

Projektoplæg

Portalen

Projektoplæg gælder for semesteropgaven på 1. semester. Opgavens kontekst er samtidigt udgangspunkt for den semesterrapport, der skal arbejdes videre med på fagets 2. semester, hvor der lægges vægt på grafisk brugergrænseflade og databaser. På 1. semester lægges der vægt på de grundlæggende Java-kundskaber.

Indhold:

- Opgaverammer
- Formål
- Portalen - introducerer det kulturcenter, som danner rammen for opgaven
- Det nye system - beskriver en række specifikke opgaver, der skal løses
- Aktører og use cases
- Krav til besvarelsen - beskriver de formelle krav til opgavebesvarelsen
- Rapport
- Forslag til rapportens kapitler

Opgaverammer

Opgaven går ud på at demonstrere anvendelse af nogle væsentlige teorier og værktøjer fra kurset og dermed vise forståelse af og evne til at kunne kombinere og anvende dem i praksis. Der lægges vægt på god struktur og læsbarhed.

Grupper

Semesterrapporten udarbejdes i grupper på højst 4 studerende. Projektoplæg udleveres i uge 40, og semesterrapporten afleveres i uge 1 (2016). Efter uge 40 kan I definere krav og udføre en analyse af jeres kommende system. Først efter uge 42 vil det være muligt at påbegynde design og dermed også programmeringen.

Ved gruppebesvarelse har gruppens medlemmer det fulde ansvar for hele besvarelsen. Det skal dog angives i rapportens indledning, hvilke afsnit og klasser den enkelte studerende har været ansvarlig for at udarbejde (af hensyn til individuel bedømmelse).

Semesterrapporten skrives på dansk og skal have et omfang på højst 50 normalsider. Der kan vedlægges bilag i redigeret form. Læs mere om rapportens format under ”Krav til besvarelsen”.

Projektaflevering

Arbejdet resulterer i en rapport samt kørende program. Alle filer pakkes i én enkelt zip-fil, der afleveres digitalt på Learn senest mandag den 4. januar kl. 12.00.

Evaluerings

Projektrapporten bedømmes med intern censur, og der gives én karakter efter 7-trins skalaen.

Formål

Der skal udvikles en prototype på et administrativt system. Projektet skal således resultere i et kørende program. Formålene med projektet er at:

- vise, på hvilken måde et IT-system kan designes - ved brug af de analyse- og designdiagrammer, der er præsenteret i kurset (UML).
- demonstrere erfaring med brug af Java-sprogets forskellige sætningskonstruktioner – herunder den objektorienterede tankegang.
- Endvidere skal den studerende vise sig som et velfungerende projektgruppemedlem.

Portalen

Portalen er et kulturhus, der afholder mange forskellige arrangementer. Som eksempler kan nævnes åben-scene-arrangementer med jazz den første søndag i hver måned, pop og rock den sidste torsdag samt viseaftener. På den professionelle scene kommer besøg fra bl.a. USA, Skotland, England, Irland, Holland og Sverige samt selvfølgelig mange kendte og nye danske talenter. Portalen har flere scener, et galleri og en café. Der er ca. 90.000 gæster om året.

Portalen har en medlemsklub - Portalis. Som medlem af Portalis får man blandt andet fri entré til udvalgte arrangementer samt tilbudt koncerter til nedsat pris. Mere præcise oplysninger om disse rabatter fremgår af Portalens program. Et medlemskab koster 100 kr. om året.

I kan finde flere oplysninger på adressen www.portalen.dk. Her vil I opdage en del mangler og uhensigtsmæssigheder, som I skal være behjælpelige med at udbedre - nemlig med et helt nyt system. Der skal konstrueres flere forskellige prototyper, der hver især belyser, hvordan forskellige delopgaver kan løses. Disse prototyper skal forevises Portalen, der så vurderer, hvilke af ideerne der skal arbejdes videre på til foråret. Portalen må naturligvis ikke kontaktes her i efterårssemestret.

I løbet af 1. semester udarbejder hele HA(it)-holdet således til sammen forskellige prototyper. De enkelte grupper vælger selv, hvad de vil have fokus på. **Hver gruppe skal have godkendt deres opgaveformulering af Inge-Lise (evt. instruktør).**

På 1. semester udarbejdes den del af koden, der håndterer logikken (domænet) – dvs. indlæsning og arkivering af relevante data, visning af data samt diverse søgninger og beregninger. Al ind- og uddata skal være tekstbaseret. På 2. semester arbejdes videre med opgaven, så arkivering af data foretages i en database og data tilgås via en grafisk web-side.

Det nye system

Portalen har således bedt jer om at være behjælpelige med udviklingen af prototyper til et nyt administrativt IT-system. Nedenfor skitseres et par forslag til opgaver.

Projektforslag 1

Der skal foretages relevante registreringer om alle kulturhusets arrangementer samt besøgstal. Om hvert arrangement registreres følgende oplysninger (eksempler er angivet i parentes): identifikationsnummer, type (koncert), scene (Store scene), dato (27. oktober), tidspunkt (20.00), titel (En gang til for prins Knud), kunstner (Ørkenens sønner), pris (425 kr.), Portalis-tilbud (intet) og besøgstal (fordelt på almindelige gæster og Portalis-medlemmer).

Det nye system skal blandt andet kunne foretage følgende:

- Indskrivning af nye arrangementer. Der kan kun foregå ét arrangement på en scene/i et galleri ad gangen.
- Sletning af et arrangement.
- Udskrivning af alle arrangementers oplysninger.
- Søgning efter et bestemt arrangement.
- Indskrivning af besøgsantal til et bestemt arrangement.
- Rettelser i kartotekets oplysninger (f.eks. søg efter identifikationsnummer og angiv en ny kunstner).
- Udskrivning af alle arrangementer af en bestemt type.

Når et arrangement er oprettet, skal der samtidig ajourføres i relevante registre (der findes ét for hver scene samt for galleriet). Det skal være muligt at få udskrevet scene- og gallerireservationsli-

ster, så man til enhver tid kan se, på hvilke tidspunkter de er ledige. Disse lister skal desuden bruges til den årlige evaluering af belægningsprocenten pr. scene/galleri.

Programmet skal være menustyret, dvs. de muligheder, man som bruger kan vælge mellem, skal vises på skærmen. Når man har foretaget sit valg i en hovedmenu, skal skærbilledet skifte, så man enten kommer til en ny menu (undermenu) eller til et skærbillede, hvor man går direkte i gang med en delopgave. Man kan kun forlade programmet ved at vælge et særligt punkt i hovedmenuen.

Det delsystem, I analyserer, designer og programmerer, skal være et enkeltstående system, der kan afvikles på en enkeltstående PC.

Flere projektforslag

Kun fantasien sætter grænser. Herunder ses eksempler på andre administrative delsystemer, hvor følgende grundlæggende funktioner skal kunne varetages: Opret, ændr, slet, vis alle, vis udvalgte og evt. udarbejdelse af statistikker.

- Kunstnere
- Forestillinger og andre arrangementer
- Billetsalg
- Portalis-medlemmer
- Lokaler
- Økonomi
- Cafeen
- Vagtplaner
- Rengøring og vedligeholdelse

Aktører og use cases

Til projektforslag 1 kan der f.eks. være følgende aktører:

- Kunde – dvs. privatperson, der vil se Portalens program og søge efter en bestemt begivenhed. Dette kræver ikke et login.
- Bruger, der opretter og vedligeholder Portalens arrangementer. Dette kræver et login.
- Administrator, der opretter og vedligeholder systemets brugere.

Herunder ses forslag til use cases:

	<i>Use case:</i>	<i>Kunde:</i>	<i>Bruger:</i>	<i>Administrator:</i>
<i>Use case 1</i>	Opret bruger			X
<i>Use case 2</i>	Slet bruger			X
<i>Use case 3</i>	Opret arrangement		X	
<i>Use case 4</i>	Slet arrangement		X	
<i>Use case 5</i>	Ændr arrangement (kunstner, lokale og/eller tids-punkt)		X	
<i>Use case 6</i>	Opdater arrangement med besøgstal		X	
<i>Use case 7</i>	Vis alle arrangementer med besøgstal		X	
<i>Use case 8</i>	Vis et bestemt arrangement (kunstner, type, tids-punkt) med besøgstal		X	
<i>Use case 9</i>	Vis alle besøgstal for en bestemt type arrangementer		X	
<i>Use case 10</i>	Vis alle arrangementer	X		

Udvalgte af de nævnte use cases beskrives i det følgende. I kan selv dokumentere de resterende – enten med flowcharts eller use case beskrivelser.

Use case 1: Opret bruger

Kort beskrivelse:

Administratoren kan oprette nye brugere.

Forudsætninger:

Administratoren er logget ind.

1. Systemet viser en administratormenu.
2. Administratoren vælger ”Opret ny bruger”.
3. Systemet spørger om brugernavn og password.
4. Administratoren indtaster brugernavn og password.
5. Systemet gemmer oplysningerne sammen med de allerede oprettede brugere.
6. Systemet udskriver en meddelelse.

Supplerende specifikationer:

- Et password skal bestå af mindst 6 tegn, og må kun indeholde bogstaver og cifre.
- Vedr. login: Systemet indeholder fra starten et antal brugere med navn og password. Alle brugeroplysninger ligger i en ArrayList.

Use case 2: Slet bruger

Kort beskrivelse:

Administratoren kan slette brugere.

Forudsætninger:

Administratoren er logget ind.

1. Systemet viser en administratormenu.
2. Administratoren vælger ”Slet bruger”.
3. Systemet spørger om brugernavn og password.
4. Administratoren indtaster brugernavn og password.
5. Systemet sletter brugeren.
6. Systemet udskriver en meddelelse.

Use case 3: Opret arrangement

Kort beskrivelse:

Brugeren kan oprette nye arrangementer.

Forudsætninger:

Brugeren er logget ind.

1. Systemet viser en brugermenu.
2. Brugeren vælger ”Opret nyt arrangement”.
3. Systemet spørger om arrangementets identifikationsnummer, type, scene, dato, tidspunkt, titel, kunstner, pris og Portalis-tilbud.
4. Brugeren indtaster de ønskede oplysninger.
5. Systemet gemmer oplysningerne sammen med de allerede oprettede arrangementer.
6. Systemet udskriver en meddelelse.

Supplerende specifikationer:

- Alle arrangementoplysninger gemmes i en ArrayList.

Use case 6: Opdater arrangement med besøgstal

Kort beskrivelse:

Brugeren kan oprette opdatere et arrangementer med besøgstal.

Forudsætninger:

Brugeren er logget ind.

1. Systemet viser en brugermenu.
2. Brugeren vælger ”Opdater arrangement med besøgstal”.
3. Systemet spørger om arrangementets identifikationsnummer.
4. Brugeren indtaster identifikationsnummer.
5. Systemet udskriver alle oplysningerne om arrangementet.
6. Brugeren indtaster besøgstal.
7. Systemet gemmer oplysningerne sammen med de allerede registrerede oplysninger om arrangementet.
8. Systemet udskriver en meddelelse.

Supplerende specifikationer:

- Systemet må ikke gå ned, hvis der indtastes tekst i stedet for tal.

Krav til besvarelsen

Der skal udarbejdes analyse- og designdokumentation til et delsystem, I selv vælger. Der skal implementeres use cases, der dækker følgende grundlæggende funktioner: Opret, ændr, slet, vis alle, vis udvalgte - samt evt. udarbejdelse af relevante statistikker.

Det er gruppen selv, der vurderer, hvilke use cases der skal udvikles – og hvor mange scenarier (dvs. hvor mange specieltilfælde, der skal tages højde for i et use case-forløb). Der skal programmeres i Java. Endvidere skal den nødvendige projektplanlægning udføres, dokumenteres og vurderes.

Planlægning

Inden et projektarbejde kan gå rigtig i gang, skal der dannes grupper, laves aftaler om samarbejdet i gruppen samt planlægges et forløb. Ved projektets start skal følgende udarbejdes:

- Gruppekонтракт. Overvej en evt. rolle-/arbejdsfordeling i gruppen så I sikrer jer, at alle får programmeret lige meget.
- Overordnet plan for hele projektperioden.
- Udkast til rapportens disposition med forslag til afsnit.

Gruppekонтракten kan f.eks. have følgende indhold:

- Hvem gælder den for og i hvilken sammenhæng?
- Hvordan skal samarbejdet foregå?
- Hvad vil I gøre for at undgå, at det går galt med samarbejdet?
- Hvad vil I gøre, hvis det går galt?

Analyse og design

Et vigtigt kriterie er, at det kommende IT-system skal være vedligeholdelsesvenligt, da resten af systemet skal implementeres senere. Der skal derfor udarbejdes relevante use case beskrivelser og klassediagrammer, der dokumenterer den implementerede kode.

Der er ligeledes vigtigt, at der er tydelig sammenhæng mellem beskrivelser og diagrammer, der er udviklet i analyse og design, og den endelige implementering. Det skal således være muligt at trace fra kravliste til analysedokumentation - til designdokumentation - og til sidst til Java-program. Rapporten kan f.eks. indeholde relevante beskrivelser vedrørende én enkelt use case (analyse- og designdokumentation). Det øvrige materiale kan vedlægges som bilag.

Programmering

Vurdér først, i hvilken rækkefølge det vil være mest fornuftigt at implementere de enkelte use cases. Overvej ligeledes hvordan de nødvendige menuer skal vises for aktørerne.

Omfanget af systemet vil typisk være et system, der kører som en Java-applikation på en bærbar computer. Der er ingen specielle krav til brugergrænseflade udover, at den skal være tekstbaseret.

Brugerdokumentation til prototypen

Der skal udarbejdes en kort vejledning til de brugere, der skal afprøve jeres system (i første omgang er det en repræsentant fra Portalen). Det kan være skærmdumps med korte beskrivelser. Denne dokumentation kan evt. samtidig være bevis for udførelse af udvalgte tests.

Test

Der skal udføres eksempler på accepttests. Disse test-cases skal beskrives (inkl. deres formål), og resultaterne af deres udførelse skal dokumenteres vha. skærmdumps. Tilstræb overskuelighed f.eks. ved hjælp af tabeller, der beskriver tests samt indeholder henvisninger til de tilsvarende testkørsler i bilagene. Husk at udføre tests med rigtige (dvs. læsbare) data.

Rapport

Der skal afleveres en rapport, der beskriver det udviklede system og det gennemførte projektforsløb, samt et program. Formålet med rapporten er at dokumentere resultatet af arbejdet samt redegøre for den proces, I har været igennem - og de overvejelser, I har foretaget jer. Det er vigtigt, at rapportens indhold er disponeret på en måde, så enhver udenforstående kan læse dens indhold i en meningsfuld sammenhæng.

Afleveringen er digital, dvs. **der skal afleveres én zip-fil** i en dertil oprettet eksamensafleveringsmappe på Learn. Zip-filen skal indeholde følgende:

- En pdf-fil, der indeholder rapporten. Husk at indsætte relevante oversigter fra "Formelle krav vedrørende individualisering" før indholdsfortegnelsen.
- En kopi af hele jeres projektmappe med java-program og tilhørende mapper og filer. Det skal være muligt at importere jeres projekt direkte, så det kan afprøves.

Rapportens omfang

Semesterrapporten skrives på dansk og skal have et omfang på højst 50 normalsider med tydelig angivelse af de enkelte studerendes indsats. Det skal dokumenteres, at kravet til omfang er overholdt. Der kan vedlægges højst 30 normalsiders bilag. Husk, at der skal refereres til alle vedlagte bilag.

Hvis en studerende ønsker at udarbejde rapporten alene, skal rapporten have et omfang på højst 30 normalsider og højst 20 normalsiders bilag.

Regler vedrørende opgørelse af sideomfang

Sideformatet er baseret på en normalsidedefinition på 2.275 anslag (inkl. mellemrum), svarende til f.eks. 35 linjer af 65 anslag. Opgavens sider må i gennemsnit ikke indeholde mere end 2.275 anslag. Siderne skal desuden have en margin på mindst 3 cm. i top og bund og mindst 2 cm. i hver side. Skrifttypen skal være på minimum 11 pitch. Tabeller, figurer, illustrationer og lignende tæller ikke med i antallet af anslag, men berettiger ikke til at overskride det fastsatte maksimale sidetal.

I sidetallet medtælles ikke forsiden, litteraturliste og eventuelle bilag. Bilagene indgår normalt ikke i bedømmelsen.

Vigtigt vedrørende koden

I starten af hver klasse skal det angives, hvem der har haft ansvaret for at udarbejde koden. Al kildekode skal vedlægges som bilag.

Semesterrapportens indhold

Alle kapitler skal nummereres og have relevante, nummererede underafsnit. HUSK:

- En forside med en for projektet sigende titel, alle jeres navne, studienumre, gruppenummer og underskrifter, samt i hvilken sammenhæng projektet er udarbejdet
- Skemaoversigten fra "Formelle krav vedrørende individualisering" - her skal I angive, hvem der har haft ansvaret for de enkelte afsnit i rapporten og de kodede metoder
- Kort problemformulering
- Diverse relevante kapitler om jeres programudvikling - lige fra analyse til test
- Konklusion – husk de 3 p'er (proces, produkt og perspektivering)
- Litteraturliste (husk også web-sider)

Forslag til rapportens kapitler

1. Indledning, problemformulering og metode
2. Procesplanlægning mm.
3. Analyse og design
4. Implementering
5. Test
6. Konklusion (husk de 3 p'er)

Litteraturliste

Relevante bilag, indekseret A, B, C, ... med sigende navne og henvisning til sidenumre

Alle kapitler skal have relevante, nummererede underafsnit. Bemærk endvidere, at indholdsfortegnelse, litteraturliste og bilag ikke har kapitelnumre, da de ikke er en del af hovedteksten. Bilag nummereres med bogstaver. Relevante bilag kan bl.a. omfatte:

- Gruppekонтракт
- Installationsvejledning (inkl. hardware-krav, software-krav og filer)
- Brugervejledning med skærmdumps
- Dokumentation til test (skærmdumps) – hvis dette ikke allerede er dokumenteret i en brugervejledning
- Java-programmet

Husk endvidere:

- at skrive alle bilagsnavne i indholdsfortegnelsen
- ved sidehenvisning til de enkelte klasser skal I angive den/de ansvarlige programmør/-er