

# Boomerang deltagare och scheman

Jag har valt att utvärklä koden på egen hand då resten i gruppen inte satsade på samma betygs nivå som mig.

Vi har tagit fram ett användargränssnitt till en hemsida där funktionärer ska kunna gå in och editera deltagarnas uppgifter kring aktiviteter de vill delta på under ett läger.

Saker vi har sagt att vi vill kunna göra på hemsidan:

- Ändra namn på deltagarna (om någon har skrivit fel).
- Lägga till till deltagare.
- Ändra priogrupp så att man är i samma som sinna vänner om det är möjligt.
- Välja vilka aktiviteter man vill gå på per dag.

Saker som inte funkar än:

- Kommunikationen till apiet för deltagarnas aktiviteter. API problem.

## Hur vi har gjort för att åsta komma detta

När vi fick uppgiften kände jag att jag inte ville göra något tråkig instagram flöde, eller liknande. Jag frågade lite löst i en grupp som kallas /dev/scout och fick snabbt svar att de hade planerat att utvärklä ett api och mobil app till det men att det skulle finnas ett web alternativ för det så de kunde tänka sig att prioritera APIet.

Vi som grupp satte oss ner och kodade upp en struktur om hur vi ville att sidan skulle se ut. Sedan delade vi upp lite vad vi kunde göra för olika områden. Jag tog stor vikt vid att vara projektledare under den tiden som APIet byggdes. Det vi gjorde medans vi inte hade ett api att jobba mot var att vi byggde upp ett document med json för att kunna starta upp att bygga hemsidan, mycket av det vi gjorde i tidigt skede är nu över skriver med ny kod. Det har också hjälpt att skriva om något snarlikt flera gånger för då lär man sig också vad och hur man ska göra. Mycket har jag lärt mig också genom att förklara för de andra hur man kan lösa vissa problem.

Om vi kollar lite i koden mer specifikt i js/deltagare.js. I botten av documentet har jag en event listener som kollar om domen är laddad av det körs funktionen init som startar igång prosesen med att hämta data för deltagare och "Ny deltagare" knappens funktion.

När jag hämtar data för användare så startar jag med att plocka ut en lista på allas aktiviteter och lägger det i en global variabel. Liknande händelser sker för att hemta data på vilka aktiviteter som finns och när aktiviteterna är (occasions) där aktivitetens uppgifter läggs in i tillfälets data. Till sista förfrågas deltagarnas data där sådant som namn, prioGrupp och userLevel (om de är admin eller inte). Varav de skrivs ut i funktionen userData (rad 327) där de skrivs ut i en lista.

Om man sedan trycker på "Ändra" så kommer en funktion som samlar upp deltagarens information vilka aktiviteter hen vill gå på. Som sedan går till nästa funktion som lägger till och ändrar saker i domen för att göra det möjligt att ändra uppgifter om användaren.

Cancel funktionen som ligger inbakad gör att det går tillbaka utan att spara några ändringar, Delete skickar uppgifter om vilken deltagare som man tar bort till apiet och sätter texten i det elementet till "Removed" om apiet svarar.

Trycker man istället spara så samlas datan ihop och ändrar tillbaka listelement till ursprung och lägger till den användarens userscheduled till listan över det.

Save new user är väldigt snarlik men med lite ändringar iordning att det inte finns någon gammal data att hämta.