

Overdracht document

Gemaakt door: Mike van Tienoven

Datum: 25-06-24

Inhoudsopgave

<i>Inleiding</i>	3
<i>Bevindingen onderzoeken</i>	4
Ai implementatie.....	4
Weektaakverdeling.....	4
Best practices andere schoonmaak apps	5
Framework	7
<i>Ustertests</i>	7
Ab-test	7
<i>Toekomstige toepassingen</i>	8
Dynamisch veranderende applicatie	8

Inleiding

Welkom bij het overdrachtdocument voor de ontwikkeling van de schoonmaakapp. Dit document is opgesteld om jou als nieuwe projectverantwoordelijke een duidelijk en gedetailleerd overzicht te geven van het project en om ervoor te zorgen dat de voortgang soepel wordt voortgezet. De schoonmaakapp is ontwikkeld om bewoners van studentenhuizen te helpen bij het efficiënt organiseren en uitvoeren van schoonmaaktaken.

De kernfunctionaliteit van de app ligt in het gebruik van kunstmatige intelligentie om schoonmaakroosters automatisch te genereren. Op basis van de beschikbaarheid van bewoners, zoals opgegeven in hun profielen, verdeelt de AI de taken gelijkmatig en op geschikte momenten. De AI leert van gebruikersinteracties en optimaliseert zodoende steeds beter de roosters, zodat de taken in de toekomst zo effectief mogelijk worden gepland.

Gebruikers ontvangen meldingen wanneer ze een taak toegewezen krijgen en hebben de mogelijkheid om deze taak te accepteren of te weigeren. Bij weigering zal de AI een alternatief schoonmaak moment voorstellen totdat een werkbare oplossing is gevonden. Zodra een taak is toegewezen, krijgt de gebruiker een herinnering om een foto van het resultaat te uploaden. Als de foto niet binnen de gestelde tijd wordt geüpload, wordt dit bijgehouden en weerspiegeld in de gebruikersprofielen, wat kan leiden tot sociale controle binnen het studentenhuis.

Dit overdracht document biedt een overzicht van de tot nu toe voltooide werkzaamheden, inclusief het ontwerpproces en de onderliggende beslissingen, de huidige status van het project en aanbevelingen voor de volgende stappen. We zijn ervan overtuigd dat deze informatie je zal helpen om de verdere ontwikkeling van de schoonmaakapp voort te zetten en te optimaliseren, zodat deze een waardevol hulpmiddel blijft voor studentenhuizen.

Bevindingen onderzoeken

Ai implementatie

Hoofdvraag: Kan AI de input van de huisbewoners meenemen in het schoonmaakschema?

AI kan zeker op basis van de input van bewoners een passend schema maken. De afweging tussen welk model het meest geschikt is komt op het volgende neer:

- Als je data eenvoudig is en geen lange-termijn afhankelijkheden heeft, zou een RNN genoeg kunnen zijn, maar
- Als je data complex is en lange-termijn afhankelijkheden bevat, is een LSTM waarschijnlijk beter

Het doel van de applicatie is om een schema te maken dat leert van de gebruikers bijvoorbeeld wanneer op de dag ze zo een taak willen uitvoeren. Uiteindelijk lijkt het mij een goed idee om gebruik te maken van een LSTM

Weektaakverdeling

Hoofdvraag: Hoe wordt het huidige schoonmaakschema ervaren?

Het huidige schoonmaakschema is niet helemaal goed verdeeld sommige taken zijn groot zoals: de vloer en boven; en sommige zijn heel klein zoals bak trekken (biertje in één keer opdrinken) en bak buiten. Hierdoor train je de bewoners niet dat je altijd in de week wat moet schoonmaken. De taken kunnen beter kleiner verdeeld worden in subcategorieën en kamers, dus keuken: vloer dweilen, afwas of apparaten (airfryer, kookplaat, magnetron/oven) hierdoor worden de taken meer behapbaar en moet iedereen elke week wel een taak doen dus krijg je niet zo een na vakantiegevoel dat je er nog in moet komen.

Best practices andere schoonmaak apps

Hoofdvraag: Hoe zijn andere schoonmaak apps vormgegeven?

Door de analyse van verschillende schoonmaakapps, zoals Tody, OurHome, Sweepy en ChoreMaster, heb ik inzicht gekregen in diverse ontwerpkeuzes en best practices binnen dit genre. Deze apps zijn uniek, maar hebben ook veel gemeen.

Tody valt op door zijn visuele en interactieve dashboard, waarbij het gebruik van tegels en het afvinken van taken de gebruiker in staat stelt om op een intuïtieve manier hun schoonmaaktaken te beheren.

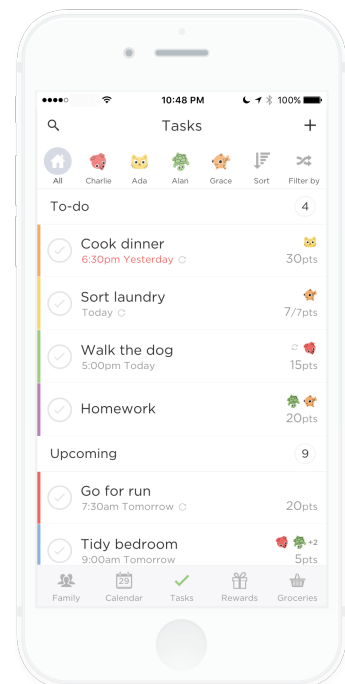
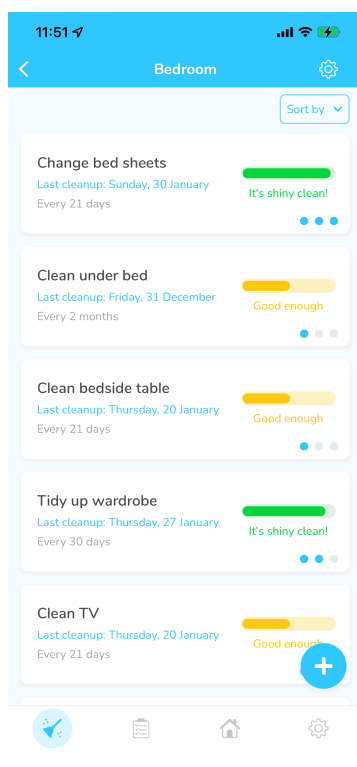
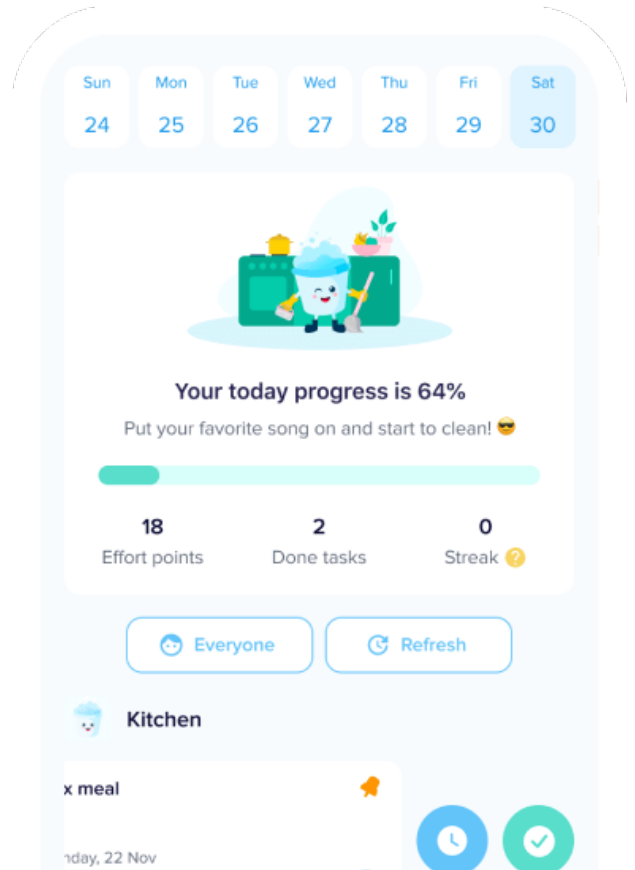
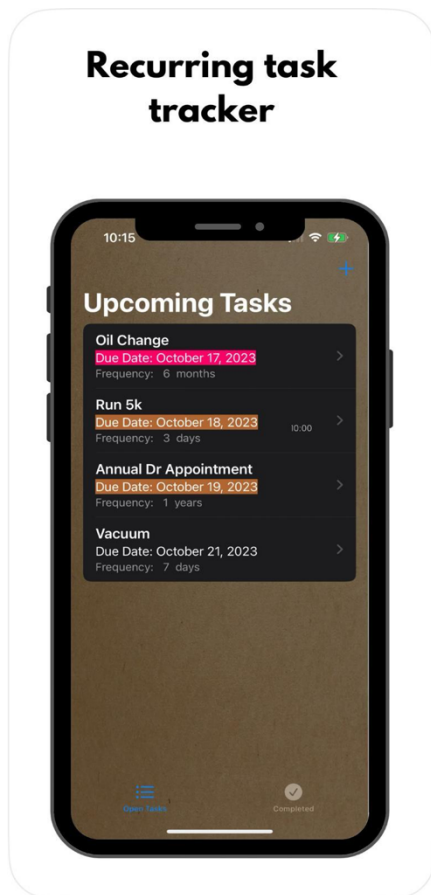
OurHome maakt gebruik van een kalenderlayout en een puntensysteem om gezinsleden te motiveren en verantwoordelijkheden te verdelen, waardoor het huishouden georganiseerd blijft.

Sweepy biedt een gestructureerde ervaring met een kalenderlayout en gedetailleerde weergave van taken, waardoor gebruikers georganiseerd blijven en geen enkele taak over het hoofd zien.

ChoreMaster kiest voor een minimalistische aanpak met een eenvoudig ontwerp en een 'slide'-methode voor het voltooien van taken, wat zorgt voor een gestroomlijnde en gebruiksvriendelijke ervaring.

Deze best practices, waaronder visuele aantrekkelijkheid, gebruiksgemak en efficiëntie, kunnen worden toegepast om nieuwe, innovatieve oplossingen te creëren die de gebruikerservaring verbeteren en bijdragen aan een georganiseerd en efficiënt huishouden. De minimalistische aanpak van choremaster is een goed startpunt, wel denk ik dat je de kalender layout van de andere apps terug moet laten komen zodat je kunt zien op welk moment in de week je de taak moet uitvoeren, aangezien de doelgroep hulp nodig heeft met de precieze planning hiervan.

Best practices bijlage



Framework

React Native biedt de voordelen van cross-platform ontwikkeling en een snelle ontwikkelingscyclus, ondersteund door een grote en actieve community. Dit maakt het een goede keuze, vooral omdat ik al ervaring heb met webontwikkeling en JavaScript. De mogelijkheid om één codebase te onderhouden voor zowel iOS als Android verlaagt de ontwikkelingskosten en -tijd aanzienlijk, wat essentieel is voor de snelle lancering en uitbreiding van de applicatie.

Hoewel Swift een uitstekende keuze is voor het ontwikkelen van native iOS-apps met hoge prestaties en naadloze integratie van iOS-specifieke functies zoals Memojis, vereist het een diepere kennis van de Swift-programmeertaal en de iOS-ontwikkelomgeving. Dit zou leiden tot een langere leercurve, vooral gezien mijn huidige ervaring. Swift kan echter in de toekomst een betere optie zijn wanneer de applicatie groeit en meer geavanceerde functies nodig heeft die alleen native ontwikkeling kan bieden.

Flutter biedt ook sterke prestaties en consistente UI/UX-ervaringen op meerdere platforms, maar de complexiteit bij het integreren van zeer specifieke iOS-functies zoals Memojis kan een belemmering vormen.

Daarom kies ik ervoor om met React Native verder te gaan voor de ontwikkeling van mijn schoonmaakapplicatie. Dit framework biedt de flexibiliteit en efficiëntie die nodig zijn om snel te kunnen ontwikkelen en uitbreiden, terwijl ik mijn bestaande kennis en ervaring optimaal kan benutten. Naarmate de applicatie groeit en meer geavanceerde functionaliteiten vereist, zal ik overwegen om over te stappen naar Swift voor de ontwikkeling van een hoogwaardige native iOS-ervaring en implementatie van native ios features zoals memoji.

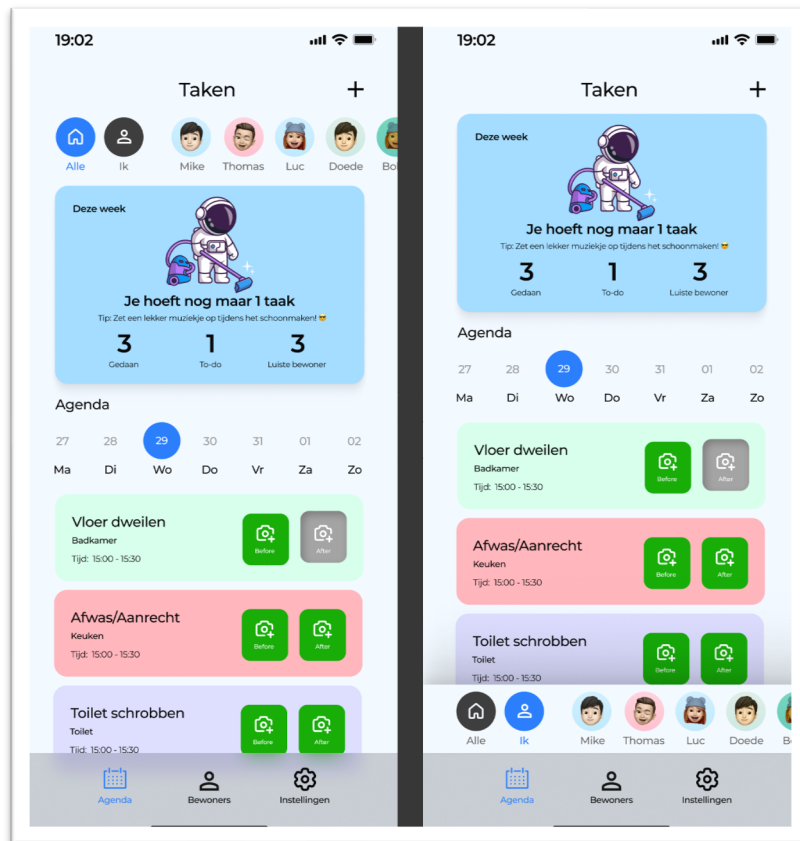
Usertests

Ab-test

De navigatiebalk zal het beste bovenaan het scherm geplaatst worden, ondanks hij makkelijker te bereiken is onderaan, voelt de flow natuurlijker aan bovenaan. Zo zou onderaan “out-of-place” lijken en verdwijnen onder de gebruikers vingers en te dicht staan op de nav balk. Bovenaan zou moeilijker te bereiken zijn, maar voor alsnog de beste keuze zijn

Hoofdvraag: in welke versie komt de navigatie balk het best tot zijn recht?

Bovenaan.



Toekomstige toepassingen

Dynamisch veranderende applicatie

In dit project probeer ik ervoor te zorgen dat als je er goed c.q. slecht voor staat je dit zult merken in de applicatie, huidig wordt die dynamiek alleen doorgevoerd in de headercards die van kleur en foto veranderen. Ik had het idee om dit over de gehele UI te implementeren zodat op welk scherm je ook bent in de app, je merkt of je er goed of slecht voor staat. Dit wilde ik bereiken d.m.v. het dynamisch veranderen van de achtergrond op basis van je voortgang.



Echter roept dit wel een aantal vragen met zich op namelijk:

- Zijn alle elementen nog steeds goed leesbaar met de nieuwe achtergrond kleuren?
- Hoe ga je dit implementeren met darkmode(bij darkmode inschakelen zorgt ervoor dat de ervaringen niet gelijk zijn aan elkaar.)
- Toen ik het liet zien aan test personen hadden sommige het niet eens door, misschien is de kleurverandering te subtiel

Om deze psychologie verder te borduren, zou je ook bijvoorbeeld, als je de app opent at er dan eerst een felle rode achtergrond met een animatie van een hysterisch personage aan het schoonmaken, en dat deze dan verdwijnt in de headercard als een soort gamificate effect. Wellicht is het sowieso een gaaf dynamisch effect om die plaatjes te laten bewegen.

Nieuwe iteratie leaderbord:

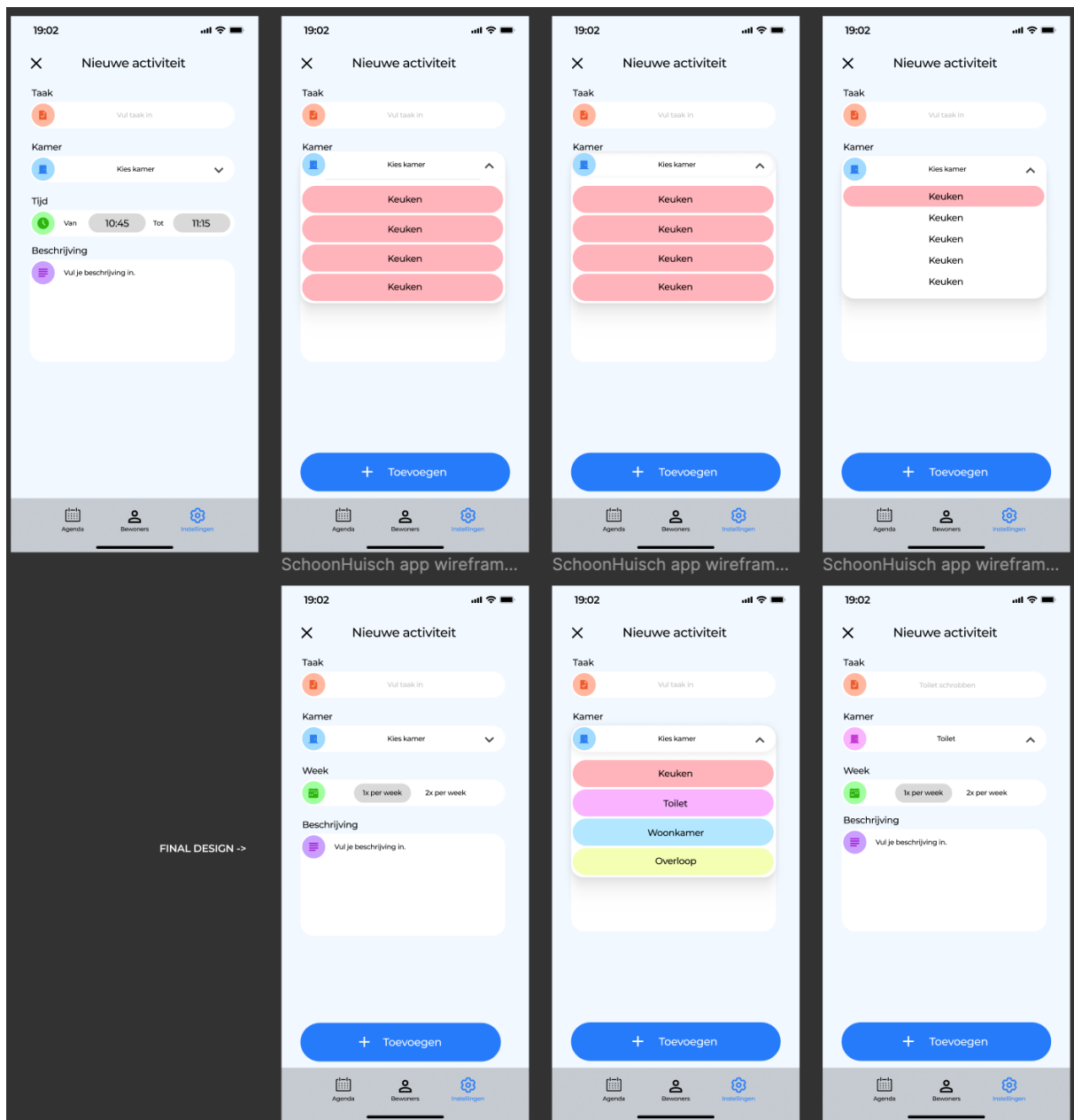
Ik heb een keer geprobeerd de kleur van de hele balk in het leaderbord aan te passen, maar misschien is het nog duidelijker wie er eerste en laatste staat als nu, maar moet je met pastelkleuren werken. Zodat de styleguide gevolgd wordt en de dynamische achtergrond.

Momenten wijzigen

In het interview over het schema is naar voren gekomen dat soms de planning van een persoon op de dag zelf verandert, de bewoners moeten de mogelijkheid hebben om eerder geaccepteerde taken toch uit te voeren. Dit roept een aantal met zich op:

- Als je altijd kan wijzigen dan levert het puntensysteem niks meer op, want dan kun je last minute telkens een dag verder schuiven. (Misschien kun je alleen in de week zelf schuiven, maar moet het aan het eind van de week af zijn)
- Kan bijvoorbeeld alleen de huischpreases taken verschuiven. Kan de persoon een aanvraag doen en moet de preases dit goed keuren? Moet de bewoner zelf 'irl' naar de preases toe?

taken toevoegen scherm



Ik heb 2 versies gemaakt, de eerste had ik de tijd op aangegeven, maar dat moet door de AI gedaan worden dus het is niet logisch dat die optie er zou staan en de tijd toch niet klopt. Ik heb nu dat als je een kamer geselecteerd hebt het logo bij kamers met de kleur mee veranderd, zo kan ik de gebruiker trainen op die kleuren. Is dit duidelijk genoeg? 1x per week, 2x per week. Is dit genoeg controle?