

Functioneel Ontwerp

Biz Brew

Tim Bouma

27-05-2021

# 

[**Requirements**](#_py5nvtdhtwku) **1**

[**Schema**](#_z9y021ie94y) **2**

[**User Interface**](#_1lla6wu9yb1w) **3**

[**Aanpassingen**](#_96yi2476bqbc) **4**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Requirements

|  |  |
| --- | --- |
| **Requirements** | **Prioriteit** |
| **Business Requirements** |  |
| Opleiden van een stagiaire | hoog |
| Overzicht van koffie voorkeur van de werknemers | hoog |
| **Functionele Requirements** |  |
| Inloggen | hoog |
| Registreren | hoog |
| Registreren van koffie voorkeur   1. Aantal suikers 2. Koffie sterkte 3. Wel of geen melk (nieuw) | hoog |
| Wachtwoord resetten (nieuw) | hoog |
| Connectiviteit controle (nieuw) | hoog |
| ***Non Functionele Requirements*** |  |
| Validatie van gegevens voordat dat ze worden opgeslagen in de database (nieuw) | hoog |
| Programmeer structuur aanhouden voor makkelijk onderhoud | medium |
| Beheersen van Flutter en Firebase fundamentals | medium |
| Firebase Firestore voor realtime updates | hoog |
| Firebase authenticate voor het inloggen en registreren | hoog |
| Github gebruiken voor het archiveren en versiebeheer van de code (nieuw) | low |

# Schema

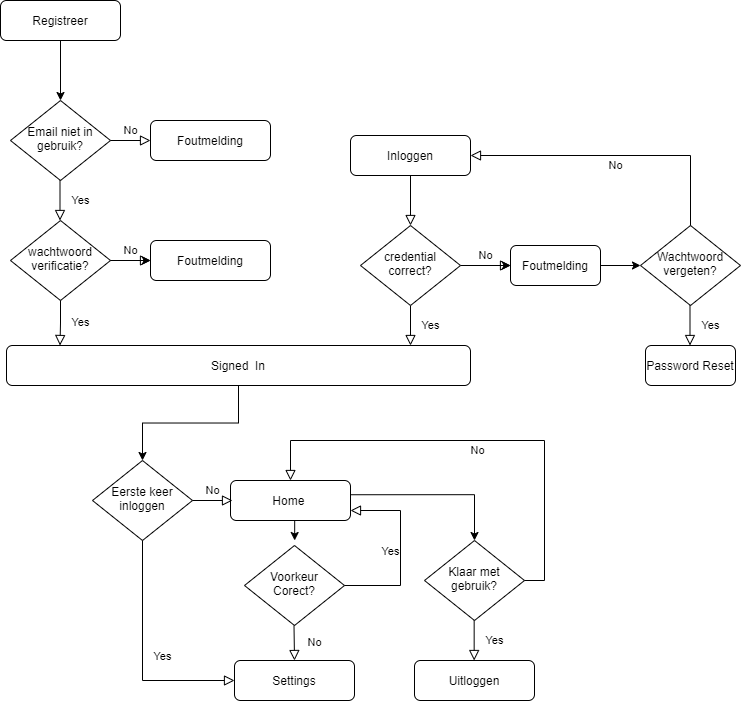
**Schema uitleg**

*Not Signed in*

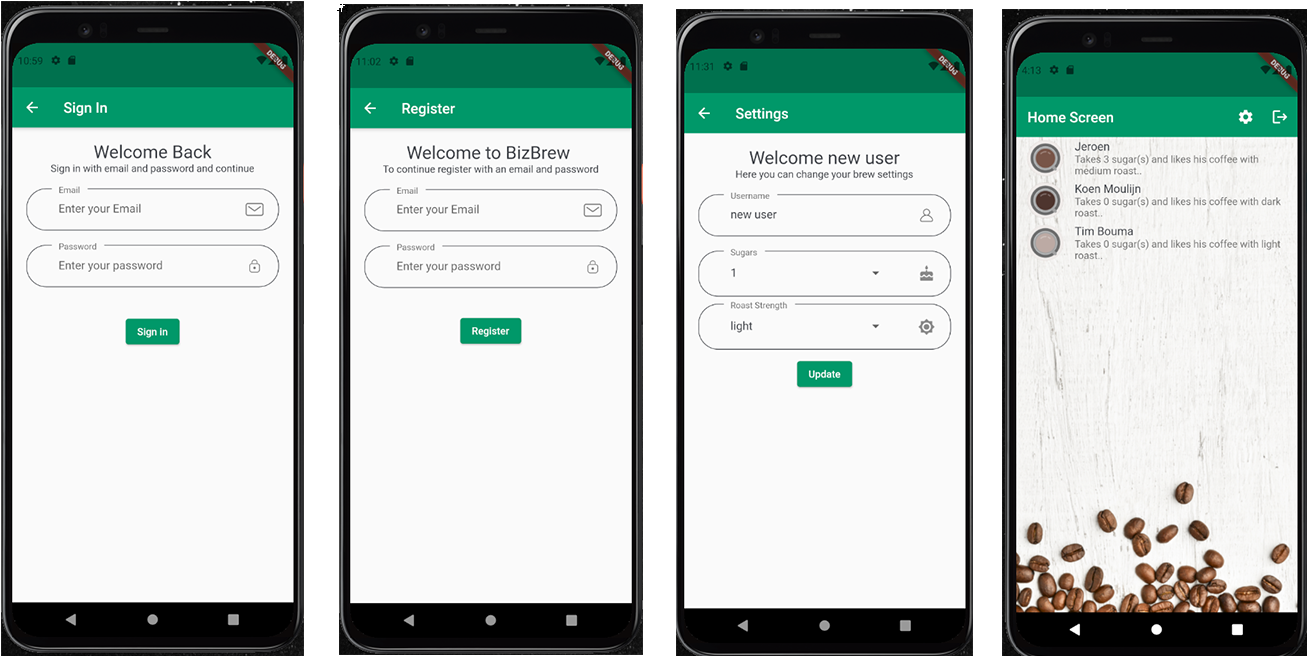
1. De gebruiker kan *inloggen* of *registreren*
2. Nieuwe gebruikers *registreren*
3. Bestaand gebruikers loggen in
4. Zowel na *inloggen* als *registeren* ben je signed in

*Signed In*

1. Gebruikers arriveren door de wrapper op *Home*
2. Nieuwe gebruiker worden gelijk naar het *settings* genavigeerd om koffie voorkeur aan te passen.
3. Gebruiker kunnen de kofie voorkeur van hun collega's bekijken.
4. Als je voorkeur niet correct dan verander je die in de *settings*.
5. Ben je klaar met het gebruik van de applicatie.
6. *Uitloggen* met een uitlog knop.



# User Interface

De applicatie word gebouw met het framework van Google Flutter. Dit framework maakt gebruik van Material Design. Je kan dit framework vergelijken met Bootstrap. Flutter bestaan uit widgets. Elk widget is ontworpen aan de hand van Material Design. Als developer kan je met decoratie en style properties het ontwerp van de applicatie aanpassen.

Het ontwerp staat tijdens dit project niet centraal. De focus ligt voornamelijk op het ontwikkelen van de applicatie. Denk hierbij aan het leren van de fundamentele functie van Flutter en Firebase.

Het design moet vooral duidelijk zijn voor de gebruiker. Tijdens het inloggen en registreren moeten foutmeldingen overzichtelijk worden weergegeven. Zodat de gebruiker gelijk zie wat er fout is gegaan. Wat voor een gebruiksvriendelijke ervaring moet zorgen. Naast het functionele design mag de applicatie er wel goed uitzien, Zodat de gebruiker ook de applicatie graag willen gebruiken. Niemand wilt namelijk een lelijk applicatie openen.

Om het ontwerpen van de applicatie zo flexibel mogelijk te maken wordt er een constants en theme file aangemaakt. In de constants.dart worden kleuren en vaste gegevens bewaard. In het theme.dart worden bepaalde widgets voor gestyled zoals buttons. Deze voor ontworpen widgets worden hergebruikt in de applicatie. Het voordeel van werken op deze manier is als de opdrachtgever zich bedenkt over bepaald ontwerp keuze. Dan hoeft alleen aan de constants en theme file code worden veranderd.

# Aanpassingen

De applicatie is getest met een acceptatietest en in gebruikgenomen. Hier is feedback uitgekomen die in de applicatie is aangepast.

**Acceptatie test**

* Via de tekst kunnen navigeren naar inlog en registratie scherm.
* Gebruiker verwacht tijdens het registreren en inloggen een foutmelding na verkeerd invoeren van e-mail of wachtwoord.
* Ook wil de gebruiker de gebruiker gelijk na het registreren zijn username en koffie settings aanpassen.

**Incidentmeldingen**

* De Google-service.json key moet een reset krijgen.
* De google-service.json moet uit de GitHub geschiedenis worden verwijderen.
* Bij het registreren van een account moet de gebruiker zijn wachtwoord kunnen verifiëren ter controle.
* Het registreren met een emailadres wat al in gebruik is zorgt voor het vastlopen van de applicatie. (Zorg voor extra validatie)
* Wachtwoord reset optie inbouwen, voor het geval dat een gebruiker zijn wachtwoord is vergeten.
* In vliegtuigmodus (ofwel wanneer de applicatie geen connectie heeft met internet) is de applicatie onbruikbaar. Geef een foutmelding.
* Wanneer een gebruiker inlogt of registreert met verkeerde gegevens zorgen de foutmeldingen voor render overflow. Daarnaast wijken de foutmeldingen af van het ontworpen design.
* De app bar titels bijwerken.
* Een veld toevoegen in het setting scherm of de gebruiker wel of geen melk in zijn koffie wilt.

De feedback is geanalyseerd en verwerkt in de code. Het verloop van de analyses en het verwerken van de acceptatietest is uitgebreid beschreven in de kerntaak 3. De incidentmeldingen zijn beschreven in het eerste deel van kerntaak P1-K1-W1. Hierin staan de mogelijk oplossen voor de meldingen. Naast de oplossingen staan er ook stukjes code. Deze geven aan hoe de problemen zijn verwerkt. De code is beschikbaar in het GitHub porject: <https://github.com/Baqiwaqi/Biz_Brew>. Het resultaat is te zien in de volgende afbeeldingen.

**UItleg**

1. Navigeren via text knoppen naar het inlog en registratie.
2. Registratie uitgebreid met display name, repeat password and password reveal.
3. inloggen voorzien van een wachtwoord reset.
4. Settings uitgebreid met extra form veld voor melk voorkeur.
5. Foutmeldingen zijn volgens het design van de applicatie.
6. Wanneer er geen internet verbinding is en de gebruiker is wel ingelogd dan wordt No Internet weergegeven.

