

TDP003 Projekt: Egna datormiljön

Projektplan

Författare

Daniel Huber, danhu849@liu.se
Jens Öhrnell, jenoh242@liu.se

Innehåll

1	Revisionshistorik	2
2	Introduktion	2
3	Tekniker	2
4	Arbetsmetodik	3
5	Kravspecifikation	3
6	Riskbedömning och åtgärder	4
7	Tidsplanering vecka för vecka	4
	Vecka 37 - Planeringsdokument	4
	Vecka 38 - Installationsmanual och LOFI	5
	Vecka 39 - Projektplan och Gem. Installationsmanual	6
	Vecka 40 - Datalagret och Korrigeringar	7
	Vecka 41 - Presentationslagret	8
	Vecka 42 - Systemdokumentation, -demonstration och Portfolio publicering	8
	Vecka 43 - Reflektionsblad och Testdokumentation	9

1 Revisionshistorik

Tabell 1: Projektplanens Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.1	Korrigerat Projektplanens kompletteringar	290920
1.0	Projektplan 1:a Utkast	240920

2 Introduktion

Projektets mål utgörs av att skapa, presentera och underhålla en webbaserad portfolio. Där presenteras de projekt som ska, i och utanför universitetet, färdigställas under de kommande 3 åren. Den kompletta kravspecifikationen skrevs av programledningen och återfinns i dokumentet *Systemspecifikation av portfoliosystemet*. En sammanfattning finns i detta dokument.

För de avsnitt där det antingen är vagt eller inte alls specificerat vad eller hur något ska göras förväntas det av studenten att egna initiativ tas. Exempel på detta är utseendet på användargränssnittet där nästintill total frihet ges.

3 Tekniker

Enligt projektets kravspecifikation används följande tekniker:

- Python3
- Git
- HTML5
- CSS3
- Flask
- Jinja2
- venv
- JSON
- Latex

Pythonpaketet venv används för att möjliggöra utveckling i virtuell miljö samt minimera risken av paketkonflikter. I pythonpaketet Flask tillhandahålls debugger, möjlighet att binda python3 funktioner till URL paths (API) och webserver. Pythonpaketet Jinja2 används för HTML5 och CSS3 templates. Versionshantering sköts med git. Datalagret representeras av en JSON fil, data.json. Dokumentationen skrivs i Latex.

4 Arbetsmetodik

Arbetsuppgifter skrivs upp i repot som issues och tas an kontinuerligt. Vid det tillfälle då någon arbetsuppgift avklarats senare under veckan meddelas detta till resten av arbetsgruppen. I de fall där det inte finns fler issues att ta av, diskuteras nya fram med resten av gruppen.

För de kunskapsområden som är vaga inhämtas först information ett par dagar innan ämnet först berörs i skolan. Sedan sammanställs kravspecifikationen för den specifika aktiviteten. För kod färdigställs en tutorial hittad online av liknande storlek som aktiviteten innan aktiviteten påbörjas.

Under arbetets gång push:as arbetet kontinuerligt till projektets git-repo. Åtminstone en gång i halvtimmen eftersträvas. Aktiviteter och delmål arbetas på och avklaras huvudsakligen individuellt. Större aktiviteter kan delas upp till mindre, men det diskuteras i arbetslaget om så är nödvändigt först och skrivs upp i issues. Vid varje avklarad aktivitet kontrolleras den fullbordade aktiviteten mot kravspecifikationen och kompletteras om nödvändigt. Dagbok skrivs enskilt efter arbetsinsats av en aktivitet. Projektplanen underhålls lättare om dagboken är detaljrik.

Sms och eller mejl skickas till berörd vid påbörjan och avslut av arbete på aktiviteter då merge konflikter och onödigt dubbelarbete minimeras. Icke-akuta frågor ställs via mejl eller sms. Vid icke lösbara problem eller missförstånd kontaktas resten av arbetslaget så snart det är möjligt via telefonsamtal.

5 Kravspecifikation

Projektet utgörs av två delar. Dels det slutanvändaren kan se, det presentativa, och dels det som denne inte kan se, datalagringen och datahämtningen.

Av webbplatsen krävs det att den utrustas med fyra html-sidor skrivna i HTML5 och CSS3. En huvudsida/första sida som antingen kan vara statiskt eller dynamisk samt 3 stycken dynamiska sidor. De tre sistnämnda utgörs av en söksida, en projektsida och en tekniksida. Krav finns att det på huvudsidan visas bilder. På söksidan kan projekt sorteras efter projektens id samt efter bokstavsordning och ålder med hjälp av knappar. På respektive projektsida visas fullständig information om det specifika projektet tillsammans med större passande bilder. Om projektet inte finns visas relevant felkod status 404 'This page does not exist'. På tekniksidan visas information om projekten utifrån vilka tekniker som använts. Ett ambitionsmål är att kunna visa i hur stor omfattning teknikerna använts i projekten. Varje projekt ska i listningar på söksidan och tekniksidan visas med en liten bild bredvid sig. Bildtext måste finnas till varje bild. Vid fel ska dessa hanteras på ett, för slutanvändaren, informativt sätt så att denne kan förstå vad som gått fel.

Datalagret utgörs av JSON-kod med UTF-8 teckenkodning i JSON-filen data.json. Varje projektinstans i JSON-koden utgörs av projektnamn, projekt-id i form av ett unikt heltalsnummer, startdatum, slutdatum, kurskod, kursnamn, kurspoäng, nyttjade tekniker, sammanfattning, full beskrivning, liten och stor bild, antal gruppmedlemmar och länk till projektsida. JSON-koden manipuleras med hjälp av ett API utgörande av sex stycken standardiserade funktioner. Samtliga namn skrivs på engelska. Funktioner som ska implementeras: load(filename), get_project_count(db), get_project(db, id), search(db, sort_by='start date', sort_order='desc', techniques=None, search=None), get_techniques(db), get_technique_stats(db). För fullständig specifikation hänvisas läsaren till dokumentet *Application Programming Interface (API)* som finns på kurshemsidan.

Vid sökning av godtyckliga termer om projektets information genereras träffar och slutanvändaren presenteras med en lista där det mest förmodade objektet presenteras högst upp eller längst ner. Tekniksidan kan sorteras på använda tekniker. Det ska noteras att funktionaliteten ska möjliggöra sökning på ett ord, sortering och filtrering av tekniker. Sökningarna ska ske samtidigt. Datum är i formatet ISO 8601. Förändringar i data.json filen ska för användaren presenteras direkt utan nödvändig omstart av webbservern. En frivillighet och alltså inte ett krav är att lägga till en administrativ sida för redigering av data.

Projektet versionshanteras med git.

Vid systemets slutförande testas systemets funktioner av två personer som ej ingått i utvecklarteamet. Systemtesten och uppkomna fel dokumenteras och ska vara åtgärdade i den slutgiltiga versionen av projektet. Testerna skrivs sedan in i systemdokumentationen.

6 Riskbedömning och åtgärder

Den största risken mot att inte bli klar i tid är sjukdom som redan drabbat en del av teamet en gång när detta skrivs (23/9). Det löstes av att den andra teammedlemmen fick skynda sig att göra klart arbetet innan deadline. Till hjälp finns betyget med beröm för tidigare inlämningar. I värsta fall kan inlämning ske efter deadline så länge detta kommuniceras till handledare innan. Ambitionen är dock att detta aldrig nyttjas.

Familjemedlemmar till arbetslaget har instruerats att kontakta handledare vid skador, sjukdomar eller andra traumatiska händelser av allvarlig sort. Detta för att resurser från universitetet i den mån det är möjligt kan tillsättas tidigt.

Beroende på tidsåtgången att inhämta ny kunskap kan nya flaskhalsar uppstå längre bort i tiden av saker som inte kunnat förutses då kunskap saknats. Ambitionsmål att läsa om och testa tekniker en eller två veckor innan de introduceras första gången har satts. Dessa har dock lägre prioritering än det som måste göras innan nästa deadline, men finns som en påminnelse att om tid finns så kan de användas till att underlätta framtida deadlines.

Slutligen finns risken att bli upptagen eller fastna med andra uppgifter i andra kurser. I denna projektplan täcks bara saker som måste göras för detta projekt. Risk finns att för stort fokus läggs på detta projekt så att andra kurser försenas. Åtgärder har satts in mot detta genom att möte kring personlig tidsdisponering hålls minst en gång i veckan. Översikt ges lättare om vem som kan göra vad när.

7 Tidsplanering vecka för vecka

Projektplanen uppdelas i veckor för lättare översikt. För varje vecka presenteras en mer specificerad uppskattning av tidsåtgång för aktiviteter som behöver färdigställas innan deadline. Tidsåtgången för deadline inkluderar också tiden för dess aktiviteter. För de aktiviteter som avklarats visas också den verkliga tidsåtgången samt datum för färdigställande. Varje veckas deadlines och milstolpar presenteras i tabeller. Ifall det saknas deadlines för någon vecka så saknas också tabell. Observera att med tidsåtgång så menas den sammanlagda tiden samtliga projektmedlemmar lagt ner tillsammans.

Aktiviteter fördelades ut över veckor. Specifik dag och tid för aktiviteter fastställdes ej i projektplanen då dygnsrytmen för projektgruppens medlemmar skiljs stort åt. Deadlines respekteras och god kvalitet på arbete garanteras. Detta har bevisats genom tidigare inlämnade dokument.

Vecka 37 - Planeringsdokument

Tabell 2: Tabell över deadlines V.37

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Tors 10/9	Hård deadline	Planeringsdokument	6 tim	7 tim	God	1

Tidsåtgången för planeringsdokumentet utgjordes av 7 timmar. Se tabell: 2. Av dessa ägnades:

- Tisdag 8/9:
 - 3 tim Innehåll för planeringsdokument sammanställdes.
 - 2 tim Planeringsdokumentet skrevs.

- Onsdag 9/9:
 - 30 min att att figurera ut hur latex dokumentet kompileras.
 - 1 tim att ta reda på hur en git --hard-reset reverseras.
 - 30 min att figurera ut hur tabeller skrivs i latex och få dem att positioneras rätt i texten.

Ambitions mål för veckan:

- Att planeringsdokumentet färdigställs en dag innan deadline. - Planeringsdokumentet färdigställdes den 9/9.
- Förståelse för relationen mellan Flask, Python3 och Jinja2 uppnås. - I och med att planeringsdokumentet lämnades in tidigare lades Torsdagen 10/9 på flask tutorials.

Vecka 38 - Installationsmanual och LOFI

Tabell 3: Tabell över deadlines V.38

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Tors 17/9	Hård deadline	Grundläggande Installationsmanual	4 tim	9 tim 15 min	God	1
	Hård deadline	Lofi-prototyp	6 tim	12 tim 29 min	God	1

Den Grundläggande installationsmanualen färdigställdes på 9 tim och 5 min. Se tabell: 3

- Måndag 14/9:
 - 1 tim 40 min Dokumentet skrevs.
 - 20 min Bash-script som tog bort onödiga latex filer skrevs.
 - 2 tim Förståelse uppnåddes gällande hur .tex dokument innehållandes bilder kompileras.
- Tisdag 15/9:
 - 4 tim Dokumentet skrevs.
 - 1 tim Förståelse uppnåddes gällande kommandon för kontroll av lyckad installation.
 - 15 min Dokumentet rättades och bildernas plats justerades.

Lofi-prototypen färdigställdes på 12 tim och 29 min. Se tabell: 3

- Tisdag 15/9:
 - 1 tim HTML för förstasidan skrevs.
 - 4 tim CSS för förstasidan skrevs.
- Onsdag 16/9:
 - 2 tim Utkast för samtliga sidor snabbskissades.
 - 3 tim Skisserna renskrevs för hand med linjal.
- Torsdag 17/9:
 - 2 tim 9 min De renskrivna skisserna scannades in och sammanställdes med förklaringar i ett latex dokument.

- 20 min Dokumentet rättades och bildernas plats justerades.

Ambitionsmål för veckan:

- LOFI-prototypen presenteras i HTML5 och CSS3 format. - På grund av sjukdom färdigställdes bara förstasidan.

Vecka 39 - Projektplan och Gemensam Installationsmanual

Tabell 4: Tabell över deadlines samt första milstolpen V.39

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Tors 24/9	Hård deadline	1:a Version Gemensam Installationsmanual	24 tim	9 tim 41 min	God	1
	Hård deadline	Projektplan: Första utkast	24 tim	22 tim 30 min	God	1
Sön 27/9	Milstolpe	Kunna ta det lugnt under helgerna.				

Egna insatser 1:a Versionen Gemensam installationsmanual färdigställdes på 9 timmar och 40 minuter. Se tabell: 4

- Måndag 21/9:
 - 20 min Skapa branch från master för development.
 - 5 tim 31 min Branch repo från master skapades. READMEs skrevs i markdown i både hemkatalagen och manuals katalogen. Lättförståelig katalogstruktur gjordes i ordning. Initiella svårigheter att push:a upp filer felsöktes.
- Tisdag 22/9:
 - 3 tim Flertalet issues för att få igång arbetet skapades. Felsökning av andras problem gjordes.
- Onsdag 23/9:
 - 50 min Inledningen kompletterades och andras merge konflikter löstes.

Projektplan 1:a utkast färdigställdes på 20 timmar 30 minuter. Se tabell: 4:

- Torsdag 17/9:
 - 1 tim Minst 2-3 aktiviteter definierades inför varje deadline.
- Tisdag 22/9:
 - 4 tim Aktiviteterna för varje deadline skrevs in. Projektplanens upplägg bestämdes.
- Onsdag 23/9:
 - 2 tim Anteckningar renskrevs och sorterades in i projektplanen.
 - 5 tim 30 min Tabeller och listor reformatrades.
 - 30 min Ytterligare information om tidsåtgång lades till.
 - 2 tim Riskbedömning och åtgärder skrevs in i projektplanen.
- Torsdag 24/9:

- 2 tim Tabeller formaterades om. Information om prioritering lades till.
- 2 tim Ambitions mål bestämdes och adderades till respektive vecka.
- 2 tim Strukturen rättades till och presentationslagret skrevs in under V.41. Datalagret i V.40.
- 1 tim Oegentligheter i tabeller rättades till med paketet makecell och tydligare rubriker skrevs.
- 1 tim Milstolpar skrevs in och rubriker formaterades om så att dess nummer inte skrevs ut i dokumentet.

Ambitions mål för veckan:

- Gemensam Installationsmanual färdigställdes.
- 1 tim Projektplanens struktur omgjordes till att bli mer estetiskt tilltalande.

Vecka 40 - Datalagret och Korrigeringar

Tabell 5: Tabell över deadlines samt andra milstolpen V.40

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Torsdag 1/10	Hård deadline	Bidra med icke-trivial förbättring: gem. installationsmanual / databas tester	1h		God	2
	Hård deadline	Korrigera brister: installationsmanualen	0-2 tim		God	3
	Hård deadline	Korrigera Brister: Projektplanen	0-2 tim	6 tim 13 min ht	God	2
Fredag 2/10	Hård deadline	Datalagret Godkänt	11 tim	6 timmar 10 min	God	1
Sön 4/10	Milstolpe	Kunna sätta in och ta bort projekt mha API				

Bidra med icke-trivial förbättring - installationsmanual. Se tabell: 4

- Tisdag 22/9:
 - 9 tim 41 min Branch av master och issues skapades, instruktioner skrevs i markdown i README:s, inledning kompletterades.

Datalagret Godkänt på 6 timmar och 10 minuter. Se tabell: 5

- **2 tim** **20 min** load funktionen implementerades.
- 1 tim I 30 min min get_project_count funktionen implementerades.
- 1 tim I 30 min get_project funktionen implementerades.
- 3 tim I 2 tim 40 min search funktionen implementerades.
- 2 tim I 1 tim 10 min get_techniques funktionen implementerades.
- 2 tim I 1 tim 20 min get_technique_stats funktionen implementerades.

Brister i Projektplanen korrigerades på 6 timmar och 13 minuter. Se tabell: 5

- 28/9:

- 1 tim 23 min Stycket om Arbetsmetodik skrevs till i projektplanen.
- 29/9:
 - 1 tim 43 min Projektplanens tempus skrevs om till passiv form.
 - 2 tim 52 min Referenser till tabeller lades till och korrigerades.
 - 15 min Milstolpar ändrades.
 - 30 min Datalagrets verkliga tid lades till. Se tabell: 5

Ambitionsmål för veckan:

- 1 tim Information om implementationen för en sqlite3 databas inhämtades för senare uppgradering av portfolions back-end.

Vecka 41 - Presentationslagret

Tabell 6: Tabell med tredje milstolpen V.41

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Fre 9/10	Milstolpe	Presentationslagret samverkar med datalagret			Vag	1

Presentationslagret. Se tabell: 6

- 1 tim Potentiella ändringar av LOFI:n diskuterades. Ändringar skrevs in i ett dokument i LOFI-mappen.
- 4 tim Temasida som agerar template åt alla andra sidor färdigställdes. Mer kunskap kring Jinja2, HTML5 och CSS3 inhämtades.
- 3 tim Statisk eller dynamisk sida med profilbild och bild för senaste projekt samt 'Om mig' text på /sidan gjordes i ordning.
- 5 tim Dynamisk sida som sorterar och listar projekten på /list slutfördes. Knappar som sorterar sökresultaten lades till. För varje sökresultat har mindre beskrivning och liten bild lagts till.
- 3 tim Fullständig infosida för specifikt proj med id på /project/id visas på samtliga projektsidor.
- 3 tim På /techniques visas en sammanställning av alla project baserat på använda tekniker.

Ambitionsmål för veckan:

- 5 tim Teknikers omfattningen i ett visst projekt lades till.
- 2 tim Ett bildspel innehållandes projektbilder på förstasidan lades till.

Vecka 42 - Systemdokumentation, -demonstration och Portfolio publicering

Tabell 7: Tabell över deadlines V.42

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Tors 15/10	Hård deadline	Portfolion Publicerad	-		Vag	1
	Hård deadline	Systemdemonstration	3 tim		Vag	1
	Hård deadline	1:a Versionen Systemdokumentation	9 tim 30 min		Vag	3

1:a Versionen Systemdokumentation. Se tabell: 7

- 1 tim Systemdokumentationens specifika innehåll togs fram.
- 1 tim Mindmap samt skiss för översiktsbild gjordes.
- 2 tim Sekvensdiagram ritades.
- 2 tim Dokumentation kring felhantering skrevs.
- 1 tim Beskrivning av metoder och program som används vid felsökning gjordes.
- 2 tim Ett första utkast skrevs.
- 2 tim Systemdokumentation renskrevs.
- 30 min Bilders position rättades till och systemdokumentationen renskrevs.

Portfolion Publicerad. Se tabell: 7

- 1-2 tim Potentiella buggar redde ut.

Ambitionsmål för veckan:

- 4 tim Portföljerna gjordes personliga.

Vecka 43 - Reflektionsblad och Testdokumentation

Tabell 8: Tabell över deadlines V.43

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Tors 22/10	Hård deadline	Testdokumentation inlämnad	3 tim 30 min		Vag	1
	Hård deadline	Individuellt Reflektionsblad	5 tim		Vag	1
	Hård deadline	Korrigerat event. brister i systemdokumentationen	0-4 tim		Vag	2

Testdokumentation inlämnad. Se tabell: 8

- 1 tim Daterad testlogg där samtliga testfall körts togs fram.
- 1 tim Minst ett relevant testfall per funktionellt krav dokumenteras.
- 1 tim min Testdokumentationen skrevs.
- 30 min Bilder och text rättades till.

Individuellt Reflektionsblad. Se tabell: 8

- 1 tim Relevanta delar plockades ut från dagboken.
- 1 tim Delarna skrevs ihop i ett latex dokument.
- 30 min Potentiella bilder formaterades och text rättades..

Korrigerat eventuella brister i systemdokumentationen. Se tabell: 8

- 0-4 tim Potentiella brister rättades till.

Ambitionsmål för veckan:

- 5 tim Portfolio laddades upp på egen webbserver.