

Post Mortem för Funkar det

Marco Hyyryläinen - 06/06/2014

Abstrakt

Denna slutrapport kommer att handla om projektet *Funkar det* som jag har utvecklat under kursen Individuellt mjukvaruprojekt samt kursen Gränssnittsutveckling. *Funkar det* är en webbapplikation som förenklar grupperns planering - för stora som små grupper, oavsett aktivitetstyp. Det är ett enkelt och lättanvänt kalenderverktyg med tillhörande medlemslista.

På *Funkar det* kan medlemmarna få en överblick över kommande aktiviteter, och se vilka som tackat ja eller nej till att närvara. Rapporteringen sker smidigt via internet.

Idag existerar ett verktyg redan för detta men där har dock utvecklingen för denna produkt stått stilla sedan hösten 2009 och det finns inga planer från personen bakom att arbeta vidare med sidan. Därav ville jag skapa en ersättare för detta verktyg.

Funkar det är skriven i Ruby on Rails med en tillhörande PostgreSQL-databas. Gränssnittet är utvecklad i HTML5 och CSS3.

Innehållsförteckning

Abstrakt	2
Inledning och bakgrund	4
Positiva erfarenheter	5
Negativa erfarenheter	6
Sammanfattning	7
Referenser	8

Inledning och bakgrund

Redan innan projektets start visste jag att jag ville använda mig av ett nytt programmeringsspråk för mig, Ruby. Men ville även använda ramverket Ruby on Rails för att snabba upp utvecklingen.

Jag hade från början en fast idé av vilken typ av applikation jag hade ville skapa. På min fritid spelar jag brassinstrument i flertalet orkestrar och alla har samma problem: när spelningarna kommer in vet dom inte vilka som kan vara med. Detta går att lösa med papper och penna när man är 5 - 10 personer men när man kommer upp i trettioalet medlemmar blir det rörigt att hantera.

Idag existerar redan ett verktyg för detta. Dock har utvecklingen för denna produkt stått stilla sedan hösten 2009 och det finns inga planer från personen bakom att arbeta vidare med sidan.

Därför blev mitt mål att skapa en ersättare till det systemet som har responsiv design, är snabbare, enklare att använda.

Jag började projekttiden med att lära mig mer om språket, göra kodexempel med mera. Betaltjänsten Treehouse var en väldigt användbar källa för mig med deras videor om hur man skapar Rails-applikationer.

Efter fyra veckor påbörjade jag arbetet på min egna applikation, vilket jag borde ha börjat tidigare med. Efter två veckor av arbete på själva projektet hade jag en körbar version som fungerade att skapa grupper och logga in med.

Som databashanterare började jag med att använda en SQLite-databas som följer med Rails men var tvungen efter två veckor byta till en PostgreSQL-databaser. Anledning är att Heroku som jag använder för att publicera applikationen inte tillåter SQLite-databaser på grund av att den sparar sin information i filer som bland annat tar upp mycket utrymmer på deras servrar.

Hade jag vetat om detta från början hade jag inte behövt lägga tid på att porta över applikationen till en ny databasmotor.

Positiva erfarenheter

Det här projektarbetet har varit en väldigt lärorik process för mig. 10 veckor går väldigt fort, speciellt när dom infaller på våren där det finns flertalet röda dagar. Men även med det har jag utvecklats som programmerare och som projektledare.

Jag har lärt mig mycket om Ruby, ett språk som jag har haft ingen kontakt med tidigare, och Ruby on Rails där man kan säga samma sak. Rails bygger på designmönstret Model-View-Controller (MVC), även det har jag haft begränsad erfarenhet om men som har hjälpt mig att strukturera upp min kod på ett bättre sätt.

Arbetsgången i projektet startade långsamt men tanken om att det fanns mycket tid kvar. - "10 veckor, det är ju en evighet!". Men jag kan bekräfta att det är en kort tid och man behöver köra igång redan i början av projektiden för att hinna med det man vill.

För min del lyckades jag med att uppfylla mina baskrav i visionen men det krävdes väldigt många timmars arbete dom sista veckorna för att få det att fungera på ett stabilt sätt.

Under hela projektet har jag även tidrapporterat varje timme jag arbetat och med vad, vilket har hjälpt mig att inse vilka bitar jag jobbar mycket på och hur jag kan planera i framtiden på ett mer säkert sätt.

Negativa erfarenheter

Det som har gått mindre bra i projektet kan nog skyllas på hur applikationen är uppbyggd. Eftersom jag ville ha en delat användarnamn och lösenord för varje grupp som varje gruppmedlem kunde använda blev det mycket krångel för att "tvina in det" i dom ramarna som Ruby on Rails förväntar sig.

Ruby on Rails är ett ramverk med åsikter om hur man ska koda. Därför går det snabbt att göra en blogg, att göra-lista, eller ett vanligt medlemshanteringssystem. Men när man går utanför de tänkta ramarna måste man göra, i stort sätt, allting manuellt. Det kan finnas olika Gems, tillägg, som kan hjälpa en på vägen men då gäller det att ha tur att dom existerar.

I min applikation fungerar nu det gemensamma kontot för medlemmarna att logga in men det är löst genom att skapa två konton i skapandet av gruppen. Om jag skulle göra om applikationen idag skulle jag sätta det delade användarnamnet och lösenordet som extra fält för gruppen istället för att skapa en extra rad för det.

Testningen är en annan negativ erfarenhet - i alla fall mot slutet. I början utvecklade jag enligt principen *Testdriven utveckling* vilket fungerade väldigt bra. Koden blev säker och fungerade som tänkt. Dock mot slutet hamnade testningen efter och fick bara uppmärksamhet när dom började lysa rött och var tvungna att fixas innan jag kunde publicera applikationen. Inga nya tester skrevs under de sista veckorna vilket jag ångrar eftersom det då infördes systemet för att svara på evenemang.

Jag borde hållit hårdare i testningen och förbjudit mig själv att lägga till funktionalitet om inte det fanns ett test för det.

Planeringen har inte heller varit det bästa under hela projektet. Vissa saker har tagit mycket mer tid än jag tänkt, vilket gjort att jag hamnade lite efter i planeringen. Jag påbörjade även själva projektet för sent tycker jag. Dom första veckorna är värdefull tid som jag borde ha använt för att koda på heltid.

Sammanfattning

För att sammanfatta projektet är jag relativt nöjd med min insats. Jag kunde ha arbetat hårdare i början av projektet men det tog jag igen på slutet. Planeringen kunde ha skötts bättre för att få det mer strukturerat. Men jag klarade av dom målen jag hade satt upp för mig själv.

Även om slutprodukten saknar funktioner som finns i det existerande verktyget som jag har som mål att ersätta så finns det utvecklingspotential. Även grundläggande fel som finns, som att gruppinloggningen är en separat användare, får leva kvar ett tag till medan jag vidareutvecklar applikationen.

Det är mitt mål att förbättra applikationen så pass att den går att lansera skarpt till kunder.

Referenser

- Läs mer om [Ruby on Rails](#).
- Läs mer om [Treehouse](#).
- Läs mer om [PostgreSQL](#).
- Läs mer om [Heroku](#).