

Ruteplanlægning for turister

P1 PROJEKT
GRUPPE A401
SOFTWARE
AALBORG UNIVERSITET
DEN. 18. DECEMBER 2014



AALBORG UNIVERSITET
STUDENTERRAPPORT

**Første Studieår v/ Det Teknisk-
Naturvidenskabelige Fakultet**
Software
Strandvejen 12-14
9000 Aalborg

Titel:

Ruteplanlægning

Projekt:

P1-projekt

Projektperiode:

Oktober 2014 - December 2014

Projektgruppe:

A401

Deltagere:

Christian Dannesboe
Frederik Børsting Lund
Karrar Al-Sami
Mark Kloch Haurum
Mikael Sandegaard Aarsnes
Rabee Mohamad Kaddoura
Søren Lyng

Hovedvejleder:

Jane Billestrup

Bivejleder:

Mona-Lisa Dahms

Synopsis:

Synopsis

Oplagstal: 10

Sidetæl: 80

Appendiks: 3

Afsluttet 18-12-2012

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.

Forord

Forord yes yes

Læsevejledning

Sådan læses rapporten

Christian Dannesboe

Frederik Børsting Lund

Karrar Al-Sami

Mark Kloch Haurum

Mikael Sandegaard Aarsnes

Rabee Mohamad Kaddoura

Søren Lyng

Indholdsfortegnelse

Kapitel 1 Indledning	1
1.0.1 Begreber	2
Kapitel 2 Problemanalyse	3
2.1 Interessentanalyse	3
2.1.1 Prioriteringen	4
2.1.2 Opsummering	5
2.2 Spørgeskema	5
2.2.1 Udformning	5
2.2.2 Resultatbehandling	5
2.2.3 Opsummering	7
2.3 Interviewet	7
2.3.1 Udformning	7
2.3.2 Resultatbehandling	7
2.3.3 Opsummering	8
2.4 Eksisterende løsninger	8
2.4.1 TripAdvisor Offline City Guides	8
2.4.2 FindTheBestRoute.com	9
2.4.3 Opsummering	9
Kapitel 3 Problembeskrivelse	10
3.1 Problemafgrensning	10
3.2 Problemformulering	10
Kapitel 4 Problemløsning	11
4.1 Kravspecifikationer	11
4.1.1 Optimale løsningsforslag	11
4.1.2 Gruppens løsningsforslag	13
4.2 Teorier	14
4.2.1 Grafteori	14
4.2.2 Matematikteori	16
4.3 Implementering	18
4.3.1 Programbeskrivelse	18
4.3.2 Beskrivelse af funktioner	19
4.3.3 Opsummering	24
Kapitel 5 Diskussion	26
Kapitel 6 Konklusion	27
Kapitel 7 Perspektivering	28
Litteratur	29

Appendiks A Spørgeskema	32
A.1 Teori	32
A.1.1 Spørgsmålene	32
A.1.2 Formulering	32
A.1.3 Rækkefølge	33
A.2 Rådata	33
Appendiks B Interview	36
B.1 Teori	36
B.1.1 Enkeltinterview	36
B.1.2 Telefoninterview	36
B.1.3 Spørgeteknikker og metoder til interview	37
B.1.4 Lukkede og åbne spørgsmål	37
B.1.5 Passive teknikker	37
B.1.6 Aktiv spørgeteknik	37

Indledning 1

Hvert år besøger flere millioner turister Danmark, hvilket er godt for den danske økonomi. Når turisterne bruger penge på en dansk vare, service eller oplevelse, bliver det sådan set eksporteret til udlandet – og derfor bliver dette betragtet som en eksportvare. I alt står denne eksport type for 3,6% af den danske eksport. Turisterne har et forbrug på 87,2 mia. kr., hvoraf de udenlandske turister bruger 35,7 mia. kr. altså godt 41%, mens de danske turister står for de resterende 59%. Udover at turismen hjælper det danske samfund økonomisk, skaber turismen ifølge VisitDanmark knap 122.500 fuldtidsjobs. [VisitDanmark, 2013]

Det ses gerne at turisterne kommer tilbage til Danmark igen. Dette sker naturligvis ved, at turisterne nyder deres ophold og får den bedst mulige ferie. Som turist i en storby kan det forekomme svært at finde rundt, og samtidigt virke let at fare vildt. Hvis en turist i København gerne vil se Rundetårn, kan turisten kigge efter det store monument, og gå i den retning hvor attraktionen nu er. Dog kan det ske at turisten undervejs mister tårnet af syne, og pludseligt ved turisten ikke i hvilken retning personen nu skal gå. Turisten kan vælge at bruge sin smartphone, hvis turisten da er i besiddelse af en, og kan eksempelvis gå på internetsiden GoogleMaps. Dette har de danske turister mulighed for, men for en udenlandsk turist vil dataen de bruger på ferie, fra deres smartphone koste mere, end det ville hjemme i deres hjemland[Vodafone]. Her kan turisten så finde en rutevejledning fra punkt A til B, dog vil der kunne opleves problematikker, hvis en flerpunktsrute ønskes. Idette projekt har gruppen valgt at afgrænse sig til at kigge på en enkelt dansk by, i dette tilfælde valgte gruppen Aalborg, da dette var mest oplagt.

For turisten vil planlægning på forhånd være en god ting, hvis turisterne vil nå så mange attraktioner som muligt på en ferie, da tiden kan være begrænset[Youth Central]. Hvis feriedestinationen er i udlandet, ville en offline løsning til ruteplanlægning være optimal, da brugen af mobildata i udlandet kan koste mange penge[TDC, 2014].

Samtidigt kan der spørges, hvad der gør en rute god: Er det hvor hurtigt turisten kommer fra den ene valgte attraktion til den anden? Kan der findes en mere interessant rute, eventuelt med attraktioner der ikke er oplyst i rejsebureauets brochurer, måske en smutvej forbi havnen eller muligheden for en flerpunktsrute mellem attraktionerne? Der kan være mange parametre der spiller ind, når den foretrukne rute skal vælges.

I dette projekt har gruppen valgt at kigge på: I hvilket omfang ruteplanlægning kan hjælpe turister med, at finde den hurtigste eller mest interessante rute mellem attraktioner i Aalborg? Herunder kan der kigges på hvad en interessant rute ville være, hvilken form for rute turisterne foretrækker og i hvilke situationer har turister brug for ruteplanlægning og hvorfor? Derudover vil gruppen kigge på hvilke typer af turister der er relevante i forhold til dette projekt og hvad deres behov kunne være.

1.0.1 Begreber

Ved en 'interessant rute' vil der i denne rapport refereres til en rute, der indkluderer unikke kulturelle og nationale oplevelser, som dog er individuel for turisten. Steder der ville kunne gøre ruten mere interessant vil bl.a. være havnefronten, gågaden eller en tur forbi nogle af byen statuer, som turisten ville kunne passere på sin vej fra A til B.

En 'attraktion' vil i denne rapport forstås som både traditionelle, og utraditionelle attraktioner. Traditionelle attraktioner i Aalborg er Aalborg Zoo, kunstmuseet og lignende. De utraditionelle indkluderer steder som City Syd, Ikea og Jensens Bøfhus.

'Turist' vil i denne rapport være til dels defineret ud fra UNWTOs (UN World Tourism Organisation) beskrivelse, som beskriver to slags turister. En-dagsturister, som maksimalt overnatter én nat på stedet, og en generel turist, som overnatter mere end en nat[Dansk Statistik, 2008]. Udover disse to typer ses også en undergruppe 'erhvervsturist'. Dette er turister, der kommer til stedet med et arbejdsformål.

En turist er en som har et bestemt formål med rejsen, til forskel for besøgende og rejsende. Formålet kan fx være, at opleve den danske kultur.

En 'storby' i Danmark, er i denne rapport en by, omgivet af områder med lavere bebyggelse. Hvor Aalborg fx har Nørresundby og Vejgaard.

Problemanalyse 2

I dette kapitel vil den initierende problemstilling blive analyseret. Dette bliver gjort i form af en interessentanalyse, hvor de væsentlige interesser bliver fundet. Udover interessentanalysen blev der sendt et spørgeskema ud, og et interview blev foretaget med VisitAalborg. Til slut er et par eksisterende løsninger blevet inddraget i dette kapitel.

2.1 Interessentanalyse

“If you want visitors to come back again — and say nice things about your town to others who might come, too — you need to have some good answers at the ready. That means offering things to see and do that are either unique or extraordinary...” [Vitcenda, 2011].

Gruppen vil i dette afsnit, kigge på diverse personer/grupper, der kan fungere som interesser i projektet, altså en person der vil have nytte af projektet. Herefter vil gruppen prioritere disse interesser, alt efter hvor relevante de er i forhold til projektet.

Turister

Turister er en væsentlig interesse i projektet, da en turist ofte vil se hvad byen har at byde på, eller nogle unikke attraktioner [Vitcenda, 2011]. Hvis turisterne planlægger hvad det er, de vil se, kan turisterne spare tid [Youth Central]. Turisterne kan vælge at gå en længere rute, og derved have mulighed for at finde andre ting, som de vælger at bruge deres tid på. Et ruteplanlægningsværktøj vil derfor være interessant for turister, da de derved kan komme til at besøge alle de attraktioner/severdigheder, som de ønsker. En-dagsturister er en mindre interesse i projektet, da en en-dagsturist maksimum overnatter en nat, og generelt har et formål med rejsen, som fx at besøge en attraktion, venner/familie eller er på et kursusophold [for Erhverv & Udvikling, 2012].

Staten

Staten er en interesse i projektet. Hvis der er nogle unikke eller ekstraordinære attraktioner i en by, vil turister huske disse, som gode oplevelser, og nogle vil derfor komme igen. Det er noget som staten er interesseret i, da der kommer flere penge ind i landet [Østergaard og Lubson, 2013]. Her vil et ruteplanlægningsværktøj kunne hjælpe turister med at se nogle af de attraktioner der er i landet, hvis der fx er en top 5 over de attraktioner der er i landet, eller i den by ferien foregår.

Retail-handel

Diverse forretninger er også interesser i projektet, da kendte brands som fx IKEA, H&M, Bilka og lignende lever af deres kunder [Ritzau, 2014]. Ved at implementere disse adresser, i et ruteplanlægningsværktøj, kan det tiltrække turister, og derved øge omsætningen.

VisitAalborg

VisitAalborgs arbejde består af at støtte lokale aktører eller andre aktører, hvis disse aktører byder på nogle turistfremmende aktiviteter/projekter, der har til formål, at hjælpe Aalborg. Det kan fx være at henvise til aktørernes hjemmesider, gennem deres egen hjemmeside, der bliver set af ca. 600.000 årligt. [VisitAalborg, 2014] Til et ruteplanlægningsværktøj er VisitAalborg en vigtig interessent, da de har informationer om turisterne i Aalborg. Ved at inddrage turistkontoret i projektet, vil gruppen gøre brug af deres ressourcer.

2.1.1 Prioriteringen

For at prioritere interessenterne i projekter, og finde de vigtige interessenter, har gruppen valgt gøre brug af indflydelse/medvirken-matrixen, som kan ses på figur 2.1 herunder.



Figur 2.1: Indflydelse/medvirken-matrixen

Gidslerne i projektet, er staten og turisterne. Disse interessenter er blevet kategoriseret som gidsler, da deres indflydelse ikke er vigtig for at gennemføre projektet, men alligevel har de nogle informationer/ressourcer som skal bruges i projektet.

VisitAalborg i dette projekt er ressourceperson, da de har informationer om turister. Derudover kan turistkontoret komme med råd og vejledning for, at en eventuel løsning vil være mest op-timal for turisterne.

De eksterne i projektet er retail-forretningerne. Deres indflydelse og medvirken er ikke nødvendig, for at kunne gennemføre projektet.

Gruppen har i dette projekt, ikke nogle interessenter der passer ind under kategorien grå emminence.

2.1.2 Opsummering

Ud fra interessentanalysen, er gruppen nået frem til, at VisitAalborg er ressourceperson, hvilket gruppen ser som en central interessent i projektet. Det er en nødvendig kilde, for at få nogle brugbare ressourcer. En anden central interessent, er turisterne, da det er dem, der skal gøre brug af ruteplanlægningsværktøjet. De andre interessenter, som gruppen har fundet frem til, er blevet vurderet som mindre vigtige, og derfor vil fokuset ligge hos VisitAalborg og turisterne. Da turister og turistbureauet er projektet væsentlige interessenter i projektet, har gruppen valgt at uddrage information fra disse to interessenter i form af et spørgeskema og et interview. I tilfælde af VisitAalborg og turisternes interesser modstrides, vil gruppen vægte turisternes interesser højest, da VisitAalborg interesser kan skyldes deres kontrakt, med deres partnere [VisitAalborg, 2014].

2.2 Spørgeskema

I dette afsnit vil udformningen af spørgeskemaet og de resultater, som der vil kan udtrages fra spørgeskemaet, blive belyst. I interessentanalysen blev turister placeret som en delvis ressourceperson, og ud fra dette har gruppen valgt at lave et spørgeskema, som skal hjælpe med at besvare den initierende problemstilling. Dette blev gjort med den hensigt at få så mange synspunkter på de opstillede problemstillinger, så det ville være muligt at generalisere ud fra besvarelserne. Spørgeskemaet bliver udarbejdet vha. den kvantitative metode, i form af et internetinterview. Se i appendix A1 og A2 om mere information om den brugte teori og rådata.

2.2.1 Udformning

Først udformede gruppen en række problemstillinger, som gruppen ville have svar på i undersøgelsen:

- Hvilke slags attraktioner tager turister til storbyerne for at se?
- Hvordan planlægger turister deres ferie?
- Hvordan finder turister rundt?
- Har turister problemer med at finde rundt?
- Fortrækker turister den hurtigste eller den mest interessante rute?
- Er den løsning gruppen har i tankerne noget respondenterne ville bruge?

Da gruppen havde valgt et internetinterview i form af en online undersøgelse, valgte gruppen at dele spørgsmålene på Facebook. Dette vil dog give nogle begrænsninger: Spørgeskemaet kan kun ses af personer som gruppen er venner med på Facebook, hvilket mest vil være unge mennesker. Derudover er spørgeskemaet lavet på dansk, for at gøre spørgsmålene så let forståelige som muligt for gruppens venner på Facebook. Da mange af de personer gruppen kender på Facebook sikkert ikke har været turister i Aalborg, blev spørgsmålene rettet mod alle storbyer.

2.2.2 Resultatbehandling

Spørgeskemaet var lagt op på Facebook i tre dage, hvorefter resultaterne blev behandlet. I alt var der kommet 60 besvarelser, hvilket giver gruppen en relativ lille respondentgruppe at arbejde med, dog giver resultaterne nogle klare tendenser, som gruppen har arbejdet ud fra. Herunder kan der ses forklaring på formålet med spørgsmålet, og hvad besvarelserne kan fortælle.

Hvad er vigtigt for dig på din storbyferie?

Formålet med dette spørgsmål er at finde ud af hvad turister gerne vil se eller opleve i en storby, for at kunne se hvad der eventuelt kunne implementeres i projektets løsning.

Ud fra besvarelserne afgivet af respondent gruppen (som kan ses i apendix A2), kan der ses hvad respondenterne vægter højest på deres storbyferie: Se byens seværdigheder, opleve kulturen, maden og shopping. Derudover var der en del der havde kommenteret, at de kom til storbyen for at se sportsbegivenheder. Dette var dog ikke en valgmulighed på spørgeskemaet, så det kan ikke uddrages til projektet, da der kan have været en mulighed for at respondenterne ikke havde tænkt over dette svar.

Hvilke hjælpemidler bruger du til at planlægge din storbysferie?

Formålet med dette spørgsmål er, at finde eksisterende planlægningsværktøjer, hvor der kunne kigges på fordele og ulemper til dette projekt.

I besvarelserne kan der ses en liste af eksisterende planlægningsværktøjer, hvor de både er elektroniske og i form af brochurer og lignende.

Hvilke redskaber bruger du til at finde rundt når du er på storbyferie?

Formålet med dette spørgsmål er at finde ud af om turister brugte redskaber til at finde rundt. Ud fra besvarelserne på dette spørgsmål kan der ses, hvilke redskaber respondentgruppen bruger, til at finde rundt på ferier. Her har respondentgruppen mulighed for at svarer på mere end én ting. Der kan ses at 81,67% bruger diverse kort og brochurer, og 55% bruger elektroniske redskaber til at finde rundt.

Har du nogensinde haft problemer med at finde vej på din storbyferie?

Formålet med dette spørgsmål er at få afklaret om et af de problemer gruppen opbygger projektet op omkring, faktisk er et problem.

Ud fra besvarelserne kan der ses, at hele 68,33% altså hele 2/3, har haft problemer med at finde rundt på deres storbyferie.

Når du skal fra en aktivitet til en anden på din storbyferie, vil du helst tage den hurtigste rute eller en langsommere men mere interessant rute?

Formålet med dette spørgsmål er, at finde ud af hvad turisterne helst ville have, en hurtig rute fra A til B, eller en interessertrute, som er langsommere, hvor turisten får set andre ting på vejen til sin destination.

Der ses tydeligt at respondenterne fortrækker den interessante rute over den hurtigste rute, med henholdsvis 80% for den interessante og 20% for den hurtigste. Folk er mere interesseret i at få flere oplevelser, end at komme hurtigt frem til næste punkt på dagsordenen.

Et program/applikation, som hjælper mig med at finde den hurtigste og/eller mest interessante vej igennem byen, via brugerens valgte "must see" destinationer, ville være noget jeg kunne bruge?

Formålet med dette spørgsmål er at finde ud af om projektet egentlig har nogen interesse hos brugeren.

I dette spørgsmål svarede 90% at den ideelle løsning ville enten kunne bruges eller ønskes.

2.2.3 Opsummering

Ud fra besvarelsene fra spørgeskemaet, blev der konstateret at det er et problem, at turisterne farer vildt på deres storbyferie, da 68,33% af respondenterne har haft problemer med at finde rundt på deres storbyferie. Hele 80% gerne ville have en interessentrute, frem for den hurtigste rute. Ved at gå den interessante rute, får turisterne set mere af byen de befinder sig i, dette kunne eksempelvis være gågader, havnefronter og diverse statuer. Det viste sig også at mange turister stadig bruger kort og brochurer, når de skal finde rundt på deres storbyferie, hvilket var hele 81,67%. Der er i alt 55% der bruger elektroniske redskaber til finde rundt, dog skal det pointeres at respondenterne havde mulighed for at krydse mere end en besvarelse af. Derudover mente lige godt 90% at et program/applikation, som hjælper turisten med at finde den hurtigste og/eller mest interessante vej igennem byen, via brugerens valgte "must see" destinationer, ville kunne bruges.

2.3 Interviewet

På baggrund af interessant analysen kontaktede gruppen VisitAalborg, for at lave et interview vedrørende turisme i Aalborg. Dette interview blev forberedt som et semistruktureret interview, med en passiv spørgeteknik. Formålet med interviewet var, at fremskaffe viden og information fra en professionel kilde, der selv arbejder med emnet.

2.3.1 Udformning

Gruppen valgte at interviewet skulle være et semistruktureret enkeltinterview med passiv spørgeteknik. Herved kan respondenterne snakke frit, så der er mulighed for længere uddybende svar, som ikke var tiltænkt af gruppen på forhånd. Disse teknikker blev brugt, da gruppen ikke havde meget viden om emnet på daværende tidspunkt, og derfor gav dette mulighed for mere information, end der var tiltænkt. For mere information om spørgeteknikkerne, se appendix B1.

Gruppen udformede en interview-guide, som kunne bruges under interviewet. Da respondenterne var meget snaksagelig, blev denne guide ikke brugt til fulde. Ikke alle forberedte spørgsmål blev stillet, og der kom mange nye spørgsmål til, som interviewet udformede sig. Spørgsmålene anvendt i interview-guiden, blev udformet efter gruppens initierende problemstilling og tilhørende underspørgsmål.

2.3.2 Resultatbehandling

Lars Bech nævner i interviewet, at den mest anvendte metode til bestemmelse af antal turister i Aalborg, er gennem målinger foretaget af Horesta, som gøres tilgængelig gennem Danmarks Statistik. Dette kan VisitAalborg bruge til at se fremgang i turismen i Aalborg. Dette gør Danmarks Statistik til en pålidelig kilde, når der ønskes resultater om antal turister, der overnatter i Aalborg. Dog gør Lars opmærksom på, at der også findes turister, som ikke overnatter, hvilket gør det svært, at finde det reelle antal turister, der besøger Aalborg. VisitAalborgs medieudvikler Kim Mikael Jensen oplyser, at en udvidet løsning lignende TripAdvisor ville være interessant for både turisten og VisitAalborg, samtidigt nævner han, at de ikke kender til en lignende løsning endnu. Udfra dette kan der bekræftes, at der er interesse for en

løsning. Dog gør Kim opmærksom på, at tidligere forsøg på løsninger, har været for komplicerede og har for mange funktioner. Han søger simplicitet i et program.

Der bliver i interviewet nævnt, at VisitAalborg er delvist kommunalt ejet. Dette viser, at staten har en interesse for turismen, hvilket stemmer overens med gruppens interessentanalyse.

Under interviewet diskuterer Lars, hvilke attraktioner der er interessante, ifølge tal, og kommer også ind på, hvad han selv mener er attraktioner i Aalborg. Som eksempler nævner han blandt andet Ikea, med 1.6 millioner besøgende om året, Aalborg Zoo og kunstmuseet, hvor han selv også mener, at gågaden, havnefronten og danske butikker (Georg Jensen, Inspiration osv.). "Hvis man køber et eller andet, som man er glad for, så kan man altid huske hvor man har købt det henne." Dette citat, taget fra interviewet med Lars, passer overens med citatet fra interessentanalysen.

2.3.3 Opsummering

Igennem interviewet blev dele af problemstillingen bekræftet, og der blev foreslået nye vinkler, samt stillet krav til en eventuel løsning. Kim Mikael Jensen sagde at programmet/appen skal være simpel, den skal altså ikke have for mange funktioner. Lars gjorde det også mere klart, hvad nogle af de populære attraktioner er i Aalborg. Interviewet er også blevet brugt til bestemmelse af begrebet "attraktion".

2.4 Eksisterende løsninger

I spørgeskemaet blev der opremset en hel række af hjælpemidler som respondenterne bruger på deres storbyferie, hvilket til projektet vil være eksisterende løsninger. I det kommende afsnit vil der blive kigget på TripAdvisor app, hvilken var en af de hjælpemidler som respondent gruppen havde nævnt. Der udover har gruppen fundet Fidinthebestrute.com, som vha. Google Maps kan lave en flerpunktsrute.

2.4.1 TripAdvisor Offline City Guides

TripAdvisor har en offline app, der kan hjælpe med at guide turister rundt, i den by de er rejst til. Den har mange forskellige funktioner, som fx et kort indlagt i appen. Dette kort kan være effektivt, hvis brugeren har forberedt sig hjemmefra. Dette skyldes, at turisten kan downloade et kort over den by, brugeren skal besøge, og derved vil den fungere offline. Grunden til at det er effektivt for turister, kan skyldes at mobildata kan være dyrt i udlandet[TDC, 2014].

Udover et kort, har app'en også nogle informationer omkring de mange forskellige byer. Disse informationer har turisten ligeledes mulighed for at downloade, så de også er tilgængelige offline. Ved hjælp af disse informationer, kan turisten fremskaffe sig hjælp, hvis turisten fx er interesseret i at finde en restaurant, finde et hotel, se en bestemt attraktion eller lignende. Ønsker brugeren at besøge en af attraktion eller lignende, kan der ved hjælp af en knap, klikkes frem til en lokalisation, som turisten enten selv kan finde vej til, eller benytte en anden knap i app'en og få indlagt ruten i det downloadede kort.

Hvert af disse kategorier indeholder en "Best in Town"-funktion, som er en liste over de mest populære attraktioner, ifølge TripAdvisor brugere af app'en, da der er et point-system, som giver brugere af app'en mulighed for at vurdere og skrive kommentar til de enkelte attraktioner, i en skala på 1-5.

TripAdvisors app har mange gode funktioner. En af de gode funktioner, er det offline kort, der giver mulighed for at undgå brugen af mobildata, på en udlandsrejse, og gør det muligt hele tiden at have et kort ved hånden. Herudover kan der fås et indblik i, hvilke ting der er at se og opleve i den valgte by, med kommentarer og ratings fra andre brugere, der har besøgt disse steder. Appen har også nogle mangler, som fx at vælge flere seværdigheder på listen, og give en rute mellem disse seværdigheder, så det er muligt at få en flerpunktsrute.

2.4.2 FindTheBestRoute.com

Google Maps er begrænset til kun at kunne vise vejen fra et punkt til et andet. Det har FindTheBestRoute.com taget kampen op imod, og har derfor lavet en hjemmeside på FindTheBestRoute.com, hvor den hurtigste rute mellem maksimalt 10 forskellige adresser kan beregnes. FindTheBestRoute.com, udnytter Google Maps JavaScript API v3, altså en grænseflade til Google Maps, der tillader andre programmer at benytte Google Maps, til fx at få vist et kort, eller beregne en rute [FindTheBestRoute.com, 2014].

Selvom der på maps.google.dk ikke er mulighed for at indtaste forskellige destinationer, og få anvist den hurtigste rute imellem punkterne, så har Google Maps faktisk allerede funktionaliteten indbygget til at foretage denne beregning, baseret på "The Travelling Salesman Problem".

For findthebestroute.com, er det derfor simpelt at sende en anmodning til Google, der indeholder informationer om de forskellige destinationer der skal forbindes med en rute. Google foretager så beregningerne, og sender den bedste rute tilbage til findthebestroute.com, hvor de så kan vise ruten til deres brugere [Google, 2014].

2.4.3 Opsummering

TripAdvisor har mange gode funktioner, så som offline kort, "Best in Town" og information om de enkelte attraktioner. TripAdvisor findes som en app, så den er tilgængelig på ferien. TripAdvisor har dog ikke mulighed for at lave en flerpunktsrute.

FindTheBestRoute.com har intet af det som TripAdvisor har, den har dog muligheden for at lave en flerpunktsrute, med maksimalt 10 adresser. Problemet med FindTheBestRoute.com er at den kun er tilgængelig på internettet og derved er den mere besværlig på ferien.

Disse synspunkter kan bruges til at udforme krav til projektets løsning.

Problembeskrivelse³

I dette afsnit vil den endelige problemformulering blive beskrevet, samt kravene til den endelige løsning. Til sidst afgrænses der i forhold til gruppens evner og de krav der er blevet opstillet.

3.1 Problemafgrænsning

Ud fra denne analyse, kan der konkluderes, at VisitAalborg og turisterne er ressourcepersoner, hvoraf turisterne har større indflydelse på programmet, da der er de endelige brugere. VisitAalborg kan bruges som guider til, hvad der kan være af indhold i programmet, men der skal stadig tages højde for, at det er i deres interesse, at få deres arbejds partnere med ind i programmet, selvom det ikke i alle tilfælde er til turistens interesse. Ved spørgeskemaet blev der uddraget, at turister allerede har løsninger fra punkt til punkt, hvor der i eksisterende løsninger blev påpeget, at der også findes løsninger for flerpunktsruter. TripAdvisor viste, at der også er programmer, som tager højde for brugerens interesse, men alt taget i betragtning, er der ikke en løsning der kombinerer alle disse funktioner, som der belyses i gruppens spørgeskema og interview, til at være i brugerens bedste interesse: En simpel løsning, der inddrager brugerens interesse, og foreslår yderligere punkter til en mere interessant rute. Dette kunne endda optimiseres ved en offline-funktion, som TripAdvisor også gør brug af ved et offline kort.

3.2 Problemformulering

Hvordan udvikles der en softwareløsning, der hjælper turisten med at finde rundt i en storby, på en interessant rute mellem turistens egne valgte attraktioner?

Problemløsning 4

I dette kapitel vil gruppen beskrive hvilke krav gruppen stiller til projektet, hvilke teorier og algoritmer der er vigtige i forhold til projektet og hvordan gruppen har valgt at lave programmet, altså implementeringsprocessen.

4.1 Kravspecifikationer

I dette afsnit vil gruppen vurdere, hvilke krav der skal indegå i en løsning. Heri vil der både blive opstillet krav til en optimal løsning og til en afgrænset løsning, som gruppen mener at være realistisk, at kunne lave.

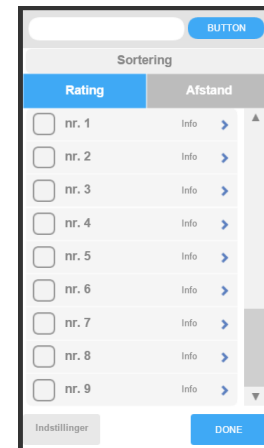
4.1.1 Optimale løsningsforslag

For at kunne udvikle en softwareløsning, der besvarer gruppens problemformulering, er det vigtigt at definere nogle krav til programmet. Til en optimal løsning, har gruppen vurderet, at der skal være følgende krav:

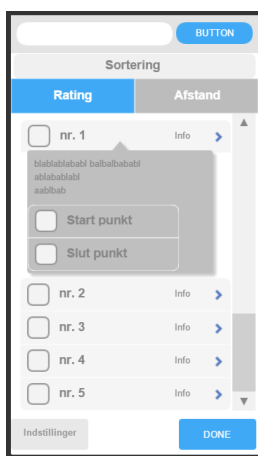
- Programmet skal udvikles som en applikation til moderne smartphones, inklusiv iOS, Android og Windows. Phone.
- Programmet skal vise et kort med ruten, og kortet skal vise attraktioner der er tæt på.
- Programmet skal kunne beregne den korteste rute mellem en række punkter.
- Programmet skal vise rutevejledningen på samme måde som normale GPS-enheder.
- Programmet skal give mulighed for at bedømme attraktioner, og derved tildele dem en rating, så de mest populære attraktioner kan findes.
- Programmet skal give forslag til en anden rute, der inkludere attraktioner der ligger tæt på ruten.
- Programmet skal kunne downloade en offline version af den rute der er valgt, inklusiv kortet for det omkringliggende område.

Til disse krav har gruppen konstrueret nogle skitser af den optimale løsning, for at give et billede af hvordan det eventuelt kunne se ud.

Gruppen ønsker, at programmet skulle fungere på den måde, at der findes to valg muligheder, forholdsvis rating og afstand, hvoraf rating viser en række attraktioner med en værdi, baseret på hvad brugerne har valgt at rate den. Funktionen afstand, vil vise hvor stor en afstand der er fra det punkt hvor brugeren står, til en attraktion. De attraktioner, som brugeren ønsker at se, skal brugeren blot tjekke af, ved at klikke på attraktionerne, og de vil derefter blive tilføjet til den nuværende rute. Figur 3.1 viser en skitse af brugergrænsefladen.

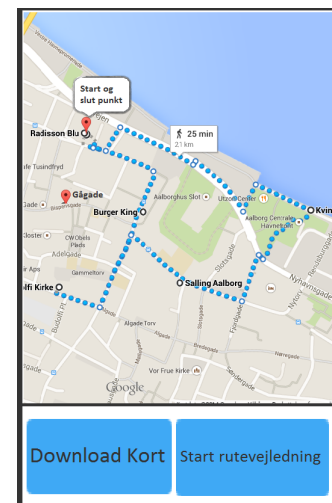


Figur 3.1:
Brugergrænseflade



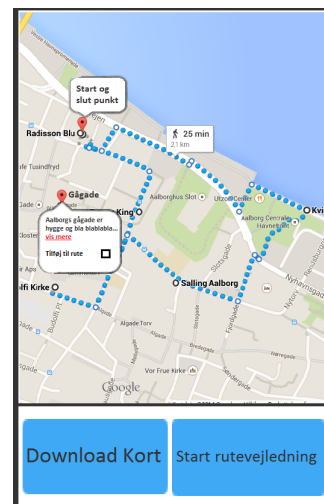
Figur 3.2:
Udvidet brugergrænseflade

Når der er valgt nogle ønskede destinationer/attraktioner, skal programmet fremvise en rute. Ruten skal vise hvor lang hele ruten er, og hvor lang tid det tager at gå ruten. Der skal desuden være to funktioner, når ruten bliver vist. Der skal være mulighed for at downloade kortet på mobilen, og derved gør det muligt at anvende programmet, uden brug af internet. Den anden funktion skal starte rutevejledningen, som fungerer som en ganske almindelig GPS. En skitse af dette kan ses på figur 3.3.



Figur 3.3: Rute - Ruten er hentet fra Maps.Google.com

Der skal desuden også her være mulighed for at få information om attraktionerne, der er i nærheden af den valgte rute. Der skal være en "tilføj til nuværende rute-funktion, som tilføjer de valgte attraktioner, som er i nærheden til den rute, der allerede er lavet. Figur 3.4 viser hvordan det eventuelt kunne laves.



Figur 3.4: Udvidet rute -
Ruten er hentet fra
Maps.Google.com

4.1.2 Gruppens løsningsforslag

Gruppen har gennem spørgeskema og interview, fået stillet en række krav til løsningen, af turister og VisitAalborg. Gennem spørgeskemaet, blev det konkluderet, at det vigtigste for turister, er at de kan opleve byen på en interessant rute. Derudover har turistbureauet givet udtryk for, at løsningen gerne skal være så enkelt som muligt, altså meget få funktioner, så brugeren ikke bliver forvirret, da de mener, at det er i turistens bedste interesse.

Der er blevet stillet krav fra universitets side, om at programmet skal være et lille specifikt program i C, af høj kvalitet. Dette stemmer godt overens, med de krav der er blevet stillet fra turistbureauets side.

Ud fra dette, har gruppen opsat nogle krav for gruppens løsningsforslag, og de er som følgende:

- Programmet skal kunne beregne den korteste rute mellem en række punkter.
- Programmet skal være i stand til at give forslag til en anden rute, der inkluderer attraktioner der ligger tæt på ruten.
- Programmet skal som output, give en liste over rutens destinationer, sorteret efter tiden til attraktionerne.

Da dette er et P1 projekt, og gruppen er begrænset af både tid og erfaring, har gruppen valgt at begrænse softwareløsningen, på følgende punkter:

- Rutevejledningen bliver i fugleflugtslinje.
- Brugeren kan kun vælge destinationer ud fra en række forudbestemte punkter.
- Tekstbaseret brugergrænseflade.

På baggrund af kravene og afgrænsningen, har gruppen tænkt sig at lave et program, som har nogle forudbestemte destinationer, der dækker over destinationerne i Aalborg, hvorefter brugeren vælger de destinationer han/hun ønsker at besøge. Programmet vil ud fra disse punkter, beregne den korteste rute, og undersøge om der er andre attraktioner, som ligger tæt på ruten, og spørge brugeren, om det kunne være interessant at besøge disse steder. Hvis ja, vil disse punkter også blive inkluderet. Resultatet bliver en liste over destinationerne, der står i rækkefølge, så turisten ved hvilken rækkefølge de skal besøge dem i, for at få den mest optimale rute.

4.2 Teorier

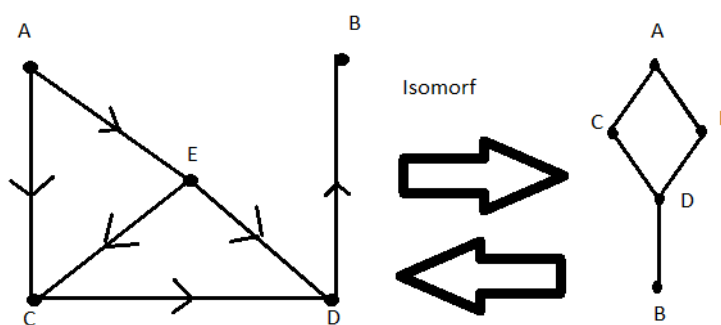
Indledning mangler.

4.2.1 Grafteori

Grafteori er et afsnit i denne rapport, som omhandler en generel forklaring på graf teori, hvorefter teorien bag "Nærmeste Nabo Algoritme" vil blive beskrevet, herefter forklares "udregningstid" i rute-algoritmer, og til sidst Traveling Salesman Problem. Alt dette beskrives, for at give et udgangspunkt for implementering af en hensigtsmæssig algoritme i programmet for denne rapport.

Matematikken bag graf teori er ét aspekt af emnet, hvor visualisering og tegning er en anden. Den matematiske del behandler kombinationerne af knuder og kanter. En knude er et punkt, i vores tilfælde en attraktion som skal besøges, hvor en kant er vejen derhen. En kant er derfor længden fra ét punkt til det næste [JMT, 2010]. I graf teori er begrebet "graf" mere fleksibelt, da punkterne ikke nødvendigvis har x, y eller z værdier, alt efter antallet af dimensioner man behandler det i. Grafen i denne form er en afbildning af punkter i den form, hvor det virker hensigtsmæssigt. Heraf opstår isomorfe modeller, som er forskellige afbildninger, af selv samme graf.

Denne matematik-type er stadig under udforskning, da der endnu ikke er en fuldstændig løsning på problemer i teorien, for blandt andet "Traveling Salesman Problem". Heriblandt findes mange typer af problemer, hvor forskellige teoretiske løsninger kan bruges. En af problematikkerne vedrørende Traveling Salesman Problem er, at der ønskes både en optimal rute, og en udregningstid



Figur 5.1: Brugergrenseflade

der er hensigtsmæssig. Dette problem opstår i det, at en optimal rute skal findes mellem et hvis antal byer (knuder), hvor man i ét kan lave en hurtig estimeret "kort" rute, hvis man starter med at lave tilfældige kanter fra knuderne, dog ikke mere end to kanter per knude, og derefter tester for, hvorvidt to nye kanter er kortere end to eksisterende kanter. På et tidspunkt vil en semi-optimal rute findes, dog er denne ikke nødvendigvis den fuldt optimale rute. Dette skyldes, at der igennem forløbet med udskiftning af kanter muligvis er truffet valg om rute som fører til, at en kortere kant ikke kan findes lokalt, men at den sammenlagte rute stadig ikke er optimal. Hvis der er fundet korte kanter lokalt, kan dette stoppe søgen i en kortere kant, da den korteste lokale kant er fundet, men ikke den korteste kant, hvis der tages hensyn til den sammenlagte kant-værdi. [Kolpack, 2013]

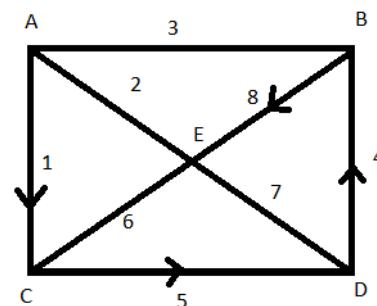
Nearest Neighbor Algoritme

En af disse løsninger er den Nærmeste Nabo Algoritme (NNA, Nearest Neighbor Algorithm), som behandler et problem der opstår, når en række knuder skal indgå, og kun skal indgå én gang, hvilket betyder, at der ikke må være en løkke (loop). Dette opnåes gennem brug af Hamiltonian Paths, hvilket er en "rute" gennem knuder på grafen. I NAA vil en kant have en værdi, og disse værdier er bestemmende for, hvilken kant der skal følges. Fra den knude der behandles, skal kan-

ten med den laveste værdi følges. Dette kan dog optimeres, ved at lave en matrice over kanterne fra alle knuder. Her tages højde for hvilke kanter der samlet set giver den korteste rute, uden brug af løkker. [Mathispower4u, 2013b]

I dette tilfælde (figur 4.2), er den endelige rutes længde: $1+5+4+8 = 18$. Spørgsmålet er så, er dette den koreste rute? Herefter opstilles en matrice, der beskriver alle kanter.

	A	B	C	D	E
A	-	3	1	-	2
B	3	-	-	3	8
C	1	-	-	5	6
D	-	4	5	-	7
E	2	8	6	7	-



Figur 4.2: Hamiltonian Path.
Følger kanter med laveste værdier.

De mulige ruter er:

$$ABDCE = 3+4+5+6 = 18.$$

$$ACDBE = 1+5+4+8 = 18.$$

$$AEBDC = 2+8+4+5 = 19.$$

$$ABECD = 3+8+6+5 = 22.$$

$$ABEDC = 3+8+7+5 = 23.$$

$$ACEBD = 1+6+8+4 = 19.$$

$$ACEDB = 1+6+7+4 = 18.$$

$$AECDB = 2+6+5+4 = 17.$$

Den korteste rute er altså AECDB, hvilket er 1 kortere end den antagede rute. Den optimale Hamiltonian Path er derfor denne rute. Dette tager NNA ikke højde for, da den starter i en valgt start-knude, og derefter følger kanten, med den derfra laveste værdi. Det smarte ved NNA er, at den ikke kræver meget kraft for en computer at udføre, hvorimod at finde den optimale Hamiltonian rute, vil være langt mere kompliceret. NNA tager dog ikke højde for, hvad konsekvenser de skridt den tager, har for det endelige resultat.

Dijkstra's algoritme

Udover NNA, findes også Dijkstra's algoritme, hvor første step er, at bestemme ende-knuden, og sætte dens distance til nul. Denne knude sættes til at være den første knude, som behandles. I det en knude er checket færdig, vil denne knude markeres som "besøgt", og kanten med den mindste værdi følges, og næste knude markeres som "nuværende" knude. En kant bliver kun fulgt, hvis det er den korteste rute, tilregnet tidligere kanter. Problematikken med Dijkstra's algoritme i forhold til dette projekt er, at den checker den korteste rute fra start-knude til slut-knude, men den inkluderer ikke nødvendigvis alle knuder som oplyses. I det denne rapport er afgrænset til fugleflugtslinjer, vil Dijkstra's ikke være den optimale. Hvis en rute igennem en by, hvor der er tilregnet veje, stier og andre knuder, vil Dijkstra's være det bedste valg. Denne algoritme vil også være i brug ved den optimale løsning. Ved brug af Dijkstra's algoritme, vil den nuværende rute altid blive testet for, hvorvidt ruten der undersøges efter, er kortere eller længere end den hidtil korteste rute. Hvis den er kortere, vil denne rute blive sat som den hidtil korteste rute. [Mathispower4u, 2013a]

4.2.2 Matematikteori

Essensen i dette projekt er at finde en flerpunktsrute mellem nogle valgte attraktioner, hvor brugeren skal have mulighed for, at vælge nogle attraktioner til deres rute. Gruppen vil ikke diktere hvad en interessant rute er for brugeren, derfor skal de have muligheden for at vælge de foreslåede attraktioner til eller fra.

Der tages nu udgangspunkt i figur 5.1. En del af brugerens rute ligger fra attraktion a til attraktion b. Der skal nu tjekkes om der ligger andre attraktioner mellem afstanden fra a til b (eller AB), og med bredden AD hvor brugeren vil blive spurgt om denne attraktion skal tilføjes til ruten. AD er i projektets program sat til at være $V * 2$. Dette vil blive udregnet vha. vektorer. Hvis der antages at punktet P er en attraktion som programmet skal tjekke, ligger denne inden for længden af ruten AB og bredden AD. Dette tjekkes med følgende formel:

$$0 < AP \cdot AB < AB \cdot AB \wedge 0 < AP \cdot AD < AD \cdot AD$$

Hvor prikproduktet af vektorerne AP og AB, skal være større end 0 og mindre end prikproduktet af vektorerne AB og AB. Det samme vil gælde for AD i stedet for AB.

Lad nu som om det de informationer der kendes er punkterne a og b, samt længden på vektor ab som vil være 6 og vektoren vil hedde:

$$\begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

Først ønskes punktet A findes, som gøres ved først at finde tværvektoren. Tværvektoren findes ved at bytte 1. og 2. koordinat rund og ændre fortegn på første koordinaten:

$$\begin{matrix} a1 & = & -a2 \\ a2 & = & a1 \end{matrix}$$

Tværvektoren hedder:

$$\begin{bmatrix} 6 \\ 0 \end{bmatrix},$$

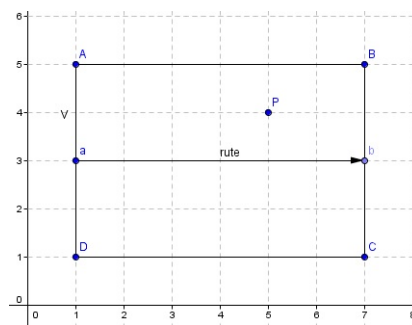
og har udgangs punkt fra punktet a.

I dette eksempel skal der søges efter ekstra attraktioner langs ruten, svarende til $1/3$ af rutens længde. Så for at finde koordinaterne til punktet A, finder vi først en enhedsvektor for tværvektoren, dette gøres med formelen:

$$\vec{e} = \frac{1}{\vec{a}} * \vec{a}$$

Dette giver en vektor:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix},$$



Figur 5.1: Eksempel på om en attraktion er indenfor punkt a og b.

som også har udgangspunkt fra punktet a. Som tidligere nævnt søges der efter ekstra attraktioner langs ruten, svarende til $1/3$ af rutens længde, så enhedsvektoren multipliceres med to, hvilket giver en vektor:

$$\begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Denne vektor lægges til koordinaterne til punktet a, hvilket vil give punktet:

$$A = \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}.$$

Punktet D vil således findes ved at tage vektoren fra før og multiplicere med -2 og lægge punktet A til:

$$D = \begin{bmatrix} 0 \\ 2 \end{bmatrix} * -2 = \begin{bmatrix} 0 \\ -4 \end{bmatrix} + \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Dog vil der først findes en vektor AP mellem punkterne A og P med formlen:

$$\overrightarrow{AP} = \begin{matrix} X2 - X1 \\ Y2 - Y1 \end{matrix}$$

Vektor AP: A(1,5) og P(5,4):

$$\overrightarrow{AP} = \begin{bmatrix} 5 - 1 \\ 4 - 5 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

Vektor AB er allerede kendt, da det er det samme som \overrightarrow{ab} .

For at projicere AP på AB skal følgende formel benyttes:

$$b_a = \left(\frac{a \cdot b}{|a|^2} \right) * a$$

Med denne formel vil vektoren b blive projiceret på vektoren a. I tælleren findes prikproduktet som kan findes ved at:

$$a \cdot b = \begin{matrix} X1 * X2 \\ Y1 * Y2 \end{matrix}$$

I nævneren findes længden på vektor a i anden, som kan regnes ved at sige:

$$\sqrt{ax^2 + ay^2}$$

Hvis der forsat kigges på eksemplet med figur 5.1, vil projektionen af AP på AB se således ud: Prikproduktet af vektorerne:

$$\overrightarrow{AP} \cdot \overrightarrow{AB} = \begin{bmatrix} 4 & 6 \\ -1 & 0 \end{bmatrix} = 4 * 6 + (-1) * 0 = 24$$

Længden af AB opløftet i anden vil være:

$$\sqrt{6^2 + 0^2} = 6$$

Ud fra dette kan vektoren fra projektionen af AP på AB findes:

$$\frac{24}{36} * \begin{bmatrix} 6 \\ 0 \end{bmatrix} \rightarrow \frac{24}{36} * 6 \wedge \frac{24}{36} * 0 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

Hvor resultatet vil give en ny vektor:

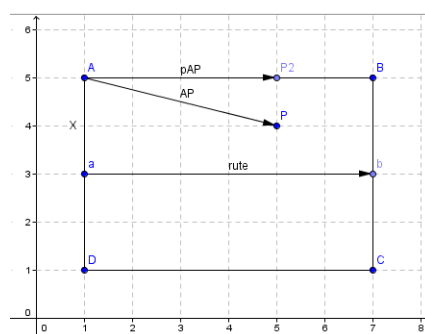
$$\begin{bmatrix} 4 \\ 0 \end{bmatrix},$$

som også vil have startpunkt i A. Hvis der igen kigges på formelen:

$$0 < AP \cdot AB < AB \cdot AB \wedge 0 < AP \cdot AD < AD \cdot AD$$

overholder punktet P første del, og ovenstående metode skal derfor gentages med vektoren AD i stedet for AB, for matematisk at finde ud af om punktet ligger inden for den afsatte bredde og længden af ruten a til b. Ved udregning af projektionen af AP på AD vil den nye vektor hedde:

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$



Figur 5.2: Fortsat eksempel på om en attraktion er indenfor punkt a og b.

4.3 Implementering

I dette afsnit vil programmet blive beskrevet, både en overordnet programbeskrivelse og en dybdegående forklaring af programmets funktioner. Derudover vil der blive beskrevet hvordan brugeren interagerer med programmet.

4.3.1 Programbeskrivelse

Programmet begynder med at indlæse alle de forudbestemte tilgængelige attraktioner fra en tekstfil. Alle disse attraktioner bliver vist på en liste for brugeren i kommandoprompten med numre ud for hver attraktion. Brugeren vælger så hvilke attraktioner vedkommende har lyst til at besøge, ved at indtaste attraktionens nummer, hvor attraktionen så vil blive tilføjet til en liste. Denne liste kører gennem programmet, og den korteste rute bestemmes. Programmet tjekker derefter, for nærliggende attraktioner der kan tilføjes til ruten, på samme måde som i begyndelsen da de valgte attraktioner til deres rute. Hvis brugeren vælger ekstra attraktioner til deres rute, kører programmet igen, og den nye rute beregnes. Når ruten er beregnet, vil ruten blive vist på skærmen, i den beregnede rækkefølge. Den samlede længde af ruten vil også blive vist.

Til programmet bruges fire forskellige structs. Den første struct er "attraktion", som indeholder to strings, "navn" og "adresse", to doubles, "kmFraGreenwich" og "kmFraAekvator", og en int, "besoegt", der beskriver om attraktionen er besøgt eller ej. Denne struct bruges til at opdele og gemme de enkelte attraktioners data i programmet.


```
1 typedef struct {
2     char navn[MAX_STRING];
3     char adresse[MAX_STRING];
4     double kmFraGreenwich, kmFraAekvator;
5     int besøgt;
6 } attraktion;
```

Næste struct er "naboRute", som indeholder en double, "ruteLaengde", og et array af attraktioner, "rute". Denne struct bruges til at lave ruter, hvor "rutelaengde" er længden af ruten og arrayet indeholder rækkefølgen af attraktioner på ruten.

```
1 typedef struct {
2     double ruteLaengde;
3     attraktion rute[ANTAL_ATTRAKTIONER];
4 } naboRute;
```

Næste struct er "kant", som indeholder to attraktioer, "startAttraktion" og "slutAttraktion", og en double, "laengde". Denne struct bliver brugt til at gemme data omkring afstanden mellem to punkter.

```
1 typedef struct {
2     attraktion start;
3     attraktion slut;
4     double laengde;
5 } kant;
```

Sidste struct er "vektor", som indeholder to doubles, "x" og "y". Denne struct bruges til at gemme x og y koordinater for vektorer (og i nogle tilfælde punkter) under et navn.

```
1 typedef struct {
2     double x, y;
3 } vektor;
```

4.3.2 Beskrivelse af funktioner

Attraktionerne til programmet bliver indlæst fra en .txt fil når programmet køres. I programmet bliver filen indlæst i en funktion som hedder initialiserAttraktioner. Hvis filen ikke er tom vil elementerne, vha. funktionen fscanf, blive indlæst i grupper af 4, hvor de bliver indlæst som "attraktion" hvilket er defineret som et struct. Hvis filen er tom, vil en advarsel blive vist i prompten og lukke programmet ned. Når alt information er indlæst fra filen, og den ikke længere er nødvendig, lukkes filen.

```
1 void initialiserAttraktioner(attraktion *attraktioner){
2     FILE *input_file_pointer;
3     int i = 0;
4     double lndgrad;
5     double brdgrad;
6
7     input_file_pointer = fopen("attraktioner.txt", "r");
8
9     if(input_file_pointer != NULL){
10         while(fscanf(input_file_pointer, " %s %s %lf %lf", attraktioner[i].navn,
11                     attraktioner[i].adresse, &brdgrad, &lndgrad) == 4){
12             attraktioner[i].kmFraGreenwich = lndgrad * KM_PR_LNDGRAD;
13             attraktioner[i].kmFraAekvator = brdgrad * KM_PR_BRDGRAD;
14             attraktioner[i].besøgt = 0;
15             i++;
16         }
17     }else{
18         printf("kunne ikke aabne fil\n"); exit(1);
19     }
```

```

19         fclose(input_file_pointer);
20     }

```

Brugeren bliver præsenteret for en liste over alle tilgængelige attraktioner fra inputparameteren, og attraktionernes tilhørende nummer i funktionen "valgafAttraktioner". Brugeren bliver bedt om at indtaste attraktionernes matchende numre, hvilket vil blive tilføjet til en liste. Taster brugeren 0, bliver brugeren præsenteret for sine valg, og beregningen af den korteste rute igangsættes. Programmet returnere listen af valgte attraktioner.

```

1 void valgafAttraktioner(attraktion *attraktioner, attraktion *valgteAttraktioner, int
    *antalValgteAttraktioner, attraktion *ikkeValgteAttraktioner){
2     int i = 0, j = 0, k = 0, l = 0, m = 0, n = 0, o = 0, valgt = 0;
3
4     for(i = 0; i < ANTAL_ATTRAKTIONER; i++){
5         printf("%d: %s\n", i+1, attraktioner[i].navn);
6     }
7
8     printf("vaelg de attraktioner du oensker at se ved at skrive det tilhoerende tal.\n");
9     printf("vaelg et tal (svarende til en attraktion) af gangen og tryk enter efter hver indtastet
        tal\n");
10    printf("indtast ikke samme tal 2 gange\n");
11    do{
12        if(scanf("%d", &k) != 1){
13            printf("Fejl i indlaesning. Farvel.\n"); exit(0);
14        }
15        if(k != 0){
16            valgteAttraktioner[j] = attraktioner[k-1];
17            j++;
18        }
19    } while(j < ANTAL_ATTRAKTIONER && k != 0);
20
21    for (l = 0; l < ANTAL_ATTRAKTIONER; ++l)
22    {
23        valgt = 0;
24        for (m = 0; m < j; ++m)
25        {
26            if(strcmp(attraktioner[l].navn, valgteAttraktioner[m].navn) == 0){
27                m = j;
28                valgt = 1;
29            }
30        }
31        if(valgt != 1){
32            ikkeValgteAttraktioner[n] = attraktioner[l];
33            n++;
34        }
35    }
36    *antalValgteAttraktioner = j;
37 }

```

Funktionen "findNaboRute" benytter NNA (Nearest Neighbour Algorithm, eller Nærmeste Nabo Algoritme), til at finde den korteste rute mellem en række attraktioner for et bestemt startsted, der er opgivet i inputparametrene. Efter at have fundet den korteste rute ud fra NNA som en beskrevet i teori afsnittet X, returnere den et array med ruten og distancen for denne rute.

```

1 void findNaboRute(attraktion *valgteAttraktioner, int antalValgteAttraktioner, attraktion
    *startAttraktion, kant *kanter, attraktion **tempRute, double *ruteLaengde){
2     int i = 0;
3     double lavesteLaengde = 10000;
4     *ruteLaengde = 0;
5
6     tempRute[i] = startAttraktion;
7
8     for(i = 0; i < antalValgteAttraktioner-1; ++i){ /*da der f.eks. kun er 4 kanter imellem 5 punkter*/
9         tempRute[i]->besoegt = 1;

```

```

10         int j = 0;
11         for (j = 0; j < antalValgteAttraktioner; ++j)
12         {
13             if(valgteAttraktioner[j].besoegt != 1 && findDist(*tempRute[i],
14                 valgteAttraktioner[j], kanter) < lavesteLaengde){
15                 lavesteLaengde = findDist(*tempRute[i], valgteAttraktioner[j], kanter);
16                 tempRute[i+1] = &valgteAttraktioner[j];
17             }
18             *ruteLaengde += lavesteLaengde;
19             lavesteLaengde = 10000;
20         }
21         tempRute[antalValgteAttraktioner] = startAttraktion;
22         *ruteLaengde += findDist(*tempRute[antalValgteAttraktioner-1], *tempRute[antalValgteAttraktioner],
23             kanter);
24     }

```

"findKortesteNaboRute" benytter funktionen "findNaboRute", til at finde ud af hvilket startsted der giver den korteste rute. Dette gøres ved at sætte en variable til en stor værdi, som rutedistancen ikke vil gå over, og opdatere den hvis "findNaboRute" returnere en distance for ruten gennem de givne attraktioner med en given start attraktion, der er lavere end de forgående ruter. "findKortesteNaboRute" returnere et array med en korteste rute, og den samlede længde af denne rute.

```

1 void findKortesteNaboRute(attraktion *valgteAttraktioner, int antalValgteAttraktioner, attraktion
   *ruteAttraktioner, kant *kanter, double *samletLaengde){
2     /* input er valgteAttraktioner arrayet, og kanter arrayet*/
3     /*output er ruteAttraktioner og samletLaengde*/
4     double ruteLaengde;
5     attraktion *tempRute[antalValgteAttraktioner+1];
6     *samletLaengde = 100000;
7
8     int i;
9     int h;
10    int j;
11    for (i = 0; i < antalValgteAttraktioner; ++i)
12    {
13        for (h = 0; h < antalValgteAttraktioner; ++h)
14        {
15            valgteAttraktioner[h].besoegt = 0;
16        }
17        findNaboRute(valgteAttraktioner, antalValgteAttraktioner, &valgteAttraktioner[i], kanter,
18            tempRute, &ruteLaengde);
19        if(ruteLaengde < *samletLaengde){
20            *samletLaengde = ruteLaengde;
21            for (j = 0; j < antalValgteAttraktioner+1; ++j)
22            {
23                ruteAttraktioner[j] = *tempRute[j];
24            }
25        }
26    }

```

Funktionen "bereg_n_dist" er en implementering af pythagoras's sætning til at finde længden imellem 2 punkter. Funktionen returnere den beregnede distance mellem de to attraktioner.

```

1 double beregn_dist(attraktion startAttraktion, attraktion slutAttraktion)
2 {
3     return sqrt(pow(startAttraktion.kmFraGreenwich - slutAttraktion.kmFraGreenwich, 2) +
4         pow(startAttraktion.kmFraAekvator - slutAttraktion.kmFraAekvator, 2));
5 }

```

Funktionen "udregn_kanter" bruges til at udregne distancen mellem punkterne. Beregningen af distancerne sker vha. "beregne_dist", ved at sende en start og slut attraktion. Derefter bliver start- og slut attraktionen plus længden imellem dem lagt ind som en kant i kanter arrayet. Dette gøres i en for løkke i en anden for løkke, hvor der for hvert punkt blive udregnet distancen til de punkter der ikke allerede er blevet oprettet en kant for i en tidligere iteration af løkkerne. For hver kant der bliver oprettet, oprettes en ekstra kant som har samme længde, men har modsat start og slut attraktion. Så hvis det oprettes en kant fra x til y, vil der også blive oprettet en kant fra y til x med samme længde. Kanterne bliver tilføjet til kantarrayet, gennem den pointer der er inputparameter til funktionen, så kanterne kan blive tilgået fra resten af programmet.

```

1 void udregn_kanter(attraktion *attraktioner, kant *kanter)
2 {
3     int i;
4     int y;
5     int indexTilKanter = 0;
6     for (i = 0; i < ANTAL_ATTRAKTIONER; i++)
7     {
8         for (y = i; y < ANTAL_ATTRAKTIONER; y++)
9         {
10             if (strcmp(attraktioner[i].navn, attraktioner[y].navn) != 0)
11             {
12                 kant k, j;
13                 k.start = attraktioner[i];
14                 j.slut = k.start;
15                 k.slut = attraktioner[y];
16                 j.start = k.slut;
17                 k.laengde = beregn_dist(k.start, k.slut);
18                 j.laengde = k.laengde;
19                 kanter[indexTilKanter] = k;
20                 kanter[indexTilKanter+1] = j;
21                 indexTilKanter += 2;
22             }
23         }
24     }
25 }

```

Funktionen "findDist" gennemgår alle kanter der allerede er oprettet, og returnere distancen mellem to attraktioner, som er sendt med kaldet af funktionen, uden at skulle beregne den igen.

```

1 double findDist(attraktion start, attraktion slut, kant *kanter){
2     int i;
3     for (i = 0; i < ANTAL_KANTER; ++i)
4     {
5         if(strcmp(kanter[i].start.navn, start.navn) == 0 && strcmp(kanter[i].slut.navn, slut.navn) == 0){
6             return kanter[i].laengde;
7         }
8     }
9     printf("Kunne ikke finde passende kant\n"); exit(0);
10 }

```

Funktionen "attraktionErTilfoejet" bruges til at finde ud af, om en attraktioner allerede er tilføjet til ens liste over ekstra attraktioner, og returnere enten true eller false.

```

1 int attraktionErTilfoejet(attraktion *ekstraAttraktioner, int antalEksktraAttraktioner, attraktion
    attraktionAtTilfoeje){
2     int i;
3     for (i = 0; i < antalEksktraAttraktioner; ++i)
4     {
5         if(strcmp(ekstraAttraktioner[i].navn, attraktionAtTilfoeje.navn) == 0){
6             return 1;
7         }
8     }
9     return 0;

```

```
10 }
```

Funktionen "prikProdukt" bruges til at finde prikproduktet mellem to vektorer, hvilket også er det funktionen returnere.

```
1 double prikProdukt(vektor vektor1, vektor vektor2){
2     return (vektor1.x * vektor2.x) + (vektor1.y * vektor2.y);
3 }
```

Mangler at skrive om "findEkstraAttraktionerFirkant".

```
1 void findEkstraAttraktionerFirkant(attraktion startAttraktion, attraktion slutAttraktion, attraktion
   *valgteAttraktioner,
2 int *antalValgteAttraktioner, kant *kanter, attraktion attraktionAtTilfoeje, double maxDist,
3 attraktion *ekstraAttraktioner, int *antalEsktraAttraktioner){
4     int i, j;
5     double vektorLaengde;
6     vektor ruteVektor, ruteVinkelretVektor, ruteEnhedsVinkelretVektor, mainHjoerne, side1Vektor,
       side2Vektor, punktVektor;
7
8     vektorLaengde = findDist(startAttraktion, slutAttraktion, kanter);
9     ruteVektor.x = slutAttraktion.kmFraGreenwich - startAttraktion.kmFraGreenwich;
10    ruteVektor.y = slutAttraktion.kmFraAekvator - startAttraktion.kmFraAekvator;
11    ruteVinkelretVektor.x = -ruteVektor.y;
12    ruteVinkelretVektor.y = ruteVektor.x;
13    ruteEnhedsVinkelretVektor.x = ruteVinkelretVektor.x / vektorLaengde;
14    ruteEnhedsVinkelretVektor.y = ruteVinkelretVektor.y / vektorLaengde;
15    mainHjoerne.x = startAttraktion.kmFraGreenwich + ruteEnhedsVinkelretVektor.x * maxDist;
16    mainHjoerne.y = startAttraktion.kmFraAekvator + ruteEnhedsVinkelretVektor.y * maxDist;
17    side1Vektor.x = ruteVektor.x;
18    side1Vektor.y = ruteVektor.y;
19    side2Vektor.x = -2 * ruteEnhedsVinkelretVektor.x * maxDist;
20    side2Vektor.y = -2 * ruteEnhedsVinkelretVektor.y * maxDist;
21
22    punktVektor.x = attraktionAtTilfoeje.kmFraGreenwich - mainHjoerne.x;
23    punktVektor.y = attraktionAtTilfoeje.kmFraAekvator - mainHjoerne.y;
24    if(0 < prikProdukt(punktVektor, side1Vektor) && prikProdukt(punktVektor, side1Vektor) <
       prikProdukt(side1Vektor, side1Vektor) &&
25    0 < prikProdukt(punktVektor, side2Vektor) && prikProdukt(punktVektor, side2Vektor) <
       prikProdukt(side2Vektor, side2Vektor)){
26        ekstraAttraktioner[*antalEsktraAttraktioner] = attraktionAtTilfoeje;
27        *antalEsktraAttraktioner += 1;
28    }
29 }
```

For at foreslå ekstra attraktioner til den valgte rute, bruges funktionen "findEkstraAttraktioner", hvor dette vil udgøre den interessante rute. Dette gøres den ved først at bruge funktionen "findDist", til at finde ikke valgte attraktioner inden for en bestemt distance af de valgte attraktioner. Er en attraktion ikke inden for denne radius, bruges funktionen "findEkstraAttraktionerFirkant" for at finde ud af, om attraktionen ligger tæt på ruten mellem to attraktioner. De attraktioner der enten er inden for den bestemte distance af enten attraktionerne eller ruterne derimellem, tilføjes til et array der returneres fra funktionen.

```
1 void findEkstraAttraktioner(attraktion *ruteAttraktioner, attraktion *valgteAttraktioner, int
   *antalValgteAttraktioner,
2 kant *kanter, attraktion *ikkeValgteAttraktioner, double maxDist, attraktion *ekstraAttraktioner, int
   *antalEsktraAttraktioner){
3
4     int i, j, antalIkkeValgteAttraktioner = ANTAL_ATTRAKTIONER - *antalValgteAttraktioner;
5
6     for (i = 0; i < *antalValgteAttraktioner; ++i)
7     {
8         for (j = 0; j < antalIkkeValgteAttraktioner; ++j)
```

```

9          {
10
11              if(attraktionErTilfoejet(ekstraAttraktioner, *antalEksktraAttraktioner,
12                                     ikkeValgteAttraktioner[j])){
13              }else if(findDist(ruteAttraktioner[i], ikkeValgteAttraktioner[j], kanter) < maxDist
14                          ||
15                          findDist(ruteAttraktioner[i+1], ikkeValgteAttraktioner[j], kanter) < maxDist){
16                  ekstraAttraktioner[*antalEksktraAttraktioner] = ikkeValgteAttraktioner[j];
17                  *antalEksktraAttraktioner += 1;
18              }else{
19                  findEksktraAttraktionerFirkant(ruteAttraktioner[i], ruteAttraktioner[i+1],
20                                                  valgteAttraktioner, antalValgteAttraktioner,
21                                                  kanter, ikkeValgteAttraktioner[j], maxDist, ekstraAttraktioner,
22                                                  antalEksktraAttraktioner);
23          }
24      }
25  }
26  }

```

Funktionen "aendre_startsted", tager en rute som inputparameter, sammen med det startsted der ønskes. Funktionen laver derefter et nyt array, der har det nye startsted som første og sidste element, så ruten starter i startstedet, og vender tilbage dertil. Dette array er hvad funktionen returnere.

```

1 void aendre_startsted(attraktion *ruten, attraktion nytStartSted, int antalAttraktioner, attraktion
2     *outputRute)
3 {
4     int i = 0, startStedIndex = 0;
5
6     for (i = 0; i < antalAttraktioner; i++)
7     {
8         if (strcmp(ruten[i].navn, nytStartSted.navn) == 0)
9             startStedIndex = i;
10    }
11
12    for (i = 0; i < antalAttraktioner; i++)
13    {
14        if (startStedIndex == antalAttraktioner-1)
15        {
16            startStedIndex = 1;
17            outputRute[i] = ruten[0];
18        }
19        else
20        {
21            outputRute[i] = ruten[startStedIndex];
22            startStedIndex++;
23        }
24    }
25 }

```

4.3.3 Opsummering

Her er en liste med de overstående funktioner, med en kort forklaring om hvad hver funktions opgave i programmet er.

- InitialiserAttraktioner – Initialisere attraktioner fra en tekstfil
- valgafAttraktioner – Lader brugeren vælge attraktioner
- findNaboRute – Bestemmer den korteste rute mellem en række attraktioner, for en bestemt attraktion
- findKortesteNaboRute – Finder det optimale startsted

- `beregn_dist` – Beregner distancen mellem to attraktioner
- `udregn_kanter` – Bruger ”`beregn_dist`” til at oprette kanter mellem attraktioner
- `output_liste` – Udskriver den korteste rute til brugeren
- `findDist` – Returnere en allerede beregnet distance mellem to attraktioner
- `attraktionErTilfoejet` – Finder ud af om attraktionen allerede er tilføjet til ens liste over attraktioner
- `prikProdukt` – Udregner prikproduktet mellem to vektorer
- `findEkstraAttraktionerFirkant` – Finder ekstra attraktioner tæt på ruten mellem attraktionerne
- `findEkstraAttraktioner` – Forslår ekstra attraktioner til ruten
- `valgAfEkstraAttraktioner` – Lader brugeren vælge mellem hvilke ekstra attraktioner der ønskes besøgt
- `aendre_startsted` – Ændre rutens startsted

Diskussion 5

Konklusion 6

Perspektivering 7

Litteratur

- Dansk Statistik, 2008.** Dansk Statistik. *Turismen - Regionalt, nationalt og internationalt*. <http://www.dst.dk/pukora/epub/upload/11676/tur08.pdf>, 2008. Set d. 19/11-2014 - side 8.
- FindTheBestRoute.com, 2014.** FindTheBestRoute.com. *Route Optimization - Find the best route between multiple addresses*. <http://findthebestroute.com/RouteFinder.html>, 2014. Set d. 27/10-2014.
- for Erhverv & Udvikling, 2012.** Center for Erhverv & Udvikling. *Erhverv og turisme*. <http://www.faxe-kommune.dk/erhverv-turisme>, 2012. Set d. 20/11-2014.
- Google, 2014.** Google. *Google Maps JavaScript API v3*. <https://developers.google.com/maps/documentation/javascript/directions>, 2014. Set d. 27/10-2014.
- Harboe, 2011.** Thomas Harboe. *Metode og projektskrivning*. 2011.
- JMT, 2010.** Patrick JMT. *Graph Theory - An Introduction!* <https://www.youtube.com/watch?v=HmQR8Xy9DeM>, 2010. Set d. 09/12-2014.
- Kolpack, 2013.** James Kolpack. *Traveling Salesman Problem Visualization*. <https://www.youtube.com/watch?v=SC5CX8drAtU>, 2013. Set d. 09/12-2014.
- Mathispower4u, 2013a.** Mathispower4u. *Graph Theory: Dijkstra's Algorithm*. <https://www.youtube.com/watch?v=KvRwplnIoEM>, 2013. Set d. 09/12-2014.
- Mathispower4u, 2013b.** Mathispower4u. *Graph Theory: Nearest Neighbor Algorithm (NNA)*. <https://www.youtube.com/watch?v=zPgsNs0fxQ8>, 2013. Set d. 09/12-2014.
- Ritzau, 2014.** Ritzau. *Forbrugerrådet: Pengeløse butikker udelukker kunder*. <http://www.information.dk/telegram/496765>, 2014. Set d. 20/11-2014.
- Østergaard og Lubson, 2013.** Mads Zahle Østergaard og Paul Lubson. *Turismens økonomiske betydning i Danmark 2011*. http://www.visitdenmark.dk/sites/default/files/vdk_images/PDF-and-other-files/Analyser/2013/turismens_oekonomiske_betydning_i_danmark_2011_revideret_udgave.pdf, 2013. Set d. 20/11-2014.
- TDC, 2014.** TDC. *Data i udlandet*. http://kundeservice.tdc.dk/privat/publish.php?dogtag=p_help_mob_ud_pdk_du, 2014. Set d. 18/11-2014.
- VisitAalborg, 2014.** VisitAalborg. *Sammen er vi både attraktive og stærke!* <http://www.visitaalborg.dk/sites/default/files/asp/visitaalborg/uploads/partner2015/partner2015.pdf>, 2014. Set d. 20/11-2014.

VisitDanmark, 2013. VisitDanmark. *Fakta og tal om turismen i Danmark.*

<http://www.visitdenmark.dk/da/denmark/fakta-og-tal-om-turismen-i-danmark>, 2013.
Set d. 26/10-2014.

Vitcenda, 2011. Mary Vitcenda. *Why should visitors love your town? Count the ways.*

<http://www.extension.umn.edu/community/news/tourism-assets.html>, 2011. Set d.
20/11-2014.

Vodafone. Vodafone. *Turismen - Regionalt, nationalt og internationalt.*

[http://www.vodafone.co.uk/shop/pay-monthly/travelling-abroad/
vodafone-data-traveller/index.htm](http://www.vodafone.co.uk/shop/pay-monthly/travelling-abroad/vodafone-data-traveller/index.htm).

Youth Central. Youth Central. *Planning your trip.* [https://www.youthcentral.vic.gov.](https://www.youthcentral.vic.gov.au/travel-transport/travelling-overseas/planning-your-trip)

[au/travel-transport/travelling-overseas/planning-your-trip](https://www.youthcentral.vic.gov.au/travel-transport/travelling-overseas/planning-your-trip). Set d. 27/10-2014.

Rettelser

Spørgeskema A

I dette appendiks vil teorien til spørgeskemaer blive beskrevet, og rådataen fra respondenterne præsenteret. Teorien er skrevet ud fra pdf'en om dataindsamling, som kan ses under [Harboe, 2011].

A.1 Teori

Spørgeskemaer er en form for interview, hvor den kvantitative metode benyttes, som kan ske både online og offline. I dette projekt er den online løsning blevet benyttet, hvilket kaldes et internetinterview. Den offline metode kaldes for postale undersøgelser. Denne metode foregår ved at respondenterne får spørgeskemaet fysisk i papirformat. Fordelen ved internetinterview er den hurtige respons fra respondenterne og den gør resultatbehandlingen nemmere. [1]

Når et spørgeskema skal udføres er det vigtigt at have helt styr på formålet, altså hvad er succes kriterierne, hvilke problemstillinger skal der være svaret på, efter spørgeskemaet er fuldført? Den anden ting som er vigtig at få afklaret, er hvem målgruppen for undersøgelsen er, hvem er det der skal svare på disse spørgsmål? Med disse to ting i baghovedet skal et spørgeskema så udføres, så formålet bliver opfyldt så godt som muligt, uden at gøre spørgeskemaet forvirrende for målgruppen.

A.1.1 Spørgsmålene

Spørgeskemaets spørgsmål skal først og fremmest være dækkende, på den måde at den skal kunne dække alle problemstillingerne som er blevet udformet. Derudover skal disse spørgsmål ikke gå ud over de opstillede problemstillingerne. Et spørgsmål som ikke svarer på en del af problemstillingen, er ikke brugbar i den efterfølgende analyse. Spørgsmålene skal være lige til sagen, der er ingen grund til for mange omveje. Som hovedregel er korte spørgsmål bedre end lange, da disse er mere direkte og overskuelige.

A.1.2 Formulering

Det er vigtigt i forbindelse med formuleringen af spørgsmålene til spørgeskemaet, at alle spørgsmålene vil blive forstået ens. Der må altså ikke være nogen tvivl hos respondenterne. Det er i denne forbindelse vigtigt ikke at bruge vage formuleringer, dobbeltspørgsmål, indforstået/faglig jargon og lange ord. Spørgsmålene kan give forskellige meninger for forskellige respondenter hvis disse ikke bliver overholdt, og dermed kan svarene ikke bruges til ret meget analytisk. Til sidst er ledende spørgsmål også farlige, da spørgeskemaundersøgelser som oftest har som mål at give objektive ikke forvrængede resultater.

A.1.3 Rækkefølge

Når rækkefølgen skal overvejes er der 4 ting der skal lægges særlig meget vægt på: -Motivationen hos respondenterne Respondenten vil efterhånden miste koncentrationen og motivationen til at besvare spørgeskemaet jo længere han eller hun kommer. Dette kan undgås ved først og fremmest kun at stille de nødvendige spørgsmål så spørgeskemaet bliver så kort som muligt. Derudover kan det være en fordel at stille de nemme spørgsmål i starten af spørgeskemaet så respondenterne får besvaret en masse spørgsmål i en fart, og derved kommer godt i gang. Det kan dog også være en god ide i nogle tilfælde at komme til sagen med det samme. -Konteksten spørgsmålet er indenfor I nogle tilfælde kræves der information fra et tidligere spørgsmål, før der kan svares på et andet. Derfor er det vigtigt at sørge for at rækkefølgen er således at respondenterne har fået stillet de krævede spørgsmål før det spørgsmål hvor respondenterne skal bruge informationen. -Den emnemæssige sammenhæng I rækkefølgen på spørgeskemaet skal der undgås alt for store emnemæssige brud, der skal være et naturligt flow mellem spørgsmålene. -Tragtmodellen Det er til stor fordel at stille de generelle spørgsmål før de specifikke, så der opstår en form for tragtmodel ned gennem spørgeskemaet.

A.2 Rådata

Antal respondenter: 60

Hvad er vigtigt for dig på din storbyferie? Sæt gerne flere krydser

Svarvalg	Besvarelser
Opleve kulturen	68.33% - 41
Se byens seværdigheder	90% - 54
Shopping	50% - 30
Fest/Bytur	26.67% - 16
Museumbesøg	11.67% - 7
Maden	63.33% - 38
Teater/Musik	18.33% - 11

Kommentarer: Sportsbegivenheder

I hvilken grad bliver din storbyferie planlagt?

Svarvalg	Besvarelser
Alt er planlagt til punkt og prikke inden ferie	0% - 0
Nogle ting er planlagt inden ferien	70% - 42
Planlægger dag for dag på ferien	28.33% - 17
Planlægger ikke	1.67% 1.67 - 1

Hvilke hjælpemidler bruger du til at planlægge din storbyferie?

- Tripadvisor
- Google
- Internettet
- Guide

- Google Maps
- Yelp
- Turen går til..
- Venner/familie som har været på stedet
- Rejsebureauer
- Bøger
- Lonely planet
- fdm-travel.dk
- booking.com
- hotels.com
- Momondo
- Ansrejser
- Expedia
- Hertz biludlejning
- Scout
- Lokale
- Kort
- Politikens de røde
- Top 10 vigtigste ting at se

Et program/applikation, som hjælper mig med at finde den hurtigste og/eller mest interessante vej igennem byen, via mine valgte "must see" destinationer, ville være noget jeg kunne bruge?

Hvilke redskaber bruger du til at finde rundt når du er på storbyferie? Vælg gerne flere

Svarvalg	Besvarelser
De lokale	46.67% - 28
Diverse kort/Brochure	81.67% - 49
GPS/Elektronisk kort	55% - 33
Taxa	10% - 6
Guider	18.33% - 11

Har du nogensinde haft problemer med at finde vej på din storbyferie?

Svarvalg	Besvarelser
Ja	68.33% - 41
Nej	31.67% - 19

Når du skal fra en aktivitet til en anden på din storbyferie, vil du helst tage den hurtigste rute eller en langsommere men mere interessant rute?

Svarvalg	Besvarelser
Den hurtige rute	20% - 12
Den interessante rute	80% - 48

Et program/applikation, som hjælper mig med at finde den hurtigste og/eller mest interessante vej igennem byen, via brugerens valgte "must see" destinationer, ville være noget jeg kunne bruge?

Meget uenig	Uenig	Enig	Meget enig
1.67% - 1	6.67% - 4	45% - 27	46.67% - 28

Interview B

B.1 Teori

Der er mange former for interviews og disse kan udføres på forskellige måder, men typisk når der snakkes om interviews, bliver de delt ind i tre forskellige former, enkeltinterview, spørgeskemaer og telefoniske interviews. Dette afsnit er udarbejdet ud fra PDF'en om dataindsamling, der kan ses i litteraturlisten, under [Harboe, 2011].

B.1.1 Enkeltinterview

Det enkelte interview forgår på følgende måde: Både interviewer og respondent mødes ansigt til ansigt. Fordelene herpå er tydelige, at give respondenterne mulighed for at besvare private og intime spørgsmål, som en almen respondent ikke er tryk ved at tale om foran andre. Dette mindsker også chancen for at spørgsmålene bliver misforstået, samtidig med at svarene kan diskuteres på et højere plan, end ved et interview over telefonen eller ved et spørgeskema, da snakkes med mere end bare ord, nemlig kropssprog. Hvis interviewer har en god situationsfornemmelse kan et vellykket interview, forventes. En udvidelse af enkeltinterviewet kan der snakke om gruppeinterview. Metoden bruges hvis der er pres på tid og ressourcer. Metoden er den samme udover den forskel at der er flere respondenter. Der opfordres ikke til dialog mellem respondenterne. Et modsvar til denne metode er fokusgrupper. Denne metode opfordrer netop til dialog mellem respondenterne, men emnet her er temmelig afgrænset. Denne metode bliver brugt til at sammenligne skabelsen af holdninger i sociale miljøer og hvilke argumenter der bliver taget i brug.

B.1.2 Telefoninterview

Det telefoniske interview er lidt en sammenblanding af de to ovennævnte interview former. Telefoninterviewet foregår ved at en eller flere interviewere sidder bag røret og stiller en række spørgsmål, som på forhånd er fastlagte. Den væsentlige forskel på telefoninterviewet som er et kvalitativt interview og spørgeskemaet som er et kvantitativt interview, er at interviewer kan uddybe deres spørgsmål på et højere plan, end et spørgeskema vil kunne. Måden hvorpå denne form for interview foregår er ved at scanne spørgeguiden ind i et program, hvorefter dette vil blive sendt til respondenterne og unødvendige spørgsmål undgås. Her har respondenterne så mulighed for at skrive sine egne svar ind, hvilket mindsker fejl ved fx transskription. Til sidst har interviewer mulighed for at gå i detaljer med hvert spørgsmål sammen med respondenterne.[?]

B.1.3 Spørgeteknikker og metoder til interview

Når der snakkes om videnskabelige spørgeteknikker er det vigtigt at kende forskellene på dette og dagligdagssproget, som normalt bliver snakket. Der er de standardiserede spørgeteknikker, hvilket er hvor spørgsmålene og rækkefølgen på disse, omhyggeligt er blevet arbejdet med, og deres rækkefølge, er valgt på forhånd for interviewet. Denne metode er nyttig at tage i brug, hvis en interviewer kender problemstillingen. Her får intervieweren svar på sine spørgsmål med så lidt spildt information som muligt. Dette udføres typisk med spørgeskemaer. Nogle forskere mener, at et standardiseret interview også har det element, at forholdene og endda tiden for interviewet er ens for alle respondenter. Alle andre former for interview er indenfor kategorien ikke-standardiseret interview. Derudover er der de strukturerede interviews. Dette forgår lidt på samme måde, som de standardiserede interviews. Den væsentligste forskel herpå, er at spørgsmålene ikke er fastlagte, så det kun er spørgeguiden, der er fastsat. Dette skaber større mulighed for en kvalitativ interviewform, hvor intervieweren kan følge op på emner der kommer, som intervieweren ikke havde regnet med. Som et modsvar på denne form for interview, findes det ikke-strukturerede interview. Denne metode har hverken fastlagte spørgsmål, eller en fastlagt spørgeguide/rækkefølge på spørgsmål. Ved brug af denne metode kan intervieweren frit følge et givent emne ud fra respondentens svar, derfor kaldenavnet "det fleksible interview".

B.1.4 Lukkede og åbne spørgsmål

Lukkede spørgsmål bruges typisk ved kvantitative spørgeteknikker såsom et spørgeskema. Altså teknikker som gør at responsen let kan sammenlignes og analyseres. Denne metode af spørgsmål falder altså ind under kategorien standardiseret spørgsmål, da respondenterne hverken kan ændre på rækkefølgen af spørgsmålene eller gå ind og uddybe sine svar. Til hvert et træk, er der et modtræk. De åbne spørgsmål, som bliver benyttet i de kvalitative aspekter indenfor spørgeskemaer, altså den mulighed at respondenterne kan uddybe nogle svar, hvis intervieweren føler det er nødvendigt og stille sådan en plads til rådighed i spørgeskemaet. [Harboe, 2011]

B.1.5 Passive teknikker

Denne teknik går ud på at stille et spørgsmål, lade respondenterne svare, hvorefter intervieweren kommer ind med nogle spørgende kommentar. Ved brug af denne teknik, mindskes interviewernes bestemmelse i retningen af interviewet, og respondenterne kan komme med mere information, om et givent emne, og endda indbringe egne meninger og holdninger, hvis dette er vigtigt ift. emnet. [Harboe, 2011]

B.1.6 Aktiv spørgeteknik

Hvis intervieweren kommer ud for, at respondenterne er meget sky og tilbageholden med information, kan intervieweren manipulere ham/hende til at tro at den information de kommer ud med, er en lille del af en manglende kæde. Fx kan intervieweren nævne en given situation, hvorefter intervieweren spørger ind til det manglende led, altså den information intervieweren mangler. På denne måde ligner respondentens svar "bare" et lille manglende led i noget intervieweren allerede ved. [Harboe, 2011]