

STYRINGS DOKUMENTER

STUDENTEVALUERINGSSYSTEM



Forord

Dette er en samling av alle styringsdokumentene gjennom prosjekt perioden. Styringsdokumentene er satt opp i rekkefølge i forhold til perioden de ble levert inn. Dokumentene ble levert inn på følgende datoer:

Dokumenter:

- | | |
|--------------------|------------|
| 1. Statusrapport | 30.10.2009 |
| 2. Prosjektskisse | 04.12.2009 |
| 3. Forprosjekt | 29.01.2010 |
| 4. Prosjektrapport | 31.05.2010 |

Innholdsfortegnelse

Forord	2
1 Statusrapport	4
2 Prosjektskisse	5
3 Forprosjektrapport	6
3.1 Presentasjon	6
3.2 Sammendrag	6
3.3 Om bedriften	6
3.4 Dagens situasjon	7
3.5 Mål og rammebetingelser	8
3.6 Løsninger	8
3.7 Analyse av løsninger	9
3.8 Konklusjon	9
4 Arbeidsplan	10
5 Fremdriftsplan	13
6 Kravspesifikasjon	15
6.1 Innledning	15
6.1.1 Innledning	15
6.3 Om bedriften	15
6.4 Bakgrunn for prosjektet	15
6.5 Forord	17
6.5.1 Forord	17
6.6 Systemkrav	18
6.6.1 Funksjonskrav	18
6.6.7 Datalagring	19
6.8 Krav Til Design	19
6.9 Dokumentasjon	20

1 Statusrapport

Leverert 30.10.2009

Gruppen John og Sara's hovedprosjekt 2010 (<http://project-s2s.blogspot.com/>) har bestemt seg for å jobbe sammen på hovedprosjektoppgaven. Vi har ikke til nå fastslått en prosjektoppgave som det er lagt frem en skisse av, men vi har kontaktet oppdragsgiver og venter på svar.

Oppdragsgiver vi har kontaktet:

HiO, Thor E.Hasle Visning av romfordeling

HiO, Thor E.Hasle Peer Evaluation

HiO

Visning av romfordeling: Ingeniørutdanningens ledelse har ytret ønske om å etablere skjermer i hver etasje som viser hvilke rom som brukes av hvem når. Det må da utvikles en applikasjon som leser data fra TimeEdit og som viser disse på en hensiktsmessig måte på en eller flere skjerm per etasje.

Hva ønsker vi:

Vi ser på disse prosjektene med hensyn til hvilke fordeler vi får ved å velge en oppgave i forhold til en annen. Vi legger stor vekt på veiledning fra oppdragsgiverens side.

Vi har enda ikke hatt møte med kontaktpersonen i HiO men vi har sendt en e-post med søknad hvor vi har uttrykket vår sterke ønske om å jobbe med deres emne i hovedprosjektet. Vi håper også å komme i kontakt med flere bedrifter etter hvert.

2 Prosjektskisse

Sted og dato: Oslo, 4. desember 2009.

Tittel: Studentevalueringssystem

Gruppemedlemmer: John Abrahamsen, Sara Khelifi

Oppdragsgiver: Høgskolen I Oslo Pb 4 St. Olavs plass, 0130 Oslo, tlf. 22 45 20 00 E-post: tor.krattebol@iu.hio.no

Kontaktperson: Thor E. Hasle Epost: thor.hasle@iu.hio.no Tlf: 22 45 32 77

Bedriften:

Ingeniørutdanningens ledelse ønsker å forbedre mulighetene for Peer Evaluation (klassefelle vurdering/det at studentene evaluerer hverandres arbeid). For å kunne gjøre slike evalueringer effektive trengs datateknisk utstyr. En mulighet kan være å videreutvikle det allerede eksisterende Mefisto. En annen mulighet er å kjøpe inn ferdig programvare. En siste mulighet er å utvikle noe selv.

Oppgaven går ut på å vurdere mulige dataløsninger og evt. implementere disse.

Prosjektet:

HiO benytter seg ikke av det eksisterende systemet Mefisto, en av årsakene er at brukergrensesnittet er svært dårlig og at det ikke er implementert i fronter gjør det også litt vanskelig. Det er hovedsakelig tre viktige oppgaver som skal løses, en av arbeidsoppgavene vi skal ta for oss er å vurdere om det eksisterende systemet kan brukes eller modifiseres. Den andre oppgaven er å vurdere muligheten for kommersielle systemer. Den siste oppgaven er å vinkle prosjektet mot å utvikle en egen løsning som kan tilfredstille oppdragsgiverens krav. Vi har hovedsakelig fokusert på siste oppgave.

3 Forprosjektrapport

3.1 Presentasjon

Tittel: Student evalueringssystem, Student-to-student (S2S)

Oppgave: Utvikle et evalueringssystem for HiO (Høgskolen i Oslo), der studenter evaluerer hverandre.

Periode: 5. januar til 29. Mai

Gruppemedlemmer: Sara Khelifi (ANV), John Abrahamsen (ANV)

Prosjektgruppe: 41

Veileder: Thor E.Hasle/Emine Göcmenoglu

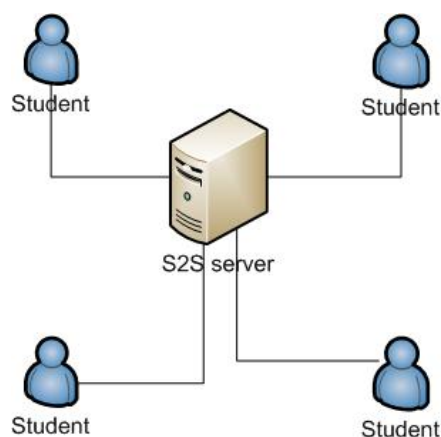
Oppdragsgiver: Høgskolen i Oslo

Kontaktperson: Thor E.Hasle

3.2 Sammendrag

Prosjektet skal gjennomføres som hovedprosjekt for HiO. Prosjektet undersøker om det kan utvikles et evalueringssystem for HiO der studenter evaluerer hverandres oppgaver. Applikasjonen skal være nettbasert, åpen kildekode og fri for bruk.

Hva er Peer review? Peer review går ut på at elever vurderer hverandres oppgaver. Peer review bærer likheter til prinsippet i peer-to-peer, som er klient til klient fildeling på nett, peer review er det samme bare man deler arbeid med retting av skolearbeid og ikke filer. I teorien skal et slikt system lette mengden arbeid som studentassistenter og lærer må gjøre når de skal vurdere elevene sine, slik at man frigjør ressurser. Vi ser for oss en web-løsning uten begrensninger til hvilken platform man måtte bruke.



Illustrasjonsfigur for vårt system.

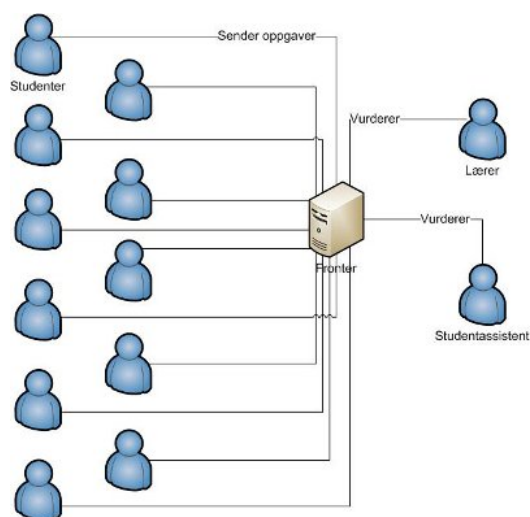
3.3 Om bedriften

Høgskolen i Oslo er landets største statlige høgskole med 12 000 studenter og 1250 tilsatte. HiO har sju avdelinger med 33 bachelorstudier og 18 masterstudier. En så stor

utdanningsinstitusjon med så mange studenter krever mye av lærere og studentassistenter så et slikt evalueringssystem vil definitivt være med å lette de ansattes hverdag på en positiv måte.

3.4 Dagens situasjon

I dag er det veldig ressurskrevende å få gjort vurdering av elevarbeid. Studenter benytter seg av skolens student portal, Fronter, for å levere inn arbeid for vurdering. Ingenting er automatisert, så det er lærere og studentassistenter som får oppgaven med å gå inn på Fronter å vurdere alt arbeid gjort av studenter. Karakterer eller bestått/ikke bestått settes inn sammen med kommentarer på Fronter så studenter kan lese tilbakemeldingen. Fronter er et bra system, men det har ingen funksjon for å muliggjøre "Peer review" som vi beskriver det. Med tanke på hvor mange studenter som studerer på HiO kan man tenke seg til at det er mye ressurser som kan frigjøres ved å kunne designe ett slikt system.



Dagens situasjon illustrert

Skolen har selv ytret ønske om å kunne lette arbeidsmengden ved å la studenter være delaktig i vurdering av hverandres arbeid.

Ingeniøravdelingen har utviklet ett system, kalt Mefisto, som har blitt brukt i mindre grad. Etter samtaler med en av utviklerne kommer det frem at systemet var for vanskelig og tidskrevende å benytte seg av når det ble lansert. Bruken av systemet ble for teknisk avansert for andre enn de som hadde vært med å utviklet det. Systemet har ikke noe grensesnitt og måtte kodes om for hver gang det skulle brukes til noe. Forøvrig var peer review bare en av funksjonene i Mefisto, og ikke den som var "mest utviklet" av de alle. En av ønskene til oppdragsgiveren er at vi skal vurdere om

Mefisto kan videreutvikles eller brukes som utgangspunkt for utvikling av en ny applikasjon. Mefisto ble bare brukt i IU-avdelingen på HiO.

3.5 Mål og rammebetingelser

De sentrale mål for systemet er hovedsakelig det som kreves for at systemet skal fungere riktig i forhold til miljøet og arbeidsplassen.

Systemet bør:

- gjøre det mulig for studenter å vurdere hverandres arbeid.
- gi vurderinger som er troverdige og kan benyttes som at de er riktige, dvs. at det ikke skal være mulighet for fusk.
- være trygt nok til at man ikke kan bryte seg inn å legge inn falske resultater.
- holde orden på hvem som har tilgang til å rette oppgaver og at den som har rettet en oppgave ikke har rett til å rette samme oppgave igjen.
- opprettholde personvernet til den enkelte.
- være 100% nettbasert.
- være åpen kildekode og fritt tilgjengelig.

Funksjonalitet som kan implementeres er:

- varsel på email/SMS når du skal evaluere en oppgave, eller har mottatt resultater.

Gruppen står fritt til å velge hvilket språk vi vil benytte for å programmere applikasjonen. Vi står også fritt til å velge hvilken grad applikasjonen skal være ferdig utviklet. I utgangspunktet tenker vi i arbeidsgruppen at vi må modellere/beskrive systemet 100% og minst lage en proof-of-concept applikasjon. Gruppen planlegger å bruke PHP som programmeringspråk og MySQL som database verktøy, ettersom det er disse verktøyene vi har mest erfaring med når det gjelder programmering av web-løsninger. Uansett kan man jo tilpasse applikasjonen relativt enkelt til å benytte annen database programvare.

3.6 Løsninger

Når det gjelder denne løsningen har vi flere ting å vurdere. Vi kan benytte oss av en ren web-applikasjon, eller så kan vi lage en "stand-alone" applikasjon. Eller en hybrid hvor de som skal administrere kan benytte en applikasjon. Ett problem som ofte oppstår da er at man ønsker at applikasjonen skal fungere på tvers av forskjellige plattformer, og man må så lage flere versjoner. Det mest hensiktsmessige vil derfor å lage en 100% web-basert løsning, fordelen med dette er jo det at man har hele systemet tilgjengelig når som helst og hvor som helst. I tillegg er det enkelte språk vi er bedre kjent med enn andre, og som egner seg godt til å løse oppgaven. Derfor ser vi for oss per i dag at vi kommer til å bruke følgende verktøy/teknologier:

Teknologi: PHP Hypertext Preprocessor

Utviklingsprogram: Coda (for Mac), Dreamweaver, MySQLWorkbench, RazorSQL, GIMP/Adobe Photoshop (grensesnitt).

Database: MySQL (fri programvare)

Programmeringspråk: XHTML, CSS, PHP, SQL, Javascript

Miljø for testing: Apache, PHP og MySQL driftes på en virtuell Debian Linux maskin fra HiO.

Programvare for modellering og planlegging: Microsoft Visio 2007, Microsoft Project Professional 2007.

3.7 Analyse av løsninger

Vi har valgt å benytte PHP for å lage applikasjonen vår fordi vi har god kunnskap til dette språket og det er veldig utbredt, samt at det er enkelt og greit å programmere med. Det krever heller ingen spesiell programvare for å kodes med. Vi skal bruke Coda for å programmere PHPen vår. MySQL setninger vil bli konstruert i MySQLWorkbench, som også er fritt tilgjengelig. Grafikken vil gjøres i GIMP eller Photoshop. Testing og produksjon av applikasjonen vil gjøres på en virtuell maskin på skolen. Tilgjengelig kun for oss.

3.8 Konklusjon

Målet med prosjektet våres er å:

- Undersøke om peer review er praktisk gjennomførbart.
- Forenkle hverdagen for de som jobber som lærere og studentassistenter ved å minske "plankekjøringsarbeid".
- Frigjøre ressurser som kan brukes på andre og mer nyttige saker.
- Produsere ett open-source bidrag for studentevaluering.

Task Name	Duration	Start	Finish
1 statusrapport	1 day?	Fri 10/30/09	Fri 10/30/09
2 prosjektskisse	1 day?	Fri 12/4/09	Fri 12/4/09
3 Påmelding av gruppe	1 day?	Mon 12/7/09	Mon 12/7/09
4 utforming av webside	1 day?	Mon 1/4/10	Mon 1/4/10
5 brainstorming	1 day	Thu 1/7/10	Thu 1/7/10
6 Veiledningstime	1 day	Mon 1/11/10	Mon 1/11/10
7 Møfesto møte	1 day	Wed 1/13/10	Wed 1/13/10
8 Veiledningstime	1 day	Mon 1/18/10	Mon 1/18/10
9 Nedlasting av programvare	1 day?	Mon 1/25/10	Mon 1/25/10
10 Forprosjekt arbeid	9 days	Mon 1/18/10	Thu 1/28/10
11 Forprosjekt innlevering	1 day	Fri 1/29/10	Fri 1/29/10
12 Datainnsamling/intervjuer	7 days?	Thu 1/28/10	Fri 2/5/10
13 Dataanalyse/seminarsamling	6 days	Thu 2/4/10	Thu 2/11/10
14 Krevsesifikasjon	4 days	Fri 2/12/10	Tue 2/16/10
15 Møte med Mark	1 day?	Thu 2/18/10	Thu 2/18/10
16 use case modellering	1 day?	Fri 2/19/10	Fri 2/19/10
17 Klasse diagram	4 days?	Mon 2/22/10	Thu 2/25/10
18 ER modellering	4 days?	Thu 2/25/10	Tue 3/2/10
19 Databasesystem	12 days?	Wed 3/3/10	Thu 3/18/10
20 Funksjonalitet Brukergren	47 days?	Thu 3/18/10	Fri 5/12/10
21 Testrapport	8 days?	Tue 4/27/10	Thu 5/6/10
22 Testing	3 days?	Wed 4/21/10	Fri 4/23/10
23 Brukermanual	0 days	Wed 5/12/10	Wed 5/12/10
24 Innlevering av Prosjektapp	1 day?	Mon 5/31/10	Mon 5/31/10
25 Arbeid med presentasjon	10 days?	Fri 6/4/10	Thu 6/17/10
26			
27			
28			
29			
30			
31	1 day?	Mon 1/25/10	Mon 1/25/10

Ready

start

Microsoft Project - Ar...

Windows Print Screen...

On-Screen Keyboard

untl

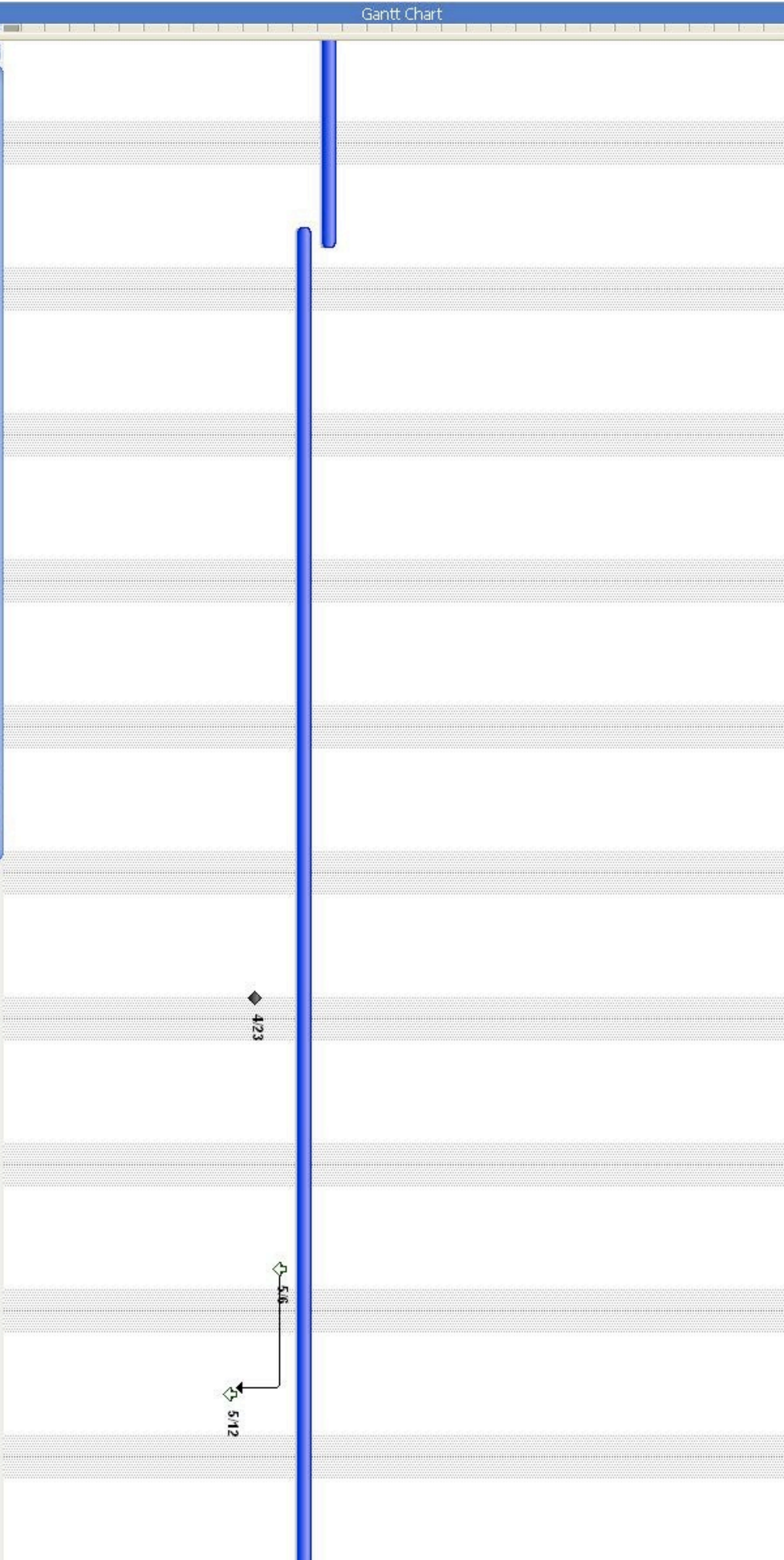
esc

On-Screen Keyboard

File Keyboard Settings Help

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 psc slk blk

10	Mar 14, '10	Mar 21, '10	Mar 28, '10	Apr 4, '10	Apr 11, '10	Apr 18, '10	Apr 25, '10	May 2, '10	May 9, '10	May 16, '10													
T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T



On-Screen Keyboard

File Keyboard Settings Help

esc F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 psc slk bk

MO

5 Fremdriftsplan

6 Kravspesifikasjon

6.1 Innledning

6.1.1 Innledning

Vi har kommet frem til at vi vil utvikle vår egen applikasjon for å løse oppdraget. Applikasjonen ble døpt s2s "Student to Student". Navnet kommer fra peer to peer fildeling, men at man deler oppgaver og ikke filer.

Prosjektet skal gjennomføres som hovedprosjekt for HiO. Prosjektet går ut på å utvikle et evalueringssystem for HiO der studenter evaluerer hverandres oppgaver. Applikasjonen skal være nettbasert, åpen kildekode og fri for bruk.

6.3 Om bedriften

Høgskolen i Oslo er landets største statlige høgskole med 12 000 studenter og 1250 tilsatte. HiO har sju avdelinger med 33 bachelorstudier og 18 masterstudier. En så stor utdanningsinstitusjon med så mange studenter krever mye av lærere og studentassistenter så et slikt evalueringssystem vil definitivt være med å lette de ansattes hverdag på en positiv måte.

6.4 Bakgrunn for prosjektet

I dag er det som påpekt over veldig ressurskrevende å få gjort vurdering av elevarbeid. Studenter benytter seg av skolens student portal, Fronter, for å levere inn arbeid for vurdering. Ingenting er automatisert, så det er lærere og studentassistenter som får oppgaven med å gå inn på Fronter å vurdere alt arbeid gjort av studenter. Karakterer eller bestått/ikke bestått settes inn sammen med kommentarer på Fronter så studenter kan lese tilbakemeldingen. Fronter er et bra system, men det har ingen funksjon for å muliggjøre "peer review" som vi beskriver det. Med tanke på hvor mange studenter som studerer på HiO kan man tenke seg til at det er mye ressurser som kan frigjøres ved å kunne designe ett slikt system. Skolen har selv ytret ønske om å kunne lette arbeidsmengden ved å la studenter være delaktig i vurdering av hverandres arbeid.

Ingeniøravdelingen har utviklet ett system, kalt Mefisto, som har blitt brukt i mindre grad. Etter samtaler med samtlige utviklere av Mefisto kommer det frem at systemet var for vanskelig og tidskrevende å benytte seg av når det ble lansert. Bruken av systemet ble for teknisk avansert for andre enn de som hadde vært med å utviklet det. Systemet har ikke noe grensesnitt og måtte kodes om for hver gang det skulle brukes til noe. Forøvrig var peer review bare en av funksjonene i Mefisto, og ikke den som

var "mest utviklet" av de alle. En av ønskene til oppdragsgiveren er at vi skal vurdere om Mefisto kan videreutvikles eller brukes som utgangspunkt for utvikling av en ny applikasjon. Mefisto ble bare brukt i IU-avdelingen på HiO.

6.5 Forord

6.5.1 Forord

Denne kravspesifikasjonen beskriver betingelsene for ”student to student”.
Krav til funksjonalitet og rammebetingelser er beskrevet i dette dokumentet.
Hovedkravene om funksjonalitet er gitt av Oppdragsgivern, mens hvordan det spesifikt og best mulig skal løses har gruppa styrt selv.

6.6 Systemkrav

6.6.1 Funksjonskrav

6.6.2 Krav til evalueringssystem

1. Studenter skal kunne rette hverandres arbeid.
2. Studenter skal kunne gi en vurdering og karakter.
3. Systemet bør holde orden på hvem som har tilgang til å rette oppgaver og at den som har rettet en oppgave ikke har rett til å rette samme oppgave igjen.
4. Være nettbasert.
Alt gjøres på nettet og ingen klient programvare kreves.
5. Ha en beskyttelse mot utryggheter som sql injections, ved bruk av SSL.
motta varsel på e-post når en oblig er rettet.

6.6.3 Krav til Administrasjon

1. Logge inn/ut
Admin skal kunne logge inn og logge ut.
2. Legge til bruker
3. Tildel kurs til lærer
4. Tildel kurs til studentassistent
5. Legge til kurs
6. export/import
Import av elevdata fra student databsen.
4. Slette bruker

6.6.4 Krav til lærer/studentassistent

1. Logge inn/ut
Studentassistenter og lærere skal kunne logge inn.
2. Velge evalueringstype for oppgave
godkjent/ikke godkjent eller karakter
3. Last opp oppgaver
4. Åpne den tildelte oppgaven
5. Se resultater.

6.6.5 Krav til student

1. Logge inn/ut
Logge inn ved bruk av studentnummer og passord som man har fått tildelt av skolen.
2. Last opp egen oppgave

- Sende inn oppgave
Oppgaven blir sendt til S2S databasen.
3. Se resultateter
 1. Åpne den tildelte oppgaven.
 2. Vurdere andres arbeid, legge inn kommentar og karakter/godkjent ikke godkjent.

6.6.6 Tekniske krav

1. **Teknologi:** PHP Hypertext Preprocessor
2. **Utviklingsprogram:** Coda (for Mac), Dreamweaver, MySQLWorkbench, GIMP/Adobe Photoshop (grensesnitt), Razor UML
3. **Database:** MySQL (fri programvare)
4. **Programmeringspråk:** XHTML, CSS, PHP, SQL, Javascript
5. **Miljø for testing:** Apache, PHP og MySQL driftes på en virtuell Debian Linux maskin fra HiO.
6. **Programvare for modellering og planlegging:** Microsoft Visio 2007, Microsoft Project Professional 2007. Argo UML, Violet UML.

6.6.7 Datalagring

6.6.8 Krav til datalagring

1. Data som sendes/lagres i en Mysql database.
2. Det skal være krav for input validation, slik at det ikke er sårbart for sql injections.
3. Det skal være bygd opp etter normaliserings krav.
Dobbelt lagring må ikke forekomme.

6.8 Krav til design

6.8.1 Generelle design mål

1. Brukergrensesnittet skal være brukervennlig og enkelt å forstå.
2. Det skal være slik at brukere intuitivt forstår hvordan systemet skal brukes, i stedet for at opplæring eller brukermanualer trengs (men vi vil uansett lage en brukerveiledning).
3. Det skal være mest mulig brukervennlig og universelt utformet slik at de med svekket syn skal klare å navigere i siden.
4. Utskriftsvennlig
5. Mulighet for navigering rundt i siden:
Brødsmuler, Frem/tilbake/home knapper.
6. søkefunksjonalitet

7. Mulighet for Engelsk oversettelse til utlandske studenter.

6.9 Dokumentasjon

6.9.1 *Krav til dokumentasjon*

Det skal i prosjektperioden bli satt opp en nettside for gruppen, der skal det føres dagbok over hva som blir gjort, hvilke utfordringer vi møter og hvordan disse blir løst.

Det ferdige prosjektrapporten skal dokumenteres med:

- Forprosjekt
- Prosessrapport

- Produktrapport
- Kravspesifikasjon
- Testrapport
- Brukermanual

