**Technisch ontwerp ROC ter Aa**

**Project: New TI Twitch Integration**

**Opdrachtgever: Nöcker**

**Projectnummer: 1**

**Auteur: Eddie Beelen, Thomas Godding**

**Datum: 9-12-2019**

**Versie: 1**

Ondergetekenden verklaren zich akkoord met de inhoud van dit technisch ontwerp.

**Opdrachtgever Projectmanager**

***Paraaf gezien: Paraaf gezien:***

Datum: <Geef de datum op.> Datum: <Geef de datum op.>

Plaats: <Geef de plaats op.> Plaats: <Geef de plaats op.>

Inhoudsopgave

[Inleiding 3](#_Toc27135406)

[Technische Eisen 3](#_Toc27135407)

[Componenten en modules 3](#_Toc27135408)

[Diagrammen 3](#_Toc27135409)

[Klassen diagram 3](#_Toc27135410)

[Activiteiten diagram 4](#_Toc27135411)

[Sequentie diagram 4](#_Toc27135412)

[Beslissingen 4](#_Toc27135413)

[Afspraken 4](#_Toc27135414)

[Grenzen 4](#_Toc27135415)

[Plan van Aanpak 4](#_Toc27135416)

[Haalbaarheidsstappen 4](#_Toc27135417)

[Test plan 5](#_Toc27135418)

[Over dit document 5](#_Toc27135419)

[Afkortingen 5](#_Toc27135420)

[Referenties 5](#_Toc27135421)

[Definities 5](#_Toc27135422)

[Gebruikte materialen 5](#_Toc27135423)

[BIJLAGEN 5](#_Toc27135424)

[BIJLAGE B Accordering 7](#_Toc27135425)

# Inleiding

Dit document is gemaakt door New TI Twitch Integration gemaakt door Eddie en Thomas .Wat je van ons gaat verwachten is een uitgebreid document waarin alles staat uitgelegd over hoe, wat en waarom we deze applicatie hebben gemaakt/gaan maken. Wat we ook verwachten is dat u zelf weet wat Twitch inhoud en wat er mee gedaan kan worden

# Technische Eisen

We willen er voor zorgen dat deze applicatie een verduidelijkend overzicht geeft van een bepaalde streamer die de gebruiker zelf kan opzoeken of toevoegen aan zijn/haar favorieten lijst.Deze applicatie willen we op de hoogste snelheid laten werken van hoeveel requests we kunnen verzenden en terug kunnen krijgen van de API. We willen hiervoor ook zo min mogelijk applicatie resources gebruiken en het zo makkelijk mogelijk maken voor de gebruiker. We willen het programma maken met c#. We willen gebruik gaan maken van een ‘drag and drop’ feature om alles aan te kunnen passen voor persoonlijk gebruik.

# Componenten en modules

We zullen 2 hoofdcomponenten hebben. 1 zal de twitch api zijn waar we onze informatie vandaan halen. Ons andere hoofcomponent

* Geef aan uit welke componenten je systeem zal bestaan, hoe die componenten gaan samenwerken.
* Geef aan uit welke modules elke component bestaat wat de benodigde functionaliteit ze hebben en hoe ze met elkaar zijn gekoppeld
* Eventuele afhankelijkheden van externe modules die NIET standaard zijn.

# Diagrammen

## Klassen diagram

* Teken het klassen diagram van alle klassen, attributen en onderlinge relaties, dat bij je applicaties hoort
* Maak een klassen tabel met uitleg over wat een klasse precies doet

## Activiteiten diagram

* Teken het activiteiten diagram (flow diagram) dat bij je applicatie hoort

## Sequentie diagram

* Teken hoe de verschillende klassen moeten samen werken

# Beslissingen

Beschrijf hier kort:

* Waarom het op deze manier opgelost gaat worden

# Afspraken

Beschrijf hier kort:

* Aanleveren materialen of code, door wie en wanneer
* Wie wat doet, en wie iets NIET doet
* We gaan samen aan het werk om de code kloppend te krijgen en hoe we de koppeling gaan leggen tussen de new Twitch API, dit wordt gedaan door middel van samen werken op school aan een programma. Als we nieuwe testjes moeten draaien wordt dit apart gedaan in een ander bestaand en niet in het hoofdbestand gewerkt. Dit wordt later pas geïmplementeerd.

# Grenzen

Beschrijf hier kort:

* Wat er NIET in dit ontwerp zit
* Waarom iets er NIET in zit

# Plan van Aanpak

Beschrijf hier een inleiding van de stappen van de oplossing:

* Zorg ervoor dat de Showstoppers in beeld zijn.
* Zorg ervoor dat beschreven is dat er een haalbaarheidsonderzoek aan ten grondslag ligt, ofwel, zorg ervoor dat de zaken die de totale haalbaarheid in gevaar brengen EERST getest worden
* De New Twitch API kan nog wel eens gaan tegenwerken met onze applicatie. Dit kunnen we door middel van kleine aanpassingen oplossen. Dit zal niet het grootste probleem zijn. Het grootste probleem zal kunnen zijn met de Privacy van elke streamer die wij willen ophalen.

## Haalbaarheidsstappen

* Beschrijf hier in een tabel welke stappen worden genomen in volgorde van afhankelijkheid.
* Voorbeeld: eerst testen of er communicatie is tussen PC en apparaat, dan pas complexe communicatieprotocollen ontwerpen.
* Of we alle benodigde onderdelen wel kunnen gebruiken en hier geen inlog voor nodig te hebben gehad vanuit Twitch

# Test plan

* Geef aan wanneer en hoe je het systeem gaat testen.
* Dit betekent dat je dit in detail moet weergeven met test invoer en verwachte uitvoer.
* Je dient dit voorgaande het coderen van het systeem al aan te geven.
* Aan het einde van de rit dienen de resultaten van het testen in een test rapport te worden vervat en als onderdeel van de documentatie te worden meegeleverd.
* Wij willen het systeem gaan testen door middel van na elke stap opnieuw de applicatie te runnen zodat we precies weten of dat er wel/geen bugs in zitten.

De eerste testen zullen gedaan worden met de API. We zijn heel erg tevreden over hoe de API in elkaar zit, maar wat we niet weten en wel graag willen weten, is of dat de API een snelle terugkoppeling kan geven aan onze applicatie in JSON en die meteen kan uitvoeren op een static page (zonder reload)

We zullen de API steeds opnieuw moeten testen met een user input om te weten te komen of dat de API geen problemen kan krijgen met onze oorspronkelijke code.

Ons test rapport zal hierdoor heel makkelijk

# Over dit document

## Afkortingen

|  |  |
| --- | --- |
| Afkorting | Omschrijving |
| API | Application Programming Interface |
| JSON | Javascript Object Notation |

## Referenties

|  |  |
| --- | --- |
| Referentie | Omschrijving |
|  | Voorbeelden: ‘bovenliggende’ documenten (ProjectPlan, Functioneel Ontwerp, etc.) |
|  |  |

## Definities

|  |  |
| --- | --- |
| Definitie | Omschrijving |
|  | API communicatie moet goed verlopen |
|  | Snelheden met de API kan er ook voor zorgen dat het wat trager verloopt |

## Gebruikte materialen

|  |  |
| --- | --- |
| Onderdeel | Omschrijving |
| Laptop | Eddie en Thomas |
| API | Twitch New API |
| Visual Studio | C# |

# BIJLAGEN

[BIJLAGE B Accordering 6](#_Toc499208804)

BIJLAGE A Protocol van acceptatie

De punten zoals beschreven zijn in de onderstaande tabel opgenomen. Tijdens de oplevering zal op deze punten worden gecontroleerd.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Acceptatiepunt** | **Akkoord** | | **Opmerkingen** |
|  |  | **wel** | **niet** |  |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |
|  | <Geef het acceptatiepunt op.> |  |  | <Geef opmerkingen.> |

**Overige opmerking:**

|  |
| --- |
| <Geef eventueel opmerkingen.> |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gezien namens: |  |  | Paraaf gezien: | Datum gezien: |
| Projectmanager | : | <Geef de eigen bedrijfsnaam op.> |  | <Geef de datum op.> |
| Opdrachtgever | : | <Geef de bedrijfsnaam van de opdrachtgever op.> |  | <Geef de datum op.> |

BIJLAGE B Accordering

Wanneer u akkoord gaat met de inhoud van dit technisch ontwerp, dan verzoeken wij u dit projectdocument ondertekend te retourneren.