

Inlämningsuppgift: Meetup

Vad handlar uppgiften om

Du ska demonstrera dina kunskaper om designmönster och testning, genom att bygga en webbapp med hjälp av TDD. Du får diskutera uppgiften med klasskamrater, och ta hjälp med manuella tester och felsökning. Men varje student gör sitt eget projekt.

En *Meetup* är en sammankomst för personer med ett gemensamt intresse. Det kan handla om platsbundna eller online evenemang. En meetup kan ha ett begränsat antal platser. Din uppgift är att göra en webbapp där besökare kan anmäla sig till evenemang. Förutom detta ska besökare kunna se kommande och passerade evenemang, diskutera och rejta dem, samt lägga upp egna.

Specifikation

Detta ska du göra:

1. UML use case-diagram, som visar vad användaren kan göra. (VG)
2. User stories - skriv ner dem i projektets README.md
3. Använda TDD, dvs. skriva testfallen först
 - a. testfallen ska återspegla dina user stories
4. Använda TypeScript-interface för att beskriva appens data
5. Implementera appen
6. Gränssnittet (CSS, UI och UX) ska vara funktionellt

Utgå från den gula texten ovan och ta fram use case, user stories och acceptanskriterier.

Exempel:

- Som en besökare vill jag se meetups som inträffar snart först, därför att de är mest intressanta för mig.
- Som en besökare vill jag att passerade meetups ska markeras tydligt, så att jag inte försöker anmäla mig till något som redan har varit.
- Som en besökare vill jag kunna rejta evenemang jag har varit på med skalan 1-5, för att ge feedback till arrangören och andra besökare

Forts. på nästa sida

Obligatoriskt för G:

- det ska gå att köra dina test:
 - `git clone` → `npm install` → `npm run test`
- alla testfall gröna
- dina user stories står i README.md
- se meetups, visas i kronologisk ordning
- anmäla sig till meetup
- skriva kommentarer till en meetup

För VG:

- use case-diagram
- lägga till ny meetup
- filtrera (söka efter meetups vars tema innehåller ett sökord)

Tips! För att göra manuella tester är det användbart att implementera "tidsresor" (möjlighet för dig och läraren att skriva in ett datum enligt "YYYY-mm-dd", som appen ska använda i stället för dagens datum)

Begränsningar

Detta behöver du *inte* göra:

1. inloggning/autentisering (tips: skapa en standardanvändare. Men om du ändå gör inloggning, se till att läraren får rätt inloggningsuppgifter)
2. databas, API och AJAX (tips: använd localStorage)
3. routing
4. publicera app (tips: surge.sh)

Inlämning och bedömning

Du ska lämna in:

1. ditt repo, zippat (tips: ladda ner zip-fil från GitHub så får du inte med node_modules eller build-mappen)
2. länk till ditt publika repo på GitHub
3. (länk till publicerad app)
4. länk till use case-diagram (VG; se till att länken går att öppna)

Om du vill få direkt feedback kan du demonstrera din app i skolan sista kursveckan.

För G ska uppgiften uppfylla alla G-kriterier och vara korrekt inlämnad.

För VG ska uppgiften dessutom uppfylla alla VG-kriterier.