

TDP003 Projekt: Egna datormiljön

Projektplan

Författare

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

Innehållsförteckning

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	6
	Vecka 39 - Projektplan och Gem. Installationsmanual	7
	Vecka 40 - Datalagret och Korrigeringar	9
	Vecka 41 - Presentationslagret	10
	Vecka 42 - Systemdokumentation, -demonstration och Portfolio publicering	10
	Vecka 43 - Reflektionsblad och Testdokumentation	11

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.1	Korrigerat Projektplanens kompletteringar	280920
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 att studenten tar egna initiativ. Exempel på detta är utseendet på användargränssnittet där nästintill total frihet ges.

3 Tekniker

För att uppfylla kravspecifikationen för detta projekt kommer följande tekniker användas:

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

Pythonpaketet venv används för att möjliggöra utveckling i virtuell miljö samt minimera risken av paketkonflikter. Pythonpaketet Flask tillhandahåller debugger, möjliggör bindandet av python3 funktioner till URL paths (API) och agerar webbserver. Pythonpaketet Jinja2 för HTML5 och CSS3 templates. Versionshantering sköts med git. En JSON fil, data.json representerar datalagret. Dokumentationen skrivs i Latex.

4 Arbetsmetodik

Varje måndag delas arbetsuppgifterna ut över arbetslaget. Mötet hålls antingen i lektionssal på universitetet eller över telefon. Vid det tillfälle då någon arbetsuppgift avklarats senare under veckan meddelas detta till resten av arbetsgruppen och en ny arbetsuppgift diskuteras fram.

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 kodfärdigställs en tutorial hittad online av liknande storlek som aktiviteten innan aktiviteten påbörjas.

Under arbetets gång push:as arbete 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. 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 en större passande bilder. Om projektet inte finns visas relevant felkod i.e. status 404 'This page does not exist'. På tekniksida 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 tekniksida 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. Tekniksida 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. Detta gör att det i värsta fall går att lämna in senare än deadline så länge detta kommuniceras till handledare innan. Ambitionen är dock att aldrig behöva nyttja detta.

Vid skador, sjukdomar eller andra traumatiska händelser av allvarlig sort har familjemedlemmar blivit instruerade att informera handledare så fort som möjligt. Detta för att resurser från universitetet i den mån det är möjligt kan sättas in tidigt.

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

Slutligen finns risken att bli upptagen eller fastna med andra uppgifter i andra kurser. Denna projektplan täcker bara saker som måste göras för detta projekt. Risk finns att för stort fokus läggs på denna kurs så att andra kurser blir försenade och stör längre in i tiden. Vi har satt in åtgärder mot detta genom att minst en gång i veckan ha möte och diskutera ens egna deadlines för veckan i alla kurser. Det gör det lättare att se 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 vara klara 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.

Då projektgruppens medlemmar har vitt skilda dygnsrytmer är det onödigt att bestämma dag och tid för samtliga aktiviteter i förväg. Det räcker med att veta inom vilken vecka aktiviteter ska vara klara. Tidigare inlämnade dokument samt denna projektplan är bevis tydliga nog för att så är fallet.

Vecka 37 - Planeringsdokument

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. Av dessa ägnades:

- Tisdag 8/9:
 - 3 tim sammanställande av vad ett planeringsdokument ska innehålla.
 - 2 tim skrivande av dokumentet.
- Onsdag 9/9:
 - 30 min åt att figurera ut hur latex dokumentet skulle 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 rätt positionerade i texten.

Ambitionsmål för veckan:

- Vara klar med Planeringsdokumentet en dag innan deadline. - Blev klara den 9/9.
- Förstå hur Flask, Python3 och Jinja2 hänger ihop. - I och med tidigare inlämning lades Torsdagen 10/9 på flask tutorials.

Vecka 38 - Installationsmanual och LOFI

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:

- Måndag 14/9:
 - 1 tim 40 min Dokumentet skrevs.
 - 20 min Bash-script som tog bort onödiga latex filer skrevs.
 - 2 tim lära sig kompilera .tex dokument innehållandes bilder samt lägga in dem och läsa på om paket.
- Tisdag 15/9:
 - 4 tim Dokumentet skrevs.
 - 1 tim att ta reda på alla kommandon som visar att installation gått som förväntat.
 - 15 min Dokumentet rättades och bildernas plats justerades.

Lofi-prototypen färdigställdes på 12 tim och 29 min:

- Tisdag 15/9:
 - 1 tim HTML Skrevs för förstasidan.
 - 4 tim CSS Skrevs för förstasidan.
- Onsdag 16/9:
 - 2 tim Utkast snabbskissades för samtliga sidor.
 - 3 tim Skissarna förbättrades och renskrevs för hand.
- 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:

- Kunna presentera LOFI-prototypen i HTML5 och CSS3 format. - På grund av sjukdom kunde bara förstasidan / färdigställas.

Vecka 39 - Projektplan och Gemensam Installationsmanual

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Tors 24/9	Hård deadline	1:a Version Gemensam Installationsmanual	24 tim	8 tim	God	1
	Hård deadline	Projektplan: Första utkast	24 tim	22 tim 30 min	God	1
	Milstolpe	Projektplan: klar				

Egna insatser 1:a Versionen Gemensam installationsmanual färdigställdes på 8 timmar:

- Måndag 21/9:
 - 20 min Skapa branch från master för development.
 - 5 tim 31 min Skriva READMEs i markdown i både hemkatalagen och manuals katalogen. Fixa lättförståelig katalogstruktur. Felsöka initieella svårigheter att push:a upp filer.
- Tisdag 22/9:
 - 3 tim skriva issues, hjälp med felsökning av andras problem och evaluera färdiga issues.
- Onsdag 23/9:
 - 50 min Kompletterat inledningen och hjälpt att lösa andras merge conflicts.

Projektplan 1:a utkast färdigställdes på 20 timmar 30 minuter:

- 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 dokumentet.
 - 5 tim 30 min Reformaterade tabeller och listor.
 - 30 min Lade till ytterligare information om tidsåtgång.
 - 2 tim Riskbedömning och åtgärder skrevs in i projektplanen.
- Torsdag 24/9:
 - 2 tim Formaterade tabeller igen. Lade till information om prioritering.
 - 2 tim Ambitions mål bestämdes och adderades till respektive vecka.
 - 2 tim Rättade till strukturen och skrev in presentationslagret i V.41 samt datalagret i V.40.
 - 1 tim Rättade till oegentligheter i tabeller med paketet makecell och skrev tydligare rubriker.
 - 1 tim Skrev dit milstolpar och såg till att rubriker fortfarande finns med i innehållsförteckningen, men att dess nummer inte skrivs ut i dokumentet.

Ambitions mål för veckan:

- Bli helt klar med Gemensam Installationsmanual.

- 1 tim Göra projektplanens struktur estetiskt tilltalande.

Vecka 40 - Datalagret och Korrigeringar

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
Tors 1/10	Hård deadline	Bidra med icke-trivial förbättring: gem. installationsmanual / databas tester	0-2 tim		God	2
	Hård deadline	Korrigera brister i installationsmanualen	0-2 tim		God	3
	Hård deadline	Korrigera Brister i Projektplanen	0-2 tim	1 tim 23 min hittills	God	2
	Milstolpe	Datalagret: Klart				
Fre 2/10	Hård deadline	Datalagret Godkänt	24 tim		Vag	1

Bidra med icke-trivial förbättring - installationsmanual

- Tisdag 22/9:
 - 8 tim Skapa branch av master, skapa issues att göra, skriva instruktioner i READMEs i markdown, komplettera inledning.

Datalagret Godkänt

- 2 tim Implementera load funktion i datalagret.
- 1 tim Implementera get__project__count funktion i datalagret.
- 1 tim Implementera get__project funktion i datalagret.
- 3 tim Implementera search funktion i datalagret.
- 2 tim Implementera get__techniques funktion i datalagret.
- 2 tim Implementera get__technique__stats funktion i datalagret.

Korrigera brister: Projektplanen

- 1 tim 23 min Stycket Arbetsmetodik skrevs till.

Ambitions mål för veckan:

- 1 tim Läsa på om och uppskatta tid för att implementera en sqlite3 databas för att uppgradera portfolions back-end senare.

Vecka 41 - Presentationslagret

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

Presentationslagret

- 1 tim Diskutera potentiella ändringar av LOFI:n.
- 4 tim Fixa temasida som agerar template åt alla andra sidor samt lära sig mer om Jinja2, HTML5 och CSS3.
- 3 tim Statisk eller dynamisk sida med profilbild, bild för senaste projekt och 'Om mig' text på / sidan.
- 5 tim Dynamisk sida som sorterar och listar projekten på /list. Knappar som sorterar sökresultaten. Varje sökresultat har lite beskrivning och liten bild.
- 3 tim Visar fullständig infosida för specifikt proj med id på /project/id. Samtliga projektsidor.
- 3 tim Sammanställning av alla project baserat på använda tekniker på /techniques.

Ambitions mål för veckan:

- 5 tim Kunna visa på tekniksida i hur stor omfattning olika tekniker använts för ett viss projekt.
- 2 tim Ha ett bildspel innehållandes projektbilder på förstasidan.

Vecka 42 - Systemdokumentation, -demonstration och Portfolio publicering

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

- 1 tim Läsa på om vad systemdokumentationen ska innehålla.
- 1 tim Mindmap samt skiss för översiktsskiss.
- 1 tim Sekvensdiagram
- 1 tim Skriva dokumentation kring felhantering.
- 1 tim Beskrivning av metoder och program som används vid felsökning.
- 2 tim Skriva ett första utkast.
- 2 tim Renskriva Systemdokumentation
- 30 min Fixa bilders position, rättstavning,

Portfolion Publicerad

- 1-2 tim Reda ut potentiella buggar.

Ambitions mål för veckan:

- 4 tim Gjort portföljerna personliga.

Vecka 43 - Reflektionsblad och Testdokumentation

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	0-2 tim		Vag	2

Testdokumentation inlämnad

- 1 tim Generera daterad testlogg där samtliga testfall körts.
- 1 tim Dokumentera minst ett relevant testfall per funktionellt krav.
- 1 tim min Skriva ihop testdokumentationen.
- 30 min Positionera bilder och rätta till text.

Individuellt Reflektionsblad

- 1 tim Plocka ut relevanta delar från dagboken.
- 1 tim Skriv ihop delarna i ett latex dokument.
- 30 min Positionera potentiella bilder och rätta till text.

Korrigerat eventuella brister i systemdokumentationen.

- 0-2 tim Rätta till potentiella brister.

Ambitionsmål för veckan:

- 5 tim Laddat upp portfolio på egen webbserver.