

Reversi documentatie

Geschreven door: Dirk-Jan Eringa
Datum: 28 februari 2021

Inhoud

Inleiding.....	3
1. Functioneel ontwerp.....	3
1.1 Prototype	3
1.2 Use cases.....	6
1.3 Use case diagrammen	7
2. Reversi.....	8
3. Technisch ontwerp.....	11
3.1 Klassendiagram	11
REST-endpoints	12
3.2 Verantwoording technisch ontwerp backend	13
3.3 Verantwoording technisch ontwerp frontend.....	13
4. Security	14
4.1 Backend.....	14
4.2 Frontend.....	14
5. Installatie handleiding.....	14
9. Git hub.....	15

Inleiding

1. Functioneel ontwerp

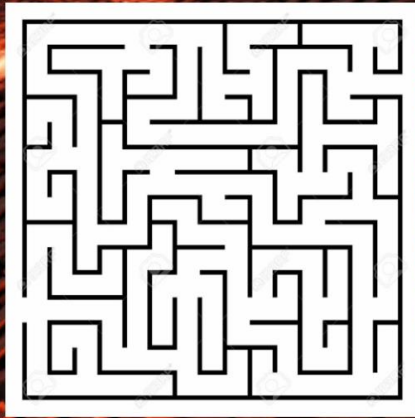
1.1 Prototype

Onderstaand staan een aantal schermprints van de webapplicatie die ik voor ogen had toen ik aan deze eindopdracht begon. Gaandeweg heb ik vanwege de complexiteit besloten om het spel Labyrinth te vervangen door Reversi. Daarmee kwam ook de achtergrond te vervallen. Daarna heb ik gekozen voor een styling die enigszins gebaseerd op die van het Nintendo Entertainment System uit de jaren '80 (zwart, rood, grijs).

Met de keuze voor een ander spel is de 'spelpagina' gewijzigd van Labyrinth naar Reversi. De overige pagina's (chat, options, score) zijn hetzelfde gebleven.

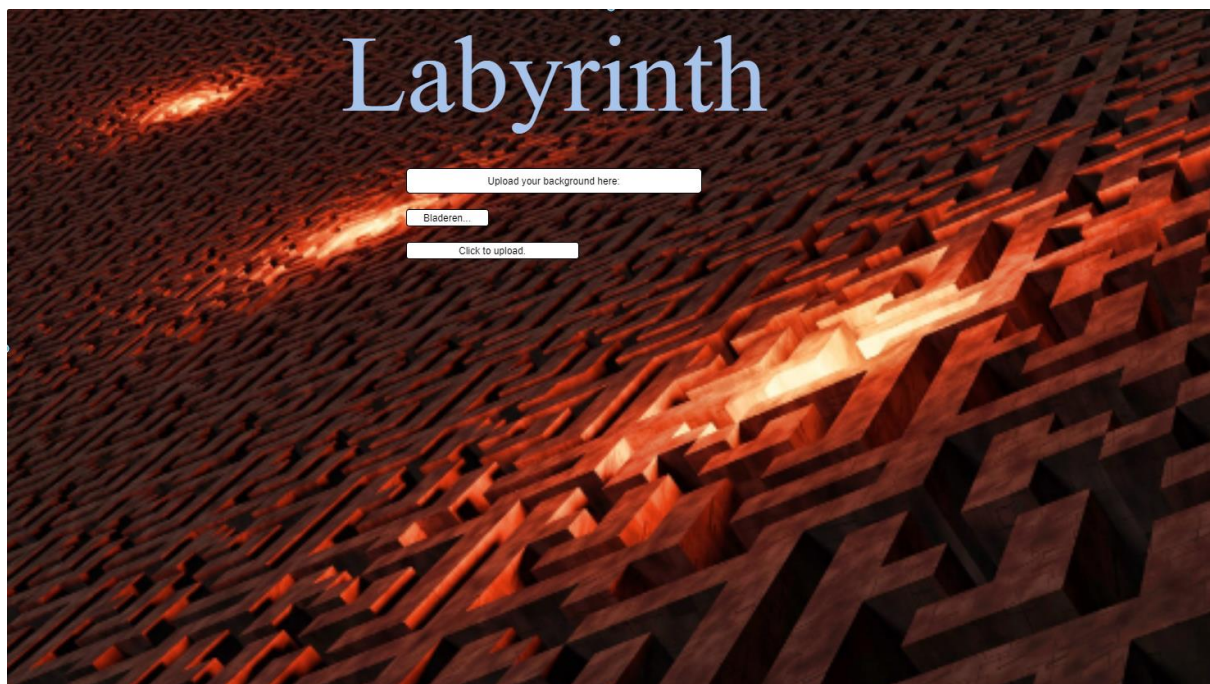


Labyrinth



Labyrinth

Messages



1.2 Use cases

Use Case:	Account aanmaken
Nummer:	1
Actor:	Gebruiker
Preconditie:	Gebruiker is op 'sign up' pagina.
Beschrijving:	Vraag de gebruiker om een emailadres, gebruikersnaam, een wachtwoord en het herhaalde wachtwoord.
Postconditie:	Gebruiker heeft een account

Use Case:	Inloggen
Nummer:	2
Actor:	Gebruiker
Preconditie:	Gebruiker is op de inlog pagina.
Beschrijving:	Vraag de gebruiker zijn gebruikersnaam en wachtwoord in te vullen.
Postconditie:	Gebruiker is ingelogd.

Use Case:	Reversi starten
Nummer:	3
Actor:	Gebruiker
Preconditie:	Er staan geen namen onder Player 1 en Player 2 op de Reversi pagina.
Beschrijving:	Gebruiker klikt op knop "Click here to start the game"
Postconditie:	De namen van spelers zijn zichtbaar onder Player 1 en Player 2.

Use Case:	Reversi spelen
Nummer:	4
Actor:	Gebruiker
Preconditie:	De namen van spelers zijn zichtbaar onder Player 1 en Player 2.
Beschrijving:	Gebruiker klikt op een veld in het Reversi bord.
Postconditie:	In het aangeklikte veld staat het token van de gebruiker.

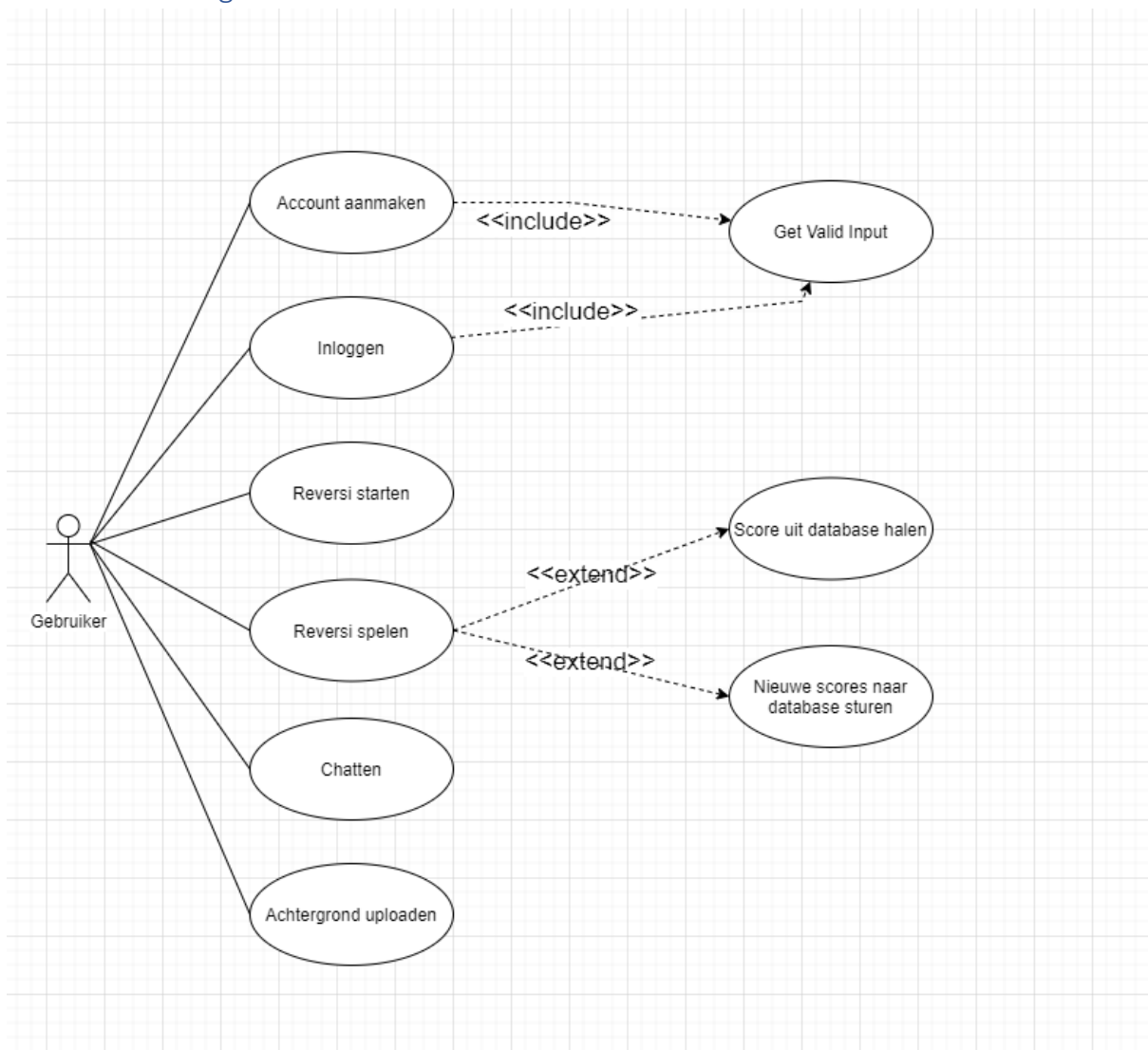
Use Case:	Naam op chat pagina
Nummer:	5
Actor:	Gebruiker
Preconditie:	Er staan geen namen in het eerste grijze kader in de chatbox.
Beschrijving:	Gebruiker gaat naar de chat pagina.
Postconditie:	De naam van de gebruiker wordt bovenin weergegeven.

Use Case:	Chatten
Nummer:	6
Actor:	Gebruiker
Preconditie:	Gebruiker is op de chat pagina
Beschrijving:	Gebruiker typt een bericht in de 'chatbox' in het veld waar 'Say something...' staat en drukt op Send.
Postconditie:	Bericht wordt weergegeven in het berichten veld in de 'chatbox'.

Use Case:	Avatar uploaden
Nummer:	7
Actor:	Gebruiker
Preconditie:	Gebruiker is op de Options pagina
Beschrijving:	Gebruiker selecteert avatar en klikt op upload button.
Postconditie:	Avatar staat in de database en wordt op de website getoond als avatar.

Use Case:	Scores laten zien
Nummer:	8
Actor:	Gebruiker
Preconditie:	Gebruiker is op scorepagina
Beschrijving:	Scores worden opgehaald uit de database.
Postconditie:	Scores worden getoond op de scorepagina.

1.3 Use case diagrammen

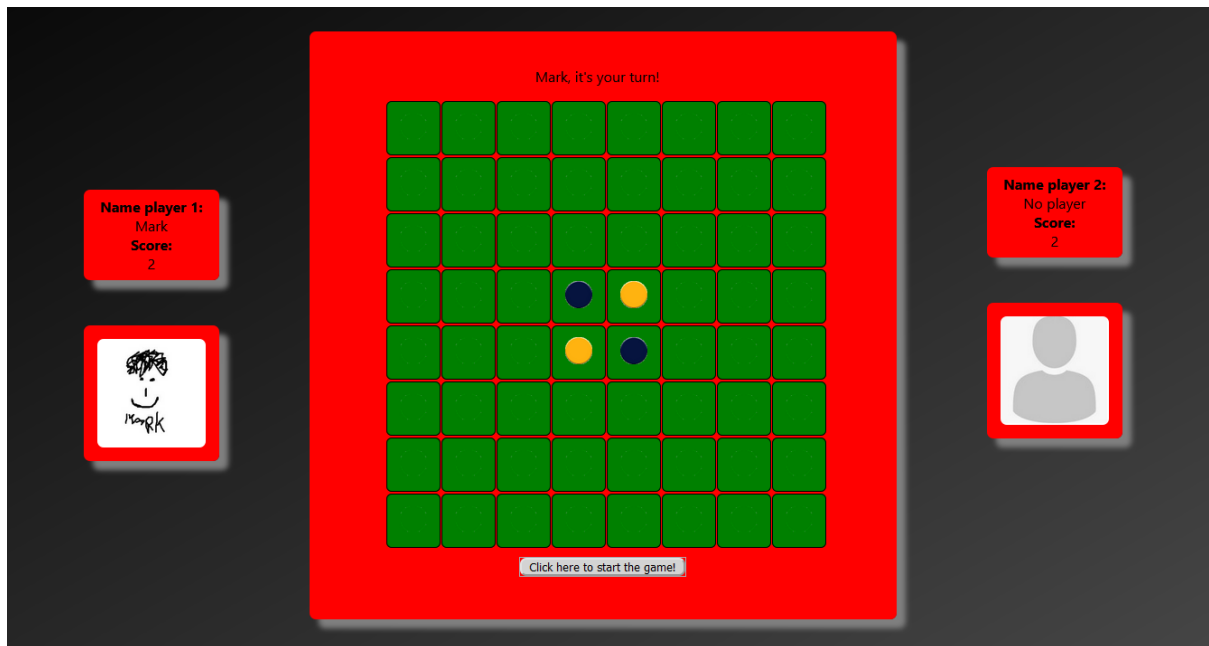


2. Reversi

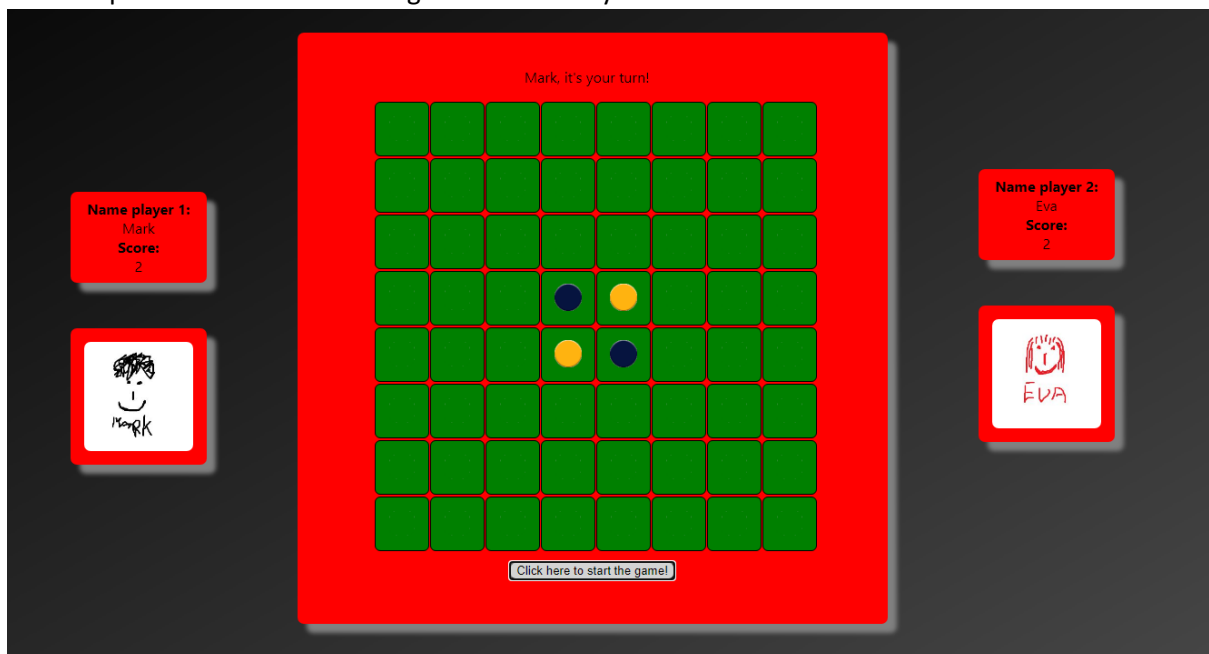
Tijdens het bouwen van deze website kreeg ik van meerdere mensen te horen dat zij het spel Reversi niet kennen. Onderstaand een korte uitleg over hoe dit spel gespeeld wordt.

Om te beginnen moeten beide spelers op 'Click here to start the game!' klikken. Dan worden de namen van de spelers boven het spel en onder Player 1 en Player 2 zichtbaar.

Na klik op "Click here to start the game!" door Player 1:

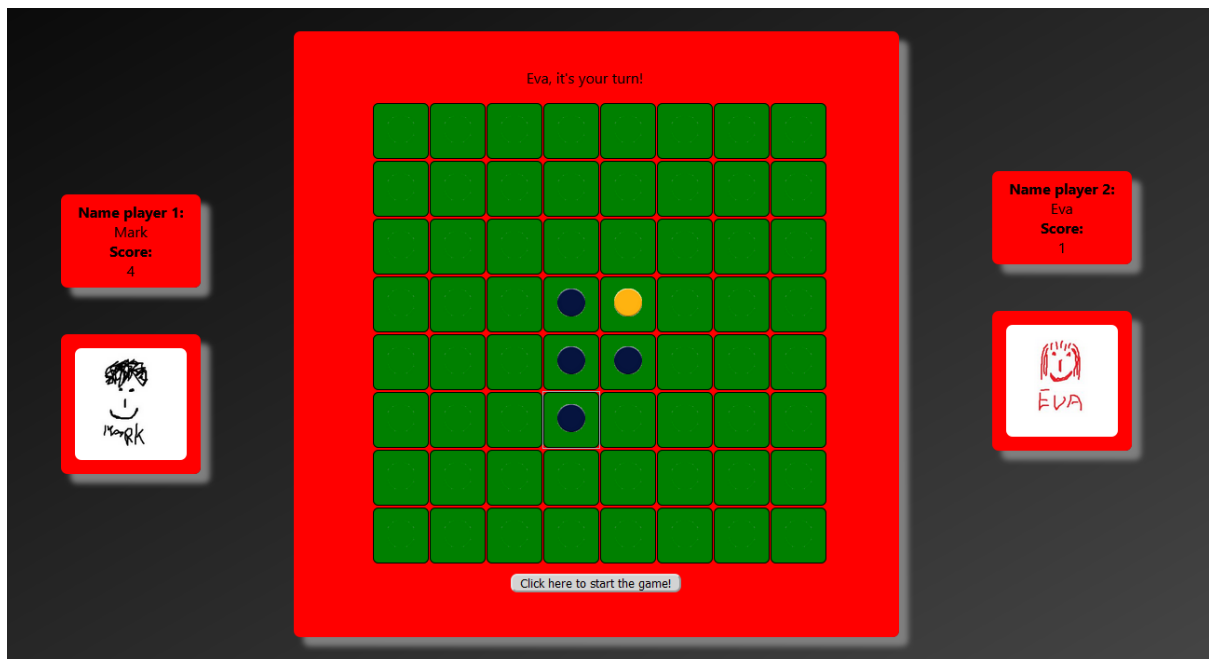


Na klik op "Click here to start the game!" door Player 2:

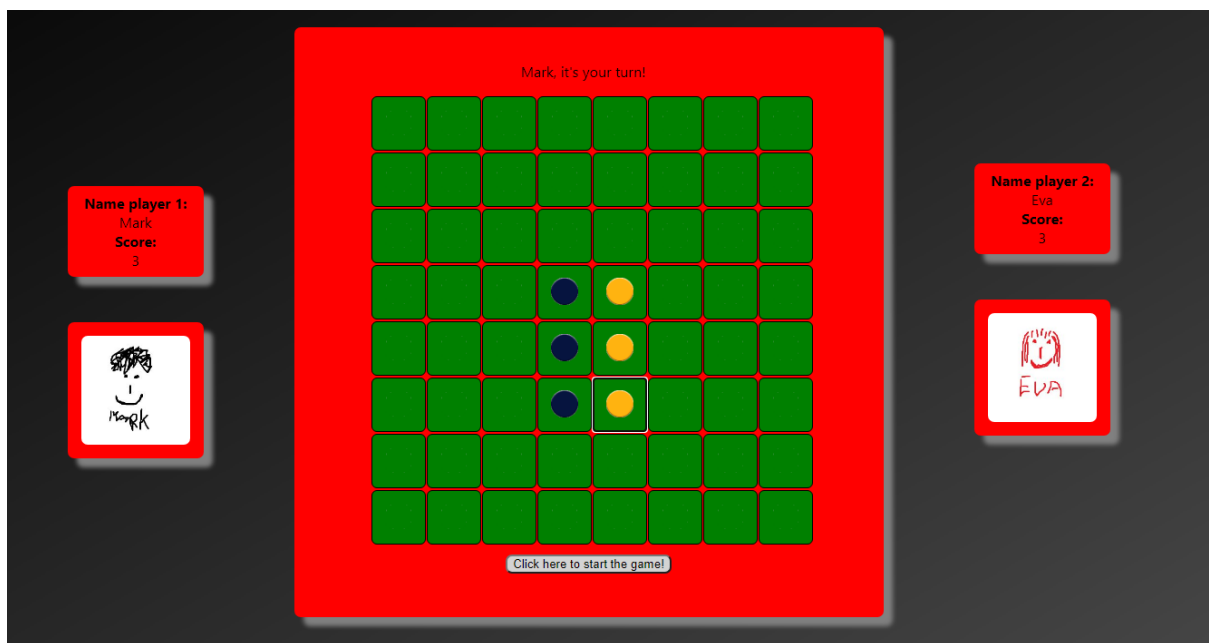


Beide spelers beginnen met 2 stenen. Player 1 heeft de donkere stenen. Player 2 heeft de lichte stenen. Het doel is om elkaars stenen te veroveren. Dit doet men door een steen direct naast een steen van de tegenstander te leggen.

Mark kan bijvoorbeeld het volgende doen:

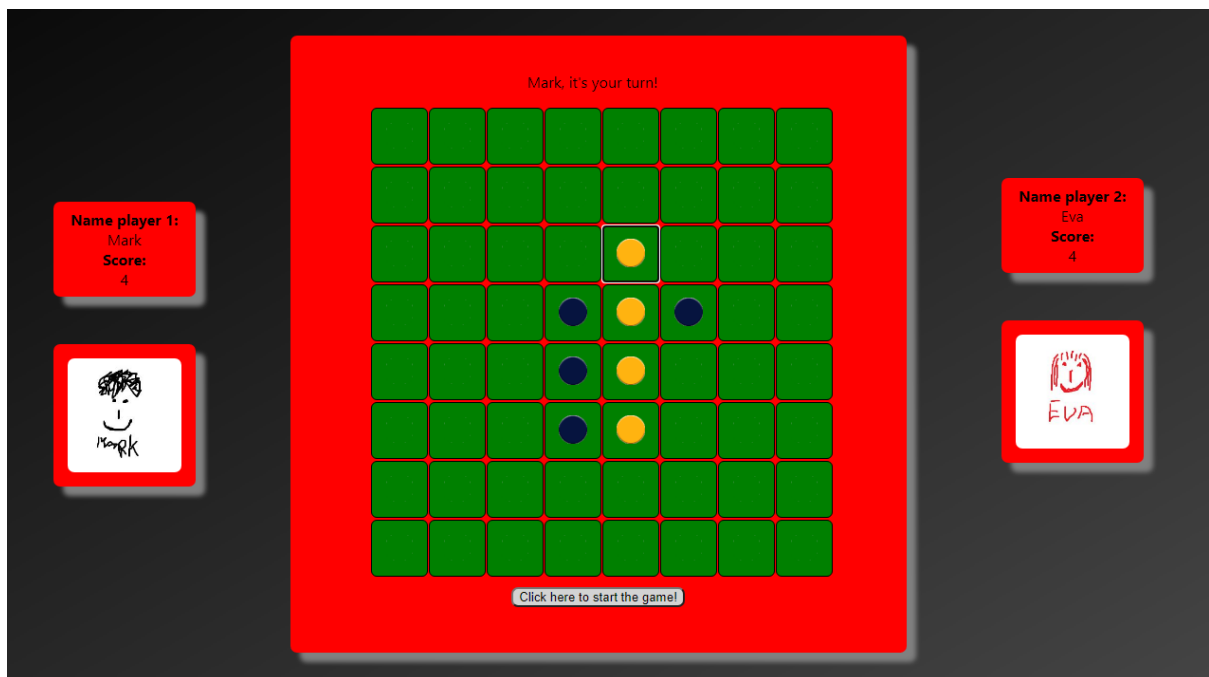
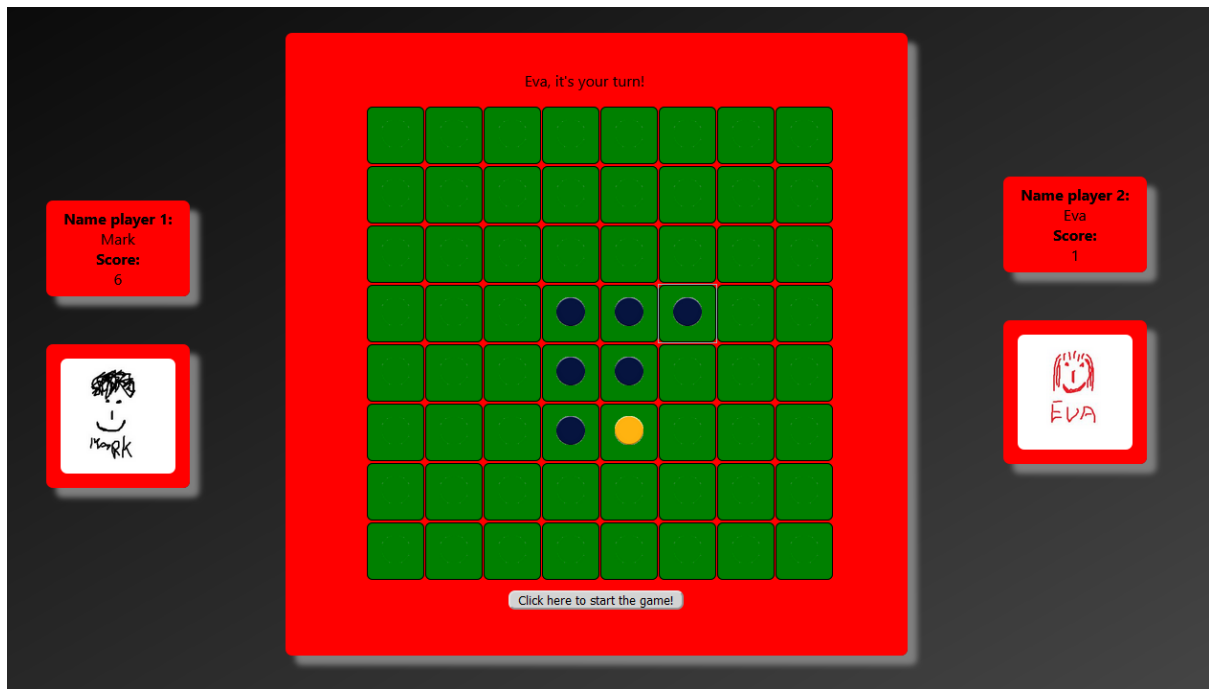


En daarna kan Eva de volgende zet doen:



Als men een horizontale, verticale of diagonale lijn trekt van de eigen steen naar de steen van tegenstander dan moet 1 van de volgende stenen in die lijn weer een eigen steen zijn.

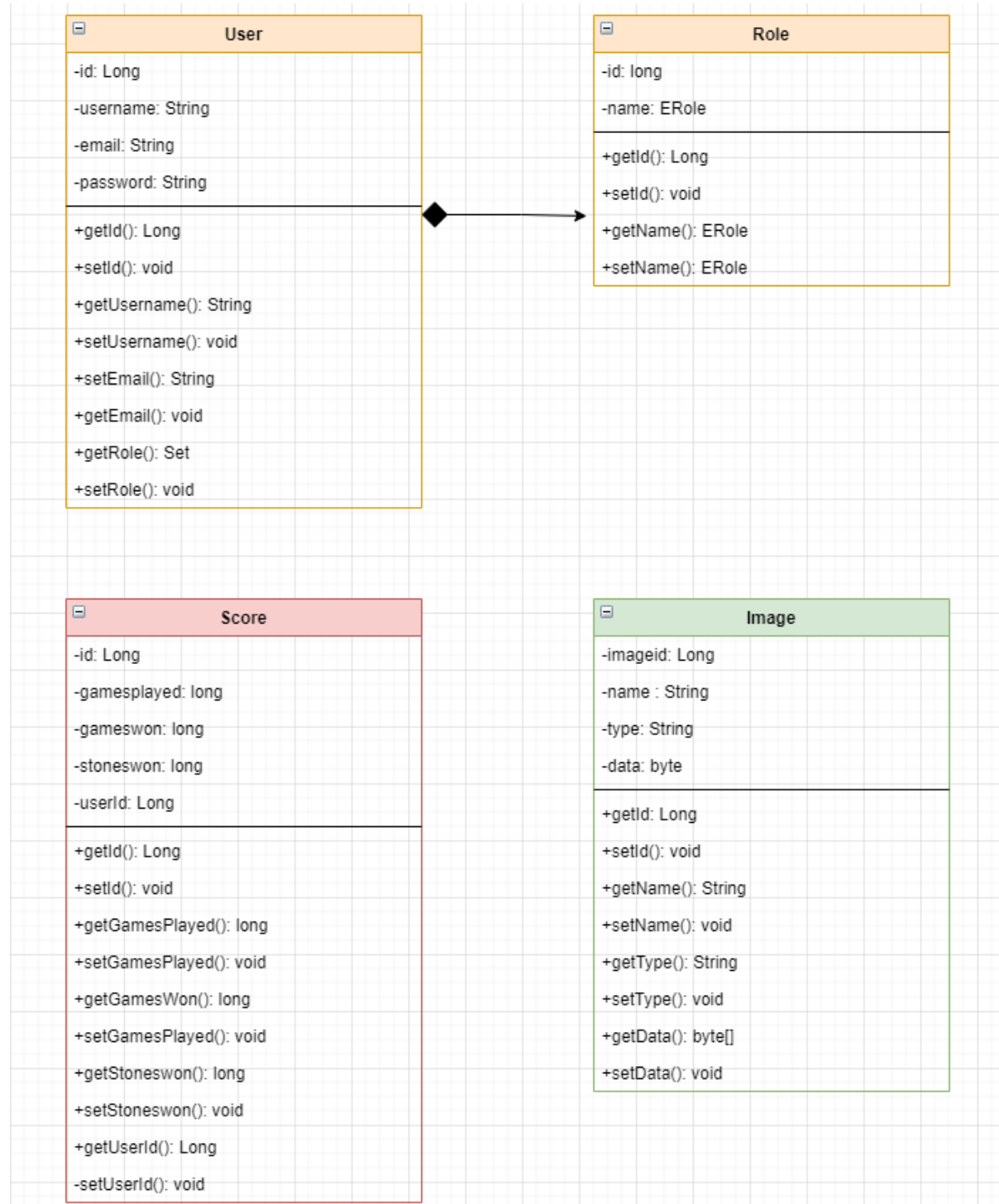
Eva kan bijvoorbeeld meerdere stenen tegelijk veroveren:



De speler die aan het eind van het spel de meeste stenen heeft veroverd wint het spel. Een gelijkspel is mogelijk.

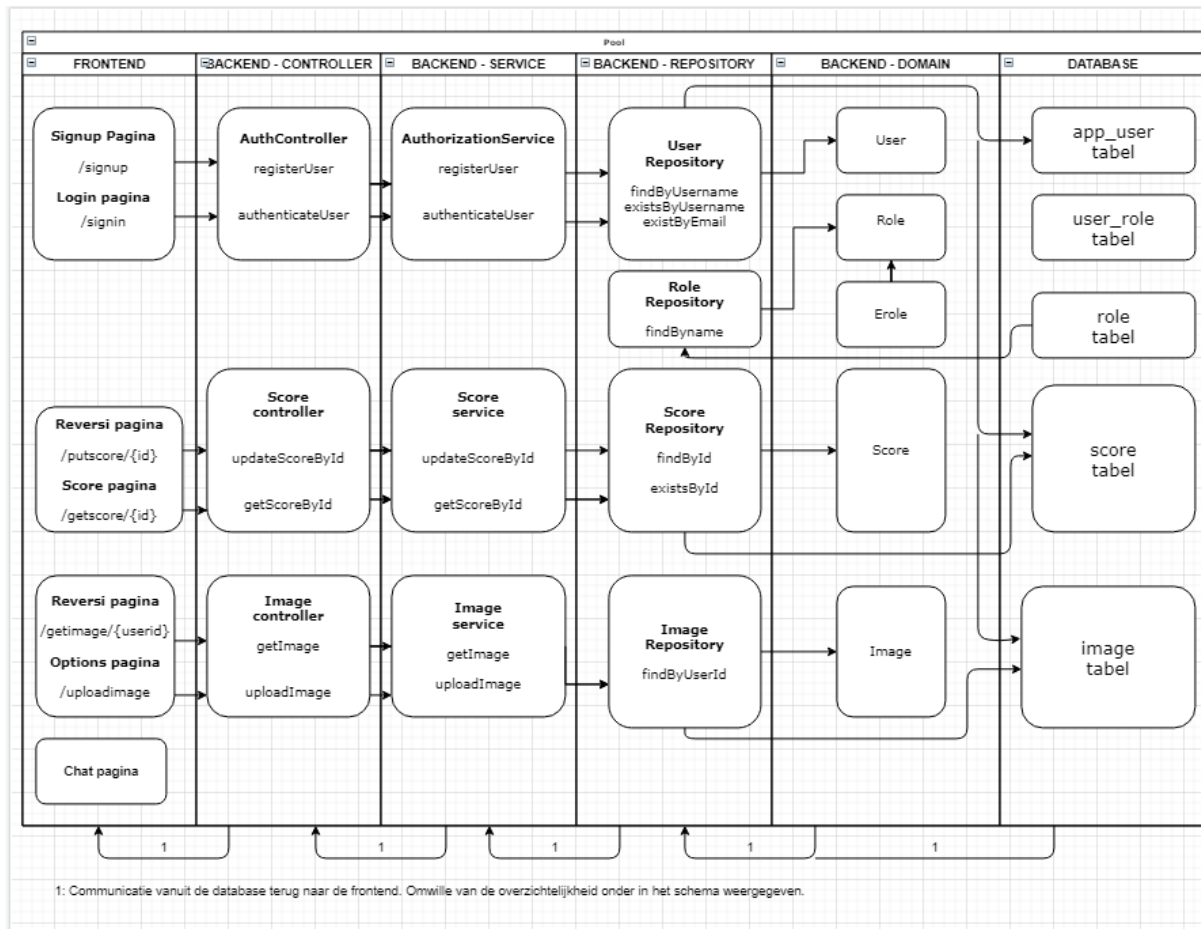
3. Technisch ontwerp

3.1 Klassendiagram



3.2 REST-endpoints

Een schematische weergave van de endpoints:



Onderstaand een beschrijving van de REST-endpoints per controller in de backend:

1. AuthController.

`/signup:`

Dit endpoint maakt nieuwe gebruikers aan. De data wordt opgeslagen in de User tabel. Dit endpoint wordt in de frontend gebruikt op `signup-pagina`.

`/signin:`

Dit endpoint verifieert gebruikersgegevens als een gebruiker probeert in te loggen. Verificatie wordt gedaan met data uit de User tabel.

2. ImageController

`/postimage/{userid}`

Dit endpoint zorgt voor een upload van avatars/foto's naar de database. De data wordt opgeslagen in Image tabel. Deze endpoint wordt in de frontend gebruikt in de options-pagina om een avatar in te stellen.

`/getimage/{userid}`

Dit endpoint haalt avatars/foto's op uit de database. De data komt uit de Image tabel. Dit endpoint om de JWT-token te verifiëren als een gebruiker na het inloggen 1 van de vier pagina's opgevraagd. In de frontend wordt dit endpoint gebruikt op de Reversi-, Chat-, Options- en de Score-pagina. Verder wordt dit endpoint op de Reversi pagina gebruikt om de avatars naast het speelbord weer te geven.

3. ScoreController

`/getscore/{id}`

Dit endpoint haalt scores op uit de database. De data komt uit de Score tabel. In de frontend wordt dit endpoint gebruikt in de Score-pagina.

`/putscore/{id}`

Dit endpoint schrijft scores weg naar de database. De data wordt opgeslagen in de Score tabel. In de frontend wordt dit endpoint gebruikt op de Reversi-pagina om de scores in de database te updaten.

3.3 Verantwoording technisch ontwerp backend

In de backend zijn twee zaken die nog beter kunnen:

1. De return statements geven nu alleen datgene terug wat al dan niet is aangemaakt. Dit kan met http status codes nog specifieker zodat dit ook beter kan worden afgestemd op de frontend.
2. De huidige library voor het up en downloaden van images heeft alleen een mogelijkheid voor post requests. Dit moet aangepast naar een library die put requests kan uitvoeren. Als nu meerdere images worden geüpload geeft dit een foutmelding als getImage wordt uitgevoerd.

3.4 Verantwoording technisch ontwerp frontend

In de frontend kunnen de volgende zaken nog worden verbeterd:

1. De website heeft nog geen styling voor gebruik op een mobiele telefoon.
2. Per pagina/component zijn er .css bestanden gemaakt. Dit kan nog efficiënter met 'styled components'.
3. Op diverse plekken zijn 'let'-variabelen gebruikt. Dit levert mogelijk een verhoogd beveiligingsrisico op en moet dus anders worden ingericht.
4. In de Reversi component zijn alle functies in hetzelfde .js bestand geplaatst. Toen ik begon met het schrijven van de code voor Reversi bleek het lastig om alle variabelen in de juiste 'state' te krijgen. Daarom heb ik vrijwel alle code in hetzelfde .js bestand geschreven. Dit heeft geleid tot zo'n 754 regels code en moet nog opgesplitst worden.
5. In de Reversi component heb ik de renderSquare functie 64 keer moeten gebruiken. Het gebruik van map i.c.m. de board array leidde ertoe dat de 'squares' niet of pas na twee keer klikken de juiste token lieten zien.
6. Tijdens het spelen van Reversi worden soms de verkeerde stenen omgedraaid. Oorzaak is nog onbekend.
7. Er worden http status codes vermeld in console. Dit kan nog verbeterd worden door de teruggave van status codes vanuit de backend beter in te richten. Zie ook punt 1 onder "Verantwoording technisch ontwerp backend".

8. In de console wordt de waarschuwing: "Can't perform a React state update on an unmounted component. This is a no-op..." gegeven. Hier heb ik nog geen oplossing voor gevonden.
9. Er kan eenmalig een avatar worden geüpload. Zie punt 2 onder paragraaf 3.3.

4. Security

Gebruikers moeten eerst een account aanmaken om gebruik te kunnen maken van Reversi. Nadat het account is aangemaakt kan de gebruiker inloggen. Het openen van de webpagina's (Reversi, chat, options, scores) door de gebruiker wordt geverifieerd met JWT tokens.

De volgende stap in security is het beveiligen van de backend en frontend met SSL. Onderstaand een beschrijving hoe dit deels al is uitgevoerd en hoe dit verder wordt gedaan als de website publiekelijk beschikbaar wordt op het internet.

4.1 Backend

In de map Eindopdracht\Reversi1\src\main\resources staat het bestand application.properties. Onderaan in dit bestand staan een aantal regels 'greyed out'. Als men deze regels op actief zet worden de security instellingen actief. De regels bevatten verwijzingen naar het certificaat reversi.p12 dat in dezelfde map staat. Met het inschakelen van de security instellingen moeten de verwijzingen naar de backend in de frontend worden aangepast naar bijvoorbeeld <https://www.reversi.nl:443/>...

Ik heb geen beschikking over een domeinnaam en heb nog niet uitgevonden hoe je een certificaat kunt instellen op een IP adres. Het voerde echter te ver door om dit voor deze eindopdracht uit te zoeken.

4.2 Frontend

Voor het inschakelen van security op de frontend kan het volgende worden toegevoegd in package.json in de root directory van de React app:

```
"scripts": {  
  "start": "set HTTPS=true&&react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
},
```

Hierna zijn alle pagina's te bereiken via <https://192.168.1.218:3000/>... Het instellen van SSL voor de frontend leidt er echter toe dat window.location.assign in bijvoorbeeld Signup.js niet meer werkt. Op dit moment weet ik nog niet hoe ik dit kan oplossen.

5. Installatie handleiding

1. Maak in pgAdmin een database aan met de naam ReversiDB.

2. Open het `server/src/main/resources/application.properties` bestand. Wijzig de `spring.datasource.url=jdbc:postgresql://localhost:5432/ReversiDB` naar jouw PostgreSQL instellingen wanneer nodig, de applicatie gebruikt de standaard PostgreSQL database setup.
3. Pas de `spring.datasource.username` en `spring.datasource.password` aan naar je eigen instellingen in PostgreSQL
4. Open in de IDE het bestand `url.js` in het pad: `...eindopdracht\Reversi1\reversi-app\src\components\url`. Pas het IP adres aan naar het IP adres van je computer. Deze React app maakt gebruik van een server (niet zijnde de backend). Het aanpassen van het IP-adres zorgt voor volledige functionaliteit van de app en voorkomt CORS-foutmeldingen.
5. Voer in een terminal in de directory `C:\Users\dirkj\IdeaProjects\Novi\Eindopdracht\Reversi1` het command `mvn spring-boot:run` uit.
6. Ga naar de directory `C:\Users\dirkj\IdeaProjects\Novi\Eindopdracht\Reversi1\reversi-app\src\server`. Voer in deze directory `npm start` uit.
7. Ga naar de directory `C:\Users\dirkj\IdeaProjects\Novi\Eindopdracht\Reversi1\reversi-app`. Voer in deze directory `npm start` uit.
8. **Ga naar: `xxx.xxx.x.xxx:3000` (ga dus niet naar `localhost:3000`!). `xxx.xxx.x.xxx` is hier het ip-adres dat je in stap 4 hebt opgezocht. In mijn geval is dit bijvoorbeeld: `192.168.1.218:3000`. Om deze webapplicatie te testen kan men het beste op dezelfde computer met twee verschillende browser naar `xxx.xxx.x.xxx:3000` gaan. Deze webapplicatie is via `xxx.xxx.x.xxx:3000` echter ook beschikbaar op andere computers zolang zij gebruik maken van hetzelfde netwerk (in de praktijk dus dezelfde router en/of WIFI-verbinding).**

De volgende pagina's zijn beschikbaar:

<code>xxx.xxx.x.xxx:3000/signup</code>	Signup-pagina
<code>xxx.xxx.x.xxx:3000</code>	Login-pagina
<code>xxx.xxx.x.xxx:3000/reversi</code>	Reversi-pagina
<code>xxx.xxx.x.xxx:3000/chat</code>	Chat-pagina
<code>xxx.xxx.x.xxx:3000/options</code>	Options-pagina
<code>xxx.xxx.x.xxx:3000/score</code>	Score-pagina

Er zijn twee gebruikers aangemaakt. Mark en Eva. Beide hebben het wachtwoord: Password1! gekregen. Bij Mark staan alle scores op 0. Bij Eva staan de scores op: `gamesplayed = 10`, `gameswon = 5` en `stoneswon = 200`.

9. Git hub

<https://github.com/DirkJanE/reversi4.git>