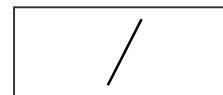


Naam cursist:

Geb. datum en -plaats:

Evaluatie

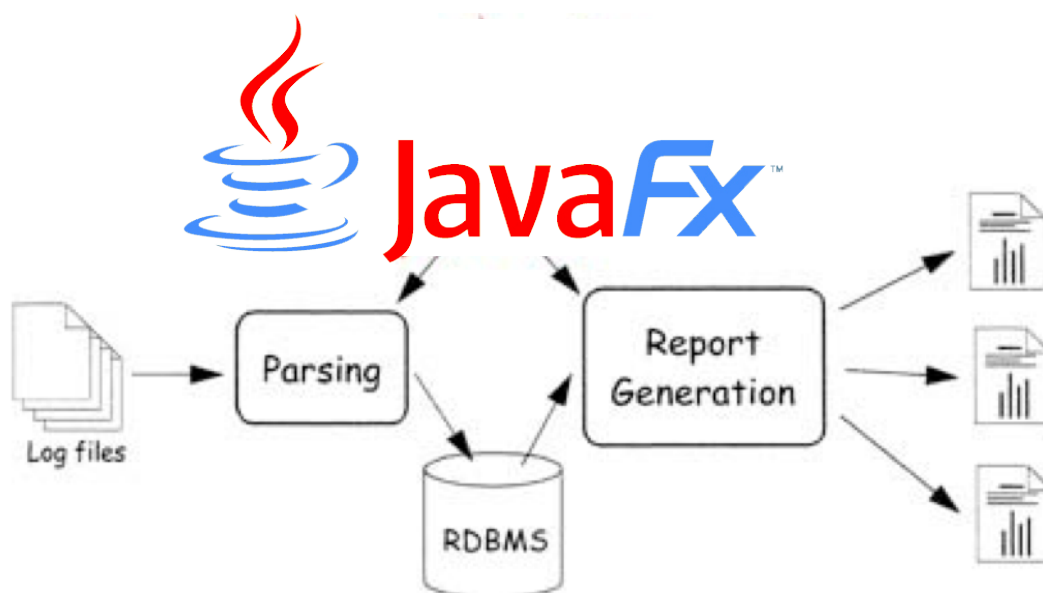


1. Inleiding

- Een CVO wil kijken wat het internetverkeer inhoudt binnen de school en welke andere CVO's gebruik maken van hun applicaties.
- De doelstelling van het project is een applicatie maken die logbestanden analyseert en van deze analyse rapporten genereert.

2. Beschrijving van het probleem

- De bestanden die worden geanalyseerd zijn relatief groot waardoor het onmogelijk is om een directe statistische analyse van deze bestanden te doen. U moet deze bestanden doorlopen en deze informatie wordt opgeslagen in een database.
- Vervolgens kunnen er rapporten gegenereerd worden met de informatie vanuit de database zoals weergegeven in figuur 1.
- De applicatie is een JavaFX-applicatie en moet in staat zijn om input te verwerken :
 - selecteren welke bestanden er moeten worden behandeld
 - selecteren welke rapporten er moeten gegenereerd worden



Figuur 1 : structuur van de applicatie

3. De logbestanden

- U moet twee types van logbestanden analyseren (extensie .log en .txt) :
- Web logs : logs komende van een webserver die een lijst bevatten van alle requests ontvangen door deze server.
- Proxy logs : logs komende van een proxy server die een lijst bevatten van alle uitgaande requests.
- Deze bestanden hebben een verschillende structuur omdat ze worden genereerd door twee verschillende types van servers : Tomcat voor de eerste en een Microsoft ISA server voor de laatste.

3.1. Structuur van de WebLog

- Elke lijn bevat informatie gerelateerd aan een request aan een webserver.

Voorbeeld	Beschrijving
10.120.238.105	IP adres van de klant
-	Gebruikersnaam
-	Remote User indien geïdentificeerd
[06/Dec/2016:07:22:29 1000]	Datum en uur van de request
"GET /ELO/tiles-config_1_1.dtd HTTP/1.1"	Eerste lijn van de request
200	Status
17888	Aantal bytes verzonden door de server

Tabel 1 : Structuur WebLog

3.2. Structuur van de ProxyLog

- Elke lijn is samengesteld uit verschillende gegevens die een http-request bevat.

Voorbeeld	Beschrijving
10.120.238.174	IP-adres van de klant
DOM1/EPU	Gebruikersnaam
Mozilla/4.0 (compatible; MSIE 7.0	Browser die door de klant gebruikt werd
2016-12-08	Datum van de request
09:09:37	Tijd van de request
SHERMAN	Naam van de server
-	Referred
10.142.64.13	Adres van de server
10.142.64.13	IP-adres
80	TCP poort
50	Time taken : tijd nodig om de request uit te voeren
3067	Grootte van de request (in bytes)
1416	Grootte van de response (in bytes)
http	Gebruikt protocol
GET	Gebruikte methode
http://www.b-rail.be/ari/N/images/Home_Ari_But_Network_Active.gif	URI

Tabel 2 : structuur ProxyLog

4. De applicatie

4.1. Model van de requests en sessions (bezoeken)

- Men kan het concept van een *session* (voor de ProxyLog) of een *visit* (voor een webLog) definiëren. Het bestaat uit een geheel van opeenvolgende requests komende van dezelfde gebruiker (eigenlijk dezelfde machine), binnen een termijn van 20' en naar dezelfde applicatie of bezochte website.
- Opmerking : de requests van de proxyLog zonder IP-adres en zonder gebruiker worden genegeerd.
- Elke school beschikt over een range van IP-adressen. U beschikt in de databank over de subnet- masks van elke school. U kan dus elke request koppelen aan een school.

4.2. De rapporten

- U wordt gevraagd om rapporten te genereren. Deze worden in een .pdf formaat. Voor de pdf generatie maakt U gebruik van iText (<http://itextpdf.com/>).
- In deze rapporten (pdf-versie) moeten ook grafieken aanwezig zijn. De grafieken worden aangemaakt aan de hand van JFreeChart (<http://www.jfree.org/jfreechart/>) of via JavaFX grafieken.
- U wordt gevraagd om de volgende rapporten te maken:
 - Voor de binnenkomende requests (visits):
 - Visits per maand per applicatie
 - Voor een bepaalde maand, de Top 10 van de scholen die onze server bezochten
 - Aantal Mbytes per maand per applicatie
 - Voor de uitgaande requests (sessions)
 - Voor een bepaalde maand, de Top 10 van de meeste downloaders (Top 10 MBytes/user)
 - Voor een bepaalde maand, de Top 10 van de mensen die het langst hebben gesurft (Top 10 Time/user)
 - Voor een bepaald jaar, het aantal MBytes/maand
 - De Top 10 van meest bezochte sites (in aantal bezoeken, doorgebrachte tijd en aantal gedownloade bytes)

5. Structuur van de databank

- De SQL-DDL van de database wordt U aangeleverd.
- Om de relatie tussen de tabel Visits en Schools worden de eerste twee bytes van het IP-adres hernomen (bv. "10.120").
- Je mag een andere DB-structuur gebruiken maar deze moet genormaliseerd zijn.

6. Wat wordt er van U gevraagd?

6.1. Applicatie

- Uw applicatie moet modulair opgebouwd zijn en gestructureerd rond drie aspecten:
 - inlezen van de dagelijkse bestanden om zo de databank te vullen met informatie
 - generatie van rapporten door het inlezen van de databasegegevens
 - een JavaFX applicatie die toelaat om die bepaalde taken uit te voeren.

6.2. Behandelen van de logbestanden

- De applicatie moet een individueel bestand kunnen behandelen maar ook een directory kunnen doorlopen waar deze bestanden inzitten.
- Indien een bestand al in de databank zit, dan moet men de mogelijkheid aanbieden om deze data te overschrijven.

7. Evaluatie

- Juistheid (testen): 20%
- Code: 60%
 - Gescheiden Logging (bv. apart logbestand voor DB)
 - Property files (configureerbaar project)
 - Uitbreidbaarheid
 - OO
 - Principes die we tijdens het jaar hebben gezien
 - Frontend in JavaFX
- Documentatie: 5%
- Gebruikte (extra) technologieën: 10%
- Presentatie: 5%

8. Indienen opdracht

- Wanneer: ten laatste op 30 mei 2017
- Hoe: op DVD/CD met daarop naam, geboortedatum, geboorteplaats, module (Programmeren 3) en schooljaar én via de opdrachten module van het ELO.
- Wat: jouw project (Java files !)
Indien je een andere DB-structuur gebruikt, niet vergeten deze mee aan te leveren.
- Presentatie: 13 juni 2017. (nog te bevestigen)

Veel succes!