

2013

# Kjørehjelperen

## Kravspesifikasjon

Høgskolen i Oslo og Akershus



## **Forord**

Kravspesifikasjonen skal fungere som et styringsdokument for hovedprosjektet. Den skal definere rammer og retningslinjer for prosjektet, utarbeides i samarbeid med oppdragsgiver, og sikre at både vi og oppdragsgiver får klarlagt hva betingelsene for prosjektet skal være. Dette gjelder både funksjonelle og ikke-funksjonelle krav, samt krav til arbeidsprosess og kvalitetssikring.

Kravspesifikasjonen skal opptre som en bindende avtale mellom oss og oppdragsgiver, men det skal også tas hensyn til krav fra Høgskolen i Oslo og Akershus. Dette dokumentet, og alle revisjoner, skal derfor være godkjent av alle parter før det kan anses som gyldig.

## **Leserveiledning**

Kravspesifikasjonen kommer først med en kort presentasjon av de ulike partene, bakgrunnen for prosjektet og en generell beskrivelse av oppgaven. Deretter følger funksjonelle krav, etterfulgt av ikke-funksjonelle krav. Til slutt tar vi for oss håndteringen av endringer og feil som måtte oppstå underveis i prosjektet.

Tekniske ord og forkortelser er forklart i fotnoter. Alle tekniske ord er dessuten samlet i dokumentet "Ordforklaringer og kilder" bakerst i rapporten.

## Innholdsfortegnelse

Forord.....	1
Leserveiledning .....	1
1 Presentasjon .....	3
1.1 Gruppen .....	3
1.2 Oppdragsgiveren.....	3
1.3 Kunden .....	3
1.4 Veileder og institusjon .....	3
2 Bakgrunn for oppgaven.....	3
3 Overordnet systembeskrivelse .....	4
3.1 Mål .....	4
3.2 Rammebetingelser .....	4
4 Funksjonelle krav .....	4
4.1 Prioritert funksjonalitet .....	5
4.2 Ønsket funksjonalitet.....	5
4.3 Eventuell tilleggsfunksjonalitet.....	5
5 Ikke-funksjonelle krav .....	5
5.1 Produktkrav.....	5
5.1.1 Brukervennlighet.....	6
5.1.2 Effektivitetskrav .....	6
5.1.3 Pålitelighetskrav.....	6
5.2 Prosesskrav .....	6
5.2.1 Utviklingsmetodikk.....	6
5.2.2 Leveransekrav .....	6
5.2.3 Implementasjonskrav.....	7
5.2.4 Krav til standarder.....	7
5.3 Eksterne krav.....	7
5.3.1 Estetiske krav .....	7
5.3.2 Lovmessige krav .....	7
6 Endringshåndtering.....	8
6.1 Interessentene .....	8
6.2 Krav ved forslag til endringer .....	8
6.3 Feilhåndtering .....	8

## 1 Presentasjon

Dette kapittelet inneholder en kort presentasjon av de forskjellige partene i prosjektet.

### 1.1 Gruppen

Gruppen består av Henrik Hermansen og Lars Smeby. Vi tar begge Bachelor i Informasjonsteknologi ved Høgskolen i Oslo og Akershus, og har jobbet sammen ved flere tidligere prosjekter. På bakgrunn av tidligere erfaringer er vi trygge på hvilke mål og forventninger vi har til hverandre og oppgaven.

### 1.2 Oppdragsgiveren

Oppgaven utføres for BEKK Consulting AS, som er et norsk konsulentselskap. De gjennomfører prosjekter for private og offentlige virksomheter innen strategisk rådgivning, utvikling av IT-systemer og design av digitale tjenester. De er i dag omkring 300 ansatte, og har kontorer i Oslo og Trondheim.

Ansvarlig for oppgaven hos BEKK er Christian Schwarz, mens vår faglige veileder er Christoffer Marcussen. Christoffer er med i BEKKs faggruppe for applikasjonsutvikling på mobiltelefon og vil derfor kunne bistå oss i arbeidet.

### 1.3 Kunden

Oppgaven er som sagt gitt oss av BEKK, men det er Statens vegvesen som i utgangspunktet har ønsket denne applikasjonen. Dette innebærer at det er BEKK vi forholder oss til, men at Statens vegvesen stiller sin kompetanse til rådighet.

Vår kontaktperson hos Statens vegvesen er Jan Kristian Jensen.

### 1.4 Veileder og institusjon

Hovedprosjektet utføres ved Institusjon for informasjonsteknologi ved Høgskolen i Oslo og Akershus. Vår interne veileder ved høgskolen er Eva Hadler Vihovde.

## 2 Bakgrunn for oppgaven

BEKK har i 2012 utviklet et REST<sup>1</sup> API<sup>2</sup> for Statens vegvesen. Tanken er at dette APIet skal gjøre data fra NVDB<sup>3</sup> tilgjengelig for alle. Vårt oppdrag er å utforske, teste og demonstrere dette APIet slik at andre inspireres til å ta dette i bruk.

---

<sup>1</sup> Representational State Transfer, REST, er en programvarearkitektur for distribuerte systemer som World Wide Web. REST har etter hvert blitt den dominerende designmodellen for web-APIer.

### 3 Overordnet systembeskrivelse

Det skal utvikles en applikasjon til iPhone<sup>4</sup> som skal gi brukeren nyttige data fra Nasjonal Vegdatabank når han/hun er ute og kjører i hverdagen. Applikasjonen skal ved hjelp av brukerens geografiske posisjon innhente og presentere informasjon som er relevant for det området eller den strekningen som brukeren kjører på, basert på brukerens preferanser.

Informasjonen skal presenteres på en oversiktlig måte, som i så liten grad som mulig tar sjåførens oppmerksomhet vekk fra trafikken. Dette er viktig med tanke på trafiksikkerhet og norsk vegtrafikklov. Tanken er at sjåføren, ved hjelp av en fastmontert telefonholder eller refleksjonen av telefonen i frontruten, enkelt skal kunne få den informasjonen han/hun trenger ved et kort blikk.

APIet for Nasjonal Vegdatabank inneholder veldig mye data, så vår oppgave blir derfor å velge informasjon som demonstrerer APIets muligheter, samtidig som det skal være en applikasjon som er interessant for folk å bruke i hverdagen.

#### 3.1 Mål

- Demonstrere nytteverdien i et åpent API med vegdata fra NVDB
- Utvikle en applikasjon som når en så stor målgruppe som mulig
- Evaluere funksjonalitet og brukervennlighet hos Statens vegvesen sitt nye REST API
- Vurdere iOS<sup>5</sup> som plattform sammenlignet med tilsvarende plattformer

#### 3.2 Rammebetingelser

- Applikasjonen skal i hovedsak benytte data fra NVDB
- Applikasjonen skal utvikles for iOS
- Utviklingen må skje på OS X<sup>6</sup>

### 4 Funksjonelle krav

Funksjonelle krav er krav til funksjonalitet som skal implementeres i systemet. De funksjonelle kravene er delt opp i tre deler, fordelt etter prioritet. Prioriteringen er først basert på hvor viktig vi anser en funksjon for å være, og deretter hvor sannsynlig vi anser det at vi får tid til å implementere den, gitt tidsrammene for prosjektet. Den første delen beskriver funksjonaliteten vi anser som

---

<sup>2</sup> 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.

<sup>3</sup> Nasjonal vegdatabank, database med vegobjekter forvaltet av Statens Vegvesen.

<sup>4</sup> Smarttelefon fra Apple Inc.

<sup>5</sup> iOS er Apples operativsystem for iPhone, iPad og iPod.

<sup>6</sup> OS X er det gjeldende operativsystemet til Mac.

essensiell for å nå målene, og den andre delen beskriver det vi håper å få til. Den tredje delen beskriver forslag til tilleggsfunksjonalitet dersom vi blir tidligere ferdig enn ventet med implementasjonen av den øvrige funksjonaliteten.

#### 4.1 Prioritert funksjonalitet

- Brukeren skal kunne se:
  - Gjeldene fartsgrense for vegstrekningen
  - Hvorvidt han er på forkjørsveg eller ikke
  - Diverse gjeldende varselskilt (som elgfare og farlig vegkryss)
- Brukeren skal kunne endre:
  - Hvilken informasjon som skal vises
  - Hvor tidlig varsler skal vises
  - Hvorvidt det brukes lyd for å varsle brukeren

#### 4.2 Ønsket funksjonalitet

- Informasjonen på skjermen skal kunne speilvendes med det formål å lage et HUD<sup>7</sup> i frontruten
- Brukeren skal kunne se:
  - Avstand til førstkommende rasteplass
  - Avstand til førstkommende SOS-lomme

#### 4.3 Eventuell tilleggsfunksjonalitet

- Brukeren skal kunne se avstand til nærmeste toalett

### 5 Ikke-funksjonelle krav

Ikke-funksjonelle krav er krav til systemets egenskaper og rammeverk. Disse er delt opp i produktkrav, prosesskrav og eksterne krav.

#### 5.1 Produktkrav

Produktkrav omhandler krav til det ferdige produktet som ikke er en direkte funksjon. Dette omhandler krav til brukervennlighet, effektivitet og pålitelighet.

---

<sup>7</sup> 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.

### 5.1.1 Brukervennlighet

- Applikasjonen skal være på norsk
- Applikasjonen skal i så stor grad som mulig følge de 5 E'ene<sup>8</sup>
- Applikasjonen skal ikke kreve interaksjon under bruk (kjøring)

### 5.1.2 Effektivitetskrav

- Applikasjonen skal begrense egen datatrafikk i så stor grad som mulig
- Applikasjonen skal være så energigjerrig som mulig
- Applikasjonen skal begrense bruk av telefonens minne

### 5.1.3 Pålitelighetskrav

- Posisjonering av bruker skal være innenfor en nøyaktighet på 10 meter
- Posisjonering av objektdata skal være innenfor en nøyaktighet på 1 meter
- Dataene som presenteres skal være oppdaterte

## 5.2 Prosesskrav

Prosesskrav omhandler krav til prosessen rundt utviklingen av produktet, deriblant metodikk, leveranser, implementasjon og standarder.

### 5.2.1 Utviklingsmetodikk

- Det skal benyttes Scrum<sup>9</sup> som utviklingsmodell

### 5.2.2 Leveransekrav

Her følger krav til leveransene i prosjektet.

#### 5.2.2.1 Underleveranser

- Det skal leveres en kjørbare applikasjon ved hver leveranse (22.02, 08.03, 22.03, 05.04 og 19.04)

#### 5.2.2.2 Endelig leveranse

- Den ferdige applikasjonen skal være godkjent og tilgjengelig i Apples App Store<sup>10</sup>
- Applikasjonen skal være kjørbare på iPhone 3GS, 4, 4S og 5 (oppdatert til iOS 6.1)
- Kildekoden for leveransen skal lisensieres under GNU GPL<sup>11</sup> og være tilgjengelig via GitHub<sup>12</sup>

---

<sup>8</sup> Prinsipper for brukervennlighet: effective, efficient, engaging, error tolerant og easy to learn (Stone, Jarret, Woodroffe, Minocha, «User Interface Design and Evaluation», 2005, Elsevier Inc.).

<sup>9</sup> Scrum er en iterativ og inkrementell systemutviklingsmodell. Les mer i prosessdokumentasjonen, kapittel 1.4.3.

<sup>10</sup> App Store er Apples digitale butikk for applikasjoner og spill.

<sup>11</sup> GNU General Public License (GNU GPL eller GPL) er en lisens som tillater fri bruk, kopiering og endring av koden.

<sup>12</sup> GitHub er en hosting-tjeneste for systemutviklingsprosjekter som benytter Git.

### 5.2.3 Implementasjonskrav

- Applikasjonen skal utvikles for iOS 6.1
- Applikasjonen skal utvikles med Xcode<sup>13</sup> 4.6 på OS X
- Applikasjonen skal testes på så mange fysiske iOS-enheter som mulig
- Objective-C<sup>14</sup> benyttes som programmeringsspråk
- Applikasjonen skal benytte Cocoa Touch<sup>15</sup> og de andre rammeverkene i iOS 6.1 SDK<sup>16</sup>
- Applikasjonen skal benytte det åpne biblioteket RestKit<sup>17</sup> for kommunikasjon og parsing<sup>18</sup> av data med NVDB APIet
- Brukerpreferanser skal lagres som "User preferences"<sup>19</sup>
- Vegdata skal lagres som Core Data<sup>20</sup>

### 5.2.4 Krav til standarder

- Det skal benyttes JSON<sup>21</sup> for kommunikasjon med NVDB API
- Applikasjonen skal benytte MVC-patternet (som forventes i iOS)<sup>22</sup>

## 5.3 Eksterne krav

Eksterne krav omhandler krav som ikke passer inn under produkt- eller prosesskrav, deriblant estetiske og lovmessige krav.

### 5.3.1 Estetiske krav

- Applikasjonen skal være i samsvar med Apples retningslinjer<sup>23</sup> for design på iOS
- Dataene som presenteres skal være store nok til å kunne sees på 1 meters avstand
- Det skal i så stor grad som mulig benyttes ikoner og grafikk fremfor tekst
- Designet skal følge de 7 designprinsippene<sup>24</sup>

### 5.3.2 Lovmessige krav

- Applikasjonen skal ikke kreve interaksjon under kjøring som strider mot norsk vegtrafikklov

---

<sup>13</sup> Xcode er et program til Mac for utvikling, testing og publisering av programmer til iOS og OS X.

<sup>14</sup> Objective-C er programmeringsspråket som brukes av Apple. Det er basert på C.

<sup>15</sup> Cocoa Touch er et brukergrensesnitt-rammeverk som brukes til utvikling til iOS.

<sup>16</sup> Software development kit (SDK) er verktøy som lar deg utvikle programmer til en spesifikk programvarepakke, et rammeverk, en hardware-plattform, et operativsystem e.l.

<sup>17</sup> RestKit er et Objective-C-rammeverk for iOS som forenkler kommunikasjonen med REST-baserte webtjenester og mappingen av objekter.

<sup>18</sup> Prosessen med å generere eller lese data fra f.eks. en JSON- eller XML-stuktur.

<sup>19</sup> User preferences er en lagringsmetode på iOS.

<sup>20</sup> Core Data er et rammeverk for databaselagring på iOS.

<sup>21</sup> JavaScript Object Notation, en enkel tekstbasert standard for datautveksling.

<sup>22</sup> Model View Controller (MVC) er en måte å dele opp og strukturere koden på i en applikasjon.

<sup>23</sup> <https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html>

<sup>24</sup> Structure, simplicity, consistency, tolerance, visibility, affordance og feedback (Stone, Jarret, Woodroffe, Minocha, «User Interface Design and Evaluation», 2005, Elsevier Inc.).



- Applikasjonen skal inneholde teksten: "Inneholder data under norsk lisens for offentlige data (NLOD) tilgjengeliggjort av Statens vegvesen."
- Applikasjonen skal opplyse om, og avskrive seg og Statens vegvesen ansvar for, eventuelle feil og mangler som måtte forekomme i NVDB APIet

## 6 Endringshåndtering

Dette kapitlet går ut på hvem som kan komme med forslag til endringer i prosjektet, og hvilke betingelser som settes for at disse skal kunne tas til følge.

### 6.1 Interessentene

I utgangspunktet er det Statens vegvesen og BEKK som er interessentene og stakeholders i dette prosjektet. Det er disse som vil kunne fremme forslag eller krav om endringer til produktet som lages. Samtidig er også prosjektgruppen en interessent som kan komme med forslag til endringer hvis dette sees fordelaktig.

### 6.2 Krav ved forslag til endringer

- Endringer i prioritert funksjonalitet skal estimeres av prosjektgruppen og drøftes sammen med alle interessenter før det tas en avgjørelse. Denne typen endringer må godkjennes skriftlig av alle parter.
- Endringer i ønsket funksjonalitet, eventuell tilleggsfunksjonalitet og ikke-funksjonelle krav skal vurderes av prosjektgruppen underveis. Dersom forslaget kommer fra prosjektgruppen skal dette godkjennes av øvrige interessenter.
- Alle interessenter må godkjenne og være enige om eventuelle endringer før de kan inkluderes i kravspesifikasjonen.

### 6.3 Feilhåndtering

Ved feil i systemet skal det ikke startes på ny funksjonalitet før feilen er rettet. Hvis det arbeides med funksjonalitet av lavere prioritet enn der feilen er oppdaget, skal arbeidet avbrytes inntil feilen er rettet. Unntak fra dette må godkjennes av alle interessenter.