

Hermann Mäkitalo

URHEILUSUORITUSTEN SEURANTA ÄLYPUHELIMELLA

Toimivan prototyypin suunnittelu ja toteutus

**Opinnäytetyö
CENTRIA AMMATTIKORKEAKOULU
Mediatekniikan koulutusohjelma
Maaliskuu 2013**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieska	Aika Maaliskuu 2013	Tekijä/tekijät Hermann Mäkitalo
Koulutusohjelma Mediatekniikka		
Työn nimi URHEILUSUORITUSTEN SEURANTA ÄLYPUHELIMELLA: Toimivan prototyypin suunnittelu ja toteutus		
Työn ohjaaja FM Hannu Puomio		Sivumäärä 26
Työelämäohjaaja		
<p>Opinnäytetyön tavoitteena oli toteuttaa toimiva tietokantapohjainen urheiluosuoritusten merkitsemisjärjestelmä. Sovelluksen tuli mahdollistaa käyttäjän merkitä päivän lihaskuntoharjoituksensa nopeasti ja mahdollisimman pienellä vaivalla. Näitä tuloksia oli tarkoitus sitten voida tarkistella selaimella selkeästi kuvaajasta, jotta käyttäjä pystyisi helposti havaitsemaan itsessään tapahtuneen fyysisen kehityksen. Tavoitteena oli että järjestelmä on toimiva, helppokäyttöinen ja jopa minimalistinen eikä tyrkytä tulosten julkaisua sosiaaliseen mediaan.</p> <p>Työ pyrkii luomaan toimivan pohjan matkapuhelimen ja tietokannan väliselle turvalliselle kommunikaatiolle. Työssä on pyritty huomiomaan tietoturva jokaisessa eri vaiheessa. Järjestelmän toteuttamiseen valittiin MySQL-tietokanta, sekä ohjelmointikielistä PHP, JavaScript sekä Java. Matkapuhelin ohjelmiston kehitettiin Android-alustalle sen ilmaisuuden ja laitteiston suuren kannan vuoksi.</p> <p>Järjestelmä valmistui tavoitellussa muodossa. Valmiilla järjestelmällä voi rekisteröidä uuden tunnuksen, merkitä suorituksia sekä seurata tulosten kehittymistä selaimella. Ulkoasu ei ole viimeistelty eikä kaupallisessa muodossa, mutta toimiva pohja antaa hyvän mahdollisuuden jatkokehitykseen. Kaikkia työn eri osia voi helposti muokata ja yhdistää toisiin projekteihin.</p>		

Asiasanat

Android, Eclipse, Java, MySQL, PHP, Palvelin, Tietokanta, Tietoturva

ABSTRACT

CENTRIA UNIVERSITY OF APPLIED SCIENCES	Date March 2013	Author Hermann Mäkitalo
Degree programme Mediatechnology		
Name of thesis KEEPING TRACK OF SPORTS RESULTS USING A SMARTPHONE: a functioning prototype design and implementation.		
Instructor M.Phil. Hannu Puomio		Pages 26
Supervisor		
<p>The goal of this thesis was to create a fully functional application to keep a record on daily exercises. The application was to be as simple and minimalistic as possible. The user should then be able to examine the progress of their results on a web browser. The goal was that the system would be as easy to use as possible, and would not even offer an option to share the results on social media.</p> <p>The project aimed to develop a fully functional template to communicate securely between a smartphone and the database. Information security was taken into account on each section. MySQL-database was selected as database system, and for the rest of programming languages, Java, JavaScript and PHP were chosen due to their popularity and a great selection of expansions available. Android was selected as an operating system of the smart phone due to its market share and wide selection of devices.</p> <p>The project was finished in a form it was planned to. With the finished application, the user can register a new account, mark down the workout results, and track the development of their results. The layout was not finalized nor commercial, but the template is good for further development. All parts of the project are easy to modify and combine to other projects.</p>		
Key words Android, Database, Eclipse, Information security, Java, MySQL, PHP, Server		

KÄSITTEIDEN MÄÄRITTELY

APT – Advanced Packaging Tool. Debianille kehitetty pakettien asennus- ja hallintasovellus.

GUI – Graphical user interface. Graafinen käyttöliittymä.

SDK – Software development kit. Kehitysympäristö tietyn tai useampien laitteiden tai ohjelmointikielien hyödyntämiseen.

UI – User Interface. Käyttöliittymä, eli se näkyvä osa jonka kautta käyttäjä käyttää tuotetta.

Thread – Säie. Saman prosessin sisällä ajettava rinnakkainen suorite.

Rainbow table – Valmiiksi generoitu taulukko, johon on kirjattu esimerkiksi SHA-1 muotoiset hashit kaikista alle 10 merkkiä pitkistä salasanoista. Käytetään salasanojen murtamisen nopeuttamiseksi.

ESIPUHE

Kiitokset filosofian maisteri Hannu Puomiolle opinnäytetyön ohjaamisesta, sekä todella opettavaisista, motivoivista ja kannustavista kursseista. Kiitokset myös filosofian maisteri Joni Jämsälle, jota ilman en osaisi puoliakaan siitä mitä nyt osaan. Erityiskiitokset vielä diplomi-insinööri Ritva Saviluodolle, joka kannusti vaihtamaan linjaa ensimmäisen vuoden lopussa sähkötekniikalta mediatekniikalle. Pyörisin varmaan vieläkin ammattikorkean tuulikaapissa ympyrää voihkien kurjuutta enkä todellakaan olisi lähimain siellä missä nyt olen.

Lisäksi tahdon osoittaa suuret kiitokset vanhemmilleni Raija ja Juhani Mäkitalolle ehtymättömästä kannustuksesta ja tuesta. Suuret kiitokset kuuluvat myös siskolleni, Minna Mäkitalo-Raumalle, jolta sain paljon hyviä ideoita ja motivaatiota tehdä tätä työtä.

Suunnaton kiitos kuuluu myös avopuolisolleni Tiina Elolle, joka jaksoi kannustaa ja patisti tekemään tätä työtä, kun unohduin tekemään jotain joutavaa.

TIIVISTELMÄ	
ABSTRACT	
KÄSITTEIDEN MÄÄRITTELY	
ESIPUHE	
SISÄLLYS	
1 JOHDANTO	1
2 TYÖN SUUNNITTELU	2
2.1 URHEILUSUORITUKSET	2
2.2 RAKENNE	3
3 TYÖHÖN VALITUT OHJELMOINTIKIELET JA KEHITYSYMPÄRISTÖT	6
3.1 JAVA, ANDROID SDK JA ECLIPSE	6
3.2 PHP, HTML JA JQUERY	7
3.3 MYSQL	8
4 PUHELIMEN OHJELMISTO	11
4.1 TYÖN HAASTEET	11
4.2 LÄHDEKODIN TÄRKEIMMÄT OSAT	12
5 PALVELIN	16
5.1 PALVELIMEN RAUTAPUOLI	16
5.2 KÄYTTÖJÄRJESTELMÄ	16
5.3 PHP-RAJAPINTA	17
6 TYÖN TOTEUTUS	19
7 TULOKSET	22
8 HUOMIOITA JATKOKEHITYKSEEN	24
8.1 JÄRJESTELMÄN KÄYTETTÄVYYS	24
8.2 TIETOTURVA	25
8.3 JÄRJESTELMÄN OMINAISUUDET	26
LÄHTEET	27
KUVIOT	
KUVIO 1. Selainpään ohjelmisto.	4
KUVIO 2. Sisäänkirjautuminen.	4
KUVIO 3. Rekisteröityminen.	5
KUVIO 4. MySQL-tietokanta ja taulujen rakenteet.	9
KUVIO 5. Esimerkkisyötteen tauluista.	10
KUVIO 6. Projektin työvaiheet ja niiden yhteydet toisiinsa.	19
KUVIO 7. Projektin työvaiheet suoritusjärjestyksessä ja tehtäväkuvaukset.	20
KUVIO 8. Pääikkuna sekä lähetyksikkuna.	23
TAULUKOT	
TAULUKKO 1. Palvelinten käyttöjärjestelmät vuonna 2000 (Rantala 2000.)	16

1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli ajatus kuntoilusuoritusten merkitsemisestä ylös siten, että tuloksista saa selaimessa selkeän kuvaajan. Tuloksista voisi sitten nopealla silmäyksellä katsoa miten toistojen määrät ovat ajan kuluessa muuttuneet suuntaan tai toiseen. Lähes vastaavia sovelluksia on markkinoilla kymmeniä useimmille 2010-vuoden jälkeen valmistetulle laitteille, mutta ne painottuvat juoksun tai muiden vastaavien ulkona tapahtuvien aktiviteettien mittaamiseen sekä merkitsemiseen.

Tarkoituksena oli luoda ohjelmisto, jolla voi merkitä sisätiloissa suoritettua kuntoilusuorituksia, kuten vatsalihasliikkeet, punnerrukset sekä selkälihasliikkeet tietokantaan. Suoritusten kirjaamiseen tarvittiin ohjelmisto matkapuhelimeen, palvelin, tietokanta sekä rajapinta, jolla suoritukset merkitään tietokantaan ja luetaan sieltä. Tarkoitus oli myös, että suoritusten kirjaamisessa ei menisi kuin korkeintaan 30 sekuntia, etteivät suoritukset jäisi merkitsemättä hankalan käytettävyyden takia.

Käyttöliittymän tuli olla selkeä, helppokäyttöinen ja kosketusnäyttöpuhelimelle optimoitu. Ohjelmiston tuli omata käyttäjäkohtaiset profiilit, eli sen tuli olla jonkinlaisen tunnistusjärjestelmän omaava. Tiedot pitäisi pystyä katsomaan selaimella laitteesta riippumatta.

Tarkoituksena oli myös luoda hyvä pohja älypuhelimien ja tietokannan väliseen keskusteluun, jota voi tarvittaessa muokata muihin tarkoitukseen sekä yhdistää muihin projekteihin.

2 TYÖN SUUNNITTELU

Seuraavissa kappaleissa käydään tarkemmin läpi miksi juuri kyseiset liikkeet valittiin projektiin. Kappaleissa kuvataan myös projektin arkkitehtuuria ja esitellään tarkemmin työn vaatimia eri osia niin palvelinpuolella kuin matkapuhelimessa.

2.1 Urheilusuoritukset

Projektissa tarkoituksena oli valita lihasliikkeitä, jotka ovat helppoja suorittaa ja joista on suurin hyöty istumatyötä tekeväälle ihmiselle. Istumatyö heikentää lihasten voimaa, rappeuttaa selän rakenteita sekä aiheuttaa selkävaivoja (Keventajat.fi 2012). Selkä- ja vatsalihaksia harjoittamalla voi parantaa ryhtiä, vähentää selkävaivojen riskiä sekä kiinteyttää keskivartaloa. Vatsa ja selkälihasten tekeminen ei kuitenkaan itsessään riitä laihduttamaan keskivartaloa, koska laihduminen ei tapahdu paikallisesti vaan tasaisesti koko kehosta (Kiloklubi.fi 2012.). Automaattista urheilijoilla usein nähtävää pyykkilautavatsaa nämä liikkeet eivät anna, sillä jotta vatsalihakset näkyisivät selkeästi, tulee rasvaprosentin olla miehellä alle 10 % ja naisella alle 25 % (Kuntoguru.com 2013.).

Punnerrukset ovat yksi parhaista liikkeistä toimistotyötä tekeväälle, sillä ne vaativat ja rasittavat selkälihaksia, vatsalihaksia, hartioita sekä käsilihaksia. (Hundredpushups.com 2013.) Vatsalihaksia, selkälihaksia sekä punnerruksia voi tehdä lähes missä tahansa ja melkein millä vaatetuksella tahansa. Liikkeet ovat tehokkaita ja pieniä, joten hikeäkään ei pitäisi hirveästi tulla suorittaessa. Todennäköisesti henkilö, joka tarvitsee tätä sovellusta, ei kykene tekemään niin montaa punnerrusta, että niiden suorittamisessa kestäisi minuuttia kauempaa. Tehokkaalla suorittamisella kaikkien kolmen liikkeen toistojen suorittamiseen kuluu alle seitsemän minuuttia, joten aika ei voi olla tekosyy tekemättä jättämiseen.

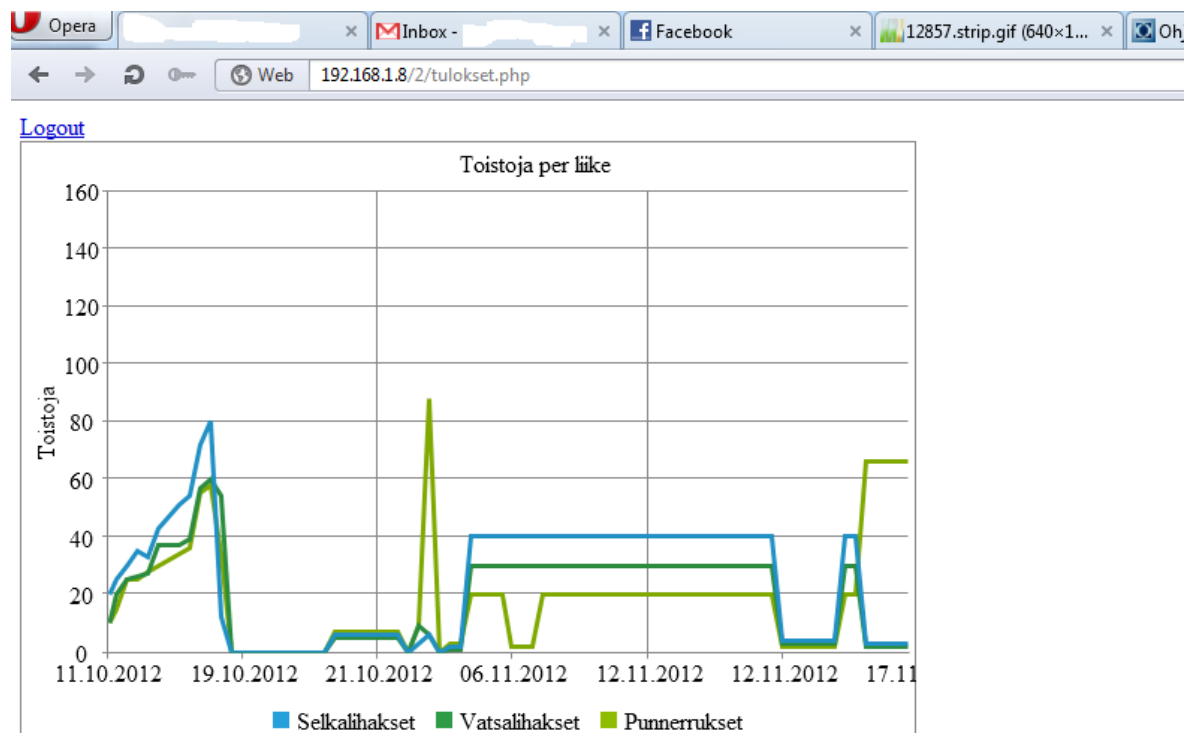
2.2 Rakenne

Järjestelmä käyttää Asiakas - Palvelin -arkkitehtuuria, jossa puhelin toimii asiakkaana niin selaimen kuin myös Android -ohjelmiston kautta ja LAMP-palvelin palvelimena PHP -rajapintoinen ja Apache -palveluinen. Järjestelmässä on kaksi selkeästi erillään olevaa osaa: puhelimen ohjelmisto ja www-sovellus. Molempien tuli toimia toistensa kanssa keskenään saumattomasti, esimerkiksi siten että käyttäjä luo selaimessa käyttäjätunnuksen ja salasanan, ja tämä käyttäjätunnus toimii sitten myös matkapuhelimen ohjelmistossa. Järjestelmän tuli toimia myös toisinpäin, eli jos käyttäjä syöttää matkapuhelimella suorituksia, tulee nämä pystyä esittämään selaimessa. Tähän yksinkertaisin ja skaalautuvin ratkaisu oli SQL-tietokanta, jota voidaan siirtää käyttäjämäärästä riippuen tehokkaampaan tai taloudellisempaan palvelimeen.

MySQL, josta myöhemmin lisää omassa kappaleessaan, oli helppo valinta koska sille on toteutettu jo valmiiksi laaja valikoima rajapintoja eri kielille, se on kevyt ja toimii todella suurillakin tauluilla ja tietokannoilla. (Stackoverflow.com 2013.) Ohjelmiston rakenteen tuli olla niin selkeä, että käyttäjän ei tarvitse kertaakaan miettiä kumpaa nappia hänen tulisi painaa tai mihin pitäisi seuraavaksi mennä. Ohjelmiston suunnittelussa tuli ottaa mallia uudemmissa konsolipeleistä, eli kehittää järjestelmä toimimaan täysin ”putkijuoksuna”.

Ei ole väliä tekeekö käyttäjä ensin liikkeet ja merkkäako hän ne sitten ohjelmaan, vai merkkäako hän jokaisen eri liikesarjan jälkeen tuloksen ylös, sillä kaikki tulokset merkitään samalle sivulle, ja kaikki suoritukset lähetetään samalla kertaa kirjautumisen jälkeen. Lähetä-nappia painaessa ohjelmisto lähettää HttpPost-requestin palvelimella sijaitsevalle PHP-rajapinnalle. Rajapinta suorittaa pyynnön vastaanotettuaan kyselyn tietokantaan, jossa tarkistetaan ovatko käyttäjätunnus ja salasana toimiva pari. Mikäli käyttäjä kirjoitti tunnukset oikein, palauttaa kysely PHP-tiedostolle tietokannasta käyttäjää vastaavan ID:n, jotta rajapinta osaa syöttää HttpPost-requestissa saapuneet suoritusten toistot oikealle käyttäjälle tietokantaan. Mikäli taas käyttäjä kirjoitti arvot väärin, palautetaan arvo nolla, jolloin rajapinta pysäyttää prosessin ja ilmoittaa tästä asiakasohjelmistolle.

Käyttäjä voi tämän jälkeen käydä sivustolla tarkistamassa kuvaajasta miten tulokset ovat kehittyneet. Suorituksista piirretään selkeä kuvaaja, kuten kuviossa 1. on nähtävissä, josta voi nopeallakin silmäyksellä huomata ovatko tulokset kehittyneet mihin suuntaan ja milloin viimeksi on merkinnyt suorituksensa ylös.



KUVIO 1. Selainpään ohjelmisto.

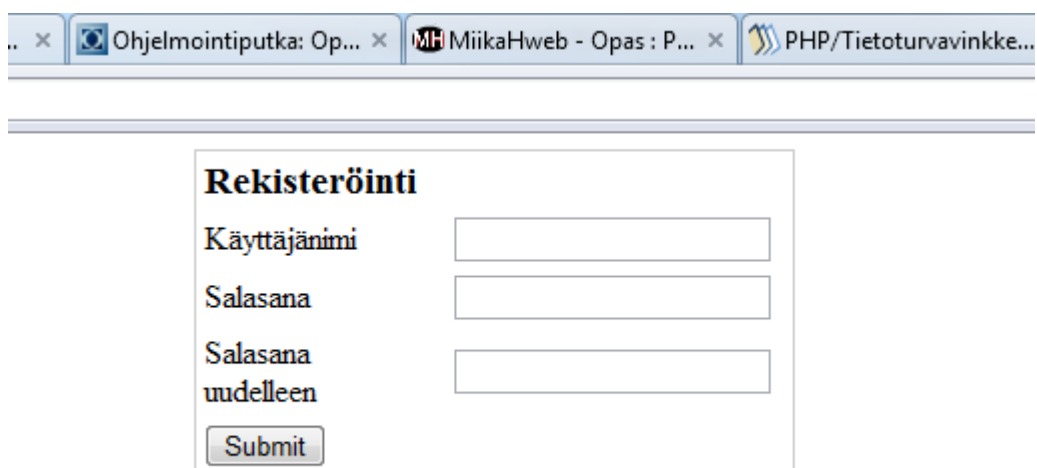
Ohjelmointiputka: Op... x MiikaHweb - Opas : P... x PHP/Tietotun

Kirjaudu

Nick :

Salasana :

KUVIO 2. Sisäänkirjautuminen.



The image shows a web browser window with three tabs: 'Ohjelmointiputka: Op...', 'MiikaHweb - Opas : P...', and 'PHP/Tietoturvakke...'. The main content area displays a registration form titled 'Rekisteröinti'. The form contains three input fields: 'Käyttäjänimi', 'Salasana', and 'Salasana uudelleen', each followed by a text input box. Below the input fields is a 'Submit' button.

Rekisteröinti

Käyttäjänimi

Salasana

Salasana uudelleen

KUVIO 3. Rekisteröityminen.

3 TYÖHÖN VALITUT OHJELMOINTIKIELET JA KEHITYSYMPÄRISTÖT

Projektiin valittiin tämän hetken yleisin matkapuhelin käyttöjärjestelmä Android. Android on 2005 Android Inc. yhtiön lanseeraama avoimen lähdekoodin käyttöjärjestelmä. Android julkistettiin kaupallisesti ensimmäistä kertaa 2007, ja jo vuonna 2010 se oli suosituin käyttöjärjestelmä. Androidille on saatavissa ilmaiseksi kehitystyökalut ja ohjelmistojen julkaisu Google Play marketissa ei maksa kehittäjälle mitään, mikä tekee Androidista erityisen hyvän kehitysalustan aloitteleville yrityksille ja yksityisille harrastajille.

Kehittäjälle kuitenkin kannattavammaksi tulisi julkaista ohjelmistot iOS-järjestelmälle, sillä tutkimukset osoittavat Applen matkapuhelinten käyttäjät ovat valmiimpia käyttämään rahaa sovelluksiinsa kuin taas Android käyttäjät (Hardy, Q. 2012), ja Androidilla moni kehittäjä saa tulonsa peleissä ja ohjelmissa näytettävistä mainoksista (Viswanathan, P. 2013). Pelikehittäjille suuri tulonlähde on myös virtuaalisen pelivaluutan ja pelihyödykkeiden myynti mikromaksuilla, eli pienillä maksuilla muutoin ilmaisessa pelissä. Mutta koska iOS kehittäjiltä vaaditaan lisenssimaksu eikä ohjelmistosta ei ollut tarkoitus hyötyä rahallisesti, päädyttiin Androidiin.

3.1 Java, Android SDK ja Eclipse

Androidille on saatavissa ilmaiseksi kääntötyökalut, joten kehittäjä voi halutessaan käyttää valitsemaansa ohjelmointiympäristö. Tässä projektissa käytetään Eclipse IDE for Java Developers kehitystyökaluja ja siihen saatavaa Android Development Tools (ADT) pakettia. Eclipse on alun perin IBM:n julkistama kehitystyökalu joka tukee C, C++, Java sekä PHP ohjelmointikieliä. Eclipse julkaistiin 2001 avoimen lähdekoodin lisenssille, ja vuodesta 2004 kehityksestä on vastannut Eclipse Foundation (Eclipse.org, 2013).

Eclipsessä on paljon samoja ohjelmointia nopeuttavia ominaisuuksia joita löytyy myös Microsoft Visual Studio kehitystyökaluista, kuten esimerkiksi automaattinen koodin muotoilu, usean rivin uloskommentointi pikanapeilla sekä automaattinen tarvittavien kirjastoiden lisääminen sekä turhien käyttämättömien kirjastoiden poistaminen.

Java on vuonna 1995 Sun Microsystemsin julkaisema laitteistosta ja käyttöjärjestelmästä riippumaton olio-ohjelmointikieli Java on saanut vaikutteita useilta kieliltä, selvimmin kuitenkin C++:sta, josta Java on pitkälti kopioinut syntaksinsa. (Niskanen, Kontio & Vierimaa 2000, 3)

3.2 PHP, HTML ja jQuery

Tietoturvasyistä Android ohjelmisto ei suoraan voi kirjoittaa tietokantaan itse mitään, vaan se lähettää PHP-rajapinnalle kyselyn, jonka osoittautuessa lailliseksi pyynnöksi rajapinta hoitaa tietokannan käsittelyn. PHP on The PHP Groupin ylläpitämä ohjelmointikieli, jota käytetään yleisesti dynaamisten web-sivustojen luonnissa. PHP:lle on olemassa tietokantatuki tässä työssä käytettyyn MySQL-tietokantaan, mikä helpotti tietokannan käsittelyä huomattavasti.

HTML on avoimesti standardoitu kuvauskieli, jolla on yleisesti luotu web-sivustojen käyttäjälle näkyvä osuus yhdessä css-tyylitiedostojen kanssa. Tässä työssä HTML:llä on toteutettu kaikkien sivujen käyttäjälle näkyvät osat, paitsi toistoja esittävä käyrä, jonka tuottamiseen on käytetty jQueryä ja PHP:tä. HTML:stä on viimeaikoina yleistynyt HTML5 versio, jossa uusia ominaisuuksia ovat mm. Canvas ja Video ominaisuudet. Canvas ominaisuutta olisi kaikesti voitu käyttää myös käyrän piirtämiseen, mutta jQuery oli helppo ottaa käyttöön ja ajoi tässä työssä asiansa moitteetta.

jQuery ei ole ohjelmointikieli, vaan maailman suosituin JavaScript kirjasto (w3techs.com 2013a). jQueryllä voidaan toteuttaa animaatioita, valikoita ym. selainpäässä suoritettavia ominaisuuksia. JavaScript on ohjelmointikieli, jota käytetään enimmäkseen selaimissa, mutta myös työpöydän pienoishjelmissä

sekä esimerkiksi Unity3D:llä pelikehityksessä komentosarjoja ohjelmoidessa (Pyykölä 2011, 13).

3.3 MySQL

MySQL ei ole ohjelmointikieli eikä kehitysympäristö, vaan se on relaatiotietokantaohjelmisto. MySQL valittiin projektiin ilmaisuuden, helppokäyttöisyyden sekä hyvien valmiiden rajapintojensa takia. MySQL:llä on valmiit rajapinnat mm. Javalle, PHP:lle ja C#:lle. MySQL on nykyään Oraclen omistuksessa, alun perin sen ovat kehittäneet vuonna 1995 Suomalainen Michael Widenius sekä Ruotsalainen David Axmark. MySQL on tällä hetkellä maailman käytetyin avoimen lähdekoodin tietokantajärjestelmä (jelastic.com 2012) (mysql.com 2008).

MySQL:lle olisi myös graafinen käyttöliittymä, jolla luoda ja hallita tauluja sekä tietokantoja, mutta konsolipohjainen järjestelmäkin osaa luoda ASCII-grafiikalla selkeitä tauluja, joten kolmannen osapuolen tuottama phpMyAdmin sovellus jätettiin asentamatta. Kuviossa 4. on tulostettu kaikki testipalvelimen tietokannat, ”sports” tietokannan taulut sekä esitetty taulujen rakenteet ”describe” komennolla. Kuviossa 5. on nähtävissä mitä miltä kyseissä tauluissa oikea data näyttää. Huomionarvoista on että ”users” taulusta haetuissa arvoissa käyttäjätunnuksella ”mursu” näkyy salasanana merkkijono ”hashi”, joka on syötetty suoraan komentoriviltä tietokantaan, eikä suinkaan ole SHA-1 muotoinen tiiviste kuten muut taulussa näkyvät salasanat.

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| sports |
| test |
| weather |
+-----+
4 rows in set (0.00 sec)

mysql> use sports;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_sports |
+-----+
| results |
| users |
+-----+
2 rows in set (0.00 sec)

mysql> describe results;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| user | int(5) | NO | | NULL | |
| vatsa | int(5) | NO | | NULL | |
| selka | int(5) | NO | | NULL | |
| punnerrus | int(5) | NO | | NULL | |
| time | datetime | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> describe users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nick | varchar(12) | NO | | NULL | |
| passhash | varchar(150) | NO | | NULL | |
| id | int(11) | NO | PRI | NULL | auto_increment |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> █

```

KUVIO 4. MySQL-tietokanta ja taulujen rakenteet.

```
mysql> SELECT * FROM results LIMIT 10;
+----+-----+-----+-----+-----+
| user | vatsa | selka | punnerrus | time                |
+----+-----+-----+-----+-----+
| 1    | 10    | 20    | 10         | 2012-10-11 21:23:43 |
| 1    | 20    | 25    | 15         | 2012-10-11 21:23:49 |
| 1    | 25    | 30    | 25         | 2012-10-11 21:23:57 |
| 1    | 26    | 35    | 25         | 2012-10-11 21:24:02 |
| 1    | 27    | 33    | 28         | 2012-10-11 21:24:07 |
| 1    | 37    | 43    | 30         | 2012-10-11 21:24:16 |
| 1    | 37    | 47    | 32         | 2012-10-11 21:24:21 |
| 1    | 37    | 51    | 34         | 2012-10-11 21:24:28 |
| 1    | 39    | 54    | 36         | 2012-10-11 21:24:34 |
| 1    | 57    | 72    | 55         | 2012-10-12 22:21:51 |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> SELECT * FROM users LIMIT 10;
+----+-----+-----+
| nick      | passhash                                     | id |
+----+-----+-----+
| henull    | 1ecfff900d8d127fcbc3e31b9374888dcf7b5025 | 1  |
| tike      | 1ecfff900d8d127fcbc3e31b9374888dcf7b5025 | 2  |
| mursu     | hashi                                       | 3  |
| hemuli    | 1ecfff900d8d127fcbc3e31b9374888dcf7b5025 | 4  |
| hemuli2   | 1ecfff900d8d127fcbc3e31b9374888dcf7b5025 | 5  |
| tiina     | 86a8f327310f6be823a2729e756f46d323393a34 | 6  |
| hemuli3   | 5fd1266207e2ec42ca1c74e4dd1f70c78ffef43d | 7  |
+----+-----+-----+
7 rows in set (0.00 sec)

mysql> █
```

KUVIO 5. Esimerkkisyötteet tauluista.

4 PUHELIMEN OHJELMISTO

Puhelimelle suunnatun ohjelmiston kehittäminen lähti käyntiin perinteiseen tapaan kynä-paperi tekniikasta. Kirjattiin ylös, mitä ominaisuuksia ohjelmistolta vaaditaan, luotiin alustavaa prosessikaaviota sekä hahmoteltiin käyttöliittymää. Puhelimen ohjelmisto näytti paperilla erittäin selkeältä ja yksinkertaiselta toteuttaa, mutta muutamat toisilla kielillä yksinkertaiset asiat tekivät työstä paikoitellen jopa rasittavan työlästä. Työläimpänä mainittakoon kahden peräkkäisen asian, esimerkiksi napin ja tekstikentän sijoittaminen peräkkäin. Windows Phone kehitysympäristössä tämä tapahtuu vetämällä nappi tai tekstikenttä sinne minne se on tarkoitus sijoittaa, kun taas Androidin kehitysympäristössä tässä projektissa käytetyssä versiossa kehittäjä pääsi taistelemaan layout asetusten kanssa. Lopulta ongelma ratkaistiin juuri siten miten ei pitäisi tehdä, ja päädyttiin käyttämään jo valmiiksi vanhentunutta layout tekniikkaa nimeltä AbsoluteLayout.

4.1 Työn haasteet

Haastavin osuus oli vähäisen kokemuksen vuoksi saada puhelinohjelmistossa kaksi eri sivua, eli kaksi eri aktiviteettiä siten, että ensimmäinen vie tietonsa toiselle. Tämä vaadittiin käyttöliittymän takia, sillä mikäli kaikki nappulat ja tekstikentät olisivat olleet samalla sivulla, olisi näyttöä joutunut vierittämään, mikä olisi sotinut mahdollisimman yksinkertaisen ja selkeän ohjelmiston tarkoituksia vastaan.

Toinen ongelmakohta oli luoda Javassa ja PHP:ssä saman lopputuloksen tuottavat hash-funktiot. PHP:ssa oli jo valmiina erittäin helppokäyttöinen funktio, joten ongelma oli koittaa tuottaa Javalla vastaava. Ongelmaksi osoittautui lopulta kuitenkin vain tekstin encodaus.

Kolmas ja turhauttavin ongelma oli datan syöttäminen PHP-rajapinnalle. Ongelman teki turhauttavaksi se, että alkuperäinen funktio oli täysin toimiva,

samaten kuin seuraavat 30 testattua versiotakin, ja virhe ei ollut ollenkaan puhelimen päässä vaan PHP-raajapinnassa tavassa hakea syötetyt arvot.

4.2 Lähdekoodin tärkeimmät osat

MainActivity.java tiedostossa tärkein osio on Lähetä-nappia painettaessa tapahtuva handler() – funktio.

```
public void handler() {
    EditText punnerrus = (EditText)
    findViewById(R.id.punnerrusbox);
    EditText selka = (EditText)
    findViewById(R.id.selkalihasbox);
    EditText vatsa = (EditText)
    findViewById(R.id.vatsalihasbox);

    vatsat = Integer.parseInt(vatsa.getText().toString());
    selat = Integer.parseInt(selka.getText().toString());
    punnerrukset =
    Integer.parseInt(punnerrus.getText().toString());

    Intent myIntent = new Intent(this, settings.class);
    myIntent.putExtra("vatsat", vatsat);
    myIntent.putExtra("selat", selat);
    myIntent.putExtra("punnerrukset", punnerrukset);
    startActivity(myIntent);
}
```

Kyseinen funktio esittelee EditText kentät, ja hakee niistä käyttäjän syöttämän tekstin, tässä tapauksessa toistojen määrät, ja muodostaa niistä uuden abstraktin operaation johon sijoitetaan annetut arvot. Tämä siksi, että settings.java kykenee lukemaan ne toisessa aktiviteetissä. Toinen mahdollinen toteutustapa olisi luoda apuluokka, ja sijoittaa data väliaikaisesti sinne myöhempää käyttöä varten. Tämä osoittautui kuitenkin olettamusten vastaisesti vaikeammaksi sekä työläemmäksi kuin Intentin käyttö.

```
public void onCreate(final Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.asetukset);
    final Button login = (Button)
    findViewById(R.id.loginbutton);
    ...
    Bundle extras = getIntent().getExtras();
    vatsat = extras.getInt("vatsat");
    selat = extras.getInt("selat");
    punnerrukset = extras.getInt("punnerrukset");

    login.setOnClickListener(new View.OnClickListener() {
        public void onClick(final View view) {
            String pass = passbox.getText().toString();
            passhash = null;
            try {
                passhash = getHash(pass);
            }
        }
    });
}
```

```

    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    user = ubox.getText().toString();
    new Thread(new Runnable() {
        public void run() { // joten lähetetään omasta
            if (postData2(vatsat, selat, punnerrukset, user,
                passhash)) {
                settings.this.runOnUiThread(new Runnable() {
                    public void run() {
                        Toast.makeText(getApplicationContext(), "
                        Onnistui", Toast.LENGTH_LONG).show();
                    }
                });
            }
        }
    }).start();
}
});

private boolean postData2(int pun, int sel, int vat, String
user, String passhash) {
    ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();
    postParameters.add(new BasicNameValuePair("nick", user));
    postParameters.add(new BasicNameValuePair("hash",
passhash));
    postParameters.add(new BasicNameValuePair("punnerrus",
Integer.toString(pun)));
    postParameters.add(new BasicNameValuePair("vatsat",
Integer.toString(vat)));
    postParameters.add(new BasicNameValuePair("selat",
Integer.toString(sel)));

    String response = null;
    try {
        response =
CustomHttpClient.executeHttpPost("http://192.168.1.8/2/ains.ph
p", postParameters);
        String vastaus = response.toString();
        Log.d("mytag", vastaus);
        if (vastaus.equalsIgnoreCase("1")) {
            Log.d("mytag", "Tiedon tallentaminen onnistui!");
            return true;
        } else {
            Log.d("mytag", "vastaus: "+vastaus);
        }
    } catch (Exception e) {
        Log.d("mytag", "tänne");
        Log.d("mytag", e.toString());
    }
    return false;
}

public String getHash(String pass) throws
NoSuchAlgorithmException {
    MessageDigest md;
    String message = pass;
    try {
        md = MessageDigest.getInstance("SHA-1");
        md.update(message.getBytes());
        byte[] mb = md.digest();
        String out = "";
        for (int i = 0; i < mb.length; i++) {
            byte temp = mb[i];
            String s = Integer.toHexString(new Byte(temp));
            while (s.length() < 2) {
                s = "0" + s;
            }
        }
    }
}

```

```

    }
    s = s.substring(s.length() - 2);
    out += s;
}
return out;
} catch (NoSuchAlgorithmException e) {
return null;
}
}
}

```

Data on suotavaa lähettää omasta threadistaan, jottei käyttäjälle näkyvä ikkuna näytä jumittuvan. Kun data lähetetään rajapinnalle omasta threadista, se toimii ikään kuin taustalla, ja käyttäjälle näkyvä ikkuna voi jatkaa muita sille määrättyjä toimenpiteitä. postData2() – funktio luo alussa uuden listamuuttujan, ja lisää siihen MainActivity.javassa Intenttiin syötetyt muuttujat. Nämä muuttujat syötetään NameValuePair muodossa, jossa ensimmäisenä syötetään arvon nimi, esimerkiksi "nick" jota käytetään käyttäjätunnuksen tallentamiseen, ja kyseiselle nimelle syötetään perään arvo, joka on jo aiemmin haettu Intentistä. Nämä NameValueParit syötetään listaan, joka annetaan CustomHttpClientille tietojen rajapintaan syöttämistä varten. Mikäli CustomHttpClient.executeHttpPost funktio palauttaa viestinä numeron 1, on käyttäjä syöttänyt toimivan käyttäjätunnus ja salasana parin, ja rajapinta on onnistunut syöttämään sille annetut toistot tietokantaan.

Funktio getHash(String pass) palauttaa String – muotoisen muuttujan, jonka arvo on funktiolle syötetyn tekstin SHA-1 tiivistefunktio. Tiivistefunktiota käytetään jotta salasanaa ei tarvitsisi säilyttää tai siirtää selvätekstisenä. Yksisuuntaisen tiivistefunktion eli hash-funktion tarkoitus on tuottaa viestistä kiinteän mittainen, lyhyt "tiivistelmä", joka edustaa koko viestiä sikäli, että on vaikea löytää mitään muuta viestiä, jolla olisi samanlainen tiiviste (sec.cs.tut.fi 2012.). Tämä tosiasia yhdistettynä riittävän pitkään salasanaan tekee salasanan siirtämisestä verkon yli riittävän turvallista.

SHA-1 on projektin toteuttamisvaiheessa jo todettu vanhentuneeksi, mutta ottaen huomioon että käyttäjistä ei säilytetä edes oikeaa nimeä, sähköpostiosoitetta tai muuta yksityistä tietoa, on SHA-1 tiivistefunktio tähän tarkoitukseen riittävä. Suuremman levityksen kannalta jatkokehityksessä on kuitenkin otettava tarkemmin huomioon käyttäjien tietoturva, ja SHA-1 voidaankin korvata esimerkiksi SHA-3 tiivistefunktiolla, jonka algoritmiksi valittiin National Institute of

Standards and Technology (NIST) järjestämässä kilpailussa Keccak. Keccakia on ollut kehittämässä muun muassa Joan Daemen, joka oli toinen Rijndaelin kehittäjistä. Rijndael valittiin Advanced Encryption Standard (AES) salauksen algoritmiksi vuonna 2001, ja on nykyisin yleisimmin käytössä oleva salausalgoritmi (kellermansoftware.com 2008).

5 PALVELIN

Työssä palvelimella oli merkittävä rooli, joten sen valinnassa täytyi olla tarkka. Palvelimen käyttöjärjestelmän osalta päädyttiin helpon seulonnan jälkeen Linux-palvelimeen, koska siitä ei aiheutunut ylimääräisiä kustannuksia. Tutkimusyhtiö IDC:n tekemän tutkimuksen mukaan uudet palvelinten käyttöjärjestelmälisenssien osuudet vuonna 2000 olivat seuraavia:

TAULUKKO 1. Palvelinten käyttöjärjestelmät vuonna 2000 (Rantala 2000.)

Microsoft	41 %
Linux	27 %
Muut Unix-muunnelmat	13,9 %
Novell Netware	13,8 %

Tammikuussa 2013 tilanne oli muuttunut siten, että Unixiin pohjaavilla käyttöjärjestelmillä oli hallussa 64,9 % osuus, ja Microsoftin Windows tuotteilla oli 35,1 %.(W3Techs.com 2013b.)

5.1 Palvelimen rautapuoli

Projektissa palvelimelle ei asetettu raskaita vaatimuksia. PHP, Apache2 ja MySQL tietokanta kahden henkilön testiryhmälle ei juuri vaadi resursseja, joten palvelimena toimi Lenovo X60 – kannettava tietokone. Koneesta löytyy Intel Core 2 Duo 2.2GHz, 2GB DDR2 muistia, 160GB kovalevy ja 1Gbps verkkokortti. Palvelinkoneen resurssit riittävät helposti muutaman käyttäjän testiryhmän tarpeisiin, ja tarvittaessa pienen ryhmän julkiseen testaukseenkin.

5.2 Käyttöjärjestelmä

Kubuntu on periaatteessa Ubuntu, mutta KDE-työtilalla varustettuna. Kubuntu on Canonicalin rekisteröimä tuotenimi, samoin kuten Ubuntu. Debian sekä Ubuntu toimivat usein palvelimina (W3Techs.com 2013c.) varsinkin pienissä yhteisöissä ja

testiympäristöissä. Projektissa käytetyt ohjelmistot asennettiin tasksel-työkalulla, joka on tarkoitettu pakettikokoelmien asentamiseen, kuten esimerkiksi LAMP. Tasksel piti asentaa erikseen komennolla `sudo apt-get install tasksel`. Tämän jälkeen Tasksel:n käyttö onnistui komennolla `sudo tasksel install lamp-server`, jonka jälkeen ruutuun ilmestyy selkeät ohjeet joita noudattamalla palvelimen saa täysin toimivaan perusasetustilaan. Pääkäyttäjän tunnusta (root) suositellaan käyttämään ainoastaan silloin, kun sitä todella tarvitaan (Rantala 2003, 32).

5.3 PHP-rajapinta

PHP-rajapinta on yksi työn tärkeimpiä osia. Rajapinta vastaanottaa Android-sovellukselta viestin, josta löytyy parametreina `$_POST[""]` funktiolla haettavat arvot, kuten käyttäjätunnus, salasanan tiiviste sekä suoritusten toistomäärät. Käyttäjätunnusta ja salasanaa verrataan tietokantaan, ja sopivan parin löytyessä palautetaan paria vastaava id-numero, jolla yksilöidään suoritukset oikealle henkilölle. Kaikki sovelluksen syöttämät arvot tarkistetaan vielä ennen SQL-kyselyä sen varalta, että käyttäjä koittaisi syöttää haitallisia kyselyitä.

Mikäli rajapinta palauttaa käyttäjätunnuksella ja salasanalla ID:n, joka on eri kuin nolla, muodostetaan SQL-kyselylause, jolla syötetään toistojen määrät sekä aiemmin saatu ID tietokantaan. Näitä tuloksia voidaan myöhemmin hakea tietokannasta käyttäjä ID:n perusteella, ja saaduista tuloksista voidaan piirtää graafi JavaScript kirjaston avulla.

```
<?php
$host="localhost"; // Host name
$username="kissa"; // Mysql username
$password="silakka"; // Mysql password
$db_name="sports"; // Database name
$table_name="users"; // Table name

mysql_connect("$host", "$username", "$password") or die("cannot
connect");
mysql_select_db("$db_name") or die("cannot select DB");

$myusername = htmlspecialchars($_POST["nick"]);
$mypassword = htmlspecialchars($_POST["hash"]);
$pun = htmlspecialchars($_POST["punnerrus"]);
$vat = htmlspecialchars($_POST["vatsat"]);
$sel = htmlspecialchars($_POST["selat"]);

$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
```

```

$pun = stripslashes($pun);
$vat = stripslashes($vat);
$sel = stripslashes($sel);

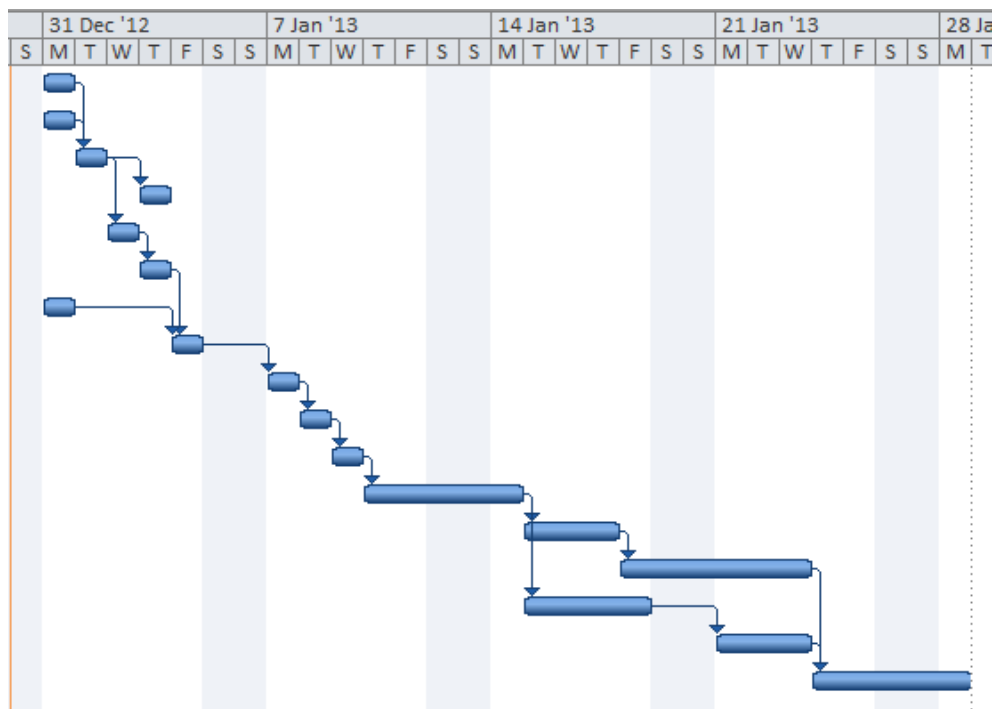
$pun = mysql_real_escape_string($pun);
$vat = mysql_real_escape_string($vat);
$sel = mysql_real_escape_string($sel);
$username = mysql_real_escape_string($username);
$password = mysql_real_escape_string($password);

$sql="SELECT id FROM $tbl_name WHERE nick='$username' and
passhash='$password'";
$result=mysql_query($sql);
$count=mysql_num_rows($result);
while($row = mysql_fetch_assoc($result)){
    $id = $row['id'];
}
if($count==1 && $id != 0){
    $sql = sprintf("INSERT INTO results
(user,vatsa,selka,punnerrus,time) VALUES
('%d','%d','%d','%d',NOW())", $id, $vat, $sel, $pun);
    if(mysql_query($sql)){
        echo "1";
    }
    else {
        echo $sql;
        echo ":(";
    }
}
else{
    echo "0";
}
?>

```


6 TYÖN TOTEUTUS

Koko projekti lähti puhelimen ohjelmiston tapaan liikkeelle kynällä ja paperilla. Näin oli paljon helpompi hahmottaa kaikki eri työvaiheet. Todella epäselvästä alustavasta paperiversiosta projektin toimintasuunnitelma siirrettiin Microsoft Officen Project ohjelmistoon, jossa suunnitelmasta sai helposti muokattua selkeämmän näköisen, lisättyä huonoja arvioita kestoista ja arvioiduista valmistuspäivämääristä. Käytännössä kestoilla ja päivämäärien arvioilla ei ollut mitään tekemistä todellisuuden kanssa. Tarkoituksena oli tuottaa tehtäväkaavio, josta näkee missä järjestyksessä asiat kannattaa tehdä. Kuvioissa 6. ja 7. on nähtävissä työn luonnostellut eri vaiheet.



KUVIO 6. Projektin työvaiheet ja niiden yhteydet toisiinsa.

Task Mode	Task Name	Duration	Start	Finish	Predecessors
1	Suunnittele mitä urheilusuorituksia halutaan merkitä	1 day	Mon 31.12.12	Mon 31.12.12	
2	Listaa palvelimelta vaadittavat ominaisuudet ja ohjelmistot	1 day	Mon 31.12.12	Mon 31.12.12	
3	Suunnittele tietokantamalli	1 day	Tue 1.1.13	Tue 1.1.13	1;2
4	Suunnittele nettisivuston rakenne	1 day	Thu 3.1.13	Thu 3.1.13	3
5	Suunnittele puhelimen ohjelmistolta vaadittavat ominaisuudet	1 day	Wed 2.1.13	Wed 2.1.13	3
6	Suunnittele puhelimen ohjelmiston käyttöliittymä	1 day	Thu 3.1.13	Thu 3.1.13	5
7	Listaa ja tutki palvelimen tietoturva koskevia asioita	1 day	Mon 31.12.12	Mon 31.12.12	
8	Korjaa mahdolliset muutokset aikaisempiin suunnitelmiin	1 day	Fri 4.1.13	Fri 4.1.13	7;6
9	Asenna palvelimelle käyttöjärjestelmä	1 day	Mon 7.1.13	Mon 7.1.13	8
10	Asenna palvelimelle vaadittavat ohjelmistot	1 day	Tue 8.1.13	Tue 8.1.13	9
11	Muodosta palvelimelle tietokanta	1 day	Wed 9.1.13	Wed 9.1.13	10
12	Rakenna kehitystä tukeva, alustava PHP-rajapinta	3 days	Thu 10.1.13	Mon 14.1.13	11
13	Androidille valmiit toimivat funktiot	3 days	Tue 15.1.13	Thu 17.1.13	12
14	Alustava toimiva versio Android ohjelmistosta valmiiksi	4 days	Fri 18.1.13	Wed 23.1.13	13
15	Toteuta nettisivusto suunnitelman pohjalta	4 days	Tue 15.1.13	Fri 18.1.13	12
16	Korjaa nettisivustolta löytyvät tietoturva-aukot	3 days	Mon 21.1.13	Wed 23.1.13	15
17	Tee viimeistelevät muutokset ja korjaukset puhelimeen ja palvelimeen.	3 days	Thu 24.1.13	Mon 28.1.13	16;14

KUVIO 7. Projektin työvaiheet suoritusjärjestyksessä ja tehtäväkuvaukset.

Aluksi piti valita, mitä urheilusuorituksia haluttiin merkitä. Urheilusuorituksia pystyisi myöhemmin halutessa lisäämään, mutta alustavasti oli hyvä valita kuitenkin ainakin kolme kaavioiden ilmeen piristämiseksi. Valituiksi päätyivät vatsalihasliikkeet, selkälihasliikkeet sekä punnerrukset. Valintaperusteena käytettiin sitä, että perinteisinä ja hyvin tunnettuina liikkeinä nämä kolme tuskin tarvitset suurempia ohjeita miten toimia, useasti toistettuina maksimitoistojen määrä kasvaa helposti (onehundredpushups.com, 2013) sekä jatkuvalla treenaamisella on selkeitä hyötyvaikutuksia monille istumatyössä tarvittaville lihasryhmille. (Turunen, P. 2012)

Toisena tarvitsi listata palvelimelta vaadittavat ominaisuudet ja ohjelmistot. Tässä projektissa ei palvelimelta vaadittu suuria, joten palvelimena toimi kannettava tietokone. Puhelimen ohjelmistoista lisää kappaleessa 5. Puhelimen suunnittelun jälkeen oli suotavaa suunnitella alustava tietokantamalli. Projektissa tietokantaan tuli vain kaksi taulua: yksi jossa on käyttäjät ja kirjautumiseen vaadittavat tiedot, sekä toinen taulu jossa on käyttäjän yksilöivä tieto, suorituspäivämäärä sekä jokaisesta liikkeestä suoritettut toistot. Alustavasti mietinnän alle jäi vielä voidaanko yhdeltä päivältä merkitä useampia suorituksia? Lopuksi päädyttiin kuitenkin ratkaisuun, että suoritukset merkitään vain kerran päivässä tietokantaan. Tähän päädyttiin sen vuoksi, että graafi on selkeämpi jos jokaiselta päivältä on vain yksi maksimitoistosuoritus liikettä kohden. Tarvittavien pohjasuunnitelmien jälkeen, projektissa siirryttiin suunnittelemaan www-sivustolta sekä puhelinohjelmistolta

vaadittavia ominaisuuksia, sekä näiden käyttöliittymiä. Käyttöliittymän tuli olla selkeä, minimalistinen ja nopea ilman turhia lisäviritelmiä niin selaimessa kuin puhelinohjelmistossakin. Suunnittelu lähti paperitasolta, eteni raakavedoksiin Microsoft Paint – ohjelmistoon, josta päätyikin jo sitten seuraavaksi koodiin.

Toiseksi viimeisenä vaiheena oli tietoturva-aspektin tuominen mukaan suunnitteluun. Alun perin tarkoitus ei ollut sotkea kehitystä liian tarkalla tietoturvaseulonnalla, mutta projektin edetessä tietoturva-aspekti tuli esiin jokaisessa vaiheessa, jokaisessa netistä haetusta neuvossa sekä lähteenä käytetystä kirjasta, joten siltä ei oikein voinut välttyä. Projektissa mukana ollut palvelin testattiin NMAP-ohjelmistolla ylimääräisten palveluiden ja avonaisten porttien varalta. Osaltaan tietoturva jäi myös jatkokehityksen ongelmaksi, mikäli palvelinta siis lähdetäisiin siirtämään suurempaan saliin tai siirtämään esimerkiksi Amazonin palveluun. Jatkokehityksestä lisää kappaleessa 8.

7 TULOKSET

Projektin lopputuotteena tuli täysin toimiva raakaversio kuntoilusovelluksesta. Kaikki projektin osasovellukset toimivat käytännön tasolla asti, mutta ovat vielä kehitysvaiheella olevan näköisiä. Alussa paperille suunniteltu ulkoasu muistutti lopputulosta suuresti, niin puhelinsovelluksessa kuin nettisivustollakin. Nettisivustosta jäi puuttumaan lähinnä värimaailma ja hieman modernimpi ulkoasu, vaikkakin musta ja valkoinen ovat aina muodissa (Edman, V. 2013). Kuviossa 8 nähtävissä mobiilisovelluksen viimeisin versio.

Kuvioissa 2. ja 3. on nähtävissä sisäänkirjautuminen ja rekisteröityminen, jotka ovat selkeitä ja pelkistettyjä. Työssä ei missään välissä ladata suuria kuvia tai vaadita palvelimelta suurta laskentatehoa, mikä tarkoittaa että kevyemmälläkin palvelinratkaisulla voidaan pyörittää palvelua suuremmalle ryhmälle. Ohjelmiston palvelinpään toteutus kuormittaa palvelinpäätä todella vähän, johtuen pääosin minimaalisista palveluista sekä sivuston karusta ulkoasusta. Palvelinpää voidaan käytännössä tarvittaessa siirtää minne tahansa, joten ongelmia kaistan riittävydestä tai palvelinkoneen tehoista ei pitäisi tulla suuremmallakaan käyttäjämäärällä. Työ lähdekoodeineen on saatavissa ainakin vuoden verran osoitteessa <http://henull.fi/opinnaytetyo/>.

The image shows two screenshots of a mobile application. The top screenshot, titled 'EpicKuntoiluSovellus', displays a list of items with their respective values: Punnerrukset (25), Selkälihakset (70), and Vatsalihakset (20). A 'Send' button is located below the list. The bottom screenshot, titled 'Kuntoilusovellus', shows a login form with fields for 'Käyttäjätunnus' (username) and 'Salasana' (password), and a 'Login and send' button.

Punnerrukset	25	Käyttäjätunnus	<input type="text"/>
Selkälihakset	70	Salasana	<input type="password"/>
Vatsalihakset	20		

Send

Login and send

KUVIO 8. Pääikkuna sekä lähetyssikkuna.

8 HUOMIOITA JATKOKEHITYKSEEN

Yleisesti jatkokehityksen kannalta olisi tärkeää lisätä ohjelmistokokonaisuuteen lisää ominaisuuksia sekä parantaa käytettävyyttä ja käyttäjän tietoturvaa. Nykyinen palvelin asettaa omat rajoitteensa jatkokehitykselle, joten projektin jatkuessa olisi luonnollista siirtää tietokanta ja sivustot tehokkaammalle palvelimelle.

8.1 Järjestelmän käytettävyys

Käytettävyyttä tulee aina kehittää, ohjelmisto tarvitsisi huomattavasti suuremman testiyleisön jolta saada palautetta ja kehitysideoita. Ohjelmiston ulkoasua voisi selkeyttää ja muutoinkin parantaa mukavammaksi käyttää. Graafinen toteutus on nykyisessä versiossa täysin lapsen kengissä, ja nettisivusto varsinkin vaatisi kipeästi graafikon kosketusta.

Puhelimen ohjelmiston valmistumisen jälkeen Androidin kehitystyökalujen mukana on tullut uusia valmiita pohjia. Muutama näistä valmiista pohjista toimisi oikeastaan jo sellaisenaan paremmin ohjelman tarkoitukseen kuin nykyinen malli, ja pienellä muokkaamisella niistä saisi jopa kaupallisesti menestyvän näköisiä, vähintään kuitenkin ainakin selkeämmän näköisiä.

Projektissa toteutettu versio testattiin vain yhdellä ulkopuolisella käyttäjällä, mutta ennen laajempaa levitystä tuotetta tulisi ehdottomasti testata suuremmalla käyttäjäryhmällä vikojen ja ongelmien minimoimiseksi, sekä käyttäjäkokemuksen parantamiseksi.

Käytettävyyttä voisi helpottaa myös tuomalla matkapuhelinohjelmistoon ominaisuuden, joka tallentaa käyttäjän käyttäjätunnuksen ja salasanan, jotta niitä ei tarvitsisi joka kerta kysyä erikseen. Tässä olisi toki huomioitava tietoturva aspekti, mikä olisi kuitenkin kohtuullisen helppoa, sillä salasananakin voisi tallentaa

suoraan hash-muodossa, joten sitä ei voisi ainakaan suoraan lukea puhelimen muistista.

8.2 Tietoturva

Historia on opettanut että koskaan ei tule aliarvioida aikaa, rahaa eikä vaivaa mitä joku on valmis käyttämään löytääkseen aukon järjestelmästä. On kannattavampaa odottaa pahinta ja yliarvioida hyökkäjä. On kannattavampaa olettaa että teknologia pystyy lähitulevaisuudessa tehdä asioita joihin se ei tänä päivänä kykene. Kehittäjän tulee luoda virhemarginaali, jossa vahinko on siedettävissä. (Schneier, B. 1997)

Tietoturvassa on aina parannettavaa; aukotonta järjestelmään ei ole eikä tulla ikinä tekemäänkään. Vähintä mitä järjestelmän tietoturvan kannalta voi tulevaisuudessa tehdä on päivittää palvelimen ohjelmistot aina uusimpaan turvalliseen versioon. Ylläpitäjän tulee myös pitää silmällä, onko PHP:ssä tai Javassa uusien versioiden myötä suositeltu kehittäjiä olemaan käyttämättä joitakin osia, joita nykyisessä järjestelmässä käytetään, ja korjata nämä osat tarvittaessa uuden suosituksen mukaiseksi.

Palvelimen tietoturvasta saisi itsessään jo väitöskirjan tai kirjasarjan, mutta huomionarvoista olisi jatkokehityksen kannalta myös palvelimen fyysinen turvallisuus. Vaikka ulkoapäin kaikki SQL-kyselyt saisikin siistittyä hyökkäyksistä ja ylimääräiset portit olisivat kiinni, mikäli hyökkäjä pääsee fyysisesti palvelimeen käsiksi, ovat tiedot yleensä helposti saatavissa.

Yksi tärkeimmistä korjauksista julkisesti käyttöön tulevaan versioon olisi ehdottomasti käyttäjien salasanojen suolaaminen. Suolaaminen tarkoittaa, että salasanaan liitetään ennen tiivisteeseen ottamista jokin satunnainen merkkijono, joka on jokaiselle salasanalle eri. Tämän johdosta mikäli hyökkäjä saa palvelimelta tietokannan käsiinsä, ei hän kykene hyödyntämään käyttäjien salasanoina ainakaan nykyisellä palvelinteholla järjellisessä ajassa, sillä hän ei kykene hyödyntämään mitään yhtä rainbow tablea murtaessaan salasanoina. Mikäli jokainen salasana on

suolattu samalla merkkijonolla, tarvitsee hyökkääjän muodostaa vain yksi uusi rainbow table, ja hän voi sitä käyttämällä murtaa salasanat. Tämä toki hidastaa hieman, mutta turvallisen ja turvattoman kirjautumisen ero PHP:ssä on noin kolme riviä. (Defuse Security 2013)

8.3 Järjestelmän ominaisuudet

Projektiin voi jatkossa lisätä enemmän harjoitusliikkeitä, kuten leuanvedot, kyykkyhyppyt, venytysliikkeitä tai oikeastaan mitä tahansa vastaavia harjoituksia. Pelkästään näiden lisäksi voisi sovellukseen sekä sivustolle lisätä ohjeet näiden harjoitusten tekemiseen oikein, mutta tämä vaatisi fysioterapeutin tarkempaa haastattelua, ja mahdollisesti videon tai kuvien ottamista. Väärin tehtynä venytykset tai lihasharjoitukset voivat pahimmassa tapauksessa olla jopa haitallisia (Tohtori.fi, 2010).

Mobiilisovellukseen voisi lisätä myös automaattiset muistutukset. Puhelin voisi muistuttaa aina haluttuun aikaan käyttäjää tekemään päivittäiset liikkeet. Kallistukseen tai kiihtymiseen perustuvaa laskuria voisi harkita joissakin liikkeissä, kuten esimerkiksi punnerruksissa, vatsalihaksissa sekä kyykkyhyppyissä.

LÄHTEET

- Arvonen, S. Runsas istuminen lihottaa. Www-dokumentti. Saatavissa: <http://www.keventajat.fi/hyvinvointi/liikunta/tauvoja-istumiseen>. Luettu 20.1.2013.
- Holmala, E. Vatsa litteäksi. Www-dokumentti. Saatavissa; <http://kiloklubi.fi/artikkelit/Vatsa-litteäksi/770/>. Luettu 20.1.2013.
- Kuntoguru.com. Www-dokumentti. Saatavissa: <http://kuntoguru.com/fyysinen.html>. Luettu 20.1.2013.
- Hundredpushups.com. Www-dokumentti. Saatavissa: <http://hundredpushups.com/why.html>. Luettu 20.1.2013.
- Stackoverflow.com. Www-dokumentti. Saatavissa: <http://stackoverflow.com/questions/1276/how-big-can-a-mysql-database-get-before-performance-starts-to-degrade>. Luettu 8.3.2013.
- Hardy, Q. Why iPhone Shoppers Buy More Apps. Www-dokumentti. Saatavissa: <http://bits.blogs.nytimes.com/2012/01/16/why-iphone-shoppers-buy-more-apps/>. Luettu 20.12.2012.
- Viswanathan, P. How to Make Money by Selling Free Apps. Www-dokumentti. Saatavissa: <http://mobiledevices.about.com/od/marketingapps/ht/How-To-Make-Money-By-Selling-Free-Apps.htm>. Luettu 4.3.2013.
- Eclipse.org. Www-dokumentti. Saatavissa: <http://www.eclipse.org/org/>. Luettu 10.3.2013.
- Niskanen, P., Kontio, M., Vierimaa, K. 2000. Enterprise Java. Helsinki: IT Press.
- W3Techs.com. 2013a. Www-dokumentti. Saatavissa: http://w3techs.com/technologies/overview/javascript_library/all. Luettu 23.3.2013.
- Pyykölä, E. 2011. Creating a chat room using Unity Game Development Tool. Opinnäytetyö. Centria ammattikorkeakoulu. Tietotekniikan koulutusohjelma.
- jelastic.com. 2012. Www-dokumentti. Saatavissa: <http://blog.jelastic.com/2012/02/23/open-source-database-market-share-within-jelastic-february-2012/>. Luettu 24.3.2013.
- mysql.com. 2008. Www-dokumentti. Saatavissa: <http://www.mysql.com/why-mysql/marketshare/>. Luettu 24.3.2013.
- W3Techs.com. 2013b. Www-dokumentti. Saatavissa: http://w3techs.com/technologies/overview/operating_system/all. Luettu 20.1.2013.
- sec.cs.tut.fi, 2012. Www-dokumentti. Saatavissa: <http://sec.cs.tut.fi/maso/materiaali.php?id=198>. Luettu 10.3.2013.

kellermansoftware.com. 2008. What is the strongest encryption algorithm? Www-dokumentti. Saatavissa: <http://www.kellermansoftware.com/t-ArticleStrongestAlgo.aspx>. Luettu 26.3.2013.

Rantala, A. 2003. LINUX. Jyväskylä: Docendo

W3Techs.com. 2013c. Www-dokumentti. Saatavissa: http://w3techs.com/technologies/history_details/os-linux. Luettu 8.3.2013.

Turunen, P. 2013. Www-dokumentti. Saatavissa: <http://www.taloussanomat.fi/tyoja-elama/2012/12/19/toissa-tarvitaan-voimaa-ala-laihduta-lihaksia/201243897/139>. Luettu 8.3.2013.

Edman, V. 2013. Black And White Is Always Right. Www-dokumentti. Saatavissa: <http://www.theblogazine.com/2013/03/black-and-white-is-always-right-2/>. Luettu 24.3.2012.

Schneier, B. 1997 Why Cryptography Is Harder Than It Looks. Pdf-tiedosto. Saatavissa: <http://www.schneier.com/essay-037.pdf>. Luettu 5.3.2013.

Defuse Security. 2013. Salted Password Hashing – Doing it Right. Www-dokumentti. Saatavissa: <http://crackstation.net/hashing-security.htm>. Luettu 12.03.2013.

Tohtori.fi, 2010. Www-dokumentti. Saatavissa: <http://www.tohtori.fi/?page=0179162&id=5986992> Luettu 2.3.2013.