**Documentatie project Steam**

**Groepsnaam: pay2win**

**Groepsdeelnemers: Thijn, Dieu, Daan, Yverdon, Glenn**

**Inleverdatum: 31 januari 2024**

**Klas: 1vo**

**Inhoudsopgave**

[**Inleiding** 3](#_Toc157385031)

[**Verplichte onderdelen** 4](#_Toc157385032)

[**Basis applicatie met data:** 4](#_Toc157385033)

[**Customer Journey Canvas:** 4](#_Toc157385034)

[**Statestiekalgoritme:** 4](#_Toc157385035)

[**Toepassingen hardware:** 4](#_Toc157385036)

# **Inleiding**

Project Steam dient is een applicatie die dient als een ondersteuning bij de data van Steam. Steam is een game platform waar je zowel games kan kopen als verbinding kan zoeken bij andere spelers en dan ook op de community pagina kan posten van het assortiment games dat Steam aanbiedt.

De toegevoegde waarde van onze applicatie is dat je al snel inzage hebt op verschillende aspecten van Steam. De nadruk ligt bij de data van de speler, dit doe je door in te loggen met je Steam account. Je kan dan bijvoorbeeld zien welke games populair zijn bij jouw vrienden, u zult later in het bestand meer lezen over de functionaliteit van de applicatie.

Het document is opgedeeld in drie onderdelen: Verplichte onderdelen, Aanvullende onderdelen en extra onderdelen. Bij deze onderdelen zijn weer verschillende functionaliteiten en deze zal ik ook toelichten bij het desbetreffende kopje.

# **Verplichte onderdelen**

## **Basis applicatie met data:**

De basis applicatie is gemaakt doormiddel van Flask. Dit is een framework waarmee je html kan runnen met Python. De data wordt binnen gehaald doormiddel van requests naar de Steam API. Dit komt terug in een Json formaat en dit wordt gebruikt om de data weer te geven.

## **Customer Journey Canvas:**

Wij hebben dit voor twee momenten gedaan de Ist(het heden), dit is het moment voordat onze applicatie in is gezet en de Soll(de toekomst), dit is het moment nadat onze applicatie in is gezet. Je kan de verbetering zien in de Soll situatie door middel van de verbeterde gedachten van de persona’s. Wij hebben verschillende mensen geïnterviewd en daar mee twee persona’s gecreëerd.

## **Statestiekalgoritme:**

Voor de kwantitatieve variabel hebben wij de speeltijd van de vrienden gebruikt om de meest populaire games onder je vrienden over de laatste twee weken te bepalen. Dit werd gedaan met behulp van het merge sort algoritme, dit is tevens ook een aanvullend onderwerp, dus hier over meer bij het volgende onderdeel. De data komt binnen via de getPlayerGames API

Voor de voorspellende statistieken werd er voorspeld welke games populair gaan worden en welke games er gaan dalen in populariteit. Dit is gedaan met lineaire regressie en wordt gevisualiseerd door middel van een grafiek. Ook de grafiek is een aanvullend onderdeel dus dat wordt later besproken.

## **Toepassingen hardware:**

De gekozen toepassing is het laten zien van personastate van jouw vrienden op de neopixel. Als het lampje bijvoorbeeld groen is, is de persoon online en als het rood is, is de persoon offline. Het ophalen van de gegevens gebeurt via de getplayersummaries api.