# **Project: Cloud storage website**

## **Plan van Eisen**

In de komende weken ga ik een cloud storage systeem maken in een webomgeving. Dit gaat een omgeving hebben voor de gebruikers van de applicatie en een omgeving voor de beheerder van het systeem. Het doel is dat de gebruikers gemakkelijk bestanden kunnen opvragen en delen met anderen.

### Functionele eisen

* De gebruiker heeft alleen toegang tot het systeem met een geregistreerd account.
* De gebruiker kan bestanden uploaden, downloaden en verwijderen.
* De gebruiker kan al zijn bestanden zien in een overzicht.
* Bestanden kunnen alleen gedeeld worden met andere users met een geregistreerd account.
* Je kunt gedeelde bestanden zien.
* Er zijn statistieken over het systeem voor de beheerder en zijn in 1 oogopslag te zien.

### Technische eisen

* Er is een database connectie.
* De webapplicatie is veilig.

### Randvoorwaarden

* Er zijn voldoende demo users & data voor de statistieken.
* De klant voorziet zijn eigen web & database hosting.

## **Projectplan**

Het doel is om een systeem te bouwen waar gebruikers bestanden kunnen uploaden, downloaden en weer verwijderen. Gebruikers moeten bestanden kunnen delen en de beheerder moet statistieken kunnen zien. De gebruikers kunnen niet inloggen in het beheerders systeem en zullen dus een aparte rol aangewezen krijgen.

Het project is onderverdeeld in 3 sprints verdeeld over ongeveer 3 weken (Ik hoop het in minder tijd af te krijgen) met aan het einde van iedere sprint een feedback ronde.

### Projectteam

Ik werk alleen aan dit project en zit in een scrumgroep om eventuele problemen mee te kunnen overleggen en feedback te geven.

### Technieken

De volgende technieken ga ik gebruiken bij dit project:

* HTML / CSS, Sass (bootstrap als CSS framework)
* Javascript / JQuery
* PHP
* MySQL (database)
* Git (version control)

Ik zal moeten uitzoeken hoe er het beste bestanden kunnen worden opgeslagen en eventueel hoe je met de sites directory structuur werkt rondom php om zo bestanden op te halen of groottes te berekenen.

### Planning

Het is de bedoeling om aan het einde van iedere sprint een feedback ronde te houden en de feedback daarna te verwerken.

#### Week/Sprint 1: (23-27 mar)

In de eerste sprint ga ik de structuur voor het opslaan en verwijderen van data opzetten en de voorkant met de login voor de gebruiker maken.

#### Week/Sprint 2: (30-03 mar-apr)

In deze sprint ga ik feedback van de vorige sprint verwerken en het delen van bestanden tussen users maken. Ook ga ik met de klant overleggen wat er voor statistieken komen in het beheergedeelte.

#### Week/Sprint 3: (6-10 apr)

In sprint 3 ga ik het beheer gedeelte bouwen. Ook gaat de site getest worden d.m.v. een testplan.

### Op te leveren producten

* Documentatie: PvE, Projectplan, functioneel & technisch ontwerp.
* Een webomgeving voor de gebruiker om zijn bestanden te beheren.
* Backend voor de beheerder een statistieken overzicht.
* Database model met structuur.
* Demo data van ten minste 100 users, 500 files en 200 shares.
* Testplan met resultaten.

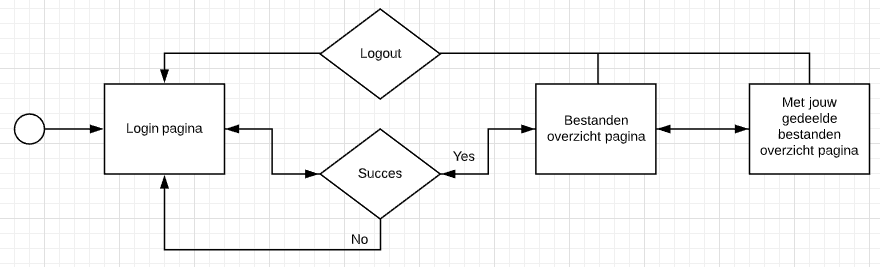
# **Functioneel ontwerp**

### Product backlog:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Wie?** | **Wat?** | **Waarom?** | **Prioriteit** |
|  | De gebruiker |  |  |  |
| 1 |  | Kan een account aanmaken | Om gebruik van de applicatie te kunnen maken | Should |
| 2 |  | Kan inloggen in het systeem | Om alleen zo toegang tot de bestanden te krijgen | Should |
| 3 |  | Kan een bestand uploaden | Om zo data te kunnen opslaan | Must |
| 4 |  | Kan een bestand downloaden | Om een bestand via een ander apparaat weer op te kunnen halen | Must |
| 5 |  | Kan een bestand verwijderen | Om het bestand niet meer op de server te hebben staan. | Must |
| 6 |  | Kan een bestand delen met een andere user | Om zo toegang tot een bestand te geven | Could |
| 7 |  | Kan een verbinding van een gedeeld bestand verbreken | Om zo een user geen toegang tot een bestand meer te geven | Could |
| 8 |  | Kan de bestanden zien die met hem gedeeld zijn. | Om een overzicht van alle gedeelde bestanden te krijgen | Should |
| 9 |  | Kan een verbinding van een bestand dat met hem is gedeeld verwijderen | Om zelf de verbinding van een gedeeld bestand te kunnen verbreken | Should |
|  | De beheerder |  |  |  |
| 10 |  | Kan inloggen op het beheersysteem | Om zo alleen als beheerder in de backend te komen | Should |
| 11 |  | Kan in 1 oogopslag de statistieken zien | Om gemakkelijk zijn systeem in de gaten te houden | Must |

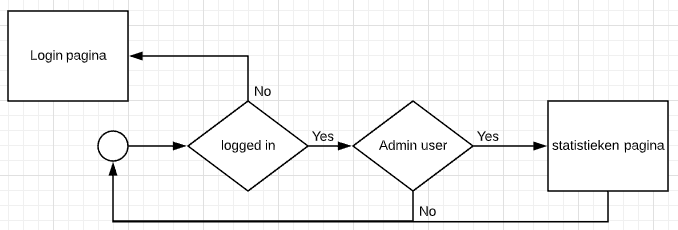
### Activity diagram

#### Flow voor de bezoekers:



#### Flow voor de beheerder:

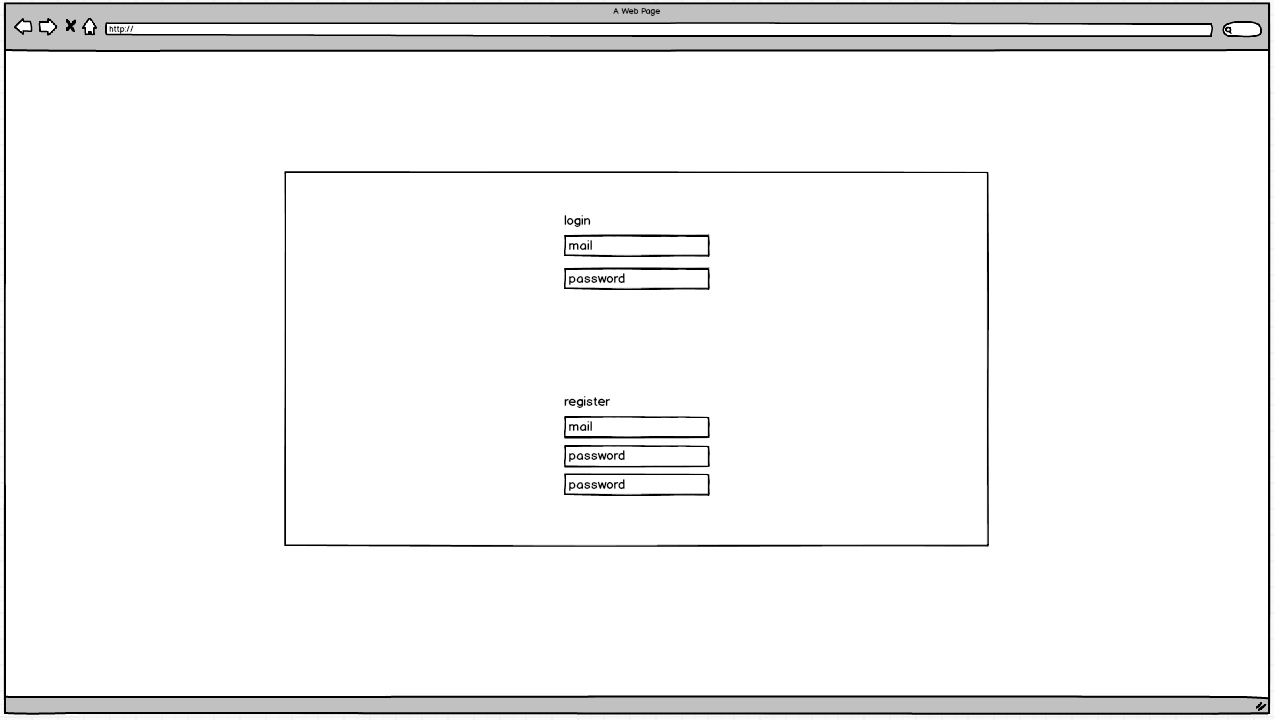
De gebruiker begint bij de front-end en heeft daar een knop in de navigatie staan om bij het admin paneel te komen. Alleen gebruikers met een admin status kunnen hier komen.



### Wireframes:

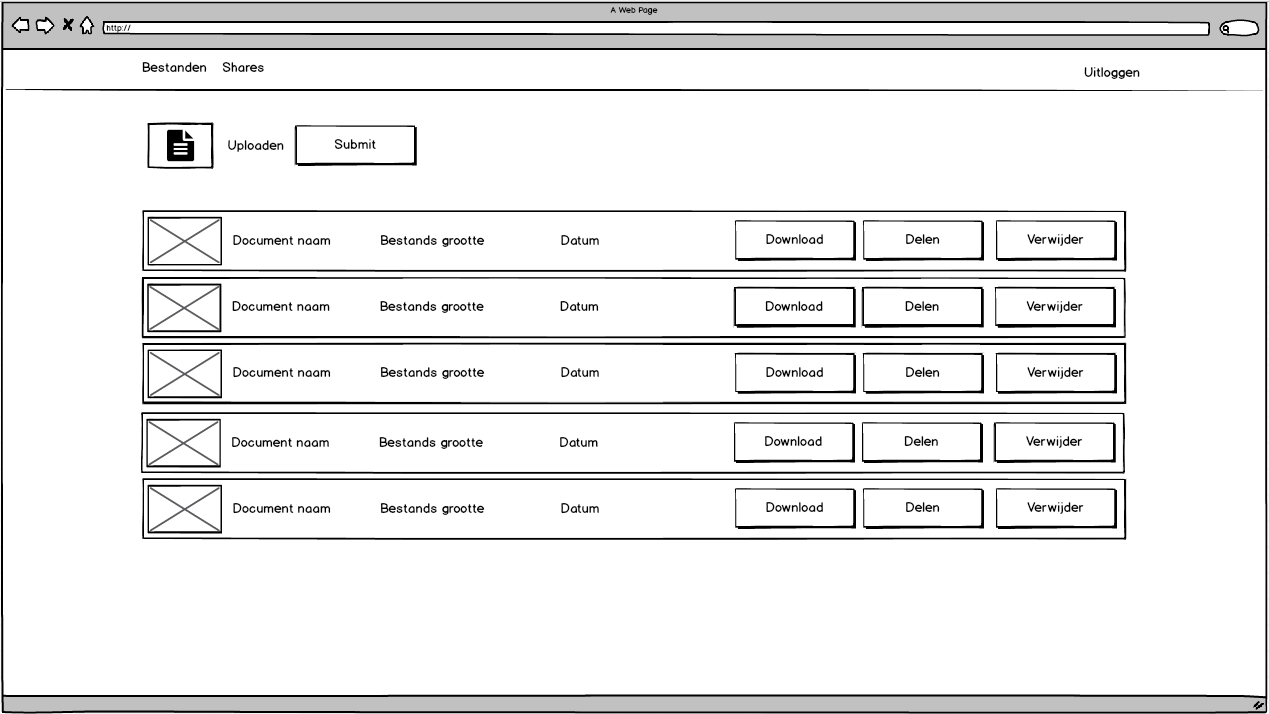
#### Login/home pagina: Userstory 1, 2

Users moeten via deze pagina inloggen om toegang te krijgen tot de rest van de website. Hier kan een user ook regristeren

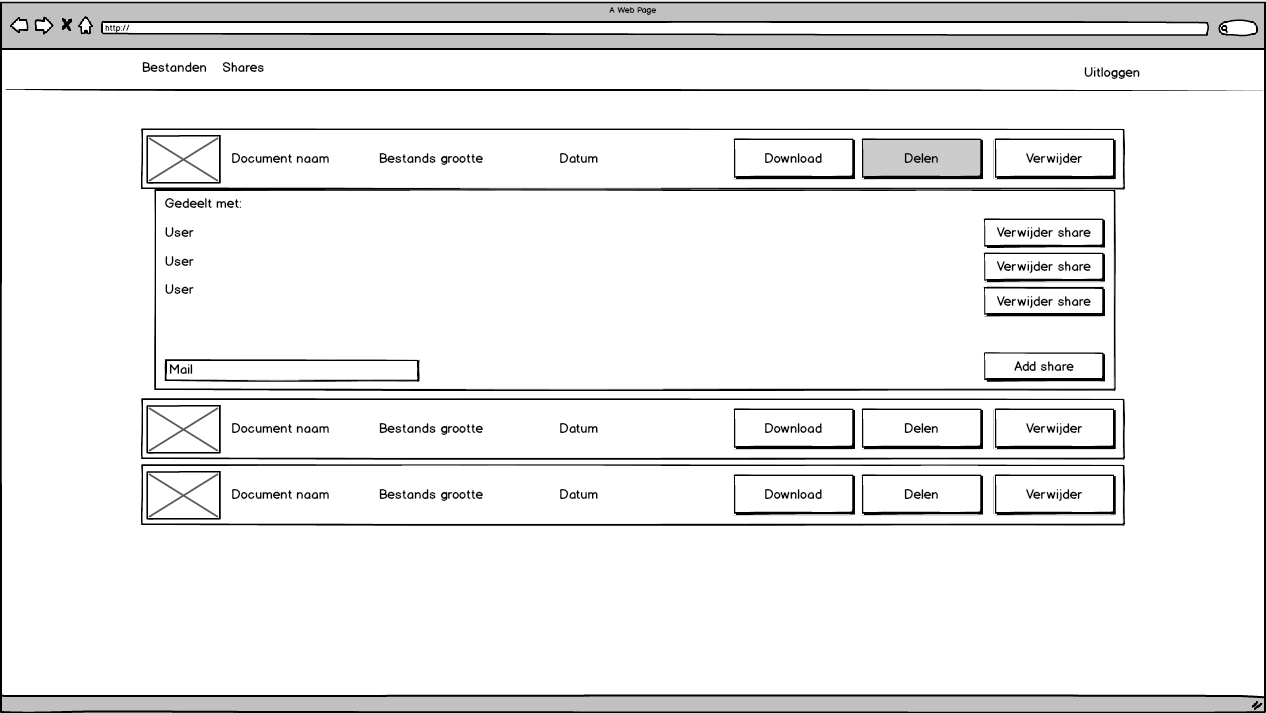


#### Bestanden overzicht: Userstory 3, 4, 5, 6, 7

Op deze pagina kun je al je bestanden zien en nieuwe bestanden uploaden. Ook kun je een bestand verwijderen en/of downloaden naar je pc.

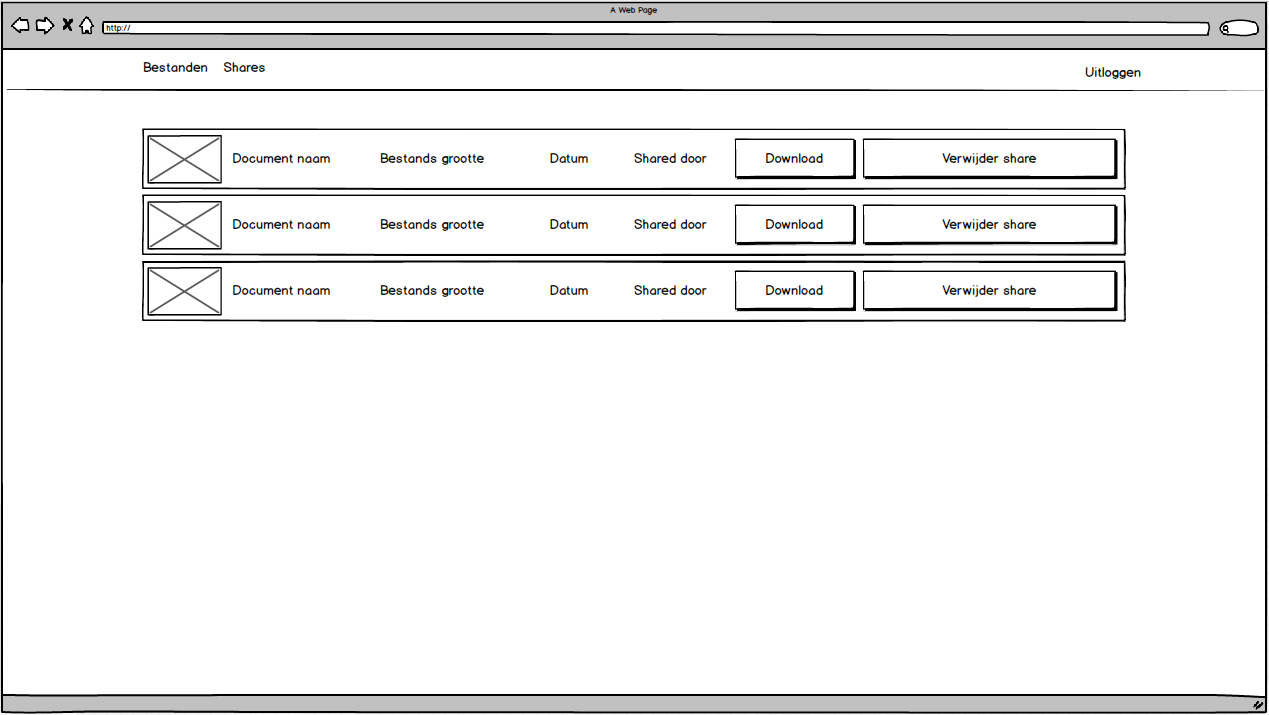


Als je op de knop voor delen klikt opent er een scherm met de informatie over met wie het bestand is gedeeld. Ook kun je de toegang hier weer verwijderen. Onderaan staat een veld om een nieuwe share toe te voegen.



#### Gedeelde bestanden: Userstory 8, 9

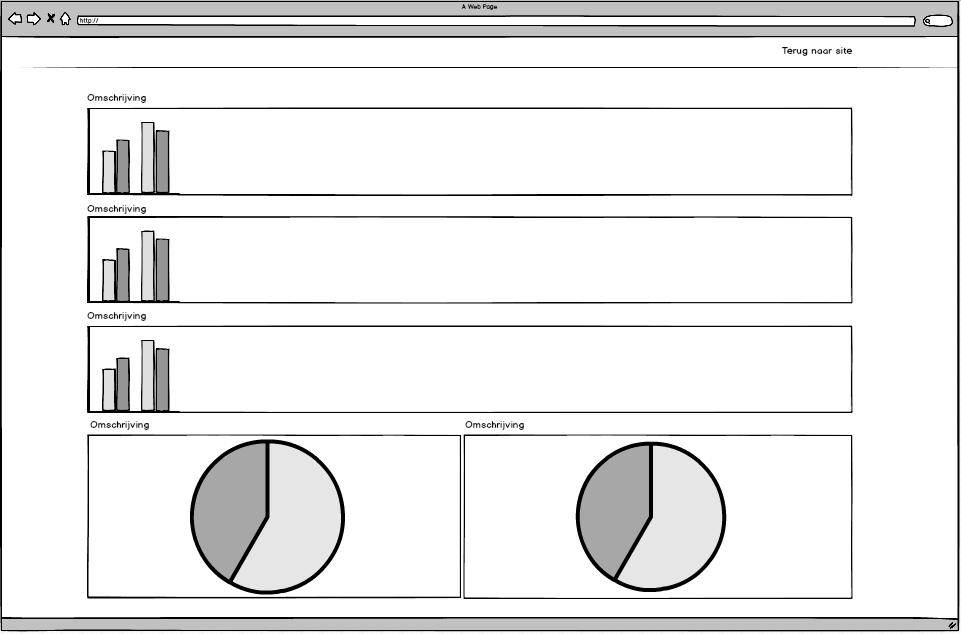
Hier kun je de bestanden zien die met jou gedeeld zijn en kun je een link van een share verwijderen.



#### Beheer/statistieken: Userstory 10, 11

Op het beheerder paneel kan de beheerder de volgende statistieken bekijken:

* Aantal gedeelde bestanden per maand (staaf)
* Aantal bestanden & Totale data grootte van alle bestanden per week (staaf) selectbox
* Aantal gebruikers & Totaal aantal bestanden per maand (lijn)
* Aantal bestanden van de 5 meest populaire bestands extensies (cirkel)
* Aantal geüploade/opgeslagen bestanden per week (staaf)



# **Technisch ontwerp**

### Libraries

Ik ga gebruik maken van de libraries: Bootstrap en JQuery. Ook maak ik gebruik van een Sass compiler.

### Folder structuur

|  |  |  |
| --- | --- | --- |
| admin |  |  |
|  | css |  |
|  | js |  |
|  | design |  |
|  | pages |  |
|  | php |  |
|  |  |  |
|  |  | ajax |
|  |  | templates |
| css |  |  |
| design |  |  |
| js |  |  |
| uploads |  |  |
|  | \*username\* |  |
| pages |  |  |
| php | ajax |  |
|  | templates |  |

### Bestand benamingen

Afbeeldingen worden door een beheerder via de sites backend geupload. Van deze bestanden maken de namen niet uit zolang er maar geen bestanden met een dubbele naam voorkomen.

Afbeelding die in de site zijn verwerkt (in de folder ‘design’) hebben altijd kleine letters en underscores.

Alle andere bestandsnamen zijn in principe met kleine letters en hebben eventueel underscores als witruimte tussen 2 woorden. Php bestanden voor een Ajax call staan in de subfolder ‘ajax’.

\*username\* Wordt gegenereerd voor iedere user en hebben hier hun eigen bestanden.

Voor CSS wordt er gewerkt in de SCSS-bestanden die gecompiled worden naar de CSS-bestanden.

### ERD

