

Projectplan

November 2019

# 

# 

# Inhoudsopgave

[**Inhoudsopgave**](#_t4ebjjtroxpo) **2**

[**Inleiding**](#_kcgtui27zbj2) **4**

[**Aanleiding**](#_dv6mh4ikn92m) **5**

[**Doelen**](#_dhtpx9b1jqxz) **6**

[Fase 1 Projectplan](#_8r2myd3f4b2j) 6

[Fase 2 Functioneel & technische ontwerp](#_3m8z2c790i2) 6

[Fase 3: Sprint 1](#_u580e0szuq7n) 6

[Fase 4: Sprint 2](#_hziupwov9pbo) 6

[Fase 5: Sprint 3](#_kh1ymahpjuy2) 7

[Fase 6: Sprint 4](#_tscbnq0oik6) 7

[Fase 7: Eindgesprek](#_7c0bmt8hg8q4) 7

[**Resultaat**](#_654wk2lof9sz) **8**

[Fase 1: het project plan](#_fb5e3wtgnb08) 8

[Fase 2: Functioneel en technisch ontwerp](#_7064pjv3qh0d) 8

[Fase 3: sprint 1](#_9xzajwa2tuhh) 8

[Fase 4: sprint 2](#_xok94nbre347) 9

[Fase 5: Sprint 3](#_tw0abcei6v3g) 9

[Fase 6: sprint 4](#_lnppzsiq7bvd) 9

[Fase 7: eindgesprek](#_4da9hub4kdfw) 9

[In de laatste fase lever ik het product op en heb ik een eindgesprek met de opdrachtgever. Ook word ik beoordeeld op de kwaliteit van mijn opgeleverde product.](#_iyl7knhvxv3s) 9

[**Afbakening**](#_cppxj8wwonpc) **10**

[Must haves:](#_908wadi5d9xu) 10

[Should haves:](#_f2rz6mgtxst5) 12

[Could haves:](#_1xx5k6ekaphl) 12

[Won’t haves:](#_4obrrjqqacik) 12

[**Planning**](#_jowl09jbbem) **12**

[**Effecten**](#_r9qufdxas0nv) **13**

[**Risico’s**](#_uo1wwibkk89r) **13**

[**Randvoorwaarden**](#_d767vi612ofa) **14**

# 

# Inleiding

Dit projectplan is samengesteld om goed in kaart te brengen wat mijn opdracht is voor het maken van een update tracker ten behoeve van bedrijven (en particulieren). In dit projectplan leg ik voor mijn docenten Frans de Boer en Richard kingma (hierna te noemen: opdrachtgevers) vast waar dit projectplan betrekking op heeft.

Zo zullen hierin onder andere de eisen en wensen van de opdrachtgevers worden vastgelegd zodat er een concreet beeld kan ontstaan van het gewenste eindproduct, aan de hand van de in de inleiding genoemde punten wordt dit projectplan dusdanig geschreven dat de opdrachtgevers geen vragen meer hebben voor de start van het project, en dat alle eisen, wensen, voorwaarden e.d. zijn vastgelegd en geverifiëerd.

Echter zal ik hier ook mijn eigen afbakeningen maken, denk aan het analyseren van het eisenpakket (wat MOET er beslist in, wat kan er eventueel in als er tijd is, wat hoeft er niet in/moet er absoluut niet in?) Verder draag ik ook mijn eisen aan die essentieel zijn voor mij om het project op een schappelijke manier te kunnen realiseren (denk hier bijvoorbeeld aan een bepaalde hosting provider).

# 

# Aanleiding

De aanleiding voor het maken van dit project is het ondersteunen met software up to date te houden. Software up to date houden is namelijk een cruciale factor in het veilig houden van systemen.

Huidige situatie: er is bijvoorbeeld software wat een bedrijf niet vaak gebruikt maar wel cruciaal is voor de veiligheid is van de systemen, doordat de software niet vaak wordt gebruikt wordt de software niet vaak geupdate en ontstaat er een veiligheidsrisico.

Gewenste situatie: gebruiker (bijvoorbeeld bedrijf) geeft aan welke software hij gebruikt en welke versie hij er momenteel van heeft. Als er een nieuwere softwareversie is krijgt de gebruiker een mailtje met daarin de info (welke software moet worden geupdate en naar welke versie) de gebruiker kan vervolgens aangeven dat hij hem heeft geupdate naar de nieuwste versie. Op die manier weet het systeem altijd de juiste versie die de gebruiker heeft.

Als de klant niet betaald voor het product kan hij maar twee software systemen toevoegen aan de update tracker en krijgt hij maar één mailtje per maand. Als de gebruiker wel betaald kan hij unlimited software systemen toevoegen en krijgt hij elke dag een mailtje als dat nodig is.

# 

# Doelen

Hier geeft ik aan wat het doel is met de uitvoering van het project. Omdat het project uit meerdere fases bestaat zal ik per fase beschrijven wat er opgeleverd en gedaan wordt.

## Fase 1 Projectplan

In deze fase maak ik het projectplan (projectomschrijving). Het doel is dat ik het project zo goed mogelijk beschrijf en documeer. Het projectplan bestaat uit een aanleiding , doelen, resultaat, afbakening, planning, risico’s en randvoorwaarden.

## Fase 2 Functioneel & technische ontwerp

In fase 2 maak ik het functioneel en technisch ontwerp. In het functioneel ontwerp wordt beschreven wat de applicatie aan functionaliteit moet bieden. In het functioneel ontwerp staan de use-case-diagrammen, use-case-tabellen, klasse diagrammen als er gebruik word gemaak van klassen en het functioneel-ontwerp rapport. Verder maak ik wireframes(om de styling af te stemmen met de leraar) om een idee te krijgen welke functionaliteit op welke pagina komt te staan en hoe het er ongeveer uit gaat zien.

## Fase 3: Sprint 1

In de eerste sprint zet ik het project skelet op. Dat houd in dat ik een kleine opzet heb om mee te beginnen zodat ik het later kan uitbreiden. Ook maak ik een inlogpagina, Mogelijkheid voor users om software toe te voegen aan de lijst en connectie met de api.

## Fase 4: Sprint 2

In de tweede sprint maak ik de overzichtspagina met de volgende functionaliteiten: inschrijven voor systeem(registreren), so ftware toevoegen, checken of er een nieuwe versie is van de software. Ook maak ik het testplan en het testrapport in sprint 2.

## Fase 5: Sprint 3

In sprint 3 maak ik de mail functionaliteiten zoals zorgen dat de mail niet in spam beland, mail juiste data meegeven en een knop maken dat huidige versie van gebruiker zijn software wordt geupdate.

## Fase 6: Sprint 4

In sprint 4 doe ik de styling van de beheerders pagina, de overzichtspagina en de e-mail verder test ik of het product goed functioneert. Ook wordt de opleverdocumentatie gemaakt in sprint 4.

## Fase 7: Eindgesprek

In de laatste fase heb ik een eindgesprek met de opdrachtgever en word ik beoordeeld op de kwaliteit van mijn opgeleverde product.

# 

# Resultaat

Het project is geslaagd als gebruikers gemakkelijk hun software up to date kan houden. De gebruiker wordt op de hoogte gehouden door middel van een mailtje. In het mailtje zal de meest recente versie van de software staan, op die manier kan de gebruiker zijn software gelijk naar de juiste versie updaten. De meest recente versies van de software zal opgehaald worden uit een externe bron (API). Het systeem controleert of de api gegevens (softwareversies) overeenkomen met de gegevens die in de database staan. Als ze overeenkomen zal het systeem een mailtje sturen naar alle gebruikers die de software in de update tracker heeft gezet.

Het project is ingedeeld in verschillende fases. Iedere fase wordt iets gemaakt en opgeleverd. Ik zal hieronder aangeven wat in welke fase wordt opgeleverd.

## Fase 1: het project plan

Het projectplan is samengesteld om goed in kaart te brengen wat mijn opdracht is voor het maken van een update tracker ten behoeve van bedrijven (en particulieren). In dit projectplan leg ik voor mijn docenten Frans de Boer en Richard kingma (hierna te noemen: opdrachtgevers) vast waar dit projectplan betrekking op heeft.

## Fase 2: Functioneel en technisch ontwerp

Het technisch ontwerp bevat informatie over de technische aspecten van ons project: de Updatetracker.

Het functioneel ontwerp bevat informatie over de functionele aspecten van ons project: de Update Tracker.

## Fase 3: sprint 1

In sprint 1 maak ik een inlogpagina, Mogelijkheid voor users om software toe te voegen aan de lijst en connectie met de api.

## Fase 4: sprint 2

In sprint 2 maak ik de overzichtspagina met de volgende functionaliteiten: inschrijven voor systeem(registreren), so ftware toevoegen, checken of er een nieuwe versie is van de software. Ook maak ik het testplan en het testrapport in sprint 2.

## Fase 5: Sprint 3

In sprint 3 maak ik de mail functionaliteiten zoals zorgen dat de mail niet in spam beland, mail juiste data meegeven en een knop maken dat huidige versie van gebruiker zijn software wordt geupdate.

## Fase 6: sprint 4

In sprint 4 maak ik de styling van de beheerders pagina, de overzichtspagina en de e-mail verder test ik of het product goed functioneert. Ook wordt de opleverdocumentatie gemaakt in sprint 4.

## Fase 7: eindgesprek

## In de laatste fase lever ik het product op en heb ik een eindgesprek met de opdrachtgever. Ook word ik beoordeeld op de kwaliteit van mijn opgeleverde product.

# 

# Afbakening

MosCoW is een samenstelling van beginletters die ergens voor staan. De 2x o zijn ingevoegd om het woord ‘moscow’ leesbaar te maken, zonder dat ze betekenis hebben. De **M** staat voor **Must-haves**, de **S** staat voor **Should-haves**, de **C** staat voor **Could-haves** en de **W** staat voor **Won’t-haves**. De ‘moscow’ analyse gebaseerd op de vragen die eerder zijn gesteld aan de opdrachtgever.

### Must haves:

* Account aanmaken
  + De gebruiker voert zijn e mail, username, wachtwoord en wachtwoord in. vervolgens drukt de user op sign up. Als de user 2 keer hetzelfde wachtwoord heeft ingevuld wordt er een account aangemaakt. Anders krijgt de gebruiker een foutmelding en wordt hem gevraagd de gegevens opnieuw in te loggen. Als de gebruiker succesvol een account heeft aangemaakt wordt de gebruiker doorgestuurd naar de inlogpagina.
* Inloggen
  + Als de gebruiker in het bezit is van een account kan hij/zij zijn username en password invullen. Vervolgens kan de gebruiker op login drukken als de gegevens juist zijn dan wordt er een sessie aangemaakt en kan de gebruiker in het systeem. Anders krijgt de gebruiker een foutmelding.
* Uitloggen
  + Als de gebruiker is ingelogd dan kan de gebruiker op de uitlog knop drukken. Vervolgens beëindigd het systeem de sessie en stuurt het systeem de gebruiker door naar de inlogpagina.
* Add software
  + Als de gebruiker is ingelogd kan de user software toevoegen. De gebruiker vult software naam en softwareversie in. vervolgens drukt de gebruiker op voeg toe. Als de gebruiker geen velden heeft leeg gelaten wordt de software toegevoegd en wordt de gebruiker doorgestuurd naar de home pagina. Anders krijgt de gebruiker een foutmelding.
* Remove user
  + Als de gebruiker is ingelogd kan de user software verwijderen. De gebruiker drukt op delete en krijgt de vraag weet u het zeker. Als de user op ja drukt wordt de software verwijderd en wordt de user doorgestuurd naar de home pagina. Anders wordt de software niet verwijderd.
* Edit software
  + Als de gebruiker is ingelogd kan de user software aanpassen. De user drukt op edit en wordt doorgestuurd naar edit pagina. De user veranderd de software naam en/of software versie en drukt op opslaan. Als alle velden gevuld zijn wordt de verandering opgeslagen. Anders krijgt de gebruiker een foutmelding en moet de gebruiker het opnieuw proberen.
* Edit user (gebruiker)
  + Als de gebruiker is ingelogd kan de user zijn eigen account data aanpassen. De user drukt op profiel en wordt doorgestuurd naar zijn profielpagina. De user veranderd de usernaam en/of wachtwoord en drukt op opslaan. Als alle velden gevuld zijn wordt de verandering opgeslagen. Anders krijgt de gebruiker een foutmelding en moet de gebruiker het opnieuw proberen.
* Edit user (admin)
  + Als de gebruiker is ingelogd als admin kan de gebruiker iemand anders zijn account aanpassen. De gebruiker drukt op edit en wordt doorgestuurd naar een aanpas pagina. De gebruiker veranderd de usernaam en/of wachtwoord en drukt op opslaan. Als alle velden gevuld zijn wordt de verandering opgeslagen. Anders krijgt de gebruiker een foutmelding en moet de gebruiker het opnieuw proberen.
* Toon data
  + Als de gebruiker is ingelogd kan de gebruiker data aanvragen bij de database.
* Get software update
  + Als de gebruiker zijn software een update heeft dan krijgt de gebruiker een mailtje. Vervolgens update de user en geeft door aan het systeem dat hij/zij heeft geupdate. User zijn huidige softwareversie wordt geupdate. Als de user niet aangeeft dat hij /zij heeft geupdate update het systeem de huidige softwareversie niet.
* Api data ophalen
  + Het systeem haalt 1 keer per dag data uit de api. En slaat het op in de database
* Controleren of user software update heeft
  + Het systeem controleert 1 keer per dag of een user zijn huidige software een update heeft.
* Mail versturen
  + Het systeem verstuurt 1 keer per dag een mailtje met info waarin staat welke updates een user moet doen. In het mailtje staat ook een linkje daar kan een user op klikken om aan te geven dat de user zijn software heeft geupdate.

### Should haves:

* Duidelijke en gebruiksvriendelijke styling (gebruiker moet zijn weg goed kunnen vinden)
  + Het is belangrijk dat de styling gebruiksvriendelijk is. Zonder gebruiksvriendelijke styling kan de gebruiker een slecht gevoel krijgen bij het gebruiken van de website. Om er achter te komen wat gebruiksvriendelijke styling is ga ik het testen in de testperiode.

### Could haves:

* Styling na eigen smaak (wel overleggen met opdrachtgever of hij het goed keurt)
  + Styling is uitgetekend in adobe xd zie functioneel ontwerp.

### Won’t haves:

# 

# Planning

Vanwege de grootte van mijn planning heb ik een [link](https://docs.google.com/spreadsheets/d/1oiJdzhOwYDzeaYQ5daNwdl1lzndIGIp0lKTxuA57Ok4/edit?usp=sharing) naar mijn planning, die kunt u online bekijken zodat u een goed beeld heeft van mijn taakverdeling. De planning is onderverdeeld in een documenteer week en verschillende programmer sprints.

# 

# Effecten

Het project zal een groot effect hebben bij gebruikers die veel software gebruiken. Door het grote aantal software is het lastig om het allemaal up to date te houden. Met behulp van de update tracker is het gemakkelijk om de nieuwste versie van de software te hebben. De nieuwste versie van software betekent meestal ook de veiligste dus gebruikers hun systemen zullen ook veiliger worden.

Verder zal het een overzicht geven aan alle software die een gebruiker heeft/ gebruikt (als de gebruiker ze allemaal toevoegt).

# 

# Risico’s

In de risico’s worden alle risico’s vastgesteld die duidelijk op papier moeten staan zodat het duidelijk is waar het project kan vastlopen.

|  |  |  |
| --- | --- | --- |
| Risico | Effect op projectresultaat | Hoe te voorkomen |
| API van derden stopt met werken | Zonder API die data levert die ik kan gebruiken komt het hele project stil te staan | Het valt niet te voorkomen omdat ik afhankelijk ben van derden. Ik heb geen invloed op hun |

# 

# Randvoorwaarden

In de randvoorwaarden worden alle eisen gesteld die benodigd zijn voor een goede oplevering van het eindproduct.

* Data leverende API (van derden)
  + Zonder API is er geen data om te controleren of er nieuwe versie van software is.
* Een test team
  + Deze mensen gaan mijn product testen om te zorgen dat mijn product geen onverwachte fouten of bugs geeft. Voor een optimaal resultaat moeten dit mensen zijn die niks met dit project te maken hebben.
* Een kwalitatief goede hosting
  + Dit is een goede manier om ons product online te testen op bijvoorbeeld beveiligingsfouten en andere obstakels.
  + Ook is het belangrijk dat de hosting mail mogelijkheid heeft.