

2013

Kjørehjelperen

Testdokumentasjon

Høgskolen i Oslo og Akershus



Forord

Dette dokumentet tar for seg to forskjellige ting. Først forklares det hvordan programmet håndterer feil og melder dette til brukeren. Deretter dokumenteres de fysiske testene av programmet, og resultatet av disse.

Innholdsfortegnelse

Forord.....	1
Innholdsfortegnelse	2
1 Feilhåndtering i programmet.....	3
2 Praktisk testing av Kjørehjelpen	5
2.1 Testtur 1.....	5
2.2 Testtur 2.....	6
2.3 Testtur 3.....	6

1 Feilhåndtering i programmet

Det er mange feil som kan oppstå i et program, og det er viktig å gi informativ tilbakemelding til brukeren. Samtidig har vi forsøkt å lage en applikasjon som krever minst mulig interaksjon under bruk med tanke på trafikksikkerhet. Vi har derfor valgt å samle feilene under to ulike feilmeldinger.

Hvis telefonen ikke klarer å finne posisjonen den befinner seg på, opprettes det en feilmelding. Dette er en dialogboks som sier: "Klarte ikke finne enhetens posisjon. Har Kjørehjelpen tillatelse til å bruke posisjonstjenester?". Brukeren må trykke "OK" for at meldingen skal forsvinne. Denne feilen kan oppstå hvis brukeren ikke har gitt Kjørehjelpen tillatelse til å bruke telefonens posisjonstjeneste. Den kan også oppstå hvis posisjonstjenesten er deaktivert på telefonen. Hvis brukeren befinner seg et sted GPS-tjenesten¹ ikke får kontakt med satellittene vi også denne feilmeldingen dukke opp, som f.eks. inne i en lang tunnel.

Selv om denne feilmeldingen krever interaksjon fra brukeren er det antatt at feilen vil oppstå sjeldent under bruk. Feilmeldingen vil dessuten ikke fortsette å komme opp etter den først er vist én gang.

Hvis det skjer en annen feil, for eksempel at NVDB² APIet³ ikke svarer, at RestKit⁴ ikke klarer å mappe⁵ resultatet, at det er en feil med Core Data⁶ eller at det rett og slett ikke finnes noe data for posisjonen, vises "Fant ingen data om din lokasjon. Prøver igjen straks!". Denne feilmeldingen krever ingen interaksjon, og forsvinner så snart applikasjonen klarer å hente nye data. Vi har kommet frem



Figur 1: Denne feilmeldingen dukker opp hvis telefonen ikke klarer å finne posisjonen den befinner seg på.

¹ NAVSTAR Global Positioning System (GPS) er et nettverk av satellitter som er plassert i bane rundt Jorden av det amerikanske forsvaret. Systemet gjør det mulig for en mottaker å fastsette egen posisjon med svært stor nøyaktighet overalt i verden, under nær sagt alle værforhold.

² Nasjonal vegdatabank, database med vegobjekter forvaltet av Statens Vegvesen.

³ Application Programming Interface (API) er et grensesnitt for kommunikasjon mellom programvare. APIet beskriver de metoder som en gitt programvare eller et bibliotek kan kommunisere med.

⁴ RestKit er et Objective-C-rammeverk for iOS som forenkler kommunikasjonen med REST-baserte webtjenester og mappingen av objekter.

⁵ Mapping er prosessen med å overføre data fra en datastruktur til en annen, f.eks. fra JSON til et NSObject (objekt i Objective-C).

⁶ Core Data er et rammeverk for databaselagring på iOS.



Figur 2: Denne feilmeldingen vises hvis det skjer en feil med NVDB APIet, RestKit eller Core Data.

til at brukeren ikke behøver å informeres om hva som gikk galt. Dette er både med tanke på trafiksikkerhet, og at det uansett ikke er noe brukeren kan gjøre med saken.

Detaljerte feilmeldinger logges allikevel til konsollen⁷ slik at feil kan oppdages og løses under testing.

⁷ Programvindu som viser tekst, ofte status- eller feilmeldinger.

2 Praktisk testing av Kjørehjelperen

Det ble foretatt tre testturer av applikasjonen. Det optimale hadde selvfølgelig vært å organisere brukertester med ulike tredjeparter. Dette lot seg dessverre ikke gjøre av ulike årsaker. Det er ikke bare å lansere en beta-versjon av en applikasjon til iOS⁸. Telefonen må kobles til utviklermaskinen, og det må installeres utviklerprogramvare på den. I tillegg må det opprettes en "provisioning"-profil som knytter telefonen til utviklerens Developer-konto på apple.com.

Siden ingen av gruppemedlemmene eier en iPhone måtte dette lånes til testen. Organiseringen rundt dette begrenset derfor antallet testturer til tre. Det er planer om å foreta flere tester før applikasjonen lanseres i App Store⁹.

Alle testene ble foretatt på en iPhone¹⁰ 4S.

2.1 Testtur 1

Den første testen ble foretatt etter sprint 1¹¹, søndag 24. februar. Den eneste funksjonaliteten som var implementert var visning av fartsgrense.

Testen viste at GPS'en på telefonen oppdaterte seg for sjelden. Vi kom frem til at en løsning hvor oppdateringsintervallet justeres kontinuerlig basert på farten ville være en bedre løsning.

Det viste seg at fartsgrensene som ble vist på skjermen var tilfeldige, og ofte viste fartsgrensen på kryssende og parallelle veger. På dette tidspunktet ble fartsgrensene valgt kun på koordinater, og ikke sammenliknet med veglenken¹² man befant seg på. Vi kom frem til at vi måtte sammenlikne fartsgrensens veglenke med den vi befant oss på for å unngå dette problemet.



Figur 3: Dette bildet er tatt under testtur 1.

Telefonens skjerm skrudde seg av etter noen minutter uten direkte interaksjon. Slik kunne det selvfølgelig ikke være, så dette måtte overstyres i programmet.

⁸ iOS er Apples operativsystem for iPhone, iPad og iPod.

⁹ App Store er Apples digitale butikk for applikasjoner og spill.

¹⁰ Smarttelefon fra Apple Inc.

¹¹ En sprint er et begrep i Scrum og omfatter en kort periode i utviklingen med bestemte mål om funksjonalitet som skal implementeres.

¹² Et objekt i NVDB som representerer en sammenhengende vegstrekning med en unik ID.

Når fartsgrensen ble tresifret, altså 100, fikk ikke tallet plass på fartsgrense-skiltet. Dette var en pinlig feil som lett lot seg løse.

2.2 Testtur 2

Den andre testturen ble utført etter sprint 4, søndag 14. april. På dette tidspunktet var fartsgrense, forkjørsveg, motorveg, vilttrekk, høydebegrensning, fartsdemper og jernbanekryssing implementert. Lydvarsling, Core Data, HUD¹³, avstand og landskapsmodus var også implementert.

Vi oppdaget at flere objekttyper manglet definert type i NVDB. F.eks. har vilttrekk typene "elg", "hjort" og "rein", mens fartsdempere har typer som "rumlefelt", "busshump", "fartsdump" osv. Når typen ikke var oppgitt ignorerte applikasjonen objektet. Vi kom frem til at applikasjonen heller burde bruke en standardtype hvis dette manglet. F.eks. skulle et vilttrekk uten type defineres som "elg" da dette er vanligst.



Figur 4: Under testtur 2 var HUD-visning implementert.

En annen ting vi la merke til var at applikasjonen fjernet avstandsteksten når den mottok nye skilt. Siden det er en forsinkelse mellom skiltene og tilhørende tekst opplevdes det at avstandsteksten blinket av og på. Det måtte tenkes ut en bedre løsning på dette.

Vi oppdaget også en mindre feil, nemlig at applikasjonen alltid viste motortrafikkveg-skiltet når man var på en motorveg, selv om man var på en vanlig motorveg¹⁴. Dette måtte undersøkes og løses.

2.3 Testtur 3

Den tredje og foreløpig siste testturen ble gjennomført etter sprint 5, onsdag 24. april. På dette tidspunktet var all funksjonalitet som er nevnt i dokumentasjonen implementert.

På denne testturen så vi at problemet med at avstandsteksten blinket av og på enda ikke var løst. Det var blitt bedre, men enda ikke helt optimalt.

Vi la også merke til at avstanden som oppgis til objekter ofte er litt unøyaktig. Vi tror dette har noe med at avstanden vi får fra MapQuest¹⁵ er navigasjonsavstand, og ikke direkte avstand. Den kan

¹³ Head-up display eller heads-up display, også kjent som HUD, er en gjennomsiktig skjerm som viser informasjon uten at brukerne må se bort fra sine vanlige synspunkter. Opprinnelsen til navnet stammer fra at en pilot skal kunne se informasjon med hodet "opp" og se frem, i stedet for å se ned på instrumentene i cockpiten.

¹⁴ Motorveg og motortrafikkveg er de nye navnene på henholdsvis motorveg klasse A og motorveg klasse B.

¹⁵ MapQuest er en online karttjeneste som tilbyr blant annet vegkart og navigasjonshjelp.



Figur 5: Under testtur 3 var all funksjonalitet implementert.

Tydeligvis er det fremdeles noen problemer knyttet til dette. Løsningen blir å simulere kjøring på en veglenke på datamaskinen, og så analysere hva som skjer hele vegen. Hvis feilen ligger i applikasjonens algoritmer lar dette seg enkelt løse, men hvis det viser seg at feilen ligger i NVDB kan dette by på større utfordringer.

dermed finne på å inkludere at man må snu kjøretøyet i avstanden. Det må undersøkes om dette lar seg forbedre.

Det største problemet vi oppdaget var at applikasjonen ikke alltid klarer å forstå hvilke objekter som befinner seg foran og bak kjøretøyet på veglenken. Det er lagt mye arbeid i å finne ut nettopp dette, i tillegg til hvilken veg kjøretøyet beveger seg.