**JMP Designdocument**

A black background with white dots

Description automatically generated

**Gemaakt door:**

**Mitchel Meskes**

**Joshua de Bruijn**

**Pedro Eduardo Cardoso**

**Datum: 11 Februari 2024**

**Inhoudsopgave**

- Inhoudsopgave 2

- Belangrijke rollen 3

- Plan van Eisen 4

- Huisstijl 5

- User story’s

- Wireframe

- Mockup

- Database model

- ERD

- UML Diagram

- Sequence Diagram

- Class Diagram

- FLOW Diagram

- Scrum

- Unit Tests

- Beveiliging

- Prestatie overwegingen

- Installatie- en implementatiehandleiding

- Onderhoud en toekomstige ontwikkelingen

**Belangrijke rollen**

**The Product Owners**

Jan Zuur

Ron van Zuilichem

Tom Sievers

**SCRUM Master**

Joshua de Bruijn

**DEV Team**

Pedro Eduardo Cardoso

Joshua de Bruijn

Mitchel Meskes

**Plan van Eisen**

**Belangrijkste Kenmerken**

De applicatie zal gebruikmaken van object georiënteerd programmeren in C# om de code te structureren.

De belangrijkste functionaliteiten van de applicatie omvatten:

Gamemode kiezen, score weergave, winaar weergeven.

**Algemene doelen**

Het bouwen van een gebruiksvriendelijke en Pong Game.

Het testen en valideren van de applicatie om ervoor te zorgen dat deze voldoet aan de gestelde eisen en normen.

**Huisstijl**

**Color pallet/ Font**

De kleurenpallet voor Pong zal uit de originele kleuren van de echte game bestaan oftewel:

**Gebruikte Kleuren:**

Achtergrond:  #000000

Details: #FFFFFF

Font: Default font gebruikt.

Deze combinatie zorgt voor een rustig en netten omgeving voor de applicatie en het roept nostalgie op naar de al bestaande game.

**Buttons**

De knoppen zullen bestaan uit zwart en wit.

Achtergrondknop:  #000000

Detailsknop / text: #FFFFFF

**User story’s**

**Algemene Storys**

De user storys zijn gemaakt met behulp van Azure DevOps/Github Projects.

Hoe wij onze User story’s maken is:

We kiezen een EPIC delen die vervolgens op in een aantal Sprints en in een van die sprints pakken wij vervolgens delen van die EPIC op zodat ieder aan een deel van het project werkt.

**(Zie het volgende voorbeeld)**

**Stap 1**

Kies een EPIC. (Gebruikersregistratie en -authenticatie)

**Stap 2**

Vervolgens delen wij die in delen op en pakt ieder een story.

**Stap 3**

Dan kiest ieder een van de volgende punten en gaat daarmee aan de slag mee.

Iedere EPIC die we behandelen draaien we alle rollen een door zodat iedereen aan alle vlakken heeft gewerkt zowel backend, frontend als databases.

**User story’s**

**Epic 1: Lorum**

**User Story 1.1: Lorum.**

Lorum

**Backlog Item 1.1.1:**

**Wireframe**

**De wireframe bestaat uit de volgende pagina’s:**

**Start Page**, Laat de Startpagina zien met een play button.

A white rectangle with black text

Description automatically generated

**SinglePlayer Page**, Laat de Singleplayer pagina zien met de twee bads en een pingpongbal en de huidige score.

A white background with black text

Description automatically generated

**Wireframe**

**SinglePlayer (na Score) Page**, Laat de Singleplayer pagina zien met de gescoorde punten na het raken van het tegenstanders goal.

A white background with black text

Description automatically generated

**SinglePlayer (na Win) Page**, Laat de Singleplayer pagina zien met de gescoorde punten en het win scherm met de speler die gewonnen heeft.

A white rectangle with black text

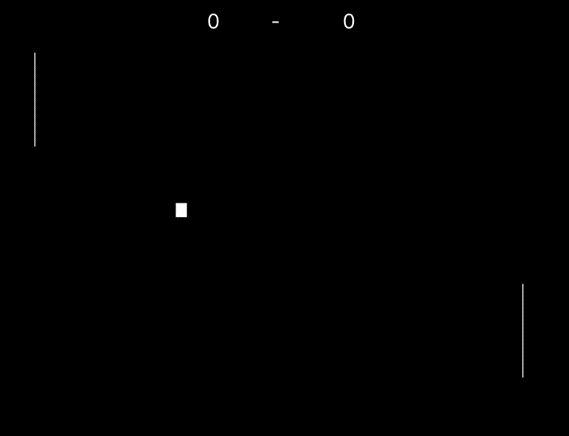
Description automatically generated

**Mockup**

**Start Page**



**SinglePlayer Page**



**Mockup**

**SinglePlayer (na Score) Page**

A screen shot of a video game

Description automatically generated

**SinglePlayer (na Win) Page**

A screenshot of a video game

Description automatically generated

**Database model**

**ERD**

**UML Diagram**

**Sequence Diagram**

**Class Diagram**

**Flow Diagram**

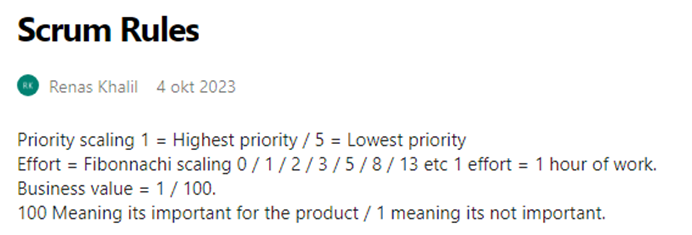
**Scrum**

**Scrum Opzet**

Wij hebben de hele planning uitgewerkt met Azure DevOps/Github Projects zoals te zien is in de volgende kopjes.

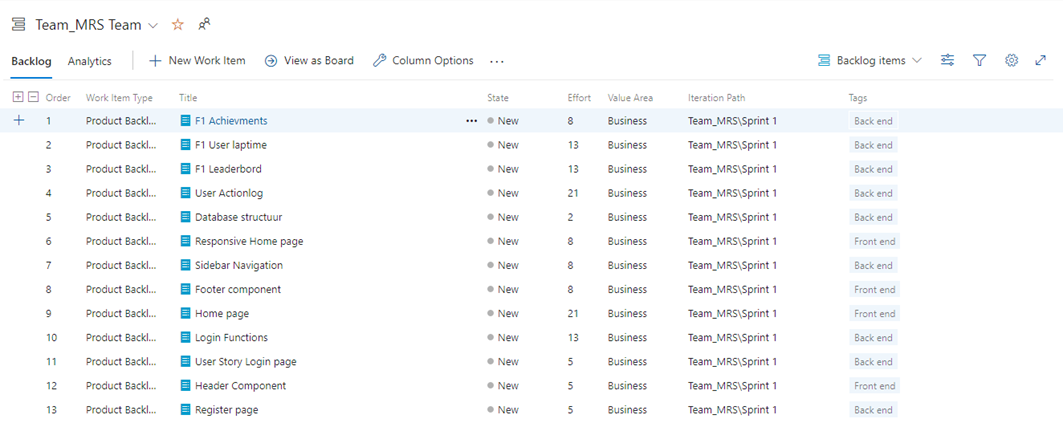
En communicatie ging via Discord en Whatsapp.

**Way of Working:**



**Backlogbeheer:**

Aan het begin van de dag gaan we in Discord en houden wij onze dagelijkse stand-up vergaderingen en sprint review waar wij bespreken wat we gaan behandelen van de backlog, en delen wij die groteren EPICS op in meerdere sprints die vervolgens in meerdere User Storys omgezet worden.



**(Zie volgende pagina)**

**Scrum**

**Sprint Oplevering/ Sprintplanning:**

De groep levert de tot zover werkende software op aan het einde van elke sprint.

En melden vervolgens ook welke problemen zij hebben opgelopen met de opgepakte user story’s of welke ze allemaal afgemaakt hebben met succes.

**Scrum**

**Definition of Done:**

Onze D.O.D. Bestaat uit:

**Stap 1**

Het begint uiteraard met het schrijven van de code.

**Stap 2**

En dan moet de code gecommenten worden.

**Stap 3**

Het testen van die code in de aparte brench in zichzelf.

**Stap 4**

Het dan samenvoegen met de Main brench en het geheel testen.

**Stap 5**

Vervolgens meld je bij de groep dat je klaar bent en ga je verder met het volgende item op de sprint.

**Stap 6**

Aan het einde van die dag gaan we in onze afsluit stand up en gaan we over alle veranderingen en progressie die er gemaakt is zover en bespreken wij de zowel geschreven code als comments zodat we alles samen dubbelchecken.

En het dan allemaal definitief goedkeuren en als alles goed is sluiten wij die dag af.

Maar als wij tegen een probleem zijn gelopen bijvoorbeeld met een push conflict zullen wij elkaar in een call in discord gelijk alles verhelpen zodat iedereen productief blijft en weer verder kan.

**Unit Test**

Wij hebben ervoor gekozen om de unit tests tijdens onze productie te doen, een voorbeeld hiervan is:

**Beveiliging**

Project Setup:

**Prestatie Overweging**

**Installatie handleiding**

**Implementatie handleiding**

**Onderhoud**

**Toekomstige ontwikkelingen**