

De Haalbaarheidsstudie

Definitie:

Met een haalbaarheidsstudie en systeemanalyse probeert men antwoorden te vinden op de volgende vragen: (waarom, hoe en wat het informatiesysteem gaat doen) Het doel van de definitiestudie is dan ook om te kijken of het te maken informatiesysteem er redelijkerwijs kan komen.

De haalbaarheidsstudie is het startpunt voor een project dat daadwerkelijk het systeem gaat maken. Tijdens de start van een project om een nieuw informatiesysteem te creëren is het ook verstandig om te kijken naar de mogelijke risico's ervan. Risicomanagement maakt dan ook deel uit van de haalbaarheidsstudie. Immers, met veel risico's komt de haalbaarheid in het geding. (Bron: Edhub, Informatiesystemen, 12.1, alinea's 2 + 3)

De haalbaarheidsstudie heeft als eerste stap het onderzoeken of er wel een informatiesysteem gemaakt moet worden als oplossing voor het gesignaleerde probleem.

Het toetsen op de haalbaarheid gebeurt vervolgens drievoudig waarbij gekeken wordt of het informatiesysteem:

- Technisch haalbaar is
- Economisch haalbaar is
- Operationeel en organiek haalbaar is

(Bron: Edhub, Informatiesystemen, 12.2, alinea 2).

Onderzoek of het nieuw systeem er moet komen

Het Backoffice-team van Don Diablo heeft een nieuw systeem nodig dat het beoordelen van demo's, en zo het ontdekken van nieuw talent efficiënter maakt. Hun huidige systeem draait als volgt:

Producers die hun demo willen laten beoordelen, doen dit via een webformulier. Hier vullen zij hun gegevens in en voegen daar de link naar hun demo aan toe. Vervolgens komt het formulier als bericht in de mailbox bij het backoffice-team terecht. Het backoffice-team gebruikt dan de link in de mail om de muziek te downloaden. Als de demo goed gevonden wordt komt deze uiteindelijk bij Don Diablo terecht. Zo niet maakt het backofficeteam dit kenbaar.

Als Don Diablo net zo enthousiast is over de demo wordt er contact met de verzender opgenomen. Vindt Don Diablo het toch niks maakt hij dat ook kenbaar.

Het Backoffice-team stelt hiervoor in de plaats een beter systeem te laten ontwikkelen welke zij het "Demodrop-systeem" gaan noemen. De bedoeling is dat kwesties als o.a. overzicht, snelheid en gebruiksvriendelijkheid met dit nieuwe systeem worden verholpen.

In het kort willen zij een systeem waarbij er gemakkelijk met opgestuurde demo's kan worden omgegaan. Gebruikers krijgen een omgeving waar zij hun demos kwijt kunnen, en het backoffice-team kan deze, in hun omgeving makkelijk goed-en-afkeuren.

De volledige use case is te vinden in de use-case diagram en de requirements zijn zowel in Requirements.pdf als de technische haalbaarheidsstudie te vinden.

Conclusie is dat het huidig systeem:

- Voor de gebruiker saai is en de mogelijkheid tot spammen biedt.
- De gebruiker de mogelijkheid geeft om de ontvanger naar een ongewenste of gevaarlijke site door te sturen.
- Aan de kant van een backoffice-medewerker: onoverzichtelijk is door al die emails.
- Een slechte mogelijkheid biedt tot het traceren van duplicaten en spam.
- Erg omslachtig is door het moeten klikken op een link en een download op te zoeken.
- Een backoffice-medewerker onnodige herhalende taken laat verrichten.

..En dus het gevraagde systeem er moet komen.

Technische Haalbaarheid:

Technisch haalbaar betekent dat de technieken die gebruikt gaan worden bestaan of kunnen worden gemaakt.

(Bron: Edhub, Informatiesystemen, 12.2, alinea 17)

Technische haalbaarheid van de algemene requirements

ID	Requirement	Haalbaar?	Haalbaar omdat/Uitwerking	Verwachte tijdsbesteding	Kosten
	Het Novi beoordelingsteam moet in staat zijn de eindopdracht zelf te kunnen draaien	copy ✓X paste	Er wordt gebruik gemaakt van maven en dependencies zodat de juiste bibliotheken met hun versies op de computer van de gebruiker worden geïnstalleerd.	n.v.t.	n.v.t.
	De applicatie moet op de modernere browsers: Chrome en FireFox draaien	✓	Chrome en Firefox zijn stabiele browsers.	n.v.t.	n.v.t.
	De website moet responsive zijn	✓	Dit gebeurt door media queries in de css te plaatsen. Met behulp van het prototype weet ik exact hoe de pagina's en hun verschillende views eruit komen te zien. Het is ook handig is om mobile first te ontwikkelen. https://youtu.be/ZYV6dYtz4HA?t=812 13:32 - 20:55 https://youtu.be/O_J9BGq57Ss 2:03 - 5:22 (1,5x speed) https://youtu.be/Z2s7IEIXIW8 1:27 - 12:10	n.v.t.	n.v.t.
	Taal van de applicatie is in het Engels	✓	Inhoud van de applicatie wordt in Engels geschreven.	n.v.t.	n.v.t.

	De applicatie bevat Unit tests	✓	Test-driven development wordt zover het kan toegepast op functies die verwerkte waardes teruggeven.	n.v.t.	n.v.t.
--	--------------------------------	---	---	--------	--------

Technische haalbaarheid van de **User-side** Requirements

ID	Requirement	Haalbaar?	Haalbaar omdat/Uitwerking	Verwachte tijdsbesteding	Kosten
01	De applicatie moet een front-end en een back-end hebben.	✓	Voor de frontend gebruik ik HTML, CSS, en Thymeleaf (Backend: Wordt uitgewerkt in Requirements 02 tot 13)	n.v.t.	n.v.t.
02	Gebruikers kunnen een account aanmaken	✓	Gegevens in de html form worden door middel van een post-request verwerkt en opgeslagen in de database.	Voorbeeld: 3 uur	n.v.t.
03	Alle benodigde informatie van de gebruiker zijn: Naam, Email, Gebruikersnaam, Wachtwoord, (Telefoonnummer?)	✓	Dit zijn de benodigde kolommen die we in elk geval in de database moeten opslaan. & Op wachtwoorden worden eerst securitymaatregelen toegepast voordat ze in de database terecht komen.		3 uur
04	Bij het aanmaken van een account moet er akkoord worden gegaan met Algemene Voorwaarden	✓	Alleen als de checkbox is aangekruist kan een gebruiker een postrequest maken. Html form, checkbox + Javascript + spring boot		n.v.t.
05	Bij het aanmaken van een account wordt een welkomsmail verstuurd met daarin ook een link om het profiel te activeren				

	Is ook een Security maatregel: Voorkomt spoofing	✓	Dit wordt 'ge-handled' in een mail-service met behulp van de Spring-Boot-Starter-Mail dependency.		
06	Gebruikers krijgen een profiel	✓	Informatie van de bijbehorende gebruiker worden uit de database gehaald		
07	Elk profiel krijgt een standaard interface. De opgeslagen data binnen de database bepaalt de input van de velden	✓	Informatie van de bijbehorende gebruiker worden uit de database gehaald, en met behulp van Thymeleaf gepresenteerd.		
08	De gebruiker kan een demo uploaden.	✓	<p>Voor upload: HTML Post & Spring Stream, https://edhub.novi.nl/study/courses/307/content/7237 https://youtu.be/lbb4FS_mjU0?t=34*read data client-s https://youtu.be/4yb16ITxbM8?t=1310 *Spring booth? DB Architecture, MVC -> Update view dashboard</p> <p>-- Ik ben hier niet helemaal uit, maar ik zal de bestanden moeten opslaan in een FileSystem en de locatie in de de db</p>		
	<p>.wav en .ogg geuploadde bestanden moeten worden omgezet naar .mp3</p> <p>.mp3 bestanden worden gecomprimeerd naar maximaal 15 mb 128 kbps</p>	✗	<p>Dit is tijdsgezien niet haalbaar en er is daarom met de opdrachtgever afgesproken dat we deze functie er niet in doen. De applicatie kan als MVP ook prima overleven zonder deze functionaliteit.</p> <p>Een alternatieve oplossing hiervoor in de plaats is om de gebruiker alleen toe te staan om een mp3 bestand van max 15 mb te uploaden.</p>		

08	De gebruiker kan bij de demo een omschrijving plaatsen.	✓	Dit wordt een vereist veld en wordt als row opgeslagen bij de demo Entity (one to one relationship)		
10	De gebruiker mag maximaal 10 demo's op zijn profiel.	Nog uitzoeken	In de Controller eerst kijken if "Demo's pending" van deze verzender == 10, redirect naar th: maxReachedPopupMessage else mag een bestand worden opgeslagen.		
11	De opgestuurde demo en informatie over de gebruiker komen bij de Backoffice-side terecht.	Dit moet	Omdat de frontend in Thymeleaf is gebouwd, kan ik makkelijk de getrequests bepalen. (get demo's in 'with status pending' (1 voor 1))		
12	De gebruiker is in staat zijn demo terug te trekken zolang een backofficemedewerker zijn Demo nog niet in behandeling heeft genomen. <i>(lees verder)</i> <i>*(Is het bestand ook hier nog steeds eigendom van Hexagon?)</i> Na het in behandeling nemen van de demo blijft het bestand eigendom van het DemoDrop-systeem	Nog uitzoeken	Met behulp van thymeleaf Of zolang th:linReview th:handled, (show het stukje html: delete demo) ** (In review modus aan laten staat. Zodat bij wegglikken de taak niet verloren gaat, blijft opgeslagen)		
13	Gebruikers moeten hun account kunnen sluiten.	✓	De demo's blijven bestaan in de database van het DemoDrop-systeem.		

	De volgende informatie wordt gewist: Alles behalve de demo en beoordeling ervan.		Er komt extra collumn: Account deleted, om zo het overzicht te kunnen houden en eventueel de mogelijkheid bieden die demo's op een later moment te verwijderen. Account deleted has one demo		
	TOTAAL	n.v.t.	n.v.t.	Duur: gelijktijdig met de BO en Admin-side businesslogic: 3 weken	Kosten: €19.99 2 Udemy Cursussen

Technische haalbaarheid van de **Backoffice-side** Requirements

ID	Requirement	Haalbaar?	Haalbaar omdat/Uitwerking (Bespreken met groep)	Verwachte tijdsbesteding	Kosten
01	Het systeem ontvangt de omgezette mp3 demo's.		MultipartFile uploaden in filesystem met MVC Coltroller.		
02	Het systeem presenteert de demo's ergens op chronologische volgorde.		Bij /review, /handled demos en /sent demos een List van demo's (in hun state) genereren.		
03	Een medewerker kan aangeven dat hij een demo "In behandeling neemt".		Een demo wordt krijgt automatisch zn state als hij aan de bo-medewerker wordt gepresenteerd		
04	Op het moment dat een demo "in behandeling" wordt genomen vervalt de mogelijkheid tot verwijdering bij de gebruiker.		De status van het demo-object bepaalt de html generatie dankzij thymeleaf		
06	Bij een demo wordt informatie over de verzender getoond.		Get requests uit de user entity Demo has a user (FK)		
07	Het systeem kan de demo afspelen.		Opvragen/lezen uit Filesystem + Html audiotag		
08	De medewerker kan commentaar geven		Deze functie is eruit gelaten omdat er dan meer functionaliteit bij de backoffice komt kijken onder andere: dat de backoffice ergens meldingen moet		

			<p>ontvangen, bij elke demo er een chatfunctie moet zijn, enzovoorts.</p> <p>Een alternatieve requirement om dit probleem op te lossen is dat het een eenrichtingsverkeer wordt. Een producer ontvangt alleen de informatie of zijn demo is doorgestuurd of niet.</p> <p>En als een backofficemedewerker écht een dringende vraag heeft, kan hij het emailadres van de verzender terugvinden.</p> <p>Dit is uiteindelijk ook met de opdrachtgever afgesproken.</p>		
08	Er wordt een email naar de producer verstuurd bij het ontvangen van commentaar.		Spring email + thymeleaf met doorverwijzing naar de dashboard. (waarna iemand vervolgens moet inloggen waarschijnlijk)		
10	De medewerker kan een demo goed of afkeuren.		Hier veranderd de state naar handled en (Bij /handleddemos kan ik een list van (state)handled demos én forwarded demos opvragen		
11	Goedgekeurde én afgekeurde demos worden ergens zichtbaar. Gaat ook Repudiation tegen		Deze worden gemapped bij /handled en /sentdemos (ik moet nog vaste namen bedenken)		
12	Bij goedkeuring zorgt het backoffice-team zelf dat de demo bij Don Diablo terecht komt.		Zij kunnen dan bij requirement 06 terecht. Er is een download-knop beschikbaar bij het afspelen van de demo.		

			Nice to have (nu we toch al bezig zijn met de mailservice) Is dat er een popupscherm verschijnt bij het klikken op “✓” *en pas wanneer er op next wordt geklikt, krijgt de demo zijn nieuwe state “sent”		
	TOTAAL	n.v.t.	n.v.t.	Duur: gelijktijdig met de User en Admin-side businesslogic: 3 weken	Kosten: €19.99 (zelfde) Udemy Crusussen

Technische haalbaarheid van de **Admin-side** Requirements

ID	Requirement	Haalbaar?	Haalbaar omdat/Uitwerking (Bespreken met groep)	Verwachte tijdsbesteding	Kosten
01	Admin kan accounts voor het backofficeteam maken/verwijderen		<p>Dit gebeurt door middel van een het invullen van een form, waarbij alleen de naam, achternaam en het mailadres van de backofficemedewerker nodig is.</p> <p>Het systeem stuurt een e-mail met gegenereerde wachtwoord voor de BO-gebruiker. De gebruiker kan vervolgens zijn wachtwoord wijzigen. *Voorkomt repudiation</p>		
02	Een admin kan de standaardteksten voor het backofficeteam instellen.		De standaardteksten van de admin worden opgeslagen een table voor standaardteksten, waaruit vervolgens de backofficemedewerker “kan bepalen welke tekst bij de demo verschijnt” (one demo has one reply)		
03	Kan handmatig demo's verwijderen.		*Nog uitzoeken hoe hierde crud-functionatliteit in de frontend kan worden toegepast		
	TOTAAL	n.v.t.	n.v.t.	Duur: gelijktijdig met de User en BO-side businesslogic: 3 weken	Kosten: €0

Technische haalbaarheid van de **security toepassingen**

ID	Requirement	Haalbaar?	Haalbaar omdat/Uitwerking (Bespreken met groep)	Verwachte tijdsbesteding	Kosten
	Spoofing: Aanval op authenticatie. Om te voorkomen dat gebruikers kunnen voordoen als iemand anders:	✓	...wordt er een welkomsmail met een bevestigingslink gestuurd om het account te activeren. Dit gebeurt met de Emailservice. (De emailservice slaat een uuid op aan een user, vervolgens wordt gecheckt op het bestaan van een user + uuid) Er geen accounts op dat mailadres worden gemaakt.		
	Tampering: Aanval op Integriteit Om te kunnen voorkomen dat gebruikers (opgeslagen) data kunnen wijzigen:	✓	... laat ik Spring Security alle input-validaties regelen.		
	Repudiation: Aanval op *geschiedenis Om te kunnen voorkomen dat gebruikers kunnen ontkennen dat ze iets (wel of niet) hebben gedaan ... moeten we een manier vinden waarop er niet ontkent kan worden dat iets (wel of niet) is gebeurt.	✓/X	... Zo kun je aan de backofficekant zien door wie een demo is beoordeeld. In de code wordt ook een "AuditorAware" interface geïmplementeerd, die bijhoudt welke wijzigingen door welke gebruiker zijn gemaakt. Verder heeft de adminpagina alle overzicht op alle demo's die binnenkomen.		
	Information Disclosure: Aanval op Vertrouwelijkheid (<i>lees verder</i>)	✓	Alle mappings en de toegang tot de DB zijn "dichtgetimmerd" door Spring boot's security.		

	Om te voorkomen dat gebruikers/ontwikkelaars informatie zien die ze niet mogen zien worden wachtwoorden ge-encrypt en de juiste restricties per gebruikerstype ingesteld.		En wachtwoorden worden ge-encrypt door Spring security zodat de wachtwoorden niet openlijk zichtbaar zijn in de DB.		
	Denial of Service: Aanval op de beschikbaarheid van een server Om te voorkomen dat gebruikers een systeem onbeschikbaar maken:	✓	...Hiertegen hoeft geen maatregelen genomen te worden omdat de applicatie op een locale host wordt gedraaid, en de doelgroep het Novi beoordelingsteam is.		
	Elevation of Privilege: Aanval op Autorisatie Om te voorkomen dat gebruikers handelingen verrichten zonder toestemming ...	✓	In de securityconfiguration kan ik rollen bij mappings toewijzen, zodat iedere type gebruiker de juiste toestemmingen krijgt.		
	TOTAAL	n.v.t.	n.v.t.	Duur: Einde van productie, circa 3 dagen	Kosten: €15Udemy Cursus

Technische haalbaarheid van de **UI** Requirements

ID	Requirement	Haalbaar?	Haalbaar omdat/Uitwerking	Verwachtte tijdsbesteding	Kosten
	De applicatie moet responsive zijn. Bij Backoffice heeft dat een lager prioriteit.	✓	Frontend: Css media queries	n.v.t.	n.v.t.
	User-side mooi zijn. Bij backoffice heeft dit een lager prioriteit.	✓	De huisstijl van Don diablo wordt aangehouden, en er wordt geprobeerd de site zo goed mogelijk na te maken. Zo kun je er qua UI niet naast zitten.	n.v.t.	n.v.t.
	TOTAAL	n.v.t.	n.v.t.	Duur: 4 dagen	Kosten: €0

Economische Haalbaarheid:

Economisch haalbaar betekent dat de opbrengsten van een informatiesysteem meer zijn dan de kosten en de inspanning. De opbrengsten van zo'n informatiesysteem zijn echter niet alleen financieel. De baten kunnen ook op menselijk of ecologisch vlak liggen. Het is vaak lastiger om deze baten goed zichtbaar te maken in de haalbaarheidsstudie. (Bron: Edhub, Informatiesystemen, 12.2, alinea 5).

Kosten Opdrachtgever:

- Nog niets.
- Tijd voor meetings en tussentijdse beoordelingen.

Opbrengsten opdrachtgever:

- Don Diablo kan nieuwe inspiratie opdoen.
 - Don Diablo kan nieuw talent ontdekken.
 - Het backoffice-team kan het onhandige mailsysteem en de frustraties daarvan schrappen.
 - Het backoffice-team kan vlugger te werk gaan.
 - Het backoffice-team kan minder personeel nodig hebben.
 - De gebruiker krijgt een unieke kans in aanraking met Don Diablo te komen.
 - Zowel de gebruiker en Don Diablo kunnen hun netwerk uitbreiden.
- De opbrengsten voor de opdrachtgever in euro's zijn nog niet bekend.

Kosten hosting:

€0, De applicatie zal draaien op een local host.

Kosten ontwikkelaar: Esther:

- Een opleiding om de opdracht te kunnen uitvoeren: €3300.
- Tijd uit privé-leven.
- Mogelijk slaaptekort.

Opbrengsten ontwikkelaar:

- De ontwikkelaar krijgt 60 ECTS en een functiediploma.
- De ontwikkelaar heeft ervaring opgedaan.
- De ontwikkelaar kan zeggen Don Diablo, of Hexagonhq van Don Diablo als opdrachtgever te hebben gehad.
- De ontwikkelaar heeft een mooi, compleet eerste portfolio-project met de volledige stack erop en eraan: Een multi-tiersysteem met: een front-end toepassing, serverside-systeem, database en beveiligingstoepassingen.

Omdat hier de opbrengsten meer zijn dan de kosten, is ontwikkelen van het te komen systeem economisch haalbaar.

Nieuwe kosten en opbrengsten worden duidelijk nadat de technische haalbaarheidsstudie compleet is en in eventueel nieuwe gesprekken met Arjen Wiersma. Nieuwe kosten en opbrengsten kunnen ook opduiken tijdens productie.

Operationele en organiek haalbaarheid:

Operationeel en organiek haalbaar betekent twee dingen: past het nieuwe informatiesysteem in de manier van werken, of moeten we de processen van de organisatie aanpassen? Zo ja, lukt het dan om de organisatie te veranderen en alle medewerkers mee te krijgen in die verandering?

(Bron: Edhub, Informatiesyste-men, 12.2, alinea 8 + 10)

Het backoffice-team is met het verzoek gekomen hun nieuwe systeem tot realisatie te brengen. Zij hebben dit systeem bedacht omdat deze, in tegenstelling tot het huidige systeem, beter aansluit bij hun manier van werken.

Dus past het nieuwe systeem bij de manier van werken? – Ja. En lukt het om alle medewerkers mee te krijgen in die in de verandering? – Daar is het backoffice-team voor verantwoordelijk.

Optioneel: Risico management: Mogelijke risico's

Tijdens de start van een project om een nieuw informatiesysteem te creëren is het verstandig om ook te kijken naar de mogelijke risico's ervan. Risicomanagement maakt dan ook deel uit van de haalbaarheidsstudie. Immers, met veel risico's komt de haalbaarheid in het geding

(Bron: Edhub, Informatiesystemen, 12.2, alinea 2)

Bij de ontwikkeling van geautomatiseerde informatiesystemen zijn er zeven belangrijke soorten risico's te identificeren:

- Commerciële en contractrisico's. Dit zijn risico's die voortkomen uit het falen van toeleveranciers, zoals niet op tijd leveren, onvoldoende kwaliteit leveren, of te hoge kosten rekenen.
Een voorbeeld zou zijn dat gebruikte libraries en dependencies achteraf voor onverwachte uitkomsten zorgen. Een hoge risico is niet het geval. De diensten die ik bij het ontwikkelproces gebruik zijn redelijk stabiel. Onder andere: Java, Spring boot, Google fonts en de browsers Chrome en Firefox.
- Economische omstandigheden. Dit zijn de risico's die gerelateerd zijn aan een veranderende markt of het niet meer beschikbaar zijn van budget.
Omdat dit een schoolopdracht is hebben wij hier geen last van. Maar de applicatie die ontwikkeld wordt is geheel kosteloos voor de opdrachtgever en voor de applicatie is er weinig sprake van een veranderende markt omdat het systeem het beoordelingsproces van de medewerkers moet verbeteren.
- Menselijk handelen. Dit zijn de risico's die te maken hebben met tekortkomingen in personeel of onvoldoende personeel, zodat het werk niet kan worden afgemaakt.
Er bestaat een risico dat het project kan worden vertraagd of stopgezet door het uitvallen van een collega, of het onderschatten van de hoeveelheid werk. (Het laatste is nu meer van toepassing)
- Politieke omstandigheden. Politici kunnen zich op de meest onverwachte manieren bemoeien met het creëren van informatiesystemen. Dat kan bijvoorbeeld door middel van

boetes of het verbieden van het gebruik van bepaalde technologie.

Het risico hiervan is echter laag omdat het huidige demodrop-systeem nog steeds draait. Ook hoeft ik me over politieke omstandigheden geen zorgen te maken als developer.

- Technologische onderwerpen. Technologie en technologische risico's hebben bijvoorbeeld te maken met het niet kunnen oplossen van problemen, of met een verkeerde introductie van nieuwe technologie.

De technologieën voor het behalen van de requirements bestaan al jaren. Toch blijft er een risico omdat een eigen implementatie voor onverwachte uitkomsten kan zorgen.

- Managementactiviteiten en controls. Management heeft een belangrijke stem in de manier waarop projecten lopen. Zo kunnen persoonlijke (dubbele) agenda's van managers een rol spelen.

Dit is voor mij moeilijk in te schatten.

- Acties van individuen. Bij deze categorie risico's speelt het feit dat over-specificatie kan plaatsvinden: men automatiseert uitzonderingen of onrealistische verwachtingen van het nieuwe informatiesysteem. Men denkt bijvoorbeeld dat elk mogelijk risico kan worden opgelost.

Voor mij is het héél belangrijk dat ik me houd aan de noodzakelijke eisen voor een mvp product.

(Bron: Edhub, Informatiesystemen, 12.2, alinea 17)

Conclusie

Uit de haalbaarheidsstudie is gebleken dat de volgende requirements niet haalbaar zijn:

- “.wav en .ogg geuploadde bestanden moeten worden omgezet naar .mp3”, omdat:
 - Dit tijdsgezien niet haalbaar is.
 - Het systeem kan zonder deze functionaliteit alsnog als mvp werken
 - Doelgroep gebruikers kunnen dit zelf doen
- “.mp3 bestanden worden gecomprimeerd naar maximaal 15 mb 128 kbps”, omdat:
 - Dit tijdsgezien niet haalbaar is.
 - Het systeem kan zonder deze functionaliteit alsnog als mvp werken
 - Doelgroep gebruikers kunnen dit zelf doen
- “De backoffice medewerker kan commentaar geven”, omdat:
 - Er anders een boel aan andere requirements hieronder komen vallen. Een backofficemedewerker moet bijvoorbeeld ergens meldingen krijgen & reacties kunnen ontvangen. Aan de userkant komen ook weer nieuwe reqs bij kijken.
 - Ik zie niet hoe ik dit logischerwijs zou vormgeven.
 - Als een backoffice medewerker dringend iemand wil benaderen, anders dan of zijn demo is doorgestuurd, kan hij de email gebruiken die bij het ontvangen van een demo wordt meegestuurd.

De volgende punten een risico vormen:

- De toepassing van security-requirements omdat:

- We hierover in de lessen geen demonstratie hebben gehad. Dit vormt mogelijk een gevaar voor de volledige veiligheid van het systeem.
- Het toepassen van Unit tests op webapplicaties
 - We hebben hierover helaas geen uitleg of demonstratie gehad. Een goede kennis van testing voorkomt dat een ontwikkelaar achteraf bugs moet fixen, nadat ze allang in het ontwikkelproces zijn meegenomen. Dit vormt een risico voor onnodig tijd besteden aan het fixen van hardnekkige bugs.
- De settingspagina: Het kunnen wijzigen van een wachtwoord, of een foto kunnen wijzigen omdat:
 - Ik tot nu toe alleen kleine beetje online heb gevonden maar nog geen concrete oplossing. Dit is een punt waar ik mogelijk vast zou kunnen zitten in het ontwikkelproces.
- De Backoffice “review”-pagina:
 - Deze is vrij complex. Ik heb nog nooit zo’n systeem gemaakt en dit wordt de eerste keer. Ik ga dus zonder enige kennis dit proces in.
- De Admin-pagina:
 - De uitdaging zit bij mij in hoe ik de gebruiker crudopties kan geven en hoe ik de tabellen ga weergeven.



Verder is uit de haalbaarheidsstudie gebleken dat de productie waarschijnlijk tot de eerste herkansing duurt, en het project mij €0 kost.

Dit is een positief resultaat voor het uitvoeren van de ontwikkeling van het gevraagde systeem.

Einde haalbaarheidsstudie.