

1 Referat d. 15-09-21

Dato: 15-09-2021

Tidspunkt: Kl. 10.15

Sted: Edison

Fremmødte: Asger, Johanne, Christina, Maggie, Andreas, Poul Ejnar (PE)

Referent: Asger og Andreas

1. Gruppen snakker med PE om Azars manglende fremmøde.
PE vil sende en mail til Azar, for at høre om han vil engagere sig i dette semester, og om vi kan regne med ham i projekt.
2. Alle introducerer sig for vejleder (PE).
3. Alle glæder sig til praktik da det giver praktisk erfaring og mere koncentreret arbejde.
4. PE har input til vores arkitektur. Web- og WPF GUI er okay.
PE siger vores idé er rigtig god til 4. Sem projekt generelt, det kommer ind over alle læringsmålene.
5. Lav test og design tidligt, for at skabe diverse rammer for systemet.
6. PE foreslår vi skal koncentrere os om krav spec.
7. User Stories giver mulighed for at lave en specifik beskrivelse, som senere kan bruges som accepttest.
8. Fast mødetid med PE, hver onsdag kl 10:15 efter GUI. Eventuelle aflysninger af møde skal bare meldes dagen inden, samt dagsorden vil PE gerne have dagen forinden.

Referat d. 22-09

Dato: 22. september 2021

Tidspunkt: 10:00

Sted: Edison

Fremmødte: Asger, Christina, Johanne, Poul og Maggi

Referat:

1. **Feedback på kravsspec:** Sproglig: Brugeren skal have mulighed for at være i stand til - lidt for krydret
2. User stories: ligger op til en stram struktur - vi følger det ikke
1 user story pr. formål - vi skal dele det lidt mere op
fx. brugeren kan oprette sig og tilgå platform - bruger skal kunne oprette en begivenhed. Se det som UC, hvor 1 bobbel er et formål. Der skal gerne et formål på, men kig på den faste struktur.
3. Moscow er en prioritering af kravene, der ALLEREDE er defineret i krav. De funktionelle skal i user stories. De andre skal med i noget skrift. Dette er gældende for Must, Could og Should.
4. Ikke funktionelle krav: Beregning - mean time er en tid og ikke en procent. Den skal kaldes "praktisk opetid" istedet for. Man behøver ikke beregning, men derimod bare beskrive det. 93 er for lavt tal, men gerne 99% i stedet for. MTBF er mere med HW at gøre. Et enkelt system er ca. 99%, hvorimod hvis man har flere enheder/komponenter med, skal man beregne de 99% for hver komponenterne ud.
5. Uendeligt antal pub crawl er et meget stort antal. I stedet for lad vær med at skrive om det, da vi så låser os fast. Relevant kunne være antal samtidige brugere - det at skulle håndtere 100 brugere burde ikke være et stort problem, men derimod 1000 eller 100.000 brugere.
6. **Review:** krav om at det skal afholdes, men det stopper os ikke fra at gå videre, at det endnu ikke er kommet igennem det
7. **Fremmøde:** PE har sendt en e-mail til Azar, men der er ikke kommet svar. Hvis der ingen kommunikation fra ham i næste uge, og ikke deltager aktivt hvis han møder op, bliver han udmeldt fra projektet, da vi allerede er kommet langt med kravene.
8. Der snakkes lidt om aktive studerende på linjen.
9. **Næste mål:** opdatere krav spec og derefter begynde på SW arkitektur
10. Handler om at få lavet et okay projekt og lave en kanon rapport. Det er nok en god ide, at der er noget, der virker, men bare kravene for systemet er opfyldt.

2 Referat d. 29-09-21

Dato: 29-09-21

Tidspunkt: Kl. 10.00

Sted: Edison

Fremmødte: Andreas, Asger, Christina, Maggi, Poul Ejnar og Johanne

Referat: Johanne

1. Gruppen havde glemt at fremsendte det udarbejdet materiale og kan derfor ikke få en dybdegående feedback, men vejleder kigger kort på user stories.
Vi får at vide at der er en god og stram struktur i user stories og vi kan derfor godt arbejde videre med dem.
Feedback til ikke-funktionelle krav: Der får vi at vide at dette også ser fint ud.
Feedback til domænemodel: Her får vi at vide at den også ser ok ud, og vi har husket at inkludere alle de forskellige systemer. Vi får at vide at det er et simpelt system, men passende til et 4. semester projekt
2. Vi skal også bruge C4-modellen, samt 4+1 modellen.
4+1 kigger på de forskellige views. Disse skal tage udgangspunkt i vores user stories.
C4 er det logiske view hvor man langsomt zoomer ned i systemer
3. Næste skridt er at gå igang med C4 og 4+1 diagrammerne. Vi skal dog ikke lave level 4 i C4-modellen (det er selvsigende hvorfor, fordi man skal kunne skrive sin kode direkte fra level 3).
4. Evt: Vi har haft skrevet til Azar ang. studie aktivitet, men vi har desværre intet hørt tilbage fra ham. Han betrages derfor ikke som værende en del af projekt gruppen længere - det er vi som gruppe desværre ærgerlig over.

Referat d. 13-10-21

Dato: 13. oktober 2021

Tidspunkt: 10:00

Sted: Edison

Fremmødte: Andreas, Christina, Johanne, Poul og Maggi

Referat:

1. Feedback:

Deltager = fjern Azar.

Alle sekundærer aktører fjernes fra kontekst diagram.

2. C4 diagram:

Apache web server til web server (mere generelt)

Teknologi analyse ville være fint (generelle diagrammer - analyse - specificere)

Component kommer når der kommer kode på - det giver mening at vente - vi er for abstrakte

3. 4+1-diagrammet:

7.2.3 er et deployment view diagram

Physical view kan slås sammen

7.2.4 RPi måske for specifikt

4. ProcessView:

Dette behøves ikke, kun hvis der er noget interessant indenfor dette, som vi vil have med.

5. Sekvensdiagrammerne:

SQ 1,, der skal ikke vises interne koblinger - hjemmesider viser siden til brugeren og ikke sig selv (fig 13 har ikke noget navn)

Kun sekvensdiagrammer for relevante User Stories, så måske et par til hver aktør

6. WebApi kunne også være en ide at sætte over databasen, som vi har lært lidt om i DAB og NGK.

ASP Net.Core kan ikke køre på RPI W Zero. Istedet for på RPI 3 og 4, og ellers skal det køre i skyen

7. Styr på til næste gang:

Rettelser, teknologianalyse, sætte projekt op/skrive lidt kode og design. Derefter 4+1 diagram.

Referat d. 10-11

Dato: 10. november 2021

Tidspunkt: 10:00

Sted: Edison

Fremmødte: Andreas, Asger, Christina, Poul og Maggi

Referat:

1. Vi har lidt Web spørgsmål: Burde vi lave vores eget login system, eller skal vi hellere udnytte det som ASP.NET kan lave for os? PE syntes vi skal holde os til vores eget, nu når vi er startet.
Kalender: PE har ikke så meget forslag til kalendere i ASP.NET. Hvis man bruger U-frontend er der nogle forskellige kalendere man kan lave og ellers google.
Min profil: PE foreslår at vi skal se hvordan ASP.NET login systemet er lavet, og drage inspiration deraf.
Test: Controllere kan testes med NUnit. Test af knapper er noget, man kan få autogeneret, men dette er ikke noget, vi lærer om i undervisningen.
2. Database: Spørger til feedback om vores nuværende database design. Databasen vil ligge lokalt, men man kan bruge 2 forskellige fysiske databaser. Man kan derimod lave en online server, så man tager web-projektet og publishere det op til ASH? PE syntes der manglede lidt (jeg nåede ikke at høre hvad han sagde...) DBZ skal afgive hvilke tabeller der er, så når disse ikke er medtaget, kommer det ikke frem.
3. Get og post til WPF. Lektion 19 GUI samt lektion 10 i NGK slides (20 ASP/WebAPI.pdf). WPF kan tilgå web database?
4. Database: PE foreslår vi bare laver code first, hvorefter vi deployer, så den automatisk sættes til skyen. Der skal laves en API controller i Web applikationen som snakker med EFcore, som så automatisk genererer posts og gets i controlleren. Vi skal tilrette gets og posts, hvorefter vi nede i WPF skal have en HTTP-client, som kan kommunikere med de autogenerede gets og posts.
5. WPF: Resultat på, at der ikke åbnes så mange modalvinduer oven i hinanden. "Show Dialog" returnerer noget med typen dialog. Man kan spørge, om den er lig med true. Man skal dog få den til at returnere et id eller objekt, som der er et eksempel i MVVM-slides i GUI. Man skal sætte nogle værdier og få dem returneret tilbage, så man fra hovedvinduet åbner de næste vinduer.
Som default lukker programmet, når mainwindow er lukket. Man kan derimod skifte content, hvor man kan have layout i dit vindue, som man så kan skifte content i. Man kan derved lave vinduet om til usercontrol.
I applikationsklassen kan man sætte shut-down til at være det sidste åbne vindue fremfor mainwindow. Man skal derefter selv til at lukke vinduerne. Man kan sætte de bagvedliggende vinduer invisible, så der kun er et vindue "åbent"/visible af gangen.
6. Error-vinduet skal vi først tænke på, når databasen er kommet op at køre, så vi har noget at få den op på.

Referat d. 03-11

Dato: 3. november 2021

Tidspunkt:10:00

Sted: Edison

Fremmødte: Andreas, Asger, Christina, Johanne, Poul Ejnar og Maggi

Referat:

1. Linket til login mellem sider. Hard-code. "A, ref".
2. Tilgås og oprettes data: "Inde i Opret bruger" - add/linket CVR nr, - 2 måder: 1. Lav en application User, hvor man bygger videre på identity user, og så bygger man de felter på. 2. Man kan godt bruge den database, der allerede er lavet. En måde: Få en ny identity id, som bruges som primær-nøgle. Man skal så selv lave noget kode inde i register. Inputtet kommer altså ned i controlleren, og så sendes derfra ned til databasen. Modelklassen er for virksomheden. Inde i identity-user er der en anden for login. De skal altså "smeltes" sammen. Denne klasse skal altså ligge i models,

som er en hel alm. model, som skal have properties fra virksomheden og fra identity-user. Controlleren skal også være derinde.

3. Scarffolder: Keep it simple!
Inde og rette i register - lave det i code-behind.
4. Virksomhedsprofil - komme med en liste hvilke pubcrawl er booket: Der bliver sendt cookies frem og tilbage fra brugeren, når denne logger in. Når man kigger i controlleren, er der noget, der hedder user.identity, hvor man så kan få brugerens id. Dette kan bruges til at finde brugeren i databasen, og dertil kan man så finde informationer herfra. Få identiteten og finde denne i databasen, hvor man så kan få info. Man få virksomheds(model)objektet tilbage, som man så kan sende over til viewet ved "model".
5. WPF - fravælge datoer, der er blackout. Dette er ikke noget PE ved.
Dependency injection i controlleren, hvor man har "ILogger", hvor man vil sige, at man vil bruge databasen. "Application dp context. og så kalde på user-tabellen.
Simpelt, da der er forskellige applikationer for bruger og virksomheder. Brug lektionen i NGK, der kommer på tirsdag d. 23. november. Web-api og cookies passer ikke med WPF. DB.context skal bruges. Vi kan også tilgå den database, der allerede er lavet til brugeren, men vent til tirsdag. User-tabellen skal udvides med "et eller andet"-password.
6. C4-model - behov for klasse? Man kan lave klassediagrammer for dele af systemet, men det overordnede er C4.
7. Test - lave interface til klassen og så lave unit-test.
8. Til næste gang: Mens web-gruppen arbejder på at snakke med databasen, kan WPF-gruppen arbejde med web-api-controlleren.
Overvejelser, om vi vil have Jenkins-server op at køre, om det så er for udseendets skyld eller om det er relevant for os.
9. Først for databasen., derefter for web og til sidst til wpf med Design, implementering og unit-test. Integrationstest kan også være derunder. Derefter et afsnit med test, hvor de resterende tests kommer ind.
10. I "læseferien" vil PE gerne læse projektet igennem

Referat d. 01/12

Dato: 1. December 2021

Tidspunkt: 10:00

Sted: Edison

Fremmødte: Asger, Christina, Johanne, Maggi og Poul Ejnar

Referat:

1. Vi skal have bestemt tidspunkt for accepttest med PE. D. 15 formiddag kl 10, kan PE.
2. C4 og 4+1 model: Hvor relevant er den at få med? C4 er den vigtigste at få med. Så skal vi bare forklare hvorfor vi ikke har den anden med, hvis vi vælger ikke at tage den med. C4 er smart at få med i hovedrapporten.
Skriv om MVVM og MVC i logical view
Vi skal ikke skrive om Process view
Vi har brugt 4+1 som refererer til en Phillip...
Physical view er vores deployment, som allerede er lavet. Process view behøver vi ikke.
Physical view: skriv om webserver, WFP applikation. Beskriv flow i systemet. Det fysiske view kan ses i figur 22 i billagsrapporten. Physical view skal omdøbes til deployment view.
Vi skal fokusere på vores logic view og physical view.
Forklar hvorfor vi ikke har development view og process view.

3. Har vi kigget i rapport skabelonen? Nej. Ingen yderligere spørgsmål heromkring, da Johanne fik svar, ud fra at vi kigger i skabelonen.
4. Accepttest: Vi burde have lavet accepttesten inden vi startede med at kode. Accepttest skal have alle detaljer. Skal bare følges slavisk.
Hvad vi har ser godt ud indtil videre.
5. Hjælp til web-app og database: Vi har fået database og view til at fungere så vi kan deploy.
Vi har lidt problemer med i databasen med foreign key til citycrawl.
PE siger vi har vendt det forkert. Foreign keys skal ligge i citycrawl og ikke sådan som vi har gjort nu.
PE siger pubcrawl skal være en selvstændig entitet... Det bliver et problem for os, da vi egentlig er færdige med WPF, og vi har lavet pubcrawls som en liste på bruger/user.
Det er pubcrawl der binder tingene sammen, derfor burde den sættes ud for sig selv.
Det er store ændringer på et sent tidspunkt... Vi skal passe på ikke at smide det væk som vi har, men han foreslår at vi forsøger, hvis vi vil have det til at fungere. Så må vi dog nedprioritere test af Web og database og forklare i rapport.
Hvad ville der så skulle ændres på WPF? Den enkelte bruger har stadig kun en liste af pubcrawls, så der burde ikke skulle ændres noget.
Lav mange til mange mellem pubcrawl og bruger samt pubcrawl og virksomhed.
Hvis vi kan få løst det og få lavet 1 x-unit, så ville det være fint.
Pubcrawl skal indeholde dato og sted. De andre skal have foreign key til pubcrawl.
I API skal vi specificere hvilke data vi skal bruge og dermed ved vi efterfølgende hvad der skal ændres i WPF
Ned prioritere test, få forklaret i rapport omkring hvorfor vi har nedprioriteret.
Skriv i rapporten hvad der nedprioriteres, at vi har lavet databasen om ret sent i projektet.
6. Vi holder os fra integrationstest udover de billeder vi har i rapporten.
7. Coverage ser fin ud og fint med Jenkins så vi kan vise continuous integration.

Referat d. 08-12-21

Dato: 08-12-21

Tidspunkt: 10:00

Sted: Edison

Fremmødte: Andreas, Christina, Johanne, Maggi, Asger og Poul Ejnar

Referat: Johanne

1. Hjælp til web-login
Der er opstået et problem i at vores web-app ikke længere husker at bruger skal forblive logget ind. Dette har web-gruppen ikke kunne løse. Selve brugeren er logget ind, men det kan ikke ses i web-applikation.
Poul Ejnar kigger på problemet sammen med Andreas.
Han tror fejlen opstår fordi vi bruger jwt tokens når databasen skal snakke sammen med WPF-applikationen. Han er dog på stående fod i tvivl om hvordan det skal løses, så vi kan blive ved med at bruge jwt.
Dog kan det løses ved at anvende cookies istedet for jwt, men det kan være det bliver en stor ændring i web api'et.
Løsning er at fjerne jwt og køre uden sikkerhed. Herefter skrive i rapporten at vi kunne få det til at virke med to forskellige teknologier, men ikke få teknologierne til at virke sammen.
Test til web og database. Vi laver et par test af controlleren for webben og vise at når en pubcrawl bookes kan data ses på web, ergo at databasen virker som hensigten.
fortælle i fremtidigt arbejde at der kunne være flere måde at løse implementeringen af web-api. At api kunne ligge i et separat projekt og ikke implementeret sammen med web-appen. Det ville være den mest optimale løsningen fordi er en god måde at snakke med flere forskellige klienter. En anden løsningen kunne være at bruge cookies i wpf (ikke optimal)
Vi skal hardcode virksomheder i cc-web.

2. Hjælp til C4-model
Poul Ejnar kigger den udarbejdet C4-model igennem for at se om vi mangler noget.
Figur nr 5 er ikke korrekt. WPF snakker ikke direkte med EF core. Pil skal flyttes fra database til api'et/CC-web.
Teksten her til skal også rettes.
Figur 6. CC-web skal vises i bunden i stedet for CC-database.
Ellers siger han god for modellernes indhold.
3. Kigger på logical view i 4+1
figur 21 indsæt reference i figur teksterne. Dette er generelt for alle referencer til figur. Derudover ønskes der en reference liste til slut i rapporten.
Figur 22. CC-app skal pege på CC-web og ikke CC-database. Samt fjern RPI, da den ikke anvendes længere. RPI skal også fjernes fra det fysiske view.
Figur 23. Udskift RPI med webserver. WPF peger på webserver og ikke CC-database. Husk at være konsistent i navne.
I domain view skal der stå CC-web og ovenover browser.
4. CC-app skal vise et billede af Jenkins server i hovedrapporten.
5. Spørgsmål til accepttest
Vi skal beslutte om vi køre med eller uden sikkerhed. Pt er valget at vi tester uden sikkerhed og derefter forventer at sikkerheds testene fejler. Sig at vi har implementeret og vist at sikkerheden virker. Men det ikke kan snakke sammen.
6. præsentere projektets delsystemer i samme række følgende som C4 modellen.
CC-app, CC-web, web-api, CC-database
7. Evt.
Ingen kommentar til evt.