Interaktiv Test

DDU min eksamensprojekt



**Gruppemedlemmer**: Cecilia Nygaard, Elisabeth Astrup Christensen, Jessie Lorentzen & Nicoline Mejlgren von Bülow

**Vejledere**: Anders Juul Refslund & John Gerhard Jensen

**Skole**: H.C. Ørsted Gymnasium Lyngby

**Periode**: 20/10-2020 - 29/11-2020

**Fag**: Digital design og udvikling

# Titelblad

**Projektets titel**: Interaktiv test

**Gruppemedlemmer**: Cecilia Nygaard, Elisabeth Astrup Christensen, Jessie Lorentzen & Nicoline Mejlgren von Bülow

**Vejledere**: Anders Juul Refslund & John Gerhard Jensen

**Skole**: H.C. Ørsted Gymnasium Lyngby

**Periode**: 20/10-2020 - 29/11-2020

**Fag**: Digital design og udvikling

**Underskrifter**:






# 

# Resumé

# I denne opgave har vi forsøgt at løse en problematik, der relaterer til, at lærer skal kunne bedømme elevernes faglige niveau og udvikling i gymnasiet. For at påvise hvordan vi bedst kunne løse denne problematik, har vi undersøgt, hvad lærere ville stille som krav og bygget vores produkt op efter dette. Det var for eksempel vigtigt for lærerne, at de selv kunne generere test og skræddersy disse til deres hold/elever. Vi har derfor taget disse overvejelser i betragtning for at designe en løsning, der bedst muligt opfyldte lærernes behov. Desuden har vi undersøgt hvilke andre krav, det vil være relevant at stille til et produkt, hvor vi fandt frem til krav om design, brugervenlighed, tilgængelighed af vores program og sikkerhed. Disse krav har vi forsøgt at opfylde ved at bruge bløde former og pastelfarver på vores hjemmeside, at bruge web konventioner, at lave en hjemmeside, som så vidt muligt er tilgængelig for alle, samt ved at hashe passwords for at sikre brugerens sikkerhed.

Vi har lavet en test-service, der kan hjælpe lærerne med at se deres elevers faglige udvikling. Vores test-service fungerer ved, at læreren kan logge ind og lave multiple-choice tests, som kan lægges ud til et hold af elever, som også selv kan logge ind og tage denne test. Læreren kan se en oversigt over, hvordan de forskellige elever på deres hold har klaret sig, samt hvordan deres udvikling er. Læreren kan lave hold ud fra et Excel-ark af elever, som er meldt til. Når eleverne er tilføjet til disse oprettede hold, vil læreren og eleverne kunne få adgang til elevernes testresultater. Eleverne kan selv oprette sig som brugere, når de logger ind på siden for første gang. Eleverne kan tage disse tests, som lærerne ligger ud til dem, og se deres egen udvikling.

# Indholdsfortegnelse

[**Titelblad**](#_x6mpp161urpz) **1**

[**Resumé**](#_g6jocomb4l9l) **2**

[**Indholdsfortegnelse**](#_h828zfiwq5o9) **3**

[**Problemidentifikation**](#_7zy4kq6kwlis) **4**

[**Problemanalyse**](#_zieu1ym1ohkt) **4**

[Problemformulering](#_cjh48uz0ez) 5

[Løsningsforslag:](#_ct0c5teaz8k8) 6

[Udformning af krav:](#_ctrkdkeahdd6) 6

[Krav og vægtning af krav til kravmatrix:](#_j5m9tuc3u9do) 7

[Begrundelse for vægtning af krav til kravmatrix](#_j92v25i1dcnw) 7

[Kravmatrix:](#_w80ekmwcwut0) 8

[Begrundelse for pointfordeling i kravmatrix](#_bufakpm0mgnh) 8

[Valg af løsningsforslag](#_pc0mq2pl4t8s) 9

[**Produktudformning**](#_z25qll7i3l96) **10**

[**Produktbeskrivelse**](#_n5tcgic6fdak) **10**

[**Designbeskrivelse**](#_83frw5rb1hvu) **10**

[Analyse af design](#_hquo8jldgh29) 10

[Krav til design:](#_7cy7qrr0fhxt) 11

[**Kravspecifikationer**](#_f0i3gnkwppm3) **12**

[Kravspecifikationer:](#_4itpgkngjj68) 12

[**Projektforberedelse**](#_ion70wbopjph) **13**

[**Realisering**](#_9k7c1qsp1314) **15**

[**Perspektivering**](#_v73yajz3m5o1) **18**

[**Litteraturliste**](#_wxqdvxua2j2x) **20**

[**Bilag 1**](#_qbccfnjh8x7j) **21**

[**Bilag 2**](#_yycze7hm94f) **23**

# 

# Problemidentifikation

Når man er gymnasielærer, er det et krav, at man kender ens elevers faglige niveau, så man kan give dem en fair karakter. Hvis læreren ikke kender elevens reelle faglige niveau, kan læreren komme til at fejlbedømme eleven og dermed give for høj eller for lav en karakter, hvilket ikke er fair overfor eleven. Det vil derfor være praktisk at udarbejde et program, hvor lærere kan oprette tests og sende dem til deres elever på et hold, så lærerne kan få et overblik over elevernes faglige niveau ud fra deres testresultater. Derfor vil vi undersøge følgende i vores projekt:

Hvordan kan vi bedst udarbejde et program, som sikrer, at en lærer kan holde styr på sine elevers faglige niveau?

# Problemanalyse

I følgende afsnit vil vi undersøge hvilke funktioner, et program skal have for at opfylde, at en lærer kan få et bedre overblik over sine elevers faglige niveau, og hvordan sådan et program skal designes for at lærerne føler, at programmet virker optimalt. Dette vil vi undersøge ved at interviewe forskellige lærere på skolen og stille dem spørgsmål omhandlende hvordan de vil foretrække, at programmet designes, så lærerne bedst muligt kan få et overblik over deres elevers faglige niveau. Dette interview kan ses i bilag 1. Ud fra interviewet kan vi analysere os frem til følgende:

Dette programs målgruppe vil være lærere, da selve hovedkernen i programmet er, at lærere skal kunne få et overblik over deres elevers faglige niveau. I interviewet har vi interviewet gymnasielærere, og vi har undersøgt, om programmet primært skal målrettes mod gymnasielærere, der specifikt underviser elever i grundforløbet, eller om programmet skal målrettes alle gymnasielærere. Grunden til, at det kan tænkes, at programmet skal målrettes gymnasielærere, der underviser elever i grundforløbet, er at gymnasielærere har undervist deres grundforløbselever i væsentligt kortere tid end gymnasielærere, der underviser i et fag på A-, B- eller C-niveau, og dermed har læreren væsentligt kortere tid til at få en fornemmelse for elevernes faglige niveau, og derfor kan det i mange tilfælde være sværere at give en fair vurdering af elevernes faglige niveau. I interviewet fandt vi frem til, at det ikke er alle fag i grundforløbet, hvor læreren skal give karakter. Derfor vurderer vi, at vores målgruppe er alle gymnasielærere og ikke gymnasielærere, der specifikt underviser elever i grundforløbet.

Ud fra interviewet fandt vi desuden ud af, at det er et problem for nogle engelsklærere at inkludere funktionel grammatik i undervisningen. Derfor vurderer vi, at det kan være en god idé at inkludere en funktion i vores program, som gør det nemmere for engelsklærere at inkludere funktionel grammatik i undervisningen. Dette kunne for eksempel være en funktion, hvor der på forhånd er lagt prøver ind i programmet, som er kategoriseret efter grammatisk emner og forskellige bøger, som eleverne arbejder med.

Herudover fandt vi i interviewet ud af, at de fleste gymnasielærere helst selv vil generere de tests, som de frigiver til deres elever, men at det ville være godt, hvis læreren både selv kan generere tests, og at læreren kan vælge imellem nogle tests, som i forvejen eksisterer i programmet.

I forhold til om spørgsmålene skal være multiple choice eller åbent svar, fandt ud fra interviewet frem til, at enten åbent svar eller begge dele er bedst. Grunden til, at åbent svar måske er bedre end multiple-choice, skyldes, at der er en større chance for, at eleverne tilfældigvis svarer korrekt ved multiple choice, selvom de ikke kan svaret, hvor denne chance er en stor del mindre ved åbne svar.

Ud fra problemanalysen er vi kommet frem til, at programmets målgruppe er alle gymnasielærere. Herudover er vi kommet frem til, at det er vigtigt, at lærerne selv kan generere deres tests i programmet, i forhold til, at testene er generet på forhånd, og at lærerne foretrækker, at spørgsmålene enten har form som åbne svar eller både åbne svar og multiple choice. Derudover ville en yderligere funktion, der vil hjælpe engelsklærere med at inkludere funktionel grammatik i engelskundervisningen, være en god ekstra funktion til vores program.

## Problemformulering

Ud fra den opgivet opgaveformulering og krav, så har vi valgt at gå med en problemformulering, som lyder sålede “*Lav et program, som kan hjælpe lærere med at vurdere deres elevers faglige niveau.”*

**Produktprincip**

Formålet med dette afsnit er at komme frem til et løsningsforslag til, hvordan vi kan løse vores problemformulering bedst muligt. For at gøre dette vil vi først generere løsningsforslag ved at lave en brainstorm, hvorefter vi vil vælge det bedste løsningsforslag ved at anvende kravmatrix.

## Løsningsforslag:

Ud fra en kort brainstorm er vi kommet frem til følgende løsningsforslag:

* *En app til telefon, hvor en lærer kan oprette prøver til elever og herefter se, hvor godt eleven har klaret sig.*
* *Hjemmeside, hvor en lærer kan oprette prøver til elever og herefter se, hvor godt eleven har klaret sig.*
* *App til telefon og/eller computer med flashcards, hvor eleverne kan teste sig selv og vurdere hvor svære, de mener spørgsmålene er, og hvor læreren kan se deres vurdering.*

## Udformning af krav:

Et væsentligt krav til vores program er, at det skal være brugervenligt. Dette betyder, at vores program skal opfylde normale konventioner, herunder web konventioner, hvis vi vælger at lave en hjemmeside, og vores program skal være simpelt opbygget, så enhver bruger kan finde ud af at bruge vores program. Dette sikrer, at alle lærere og elever kan bruge vores uden at behøve at bruge lang tid på at sætte sig ind i, hvordan man bruger programmet.

Når man laver et program, hvor lærere skal kunne give mange forskellige elever tests, er det desuden vigtigt, at programmet er tilgængeligt for alle elever. Det nytter for eksempel ikke noget, hvis det kun er elever med en MacBook, som kan svare på testene; alle elever skal kunne svare på testene, uanset om de bruger Macbook, Windows computer eller noget helt tredje. Derfor vil *tilgængelighed* af vores program være et væsentligt krav til vores program.

Det er vigtigt, at vores program følger persondatalovgivningen, da vores program selvfølgelig skal overholde lovgivningen. Derfor vil endnu et krav til vores program være, at vores program er sikkert at bruge, hvilket blandt andet medfører, at koderne i vores program skal hashes.

## Krav og vægtning af krav til kravmatrix:

*Kravene vægtes fra 1-3. Vægtning af kravene bruges til beregning af totale point i kravmatrix.*

* *Brugervenlighed [****vægt: 2****]*
* *Tilgængeligt [****vægt: 3****]*
* *Sikkerhed [****vægt: 3****]*

## Begrundelse for vægtning af krav til kravmatrix

Grunden til, at vi har valgt at vægte *brugervenlighed* med 2 er, at det er vigtigt, at brugeren ved, hvordan man bruger programmet, og at det er nemt at bruge programmet. Dog er det vigtigere, at eleverne rent faktisk har adgang til programmet, så de kan udføre opgaverne, og derfor har vi vægtet tilgængelighed højere, og dermed har vi ikke vægtet brugervenlighed med 3.

Vi har valgt at vægte *tilgængelighed* med 3, da det er vigtigt, at alle har adgang til programmet, så det ikke er sådan, at nogle elever ikke kan udføre en test, da de fx har en Macbook og ikke en PC.

Vi har valgt at vægte *sikkerhed* med 3, da det er vigtigt, at vores program er sikkert, så vi følger data sikkerhedsloven.

## Kravmatrix:

*Tabel 1: Kravmatrix over løsningsforslag. Point fordeles fra 1-5.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***App*** | ***Hjemmeside*** | ***Flashcards*** |
| ***Brugervenlighed  Vægt: 2*** | *4* | *5* | *2* |
| ***Tilgængelighed Vægt: 3*** | *3* | *5* | *3* |
| ***Sikkerhed  Vægt: 2*** | *5* | *5* | *5* |
| ***I alt:*** | ***32*** | ***40*** | ***28*** |

## Begrundelse for pointfordeling i kravmatrix

I forhold til brugervenlighed har vi valgt at give hjemmesiden flest point, da programmets målgruppe er lærere og unge, og mange unge er ofte på internettet.[[1]](#footnote-1) Herudover har vi valgt at give appen næstflest point, da unge bruger også bruger en del apps.[[2]](#footnote-2) Vi har valgt at give flashcards færrest point, da der ikke er lige så mange unge, som benytter sig af flashcards som hjemmesider og apps, og derfor er unge ikke lige så vant til at bruge flashcards i forhold til hjemmesider og apps, og derfor vil de have nemmere ved at bruge hjemmesider og apps.

I forhold til tilgængelighed vil en hjemmeside være tilgængelig på både computer, telefon, iPad mm, hvilket de fleste elever i dag har adgang til. Derfor har vi valgt at give flest point til hjemmeside i forhold til apps og flashcards, som ofte også vil være app-baseret, som ikke altid kan fås til alle devices, herunder både Windows, Android og Apple devices.

I forhold til sikkerhed har vi valgt at give alle lige løsningsforslag lige mange point, da vi kan lave alle programmer sikre ved at kryptere adgangskoderne ordentligt, at have en god og sikker database, og at følge datasikkerheds lovgivningen.

## Valg af løsningsforslag

Ud fra vores kravmatrix får vi, at løsningsforslaget “hjemmeside” har fået flest point. Derfor går vi videre med dette løsningsforslag.

# 

# Produktudformning

*I dette afsnit vil vores produkt blive udformet. Dette gøres ved brug af user story, designbeskrivelse, samt kravspecifikationer.*

## Produktbeskrivelse

For at definere vores program har vi lavet følgende user story, som forklarer vores program:

**User story**

En hjemmeside, hvor lærere kan lægge tests op til et hold elever, og eleverne skal kunne tage testen to gange i alt. Efter eleverne har taget testen en gang, skal både lærere og elever kunne se elevernes resultater fra testen. Når eleven så har taget testen anden gang, skal lærer og elev også igen kunne se resultaterne for begge tests, og de skal kunne se udviklingen af elevens faglige kompetencer fra første gang de tog testen til anden gang, de tog testen. Udviklingen skal stå som et procenttal, som fortæller hvor mange procent bedre, eleven var anden gang, han/hun tog testen, i forhold til da han/hun tog testen første gang.

# Designbeskrivelse

*I følgende afsnit opstilles krav til design af vores hjemmeside.*

## Analyse af design

Det primære design fokus er, at vores hjemmeside skal udsende beroligende og behagelige signaler, så eleverne ikke bliver stressede af programmet, mens de tager deres tests. Dette vil vi opnå ved at have bløde former frem for at have hårde kanter på diverse knapper. Vi vil derfor benytte os mere af cirkler frem for firkanter, da cirkler symboliserer fokus, hamoni og dynamik.[[3]](#footnote-3) Ved brug af firkanter vil vi afrunde hjørnerne, så der ikke er nogen skarpe hjørner.

Udover dette vil vi også have, at farverne er rolige. Rolige farver kan opnås ved at have lysere nuancer, eksempelvis pastelfarver eller udtyndet versioner af farver.[[4]](#footnote-4) Dette vælger vi frem for hårde og skarpe farver for at berolige eleven, inden den kommende test går i gang.

Vi har på vores hjemmeside valgt at benytte os af loven om bløde former, som skal give en følelse af blødhed, rolighed og harmoni. Det har vi gjort, for at skabe et mere behageligt miljø for de studerende, som skal tage testen versus at have hårde kanter, som skaber et mere skarpt og streng brugeroplevelse.

Som farver på vores hjemmeside har vi valgt at bruge mere naturlige og rolige jordfarver, for igen at skabe en behagelig brugeroplevelse. Og får at gøre eleven mere rolig under prøven, se vores farvepalette på figur 1.

Figur 1: farvepalette

Som generelt design layout har vi

benyttet os af loven om åbenhed for at skabe et mere frit og luftigt design.

## Krav til design:

Ud fra vores analyse af design er vi kommet frem til følgende designkrav til vores program:

* *Beroligende farver*
* *Bløde former*
* *KISS*
* *Brug af web konventioner*
* *Balance ved hjælp af symmetri*
* *Kvadrater med afrundede hjørner & cirkler*

# 

# Kravspecifikationer

*I følgende afsnit opstilles nogle generelle krav til vores program, udover vores krav til design.*

## Kravspecifikationer:

Ud over kravene til design skal vores program desuden overholde følgende krav:

* *Brugervenlig.*

Vi vil gerne have, at vores program er så brugervenligt som muligt, så både lærer og elever i alle aldersgrupper har muligheden for at benytte sig af programmet. Dette kan blandt andet opnås ved brug af web konventioner og optimalt design (se designbeskrivelse & krav).

Dette krav måles via. brugertest.

* *Tilgængelig.*

Det er vigtigt, at hjemmesiden er nemt tilgængeligt, så elever kan komme på programmet gennem deres telefon, hvis de ikke har eller har glemt deres computer.

Dette krav efterprøves ved at se, om vores hjemmeside kan køre på forskellige devices, fx Android-telefon, iPhone, PC og MacBook.

* *Sikkerhed.*

Når man arbejder med folks persondata, er det også vigtigt, at vi har en form for sikkerhed på vores hjemmeside, så folk ikke risikerer at få delt eller stjålet kodeord, testresultater eller lignende. Vi hasher derfor passwordet (vi krypterer ikke, for vi har ikke behov for at kende det egentlige password igen senere; vi har kun behov for, at computeren kan finde ud af, om det tastede password passer til det krypterede password eller ej).

Vi sørger for at implementere sikkerhed i koden. Vi tjekker, at man rent faktisk kan logge ind som bruger med det hashede password.

# Projektforberedelse

*I følgende afsnit gennemgås de forskellige IT-værktøjer, vi har brugt som forberedelse af vores produkt, samt vores projektstyring.*

Som IT-værktøj til vores forberedende arbejde benyttede vi os blandt andet af Access. I Access udformede vi en skitse over databasen og relationerne i databasen. Access-filen kan findes under mappen “data” på GitHub.

Herudover valgte vi også at foretage nogle interviews med nogle af skolens lærere for at få et overblik over, hvordan vores program skulle designes. Disse interviews kan ses i bilag 1. Samtidig med at vi foretog os nogle interviews, så tegnede vi også nogle skitser af hvordan vi nogenlunde kunne tænke os at hjemmesiderne så ud, samt hvor henne diverse knapper og funktioner skulle være på de individuelle sider. Disse skitser kan ses i vores kontrakt i bilag 2. Vi lavede også et farveskema inden vi påbegyndte design programmeringen, for at have et ensartet design over hele hjemmesiden. Dette farveskema ses på figur 1.

Inden vi startede kodningen, planlagde vi så meget af projektet som muligt. Dette inkluderede at ligge tidsplaner og at lave en foreløbig arbejdsfordeling og tidsvurdering. Den meste af denne planlægning har vi lavet ved brug af GitHub. Her har det været let for os at overskue hvem der er ansvarlig for hvad, samt hvilke issues og tasks der var tilbage, hvilke der var i gang og hvilke der var færdige. Vi startede med at lave et burndown chart, hvor vi estimerede hvor langt vi skulle være efter hver arbejdsdag på skolen, og desuden hvor meget tid vi skulle bruge på hver iteration mm.

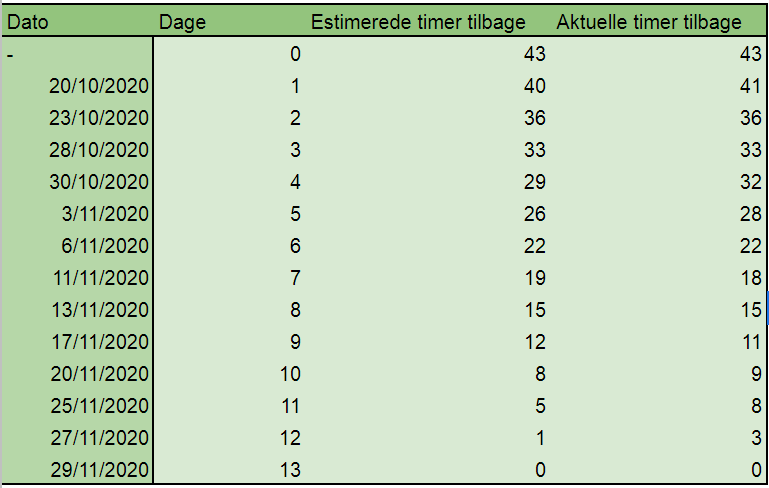
|  |  |
| --- | --- |
|  | **GitHub-Link** |
| **Userstories** | https://github.com/cecilianygaard/Interaktiv-Test/issues?q=is%3Aissue+label%3AUserstory |
| **Tasks** | https://github.com/cecilianygaard/Interaktiv-Test/issues?q=+is%3Aissue+label%3ATask+ |
| **Iterationer** | https://github.com/cecilianygaard/Interaktiv-Test/projects?query= |

Tabel 1: tabel med GitHub links

Undervejs har vi løbende opdateret dette burndown chart og planlægningsskema i forhold til, hvor langt vi var nået sammenlignet med vores forventninger:

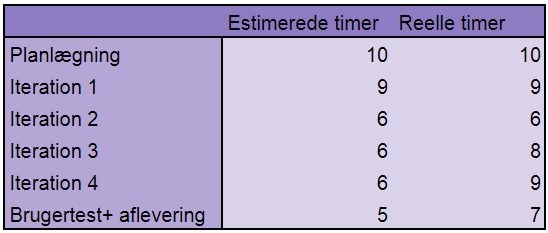
*Tabel 3: Oversigt over tidsplan brugt til Burndownchart.*

Tabel 2: Oversigt over tidsplan brugt til Burndownchart.



*.*

Tabel 3: Oversigt over estimeret tid brugt og aktuel tid brugt



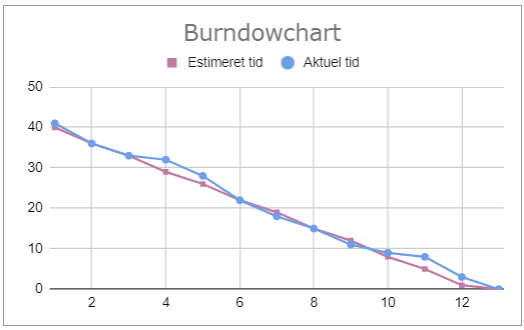


Figure 2: Burndownchart (data i Tabel 3).

# Realisering

*I følgende afsnit beskrives processen gennem realisering af vores program, herunder vores brug af iterationer, møder, samt en beskrivelse af vores endelige produkt.*

**Dokumentation og analyse af iterationer**

Vi har haft oprettet 4 iterationer. Planen har bestået af at starte den første iteration 1 d. 28/10, og derefter åbne en ny iteration hver uge. På denne måde kunne vi fra starten af have et overblik over hvad der skulle laves, og vi har nemt kunnet opdatere og udvikle vore produkt hver gang vi har startet en ny iteration. På denne måde har det ikke været en lineær progression, men en konstant evolution.

Vi fandt tidligt i projektet ud af at det ikke ville virke for vores program at kode databasen i SQLite, fordi denne ikke kan arbejde sammen med JavaScript. Fordi vi konstant har tjekket op på hvordan vores kode skulle udformes kunne vi derfor lave om på hvordan denne skulle skrives, og i stedet skrive den i Indexed DB. Dette gjorde at vi havde en glidende produktudvikling. Dette har også bidraget til en flydende arbejdsprocess, der var under vedvarende udvikling.

**Dokumentation og analyse af test (Brugertest)**

Efter både at have testet vores program hos en lærer og en elev. Har vi fundet ud af, at vi ifølge læreren har overholdt vores krav godt. Vores hjemmeside var ifølge læreren brugervenlig, og det var nemt for læreren, at finde rundt på hjemmesiden, og nemt finde ud af, hvilket knapper der går hen til de forskellige ting. Et af vores andre krav var også, at vores hjemmeside skulle være så minimal som muligt, så der ikke var for mange forstyrrende elementer. Dette fik vi også ifølge læreren opnået, med en god gennemsigtighed og fine farver. Helt generelt var det fra lærerens side også en god måde at lave elektroniske prøver på, samt en god måde at få et overblik over sine elevers faglige kompetencer på. Som slut kommentar fra læreren fik vi afvide at vores hjemmeside er en side, som læreren godt kunne se sig selv bruge i fremtiden til sin egen undervisning.

Ifølge eleven er vores hjemmeside også dejlig brugervenlig og lige til højrebenet. Fra elevens side fik vi dog afvide at vores farver på hjemmesiden, var lidt underlige at kigge på. Dette kan være skyld af at vi ikke har benyttet os af websikre farver, så de kan derfor se underlige ud og variere en smule. Ifølge eleven er vores hjemmeside en god måde for elever selv at kunne holde øje med sin egen udvikling, og blive mere motiveret, når man som elev ser, at ens udvikling får den rigtige vej. Desuden mente eleven også, at vedkommende sagtens kunne forestille sig, at et program som vores kunne blive brugt til fremtidig undervisning.

Efter at have afprøvet vores hjemmeside på en 8. klasses dreng og en udviklingssygeplejerske, kan vi også konkludere at vores hjemmeside egner sig godt til alle aldersgrupper, så dette er en hjemmesiden man både kan benytte sig af i gymnasiet, men også i folkeskolen til de lidt yngre elever.

**Dokumentation af analyse af møder**

Vi har i den første time til alle vores DDU timer holdt et møde, hvor vi har sat vores hoveder sammen og diskuteret hvad vi mangler, hvad vi er i gang med, og hvad vi skal begynde på hver især. På den måde er der aldrig nogen, som der er i tvivl om, hvad de andre foretager sig, samt hvad man selv skal begynde på. Vi har holdt styr på vores møder ved, at en person hver dag skrev ned, hvem der gjorde hvad nederst i vores kontrakt, se bilag 2.

**Runtime beskrivelse af det endelige produkt**

Vores endelige produkt er en hjemmeside, som vi har kaldt “Test-Fest”. Vores hjemmesider har den egenskab, at læreren kan følge med i hver enkelt elevs udvikling hen over et alm skoleforløb. Det vil sige, at i starten af eksempelvis et nyt matematikforløb lægger læreren en test ud til eleverne med det, som læreren gerne vil have, at de skal kunne, eller forventer, at de kan i forvejen. Eleven vil så i starten af forløbet tage denne test og få et procenttal ud til sidst, der viser, hvor godt eleven klarer den første test. Når forløbet så er ved at være slut, vil læreren lægge den samme test ud, som eleverne så igen skal tage. Deres to resultater vil så blive sammenlignet til højre for de to testresultater i procent, så både elev og lærer vil kunne se, hvor godt eleven har klaret sig, og hvilken retning elevens udvikling bevæger sig i.

**Teknisk beskrivelse af endeligt produkt**

Herunder ses et flowchart over, hvordan de forskellige filer i vores kode interagerer med hinanden.

style.css

Elev\_test.htm

login\_page.html

main\_elev.html

main\_lærer.html

newTest.html

opret\_bruger.html

opret\_hold.html

resultat\_elev.html

resultat\_lærer.html

brugere

database\_user

hash

hash.html

*Figur 3: Flowchart over fil interaktion i koden.*

**Konklusion**

Ud fra vores problemformulering, som lød på følgende: “Lav et program, som kan hjælpe lærere med at vurdere deres elevers faglige niveau.”, og den opgivne projektbeskrivelse som lød på følgende “Et system, hvor en lærer kan oprette en klasse med 30 elever, taste opgaver ind i systemet, lave en test af klassen og få en oversigt med elevernes resultater.” Ud fra disse to opgavebeskrivelser, kan vi konkludere, at vi har lavet et program, som hjælper lærerne med at holde overblik over, hvor deres elevers faglige niveau ligger inden for det pågældende emne/fag. Herudover har vi i vores projekt haft flere krav. Vi har blandt andet haft nogle krav om design af vores hjemmesider, herunder at vores hjemmesider skal indeholde bløde former og pastelfarver. Dette krav har vi opfyldt ved at afrunde hjørner, så vi ikke har nogen skarpe kanter, og vi har lavet et farveskema med pastelfarver, som vi har benyttet os af på vores hjemmeside. Et andet krav var, at vores program skulle være sikkert at bruge. Dette har vi opfyldt ved at hashe passwords. Et tredje krav var, at vores program skulle være brugervenligt, hvilket vi fandt ud af i vores brugertest, at det var det. Vores sidste og fjerde krav var, at vores program skulle være tilgængeligt. Dette har vi sikret ved at lave en hjemmeside. Vi har dog nogle mangler, når det kommer til tilgængelighed af vores database, herunder vores valgt af en klient baseret database, hvilket vi vil komme mere ind på under afsnittet ”perspektivering”. Alt i alt kan vi konkludere, at vores hjemmeside har besvaret opgaveformuleringen og diverse krav, som hørte med, men at vi kunne gøre vores hjemmeside mere tilgængelig, hvis vi havde mere tid, hvilket forklares under afsnittet ”perspektivering”.

# Perspektivering

I dette projekt har vi arbejdet med databasen Indexed DB, hvilket er en klientbaseret database.[[5]](#footnote-5) Dette har medført, at alt det, som en bruger opretter af hold, kun vil eksistere på den browser, som brugeren har oprettet holdet på, da en selvstændig database vil blive oprettet alt efter hvilken browser, man bruger. Derfor ville det være smart at bruge en anden database, som er serverbaseret, i stedet for, så alle browsere kører med den samme database, så man kan logge ind på forskellige computere og se de oprettede hold, i stedet for at have en database, som resulterer i, at læreren opretter et hold og nogle prøver på samme browser, og at eleven bliver nødt til at løse opgaverne på samme computer, som holdene er oprettet på. Ved at bruge en anden database vil vi på den måde også sikre programmets tilgængelighed en del bedre, og tilgængelighed var jo et af vores krav til vores program.

I vores problemformulering fandt vi ud af, at det ville være smart at have færdiglavede tests, som er kategoriseret efter emner indenfor funktionel grammatik i engelskfaget. Dette nåede vi aldrig at lave, men hvis vi havde længere tid, vil dette være en af de ting, vi ville arbejde på at inkludere i vores program. Herudover fandt vi i problemformuleringen ud af, at åbne svar var mere optimale end multiple choice svarmuligheder, og vi har i vores program kun nået at lave multiple choice svarmuligheder. Hvis vi havde længere tid, ville vi derfor have udvidet test funktionerne til, at man også kunne lave åbne svar.

I vores program har vi lagt op til, at brugeren også skal angive sin email, når nye brugere oprettes. Vi har dog ikke endnu lavet nogle funktioner endnu, hvor vi bruger mailen til noget. Hvis vi havde længere tid, kunne man for eksempel lave en funktion, hvor programmet via. email sender brugeren testresultater, besked når læreren har oprettet nye tests til eleven og lignende.

En af grundidéerne ved vores program er, at en lærer kan sende et hold elever en test to gange, hvorefter eleven og læreren kan se testresultaterne og et procenttal, der viser hvor meget bedre eleven var anden gang i forhold til første gang, eleven tog testen. Vi har dog kun nået at gøre, så en lærer kan sende testen til en bestemt elev eller et bestemt hold af elever, og eleverne kan tage testen lige så mange gange, eleven vil. Hvis vi havde længere tid, ville vi arbejde på at skrive koden, så en lærer kan sende en test til en enkelt elev eller et bestemt hold to gange, så hver elev kun kan tage testen to gange i alt.

Lige nu er det desuden kun muligt at tilføje enkelte elever, uden eleven er tilføjet et hold, hvilket betyder, at enkelte elever kan tilføjes til programmet uden at være tilknyttet et hold. I fremtiden ville det være smart, hvis man kunne vælge, om en elev skulle være tilknyttet et hold eller ej, når man tilføjer enkelte elever (ikke ud fra en Excel-fil). Koden er endda skrevet til, at man kan tilføje en elev til et hold, men koden er blevet udkommenteret. Koden står under filen “Brugere” og “opret\_bruger.html”.

Til sidst ville et være fedt, hvis man kunne omdanne vores program til en app, som man kan bruge på en telefon. Dette ville gøre programmet mere tilgængeligt, så selv elever, der har glemt deres computer, også kan tage testene. Da vores program er skrevet i Javascript og i HTML, vil det være nememre at omdanne vores program til en app, end hvis vores program for eksempel var skrevet i Java i Processing.

# Litteraturliste

[Farver, Former og Symbolik! – Jimmy Madsen (wordpress.com)](https://ironbrowblog.wordpress.com/portfolio/farver-former-og-symbolik/)

[Farvesymbolik | Lær om farver og farvernes betydning](https://www.farvesymbolik.dk/)

<https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API>

[Internetbrug og enheder: Kulturstyrelsen (slks.dk)](https://mediernesudvikling.slks.dk/2015/internetbrug-og-enheder/)

# Bilag 1

**Interview:**

*Føler du, at du har styr på dine elevers faglige niveau?*

**Morten**: Både og; nogenlunde.

**Anja**: Ja.

**Lene**: Ja, men hvordan får man dem i bunden højere op? Hvorfor ligger de så lavt?

**Jan**: Nej , men føler heller ikke, at jeg skal det - skal ikke give karakter i grundforløbet.

*Tror du, at du ville kunne få et bedre overblik over dine elevers faglige niveau ved at bruge et program til prøver, og hvor du kan se elevernes faglige udvikling?*

**Morten**: Indenfor visse dele af stoffet. Fagbegreber (man lærer ikke, om eleverne kan bruge begreberne); sværere med humanistiske fag.

**Anja**: Ja.

**Lene**: Ville ønske, der var sådan et program - spændende.

*Vil du have det bedst med selv at kunne generere spørgsmålene, eller at det bliver genereret via. AI?*

**Morten**: Vil selv generere.

**Anja**: Vil selv generere.

**Lene**: Smart hvis det kunne begge dele.

*Hvorfor?*

**Morten**: Tilpas spørgsmål; for at få kontekst og mere individuelt til selve klassen.

**Anja**: Formuleret på en anden måde, andre begreber bliver brugt, ikke lært alt i prøven (prøver passer ikke til klassen).

**Lene**: For at få begge muligheder.

*Multiple choice vs. helt/åbent svar?*

**Morten**: Åbent svar.

**Anja**: Åbent svar. Ellers kan eleverne ”tilfældigvis” svare rigtigt.

**Lene**: Begge dele.

*Andet?*

**Morten**: Upload materialer til prøverne.

**Anja**: Kan man indkredse, hvorfor eleverne svarer forkert? AI? Matematik, fysik; hvor går det galt?

Eleverne kan skrive inde i programmet, hvad de føler, der gik galt.

**Lene**: Funktionel grammatik er et problem, kan man gøre det nemmere via. program?

# Bilag 2

Herunder ses vores kontrakt, hvori der blandt andet er link til GitHub, herunder tasks, userstory, burndown chart, iterationer og koden. I koden ligger skitsen til databasen i Access under mappen “data”.

|  |  |
| --- | --- |
|  | |
| Gruppemedlemmer | *Cecilia Nygaard, Elisabeth Astrup Christensen, Jessie Lorentzen & Nicoline Mejlgren von Bülow* |
| ***FASE 0: Ide-udvikling*** | |
| Problemidentifikation | *Hvordan kan læren holde styr på elevernes faglige niveau?* |
| Problemanalyse | *En lærer vil kunne holde styr på elevernes faglige niveau i et program på computer/telefon eller en hjemmeside, hvor læreren kan lave små og/eller større prøver, som kan bruges til at teste elevernes faglige niveau.* |
| Produktprincip | ***Løsningsforslag****:*   * *En app til telefon, hvor en lærer kan oprette prøver til elever og herefter se, hvor godt eleven har klaret sig.* * *Hjemmeside, hvor en lærer kan oprette prøver til elever og herefter se, hvor godt eleven har klaret sig.* * *App til telefon og/eller computer med flashcards, hvor eleverne kan teste sig selv og vurdere hvor svære, de mener spørgsmålene er, og hvor læreren kan se deres vurdering.*   ***Krav og vægtning af krav til kravmatrix****:*  *Vægt fra 1-3.*   * *Brugervenlighed [vægt: 2]*   + *KISS*   + *Konventioner*   + *KIT* * *Tilgængeligt (alle elever skal kunne tage testen (fx ikke på iPhone, for alle har ikke iPhone)) [vægt: 3]* * *Sikkerhed [vægt: 2]*   ***Kravmatrix:***  *Point fordeles fra 1-5.*   |  |  |  |  | | --- | --- | --- | --- | |  | ***App*** | ***Hjemmeside*** | ***Flashcards*** | | ***Brugervenlighed  Vægt: 2*** | *4* | *5* | *2* | | ***Tilgængelighed Vægt: 3*** | *3* | *5* | *3* | | ***Sikkerhed  Vægt: 2*** | *5* | *5* | *5* |   ***Point i alt****:*  *App: 27*  *Hjemmeside: 35*  *Flashcards: 23*  *Som vi kan se på vores kravmatrix, har løsningsforslaget “hjemmeside” fået flest point. Derfor går vi videre med dette løsningsforslag.* |
| ***FASE 1: Planlægning*** | |
| Produktbeskrivelse & krav | ***User story:***  En hjemmeside, hvor lærere kan lægge tests op til et hold elever, og eleverne skal kunne tage testen to gange i alt. Efter eleverne har taget testen en gang, skal både lærere og elever kunne se elevernes resultater fra testen. Når eleven så har taget testen anden gang, skal lærer og elev også igen kunne se resultaterne for begge tests, og de skal kunne se udviklingen af elevens faglige kompetencer fra første gang de tog testen til anden gang, de tog testen. Udviklingen skal stå som et procenttal, som fortæller hvor mange procent bedre, eleven var anden gang, han/hun tog testen, i forhold til da han/hun tog testen første gang.  ***Kravspecifikationer****:*   * *Brugervenlig.*   Vi vil gerne have, at vores program er så brugervenligt som muligt, så både lærer og elever i alle aldersgrupper har muligheden for at benytte sig af programmet. Dette kan blandt andet opnås ved brug af webkonvention og optimalt design (se designbeskrivelse & krav).  Dette krav måles via. brugertest.   * *Tilgængelig.*   Det er vigtigt, at hjemmesiden er nemt tilgængeligt, så elever kan komme på programmet gennem deres telefon, hvis de ikke har eller har glemt deres computer.  Dette krav efterprøves ved at se, om vores hjemmeside kan køre på forskellige devices, fx Android-telefon, iPhone, PC og MacBook.   * *Sikkerhed.*   Når man arbjeder med folks persondata, er det også vigtigt, at vi har en form for sikkerhed på vores hjemmeside, så folk ikke risikerer at få delt eller stjålet kodeord, testresultater eller lignende. Vi hasher passwordet (vi krypterer ikke, for vi har ikke behov for at kende det egentlige password igen senere; vi har kun behov for, at computeren kan finde ud af, om det tastede password passer til det krypterede password eller ej).  Vi sørger for at implementere sikkerhed i koden. Vi tjekker, at man rent faktisk kan logge ind som bruger med det hashede password. |
| Designbeskrivelse & krav  *Designet skal gennemtænkes og være underbygget af teori.* | ***Designbeskrivelse****:*  Det primære design fokus er, at vores hjemmeside skal udsende beroligende og behagelige signaler. Dette vil vi opnå ved at have bløde former frem for at have hårde kanter på diverse knapper. Vi vil derfor benytte os mere af cirkler frem for firkanter, da cirkler symboliserer fokus, hamoni og dynamik.  Udover dette vil vi også have, at farverne er rolige. Rolige farver kan opnås ved at have lysere nuancer, eksempelvis pastelfarver eller udtyndet versioner af farver. Dette vælger vi from for hårde og skarpe farver, for at berolige eleven inden den kommende test går i gang.  ***Krav****:*   * *Beroligende farver* * *Bløde former* * *KISS* * *Brug af webkonventioner* * *Balance ved hjælp af symmetri* * *Kvadrater med afrundede hjørner & cirkler* |
| Prototyper & brugertest | ***Prototype link****:* [*https://drive.google.com/drive/u/0/folders/1f5V3Qlx5HV4ow3LbGJj9pbvDTQZUFhJM*](https://drive.google.com/drive/u/0/folders/1f5V3Qlx5HV4ow3LbGJj9pbvDTQZUFhJM)  ***Brugertest link****:*  [*https://youtu.be/GxkfulqY4fjMw*](https://youtu.be/GxkfulqY4fjMw) |
| User stories & tasks | ***Userstory:***  <https://github.com/cecilianygaard/Interaktiv-Test/labels/Userstory>  ***Tasks:***  <https://github.com/cecilianygaard/Interaktiv-Test/labels/Task> |
| Iterationer  *Der afholdes minimum 2 statusmøder. Vejleder bestemmer tidspunkt/dato.* | ***Iteration 1: Fredag, d. 6/11-2020***   * [*https://github.com/cecilianygaard/Interaktiv-Test/projects/1*](https://github.com/cecilianygaard/Interaktiv-Test/projects/1) |
| ***Iteration 2: Fredag, d. 13/11-2020***   * [*https://github.com/cecilianygaard/Interaktiv-Test/projects/2*](https://github.com/cecilianygaard/Interaktiv-Test/projects/2) |
| ***Iteration 3: Fredag, d. 20/11-2020***   * [*https://github.com/cecilianygaard/Interaktiv-Test/projects/3*](https://github.com/cecilianygaard/Interaktiv-Test/projects/3) |
| ***Iteration 4: Fredag, d. 27/11-2020***   * [*https://github.com/cecilianygaard/Interaktiv-Test/projects/4*](https://github.com/cecilianygaard/Interaktiv-Test/projects/4) |
| ***FASE 2: Udvikling*** | |
| Kode repository | <https://github.com/cecilianygaard/Interaktiv-Test> |
| Burndown chart | <https://docs.google.com/spreadsheets/d/1aoFKFnExKKFxEHmaRt9EJqDpgefYuxIsdb-hEe2nacE/edit?usp=sharing> |
| Test & møde log.  *Planlæg meningsfulde brugertest, systemtest og møder* | **Møder:**  Vi holder måder hver gang, vi afslutter en iteration. Altså holder vi møder på følgende tidspunkter:  **Opdatering d. 11/11-2020**: Vi holder statusmøde hver dag, vi har DDU, når vi møder op.   * Fredag, d. 6/11-2020   Vi er færdige med iteration 1 med undtagelse af tasken “lav ‘tekstfelt’ class, som vi næsten er færdige med. Pico færdiggører tasken, og vi andre går videre til iteration 2. Pico joiner iteration 2, når tasken er færdig.   * Onsdag, d. 11/11-2020   Vi starter med at merge.  Pico: Systemfunktioner.  Jessie: Loginside interface.  Cecilia: ElevTestSide interface  Elisabeth: NewTest interface.   * Fredag, d. 13/11-2020   Vi starter med at merge.  Pico: Database (opret database og implementer database i kode).  Jessie: Interface.  Cecilia: OpretHold interface.  Elisabeth: NewTest interface & skriv på rapport.   * Tirsdag, d. 17/11-2020   Vi starter med at merge.  Pico: Merge error.  Jessie: Interface resultatside.  Cecilia: Rapport.  Elisabeth: Rapport.   * Fredag, d. 20/11-2020   Vi starter med at merge.  Pico: Oprettelse af test og gem test.  Jessie: Rapport.  Cecilia: Rapport.  Elisabeth: Rapport, hashing, ResultatPage og relationer mellem sider.   * Onsdag, d. 25/11-2020   Vi starter med at merge.  Pico: Færdiggør opret test og gør det muligt at tage test, samt gør det muligt at oprette bruger.  Jessie: “Opret hold” og “se hold” siderne interface, samt rapport.  Cecilia: “Opret hold” og “se hold” siderne.  Elisabeth: Rapport, samt “opret hold” og “se hold” siderne, Excel-ark.   * Fredag, d. 27/11-2020   Vi færdiggører det sidste kode og skriver rapport i dag.  **Brugertest:**   * Vi laver brugertest hjemmefra i weekenden, inden vi afleverer. |
| ***Aflevering*** | |
| Aflever Rapport og Produkt | Se følgende link (omfang halv størrelse):  <https://digitaltdesignlyngby.github.io/mini%20eksamensprojekt/projektstyring_brugertest_rapport/Evaluerings%20Eksempel.html> |

**PS**:

Elev:

Password til dude er “ho”.

Lærer:

Password til p er “”.

1. [Internetbrug og enheder: Kulturstyrelsen (slks.dk)](https://mediernesudvikling.slks.dk/2015/internetbrug-og-enheder/) [↑](#footnote-ref-1)
2. [Internetbrug og enheder: Kulturstyrelsen (slks.dk)](https://mediernesudvikling.slks.dk/2015/internetbrug-og-enheder/) [↑](#footnote-ref-2)
3. [Farver, Former og Symbolik! – Jimmy Madsen (wordpress.com)](https://ironbrowblog.wordpress.com/portfolio/farver-former-og-symbolik/) [↑](#footnote-ref-3)
4. [Farvesymbolik | Lær om farver og farvernes betydning](https://www.farvesymbolik.dk/) [↑](#footnote-ref-4)
5. <https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API> [↑](#footnote-ref-5)