

## Eksamensprojekt for 2. semester datamatiker E2024

I skal lave et **projektkalkulationsværktøj** for jeres kravstiller (kunde) Alpha Solutions. Kravene til systemet er blevet præsenteret til kickoff onsdag d.20.11 og beskrevet i slides fra Alpha Solutions projektpræsentation.

Derudover er der en række krav til projektet og rapporten, som beskrives nedenfor.

### Generelle krav til rapporten

Der skal være sporbarhed ml. de forskellige systemudviklingsaktiviteter i jeres projekt (dvs. virksomhedsanalyse, kravspecificering, design, kodning, test og deployment). Det betyder, at der i rapporten skal være sporbarhed mellem jeres udarbejdede i virksomhedsfaget til user stories og UML-diagrammer og videre fra UML-diagrammerne til jeres kildekode, database og test cases.

Diagrammer (f.eks. UML og ER diagrammer) og udsnit af kode, som I vælger at fremhæve i rapporten skal ledsages af en supplerende tekst, hvor I beskriver jeres valg og reflekterer over fordele og ulemper ved de valg, som I har truffet. Det gælder generelt i rapporten at I skal begrunde de valg, som I træffer undervejs i projektet.

### Virksomheden

I skal udarbejde:

- En uddybende Feasibility-study (analyse) af projektet inden I går i gang. Den skal danne baggrund for at projektet er blevet igangsat, og skal afspejle dette. Bæredygtighed skal indgå i analysen, som en del af argumentationen for at projektet er igangsat. I skal begrunde, ud for de forskellige emner der indgår i jeres analyse – hvorfor projektet er igangsat.

### Systemudvikling

Systemudviklingen baseres på agil udvikling med brug af Scrum og hver projektgruppe skal udpege en Scrum Master (rollen skal gå på tur i de 3 Sprints). Er I mere end 3 i gruppen, står en over.

I skal arbejde iterativt i Sprints og der er afsat nogle faste (obligatoriske) Sprint Planning og Sprint Review møder med Product Owner i projektperioden.

Før hver Sprint opdateres Sprint Backloggen, så det er tydeligt, hvilke user stories og andre artefakter teamet planlægger at arbejde med i den kommende Sprint.

Hver Sprint Backlog skal indgå i rapporten sammen med Product Backlog.

I skal anvende GitHub Projects til at dokumentere jeres Scrum proces.

## Teknologivalg

I skal beskrive hvilke teknologier der er brugt (JDBC, MySql, SpringBoot ...). Her er det ikke meningen at I skal beskrive disse teknologier, men I skal nævne hvilke der er brugt, sådan at de, der skal overtage projektet ved hvilken software, der skal bruges. Der skal versionsnumre på (f.eks. IntelliJ 2024.2.4, ikke blot "IntelliJ"). I kan evt. henvise med link i rapporten til \*.md filer i jeres GitHub repository (se senere afsnit om README.md og CONTRIBUTE.md).

## Krav

I skal beskrive:

- hvad der er firmaets håb med dette system (hvad er deres vision for systemet eller hvilken værdi er det jeres system skal tilføre deres virksomhed).
- User stories (Product Backlog) og hvilke user stories der har været med i de enkelte sprint (Sprint backlogs). Hver user story består af en kort beskrivelse samt acceptkriterier, dvs. betingelserne for at den kan siges at være færdig.

Jeres løsning skal tage højde for følgende ikke-funktionelle krav:

- MySQL database
- SpringBoot webapplikation med HTML, CSS og Thymeleaf
- User interface designet til brugerne med hensyntagen til heuristikker og The Golden Rules.

## Domænemodel og Database

Det interessante ved domænemodellen er at den langt hen ad vejen er grundlaget for resten af systemet. Databasens tabeller og relationer siger noget om *hvad* systemet arbejder med, ikke *hvordan*. Så det er godt sted at starte.

I skal udarbejde:

- Domænemodel over problemdomænet
- ER diagram over databasen
- SQL Scripts til oprettelse af database og indsættelse af testdata

Mht. databasen bør I beskrive overvejelser i jeres design, f.eks.

- Tabeller der ikke er på 3. normal form, herunder forklare hvorfor det er gjort (tidspres eller anden overvejelse)
- Tabeller der ikke benytter automatisk genereret ID som nøgle
- Valg af fremmednøgler, constraints, unik indeks, osv.

## **Navigationsflow mellem skærbilleder**

I jeres webapplikation oplever brugeren en række websider hvor man kan indtaste oplysninger og gå videre til andre sider. I større systemer kan det være svært at bevare overblikket over hvilke sider der er, og hvordan man kommer fra den ene til den anden.

Udarbejd et aktivitetsdiagram til at vise dette på en overskuelig måde samt screenshots (som bilag).

## **Jeres softwaredesign**

Som slutdokumentation af jeres endelige programdesign, skal I udarbejde:

- et diagram over systemets overordnede arkitektur, f.eks. et UML package diagram eller et screen shot af jeres pakkestruktur.
- et klassediagram over softwareklasserne i jeres program (ikke autogenereret og ulæseligt, men på et detaljeringsniveau, hvor andre programmører rent faktisk vil kunne bruge det til at få overblik over systemet).

Beskriv jeres designovervejelser, herunder brug af designmønstre og andre designprincipper.

## **Særlige forhold**

Dette afsnit bruges til at beskrive særlige forhold der benyttes i programmet. Det kan f.eks. være:

- Hvilke informationer gemmes i session
- Hvordan håndterer I exceptions
- Hvordan har I valgt at lave brugerinput validering
- Hvordan har I (eventuelt) valgt at lave sikkerhed i forbindelse med login
- Hvilke brugertyper er der defineret og hvilke rettigheder har de
- ... andre elementer

Husk: det er bedre med 2 linjers dokumentation end ingen.

## **Udvalgte kodeeksempler**

Det er ikke sikkert at censor (eller eksaminator) finder alle jeres guldskatte i selve koden. Derfor er det en god ide at vælge særlige kode stumper ud og vise dem i rapporten.

De eksempler der er givet under "særlige forhold" afsnittet kan man godt tage og illustrere med kode direkte i rapporten.

Det kommer til at virke særligt overbevisende hvis den kode man vælger ud, har en kompleksitet, som I kan dokumentere at I har testet i projektet enten manuelt eller automatiseret.

## **Status på implementering**

Dette afsnit skal give overblik over hvor langt I er nået med implementeringen.

Typiske ting man kan have sprunget over er:

- Man har ikke nået at lave alle de websider man har med i aktivitetsdiagrammet.
- Man har ikke nået at lave alle CRUD-metoder til alle tabellerne
- Man har ikke fået stylet sine sider
- Man har fundet en fejl "i sidste øjeblik", som der ikke har været tid til at rette. F.eks. forkert brug af sessions, sådan at man på en af siderne kan komme ind uden at være logget ind.
- tests der fejler på afleveringstidspunkt
- ... andre mangler

## Test

Der skal være lavet test. I kan dokumentere tests ved at beskrive i tabelform:

- Hvilke klasser er testet
- Hvilke metoder er testet
- Dækningsgrad af jeres tests for de valgte metoder og klasser.

Desuden kan I beskrive hvordan I systematisk har arbejdet med at teste koden, før den er blevet gjort til en del af master branch.

SQL-scripts

I skal vedlægge jeres sql-scripts i 'resources' folderen i jeres projekt

## Scrum Proces

Der skal være et afsnit hvor I beskriver jeres arbejdsproces i projektperioden. Der skal dels være et faktisk afsnit og dels et refleksionsafsnit.

### Arbejdsprocessen faktisk

Dette afsnit skal beskrive:

- **Daily stand-up** møder - Hvad fik I ud af disse.
- Et eksempel på et **Sprint Planning** møde med Product Owner. Hvad der var planlagt fra jeres side, og hvordan det gik.
- Et eksempel på et **Sprint Review** møde med Product Owner. Hvad der var planlagt fra jeres side, og hvordan det gik.
- **Sprint Retrospective** møder - Hvad fik I ud af disse.

### Arbejdsprocessen reflekteret

Dette afsnit skal beskrive jeres overvejelser over hvilke dele der har fungeret godt og hvilke dele der måske er faldet lidt på gulvet. I kan f.eks. beskrive:

- Om Scrum Master rollen fungerede, hvilke problemer I så i den, og hvad I gjorde for at rette op på det.
- Hvad der var de væsentligste emner på jeres retrospektive møder

- Om I havde problemer med at nedbryde user stories i tasks
- Hvordan I sammen med PO deltog i at planlægge hvilke features (user stories), der egnede sig til hvilke sprints.
- Refleksioner vedr. jeres estimeringer.
- Hvor langt inde i processen I fandt en rytme der var produktiv
- Andre elementer der har at gøre med at forsøge at arbejde i et Scrum team
- Nåede I det I havde planlagt – hvorfor/hvorfor ikke, set i bakspejlets klare(re) lys?

## Teknologi

Jeres webapplikation skal deployes til Azure, så den er offentligt tilgængelig. Jeres MySQL database skal ligeledes deployes til Azure, så den er tilgængelig for webapplikationen.

Jeres GitHub repository skal indeholde et workflow i GitHub Actions som implementerer elementer af:

- *Continuous Integration* (automatiseret integration af feature-branches eller pull requests til main/master efterfulgt af build og test).
- *Continuous Delivery* (deploy til Azure cloud (PaaS)).

I skal tilbyde både et udviklingsmiljø og et produktionsmiljø, som I kan skelne mellem gennem brug af Spring Boot profiles.

Jeres README.md skal som minimum indeholde en kort beskrivelse af de softwaremæssige forudsætninger for at kunne anvende applikationen, et link til den kørende applikation, samt hvad I vurderer, vil være relevant.

Jeres CONTRIBUTE.md skal beskrive hvad der er nødvendigt at vide for et nyt teammedlem, som ønsker at bidrage til koden.

## Studieordning: Krav til rapport og programdokumentation

I skal lave en projektrapport, der dokumenterer systemudviklingen og implementeringen af systemet.

Vigtigt: Det anbefales at I i rapporten og i kildekoden angiver hvem fra projektgruppen, der er ansvarlig for det pågældende kapitel, afsnit, diagram eller kodedel. Dette er primært for at beskytte gruppen som helhed mod eventuelle plagiatanklager.

**Projektrapporten udgør den skriftlige del af eksamen og skal ifølge studieordningen minimum indeholde:**

- Forside med titel, navn og fødselsdato, hold betegnelse og dato samt link til GitHub repository med kildekoden, link til GitHub Projects og GitHub navn på alle deltagere.
- Indholdsfortegnelse

- Indledning inkl. problemformulering
- Hovedafsnit
- Konklusion
- Litteraturliste (inkl. alle kilder, der er lavet henvisninger til i projektet)
- Bilag (inkluder kun bilag, som er centrale for rapporten)
- Der skal pagineres (sidetal) på alle sider

Projektrapporten må maksimalt have et omfang på 40 normalsider, hvis en studerende skriver alene, 50 normalsider for 2 studerende, 55 normalsider for 3 studerende og 60 sider for 4 studerende.

En normalside er 2.400 tegn inkl. mellemrum og fodnoter. Forside, indholdsfortegnelse, litteraturliste samt bilag tæller ikke med i det afleverede antal sider. Bilag er uden for bedømmelse. Hver enkelt figur eller diagram tæller ikke med.

## Projektforløb og vejledning

**Projektaflevering er den 18. december 2024. Projektrapporten afleveres inden kl. 12:00 i Wiseflow.**

Der er vejledning i den angivne projektperiode. Vejledningen finder sted i forhold til det gældende skema og efter aftaler med vejlederne i de enkelte fag.