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.

Wat Kotter uniek maakt is het concept van de totale prijs/geen extra kosten. Oftewel heeft een kot een vaste totale prijs (dit wordt aangegeven in de app) oftewel een geschatte vaste prijs.  
Zo weet men exact hoeveel een kot gaat kosten, dit voorkomt ook dat een kotbaas een kot kan adverteren voor €450, en dat er dan nog eens €150 kosten bijkomen.  
De totale/geschatte totale prijs werd in de app geimplementeerd nadat studenten die op kot zitten en hun ouders de ‘extra kosten’ als één van de onaangenaamste dingen noemden in hun kot-zoek-ervaring.

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)

Totale prijs/Geschatte totale prijs

Elk kot heeft oftewel een vaste prijs, oftewel een geschatte vaste prijs. Bij een vaste totale prijs is er geen discussie mogelijk. Bij een geschatte vaste prijs wordt de prijs bepaalt door het gebruik van de gemiddelde student, en zo kan men tot een geschatte totale prijs komen zodat studenten en hun ouders weten hoeveel het kot hun gaat kosten, zonder voor verassingen te komen te staan.

## 1.2 Functionaliteit

**User flow app**

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 email adres 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.Hier kan hij telkens ook de extra info bekijken door op de info knop te duwen. Hier krijgt hij dan de lokatie te zien op een mapje, een beschrijving van het kot al de feuteres van het kot en de contact gegevens van de beheerder.

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.Wanneer de gebruiker een bepaald kot wilt gaan bezichtigen of nog wat meer info wilt kan hij zeer eenvoudig de kot beheerder contacteren door het extra info venster te openen en op het email adres te duwen of telefoon nummer. Dit zorcht er voor dat er een nieuwe mail word opgemaakt met al het juiste email adres ingevuld en het onderwerp van de email of waneer men op de telefoon nummer duuwt zal de telefoon dailer openspringen met de nummer van de kot beheerder.

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,...

**User flow website**

Een kot beheerder komt op onze website dit kan puur toevallig zijn of omdat hij al eens iets gehoord heeft over kotter. De kot beheerder kijkt op onze website installeerd de app en heeft intressen om zijn koten ook aan onze database toe te voegen. De kot beheerder kan ons dan contacteren via onze website. Hij kan ons dan laten weten dat hij intressen heeft om zijn koten toe te voegen.

Dan kunnen wij van het kotter team voor deze kot beheerder een acount aanmaken. De kot beheerder zal dan een email krijgen met zijn inlog gegevens en nu kan hij aan de slag met zijn kotten toe te voegen.

Bij het toevoegen van kotten word er vanalle data gevraagd zo als: adres, contact gegevens, prijs, groten en telkens 4 afbeeldingen per kot. De prijs moet ook steeds de totaal prijs zijn dis d.w.z. dat de verwarmings kosten, energie kosten en kosten van het water er telkens zitten mee ingerekend dus deze prijs is steeds wat de gebruiker ongeveer zal betalen op het einde van de maand.

Wanneer deze persoon zijn kotten heeft toegevoegd zullen zij eerst worden gereviewd worden door het kotter team vooralleer deze in de app zullen verschijnen. Het kotter team zal dan bepalen welke kotten worden geacepteerd en welke gewijgerd. Wanneer deze review klaar is en zijn kotten zijn geacepteerd zullen deze verschijnen in de kotter app.

De kot beheerder dan na een tijde altijd eens komen terug kijken hoe zijn kotten het doen want telkens bij elke kot staat er ook bij hoeveel likes het kot al heeft gekregen van de gebruikers van de app.

Wanneer de beheerder het kot niet meer wil verhuren kan hij het snel uit onze app halen door het gewoon te deleten uit onze database.

## 1.3Screenshots

## 1.4URL

# 2.Planning

Voorafgaand:

* Brainstormen
* Feedback momenten

Week 1: 9-15 maart

* Database shema (Maico)
* Laravel project aanmaken + login (Maico)
* Laravel koten toevoegen aan db (Maico)
* Contact zoeken kotbedrijf voor fotos (Matthias)

Week 2: 16-22 maart

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

Week 3: 23-29 maart

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

Week 4: 30 maart – 5 april

* Wireframes + design 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 + design nieuwe versie (Matthias)

Week 6: 13-19 april

* Wireframes +design 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 trekken + bewerken (Matthias)
* App images fetchen (Maico)
* Google maps toevoegen app (Maico)
* Filters toepassen bij uitlezen kotten (Maico)

Week 9: 4 mei – 10 mei

* Tutorial app (Matthias)
* App design updates (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)
* Front end app (Matthias)
* Nieuw design app (Matthias)

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 spinner toevoegen(Maico+Matthias)
* Fb login via andere plugin (Maico)
* Feedback website promotor (Maico+Matthias)
* Timelapses shooten (Matthias)
* Front-end fixes website (Matthias)
* Start ontwikkeling specifiek iOS (Matthias)

Week 12: 25-31 mei

* Database shema updaten (Maico)
* Tinder swipe card fix (Maico)
* Login + logout toegevoegd app(Maico)
* Reset favorite koten(Maico)
* Timelapses shooten (Matthias)
* Scènes shooten (Matthias)
* Help beheerderspaneel (Matthias)
* Front-end app (Matthias)
* Finale versie promofilm script (Matthias)
* Stand van zaken + to do (Matthias)
* Usability testing website (Matthias)

Week 13: 1-7 juni

* Usability testing app (Matthias)
* Laatste timelapses shooten (Matthias)
* Bug fixes (Maico)
* Contact pagina toevoegen website (Maico)
* Admin paneel koten in laravel(Maico)
* iOS bug fixes (Matthias)
* Admin paneel front-end (Matthias)
* Alle HTML valid maken (Matthias)

Week 14: 8-14 juni

* Feedback promotoren
* Laatste scenes filmen promofilm (Matthias)
* Monteren promofilm (Matthias)
* Feedback aanpassingen doorvoeren(Maico+Matthias)
* Code nakijken + opruimen (Maico+Matthias)
* Aan bundel werken (Maico+Matthias)
* Afspraak met kotbedrijf (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)

## App Website: 5 proefpersonen 16-55jaar

Rol student/ouder

Vraag: Is het duidelijk waarover de website gaat? Zo ja, waarover gaat de website?

Alle proefpersonen wisten onmiddelijk waarover de website ging toen ze de homepagina zagen.

Taak: App downloaden  
Scenario: Download de app voor jou smartphone

Dit verliep moeiteloos voor alle proefpersonen.

## Admin paneel

Rol koteigenaar/bedrijf

Taak: Account aanvragen  
Scenario: Je bent een koteigenaar en bent geïnteresseerd in onze app. Contacteer ons.

1 van de 5 proefpersonen vond niet onmiddelijk de contact optie, suggestie van proefpersoon: Achtergrond al direct wit opvullen voor meer visibiliteit.  
Suggestie van andere proefpersoon: Wat als een ouder/student een vraag heeft over de app.   
(enkel contactoptie voor koteigenaars)

Taak: Inloggen  
Scenario: Je account is aangemaakt en je hebt de registratiedetails via mail ontvangen. Log in op de website.

Dit verliep moeiteloos voor alle proefpersonen.

Taak: Kot aanmaken  
Scenario: Je bent ingelogd en wilt je eerste kot toevoegen. (doe dit met je eigen thuisadres)

Dit verliep moeiteloos voor alle proefpersonen.

Taak: Kot editen  
Scenario: Je eerste kot is toegevoegd maar er is toch een fout ingeslopen. Corrigeer één van de gegevens.

Dit verliep moeiteloos voor alle proefpersonen.  
Suggestie proefpersoon: Eerst straatnaam en nummer, dan pas gemeente en postcode.

Taak: Kot deleten  
Scenario: Je gaat je kot verkopen, het is dus niet meer te huur voor studenten. Verwijder het uit de app.

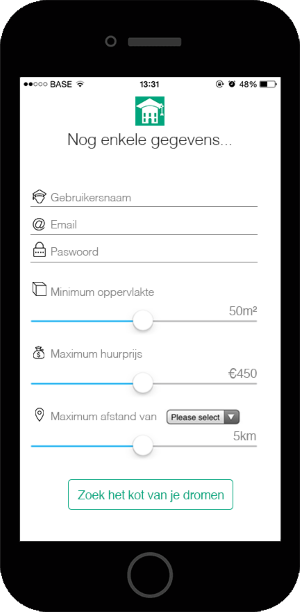
Dit verliep moeiteloos voor alle proefpersonen.

Taak: uitloggen.  
Scenario: Log uit.

2 proefpersonen moesten langer zoeken, en 1 vond het helemaal niet.  
-> Uitlog knop misschien beter in submenu

# 4.Technische documentatie

## 4.1Wireframes App

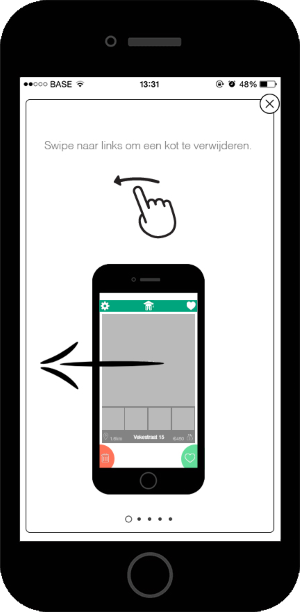
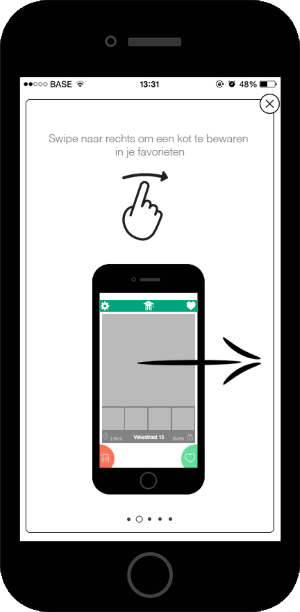


Beginscherm:

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

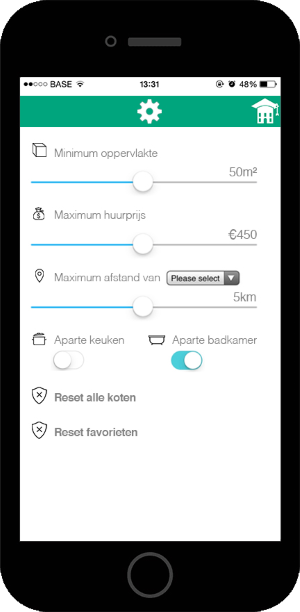
Registratie gegevens:

* 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

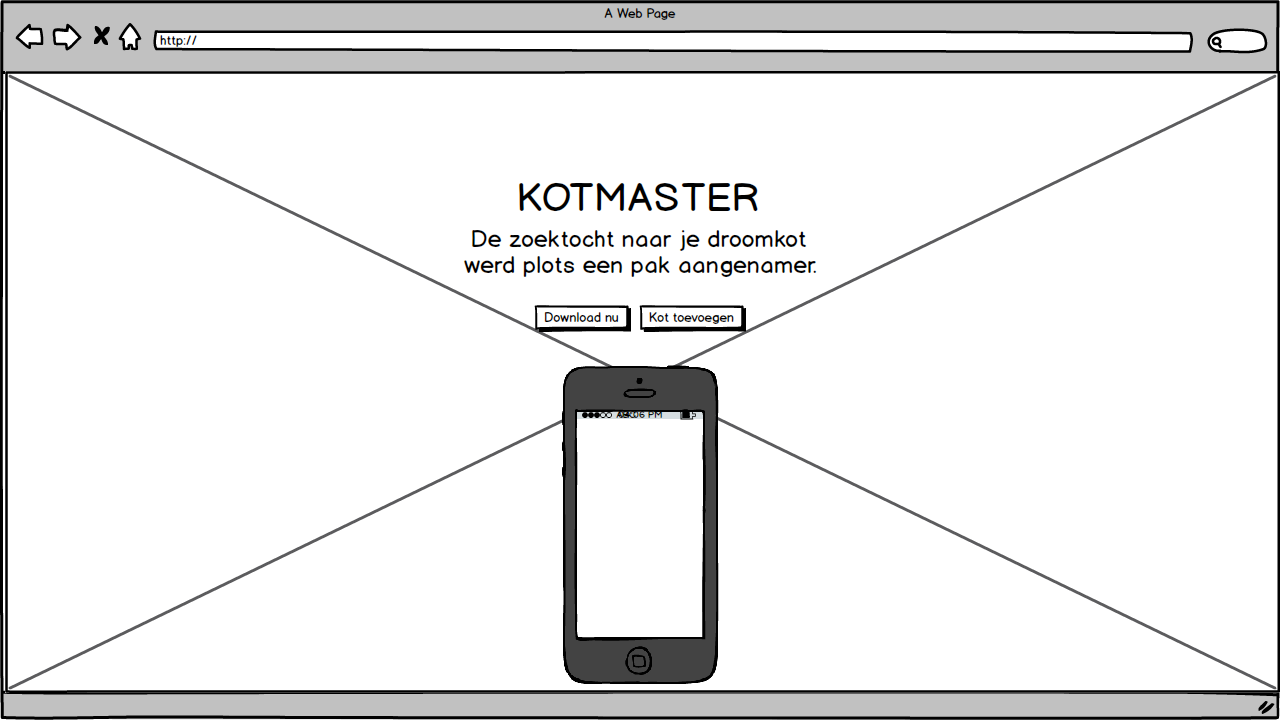


Favorieten:

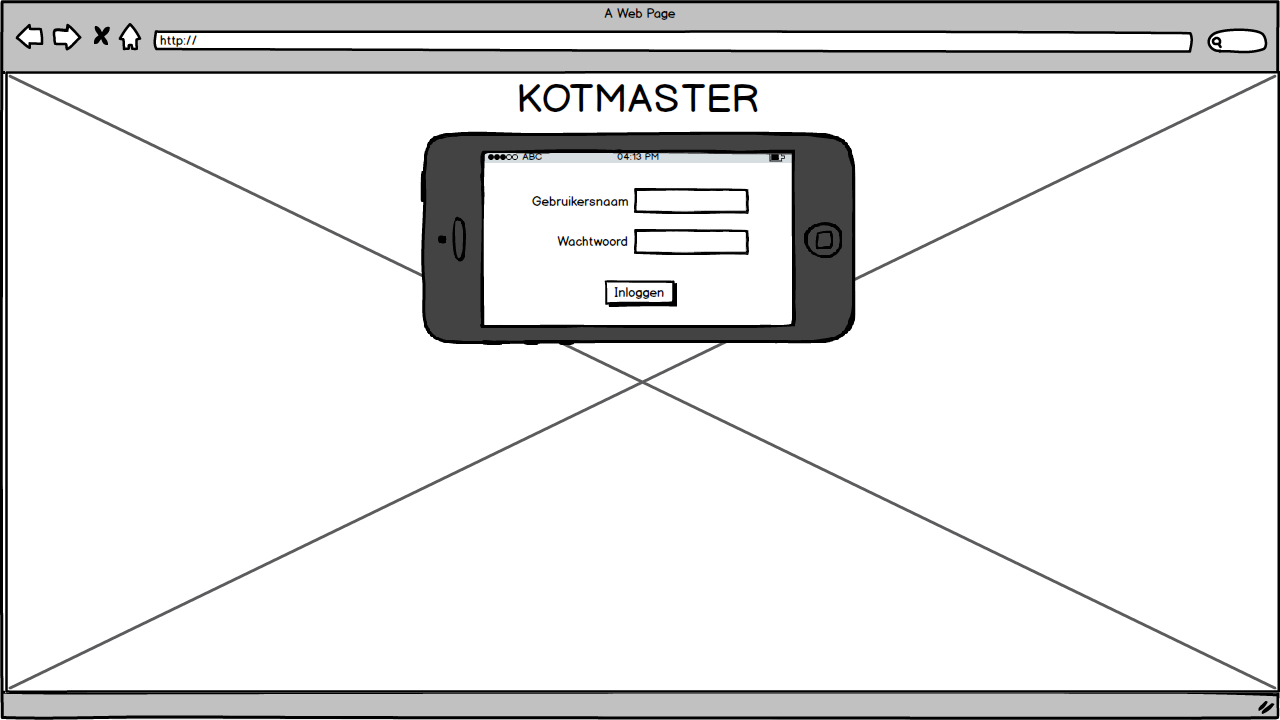
* Beheer favorieten
* Verschillende weergave opties
* Mogelijkheid om te her-selecteren tussen favorieten

## 4.2 Wireframes Website

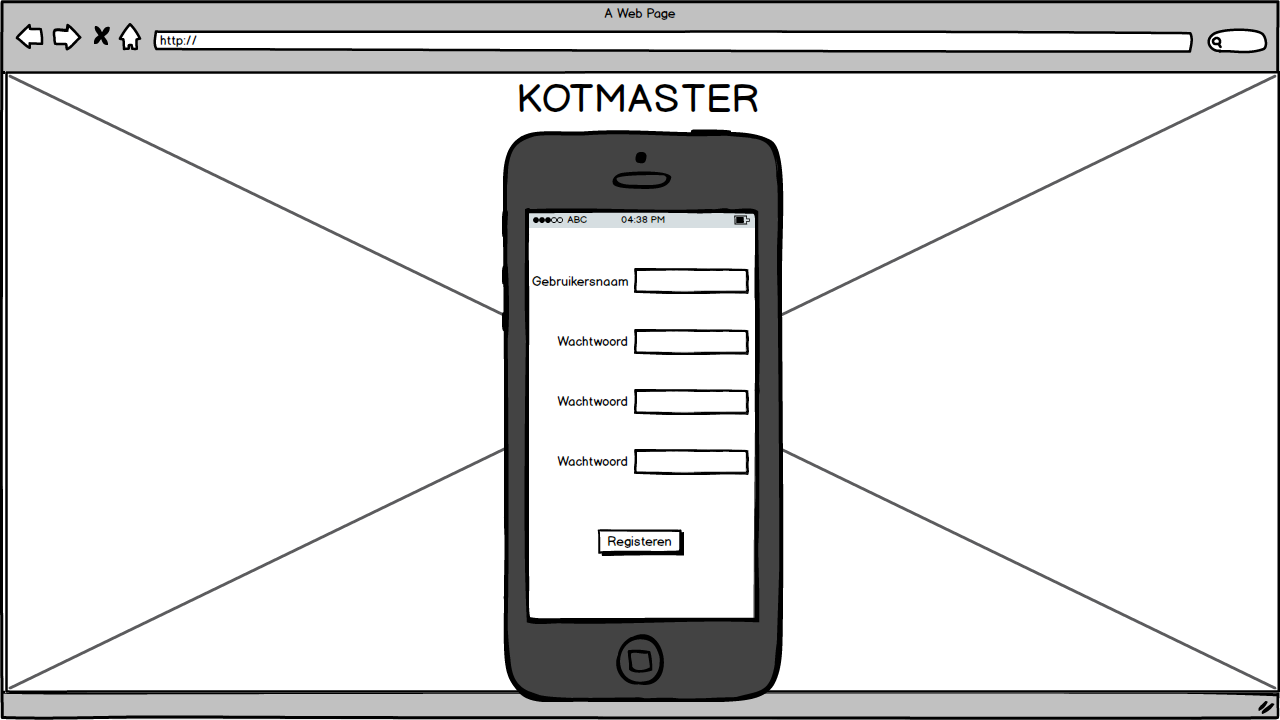
**Oorspronkelijke Balsamiq wireframes**



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



Login scherm in iPhone mockup



Registratie scherm in verticale iPhone mockup

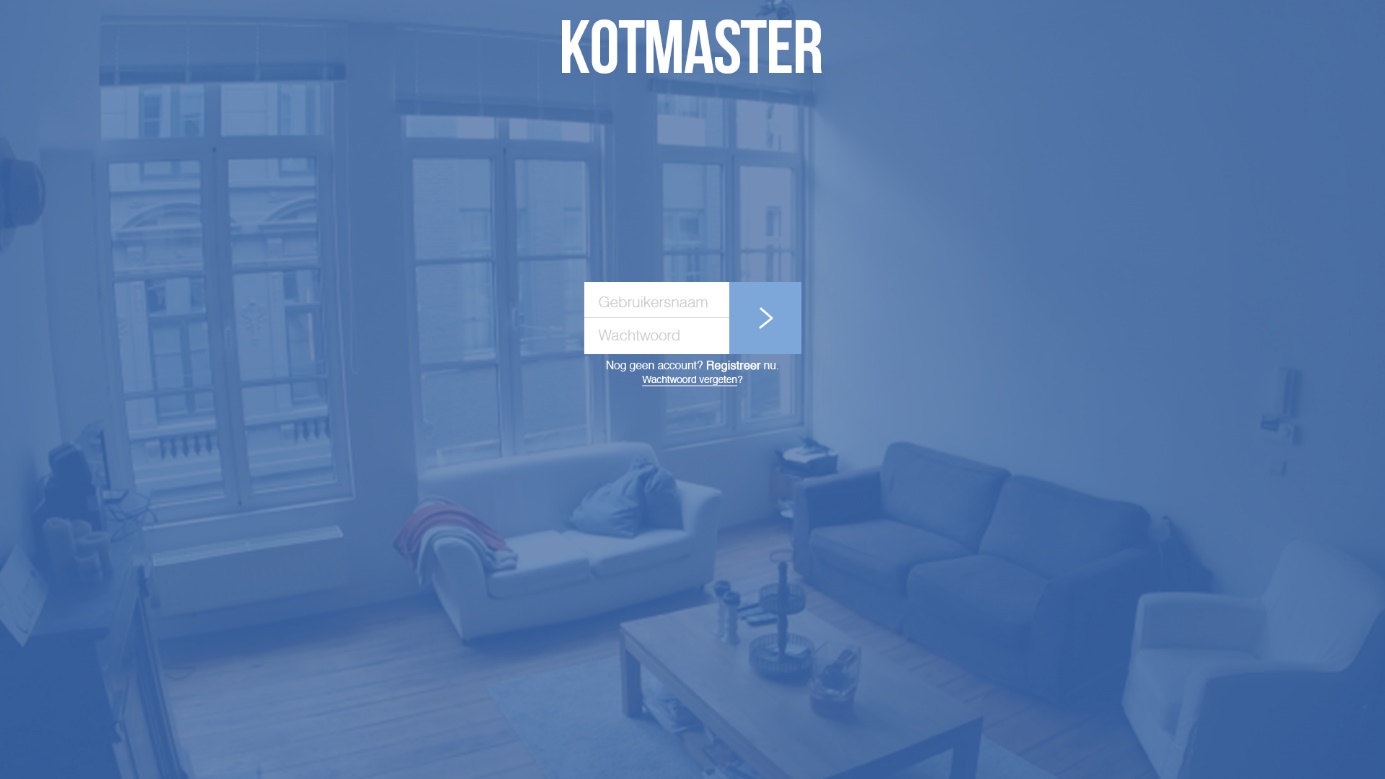
**Photoshop wireframes**



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



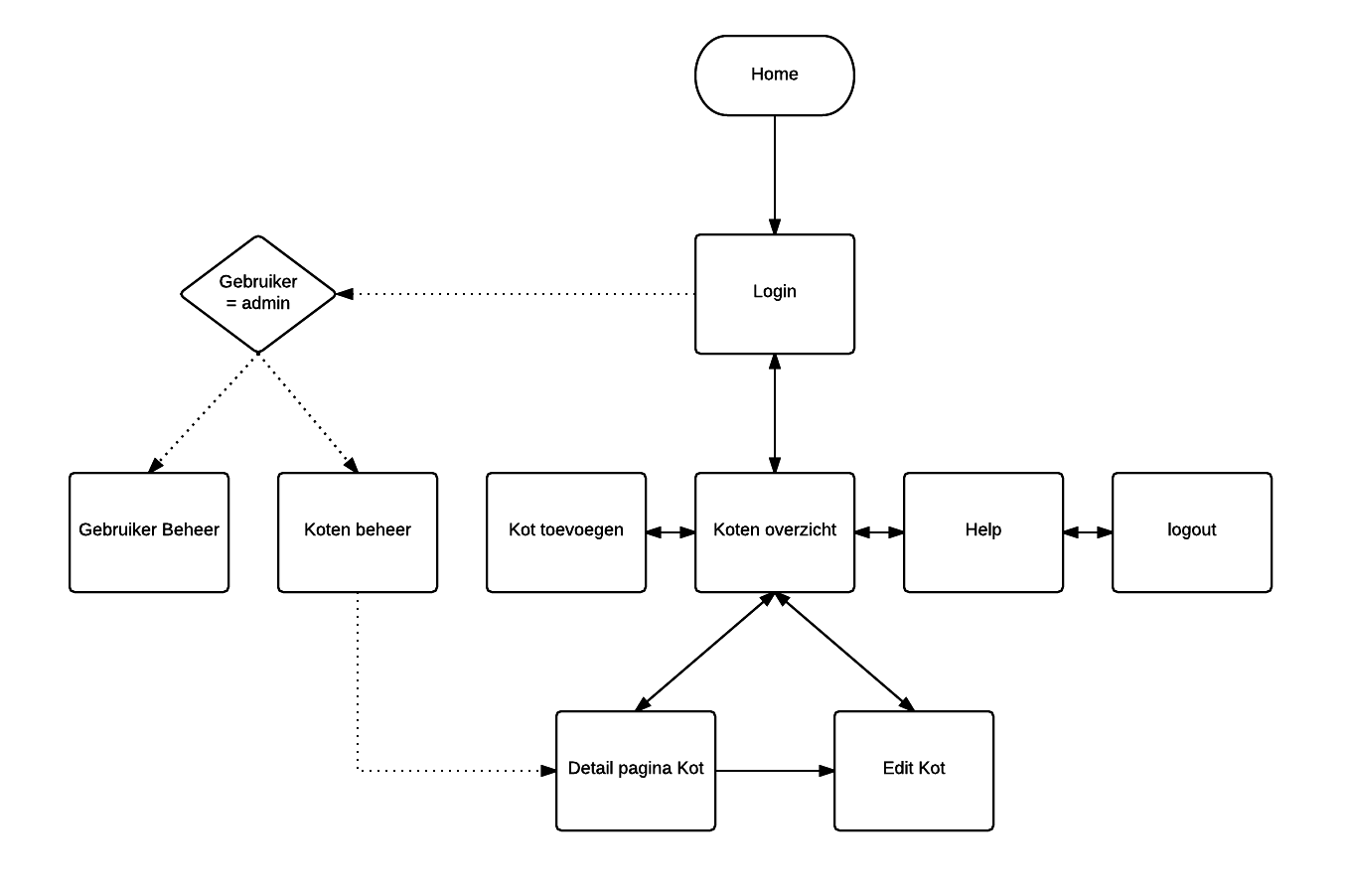
Registratiescherm op zelfde scherm als landing-page  
Eenvoudig clean formulier



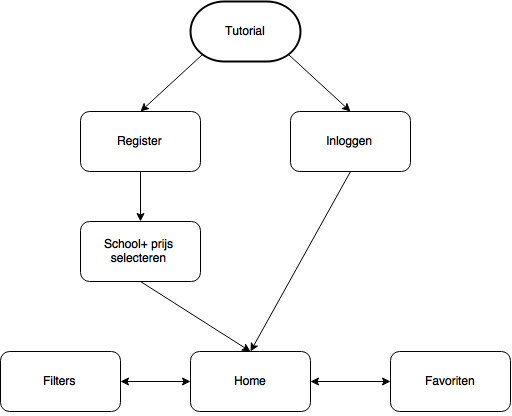
Loginscherm op zelfde pagina als landing-page  
Eenvoudig clean formulier

## 4.3Sitemap

**Website**

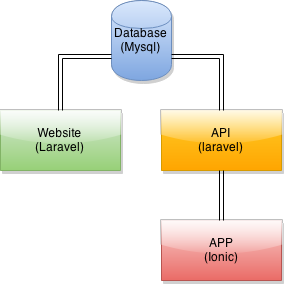


**App**

****

## 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,...

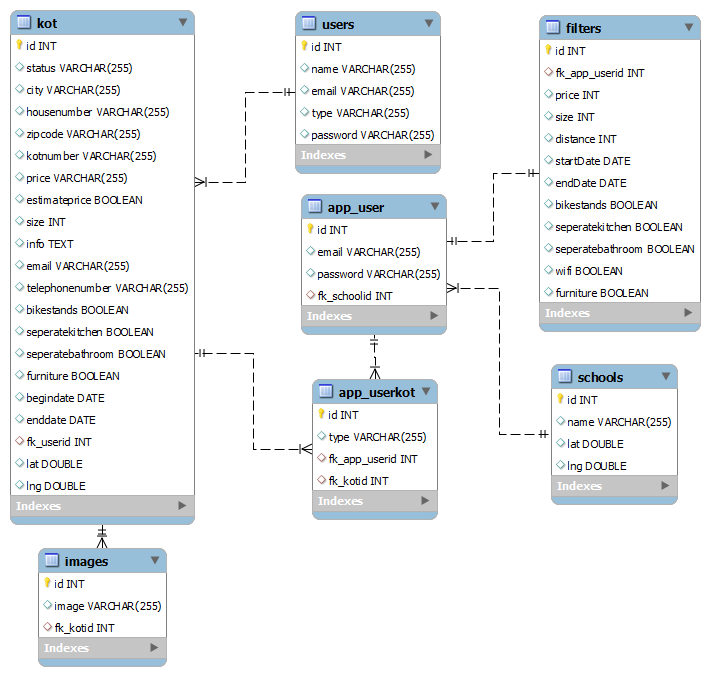
**Ionic**

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>