



Performance van een webgebaseerd E-learning pakket

Scriptie door Stefan Postma

Student: Stefan Postma
Studentnr: 1557397
Instituut: Hogeschool Utrecht
Opleiding: B Informatica - Information engineering
Examinator: Roelant Ossewaarde

Managementsamenvatting

In deze scriptie is een antwoord gezocht naar de vraag:

Welke keuzes voor Eterna, uitgewerkt in een technisch ontwerp, zijn nodig voor optimale performance?

Om deze vraag te kunnen beantwoorden is er een literatuuronderzoek uitgevoerd. Uit dit onderzoek is gebleken dat het woord performance zeer breed opgevat kan worden. Verder is het niet alleen van belang dat het pakket snel is, maar dat het belangrijker is dat het pakket voldoet aan de verwachtingen van de gebruiker. Deze verwachtingen zijn in kaart gebracht en vervolgens getoetst. Als afsluiting is er een advies gegeven over hoe hier nu verder mee om moet worden gegaan en hoe er in de toekomst kan worden gezorgd dat er aan de gebruikersverwachtingen kan worden voldaan.

Voorwoord

ICT in het onderwijs is al jaren een “hot” item. Afgelopen jaar is daar nog de Steve Job school van Maurice de Hond bijgekomen. Mijn interesse in het onderwijs komt voort uit de minor docent in techniek. In een half jaartijd werden we compleet voor de leeuwen gegooid en stonden we les te geven voor een klas van een man of 40. Ook ICT in het onderwijs werd er even tussendoor geschoven. Niet te geloven dat er zo veel ontwikkeling in die onderwijswereld is, maar de vooruitgang soms ver te zoeken is. Toen Martijn Versteeg mij belde met zijn verhaal en visie over de verandering van het voortgezet onderwijs, twijfelde ik geen moment om mee te doen. Zijn visie is inmiddels ruim een jaar later omgezet in het bedrijf waar ik nu deze opdracht heb uitgevoerd.

Inhoudsopgave

Managementsamenvatting	1
Voorwoord	2
1.0 Inleiding	6
2.0 Organisatie.....	7
3.0 Probleemstelling	8
3.1 De opdrachtoomschrijving	8
4.0 Analyse van het probleem.....	9
4.1 Hoofdvraag en deelvragen	9
4.1.1 Hoofdvraag	9
4.1.2 Deelvragen.....	9
5.0 Scope	10
6.0 Opbouw en aanpak	10
7.0 Onderzoeksfase - Literatuuronderzoek	11
7.1 Aanpak	11
7.2 Wat is performance?	11
7.2.1 Performance volgens Cary Millsapp	11
7.2.2 Performance volgens Robert Miller	11
7.2.5 Samenvatting	12
7.3 Wat is efficiency?.....	12
7.3.1 Efficiency volgens Cary Millsapp	12
7.3.2 Efficiency volgens Robert Miller	13
7.3.3 Samenvatting	13
7.4 Wat is Load / utilization?	13
7.4.1 Volgens Cary Millsapp.....	13
7.4.2 Volgens Robert Miller	13
7.4.3 Volgens Jakob Nielsen.....	14
7.4.4 Overdenking en toepassing op Elerna.....	14
7.4.5 Samenvatting	14
7.5 Kendall's notation	14
7.5.1 Volgens Cary Millsapp (Thinking Clearly About Performance).....	14
7.6 The knee	14
7.7 Wat is snel genoeg?	16
7.7.1 Volgens Cary Millsapp (Thinking Clearly About Performance).....	16
7.7.2 Volgens Robert Miller	17
7.7.3 Volgens Nielsen	17

7.7.4 Samenvatting	18
7.8 Welke gebruikersrollen zijn relevant voor Elerna vanuit technisch aspect?	18
7.9 Welke parameters zijn van belang voor Elerna?	19
7.10 Welke waarden zijn te koppelen aan deze parameters?	20
7.11 Welke kritische performance indicatoren voor Elerna kunnen daaruit worden opgesteld?.....	21
7.12 Digitaal gepersonaliseerd leren en big data verzameling	21
7.13 Conclusies literatuuronderzoek	22
7.13.1 beantwoorde deelvragen	22
8.0 Realisatiefase	23
8.1 Verwachte belasting per school	23
8.2 Technisch ontwerp	23
8.2.1 Samenvatting technisch ontwerp.....	23
8.3. Scenario's voor performance zware handelingen	23
8.3.1 Scenario 1.....	24
8.3.2 Scenario 2.....	24
8.3.3 Scenario 3.....	24
8.4. Dashboard voor Elerna administrators.....	25
8.5 Blackbox testen	25
8.6 Conclusies realisatiefase	25
8.6.1 Welke technische keuzes zijn van belang?	25
8.6.2 Wat zijn de verwachte waarden van de belasting van het product?.....	26
9.0 Verificatiefase	27
9.1 JMeter en de implementatie hiervan.....	27
9.2 Testverslag performancetesting.....	27
9.1.1 Performance van sign in use case	28
9.1.2 Performance van database	29
9.1.3 Conclusies verificatiefase.....	30
10. Conclusie en discussie.....	31
10.1 Conclusie.....	31
10.2 Discussie	31
11. Evaluatie en aanbevelingen	32
11.1 Evaluatie.....	32
11.2 Aanbeveling naar Elerna	32
12. Bronnen	32
Bibliografie	32

Bijlage 1 Indexation of different response times	33
Bijlage 2 The knee	35
Bijlage 3 The knee and there utilization values	36
Bijlage 4 Use cases Elna	37
Bijlage 5 Use cases gekoppeld aan responstijden Robert Miller	38
Bijlage 6 Ontwerp managementdashboard Elna	39
Bijlage 7 Datamodel user role student.....	40
Bijlage 8 SQL Query's.....	41
Bijlage 9 Testplannen Jmeter	43
Bijlage 10 Technisch ontwerp	44
Bijlage 11 Blackbox testplan	45

1.0 Inleiding

Performance is een breed begrip en je komt het overal om je heen tegen. Je hoort erover op de werkvloer maar ook tijdens wielervedstrijden en uiteraard hoor je er in de ICT wereld over. Maar wat is het nou eigenlijk? Deze scriptie vertelt meer over performance gezien vanuit het technische aspect. Eerst zal er worden gekeken naar wat andere bekende onderzoekers hierover te zeggen hebben, vervolgens zal dit worden toegepast op de organisatie waar deze opdracht is uitgevoerd en tot slot zullen er een serie testen worden uitgevoerd om het geheel te valideren.

2.0 Organisatie

In dit hoofdstuk wordt het stagebedrijf en het daar werkzame team beschreven. Ook wordt duidelijk welke positie de afstudeerder heeft binnen het bedrijf.

Elerna

Het bedrijf waar deze stage plaats vindt is Elerna, een splinternieuw bedrijf dat in september 2013 is opgericht. Hun doel is het verbeteren van het voortgezet onderwijs door het bieden van digitale hulpmiddelen en lesmaterialen. Het huidige project is het vervangen van papieren werkboeken door een digitale variant.

Het team

Het team bestaat uit een zestal starters die eind augustus gezamenlijk met Elerna zijn begonnen. Het is een klein team en een erg platte organisatie waar goed naar elkaar wordt geluisterd en veel wordt overlegd.

Martijn Versteeg – Communicatie, sales

Martijn Versteeg is de oprichter en mede-eigenaar van Elerna. Hij is het gezicht naar de buitenwereld en handelt het meeste contact met rectoren en docenten af.

Jeroen Hauser – Projectmanager - Bedrijfsbegeleider

Jeroen Hauser heeft aan de Hogeschool Utrecht een Bachelor Digital Communication behaald. Na zijn studie heeft Jeroen zes jaar gewerkt voor zijn bedrijf KiwiMedia. Bij Elerna houdt hij het team scherp en gefocust en zorgt hij dat de organisatie blijft draaien. Hij heeft vier jaar technische programmeerervaring, maar zijn focus ligt tegenwoordig meer bij projectmanagement.

Hayo Bluijs – Functioneel ontwerper

Hayo's taak is het bepalen van de functionaliteiten van het programma en het uittekenen van de eerste wireframes. Hij gaat af en toe met Martijn mee naar gesprekken met toekomstige klanten om informatie in te winnen.

Stefan Postma – Technisch ontwerper (afstudeerder)

Stefan's taak binnen het project is opstellen van het technisch ontwerp van het E-learning product. Ik let hierbij in het bijzonder op de niet functionele eisen van de klant. De focus in het ontwikkelproces zal liggen op toekomstige performance gezien vanuit de leerling.

Zoran Trpevski - Programmeur

Zoran woont in Macedonië en dat is tevens de reden dat een groot deel van de communicatie in het Engels verloopt. Zijn voornaamste taak is het bouwen van de back en front-end van de applicatie.

Micha Vinkesteijn – Communicatie

Micha is ons nieuwste teamlid. Hij helpt Martijn met het maken van afspraken met scholen en is bezig met het opzetten van een communicatieplan voor Elerna.

3.0 Probleemstelling

3.1 De opdrachtomschrijving

Deze opdracht bestaat uit twee onderdelen:

Het onderzoek

De student doet onderzoek naar performance van webapplicaties. Dit onderzoek zal beginnen met een literatuuronderzoek waarin duidelijk wordt welke studies er in het verleden zijn gedaan en relevant zijn voor deze scriptie.

Het technisch ontwerp

De student beschrijft de technische omgeving waarbinnen het web-based product van Elerna ontwikkeld wordt. Deze informatie verzamelt de student in een technisch ontwerp waarvan het belangrijkste criterium is dat het product van Elerna een goede performance blijft bieden tijdens piekmomenten en dat tegelijk alle functionaliteiten uit het functioneel ontwerp geïmplementeerd kunnen worden.

De keuzes gemaakt in het technisch ontwerp komen voort uit de volgende punten:

- Functionele eisen gesteld door de eindgebruikers (Bron: Functioneel ontwerp)
- Reeds gemaakte technische keuzes vanuit het bedrijf
- Eigen onderzoek van de student gericht op performance van webapplicaties

Als ondersteuning en onderbouwing van dit document zal de student een blackbox-testplan opstellen voor de functionele acceptatietest. Concreet zal er met behulp van Jmeter (<http://jmeter.apache.org>) een test worden gemaakt en uitgevoerd waarmee performance onder verwachte belasting kan worden gemeten.

4.0 Analyse van het probleem

4.1 Hoofdvraag en deelvragen

4.1.1 Hoofdvraag

Welke keuzes voor Elerna, uitgewerkt in een technisch ontwerp, zijn nodig voor optimale performance?

4.1.2 Deelvragen

1. Welke definitie voor performance wordt er gehanteerd?
 - a. Welke indicatoren kunnen worden gehanteerd om het begrip 'performance' operationeel te maken?
 - b. Wat zijn relevante parameters die effect hebben op de indicatoren van performance van (a)?
 - i. In bestaande systemen
 - ii. In Elerna
 - c. Welke waardes kunnen worden toegekend aan de gevonden KPI's?
2. Welke technische keuzes zijn van belang om 1c te implementeren?
 - a. In een client/server-omgeving; waar liggen keuzes voor elk van client en server?
 - b. Welke keuzes liggen aan de kant van Elerna?
3. Wat zijn de verwachte waarden van de belasting van het product?
 - a. Waardoor wordt of worden deze waarde(n) beïnvloed?
 - i. Aan de kant van Elerna
 - ii. Aan de kant van de klant
4. Welke gebruikersrollen zijn relevant voor de applicatie?
 - a. Wat betekent dit voor de performance indicatoren?
5. Hoe worden deze keuzes gevalideerd?
 - a. Welke KPI's zijn relevant om te tonen op een dashboard?
 - i. Aan de kant van Elerna
 - ii. Aan de kant van de klant

5.0 Scope

Deze scriptie richt zich op de technische performance van het E-learning pakket van Elerna. Functionele aspecten, keuzes en overwegingen zullen niet worden behandeld in deze scriptie. Alle benodigde functionele aspecten komen uit het functioneel ontwerp. Omdat Elerna zich in de ontwikkelfase bevindt en mijn stage slechts vijf maanden duurt zal de technische uitwerking in deze scriptie zich beperken tot de gebruikersrollen “writer” en “student”. De andere rollen komen pas na mijn scriptie aan bod. Omdat zowel het technisch ontwerp als de scriptie de basis biedt voor verdere ontwikkeling moet dit voldoende informatie zijn om de rest van het pakket verder te ontwikkelen op dezelfde wijze. Het literatuuronderzoek focust zich op het in kaart brengen van het begrip performance en dit meetbaar te maken voor het toekomstige E-learning pakket van Elerna.

6.0 Opbouw en aanpak

Om theoretische verdieping aan te brengen zal eerst de **onderzoeksfase** worden uitgevoerd waarin een literatuuronderzoek wordt gedaan. Tijdens de onderzoeksfase wordt er naar antwoorden gezocht op de volgende twee deelvragen:

- Deelvraag 1: Welke definitie voor performance wordt er gehanteerd?
- Deelvraag 4: Welke gebruikersrollen zijn relevant voor de applicatie?

Na het theoretische kader wordt de **realisatiefase** uitgevoerd waarin de technische keuzes worden uitgewerkt in het technisch ontwerp. Daarna wordt er in deze fase een testplan geschreven. Deze beroepsproducten ondersteunen het beantwoorden van de volgende deelvragen:

- Deelvraag 2: Welke technische keuzes zijn van belang om 1c(H. 4.1.2) te implementeren?
- Deelvraag 3: Wat zijn de verwachte waarden van de belasting van het product?

Om het bovengenoemde te valideren zal de scriptie worden afgesloten met de **verificatiefase** in deze fase zal het testplan opgesteld in de realisatiefase worden uitgewerkt in het testverslag. Dit beroepsproduct heeft ten doel om de volgende deelvraag te beantwoorden:

Deelvraag 5: Hoe worden deze keuzes gevalideerd?

Meer over de aanpak en uitwerking hiervan is terug te vinden in het plan van aanpak (Postma, 2013, pp. 13-18)

Doordat bij het stage bedrijf een verandering is opgetreden in de uitgewerkte gebruikersrollen, zijn er ook een onderdelen gewijzigd in de aanpak namelijk het volgende;

- Er zal een fictief datamodel worden gemaakt voor de actor student
- Er zullen performance zware scenario's worden gemaakt voor de actor student
- Er zullen testen worden gemaakt en uitgevoerd voor de bovengenoemde scenario's

7.0 Onderzoeksfase - Literatuuronderzoek

In het literatuuronderzoek wordt een theoretische verdieping gedaan, waarna er een koppeling wordt gelegd naar het stagebedrijf en het beroepsproduct. In deze verdieping wordt gekeken wat diverse experts op het gebied van software, usability en menselijke psychologie hebben onderzocht en daaruit hebben vastgesteld.

7.1 Aanpak

In het literatuuronderzoek zijn drie onderzoeken uitgewerkt: dat van Carry Millsap, Robert E. Miller en Jakob Nielsen. Deze drie onderzoekers zijn leidend in onderzoeken naar usability, performance en onderzoek naar wachttijden van applicaties. Er is hier gekozen om deze drie onderzoekers te behandelen omdat in veel onderzoek naar performance één van deze personen genoemd worden. Verder staat Jakob Nielsen naast Steve Krug bekend als één van de meest gewaardeerde usability-engineers. Naast het literatuuronderzoek naar performance zal ik nog onderzoek doen naar het belang van ICT, Big Data gebruik bij het toepassen van gepersonaliseerd leren.

7.2 Wat is performance?

Performance is een zeer breed gebruikt begrip. Omdat ik duidelijkheid wil creëren over dit begrip zal in dit hoofdstuk worden uitgewerkt welke definitie van performance ik hanteer voor Elerna.

7.2.1 Performance volgens Cary Millsapp

Cary (Millsap, 2010, p. 2) noemt in zijn stuk “thinking clearly about performance” een mooie uitleg over het begrip performance. Hij geeft aan dat het begrip performance zeer breed wordt gebruikt, van wielrenners ranglijsten tot jaarlijkse werknemers beoordelingen. Er wordt ook een definitie genoemd welke gericht is op software performance: Carry (Millsap, 2010, p. 2) “the time it takes for computer software to perform whatever task you ask it to do.” Deze definitie van performance geeft de mogelijkheid één ding te meten, namelijk tijd welke het kost om de taak uit te voeren, de zogenoemde “**Response time**” gemeten in tijd per taak. Blijkbaar kan er niet één vaste waarde worden aangewezen. Afhankelijk van de taak die de gebruiker uitvoert mag iets langer of korter duren. Daarnaast beschrijft hij in zijn onderzoek ook een tweede groep gebruikers; deze gebruikers zijn niet hoofdelijk geïnteresseerd in de response time maar in de hoeveelheid taken die het systeem binnen een bepaalde tijd kan afhandelen. Hier gebruikt hij de term “**Throughput**” voor. Beide van deze begrippen toepasbaar op de gestelde definitie van Millsapp. Deze begrippen hebben uiteraard een relatie met elkaar. Maar toch kan je response tijd niet afleiden uit throughput en visa versa. Waarom deze beide niet herleidbaar zijn zou blijken uit de onderstaande hoofdstukken.

7.2.2 Performance volgens Robert Miller

(Miller, 1968, p. 271)

Definitions of response time.

Response to human inquiry, with few exceptions, serves as feedback to a continuity of thought. Human behavior occurs at a variety, of information handling levels. Different kinds of response and response delay will be appropriate at different behavior levels. The definitions that follow depend in part for their time estimates on psychological Relations given in Part I of this report.”

Uit dit psychologisch onderzoek van Miller is te halen dat er een bepaald verwachtingspatroon bestaat dat kan worden geschetst bij elke aanvraag die wij doen als persoon, zowel gericht op personen als op systemen. Deze verwachting bestaat uit een vaste tijd die het kost om antwoord te krijgen op een specifieke aanvraag die wij doen. Ook al is de reactie van een mens bijvoorbeeld slechts een “euhh” of “mmmmh” geluid. Robert Miller werkt in zijn onderzoek alle door hem vastgestelde responstijden uit in zijn onderzoek, onderverdeeld per taaksoort.

7.2.3 Performance volgens Jakob Nielsen

Jakob Nielsen (Nielsen, 1993) vertelt in zijn boek “usability engineering” helaas niks over hoe hij het begrip performance interpreteert. Wel vertelt hij ons meer over vastgestelde responstijden. Dit komt later in het literatuuronderzoek aan bod.

7.2.5 Samenvatting

- Performance is een breed begrip en moet altijd gespecificeerd worden
- Performance kan worden opgesplitst in de indicatoren “response time” en “throughput”
- Er zijn vaste verwachtingspatronen voor “response time”
- Van belang is om altijd een reactie te geven binnen een vast gestelde tijd

7.3 Wat is efficiency?

7.3.1 Efficiency volgens Cary Millsap

Om Cary (Millsap, 2010, p. 7) maar direct te citeren: “Efficiency is the inverse of how much of a task execution’s total service time can be eliminated without adding capacity, and without sacrificing required business function. In other words, efficiency is an inverse measure of waste. “

Als ik het bovenstaande citaat lees interpreteer ik dat als volgt: efficiency kan worden behaald door het aantal onnodige handelingen binnen het verwerken van een taak te verminderen of handelingen te verbeteren, waardoor deze minder capaciteit in beslag nemen zonder de uitkomst van het gedane werk te veranderen.

Carry (Millsap, 2010, p. 8) beschrijft tevens het belang van het opsporen van inefficiëntie. Hij maakt onderscheid tussen globale problemen en lokale problemen. Globale inefficiëntie kan bijvoorbeeld optreden door de onderstaande oorzaken:

- Ondoordachte indexering
- Slecht ingestelde parameters
- Slecht geconfigureerde hardware

Lokale efficiency kan bijvoorbeeld optreden door slecht geconfigureerde query’s of slechte filters in de software.

Wanneer lokale inefficiënties worden opgelost heeft dit vaak ook impact op andere niet-gerelateerde softwarepakketten welke op hetzelfde systeem draaien. Dit fenomeen heeft volgens Cary (Millsap, 2010, p. 8) te maken met “**Load upon the system**”

7.3.2 Efficiency volgens Robert Miller

Miller bekijkt efficiency niet vanuit de systeemkant maar vanuit de gebruikerskant. De gebruiker kan efficiënt werken met zijn systeem als zijn verwachtingen in lijn liggen met wat het systeem doet. Wanneer er verandering wordt aangebracht in de efficiëntie van de gebruiker is het van belang te kijken naar de taken die deze gebruiker wil uitvoeren op het systeem. Hier een voorbeeld wat (Miller, 1968, p. 269) geeft:

“The statement that “In the past it took two days to get an answer to a question that now is given in fifteen minutes” means, perhaps, an increase in operating efficiency for the system, but does not in itself materially change the cognitive (psychological) behavior of the person getting the information.”

Dit voorbeeld vertelt ons meer over het cognitieve gedrag van de gebruiker. Voor deze gebruiker is het fijn dat hij het antwoord nu eerder krijgt maar zijn gedrag verandert er niet door. Wanneer ik dus efficiënt response tijden per use case wil indexeren moet ik kijken naar de verwachting van de gebruiker en zijn cognitieve gedrag bij dat proces.

7.3.3 Samenvatting

- Efficiency is van belang voor het optimaliseren van performance
- Inefficiëntie kan op gebruikersniveau en systeemniveau optreden en ook veroorzaakt worden door andere applicaties op het systeem.
- Belasting van het systeem gaat nauw samen met de performance van het systeem
- Zowel de verwachting van de gebruiker als zijn cognitieve gedrag is van belang.

7.4 Wat is Load / utilization?

In dit hoofdstuk probeer ik meer te weten te komen over belasting van systemen. Ook vraag ik mij af wat verschillende onderzoekers hierover zeggen. Wederom heb ik de onderzoekers verdeeld in subhoofdstukken.

7.4.1 Volgens Cary Millsapp

Carry (Millsap, 2010, p. 8) beschrijft load als volgt:

“Load is competition for a resource induced by concurrent task executions”

Cary Millsap leert ons dus dat wordt load veroorzaakt door gelijktijdige taakbewerkingen op een systeem, hoe meten we dit dan? Carry (Millsap, 2010, p. 8) geeft hier slechts één antwoord op: “one measure of load is **utilization**”

Hieruit lijkt te halen dat er nog meer manieren zijn om load te meten, echter dit wordt niet duidelijk uit dit stuk. Utilization kan worden gemeten door belasting ten opzichte van benutting van het systeem te meten over een vastgesteld tijdsbestek.

7.4.2 Volgens Robert Miller

Robert Miller is in zijn paper “Response time in man-computer conversational transactions helaas niet duidelijk over belasting of capaciteit van systemen, zijn onderzoeken richten zich alleen op psychologische vlakken van mensen. Daarom wordt er wel gesproken over de belasting van personen in het stuk Postscript 1 (Miller, 1968, p. 277): hier legt hij uit dat wanneer iemand zeer druk is en hij moet wachten op het systeem dit de persoon dermate demotiveert dat hij mogelijk helemaal niet meer tot werken toekomt. Daarom is volgens Miller van belang de gebruiker op de hoogte te stellen van de tijd die het kost om de taak van de gebruiker af te handelen zodat deze ook iets anders kan ondernemen in die tijd.

7.4.3 Volgens Jakob Nielsen

Jakob Nielsen gaat niet in op dit onderwerp. In het onderdeel 6.7 Performance Measurement (Nielsen, 1993, p. 191) gaat hij alleen in het op meten van de snelheid van de taak van de gebruiker ten opzichte van interface.

7.4.4 Overdenking en toepassing op Elna

Als ik de bovenstaande stukken lees denk ik bij mezelf “hoe ga ik dit toepassen op Elna”? Het hoofdstuk belasting is één van de belangrijkste genoemde onderdelen van dit onderzoek en waarschijnlijk de meest onduidelijke. Als we verder kijken dan de drie onderzoekers kom ik al snel uit op stresstesting en de Ramp-Up waarde. Deze testen zal ik uiteindelijk ook moeten gaan uitvoeren bij Elna. Maar wat maakt het product van Elna nou anders dan andere producten? Elna bouwt een onderwijsproduct welke gebonden is aan vaste lestijden. Dit betekent dat om half negen een school begint en studenten de klas inlopen, gaan zitten en vervolgens binnen de eerste tien minuten zijn ipad pakt en inlogt, waar zij normaal hun boeken op tafel zou leggen. Hier bevindt zich de technische uitdaging van Elna. Om dit moment telkens weer te overwinnen. Hoe meer scholen zich gaan aansluiten bij Elna hoe groter deze piek belasting op de vaste tijden gaat worden.

7.4.5 Samenvatting

- Het is belangrijk om het product onder belasting te testen.
- Er moet een duidelijke capaciteitsplanning worden gemaakt wanneer er uitbreiding van het aantal gebruikers plaatsvindt
- Er moet een waarschuwing worden getoond wanneer de belasting boven normaal is.

7.5 Kendall's notation

Alleen Cary Millsap behandelt in zijn onderzoek de Kendalls's notation. Omdat dit onderdeel interessant kan zijn in de vervolguutwerking van het beroepsproduct, behandel ik ook deze in het literatuuronderzoek.

7.5.1 Volgens Cary Millsapp (Thinking Clearly About Performance)

De relatie tussen belasting en responstijd kan worden uitgedrukt in een wiskundeformule. Deze formule staat ook bekend als Kendall's notation. Deze wiskundige notatie wordt gebruikt om wachttijden van diverse situaties te beschrijven. Wachttijden zijn namelijk direct te linken aan belasting van systemen. Een goed voorbeeld hiervan is de wachtrij bij de kassa in een winkel. Zodra er één kassa geopend is zal de wachtrij worden afgehandeld op first in, first out methode. Veel hedendaagse processors werken ook op deze methode. Kendall's notation simuleert echter een perfect schaalbare situatie. De notatie voor deze situatie is M/M/m. (Millsap, 2010, p. 9) Met deze berekening kunnen we de Response time(R) berekenen door de service time(S) en de Queuing delay(Q) op te nemen in de volgende formule: $R = S + Q$. Omdat deze berekeningen zeer complex zijn en er bijna nooit van een perfect schaalbare situatie uitgegaan kan worden verwijst Carry Millsap in zijn onderzoek naar reeds uitgewerkte waardes.

7.6 The knee

Cary (Millsap, 2010, p. 9) verteld ons meer over het optimaal inrichten van een systeem in het volgende citaat:

“When it comes to performance, you want two things from a system:

- The best response time you can get: you do not want to have to wait too long for tasks to get done.
- The best throughput you can get: you want to be able to cram as much load as you possibly can onto the system so that as many people as possible can run their tasks at the same time.

Unfortunately, these two goals are contradictory. Optimizing to the first goal requires you to minimize the load on your system; optimizing to the second goal requires you to maximize it. You can not do both simultaneously. Somewhere in between at some load level (that is, at some utilization value) is the optimal load for the system. The utilization value at which this optimal balance occurs is called the knee.“

Vervolgens legt Cary Millsap uit:

“The knee is the utilization value for a resource at which throughput is maximized with minimal negative impact to response times. Mathematically, the knee is the utilization value at which response time divided by utilization is at its minimum. One nice property of the knee is that it occurs at the utilization value where a line through the origin is tangent to the response-time curve. On a carefully produced M/M/m graph, you can locate the knee quite nicely with just a straightedge, as shown in Exhibit 9.”

[Exhibit 9 is in bijlage 2 geplaatst]

Carry Millsap beschrijft het fenomeen genaamd “The knee”, de knee treedt op voorspelbare momenten op, afhankelijk van het aantal servicenodes (geopende kassa’s). Zie ook Kendalls notation. Dit is het moment waarop de responstijd en de belasting in theorie optimaal verdeeld zijn. Veel gebruikers kunnen dan gelijktijdig hun taak uitvoeren en het antwoord van deze taak wordt binnen een snelle tijd gegeven. Na dit punt loopt de wachttijd dus ook de responstijd enorm op. Voor de knee word het systeem niet optimaal benut en is er dus sprake van overcapaciteit van het systeem. Omdat het systeem van Elerna op vaste tijden piekbelastingen zou moeten kunnen verwerken is het belangrijk om in kaart te brengen op welk moment welke belastingen optreden, anders zal de software namelijk op piekmomenten slecht presteren of ten alle tijden veel overcapaciteit hebben, wat uiteraard te hoge kosten met zich meebrengt. Carry Millsap heeft een overzicht gemaakt van de belastingwaardes waarop de knee optreedt. Dit overzicht [Exhibit 10] is terug te vinden in bijlage 3. Verder noemt Carry Millsap in zijn stuk nog een aantal punten welke in acht moeten worden genomen.

Carry (Millsap, 2010, p. 10):

“To recap:

- Each of the resources in your system has a knee.
- That knee for each of your resources is less than or equal to the knee value you can look up in table 1. The more imperfectly your system scales, the smaller (worse) your knee value will be.
- On a system with random request arrivals, if you allow your sustained utilization for any resource on your system to exceed your knee value for that resource, then you'll have performance problems.
- Therefore, it is vital that you manage your load so that your resource utilizations will not exceed your knee values.”

Carry Millsap maakt in dit citaat duidelijk dat elk deel van het onderliggende systeem zijn eigen kneewaarde heeft. Ook wordt in deze paragraaf duidelijk dat de waarde gegeven in bijlage 4 af kunnen wijken, omdat ons systeem mogelijk niet perfect volgens de M/M/m principe schaalbaar is. Voor Elerna is het dus van belang bij het maken van een capaciteitsplanning een bijna perfect schaalbaar systeem te bouwen of de “knee values” opnieuw te berekenen. In het derde bovengenoemde punt wordt duidelijk gemaakt dat wanneer de belasting op het systeem “random” is, het van belang is te monitoren of de belasting van de onderliggende onderdelen niet zijn kneewaarde overschrijdt, omdat er anders namelijk zeker performanceproblemen gaan optreden. De uitdaging voor Elerna zit hem in het monitoren van de belasting op de verschillende systeemonderdelen en in kaart te brengen wanneer dit optreedt.

7.7 Wat is snel genoeg?

Om te weten wat de maximale responstijden mogen zijn voor de taken die de leerlingen en schrijvers willen uitvoeren moeten we eerst meer te weten komen over de verschillen in responstijden. Daarom zal er in dit hoofdstuk gekeken worden naar de gewenste responstijden zoals deze worden genoemd in verschillende bekende onderzoeken.

7.7.1 Volgens Cary Millsapp (Thinking Clearly About Performance)

(Millsap, 2010, p. 11):

“The reason the knee value is so important on a system with random arrivals is that these tend to cluster and cause temporary spikes in utilization. These spikes need enough spare capacity to consume so that users don't have to endure noticeable queuing delays (which cause noticeable fluctuations in response times) every time a spike occurs.

Temporary spikes in utilization beyond your knee value for a given resource are OK as long as they don't exceed a few seconds in duration. But how many seconds are too many? I believe (but have not yet tried to prove) that you should at least ensure that your spike durations do not exceed eight seconds.”

Cary Millsap hanteert dus voor alle taken de vrij bekende eight second rule, maar hij geeft heel duidelijk aan dat hij hier geen verder onderzoek gedaan heeft.

7.7.2 Volgens Robert Miller

Miller is duidelijker over wat snel genoeg is. In totaal noemt Miller een overzicht van zeventien verschillende situaties waarin verschillende response tijden mogelijk zijn. Deze situaties onderverdeeld in verschillende “use cases” welke Miller beschrijft zijn opgenomen in bijlage 1. Ik heb een aantal onderwerpen van Miller eruit gelaten omdat deze niet relevant zijn op het huidige product. Latere vervolgonderzoeken op dat van Miller beschrijven ook nog meer over verwachtingen van gebruikers. Één daarvan wordt beschreven in het stuk van (Shneiderman, 1984, p. 273) Hieruit blijkt dat de deviatie in responstijd een belangrijkere trigger is tot ergernis van de gebruiker, naast de daadwerkelijke langere responstijden. Robert Miller is over deze situatie in zijn onderzoek uit 1968 niet erg duidelijk over behalve dat wanneer de gebruiker op de hoogte is van de berekening die het systeem moet doen, hij een langere responstijd zal accepteren. Hij legt hier uit dat wanneer de verwachting van de gebruiker in lijn ligt met het systeem de gebruiker langere response tijden zal accepteren. Voor Elerna kan dit erg belangrijke informatie zijn aangezien de gebruikers dagelijks op het systeem moeten werken en ook dagelijks vrijwel dezelfde handelingen moeten uitvoeren op het systeem. Het gevaar dat Schneiderman (1984) beschrijft komt hier dan al snel om de hoek kijken. Wanneer de gebruiker namelijk de ene dag vrijwel geen vertraging ondervindt en de andere dag op de grens van de maximale wachttijden zit te werken zal hij zeer geïrriteerd reageren op het systeem. Het effect van korte wachttijden vs. maximale wachttijden (standaarddeviatie) zal dus ook zeer goed moeten worden gemonitord .

7.7.3 Volgens Nielsen

Jakob (Nielsen, 1993, p. 135) beschrijft in zijn boek Usability Engineering op bladzijde 135 dat het advies voor responstijden al jaren hetzelfde is. Hij verwijst hier naar het hier gebruikte onderzoek van Miller uit 1968 en een onderzoek van Card uit 1991.

Hij noemt aantal verschillende responstijden met beschrijving:

- 0.1 seconde wordt gezien als het limiet welke gebruikers ervaren als een directe feedback, er is geen andere feedback nodig dan het resultaat.
- 1.0 seconde is het limiet om ononderbroken te kunnen werken, echter de gebruiker ervaart wel vertraging. Normaliter is er geen extra feedback nodig als de informatie wordt getoond in meer dan 0.1 seconden maar binnen 1 seconde, echter de gebruiker ervaart wel dat hij niet direct alle informatie beschikbaar heeft. Er treedt op dit moment een mentale context switch op. Dit zorgt ervoor dat de gebruiker de focus verliest op zijn werk.

10 seconden is zo ongeveer het limiet dat gebruikers willen wachten op hun antwoord. Als de taak langer duurt dan willen gebruikers graag wat anders doen, terwijl ze op het systeem moeten wachten. Feedback is van groot belang tijdens deze taken.

7.7.4 Samenvatting

- De afgelopen jaren is er weinig veranderd op het gebied van acceptabele wachttijden
- 1 seconde is de grens van de mentale context switch, dit blijkt een cruciale waarde te zijn om gefocust te kunnen blijven werken.
- Feedback is bij responstijden boven 1 seconde altijd van belang.
- De standaarddeviatie in responstijden speelt een rol in de verwachting van de gebruikers en kan weleens een cruciale factor spelen in het dagelijks gebruik.

7.8 Welke gebruikersrollen zijn relevant voor Elna vanuit technisch aspect?

In het functioneel ontwerp zijn in de eerste twee sprints de volgende drie rollen beschreven:

- Writer
- Student
- Teacher

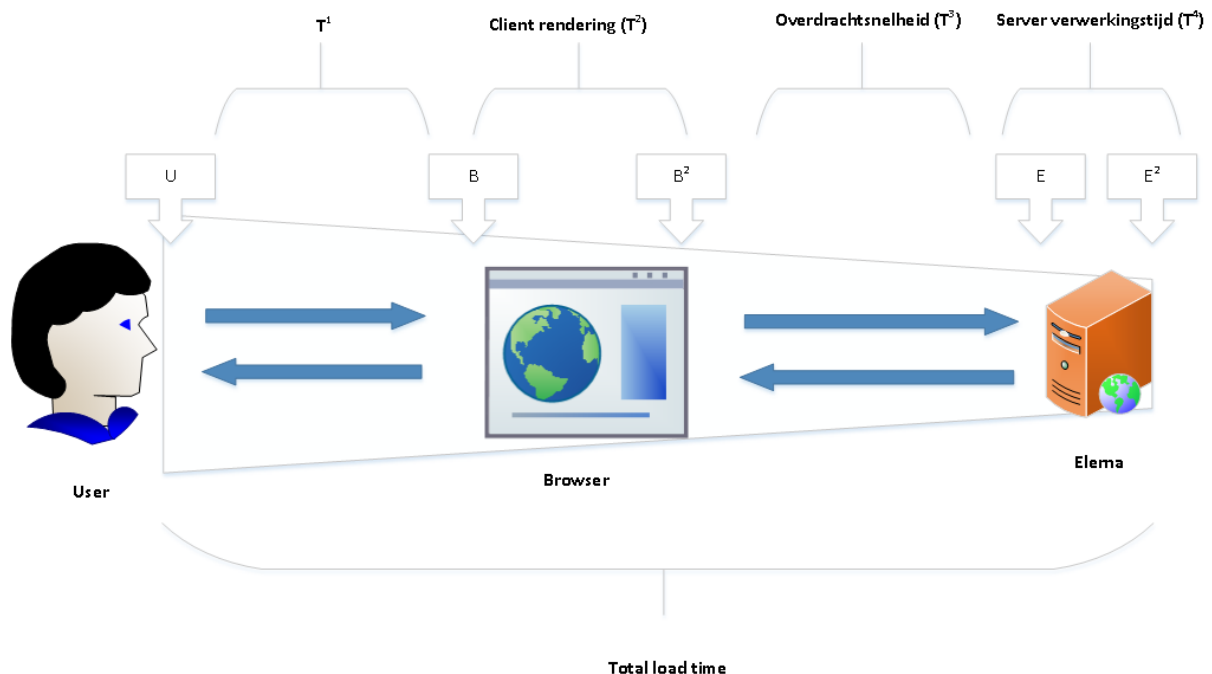
Na het schrijven van het plan van aanpak is er een wijziging doorgevoerd binnen de organisatie van Elna. Het eerste plan was dat scholen zelf digitaal lesmateriaal gingen maken in het pakket van Elna en dit vervolgens zelf gingen verspreiden onder hun leerlingen. De use cases “create chapter” en “edit chapter” zullen dan toekomen aan de user role teacher. In het huidige plan is dit echter gewijzigd, er is een nieuwe rol bijgekomen namelijk de writer. De writer is een losse persoon welke alleen lesmateriaal gaat schrijven. Hij kan eventueel in een team werken met andere writers, maar het systeem maakt hier geen onderscheid in. De studentrol oftewel de leerling in de klas blijft gewoon hetzelfde. Naast de rollen welke beschreven zijn in het functioneel ontwerp is er nog een rol die interessant is voor het technische aspect en dit is de superadmin. Deze rol zal vervuld worden door één van de beheerders van Elna. We kijken in deze scriptie dus naar de volgende rollen:

- Writer
- Student (Key users)
- Teacher
- Superadmin

7.9 Welke parameters zijn van belang voor Elerna?

Het E-learningproduct van Elerna zal op specifieke momenten (bijvoorbeeld het begin van de les) een hoog aantal aanvragen moeten kunnen verwerken binnen een korte tijd, zonder dat dit ten koste gaat van de taak die je als gebruiker aan het verrichten bent. Deze vastgestelde piekmomenten maakt het pakket Elerna anders dan andere web-based applicaties.

Uit diverse onderzoeken blijkt dat response van het systeem van belang is voor het wel of niet goed kunnen uitvoeren van de taken van een gebruiker. Daarom moeten er vaste waarde of waardes gekoppeld worden aan de verschillende taken die de gebruiker uitvoert. Van belang voor de gebruiker is dus de tijd die het kost van het moment dat de gebruiker een aanvraag of handeling doet op het systeem, tot het moment dat hij de reactie hiervan op het scherm ziet. De volgende weergave geeft een schematisch overzicht van de wijze waarop deze response tijd tot stand komt.



De volgende parameters met betrekking tot response time zijn dus van belang:

- Client renderingtijd (T^2)
- Overdrachtsnelheid (T^3)
- Verwerkingstijd server(T^4)

De verwerkingstijd van het scherm, opname in de hersenen en dergelijke zijn weergegevens als T^1 . Deze word als een constante gezien. De waarde is voor de volledigheid wel opgenomen in dit schema.

7.10 Welke waarden zijn te koppelen aan deze parameters?

Afhankelijk van de taken (use cases) die de gebruiker wil uitvoeren kunnen er verschillende maximale responstijden worden gekoppeld. Per use case zal worden bepaald wat de maximale responstijd ($t_1+t_2+t_3+t_4$) zou mogen zijn. In het onderzoek van (Miller, 1968) wordt gesproken over de limiet van het kortetermijngeheugen. Volgens (Miller, 1968) speelt het kortetermijngeheugen een grote rol in het verwerken van informatieve taken. Daarom stelt hij dat bij taken die langer dan **2 seconden** op zich laten wachten een tussenkomst van het tijdelijke geheugen kunnen veroorzaken, hierdoor raakt de gebruiker de focus op de taak kwijt. Naast het onderzoek van Miller beschrijft ook (Nielsen, 1993) een stuk over het verwerken van taken. (Nielsen, 1993) suggereert dat bij een taak waar continue focus nodig is de systeemreactie binnen **1 seconde** moet plaatsvinden, omdat de gebruiker anders zijn focus verliest op zijn taak. Aangezien het stuk van Nielsen een stuk recenter is en hij hetzelfde fenomeen beschrijft, gebruik ik het overschrijden van 1 seconde als de waarde voor het verliezen van continue focus.

Wanneer **continue focus** niet vereist is mag de wachttijd oplopen tot **10 seconden**. Nielsen baseert dit op een uitspraak van (Miller, 1968, p. 268) waarin wordt gezegd: 'a system with response delays of a standard ten seconds will not permit the kind of thinking continuity essential to sustained problem solving'

In bijlage 4 staat een overzicht van de ontworpen use cases van Elerna

In het volgende opgestelde overzicht is een koppeling gemaakt tussen de tabel van Miller, de use cases van Elerna en de continue focus welke Jakob Nielsen beschrijft.

User role	Elerna use case number	Use case description	Taks number according to Robert Miller	Millers task description	Continuous focus required	Maximum response time
All users	-	Laden applicatie	2	Start session	No	3 seconds
	1	Sign in	3	Simple request	No	2 seconds
	2	Retrieve password	10,3	New page, simple request	No	1 and 2 seconds
	3	Edit account	10	New page	No	1 second
Writer	4	Logout	3	Simple request	No	2 seconds
	5	Create chapter	10	New page	Yes	1 second
	6	Arrange chapter	10,1	New page, control activation	Yes	1 second, 100ms
	7	Edit chapter	10	New page	Yes	1 second
	8	Add exercise content block	3,10	Simple request, new page	Yes	1 second
Teacher	9	Upload media	3	Simple request	No	2 seconds
	10	Read student note	10	New page	Yes	1 second
	11	Browse collection	10	New page	Yes	1 second
	12	Assign chapter to classes	3	Simple request	Yes	1 second
	13	Give feedback	10	New page	Yes	1 second
Student	14	Check student exercise	3	Simple request	Yes	1 second
	15	Make exercise	10	New page	Yes	1 second
	16	Add note	10	New page	Yes	1 second
	17	Check own exercises	3	Simple request	Yes	1 second

7.11 Welke kritische performance indicatoren voor Elerna kunnen daaruit worden opgesteld?

Nu bekend is welke taken van de gebruiker welke tijden in beslag mogen nemen kunnen er performance indicatoren worden opgesteld voor het beheer van Elerna. In het volgende overzicht staan de opgestelde KPI's voor Elerna.

Bedrijfsproces / use cases	User role	Kritische Succesfactor	Metrics / KPI naam	Kritieke Prestatie indicator doelstelling	Marge / streefwaarde / target	Balanced Score Card categorie
2,3,5,6,7,8,10,11,13,15,16	All	Performance	Response time to page request	Response time query's in miliseconds	> 100ms = Groen 100 - 1000ms = Geel < 1000 ms red	Back end performance
Volledige proces	All	Performance	Queing delay	Queing delay on server	If any queing = Red	Back end performance
Laden applicatie / session start	All	Performance	Start session time		> 1000ms = Groen 1000-3000.ms = Geel < 3000ms red	Back end performance
General	all	Performance	Amount of Active sessions			Load

7.12 Digitaal gepersonaliseerd leren en big data verzameling

Viktor Mayer-Schönberger en Kenneth Cukier beschrijven in hun E-book Learning With Big Data (Cukier, 2014) de mogelijkheden van Big Data gebruik in het onderwijs.

Big Data zoals gemaakte opgaven geven inzicht op het leerproces, hierdoor kan de docent aansturen en extra informatie leveren. Hieruit kan bijvoorbeeld een heel nieuw inzicht naar voren komen, een goed voorbeeld hiervan staat op bladzijde negen. (Cukier, 2014, p. 9) "in tracking the sequence of video lessons that students see, a puzzling anomaly surfaced. A large fraction of students would progress in order, but after a few weeks of class, around lesson 7, they'd return to lesson 3. Why?" Dit is een goed voorbeeld van informatie die je anders niet zou kunnen weten. Dergelijke informatie is van groot belang als je wilt aansturen op het individuele leergedrag van leerlingen. Voor Elerna zijn ook enkele performance zware scenario's te bedenken voor het ophalen van informatie die de gebruiker graag zou willen weten. Kanttekening hierbij, dat het verzamelen en bewaren van data niet zomaar zonder gevaar kan. In het hoofdstuk Consequences (Cukier, 2014) benadrukken Viktor en Kenneth ook het gevaar van Big Data. Het opslaan van leerlinggegevens voor een langere periode op de verkeerde manier kan grote gevolgen met zich meebrengen. Denk aan een recruiter die schoolstatistieken van zijn sollicitant in handen krijgt. De scenario's moeten dus worden toegepast op het gebruik binnen de schoolomgeving en niet gericht worden op individuele prestaties.

7.13 Conclusies literatuuronderzoek

In de onderzoeksfase zijn de antwoorden gezocht op deelvragen 1, 4 en de subdeelvragen.

7.13.1 beantwoorde deelvragen

Begrip performance zoals ik deze hanteer

Als ik het bovenstaande uiteenzet zijn Carry Millsap en Robert Miller het eigenlijk wel overeen dat wanneer men over performance van een software product praat we altijd kijken naar responstijden. Toch vind ik de aanvulling over “throughput” van Carry Millsap interessant om ook te hanteren aangezien Elerna met veel gebruikers tegelijk op haar systeem gaat werken.

Performance is dus te meten met behulp van de volgende indicatoren:

- Response time
- Throughput

Bestaande systemen hebben vervolgens te maken met de volgende parameters die effect hebben op de bovenstaande indicatoren:

- Taak die de gebruiker uitvoert
- Verwachting van de gebruiker
- Aantal gelijktijdige gebruikers

Elerna heeft nog één speciale aanvulling op de onderste parameter, namelijk het lesrooster van de studenten. Deze bepaalt namelijk veelal de invulling van de onderstaande parameter

Daarom hanteer ik het volgende begrip voor performance:

Het systeem van Elerna vertoont goede performance zodra: De response tijden van het systeem voldoen aan de verwachtingen van de gebruiker, de gebruiker psychologisch ononderbroken gebruik kan maken van het systeem tijdens het uitvoeren van haar taken en het bovengenoemde ook zo is wanneer de verwachte piekbelastingen op het systeem optreden.

Relevante gebruikersrollen voor Elerna

De volgende gebruikersrollen zijn gedefinieerd:

- Writer
- Student (Key users)
- Teacher
- Superadmin

Op dit moment worden volgens de scope alleen de rollen “writer” en “student” uitgewerkt. De koppeling van de waarde die gevonden zijn voor deze gebruikers zijn te vinden in bijlage 5. Op het moment van schrijven is alleen de rol van “writer” nog maar uitgewerkt. Daarom zal de realisatiefase zich richten op een fictieve uitwerking van het datamodel van student en de mogelijkheden om deze in de toekomst te testen op performance.

De studenten rol is de gebruiker waar performance wel een uitdaging kan worden, de schaalgrote is de bepalende factor hierin.

8.0 Realisatiefase

Tijdens de realisatiefase zullen een aantal milestones worden uitgewerkt:

- De technische uitwerking van de use cases (beroepsproduct - technisch ontwerp)
- Opstellen van het testplan (beroepsproduct - testplan)
- Opstellen van een mogelijk databaseplan voor de user role student

Tijdens het uitwerken van deze milestones worden de volgende deelvragen beantwoord:

- Deelvraag 2: Welke technische keuzes zijn van belang om 1c(H. 4.1.2) te implementeren?
- Deelvraag 3: Wat zijn de verwachte waarden van de belasting van het product?

8.1 Verwachte belasting per school

Elerna gaat haar pakket als Software as a Service(SaaS) aanbieden aan scholen. Daarom is het van belang bij elke verkoop te weten hoeveel gebruikers er per verkoop meer gebruik gaan maken van het systeem. Middels deze methode kan er namelijk voor voldoende capaciteit worden gezorgd. Elerna heeft meer onderzoek gedaan naar haar markt tijdens een marktonderzoek. Hieruit is gebleken dat veel scholen met één leerjaar en twee klassen zullen gaan werken met de software. Met deze informatie kan er een capaciteitsplanning worden opgesteld. Met dergelijke informatie kan er bij elke opschaling telkens een nieuwe performance test worden uitgevoerd.

8.2 Technisch ontwerp

In het technische ontwerp zijn de technische keuzes en standaarden gedocumenteerd waaraan het pakket van Elerna moet gaan voldoen. De keuzes gemaakt aan de hand van de gevonden informatie tijdens de onderzoeksfase van deze scriptie en de technische eisen die Elerna van te voren heeft vastgesteld. Daarnaast is er onderzoek gedaan naar de gehanteerde standaarden in onderwijssoftware. Dit document is opgesteld voor de eerste twee sprints welke zij gaat doen.

Het datamodel voor de user role student is ontworpen op basis van mogelijke use cases. Elerna was op het moment van schrijven nog niet zo ver dat dit uitgewerkt was. Daarop heb ik een datamodel gemaakt in SQL waarin fictieve testdata is geplaatst, dit ontwerp zal worden gebruikt om aan te tonen hoe de performance gestest kan worden. Het datamodel is in bijlage 7 geplaatst.

8.2.1 Samenvatting technisch ontwerp

- PHP Framework Laravel
- Gebruik van zowel client-side technieken als server-side technieken
- Database bestaat uit zowel SQL als NoSQL
- Gebruik maken van edustandaarden zoals NL QTI en Kennisnetauthenticatie (entree)

8.3. Scenario's voor performance zware handelingen

Elerna krijgt in de toekomst te maken met een aantal performance zware handelingen. Om de gewenste performance te kunnen testen en te kunnen behalen heb ik een aantal voorbeeld scenario's opgesteld.

8.3.1 Scenario 1

De school wilt aansturen op het leergedrag van de leerlingen door te kijken welke leerlingen goed score bij welke docenten. De eerste stap hierin wordt gezet door het systeem te laten bepalen door te kijken of leerlingen beter presteren door feedback van de docent.

Om deze vraag te kunnen beantwoorden willen we antwoord op de volgende vraag:
Welke docent heeft gemiddeld de meeste feedback gegeven, presteren de leerlingen van deze docent hierdoor ook beter?

Om deze vraag te kunnen beantwoorden willen we de volgende informatie uit onze database halen.

- Welke docenten hebben feedback gegeven
- Hoeveel feedback hebben deze docenten gegeven
- Hoe presteren de leerlingen per docent

Deze vragen worden ondergebracht in SQL query's, welke in de bijlage 8 zijn te vinden.
Dit is een relatief eenvoudige query, maar toch telkens weer performance intensief omdat:

- SQL telkens alle records van de studentenopgave moet doorlopen en deze moet koppelen aan de docenten
- De hoeveelheid data die door de studenten geproduceerd wordt enorm kan groeien
- SQL geen gebruik kan maken van caching door de grote mate waarin de data veranderd.

8.3.2 Scenario 2

Een school wil gepersonaliseerd leren aanbieden voor leerlingen, zodra de leerling inlogt kan hij op basis van het niveau van de lesstof lessen gaan maken.

Hiervoor moet het systeem het makkelijkst te maken hoofdstuk selecteren.

Aannames:

- hoofdstukken zijn onafhankelijk qua moeilijkheid (niet op volgorde)
- makkelijkst' wordt bepaald door middel van scores uit het verleden van studenten.

Om deze vraag de kunnen beantwoorden moeten we de volgende informatie uit onze database halen:

- geef een telling per hoofdstuk van alle incorrecte antwoorden.
- geef een telling per hoofdstuk van alle correcte antwoorden.
- geef een telling per hoofdstuk van alle correct en incorrecte antwoorden

Deze vragen worden ondergebracht in SQL query's, welke in de bijlage zijn te vinden.
Dit is een relatief zware query voor SQL omdat hij een grote verzamel tabel moet creëren waarin alle data uit te lezen is. Daarnaast is dit een scenario wat voor alle studenten uitgevoerd wordt bij het selecteren van een hoofdstuk.

8.3.3 Scenario 3

Docenten willen weten hoe leerlingen presteren om zo aan te kunnen sturen en extra lessen of opgave te geven

Een voorbeeld vraag die hierbij zou passen:

Geef alle studenten die meerdere keren een multiple choice vraag gegokt hebben, gegeven dat de multiple choice vragen drie mogelijke antwoorden hebben.

Om deze vraag de kunnen beantwoorden moeten we de volgende informatie uit onze database halen:

- Geef alle studenten welke multiple choice vragen gemaakt hebben met meer dan 2 pogingen.
- Geef een overzicht van alle studenten die meerdere malen in deze lijst voorkomen, en het aantal vragen wat overeenkomt met bovenstaande.
- Geef de poging waar ze het juiste antwoord hadden geselecteerd.

Deze vragen worden ondergebracht in SQL query's, welke in de bijlage zijn te vinden.

Dit zijn relatief zware query's voor SQL omdat er telkens verbanden moeten worden gezocht tussen de vragen en de studenten. Ook moet SQL telkens de tabel met alle opgaven doorlopen om te controleren wat de relatie tot de vorige vraag is. Omdat de opgaven tabel 131 duizend records omvat verwacht hier zware performance degradatie.

8.4. Dashboard voor Elerna administrators

Vanuit de KPI's opgesteld in hoofdstuk 7.11 heb ik een managementdashboard ontwikkeld, welke te vinden is in bijlage 6. Dit dashboard is onderdeel van het superadmingedeelte van het softwarepakket. Dit dashboard kan live informatie bieden over de huidige belasting en het de waarde gevonden bij de KPI's.

8.5 Blackbox testen

Het E-learning pakket van Elerna dient op verschillende onderdelen en verschillende manieren getest te worden. Twee van die testen zullen door mij opgezet en deels ook worden uitgevoerd en één door mijn duo stagair Hayo Bluijs. Volgens het boek TMap Next resultaatgericht testen (Koomen, TMap Next:) zal er logischerwijs in deze situatie een mastertestplan opgesteld moeten worden. Maar helaas ontbreekt het in de organisatie aan kennis en mankracht om dit op deze manier uit te werken. Daarom zal ik kort uitwerken hoe er getest zal gaan worden. De volgende testen komen tijdens mijn periode bij Elerna aan bod:

Stefan

- Systeemtest (tevens functionele acceptatietest)
- Performancetest (Jmeter, login and database)

Hayo

- Usabilitytest

Systeemtest en performancetest

Als beroepsproduct voor Elerna is een blackbox-testplan opgesteld. Deze zal worden gebruikt om een systeemtest uit te voeren en dit zal tevens de basis zijn om ook een performance test op te stellen met Jmeter. De resultaten van de performancetest zullen uitgewerkt worden in de verificatiefase van de scriptie. De systeemtest van Elerna is alleen voor intern gebruik.

8.6 Conclusies realisatiefase

Tijdens de realisatiefase is antwoord gegeven op de onderstaande deelvragen

8.6.1 Welke technische keuzes zijn van belang?

In het technisch ontwerp is de keuze gemaakt om gebruik te maken van twee database typen namelijk, SQL en NoSQL. Op deze manier kan er gebruik worden gemaakt van het

beste uit twee werelden. Voor het opslaan van hoofdstuk-en collectieobjecten zal gebruikt gemaakt worden van NoSQL en voor relationele data zoals studentenrelaties met vakken en klassen zal gebruikt gemaakt worden van MySQL. Deze keuzes zijn natuurlijk aan het ontwikkelteam van Elerna en de ik geef hier alleen een aanzet van wat mogelijk zou zijn.

Hiernaast wordt er gebruikt gemaakt van veel client side javascript zodat veel interacties met het scherm direct op de client worden verwerkt, hierdoor wordt er een zeer snelle responstijd behaald. Deze gekozen technieken gaan ook goed samen met de platformafhankelijkheid die Elerna graag wil hebben.

8.6.2 Wat zijn de verwachte waarden van de belasting van het product?

Aan de kant van Elerna

De beheerders bij Elerna moeten in goed overleg onderhoud en andere werkzaamheden plannen. Omdat Elerna nu inzichtelijk heeft gemaakt wanneer hoeveel studenten het systeem zullen gebruiken hebben ze de mogelijkheid een goede capaciteitsplanning te maken met behulp van deze scriptie. Door de capaciteit van het systeem goed te kennen kan Elerna database cron jobs en andere zware beheertaken plannen buiten de piekmomenten om zo het systeem optimaal te belasten.

Het aangeleverde datamodel met studentengegevens en test data is een voorbeeld van hoe er getest kan worden of het systeem doet wat het moet doen. Gebaseerd op use cases en scenario's kunnen er soortgelijke testen worden geschreven. De test zoals hij nu is uitgevoerd zou dan de basis kunnen zijn voor de programmeur om zijn database zo te ontwerpen dat deze testen slagen. Ook na uitvoerige updates of aanpassingen aan het systeem kunnen de testen dienen als verificatie of de wijzigingen geen impact op de performance hebben.

Aan de kant van de klant

Aan de klant is het aantal gelijktijdige gebruikers van belang. Zoals te lezen in 8.1 zal er een groot aantal gebruikers gelijktijdig van het systeem gebruik gaan maken. Uiteraard moet de school hier ook op voorbereid zijn. Systemen zoals netwerkcomponenten van de school moeten dergelijke belastingen ook wel aankunnen. Dit ligt echter buiten de scope van deze scriptie. Verder is het belangrijk dat de klant voldoet aan de gestelde eisen met betrekking tot client hardware. De software van Elerna maakt gebruik van diverse client-side technieken om een responsieve interface te geven. De client hardware moet hier wel voldoende mee om kunnen gaan.

9.0 Verificatiefase

In de realisatiefase is een testplan opgesteld, die in bijlage 9 is toegevoegd. Dit testplan word door Elerna straks gebruikt om het product functioneel te testen en deze is nu door mij gebruikt om ook een performancetestplan te maken met JMeter.

Het testverslag richt zich nu slechts op de eerste vier uitgewerkte use cases namelijk:

1. Sign in
2. Retrieve password
3. Edit account
4. Logout

Helaas is het op het moment van schrijven niet mogelijk om andere performance testen uit te voeren op uitgewerkte use cases van Elerna. Daarom heb ik in de realisatiefase een fictief datamodel gemaakt wat soortgelijk zou kunnen zijn als dat wat nu gemaakt wordt door Elerna. Dit datamodel heb ik vervolgens gebruikt om de door mij opgestelde scenario's op te voeren. Dit moet voldoende informatie leveren aan Elerna of aan andere ontwikkelaars om een product te maken welke een goede performance levert.

9.1 JMeter en de implementatie hiervan

Er is gekozen om een performance simulatie te maken met JMeter. Tijdens het schrijven van mijn plan van aanpak kreeg ik al direct de tip van mijn begeleider om JMeter serieus te overwegen als tool om het pakket te testen. Er volgt nu een korte introductie voor de lezers die er niet mee bekend zijn.

JMeter is een product van de Apache foundation en kan worden gebruikt om realtime stress testen uit te voeren op diverse omgevingen waaronder webapplicaties. Het pakket biedt een eenvoudige grafische interface aan om een testplan te bouwen en daarbij diverse tools om de resultaten hiervan te verwerken. Jmeter is een veelgebruikte tool om load en stresstesten uit te voeren. Hierdoor is de community hierachter zeer groot en is er veel en gemakkelijk informatie over te vinden. In deze situatie heb ik een tweetal JMeter testplannen opgezet op basis van opgestelde testcases uit het blackboxtestplan en op basis van de gemaakte scenario's en het bijbehorende datamodel. In bijlage 9 zijn deze testplannen getoond.

9.2 Testverslag performancetesting

De performance test is gedraaid met de bovenstaande testplannen en met de volgende settings:

Use cases:

Aantal Threads: 120

Ramp-Up time: 60

Database

Scenario 1

Aantal threads: 20

Ramp-Up time 1

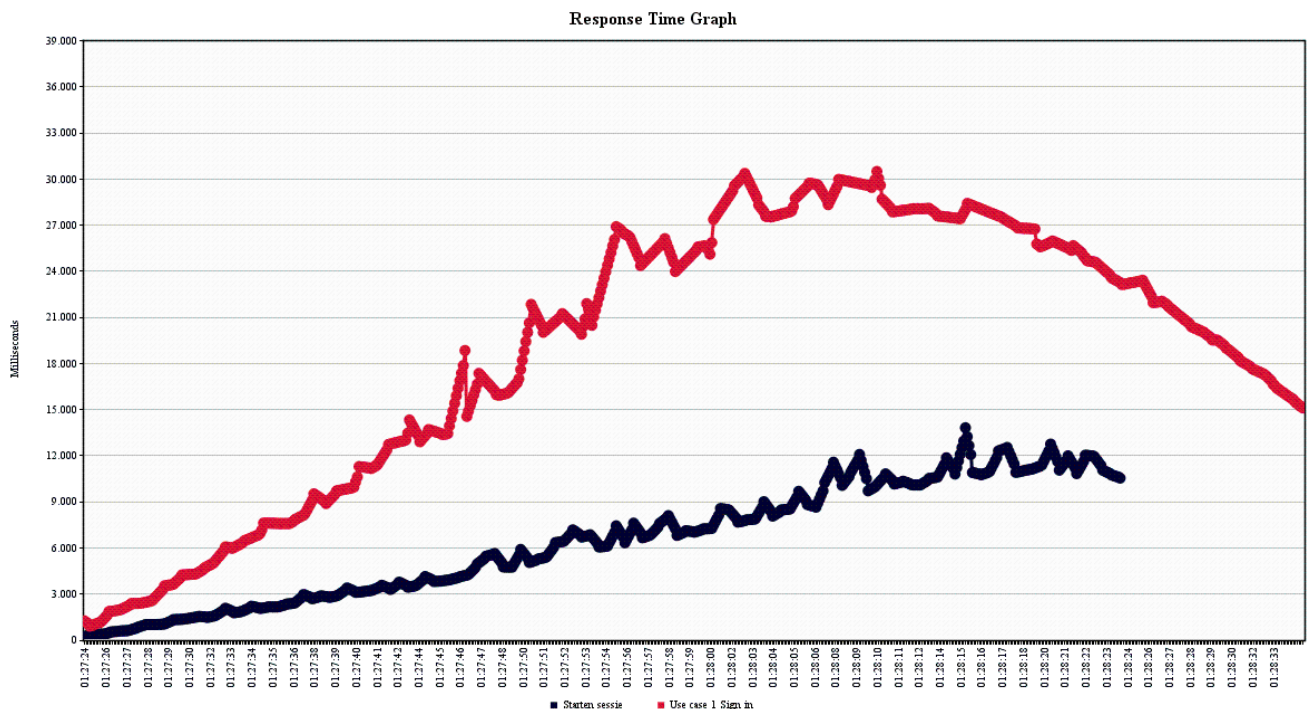
Scenario 2 en 3
Aantal threads: 200
Ramp-Up time 10

Het aantal threads duidt het aantal sessies (gebruikers) aan welke JMeter gaat simuleren. De ramp-up waarde is een instelling welke een gebruikt wordt om een zo reëel mogelijke situatie na te bootsen. Niet alle leerlingen loggen namelijk exact gelijktijdig in, maar bijvoorbeeld over een looptijd van één minuut (60 seconden) zoals hier nu ingesteld.

Verder is het testsysteem (MySQL server) als volgt ingesteld
CPU 2,2GHz dual core
4 GB RAM geheugen
SSD harddisk

9.1.1 Performance van sign in use case

Uit een test met de bovenstaande parameters blijkt ons login systeem als snel een langere responstijd te gaan geven. Hieronder de responstijd van zowel het starten van de sessie als het inloggen.



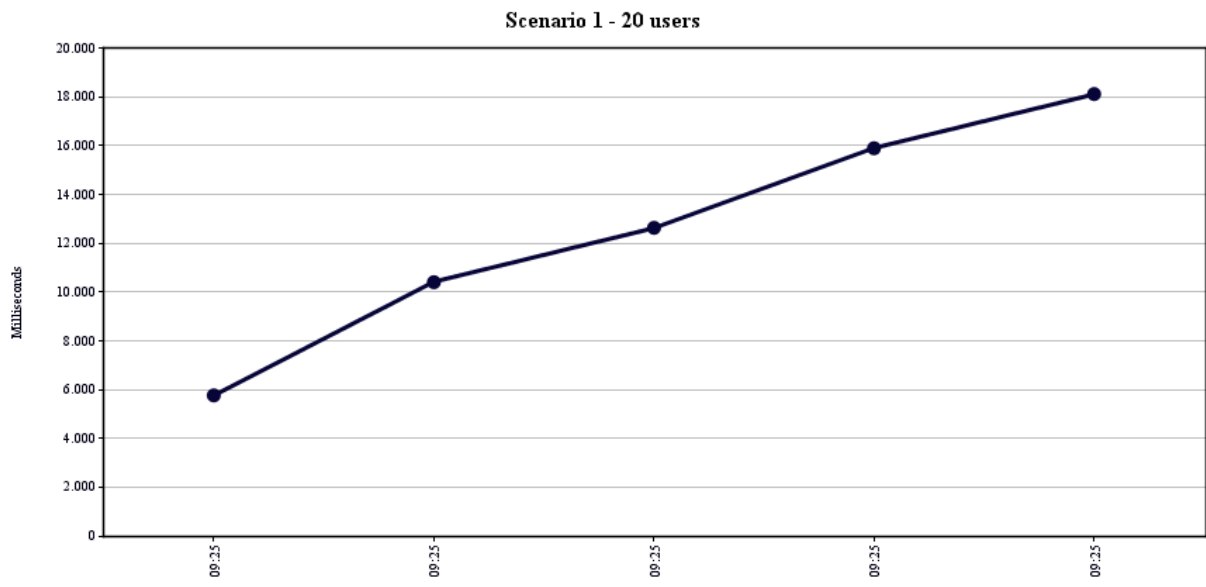
De rode lijn simuleert de responstijd over de loop van de tijd van het inlogproces. En de donkere lijn laat ons zien hoelang het laden van de applicatie duurt voor nieuwe gebruikers. Bijbehorende bij deze grafiek levert JMeter ons de volgende tabel.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Starten sessie	120	6447	245	13827	3766,14	0,00%	1,7/sec	11,73	7018,6
Use case 1 Sign in	120	18744	906	30860	8661,66	0,00%	1,4/sec	22,85	16569,4
TOTAL	240	12595	245	30860	9077,76	0,00%	2,8/sec	32,44	11794,0

9.1.2 Performance van database

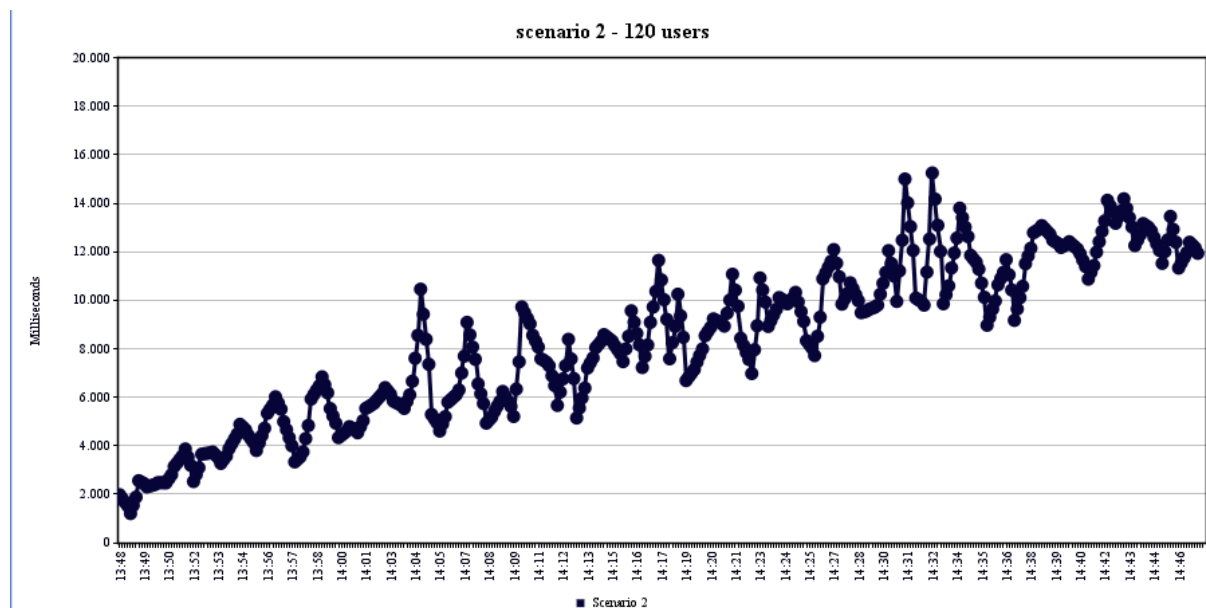
Uit de database test komen interessante uitslagen naar voren. Scenario 1 welke bedacht was als een scenario wat toepasbaar zou zijn op elke leerling laat al gauw een zwakte van SQL zien, namelijk de tijd die het SQL kost om de gewenste gegevens samen te voegen tot een leesbare tabel. Zodra 5 gebruikers gelijktijdig de server om het makkelijkste hoofdstuk vragen. Doet mijn testsysteem er al ruim 8 seconden over om antwoord te geven. Omdat 8 seconden wel de grens is van hoelang een handeling mag duren, zal hiervoor een andere oplossing bedacht moeten worden. De aangeleverde testen zijn een ideale opstap om dit verder te ontwikkelen.

Uitkomsten scenario 1



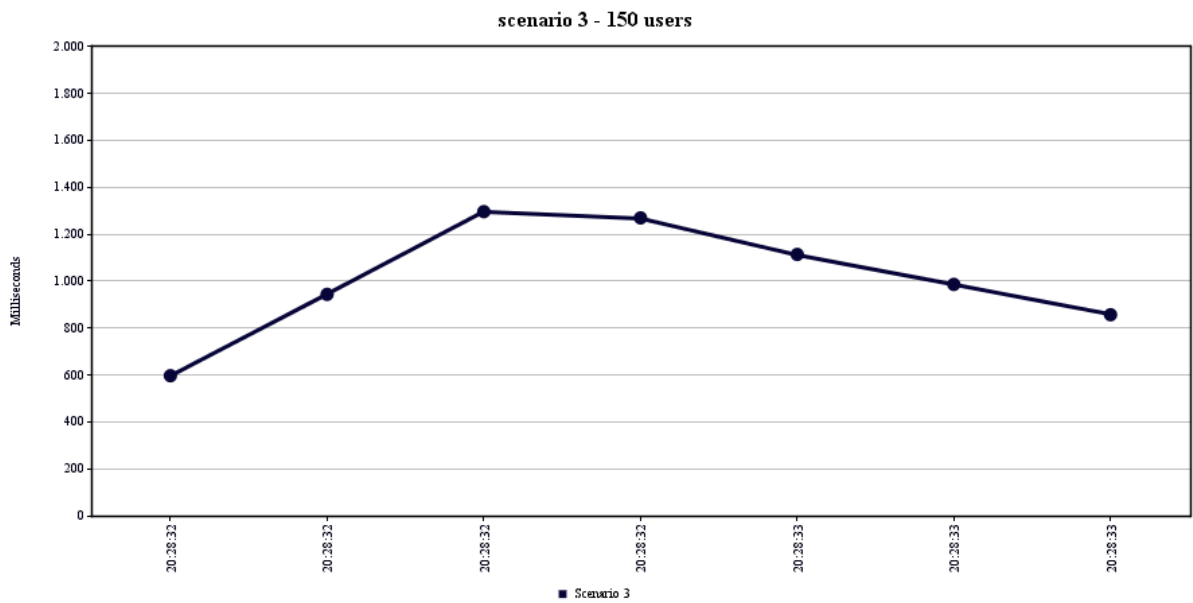
Scenario 1, toont een relatief nette curve, opvallend is dat na ongeveer 4 aanvragen in de wachtrij de server al de limiet van 8 seconden overschrijdt.

Uitkomsten scenario 2



Scenario 2 is met 150 gebruikers gelijktijdig binnen 1 minuut gedaan. Ook hier zien we de reponsetijden oplopen, opvallend zijn die pieken naar hoge response tijden. Deze query is uit enkele verschillende query's opgebouwd. Het lijkt erop dat de server de ene query een stuk sneller afhandelt dan de andere waardoor de pieken en dalen ontstaan.

Uitkomsten scenario 3



Scenario 3 heeft het systeem nauwelijks moeite mee. Blijkbaar kost het weinig moeite om veel data uit 1 tabel door te nemen. Opvallen wel is dat wanneer de aanvragen opstapellen we die response tijden snel zien oplopen. Daarna een piekmoment bereiken en daarna geleidelijke daling.

9.1.3 Conclusies verificatiefase

Tijdens de verificatiefase blijkt toch dat onder een hoge belasting de responstijd enorm oploopt en uiteindelijk de voorgestelde waarde overschrijdt. Het huidige testsysteem voldoet bij lange na niet aan de verwachtingen. Halverwege de test stapelt het aantal aanvragen dermate op dat de heilige 8 seconden regel al wordt overschreden. In de opgestelde tabel heb ik gezegd dat dit proces binnen 3 seconden voor elke gebruiker moet kunnen worden uitgevoerd. De bijgeleverde testen kunnen een goede hulp zijn om het systeem hier op voor te bereiden. De testsituatie waarin deze testen zijn uitgevoerd kunnen een rol spelen in de uitkomst van de testen. In mijn aanbeveling zal ik terugkomen op hoe dit verholpen kan worden.

10. Conclusie en discussie

10.1 Conclusie

In deze scriptie wordt een onderzoek gedaan naar performance gezien vanuit de gebruiker. Er zijn relatief veel onderzoeken verricht naar performance en dan met name performance verbetering. Toch duiden al deze onderzoeken toch vaak weer op dezelfde punten, namelijk de verhouding tussen “Throughput” en “Response time” of zoals Robert Miller het ook verwoord: “Je wilt twee dingen van een systeem: De snelste responstijd, want je wilt niet wachten en de meeste doorvoer, want je wilt dit met zoveel mogelijk gebruikers tegelijk doen.” Zonder een enorme overcapaciteit te creëren zullen er dus ergens keuzes gemaakt moeten worden. Bij Elerna moet deze keuzes gemaakt worden door een goede afweging van client-side en server-side technieken te maken en een degelijke capaciteitsplanning op te stellen. Deze gemaakte keuzes kunnen daarna worden gevalideerd door soortgelijke testen op te stellen zoals hier zijn uitgevoerd. Enkele van deze testen zijn op dit moment ook bruikbaar om te gebruiken. Om leerlingen met continue focus te kunnen laten studeren moet het systeem aan hoge performance eisen kunnen voldoen. Jmeter is een uitstekende tool om te testen of er aan deze standaarden voldaan wordt.

10.2 Discussie

Elk onderzoek naar performance geeft uiteraard mogelijkheid tot discussie. Er is in tijdens het schrijven van deze scriptie ook gekeken naar de discussies welke reeds gevoerd zijn over de onderzoeken die hier gebruikt zijn. Één van de discussiepunten wil ik toch graag citeren in mijn scriptie namelijk:

(Miller, 1968, p. 270) “Therefore, the critic will ask: “Isn't a response time of 30 seconds or even an hour, better than 24 hours? If so, why isn't it good enough?” The answer must be, “Yes, 30 seconds is better than 24 hours for some purposes, but it is not good enough to maintain the continuity of human thought processes.” Where discontinuities in human thought processes are irrelevant or unimportant, both to effective problem solving and to effective use of the professional's time, then the conversational mode is beside the point.” Robert Miller geeft hier direct antwoord op een reactie die hij al verwachtte op zijn artikel. Het interessante aan zijn antwoord is de uitleg over “the continuity of human thought processes” het wel of niet overschrijden van de mentale context switch is eigenlijk de enige discussie die er is te voeren over zijn onderzoek. Tweede discussie punt die er mogelijk bij deze scriptie naar voren komt is de overweging om zelf een datamodel te gaan schrijven voor de student role. Ik heb bewust gekozen om dit model op te stellen om de scriptie en het onderzoek voldoende diepgang te geven. Met behulp van het opgestelde datamodel heb ik scenario's en testen kunnen opstellen welke vrij gemakkelijk te zijn herschrijven naar een eventueel toekomstig datamodel.

11. Evaluatie en aanbevelingen

11.1 Evaluatie

Als ik terugkijk op mijn stageperiode bij Elerna en had geweten wat ik nu wist, dan had ik de opzet van voor af anders gedaan dan nu het geval is geweest. Ik denk tevens dat het bedrijf nog niet ver genoeg ontwikkeld was bij de aanvang van mijn stage. Dit heeft ertoe geleid dat ik veel andere zaken (use cases) eerst moest uitzoeken of oplossen alvorens aan mijn eigen project verder te kunnen. Jeroen Hauser en Roelant Ossewaarde hebben mij hierin wel ten alle tijde gesteund, waarvoor mijn dank. Ik heb tijdens deze afstudeerperiode ontzettend veel geleerd en kennis gemaakt met technieken en tools waar ik voor mijn afstuderen nog nooit van gehoord had.

11.2 Aanbeveling naar Elerna

Allereerst wil ik Elerna aanbevelen om de door mij gevonden waardes welke gekoppeld zijn aan de use cases serieus te gebruiken. Ik ben van mening dat deze wel eens cruciaal kunnen zijn voor het succes van het product. Verder adviseer ik een capaciteitsplanning op te stellen voor de benodigde hardware per aangesloten klant. Het inzichtelijk maken van verwachtingen van gebruikers is en zal ook altijd een moeilijke taken blijven, zeker in het onderwijs. Ik hoop jullie dan ook een zet in de juiste richting gegeven te hebben met dit onderzoek.

Als aanbeveling op de uitgevoerde testen adviseer ik de opzet van een afgesloten test omgeving. In de bovenstaande database testen heb ik een setup gemaakt van 2 systemen met een eigen netwerk ertussen. Hierdoor worden mijn testen niet beïnvloed door andere gebruikers op het netwerk danwel de snelheid van de internet verbinding. De testen kunnen gebruikt worden voor de fijn tuning en ontwikkeling van de database. Dat wilt zeggen, door gebruik te maken van de testen kan ik experimenten uitvoeren met bijvoorbeeld indexeren of caching op tabellen. Met deze methodes zal er waarschijnlijk een flinke performance winst gehaald kunnen worden.

12. Bronnen

Bibliografie

- Cukier, V. M.-S. (2014). *Learning With Big Data*. New York: Houghton Mifflin Harcourt Publishing Company.
- Miller, R. B. (1968). Response time in man-computer conversational. *AFIPS Fall Joint Computer Conference Vol. 33*.
- Millsap, C. (2010). *Thinking Clearly About Performance*.
- Naeghel, K. d. (2014, 5 31). *www.freewebs.com*. Opgehaald van *www.freewebs.com*: <http://www.freewebs.com/koendenaeghel/giscorrectie.pdf>
- Nielsen, J. (1993). *Usability Engineering*.
- Postma, S. (2013). *Plan van Aanpak*.
- Shneiderman, B. (1984). *Response Time and Display Rate in Human Performance with Computers*.

Bijlage 1 Indexation of different response times

Use case according to Miller	Maximum response	notes
Topic 1. Response to control activation	No more than 0.1 second	Bijvoorbeeld toetsaanslagen
Topic 2. Response to "System, are you listening?"	Up to three seconds	starten van een sessie
Topic 3. Response to "system can you do work for me?"	Simple request: within 2 seconds Complex request 5 seconds Load program: within 15 seconds	
Topic 6. Response to "Here I am, what work should I do next?"	If the student has completed a segment of study and wishes to continue into another. topic, the delay should be less than five seconds.	
Topic 7. Response to simple inquiry of listed information	If a terminal is frequently used by an employee for this kind of inquiry (say, more than once an hour), the response should be within two seconds	Opvragen van simpele informatie
Topic 8. Response to simple inquiry of status	in principe 2 seconden. Maar als de users het proces begrijpt kan het worden opgerekt van 7 tot 10 seconden	Opvragen van status of huidige voorraad
Topic 9. Response to complex inquiry in tabular form	Within four seconds	2 seconden wordt in alle gevallen geprefereerd
Topic 10. Response to request for next page	Time delay should be no more than one second until (at least) the first several lines of text on the new page appears	Delays of longer than one second will seem intrusive on the continuity of thought
Topic 11. Response to "Now run my problem."	If the result is returned to him within 15 seconds, he may remain at the terminal "in the problem-	We assume he wants To see an "answer" before he tries another hypothesis

	solving frame of mind"	
Topic 14 Response to complex inquiry in graphic form	The graphical response should begin within two seconds and certainly be completed within ten seconds	If the user is to maintain thought continuity in an ongoing task
Topic 17. Response to "Execute this command into the operational system."	the user should be informed by the system within four seconds that it has understood and can interpret the command, its' execution and final confirmation to the user that the command has been executed may have long and variable delays of minutes	

Bijlage 2 The knee

Bron: (Millsap, 2010)

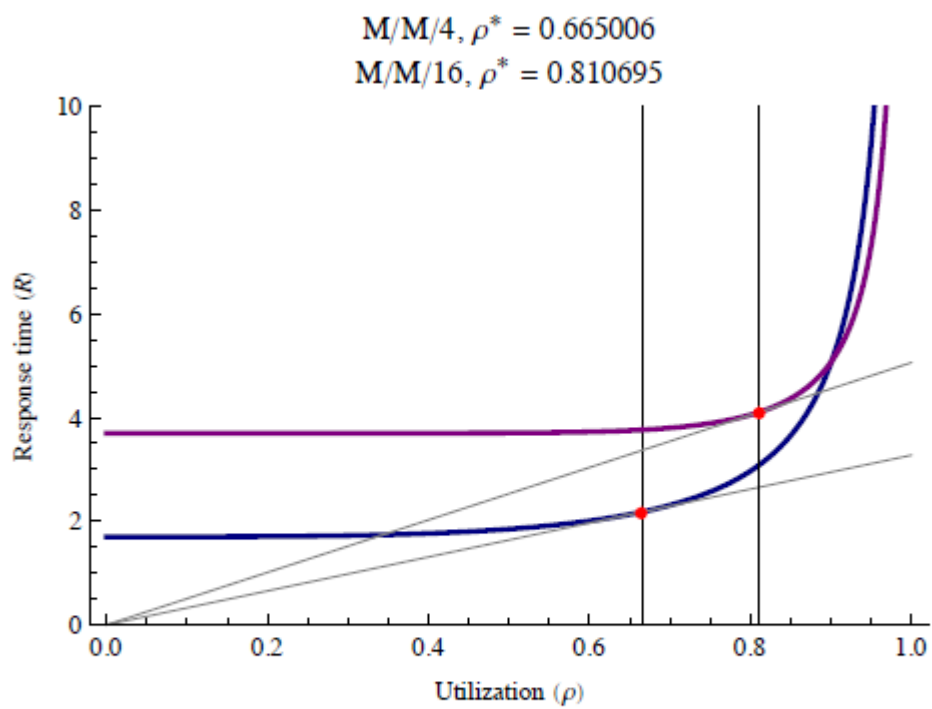


Exhibit 9. The knee occurs at the utilization at which a line through the origin is tangent to the response time curve.

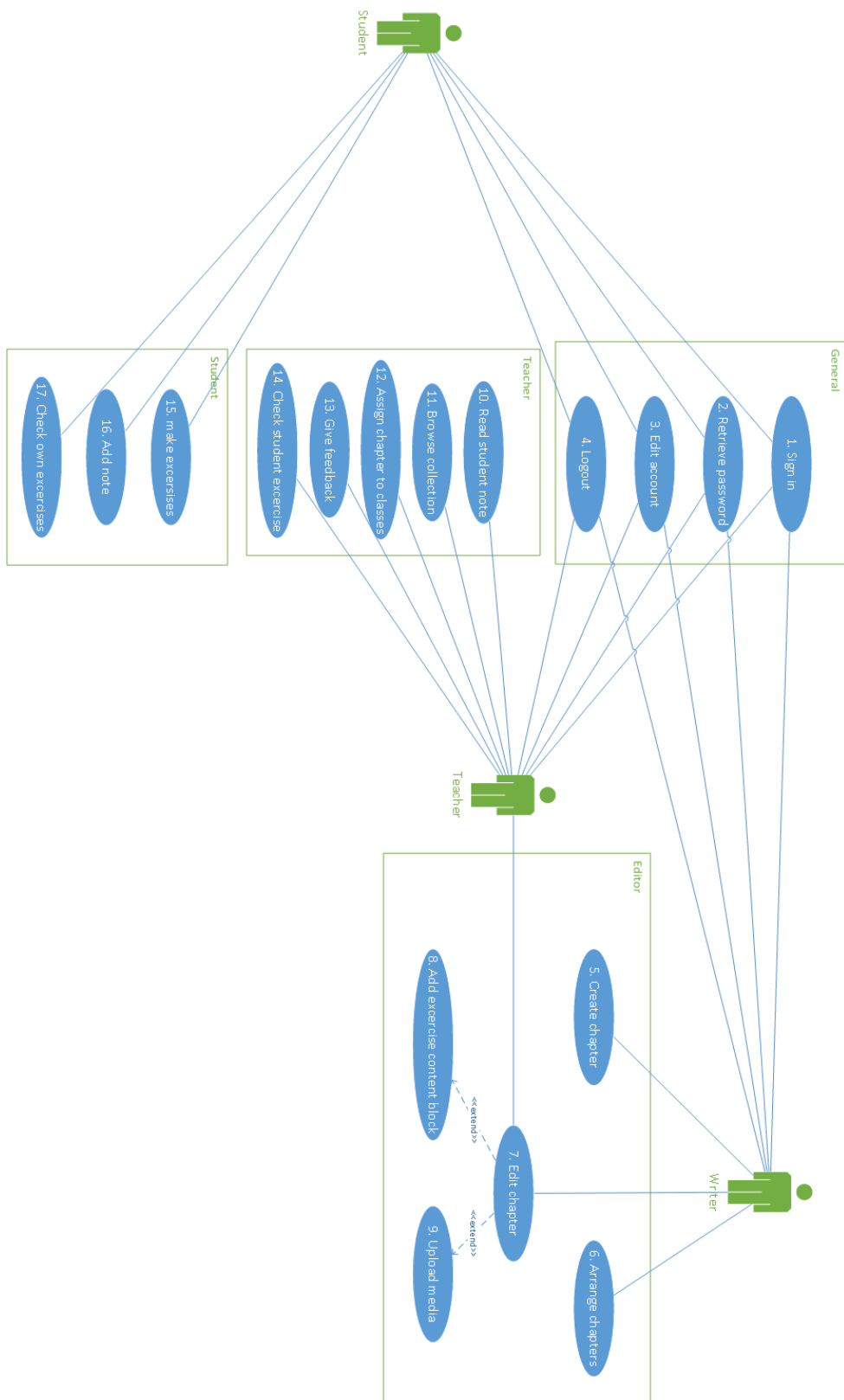
Bijlage 3 The knee and there utilization values

Bron: (Millsap, 2010)

Service channel count	Knee utilization
1	50%
2	57%
4	66%
8	74%
16	81%
32	86%
64	89%
128	92%

Exhibit 10. M/M/m knee values for common values of m.

Bijlage 4 Use cases Elnerna



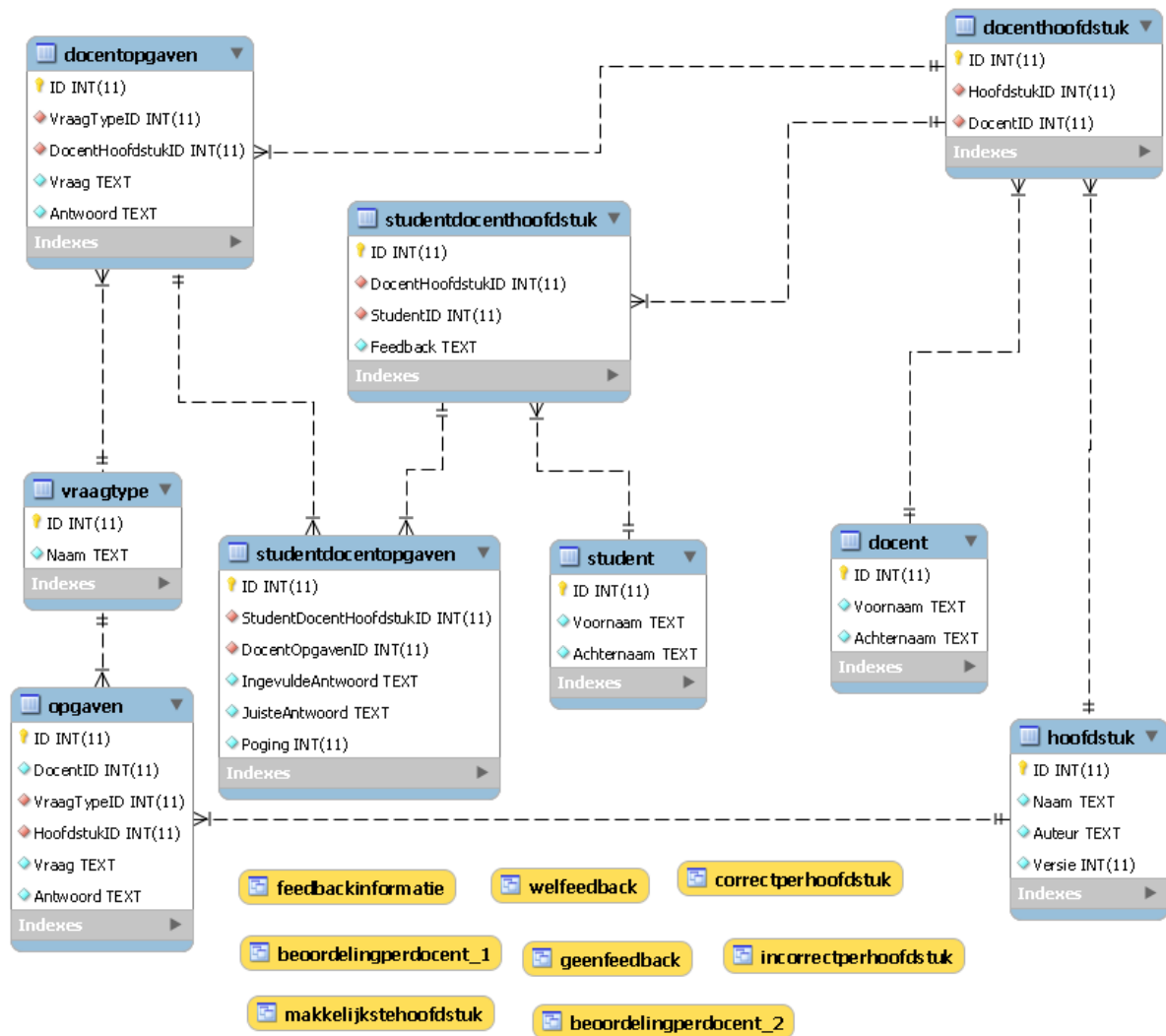
Bijlage 5 Use cases gekoppeld aan responstijden Robert Miller

User role	Elerna use case number	Use case description	Taks number acording to Robert Miller	Millers task description	Contingues focus required	Maximum response time
All users	-	Laden applicatie	2	Start session	No	3 seconds
	1	Sign in	3	Simple request	No	2 seconds
	2	Retreive password	10,3	New page, simple request	No	1 and 2 seconds
	3	Edit account	10	New page	No	1 second
	4	Logout	3	Simple request	No	2 seconds
Writer	5	Create chapter	10	New page	Yes	1 second
	6	Arrange chapter	10,1	New page, control activation	Yes	1 second, 100ms
	7	Edit chapter	10	New page	Yes	1 second
	8	Add excercise content block	3,10	Simple request, new page	Yes	1 second
	9	Upload media	3	Simple request	No	2 seconds
Teacher	10	Read student note	10	New page	Yes	1 second
	11	Browse collection	10	New page	Yes	1 second
	12	Assign chapter to classes	3	Simple request	Yes	1 second
	13	Give feedback	10	New page	Yes	1 second
	14	Check student excercise	3	Simple request	Yes	1 second
Student	15	Make excersise	10	New page	Yes	1 second
	16	Add note	10	New page	Yes	1 second
	17	Check own excercises	3	Simple request	Yes	1 second

Bijlage 6 Ontwerp managementdashboard Elerna



Bijlage 7 Datamodel user role student



Bijlage 8 SQL Query's

Scenario 1

```
5 VIEW `welfeedback` AS
6 select
7   `docent`.`ID` AS `ID`,
8   count(`studentdocenthoofdstuk`.`ID`) AS `hoofdstukkenwelfeedback`,
9   sum(length(`studentdocenthoofdstuk`.`Feedback`)) AS `lengtefeedback`
10 from
11   ((`docenthoofdstuk`
12   join `studentdocenthoofdstuk` ON ((`studentdocenthoofdstuk`.`DocentHoofdstukID` = `docenthoofdstuk`.`ID`)))
13   join `docent` ON ((`docenthoofdstuk`.`DocentID` = `docent`.`ID`)))
14 where
15   (`studentdocenthoofdstuk`.`Feedback` <> '')
16 group by `docent`.`ID`
17 order by `hoofdstukkenwelfeedback`

5 VIEW `geenfeedback` AS
6 select
7   `docent`.`ID` AS `docentID`,
8   count(`docenthoofdstuk`.`DocentID`) AS `hoofdstukkengeenfeedback`
9 from
10   ((`docent`
11   join `docenthoofdstuk` ON ((`docenthoofdstuk`.`DocentID` = `docent`.`ID`)))
12   join `studentdocenthoofdstuk` ON ((`studentdocenthoofdstuk`.`DocentHoofdstukID` = `docenthoofdstuk`.`ID`)))
13 where
14   (`studentdocenthoofdstuk`.`Feedback` = '')
15 group by `docenthoofdstuk`.`DocentID`

5 VIEW `beoordelingperdocent_1` AS
6 select
7   `studentdocenthoofdstuk`.`StudentID` AS `studentID`,
8   `docenthoofdstuk`.`DocentID` AS `docentID`,
9   `studentdocentopgaven`.`ID` AS `opgaveID`,
10   (case
11     when (`studentdocentopgaven`.`JuisteAntwoord` = `studentdocentopgaven`.`IngevuldeAntwoord`) then 'goed'
12     else 'fout'
13   end) AS `beoordeling`
14 from
15   ((`studentdocentopgaven`
16   join `studentdocenthoofdstuk` ON ((`studentdocenthoofdstuk`.`ID` = `studentdocentopgaven`.`StudentDocentHoofdstukID`)))
17   join `docenthoofdstuk` ON ((`docenthoofdstuk`.`ID` = `studentdocenthoofdstuk`.`DocentHoofdstukID`)))

5 VIEW `beoordelingperdocent_2` AS
6 select
7   `beoordelingperdocent_1`.`docentID` AS `Docentid`,
8   sum((`beoordelingperdocent_1`.`beoordeling` = 'fout')) AS `aantalfouteantwoorden`,
9   sum((`beoordelingperdocent_1`.`beoordeling` = 'goed')) AS `aantalgoedeantwoorden`,
10   count(`beoordelingperdocent_1`.`beoordeling`) AS `totaalaantalvragen`
11 from
12   `beoordelingperdocent_1`
13 group by `beoordelingperdocent_1`.`docentID`
```

Scenario 2

```

5 VIEW `correctperhoofdstuk` AS
6   select
7     `studentdocenthoofdstuk`.`DocentHoofdstukID` AS `DocentHoofdstukID`,
8     count(0) AS `aantalgoedeantwoorden`
9   from
10    (((`studentdocentopgaven`
11      join `studentdocenthoofdstuk`)
12      join `docenthoofdstuk`)
13      join `hoofdstuk`)
14   where
15    ((`docenthoofdstuk`.`HoofdstukID` = `hoofdstuk`.`ID`)
16      and (`studentdocenthoofdstuk`.`DocentHoofdstukID` = `docenthoofdstuk`.`ID`)
17      and (`studentdocentopgaven`.`StudentDocentHoofdstukID` = `studentdocenthoofdstuk`.`ID`)
18      and (`studentdocentopgaven`.`Inge vuldeAntwoord` = `studentdocentopgaven`.`JuisteAntwoord`))
19   group by `studentdocenthoofdstuk`.`DocentHoofdstukID`

5 VIEW `incorrectperhoofdstuk` AS
6   select
7     `studentdocenthoofdstuk`.`DocentHoofdstukID` AS `DocentHoofdstukID`,
8     count(0) AS `aantalfouteantwoorden`
9   from
10    (((`studentdocentopgaven`
11      join `studentdocenthoofdstuk`)
12      join `docenthoofdstuk`)
13      join `hoofdstuk`)
14   where
15    ((`docenthoofdstuk`.`HoofdstukID` = `hoofdstuk`.`ID`)
16      and (`studentdocenthoofdstuk`.`DocentHoofdstukID` = `docenthoofdstuk`.`ID`)
17      and (`studentdocentopgaven`.`StudentDocentHoofdstukID` = `studentdocenthoofdstuk`.`ID`)
18      and (`studentdocentopgaven`.`Inge vuldeAntwoord` <> `studentdocentopgaven`.`JuisteAntwoord`))
19   group by `studentdocenthoofdstuk`.`DocentHoofdstukID`

5 VIEW `makkelijkstehoofdstuk` AS
6   select
7     `correctperhoofdstuk`.`DocentHoofdstukID` AS `DocentHoofdstukID`,
8     `incorrectperhoofdstuk`.`aantalfouteantwoorden` AS `aantalfouteantwoorden`,
9     `correctperhoofdstuk`.`aantalgoedeantwoorden` AS `aantalgoedeantwoorden`
10  from
11    (`correctperhoofdstuk`
12     join `incorrectperhoofdstuk` ON ((`incorrectperhoofdstuk`.`DocentHoofdstukID` = `correctperhoofdstuk`.`DocentHoofdstukID`

```

Scenario 3

```

1 • select
2   studentID, count(studentdocentopgaven.ID) as aantalopgaven
3 from
4   studentdocentopgaven
5   inner join
6     studentdocenthoofdstuk ON studentdocenthoofdstuk.ID = studentdocentopgaven.StudentDocentHoofdstukID
7   inner join
8     docentopgaven ON docentopgaven.id = studentdocentopgaven.DocentOpgavenID
9 where
10  VraagTypeID = 2 and Poging > 1
11 group by studentID

```

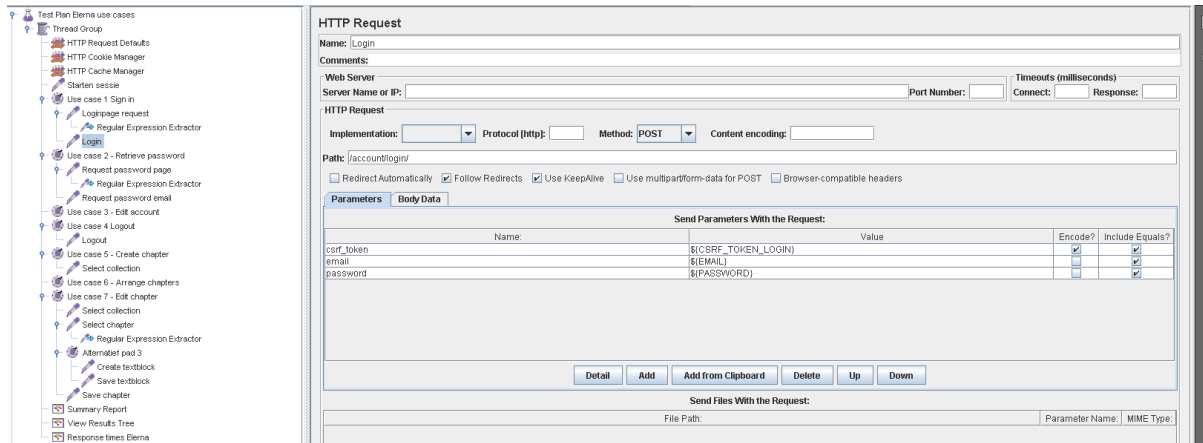
```

1 • select
2   studentdocentopgaven.ID as opgavennummer,
3   studentID,
4   Poging,
5   (CASE
6     WHEN
7       JuisteAntwoord = Inge vuldeAntwoord
8     THEN
9       'antwoord goed'
10    ELSE 'antwoord fout'
11  END) as beoordeling
12 from
13   studentdocentopgaven
14   inner join
15     studentdocenthoofdstuk ON studentdocenthoofdstuk.ID = studentdocentopgaven.StudentDocentHoofdstukID
16   inner join
17     docentopgaven ON docentopgaven.id = studentdocentopgaven.DocentOpgavenID
18 where
19   VraagTypeID = 2 and Poging > 1
20
select
*
from
  studentenbeoordelingmcvragen
  inner join
    studentenbeoordelingmcvragen_1 ON studentenbeoordelingmcvragen.studentid = studentenbeoordelingmcvragen_1.studentid

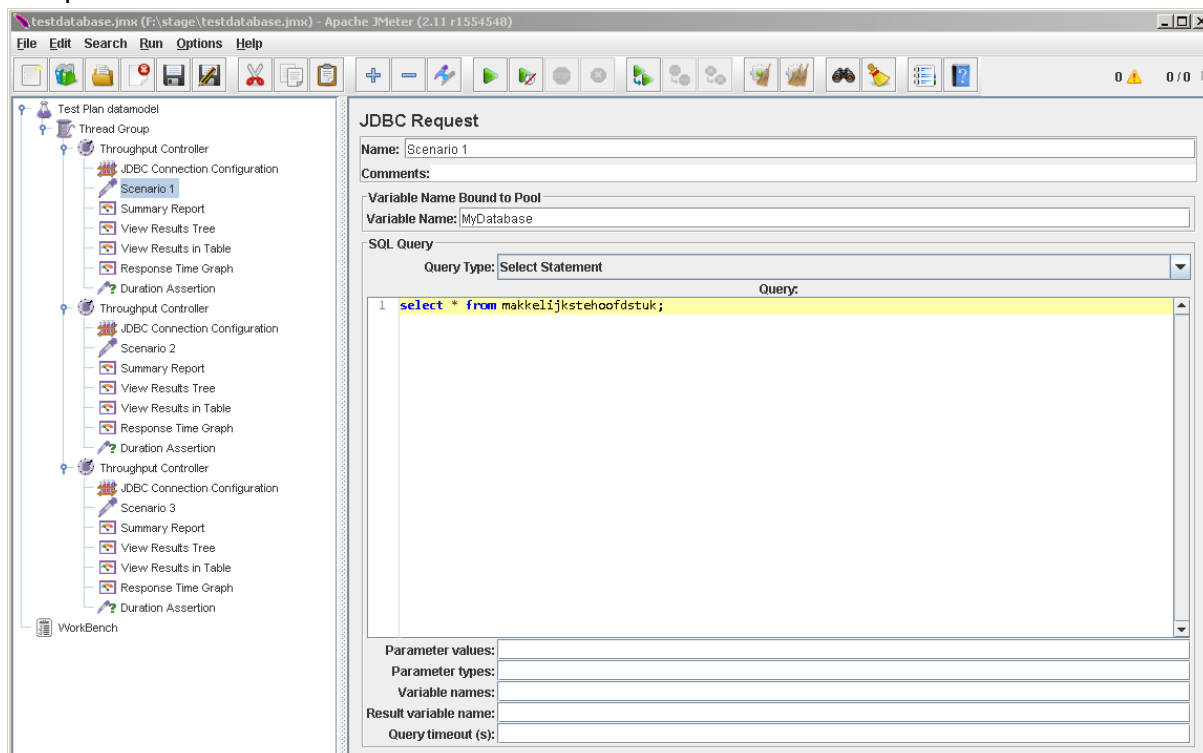
```

Bijlage 9 Testplannen Jmeter

Testplan use cases



Testplan datamodel



Bijlage 10 Technisch ontwerp

Bijlage 11 Blackbox testplan