IoT, Systemintegrations, grupparbetesuppgift.

Gruppuppgiften är att bygga en fullstack IoT-integrationslösning där en sensor av något slag är kopplat till ett device och man ska kunna se senorns historiska och senaste värde på en webbsida som hämtas från en SpringBoot-backend. Sensorns historiska mätvärden ska lagras i en databas.

Lösningen ska bestående av följande delar:

- En eller flera sensorer, kopplade till en Raspberry Pi, Arduiono, Feather eller annat device. Ni bestämmer själva vilken eller vilka sensorer ni jobbar med. Det räcker med ett device och en sensor, men det går bra att integrera flera.
- En databas där sensorernas värden loggas. (En vanlig databas på localhost går jättebra men ni får, om ni vill, använda er av en gemensam moln-databas. Ni är i så fall själva ansvariga för att klura ut hur man konfigurerar den och kopplar upp sig. Håll också örnkoll på kostnaderna eftersom det kan dra iväg rejält. Nackademin betalar inga databaskostnader!)
- En web service (=SpringBoot backend) där värden på sensorn kan hämtas ifrån (web servicen hämtar i sin tur datat från databasen och/eller sensorn). Ni bestämmer själva om web servicen ska köra på devicet eller på er ”vanliga” dator.
- En dashboard-webapp (som alltså visas i en webbläsare) som visar:
    - Sensorns historiska data, gärna organiserat antingen som en tabell eller som en graf. Datat hämtas från web servicen.
    - Sensorns senaste data
    - Använd gärna Thymeleaf för att presentera datat snyggt, Thymeleaf är enkelt att använda ihop med Spring. Om ni vill är det ok att kolla runt på olika JavaScript-bibliotek och experimentera med dem.

Uppgiften är väldigt fri, ni har stora befogenheter att själva bestämma hur ni bygger ert system och vilka tekniker som ska ingå, så länge som det finns minst en sensor, en databas, och en web service som på något sätt integreras. Att lösa hur datat från sensorn kommer in till databasen eller web servicen är upp till er, det finns många tänkbara sätt att få till detta. Det är helt ok att använda andra tekniker än dem som vi gått igenom i kursen, men berätta hur ni tänker för Sigrun först.

Grupparbete

Detta är en grupp-uppgift och ska genomföras och redovisas i grupp. Grupperna bör vara ca 3-4 pers/grupp. Ni bestämmer själva era grupper och ska presentera era grupper för Sigrun senast på måndag (18/9), ni kan skriva dem på Discord eller maila. Om några personer inte har kommit med i en grupp får dessa bilda en egen grupp. Om en person blir över kommer Sigrun att använda sitt lärar-veto och placera in den personen i en befintlig grupp.

I gruppen får ni själva komma överens om hur, var och när ni ska ses, jobba och ev. dela upp arbetet.

Betygskriterier

Krav för godkänt betyg (G):

- All funktionalitet nämnd ovan skall vara implementerad
- Projektet ska redovisas för klassen.
- Samtliga medlemmar som redovisar uppgiften ska ha deltagit i grupparbetet.
- Samtliga gruppmedlemmar som redovisar uppgiften ska ha checkat in kod i ert projekts repo på gitHub.
- Samtliga gruppmedlemmar som redovisar uppgiften måste förstå all kod i ert projekt.

Inga ytterligare krav för VG finns, fick du VG på första inlämningsuppgiften och blir godkänd på grupparbetet får du VG på kursen.

För er som snabbt blir klara

Spring-projekt är lätta att bygga ut och är kompatibla med massor med olika tekniker. Några tekniker om skulle vara lämpliga att lägga in i era projekt är t.ex WebSockets (för realtidskommunikation mellan backend och webbsida eller Thymeleaf (för snygg front). Välj en teknik som verkar intressant och bygg ut ert projekt. Detta projekt är också ett lysande tillfälle att träna på streams och lambdas.

Tips på bra tutorials: https://spring.io/guides

Redovisning

Uppgiften redovisas på fredagen den 29 sept, muntligt inför klassen. I redovisningen ska ingå demo av hela systemet, att ni berättar om arbetet, vilka designval ni gjorde, att ni visar och pratar om de intressantaste delarna av koden.

För att bli godkänd måste koden dessutom läggas in i inlämningsmappen på StudentPortalen.