



postnord
LOGISTICS

JOAKIM RISHAUG

SONDRE SPARBY BOGE

MARTIN HAGEN

LARS-ERIK KASIN

HOVEDPROSJEKT - GRUPPE 33

FORORD

Denne rapporten er en presentasjon av arbeidet med hovedprosjekt ved Høgskolen i Oslo og Akershus, våren 2015. Prosjektet er gjennomført av fire studenter ved ingeniørfag - data, og gikk ut på å utvikle en android-applikasjon i samarbeid med IT-konsulentenselskapet Mesan, for en av deres kunder, logistikkonsernet PostNord.

Applikasjonen skulle være en proof of concept-løsning, og skulle ikke utgis direkte, men være et forslag til utforming som viser hvilke muligheter som finnes. Konkret skulle applikasjonen være en forbedring av en eksisterende løsning, kalt håndterminalen, som er et verktøy for PostNords sjåfører i arbeidshverdagen. Sjåførene bruker blant annet håndterminalen til innscanning av pakker, oversikt over dagens arbeidsoppgaver og levering av pakker med innhenting av signatur fra kunde.

Dokumentet er hovedsakelig rettet mot sensor og arbeidsgiver, og er inndelt i innledning, prosessdokumentasjon, produktbeskrivelse, brukerveiledning og avslutning. De ulike delene beskriver henholdsvis involverte aktører og oppgaven, arbeidsprosessen og avgjørelser underveis, det endelige produktet, enkel brukerveiledning og en evaluering av prosjektperioden. Det forutsettes lite kunnskap for å lese rapporten, men den inneholder en del tekniske elementer innen androidutvikling som krever noe programmeringsforståelse. Skulle noe være uklart finnes også en ordliste med beskrivelse av tekniske uttrykk sist i dokumentet.

Vi vil rette en stor takk til vår arbeidsgiver, Mesan, som tok oss seriøst fra første stund, og ga oss en spennende oppgave og god veiledning underveis. I tillegg vil vi takke vår veileder, Geir Skjevling, som hjalp oss med å komme i gang med prosjektet, og var svært behjelpeelig når vi hadde usikkerheter rundt dokumentasjon og prosessen overfor skolen.

INNHOLD

1. Innledning

- 1.1 Prosjektgruppen
- 1.2 Arbeidsgiver
- 1.3 Kunde
- 1.4 Problemstilling
- 1.5 Rammebetingelser
- 1.6 Mål
 - 1.6.1 Resultatmål
 - 1.6.2 Effektmål
 - 1.6.3 Læringsmål

1.7 Kravspesifikasjon

- 1.7.1 Arbeidsgivers Kravspesifikasjon
- 1.7.2 Funkjonelle Krav
- 1.7.3 Ikke-funksjonelle Krav
- 1.7.4 Tekniske Krav

2. Prosess

- 2.1 Bakgrunn for prosjekt
- 2.2 Smidig metodikk
- 2.3 Arbeidsmetode
 - 2.3.1 Parprogrammering
- 2.4 Valg av verktøy
 - 2.4.1 Jira
 - 2.4.2 Git
 - 2.4.3 Confluence
 - 2.4.4 Android studio
 - 2.4.5 Genymotion
 - 2.4.6 Photoshop
- 2.5 Prosjektstyring
 - 2.5.1 Scrum
 - 2.5.1.1 Planning poker
 - 2.5.1.2 User Story
 - 2.5.1.3 Standup-møter
 - 2.5.1.4 Scrum-master
 - 2.5.2 Git
 - 2.5.3 Jira
- 2.6 Prosjektstyringsdokumenter

- [2.6.1 Risikomatrise](#)
- [2.6.2 Arbeidsplan](#)
- [2.6.3 Milepælsplan](#)
- [2.6.4 Fremdriftplan](#)

- [2.6.4.1 Opprinnelig estimat](#)
 - [2.6.4.2 Oppdatert estimat](#)

- [2.6.5 Sprinter](#)

- [2.6.5.1 Pre-sprint](#)
 - [2.6.5.2 Arbeidsliste](#)
 - [2.6.5.3 Innscanning](#)
 - [2.6.5.4 Ruteliste & Signatur](#)
 - [2.6.5.5 Buffer - CatchUp](#)
 - [2.6.5.6 Misoptimalisering](#)
 - [2.6.5.7 Wrap-up app](#)
 - [2.6.5.8 Wrap-up dokumentasjon](#)

- [2.6.6 Dagbok](#)

- [2.7 Møter](#)

- [2.7.1 Planleggingsmøte med Harald](#)
 - [2.7.2 Møte med Postnord-teamet](#)
 - [2.7.3 Mockuptegning](#)
 - [2.7.4 Presentasjon for Postnord](#)

- [2.8 Testing](#)

- [2.8.1 Akseptansetester](#)
 - [2.9 Utfordringer og avgjørelser](#)
 - [2.10 Presentasjon på Mesan-møte](#)
 - [2.11 Kodestandard](#)
 - [2.11.1 Kodestruktur](#)
 - [2.11.2 JavaDoc](#)
 - [2.11.3 Refaktorering](#)

- [3. Produkt](#)

- [3.1 Målgruppe](#)
 - [3.2 Teknologivalg](#)
 - [3.2.1 Android](#)
 - [3.2.2 Android Lollipop](#)
 - [3.2.3 Signatur](#)
 - [3.2.4 Maps/directions](#)
 - [3.2.5 Zbar](#)
 - [3.2.6 SQL / Sugar ORM](#)

- [3.2.7 AndroidAnnotations](#)
- [3.2.8 AndroidMP](#)
- [3.2.9 Gradle](#)
- [3.3 Samsvar mellom kravspesifikasjon og endelig produkt](#)
- [3.4 Designvalg](#)
 - [3.4.1 Android standarder](#)
 - [3.4.2 Farger](#)
 - [3.4.3 Navigasjon](#)
 - [3.4.4 Floating Action Button](#)
 - [3.4.5 Google Maps](#)
- [3.5 Backend](#)
- [3.6 Ferdig løsning - Skjermbilder med hovedoppgaver](#)
 - [3.6.1 Login](#)
 - [3.6.2 Ruteliste](#)
 - [3.6.3 Scanning](#)
 - [3.6.4 Arbeidsliste](#)
 - [3.6.4.1 Pakkecard](#)
 - [3.6.5 Levering](#)
 - [3.6.5.1 Avvik](#)
 - [3.6.6 NavDrawer](#)
 - [3.6.6.1 Profilvindu](#)
 - [3.6.6.2 Pakkeoverføring](#)
- [3.7 Flytdiagram](#)
- [4. Brukerveiledning](#)
- [5. Avslutning](#)
 - [5.1 Målloppnåelse](#)
 - [5.1.1 Resultatmål](#)
 - [5.1.2 Effektmål](#)
 - [5.1.3 Læringsmål](#)
 - [5.2 Oppfyllelse av krav](#)
 - [5.2.1 Funksjonelle krav](#)
 - [5.2.2 Ikke-funksjonelle krav](#)
 - [5.3 Videreutvikling](#)
 - [5.4 Vurdering av prosjektperioden](#)
 - [5.5 Attest fra arbeidsgiver](#)
 - [5.6 Konklusjon](#)
- [6. Vedlegg](#)
 - [6.1 Sprintlogs / burndown](#)

[6.2 User stories](#)

[6.2.1 User storis før prosjekt start:](#)

[6.2.2 User stories etter prosjekt slutt:](#)

[6.3 Akseptansetester](#)

[6.4 Dagbok](#)

[6.5 Kravspek fra Mesan](#)

[6.6 Komplette møtereferater](#)

[6.7 User stories](#)

[6.8 Transition forklaring](#)

[6.9 Mockuptegninger](#)

[6.10 Kodeanalyse](#)

[6.11 Teknisk ordliste](#)

[6.12 Kilder](#)

1. INNLEDNING

1.1 PROSJEKTGRUPPEN

Gruppen består av Joakim Rishaug, Lars-Erik Kasin, Martin Hagen og Sondre Sparby Boge. Alle fire studerer 'ingeniørfag - data' ved HiOA, og vi ble kjent de første dagene på studiet. Vi ble tidlig en slags kollokviegruppe, og har deretter slått oss sammen på alle prosjekter hvor vi hadde mulighet disse tre årene. Vi jobber godt sammen og har like mål for studiene, og det var derfor naturlig å også jobbe sammen på bacheloroppgaven. Ikke minst var alle veldig interessert i mobilapplikasjoner, noe som gjorde at vi alle ble motivert til å legge inn mye arbeid i oppgaven.

1.2 ARBEIDSGIVER

Arbeidsgiver er Mesan, et konsulentselskap som spesialiserer seg på å lage skreddersydde IT-løsninger. Mesan ble grunnlagt i år 2000, av en gruppe erfarte mennesker med lang erfaring innen konsulentbransjen. De har alltid hatt et spesielt fokus på å levere produkter av god kvalitet. Mesan har flere profilerte kunder som blant annet Møllergruppen, Norgesgruppen, NRK, og Norsk Forskningsråd. I Norge har Mesan rundt 100 fast ansatte som leverer produkter og tjenester til sine kunder. I de siste årene har Mesan gjort det særlig bra økonomisk og ansetter stadig flere konsulenter.

Ansvarlig for hovedprosjektgruppen, og viktigste kontaktperson i Mesan har vært konsulentsjef, Harald A. Kulø, men vi har også hatt mye kontakt med direktør for teknologitjenester, Christoffer Daae-Qvale. Disse har veiledet oss gjennom prosessen og sørget for et godt grunnlag og rammeverk i utviklingen. Harald har i tillegg representert kunden underveis, da PostNord ikke var direkte involvert i utviklingen. Mer om dette under punkt 1.3, Kunde.

I tillegg hadde vi en teknisk veileder i Anders Ullnæs, som hadde gode kunnskaper innen utvikling for mobile platformer, og var til god hjelp når vi møtte på ekstra store utfordringer.

1.3 KUNDE

Kunden i vårt prosjekt er logistikkonserten PostNord. PostNord er en kunde av Mesan som driver med transport og logistikk. Postnord er en aktør med aktiviteter i hele Norden, med ca 1800 ansatte i Norge, hvorav 800 er sjåfører. Et selskap av denne størrelsen har naturligvis store konkurrenter som Posten/Bring og DB Schenker.

Distribusjonstjenesten til Postnord omfatter alle aspekter ved frakt av gods, fra sender til mottaker. Det er flere ledd i kjeden som kontinuerlig skal overvåkes for å sikre god kvalitet på leveransene, og det kreves et godt system i bakkant for å kunne leve opp til forventningene. Appen som utvikles i dette prosjektet vil hovedsakelig gjelde for sjåfører i PostNords lokaltransportavdeling, som kjører små til mellomstore pakker rundt om i Norge.

PostNord var ikke direkte involvert i utviklingen eller utformingen av løsningen, da dette skulle være en slags "overraskelse" for dem. De visste heller ikke at vi jobbet med det før prosjektet nærmet seg slutten, og visste ingenting om resultatet før presentasjonen 20. mai. Dette var for å fremme følelsen av at dette var noe Mesan foreslo for dem på en overbevisende måte, uten at de selv trengte å bruke tid på utviklingen, før avgjørelsen om å satse på det var tatt.

1.4 PROBLEMSTILLING

Oppgaven vi fikk hos Mesan, var å lage et proof of concept for deres kunde, PostNord.

Helt konkret skulle oppgaven omfatte det håndholdte systemet sjåførene bruker ved utlevering av pakker, kalt håndterminalen. Det eksisterende systemet er bransjestandarden, men har definitivt stort potensiale for forbedring. De håndholdte terminalene sjåførene bruker koster opp mot 15 000 kroner per stykk, og eneste fortrinn vi kan finne for disse, over for eksempel Android-nettbrett, er hardførhet. Håndterminalene kjører på Windows Mobile 6.5-plattformen, og bærer sterkt preg av sine konkurrenter på tiden, i dag ansett som utdaterte systemer, slik som Symbian.

Dette fører til dårlige muligheter for hurtig iterasjon og videreutvikling. Ikke minst lider sjåførene under en forferdelig brukervennlighet. Dette er som regel heller ikke de mest teknologivante folkene, og vil mest sannsynlig kunne effektivisere sitt arbeid ved å få en

oppdatert user experience. Kunden er for tiden i et større moderniseringsprosjekt, og vil dermed mest sannsynlig ha behov for en mer fleksibel løsning på sjåførsiden.

Mesans mål med dette prosjektet er å demonstrere for kunden hvordan en oppgradering av denne plattformen vil gi stor avkastning i det lange løp. Formfaktoren det siktes på er en slags tablet eller phablet. Dermed vil PostNords teknologivalg være godt dekket av det vanlige consumer-markedet, og stykkprisen vil havne i en mer rimelig 3000-4000-kronersklasse. Trenden med tablets ser ikke ut til å sakke ned, og det ser ut som at Android vil være ledende på markedet en god stund fremover. Videre vil det ved å designe en ny løsning kunne brukes best practices, som ikke ser ut til å ha blitt tatt hensyn til i den gamle løsningen.

Det er med andre ord et viktig profileringsprosjekt for Mesan for å bryte inn i et relativt lukket marked som er preget av gamle og arkaiske produkter.

Gruppen har fått stor frihet til å velge teknologi og tenke innovativt for å best mulig addressere problemene ovenfor. Det er dog å foretrekke at vi fokuserer på "wow-faktor", da dette er et proof of concept.

1.5 RAMMEBETINGELSER

Siden oppgaven er en proof of concept-løsning har vi ikke fått mange funksjonelle krav fra oppdragsgiver. Det viktigste ved denne oppgaven var å forbedre nyttig funksjonalitet fra den eksisterende løsningen, samt utvide med funksjonalitet vi mener vil forenkle sjåførene sine arbeidsdager. Uten om dette har vi i samarbeid med oppdragsgiver avgrenset oppgaven noe med tanke på tiden vi har tilgjengelig.

SYSTEMET SKAL:

- Inneholde all nødvendig funksjonalitet fra den eksisterende løsningen
- Ha en fremvisning av ruten til sjåføren
- Bedre opplevelse av systemet for brukeren
- Tilby flere hjelpeemidler til brukerne i løsningen

SYSTEMET BØR:

- Være intuitivt for nye brukere

- Være effektivt for både gamle og nye brukere
- Bruke versjonskontroll under utvikling

EKSTRA:

- Hvis produktet blir bra nok kan vi presentere for Postnord.
- Sjokkere/Overgå forventinger

Tidsrammen for prosjektet, fra første dag til siste innlevering av rapport, er fra 05.01.2015 - 26.05.2015.

1.6 MÅL

1.6.1 RESULTATMÅL

Målet med prosjektet er å utvikle et proof of concept for en ny løsning for PostNord-sjåførenes håndterminal, i form av en android app til tablet. Produktet vi lager, bør være overbevisende nok til at PostNord ønsker å sette det ut i produksjon videre.

For å oppnå dette, må vi lage en løsning som i det store og hele gjør hverdagen til sjåførene mest mulig effektiv. Appen bør da være enkel og oversiktlig å bruke. Den må være effektiv, med færrest mulig klikk for å utføre oppgaver som gjøres ofte. Den bør ha minst mulig rom for feil, og gode muligheter for retting av feil som eventuelt måtte forekomme. Samtidig bør den dekke alle kritiske behov sjåføren har i arbeidsdagen, og i tillegg tilby best mulige ekstrafunksjoner, som sjåføren kan dra nytte av i arbeidsdagen. Krav til sikkerhet og gode backend-systemer vil ikke være prioritert i proof of concept-løsningen, men vil så klart være viktig for videre utvikling om produktet skal ut i markedet.

Konkret vil dette bestå av følgende punkter:

1. Enkel og oversiktlig innlogging på sjåførens bruker.
2. Oversiktlig valg av rute, hvor vi også inkluderer rutedetaljer og en visuell fremstilling av ruten på kart.

3. Enkel og effektiv registrering av pakker i form av innscanning av QR- og strekkoder ved hjelp av tabletens eget kamera.
4. God oversikt over dagens rute, med en optimalisert rute med detaljer om nåværende fremdrift, samt gjenstående oppgaver. I tillegg til estimering av distanse og tidsbruk for disse.
5. Enkel og sikker levering/henting av pakker med rom for håndtering av avvik fra normal fremgangsmåte. Dette ved hjelp av utscanning av pakker, signering direkte på tabletten, og kontroll av informasjon hos mottaker.
6. Gode hjelpefunksjoner, tilgang til god informasjon og enkel navigering i appen, samt effektiv håndtering av avvik og nye hentinger.

1.6.2 EFFEKTMÅL

Effekten av prosjektet for arbeidsgiver, Mesan, bør være å kunne åpne opp en ny arena, i form av sjåførenes verktøy i arbeidsdagen, hos en av deres større kunder. Dette vil åpne for flere oppdrag hos kunden fremover, både innen utvikling og vedlikehold, og dermed også større omsetning for Mesan. Prosjektet fungerer også til en viss grad som rekrutteringsprosess for Mesan, om studentene gjør en god jobb.

For brukerne av appen, PostNords sjåfører, er effektmålet at arbeidsdagen skal være lettere. Dette ved at de har bedre oversikt over dagens gjøremål, mer effektiv innscanning, utlevering og rapportering underveis. Ved å optimalisere sjåførens rute ved hjelp av estimering, men samtidig la sjåføren bruke sin kunnskap til å justere ruten etter eget ønske, er det også et mål at selve kjøringen fra adresse til adresse, skal effektiviseres.

Raskere levering, en enklere hverdag for sjåførene, og mindre rom for feil, er noe som vil bedre PostNords resultater i form av raskere og mer pålitelige leveringer, og samtidig gi mer fornøyde ansatte og kunder. Dette gir videre økt omsetning også for PostNord.

1.6.3 LÆRINGSMÅL

Målene for læring i prosjektet er mange, men først og fremst ønsket vi å få erfaring fra en mest mulig reell arbeidssituasjon, som tilsvarer hverdagen vår når vi går ut i arbeid. Dette gjelder alt fra planleggingsprosess, disponering av tid, fremgangsmåte for utvikling av app, prosjektstyring og dokumentasjon.

Mer konkret ønsket vi å lære mer om Androidutvikling, og særlig da gode standarder for design og utforming, i tråd med Androids egne retningslinjer. I tillegg er vi tidlig ute med å ta i bruk Androids nyeste operativsystem, Lollipop (Android 5.0), som også kan gi oss et fortrinn i arbeidslivet fremover.

Vi hadde også et mål om å få erfaring med detaljert prosjektplanlegging og smidig metodikk, ved hjelp av scrum. Dette er noe vi har noe erfaring med fra skolen, men ikke på en "reell" skala, som her. I tillegg til dokumentasjon av fremdrift, og jevnlige møter for justeringer underveis.

1.7 KRAVSPESIFIKASJON

1.7.1 ARBEIDSGIVERS KRAVSPESIFIKASJON

Da vi begynte på prosjektet, fikk vi et utkast til kravspesifikasjon av arbeidsgiver. Dette inneholdt ønsker fra kunden, og forslag til funksjoner vi kunne implementere i vår løsning. Det var likevel opp til oss å justere denne utifra rammene for vårt prosjekt, hva vi så på som gode/dårlige løsninger og hvilke funksjoner som var nødvendige eller kunne uteslippes. Dette dokumentet finnes i sin helhet under vedlegg, punkt 6.5.

1.7.2 FUNKSJONELLE KRAV

Vi utviklet vår egen kravspesifikasjon på grunnlag av utkastet vi fikk av arbeidsgiver. Hele kravspesifikasjonen finnes under vedlegg, punkt 6.5. Denne ble et veiledende dokument i prosessen, men flere av punktene ble endret underveis, i tråd med vår smidige utviklingsmetodikk. Her er hovedpunktene for funksjonelle krav i den endelige løsningen:

Appen skal inneholde følgende funksjoner:

- Enkel innlogging og valg av rute
- Scanning av strekkode/QR ved hjelp av kamera
- Optimalisering av valgt rute (korteste vei), ved hjelp av google directions API
- Mulighet for å endre rutens forløp etter sjåførens eget behov underveis
- Visuell fremstilling av rute på kart, og visning av fremdrift underveis.
- Varslingssystem for å motta forespørsel om ad-hoc-henting/nødssituasjoner
- Sending sms til mottaker om at pakke er på vei (geofencing)
- Mulighet for å ringe kjøreleder/motta samtale direkte i appen (handsfree)
- Mulighet for å motta signatur fra mottaker på tabletten
- Mulighet for å melde avvik ved feil eller uforutsette hendelser
- Mulighet for å overlevere en pakke til en annen sjåfør
- Visning av statistikk og annen informasjon om innlogget sjåfør

1.7.3 IKKE-FUNKSJONELLE KRAV

Ettersom dette er et proof of concept, har vi ikke like mange ikke-funksjonelle krav, ettersom det kreves at appen er videre stabil, sikker, skalerbar eller lett å vedlikeholde før den settes i produksjon. Selvfølgelig er dette noe vi gjerne ønsker å holde på også i dette prosjektet, men siden appen hovedsakelig skal gjøre seg bra på en presentasjon med egenprodusert dummy-data, så er det stort sett funksjonalitet som står i fokus. Likevel har vi noen krav i forhold til ytelse og gui. Vi krever derfor at appen skal ha følgende:

- Brukervennlig navigasjon etter androids designprinsipper
- Kort vei til viktige funksjoner
- Forutsigbarhet og konsistent oppførsel fra appen utifra brukers input
- Kort responstid og en behagelig brukeropplevelse

1.7.4 TEKNISKE KRAV

Siden dette er en proof of concept-løsning har det heller ikke vært stilt mange tekniske krav til oppgaven. For selv å holde kontroll underveis satt vi likevel opp noen punkter til hvordan vi mente oppgaven skulle løses.

- Oppgaven skal kodes i Java og XML, altså for Android
- Git skal brukes til versjonskontroll
- Android Studios skal brukes for utvikling

- Google sine API's skal benyttes der det er mulig for kart osv.
- JIRA skal brukes for å holde orden på sprintene og arbeidet
- Dokumentasjon underveis skal foregå på Confluence

2. PROSESS

2.1 BAKGRUNN FOR PROSJEKT

Vi begynte tidlig letingen etter et bachelorprosjekt, og allerede i slutten av august hadde vi møter med flere aktører innenfor konsulentbransjen. Dette var naturlig nok de som har høyest eksponering i feltet, og ingen av oss kjente til en konkret bedrift som trengte en løsning implementert. Etter møter med diverse aktører, som Bekk og KnowIt, Endte vi til slutt opp hos Mesan, hvor vi mååt lage en hurtig implementasjon av en restaurant-app som bevis på at vi kunne kode. Denne ble dermed vurdert opp mot andre kandidater hos Mesan, og vi sto igjen som de mest kvalifiserte.

Vi sa tidlig ifra at vi helst ønsket å lage en Android-applikasjon, men at vi var åpne for å jobbe en del med backend og andre systemer i tillegg om dette skulle vise seg nødvendig. Mesan var også klare på at de så på dette som en del av en rekrutteringsprosess i selskapet, og at de aldri hadde hatt en gruppe hos seg uten å ansette noen.

Mesan jobber for tiden med å oppgradere mye av kjernestrukturen i logistikkdelen i Postnord. Mye av planleggingen for utkjøring av pakker foregår i dag på post-it-lapper. Målet til Mesan er å få Postnord bort fra mye av dette manuelle arbeidet, og heller automatisere disse prosessene. Oppgradering av håndterminalen til Postnord er i utgangspunktet ikke en del av denne moderniseringen. I den forbindelse ønsker Mesan å vise Postnord et proof of concept for hvor mye enklere det vil være å forholde seg til en håndterminalløsning på et Android-nettbrett enn den løsningen som brukes i dag.

2.2 SMIDIG METODIKK

En viktig del av ethvert prosjekt er hvordan man skal arbeide. Hvilken prosess som brukes for å nå et mål kan ha veldig stort utfall, og kan i noen tilfeller avgjøre om prosjektet feiler eller lykkes. En av de mest populære metodene innen prosjektutvikling er smidig metodikk. Smidig utvikling passer ypperlig i mindre grupper på rundt fire personer, der kravspesifikasjonen kan endres ganske drastisk. I motsetning til typisk fossefallsutvikling, kan smidig metodikk

håndtere uventede situasjoner på kort tid med minimal ressurskostnader. Faktisk er det nesten forventet at prosjektet endres i løpet av prosjektperioden. En av de mest populære og kjente smidige arbeidsmetode er Scrum. Vi valgte Scrum siden vi allerede er kjent med dette fra skolen, og arbeidsgiver også foreslo dette.

2.3 ARBEIDSMETODE

Gruppen jobber med en mer eller mindre flat struktur, hvor vi fordeler oppgavene jevnt mellom medlemmene. Det eneste unntaket kan sies å være Joakim, som har fungert som scrum-master i prosjektperioden. Vi har fått kontorplass hos Mesan i Henrik Ibsens gate, og har stort sett jobbet derfra i hele prosjektperioden. Vi bestemte oss tidlig for å sette av tre fulle arbeidsdager i uka for å jobbe med oppgaven, og valgte å bruke mandag, onsdag og torsdag til dette. Dette er et opplegg vi har fulgt gjennom hele prosjektet, med unntak av uker med helligdager eller andre grunner til å flytte eller avlyse arbeidsdager.

Som sagt var det ingen utpreget leder av gruppen, og utifra sprintene vi satt opp for hver uke, valgte gruppemedlemmene mer eller mindre selv hva de ønsket å jobbe med. Dette ble gjort fra dag til dag, på stand up-møter på morgenen. Om noen oppgaver så ut til å bli for vanskelige eller ta for lang tid å bli ferdig med, hjalp vi så klart hverandre med å komme i mål.

I tillegg hadde vi en representant fra Mesan, Anders Ullnæs, som fulgte opp gruppa og var tilgjengelig om vi skulle trenge noe. Anders har gode kunnskaper innen utvikling for mobile plattformer, og var til god hjelp når vi møtte på ekstra store utfordringer.

Denne arbeidsmetoden har fungert svært godt, og vi føler vi har hatt god flyt og effektivitet i arbeidet.

2.3.1 PARPROGRAMMERING

Mange av oppgavene har vært særdeles krevende utfordringer innen temaer ingen av oss har hatt noen tidligere erfaring med. Derfor måtte vi lære veldig mye nytt på veldig kort tid for å kunne lage en solid løsning innen den gitte tiden. For å gjøre dette mest mulig effektivt med minst mulig risiko med tanke på feil, valgte vi å sette oss i par for å løse mange av disse problemene. Dette fungerte stort sett veldig bra, for vi kunne fylle ut hverandres erfaringer og idéer, og komme med en best mulig løsning.

Kallet for å hente ruteinformasjon var i et tilfelle unødvendig komplekst og hindret oss å legge til nye funksjoner for kartet, defor trengte vi å fikse denne flyten. En av oss satte oss sammen med mentoren vår for prosjektet og ryddet opp i flyten. Uten om det jobbet vi en del i par når vi skulle implementere blant annet animasjoner, transitions og kart. I tillegg ga dette oss muligheten til å ikke være like utsatt om én person måtte være hjemme en dag. Det ville som oftest være flere personer i gruppen som hadde dyp kunnskap om kjernefunksjonene vi utviklet.

2.4 VALG AV VERKTØY

For prosjektstyring brukte vi flere av systemene Mesan selv bruker.

2.4.1 JIRA

Oversikt over user-stories, tasks og tidsbrukt. En standard i bransjen for akkurat disse oppgavene.

The screenshot shows the JIRA backlog interface. At the top, there are navigation links for 'Projects', 'Issues', 'Agile', and 'Create'. Below that is a search bar and a 'Recently Updated' filter. The main area is titled 'Backlog' and shows '6 issues'. Each issue is listed with a small icon, a priority level (e.g., P1, P2), a title, a brief description, and a status indicator (e.g., L, J, S, M). A 'Create Sprint' button is located at the top right of the backlog table. To the right of the backlog, there is a sidebar with a legend for 'Android Applikasjon' and labels 'L', 'J', 'S', and 'M' corresponding to different colors.

Figur 1 Backlog i JIRA

2.4.2 GIT

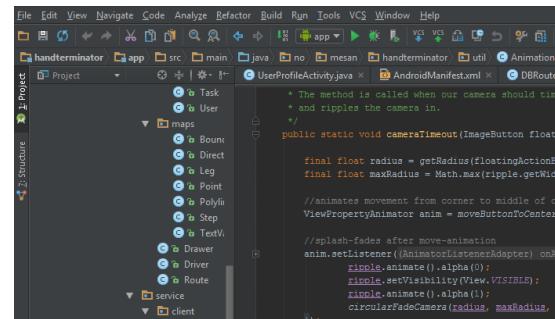
Versjonskontroll av kode. Gjør det mye enklere å slå kode sammen siden vi er fire stykker, og overlapping er noen ganger ikke til å unngå. Git er en standard i bransjen.

2.4.3 CONFLUENCE

Bruk som en wiki som vi selv fylte ut i løpet av arbeidsperioden. Vi brukte Confluence mest til å skrive prosjektdagbok, møtereferater og testrapporter siden dette var veldig enkelt å bruke.

2.4.4 ANDROID STUDIO

Utviklermijø som bygger på IntelliJ IDEA plattformen. Android Studio¹ er utviklet for å utvikle Android systemer. Systemet er enkelt å bruke og gir en rekke av muligheter for debugging, testing og påkobling av andre enheter. En stor fordel her kontra Eclipse, er en mye bedre auto-complete, og integrasjon med teknologi som Git. Figur 2 viser et utsnitt av Android Studios grensesnitt.



Figur 2Android Studio IDE

2.4.5 GENYMOTION

En solid Android-emulator² med støtte for blant annet rotasjon og dynamisk skalering av vindu-størrelse. Den er mye mer responsivt enn Android SDK's egen standard Android-emulator, og ble derfor et naturlig valg. Den baserer seg på Oracles VirtualBox, og blir derfor egentlig mer som en VM enn en emulator.

2.4.6 PHOTOSHOP

Et veldig populært redigeringsprogram vi brukte til bilder og ikoner. Ettersom vi ville lage visuelle elementer i pixel-presisjon var det viktig for oss å bruke noe som var enkelt og bra å bruke da vi skulle både lage, og endre på bilder.

¹ <http://developer.android.com/tools/studio/index.html>

² <https://www.genymotion.com>

2.5 PROSJEKTSTYRING

2.5.1 SCRUM

Siden gruppen har jobbet sammen så lenge ble det tidlig klart at vi burde ha en smidig arbeidsprosess. Scrum er åpenbart den mest populære, og vi bestemte oss derfor for å lære mer om dette ved å praktisere det. Bortsett fra hovedelementene nevnt under har vi stort sett jobbet som vi alltid har gjort, hvor hver enkelt stort sett selv tar ansvar. Den største utfordringen som kom med et prosjekt av denne størrelsen har naturlig nok vært arbeidsfordeling. Her har det vært scrum-master som har styrt hovedretningen og sørget for at alle har noe å gjøre til enhver tid.

Sprintene våre har vært variable i lengde, men stort sett holdt seg på 2 uker hver. Det har vært kortere sprinter, da som en slags pitstop for prosjektet hvor vi sørget for at alt var på stell, både i koden og i måloppnåelse.

2.5.1.1 PLANNING POKER

På slutten av startfasen i prosjektet satte gruppen seg ned for å estimere arbeidsmengden i forhold til kompleksitet. Dette ble gjort ved en såkalt planning poker. Her skal alle deltakerne velge et kompleksitetstall, i forhold til en oppgave alle har blitt enige om har verdi "1". I vårt tilfelle ble dette å lage en innloggingsskjerm. Alt annet arbeid ble derfor estimert i henhold til dette. Gruppen var på dette tidspunktet enige om at oppgaven med høyest kompleksitet var håndteringen av tidssensitive pakker i ruteoptimaliseringen for kartfunksjonaliteten. Her var det uenighet om hvor komplekst det faktisk var, men konklusjonen var alt fra 13 til "ukjent" kompleksitet. Det skulle senere vise seg at dette var en oppgave vi ikke fikk gjennomført.

2.5.1.2 USER STORY

Etter planning poker, gikk vi løs på user-stories for prosjektet. Våre user-stories så tidlig ut som "functionality stories", som vi kalte de. Dette fordi vi måtte bygge et system opp fra bunnen, og alle hele stories ville endt opp med å være en sprint i seg selv. Slik som for eksempel "ruteoptimalisering i kart". Dermed ble det mer hensiktsmessig for oss å bryte ned alt dette til konkrete ting vi måtte gjøre for å komme i mål. Alle user stories fra starten av prosjektet er lagt med som vedlegg under punkt 6.2.1.

2.5.1.3 STANDUP-MØTER

Vi har flittig brukt daglige standups som en god måte å holde styr på hovedmål for dagen. Det ble en fast rutine å samles ved kaffemaskinen for gå gjennom en sjekkliste med punkter, som "Hva gjorde du i går?", "Hva planlegger du å gjøre i dag?" og "Hva blir de største problemene?". Slik ble det altså lett å samkjøre gruppen, slik at alle dro i riktig retning.

2.5.1.4 SCRUM-MASTER

Vi har hatt en relativ demokratisk prosess gående, men Joakim har til en viss grad tatt rollen som Scrum-master. Scrum-master sørger for at de andre gruppemedlemmene kan fokusere mest mulig på arbeid, og ikke tenke på møteplanlegging eller hvordan prosjektet skal utspille seg. Dette har vært en rolle som har tatt litt tid her og der, men ingenting som har stoppet han fra å utføre utviklerarbeid i tillegg.

Som tidligere nevnt har den største arbeidsmengden her kommet fra fordeling av tasks og stories fra JIRA til medlemmene av gruppen.

2.5.2 GIT

Git er for tiden det mest populære versjonskontrollsystemet i programmeringsbransjen. Det har sett en stødig vekst som følge av populærheten til GitHub og sosial koding-fenomenet. Git er utviklet av Linus Thorvalds. Det kan være litt vanskelig å sette seg inn i konsollkommandoene man trenger i starten, men de blir etterhvert ganske oversiktlig å forholde seg til. Det er mange ting man kan gjøre med Git, men vi har hovedsaklig holdt oss til det grunnleggende, som git pull, push, commit og reset. Disse kommandoene er også integrert grafisk inn i Android Studio, som gjorde det lett for oss å bruke.

2.5.3 JIRA

Jira er et populært prosjektstyrings-system for oppfølging og oversikt av oppgaver og bugs i typiske programmeringsprosjekter. Direkte i JIRA finnes en "Agile"-modul som passer perfekt til vårt bruk, siden vi driver prosjektet med Scrum-metodikk. Her går flyten av oppgaver fra "ikke påbegynt" til, "i prosess" før de tilslutt blir fullført. Hver utvikler kan dermed lett velge en task herfra og få god innsikt i hva som trengs å gjøres i en prioritert liste. Med JIRA får vi flere grafer som indikerer hvordan prosjektet går basert på story points og tid brukt. Slik kan man også finne ut omtrent hvor mye arbeid som gjenstår, målt i timer. Dette ble veldig praktisk da vi nærmet oss slutten og måtte ta noen harde valg om hva som skulle bli med, og hva som skulle kuttes. Jira er også i bruk hos Mesan så vi ble oppfordret til å bruke det og vi fikk et heldags-kurs i hvordan man best kan benytte seg av det.

2.6 PROSJEKTSTYRINGSDOKUMENTER

2.6.1 RISIKOMATRISE

En viktig del av ethvert prosjekt er bedømming av risiko. Vi lagde derfor en risikomatrise som tar for seg risikoer i forhold til:

- Konsekvensen, eller hvor alvorlig risikoen er **K** (0-10)
- Sannsynligheten for at problemet skjer **S** (0-10)
- Risikopoeng, produktet av de to. **RP** (K * S)

Jo høyere risikopoeng en risiko har, jo mer alvorlig er den. Disse risikoene er de vi måtte være ekstra oppmerksomme mot.

Risiko	K	S	RP	Forebyggende tiltak	Tiltak dersom problem oppstår
Defekt/stjålet laptop	5	2	10	Ha en reserve-laptop med nødvendig software installert og klar til bruk.	Ta i bruk reserve laptopen
Fravær ved kortvarig sykdom	2	6	12	God dokumentering av alle arbeidsoppgaver, slik at en annen person enkelt kan sette seg inn i oppgavene som skal løses.	Finn ut hvordan dette påvirker prosjektet, og om dette fører til endringer i størrelse/kompleksitet.
Fravær ved langvarig sykdom	6	2	12	God dokumentering av alle arbeidsoppgaver, slik at en annen person enkelt kan sette seg inn i oppgavene som skal løses.	Finn ut hvordan dette påvirker prosjektet, og om dette fører til endringer i størrelse/kompleksitet.
Gruppekonflikt/ misnøye innad i gruppen	3	2	6	Alltid beholde humoren, og kunne se de lyse sidene ved prosjektet.	Prøve å løse problemet innad i gruppen, dersom det ikke lar seg gjøre ta et møte med intern veileder.

Gruppen overholder ikke tidsfrister ovenfor kunde underveis i prosjektet	7	3	21	Sørge for god planlegging i en tidlig fase av prosjektet, samt sørge for at hver sprint består av realistiske mål og arbeidsmenger.	Gi beskjed så tidlig som mulig til kunde dersom vi ser at vi ikke klarer å overholde fristene.
Ikke mestrer bruk av ukjent teknologi	7	2	14	Sette seg tidlig inn i teknologiene som skal benyttes for å løse oppgaven/prosjektet.	Søke råd fra intern og ekstern veileder.
Løsningen tilfredstiller ikke kundens krav	9	3	37	Benytte skrum hvor vi legger opp sprintbackloger som er godt planlagte samt ha akseptansetester. Finne kundens behov/krav tidlig i prosjektet, slik at vi kan planlegge sprintene ut ifra dette.	Gjøre nødvendige endringer slik at kundens krav oppfylles. Be om hjelp fra tilgjengelige ressurspersoner.
Utvikling av "unødvendige" funksjoner	4	3	12	Sette opp grundige sprint backloger med godt definerte og detaljerte oppgaver.	Fjerne funksjoner som ikke bidrar til å gjøre løsningen bedre.
Nye brukerkrav	8	1	8	Kartlegge brukerkrav, samt sørge for en oppdatert backlog.	Finne ut viktigheten av det nye kravet i samsvar med kunde, samt legge det inn i backlog.
Tap av data	10	1	10	Bruke Git slik at all kode og dokumenter lastes opp dit. Ta backup av data	Vurdere omfanget og viktigheten av tapt data, deretter sette opp en ny plan

				etter hver sprint. (lagres eksternt)	på hvordan og hvor mye vi må jobbe for å ta igjen det tapte.
--	--	--	--	--------------------------------------	--

2.6.2 ARBEIDSPLAN

For å estimere arbeidsmengde holdt vi en planning poker. Hensikten er å estimere vanskelighetsgrad på hver enkelt oppgave i Fibonacci-tall, kalt "story points". Denne gir som regel en ganske nøyaktig estimering av arbeidsoppgavene om man runder oppover etter hele gruppen har diskutert seg frem til et tall.

Den originale planen finnes i sin helhet i vedlegg 6.2.1.

Det var relativt stort samsvar mellom planen vi satte opp i starten, og det vi fullførte med. En del av endringene kom som følge av at vi spesifiserte mer hva oppgaven faktisk dreide seg om. Vi føler det var en stor fordel å ta seg god tid i starten, og kunne se at vi traff godt på estimatene våre. Eksempelvis hadde vi estimert vår vanskeligste oppgave til 13-? antall story points, med andre ord så vi for oss at dette kom til å bli meget vanskelig å få til. Denne oppgaven ble ikke mulig å gjennomføre på et såpass lite prosjekt som vårt, med våre begrensede kunnskap innenfor feltet. Dette stemte da godt overens med vårt estimat, og de andre mindre oppgavene, så også ut til å være ganske godt estimert i forhold til hverandre.

De resulterende "functionality storyene" vi satt igjen med tilslutt på JIRA finnes også i sin helhet i vedlegg 6.2.2.

2.6.3 MILEPÆLSPLAN

Under denne perioden har vi brukt sprintinndelingen vår som en måte å holde styr på milepæler i prosjektet. Det er vanlig å lage sprinter etter user stories (eller tasks i JIRA), men da dette er et produkt som skulle lages fra bunnen av, var det ingen funksjonalitet å bygge på. Dette gjorde at vi opprettet flere stories(milepæler), som "Arbeidsliste", og konsentrerte oss om disse med relevante tasks hver sprint. Dette gjorde at vi kunne konsentrere oss om å få funksjonalitet ferdig, og utbedre appen ved hver sprint.

For å se inndelingen av milepælene, se 2.6.4 Fremdriftsplan.

2.6.4 FREMDRIFTPLAN

2.6.4.1 OPPRINNELIG ESTIMAT

Vi valgte å ha fem hovedsprinter for å implementere hovedfunksjonaliteten i prosjektet, som i diagrammet er merket oransje. I "Misc-sprinten" skulle flere mindre funksjonaliteter som ikke var store nok til å dedikere en hel sprint til implementeres.

Vi hadde to buffer-sprinter merket mørkeblå i diagrammet. Den første av dem, "Buffer - PreSprint", var tenkt for å bruke tid på å sette opp utviklingsmiljø, og så smått begynne å kode for å få en bedre forståelse for bedre og mer effektivt planlegging av hovedsprintene.

Den andre (Buffer - CatchUp) var tenkt som en mellomsprint for å fikse opp i all funksjonalitet som vi muligens ikke ville rekke å implementere i hovedsprintene. Vi forventet at noe slikt kom til å skje, siden vi har holdt på med mye ukjent teknologi vi ikke har vært borti tidligere, og estimering ble derfor betydelig vanskeligere.

"Wrap-up App" var den siste programmeringssprinten for å fikse alle mangler eller feil appen skulle ha, og ferdigstille appen til noe presenterbart.

Selv om planen var å jobbe jevnt med dokumentasjonen gjennom hele perioden, dedikerte vi to uker på slutten til "Wrap-up Dokumentasjon", for ferdigstilling av sluttrapporten.

Dokumentasjon og testing skulle foregå jevnt utover hele prosjekttiden.

Orginalt estimat, vist i Figur 3:

	January					February				March				April				May				
Activity	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Buffer - Pre Sprint					5																	
Arbeidsliste						6	7	8	9													
Innscanning									9	10												
Ruteliste & Signatur										11	12											
Ruteoptimalisering											12	13										
Buffer - CatchUp												14	15									
Misc													15	16	17							
Wrap-up App																	17	18				
Wrap-up Dokumentasjon																			19	20	21	22
Dokumentasjon									4													
Testing									5													

Figur 3 Orginal fremdriftsplan

2.6.4.2 OPPDATERT ESTIMAT

Underveis i prosjektperioden ble vi oppmerksomme på at vi ikke hadde tatt godt nok hensyn til diverse bug-fixes og utbedring av tilsvynelatende ferdig kode. Dette førte til at vi i første omgang flyttet vår "Buffer - CatchUp" sprint frem en plass. Vi satset på at denne justeringen skulle være nok for å få prosjektet på riktig spor igjen.

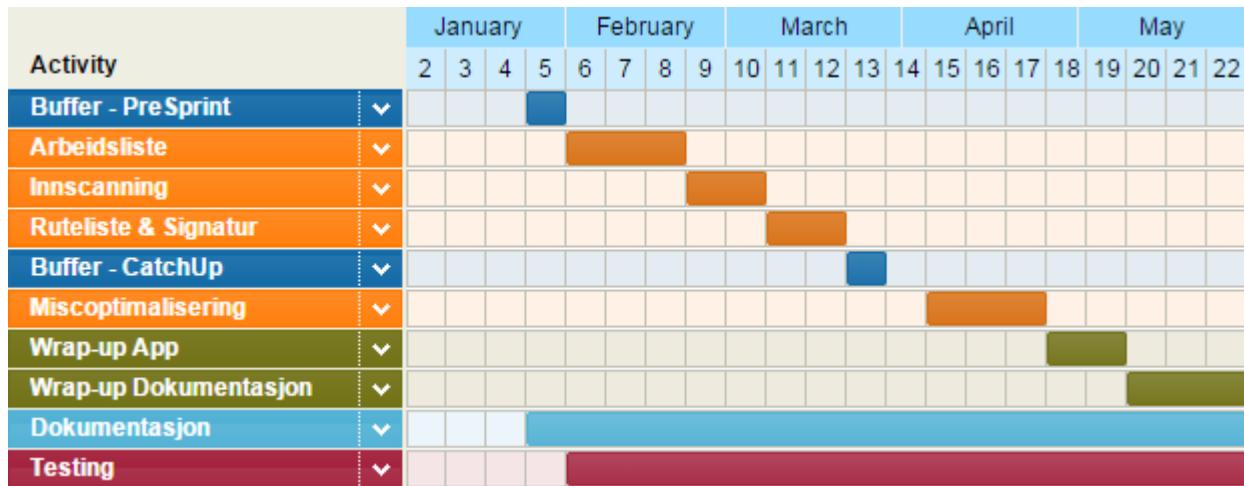
Oppdatert plan 1, vist i figur 4:

	January					February				March				April				May				
Activity	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Buffer - Pre Sprint					5																	
Arbeidsliste						6	7	8	9													
Innscanning									9	10												
Ruteliste & Signatur										11	12											
Buffer - CatchUp											12	13										
Ruteoptimalisering												13	14	15								
Misc													15	16	17							
Wrap-up App																	17	18				
Wrap-up Dokumentasjon																		19	20	21	22	
Dokumentasjon									4													
Testing									5													

Figur 4 Første oppdaterte fremdriftsplan

I påskeuka hadde vi en innlevering på skolen som måtte prioriteres. Ruteoptimalisering var hovedsakelig løst med Google Directions fra før, og derfor slo vi sammen Misc- og Ruteoptimalisering til en samlet sprint. Siden vi lå langt framfor planen, justerte vi planen deretter.

Oppdatert plan 2, vist i figur 5:



Figur 5 Andre oppdaterte fremdriftsplan

2.6.5 SPRINTER

I løpet av prosjektperioden har vi hatt flere sprinter. Under noen av disse fokuserte vi mer på å legge til ny funksjonalitet, mens i andre sprinter fikset vi opp i feil i programmet eller la til ting vi ikke klarte å få ferdig tidligere.

2.6.5.1 PRE-SPRINT

I den aller første sprinten gjorde vi oss kjent med rammeverkene, både for dokumentføring og utvikling. Vi lærte oss å bruke Confluence og JIRA for dokumentasjon. For utvikling lærte vi mer om "AndroidAnnotations"

2.6.5.2 ARBEIDSLISTE

Den første reelle sprinten var sprinten for arbeidslista. Hovedmålet med denne sprinten var å bli ferdig med arbeidslista, altså vinduet med kart og liste over pakkene sjåføren skulle levere ut. Vi implementerte både lister og kart i løpet av denne sprinten, sammen med første utgave av pakkecard.

EKSEMPEL PÅ USER STORIES INKLUDERT I DENNE SPRINTEN:

- ha et kart over ruta fordi jeg vil se hvor jeg skal kjøre
- endre rekkefølgen på leveringene, for å kunne optimalisere ruten etter egen kunnskap
- se hvilken pakke som er den neste som skal leveres, for å lett holde styr på ruta

2.6.5.3 INNSCANNING

For oss var scanning den neste store oppgaven som måtte på plass ettersom det er en veldig sentral del av applikasjonen. Da lagde vi innscannings-vinduet.

Eksempel på User stories inkludert i denne sprinten:

- scanne inn pakker for å verifisere at jeg har med korrekt last
- kunne manuelt legge inn pakker i systemet, for å håndtere feil ved scanning

2.6.5.4 RUTELISTE & SIGNATUR

Denne sprinten tok vi for oss ruteliste- og signaturvinduet. Dette var viktig for å knytte hele programmet sammen. Hovedflyten ble ferdig denne sprinten, dette medførte også flere transitions-oppgaver.

Eksempel på User stories inkludert i denne sprinten:

- Ha god oversikt over hva jeg signerer på, slik at jeg vet jeg får korrekt pakke(r).
- enkelt ha oversikt over mine ruter, for å lett kunne velge min rute
- kunne motta signatur fra kunde ved levering, for å dokumentere leveransen

2.6.5.5 BUFFER - CATCHUP

Rundt påske hadde vi en innlevering i et annet fag som gikk parallelt med hovedprosjektet, derfor la vi ikke til noen nye oppgaver denne sprinten. Fokuset lå derfor på å fikse forskjellige bugs med appen og ta igjen arbeid vi utsatte av forskjellige grunner.

2.6.5.6 MISCOPTIMALISERING

Denne sprinten skulle hovedsakelig brukes på å få ruteotimaliseringen på plass, men da dette viste seg å være en for stor oppgave ble sprinten kun en optydning og fiksing av bugs. Vi la også til nødsituasjonskjerm og noe funksjonalitet rundt den.

Eksempel på User stories inkludert i denne sprinten:

- kunne se statistikk om min arbeidsdag/måned, fordi dette kan være nyttig
- kunne hente ad-hoc pakker for å være fleksibel i arbeidet
- ha en hjelpefunksjon, for å vite hvordan jeg skal bruke systemet

2.6.5.7 WRAP-UP APP

Siste kodesprint i prosjektet. Her var det kun fiksing av kjente bugs, og fullføre alle deler av appen.

2.6.5.8 WRAP-UP DOKUMENTASJON

Dette var den siste sprinten, og den ble kun brukt på dokumentasjon. Det ble ikke åpnet en ny sprint på JIRA for dette.

2.6.6 DAGBOK

En viktig del av dokumentasjonen for oss underveis i prosjektet var dagboken. Vi satt 3 dager i uka hos Mesan, og skrev hver dag ned hva vi hadde gjort, arbeidsoppgaver utført, avgjørelser tatt og eventuelle problemer eller planer vi hadde lagt videre. Her ble det dokumentert hvem som jobbet med hva, og fremgangen som ble gjort med disse oppgavene. Alle dagbøkene er lagt ved i vedlegg 6.4. I starten av prosjektet var ikke disse utformet like bra, og vi skulle gjerne likt å ha flere av de tidlige valgene våre dokumentert bedre der.

Eksempel på et innlegg i dagboka.

Dagbok 27.04.2015

Dato:	27. April 2015
	<p>Idag startet vi ny sprint, så vi hadde møte fram til lunsj hvor vi gikk gjennom tasks og bugfixes som skal være med i denne sprinten.</p> <p>Dette er siste sprint vi har muligheten til å utvikle appen videre/rydde opp, så det ble gjort noen valg på hvordan dette skulle håndteres.</p> <p>Vi har også avtalt tid for en akseptansetest neste uke med Harald og Anders.</p> <p>Sondre har sett på hvordan keyboard på emulatoren skal oppføre seg når vi må åpne dette. Har også ryddet opp i kode som nå ikke er relevant mer.</p> <p>Lars-Erik har fikset arbeidslista. Denne lista har vært veldig bugga i flere uker. Etter dette hjalp han Joakim med tutorials.</p> <p>Joakim har laget og implementert flere av tutorial skjermene i appen, men dette krever fortsatt noe arbeid.</p> <p>Martin har brukt store deler av dagen på å jobbe med nødsituasjon-skjermen. Mangler nå kun animasjonen for å simulere overføring av pakker.</p>
Avgjørelser:	Vi lagt til flere tasks og ryddet opp de som var overflødige på JIRA for å forberede siste sprint(med koding).
Problemer:	Ingen store problemer.
Til neste gang:	Forberede akseptansetest til neste uke.

2.7 MØTER

2.7.1 PLANLEGGINGSMØTE MED HARALD

Siden vi ikke har hatt direkte kontakt med Postnord under prosjektperioden, har Mesan fungert som produkteier. Harald Kulø, konsulentsjef ved Mesan, har derfor vært svært delaktig under planleggingen av prosjektet. Harald har vært med på nesten alle møter gjennom perioden, og har vært han vi hovedsakelig har spurt om løsninger og problemstillinger.

I starten av prosessen ble det holdt flere møter for å få et overblikk over oppgaven. Harald ledet disse møtene, og hjalp oss med denne delen.

I vedlegg 6.6 er et referat fra det første planleggingsmøtet inkludert.

2.7.2 MØTE MED POSTNORD-TEAMET

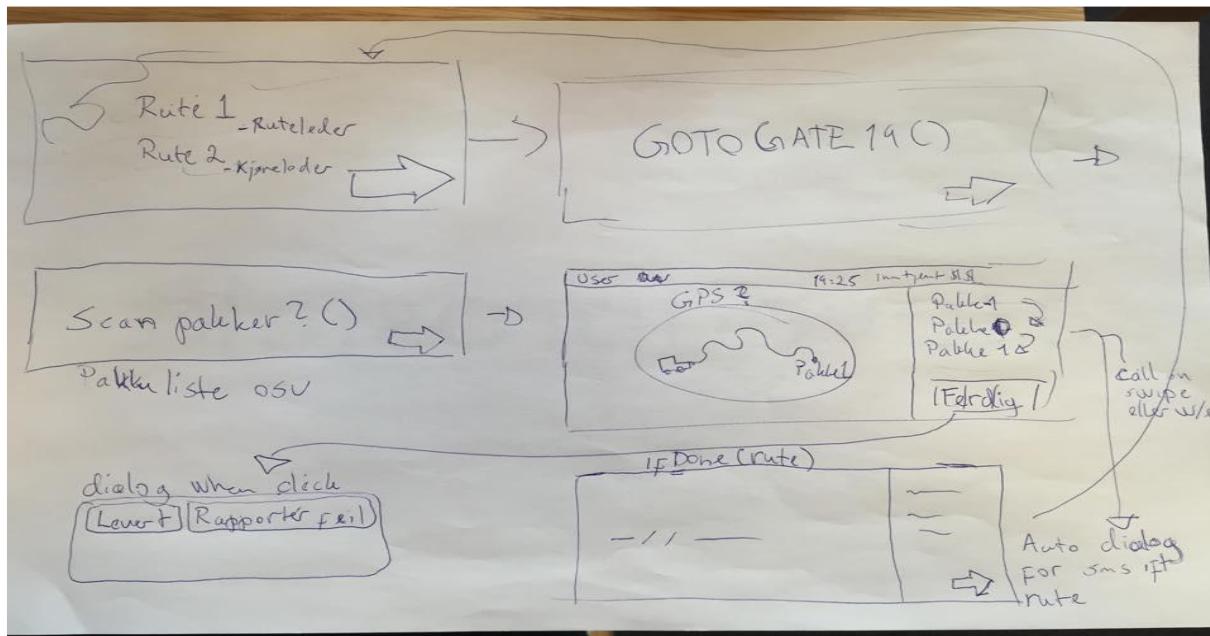
I hovedkontoret til Mesan, sitter det allerede ett team som jobber med løsninger knyttet til Postnord. De sitter for øyeblikket med utvikling av det såkalte plottebordet, som er en annen del av PostNords logistikkvirksomhet. På grunn av dette har de noe innsikt i hvordan sjåførenes hverdag utarter seg, og derfor har vi også hatt et par møter med dem i prosjektperioden. Her har de forsøkt å gjengi hvordan en hverdag for en sjåfør ser ut, og kommet med ideer til hvordan vi kan løse våre oppgaver.

Vi tok flere viktige valg rundt utviklingen videre på grunnlag av disse møtene. Vi ble blandt annet enige om at løsningen skulle lages for én type device, fikk input på flere av ideene til ny funksjonalitet og ble enige om at vi selv skal gjøre antagelser underveis. Mer detaljer rundt hva som ble diskutert på disse møtene ligger i vedlegg, under punkt 6.3.

Ettersom dette teamet har hatt mye å gjøre gjennom hele perioden, har vi ikke fått til mer enn to møter. Likevel var disse møtene veldig verdifulle, da de hadde unik innsikt i flere av oppgavene vi skulle starte med.

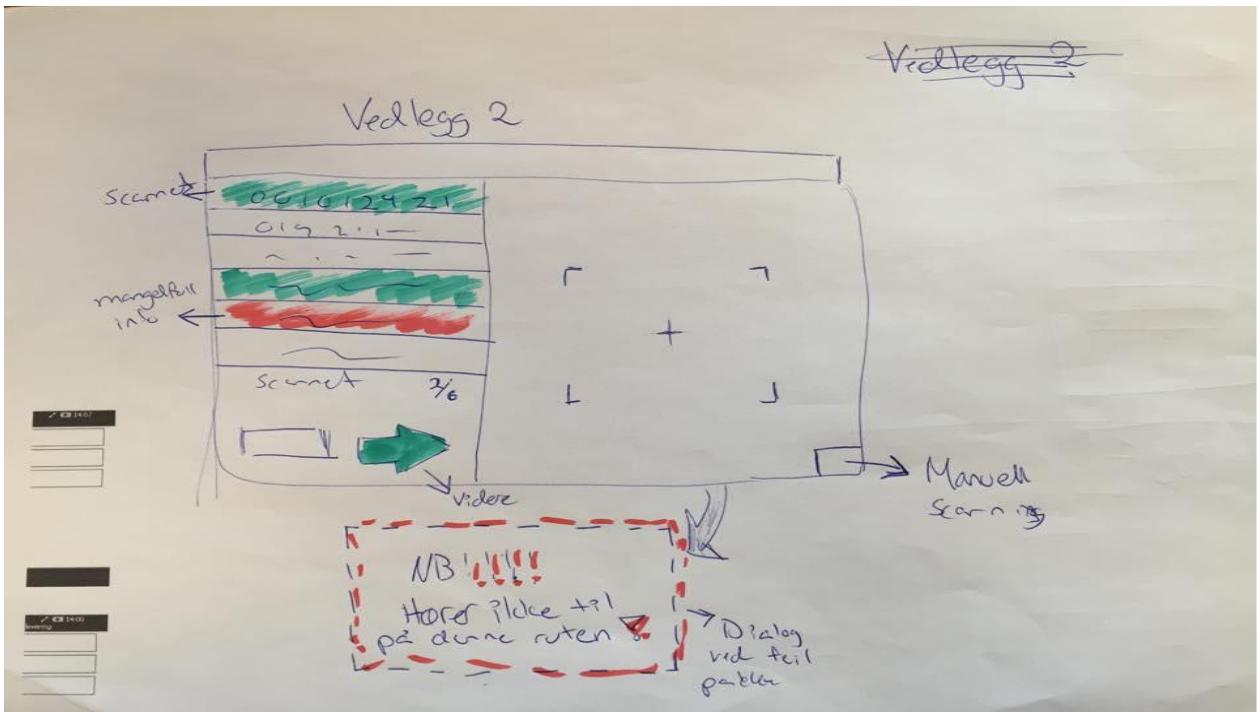
2.7.3 MOCKUPTEGNING

Under planleggingen av prosjektet skisserte vi opp flere mulige løsninger for de skermene vi viste måtte være med. Det ble flere utkast, og etter mye diskusjon kom vi frem til versjoner vi mente var gode nok til å fungere som retningslinjer videre.



Figur 6 Mockup 1

Figur 6 viser første utkast av hovedflyten. Sluttproduktet skiller seg ikke mye fra denne skissen, men Go to gate-skjermen er fjernet. Dette bildet viser også ønsket om å ha en felles knapp for å indikere flyt(pil på bildet), noe vi løste med floating action button.

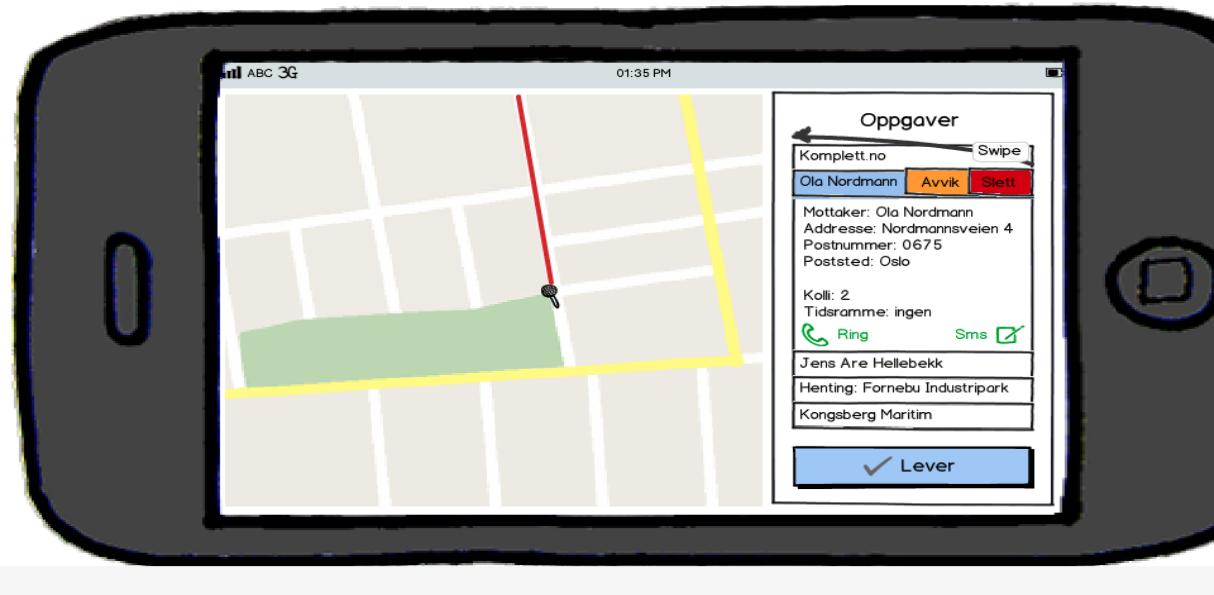


Figur 7 Mockup 2

Figur 7 viser første utkast av innscanningsskjermen. Hovedtrekkene ble bevart i løsningen vår, men pil ble erstattet med floating action button. Samtidig ble noen av funksjonene flyttet noe rundt.

Mesan sin wiki-løsning har også et verktøy de bruker ved modellering av løsninger. Dette verktøyet kalles Balsamic og brukes for å lage skisser på maskin, som ser håndtegnet ut. Vi fikk en rask innføring i dette verktøyet og bestemte oss for å lage de endelige skissene våre med dette.

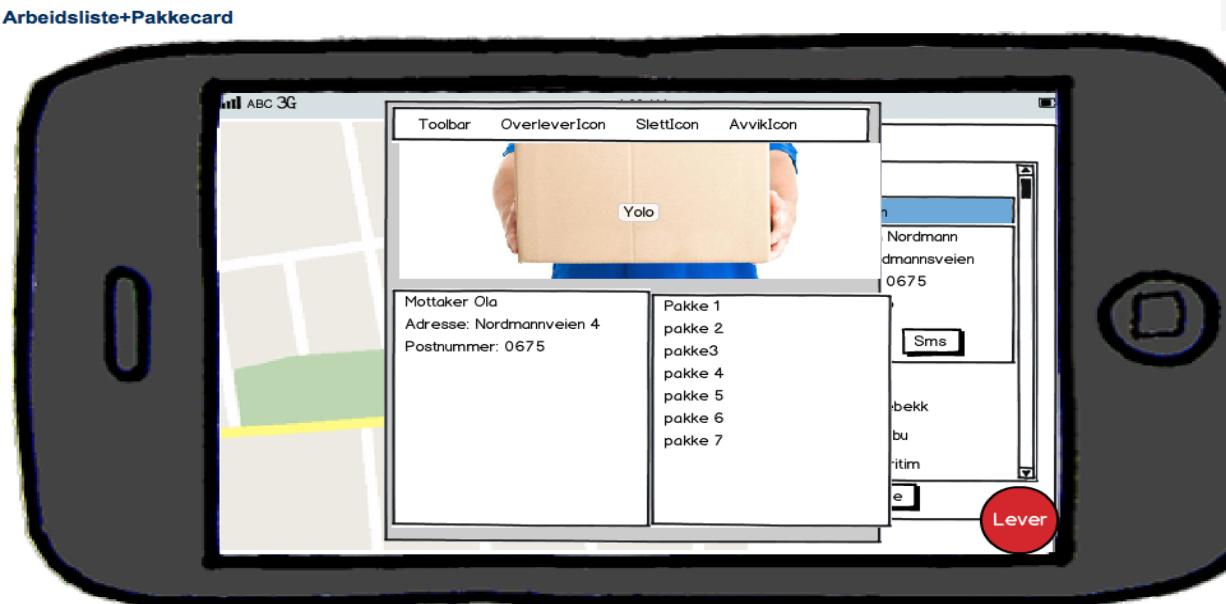
Kart/Arbeidsliste



Figur 8 Mockup 3

Figur 8 viser en endelig detaljert skisse av arbeidsliste. Rent grafisk har ikke skjermen endret seg stort, men mye ny funksjonalitet har blitt lagt til etter dette, og mye har blitt mer forenklet for å ikke virke rotete.

Arbeidsliste+Pakkecard



Figur 9 Mockup 4

Figur 9 viser en skisse over pakkecard. Denne skjermen ble lagt til på et noe senere tidspunkt. Hovedsakelig har ikke skjermen endret seg stort, men som vist på bildet har arbeidslista fått floating action button.

Disse bildene ble på flere tidspunkt gått gjennom med Mesan for å forsikre oss om at vi er på rett vei med løsningen vår. Resten av bildene legges ved som vedlegg, punkt 6.9.

2.7.4 PRESENTASJON FOR POSTNORD

Som siste oppgave vi utførte for Mesan presenterte vi vår løsning for PostNord selv, i et møte vi ledet. Vi gjorde klar en presentasjon hvor vi fortalte om produktet, arbeidsprosessen og antagelser vi har gjort under prosessen. Hele møtet varte i drøye 2 timer. Figur 10 viser 2 av slidsene til presentasjonen.



Figur 10 Bilder fra presentasjonen for PostNord. (V) Hovedflyt, (H) utbytte av løsningen vår.

Etter presentasjonen gjorde vi en kort demo av appen, hvor PostNord fikk bruke tableten samtidig som vi demonstrerte på storskjermen. Vi gikk deretter gjennom alle skjermer og funksjonaliteter i appen, og svarte på diverse spørsmål underveis.

Vi fikk en veldig god respons under hele møtet, og brukte den neste timen på å diskutere mulige fremgangsmåter, praktiske løsninger og funksjonalitet de likte og ville legge til i en ferdig løsning. Mesan var også representert på møtet, og etter møtet var ferdig ble det avtalt et oppfølgingsmøte med PostNord for bestemme veien videre.

PostNords representanter så nytten av vår løsning, og har allerede sett etter lignende løsninger selv. De forklarte også at det var flere bruksområder hvor de gjerne ville implementere en en forenklet variant av vår proof of concept, før de eventuelt skifter ut deres eksisterende håndterminal for sjåførene. Blant annet var det snakk om en lignende løsning for mobil for andre aktører de har samarbeid med. Blant annet gjelder dette bussjåfører som kan komme til å måtte levere en pakke eller to. Det var også snakk om en løsning for butikker og bensinstasjoner som håndterer "Post i butikk".

Både PostNord og Mesan virket veldig fornøyd, og så flere muligheter med vår løsning. Mesan ville også få tilsendt presentasjonen vi brukte som en basis for videre møter.

2.8 TESTING

2.8.1 AKSEPTANSETESTER

Som en måte å kvalitetsteste vår app og å se om produktet møtte kundens forventninger underveis, utførte vi flere akseptansetester. Akseptansetester utføres for å avklare om et system oppfyller kriteriene til kunden, og stemmer overens med spesifikasjonene og annen dokumentasjon.

Siden vi ikke har hatt kommunikasjon med Postnord under utviklingen av denne appen, var det Mesan som som fylte rollen som kunde under disse testene. Harald Kulø var produkteier under disse testene, og gjennomførte våre tester sammen med kollegaer han mente burde være med.

Disse møtene ble veldig viktige for oss. Funksjonalitet som virket åpenbar for oss men ikke for alle andre, ble påpekt. Ideer til forbedringer ble foreslått, og feil & bugs i appen ble funnet. Etter hver test gikk vi så gjennom alle punkter vi tok opp i møtene, og gjorde nødvendige endringer i appen deretter.

Resultatene av disse testene er lagt ved i vedlegg 6.3.

2.9 UTFORDRINGER OG AVGJØRELSER

Under denne perioden har vi gjort en rekke valg og beslutninger rundt hvordan appen skal se ut og hvilken funksjonalitet vi vil implementere. Vi har alle vært svært aktive i diskusjonene rundt disse beslutningene, og noen ting har blitt endret flere ganger.

Foruten løsninger på problemer og bugs, har vi hatt flere møter, samtaler og tester for å bestemme hvordan vårt produkt kan bli best mulig. Akseptansetestene har vært svært viktige for å finne ut om det vi har laget matcher kundes forventninger.

Flere steder i denne dokumentasjonen har vi forklart tanken bak flere av de viktige valgene som er tatt for dette prosjektet. Dagbøkene vi har ført gjennom prosjektet, sammen med bl.a møtereferatene våre, dokumenterer flere av valgene, men for å gi en kort oversikt over endringer gjennom prosjektet har vi laget en liste over de viktigste avgjørelsene her.

Problemstilling	Valg	Begrunnelse
Android 5.0, Lollipop, med Material Design-standard, fremfor tidligere versjoner av Android.	Android 5.0, Lollipop er den nyeste versjonen av Android, og har en del ny funksjonalitet, særlig for design og GUI. Utfordringen med å bruke en så ny versjon, er at den fortsatt kan inneholde bugs, og har ikke like mye dokumentasjon som tidligere versjoner.	Vi valgte å gå for Lollipop og Material Design ettersom vi lager en proof of concept-løsning, med vekt på GUI og "flair". Ved å utnytte bl.a. transitionsbiblioteket i Lollipop, får vi en løsning som ser bedre ut for kunden. I tillegg ser vi på dette som en mulighet til å lære den nye standarden innen vårt felt tidlig.
Teknologi: RecyclerView	Appen vår inneholder en rekke lister (arbeidsliste, ruteliste, osv.), og her har vi valget mellom å bruke vanlige listviews, eller androids nye	Vi valgte tidlig å gå for recyclerview, fordi det har en fordel i form av ytelse, og den så ut til å være en fleksibel klasse, som er en fordel med tanke på

	listeklasse, RecyclerView.	hvor mye funksjonalitet vi har i våre lister.
Teknologi: Annotations	Vi valgte å ta i bruk Android Annotations, fremfor å sette opp Activities og andre klasser på vanlig måte.	Vi valgte å bruke Annotations, ettersom dette sparer oss for mange unødvendige linjer med kode, og gir noen nye muligheter på en mye enklere måte enn om man gjør det på vanlig måte. På samme tid minker det sjansen for feil, siden mye kode blir generert for oss.
Teknologi: Activities, ikke Fragments	Vi lurte lenge på om vi skulle bygge appen rundt fragments eller activities, som er to forskjellige måter å bygge opp skjermbilder i Android. De har begge sine fordeler og ulemper.	Vi valgte til slutt å gå for Activities, ettersom de har et eget transitions-bibliotek i Android 5.0, og i tillegg er oversiktlig og greit å jobbe med. Dette bestemte vi ca i uke 2, etter å ha møtt på et par problemer med fragments i forbindelse med transitions. Ikke minst har activities sakte tatt over store deler av fragments sin rolle i Android.
Teknologi: Transitions	For å skape mer "flair" i appen, ville vi ha fine overganger og animasjoner i appen for å tydeliggjøre flyten, bedre brukeropplevelsen og imponere kunden. Vi valgte derfor tidlig å ta i bruk Lollipops Transition-klassen.	Ved å bruke transitions, som er en helt ny måte å lage animasjoner mellom activities i Android 5.0, kunne vi skape fine overganger mellom activities, med relativt lite arbeid. Dette sparte oss for mye tid i forhold til å kode animasjoner manuelt.
Teknologi: Animasjoner	Igjen var ønsket vårt å gjøre appen så fancy, spennende som mulig, visuelt. En viktig del av det var animasjon av skjermelementer der	Igjen tok vi i bruk ny teknologi i Android 5.0, med Animation-klassen. Dette lot oss animere skjermelementer som knapper og listerader, både i form og posisjon, noe som gir appen et mer

	dette var hensiktsmessig.	levende preg, og føles mer responsivt for bruker. Dette ser også mer imponerende ut for kunden.
Grafisk hovedvekt, fremfor fokus på stabilitet og sikkerhet.	I enhver app må utvikler ta stilling om tiden skal disponeres til å sørge for et godt brukergrensesnitt og design, eller mer mot ytelse, stabilitet, sikkerhet, osv. En kombinasjon av disse vil alltid finne sted, men vektingen varierer.	Ettersom vårt prosjekt går ut på å lage en proof of concept-løsning, har vi valgt å fokusere på det grafiske designet, brukervennlighet, og til en viss grad ytelse i form av et responsivt design. Sikkerhet og andre backend-relaterte faktorer, ble nedprioritert, da hovedoppgaven til et proof of concept, er å overbevise kunden om at produktet har potensiale. Hvis kunden liker resultatet, vil utviklingen på en endelig versjon starte i etterkant, og det er først der backend må være sikkert og stabilt.
Kartvisning og gps	Som en av de viktige nye funksjonalitetene vi ville legge til i appen, var en type fremvisning av ruten den viktigste. Her var det mange faktorer vi måtte ta hensyn til, men vi ble fort enige om at et overblikk med lette instruksjoner var det vi ville gå for.	Med tanke på kompleksiteten ved oppgaven, og tiden vi hadde til bruk ble vi enige om at et overblikk ville være et bedre valg. Her fantes det Google Api'er som ville hjelpe oss og fjerne noe av behovet for å lage alt fra bunn av. Vi tenkte også at siden en rute består av flere stopp, kunne det være nyttig for en sjåfør å se neste stoppesteder osv underveis. Dette gjør det også lettere å se/endre på hvilken rekkefølge leveransene skal leveres.
Google maps libs	For å vise kart trengte vi en løsning som var simpel nok å implementere, men hadde mye funksjonalitet.	Ettersom kart er et helt eget fag var det utenkelig at vi skulle lage noe slikt fra bunnen av. Google Maps har biblioteker som er direkte støttet i

	Derfor valgte vi Google Maps.	Android, som gjorde det veldig enkelt å implementere.
Google Directions	For å gjøre levering av pakker mest mulig effektiv, tenkte vi å beregne og vise en optimalisert kjørerute til sjåføren. Vi gikk derfor for Google Directions.	Google Directions har støtte for å optimalisere rekkefølgen for en liste med adresser. Dataene som vi får fra Directions kan også brukes til å tegne hele den egentlige ruten som sjåføren faktisk kjører.
Forenkle dagens flyt	Vi valgte at et av punktene vi skulle prioritere mest med denne løsningen var å forenkle og forkorte dagens flyt.	Etter de første møtene med Mesan fikk vi en oversikt over PostNord sin eksisterende løsning. Denne løsningen var ganske uoversiktlig og rotete. Den krevde mange tastetrykk for å utføre vanlige oppgaver, og var ikke intuitiv å bruke. Derfor ble derfor et av hovedfokusene i appen.
Fjerne unødvendig funksjonalitet	Fra dagens løsningen er det noe funksjonalitet som ikke ble med i vår app. Dette var ting vi mener kunne løses på andre måter, eller ikke lengre er nødvendig med våre løsninger. Et eksempel på dette er "printing av skrekkoder", dette ser vi for oss blir løst på andre måter. F.eks tredjepart.	For å korte ned flyten i denne appen har vi slått mye funksjonalitet sammen og fjernet noe. Vi satt tidlig opp en liste over alt som skulle med i vår løsning, og funksjonalitet som virket overflødig etter dette ble fjernet.
Tablet 7.5 inch (ble 8.4)	Vi utviklet denne appen for størrelsen 8.4 inch. Det var flere ting å ta hensyn til rundt dette valget, og det viktigste var å kunne balansere det praktiske og funksjonaliteten.	Etter litt testing og diskusjoner ble enige om at størrelsen rundt 8.4 inch ville være ideelt med tanke på mengde funksjonalitet vi får plass til pr side, og hvordan den vil føles i hånden for sjåføren. Ikke minst vil den passe som

		en konsoll montert på et stativ i bilen.
Inndeling av leveranse/pakker.	Under plaleggingen av prosjektet, utviklet vi også en plan på hvordan leveranser/sendinger/pakker skal håndteres. Vi ser her for oss at et stop, eller en leveranse på en adresse, kalles en Task. Denne tasken har egen informasjon og består av en liste med en eller flere pakker.	For å bygge opp strukturen til vår løsning var det viktig å få oversikt over leveranser og pakker. Dette har gjort det lettere å samle pakker og organisere de forskjellige delene av appen.
Håndtering av avvik.	Vi valgte å gjøre noen endringer i hvordan avvik blir håndtert for leveranser. I vår løsning tenker vi at en sjåfør nå melder avvik for en Task, og tar deretter med seg den tasken tilbake. Dette skiller seg fra eksisterende løsning med at det kunne meldes avvik pr pakke.	Dette er hovedsaklig en proof of concept-løsning på avvikshåndtering. Pga tidsmangel og viktigheten av funksjonen i en testsammenheng virket dette som en god løsning. I en ferdig løsning ville det vært logisk å fortsette med avvik pr pakke, og kunne levere f.eks. % pakker hvis en er skadet/mistet.
Lokal database - Sugar ORM (Se 3.2.5)	Vi valgte å bruke databasesystemet Sugar ORM i vår app. Dette er et objekt-relasjonelt framework på toppen av SQLite.	Vi trodde lenge vi ikke ville trenge noe lokal lagring av data på vår app, men det ble fort klart at dette var naivt. Grunnet negativ erfaring med SQLite i Android, gikk vi derfor for Sugar ORM. Vi diskuterte også i starten å bruke informasjon fra deres eksisterende backend, men vi skjønte fort at dette ville være for komplisert for dette prosjektet.
Innbokssystem for	Vi tenkte i starten på å legge til et system for å samle meldninger og	Dette virket som en funksjonalitet som ville gi lite nytt og nødvendig til

sjåfører	anrop mellom andre sjåfører og kjøreleder. Dette var noe vi gikk bort fra tidlig.	sjåførene, så vi valgte å ikke gå for dette. Men hovedsaklig ville dette tatt lang tid å implementere ordentlig.
Melde fra om trafikkavvik	En tanke tidlig i utviklingen var å la sjåfører kunne fortelle andre om eventuelle avvik i trafikken. Planen var å gjøre dette i appen, så andre sjåfører kunne endre sine ruter etter den nye informasjonen. Pga tidsmangel ble ikke dette prioritert.	Denne funksjonaliteten kunne være veldig nyttig for sjåførene, men pga kompleksiteten ble det bestemt at vi ikke skulle gjøre dette.
"Kjøreleder kontroll" - ved innscanning.	Etter innscanning i den eksisterende løsningen må sjåførene kontakte kjøreleder og få godkjent sin "rute". Dette tenkte vi å digitalisere å automatisere. Dette ble ikke noe av pga tidsmangel. Vi erstattet dette heller med en sjekk på hvor mange av paklene som blir scannet inn. Hvis du scanner mer enn en gitt prosent går du automatisk videre, hvis ikke må du fortsette å scanne.	Dette var nok en av funksjonalitetene fra den eksisterende løsningen som bør være med i den ferdige appen, men pga tidsmangel og vanskeligheten av å teste funksjonalitet som dette, ble det ikke med i vår app. Vi mener en test på antall scannede pakker vil være nok for å vise frem appen på en god måte.
Bilde ved levering uten signatur	Når en task skal leveres og ingen kan signere for den, lagres det idag en kommentar og koordinatene pakken ble satt. Vi tenkte å legge med et bilde av pakken, og beholde kommentar/koordinater. Vi fikk ikke lagt til kamera pga tidsmangel.	Vi ble enige om at dette ville vært nyttig funksjonitet, men med tiden vi hadde igjen ble det ikke noe av. Vi lagrer heller da dataen som gjort i eksisterende løsning.
Prioritert liste av Tasks	Vi har valgt å sortere lista over task i arbeidslista etter optimal leveranse tidspunkt. Dette gjør at lista hele	For å holde oversikten i vår arbeidsliste bestemte vi oss for å ha en standard lista blir sortert etter. Siden vi har en

	tiden er oversiktlig og stemmer med kartet. Ved ønske er det også mulig for en sjåfør å sortere lista etter eget ønske.	rute som er optimalisert virket dette også logisk å sortere lista etter.
Tidssensitive pakker	En av funksjonene PostNord gjerne ville ha i sitt nye system, var muligheten til å optimalisere rutene med hensyn på pakker som må leveres innen et tidsrom. Dette var funksjonalitet vi fikk implementert kun med eksempeldata, siden dette krever mye tid å utvikle.	Vi brukte mye tid på å finne løsninger på dette, men grunnet tidsrammene for prosjektet ble vi nødt til å nedprioritere dette. Pr idag er kun det grafiske på plass her.
Fargevalg (se 3.4.2)	Vi har brukt PostNord sin egen logo farge som hovedfarge i appen. Alle farger etter dette er nyanser av farger som passer denne. Et unntak er fargen på Floating Action Button som er valgt for å skille seg ut, men holde seg innenfor samme stil.	Vi syntes det var viktig å finne farger som passet vår oppe da mye av vårt hovedfokus har vært lagt på brukeropplevelsen. Vi har derfor tatt utgangspunkt i logo fargen, og jobbet oss videre derfra.
Floating Action Button	Som en del av den nye standarden i Android 5.0, Material Design, har FAB blitt en ny måte å indikere hovedflyt i et program. Dette er en rund knapp, som bryter med linjene i farge og plassering som skal være hovedveien videre.	Siden vi har bestemt oss for å følge Material Design prinsippene var det også naturlig for oss å implementere den nye Floating Action Button også. Denne knappen er med på alle hovedskjermer i appen, og blir brukt som standarden sier. Denne viser også fremgang ved store nedlastninger.
Sortert rute allerede ved innskanning	Som en ekstra hjelp til sjåførene tenkte vi tidlig på å kunne optimalisere rute så tidlig som mulig så man kunne laste inn pakkene i	Under planleggingen av prosjektet så vi på dette som en mulighet til å kunne gi noe ekstra til sjåførene, og i tillegg tjene noe tid. Pga tidsmangel ble det

	bilen med tanke på når de skal ut igjen. Dette ble det ikke tid til å legge til, men vi tror dette hadde vært en fin funksjonalitet.	desverre ikke tid til å implementere dette, men under presentasjonen for PostNord virket de veldig imponert av denne muligheten. Derfor er dette noe vi definitivt ville lagt til på et senere stade.
Kommunikasjon med mottaker eller kjøreleder	I en hverdag for en sjåfør er det nødvendig å kunne kontakte en kunde underveis for å informere angående leveranse, eller ringe sin kjøreleder for assistanse. En av funksjonene vi la til i denne appen var at alt dette skulle være inkludert i vår løsning. Derfor valgte vi også en tablet som med simkort, for anrop og meldinger over nett.	Vi ser for oss at vår app/device i seg selv skal være et godt nok verktøy for en sjåfør så det ikke er behov for å bruke andre enheter i tillegg. Derfor har vi gjort klart for direkte kommunikasjon fra vår app, med hjelp av native løsninger devicen allerede har. Dette er også sånn beskjeden om fremgang i ruta blir sendt ut til kunden som venter en leveranse.
SMS med leveransefremgang til kunde	Siden en sjåfør idag selv må sende en melding til kunden når han estimerer det å være 20 min igjen leveranse, tenkte vi at dette ville være en fin funksjon å automatisere. Denne funksjonaliteten rakk vi å få implementert, men siden det blir vanskelig å teste og vise frem har vi ikke den med i det ferdige produktet.	Vi var alle enige om at dette er en funksjonalitet som virket helt nødvendig for alle sjåfører. Dette ble også lagt til tidlig i prosessen, men det ble også snart klart at vi ikke kunne teste og bruke det på en fornuftig måte under utvikling, så den ble kommentert ut. Vi har allerede forskjellige estimater som blir regnet ut angående pakkens leveringstidspunkt, og det er også disse tallene denne servicen bruker for å sende melding ved korrekt tidspunkt. Dette vil også gjøre at en pakke som blir utsatt noe kan sende en ny melding etterhvert hvor den forteller om forsinkelser i leveransen og nytt tidspunkt.

Inn- og utscanning	I dag er det flere forskjellige krav til dokumentering av pakkens posisjon, og en av disse er scanning på og av bil. Dette følges i en idiell verden, men kan fort bli til at man kun scanner 50% inn og 100% ut f.eks. Vi har derfor satt en grense som må passeres når man scanner inn, og behold 100% grensen ved utscanning.	På et av de første møtene vi hadde med Mesan sitt PostNord team fikk vi høre at kravene til scanning ikke var spesielt strenge, og tidsbesparing ble ofte prioritert. Derfor åpnet vi for muligheten at det i vår løsning også skulle være mulig å scanne kun en del av pakkene ved innscanning. Vi håper derimot at scanneren vår skal gjøre det så lett som mulig for sjåførene å scanne alt, så dette kan bli den nye standarden. Pr nå er løsningen i vår app sånn at hvis en sjåfør scanner over 70% av pakkene inn vil han gå gjennom en test fra kjøreleder (Ikke implementert funksjonalitet, da vi ikke har noe backend, men meningen å skulle legge til), og hvis han scanner over 90% går han automatisk videre. Dette er for å hindre følgefeil.
Forskjellige strekkoder.	I dag er alle pakkene til PostNord utstyrt med strekkoder (barcodes) så dette var teknologien vi måtte få inn i vår scanner. Ellers la vi også til støtte for alternative koder, som QR og diverse andre formater. Vi har også tenkt på NFC, noe devicen støtter, men ikke er realistisk for pakkene pr dags dato.	Angående strekkodene var det ikke mye vi fikk endret på her. Pakkene har ingen andre formater enn vanlige strekkoder. Vi prøvde likevel å modernisere løsningen noe, samtidig som vi la opp til støtte for fremtidige løsninger.
Fire skjermer hovedflyt.	Siden en av de store oppgavene for dette prosjektet var å modernisere å utbedre den eksisterende løsningen var et av våre fokusområder å korte ned på hovedflyten fra 30+	Vi brukte den eksisterende løsningen til å finne det vi mente var de viktigste oppgavene til prosjektet. Utifra disse punktene klarte vi å komprimere

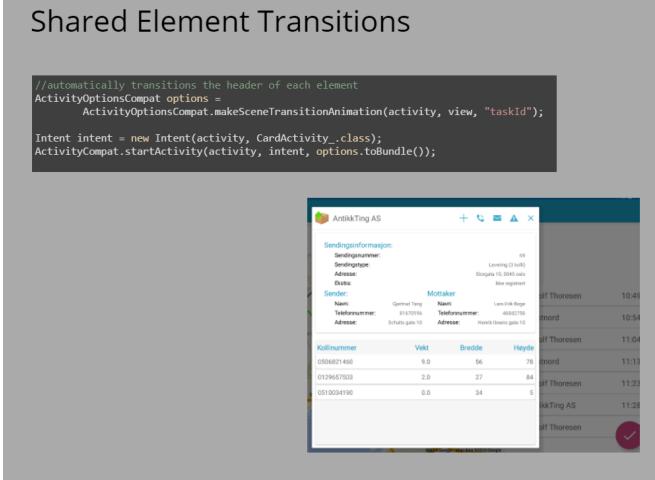
	skjermer. Dette klarte vi og sluttproduktet gir 4 hovedaktiviteter hvor den viktigste funksjonaliteten er.	løsningen kraftig.
Dokumentasjon underveis	Vi fikk en tidlig innføring i systemene Mesan bruker for dokumentasjon, og bestemte oss da for å bruke disse systemene til noe av vår dokumentasjon også. JIRA, som er et verktøy for smidig utvikling, ble brukt for oversikten over oppgaver som skulle gjøres hver sprint og fremgang. For dagbøker og andre dokumenter brukte vi Mesan sin wiki-løsning.	Siden vi utførte denne oppgaven for Mesan syntes vi det var logisk å bruke deres standarder for dokumentene som skulle lages underveis. Dette var også dokumenter som skulle brukes som vedlegg i sluttarkiv, så hvor dette lagdes var ikke veldig viktig.
Ny funksjonalitet: Nødsituasjoner	I dag håndteres alle avvik i en arbeidsdag manuelt, så en av forslagene til ny funksjonalitet vi har lagt til er automatisering av dette. I vår versjon er dette kunn et skall av tenkt funksjonalitet, med den demonstrerer muligheten for å f.eks finne en sjåfør som er i nærheten og overføre lageret til den andre sjåføren.	Dette er et av forslagene til ny funksjonalitet vi har foreslått. Vi har ikke tatt tid til å lage en komplett løsning for dette, men GUI delen er ferdig.
Funksjonalitet som ikke ble med: Avansert ruteoptimalisering med Google Distance Matrix	Sjåfører har ofte pakker som skal leveres innen et tidsrom. Google Directions tar ikke hensyn til slike pakker i optimaliseringen. Google Distance Matrix kan brukes med en egen algoritme for å kunne beregne en rute som tar hensyn til	Tanken var å bruke Google Distance Matrix for å få en kostnadsmatrise mellom adresser, for så å bruke den til å beregne en optimal rute som tar hensyn til tidssensitive pakker. Problemene var at vi kun hadde fem dager å implementere dette på. De eneste referansene for slike algoritmer

	tidssensitive pakker.	(Traveling Salesman Problem with Time Windows) er i doktor- og doktorgradsoppgaver som har et mye større skop enn hele hovedoppgaven vår alene. På grunn av tidsmangel og kompleksiteten kunne dette ikke implementeres.
--	-----------------------	--

2.10 PRESENTASJON PÅ MESAN-MØTE

Mesan har hver måned et møte hvor nye prosjekter, IT-løsninger og finansiell status blir gjennomgått i plenum. I Februar fikk vi en litt brå varsling om at vi gjerne kunne presentere på dette møtet om vi ville, og vi konkluderte at om vi fikk litt mer tid ville dette vært en kjekk ting for både oss og Mesan. Vi presenterte derfor i Mars en kort oppsummering for hele selskapet om hvem vi var, hva vi drev med og hva de interessante aspektene ved vårt prosjekt var.

Siden de fleste på Mesan er utviklere, vinklet vi presentasjonen mot de tekniske løsningene vi hadde tatt i bruk fra Android 5.0. Figur 11 en



Figur 11 Slide fra presentasjon for Mesan

av slidesene fra dette foredraget, hvor vi demonstrerte kode og animasjon for delte elementer.

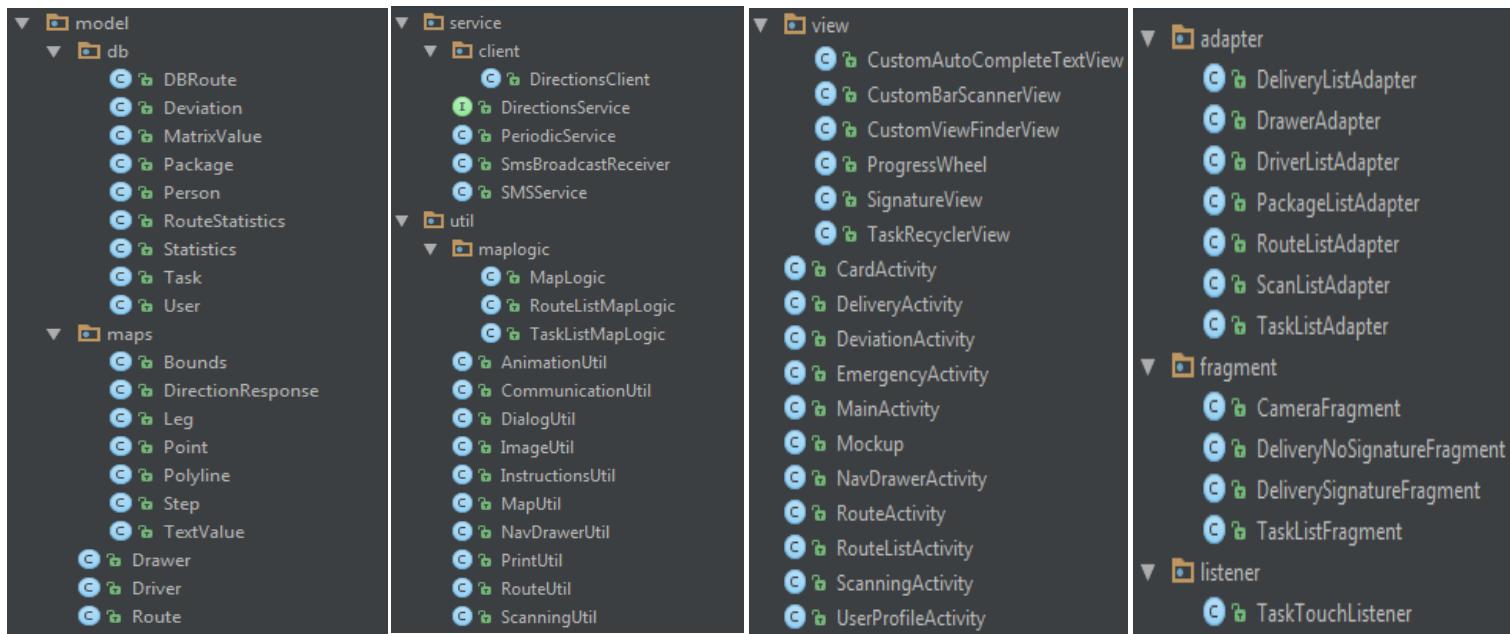
Vi konkluderte med at vi var glad vi hadde valgt denne ruten, på tross av at vi også så farene ved å velge en såpass ny plattform for utvikling. Ved oppstarten av prosjektet, fantes det nemlig nesten ingen svar på spørsmål på store nettforum som StackOverflow angående nyvinningene i Android 5.0. Det var såpass ille at den eneste vi så som ga svar, var en utvikler på

Androids eget utviklingsteam for animasjoner, med andre ord, han som hadde laget disse funksjonene.

2.11 KODESTANDARD

2.11.1 KODESTRUKTUR

Vi har gjennom hele prosjektet forsøkt å ha en god struktur på filene slik at hvem som helst av oss kan ta over for en annen, og forstå hvor en gitt funksjonalitet befinner seg. Vi grupperte klassene i Activities, Fragments, Adapter, Listeners, Models, Services, Utils og Views. Vi prøvde til en viss grad å ta i bruk en MVC-standard i prosjektet, da alle var kjent med det. Videre har vi i tilfeller hvor mange klasser deler funksjonalitet, opprettet Util-klasser hvor vi samler lignende metoder. Figur 12 viser klasse strukturen i Android Studio.



Figur 12 Fremvisning av filene i prosjektet i Android Studio

2.11.2 JAVADOC

En ting vi har lagt en del arbeid i, som ikke vi ikke får demonstrert hverken i dokumentasjonen eller i applikasjonen, er kommentering av kode. Her har vi fulgt JavaDoc-standarden, som figur 13 viser, både for vår egen del og det eventuelle teamet som skal jobbe med denne løsningen i ettertid. Det kan derfor genereres en wiki-lignende html-side med kommentarene som finnes over hver metode. Her har vi sørget for å dekke alt som ikke er grunnleggende get og set-metoder. I et prosjekt som ble såpass langt, var det for de fleste medlemmene til stor nytte selv om ingen ble syke eller forsvant i lange perioder.

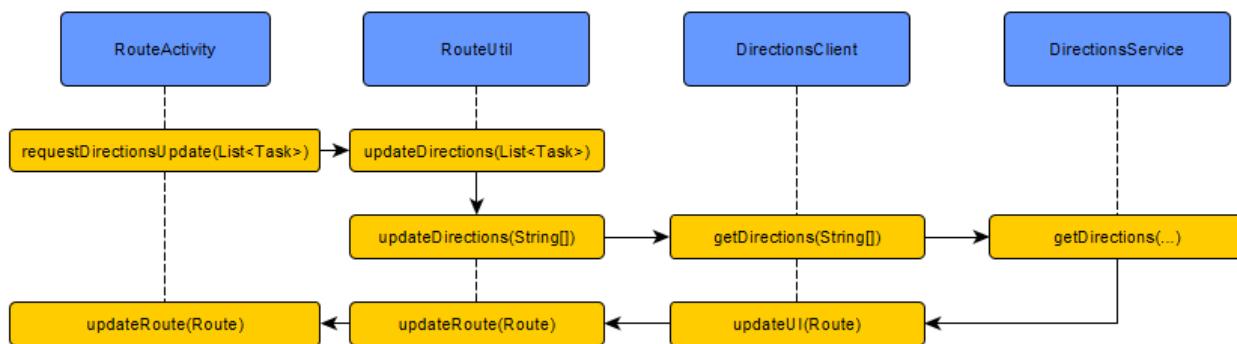
```
/**  
 * Creates a circular animation for the invisible view passed in, from the button passed in.  
 * @param floatingActionButton the button to press when animating  
 * @param ripple the ripple we want to animate over the camera  
 * @param exitButton sets an button visible after animation - optional  
 */  
public static void circularRevealCamera(ImageButton floatingActionButton, final View ripple,  
    CameraFragment fragment, final FrameLayout layout, final ImageButton exitButton){
```

Figur 13 Eksempel på hvordan JavaDocs fungerer

2.11.3 REFAKTORERING

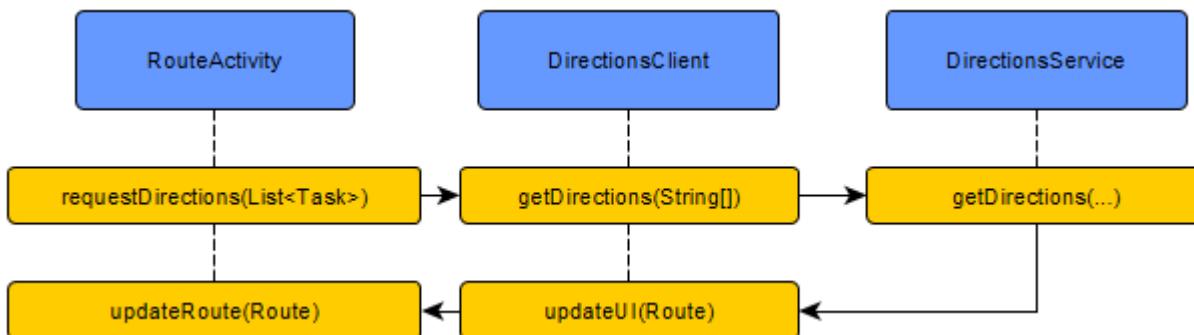
Når man lager et førsteutkast av kode er det vanlig å lage kode som i første omgang bare fungerer. Det er av og til litt mindre fokus på at koden er veldig bra eller skalerbar. For oss var det viktig å ha god kode, så vi var veldig strenge med oss selv at ting ikke bare skulle fungere, men det skulle være lesbart og kunne bygges på uten å komplisere flyt og logikk unødvendig mye.

Et eksempel på en større refaktorering vi gjorde var kallet for Google Directions, som hadde blitt noe komplisert med tiden. Her gjør vi et kall til Google og får svar tilbake med en utregnet rute. Flyten i kall-rekken gikk gjennom tre forskjellige klasser, og flere metoder var ikke alltid navngitt korrekt i forhold til funksjonalitet. Litt forenklet så dette slik ut(Firgur 14):



Figur 14 Original oversikt over flyten til directions-kallet

For å forenkle flyten til noe mer skalerbart fjernet vi RouteUtil fra flyten ettersom dette var en hjelpeklasse og ikke en hovedklasse. Vi flyttet også alle parse-operasjoner slik at `requestDirections` i RouteActivity fikk ansvar for forespørselen. Den nye flyten ble slik(Figur 15):



Figur 15 Oppdatert oversikt over flyten til directions-kallet

Etter denne refaktoreringen ble oversikten mye tydeligere og det ble enklere å håndtere resultatet fra Google Directions. Utbyttet av denne faktoreringen var at det ble enklere å bygge på ny funksjonalitet rundt oppdatering av kartet, uten at man risikerte å ødelegge noe annet i appen. Ikke minst sparte vi tid i det lange løp, da vi ikke måtte bruke like lang tid for å sette oss inn i systemet.

3. PRODUKT

3.1 MÅLGRUPPE

Produktets målgruppe er sjåførene til PostNord. I første omgang er det hovedsakelig de sjåførene som tar seg av siste ledd av transporten, fra terminal til butikk/mottaker.

Ved eventuelle videre utvikling av appen, kan dette utvides til alle ledd i transporten til PostNord, og til butikker som tar imot og lever pakker.

Appens hovedoppgave er å forenkle hver sjåførs arbeidsdag. Tanken er at appen skal organisere arbeidsdagen fra start til slutt, og fungere som et verktøy til alle arbeidsoppgaver sjåføren skal gjennom den dagen.

3.2 TEKNOLOGIVALG

Det ble i prosjektet brukt en del biblioteker og moduler av funksjonalitet vi bygde videre på for å oppnå ønsket resultat. Her nevner vi de viktigste og største bidragsyterene.

3.2.1 ANDROID

Android er et open-source operativsystem utviklet av Google som er lett å utvikle for, siden det i hovedsak baserer seg på Java for programkode og XML for design. Det er også et stort miljø for denne type utvikling på nettet.

Android er det et par superklasser som er særlig viktige, som Activity, Fragment og Adapter. Disse står for størsteparten av interaksjonen mellom kjernefunksjonaliteten og bruker-input.

- Activity fungerer som et skjermbilde i applikasjonen. Layouten settes med en XML-fil. Det er vanlig å ha en Activity for hver skerm i applikasjonen.
- Layout er en XML-fil som definerer hvordan et skjermbilde konkret skal se ut på skjermen, og hvilke elementer som skal inkluderes.
- Fragment fungerer på nesten samme måte som en Activity, bortsett fra at et Fragment må være inni en Activity. Man kan se på et Fragment som en seksjon av skjermen, og derfor kan en Activity ha flere Fragments. Man knytter stort sett en XML-fil til hvert Fragment.
- Adapter brukes for å knytte funksjonaliteten mellom layout/design til programkoden bak. Et adapter kan ligge inni både Fragment og Activity, alt etter hva som er ønskelig.

Slik får blant annet listene i programmet fortalt Android hva som skal skje ved forskjellige hendelser, som når et element blir trykket på eller noen andre data endres.

3.2.2 ANDROID LOLLIPOP

Valget om å bruke Android Lollipop, altså versjon 5.0 av Android, ble tatt ganske tidlig i prosessen. Vi hadde hørt mye om hvordan fokuset i denne utgaven var på animasjoner og et mer helhetlig utseende³. Dette sto helt i tråd med hvilke verdier vi ville fokusere på i utviklingen. I teorien får vi mye gratis animasjonsmessig med Lollipop, noe som ble fristende. Det skulle vise seg at teknologien fortsatt er veldig ny, og har en del spebarnsproblemer som det ikke var mulig for oss å fikse. I begynnelsen av prosjektet var det bare en håndfull spørsmål på nettet angående de nye funksjonalitetene, som gjorde at vi måtte improvisere en del, og falle tilbake på de eldre måtene å gjøre oppgaver som egentlig skulle gått forttere i Lollipop. Vi føler dog at vi sitter igjen med mange positive ting som følger av valget vi tok, spesielt transitions mellom aktiviteter i applikasjonen. Her sørger Android for å selv konstruere en animasjon basert på korte instruksjoner.

Det kan lett defineres hvordan animasjonen skal se ut slik som dette:

```
public static Transition getSlide(int gravity, View target){  
    Slide s = new Slide(gravity);  
    s.addTarget(target);  
    s.setInterpolator(new AccelerateDecelerateInterpolator());  
    return s;  
}
```

Etterpå trenger man bare å la et vindu sette dette som enten exitTransition eller enterTransition. For bildeeksempel se vedlegg 6.8.

Det finnes også animasjoner for delte elementer mellom aktiviteter. Det vil si at vi kan skape illusjonen av at et element forandrer størrelse og posisjon, men det vil alltid fremgå tydelig hvilken funksjonalitet det har, og hva brukeren kan forvente.

³ <http://www.android.com/versions/lollipop-5-0/>

Ved å definere et enkelt felt på elementet som deles mellom de to aktivitetene i layout-xml, vil Android selv konstruere en standard animasjon som endrer størrelse og posisjon.

```
<ImageButton  
    android:id="@+id/continueButton"  
    android:transitionName="buttonTransition"  
/>>
```

Om flere elementer skal endre posisjon synkront, og med forskjellige effekter, blir det noe mer komplisert, men prinsippet er det samme. Her skiller det mellom ContentTransitions og SharedElementTransitions.

ContentTransitions vil si animasjoner inn og ut av skjermen for skjermelementer. Ved å definere et TransitionsSet for overgangen mellom aktiviter kan du velge hvilke elementer som skal være med i overgangen, og hvordan de skal animeres. Det kan gjøres i XML, eller i java på denne måten:

```
TransitionSet transitionSet = new TransitionSet();  
transitionSet.addTransition(getSlide(Gravity.RIGHT, view1));  
transitionSet.addTransition(getSlide(Gravity.LEFT, view2));  
getWindow().setExitTransition(transitionSet);
```

Dette vil gjøre at view1 sklir ut til høyre, og view2 sklir ut til venstre når aktiviteten lukkes.

SharedElementTransitions håndterer, som navnet tilsier elementer som deles mellom to aktiviteter. Her kan du legge inn alle delte views for en såkalt Scene Transition, som er en overgang mellom to "scener", eller tilstander, for appen. Dette lagres i en ActivityOptions med metoden makeSceneTransitionAnimation. Dette ActivityOptions-objektet blir så sendt med når den neste aktiviteten startes. Som her:

```
ActivityOptions options = ActivityOptions.makeSceneTransitionAnimation(this,  
    Pair.create(view1, "trans1"),  
    Pair.create(view2, "trans2"));  
    );  
ActivityCompat.startActivity(this, intent, options.toBundle());
```

Dette vil føre til at view1 og view2 vil endre først og fremst posisjon, men i noen tilfeller også form og størrelse, til de matcher viewene med det oppgitte TransitionsName i destinasjonsaktiviteten.

3.2.3 SIGNATUR

Det er ingen funksjonalitet direkte i Android som lot oss lage en signaturløsning fra bunnen av. Vi fant imidlertid en bibliotek for et signatur-view, som heter Android Signature Pad⁴, og det var rett fram å implementere. Biblioteket inneholder et view som har en lytter for touch på skjermen. Signaturen som vises er basert på bezier-kurver for å lage en litt mer naturlig look på selve signaturen. Vi ville ha en minimumslengde på en signatur for å la den bli godkjent, slik at det ikke er nok å bare tegne en prikk for eksempel. Dette fantes ikke i "Android Signature Pad", så vi lagde en metode som regner ut størrelsen på signaturen før den godkjennes.

3.2.4 MAPS/DIRECTIONS

Vi visste tidlig at vi ville ha et kart som viste kjøreruta til sjåføren. Det åpenbare valget var å bruke Google Maps⁵ og Directions⁶ for dette. Dette er to API fra Google som tilbyr kart- og rutevisning i blant annet Android. Google Maps viser i hovedsak kun kartet og eventuelle markører man skulle plassere der. Google Directions genererer en rute fra et startpunkt til et slutt punkt, med mulighet for flere valgfrie mellomstopp som kan optimaliseres for å finne den mest effektive ruta mellom start- og slutt punktet. Vi får en liste med rutekoordinatene, og vi kan tegne rett på kartet med linjer mellom disse punktene. Optimaliseringen fra Directions gjorde at vi kunne regne ut den mest effektive kjøreruten for sjåføren.

Før vi kan gjøre et kall til Google Directions trenger vi to ting. Vi trenger adressene vi skal levere til, og vi trenger å fortelle applikasjonen hva den skal gjøre med resultatet fra Google Directions.

⁴ <https://github.com/gcacace/android-signaturepad>

⁵ <https://developers.google.com/maps/documentation/android/>

⁶ <https://developers.google.com/maps/documentation/directions/>

Callback-objektet sørger for at vi kjører metoden updateUI() med resultatet som parameter dersom alt går bra, hvis ikke vises en dialog for feilmeldingen:

```
Callback<DirectionResponse> callback = new Callback<DirectionResponse>() {  
    @Override  
    public void success(DirectionResponse response, Response status) {  
        updateUI(response);  
    }  
};
```

Vi sender en forespørsel via DirectionsService der vi sender med callback-objektet, og startadresse, sluttadresse og alle mellomstoppadressene:

```
RestAdapter restAdapter = new RestAdapter.Builder()  
    .setEndpoint("https://maps.googleapis.com/maps/api/directions")  
    .build();  
  
DirectionsService directionsService = restAdapter.create(DirectionsService.class);  
directionsService.getDirections(origin, destination, waypoints, "no", callback);
```

Da kaller vi på DirectionsService som er et interface, og ser slik ut:

```
public interface DirectionsService {  
    @GET("/json")  
    void getDirections(@Query("origin") String origin, @Query("destination") String  
    destination, @Query("waypoints") String waypoints, @Query("language") String  
    language, Callback<DirectionResponse> callback);  
}
```

Fra Callback-objektet kjøres metoden updateUI() når vi har fått svar fra Google. Derfra oppdaterer vi kart, databaser og annet som er avhengig av kartet.

3.2.5 ZBAR

Den gamle håndterminalløsningen hadde allerede funksjonalitet for å scanne strekkoder. Siden vi ville matche denne funksjonaliteten, og gjerne utvide den, valgte vi å bruke kameraet på nettbrettet som scanner. Det mest kjente biblioteket for kodescanning med kamera er ZXing (ZebraCrossing). Dette biblioteket er ikke optimalisert for Android-kameraet, men Zbar⁷ er et annet bibliotek basert på ZXing som er laget for nettopp Android. Zbar bruker kameraet til å scanne alle typer strek- og QR-koder. Vi opplevde at Zbar alltid scannet hurtigt og nøyaktig.

Scanneren initialiseres med følgende kode:

```
BarScannerView scannerView = new BarScannerView(this);  
scannerView.startCamera();
```

Metoden onPreviewFrame blir kalt hver gang kameraet oppdaterer seg. Her kan vi sjekke om resultatet fra scanningen er en kode, og sende den videre:

```
@Override  
public void onPreviewFrame(byte[] data, Camera camera) {  
    Image barcode = new Image(width, height, "Y800");  
    barcode.setData(data);  
    int result = scanner.scanImage(barcode);  
    if(result != 0)  
        resultHandler.handleResult(rawResult);  
}
```

⁷ <https://github.com/dm77/ZBarScanner>

3.2.6 SQL / SUGAR ORM

På grunn av at dette var et prosjekt som handlet mest om frontend, var vi i starten usikker på om vi i det hele tatt trengte en intern database, da vi var kjent med hvor plagsomt det var å utvikle for SQLite på Android. Det ble dog tidlig klart at dette var en naiv tilnærming, og at vi hadde stort behov for dette.

På grunn av negative erfaringer med utvikling av databaser fra scratch i Android, fant vi Sugar ORM⁸, som er et objektrelasjonelt framework på toppen av SQLite. Dette lot oss lage relativt store databaser med mange objektrelasjoner som lett ble håndtert i koden. Her slipper vi å bekymre oss for å åpne eller lukke koblingen til databasen, og vi får en del av standard CRUD-funksjonalitet gratis.

Her vil vi ikke gå i detalj på hvordan koden ser ut om man skal lage alt fra bunn, da det kan sies å være noe av det mest verbose innenfor Android-utvikling.

Men i Sugar ORM trengs det kort sagt bare å extende en superklasse på formen:

```
public class Package extends SugarRecord<Package> { ... }
```

Herfra vil Sugar ORM legge ved alle felter i objektet som relasjoner eller poster i databasen.

Sugar ORM har ikke støtte for en-til-mange relasjoner, og man må derfor bygge opp dette litt kreativt. En mulighet er at mange objekter har en referanse, eller id, til sin parent. En get-metode av alle Packages til en gitt Task-post vil derfor kunne se slik ut:

```
public List<Package> getPackages() {
    return Package.find(Package.class, "task = ?", String.valueOf(this.getId()));
}
```

Vi finner altså alle packages med taskId = this.getId().

Dette er absolutt ikke optimalisert for effektivitet, og ville ikke fungert på et større system, men for et mindre internt system som kun forholder seg til seg selv, er ikke tregheten med store mengder I/O og store søk et problem.

⁸ <http://satyan.github.io/sugar/>

3.2.7 ANDROIDANNOTATIONS

Vi har hatt et fokus på lesbarhet av kode i prosjektet. Dette for vår egen del, og om det eventuelt skal brukes noe til videreutvikling. Med AndroidAnnotations⁹ har vi kunnet unngå mye av koden som trengs for å koble view med controller og modell i Android. Dette inkluderer ting som onclicklistener på knapper for å utføre en handling, og mer avanserte ting som å fjerne oncreate-metoder helt, og lage asynkrone metodekall som bruker en ekstern tråd. Sistnevnte involverer i Java og Android ganske mye boilerplate-kode, og tar fokus bort fra essensen av metoden.

```
@OptionsMenu(R.menu.menu_task_list)  
@EActivity(R.layout.activity_tasklist)  
public class TasklistActivity extends RouteActivity{ ... }
```

Fra bare disse to linjene vil vi generere en kobling til actionBar, og all elementær kode som trengs for å starte en Activity i Android. Videre vil funksjonalitet for menyen lett defineres slik.

```
@OptionsItem(R.id.quickDeliver)  
void quickDeliver(){ ... }
```

Bare det å sende med Extras mellom aktiviteter i Android krever vanligvis mye mer enn man skulle trodd. Her må det spesifiseres konkret hvilken primitive type som skal sendes med, og hva denne heter, før den må hentes med akkurat samme verdier på den mottakende aktiviteten.

Koden går dermed fra denne snutten ved sending:

```
Intent intent = new Intent();  
intent.putExtra("route", extraRoute);  
startActivity(this, intent, null);
```

Og mottak:

```
Intent intent = getIntent();  
Route extraRoute = intent.getStringExtra("extraRoute");
```

⁹ <https://github.com/excilys/androidannotations/>

Til en enkelt linje ved sending:

```
ScanningActivity_.intent(this).extra("extraRoute", extraRoute).start();
```

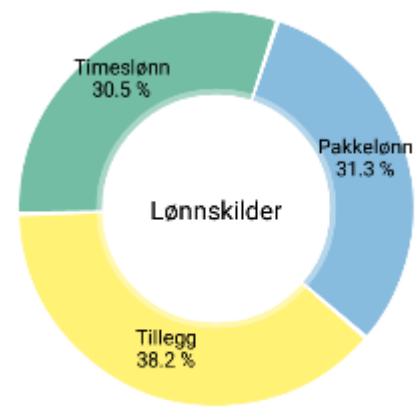
Mens for å ta imot denne intenten med en extra, trenger man kun å reference navnet i den andre aktiviteten slik:

```
@Extra  
Route extraRoute;
```

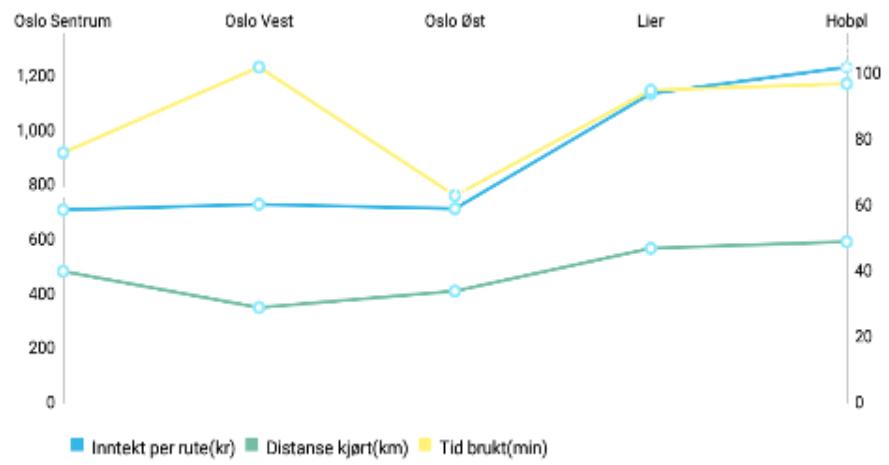
I disse tilfellene er det mest praktisk for utvikleren som skal skrive koden, men i alle andre tilfeller hjelper det teamet å tvinge frem en lettleselig standard, med færrest mulig gjentatte linjer. Koden som genereres bak dette er ganske vesentlig til og med i et såpass kort prosjekt som vårt. AndroidAnnotations genererer hele 3300 linjer kode ut ifra vår base på 11900 kodelinjer. Om man da er raus og sier det er lignende håndskrevet kode, kan det sies å ha minsket størrelsen på kodebasen med 27%.

Oversikten over linjeantall og distribusjon av dette finnes i vedlegg 6.10.

3.2.8 ANDROIDMP



Figur 16 Kakediagram brukt til brukerprofil



Figur 17 Graf til brukerprofil

Det er ingen gode diagrambiblioteker inkludert i vanlig Android. AndroidMP¹⁰ er et grafbibliotek for å visualisere statistikk. Vi implementerte det for å vise statistikken i brukerprofilen til sjåføren, som figur 16 og 17 viser. AndroidMP er særdeles fleksibelt og har mange flere bruksområder og funksjonaliteter enn det vi selv trengte det til. Rent kodemessig er det lite komplisert, og veldig mye følger typiske standarder for initialisering av objekter og views i Android. Om man skal ha det tilpasset sitt eget bruk må man regne med å skrive en del kode for dette.

¹⁰ <https://github.com/PhilJay/MPAndroidChart>

3.2.9 GRADLE

Gradle¹¹ er et byggeredskap og er den nye standarden for Android-prosjekter. Her kan man lett inkludere andre libraries ved å legge til en ekstra linje kode med navn og versjonsnummer.

```
compile 'com.android.support:recyclerview-v7:21.+'  
compile 'com.android.support:cardview-v7:21.+'  
compile 'com.github.PhilJay:MPAndroidChart:v2.0.9'
```

Dette vil deretter synces opp mot MavenCentral som er en online database over mange kjente prosjekter. Dette gjør at man slipper å inkludere rå .jar-filer i prosjektet, og man blir mye mindre avhengig av å sette opp ting lokalt på sin egen maskin. Istedet kan man konfigurere hvordan prosjektet skal se ut for alle som åpner det, uavhengig av hva man har installert på forhånd. Gradle's Groovy er da et slags metaspråk for å definere prosjektet i denne sammenhengen.

I korte trekk kunne vi hente inn biblioteker fra andre steder med en enkelt kodelinje, istedenfor å manuelt måtte laste dem ned og importere dem gjennom en lang og tungvint prosess. Alle i gruppen er enig om at dette er en god ressurs for å slippe å bruke tid på unødvendig arbeid.

Man definerer også Androids laveste nivå-settings her, slik som:

```
defaultConfig {  
    applicationId "no.mesan.handterminator"  
    minSdkVersion 21  
    targetSdkVersion 21  
}
```

¹¹ <https://gradle.org/>

3.3 SAMSVAR MELLOM KRAVSPESIFIKASJON OG ENDELIG PRODUKT

Som nevnt tidligere, under punkt 2.9, *Utfordringer og avgjørelser*, var vi nødt til å gjøre en del justeringer underveis, av ulike grunner. Noe funksjonalitet valgte vi å utelate når vi begynte å få en følelse for hvordan flyten og brukerfølelsen i appen ble, og det rett og slett det ikke passet inn i helheten. Andre ting rakk vi ikke på grunn av tidsrammen for prosjektet, mens noe måtte droppes fordi det viste seg å være vanskeligere enn antatt, eller fordi vi fant en bedre eller enklere løsning. Alt i alt føler vi likevel produktet som helhet svarer forholdsvis godt til kravspesifikasjonen, i form av at den besvarer de krav og utfordringer som stilles, og inneholder det aller meste av funksjonalitet. Hovedsakelig er endringene i positiv retning, enten i form av bedre funksjonalitet i sluttproduktet, eller ved at implementasjonen er enklere og mer oversiktlig i kode.

Vi holdt gjennom hele prosjektet en løpende dialog på status for de ulike delene av løsningen, og hvorvidt disse burde droppes, satses videre på eller endres, i tråd med vår smidige utviklingsmodell.

De mest merkbare forskjellene mellom den opprinnelige kravspesifikasjonen og det endelige produktet er nok fraværet av punktet "Hjelp til lasting ved terminal", altså optimalisering av ruten før innlasting, slik at pakkene kan lastes inn i bilen, etter når de skal leveres. Første levering, sist inn. Dette høres ut som god hjelp i utgangspunktet, men ettersom vi visste såpass lite om hvordan scanning og innlastingsprosessen foregår, konkluderte vi med at dette ikke var verdt å satse på.

Ellers er det design-endringer fra de opprinnelige skissene og mindre endringer i brukergrensesnittet, som har endret seg mest siden starten av planleggingen. Eksempler på dette er funksjonaliteter i arbeidslista, utforming av ad-hoc-håndtering og visning av ruter i rutelista. Arbeidslista skulle opprinnelig støtte swiping for å slette pakker og rapportere avvik. Sletting ble til slutt fjernet og avvik ble lagt i leveringsvinduet isteden. Ad-hoc-dialogen ble gjort bygd helt fra bunnen av, istedenfor å bruke standard dialogvinduer. Dette ga bedre funksjonalitet og et finere design. Visning av ruter i rutelista startet som en oppramsing av kolli, men utviklet seg til å bli visning av egen ruteinfo, med en forhåndsvisning av ruta på kart. Dette er tre eksempler på avvik fra kravspesifikasjonen, og vi føler alle tre tilførte appen noe nytt og bedre. Dette viser at vår smidige metodikk fungerer, og gir et bedre resultat i lengden. Flere slike justeringer finnes som sagt under punkt 2.9.

3.4 DESIGNVALG

3.4.1 ANDROID STANDARDER

Google introduserte i juni 2014 en designstandard, *Material Design*¹², som i prinsippet skal gjelde som retningslinjer for visuelt design i Android. Material Design bruker mye rutebasert layout, responsive animasjoner, padding og dybdeeffekter som lys og skygge. Dette for å gi illusionen av at det som er på skjermen skal se og føles så ekte og naturlig ut som mulig. Google selv sier at Material Design er basert på papir og blekk, men forskjellen er at Material Design digitalt kan endre og tilpasse seg enhver situasjon. Material Design ble hovedsakelig utviklet for Android 5.0 (Lollipop), og design er også mye av hovedfokuset i Android 5.0. Derfor anser vi Material Design og Android 5.0-standarden som den samme.

Vi valgte å følge og bruke flere av retningslinjene til Material Design, siden det er utviklet for og av profesjonelle designere. Gevinsten av å bruke dette har vært bedre brukervennlighet samt et enkelt, men utfyllende, visuelt design.

3.4.2 FARGER

En viktig del av design er farger. Brukeropplevelsen kan gå fra fantastisk til forferdelig om fargevalgene kolliderer.

I moderne visuelt design er det veldig vanlig å bygge opp layouter med mye fokus på harmoni og balanse. Nærliggende farger på fargespekteret, eller f.eks. nyanser av samme farge (som rød og mørkerød), skaper ofte en rolig harmoni som er behagelig for øyet å se på. Rent stilmessig blir designet også mye penere.¹³

For å drive oppmerksomhet til et spesielt sted i designet er det veldig vanlig å bruke farger som er direkte kontraster (altså farger på motsatt side av fargespekteret) av hverandre, ofte kalt komplimenterende farger i visuelt design. For mange konkurrerende farger kan bli ubehagelig

¹² <http://www.google.com/design/spec/material-design/introduction.html>

¹³ <https://cios233.community.uaf.edu/design-theory-lectures/color-theory/>

å se på. Derfor er det viktig å bruke kontraster sparsommelig. Slike kontraster kan f. eks. være lyseblå og rød, eller grønn og lilla.

Hovedfargen vi valgte var blåfargen i PostNord-logoen. For å balansere designet valgte vi en litt mørkere blåfarge som nyanse. Vi valgte hvit til bakgrunn og tekst, som en litt tammere halvkontrast til begge fargene, som samtidig ikke drar mye fokus.

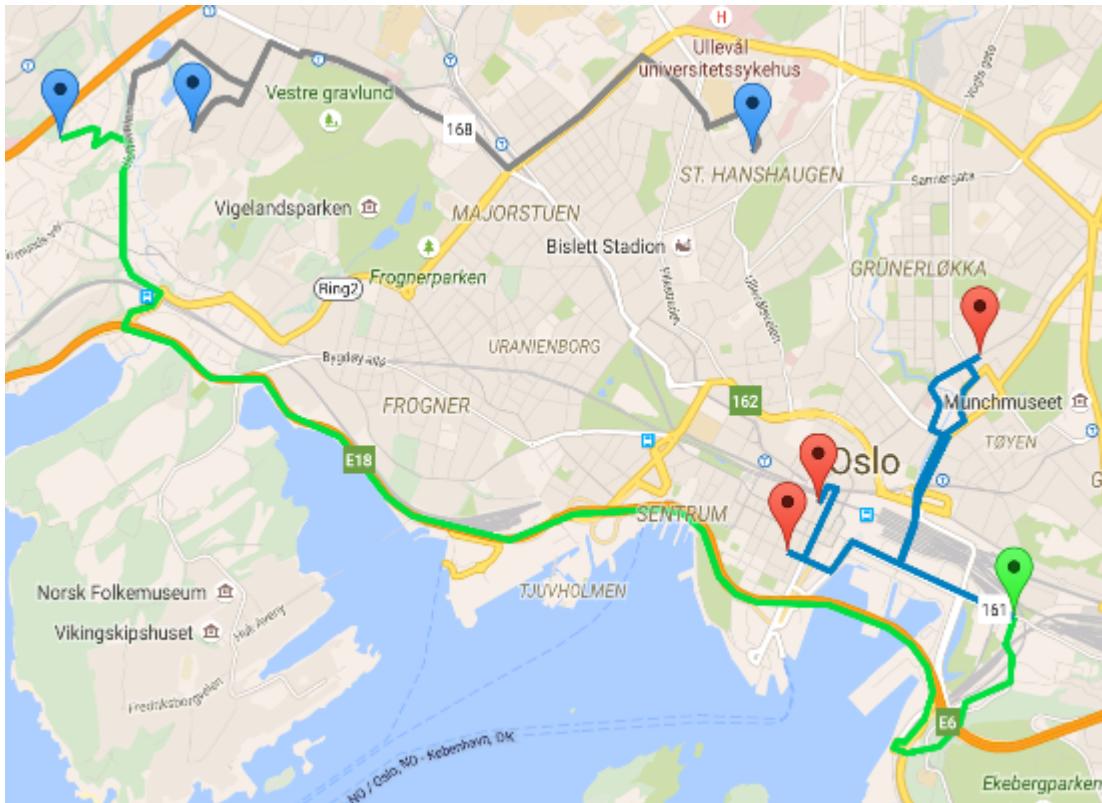
Floating Action Button er en sirkulær knapp for den viktigste funksjonaliteten der du er i applikasjonen. Denne måtte tydelig skille seg fra det hovedsakelig blå hovedtemaet. For å ha tydelig kontrast valgte vi en slags rosa/lilla-farge, som skilte seg fra alle andre farger brukt i appen. (Mer om Floating Action Button under 3.4.4) Alle hovedfargene i appen er vist i Figur 18.



Figur 18 Hovedfarger, mørk-nyanse, tekst, floating action button.

For å bekrefte innscanning brukte vi en relativt lys grønnfarge siden grønn er universelt brukt for bekreftelser.

Google Maps bruker sine egne farger som vi egentlig ikke kan gjøre så mye med, men den opp tegnede ruta og markørene valgte vi egne farger for.



Figur 19 Kart med eksempelrute

Vi delte opp den opptegnede ruta i tre deler, med hver sin farge(Demonstrert på figur 19):

- Tidligere leveringer, som allerede har blitt levert (grå). Skal egentlig bare fylle ut kartet, og trenger derfor lite oppmerksomhet. De har relativt liten kontrast i forhold til kartet og drar derfor lite fokus.
- Aktiv levering, som er den neste som skal leveres (grønn). Dette er den viktigste delen av ruten, og krever mye fokus. Grønnfargen har en veldig sterk kontrast til resten av kartet, og grønn er en farge som betyr " neste" eller " å gå videre".
- Fremtidige leveringer, som skal leveres senere (Blå). Det er viktig å se hvor man skal senere på ruta, og derfor er det viktig at de har litt, men ikke all fokus. Den har litt kontrast, men ikke like mye som grønnfargen til den aktive leveringen.

Adresser sjåføren skal innom viser vi med markører. Vi delte disse opp i de samme kategoriene som den opptegnede ruta:

- Tidligere leveringer (blå). En litt svakere blåfarge som drar relativt lite fokus.
- Aktiv levering (grønn). Veldig lik grønnfarge som den opptegende aktive ruta. Drar mye fokus ettersom kontrasten er betydelig.

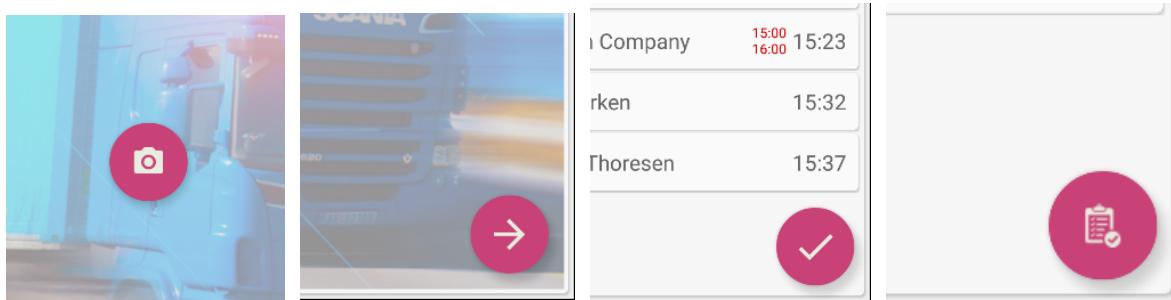
- Fremtidige leveringer (Rød). En relativt svak rødfarge som blander seg litt inn i rødtone til veiene fra Google Maps. Litt kontrast som drar litt oppmerksomhet, men som fortsatt ikke stjeler hovedfokuset fra den aktive leveringen.

3.4.3 NAVIGASJON

Flyten i programmet er lagt opp slik at brukeren skal føle det når han beveger seg "fremover" i flyten. Derfor skal alle interaksjoner som tar deg videre i flyten, være en bevegelse mot høyre, mens bevegelser bakover i flyten, er en bevegelse mot venstre. Dette er typisk Slide-transitions. Andre flytelementer vil være informasjon om bruker eller pakker. Disse vises som et infovindu, som ligger over den gjeldende skjermen, og dermed viser en midlertidig utstikker fra hovedflyten, mens hovedvinduet fortsatt synes bak. Slik vil brukeren få en klar og tydelig beskjed om hvor man er på vei, og hvor man befinner seg. Funksjonelt sett har ikke dette mye verdi, men for brukere som er vant til iOS og Android-applikasjoner slik mange er i dag, handler dette om å møte forventningene, som er opparbeidet gjennom standarder i bransjen.

3.4.4 FLOATING ACTION BUTTON

I Android 5.0 er det stort fokus på at det alltid skal være tydelig for brukeren hva hovedfunksjonaliteten i skjermbildet er. En Floating Action Button¹⁴, er i essensen bare en rund knapp med varierende funksjonalitet avhengig av hvor i programmet man er. Vi bruker det blant annet for å la bruker fortsette videre i programflyten når alle andre krav er tilfredsstilt. I tillegg må brukeren scanne inn pakker på forskjellige tidspunkt, og her vil knappen endre ikon til et kamera, og åpne et kamera for scanning ved trykk. Når kravene er innfridd endrer knappen igjen funksjonalitet. Fargevalget for knappen er av betydning ettersom vi vil at den klart og tydelig skal skille seg fra resten av palletten. Ikke minst er plasseringen viktig. Vi ville gjerne plassere den i brytningene på kanten mellom forskjellige elementer på skjermen, men slik som designet endte opp ga det ikke mye mening. Den ble derfor plassert sentralt når man må gjennomføre noe, og nede i høyre hjørne når man kan gå videre i flyten. Eksepler på bruk av FAB er vist i figur 20.



Figur 20 Eksempelbruk av Floating action button

3.4.5 GOOGLE MAPS

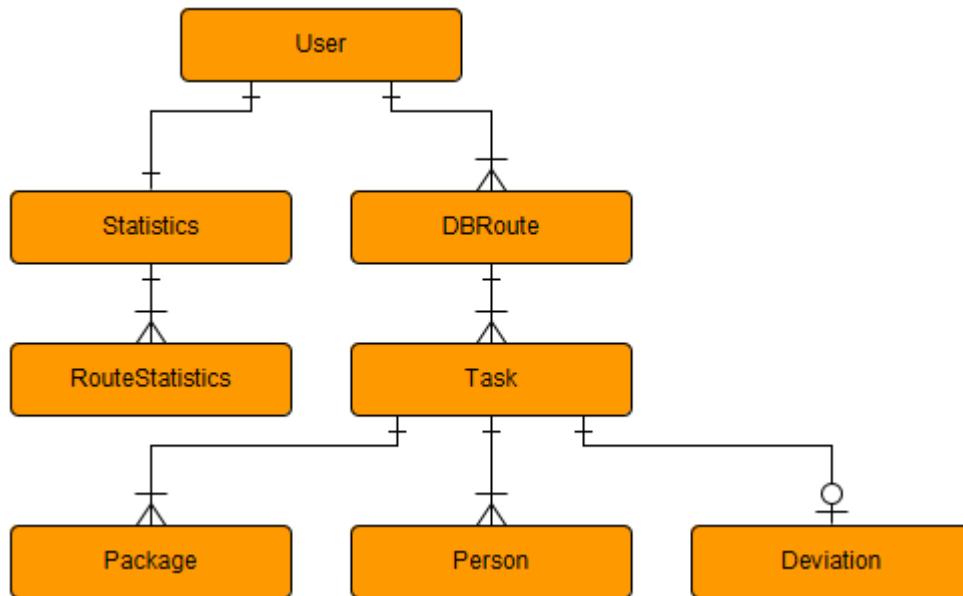
For å gi en følelse av hvor arbeidsoppgavene befinner seg i forhold til hverandre, og samtidig gi funksjonalitet for navigasjon til sjåføren, valgte vi å bruke Google Maps. Valg rundt farger for Google Maps står under punkt 3.4.2.

¹⁴ <http://www.google.com/design/spec/components/buttons-floating-action-button.html>

3.5 BACKEND

Appens backend er hundre prosent intern, siden vi ikke ville ha noen fordel av å kunne kommunisere med den eksisterende løsningen til PostNord i et proof of concept.

Med Sugar ORM ser derfor databasestrukturen omtrent slik ut(Figur 21):



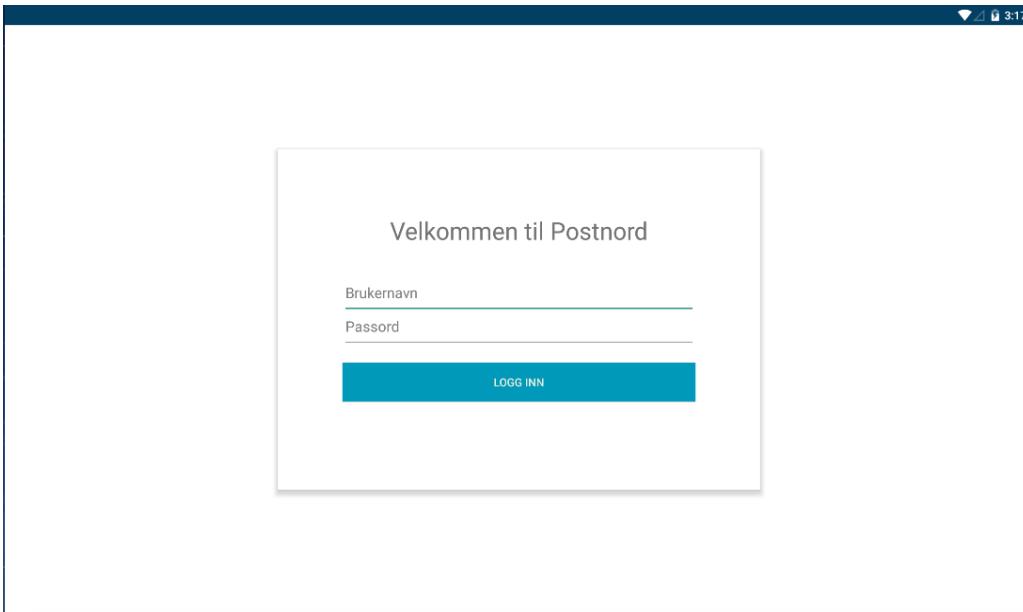
Figur 21 Database relasjonsdiagram i Sugar ORM

Skulle dette vært en realistisk løsning ville vi implementert en kommunikasjonsbro mellom den interne databasen i applikasjonen og PostNords sentralsystem, som i den grad det var mulig skulle overført informasjon over mobilt nett fra tabletten. Det er ikke alltid mulig å få tilgang på nett for alle sjåførene rundt omkring i landet, så det måtte også ha vært et robust system til å håndtere når dette burde pushes eller ikke. Dette er dog ikke problemstillinger vi har tatt spesielt hensyn til.

3.6 FERDIG LØSNING - SKJERMBILDER MED HOVEDOPPGAVER

Ettersom målet med appen var å forbedre og bygge på den gamle løsningen, valgte vi å beholde det viktigste av funksjonaliteten og implementere ny funksjonalitet vi antok ville hjelpe sjåførene i deres arbeidsdag. I essensen er det aller viktigste å ha en oversikt over aktuelle ruter, enkelt valg av rute, innscanning av pakker, oversikt over rutens arbeidsoppgaver og levering av pakker med signatur. I tillegg ville vi ha noe funksjonalitet for info og statistikk om bruker, og nødsituasjonshåndtering. Dette resulterte i følgende skjermbilder.

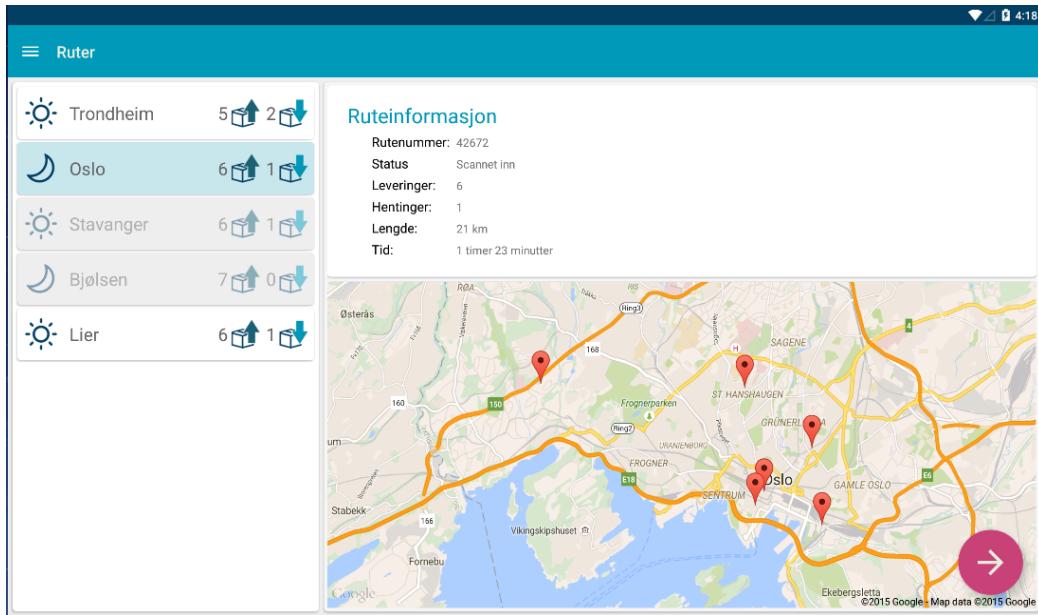
3.6.1 LOGIN



Figur 22 Innloggingsvindu

Innloggingsvinduet vårt er enkelt utformet, med en ren og enkel interface uten forstyrrende elementer. Funksjonaliteten er kunne logge inn med brukernavn og passord. Ved mislykket innlogging, vises en enkel feilmelding under knappen.

3.6.2 RUTELISTE

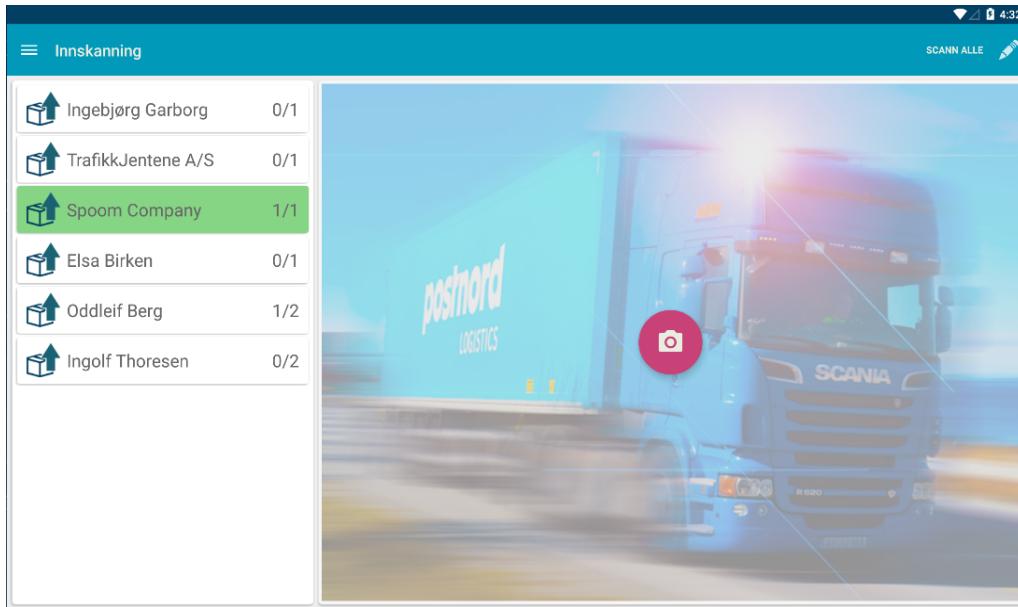


Figur 23 Rutelistevindu med oversikt over sjåførens ruter

Rutelisten er første skjermbilde i hovedflyten i appen (innlogging ekskludert). Hovedfunksjonaliteten her er å gi en god oversikt over de ulike rutene, vise utvidet info om en valgt rute fra listen, og muligheten til å velge en rute å gå videre med til innscanning.

Rutelisten skiller mellom dag- og nattskift med et ikon (sol/måne), og viser hvor mange hentinger og leveringer som finnes i hver rute. Ferdige ruter blir markert som grå i listen, og får også et "fullført"-stempel over kartet når ruten er valgt. Utvidet info omfatter en rute-ID, status for ruten, antall pakker og estimert kjøretid og avstand, samt et kart over alle addressene ruten inneholder.

3.6.3 SCANNING



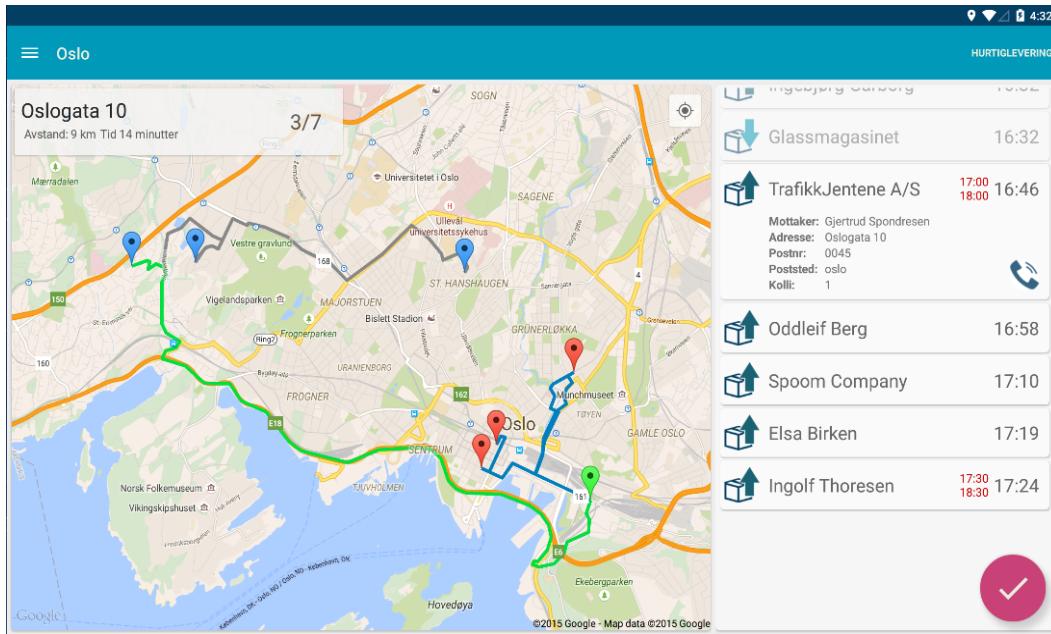
Figur 24 Innskanningsvindu med oversikt over rutens leveranser

Innskanningvinduet har som hovedoppgave å få scannet inn alle pakkene før opplasting i bilen. Skjermbildet er enkelt, og inneholder et stort kameravindu på høyreside av figur 24, og en liste over alle leveringer på venstre side.

Listen indikerer hvor mange kolli som er scannet for hver levering og skifter farge om alle kolli for leveringen er scannet inn. Scanning foregår via kameraet til høyre, som gjenkjenner strekkoder, inkludert QR-koder, når disse pekes på.

Vi har også her mulighet for "manuell innskanning", hvor bruker kan skrive inn pakkens kollinummer direkte, om pakkens strekkode er ødelagt. Manuell innskanning inneholder også autfullføring, hvor bruker kan velge kollinummer fra en liste med forslag.

3.6.4 ARBEIDSLISTE



Figur 25 Arbeidslistevindu med oversikt over rutens leveranser og hentingar på kart og i liste

Arbeidslisten er på mange måter det viktigste skjermbildet i appen. Det er her bruker vil bruke mest tid, da dette er vinduet som vises under kjøring, som jo er det mesteparten av en sjåførarbeidsdag består av. Hovedoppgavene her er å vise rutens hentingar og leveringer (oppgaver), samt kartvisning av ruten som helhet, og fremdrift i ruten på kart og i listen. I tillegg skal det være mulig å se detaljert info om de ulike oppgavene, kontakte mottaker eller kjøreleder, og omorganisere rutens rekkefølge etter behov. Bak kulissene foregår det også en automatisk optimalisering av ruten ved hjelp av Google Directions, oppgavene legges i optimal rekkefølge for å få levert og hentet alle pakker fortest mulig.

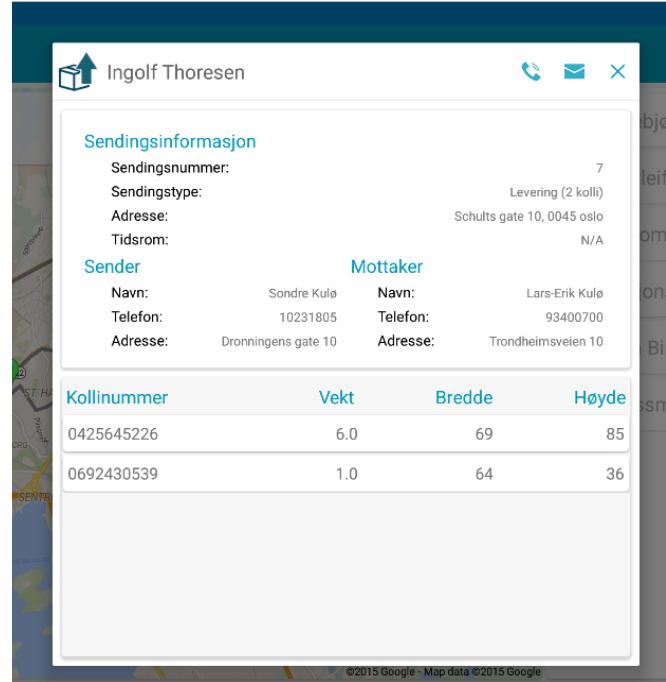
På venstre side vises altså et kart over hele ruten. Her vises ferdige oppgaver og veistrekninger med blå markører og grå linje, aktiv oppgave med grønn markør og linje, og kommende oppgaver med rød markør og blå linje (mer om fargevalg under Farger punkt 3.4.2). Dette viser tydelig fremdriften i ruten, og vi kan også se sjåførens nåværende posisjon ved hjelp av GPS. Øverst til venstre vises neste oppgaves adresse, samt estimerer for avstand og tid til ankomst.

I arbeidslisten til høyre vises alle rutens oppgaver i en liste, med estimert ankomst. Noen ruter har i tillegg et satt tidsrom, som pakken skal leveres i (f.eks mellom 12 og 14), som vises i rødt,

ved siden av ankomstestimatet. Den aktive oppgaven vil i tillegg være utvidet, og viser noe utvidet info. Her har også bruker mulighet for å enkelt ringe til mottaker. Fullførte oppgaver er også her grået ut. Ved klick på en oppgave i listen åpnes et detaljvindu med utvidet info om denne oppgaven.

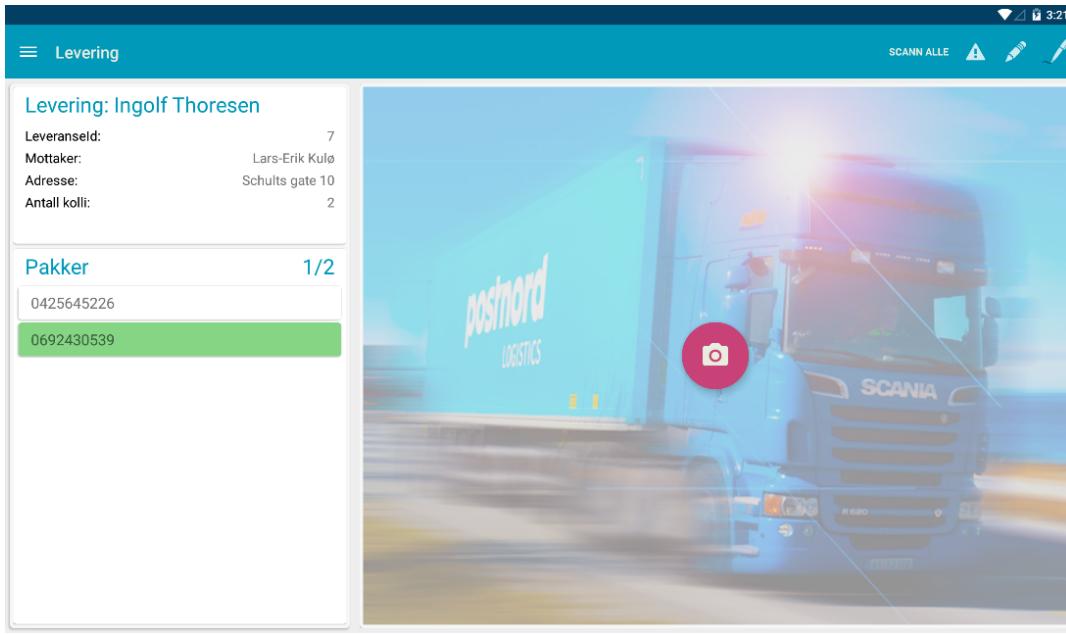
3.6.4.1 PAKKECARD

Infoinduet for oppgaver, kalt pakkecardet, viser detaljert info om valgt oppgave, inkludert kontaktinfo for sender og mottaker og en liste over kolli, med utvidet info om disse også.



Figur 26 Pakkecard med detaljert informasjon om en leveranse

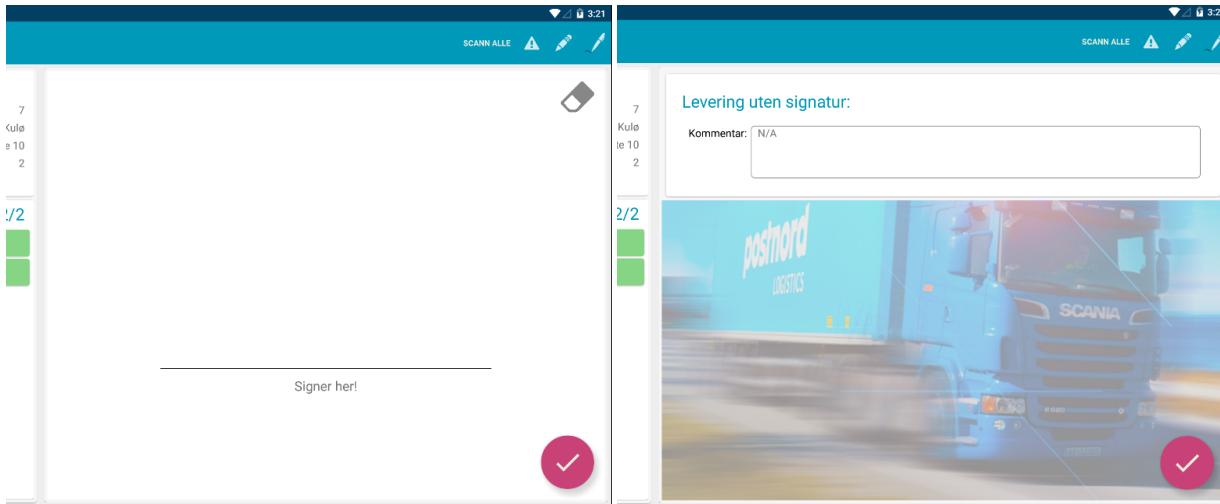
3.6.5 LEVERING



Figur 27 Leveringsvindu med knapp for å åpne kamera

Leveringsvinduet er det siste skjermbildet i hovedflyten vår. Her håndteres levering og henting av pakker hos mottaker/sender. Hovedfunksjonaliteten er utscanning av pakker, lagring av signatur og rapportering av avvik, hvis noe står i veien for at pakken kan leveres på vanlig måte.

3.6.5.1 Signatur

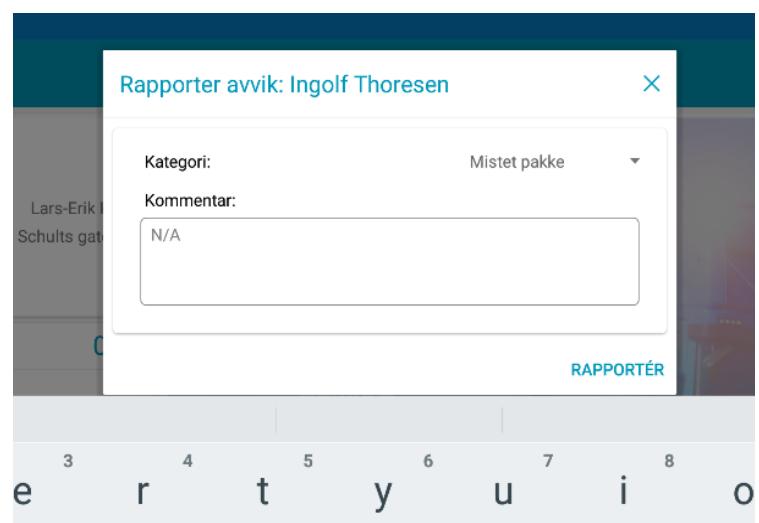


Figur 28 Signatursområde i leveringsvindu (V), kommentarfelt i leveringsvindu (H)

Etter at pakkene er scannet, skal bruker motta signatur fra mottaker. Dette gjøres også ved at kameravinduet gjøres om til et signaturvindu. Her kan bruker signere direkte på skjermen ved hjelp av touch. Her ser vi for oss at en touch-penn benyttes, men det fungerer også fint å bruke fingeren. Det er også mulig å levere uten signatur, men heller med en kommentar ed begrunnelse for dette valget.

3.6.5.1 AVVIK

Rapportering av avvik (f.eks: skade på pakken) foregår i et eget skjermbilde som er utformet som en popup, eller dialog, som vises over leveringsvinduet. Det åpnes ved å trykke på avvikikonet i actionbar. Her kan den velges en kategori for avviket, og legges inne kommentar for å beskrive situasjonen nærmere. Rapportering av avvik fullfører også oppgaven.

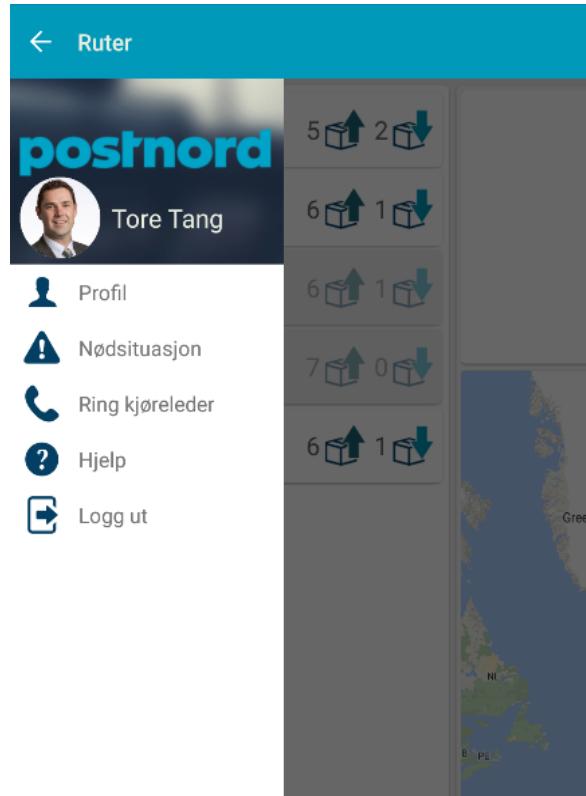


Figur 29 Vindu for avvikrapprtéring

3.6.6 NAVDRAWER

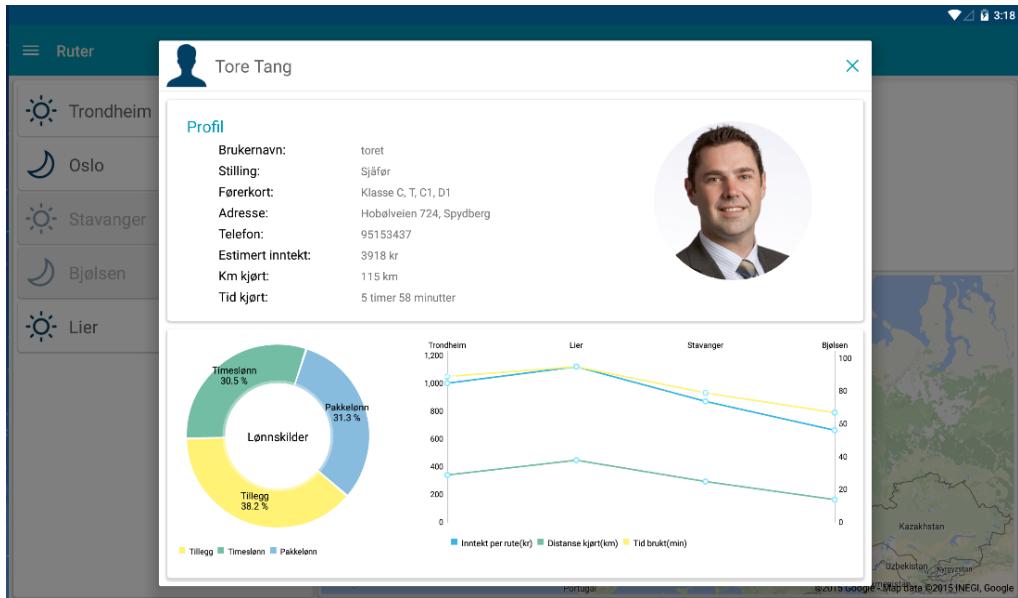
Navigasjonsmenyen er et element som følger appen vår gjennom alle skjermbildene (utenom login). I denne menyen har bruker tilgang til en rekke funksjoner, som skal være tilgjengelig fra hvor som helst i flyten. Hovedoppgaven til navigasjonsmenyen er å gi bruker tilgang til viktige funksjoner som hjelpe, utlogging, og nødsituasjonshåndtering. Menyen fungerer som en drawer som kan dras inn fra siden, og legger seg over det aktive skjermbildet.

Menyens inneholder visning av profilvindu for innlogget sjåfør (se under), visning av nødsituasjonsskjerm, oppringning av kjørelder for å håndtere ulike usikkerheter, visning av hjelpefunksjoner (for øyeblikket en bruksveiledning i form av overlays på skjerm), og utlogging.



Figur 30 Skuffmeny med menyvalg for blandt annet brukerprofil

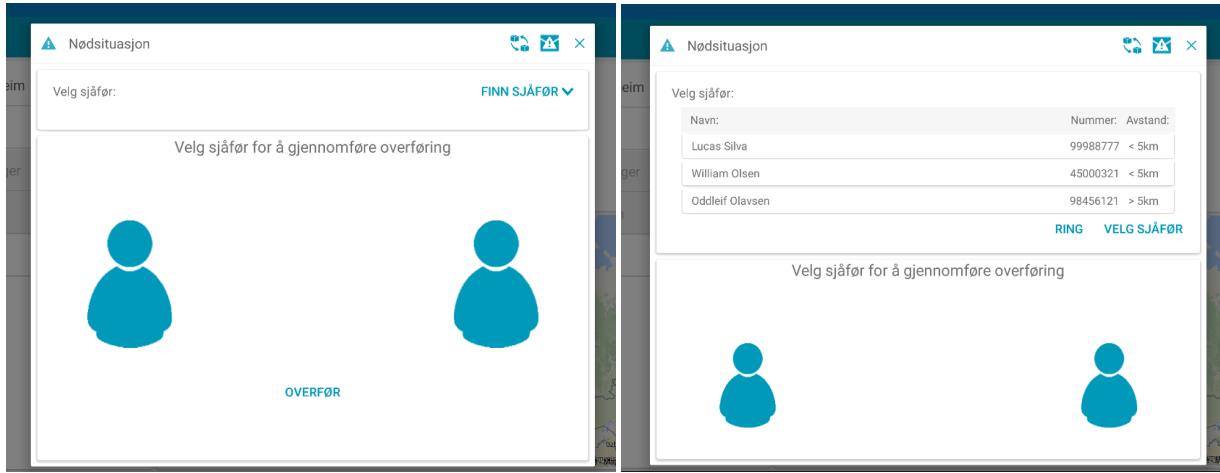
3.6.6.1 PROFILVINDU



Figur 31 Profilvindu som viser informasjon og statistikk om bruker

Som en del av kravspesifikasjonen, fikk vi i oppgave å kunne vise info om innlogget bruker med statistikk for arbeidstid, kjørelengde og lønn. Vi valgte å lage et eget vindu hvor diverse info vises, sammen med bilde av brukeren, og statistikk i form av grafer og diagrammer. Her får bruker oversiktlig og enkelt presentert statistikk på en visuell måte, som gjør det lett å se nåværende status og utvikling over tid.

3.6.6.2 PAKKEOVERFØRING



Figur 32 Nødsituasjons vindu for overføring av pakker til andre sjåfører

Vi har for øyeblikket implementert to nødsituasjonsskjærmer. Den første gjelder rapportering av avvik for hele bilen eller ruten, altså ikke knyttet til en enkelt levering eller pakke, med en kommentar som beskriver avviket. Den andre gjelder pakkeoverføring, hvor en sjåfør kan kontakte en annen sjåfør og overlevere sine pakker til ham. Hovedoppgaven i dette skjermbildet er å enkelt finne sjåfører i nærheten, og velge en som pakkene skal overføres til. Pakkeoverføring er bare simulert i appens nåværende tilstand, men vil enkelt kunne implementeres i en endelig løsning. Vi ser for oss at vi her vil kunne overføre alle pakker fra den ene sjåførens rute, over på den andres. Eller en og en pakke, når det skulle være aktuelt.

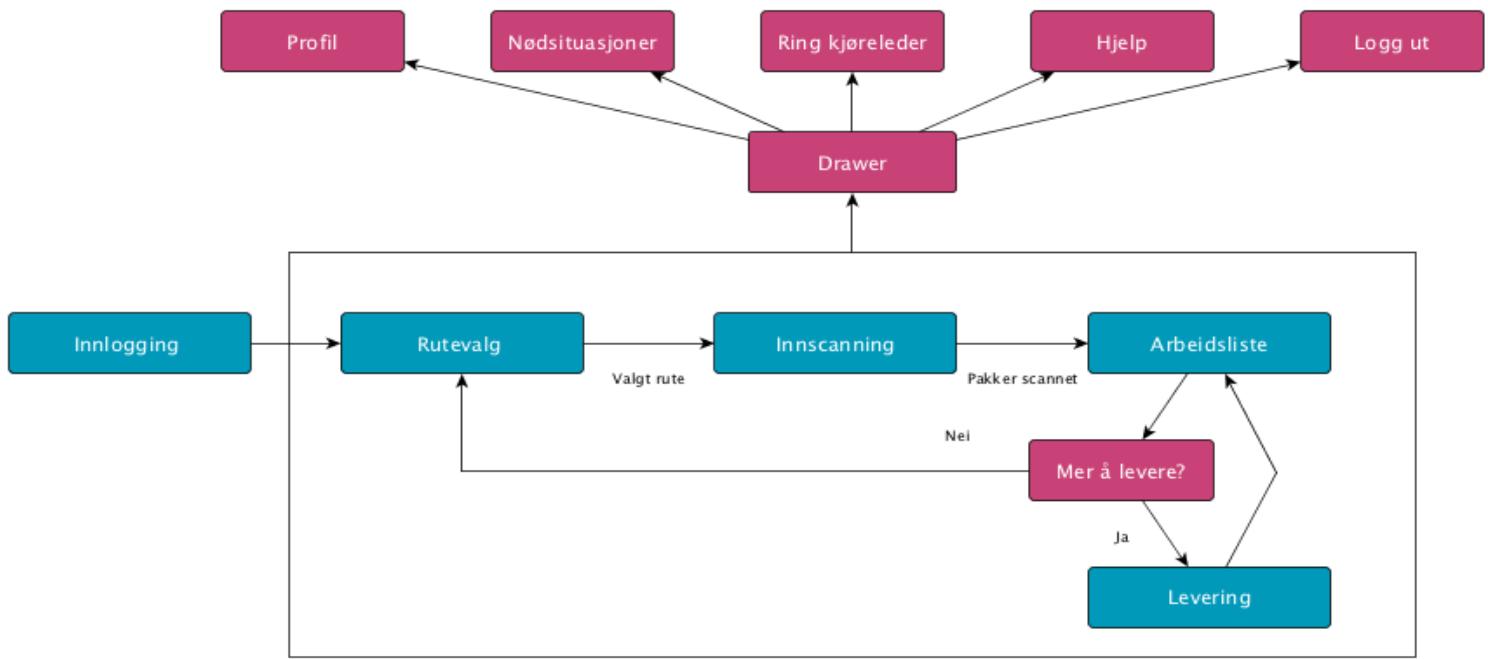
3.7 FLYTDIAGRAM

En av våre hovedfokus i dette prosjektet var å forenkle den flyten til PostNords eksisterende håndterminal. Denne løsningen bestod av rundt 40 skjermbilder, og var ganske komplisert og kaotisk. Vi analyserte oss frem til de hovedfunksjonene vi mente burde være med, og slo sammen/fjernet det vi kunne.



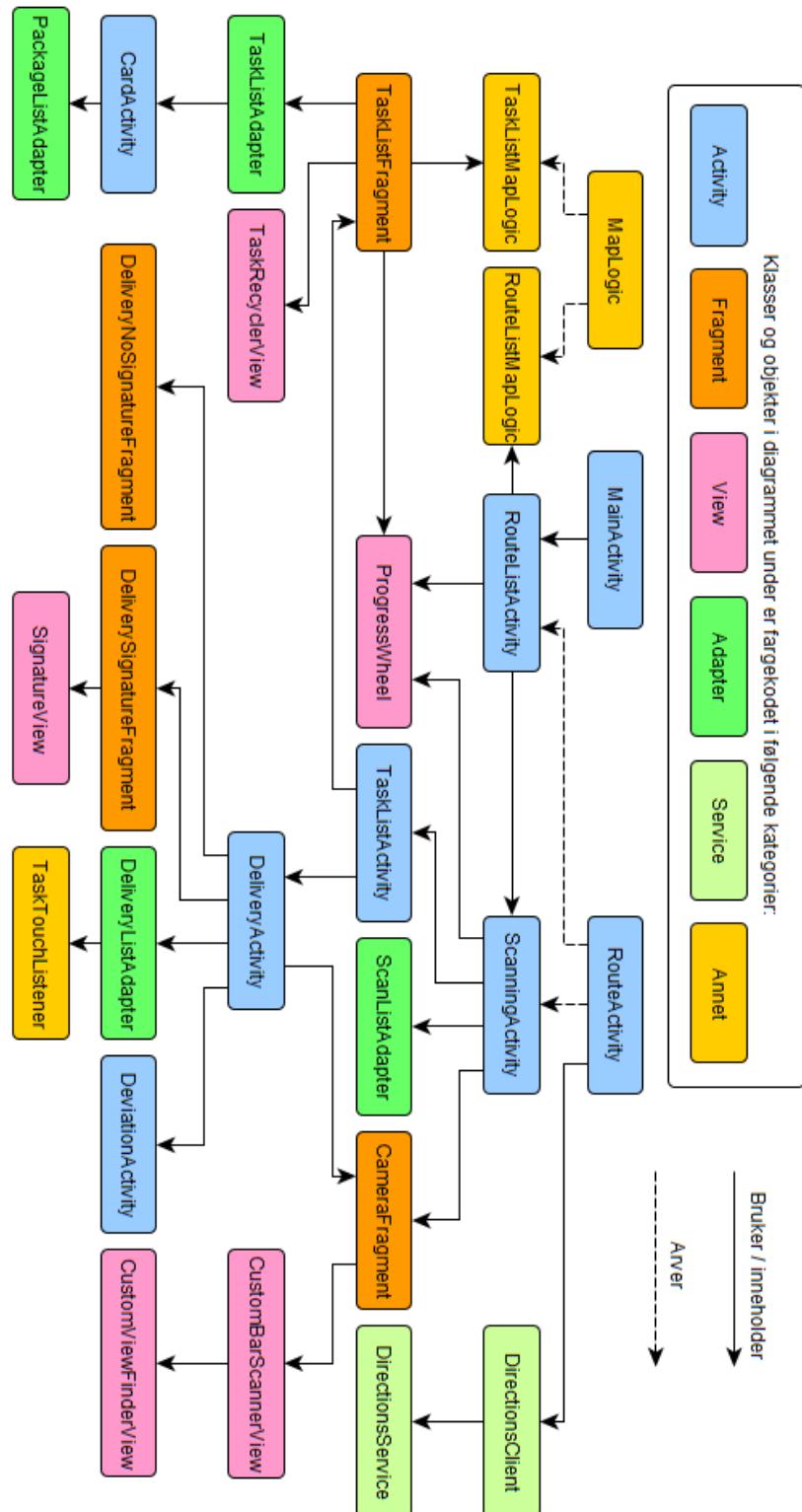
Figur 33 Flyt for PostNords nåværende håndterminal

For vår løsning kom vi frem til 5 hovedoppgaver som løses med fire skjermer. Resten av funksjonaliteten er koblet inn i disse skermene, eller tilgjengelig via sidemenyen i appen(Drawer). Dette forenklet flyten mye.



Figur 34 Flyten for vår Android-løsning

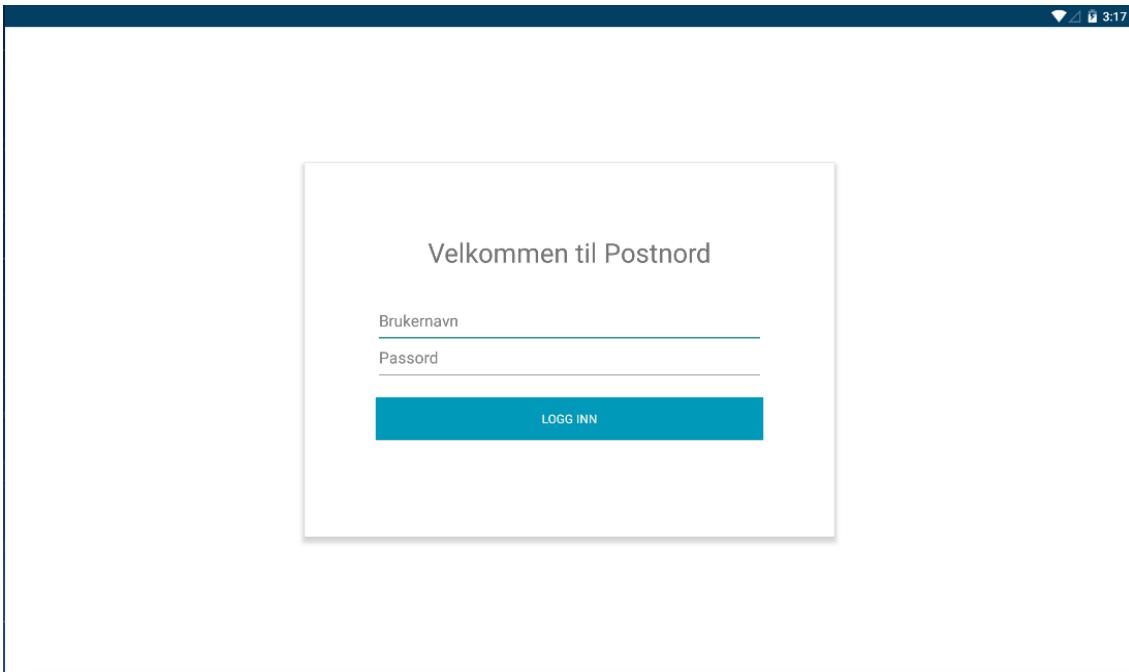
3.8 KLASSEDIAGRAM



Figur 35 Klassediagram for vår løsning

4. BRUKERVEILEDNING

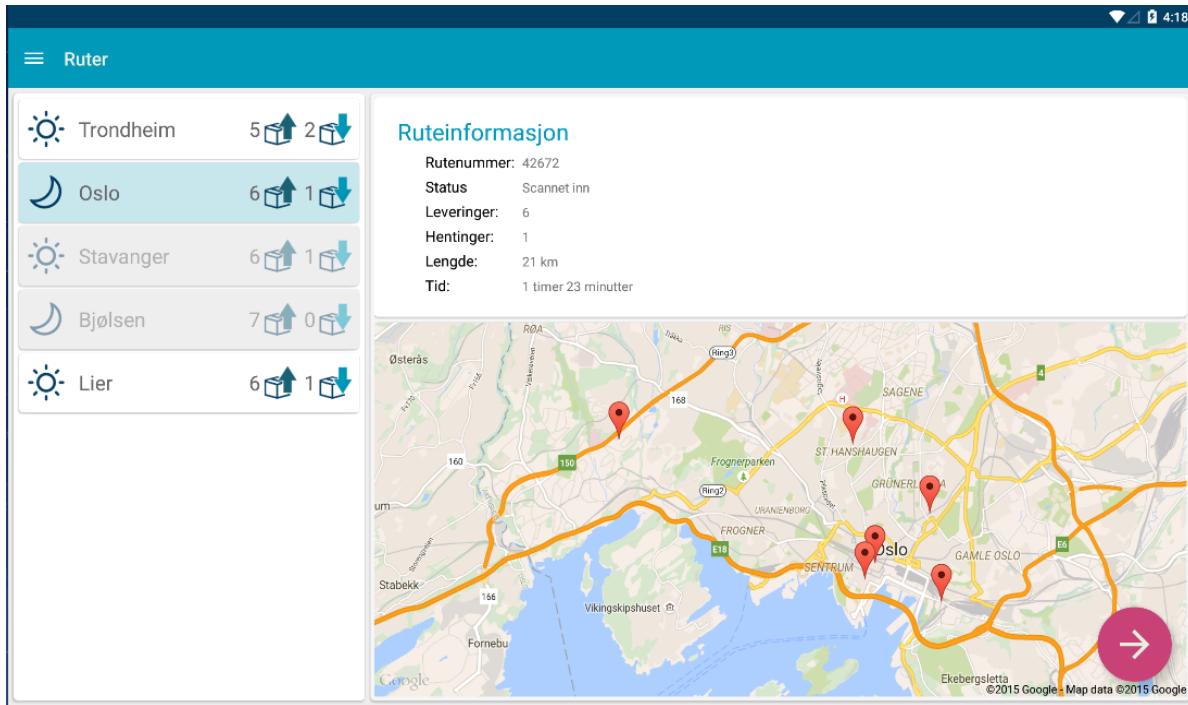
INNLOGGING



Figur 36 Innloggingsvindu

Det første som møter brukeren når appen startes opp, er innloggingsskjermen. For å logge inn skriver man inn brukernavn og passord, og trykker på "Logg inn". Ved mislykket innlogging, vises en enkel feilmelding under knappen.

RUTELISTE

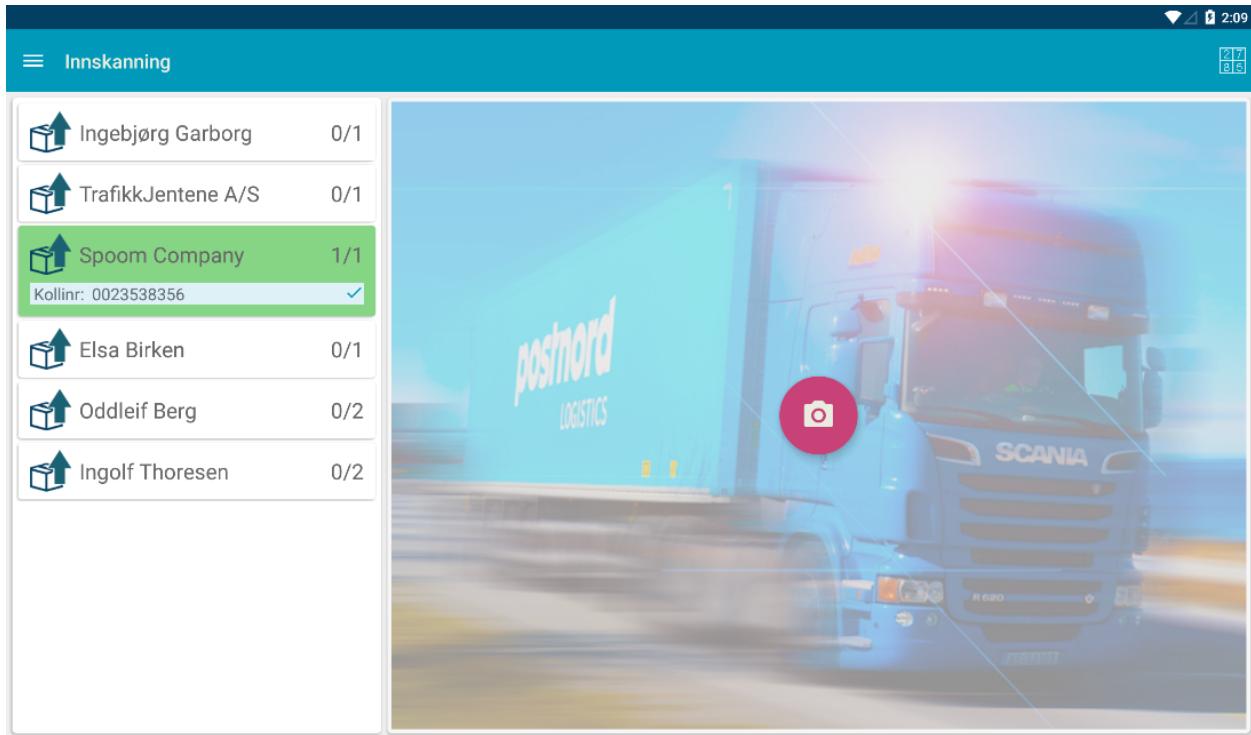


Figur 37 Rutelistevindu med oversikt over alle ruten en sjåfør kan velge

På rutelistskjermen ser man rutene som sjåføren skal levere for dagen. Her kan bruker klikke på de ulike rutene i listen, og få opp detaljert info til høyre. Ruteinformasjonen til høyre inneholder diverse info om ruten, inkludert en status, som forteller om ruten er fullført, underveis eller ikke påbegynt.

For å velge en rute å begynne på, trykker man først på ønsket rute, og så på FAB-knappen nederst til høyre for å gå videre.

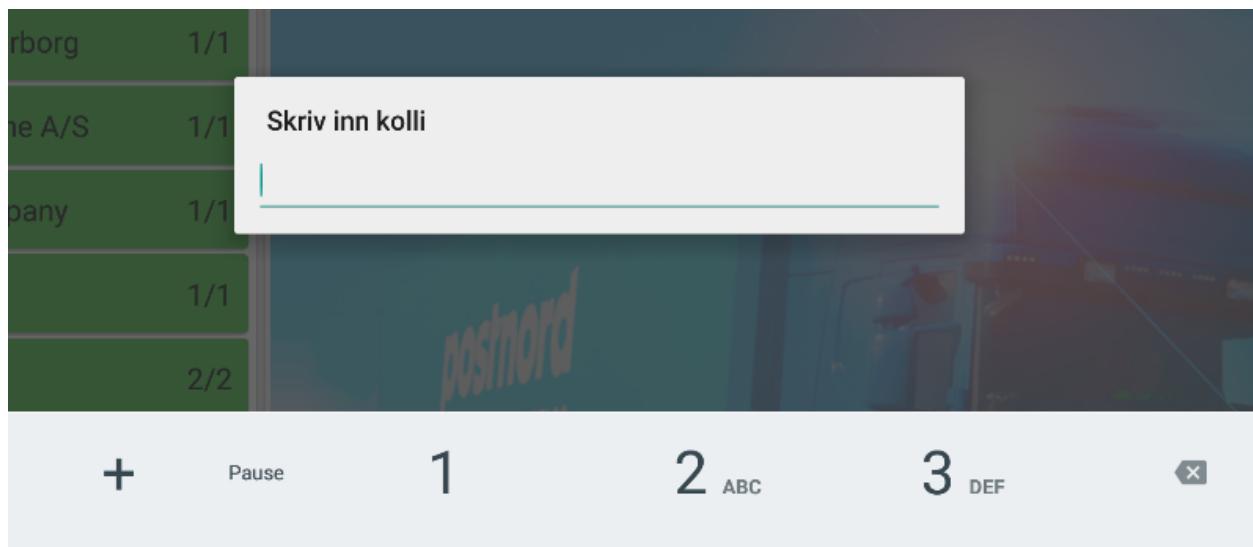
INNSCANNING



Figur 38 Innsanningsvindu med camera for scanning

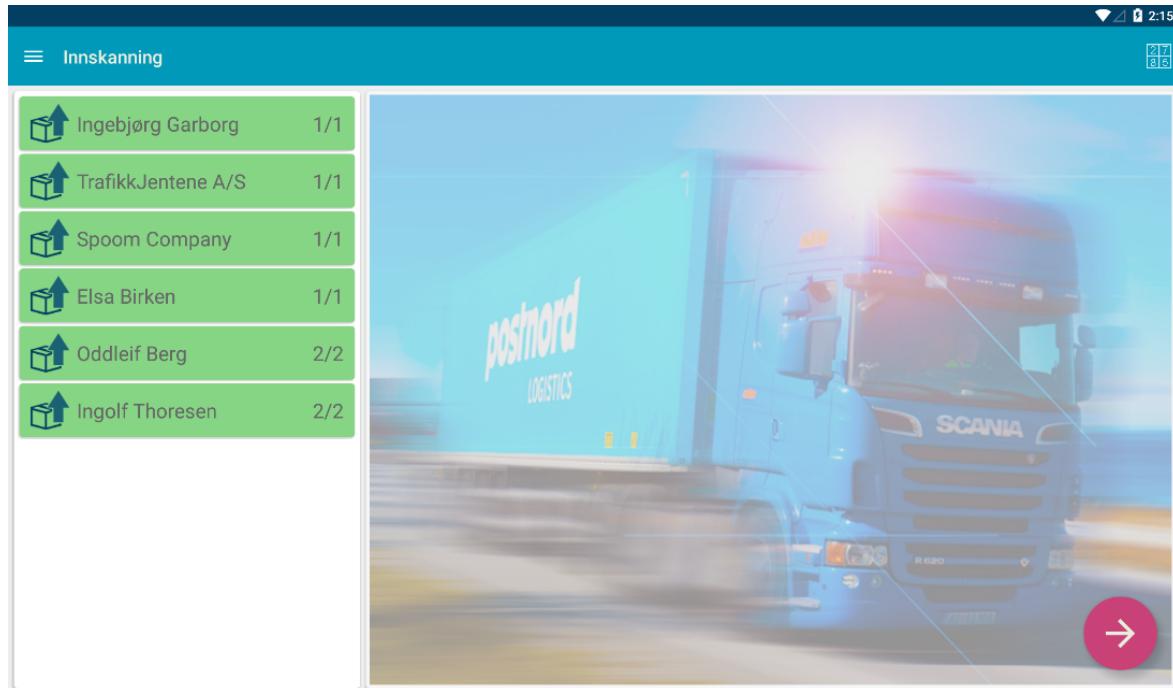
Før bruker kan begynne på ruta må pakkene scannes inn. I innsanningsvinduet kan finnes en liste over leveringer til venstre, og et kamervindu til høyre. Ved å trykke på kameraknappen i midten av kameravinduet, åpnes kameraet som brukes som scanner. Ved å peke kameraet mot en strekkode eller QR-kode, gjenkjenner appen koden, og registrerer om koden finnes i leveringslisten.

Hver rad i listen indikerer hvor mange kolli som er scannet for hver levering, og skifter farge om alle kolli for leveringen er scannet inn. Ved å trykke på en rad i listen ekspanderes raden til å vise alle kolli for leveringen. Ved å trykke et par sekunder på et kollinummer, blir pakken avscannet igjen.



Figur 39 Dialog for manuell innscanning av kolli

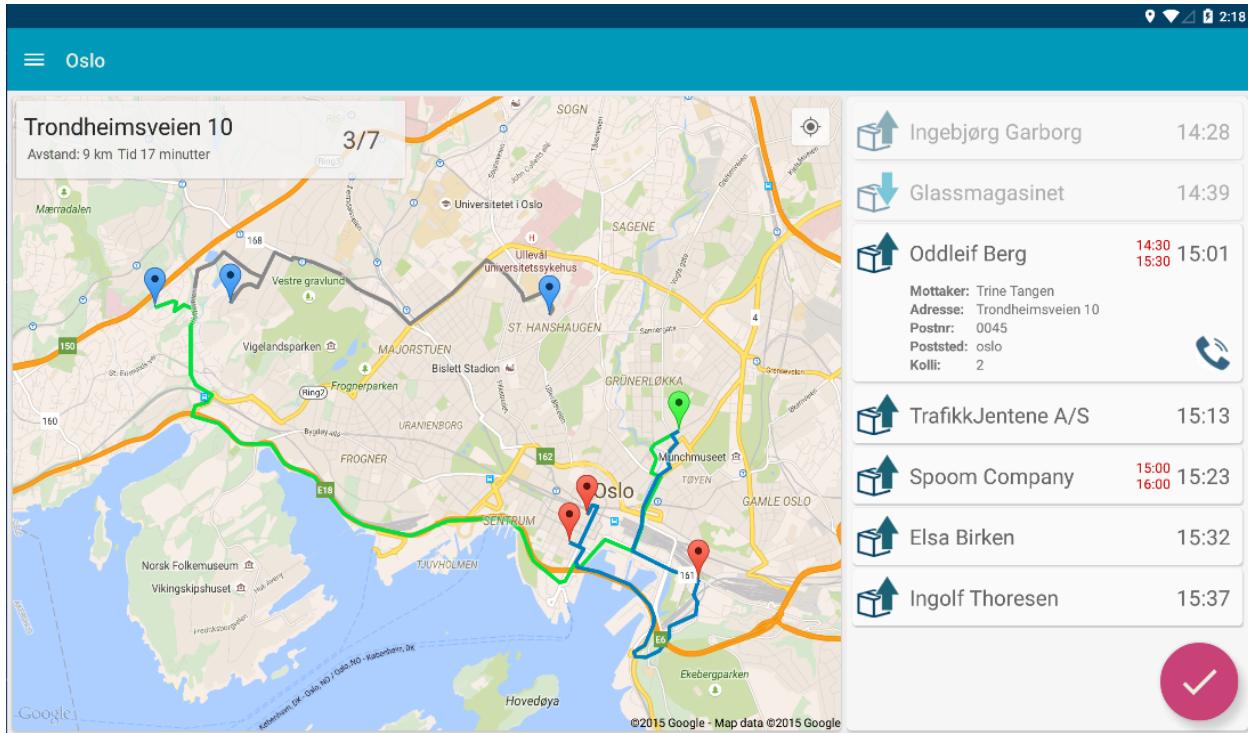
For å starte manuell innscanning, altså skrive inn kollinummeret manuelt, kan du trykke på ikonet i actionbar. Da får du opp et tekstfelt hvor bruker kan skrive inn pakkens kollinummer. Etter to inntastede siffer, vises en liste over forslag til kollinummer som finnes blant leveringene i ruten som matcher sifferne som er skrevet inn så langt.



Figur 40 Innskanningsvindu når alle pakker er scannet inn

Når alle pakker er scannet inn, kan bruker gå videre til arbeidslisten ved hjelp av knappen nede til høyre.

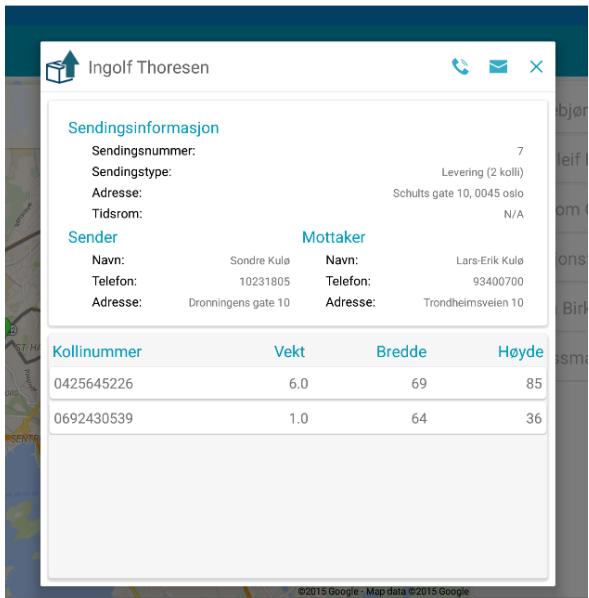
ARBEIDSLISTE



Figur 41 Arbeidslistevindu med oversikt over alle leveranser og henting til aktiv ruta

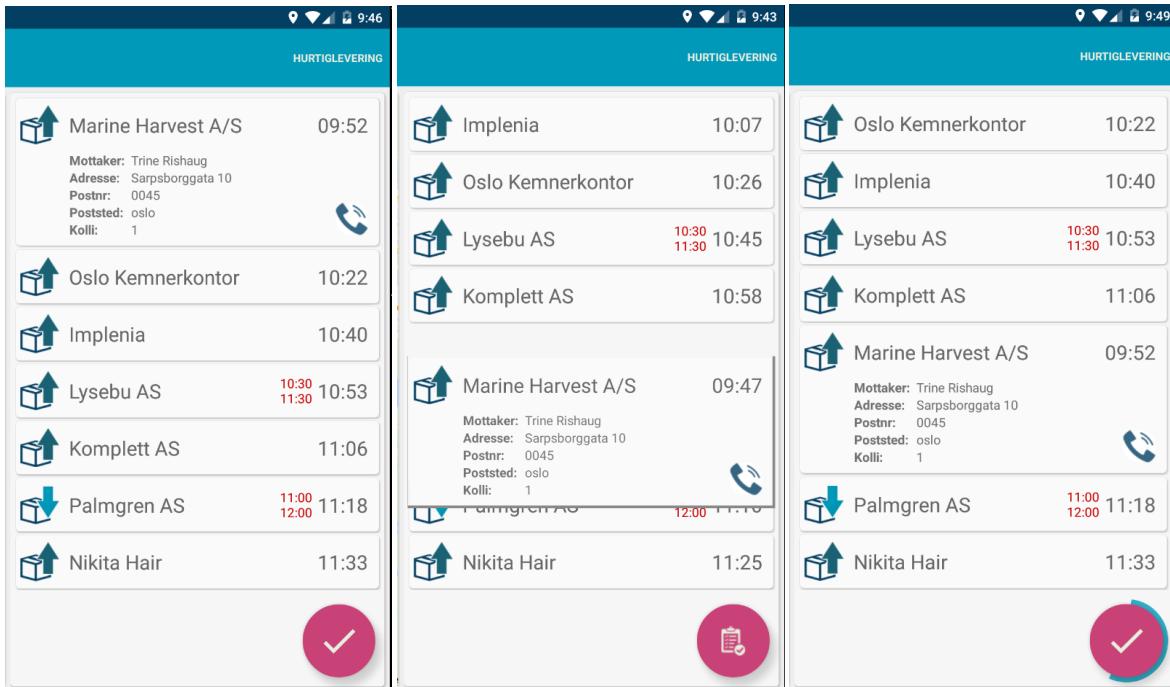
Etter innskanning kommer man til arbeidslisteskjermen. Til høyre ser man alle pakkene som skal leveres eller hentes, i rekkefølge etter når de skal leveres. Til venstre vises kartet, med markører for alle adresser som skal besøkes på ruten, og linjer som viser ruten mellom dem. Ruten opp til venstre i kartet viser info om neste henting og fremdrift i ruta. Mer info om info som vises på denne skjermen, finnes under punkt 3.6.4 i produktdokumentasjonen.

Også i denne listen er fullførte oppgaver grået ut. Og neste levering på lista, eller den aktive oppgaven, er utvidet til et større view enn de andre. Herfra kan du også enkelt ringe mottaker ved å klikke på telefonen i viewet.



Figur 42 Pakkecardet som viser detaljert informasjon om en Task

Ved klick på en oppgave i listen, åpnes et detaljvindu med utvidet info om denne oppgaven. Herfra kan du ringe mottaker, eller sende SMS om at sjåføren er på vei, ved hjelp av ikonene i actionbaren.

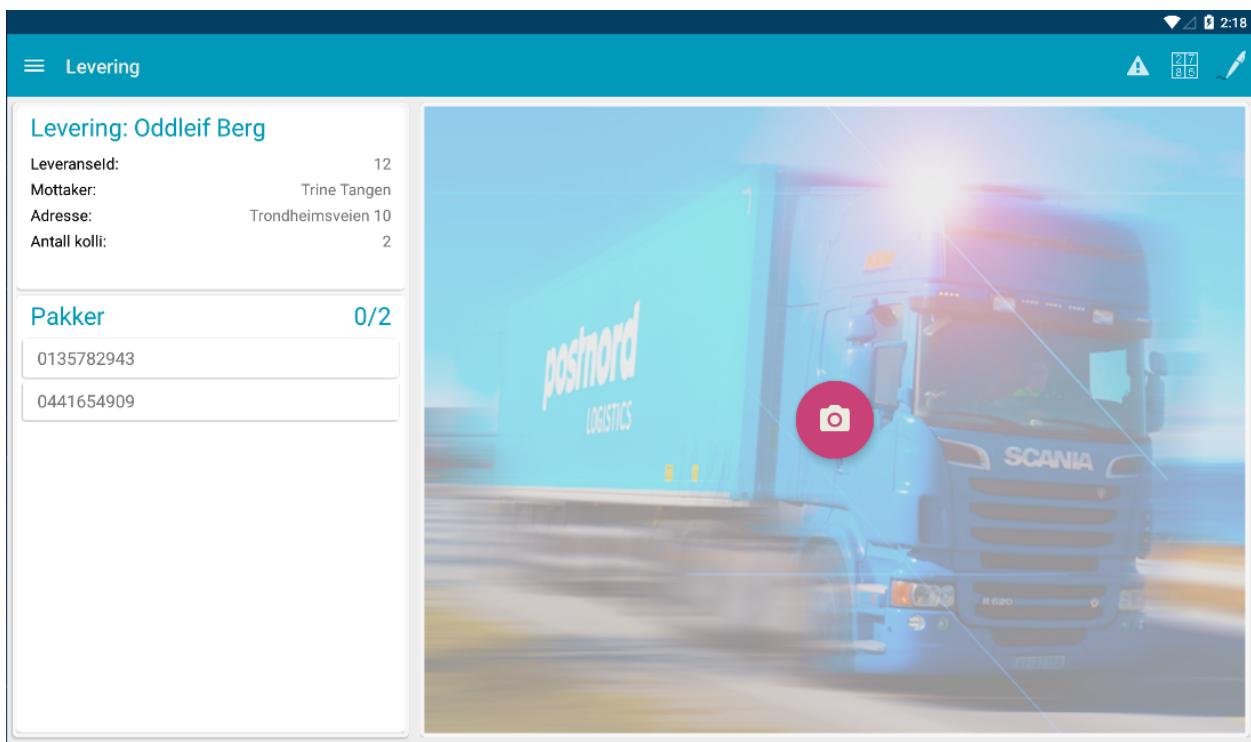


Figur 43 Bildene viser hvordan sortering av Tasks i arbeidslista foregår

For å omorganisere ruten manuellt, kan bruker longclicke på et listeelement i arbeidslisten, og så dra det opp eller ned, i klassisk drag&drop-stil. Her kan bruker bytte om på så mange rader som ønskelig, for så å trykke på FABen for å lagre endringene.

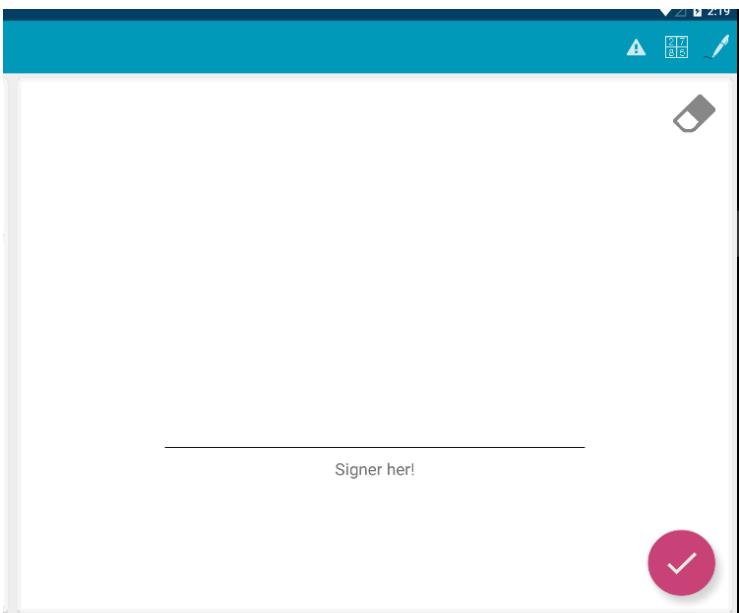
For å fullføre en oppgave (levere eller hente pakken) når sjåføren har ankommet leveringsadressen, trykker man enkelt på FABen for å gå videre til leveringsvinduet. Når alle pakker er levert, vil FABen fullføre ruten og føre deg tilbake til rutelisten.

LEVERING



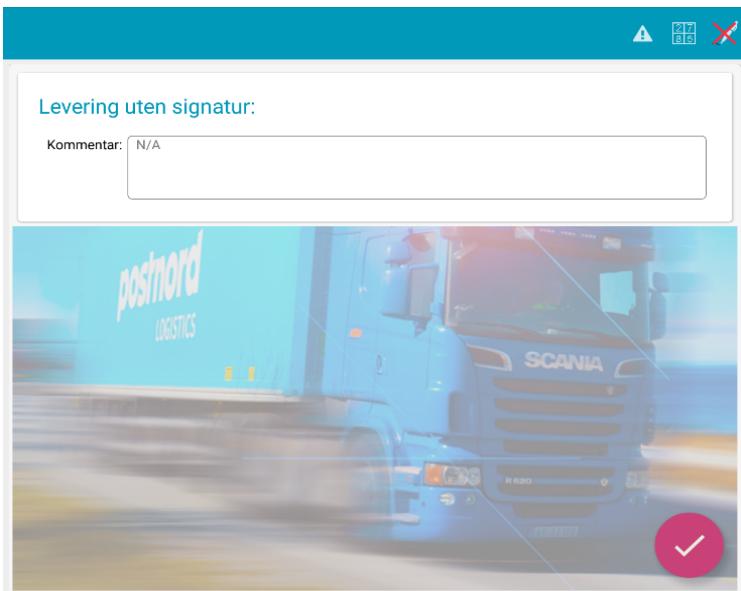
Figur 44 Leveringsvindu

I leveringsvinduet, vises først grensesnitt for utscanning, før man går videre til signatur. Utscanning foregår på samme måte som innscanning i ScanningActivity, med et kameravindu til høyre, som kan åpnes med knappen i midten, og lukkes igjen etter et gitt tidsrom. Strekkoder scannes ved hjelp av kameraet, og listen til venstre oppdateres ettersom de ulike kolli blir scannet.



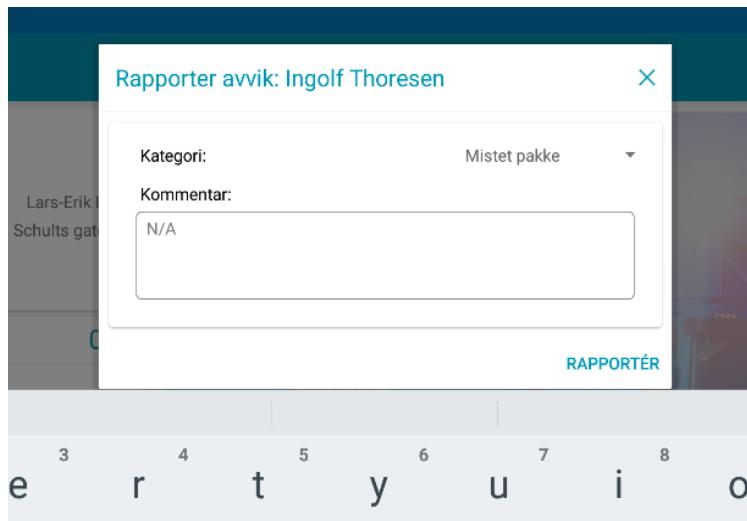
Figur 45 Signatursområde for levering med signatur

Når alle kollinummerene er scannet inn dukker signeringssvinduet opp. Her kan mottaker signere direkte på skjermen men en touch-penn. Går noe galt under signering, kan signaturn fernes ved å klikke på viskelæret opp til høyre. For å fullføre leveringen, kan bruker trykke på knappen nederst til høyre.



Figur 46 Kommentarfelt for levering uten signatur

Dersom innhenting av signatur ikke er mulig, kan man gå til levering uten signatur ved å trykke på ikonet med bildet av en penn, øverst til høyre. Her kan det skrives inn en kommentar for hvorfor signatur ikke ble benyttet. Også her kan leveringen fullføres ved hjelp av FABen.

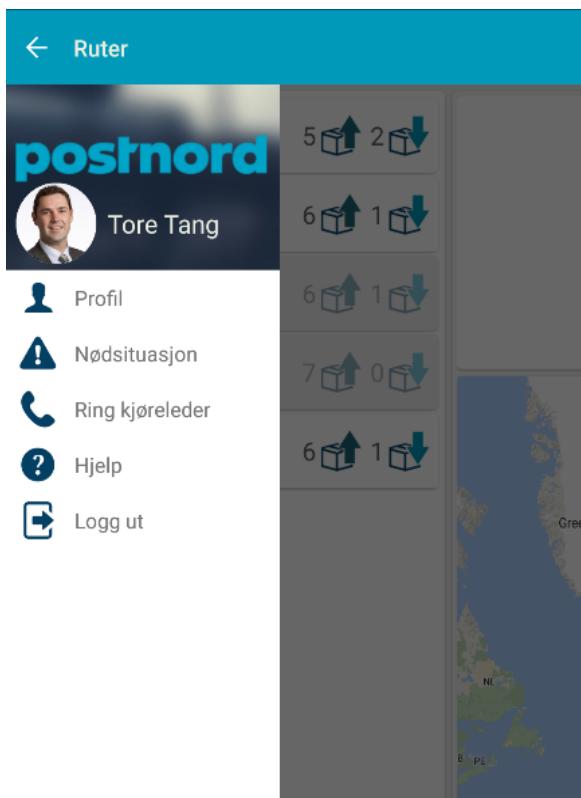


Figur 47 Rapportér avvik dialog

Ved skadet pakke, eller andre avvik, kan bruker åpne avvikdialogen, ved å trykke på avvikikonet i actionbar. Her kan bruker velge kategori for avviket fra en dropdown-liste, og skrive en kommentar i tekstfeltet under. Ved ferdig utfylling av feltene, benyttes "Rapportér"-knappen for å sende inn rapporten, avslutte leveringen og gå tilbake til arbeidslisten. For å gå tilbake til leveringsvinduet, kan bruker enkelt trykke på krysset i hjørnet av vinduet, eller utenfor vinduet generelt.

Etter en fullført levering føres bruker tilbake til arbeidslisten, og man kan fortsette på ruta.

NAVIGASJON/MENY



Figur 48 Skuffmeny for vår app. Også kalt drawer

I alle hovedskjermbilder, utenom innloggingsvinduet, finnes en navigasjon meny som en drawer på venstre side. Den kan dras inn fra siden med en swipe-bevegelse, eller ved klick på menyikonet oppe i venstre hjørne. Menyen legger seg over det aktive skjermbildet, og kan lukkkes ved å dra den ut igjen, eller klikke utenfor menyen. Her kan bruker velge blant følgende funksjoner ved å klikke på dem i lista:

Profil - Her kan man blant annet se informasjon og statistikk om sin egen brukerprofil.

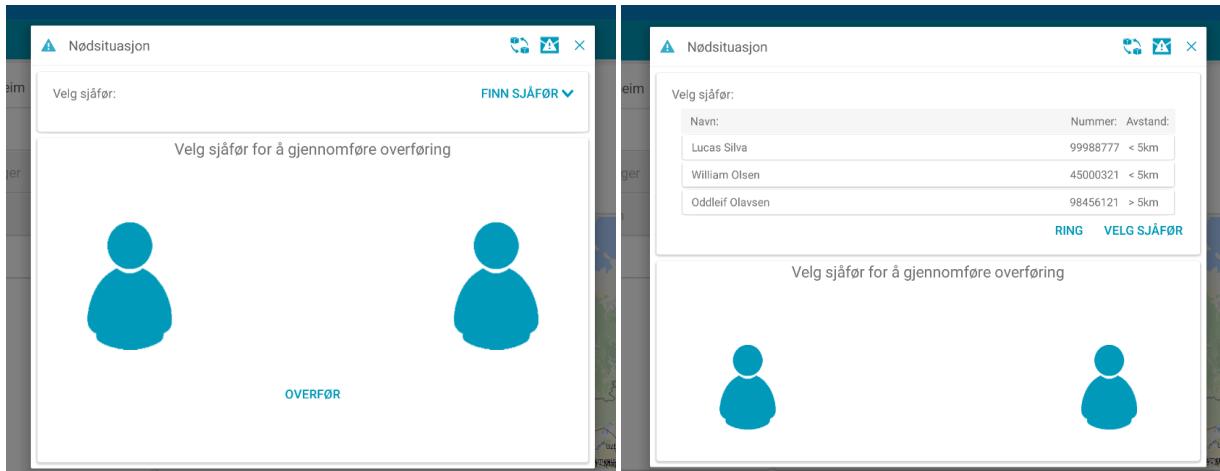
Nødsituasjon - Melde fra om større nødsituasjoner som punktering eller trafikkuhell.

Ring kjøreleder - Kun et tastetrykk for å ringe kjøreleder, ansvarlig for sjåfører og ruter.

Hjelp - Få hjelp om hvordan skjermen man er i fungerer.

Logg ut - Avslutte applikasjonen når man er ferdig for dagen.

Profil og nødsituasjon legger dialogvinduer over nåværende skjerm bilde, som kan lukkes ved å trykke på krysset opp i høyre hjørne eller utenfor vinduet. Nødsituasjon inneholder også noe mer funksjonalitet.



Figur 49 Nødsituasjonsvindu for overføring av pakker til andre sjåfører

Under nødsituasjon, har vi implementert to funksjonaliteter. Den ene er rapportering av avvik for hele ruten. Dette gjøres enkelt ved å trykke på avvikikonet i actionbar, som bringer opp et enkelt tekstmeldt. Her kan bruker skrive inn en kommentar som beskriver avviket og sende det inn ved å trykke på rapportér.

Valg av sjåfør gjøres med en dropdown-liste, som viser sjåfører i nærheten, med telefonnummer og avstand. Her har du mulighet til å velge en sjåfør, ringe han direkte, eller gå videre til pakkeoverføring. For å overføre pakkene, trykker du enkelt på overfør-knappen, og en animasjon vises for å illustrere at pakkene sendes. Dette er, som nevnt tidligere kun simulert i proof-of-concept-løsningen, men er funksjonalitet vi ser for oss vil kunne implementeres med tilkobling til PostNords interne systemer.

5. AVSLUTNING

5.1 MÅLOPPNÅELSE

5.1.1 RESULTATMÅL

Resultatsmessig føler vi at appen stort sett har nådd de målene vi satte. Likevel mangler det noe, som blant annet en visualisering av oppkoblingen mot Plottebordet. Det ble rett og slett prioritert å visualisere og perfeksjonere andre elementer mer. Utenom dette har vi oppnådd det vi anså som vesentlige prioriteringer, slik som et design som følger Material Design sine retningslinjer i størst mulig grad, et høyt fokus på forklarende animasjoner og automatisk ruteoptimalisering. Ved å bruke rammeverk som AndroidMP ble vi også i stand til å innfri på en brukerprofil med visualisert statistikk, med interaktive kakediagram og grafer.

1. Bruker kan logge inn og har ruter som kun er tilgjengelige for seg. Brukeren vil også få statistikk som er relativ til seg. Her har vi gjort alt vi gikk inn for.
2. Dette ble akkurat slik vi ville det skulle bli. Brukeren får oversikt over de rutene som er aktuelle for seg selv, og blir vist informasjon om denne. Det vil si, han får se et kart med alle stoppsteder og et sammendrag av informasjon. Her er også andre praktiske ting som titsestimat og estimert distanse for ruten.
3. Innscanning av pakker fungerte bedre, og lettere enn vi hadde antatt. Her fikk vi en viss kompetanse av Mesans Anders Ullnæs, som hadde brukt lignende funksjonalitet i andre applikasjoner. Dette fikk vi i løpet av en sprint tilpasset vårt eget bruk, og allerede i dagens utgave fungerer det nok bedre enn PostNords PDA laserscanner.
4. Når det kommer til ruteoptimalisering og hver enkelt sjåførs oversikt over ruten, føler vi at vi fikk til mer enn det man kan forvente i det gitte tidsrommet. Vi fikk vist frem en optimalisert rute i et google map, med opptil åtte mellomledd. Her vil PostNord selv kunne implementere en løsning med opptil tjuefem mellomledd, og eventuelt kjøre flere kall i sekvens for å få tilbake en større løsning. Det vi derimot ikke fikk til å implementere en realistisk løsning for var pakker som må leveres innenfor en gitt tidsfrist. Her endte vi kun med en mockup-løsning som ikke har funksjonaliteten, men illustrerer poenget vårt. Det ble satt av en og en halv uke for en av medlemmene i gruppa til å jobbe med dette problemet, men det viste seg vanskelig å implementere noe som fungerte på toppen av google maps, og i tillegg fungerte tilstrekkelig.
5. Det var aldri noe problem å implementere en løsning for å håndtere avvik på leveranser, og heller ikke å gi mottakeren tilstrekkelig med informasjon om leveransen. Dette vises på leveranse-skjermen hvor vi viser kollinummer og kundenavn, slik at det er synlig for kunde når man signerer.

- All tilleggsfunksjonalitet i applikasjonen har blitt lagt inn slik at det ikke bryter med hovedflyten, men er tilgjengelig der man forventer det. Altså om man trykker på en leveranse i arbeidslisten vil dette vise alle kolli, mottaker og sender. Disse industristandardene er ikke tatt høyde for i den gamle løsningen da skjermstørrelsen man jobber med på en PDA er i en helt annen liga. Det kan letttere sammenlignes med en mobil enn en tablet.

Alt i alt føler både vi, Mesan og PostNord at dette produktet belyser problemstillingene i logistikkbransjen på en realistisk måte. Det ble også nevnt flere bruksområder for denne applikasjonen av PostNord selv, enn vi var klar over. For eksempel for utlevering av post i butikker, ville en tablet vært ypperlig, mens sjåfører på avsideliggende steder, som ikke har tilgang til en PDA, ville ha stort behov for en mobilversjon man kan laste ned. Dette er problemstillinger som er mye enklere å implementere, som følge av bruk av Android-plattformen. Personlig mener vi at all vesentlig funksjonalitet ble implementert, og føler vi har nådd resultatmålene i stor grad. Vi er i alle fall fornøyde og stolte av produktet vårt.

5.1.2 EFFEKTMÅL

Ettersom oppgaven ikke er satt i produksjon, og kun var ment for å så et frø om at håndterminalen trenger å oppdateres, gjenstår det å se hvorvidt effektmålene vil bli oppnådd. Likevel merket vi etter presentasjonen av prosjektet, at PostNords representanter var entusiastiske og interesserte i å få i gang arbeidet med å fornye systemene deres. Dette er i så fall en oppfyllelse av vårt effektmål om å starte en dialog mellom Mesan og PostNord, som potensielt kan føre til flere nye oppdrag for Mesan. Dette vil på sikt gi økt omsetning for Mesan og muligheten til å bryte inn i et stort og avlukket marked, som også er noen av målene for arbeidsgiver.

Et annet effektmål var å hjelpe studentene i prosjektgruppen inn i arbeidslivet. Dette føler vi vi til en viss grad har nådd, etter at alle på gruppen har søkt jobb hos Mesan. Vi måtte da gjennom en normal rekrutteringsprosess, med søknad, intervjuer og forhandlinger. Minst en av oss har også fått jobbtilbud hos dem, og alle har fått verdifull erfaring og veiledning når det gjelder jobbjakt.

Når det gjelder effektmålene for brukere av appen, og kunden, PostNord, avhenger dette av PostNords avgjørelse. Hvis appen, eller en variant av den, går i produksjon, vil dette kunne skape verdi både for brukerne og PostNord som selskap. Men dette gjenstår altså å se.

5.1.3 LÆRINGSMÅL

Vi føler vi har lært mye gjennom prosjektet generelt, både innen prosjektarbeid og androidutvikling, og føler prosjektet ble en god forberedelse på arbeidslivet. Vi fikk prøvd oss på arbeid utifra en kravspesifikasjon som vi selv måtte tilpasse og forhandle oss frem til løsninger på. Ulempen var at vi ikke hadde kontakt med den faktiske kunden, PostNord, før presentasjonen på slutten av prosjektet. Dette skyldtes at prosjektet skulle være en "overraskelse" for PostNord, hvor de skulle få overlevert en mer eller mindre ferdig prototype, som skulle være imponerende nok til at de innså at dette var noe å satse på. Mesan steppet inn og representerte kunden underveis, men prosessen ble dermed noe mindre "reell" enn vi hadde håpet. På den annen side fikk i mulighet til å presentere produktet for kunden, og ha et møte med dem, hvor vi diskuterte løsningen. Her representerte vi Mesan 100% overfor en av deres største kunder. Dette følte vi vi lærte mye av, og var en god realistisk erfaring. Alt i alt føler vi vi nådde målene om ekte arbeidserfaring.

Vi fikk også prøvd oss på en rekke utfordringer og kompliserte løsninger i Androids nyeste operativsystem, og føler vi har lært veldig mye her. I og med at dette er såpass nytt, er det faktisk mulig at vi er blant de i Norge med mest erfaring med Android 5.0 for øyeblikket. Dette skyldes at mange venter med å utvikle i nyeste versjon av Android, både for å komme forbi babyfasen hvor nye operativsystemer ofte har mange feil, men også fordi bakoverkompatibilitet er veldig viktig i Android ettersom kun en liten andel av devicene kjører de nyeste versjonene. Ettersom vår løsning var en proof of concept kunne vi likevel satse på siste versjon, siden vi selv kunne velge hvilken device og Android-versjon som skulle brukes av våre brukere, og ikke trengte å ha et 100% feilfritt system ved levering. Vi føler altså vi har nådd målene om læring innen Android-utvikling.

Innen prosjektplanlegging og arbeidsmetodikk, fikk vi prøvd oss på planlegging og smidig metodikk på større skala. Vi lagde en 4-måneders arbeidsplan, med sprinter og milepæler, og fulgte disse hele veien. Vi dokumenterte arbeidet og fremdriften, brukte scrum-metoder for fordeling av arbeid og hadde daglige møter hvor status og videre arbeid ble diskuterte. Alt dette føler vi fungerte bra, og vi lærte veldig mye om arbeidsprosessen og hverdagen en

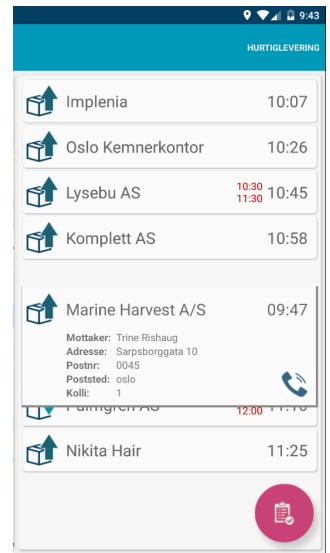
utvikler har. Vi føler dermed vi oppnådde målene vi satte for læring innen arbeidsmetodikk og planlegging.

For prosjektet som helhet, føler vi vi nådde og overgikk målene vi satte for læring.

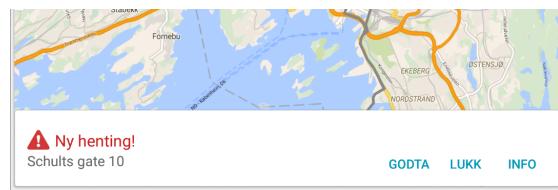
5.2 OPPFYLLElse AV KRAV

5.2.1 FUNKSJONELLE KRAV

Produktet oppfyller i stor grad de funksjonelle kravene som ble satt for oppgaven, og vi la også til noen flere funksjoner i den egendefinerte kravspesifikasjonen. Et eksempel er at vi implementerte muligheten for at sjåfører lett kunne omorganisere kjørelisten sin etter eget ønske. Her kan man bare holde fingeren på en leveranse i listen, og dra den opp eller ned. Kartet vil deretter oppdatere seg når man trykker på den røde knappen.



Videre implementerte vi henting av ekstra pakker underveis i skiftet, ved at sjåfør får opp en liten popup på bunnen av skjermen, som spør om han har tid til å hente en pakke på en gitt adresse, som forhåndsvises på kartet.



Figur 51 Ny henting beskjed

Figur 50 Task oversikten i arbeidslista (Under manuell sortering)

En av tingene vi ikke implementerte som vi gjerne ville ha med, var automatisk sending av SMS til kunde basert på geofencing eller avstandsberegning til destinasjon. Ettersom vi ikke testet applikasjonen ute i felten, ble dette litt vanskelig å gjennomføre, men vi har alle de grunnleggende komponentene på plass for at dette skulle blitt fullført.

Utenom de nevnte punktene skal resten være gjennomgått tidligere, og fullført på en tilstrekkelig måte.

5.2.2 IKKE-FUNKSJONELLE KRAV

Som tidligere nevnt var dette ikke en stor del av oppgaven vår, da applikasjonen hovedsaklig kun skulle vekke interesse for produktet hos kunden. De tingene vi valgte å fokusere på her ble derfor estetiske og praktiske ting, slik som design og kort vei til nyttig funksjonalitet.

Her føler vi at det har vært høy oppnåelse. Ingen av valgene har lengre enn 3 klikk for å nås, og vi forholder oss til mange av de beste praksisene innenfor apputvikling på design-siden av ting. Designdokumentene til Material Design ble alltid en plass vi letet etter svar om vi lurte på noen vesentlige valg. Til og med på emulator fungerer applikasjonen knirkefritt, og føles mye mer responsiv enn de fleste applikasjoner på brukermarkedet.

Bevis på disse antagelsene våre finner man blant annet i demonstrasjonen vår for PostNord, og akseptansetestene med Mesan. Det var sjeldent spørsmål om hvordan ting fungerte, og det ble mer en naturlig læreprosess av at man trykker seg litt frem. Deretter ville brukerne forstå helt til neste gang hva resultatet av hver handling var. Det kan hende barriären er litt høyere for faktiske yrkessjåfører å lære, men PostNord kunne likevel se for seg at man sparte mye på opplæringsprosessen, kontra den gamle løsningen.

5.3 VIDEREUTVIKLING

Siden vårt prosjekt er en proof of concept, skal ikke løsningen videreutvikles direkte. Poenget med oppgaven er å vise kunden hvilke muligheter som ligger i en modernisering og oppgradering av programvare, samt skifte a plattform. Likevel er målet å sette i gang en prosess hos kunden, hvor resultatet er at vår løsning eller tilsvarende blir satt i produksjon. Avhengig av hvor fornøyd kunden er med løsningen, kan den bli mer eller mindre direkte overført til den faktiske løsningen som blir satt i produksjon og tatt i bruk. Sjansen for at en løsning utviklet uten dialog med kunden, som vår, treffer kundens forventninger 100% er veldig liten, og det er forventet at den endelige løsningen vil variere en god del fra dette utkastet.

I vårt tilfelle var kunden svært fornøyd med valgene vi tok og løsningene som ble presentert. De var imponert over hvor gjennomtenkt løsningen var, og hvordan gruppen hadde sett koblingene mellom sjåførens behov, businessaspektene og teknologivalg. Dette betyr at det endelige løsningen kan bli ganske lik den vi har i dag. Representantene fra PostNord sa de skulle snakke videre om prosjektet internt, og virket meget positive til å sette i gang arbeidet rundt problemstillingen.

De nevnte også at dette i første omgang kunne være mest aktuelt for andre brukere innen deres virksomhet. Et konkret eksempel var ansatte innen post i butikk. Disse bruker en håndholdt device til scanning av pakker og sortering av disse. Her kan deler av funksjonaliteten fra vår løsning implementeres i en mindre applikasjon, som er lettere å få gjennomslag for å begynne produksjon av. Videre kan da utviklingen spre seg til andre deler av virksomheten, inkludert lokaltransportsjåførene, som vi har fokusert på.

5.4 VURDERING AV PROSJEKTPERIODEN

Gruppen har jobbet iherdig fra første stund, og tatt arbeidet på alvor. Fordelen vi satt med fra dag én var naturlig nok at vi alle hadde samme ambisjoner med prosjektet. Alle var der for å lære, og for å utvikle et godt produkt. På toppen av dette ønsket alle en god karakter, men dette ble faktisk nedprioritert litt sammenlignet med å prestere på den tekniske og praktiske siden av ting. Ingen av oss hadde heller gjennomført et prosjekt av denne skalaen før, noe som gjorde ting utfordrende, men spennende.

Vi fikk også alle vært gjennom et intervju hos Mesan etter dette prosjektet da det fra starten av ble ymtet frempå om at prosjektoppgaven vår var en del av Mesans rekrutteringsprosess.

Alt i alt sier gruppen seg meget fornøyd med opplevelsen, og støtten vi har mottatt fra Mesan og veileder.

5.5 ATTEST FRA ARBEIDSGIVER

Konsulentsjef Harald Kulø fra Mesan sier følgende om prosjektet:

Prosjektgruppen fra HIOA har hatt hovedprosjekt hos Mesan våren 2015 der de leverte en proof-of-concept applikasjon for håndterminalen til PostNord. Gruppen har vist stor grad av selvstendighet og læreruilje i prosjektet og har levert en løsning som imponerer både Mesan og PostNord. Spesielt imponerende er hvordan gruppen har tatt innover seg det funksjonelle og demonstrert forretningsmessig forståelse for løsningen. Gruppen har fulgt et scrum-oppsett under perioden som har fungert bra. Mesan har hatt rollen som produkteier og har vært med på akseptansetester mm. Dokumentasjonen som ble skrevet underveis er god og gir et godt innblikk i løsningen. Under demonstrasjon for PostNord der IT-direktør, utviklingssjef, transportsjef og prosjektleder for håndterminalen var til stede, viste gruppen at de hadde forstått oppgaven og imponerte kunden med sin nytenkning.

Vi er meget godt fornøyd med gruppens innsats. Prosjektet har gitt både Mesan og PostNord et konsept som kan utvikles videre i tiden fremover.

5.6 KONKLUSJON

Etter avsluttet prosjekt er gruppen enige om at dette har vært en svært lærerik prosess, hvor både gruppen, arbeidsgiver, kunde og skolen sitter igjen med gevinst. Vi har alle fått mange nye erfaringer innen prosjektarbeid, arbeidsprosess og planlegging, og har også lært mye teknisk innen Android-utvikling. Arbeidet har vært interessant, spennende og utfordrende, og vi føler vi også har fått tatt i bruk mye av kunnskapen vi har opparbeidet oss gjennom studiet. Vi opplever prosjektet som ikke bare en skoleoppgave, men en modningsprosess og en god forberedelse på arbeidslivet.

Vårt endelige mål gjennom hele prosessen har vært at kunden og arbeidsgiver skulle få et produkt de var fornøyde med, og at gruppen skulle kunne være stolte av det vi har fått til. Dette føler vi vi har lykkes med etter å ha mottatt gode tilbakemeldinger fra Mesan underveis, og også meget god respons fra PostNord under presentasjonsmøtet med dem.

I forkant var vi spente på om vi kom til å klare å legge en god plan og disponere tiden godt i prosjektperioden. Dette føler vi har gått veldig bra, og vi har ikke opplevd å havne bakpå på noen måte, eller måtte jobbe mye overtid for å nå frister. Noen av sprintene ble justert underveis, for å kompensere for uforutsette feriedager og lignende, men ellers har vi jobbet jevnt og trutt gjennom prosjektperioden, fulgt planen, og ikke endt opp med å måtte ta et skippertak på slutten. Dette er vi godt fornøyde med, og føler har forberedt oss godt på estimering av tid i fremtiden.

Alt i alt er vi godt fornøyd med prosjektet, og føler vi kan være stolte av produktet vårt, og arbeidet underveis. Til slutt vil vi rekke en takk til Mesan, for en spennende oppgave og godt samarbeid.

6. VEDLEGG

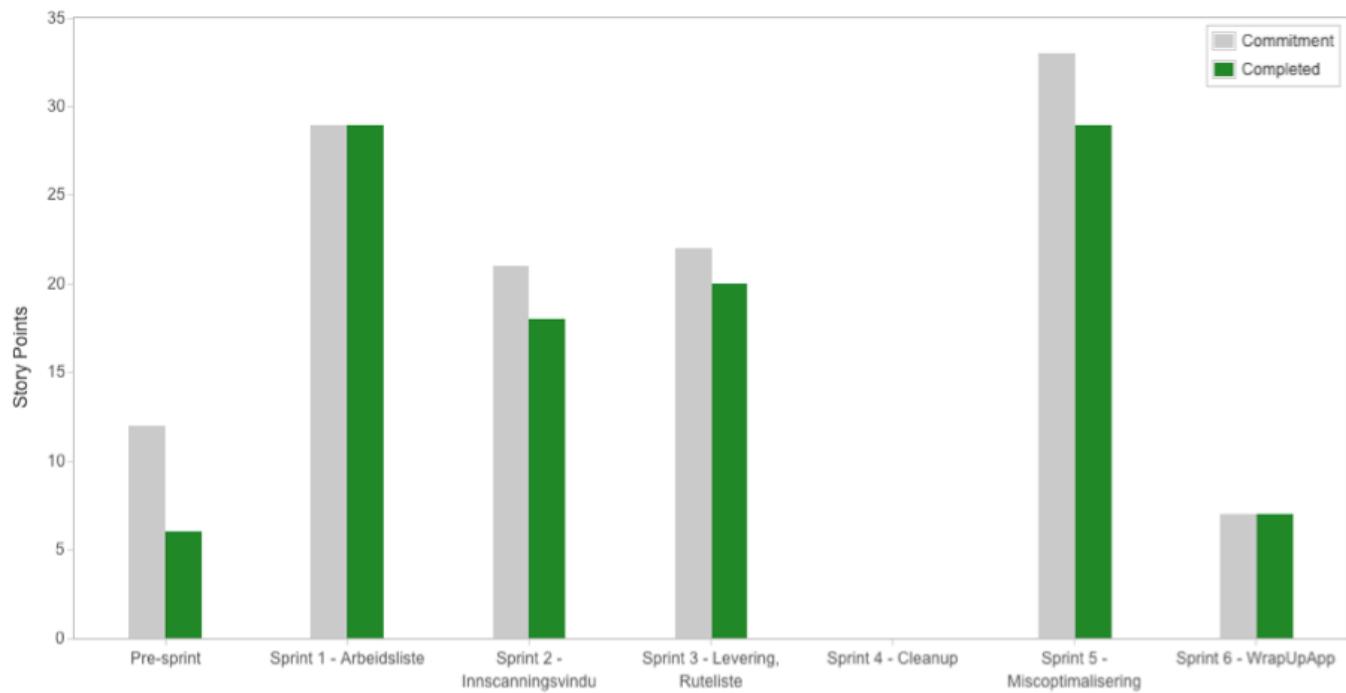
6.1 SPRINTLOGS / BURNDOWN

SPRINT RAPPORT

- Velocity Chart - Hele prosessen:
- Sprint 1 - Arbeidsliste
 - Burndown graf
- Sprint 2 - Skanning
 - Burndown graf
- Sprint 3 - Levering, Ruteliste
 - Burndown graf
- Sprint 4 - Cleanup
 - Burndown Chart
- Sprint 5 - Miscoptimalisering
 - Burndown Chart
- Sprint 6 - WrapUpApp
 - Burndown Chart

Velocity Chart - Hele prosessen:

Velocity Chart [\(?\) How to read this chart](#)

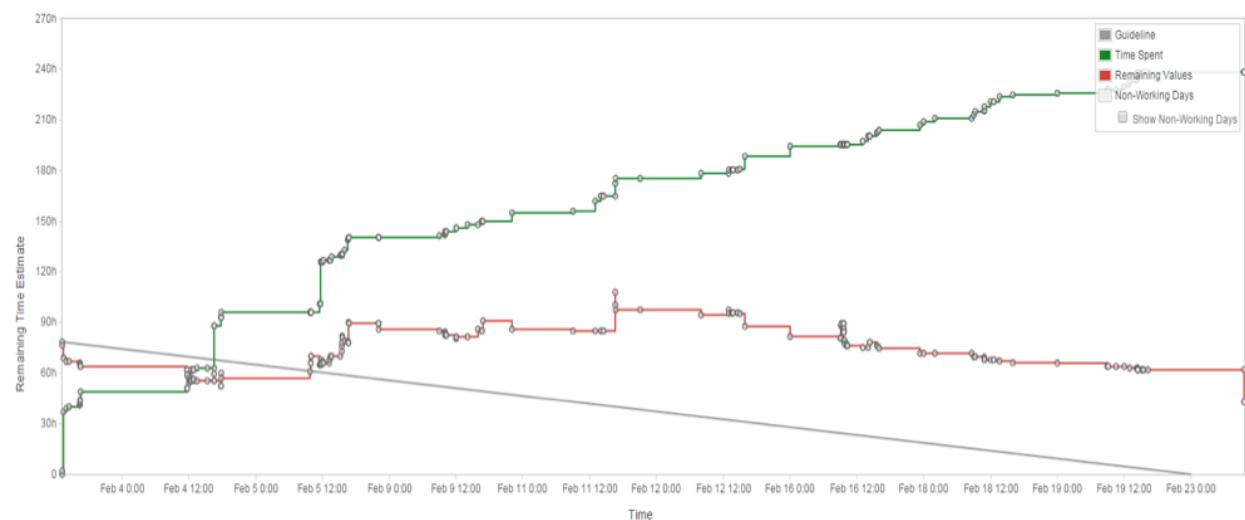


Sprint 1 - Arbeidsliste

Periode:	2.Feb - 23. Feb
Story points:	29 SP
Tid pr SP:	8.22 timer
Gjenst��ende arbeidstimer:	571 timer (est.)

Burndown graf

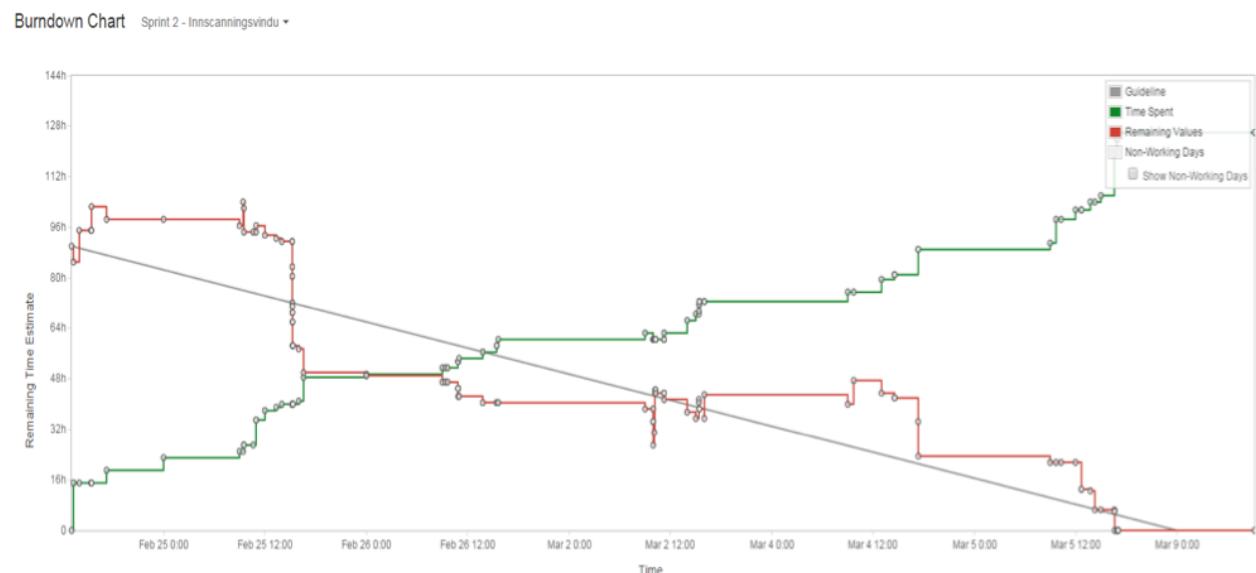
Burndown Chart Sprint 1 - Arbeidsliste ▾



Sprint 2 - Scanning

Periode:	23.Feb - 9. Mars
Story points:	18 SP
Tid pr SP:	7 timer
Gjenst��ende arbeidstimer:	465,6 timer (est.)

Burndown graf



Sprint 3 - Levering, Ruteliste

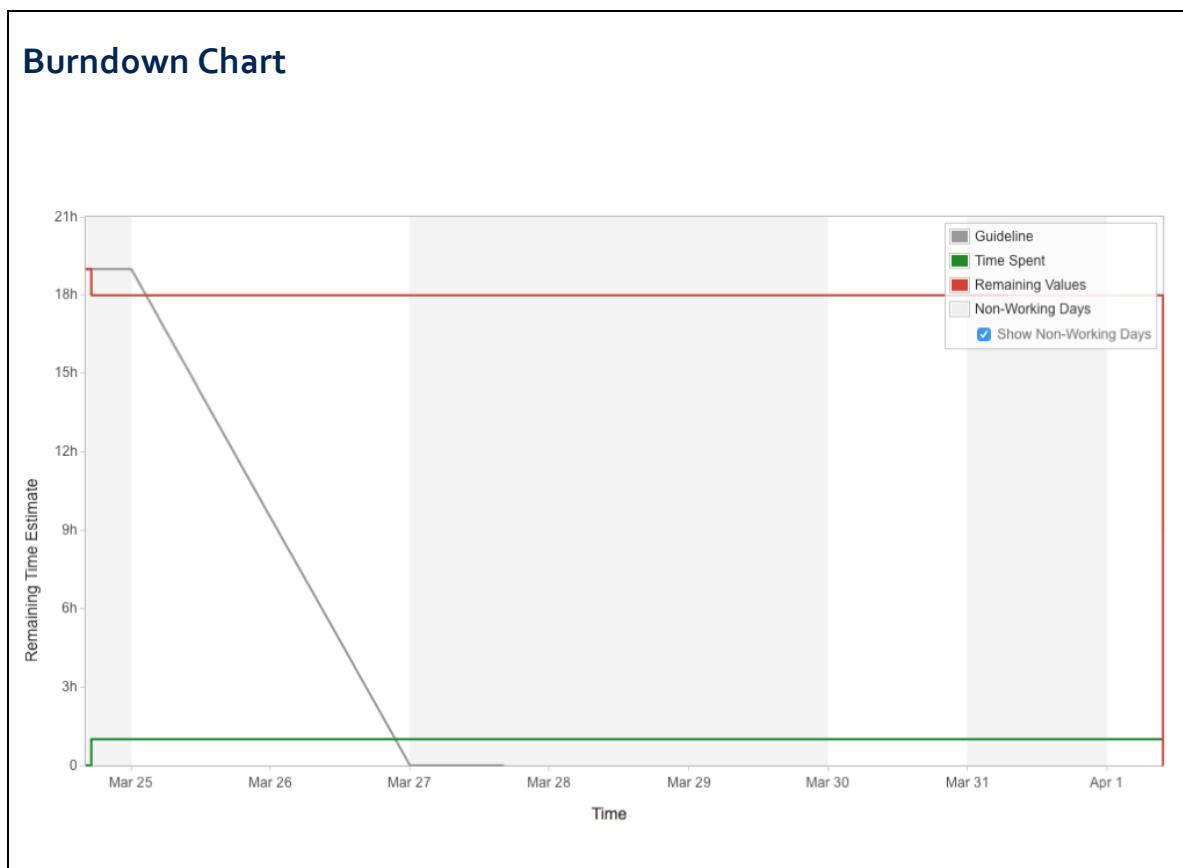
Periode:	9.Mars - 24. Mars
Story points:	20 SP
Tid pr SP:	4.85 timer
Gjenst��ende arbeidstimer:	150,4 timer (est.)

Burndown Chart Sprint 3 - Levering, Ruteliste ▾



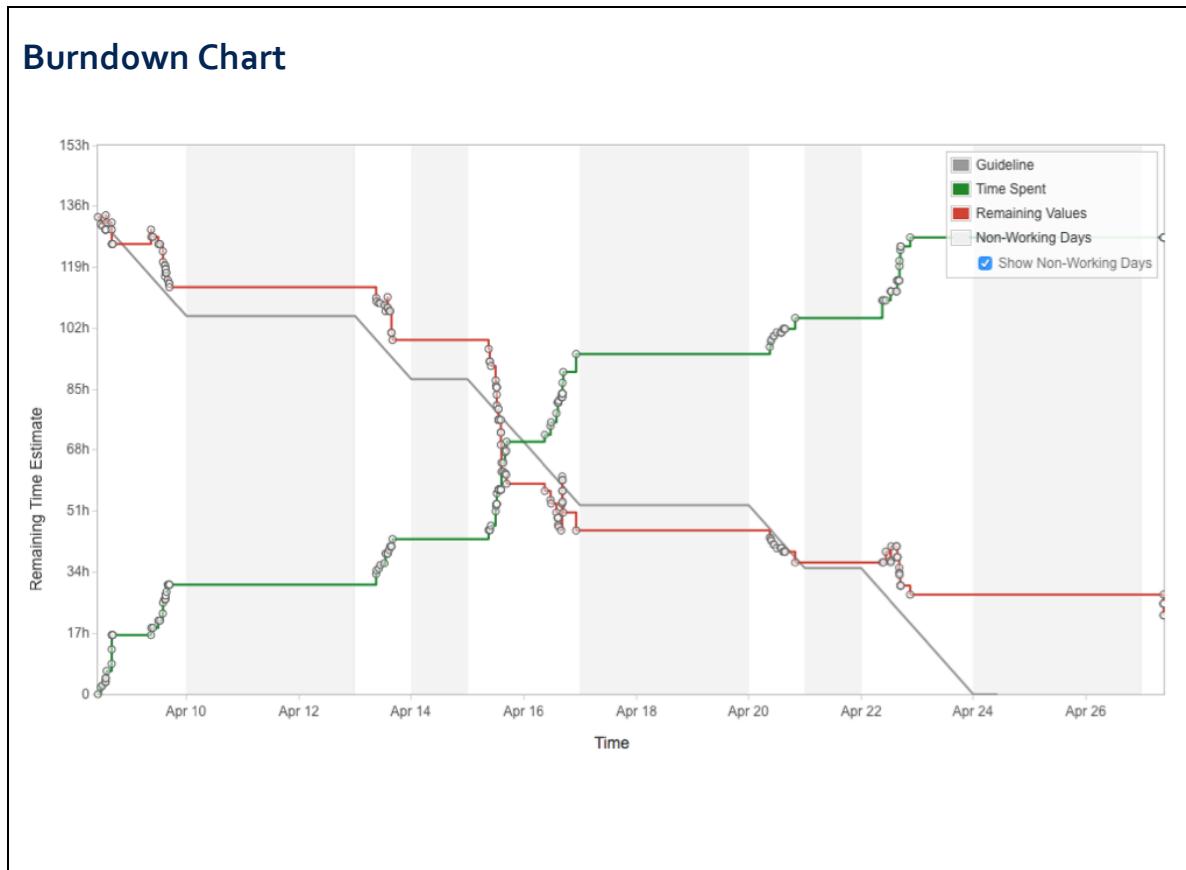
Sprint 4 - Cleanup

Periode:	24.Mars - 1.April
Story points:	0
Tid pr SP:	-
Gjenst��ende arbeidstimer:	150,4 timer (est. fra forrige sprint)



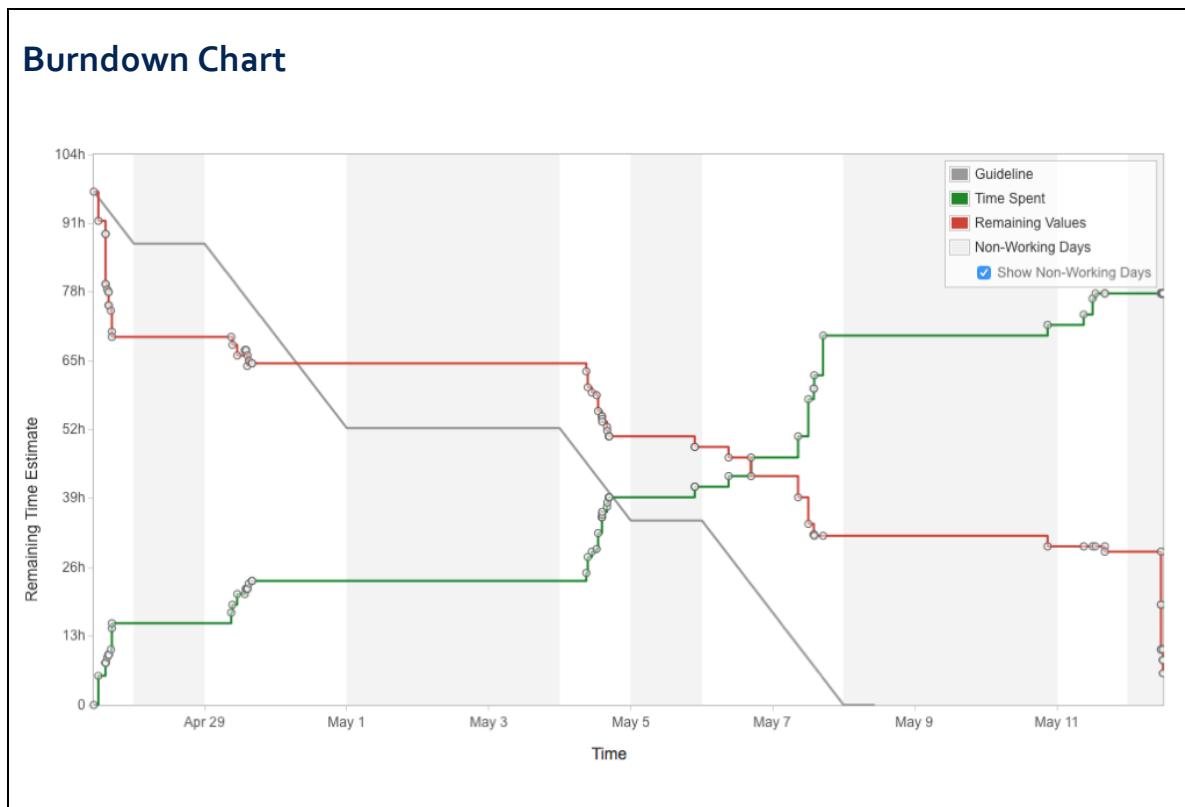
Sprint 5 - Misoptimalisering

Periode:	8.April - 27.April
Story points:	29
Tid pr SP:	4.5 timer
Gjenst��ende arbeidstimer:	31.5 timer (est.)



Sprint 6 - WrapUpApp

Periode:	27.April - 12.Mai
Story points:	7
Tid pr SP:	11.5 timer
Gjenst��ende arbeidstimer:	0 timer (est.)



6.2 USER STORIES

6.2.1 USER STORIES FØR PROSJEKT START:

SP - Story points

Funksjonalitet	SP
Grafisk brukergrensesnitt	36.5
Kart	2
Transitions	5
Skjermbilder	
- Innlogging	1/2
- Ruteliste	5
- Scanning	3
- Arbeidsliste (Uten kart-delen)	8
- Signaturvindu + signaturlibrary	3
- Unified look på lister/meny og design av ikon/bilder (3 + 5 + 2)	10
Ruteliste	5
Hente rute- og pakkeinformasjon	1

Fjerne pakker fra liste	1
Drag-and-drop listejustering	3
Arbeidsliste	19,5/8
Vise detaljert informasjon om kunde/pakke	1/2
Fjerne pakker	3
Melde avvik	
Kart	8
Ring/sms til kunde	1
- Ta høyde for tidsestimat	2
Lever pakke	
- Upload-funksjon	5
- Signatur-funksjon	
- Levering uten kvittering	
Innscanning	4 / 3
Strekkode og QR	

Legge til pakker manuelt med kollinummer	1
Ruteoptimalisering	18+
Google Directions api	0
Drag-and-drop rutejustering	5
Ta høyde for endringer av tidssensitive pakker	13-?
Hjelpefunksjoner	3
Hjelp-knapp	1
Ring kjøreleder-knapp	
Nødsituasjons-knapp	2
Hastepakker/adhoc	11
Spontane henting pop-up	8
Overlevere pakker mellom sjåfører	3
Ekstra	1
Lydeffekter	1
Animasjoner	

Opprett henting	2
Søkefunksjon for å finne kunde	2

6.2.2 USER STORIES ETTER PROSJEKT SLUTT:

7.

Task-ID	Navn	101 SP
OPHT-189	Dele opp ruter i skift (natt/dag) med dato	2
OPHT-182	Code Cleanup!	3
OPHT-156	Dialog for levering. (I Android 5.0 stil)	1
OPHT-159	Avvik-Card	2
OPHT-154	Vise informasjon om task og liste av kolli	3
OPHT-145	FAB og navigering	2
OPHT-139	Ruteliste - Kart	2
OPHT-135	Backend kommunikasjon mellom liste, map og view	3

OPHT-134	Ruteliste - Vise detaljer om valgt rute (ikke kart)	2
OPHT-116	Overgang(animasjon) fra skannefragment til arbeidslistefragment	2
OPHT-102	Opprette leveranseliste med sub-items	3
OPHT-97	Implementere kamera-fragment	2
OPHT-99	Manuell scanning av kolli (input av kollinummer)	2
OPHT-98	Implementere scanne-funksjonalitet	5
OPHT-101	Lokal database med SugarORM	3
OPHT-38	Arbeidsliste - Visning	3
OPHT-41	Signatur	5
OPHT-40	Arbeidsliste - Drag and Drop	5
OPHT-43	Pakke-Card	5
OPHT-42	Fullføring av rute	5
OPHT-47	Ruteliste - Liste av valgbare ruter	2

OPHT-46	Ikoner til all funksjonalitet i app-bar	1
OPHT-39	Feilmelding ved feilet innscanning	2
OPHT-45	Ring/SMS	2
OPHT-37	Kart-GUI og funksjonalitet	8
OPHT-21	Definer Fragment Framework	2
OPHT-16	Navdrawer-Meny	3
OPHT-15	Definer Backend Relasjoner	4
OPHT-14	Research	3
OPHT-13	Statistikk om sjåfør/leveringer	2
OPHT-186	Tidssensitive pakker	1
OPHT-63	Fiks flyt. Blokkér flyt i programmet som ikke skal gjøres.	2
OPHT-185	SjårførprofilVindu	2
OPHT-55	Ad-hoc henting	2
OPHT-150	Tutorial	2

OPHT-53	Nødsituasjon-screen	2
---------	---------------------	---

6.3 AKSEPTANSETESTER

- Akseptansetest 11. Mars
 - Introduksjon
 - Innskanning:
 - Arbeidsliste:
 - Kommentarer til test:
 - Inntrykk og forbedringspotensialet:
- Akseptansetest 6. Mai
 - Introduksjon
 - Innlogging:
 - Hovedmeny(Drawer):
 - Ruteliste:
 - Innscanning:
 - Arbeidsliste:
 - Levering:

AKSEPTANSETEST 11. MARS

Deltakere	Anders Ullnæss, Harald Alexander Kulø
-----------	--

INTRODUSKJON

Akseptansetesten har som mål å teste applikasjonens funksjonaliteter. Det er ment at testen skal bli gjennomført av "produkteier"/ansvarlig, så utviklerne kan forsikre seg om at de er på rett vei og at eksisterende funksjonalitet møter de diverse kravene. Eventuelle feil og rettelser vil komme frem under kommentar feltet. Det blir satt pris på evt ekstra kommentarer og innspill i feltene nederst.

FAB = Floating Action Button

Nr.	Test	Beskrivelse	J a	N e i	Kommentar
Innskanning:					
1	Kan du skanne inn en enkelt kolli?	På innskanning skjermen, bruk kamera til å scanne inn qr/barcode.	x		1. Bildet henger fra forrigje scan 2. kule animasjoner 3. intuitivt 4. Fjerne scanner manuelt ?(toggle kamera)
2	Får du sett alle kolli på en enkelt leveranse?	Ved å trykke på et element i listen, utvides feltet til å også vise en liste av kolli. Disse kolliene får en "v" hvis de er scannet inn".	x		1. nederste element er skjult av kamera
3	Kan du manuelt "skanne" inn et kollinummer?	Bruk knappen "Skriv inn kolli" i høyere hjørne til å manuelt skanne kolli.	x		1. litt stygg dropdown ved < 3 elementer
4	Får du startet kamera	Etter kamera blir	x		

	etter at det går i dvale?	aktivert vil en timer starte. Hvis kamera ikke blir brukt til å skanne en pakke på 15 sek(test verdi) går det i dvale. Ved å trykke på knappen midt på skjermen skal kamera starte på nytt.			
5	Får du gått videre fra innskanningen(til arbeidsliste) etter fullført skanning?	Etter en innskanning dukker FAB opp nede i høyere hjørne. Når du vil gå videre fra denne skjermen trykker du på denne knappen. Idag er det en grense på minimum 80% innskannede kolli for å gå videre. (Kan her benytte "Scann alle"-knappen)	x	x	1. Scann alle -> lager en pil som ikke fungerer

Arbeidsliste:

6	Kan du se informasjon om neste stoppested?	Listen på høyere side forteller om alle stopp på denne ruten.	x		1. slå sammen S/H på samme sted
7	Får du oversikt over forventet leveransetidspunkt?	Både i dialogen i venstre hjørne(tid igjen) og listen på høyere side(leveransetidspunkt) er det informasjon om	x		

		avstand til destinasjonen.		
8	Får du "fullført" en leveranse og gått videre til neste på listen?	Når en det er tid for å levere trykkes det på FAB og tider samt aktiv leveranse vil oppdateres(også på kart). Ved neste sprint er det planlagt å legge til en "Leveranse"-skjerm som skal kjøres etter knappen blir trykket på.	x	1. feil waypoint markert
9	Kan du sortere listen etter eget ønske?	Ved å holde på det elementet du ønsker å flytte på aktiveres "edit mode", listeobjekter kan da flyttes. Når du er ferdig med endringene trykker du på FAB for å lagre, ruten oppdateres da.	x	1. Tidene ble feil etter sortering 2. man kan reberegne rute uten å faktisk endre rekkefølge
10	Får du sett utvidet informasjon om alle elementene på arbeidslisten?	Når du trykker på et element i listen åpnet et nytt "card". Dette inneholder informasjon om leveransen/henting og alle kolli som hører med. Her vil også eventuelle avik osv håndteres.	x	

KOMMENTARER TIL TEST:

- Generelt lite bugs
- Bra og intuitivt grensesnitt

INNTRYKK OG FORBEDRINGSPOTENSIALET:

- menyknapp må være hvit
 - litt eye-candy i bakgrunn av innskanning
 - litt dus, smakfult
 - ikon på "Skriv inn kolli"
-

AKSEPTANSETEST 6. MAI

Deltakere	Anders Ullnæss, Lars Andreas Eidsheim, Harald Alexander Kulø
-----------	--

INTRODUSKJON

Akseptansetesten har som mål å teste applikasjonens funksjonalitet. Det er ment at testen skal bli gjennomført av "produkteier"/ansvarlig, så utviklerne kan forsikre seg om at de er på rett vei og at eksisterende funksjonalitet møter de diverse kravene. Eventuelle feil og rettelser vil komme frem under kommentar feltet. Det blir satt pris på evt ekstra kommentarer og innspill i feltene nederst.

FAB = Floating Action Button

Nr.	Test	Beskrivelse	Ja	Nei	Kommentar

INNLOGGING:					
1	Kan du logge inn på Tore Tang sin bruker?	Brukernavn: toret Passord: ang	x		
HOVEDMENY(DRAWER):					
1. Er sidemenyen lett forståelig?					
		Her vil du kunne se brukerprofil, nødsituasjons-skjerm og tutorial.(hjelp)	x		Gode ikoner
RUTELISTE:					
1	Er rutealternativene godt oversiktlig?	Etter du kommer videre fra innlogging blir du ført til rutelista. På venstre side av skjermen	x		
2	Er forskjellen på natt- og dag-ruter tydelig?	Alle ruter i listen har et ikon som forteller hvilken type rute det er.	x		
3	Kan du se utvidet informasjon om hver enkelt rute?	Ved å trykke på en av rutene kommer mer informasjon opp i feltet til høyre for listen. Kartet under vil også vise en preview av den valgte	x		

		ruten.			
4	Kan du gå videre med ønsket rute?	Når du har valgt en rute vil FAB bli synlig ned i høyre hjørne, for å gå videre til innscanning trykker man på denne.	x		

INNSCANNING:

1	I testens sammenheng, bruk "scan alle"		x		
2	Får du avsluttet kamera etter at det har åpnet seg?	I høyre hjørne av kamera er det en knapp for å avslutte scanning.	x		
3	Får du gått videre fra innskanningen(til arbeidsliste) etter fullført skanning?	Etter en innskanning dukker FAB opp nede i høyere hjørne. Når du vil gå videre fra denne skjermen trykker du på denne knappen. I dag er det en grense på minimum 8% innskannede kolli for å gå videre. (Kan her benytte "Scann alle"-knappen)	x		

ARBEIDSLISTE:

1	Får du "fullført" en leveranse og gått videre til neste på listen?	Når en det er tid for å levere trykkes det på FAB og tider samt aktiv leveranse vil oppdateres(også på kart). Ved neste sprint er det planlagt å legge til en "Leveranse"-skjerm som skal kjøres etter knappen blir trykket på.	x		
2	Kan du sortere listen etter eget ønske?	Ved å holde på det elementet du ønsker å flytte på aktiveres "edit mode", listeobjekter kan da flyttes. Når du er ferdig med endringene trykker du på FAB for å lagre, ruten oppdateres da.	x		
3	Det er også en "ad-hoc" GUI implementert. (I testen kan denne vises ved å trykke ad-hoc i actionbar) Denne skal i løpet av i morgen simulere ad-hoc pakker, men i denne sammenheng vil kun GUI være testbart. Gjerne kom med innspill her og! ■ Fin GUI				

LEVERING:

1.	Kan du skanne inn en enkelt kolli?	Bruk kamera for å skanne inn kolli, ved å trykke på kamera-ikonet.	x		
----	------------------------------------	--	---	--	--

2.	Kan du manuelt skrive inn en kolli?	Dersom strek/QR-koden mangler på pakken kan man skrive inn kolli manuelt med å trykke på "blyant"-ikonett	x		Angre-bug
3.	Har du god oversikt over mottaker-informasjon og kolli?	På venstre side av skjermen vises mottaker-informasjon og en liste av kolli for denne leveransen	x		
4.	Kan man levere pakkene med signatur?	Når alle kolli'ene er skanna inn, trykk på FAB for å komme til signatur-viewet.	x		■ signatur ikon (for likt manuel input)
5.	Kan man levere pakkene uten signatur?	Dersom man ikke vil/kan levere med signatur kan man trykke på signatur-ikonet øverst til høyre for å "toggle" mellom levering med/uten signatur	x		
6.	Kan man rapportere et avvik på leveringen?	Dersom noe skulle være galt med leveringen kan man rapportere et avvik med å trykke på trekant-ikonet	x		

Bugs og kommentarer til skjermene:

Ruteliste

- Hva er sol og måne?
- Ikke helt innlysende at grå ruter er fullførte.
- Få med i tutorial at grå ruter er fullførte.
- Tydeligere på høyre side at ruten er fullført.

NavDrawer

- Fancy animasjon.
- Er drivstoff en inntektskilde?
- Skjønner ikke helt hva jeg gjør på nødsituasjonsskjerm.

Arbeidsliste

- Avslutt rute-dialog er litt utydelig. "Avslutt" virker skummelt.
- Tidslukelevering - Kan levere før tidsgrensen nå.
 - Bør ha en advarsel for om man har plassert en pakke utenfor tidssonen.
- Scrolling bør vise at det ligger flere pakker over. (Vise litt av forrige task øverst)
- I pakkecard - Adresse bør være leveringsadresse.
- Leveringsknappen funker ikke til slutt?

Levering

- Trenger signaturfjerning
- Angring av innskannet kolli funker ikke?
- Kommentarer tåler ikke æøå
- Burde ha limit på antall tegn i kommentar for levering uten signatur
- Mer logisk at tilbakeknappen lukker kamera.
- Ser ut til å kunne levere uten signering?
- OOM error på signatur.

Avvik:

- Kategoritekst og dropdown-knapp er for like
- Rapporter skrivefeil

Tutorial:

- Stavefeil
- Bruk bokser for å inndele grå-soner i tutorial? Ikke hardcodet i png(forslag, men tidkrevende)

6.4 DAGBOK

- Prosjektdagbok
 - Dagbok 05.01.2015
 - Dagbok 14.01.2015
 - Dagbok 19.01.2015
 - Dagbok 21.01.2015
 - Dagbok 22.01.2015
 - Dagbok 26.01.2015
 - Dagbok 28.01.2015
 - Dagbok 29.01.2015
 - Dagbok 02.02.2015
 - Dagbok 04.02.2015
 - Dagbok 05.02.2015
 - Dagbok 09.02.2015
 - Dagbok 11.02.2015
 - Dagbok 12.02.2015
 - Dagbok 16.02.2015
 - Dagbok 18.02.2015
 - Dagbok 19.02.2015
 - Dagbok 02.03.2015
 - Dagbok 04.03.2015
 - Dagbok 05.03.2015
 - Dagbok 09.03.2015
 - Dagbok 11.03.2015
 - Dagbok 12.03.2015
 - Dagbok 16.03.2015
 - Dagbok 24.03.2015
 - Dagbok 01.04.2015
 - Dagbok 08.04.2015
 - Dagbok 09.04.2015
 - Dagbok 13.04.2015
 - Dagbok 20.04.2015
 - Dagbok 22.04.2015
 - Dagbok 27.04.2015
 - Dagbok 11.05.2015
 - Dagbok 04.05.2015
 - Dagbok 06.05.2015
 - Dagbok 07.05.2015

- Dagbok 11.05.2015

DAGBOK 05.01.2015

I dag var første arbeidsdag for prosjektet. Vi var hos Mesan og fikk en grundig beskrivelse av prosjektoppgaven og omfanget. Det ble ikke så veldig mye jobbing den dagen grunnet innføring av oppgaven/prosjektet og introduksjoner for nøkkelpersoner for prosjektet.

DAGBOK 14.01.2015

I dag hadde vi først et møte om Postnords nåværende løsning for håndterminalen. Vi fikk en grundigere forståelse i hvordan systemet fungerer i dag, samt hvordan arbeidsprosessen til sjåførene er. Dette var nyttig kunnskap for at vi skal kunne lage et produkt som skal kunne tilfredsstille sjåførene selv.

Senere hadde vi et møte med innføring i Confluence og Jira. Det vi lærte vil være av god nytte for prosjektføringen for resten av prosjektet.

DAGBOK 19.01.2015

Dato:	19. Januar 2015	
	<p>I dag hadde vi møte med veileder Geir fra HiOA. Vi diskuterte og klarerte flere punkter angående dokumentasjon.</p> <p>Vi har også jobbet med forskjellige deler av dokumentasjonen, bl.a:</p> <ol style="list-style-type: none">1. standardisere møtereferatet2. opprettelse av dagbok3. risikoanalyse4. kravspesifikasjon5. mer forprosjekt6. Vi har også planlagt hva som skal gjøres de neste dagene.7. Ellers har det vært en del korrespondanse via mail til IT-	<p>1. Martin Hagen</p> <p>2. Sondre Sparby Boge, Martin Hagen, Lars-Erik Kasin</p> <p>3. Joakim Rishaug</p> <p>4. Sondre Sparby Boge, Martin Hagen, Lars-Erik Kasin, Joakim Rishaug</p> <p>5. Joakim Rishaug</p>

	<p>ansvarlig, og Christoffer i løpet av dagen.</p> <ol style="list-style-type: none"> 8. Vi har med dette sikret oss tilgang til et redskap som blir brukt for mockups av interface, og den interne github-kontoen til Mesan. 9. kontoen til Mesan. 	
Problemer:	Ingen store problemer, en del oppklaring/diskusjon om oppgaven.	
Til neste gang:	<p>Oversikt over funksjonalitet til "planning poker".</p> <p>Ha klart tanker rundt teknologivalg til møte torsdag.</p>	

DAGBOK 21.01.2015

Dato:	21. Januar 2015
	<p>I dag jobba vi hovedsakelig kun med:</p> <ol style="list-style-type: none"> 1. Dokumentasjon 2. Planning poker 3. I dokumentasjons-delen gjorde vi ferdig kravspesifikasjonen, 4. og fylte ut arbeidsplanen som vi så skulle bruke i "planning poker". 5. Vi lagde også mock-ups (skisser) for de ulike skermene for å ha en 6. visuell indikasjon på hva vi forestiller oss hvordan løsningen skal være. 7. 8. Vi var stort sett enige om antall story-points vi skulle gi til de

	9. forskjellige funksjonalitetene.
Problemer:	Ingen store problemer. Martin var borte denne dagen, grunnet sykdom i familie.
Til neste gang:	Ha klart tanker rundt teknologivalg til møte torsdag.

DAGBOK 22.01.2015

Dato:	22. Januar 2015
	<p>I dag hadde vi et møte med Anders der vi gikk gjennom</p> <ul style="list-style-type: none"> 1. Funksjonalitet 2. User-stories 3. Planning poker 4. Teknologi-valg 5. 6. Anders var enig i det meste vi hadde gjort under de 3 første punktene. 7. Mot slutten av møtet satte vi opp arbeidsmiljøet med hjelp av Anders. 8. Vi lagde innloggings-siden i applikasjonen for å teste noe av 9. teknologien Anders anbefalte oss å prøve. 10. 11. Vi la også inn tasks og user-stories i JIRA, og skrev ferdig 12. forprosjekt-dokumentasjonen. 13. Etter lunsj samarbeidet vi om å lage en fremdriftsplan med oversikt over fokusområder for 14. sprintene de neste ukene. Gjorde det først på tavle på kursrommet, og ført inn i et 15. gantt-diagram.
Problemer:	Ingen store problemer. Martin var borte denne dagen, grunnet

	sykdom i familie.
Til neste gang:	Gå gjennom og høre om kravspesifikasjon og user-stories er i samsvar til forventninger.

DAGBOK 26.01.2015

Dato:	26. Januar 2015
	<p>Startet dagen med at Joakim skrev notater om valg av device, Sondre justerte på kravspesifikasjonen, og Martin og Lars-Erik jobbet med å definere klasser og relasjoner for backenden. Dette ble etterhvert alle med på.</p> <p>Vi snakket også så vidt med Harald, og ble enige om å sende han alt av dokumentasjon og møtes på onsdag.</p> <p>Etter lunsj fordelede vi oppgaver og begynte å jobbe på fragmentmal for de ulike layoutene og utkast til menyen i appen, lese om Google-API og undersøke Spider (ruteoptimaliseringssoftware som Harald nevnte).</p> <p>Konkluderte med at Spider var mer egnet for desktop/backend-delen av systemet, og ikke noe vi vil implementere i vår app.</p> <p>Hadde et kort standupmøte vedrørende strukturen på appen, da Android 5.0 ser ut til å være tungt basert på fragments, og ikke activities. Bestemte oss for å sikte oss inn på å bruke fragments mest mulig, men tenker å diskutere dette med Anders, når han er tilbake.</p>

	Jobbet resten av dagen videre med fragments, meny og utvidbare lister, i tillegg til research rundt google API.
Problemer:	Martin var ikke her forrige uke, og måtte dermed sette opp utviklingsmiljøet. Støtte på en del problemer med Android Studio, og brukte store deler av dagen på fikse dette.
Til neste gang:	Gå gjennom og høre om kravspesifikasjon og user-stories er i samsvar til forventninger.

DAGBOK 28.01.2015

Dato:	28. Januar 2015
	<p>Startet dagen som vi avsluttet forrige hvor alle hadde egne oppgaver. Lars-Erik jobbet med arbeidsliste, Joakim med meny, Martin med fragments og Sondre undersøkte rundt ruteoptimalisering. Vi har også gjort flere oppdateringer og opplastninger på wiki, med bl.a fremdriftsplan.</p> <p>Vi hadde også et møte med Harald og Anders hvor vi snakket om statusen til prosjektet og planen videre.</p> <p>Hadde et kort standupmøte etter lunsj hvor vi diskuterte det vi hadde merket oss gjennom dagens arbeid og hvordan dette påvirker valgene videre. Sondre hadde bl.a funnet noen begrensinger i kart-Api'et. Api-kall vil ta lengre tid basert på hvor detaljert ruten vi får ut skal være, noe vi tenker å jobbe dynamisk rundt, ved for eksempel å kjøre et udetaljert kall først, og deretter et mer detaljert et.</p>

	<p>Vi bestemte oss her for å godta begrensningene og fortsette som planlagt, da vi mener at produktet ikke vil ta betydelig skade av dette.</p> <p>Resten av dagen fortsatte vi med våre arbeidsoppgaver i pre-sprinten.</p>
Problemer:	Møtt flere små og store utfordringer med Android Studio og Android 5.0 som har tatt tid å løse.
Til neste gang:	Fortsette med pre-sprint.

DAGBOK 29.01.2015

Dato:	29. Januar 2015
	<p>På starten av dagen jobbet alle noe med sine egne oppgaver fra igår, før vi gikk over til å lese Googles Material design dokument. Her så vi gjennom retningslinjer og designråd.</p> <p>Hadde et kort standupmøte før lunsj hvor vi diskuterte designvalg for prosjektet og hva vi hadde fått ut av "Material design" dokumentet.</p> <p>Vi ble også enige om å gjøre noen endringer i arbeidsliste-fragmentet for å utnytte flere designprinsipper samtidig som vi unngår unødvendig kompleksitet i selve lista. Her er det nå tenkt å bruke en ny "dialog" ved klick på hver enkel pakke. All ekstra informasjon og valgmuligheter vil legges her.</p> <p>Resten av dagen har gått med til å forbre første sprint og gjøre mer research rundt teknologi og design.</p>
Problemer:	Lars-Erik og Martin har mistet tilgang til git og har derfor brukt mye tid av dagen på å fikse dette. Problemet er ikke løst og tatt opp med Lars Reed.

Til neste gang:	Klargjøring til første sprint.
------------------------	--------------------------------

DAGBOK 02.02.2015

Dato:	02. Februar 2015
	<p>Siden dette var første dag i første sprinten hadde vi et kort standupmøte på starten av dagen. Her gikk vi gjennom hva vi har gjort siden forrige torsdag, hva vi skal idag og evt problemer med dette. Joakim er scrummaster denne sprinten.</p> <p>Etter møtet jobbet alle med sine egne oppgaver.</p> <p>Etter lunsj møtte vi med Knut fra Postnord-teamet. Han gikk gjennom bruken av JIRA og hvordan vi videre skulle sette opp prosjektet vårt der. Vi satt så igjen og opprettet nye stories/taskses før vi offisielt startet den første sprinten.</p> <p>Resten av dagen jobbet vi videre med taskene i sprinten.</p>
Problemer:	Sondre har mistet tilgang til git og har derfor brukt mye tid av dagen på å fikse dette. Problemene er ikke løst og tatt opp med Lars Reed.
Til neste gang:	Arbeide med satte oppgaver

DAGBOK 04.02.2015

Dato:	04. Februar 2015
	<p>Vi startet dagen med et kort standupmøte, og gikk fort videre til å arbeide hver for oss med våre oppgaver. Martin ble ferdig med malene for fragmentene og implementasjon av disse, Sondre fikk</p>

	<p>implementereret google maps api i løsningen og Lars-Erik og Joakim nærmer seg en ferdig verjson av sine oppgaver.</p> <p>Vi fikk også delt ut tabelten som skal brukes idag. En Samsung Galaxy Tab S, 8.4". I lunsj og noe tid etter testet vi tableten, men vi kunne ikke installere appen da Android 5.0 har ikke kommet ut enda.</p> <p>Resten av dagen har vi jobbet videre med tasks fra JIRA.</p>
Problemer:	Foruten noen små git problemer og vanskeligheter med google maps api gikk dagen relativt smertefritt.
Til neste gang:	Arbeide med satte oppgaver

DAGBOK 05.02.2015

Dato:	05. Februar 2015
	<p>Vi startet dagen med et kort standupmøte, og gikk fort videre til å arbeide hver for oss med våre oppgaver. Siden siste møte har alle blitt "ferdig" med sine task, og vi diskuterte hvordan vi skulle jobbe videre. Sondre kom ikke i møte idag, siden han var på hioa for å møte Geir. Geir var desverre ikke på skolen idag(Sondre dro fra skolen igjen 09.45) så vi må henne avtalen han har signert en annen dag.</p> <p>Vi har gått mer gjennom arbeidsliste fragmentet og innholdet vi skal ha med der + leveransecardet.</p> <p>Resten av dagen har vi jobbet videre med tasks fra JIRA.</p>
Problemer:	Ingen store problemer idag.
Til neste	Arbeide med satte oppgaver

gang:	
--------------	--

DAGBOK 09.02.2015

Dato:	09. Februar 2015
	<p>Vi startet dagen med et kort standupmøte, og gikk fort videre til å arbeide hver for oss med våre oppgaver.</p> <p>Martin og Joakim var på møte hos Geir, der vi ble enige om at det ikke skulle bli jevnlige møter med han videre.</p> <p>Martin og Joakim hadde også et internt møte rundt klasser til leveranse-kortene. Martin lagde disse klassene etterpå, og jobbet så med floating actionbar. Joakim jobbet med Transitions. Lars-Erik jobbet med videre med arbeidslista, mest bugfix. Sondre jobbet videre med Google Maps og Directions api'ene, og senere noe på arbiedslista.</p>
Problemer:	Joakim fant idag ut at planen med transitions og hvordan vi har lagt opp strukturen i oppgaven ikke kommer til å fungere om hverandre. Dette skyldes at transitions ikke støtter fragments på måten vi ønsker. Joakim har brukt mye tid å finne veier rundt dette, men det ser ut som om vi må gjøre noen endringer i strukturen for å løse dette. Joakim og Martin skal se på dette i morgen.
Til neste gang:	Arbeide med satte oppgaver

DAGBOK 11.02.2015

Dato:	11. Februar 2015
--------------	------------------

	<p>Dagen startet med det vanlige standupmøtet. Vi hadde alle mye å gjøre, så vi startet rett etter. Joakim fortsatte med å finne løsninger på transitions problemet som oppsto på mandag. Martin jobbet med profil delen i drawern. Sondre jobbet videre med maps, og Lars-Erik jobbet med autoscrolling av lista.</p> <p>Etter lunsj jobbet Joakim og Martin sammen for å finne løsninger på transitions. De satt resten av dagen med dette, men har nå funnet en mulig løsning. Det innebærer å flytte pakkecard til en egen activity og animere overgangen etter beste evne. De skal prøve videre i morgen.</p> <p>Lars-Erik har brukte mye av dagen på sin liste og autoscrolling, men ser ut til å ha kommet et stykke framover. Sondre har idag fått implementert flere funksjoner i kartet, som en "ekte" rute, gps og next leg.</p>
Problemer:	Mye bugs som har kommet under arbeid idag, men ikke noe vi ikke har løst.
Til neste gang:	Arbeide med satte oppgaver

DAGBOK 12.02.2015

Dato:	12. Februar 2015
	<p>Vi startet dagen med standupmøtet. Martin og Joakim gikk så igang med transitions, mens Sondre og Lars-Erik fortsatte med kart og liste respektivt. Lars-Erik har også fikset mer på lista hjemme til idag, og har nå en liste hvor du kan utvide neste levering. Lista og kartet samhandler nå, og du kan trykke "neste levering" og både kart og liste oppdateres.</p> <p>Joakim og Martin har "fikset" transitions, og har en fungerende løsning oppå selv med noen bugs. Dette vil bli implementert i hoved</p>

	branch'en neste uke.
Problemer:	Ingen store problemer
Til neste gang:	Arbeide med satte oppgaver

DAGBOK 16.02.2015

Dato:	16. Februar 2015
	<p>Denne uken er Joakim borte, så vi andre deler rollen som scrummaster og forsetter ellers som vanlig.</p> <p>Vi tre startet dagen med det vandlige standupmøte. Etter en merge av alt vi drev med frem til idag gikk vi også inn på en møterom og disuterte hvordan denne uken skulle bli sende ut.</p> <p>Vi ryddet opp i taskene på JIRA og bestemte hvem som starter med hvilke oppgaver. Etter disse møtene gikk vi igan med arbeidet.</p> <p>Alle har idag ryddet opp i kode og lagd flere kommentarer. Lars-Erik har etter det gjort flere forbedringer på både logg inn og pakkecard. Sondre har idag fikset bugs med kart og rute, og justert der det skulle trenges. Martin har gjort flere tilpassninger på GUI til appen, laget liste over kolli på en levering og laget en header til kartet, som Sondre etterpå la til funksjonalitet på.</p> <p>Idag har Anders vært i bygget og hjulpet oss med flere småting underveis. Vi har også avtalt møte neste mandag hvor vi skal få en innføring i barcodes og qr.</p>
Problemer:	Ingen problemer, men Joakim er borte denne uken.

Til neste gang:	Arbeide med satte oppgaver
------------------------	----------------------------

DAGBOK 18.02.2015

Dato:	18. Februar 2015
	<p>Vi tre startet dagen med det vandlige standupmøte. Siden dette er siste uka i denne sprinten hadde vi et ganske godt bilde på ting som måtte gjøres, så vi startet med en gang med de oppgavene som var igjen. Sondre fortsatte å jobbe med kartet, og utførte div cleanup etter dete. Lars-Erik la til autoscroll på arbeidsliste, sørget for at lista startet på rett task etter optimalisering, og jobbet med innhold i task(la til lister med pakker). Martin fikset kollicardet til en levering og la til diverse ringe og sms funksjoner. Av disse funksjonene var det sms servicen som tok mest tid.</p> <p>Etter lunsj fortsatte vi med oppgavene våre.</p>
Problemer:	Ingen problemer, men Joakim er borte denne uken.
Til neste gang:	Arbeide med satte oppgaver

DAGBOK 19.02.2015

Dato:	19. Februar 2015
	<p>Dagen idag er siste i denne sprinten, og under dagens standupmøte planla vi noe hvordan neste uke skal se ut. Vi ble enige om å ta et "analysemøte" når Joakim kommer tilbake, så vi kan gå gjennom hva vi gjorde rett og hva vi kan forbedre til de neste sprintene.</p>

	Resten av dagen gikk med til mye finpuss på eksiterende løsninger, og cleanup av kode. Vi har også nå erstattet "alle" verdier i appen og laget disse mer realistiske. Pakkecardet har også fått alle felter utfylt.
Problemer:	Ingen problemer, men Joakim er borte denne uken.
Til neste gang:	Arbeide med satte oppgaver

DAGBOK 02.03.2015

Dato:	2. Mars 2015
	<p>I forrige uke ble det litt indre strid om hvem som skulle skrive dagboken. Dette endte med at den ikke ble skrevet i det hele tatt, noe som er beklagelig.</p> <p>En kort oppsummering av forrige uke: Vi startet sprint 2: "Innscanningsvindu", hvor vi lager vinduet i programmet hvor vi scanner inn barcodes som skal representere pakkene Postnord leverer.</p> <p>På dette tidspunktet har vi implementert nesten all funksjonalitet, og det trengs hovedsaklig optimalisering og finpuss av vinduet. Derfor har vi også begynt med unit-tests siden det var tid til overs.</p> <p>Det har også blitt lagt til en intern database med SQL i bunn, kalt Sugar ORM.</p>
Problemer:	Vi skrev ikke dagbøker forrige uke.
Til neste gang:	<p>Optimalisere kamera for scanning av barcodes.</p> <p>Spør Richard om JIRA rapporter og estimering av storypoints.</p>

DAGBOK 04.03.2015

Dato:	4. Mars 2015
	<p>Arbeidsdagen gikk stort sett ganske greit. Alle møtte på litt motgang på sine oppgaver, men alle klarte likevel å gjøre mye fremgang. Mesteparten av funksjonalitet er på plass. Optimalisering for scanning er ferdig. Litt finpuss med animasjoner og transitions gjenstår.</p> <p>Lars-Erik jobbet med å omstrukturere koden rundt Google-kallene og transitions mellom scanne-vindu og TaskList-vindu. Martin og Joakim jobbet begge med FloatingActionButton. Hovedsakelig hvor den skal plasseres og hvordan den skal animeres. Sondre jobbet med å optimalisere kamera-scanning.</p> <p>Prosjektet er fortsatt på skinner og vi ser ut til å bli ferdig med alle story'ene innen sprinten er omme.</p>
Problemer:	Mye tid gikk til å fikse problemer med kamera mellom Android Studio og Genymotion. Animasjoner har vært ganske vanskelig å få til å oppføre seg akkurat slik som ønsket.
Til neste gang:	Spør Richard om JIRA rapporter og estimering av storypoints.

DAGBOK 05.03.2015

Dato:	5. Mars 2015
	<p>Dagen har gått forholdsvis greit, til tross for mye bugs i den avsluttende fasen. Prosjektet ser ut til å gå etter planen, og vi føler oss komfortable med å avslutte sprint 2 i dag.</p> <p>Martin har jobbet med animasjon på FAB og kamera, og knyttet dette opp mot kamerafragmentet, i tillegg til å knyte de rette metodene til</p>

	<p>FAB.</p> <p>Joakim har jobbet med animasjon sammen med Martin, en del javadoc, og til slutt jobbet med kodenstruktur og bugfixes.</p> <p>Sondre har jobbet litt med dokumentasjon, kommentarer og bugfixes hele dagen.</p> <p>Lars-Erik har jobbet mest med transition mellom Scanning og Arbeidslista, men brukte de siste timene på bugfixes og å knytte sammen de ulike delene etter merge.</p>
Problemer:	Mye småfeil og bugs har preget dagen. Lars-Erik slet mye med at content transition ikke fungerte, men fant til slutt løsninger.
Til neste gang:	Planlegge sprintmøte, og tenke på tasks og oppgaver til neste sprint.

DAGBOK 09.03.2015

Dato:	9. Mars 2015
	<p>I dag startet vi på tredje sprint, i disse to ukene er det ruteliste og levering som er hovedfokuset. I forbindelse med dette brukte vi de første timene av dagen på sprint oppstart. Her ble vi enig over flere desig/funksjonalitet valg, opprettet stories og tasks(til JIRA) og planla arbeidet videre.</p> <p>Når møtet var ferdig gikk vi igang med forbredelser og de nye oppgavene.</p> <p>Lars-Erik gikk igang med grunnarbeidet for ruteliste aktiviteten og lagde nesten ferdig listen over ruter.</p> <p>Sondre har fikset noen mindre bugs med kartet, og senere har han hjulpet Joakim som har satt igang med diverse opprydding og</p>

	<p>effektivisering av kode.</p> <p>Martin gjorde en del opprydding og effektivisering i helgen i de klassene han jobbet i sist. Etter møtet har han også laget akseptansetesten som Harald Alexander Kulø og/eller Anders Ullnæss skal gå gjennom på kommende onsdag.</p>
Problemer:	Ingen problemer i dag
Til neste gang:	Forberede akseptansetest til onsdag.

DAGBOK 11.03.2015

Dato:	11. Mars 2015
	<p>Vi startet dagen med en akseptansetest med Harald Alexander Kulø og Anders Ullnæss.</p> <p>Den gikk stort sett ganske greit. Det var noen bugs i appen, men det var forventet på nåværende tidspunkt. Utenom disse bugs'ene var appen særdeles godt mottatt.</p> <p>Vi hadde også et kort møte med Richard Busch Nordli om Jira, dokumentasjon og sprintrapport.</p> <p>Både Martin og Joakim jobbet med sprintrapport og akseptansetestrapport. Martin jobbet så med Leverings-aktiviteten, og Joakim senere med ruteobjekt for databasen og integrasjon av den.</p> <p>Både Lars-Erik og Sondre jobbet med Ruteliste-aktiviteten. Senere jobbet Lars-Erik mer med lista og informasjons vinduet i aktiviteten, og transitionen mellom RuteListe-aktiviteten og Scanne-aktiviteten. Sondre implementerte Google Maps og Google Directions API'et i</p>

	RuteListe-aktiveten.
Problemer:	Ingen problemer i dag
Til neste gang:	Fikse database og liste interaksjon

DAGBOK 12.03.2015

Dato:	12. Mars 2015
	<p>I dag var første dag av ny sprint hvor alle hadde arbeidsoppgaver fra start av, og det ikke var behov for nye planleggingsmøter.</p> <p>Vi starter som alltid dagen med et stadupmøte, og gikk raskt videre til våre egne arbeidsoppgaver.</p> <p>Sondre startet med dialogen for "manuell kolli" og flyttet flere metoder ut i en ny util. Han startet så på signatur fragmentet.</p> <p>Martin har idag jobbet med den nye activity'n, DeliveryActivity, og lagt inn flere av elemente som skal inn, bla. kamera, metoder for scanning og informasjons felter Joakim har arbeidet med databasedelen av applikasjonen idag, hovedsaklig det å lagre/hente en rute i databasen. Dette vil også gjøre at vi kan ha flere ruter samtidig.</p>
Problemer:	Ingen problemer i dag, Lars-Erik var ikke her idag men er tilbake til mandag.
Til neste gang:	Videre arbeid.

DAGBOK 16.03.2015

Dato:	16. Mars 2015
	<p>Som vanlig startet vi dagen med et standupmøte og alle hadde sine oppgaver å jobbe med, så vi satte fort i gang med disse.</p> <p>Dagen ble forøvrig sterkt preget av vanskelig feil og bugs, og vi ble sittende lenge for å løse disse, med varierende hell.</p> <p>Sondre har jobbet videre med signaturfunksjonalitet. Fikset først cropping av signaturbildet og jobbet så videre med utseende for signaturer.</p> <p>Martin jobbet med å koble sammen scanner og liste i deliveryactivity, og en del clean up, og satt etterhvert med dialogstyling og loading FAB-animasjon.</p> <p>Joakim har jobbet med DB, og fikset først generering av flere ruter til rutelista, og satt videre med bugs og prøvde å kombinere gammel kode med ny kode, som ikke funket så bra overens.</p> <p>Lars-Erik jobbet først med å koble deliveryactivity sammen inn i flyten i programmet og koblet det til et faktisk task-objekt. Jobbet så med transitions, og møtte her på rare bugs, som ikke ble oppklart.</p>
Problemer:	Ingen problemer utenom vanskelige programatiske utfordringer. SharedElementTransitions skaper rare feil på tilsynelatende tilfeldige måter, og appen sliter med å sende rundt ruteobjekter mellom activities, noe som førte til at vi måtte prøve å legge om databasen. Dette ble svært avansert da mye tidligere kode ble bygget på litt feil premisser.
Til neste gang:	Tenke på løsninger for ulike bugs.

Dato:	24. Mars 2015
	<p>I dag fullførte vi offisielt "Sprint 3 - Levering, Ruteliste". Vi ble ikke helt ferdig med alle task'ene fra den sprinten. Dette gjelder hovedsakelig tasks for transitions og avvik.</p> <p>Vi begynte dagen med å begynne å lage presentasjonen til mesanmøtet i morgen, den 25.03.2015. Dette tok noen timer og alle var med på dette.</p> <p>Senere startet vi "Sprint 4 - Cleanup", der hovedfokuset er på å rydde opp i kode, struktur og kommentarer. Vi jobbet videre med å bli ferdig med tasks fra forrige sprint og andre bugfikser, samt de nye tasks fra den nye sprinten.</p> <p>Lars-Erik jobbet mest med å fikse bugs fra akseptansetesten fra forrige uke. Martin jobbet hovedsakelig med avvik. Joakim fikset og skrev kommentarer rundt om i koden. Sondre la til Google Distance Matrix dagen før, og i dag skrev stort sett kommentarer og fiksa småbugs.</p>
Problemer:	Noen mindre tasks fra forrige sprint er ikke fullført. Ellers ingen store problemer.
Til neste gang:	Bli ferdig med presentasjonen, og tasks fra forrige sprint.

Dato:	1. April 2015	
	<p>Avgjørelse ble tatt om å lage et eget fragment for "levering uten kvittering".</p> <p>I dette fragmentet vil brukeren kunne ta et bilde og skrive med beskrivende tekst for leveransen, men det vil ikke lagres noe info annet enn at pakken er levert. Dette vil ta tid å implementere, men gir lite tilbake i wow-faktor.</p>	1. Martin Hagen
Problemer:	Ingen store problemer, en del oppklaring/diskusjon om oppgaven.	
Til neste gang:	<p>Oversikt over funksjonalitet til "planning poker".</p> <p>Ha klart tanker rundt teknologivalg til møte torsdag.</p>	

DAGBOK 08.04.2015

Dato:	8. April 2015	
	<p>Idag har vi startet for fullt igjen, etter par uker med mye arbeid med det andre faget som kjører parallellt i år.</p> <p>Vi startet dagen med et lagt møte. På dette møtet gikk vi gjennom innholdet i denne sprinten vi starter idag, la til nye punkter og diskuterte nye løsninger.</p> <p>Vi gikk også gjennom punktene fra forrige akseptansetest og sjekket at disse var fikset.</p> <p>Etter lunsj startet vi arbeidet med den nye sprinten.</p>	

	<p>Lars-Erik har fikset jobbet med flyten til programmet og hvordan den skal avsluttes, og jobbet med forskjellige bug i eksisterende løsning.</p> <p>Joakim har ordnet noen ikoner til valgene i actionbaren. Han har også sett på løsninger, sammen med Lars-Erik, til transitionsproblemene vi har hatt en stund(ingen løsning funnet enda).</p> <p>Martin har jobbet med avvik-cardet og gjort noen endringer der. Etter dette laget han løsningen for å kunne levere en task uten signatur. Har også sett på flere mindre bugs.</p> <p>Sondre har etter lunsj hatt dataproblemer, men har sett på noen bugs i koden.</p>
Avgjørelser:	<p>Et punkt til forbedring fra aksseptansestest var "Bør flere leveringer til samme adresse bli én task?".</p> <p>Vi ble enige om at dette ikke vil ha noe mening da vi fra før annser en task som en samling av pakker som skal til denne adressen.</p> <p>Dette blir derfor bare et "problem" nå under testing, siden vi nå bruker tilfeldige adresser og kan få flere tasks på samme.</p>
Problemer:	<p>Ingen store problemer, en del oppklaring/diskusjon om oppgaven. Sondre har slitt med Android Studio idag, og har derfor ikke fått jobbet med andre oppgaver mye etter lunsj.</p>
Til neste gang:	

Dato:	9. April 2015
	<p>I dag jobbet vi videre med oppgavene. Det var ingen store hendelser eller avgjørelser. Mesteparten av dagen var bare ren jobbing med tasks og issues.</p> <p>Joakim var her ikke i dag.</p> <p>Martin gjorde ferdig avvikcard, la til lukk-kamera knapp, endret en del i utility-klassene, og begynte på en progressbar til FloatingActionButton.</p> <p>Lars-Erik jobbet med scrolling til taskList, la til ikoner og annet styling i navigasjonsdrawer'en, og begynte på overlays og hjelpefunksjoner.</p> <p>Sondre fiksa dependencie-issue i gradle, og ellers på ruteoptimaliseringsalgoritmen.</p>
Avgjørelser:	
Problemer:	Ingen store problemer.
Til neste gang:	

DAGBOK 13.04.2015

Dato:	13. April 2015
	<p>Stort sett normal dag. Jobbet videre med våre utdelte oppgavene, og støtte ikke på nevneverdige problemer.</p>

	<p>Joakim og Martin var ikke tilstede, grunnet andre forpliktelser.</p> <p>Lars-Erik jobbet med diverse småting, som visning av skift og tidssensitive pakker i lister, bakgrunn i kameraview, litt tutorials og starten på sjåførinfoindu.</p> <p>Sondre jobbet mye med ruteoptimalisering med hensyn til tidssensitive pakker.</p>	
Avgjørelser:		
Problemer:	Ingen store problemer.	
Til neste gang:		

Dato:	16. April 2015	
	<p>Joakim har jobbet videre med statistikk for sjåfører, og øvrige oppgaver i profilvinduet. Martin har også hovedsakelig jobbet med profilvinduet, da særlig med visning av statistikk.</p> <p>Sondre har jobbet videre med ruteoptimalisering og justeringer i maps.</p> <p>Lars-Erik har jobbet med implementering av Ad-Hoc-henting, som nå lagres og optimaliseres i kart, ruteliste og arbeidsliste.</p>	
Avgjørelser:	Bestemte oss for å ha med grafer for statistikk i profilvinduet, da dette gir en oversiktlig presentasjon av info, samt gir litt krydder	

	til appens utseende.	
Problemer:	Google Distance Matrix API har en call-limit, som stadig toppes når vi jobber med optimalisering.	
Til neste gang:		

Dato:	15. April 2015	
	<p>Joakim og Sondre har sett videre på ruteoptimalisering og bestemt at veien videre for prosjektet er en av 3 valg listet under.</p> <p>Statistikk og bruker ble lagt til i databasen. Martin har laget en progress-bar som er koblet til FAB'en vår, istedet for en overlaying dialog med spinner som vi hadde før. Sondre satt en liten time med å ikke la user gå videre i en "fullført rute". Lars-Erik har sittet med dialog for ad-hoc pakker og lister.</p>	
Avgjørelser:	<p>Plan videre for ruteoptimalisering:</p> <ul style="list-style-type: none"> ● Plan A: Brute-Force løs en rute med $8!$ muligheter. ● Plan B: Implementer en viss måte å ta høyde for tidssensitive pakker i gammel løsning. ● Plan C: Ikke ta høyde for tidssensitive pakker. 	
Problemer:	<p>Hvilken metode burde brukes for ruteoptimalisering? Mye graf-teori, og ingen vitenskaplig enighet om temaet generelt.</p> <p>Sondre møtte flere problemer med ruteoptimalisering i løpet av dagen, vanskelig tema.</p>	

Til neste gang:	Implementer statistikk-telling i programmet ved fullførte leveringer og rute.
------------------------	---

DAGBOK 20.04.2015

Dato:	20. April 2015
	<p>Vi starta dagen med et stand-up møte. Ingen store avgjørelser ble gjort da.</p> <p>Etter lunsj hadde vi et møte der vi ble enige om at vi ikke hadde tid til å fikse ruteoptimaliseringen, og kom fram til et alternativ som fortsatt fremhever idéen ruteoptimaliseringen skulle vise.</p> <p>Joakim jobbet hovedsakelig med grafer og statistikk.</p> <p>Lars-erik jobbet med lister og layout for Ad-hoc.</p> <p>Sondre fikset placeholder-funksjonaliteten, som erstatter ruteoptimaliseringa, for visning av tidsrom for pakker. Han jobbet så siden med statistikk og litt bugfiksing.</p> <p>Martin var ikke her i dag.</p>
Avgjørelser:	Etter mye problemer ble det avgjort at vi ikke kan ta med tidssensitive pakker fullstendig. Dette vil istedet bli implementert som en slags placeholder-funksjonalitet hvor det i ettertid av directions-call vil settes tidsrom for pakken.
Problemer:	Ruteoptimalisering er for avansert til å implementere i løpet tiden vi har på oss.
Til neste	Begynne på tutorial-skjermene. Finne en måte å vise flere soyler

gang:	med forskjellig type informasjon, i samme søyle-diagram.	
--------------	--	--

DAGBOK 22.04.2015

Dato:	22. April 2015	
	<p>I dag har Sondre lagd ikoner til nødsituasjon-activityen, samt påbegynt activity. Joakim har jobbet med bugs med statistikk og tutorial-overlays. Lars-Erik har stort sett jobbet med ad-hoc-pakkers funksjonalitet i samhold med arbeidslista.</p> <p>Martin har lagd en forbedret graf for userprofile, samt nødsituasjon-screen og flyten i programmet (mtp hvor user kan manøvrere seg).</p>	
Avgjørelser:	Nødsituasjon-skjermen ble påbegynt og nesten fullført. Her ble det bestemt at vi skal ha en recyclerview som fungerer som en dropdown-meny for å velge "den nærmeste sjåføren" i en nødsituasjon. Herfra vil man senere også kunne overføre pakker til den valgte sjåføren.	
Problemer:	Ingen store problemer.	
Til neste gang:	Påbegynne ny sprint. Fullføre nødsituasjons-skjermen, fullføre tutorial	

DAGBOK 27.04.2015

Dato:	27. April 2015	
--------------	----------------	--

	<p>Idag startet vi ny sprint, så vi hadde møte fram til lunsj hvor vi gikk gjennom tasks og bugfixes som skal være med i denne sprinten.</p> <p>Dette er siste sprint vi har muligheten til å utvikle appen videre/rydde opp, så det ble gjort noen valg på hvordan dette skulle håndteres.</p> <p>Vi har også avtalt tid for en akseptansetest neste uke med Harald og Anders.</p> <p>Sondre har sett på hvordan keyboard på emulatoren skal oppføre seg når vi må åpne dette. Har også ryddet opp i kode som nå ikke er relevant mer.</p> <p>Lars-Erik har fikset arbeidslista. Denne lista har vært veldig bugga i flere uker. Etter dette hjalp han Joakim med tutorials.</p> <p>Joakim har laget og implementert flere av tutorial skjermene i appen, men dette krever fortsatt noe arbeid.</p> <p>Martin har brukt store deler av dagen på å jobbe med nødsituasjon-skjermen. Mangler nå kun animasjonen for å simulere overføring av pakker.</p>
Avgjørelser:	Vi lagt til flere tasks og ryddet opp de som var overflødige på JIRA for å forberede siste sprint(med koding).
Problemer:	Ingen store problemer.
Til neste gang:	Forberede akseptansetest til neste uke.

Dato:	11. Mai 2015
	<p>På grunn av jobbintervju for Martin og Lars-Erik, tok vi en ekstra dag med programmering for å rette ut litt flere bugs før vi går over i dokumentasjonsdelen.</p> <p>Sondre jobbet hovedsakelig med delvis optimalisering, for å kunne optimalisere tasklists ekskludert ferdige tasks.</p> <p>Lars-Erik fikset bugs i deliveryactivity og andre småfikser etter akseptansestest.</p> <p>Martin har lagt inn rutestatus for bedre kontroll på fortsettelse og avslutning av rute, og fikset litt småbugs.</p>
Avgjørelser:	Bestemte oss for å legge inn rutestatus, og antageligvis å legge transition fra scanning til tasklist død.
Problemer:	Ingen store problemer konkret, men jobb og jobbintervjuer tok opp mye av arbeidstiden.
Til neste gang:	

DAGBOK 04.05.2015

Dato:	4. Mai 2015
	<p>Forrige torsdag var vi ikke hos Mesan fordi vi valgte å prioritere en innlevering i et annet skolefag.</p> <p>I dag fortsatte vi som normalt, og jobbet stort sett med kun bugs</p>

	<p>og småjusteringer.</p> <p>Martin gjorde ferdig nødsituasjon-vinduet, jobbet med avvik og fikset FAB i scanne-vinduet.</p> <p>Lars-Erik arbeida layout for drag&drop-lista og ruteliste-vinduet, og en del justeringer på tidspakker og transitions.</p> <p>Joakim slettet mest gammel kode og gjorde ellers en del refaktorering.</p> <p>Sondre jobbet med layout-justeringer for forskjellige cards og lagde ikon for signeringsvinduet.</p>	
Avgjørelser:	<p>Vi avgjorde hvordan nødsituasjons-vinduet skulle se ut.</p> <p>Siden vi kan scanne kolli manuelt, kan det hende at man uheldigvis scanner en pakke som f.eks har forsvunnet eller blitt for skadet til å leveres. Derfor må vi kunne unscanne/angre den siste scanningen hvis en slik uventet situasjon skulle skjedd.</p>	
Problemer:	Ingen store problemer.	
Til neste gang:	Forbrede akseptanse-test til onsdag.	

DAGBOK 06.05.2015

Dato:	6. Mai 2015	
	I dag har gruppen gjennomført en akseptansetest med Anders	

	<p>Ullnæss, Harald Alexander Kulø og Lars Andreas Eidsheim.</p> <p>Det ble funnet en hel del småting vi skal fikse på før endelig presentasjon for Postnord, men alt i alt er vi fornøyde. Dagen etter lunsj, har dermed gått med på å fikse bugs og lignende.</p>	
Avgjørelser:		
Problemer:	<p>Det har vært flere store problemer i dag. Blant annet sluttet kamerafunksjonen på den ene aktiviteten å fungere rett før akseptansetesten. Vi fant også ut at transitions lagger i devicen.</p> <p>Dette tyder på at vi har en del optimaliseringsarbeid å gjøre. Vi har funnet noen av feilene allerede, men antar at det blir en del jobbing i helgen denne gangen.</p>	
Til neste gang:	Bugs bugs bugs	

DAGBOK 07.05.2015

Dato:	7. Mai 2015	
	<p>Dagen idag har for det meste gått med til å fikse forskjellige bugs i appen, og gjøre flere endringer for å forberede oss til innlevering.</p> <p>Vi har fikset flere av punktene som kom frem under akseptansetesten i går, og flere andre bugs vi selv har funnet.</p> <p>Blandt annet fikset Joakim og Lars-Erik måten vi tegner ruta på kartet i arbeidslista på. Dette har gjort en stor forskjell på ytelsen</p>	

	<p>til appen, og innlastingen går nå mye forttere.</p> <p>De har også sett over flere andre mindre bugs hver for seg.</p> <p>Martin har jobbet med angre funksjonen til scanninga, og innført onLongPressed lytter til disse skjermene. Han har også sett over flere andre klasser for bugs og cleanup.</p> <p>Sondre har jobbet med optimalisering av delvis fullførte ruter, dette er fortsatt under arbeid. Uten om dette har han sett på mindre bugs rundt om.</p>
Avgjørelser:	
Problemer:	Ingen store problemer i dag
Til neste gang:	Bugs bugs bugs

DAGBOK 11.05.2015

Dato:	11. Mai 2015
	<p>På grunn av jobbintervju for Martin og Lars-Erik, tok vi en ekstra dag med programmering for å rette ut litt flere bugs før vi går over i dokumentasjonsdelen.</p> <p>Sondre jobbet hovedsakelig med delvis optimalisering, for å kunne optimalisere tasklists ekskludert ferdige tasks.</p> <p>Lars-Erik fikset bugs i deliveryactivity og andre småfikser etter akseptansestest.</p> <p>Martin har lagt inn rutestatus for bedre kontroll på fortsettelse og</p>

	avslutning av rute, og fikset litt småbugs.	
Avgjørelser:	Bestemte oss for å legge inn rutestatus, og antageligvis å legge transition fra scanning til tasklist død.	
Problemer:	Ingen store problemer konkret, men jobb og jobbintervjuer tok opp mye av arbeidstiden.	
Til neste gang:		

6.5 KRAVSPEK FRA MESAN

Postnord - Hovedprosjektoppgave



**Proof of concept: Håndterminalen på moderne
plattform / Android**



Innhold

1	Introduksjon til Postnord.....	3
1.1	Mål med prosjektet.....	4
2	Funksjonalitet: håndterminalen mobilapp.....	5
2.1	Kobling til Lokaltransport.....	5
2.2	Plottebord til terminal.....	6
2.3	Oversikt over sjåførinformasjon.....	6
	Backoffice.....	8
2.4	Sjåførdatabase.....	8
2.5	Kjøretøyoversikt.....	9
2.6	Kjøreruteproduksjon.....	9
2.7	Oppgjør.....	9
3	Arkitektur.....	10
4	Ikke-funksjonelle krav.....	12

1 Introduksjon til Postnord

Postnord er en kunde av Mesan som driver med transport og logistikk. Den norske delen av postnord var tidligere Tollpost Globe som ble kjøpt opp av den danske og svenske Posten. Postnord er en aktør med aktiviteter i hele Norge og et selskap av den størrelsen har naturligvis store konkurrenter som Posten/Bring og DB Schenker (+ noen mindre aktører).

Postnord leverer alle typer sendinger, dette er det vi kaller produktportefølgen til Postnord. Distribusjonstjenesten til Postnord omfatter alle aspekter ved frakt av gods, fra sender til mottaker. Det er flere ledd i kjeden som skal kontinuerlig overvåkes for å sikre god kvalitet på leveransene og det kreves et godt system i bakkant for å kunne levere opp til forventningene.

Produktportefølgen til Postnord består i hovedsak av:

- *Pakker*
- *Express*
- *Stykgods*
- *Parti*
- *Termo*

Pakker

Pakker omfatter forsendelser opp til 35 kg og kan leveres på forskjellige måter til privatpersoner og mellom bedrift til bedrift. Dette omfatter

- Hjemlevering(dag- og kveldstid),
- Utlevering i butikk
- Pakkestasjoner (automatisk utlevering)
- Pakker i Norden og Europa

Express

Postnord har et omfattende distribusjonapparat for expressleveranser som muliggjør leveranser raskt. Expressleveranser er leveranser som leveres samme dag eller dagen etter samt in-night nattdistribusjon som sørger for leveranser før kl 0700.

Stykkgods

Stykkgods er forsendelser som er under 2 tonn og består stort sett av palleleveranser både innlands og til Norden.

Parti

Partileveranser er forbeholdt kunder med store leveransekrav som f.eks Komplett og består av skreddersydde løsninger med rutebaserte strekninger. Partifrakt er produktet for forsendelser som kan gå direkte fra avsender til mottaker og er løsningen for forsendelser som er større en 2,5 tonn fraktberegningsvekt.

Termo

Termo er normalt fulle lasteeenheter som semitrailere og containere, men mindre forsendelser kan transporteres etter avtale. PostNord logistics dekker termotransporter i Norge, Norden, Europa og oversjøiske destinasjoner dels i egen regi, dels i samarbeid med faste samarbeidspartnere.

Som et ledd i å kvalitetssikre og ha kontroll over alle leveranser har Postnord i løpet av ca 25 år utviklet et system de samlet kaller for DDS som består av et knippe Informix 4GL applikasjoner, Java Swing grensesnitt samt noen web grensesnitt laget på en eldre web-plattform (Struts). En av fraksjonene til Postnord kalles Lokaltransport og det er lokaltransport som tar for seg distribusjon av forsendelse fra terminal og til sluttkonde. Under lokaltransport benyttes Håndterminalen som hovedverktøy.

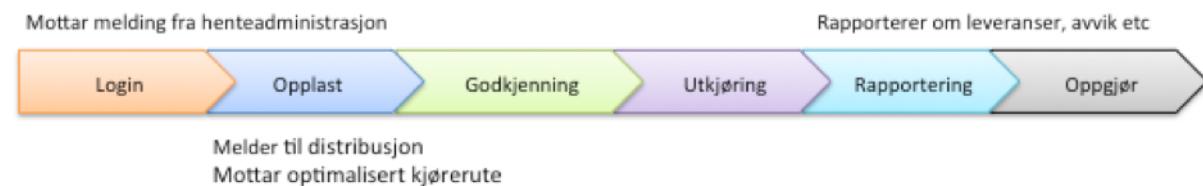
1.1 MÅL MED PROSJEKTET

Postnord har i dag en Håndterminal som er hovedverktøyet til alle sjåførene. Håndterminalen benyttes til alt fra å laste bilen til å lever pakker på døren. Håndterminalen slik den er i dag er en løsning implementert på Windows Mobile 6.5 med Motorola MC65 enheter. Den mobile

plattformen i bakkant er Antenna av Pega. Som kjent er Windows Mobile en enhet som er på vei ut døren, men har fortsatt et godt tak på transportbransjen i Norge pga eksisterende software og gode strekkodescanningsløsninger. Selv om Windows Mobile har sine styrker har den også mange svakheter. Noen åpenbare svakheter er dyr-utvikling, trenge enheter, dårlig brukeropplevelse og mangel på moderne bruksmønstre og teknologi. Vi ønsker å utfordre posisjonen til WM6.5 ved å lage en PoC-løsning som viser potensialet i å bytte plattform, først og fremst for Postnord.

Hovedmålet med løsningen er å kunne korte ned på utvikling av ny funksjonalitet tilpasset bedriftens behov over tid samtidig som det gir brukerne en vesentlig bedre opplevelse av systemet. Det er også et mål at det kommer flere hjelpefunksjoner til brukerne i løsningen som f.eks kart og navigasjon.

2 FUNKSJONALITET: HÅNDTERMINALEN MOBILAPP



Figur 1 Flyt for sjåførens arbeidsprosess

2.1 KOBLING TIL LOKALTRANSPORT

Utlevering av sendinger

Ved utlevering av forsendelser må det registreres i databasen når den ble levert, hvem som mottok sendingen og signatur. Det må også registreres eventuelle avvik ved leveransen slik som: skader, mottaker ikke til stede, eventuelle steder forsendelsen har blitt satt (f.eks bak garasjen etter avtale pr telefon) osv..

Det må også implementeres en varsling om at leveransen skal finne sted 20 minutter før sjåfør ankommer stedet. Hvordan dette implementeres er opp til gruppen, men noen forslag kan

være: en knapp for å sende sms til mottaker eller automatisk sending av sms beregnet fra kjøreruten og hvor langt sjåføren har kommet på ruten.

Sendinger som ikke er levert etter at sjåføren har kjørt ferdig ruten sin skal håndteres i systemet slik at det er enkelt å finne ut hvorfor de ikke har blitt levert. Normen er at disse sendingene taes med den påfølgende dagen, men det er lov å tenke ut alternative løsninger.

Enkelte leveringer er såkalte tidslukeleveranser og disse sendingene skal prioriteres slik at de leveres i det tidsrommet som er registrert i avtalen med kunden. Tidsluker er som regel på 2 timer i løpet av en dag, altså merket med "skal leveres mellom 12 og 14".

- Plukkliste

Sjåførene henter varene på terminalen, de er ikke sortert på annet enn postnummer og ruten de skal kjøre består av en rekke postnummer f.eks 3000-3030 (evt adresser dersom ruteoptimalisering er implementert). Plukklisten skal lages av kjøreleder som produseres i backofficeløsningen og sendes til app, denne plukklisten skal sjåføren ha tilgjengelig på appen og med muligheter for å sortere listen på forskjellige kriterier (med mindre ruteoptimalisering er implementert) slik som: adresser, postnummer, alfabetisk på navn, rotert baklengs osv..

Når sjåføren har scannet alle sendingene inn på bilen skal disse dataene sendes inn til kjøreleder som skal kvalitetssikre og godkjenne listen til sjåføren før sjåføren har lov til å begynne oppdraget.

For å registrere forsendelsene i systemet og for å sikre kvalitet på leveranser skal ALLE sendinger scannes av appen og registreres på kjøreturen. Det er stort sett strekkoder som skal scannes, men det kan også forekomme QR-koder og NFC brikker i fremtiden dersom man finner et fornuftig bruksmønster for denne typen merking.

- Innhenting av sendinger

Som en del av kjøreturen skal sjåførene også hente inn sendinger som enten er forhåndsregistrert eller som kommer inn i løpet av kjøreturen. Hentinger som kommer inn ad-hoc registreres av kjøreleder og sendes til den bilen som er nærmest/har lokasjonen som en naturlig del/avstikker på kjøreruten. Det finnes også faste avtaler på hentinger som som regel har en fast frekvens slik som "hver torsdag ca kl 10:00".

- Kjøreruter

Kjørelederen som sitter inne på kontoret setter sammen alle sendingene og hentingene som skal gjennomføres den aktuelle dagen. PT planlegges dette på morgenen/natten eller dagen i forvegen. Jobben består av å sette sammen sendingene/hentingene på aktuelle ruter og angi hvilke biler og sjåfører som gjennomfører. Disse kjørerutene sendes til de aktuelle apper, det kan være flere kjøreruter pr bil pr dag.

2.2 PLOTTEBORD TIL TERMINAL

Som en del av transporten fra terminal til terminal burde appen kunne brukes for å forutse når bilen ankommer terminal slik at plottebordet og vakta er klare til å motta bilen.

Funksjonalitet:

- GPS tracking av biler

Spore alle biler og "geofence" området som ligger rundt terminalen slik at plottebordapplikasjonen kan generere de data som skal til for å kunne plassere bilen når den ankommer terminalen. Dette består av å angi hvilken port bilen skal kjøre til på terminalen og det avhenger av hva frakten består av. Det er forskjellige typer utstyr på terminalens porter slik som sorteringsanlegg, pallehåndtering osv.

- Av og pålasting (Go To Gate)

Appen skal kunne motta informasjon om hvor/når bilen skal laste av eller losses med gods.

- Go To Container

Motta informasjon om hvilket kontainernummer/hvilken port kontaineren finnes på som skal hentes og lastes og kjøres videre.

2.3 OVERSIKT OVER SJÅFØRINFORMASJON

- Førerkort informasjon
 - Klasser (A,B,C,C1,D,DE...osv..) pr sjåfør
 - Sertifisering

Hva slags gods har sjåføren lov til å håndtere, farlig gods osv..

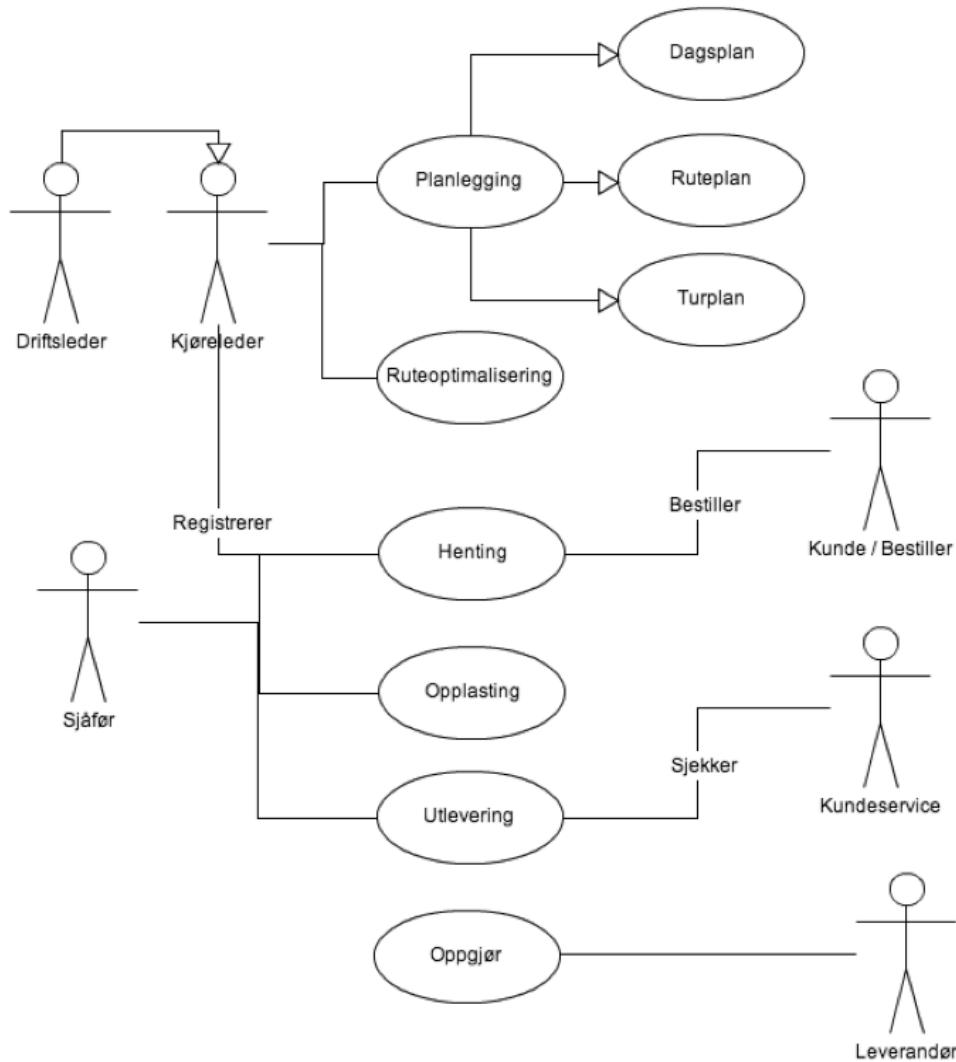
- Varslinger og fornying av sertifikat

Ved utløp av sertifikat og ved behov for ettersyn av sertifikater skal det komme et varsel til sjåføren om hva som er nødvendig å gjøre for å kunne fortsette å jobbe innenfor lovens rammer.

- Personalia
- Bilinformasjon
 - o Lastkategorier – hva kan bilen frakte
 - o Bilpersonalia
 - Merke, modell, regnr, vekt osv.
 - o Feilrapportering
 - § Synliggjøre sensormuligheter
 - § Status på bilen – proaktiv
 - o Kobling mot verksted
- Sjåførinntjening
 - o Historikk
 - Vise hvor mye sjåføren har tjent på de forskjellige oppdragene han har hatt.
 - o Løpende inntjening (basert på avtaler om km/timer/antall leveranser eller lignende)
 - Lage en slags løpende inntjeningsticker som viser til en hver tid hvor mye sjåføren tjener på det inneværende oppdraget. Dette er meningsløst for de med fastlønn, men for noen som har avlønning på oppdrag, km-kjørt osv er det en morsom ting å vise.

BACKOFFICE

Backofficeløsningen må være på plass for å kunne administrere alt fra brukere, apper, sjåfører, turer, sendinger, hentinger osv. Dette skal være en moderne web-applikasjon som gir brukerne en naturlig arbeidsflate for å administrere systemet og den daglige driften.



Figur 2 Usecases for backofficeløsningen

2.4 SJÅFØRDATABASE

Inneholder all informasjon om alle sjåførene i systemet.

- Førerkortinfo
 - o Klasser
 - o Sertifisering

- Varsling og fornyelser
- Personalia
- Bilinformasjon (samme som i appen)
 - Lastkategorier
 - Bilpers

2.5 KJØRETØYOVERSIKT

Inneholder all grunndatainformasjon om alle bilene i systemet samt sanntidsinformasjon på alle biler.

- Realtime oversikt over hvor bilene er
- Status på alle kjøretøy
- Status på kjøreruter / leveranser

Ha en oversikt som til en hver tid viser kapasiteten på kjøretøyene i flåten samt vise restkapasitet på bilene etter hvert som forsendingene leveres til mottaker.

2.6 KJØRERUTEPRODUKSJON

Kunne produsere kjøreruter og turer for oppdrag som skal sendes til appen.

- *Push kjørerute til mobil*
- *Kjøreruteoptimalisering*

Ruteoptimalisering omhandler hvordan man tar et sett med sendinger og hentinger og beregner hvilke kjøretøy som skal gjennomføre disse aktivitetene. Sendinger som har ankommet terminal skal i sin helhet kunne kjøres gjennom en optimaliseringsmotor der resultatet er et sett med ruter fordelt på kjøretøy. Disse rutene skal være så gode som systemet klarer å lage dem og i utgangspunktet klare for utkjøring. Men det er naturlig å tenke at en manuell kvalitetssikring av resultatet gjennomføres før faktisk start på aktivitetene.

Rutene skal fordeles på en slik måte at de er mest mulig kostnadseffektive med tanke på ressursbruk kontra tiden det tar å levere. Kostnader må inn som en parameter i tillegg til kjørelengde, tid og antall stopp pr kjøretøy.

Alle ruter som er generert for dagen skal vises i både tabellform med mulighet for å hente opp detaljer samt kartvisning som viser geolokasjonen til leveranser og hentinger.

I tillegg til en kartløsning skal distribusjonsplanen vises som en tidsliste (gant diagram). Denne visningen vil gi en oversikt over tidsbruken ift leveranser og hentingar for alle kjøretøy som er omtalt i planen.

Noen momenter til tidslistevisningen:

- Før oppstart av dagen vil denne gi driftsleder en opplastingsplan
 - Oppmøtetid for sjåføren
 - Hvor mye som skal lastet
 - Forventet lastetid
- Sanntidsvisning av status for leveringer og hentingar.
- Sette første leveringstidspunkt
- Vise hviletid
- Vise kjøretid
- Ventetid ståsted.
- *Kjørerutetilpasning og administrasjon*

2.7 OPPGJØR

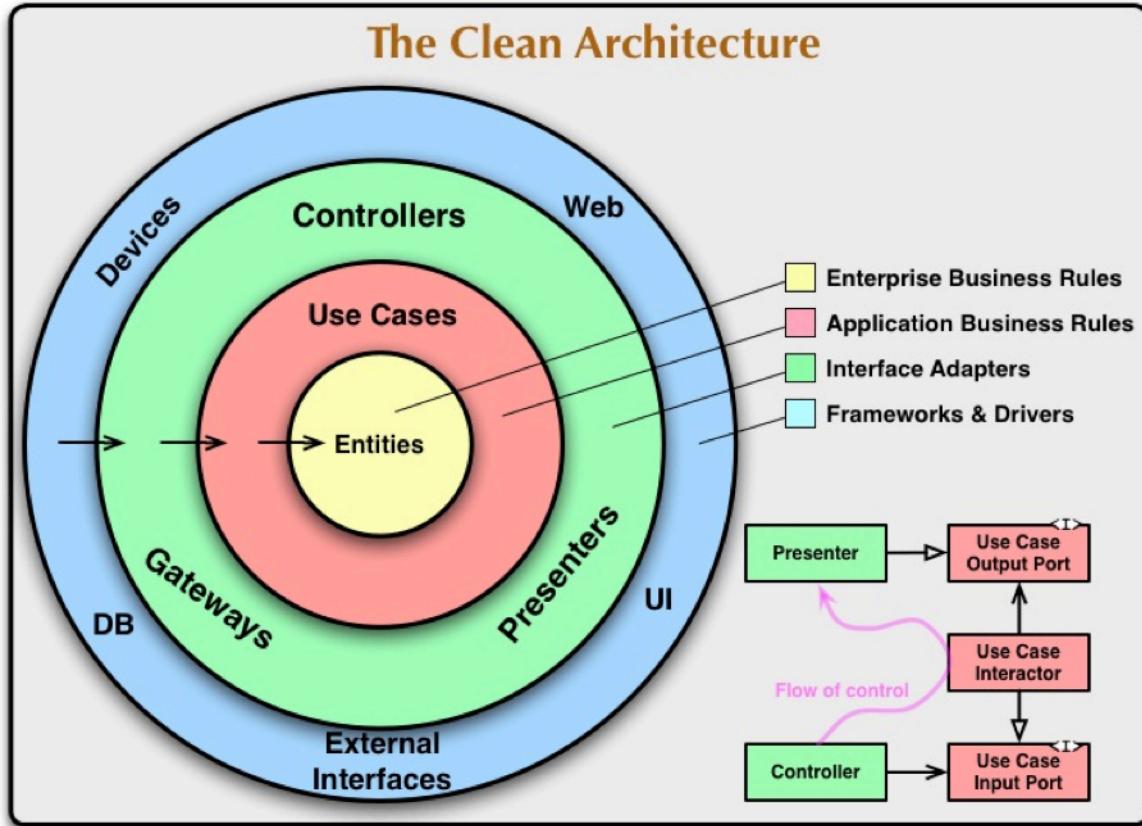
Ved månedsoppgjør og halvmånedsoppgjør skal det kunne genereres informasjon om alle utgifter ifb avregning med sjåførene. Denne rapporten

- Avregning og informasjon om inntjening pr sjåfør
- Administrasjon rund oppgjør.

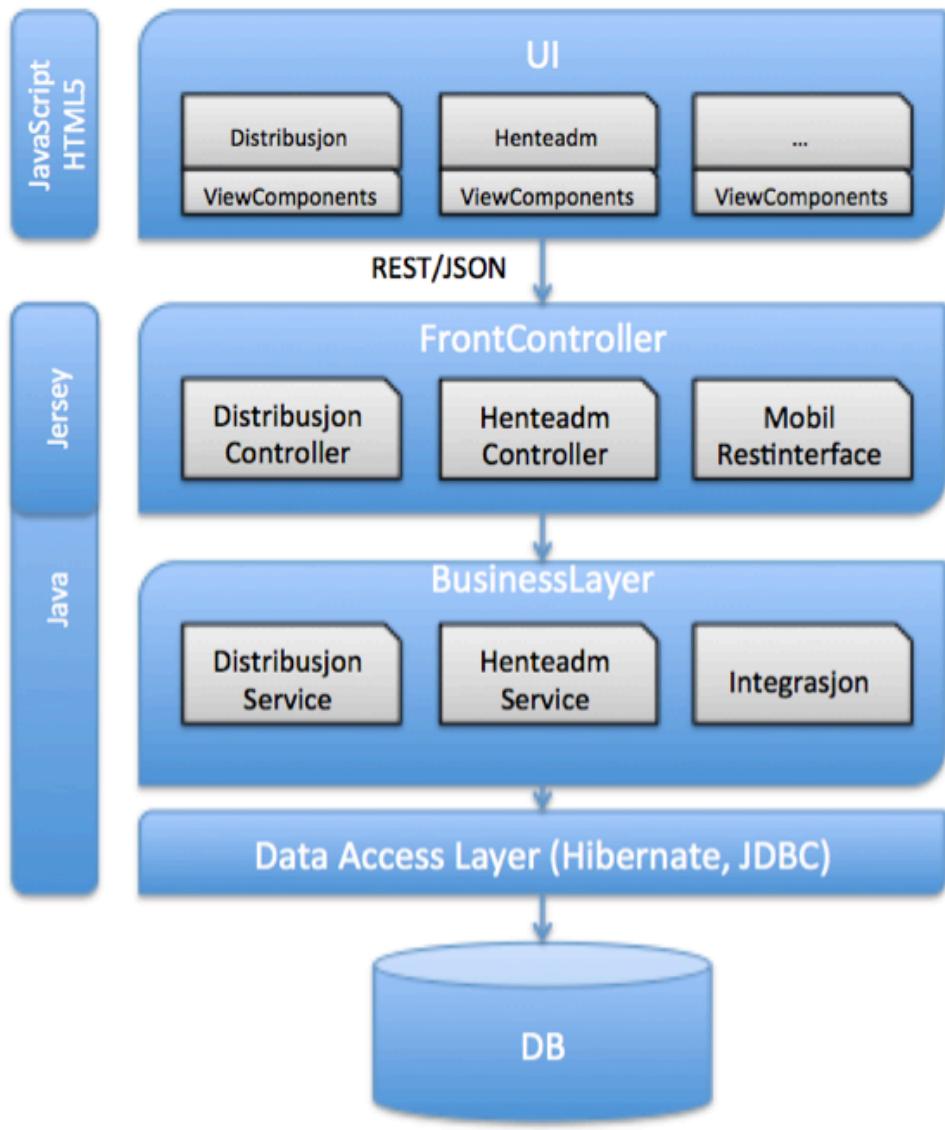
3 ARKITEKTUR

I all hovedsak burde arkitekturen forholde seg til noen enkle standarder slik at man enkelt kan utvikle systemet videre. Dette innbærer blant annet at man bør benytte seg av REST som backendarkitektur og med JSON som overføringsformat.

Beskrivelsen av "The Clean Architecture" av Uncle Bob er et bra utgangspunkt for hvordan man burde tenke kodearkitektur: <http://blog.8thlight.com/uncle-bob/2012/08/13/the-clean-architecture.html>



Konseptuelt kan systemet f.eks se slik ut:



De forskjellige lagene i systemet har helt klare oppgaver for hva som skal løses, og selv om det i dette tilfellet er en-til-en mellom frontcontrollere og forretningslogikk er det ikke sikkert det er riktig i alle tilfeller. Frontcontrollere har som hovedoppgave å servere forretningslogikken til presentasjonslaget (det er f.eks her REST er implementert) og forretningslogikken skal kun innholde regler for dataene og ikke forholde seg til verken lagring eller presentasjon.

Hva man velger til persistens er mer eller mindre fritt frem, men det anbefales å benytte seg av en felles database (enten RDBMS eller Document-store-type).

4 IKKE-FUNKSJONELLE KRAV

<Lages av prosjektgruppen>

Bør inneholde noe om ytelse, sikkerhet, brukervennlighet, oppløsning på devicer/skjermstørrelser osv.

6.6 KOMPLETTE MØTEREFERATER

- Møte PostNord-team 14.01.2015
- Møte med Anders Ullnæss 22.01.2015
- Standup-møte 26.01.2015
- Møte med Harald og Anders 28.01.2015

MØTE POSTNORD-TEAM 14.01.2015

Dato:	14. januar 2015
Tilstede:	Hovedprosjekt: Joakim Rishaug , Sondre Sparby Boge , Lars-Erik Kasin , Martin Hagen . Mesan: Anders Ullnæss , Knut Esten Melandsø Nekså, Anders Grotthing Moe
Forhindret:	
Referent:	Lars-Erik Kasin

Agenda: Oppstartsmøte med PostNord-teamet hos Mesan. Målet med møte er å få en oversikt over oppgaven og få svar på en del spørsmål.

Sak	Ansvarlig
Beslutninger <ol style="list-style-type: none">1. Gjennomgang av programflyt<ol style="list-style-type: none">a. Avhengig av god kontroll ved avvikb. Sjekke feil ved scanning. F.eks ved sjekk av leveringssted for pakken eller lignendec. Scanning bør være automatisk og ikke få opp masse info. Fortsatt ikke sikker på hva de 4 bildene er. Ser ut til å være manuell registrering.	

<ul style="list-style-type: none"> d. Biler er stort sett leid inn. Kan også være vikarer, så vi kan ikke garantere noen kobling mellom sjåfør og bil. e. Vil ikke koble oss direkte til database. Den er alt for tung. Muligens hente ut realistisk data. Altså noe som ser ut sånn som dataen kommer til å se ut. f. Antar at en rute er fra Terminal og tilbakt til Terminal. g. Lag ny henting bør være med for unntakstilstander. h. Utlevering uten kvittering er nattlevering, hvor pakken scannes og geotagges. <p>2. Løsningen skal utformes for én device</p> <p>3. Kvalifisert gjetning på det de(PostNord-teamet) ikke kan svare på</p> <p>4. Ny funksjonalitet</p> <ul style="list-style-type: none"> a. Knapp for å ringe/sms kjøreleder b. GPS c. Ha med liste om hva som skal plukkes opp er ny funksjonalitet. d. Muligens automatisk godkjenning. e. Sortering etter rute allerede ved opplastning! For å få ut de første pakkene først. <p>5. Annen kunnskap vi kan dra nytte av</p> <ul style="list-style-type: none"> a. Visse ting går gjennom scanning-system, andre ting scannes for hånd. Det som scannes for hånd blir ofte bare scannet 5/x, og så antar man at alle (fra liste) er tilstede. b. Scanner ofte f.eks 5 første og/eller 5 siste, ikke hele paller c. Hørt under et møte: Rutenummer brukes som id for sjåfør? Kjører samme rute? d. Innebygd printer i bilen til å skrive ut barcodes osv. Bluetooth antagelig. 	
<p>Oppfølgingspunkter</p> <ol style="list-style-type: none"> 1. Videre spørsmål <ul style="list-style-type: none"> a. Kjører hele døgnet, Lokaltransport skjer mest på dagtid, men trenger vi da hele døgnet? 2. Finne løsninger på 	<p>1. Gruppa</p> <p>2. Gruppa (innspill fra</p>

<ul style="list-style-type: none"> a. Trafikk-RSS? Få frem at dette er mulig. b. GPS - google API er visst lett å bruke. Bør ses på. <ul style="list-style-type: none"> i. Veldig interessert i ruteoptimalisering. Fokus på den. ii. Prøv å få opp alternative ruter som sjåfør kan velge mellom iii. Gjerne dynamisk oppdatering ved tillegg av pakker iv. Målet er tredjeparts autogenerering – Google Directions API c. Mindre pakker og lignende har kommet først senere, derav litt kunstig skille. Finne en logisk løsning på dette skille. <p>3. Fokus videre</p> <ul style="list-style-type: none"> a. Ruteoptimalisering b. Sortering av pakker til bil c. <i>Optimaliserer brukervennlighet - intuitivisere</i> 	Mesan)
Fokus videre <ul style="list-style-type: none"> 1. Ruteoptimalisering 2. Sortering av pakker til bil 3. <i>Optimaliserer brukervennlighet - intuitivisere</i> 	

MØTE MED ANDERS ULLNÆSS 22.01.2015

Dato:	22. januar 2015
Tilstede:	Hovedprosjekt: Joakim Rishaug , Sondre Sparby Boge , Lars-Erik Kasin . Mesan: Anders Ullnæss
Forhindret:	Martin Hagen

Referent:

Lars-Erik Kasin

Agenda: Gjennomgang av arbeidsplan(estimering) og teknologivalg, samt veiledning videre.

Sak

Gjennomgang av arbeidsplan

- Enige om at estimeringen ser ok ut.
- Punktene bør skrives litt mer detaljert.
- Bør ta utgangspunkt i skjermbilder for å segmentere fremdriften (Eks: Gjøre oss ferdig med arbeidsliste først)

Teknologivalg

- Enige om at ca 8 tommer er ideelt.
- Samsung Galaxy Note 8.0 ser mest lovende ut, men den kommer kun med Android 4.1 med mulighet for 4.4.
- På grunn av nye overganger og stifulksjoner er Android 5.0 helt klart å foretrekke for å lage en tiltalende POC.
- Også mulig å bruke Android 4.4 med support libraries for å få ca samme resultat, men 5.0 er enklest og gir best resultat.
- Kun Google Nexus 9 har 5.0 installert.
- Ellers ble vi enige om å ta i bruk
 - Good Maps API
 - Google Directions API
 - Bibliotek for scanning av strekkoder og QR fra Anders' prosjekt
 - Finne et bibliotek for å motta signatur
 - Android Annotations
 - Retrofit
 - Genymotion (emulator)

Utviklingsmiljø

I samarbeid med Anders startet vi med å sette opp utviklingsmiljøet. Vi bruker Android Studio med Genymotion-emulator, og Github for versjonskontroll.

Fikk koblet til Github for alle som var tilstede fra prosjektgruppa, opprettet prosjektet og lagde de første activityene.

Både Android Studio, Gradle og Android SDK (5.0) var oppdatert siden sist vi brukte det, så dette tok noe tid å tilpasse, men etterhvert fikk alle koblet seg opp mot prosjektet og syncet med Github. Skal nå være klart for å begynne videre utvikling.

STANDUP-MØTE 26.01.2015

Dato:	26. januar 2015
Tilstede:	Hovedprosjekt: Joakim Rishaug , Sondre Sparby Boge , Lars-Erik Kasin , Martin Hagen .
Forhindret:	
Referent:	Joakim Rishaug

Etter den første dagen med små-programmering ble det fort klart at vi måtte bestemme oss for hvilken struktur appen skal ha fremover. Navigation Drawers fungerer bedre med fragments enn activities, i tillegg til at flyten er bedre i programmet. Det er også en klar trend i markedet mot fragment-baserte applikasjoner.

Agenda: Møte angående valg av app-struktur. Fragments vs Activities

Sak	Ansvarlig

Beslutninger	
<ol style="list-style-type: none"> 1. Benytte navigation drawer for meny til valg som er utenfor hovedflyt i programmet <ol style="list-style-type: none"> a. Dynamisk endre navigation drawer avhengig av hvilket fragment som befinner seg på topp i manager. 2. Ha to hovedactivities <ol style="list-style-type: none"> a. Login-activity b. "FragmentHolderActivity" 	
Fokus videre	Hovedsaklig Martin og Lars-Erik

MØTE MED HARALD OG ANDERS 28.01.2015

Dato:	28. januar 2015
Tilstede:	Hovedprosjekt: Joakim Rishaug , Sondre Sparby Boge , Lars-Erik Kasin , Martin Hagen . Mesan: Anders Ullnæss , Harald Alexander Kulø
Forhindret:	
Referent:.	Lars-Erik Kasin

Agenda: Gå gjennom status for prosjektet, og planen fremover, med Harald.

Sak	Ansvarlig
<p>Saker</p> <ul style="list-style-type: none"> ● Planen på mandag: <ul style="list-style-type: none"> ● Bør legge en klarer plan i Jira og ellers, så vi vet hva vi starter med når vi kommer. ● Fremdriftsplan/Storypoints: <ul style="list-style-type: none"> ● Har vi sett på helheten? - Bør kanskje gå over og vurdere storypoints på nytt i sammenheng når vi har flere referansepunkt. ● Backend: <ul style="list-style-type: none"> ● Harald er enig i at Mocking av databasen er helt greit. ● Kart: <ul style="list-style-type: none"> ● Hva viser kartet? - Mulig vi bør gå for en litt mer overordnet rutevisning på kart. Ikke full GPS, da det kan bli mye jobb. Vil uansett vise ruta. ● Arbeidsliste: <ul style="list-style-type: none"> ● Mulig lista kommer til å bli veldig komplisert. <ul style="list-style-type: none"> ● Bør vi her beregne flere storypoints? ● Mulig vi bør legge til "edit"-knapp før drag&drop, så lista ikke trenger å ha alle touch-funksjoner til alle tider. ● Harald sier vi kan gå ut ifra at alle pakker scannes! ● Risikomatrise: <ul style="list-style-type: none"> ● Bør holdes et øye med. Kan oppdateres underveis. Husk at denne er dynamisk. ● Testing: <ul style="list-style-type: none"> ● Interaksjonstesting bør vi se på, men ikke vits å legge masse tid i det, hvis det er komplisert. ● Test av kode og logikk må vi ha med. Anders har noen tips. 	
Device	

Enige om at Samsung Galaxy Tab S er et godt alternativ.

Først og fremst pga at den får Android 5.0 tidlig i prosjektperioden, men vi er også enige om at skjermstørrelsen

er fin, og Harald sier den tåler mye, noe som er en fordel med sjåfører som ikke bryr seg som mye om devicen.

Avgjorde ikke helt når den skulle skaffes, men var enige om at den er en fordel å ha så tidlig som mulig, men

haster ikke veldig, ettersom genymotion fungerer møter behovet foreløpig.

Jira / Scrum

Vi må få inn mer detaljerte stories og issues i Jira, for å holde oversikten over arbeidet.

Ble rådet å ta en kikk på PostNord-teamets Jira, som skal være veldig bra.

Har også avtalt å møte Knut og muligens fler fra PostNord-teamet på mandag, for å gjennomgå Jira.

Bør for øvrig kanskje prøve å fordele roller mer klart. Særlig bør vi ha en Scrum-master som har ansvaret for

alt av Scrum. Vi tenker å rottere på dette, for at alle skal få prøve å kjenne på det å lede arbeidet, men det er

viktig at vi har en konkret og klar plan på dette.

6.7 USER STORIES

SOM SJÅFØR VIL JEG

- logge inn for å enkelt ha tilgang til min informasjon og egne instillinger
- enkelt ha oversikt over mine ruter, for å lett kunne velge min rute
- få beskjed om hvilken gate på terminalen jeg skal hente pakkene mine fra for å lett finne min last
- ha et kart over ruta fordi jeg vil se hvor jeg skal kjøre
- endre rekkefølgen på leveringene, for å kunne optimalisere ruten etter egen kunnskap
- se hvilken pakke som er den neste som skal leveres, for å lett holde styr på ruta
- kunne hente ad-hoc pakker for å være fleksibel i arbeidet
- kunne godkjenne eller avslå ad-hoc-pakker, for å ha riktig arbeidsmengde
- ha en hjelpefunksjon, for å vite hvordan jeg skal bruke systemet
- kunne kontakte kjøreleder, for å løse eventuelle problemer underveis
- sende sms eller ringe til kunden, for at kunden skal vite når jeg kommer
- kunne legge til nye henting, for å ta med pakker fra leveringspunkter
- kunne legge til nye leveranser, for å kunne håndtere etterlastning
- kunne se utvidet informasjon om hver enkelt pakke slik at uklarheter kan oppklares
- kunne melde fra om avvik ved levering for å dokumentere feilsituasjoner
- kunne se utvidet informasjon om hver mottaker for å oppklare uklarheter
- få lydeffekter for å vite når noe skjer på ruta uten å måtte se på skjermen
- kunne søke blant kunder for å manuelt legge disse inn under oppretting av uregistrert henting
- kunne rotere skjermen, fordi det noen ganger gir bedre brukervennlighet
- lett kunne tilkalle hjelp ved nødsituasjon, fordi rask hjelp minsker konsekvensene
- kunne se statistikk om min arbeidsdag/måned, fordi dette kan være nyttig
- kunne motta signatur fra kunde ved levering, for å dokumentere leveransen
- kunne levere uten signatur fordi kunden kan ha gjort slike avtaler
 - Dokumentere leveranse på annen måte (Gps/bilde)

Scanning

- scanne inn pakker for å verifisere at jeg har med korrekt last
- scanne inn pakker ved utlevering for å dokumentere utlevering

Manuelle endringer

- kunne fjerne kolli fra arbeidsliste, for å manuelt justere om endringer trengs
- kunne manuelt legge inn pakker i systemet, for å håndtere feil ved scanning

- kunne manuelt legge til mottaker i systemet, for å fylle inn mangelfull informasjon

Statusendringer av pakker

- registrere at pakken er levert så systemet er oppdatert
- registrere at pakken er lastet på bilen, slik systemet er oppdatert

SOM KJØRESLEDER VIL JEG

- kunne sende hentemelding til nærmeste sjåfør, for å få hentet nye pakker med en gang
- kunne ringe sjåfører for å kunne stille spørsmål under ruten
- kunne sende en optimalisert rute til sjåfører så pakkene kan bli levert effektivt
 - inkludert prioritering på leveranser med satt tidsrom
- se hvor sjåfører er i deres rute for å ha oversikt over hver enkelt sjåfør
- få beskjed raskt dersom feil oppstår under opplastning for å kunne håndtere situasjoner umiddelbart
- kunne se info om sjåfører fordi jeg vil vite hvilke pakker sjåførene kan håndtere
- kunne se info om biler fordi jeg vil vite type last bilen kan leve
- kunne finne informasjon om en pakkes status for å ha innsikt i flyten

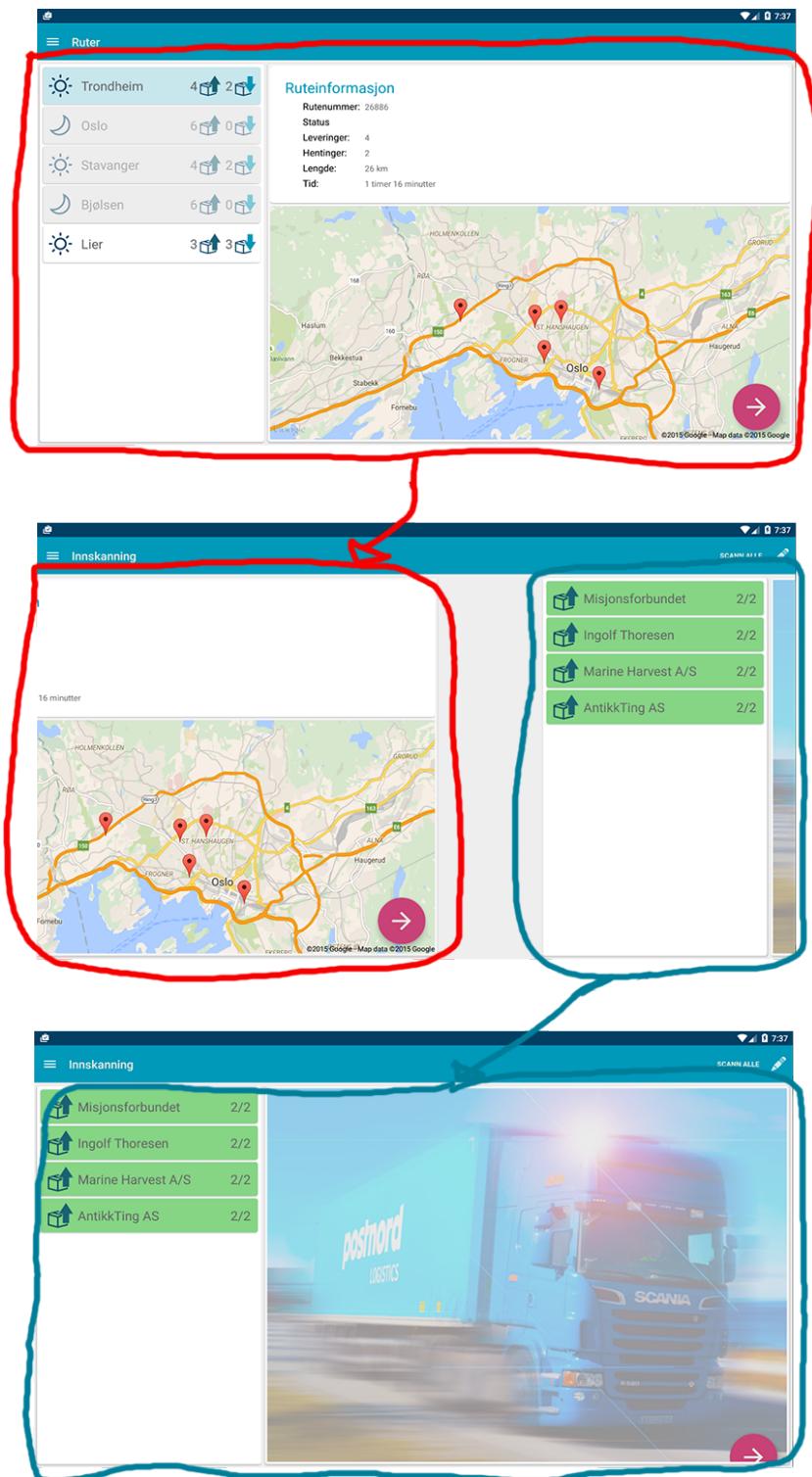
SOM KUNDE VIL JEG

- Ha god oversikt over hva jeg signerer på, slik at jeg vet jeg får korrekt pakke(r).
- kunne levere pakke til sjåføren og få den inn i systemet, for å få den sen

SOM UTVIKLER VIL JEG

- Ha oversikt over valgene som er tatt i PoC-appen, for å kunne få flere perspektiv på problemstillingene.
- Spør Postnord-gjengen angående hva **de** gjerne vil ha videre
- Ha en app som går an å skalere opp fordi jeg vil kunne utvide de

6.8 TRANSITION FORKLARING



6.9 MOCKUPTEGNINGER

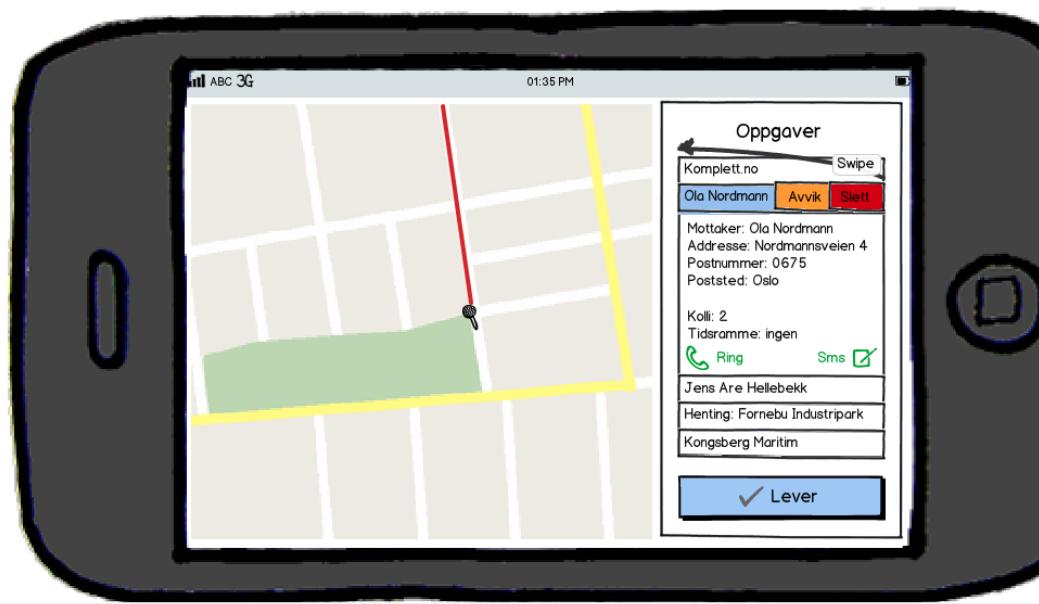
Utlevering

Horizontal og vertikal view for denne

Ved levering, må man scanne alle pakker. Etter alle kolli i leveranse er scannet, slider vi kamera vekk, og signatur-felt vil vises



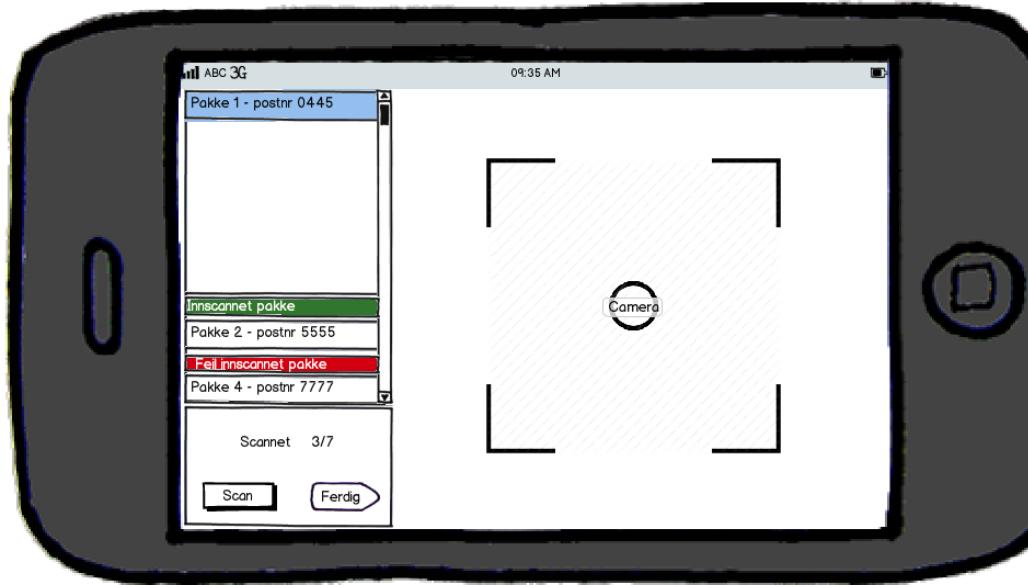
Kart/Arbeidsliste

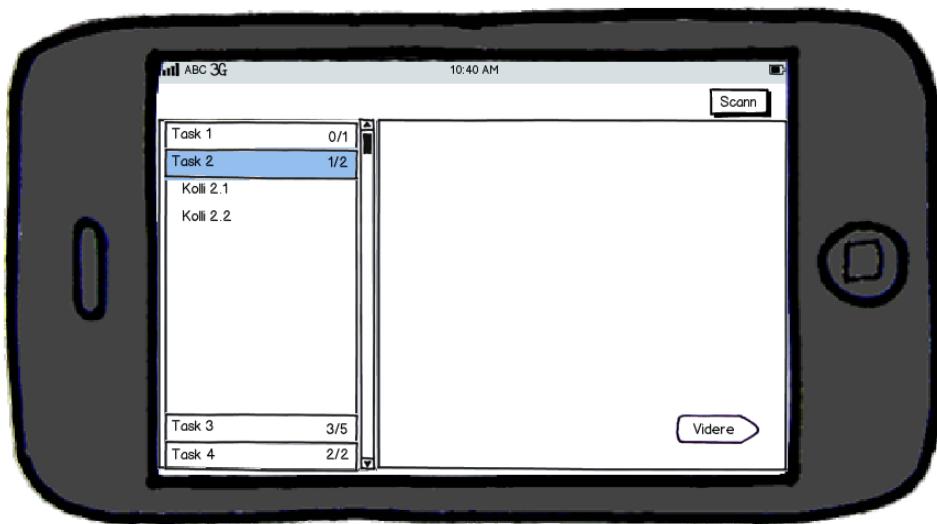


Arbeidsliste+Pakkecard

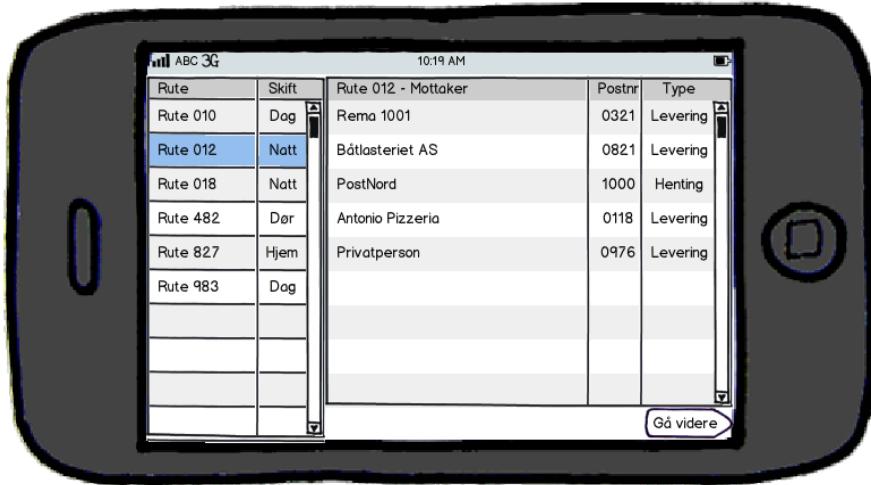


Pakke-scanning



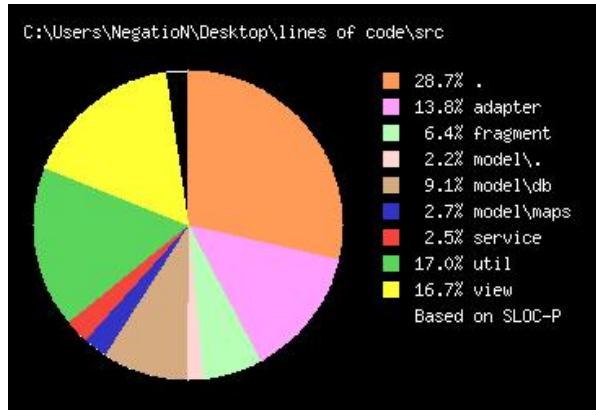


RuteListe



6.10 KODEANALYSE

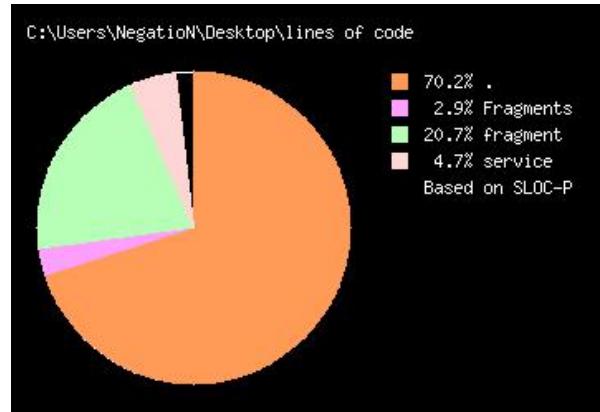
Selvskrevet kildekode for prosjektet. Tar kun hensyn til .java-filer, så alt av grafisk XML vil komme i tillegg.



Symbol	Count	Definition
Source Files	66	Source Files
Directories	12	Directories
LOC	11969	Lines of Code
BLOC	2169	Blank Lines of Code
SLOC-P	7510	Physical Executable Lines of Code
SLOC-L	5718	Logical Executable Lines of Code
MVG	807	McCabe VG Complexity
C&SLOC	75	Code and Comment Lines of Code
CLOC	2290	Comment Only Lines of Code
CWORD	12055	Commentary Words

HCLOC	15	Header Comment Lines of Code
HCWORD	97	Header Commentary Words

I tillegg kommer kode generert av
AndroidAnnotations som kjører under vår
selvskrevne kode.



Symbol	Count	Definition
Source Files	28	Source Files
Directories	7	Directories
LOC	3315	Lines of Code
BLOC	650	Blank Lines of Code
SLOC-P	2581	Physical Executable Lines of Code
SLOC-L	1738	Logical Executable Lines of Code
MVG	237	McCabe VG Complexity
C&SLOC	0	Code and Comment Lines of Code

CLOC	84	Comment Only Lines of Code
CWORD	336	Commentary Words
HCLOC	84	Header Comment Lines of Code
HCWORD	336	Header Commentary Words

6.11 TEKNISK ORDLISTE

Ord/uttrykk	Forklaring
Android	
Activity	Representerer presentasjonslaget for en Android applikasjon. Oppretter skjermbildet og håndterer interaksjoner med device.
Fragment	En del av et user interface eller oppførsel som kan plasseres inne i en Activity. En Activity kan inneholde flere fragments
Transition	En animasjon Android 5.0 skaper i overgangen mellom to skjermer ut ifra en start- og sluttstate.
Floating action button (FAB)	En type knapp som kom med Material Design mønsteret. Blir brukt for å indikere hovedflyt
Drawer	Meny som kommer frem fra venstre side av skjermen. Brukes ofte som navigasjon til skjermer utefor hovedflyt.
Lollipop	Navnet på versjon 5.0 av Android.
Card	Cards er en del av Material Design, hvor informasjon presenteres på en papirlignende overflate. Har ofte en høydeforskjell fra andre elementer.
View	Et enkelt element for fremvisning av informasjon på Android-device.

Longclick	En gest hvor brukeren holder lenge inne på touch-devicen.
AndroidAnnotations	Et rammeverk med åpen kildekode for å gjøre Android utvikling raskere.
Adapter	Et Adapter-objekt knytter visningen av for eksempel ListView eller Spinner til de dataene som skal vises. Adapteren gir tilgang til dataelementene.
Teknologi	
Java	Objektivt programmeringspråk som brukes under Android utvikling.
XML	Et markup språk, mye som HTML, som Android bruker for å lage sine layouter.
SQL	Et standard språk for å bruke og manipulere databaser. I vår app brukes et program som bygger på Android sin SQLite.
API	HTTP-tilkoblingspunkt for aksess av database og skriving til database. Vanligvis overført via JSON eller XML
Git	Versjonskontroll for kode.
Annet	
Håndterminal	PDA som brukes for utlevering av pakker hos PostNord.

PoC	Proof of concept
User Experience	Brukeropplevelse
Best Practices	Gode retningslinjer for hvordan man skal skrive kode og best mulig kunne vedlikeholde den.
Phablet	En størrelse av devicer mellom mobil og tablet.
CRUD	Create, Read, Update og Delete. Hovedfunksjonaliteten for en database.
Boilerplate-code	Kode som må være tilstede for å starte den ekte utviklingen. Anses som "bortkastet tid" og gjenbrukes ofte om man har muligheten.

6.12 KILDER

Google, Android 5.0 (2014), hentet 05.2015 fra google.com

<http://www.android.com/versions/lollipop-5-0/>

Google, Material Design (2014), hentet 05.2015 fra google.com

<http://www.google.com/design/spec/material-design/introduction.html>

Heidi Olson, Color theory (2014), hentet 01.2015 fra uaf.edu

<https://cios233.community.uaf.edu/design-theory-lectures/color-theory/>

Google, Android Studio (2013), hentet 05.2015 fra android.com

<http://developer.android.com/tools/studio/index.html>

Pnikosis, Progress Wheel (2015), hentet 04.2015 fra github.com

<https://github.com/pnikosis/materialish-progress>

Genymobile, Genymotion (2013), hentet 01.2015 fra genymotion.com

<https://www.genymotion.com>

Satyan Arayan, Sugar ORM (2012), hentet 02.2015 fra github.com

<http://satyan.github.io/sugar/>

Gcacace, Android Signaturepad (2014), hentet 03.2015 fra github.com

<https://github.com/gcacace/android-signaturepad>

Gradle Inc, Gradle (2008), hentet 01.2015 fra gradle.org

<https://gradle.org/>

Excilys, AndroidAnnotations (2012), hentet 01.2015 fra github.com

<https://github.com/excilys/androidannotations/>

Dushyanth Maguluru, Zbar (2013), hentet 02.2015 fra github.com

<https://github.com/dm77/ZBarScanner>

Google, Google Directions (2006), hentet 01.2015 fra google.com

<https://developers.google.com/maps/documentation/directions/>

Google, Google Maps (2005), hentet 01.2015 fra google.com

<https://developers.google.com/maps/documentation/android/>