

24-timers opgave - Event Management System

Programmering-2, 3. semester KEA, forår 2023



Du skal i dette eksamensprojekt bygge en simpel full stack web application "til" et Event Management firma. Fokus vil være på at oprette events, administrere deltagere (attendees) samt tildele lokationer til events.



[Meget kort \(< 4 min\) video introduktion til opgaven](#)

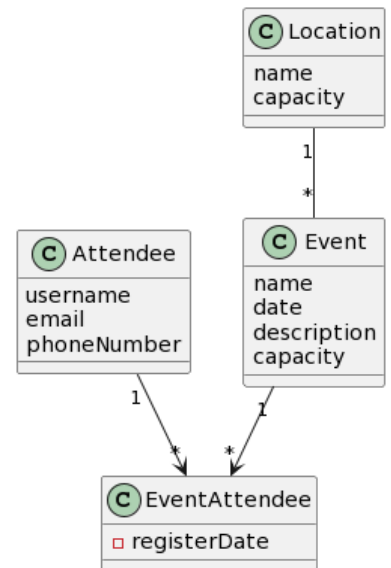
Backenddelen skal implementeres med Spring-Boot, JPA/Hibernate og en MySQL-database. Frontend skal være en separat HTML/CSS/JavaScript-applikation.

Det forventes, at du arbejder alene på øvelsen og ikke deler nogen form for kode/idéer med andre, hvilket vil blive betragtet som eksamenssnyd.

Se i denne forbindelse også afsluttende afsnit om brug af ChatGPT og lignende.

Det er ikke krævet, men en valgmulighed du får, at der implementeres adgangskontrol. Læs hele opgaven før du starter, og hvis du mener du vil kunne implementere sikkerhed, vil det nok være nemmere at tænke dette ind helt fra start.

"Product owner" har udarbejdet denne indledende domænemodel, som du kan bruge som inspiration for dit design (du er mere end velkommen til at ændre, så længe det giver mening for den overordnede opgave)



Følgende opgaver er opstillet af product owner i prioriteret rækkefølge:

Bemærk: Selvom det ikke er nævnt specifikt under delopgaverne herunder, skal du sørge for at inkludere relevante testcases i din backend.

Opgave-1: Backend-projektet og database

Implementer et Spring Boot-projekt med database og JPA for de entiteter, der er beskrevet ovenfor. Tilføj, for alle entiteterne, to ekstra felter, created og updated, begge af typen LocalDateTime.

Opgave-2: Events

Det er ikke krævet, at du tilføjer sikkerhed til dette projekt. Se dog opgave 4-mulighed 2. Hvis du regner med at implementere denne del, bør du tilføje en kommentar over hvert endpoint, der beskriver, "hvem" du mener skal være autoriseret til at benytte det - authenticated users (eventuelt med en rolle), eller alle.

Tilføj nødvendig kode for at implementere et REST API + en tilsvarende HTML/CSS/JavaScript-klient, der implementerer funktionalitet angivet herunder:

- Oprette en Event
- Se alle events, men kun → id, navn, dato, beskrivelse.
- Find og se detaljer for et enkelt Event, kun → id, navn, dato, beskrivelse
- Opdatere en Event

For de følgende krav skal du **kun** fokusere på Backend-delen, og udelukkende tilføje frontend-kode hvis du har tid efter at have fuldført det meste af opgave-3+4:

- Find en Event givet dens navn
- Slette en Event

Opgave-3: Events og Attendees

Mangler du tid, behøver du ikke tilføje frontend-kode, der tilføjer/redigerer attendees. Brug i stedet blot hardkodede attendees til at implementere følgende (backend og frontend):

- Et endpoint til at registrere en Attendee til en bestemt Event (hvis der er ledige "pladser")
Bestem selv hvilken information det vil give mening at returnere efter en succesfuld registrering
- Et endpoint til at afmelde en Attendee fra en Event (husk at opdatere ledige pladser)
- Opdater de endpoints, der returnerer en eller flere Events til at inkludere resterende antal ledige pladser

Vælg nu **en, og kun en**, af følgende muligheder, **Location** eller **Security**

Opgave-4 (mulighed 1) - Location

Tilføj nødvendige ændringer til din back/front-end for at implementere følgende funktioner:

Benyt blot nogle få hardcoded locations for det følgende

- Tildele en Location til en Event (kun hvis kapacitet for Location er tilstrækkelig i forhold til det maksimale antal tilladte deltagere for den pågældende event)
- Se alle Locations
- Se Location detaljer, inklusiv tilknyttede events (find selv ud af hvordan du vil implementere dette på klienten)

Implementer ovenstående features én ad gangen (backend/frontend)

Opgave-4 (mulighed 2) - Security

Tilføj sikkerhed med en login-funktion og opsæt sikkerhed, i rækkefølgen angivet nedenfor

Tilføj nødvendige ændringer, så alle endpoints, du mener kræver en bruger der er logget ind, eventuelt med en krævet rolle, **kun** kan benyttes af brugere logget på, og hvis krævet, med nødvendig rolle.

Opret et, eller ændr et eksisterende, endpoint, der tillader attendees logget ind, at se alle events, han er tilmeldt (kun token må sendes med i request, **ikke username**)

Tips:

Brug ikke for meget tid på, hvem der kan gøre hvad, tag bare en hurtig beslutning for hvert endpoint

Du er velkommen til at bruge hardkodede managers/admins og brugere opsat i koden til denne del.

Yderligere Features - **kun** hvis du har tid/lyst

Du er velkommen til at tilføje yderligere features, som du finder relevante for at vise dine færdigheder, hvis de ikke er inkluderet ovenfor. Dette kan omfatte pagination, responsive design, den anden valgmulighed for del-4, eller endda en "søg efter andre events" funktion, der henter events fra eksterne API'er eller andet.

Start **KUN** på denne del, hvis/når opgave 1-4 er **fuldført**.

Krav til aflevering

Du skal aflevere **et enkelt dokument** via Wiseflow med følgende oplysninger senest kl. 09.00 dagen efter, du har modtaget opgaven:

Dit fulde navn og KEA-e-mail

Et link til dit **PRIVATE** GitHub-repo (to når backend og frontend er i separate repositories)

Bemærk → Offentlige repositories vil IKKE blive accepteret

Når du opretter dine repositories, skal du tilføje følgende GitHub-brugere som collaborators (husk dette for både frontend og backend).

- lmar@kea.dk
- erlm@kea

Dette dokument skal også indeholde en kort beskrivelse (8-15 linjer), der beskriver, hvor langt du er kommet med projektet. Hvis du er nået til del-4, bør det fremgå hvilken mulighed du har valgt.

Vigtigt:

Du må IKKE uploade til dit GitHub-repo efter kl. 09.00, 24 timer efter du har modtaget opgaven. Dette vil gøre din aflevering ugyldig.

Forberedelse til den efterfølgende mundtlige eksamen

Selvom du ikke må uploade til dit repo som forklaret ovenfor, er du mere end velkommen til at lave lokale ændringer i din kode, og hvis du beder om det, vil du få et par minutter til at præsentere disse ideer.

Under eksamen vil du sandsynligvis blive bedt om at foretage mindre ændringer i din kode.

Om brug af ChatGPT eller lignende AI-værktøjer

Bruger du ChatGPT, eller en tilsvarende sparringspartner, er det vigtigt at forstå, at **du** er ansvarlig for den kode, du afleverer. Det betyder, at du forventes at kunne forklare, og eventuelt rette, koden, som, at du selv havde skrevet den. Husk altid at læse ChatGPT's supplerende forklaringer og generelt vurdere, om et AI-forslag nødvendigvis er det rigtige. At skrive supplerende tests (selv) for code snippets givet af en AI, er en god måde at både komme til at forstå koden bedre, og sikre at den virker som forventet.