
E-VALG

E-valg i fremtiden

P2-projekt

af Datalogi grp. A317



Aalborg Universitet
Institut for Datalogi
P2-projekt - 2. semester
Hovedvejleder: Jane Billestrup
21. maj 2013

Institut for Datalogi

Aalborg Universitet

2. Semester

TITEL:

E-VALG -

E-valg i fremtiden

PROJEKTPERIODE:

P2, 5. feb - 22. maj 2013

PROJEKT GRUPPE:

A317

GRUPPEMEDLEMMER:

Anders Hermansen

Isabella Kaufmann

Joachim Klockervoll

Lynge Poulsgaard

Mike Pedersen

Samuel Nygaard Pedersen

Søren Moss Nielsen

VEJLEDER:

Jane Billestrup

ANTAL KOPIER: 9

SAMLET SIDEANTAL: 96

NORMALSIDEANTAL: 58

ANTAL CD'ER VEDLAGT: 1

SYNOPSIS:

Dette projekt omhandler elektronisk valg (forkortet e-valg). Projektet er udarbejdet med fokus på, at forbedre den nuværende valgproces i Danmark. Projektet indeholder et løsningsforslag til et e-valg baseret på en grundig problembaseret analyse og datalogiske overvejelser. Løsningsforslaget er udarbejdet som en softwareapplikation kaldet EVS. I rapporten vurderes det, at e-valgsteknologien ikke klar til at blive implementeret i det danske demokratis valg-system. Selvom teknologien ikke har klare fremtidsudsigter, så er teknologiens aktualitetsværdi beskrevet og demonstreret med projektets e-valgsløsning.

Aalborg, 21. maj 2013

Anders Hermansen
studienr.: 20124269
mail: aherm12@student.aau.dk

Isabella Kaufmann
studienr.: 20125900
mail: ikaufm12@student.aau.dk

Joachim Klockervoll
studienr.: 20124277
mail: jklokk12@student.aau.dk

Lynge Poulsen
studienr.: 20124272
mail: lkpo12@student.aau.dk

Mike Pedersen
studienr.: 20124283
mail: mipede12@student.aau.dk

Samuel Nygaard Pedersen
studienr.: 20124278
mail: snpe12@student.aau.dk

Søren Moss Nielsen
studienr.: 20124275
mail: smni12@student.aau.dk

Forord

Denne rapport er skrevet som et P2 projekt i Datalogi på Aalborg Universitet, og målgruppen er derved også datalogi-studerende på første og andet semester. I denne rapport er afsnittene inddelt på en sådan måde, at de, i den valgte rækkefølge, er nemmere at forstå og bygger op omkring hinanden, end hvis de læses individuelt. Indledningen og konklusionen kan desuden læses for sig selv, for at få et hurtigt overblik over projektet og de overvejelser, vi er kommet frem til i udarbejdelsen. I rapporten er der desuden brugt fodnoter og kildehenvisninger. Kildehenvisninger er markeret ved brug af tal, som f.eks. “[1]”, der henviser til kilderne i listen bagerst i rapporten. Fodnoter er markeret med et tal i superscript, som f.eks. “udtryk¹”, som henviser til fodnoten nederst på den side, hvor den er benyttet. Fodnoter er kun brugt i forbindelse med ordforklaringer. Denne rapport er udarbejdet af Anders Hermansen, Isabella Kaufmann, Joachim Klockervoll, Lynge Kærlund Poulsgaard, Mike Pedersen, Samuel Nygaard og Søren Moss Nielsen, under vejledning af Jane Billestrup ved det Tekniske-Naturvidenskabelige Fakultet, Aalborg Universitet.

Indhold

1	Indledning	9
I	Problemanalyse	10
2	Valg	11
2.1	Demokrati	11
2.2	Valg i praksis	11
2.2.1	Fremmøde	11
2.2.2	Brevstemme	12
2.2.3	Effektivitet	12
2.3	Lovgivning	13
2.4	Valgret	14
2.5	Opsamling	15
3	E-valg	16
3.1	Forskellige former for E-valg	16
3.2	Fordele og ulemper ved traditionelt valg og e-valg	17
3.2.1	Traditionelt valg	17
3.2.2	E-valg	18
3.3	Eksisterende løsninger	19
3.4	Erfaring med e-valg	20
3.4.1	Positive erfaringer med e-valg	21
3.4.2	Negative erfaringer med e-valg	21
3.5	Opsamling	22
4	Interessenter	23
4.1	Politikerne	23
4.2	Vælgerne	24
4.3	Handicappede	24
4.4	Forventninger til e-valg	25
4.5	Opsamling	26
5	Problembeskrivelse	27
5.1	Afgrænsning	27
5.2	Problemformulering	28

II	Løsningsforslag	29
6	Introduktion til Løsning	30
6.1	Kravspecifikation	30
6.2	Introduktion til <i>EVS E-Voting System</i>	31
6.2.1	Praktisk overblik	31
6.2.2	Teknisk overblik	32
7	Database	33
7.1	Databasesystemer	33
7.1.1	Valg af databasesystem	35
8	Datakommunikation	36
8.1	Fysik datadeling	36
8.2	Lokalt netværk	36
8.2.1	HTTP	37
8.2.2	HTTPS	37
8.2.3	Opsamling	37
9	Grafisk brugergrænseflade	38
9.1	Brugervenlighed	38
9.2	De ti Heuristikker	39
9.3	Implementering af brugergrænseflade	41
9.4	Opsamling	46
10	Programbeskrivelse	47
10.1	Programstruktur	47
10.1.1	Kandidater og lister	47
10.1.2	Stemmeafgivning	48
10.1.3	Vælgeridentifikation	49
10.1.4	Databaseimplementering	49
10.1.5	Netværskommunikation	53
10.2	Systemgennemgang	57
10.2.1	Inden valget	57
10.2.2	Under valget	58
10.2.3	Efter valget	59
10.2.4	Valgafbrydelser	60
11	Test	62
11.1	Netværkstest	62
11.2	Databasetest - Stemmer	64
11.3	Databasetest - Vælgere	67
11.4	Black-box test	69
11.5	Opsamling	69

III	Implikationer	70
12	Diskussion	71
12.1	Afgivning og opbevaring af stemmer	71
12.1.1	Anonymitet og identifikation	71
12.1.2	Kravopfyldelse	73
12.2	Netværkstyper og kommunikation	74
12.2.1	Kravopfyldelse	74
12.3	Samling og optælling af stemmer	75
12.3.1	Kravopfyldelse	75
12.4	Brugerflade	75
12.4.1	Kravopfyldelse	76
13	Konklusion	77
14	Perspektivering	81
14.1	Videreudvikling	81
14.2	Udvidede brugsmuligheder	82
15	Bibliografi	83
15.1	Bøger	83
15.2	Artikler og rapporter	83
15.3	Websider	84
IV	Bilag	87
16	Brugervejledning	88
16.1	Opsætnings- og fejlbehandlings-guide	88
16.2	Brugervejledning til vælgeren	94

Kapitel 1

Indledning

Demokrati er væsentligt for den moderne verden og menneskets eksistens deri. Demokrati samler befolkningen, fordi magten i folkestyret fordeles ved folkevalg. I Danmark deltager næsten 88 % af den stemmeberettigede befolkning til folketingsvalget, og bidrager dermed til, at opretholde folkestyret i sin nuværende form [10]. Men folketingsvalget er ikke den eneste type valg i Danmarks folkestyre. Der er også folkeafstemninger, kommunal- og regionsvalg samt valg til Europa-Parlamentet.

Et valg i sin nuværende form (med valgkort, stemmesedler og optælling) er en ressourcekrævende og bekostelig procedure, der er nødvendig for folkestyret. For at et valg er pålideligt, skal vælgerens stemmefrihed og stemmehemmelighed opretholdes, og der skal være klare retningslinjer for valgsystemet. I Danmark er retningslinjerne beskrevet i grundloven og valgloven, men der findes ingen undersøgelser eller dokumentation for, at valget i Danmark er pålideligt og dermed demokratisk, men der er heller ikke blevet opdaget noget, som gør valget upålideligt [11].

Dette projekt har til hensigt, at undersøge mulighederne for, at forbedre det nuværende valgsystem med datalogiske midler. Dette gøres ved, at undersøge mulighederne for en implementering af et e-valgssystem i Danmark. Hertil er formålet med projektet, at udvikle et forslag til et e-valgssystem. Forslaget vil indeholde en undersøgende problemanalyse og en program-prototype, som er baseret på viden fra den forudgående problemanalyse. Projektet vil også indeholde de datalogiske overvejelser, der er gjort i forhold til udarbejdelsen af program-prototypen.

Problemanalysen vil belyse områder omkring lovgivningen og retningslinjerne for valget i Danmark. Problemanalysen vil dertil indeholde en analyse af forudgående e-valgssystemer og interessenter, der vil blive påvirket af denne implementering. I implementeringsafsnittet bliver der beskrevet, hvilke datalogiske overvejelser, der er gjort forud for programmeringen af program-prototypen, blandt andet overvejelser omkring databaser, netværkskommunikation og grafisk brugerflade.

Formålet med dette projekt er overordnet, at udarbejde et elektronisk valgsystem til valg i Danmark og undersøge mulighederne for implementering af dette i dansk kontekst. Derfor ender projektet ud med udarbejdelsen af et sådant system, som testes, diskuteres, konkluderes på og perspektiveres ud fra. I det følgende afsnit påbegyndes problemanalysen, som starter med de demokratiske grundprincipper i forbindelse med valg i Danmark.

Del I

Problemanalyse

Kapitel 2

Valg

Enevældet i Danmark blev i 1849 erstattet af grundloven, og dette medførte de første skridt mod et reelt folkestyre og den form for demokrati, vi kender i dag [12]. Som en del af det moderne demokratiet, skal der vælges kandidater til folketinget. Det foregår igennem en stemmeafgivelse, som følger en række regler og love. I dette kapitel vil omstændighederne, traditionerne og lovgivningen blive beskrevet.

2.1 Demokrati

Demokrati eller folkeligt selvstyre er væsentligt, når det gælder fordelingen af magten i samfundet. Det danske Demokrati er opbygget om grundloven, som sikrer den individuelle rettigheder, så som politisk frihed, stemmefrihed, ytringsfrihed og valgret [13].

Stemmeafgivelsen ved et valg er en forudsætning for demokratiet. Valgsystemet er kompliceret og er derfor defineret i valgloven.

2.2 Valg i praksis

Når der skal stemmes til en folkeafstemning, et folketingsvalg eller et kommunal- og regionalvalg, foregår det på en af to måder, hhv. via brevstemme eller den mest brugte; hvor der mødes op på et valgsted.

2.2.1 Fremmøde

Forud for valget oprettes der valgsteder i hele landet, og alle stemmeberettigede danske statsborgere over 18 år med bopæl i Danmark får tilsendt et valgkort. Hvilket valgsted hver enkelt borger skal møde op på, er angivet på valgkortet. Valgkortet indleveres på valgstedet til en valglistefører, som afkrydser borgeren fra valglisten og dernæst udleverer en stemmeseddel. I tilfælde af tvivl om borgerens identitet kan legitimation forlanges. Der er ingen krav om, at denne dokumentation skal være billedlegitimation. Stemmesedlen medbringes så til en stemmeboks, for at sikre at afstemningen bliver hemmelig. I stemmeboksen afkrydser vælgeren

sit ønske i enerum. Stemmesedlen lægges så i en valgurne, som tømmes når valgstedet lukker, hvorefter alle stemmerne tælles op. Stemmesedlen kan dømmes ugyldig, hvis der er tvivl om, hvor krydset er sat. For at sikre, at ingen stemmer kan føres tilbage til enkelte personer, kan mærker, og andre måder at skelne en seddel fra andre, heller ikke accepteres, og sådanne stemmer bliver også gjort ugyldige [14] [15].

2.2.2 Brevstemme

Under bestemte omstændigheder kan en stemme afgives, uden at der mødes op på valgstedet. I tilfælde af sygdom, fængsling eller udenlandsophold, kan stemmen afgives som brevstemme. Brevstemmer bliver talt op forud for valget, og brevstemmerens navn bliver fjernet fra valglisten, for at der ikke kan stemmes igen ved fysisk fremmøde. Brevstemmen udfyldes i hemmelighed og forsegles da i en farvet konvolut. Dernæst udfyldes en følgeseddel, som underskrives af vælgeren og stemmemodtageren. Stemmемodtageren er den embedsmand, som modtager brevstemmen og får den sendt til valgstedet på valgdagen. Efter følgesedlen er udfyldt, kan yderkonvolutten forsegles, indeholdende farvet konvolut med stemme og følgeseddel med underskrifter fra vælgeren og stemmemodtageren [16] [17].

2.2.3 Effektivitet

Både valg ved fremmøde og brevstemme er forholdsvis ressourcekrævende. Udfordringen med effektivisering er lige nu meget aktuel i samfundet, og der er derved relevant at undersøge om der kan spares ressourcer, når der stemmes.

Det tager tid at afgive sin stemme. Set fra den enkelte vælgers perspektiv er det en hel række af ting, vælgeren aktivt skal gøre på valgdagen. På en hverdag, gerne på vej hjem fra arbejde, hvor alle andre også skal stemme, skal valgkortet medbringes. Kortet udveksles for en stemmeseddel, der dernæst udfyldes i en stemmeboks. Som en del af denne proces sker der en identifikation af vælgeren separat fra vælgerens stemmeafgivelse, for at sikre, at valget foregår hemmeligt [14].

Optællingen af stemmer på valgdagen sker af to omgange. Begge gange bliver dette gjort manuelt. Den dobbelte optælling skal sikre, at muligheden for en fejloptælling er minimal. Der skal udover optællingen af stemmer også udføres udregning af mandater. Mulighederne for at effektivisere disse processer kunne ske ved en digitalisering af stemmerne. Grundet det danske valgssystem, kan den endelige uddelegering af mandater heller ikke ske korrekt før alle stemmer er optalt [18]. Optælling og udregning af over 4.000.000 stemmer på maskiner kan give de endelige resultater ganske kort efter valgstederne lukker.

2.3 Lovgivning

For at E-valg skal kunne implementeres i Danmark, må det naturligvis ikke være i strid med grundloven. Selvom grundloven kan ændres ved folkevalg, er det en besværlig og bekostelig procedure, som langt fra er nødvendig for at muliggøre E-valg. I grundloven omhandler paragraf 31 valg til Folketinget og lyder som følger:

Stk. 1. Folketingets medlemmer vælges ved almindelige, direkte og hemmelige valg.

Stk. 2. De nærmere regler for valgrettens udøvelse gives ved valgloven, der til sikring af en ligelig repræsentation af de forskellige anskuelser blandt vælgerne fastsætter valgmåden, herunder hvorvidt forholdstalsvalgsmåden skal føres igennem i eller uden forbindelse med valg i enkeltmandskredse.

Stk. 3. Ved den stedlige mandatfordeling skal der tages hensyn til indbyggertal, vælgertal og befolkningstæthed.

Stk. 4. Ved valgloven gives nærmere regler vedrørende valg af stedfortrædere og disses indtræden i folketinget samt angående fremgangsmåden i tilfælde, hvor omvalg måtte blive nødvendigt.

Stk. 5. Særlige regler om Grønlands repræsentation i folketinget kan gives ved lov.

(§31 Grundloven [19])

Mange af disse stykker af paragraf 31 refererer til valgloven, en adskilt lov fra grundloven, der er blevet ændret oftere gennem tiden [20]. I forhold til E-valg er Stk. 1. den mest relevante, da det er dette, der fastsætter de mest basale rammer for, hvordan et valg skal foregå. Det er derfor et krav, at et E-valg system skal gøre det muligt, at have almindelige, direkte og hemmelige valg.

Grundloven bestemmer dog intet i forhold til andre valg, så som kommunale valg eller valg til EU-Parlamentet, men der findes dog love, der stiller lignende krav til disse valg (se hhv. [21] og [22]). I princippet betyder dette, at Folketinget kan lave store ændringer på disse valg uden en folkeafstemning, men i praksis ville dette sandsynligvis ikke være i folkets interesse. Hvis et E-valgs system skal indføres, skal det derfor følge grundloven, både for valg til Folketinget og andre valg.

Hvis et E-valgs system er i overensstemmelse med grundloven, vil det stadig kræve en række ændringer i den nuværende lovgivning omkring valgprocessen, for at muliggøre E-valg ([22] [20] [21]). Lovændringer i valgloven kan dog udføres af Folketinget uden en folkeafstemning. Ændringer i valgloven er en nødvendighed, da denne er meget detaljeret i sin beskrivelse af, hvordan et valg skal forløbe. Blandt andet står der specifikt, at der skal bruges papir [20].

2.4 Valgret

Som udgangspunkt kræver demokrati, at borgerne har indflydelse på, hvem de siddende medlemmer i folkestyret er. Dette opnås oftest ved at holde valg, enten i form af kommunale valg eller folketingsvalg. Hvordan disse valgsystemer fungerer, varierer dog fra land til land.

For at vurdere valgsystemer, er det nødvendigt at undersøge, hvilke kriterier der er nødvendige for et effektivt valg. Der tages her udgangspunkt i Danmark, men mange andre vestlige demokratier følger lignende principper. De lovmæssige kriterier, som dette projekt lægger sig op af, på baggrund af grundloven er:

Almindelighed

I et demokrati er befolkningen repræsenteret i regeringen. For at kunne repræsentere hele befolkningen, er det nødvendigt, at alle, over 18 med dansk statsborgerskab, har stemmeret. Derfor er det vigtigt, at have almindelig stemmeret, hvor der ikke diskrimineres på baggrund af race, køn, tro, rigdom, osv.

Lighed

Et valg skal være lige. Dvs. at alle stemmeberettigede borgere har den samme vægtning i valget. En vælger kan ikke opnå en større indflydelse på grund af hans rigdom, sociale klasse, eller lignende. Ændring af ligheden i et valg er, i en demokratisk sammenhæng, valgsvindel.

Direkte

I et direkte valg stemmes der på en enkelt person eller et parti. Her har vælgeren en direkte indflydelse, i modsætning til et indirekte valg, hvor der stemmes på en repræsentant, som derefter stemmer på andres vegne. Et direkte valg sikrer i større grad, at vælgerens holdninger bliver repræsenteret.

Stemmehemmelighed

For at beskytte vælgeren fra trusler, bestikkelse, eller andre indflydelser, er det nødvendigt at vælgerens stemme er anonym. Med stemmehemmelighed er en vælger mere beskyttet imod trusler, da der ikke er nogen måde, at vide, hvem en person har stemt på. Stemmehemmeligheden bruges også til, at beskytte imod opkøb af stemmer, ved ikke at gøre det muligt, at kvittere for en stemme. På denne måde er det heller ikke muligt for vælgeren selv, at bevise hvem vælgeren har stemt på.

Pålidelighed

Et valgsystem skal kunne tælle stemmerne korrekt op og skal dermed være modstandsdygtig overfor manipulation og fejl.

Pålideligheden og stemmehemmelighed er ofte i konflikt med hinanden i den forstand, at der ofte kan forøge pålideligheden, men samtidigt tabe stemmehemmeligheden. Eksempelvis kunne et valgsystem med offentlige stemmer (altså ingen stemmehemmelighed), gøre det meget tydeligt om stemmerne er blevet optalt korrekt. Ligeledes kunne et løbenummer på stemmesedlerne gøre det betydeligt sværere at forfalske stemmer uden at det bliver opdaget. Disse forslag er dog på bekostning af stemmehemmeligheden,

og det er derfor vigtigt, at overveje om den øgede pålidelighed er nedsættelsen af stemmesikkerheden værd.

2.5 Opsamling

Det nuværende valgsystem udgør en velkendt del af det danske demokrati. Hvis E-valg skal indføres i Danmark, skal det kunne opfylde kravene lige så godt eller bedre end det nuværende system. Det skal både tage højde for at folk møder op på valgstedet, og de særtilfælde, hvor vælgerne ikke kan komme frem til valgstedet. Det skal kunne optælle de afgivne stemmer mindst lige så effektivt, og kunne kontrolsikre, at optællingen er korrekt. Den nuværende valglov skal dog ændres, for at muliggøre e-valg, men grundloven behøver ikke ændres, så længe at løsningsforslaget overholder grundlovens krav om det “almindelige, direkte og hemmelige valg”. Der vil derfor i næste kapitel blive undersøgt hvilke former for E-valg, der findes, og hvordan de lever op til kravene om valg.

Kapitel 3

E-valg

Et e-valg, eller et elektronisk valg, er et valg, hvor nogle processer er digitaliseret, og det kan dermed automatisere og effektivisere nogle processer i valgsystemet. I følgende kapitel vil emnet e-valg blive yderligere defineret. Der vil desuden blive set på de forskellige måder, at implementere e-valg på. Dette efterfølges af en analyse og diskussion af det nuværende valg kontra et muligt elektronisk valg. Til sidst vil der blive kigget på allerede eksisterende løsninger samt andre landes erfaringer med e-valg.

Det østrigske e-valgskonsulentfirma E-Voting.CC definere e-valg som bredere brug af elektroniske midler i en af følgende tre processer: [23]

1. Identifikation af vælgerne
2. Stemmeafgivelse
3. Stemmeoptælling

Disse tre funktioner er kerneelementer i ethvert valg, og det er derfor vigtigt, at undersøge forskellene i disse områder mellem E-valg og traditionelt valg.

3.1 Forskellige former for E-valg

E-valg findes i en række forskellige afskygninger. De to overordnede former er “nærvalg” og “fjernvalg”. Ved nærvalg er stemmeberettigede til stede på valgstedet, hvorimod fjernvalg foregår udefra, oftest fra stemmeberettigedes eget hjem. Den form, som regeringen i Danmark har foreslået at indføre, omhandler brugen af elektroniske afstemningsmaskiner på valgstedet [3]. Valget, som foreslås, vil altså skulle afholdes, nogenlunde ligesom det gøres nu. Forskellen vil blot være, at stemmeafgivelsen vil blive gjort via en afstemningsmaskine frem for, at aflevere en stemmeseddel i en stemmeurne. Denne type valg vil i denne rapport gå under betegnelsen “nærvalg”.

Der findes desuden forskellige typer af nærvalg. Nærvalg kan foretages direkte på en stemmemaskine, hvor vælgeren afgiver sin stemme på valgmaskinen, enten via en trykfølsom skærm eller en anden form for inputenhed. Med denne metode kan brug af papir til afstemningen potentielt afskaffes. Det er dog ikke altid hensigten, og der findes også varianter, hvor en

stemmeseddel eller kvittering for afstemningen printes ud, hvorefter den lægges i en stemmeurne til eventuel manuel optælling som normalt. Der findes dog også typer af valg, hvor papiret stadig er en central del af afstemningen, men hvor det scannes ind på en maskine, så afstemningen digitaliseres til optælling. Stemmesedlen bliver herefter lagt i en stemmeurne, som ved den nuværende valgmetode. Ved brug af de sidstnævnte løsninger er det muligt, at tælle stemmerne på traditionel vis, hvis der skulle være tvivl om troværdigheden af den elektroniske optælling [23].

En anden form for e-valg er det såkaldte “fjernvalg”, hvor afstemningen ikke foregår på et valgsted, men kan foretages udefra. Denne metode kan sammenlignes med den nuværende metode, hvor afgivning af stemmen sker via brevstemme. Der er dog nogle krav, der skal opfyldes, for at en person må brevstemme, og der vil også være en valgtilforordnet til stede under stemmeafgivelsen [24]. Et “fjernt” e-valg foregår dog i stedet via internettet, hvor stemmeberettigede, via sin computer, er i stand til, at afgive sin stemme [23]. Denne type valg foregår uden overvågning i et ukontrolleret miljø, og det kan derfor ikke sikres, at stemmeanonymitet og stemmehemmelighed bliver overholdt, da der ikke kan holdes tilsyn med dette. Derfor opfylder et “fjernt” e-valg, i sin nuværende form, ikke umiddelbart kravene til det valg, som er beskrevet i grundloven § 31 Stk. 1 [19]. Det vil derfor kræve meget arbejde, at udforme en løsning baseret på “fjernvalg”, der vil være lovlig at implementere. Denne form for valg vil fremover i rapporten blot blive betegnet som “fjernvalg”.

3.2 Fordele og ulemper ved traditionelt valg og e-valg

I diskussionen om e-valg er det vigtigt at gå ind og analysere på de forskellige former for valg, og belyse deres individuelle styrker og svagheder. I den følgende sektion vil der komme en kort analyse af henholdsvis det traditionelle valg der allerede benyttes i Danmark, samt det elektroniske valg der måske skal indføres.

3.2.1 Traditionelt valg

Det eksisterende valgsystems troværdighed bygger på en række af ting; her i blandt gennemsigtighed, folkelig kontrol, og tradition. Med gennemsigtighed menes der, at langt de fleste danskere har en forståelse for, hvordan valgsystemet og optællingen forløber i praksis. Det er udformet på en sådan måde, at enhver kan se, hvorfor det fungerer, og hvordan det opfylder de nødvendige krav til et demokratisk valg. Folkelig kontrol er tilstede, da alle har en mulighed for selv, at være med i optællingen og dermed kontrollere gyldigheden af det endelige resultat. Dette bidrager til følelsen af tryk og tillid i forhold til valgene og de endelige resultater [4].

Det eksisterende valgsystem er forblevet stort set det samme i det seneste århundrede. Dette betyder hovedsageligt to ting i forhold til troværdigheden. For det første er det et gennemtestet og stabilt valgsystem, som har fungeret tilfredsstillende for befolkningen. For det andet betyder det, at der for mange vælgere er en eller flere traditioner forbundet med de

danske valg. For nogle er det vigtigt, at gå ned og stemme i samlet flok ved et fælles valgsted og herefter se optællingen af stemmer på TV, efterhånden som tallene kommer ind. Disse traditioner gør valgdagen mere speciel og bidrager til følelsen af, at være en aktiv deltager i et fungerende demokrati [25].

Herudover er der en høj grad af troværdighed grundet den manglende kritik af systemet. Valgene i Danmark har aldrig været udsat for skandaler i form af svindel eller store fejl, som har kunne give anledning til en debat. Det politiske system i Danmark indeholder en høj grad af tillid fra befolkningen, og dette inkluderer valgene. Dette har blandt andet medført, at der i debatten om et eventuelt e-valg, sjældent er en analyse af det eksisterende system og de sikkerhedshuller, det indeholder, udnyttede eller ej [26].

Som beskrevet i afsnit 2.3, stiller det danske demokrati krav om, at ingen må kunne kvittere for deres stemme. Dette sikre, at ingen kan sælge deres stemme eller eventuelt trues til, at stemme imod sin overbevisning. Sikringen mod, at stemmer kan spores tilbage til enkelte personer, frembringer en anden brist. Identifikationen, som sikrer, at alle kun stemmer én gang, er uden krævet billedidentifikation(som beskrevet i afsnit 2.3). Hvis både valgkort og sygesikring mistes eller på anden vis overlades til en tredjepart, kan denne stemme på en andens vegne. Det er derved muligt at stemme for andre, ved blot at være i besiddelse af disse to elementer.

Efter vælgerne har afgivet deres stemmer, er systemet afhængigt af tilliden til dem, som optæller stemmerne. Desuden bliver disse stemmer talt op manuelt; en optælling som tager lang tid. Denne optælling bygger på, at en række mennesker skal have stemmerne mellem hænderne. På trods af det papirspor, som kan følges, kan der i processen opstå menneskelige fejl.

Nogle af svaghederne ved det nuværende system, er opstået sammen med andre teknologiske frembrud. Muligheden for, at tage et billede af stemmeseddelen, eller ligefrem filme stemmeafgivningen med kameratelefoner, hindrer det demokratiske princip om, ikke at kunne kvittere for sin stemme. Ugyldige stemmer er en anden problemstilling, systemet står overfor. Til folketingsvalget i 2011 var der 34307 ugyldige stemmer [27]. Dette var beskadigede stemmesedler, markerede, mærkede eller på anden måde genkendelige eller unikke stemmesedler som blev dømt ugyldige.

3.2.2 E-valg

E-valg er en ny teknologi, som kan hjælpe med, at gøre valgprocessen nemmere. E-valg vil bl.a. gøre optællingen af stemmer nemmere, da dette kan ske elektronisk, og derfor ikke skal gøres af optællere efter selve afstemningen. Det vil evt. også være muligt, at lave foreløbige valgresultater, eftersom stemmerne bliver registreret i systemet. Samtidigt kan risikoen for menneskelige fejl mindskes, da der skal mindre personale ind over selve optællingen og håndteringen af stemmerne [28].

På lang sigt kunne et e-valgssystem blive mere økonomisk pga. effektiviteten og genbrugelige valg-metoder, da det elektroniske udstyr kan bruges af mange vælgere og også kan bruges

til flere valg. Systemet vil også kunne eliminere ugyldige stemmer, da et elektronisk system ikke vil tillade en vælger, at indsende eller afgive en ugyldig stemme i modsætning til en valgmetode baseret på papirstemmer [28]. En anden fordel ved det elektroniske valgssystem vil være, at de afgivende stemmer vil kunne sikres nemmere, da disse vil kunne sikkerhedskopieres og på den måde bevares flere steder direkte efter afgivelsen af stemmen.

Handicappede vil desuden kunne gøre nytte af et e-valgssystem, da det vil mindske behovet for hjælp ved stemmeafgivelsen. På denne måde vil flere handicappede kunne stemme ved egen hjælp og opnå stemmehemmelighed, som ikke har været muligt for dem tidligere [29].

Der har været en del kritik af e-valget fra forskellige it-kyndige organisationer. IT-Politisk forening er én af de grupper, der er stærkt kritiske over for et e-valg. De nævner blandt andet pålideligheden som et af kritikpunkterne. De skriver, at man skal kunne sikre sig, at systemet ikke kan manipuleres på nogen måde. Hverken af brugere, teknikere eller programmører. Der må på samme tid ikke være nogle muligheder for fejl i hverken software eller hardware. Alene disse punkter er meget svære, og meget dyre at løse. Et andet af deres vigtige kritikpunkter er anonymitet. De mener ikke, at det er muligt, at sikre sig, at der ikke er nogen, der kan se, hvem der har stemt på hvad [5].

Et kritikpunkt som ofte bliver benyttet i forbindelse med e-valg generelt, er gennemskueligheden. Den almene dansker har god tillid til det nuværende valgssystem, da det er nemt at forstå og følge med i, hvad der sker. Dette vil ikke være tilfældet med et e-valg, og et sådant skift kan få mange danskere til, at miste troen på valget. De fleste personer vil ikke kunne forstå, hvordan e-valget fungerer, og de har derfor ingen mulighed for, at sikre sig, at det foregår, som det skal [25].

Til sidst har der også været en del kritik af det økonomiske aspekt af e-valg. De fleste IT-organisationer er enige i, at det nok bliver lige så dyrt som det nuværende valg, hvis ikke dyrere, at afholde e-valg i stedet for traditionelt papirvalg. Blandt andet skriver PROSA, at der skal anskaffes for meget udstyr, og at det formegentlig ikke holder ret længe. ITEK er enige i, at det nok bliver dyrere, at skifte til e-valg. De mener dog, at hvis det bliver gjort ordentligt, vil det være pengene værd. Der er dog kun spekulationer til rådighed omkring det økonomiske aspekt af et e-valg, og dette argument bliver derfor ofte brugt af både fortalere og kritikere. [30]

3.3 Eksisterende løsninger

I forbindelse med Regeringens lovforslag, der kunne tillade kommunerne, at prøve forskellige e-valgsløsninger, hvor udfaldet af afstemningerne kunne være bindende, vil en undersøgelse af statens krav og holdning til en sådan e-valgsløsning kunne hjælpe med, at udforme en problemstilling. Som en forberedelse på lovforslaget havde Økonomi- og Indenrigsministeriet en teknisk dialog med 7 leverandører af e-valgssystemer [3]. Denne dialog havde til formål at undersøge, hvad markedet kunne tilbyde, og om en løsning kunne opfylde forskellige krav til både sikkerhed såvel som tilgængelighed, og samtidig holde sig inden for et rimeligt budget.

Ifølge de adspurgte leverandører, kunne en indførelse af e-valg give følgende fordele:

- Hurtigere, mere præcis stemmeoptælling og udregning af valgresultat, samt en reduktion af administrative omkostninger i forbindelse med manuel optælling af stemmer.
- Hver vælger har mulighed for at verificere, at deres stemme er registreret.
- Give handicappede mulighed for at stemme uden, at skulle have brug for hjælp (se afsnit 4.3).
- Eliminere muligheden for ugyldige stemmer, hvor det samtidig er muligt stadig, at stemme blankt.
- Minimalt tab af stemmer, da der kan tages back-up af stemmer.

I forbindelse med svagheder ved elektronisk valg, var det leverandørernes erfaringer, at sikkerhed eller tekniske vanskeligheder ikke er det man bør lægge vægt på, men at det mere er et spørgsmål om vælgerne tillid til og opfattelse af valgsystemet, der skal sættes i fokus [3].

Økonomi- og Indenrigsministeriet hæftede sig specielt ved, at en løsning skal kunne:

- Beskytte vælgerne i overensstemmelse med Grundlovens § 31.
- Valideres f.eks. ved hjælp af manuel optælling.
- Tjekkes for, om det er blevet manipuleret under en afstemning.
- Bruges af handicappede.
- Videreudvikles og skales, som ønsket.
- Holde sig inden for et overkommeligt budget.

Ifølge dialogen, bør en e-valgsløsning kunne opfylde alle de tekniske krav, som Økonomi- og Indenrigsministeriet havde hæftet sig ved, for at kunne komme i betragtning som potentiel løsning. Den præsenterede løsning i denne rapport har, derfor haft fokus på, at udvikle et e-valgssystem, der følger Grundlovens og valglovens krav om et hemmeligt og demokratisk valg.

3.4 Erfaring med e-valg

Når der diskuteres, hvorvidt teknologisk hjælp under afholdelse af valg og afstemninger er hensigtsmæssigt, kan det være relevant, at se på de tidligere erfaringer, andre lande har haft med e-valg. Her findes der både eksempler på lande, der har taget imod denne teknologi og gjort den til en fast del af hele det nationale system, samt lande, der fuldstændig har udelukket anvendelsen af e-valg med den teknologi, der eksisterer i dag. Dette afsnit tager udgangspunkt i [6], og beskæftiger sig udelukkende med nærvalgsformen af e-valg.

3.4.1 Positive erfaringer med e-valg

Overordnet set er det verdensdelene Asien og Sydamerika, som har taget ideen om e-valg til sig, og herunder er det især landene Brasilien og Indien, som benytter metoden nationalt. Den modsatte tendens ses i de europæiske lande, hvor e-valg kun i få tilfælde bliver benyttet og aldrig i national skala. Her blev e-valg afvist af 6 europæiske lande, enten efter at have været indført eller efter en testperiode.

Den store forskel på, hvorvidt e-valg opnåede succes eller ej findes hovedsagligt, når der tages højde for de systemer det erstattede, og den generelle holdning, der var til dem. I både Brasilien og Indien var der store problemer med de eksisterende valgmetoder; både organisatoriske problemer såvel som demokratiske. Her bidro e-valg i en positiv retning og gjorde afholdelse af valg lettere for regeringen og mere sikker. Grundlæggende benyttes der to enheder: en til at registrere vælgeren og en til at registrere stemmen. I både Indien og Brasilien udføres identifikationen af vælgeren af en medarbejder ved det lokale valgsted. Problemerne her er hovedsagligt, at der ikke findes et papirspor, som kan verificere stemmen. Systemerne er dog konstant under udvikling, og der arbejdes på, at indføre det.

3.4.2 Negative erfaringer med e-valg

I Europa findes den modsatte problemstilling når der diskuteres e-valg. De eksisterende metoder føles sikre og trygge for befolkningen, og de fordele, der kan opnås ved indførelsen af e-valg, opvejer i mange tilfælde ikke de ulemper, som kan følge med. Her tænkes der især på, at der gives afkald på et papirspor, samt den generelle mangel på gennemsigtighed i de eksisterende systemer.

Den europæiske mistillid til e-valg startede i Irland, hvor regeringen, efter at have investeret 53 millioner euro, måtte opgive projektet. Efter en grundig undersøgelse foretaget af en nyligt oprettet komite for elektronisk valg, fandt man grund til, at tvivle på systemets pålidelighed, og det blev påpeget, at mængden af system-tests var utilstrækkelig. Hertil bemærkede man, at stemmemaskinen gav en bib-lyd, som kunne være en mulig trussel mod vælgernes stemmehemmelighed. Det blev vurderet, at det eksisterende system var bedre end det elektroniske valgssystem, og det blev derfor ikke erstattet.

Dette gav anledning til skepsis i de europæiske lande, der allerede benyttede e-valg, og i løbet af en periode på 5 år blev e-valg afskaffet i henholdsvis Holland og Tyskland. Et af de vedvarende kritikpunkter i begge lande var stemmemaskinerne. De var af samme type, som dem, der til sidst blev afvist i Irland, og i den efterfølgende periode blev der fundet flere mangler. En gruppe modstandere af e-valg i Holland fik fat i en af disse maskiner og viste, hvor følsomme de var overfor sabotage. Nogle kunne uden de store problemer udskifte programmerede chips, og derved manipulere resultaterne fra maskinen. Da maskinerne ikke producerede et papirspor kunne ændringen ikke kunne opdages. I Tyskland blev det et lovæssigt spørgsmål, da to borgere argumenterede for, at stemmemaskinerne var imod landets forfatning. De fik medhold i sagen på baggrund af, at tyske statsvalg skal være offentlige af natur. Det vil sige, at en almen borger skal kunne følge valgprocessen hele vejen,

uden at have nogen form for specialviden. I elektroniske valg er det dog nødvendigt, at beskytte sig ekstraordinært mod eventuelle hackere og fejl, hvilket gjorde det umuligt for systemet at opfylde forfatningens krav.

3.5 Opsamling

E-valg kommer i forskellige udformninger, og er hovedsageligt delt ind i nærvalg og fjernvalg. Den eneste form der er relevant i denne rapport er imidlertid nærvalg. Der er både fordele og ulemper ved denne form for e-valg, men dette er dog også gældende for det nuværende valgsystem. En stor del af de sikkerhedsrisici, der findes, er dog mulige at efterkomme, hvis blot det tænkes ind i designfasen, og der tages højde for de forskellige problemer tidligere erfaringer har vist. Herved får man til gengæld et system, som ikke blot vil kunne have optalt valgresultatet umiddelbart efter valgets afslutning, men også et system, som samtidigt modvirker nogen af de problemer, der er opstået som følge af udviklingen af nye elektroniske apparater såsom kameratelefoner.

For at kunne vurdere hvordan et e-valgssystem skal udformes så det egner til det danske samfund, er der lavet en analyse af brugere og interessenter i næste kapitel.

Kapitel 4

Interessenter

I følgende afsnit bestemmes og analyseres de forskellige interessenter, der findes i forbindelse med indførelsen af e-valg. Under analysen undersøges det, hvilke fordele og ulemper de forskellige interessentgrupper kan opleve, og dette ender ud i en række forventninger til en eventuel løsning.

I forhold til e-valg er der mange forskellige interessenter, da en ændring af valgmetoden vil få betydning for alle stemmeberettigede. Det er regeringen der har fremsat forslaget og det er derfor dem der er hovedinteressent. Vælgerne vil naturligvis også blive berørt af e-valg, hvilket vil give nye udfordringer og muligheder for vælgerne. Dem der umiddelbart står til at vinde mest ved indførelse af e-valg, er vælgere med motoriske handicap, som ellers kan have svært ved at stemme ved papirvalg.

4.1 Politikerne

Som nævnt tidligere er det regeringen, der er de primære initiativtagere for e-valg. Dog er nogen af regeringens støtter imod e-valg, hvilket betyder at det ikke kun er regeringen, der har indflydelse på udformningen af e-valg. Regeringen har behov for at finde støtte til e-valgstiltaget i resten af folketinget. Folketinget har derfor meget magt i forhold til e-valg, da det er dem der i sidste ende skal stemme på om e-valg skal indføres. Regeringen med Margrethe Vestager, Økonomi- og Indenrigsminister i front, skal derfor forhandle om, hvordan projektet kommer i gang, samt hvordan det endeligt kommer til at se ud.

Der er flere årsager til, at den siddende regering overvejer at indføre e-valg. Mange ser e-valg som fremtidens måde at stemme på og forudser, at det kun er et spørgsmål om tid, før det fuldstændigt erstatter traditionelt papirvalg. Ved allerede nu at indføre forsøg med e-valg viser regeringen initiativ til, at prøve det nye, men giver samtidig befolkningen en tilvænningsperiode, så indførelsen af e-valg ikke kommer til, at erstatte papirvalg fra den ene dag til den anden. For at sikre denne overgang minder udformningen af det nye valgsystem meget om det gamle. Margrethe Vestager udtaler følgende: „*Blandt andet er det afgørende for mig, at det system, vi afprøver med e-valg, udskriver et papirspor, når vælgeren afgiver sin stemme elektronisk*“ [31]. Dette krav om et papirspor er derfor en nødvendig del, for at systemet kan vedtages og indføres.

Der er dog også risici involveret i overgangen til e-valg. For det første vil der være en del udgifter forbundet med projektet. Hvis projektet ikke ender med, at blive en succes, vil disse være spildte penge. Selv hvis forsøgene med e-valg bliver godkendt, er det begrænset hvor lang levetid de nuværende teknologiske løsninger har. Det kan diskuteres, hvorvidt det er fornuftigt, at investere på et tidspunkt, hvor teknologien stadig udvikles hurtigt. Herudover er der risiko for, at befolkningen vil miste tillid til regeringen, hvis der opstår seriøse fejl ved e-valg.

4.2 Vælgerne

Under udviklingen af det nye projekt skal der tages stilling til brugerne af dette, nemlig den stemmeberettigede danske befolkning. Der skal altså ses på hvilke fordele og hvilke ulemper indførelsen af et nyt valgsystem kan medføre. I denne sektion fokuseres der på den almindelige vælger, som ikke lider af svære fysiske hæmninger eller andre former for handicap. Vælgere med handicap vil blive beskrevet i sektion 4.3.

Den gennemsnitlige vælger vil, ved indførelsen af den form for e-valg regeringen har fremlagt, ikke opleve de store forskelle i forhold til de fysiske omgivelser og rutiner på valgdagen. Det er først inde i selve stemmeboksen, at der kommer en mærkbar forandring. Når stemmesedlen udfyldes via en trykfølsom skærm og efterfølgende printes, elimineres den mængde af ufrivilligt ugyldige stemmer, der i det nuværende system bliver afgivet. E-valg kan indeholde muligheden for, at afgive en blank stemme, men stemmesedler med flere krydser, tegninger ol., der før blev regnet ugyldige, bør ikke længere kunne afgives. Efter stemmesedlen er udfyldt er der mulighed for, at verificere sin stemme. Dette kan forgå ved at skanne den stemmeseddel, som vælgeren får printet, og derved få bekræftet at ens stemme er registreret og bliver talt med [3].

Ulemperne ved det nye valgsystem vil, for de fleste, have baggrund i den manglende gennemskuelighed af systemet. Som beskrevet i afsnit 3.2.2 kunne tilliden til valgformen blive et problem. Det vil derfor være afgørende for en løsning, at vælgerne føler sig trygge ved valgsystemet. Nogle dele af befolkningen kan også have forbehold i forhold til, hvordan selve afgivelsen af stemmen skal forgå og kan have brug for vejledning. Her tænkes der især på den ældre del af befolkningen, som ikke altid er opdaterede på den nye teknologi. Det kan muligvis virke afskrækkende for en ældre person, der aldrig har ejet eller i særlig grad benyttet sig af en computer, at skulle stemme via en trykfølsom skærm.

4.3 Handicappede

Den gruppe af befolkningen, som umiddelbart har udsigt til flest fordele ved indførelsen af e-valg, er de handicappede. Her tænkes der hovedsageligt på mennesker, der lider af f.eks synshandicap eller motoriske hæmninger.

Brugervenligheden inde i selve valgboksen er essentiel for denne del af befolkningen og kan udformes på flere måder. I forhold til blinde og synshandicappede er der, for det første, mulighed for, at indføre en form for oplæsning af valgsedlen, eventuelt med et par høretelefoner. Til at hjælpe dem, der blot lider af dårligt syn, kan der laves en mere letlæselig stemmeseddel på skærmen, og ved hjælp af grafiske hjælpemidler, som forstørret skriftstørrelse m.m., kan det gøres nemmere, at læse de forskellige kandidat-navne. Herudover er der forslag om knapper, joysticks o.l. Disse funktioner kan aktiveres, når det er nødvendigt, ved hjælp af et lille nøglekort, som har indregistreret en bestemt form for handicap [3].

For mennesker, der lider af f.eks. sklerose og Parkinsons sygdom, kan en digital løsning også være en hjælp [32]. Da der blot skal trykkes på en skærm, vil det ikke give lige så store problemer, som den nuværende løsning. Det nye system vil desuden give vælgeren mulighed for at ændre stemmesedlen, og rette eventuelle fejlindtastninger indtil alt er, som vælgeren ønsker det.

Vælgere med handicap har i det nuværende valgsystem ikke de samme rettigheder under selve afstemningen som den almene vælger. Dette kommer til udtryk i manglende stemmehemmelighed og generel mangel på selvstændighed under valghandlingen. Mennesker med handicap vil sandsynligvis ikke have samme tilgang til e-valg som den almene vælger, men vil opleve at der for dem er langt flere fordele end ulemper ved et elektronisk valgsystem.

4.4 Forventninger til e-valg

På baggrund af interessentanalysen er følgende forventninger til løsningen fundet og opstillet i form af en række punkter:

- Ingen kompromis i forhold til sikkerhed, beskyttelse af stemmerne og vælgernes stemmehemmelighed.
- Layout skal formes efter de nuværende stemmesedler, for at give større brugervenlighed.
- Der skal implementeres en eller flere meget brugervenlige udgaver, som aktiveres når en handicappet vælger skal stemme.
 - Større skriftstørrelse til dem med svagt syn, samt oplæsning til de blinde.
 - Store ikoner, så mennesker, der har problemer med Parkinsons eller Sklerose, har nemmere ved at stemme.
- Mulighed for at verificere den afgivne stemme.
- Mulighed for at ændre stemmen indtil den endelig godkendelse.

Dette er blot en vejledende liste, og de enkelte punkter skal derved ikke nødvendigvis opfyldes i projektets løsning.

4.5 Opsamling

Hvordan et system er designet, kommer naturligvis meget an på dem, der bruger systemet. For et e-valgssystem er dette specielt vigtigt, da det skal kunne være muligt for alle, at kunne stemme. De handicappede har specielt mange vanskeligheder med det nuværende valg, som muligvis kan løses helt eller delvist af e-valg. Derfor er deres behov vigtige at overveje i designet af en e-valgsløsning.

I næste kapitel afgrænses nogle af disse aspekter, for at gøre det muligt, at lave et initierende løsningsforslag.

Kapitel 5

Problembeskrivelse

Som sidste del af problemanalysen er problemet blevet afgrænset og formuleret. Problembeskrivelsen gennemgår afgrænsningen af problemstillingen for at ende ud i endelig problemformulering. Afgrænsningen sammendrager de elementer og aspekter af det initialiserende problem, som problemanalysen har vist, tager højde for demokratiet, valgloven, interessenterne og de teknologiske muligheder forbundet med elektroniske valg. Den endelige formulering, danner på den måde et spørgsmål, hvis besvarelse er konkluderende på de dertil afgrænsede aspekter.

5.1 Afgrænsning

Givet de mange indfaldsvinkler til emnet e-valg, har det været en nødvendighed, at afgrænse projektet til nogle bestemte fokusområder. Dette projekt tager derfor ikke højde for eventuelle problemer, som e-valg kunne have med brugen af økonomiske ressourcer. Det vil hovedsageligt fokusere på besværligheder i forbindelse med teknisk implementering og folkelig accept på en sådan måde, at den danske Grundlov bliver overholdt. I forbindelse med dette, vil projektet så vidt muligt forsøge, at opfylde interessenternes behov for brugervenlighed, samt give vælgeren mulighed for, at verificere sin endelige beslutning. Der afgrænses yderligere fra, at implementere specielle funktioner til handicappede. Der vil desuden blive undersøgt hvordan forsøg på snyd kan mindskes i forhold til ny teknologi. Projektet beskæftiger sig med valg afviklet ved valgsteder og ser dermed bort fra fjernvalg, hvor vælgerne kan stemme f.eks. over internettet.

Afgrænsningen, at arbejde med valg afviklet på valgstederne i modsætning til fjernvalg, er lavet på baggrund af afsnit 3.1, hvor der bliver argumenteret for, at dette ikke passer ind i de nuværende rammer, som grundloven sætter op; at valg til folketing skal foregå ved hemmelig afstemning. Det bliver derfor konkluderet, at det vil være umuligt at lave en fjernvalgs-løsning, der ikke rejser besværligheder i form af uoverensstemmelse med Grundloven. Derfor vil dette projekt fremover ikke behandle idéer omkring fjernvalg.

Afgrænsningen af implementering og accept er lavet, på baggrund af en af de centrale tenderende kritikpunkter af e-valg, som beskrevet nærmere i problemanalysen: At det er svært, hvis ikke umuligt, at lave en e-valgsløsning, der vil være gennemsigtig nok, til at vælgerne vil stole på valgformen og samtidig sørge for et sikkert og hemmeligt valg (se afsnit 3.2.2).

5.2 Problemformulering

Dette leder frem til, at opstille en problemformulering, som skal hjælpe med, at sætte nogle specifikke krav til løsningen. Ud fra nøgleproblemet: *Kan det nuværende valg forbedres med eksisterende teknologi?*, lyder vores problemstillinger som følger:

Hvordan ville det kunne lade sig gøre, at udvikle et pålideligt og elektronisk valg-system, til fremmødevalg i Danmark?

- *Hvordan kan et valg afvikles, så vælgeren kan afgive sin stemme på valgstedet elektronisk?*
- *Hvordan kan den fordelagtige hastighed af elektronisk stemmeoptælling og pålideligheden ved papirstemmer opnås?*
- *Hvordan kan man sikre, at et e-valgssystem er både sikkert og gennemsækeligt?*
- *Til hvilken grad kan man udvikle et brugervenligt e-valgssystem til gavn for flest mulige vælgere?*
- *Hvordan kan vælgerens anonymitet sikres ved et e-valgssystem?*

Del II

Løsningsforslag

Kapitel 6

Introduktion til Løsning

Med udgangspunkt i problemanalysen er der, som løsningsforslag, udviklet et elektronisk valgsystem. Implementeringen, kaldet *EVS E-Voting System*, er baseret på diverse overvejelser omkring pålidelighed og sikkerhed ved både de eksisterende valgmetoder og andre metoder til E-valg.

Løsningsforslaget er skrevet i programmeringssproget C#, som giver meget abstraktion fra hardwaren. Dette gør det alsidigt, da meget abstraktion gør, at der ikke er særligt mange forudsætninger for det hardware, det skal køre på.

I de følgende afsnit forklares, hvilke metoder, der er benyttet, og de tekniske valg, som er foretaget under udviklingen af løsningsforslaget.

6.1 Kravspecifikation

Med baggrund i analysen er der lavet en mindre kravspecifikation for Løsningsforslaget. Det skal fungere som en prototype og ikke en endelig løsning. Der er derfor nogle krav, som ikke bliver opfyldt af det fremlagte løsningsforslag. De krav, som løsningsforslaget bør opfylde er:

1. Løsningen skal laves i C#. Dette er et eksternt krav til projektet.
2. Løsningen må ikke gå på kompromis med sikkerheden i forhold til nuværende valg (se afsnit 4.1).
3. Løsningen skal sikre vælgerens anonymitet.
4. Løsningen skal have en brugerflade, der ikke gør det sværere for en stemmepligtig at stemme. Alle dem, der kan stemme nu, skal også kunne stemme med det elektroniske valgsystem.
5. Løsningen skal så vidt muligt holde sig indenfor de nuværende lovmæssige begrænsninger for valg.

Udover disse er der en række krav, som et fuldstændigt E-valgsystem skal implementere, men som er fravalgt i løsningsforslaget. Der er derfor ikke fokuseret på at opfylde disse krav, men der er heller ikke lukket af for, at systemet kan udvides til at opfylde dem på et senere tidspunkt. De krav, som løsningsforslaget ikke opfylder er:

6. Brugerfladen skal være handicap-venlig. Dette kan for eksempel være gennem oplæsningssystemer, andre inputmetoder, osv. (se afsnit 4.3).
7. Løsningen skal have et papirspor, for at sikre holdbarheden af systemet. Regeringen har, på rapportens afleveringstidspunkt, udtalt, at dette er et krav, for at E-valg kan indføres i Danmark (se afsnit 4.1).

6.2 Introduktion til *EVS E-Voting System*

Følgende beskrivelse af EVS er den overordnede vision for funktionaliteten af løsningen. Gennemgangen skal give overblik samt indlede, hvordan den kan implementeres i det nuværende valgmiljø. Herefter gives der et kort overblik over efterfølgende kapitlers beskrivelse af løsningens programstruktur.

6.2.1 Praktisk overblik

Overordnet er systemet sat op til, at vælgeren skal benytte en tilfældigt genereret kode, som er udleveret af en valgsagkyndig ved fremvisning af valgkort. Først skal vælgeren bekendtgøre sin identitet, som sammenlignes med listen over vælgere, hvorefter vedkommende får udleveret koden i en forseglet konvolut, hvis han/hun er stemmeberettiget og ikke allerede har fået udleveret en kode. Herefter skal vælgeren gå ind i en stemmeboks af eget valg, hvor en stemmeterminale vil være til rådighed. Koden indtastes da på terminalens identifikationsskærm og bliver sendt til valgstedets identifikations-server, som tjekker kodens gyldighed. Efter koden er blevet verificeret guides vælgeren igennem valgprocessen, der ender med at vælgeren vil afgive en parti-, kandidat- eller blank stemme. Stemmen ville på dette tidspunkt, i den endelige løsning, blive fysisk udskrevet (som foreslået af Margrethe Vestager [31]) det kan eksempelvis foregå bag en glasskærm og vil, efter vælgerens godkendelse, blive lagret i en, af vælgeren utilgængelig, stemmeurne på valgstedet. Den kode, som vælgeren fik udleveret vil så endnu engang blive sendt til valgstedets identifikations-server, som vil ugyldiggøre koden i fremtiden; med andre ord bliver koden forbrugt på dette tidspunkt. Herefter vil en digital version af stemmen blive gemt i stemmeterminalens stemme-database. Valghandlingen er nu afsluttet for vælgeren, og stemmeterminalen returnerer til identifikations-skærmen, der skal imødekomme den næste vælger. Ved valgets afslutning lukkes alle valgstedets stemmeterminaler, som, i en del af deres nedlukningsproces, sender deres stemme-databaser til valgstedets centrale stemme-database, der efterfølgende kan lave en elektronisk stemmeoptælling. Ved at dele identifikations-processen og stemme-processen op i to dele, der kører uafhængigt og upåvirket af hinanden, kan antallet af stemmer og antallet af koder, der er brugt, blive sammenlignet, for at afsløre eventuelt valgfusk. Ved en endelige løsning, der implementerer et papirspor, kan de udskrevne stemmer fra stemmeterminalerne tælles op i tilfælde af tvivl.

6.2.2 Teknisk overblik

Løsningsforslagets tekniske implementeringer er i detaljer beskrevet i den følgende del af rapporten, som er inddelt i separate kapitler. De overvejelser, der er foretaget i forbindelse med projektets brug af databaser, netværk og brugergrænseflade, er beskrevet i hver sit afsnit. Disse afsnit redegør for de bagvedliggende overvejelser, som ligger til grund for de valg, der er truffet i forbindelse med løsningens struktur og brug af teknologier. De første to afsnit, der hhv. omhandler database- og netværksvalg, beskæftiger sig med selve behandlingen af data og kommunikationen imellem klient-terminal og server. Det tredje afsnit belyser, hvordan brugervenlighed kan være en del af et vælgersystem, og hvilke overvejelser, der er blevet gjort under udviklingen af en passende grafisk brugerflade til løsningsforslaget. I de sidste to afsnit vil prototypens endelige struktur og funktionalitet blive gennemgået og derefter testet.

I de følgende afsnit påbegyndes disse tekniske overvejelser, og helt grundlæggende beskæftiger det næste afsnit sig med valg af database til projektes prototype.

Kapitel 7

Database

I mange IT projekter er det nødvendigt at indsamle, opbevare og behandle forskellig data. For at tilgå og organisere disse data er det oftest nødvendigt at bruge et databasesystem. I dette afsnit undersøges og udvælges der et databasesystem til EVS.

7.1 Databasesystemer

At bruge en database har mange fordele i forhold til at opbevare data direkte i programhukommelsen eller en data-fil. Databasesystemer kan garantere større datasikkerhed, lagre en stor mængde data og understøtter flere simultane læse-/skrive-operationer. At opbevare og behandle store mængder data sikkert og effektivt er et meget stort problemområde og der findes derfor mange forskellige tilgangsvinkler til dette. Der er ofte meget strenge krav til pålideligheden af disse databasesystemer. For eksempel er det ofte et krav at data ikke må gå tabt i tilfælde af strømsvigt.

Et af de mest brugte principper er den *transaktionelle database* som er udviklet af Haerder og Reuter i 1983 [7]. Her deles hver operation på databasen op i *transaktioner*. En transaktion kan være indsættelse af data, behandling, eller en hvilken som helst anden operation der ændrer databasens tilstand. Haerder og Reuter udviklede desuden også ACID princippet der stiller krav til hvad en database skal kunne garantere, for at den kan ses som at værende en pålidelig transaktionel database [7].

I nyere tid er der kommet en række forskellige databasesystemer, som ikke følger ACID helt for at kunne skalere mere effektivt op til meget store databaser. Dette er for eksempel tilfældet i Googles BigTable databasesystem, som blandt andet bruges i deres søgemaskine [8].

I forhold til E-valg er databasen meget vigtigt, da der er meget store krav til sikkerheden når man opbevarer stemmedata. ACID princippet er derfor et afgørende krav i forhold til valget af database. Vi har til EVS overvejet og undersøgt tre forskellige databasesystemer: SQLite, MS SQL og MySQL.

SQLite

SQLite er, som navnet antyder, et meget lille databasesystem (ca 350 KB [33]). SQLite

er ofte inkluderet direkte i programmet, hvilket adskiller sig fra de fleste andre databasesystemer som oftest er i en adskilt proces og evt. adskilt computer. SQLite er blevet brugt i tusindvis af forskellige stykker software som blandt andet Google Chrome, Skype, og iOS. [34].

SQLite er primært til mindre databaser. I forhold til E-valg betyder dette at SQLite kan være egnet til en enkelt valgmaskine eller valgsted, men det egner sig sandsynligvis ikke til at indeholde al valginformationen for hele landet. SQLite følger dog alle ACID principperne [35], hvilket gør at det har den nødvendige holdbarhed til at kunne håndtere valginformationen.

Microsoft SQL Server

Microsoft SQL Server (MS SQL) er et SQL databasesystem som er meget alsidigt. MS SQL skal dog køre som en individuel proces, hvilket oftest bliver gjort på en dedikeret computer som så tilgås via netværk, f.eks. internettet. Systemet kan opsættes til at håndtere både mindre og store arbejdsopgaver, og er derfor brugsmæssigt meget alsidigt [36]. MS SQL byder desuden på en lang række af sikkerhedsprocedurer, heriblandt kryptering af data, brugerstyring [37] samt features der nedsætter eventuel nedetid ved afbrydelser, strømsvigt, etc. [38]. MS SQL er brugt af mange virksomheder og organisationer som f.eks. Microsoft, Maersk, Great Western Bank og det Amerikanske forsvarsministerium [39]. MS SQL følger desuden ACID princippet hvilket gør MS SQL brugbar i E-valgs øjemed.

MS SQL er som sagt et større system der skal køre separat og oftest bliver afviklet på en dedikeret maskine. Derfor vil MS SQL være mest egnet som en central løsning hvor de enkelte valgmaskiner via et netværk kommunikerer med MS SQL serveren.

MySQL

MySQL er et open-source databasesystem som tilbyder en lang række muligheder til brug i større og mindre databasestrukturer. MySQL kan f.eks. opdele læsearbejdet fra databasen i såkaldte slave-servere, der så synkroniseres med en enkelt master-server som alene håndterer skrivearbejdet [40]. Denne løsning er altså utroligt effektiv til programmer der kræver mange læsninger af store mængder data og det er ikke usandsynligt med op til 30 slave-servere til en enkelt master-server i større online tjenester [40]. Det er ligeledes muligt at koble flere master-servere sammen enten i et såkaldt loop eller en cluster hvor serverne skriver til hinanden og alle kan modtage data. På den måde kan både læse og skrive hastigheden øges drastisk samtidigt med at risiko for fejl og nedbrud mindskes, da flere servere vil have den samme information liggende samtidigt [40].

MySQL har desuden en høj sikkerhed mod tab af data og nedetid, da en af slave-serverne kan tage over for master-serveren og synkroniseres med andre slave-servere således at alt data er bevaret på den nye master-server [41].

I E-valgs øjemed opfylder MySQL ACID princippet og er dermed egnet til vores projekt. MySQL tilbyder desuden en masse sikkerhedsforanstaltninger der gør at hvis en enkelt server går ned er der straks en til at tage over, hvilket gør at valget ikke vil blive afbrudt på grund af servernedbrud.

7.1.1 Valg af databasesystem

I E-valgsprojektet skal der bruges en database der kan håndtere store mængder data samt mange brugere samtidigt. Valget af database vil dog ikke nødvendigvis falde ud fra disse kriterier alene, da man ved at dele byrden op vil kunne nøjes med flere mindre databaser, frem for en enorm database der skal håndtere alle data og alle brugere på en gang.

En mulighed er at dele arbejdsbyrden op i flere dele, dette kan gøres ved at lade hver enkelt valgmaskine på valgstedet have en separat database som indeholder stemmerne for den enkelte maskine. På den måde er kravet til den enkelte database ikke nært så stort og det er derfor muligt at bruge en databasetype som f.eks. SQLite. En anden mulighed er at samle alle dataene fra de forskellige valgsteder umiddelbart efter stemmeafgivelsen. Dette forgår ved hjælp af et netværk, evt. internettet, hvor stemmedataene krypteres og sendes til en central database der afvikler f.eks. MS SQL eller MySQL. Arbejdsbyrden kan ydermere opdeles til flere servere hvorved belastningen vil mindskes, eller flere servere kunne modtage samme information hvorved man vil sikre sig imod eventuelle manipulationer af den centrale servers data.

På baggrund af disse overvejelser og da vi ønsker at bruge en enkelt database på hver klientterminal, er SQLite blevet valgt som databasesystem for EVS. SQLite benyttes i løsningen til at samle og behandle data for det enkelte valgsted under valgdagen.

Hvis en endelig e-valgsløsning skal udvikles på samme måde, som EVS løsningsforslaget, vil SQLite stadig være at foretrække. Dens størrelse gør den ideel, når hver stemme-terminal skal have en lokal database kørende under valget. Først hvis en løsning skulle bygges på, ikke at opbevare stemmerne på de enkelte stemme-terminaler, kunne det være relevant, at se på andre databasesystemer.

Dog er det ikke alene nok at opbevare denne data på en enkelt maskine. For at kunne håndtere flere valgmaskiner og dataene imellem dem er det nødvendigt at bruge en form for kommunikation mellem computer. Derfor undersøger vi i det næste kapitel de forskellige muligheder for datakommunikation.

Kapitel 8

Datakommunikation

For at kunne håndtere et valg, er det nødvendigt, at de forskellige valgmaskiner kan kommunikere og samle stemmerne ved valgets afslutning. Derfor er det nødvendigt, at have en form for kommunikationsstruktur, som gør det muligt for stemmerne, at blive afsendt sikkert. I dette afsnit vil der blive beskrevet og vurderet forskellige former for datakommunikation, og afslutningsvist valgt en metode, der skal implementeres i EVS.

8.1 Fysik datadeling

Den simpleste måde at samle stemmerne efter valget, er at flytte de fysiske medier, som dataene bliver oplagret på. Disse fysiske medier kan være SD-kort, harddiske eller lignende. Ved endt valg skal en valgsagkyndig så samle alle disse medier sammen og sætte det til en computer, som så kan lave optællingen. Dette system er simpelt, men det gør det betydeligt nemmere, at lave menneskelige fejl. Problemet ved denne metode er desuden også, at den ikke egner sig til, at sende data løbende.

8.2 Lokalt netværk

At have et lokalt netværk¹ på det enkelte valgsted, kan løse nogle af disse problemer. Data kan effektivt sendes løbende mellem valgmaskiner, og systemet er mere resistent over for menneskelige fejl.

Efter at det enkelte valgsted har foretaget optællingen af stemmerne, skal alle valgstedernes resultater tælles sammen hos Indenrigsministeriet. Dette kunne gøres ved, at sende de elektroniske stemmer over internettet. Det medfører dog en række sikkerhedsbekymringer, og løsningsforslaget vil derfor også kun behandle datakommunikation internt på valgstedet. Til dette vil der gøres brug af et lokalt netværk.

¹*Engelsk:* Local Area Network

8.2.1 HTTP

For at kommunikere mellem to computere over LAN, er det nødvendigt, at bruge en form for protokol, så dataene er struktureret på en måde, begge computere kan forstå. I EVS bruges HTTP (HyperText Transfer Protocol), en meget udbredt protokol, som blandt andet bruges i World Wide Web [42]. HTTP følger en “spørg-svar” model², hvor en klient opretter forbindelse til en server, sender en forespørgsel for så, at modtage et svar fra serveren. Serveren kan være en dedikeret computer på selve valgstedet, hvortil stemmeterminalerne sender forespørgsler.

8.2.2 HTTPS

HTTPS (HyperText Transfer Protocol Secure) er en krypteret udgave af HTTP, som bruger enten SSL (Secure Socket Layer) eller TLS (Transport Layer Security) til, at sikre sikker og fortrolig afsendelse af data [43]. HTTPS er, ligesom HTTP, vidt udbredt og bruges blandt andet på websider hvor øget sikkerhed er nødvendigt. Dette kan for eksempel være netbank, betalings-websider eller offentlige websider hvor personlige oplysninger benyttes.

For at kunne oprette forbindelse til en server med HTTPS, er det nødvendigt, at kende dens certifikat. Normalt findes dette certifikat på internettet ved at bruge *certifikat autoriteter*. Men da der ikke er adgang til dem på et lukket lokalt netværk, er dette ikke muligt. Derfor er det nødvendigt, at distribuere certifikatet til alle valgmaskinerne før valgets start. Dette kunne for eksempel gøres ved enten indtastning eller stregkode-scanning.

.NET-biblioteket understøtter HTTPS, men for at køre en sikker HTTPS forbindelse, er det nødvendigt, at opsætte de nødvendige certifikater hvilket kan være udfordrende. Derfor understøtter EVS kun HTTP, men det er muligt at udvide funktionaliteten sådan at HTTPS kan understøttes, såfremt valgcomputerne har det krævede certifikat.

8.2.3 Opsamling

HTTP egner sig meget godt til EVS og er derfor brugt til alt netværkskommunikation. Som nævnt i afsnit 8.2.2, så implementeres HTTPS ikke i EVS på grund af at det er en del omfattende at sætte op. Hvis EVS skal bruges til et virkeligt valg, burde dette dog ændres for at sikre systemet mod valgsvindel.

At opbevare og overføre data er dog ikke tilstrækkeligt for et e-valgssystem: interaktion med brugerene er bestemt ligeså, hvis ikke mere, vigtig. Derfor undersøger vi i næste afsnit teorien og praksis i at designe en hensigtsmæssig brugergrænseflade.

²*Engelsk*: Request-response model

Kapitel 9

Grafisk brugergrænseflade

For at optimere kommunikationen med vælgerne, er brugervenligheden vigtig for løsningsforslaget. Brugervenlighed er et begreb, der forklarer hvor enkelt og let-anvendeligt en brugergrænseflade er, at bruge. Brugervenlighed bør være en vigtig del af alle softwarebaserede applikationer, og den skal tilpasses de rigtige brugere af softwaren. I dette afsnit vil der blive set på en metode til, at implementere brugervenlighed i løsningsforslagets grafiske brugergrænseflade.

9.1 Brugervenlighed

I en e-valgsapplikation er det nødvendigt, at ramme en bred gruppe af brugere, og der vil være store udfordringer, når det kommer til f.eks. brugere med andre sprogkunderskaber end dansk, vælgere, som aldrig har benyttet en computer, og måske vil den største udfordring være, at gøre applikationen brugervenlig for handicappede. I dette projekt vil der ikke blive udarbejdet store gennemanalyserede undersøgelser af vælgernes behov for brugervenlighed, men blot en meget generel og minimalistisk brugergrænseflade, som henvender sig til den alment computer-vante bruger, der kan læse dansk og hverken er blind eller motorisk hæmmet.

God brugervenlighed kan defineres på flere måder. En globalt meget anerkendt tilgang til brugervenlighed er udarbejdet af Jakob Nielsen. Jakob Nielsen er en dansk datalog med speciale i IT-brugervenlighed, og er internationalt anerkendt for sin ekspertise. Brugervenligheden kan defineres ud fra fem kvalitets komponenter opsat af Jakob Nielsen: [44]

Let at lære

Hvor let det er for brugeren, at forstå og udføre basale opgaver, første gang de bruger applikationen. Dette er i høj grad vigtig for en valg-løsning, hvor brugeren kun skal igennem valg-processen én gang i løbet af et valg.

Effektivt at bruge

Hvor hurtigt brugeren kan udføre de basale opgaver, når brugergrænsefladen er lært. Dette er ikke så vigtigt i forhold til en valg-løsning, da brugeren ikke skal igennem valg-processen flere gange i løbet af et valg.

Let at huske

Hvor nemt brugeren har ved, at genetablere den grundlæggende forståelse af brugergrænsefladen.

Fejl

Hvor mulige fejl brugeren laver, og hvor alvorlige disse fejl er.

Tilfredsstillende at bruge

Hvor behagelig brugergrænsefladen er af benytte.

Disse kriterier er en overordnet definition af brugervenlighed, der kan bruges, som en generel vejledning til et mere brugervenligt design. Jakob Nielsen opdeler de fem kvalitetskomponenter op i ti brugervenligheds-heuristikker.

9.2 De ti Heuristikker

Jakob Nielsen har også defineret ti heuristikker, som er de ti mest generelle principper indenfor brugervenlighed [45]. De dækker over de fleste aspekter inden for brugervenlighed på en meget generel og håndgribelig måde. Her vil alle de ti heuristikker nævnes. Først beskrives det kort, hvad heuristikken står for, og herefter hvordan den implementeres af løsningsforslaget:

1. Synlighed af systemets status

Brugeren skal regelmæssigt underrettes om systemets status, og hvordan systemet har forstået brugeren, gennem passende feedback indenfor en passende tidsramme. Dette implementeres ved, at brugeren får vist en bekræftelses-skærm når valghandlingen er gennemført. Brugeren vil også få vist en hjælp-skærm, hvis et input ikke blev godkendt.

2. Match mellem system og den virkelige verden

Brugeren bør ikke introduceres for nye eller ukendte ord i systemet, og sproget skal være virkelighedsnært og forståeligt for brugeren. Dette implementeres ved, at bruge de samme udtryk, som bliver brugt ved den nuværende valgmetode. Der introduceres kun en ny vending; Det personlige “Vælger-ID”, som, ud over at det bliver forklaret af programmet, også er en fysisk del af den foreslåede løsning.

3. Bruger-kontrol og -frihed

Brugeren skal have tydeligt markerede “nødudgange”, eller “returbilletter”, for at forlade en uønsket tilstand. Dette implementeres ved, at brugeren altid vil have en “tilbage”-knap eller en “annuller”-knap til rådighed under hele valghandlingen.

4. Konsistens og standarder

Brugeren bør ikke spekulere på, om forskellige ord, situationer eller handlinger betyder det samme. Dette implementeres ved, at lave brugergrænsefladen så flydende så muligt, for ikke at skabe forvirring for vælgeren. Den flydende brugergrænseflade betyder, at brugeren hele tiden kan se og læse sig til, hvad næste skridt skal være.

5. Forebyggelse af fejl

Frem for at brugeren i første omgang laver fejl, foretrækkes en omhyggelig brugergrænseflade, som forhindrer en fejl i at opstå. Dette implementeres ved, at programmet

selv sørger for at håndtere eventuelle fejl, som brugeren kunne komme til at lave. Brugeren vil blive underrettet herom, eller vises en passende hjælp-skærm. De steder hvor brugeren kan komme i tvivl er en "hjælp"-knap tilgængelig, som kan informere brugeren om den pågældende situation.

6. Anerkendelse snarere end tilbagekaldelse

Minimer brugerens hukommelses belastning, ved at gøre objekter, handlinger og muligheder synlige. Brugeren skal ikke huske oplysninger fra én del af dialogen til en anden. Dette implementeres ved, at brugeren kun skal huske på hvem, eller hvad, vedkommende vil stemme på, så klarer systemet resten. Selv når brugeren befinder sig i kandidat listen, efter at have valgt partiliste, er det muligt at se hvilken liste, der er valgt, uden at være nødsaget til, at gå tilbage i processen og tjekke efter. Den endelige bekræftelses-skærm, der underretter brugeren om at verificere sit valg, indeholder derfor også alle de nødvendige oplysninger, som vælgeren kunne få brug for, så som navn og parti.

7. Fleksibilitet og effektivt brug

Genveje, uset af uerfarne brugere, fremskynder ofte interaktion for ekspert brugeren, således at systemet kan tage højde for både uerfarne og erfarne brugere. Tillad brugere at skræddersy hyppige handlinger. Dette implementeres ofte ved lave genveje til de erfarne brugere. I dette projekt vil det implementeres omvendt ved at indføre så få og enkle trin som muligt så den erfarne bruger nemt og hurtigt kan komme igennem systemet, og tilføje valgfri hjælp til mere uerfarne brugere vha. de førnævnte "hjælp"-knapper.

8. Æstetisk og minimalistisk design

Gør designet simpelt, minimalistisk og æstetisk flot, for at gøre det indbydende for brugeren. Enhver unødvendig eller irrelevant information kan forhindre de relevante informationer fra, at nå brugeren, og dermed begrænses den optimale dialog mellem brugeren og systemet. Dette vil implementeres ved, at gøre designet så simpelt og rent, som muligt, for at indbyde brugeren til, at benytte systemet. Den minimalistiske stil skal dominere systemets ydre og dermed implementeres æstetikken på de simple elementer. Baggrundsfarven er valgt til, at være lys-grøn, fordi den ikke generer øjet og stadig er i god kontrast til den sorte skift. Den sorte skift er det mest behagelige for øjet og tilføjer derfor mindst "støj" til designet.

9. Hjælpe brugerne genkende, diagnosticere og gendanne fra fejl

Fejlmeddelelser skal udtrykkes i et klart sprog (ingen koder), der præcist angiver problemet, og konstruktiv foreslår en løsning. Dette implementeres ved, at bruge tydeligt og klart "hverdags-dansk" i løsningens fejlbeskeder, der samtidig henviser brugeren til, at spørge de valgtilforordnede hvis problemet er teknisk vanskeligt. Er fejlsens omfang simpelt nok, til at brugeren selv kan løse den, beder programmet blot brugeren følge én eller to enkle handlinger.

10. Hjælp og dokumentation

Selvom det er bedre, hvis systemet kan anvendes uden dokumentation, kan det være nødvendigt at yde hjælp og dokumentation. Alle sådanne oplysninger bør være lette at finde, fokuseret på brugerens opgaver, liste de konkrete opgaver og skridt og ikke være for omfattende.

Dette implementeres ved, at udarbejde nemme brugervejledninger, både til den generelle bruger (se afsnit 16.2 i Bilag) og til de administratorer (se afsnit 16.1 i Bilag), som skal hjælpe brugeren, hvis der opstår tekniske fejl.

9.3 Implementering af brugergrænseflade

Brugervenligheden i et IT system implementeres ofte i en brugergrænseflade, også kaldet et GUI (Graphical User Interface). Et GUI er en grafisk repræsentation af de muligheder brugeren har i programmet. På denne måde kan brugere, på en mere overskuelig måde, benytte programmet. I dette projekt er brugergrænsefladen baseret på det forrige afsnit om brugervenlighed, i særdeleshed de ti Heuristikker (se afsnit 9.2). Hensigten er, at gøre afgivelsen af elektroniske stemmer simpel og intuitiv for vælgeren. Denne brugergrænseflade er en del af prototypen, og giver en ide om, hvordan en endelig løsning kunne se ud fra brugerens synsvinkel. Brugergrænsefladen er lavet på dansk og vælgeren har ikke mulighed for, at vælge et andet sprog. Muligheden for, at vælge sprog vil selvfølgelig være en foretrukken løsning, der helst skulle være tilgængeligt i en endelig løsning, men som her i prototypen er udeladt, for at holde programmet simpelt. Programmet er designet til, at skulle bruges på en computer med en trykfølsom skærm. Knapperne og resten af de grafiske elementer vil derfor være store, og tekststørrelsen vil samtidig være tilsvarende større.

Brugeren vil i brugergrænsefladen først blive mødt af et vindue (se figur 9.1), hvor brugeren skal opgive sit personlige “Vælger-ID”. Dette vil i den endelige løsning kunne komme til at foregå via en strekcode-læser, som automatisk indlæser koden og aktiverer næste trin.



Figur 9.1: Velkomstskræmen. Her skal vælgeren indtaste sit Vælger-ID til validering

Når brugeren har opgivet sit Vælger-ID til klienten, og serverdatabasen har valideret den, bliver brugeren sendt videre til “valgkortet”, som hentes fra en af valgstedets servere (se figur 9.2). Her får brugeren mulighed for, at vælge en partiliste. Heriblandt findes også en liste med løsgængere. I vinduet er der to knapper, der kan benyttes af brugeren til at fortsætte valghandlingen; en knap til, at stemme blankt, og en knap til, at fortsætte med den valgte liste.



Figur 9.2: Selve “valgkortet” hvor brugeren har mulighed for, at stemme blankt eller vælge det parti, vedkommende vil stemme på

Vælges en partiliste efterfulgt af et tryk på knappen “Videre”, skifter vinduets indhold til nu i stedet, at vise en liste med de mulige kandidater fra den valgte partiliste (se figur 9.3). Her er det så muligt, at markere og stemme på kandidat. Brugeren kan også vælge blot at stemme på selve listen. Til det formål er der to knapper, der hhv. aktiverer en bekræftelses-dialog for valg af kandidat eller parti. Vil vælgeren stemme blankt, er han/hun nødt til, at gå tilbage til forrige skærm ved, at trykke på “Tilbage”-knappen.

The screenshot shows a web interface for voting. At the top left, the text 'Vælg Kandidat:' is underlined. To its right, the party name 'Det Konservative Folkeparti' is displayed. In the top right corner, there is a button labeled 'Hjælp'. Below the party name, a list of seven candidates is shown: Lars Barfoed, Tom Behnke, Lene Espersen, Mai Henriksen, Benedikte Kjær, Mike Legarth, Brian Mikkelsen, and Per Stig Møller. At the bottom left, there is a button labeled 'Tilbage'. At the bottom right, there are two buttons: 'Stem på Parti' and 'Stem på Kandidat'.

Det Konservative Folkeparti	
Lars Barfoed	
Tom Behnke	
Lene Espersen	
Mai Henriksen	
Benedikte Kjær	
Mike Legarth	
Brian Mikkelsen	
Per Stig Møller	

Figur 9.3: Et eksempel på en kandidat-liste hvor brugeren har mulighed for, at stemme på en bestemt kandidat eller på den, fra sidste skærm, valgte partiliste, hvis partinavn ses over kandidatlisten

Inden brugerens stemme bliver registreret, hvad enten den er en blank stemme, en kandidatstemme eller en partistemme, skal brugeren bekræfte sit valg. Dette sker på en ny side, hvor brugeren evt. kan se et billede af kandidaten eller partiets logo samt opsummerende oplysninger om stemmen (se figur 9.4). Denne funktion skal sikre, at brugeren er sikker i sit valg, og ikke har forvekslet kandidaterne.



Figur 9.4: Et eksempel på en kandidat, det er muligt at stemme på. Dette er den sidste skærm, brugeren skal interagere med hvis vedkommende vælger at stemme ved at trykke på “stem”-knappen

Når brugeren har afgivet sin stemme, enten som en blank stemme eller som en stemme på en liste eller en kandidat, får vælgeren vist en skærm, der fortæller at valghandlingen er udført korrekt (se figur 9.5). Vælgeren bliver herefter bedt om, at overlade stemmeboksen til det næste vælger. Denne besked vil kun være synlig i 5 sekunder, for ikke potentielt at kunne forvirre den næste vælger, der benytter stemme-terminalen.



Figur 9.5: *Brugerens bekræftelse af valghandlingen. Det er ikke muligt på denne skærm, at se, hvad vælgeren har stemt*

Nu vil velkomstskaermen blive vist igen, og en ny vælger kan benytte valgterminalen.

9.4 Opsamling

Den overstående beskrivelse af brugervenlighed og brugergrænseflade er udelukkende bygget på Jakob Nielsens definition, og bygger ikke på nogen brugervenligheds-undersøgelse eller andre former for analyser af brugerene. Der er ikke lavet tests eller undersøgende evalueringer af løsningens brugergrænseflade, andre end de af projektgruppen interne evalueringer og overvejelser. Som et resultat af en brugervenlig implementering er brugerens interaktion med systemet blevet forsimplet, så gængse brugere af IT-systemer hurtigt vil kunne komme igennem systemet, og mindre gængse brugere har mulighed for, at få hjælp og dermed forebygge fejl i processen. Brugergrænsefladen er udviklet med fokus på, at være nem og behagelig at benytte.

Med dette afsnit er de endelige overvejelser vedrørende løsningsforslag slut, og i næste afsnit beskrives opbygningen af systemet.

Kapitel 10

Programbeskrivelse

EVS er designet til at passe så vidt muligt på det nuværende valgsystem. Der er derfor fokus på at separere stemmeafgivning og -optælling. Dette betyder at alle stemmerne opbevares som enkelte, diskrete stykker data indtil der endeligt foretages optælling af stemmerne. Samtidigt ønskes det også at adskille stemmeafgivning og vælgeridentifikation, da det giver bedre stemmehemmelighed. I dette kapitel bliver programmets struktur gennemgået efterfulgt af en gennemgang af programmets funktionaliteter.

10.1 Programstruktur

Løsningen er designet ud fra objektorienterede principper i programmeringssproget C#. Selve systemet er delt op i to uafhængige hovedsystemer: stemmeafgivning og vælgeridentifikation. For at gøre det muligt at udvide EVS er disse to moduler specificeret som *interfaces*, hvilket betyder at den konkrete implementering kan ændre sig, da et interface blot specificerer den nødvendige funktionalitet, som kan implementeres af en konkret løsning. Dette gør det f.eks. muligt at udskifte det nuværende databasesystem med et andet, ved kun at implementere database-interfacet.

10.1.1 Kandidater og lister

I et valgsystem skal der være mulighed for både at kunne stemme på en bestemt kandidat, et parti eller at stemme blankt. I EVS er alle former for stemmer og kandidater repræsenteret som en datastruktur med navnet **ElectoralCandidate**, der indeholder et ID, navn og liste (se figur 10.1). Dette betyder, at en blank stemme, selve partilisterne og de enkelte kandidater, hver har sit eget kandidat-objekt. En “blank” stemme er derved også repræsenteret som et **ElectoralCandidate** objekt, her er navnet og partilisten blot en nulværdi. Ved en stemme på et parti er kun navnet en nulværdi, mens at listen ikke er nul. Ved en stemme på en personlig kandidat, er navnet ikke en nulværdi.

Type	Navn	Beskrivelse
int	Id	Et unikt numerisk ID der identificerer kandidaten
string	Name	Kandidatens navn (eks. “Per Stig Møller”)
string	Liste	Kandidatens liste (eks. “C”)

Tabel 10.1: *Strukturen af `ElectoralCandidate`*

Ligeledes bliver hver partiliste repræsenteret med en lignende datastruktur, der indeholder listenavn og partinavn (se figur 10.2).

Type	Navn	Beskrivelse
string	Name	Listenavnet (eks. “C”)
string	Party	Partinavnet (eks. “Det Konservative Folkeparti”)

Tabel 10.2: *Strukturen af `ElectoralList`*

10.1.2 Stemmeafgivning

Stemmeafgivning håndterer stemmeafgivning, optælling og de forskellige kandidater der kan stemmes på. Dette interface hedder i EVS `IVotes`, og specificerer de følgende metoder og felter:

Metode	Beskrivelse
<code>IEnumerable<ElectoralCandidate> Candidates()</code>	Returnerer valgkandidaterne
<code>IEnumerable<ElectoralList> Lists()</code>	Returnerer valglisterne
<code>void Vote(ElectoralCandidate)</code>	Afgiver en stemme på en bestemt kandidat
<code>IEnumerable<KeyValuePair<ElectoralCandidate, int>> Tally()</code>	Optæller de afgivne stemmer
<code>Stream GetStream()</code>	Åbner databasen som en data-strøm sådan at den eksempelvis kan sendes via et netværk
<code>void MergeFromStream(Stream)</code>	Fletter en datastrøm fra en anden database ind i denne database

Tabel 10.3: *Metoder defineret af `IVotes`*

Selve implementeringen af dette interface kan variere, så længe at den dækker disse metoder. Dette giver en stor mængde frihed i forhold til den underliggende opbevaringsmetode. Dog overføres databasen med metoderne `GetStream` og `MergeFromStream`, hvilket forudsætter at både server og klient bruger den samme slags indkodning. Implementeringen, der benyttes i EVS, bruger SQLite til at lagre stemmerne (se afsnit 7).

10.1.3 Vælgeridentifikation

Vælgeridentifikationsmodulet fungerer på en lignende måde. Dette interface er **IVoters**, som indeholder de følgende metoder og felter:

Metode	Beskrivelse
<code>bool HasVoted(string)</code>	Returnerer om en vælger har stemt
<code>void Vote(string)</code>	Forbruger en vælgers stemme

Tabel 10.4: *Metoder defineret af IVoters*

Både stemmeafgivnings- og vælgeridentifikation-interfacet har en `vote` metode. Dette skyldes, at der differentieres mellem at afkaste en stemme og at forbruge vælgerens stemme, da dette sker i to forskellige systemer. Ethvert kald til `vote` i det ene system skal derfor medføre et tilsvarende metodekald i det andet system.

Da funktionaliteten for klient og server ikke er ens, implementeres de individuelle interfaces forskelligt i klasserne brugt på klient og server siden (se figur 10.1).

10.1.4 Databaseimplementering

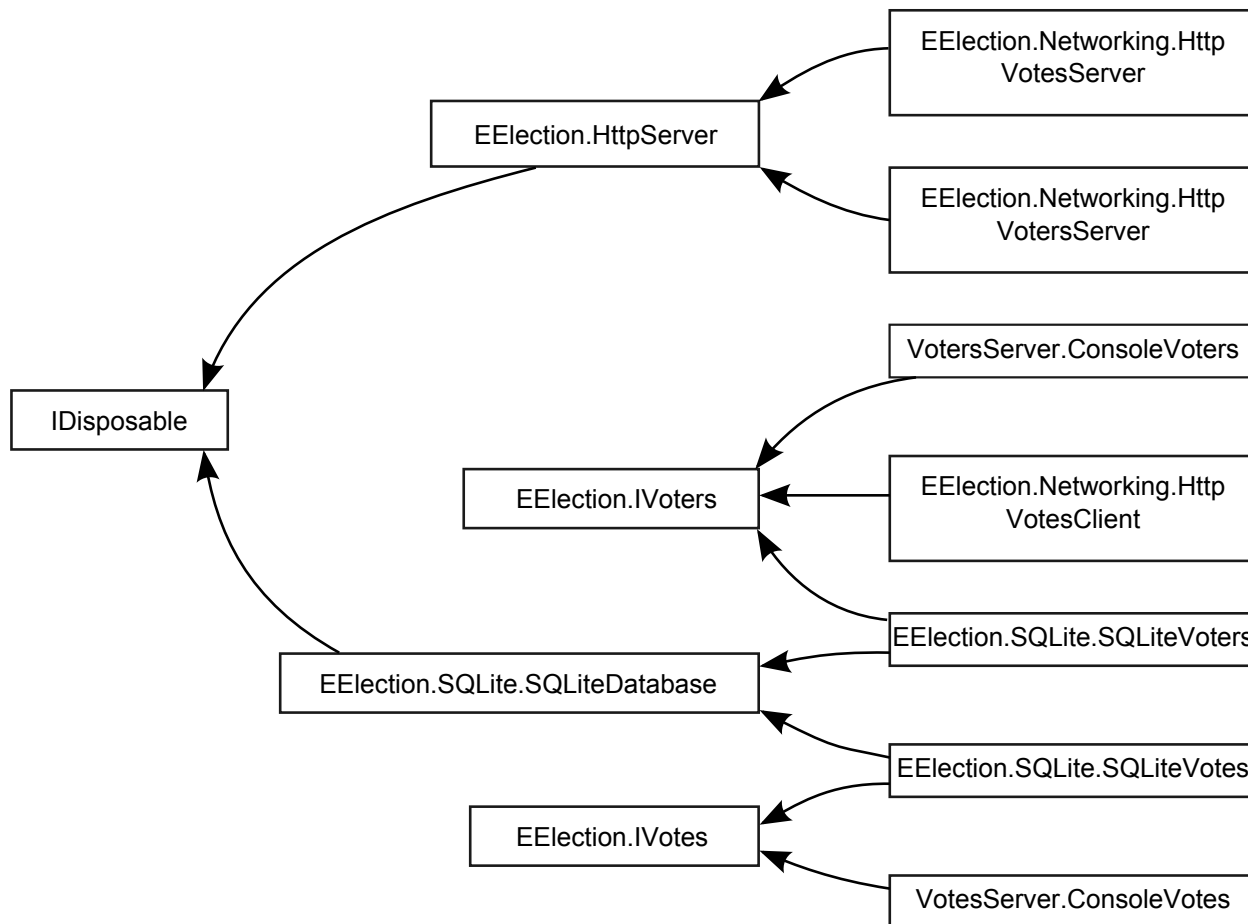
På det mest basale niveau er det nødvendigt at håndtere valgdata på en lokal database. Der findes flere forskellige databasesystemer til sådanne opgaver, men EVS bruger SQLite databasesystemet (se kapitel 7). Dette er implementeret med **SQLiteVoters** og **SQLiteVoter**, som henholdsvis implementerer de førnævnte **IVoters** og **IVotes**. Da disse klasser desuden har meget fælles funktionalitet, arver de begge fra base-klassen **SQLiteDatabase**.

For at gøre det muligt at kommunikere med en SQLite database i C# bruges `System.Data.SQLite`, en open source adapter til SQLite. For at ændre SQLite databasens tilstand eksekveres SQL kode på den. Dette gøres ved at først forberede *prepared statements*, for så senere at eksekvere dem med parameter, når de skal bruges. Kodeeksempel 10.1 fra **SQLiteVotes** demonstrerer, hvordan at `getCandidates` initialiseres med SQL forespørgslen.

Kodeeksempel 10.1: *SQL Kandidat initialisering*

```
1  getCandidates = new SQLiteCommand("SELECT * FROM candidates",  
    Connection);
```

Senere eksekveres dette prepared statement, sådan at alle kandidater bliver hentet (se figur 10.2).



Figur 10.1: Nedarvningsdiagram for EVS. Klasserne ved bunden af pilene implementerer klasserne for enden af den tilsvarende pil.

Kodeeksempel 10.2: *ReadCandidate*

```

1 public IEnumerable<ElectoralCandidate> Candidates
2 {
3     get
4     {
5         using (SQLiteDataReader reader = getCandidates.ExecuteReader())
6             while (reader.Read())
7                 yield return ReadCandidate(reader);
8     }
9 }

```

Hver metode i `IVoters` og `IVotes` har så en tilsvarende `SQLiteCommand`, som eksekveres når metoden kaldes. De fleste af disse er simple `SELECT` kommandoer, der returnerer en tabel i databasen ligesom i eksemplet før. Der er dog to mere avancerede SQL forespørgsler, som bliver brugt til indsættelse og optælling. Disse vil blive forklaret senere i dette afsnit (se

overskrifterne **indsættelse** og **optælling**).

Databasestruktur

De brugte databaser i løsningen er delt op i to dele, henholdsvis en vælgerdel og en stemmedel. Vælgerdelen indeholder informationen om de gyldige vælgere, og om disse har stemt eller ej. Stemmedelen indeholder den information der skal til for at kunne afgive og lagre en stemme. Den indeholder en tabel over gyldige kandidater samt dertilhørende lister og en tabel til de afgivne stemmer. Strukturen på disse tabeller afspejler `ElectoralCandidate` og `ElectoralList`, som er nævnt i afsnit 10.1.1. Vælgerdelen er konstrueret ud fra SQL forespørgsel som ses i kodeeksempel 10.3.

Kodeeksempel 10.3: SQL Vælgerdel

```
1 CREATE TABLE IF NOT EXISTS voters (id TEXT PRIMARY KEY, voted  
   INTEGER);
```

Her angiver *id*, som er en tilfældigt genereret unik primær nøgle, den gyldige kode og *voted* angiver en boolsk værdi, der definerer om koden er brugt til at stemme eller ej.

Stemmedelen er ligeledes konstrueret ud fra en SQL forespørgsel i kodeeksempel 10.4.

Kodeeksempel 10.4: SQL Stemmedel

```
1 CREATE TABLE IF NOT EXISTS candidates (id INTEGER PRIMARY KEY, name  
   TEXT, list TEXT);  
2 CREATE TABLE IF NOT EXISTS lists (name TEXT PRIMARY KEY, party TEXT  
   UNIQUE);  
3 CREATE TABLE IF NOT EXISTS votes (id INTEGER, name TEXT, list TEXT);
```

I stemmedelen er der som før nævnt tre dele. *candidates* tabellen definerer den gyldige liste af kandidater til valget og er bestående af et id, kandidaternes navn og listen (se figur 10.2). Kandidatens ID er unikt, og er med for at kunne skelne eventuelt identiske kandidater (kandidater med samme navn og liste) fra hinanden.

ID	Name	List
14	Sophie Hæstorp Andersen	A
32	Liv Holm Andersen	B

Figur 10.2: Eksempel på *candidates* tabel

Tabellen *lists* definerer ligesom *candidates* en liste af gyldige lister, eller partier, til valget og er bestående af listernes navn samt partibogstav (se figur 10.3). Navn og partibogstav er dog i modsætning til kandidater unikt, da to partier skal have forskelligt navn og partibogstav.

Name	Party
Socialdemokraterne	A
Det Radikale Venstre	B

Figur 10.3: *Eksempel på lists tabel*

votes tabellen er en tabel, hvor afgivne stemmer kan oplagres, tabellen er strukturmæssigt identisk med *candidates* tabellen, dog er id ikke en primær nøgle, da det skal være muligt for en kandidat at have modtaget flere stemmer. Udover de kolonner, der her er angivet, har hver tabel også en kolonne med *rowid*, som angiver rækkens nummer. Denne kolonne er unik og bruges normalt ikke eksplicit. Til indsættelse af stemmer angives der dog en værdi til denne.

Indsættelse

For at sikre stemmehemmeligheden, må det ikke være muligt at se, rækkefølgen stemmerne er blevet afgivet. I SQLite har hver række et unikt tal, som bestemmer rækkefølgen af de forskellige rækker. Dette tal, kaldet *rowid*, er derfor som standard i den samme rækkefølge at stemmerne blev indsat, hvilket naturligvis afslører, rækkefølgen stemmerne bliver afgivet i. Derfor gives hver stemme et vilkårligt *rowid*. SQL forespørgslen, der indsætter en stemme, ses i kodeeksempel 10.5.

Kodeeksempel 10.5: *SQL Stemmeafgivning*

```
1 INSERT INTO votes(rowid, id, name, list) VALUES (random(),?,?,?)
```

Da *rowid* er 64 bit stort, giver dette $2^{64} \approx 1.8 \times 10^{19}$ forskellige kombinationer. I det usandsynlige tilfælde, at der er en kollision mellem to *rowid*, prøves der med et nyt vilkårligt *rowid* indtil, at der findes et som ikke er brugt. På denne måde sikres det at rækkefølgen af rækkerne ikke svarer til rækkefølgen, som stemmerne af afgivet i.

For at kunne modtage stemmer fra flere valgmaskiner (se sektion 10.1.5), er det nødvendigt at kunne “flette” stemmedatabaser sammen. Derfor findes metoden `MergeFromStream`, som først gemmer en datastrøm til en fil og åbner den som en database, for så at tilføje alle stemmerne fra den database til sin egen. Dette gøres desuden i en transaktion (se sektion 7.1), for at formindske tiden, det kræver at udføre det.

Kodeeksempel 10.6: *SQL Stemmesamling, med transaktion*

```
1 public void MergeFromStream(Stream source)
2 {
3     using (FileStream temp = new FileStream("temp.sqlite",
4         FileMode.Create, FileAccess.Write))
5         source.CopyTo(temp);
6
7     using (SQLiteVotes other = new SQLiteVotes("temp.sqlite"))
8     using (SQLiteTransaction transaction =
9         Connection.BeginTransaction())
10    {
11        foreach (ElectoralCandidate vote in other.Votes)
12            Vote(vote);
13    }
14 }
```

I kodeeksempel 10.6 ses det, at en datastrøm gemmes til en midlertidig databasefil, databasefilen åbnes og alle stemmer tilføjes. Til sidst kaldes `Commit` metoden, der tilføjer transaktionen til databasen.

Optælling

For at lave optællingen af stemmerne, eksekveres en forespørgsel, som går igennem hver kandidat og tæller alle stemmer, hvor at ID'et stemmer overens. Dette gøres med den SQL forespørgslen som ses i kodeeksempel 10.7.

Kodeeksempel 10.7: *SQL Optælling*

```
1 SELECT id, name, list, (SELECT count(*) FROM votes WHERE votes.id
   == candidates.id) AS votecount FROM candidates ORDER BY
   votecount DESC
```

Her vælges alle kandidater samt antallet af stemmer for hver kandidat. Kandidaterne bliver desuden returneret i faldende sorteret orden, sådan at de kandidater med flest stemmer fremkommer først.

10.1.5 Netværskommunikation

For at kunne fordele EVS ud over flere valgcomputere, er det nødvendigt at implementere en eller anden form for netværskommunikation. Dette er gjort gennem `HttpVotersServer` og

`HttpVotesServer`. Ligesom med SQLite implementeringen har disse to klasser noget fælles funktionalitet og arver derfor fra `HttpServer`.

Vælgeridentifikation og stemmeinformation fungerer på to vidt forskellige måder i EVS. Vælgeridentifikation bliver opbevaret på en central server (eksempelvist en server pr. valgsted), og klienten spørger løbende om denne vælgeridentifikation, for at sikre at ingen stemmer mere end én gang. Hvis stemmerne sendes på en lignende måde, ville det for serveren være tydeligt at se, hvem der har stemt på hvad, da begge forespørgsler vil sendes inden for kort tid af hinanden. Derfor sendes stemmerne først ved afslutningen af valget. På grund af disse forskelle har `HttpVotersServer` en tilsvarende `HttpVotersClient`, hvilket `HttpVotesServer` ikke har.

For at understøtte at begge servere køres på én maskine samtidigt, er det nødvendigt at bruge en form for trådning. Dette er gjort ved hjælp af såkaldte asynkrone metoder, som er en nyere funktion i C#¹. Asynkrone metoder betyder, at eksekveringen af en metode kan startes, så udføres noget andet arbejde, for så at få svar når den asynkrone metode, når denne er eksekveret. Eksempelvis startes Http serveren op som i kodeeksempel 10.8 (fra `HttpServer`).

Kodeeksempel 10.8: *HttpServer.Start*

```
1 public void Start()
2 {
3     running = true;
4     Listener.Start();
5     Task.Run( async () =>
6     {
7         while (true)
8             OnContext(await Listener.GetContextAsync());
9     }, cancellation.Token);
10 }
```

Her startes en listener (det objekt der opfanger Http forespørgsler) og en separat tråd startes asynkront. Udover dette tilføjes der også et såkaldt *cancellation token*, hvilket gør det muligt at stoppe den anden tråd.

Når en HTTP forespørgsel så bliver modtaget af `HttpVotersServer` eller `HttpVotesServer`, så behandles det af det tilsvarende objekts `OnContext` metode. Hvilken handling en forespørgsel medfører, afhænger af dens metode. Det skal understreges, at en metode i HTTP terminologi ikke er det samme, som det er i C# terminologi. I HTTP terminologi er en metode en tekststreng, der betegner den ønskede handling.

EVS arbejder kun med de minimalt krævede HTTP metoder: *GET* og *POST*. *GET* skal ifølge HTTP specifikationen kun returnere data, og bør derfor ikke ændre tilstanden på

¹Siden C# 4.5

serveren. *POST* er derimod specificeret til at ændre tilstanden for en ressource på serveren, med en anden mængde data som forespørgslen indeholder.

Dette betyder eksempelvis at `HttpVotersServer` behandler en *GET*, som en forespørgsel om en vælger har stemt, mens at den behandler en *POST* som en stemmeafgivning. Kodestykke 10.9 viser, hvordan at der skelnes mellem disse to metoder.

Kodeeksempel 10.9: HTTP POST/GET metoder

```
1 // When voting
2 if (request.HttpMethod == "POST")
3 {
4     voters.Vote(id);
5     response.StatusCode = (int)HttpStatusCode.NoContent;
6 }
7 else if (request.HttpMethod == "GET")
8 {
9     // When checking if id has voted
10    if (voters.HasVoted(id))
11    {
12        // If id has voted send "true" through outputStream
13        response.StatusCode = (int)HttpStatusCode.OK;
14        using (response.OutputStream)
15            response.OutputStream.Write(trueData, 0, trueData.Length);
16    }
17    else
18    {
19        // If id has not voted send "false" through outputStream
20        response.StatusCode = (int)HttpStatusCode.OK;
21        using (response.OutputStream)
22            response.OutputStream.Write(falseData, 0, falseData.Length);
23    }
24 }
25 else // BadRequest: server has no response
26     throw new ArgumentException("Ikke en gyldig serverhandling");
```

Her undersøges hvilken metode, der bruges af HTTP forespørgslen. Da HTTP ikke har nogen måde til, at sende boolske sand/falsk-værdier som svar, sendes strengen “true” eller “false” i stedet for. Hele dette kodestykke er desuden inde i en try/catch-blok hvilket gør, at serveren returnerer passende fejlkoder i tilfælde af fejl.

For at overføre kandidater, lister og stemmer, sendes hele databasen med stemmerne i en HTTP POST forespørgsel. Klienten skal ved opstart hente de mulige kandidater og lister, som der kan stemmes på. Dette gøres ved at sende en *GET* forespørgsel til serveren, som så svarer, ved at sende en “skabelon” af valget, der indeholder alle kandidater og lister men ingen stemmer. Filen åbnes så af klienten og bruges som database. Efter endt valg lukkes databasen

og sendes så via en *POST* forespørgsel til serveren. Når serveren modtager databasen, gemmes denne som en fil, og alle stemmer fra den modtagne database tilføjes til serverens eksisterende database.

10.2 Systemgennemgang

I dette afsnit vil der komme en teoretisk gennemgang af et valg på et enkelt valgsted. Det vil blive forklaret, hvilke opgaver der skal udføres, og hvordan dette gøres i forhold til løsningens metoder og klasser. Samtidigt vil der også være en forklaring af netværksdelen under valget, her vil det være illustreret og forklaret i en trinvis gennemgang, der viser hvordan kommunikationen mellem klient og server sker ved valget.

Løsningens elektroniske valgsted er bygget op af én eller flere valg-terminaler og én eller to servere. Serverne kan køres på en maskine, der både holder øje med vælger-identifikation og stemme-modtagelse samt opsætning af valg-terminaler. Serverdelen kan dog også deles mellem to maskiner, så arbejdsopgaven deles i to; en identifikations-server og en stemme-server.

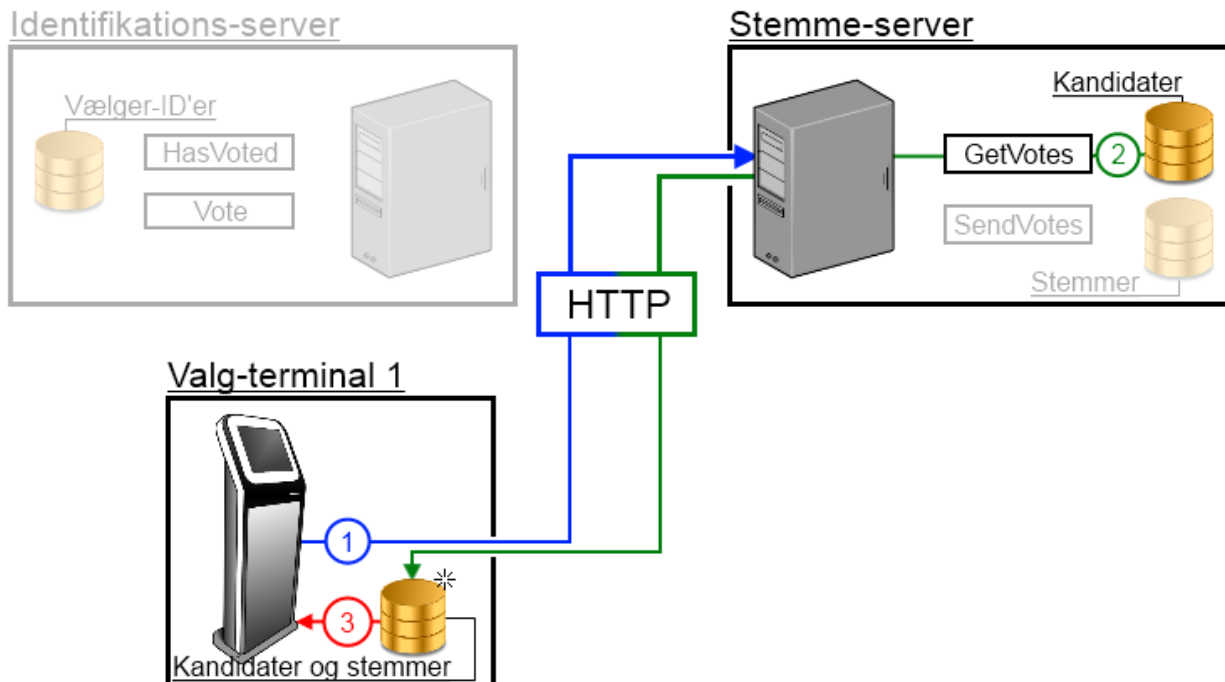
Identifikations-serveren står for vælger-validering, og er illustreret i figur 10.5. Stemme-serveren står for opsætning af valg-terminaler, samt indsamling og optælling af stemmer, disse er illustreret i figur 10.4 og figur 10.6.

10.2.1 Inden valget

Inden valget er det nødvendigt at opsætte databaserne på både server og klienter. Serveren indeholder en liste over de gyldige kandidater, og hvilke lister de tilhører. Denne liste indlæses ved hjælp af `GetCandidates` og `GetLists` metoderne. Serveren skal desuden have oprettet en database med gyldige vælgerID'er, som skal bruges af de stemmeberettigede under valget. Disse oprettes ligesom kandidater og lister fra en fil med et forudbestemt format beliggende på serveren. ID'erne indlæses via `GetIDs` metoden. Klientterminalerne opstartes med en åbningskode, og der vælges en nedlukningskode, som skal indtastes i vælgerID feltet, for at lukke maskinen når valgstedet skal lukke. Klienterne skal desuden vide, hvilke kandidater og lister, der er til valget. Hertil kaldes `GetVotes` metoden, der opretter forbindelse til serveren og anmoder om, at få tilsendt en databaseskabelon indeholdende kandidater, lister og en tabel til stemmer. Når en vælger stemmer, og er blevet godkendt af identifikations-serveren, lagres stemmen i denne skabelon-database.

Opsætningen af en enkelt valg-terminal er illustreret i figur 10.4. Her følger forklaringen af figuren:

1. Valg-terminalen kontakter valgstedets stemme-server og anmoder om de nødvendig oplysninger.
2. Stemme-serveren sender en database-skabelon tilbage til terminalen. Denne skabelon indeholder de lister og kandidater, som vælgerne på valgstedet kan stemme på.
3. Terminalen gemmer denne skabelon på sin harddisk og indlæser kandidaterne og partierne i sin hukommelse.



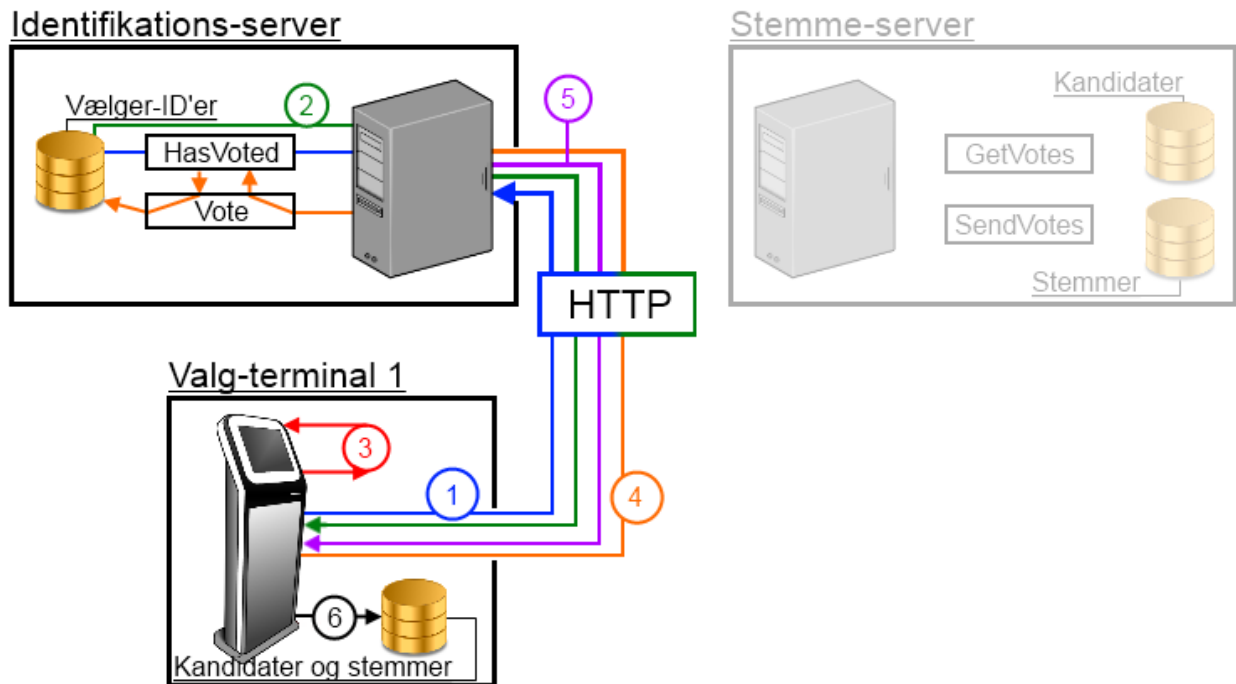
Figur 10.4: Opsætningsprocessen for den enkelte valg-terminal, trinvist illustreret

10.2.2 Under valget

Under valget er den enkelte stemme-terminal ansvarlig for at håndtere de afgivne stemmer, samt at få vælgerens ID valideret inden stemmeafgivelsen finder sted. Dette sker ved at vælgeren tilgår maskinen med sit Vælger-ID, dette ID bliver via `HttpVoters.HasVoted` metoden, valideret på serveren og klienten, ved nu om ID'et er brugt. Er ID'et gyldigt, får vælgeren lov til at vælge den ønskede valgkandidat eller liste, som vedkommende ønsker at afgive sin stemme til. Vælgerens valg afgives nu til databasen ved, at kalde `Vote` metoderne for både `HttpVoters` og `SQLiteVotes`, som opretter et kandidat objekt med vælgerens valg på klienten, men også forbruger vælgerens ID på identifikations-serveren.

Identifikations-processen kan beskrives ud fra de nummererede trin vist på illustrationen i figur 10.5.

1. Hver vælger har en unik kode, som de indtaster på en valg-terminal. Valg-terminalen kontakter herefter valgstedets identifikations-server og spørger om, den indtastede kode er gyldig og ubrugt. Identifikations-serveren afgør om, en kode er gyldig ved at kigge i databasen med gyldige koder, der kan bruges.
2. Findes koden, og er den ubrugt, sender serveren et positivt svar tilbage til valg-terminalen. Findes koden ikke, eller er den allerede blevet brugt, sender serveren et negativt svar tilbage.
3. Hvis koden var gyldig, giver valg-terminalen vælgeren lov til, at vælge hvad han/hun vil stemme på.



Figur 10.5: Valgproces for den enkelte vælger, trinvist illustreret

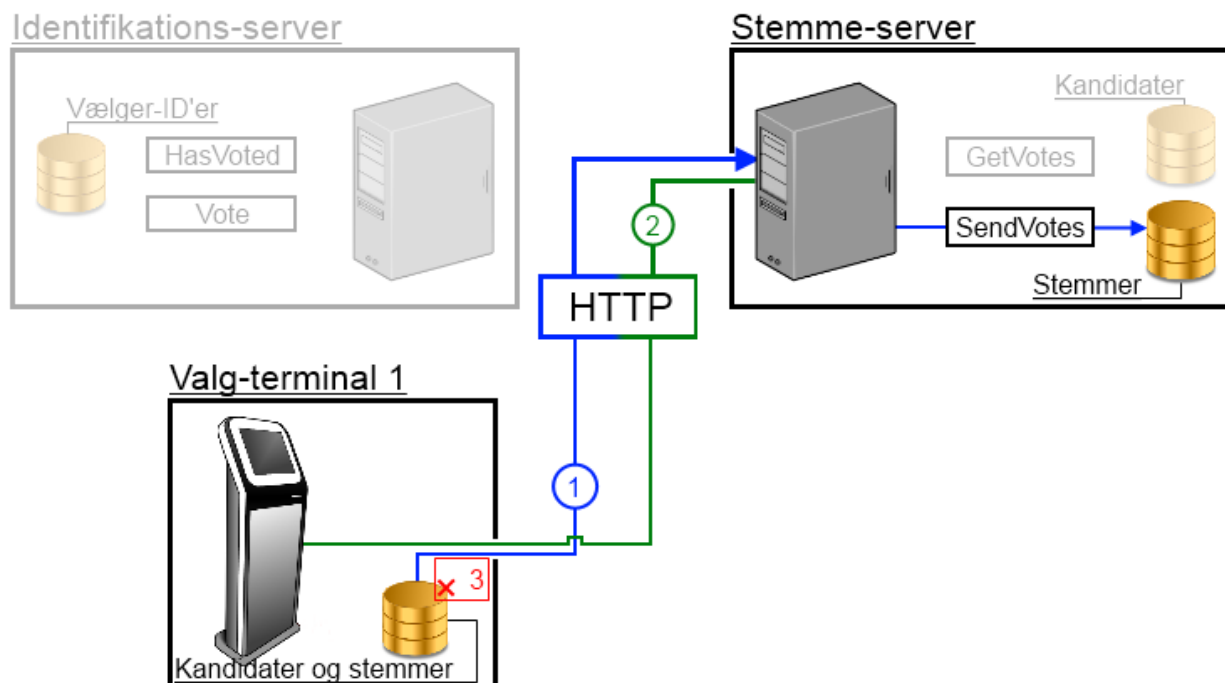
4. Når vælgeren har besluttet sig, bliver deres kode igen sendt til identifikations-serveren. Denne gang bliver koden ikke bare tjekket som før, men også krydset af i database over koder som værende brugt, hvis koden på forhånd var gyldig og ubrugt.
5. Igen bliver valg-terminalen oplyst om situationen.
6. Bliver koden godkendt og krydset af på serveren, bliver vælgeres stemme lagret på valg-terminalen, og valghandlingen for vælgerens vedkommende er færdig.

Nu er vælgerens valghandling overstået, og han kan forlade klientterminalen. Terminalen nulstiller sig selv, så den nu er klar til den næste vælger. Database med stemmer er selvfølgelig intakt og bliver udvidet hver gang, en vælger foretager en stemmehandling på terminalen.

10.2.3 Efter valget

Efter valget lukkes alle klientterminalerne med den nedlukningskode, som blev valgt ved opstart. Ved nedlukning af klientterminalen videresendes stemme-database fra hver enkelt klient til serveren via `SendVotes` metoden. Alle stemmerne, der modtages på serveren, skal sammensættes til en enkelt database, så det er nemmere at tælle stemmerne sammen. Dette gøres via `MergeFromStream` metoden, som sammenfletter data modtaget fra klienten i en samlet databasefil på serveren. Dette gøres alt sammen i en enkelt transaktion, som sikrer at kun en database sammenflettes af gangen, på trods af at flere klienter kan blive lukket samtidigt. Samtidigt er dette med til, at øge hastigheden dramatisk, hvilket er gavnligt ved store valg, hvor mange stemmer skal behandles.

Nedlukningsprocessen er illustreret trinvist i figur 10.6, hvor der sker følgende:



Figur 10.6: Nedlukningsprocessen for den enkelte valg-terminal, trinvist illustreret

1. Idet en valg-terminal lukkes, sender den sin skabelon-database, der indeholder alle terminalens stemmer, til valgstedets stemme-server. Stemme-serveren fletter så terminalens stemmer med sin egen database med stemmer.
2. Hvis alt gik godt, sender stemme-serveren et positivt svar tilbage til valg-terminalen.
3. Terminalen sletter nu sin database med stemmer. Dette gøres for at sikre, at de ikke ved en fejl skulle blive sendt til stemme-serveren igen. Herefter lukkes terminalen.

Efter at alle stemmerne fra valgstedet er samlet i én database på serveren, kan stemmerne enten tælles op via Tally metoden eller sendes videre i systemet ved hjælp af førnævnte proces.

10.2.4 Valgafbrydelser

Under valget kan der ske uventede ting, som kan betyde afbrydelser af hele eller dele af valget. Dette kan gøre at en eller flere maskiner går ned under valget eller må genstartes. Hertil er det nødvendigt, at en maskine kan genstartes uden, at der herved mistes data. Ved normal initialisering af både klient og server overskrives eventuelle eksisterende databaser med nogle nye, for at undgå "levn" fra tidligere databaser. Det er dog imidlertid muligt at initialisere både klient og server undtagelsesvist og bruge de allerede eksisterende databasefiler. På denne måde kan valget fortsætte, hvor det slap og ingen data vil være gået tabt. Der kan dog fra

klientens side ikke genskabes efter en vellykket nedlukning, da terminalens stemmer bliver slettet, efter at serveren har givet til kende, at de er overført (se afsnit 10.2.3).

For at validere funktionaliteten af løsningsforslaget, er der blevet udviklet nogle tests, der skal teste de forskellige bestanddele af programmet. Disse tests vil blive forklaret og gennemgået i næste kapitel.

Kapitel 11

Test

Test af systemet er essentielt for den troværdighed systemet har. Da denne prototype, er en demonstration af en løsning for e-valg, står og falder meget af pålideligheden til systemet med, om systemet egentlig virker. For at være sikker på, at systemet ikke havde fejl, blev der taget udgangspunkt i at systemet havde fejl, da de første tests blev foretaget. Det resulterede i at programmet blev testet med det formål at finde de fejl, og at disse skulle rettes senere for at lave systemet fejlfrit[1]. Til at teste om programmet virkede efter hensigten, er der blevet foretaget en række unit tests af de forskellige programdele, efterfulgt af en Black Box test af det komplette system. For at isolere de enkelte dele, er de elementer af programmet, som ikke bliver testet, erstattet af “dummy”-værdier - metode kald til andre klasser, som returnerer hardcodede værdier fremfor at kalde metoden. Elementerne kaldes for stubbe[46][1]. En stub¹ er en substitution af et eller flere objekter og tilbyder et tilregneligt grundlag at teste med. Stubbene indsættes, så den del af koden, som man vil teste, bliver isoleret og ikke afhænger af andre programdele. Der testes på tre programdele, hvor hver del bliver testet, med de to andre som stubbe. I netværkstesten er det klasserne `HttpVotersServer` og `HttpVotersClient`, der bliver testet. I den første database test bliver `SQLiteVotes`-klassen testet, og i den tredje del er det den anden database klasse, `SQLiteVoters`, der testes. Til at foretage unit testene, er programmet NUnit version 2.0 blevet benyttet[9]. Koden til de tests, der er foretaget, kan findes som helhed i de vedlagte bilag. De viste kode-eksempler i dette kapitel er kun udvalgte dele af de tests, der er blevet foretaget.

11.1 Netværkstest

Selve netværksklasserne `HttpVotersClient` og `HttpVotersServer` er blevet unit testet ved brug af følgende stub. Fordi klassen `TestVoters` implementerer interfacet `IVoters`, kan den benyttes sammen med klient-programmet.

¹*Engelsk:* Mock object

Kodeeksempel 11.1: *TestVotersClass*

```
1 private class TestVoters : IVoters
2 {
3     public bool HasVoted(string id)
4     {
5         return id[0] == '0';
6     }
7
8     public void Vote(string id)
9     {
10        if (HasVoted(id))
11            throw new ArgumentException();
12    }
13 }
```

Klassen `TestVoters` blev unit testet for tre forskellige scenarier, henholdsvis om en bruger har stemt, ikke har stemt og om der kastes en “exception”, når der stemmes med et allerede brugt ID. Et udsnit af klassen vises i listing 11.1, her ses at metoderne `HasVoted` og `Vote` er blevet erstattet med hardcodede værdier. De to metoder returnere nu boolske værdier eller en exception, da klassen kun skal teste om netværks programmeringen virker efter hensigten.

Kodeeksempel 11.2: *Http TestCases*

```
1 [TestCase]
2 public void SingleVoterVote()
3 {
4     HttpVotersClient client = new
5         HttpVotersClient("http://localhost:8080/");
6     Assert.IsTrue(client.HasVoted("0111110101"));
7 }
8 [TestCase]
9 public void SingleVoterNoVote()
10 {
11     HttpVotersClient client = new
12         HttpVotersClient("http://localhost:8080/");
13     Assert.IsFalse(client.HasVoted("111110101"));
14 }
15 [TestCase]
16 public void IllegalVote()
17 {
18     HttpVotersClient client = new
19         HttpVotersClient("http://localhost:8080/");
20     Assert.Catch(() => { client.Vote("011110101"); });
21 }
```

I kodeeksempel 11.2, vises de tre testscenarier. I `SingleVoterVote()` testes en id streng, der starter med 0, for at repræsentere et id, som allerede har været brugt. Returværdien af `database.HasVoted()`, som forventes at være true på strengen "0111110101", bliver så tjekket af `Assert.IsTrue()`, for at bekræfte, at den forventede værdi bliver returneret. `SingleVoterNoVote()` sender den samme forespørgsel med en ubrugt kode. Eftersom strengen i denne kode ikke starter med 0, er den forventede returværdi "false". Derfor bliver denne returværdi tjekket af `Assert.IsFalse()`-metoden. `IllegalVote()` prøver at kalde `database.Vote()`-metoden med en brugt kode. Dette skulle resultere i en "exception", som bliver tjekket af `Assert.Catch()`. Alle tre netværkstests returnerer de forventede værdier. HTTP forespørgslerne bliver sendt rigtigt. Netværkss kommunikationen må derfor, på baggrund af denne test, virke korrekt.

11.2 Databasetest - Stemmer

Systemets to databaser er også blevet unit testet. Databasen som indeholder kandidater, lister og stemmer, er i testen beskrevet med følgende klasse:

Kodeeksempel 11.3: *SQLVotesTestClass*

```
1 public class SQLVotesTest
2 {
3     SQLiteVotes database;
4     List<ElectoralCandidate> candidates;
5     Dictionary<ElectoralCandidate, int> voteCount;
6
7     ...
8
9     database = new SQLiteVotes("test.sqlite", candidates, lists);
10    candidates = new List<ElectoralCandidate>(database.Candidates);
11    voteCount = candidates.ToDictionary(x => x, x => 0);
12 }
```

Testens klasse, vist i kodeeksempel 11.3, opretter en `SQLiteVotes`-database, samt en liste til kandidaterne. Disse bruges i testene, hvor databasen bliver opdateret med en eller flere stemmer. Herefter tjekkes det med `CollectionAssert.AreEqual` om `voteCount`, der er den faktiske samling af kandidater og deres stemmer, svarer til samlingen af kandidater og stemmer, som er lagret i databasen. Samlingen af kandidater og stemmer i databasen tælles op af `database.Tally()`-metoden.

Første unit test tester med kun en enkelt afgivet stemme. Først stemmes der, så tæller `voteCount` en op, efterfulgt af et tjek for om `Tally()` også tæller én stemme i databasen. Metoden til denne test kan ses i kodeeksempel 11.4.

Kodeeksempel 11.4: *SingleVote TestCase*

```
1 [TestCase]
2 public void SingleVote()
3 {
4     database.Vote(candidates[0]);
5     voteCount[candidates[0]]++;
6     CollectionAssert.AreEqual(voteCount, database.Tally());
7 }
```

Den anden unit test tester det scenarie, hvor flere stemmer afgives på én kandidat (se kodeeksempel 11.5). Der bliver her afgivet 40 stemmer. `AreEquivalent()` tester igen, at samlingen af faktiske kandidater og deres stemmer svarer til databasens samling.

Kodeeksempel 11.5: *SingleCandidate TestCase*

```
1 [TestCase]
2 public void SingleCandidate()
3 {
4     for (int i = 0; i < 40; i++)
5     {
6         database.Vote(candidates[0]);
7         voteCount[candidates[0]]++;
8     }
9
10    CollectionAssert.AreEqual(voteCount, database.Tally());
11 }
```

I den tredje unit test bruges en række forskellige værdier til, at teste lagringen af stemmer. Hver TestCase giver `RandomCandidates()` to heltals-parametre, der hhv. er antallet af stemmer, som skal sættes, og en værdi, som bruges til at oprette en tilfældighedsgenerator med `Random(int Seed)`. Denne tilfældighedsgenerator bruges i for-løkken til at udvælge en tilfældig kandidat. Igen bliver `AreEquivalent()` kaldt for at bekræfte, at den egentlige samling af stemmer, `voteCount`, og samlingen af registrerede stemmer i databasen, `database.Tally()` er ækvivalente. Dette konkrete eksempel vises i kodeeksempel 11.6.

Kodeeksempel 11.6: *RandomCandidates TestCases*

```
1
2 [TestCase(40, 12)]
3 [TestCase(60, 432434)]
4 [TestCase(40, 22222222)]
5 [TestCase(40, 123132)]
6 [TestCase(15, 2123)]
7 [TestCase(10, 10245)]
8 public void RandomCandidates(int votes, int n)
9 {
10     Random rand = new Random(n);
11     for (int i = 0; i < votes; i++)
12     {
13         ElectoralCandidate candidate =
14             candidates[rand.Next(candidates.Count)];
15         database.Vote(candidate);
16         voteCount[candidate]++;
17     }
18
19     CollectionAssert.AreEqual(voteCount, database.Tally());
20 }
```

Denne database opfører sig, som den skal. Metoden `Vote` stemmer korrekt i databasen, hvor `Tally()` tæller stemmerne korrekt op. Dette fører til, at de enkelte kodedele til denne database opfører sig, som forventet. Stemmerne kan altså lagres i databasen og opgøres af de metoder, som er designet til det. Ingen af de udførte test foretager nogen uventede handlinger.

11.3 Databasetest - Vælgere

Database-typen `SQLiteVoters` bliver unit testet uden, at involvere netværkskommunikationen og `SQLiteVotes`. Klassen `SQLVotersTest` er sat op til dette formål (se kodeeksempel 11.7). Til dette bliver der lavet et `SQLiteVoters` objekt og en liste af ID'er, som skal håndtere de koder, der bliver testet. Samtlige ID'er er streng-værdier.

Kodeeksempel 11.7: *SQLVotersTestClass*

```
1 public class SQLVotersTest
2 {
3     SQLiteVoters database;
4     List<string> ids;
5
6     ...
7
8     ids = new List<string>();
9     Random rand = new Random();
10    for (int i = 0; i < 100; i++)
11        ids.Add(string.Format("{0:0000000000}-{1:0000}",
12                               rand.Next(10000000000), rand.Next(10000)));
13
14    database = new SQLiteVoters("test.sqlite", ids);
15 }
```

I den første unit test af `SQLiteVoters` (se kodeeksempel 11.8), tester `Assert.IsFalse` først om ID-strengen er blevet brugt. `Vote()`-metoden bliver så kaldt, for at simulere en enkelt stemmeafgivelse. Da HTTP-serveren er en stub, sættes kodens første ciffer til nul. `Assert.IsTrue` tjekker dernæst om dette lykkedes, ved at se på om `database.HasVoted()` nu returnere "true". Afslutningsvist tjekkes det at de andre koder i databasen ikke er blevet brugt.

Kodeeksempel 11.8: *SingleVoter TestCase*

```
1 [TestCase]
2 public void SingleVoter()
3 {
4     Assert.IsFalse(database.HasVoted(ids[0]));
5     database.Vote(ids[0]);
6     Assert.IsTrue(database.HasVoted(ids[0]));
7     Assert.IsTrue(Enumerable.Range(1, ids.Count - 1).All(x =>
8         !database.HasVoted(ids[x])));
9 }
```

I den anden unit test simuleres det, at alle koderne bruges én af gangen. Kodestykket kan ses i kodeeksempel 11.9. Løkken tjekker i hvert gennemløb, om koden er blevet brugt både før og efter, `database.Vote` bliver kaldt. Løkkens antal af gennemløb bestemmes af af listen `ids`, der med sin `Count` metode giver antallet af elementer i listen. Det testes derefter udenfor løkken, om alle koderne er blevet brugt.

Kodeeksempel 11.9: *MultipleVoter TestCase*

```
1 [TestCase]
2 public void MultipleVoter()
3 {
4     for (int i = 0; i < ids.Count; i++)
5     {
6         Assert.IsFalse(database.HasVoted(ids[i]));
7         database.Vote(ids[i]);
8         Assert.IsTrue(database.HasVoted(ids[i]));
9     }
10
11     Assert.IsTrue(Enumerable.Range(0, ids.Count).All(x =>
12         database.HasVoted(ids[x])));
13 }
```

Alle unit tests på ID-databasen gennemløber korrekt. `HasVoted` og `Voted` metoderne opfører sig, som de er blevet designet til. Listen med elementer af ID-strengene fungerer også som tiltænkt. Det betyder at koderne, som vælgerne indtaster, for at identificere sig som vælger til klienten, kan tilgås som planlagt. Indtastning af forkerte koder, er også blevet eksperimenteret med. Dette er yderligere beskrevet i afsnittet om Black-box testen 11.4.

11.4 Black-box test

Systemet har også gennemgået en Black-box test, hvor de enkelte funktionaliteter blev testet i sammenhæng med resten af systemet, for at afsløre defekter i systemet[2]. Systemet skal i praksis sættes op på et lukket netværk, for at virke. Dette er blevet testet med fire computere sluttet til samme switch. Tre computere virkede som klient-simulatorer, og den fjerde kørte både identifikations-serveren og stemme-serveren. Hver klient-computer blev sat op til, at kommunikere med server-computeren, og et valg kunne simuleres. Under forsøget blev koderne, som klient-computerne benyttede, korrekt valideret og afkrydset af server-computeren, og ved gentagen brug af samme kode, genererede klienten en fejlbesked som forventet. Efter at brugerfladen havde været brugt, til at simulere et valg, blev klienterne lukket, og en optælling blev foretaget af serveren. Testen med det input, systemet er sat op til, forløb altså korrekt og gav det forventede output. Systemet blev derefter udsat for mere varierede input. Alle forsøg med, at indtaste forkerte koder, gav den korrekte fejlmeddelelse. Det samme skete da koder med forskellige tegn og bogstaver blev benyttet, i stedet for tal. Både specialtegn og udvidede alfabeter, gav den korrekte output. Dernæst blev systemet testet for, hvad der skete når tingene blev tændt i forkert rækkefølge. Her opstod de første udfordringer, der blev fundet i systemet. Når klienterne blev startet før serverprogrammerne, kunne koderne ikke valideres ordenligt, og der blev kastet en “exception”. Hvis man lukkede serveren før alle klienterne havde overført blev et valg gjort forkert op, da der på dette tidspunkt ikke var mulighed for at genskabe databasen. Dette ledte til, at serverprogrammet fik implementeret en ny funktionalitet for at genbruge den gamle database ved opstart. Klient-programmet er også, af denne grund, blevet udvidet til, at genskabe stemmerne fra sidste session, hvis serveren ikke modtog dem korrekt, og klienten lukkede. Dette rettede problemet, hvilket blev valideret i en ny test.

11.5 Opsamling

Til at teste programmet, er der blevet gjort brug af stubbe, unit testing og Black-box testing. Værktøjet, som fortog testene, var NUnit 2.0, og selve testene og stubbene blev defineret i et separat projekt dedikeret til tests. I de tre dele af programmet, der blev unit testet, forløb alt som forventet. De enkelte dele af programmet fungerer derfor, som de bør. Systemet blev også testet som helhed i afsnit 11.4, og de rette værdier blev talt op, efter at de var blevet afgivet gennem klientens grafiske brugergrænseflade. Her blev der fundet en funktionalitetsfejl, som efterfølgende blev rettet. Black-box testens formål med, at vise defekter i systemet, var vellykket. Unit testen havde til formål, at vise om der var fejl i de enkelte klasser. Fraværet af fejl i klasserne og den rettede fejl i funktionaliteten, leder videre til, om løsningen kan leve op til de kravspecifikationer, der blev opstillet i afsnit 6.1. Dette vil blive diskuteret i diskussionen.

Del III

Implikationer

Kapitel 12

Diskussion

I dette afsnit diskuteres og vurderes de forskellige stadier af programmet. For at gøre gennemgangen overskuelig, deles diskussionen ind i separate afsnit, der hver behandler en enkelt del af programmet. I hver af disse afsnit diskuteres de overvejelser og beslutninger, der gik forud for den egentlige implementering. Hvorvidt de endelige beslutninger, der blev taget i projektet, fungerer, vurderes på baggrund af programmets relevante test resultater (se afsnit 11), og deres opfyldning af de opsatte kravspecifikationer, som følger:

1. Løsningen skal laves i C#. Dette er et eksternt krav til projektet.
2. Løsningen må ikke gå på kompromis med sikkerheden i forhold til nuværende valg (se afsnit 4.1).
3. Løsningen skal sikre vælgerens anonymitet.
4. Løsningen skal have en brugerflade, der ikke gør det sværere for en stemmepligtig at stemme. Alle dem, der kan stemme nu, skal også kunne stemme med det elektroniske valgsystem.
5. Løsningen skal så vidt muligt holde sig indenfor de nuværende lovmæssige begrænsninger for valg.

Intentionen med dette er, at skabe et samlet overblik over arbejdsprocessen, og de resultater projektet har udledt.

12.1 Afgivning og opbevaring af stemmer

Under afholdelsen af ethvert valg er vælgerens stemmeafgivelse essentiel. I EVS indeholder denne del af valget to hovedpointer; identifikation af en godkendt vælger, og afgivelsen af stemmen til systemet.

12.1.1 Anonymitet og identifikation

Under projektforløbet blev der diskuteret forskellige måder, at identificere en gyldig vælger. Her var det især kravet om vælgerens anonymitet, samt kravet om én stemme pr. vælger, som

kom i fokus. I starten af processen var idéen om, at have en iris-scanner ved hver terminal, samt afskærmning af indgangene til de enkelte valgbokse på valgstedet blevet overvejet. Disse blev dog hurtigt afvist. For at sikre, at den enkelte vælger kunne identificeres anonymt og samtidig kun have mulighed for, at stemme én gang, blev der i processen valgt, at identifikationen skulle benytte en unik kode. Det blev også bestemt, at denne kode skulle benyttes til, at starte selve afstemningen, og derved sikre, at der kun var én stemme tilgængelig pr. kode. Denne kode kan udleveres til vælgeren på forskellige måder; den kan f.eks. sendes med valgkortet, udleveres på selve valgstedet eller eventuelt sendes i en personlig mail. I dette projekt antages det, at koden udleveres på valgstedet mod fremvisning af et gyldigt valgkort. Denne kode skal selvfølgelig genereres, hvilket der ikke direkte er fremsat nogle værktøjer til i projektet.

I vores løsning er der blevet genereret en række tilfældige koder ved hjælp af C#. Disse koder er gemt i en tilsvarende ID-fil, som indlæses ved opstart af serveren. Efter identificering af vælgeren kan der afgives en stemme. For at denne stemme ikke kan kobles sammen med den benyttede kode og dermed risikere, at stemmehemmelighed mindskes, er der benyttet to separate databaser til henholdsvis koder og stemmer. Der er desuden blevet diskuteret yderligere tiltag, for at forbedre sikkerheden. Herunder var kryptering det, der blev overvejet mest. Dette var både i forhold til kryptering af stemmerne under afsendelse mellem klient og server, men også til, at verificere stemmerne. Det blev imidlertid fravalgt, at implementere dette i prototypen. I stedet blev der foretaget mindre foranstaltninger mod eventuelle sikkerhedsbrud. Disse bestod hovedsageligt i at adskille data tilhørende vælgerens identitet, fra vælgerens stemme. Ligeledes er det forsøgt, at gøre de forskellige dele af systemet så uafhængige af hinanden som muligt, så sikkerhedsbrud ét sted giver minimal skade.

Databaser

Identifikation og stemmeafgivningen bygger på vælger- og stemme-databasen. Til at kommunikere med disse, benyttede gruppen simple SQL-queries, dette kunne lade sig gøre, da de arbejdsopgaver databasen skulle udføre var forholdsvist simple. Til et mere komplekst system bør der implementeres Object Relational Mapping, som gør, at man kan arbejde med databasen, som med et hvilket som helst andet objekt i programmeringssproget. Dette vil afhjælpe problemer i forbindelse med vedligeholdelse og videreudvikling, da man i så fald ikke behøver være bekendt med SQL-sproget.

Vælger- og stemme-databasen fik hver især foretaget en række unit tests. Selve identifikationen af en vælger er en essentiel del af programmet, og den tilhørende database, vælger-databasen, blev dermed også gennemtestet. Disse test beskæftiger sig med, at bekræfte at en kode først og fremmest virker, hvis den er i databasen, når den bliver indtastet. Dernæst blev det testet at koder, der benyttes til at stemme med, registreres som værende brugte, samt at alle koder i starten af et valg er registrerede som ikke brugte. Selve stemme-afgivningen og -optællingen blev testet med unit tests på stemme-databasen. Her simulerede testen forskellige opdateringer af stemme-databasen, og bekræftede resultatet ved, at sammenholde databasens antal af stemmer med den forventede værdi. Disse test viste, at databasen er i stand til at modtage stemmer, opbevare dem og herefter optælle dem korrekt. Tilsammen viser disse test, at de to databaser virker efter hensigt og er i stand til at modificere data, efterhånden som

valget skrider frem.

12.1.2 Kravopfyldelse

Der er i EVS krav om at beskytte vælgerens generelle anonymitet og stemmehemmelighed. Det første krav kaldet “anonymitet” omhandler vælgerens anonymitet, når der trædes ind i stemmeboksen, og vælgeren skal til, at afgive sin stemme. Det andet krav er en del af et større ønske om, at opfylde grundlovens bestemmelser angående demokratiske valg, og omhandler stemmehemmeligheden.

Anonymitet

Det er essentielt for løsningen, at koderne ikke kan kobles sammen med en vælger eller dennes stemme, hvis grundloven skal overholdes (se afsnit 2.3). Dette løses med tilfældige distribution af koderne, samt den separate database, der opbevarer koderne. Koderne kan på denne måde bruges på hele valgstedet og ikke kun på en udvalgt terminal. Koden udleveres desuden umiddelbart før stemmehandlingen og sikrer dermed, at koden kun er kendt af vælgeren, der modtager koden, og minimerer desuden risiko for tyveri af denne. I forbindelse med stemmehemmeligheden menes der, at en kobling mellem en bestemt vælger og deres tilhørende stemme skal være umulig. Dette krav opfyldes først og fremmest ved, at skille de to databaser, som håndterer stemmer og koder. Adskillelsen af disse databaser er afgørende, da det giver muligheden for, at de kan ligge på to forskellige servere. Dette betyder rent praktisk, at der skal brydes ind i både den lokale stemme-terminals stemme-database og valgstedets identifikations-server for, potentielt at kunne koble en stemme afgivet på netop denne stemme-terminal sammen med dens tilhørende kode. Herefter skal koden så spores tilbage til en vælger, hvilket også burde volde problemer. Er stemmerne først samlet på valgstedets stemme-server, er det umuligt at koble en kode sammen med en stemme, uanset hvor meget adgang til systemerne der måtte have.

Sikkerhed

Afgivelse og opbevaring af stemmerne falder desuden også ind under kravet om sikkerhed. Med sikkerhed menes der i dette projekt, at der skal sikres imod ethvert forsøg på, at ændre på valgresultatet. Her var det især, de små tiltag, som skulle opfylde kravet. I databasen der indeholder kandidater og stemmer, er tabellerne struktureret således, at en simpel ændring af kandidatlistens ID ikke vil gøre de afgivne stemmer korrupte (se afsnit 10.1.4). En anden foranstaltning, som kan gøre det mere besværligt, at bryde ind og ændre stemmer i systemet er, at alle de forløbne stemmer ligger lokalt på de forskellige stemme-terminaler indtil valget afsluttes. Ved et sikkerhedsbrud i en valgterminal, begrænser antallet af ugyldige stemmer sig dermed umiddelbart til den ramte terminal. Dog vil en optælling af alle valgstedets fysiske papir-spor være nødvendig i forbindelse med valgfusk, da de ugyldige stemmer, når disse

samles med de gyldige stemmer på valgstedets stemme-server, ikke kan spores tilbage til en bestemt klient-terminal.

12.2 Netværkstyper og kommunikation

Der blev overvejet forskellige teknologier inden valget faldt på HTTP protokollen. HTTP opfylder de behov, løsningen har til kommunikation lokalt. Det faldt sammen med, at valgstedet holdes lukket for kommunikation udefra, grundet sikkerhedsmæssige årsager, såvel som for tillidens skyld. Det blev vurderet, at en løsning på et lukket system ville være mere troværdig for befolkningen. Med udgangspunktet om, at vælgeridentifikationen skulle centraliseres, passede “spørg-svar” modellen fra HTTP protokollen til løsningen, hvor en indtastet kode skal korrespondere til en gyldig kode på en database. HTTP blev lavet asynkront, for at have muligheden for, at flere klienter kan køre samtidigt og tilgå samme server. I en endelig løsning burde der implementeres HTTPS, da dette ville sikre dataene under overførsel.

For at teste om EVS implementering af et lokalt netværk var succesfuld, blev der lavet unit tests på de to klasser `HttpVotersClient` og `HttpVotersServer`. Testene undersøgte programmets korrekthed i tre forskellige scenarier. Her blev der tjekket, om en gyldig kode var benyttet eller ej, og hvorvidt en kode, der allerede var benyttet, kastede en passende “exception”. Alle disse test fik de forventede resultater, og bekræftede, at kommunikationen over netværket fungerede efter hensigten.

12.2.1 Kravopfyldelse

Netværket skal først og fremmest opfylde kravet om sikkerhed og gennemsikkelighed. Dette opnås i netværksdelen af EVS, i form af det lukkede system. Sikkerheden i forhold til netværket er dog et område, som skal forbedres i en eventuel videreudvikling, da dette er en meget essentielt del af et e-valgssystem (se afsnit 14.1). Netværkskommunikationen giver mulighed for, at have flere terminaler på et valgsted, som tilgår samme server med koder. Dette gør det sværere, at koble en vælger sammen med en stemmeafgivelse. Når koden, set fra serverens side, bliver brugt, kan det have været en vilkårlig klient, der har sendt anmodningen til serveren. Eftersom sikkerheden har en betydelig effekt på gennemsikkeligheden, bærer målet om, at udvikle en pålidelig løsning, tydeligt præg af, at finde en balance her imellem. Et pålideligt system er kun troværdigt så længe, at tilliden er vedholdende. Hvis brugerne ikke tror på, at systemet er sikkert, mister det pålideligheden. Samtidig udfordres et valgssystem også af vælgernes ønske om, at være sikre på, at de bliver hørt. De vil være sikre på, at deres stemmer bliver ordenligt talt op, og at der ikke ændres på resultatet undervejs. Projektet er her løbet ind i udfordringer med, at sikre, at stemmerne ikke kan spores tilbage til den enkelte vælger, kontra at den enkelte vælger ønsker, at kunne følge sin stemme så langt så muligt. Sikkerheden og sikring af anonymiteten kommer derfor i vejen for, at den enkelte stemme kan følges helt til sin optælling.

12.3 Samling og optælling af stemmer

Efter at have EVS basisfunktionalitet på plads, blev der udarbejdet en reel samling og optælling af de stemmer, der er afgivet i løbet af en valgdag. Denne funktionalitet giver mulighed for, at sende og samle stemmerne fra de enkelte valgmaskiner på server maskinen. Dette gør programmet brugbart til både store som små valg, da opsætningen kan konfigureres til, at fungere mellem mange maskiner, eller helt ned til kun at køre på en enkelt maskine. Dette giver mulighed at benytte EVS til både kommunale og nationale valg. Dette kan resultere i en økonomisk fordel, da det dermed ikke kræver forskellige former for udstyr til afholdelse af forskellige slags valg. En enkelt begrænsning af systemet er, at SQLite er begrænset til, at kunne arbejde med databasestørrelser på 140 terabyte. Dette svarer dog til omtrent 10 trillioner dataindgange i databasen [47], og denne størrelse burde være rigeligt til valg på nationalt plan. Til et endeligt valgssystem kunne det være en fordel, at implementere en anden databasetype. Dette kunne f.eks. være MS SQL (se afsnit 7.1), som tilbyder flere sikkerhedsforanstaltninger og egner sig mere til centralt brug.

I forhold til optælling af de samlede stemmer blev det diskuteret, hvorvidt optælling også skulle udføre udregning af mandater. Hvis dette skulle gøres, ville det kræve, at der blev foretaget ekstra udregninger, i overensstemmelse med det reglement, der findes for optælling af stemmer [26]. Dette blev valgt fra, og EVS foretager derfor kun en simpel optælling af stemmerne. Samling og optælling blev testet via unit tests, som oprettede stemmer i databasen, hvorefter det blev undersøgt, om optællings-metoden svarede til det forventede. Optællingen gik i testen, som forventet, og metodens resultat svarede overens med det forventede.

12.3.1 Kravopfyldelse

I denne del af programmet opfyldes kravet om vælgerens anonymitet, ved at stemme-serveren får en hel valgterminals stemmer tilsendt af gangen. Når en stemme-terminal lukker ned, bliver dens databasefil med stemmer sendt til stemme-serveren. Stemmerne flettes ved modtagelse på serveren med serverens andre stemmer. Der kan derfor ikke kigges tilbage i systemet for at se, hvornår, fra hvilken terminal eller med hvilken kode, der er stemt.

12.4 Brugerflade

Til EVS er der udviklet en brugerflade, der skal hjælpe vælgeren med, at benytte systemet. Brugerfladen er udviklet i Windows Forms, hvilket ikke er den mest optimale løsning, og gruppen har efterfølgende overvejet andre muligheder, så som Windows Platform Foundation. Windows Platform Foundation er i forhold til Windows Forms et kraftigere værktøj og er hyppigere brugt. Til udviklingen af selve brugergrænsefladens design, besluttede gruppen at tage udgangspunkt i nogle generelle principper om brugervenlighed, med udgangspunkt i Jakob Nielsens ti heuristikker (se afsnit 9.2 og 9.1). Dette indebærer blandt andet hjælp i de forskellige stadier af afstemningen, samt et minimalistisk design, for at undgå forvirring.

Realistisk set er der udvalgte grupper, som kan have problemer med prototypen. Der er for mennesker med former for fysiske handicap ikke sket nogle forbedringer, der bevirker, at de kan komme til at stemme selvstændigt. Sådanne forbedringer vil, f.eks. kunne implementeres i programmet i form af en oplæser til synshandicappede (se afsnit 4.3). Herudover tager løsningen ikke hensyn til mennesker, der ikke kan læse dansk, da alt dokumentation og hjælp i selve brugergrænsefladen er på dansk. Dette vil dog forholdsvis nemt, kunne forbedres ved at implementere versioner af den grafiske brugerflade på andre sprog, hvor brugeren kan vælge sit ønskede sprog i starten af afstemningen. En sidste gruppe, som kan have problemer med systemet, er mennesker med meget lidt eller ingen erfaring med IT-systemer. Den mest effektive måde at imødekomme disses behov vil højst sandsynligt være, at give undervisning i systemet før brug, frem for at ændre i det. Her kan der med fordel distribueres instruktioner til programmet op til et valg. Der er ikke foretaget en brugervenligheds-analyse af løsningen eller foretaget brugertests, da disse ligger udenfor pensum. Dette vil dog, til et endeligt system, være en nødvendighed for, at kunne udvikle en optimal brugergrænseflade. Brugerfladen er blevet gennemgået i forbindelse med afprøvning af programmet og herefter vurderet ud fra de informationer om brugervenlighed, diskuteret i rapporten (se afsnit 9.1).

12.4.1 Kravopfyldelse

Den grafiske brugerflade falder ind under kravet om generel brugervenlighed, samt lovmæssige begrænsninger i form af det direkte valg. For at valget forbliver, og ikke mindst fremstår direkte, kan der i den grafiske brugerflade ses alle de enkelte kandidater og lister, der kan stemmes på. Dette opfylder det lovmæssige krav om et direkte valg. Projektes krav om brugervenlighed fastslår for det første, at valget ikke skal blive sværre at benytte. Ifølge dette krav er der kun et enkelt problem med brugergrænsefladen, da prototypen ikke tilbyder nogen hjælp til de mennesker, som ikke kan finde ud af at benytte en computer, og personer med nogle former for fysiske handicap. Sidstnævnte gruppe ville dog også umiddelbart have problemer i det nuværende valgsystem, og falder derfor ikke under den del af befolkningen, der vil have sværre ved, at bruge denne løsning. Hertil skal der være mulighed for, at fortryde valg af en kandidat eller liste indtil den endelige stemme er afgivet. Dette opfyldes ved, at vælgeren under hele stemmeprocessen kan gå tilbage og vælge om. Vælgeren har derfor mulighed for at ombestemme sig, indtil stemmen er endeligt accepteret af brugeren.

Kapitel 13

Konklusion

I dette projekt er der set på, hvordan et elektronisk valg kan afvikles. Hertil blev der udviklet en prototype, der viser, hvordan en endelig løsning kunne udformes. Inden udviklingen af prototypen fravalgtes enkelte funktionaliteter. Herunder en streghode-scanner, handicap-tilgængelighed samt et fysisk papirspor. Disse funktionaliteter kunne hæve standarden for et system, hvis formål ville være at afvikle et egentligt valg, men er udeladt i dette projekt.

Projektets hovedmål, formuleret i problemformuleringen (se afsnit 5.2), er nået som følger:

- *Hvordan kan et valg afvikles, så vælgeren kan afgive sin stemme på valgstedet elektronisk?*

Rapportens afsnit, der omhandler programgennemgangen af den udviklede prototype, beskriver i detaljer, hvordan et system til elektronisk valg-afgivelse kunne se ud (se afsnit 10.2). Afsnittet beskriver, hvordan et sådant system kan opnås ved brug af en eller to servere og et, teoretisk ubegrænset, antal af stemme-terminaler ved hvert valgsted. Besvarelsen af dette delspørgsmål førte til udviklingen af selve prototypen, der opfylder de tilvalgte og eksterne krav til projektet (se afsnit 6.1). På mange måder minder den foreslåede metode meget om den nuværende metode til afvikling af valg. Den fundamentale base for løsningsforslagets metode er dog ikke papir, men elektronisk udstyr. Det er blevet testet og konkluderet, at den fremsatte prototype til valg-afvikling fungerer, og dermed er en indirekte besvarelse af dette delspørgsmål.

- *Hvordan kan den fordelagtige hastighed af elektronisk stemmeoptælling opnås, så pålideligheden ved fysiske papirstemmer samtidigt bevares?*

Der blev i projektet set på, om stemmerne i sig selv sikkert kunne lagres elektronisk. Efter diskussion førte dette til, at prototypen skulle have som krav, at benytte en database-datastruktur. Løsningsforslaget gør brug af SQLite til databasehåndtering, hvilket gør stemmeopbevaringen mere pålidelig, da SQLite benytter ACID-principperne (se afsnit 7.1). Udover at kunne gå tabt ved brand ligesom papirstemmer, er de elektronisk lagrede stemmer også truet af problemer, der altid har plaget elektronisk data. I forhold til den nuværende valgmetode kan de elektroniske stemmer derfor gå tabt på flere måder, eller endnu værre; blive manipuleret. Det er derfor blevet konkluderet, at der på nuværende tidspunkt ikke vil kunne udvikles et valgsystem, der udelukkende er elektronisk baseret, som vil kunne opnå pålideligheden af fysisk opbevarede stemmer. Det foreslås derfor, at det elektroniske valgsystem udvikles så det, samtidig med at opbevare stemmerne elektronisk, kan udskrive fysiske stemmer, der kan bruges til, at validere de elektroniske stemmer. Dermed kan den øgede hastighed i forbindelse

med optælling af elektroniske stemmer opnås, samtidig med at der altid vil være de fysiske papirspor at falde tilbage til. Denne fysiske manifestation af de elektroniske stemmer er ikke implementeret i løsningsforslaget, der udelukkende fokuserer på, at opbevare stemmerne elektronisk.

- *Hvordan kan man sikre at et e-valgssystem er både sikkert og gennemsommeligt?*

Sikkerheden af et valgsystem afhænger af flere faktorer, så som sikkerheden af de allerede afgivne stemmer og sikkerheden i forbindelse med opdagelse af, eller forebyggelse mod, valgfusk. Gennemsommeligheden af et valgsystem afhænger af, hvorvidt selve valgprocessen, fra stemmeafgivelse til valgresultat, er synlig og forståelig for den enkelte borger. Gennemsommeligheden af det fremsatte løsningsforslag, er vurderet til, at være mindre tilstedeværende end ved det nuværende valg, da selve afgivningen og optællingen af stemmer sker elektronisk. Vælgerne har ikke længere selv ansvaret for, at lægge deres stemme i en stemmeurne, men er nødsaget til at tage til takke med den besked fra stemme-terminalen, der fortæller vælgeren, at hans/hendes stemme er lagt i den elektroniske stemme-database. Vælgerne kan heller ikke længere bevidne selve optællings-processen ved hvert enkelt valgsted for at sikre sig, at stemmerne er blevet talt korrekt op. Sikkerhedsmæssigt opfylder løsningsforslaget grundlovens bestemmelser i forbindelse med en anonym afstemning, og sikrer derfor kravet om stemmehemmelighed. Det vil ikke være muligt at spore en elektronisk stemme tilbage til vælgeren, som har afgivet stemmen. Disse foranstaltninger bevirker dog, at systemets gennemsommelighed mindskes, og derfor er sikkerheden og gennemsommeligheden af systemet afhængige af hinanden. Det konkluderes derfor, at et elektronisk valg-system ikke kan opnå både sikkerhed og gennemsommelighed, da en forøgelse af den ene faktor vil medføre en reduktion af den anden. Løsningsforslaget implementerer desuden ikke krypterede databaser eller netværksforbindelser, som en brugbar løsning bør indeholde.

- *Til hvilken grad kan man udvikle et brugervenligt e-valgssystem til gavn for flest mulige vælgere?*

Løsningens stemme-terminalen bygger på en grafisk brugergrænseflade, som vælgeren kan navigere rundt i ved hjælp af en trykfølsom skærm. I prototypen, fremsat i dette projekt, er nogle aspekter i brugervenligheden udeladt. Der var blandt andet ikke midler til, at prøve brugergrænsefladen af på en trykfølsom skærm. Dog er programmet designet til at være behagelig at benytte med en sådan skærm (se afsnit 9.3). Der blev heller ikke implementeret udvidet hjælp til handicappede eller mulighed for, at vælge et andet sprog end dansk (se afsnit 9.3 og 9.1). De funktioner der blev implementeret i brugergrænsefladen, er mulighed for, at indtaste en vælger-kode, afgive en stemme gennem et grafisk interface, samt muligheden for, at gå tilbage gennem hele stemmeprocessen. Brugergrænsefladens design er udarbejdet ud fra en række principper opstillet af Jacob Nielsen (som beskrevet i afsnittene 9.1 og 9.2). Den indeholder desuden muligheden for, at få hjælp, til at stemme på stemmeterminalen, hvor dette kunne være en nødvendighed. Det blev overvejet hvilke designmæssige initiativer, der skulle gøres brug af, for så vidt muligt at forsimple valgprocessen, og mindske evt. forvirring. Brugergrænsefladen er dog ikke blevet bygget på brugervenligheds-undersøgelser eller andre analyser af brugerne, og har heller ikke været udsat for nogen brugertests. Dette skyldes kort og godt, at disse opgaver ligger uden for pensum. Det er imidlertid blevet konkluderet, at en brugervenlig og enkel brugergrænseflade kan udvikles til brug i en

elektronisk valgløsning. Dette er muligt, da det er den brede befolkningsgruppe, der rammes af løsningens grafiske design, og der er derfor ikke taget højde for andre udsatte grupper. At udvikle en løsning til handicappede vil kræve, at der bliver udviklet en række værktøjer, der vil gøre stemmehandlingen lettere for bestemte typer handicap. Om det er muligt, at få dækket alles behov i én brugergrænseflade, vil projektet ikke tage stilling til, da dette vil kræve en lang og dybdegående foranalyse.

- *Hvordan kan vælgerens anonymitet sikres ved et e-valgssystem?*

For at kunne afgive en stemme i EVS e-valgsløsningen, er det påkrævet, at man skal indtaste en gyldig kode på valgterminalen. Denne kode tjekkes af valgstedets server, som giver svar tilbage vedrørende kodens gyldighed (se afsnit 10.2 for teknisk uddybning). Koden udleveres, ligesom valgmappen ved det nuværende valg, af en valgtilforordnet. Koden er tilfældigt genereret og generisk, hvortil den, ligesom valgmapperne, kan benyttes af en vilkårlig stemmeberettiget. Der er i løsningsforslaget derfor ingen fysisk tilknytning mellem den enkelte kode og den vælger, som måtte benytte koden. Det er, som i prototypen, også muligt at sikre anonymiteten yderligere ved, at dele processen op i en identifikations-del og en stemme-del, så stemmerne heller ikke elektronisk kan føres tilbage til den enkelte vælger. Det er i løsningsforslaget implementeret ved, at have to databaser, der kører på to forskellige servere. Én server med koder og en anden med stemmer. Disse to servere, som også kan køres på en enkelt computer, arbejder uafhængigt af hinanden, hvortil der heller ikke kan drages forbindelser mellem stemmer og vælgere, selv hvis oplysningerne fra begge servere skulle være samlet på ét sted. Det konkluderes derfor at vælgeren, ved brug af prototypen, er anonym, i samme grad som han/hun ville være i nuværende valgssystem.

EVS løsningsforslaget fremstiller og demonstrerer altså et elektronisk system, der kan indsamle, hemmeligholde og optælle stemmer afgivet til et valg. Løsningen kan implementeres med et papirspor, så man ligesom nu kan gøre valget op ved hjælp af papirstemmer og manuel optælling, hvis dette bestemmes at være nødvendigt. Løsningen opfylder desuden Grundlovens krav om et almindeligt, direkte og hemmeligt valg (se afsnit 2.3). Et almindeligt valg opnås, da ingen mister deres ret til at stemme i den elektroniske løsning. Direkte valg opfyldes også, da stemmeberettigede afgiver sin stemme direkte til den kandidat eller parti, som vedkommende ønsker at stemme på. Herudover afgives stemmen i hemmelighed, ligesom ved det nuværende valg, og stemmen kan ikke sammenkobles med vælgeren, da de bagvedliggende systemer fungerer uafhængigt. Sikkerheden er i denne prototype mangelfuld, men kan forbedres ved at bruge kryptering af databaserne og forbindelsen mellem stemme-terminal og servere. EVS henvender sig til en så bred en del af befolkningen som muligt, og for at styrke pålideligheden og tilliden til systemet er den grafiske brugerflade lavet til, at efterligne det nuværende systems valgmappe. Herudover er der i brugergrænsefladen inddraget generelle brugervenligheds-principper, som letter brugen af programmet. Brugergrænsefladen mangler at implementere oplæsning og yderligere hjælp til handicappede vælgere, der ikke kan stemme med den nuværende fremsatte løsning. I tilfælde af en udrulning af systemet ville bruger-rettet dokumentation, om systemet og dets brug være en nødvendighed.

Projektets formål har været, at se på de forhindringer og krav, der er under udviklingen af et pålideligt elektronisk valgssystem og herefter fremstille en prototype til et sådant system. Prototypen demonstrerer mange af de funktionaliteter og egenskaber, et elektronisk valgssystem

skal indeholde. EVS giver derfor en ide om, hvordan et pålideligt elektronisk valgsystem kan udformes til den danske befolkning. En elektronisk valgløsning, baseret på det fremsatte løsningsforslag, er dog ikke på nuværende tidspunkt hverken sikkert eller gennemskueligt nok, til at leve op til de krav, som det danske folk og dets regering stiller. Det er tvivlsomt at en elektronisk valgløsning, der vil kunne leve op til nævnte krav, kan udvikles indenfor de næste få år, men uanset hvilke former for e-valgssystemer der kommer i fremtiden, vil det være nødvendigt at undersøge og behandle nogle af de samme problemstillinger, som dette projekt har arbejdet med. Herved giver dette projekt et billede af e-valgsteknologien og dets nuværende plads i det danske samfund.

Kapitel 14

Perspektivering

I dette afsluttende afsnit ses der på mulighederne for videreudvikling af selve programmet, herunder mulige forbedringer af væsentlige dele i systemet. Andre brugsformer for projektets e-valgsløsning vil også blive beskrevet.

14.1 Videreudvikling

Løsningsforslaget, som er udviklet i dette projekt, er kun en prototype af et færdigt program. Hvis dette skal videreudvikles til en egentlig løsning, er der flere ting, som skal implementeres.

Da en endelig løsning ville komme til at blive brugt af alle stemmeberettigede, er det også vigtigt at se på eventuelle udvidelser af brugervenligheden. Til dette formål vil en brugerundersøgelse, baseret på analyser og involvering af brugerne, være nødvendig for at få den bedste indsigt i, hvilke forbedringer vælgerne vil have gavn af. Dette er essentielt, specielt fordi hele befolkningen skal kunne benytte sig af systemet, men også fordi en veludviklet brugergrænseflade med høj fokus på brugervenlighed formegentlig vil øge brugerens tillid og lyst til at bruge systemet.

Der skal desuden være fokus på at udvide systemet med henhold til de handicappede samt de mindre gængse IT-brugere. Dette kan blandt andet gøres ved hjælp af oplæsning af tekst eller mulighed for tekstforstørrelse til svagsynede, tastatur-styring til folk der har svært ved at bruge musen samt introduktionsfoldere eller kurser til folk, der føler sig utrygge ved teknologien. Desuden kan der ses på at implementere en strekcode-scanner, der vil kunne gøre indlæsningen af stemmeberettigedes kode nemmere, da denne i så fald kun skal scannes, hvorved brugeren ikke selv skal sørge for at indtaste sin kode. Disse udvidelser vil være med til at sikre en forbedring af stemmemulighederne for de handicappede, i forhold til den nuværende valgmetode, og kunne gøre det nemmere for dem, at deltage ved valg. Det sikrer også, at de fleste vil føle sig trygge ved skiftet til et elektronisk valg.

Derudover skal der ses på en udvidelse af sikkerheden på flere punkter. Der kan blandt andet bruges HTTPS til netværskommunikationen. Dette vil sikre, at der ikke er nogen, der kan opfange og ændre kommunikationen over netværket. Desuden skal de forskellige data krypteres, så det ikke kan læses af folk, der ikke burde have adgang til det. Der skal også ses på flere sikkerhedsforanstaltninger, der sikrer, at valgdatabaserne ikke er blevet manipuleret, og

at valget er forløbet fejlfrit. Der skal også køres mange sikkerhedstest, da der ikke må være nogen muligheder for fejl, der kan føre til snyd.

Til en eventuel videreudvikling kan det også være en fordel at skifte til nogle andre værktøjer, der vil gøre udviklingen af programmet nemmere. Til arbejdet med databasen kan der bruges Object Relational Mapping, der implementerer en fortolker, som gør, at databasen vil kunne bruges på samme måde, som andre datastrukturer i programmeringssproget. Til udviklingen af brugerfladen kan der skiftes til Windows Presentation Foundation, der er et kraftfuldt værktøj til udvikling af brugerflader. Disse værktøjer vil gøre det muligt at udvikle løsningen både nemmere og hurtigere.

14.2 Udvidede brugsmuligheder

Dette projekt har udviklet et e-valgssystem, som ville skulle implementeres ved et dansk folketingsvalg, men i Danmark foretages der flere “afstemnings-valg” end folketingsvalget. I Danmark holdes der også folkeafstemninger, kommunal- og regionsvalg samt valg til Europa-Parlamentet, disse “afstemnings-valg” er dækket af lignende love (se afsnit 2.3). Projektets løsningsforslag vil derfor også kunne benyttes til disse typer valg.

Hvert demokratisk land har hvert sit valg-system, med dertilhørende regler og love. Dette projekts løsningsforslag er som nævnt udarbejdet specielt til Danmarks valg-system, derfor vil en direkte implementering til andre lande med demokratiske “afstemnings-valg” ikke være mulig. De grundliggende valg- og sikkerhedselementer, fra dette projekts løsningsforslag, vil dog kunne genbruges i andre e-valgsløsninger.

E-valgsteknologien har mange muligheder, og skal ikke nødvendigvis kun benyttes ved nationale-valg. Teknologien har potentiale til at blive en essentiel del i mange andre former for afstemninger og valg, både i den offentlige og private sektor. På baggrund af de resultater og erfaringer projektet har givet, vil det ikke blive anbefalet at benytte systemet i store afstemninger, men derimod holde det forholdsvis lokalt. Anvendelsen af dette system kan i stedet være valg af bestyrelsesformand for et stort aktieselskab, eller små afstemninger i en skolebestyrelse.

Kapitel 15

Bibliografi

15.1 Bøger

- [1] S. Bang m.fl. *Kvalitetsstyring i Systemudvikling*. 1st. Teknisk Forlag, 1991, s. 190. ISBN: 87-571-1392-0.
- [2] I. Sommerville. *Software Engineering*. Udg. af A.D McGettrick. 6th. Addison Wesley, 2001, s. 693. ISBN: 0-201-39815-X.

15.2 Artikler og rapporter

- [3] *Technical dialogue on system for e-voting in Denmark - Summary report*. Tek. rap. 2012-00363. Doc. no. 4765. Økonomi- og Indenrigsministeriet, sep. 2012.
- [4] Ida-Elisabeth Andersen, Katrine Lindegaard Juul og Jakob Vedelsby. *E-valg - et valg for fremtiden?* Tek. rap. Teknologirådet, mar. 2011. URL: http://www.tekno.dk/pdf/projekter/p10_evalg/p11_E-valg_Baggrundsrapport.pdf (sidst set 11.02.2013).
- [5] *E-valg*. Tek. rap. IT-Politisk Forening, 2012. URL: <http://www.itpol.dk/politik/evalg> (sidst set 11.02.2013).
- [6] Jordi Barrat i Esteve, Ben Goldsmith og John Turner. *International Experience with E-Voting*. Tek. rap. International Foundation for Electoral Systems, 2012, Juni. URL: http://www.ifes.org/Content/Publications/News-in-Brief/2012/June/~media/Files/Publications/Reports/2012/EVote_International_Experience_2012.pdf (sidst set 21.05.2013).
- [7] Theo Haerder og Andreas Reuter. „Principles of Transaction-Oriented Database Recovery“. I: *ACM Computing Surveys* 15 (dec. 1983), s. 287–317.
- [8] Fay Chang m.fl. „Bigtable: A distributed storage system for structured data“. I: *ACM Transactions on Computer Systems (TOCS)* 26.2 (2008), s. 4.
- [9] P. A. Craig. *NUnit V2.0 Manual*. 2002. URL: <http://www.nunit.org/index.php?p=documentation> (sidst set 16.05.2013).

15.3 Websider

- [10] *Artikel vdr. stemmeprocent (i årene 1973, 2007, 2011)*. Feb. 2013. URL: <http://www.dr.dk/Nyheder/Temaer/2011/Valg/2011/09/16/005326.htm> (sidst set 15.02.2013).
- [11] IT-Politisk Forening. *Høring om digitalt valg*. URL: <http://www.itpol.dk/taxonomy/term/29> (sidst set 03.03.2013).
- [12] Den store Danske. *Enevælde i Danmark*. URL: http://www.denstoredanske.dk/Danmarks_geografi_og_historie/Danmarks_historie/Danmark_1536-1849/enev%C3%A6lde (sidst set 22.02.2013).
- [13] Den Store Danske. *Demokrati*. URL: http://www.denstoredanske.dk/Samfund,_jura_og_politik/Samfund/Moderne_demokrati_og_konstitutionelt_monarki/demokrati (sidst set 20.02.2013).
- [14] *Folketingsvalg*. Aug. 2012. URL: <http://www.ft.dk/Undervisning/Folketingsvalg.aspx> (sidst set 14.02.2013).
- [15] *Hvordan stemmer jeg?* Aug. 2010. URL: <http://www.sum.dk/sitecore/content/Campaign/Home/Vaelgere/stemme.aspx#valgdagen> (sidst set 14.02.2013).
- [16] *Brevstemme til folketingsvalg og folkeafstemninger*. Sep. 2012. URL: <https://www.borger.dk/Sider/Brevstemme-til-folketingsvalg-og-folkeafstemninger.aspx> (sidst set 14.02.2013).
- [17] *Brevstemmeafgivning*. Aug. 2012. URL: <http://valg.oim.dk/vaelgere/brevstemmeafgivning.aspx> (sidst set 14.02.2013).
- [18] *Valgmetode*. Aug. 2012. URL: <http://da.wikipedia.org/wiki/Valgmetode#Flerrunde-systemer> (sidst set 14.02.2013).
- [19] *Danmarks Riges Grundlov*. Jun. 1953. URL: <https://www.retsinformation.dk/forms/R0710.aspx?id=45902> (sidst set 14.02.2013).
- [20] *Bekendtgørelse af lov om valg til Folketinget*. Feb. 2013. URL: <https://www.retsinformation.dk/forms/R0710.aspx?id=144959> (sidst set 14.02.2013).
- [21] *Bekendtgørelse af lov om kommunale og regionale valg*. Feb. 2013. URL: <https://www.retsinformation.dk/forms/R0710.aspx?id=144947> (sidst set 14.02.2013).
- [22] *Bekendtgørelse af lov om valg af danske medlemmer til Europa-Parlamentet*. Feb. 2013. URL: <https://www.retsinformation.dk/forms/R0710.aspx?id=144942> (sidst set 14.02.2013).
- [23] *E-valg definitioner*. URL: <http://www.e-voting.cc/en/it-elections/definitions/> (sidst set 15.02.2013).
- [24] Borger.dk. *Brevstemme til folketingsvalg og folkeafstemninger*. URL: <https://www.borger.dk/Sider/Brevstemme-til-folketingsvalg-og-folkeafstemninger.aspx> (sidst set 15.02.2013).
- [25] Lasse Lange. *Elektroniske valg kommer trods kritik*. altinget.dk. Feb. 2013. URL: <http://www.altinget.dk/artikel/elektroniske-valg-kommer-trods-kritik> (sidst set 11.02.2013).

- [26] *Stemmeoptælling*. Aug. 2012. URL: [http://www.denstoredanske.dk/Samfund%2c_jura_og_politik/Jura/Danmarks_statsforfatning/folketingsvalg/folketingsvalg_\(Stemmeopt%C3%A6lling\)](http://www.denstoredanske.dk/Samfund%2c_jura_og_politik/Jura/Danmarks_statsforfatning/folketingsvalg/folketingsvalg_(Stemmeopt%C3%A6lling)) (sidst set 14.02.2013).
- [27] Danmarks Statistik. *Tal fra Danmarks Statistik vdr. valget i 2011*. URL: <http://www.statistikbanken.dk> (sidst set 17.02.2013).
- [28] *Impact of e-voting*. URL: <http://www.e-voting.cc/en/it-elections/impact/> (sidst set 17.02.2013).
- [29] Jesper Kildebogaard. *Vestager: Vi skal prøve e-valg af, før vi beslutter os*. version2.dk. Jan. 2013. URL: <http://www.version2.dk/artikel/vestager-vi-skal-proeve-e-valg-af-foer-vi-beslutter-os-49771> (sidst set 14.02.2013).
- [30] Jesper Kildebogaard. *Massiv kritik af lov om e-valg: Det bliver dyrere og mere usikkert*. version2.dk. Dec. 2012. URL: <http://www.version2.dk/artikel/massiv-kritik-af-lov-om-e-valg-det-bliver-dyrere-og-mindre-sikkert-49461> (sidst set 11.02.2013).
- [31] Margrethe Vestager. *Derfor skal vi afprøve e-valg*. version2.dk. Feb. 2013. URL: <http://oim.dk/nyheder/nyhedsarkiv/2013/feb/derfor-skal-vi-afproeve-e-valg.aspx> (sidst set 17.02.2013).
- [32] Margrethe Vestager. *Vestager svarer på kritik*. version2.dk. Feb. 2013. URL: <http://www.version2.dk/artikel/margrethe-vestagers-svar-til-laeserne-om-e-valg-50476> (sidst set 27.02.2013).
- [33] *Distinctive Features Of SQLite*. URL: <http://www.sqlite.org/different.html> (sidst set 26.02.2013).
- [34] *Most Widely Deployed SQL Database*. URL: <http://sqlite.org/mostdeployed.html> (sidst set 26.02.2013).
- [35] *SQLite Features*. URL: <http://www.sqlite.org/features.html> (sidst set 12.05.2013).
- [36] *Microsoft Big Data*. URL: <http://www.microsoft.com/en-us/sqlserver/solutions-technologies/business-intelligence/big-data.aspx> (sidst set 27.02.2013).
- [37] *Microsoft SQL Security & Compliance*. URL: <http://www.microsoft.com/en-us/sqlserver/solutions-technologies/mission-critical-operations/security-and-compliance.aspx> (sidst set 27.02.2013).
- [38] *Microsoft SQL Availability*. URL: <http://www.microsoft.com/en-us/sqlserver/solutions-technologies/mission-critical-operations/high-availability.aspx> (sidst set 27.02.2013).
- [39] *Microsoft SQL Case Studies*. URL: <http://www.microsoft.com/en-us/sqlserver/product-info/case-studies.aspx> (sidst set 27.02.2013).
- [40] *MySQL Replication: An Introduction*. URL: <http://www.mysql.com/why-mysql/white-papers/mysql-replication-introduction/> (sidst set 02.03.2013).
- [41] *13.4.10. mysqlfailover — Automatic replication health monitoring and failover*. URL: <http://dev.mysql.com/doc/workbench/en/mysqlfailover.html> (sidst set 02.03.2012).

- [42] T. Berners-Lee m.fl. *Hypertext Transfer Protocol – HTTP/1.0*. Maj 1996. URL: <http://www.ietf.org/rfc/rfc1945.txt> (sidst set 15.02.2013).
- [43] E. Rescorla. *HTTP Over TLS*. Maj 2000. URL: <http://tools.ietf.org/html/rfc2818> (sidst set 12.05.2013).
- [44] Jakob Nielsen. *Usability quality components*. URL: <http://www.nngroup.com/topic/web-usability/> (sidst set 12.04.2013).
- [45] Jakob Nielsen. *10 Usability Heuristics*. URL: <http://www.nngroup.com/articles/ten-usability-heuristics/>.
- [46] Microsoft. *Using stubs to isolate parts of your application from each other for unit testing*. URL: <http://msdn.microsoft.com/en-us/library/hh549174.aspx> (sidst set 12.05.2013).
- [47] *SQLite Limits*. URL: <http://www.sqlite.org/limits.html> (sidst set 05.05.2013).

Del IV

Bilag

Kapitel 16

Brugervejledning

Dette bilag vil indeholde to gennemgående brugervejledninger, en detaljeret vejledning i selve opsætning og fejlbehandlingen af programmet til de administrerende valg-ordinerede og en mindre dejtaljeret brugervejledning (brugsanvisning).

16.1 Opsætnings- og fejlbehandlings-guide

Denne vejledning er til den administrerende valg-ordinerede og skal bruges til at starte alle dele af EVS E-Voting System og til sidst vil der være et afsnit om fejlbehandling.

For at benytte EVS E-Voting System skal følgende IT-redskaber være tilstede:

Server

En server til ekstern behandling og bevaring af valg-materiale. Eventuelt en ekstra server til opdeling af serverne.

Klient

Det ønskede antal klienter skal opstilles på valgstedet.

Lokalt netværk

Et lokalt netværk til intern kommunikation mellem serverne og klienter, dette kan laves med en switch el.l. og dertil hørende netværkskabler.

Disse redskaber sluttet sammen og systemmet er nu klar til brug.

Opstart af serverne

Dette afsnit vil trinvist forklare hvordan serverne i EVS E-Voting System startes og benyttes.

Identifikations-serveren

1. For at starten identifikations-serveren køres programmet "VotersServer.exe" fra "ID-Server"-mappen.

2. Programmet spørger nu om serveren skal lytte til localhost. Hvis “Ja” vælges, spring til trin 4.
3. Indtast nu adressen, som serveren skal lytte til. Syntaks er: “http://[abc.]Abc123[.abc]:1234/Abc12” hvor led i kantet parentes er valgfrie.
4. Programmet spørger nu om serveren skal genskabe sidste session. Vælg “Ja” hvis strømmen gik, og valget skal fortsætte.
5. Programmet vil nu konfigurere sig selv efter de ønskede indstillinger. Vent indtil beskeden “Serveren er klar” efterfulgt af “Tast "help"for at vise en liste over mulige kommandoer” vises på skærmen. Herefter er identifikations-serveren klar til brug.

Stemme-serveren

1. For at starten stemme-serveren køres programmet “VotesServer.exe” fra “Votes-Server”-mappen.
2. Programmet spørger nu om serveren skal lytte til localhost. Hvis “Ja” vælges, spring til trin 4.
3. Indtast nu adressen, som serveren skal lytte til. Syntaks er: “http://[abc.]Abc123[.abc]:1234/Abc12” hvor led i kantet parentes er valgfrie.
4. Programmet spørger nu om serveren skal genskabe sidste session. Vælg “Ja” hvis strømmen gik, og valget skal fortsætte eller stemmerne ikke blev talt op inden strømmen gik.
5. Programmet vil nu konfigurere sig selv efter de ønskede indstillinger. Vent indtil beskeden “Serveren er klar” efterfulgt af “Tast "help"for at vise en liste over mulige kommandoer” vises på skærmen. Herefter er stemme-serveren klar til brug.

Kommandoer

I dette afsnit beskrives de forskellige kommandoer, som serverne kan benytte.

Fælles kommandoer. Disse kan benyttes af både identifikations-serveren og stemme-serveren. De indtastes direkte i serverens konsol-vindue.

clear

Rydder konsol-vinduet for tekst.

help

Printer en hjælp-skærm ud på konsol-vinduet, der kort fortæller om de forskellige kommandoer tilgængelige på serveren.

exit

Lukker serveren. Stopper serveren og lukker server-programmet.

quit

Samme som **exit**.

status

Printer en kort status af serveren ud på konsol-vinduet. Status-beskeden består af, hvis serveren kører, hvilken URL serveren lytter på. Hvis serveren af en eller anden grund ikke kører som den skal printes dette til konsol-vinduet.

Stemme-server kommandoer. Disse kommandoer er kun tilgængelige på stemme-serveren og kan *ikke* bruges på identifikations-serveren.

tally

Printer valgresultatet ud på konsol-vinduet. Resultatet består af hver kandidats navn og antallet af stemmer, som kandidaten har fået.

tallyex

Printer ligesom **tally** valgresultatet ud på konsol-vinduet, men udelader at udskrive de kandidater, der ikke har fået nogle stemmer.

Opstart af klient

I dette afsnit beskrives det, hvordan en stemme-terminal skal startes op.

1. Først spørger programmet, om standard-indstillingerne skal indlæses.
2. Derefter spørger programmet, om den sidste session skal genskabes. Dette skal gøres hvis stemmerne ikke blev ordenligt sendt til stemme-serveren ved nedlukning, eller hvis der var en strømafbrydelse. Bemærk: Valget kan ikke genskabes hvis terminalen lukkede ned korrekt ved sidste session.
3. Herefter vises opsætnings-skærmen. Er standard-indstillingerne indlæst, og kører både identifikations-serveren og stemme-serveren på samme maskine som stemme-terminalen, kan du springe til trin 7.
4. Indtast URL'en til identifikations-serveren og URL'en til stemme-serveren i de respektive tekstbokse i kassen *Opsætning*.
5. (Valgfri) Vælg en anden nedlukningskode. Denne kode benyttes som Vælger-ID, når terminalen skal lukkes. Bemærk: Den behøver hverken at udfylde begge tekstbokse, bestå udelukkende af tal eller overhovedet være defineret.
6. (Genskabelse) Hvis sidste session blev genskabt kan der nu trykkes på knappen "Send stemmer" for at afslutte e-valget eller springes til næste trin for at forsætte e-valget.
7. Tjek at Åbningskoden er gyldig og tryk på knappen "Åben". Terminalen kan også lukkes uden at starte op ved at trykke på knappen "Luk Maskine".
8. Hvis serverne kører, ses nu forsiden, hvor et Vælger-ID skal indtastes. Indtast her et gyldigt Vælger-ID, så som "111111111-1111", for at starte en stemme-proces, eller

indtast nedlukningskoden, for at lukke terminalen og sende de afgivne stemmer på terminalen til stemme-serveren.

Fejlbehandling

I forbindelse med brugen af den grafiske klient i EVS E-Voting System, kan der opstå forskellige fejl, i forskellige dele af programmet. Disse fejl, og deres løsning, vil blive beskrevet i det følgende afsnit.

Netværksfejl

Hvis en **Netværksfejl** udløses når en **stemme-terminal prøver at åbne**, er det enten fordi stemme-serveren ikke er tilgængelig på den indtastede “Stemme-server URL”, eller fordi stemme-serveren ikke er startet. For at løse dette problem gøres følgende;

1. Hvis stemme-serveren er startet gå til trin 2. Ellers, kørs først stemme-serveren ved at starten den på en computer, der er tilkoblet det samme netværk, som stemme-terminalerne (for en gennemgang af, hvordan stemme-serveren startes, se afsnit 16.1).
2. Tjek at den indtastede URL for stemme-serveren er korrekt og stemmer overens med den URL, som stemme-serveren lytter på. For at se hvilken URL stemme-serveren lytter på indtastes kommandoen “status” i stemme-serverens konsol-vindue.
3. Prøv igen at åbne stemme-terminalen.

Hvis en **Netværksfejl** udløses når der trykkes på “**acceptér**”-knappen, er det enten fordi identifikations-serveren ikke er tilgængelig på den indtastede “Identifikations-server URL” under opsætningen, eller fordi identifikations-serveren ikke er startet. For at løse dette problem gøres følgende;

1. Hvis identifikations-serveren er startet gå til trin 2. Ellers, kørs først identifikations-serveren ved at starten den på en computer, der er tilkoblet det samme netværk, som stemme-terminalerne (for en gennemgang af, hvordan identifikations-serveren startes, se afsnit 16.1).
2. Genstart stemme-terminalen, ved først at indtaste nedlukningskoden i boksen til Vælger-ID og derefter starte programmet igen.
3. Tjek at den indtastede URL for identifikations-serveren er korrekt og stemmer overens med den URL, som identifikations-serveren lytter på. For at se hvilken URL identifikations-serveren lytter på indtastes kommandoen “status” i identifikations-serverens konsol-vindue.
4. Åben stemme-terminalen, og prøv igen.

Hvis en **Netværksfejl** udløses i forbindelse med en **stemme-terminals nedlukning**, er det fordi stemme-serveren ikke længere kører. For at løse dette problem skal stemme-serveren

genstartes i sessiongenskabelses-tilstand, hvorefter stemme-terminalen skal startes igen. Når dialog-boksen, der spørger om valget skal genskabes, vises, vælges “ja”. Herefter kan valget fortsætte, eller stemmerne kan sendes, ved at trykke på knappen “Send stemmer” ned i højre hjørne af opsætnings-vinduet.

Hvis en **Server Timeout** fejl udløses i forbindelse med **åbning af en stemme-terminal**, er det enten fordi stemme-serveren er travlt optaget, eller fordi der er indtastet en stemme-server URL, der peger på et gyldigt websted, men ikke en gyldig stemme-server. Hvis stemme-serveren er travlt optaget, løses problemet ved at vente et øjeblik, for derefter at prøve igen. Ellers tjekkes det at den indtastede “Stemme-server URL” peger på den URL, som stemme-serveren lytte på. For at se hvilken URL stemme-serveren lytter på indtastes kommandoen “status” i stemme-serverens konsol-vindue.

Hvis en **Server Timeout** fejl udløses i forbindelse med **stemme-terminalens nedlukning, tryk på “accepter”-knappen eller tryk på “Stem”-knappen**, er det fordi stemme- og eller identifikations-serveren er travlt optaget. Dette løses ved at vente et øjeblik og derefter prøve igen.

Serverfejl

Hvis en **Serverfejl** udløses i forbindelse med **stemme-terminalens nedlukning eller tryk på “Acceptér”-knappen**, er det fordi stemme-serveren ikke opfører sig som forventet. Tjek at det er den nyeste version af stemme-serveren, der kører på den indtastede “Stemme-server URL” og prøv igen.

Hvis en **Serverfejl** udløses **ved tryk på “Stem”-knappen**, er det fordi identifikations-serveren imellem validerings-tidspunktet og stemme-tidspunktet for terminalen, er blevet genstarten på samme URL med et andet sæt gyldige Vælger-ID’er. Dette kan løses ved at lukke den nye identifikations-server og starte den gamle op igen, hvorefter vælgeren på stemme-terminalen er nødt til at gå igennem stemme-processen igen.

Hvis en **Identifikationsfejl** udløses **ved tryk på “Stem”-knappen**, er det fordi det indtastede Vælger-ID er blevet brugt på et tidspunkt imellem identifikations-tidspunktet og stemme-tidspunktet på terminalen. Dette kan kun ske, hvis to terminalen tjekker-ind sammen ID, inden for et kort tidsrum, og én af dem stemmer med ID’et. Den anden terminal vil så melde fejlbeskeden når den prøver at stemme med ID’et. Denne fejl kan umiddelbart ikke løses, da der kun kan stemmes én gang med hvert ID, og den forrige stemme med ID’et ikke kan spores tilbage til ID’et.

Klientfejl

Hvis en **Ugyldig URL** fejlbesked udløses når en **stemme-terminal prøver at åbne**, er det fordi mindst én af de indtastede URL’er ikke opfylder den krævede syntaks. Serverne skal også opfylde denne syntaks, så der kan umuligt være en server kørende på den indtastede URL. For at løse fejlen, indtastes de URL’er, som hhv. identifikations-serveren og stemme-serveren

lytter på i de respektive tekstbokse. For at se hvilken URL en server lytter på indtastes kommandoen “status” i serverens konsol-vindue.

Hvis en **Klientfejl** udløses i forbindelse med **stemme-terminalens nedlukning**, er det fordi terminalens stemme-database ikke er tilstede på nedluknings-tidspunktet. Hvis databasen “valgdata.sqlite” ikke ligger i stemme-terminalens installations-mappe, tjek da, at ClientGUI.exe køres med administrator-rettigheder, eller sørg for at den Windows-bruger, der kører terminalen har fil-rettigheder til installations-mappen.

Hvis en **Klientfejl** udløses **ved tryk på “Stem”-knappen**, er det pga. en teknisk fejl på terminalen. Terminalen kan genstartes for, at prøve, at sikre fremtidige stemme-processer, men den stemme, som udløste fejlen vil gå tabt, og ID’et vil ikke kunne få lov til at stemme igen.

16.2 Brugervejledning til vælgeren

Denne brugervejledning er opstillet efter de forskellige stadier du kan komme igennem i systemet. Der vil ved et par af de forskellige stadier være en “hjælp” knap, som kan aktiveres for en forklaring af stadiet. Det er desuden muligt at gå tilbage ved alle stadier, indtil at stemmen er afgivet.



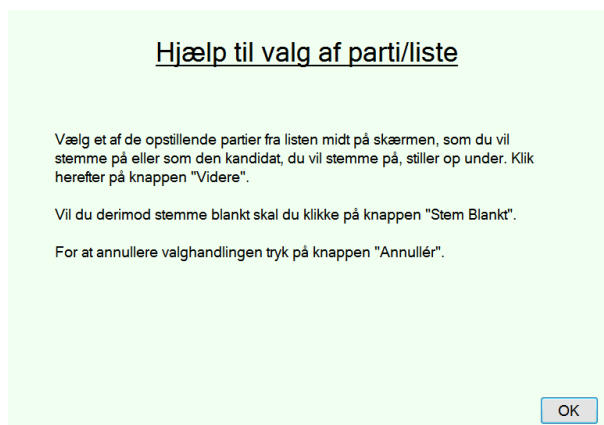
1: Velkommen til velkomstskærmen her skal du indtaste dit “Vælger-ID” som du har fået udleveret af en valgtilforordnet.

Hvis systemet ikke godkender dit ID skal du tjekke om ID’et er korrekt indtastet ellers skal du kontakte en valgtilforordnet.



2: Velkommen til valgekortet. Du kan nu vælge det parti du ønsker at stemme på ud fra listen. Efter valg af parti kan du klikke dig videre for at få mulighed for at vælge en specifik kandidat fra det valgte parti, eller at afgive stemmen til selve partiet.

Ønsker du at stemme blankt kan du vælge den dertilhørende knap og godkende dit valg.



2a: Her ses hjælpe informationen for valg af parti/liste, hvor skærmens formål og brugs-måde er forklaret. Ved klik på “OK” knappen føres man tilbage til valg-skærmen.

Vælg Kandidat: Det Konservative Folkeparti Hjælp

Lars Barfoed
Tom Behnke
Lene Espersen
Mai Henriksen
Benedikte Kjør
Mike Legarth
Brian Mikkelsen
Per Stig Møller

Tilbage Stem på Parti Stem på Kandidat

3: Her ses kandidatlisten for et valgt parti. Det er desuden herfra stemmen afgives, der er to måder dette kan gøres på. Stemmen kan afgives til det allerede valgte parti, eller der kan vælges en kandidat fra listen og afgive stemmen til denne.

Hjælp til valg af kandidat

Vælg en af kandidaterne fra listen midt på skærmen, som du vil stemme på. Klik herefter på knappen "Stem på Kandidat".


Vil du derimod stemme på det valgte parti, hvis navn kan ses over listen af kandidater, skal du trykke på knappen "Stem på Parti".

For at vælge et andet parti tryk på knappen "Tilbage".

OK

3a: Her ses hjælpe informationen for valg af kandidat, hvor skærmens formål og brugs-måde er forklaret. Ved klik på "OK" knappen føres man tilbage til valg-skærmen.

Afgiv Stemme

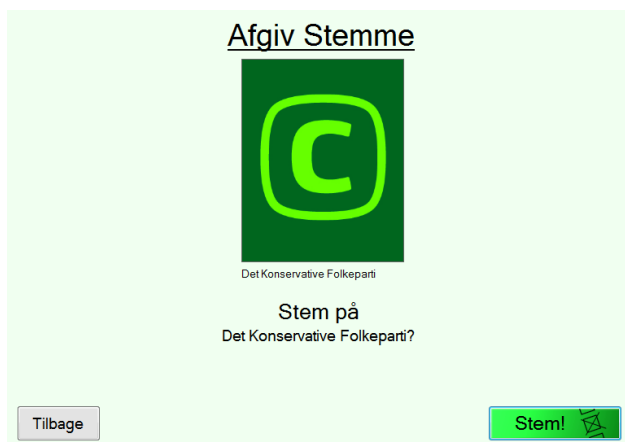


Per Stig Møller
Det Konservative Folkeparti

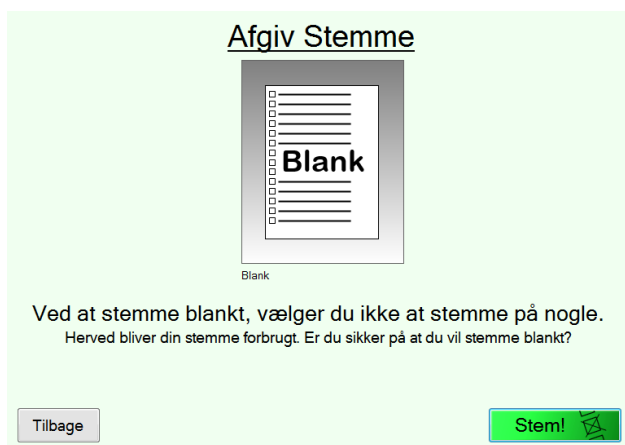
Stem på
Per Stig Møller fra Det Konservative Folkeparti?

Tilbage Stem! 

4a: Her ses stemme-bekræftelsen, her er der vist den kandidat du ønsker at stemme på. Du kan afgive din stemme ved tryk på den grønne "stem"-knap, hvorefter din stemme vil være lagret elektronisk. Du kan som altid også gå tilbage i tilfælde af fejltryk.



4b: Her ses stemme-bekræftelsen, her er der vist den liste du ønsker at stemme på. Du kan afgive din stemme ved tryk på den grønne “stem”-knap, hvorefter din stemme vil være lagret elektronisk. Du kan som altid også gå tilbage i tilfælde af fejltryk.



4c: Her ses stemme-bekræftelsen, her er der vist den blanke stemme du ønsker at afgive. Du kan afgive din stemme ved tryk på den grønne “stem”-knap, hvorefter din stemme vil være lagret elektronisk. Du kan som altid også gå tilbage i tilfælde af fejltryk.



5: Din stemme er nu afgivet og lagret i den elektroniske database. Afstemningen er nu ovre, og du kan forlade stemmeboksen.