

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Jan De Geest Michaël Depover Thomas De Bauwer Sander Meyfroot |  | Schakeljaar INFORMATICA |  | Sprint 1  06/03/2014 |

Racing

Team 11

Inhoud

1. [Algemeen 2](#_Toc381862791)
2. [Projectstructuur 2](#_Toc381862792)

[Opbouw modulestructuur](#_Toc381862793)

1. [Gebruikte technologiëen 5](#_Toc381862794)
2. [Use Cases 6](#_Toc381862795)

[De circuitontwerper maakt een nieuw circuit aan](#_Toc381862796)

[De circuitontwerper plaatst wegtegels in het raster](#_Toc381862797)

[De circuitontwerper bewerkt een van zijn reeds aangemaakte circuits](#_Toc381862798)

# Algemeen

We ontwerpen een racing game waarbij het mogelijk moet zijn om te racen op een circuit, om zo de snelste tijd te rijden en de overwinning te bereiken. De gebruikers kunnen zelf het circuit ontwerpen in de webapplicatie. Hierna is het mogelijk via de Java desktop applicatie te spelen.

Het ontwerpen van het circuit gebeurt alleen via de webappicatie en speciale functionaliteit is daar beschikbaar:

* De gebruiker kan zelf de grootte van het circuit instellen
* De gebruiker kan zelf de juiste tegels op het raster plaatsen om zo het circuit te bouwen

De snelste rondetijden kunnen dan via de webapplicatie bekeken worden

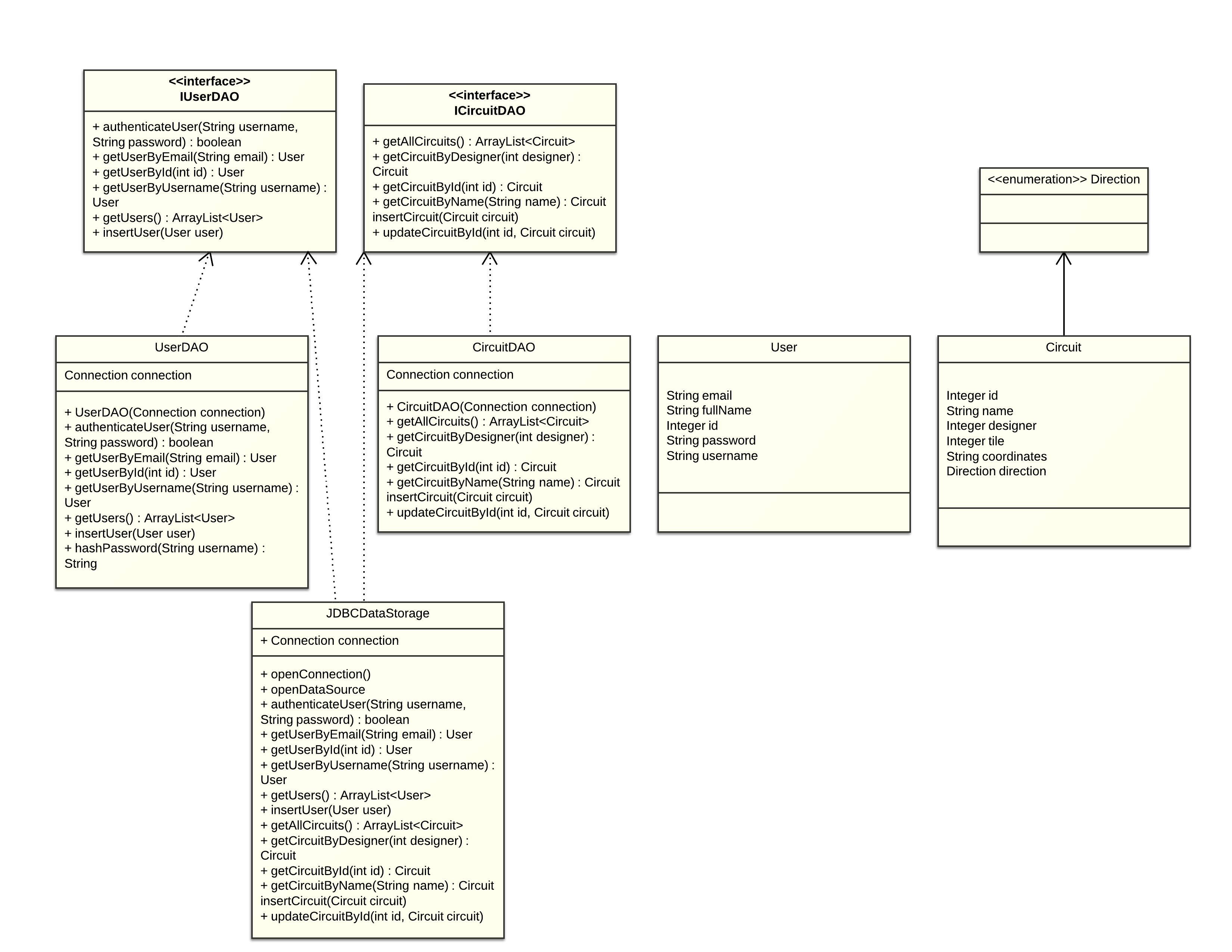
# Projectstructuur

## Opbouw modulestructuur

We gebruiken een Maven Applicatie die bestaat uit de volgende modules:

* RacingDomain

Hier vinden we de persistentielaag terug. We definieren de klassen die we zullen gebruiken, en implementeerden de juiste interfaces en DAO’s. Via een eigen ontworpen IJDBCDataStorage service kunnen we elke interface van elke DAO op een gemakkelijke manier aanspreken in de controller, deze bevindt zich in de API. Het klassendiagram van de RacingDomain module staat hieronder beschreven.



* RacingAPI

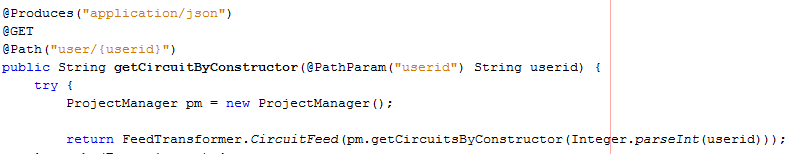
We gebruiken een restfull service om alle aanvragen van de Webapp en Java desktop app te behandelen. Deze service zal de aanvragen behandelen door de controller aan te spreken. Deze zal op zijn beurt naar de IJDBCDataStorage service in het RacingDomain gaan om de data uit de database op te halen. Uiteraard wordt er gebruik gemaakt van de nodige authenticatie voor elke aanvraag aan de API. Alle uitwisseling van gegevens van de API gebeurt via JSON.

De authenticatie wordt verzorgt dmv een custom header. Tijdens het aanmelden wordt deze custom header opgevult met een base64 encoding van de gebruikersnaam in combinatie met het passwoord. Deze gegevens zullen bij elke aanvraag aan de service gebruikt worden om de toegang te verkrijgen

De API is aanspreekbaar op volgende manier:

Vb: Circuits ophalen van een gebruiker: <ip>:<port>/circuits/user/<userid>

Om de id uit de url te halen gebruiken we de @PathParam annotation



* RacingWebApp

Voor de acties die via de webapplicatie moeten gebeuren hebben we een apparte module. We zullen alle call’s naar de service laten verlopen via Servlets. Een formulier gaat dus zijn gegevens posten naar een Servlet die dan op zijn beurt de data in JSON naar de service zal sturen. De service zal ten aller tijde zijn antwoord formuleren in JSON, en terug geven naar de Servlet, die dan op zijn beurt de gegevens eventueel kan terug geven naar de webpagina’s. Voor de lay-out gebruiken we HTML5, CSS3, twitterbootstrap en javascript.

* RacingDesktopApp

Via de desktopapplicatie kunnen we rechtstreeks de API aanspreken om de nodige informatie te versturen of ontvangen. We gebruiken hier een apparte klasse waarin alle informatie wordt bij gehouden en dus over de verschillende formulieren en forms kan gebruikt worden. De call’s naar de API gebeuren via threading, omdat de applicatie niet zou vastlopen als de connectie even op zich laat wachten.

# Gebruikte technologiëen

* Database
  + Mysql
* API:
  + Restfull service
* Authenticatie (structuur)
  + Jeresey
* Authenticatie (passwoord encryptie)
  + SHA-512
* Grafisch (Webapp)
  + Twitterbootstrap
  + JQuery
  + HTML5
  + CSS3
  + Javascript
* Grafisch (Java desktop app)
  + Swing

# Use Cases

## De circuitontwerper maakt een nieuw circuit aan

### Beschrijving

De gebruiker kruipt in de rol van circuitontwerper een maakt een nieuw circuit aan.

### Prioriteit

Hoog

### Complexiteit met ontwerp en code.

Hoog

### Precondities

* De gebruiker is ingelogd

### Postcondities

* De ontwerper plaatst wegtegels (Zie aparte use case)
* Het circuit is geldig
* Het circuit werd opgeslagen in de database

### Primaire actoren

* Gebruiker in de rol van circuitontwerper

### Secundaire actoren

### Successcenario

1. De ontwerper kiest de grootte van het raster (?)
2. De ontwerper geeft een naam aan het circuit
3. De ontwerper legt de nodige wegtegels
4. Het resultaat is een gesloten circuit.
5. De ontwerper submit zijn circuit

### Alternatieve scenario’s

3. a De circuitontwerper legt de tegels foutief op het raster (Zie aparte use case)

4.a Het circuit is niet gesloten, terug naar stap 3.

## De circuitontwerper plaatst wegtegels in het raster

### Beschrijving

Tijdens het ontwerpen van het circuit kiest de gebruiker een wegtegel. Er zijn twee soorten wegtegels, een recht stuk en een L-vormige bocht.

### Prioriteit

Gemiddeld

### Complexiteit met ontwerp en code.

Gemiddeld

### Precondities

* De ontwerper is ingelogd
* De ontwerper heeft reeds een circuit aangemaakt

### Postcondities

* Het circuit wordt geschreven naar de database

### Primaire actoren

* Ontwerper

### Secundaire actoren

### Successcenario

1. De ontwerper kiest een reeds bestaand circuit van zijn hand
2. De ontwerper maakt het circuit door de wegtegels op het raster te leggen.
3. Het circuitontwerp is geldig.
4. De ontwerper slaat het circuit op.
5. Het circuit worden weggeschreven naar de databank

### Alternatieve scenario’s

3.a Het ontwerp is niet geldig

5.a De wijzigingen konden niet worden opgeslaan, terug naar stap 4.

## De circuitontwerper bewerkt een van zijn reeds aangemaakte circuits

### Beschrijving

Een ontwerper kan een van zijn vorige ontworpen circuits aanpassen

### Prioriteit

Gemiddeld

### Complexiteit met ontwerp en code.

Gemiddeld

### Precondities

* De gebruiker is ingelogd
* De gebruiker heeft reeds één of meerdere circuits ontworpen

### Postcondities

* De wijzigingen worden upgedate in de database
* De tijden van het circuit vallen weg.

### Primaire actoren

* Ontwerper

### Secundaire actoren

### Successcenario

1. De gebruiker selecteert een vorig ontwerp

### Alternatieve scenario’s