

# TDP003 Projekt: Egna datormiljön

## Projektplan

Författare

Daniel Huber, [danhu849@liu.se](mailto:danhu849@liu.se)  
Jens Öhrnell, [jenoh242@liu.se](mailto:jenoh242@liu.se)

## Innehållsförteckning

<b>1</b>	<b>Revisionshistorik</b>	<b>2</b>
<b>2</b>	<b>Introduktion</b>	<b>2</b>
<b>3</b>	<b>Tekniker</b>	<b>2</b>
<b>4</b>	<b>Kravspekifikation</b>	<b>3</b>
<b>5</b>	<b>Riskbedömning och åtgärder</b>	<b>3</b>
<b>6</b>	<b>Tidsplanering vecka för vecka</b>	<b>4</b>
6.1	Vecka 37 . . . . .	4
6.2	Vecka 38 . . . . .	5
6.3	Vecka 39 . . . . .	6
6.4	Vecka 40 . . . . .	7
6.5	Vecka 41 . . . . .	7
6.6	Vecka 42 . . . . .	7
6.7	Vecka 43 . . . . .	7
6.8	Schema/Deadlines . . . . .	8
6.9	Tidsåtgång - Presentativ Del . . . . .	8
6.10	Tidsåtgång - Funktioner i datalagret . . . . .	8

## 1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.3	Tabellformatering, upplägg samt riskbedömning	240920
1.2	Innehållsförteckning och tekniker	230920
1.1	Aktiviteter inför varje deadline	220920
1.0	Kravspec och samling deadlines	200922

## 2 Introduktion

Projektets mål utgörs av att skapa, presentera och underhålla en webbaserad portfolio. Där presenteras de projekt som vi i och utanför universitetet ska färdigställa 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 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 bild. Om projektet inte finns visas relevant felkod i.e. 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 ingår i utvecklarteamet. Systemtesten och uppkomna fel dokumenteras och ska vara åtgärdade i den slutgiltiga versionen av projektet. Testerna skrivs sedan in i systemdokumentationen.

## 5 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 ruscha arbetet innan deadline. Till vår hjälp har vi fått betyget med beröm för tidigare inlämningar. Detta gör att vi i värsta fall har möjligheten 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 vi inte kunnat förutse då vi saknar kunskap. För att underlätta har vi satt ambitionsmål 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.

## 6 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, milstolpar och ambitionsmål presenteras i tabeller. Ifall det saknas deadlines, milstolpar och ambitionsmål för någon vecka så saknas också tabell. Observera att med tidsåtgång så menas den sammanlagda tiden samtliga projektmedlemmar lagt ner tillsammans.

### 6.1 Vecka 37

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

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

- 8/9:
  - 3 tim sammanställande av vad ett planeringsdokument ska innehålla.
  - 2 tim skrivande av dokumentet.
- 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 i texten.

## 6.2 Vecka 38

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

Den Grundläggande installationsmanualen färdigställdes på 9 tim och 5 min:

- 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.
- 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:

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

### 6.3 Vecka 39

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
24/9	Hård deadline	1:a V Gemensam installationsmanual	24h	8h hittills	God	1
	Hård deadline	Projektplan, Första utkast	24h	ca 6h hittills	God	1

1:a Versionen Gemensam installationsmanual färdigställdes på 8 timmar hittills:

- 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 initiella svårigheter att push:a upp filer.
- 22/9:
  - 3 tim skriva issues, hjälp med felsökning av andras problem och evaluera färdiga issues.
  - 50 min Kompletterat inledningen och hjälpt att lösa andras merge conflicts.

Projektplan 1:a utkast färdigställdes på 6 timmar hittills:

- 17/9:
  - 1 tim Minst 2-3 aktiviteter definierades inför varje deadline.
- 22/9:
  - 4 tim Aktiviteterna för varje deadline skrevs in. Projektplanens upplägg bestämdes.
- 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.
- Sätta ambitionsmål ifall projektet blir klart i förtid.

## 6.4 Vecka 40

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
1/10	Ambitionsmål	Bidra med icke-trivial förbättring git eller tester	1h		God o inget	2
	Ambitionsmål	Korrigera eventuella brister installationsmanualen	1-2h		okänt	3
	Ambitionsmål	Korrigera Brister, Projektplanen	1-2h		Beror på	2
	Hård deadline	Datalagret Godkänt	24h		Vag	1

- 1/10:
  - Bidra med icke-trivial förbättring - installationsmanual eller tester
  - Korrigera eventuella brister gemensamma installationsmanualen
  - Korrigera Brister, Projektplanen
  - Datalagret Godkänt

## 6.5 Vecka 41

## 6.6 Vecka 42

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
	Hård deadline	Portfolion Publicerad	-		Vag	1
	Hård deadline	Systemdemonstration	3h		Vag	1
	Ambitionsmål	1:a Versionen Systemdokumentation	8h		Vag	3

- Portfolion Publicerad
- 1:a Versionen Systemdokumentation

## 6.7 Vecka 43

Datum	Typ	Beskrivning	Uppskattad tid	Tidsåtgång	Kännedom	Prio
22/10	Hård deadline	Testdokumentation inlämnad	8h		Vag	1
	Hård deadline	Individuellt Reflektionsblad	6h		Vag	1
	Ambitionsmål	Korrigerat event. brister i systemdokumentationen	2-3h		Vag	2

- Testdokumentation inlämnad
- Individuellt Reflektionsblad
- Korrigerat eventuella brister i systemdokumentationen



## 6.8 Schema/Deadlines

Vecka	Datum	Beskrivning	Uppskattad tid	Känedom
	12/9	Sätta emacs i python-mode	3h	Vag
	13/9	Inkorporera Magit i Emacs	2h	Vag
38	15/9	Bekantskap med Seleniumhq.org	3h	Vag
	17/9	Lofi-prototyp	6h	God
		Grundläggande Installationsmanual	4h	God
39	24/9	Projektplan, Första utkast	24h	Vag
		1:a Versionen gemensam installationsmanual	24h	God
	25/9	Flask o Jinja2 Föreläsning	2h	Vag
	27/9	Ha deploy:at första testhemsidan med flask	6h	Vag
		Fatta Jinja2	12h	Vag
40	1/10	Bidra med icke-trivial förbättring git eller tester	1h	God o inget
		Korrigera eventuella brister installationsmanualen	1-2h	okänt
		Korrigera Brister, Projektplanen	1-2h	Beror på
		Datalagret Godkänt	24h	Vag
42	15/10	Portfolion Publicerad	-	Vag
		Systemdemonstration	3h	Vag
		1:a Versionen Systemdokumentation	8h	Vag
43	22/10	Testdokumentation inlämnad	8h	Vag
		Individuellt Reflektionsblad	6h	Vag
		Korrigerat event. brister i systemdokumentationen	2-3h	Vag
Alla		Dokumentera Dagbok	20min/dag	God

## 6.9 Tidsåtgång - Presentativ Del

5h	Statisk eller dynamisk sida med bilder på /
8h	Dynamisk sida som sorterar och listar projekten på /list
2h	Visar fullständig infosida för specifikt proj med id på /project/id
3h	Sammanställning av alla project baserat på använda tekniker på /techniques

## 6.10 Tidsåtgång - Funktioner i datalagret

Nedan listas den specifika tidsåtgången för implementeringar av respektive funktion i datalagret.

Tid	Funktion
2h	load
1h	get_project_count
1h	get_project
3h	search
2h	get_techniques
2h	get_technique_stats