

Tietokantasovellus

Tietokantasovellus

- Kurssin sisältö
- Suoritustapa
- Aikataulut
- Oppimateriaali

Tietokantasovellus - Sisältö

- Kurssilla opetellaan
 - Tietokannan suunnittelua käytännössä
 - Tietokantaohjelmointia
 - Verkkosovelluksen periaatteita (erityisesti web-sovellus)
 - Web-käyttöliittymän toteutusperiaatteita
- Esitietovaatimukset:
 - Tietokantojen perusteet, Ohjelmoinnin perusteet, Ohjelmistotekniikan menetelmät (Ohjelmistojen mallintaminen) (Käyttöliittymät)

Tietokantasovellus - suoritus

- Kurssi suoritetaan tekemällä harjoitustyö
 - pieni selainpohjaiseen käyttöliittymään perustuva tietokantasovellus
 - suunnittelu + toteutus + esittely
 - kaikki osat vaaditaan

Tietokantasovellus - suoritus

- Aikatauluhahmotelma:
- viikko 1: aiheen täsmennys + yleiskaavio + luettelo käyttötapauksista + tietosisältö
- viikko 2: suunnitelma + tietokannan rakenne + toimiva ympäristö + kannan pystytys
- viikko 3: käyttöliittymä + hahmotelma ohjelmiston rakenteesta + minitoteutus
- viikko 4: lisää toiminnallisuutta + käyttöliittymän yksityiskohdat
- viikko 5: toteutus edistyy, lisää toiminnallisuutta, ohjelmointi, testaus
- viikko 6: dokumentti, demo
- viikko 7: työ tarkastukseen
- viikko 8 tai 9: palaute opiskelijalle

Tietokantasovellus - suoritus

- Suoritukseen kuuluvat dokumentit:
 - Suunnitelma (1-3s)
 - sovelluksen suunnitelma
 - viikolla 1-2 (palaute samalla viikolla)
 - korjattavissa palautteen perusteella,
 - jotain konkreettista 'paperilla' – ei arvostella, mutta annetaan palautetta
 - Loppudokumentti
 - Suunnitelma + toteutus dokumentoituna

Tietokantasovellus - suoritus

- Dokumenteissa pyritään tiiviiseen ilmaisuun, joka kuitenkin on
 - Riittävä ratkaisun ymmärtämiseksi
 - Noudattaa edeltävillä kursseilla opittuja periaatteita (menetelmät & tekniikat)
 - Oppimateriaalisivulta löytyy dokumentointiohje

Suunnitelma

- Yleiskuva järjestelmästä
 - Sidosryhmäkaavio
 - Käyttäjärühmät
- Käyttötapaukset
- Järjestelmän tietosisältö
 - Käsitelmä
- [Käyttöliittymän hahmotelma]
- Relaatiotietokannan rakenne

Toteutusdokumentti

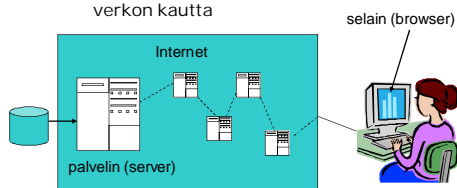
- Ohjelmiston yleisrakenne
- Järjestelmän komponentit
 - Sivut (staattiset / generoidut)
 - Ohjelmat – ohjelman tehtävä
- Asennustiedot
- Käyttöohje (jos tarpeen)
- Ohjelmat liitteenä

Tietokantasovellus - arvostelusta

- toiminnallisuus
 - sovitut toimivat perustoiminnot riittävät
 - tietoa pitää pystyä viemään kantaan ja hakemaan sieltä, tarkistukset
 - puutteet ja virheet vähentävät arvosanaa
 - **samanaikainen käyttökin hallittava**
- käytettävyyden
 - käytettävyyden tulisi olla kohtuullisen hyvä - ei kirjoiteta asioita, jotka järjestelmä tietää, ei erityisen kompeloä
 - toimiva, mutta ei välttämättä loistokas
 - selvät käytettävyydspuutteet alentavat arvosanaa
 - Kurssimateriaaliin liitetään lyhyt käytettävyydsmateriaali
- dokumentin laatu
 - dokumentin tulisi olla riittävän laaja sovelluksen rakenteen ymmärtämiseksi
 - sen tulisi olla yhdenmukainen toteutuksen kanssa
 - ja käyttää kursseilla opittuja tekniikoita
 - puutteet ja virheet alentavat arvosanaa
- toteutuksen laatu
 - ohjelmakoodin tulisi olla kohtuullisen laadukasta - ei copy/paste -tyyliä eikä kokkokoodia
 - tietokannan pitää olla toimiva

Web-sovellus

- Asiakas – palvelin malli
 - Käyttöliittymä toimii selaimessa
 - Sivut tuotetaan palvelimessa, saavutettavissa verkon kautta



Web-sovellus

- Palvelin voi olla monitasoinen
 - Sovelluspalvelin
 - Tuottaa sivut
 - Ottaa vastaan käyttäjän lähettämät pyynnöt
 - Hoitaa toiminnallisuuden
 - Tietokantapalvelin
 - Hoitaa tietokantaa

Web-sovellus

- Selain osaa
 - Näyttää HTML-kielistä aineistoa mahdollisen tyyliä määrittelyn (css) ohjaamana
 - Tallentaa ja välittää evästeitä (cookie)
 - Ottaa vastaan käyttäjän syöttämiä lomaketietoja ja välittää ne palvelimelle pyynnön yhteydessä
 - (Suorittaa JavaScript-koodia)
 - (Suorittaa appletteja tai muita ladattavia komponentteja, jos käyttäjä sallii - ympäristöriippuvaa)

Web-sovellus

- o Sovelluspalvelimen tehtävänä on tuottaa vastauksia selaimelta tuleviin aineistopyyntöihin
- o Vastaus aineistopyyntöön voi olla
 - Staattinen sivu
 - Ohjelman dynaamisesti tuottama sivu

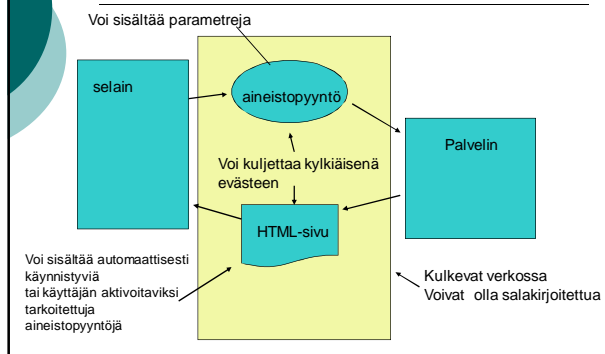
Web-sovellus

- Aineistopyyntö esitetään URL (universal resource locator) –muodossa
- Tyypillisesti pyyntö aktivoidaan näpäyttämällä linkkiä tai lähettämällä lomake, johon URL on liitetty.
- HTML-sivuun voi olla upotettuna aineistopyyntöjä, esim. kuvaviitteet, jotka aktivoituvat automaattisesti ilman käyttäjän toimenpiteitä

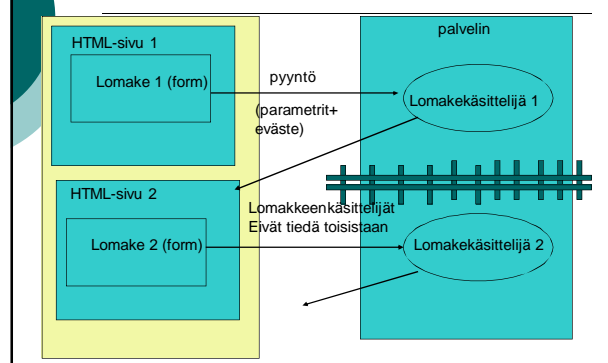
Web-sovellus

- URL identifioi tiedoston tai ohjelman. Pyynnön mukana voidaan toimittaa evästeitä. Palvelimelle lähetetään pyynnön yhteydessä vain siltä itseltään saatu eväste.
- Aineistopyynnön yhteydessä voidaan välittää myös parametreja:
 - lomakkeen kenttiin täytetyt tiedot
 - URL:iin sisältyvät parametrit
 - Get-pyyntössä parametrit osana URL:a
 - Post-pyyntössä erillisenä tiedostona

Web-sovellus



Web-sovellus



Web-sovellus

- Tiedon välittäminen lomakekäsittelijältä toiselle tapahtuu selaimen kautta.
- Välitettävä tieto voi vähimmillään olla vaikkapa istuntotunnus (evästeenä, lomakkeen piilokenttänä, linkin parametrina). Istuntotunnuksen perusteella voidaan järjestää pääsy palvelimen säilyttämiin istuntokohtaisiin tietoihin.

Lomakekäsittelijä

- Kuvassa lomakekäsittelijä on piirretty yhtenä 'oliona'. Käytännössä se voi jakautua useaksi eri olioksi:
 - Pyynnön vastaanottaja
 - Tutkii parametrit ja selvittää, mitä pitää tehdä
 - Palvelun tarjoaja
 - Varsinainen toiminnallisuus, tietokantakäsittely
 - Palautteen muodostaja
 - Rakentaa palautteena tuotettavan html-sivun

Toiminta palvelimessa

- Toiminta palvelimessa:
 - Pyynnön vastaanotto
 - Käyttöoikeuden tarkistus (& istunnon tilan selvitys)
 - Pyyntöparametrien purkaminen
 - Palvelun valinta parametrien perusteella
 - Palvelun suoritus / datan haku
 - Palautteen sisällön määrittely
 - (Istunnon tilan tallennus)
 - Palautteen valmistelu asiakkaalle
 - Palautesivun muotoilu
- Huom. palvelupyyntöjä voi tulla useita samanaikaisesti

Toteutuksesta

- Ohjeistusta tarjolla Oracle tietokannalle ja Java servlet -tekniikalle, hieman myös PHP ja PostgreSQL ohjeita.
- Miksi Oracle
 - Oracle oli pitkään ainoa hallinnoitu tkhj laitoksella, ts. varmistukset yms toimivat. Helppokäyttöisin. Ei tarvitse erikseen käynnistellä. Nyt myös PostgreSQL on käytettävissä samalla tavoin.
 - Muita: MySQL
- Miksi java
 - Kurssille voi tulla Ohjelmoinnin perusteet kurssin jälkeen, jolloin Java on ainoa opittu kieli
- Miksi servletit ilman mitään kirjastoja
 - On hyvä tietää, miten homma oikeasti toimii, jotta ymmärtäisi, mitä erilaisten kirjastojen taustalla tapahtuu.
 - (JSP on käytettävissä.)