Projectrapportage  
  
GROEPSPROJECT STEAM  
PERIODE B   
  
Aimane Chakiri - 1746339

Anass Arazouk - 1861846

Amber Linge - 1862245

Barrelt Wittenberg - 1798313

Kylian van den Beukel - 1801756

Max Boon - 1862048



INHOUDSOPGAVE  
Inleiding  
Doelstelling  
Ontwerp en proces  
 BIM  
 AI  
 CCS  
 SD  
 TI  
Groepsdynamiek, samenwerking en het proces  
Eindresultaat  
Reflectie  
Referenties  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
INLEIDING  
Dit rapport is een documentatie van het project Steam.1, 2, 3

De deelnemers van dit projectgroepje, zoals te zien is op de voorpagina, zijn: Aimane Chakiri, Anass Arazouk, Amber Linge, Barrelt Wittenberg, Kylian van den Beukel en Max Boon.

Het groepsproject is onderdeel van de opleiding HBO-ICT aan de Hogeschool Utrecht, specifiek in de propedeuse periode B.

De casus en haar eisen zijn terug te vinden in de referenties. Eerst is er een korte samenvatting van de doelstelling te lezen, gevolgd door een behandeling van het proces en het ontwerp per onderdeel. Dit komt allemaal weer samen wanneer het eindresultaat wordt besproken. Ten slotte is er een reflectie op de afgelopen weken en natuurlijk een lijst met referenties.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
DOELSTELLING  
Voor het project moeten er meerdere doelstellingen behaald worden binnen verschillende ICT-disciplines.

Steam fungeert als een sociaal platform voor haar gebruikers en streeft ernaar haar klanten beter te bedienen met grafische weergaves en hardwarecomponenten. Een voorbeeld hiervan is bijvoorbeeld 'welke spellen worden het meest gespeeld?'

Het kloppende hart van onze oplossing moet een basisapplicatie zijn waar gebruikers mee kunnen interacteren en data kunnen aflezen. Als kers op de taart is er een koppeling met hardwarecomponenten.

Naast al het technologische worden er ook BIM-taken uitgevoerd, zoals het maken van een customer journey canvas.

Er zijn verplichte onderdelen die als volgt zijn:

Basisapplicatie met data

Customer Journey Canvas

Statistiekalgoritmen

Toepassing hardware

Naast de verplichte onderdelen zijn er ook aanvullende onderdelen, zoals het maken van een stakeholderanalyse, het gebruik van Steam API’s en AI sorteermogelijkheden.

Naast de aanvullende onderdelen zijn ook extra onderdelen toegestaan; deze ideeën moeten van tevoren worden besproken met de docenten.

Voor dit project worden punten toegekend voor het geven van feedback aan groepsleden, het bijhouden van schriftelijke communicatie en de capaciteit tot samenwerking van groepsleden.  
  
ONTWERP EN PROCES  
- BIM  
De verantwoordelijkheid voor BIM lag bij Anass en Amber. Zij hebben een stakeholderanalyse, BMC, Value Proposition Canvas en Customer Journey Canvas gemaakt. Helaas is de Customer Journey Canvas niet volledig af. Het BIM-onderdeel is pas laat in het project gemaakt, waardoor de BIM-onderdelen met terugwerkende kracht aan het eindresultaat zijn gekoppeld. Voor het BIM-onderdeel is er veel gewerkt met visuele weergaves. Zo zijn de BMC en Value Proposition Canvas één afbeelding.

De BIM-onderdelen zijn allemaal terug te vinden in de bijlagen.4, 5, 6, 7   
Voor de stakeholderanalyse heeft Aimane aantekeningen gemaakt tijdens een van de BIM-lessen, en Amber heeft deze omgezet in een verslag. De BMC en de Customer Journey Canvas zijn ook door Amber gemaakt, terwijl Anass de Value Proposition Canvas heeft gemaakt.

Het oorspronkelijke idee was dat het BIM-onderdeel een sturende rol zou hebben in het project, maar helaas waren Anass en Amber niet actief in hun communicatie. Hierdoor hebben we pas aan het einde van het project de BIM-onderdelen ontvangen, omdat deze nog lang niet voltooid waren.

Hierdoor is er weinig te vertellen en te schrijven over BIM in de documentatie, omdat er helaas nu geen tijd meer voor is. Anass heeft ook geprobeerd een dashboard te ontwerpen, maar het concept dat hij had gemaakt kon helaas niet worden uitgevoerd vanwege complicaties met Streamlit. Daarom verwijzen wij jullie naar de bijlagen voor meer informatie.4, 5 ,6 ,7, 21  
  
- AI  
Voor het onderdeel AI waren Kylian, Amber en Anass verantwoordelijk gesteld.

Met betrekking tot AI moesten data worden gezocht en gesorteerd, en deze data moest vervolgens worden weergegeven in grafisch leesbare vormen.

Op GitHub, in de branch 'Artificial\_Intelligence', staan Python-bestanden met de code met betrekking tot het AI-onderdeel.8 In deze gehele documentatie zal de GitHub-repo vaak worden genoemd.

Er is code geschreven om zowel kwalitatieve als kwantitatieve data op het dashboard te tonen, evenals code voor lineaire regressie en code om data te sorteren. De algoritmes zijn buiten het dashboard bedacht en geschreven, zoals te zien is op de repo. Vervolgens zijn de algoritmes, nadat ze in hun eigen Python-bestanden leken te functioneren, geïmplementeerd in het dashboard. Aimane en Kylian hebben de code aangepast, waarbij placeholders zijn vervangen door variabelen uit de code en er is gewerkt aan visuele vormgeving.

De geschreven algoritmes kunnen het volgende: er is een algoritme geschreven om lineaire regressie weer te geven op onze basisapplicatie. Hiervoor gebruiken we op de X-as de gemiddelde speeltijd van games en op de Y-as de positieve beoordelingen van een game. Ook kunnen gebruikers van onze basisapplicatie de gemiddelde prijzen van gekozen genres en categorieën bekijken via een ander geschreven algoritme. Er is ook een algoritme geschreven om op basis van een gekozen genre of categorie en datapunt zoals prijs of aantal positieve beoordelingen te sorteren. Vervolgens kan de gebruiker vragen om de top 10, 25, enz. te zien van het gekozen datapunt.

Voor AI heeft Kylian bepaalde bronnen gebruikt en ook ChatGPT.9, 10, 11  
  
- CCS  
Voor CCS hebben Max en Barrelt de volledige verantwoordelijkheid genomen.

Er is gebruik gemaakt van Azure om een kleine database op te zetten. Deze database fungeert als een 'placeholder' om een demo te demonstreren.

In de database bevindt zich gebruikersinformatie. Op het dashboard kunnen Steam-gebruikers een gebruikersnaam opgeven. Vervolgens zoekt het dashboard in de database naar een SteamID dat is gekoppeld aan de opgegeven gebruikersnaam. Wanneer het dashboard in de database een SteamID heeft gevonden, worden met behulp van verschillende API's diverse gegevens over de gebruiker, wiens gebruikersnaam is ingevoerd, opgevraagd en weergegeven op het dashboard.  
  
- SD  
De verantwoordelijkheden voor SD lagen bij Aimane, Barrelt, Kylian en Max. Vier van de zes groepsleden hebben zich beziggehouden met SD vanwege de noodzaak om andere onderdelen aan de basisapplicatie te koppelen. Aimane heeft de architectuur van het dashboard opgezet, terwijl Barrelt, Kylian en Max hun onderdelen hebben gekoppeld aan het dashboard. Voor het dashboard is 'Streamlit' gebruikt. In eerste instantie overwogen we Django te gebruiken, maar na ongeveer een week hebben we als groep ingezien dat dit misschien te lastig zou zijn. Streamlit bleek een tamelijk gebruiksvriendelijke module te zijn die ontwikkelaars veel voorgekauwde functionaliteiten biedt, zoals auto-fill functies voor zoekbalken. Het dashboard heeft meerdere interactieve pagina's waar alle andere onderdelen, op BIM na, samenkomen. Hoewel er hier en daar problemen waren bij het koppelen van alle onderdelen, hebben we deze als groep gezamenlijk opgelost. In de code van het complete dashboard staat extra noodzakelijke documentatie. Zoals eerder vermeld, heeft Aimane aan de architectuur gewerkt op een manier zodat alle andere onderdelen in de code konden worden gekoppeld. In eerdere versies van de code op GitHub is te zien dat alleen placeholders hoefden te worden aangepast. Op momenten dat andere onderdelen gereed waren voor gebruik, werden ze door Aimane en de verantwoordelijke van dat onderdeel (Barrelt, Max of Kylian) aan de code toegevoegd. Ook hier stuitte Aimane soms op problemen waarvoor hij gebruik heeft gemaakt van de Streamlit-documentatie of ChatGPT.12, 13, 14, 15  
  
- TI  
Voor TI waren Aimane en Barrelt volgens het contract verantwoordelijk. Uiteindelijk hebben vooral Barrelt en Max aan TI gewerkt, maar heeft Aimane hier en daar kunnen helpen en zich vooral beziggehouden met het koppelen van de hardwarecomponenten aan het dashboard.

Voor TI hadden we oorspronkelijk ambitieuze plannen, maar uiteindelijk hebben we gekozen voor het koppelen van een LCD-scherm en een RFID-lezer. De RFID-lezer kan voorwerpen, die gebruik maken van NFC, uitlezen, en dit is mogelijk gemaakt door Max. In de database is elke gebruiker gekoppeld aan een nummer dat op de NFC-voorwerpen staat.

We maken ook gebruik van een LCD-scherm waarop wordt gevraagd om een kaart tegen de RFID te houden. Wanneer een NFC-voorwerp wordt gelezen door de RFID, toont het LCD-scherm een welkomstbericht als het voorwerp gekoppeld is aan een gebruikersnaam in de database. Als het een NFC-voorwerp is dat niet is gekoppeld aan een gebruiker, staat er op het LCD-scherm dat het NFC-voorwerp niet wordt herkend.

Vervolgens wordt de gebruiker gevraagd om hun SteamID op te geven, en worden via API’s verschillende gegevens opgevraagd en getoond op het dashboard. We hebben ook hard gewerkt aan het toevoegen van de NeoPixel als hardwarekoppeling, maar dit wilde niet functioneren vanwege complicaties met Streamlit.

Ook voor het onderdeel TI hebben we een video bron geraadpleegd en ChatGPT gebruikt. 16, 17  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
GROEPSDYNAMIEK, SAMENWERKING EN HET PROCES  
Het project is gestart met het opstellen van een contract waarin alle groepsleden hun taken en afspraken hebben vastgelegd. Dit contract, te vinden onder bijlage 18, benoemt specifieke rollen, zoals Max als scrum master en Aimane als notulist. Consequenties voor het niet nakomen van taken of afspraken zijn eveneens in het contract vastgelegd. Daarnaast wordt gedetailleerd beschreven hoe we samenwerken, met tools zoals Trello, Discord en WhatsApp.

Ons Discord-platform omvat diverse kanalen, elk met een specifiek doel, zoals het bijhouden van individuele taken, het delen van informatie en voice channels voor online gesprekken. In bijlage 19 zijn enkele screenshots opgenomen om een visuele indruk te geven van dit platform. Max, onze scrum master, heeft ook een logboek bijgehouden, dat terug te vinden is in bijlage 20.

Vanaf het begin waren er ambitieuze plannen en ideeën voor elk projectonderdeel, zoals schetsen van het visuele ontwerp van het dashboard en plannen om gebruikers via hun Steam-account in te laten loggen met behulp van een third-party inlogknop, evenals het overwegen van het gebruik van Django en HTML voor het dashboard. Al snel werd duidelijk dat veel van deze plannen niet haalbaar waren, maar gelukkig reageerde de groep flexibel en paste zich aan.

De communicatie tussen de groepsleden verliep over het algemeen soepel, met zowel fysiek contact op school als online interacties. Er werd snel en duidelijk gereageerd op problemen of vragen. Door gebruik te maken van Trello-agenda's, Discord-berichten en regelmatig WhatsApp-contact was iedereen goed op de hoogte van elkaars werk. Over het algemeen was dit een effectieve manier van samenwerken, waarbij de groep proactief communiceerde en verantwoordelijkheden werden nagekomen, hoewel dit helaas niet altijd het geval was.  
  
De communicatie en het proces verliepen minder soepel met betrekking tot het BIM-gedeelte; er was veel onduidelijkheid over wat wel of niet was voltooid. Zo kwam Amber pas in de laatste week te weten dat er interviews moesten worden afgenomen voor de Customer Journey Canvas, een verplicht onderdeel.

Bij het schrijven van deze documentatie op 29 januari 2024 was er voor BIM alleen een BMC ingeleverd door Amber, die ook beweert een stakeholderanalyse te hebben gemaakt maar deze nog niet heeft opgestuurd. Er is weinig informatie beschikbaar over de Customer Journey Canvas, ondanks herhaalde verzoeken, zowel fysiek als via bijvoorbeeld WhatsApp, van andere groepsleden. We hebben actief geprobeerd deze informatie te verkrijgen voor onze documentatie en presentatie, maar helaas hebben we weinig concrete informatie.

Op 30 januari 2024 heeft Anass een Value Proposition Canvas van Steam verstuurd, en Amber heeft een stakeholderanalyse gestuurd. Op 31 januari heeft Amber de Customer Journey Canvas half afgemaakt opgestuurd.

Het verloop van BIM en de communicatie met onze BIM-groepsgenoten verliepen helaas minder goed. Ook tijdens het schrijven van de documentatie ging het weer minder soepel, mede door materiaal dat veel te laat werd geleverd, wat resulteerde in een documentatie die met veel moeite, haast en stress is geschreven.

Daarnaast hebben we op de poster, waar groepen hun voortgang konden aangeven per onderdeel, bijvoorbeeld de Journey Customer Canvas volledig ingevuld omdat een van de twee ons had verteld dat deze bijna klaar was. Dit bleek echter niet het geval te zijn.

Dit is zeer teleurstellend, aangezien alle andere onderdelen goed op schema liepen. We waren bijna volledig klaar met alle andere aspecten van het project en hadden veel aanvullende onderdelen gemaakt. De laatste week was zeer stressvol en moeizaam vanwege de noodzaak om snel het BIM-gedeelte te voltooien, wat ook invloed had op het schrijven van onze documentatie en de algehele projectuitvoering. Er was duidelijk sprake van een tweedeling binnen ons projectgroepje.  
  
  
  
  
  
  
  
  
   
  
EINDRESULTAAT  
Het eindresultaat is een prachtig dashboard dat er stilistisch eenvoudig en strak uitziet. Bij het openen van de webpagina verschijnt er een hoofdpagina met een kort welkomstbericht.Afbeelding met tekst, schermopname, Lettertype, grafische vormgeving

Automatisch gegenereerde beschrijving  
  
Er zijn tal van verschillende pagina's die toegankelijk zijn via een zijbalk.  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving  
  
  
  
  
  
  
Voor het visuele aspect is er ook veel te ontdekken. Bijvoorbeeld, op de pagina 'Search for game information' kun je aan de hand van afbeeldingen zien op welke platforms een game beschikbaar is. Daarnaast is er een grafiek beschikbaar waarin je de beoordelingen van een game kunt bekijken.  
Afbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijving  
  
  
  
  
  
  
  
Interactief is de webpage ook, je kunt van alles zelf zoeken. Zo kun je bijvoorbeeld twee zelf uitgekozen games met elkaar vergelijken en indien je wat dieper wilt vergelijken kun je een datatype uit het JSON bestand selecteren om daar de verschillen van te zien.  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving   
Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Bovendien is er diverse data beschikbaar en zichtbaar via verschillende algoritmes. Sommige van deze data zijn interactief, terwijl andere statisch zijn. Hieronder staat een voorbeeld van een statisch staafdiagram dat te zien is op ons dashboard.  
Afbeelding met tekst, schermopname, scherm, software

Automatisch gegenereerde beschrijving  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Op die zelfde pagina is er ook interactieve informatie te verkrijgen.  
Hier volgt een voorbeeld daarvan.  
Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving  
  
Via de TI-koppeling kan een gebruiker van het dashboard eenmalig hun gebruikersinformatie van Steam inzien. Helaas is deze TI-widget slechts één keer te gebruiken en kan pas opnieuw worden gebruikt wanneer de pagina volledig opnieuw wordt opgestart. Dit komt door complicaties tussen Streamlit en de Pico.

Tevens is het mogelijk om via een gebruikersnaam informatie op te halen over iemands Steam-gebruikersaccount. Dit is tot stand gebracht door het opzetten van een demodatabase die eruitziet zoals te zien is op het onderstaande plaatje.Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving  
  
  
  
  
  
  
  
  
REFLECTIE  
Hier volgen de reflecties van onze groepsleden.  
  
De reflectie van Kylian:  
Wat goed verliep tijdens deze periode was dat we een duidelijke rolverdeling hadden binnen de groep. Iedereen was goed op de hoogte van zijn of haar taken. Als ik naar mijn eigen bijdrage kijk, ben ik in het begin actief geweest in de projectwerkzaamheden. Later merkte ik dat mijn inzet wat verminderde, met name rond de vakantie, maar daarna heb ik het weer opgepakt. Uiteindelijk heb ik mijn oorspronkelijke doelen voor het project bereikt.

Een leerpunt voor de volgende keer is om beter te plannen, zodat ik niet op het laatste moment meer taken moet afronden dan oorspronkelijk gepland was in de laatste week. In ons groepje ondervonden we zelf geen hinder als bepaalde taken nog niet waren afgerond, omdat we van elkaar wisten dat het tegen de deadline klaar zou zijn. Ik begrijp echter dat in andere groepen of in een werkomgeving het vervelend kan zijn als mensen taken op het laatste moment afronden.  
  
De reflectie van Max:  
Ik vond het project over het algemeen prima verlopen, behalve in de laatste weken door het BIM-gedeelte. Volgens mij verliep alles volgens plan (behalve BIM), maar ik had zelf meer willen bijdragen aan het project. Het is natuurlijk ook lastig om samen te werken aan één specifiek aspect. Gelukkig heb ik voornamelijk kunnen bijdragen aan TI en CSC. Helaas kreeg ik van onze klasgenoten die aan BIM werkten te horen dat alles zo goed als af was, terwijl dat in werkelijkheid niet het geval was. Daardoor wisten ik en anderen niet wat wel en niet was afgerond.  
  
De reflectie van Barrelt:  
Het project vond ik redelijk, maar er was wat onduidelijkheid. Het verliep allemaal wel prima, vooral in het begin, maar op een gegeven moment liep het met BIM niet zo soepel. Ikzelf heb voornamelijk gewerkt aan TI en wat ondersteunende taken op me genomen. Ik had graag meer willen doen, maar wist niet precies wat. Dit kwam doordat alles aanvankelijk goed leek te verlopen. Later bleken er wat problemen te zijn met BIM. Aangezien ik niet wist wat al wel of niet was gedaan, kon ik er ook niet mee aan de slag.  
  
  
  
  
  
  
  
De reflectie van Aimane:  
Ik ben zeer tevreden met mijn eigen bijdragen aan de groep. Ik heb het gevoel dat ik de sfeer goed weet te behouden en actief contact zoek met mijn groepsgenoten. Dit sociale aspect is natuurlijk van belang, want het is essentieel dat iedereen zich vrij voelt om zich te uiten en zich op hun gemak voelt. Op die manier verloopt de samenwerking makkelijker.

Ook ben ik zeer trots op de architectuur van het dashboard. Wel vind ik het jammer dat Streamlit bepaalde complicaties heeft veroorzaakt. Soms twijfel ik of ik toch Django of Flask had moeten gebruiken voor het dashboard.

Ik denk dat ik de volgende keer meer mijn best moet doen, in samenwerking met andere groepsgenoten, om degenen die wat zwakker zijn te helpen. Helaas kwamen we laat achter dat BIM heel erg achterliep, zoals vaker benoemd in het verslag. Dit had misschien voorkomen kunnen worden als we allemaal iets actiever hadden opgelet.

Ook moet ik letten op mijn werksnelheid; soms ging ik te snel vergeleken met de rest van de groep.  
  
De reflectie van Anass:  
Als onderdeel van deze zelfreflectie richt ik mij op mijn ervaringen en prestaties gedurende het afgelopen Blok B. Mijn doel is om mijn sterke punten en verbeterpunten te identificeren.

Gedurende blok B heb ik samen met mijn groepje aan project Steam gewerkt. We moesten in dit project een toevoeging maken aan het bestaande Steam-platform. Tijdens deze opdracht hebben we geleerd om in samenwerking met anderen kennis en vaardigheden toe te passen in een groepsproject. Ook hebben we onze persoonlijke vaardigheden ontwikkeld, zoals samenwerken, communiceren, plannen, presenteren, initiatief nemen en verantwoordelijkheid dragen.

Aan het begin van blok B heb ik voor mezelf doelen gesteld, zoals het op tijd afmaken van mijn taken en veel tijd stoppen in het project, zodat ik mijn groepsgenoten niet in de weg zat. Ik verwachtte dat dit zou lukken. Deze doelen dienden als leidraad voor mijn inzet en focus gedurende Blok B.

Mijn prestaties in dit blok waren een mix van veel succesvolle en zware momenten. Ik ben trots op het feit dat ik mijn taken heb afgerond, maar ik erken ook dat er moeilijkheden waren tijdens de opdrachten, vooral doordat ik veel dagen/weken ziek ben geweest. Dit had effect op de hoeveelheid werk die ik kon doen tijdens dit blok. Deze reflectie helpt mij een goed beeld te krijgen van mijn vooruitgang.

Om mijn prestaties verder te verbeteren, ben ik van plan eerder te beginnen met het maken van mijn opdrachten, ondanks dat er genoeg tijd zou moeten zijn om de taken te voltooien. Hierdoor houd ik rekening met eventuele afwezigheid door ziekte of andere problemen.

In vergelijking met het vorige blok zie ik een duidelijke groei in het op tijd maken van mijn werk. Door samen te werken word ik gemotiveerd om mijn werk goed en snel te maken. Deze groei motiveert mij ook om persoonlijke zaken voortaan goed en op tijd te regelen.

Deze zelfreflectie heeft mij geholpen mijn studiereis te evalueren en te begrijpen. Ik ben vastbesloten mijn sterke punten mee te nemen naar volgende projecten en mijn zwakke punten te verbeteren. Ik kijk uit naar toekomstige samenwerkingen en nieuwe kansen om door te groeien als student.  
  
  
  
  
  
  
  
  
  
  
  
REFERENTIES EN BIJLAGEN  
***1:*** Opzet en inhoud Project Steam. Hogeschool Utrecht, van <https://canvas.hu.nl/courses/39933/pages/opzet-en-inhoud-project-steam>  
  
***2***: Steam: Casus. Hogeschool Utrecht, van <https://canvas.hu.nl/courses/39933/pages/steam-casus>  
  
***3:*** Steam: Functionaliteiten en deliverables. Hogeschool Utrecht, van <https://canvas.hu.nl/courses/39933/pages/steam-functionaliteiten-en-deliverables>  
  
***4:*** Zie bijlagen: Stakeholderanalyse  
  
***5:*** Zie bijlagen: Value Proposition Canvas  
  
***6:*** Zie bijlagen: Customer Journey Canvas  
  
***7:*** Zie bijlagen: Business Model Canvas  
  
***8:*** GitHub: De repo van het groepsproject: <https://github.com/Maxi-Cosi/project-steam>  
  
***9:*** YouTube gebruiker NeuralNine, van <https://www.youtube.com/watch?v=VmbA0pi2cRQ&ab_channel=NeuralNine>  
  
***10:*** Geeksforgeeks, van <https://www.geeksforgeeks.org/quick-sort/>  
  
***11:*** Bron: ChatGPT   
Doel: De geschreven algoritme code zocht alleen naar games met 1 tag, games met meer dan 1 tag werden niet geselecteerd door het algoritme.  
Bijdragen: Na het raadplegen van ChatGPT werden games met meerdere tags ook geselecteerd.  
  
PROMPT: het probleem met mijn code is dat hij alleen games pakt met maar 1 tag bv Multi-player. maar hoe pak je ook de games met meerdere categorien? dit is mijn code:  
[ CODE GEDEELD MET CHATGPT ]  
  
ANTWOORD: Ah, ik begrijp het. Als een game meerdere categorieën heeft en je wilt games vinden die overeenkomen met een specifieke categorie, moet je controleren of de opgegeven categorie aanwezig is in de lijst met categorieën van een game.  
Hier is een aangepaste versie van de code om te zoeken naar games met de opgegeven categorie, zelfs als ze meerdere categorieën hebben:  
[ CODE DIE CHATGPT HEEFT GEDEELD MET KYLIAN, DEZE WAS NOG NIET GOED GENOEG ]  
Hier gebruik ik de in operator om te controleren of de opgegeven categorie aanwezig is in de lijst met genres of categorieën van een game. Op deze manier worden ook games geselecteerd die meerdere categorieën hebben. Probeer deze aanpassing en kijk of het nu werkt zoals gewenst.  
  
PROMPT: hij is nu alleen niet gesorteerd, want sommige positive rating staan hoger dan anderen terwijl ze er minder hebben  
  
ANTWOORD: Mijn excuses voor de verwarring. Het lijkt erop dat ik een fout heb gemaakt in de sorteerfunctie. Laten we dat corrigeren. Hier is de bijgewerkte code:

import json

def quicksort(arr, key, secondary\_key):  
 if len(arr) <= 1:  
 return arr   
 stack = [(0, len(arr) - 1)]  
 while stack:  
 low, high = stack.pop()  
 pivot\_idx = divide(arr, low, high, key, secondary\_key)   
 if pivot\_idx - 1 > low:  
 stack.append((low, pivot\_idx - 1))   
 if pivot\_idx + 1 < high:  
 stack.append((pivot\_idx + 1, high))  
 return arr  
def divide(arr, low, high, key, secondary\_key):  
 pivot = arr[high][key]  
 i = low – 1  
 for j in range(low, high):  
 if key in arr[j] and pivot in arr[j][key] and arr[j][secondary\_key]  
>= arr[high][secondary\_key]:  
 i += 1  
 arr[i], arr[j] = arr[j], arr[i]  
 arr[i + 1], arr[high] = arr[high], arr[i + 1]  
 return i + 1  
# Gegevens lezen uit json-file  
with open('steam.json', 'r') as file:  
 game\_data = json.load(file)  
# Vraag de gebruiker naar het gewenste genre of category  
search\_key = input("Welk genre of category zoek je? ")  
# Kopie maken zodat de originele lijst niet wordt gewijzigd  
game\_data\_sorted\_genres = quicksort(game\_data.copy(), key="genres", secondary\_key="positive\_ratings")  
game\_data\_sorted\_category = quicksort(game\_data.copy(), key="category", secondary\_key="positive\_ratings")  
# Zoek het gewenste genre of category in de gesorteerde lijst  
result\_genres = [game for game in game\_data\_sorted\_genres if search\_key in game["genres"]]  
result\_category = [game for game in game\_data\_sorted\_category if search\_key in game["category"]]  
# Sorteer de resultaten op positive\_ratings  
result\_genres\_sorted = sorted(result\_genres, key=lambda x: x["positive\_ratings"], reverse=True)  
result\_category\_sorted = sorted(result\_category, key=lambda x: x["positive\_ratings"], reverse=True)  
# Print de resultaten  
print(f"\nGesorteerde gegevens op {search\_key} en positive\_ratings:")  
for game in result\_genres\_sorted[:100]:  
 print(f"Naam: {game['name']}, Genres: {game['genres']}, Positive Ratings: {game['positive\_ratings']}")  
for game in result\_category\_sorted[:100]:  
 print(f"Naam: {game['name']}, Category: {game['category']}, Positive Ratings: {game['positive\_ratings']}")

In deze bijgewerkte code gebruik ik de sorted-functie om de resultaten opnieuw te sorteren op basis van "positive\_ratings" nadat ze zijn gefilterd op genre of categorie. Hierdoor worden de resultaten op positive\_ratings gesorteerd, zoals je hebt gevraagd. Probeer dit uit en kijk of het nu werkt zoals verwacht.  
  
***12:*** Streamlit documentatie, van https://docs.streamlit.io/  
  
***13:*** Bron: ChatGPT  
Doel: Vragen waarom de /n line break niet werkte met Streamlit.  
Bijdragen: ChatGPT heeft uitgelegd dat Streamlit gebruik maakt van HTML tags.  
  
PROMPT: hoi waarom werkt \n niet in streamlit als line break wanneer ik st.markdown gebruik  
  
ANTWOORD: In Streamlit wordt de \n-sequentie niet automatisch geïnterpreteerd als een line break (regelovergang) wanneer je het gebruikt binnen een st.markdown-functie. In plaats daarvan kun je de HTML-tag <br> gebruiken om een line break te creëren.|

Hier is een voorbeeld:

python

import streamlit as st

tekst\_met\_linebreak = "Dit is de eerste regel.<br>Dit is de tweede regel."

st.markdown(tekst\_met\_linebreak, unsafe\_allow\_html=True)

In dit voorbeeld wordt de st.markdown-functie gebruikt om tekst met een HTML-line break-tag (<br>) weer te geven. De parameter unsafe\_allow\_html=True is nodig omdat Streamlit standaard geen HTML-injecties toestaat om de app veilig te houden.

Houd er rekening mee dat het gebruik van unsafe\_allow\_html potentieel gevaarlijk kan zijn als de HTML-code afkomstig is van onbetrouwbare bronnen. Zorg ervoor dat je de bron van de tekst vertrouwt voordat je deze instelling gebruikt.  
  
***14:*** Bron: ChatGPT  
Doel: Vragen om hulp omtrent het opvragen van steamnews api nieuwsartikelen, deze kwamen in 2 vormen één met HTML tags en één met tags die niet HTML waren.  
Bijdragen: ChatGPT heeft mij een stukje code gegeven waarmee ik de nieuwsartikelen kon overslaan als er bepaalde tags in zaten.  
  
PROMPT: Hoi ChatGPT! Ik heb een probleem, ik gebruik een api die nieuwsartikelen toont in 2 verschillende formats 1 met html tags en de andere met tags zoals [h1], [list] en zo  
Hoe kan ik er voor zorgen dat die gedeeltes van de api nieuwsartikelen die ik krijg, in json format, kan overslaan?  
[ CODE MET CHATGPT GEDEELD ]  
  
ANTWOORD: Om specifieke nieuwsartikelen over te slaan die je niet wilt verwerken kun je de volgende code gebruiken.  
Voorbeeld Python:  
tags\_to\_skip = ['[list]', '[\*]', '[/list]', '[h1]', '[/h1]', '[b]', '[u]', '[i]', 'url=', '[previewyoutube=']  
 if any(tag in contents for tag in tags\_to\_skip):  
 continue  
  
Het controleert of ten minste één van de tags in tags\_to\_skip aanwezig is in de contents en slaat de huidige iteratie over als dat het geval is. Dit is een handige manier om specifieke tags te vermijden bij het verwerken van tekstgegevens.  
  
***15:*** Bron: ChatGPT   
Doel: Vragen om hulp omtrent het aanpassen van bepaalde tags naar HTML tags.  
Bijdragen: ChatGPT heeft een functie voor mij geschreven om de tags te veranderen naar HTML tags.  
  
PROMPT: User

Okay dit is wat ik krijg uit mijn code als ik mijn code run

[img]https://clan.cloudflare.steamstatic.com/images//3381077/6eba6fdad2eeb08061c08631ae1ebe7e14099427.png[/img]  
Celebrate your time in CS2 with the all-new 2024 Service Medal, available starting January 1st!  
To get your service medal, earn XP by playing in official game modes and rank up your CS2 Profile. When you reach Global General (Rank 40), you'll have the option to reset your Profile Rank and earn (or upgrade) a 2024 Service Medal. The medal is an inspectable display item you can show off wherever your avatar is shown.  
  
ik wil de [img] en [/img] veranderen in html leesbare code zodat de afbeelding te zien is.  
Hoe kan ik dat fatsoenlijk geregeld hebben in een functie?  
[ CODE GEDEELD MET CHATGPT ]  
  
ANTWOORD: Het lijkt erop dat je de afbeelding wilt weergeven die in het Steam-nieuwsartikel is ingesloten. Momenteel wordt de tekst weergegeven, maar de [img]-tags worden niet correct weergegeven in HTML.

Hier is een mogelijke oplossing:  
import re  
# ...  
# Bovenin je script of boven de loop waar je door de nieuwsitems gaat  
def replace\_img\_tags(contents):  
 img\_tag\_pattern = r'\[img\](.\*?)\[/img\]'  
 img\_replace\_pattern = r'<img src="\1" alt="Steam News Image">'  
 return re.sub(img\_tag\_pattern, img\_replace\_pattern, contents)  
# ...  
for index, item in enumerate(news\_items):  
 # ... (andere code hierboven)  
 contents = replace\_img\_tags(item.get('contents', ""))  
 # ... (andere code hierboven)

Deze functie (replace\_img\_tags) zoekt naar alle [img]-tags in de inhoud van het nieuwsartikel en vervangt deze door HTML <img>-tags. Dit zou moeten helpen om afbeeldingen correct weer te geven in de Steam-nieuwsartikelen. Voeg deze functie toe aan je script en gebruik de gewijzigde contents-variabele in de st.markdown-call.

***16:*** RC522 RFID Reader Module with Raspberry Pi Pico. microcontrollerslab.com, van https://microcontrollerslab.com/raspberry-pi-pico-rfid-rc522-micropython/  
  
***17:*** Bron: ChatGPT  
Doel: Het vragen hoe de serial aan de andere kant van de Pico er uit ziet.  
Bijdragen: ChatGPT heeft uitgelegd en getoond hoe de serial vanaf een andere kant er uit ziet om te communiceren met de Pico.  
  
PROMPT: dit heb ik aan de kant van de pico met serial en ik heb hier een vraag over:

from mfrc522 import MFRC522  
from machine import I2C, Pin  
from pico\_i2c\_lcd import I2cLcd  
import time  
import utime  
#Druppel: 2041618940  
#Card: 3118640173  
reader = MFRC522(spi\_id=0,sck=2,miso=4,mosi=3,cs=1,rst=0)  
print("Bring TAG closer...")  
print("")  
#LCD DINGETJES!  
i2c = I2C(0, sda=Pin(8), scl=Pin(9), freq=400000)  
I2C\_ADDR = i2c.scan()[0]  
lcd = I2cLcd(i2c, I2C\_ADDR, 2, 16)  
def send\_rfid\_data(card\_id):  
 """Verstuur RFID-gegevens via seriële communicatie."""  
 serial = machine.UART(0, baudrate=115200, tx=Pin(0), rx=Pin(1))  
 serial.write(str(card\_id) + '\n') # Stuur de kaart-ID naar de computer   
while True:  
 lcd.move\_to(0, 0)  
 lcd.putstr("Presenteer kaartA.U.B.")  
 reader.init()  
 (stat, tag\_type) = reader.request(reader.REQIDL)  
 if stat == reader.OK:  
 (stat, uid) = reader.SelectTagSN()  
 if stat == reader.OK:  
 card = int.from\_bytes(bytes(uid),"little",False)  
 print(int(card)) and send\_rfid\_data(card)  
 #if stat == reader.OK:  
 # uid = ":".join([str(i) for i in raw\_uid])  
 # print("UID tag: {}".format(uid))  
 if card == 3118640173:  
 lcd.clear()  
 lcd.move\_to(0, 0)  
 lcd.putstr("Welkom:")  
 lcd.move\_to(0, 1)  
 lcd.putstr("snelboertje!")  
 time.sleep(2)  
 lcd.clear()  
 elif card == 2041618940:  
 lcd.clear()  
 lcd.move\_to(0, 0)  
 lcd.putstr("Welkom:")  
 lcd.move\_to(0, 1)  
 lcd.putstr("max-cosi")  
 time.sleep(2)  
 lcd.clear()  
 else:

lcd.clear()  
 lcd.move\_to(0, 0)  
 lcd.putstr("Niet gemachtigd")  
 time.sleep(2)  
 lcd.clear()  
utime.sleep\_ms(500)

ANTWOORD:

Natuurlijk, ik ben er klaar voor om je vragen te beantwoorden. Je kunt je vragen nu stellen.

PROMPT: hoe ziet de andere kant eruit dan op de pc-serial?

ANTWOORD: Aan de ontvangende kant (de computer) kun je een eenvoudig Python-script gebruiken om de seriële gegevens van de Raspberry Pi Pico te lezen. Hier is een eenvoudig voorbeeld van een Python-script dat gebruikmaakt van de PySerial-bibliotheek:

Python  
Copy code  
import serial  
# Vervang 'COMx' door de juiste COM-poort van de USB-naar-seriële converter op je computer  
ser = serial.Serial('COMx', 115200, timeout=1)  
while True:  
 line = ser.readline().decode().strip()  
 if line:  
 print(f"Ontvangen RFID-gegevens: {line}")  
 # Voeg hier verdere verwerkingslogica toe, afhankelijk van je vereisten

Zorg ervoor dat je de juiste COM-poort (bijvoorbeeld 'COM3' op Windows, '/dev/ttyUSB0' op Linux) gebruikt bij het instellen van ser. Je kunt de juiste COM-poort vinden in het apparaatbeheer of door ls /dev/tty\* (Linux) of python -m serial.tools.list\_ports (Windows) uit te voeren.  
Dit script leest continu de seriële poort en drukt de ontvangen RFID-gegevens af. Je kunt de verwerkingslogica aanpassen op basis van wat je met de RFID-gegevens wilt doen op de computer.  
Vergeet niet om de PySerial-bibliotheek te installeren als je dat nog niet hebt gedaan. Dit kan worden gedaan met het commando pip install pyserial in je terminal.  
Laat me weten als je nog verdere vragen hebt of als er iets anders is waarbij ik je kan helpen!  
  
  
  
***18:*** Zie bijlagen: Contract\_Projectgroep  
  
***19:*** Zie bijlagen: Communicatie middelen  
  
***20:*** Zie bijlagen: Sprint Review en Retrospective  
  
***21:*** Zie bijlagen: Concepten dashboard