Contents

[1.Omschrijving 2](#_Toc421790054)

[1.1Concept-doelgroep 2](#_Toc421790055)

[1.2 Functionaliteit 2](#_Toc421790056)

[1.3Screenshots 3](#_Toc421790057)

[1.4URL 3](#_Toc421790058)

[2.Planning 3](#_Toc421790059)

[3.Usability test report(s) 5](#_Toc421790060)

[4.Technische documentatie 5](#_Toc421790061)

[4.1Wireframes App 5](#_Toc421790062)

[4.2 Wireframes Website 10](#_Toc421790063)

[**Oorspronkelijke Balsamiq wireframes** 10](#_Toc421790064)

[4.3Sitemap 13](#_Toc421790065)

[4.4Architectuur 15](#_Toc421790066)

[4.5 Database Schema 16](#_Toc421790067)

[4.6 Code 16](#_Toc421790068)

[5.Conclusie 24](#_Toc421790069)

[6.Resources 24](#_Toc421790070)

# 1.Omschrijving

## 1.1Concept-doelgroep

Kotter is een app waarmee studenten maar ook ouders van studenten op een leuke manier het ideale kot voor zichzelf of voor hun zoon/dochter kunnen vinden. Bij kotter ben je zeer snel ‘up and running’ zo moet je maar enkel je inlog gegevens ingeven , school en prijs. Hierna kan men al direct door onze databank lopen van koten. Dit is zeer eenvoudig, en de gebruikers kunnen elk kot liken of disliken. Dit kan men doen door de icoontjes te gebruiken, maar ook door naar links of naar rechts te swipen. De gebruikers kunnen ook specifieker zoeken door de ingebouwde filters. Zo kan men de minimum opervlakte bepalen, de maximum afstand tot hun campus, maximum prijs,... De gebruikers kunnen ook vie de app eenvoudig de kot eigenaars contacteren zowel via mail als telefonisch.

Bij kotter is er ook een website die als promo dient. Maar hier kunnen geintreseerde kot eigenaars ook contact met ons opnemen wanneer ze hun koten ook in onze databank van koten willen steken. Wanneer we dit accepteren sturen wij hun inlog gegevens door voor het koten beheer paneel. Hierop kunnen ze dan hun eigen koten gaan toevoegen. Deze worden dan door het kotter team gereviewed en wanneer deze goedgekeurd worden zullen deze dus in de app beschikbaar gemaakt worden. Wanneer deze afgekeurd worden dus niet. Als een kot al een tijdje in de app zit, kan men ook het aantal likes terugvinden in het koten beheer paneel.

**Commerciël aspect (theoretisch)**

We willen de gebruikers niet hinderen om op een zo gemakkelijk mogelijke manier contact te zoeken met de koteigenaar/verhuurder. Daarom gaat deze contactinformatie direct beschikbaar zijn.

We gaan met één kotbedrijf starten, daarna zullen andere kotbedrijven contact kunnen aanvragen via de website. (kotterapp.be) Ze nemen contact op via email, dan sturen wij daarna onze prijzen door.

Een kotbedrijf dat bij ons een aanvraag heeft gedaan gaan we evalueren, en accepteren/weigeren.  
Als we een bedrijf accepteren en de financiële kant is achter de rug, dan sturen we een login en paswoord door.  
Dit wilt nog niet zeggen dat koten automatisch in de app terecht komen, zie: kwaliteit

Kwaliteit

Die kwaliteit van koten zullen we garanderen door manueel koten te *approven*, hierdoor sluipen er geen koten in die niet op hetzelfde niveau zitten. (de huisjesmelkers eruit filteren)

## 1.2 Functionaliteit

**User flow**

De app is ontwikkeld voor tablets en smartphones, je kan hem dus downloaden via de Apple App Store en de Google Play Store. Via de website vindt men ook links naar beide stores.

Registratie is eenvoudig en de enige gegevens die we vragen zijn een gebruikersnaam en paswoord.  
Na de registratie geeft een gebruiker zijn/haar toekomstige school op en een maximum budget.

That’s all! De gebruiker krijgt nu koten te zien met het zelfgekozen budget en in de buurt van de gekozen school/universiteit.

Door koten browsen is super eenvoudig. Bij elk kot moet de gebruiker ‘liken’ of ‘disliken’. Een kot disliken wilt zeggen dat het verwijderd wordt uit de pool waaruit de gebruiker kan kiezen. Een geliked kot verschijnt in de favorieten lijst.  
Zo eindigt de gebruiker uiteindelijk met een lijst van koten die hij/zij geweldig vindt + meer koten die de gebruiker nog nooit eerder heeft gezien indien hij nog meer opties wil.

Via de instellingen kan de gebruiker eenvoudig het budget veranderen, de school/universiteit aanpassen en een massa andere instellingen: badkamer ja/nee, keuken ja/nee, minimum oppervlakte,...

## 1.3Screenshots

## 1.4URL

# 2.Planning

Voorafgaand:

* Brainstormen
* Feedback momenten

Week 1: 9-15 maart

* Database shema
* Laravel project aanmaken + login(Maico)
* Laravel koten toevoegen aan db(Maico)

Week 2: 16-22 maart

* Experimenteren met Ionic Framework(Maico)
* Laravel koten overzicht + aanpassen + deleten(Maico)

Week 3: 23-29 maart

* IONIC: Tutorial pagina & registratie pagina(Maico)
* Wireframes app: 1e versie(Matthias)

Week 4: 30 maart – 5 april

* Rest wireframes app(Matthias)
* Brainstormen naam(Matthias + Maico)
* Facebook register (Maico)
* Info kot ophalen via AJAX (Maico)

Week 5: 6-12 april

* Images Kot uitlezen (Maico)
* Basic filters van user (Maico)
* Wireframes nieuwe versie (Matthias)

Week 6: 13-19 april

* Wireframes website (Matthias)

Week 7: 20-26 april

* Verschillende kleurencombinaties website(Matthias)
* Afstand berekening van school tot kot (Maico)

Week 8: 27 april – 3 mei

* Foto’s koten
* App images fetchen
* Google maps toevoegen app (Maico)
* Filters toepassen bij uitlezen kotten (Maico)

Week 9: 4 mei – 10 mei

* Tutorial stijl + tutorial zelf (Matthias)
* Update: location.html, main\_kot.html, settings.html (Matthias)
* Filter bug fixes (Maico)
* Favorite pagina + afstand berekening van kotten server-side (Maico)
* Internet connection check (Maico)

Week 10: 11-17 mei

* Server aankopen + laravel project online zetten(Maico)
* Extra filters toegevoegd (Maico)
* Google maps bug fix + favorite structuur aanpassen (Maico)
* Campusen antwerpen opzoeken en in db plaatsen (Maico)
* Performance app verbeteren (Maico)

Week 11: 18-24 mei

* API in laravel maken voor app (omzetten van basic php files in laravel) (Maico)
* Tinder styled swipe cards toegevoegd(Maico)
* Loading spinne toevoegen(Maico)
* Fb login via andere plugin (Maico)

Week 12: 25-31 mei

* Timelapses shooten (Matthias)
* Database shema updaten (Maico)
* Tinder swipe card fix (Maico)
* Login + logout toegevoegd app(Maico)
* Reset favorite koten(Maico)

Week 13: 1-7 juni

* Usability testing app
* Timelapses shooten
* Bug fixes (Maico)
* Contact pagina toevoegen website (Maico)
* Admin paneel kotten in laravel(Maico)

Week 14: 8-14 juni

* Feedback promotoren
* Afwerken promofilm-> SELF-DEADLINE: 12 juni
* Feedback aanpassingen doorvoeren(Maico+Matthias)
* Code nakijken + opruime (Maico)
* Aan bundel werken (Maico+Matthias)

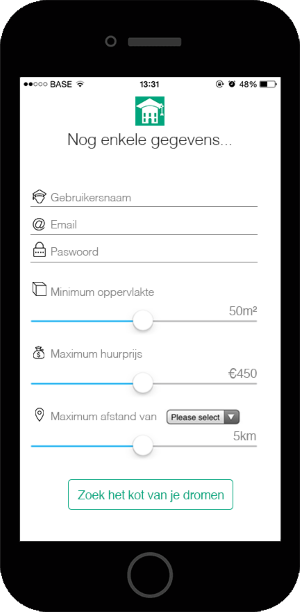
Week 15: 15, 16, 17 juni

* Voorbereiding presentative (Maico+Matthias)
* Final touches bundels (Maico+Matthias)
* Bundels laten bundelen (Maico+Matthias)

# 3.Usability test report(s)

# 4.Technische documentatie

## 4.1Wireframes App

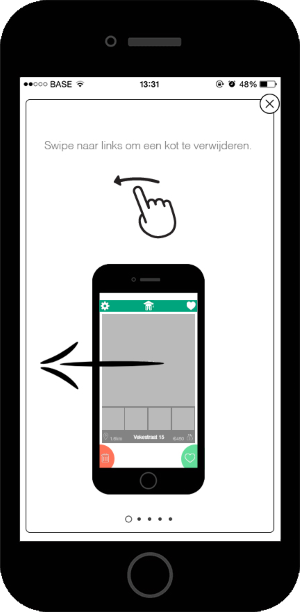
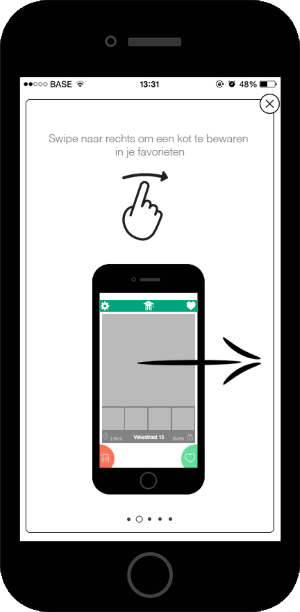


Beginscherm:

* Slogan
* Inlog/registreer opties (+facebook)
* Start tutorial

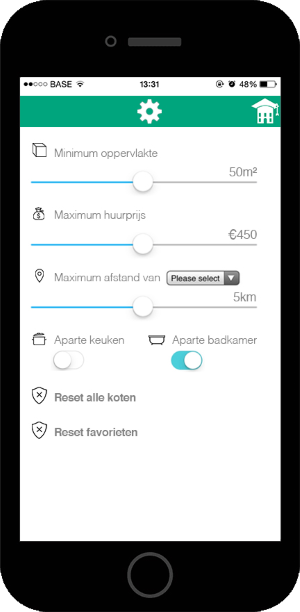
Registratie gegevens:

* Gebruikersnaam, email, paswoord
* Minimale oppervlakte
* Maximum huurprijs
* Maximale afstand van …



Tutorials:

* Basisuitleg applicatie
* Swipe gestures



Instellingen:

* Extra opties zijn hier te zien: Meer opties dan bij initiële registratie
* Mogelijkheid om favorieten te resetten, en alle koten te wissen

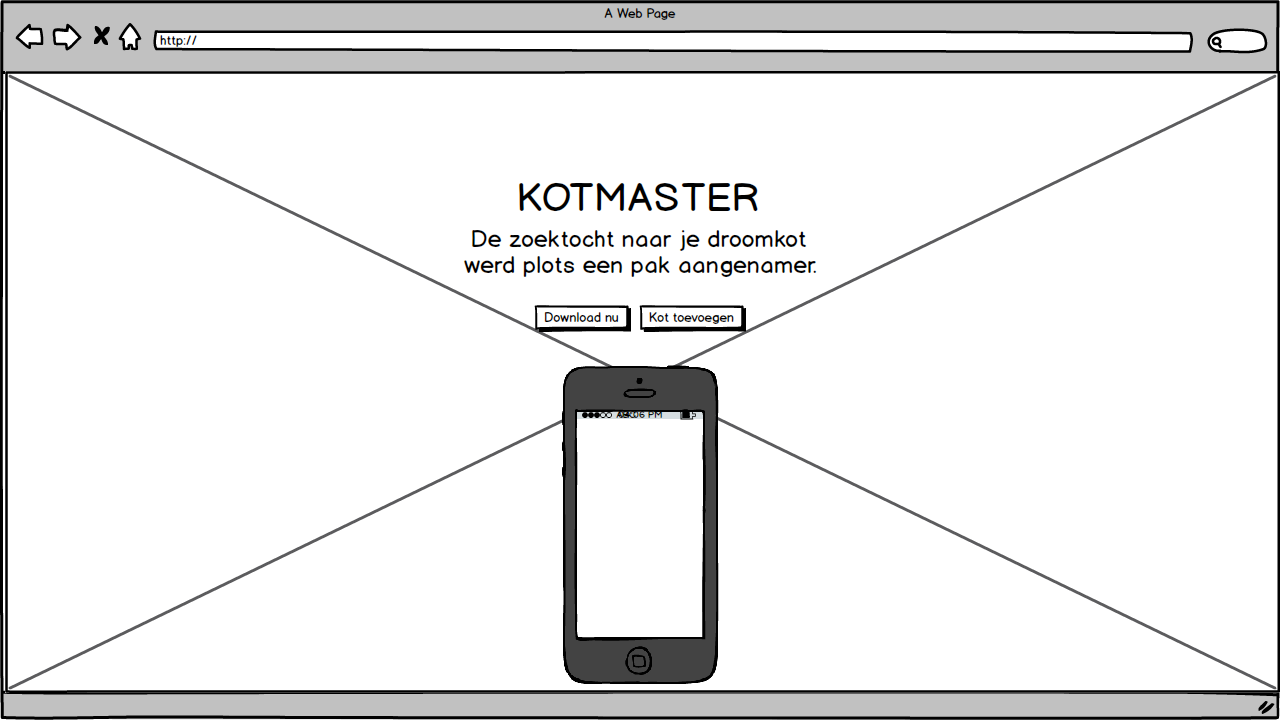


Favorieten:

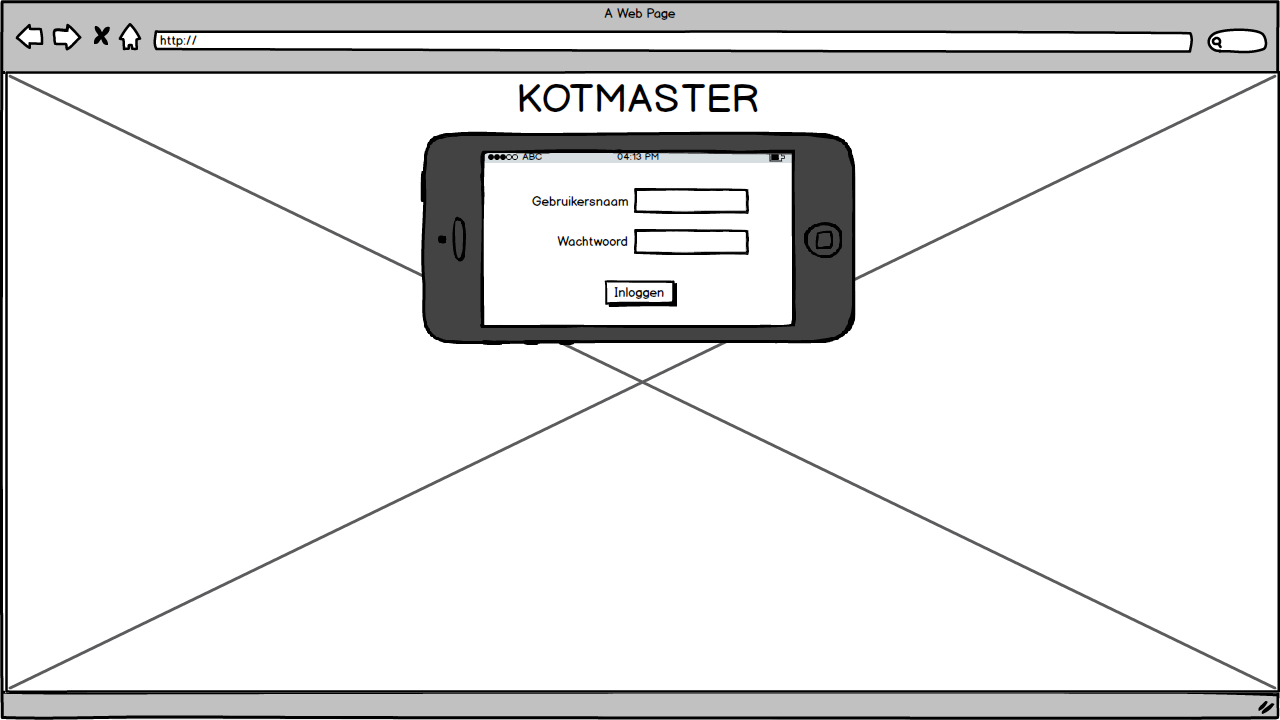
* Beheer favorieten
* Verschillende weergave opties
* Mogelijkheid om te her-selecteren tussen favorieten

## 4.2 Wireframes Website

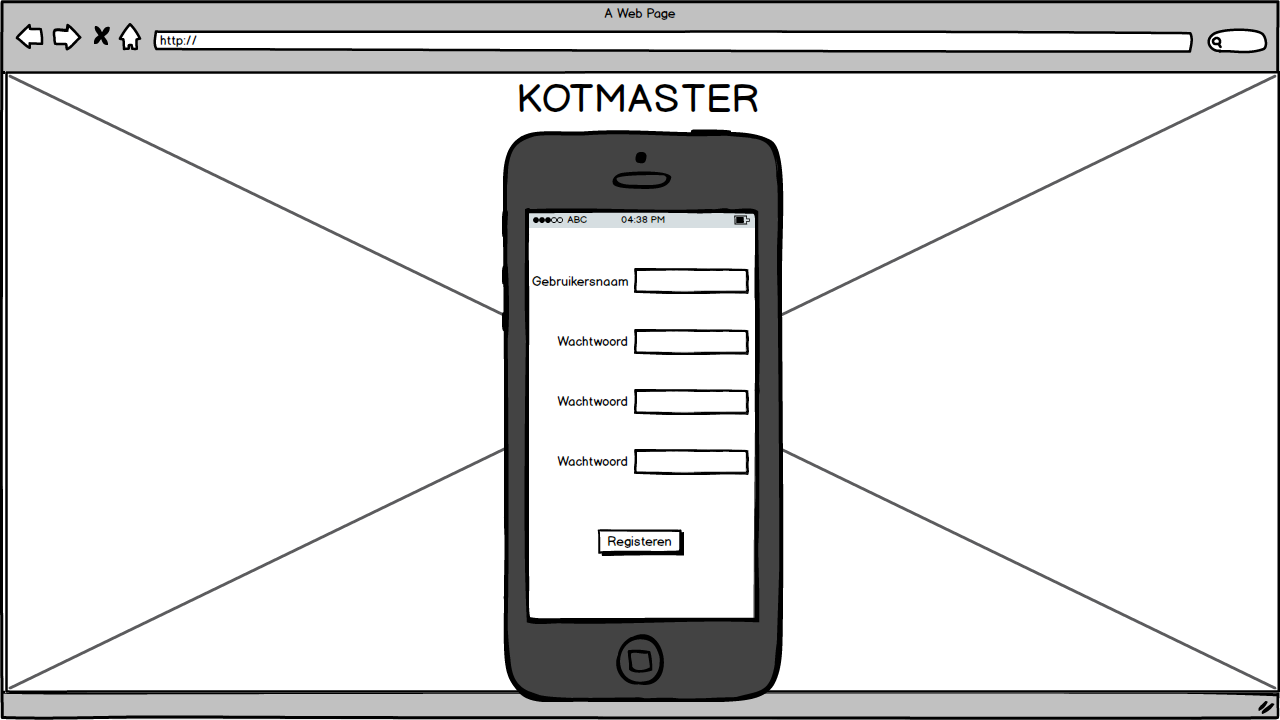
**Oorspronkelijke Balsamiq wireframes**



Landing-page: Naam + Slogan + Download button + Login button



Login scherm in iPhone mockup



Registratie scherm in verticale iPhone mockup

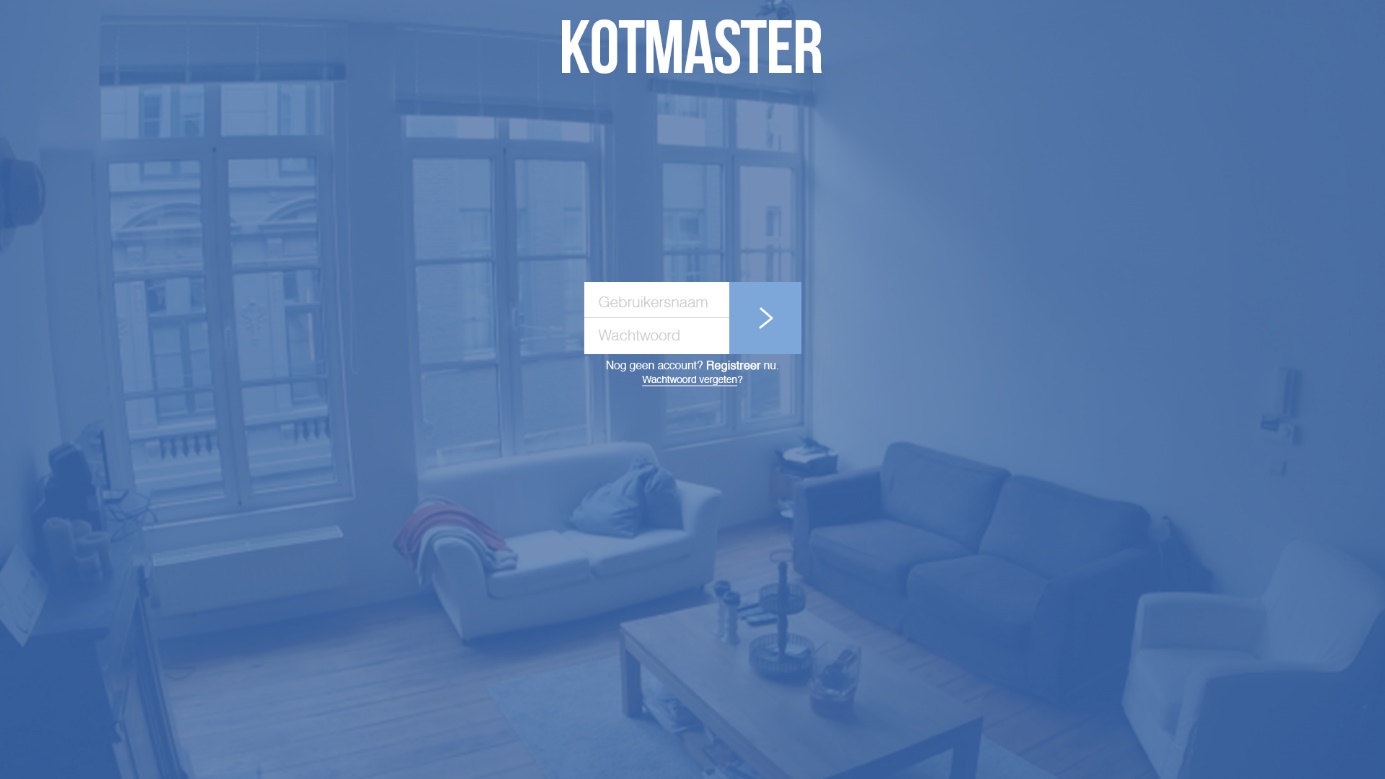
**Photoshop wireframes**



Landing-page: Naam + Slogan + Downloadknop + iPhone mockup  
Achergrondafbeelding = luxueus kot



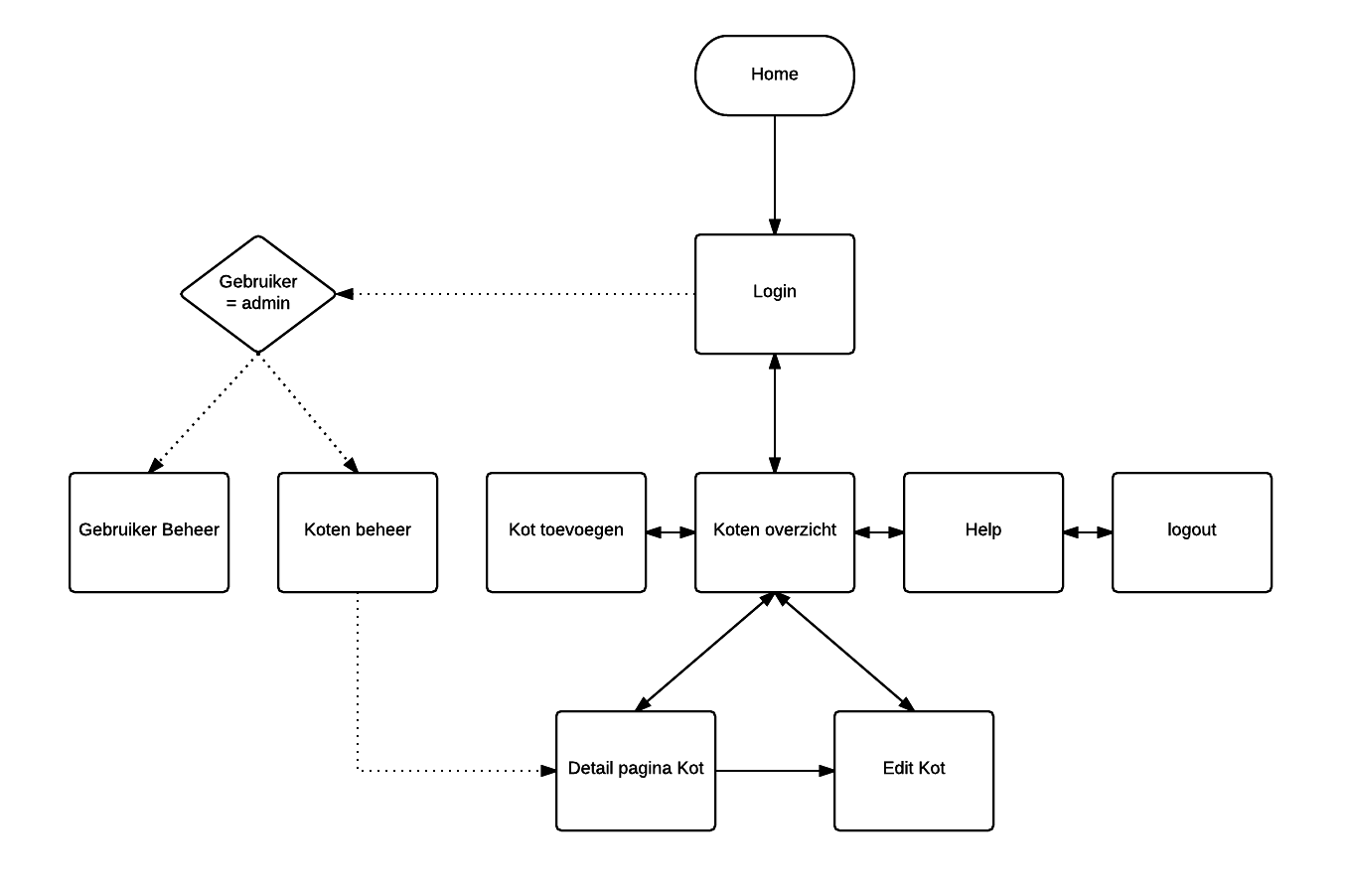
Registratiescherm op zelfde scherm als landing-page  
Eenvoudig clean formulier



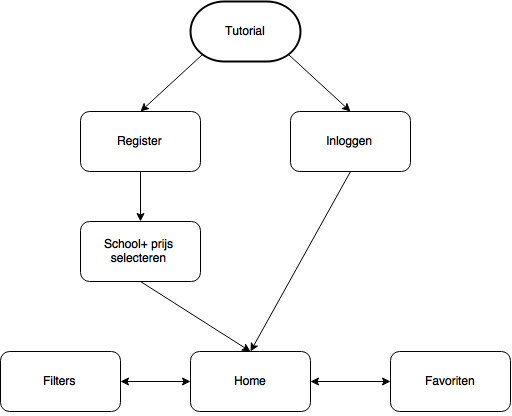
Loginscherm op zelfde pagina als landing-page  
Eenvoudig clean formulier

## 4.3Sitemap

**Website**

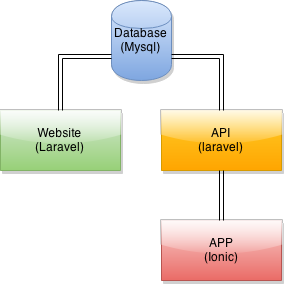


**App**

****

## 4.4Architectuur

Ons project bestaat dus uit 2 delen. Een website en een app. De website gemaakt in laravel. De applicatie is gemaakt in ionic. De app doet telkens ajax calls naar de website namelijk naar de apicontroller deze controller verzorgd alle request van de app.



**Laravel**

Laravel is een backend MVC framework of Model-View-Controller framework. Laravel maakt het development procces veel eenvoudiger en leuk door gebruik te maken van enkelen geweldige feutures zo als: Eloquent ORM, Flexible routing, Migrations, Modular packaging system,...

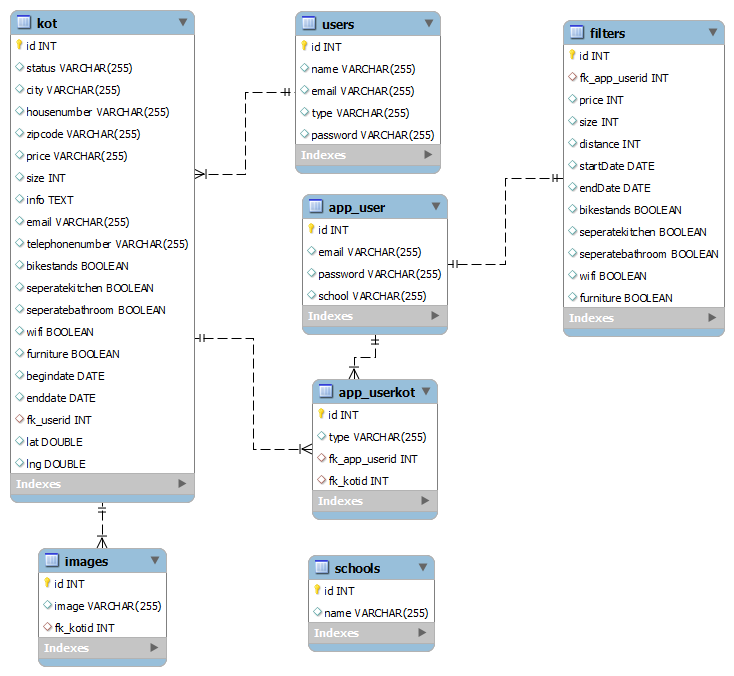
**Ionic**

Ionic is Gratis en open source , Ionic biedt een bibliotheek van mobiele - geoptimaliseerde HTML , CSS en JS componenten , gestures , en tools voor het bouwen van zeer interactieve apps. Gebouwd met Sass en geoptimaliseerd voor angularjs .

Ionic maakt gebruik van angularjs , cordova en ionic framework zelf. Angularjs is een MVW framework of Model-View-Whatever framework. Deze wordt dus gebruikt om heel de logica te developen bij een ionic app. Dan hebben we nog cordova dit word gebruikt om de native functionaliteiten van het aparaat te gebruiken zo als de camera, accelerometer,geheugen ...

Het ionic framework zelf is zo als eerder gezecht ene bibliotheek van css, js enhtml componenten. Vergelijkbaar met bv: bootstrap of een foundation.

## 4.5 Database Schema



## 4.6 Code

**Kotten ophalen uit database**

Deze code is van de app. Dit is een functie die ik oproep bij het openen van de view en telkens opnieuw wanneer er gestemd werd. Hier word er gebruik gemaakt van KotService.getKot deze zal een get request doen naar de server. Wanneer de data terug komt zal er een marker op de map worden gezet en wordt de data uitgelezen.



Hier kan je de KotService zien. Bij elke functie word er een get request naar de server gestuurd met bepaalde paramaters.



Bij de getKot request word deze functie aangeroepen in de apicontroller op de server. Hier wordt de functie getkot van het kot model opgeroepen. Na dat deze functie is uitgevoerd zal men het kot als data terug sturen en wordt deze uitgelezen in de app.



Hier kan je de functie zien die het ophalen van het kot behandeld. Er wordt dus een qeury opgemaakt aan de hand van de gebruiker zijn filters en gestemde koten om het correcte kot te vinden voor hem. Wanneer dit klaar is gaan we nog de afstand van het kot tot de gebruiker zijn campus berekenen aan de hand geotool. Wanneer deze kleiner is dan de gebruiker zijn maximum afstand wordt deze gereturned. Wanneer dit niet het geval is zal deze functie opnieuw worden uitgevoerd tot dat de afstand kleiner is dan de gebruiker zijn maximum afstand tot zijn campus.





**Kotten opslaan in database**

Wanneer beheerders van koten het formulier verzenden om een nieuw kot toe te voegen zal deze naar volgende functie worden gestuurd. Deze het kot zal opslaan en de afbeeldingen in de database.



Voor dat deze code wordt uitgevoerd moet er eerst een validatie gebeuren. Deze gebeurt door addKotRequest. Hieronder ziet u hoe deze eruit ziet. Deze zal er voor zorgen dat er telkens 4 afbeeldingen worden geupload per kot en dat alle nodige velden correct worden ingevu

**Stemmen via swipen**

Voor de swipe cards heb ik gebruikt gemaakt van een plugin ionic-contrib-tinder-cards die het gedeelte van het animeren van de swipe cards al voor je doet. Je moet alleen zelf nog bepaalde acties vast hangen aan het swipen zelf. Dit is dankzij ionic gestures vrij simpel. Je kan hiervoor custom attributes gebruiken zoals: on-swipe-left , on-swipe-right , on-release,...



Deze zullen dan een functie oproepen. Buiten het links en rechts swipen is er ook nog een functie onRelease. Waneer je niet snel genoeg swiped zal deze anders als een drag gezien worden. Wanneer de drag stopt zal de onRelease functie opgeroepen worden. Hier zal gecontroleerd worden of je naar links of naar rechts geswiped hebt. Aan de hand hiervan wordt er dan gestemd. De timeout zorgt er voor dat er zeker niet 2 keer gestemd wordt, dus voorkomt dat de onRelease functie samen met een andere functie, zoals bv de swipeLeft functie, samen zullen worden opgeroepen. 

# 5.Conclusie

# 6.Resources

<http://ionicframework.com/>

<http://laravel.com/docs/5>

<http://blog.ionic.io/tinder-for-x/>

<https://blog.nraboy.com/2014/10/implement-google-maps-using-ionicframework/>

<https://blog.nraboy.com/2014/06/check-network-connection-with-ionicframework/>

<http://ngcordova.com/docs/plugins/facebook/>

<http://ccoenraets.github.io/ionic-tutorial/create-ionic-application.html>

<https://daneden.github.io/animate.css/>

<https://github.com/toin0u/Geotools-laravel>

<http://image.intervention.io/>

<http://stackoverflow.com/>

<http://forum.ionicframework.com/>

<http://specials.han.nl/themasites/studiecentra/verwerken-en-delen/bronnen-vermelden/apa-normen/#comp00004b902de60000000b27453d>