

TDP003 Projekt: Egna datormiljön

Projektplan

Författare

Hadi Ansari, hadan326@student.liu.se

Nils Bark, nilba048@student.liu.se

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första versionen av Projektplan	2020-09-24
1.1	Kompletterad version av Projektplan	2020-09-29

2 Introduktion till projektet

Projektet “Egna datormiljön” handlar om att skapa ett portfoliosystem i Python 3 som hanterar olika projekt i form av en webbsida. Webbsidan visar en översikt över de senaste projekten samt information om ägaren på startsidan. De andra sidorna i webbplatsen innefattar en lista av alla projekt som kan sökas igenom inom olika kategorier, en lista där projekt filtreras efter använda tekniker, och en sida där mer information om ett projekt visas när detta projekt har valts av användaren. Portfoliosystemets planerade utseende beskrivs i LoFi-prototypen.

2.1 Arbetsplan

Arbetet på projektet kommer att utföras i grupper om två personer. Samarbetet sker främst under de arbetspass som beskrivs i planeringsdokumentet och en daglig rapport skrivs i respektive medlems dagbok efter varje arbetspass. Arbetet görs främst i gruppens datorsal men kan komma att göras på distans eller enskilt i hemmet ifall förutsättningarna för grupparbete på plats försämras. Det som görs under arbetspasset bestäms av de deadlines och arbetsområden som specificeras i tidsplaneringen. Kontroll av projektets framfart gentemot planerade deadlines sker löpande under projektets gång och justeringar av arbetets områden eller hastighet kan baseras på detta.

En viktig del av arbetet är det GitLab-projekt som används för att alla filer ska hålla synkroniserade mellan gruppmedlemmarna. Ändringar av projektet pushas via Git i slutet av alla arbetspass, även de som sker individuellt eller utanför planeringen.

2.2 Tekniker

För att åstadkomma ett fungerande system behövs det flera tekniker. Installation, kontroll, och enkel felsökning av nedanstående tekniker som inte redan finns installerade görs enligt stegen som beskrivs i installationsmanualen. Teknikerna som kommer att användas för att uppnå målet för detta projekt är:

- **Python:** python är ett programmeringsspråk som kommer att skapa de program som krävs av projektet. Det utgör grunden för bl.a sökfunktionen och datahantering.
- **Flask:** Flask är ett python-baserat webbramverk som används för att skriva webbgränssnitt. Det innehåller massor av moduler eller paket som är användbara för att skapa webbapplikationer. Flask kommer att hantera och köra webbgränssnittet för projektet, och kommer att underlätta visning av individuella projekt på projektsidan när användaren har klickat på det.
- **Jinja:** Jinja är ett underpaket som Flask använder sig av för att skapa HTML-mallar för python. Det kommer att underlätta designen av webbsidorna genom att via sitt mall-system t.ex låta projektets navbar och footer automatiskt visas på alla sidor, och förenklar uppdatering av sådana element.
- **HTML:** Ett standardspråk för att skapa webbsidor. Utgör grunden för skapandet av webbsidan.
- **CSS:** CSS beskriver utseendet av HTML sidor. Används för att designa webbsidans utseende genom projektets gång.
- **Git:** Används för att versionhantering i projektet.
- **JSON:** Ett format som används för sparande av projektdata.

3 Planering

Nedan följer det milstolpar som ska uppnås vid eller innan specifika datum. Milstolparna är baserade både på angivna deadlines och den interna planeringen kring hur de olika delarna av projektet bör ha genomförts för att projektet ska bli färdigt i tid. Tidsplaneringen är baserad på en uppdaterad version av planeringsdokumentet.

3.1 Milstolpar

27 sep 18:00: All planering är färdig och gruppen är påläst om datalagrets krav

01 okt 17:00: En fungerande sökfunktion har skapats och lite eller inget kvarvarande arbete på den ska finnas

09 okt 17:00: Presentationslagret är färdigt

12 okt 20:00: Hemsidan är färdig och kontrollerad inför presentation

15 okt 20:00: Portfolion publicerad och första versionen av systemdokumentationen inlämnad.

22 okt 21:00: Testdokumentation och reflektionsdokumentation är färdiga och inlämnade

3.2 Tidsplanering

Denna del innehåller en detaljerad planering av hur arbetet på de olika delarna av projektet kommer att läggas upp. Tiden som ges till varje arbetspass är baserat på hur lång tid den delen är beräknad att ta.

Vecka 38

Deadlines

Tisdag: Var mestadels färdig med LoFi-prototypen.

Torsdag: LoFi-prototyp inlämnad.
Grundläggande installationsmanual inlämnad.

Arbetspass

Måndag: Börja arbeta med LoFi-prototyp, 2h.

Tisdag: Skriv färdigt och lämna in LoFi-prototyp, 4h.

Torsdag: Börja arbeta med Installationsmanualen, 5h.

Fredag: Skriv färdigt och lämna in Installationsmanualen, 2h.

Vecka 39

Deadlines

Tisdag: Bidra till den gemensamma installationsmanualen.

Torsdag: Första utkast av projektplanen är inlämnat.
Första versionen av den gemensamma installationsmanualen inlämnad.

Arbetspass

Måndag: Hitta och välj uppgift att göra i gemensamma installationsmanualen, 2h.

Tisdag: Gör färdigt vald uppgift i gemensamma installationsmanualen, 4h.

Onsdag: Skapa mall och sätt alla rubriker. Flytta över planeringen från planeringsdokumentet 4h.

Torsdag: Skriv innehållet under resterande rubriker, 4h.

Fredag - Söndag: Läs på individuellt om datalagrets krav på egen tid.

Vecka 40

Deadlines

Torsdag: Bidrag till den gemensamma installationsmanualen inlämnat.

Eventuella fel i projektplan och gemensam installationsmanual åtgärdade.

Fredag: Datalagret godkänt av assistent.

Arbetspass

Måndag: Komplettera projektplan vid behov, annars jobba på JSON-fil, 2h.

Tisdag: Jobba på JSON-fil, sen sökfunktion, 4h

Onsdag: Hitta område att förbättra och bidra till gemensamma installationsmanualen, 2h, annars sökfunktion

Torsdag: Gör färdig sökfunktion, 4h

Fredag: Finputsa koden, 2h (innan lab).

Vecka 41

Deadlines

Måndag: Start sida färdigt

Tisdag: Teknisk sida färdigt

Onsdag: Projektlistsida färdigt

Torsdag: Specifik projektsida färdig

Fredag: Hela presentationslagret färdigt

Arbetspass

Måndag: Jobba på startsidan, 4h

Tisdag: Jobba på tekniksidan, 4h

Onsdag: Jobba på Projektlistsidan, 4h

Torsdag: Jobba på den specifika projektsidan, 4h

Fredag: Färdigställ hela presentationslagret, åtgärda eventuella fel eller förseningar, 2h

Vecka 42

Deadlines

Onsdag: Systemet ska vara redo för demonstration

Torsdag: Systemdemonstration för andra grupper.

Portfolion publicerad.

Första versionen av systemdokumentationen inlämnad.

Arbetspass

Måndag: Färdigställ och kontrollera hela systemet inför presentationen, 3h

Tisdag: Skriv på systemdokumentationen, 3h

Onsdag: Öva inför presentationen (1h), skriv på systemdokumentationen, 4h

Torsdag: Demonstrera systemet för andra grupper, ?h

Fredag: Skriv på reflektionsdokumentet, 2h

Vecka 43

Deadlines

Torsdag: Testdokumentation inlämnad.

Individuellt reflektionsdokument inlämnat.

Eventuella brister i systemdokumentationen korrigerade och inlämnade.

Arbetspass

Måndag: Skriv färdigt och lämna in reflektionsdokumentet, 3h

Tisdag: Börja skriva på testdokumentationen, 2h

Onsdag: Skriv färdigt och lämna in testdokumentationen, 2h

Torsdag: Komplettera systemdokumentationen ifall det behövs, 3h

4 Beräknad tid jämfört med faktiskt tid

I tabellen nedan listas den tid varje delmoment beräknas ta. Den uppdateras löpande under projektens gång med hur lång tid delmomenten tog i praktiken. Den beräknade tiden är den sammanlagda tiden av alla arbetspass som planerats för respektive delmoment.

Delmoment	Beräknad tid	Faktisk tid
LoFi-prototyp	6h	2h
Installationsmanual + Komplettering	7h	5h
Bidrag till gemensam installationsmanual	6h	3h
Projektplan + Komplettering	10h	6h
Datalager	12-14h	15h
Presentationslager	18h	?h
Sista kontroll av hela systemet	3h	?h
Systemdokumentation	7h	?h
Reflektionsdokument	5h	?h
Testdokumentation	4h	?h

5 Risker

Flera olika typer av fel kan dyka upp i ett projekt. T.ex sjukdom, dålig tidsplanering eller strulande programvara. För att minimera effekten av eventuella problem så kontrolleras ständigt gruppens arbete med de deadlines som är utsatta i tidsplaneringen så att åtgärder kan tas i god tid. Möjligheten finns att arbeta på helger eller hemma på egen tid om tidsbrist skulle uppstå.

Vid sjukdom finns det goda möjligheter att jobba över distans, och kontakt kan hållas via Discord och Zoom. Arbetet laddas regelbundet upp till respektive GitLab-projekt och därmed kan den som arbetar hemifrån med säkerhet ha en relevant version av projektet.

Ifall problem uppstår på grund av programstrul eller bristande kunskap så finns det alltid möjligheter att ta hjälp av labbassistenter eller internetsökning.

Vid seriösa problem som leder till akut tidsbrist finns det mindre nödvändiga delar av projektet som kan prioriteras ned eller tas bort helt och hållet, främst sådant som ligger utanför de specifika kraven för projektet, t.ex extra CSS-design på hemsidan, eller mindre nödvändiga delar av de många dokument som skrivs.