

Harjoitus 1

Tehtävä 1

Täällä on selostettu Java EE –teknologioilla toteutetun pienen web-applikaation arkkitehtuuri:

<https://docs.oracle.com/javaee/7/firstcup/creating-example001.htm>

Mitä *staattisia* ja *dynaamisia* elementtejä tunnistat kuvauksen perusteella? Listaa elementit ja anna lyhyt kuvaus kullekin. Voit joutua selvittämään muista lähteistä joitakin käsitteitä (esim Wikipedian artikkelit Java EE:stä ja EJB:stä).

Tehtävä 2

Tutustu johonkin web-sovellusten (asiakas- tai palvelupään) toteuttamiseen tarkoitettuun ohjelmistokehykseen (*framework*, esim. Spring, Vaadin, Apache Struts, AngularJS, Node.js jne.). Mitä laatuviipuja löydät, eli mitä hyviä laatuominaisuuksia kehyksen käytöstä (lähes) automaattisesti seuraa?

Tehtävä 3

Tkt-laitoksen ("perinteisten") laskuharjoitusten vetäjät haluaisivat helpottaa ja nopeuttaa tehtyjen harjoitustehtävien ilmoittamista ja kirjanpitoa. Paperilistojen kierrättäminen harjoituskerran aluksi on paitsi antiikkista myös aikaa vievää. Harjoitusten jälkeen vetäjät joutuvat sitten syöttämään tehdyt tehtävät paperilistoista laitoksen KURKI-kurssikirjanpitojärjestelmään, mihin siihenkin kuluu aikaa, ja tiedonsyöttövirheitäkin tapahtuu helposti. Olisi myös opiskelijoiden kannalta mukavampaa, että he voisivat ilmoittaa tehtävät tehdyksi jo ennen laskuharjoitustilaisuutta (esimerkiksi heti tehtyään tehtävät). Jos opiskelijat voisivat ilmoittaa tekemänsä tehtävät etukäteen ja tilaisuuden vetäjä näkisi ilmoitukset, hän voisi myös valmistautua vähän tehtyjen tehtävien perusteelliseen käsittelyyn.

Ryhmien vetäjät ovat visioineet modernin ratkaisun, joka mahdollistaisi seuraavat toiminnalliset skenaariot

- Opiskelijoilla on käytettävissään erilaisilla päätelaitteilla (PC, tabletti, älypuhelin) ohjelma, jonka kautta he merkitsevät tekemänsä tehtävät
- Opiskelijat voivat nähdä kaikilla harjoituskerroilla merkitsemänsä tehtävät
- Ryhmän vetäjä voi päättää aikaikkunan, jolloin tehtäviä voi merkitä ja merkintöjä peruuttaa
- Tarkoitus on, että opiskelija voisi merkitä tehtävät heti ne tehtyään (siis jo hyvissä ajoin ennen harjoitustilaisuuden alkua)
- Ryhmän vetäjällä on käytössään oma ohjelma tai käyttöliittymä, josta hän voi reaaliajassa nähdä tietyn harjoituskerran merkintätilanteen
- Ryhmän vetäjä voi omassa ohjelmassaan merkitä harjoituskerran tietoihin, ketkä opiskelijat ovat esittäneen oman ratkaisunsa

- Ryhmän vetäjä voi nähdä tilastotietoa, kuinka monta tehtävää kukin opiskelija on esittänyt koko kurssin aikana
- Ohjelma auttaa vetäjää valitsemaan ratkaisun esittäjän käyttäen erilaisia algoritmeja (esim. satunnainen valinta suodattaen jo aikaisemmin ratkaisuja esittäneet pois). Valinta-algoritmeja pitäisi pystyä lisäämään ohjelmaan.
- Myös ryhmän vetäjän liittymä on käytettävissä erilaisilla päätelaitteilla
- Ryhmän vetäjä voi lisätä yksittäisiin tehtäviin omia merkintöjään (vaikea/liian helppo tehtävä jne.)
- Ryhmän vetäjä voi tehdä muutoksia opiskelijoiden merkintöihin (virheellisen merkinnän korjaus aikaikkunan sulkeutumisen jälkeen)
- Tehdyt merkinnät ja mahdolliset korjaukset päivittyvät automaattisesti KURKI-järjestelmään ilman, että ryhmän vetäjän tarvitsee niitä käsin sinne syöttää

Rajoituksia ratkaisulle

- Järjestelmä pyritään toteuttamaan pääasiassa oppilastyönä (OhTu-projekteissa)
- Halutaan käyttää opensource- tai muuten ilmaiseksi saatavilla olevia kirjastoja/kehyskiä
- Järjestelmän täytyy toimia laitoksen IT-ympäristössä, uutta laitteistoa ei ole tarkoitus hankkia
- Opiskelija- ja opettajaliittymien/applikaatioiden on toimittava yleisimmillä päätelaitteilla

Kysymys: pohdi, millä yllämainituista vaatimuksista ja rajoitteista voi olla merkitystä *arkkitehtuuri-ratkaisujen* kannalta.

Tehtävä 4

Minkälaisia *riskejä* edellisen tehtävän toteutusprojektiin (tai projekteihin) liittyy? Mistä seikoista haluaisit lisää tietoa tai täsmennyksiä, jotta voisit paremmin arvioida niiden arkkitehtuurisia vaikutuksia sekä riskien suuruutta?

Tehtävä 5

Lue Alistair Cockburnin artikkelit:

- Walking Skeleton <http://alistair.cockburn.us/Walking+skeleton>
- Incremental Rearchitecture <http://alistair.cockburn.us/Incremental+rearchitecture>

Selitä kummankin periaatteen (projektimallin) pääideat ja perustelet, miksi ne sopivat juuri ketterään ohjelmistokehitykseen.