

# Testspecifikation

Førsteårsprojekt

## Gruppe 1:

Troels Henriksen (athas@sigkill.dk)

Jesper Reenberg (reenberg@kampsax.dtu.dk)

Martin Dybdal (dybber@dybber.dk)

Vejledere: Dina og Kasper

6. juni 2007

## Indhold

<b>1 Strategi</b>	<b>3</b>
<b>2 Kravtest</b>	<b>3</b>
Krav 1: Analyse af et helt websted . . . . .	4
Krav 1.1: Respekter robots.txt . . . . .	4
Krav 1.2: Dybde af crawling . . . . .	5
Krav 2: Analysemetoder . . . . .	6
Krav 2.1: Læsbarehedsindeks, LIX . . . . .	7
Krav 2.2: Flesch-Kincaid Readability Test (FKRT) . . . . .	8
Krav 2.3: Stavekontrol . . . . .	10
Krav 2.3.1: Flere sprog . . . . .	10
Krav 2.4: Gentagne ord . . . . .	11
Krav 2.5: Beregning af sidesværhedsgrad . . . . .	12
Krav 3: Analyse baseret på HTML-tags . . . . .	13
Krav 3.1: em og strong . . . . .	13
Krav 3.2: hN . . . . .	13
Krav 3.3: abbr og acronym . . . . .	13
Krav 3.4: Citater skal ikke analyseres. . . . .	14
Krav 3.5: Tekst i andre sprog angivet med (lang) . . . . .	15
Krav 3.6: kbd, var og code . . . . .	16
Krav 3.7: bdo . . . . .	20
Krav 4: Konfiguration af analyse . . . . .	21
Krav 5: Kommandobaseret interface . . . . .	23
Krav 6: Resultater i HTML-format. . . . .	23
Krav 7: Platform . . . . .	24
Krav 8: Håndtering af HTML/XHTML . . . . .	25
Krav 8.1: Indkodning . . . . .	27
<b>3 Brugertest</b>	<b>28</b>
Opgaver . . . . .	29
Opgave 1: Simpel analyse . . . . .	29
Opgave 2: Analyse med maks. dybde . . . . .	29
Opgave 3: Begrænset analyse . . . . .	29
Valg af forsøgspersoner . . . . .	30
Afvikling af brugertest . . . . .	30
Opgave 1 . . . . .	30
Opgave 2 . . . . .	30
Opgave 3 . . . . .	30
Konklusion på brugertest . . . . .	31

## 1 Strategi

Vores test er delt i to dele, Den første del — funktionstests — skal afgøre om programmet opfylder de krav vi har stillet i kravspecifikationen og at alle disse funktioner virker korrekt. Hvor kravene ikke er opfyldt (måske fordi det er blevet besluttet at de ikke var værd at implementere) vil dette blive noteret. Det optimale ville være hvis denne afprøvning kunne automatiseres, men i de fleste af testene vil det, grundet vores output i form af menneske-orienteret HTML, ikke kunne lade sig gøre uden at bruge mere tid på det end vi har tilgængelig. Hvis man eksempelvis vil automatisere afprøvning af LIX-tal implementationen ville det teoretisk kunne gøres ved at analysere programmets HTML-output (evt. via et XML-queriesprog), men da outputformatet ikke er fast defineret vil det for det første være en yderst skrøbelig blackbox-test, og for det andet være uhyre besværligt. For næsten alle funktionstestene vil der være en test af ugyldig inddata, for at se om programmet giver en brugbar fejlmeddelelse. Vi har dog valgt at samle dette under et afsnit.

Den anden del af testen — brugertesten — skal bruges til at finde ud af om brugervejledningen er forståelig og om brugeren kan finde ud af at bruge programmet ved at læse brugermanualen. Derudover skal testen bruges til at finde ud af om resultatsiderne er vanskelige at læse.

Bemærk at en del tests omhandler muligheden for at brugeren kan slå dele af analysen fra — i disse tilfælde vil det ikke blive testet at den relevante del ikke bliver slået fra når det ikke er angivet — f.eks. vil det i forbindelse med afprøvningen af muligheden for at slå LIX-tal fra ikke også blive afprøvet at LIX-tal bliver vist når de ikke er slået fra, og når det testes at stavfejl angivet som forkortelser ikke bliver markeret som stavfejl testes det *ikke* om forkert stavede ord markeres som stavfejl, når de ikke er angivet som forkortelser. Det forventes at de dedikerede tests for den relevante feature vil afprøve dette generelle scenarie. Det testes altså ikke hvad der sker når features *ikke* aktiveres.

Det absolut vigtigste funktionalitet i vores program er naturligvis den korrekte omdannelse af HTML-dokumenter til parsetræer, leksikalske enheder og analyseresultater - det er vigtigt at analyserne er korrekte, og for at de kan være det er det nødvendigt at opdelingen i sætninger og ord er korrekt. Derfor vil vi gå mere op i at afprøve denne funktionalitet end mere undværlige features som de forskellige konfigurationsmuligheder.

Kørselsresultater, såvel som inddata, er at finde sammen med vores kode i undermappen `functionalitytesting`.

## 2 Kravtest

Vi vil systematisk gennemgå de krav vi har stillet til vores programs funktionalitet og beskrive hvorledes hvert enkelt krav kan afprøves. Vi henviser til kravspecifikationen for en uddybning af kravene, følgende afsnit vil kun omhandle hvorledes de kan testes.

## Krav 1: Analyse af et helt websted

Det testrelevante i dette krav er programmets evne til at følge links rundt på det angivne website, og ikke at følge links der fører ud fra websitet. Andre tests dækker hvorvidt den resulterende analyse er korrekt, formålet med denne test er udelukkende at undersøge om alle sider rent faktisk bliver besøgt. Det er også relevant at sikre at sider ikke bliver analyseret mere end én gang, og at cirkulære links (side *A* linker til side *B* der igen linker tilbage til side *A*). ikke forårsager at programmet går i et uendeligt loop. Det kan ses hvilke sider der er blevet analyseret ved at se på den genererede side-liste, hvilket betyder at denne test er afhængig af at HTML-output (krav 6) fungerer korrekt.

### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
1	Et website der ikke indeholder links, og kun én side.	Kun den ene side analyseres.	Som forventet.	✓
2	Et website der ikke indeholder interne links, men links til sider på andre websites.	Kun websitets egen sider skal analyseres.	Som forventet.	✓
3	Et website der indeholder interne links uden cirkularitet, og hvor hver side kun kan findes af én sti.	Alle tilgængelige sider skal analyseres.	Som forventet.	✓
4	Et website der indeholder interne links uden cirkularitet, og hvor hver side kan tilgås af flere forskellige veje.	De tilgængelige sider skal kun analyseres én gang hver.	Som forventet.	✓
5	Et website der indeholder interne links med cirkulære links.	Hver side skal kun analyseres én gang, og programmet skal terminere.	Som forventet.	✓

### Krav 1.1: Respekter robots.txt

Programmet skal tjekke det angivne websteds robots.txt og sørge for at sider angivet i filen ikke bliver hentet og analyseret. Det giver kun mening at teste dette krav såfremt det er sikret at krav 1 (og krav 6) virker korrekt. Undersøgelsen af hvilke sider der ender med at blive analyseret kan

foregå på samme måde som i krav 1.

I de angivne tests adskilles der ikke mellem forbud hvor alle web-crawlere har forbud mod at besøge en side, og de, hvor vores program specifikt har forbud. Vi opfatter denne distinktion som så teknisk at den i højere grad hører hjemme i whitebox unit-tests, især fordi denne distinktion ikke er nævnt i krav 1.1.

## Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
6	Et website, der ikke har en robots.txt-fil.	Alle tilgængelige sider analyseres.	Som forventet.	✓
7	Et website hvor programmet har forbud mod at besøge nogen sider overhovedet.	Ingen sider bliver analyseret, programmet giver en fejlmeddelelse.	Som forventet.	✓
8	Et website hvor programmet har forbud mod at besøge en side der har eksklusive links til undersider på det analyserede website.	Den forbudte side analyseres ikke, og ej heller gør de sider som kun er tilgængelige via den forbudte side.	Som forventet.	✓
9	Et website hvor programmet har forbud mod at besøge en side der har ikke-eksklusive links til undersider (som kan tilgås via andre, ikke-forbudte, sider) på det analyserede website.	Den forbudte side analyseres ikke, men det gør de sider som den linker til.	Som forventet.	✓

## Krav 1.2: Dybde af crawling

Hvis man angiver en maksimal dybde af analysen (reelt, hvor dybt links skal følges, idet forsiden er dybde 0), så skal programmet respektere denne og gøre det korrekt.

## Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
10	Ingen angivet dybde og et 3 sider dybt website.	Alle sider skal analyseres.	Som forventet.	✓
11	Maksimal dybde 0 angivet og et 3 sider dybt website.	Kun den øverste side (forsiden) skal analyseres.	Som forventet.	✓
12	Maksimal dybde 1 angivet og et 3 sider dybt website.	Alle sider, med undtagelse af de(n) dybeste side, skal analyseres.	Som forventet.	✓
13	Maksimal dybde 1 angivet og et 3 sider dybt website, hvor en side har både dybde 2 og dybde 1. Hvis sider bliver undersøgt i den rækkefølge de bliver set i vil den først blive undersøgt som dybde 2.	Alle sider op til dybde 2 skal analyseres, og den nævnte side der både har dybde 2 og dybde 1 skal også analyseres.	Som forventet.	✓

## Krav 2: Analysemetoder

Dette krav kan i sig selv ikke testes, da det fungerer som “paraplykrav” for 3 underkrav. Disse kan dog alle testes på omtrent samme måde. En side med indhold der er repræsentativt for en ækvivalensklasse (og som opfylder andre krav som f.eks. tegnindkodning) analyseres, og programmets output sammenlignes med det ønskede resultat. Denne test er således afhængig af at krav 6 omkring HTML-output er korrekt. Langt størsteparten af disse tests er baseret på korrekt at opdele tegnfølgerne på HTML-sider i afsnit, sætninger og ord, og at teste uddybende, selv med ækvivalensklasser, ville være en ekstremt omfangsrig opgave som vi hverken har tid eller grammatisk viden til at udføre. Derfor vil tests af opdelingen i ord og sætninger være rimeligt rudimentær. Det er også værd at notere at det fra et whitebox-perspektiv er unødvendigt at teste korrekt leksikalsk analyse for både LIX og FKRT, idet programmet er implementeret således at opdelingen i leksikalske enheder er uafhængig af analysemetoderne, men fra et blackbox perspektiv, som disse tests er designet ud fra, kan man ikke gå ud fra at det er tilfældet.

Når vi i det følgende henfører til at noget bliver “markeret” i programmets output (f.eks. “markering af stavfejl”) henfører vi til brugervejledningen for en beskrivelse af hvordan denne markering helt præcist ser ud.

### Krav 2.1: Læsbarhedsindeks, LIX

Det essentielle i dette krav er at teksten korrekt opdeles i sætninger og ord, og at det resulterende LIX-tal er korrekt. Idet vores outputformat ikke angiver LIX-tal for hver enkelt sætning kan denne sammenligning kun foregå for hele tekstafsnit, men vi er i stand til at bedømme om sætningsopdeling sker korrekt.

#### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
14	En side uden indhold.	Programmet skal køre som normalt, men output skal naturligvis ikke indeholde analyse af nogen specifik tekst.	Som forventet, men resultaterne for hele dokumentet og titlen er uden mening og burde ikke være med.	Delvist ok, fejlen er så lille og sker så sjældent at vi ikke har tænkt os at rette det.
15	En side med et enkelt afsnit indeholdende en enkelt simpel sætning.	Ordene i sætningen skal opdeles korrekt og det beregnede LIX-tal skal ligeledes være korrekt.	Som forventet.	✓
16	En side med et enkelt afsnit indeholdende en enkelt kompliceret sætning indeholdende parenteser, apostroffer og citationer (som ikke adskiller sætninger).	Ordene i sætningen skal opdeles korrekt og det beregnede LIX-tal skal ligeledes være korrekt.	Som forventet.	✓

17	En side med et enkelt afsnit indeholdende adskillige komplicerede sætninger indeholdende parenteser, apostroffer og citationer (som ikke adskiller sætninger).	Sætningerne i afsnittet og ordene i sætningerne skal opdeles korrekt og det beregnede LIX-tal skal ligeledes være korrekt.	Som forventet.	✓
18	En side med adskillige afsnit indeholdende adskillige komplicerede sætninger indeholdende parenteser, apostroffer og citationer (som ikke adskiller sætninger).	Afsnittene på siden, sætningerne i afsnittene og ordene i sætningerne skal opdeles korrekt og det beregnede LIX-tal skal ligeledes være korrekt.	Som forventet.	✓

## Krav 2.2: Flesch-Kincaid Readability Test (FKRT)

∞

Det essentielle i dette krav er at teksten korrekt opdeles i sætninger og ord, og at de resulterende FRE- og FKGL-tal er korrekte. Idet vores outputformat ikke angiver disse værdier for hver enkelt sætning kan denne sammenligning kun foregå for hele tekstafsnit, men vi er i stand til at bedømme om sætningsopdeling sker korrekt.

**OBS!** I udregningen af FRE og FKGL skal der bruges oplysninger om antallet af stavelser. I de fleste danske ord er antallet af stavelser lig antallet af vokaler, men der er mange fremmedord hvor dette ikke er tilfældet. Det franske "niveau" er således kun 2 stavelser, og vi har i programmet ikke taget højde for dette. Vi er opmærksomme på denne fejl og vi vil i testen ikke bruge ord hvor antallet af stavelserne er forskellige fra antallet af vokaler. Dette gør vi så vi kan teste at beregningen er korrekt, hvis man ser bort fra dette.



## Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
19	En side uden indhold.	Programmet skal køre som normalt, men output skal naturligvis ikke indeholde analyse af nogen specifik tekst.	Som forventet, men resultaterne for hele dokumentet og titlen er uden mening og burde ikke være med.	Delvist ok, fejlen er så lille og sker så sjældent at vi ikke har tænkt os at rette det.
20	En side med et enkelt afsnit indeholdende en enkelt simpel sætning.	Ordene i sætningen skal opdeles korrekt og de beregnede FKRT-værdier skal ligeledes være korrekte.	Som forventet.	✓
21	En side med et enkelt afsnit indeholdende en enkelt kompliceret sætning indeholdende parenteser, apostroffer og citationer (som ikke adskiller sætninger).	Ordene i sætningen skal opdeles korrekt og de beregnede FKRT-værdier skal ligeledes være korrekt.	Som forventet.	✓
22	En side med et enkelt afsnit indeholdende adskillige komplicerede sætninger indeholdende parenteser, apostroffer og citationer (som ikke adskiller sætninger).	Sætningerne i afsnittet og ordene i sætningerne skal opdeles korrekt og de beregnede FKRT-værdier skal ligeledes være korrekt.	Som forventet.	✓

23	En side med adskillige afsnit indeholdende adskillige komplicerede sætninger indeholdende parenteser, apostroffer og citationer (som ikke adskiller sætninger).	Afsnittene på siden, sætningerne i afsnittene og ordene i sætningerne skal opdeles korrekt og de beregnede FKRT-værdier skal ligeledes være korrekt.	Som forventet.	✓
----	---	--	----------------	---

### Krav 2.3: Stavekontrol

Hvis en side er angivet til dansk og en dansk ordbog til `aspell` (samt naturligvis selve programmet `aspell`) er installeret, så skal programmet kunne tjekke for stavefejl. I vores krav angiver vi ikke hvorledes stavekontrollen er implementeret, men i vores dokumentation beskriver vi at systemets `aspell`-installation benyttes, derfor mener vi at det er korrekt at basere blackbox-tests på eksistensen af `aspell`. Som med FKRT- og LIX-testene er afprøvning af stavekontrollen baseret på at køre programmet på en side og sammenligne resultatet med det korrekte resultat. Idet vi erkender at `aspell` ikke er perfekt tager vi forbehold for at der er en stor klasse af gyldige ord som vil blive markeret som forkerte. Vi kræver blot at vores program angiver de samme ord som forkert stavede som `aspell` ville hvis `aspell` blev kørt direkte.

### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
24	Side med dansk tekst på en maskine der ikke har <code>aspell</code> installeret.	Brugeren informeres om problemet men programmet fortsætter, ingen ord markeres som forkert stavede.	Som forventet.	✓
25	Side med dansk tekst på en maskine der har <code>aspell</code> installeret, men ingen dansk ordbog.	Ingen ord markeres som forkert stavede.	Som forventet.	✓

26	Side med dansk tekst indeholdende både korrekt og ukorrekt stavede ord på en maskine der har <code>aspell</code> og en dansk ordbog installeret.	Ord markeres efter deres korrekthed.	Som forventet.	✓
----	--	--------------------------------------	----------------	---

### Krav 2.3.1: Flere sprog

Det skal testes at programmet fungerer med en ikke-dansk `aspell`-ordbog. Krav 2.3-testen udføres for to forskellige sider indeholdende hhv. engelsk og tysk tekst.

### Resultater

11

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
27	Side med engelsk tekst på en maskine der ikke har <code>aspell</code> installeret.	Brugeren informeres om problemet men programmet fortsætter, ingen ord markeres som forkert stavede.	Som forventet.	✓
28	Side med engelsk tekst på en maskine der har <code>aspell</code> installeret, men ingen engelsk ordbog.	Ingen ord markeres som forkert stavede.	Som forventet.	✓
29	Side med engelsk tekst indeholdende både korrekt og ukorrekt stavede ord på en maskine der har <code>aspell</code> og en engelsk ordbog installeret.	Ord markeres efter deres korrekthed.	Som forventet.	✓
30	Side med tysk tekst på en maskine der ikke har <code>aspell</code> installeret.	Brugeren informeres om problemet men programmet fortsætter, ingen ord markeres som forkert stavede.	Som forventet.	✓

31	Side med tysk tekst på en maskine der har <code>aspell</code> installeret, men ingen tysk ordbog.	Ingen ord markeres som forkert stavede.	Som forventet.	✓
32	Side med tysk tekst indeholdende både korrekt og ukorrekt stavede ord på en maskine der har <code>aspell</code> og en tysk ordbog installeret.	Ord markeres efter deres korrekthed.	Som forventet.	✓

#### Krav 2.4: Gentagne ord

Det skal testes at to på hinanden følgende identiske ord på en side markeres som gentagne ord, dog ikke hvis de befinder sig i forskellige sætninger. Alle følgende tests kan godt udføres med samme HTML-dokument.

#### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
33	To på hinanden følgende ord, som ikke er ens.	De to ord skal ikke markeres som gentagne ord.	Som forventet.	✓
34	To på hinanden følgende ord, som er ens, og som befinder sig i samme sætning.	De to ord skal markeres som gentagne ord.	Som forventet.	✓
35	To på hinanden følgende ord, som er ens, og som befinder sig i to forskellige sætninger (adskilt af punktum, udråbstegn, osv).	De to ord skal ikke markeres som gentagne ord.	Som forventet.	✓

#### Krav 2.5: Beregning af sidesværhedsgrad

Ved en blackbox test kan vi ikke teste at der bliver beregnet en sidesværhedsgrad, da vi har valgt ikke at vise sidesværhedsgraden på siden, men bare brugt den til at farve og sortere siderne efter. Ydermere er sidesværhedsgraden ikke en kravdefineret størrelse. Beregningen af sidesværhedsgraden

er baseret på LIX- og FKRT-resultaterne, så afprøvning af dette krav giver kun mening hvis afprøvningen af LIX- og FKRT- er positiv.

Vi kan i stedet teste at oversigtssiden bliver sorteret efter sidernes sværhedsgrad. Dette kræver dog at vi på forhånd har en idé om den “rette” sidesværhedsgrad for en enkelt side, hvilket betyder af afprøvningen af dette bliver lidt vag og ulden i kanten fordi det er nødvendigt at inddrage sider som har “åbenlyst svær” og “åbenlyst nem” tekst. Idet sidesværhedsgraden kun bruges til at sortere resultaterne er det dog ikke kritisk at den er hundrede procent eksakt, og vi mener derfor også at vi kan slippe afsted med en så informel test som vi udfører her.

### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
36	Et website indeholdende tre sider, hvoraf en er “åbenlyst svær”, en er “åbenlyst nem” og den tredje er mellemsvær.	Den svære side skal tildeles den højeste sidesværhedsgrad, den mellemsvære en lavere sidesværhedsgrad, og den nemme den laveste sidesværhedsgrad.	Som forventet.	✓

### Krav 3: Analyse baseret på HTML-tags

Dette krav kan ikke testes for sig selv, idet det fungerer som “paraplykrav” for en række mere specifikke underkrav der omhandler behandling af specifikke HTML-tags. Afprøvning af disse underkrav foregår alle på samme måde — for hvert underkrav produceres en side indeholdende repræsentanter for de relevante ækvivalensklasser, programmet køres på disse sider, og resultatet sammenlignes med det forventede. Det er ikke nødvendigt at lave en side for hver eneste ækvivalensklasse. Disse krav afhænger af at krav 2 og krav 8 er opfyldt og fungerer korrekt.

#### Krav 3.1: em og strong

Dette krav har vi valgt ikke at opfylde, da vi under implementationen af vores program nåede frem til den vurdering at det ikke ville forbedre analysen på en meningsfyldt måde. Det vil derfor ikke blive testet.

**Krav 3.2: hN**

Dette krav har vi valgt ikke at opfylde, da vi under implementationen af vores program nåede frem til den vurdering at det ikke ville forbedre analysen på en meningsfyldt måde. Det vil derfor ikke blive testet.

**Krav 3.3: abbr og acronym**

I det følgende opfattes et ord som værende “angivet som forkortelse” hvis det befinder sig i et `abbr`- eller `acronym`-tag. Det skal testes at ord, der normalt ville markeres som stavefejl i det sprog som sidens tekst er angivet til at være skrevet i, ikke bliver opfattet som stavefejl hvis de er angivet som forkortelser. Dette krav er afhængigt af at krav 2.3 er opfyldt og fungerer.

**Resultater**

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
37	Et forkert stavet ord, der er angivet som forkortelse.	Ordet skal ikke markeres som stavefejl.	Som forventet.	✓
38	Et korrekt stavet ord, der er angivet som forkortelse.	Ordet skal ikke markeres som stavefejl.	Som forventet.	✓

**Krav 3.4: Citater skal ikke analyseres.**

Citater skal ikke medtages når sidesværhedsgraden beregnes. Afprøvning af dette foregår ved at fjerne den citerede tekst og undersøge om der bliver beregnet samme sidesværhedsgrad som når den citerede tekst er der. Denne afprøvningsmetode fungerer kun fordi stavefejl ikke påvirker sidesværhedsgraden, og derfor burde det ikke påvirke resultatet at citater indeholdende stavefejl fjernes.

**Resultater**

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
39	En side indeholdende et afsnit som er designeret som et citat via <code>blockquote</code> og som ikke indeholder stavefejl.	Sidesværhedsgraden skal være den samme om citatet findes på siden eller ej, men når citatet findes skal det vises i output.	Som forventet.	✓
40	En side indeholdende et afsnit som er designeret som et citat (via <code>blockquote</code> ) og som indeholder stavefejl.	Sidesværhedsgraden skal være den samme om citatet findes på siden eller ej, men hvis citatet findes skal det vises i output og de forkert stavede ord skal være markeret.	Som forventet.	✓

### Krav 3.5: Tekst i andre sprog angivet med (`lang`)

Det skal testes at ord der har fået et sprog angivet via HTML-attributten `lang` bliver stavekontrolleret efter det angivne sprog, om muligt. Hvis det angivne sprog ikke kendes skal ordene ikke markeres som værende stavet forkert.

### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
41	En side skrevet på dansk som indeholder et ord der ikke er kendt på dansk (og som ville blive markeret som stavefejl), men som har fået angivet en ugyldig sprogkode.	Ordet skal ikke markeres som stavefejl.	Som forventet.	✓

42	En side skrevet på dansk som indeholder et ord der ikke er kendt på dansk (og som ville blive markeret som stavefejl), men som har fået angivet en ukendt (men gyldig) sprogkode.	Ordet skal ikke markeres som stavefejl.	Som forventet.	✓
43	En side skrevet på dansk som indeholder et ord der ikke er kendt på dansk (og som ville blive markeret som stavefejl), men som har fået angivet en kendt sprogkode (ordbogen er installeret) og som ikke er en stavefejl på det angivne sprog.	Ordet skal ikke markeres som stavefejl.	Som forventet.	✓
44	En side skrevet på dansk som indeholder et ord der ikke er kendt på dansk (og som ville blive markeret som stavefejl), men som har fået angivet en kendt sprogkode (ordbogen er installeret) og som også er en stavefejl på det angivne sprog.	Ordet skal markeres som stavefejl.	Som forventet.	✓

### Krav 3.6: kbd, var og code

Ord angivet med disse tags (herefter refereret til som “kode-ord”) bør ikke stavekontrolleres og ikke tælles med når sidesværhedsgraden beregnes. Afprøvningen af dette er praktisk identisk med afprøvningen af citat-håndteringen (se afprøvning af krav 3.4). Det bliver testet af hvert eneste af ovenstående tags behandles korrekt (identisk).

### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
-----	---------	--------------------	------------------	------------



45	En side med dansk tekst, hvori dele af sætninger er designeret som værende kode-ord (med <code>kbd</code> ), og disse kode-ord er korrekt stavede ord på dansk, samt en side med identisk indhold, bortset fra at kode-ordene er fjernet.	De to sider har identisk sidesværhedsgrad, kode-ordene er at finde i output når de er til stede på HTML-siden.	Kode-ordene bliver talt med når sidesværhedsgraden beregnes, de to sider har forskellig sidesværhedsgrad.	Grundet tidsnød blev det valgt ikke at rette dette, det blev opfattet som en mindre detalje.
46	En side med dansk tekst, hvori dele af sætninger er designeret som værende kode-ord, og disse kode-ord (med <code>kdb</code> ) er forkert stavede ord indenfor sidens angivne sprog, samt en side med identisk indhold, bortset fra at kode-ordene er fjernet.	De to sider har identisk sidesværhedsgrad, kode-ordene er at finde i output når de er til stede på HTML-siden. Kode-ordene skal ikke være markeret som stavfejl.	Kode-ordene bliver talt med når sidesværhedsgraden beregnes, de to sider har forskellig sidesværhedsgrad, men kode-ordene markeres dog ikke som stavfejl.	Grundet tidsnød, og det faktum at den vigtigste detalje (at kode-ordene ikke bliver markeret som stavfejl) fungerer, valgte vi ikke at rette dette.
47	En side med dansk tekst, hvori dele af sætninger er designeret som værende kode-ord (med <code>var</code> ), og disse kode-ord er korrekt stavede ord på dansk, samt en side med identisk indhold, bortset fra at kode-ordene er fjernet.	De to sider har identisk sidesværhedsgrad, kode-ordene er at finde i output når de er til stede på HTML-siden.	Kode-ordene bliver talt med når sidesværhedsgraden beregnes, de to sider har forskellig sidesværhedsgrad.	Grundet tidsnød blev det valgt ikke at rette dette, det blev opfattet som en mindre detalje.
48	En side med dansk tekst, hvori dele af sætninger er designeret som værende kode-ord, og disse kode-ord (med <code>var</code> ) er forkert stavede ord indenfor sidens angivne sprog, samt en side med identisk indhold, bortset fra at kode-ordene er fjernet.	De to sider har identisk sidesværhedsgrad, kode-ordene er at finde i output når de er til stede på HTML-siden. Kode-ordene skal ikke være markeret som stavfejl.	Kode-ordene bliver talt med når sidesværhedsgraden beregnes, de to sider har forskellig sidesværhedsgrad, men kode-ordene markeres dog ikke som stavfejl.	Grundet tidsnød, og det faktum at den vigtigste detalje (at kode-ordene ikke bliver markeret som stavfejl) fungerer, valgte vi ikke at rette dette.

49	En side med dansk tekst, hvori dele af sætninger er designeret som værende kode-ord (med <code>code</code> ), og disse kode-ord er korrekt stavede ord på dansk, samt en side med identisk indhold, bortset fra at kode-ordene er fjernet.	De to sider har identisk sidesværhedsgrad, kode-ordene er at finde i output når de er til stede på HTML-siden.	Kode-ordene bliver talt med når sidesværhedsgraden beregnes, de to sider har forskellig sidesværhedsgrad.	Grundet tidsnød blev det valgt ikke at rette dette, det blev opfattet som en mindre detalje.
50	En side med dansk tekst, hvori dele af sætninger er designeret som værende kode-ord, og disse kode-ord (med <code>code</code> ) er forkert stavede ord indenfor sidens angivne sprog, samt en side med identisk indhold, bortset fra at kode-ordene er fjernet.	De to sider har identisk sidesværhedsgrad, kode-ordene er at finde i output når de er til stede på HTML-siden. Kode-ordene skal ikke være markeret som stavfejl.	Kode-ordene bliver talt med når sidesværhedsgraden beregnes, de to sider har forskellig sidesværhedsgrad, men kode-ordene markeres dog ikke som stavfejl.	Grundet tidsnød, og det faktum at den vigtigste detalje (at kode-ordene ikke bliver markeret som stavfejl) fungerer, valgte vi ikke at rette dette.

### Krav 3.7: bdo

Det skal testes at ord angivet som havende ikke-standard tegnretning (højre mod venstre) bliver vendt om før de bliver stavekontrolleret. Ydermere skal det kontrolleres at kun det inderste af på hinanden indlejrede `bdo`-tags benyttes til at finde tekstretningen (to gange "højre mod venstre" giver altså ikke "venstre mod højre").

### Resultater

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
-----	---------	--------------------	------------------	------------

51	En side skrevet på dansk med et ord angivet som værende skrevet med “højre mod venstre” tegnretning, som ville være stavefejl på dansk hvis det ikke blev vendt, men som ikke ville være en stavefejl hvis det blev vendt.	Ordet skal vendes og ikke markeres som stavefejl.	Som forventet.	✓
52	En side skrevet på dansk med et ord angivet som værende skrevet med “højre mod venstre” tegnretning, som ville være stavefejl på dansk hvis det blev vendt, men som ikke ville være en stavefejl hvis det ikke blev vendt.	Ordet skal vendes og markeres som stavefejl.	Som forventet.	✓
53	En side skrevet på dansk med et ord angivet som værende skrevet med “højre mod venstre” tegnretning, som ville være stavefejl på dansk hvis det ikke blev vendt, og som ville være en stavefejl hvis det blev vendt.	Ordet skal vendes og markeres som stavefejl.	Som forventet.	✓
54	En side skrevet på dansk med et ord angivet som værende skrevet med “højre mod venstre” tegnretning, som ikke ville være stavefejl på dansk hvis det ikke blev vendt, og som ikke ville være en stavefejl hvis det blev vendt.	Ordet skal vendes og ikke markeres som stavefejl.	Som forventet.	✓

#### Krav 4: Konfiguration af analyse

Dette krav har vi af simplicitetshensyn slækket lidt på — når analyser “slås fra” udføres de stadigvæk, og benyttes stadigvæk i beregningen af sidesværhedsgraden, men de vises ikke i programmets output.

Kravet kræver også at man kan undtage elementer fra analyse baseret på deres `id` og `class`-attributter. I vores implementation endte vi med

kun at understøtte frafiltrering baseret på id-attributten, men til gengæld også understøtte frafiltrering baseret på tagnavnet.

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
55	Den samme website to gange, med visning af LIX-tal slået hhv. fra og til.	De to analyser producerer identiske resultater, men i den hvor visning af LIX-tal er slået fra figurerer LIX-tallene ikke i analysen.	Som forventet.	✓
56	Den samme website to gange, med visning af FKGL-tal slået hhv. fra og til.	De to analyser producerer identiske resultater, men i den hvor visning af FKGL-tal er slået fra figurerer FKGL-tallene ikke i analysen.	Som forventet.	✓
57	Den samme website to gange, med visning af FRE-tal slået hhv. fra og til.	De to analyser producerer identiske resultater, men i den hvor visning af FRE-tal er slået fra figurerer FRE-tallene ikke i analysen.	Som forventet.	✓
58	En side uden stavfejl, og hvor mærkning af stavfejl er slået fra i analysen.	Ingen ord markeres som stavfejl i output.	Som forventet.	✓
59	En side med stavfejl, og hvor mærkning af stavfejl er slået fra i analysen.	Ingen ord markeres som stavfejl i output.	Som forventet.	✓
60	En side der ikke indeholder gentagne ord, og hvor mærkning af gentagne ord er slået fra i analysen.	Ingen ord markeres som gentagne ord i output.	Som forventet.	✓
61	En side med der indeholder gentagne ord, og hvor mærkning af gentagne ord er slået fra i analysen.	Ingen ord markeres som gentagne ord i output.	Som forventet.	✓

62	En side der indeholder et element med en given <code>id</code> -attribut og en side der er identisk med den første, men ikke indeholder elementet med den relevante <code>id</code> -attribut. Programmet er sat til at filtrere elementet med den relevante <code>id</code> -attribut fra før analysen.	Analyseresultatet for de to sider er identisk.	Som forventet.	✓
63	En side der indeholder elementer, hver med en af flere givne <code>id</code> -attributter og en side der er identisk med den første, men ikke indeholder elementerne med de relevante <code>id</code> -attributter. Programmet er sat til at filtrere elementer med de relevante <code>id</code> -attributter fra før analysen.	Analyseresultatet for de to sider er identisk.	Som forventet.	✓
64	En side der indeholder et element med en givent tag og en side der er identisk med den første, men ikke indeholder elementet med det relevante tag. Programmet er sat til at filtrere elementet med det relevante tag fra før analysen.	Analyseresultatet for de to sider er identisk.	Som forventet.	✓
65	En side der indeholder elementer, hver med en af flere givne tags og en side der er identisk med den første men ikke indeholder elementerne med de relevante tags. Programmet er sat til at filtrere elementer med de tags fra før analysen.	Analyseresultatet for de to sider er identisk.	Som forventet.	✓

### Krav 5: Kommandobaseret interface

Det er temmelig åbentlyst hvad der skal testes her: det skal være muligt at starte en analyse direkte fra en kommandolinje.

### Krav 6: Resultater i HTML-format.

Det skal kontrolleres at resultaterne gemmes i HTML-format og at der genereres en index-side der giver adgang til alle de analyserede undersider, som skal indeholde en vis mængde information om det analyserede website. Afprøvningen af dette krav har et vist overlap med afprøvningen af krav 1, og de to krav er gensidigt afhængige af at hverandre fungerer korrekt for at deres tests kan udføre. Afprøvningen af dette krav vil ikke gå op i hvorvidt analyseresultaterne er korrekte, kun at de findes i programmets output.

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
66	Et website indeholdende en enkelt side.	Den resulterende index-fil skal være beliggende i en mappe med navn efter det domæne som websitet er beliggende på, og være navngivet <code>index.html</code> . Filen skal indeholde et link til en enkelt analyseside, som også ligger inde i mappen sammen med index-filen.	Som forventet.	✓
67	Et website indeholdende en enkelt side indeholdende flere afsnit og sætninger.	Analysesiden for websitets ene side skal indeholde en liste over afsnit på siden, hvert afsnit skal indeholde analyseresultaterne for siden, og hver sætning i afsnittet skal være farvekodet efter dets sværhedsgrad.	Som forventet.	✓

## Krav 7: Platform

Vi mener ikke at dette program kan testes på en systematisk måde — for det første er det valgt (af nødvendighed, det er meget svært at definere en præcis platform), og for det andet er det ikke rimeligt at forvente at vi har en meningsfyldt måde at reagere på u-understøttede platforme i programmet. I vores kravspecifikation nævner vi som eksempel at programmet kan køre på DIKU's maskiner, men da vores program er afhængigt af en SML/NJ<sup>1</sup>-installation (som ikke findes på DIKU's system) kan dette heller ikke uden videre gøres. Ydermere understøtter vi ikke proxy forbindelser, hvilket også forhindrer at teste på DIKU's system.

Dog har vi testet programmet som en “standalone” (SML/NJ runtime fil, heap image og et shell script til starte programmet) på maskiner uden SML/NJ installeret. Disse filer er leveret sammen med koden, og er beskrevet i brugsvejledningen.

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
68	Kørsel program på GNU/Linux-baseret system, uden SML/NJ installeret, med direkte forbindelse til Internettet.	Programmet virker og kan hente og analysere en hjemmeside.	Som forventet.	✓

## Krav 8: Håndtering af HTML/XHTML

Det er ikke muligt at lave en dybdegående test af dette krav - mængden af ækvivalensklasser indenfor gyldige HTML-dokumenter ville indeholde alle permutationer af alle gyldige HTML-elementer, en ekstrem størrelse. Derfor nøjes vi med at afprøve repræsentative elementer (f.eks. minder `br` og `hr` en del om hinanden i brug), og ikke inddrage hvert eneste tag. Vi placerer så at sige vores ækvivalensklasser på element-niveau i stedet for på dokument-niveau.

Det gør det yderligere svært at black-box-teste dette at der ikke kan trækkes en direkte linje fra HTML-parsetræet til programmets output, så de testede ækvivalensklasser er nogen særligt udvalgte som ville have en direkte effekt på analyseresultatet såfremt de ikke blev parset ordentligt.

I vores krav nævner vi behovet for at tage særlig hånd om `style` og `script`-elementer, idet disse kan indeholde HTML-lignende tegn som kan forstyrre parseren. Under implementationen fandt vi frem til at det ville være en fejlallokering af ressourcer at implementere dette, idet problemet

<sup>1</sup><http://www.smlnj.org/>

var noget mere komplekst end først antaget, og at det desuden ikke er så stort i praksis, idet mange HTML-sider enten refererer til eksterne scripts og stylesheets, eller omgiver dem med HTML-kommentarer for ikke at forvirre ældre browsere. Disse features vil således ikke blive testet.

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
69	En side på dansk indeholdende ord, der ikke er korrekt stavede danske ord, som er angivet som forkortelser og indlejret i <code>em</code> - og <code>strong</code> -tags.	De ord der er angivet som forkortelser er ikke markeret som stavfejl.	Som forventet.	✓
70	En fil opdelt i afsnit via <code>p</code> -tags, men hvor ingen <code>p</code> -tags bliver eksplicit afsluttet. <sup>2</sup>	Analyseresultat-filen viser en korrekt opdeling i afsnit, som hvis sidens <code>p</code> -tags var blevet eksplicit afsluttet	Som forventet	✓
71	En side indeholdende teksten <code>&lt;br lang=en&gt;</code> i et dansk afsnit. <sup>3</sup>	Forventet uddata er en korrekt dansk stavekontrol efter <code>br</code> -tagget	Som forventet	✓
72	En side indeholdende korrekt stavede danske ord indeholdende danske specialtegn, hvor disse specialtegn er repræsenteret ved <i>HTML-entities</i> ( <code>&amp;oslash;</code> , <code>&amp;aelig;</code> , <code>&amp;aring;</code> ), og det derfor er nødvendigt at erstatte disse entities med de tilsvarende tegn for at opnå en korrekt stavekontrol.	Ingen af de danske ord er markeret som forkert stavede.	Som forventet.	✓

<sup>2</sup>I HTML er det tilladt ikke at afslutte f.eks. `p`-tags, de opfattes i stedet som implicit afsluttede næste gang der bliver åbnet et *block*-tag

<sup>3</sup>`br`-tagget er ikke afsluttet, idet dette ikke er nødvendigt i traditionel HTML



73	2 identiske filer, hvoraf en af filerne dog indeholder lange HTML-kommentarer.	Analyseresultat for filerne er ens med henblik på dokumentresultatet og de individuelle afsnits-resultater, indholdet af HTML-kommentarerne skal ikke påvirke analyseresultatet.	Som forventet.	✓
----	--	--	----------------	---

### Krav 8.1: Indkodning

Undervejs i programmets udvikling skiftede vi SML-implementation fra Moscow ML til SML/NJ, hvilket også ændrede dette krav. I stedet for at understøtte Moscow ML's indbyggede indkodning understøtter vi explicit Latin-1.

25

Nr.	Inddata	Forventet resultat	Faktisk resultat	Konklusion
74	En HTML fil indkodet i Latin-1, direkte indeholdende danske specialtegn som del af danske ord.	Korrekt stavekontrol af de danske ord der indeholder danske specialtegn indkodet efter Latin-1.	Som forventet.	✓

## Funktionstest-konklusion

Generelt opfylder vores program kravene korrekt. De få fejl der eksisterer er relativt minimale, og forhindrer ikke programmet i at være et brugbart værktøj til typiske opgaver. Grundet den ekstremt store mængde af mulige input-data kan det ikke lade sig gøre at lave en bare tilnærmelsesvist dækkende test (især ikke når man overvejer kombinationer), og vi har således afprøvet programmet baseret på inddata der kun tester en lille del af programmet af gangen, og ikke håndteringen af kombinationer af forskellige grænsetilfælde.

## 3 Brugertest

Brugertesten skal afsløre brugervenlighedsproblemer med vores program og skal samtidigt bruges til at evaluere detaljeniveauet i vores brugermanual. Vi vil udføre testen som et tænke-højt forsøg hvor brugeren skal udføre nogle opgaver og samtidigt fortælle hvad personen tænker og gør. Vi vil undervejs ikke hjælpe brugeren med ting der omhandler vores program, men hvis der er spørgsmål til andre ting, så vil vi gerne hjælpe.

Man kan sige at programmet brugsmæssigt er delt i to. Den ene del handler om at starte en analyse og den anden del om at aflæse resultatet. I udførelsen af brugertesten vil vi ikke direkte lave denne adskillelse, men det er sådan vi vil evaluere resultaterne.

## Opgaver

Dette afsnit indeholder de opgaver vi vil stille brugeren.

### Opgave 1: Simpel analyse

- Sæt en analyse i gang af <http://dybber.dk/>
- Åben analyseresultatet når analysen er afsluttet.
- Hvilken side er den sværeste på hjemmesiden?

### Opgave 2: Analyse med maks. dybde

- Start en analyse af <http://da.wikipedia.org/> hvor kun forsiden analyseres, dvs. der skal ikke følges nogen links.
- Åben analyseresultatet og klik ind på den analyserede side.
- Hvad er sidens læsbarhedsindeks/LIX-tal?
- Er der nogen stavefejl på siden?
- Er der nogen ord der fejlagtigt er gentaget?

### Opgave 3: Begrænset analyse

- Start en analysen på samme måde som sidste opgave, dog skal analysen begrænses så HTML-tags med `id="column-one"` ikke medtages i analysen.
- Hvad er sidens læsbarhedsindeks nu?

### Valg af forsøgspersoner

Til at udføre vores brugertest skal vi bruge en eller flere personer indenfor vores målgruppe. I kravspecifikationen er vores målgruppe specificeret som:

En person i vores målgruppe er en hjemmesideskribent der er bekendt med HTML. Personen har en professionel interesse i at teksten er læsbar, således at vedkommende selv vil sætte sig ind i betydningen af de udførte analysers resultater.

I kravspecifikationen står der også at programmet skal være udformet som et kommandolinjeværktøj og at det skal kunne køre på GNU-baserede Linux maskiner. Dette stiller herved ét yderligere krav til forsøgspersonerne: de skal have forudgående kendskab til kommandolinjebaserede programmer.

### Afvikling af brugertest

Onsdag den 6. juni afviklede vi vores brugertest. Vi havde fået fat i Anders Boesen Lindbo Larsen, der indvilgede i at afprøve vores program. Han går ligesom os på første år på DIKU. Han kendte både til HTML, Linux og kommandobaserede programmer, så han opfylder alle vores krav til en forsøgsperson. Vi satte ham i gang med opgaverne og viste ham brugermanualen.

### Opgave 1

Det var ikke svært for ham at få lavet første del af opgave 1 dog var blev han lidt forvirret og troede at analysen var færdig, da programmet i et stykke tid ikke skrev noget på skærmen, fordi det analyserede en stor side. Da han skulle kigge på index-siden og for at finde ud af hvilken side der var sværest gik han lidt i stå. Der stod intet sted på siden at hjemmesiderne var sorteret efter sidernes sværhedsgrad og at den sværeste side stod øverst. Han endte med at lave et gæt på at den sværeste side ville stå øverst.

### Opgave 2

For at få løst opgave to måtte han kigge i manualen. Han fandt hurtigt "-d" indstillingen der angiver den maksimale dybde, men han misforstod forklaringen. Han skrev "-d 1" hvor han skulle have skrevet "-d 0" for at få programmet til kun at analysere en side. Resten af opgaven gik uden problemer, selvom han var lidt hurtig og ikke fik læst toppen af resultatsiden, hvor der stod hvordan han kunne identificerede stavfejl mm.

### Opgave 3

I opgave 3 skete der adskillige fejl. Først misforstod han manualen og brugte “-ignore-tag” i stedet for “-ignore-id” til at fra sortere Wikipedia’s menuer. Derudover brugte han et forkert argument til funktionen. Han kopierede direkte fra opgaven og skrev `id="column-one"`, meningen var at han skulle nøjes med `column-one`. Han fandt efter noget tid ud af at gøre det på den rigtige måde. De fejl der blev lavet her mener vi ikke skyldes at vores manual ikke var god nok, han var utålmodig og brugte ikke tid på at læse grundigt i manualen. Vi vil dog sige at opgaven kunne være formuleret bedre, den skulle nok have lagt mere op til at han ikke skulle have medtaget `id=""`.

### Konklusion på brugertest

Vi har fundet 3 brugervenlighedsproblemer i vores projekt.

1. Det er svært at se om programmet arbejder når det analyserer en stor side med mange links. Det kan se ud som om at det er gået ned. Dette problem ville kunne løses ved at udskrive et punktum hvert andet sekund så længe programmet analyserer en side. Vi har ikke rettet dette fordi det ville kræve at programmet var flertrådet eller at vi brugte en system-timer, begge løsninger vil tage
2. Når man ser index-siden er der ingen information om hvordan siderne er sorteret og om de overhovedet er sorteret. Der er heller ikke nogen forklaring af farverne. Dette kan nemt løses ved at tilføje noget forklarende tekst.
3. I manualen var forklaringen af dybdeangivelsen tvetydig. Dette er rettet i den endelige manual.