TICKET RUIL SYSTEEM

**Table of Contents**

[Inleiding 1](#_Toc431814343)

[Aantal personen: 2 1](#_Toc431814344)

[Vereisten 1](#_Toc431814345)

[Front End 1](#_Toc431814346)

[Back End 1](#_Toc431814347)

[Uitwerking 2](#_Toc431814348)

[Technologien te gebruiken 2](#_Toc431814349)

[Vereisten 2](#_Toc431814350)

# Inleiding

Heel wat organizaties, clubs, conferenties verdelen tickets (al dan niet betalend) om toegang te verlenen. Dit kan een ticket via Eventbrite zijn of een abonnement bij KAA GENT (beste club van België) of de Vlaamse Opera.

Meestal worden die tickets maanden op voorhand aangevraagd, denk maar aan Werchter of Tomorrowland,

Problemen stellen zich wanneer je op het moment van de conferentie, het optreden, de voetbalmatch plots ziek wordt en je niet aanwezig kan zijn. Of misschien wil je gewoon niet langer naar dit optreden gaan? Op dat moment zou het ideaal zijn om via een “legaal” platform je ticket beschikbaar te stellen of om te ruilen voor een ander.



**LET OP**: het doorverkopen van tickets is illegaal. Dit is een oefening op school en niet iets wat je zo maar online mag zetten. Indien dit project levensvatbaar is kan er altijd een demo gegeven worden bij een mogelijke organizator die zelf kan beslissen om dit project te gebruiken.

## Aantal personen: 2

# Vereisten

## Front End

De front van de applicatie bestaat uit verschillende delen.

* Welkom pagina met een summier maar duidelijk overzicht van actueel beschikbare tickets.
* Pagina die de werking van de applicatie beschrijft. Maak dit visueel, zorg dat dit niet een gewone tekstpagina is
* Pagina waar je uw evenement kan opvragen of zoeken op basis van verschillende criteria ( soort, kost, tijd).
* Overzicht van beschikbare tickets
* Overzicht van jouw tickets
* Overzicht van jouw tickets die je geruild hebt.
* Een aantal statistieken voor de administrator.

## Back End

* Alle data wordt opgeslagen en is voorzien van een timestamp.
* Voorzie een onderhandelmogelijkheid waarbij de verruiler in real-time kan communiceren met de persoon die een ticket aanvraagt.
* Een administrator kan na inloggen een aantal statistieken opvragen: welke regio meest actief, welke type ticketten het meest geruild, gemiddelde prijs.

# Uitwerking

## Te gebruiken technologieën

|  |  |
| --- | --- |
| **Front End** 8pt van de 20pt | **Back End**\*7pt van de 20pt |
| * CSS Preprocessor, * taskRunnner, * front -end testing, * LocalStorage, * Maps, * gebruik van framework naar keuze (angular). | * volledig asynchroon in javascript – node.js, * persistentie in een NoSQL database (mongodb), * sockets voor multi-user integratie(socket-io), * autorisatie, authenticatie, * minstens één integratie en één unittest(jade), * deployment op cloud naar keuze(azure, heroku), * gebruik van framework naar keuze (express).   \* Default zaken, zoals gezien tijdens de lessen, staan tussen haakjes maar mogen aangepast worden. |

Noot:

* Wie geen Front End volgt kan gebruik maken van een standaard HTML, Bootstrap of Express lay-out.   
  Eventueel kan een student, die enkel Front End volgt, deze front-end aanmaken als project.
* Wie geen Back End volgt kan alle data stockeren in een lokale JSON file.  
  Eventueel kan een student, die enkel Back End volgt, deze backend aanmaken als project.

## Vereisten

* Programmeer in het Engels
* Javascript volgens de regels van de kunst (jshint als evaluatie tool)
* CSS volgens de regels van de kunst (csslint als evaluatie tool)
* Conventies:
  + Gebruik de juiste css naamgevingen.
  + Gebruik duidelijke code.
* Performantie:
  + minifying,
  + concat
  + performante selectoren,
  + performante javascript,
  + foutcontrole op kritische onderdelen,
  + zoveel mogelijk DRY en abstract (= vermijden van herhalingen)
* Mooie, acceptabele (UI) en gebruiksvriendelijke(UX) userinterface voor zowel mobile of desktop gebruik.
* Gedocumenteerd:
  + Waar nodig, als verduidelijking, in de code
  + Aanvullend document (min. 2 bladzijden) met de nodige paswoorden, online URL’s, werkverdeling, schatting van werkuren, grootste moeilijkheden(2) en successen(2), eventueel architectuur en database schema, eventuele referenties en persoonlijke conclusie
* Project beschikbaar op een source control systeem naar keuze.
* Indien datum: ten laatste afgewerkt in de week vóór het examen. Tussentijdse evaluaties zijn mogelijk.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_