info pagina en kiest voor ticket refunden	
Postconditions:	Refund aanvraag wordt aangemaakt deze moet vervolgens afgehandeld worden door een user	
Flow of activities:	Actor	System
	1. Customer open ticket info pagina en kiest voor refund 2. User moet het aanvraag accepteren of afwijzen.	1. Refund aanvraag wordt aangemaakt. 2. Customer krijgt op de info pagina status van het aanvraag te zien.
Exception condition:	1. Customer probeert een afgelopen event te refunden.	

Algoritme



Acceptatietest

Code	Instructie	Verwacht resultaat	Werkelijk resultaat	Geslaagd
Manage users				
MU_0	1.Vul in alle velden als volgt: Voornaam: 'Piotr' Achternaam: 'Tadralla' Email: 'test@hotmail.com' 2.Klik op 'Bevestigen'	Er wordt een nieuwe user aangemaakt		
MU_1	1.Vul in alle velden als volgt: Voornaam: 'Piotr' Achternaam: 'Tadralla' Email: Leeg 2.Klik op 'Bevestigen'	Systeem meldt dat email niet leeg kan zijn		
MU_2	1.Vul in alle velden als volgt: Voornaam: 'Piotr' Achternaam: 'Tadralla' Email: 'test@hotmail.com' 2.Klik op 'Bevestigen'	Systeem meldt dat er al een user met de ingevulde email bestaat.		
Add Event				
AE_0	1. Vul in alle velden als volgt: Naam: 'Test' Datum: '01-01-2023' Locatie: 'Eindhoven' 2. Klik op 'Bevestigen'	Er wordt een nieuw event aangemaakt.		
AE_1	1. Vul in alle velden als volgt: Naam: 'Test' Datum: Leeg Locatie: 'Eindhoven' 2. Klik op 'Bevestigen'	Systeem meldt dat datum niet leeg kan zijn.		
AE_2	1. Vul in alle velden als volgt: Naam: 'Test' Datum: '01-01-2000' Locatie: 'Eindhoven' 2. Klik op 'Bevestigen'.	Systeem meldt dat er onjuiste datum is ingevuld.		
Buy ticket				
BT_0	1.Selecteer het event 'Test'. 2.Selecteer ticket groep 'Normal'. 3.Vul alle NAW-gegevens in. 4. Klik op 'Betalen'.	Customer wordt geredirect naar een externe payment gateway		
BT_1	1.Selecteer het event 'Test' 2.Selecteer geen ticket groep. 3.Vul alle NAW-gegevens in. 4. Klik op 'Betalen'	Systeem meldt dat de customer het ticket moet selecteren.		
BT_2	1.Selecteer het event 'Test' 2.Selecteer ticket groep 'Normal'.	Systeem meldt dat NAW-gegevens verplicht zijn.		

	3.Vul Geen NAW-gegevens in. 4. Klik op 'Betalen'.			
--	--	--	--	--

Algoritme				
A_0	1.Kies voor een event met een statische prijs. 2.Laat kortingscode leeg.	Systeem retournt prijs van het event zonder aanpassingen.		
A_1	1.Kies voor een event met een statische prijs. 2.Vul een geldige kortingscode in.	Systeem retournt prijs van het event min de korting.		
A_2	1. Kies voor een event met een statische prijs.. 2.Vul een ongeldige kortingscode in.	Systeem meldt dat de ingevulde kortingscode onjuist is.		