

KANDIDATNUMMER(E)/NAVN:

DATO:

FAGKODE:

STUDIUM:

ANT SIDER/BILAG:

6 /

FAGLÆRER(E) :

TITTEL :

SAMMENDRAG:

En kort oversikt over hva denne rapporten handler om, de viktigste resultatene og konklusjonene.

De fleste vil lese dette sammendraget for å vurdere om det er interessant nok til å sette seg ned og lese hele rapporten. Bruk derfor god tid på å formulere sammendraget på en slik måte at du trigger leseren sin interesse til å lese hele rapporten.

Sammendraget skal kunne leses isolert, uten at man behøver å ha kjennskap til rapporten forøvrig.

Denne oppgaven er en besvarelse utført av student(er) ved NTNU.

INNHold

1	Sammendrag.....	1
2	Begreper og forkortelser.....	1
3	Introduksjon.....	1
3.1	Bakgrunn.....	1
3.2	Avgrensninger.....	1
3.3	Begreper/Ordliste.....	1
4	Teori.....	2
5	Kravspesifikasjon.....	2
6	Teknisk Design.....	2
7	Utviklingsprosess.....	2
8	Implementasjon.....	3
9	Testing.....	3
10	Utrulling til sluttbruker (deployment).....	3
11	Drøfting.....	3
12	Konklusjon - Erfaring.....	3
13	Referanser.....	4
14	Vedlegg.....	4

Figurliste

Fant ingen figurlisteoppføringer.

Tabelliste

Fant ingen figurlisteoppføringer.

*[Denne rapporten inneholder ferdigdefinerte **stiler** som du/dere kan benytte for de mest vanlige avsnittene. Følgende stiler er definert:*

Heading 1/Overskrift 1 Overskrift på nivå 1

Heading 2/Overskrift 2 Overskrift på nivå 2

Heading 3/Overskrift 3 Overskrift på nivå 3

Brødtekst Standard tekst i et avsnitt. Benytt denne for all "vanlig" tekst

Definition Benyttes hovedsakelig i avsnittet "TERMINOLOGI"

References Benyttes i REFERANSER-avsnittet.

Comment Denne grønne teksten. Fjern all tekst av denne typen i rapporten.]

1 SAMMENDRAG

[Samme tekst som på forsiden.

En kort oversikt over hva denne rapporten handler om, de viktigste resultatene og konklusjonene.

De fleste vil lese dette sammendraget for å vurdere om det er interessant nok til å sette seg ned og lese hele rapporten. Bruk derfor god tid på å formulere sammendraget på en slik måte at du trigger leseren sin interesse til å lese hele rapporten.

Sammendraget skal kunne leses isolert, uten at man behøver å ha kjennskap til rapporten forøvrig.]

2 BEGREPER OG FORKORTELSER

[Definisjoner, begreper og symboler som kan være ukjente for leseren. (Bruk stilen "Definition", som vist nedenfor). Utelat dette kapitlet dersom du ikke anvender begreper og symboler som det er behov for å definere.

*NB! Ta med kun begreper og forkortelser som du **faktisk** benytter i rapporten!]*

UML	Unified Modeling Language
UP	Unified Process
OOP	Objekt Orienteret Programmering

3 INTRODUKSJON

3.1 Bakgrunn

*[Dette er første kapitlet i den faglige rapporten. Det bør behandle bakgrunnen for oppgaven, eventuell oppdragsgiver, problemstillingen og/eller oppgaven som skal løses. Det er også viktig å beskrive hva som **ikke** skal løses (altså systemavgrensninger).]*

3.2 Avgrensninger

*[Er det gitt noen avgrensninger/begrensninger i oppgaven? Beskriv i så fall disse her. Avgrensninger kan f.eks. være teknologier dere **må** bruke, el.l. Dersom prosjektet ikke har noen avgrensninger, kan dette kapitlet utelates.]*

3.3 Begreper/Ordliste

*[Når man utvikler programvare for en kunde, er det viktig å etablere en felles forståelse for begreper/terminologi/ord som benyttes **av kunden**. Det er derfor svært vanlig å lage en "ordliste" og/eller en "Domene modell". Denne ordlisten er også et svært godt utgangspunkt for å finne frem til hvilke mulige **klasser** det kan være aktuelt å implementere i løsningen når denne skal utvikles. Bruk tid på denne slik at du har en god forståelse for begrepene.*

*Her kan mang også bruke klasse-diagrammer for å illustrere hvordan **begreper** henger sammen. NB! Klasser i dette diagrammet er **ikke** klassene du har valgt å implementere i prosjektet.]*

]

Begrep (Norsk)	Begrep (Engelsk)	Betyding/beskrivelse
Produkt	Product	Varehuset skal håndtere produkter inn og ut av et lager

Begrep (Norsk)	Begrep (Engelsk)	Betyding/beskrivelse
Lager	Storage
Konto	Account	En konto til en kunde i banken. En konto kan ha en balanse, og man kan sette penger inn eller ta ut penger fra konto
Transaksjon	Transaction	Når et beløp tas ut av en konto og føres inn på en annen konto, utføres en transaksjon . En transaksjon har en dato, en tid, hvilken konto beløpet trekkes fra, hvilken konto beløpet overføres til, samt beløpet.

4 TEORI

[Her beskriver du de **teoriene** og **beste praksiser** som man benytter på generelt grunnlag ved utvikling av en programvare, som kobling, cohesion, SOLID osv.]

Når du senere skal skrive drøfting og konklusjon, må du henvise tilbake til dette kapittelet og svare på om løsningen du har levert er løst i henhold til disse. Unngå å beskrive om grunnleggende teorier som «hva er OOP», «Programmeringsspråket Java» osv.]

5 KRAVSPESIFIKASJON

[Her beskriver du både de **funksjonelle** kravene og de **ikke-funksjonelle kravene** til løsningen du skal utvikle.]

Dersom det er gitt en kravspesifikasjon vil du kunne hente det meste av informasjon fra denne. Husk at du her IKKE skal beskrive noen av de valg du har gjort i prosjektet, eller det du konkret har utviklet i prosjektet.

Bruk her gjerne **UML-diagrammer** som **Use-Case**, **Aktivitetsdiagram** osv for å beskrive krav til funksjonalitet (NB! Uten å dra inn **hvordan** du/dere har løst det.).

Hvis en klassisk fossefalls-utviklingsprosess følges, er dette kapittelet et produkt av kravfasen. Hvis du har valgt en mer smidig modell som, f.eks. SCRUM, vil kravene vises gjennom prosjektet som for eksempel brukerhistoriene utviklet i gjennom sprintplanleggingsmøtene.

Når du senere skal skrive drøfting og konklusjon, må du henvise tilbake til dette kapittelet og svare på om løsningen du har levert svarer på kravspesifikasjonen].

6 TEKNISK DESIGN

[Kapittelet om teknisk design beskriver det store bildet av valgt løsning. For et programvareutviklingsprosjekt vil dette vanligvis inneholde systemarkitekturen (klient-server, sky, databaser, tjenester, desktop-applikasjon osv.); både hvordan det ble løst, og, enda viktigere, hvorfor denne arkitektur ble valgt]

7 UTVIKLINGSPROSESS

[I dette kapittelet skal du beskrive prosessen som du fulgte. Den skal dekke prosessmodellen, hvorfor den ble valgt og hvordan den ble implementert, inkludert verktøy for prosjektledelse, dokumentasjon osv. Avhengig av hvordan du skriver de andre kapitlene i rapporten, kan det være gode grunner til å plassere dette kapitlet et annet sted i oppgaven.]

I en typisk mappe-oppgave der dere har jobbet gjennom flere stadier/deler av prosjektet, og fått tilbud om tilbakemelding, bør dette beskrives som en del av prosessen. NB! Ikke beskriv når- og hvor mange tilbakemeldinger dere fikk, men hva som var den opprinnelige rammen/planen.]

8 IMPLEMENTASJON

[Her beskriver du de tekniske detaljene til løsningen du har endt opp med. Hvilke verktøy ble brukt (programmeringsspråk, biblioteker, IDEer), hvilke eksterne biblioteker og rammeverk, APIer osv.]

*Vis med **klassediagrammer** hvordan klassene i løsningen din henger sammen. Bruk **aktivitetsdiagramm(er)** for å beskrive logikken/flyten i løsningen din. Fint også om du kan vise hvordan objekter av klassene dine **samhandler** for å løse **de mest sentrale** oppgavene i løsningen, da i form av **sekvensdiagramm(er)**.*

Klassene dine bør beskrives i forhold til hvilken rolle og ansvar de har, men det er ikke nødvendig å beskrive hver enkelt metode eller felt til klassen. De står uansett dokumentert i Javadoc'en din.]

9 TESTING

[Beskriv hvordan løsningen din har blitt testet, både i form av enhetstesting, men også brukertesting. Hvor mange brukere har testet løsningen? Hvordan ble brukertesting gjennomført? Vis også eventuelt resultatene fra brukertesting her. (svar på spørreskjema el.l.)]

10 UTRULLING TIL SLUTTBRUKER (DEPLOYMENT)

[Her beskriver du hvordan programvaren din gjøres tilgjengelig for sluttbruker. Rulles den ut å en web-server? I så fall hvordan? Lages det en desktop-applikasjon som bruker kan dobbelklikke på for å starte? Osv.]

11 DRØFTING

[Her oppsummerer du/dere oppgaven. Hvor langt kom du/dere (resultat)? Hva fikk du/dere ikke gjort i forhold til oppgaveteksten? Hva var de store utfordringene/problemene du/dere møtte, etc..]

Spesielt viktig er det å drøfte din egen løsning i forhold til det du har lært om gode prinsipper for design av programvare (robust kode, kodelstil, designprinsipper osv) som beskrevet i teori-kapittelet.

Her bør man også gjøre seg tanker rundt kvaliteten av det arbeidet som er nedlagt.

Er de kildene du/dere bruker pålitelige, er det sprik mellom forskjellige kilder (og i så fall hvorfor), er det andre forhold som kan være med å gjøre noen av de vurderinger og valg du/dere har gjort usikre?]

12 KONKLUSJON - ERFARING

[Overbevisninger /erfaring som en er kommet fram til på grunnlag av det presenterte materialet.]

- *Fikk du realisert hele problemstillingen fra kapittel «5 Kravspesifikasjon»?*
- *Hva ville du ha gjort annerledes dersom du kunne begynne på nytt?*
- *Hva slags begrensninger kan en forvente når en bruker løsningen?*
- *Hva skal tas opp i fremtidige arbeid dersom du eller noen andre ville ha tatt utvikling videre?]*

13 REFERANSER

[Forfatter, årstall, tittel på bok eller artikkel, navn på tidsskrift eller forlag/utgiver, nr. eller dato for tidsskrift, sted som det vises til eller refereres fra i oppgaven.]

Konkret for programmeringsemner: Regner med at du/dere kommer til å måtte slå opp litt i læreboka, så den er en innlysende referanse. Dersom du/dere i tillegg benytter internett, så list URL'er til sidene du/dere har benyttet.]

- [1] "Objects First With Java", Sixth edition, av Barnes og Kölling. ISBN
- [2] <http://.....>
- [3] Osv.

14 VEDLEGG

[Materiell som er utarbeidet eller innsamlet i tilknytning til rapporten, men som det ikke er naturlig eller hensiktsmessig å ta inn i hoveddelen, som feks brukerveiledning, skal tas inn som vedlegg.]

Vedleggene skal være nummererte og ha en overskrift.

Har du/dere ingen vedlegg, så droppes dette kapittelet.]