Introduksjon (*Mobil-applikasjon*)

Vi i gruppe 2 har bestemt oss for å prøve å gi Kolonial.no en ny ekstra løsning som vi tenker vil løfte interessen hos dagens foreldre med mange daglige oppgaver og store ansvarsområder. Da kolonial.no allerede har en ganske solid og strukturert løsning både på nettsiden deres og mobil-applikasjonen var dette en kjempeutfordring, men vi kjempet oss igjennom. Da småbarnsforeldre er målgruppen tenkte vi å redusere deres ansvar enda mindre med et abonneringssystem som vil tillate enkelte å få varer levert til en fast tid. Dette kan være dagligdagse behov som tannkrem, håndsåpe , tannbørster som må byttes jevnlig, tørkeruller og slike som ikke går lett ut av dato og som brukes jevnlig i huset. Ting som ikke vil plage kunder om en skulle hatt en ekstra i huset da det vil etter hvert bli behov for bruken av det om kort tid. Et abonneringssystem vil også bidra med å løfte en sikker økonomi for Kolonoal.no samtidig som det vil forenkle brukers behov.

Det ble litt vanskelig å finne en løsning og oppgave der alle kunne bidra. Derfor tenkte vi at et stort selskap som Kolonial.no, som selvfølgelig også har en mobil applikasjon som forenkler bruken på mobil, også kunne trengt dette der. Selv om vårt hovedfokus har vært en webside, bestemte vi oss for at noen av oss skulle prøve oss på å legge til funksjonen på en applikasjon på en mest mulig brukervennlig måte for å se for oss hvordan den ville blitt der og for at alle skulle bidra til å gjøre noe.

Grunnidé og forventninger *(Mobil-Applikasjon)*

Vi var en gruppe på 3 som tok for oss oppgaven. Ideen kom inn i bilde da vi innså at alle tre hadde Android programmering samme semester, men grunnet sen avgjøring for endelig løsning gikk vi ikke for native Android løsning da dette er mye mere tidskrevende da det krever mye mer koding for små operasjoner. Løsningen ble til at en av oss hadde Kryssplattform emnet samme semester og foreslo at vi skulle gå for en hybrid applikasjon. Fordi Hybrid teknologi er web-basert teknologi ville dette likevel gi gode arbeidsfordelinger til alle tre. Med bl.a ferdiglagde-komponenter levert fra rammeverket Ionic ble den eneste utfordringen å fordype Typescript kunnskapene da back-end delen bygges på dette. Andre naturlige årsaker til at vi valgte hybrid teknologi er at det er kryssfunksjonelt for ios, Android og Sony på den måten at hele eller deler av koden kan gjenbrukes for fremtidig overføring til ios eller Sony plattformer.

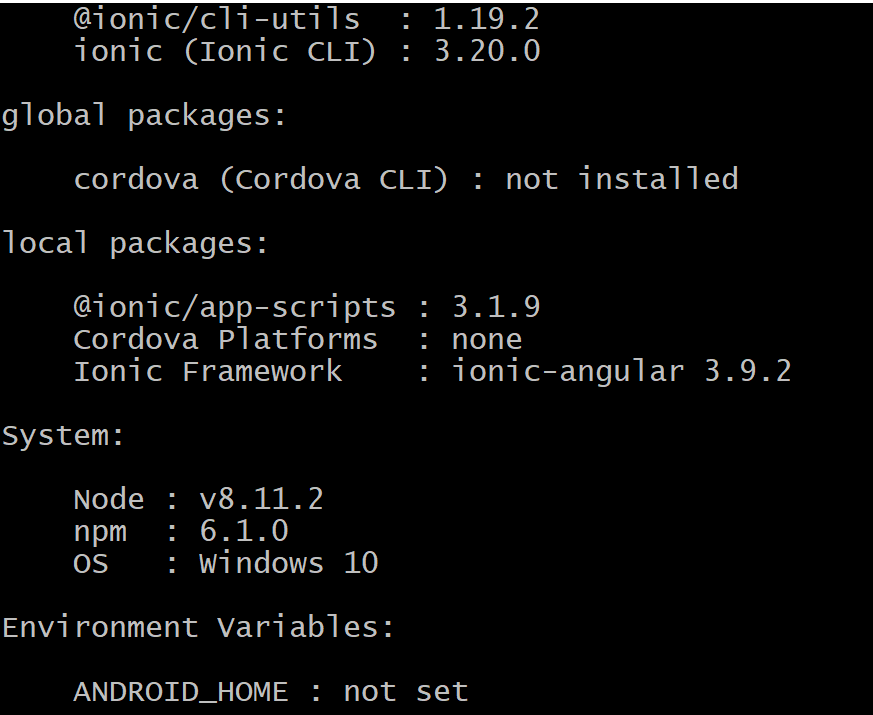
Forventningene var ikke veldig store da dette er relativt ukjent og nytt for alle, men vi utfordret det likevel da vi så på dette som eneste løsning og en spennende utfordring. Appen skulle se ut som den opprinnelige så mye som mulig der vi kun skulle prøve å finne plass for å dytte inn den nye ide’en vår. Fordi idéen baserer seg på å abonnere på varer så vi på det som lite hensiktsmessig å bruke tid på å få mange kategorier og sider til å fungere fra den opprinnelige og gikk da for å fokusere på en side der en kan plukke ut varer for så å kunne legge til i abonneringer og en side der en kan få oversikt over sine abonnerte varer eller informasjon om opprettet abonneringsavtale. Applikasjonen trenger også en side der brukeren oppretter selve abonnementet for levering etter intervaller og muligheten for å kunne endre på avtalen.

Verktøy (*Mobil-Applikasjon*)

*Teknologi, IDE og Rammeverk*

***Ionic3***: Basert på rammeverkene Angular4 (web-rammeverket og viewlaget) , Cordova (som eksponerer native-funksjonalitet til hybrid-appen) og teknologien TypeScript (ecmascript6) som kompileres til javascript.

Ionic og Node versjon



**Utviklingsmiljø (IDE):** Selv om gratis verktøyet VScode er mest brukt og anbefalt på nettet gikk vi for Webstorm, som er utviklet av JetBrains da vi får dette gratis gjennom skolen. Webstorm tilbyr visse funksjonaliteter som VScode ikke har som bl.a gjør det enklere å navigere til forskjellige områder på en hensiktsmessig måte. Andre grunner er fordi vi hele veien til nå har brukt Intellij innen programmering som også er utviklet av samme selskap og som ligner veldig.

*Plattform og emulator*

**Utviklet med operativsystem**: windows 10 home edition

**Utviklet for mobil plattform :** Android

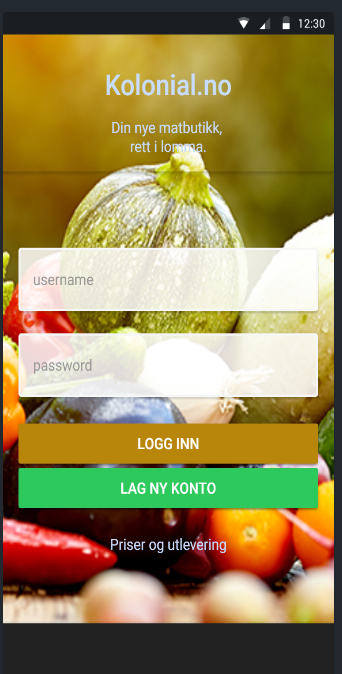
**Testet med emulator:** Nexus 5x API 24

Funksjonalitet og design (mobil-applikasjon)

*LoginPage*

Det første brukeren blir nødt til å gjøre er å registrere seg med et nytt brukernavn og passord. Firebase tilbyr et ferdigutviklet autentiseringssystem med passord og brukernavn der en kan velge mellom brukernavn-typer som bl.a facebook-id, github-id eller en ny email-id som unike id-er. Vi har gått for ny email som brukernavn-type med 6 tegn/siffer passord. Av andre tekniske oppgaver firebase tar seg av er feilmeldinger ved feil inntasting av brukernavn eller ugyldig format med passende feilmeldinger. Ved oppretting av ny konto vil bruker navigeres til HomePage med en passende velkomstmelding som er blitt lagt til manuelt.

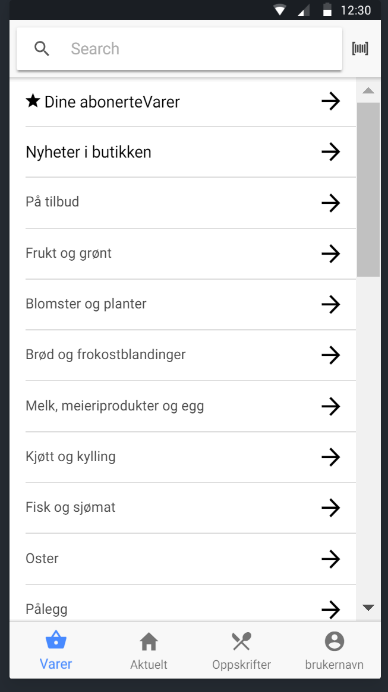
*Login-page*



*HomePage*

Denne siden ser nesten identisk ut som kolonial sin der elementer i listen som har funksjoner er farget med sort med litt større skrift og elementer uten funksjon har grå farge. Som tidligere nevnt så vi det ikke hensiktsmessig å legge til alle funksjoner i denne listen da alle har en og samme intensjon så vi valgte en å gå ut fra hvor ‘Nyheter i butikken’ var den som ble valgt.

Home-Page

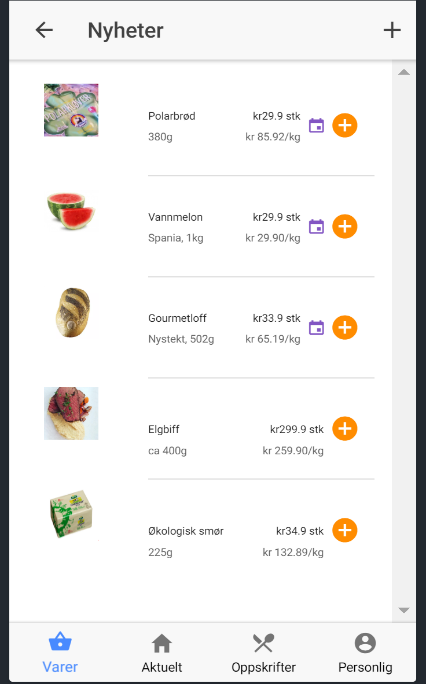


*NyheterPage*

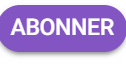
Denne siden har navigasjonsfelt med en pil-knapp som tillater bruker å navigere tilbake til HomePage og et ‘+’ tegn for å opprette et abonnement. Bruker får en rask overblikk over alle varer som er kommet inn. Alle varer er klick-bare og navigerer bruker videre til ‘*DetailPage*’. Varene og detaljer om de enkelte varene som vises er hentes fra databasen i sanntid asynkront og varer som bruker abonnerer på vil merkes med en liten kalender ikon

 tilbydt av ionics ikon samling ‘Ionicons’. Ikonet er også trykkbar og vil navigere bruker til DineEnkeltvarerPage som gir en oversikt over alle abonnerte varer. Den oransje pluss knappen  har ingen funksjon , men er der fordi kolonial har en slik i den opprinnelige der funksjonen er å velge antall varer en vil handle for gjennom denne siden. Vi valgte bort funksjonen i denne siden da fokuset rettet seg mer mot abonneringen av varen som kan gjøres i DetailPage hvor antall kan velges på samme måte som en skulle kjøpe en vare.

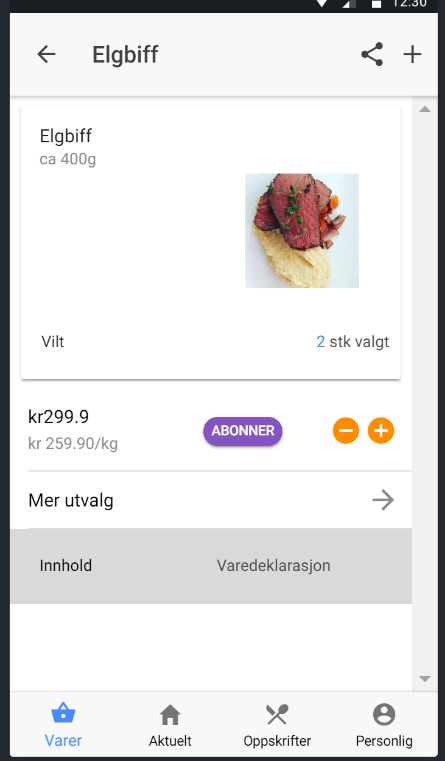
*Nyheter-Page*



*DetailPage*

Her ser siden også nesten helt identisk ut som den opprinnelige hvor vi har lagt til en ekstra ’+’ knapp i navigasjonsbaren øverst der bruker også kan navigeres til for opprettelse av abonnement på samme måte som i NyheterPage. Når en bruker trykker på en vare i NyheterPage vil bruker bli navigert til denne siden. Siden skal vise detaljer om det valgte produktet og bruker skal kunne velge antall varer han vil handle eller abonnere på. Ved å trykke på ‘abonner’ knappen  vil gitt antall bli lagt til i abonnementet under ‘Abonneringer/Enkeltvarer som en kan navigeres til ved å trykke på ‘Personlig’ ikonet nederst i tabben. Knappen vil ikke bli tilgjengelig hvis bruker ikke har valgt antall han/hun vil ha , men vil fremdeles være synlig i bakgrunn for å påminne bruker om muligheten . Knappen vil heller ikke være tilgjengelig hvis bruker allerede abonnerer på varen.

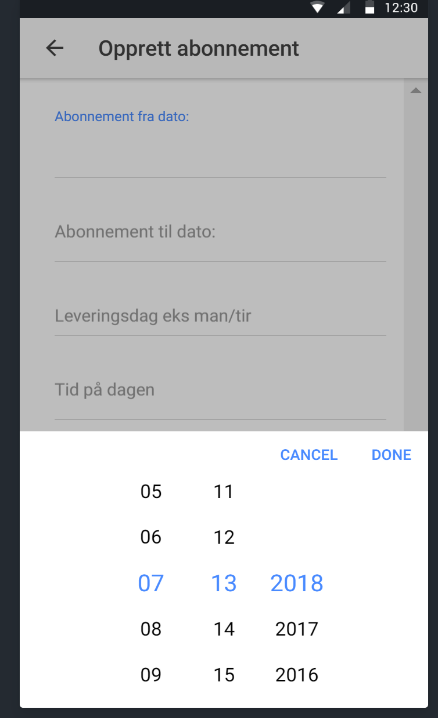
DetailPage



*AbonnerPage*

Fordi ‘+’ knappen, som navigerer bruker til denne siden, er litt er ukjentfor bruker vil ved navigering til siden starte med en passende melding som forklarer hva en skal gjøre og hvorfor. Her kan bruker opprette et abonnement ved å velge *dato fra* og *dato* *til* periode for når fast levering skal forekomme. Inputfeltet som er brukt her er hentet fra Ionic’s utvalgte ferdig componenter (ion-datetime) som har en fin rullerende animasjon for dato valg. Dette forenkler valget og gir bruker en behagelig følelse når formatet er satt opp riktig og sparer også bruker for litt tid. Etter opprettelsen vil all data lagres i firebase i en egen kolleksjon som heter ‘Abonneringer’ med brukerens id lagret i samme dokument knyttet til abonnentet. På denne måten vil det bli enkelt å hente ut riktig data knyttet til innlogget bruker. Deretter vil bruker navigeres ut tilbake til DetailPage eller NyheterPage avhengig av hvor han/hun var først med en passende melding som bekrefter om opprettelse er fullført eller ikke.

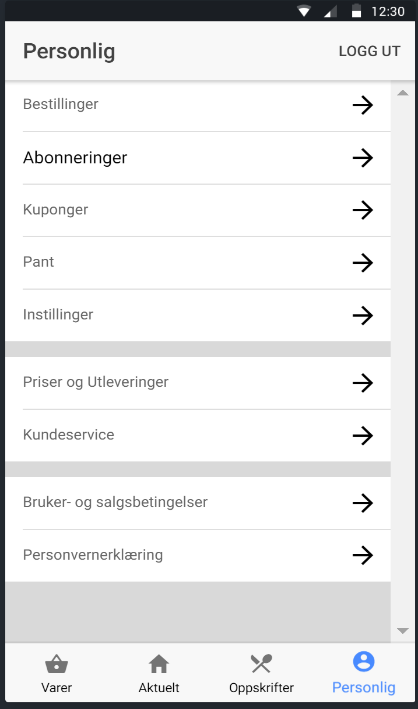
AbonnerPage



*UserInfoPage*

Ved å trykke på ‘Personlig ‘ knappen nederst i tabs-baren navigeres brukeren til denne siden. Siden er i bunn og grunn helt lik Kolonial.no sin der vi fant det mest passelig å kunne presse inn en vei inn til en side hvor bruker skal kunne få oversikt over informasjon om sitt abonnement. Opprinnelig skal overskriften ‘Personlig’ erstattes med et brukernavn, bruker kan foreløpig ikke velge et så det ble erstattet med noe mer passelig som gir mening for bruker.

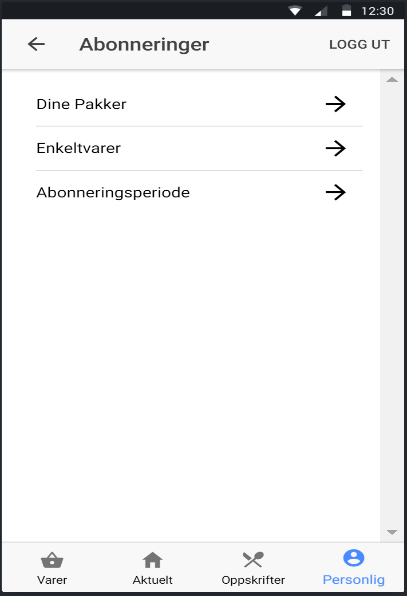
*UserInfoPage*

**

*AbonneringerPage*

Her vil det være en oversikt over alle abonneringer. Det kan være abonnerte pakker eller enkelt varer. Pakkeløsningen rakk vi ikke å fullføre, men prinsippene vil være helt like da Kolonial.no har pakketilbud på varer. Nederste alternativet navigerer bruker til ‘AbonneringsPeriodePage’ om gir detaljer om avtale perioden som ble registrert.

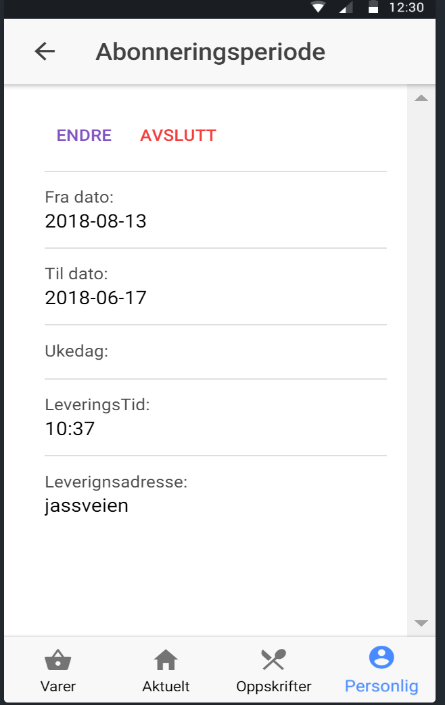
AboneringerPage

**

*AbonneringsPeriodePage*

Her kan bruker få en enkel oversikt over hva personen har preferert som leveringstid, sted og ukedag. Det er foreløpig ikke mulig å endre på abonnementet av tidsmessige årsaker, men knappen er der likevel for å illustrere mulighetene, men abonnementet er mulig å avslutte som derpå vil vise frem en passende melding om at bruker har avsluttet. Knappen har fått en rød farge som fordi dette vil tydeliggjøre at personen gjør en stor avgjørelse. All data er lagret i firebase med brukerens id og email knyttet til dem.

*AbonneringsPeriodePage*

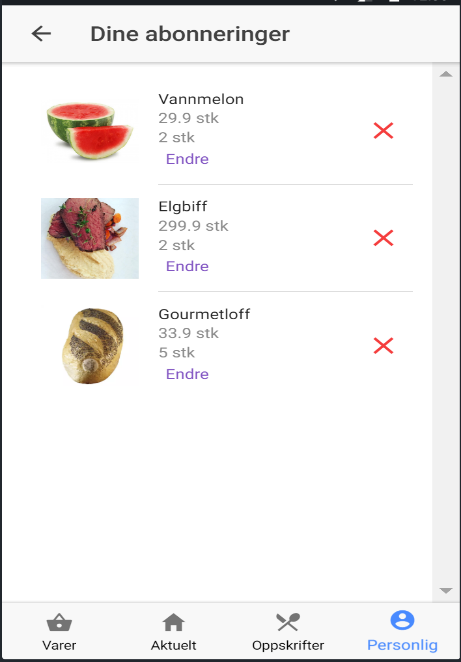
**

*DineEnkeltvarerPage*

Denne siden viser frem enkeltvarer som kunden har valgt å abonnere på. Bruker kan navigere hit enten ved å trykke på ‘Enkeltvarer’ under ‘Abonneringer’ eller ved å trykke på kalender ikonet  under liste over varer kunden handler fra.

Hver vare viser frem enhetsprisen og antall kunden har valgt å abonnere på å få levert for hver levering. Bruker kan velge å slette en vare ved trykke på eller endre antall personen vil abonnere på ved å trykke på endre. Ved sletting av vare vil bruker få en melding som varsler om hvilken vare som ble slettet.

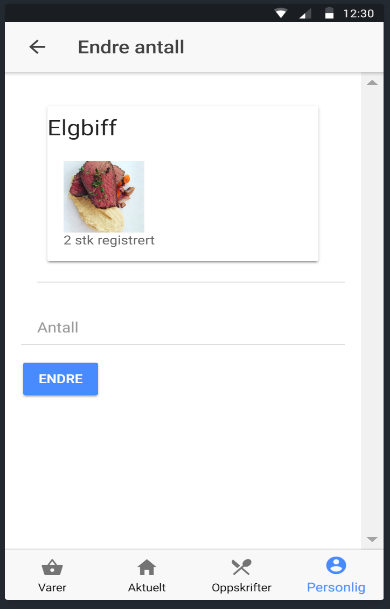
EnkeltVarePage



*EndreAntallPage*

Når bruker abonnerer på flere varer er det da naturlig at personen må kunne endre antallet *han/hun vil abonnere på en enkel måte uten å måtte slette hele varen for deretter å handle på nytt. Dette kan bruker gjøre som nevnt tidligere ved å klikke på endre knappen som også er farget med lilla. Fargevalget har blitt tatt for å holde oss til en bestemt farge knyttet til funksjonene abonnering har. Dette vil gjøre det lettere for bruker å kunne huske hva de forskjellige ikonene gjør i applikasjonen og skille det fra alle andre.*

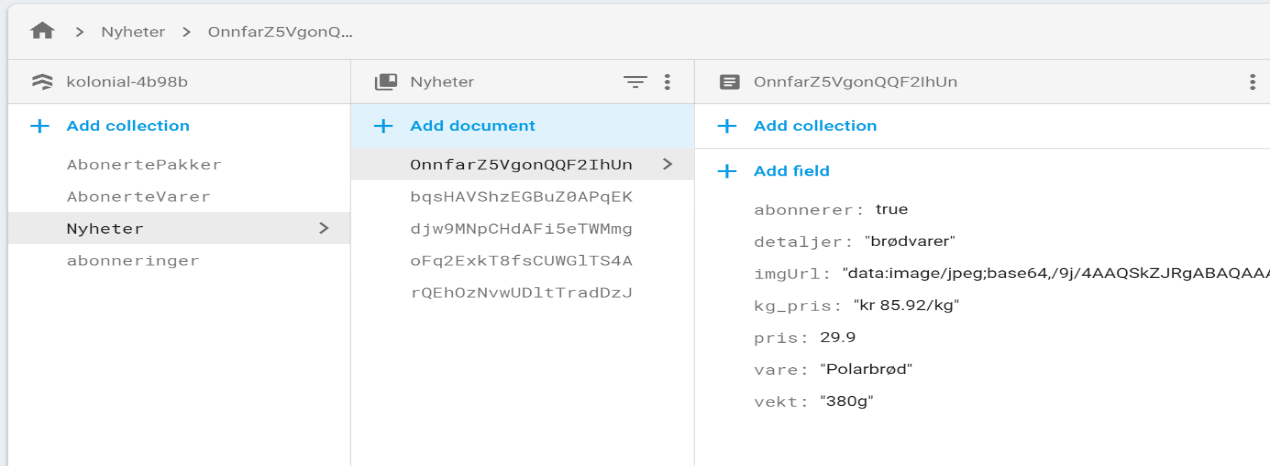
*EndreAntallPage*

**

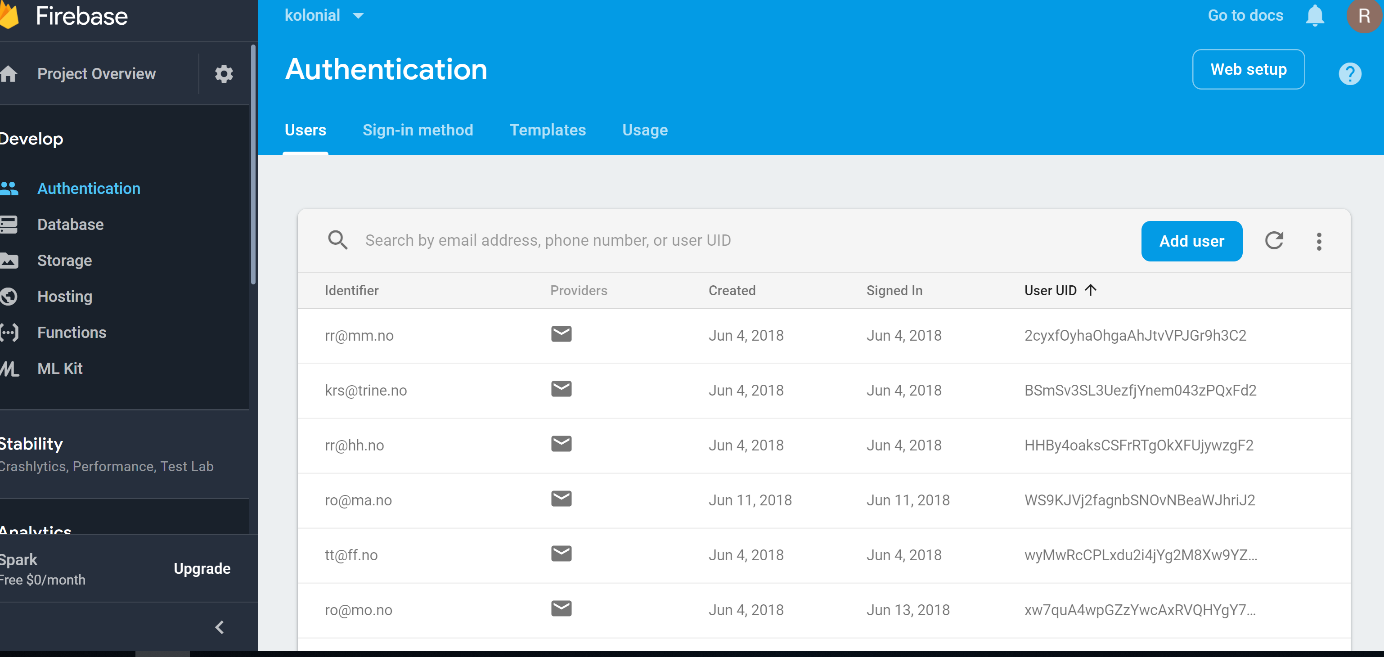
*Databasevalg*

***Firebase***: Tilhører google og er en NoSql-database med kolleksjon, dokument og kolonne databaser i motsetning til SQL databaser som baseres på tabeller med kolonner, rader og relasjoner. Vi har valgt dette som database fordi Firebase bl.a tilbyr 5 gb gratis lagring å bli overført innen en måned og har en egen autentiseringssystem og backend. En kan ha 50 aktive brukere på en gang og lagre 100mb av sin egen data (som documenter, f.eks ‘pris’, varenavn, ‘detaljer’) og overfører data asynkront i sanntid som betyr at brukers endringer og valg vil endre seg samtidig i firebase som hos bruker.

Firebase database



Firebase autentisering



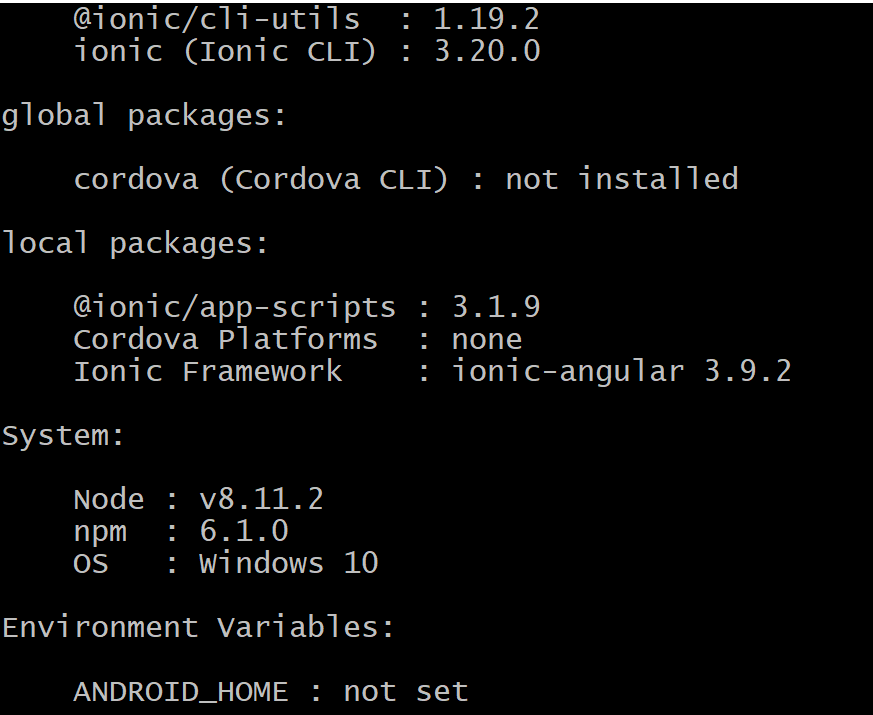
Bruksanvisning

For å kjøre applikasjonen er det forbeholdt at en har installert følgende på forhånd:

-Node verson 8.11.2 (<https://nodejs.org/en/download/> velg 32/64 bit avhengig av din maskin).

-Installer git bash (<https://gitforwindows.org/>) for å kunne installere rammeverket Ionic og Cordova som en trenger for å kunne bruke Ionics Cli. Med cli-commandoer i git-bash kan en installere npm modulene applikasjonen er bygget opp av.

-Etter at git bash er installert åpne commando linjeverktøyet og tast ‘npm install -g ionic cordova’.

Deretter verifiser at alt er installert ved å taste ‘ionic info’:

-Du kan også sjekke individuell verson ved å f.eks taste ‘ionic -v’ eller ‘cordova -v’ eller ‘npm -v’

-Deretter naviger til prosjektet med cd etterfulgt med <../../prosjekt lokasjon>

-Når du er inne på prosjektet installerer npm moduler ved å taste ‘npm install’

Her har jeg erfart gjennom studie-semesteret at en kan møte mange forskjellige komplikasjoner som hindrer en i å få installert bl.a Angularfirebase modulene. Veldig lite svar å få på internett og har vært et kjent problem siden 2016 der problemene er individuelle fra maskin til maskin. Selv tok det oss nesten hele 3 uker å få på plass. Det ble testet på windows , mac og linux og samme problem viste seg å være på alle. Av den grunn så laster jeg opp node\_modules mappen i dropbox LENKE(………………….) som en må hente og plassere inn i samme lokasjon som plugins , platforms og resources ligger i applikasjonen.

Hvis ikke dette løser seg så står de mest kjente problemene en møter på i en av disse lenkene:

--<https://blog.risingstack.com/node-js-windows-10-tutorial/>

--<https://stackoverflow.com/questions/49963223/property-automaticdatacollectionenabled-is-missing-in-type-firebaseapp>

-- https://github.com/angular/angularfire2/issues/1338

-- https://github.com/angular/angularfire2/pull/1572

-Når alt er på plass kjør så applikasjonen ved å taste ‘ionic serve -l’ eller ‘ionic cordova emulate Android for å kjøre det på en emulator (Krever at en må ha en emulator installert på maskinen).

Sluttresultat

På bakgrunn av at avgjørelsen vi tok for å lage en mobil-applikasjon var litt sen følte vi ikke at vi hadde tid til å få frem en prototype for mobil-applikasjonen, men vi hadde en visjon av hvordan den skulle se ut og vi føler vi har fått til hovedfunksjonene selv om detaljene ikke er der.

konklusjon/veien videre

*Applikasjonens funksjoner og kodemessig*

Da ingen av oss hadde særlig javascript / typescript kunnskaper til å begynne med var det veldig tidskrevende å få funksjoner til å fungere og vi rakk av den ikke å rett opp i feil. Derfor er det visse funksjoner som mangler som burde vært på plass og blant de funksjonene som mangler er :

-Å sperre knappen ‘Abonner’ fra å kunne legge til en vare hvis personen ikke har opprettet et abonnement. Den bør heller sprette opp en melding og varsle om at brukeren må opprette et abonnement først.

-Kalender ikonet som viser bruker hvilken vare som abonneres på fjernes ikke når bruker sletter en vare fra abonnerte enkeltvarer.

-Bruker får ikke totalpris oversikt på de ulike sidene ved abonnerte varer og varer som er til salgs.

Dette er blant de viktigste tingene vi ville prioritert først med applikasjonen. Ellers er trenger kodedelen også en opprydding. Har ikke brukt providere bl.a og klassenavnene burde vært endret til mere passende forhold. Lot være å endre fordi det var en del ukjente endringer en måtte ta hensyn til.

Kilder (mobilapplikasjon)

***Kode***

All kode som er skrevet i applikasjonen er skrevet ved å følge eksempler som er gitt i skoleundervisningene i kryssplattform.

Ellers er Ionic componenter som ikoner og date-time componenter hentet fra ionics dokumentasjoner.

Ikoner: <https://ionicframework.com/docs/ionicons/>

komponenter: https://ionicframework.com/docs/api/components/datetime/DateTime/

***Bilder***

*Fersken img:* [*https://cdn1.medicalnewstoday.com/content/images/articles/274/274620/two-peaches.jpg*](https://cdn1.medicalnewstoday.com/content/images/articles/274/274620/two-peaches.jpg)

*Vannmelon:*[*https://www.supermarket.no/media/catalog/product/cache/2/image/1200x1200/9df78eab33525d08d6e5fb8d27136e95/v/a/vannmelon.jpg*](https://www.supermarket.no/media/catalog/product/cache/2/image/1200x1200/9df78eab33525d08d6e5fb8d27136e95/v/a/vannmelon.jpg)

*Polarbrød:* [*http://www.instapuk.com/tag/rema1000*](http://www.instapuk.com/tag/rema1000)

*Smør:* [*https://www.arla.dk/globalassets/arla-dk/products/all-our-brands/arla-harmonie/harmonie-oeko-smoer-200g.png?preset=product-desktop*](https://www.arla.dk/globalassets/arla-dk/products/all-our-brands/arla-harmonie/harmonie-oeko-smoer-200g.png?preset=product-desktop)

*Elgbiff:* *https://scontent-atl3-1.cdninstagram.com/vp/812a503efedec14d4b55b633119e881e/5B8DAE71/t51.2885-15/e35/32212381\_199622584010725\_6376923568562765824\_n.jpg*

Husk drop-box /git\_hub moduler