• Alle underdeler teller likt.

• Dokumenter og begrunn eventuelle forutsetninger.

Oppgave 1: Flervalgsoppgave (25%)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på den ene selv). Du kan få nytt ark av eksamensvaktene dersom du trenger dette. Kun ett svar er helt riktig. For hvert spørsmål gir korrekt avkryssing 1 poeng. Feil avkryssing eller mer enn ett kryss gir −1/2 poeng. Blankt svar gir 0 poeng. Du får ikke mindre enn 0 poeng totalt på denne oppgaven. Der det er spesielle uttrykk står den engelske oversettelsen i parentes.

1) Hva omfattes av de følgende i definisjonen av programvare?
   a. Det kjørende programmet
   b. Det kjørende programmet og all programkode som blir utviklet, inkludert skript
   c. Alle dokumenter og skriftlige artefakter produsert under utviklingsprosessen
   d. Alle alternativene er riktige

2) Hvilke prinsipper gjelder for god design?
   a. Høy kohesjon (sammenheng/lim) og løs kobling
   b. Mest mulig gjenbruk av eksisterende programvare
   c. Bruk av et effektivt programmeringsspråk og lesbar kode
   d. Bruk av designmønstre (patterns) så mye som mulig

3) Det finnes to hovedstrategier/prosesser å utvikle programvare på:
   a. Fossefall eller inkrementell
   b. Plandrevet eller smidig
   c. Rask eller treg
   d. Rational Unified Process eller Scrum

4) Hvilken fase i programvareprosessen tar vanligvis lengst tid og koster mest penger?
   a. Design og implementering
   b. Validering og verifisering
   c. Evolusjon
   d. Spesifikasjonsarbeid

5) Under utvikling av et system er det ønskelig å …
   a. ha mest mulig kunde/brukerkontakt for å sikre at man utvikler riktig system
   b. bruke enten en plandrevet eller smidig prosess
   c. utvikle med tanke på mest mulig gjenbruk
   d. optimalisere koden mest mulig for å sikre høy ytelse

6) Hva er hovedforskjellen mellom COTS og egenutviklede komponenter?
   a. Større tillit til kvalitet på egenutviklede komponenter enn COTS
   b. Kvalitet- og kvalitetssikring håndteres bedre for egne komponenter
   c. Kostnader knyttet til COTS er normalt lavere enn til egne komponenter
   d. Support og feilretting er billigere for egne komponenter

7) Inspeksjon av designdokumenter kan benyttes til å sjekke om kravoppfyllelse i forhold til...
   a. Systemytelse
   b. Pålitelighet (dependability)
   c. Kodestandarder
   d. Alle er riktig

8) Når er det som regel lurt å bruke gradvis leveranse av programvaren?

a. Når systemet har høyt krav til sikkerhet (safety)
b. Når teamet er distribuert
c. Når det forventes endring i kravene
d. Alle er rett

9) Testdrevet utvikling betyr at...
a. Man utfører ekstremprogrammering (XP)
b. Tester utvikles etter hvert som man finner feil
c. Man skriver tester før selve koden slik at riktig kode gjør at testen passerer
d. Hele testregimet er under utvikling gjennom utviklingsperioden

10) Hovedformålet med forstudiefasen (fase 0) i utvikling av programvaresystemer er å …
a. Dokumentere brukerkrav til systemet
b. Designe og utvikle systemet
c. Gjennomføre en forberedende analyse av behov, virksomhet og interessenter
d. Velge utviklingsprosess og etablere prosjektgruppe

11) Hva gjør en enhetstest?
a. Tester at ulike deler av systemet fungerer sammen på korrekt måte
b. Tester at selve datamaskinen (maskinvaren) fungerer
c. Tester individuelle deler av programvaren
d. Ingen av alternativene er riktig

12) Etiske retningslinjer innenfor profesjonen programvareutvikling …:
a. Er definert av arbeids- eller oppdragsgiver
b. Er definert basert på vurdering av hvilken type programvaresystem som skal utvikles
c. Er definert av fagfeller som et sett av forventninger og regler til oppførsel, kvalitet og kunnskap til en programvareutvikler.
d. Er avhengig av at arbeids- eller oppdragsgiver betaler for gjennomført arbeid.

13) Hva kjennetegner best prosjektrammeverket Scrum?
a. Stand-up møte, sprintkø, Scrum-team og produkteier
b. Definerte seremonier, artefakter og roller
c. Raske leveranser og mange kundemøter
d. Par-programmering, test-først og kontinuerlig integrasjon

14) Hvilke faktorer påvirker mest til lavere pris til kunde dersom antatt høyere kostnadsestimat?
a. Antall komponenter som kan gjenbrukes
b. Markedsmuligheter og eierskap til kode
c. Veldefinerte krav og teknologi
d. Dokumentert høy produktivitet på prosjektmedarbeidere

15) Risikostyring innbefatter?
a. Identifisering, planlegging, analyse, overvåkning og håndtering av forskjellige type risikoer
b. At prosjekter stanses når gitte hendelser er svært sannsynlighet og har store konsekvenser
c. At prosjekter blir forsinket pga. mye byråkrati og administrasjon
d. Alle er riktig

16) Hvilke egenskaper ved programvare påvirkes mest av OWASP TOP-10?
a. Brukbarhet
b. Pålitelighet (dependability)
c. Ytelse
d. Mulighet for gjenbruk

17) Hva betyr FURPS i kravsammenheng (RE)?
a. Findability, updateability, reconcilability, programmability, specification

b. Functionality, usability, reliability, performance, supportability
c. Functionality, understandability, reliability, possibility, standardisation
d. FUture Requirement Process System

18) Hva er hovedfordelen med klient/tjener arkitekturmønster?
a. Tjenere kan være distribuert over et nettverk
b. Ytelsen blir som regel bedre
c. Høyere robusthet mot feil
d. Spredning av tjenester og bedre sporbarhet av klienter

19) Hva er hovedfordelen med standarder innenfor programvareutvikling?
a. Muliggjør at eldre, «legacy» programvare fortsatt kan benyttes
b. Muliggjør at man kan utvikle i hvilket som helst programmeringsspråk
c. Muliggjør raskere teknologiskifter
d. Muliggjør gjenbruk av kompetanse og forståelse på tvers av systemer og bedrifter

20) DIFI har definert sju arkitekturprinsipp. Seks av disse er tjenesteorientering, interoperabilitet, sikkerhet, åpenhet, fleksibilitet og skalerbarhet. Hva er det sjuende?
a. Testbarhet
b. Funksjonsstabilitet
c. Tilgjengelighet
d. Gjenbrukbarhet

INNLEVERING

Svarskjema: Oppgave 1

Oppgavenr	A	B	C	D
1.1				X
1.2	X			
1.3		X		
1.4			X	
1.5	X			
1.6			X	
1.7		X		
1.8			X	
1.9			X	
1.10			X	
1.11			X	
1.12			X	
1.13		X		
1.14		X		
1.15	X			
1.16		X		
1.17		X		
1.18	X			
1.19				X
1.20			X	

Oppgave 2 – Programvareutvikling (30%)

Personvern innenfor databehandling og informasjonsteknologi er et tema som opptar mange.

Mange virksomheter har IKT-løsninger som samlet holder til dels veldig detaljert informasjon knyttet til enkeltpersoner. Personopplysninger i en IKT-løsning kan gjerne være kopiert fra andre IKT-løsninger og mye av informasjonen som lagres i løsningene er/eller burde vært den samme. En del av denne informasjonen kan være sensitiv og/eller kan (mis)brukes til andre formål enn det som opprinnelig er gitt tillatelse til (fra myndigheter eller personen selv). Dette fører til en rekke problemstillinger:

- Det finnes ikke en klar autoritativ kilde til personopplysninger. IKT-løsninger kan etterhvert inneholde til dels motstridende og utdatert informasjon.
- Vedlikehold og lagring av personinformasjon skjer i mange systemer og krever til dels mye ressurser av virksomheter – både persontimer, utstyr, programvare og energi – for å holde opplysningene oppdaterte og konsistente.
- Det eksponeres sensitiv informasjon til personer som ikke skal ha eller som ikke har behov for det i sin jobb.
- Det er vanskelig for individer å få slettet data etter eget ønske.
- Outsourcing av drift gjør at man ikke har god nok oversikt over hvem som har tilgang på informasjon.

Fra et personvernssynspunkt ønskes det at:

- Hver enkelt person skal kunne vite hvilken informasjon som er registrert om seg og av hvem.
- Eierskap til en del informasjon tilfaller den person informasjon er registrert på, og det bør være mulig for personer å begrense eller hindre tilgang til slik informasjon dersom det ikke er et definert behov for tilgang.
- Personer skal ha tillit til systemer som inneholder og behandler informasjon og transaksjoner knyttet til dem. Informasjonen skal lagres slik at den ikke skal kunne misbrukes eller brukes uten et klart og lovlig formål.
- Systemer som skal bruke personlig informasjon trenger samtykke fra personen
- Vedlikehold, kvalitet og sikkerhet knyttet til personinformasjon sikres gjennom bruk av troverdige kilder for denne type informasjon

Firmaet *IdMegler* har blitt engasjert av den norske stat til å utvikle en IKT-løsning med hensikt å lage en megler- og innsynstjeneste for bruk av personinformasjon. IdMegler etableres som en uavhengig aktør finansiert av den norske stat med ansvar for å utvikle og drifte tjenesten.

Løsningen skal

- Muliggjøre et fysisk og elektronisk skille mellom informasjon om en person og transaksjoner/ dynamisk informasjon som er knyttet til person (f.eks. helseopplysninger, banktransaksjoner, studentresultater, osv.).
    - Direkte personidentifiserbar informasjon (navn, fødselsnummer, o.l.) skal skilles ut på en slik måte at denne informasjon ikke kan benyttes for å finne transaksjoner knyttet til personen i andre IKT-løsninger.
    - Et felles personregister (Folkeregisteret) skal benyttes som den eneste autoritative kilde for personinformasjon.

- Folkeregisteret inneholder basisinformasjon om personer (fødselsnummer, navn, kjønn, fødested, fødeland, fødselsdato, foreldre, barn) og er autoritativ på nåværende og tidligere

navn og adresser. Folkeregistret tilbyr en tjeneste som kun kan benyttes av IdMegler: *PersonInformasjon* hentPerson(*fødselsnummer*)

- IdMegler er ansvarlig for å skape, vedlikeholde og tilgjengeliggjøre nøkler for andre IKT-løsninger (studentsystem, banksystem, helseinformasjonsystem, osv.), samt koble disse med fødselsnummer fra Folkeregisteret.

- Virksomheter og IKT-løsninger må derfor meldes inn i IdMegler med informasjon om hvilken virksomhet som benytter tjenesten og hvilken IKT-løsning som skal benytte personinformasjon. Virksomheter eller systemer som ikke er medlem, kan ikke benytte IdMegler. Dette for å sikre at det er mulighet for en trygg sammenkobling mellom Folkeregisteret og IKT-løsningen.

- Alle transaksjoner relatert til person lagres dermed med nøkler tilgjengeliggjort gjennom IdMegler, og IdMegler benyttes for å sammenkoble informasjon ved hjelp av nøkkelmegling mellom virksomhet og Folkeregister.
- All bruk av IdMegler skal logges hos IdMegler.
    - Oppslag av personinformasjon logges med *hvem* (virksomhet/saksbehandler/ ip-adresse/system/systemprosess etc.), hva, når og hvorfor ble det gjort et oppslag.
- Det skal lages en innsynstjeneste for privatpersoner som skal kunne logge seg inn på IdMegler for å få en oversikt over hvilke virksomheter og IKT-løsninger som benytter IdMegler og transaksjoner knyttet til disse.

Forutsetninger:

- All kommunikasjon skjer kryptert

- IdMegler lager, vedlikeholder og lagrer nøkkelkoblinger mellom Folkeregisteret og virksomhetssystemer.

- Privatpersoner benytter IdPorten, BankID eller MinID for innlogging på IdMegler

- IdMegler har ikke innsyn til opplysninger i Folkeregister eller til IKT-løsninger som benytter IdMegler. Koblinger mellom Folkeregisteret og IKT-løsningen skjer bare når IKT-løsningen ber om at IdMegler gjør en kobling mellom en identifikator (enten fødselsnummer eller tidligere generert surrogatnøkkel) og informasjon i Folkeregisteret.

a) Definer de viktigste funksjonelle kravene til IdMegler basert på beskrivelsen over.

b) Beskriv de viktigste interessentene/aktørene til IdMegler og hvordan IdMegler kan tilfredsstille disse. Lag UML Use-case-diagram for IdMegler for hver aktør i løsningen.

c) Definer de tre viktigste ikke-funksjonelle kvalitetskrav/arkitekturkrav til IdMegler.

d) Hvilken eller hvilke arkitekturmønstre (patterns) bør et slikt system benytte? Beskriv de tre viktigste.

e) Lag en arkitekturskisse for IdMegler. Hvilke hovedkomponenter og koblinger/grensesnitt må løsningen tilby? Fokuser på grensesnittene mot aktørene (bruker, systemer fra oppgave 2a). Definer de viktigste klassene og lag et UML klassediagram for IdMegler

f) Definer de viktigste tjenestene/metodene for IdMegler. Beskriv disse på formen <response> metodeNavn(<parametere>)

Løsningsforslag oppgave 2.

Som med alle LF så er dette kun et forslag til løsning.
God argumentasjon og tydelige antagelser teller i studentens favør.

a) En god løsning kan inneholde flere av disse:

IDmegler skal (funksjonelle krav):

Mot virksomhet/tjeneste:

- Tilby en registreringstjeneste for virksomhet
- Tilby en registreringstjeneste for IKT-løsning hos virksomhet
- Tilby en tjeneste for avslutning av bruk for tjeneste/virksomhet

Kobling av id mellom Folkeregister og Tjeneste hos virksomhet

- Tilby en koblingstjeneste for oppslag av id mot Folkeregister
- Tilby et sentralt sted hvor tjenester kan få oppdatert, konsistent og autorativ data om brukere.
- Tjenester skal kunne foreta autentisering mot IDmegler og få en anonymisert bruker-ID tilbake.
- Tilby oversettelsestjeneste fra anonymisert bruker-ID til tilhørende informasjon som brukeren har valgt å eksponere til tjenesten.

Privatperson/bruker:

- La bruker kontrollere hvor mye informasjon de ønsker å tilby tjenester.
- Tilby bruker oversikt over all deres informasjon som ligger på tilkoblede tjenester
- Tilby bruker transaksjonslogg over hvilke tjenester som har aksessert en persons personlige data.

b) Interessenter / aktører:

- Privatperson/bruker – Sentral interessent som systemet skal ivareta gjennom styrking av personvern, gjennom sikker (konfidensiell), pålitelig (tilgjengelighet) og korrekt informasjon (integritet) til databehandlere som har eller må ha tilgang til personopplysninger. Interessenten gis også mulighet til å sjekke hvilken informasjon som deles med hvem og i hvilken kontekst.
- Virksomhet/bedrift m/ tilhørende tjenester – databehandler og bruker av tjenesten. Krever høy tilgjengelighet og pålitelighet ifht. IdMegler. Hvis dette ikke oppnås, vil egne tjenester ikke

fungere, noe som vil gå ut over privatpersoner som benytter tjenester hos virksomhet samt egen mulighet til inntjening eller tjenestenivå.

- MinID/BankID (eksterne tjenester som brukes i IdMegler) – vil være en «passiv» aktør/ interessent.
- Folkeregisteret – Informasjonsforvalter av personinformasjon. Tilgang til informasjon kun gjennom autentiserte og autoriserte tjenester. Sikrer mot misbruk og spredning av personinformasjon gjennom meglertjenesten.
- Stat: Lov- og kontrollmyndigheter (f.eks. Stortinget, Politi, Datatilsynet). Setter premissene for tjenesten gjennom lover og regelverk, samt står som finansiør.

Bruker Use Cases:

- Login/logout gjennom MinID, BankID etc.
- Rapportere hvilke virksomheter og tjenester som har adgang til sin personlige informasjon.
- Begrense adgang til egen personlig informasjon.
- Gi samtykke til tjenester som vil ha innsyn
- Rapportere hvilke virksomheter og tjenester som har hentet ut personlig info om deg
- Se aggregert informasjon fra tilkoblede tjenester.

Bedrift Use Cases:

- Login (Sende brukere til IDmegler, få anonymisert ID i retur)
- Melde seg inn i IDmegler
- Melde tjeneste inn i IDmegler
- Melde tjeneste ut av IDmegler
- Melde seg ut av IDmegler
- Kunne søke opp gitte personer pr anonymisert ID
- Få ut informasjon om gitte personer pr anonymisert ID

c) Ikke-funksjonelle krav kan inneholde:

Ikke-funksjonelle krav til IDMegler vil i hovedsak være knyttet til Pålitelighet (Dependability) og ytelse (Performance). Tjenesten må være funksjonsstabil og ha høy integritet på informasjon som forvaltes og utveksles (reliability), tilgjengelig (availability) og sikker (security og safety). Personvern (privacy) må ivaretas. All data må krypteres og adgangbegrenses.

- Tjenester som benytter IDmegler skal ikke trenge- eller kunne vite om hverandre, eller kunne benytte utdelte tokens / informasjon til å gjøre kryssoppslag i hverandres databaser.
- Informasjon må lagres slik at den ikke kan misbrukes.
- Sikre at alle tjenester som skal ha tilgang på data har tilgang til en oppdater og autorativ kilde på disse med høy oppetid.

Fra et brukerperspektiv så må tjenestene være slik at

- Det skal være enkelt for brukere å få oversikt over hvilke aktører som har tilgang til deres personlige informasjon.
- Det skal være enkelt for brukerne og velge hva slags informasjon som de ikke ønsker å dele

Krav:

Tjenesten må være tilgjengelig 24/7.

Tjenesten skal kunne skaleres slik at alle bedrifter kan benytte tjenesten.

IdMegling med tilhørende flyt av personinformasjon fra Folkeregister til virksomhet må kunne gjøres innen ett sekund (tidskrav kan diskuteres).

Tjenesten skal sikres på en slik måte at virksomheter må være både autentisert og autorisert til å gjøre oppslag.

Informasjon som logges gjennom tjenesten må være lagret sikkert og kryptert.

Tilgang til informasjon kan kun gis med riktig autentisering og autorisasjon.

Informasjon som håndteres må beholde sin integritet og systemet må aldri levere feile nøkkelkoblinger gitt at en kobling har blitt utført (pålitelighet – funksjonsstabilitet)

d) Følgende arkitekturmønstre (patterns) er blant de relevante:

Client / server.

Siden tjenesten må kunne håndtere mange brukere på en gang, er client / server en godt egnet arkitektur. En kan lett tilby tjenesten som et cluster av servere, og dermed tåle høy pågang (Noe tjenesten kan oppleve hvis den f.eks er koblet opp imot altinn og selvangivelsen kommer)

Komponentbasert og/eller Service oriented architecture.

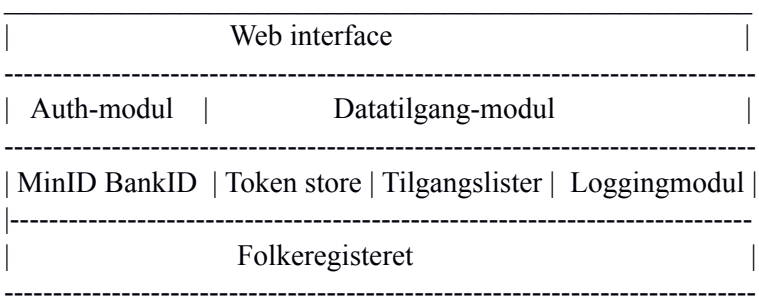
Tjenesten består av en rekke komponenter, mange modulære, som lett kan byttes ut. Autoriseringstjenesten kan benytte en rekke tjenester, som MinID og BankID.

Tjenesten aksesseres via web-sider for vanlige brukere, eller via et veldefinert api som tjenester kan benytte for å programmatisk hente ut tilgjengeliggjort informasjon. Alt over https.

Layered architecture.

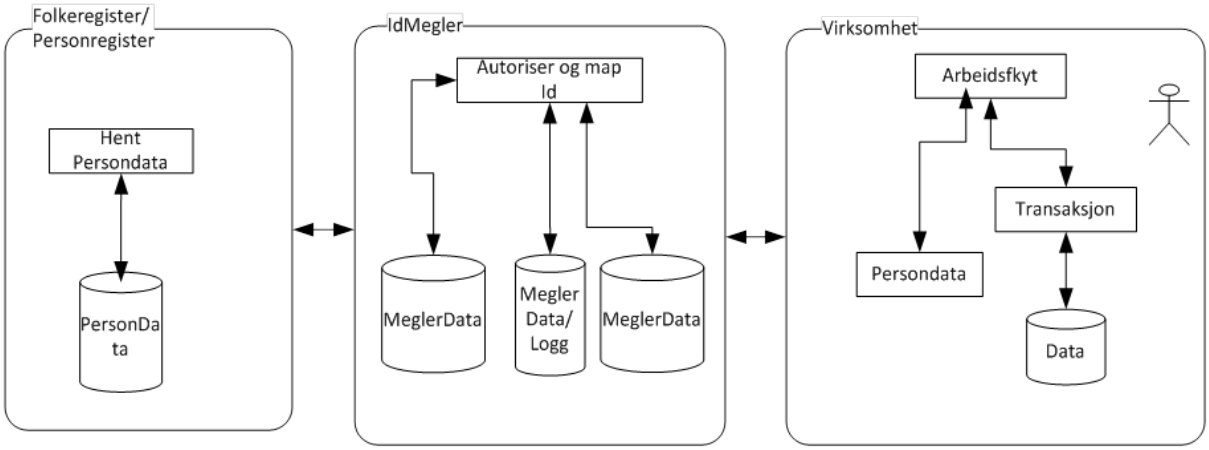
Selve IDmegler kjernen kan lett designes som et lagbasert system, der hvert element kun vet om laget som er direkte under. Dette sikrer modularitet samt god struktur.

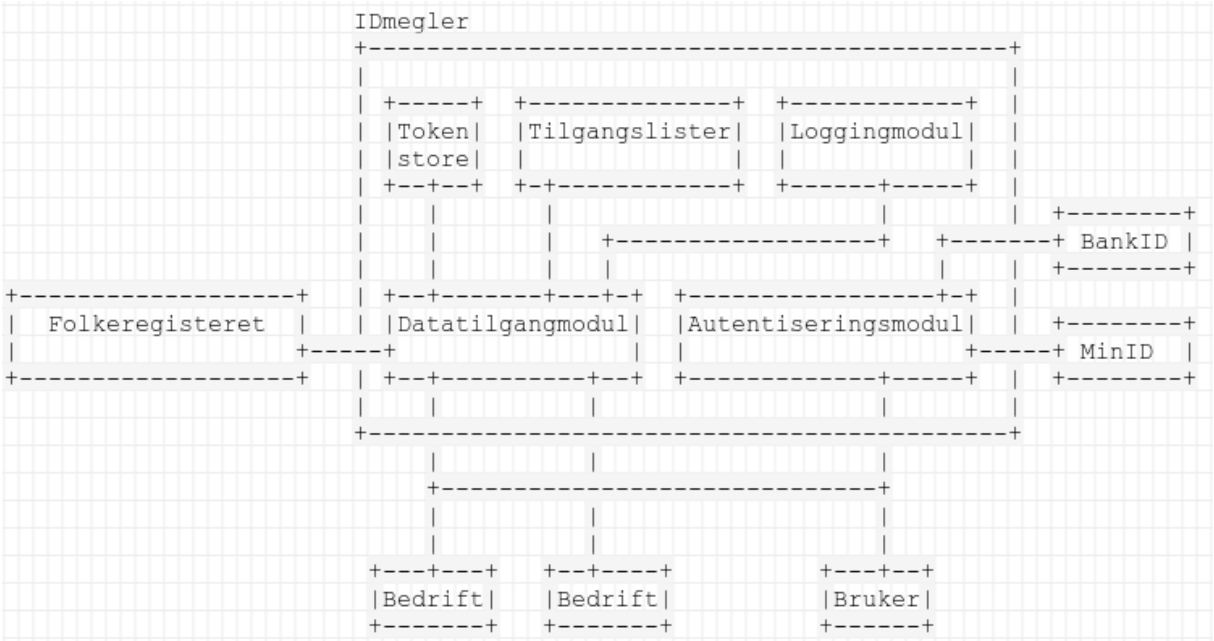
Eksempel på layered architecture for IDmegler



Det kan også argumenteres for en repository-arkitektur.
Da bør man få fram at de forskjellige tjenestene har forskjellige adgangsnivåer til den delte datakilden.

e) Under følger eksempelskisser på systemarkitektur:





Token store inneholder mappinger fra anonymiserte bruker-IDer (tokens) til ekte brukerID.
Tilgangslistene inneholder hvilke rettigheter brukere har gitt til forskjellige tjenester.
Loggingmodulen inneholder alle spørringer som er gjort mot en bruker gjennom datatilgangmodulen.

Når en bruker skal logge inn i en tjeneste, så videresendes man til autentiseringsmodulen der man logger inn med MinID o.l. Etter sukksessfull innlogging vil tjenesten få en anonymisert bruker-ID (et token) i return samt et session-token med begrenset levetid som kan brukes for å gjøre oppslag i datatilgangsmodulen.

Det viktige her er altså at tjenesten aldri får vite personnr. til bruker, siden innlogging skjer på et annet nettsted (IDmegler.no), og tjenesten kun får tokenet tilbake. Dette fører til at brukeren eksplisitt må gi tjenesten tilgang til personnr fra kontrollpanelet. Dette åpner for bruk av IDmegler til innlogging i tjenester som ikke trenger å vite personnr, men kun trenger en garanti for at vedkommende er norsk statsborger.

Datatilgangmodulen godtar spørringer etter anonymiserte brukerIDer og en tilhørende ServiceSessionToken. Hvis det fins en match med denne kombinasjonen vil data returneres til tjenesten, etter de begrensninger som er satt opp av brukeren.

Denne informasjonen kan godt presenteres som UML-komponenter (trenger / leverer), oppramsing, inlinet i UML-diagrammet o.l.

Klassediagrammet tar utgangspunkt i det som er inne i IDMegler-delen av arkitekturen.

Klasser: Autorisering, Tjenestegrensesnitt, IKT-tjenester, Nøkkelkobling, Brukslogg

f) De viktigste metodene vil bl.a være (Med fokus på bruker / bedriftseksponerte APIer):

// Kun tjenester kan benytte disse
[PrivatizedUserID, ServiceSessionToken] signIn(ServiceIdentifier)
Brukt når tjenester redirecter brukere til IDmegler sign-in page. Funksjonen tar ikke inn noe mer enn serviceIdentifieren, slik at man vet hvor man skal returnere. Tjenesten skal ikke vite noe mer brukerdata enn den trenger. Brukeren får ikke tilgang til verken PrivatizedUserID eller ServiceSessionToken.

[FilteredUserData] findPerson(PrivatizedUserID, ServiceSessionToken)
Denne leverer begrenset brukerdata, kun hvis den anonymiserte brukerIDen har en match for en spesifikk ServiceIdentifier (som kan utledes fra ServiceSessionTokenet), samt at SessionTokenet viser at den som spør faktisk er tjenesten med denne ServiceIdentifieren (unngå at andre tjenester stjeler data)

// Kun brukere kan benytte disse

[UserSessionToken] signIn() // Brukt for Brukere
[AccessLogs] getAccessLog(UserSessionToken)
[Documents] getAggregateUserDocuments(UserSessionToken)
[Boolean] setAccessRights(UserSessionToken, ServiceIdentifier)

Disse kan være spesifisert i varierende grad, jo sikrere de er, jo bedre!
Et sekvensdiagram som inneholder de viktigste metodene er også et godt svar.

Andre metoder vil være knyttet til virksomhetene som er «medlem» av systemet:

RegistrerAktør, RegistrerTjeneste, VisRegistrerteTjenester, DeaktiverTjeneste, DeaktiverAktørPerson, etc.

Oppgave 3: Testing (20%)

Ta utgangspunkt i IdMegler

a) Beskriv de to viktigste bruksscenariene for IdMegler.

Her forventes det at studenten skal liste opp de to viktigste bruksscenarioene og beskrive (ganske detaljert) hvorfor de er så viktige.

De viktigste scenariene kan være
- Bruker ønsker å logge inn, se hvilke data som er tilgjengelig for tjenester, og endre synlighet / revoke adgang til tjenester.
- Tjeneste ønsker data fra bruker. Tjenesten sender bruker til IDmegler som autentiserer seg, og kan så bruke det resulterende tokenet til å gjøre oppslag etter personlig data hos IDmegler.

b) Med utgangspunkt i scenariene definert i oppgave 3a) spesifiser de viktigste funksjonene i systemet som skal testes.

Her forventets det at studenten ikke bare lister opp funksjonene, men også beskriver hvorfor de er viktige. F.eks Denne funksjonen tester oppfyllelsen av krav X, som tester at…

c) **Definer minst fire Test Cases for Unit Testing for metoden findPerson(Id) i en av klassene fra IdMegler.**

Her bør man ha med tester for alle edgecases av input. Både gyldig, ugyldig, og varierende grad av gyldighet.

Mulige edge-cases:
- Token har ingen mapping til ekte brukerID i databasene til IDmegler (Brukeren finnes ikke)
- Token har ekte mapping til brukerID
- Tjeneste prøver å bruke et token som brukeren har trukket tilbake (fjernet adgang til tjenesten)
- Riktig token, men nøkkelen til bedriften (Den som garanterer at det faktisk er bedriften som gjør spørringen) er feil. F.ex at noen har tokenet nøkkelen fra Skatteetaten.

Her antas det at ID er den anonymiserte bruker-IDen (tokenet) tjenester får utdelt etter sukksessfull autentisering. Dette kan også tolkes som at findPerson er metoden IDmegler bruker for å gjøre oppslag på personnummer.

Funksjon	**Input**	**Pre-betingelse**	**Forventet output**
findPerson	“DetteErIkkeGyldingBrukerID”	Bedriften er autentisert mot IDmeger	En feilmelding om at brukeren man prøver å søke på ikke finnes

findPerson	“DetteErFaktiskGyldigBrukerID”	Bedriften er autentisert mot IDmeger, og har fått et token de kan gjøre oppslag med.	Den dataten brukeren har valgt å eksponere til tjenesten. I tilfelle tannlegekontor, så vil de få tilgang til alt det de trenger.
findPerson	“DetteErFaktiskGyldigBrukerIDOgså”	Bedriften er autentisert mot IDmegler, men bruker har i ettertid fjernet bedriftens rettigheter.	En feilmelding om at denne tjenesten ikke lengre har tilgang til brukerdata fra vedkommende.
findPerson	“DetteErEnStjåletNøkkelSomErGyildigBrukerID”	Bedriften prøver å autentisere seg mot IDmegler, men med feil autentiseringsnøkkel (Den som viser at det faktisk er bedriften)	Gir en feilmelding om at bedriften ikke har adgang til å gjøre et søk på brukerdataene.

d) Beskriv hvordan man kan utvikle tester for å undersøke om systemet tilfredsstiller krav til pålitelig.

Security: Lage tester som prøver å benytte løsningen uten pålogging, innbruddstester, OWASP-TOP 10

Reliabilitet: Lage tester med tanke på å nå informasjon om personer med forskjellige ID-kombinasjoner, autorisasjonskombinasjoner, autentiseringskombinasjoner.

Availability er et viktig punkt under det å teste påliteligheten til et system (oppetid, feildetektering og automatisk recovery/handover). Hvordan man kan teste systemets ytelse under stort press (Teste lastbalansering, caching etc).

Safety: Tilby alternative løsninger og tjenere, innside-cache hos virksomhet

Løsninger kan beskrive hvordan man f.eks. kan bruke:
- Unit- og system testing:
  - Black box tester (testing av krav)
  - White box tester (testing av krav med tilgang på kode)

Oppgave 4: Prosjektledelse (25%)

IdMegler skal utvikles i et prosjekt og tilbys som en tjeneste av selskapet IdMegler AS. IdMegler AS er i oppstartsfasen og må derfor etablere et prosjektteam av innleide studenter samt de interne ressursene Line (stillingsandel 80%), Pål (stillingsandel 40%), Oskar (stillingsandel 100%) og Helene (stillingsandel 100%). Prosjektet er planlagt utviklet i løpet av 10 uker for en integrasjonstest med henholdsvis Folkeregisteret og to studentsystemer ved NTNU. Alle ressurser jobber en normal arbeidsuke på 37,5 timer. Anta at det ikke er behov for opplæring før prosjektstart.

Du er innleid som prosjektleder (PL) for prosjektet.

a) Definer en grov prosjektplan for utvikling av IdMegler. I prosjektplanen må det defineres hovedaktiviteter, milepæler og avhengigheter. Skisser hvordan prosjektet eventuelt kan deles inn i delprosjekter.

Her bør det diskuteres hvilken utviklingsmetodologi som skal brukes, da dette vil påvirke en framstilling av milepæler i prosjektet. Både scrum og vannfall er alternativer, men det bør argumenters hvorfor.

Her legges det opp til en modell med hyppige leveranser til en tiltenkt produkt manager. Backend og frontend kan i stor grad utvikles separat. Tjenesten har fire hovedgrensesnitt utad: Virksomhet, Folkeregister, Privatperson, Pålogging.
Virksomhetsgrensesnitt og funksjonalitet koblet mot privatperson kan separeres som delprosjekter. Databasedesignet vil være sentral i løsningen og andre tjenester må ha veldefinerte grensesnitt mot denne. Disse grensesnittene bør designes tidlig slik at andre komponenter kan utvikles.

Hovedfaser
- Planlegging og risikostyring
- Kravspesifisering (Gjennomførbarhet, analyse, spesifisering, validering)
- Applikasjonsdesign (Arkitektur, Brukergrensesnitt, Komponenter, Database)
- Implementering
- Validering (Akseptansetesting, systemtesting etc)

Innsikt i hvordan disse delene er sammenkoblet er fint.

Eksempel på aktiviteter:
- Grensesnitt mot Folkeregister
- Grensesnitt mot innloggingsløsning for virksomhet
- Grensesnitt mot innloggingsløsning BankID, IdPorten, MinID, etc.
- Grensesnitt mot database for registrering og forvaltning av virksomhet, tjeneste
- Grensesnitt mot nøkkelmeglerkomponent
- Grensesnitt mot logg (som kan ligge i databasen)

Eksempel på leveransemilepæler:
- Milepæl 1
  - Fungerende MinID autentisering
  - Anonymiseringstjeneste for IDer samt utdeling av authentication keys for bedrifter.
  - Fungerende innhenting av data fra Folkeregisteret (uten adgangsbegrensning).

- Milepæl 2
  - Fungerende adgangsbegrensning for ressurser fra Folkeregisteret.
    - Testbrukere kan begrense tilgang, og testbedrifter kan få ut begrenset informasjon.
  - Logging av adganger til persondata.

- Milepæl 3
  - Samling av aggregatdata fra andre tjenester fungerer.
  - Aggregatdata presenteres fint på nettstedet.

- Milepæl 4
  - Nettsted så godt som ferdigutviklet
  - Brukere kan nå benytte nettsiden til å gi adgang / begrense / se personlig informasjon
  - Bedrifter kan sende brukere til IDmegler, og få anonymiserte ID'er og authentication keys tilbake.

b) Identifiser risikoer som IdMegler AS har i forhold til IdMegler-prosjektet. Anslå og begrunn sannsynlighet og konsekvenser for hver risiko. Beskriv strategier for å minske de to største risikoene (høyest sannsynlighet og størst konsekvens). Hvilke risikoer øker med et økende antall personer i prosjektet?

Her forventes det at studenten lager en risikotabell hvor det er listet opp noen av de største risikoene (2 stk). Deretter skal det beskrives med tekst hva slags strategier man kan bruke for å minske risikoen til disse og hvilke risikoer som øker med økt antall personer i prosjektet.

Disse kan inneholde:
- At utviklere dropper av teamet i kritiske faser.
- Kravendringer
- Hvis minID og bankID går ned så er systemet ikke tilgjengelig
- Hvis det er sårbarheter i off-the-shelves(programvare vi ikke har laget selv, som minID og bankID) tilleggsprogramvare så har vi ikke kontroll over det.
- Budsjettkutt eller budsjettoverskridelser
- Feilestimering
- Umoden teknologi

Risiko	**Beskrivelse**	**Sannsynlighet (Lav, Medium, høy)**	**Påvirkning (Lav, Medium, høy)**	**Påvirker (Sansynlighet*Påvirkning) (Lav, moderat, høy)**	**Plan for å unngå risiko**	**Plan for å minske påvirkning**

c) For hver aktivitet i a), anslå varigheten i dager, innsats (dagsverk), avhengigheter, og ressursfordeling. Hver person kan jobbe på en aktivitet en prosentandel av sin tid. Angi innleide studenter som S1, S2, S3 osv. Benytt de to tabellene på de neste sidene (behold den ene selv).

En god løsning vil:

- Estimere høyere på aktiviteter som oppleves å ha en høyere risiko en andre. En god tommelfingerregel er at man først estimerer hvor mange timer man tror en fase vil ta, og så legge inn slakk koblet til estimeringsrisiko (avhengig av erfaring på personell, prosjektleder og kunnskap om problemområdet).
- Planlegge i dagsverk fremfor timeverk. Aktiviteter som varer lengre enn 2-3 dager tenderer til å ha høyere risiko for feilestimering.

-Man har 3,2 personer over 10 uker (50 dager pr. person), noe som gir ca 160 dagsverk fra fast ansatte. For et så stort offentlig prosjekt vil det ikke være mulig å ferdigstille det kun med de delegerte ansatte på 10 uker.
- Flere deler av prosjektet kan utføres på en gang (med begrensning på totalt antall tilgjengelige personer på en gang)

Timeestimat:
Prosjektledelse: 10 uker
T0, T1, T2: Planlegging, analyse og kravspesifikasjon: 1 uke med 3,2 personer (interne ressurser)
T3: Overordnet design/arkitektur: 1 uke med 3,2 personer (man kan argumentere for at studentene som skal leies inn, deltar i dette arbeidet).
T4-T7 med mulige underaktiviteter: Utvikling: 4 milepæler med 5 personer på hver, med 2 uker for hver milepæl
T8-T10: Integrering med andre tjenester: 2 uker med 4 personer.
T11: Akseptansetesting og leveranse: 2 uker med 4 personer

Eksempel på et estimat: 3,2 personuker + 3.2 personuker + 40 personuker + 8 personuker + 8 personuker = 62 personuker + prosjektledelse 10 uker.

Det kan benyttes estimeringsteknikker som fra COCOMO 2 ifht estimeringsrisiko i fht organisasjon, erfaring, teknologimodenhet osv. Med høy risiko benytter noen faste variable for en organisasjon, f.eks. 2-gangen/e/PI-approksimering) = 100 personuker, dvs 500 dagsverk.

IdMegler er en ny bedrift og vil dermed ikke ha slike data tilgjengelig. Det vil imidlertid benyttes skjønn i planleggingen basert på erfaringen til den enkelte medarbeider. Antakelsen om at det ikke vil være behov for opplæring av studenter, gjør at det forventes at hver student vil være like produktiv og ha tilstrekkelig kunnskap til å gjennomføre sin jobb på normert/planlagt tid.

Gitt et estimat på 100 personuker, så vil det være behov for å dekke inn 100-32 ukeverk (T0-T2) = 68 ukeverk fra uke 3-10, etter T1-T3 er gjennomført.

T4- T11: Dette betyr 8 uker tilgjengelig for utvikling og dermed behov for ca 9 personer per uke (fra uke 3) hvis alt skal bli ferdig.
Det er 3.2 faste utviklingspersoner i snitt per uke, altså må man leie inn ca ~6 fulltidsstudenter for å få det ferdig innen sommeren er over.

d) Hva er totalbudsjettet i timer for prosjektet? Hvor mange studenter må leies inn for å levere løsningen på 10 uker? Er det en realistisk tidsplan å levere løsningen på ti uker? Begrunn hvorfor eller hvorfor ikke.

Timeverk gitt 100 ukeverk + 10 prosjektledelse: 3750 + 375 timer

Innleide studenter: ca 6 fra uke 3-10, men spesielt stort behov i utviklings/testperioden.

Her bør det argumenteres godt hvorfor resultatet man har fått er realistisk eller ikke. Innsikt i at integreringsprosesser og kravendringer drastisk kan påvirke utviklingstiden. Det bør argumenteres for at testperioden sannsynligvis blir altfor kort. Prosjektteamet bør være veldig velsmurt og kunne være produktiv fra første stund. Dvs. planen må være detaljert og oppgavene veldefinert når de forskjellige delene av systemet skal implementeres. Arkitektur og det overordnede designet vil være ekstremt viktig for å kunne lage en burndown – kart, og at kritiske komponenter blir laget tidlig. Teknologivalg

for utvikling, tilgang på eksisterende komponenter, og tilgjengelig utstyr kan påvirke tiden betraktelig – både positivt og negativt. Utviklingsprosessen som benyttes kan også påvirke tidsplanen. Det vil være ekstremt lite slakk i tidsplanen og avhengigheter mellom aktiviteter kan medføre at forsinkelser i en aktivitet direkte vil påvirke sluttdato.

INNLEVERING

Aktivitetsnr.	Aktivitet	start (uke)	ferdig (uke)	varighet (dager)	innsats (dagsverk)	avhengigheter	ressurser
t0	prosjektledelse	1					PL (100%)
t1							
t2							
t3							
total (dager)							

Budsjett							