

Jonathan Aalto

**TIETOJEN KERÄYKSEEN SUUNNITELTU MOBIILISOVELLUS
ESTIMOTE BEACON- JA NEARABLE-LAITTEILLE**

**TIETOJEN KERÄYKSEEN SUUNNITELTU MOBIILISOVELLUS
ESTIMOTE BEACON- JA NEARABLE-LAITTEILLE**

Jonathan Aalto
Opinnäytetyö
Syksy 2016
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehitys

Tekijä(t): Jonathan Aalto

Opinnäytetyön nimi: Tietojen keräykseen suunniteltu mobiilisovellus Estimote Beacon- ja Nearable-laitteille

Työn ohjaaja(t): Eero Nousiainen

Työn valmistumislukukausi ja -vuosi: Syksy 2016

Sivumäärä: 52

Työn tavoitteena oli luoda kokonaisuus, joka koostui verkkosivuista, mobiilisovelluksesta ja palvelimesta. Mobiilisovelluksessa tavoitteena oli kerätä sensori- ja laitetietoja Estimote Beacon- ja Estimote Nearable -laitteista, jotka ovat avaimenperän kokoisia lähettäjiä. Verkkosivuilla tavoitteena oli luoda sisäänkirjautumisprosessi, jonka takana verkkosivuilla oli tarkoitus esittää kerätyt sensori- ja laitetiedot taulukkoina käyttäjälle. Palvelimen tavoitteena oli vastaanottaa mobiilisovelluksesta kerätyt tiedot ja sijoittaa ne palvelimelle luotuihin tietokantoihin, josta ne pystyttäisiin poimimaan verkkosivujen taulukoihin katselmoitaviksi.

Työssä asennettiin ensin käyttöympäristö eli kehitykseen tarvittavat ohjelmistot, jonka jälkeen työssä tutustuttiin Estimote Beacon- ja Nearable -laitteiden tekniikoihin. Ensimmäisenä vaiheena oli luoda sisäänkirjautuminen verkkosivuille, jossa hyödynnettiin palvelimella olevaa tietokantaa, jossa tulevat sensori- ja laitetiedot pystyttäisiin esittämään käyttäjälle. Toisena vaiheena luotiin palvelin, johon kerättävät tiedot ja sisäänkirjautumiseen tarvittavat tiedot pystyttäisiin säilyttämään. Viimeisenä vaiheena luotiin mobiilisovellus, joka kerää tarvittavat tiedot Estimote Beacon- ja Nearable -laitteista ja lähettää kerätyt tiedot palvelimelle.

Lopputuloksena saatiin toimiva ohjelmakokonaisuus, jossa yhdistyi palvelin, jonka sisällä tietokanta, mobiilisovellus ja verkkosivut. Mobiilisovelluksesta pystyttiin lähettämään kerättyjä tietoja palvelimelle, josta verkkosivut pystyivät ne helposti keräämään ja näyttämään käyttäjälle sisäänkirjautumisen jälkeen tulevassa taulukossa.

Asiasanat: Estimote, Bluetooth Low Energy, mobiilisovellus, verkkosivut, palvelin

ALKULAUSE

Haluan erityisesti kiittää opinnäytetyönohjaajaani Eero Nousiaista, jonka loistavalla ohjauksella sain opinnäytetyöni kunnialla valmiiksi ja joka auttoi minua rakentamaan ja testaamaan työtäni. Aihe oli hyvin mielenkiintoinen ja sain paljon käytännönkokemusta useista eri tekniikoista, jotka auttoivat minua kehittymään ohjelmoitsijana. Lisäksi haluan myös kiittää Tuula Hopeavuorta, joka jaksoi käydä tekstini läpi ja auttoi minua korjaamaan virheet tekstistäni ja näiden lisäksi auttoi minua luomaan kieliopillisesti parempaa tekstiä.

Oulussa 9.10.2016

Jonathan Aalto

SISÄLLYS

TIIVISTELMÄ	3
ALKULAUSE	5
SISÄLLYS	6
SANASTO	8
1 JOHDANTO	10
2 KÄYTETYT TEKNOLOGIAT	11
2.1 Estimote Beacon	11
2.1.1 Estimote Proximity Beacon	11
2.1.2 Estimote Location Beacon	12
2.2 Estimote Nearable	13
2.3 Bluetooth Smart	14
2.4 ARM Cortex M0 -proessori	15
3 KÄYTETYT OHJELMISTOT	16
3.1 XAMPP	16
3.2 CodeIgniter	17
3.3 Brackets	18
3.4 Android Studio	19
3.5 PHP	20
3.6 HTML ja CSS	21
3.7 Estimote Cloud	21
4 TYÖN TAVOITE JA AIKATAULU	23
4.1 Työn tavoite	23
4.2 Työn aikataulu	24
5 TYÖN KÄYTTÖYMPÄRISTÖN ASENNUS JA KÄYTTÖÖNOTTO	28
5.1 Android Studion asennus ja projektin luominen	29
5.2 XAMPP-ohjelman asennus	35
5.3 Tietokannat ja rajapinnat	41
5.3.1 Login-tietokanta	41
5.3.2 Beacon_nest-tietokanta	43
5.3.3 JSON-tekstin lähettäminen palvelimelle	43
5.3.4 JSON-tekstin vastaanottaminen palvelimella	44

5.4 CodeIgniter-viitekehiksen käyttöönotto ja testaaminen	45
6 TYÖN TULOKSET	46
6.1 Verkkosivut	46
6.2 Mobiilisovellus	47
6.3 Tietokantakokonaisuus	48
7 YHTEENVETO	49
LÄHTEET	50

SANASTO

Activity	Android Activity on Android-sovelluksen komponentti, jossa käyttäjä voi olla vuorovaikutuksessa ohjelman kanssa. Esimerkiksi Android-sovelluksen osa, jossa käyttäjä voi painaa painonappeja, jotka suorittavat erilaisia komentoja.
HTTP	Hypertext Transfer Protocol on World Wide Web -verkon käyttämä protokolla tiedon muotoilemiseen ja tiedonsiirtoon.
Indoor Location SDK	Indoor Location Software Development Kit on Estimote-ten työkalu, jolla voidaan kartoittaa huone tai rakennus käyttäen hyväksi Estimote Beacon -laitteita.
IP-osoite	Internet Protocol -osoite on laitteelle annettu tunnistin, jonka avulla laite voidaan tunnistaa verkossa.
IPv4	Internet Protocol version 4 on internetprotokolla, joka on neljän tavun mittainen osoite, jota käytetään laitteiden tunnistamiseen internetissä.
JsonObject	JsonObject on objekti, johon voidaan tallentaa erilaisia arvopareja, kuten merkkijono (string) -tyyppinen avainkenttä, jolla on kokonaisluku (int) -tyyppinen arvo.
Localhost	Selainosoite, jolla viitataan kyseisen tietokoneen paikalliseen osoitteeseen.
Major	IBeacon-protokollalle ominainen tunniste, jolla voidaan jakaa Estimote Beacon -laitteita erityyppisiin ryhmiin.
Minor	Samankaltainen tunniste kuin Major, mutta tällä voidaan jakaa Estimote Beacon -laitteita kolmanteen ryhmätyyppiin.

Pysical Web	Konsepti, joka mahdollistaa lähellä olevien Bluetooth Low Energy -laitteiden lähettämien nettiosoitteiden avaamisen selaimella.
UUID	Universally Unique Identifier on tunnistus, jolla voidaan erottaa laitetta toisistaan ohjelmakoodissa.
URL	Uniform Resource Locator on verkkoresurssi, joka viittaa tiettyyn polkuun tietokoneella tai internetissä.
W3C	The World Wide Web Consortium on kansainvälinen organisaatio, jonka tarkoitus on ylläpitää ja edistää internetin eli World Wide Web -verkon erilaisia standardeja.

1 JOHDANTO

Työn tavoitteena oli luoda ohjelmakokonaisuus, joka koostui kolmesta osa-alueesta: verkkosivut, mobiilisovellus ja tietokantajärjestelmä. Pääpaino työssä oli perehtyä Estimote Beacon- ja Estimote Nearable -laitteisiin, jotka ovat avaimenperän kokoisia laitteita, jotka sisältävät erilaisia sensoreita, kuten lämpötilasensorin ja läheisyysensorin. Laitteet lähettävät kerättyjä sensoritietoja ja laitteen laitetietoja eteenpäin Bluetooth Low Energy -tiedonsiirtomenetelmällä, joka on uuden tyyppinen langaton lyhyen kantaman radiotekninen yhteys.

Verkkosivujen tavoitteena oli luoda vakaat verkkosivut Estimote-laitteista kerätävien sensori- ja laitetietojen näyttämiseen. Tarkoituksena oli luoda verkkosivut, jossa Estimote-laitteiden tietoja ei kuka tahansa pysty näkemään, vaan käyttäjältä vaaditaan sisäänkirjautumista verkkosivuille. Sisäänkirjautumisen jälkeen Estimote-laitteista kerätyt tiedot tulevat käyttäjälle näkyviin. Uloskirjautumisen jälkeen verkkosivuille ei pääse muuten kuin uudelleen sisäänkirjautumalla.

Mobiilisovelluksessa tavoitteena oli kerätä tietoja Estimote-laitteista, näyttää ne puhelinsovelluksessa ja lähettää laitteista kerättyjä tietoja eteenpäin palvelimella oleviin rajapintoihin, jotka sen jälkeen sijoittavat ne tietokantaan. Liikkumisen oli oltava luontevaa ja ulkonäön selkeä, jotta käyttäjä pysyy perillä, miten sovellusta käytetään.

Tietokantajärjestelmän tavoitteena oli luoda tietokanta, joka koskee verkkosivuille kirjautumista eli toisin sanoen sisältää käyttäjätunnukset ja salasanat, joilla pääsee katsomaan Estimote-laitteista kerättyjä tietoja selaimen kautta. Toinen vaatimus tietokantajärjestelmälle oli luoda rajapinnat Estimote-laitteille, jotta kaikista laitteista kerätyt tiedot pystytään sijoittamaan oikein omaan tietokantaan.

2 KÄYTETYT TEKNOLOGIAT

2.1 Estimote Beacon

Estimote Beacon on avaimenperän kokoinen laite, joka toimii CR2477-nappiparistolla. Laitteesta on olemassa kahta eri versiota: Estimote Location Beacon ja Estimote Proximity Beacon. Estimote Beacon -laitteen kantama on maksimissaan noin 70 metriä. (1.)

2.1.1 Estimote Proximity Beacon

Estimote Proximity Beacon on laite, joka toimii majakkana erilaisille mobiilisovelluksille. Se sisältää kiihtyvyyssanturin, lämpötilasensorin sekä 2,4 gigahertsin radion. Radio hyödyntää Bluetooth Smart -tekniikkaa, jonka ansiota laitteen elinikä yhdellä paristolla on yli kolme vuotta, jos käytössä on laitteen vakioasetukset. (Kuva 1.) (1.)



Kuva 1. Estimote Proximity Beacon -laitteen koko verrattuna avaimenperään

Estimote Proximity Beacon tukee kahta erilaista datapakettityyppiä: iBeacon ja Eddystone. IBeacon on Applen kehittämä Bluetooth 4.0 -viestintäprotokolla, kun

taas Eddystone on Googlen kehittämä Bluetooth 4.0-viestintäprotokolla. Vaikona Estimote Proximity Beacon käyttää iBeacon Bluetooth -protokollaa, mutta protokollan voi vaihtaa kirjautumalla Estimote Cloud -pilvipalveluun. Kumpaa tahansa protokollaa voidaan käyttää riippumatta siitä, tehdäänkö mobiilisovellus Android- tai iOS-laitteelle. (2.)

iBeacon-protokolla lähettää datapaketteja, jotka sisältävät iBeacon-protokollille ominaiset tunnistet (UUID, Major ja Minor). iBeacon-tunniste on aina 20 tavua pitkä (UUID = 16 tavua, Major = 2 tavua ja Minor = 2 tavua). Näiden tunnisteen yhdistämisellä voidaan viitata yksittäiseen Estimote Beaconiin tai halutessa voidaan myös tunnistaa Estimote Beacon -ryhmä, jossa on useampia Estimote Beacon -laitteita. (3.)

Eddystone sisältää Estimote Proximity Beacon-laitteelle kaksi erilaista alakategoria datapakettia: Eddystone-UID ja Eddystone-URL. Eddystone-UID on samankaltainen protokolla kuin iBeacon, mutta sen sijaan, että se jakautuisi kolmeen osaan, niin kuin iBeacon, se jakautuu vain kahteen osaan ja sen tunnisteen yhteispituus on aina 16 tavua. Eddystone-UID koostuu 10 tavun pituisesta nimiavaruudesta sekä 6 tavua pitkästä instanssista. Eddystone-URL on poikkeuksellinen protokolla iBeacon-protokollasta ja Eddystone-UID-protokollasta, sillä sen tarkoitus on lähettää Estimote Beacon-laitteesta pelkästään URL-tyyppinen nettiosoite, jonka pystyy avaamaan selaimella, joka tukee Physical Web -konseptia. Tämä protokolla ei vaadi mobiilisovellusta, vaan voi toimia pelkästään Physical Web -konseptin avulla. (2.)

2.1.2 Estimote Location Beacon

Estimote Location Beacon on uudemman sukupolven Estimote Beacon, joka verraten Proximity Beacon-laitteeseen voi mm. lähettää useamman datapaketin kerrallaan. Se myös sisältää sensoreita, joita ei ole Estimote Proximity Beacon -laitteissa, kuten ympäröivän valon sensori, magnetometri ja reaaliajassa toimiva kello. Estimote Location Beacon sisältää samat tekniset spesifikaatiot kuin edeltäjänsä, eli 2,4 gigahertsin radion, 32-bit ARM Cortex M0-prosessorin, taajuusalueen 2400 megahertsistä 2483,5 megahertsiin, 256 kilobitin flash-muistin

ja 4 desibelin lähtötehon. Estimote Location Beacon-laitteen pariston elinikä vakioasetuksilla on 7 vuotta eli huomattavasti pidempi kuin Estimote Proximity Beacon -laitteilla, vaikka kummatkin käyttävät samaa nappiparistoa. (4.)

Sen lisäksi, että Estimote Location Beacon voi lähettää useampia eri datapaketteja kerrallaan, saa se myös edeltäjäänsä verrattuna neljä uutta datapakettityyppiä käyttöönsä: Eddystone-TLM, Estimote Connectivity, Estimote Location ja Estimote Telemetry. Jokaisen protokollan lähetysaikaväli on määriteltävissä erikseen käyttäjän tarpeen mukaan. (5.)

Eddystone-TLM-datapaketti on suunniteltu käytettäväksi apupakettina Estimote Location Beacon-laitteiden iBeacon-protokollan tai Eddystone-UID ja/tai Eddystone-URL-datapakettien kanssa. Tämän datapaketin kanssa voidaan käyttäjää esimerkiksi varoittaa, jos laitteen paristo on loppumaisillaan tai jos Estimote Location Beacon-laitteen ympäröivä lämpötila nousee valitun kynnyksen yli. (2.)

Estimote Connectivity-protokolla on protokolla, jonka avulla voidaan luoda yhteys Estimote Location Beacon-laitteeseen. Tämä protokolla on aina aktiivinen laitteessa, eikä sitä voi sulkea. Estimote Connectivity-protokollan avulla voidaan laitteen sisällä oleva laiteohjelmisto päivittää. Estimote Location-protokolla on suunniteltu tukemaan Estimoten Indoor Location SDK-pakettia. Estimote Telemetry on protokolla, jolla voidaan lukea Estimote Location Beacon-laitteiden sensorien ja laitteen tietoja, esimerkiksi reaaliaikaisen kellon aikaa, lämpötilanturin arvoja ja pariston tietoja. (5.)

2.2 Estimote Nearable

Estimote Nearable, tai toisella nimellä Estimote Sticker, on samankaltainen laite kuin Estimote Beacon, mutta siinä on osittain eri ominaisuuksia Estimote Beacon-laitteeseen verrattuna. Sen elinikä yhdellä CR2020-nappiparistolla on noin yksi vuosi ja sen signaalin kantama on noin 15 metriä. Estimote Nearable-laite on yli puolet pienempi kuin Estimote Beacon-laitteet. (Kuva 2.) (6.)

Estimote Sticker sisältää lähes samat tekniset tiedot kuin Estimote Beacon-laitteet. Suurimpana erona on lähtötehon määrä, joka vaihtelee minimistä -12 dBm:stä 4 dBm:iin asti. Laite sisältää liiketunnistimen ja lämpötilanturin. (4.)

Estimote Nearable-laitteella on mahdollista käyttää kolmea eri protokollaa: Nearable, iBeacon ja Eddystone-URL. Nearable-protokolla sisältää enemmän Estimote Nearable-laitteeseen liittyvää dataa yhdessä paketissa. Tämä protokolla on suunniteltu varta vasten keräämään kaikkien sensorien ja ominaisuuksien tiedot Estimote Nearable laitteesta yhteen pakettiin. (7.) (8.)



KUVA 2. Estimote Nearable

2.3 Bluetooth Smart

Alkuperäisin Bluetooth Low Energy (BLE) lähti liikkeelle Nokian suunnittelemasta ja vuonna 2006 esittelemästä Wibree-tekniologiasta. Vuonna 2010 Bluetooth Special Interest Group (SIG) hyväksyi Wibreen yhdistämisen omaan Bluetooth-standardiin ja loi siitä nykyisen Bluetooth Low Energy-standardin. Tavoitteena oli kehittää siitä mahdollisimman energiaa säästävä, halpa ja yksinkertainen radiostandardi, jonka SIG lopulta saavutti. Tämän jälkeen Bluetooth Low Energy-tekniologian suosio kasvoi älypuhelimien, tablettien ja kannettavien tietokoneiden tekniologioiden mukana. Suurina vaikuttajina olivat puhelinvalmistajat, kuten Apple ja Samsung, jotka laajensivat Bluetooth Low Energy-tekniologian kehitysmahdollisuuksia kannettavissa laitteissaan. (9.)

Bluetooth Smart-tekniologiassa on nykyään monta nimeä: Bluetooth Smart, Bluetooth 4.0 ja Bluetooth Low Energy (BLE). Se on suunniteltu olemaan energiatehokkaampi, kuin edeltäjänsä perinteinen Bluetooth, ja toimimaan tilanteissa, joissa tiedon lähettäminen ei vaadi katkeamatonta yhteyttä. Esimerkiksi tilanteessa jossa laite mittaa lämpötilaa se varoittaa Bluetooth Smart-tekniologian avulla, jos lämpötila nousee tietyn asteikon yli. Tässä tapauksessa laitteen Bluetooth Smart voi olla unitilassa niin kauan, että esimerkin lämpötila nousee rajan yli. Tämän jälkeen Bluetooth Smart menee päälle ja lähettää datapaketin, jonka jälkeen voi se mennä takaisin lepotilaan. (10.)

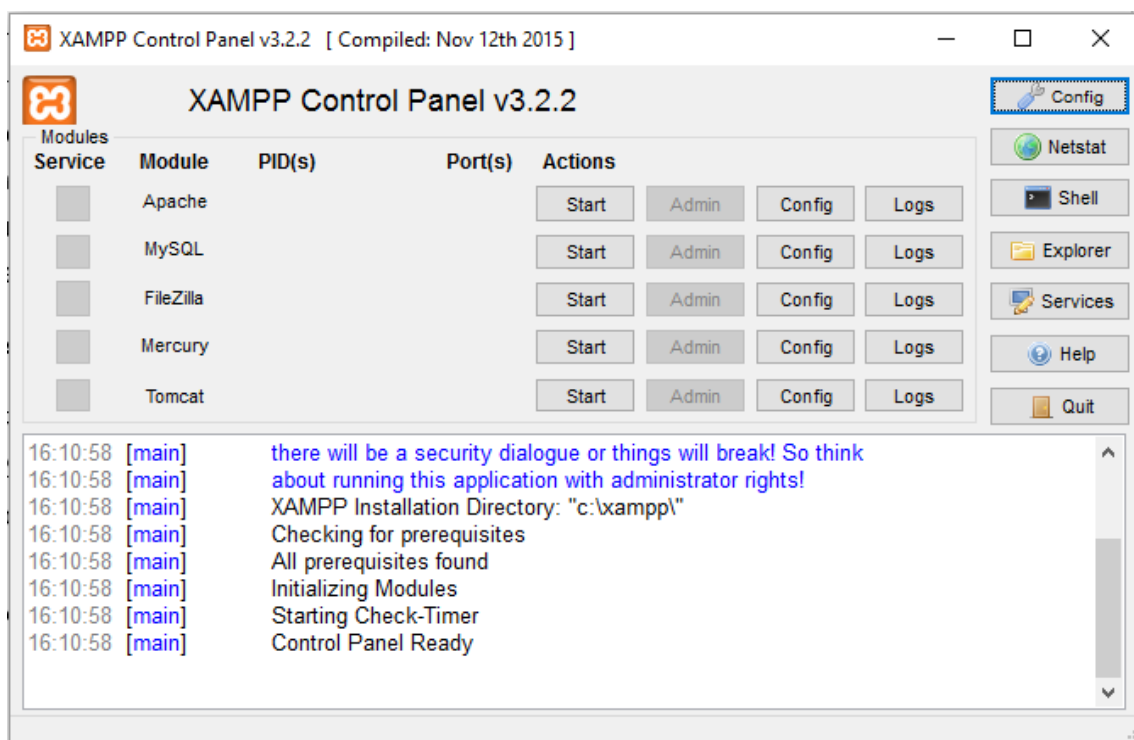
2.4 ARM Cortex M0 -prosessori

ARM Cortex M0 -prosessori on suunniteltu olemaan mahdollisimman pieni 16-32-bittinen prosessori. Tällä hetkellä se on ARM-yhtiön pienin tarjolla oleva prosessori ja sillä on laajat käyttömahdollisuudet alueilla, joissa tarvitaan pientä ja energiatehokasta prosessoria. Käyttömahdollisuuksiin kuuluu mm. älykellot, älyvalaistukset, robotiikka ja lääketieteelliset laitteet. (11.)

3 KÄYTETYT OHJELMISTOT

3.1 XAMPP

XAMPP on avoimen lähdekoodin verkkopalvelinohjelma, joka tulee sanoista Cross-Platform (X), Apache (A), MySQL (M), PHP (P) ja Perl (P). Sen on luonut Apache Friends -niminen hanke, jonka tarkoitus on edistää Apache-verkkopalvelimen käyttöä ja helpottaa kehittäjiä yhdistämään palvelimeensa tietokantoja ja käyttämään ohjelmointikieliä, kuten PHP:ta ja Perliä. XAMPP on kevyt ja yksinkertainen Apache-palvelin, jolla kehittäjät voivat luoda helposti paikallisen HTTP-palvelimen. Tämä ohjelma mahdollistaa nettisivun paikallisen ylläpidon localhost-osoitteen avulla ja tämän avulla voidaan omasta tietokoneesta luoda palvelin, jota pääsee katsomaan muista tietokoneista localhost-tietokoneen IP-osoitteella. (Kuva 3.) (12.)



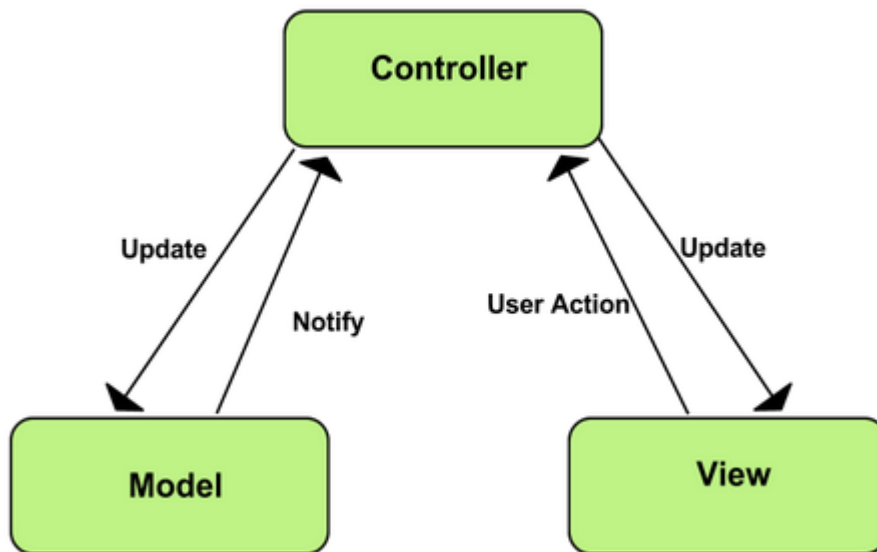
KUVA 3. XAMPP-ohjauspaneeli

XAMPP mahdollistaa seuraavien ohjelmistojen yhdistämisen: Apache, MySQL, PHP + PEAR, MiniPerl, OpenSSL, phpMyAdmin, XAMPP Control Panel, Webalizer, Mercury Mail Transport System, FileZilla FTP Server, SQLite, ADODB, Send Optimizer, XAMPP Security ja Tomcat. Näiden lisäksi XAMPP-ohjelmaan on saatavilla lisäosia, kuten esimerkiksi WordPress, Joomla! ja CMS Made Simple. Näiden lisäosien ja laiteohjelmistojen avulla kehittäjä saa laajat mahdollisuudet XAMPP-ympäristönsä kehittämiseen, muokkaukseen ja optimoimiseen. (13.)

3.2 CodeIgniter

CodeIgniter lähti liikkeelle vuonna 2006 Rick Ellisen luomana viitekehystenä, joka sai alkunsa ExpressionEnginestä eli kaupallisesta verkkoalustasta. CodeIgniter on kevyt, nopea ja ilmainen viitekehystyökalu, jonka tarkoitus on säästää aikaa ja vaivaa verkkosovellusten kehittämisessä. Se perustuu avoimeen lähdekoodiin ja siitä löytyy valtavat määrät dokumentaatiota sekä CodeIgniterin kehittäjiltä että CodeIgniterin käyttäjiltä. (14;15.)

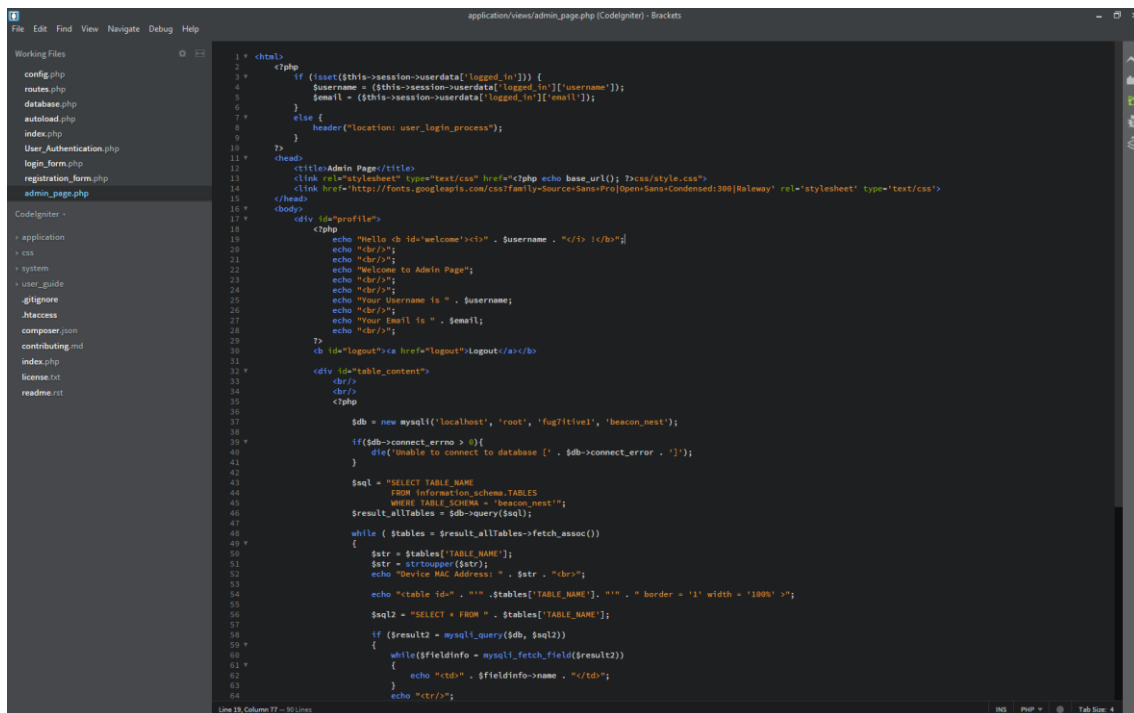
CodeIgniter käyttää Model-View-Controller (MVC) -logiikkaa, joka tarkoittaa sitä, että viitekehys jakautuu kolmeen osaan: malli, näkymä ja käsittelijä. Malli on objekti, joka hoitaa tiedon tallentamisen, ylläpidon ja käsittelyn. Näkymä on käyttäjälle näkyvä käyttöliittymä. Käsittelijä on osio, joka hoitaa käyttäjän antamat käskyt ja muokkaa mallia ja näkymää näiden mukaan. Näitä kolmea osiota voidaan kehittää hyvin pitkälle itsenäisesti, mikä helpottaa työympäristöissä, jossa kehittäjinä on ohjelmoijia ja graafisia suunnittelijoita. (Kuva 4.) (16.)



KUVA 4. MVC-arkkitehtuuri

3.3 Brackets

Brackets on kesällä vuonna 2012 julkaistu avoimen lähdekoodin tekstieditori, jonka omistaa yhtiö nimeltä Adobe. Brackets-tekstieditorin pääasiallinen painopiste on verkkokehityksessä. Se on luotu käyttäen HTML-, JavaScript- ja CSS-ohjelmointikieliä ja siinä on ominaisuuksia, joita kaikista ilmaisista tekstieditoreista ei löydy. Ominaisuuksiin kuuluu mm. suora esikatselumahdollisuus, HTML-elementtien nopea muokkaus ilman, että tarvitsee mennä CSS-tyylitiedostoon, ja laajat valikoimat yhteensopivia laajennuksia. Brackets tukee yli 38:aa tiedostotyyppiä, joista suuri osa on eri ohjelmointikieliä. Näihin kieliin kuuluvat mm. HTML, CSS, JavaScript, PHP ja C++. Brackets saa jatkuvasti uusia päivityksiä ja laajennuksia, jotka tuovat uusia ominaisuuksia ja mahdollisuuksia tekstieditorin käyttöön. (Kuva 5.) (17.)

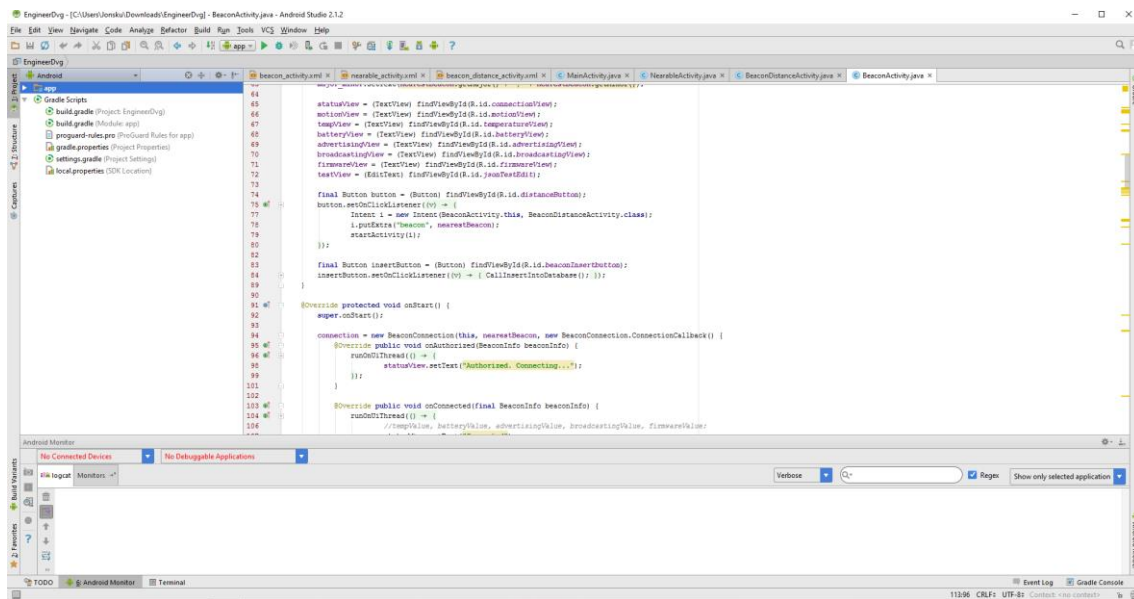


```
1 <html>
2 <?php
3 if (isset($_SESSION->userdata['logged_in'])) {
4     $username = ($_SESSION->userdata['logged_in']['username']);
5     $email = ($_SESSION->userdata['logged_in']['email']);
6     }
7 else {
8     header("location: user_login_process");
9     }
10 }
11 <?php
12 <title>Admin Page</title>
13 <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
14 <link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro|Open+Sans|Condensed:300|Raleway" rel="stylesheet" type="text/css">
15 </head>
16 <body>
17 <div id="profile">
18 <?php
19     echo "Hello <span id='welcome'><?php echo $username . "</span>";
20     echo "<br/>";
21     echo "<br/>";
22     echo "Welcome to Admin Page";
23     echo "<br/>";
24     echo "Your Username is " . $username;
25     echo "<br/>";
26     echo "Your Email is " . $email;
27     echo "<br/>";
28     echo "<br/>";
29     ?>
30 <a href="#">Logout</a>
31 </div>
32 <div id="table_content">
33 <br/>
34 <br/>
35 <?php
36
37 $db = new mysqli('localhost', 'root', 'fug7itvel', 'beacon_mest');
38
39 if ($db->connect_errno > 0) {
40     die("Unable to connect to database [" . $db->connect_error . "].");
41 }
42
43 $sql = "SELECT TABLE_NAME
44 FROM information_schema.TABLES
45 WHERE TABLE_SCHEMA = 'beacon_mest'";
46 $result_allTables = $db->query($sql);
47
48 while ( $tables = $result_allTables->fetch_assoc() )
49 {
50     $str = $tables['TABLE_NAME'];
51     $str = strtoupper($str);
52     echo "Device MAC Address : " . $str . "<br/>";
53
54     echo "<table id=' " . $tables['TABLE_NAME'] . "' border = '1' width = '100%'>";
55
56     $sql2 = "SELECT * FROM " . $tables['TABLE_NAME'];
57
58     if ($result2 = mysqli_query($db, $sql2))
59     {
60         while($fieldInfo = mysqli_fetch_field($result2))
61         {
62             echo "<tbody> . $fieldInfo->name . "</tbody>";
63         }
64     }
65 }
```

KUVA 5. Brackets-tekstieditori

3.4 Android Studio

Android Studio on virallinen ohjelmointiympäristö Androidille, joka on saanut alkunsa IntelliJ IDEA -ohjelmointiympäristöstä. IntelliJ IDEA on kehitetty Javan ohjelmointiympäristölle. Android Studio tarjoaa Android-kehittäjille kattavan määrän ominaisuuksia, jotka nopeuttavat ja helpottavat Android-sovellusten kehittämistä. Näihin ominaisuuksiin kuuluu mm. laaja valikoima erilaisia emulaattoreita, joilla pystytään testaamaan kehitettyjä Android-sovelluksia, mahdollisuuden kehittää ohjelmistoja kaikille Android-laitteille ja mittavat valikoimat testaukseen käytettäviä työkaluja ja viitekehyksiä. (Kuva 6.) (18.)



KUVA 6. Android Studion työikkuna.

3.5 PHP

PHP on lyhenne, joka tulee rekursiivisesti sanoista Hypertext Preprocessor. Se on avoimen lähdekoodin ohjelmointikieli, joka on tarkoitettu pääasiallisesti serverityyppiseen ohjelmointiin. PHP-kieli keksittiin vuonna 1994 ja sen alkuperäinen luoja oli nimeltä Rasmus Lerdorf. Sen alkuperäinen tarkoitus oli jäljittää verkkosivun kävijämääriä, mutta pian alun jälkeen se alkoi saamaan lisää toiminnallisuuksia, kuten esimerkiksi sillä pystyttiin kommunikoimaan tietokantojen kanssa ja luomaan viitekehyksiä verkkosivuille. Tästä kehittyi lopulta, monien mutkien jälkeen, nykyinen PHP-kieli, joka on laajassa käytössä verkkosivujen kehityksessä. (19.)

PHP-kieltä käytetään paljon verkkosivujen luonnissa, koska sen avulla pystytään kommunikoimaan esimerkiksi tietokantojen kanssa sekä luomaan dynaamisia verkkosivuja. Se toimii kaikissa suurimmissa käyttöjärjestelmissä, kuten Windowsissa, iOS:ssä ja Linuxissa. PHP tukee useimpia verkkopalvelimia, kuten mm. Apache ja IIS. Näiden lisäksi PHP tukee myös hyvin laajaa valikoidaan erilaisia tietokantoja, kuten MySQL, DB++ ja SQLite. (20.)

3.6 HTML ja CSS

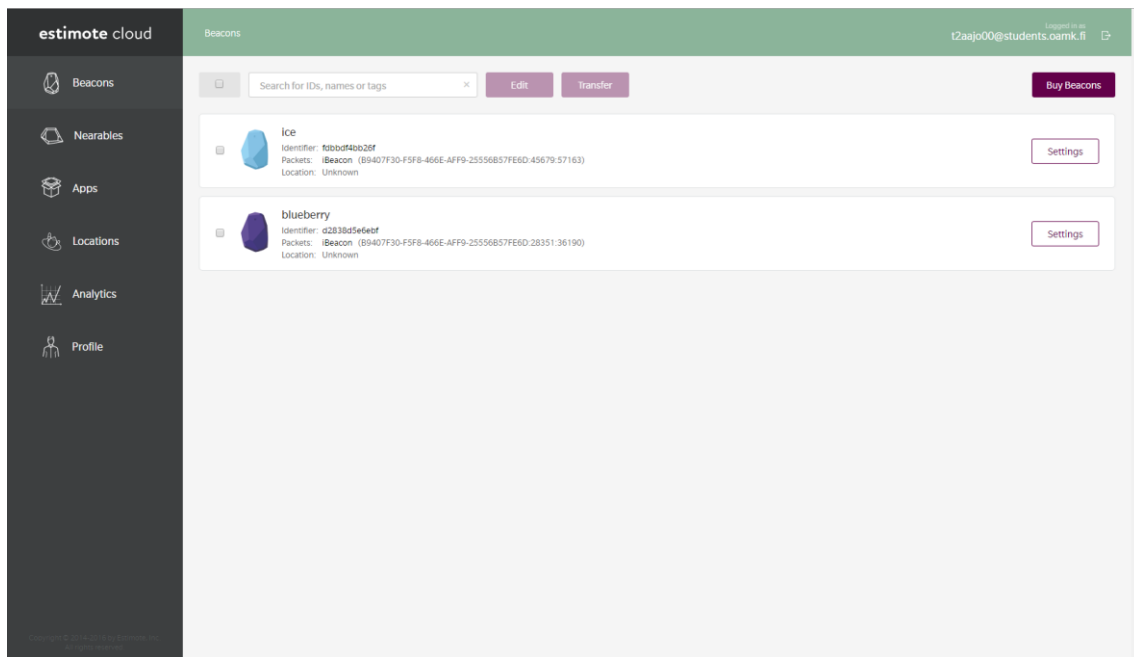
HTML tulee sanoista Hypertext Markup Language. Se on verkkosivujen virallinen ohjelmointikieli, joka sai alkunsa Euroopan hiukkasfysiikan tutkimuskeskuksessa CERNissä vuonna 1990 eli vuoden internetin keksimisen jälkeen. Sen alkuperäinen tarkoitus oli helpottaa teknisten dokumenttien jakoa, mutta pian HTML-kielen keksimisen jälkeen huomattiin, että sitä voidaan käyttää myös muihinkin tarkoituksiin kuin pelkkien dokumenttien jakamiseen. Internetin kehityksen myötä myös HTML-kielen käyttömahdollisuudet kasvoivat ja sitä lähdettiin hyödyntämään erilaisten nettisivujen luonnissa. (21.)

CSS, eli Cascading Style Sheets, on tyylikieli, jolla pystytään määrittelemään HTML-kielen ulkonäöllisiä elementtejä erillisellä tiedostolla. Se sai alkunsa vuonna 1996, kun W3C päätti, että verkkosivun sisältö ja tyyli pitäisi pitää erillään toisistaan. CSS-tyylikieli sisältää paljon hyviä ominaisuuksia. Sen avulla pystytään esimerkiksi muuttamaan sivuston kaikkia elementtejä yhdeltä CSS-sivulta, sen avulla säästää aikaa ja se selkeyttää verkkosivun koodin kirjoittamista. (22.)

3.7 Estimote Cloud

Estimote Cloud on verkkosivupohjainen alusta, jonka avulla pystytään hallinnoimaan Estimote Beacon- ja Estimote Nearable -laitteita, Estimotelle luotuja sovelluksia, Estimote Indoor Location SDK:n sijainteja ja omaa Estimote-profiilia. Se on luotu helppokäyttöiseksi ja monipuoliseksi verkkosivuksi, jota on helppo oppia käyttämään. Estimote Cloud on linkitetty Estimoten omaan mobiilisovellukseen, jonka avulla Estimote Cloud-pilvipalvelussa luodut muutokset ovat päivitettävissä laitteeseen kirjautumalla mobiilisovellukseen ja viemällä puhelin tarpeeksi lähelle päivitettävää laitetta. (Kuva 7.) (23.)

Estimote Cloud mahdollistaa Estimote Beacon -laitteiden nimen, tunnistimien ja lähtötehon muuttamisen. Se myös auttaa Estimote Beacon ja Estimote Nearable -laitteiden omistuksien siirtämisessä toiselle tilille sekä niiden jakamisessa toisten sovellusten kanssa. Näiden lisäksi Estimote Cloudissa näkyy kaikkien Estimote Beacon ja Estimote Nearable -laitteiden tiedot, kuten maksimietäisyydet, paristojen odotetut eliniät ja laitteiden versionumerot. (23.)



KUVA 7. Estimote Cloud -pilvipalvelun etusivu sisäänkirjautumisen jälkeen.

4 TYÖN TAVOITE JA AIKATAULU

Tässä luvussa käsitellään työn tavoitetta ja käydään läpi vaihe vaiheelta työn aikataulua. Työn tilaajana toimi Oulun ammattikorkeakoulu, jonka kautta työ sai myös vaatimusmäärittelyt.

4.1 Työn tavoite

Työn tavoitteena oli luoda kokonaisuus, joka koostuisi kolmesta osasta: mobiili-sovellus, verkkosivut ja tietokanta. Kaikkien kolmen osan täytyy pystyä kommunikoimaan keskenään ja toimimaan yhtenä kokonaisuutena.

Mobiilisovelluksessa vaatimuksena oli, että se pystyy listaamaan lähistöllä olevat Estimote-laitteet niiden läheisyyden perusteella ja kommunikoimaan Estimote Beacon- ja Estimote Nearable -laitteiden kanssa sekä keräämään näistä Estimote-laitteiden tietoja ja sensoridatoja kuten lämpötilaa, läheisyystietoja ja laitteen nappipariston elinikää ja näyttämään kerätyt tiedot mobiilisovelluksen käyttöliittymässä. Lisäksi sovelluksen täytyi pystyä lähettämään kerätyt tiedot palvelimelle, jossa ne sitten sijoitettaisiin tietokantaan.

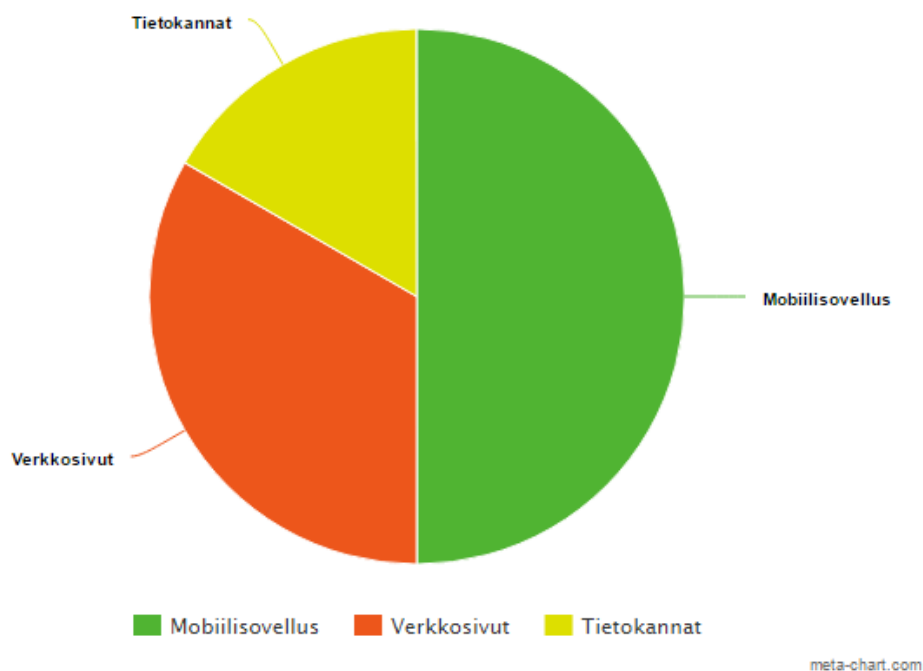
Verkkosivujen tavoitteena oli luoda sisäänkirjautumissivu, jonka ohi pääsee pelkästään syöttämällä tietokannassa olevat käyttäjätunnuksen ja salasanan. Ilman näitä tietoja käyttäjällä ei pitä olla mahdollisuuksia päästä käsiksi salasanasuojattuun verkkosivuun, jossa tarkoitus on näyttää tietokannan toisen taulukon tuloksia eli tuloksia, joita mobiilisovellus on lähettänyt palvelimen rajapinnan kautta tietokantaan. Verkkosivujen sisäänkirjautumissuojauksen vaatimuksena oli, että sitä ei pysty ohittamaan esimerkiksi kopioimalla sisäänkirjautumissivun jälkeen tulevaa käyttäjäisivun selainosoitetta tai kirjautumalla ulos ja painamalla backspace-näppäintä, vaan näissä esimerkkien tapauksissa käyttäjä ohjattaisiin sisäänkirjautumissivulle.

Tietokannassa vaatimuksena oli luoda kaksi erillistä tietokantaa, käyttäjätunnus- ja salasananatietokanta sekä Estimote-laitteiden tietojenkeruutietokanta. Näiden lisäksi tietojenkeruutietokanta tarvitsi kolme erillistä rajapintaa palvelimeen, joiden avulla tietokantaan pystytään syöttämään kolmea erilaista taulukkotyyppiä.

Nämä kolme taulukkotyyppiä perustuvat siihen, että Estimote Nearable-laite lähettää jonkin verran erilaista tietoa verrattuna Estimote Beacon-laitteen laitetieto-osuuteen tai Estimote Beacon-laitteen läheisyystieto-osuuteen. Estimote Beacon-laitteen läheisyystieto-osuutta ei voida yhdistää Estimote Beacon-laitteen laitetieto-osuuteen, koska käytössä oli Estimote Proximity Beacon, joka pystyy lähettämään pelkästään yhtä datapakettia kerrallaan ja tässä laitteessa nämä kaksi osuutta ovat erillisissä paketeissa.

4.2 Työn aikataulu

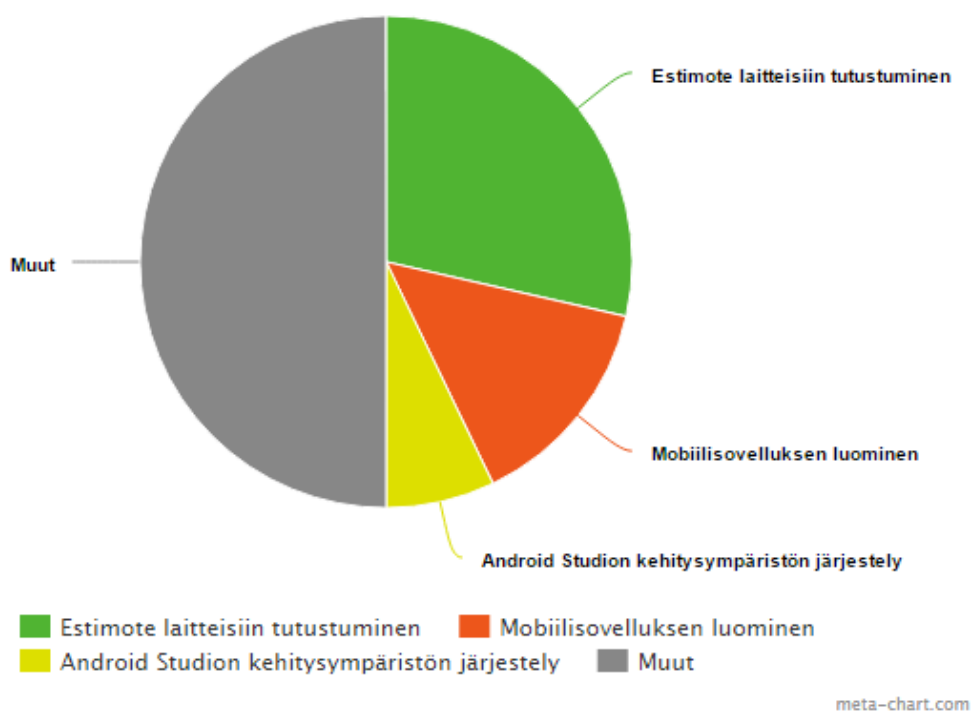
Aikataulullisesti työ jakautui kolmeen vaiheeseen: mobiilisovellus, verkkosivut ja tietokannat. Näistä mobiilisovellus vei noin puolet kokonaisajasta, verkkosivut kolmasosan kokonaisajasta ja tietokannat kuudeosan kokonaisajasta. Kokonaisaika työn suorittamiseen oli noin kaksi kuukautta, joka suoritettiin tahdilla 25 h/viikko. (Kuva 8.)



KUVA 8. Kokonaisaikataulu.

Mobiilisovelluksen rakentamisessa vaiheina toimi Estimote Beacon- ja Estimote Nearable -laitteisiin tutustumisen, Android Studion kehitysympäristön rakentami-

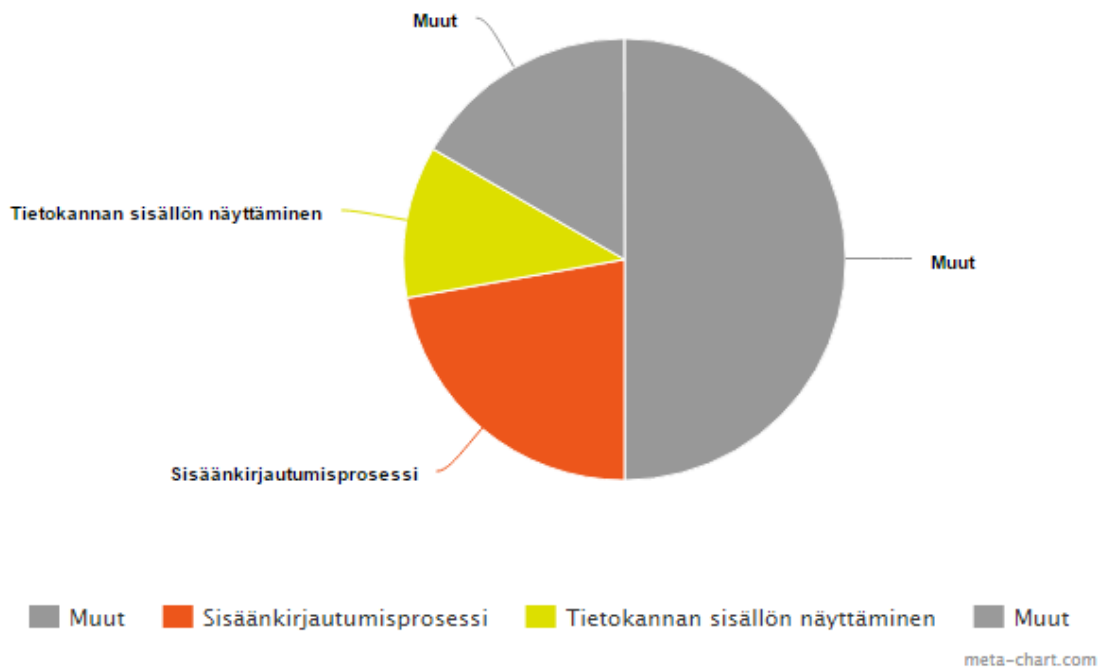
nen ja testaaminen ja itse mobiilisovelluksen luominen. Näistä ylivoimaisesti aikaa vievin vaihe oli Estimote Beacon- ja Estimote Nearable -laitteisiin tutustuminen ja testaaminen, koska Estimote-laitteet eivät olleet ennestään tuttuja. Estimote-laitteisiin tutustuminen vei arviolta noin neljä seitsemäsosaa mobiilisovelluksen rakentamiseen varatusta ajasta, mobiilisovelluksen luominen vei noin kaksi seitsemäsosaa mobiilisovelluksen rakentamiseen varatusta ajasta ja viimeinen yksi seitsemäsosaa mobiilisovellukseen varatusta ajasta meni Android Studion kehitysympäristön järjestelyihin. (Kuva 9.)



KUVA 9. Mobiilisovelluksen aikataulullinen jaottelu.

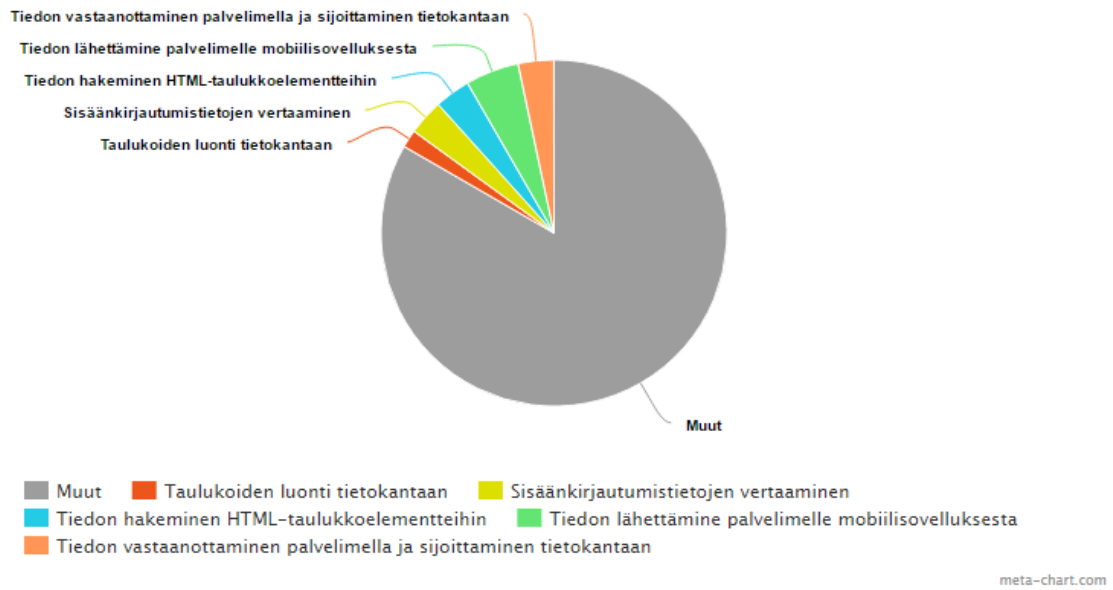
Verkkosivujen luonti jakautui kahteen aikataululliseen alakategoriaan: sisäänkirjautumisprosessi ja tietokannan sisällön näyttäminen. Näistä sisäänkirjautumisprosessi vei suurimman osan, eli noin kaksi kolmasosaa, verkkosivuille varatusta ajasta ja tietokannan sisällön näyttäminen taulukkomuodossa vei loput kolmasosan verkkosivuille varatusta ajasta. Sisäänkirjautumisprosessissa aikaa vievin osuus oli sisäänkirjautumisen varmistaminen, jotta käyttäjä ei pysty millään keinoilla ohittamaan sisäänkirjautumista, kuten esimerkiksi backspace-näppäintä painamalla. Tietokannan sisällön näyttämisessä aika kului pääasiassa

HTML-taulukkoelementtien järjestelmälliseen luomiseen, sen mukaan montako tietokannan taulukkoa tietokannassa on. (Kuva 10.)



KUVA 10. Verkkosivujen aikataulullinen jaottelu.

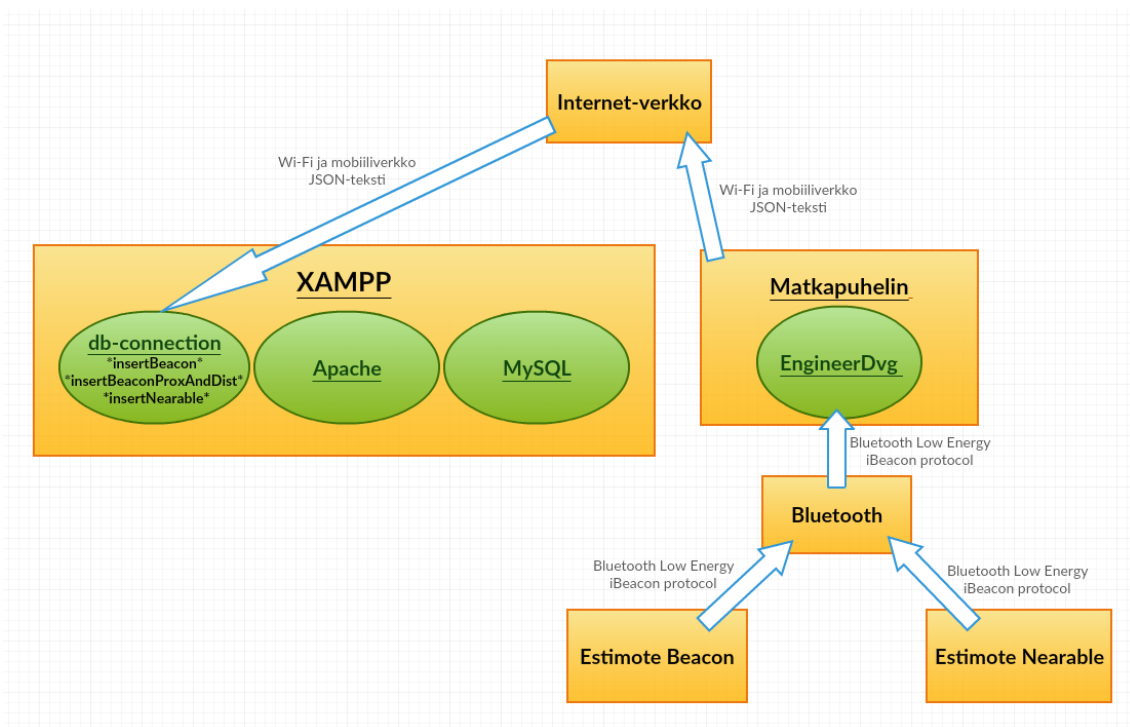
Tietokantojen luonti jakautui viiteen alakategoriaan: tietokannan taulukoiden luonti, johon kului yksi kymmenesosa varatusta ajasta, sisäänkirjautumisessa käytettävien tunnusten vertaaminen tietokannassa oleviin sisäänkirjautumistietoihin, johon kului yksi viidesosa varatusta ajasta, tietokannasta tiedon hakeminen HTML-taulukkoelementteihin (sisäänkirjautumisen jälkeen), johon kului yksi viidesosa varatusta ajasta, mobiilisovelluksessa tiedon lähettäminen palvelimelle, johon kului kolme kymmenesosa varatusta ajasta ja palvelimella tiedon vastaanottaminen ja sijoittaminen tietokantaan, johon kului yksi viidesosa varatusta ajasta. (Kuva 11.)



KUVA 11. Tietokantojen aikataulullinen jaottelu.

5 TYÖN KÄYTTÖYMPÄRISTÖN ASENNUS JA KÄYTTÖÖNOTTO

Ohjelma rakentuu XAMPP-palvelinohjelmasta, joka sisältää rajapinnat insertBeacon, insertBeaconProxAndDist ja insertNearable. Lisäksi XAMPP-ohjelmassa on Apache, eli varsinainen palvelin, ja MySQL-tietokanta, jossa on kaksi taulukkoa: user_login ja beacon_nest. Näiden lisäksi työssä käytetään matkapuhelinta, joka sisältää työn Android-ohjelman, eli EngineerDvg, ja kommunikointiin käytettävän Bluetoothin. Matkapuhelimen ja XAMPP-ohjelman väliseen kommunikointiin käytetään Internet-verkkoa, joka voi olla esimerkiksi Wi-Fi tai mobiiliverkko. Tätä kommunikointimenetelmää käytetään JSON-tekstin lähettämisessä matkapuhelimesta Apache-palvelimella oleviin rajapintoihin. Estimote Beacon- ja Estimote Nearable -laitteet käyttävät Bluetoothin Low Energy -menetelmää tietojen lähettämisessä laitteista matkapuhelimiin, jossa protokollana toimii iBeacon. (Kuva 12.)



KUVA 12. Komponenttikaavio ohjelmakokonaisuuden rakenteesta.

5.1 Android Studion asennus ja projektin luominen

Android Studion asentaminen alkoi lataamalla viimeisin ohjelmaversio Android Studiosta Androidin kotisivulta Developers-osiosta

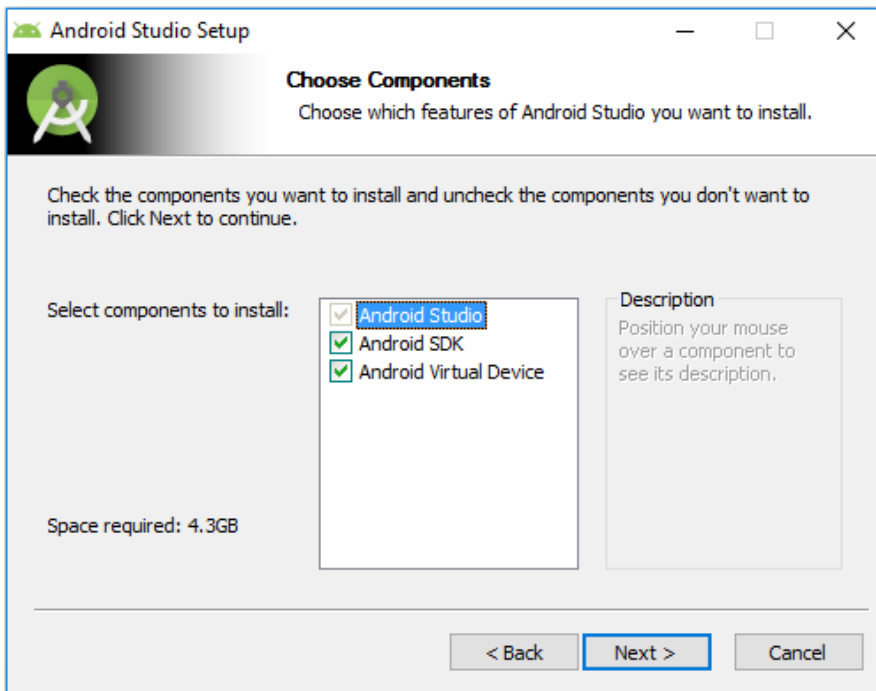
<https://developer.android.com/studio/index.html>.

Työssä käytetty ohjelmaversio numero oli Android Studio 2.1.2, jonka lisäksi Android Studio vaati, että työkoneella on asennettuna myös Java SE Development Kit 8, jonka sai ladattua osoitteesta

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.

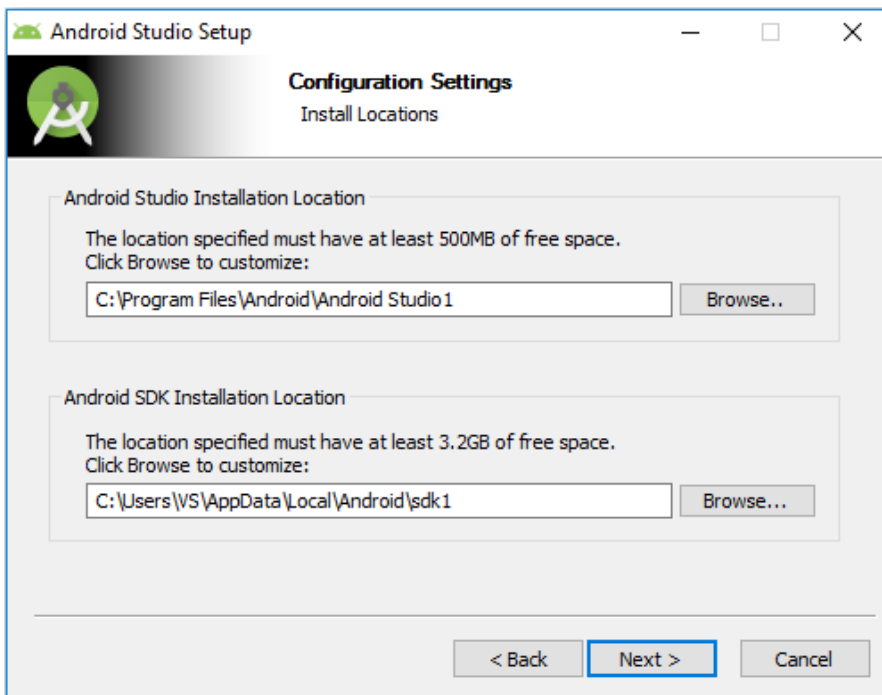
Java SE Development Kit 8, eli Java Standard Edition Development Kit, on Java-ohjelmointikielelle suunniteltu kehitysympäristö, joka auttaa ohjelmistokehittäjiä rakentamaan ja testaamaan Java-alustalla toimivia ohjelmia (24).

Sen jälkeen, kun Android Studio ja Java SE Development Kit 8 oli saatu ladattua, alkoi näiden ohjelmien asennus. Android Studion asennuksessa Android Virtual Devicen asentaminen ei ollut pakollista, koska ohjelmaa rakentaessa oli helpompaa testata ohjelmaa suoraan Android-puhelimella. (Kuva 13.)



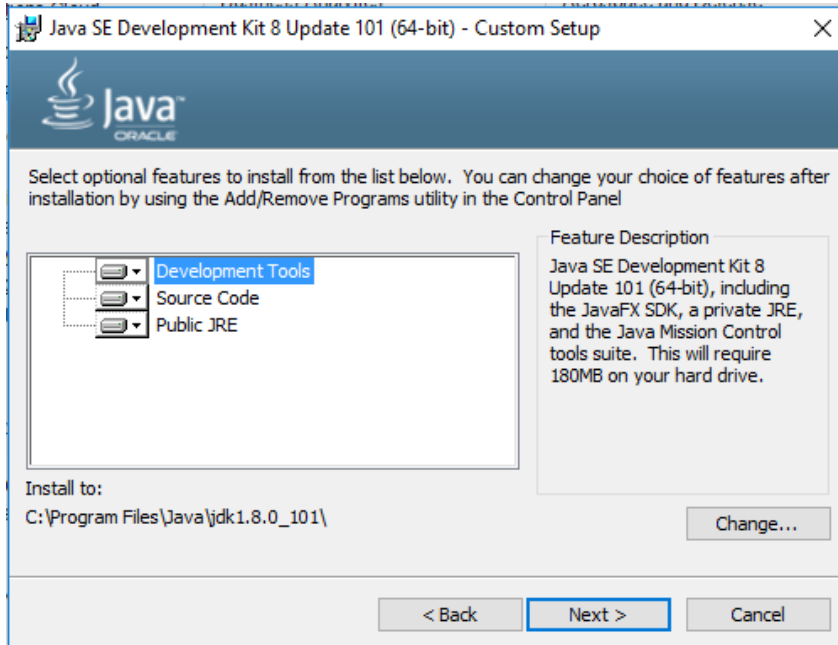
KUVA 13. Android Studion asennusvalikko

Android Studion asentamisessa asennuspaikan valitsemisella ei ollut muuta merkitystä kuin se, että tilaa löytyi tarpeeksi (kuva 14).

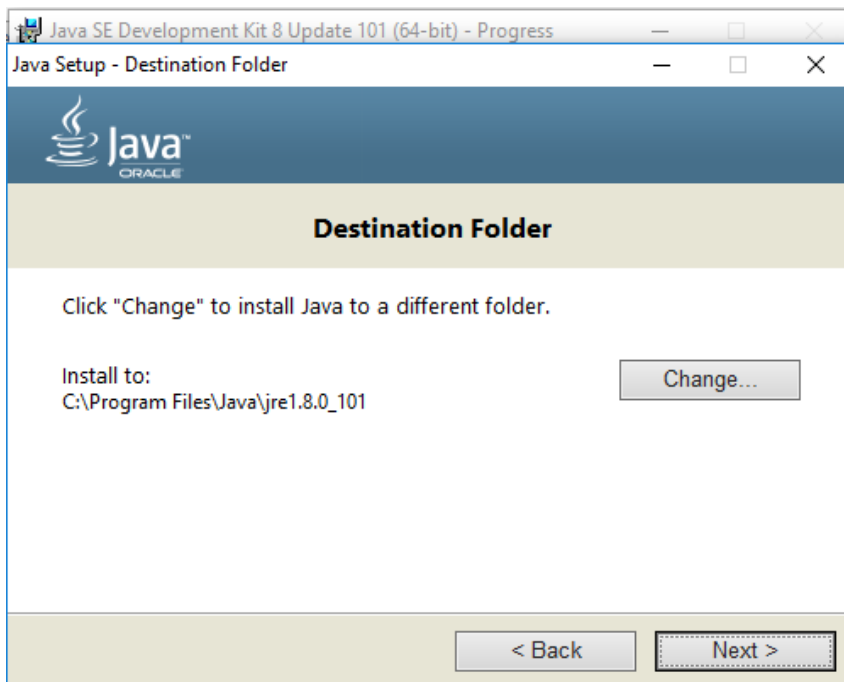


KUVA 14. Android Studion asennuspaikka.

Java SE Development Kit 8:n asennuksessa ei tarvinnut muuttaa asetuksia tai polkuja, vaan työkalun pystyi asentamaan vakioasetuksilla toimivuuden takaamiseksi (kuvat 15 ja 16).

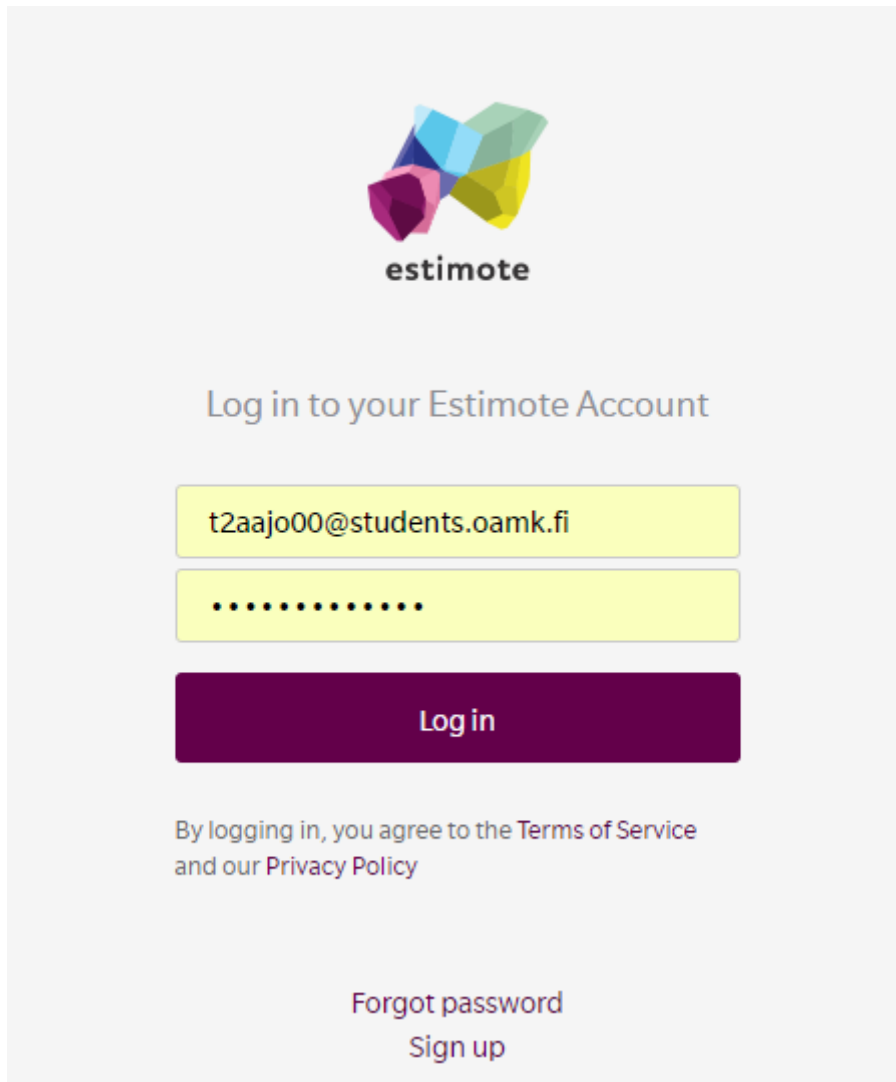


KUVA 15. Java SE Development Kit 8:n komponenttien asennus.



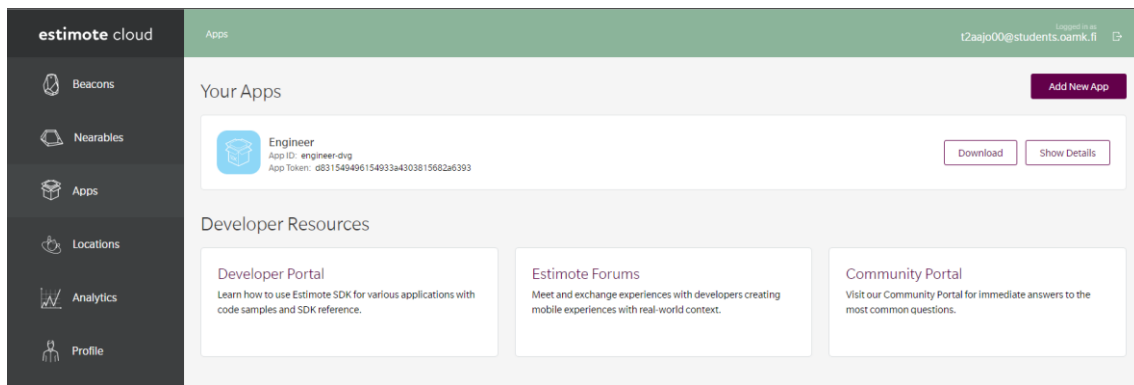
KUVA 16. Java SE Development Kit 8:n asennuskansio.

Tämän jälkeen, kun Android Studio ja Java SE Development Kit 8 oli saatu la-
dattua ja asennettua, alkoi itse Android-ohjelman luominen. Koska ohjelmalla
haluttiin kommunikoida Estimote Beacon- ja Estimote Nearable -laitteiden
kanssa, täytyi ohjelma luoda Estimote Cloud -pilvipalvelun kautta. Tätä varten
täytyi luoda käyttäjätunnukset osoitteessa <https://cloud.estimote.com/#/login>.
(Kuva 17.)



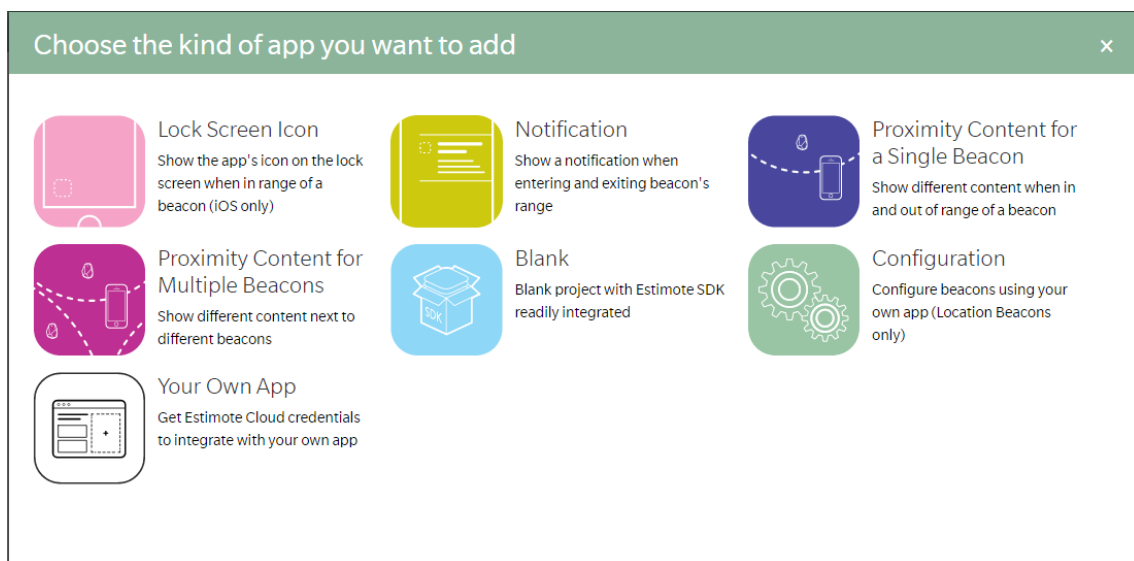
KUVA 17. Estimote Cloudin sisäänkirjautumisikkuna.

Sisäänkirjautumisen jälkeen ohjelman luominen onnistui menemällä Apps-väli-
lehteen, jossa oikeassa yläkulmassa on painike Add New App (kuva 18).



KUVA 18. Estimote Cloud Apps -välilehti.

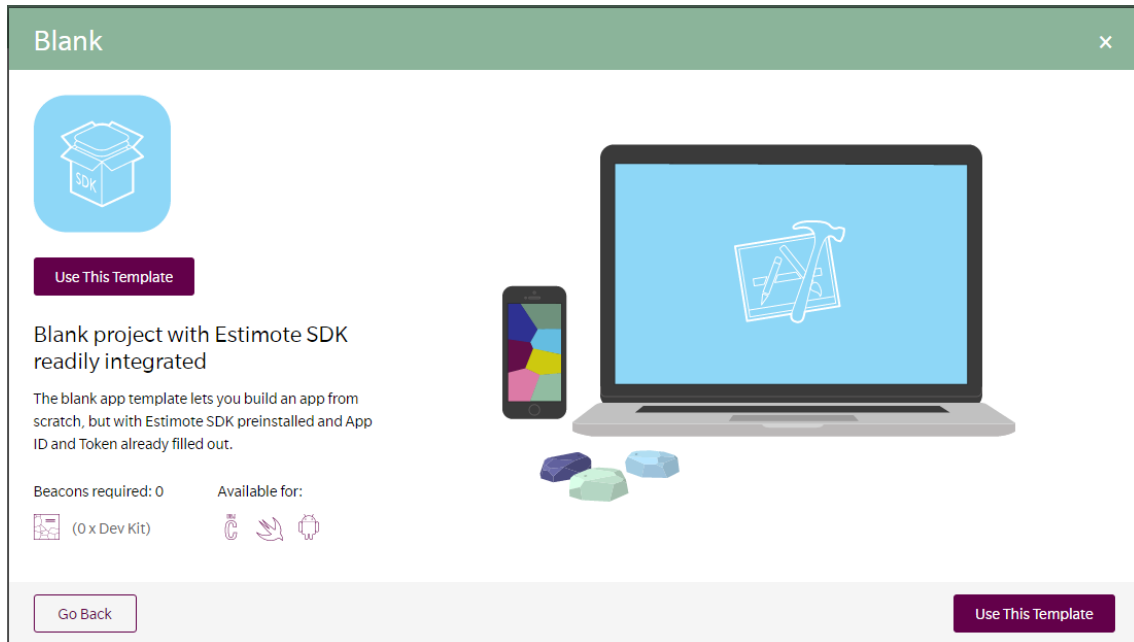
Estimote tarjoaa pilvipalvelussaan helpon tavan luoda Android-ohjelmien pohjia erilaisilla vakioasetuksilla, kuten ohjelmapohjan, jossa on valmiina koodit ilmoituksen lähettämiseen, jos puhelin on lähellä Estimote-laitteita, tai ohjelmapohjan, jossa näytetään tietyn Estimote Beacon-laitteen läheisyystiedot puhelimessa. (Kuva 19.)



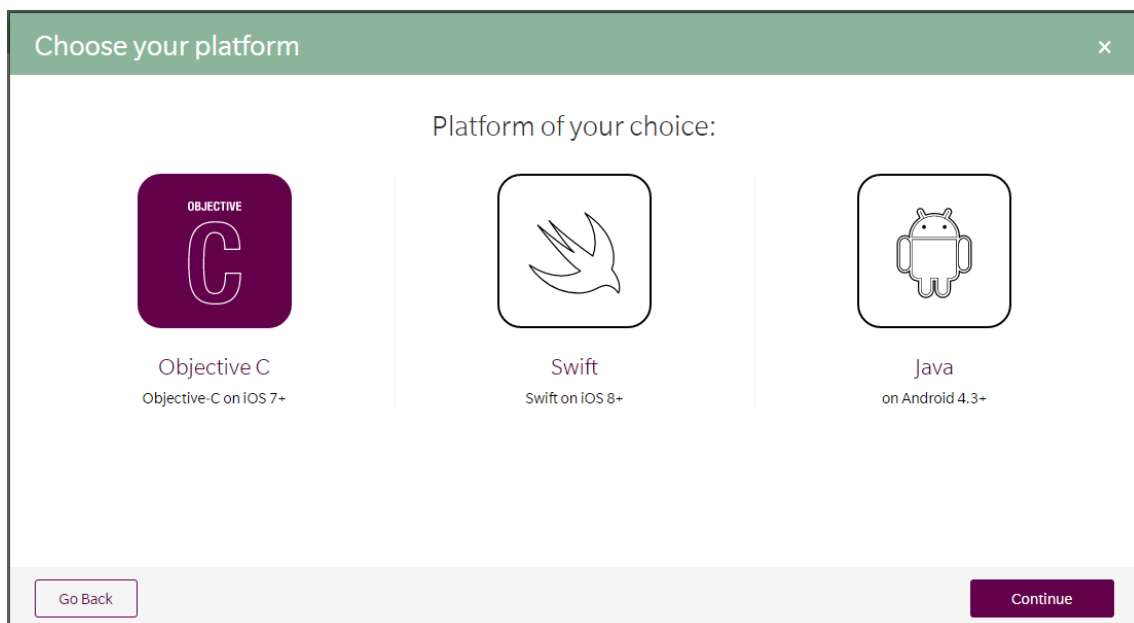
KUVA 19. Estimote Cloudissa olevat valmiit ohjelmapohjat.

Android-ohjelmalle valittiin Blank eli tyhjä projekti, jotta ohjelmasta saatiin luoda halutun näköinen. Ohjelmapohjan voi luoda kolmelle alustalle Estimote Cloudissa: Android-puhelimelle (minimi Android-versiovaatimus on 4.3), Objec-

tive C:lle (minimi iOS versiovaatimus on 7) ja Swiftille (minimi iOS versiovaatimus on 8). Koska ohjelma oli tarkoitus tulla Android-puhelimelle, niin valittiin alustaksi Javan eli Androidille suunnatun pohjan (kuvat 20 ja 21).

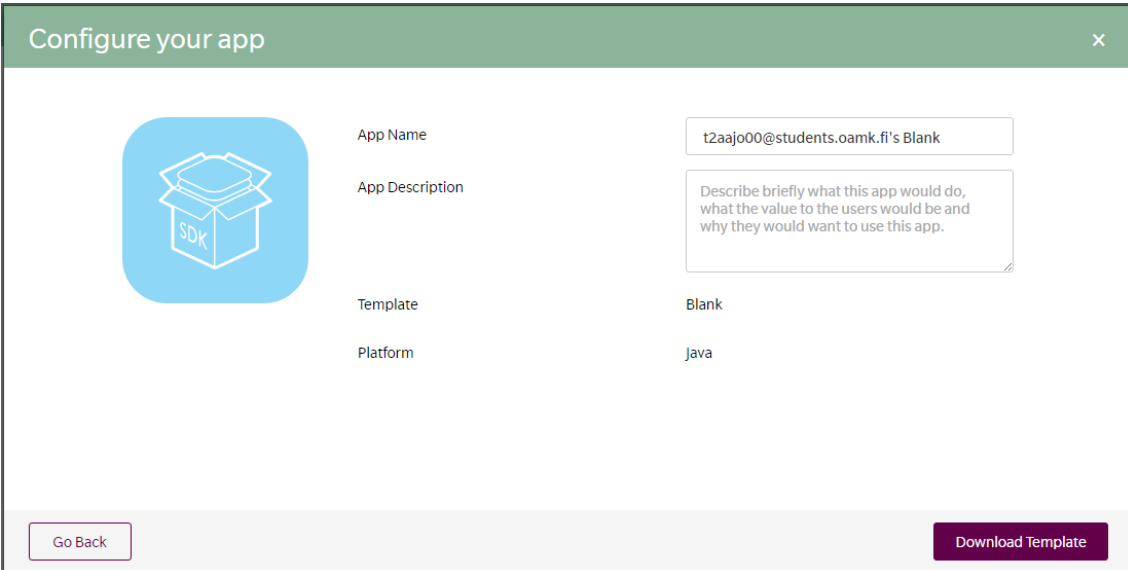


KUVA 20. Blank Android -malli.



KUVA 21. Estimote Cloudissa luodun ohjelman alusta.

Tämän jälkeen ohjelmalle sai valita nimen ja kirjoittaa vapaan kuvauksen tulevasta ohjelmasta. Tämän jälkeen pohjan pystyi lataamaan Estimote Cloud Apps -valikosta, jossa samalla näkee ohjelman App ID:n ja App Tokenin, jotka tarvitaan yhteyden luomiseen Estimote Beacon -laitteeseen. Estimote Nearable -laite ei tarvitse näitä tunnuksia, koska ne on tarkoitettu pelkästään Estimote Beacon -laitteita varten. Ohjelman lataamisen ja purkamisen jälkeen voidaan se suoraan avata Android Studiolla valitsemalla Android Studiossa File – New – Import Project, jonka jälkeen valitaan purettu ohjelmapohjan kansio ja lopuksi painetaan OK-painiketta. (Kuva 22.)

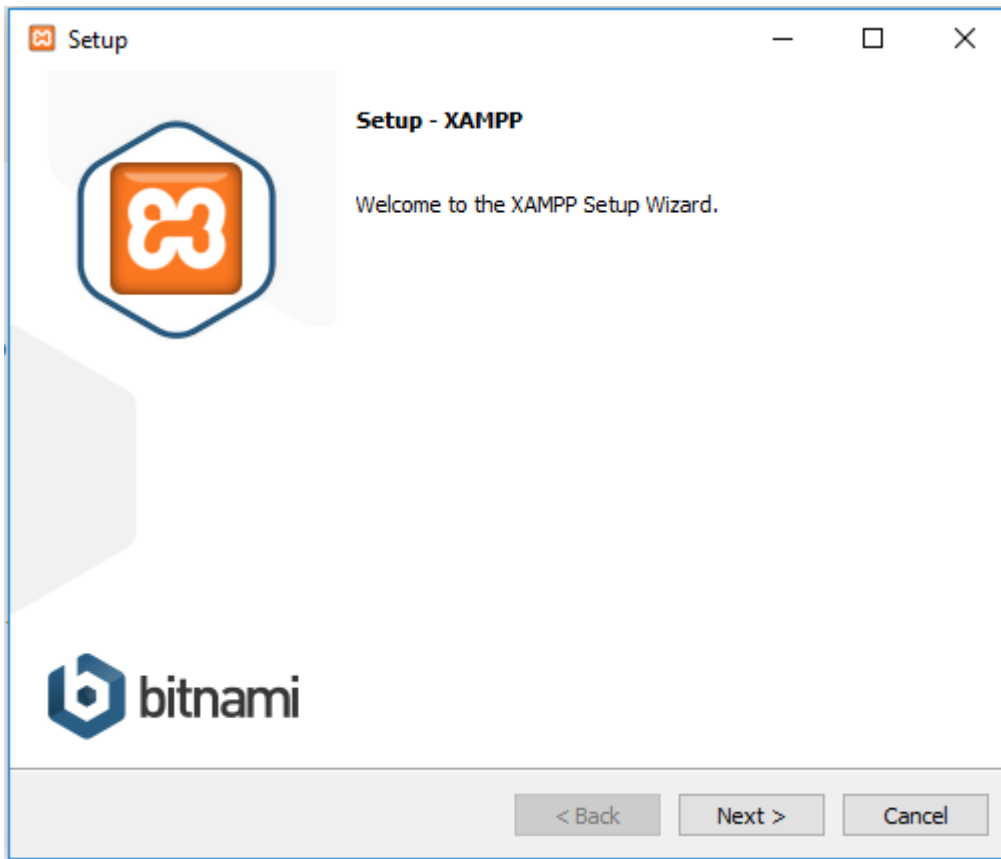


The screenshot shows the 'Configure your app' dialog box in Android Studio. It has a green header with the title 'Configure your app' and a close button (X). On the left, there is a blue icon of an open box with 'SDK' written on it. To the right of the icon are four labels: 'App Name', 'App Description', 'Template', and 'Platform'. The 'App Name' field contains the text 't2aajo00@students.oamk.fi's Blank'. The 'App Description' field contains the placeholder text 'Describe briefly what this app would do, what the value to the users would be and why they would want to use this app.'. The 'Template' field is set to 'Blank' and the 'Platform' field is set to 'Java'. At the bottom left, there is a 'Go Back' button, and at the bottom right, there is a 'Download Template' button.

KUVA 22. Luodun ohjelman nimen ja kuvauksen antaminen.

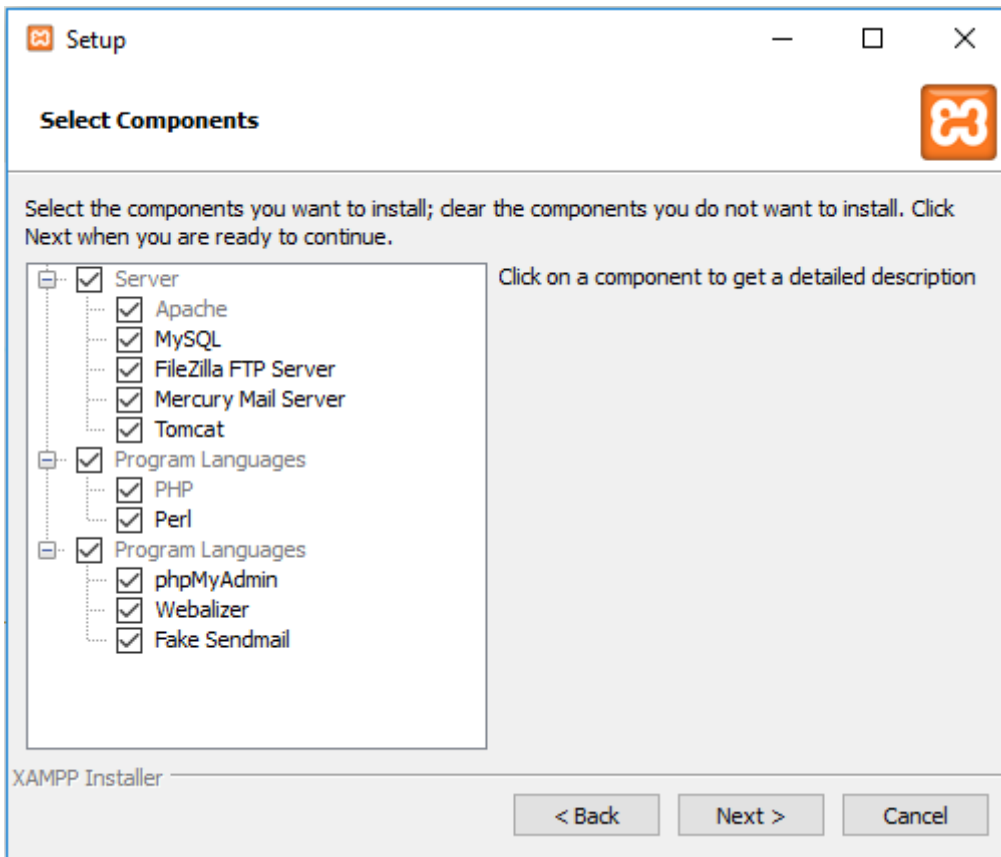
5.2 XAMPP-ohjelman asennus

XAMPP-ohjelman asentaminen alkoi lataamalla ohjelma Apache Friends-sivustolta <https://www.apachefriends.org/index.html>. Sivustolta pystyy valitsemaan, mille käyttöjärjestelmälle XAMPP-ohjelma halutaan asentaa, jonka jälkeen lataaminen pystytään aloittamaan. Samalla Apache Friends-sivustolta löytyy paljon hyödyllistä informaatiota mm. XAMPP-ohjelmasta ja yhteensopivista liitännäisistä. Ohjelman asentaminen on hyvin yksinkertaista ja seuraa normaalia toimintamallia, joka alkaa Tervetuloa-ikkunalla (kuva 23).



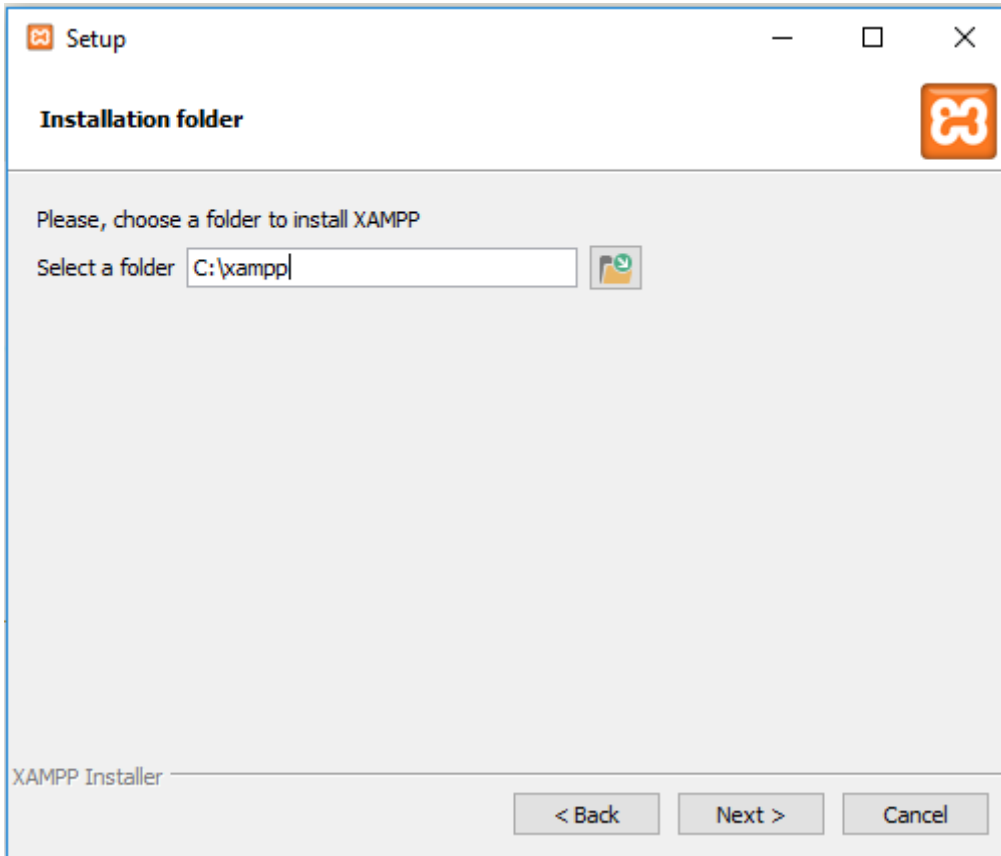
KUVA 23. XAMPP-ohjelman Tervetuloa-ikkuna.

Ohjelman toisessa asennusikkunassa pystytään määrittelemään ohjelmaan halutut komponentit ja ohjelmointikielet. Näistä tärkeimmät tätä ohjelmaa varten ovat MySQL ja phpMyAdmin. Näiden avulla ohjelmassa pystytään hyödyntämään MySQL-tietokantaa sekä luomaan sisäänkirjautumisprosessi käyttäen phpMyAdmin-komponenttia. Muiden komponenttien asennus on vaihtoehtoista. (Kuva 24.)

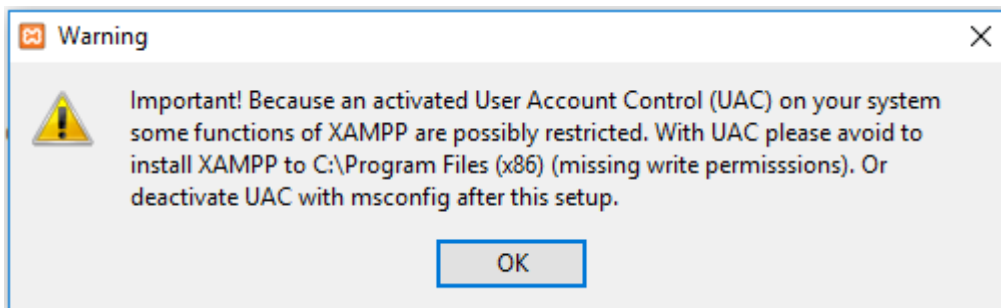


KUVA 24. XAMPP-ohjelman asennuksessa valittavat komponentit.

XAMPP-ohjelman kolmannessa asennusikkunassa on hyvä olla tarkkana, koska ohjelman asentaminen vaatii enemmän toimenpiteitä, jos halutaan asentaa ohjelma esimerkiksi Program Files (x86) -kansioon. Tämän takia ohjelma kannattaa asentaa suoraan oletuslevyn juureen, esimerkiksi C-levyn juureen (C:\). Ohjelman asentaminen Program Files -kansioon on turvallisuussyistä vaikeutettu, jotta ohjelman asentaja ei vahingossa heikennä oman tietokoneen turvallisuutta. (Kuvat 25, 26.)

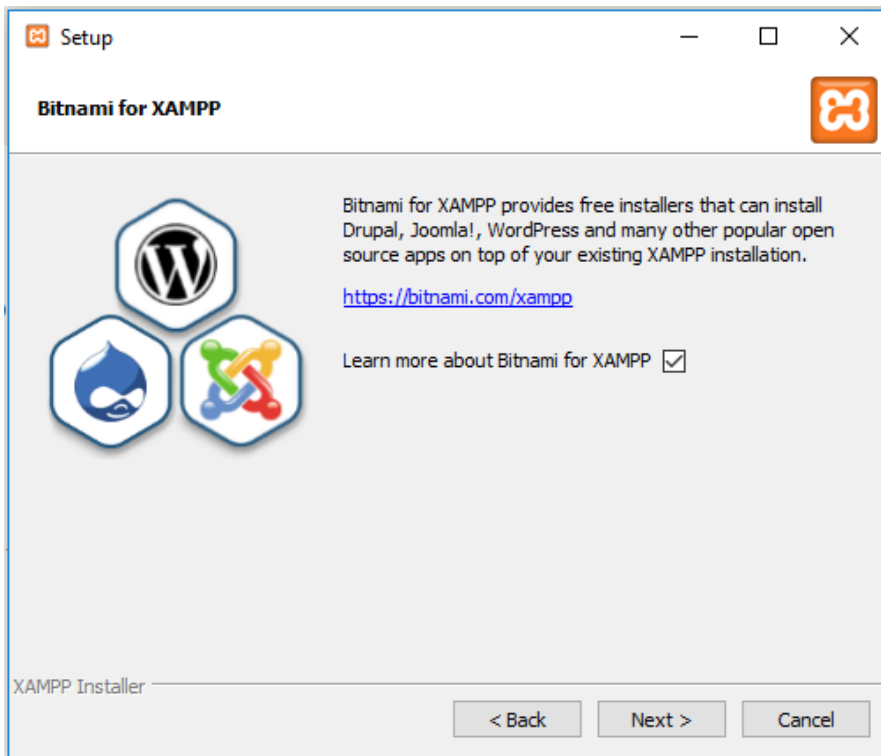


KUVA 25. XAMPP-ohjelman asennuskansio.

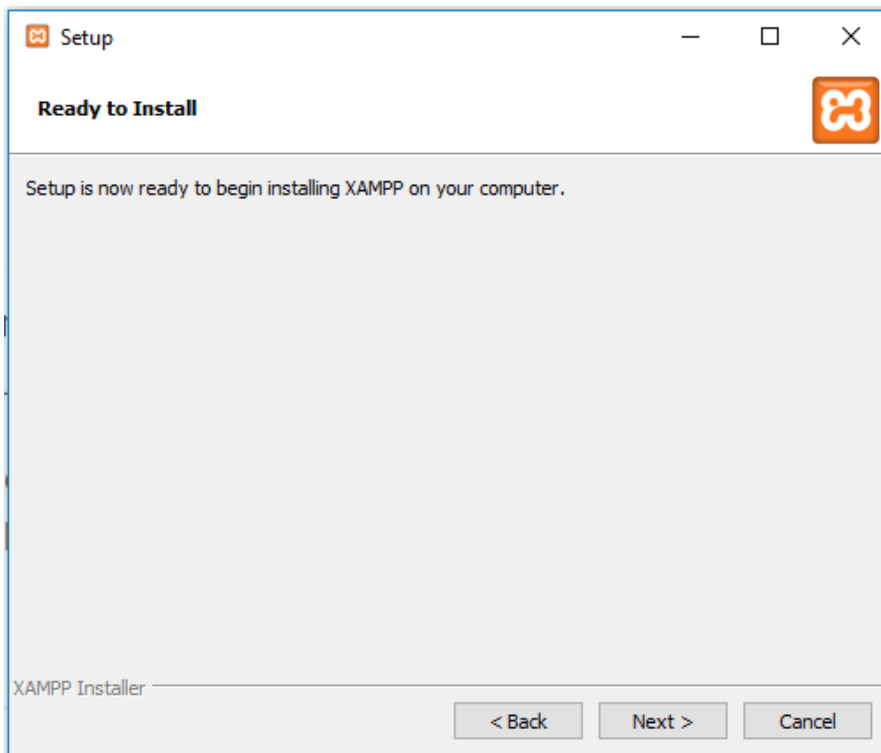


KUVA 26. XAMPP-asennuskansion varoitus.

Ohjelman asennuskansion valinnan jälkeen käyttäjää ohjeistetaan tutustumaan tarkemmin XAMPP-ohjelman käyttöön. Kotisivuilta löytyy paljon hyödyllistä tietoa yhteensopivista ohjelmista, kuten WordPressin käyttämisestä XAMPP-ohjelman kanssa. Tämän jälkeen itse asennus voidaan aloittaa. (Kuvat 27, 28.)

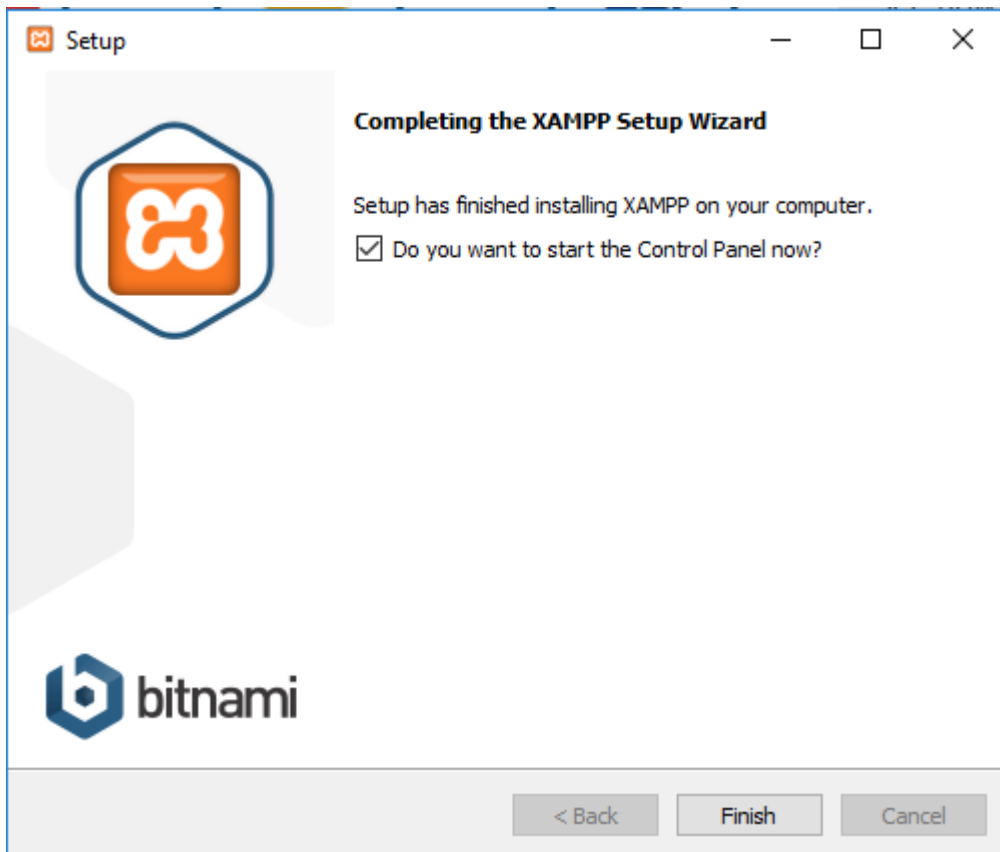


KUVA 27. Käyttäjä ohjeistetaan menemään XAMPP-ohjelman kotisivuille.

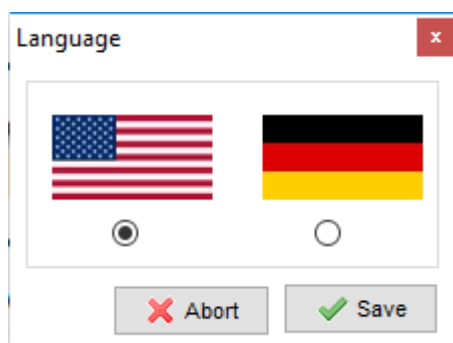


KUVA 28. XAMPP-ohjelman asennuksen aloittaminen.

Asennusvaiheen viimeisenä osana käyttäjä saa valita, käynnistetäänkö XAMPP-ohjelman ohjauspaneeli eli XAMPP-ohjelma itsessään saman tien vai viimeistelläänkö asennus ilman ohjelman käynnistämistä. Ensimmäisellä kertaa, kun käyttäjä käynnistää ohjelman, saa hän valita ohjelman kielen. Vaihtoehtoina XAMPP-ohjelmaa voidaan käyttää joko englanninkielisenä tai saksankielisenä. (Kuva 29.) (Kuva 30.)



KUVA 29. XAMPP-ohjelman asennuksen päättäminen.



KUVA 30. Ohjelman kielen valitseminen.

Näiden vaiheiden jälkeen ohjelma on käyttövalmis ja palvelimen päälle laittaminen onnistuu helposti valitsemalla XAMPP-ohjelman ohjauspaneelista Apachen käynnistämisen ja tietokannan saa päälle valitsemalla ohjauspaneelista MySQL-palvelimen käynnistämisen.


5.3 Tietokannat ja rajapinnat

5.3.1 Login-tietokanta

Ohjelma sisältää login-nimisen tietokannan, jossa on taulukko nimeltä `user_login`. Tässä taulukossa säilytetään käyttäjätunnukset ja salasanat verkkosivujen sisäänkirjautumista varten. Taulukko sisältää neljä kenttää: `id` (tunnistin), `user_name` (käyttäjätunnus), `user_email` (käyttäjän sähköpostiosoite) ja `user_password` (käyttäjän salasana).

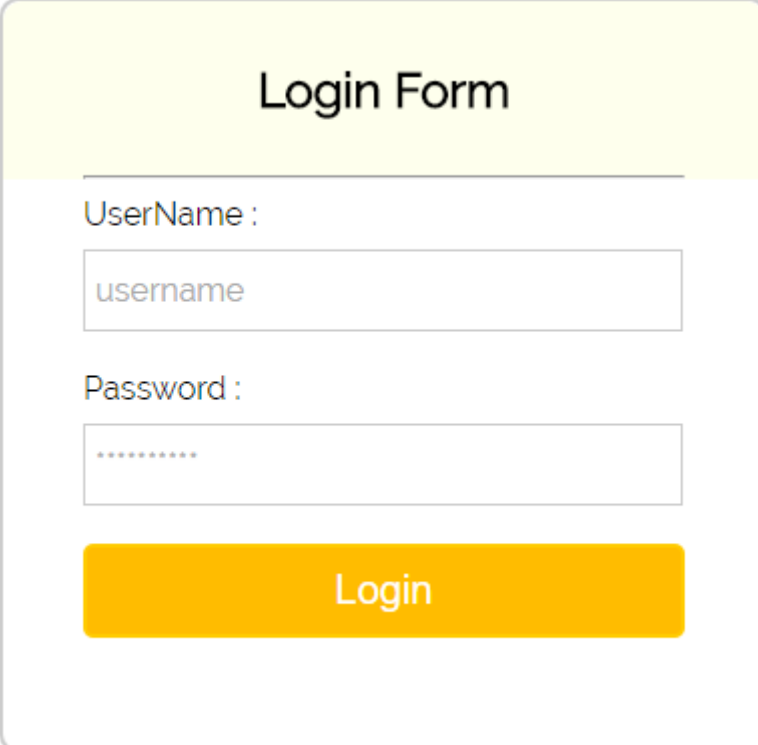
Tietokantaa pääsee muokkaamaan XAMPP-ohjelman ohjauspaneelin kautta käynnistämällä kummatkin Apache-palvelimen ja MySQL-palvelimen. Kun kummatkin palvelimet ovat käynnistyneet, voidaan painaa MySQL-rivillä olevaa "Admin"-painonappia, joka ohjaa käyttäjän selaimessa phpMyAdmin-sivustolle. PhpMyAdmin-sivustolla pystytään luomaan uusia tietokantoja ja hallinnoimaan olemassa olevia tietokantoja halutun näköisiksi.

Login-tietokantaa tehdessä kaikki neljä kenttää asetettiin tilaan, jossa mikään kenttä ei voi olla tyhjänä (not null). Taulukon ensimmäinen kenttä, eli `id`-kenttä, luotiin `int`-arvoiseksi eli kentäksi, joka hyväksyy pelkästään kokonaislukuja. `id`-kenttä asetettiin myös ensisijaiseksi kentäksi eli `primary key` arvoon, mikä tarkoittaa, että tässä kentässä ei voi olla kaksoiskappaleita. Lisäksi `id`-kenttä asetettiin automaattisesti kasvavaksi eli `auto increment` -arvoon, joka johtaa siihen, että edellistä arvoa kasvatetaan aina automaattisesti yhdellä aina uuden rivin luonnin yhteydessä. (Kuva 31.)

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 	int(11)			No	None	AUTO_INCREMENT
2	user_name	varchar(255)			No	None	
3	user_email	varchar(255)			No	None	
4	user_password	varchar(255)			No	None	

KUVA 31. User_login-taulukon kentät.

Taulukon seuraavat kolme kenttää ovat lähes identtisiä, lukuun ottamatta kenttien nimiä ja sisältöjä. Jokainen näistä kentistä luotiin varchar-arvoiseksi, eli kentiksi, jotka hyväksyvät tekstiä arvoikseen. Verkkosivujen sisäänkirjautumisessa palvelin käy katsomassa tietokannan login tietokannan user_login-taulukkoa ja katsoo, että user_name- ja user_password-kentät vastaavat kirjoitettuja käyttäjätunnusta ja salasanaa sisäänkirjautumisikkunassa. (Kuva 32.)



Login Form

UserName :

Password :

Login

KUVA 32. CodeIgniter-viitekehyksellä luotu sisäänkirjautumisikkuna.

Sisäänkirjautumisen onnistuttua tulostetaan käyttäjän käyttäjätunnus ja sähköpostiosoite verkkosivujen pääsivulle (kuva 33).



KUVA 33. Sisäänkirjautumisen onnistuessa päästään verkkosivujen pääsivulle.

5.3.2 Beacon_nest-tietokanta

Beacon_nest-tietokanta on tyhjä tietokanta, jossa ei ole luontivaiheessa yhtään taulukkoa. Tämä sen takia, koska beacon_nest-tietokantaan on tarkoitus luoda uusia taulukoita sitä mukaan, kun päästään keräämään sensori- ja laitetietoja uusista Estimote Beacon- ja Estimote Nearable -laitteista. Jos laitteesta on ennestään jo kerätty sensori- ja laitetietoja, lisää palvelin ne jo olemassa olevaan taulukkoon. Tietokantaa täydennetään Android-sovelluksen ja tietokannan rajapinnassa eli db-connection-osiossa.

5.3.3 JSON-tekstin lähettäminen palvelimelle

Android-ohjelmassa on kolme paikkaa, jossa JSON-tekstiä lähetetään palvelimelle NearableActivity, BeaconActivity ja BeaconDistanceActivity. NearableActivity on ohjelman osa, jossa kerätään Estimote Nearable -laitteen sensori- ja laitetietoja. BeaconActivity on ohjelman osa, jossa kerätään Estimote Beacon -laitteen lämpötila- ja laitetietoja. BeaconDistanceActivity on ohjelman osa, jossa kerätään Estimote Beacon -laitteen läheisyystietoja.

Jokainen osuus koostuu samankaltaisista vaiheista. Ensimmäinen vaihe on Estimote-laitteiden sensori- ja laitetietojen keräys. Toisessa vaiheessa ohjelma luo JsonObjectin eli objektin johon voidaan tallentaa sensori- tai laitetietoa kuvaava

merkkijonotyyppinen avainkenttä, sekä tätä avainkenttää vastaava kerätty sensori- tai laitetieto. Esimerkiksi ensimmäisessä vaiheessa ohjelma kerää Estimote-laitteen UUID:n, eli tunnistimen, josta luodaan merkkijonotyyppinen avainkenttä "uuid". Tälle avainkentälle annetaan arvoksi merkkijonotyyppinen arvo "B9407F30-F5F8-466E-AFF9-25556B57FE6D". JsonObject kerää kaikista kerätyistä sensori- ja laitetiedoista vastaavanlaisia arvopareja, jotka voidaan helposti lähettää Android-ohjelman ja tietokannan väliselle rajapinnalle.

Kolmannessa vaiheessa luodaan URL-tyyppinen osoite, jossa Android-ohjelman ja tietokannan rajapinta sijaitsee. NearableActivityn osalta rajapinta sijaitsee osoitteessa "http://[palvelinkoneen_IPv4_osoite]/db-connection/insertNearable.php". Osoitekenttä [palvelinkoneen_IPv4_osoite] korvataan palvelinkoneen IPv4-osoitteella. Esimerkiksi osoitteella "87.92.86.74". Db-connection-kansio sisältää kaikki kolme rajapintatiedostoa (insertNearable.php, insertBeacon.php ja insertBeaconProxAndDist.php) ja kyseinen kansio löytyy samasta XAMPP-kansiosta, jossa CodeIgniter-viitekehys on.

Neljännessä vaiheessa luodaan yhteys haluttuun rajapintaan käyttäen HttpURLConnection-luokkaa, joka mahdollistaa merkkijonotyyppisen tiedon lähettämisen internet-yhteyden avulla. Koska JSON on merkkijonotyyppinen formaatti, niin HttpURLConnection-luokan käyttäminen soveltuu tiedon lähettämiseen ilman suurempia ongelmia. Ennen tiedon lähettämistä täytyy HttpURLConnection-asetukset määrittää kuitenkin oikeanlaisiksi. Tärkeimpänä asetuksena on HttpURLConnection-luokan pyyntömenetelmän valitseminen. Koska tässä tapauksessa halutaan JSON-teksti lähettää rajapinnalle, täytyy pyyntömenetelmän olla "POST" eli lähetystyyppinen.

Viimeisenä vaiheena on itse JSON-tekstin lähettäminen OutputStream-luokan avulla. Ensimmäisenä HttpURLConnection-luokka pilkkoo JSON-tekstin tavuiksi, jonka jälkeen OutputStream-luokka lähettää pilkotun JSON-tekstin tavujona HttpURLConnection-luokan määrittelemään osoitteeseen.

5.3.4 JSON-tekstin vastaanottaminen palvelimella

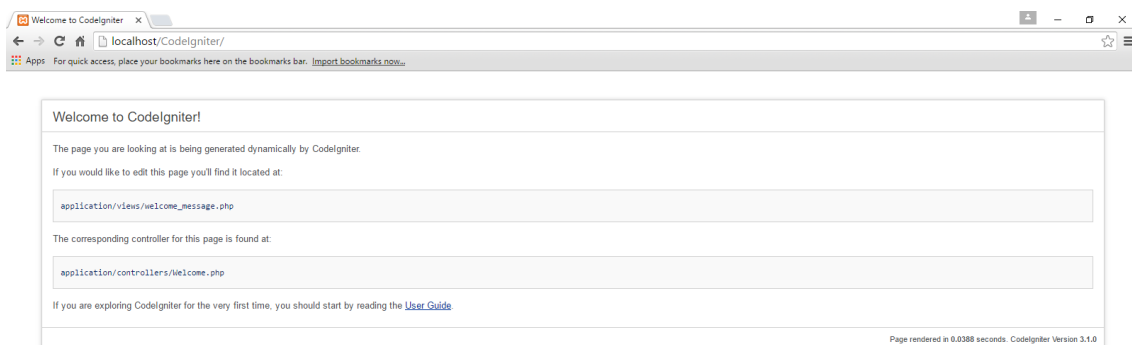
Db-connectionin jokainen rajapintatiedosto ottaa vastaan merkkijonon, joka vastaanottamisen jälkeen purkaa vastaanotetun merkkijonon JSON-tyyppiseksi.

Tämän jälkeen rajapinta hakee jokaista JsonObjectin avainkenttää vastaavat arvot ja sijoittaa ne erillisiin muuttujiin. Kun koko JsonObjectin arvot ovat pilkottu erillisiin muuttujiin luo rajapinta yhteyden palvelimella olevaan beacon_nest-tietokantaan. Jos yhteyden luominen onnistuu tietokantaan, niin tarkistetaan, onko tietokannassa jo olemassa taulukko kyseistä Estimote-laitetta varten. Tämä tarkistus perustuu siihen, jos yksi JsonObjectista tullut muuttuja löytyy tietokannan sisällä olevista taulukon nimistä. Jos nimeä ei löydy taulukosta, luodaan tälle Estimote-laitteelle uusi taulukko beacon_nest-tietokantaan. Jos taas nimi löytyy tietokannasta, lisätään kerätyt sensori- ja laitetiedot kyseiseen taulukkoon.

5.4 CodeIgniter-viitekehysten käyttöönotto ja testaaminen

CodeIgniter-viitekehysten saa ladattua suoraan heidän kotisivuilta <https://www.codeigniter.com/>. Heiltä löytyy myös suuret määrät tietoa viitekehyksestä ja kattavat ohjeet sen käytöstä.

Käytännössä CodeIgniter-viitekehysten käyttöönotto ei vaadi mitään muuta, kuin viitekehysten lataamisen ja sen sijoittamisen palvelimelle. XAMPP-ohjelman asennuksen jälkeen CodeIgniter-viitekehys voidaan esimerkiksi sijoittaa suoraan kansioon "C:\xampp\htdocs". Sen jälkeen, kun CodeIgniter-kansio on siirretty XAMPP-ohjelman htdocs-kansioon ja XAMPP-ohjelman Apache-palvelin on laitettu päälle, päästään suoraan katsomaan CodeIgniter viitekehystä menemällä osoitteeseen localhost/[CodeIgniter_kansion_nimi]. Suositeltavaa on CodeIgniterin latauksen jälkeen muuttaa ladatun kansion nimi "CodeIgniter"-nimiseksi, koska tämän jälkeen selaimen tulevan kansion nimi on selvempi, eikä version numero näy selaimen osoitteessa. (Kuva 34.)



KUVA 34. CodeIgniter viitekehys asennuksen jälkeen.

6 TYÖN TULOKSET

Tässä luvussa käydään läpi työn tuloksia koskien kaikkia kolmea luotua osa-alueetta. Yhtenä osa-alueena työssä oli luoda toimivat verkkosivut, jossa pääpaino oli verkkosivujen sisäänkirjautumisessa. Sisäänkirjautumisosiossa vaatimuksena oli luoda luotettava sisäänkirjautuminen jonka kiertäminen mm. admin-sivun selainosoitteen kopioimisella ei pystytä kiertää. Lisäksi admin-sivulle ei pääse käyttämällä selaimen ”Palaa takaisin”-painikkeella uloskirjautumisen jälkeen. Ensimmäiseen osuuteen kuului myös tietokannasta tiedon haku ja tulostaminen admin-sivulla näkyviin taulukoihin, jotka on jaoteltu eri Estimote-laitteita kohden.

Toisena osuutena oli luoda Android-mobiilisovellus, joka Estimote-laitteeseen yhdistyttyä pystyy keräämään Estimote-laitteista sensori- ja laitetietoja, sekä näyttämään näitä tietoja Android-sovelluksessa. Sensori- ja laitetiedot pitää, myös pystyä lähettämään rajapinnoille asynkronisesti, sekä sovelluksessa ikkunasta toiseen vaihtaminen pitää onnistua luontevasti.

Kolmantena osuutena työssä oli tietokannan luonti, ylläpito ja päivittäminen. Tähän osioon sisältyy myös mobiilisovelluksen ja tietokannan väliin tulevien rajapintojen luonti ja huolehtiminen siitä, että ohjelma pystyy ottamaan vastaan kaikkia kolmea Estimote-laitetyyppejä vastaavia tietoja.

6.1 Verkkosivut

Kokonaisuudessaan verkkosivuista tuli luotettavat toimivuuden ja sisäänkirjautumisen suhteen. Lisäksi verkkosivuilla Estimote-laitteiden tietojen näyttäminen taulukkotyyppisenä toimii hyvin, vaikka laitteiden tunnistamista olisi voinut hiukan parantaa. Verkkosivut koostuvat käytännössä kahdesta sivusta: sisäänkirjautumissivusta ja verkkosivujen pääsivusta. Verkkosivut päivittyvät muutaman sekunnin välein, jotta käyttäjän ei tarvitse aina manuaalisesti päivittää verkkosivuja, kun tietokantaan tulee uutta dataa Estimote-laitteista. Tämä ominaisuus helpottaa ja selkeyttää tietojen katsomista verkkosivuilla ja tekee verkkosivujen toiminnasta huomattavasti sujuvampaa. (Kuva 35.)

Admin Page

Hello Jonsku!

Welcome to Admin Page

Your Username is Jonsku
Your Email is t2aajo00@students.oamk.fi

Device MAC Address: 2142063EBDE4C9DD

ID	UUID	Major	Minor	Device	Temperature	Device_moving	Accelerometer	Orientation	Proximity	Battery_level	Firmware	Broadcasting_power	Date
1	b043b86-ca76-45ec-9bdf-6af92142063e	48612	51677	Neearable	23.6°C	No	x: 47 y: 969 z: 250	VERTICAL	NEAR	HIGH	Unknown	0 dBm	2016-08-22 08:12:36
2	b043b86-ca76-45ec-9bdf-6af92142063e	48612	51677	Neearable	23.6°C	No	x: 47 y: 969 z: 250	VERTICAL	NEAR	HIGH	Unknown	0 dBm	2016-08-22 08:12:37

Device MAC Address: FDBBDF4BB26F

ID	UUID	Major	Minor	Device	Temperature	Device_moving	Battery_level	Firmware	Broadcasting_power	Advertising_interval	Date
1	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	25.8°C	Not in motion	100%	A3.2.0	-4 dBm	950ms	2016-08-15 09:10:40
2	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	25.5°C	Not in motion	100%	A3.2.0	-4 dBm	950ms	2016-08-15 09:10:40
3	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	25.2°C	Not in motion	100%	A3.2.0	-4 dBm	950ms	2016-08-22 08:12:21
4	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	24.8°C	Not in motion	100%	A3.2.0	-4 dBm	950ms	2016-08-22 08:12:22

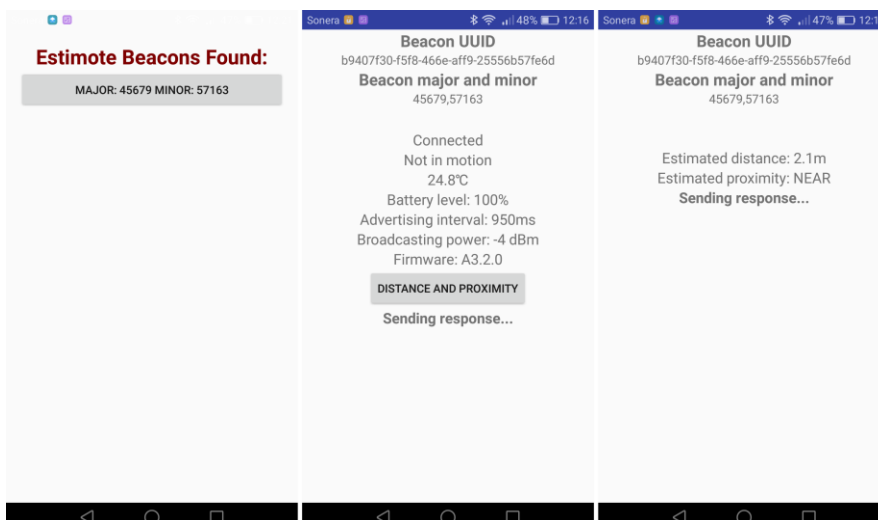
Device MAC Address: FDBBDF4BB26F_DP

ID	UUID	Major	Minor	Device	Distance	Proximity	Date
1	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	7.5m	FAR	2016-08-15 09:10:49
2	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	3.9m	FAR	2016-08-15 09:10:51
3	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	3.9m	FAR	2016-08-15 09:10:51
4	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	5.4m	FAR	2016-08-15 09:10:52
5	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	4.3m	FAR	2016-08-15 09:10:55
6	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	0.9m	NEAR	2016-08-22 08:12:29
7	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	0.9m	NEAR	2016-08-22 08:12:29
8	b9407f30-f5f8-466e-aff9-25556b57fe6d	45679	57163	Beacon	0.7m	NEAR	2016-08-22 08:12:30

KUVA 35. Verkkosivujen pääsivu.

6.2 Mobiilisovellus

Mobiilisovelluksesta tuli toimiva kokonaisuus, jossa yhdistyy Estimote-laitteiden tietojen keräys, tietojen suora näyttö sovelluksessa ja tietojen lähettäminen puhelimen ja tietokannan välisiin rajapintoihin. Sovelluksessa navigointi eri Estimote-laitteisiin toimii kätevästi painonapeilla ja ”Takaisin”-painikkeita painamalla. Estimote-laitteiden tunnistaminen mobiilisovelluksessa on luotettavinta silloin kun Estimote-laitteiden etäisyys puhelimeen on mahdollisimman lyhyt. (Kuva 36.)



KUVA 36. Estimote Beacon -laitteen koko ohjelmapolku.

6.3 Tietokantakokonaisuus

Projektin tietokanta kokonaisuus koostui lopulta kahdesta MySQL-tietokannasta, login ja beacon_nest ja kolmesta rajapinnasta: insertBeacon, insertBeaconProxAndDist ja insertNearable. Tietojen lähettämisvaiheessa luotiin tarkistuksia mobiilisovelluksessa, jotta kaikki asetukset JSON-tekstin lähettämiseen ovat oikein, jotta data pystytään ottamaan vastaan kaikissa rajapinnoissa, kuten niiden kuuluu. Jos tiedon lähettämässä on jotain, joka estää tiedon vastaanottamisen rajapinnassa, niin mobiilisovellus ilmoittaa siitä ennen tiedon lähettämistä. Tämä loi tietokantajärjestelmästä vakaan ja mahdollisti tiedon onnistuneen lähettämisen. Näiden tarkistusten takia pystyttiin luottamaan siihen, että tiedon vastaanottaminen palvelimen puolella onnistui hyvin, eikä data katoa, kun tieto lähetetään mobiilisovelluksesta.

Kummatkin tietokannat, login ja beacon_nest, ovat hyvin yksinkertaisia tietokantoja, mikä edesauttoi tietojen poimimista tietokannoista verkkosivuille ja tietojen lisäämistä näihin tietokantoihin. Koska tietokantaan lähetetyt datat on tarkistettu ennen rajapintakutsuja, näiden tietojen sijoittaminen tietokantoihin sujui ongelmitta ja tietojen poimiminen näistä tietokannoista toimii luotettavasti.

7 YHTEENVETO

Opinnäytetyön tarkoituksena oli luoda toimiva kokonaisuus, jossa pääpaino olisi Estimote Beacon- ja Estimote Nearable -laitteilla. Näiden lisäksi opinnäytetyössä haluttiin lisätä monipuolisuutta yhdistämällä Estimote-laitteisiin verkkosivujen sisäänkirjautumisprosessi ja tietokannat. Tämä loi opinnäytetyöstä kattavan kokonaisuuden ja teki työstä tarpeeksi haastavan, mutta ei liian haastavaa, että opinnäytetyötä ei saataisi valmiiksi halutussa ajassa. Luotua kokonaisuutta olisi mahdollista vielä jatkokehittää usealla eri alueella. Esimerkiksi sovelluksen avulla voitaisiin mitata talon lämpötiloja eri paikoissa Estimote Beacon- ja Nearable -laitteiden avulla ja kirjata tulokset luotuun verkkosivuun.

Työssä haastavin osuus oli tutustua Estimote-laitteisiin ja tutkia, mihin kaikkeen nämä laitteet pystyvät. Estimoten omat verkkosivut auttoivat melko paljon tässä, mutta osittain tieto oli melko vaikeasti löydettävissä tai tieto oli vanhentunutta. Toisena ongelmana oli joidenkin laitteiden paristojen kestot, koska valmistajan kertomat paristojen kestoajat olivat ilmeisesti pääasiassa maksimiaikoja. Tämä ilmeni siinä vaiheessa, kun laite oli ollut pari kuukautta käytössä. Tämän jälkeen osa laitteista sammui eikä näihin saatu yhteyttä tämän jälkeen, vaikka paristoille oli luvattu käyttöikä vuodesta ylöspäin.

Lopputuloksena kokonaisuudesta saatiin toimiva ja stabiili, eikä ohjelman kovaan käyttö kaatanut luotua ohjelmaa. Verkkosivut toimivat moitteettomasti ja näyttivät kerätyt tiedot loogisesti käyttäjälle. Tietokannat olivat suhteellisen yksinkertaisia, joten näiden toiminta rajapintojen kanssa oli luotettavaa, eikä rajapintojen kanssa näin tullut ongelmia. Mobiilisovellus toimi sulavasti, ja jos sovelluksessa kohdattaisiin jokin ongelma, kuten tiedon lähettämisessä rajapinnalle, ilmoittaa sovellus siitä käyttäjälle ilman, että sovellus kaatuu.

LÄHTEET

1. Beacon Tech Overview. 2014. Estimote Inc. Saatavissa: <http://developer.estimote.com/>. Hakupäivä 27.7.2016.
2. "What is Eddystone™?". 2014. Estimote Inc. Saatavissa: <http://developer.estimote.com/eddystone/>. Hakupäivä 27.7.2016.
3. "What is iBeacon?". 2014. Estimote Inc. Saatavissa: <http://developer.estimote.com/ibeacon/>. Hakupäivä 27.7.2016.
4. Socha, Witek 2016. Technical specification of Estimote Beacons and Stickers. Saatavissa: <https://community.estimote.com/hc/en-us/articles/204092986-Technical-specification-of-Estimote-Beacons-and-Stickers>. Hakupäivä 28.7.2016.
5. Borowicz, Wojtek 2016. "What is a beacon protocol? Can beacons broadcast multiple packets simultaneously?". Saatavissa: <https://community.estimote.com/hc/en-us/articles/208546097>. Hakupäivä 28.7.2016
6. "What are Estimote Stickers? What are nearables?". 2014. Estimote Inc. Saatavissa: <http://developer.estimote.com/nearables/>. Hakupäivä 29.7.2016.
7. "Which beacons should I choose?". 2015. Estimote Inc. Saatavissa: <http://estimote.com/products/>. Hakupäivä 29.7.2016.
8. Wojtek Borowicz. "What are nearables? What is Nearable protocol?". 2015. Saatavissa: <https://community.estimote.com/hc/en-us/articles/206409488-What-are-nearables-What-is-Nearable-protocol->. Hakupäivä 29.7.2016.
9. Getting Started with Bluetooth Low Energy. 2016. Safari Books Online. Saatavissa: <https://www.safaribooksonline.com/library/view/getting-started-with/9781491900550/ch01.html>. Hakupäivä 29.7.2016.

10. Low energy. 2016. Bluetooth SIG. Saatavissa: <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>. Hakupäivä: 29.7.2016.
11. Cortex-M0 Processor. 2016. ARM Ltd. Saatavissa: <https://www.arm.com/products/processors/cortex-m/cortex-m0.php>. Hakupäivä 1.8.2016.
12. Technosip review – XAMPP. 2016. TechnoSIP Inc. Saatavissa: <http://www.technosip.com/toolsreviews/technosip-review-xampp/>. Hakupäivä 1.8.2016.
13. XAMPP. 2016. PortableApps.com. Saatavissa: <http://portableapps.com/apps/development/xampp>. Hakupäivä 1.8.2016
14. CodeIgniter at a Glance. 2016. British Columbia Institute of Technology. Saatavissa: http://www.codeigniter.com/user_guide/overview/at_a_glance.html. Hakupäivä 2.8.2016.
15. A Brief History of CodeIgniter. 2016. EllisLab.com. Saatavissa: <https://ellislab.com/codeigniter>. Hakupäivä 2.8.2016.
16. Design Patterns – MVC Pattern. 2016. Tutorialspoint.com. Saatavissa: http://www.tutorialspoint.com/design_pattern/mvc_pattern.htm. Hakupäivä 2.8.2016.
17. Camden, Raymond 2013. Deeper In the Brackets Editor. Tutspus. Saatavissa: <http://code.tutspus.com/tutorials/deeper-in-the-brackets-editor--net-35527>. Hakupäivä 2.8.2016.
18. Meet Android Studio. 2016. Developer.android.com. Saatavissa: <https://developer.android.com/studio/intro/index.html>. Hakupäivä 3.8.2016.
19. History of PHP. 2016. The PHP Group. Saatavissa: <http://php.net/manual/en/history.php.php>. Hakupäivä 3.8.2016.
20. "What can PHP do?". 2016. The PHP Group. Saatavissa: <http://php.net/manual/en/intro-whatcando.php>. Hakupäivä: 3.8.2016.

21. Darrell, Robert 2011. The History of HTML. Saatavissa: <http://www.ironspider.ca/webdesign101/htmlhistory.htm>. Hakupäivä 4.8.2016.
22. Wise, Cheryl D. 2003. History of HTML & CSS. Saatavissa: <http://www.wdtonline.com/wdtMagazine/MemberWorks/Wiser-Ways/csshtml.htm>. Hakupäivä 5.8.2016.
23. Borowicz, Wojtek. 2015. First steps with Estimote Cloud. Saatavissa: <https://community.estimote.com/hc/en-us/articles/213844478-First-steps-with-Estimote-Cloud>. Hakupäivä 9.8.2016.
24. Java SE Development Kit 8 Downloads. 2016. Oracle.com. Saatavissa: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Hakupäivä 27.8.2016.