

Jari Korhonen

Ajax-pohjaisen vedonlyöntikirjanpitosovelluksen toteutus

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Tietotekniikan koulutusohjelma
Insinöörityö
30.11.2011

Tekijä Otsikko	Jari Korhonen Ajax-pohjaisen vedonlyöntikirjanpitosovelluksen toteutus
Sivumäärä Aika	38 sivua 30.11.2011
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Ohjaaja	lehtori Olli Hämäläinen
<p>Insinöörityön aiheena oli selainkäyttöisen vedonlyöntikirjanpitosovelluksen toteutus. Tavoitteena oli suunnitella ja toteuttaa vedonlyöntiä harrastavan tarpeita vastaava apuväline, jolla olennaiset vetotiedot saisi kätevästi kirjattua ja jolla niitä voisi tarkastella monipuolisesti. Englanninkielinen sovellus sai nimekseen Bet Organizer.</p> <p>Työssä selvitettiin vedonlyönnin periaatteet kirjanpidon merkityksen kannalta ja perehdyttiin käytettävyyttä parantaviin tekijöihin, jotta toteutettavasta käyttöliittymästä saataisiin käyttäjäystävällinen ja helppokäyttöinen. Tavoitteena oli luoda sovelluksen ulkoasusta selkeä ja miellyttävä.</p> <p>Työ toteutettiin hyödyntäen Ajaxia, joka on kasvanut muutaman viime vuoden aikana hyvin suosituksi web-sovellusten kehitystavaksi. Ajax koostuu useista web-sovelluskehitystekniikoista, joista keskeisimmät käytiin työn teknisen osuuden selvityksessä läpi. Palvelinpuolen ohjelmointi toteutettiin PHP-ohjelmointikielellä ja tiedon tallennuspaikkana toimi MySQL-tietokanta.</p> <p>Lopputuloksena syntyneen käyttöliittymän toiminnot ja ulkoasu vastasivat suurelta osin määrittelyvaiheessa asetettuja vaatimuksia ja tavoitteita. Insinöörityön tekoaikana sovelluksesta saatiin kehitettyä valmiiksi kuitenkin vain melko rajoittunut perusominaisuudet sisältävä runko, jota tullaan jatkokehittämään usealla osa-alueella.</p>	
Avainsanat	Ajax, kirjanpito, käyttöliittymä, vedonlyönti, web-sovellus

Author Title	Jari Korhonen Implementation of Ajax-based sports betting bookkeeping application
Number of Pages Date	38 pages 30 November 2011
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Instructor	Olli Hämäläinen, Senior Lecturer
<p>The topic of this thesis was to design and implement a bookkeeping application for sports betting. The goal was to develop a tool that provides an easy and convenient way for bettors to keep book of bets they have placed and to view the collected bet data. A browser-hosted application was made in English, and it was named Bet Organizer.</p> <p>Principles of betting and the significance of bookkeeping in sports betting are covered in the thesis. Usability measurements were studied briefly to understand the key points in the development of user-friendly applications. The purpose was to create a clear and pleasant appearance for the bookkeeping application.</p> <p>Bet Organizer was implemented by taking advantage of Ajax which has gained high popularity in web development in the past few years. Ajax is a group of web development methods that are reviewed in the technical part of this document. The PHP programming language was used for server side programming and data was stored in the MySQL database.</p> <p>The functionality and appearance of the finished application corresponded largely to the goals and requirements that were set at the start of the project. However, only the basic functions of the application were completely finished during this thesis work, and many more features and improvements are to be added in the future.</p>	
Keywords	Ajax, bookkeeping, sports betting, user interface, web application

1 Johdanto	1
2 Tekniset lähtökohdat	2
2.1 Modernit internetsovellukset	2
2.2 Ajax ja käytettävyys	3
3 Vedonlyönti ja kirjanpito	5
3.1 Pelikohteiden valinta	5
3.2 Pelikassa	6
3.3 Vedonvälittäjät	7
3.4 Panoskoon määrittäminen	8
3.5 Kirjanpito	9
4 Määrittely ja tavoitteet	10
4.1 Toiminnallinen määrittely	10
4.2 Käytettävyystavoitteet	12
5 Käytetyt tekniikat ja työkalut	15
5.1 Ajax	15
5.1.1 XMLHttpRequest	16
5.1.2 Tiedon käsittely	20
5.1.3 JavaScript-kirjastot	22
5.2 Palvelinympäristö	23
5.2.1 PHP	23
5.2.2 MySQL-tietokanta	24
5.3 Kehitystyökalut	25
6 Vedonlyöntikirjanpitosovellus	26
6.1 Rekisteröityminen	26
6.2 Sisäänkirjautuminen	27
6.3 Salasanan vaihtaminen	27
6.4 Laskurisivu	28
6.5 Vetosivu	29
6.5.1 Kalenteri	29
6.5.2 Vetotietojen hakeminen	30
6.5.3 Vetojen hallinta	31
6.6 Asetussivu	34
7 Yhteenveto	36
Lähteet	37

Lyhenteet

Ajax	<i>Asynchronous JavaScript and XML</i> . Joukko selaimessa toimivia tekniikoita, joiden avulla luodaan vuorovaikutteisia web-sovelluksia.
DOM	<i>Document Object Model</i> . Ohjelmointirajapinta, jota käyttäen HTML-dokumentin sisältöä voidaan muokata esimerkiksi JavaScriptillä.
JSON	<i>JavaScript Object Notation</i> . Yksinkertainen ja kevyt tiedonsiirtomuoto, jota voidaan käyttää JavaScriptin lisäksi monissa muissa ohjelmointikielissä.
PHP	<i>PHP: Hypertext Preprocessor</i> . Ohjelmointikieli, jota käytetään erityisesti palvelinympäristöissä dynaamisten web-sivujen luonnissa.
RIA	<i>Rich Internet Applications</i> . Nimitys, jota käytetään runsaasti toiminnallisuutta sisältävistä työpöytäsovelluksista muistuttavista internetsovelluksista, joita voidaan toteuttaa muun muassa Ajaxilla, Javalla tai Adobe Flashilla.
SQL	<i>Structured Query Language</i> . Standardoitu relaatiotietokantojen hallintaan käytettävä kieli.
URL	<i>Uniform Resource Locator</i> . Kertoo internetissä olevan tiedon paikan, web-sivun osoite.
XHR	<i>XMLHttpRequest</i> . Ohjelmointirajapinta, joka mahdollistaa selaimessa ajettavan ohjelmointikielen muodostavan HTTP(S)-pyyntöjä palvelimelle.
XML	<i>eXtensible Markup Language</i> . Merkintäkieli, jolla kuvataan tiedon rakenne ja tyyppi sekä itse rakenteen sisällä oleva tieto. Sen avulla laajat tietokonekonaisuudet saadaan jäsenettyä.

1 Johdanto

Tämä insinööriyöraportti käsittää kuvauksen selainkäyttöisen vedonlyöntikirjanpitosovelluksen suunnittelusta ja toteutuksesta. Sovelluksen kehittämispäätös syntyi tarpeesta perinteistä taulukkosovelluskirjanpitoa monipuolisemmalle ja mielekkäämmälle tavalle arkistoida ja tarkastella vetotietoja.

Kirjanpito on olennainen osa vedonlyöntiharrastusta etenkin sellaisille henkilöille, joille se on satunnaista huvia säännöllisempi, voitolliseen tulokseen tähtäävä harrastus. Riittävän kirjanpidon tuloksena kerääntyvästä tiedosta vedonlyöjä saa ammennettua pitkällä aikavälillä hyödyllistä tietoa omasta pelitavastaan ja -menestyksestään. Näin vedonlyöjä kykenee hahmottamaan, minkälaisia pelikohteita on kannattavinta pelata voiton maksimoimiseksi.

Hyvin web-painotteiseksi kehittyneen vedonlyönnin apuvälineeksi päätettiin toteuttaa Ajax-tekniikoita käyttäen niin ikään selainpohjainen palvelimella toimiva kirjanpitotyökalu, joka nimettiin Bet Organizeriksi. Ohjelman avulla käyttäjä voi kirjata tiedot asettamistaan erityyppisistä vedoista ja tarkistella kerääntynyttä vetotietoa monipuolisesti. Tavoitteena oli luoda kirjanpitovaiheesta hyvin samankaltainen itse vedonlyöntitapahtuman kanssa, jotta kaikki olennainen vetotieto saataisiin mahdollisimman luonnollisesti kerättyä ja tallennettua tietokantaan. Sovellusta kehitettäessä kiinnitettiin erityistä huomiota käytettävyyttä parantaviin tekijöihin käyttäjäystävällisen käyttöliittymän aikaansaamiseksi.

Työssä selvitetään ensin lähtökohdat niin tekniikan kuin vedonlyönninkin kannalta. Tämän jälkeen käydään läpi sovelluksen määrittelyvaihe ja perehdytään sen toteuttamiseen käytettyihin tekniikoihin esimerkkeineen. Lopuksi otetaan käsittelyyn määritelmien pohjalta rakennetun sovelluksen käyttöliittymän esittely ja raportti päätetään yhteenvetoon, jossa pohditaan lopputulosta ja jatkokehitysmahdollisuuksia.

2 Tekniset lähtökohdat

2.1 Modernit internetsovellukset

Työpöytä- ja web-sovellusten väliset erot käyttötarkoituksessa ja -kokemuksessa ovat muutaman viime vuoden aikana muuttuneet häilyviksi, kun rikkaiksi internetsovelluksiksi (RIA, Rich Internet Applications) kutsutut sovellukset ovat yleistyneet. Tyypillisesti tällaiset sovellukset ovat nimensä mukaisesti selainkäyttöisiä ja sisältävät perinteisiin web-sovelluksiin nähden runsaammin toiminnallisuutta, vuorovaikutteisuutta ja mahdollisesti myös visuaalista antia, luoden näin käyttäjäkokemuksesta entistä paremman. [1, s. 5–7.] James Ward kuvaili blogissaan [2], kuinka RIA-sovelluksille ominaista on niiden miellyttävän luonnollisen tuntuinen käyttökokemus vertailemalla niiden ominaispiirteitä elävän elämän esimerkkeihin.

Teknisen toteutuksen osalta RIA-sovelluksille yhteistä on, että niiden toimintoja suoritetaan sekä paikallisesti käyttäjän puolella että palvelimella tarpeen mukaan. Palvelimelta ladataan kerrallaan pienempiä tietomääriä käsiteltäväksi kuin perinteisissä internetsovelluksissa, mutta tyypillisesti useammin. Käyttöliittymän toiminnallisuuksien siirtäminen käyttäjän tietokoneen suoritettavaksi mahdollistaa runsaan ja reaaliaikaisen kommunikaation käyttäjän ja sovelluksen välille, ja toimintojen yhteyteen voidaan saadun lisätehon turvin lisätä visuaalisia parannuksia kuten animaatioita. [1, s. 7–8.]

Modernien internetsovellusten kehittämiseen käytetään muun muassa Adoben Flash-tekniikkaa, Javaa, jonka kehityksestä vastaa nykyisin Oracle, tai Ajaxia, joka koostuu useiden eri web-tekniikoiden yhteiskäytöstä. Näistä kaksi ensiksi mainittua ovat selaimesta erillään toimivia kokonaisuuksia, jotka voidaan liittää osaksi selaimen näkymää tai käynnistettäväksi sen kautta. Ne siis vaativat omat ajoympäristönsä, Flash Playerin tai vastaavasti Java Runtimen, kun Ajax puolestaan toimii selaimen prosessoimassa JavaScriptissä. Flash ja Java ovat Ajaxia raskaampia tekniikoita, joten niillä toteutetut ratkaisut latautuvat yleensä hitaammin ja vaativat sulavasti toimiakseen enemmän puhtia prosessorilta. Vastapainoksi ne mahdollistavat visuaalisesti näyttävämpien sovellusten kehittämisen. [3.] Työssä keskitytään tästä eteenpäin vedonlyöntikirjanpitosovelluksen toteutustavaksi valitun Ajaxin toimintaan.

Ajaxin kehityskaaren alkutaipaleella sitä pidettiin liian vaikeana ja monimutkaisena tapana toteuttaa laajoja internetsovelluksia. Lukuisten sen suosion kasvun myötä kehitettyjen apuvälineiden ansiosta Ajax on nykyään yksi helpoimmista ja kustannustehokkaimmista tavoista luoda dynaamisia web-sovelluksia. Tuloaan tekevä HTML5-standardi mahdollistaa entistä näyttävämpien JavaScriptiin pohjautuvien käyttöliittymien toteuttamisen, kun niihin on mahdollista upottaa videokuvaa ja muuta mediaa.

Kuten edellä todettiin, Ajaxia voidaan hyödyntää runsaasti toiminnallisuutta vaativien web-sovellusten rakentamisessa. Tunnetuimpia Ajaxin käyttökohteita ovat useat Googlen kehittämät sovellukset, kuten karttapalvelu Google Maps, joka hakee huomaamattomasti tietoa palvelimelta kartan päivittämiseksi, ja Google Docs, joka soveltuu erinomaisesti ryhmätyönä tehtävien projektien dokumentointityökaluksi. Erityisen yleistä Ajaxin käyttö on erilaisissa hakutoimintoja sisältävissä käyttöliittymissä, kuten verkkokaupoissa tai tietoarkistoissa. Esimerkiksi Googlen hakukone listaa ja rajaa haulilla löydettyjä tuloksia suorittaen jatkuvasti Ajaxilla tietokantahakuja, kun käyttäjä muuttaa hakukentän sisältöä kirjoittamalla.

2.2 Ajax ja käytettävyys

Ajaxin tarpeellisuus käytettävyyden parantamisessa korostuu, kun rakennettava käyttöliittymä muodostuu suureksi ja monimutkaiseksi. Tilanteissa, jolloin on tarpeen päivittää vain pieniä osia näkyvästä tietomäärästä, tarvittavan tiedon haku palvelimelta hoi-tuu ripeästi, eikä käyttäjä joudu odottelemaan turhaan. Tarvittaessa käyttöliittymän näkymää saadaan päivitettyä myös reaaliaikaisesti, kuten Google Mapsin tapauksessa.

Ajax tuo mukanaan myös joitakin käytettävyyttä koskevia haittapuolia, mistä syystä kehittäjältä vaaditaan perehtymistä muutamiin käytettävyystekijöihin. Olennaisimmat esille tulleet käytettävyysongelmat liittyvät siihen, ettei osoiterivin sisältö (URL) oletusarvoisesti vastaa sovelluksen senhetkistä tilaa eli sitä, mitä sivulla näytetään. Tämä taas johtaa siihen, ettei juuri sillä hetkellä näkyvää sisältöä voi lisätä kirjanmerkkeihin tai linkittää muille käyttäjille, eivätkä selaimen toimintopainikkeet edelliselle ja seuraavalle sivulle siirtymiseen toimi. Käytettävyysasiantuntija Jakob Nielsen suhtautuikin tästä syystä Ajaxiin hyvin negatiivisesti sen yleistymisen kynnyksellä muutama vuosi sitten; paluu-painikkeen napsautus kun oli käytettävyystutkimusten mukaan eräs yleisimmistä

web-sovelluksen käyttötapauksista, ja epäilemättä on sitä edelleen [4].

Edellä mainitut ongelmat saa korjattua ohjelmoimalla sovelluksen tekemään automaattisesti muutoksia URL:iin sen mukaan, mitkä osat sivusta on ladattu näytettäväksi. Myös lukuisia tästä huolehtivia JavaScript-lisäosia on saatavilla. Uuteen HTML5-standardiin on niin ikään lisätty JavaScript-komponentit istunnon historian ja sovelluksen tilan säilyttämiseksi, jolloin lisäosille ei ole enää tarvetta [5].

Perinteisestä internetsovelluksesta poiketen Ajax hoitaa keskustelun palvelimen kanssa näkymättömästi, joten käyttäjä ei saa tietoa sovelluksen tapahtumista. Esimerkiksi, jos yhteys palvelimeen sattuu katkeamaan toviksi, ei sitä ole nähtävissä sovelluksen käyttäjäliittymästä millään tavalla. Jää ohjelmoijan tehtäväksi huolehtia, että käyttäjälle näytetään, mitä osaa sisällöstä ollaan kulloinkin lataamassa, tai että virhetapauksessa siitä myös tulostetaan ilmoitus. Aktiivisuuden ilmoittamiseksi voidaan käyttää pieniä animoituja kuvakkeita, ja hiiren kohdistin voidaan muuttaa osoittamaan, että tiedonsiirto on parhaillaan käynnissä. [6, s. 336–340.]

Ajaxilla toteutettujen käyttäjäliittymien sisältämä runsas toiminnallisuus ja valintojen mahdollisuuksien määrä voi johtaa ei-toivottujen käyttötapahtumien todennäköisyyden kasvuun. Nielsenin tutkimusten mukaan [7] esimerkiksi useiden Ajaxilla toteutettujen verkkokauppojen ostoskoritoiminnot aiheuttivat totutusta poikkeavalla toimintatavallaan sekavuutta, ja käyttäjiltä saattoi jäädä huomaamatta pienemmät käyttäjäliittymässä tapahtuvat muutokset. Käytettävyyssasioita pohditaan lisää käytettävyyssuositteissa luvussa 4.

3 Vedonlyönti ja kirjanpito

Nykyaikana internet tarjoaa erinomaiset puitteet vedonlyönnin harrastamiseen itse vedonlyöntitapahtumien laajan tarjonnan lisäksi myös sieltä saatavan vedonlyöntiä koskevan tiedon vuoksi. Vedonlyöntiharrastajien yhteisöt tarjoavat mahdollisuuden hyödyllisen ja ajankohtaisen vetokohteita koskevan tiedon saamiseen. Esimerkiksi vetokohteille pohtimiaan todennäköisyyksiä voi vertailla asiantuntijoiksi osoittautuneiden vihjaajien arvioiden kanssa saadakseensa niistä paremman käsityksen.

Kirjanpidon merkitys vedonlyönnissä on paremmin ymmärrettävissä, kun on tietoinen minkälaisista tekijöistä vedonlyönti muodostuu. Vedonlyöntitapahtumien analysoimiseksi hyödynnetään tilastomatematiikkaa ja todennäköisyyslaskentaa. Vaikka vedonlyönnissä on lukuisista epävarmuustekijöistä johtuen erittäin vaikea tehdä tarkasti paikkansapitäviä arvioita, siihen tarpeeksi perehtymällä on kuitenkin todennäköistä päästä voitolliseen tulokseen pitkällä aikavälillä.

3.1 Pelikohteiden valinta

Voitolliseen vedonlyöntiin pyrittäessä on olennaista pelata sellaisia kohteita, joiden voitotapahtuman odotusarvo on yli yksi. Tällaisia kohteita kutsutaan ylikertoimiksi, sillä tällöin tarjottu kerroin, eli voiton tapauksessa palautuvan rahan suhde panostettuun rahamäärään, on suurempi kuin sen vedon osumistodennäköisyyden puolesta pitäisi olla. Odotusarvo muodostuu vedonvälittäjän kohteelle tarjoaman kertoimen ja vedonlyöjän kohteelle arvioiman osumistodennäköisyyden tulona. Luotettavan osumistodennäköisyyden arvioimisen perustana on syvä perehtyneisyys kohteen aihepiiriin.

Esimerkiksi jalkapallo-ottelun todennäköisyyksien muodostumiseen vaikuttavat muun muassa joukkueiden pelaajamateriaalien laatu suhteessa toisiinsa, kotipelin merkityksen suuruus kotijoukkueelle, vierasjoukkueen viimeaikaiset esiintymiset vieraskentillä, mahdolliset motivaatioedut esimerkiksi toisen joukkueen taistellessa putoamista vastaan toisen ollessa keskikastissa vailla mahdollisuutta parempiin sijoihin, ottelutahdin senhetkinen tiheys, mahdolliset lähiaikoina pelattavat tärkeämmät ottelut sekä joukkueiden loukkaantumistilanteet. Mitä enemmän asioita tutkii ja mitä ajankohtaisempaa tietoa kohteesta saa kaivettua tietoonsa, sitä todennäköisemmin käsittää joukkueiden

voimasuhteet kertoimen tarjoavaa vedonvälittäjää paremmin.

Ylikertoimien pelaamisen merkitystä pitkällä aikavälillä voidaan havainnollistaa kolikonheittoesimerkillä. Jokaisella heitolla klaavan todennäköisyys on tunnetusti 0,5, ja mikäli klaavalle tarjotaan kertoimeksi 2,2, muodostuu tapahtuman toteutumisen odotusarvoksi 1,1 ja pelikohde on siis ylikerroin. Toistuvasti tällaista kohdetta pelaamalla voi odottaa saavansa keskimäärin kymmenen prosentin tuoton sijoitetulle rahalle. Voitolla olemisen todennäköisyys kasvaa toistokertojen kasvun myötä, sillä klaavojen ja kruunien esiintymiskertojen suhde lähenee suurten lukujen lain [8] mukaisesti lukua yksi, ja jokaisella voitollisella heitolla etu on ylikertoimen pelaajalla. Pienellä otannalla, esimerkiksi kymmenen kolikonheitto-kerran jälkeen, ei voida vielä sanoa, ollaanko voitolla vai tappiolla, sillä seitsemän ensimmäisen heiton tuloksena voi olla kruuna. Samalla periaatteella urheiluviedonlyöntimenestys on määriteltävissä vasta pitkällä aikavälillä kerätyn tiedon perusteella, eikä yksittäisten kohteiden todennäköisyysarvioiden oikeellisuuksia voida tarkasti määritellä.

Vetokohteista tarjotaan usein monia erilaisia versioita esimerkiksi tasoitusvetojen muodossa. Etukäteen heikommalle joukkueelle tai kilpailijalle voidaan antaa etumatkaa, jolloin altavastaajan kertoimet pienenevät normaaliin vetoon nähden, vastaavasti suosikin kerroin on tällöin suurempi. Vedonlyöjä voi siis tasoitusvetokohteen valitsemalla joko suurentaa tai pienentää vedon osumistodennäköisyyttä saaden näin pienemmän tai vastaavasti suuremman kertoimen. Tyypillisissä vedoissa on kaksi lopputulosta: veto on joko häviävä, eikä palauta yhtään rahaa, tai se osuu, eli palauttaa rahaa kertoimen ja panoksen tulon verran. Erilaiset tasoitusvedot lisäävät monimutkaisempia lopputilanteita, joissa vedon voi esimerkiksi hävitä tai voittaa puolittain.

3.2 Pelikassa

Vedonlyöntiä varten muodostetaan pelikassa, jonka koko riippuu siitä, minkä suuruisen osan omaisuudestaan vedonlyöjä on halukas omistamaan pelkästään vedonlyönnille. Pelikassa muodostaa erillisen osion, joka on riippumaton muihin asioihin käytetystä rahasta. Tällöin vedonlyöjälle ei synny paineita ja pakollista tarvetta saada voittoja esimerkiksi tappiollisen ajanjakson aikana, kun rahan voi ajatella olevan ylimääräistä. Pelaajan kannattaa asettaa riskien pienentämiseksi maksimipanokselle maltillinen raja,

esimerkiksi 5 % pelikassan koosta. [9.]

3.3 Vedonvälittäjät

Vedonlyöntiä harrastavalla on hyvä olla pelitili usealla eri vedonvälittäjällä, jolloin mahdollisuus kannattavaan pelaamiseen kasvaa, kun laajasta tarjonnasta saa poimittua kohteille tarjotut suurimmat kertoimet, ja täten mahdollisuus ylikertoimisten kohteiden pelaamiseen kasvaa. Vedonvälittäjien tarjoamien kertoimien vertailu onnistuu kätevästi esimerkiksi Oddsportal.com-sivustolla. Vedonvälittäjien laatua voidaan mitata muun muassa palautusprosentin suuruuden, tarjonnan laajuuden sekä tarjottujen bonusten perusteella. Vedonvälittäjien antamien kertoimien määrittelemä palautusprosentti vaihtelee, ja on yleensä 90–95 %. Olkoon esimerkiksi vedonvälittäjän tarjoamat kertoimet kolme mahdollisuutta sisältävään kohteeseen 2,1 – 3,9 – 2,8. Näiden käänteislukujen summan ($0,476 + 0,256 + 0,357 = 1,089$) käänteisluvuksi saatu 0,918 ilmoittaa palautusprosentin, joka tässä tapauksessa on siis hieman alle 92 %. Vedonvälittäjien tarjoamat bonukset voivat olla esimerkiksi ilmaisia vetoja tai talletusbonuksia, joilla pelikassaa on mahdollista kasvattaa riskittömästi.

Vedonvälittäjän avaaman kohteen kertoimet muotoutuvat tyypillisesti ajan myötä vastaamaan suuren pelaajajoukon käyttäytymistä. Toisinaan suurten ennakkosuosikkien kertoimet laskevat niiden suosion myötä selvästi alemmalle tasolle kuin niitä vastaava oikea todennäköisyys olisi, ja tällöin altavastaajan kerroin nousee erittäin maukkaaksi ylikertoimeksi.

Toisinaan vedonvälittäjien arviot pelikohteesta poikkeavat toisistaan siinä määrin, että vedonlyöjälle tarjoutuu mahdollisuus ansaita ”varmaa rahaa”, kun esimerkiksi yksi vedonvälittäjä tarjoaa merkkipaihtoehdot 1 ja 2 sisältävään kohteeseen merkille 1 kerrointa 2.1 toisen vedonvälittäjän tarjotessa samaa kerrointa merkille 2. Tällöin asettamalla molemmille merkeille samansuuruisen panoksen saa sijoitukselleen viiden prosentin tuoton riippumatta lopputuloksesta. Tällaisia vetoja kutsutaan arbitraaseiksi. Täysin riskittömiä eivät nekään ole, sillä selvästi virheellisen kertoimen tarjotessaan vedonvälittäjä voi mitätöidä asetetun vedon, jolloin vain toiselle vedonvälittäjälle asetettu arbitraasin osa jää voimaan toisen osan siis palauttaessa panoksen takaisin. Mikäli voimaan jäänyt osa on häviävä, voi vedonlyöjä tehdä runsaastikin tappiota.

3.4 Panoskoon määrittäminen

Vedonlyöntiin on olemassa erilaisia laskukaavoja, joiden avulla pyritään parantamaan sen kannattavuutta. Todennäköisyysteoriaa tutkinut John L. Kelly jr. [10] kehitti teorian, jota hyödyntäen voidaan optimoida vetoikohteisiin sijoitettava panoskoko niin, että pelikassan kasvu saadaan maksimoitua. Kellyn kaava määrittelee, kuinka suurella osuudella pelikassasta kohdetta tulisi pelata riippuen sille tarjotun kertoimen ja arvioidun osuistodennäköisyyden suuruuksista. Yksinkertaisille vedoille, joissa on ainoastaan mahdollisuus joko voittaa kertoimen ja panoksen määrittelemä summa tai hävitä koko panos, on Kellyn kaava muotoa

$$\frac{kp-1}{k-1}$$

jossa k on tapahtumalle tarjottu kerroin ja p tapahtumalle arvioitu todennäköisyys. Olkoon esimerkiksi vedonvälittäjän tarjoama kerroin merkille 1 kohteeseen Jokerit – HIFK 2,50, ja vedonlyöjän merkille 1 arvioima osuistodennäköisyysprosentti merkille 42 %. Tapahtuman odotusarvoksi saadaan $2,50 \times 0,42 = 1,05$, joten kohde on pelattavissa. Kellyn kaavan mukainen optimaalinen panostus kohteeseen on noin 3,3 % pelikassasta.

Rahamääräisesti ilmoitettuna panoksen suuruus on pelikassan koko kerrottuna kaavan antamalla prosentuaalisella osuudella. Laskussa käytettävän pelikassan koon tulee olla dynaaminen eli senhetkisen tilanteen mukainen; edellisten vetojen aiheuttamat pelikassan suuruuden muutokset otetaan huomioon. Pienikertoimisiin eli useammin osuviin vetoihin on Kellyn kaavan mukaisesti kannattavaa panostaa enemmän kuin suurempikertoimisiin eli harvemmin osuviin. Kellyn kaavan antama panostussuositus vaihtelee hyvin voimakkaasti syötetyn todennäköisyyden mukaan, ja tavallista onkin panostaa vähemmän aggressiivisesti jakamalla kaavan antama tulos esimerkiksi neljällä, jolloin mahdollinen virhearvio ei koidu niin tappiolliseksi. [9.]

3.5 Kirjanpito

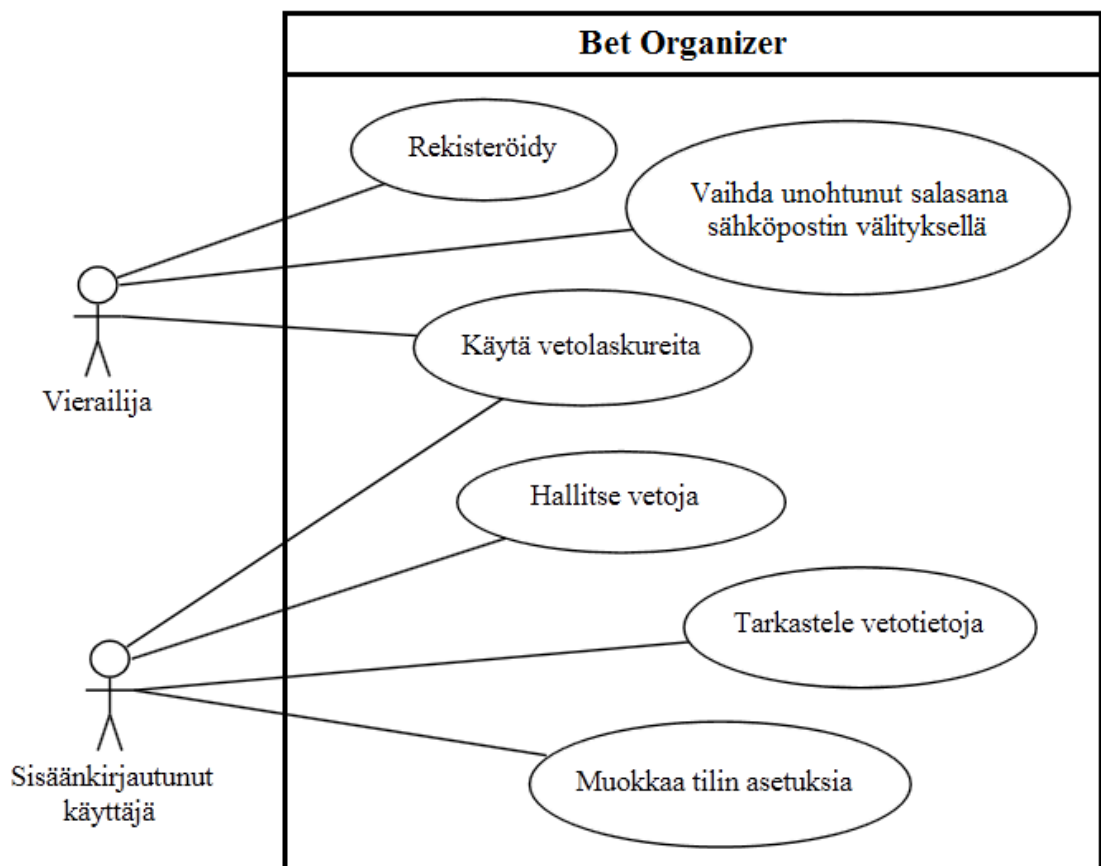
Vedonlyöntikirjanpidon avulla asetettujen vetojen kokonaistilanne on helposti hahmotettavissa, kun eri vedonvälittäjille hajautetuista vedoista saa koostettua tiedot samaan paikkaan. Merkittävin kirjanpidosta nähtävä asia on palautusprosentti eli voitettun rahamäärän suhde kokonaisuudessaan panostettuun rahamäärään. Näin myös pelikassan tarkka koko on aina ajan tasalla, ja oikeansuuruisen panoskoon määrittäminen Kellyn kaavan avulla on mahdollista. Kustakin vedosta kannattaa kirjata ainakin panos, pelimuodon tyyppi, pelikohteen tiedot, kerroin, arvioitu osumistodennäköisyys, vedonvälittäjä ja tulos. Myös erityisten lopputulokseen vaikuttaneiden seikkojen kirjaaminen voi olla hyödyllistä.

Riittävän kirjanpidon tuloksena kerääntyvästä tiedosta vedonlyöjä saa ammennettua pitkällä aikavälillä hyödyllistä tietoa omasta pelitavastaan ja -menestyksestään. Vedonlyöjä voi runsaan otannan ja tilastojen pohjalta hahmottaa, minkälaiset pelikohteet ovat olleet kannattavimpia ja vastaavasti minkälaiset kohteet eivät ole pitkällä aikavälillä tuottoisia. Näin pelaajan harkintakyky ajan myötä paranee ja pelitapa muovautuu tuottoa ajatellen varmempaan suuntaan.

4 Määrittely ja tavoitteet

4.1 Toiminnallinen määrittely

Bet Organizer on palvelimella toimiva vedonlyöntikirjanpitosovellus, jonka avulla käyttäjä voi kirjata olennaiset tiedot asettamistaan vedoista sekä hallita ja tarkastella vetotietoa monipuolisemmin ja kätevämmän kuin perinteisellä taulukko-ohjelmapohjaisella kirjanpidolla. Kirjanpitosovellusta käytetään internetselaimella, ja sen toimintoihin käsiksi pääseminen vaatii rekisteröitymisen. Kuvassa 1 on esitettyä sovelluksen käyttötapauskaavio korkealla tasolla kuvattuna.



Kuva 1. Bet Organizerin käyttötapauskaavio.

Käyttötapauskaavion pohjalta laadittiin tarkemmat toiminnalliset vaatimukset, jotka on esitettynä taulukossa 1.

Taulukko 1. Toiminnalliset vaatimukset.

ID	Toiminto	Kuvaus	Prioriteetti
T1	Rekisteröityminen	Käyttäjä voi avata uuden tilin rekisteröitymällä.	1
T2	Kirjautumiset	Käyttäjä voi kirjautua sisään ja ulos järjestelmästä.	1
T3	Vetojen hallinta	Käyttäjä voi lisätä uusia vetoja, muokata ja poistaa aiemmin kirjattuja vetoja sekä kopioida vetoja uusien pohjaksi.	1
T4	Kalenterinavigointi	Käyttäjä voi navigoida päivien välillä kuukausikalenteria käyttäen. Kalenterin kuukausia ja vuosia voi vaihtaa erikseen.	1
T5	Vetotietojen tarkastelu	Käyttäjä voi tarkastella vetotietoa päiväkohtaisesti T4:n mukaista kalenteria käyttäen, etsiä vetoja sanahakua käyttäen, tarkastella keskeneräisiä vetoja erikseen sekä nähdä koostetun tiedon haetuista vedoista kussakin tapauksessa.	1
T6	Asetusten muuttaminen	Käyttäjä voi muuttaa henkilökohtaisia asetuksiaan, mm. asettaa pelikassan koon, lisätä ja valita käyttämänsä vedonvälittäjät asetuksista sekä vaihtaa salasanansa.	1
T7	Salasanan vaihtaminen sähköpostin välityksellä	Käyttäjä saa uudelleenasetettua salasanansa sen unohtumistapauksessa käyttämällä rekisteröitymisen yhteydessä antamaansa sähköpostiosoitetta.	1
T8	Laskurit	Käyttäjä voi tehdä yleisimpiä vedonlyönneissä tarvittavia laskutoimituksia käyttämällä laskureita.	1
T9	Graafiset tilastot	Käyttäjä näkee koostettua vetotietoa myös graafisessa muodossa.	2
T10	Vetotietojen jakaminen	Käyttäjä voi jakaa haluamaansa tietoa kirjaamistaan vedoista muille käyttäjille.	2

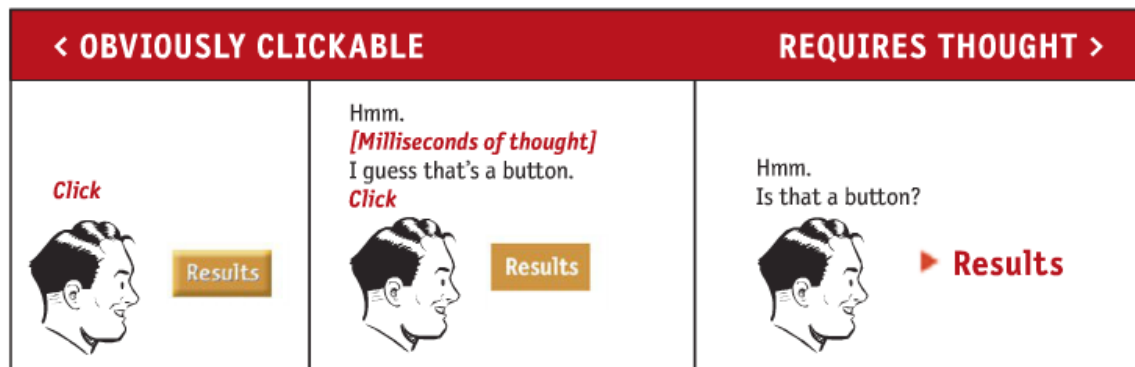
4.2 Käytettävyystavoitteet

Vedonlyöntikirjanpitosovelluksen kehityksessä kiinnitetään huomiota käytettävyyttä parantaviin tekijöihin käyttäjäystävällisen lopputuloksen aikaansaamiseksi. Käytettävyydellä mitataan, kuinka tuottavaa, tehokasta ja miellyttävää jonkin tuotteen käyttö on. Seuraava lainaus on mielestäni erittäin osuva sellaiselle järjestelmälle, jonka käytettävyys on hyvä:

The user can do what he or she wants to do the way he or she expects to be able to do it, without hindrance, hesitation, or questions. [11, s. 4]

Jakob Nielsen on määritellyt internetsovelluksen käytettävyyden mittaamiseksi viisi eri komponenttia: opittavuus, tehokkuus, muistettavuus, virheiden estäminen ja miellyttävyys [12, s. 120-121]. Nielsen on myös esittänyt kymmenen käyttöliittymäsuunnittelun periaatetta [13], joita voidaan käyttää ohjenuorana paremman käyttäjäystävällisyyden saavuttamiseksi. Seuraavaksi käydään lyhyesti läpi käytettävyyden muodostavat seikat pohtien niiden merkitystä Ajax-sovelluksen toteutuksessa.

Opittavuus kuvaa, kuinka helposti käyttäjä oppii tunnistamaan ja käyttämään sovelluksen eri toimintoja käyttäessään sitä ensimmäistä kertaa. Ajaxilla toteutettujen käyttöliittymien sisältämä runsas toiminnallisuus ja valintojen määrä voivat johtaa sovelluksen käytön opittavuuden vaikeutumiseen. Järjestelmä kannattaa rakentaa siltä osin kuin se mahdollista on, tavalla, johon käyttäjä on jo tottunut käyttäessään samantyyppisiä sovelluksia, jotta sen käyttäminen on intuitiivista ja oppiminen nopeaa. Kuvan 2 esimerkki valaisee ulkoasuvalinnan merkitystä havainnollisuuden lisäämisessä.



Kuva 2. Toimintopainikkeen esitystapa vaikuttaa opittavuuteen. Myös hiiren kohdistimen muuttaminen erilaiseksi painikkeen päällä lisää selkeyttä. [14, s. 15]

Turhan omaperäisiä käytettävyyssratkaisuja on hyvä välttää, ja monet tutuiksi muodostuneet käyttötapaukset, kuten lomaketoiminnot, kannattaa toteuttaa standardien mukaisesti. Samantyyppisiä asioita varten toteutettujen toimintopainikkeiden koon, muodon ja värin pysyminen yhteneväisenä läpi järjestelmän helpottaa toimintojen hahmottamista. Tarpeen vaatiessa toimintojen yhteyteen voidaan lisätä myös ohjeistusta. [12, s. 120; 13.]

Tehokkuudella mitataan, missä ajassa käyttäjä voi suorittaa jo oppimiansa toimintoja. Ajaxia hyödyntämällä on perinteisiin websovelluksiin verrattuna mahdollista parantaa tehokkuutta, kun tarpeellista sisältöä saadaan näytettyä dynaamisesti ilman sivulatauksia. Tämän ansiosta myöskin sovelluksen muistettavuutta, eli sitä, millä tehokkuudella käyttäjä kykenee suorittamaan aiemmin oppimansa toiminnot käyttötaun jälkeen, on joissakin tilanteissa mahdollista parantaa, kun käyttötapauksista saadaan entistä suora- viivaisempia. Jos sovellus ilmoittaa toimintojensa käyttötarkoitukset riittävän selkeästi, vähenee käyttäjän tarve muistaa asioita, ja sujuva käyttäminen onnistuu yleisellä päätelykyvyllä. [12, s. 120–121; 13.]

Interaktiivinen sovellus voi antaa monenlaista palautetta toiminnastaan estääkseen virheitä ja pitääkseen käyttäjän tietoisena tapahtumien kulusta. Esimerkiksi pidempiaikaisemman tiedonsiirron yhteydessä on paikallaan esittää edistymistä kuvaava palkki tai yleinen lataustoimintoa kuvaava animaatio. Ennen kuin virheistä ilmoitetaan, on hyvä pyrkiä siihen, ettei niitä ylipäätään pääse syntymäänkään. JavaScriptin avulla käyttäjälle on mahdollista antaa välitöntä palautetta ja estää ei-toivottujen tapahtumien synty-

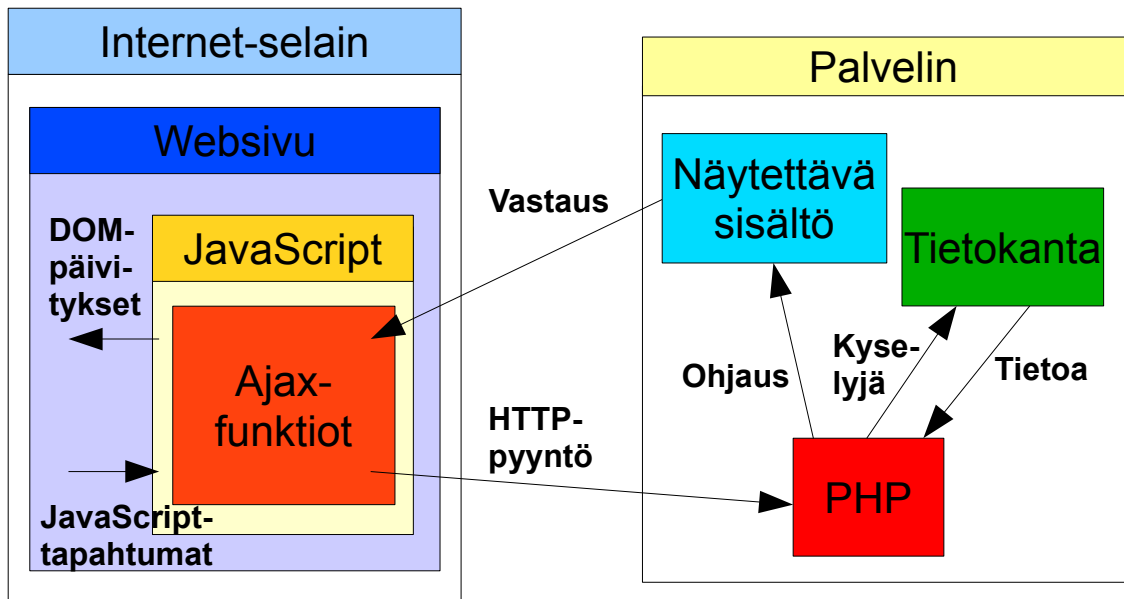
minen lennossa, mikä auttaa mahdollisissa epäselvissä tilanteissa etenemään sujuvasti. Huonona puolena JavaScriptin runsas käyttö Ajax-sovelluksessa saattaa aiheuttaa itsessään virheitä, joiden syntymisen vähentämiseksi koodia yksinkertaistavien JavaScript-kirjastojen käyttö on suositeltavaa. [12, s. 120; 13.]

Miellyttävä käyttöliittymä on mahdollisimman yksinkertainen ja selkeä, minkä ansiosta navigointi haluttuun paikkaan on helppoa. Olennainen tieto on paremmin esillä ja vastaavasti kullakin hetkellä vähemmän tärkeät asiat jäävät taustalle eivätkä vedä käyttäjän huomiota puoleensa. Muun muassa sovelluksen värimaailmalla ja fonttikoolla on suuri merkitys miellyttävyyden luomisessa.

Kirjanpitosovelluksen käyttöliittymän toteutuksessa pyritään ratkaisemaan kohdassa 2.3 kuvatut käytettävyysongelmat ja noudattamaan tässä luvussa pohdittuja käytettävyysoasioita, jotta sovelluksen käyttäjäystävällisyys saataisiin hyvälle tasolle. Vedonlyöntikirjanpitosovelluksen käyttöliittymästä on tarkoitus tehdä mahdollisimman selkeä ja nopeasti omaksuttava sijoittelemalla ja ryhmittelemällä toiminnot loogisesti. Ulkoasusta on pyrkimys saada miellyttävän yksinkertainen ja toimintoja tukeva, jotta kokonaisuus on helposti hahmotettavissa.

5 Käytetyt tekniikat ja työkalut

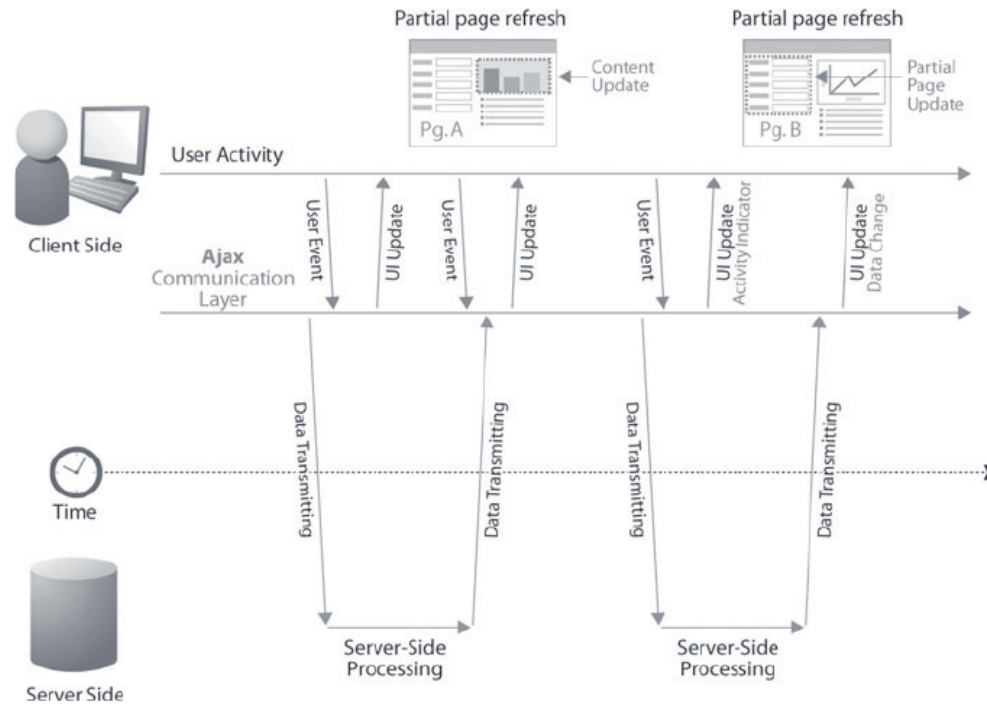
Tässä luvussa käydään läpi kirjanpitosovelluksen toiminnasta vastaavat tekniikat ja kerrotaan esimerkkejä apuna käyttäen, kuinka ohjelma niitä hyödyntää. Kuva 3 havainnollistaa Ajax-tekniikoilla toteutetun sovelluksen toimintaperiaatetta kuvaten sen eri komponenttien välistä kommunikaatiota.



Kuva 3. Ajax-sovelluksen toimintaperiaate.

5.1 Ajax

Ajax (Asynchronous JavaScript and XML) mahdollistaa työpöytäsovellusmaisten web-sovellusten toteuttamisen. Sitä hyödyntäen vuorovaikutteisuus voi olla runsasta, kun selaimen ja palvelimen välinen tiedonsiirto onnistuu ripeästi halutun suuruisina palasina, ilman että koko sivua on tarve ladata uudelleen kokonaisuudessaan. Selaimessa JavaScriptissä toimiva Ajax muodostaa perinteisiin internetsovelluksiin nähden yhden lisäkerroksen käyttäjän ja palvelimen välille, kuten kuvasta 4 on nähtävissä. Ajax-sovelluksessa käyttäjän toiminnot eivät välttämättä ohjaa palvelimen toimintaa suoraan, vaan JavaScript-kerroksessa määritetään, miten tiedonsiirto kullakin hetkellä tapahtuu.



Kuva 4. Ajax-sovelluksen toiminta ajan funktiona. [6, s. 4]

Ajaxin toiminta perustuu useiden eri tekniikoiden yhteiskäyttöön. Keskeisimmät Ajaxin mahdollistavat tekniikat itse JavaScriptin ohella ovat tiedonsiirrosta huolehtiva XMLHttpRequest-objekti, esitettävän tiedon muokkaamisen mahdollistava DOM-rajapinta sekä tiedon esittämiseen käytettävät XML ja (X)HTML.

5.1.1 XMLHttpRequest

XMLHttpRequest-objekti, lyhyemmin XHR, on ohjelmointirajapinta, joka mahdollistaa selaimessa ajettavan skriptauskielen, kuten Ajaxin tapauksessa JavaScriptin, muodostavan HTTP(S)-pyyntöjä palvelimelle. Objektiin nimessä esiintyvän XML-tietomuodon lisäksi tietoa on mahdollista lähettää palvelimelle monella muullakin tarkoituksenmukaisella tavalla ohjelmoijan tarpeiden mukaan. Käyttörajoituksena se mahdollistaa pyyntöjen tekemisen ainoastaan samaan verkko-osoitteeseen, josta sitä on kutsuttu, mikä on tietoturvan kannalta hyvä asia, sillä haitallisen koodin ujuttaminen sivustolle on näin vaikeampaa. [12, s. 16.]

Koodiesimerkki 1 sisältää funktion, jolla luodaan XMLHttpRequest-objekti. Ensin se pyritään luomaan modernien selainten, kuten Mozilla Firefoxin tai Google Chromen, toimintaa vastaavalla tavalla, ja tämän epäonnistuesssa Internet Explorer 5 - ja 6 -selaimilla toimivaksi käyttäen Microsoftin ActiveX-objektia. Lopuksi objekti palautetaan, mikäli sen luonti onnistui. Muussa tapauksessa näytetään varoitusikkuna, joka ilmoittaa epäonnistuneesta luontiyrityksestä.

```

1 function createRequestObject() {
2
3     var requestObject = null;
4     try {
5         // Suoritetaan, kun käytössä on moderni selain, josta
6         // XMLHttpRequest-objekti löytyy valmiina.
7         requestObject = new XMLHttpRequest();
8     } catch (e) {
9         try {
10            // Käytössä saattaa olla vanha selain, kuten IE6,
11            //joka käyttää ActiveX-objektia.
12            requestObject = new ActiveXObject("Microsoft.XMLHTTP");
13        } catch (e) {
14            alert("XMLHttpRequest object could not be created.");
15        }
16    }
17
18    if (!requestObject) {
19        // Pyynnön luominen ei onnistunut.
20        alert("XMLHttpRequest object could not be created.");
21    } else {
22        // Pyynnön luominen onnistui, palautetaan objekti.
23        return requestObject;
24    }
25 }

```

Koodiesimerkki 1. XMLHttpRequest-objektin luominen.

Tärkeimmät metodit

XMLHttpRequest-objektin olennaisimmat metodit ovat *open()* ja *send()*. Pyyntöä valmistelevalle *open*-funktiolle voidaan antaa seuraavanlaiset viisi parametria, joista kaksi ensimmäistä ovat pakollisia: pyynnön tyyppi, tiedon sijainti palvelimella (URL), pyynnön mahdollinen asynkronisuus, käyttäjätunnus ja salasana.

Yleisimmät pyyntötyypit ovat *GET* ja *POST*, joista ensimmäistä käytetään, kun siirrettävän tiedon yhteyteen ei liity toiminnallisuutta, esimerkiksi korvattaessa jonkin elemen-

tin sisältö palvelimelta saatavalla uudella sisällöllä. GET-metodia käytettäessä URL-parametriin sisällytetään palvelimella olevan tiedostonimen lisäksi myös lähetettävät tiedot. HTTP-standardin mukaisesti GET-pyyntöjä ei tule käyttää pysyvien muutosten tekemiseen eikä tietoja tallenneta selaimen välimuistiin. POST-pyyntöillä käsitellään tyypillisesti erilaiset toiminnot, kuten lomaketietojen lähetys tai palvelimella sijaitsevan tiedon muokkaaminen. [6, s. 63; 12, s. 74.]

Välimuistin käytöstä poistamisen pakottamiseksi GET-pyyntöjä käsittelevän PHP-tiedoston alkuun voidaan lisätä seuraavat HTTP-tunnistetta muuttavat rivit, joista jälkimmäiseen vanhentumisajankohdaksi asetetaan jokin menneisyyden ajankohta [15]:

```
header("Cache-Control: no-store, no-cache, must-revalidate");  
header("Expires: Mon, 01 Jan 1990 05:00:00 GMT");
```

Open-metodin kolmas parametri kertoo, käsitelläänkö lähetettävä pyyntö asynkronisesti vai ei. Mikäli parametria ei määritellä, käytetään oletusarvoa eli asynkronista lähetystä. Viimeiset parametrit, käyttäjä ja salasana, voidaan antaa, mikäli URL:n osoittama tieto vaatii tunnistautumisen, jolloin käyttäjän ei tarvitse syöttää tunnuksia erikseen.

Send-metodi lähettää pyynnön palvelimelle. Mikäli metodia kutsutaan asynkronisesti, eli open-funktion kolmannen parametrin arvo on TRUE tai sitä ei ole määritelty, kutsu suoritetaan välittömästi riippumatta pyynnön vaiheesta. Käyttäjän näkökulmasta tämä tarkoittaa, että pyyntö suoritetaan taustalla aiheuttamatta sovelluksen käytön keskeytymistä. Jos parametrin arvo taas on FALSE, palataan metodista vasta, kun pyyntö on suoritettu loppuun. Send-funktion parametrina on POST-tyyppisen tiedonsiirron tapauksessa merkkijono, jossa siirrettävä tieto kuvataan. GET-tyyppisessä pyynnössä parametria ei määritellä. [6, s. 62–63.]

Ominaisuudet

XHR-objektilla on *onreadystatechange*-tapahtumankäsittelijä, joka määrittellään kutsu-
maan tiedonkäsittelystä vastaavaa funktiota pyynnön suorituksen edetessä. Tapahtu-
mankäsittelijässä tarkkaillaan pyynnön etenemistä *readyState*- ja *status*-ominaisuuksien ilmoittamien tietojen perusteella. *readyState*-muuttuja ilmoittaa pyynnön tilan kuvaten sen edistymistä lukuina nollasta neljään taulukon 2 mukaisesti. *Status*-muuttuja kertoo HTTP-statuskoodin, esimerkiksi luku 200 tarkoittaa, että pyyntö on suoritettu onnistuneesti. Mahdollinen HTTP-statuskoodin mukana tullut teksti löytyy *statusText*-muuttujasta.

Taulukko 2. *readyState*-muuttujan ilmoittamat pyynnön eri tilat.

Tila	Kuvaus	Suoritusvaihe koodissa
0	palvelinyhteyttä ei ole aloitettu	pyyntöobjekti on luotu, mutta <i>open()</i> -metodia ei ole vielä kutsuttu
1	yhteys palvelimeen on muodostettu	<i>open()</i> -metodi ollaan käsitelty, mutta <i>send()</i> -metodia ei olla vielä kutsuttu
2	pyyntö vastaanotettu	<i>send()</i> -metodi käsitelty, mutta tietoa ei olla vielä vastaanotettu
3	pyynnön suoritus kesken	tietoa vastaanotetaan
4	pyyntö suoritettu	kaikki tieto on vastaanotettu ja valmiina esitettäväksi

Kun pyyntö on suoritettu, on palvelimelta saapunut tieto saatavissa tietotyypistä riippuen joko merkkijonona *responseText*-ominaisuudesta tai DOM-yhteensopivana XML-dokumenttina *responseXML*-attribuutista. [6, s. 62–63, 88.]

Esimerkkikoodissa 2 on esitettynä tiedonhaun suorittava funktio, jolle annetaan parametreina tiedon sijainti palvelimella sekä palvelimelta saadun tiedon käsittelystä vastaava funktio. Siinä käytetään hyväksi aiemman esimerkin XHR-objektin luomisfunktiota, ja esitetään esimerkki tässä osiossa kuvattujen metodien ja ominaisuuksien käytössä.


```

1 function getData(url, handler) {
2     // Sijoitetaan uusi XMLHttpRequest-objekti muuttujaan.
3     var request = createRequestObject();
4     // Määritetään tapahtumankäsittelijä kutsumaan tiedonkäsittelystä
5     // vastaavaa funktiota, joka saadaan parametrina.
6     request.onreadystatechange = handler;
7     // Alustetaan GET-tyyppinen pyyntö, URL saadaan parametrina.
8     request.open("GET", url);
9     // Lähetetään pyyntö palvelimelle.
10    request.send();
11 }

```

Koodiesimerkki 2. Tiedonhaku palvelimelta.

5.1.2 Tiedon käsittely

Document Object Model

Document Object Model on ohjelmointirajapinta, jota käyttäen puurakenteena kuvattavan dokumentin, kuten HTML- tai XML-tiedoston, sisältöä voidaan muokata useilla ohjelmointikielillä. Ajax-sovelluksessa sitä käytetään selaimen puolella yhdessä JavaScriptin kanssa muuttamaan websivun rakennetta dynaamisesti, lukemaan ja jäsentelemään palvelimelta saapuvaa tietoa sekä uusien XML- ja HTML-dokumenttien luomiseen. Palvelimen puolella sen avulla muotoillaan yhdessä PHP:n kanssa selaimen lähetettävää tietoa sekä luetaan puurakenteisia dokumentteja. [16, s. 36.]

Palvelimelta saapuvan vastauksen tarkastelusta ja käsittelystä vastaava funktio, jonka koodiesimerkissä 2 esitetty getData-funktio saa toisena parametrina, voisi olla koodiesimerkin 3 mukainen.

```

1 function replace() {
2     // Haku suoritettu.
3     if (request.readyState == 4) {
4         // Haku onnistui.
5         if (request.status == 200) {
6             // Muuttuja palvelimelta saadulle uudelle XML-muotoiselle
7             // sisällölle.
8             var xmlData = request.responseXML;
9             // Määritetään elementti, jonka sisältö korvataan.
10            var toBeReplaced = document.getElementById("content");
11            // Korvataan elementin sisältö.
12            toBeReplaced.innerHTML = xmlData;
13        } else {
14            // Pyyntö palvelimelle epäonnistui.
15            alert("Information request failed: " + request.statusText);
16        }
17    }
18 }

```

Koodiesimerkki 3. Funktio tarkastelee pyynnön etenemistä ja määrittelee, mitä palvelimelta saatavalle tiedolle tehdään.

Esimerkki havainnollistaa XHR-objektin eri ominaisuuksien käytön lisäksi DOM-mallin käyttöä Ajaxin yhteydessä. Mallissa määriteltyä *getElementById()*-metodia käytetään etsimään HTML-tiedostosta yksittäinen elementti, jonka tunniste on *content*. Kyseinen elementti voi olla vaikkapa lohko, jolloin se olisi HTML-tiedostossa esitettyä seuraavasti: `<div id="content"></div>`. Esimerkissä löydetyn elementin sisältö korvataan palvelimelta saadulla tiedolla käyttämällä *innerHTML*-ominaisuutta.

JSON

XML:n ohella selaimen ja palvelimen välisen tiedon siirtomuotona käytetään JSON:a (JavaScript Object Notation). Se on XML:ää kevyempi vaihtoehto, eli sen jäsentely ja muodostaminen onnistuu tietokoneelta lyhyemmässä suoritusajassa. Tämä on pääasiassa sen ansiota, ettei XML:stä tuttuja tiedon ympäröiviä aloitus- ja sulkemistageja tarvita. JSON-muotoinen tieto on helppolukuinen, ja sen kirjoittaminen on yksinkertaista, tiedon sisältämä muuttuja ja tieto erotetaan puolipisteellä. Taulukossa 3 on esitetään vertaus XML- ja JSON-merkintätavan eroista. [16, s. 107–108.]

Taulukko 3. JSON- ja XML-tietomuotojen vertailu.

JSON	XML
<pre>"message": { "id": "1", "time": "2011-10-10 14:15", "text": "Moi!" } </pre>	<pre><message> <id>1</id> <time>2011-10-10 14:15</time> <text>Moi!</text> </message> </pre>

5.1.3 JavaScript-kirjastot

JavaScriptiin on kehitetty useita koodikirjastoja, joiden tarkoituksena on tehostaa sen käyttöä weisivuilla. Niitä hyödyntämällä muun muassa tapahtumien käsittelyn ja elementtien animoinnin toteuttaminen yksinkertaistuu ja helpottuu. Suositut kirjastot myös laajenevat ja monipuolistuvat jatkuvasti ohjelmoijien kehittäessä niihin käyttökelpoisia lisäosia. Suuria ja paljon käytettyjä kirjastoja ovat esimerkiksi jQuery ja MooTools. Näistä ensimmäistä on hyödynnetty myös vedonlyöntikirjanpitosovelluksen toteuttamisessa. jQuery on helppokäyttöinen ja siihen on netissä kattava ohjeistus esimerkkeineen. jQuery-kirjasto on erillinen JavaScript-tiedosto ja liitetään HTML-dokumenttiin samalla tavalla kuin muutkin tiedostot. Erityisesti Ajax-toimintojen toteuttaminen helpottuu sen avulla huomattavasti. Lyhykäisyydessään koodiesimerkin 4 mukaisesti jQuery-koodilla saadaan haettua Ajaxin GET-pyyntöä käyttäen elementin, jonka id on *info*, sisällöksi info.php-tiedostosta saatu tieto.

```

1 $.get("info.php", function(data) {
2     $("#info").html(data);
3 });

```

Koodiesimerkki 4. Yksinkertainen GET-pyyntö jQueryllä toteutettuna.

Kirjanpitosovelluksen kehitysvaiheessa syntyi toisinaan haasteellisia tilanteita JavaScript-tapahtumien liittämisen dynaamisesti luotuihin elementteihin. jQueryn *live()*-tapahtumankäsittelijästä oli apua dokumentin lataamisen jälkeen lisättyjen elementtien toimintojen kuntoon saamisessa.

5.2 Palvelinympäristö

5.2.1 PHP

Palvelimen puolella suoritettavalla PHP-ohjelmointikielellä luodaan dynaamisia web-sivuja. PHP-koodi tulkitaan ohjelman suoritusvaiheessa, ja sitä voidaan upottaa HTML-dokumenttiin. Ajax-sovelluksessa PHP:lla rakennetaan näytettävä sisältö HTTP-pyyntön määrittelemällä tavalla. PHP:lla voidaan sisällyttää palvelimella sijaitsevia tiedostoja kätevästi toisiinsa koodiesimerkin 5 mukaisesti. Siinä haetaan Ajaxin GET-pyyntöä vastaava sivu näytettäväksi.

```
//Sijoitetaan haettu sivu muuttujaan
if(isset($_GET["page"]))
    $page = $_GET["page"];

//Tarkistetaan löytyykö haettua sivua mahdolliset sivut
//määrittelevästä taulukosta
if(isset($page) && in_array($page, $pages))
{
    //Sisällytetään löydetty sivu näytettäväksi
    include($page.".php");
}
else
{
    //Haettua sivua ei löytynyt, näytetään tästä ilmoittava sivu
    include("not_found.php");
}
```

Koodiesimerkki 5. GET-pyyntöllä haetun sivun sisällyttäminen palvelimelta palautettavaan vastaukseen.

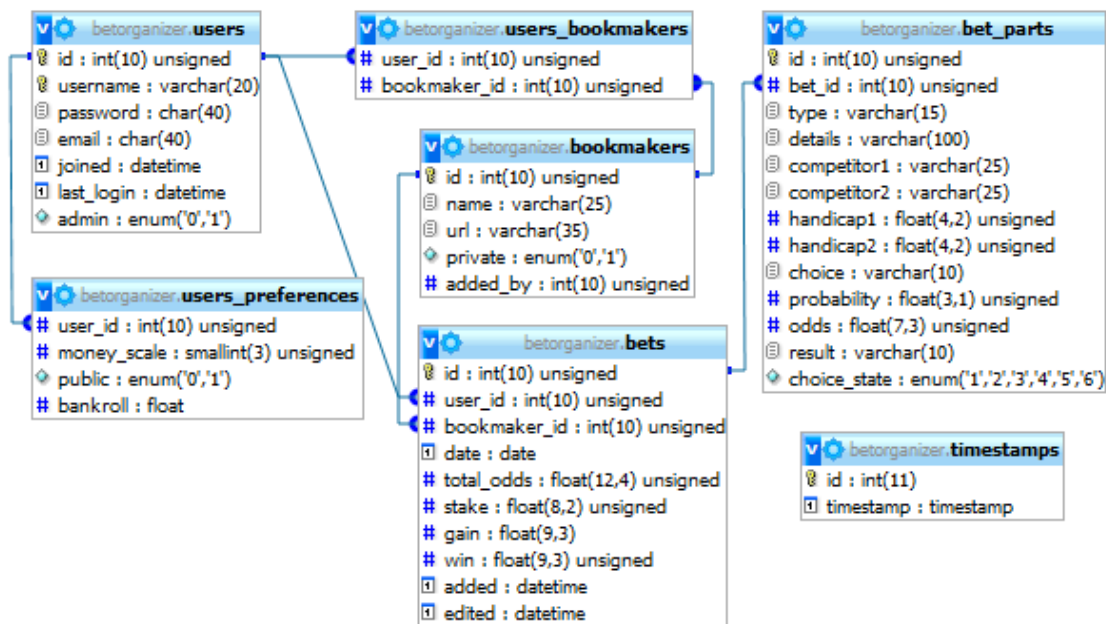
PHP:lla voidaan muodostaa tietokantayhteyksiä ja tehdä tietokantakyselyjä. PDO-luokan (PHP Data Objects) kautta onnistuu kaikkien PHP:n tukemien tietokantojen, kuten MySQL- ja Oracle-tietokantojen, käyttäminen. Koodiesimerkissä 6 luodaan PDO-olio MySQL-tietokantayhteydelle. Parametreiksi annetaan tietokannan asetukset [17].

```
$db = new PDO("mysql:host=".$host.";dbname=".$db_name,$user,$pw);
```

Koodiesimerkki 6. Uuden PDO-olion luominen.

5.2.2 MySQL-tietokanta

Sovelluksen käyttämän tiedon varastointipaikkana toimii MySQL-tietokanta. Tietokanta koostuu tauluista, joiden sisältämät tiedot on ryhmiteltyinä sarakkeisiin. Eri taulujen tietojen välille voidaan luoda yhteyksiä. Vedonlyöntikirjanpitosovelluksen tietokantarakenne on kuvan 5 mukainen.



Kuva 5. Bet Organizerin tietokantarakenne.

Esimerkkikoodissa 7 tehdään kysely vedonlyöntikirjanpitosovelluksen tietokannasta. Tietokantaluokassa määritelty tietokantaolio on sijoitettuna *\$db*-muuttujaan. PDO-oliolla toteutettuun kyselyyn ei tietoturvan parantamiseksi, eli SQL-injektoiden estämiseksi, tule laittaa määriteltyjä hakuparametreja suoraan, vaan ne lisätään esimerkin mukaisesti viitteinä. Parametrit sijoitetaan paikalleen myöhemmin ennen kyselyn suorittamista.

```

//Luodaan kyselylause, joka hakee niiden vetojen lukumäärän, joiden
//yhdessä tai useammassa osassa esiintyy määritelty vedon tila
//(viite :state) ja joilla on tietty käyttäjä (viite :user_id).
$sql = "SELECT count(DISTINCT b.id) FROM bets AS b
        INNER JOIN bet_parts AS bp ON b.id = bp.bet_id
        WHERE bp.choice_state = :state AND user_id = :user_id";

//Muodostetaan yhteys tietokantaluokan kautta.
global $db;
$connection = $db->connect();
//Luodaan valmisteltu lause.
$query = $connection->prepare($sql);
if($query)
{
    //Sijoitetaan viitteisiin parametrit.
    $query->bindParam(":user_id", $_SESSION["user_id"]);
    $query->bindParam(":state", $state);

    //Suoritetaan kysely.
    if($query->execute())
        $results = $query->fetch();
}

```

Koodiesimerkki 7. Tietokantakysely PDO-rajapintaa käyttäen.

5.3 Kehitystyökalut

Ohjelmointityökaluna käytettiin avoimeen lähdekoodiin perustuvaa Eclipse-ohjelmaa, jonka useista eri jakelupaketeista löytyvät kehitysympäristöt lukuisille eri ohjelmointikielille. Sovelluksen paikallinen testaus tapahtui avoimeen lähdekoodiin perustuvan XAMPP-kehitystyökalupaketin tarjoamalla Apachella, PHP:lla sekä MySQL-tietokannalla.

Firefox-selaimen Firebug-lisäosaa käytettiin apuna HTML- ja JavaScript-koodissa esiintyvien virheiden löytämiseksi. Sillä saa myös helposti tarkastettua, mitä tietoa palvelimelle käytännössä lähetetään ja minkälaisen vastauksen palvelin antaa. Sovelluksen vähäiset graafiset elementit luotiin kuvankäsittelyohjelma Adobe Photoshopilla.

6 Vedonlyöntikirjanpitosovellus

Tässä luvussa esitellään toteutetun vedonlyöntikirjanpitosovelluksen käyttöliittymä toiminnallisuuksineen. Sovelluksen sisältämien eri sivujen ja osien päivitykset tapahtuvat Ajaxilla GET-pyyntöä käyttäen, ja yleisesti tieto ladataan XML-muotoisena. Käyttäjän tekemät tietokantaan vaikuttavat toiminnot puolestaan tapahtuvat POST-tyyppisillä pyynnöillä, joiden tiedonsiirtomuotona käytetään JSON:a. Käyttäjän syötteet tarkastetaan ensin selaimen puolella ja lopullinen validointi tehdään aina palvelimella.

Sisäänkirjautumaton käyttäjä voi siirtyä rekisteröitymissivulle, kirjautua sisään järjestelmään tai vaatia salasanan unohtumistapauksessa salasanan palautustoimenpideohjeita lähetettäväksi sähköpostiosoitteeseensa. Vierailijan käytettävissä ovat kohdissa 6.1–6.4 kuvatut toiminnot. Vain sisäänkirjautuneen käyttäjän käytettävissä olevat toiminnot ovat vetotietojen tarkastelu, hallinta ja asetussivun toiminnot, ja ne on kuvattu kohdissa 6.5 ja 6.6.

6.1 Rekisteröityminen

Rekisteröitymissivun kautta käyttäjä voi luoda uuden käyttäjätilin täyttämällä rekisteröitymislomakkeen. Käyttäjälle näytetään mahdollinen virheilmoitus kunkin virheellistä tai puutteellista tietoa sisältävän kentän perässä, jolloin myös rekisteröintipainike muuttuu harmaaksi eikä ole käytettävissä (kuva 6). Onnistuneen rekisteröitymistapahtuman jälkeen uusi käyttäjä kirjataan automaattisesti sisään sovellukseen ja ohjataan asetussivulle, missä hän voi ensitöikseen määrittää asetukset tarpeidensa mukaisiksi.

Register a new Bet Organizer account

Username:

Password:

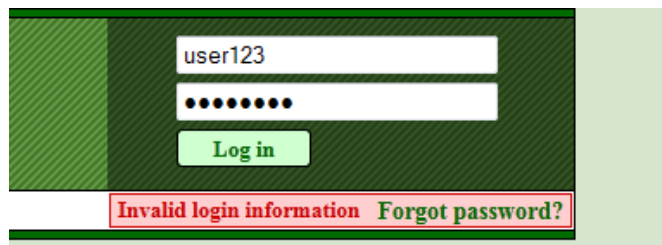
Password again: * Passwords do not match!

E-mail: * E-mail address is not valid.

Kuva 6: Rekisteröitymislomake.

6.2 Sisäänkirjautuminen

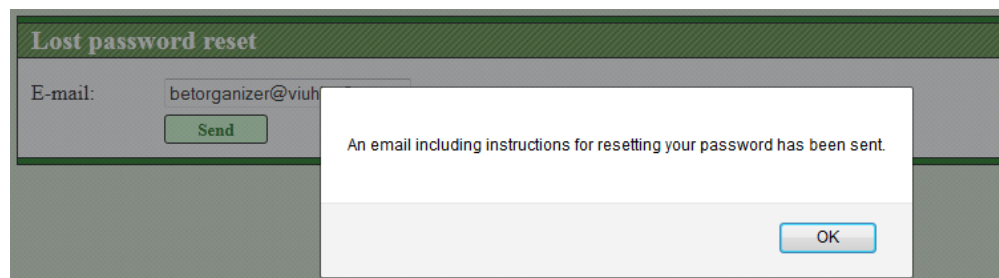
Käyttäjä pääsee kirjautumaan sovellukseen syöttämällä käyttäjätunnuksensa ja salasansa kuvassa 7 näkyviin sivun yläosassa sijaitseviin kenttiin ja painamalla sisäänkirjautumispainiketta. Mikäli käyttäjän syöttämät tunnukset ovat virheelliset, kirjautumislomakkeen läheisyydessä näytetään virheilmoitus sekä linkki salasanan unohtuslomakkeeseen (kuva 8). Sisäänkirjautunut käyttäjä ohjataan automaattisesti vetosivulle, ja kirjautuneelle käyttäjälle näytetään sisäänkirjautumislomakkeen tilalla yhteenveto veto-tilastoista ja uloskirjautumispainike.



Kuva 7: Sisäänkirjautumispaneeli.

6.3 Salasanan vaihtaminen

Mikäli käyttäjän salasana on päässyt unohtumaan, saa hän kuvan 8 mukaista salasananpalautustoimintoa käyttämällä rekisteröitymisen yhteydessä antamaansa sähköpostiosoitteeseen linkin sen uudelleenasettamiseksi. Linkin voimassaoloaika on rajoitettu, ja sen takana oleva salasananvaihtotoiminto toimii vain kerran. Salasananvaihtolomake koostuu yksinkertaisuudessaan salasana- sekä salasana uudelleen -syöttökentistä ja tallennuspainikkeesta.



Kuva 8: Sähköpostiosoitetta käytetään unohtuneen salasanan tapauksessa sen uudelleenasettamiseksi.

6.5 Vetosivu

Vetosivulla (kuva 10) on varsinainen vedonlyöntikirjanpitosovellus, jolla sisäänkirjautunut käyttäjä voi hallita ja tarkastella vetojaan. Sivun yleisnäkymän olennaisimmat osat ovat vasemmassa laidassa sijaitseva kalenteri, sivun alaosan täyttävä vetojen listausalue sekä oikean ylälaidan hakupalkki.

betmaster [Log out](#)

Bets: 249 | **Money** | **Percentages**
 Wins: 79 | Stakes: **1210.6 €** | Win: 32.1 %
 Void: 3 | Open: **0.0 €** | Return: 111.2 %
 Losses: 167 | Wins: 1346.6 €

[About](#) | [Calculators](#) | [Bets](#) | [Preferences](#)

Bets | Limit by period: 7th - 27th | jokerit | [Search](#)

Summary for bets between 7.3.2011-27.3.2011 matching 'jokerit'

Bet count		Money	
Wins: 5	Stakes: 48.45 €	Void: 0	Open: 0.00 €
Losses: 3	Wins: 88.94 €	Total: 3	Profit: 40.49 €
Percentages		Greatest	
Win: 62.5 %	Profit: 23.00 €	Odds: 3.45	EV: 1.06
Return: 183.6 %			

Bets between 7.3.2011-27.3.2011 matching 'jokerit'

Date	Bookmaker	Parts	Events	Total odds	Stake	Win	Profit	
26.3.2011	Unibet	1	Jokerit - HIFK	2.30	6.75	0.00	-6.75	Edit Copy
25.3.2011	Unibet	1	HIFK - Jokerit	3.35	5.00	16.75	11.75	Edit Copy
23.3.2011	Unibet	1	Jokerit - HIFK	2.40	7.00	0.00	-7.00	Edit Copy
Type	Details	Game	Bet	Odds	P	EV	Result	
1X2	SM-liiga QF 4/7	Jokerit - HIFK	1	2.400	43.0 %	1.03	2 - 3 OT	
21.3.2011	Unibet	1	HIFK - Jokerit	3.20	4.50	0.00	-4.50	Edit Copy
19.3.2011	Unibet	1	Jokerit - HIFK	2.30	5.00	11.50	6.50	Edit Copy
18.3.2011	Veikkaus	1	HIFK - Jokerit	3.30	10.00	33.00	23.00	Edit Copy
Type	Details	Game	Bet	Odds	P	EV	Result	
1X2	SM-liiga QF 1/7	HIFK - Jokerit	2	3.300	32.0 %	1.06	0 - 1	
10.3.2011	Paf	1	Jokerit - Jyp	2.20	6.00	13.20	7.20	Edit Copy
8.3.2011	Betsson	1	Ässät - Jokerit	3.45	4.20	14.49	10.29	Edit Copy

Kuva 10. Vetosivun yleisnäkymä.

6.5.1 Kalenteri

Kuukausikalenteri toimii sivun päänavigointivälineenä. Kalenterista käyttäjä voi valita päivän, jolle haluaa lisätä vetoja tai jonka vetoja haluaa tarkastella tai muokata. Kalenterin ylälaidassa on painikkeet eri kuukausien välillä siirtymiselle sekä vuoden vaihtamiselle. Kuukauden päivän värikoodaus havainnollistaa, kuinka voitollinen tai tappiollinen

kukin vetopäivä on suhteutettuna asetuksista valitun rahakertoimen suuruuteen. Valikoisella taustavärillä näkyvät päivät eivät sisällä vetoja. Päiväkohtaisen vetomäärän ja rahatilanteen voi tarkastaa pikaisesti viemällä kohdistimen informaatiokuvion päälle, kuten kuvan 8 tapauksessa on tehty.

Päivän valinnan tapahtuessa päivitetään vetolistausosio sekä kalenterin oikealla puolella sijaitseva koostetieto-osio. Kuukauden tai vuoden vaihtamisen yhteydessä asetetaan valituksi päiväksi kuukauden ensimmäinen päivä, minkä lisäksi hakupalkin päivien valintalistat päivittyvät vastaamaan nykyistä kuukautta.

6.5.2 Vetotietojen hakeminen

Kalenterin tarjoaman päiväkohtaisen vetotiedon näyttämisen lisäksi tietoa voi hakea käyttämällä hakupalkkia. Vetoja voi hakea muun muassa vedonvälittäjän tai joukkueen nimen perusteella, ja haun voi rajoittaa valitulle ajanjaksolle. Uusi haku suoritetaan Ajaxia käyttäen aina, kun hakukriteereissä tapahtuu muutos, mutta hakupainike on silti valinnaisena olemassa.

Haun toteutuksessa kokeiltiin MySQL:n tarjoamaa Full-text -hakumuotoa. Siinä hakutuloksien rajaus määritetään tietokantakyselyssä MATCH...AGAINST-rajauksella normaalin WHERE-lauseen sijaan. Hakuja suoritetaan siten, että erilliset sanat poissulkevat sellaiset hakutulokset, joissa ei esiinny kaikkia näistä sanoista. Lisäksi hakutuloksesta voi poistaa tiettyjä sanoja sisältäviä tuloksia lisäämällä "-"-merkin näiden sanojen eteen.

Koostetut tiedot haetuista vedoista näkyvät kalenterin oikeanpuoleisessa osiossa. Yhteenveto käyttäjän kaikista vedoista on nähtävissä sivun oikean ylälaidan profiilipaneelistä (kuva 11). Keskeneräiset vedot saa listattua profiiliosioon ilmestyvää niistä ilmoitettavaa punertavaa laatikkoa napsauttamalla.

betmaster			Log out
Bets: 258	Money	Percentages	You have 2 incomplete bets.
Wins: 81	Stakes: 1261.0 €	Win: 32.0 %	
Void: 5	Open: 15.0 €	Return: 111.3 %	
Losses: 172	Wins: 1403.7 €		

Kuva 11: Profilipaneeli.

Hakujen tuloksena löydetyt vedot näytetään sivun alalaidan vetolistausalueella. Haettujen vetojen listausjärjestystä voi muuttaa halutun tietokentän arvon mukaan otsikkokenttää napsauttamalla. Kaikkien listattujen vetojen tarkempien tietojen näyttäminen tai piilottaminen kerralla tapahtuu napsauttamalla otsikkorivin tasalla oikealla puolella olevaa pientä "+"- tai "-"-painiketta (kuva 12). Vastaava vetokohtainen painike sijaitsee jokaisen vedon kohdalla samassa kohdassa vaakatasossa. Aivan oikeassa reunassa sijaitsevat vetojen hallintapainikkeet, joiden toiminnoista lisää seuraavassa kohdassa.

6.5.3 Vetojen hallinta

Win	Profit			
0.00	-5.12	+	?	
0.00	-10.00	+	Edit	Copy
32.00	12.00	+	Edit	Copy

1) Click to copy bet details

2) Select a date using the calendar

3) Modify details and save new bet

Kuva 12: Vetojen hallintapainikkeet sijaitsevat vetolistausalueen oikeassa laidassa.

Uuden vedon lisääminen tapahtuu painamalla vedonlisäyspainiketta (kuva 12), jolloin vetosivun päälle avautuu kuvan 13 mukainen ikkuna. Vedonlisäyspainike on näkyvä vetojenlistausosion yläosassa vain, kun kalenterista on valittuna jokin päivä.

Bet Organizer

betmaster Log out

Bets: 249
Wins: 79
Void: 3
Losses: 167

Money
Stakes: 1210.6 €
Open: 0.0 €
Wins: 1346.6 €

Percentages
Win: 32.1 %
Return: 111.2 %

[About](#) | [Calculators](#) | [Bets](#) | [Preferences](#)

New Bet - Thursday 25.8.2011
X

Bookmaker
NordicBet

Stake
5 €

Add part
Remove

Type	Details	Competitor 1	Competitor 2	Bet	Add result
1X2	European Trophy	Slavia Prague	Jokerit	Choice: 2 Odds: 3.2 33 %	Score: 0 - 0 Decision: Regular time
		<input checked="" type="checkbox"/> Handicap + 1	<input type="checkbox"/> Handicap + 1		
O/U	European Trophy	Luleå	Kärpät	Choice: Over 5.5 Odds: 2.3 46 %	Score: 0 - 0 Decision: Regular time
		<input type="checkbox"/> Over/Under	<input type="checkbox"/> Over/Under		

Save

Kuva 13. Vedonlisäysikkuna.

Lisättävissä vedossa voi olla useita vedon osia, jotka voivat olla erityyppisiä. Vedon osien määrää muutetaan kuvassa 13 näkyvillä ikkunan oikeassa yläkulmassa sijaitsevilla painikkeilla. Tallennusvaiheessa ohjelma määrittää syötettyjen tietojen perusteella vedon tilan automaattisesti. Kaikilla vedon osilla on syötetyn tuloksen mukaan määräytyvä tila, jotka yhdessä muodostavat koko vedon tilan. Kukin vedon osa saa jonkin taulukossa 4 kuvatun tilan.

Taulukko 4. Vedon eri tilat.

Tilanumero	Kuvaus
1	Voittava vedon osa, jolloin se lisää vedon kokonaiskerrointa kertoimensa verran, eli palauttaa rahaa panoksen ja kertoimen määrittelemän tulon verran.
2	Häviävä vedon osa, nolaa vedon kertoimen, jolloin veto ei palauta yhtään rahaa riippumatta muista vedon osista.
3	Panoksen suuruisen rahamäärän palauttava vedon osa, ei vaikuta vedon kokonaiskerroimeen mitenkään.
4	Keskeneräinen vedon osa, sen lopputulos on antamatta. Vetoon panostetun rahamäärän tila merkitään avoimeksi olettaen että yhtäkään osaa ei ole merkitty häviäväksi.
5	Vedon osa, joka on puoliksi voittava. Puolet panoksesta palautetaan suoraan ja puolet kerrotaan kertoimen kanssa. Vaikuttaa vedon kokonaiskerroimeen siis $\frac{1}{2} \times \text{kerroin} + \frac{1}{2}$ verran .
6	Puolittain häviävä vedon osa, palauttaa puolet panoksesta, kokonaiskerroin kerrotaan puolikkaalla.

Vedon muokkaaminen tapahtuu vastaavanlaisessa vetosivun päälle avautuvassa ikkunassa kuin uuden vedon kirjaustapahtumakin. Myös vedon poistaminen tapahtuu muokkaussivun kautta tallennuspainikkeen viereen ilmestyvällä poistopainikkeella. Vedon kopiointitapahtumassa kopioidun vedon tiedot sisältämä vetoikkuna avautuu, kun napsautetaan ensin kopiointipainiketta ja valitaan tämän jälkeen kalenterista päivä, jolle veto asetetaan.

Vetotietojen syöttämisen nopeuttamiseksi vetoikkunaan haetaan valittaviksi käyttäjän aiemmin kirjaamia tietokannasta löytyviä tekstimuotoisia tietoja. Kuvassa 14 näkyvän *Details*-kohdan yläreunan valintalistasta käyttäjä saa valittua yleisimpiä lähiaikoina kirjaamiaan vetokuvauksia kopioitavaksi tekstikenttään ja kilpailijakenttien (*competitor*) alapuolelle haetaan sopivia ehdotuksia kentän sisällön perusteella. Kaikkia kenttiä täytettäessä niiden läheisyydessä näytetään myös käytettävissä olevien merkkien jäljellä oleva määrä.

Details	Recent	Competitor 1	Competitor 2	23
NHL		Nashville	- An	
		<input type="checkbox"/> Handicap	Anaheim	
		+ 1	Andy Murray	

Kuva 14. Vetotietojen lisääminen yksityiskohtaisemmin esitettynä.

6.6 Asetussivu

Kuvan 15 mukaisella asetussivulla käyttäjä voi asettaa käyttämänsä vedonvälittäjät, pelikassan, rahakertoimen ja vaihtaa salasanansa. Vedonvälittäjien asettamisessa ideana on lisätä kaikki ne vedonvälittäjät, joilla käyttäjällä on tili ja joita ei oletuksena ole listattuna. Valittuna taas on tarkoituksenmukaista olla näistä sellainen joukko, joka tiettyinä ajanjaksona on aktiivisessa käytössä, sillä vain valittuna olevat vedonvälittäjät listataan vetoikkunan valintalistassa.

Käyttäjä voi määritellä käyttämänsä pelikassan koon alkutilanteen ja valuutan. Pelikassan senhetkinen koko lisätään automaattisesti Kelly-laskuriin, kun käyttäjä on kirjautuneena sisään.

Preferences

Bookmaker selection

<input type="checkbox"/> 10bet	<input checked="" type="checkbox"/> Betfair	<input type="checkbox"/> Dafabet	<input type="checkbox"/> Legends	<input type="checkbox"/> Stan James
<input type="checkbox"/> 188BET	<input type="checkbox"/> Betsafe	<input type="checkbox"/> Eurobet	<input checked="" type="checkbox"/> NordicBet	<input type="checkbox"/> The Greek
<input type="checkbox"/> 5Dimes	<input checked="" type="checkbox"/> Betsson	<input checked="" type="checkbox"/> Expekt	<input type="checkbox"/> Paddy Power	<input type="checkbox"/> Titanbet
<input checked="" type="checkbox"/> Bet-at-home	<input type="checkbox"/> Betway	<input type="checkbox"/> Gamebookers	<input checked="" type="checkbox"/> Paf	<input checked="" type="checkbox"/> Unibet
<input checked="" type="checkbox"/> Bet24	<input type="checkbox"/> Bodog	<input type="checkbox"/> Globet	<input checked="" type="checkbox"/> Pinnacle	<input checked="" type="checkbox"/> Veikkaus
<input checked="" type="checkbox"/> Bet365	<input type="checkbox"/> Bwin	<input type="checkbox"/> Interwetten	<input type="checkbox"/> SBOBET	<input type="checkbox"/> Victor Chandler
<input type="checkbox"/> BetCRIS	<input type="checkbox"/> Canbet	<input type="checkbox"/> Island Casino	<input type="checkbox"/> Sportbet	<input type="checkbox"/> William Hill
<input type="checkbox"/> Betdaq	<input type="checkbox"/> Centrebet	<input checked="" type="checkbox"/> Ladbrokes	<input type="checkbox"/> Sportingbet	<input type="checkbox"/> Youwin.com

Add bookmaker

Name:


URL (optional):

Bankroll

Starting bankroll: Current bankroll: Currency:

Money scale selection

Determines the sensitivity of color variation by profit in the calendar view.



- Loss is equal or greater than 10 * money scale
- Profit/loss is less than money scale
- Profit is equal or greater than 10 * money scale

Change password

Current password:

New password:

New password again:

Kuva 15. Asetussivu.

Asetuksista valittava rahakerroin määrittelee kuukausikalenterin päivän värin vaihtelun herkkyys suhteessa vetopäivän voitollisuuteen tai tappiollisuuteen. Rahakerroin kannattaa asettaa omaa rahamääräistä panostustaan parhaiten vastaavaksi eli suureksi, mikäli pelaa isoilla panoksilla ja päinvastoin. Myös salasanan vaihto onnistuu asetussivun kautta. Myöhemmissä versioissa asetukseen on tarkoitus lisätä ominaisuuksia, joilla käyttäjä saa muokattua käyttöliittymästä paremmin tarpeitansa tukevan. Tällainen ominaisuus olisi esimerkiksi käyttäjän mahdollisuus vaikuttaa tiedon esitystapaan: mitä tietoa näytetään missäkin ja kuinka se esitetään.

7 Yhteenveto

Insinööriyössä toteutettiin Ajaxia hyödyntäen urheiluedonlyöntiin erikoistunut kirjanpitosovellus, joka nimettiin Bet Organizeriksi. Suunnittelu- ja toteutusvaiheissa pohdittiin, kuinka tehdä vedonlyöntiä harrastavan tarpeita vastaava sovellus, jonka avulla kirjanpito olisi mahdollisimman helppoa ja jonka käyttöliittymä olisi miellyttävä. Kalenterinavigointi oli ensimmäinen mieleen juolahtanut ominaisuus, jolla saataisiin parannettua käytettävyyttä perinteiseen taulukkokirjanpitoon verrattuna. Vetojen hallintaosuudesta oli tavoitteena tehdä varsinaista vedonlyöntitapahtumaa mukaileva, jolloin olennainen tieto tulee varmemmin kirjattua ja virheellisesti syötettyjen tietojen mahdollisuus pieneä.

Keskeisimmät suunnitellut toiminnot onnistuttiin toteuttamaan määritysten ja tavoitteiden mukaisesti ja sovelluksen käyttöliittymän ulkoasusta saatiin suurin piirtein halutunlainen. Ajaxin oletusarvoisesti mukanaan tuomat käytettävyysongelmat ratkaistiin ja käytettävyyssasioita pohdittiin toteutuksen eri vaiheissa. Käyttäjälle pyritään antamaan riittävästi palautetta eri toimintojen suoritusten yhteydessä sekä virhetapahtumissa.

Prioriteetiltaan alempia määrittelyvaiheessa suunniteltuja toimintoja, graafisia tilastoja sekä vetotietojen jakamismahdollisuutta muiden käyttäjien kesken, ei ehditty insinööriyön puitteissa toteuttamaan, mikä oli oikeastaan tiedossa jo varhaisessa vaiheessa. Jatkokehittävää jäi muiltakin osin runsaasti; vetotietojen kirjaaminen on vielä rajoittunut ja kaipaa runsaasti lisää erilaisia vetomuotoja ja esimerkiksi järjestelmävetojen mahdollisuutta. Jatkossa tarkoituksena on myös mahdollistaa käyttöliittymän monipuolinen muokattavuus vedonlyöjän tarpeita parhaiten vastaavaksi. Päättäneiden vetojen lopputulosten hakeminen automaattisesti tuloksia tarjoavalta sivustolta olisi myös kätevä ominaisuus.

Lähteet

- 1 Busch, Marianne & Koch, Nora. 2009. Rich Internet Applications – State-of-the-Art. PDF-dokumentti. <http://uwe.pst.ifi.lmu.de/publications/maewa_rias_report.pdf>. Luettu 14.10.2011.
- 2 Ward, James. 17.10.2007. What is a Rich Internet Application? Verkkodokumentti. <<http://www.jamesward.com/2007/10/17/what-is-a-rich-internet-application>>. Luettu 14.10.2011.
- 3 Noda, Tom & Helwig, Shawn. 16.12.2005. Rich Internet Applications. PDF-dokumentti. <citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.4482&rep=rep1&type=pdf>. Luettu 1.10.2011.
- 4 Nielsen, Jakob. 12.2005. Why Ajax Sucks (Most of the Time). Verkkodokumentti. <<http://www.usabilityviews.com/ajaxsucks.html>>. Luettu 9.10.2011.
- 5 Hickson, Ian. 7.10.2011. HTML5. A vocabulary and associated APIs for HTML and XHTML. Verkkodokumentti. <<http://dev.w3.org/html5/spec/Overview.html#history>>. Luettu 9.10.2011.
- 6 Powell, Thomas. 2008. Ajax: The Complete Reference. McGraw-Hill Professional Publishing.
- 7 Nielsen, Jakob. 17.12.2007. Web 2.0 Can Be Dangerous... . Verkkodokumentti. <<http://www.useit.com/alertbox/web-2.html>>. Luettu 9.10.2011.
- 8 Weisstein, Eric W. "Weak Law of Large Numbers." From MathWorld--A Wolfram Web Resource. <<http://mathworld.wolfram.com/WeakLawofLargeNumbers.html>>. Luettu 1.11.2011.
- 9 Bankroll Management - What's the best staking plan for your betting?. Verkkodokumentti. <<http://www.racetips.co.uk/articles/bankroll-management-3>>. Luettu 1.11.2011.
- 10 Kelly, John L. Jr. 21.3.1956. A new interpretation of information rate. PDF-dokumentti. <<http://www.racing.saratoga.ny.us/kelly.pdf>>. Luettu 19.10.2011.
- 11 Rubin, Jeffrey, Chisnell, Dana & Spool, Jared. 2008. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. John Wiley & Sons.
- 12 Eichorn, Joshua. 2007. Understanding AJAX: Using JavaScript to Create Rich Internet Applications. Prentice Hall.

- 13 Nielsen, Jakob. 2005. Ten Usability Heuristics. Verkkodokumentti.
<http://www.useit.com/papers/heuristic/heuristic_list.html>. Luettu 1.10.2011.
- 14 Krug, Steve. Don't Make Me Think!: A Common Sense Approach to Web Usability, Second Edition. New Riders, 18.8.2005.
- 15 PHP-header. PHP-manuaali. Verkkodokumentti.
<<http://php.net/manual/en/function.header.php>>. Luettu 15.10.2011.
- 16 Brinzarea-lamandi, Bogdan, Darie, Cristian & Hendrix, Audra. 2009. AJAX and PHP: Building Modern Web Applications – Second Edition. Birmingham: Packt Publishing Ltd.
- 17 The PDO class. PHP-manuaali. Verkkodokumentti.
<<http://php.net/manual/en/class.pdo.php>>. Luettu 20.10.2011.