LINKÖPINGS UNIVERSITET

TDP003 Projekt: Egna datormiljön

Planeringsdokument

Författare

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

INSTITUTIONEN FÖR DATAVETENSKAP

Höstterminen 2020
Version 1.0

2020-09-15

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Projektplan 1:a utkast	200915

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. Kravspecifikationen skrevs av programledningen och återfinns i dokumentet **DOkumentets namn här!**. 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 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 webplatsen 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 biler. På söksidan finns funktionaliteten att kunna sortera projekt på projektets id samt möjliggöra sökning med hjälp av söktermer i ett formulär. På respektive projektsida visas fullständig information om det specifika projektet tillsammans med en stor 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 och i hur stor omfattning dessa 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 komplett specifikation hänvisas läsaren till **Systemkravspecifikationens namn**.

Vid sökning med godtyckliga termer om projektets information genereras träffar och slutanvändaren presenteras med en lista där det mest förmodade objektet presenteras högst upp eller längst ner. Söklistan 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.

4 Tidsplanering vecka för vecka

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 utgås 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 - 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

4.3 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