

# GamePlaza, Geneerinen pelifoorumi

## Alkusanat

Ideana oli tehdä pelifoorumi, jossa käyttäjät pystyvät julkaisemaan esimerkiksi peliarvosteluita, mielipiteitä, julkaisuja ja keskustelemaan niistä. Valitsin aiheen, koska siihen on helppo lisätä vaadittuja ominaisuuksia ja ne voidaan lisätä mielekkäästi sekä idean toteutus oli ohjelmointi tasolla mielenkiintoinen.

Koodi ei ole parhainta mahdollista (sisältää purkka -ratkaisuja) ja siitä puuttuu ominaisuuksia, jotka olisin halunnut sivustoon lisätä (ajan puutteen takia). Tiedostoja olisi voinut myös jäsenellä eri kansioihin. Sivustossa on myös joitakin tietoturvaongelmia, mutta en löytänyt aikaa niiden korjaamiseen. Huomaa ettei facebook login toimi, vaikka nappula ja kirjautuminen facebookkiin toimii.

Jos asennuksessa ja alustuksessa ilmenee ongelmia, niin voi minuun ottaa yhteyttä sähköpostitse: [simo.viljakainen@student.lut.fi](mailto:simo.viljakainen@student.lut.fi).

## Asennus ja alustaminen

### Asennus

Sivusto on kehitetty käyttämällä Windows käyttöjärjestelmälle tehtyä Xampp PHP kehitysympäristöä.

- Käyttöjärjestelmä : Windows 8.1
- Xampp 5.6.24
- Pavelin ohjelmisto: Apache 2.4.23
- PHP versio: PHP 5.6.24 (compiler: VC11 X86 32bit)
- Tietokanta: MariaDB 10.1.16
- memcache: memcache 3.0.8

Sivuston toimimiseen tarvittavat ohjelmistot saa helpoiten toimimaan lataamalla kyseisen Xampp:n version (tulee kaikki mukana). Olen lisännyt kaikki tarvittavat tiedostot "gameplaza.zip" -tiedostoon( <http://www.filedropper.com/gameplazasetup> ), (mukaan lukien memcache:n dll ja PHP.ini). Kun Xampp ja memcached on asennettuna, voidaan alustaa ohjelmisto.

### Memcache

Memcache:n voi käynnistää menemällä (järjestelmänvalvoja) komentoikkunalla kansioon missä memcached.exe sijaitsee ja kirjoittaa konsoliin käsky: "memcached.exe -d start".

Memcache:n asentaminen tapahtuu myös komentoriviltä, komennolla "memcached.exe -d install"

## Alustaminen

### Tiedostot

Järjestelmässä pitää alustaa tietoja, jotta ympäristö olisi toimiva. PHP.ini tiedostoa pitää muokata, jotta se löytäisi memcache\_dll:n ja voitaisiin käyttää ssl -yhteyksiä. Tämä tapahtuu helpoiten korvaamalla juuri asentamasi PHP.ini gameplaza\_setup.zip:ssä sijaitsevalla PHP.ini tiedostolla. Memcache myös tarvitsee versionsa/kääntäjän mukaisen dll -tiedoston (php\_memcache.dll). Tämä myös löytyy annetusta zip -tiedostosta. Kyseinen dll pitää siirtää polkuun xampp/php/ext/.

### Vakiot

Tiedostoista: datahandler.php, databasehandler.php, logout.php

Alusta kyseisten .php tiedostojen ( sivun alussa, define(vakion nimi, vakion arvo) ) vakiot käyttämiesi vakio arvojen mukaan.

WEBSITE -vakio on sivuston url , muista lisätä nimen alkuu "http://" ja loppuun "/"(jos et ole määritellyt muokattua url -osoitetta missä sivusto on, aseta arvoksi 'http://localhost/' ).

Vaihda tietokantavakiot DATABASE, HOST, USERNAME, PASSWORD tietokantasi mukaan.

### Recaptcha, SMTP

Jotta Captcha ja sähköpostiviestien lähetykset toimisi, tulee luoda recaptcha tili ja (<https://www.google.com/recaptcha/>) ja smtp -palvelun käyttäjätunnukset (googlen gmail suositus)

Alusta "site.php" sivulla olevat \$publickey muuttujat omalla recaptcha public avaimella (rivit: 303, 333) ja \$privatekey tiedostosta "datahandler.php" (rivi 285, recaptcha funktio).

SMTP -asetukset tulee muuttaa tiedostosta "mailer.php". Riville 17 (mail->host) tulee smtp palvelimesi (gmail valmiina), alusta riville 21, username eli sähköpostiosoitteesi ja seuraavalla riville sähköpostisi salasana (jotta palvelin voi lähettää sähköposteja tilisi kautta). Riippuen mitä smtp -palvelinta käytät, muuta arvot (portit yms) sen mukaan.

### Tietokannan rakenne

Tietokannalla on vain yksi taulu mikä pitää luoda ennen sivuston käyttöä, muut lisääntyvät dynaamisesti suorituksen aikana. Taulun nimi on users ja siinä on sarakkeet, uid INT(10) AUTOINCREMENT, username VARCHAR(50), email VARCHAR(50), phash VARCHAR(80), salt VARCHAR(50), rigths INT(2) DEFAULT 1. Owner/admin on ainoastaan mahdollista luoda (rigths=3) tietokannasta käsin. Lisäsin myös testitietokannan gameplaza\_setup.zip:iin jos haluaa vain lisätä tietokannan suoraan (sisältää myös valmiita threadeja ja forum posteja). Tietokannan voi lisätä mysql:ään phpmyadminin kautta (localhost/phpmyadmin ->import/tuonti).

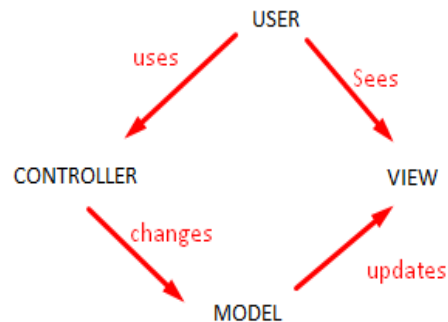
## Sivuston rakenne

Sivusto noudattaa MVC (5p) ja FRONT-controller (3p) -mallia (tai ainakin sivuaa).

Tiedostot:

fblogin	facebook library	3.12.2016 12:18	Tiedostokansio	
adminpanel.php		14.12.2016 23:36	PHP-tiedosto	1 kt
class.phpmailer.php	PHPMailer libraries	9.12.2016 2:03	PHP-tiedosto	141 kt
class.phpmaileroauthgoogle.php		9.12.2016 2:03	PHP-tiedosto	3 kt
class.smtp.php		9.12.2016 2:03	PHP-tiedosto	42 kt
databaseHandler.php	Model, handles all the data	15.12.2016 1:36	PHP-tiedosto	5 kt
datahandler.php		15.12.2016 20:51	PHP-tiedosto	10 kt
fblog.php	Facebook login (not used)	14.12.2016 21:41	PHP-tiedosto	1 kt
forumposts.php	view callers	14.12.2016 21:31	PHP-tiedosto	1 kt
gamesearch.php		11.12.2016 12:56	PHP-tiedosto	1 kt
home.php		14.12.2016 17:07	PHP-tiedosto	1 kt
index.php	Controller	15.12.2016 22:16	PHP-tiedosto	2 kt
js.js		14.12.2016 17:56	JavaScript-tiedosto	2 kt
login.php	view callers	15.12.2016 2:10	PHP-tiedosto	1 kt
logout.php		15.12.2016 3:10	PHP-tiedosto	1 kt
mailer.php	Email sender	14.12.2016 21:36	PHP-tiedosto	2 kt
moderator.php		14.12.2016 21:38	PHP-tiedosto	1 kt
print.css		14.12.2016 19:37	Cascading Style S...	1 kt
recaptchalib.php	Google captcha library	25.5.2010 17:39	PHP-tiedosto	10 kt
register.php		15.12.2016 1:46	PHP-tiedosto	1 kt
site.php	View	15.12.2016 22:12	PHP-tiedosto	10 kt
style.css		15.12.2016 3:12	Cascading Style S...	4 kt
threads.php		15.12.2016 3:07	PHP-tiedosto	1 kt
u.php	Utils library, functions	15.12.2016 19:26	PHP-tiedosto	2 kt

Sivuston toimintaperiaate:



Sivustolla navigointi tapahtuu Index.php view-kontrollerin kautta. Index.php ohjaa käyttäjän sivupyynnöt view caller sivuille (esim. login, threads), jotka taas pyytävät View-kontrollerilta view-mallia (sivulta "site.php" html, javascript), joka sitten esitetään käyttäjälle. Tällä voidaan rajoittaa käyttäjälle annettavaa viewiä, esim. voidaan esittää sivu erilaisena käyttäjälle jolla on enemmän oikeuksia kuin tavallisella käyttäjällä.

Käyttäjältä pyydettyä tai käyttäjän kysyessä dataa menee syöte/pyyntö Ajaxin/POST/GET:ien kautta datahandler.php (model controller) sivulle, joka tekee pyyntöjä käskyjä databaseHandler.php (model) sivulle. DatabaseHandler.php palauttaa halutun datan datahandler.php sivulle, joka taas kontroloi käyttäjän näkymää.

## Sivuston toiminta käytännössä ja ominaisuudet

GamePlaza -foorumi tarjoaa käyttäjilleen mahdollisuuden luoda uusia keskustelun aiheita (luodaan uusi taulu tietokantaan), lisätä kommentteja/julkaisuja avoimiin keskusteluihin ja käyttäjät voivat tehdä pelitietohakuja pelin nimellä (gamesearch). Palveluiden jakaminen perustuu käyttäjien eri tasoihin oikeuksiin. Alla olevat selitykset selittävät myös sivuston toimintaa.

### Toteutus (lisäselityksiä suurin osa selitetty alempana)

Kirjautuminen ja rekisteröityminen tapahtuu html formeilla ja data siirretään PHP:lle POSTia käyttäen, koska pitää käyttäjä muutenkin siirtää uuteen viewiin, niin en käyttänyt Ajaxia siihen. Keskusteluiden ja kommentointi tapahtuu Ajaxin välityksellä, jotta saadaan turha sivun päivittäminen pois. Keskustelun luonti luo uuden taulun tietokantaan, jotta voidaan lisätä ja hakea aina tiettyyn keskusteluun liittyvät kommentit. Gamesearch käyttää ulkoista APIa, jolla se saa datan ja parsee sitten XML datan viewiksi käyttäjälle. Svg graaffi piirretään chartist.js kirjastoa käyttäen, dynaamista tietokannan datalla. Graafilla on JQuery funktio (kutsutaan kun dokumentti on valmis eli sivu latautunut) joka lähettää get pyynnön eteenpäin, mikä sitten lähettää päivitetyn datan graafilelle.

## Roolipohjainen käyttäjänhallinta (2p)

GamePlaza -foorumeilla on kolmen tasoista käyttäjiä oikeuksiltaan. Käyttäjien oikeudet määritellään tietokannassa rekisteröinnin yhteydessä. Kun käyttäjä rekisteröityy, on hänen oikeudet aina 1 vakiona. Eli vierailijat (käyttäjät jotka eivät ole rekisteröityneet/kirjautuneet) ”oikeusarvo” on 0/null. Moderaattoreita pystyy ainoastaan owner/admin tasoinen käyttäjä luomaan sille suunnatusta admin -paneelistä. Admin -tasoisia käyttäjiä ei voi luoda muuten kuin suoralla tietokannan muuttamisella (tietoturvan takia).

Oikeudet = 0, vieras:

Sivustolla vierailijoilta on riisuttu oikeudet tehdä pelihakuja, tehdä uusia keskusteluita ja kommentoida keskusteluita.

Oikeudet=1, käyttäjä:

Käyttäjillä on perusoikeudet sivuston ominaisuuksiin. He voivat tehdä pelihakuja, luoda keskusteluja ja kommentoida toisten luomia/omia keskusteluja.

Oikeudet=2, moderaattori

Moderaattorin oikeuksiin kuuluu peruskäyttäjien oikeudet ja oikeudet poistaa muiden käyttäjien tekemiä kommentteja keskusteluista.

Oikeudet=3, Owner/Admin

Adminin oikeuksiin kuuluu moderaattorin oikeuksien lisäksi oikeudet poistaa käyttäjiä ja ylentää käyttäjiä moderaattoreiksi. Admin käyttäjät pääsevät kontrolleihinsa käsiksi admin -paneelin kautta (index.php?page=admin).

## Käyttäjien autentikointi (5p)

Vierailijat pystyvät rekisteröitymään sivustolle, jolloin heistä tulee käyttäjiä. Rekisteröitymis -prosessi on monivaiheinen. Ensiksi käyttäjä täyttää kentät (username, email, password, password again) ja täyttää captcha:n. Käyttäjän antama syöte lähetetään tarkastettavaksi datahandler.php sivulle. Jos syöte on sallittu ja captcha on täytetty oikein, tallennetaan käyttäjän antama syöte memcacheen myöhempää käyttöä varten. Tämän jälkeen generoidaan satunnainen merkkijono (tallennetaan myös memcacheen myöhempää vertailua varten), joka sitten lähetetään annettuun sähköpostiin (mailer.php, käyttää PHPMailer lib). Tämä tapahtuma siirtää käyttäjän uuteen näkymään, rekisteröinti kentät katoaa ja ilmestyy sähköpostivarmistus -kenttä johon käyttäjä voi kopioida vahvistuskoodin. Jos koodi on oikea, haetaan data memcachesta (käyttäjän syöttämät tiedot) ja tallennetaan ne tietokantaa databaseHandler.php:n avulla. Salasanasta muodostetaan suolattu bcrypt tiiveste, joka tallennetaan tietokantaan suolan kanssa (ei tarvitsisi tallentaa suolaa, koska sitä ei tarvita bcryptin kanssa). Samalla käyttäjä ohjataan login sivulle kirjautumista varten.

Kirjautumis -prosessissa käyttäjä syöttää username ja password kenttään tunnuksensa ja salasanansa ja täyttää captchan. Jos captcha on oikea ja syöte sallittu, haetaan tietokannasta annetun tunnuksen salasanatiiviste ja verrataan sitä password\_verify -funktion avulla haettuun tiivisteeseen. Jos nämä täsmää, tallennetaan session -muuttujiin kirjautujan oikeudet ja käyttäjänimi. Nämä session muuttujat kertovat

onko käyttäjä kirjautunut vai ei.

### **Välimuistin käyttö (3p)**

Välimuistia (memcache) käytetään rekisteröitymisessä. Generoitu satunnainen sähköpostin varmistuskoodi tallennetaan välimuistiin ja käyttäjän antamat rekisteröitymis -syötteet. Joita sitten myöhemmin haetaan memcachesta myöhempää käyttöä varten. Memcachen käyttö tapahtuu datahandler.php tiedostossa.

### **Ajaj (/ajax) -ohjelmoinnin hyödyntäminen (3p)**

Ajaxia hyödynnetään mm. keskusteluiden luomisessa, pelitietojen haussa ja kommentoidessa (keskustelussa). Sillä lähetetään dataa datahandler.php:lle ja viewille.

### **Dynaamisen SVG-grafiikan käyttö (3p)**

Home -sivulle generoidaan kontrollereiden kautta tietokannasta haettujen tietojen perusteella, keskustelut/kommentit -graaffi. Tietokannasta haetaan kaikki keskusteluiden nimet ja kommenttimäärät. Tämän jälkeen piirretään svg -graaffi chartist.js kirjaston avulla (koodi: site.php sivulla, chart -funktio).

### **Canvas-elementin käyttö (3p)**

Canvas elementtiä käytetään sivulla tapahtuvassa mainonnassa. js.js tiedostossa olevassa koodissa canvas -elementtiin generoidaan kuvia ja niihin liitetään linkkejä. Linkit ja kuvat vaihtuvat 5 sekunnin välein.

### **JSONin (/XML:n) käyttö tiedon liikuttelussa/tallennuksessa(3p)**

Sivustolla käytetään APIa (gamesearch.php) joka antaa tiedon XML muodossa. Data parsetaan datahandler.php sivulla ja lähetetään taulukkona gamesearch.php viewiin.

### **Kolmannen osapuolen palvelun käyttö (5p)**

Kirjautumisessa ja rekisteröitymisessä käytetään kolmannen osapuolen luomaa (google) recaptcha -palvelua, bottien ja bruteforce -hyökkäysten estämiseksi. Recaptcha tukee kirjautumis- ja rekisteröitymisprosessia.

### **Tietokannan käyttö tietojen tallennukseen(5p)**

Tietokantaan tallennetaan rekisteröitymisessä käyttäjän tiedot. Tietokantaan myös tallennetaan keskusteluiden aiheet (omat taulut) ja keskusteluiden kommentit + kommentoijan tiedot.

### **Responsiivinen ulkoasu(5p)**

Sivuston ulkoasuun vaikutetaan "style.css" tiedostossa. Siellä on määritelty näytön maksimi ja minimi koko, jotka vaihtelevat ulkoasua. Nämä ulkoasun määrittelijät pitävät huolen siitä, että sivusto pysyy aina luettavassa/käytettävässä muodossa. Elementit on määritelty prosentteina pikseleiden sijaan, jotta ne skaalautuisivat ikkunan koon vaihdellessa.

### **jQueryn / Reactin / AngularJS:n käyttö(3p)**

mm. Ajaxin pyörittely tapahtuu jQueryn avulla. JQueryä käytetään myös linkkien värjäämiseen js.js tiedostossa sijaitsevalla koodilla.

### **Toinen kolmannen osapuolen palvelun käyttö(3p)**

Varmistus-sähköpostien lähettämiseen käytetään googlen Gmail SMTP -palvelinta. Sähköpostien lähettäminen tapahtuu mailer.php tiedostossa. Mailer.php käyttää PHPMailer -kirjastoa. (Jos Gmail-smtp ei kelpaa, käytetään gamesearchsissa [www.thegamesdb.net](http://www.thegamesdb.net) luomaa APIa pelitietojen hakemiseen.)

### **Sivuston sisältö tulostuu paperille luettavassa muodossa (2p)**

Kun sivu saa print -komennon ottaa se muotoiluun käyttöön print.css tiedoston. Print.css tiedostossa on asetettu "display:none" kaikille elementeille joita emme halua paperille tulostaa (aikalailla pelkkää tekstiä, koska ei ole kuvia sivustolla).

### **Sivusto tarjoaa sisältöä ladattavaksi PDF-muodossa(2p)**

Jokaisessa keskustelu-sivussa on vasemmassa yläkulmassa linkki mistä voi ladata keskustelun pdf muodossa. Pdf muodostetaan käyttäen jsPDF -kirjastoa (rivi 207, site.php).

### **Yksikkötestit olemassa(5p)**

Käyttäjän syötteet tarkistetaan jokaisessa kentässä missä kysytään käyttäjän syötettä. Rekisteröitymisessä tarkistetaan käyttäjän antamat tiedot. Käyttäjänimestä katsotaan, että se on alfanumeerinen, eikä se löydy tietokannasta tai memcachesta (varattu tunnus), annetut salasanat tarkistetaan, että ne täsmäävät ja että salasana on tarpeeksi pitkä, eikä liian pitkä (9-255). Sähköpostin muoto tarkistetaan oikeelliseksi regexin avulla (u.php). Kirjautumisessa tarkistetaan käyttäjän antama käyttäjänimi, että se on alfanumeerinen. Keskustelun luomisessa tarkistetaan, että käyttäjän syöttämä nimi on alfanumeerinen ja sallitun merkkimäärän mittainen.

Mielestäni harjoitustyöni täyttää seuraavat vaatimukset:

yhteispisteet: 65 (kattava dokumentaatio laskettu mukaan, vertaisarviointia ei)

Palikka	Pisteet
Responsiivinen ulkoasu	5
Tietokannan käyttö tietojen tallennukseen	5
Käyttäjän autentikointi (luonnollisesti hashit ja suolat mukana)	5
Roolipohjainen käyttäjänhallinta (vähintään pääkäyttäjä, moderaattori/vastaava, peruskäyttäjä ja vieras)	+2
Front-controllerin käyttö	3
MVC-mallin mukainen sivusto	5
Välimuistin käyttö	3
Kolmannen osapuolen palvelun käyttö (esim. FB-login, [GA ei riitä])	5
Toinen kolmannen osapuolen palvelun käyttö (sää tiedot, smartpostit, elokuvatarjonta...)	3
Yksikkötestit olemassa	5
JSONin (/XML:n) käyttö tiedon liikuttelussa/tallennuksessa	3
jQueryn / Reactin / AngularJS:n käyttö	3
Dynaamisen SVG-grafiikan käyttö	3
Canvas-elementin käyttö	3
Sivuston sisältö tulostuu paperille luettavassa muodossa (aka on erillinen printtityyli)	2
Ajaj (/ajax) -ohjelmoinnin hyödyntäminen	3
Sivusto tarjoaa sisältöä ladattavaksi PDF-muodossa (eli sisällöstä generoidaan PDF)	2
Kattava dokumentaatio	5