

Opgaven

I skal lave et **projektkalkulationsværktøj** for jeres kunde Alpha Solutions. Kravene til systemet præsenteres til kickoff mandag 23.11 og er supplerende beskrevet i slides fra Alpha Solutions projektpræsentation.

Derudover er der følgende krav til projektet:

Projektkrav

Generelle krav

Der skal være sporbarhed ml. de forskellige UP discipliner i jeres projekt (dvs. virksomhed, krav, design og kode og test). Det betyder, at I i rapporten skal sikre sporbarhed mellem jeres udarbejdede artefakter i faget Virksomheden til systemkrav og UML-diagrammer og videre fra UML-diagrammerne til jeres kildekode og test cases.

UML diagrammer og udsnit af kode skal ledsages af en supplerende tekst, hvor I beskriver jeres valg og reflekterer over fordele og ulemper ved de valg, som I har truffet.

Krav til Virksomheden

- Stakeholder analyse (Interessent analyse)
- Udvidet Risikoanalyse (Inklusiv Præventive tiltag og PlanB)

Krav til Systemudvikling

Systemudviklingen skal være baseret på Unified Process (UP). Start med at lave en faseplan, der angiver start- og slutdato for faserne: Inception-Elaboration-Construction.

Udarbejd følgende artefakter:

- En Supplementary Specification, der identificerer ikke-funktionelle krav.
- Lav et Use Case Diagram og use case beskrivelser. I skal implementere mindst to centrale use cases. De use cases, som I ender med at implementere, skal beskrives i Fully dressed format. De øvrige use cases beskrives i Brief eller Casual format.
- De to Fully dressed use cases danner udgangspunkt for yderligere diagrammer i rapporten.
- Identificer system events og vis dem i System Sequence Diagrams (SSD)
- Identificer konceptuelle klasser og vis dem i en Domain Model.
- Lav Sequence Diagrams (SD) med udgangspunkt i jeres System Sequence Diagrams (SSD).

- Lav et Design Class Diagram (DCD) på baggrund af jeres Domain Model og Sequence Diagrams. Jeres DCD er er første udkast og behøver ikke være komplet.
- I skal desuden vedlægge:
 - et opdateret Design Class Diagram (DCD), der fungerer som slutdokumentation af jeres program. Husk at det skal være læsbart for at være brugbart.
 - 1-3 Sequence Diagrams (SD), der viser nogle interessante programforløb. Det kunne være et sekvensdiagram over, hvordan et typisk forløb foregår, og et specielt særtilfælde.
- Beskriv de GRASP-mønstre, I har anvendt, eller hvor I har kunne have brugt dem i jeres projekt.
- Lav et pakkediagram, der viser systemets logiske arkitektur
- Lav et State Machine diagram, der illustrerer navigationsflowet i jeres applikation
- Lav en tabel med testcases for de 2 implementerede use cases

Krav til softwarekonstruktion

Jeres løsning skal tage højde for følgende tekniske og designmæssige krav:

- Bruge en MySQL database.
- En logisk lagdelt arkitektur, der kan køre på en Java server (Tomcat).
- Applikationen skal bruge både almindelige Java klasser, Spring Controllere, HTML, CSS og Thymeleaf.

Produktkrav

- GUI er designet til brugerne med hensyntagen til Gestaltlovene og The Golden Rules.

Database

- SQL-script til oprettelse af database og testdata
- Et fysisk ER diagram, inkl. en beskrivelse af tabellerne i databasen (skema)
- En beskrivelse, analyse og vurdering af databasen, herunder til 3. normalform samt overvejelser om, hvorledes I har understøttet databasens integritet.

Programdokumentation

- Et implementeringsafsnit i rapporten med:
 - Beskrivelse af de programdele, der har særlig kompleks karakter (illustrér med indsættelse af kommenteret udklip af kode).
 - En beskrivelse og dokumentation af de løsninger, der kræver en særlig argumentation, f.eks. brug af design patterns, valg af datastrukturer, anvendelse af arv og/eller interfaces.
 - Kommentarer til programkoden om særlige forhold. Det kan f.eks. være:
 - Hvordan håndterer I exceptions?
 - Hvordan har I valgt at lave brugerinput validering?
 - Har I lavet sikkerhed i forbindelse med login – og hvordan?
 - SQL i løsningen (både DDL og DML).

Test

- I kan dokumentere automatiserede tests ved at beskrive i tabelform:
 - Hvilke klasser og metoder er testet
 - Dækningsgrad af jeres tests for de valgte metoder og klasser

Kørselsvejledning

- En beskrivelse af de softwaremæssige forudsætninger for at kunne anvende applikationen, og en vejledning i, hvordan man installerer, starter og kører systemet. Vejledningen skal dække både Java og databasedelen.

Krav til rapport og programdokumentation

I skal lave en projektrapport, der dokumenterer systemudviklingen og implementeringen af systemet. Rapportens målgruppe er underviserne.

VIGTIGT: I rapporten og i kildekoden skal det tydeligt fremgå hvem fra projektgruppen, der er ansvarlig for det pågældende kapitel, afsnit, diagram eller kodedel.

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

- Forside med titel, navn og fødselsdato, hold betegnelse og dato
- Indholdsfortegnelse
- Problemformulering eller problemstilling
- 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
- Link til Github repository med kildekoden og github navn på alle deltagere.

Skal det være muligt for både chef og ansatte at se de forskellige projekter de r tilknyttet

Antal studerende	Max sideantal
1	40
2	50
3	55
4	60

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 800 tegn.

Projektforløb og vejledning

Projektaflevering er den 21. december 2020. Projektrapporten afleveres inden kl. 12:00 i Wiseflow.

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