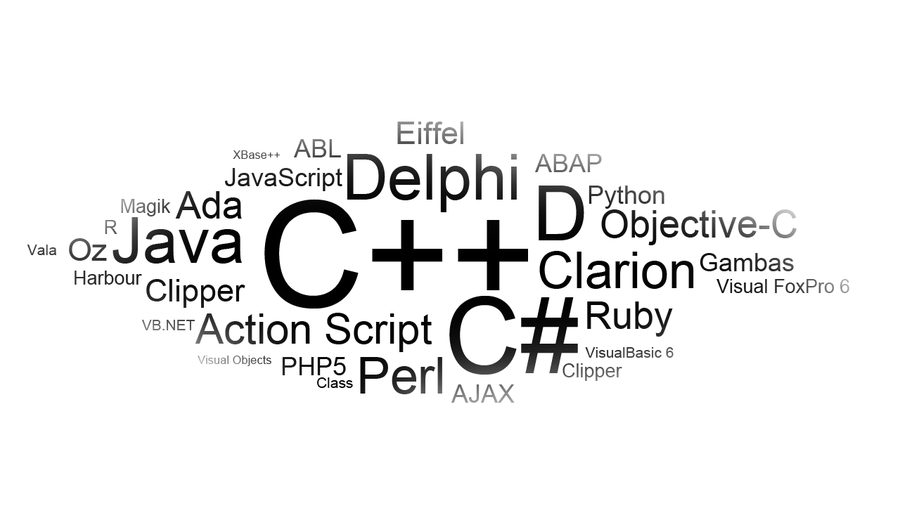
Yuri Soens

Programmeren 4

Project Quiz

Docent: Tom Verbesselt

Programmeren 4 Dossier



Inhoudstafel:

* Probleemomschrijving (pagina 2)
* Analyse (pagina 3 - 5)
* Problemen (pagina 6 - 8)
* Besluit (pagina 8)

Probleembeschrijving

Ik ben de laatste tijd vrij vaak bezig met (gezelschap)spellen te spelen onder vrienden. Één van die spellen is een quiz. Om de 2 weken maakt iemand in onze vriendenkring een quiz op en presenteert deze ook. Wij hebben hier allemaal al veel plezier aan beleefd en daarom had ik het idee om als mijn project een grote quiz app te maken.

Het is ondertussen al een half jaar sinds onze eerste quiz en het aantal vragen dat al zijn gesteld is fenomenaal. Het zou zonde zijn als al deze vragen en moeite gewoon de vuilbak in gaan. Als ik een webapp kan maken dat al deze vragen en statistieken over de vragen bijhoud kunnen we er nog lang van genieten. Mijn idee was om elke quiz via een desktop applicatie toe te voegen aan een database. Deze database kan dan geaccessed worden via de webapp. Je zal de optie hebben om de laatste nieuwe quiz volledig opnieuw te spelen. Deze zal voltooid worden en dan wordt er door het systeem bijgehouden welke vragen makkelijk, normaal of moeilijk waren (aan de hand van een percentage goed/fout). Een quiz in mijn applicatie bestaat uit 10 vragen, je hebt de optie om de moeilijkheidsgraad te kiezen. Ik zal ook een tag geven aan elke vraag tot welke categorie het behoord zodat ik ook kan uitbereiden naar categorie specifieke quizzen. Om een nieuwe quiz toe te voegen op de desktop applicatie zal ik een template maken van wat soort vraag het is (multiple choice, enkel antwoord, foto, etc.) Ik weet nog niet of ik topscores of gemiddelde scores etc. wil bijhouden want uiteindelijk zullen mensen de vragen van buiten weten. Er zal misschien ergens nog wel een punten systeem zijn maar wat voor iets of hoe ik het zal doen weet ik nog niet.

Ik denk dat dit een heel leuk project kan worden, zeker als er veel gebruikers vragen beginnen aanmaken, maar of ik dat in mijn origineel project zal steken weet ik nog niet. Dit project zal misschien niet het meest business georienteerd zijn, maar ik zal er vast plezier aan hebben om het te maken en te delen met vrienden.

Analyse

Fasering:

* Ik begin dit project met het opstellen van enkele vragen zodat ik kan zien wat voor velden er uiteindelijk nodig zullen zijn voor mijn database. Gebruikers moeten het spel ook leuk vinden dus zal ik ook wat research doen in deze fase over wat mensen verwachten als ze een quiz programma spelen. Mijn eerste gedachten waren “clean en simpel”, de meeste kleine spellen die leuk blijven zijn de diegene zonder al de toeters en bellen (bv. Pac-man, tetris, angry birds).
* Vervolgens zal ik mijn database en schemas aanmaken, het ik veel makkelijker om dit eerst te doen voor ik aan de code begin zodat ik een goed overzicht heb van het project.
* De volgende logische stap lijkt mij om de desktop applicatie te maken. Hier zal ik een form maken om vragen toe te voegen. Ik denk niet dat het nodig is om ook van hieruit al gebruikers toe te voegen want dan zou ik usernames en passwords aan mensen moeten toewijzen. De applicatie zal de velden hebben die de database tabel vullen, en ook een lijst van alle aangemaakte vragen. Sinds ik verschillende soorten vragen zal hebben maak ik enkele velden ontoerijkbaar tenzij het specifiek aangegeven word dat ik dat soort vraag ga aanmaken (bv. Multiple choice).
* Uiteindelijk maak ik mijn web applicatie aan die door de gebruiker geaccessed kan worden. De begin pagina zal een login screen zijn die de optie heefd om een nieuwe gebruiker aan te maken. Vervolgens kom je bij de home pagina waar je de moeilijkheidsgraad kan bepalen. Dan zal de quiz 10 willekeurige vragen onder die moeilijkheidsgraad uitkiezen en die om de beurt tonen op het scherm. Tot slot kan de gebruiken zijn score opslaan.

Ik zal gebruik maken van:

* Java EE
* Netbeans IDE
* Glassfish server
* MySQL
* JPA
* JSF

Schatting uren die ik zal besteden:

* 60 tot 100 uren naar gelang hoe ik ver ik mijn applicatie uitbereid

Schemas:







Problemen die ik tegen kwam:

* Om te beginnen heefd Glassfish mij veel problemen gegeven, maar uiteindelijk was het een kwestie van juist te configureren.
* Voor multiple choice heb ik alle velden in 1 gestoken met een splitsingsteken. Ik weet niet of het beter is dan alles appart, maar het ik wel cleaner dan private String multipleChoiceA, etc in de code te hebben.

else if(v.getType().equalsIgnoreCase("Multiple choice"))

{

cmbType.setSelectedIndex(1);

String string = v.getExtraInfo();

String[] parts = string.split(";");

txtKeuzeA.setText(parts[0]);

txtKeuzeB.setText(parts[1]);

txtKeuzeC.setText(parts[2]);

txtKeuzeD.setText(parts[3]);

}

* Ik heb wel meer problemen gehad in mijn webApp. Ik heb vrij weing ervaring met html dus waren er enkele dinges die ik nog wat moest uitzoeken. De code voor de webApp was in het algemeen ook meer complex.
* Ik heb bij elke vraag een nieuwe pagina gemaakt sinds er steeds een random vraag komt en ik weet niet wat voor type het zal hebben (normaal, multiple choice, etc).

public String nextQuestion()

{

checkAntwoord();

updateLijst();

if(indexVragen == 0)

{

return "/index";

}

else if(current.getType().equalsIgnoreCase("Normaal"))

{

return"/questions/normalQuestion";

}

* Ik heb gebruik kunnen maken van enkele default programmas die door de wizard waren aangemaakt. Wat soms ook wel door verwarring zorgde maar nu weet ik hoe ik met deze defaults moet werken.

public void setVragenLijst()

{

indexVragen = 9;

List<TblVraag> list = new ArrayList<>();

list = getFacade().findAll();

* De defaults om een user of vraag aan te maken/updaten gebruikt altijd een tabel, want ik niet zo handig vond. Ik moest een apparte pagina aanmaken om een user zijn score te saven.

public String prepareEdit() {

current = (TblVraag) getItems().getRowData();

selectedItemIndex = pagination.getPageFirstItem() + getItems().getRowIndex();

return "Edit";

}

* Ik heb ook een fout gemaakt in het maken van mijn databank om geen 2 velden te hebben om de score te calculeren (aantal juiste antwoorden en aantal geantwoord). Nu neem ik een gemiddelde van mijn oude score en de nieuwe score, wat er voor zorgt dat de quiz die je neemt mee telt voor 50%.

public void setSkillLevel(double score)

{

double currentScore = current.getAverageScore();

double currentAverage = (currentScore+score)/2;

current.setAverageScore(currentAverage);

}

* Bij de login pagina kwamen de defaults ook weer goed van pas.

public String validateLogin()

{

TblUser user = new TblUser();

List<TblUser> list = new ArrayList<>();

list = getFacade().findAll();

for(TblUser target: list)

{

if(target.getUsername().equals(current.getUsername()))

{

user = target;

}

}

Besluit:

Ik heb veel plezier gehad aan het maken van deze applicate. Het was niet te groot maar ook niet te klein zodat het nog wel een uitdaging bleef. De tijd is wel snel vooruit gegaan als je 3x per week les hebt en dit vond ik wel spijtig, als ik meer tijd had gehad zouden er meer details mogelijk geweest zijn. Dit programma heefd mij goed leren werken met JSF, wat eigenlijk perfect is voor mij want ik ben altijd al fan geweest van java en had weinig ervaring met html. Dit project heefd mij met een java approach naar html gebracht. Het is niet alleen leerrijk geweest voor mij, maar het zal ook leerrijk zijn voor de gebruikers van mijn applicatie sinds het een quiz is. Ik heb mijn familie al eens laten testen, en het was de functionaliteit dat ze verwachtte maar qua layout mocht het beter. Ik heb ook niet zoveel tijd gestoken in de layout omdat ik de functionaliteit het voornaamste vind en omdat ik niet zoveel ervaring heb met css. Dit is iets waar ik later nog tijd aan kan besteden om mijn applicatie cleaner te maken, wat ik zeker ga doen. Dit programma heefd mij zin gegeven om in mijn vrije tijd meer programmas voor mezelf te schrijven en er in het algemeen meer mee bezig te zijn. Het was een fijn project en ik hoop dat mijn score even positief zal zijn als mijn ervaring.