

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)

De requirements zijn dit keer smart geformuleerd. Onder het kolom "Requirement" valt specifiek, onder "Haalbaar omdat" vallen de punten Meetbaar, Acceptabel en Realistisch en onder "Verwachtte tijdsbesteding" valt tijdsbesteding van het SMART model. Verder worden ook de verwachtte kosten in kaart gebracht.

Technische haalbaarheid van de algemene requirements

ID	Requirement (Specifiek)	Haalbaar?	Haalbaar omdat/Uitwerking (Meetbaar, Acceptabel, Realistisch)	Verwachtte (Tijdsbesteding)	Kosten
	Het Novi beoordelingsteam moet in staat zijn de eindopdracht zelf te kunnen draaien	✓	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	(Zie fontend development: Binnen 7 dagen (54 uur))	n.v.t.

			https://youtu.be/O_J9BGq57Ss 2:03 - 5:22 (1,5x speed) https://youtu.be/Z2s7IEIXIW8 1:27 - 12:10		
	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 van een aantal belangrijke handelingen binnen de applicatie :Userside	✓	Omdat dit de eerste keer programmeren is, is het handig om in dit geval het schrijven van Unit tests achteraf te doen.	Tijdens gehele backend productie (binnen 21 dagen)	n.v.t.

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

ID	Requirement (Specifiek)	Haalbaar?	Haalbaar omdat/Uitwerking (Meetbaar, Acceptabel, Realistisch)	Verwachtte (Tijdsbesteding)	Kosten
01	De applicatie moet een front-end en een back-end hebben.	✓	Voor de frontend plan ik te gebruiken: HTML, CSS, Javascript, Bootstrap en Thymeleaf (Backend: Wordt uitgewerkt in Requirements 02 tot 13)	n.v.t.	€19.99 2 Udemy Cursussen voor kennis voor bouw van de applicatie
02	Gebruikers kunnen een account aanmaken	✓		(Backend)	n.v.t.

			<p>Realistisch omdat deze methode van authenticatie/authorisatie bijna overal online is geïmplementeerd.</p> <p>Gegevens in de html form worden verwerkt en opgeslagen in de database.</p>	<p>Authenticatie en Authorisatie:</p> <p>1 werkdag (8 uur)</p>	
03	<p>Alle benodigde informatie van de gebruiker zijn:</p> <p>Naam, Achternaam, Artiestennaam, Bio, Email, Wachtwoord, Profielfoto</p>	✓	<p>Dit zijn de benodigde informatie die we in de database opslaan, en die van belang zijn voor de gebruikers van het gehele systeem.</p> <p>Op wachtwoorden worden eerst securitymaatregelen toegepast voordat ze in de database terecht komen.</p>	n.v.t.	n.v.t.
04	<p>Bij het aanmaken van een account moet er akkoord worden gegaan met Algemene Voorwaarden</p>	✓	<p>Alleen als de checkbox is aangekruist kan een gebruiker een postrequest maken.</p>	n.v.t.	n.v.t.
05	<p>Bij het aanmaken van een account wordt een welkomsmail verstuurd met daarin ook een link om het profiel te activeren</p> <p>Is ook een Security maatregel: Voorkomt spoofing</p>	✓	<p>Dit wordt 'ge-handled' in een mail-service met behulp van de Spring-Boot-Starter-Mail dependency.</p>	<p>(Backend) Emailservice opzetten</p> <p>(3 uur)</p>	n.v.t.
06	<p>Gebruikers krijgen een profiel (Vergelijkbaar met facebook/gmail .etc)</p>	✓	<p>Om de applicatie te gebruiken moet je ingelogd zijn.</p> <p>Informatie van de bijbehorende gebruiker worden uit de database gehaald</p>	<p>(Backend) Database structuur en Authenticatie+</p>	n.v.t.

				Authorisatie opzetten 2 dagen (16 uur)	
07	Elk profiel krijgt een standaard interface. (Vergelijkbaar als bij facebook/gmail .etc) De opgeslagen data wordt uit het database gehaald.	✓	Informatie van de bijbehorende gebruiker worden uit de database gehaald, en met behulp van Thymeleaf gepresenteerd.	^ *(hoort bij het bovengenoemde)	n.v.t.
08	De gebruiker kan een demo uploaden.	✓	Voor upload: HTML Post & Spring Stream, https://edhub.novi.nl/study/courses/307/content/7237 https://youtu.be/lbb4FS_mjU0?t=34 https://youtu.be/4yb16ITxbM8?t=1310 Het bestand wordt als multipartfile geupload, verwerkt en opgeslagen in een filesystem. In de database wordt dan de locatie van het bestand als tekst opgeslagen.	(Backend) 2 werkdagen (16 uur)	n.v.t.
	.wav en .ogg geuploadde bestanden moeten worden omgezet naar .mp3 .mp3 bestanden worden gecomprimeerd naar maximaal 15 mb 128 kbps	✗	Dit is tijdsgezien niet haalbaar en is daarom met de opdrachtgever besloten dat we deze functie er niet in doen. Verder kan de applicatie zonder deze functionaliteit prima als MVP overleven.	n.v.t.	n.v.t.

			- 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.		
08	De gebruiker kan bij de demo een omschrijving plaatsen.	✓	Dit veld wordt verplicht: Een demo kan 1 omschrijving bevatten voor in de database	Verwerking in backend: 3 uur Verwerking frontend + css: 2 uur	n.v.t.
10	De gebruiker mag maximaal 10 demo's tegelijkertijd in behandeling hebben.	✓	Het systeem kijkt hoeveel demo's de staat "In behandeling" hebben in de database. Zijn het er 10, krijgt de gebruiker geen toegang tot de upload-pagina en verschijnt er een popupvenster waarin staat dat de gebruiker moet wachten tot een BO-medewerker een demo in behandeling heeft genomen.	Verwerking in backend: 1 uur Verwerking frontend + css: 1 uur	n.v.t.
11	De opgestuurde demo en informatie over de gebruiker komen bij de Backoffice-side terecht.	Dit moet ✓	Een goede database-structuur staat voorop. Door de frontend in Thymeleaf te bouwen, kan ik gemakkelijk de getrequests schrijven. Hiermee kan ik eenvoudig de informatie laten opvragen die voor de BO-medewerker van belang zijn: Artiestennaam gebruiker, naam, email-adres, bio. Demo zelf, titel demo, omschrijving demo.	(Backend) 3 dagen (36 uur)	n.v.t.
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>	✓	Het systeem controleert in welke staat de demo is: Al beoordeeld of niet.	(Backend) 1 uur	n.v.t.

	Na het in behandeling nemen van de demo blijft het bestand eigendom van het DemoDrop-systeem		Aan de hand daarvan zal voor de gebruiker wel of geen delete-knop aanwezig zijn. Oo dit kan worden gedaan met Thymeleaf		
13	Gebruikers moeten hun account kunnen sluiten. De volgende informatie wordt gewist: Alle gebruikersinfo + de locatie naar de geüploade demo's	✓	Account van de gebruiker wordt gesloten door middel van een delete-request. De demo's blijven bestaan in de filesystem van de upload-locatie (omdat alle demo's eigendom moeten blijven van Hexagon) Echter wordt de gehele record van de gebruiker en met zijn relaties verwijderd.	(Backend) 3 uur	n.v.t.
	TOTAAL	n.v.t.	n.v.t.	Duur: 73 uur	€19.99

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

ID	Requirement (Specifiek)	Haalbaar?	Haalbaar omdat/Uitwerking (Meetbaar, Acceptabel, Realistisch)	Verwachte (Tijdsbesteding)	Kosten
01	Het systeem ontvangt de omgezette mp3 demo's.		<p>Het bestand wordt als multipartfile geüpload en verwerkt. De namen van de bestanden blijven hetzelfde en worden opgeslagen op één bestemming.</p> <p>In de database wordt de verwijzing naar de locatie opgeslagen. Om dit succesvol te doen wordt een pre-fix aan het originele bestandsnaam gekoppeld.</p> <p>Dit wordt allemaal uitgewerkt in de backend.</p>	5 uur	n.v.t.
02	Het systeem presenteert de demo's op chronologische volgorde.		<p>Op de pagina's: /review, /handled demos en /sent demos, word een lijst met demo's gepresenteerd.</p> <p>Om deze demo's op volgorde te kunnen presenteren moeten deze demo's in een List opgeslagen worden, en gepresenteerd in de volgorde van tijd.</p>	8 uur	n.v.t.
03	Een medewerker kan aangeven dat hij een demo "In behandeling neemt".		Een demo krijgt bij upload direct de status "Pending" (in wachtrij).	8 uur	n.v.t.

			Als een Bo-medewerker aan de Bo-side een oordeel submit, verandert de state (status) van de demo naar "rejected" of "sent". Vervolgens wordt ook de beoordelaar aan de demo gekoppeld.		
04	Op het moment dat een demo "in behandeling" wordt genomen vervalt de mogelijkheid tot verwijdering bij de gebruiker.		<p>Dit kan eenvoudig worden gedaan bij de frontend. De deleteteknop wordt niet zichtbaar bij demo's met de status: "rejected" of "sent".</p> <p>Maar deze mogelijkheid moet ook in bij de API worden gedicht.</p>	4 uur	n.v.t.
06	Bij een demo wordt informatie over de verzender getoond.		<p>Informatie over de gebruiker moet benaderd worden via de (het) demo (object) zélf. Dit kan dan aan de hand van de database architectuur die erachter zit.</p> <p>Een demo (object) zit dan gekoppeld aan een gebruiker (object) die het heeft geupload.</p>	6 uur	n.v.t.
07	Het systeem kan de demo afspelen.		<p>Google Chrome, en Mozilla Firefox ondersteunen het afspelen van audiobestanden. In de front-end zal dit uiteindelijk resulteren in een audio-tag met daarin de locatie van het audiobestand.</p> <p>De om de audiotag een beetje passend te maken, moet de vormgeving a.d.h.v. Css en Javascript aangepast worden</p>	8 uur	n.v.t.

08	De medewerker kan commentaar geven		<p>Deze functie is eruit gelaten omdat er dan meer functionaliteit bij de backoffice komt kijken onder andere: dat de backoffice ergens meldingen moet 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>	n.v.t.	n.v.t.
08	Er wordt een email naar de producer verstuurd bij het ontvangen van commentaar.		Spring email + Thymeleaf	5 uur	n.v.t.
10	De medewerker kan een demo goed of afkeuren.		<p>In de frontend kan medewerker gebruik maken van een radiobutton keuze.</p> <p>In de backend veranderd hier de staat van de demo van "pending" naar de gekozen beoordeling van de beoordelaar: "rejected" of "sent".</p> <p>Aan de hand van de status worden dan andere handelingen uitgevoerd zoals het verzenden van een</p>	8 uur	n.v.t.

			passende email en andere requirements die hier worden genoemd.		
11	Goedgekeurde én afgekeurde demos worden ergens zichtbaar. Gaat ook Repudiation tegen		Dus voor de beoordelaars komt er een pagina met een lijst van demo's met de statussen: "rejected" en "sent" (uiteraard zijn de demo's met status "pending" zichtbaar bij de pagina van de inkomende demo's)	5 uur	n.v.t.
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 emaildienst) is dat dit proces ook geautomatiseerd wordt. Zo zou een medewerker direct bij het kiezen van een beoordeling een email naar Don Diablo sturen met daarin de demo als bijlage en een standaard tekst.	In geval van de nice to have: 7 uur.	n.v.t.
	TOTAAL	n.v.t.	n.v.t.	Duur: 64 uur	n.v.t.

Technische haalbaarheid van de Admin-side Requirements

ID	Requirement (Specifiek)	Haalbaar?	Haalbaar omdat/Uitwerking (Meetbaar, Acceptabel, Realistisch)	Verwachte (Tijdsbesteding)	Kosten
01	De admin kan accounts voor het backofficeteam maken/verwijderen		<p>Dit kan door middel van een html form, waarbij alleen de naam, achternaam en het mailadres van de backofficemedewerker nodig is.</p> <p>In de backend stuurt het systeem vervolgens een e-mail met een gegenereerd wachtwoord naar de BO-gebruiker.</p> <p>De gebruiker kan vervolgens zijn wachtwoord wijzigen. *Voorkomt repudiation</p>	6 uur	n.v.t.
02	Een admin kan de standaardteksten voor het backofficeteam instellen.		De standaardteksten van de admin wordt bij de table van de statussen opgeslagen. Dit kan in de vorm van een forgeign key zijn of als kolom.	3 uur	n.v.t.

			<p>Zo kan ook de standaard-tekst benaderd worden via de statussen.</p> <p>De tekst wordt door middel van postrequest in een form verzonden.</p>		
03	Kan handmatig gebruikers verwijderen.		Door middel van een request. Een account wordt op dezelfde manier verwijderd als wanneer een gebruiker dat zelf zou doen. (betreft de verwijderde informatie)	2 uur	n.v.t.
	TOTAAL	n.v.t.	n.v.t.	Duur: 11 uur	n.v.t.

Technische haalbaarheid van de security toepassingen

ID	Requirement (Specifiek)	Haalbaar?	Haalbaar omdat/Uitwerking (Meetbaar, Acceptabel, Realistisch)	Verwachtte (Tijdsbesteding)	Kosten
	<p>Spoofing: Aanval op authenticatie.</p> <p>Om te voorkomen dat gebruikers kunnen voordoen als iemand anders:</p>	✓	<p>...wordt er een welkomsmail met een bevestigingslink gestuurd om het account te activeren.</p> <p>Dit gebeurt met de Emailservice. (De emailservice slaat een uuid op aan een user, vervolgens wordt gecheckt op het bestaan van een user + uuid)</p> <p>Er geen accounts op dat mailadres worden gemaakt.</p>	3 uur	n.v.t.

	Tampering: Aanval op Integriteit Om te kunnen voorkomen dat gebruikers (opgeslagen) data kunnen wijzigen:	✓	... laat ik door Spring Security alle bedreigingen tegen sql injection regelen.	1 uur	n.v.t.
	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 gebeurd.	✓/X	... Zo kun je aan de backofficekant zien door wie een demo is beoordeeld. In de code kan 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.	3 uur	n.v.t.
	Information Disclosure: Aanval op Vertrouwelijkheid (<i>lees verder</i>) Om te voorkomen dat gebruikers/ontwikkelaars informatie zien die ze niet mogen zien worden wachtwoorden ge-encrypt en de juiste restricties per gebruikerstype ingesteld.	✓	Alle mappings en de toegang tot de DB zijn "dichtgetimmerd" door Spring security csrf. En wachtwoorden worden ge-encrypt door Spring security zodat de wachtwoorden niet openlijk zichtbaar zijn in de DB.	4 uur	n.v.t.
	Denial of Service: Aanval op de beschikbaarheid van een server	✓	...Hiertegen hoeft geen maatregelen genomen te worden omdat de applicatie op een locale host wordt gedraaid, en de doelgroep het Novi beoordelingsteam is.	0 uur	n.v.t.

	Om te voorkomen dat gebruikers een systeem onbeschikbaar maken:				
	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.	6 uur	n.v.t.
	TOTAAL	n.v.t.	n.v.t.	Duur: 14 uur	n.v.t.

Technische haalbaarheid van de **UI** Requirements

ID	Requirement (Specifiek)	Haalbaar?	Haalbaar omdat/Uitwerking (Meetbaar, Acceptabel, Realistisch)	Verwachtte (Tijdsbesteding)	Kosten
	De applicatie moet responsive zijn. Bij Backoffice heeft dat een lager prioriteit.	✓	Frontend: Wordt uitgevoerd met behuld van Css en css media queries	(Alle frontend) 7 dagen (56 uur)	n.v.t.
	User-side moet mooi zijn.	✓			n.v.t.

	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.	(Tijdens ux-ui design) 3 dagen (36 uur)	
	TOTAAL	n.v.t.	n.v.t.	Duur: 92 uur - 10 dagen	n.v.t.

Totale duur: **508 uur (21 dagen)**.

Totale kosten: **€19.99**

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. En daarnaast maakt het de developers ook risico-vol om veranderingen in code te willen maken of code toe te voegen omdat elke nieuwe verandering iets in de code kan breken.
- De settingspagina: Het kunnen wijzigen van een wachtwoord, of een foto kunnen wijzigen omdat:
 - Ik tot nu toe alleen kleine beetjes 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 **21 dagen** duurt (508 uur) en **kosten** voor het maken van het project **€19,99** kost.

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

Einde haalbaarheidsstudie.