

Samu Peltonen
samu.m.peltonen@student.jyu.fi
0405138355
Tietotekniikka

28.5.2018

Tietotekniikan ohjelmointityö: Yrityksen viestintäseinä

Tiivistelmä ohjelmointityöstä

Toteutettu ohjelmisto on viestintäjärjestelmä web-sovelluksena. Sovellus näyttää käyttäjälle viestejä, kalenterinäkymän deadlineista ja lähes-reaaliaikaisen chatin. Lisäksi sovellus tarjoaa mahdollisuuden kirjoittaa omia viestejä, kirjoittaa chatiin, suorittaa hakuja viesteistä, poistaa viestejä, sekä muokata asetuksia. Admin voi myös vaihtaa päivän viestiä ja tyhjentää chatin. Sovelluksen käyttäminen vaatii kirjautumisen. Kaikki säilöttävä tieto säilötään tietokannassa.

Sisällysluettelo

Tietotekniikan ohjelmointityö: Yrityksen viestintäseinä.....	1
Tiivistelmä ohjelmointityöstä.....	1
Tehtävän kuvaus, tausta ja tavoitteet.....	3
Käytännön toteutus.....	3
Oman työn arviointi.....	4
Työssä käytetyt lähteet.....	4
Ylläpitäjän tarvitsemat tiedot ja ohjeet.....	5
Sovelluksen mahdollinen jatkokehitys sekä ohjeet jatkokehittäjälle.....	5
Yhteenveto.....	5

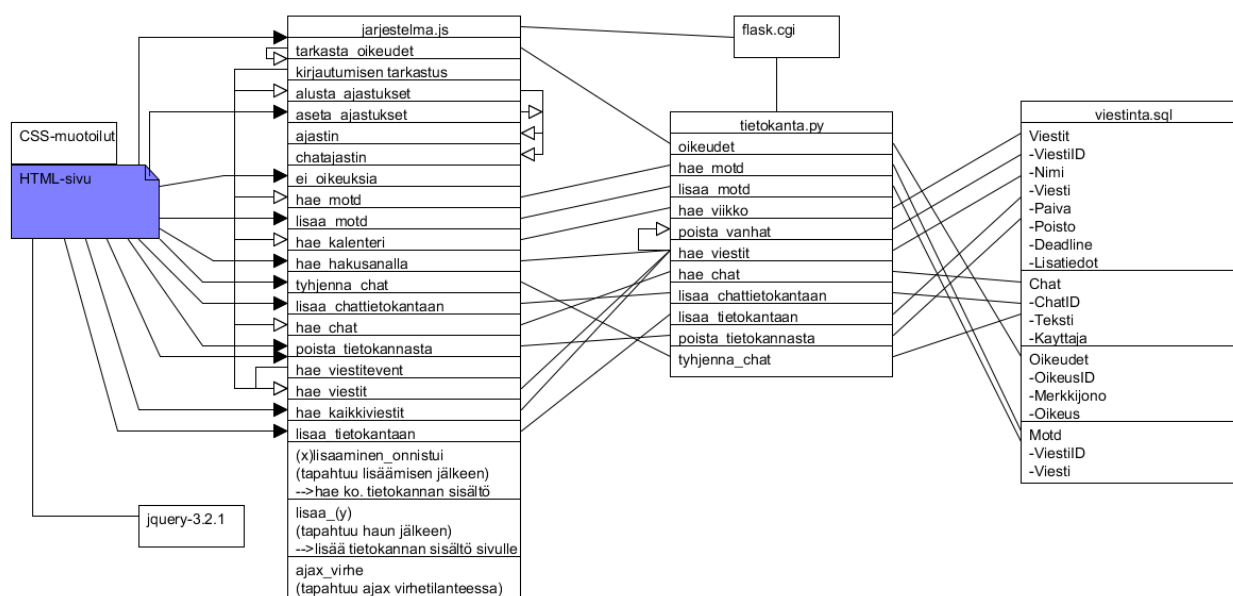
Tehtävän kuvaus, tausta ja tavoitteet

Tehtävänä oli toteuttaa sovellus, joka tietokantoja käyttämällä pystyy varastoimaan käyttäjien syötteitä ja esittämään niitä yksittäisellä HTML-sivulla joka ei vaadi uudellenlatauksia tai osoitteen vaihtoa, muodostaen näin viestintäsovelluksen.

Tehtävän taustalla oli hiljattain käyty web-ohjelmoinnin kurssi, joka myös näkyy ohjelmiston toteutuksessa.

Tavoiteina oli luoda ohjelmisto, jolla oikeasti voi harjoittaa tiimin välistä viestintää. Henkilökohtainen tavoite oli oppia lisää web-ohjelmointia ja parantaa ohjelmointirutiinia tällä saralla.

Käytännön toteutus



Kuva 1: Ohjelmiston rakenne (täysikokoinen kuva myös hakemistossa)

HTML-sivun toiminnallisuutta hoitaa Javascript tiedosto jossa on mukana valmiina haettu JQuery-kirjasto (JS Foundation). Käyttäjän toiminnot sivulla kutsuvat tapahtumankäsittelijöiden kautta Javascript-funktioita (merkitty mustilla nuolilla). Nämä funktiot taas saattavat kutsua vielä muita Javascript-funktioita (merkitty valkoisilla nuolilla). Javascriptin ja Python-ohjelman välinen tiedonvälitys toimii Ajaxina käyttäen flask.cgi-ohjelmaa (Javascript funktiot jotka kutsuvat Python-funktiota on linkitetty kuvassa viivalla). Python ohjelmat taas ovat ensisijaisesti vain tietokantakyselyitä ja -muokkauksia varten. Kyselyt ja muokkaukset tehdään SQL-tietokantaan (kunkin python funktion käyttämä taulu on linkitetty kuvassa viivalla). SQL-kyselyiden jälkeen Python ohjelma palauttaa Javascriptille tuloksen merkkijonona, joka sisältää HTML-elementtejä. Kun Javascript lisää tämän sivulle, saadaan valmiita HTML-rakenteita ja niiden sisällä on tarvittava

sisältö. SQL-muokkausten tapauksessa Python-ohjelma lisää HTML-sivulta Javascriptin kautta saadut syötteet tietokantaan mahdollisien muiden tietojen kanssa (esim. tämänhetkinen päivämäärä) ja palauttaa vain tiedon mahdollisesta onnistumisesta. HTML-sivun muotoilu on toteutettu CSS:llä.

Oman työn arviointi

Alkuperäinen tavoite toteutui lähes kokonaan ja lisäyksiä suunnitelmaan tuli huomattavasti. Erillisellä sivulla olevia kaikkia viestejä ei toteutettu koska päädyin toteuttamaan koko sovelluksen yhdelle sivulle. Prioriteettia adminin viesteille ei toteutettu, päivän viesti hoitaa osittain tarvittavan näkyvyyden saamisen. ”Työvaiheiden päivittämistä ja seuraamista” oli turhan abstrakti käsite enkä nähnyt sille tarvetta tässä sovelluksessa.

Suurin osa vaikeuksista oli Python-ohjelman kanssa, varsinkin muuttujien tyyppien, datetime logiikan, pienten syntaksivirheiden ja varsinkin kalenterinäkymän merkkijonon muodostuksen kanssa. Todennäköisesti alusta aloittaessa käyttäisin jotakin IDEä sujuvamman kehityksen takaamiseksi.

Käyttäjien yksilöimisessä oli hieman ongelmia mutta lähes-reaaliaikaisen chatin toteuttamisessa ei ollut suurempia vaikeuksia, joten olettamani ongelmakohdat olivat hieman väärässä.

Ohjelma ei näytä heti kaatuvan mistään, mutta säätämällä viestien noutamisen aikavälin tarpeeksi lyhyeksi ja antamalla sovelluksen suorittaa hetken aikaa, jokin todennäköisesti hajoaa.

Editorina toimi Notepad++. Vaikka kehittäminen tällä oli yksinkertaista, kunnon IDEllä kehittäminen olisi voinut olla huomattavasti tehokkaampaa ja bugien löytäminen ja mahdollisesti automaattinen korjaaminen olisi voinut nopeuttaa toimintaa. Valitut kielet, kirjastot ja muut tekniikat toimivat kehityksessä hyvin enkä tiedä niille parempia vastineita.

Työn alustava aikataulu piti melko hyvin, käyttöliittymän ulkoasua tuli tehtyä hieman pitkin projektia, mutta vähemmän kuin mitä sille oli allokoitu aikaa. Aikataulussa pysymistä helpotti se, että töitä oli aikaa tehdä päivittäin.

Jatkuva työn dokumentointi vastasi mielikuvia ohjelmistoprojektin työstämisestä. Toisaalta kehittämisen hoitaminen yksin tuntui normaalilta omalta ohjelmoinnilta, kun ei tarvinnut miettiä muiden kehittäjien lisäyksiä lähdekoodiin.

Työssä käytetyt lähteet

Web-ohjelmointi on suurelta osin opittu kurssilta TIEA218 joten se näkyy lopputuloksessa vahvasti.

<http://appro.mit.jyu.fi/web-sovellukset/> (Tommi Lahtonen)

Ylläpitäjän tarvitsemat tiedot ja ohjeet

Ohjelmisto ei vaadi suurempaa ylläpitoa. Viestit poistuvat automaattisesti (jos valittuna) ja chat-tietokannan tyhjentäminen vaatii yhden painalluksen adminilta. Uusien käyttäjien lisääminen tietokantaan vaatii tietokannan manuaalista käsittelyä. Tällä hetkellä käyttäjän lisääminen tietokantaan vaatii käyttäjän nimen ja salasanan liittämistä peräkkäin ja sha1 hajautustaulukolla encoodaamista, josta saadaan tarvittava tarkastusmerkkijono tietokantaan lisättäväksi.

Ohjelmisto ajetaan yliopiston palvelimen cgi-bin kansioista. Kaikki tiedostot ovat samassa kansiossa, poislukien tietokanta ja Python-ohjelman loki, jotka ovat samalla tasolla eri kansiossa nimeltä "hidden".

Sovelluksen mahdollinen jatkokehitys sekä ohjeet jatkokehittäjälle

Käyttäjien helpompi rekisteröiminen voisi olla joissain tapauksissa järkevä lisäys (ks. edellinen kappale). Käyttäjäkunnasta riippuen sovelluksen ulkoasua voisi myös muokata hieman viehättävämmäksi.

Yhteenveto

Kehitetty sovellus tarjosi haasteita, mutta työn tapahtuessa iteratiivisesti ja pikkuhiljaa, ohjelmointi oli verrattain yksinkertaista. Ohjelmiston rakenne jäi melko yksinkertaiseksi, hyvin suuri osa ohjelmistoa ovat yksinkertaisia tietokantakyselyitä ja muokkauksia tekevät funktiot. Halutut ominaisuudet valmistuivat aikataulussa ja tarjoavat ehjän kokonaisuuden. Git-versionhallinnan ja Githubin käyttäminen oli tehokasta ja tulen käyttämään sitä myös jatkossa tämentyylisissä projekteissa. Kunnollisen IDEn käyttämättömyys saattoi hidastaa kehitystä. Jatkokehitykselle on sijaa, muttei merkittävästi.