

TDP003 Projekt: Egna datormiljön

Planeringsdokument

Författare

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

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Planeringsdokument för TDP003	200909

2 Översikt

Projektets mål är att skapa, presentera och underhålla en webbaserad portfolio. Portfolion innehåller de projekt som vi i och utanför universitetet kommer fullgöra under de kommande 3 åren. Kravspecifikationen är skriven av programledningen. 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 fria tyglar ges.

3 Kravspecifikation

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

Projektet kräver att webbplatsen 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. Det är krav på att huvudsidan har bilder. Söksidan ska inneha funktionaliteten att kunna sortera projekt på projekt id samt göra det möjligt att kunna söka efter dem med hjälp av söktermer i ett formulär. Projektsidan ska visa fullständig information om det specifika projektet med en stor relevant bild. Om projektet inte finns visas relevant flekod i.e. status 404 'This page does not exist'. Tekniksidan innehåller information om projekten utifrån vilka tekniker och i hur stor omfattning dessa används i projekten. Varje projekt ska i listningar på söksidan och tekniksidan visas med en liten bild bredvid sig. Bildtext måste finnas till varje bild. Fel ska hanteras på ett, för slutanvändaren, informativt sätt så att denne kan förstå vad som gått fel.

Utöver det presentativa utgörs projektets andra del av datalagring och implementering av funktioner som hämtar lagrad data och presenterar detta för användaren. Vi kallar detta för simpelhetens skull för back-end. Varje projekt-data-del i portföljen 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. Back-end ska kunna hantera samtliga innan nämnda informationsdelar om respektive projekt.

Sökning med godtyckliga termer om projektets information genererar träffar och presenterar slutanvändaren med en lista där det mest förmodade objektet presenteras högst upp eller längst ner. Söklistan ska också kunna gå att sortera på använda tekniker. Det ska noteras att funktionaliteten ska möjliggöra sökning på ett ord, sortering och filtrering av tekniker kan ske samtidigt. Datan finns lagrad med UTF-8 teckenkodning i JSON-filen data.json. Datum är i formatet ISO 8601. Tilläggning av data i data.json filen är tänkt att antingen ske manuellt eller av andra verktyg i systemet. Förändringar i data.json filen ska 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.

De sex funktioner som hanterar back-end ska följa en bestämd standard. Denna standard möjliggör att delsystemen kan bytas ut. Samtliga namn ska vara skrivna på engelska. Funktionerna kommer vara: 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).

Projektet ska versionshanteras med git.

Systemet ska testas av två personer som inte ingår i utvecklarteamet för projektet. Testpersonerna är tänkta att testa de funktioner projektet huvudsakligen tillhandahåller. Systemtesten och uppkomna fel ska dokumenteras och vara åtgärdade i den slutgiltiga versionen av projektet. Testerna ska skrivas in i systemdokumentationen.

4 Tidsplanering med grov uppskattning på tidsåtgång

Nedan följer en grov uppskattning av tidsåtgång för de olika momenten nödvändiga att behandla innan projektet kan anses slutfört. Tidsuppskattningen baseras på att gruppens medlemmar dels skummat igenom innehållet för respektive ämne och sedan efter diskussion kommit fram till en förväntad tidsåtgång för att bemästra sagt ämne. Uppskattningen anses vara grov då det utgår ifrån nuvarande kunskap idag (2020-09-09).

4.1 Schema/Deadlines

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

4.2 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