

Software S-DB-S2

Individueel project verslag

Thijmen Brand | 480490 | 07-02-2022

# 

# Inleiding

Inhoud

[Inleiding 1](#_Toc97734981)

[One-slide idee. 4](#_Toc97734982)

[Project beschrijving 5](#_Toc97734983)

[I. Introductie 5](#_Toc97734984)

[1.1 doel 5](#_Toc97734985)

[1.2 Publiek 5](#_Toc97734986)

[1.3 Project omvang 5](#_Toc97734987)

[2 Algemene beschrijving 5](#_Toc97734988)

[2.1 Product details 5](#_Toc97734989)

[2.2 Requirements 5](#_Toc97734990)

[2.3 Requirements 8](#_Toc97734991)

[Must have 8](#_Toc97734992)

[Should have 8](#_Toc97734993)

[Could have 10](#_Toc97734994)

[Minimal viable product 11](#_Toc97734995)

[Planning 12](#_Toc97734996)

[Sprint 1 12](#_Toc97734997)

[Sprint 2 12](#_Toc97734998)

[Sprint 3 12](#_Toc97734999)

[Sprint 4 12](#_Toc97735000)

[Sprint 5 13](#_Toc97735001)

[Sprint 6 13](#_Toc97735002)

[Planning 2 13](#_Toc97735003)

[Sprint 1 13](#_Toc97735004)

[Week 1 13](#_Toc97735005)

[Week 2 13](#_Toc97735006)

[week 3 13](#_Toc97735007)

[Sprint 2 13](#_Toc97735008)

[Week 1 13](#_Toc97735009)

[Week 2 13](#_Toc97735010)

[Week 3 14](#_Toc97735011)

[Sprint 3 14](#_Toc97735012)

[Week 1 14](#_Toc97735013)

[Week 2 & 3 14](#_Toc97735014)

[Week 1 14](#_Toc97735015)

[Week 2 14](#_Toc97735016)

[Week 3 14](#_Toc97735017)

[Sprint 5 14](#_Toc97735018)

[Week 1 14](#_Toc97735019)

[Week 2 15](#_Toc97735020)

[Week 3 15](#_Toc97735021)

[Sprint 6 15](#_Toc97735022)

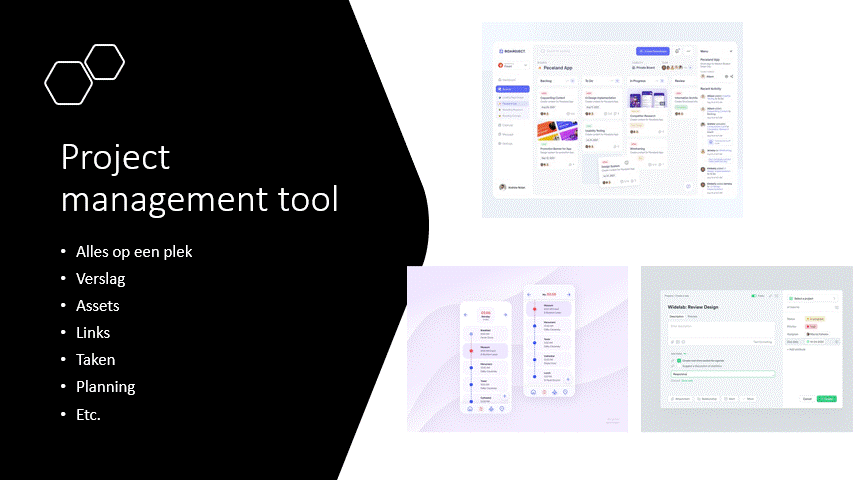
[Week 1&2&3 15](#_Toc97735023)

[Data 15](#_Toc97735024)

[Techniek onderbouwing 16](#_Toc97735025)

[Bronnen 17](#_Toc97735026)

# One-slide idee.

Een succesvol project begint altijd met een idee. Deze moet uiteindelijk duidelijk en helder zijn. Maar om te kunnen beginnen is het belangrijk om een globaal plan te schetsen. Dit niet een te veel detail of woorden. Hier komt een one-slide pitch goed van pas. Bij deze aanpak wordt je gedwongen in een korte vorm te formuleren wat je wilt gaan doen.

Met dit idee wil ik het process van het starten en onderhouden van een project gemakkelijker maken. Ik heb gemerkt bij mijn eigen projecten dat ik op veel verschillende plekker resources heb staan. Zo heb ik een word document voor mijn verslag, een ander document voor mijn documentatie. Dan sla ik in mijn brouwser of youtube handige links en video’s en mijn taken staan in trello. Dit zijn een heleboel verschillende plekken waardoor ik het af en toe moeilijk vind het overzicht te houden over wat waar staat. En dit probleem wil ik oplossen door het hele project process op een plek te combineren. Hier kan ik dus een verslag maken, Maar ook kan ik er andere bestanden zoals fotos, code of documentatie opslaan en bewerken. Daarnaast kan ik handige links of assets opslaan en makkelijk terug vinden, ik kan mijn taken handig op een rijtje zetten onder to-do, doing of done. En ik kan een dag/ week/ sprint planning maken.

# Project beschrijving

## 1. Introductie

### 1.1. doel

Het doel van dit document is het bouwen van een project management systeem waar meerdere aspecten van het ontwikkelen van een software project gebundeld kunnen worden op een plek om de gemakkelijkheid en het proces te versnellen.

### 1.2. Publiek

Deze applicatie is vooral gericht op software developers die een nieuw software project starten. Echter kan het ook gebruikt worden door menig ander persoon die verschillende resources en bestanden wil bundelen in een project.

### 1.3. Project omvang

De doel van het project management systeem is er voor zorgen dat alle bestanden en resources die bij een bepaald project horen overzichtelijk op een plek staan opgeslagen in plaats van verspreid over verschillende applicaties en mappen. De software zal worden gebaseerd op een relationele SQL database waar ieder project met zijn verschillende data in opgeslagen wordt.

## 2. Algemene beschrijving

### 2.1. Product details

De projectdatabase zal aan de hand van verschillende tabellen die een onderlinge relatie delen de data van verschillende projecten opslaan. Deze projecten hebben een ID die zal worden gebruikt binnen alle andere tabellen als foreign key.

### 2.2. Requirements

#### Project verslag

De gebruiken kan een verslag opstellen van het project. Hierin kan onderzoek in worden gedocumenteerd of iets anders. Dit is een plain tekst document waar verschillende tekst stijlen kunnen voorkomen. Alles kan hierin getypt worden zonder dat het door de server wordt uitgevoerd. Dit elimineert de optie om foutieve code in de website applicatie te injecteren. Ook bevat het de functie om automatisch op te slaan wanneer er een bewerking plaatsvind. Wanneer het opslaan of het uitlezen van het document mislukt zal er een duidelijke rode foutbalk verschijnen boven in het scherm met een unieke foutcode en de tekst die gekoppeld is aan deze unieke foutcode.

#### Taken

De gebruiker moet de mogelijkheid hebben om verschillende taken aan te maken. Naast taken moet hij ook statussen voor deze taken aan kunnen maken, bijvoorbeeld to-do, doing en done. Deze taken kunnen een deadline hebben en deze deadlines kunnen aan een agenda worden bekeken. Ook kan een gebruiker een beschrijving bij een taak zetten.

De taak namen mogen geen speciale tekens bevatten bahalve vraagtekens en uitroeptekens. Verder mogen de titels een beperkte lengte van 50 tekens bevatten. Wanneer de gebruiker meer informatie bij een taak wil opslaan zal dit in de beschrijving gezet moeten worden. De beschrijving mag ook geen special tekens bevatten behalve een uitroepteken of een vraagteken.

Wanneer een taak niet aangemaakt kan worden door bekende redenen, bijvoorbeeld een verkeerde datum (die al is verlopen) of een verkeerde titel of beschrijving zal dit direc tijdens de invoer duidelijk worden gemaakt door middel van rode letters onder de invoer. Wanneer het niet aangemaakt of bewerkt kan worden door onbekende redenen zal er een rode balk verschijnen binnen de modal die wederom een error code geeft met de corosponderende tekst.

#### Files

Een gebruiker moet verschillende bestanden kunnen uploaden. Deze bestanden moeten daarnaast te in mappen te plaatsen zijn om ze te ordenen. Ook moet de gebruiker zelf tekst bestanden kunnen aanmaken en deze moeten net als het project verslag voorzien zijn van verschillende tekst stijlen. En ten derde moet een gebruiker bestanden kunnen verwijderen. Daarnaast moet de naam van een bestand aangepast kunnen worden en bestaande bestanden moeten bewerkt kunnen worden. Aan de hand van het mime type zal er een icon worden gekozen voor het bestand.

De namen van de bestanden en mappen mogen geen speciale tekens bevatten. Behalve een punt voor het Mime type. Binnen de file mag alles geplaats worden. Dit zal door de server niet geïnterpreteerd worden als code en daardoor ook niet uitgevoerd. Bestanden mogen enkel een beperkte grootte hebben, op deze manier zal de opslag niet te snel vol staan en zal het uploaden ook sneller gaan.

Wanneer er een foutieve titel aan een map of bestand word gegeven zal dit bij de invoer al duidelijk gemaakt worden doormiddel van een rode tekst onder de invoer balk. Wanneer een bestand te groot is zal er bij het bestand upload scherm een rode tekst verschijnen met de desbetreffende error. Wanneer er een onbekende fout is bij het uploaden zal dit door middel van een rode balk bekend worden gemaakt met een error code en een korte beschrijving van de code.

#### Beschrijving

Een project kan een beschrijving hebben. Deze word enkel weergegeven bij het project overzicht.

Een gebruiker kan hier geen tekens invoeren die kunnen leiden tot foutieve code. Wanneer dit wel het geval is zal er geen error komen. Maar zal de tekst zonder speciale tekens worden opgeslagen.

Wanneer er een onbekende fout optreed bij het opslaan zal dit door middel van een rode balk boven aan het scherm bekend worden gemaakt Wederom met een error code en een korte beschrijving hierbij.

#### Links

Een gebruiker moet de mogelijkheid hebben om links van webpagina’s op te slaan. Wanneer er op een van deze links geklikt wordt zal hij doorverwezen worden in een nieuw tabblad naar de desbetreffende internet pagina. Deze links moeten ook weer verwijderd of aangepast moeten kunnen worden.

Hier mogen enkel teksten worden ingevoerd die een valide url zijn. Wanneer dit niet het geval is kan de link niet worden opgeslagen en zal er onder de link invoer een rode tekst verschijnen met de vraag een valide url in te vullen. Ook de naam van het item mag geen speciale tekens bevatten en dit word direct bij invoer gecheckt. Wanneer er een foutieve invoer is zal dit direct onder de invoer bekend worden gemaakt door middel van een rode tekst.

#### Herinneringen

Een gebruiker moet herinneringen aan kunnen maken. Deze kan dan op interval en tijd worden ingesteld om aan het project te werken, of om voor een bepaalde tijd een taak af te hebben. Daarnaast moet er een deadline herinnering ingesteld kunnen worden.

De data die bij deze herinneringen word ingevuld moeten in de toekomst liggen. Wanneer dit niet het geval is zal er direct een foutmelding onder de invoeg gegeven worden in rode tekst. Wanneer er door een onbekende fout geen herinnering kan worden toegevoegd zal er boven aan de pagina een rode foutbalk worden weer gegeven met een foutcode en een tekst.

#### Sprints

Een gebruiker moet sprints kunnen aanmaken. Deze sprint kan dan een duur hebben waarna de volgende sprint start tot dat het project ten einde loopt. Aan deze sprint kunnen van taken worden toebedeeld die aan het einde van de sprint af moeten zijn.

Een sprint duur mag geen negatieve waarde hebben. Ook mag het niet boven een bepaalde waarden liggen. Wanneer dit wel het geval is krijg een direct een rode foutmelding.

#### Agenda

Een gebruiker moet de mogelijkheid hebben om alle data die hij op slaat bij bijvoorbeeld taken, herinneringen en sprints in een agenda te zien. Ook kan de gebruiker van zelf agenda items aanmaken met weer individuele herinneringen.

## 2.3. Requirements

### Must have

* Een gebruiker kan inloggen
  + Als project manager wil ik in kunnen loggen om mijn projecten te beschermen tegen de buitenwereld.
* Een gebruiker kan projecten aanmaken
  + Als project manager wil ik een nieuw project aan kunnen maken om project details in op te slaan
* Een gebruiker kan projecten verwijderen
  + Als project manager wil ik een project die niet meer gebruikt wordt of waar de development mee gestopt is kunnen verwijderen om mijn account schoon en opgeruimd te houden.
* Een gebruiker kan een project een naam geven
  + Als project manager wil ik een project een naam geven om te weten wat het project is en om het te onderscheiden van andere projecten
* Een gebruiker kan sprints een duur geven
  + Als project manager wil ik de mogelijkheid hebben om sprints een bepaalde duur te geven om zo mijn taken goed geordend in het tijdsbestek wat ik wil te kunnen doen
* Een gebruiker kan taken aanmaken
  + Als project manager wil ik taken aan kunnen maken om op een rij te zetten wat er allemaal moet gebeuren
* Een gebruiker kan taken bewerken
  + Als project manager of developer wil ik de naam van een taak kunnen bewerken om eventuele fouten op te lossen
* Een gebruiker kan taken verwijderen
  + Als project manager wil ik onnodige taken of outdated taken kunnen verwijderen om de werkomgeving schoon te houden.
* Een gebruiker kan taken in sprints inplannen
  + Als project manager wil ik taken in kunnen plannen in sprint planningen om te weten wanneer de taak aan de beurt komt.
* Een gebruiker kan taken uit de planning verwijderen
  + Als project manager wil ik taken uit te planning kunnen verwijderen om eventuele fouten op te lossen of om oude taken weg te halen.
* Een gebruiker kan van sprint weergave wijzigen, uur planning / dag planning / week planning / sprint duur planning
  + Als project manager of developer wil ik de weergaven van mijn planning kunnen veranderen om een duidelijker beeld te krijgen van de werkzaamheden en taken

### Should have

* Een gebruiker kan een project een beschrijving geven
  + Als project manager wil ik een project een beschrijving kunnen geven om een beter beeld te krijgen en te geven aan developers over wat het project precies inhoudt
* Een gebruiker kan een project naam veranderen
  + Als project manager wil ik de naam van mijn project kunnen veranderen om eventuele fouten op te lossen, of veranderingen door te voeren
* Een gebruiker kan een project beschrijving veranderen
  + Als project manager wil ik de beschrijving van een project veranderen om eventuele fouten op te lossen of om een aanpassing te maken wanneer de project details veranderen
* Een gebruiker kan bestanden verwijderen
  + Als project manager wil ik een bestand verwijderen om verwarring met onnodige documenten te voorkomen
* Een gebruiker kan tekst bestanden aanmaken
  + Als project manager wil ik bestanden kunnen aanmaken om informatie globaal bereikbaar te hebben.
* Een gebruiker kan links opslaan met titel
  + Als developer of project manager wil ik een url kunnen opslaan om later te kunnen gebruiker of om een handige pagina op te slaan voor later
* Een gebruiker kan links en titels aanpassen
  + Als developer of project manager wil ik de titel van een link kunnen aanpassen om eventuele fouten op te lossen
* Een gebruiker kan links verwijderen
  + Als developer of project manager wil ik een link kunnen verwijderen om al gebruikte of niet meer relevante informatie op te schonen
* Een gebruiker kan taken een beschrijving geven
  + Als project manager of developer wil ik een taak een beschrijving kunnen geven om een duidelijker beeld te krijgen wat een specifieke taak inhoudt
* Een gebruiker kan taken een deadline geven
  + Als project manager wil ik een taak een deadline geven om te weten wanneer iets af moet zien.
* Een gebruiker kan taken onder verdelen in containers
  + Als project manager of developer wil ik taken kunnen schuiven tussen containers wanner de status is veranderd
* Een gebruiker kan taak containers aanmaken en een naam geven
  + Als project manager wil ik verschillende taak container aanmaken om taken te ordenen in status
* Een gebruiker kan taak weergave wisselen tussen container en lijst
  + Als project manager of developer wil ik de weergaven van mijn taken kunnen wisselen om een overzichtelijk beeld te krijgen van de taken.
* Een gebruiker kan bij een taak zien of en wanneer deze staat ingepland
  + Als project manager of developer wil ik bij een taak kunnen zien wanneer en of deze staat ingepland om te weten of het nog ingepland moet worden en wanneer deze aan de beurt komt.

### Could have

* Een gebruiker kan alle soorten bestanden opslaan
  + Als gebruiker wil ik handige bestanden kunnen opslaan om zo documentatie te schrijven of belangrijke bestanden globaal bereikbaar te hebben.
* Een gebruiker kan bestanden bekijken
  + Als project manager of developer wil een opgeslagen bestanden bekijken om informatie over het project of andere project gerelateerde documenten te lezen
* Een gebruiker kan tekst bestanden bewerken
  + Als project manager wil ik een bestand kunnen bewerken om mijn developers de meest actuele informatie te geven
* Een gebruiker kan mappen aanmaken
  + Als project manager wil ik mappen aan kunnen maken om een overzichtelijke en schonen werk omgeving te creëren
* Een gebruiker kan bestanden ordenen in mappen
  + Als project manager wil ik bestanden in mappen kunnen plaatsen om een overzichtelijke omgeving te creëren
* Een gebruiker kan mappen met alle bestanden erin verwijderen
  + Als project manager wil ik een onnodige map kunnen verwijderen om verwarring met informatie te voorkomen
* Een gebruiker kan een requirements document exporteren naar taken
  + Als project manager wil ik requirements en user stories kunnen exporteren naar taken om op deze manier makkelijk en snel een planning te kunnen maken
* Een gebruiker kan taken een kleur tag geven
  + Als project manager of developer wil een taak een tag kunnen geven om te weten in welke categorie deze valt
* Een gebruiker kan kleur tags aanmaken
  + Als project manager wil ik tags aan kunnen maken om categoriën bij te houden
* Een gebruiker kan bijhouden hoe lang een activiteit duurt
  + Als developer wil ik kunnen bijhouden hoelang iets duurt om in de toekomst een beter beeld te krijgen over te tijd wat dingen innemen
* Een gebruiker kan aan deze meting een naam geven
  + Als developer wil ik een naam aan deze meting geven om duidelijk te weten waar de meting voor was
* Een gebruiker kan aan deze meting een beschrijving geven
  + Als developer wil ik een beschrijving aan de meting geven om een gedetailleerde beschrijving te krijgen van waar de meting voor was en of er enige bijzonderheden waren.
* Een gebruiker kan aan deze meting een taak hangen
  + Als developer wil ik een bepaalde taak aan de meting kunnen handen om inzicht te krijgen in hoelang een bepaalde taak duurt.
* Een gebruiker kan een project Wiki aanmaken
  + Als project manager wil ik een project Wiki kunnen maken om te documenteren of een gedetailleerde beschrijving te geven van het project
* Een gebruiker kan verschillende pagina’s binnen de Wiki aanmaken
  + Als project manager wil ik verschillende pagina’s aan de wiki hangen om verschillende onderwerpen toe te lichten
* Een gebruiker kan tekst, afbeeldingen en code blokken aan de pagina’s toevoegen
  + Als project manager wil ik verschillende elementen aan de wiki toevoegen om een duidelijk beeld te geven

### Minimal viable product

* Een gebruiker kan een project aanmaken
* Een gebruiker kan een sprints planning zien
* Een gebruiker kan een taak maken
* Een gebruiker kan een taak aan een sprint toevoegen

#### Non-functionele requirements

* De applicatie staat 3 inlog pogingen toe
* Wanneer de inlog pogingen op zijn wordt het account geblokkeerd
* Wanneer het account geblokkeerd is zal er een reset mail naar het geregistreerde mail adres gestuurd worden
* De applicatie hoeft enkel op desktop te functioneren
* Wanneer er een bekende fout op treed zal deze duidelijk getoond worden met beschrijving wat er fout ging.
* Wanneer er een onbekende fout op treed zal de foutcode getoond worden waarmee contact gelegd kan worden met de admin
* Een gebruiker ken .txt bestanden openen

# Sprint planning

## Sprint 1

### Week 1

* Intro bootcamp
* Project idee bedenken
* Project idee beschrijven
* Project requirements stellen
* Planning maken

### Week 2

* Prototype: Project aanmaken
* Prototype: Sprints plannen
* Prototype: Taken maken

### week 3

* Unit test: Project aanmaken
* Unit test: Sprints plannen
* Unit test: Taken maken

## Sprint 2

### Week 1

* Development: Inloggen / registreren
* Unit test: Inloggen / registreren
* Development: Project Aanmaken / Verwijderen / Aanpassen
* Unit test: Project Aanmaken / Verwijderen / Aanpassen

### Week 2

* Development: Bestanden uploaden / verwijderen
* Unit test: Bestanden uploaden / verwijderen

### Week 3

* Development: Bestanden Aanmaken / bestanden bewerken
* Unit test: Bestanden Aanmaken / bestanden bewerken

## Sprint 3

### Week 1

* Development: Mappen aanmaken
* Unit test: Mappen aanmaken
* Development: Bestanden ordenen in mappen
* Unit test: Bestanden ordenen in mappen

### Week 2 & 3

* Development: Sprint planning
* Unit test: Sprint planning
* Development: Sprint weergave
* Development: Taken toevoegen aan sprint planningSprint 4

### Week 1

* Development: Taken aanmaken / verwijderen
* Unit test: Taken aanmaken / verwijderen
* Development: Taken bewerken / duur geven / due date koppelen
* Unit test: Taken bewerken / duur geven / due date koppelen

### Week 2

* Development: Requirements document exporteren naar taken
* Unit test: Requirements document exporteren naar taken
* Development: Taken toevoegen aan sprint planning
* Unit test: Taken toevoegen aan sprint planning

### Week 3

* Development: Time tracking
* Unit test: Time tracking
* Development: Time tracking opslaan en naam geven
* Unit test: Time tracking opslaan en naam geven

## Sprint 5

### Week 1

* Development: Time tracking weergeven in schema
* Development: Time tracking koppelen aan taak
* Unit test: Time tracking koppelen aan taak

### Week 2

* Development: Project wiki aanmaken
* Unit test: Project wiki aanmaken
* Development: Verschillende pagina’s koppelen aan project wiki
* Unit test: Verschillende pagina’s koppelen aan project wiki

### Week 3

* Development: Tekst document als project wiki instellen
* Unit test: Tekst document als project wiki instellen
* Development: Integratie tussen verschillende componenten polishen

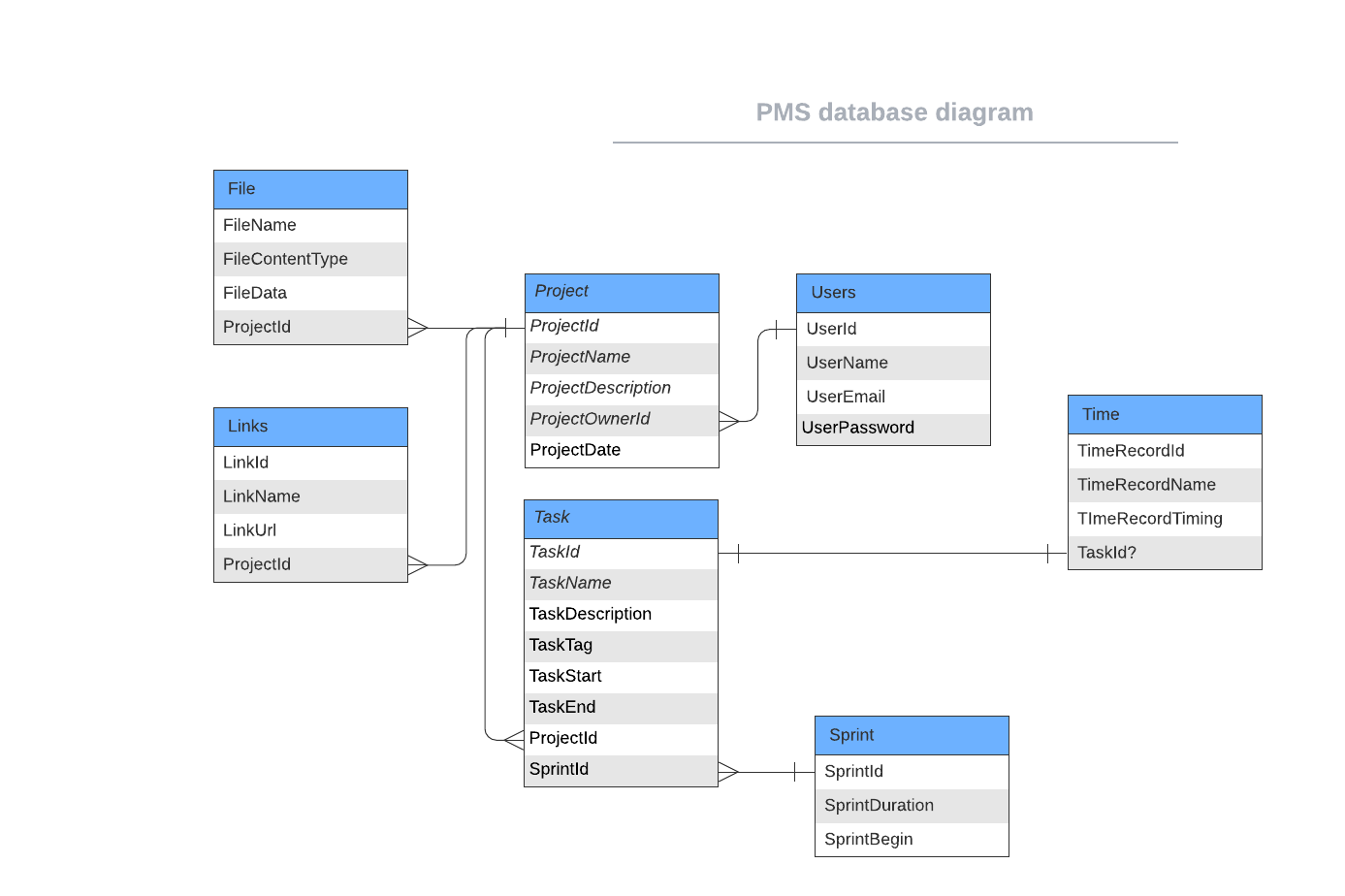
## Sprint 6

### Week 1&2&3

* Bugs fixen, refractoren en presentatie voorbereiden.

# Data

Om alle functionaliteiten goed tot zijn recht te laten komen is het van belang goed te weten hoe verschillende data typen en sources tot elkaar in verband staan. Hiervoor heb ik een UML diagram gemaakt.



# Begrippen

PMS: Project management systeem. Je project waar ik momenteel aan werk en waar je verschillende projecten kan beheren.

# Techniek onderbouwing

De applicatie zal gaan bestaan uit een front- en een backend. Deze twee componenten zullen strikt gescheiden blijven. Dit is voordelig omdat op deze manier de frontend makkelijk aan te passen of te updaten is zonder dat dit complicaties met zich mee brengt. De enige manier waarop deze twee lagen namelijk afhankelijk van elkaar zijn is een http endpoint die makkelijk te veranderen zou zijn. Wanneer deze manier niet toegepast zou worden en de front- en backend grotendeels met elkaar verweven is, word het heel lastig om iets makkelijk aan te passen aan een van beide delen.

Als frontend techniek werk ik met Vue.js. Dit is een javascript library welke zich focust op de het zo simpel mogelijk maken maar toch vele features bieden. Zo zijn er duizenden packages voor te vinden en is het een no-config of low-config framework. Dit houdt in dat alles voor je wordt geregeld voor de vue CLI tool. Deze installeert de juiste packages en configureert ze naar jouw behoefte. En wanneer je toch iets meer wil aanpassen kan dit in de Vue.Config file. Daarnaast maakt hebt gebruik van html templates in combinatie met JSX. Dit is een voordeel omdat het duidelijker is om te lezen maar toch krachtig en snel is. Daarnaast is de documentatie erg uitgebreid wat het makkelijk maakt om met vue te beginnen.

# Backend api

De backend van PMS bestaat is gebouwd als een restful web-api met verschillende lagen om op deze manier de kwaliteit en onderhoudtbaarheid van de applicatie te waarborgen. De eerste laag is de Controller laag, deze ontvangt ieder http request met de bijbehorende data, hier wordt de data gefilterd en gecontroleerd. Vervolgens gaat de data naar de Business laag waar de data verder gefilterd en gevormd word naar de vorm waarop het bruikbaar is. Wanneer dit is gebeurt word de data wederom door gestuurd naar de laatste laag en dit is de data laag. Deze voegt met wat laatste checks de data toe aan de database. Dit dezelfde proces werkt ook omgekeerd met wanneer er data wordt opgevraagd.

De Controller laag: Deze laag heeft een aantal verschillende controllers. Voor iedere hoofd functionaliteit een. Zo is er bijvoorbeeld een project controller, een taak controller en een sprint controller welke allemaal zorgen voor zijn eigen taken.

De “SprintsController.cs” heeft twee functies welke ook endpoints zijn. Dit zijn “AddSprint” en “GetSprintsByProject” De namelijk speken voor zich.

De “TasksController.cs” heeft vier functies welke wederom ook endpoints zijn. Deze zijn “AddTask”

# Unit tests

Om de kwaliteit van mijn software te garanderen schrijf ik voor iedere backend functie een unit test. Dit doe ik echter alleen voor mijn business layer waar ik de logica af handel. De andere lagen maken namelijk gebruik van standaard functies en geven alleen maar data door waardoor er niks fout kan gaan.

Ik gebruik voor mijn unit tests twee verschillende libraries. Dit zijn Moq en NUnit. De Moq library zorgt ervoor dat ik gemakkelijk een depandancy kan isoleren. Dit is nodig zodat mijn functie die een test ondergaat geen andere dependancies in de applicatie gebruikt (24-03-2009, Van Fosson, T.).

# Terug blik op sprint twee

Ik begon sprint twee met een al staande web app. Ik kon projecten aanmaken, ophalen en selecteren. Echter was ik niet tevreden met de opzet van mijn backend waardoor ik deze besloot te veranderen. Wat ik namelijk in het verleden had was een eenlaagse structuur waar een request vanuit de api binnen kwam en direct in de database werd gezet. Dit is natuurlijk geen goede aanpak en heb dit dan ook veranderd naar een drie laags model waarbij iedere laag een aparte taak heeft. Zo komt er een request binnen bij de API laag. Vanuit hier gaat de data naar de tweede laag waar er wordt gekeken of de data correct is, de data wordt gevormd naar een vorm dat in de database past en andere benodigde dingen worden met de data gedaan. Wanneer dit allemaal klopt gaat de data naar de laatste laag; de data laag. En deze zorgt er enkel voor dat de data in de database terecht komt.

Na dit te hebben geïmplementeerd heb ik me vooral gefocust op de front-end van mijn applicatie. Ik heb deze sprint de taak pagina geïmplementeerd met alle bijbehorende functies. Dit is niet helemaal wat ik voor deze sprint voor ogen had om te doen. Ik had verwacht dat ik sneller verschillende features zou kunnen implementeren, maar dit viel helaas tegen. Daarom zal ik mijn planning ook aanpassen naar deze nieuwe inzichten.

Ook liep ik tegen een ander probleem aan. Dit was namelijk dat ik unit testen heel erg stom vond en het niet graag deed en daarnaast was ik zo gefocust op de front-end dat ik on-the-go verschillende backend features implementeerde waardoor ik unit testen totaal over het hoofd zag. Dit is iets wat ik ook in de volgende sprint bewuster mee bezig wil zijn en al het achterstallig werk wil inhalen.

# Sprint planning V2

## Sprint 2

### Week 1

* Development: Sprint instellingen
* Unit test: Taken / Sprint

### Week 2

* Unit test: Inloggen / registreren

### Week 3

* Development: Inloggen / registreren

## Sprint 3

### Week 1

* Development: agenda

### Week 2 & 3

* Development: agenda
* Unit test: agenda

## Sprint 4

### Week 1

* Development: Bestanden uploaden

### Week 2

* Development: Bestanden uploaden

### Week 3

* Development: bestanden uploaden
* Unit test: Bestanden uploaden

## Sprint 5

### Week 1

* Development: Time tracking

### Week 2

* Development: Time tracking
* Development: Settings

### Week 3

Unit test: Time tracking

Unit test: Settings

## Sprint 6

### Week 1&2&3

* Bugs fixen, refractoren en presentatie voorbereiden.

Voor uitgebreide taak beschrijvingen en kleinere stukken zie:

Github > ThijmenBrand/S02-Software-Individueel-Project > Projects > pms-planning

# Bronnen

AA01: W. (2021, 9 september). *9-Step Guide: How to Plan Effective Software Development Projects*. Waydev. Geraadpleegd op 7 februari 2022, van <https://waydev.co/software-projects-planning/>

AA02: Bandakkanavar, R. (2019, 17 oktober). Software Requirements Specification document with example. Krazytech. Geraadpleegd op 7 februari 2022, van <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>

AA03: Burak, A. (2021, 19 augustus). *How to Write a Software Requirement Specification (SRS) in 2021*. Relevant Software. Geraadpleegd op 7 februari 2022, van <https://relevant.software/blog/software-requirements-specification-srs-document/>

AA04: *Requirements*. (z.d.). s2-db-documentatie. Geraadpleegd op 8 februari 2022, van <https://rubensteins.github.io/s2-db-documentatie/Onderwerpen/Requirements>

AA05: Van Fosson, T. (2009, 24 maart). What is use of Moq? Stack Overflow. Geraadpleegd op 25 maart 2022, van <https://stackoverflow.com/questions/678878/what-is-use-of-moq#:%7E:text=Moq%20is%20a%20mocking%20framework,Wikipedia%20article%20on%20Mock%20Objects>.