

Applikationsutveckling för internet

Samverkande system

Vincent odoemelam | 2025-02-07

Table of Contents

[Inledning 2](#_Toc189743192)

[Systembeskrivning 2](#_Toc189743193)

[API 3](#_Toc189743194)

[Applikation 3](#_Toc189743195)

[Begränsningar 4](#_Toc189743196)

[Reflektioner 4](#_Toc189743197)

# Inledning

I denna Laboration ska två backend system som interagerar skapas. Systemen ska kunna köras oberoende av varandra men ska kompletteras av varandra. Den ena system ska hantera personuppgifter (userAccounts), det andra systemet ska hantera annonser (ad).  
CRUD funktionalitet implementeras så att man kan lägga till, läsa, ta bort och uppdatera med hjälp av GUI.   
I genomförandet av denna laboration fanns det möjlighet att utforska nya ramverk, kodspråk och teknologier eller fördjupa sig i något man redan använt.  
*Denna rapport är inte på plus nivå.*

# Systembeskrivning

Som användare börjar man med att skapa en USER med hjälp av ett formulär som hittas under fliken *Registrera.*

A screenshot of a red box

Description automatically generated

En användare kan logga in med sin Email och lösenord, man måste även välja vilken typ av användare man är. System behandlar SUBSCRIBER och COMPANY användare lite olika. Man kan bara skapa annonser när man är inloggad. Användare som har rollen SUBSCRIBER får vis information hämtad och visad i vyn ”Skapa annons”, om man har rollen COMPANY så kan man manuellt mata in informationen.

A screenshot of a red box

Description automatically generated

Annonserna är kopplade till annonsörerna genom email då den är unik i användartabellen. Detta betyder att det blir en *1->n* relation i tabellerna så EN användare kan har FLERA annonser. Eftersom tabellerna har en koppling genom email så behövs det bara två tabeller och all information är tillgänglig

# API

Det finns två API:er, både är skrivna i Java springboot och har fullständig CRUD funktionalitet. Frontend nyttjar inte alla API endpoints då det inte behöva men allt finns implementerat i BE.  
nedan ser vi exempel på updateUser, en PutMapping som inte används  
A screen shot of a computer screen

Description automatically generated

# Applikation

Två olika IDE:er användes under utveckling att applikationerna. För backend användes IntelliJ och för frontend används VS Code. IntelliJ är ett utmärkt verktyg för Javautveckling och ramverket Springboot förenklar mycket, tyvärr måste man ha Intellij Ultimate för att verkligen kunna nyttja Springboot till sin fulla potential, det går dock bra att bara använda community edition. Angular, till skillnad från springboot, blir inte begränsat beroende på IDE.   
Som applikationen är byggd idag så blir vyerna olika beroende på vilken roll användaren har. Det är egentligen samma komponenter som visas. Det finns villkor som döljer eller visar exempelvis logga in knappen, eller som gör att systemet fyller i information åt användaren.  
backend består at två microtjänster som tillhandahåller API åt frontend. Det finns full CRUD-funktionalitet men allt är inte implementerat gentemot frontend då det inte är nödvändigt för att åstadkomma det vi vill.

# Begränsningar

Trots att jag har gjort lite ändringar för att jobba runt vissa begräsningar och onödiga implementationer så finns det en del begräsningar i systemet. En del begränsningar finns på grund av at det är en skoluppgift och är irrelevanta för G nivå och andra är sakar som försummats, exempelvis utseende  
Här är en punktlista på kända begräsningar som systemet har:

1. Genvägar har tagit i styling, material UI används för att göra det lite snyggare men man har inte tagit hänsyn till exempelvis WCAG.
2. Vis information har dubbellagring, eller vad jag skulle anse vara felaktig informationshantering
3. Validering är extremt begränsad, man kan skapa en annons utan att fylla i någon information alls.

# Reflektioner

Efter att ha släpat på den här laborationen nu i 5 - 6 år så vill jag ändå säga att den var rolig. Mitt arbetsliv är väldigt likt det man jobbar med i denna laboration, och jag kan inte annat än hylla bredden man får lära sig. Jag önskar att jag tog den på lite större allvar så att jag kunde tagit examen i tid.   
något man jag inte riktigt reflekterade över första gången jag tog mig an laborationen var kravspecifikationerna, otydlighet och tvivelaktiga val sker ganska frekvent i skarpa projekt och efter att ha jobbet i ett nyutvecklingsprojekt kan jag konstatera ett det är något man får infinna sig i, och sen bara göra det bästa man kan i situationen.

Exemplet från denna laboration enligt mig är strukturen av databaserna, jag har valt att göra lite annorlunda för att slippa ha så mycket duplicerade data. Men en intressent kan ju komma tillbaka efteråt och be mig göra om och göra rätt eftersom jag inte följt instruktionerna. Även om jag ser något som jag tycker är konstigt skulle jag inte ändra på det innan jag tar det med kollat med kollegor, inte minst sagt när det gäller en så övergripande databas ändring.