

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2020

Iiro Koskialho

SHIMMER-RAJAPINTADATAN HYÖDYNTÄMINEN IMOTIONS- SOVELLUKSESSA

Iiro Koskialho

SHIMMER RAJAPINTADATAN HYÖDYNTÄMINEN IMOTIONS SOVELLUKSESSA

- Shimmer-laitteen rajapinnan luonti ja hyödyntäminen iMotions sovelluksessa käyttäen MATLAB-sovellusta

Rajapinta on ohje tai määritelmä, miten eri ohjelmat voivat ottaa yhteyksiä toisiinsa ja vaihtaa tietoja. Terveysteknologian laboratoriossa ei ole vastaavanlaista ratkaisua tai esimerkkiä, miten rajapinta voidaan ratkaista iMotionsiin. Työn tavoite oli kehittää Turun Ammattikorkeakoulun terveysteknologian Health Tech Labille rajapintaratkaisu esimerkkinä iMotions-sovellukselle saapuvaan dataan. Rajapinnan tarkoitus on lähettää Shimmer-laitteen mittaamaa fysiologista dataa MATLAB-sovelluksen kautta iMotions-sovellukseen.

Työ toteutettiin laboratorion Shimmer-laitteilla sekä käyttäen Mathworksin MATLAB-sovelluksella ja iMotionsin iMotions-sovelluksella, jonka tarkoitus on kerätä ja yhdistää eri fysiologisia dataja sisältäviä mittaustietoja yhteen tutkimukseen. Työssä käydään läpi rajapintatyyppisiä, niiden standardeja ja toimintamalleja. Työn toteutuksessa käydään läpi, miten rajapinta muodostettiin MATLABissa ja datan muotoilua tiettyyn muotoon. Tehty rajapinta vaatii toistaiseksi ohjelmoidun funktion ajamisen manuaalasti, jotta dataa aletaan mittaamaan ja lähettämään eteenpäin. Työssä kohdattiin useita ongelmia Shimmer-laitteiden yhdistyvyydessä ja toimivuudessa etenkin MATLABin sovelluksen kanssa. Ongelmat saatiin pääosin ratkaistua yhdistämällä laite uudestaan ja kirjoittamalla uudet asetukset Shimmeriin.

Työn tavoite eli rajapinnan luominen onnistui suhteellisen hyvin. Toimiva yhteys saadaan muodostettua iMotionsiin. Ainoastaan iMotions-sovelluksessa oleva tyylitiedosto vaatii jatkokehitystä, jotta data saataisiin visualisoitua ja se olisi luettavissa mittauskäyriltä. Työllä on tarvetta jatkokehitykselle ja yleiselle hienosäädölle. Siitä on kuitenkin hyötyä sen toimiessa esimerkkinä jatkokehitykselle, sillä vastaavaa esimerkkiä juuri näillä laitteilla ei ole aiemmin tehty Health Tech Labissa.

ASIASANAT:

Rajapinta, Ohjelmointirajapinta, Shimmer, Matlab, API,

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2020 | 22 number of pages, 2 number of pages in appendices

Iiro Koskialho

UTILIZING SHIMMER PROGRAMMING INTERFACE DATA IN IMOTIONS

Creating programming interface and utilizing Shimmer-data in iMotions using MATLAB-application.

Objective of the thesis was to develop an programming interface for Health Tech Lab of Turku University of Applied Sciences. The main purpose for the programming interface is to send Shimmer's measured physiological data to iMotions-application, via iMotions.

Thesis was carried out using Health Tech Lab's Shimmer devices and using Mathworks MATLAB application, as well as using iMotions iMotions-application. iMotions is mainly used to collect and combine measured physiological data to a one study. This study can be used for example research purposes. In the theory part, the study goes through different API types, and their standards and operation models. Implementation part of the thesis focuses on how the programming interface was built in MATLAB-application and how the data was formatted to be usable for iMotions. Programming interface need to be manually used as of now. Functions manual usage is required to be able to forward the measured data. During the thesis, few problems stood out. Problems were mainly from Shimmer device, usually considering the connection to the MATLAB-application. These were mainly sorted, by removing the device pairin from device manager, and rewriting the configuration to the device.

Objective of the work, creating programming interface was mostly successful. Functional connection to the iMotions from MATLAB was formed. The XML-file in iMotions needs to be fine tuned more, for iMotions to be able to plot the data visually. The programming interface requires more further development, and general fine adjustments. However there are advantages for the work as an example. The works serves as an example for the laboratory to make further similar developments. This kind of example has not been made in the laboratory before, so it's uniqueness is useful to develop the Health Tech Lab further

KEYWORDS:

Shimmer, API, Programming Interface, Matlab.

SISÄLTÖ

KÄYTETYT LYHENTEET	6
1 JOHDANTO	7
2 RAJAPINTA(API) JA TIEDONSIIRTO	8
2.1 Rajapinta eli API	8
2.2 Rajapintojen käyttäminen ja hyödyntäminen	9
2.3 Rajapinta-protokollat	10
2.4 Rajapintadatan tiedonsiirto	12
3 TOTEUTUS	13
3.1 Shimmer rajapinta	13
3.1.1 Shimmer API:n tarkoitus ja teknisiä tietoja	13
3.1.2 Shimmer-rajapinnan luominen ja käyttöönotto	14
3.2 Shimmer-rajapinta datan käsittely iMotions-sovelluksessa	14
3.2.1 iMotions-sovelluksen tarkoitus	14
3.2.2 Shimmerin rajapinta yhteyden luonti iMotions-sovellukseen	15
3.3 Rajapinnan luominen MATLAB-ympäristöön.	18
3.3.1 MATLAB-ympäristön esittely	18
3.3.2 MATLAB-ympäristön luonti	18
3.3.3 Shimmer-rajapinnan luonti MATLAB-ympäristöön	20
4 YHTEENVETO	22
4.1 Kohdatut ongelmat.	22
4.2 Jatkokehitys	23
LÄHTEET	25

TAULUKOT

Taulukko 1. iMotions-sovelluksesta tallennettu EKG:n data.	17
Taulukko 2. MATLAB:n funktion tallentama EKG:n data.	19

KUVAT

Kuva 1. URL-palkin osat.	12
Kuva 2. Shimmer-yksikkö.	13
Kuva 3. Consensys-ohjelman päänäkymä, jossa valittuna Shimmer-yksikkö ohjelmoitavaksi.	14
Kuva 4 EKG-Shimmer asetukset iMotions-sovelluksessa.	16
Kuva 5. Lisätty koodi, jolla dataa lähetetään iMotions-sovellukseen.	20

KÄYTETYT LYHENTEET

ACID	SOAP-standardin periaate. Atomicity, Consistency, Isolation, Durability., joka määrittää tietojen eheyden
API	Ohjelmointirajapinta. Application Programming Interface
CSV	Comma-separated values. Käytetään tallentamaan yksinkertaista taulukkomuotoista tietoa tekstitiedostomuotoon.
HTML	HyperText Markup Language. Standardi jolla merkitään hyperlinkkejä.
HTTP	Protokolla, jota käytetään tiedon siirtoon selaimissa ja WWW-palvelimissa. Lyhenne sanoista Hypertext Transfer Protocol
JSON	JavaScript Object Notation. Avoimen standardin tiedonvälitysmuoto
REST	Representational State Transfer. HTTP-protokollan arkkitehtuurimalli rajapintoihin.
RESTful	REST-arkkitehtuuria vastaava API
SMTP	Simple Mail Transfer Protocol. TCP-protokolla, jolla välitetään viestejä.
SOAP	Simple Object Access Protocol. Viestipohjainen tietoliikenneprotokolla
SSL	Secure Sockets Layer. Salausprotokolla, jolla suojataan käyttäjien tietoliikenne.
TCP	Transmission Control Protocol
URL	Uniform Resource Locator. Merkkijono esimerkiksi selaimessa, jolla osoitetaan WWW-sivustoja.
XML	Extensible Markup Language. Tiedonvälityformaatti järjestelmien välillä.

1 JOHDANTO

Ohjelmointirajapintoja käytetään määrittelemään, miten eri ohjelmat toimivat ja vaihtavat tietoja keskenään. Rajapinnan käyttö ohjelmistokehityksessä helpottaa työn määrää. Ohjelmistokehittäjien ei tarvitse kehittää pyörää uudelleen, vaan voivat käyttää jo olemassa olevia asioita kehityksessään. Rajapintojen käyttö ohjelmistoOpinnäytetyön toimeksiantajana toimii Turun ammattikorkeakoulun Health Tech Lab. Health Tech Lab sijaitsee ICT-cityn tiloissa, ja se on osa Turun ammattikorkeakoulun Tieto ja Viestintäteknikan Insinööri-koulutuksen terveysteknologian erikoistumisalaa. Terveysteknologian yksikössä on tutkimusryhmä, ja osana terveysteknologiaa on laboratorion kehitysympäristö ja toimiva test bed-ympäristö. Laboratorio tarjoaa koulutusta, tuotekehitystä, käytettävyydestä ja innovaatiota yksityiselle sektorille, sekä osallistuu olennaisena osana terveysteknologian erikoistumisalan koulutusta. Health Tech Lab ottaa työharjoittelijoita ensisijaisesti terveysteknologian opiskelijoista. Laboratorio sisältää useita käytettävyydestä -laitteita, kuten esimerkiksi Tobii Pro Glasses 2- lasit, joilla voidaan seurata katsetta. Laboratoriossa voidaan suorittaa fysiologisia mittauksia ihmisestä ja hyödyntää niitä tutkimustarkoituksiin. Health Tech Lab tarjoaa opiskelijoilleen myös opinnäytetyö ja työharjoittelu mahdollisuuksia. Terveysteknologian tutkimusryhmän tutkimus ja kehittämis teemoihin kuuluu esimerkiksi terveysteknologinen tuotekehitys, terveyteen ja hyvinvointiin liittyvien palvelujen kehittäminen, tiedonhallinta ja käytettävyys [19, 20, 21]

Toimeksiannon tarkoitus oli kehittää rajapinta siirtämään Shimmerin dataa iMotionin ja MATLABin välillä. Opinnäytetyön toimeksianto suoritettiin fysiologiseen mittaukseen tarkoitetun Shimmer-yksikön mittausdatan rajapintoja käyttäen iMotions-sovelluksessa. iMotions-sovellus on tarkoitettu yhdistämään monien eri sensorien mittaamaa dataa ja visualisoimaan niitä tutkimuksissa. Shimmerin mittaamaa dataa hyödynnetään MathWorksin MATLAB-sovelluksessa. MATLABia käytetään algoritmien toteuttamiseen ja ohjelmien toteuttamiseen. MATLABia käytetään analysoimaan Shimmerin mittaamaa fysiologista dataa, ja viemään sitä iMotions-sovellukseen rajapintoja hyödyntämällä. Valmistettu esimerkki-rajapintaratkaisua voidaan tulevaisuudessa käyttää terveysteknologian laboratorion edelleen kehittämiseen ja iMotions API:en luomiseen MATLABin kautta.

2 RAJAPINTA(API) JA TIEDONSIIRTO

2.1 Rajapinta eli API

Rajapinta eli ohjelmointirajapinnan tarkoitus on määritellä miten eri ohjelmat voivat vaihtaa tietoja ja kommunikoida keskenään. Ohjelmointirajapintoja eli API:a (Application Programming Interface) on erilaisia. API voi olla tyypiltään datarajapinta, jolla rajapintaa hyödyntävät ohjelmat voivat siirtää ja hyödyntää dataa toisissa järjestelmissä. API voi myös olla toiminnallinen rajapinta, jolloin käyttäjä tai toimija voi muuttaa järjestelmän tietoja. Rajapintoihin on kolme erilaista lähestymistapaa. Yksityinen, jossa API on vain kehittäjien sisäisessä käytössä. Yksityisen API:n suurin hyöty on käyttäjien ja yhtiöiden määräämisoikeus, mitä API:lla tehdään. Yhteistyömallissa API jaetaan tietyille yhteistyökumppaneille, jotka hyödyntävät rajapintadataa omissa tarkoituksissaan. Yhteistyömallin hyöty on ylläpitää laatua, ja saada ylimääräistä tuottoa. Avoimessa rajapinta mallissa API on saatavilla kaikille. Avoin rajapinta mahdollistaa kolmannen osapuolien kehittäjien kehittää ratkaisuja API:a hyödyntäen. Avoin rajapinta ei kuitenkaan mahdollista hallintaa, ketkä sitä käyttävät. [1, 2, 3]

Ensimmäiset API:t nousivat esille tietojenkäsittelyn alkuvaiheessa. API:en tyypillinen käyttötarkoitus oli varhaisvaiheessa toimia käyttöjärjestelmien kirjastona. API:t olivat tyypillisesti paikallisia ja sidottuja yhteen järjestelmään, niiden joskus välittäessä tietoa eri järjestelmien välissä. [3]

Etäkäyttöisistä API:sta käytetään nimitystä Remote API. Etäkäyttöiset API:t on suunniteltu olemaan vuorovaikutuksessa viestintä verkon kautta. Etäkäyttöisyys tarkoittaa, että API:n käyttämät resurssit ovat paikallisen järjestelmän ulkopuolella. Koska eniten käytetty viestintäverkko on internet, on valtaosa rajapinnoista suunniteltu käytettäväksi Web-standardeilla. Näistä käytetään nimitystä Web API. Suurin osa etäkäyttöisistä API:sta on Web API-muotoisia. Web-API:t yleensä käyttävät HTTP-pyyntöjä ja muodostavat vastauksista yleensä XML- tai JSON-muotoisen tiedoston. Edellä mainitut tiedostomuodot ovat ensisijaisia, koska näistä on toisten sovellusten yksinkertaista hallita, hyödyntää ja muuttaa tietoja. Esimerkki etäkäyttöisestä API:sta on Googlen API, jota hyödyntämällä Google voi hallita pääsyä esimerkiksi Google Maps-palveluun. Tätä hyödyntämällä voidaan Google API:a käyttävällä sivustolla esimerkiksi näyttää palveluntarjoajan toimitilojen sijainnin tai reitin sinne käyttäjän sijainnista. [3, 4]

2.2 Rajapintojen käyttäminen ja hyödyntäminen

Rajapintoja käytetään monissa ohjelmistokehitys asioissa. API:n käyttö helpottaa kehittäjien työtä. Jos esimerkiksi kehittäjä haluaa sulauttaa selaimelle kaksi web sivustoa, kehittäjän ei tarvitse ohjelmoida omaa selainta ratkaisua varten. Kehittäjä voi rajapintojen avulla sulauttaa toisen web-sivuston ratkaisuna. Jos kehittäjät eivät voisi hyödyntää rajapintoja, he joutuisivat kehittämään aina oman ohjelmistoratkaisun kommunikoimaan sulautettavien asioiden kanssa. [4]

API:n toinen tärkeä käyttökohde on tietoturva. API:n avulla ohjelmisto voi viestiä toisten tuotteiden ja palveluiden kanssa ilman tarvetta tietää tarkkaa toteutusta. API:a hyödynnetään tilanteissa, jossa ohjelmistolla ei ole pääsyä tiettyyn ohjelmistoon tai tietoihin. API avaa pääsyn ohjelmiston resursseihin, samalla pitäen API:n kehittäjillä tietoturvan ja hallinnan. Rajapinnan kehittäjän päivittäessä rajapintaansa, päivittyä ja paranee myös jo käytössä oleva rajapinta valmiissa ratkaisussa. Esimerkkinä selaimessa sivusto, joka käyttää Google Maps APIa. Sivustolla ei ole suoraa pääsyä käyttäjän sijaintiin, ja sen saamiseksi, tulisi ohjelmistokehittäjien kehittää ensin omat karttatiedot ja ohjelmoida ratkaisu, joka selvittäisi käyttäjän sijainnin. Sen sijaan käyttämällä Googlen API:a voi ohjelmisto vain kysyä selaimessa käyttäjän sijaintia, koska API:lla on pääsy näihin tietoihin. Käyttäjä voi sallia tai kieltäytyä luovuttamasta sijaintitietojaan. [2, 6]

Rajapintojen tietoturva asioihin on muodostunut OAuth-standardi. OAuth-standardi muodostettiin rajapintoihin, jotta lisättäisiin esimerkiksi käyttäjien tietoturvaa. Käyttäjän tunnistautuessa sivustolle, voidaan käyttää perinteistä SSL-salausprotokollaa. Käyttäjän tiedot salataan tiedonsiirtoon, ja niitä verrataan palvelimella tallessa oleviin tietoihin. Tietojen vastatessa tallessa oleviin tunnistautumistietoihin sallitaan tunnistautuminen. Haittapuolena käyttäjä lähettää samalla tunnistautumistietonsa myös palvelimelle. OAuth-standardilla API:n lähettäessä tunnistautumispyynnön API ei lähetä käyttäjän tunnistautumistietoja, vaan tekee todennusprosessin API:n kautta. Näin kolmannen osapuolen sivusto ei saa käyttäjän tietoja, vaan todennuksen hoitaa API:n hallitsija. OAuth antaa sivustolle pääsyn vain osaan tiettyihin tietoihin, kuten nimeen. Jos API on avoin ja sitä aletaan käyttää väärin, voi API:n julkaisija estää väärinkäytön epäämällä käyttäjältä käyttöoikeuden. OAuth-standardilla on julkaisijan helppo paikantaa ja hallinnoida API:n käyttöä, ja evätä käyttöoikeuksien antamista väärinkäyttäjille. OAuth standardia käytetään myös joissakin yrityksissä. Ratkaisussa käytetään yhtä sovellusta, johon yrityksen työntekijä kirjautuu. Kaikki muut sovellukset saavat tunnistautumistiedon sovelluksen API:n

kautta. Jos esimerkiksi työntekijän puhelin tai tietokone katoaa, eivät salasanat tällöin vuoda mihinkään, vaan yksinkertaisesti API:n tunnistetiedot evätään.[7]

OAuth-standardoitu API:n käyttö mahdollistaa myös yritysten enemmän integroida toisten yritysten palveluita, ja tehdä yhteistyötä keskenään. API:n käyttö yritysten välisessä yhteistyössä tulee olla turvallista, jotta tietoturva säilyy mahdollisimman hyvänä. Yritykset voivat esimerkiksi OAuth-standardilla todentaa kirjautumisen yhteistyökumppanin palveluun yksinkertaisesti heidän sivustonsa kautta. Turvallinen ja helppo kehitysympäristö mahdollistavat API:en innovaatioivaa käyttöä, ja lisätä huomattavasti yritystuloja kumpaankin suuntaa. Jokainen yhteistyömahdollisuus kasvattaa yrityksen näkyvyyttä ja brändiä. Innovoiva yhteistyö yritysten välillä voi johtaa uusien kehitysideoin tuloon täysin yrityksen kehitystiimin ulkopuolelta. [2, 5, 6]

2.3 Rajapinta-protokollat

API:en kehittyessä verkko-API tyyppisiksi ratkaisueiksi, ollaan monia ponnistuksia ja kehityksiä tehty tarkoituksena helpottaa kehitystyötä ja tehdä kehityksestä enemmän hyödyllistä. Isoimmat verkko-API:en standardityypit on: SOAP(Simple Object Access Protocol) ja REST(Representational State Transfer).[7]

SOAP on API-standardi, jonka tehtävä on standardisoida API:en tiedonvaihto. SOAP-standardilla suunnitellut API:t käyttävät viestintämuotona XML-tiedostoja ja vastaanottavat pyyntöjä http:n tai SMTP:n kautta. SOAP-standardi edesauttaa applikaatioiden sulavan toiminnan eri ympäristöissä, vaikka ne olisi koodattu eri koodikielellä. SOAP:n ollessa protokolla, se määrittää toimintaohjeet API:lle ja lisää monimutkaisuutta, jotka voivat johtaa API:n käytön hitauteen. Standardin noudatettavat toimintaohjeet sisältävät turvallisuus käytänteet ja ACID (atomicity, consistency, isolation, and durability) käytänteet, joka on joukko ominaisuuksia, joilla varmistetaan varma ja luotettavatietokantojen vuorovaikutus. SOAP:n ollessa protokolla, sen alla kehitettävien API:hin on monta sääntöä ja ohjetta. Tarkoin määritellyt toimintaohjeet ja toiminnot tekevät SOAP-standardin API:sta hitaampia ja jäykempiä kuin REST-tyypin API:sta.

SOAP-API:lle lähetetyt pyynnöt voidaan lähettää missä tahansa sovelluserroksessa. Kuten esimerkiksi HTTP-, SMTP- tai TCP- muodossa. SOAP-API:n vastaanottaessa pyynnön esimerkiksi HTTP-muodossa, se käsittelee sen ja palauttaa aina vastauksen

XML-dokumenttimuodossa, joka mahdollistaa helpon ohjelmistojen välisen kommunikoinnin ohjelmointikielestä huolimatta.

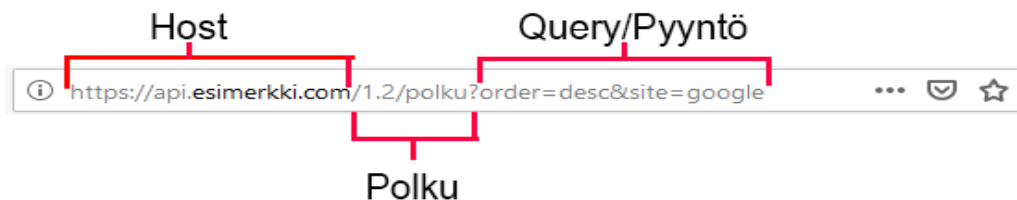
SOAP:n ollessa standardi API:lle REST on taas arkkitehtuurinen tyyli API:lle. REST-tyyppistä API:a käytetään, kun API:n käyttö tulee olla kevyttä ja sujuvaa esimerkiksi web-sivustoilla tai mobiiliapplikaatiolla. Koska REST ei ole standardi vaan joukko ohjeita, jää ohjeiden käyttöön ottaminen API:n kehittäjille. REST-API:lle lähetetyt pyynnöt tehdään yleensä HTTP-muodossa. REST-API voi palauttaa lähetetyt pyynnöt monessa muodossa, kuten HTML, XML, tekstitiedostoina tai JSON-objekteina. JSON-muotoa suositetaan, koska objekteja voidaan lukea ja käyttää millä tahansa ohjelmointikielellä. Tämän lisäksi JSON-objektit ovat kevyitä käyttää ja kehittää, mikä mahdollistaa helpon ja kevyen käytettävyyden. Koska SOAP palauttaa vain XML-tiedostoja, on REST tehokkaampi [7, 8]

API:a, joka on REST:n tyyppinen kutsutaan RESTful-tyyppiseksi. API on RESTful, jos se seuraa kuutta arkkitehtuurityyliä. Ensimmäinen on asiakas-palvelin arkkitehtuuri, jolloin API:n arkkitehtuuri koostuu asiakkaista, palvelimista ja resursseista. Kaikki arkkitehtuurissa olevat osat kommunikoivat keskenään. Toinen on tilaton asiakas-palvelin kommunikaatio. Tilaton kommunikaatio ei säilytä asiakkaan tietoja palvelimella pyyntöjen välissä. Session tietoja ja tilastoja säilytetään sen sijaan asiakkaan päässä. Kolmas on välimuistillinen data, eliminoimaan asiakas-palvelin vuorovaikutuksia. Neljäs on yhdenmukainen käyttöliittymä komponenttien välissä. Näin tiedonvälitys on standardoidussa muodossa, sen tarvitsematta olla tarkka sovelluksen tarpeille. Viides on tasoitettu järjestelmärajana, jossa asiakas-palvelin vuorovaikutus voidaan välittää hierarkkisilla tasoilla. Kuudes on koodi tarvittaessa. Ainut vaihtoehtoinen ohjetyyli. Tämä tarkoittaa, että palvelimien on mahdollista lisätä uusia, tai muokata valmiina olevia resursseja. Tämä toteutuu palvelimen lähettämällä toteutuskelpoisia koodin osia asiakkaalle. [7, 8]

Monet vanhemmat API ratkaisut käyttävät SOAP:a, REST:n ollessa uudempi. RESTful API:t mielletään yleisesti kevyemmiksi ratkaisuisiksi web-pohjaisiin ratkaisuihin. REST on joukko ohjesääntöjä, jotka mahdollistavat joustavan kehitysalustan ja implementaation eri tarkoituksiin. SOAP:n ollessa protokolla, tiukasti jäsennetyillä vaatimuksilla, kuten XML-muodon viestinnässä. RESTful API:t ovat ideaali ratkaisuja esimerkiksi mobiiliapplikaatioille, tai IoT-laitteille. Kun taas SOAP:n verkko ratkaisut sisältävät sisäänrakennettua turvallisuutta ja vuorovaikutusten ohjeiden mukaisuutta, joka sopii monien yritysten tarpeisiin, tehden niistä kuitenkin raskaampia. Monet julkiset API:t ovat RESTful tyyppiä ratkaisuja juuri keveyden ja joustavamman kehitysmahdollisuuden ansiosta.

2.4 Rajapintadatan tiedonsiirto

Asiakas-palvelinkommunikaatiossa on kaksi osapuolta, palvelin ja asiakas. API:n kommunikaatiosta on hyvä tietää muutama aihe. Request(pyyntö) tehdään asiakkaalta palvelimelle. Pyyntöä voidaan pyytää dataa, välittää palvelimelle tietoja ja tapahtumia. Esimerkiksi käyttäjän aikomus kirjautua sisään tunnistautumistiedoillaan. Vastaus(response) lähetetään palvelimelta asiakkaalle reaktiona palvelimille tehdyille pyynnöille. Esimerkiksi tunnistautumispyynnön palautus onnistumisena. Palvelu (Service) on tietty tehtävä, jonka palvelin sallii käyttäjälle. Kuten esimerkiksi datan lataamisen. URL:lla on eri osia. API:n tapauksissa tärkeimmät osat näkyvät kuvassa 1.



Kuva 1. URL-palkin osat.

RESTful API:n kannalta tärkeimmät osat ovat Host(palvelin), Path(polku) ja Query(pyyntö). URL-osoitteessa nämä ovat tyypillisesti järjestyksessä host, polku, query/pyyntö (Kuva 1) Palvelin on useimmiten nimi tai IP-osoite, joka identifioi minne otetaan yhteys tietoverkossa. Polku identifioi asiakkaan hakemaa resurssia. Query on vaihtoehtoinen, jolla asetetaan lisä parametreja tarvittaessa pyynnölle. REST-API:en yleensä toimiessa web-pohjaisina API:eina, käyttävät ne HTTP-protokollaa. HTTP-pyyntö yleensä sisältää URL:n haluttuun resurssiin, HTTP-metodin määrittämään halutun toiminnon, sekä valinnaisia parametreja HTTP-otsikkomuodossa ja valinnaista dataa, joka lähetetään palvelimelle. REST-API kutsussa(call) käytetään vain osaa HTTP-metodeista. Näihin toimintoihin kuuluvat GET, jolla haetaan tarvittu resurssi. POST, jolla luodaan resurssi. DELETE, jolla poistetaan resurssi. Kun HTTP-pyyntö tehdään palvelimelle, vastaus yleensä tulee jäseneltynä tietona. Vastaus yleensä sisältää status koodin, joka on yleensä jokin numero, joka kertoo toimiko API:n kutsuminen vai sisältääkö se virheitä. Joitakin HTTP-otsikkomuotoja, joilla tarkennetaan lisä informaatiota vastauksesta. Vastaus sisältää myös dataa, jota pyydettiin.

3 TOTEUTUS

3.1 Shimmer rajapinta

3.1.1 Shimmer API:n tarkoitus ja teknisiä tietoja

Shimmer on Realtime Technologies yrityksen sensori yksikkö, joka on kehitetty mittaamaan useita fysiologisia suureita ihmisestä. Shimmer-yksiköllä voidaan muun muassa mitata sykettä, EKG:tä, EMG:tä ja GSR:ää. Shimmer-yksikköjä on erilaisia, ja opinnäytetyössä käsitelty työ tehdään EKG-yksiköllä. (Kuva 2)



Kuva 2. Shimmer GSR-yksikkö.

EKG eli elektrokardiographia on mittaus, jolla mitataan sydämen supistumista. Sydämen supistuessa EKG mittaa sitä säätelevää sähköimpulssia. EKG:tä käytetään yleisesti sydämen sähkötoiminnan mittaamiseen, sen yksinkertaisuuden ansiosta. [23] Realtime Technologies on kehittänyt C#, Java/Android, LabVIEW ja MATLAB ohjelmointiin. Java/Android API:a käytetään pääasiallisesti mobiilisovelluksiin. Niiden ollessa Rest-tyyppisiä API:ja, juuri koska mobiilityyppisten API:en tulee olla kevyitä. MATLAB on oliopainotteinen ohjelmointi ratkaisu Shimmerin datan tallentamiseen. Shimmer pystyy myös jakamaan tietoa toisille sovelluksille, jossa on valmiiksi jo integroitu Shimmer API.

Esimerkkinä iMotions-sovellus, jota käytetään opinnäytetyössä. Shimmer paritetaan koneeseen Bluetooth yhteydellä. Shimmer API on täysin avoin, jolloin kaikki halukkaat kehittäjät voivat käyttää sitä omissa ratkaisuissaan. iMotions sovellus on esimerkiksi sisällyttänyt API:n jo valmiiksi heidän sovellukseensa.

Shimmer API:lla saadaan suoratoistettua laitteen tallentamaa fysiologista dataa, kuten sykettä. API:n avulla on mahdollista käyttää kaikkia Shimmer palikoita ja niiden käytölle tärkeimpiä ominaisuuksia (Paritus, mittaus ja kalibrointi). Shimmer API:n tarkoitus on mahdollistaa Shimmerien ominaisuuksien helppo integraatio. Opinnäytetyössä käytetään iMotions-sovellusta ja MATLAB:a luomaan rajapinta kaiken välille. Mittausfrekvenssi on 128 Hz, 256 Hz 512 Hz tai 1024 Hz. Shimmer suosittelee eri mittausyksiköille eri taajuutta. EKG-yksikössä suositellaan 512 Hz:n taajuutta. Rajapintojen kannalta ei ole väliä taajuudella. Mitä isompi taajuus, sitä enemmän data tulee tallennettua tai näytettyä. Toisaalta pienempi taajuus tarkoittaa epätarkempaa dataa, koska tieto ei päivity niin usein.

3.1.2 Shimmer-rajapinnan luominen ja käyttöönotto

Shimmer rajapinnan luominen aloitetaan lataamalla Shimmerin valmistajan sivustolta mahdollisesti käyttötarkoitukseen tarvittavat ajurit, sekä kirjastot. Käyttäjän tarvitsee joidenkin ajureiden ja kirjastojen lataamiseen käyttäjätunnuksen kyseessä olevalle sivustolle. Esimerkiksi MATLAB-kirjastoon lisättävillä Shimmer-ajureille tarvitsee Mathworks tunnuksen ja lisenssin. Käyttäjän tehtyä tarvittavat tunnukset, on mahdollista ladata kirjastoja. Opinnäytetyössä käytetään MathWorks:n kirjastoa ja ajureita Shimmerin rajapintadatan kommunikaatioon.

3.2 Shimmer-rajapinta datan käsittely iMotions-sovelluksessa

3.2.1 iMotions-sovelluksen tarkoitus

iMotions-sovellus on tanskalaisen iMotions yhtiön luoma työkalu, jolla luodaan tutkimuksia fysiologisista mittauksista. Ohjelman tarkoitus on yhdistää erilaisia sensoreita, osittain suoraan sovellukseen integroiduilla rajapinnoilla, sekä antaen kehittäjille avoimet

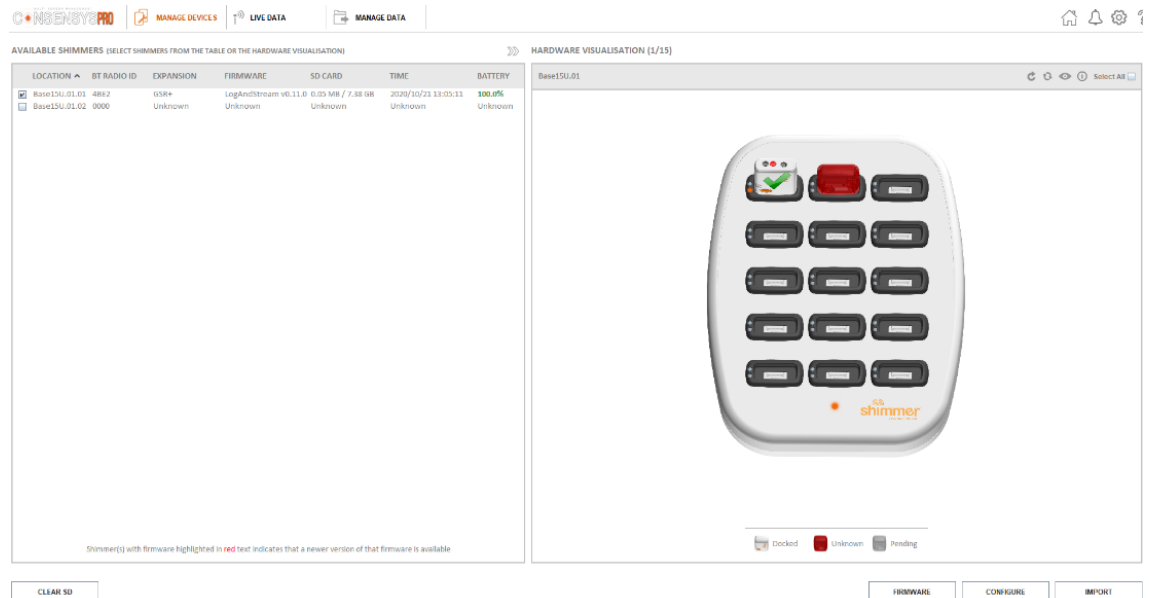
kädet yhdistävät omien sovellusten rajapintoja iMotionsiin. Tutkimuksiin voidaan yhdistää esimerkiksi Shimmerin eri palikoita, jotka mittaavat esimerkiksi GSR:ää EKG:tä ja EMG:tä. Samaan tutkimukseen voidaan lisätä esimerkiksi katseenseuranta. iMotions koottaa yhteen kaikkien tutkimuksessa olevien sensorien datan, ja tarvittaessa rajapintojen avulla voi jakaa niitä edelleen. Tutkimuksessa iMotions-sovellusta käytetään luomaan jo sovellukseen valmiina sisälletyn Shimmerin rajapinnan avulla lukemaan EKG-dataa, ja uudelleen jatkamaan sitä Matlab sovellukseen. [12, 13,14]

iMotions-sovelluksessa on mahdollista tallentaa yhteenvetovideo testitilanteesta, mikäli se nauhoitettiin. Sovelluksessa on mahdollista editoida, lisätä videoon erilaisia kuvaajia synkronoituna tilanteeseen. Sovelluksessa on mahdollista luoda erilaisia analyyseja testitilanteista. Esimerkiksi katseenseurannan dataa voidaan analysoida, ja luoda lämpökarttoja katseen pääpisteestä. Sovelluksesta voidaan myös viedä sensorien raakadataa tiedostoina analysoitavaksi toisiin sovelluksiin. Sovelluksessa on myös mahdollista rajapintojen avulla hyödyntää tulevaa dataa, ja vietävää dataa. Opinnäytetyössä hyödynnetään sovelluksen valmiina olevia rajapintoja Shimmerin tapauksessa, ja luodaan rajapinta Matlab-sovellukseen Shimmerin EKG-datan käyttöön.

Pääasiallisesti iMotions-sovelluksessa tehtävät testaukset tehdään luomalla käyttäjälle ärsykeitä(stimuli). Käyttäjän reagoidessa ärsykkeisiin sensorit saavat dataa. Opinnäytetyössä keskitytään kuitenkin rajapintojen väliseen kommunikaatioon, jolloin ärsykkeiden sisällyttäminen työhön ei ole olennaista, vaan se tapahtuu ilman ärsykeitä. Dataa on mahdollista kerätä ilman ärsykeitä, kuitenkin työn lopputarkoitus on mahdollistaa rajapintadatan hyötykäyttö tulevaisuuden tutkimuksissa.[13, 14]

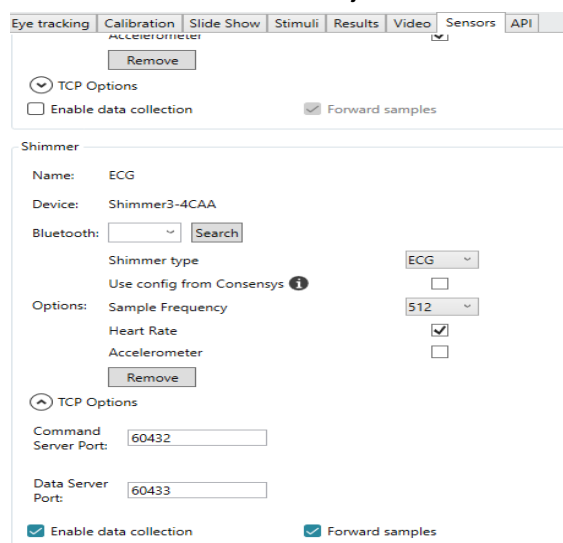
3.2.2 Shimmerin rajapinta yhteyden luonti iMotions-sovellukseen

Shimmerin käyttö aloitetaan muokkaamalla valittu Shimmer-yksikkö asemasta ja asettamalla se oikean tyyppiseksi. Shimmerin päänäkymässä on telakka, jossa näkyy kytketyt yksiköt. Valitaan oikea yksikkö ja oikeat asetukset. [Kuva 3] Shimmer-yksikköjä on erilaisia, joten kaikilla yksiköillä ei voi mitata kaikkea. Tutkimuksessa käytetään EKG-Shimmer yksikköä, joka ohjelmoidaan EKG-asetuksille ensin Consensys-ohjelmassa. Consensys on ohjelma, jolla voidaan ohjelmoida ja toistaa Shimmer-yksiköiden tallentamaa dataa ja hallita sitä. Työssä Consensys ei kuitenkaan ole olennainen muuhun, kuin Shimmer-yksiköiden ohjelmointiin.



Kuva 3. Consensys-ohjelman päänäkymä, jossa valittuna Shimmer-yksikkö ohjelmoitavaksi.

EKG-Shimmerin ollessa valmis, lisätään Shimmer seuraavaksi iMotions-sovellukseen. Shimmer-yksikkö kommunikoi sovellukselle Bluetooth yhteydellä. Shimmer hyödyntää iMotions-sovellukseen valmiiksi sisällytettyä rajapintaa. Shimmerin lisääminen tehdään iMotions sovelluksessa kohdasta Preferences-Sensor Settings. Shimmeriin tehdään seuraavat asetukset EKG-datan tallentamiseksi ja edelleen vietäväksi. [Kuva4]



Kuva 4. EKG-Shimmer asetukset iMotions sovelluksessa.

Shimmeriin asetetaan tapauksessa EKG-sensoryyppi vastaamaan ohjelmoitua Shimmeriä. Tarvittaessa käytetään vaihtoehtoa "Use Config from Consensys", jolloin iMotions saa laitteen asetukset Consensys-ohjelmasta, jossa laite konfiguroidaan. Sample Frequency(näytetaajuus) arvoksi valitaan hyvä arvo haluttuun tarkoitukseen, 128Hz, 256Hz 512Hz tai 1024Hz. EKG-tarkoitukseen valitaan arvoksi 512Hz. Näytetaajuus määrittää hertzeinä mitattuna, kuinka tiheästi näytteitä otetaan. Heart Rate(syke) tai Accelerometer(kiihtyvyyssanturi) ovat valinnaisia vaihtoehtoja. TCP-portti asetukset tulevat yleensä automaattisesti. Käyttäjä voi kuitenkin halutessaan asettaa ne halutulle arvolle. Asetuksissa Command Server Port on 60432 ja Data Server Port 60433. Rajapinnan käyttöä datan ulosvientiin tulee vaihtoehdot "Enable data collection" ja "Forward Samples" olla valittuna. Data collection kerää talteen Shimmeriin tallentamaa dataa ja Forward samples mahdollistaa datan edelleen viennin. Tämän jälkeen Shimmer suoratoistaa mitattua dataa iMotions-sovellukseen, jossa voidaan tarkkailla, tallentaa ja edelleenviedä dataa. Shimmerin raakadataa voidaan myös tulostaa manuaalisesti .csv muodossa. [Taulukko1] [Kuva 4]

Taulukko 1. iMotions-sovelluksesta tallennettu EKG:n data

Timestamp	ECG LA-RA RAW	Heart Rate ECG LL-RA ALG
1197.2826	-100827	76.535211267605632
1229.272825	-100730	76.535211267605632
1231.22595	-101803	76.535211267605632
1233.179075	-103669	76.535211267605632

Taulukossa näkyy esimerkkietoina Timestamp, ECG LA-RA RAW sekä Heart Rate ECG LL-RA ALG. Timestamp näkyy tehdyn mittauksen aikaleiman. ECG LA-RA näyttää raakadatan käsien väliltä mitatuista EKG:n arvoista. Heart Rate ECG näyttää sydämen sykkeen mitattuna.

3.3 Rajapinnan luominen MATLAB-ympäristöön.

3.3.1 MATLAB-ympäristön esittely

MATLAB on MathWorks-yhtiön kehittämä ohjelma erilaiseen tiedon laskentaan ja datan käsittelyyn. MathWorks on kehittänyt myös MATLAB:ssa käytettävän ohjelmointikielen. MATLAB-sovellusta käytetään algoritmien toteuttamiseen ja ajamiseen, datan käsittelyyn ja visualisointiin sekä esimerkiksi vuorovaikuttamaan rajapintojen avulla muiden sovellusten kanssa. MATLAB:a käytetään pääasiassa opetus ja tutkimustarkoituksissa.

MATLAB:ssa tehtävät ohjelmat kehitetään toimimaan yleensä konsoliin tai ikkunaan, jossa ne suoritetaan. Vaihtoehtoisena kehitystapana on suorittaa MATLAB-ohjelmat .m-tiedostoissa. Tiedostoista tai ikkunoista on mahdollista saada tietoa useissa eri muodoissa. MATLAB:n suosittuja tiedostomuotoja ovat muun muassa HTML, CSV, TXT, XML tai JSON-objekteina. MATLAB:a voidaan käyttää myös oliomuotoiseen ohjelmointiin. Sovellusta voidaan myös käyttää rajapintojen avulla analysoimaan ulkopuolelta tulevaa dataa ja edelleen lähettää sitä. MATLAB- tallentaa ja analysoi Shimmer dataa. Josta se edelleen vie iMotions sovellukseen. iMotions tukee ulkopuolisten API:en sisällyttämistä tutkimuksiin. iMotions saa dataa MATLAB:lta, joka analysoi Shimmerin mittaamaa dataa..[16, 17, 18]

3.3.2 MATLAB-ympäristön luonti

MATLAB ympäristön luonti alkaa lataamalla tarvittavat ajurit ja kirjastot MATLAB-sovellukseen. Shimmer on yhdistettynä tietokoneeseen Bluetooth yhteydellä. Bluetooth yhteyden COM-portti tulee selvittää. Tässä tapauksessa Shimmer viestii tietokoneelle COM4-portin kautta. MATLABin työskentely tilaksi asetetaan Shimmerin esimerkki kirjasto. Yksinkertaisen testauksena käytetään Shimmer-yhteyden luomiseen funktiota `plotandwriteecgexample()`. Tämä kutsuu valmista skriptiä, joka mittaa x-ajan ja kirjoittaa nimettyyn tekstitiedostoon mitatun datan. Skripti näyttää myös mittaustulokset reaaliaikaisesti. Esimerkki skriptin käytöstä: `plotandwriteecgexample ('4', '10', 'shimmertestdata.dat')`, jossa ensimmäinen arvo(4) määrittää COM-portin jota Shimmer käyttää viestintään. Toinen arvo(10) määrittää mittaussajan pituuden sekunneissa, ja viimeisenä tiedoston nimi johon data tallennetaan. Funktio asettaa Shimmerin näytteenottotaajuudeksi 512Hz. Testitiedoston muotoa voi myös komennossa päättää.

Testauksessa kokeiltiin esimerkki komentoa .dat .xml sekä .csv tiedostomuodossa. Dat-
muodossa, data on manuaalisesti luettavissa, sekä hyödynnettävissä eri funktioilla MAT-
LAB:ssa. Sydämen sykettä säätelevien sähköimpulssien ollessa luontaisesti pieniä, vas-
taavat funktion tallentamat mittausravotkin pieniä arvoja. [Taulukko 2]

Taulukko 2. MATLABin funktion tallentama EKG:n data.

Time Stamp	ECG LL-RA	ECG LA-RA	ECG RESP	ECG Vx-RL
CAL	CAL	CAL	CAL	CAL
milliseconds	millivolts *	millivolts *	millivolts *	millivolts *
284611.267089 8438	2.01832378129 0505	- 1.63817067601 3312	- 1.73856398326 9213	- 48.9824138858 8117
284613.220214 8438	1.84927068344 0052	- 1.62569363423 51	- 1.71324929156 891	- 49.0388851212 1262
284615.173339 8438	1.86946473949 7273	- 1.64357979817 1496	- 1.72197600865 0781	- 49.0613149477 6189

MATLAB:ssa luotiin esimerkkifunktio, jolla testataan datan vientiä iMotionsiin. Funktio on yksinkertainen, jolla lähetetään yhden hertzin signaali käyttäen porttia 8089. Kun funktio ajetaan MATLAB-sovelluksessa, saa iMotions signaalin portista 8089. Data näkyy iMotions sensorien välilehdessä yhden hertzin voimakkuudelle, jolloin data liikkuu onnistu-
neesti sovellusten välillä.

```
function [] = testpnet ()

str=sprintf('E;1;VoiceVolume;1;0.0;;;VoiceVolumeSample;1.0\r\n')

u=udp('127.0.0.1',8089);

for i = 1:100

    fopen(u);

    fwrite(u,str);

    fclose(u);

    pause(0.5);

end
```

Testidatan onnistunut edelleen vienti rajapintoja käyttämällä käytetään pohjana luomaan Shimmerin mittaaman datan vientiä iMotionsiin.

3.3.3 Shimmer-rajapinnan luonti MATLAB-ympäristöön

Shimmer rajapinta luotiin MATLAB ympäristöön käyttämällä Mathworksin kirjastoja. Esimerkki koodissa EKG-käyrä tulostuu MATLAB:n eri käyriin. Funktion saa kirjoittamaan dataa halutuissa muodoissa. iMotions sovellusta varten valitaan muodoksi xml. Valmiiseen esimerkkiin kirjoitetaan lisää koodia, jolla avataan yhteys ja muotillaan data oikeanlaiseksi. (Kuva 5)

```

u=udp('127.0.0.1',8089); %Choosing where to export the data. iMotions as default, using port 8089.
fopen(u); %Opening the connection to the path.

line = ""; %Declaring line as empty
for row = 1 : 7 % Formatting the data to a format readable by iMotions XML config.
    line = "q;";
    for column = 1 : 5
        line = append(line, num2str(newData(row,column)));

        if column ~= 5
            line = append(line, ",");
        end
    end
    fwrite(u,line);
    disp(line);
    line = "";
end

```

Kuva 5. Lisätty koodi, jolla dataa lähetetään iMotions-sovellukseen.

MATLAB-sovelluksessa luodussa koodissa määritellään ensin mihin yhteys avataan. Osoite on 127.0.0.1, sekä portti on 8089. Näitä käytetään lähettämään Shimmerien mittausdata oikeaan paikkaan, jota iMotions sovellus on asetettu kuuntelemaan. Tämän jälkeen luodaan for-loop, jossa data muutetaan oikeanlaiseksi, jotta iMotions pystyy sitä tulkitsemaan. Datalle on asetetta tietty etuosa, ja sen tulee olla jäsennelty puolipilkulla (;). Näin iMotions huomaa mittausdatasta halutut arvot, ja osaa erotella ne. For-loop käy läpi datan ja muotoilee sen oikean tyyppiseksi. Tämän jälkeen oikein muotoiltu data on luettavissa iMotions-sovelluksessa, kun sinne asetettu xml-tiedosto sen jäsentelee. iMotions sovellukseen tulee asettaa erikseen vielä XML-tiedosto, joka toimii MATLAB:sta tulevan datan muotoiluna. XML-tiedostossa määritellään, miten data näytetään. Datan visuaalisille kentille asetetaan esimerkiksi nimet, määritellään ylä- ja ala-arvot. Tiedostossa asetetaan myös tunnistettava mittausarvo, tässä tapauksessa ECGshimmer. ECGshimmer on koodissa valittu etuliite lähetettävään pakettiin, jotta iMotions osaa tunnistaa paketin alun ja lopun. Näin iMotions tunnistaa mittausdatan, ja alkaa visualisoida sitä. Lopputuloksena on toimiva rajapinta MATLAB-sovelluksesta iMotions-sovellukseen. Valmis funktio välittää mittausdataa Shimmeristä MATLAB:n kautta iMotions:iin.

Muotoiltu data voidaan lukea oikeanlaisen XML-tiedoston kanssa iMotions:issa. Event-source tiedoston toiminto on toimia tyyliohjeena, jossa määritellään esimerkiksi mittausdatan ylä- ja ala arvot. Tämän lisäksi siinä määritellään mittaussarakkeiden nimet. iMotionsiin luotiin seuraavanlainen XML-tiedosto:

```

1 <EventSource Version="1" Id="ShimmerECG" Name="ShimmerECG">
2
3   <Sample Id="ShimmerECG" Name="ShimmerECG">
4
5     <Field Id="ShimmerECG" Range="Variable" Min="0" Max="20000000"></Field>
6
7     <Field Id="ECG LL-RA" Range="Variable" Min="-20" Max="0.2"></Field>
8
9     <Field Id="ECG LA-RA" Range="Variable" Min="0" Max="10"></Field>
10
11    <Field Id="ECG RESP" Range="Variable" Min="0" Max="5"></Field>
12
13    <Field Id="ECG Vx-RL" Range="Variable" Min="-200" Max="0"></Field>
14
15  </Sample>
16
17 </EventSource>

```

Kuva 6. iMotions-sovellukseen luotu tyyliohje saapuvalla datalle.

Ensin määritellään versio malli ja asetetaan tunnistetiedoksi(Id) sopiva arvo datasta. Sample osiossa määritellään saapuvassa datassa esiintyvän paketin alkuosa. Tässä tapauksessa arvoksi asetettiin ShimmerECG, koska MATLAB:ssa on asetettu sama tunniste jokaiselle saapuvan datapaketin arvoksi. Field kohdissa asetetaan jokaiselle paketin arvolle nimi, ja voidaan osoittaa yläarvo ja ala arvo. Tämä suodattaa yli- ja aliarvot mittausarvoista ja asettaa mittauskäyrälle ylä- ja alakohdat. Arvot voidaan antaa kokonaislukuina, tai desimaalimuodoissa. Tärkeää on MATLAB:n koodissa asettaa puolipilkku(;) jakamaan saapuvat paketit.[Kuva 6]

4 YHTEENVETO

Opinnäytetyön tarkoitus oli esimerkillistä iMotionsin rajapintojen toimivuutta, ja luoda rajapinta iMotionsiin. Rajapinnan tarkoitus oli mitata Shimmerin dataa MATLAB-sovelluksessa, jossa se myös visualisoidaan vertailun vuoksi. Samanaikaisesti mitattu data muutetaan oikeanlaiseen muotoon, jotta se olisi luettavissa iMotions-sovelluksessa. iMotionsiin luotiin lähteeksi sensori, joka kuuntelee porttia. 8089. API-lähde kuuntelee iMotions:n ollessa päällä saapuvia paketteja. iMotionsiin tehtiin XML-tiedosto, joka muodostaa datan visualisoinnin. MATLAB:iin tehtiin funktio, joka ensin mittaa käyttäjistä ekg-dataa Shimmerillä. Tämän jälkeen funktio näyttää datan MATLAB:n kuvaajissa ja avaa yhteyden iMotions-sovellukseen. Avattuaan yhteyden funktio muuttaa datan, kuten on mainittu. Opinnäyte toi haasteita ja ongelmia, joista lisää kohdassa 4.1.

Opinnäytetyö tehtiin tiukalla aikataulutuksella lokakuun aikana. Vaikka varsinaista koodaamista ei ollut paljoa, nousivat ongelmakohdiksi minulle osittain käytettävien sovellusten tuntemattomuus. MATLAB on sovelluksena itselleni hieman vieras, vaikkakin sitä ennen opinnäytetyötä olin käyttänytkin. Opinnäytetyö vaati paljon tutustumista MATLAB:n syntaksiin, sekä ohjelman yhteensopivuusasioiden selvittämiseen ja ratkomiseen. Positiivisena puolena opin lisää rajapinnoista, ja niiden periaatteista. Opin paljon toiminoista, joita en aiemmin ollut käyttänyt. Esimerkiksi iMotions-sovelluksen rajapintojen muodostaminen oli minulle tuntematon aihe, ennen opinnäytetyön aloitusta.

Opinnäytetyöstä on hyötyä hyötyä Health Tech Labin jatkokehityksessä. Luotua rajapinta esimerkkiä on helppo jatkokehittää, ja sen piiriin voi luoda uusia projekteja. Rajapinta-esimerkin luonti oli hyödyllistä laboratorion ympäristöön, koska selvää esimerkkiä juuri samankaltaisista mittausasioista ei ole olemassa. Vaikkakin se on erityisen supistettu esimerkki, uskon laboratorion monilla Shimmereillä ja etenkin iMotions-sovelluksella olevan selvää kiinnostusta ja jatkokehitystä.

4.1 Kohdatut ongelmat

Opinnäytetyö ei sujunut ongelmitta. Isoimmaksi ongelmaksi nousi lukuisat toimimattomuudet. Shimmer toimii moitteetta yksittäisen mittausvälineenä sen omassa sovelluksessa sekä iMotions sovelluksessa. Kuitenkin MATLAB-sovelluksessa nousevat esiin sen yhteysongelmat. MATLAB antoi useasti virheitä, koska ei pystynyt avaamaan tiettyä

sarjaporttia (COM4). Virhettä yritettiin paikallistaa pakottamalla COM4-portti auki sovelluksessa sekä mahdollisesti paikantamalla, käyttäkö jokin muu sovellus Shimmeriä. Ongelma tuli esiin usein, ja parhaiten siitä päästiin eroon poistamalla laitehallinasta COM4 portti ja Shimmer. Tämän jälkeen se asennettiin uudestaan, jolloin virhe lähti. Tämä saatettiin joutua toistamaan monia kertoja peräkkäin. Toinen Shimmeristä peräinen ongelma oli aikaviiveestä johtuva yhteyden katkaisu Shimmeriin MATLAB-sovelluksessa. Täytä ongelmaa helpotti, jos Shimmeriin kirjoitettiin uudet asetukset Consensus-sovelluksessa. Tämäkään ei välillä auttanut, joten virhe jäi selvittämättä sen esiintyessä harvemmin. Ongelmakohtaksi jäi myös datan näkyvyys iMotions sovelluksessa. Vaikka saapuvassa datapaketissa on selkeästi määritelty etuosa ja loppuosa, ei iMotions näytä saapuvaa dataa sen graafisessa käyrässä. Kuitenkin samassa koodissa testattu yhden Hertzin testi signaali saadaan näkymään ongelmitta. Uskoisin ratkaisun olevan XML-tiedoston hienosäätö, tai mahdollisesti saapuvan datan tarkempi muotoilu.

Ongelmakohtaksi jäi myös kokemattomuus MATLAB-sovelluksen kanssa valmiin ulkoasun luomiseen. Käyttäjä ystävällisyys jää puuttumaan, sillä MATLAB sovelluksessa tulee ajaa komento funktion toimiakseen. Funktion tulee olla päällä sen toimiakseen kunnolla. Funktioon on asetettu tietty aika-arvo, jonka puitteissa funktio on päällä ja mittaa tietoa. MATLAB:n ollessa vielä hieman tuntemattomampi työkalu, jäi tietynlainen toimivuuden kannalta hyödyllinen automatisaatio tekemättä. Tiettyyn pisteeseen asti tehty automatisointi helpottaisi käyttäjä ystävällisyyttä. Ongelmana oli myös saatavan materiaalin vähäisyys. Saatavilla olevien esimerkkien ollessa hyödyllisiä, ei ole olemassa ainaakaan toistaiseksi vaativampia esimerkkejä. Ohjelmoinnissa joutuu soveltamaan toivottua enemmän. MATLAB:ssa kohdattuihin ongelmiin kuitenkin ratkaisuna voidaan pitää tarvittavien ohjelmointitaitojen puuttuminen ja itselle tuntematon toimintaympäristö.

4.2 Jatkokehitys

Jatkokehitysideoiksi jäi useita asioita. Monet asiat pystytään jatkokehityksessä korjaamaan ja yksinkertamaan. Ensimmäisenä ja suurimpana tulisi data muotoilla muotoon, jossa iMotions-sovellus osaisi sitä lukea ja muodostaa mittauskäyrän. Tämä vaatisi tarkempaa tutustumista iMotions:n tarvittaviin parametreihin, joita viestin tai XML-tiedoston sisältää. Jatkokehityksessä olisi hyvä huomioida asioiden yksinkertaistaminen ja esimerkkien teko. Vaikka selkeitä esimerkkejä löytyy, ne ovat erittäin yksinkertaisia, jolloin monimutkaisimmissa ratkaisuissa joutuu paljon soveltamaan.

Toisena jatkokehitys ideana voitaisiin pitää käyttöliittymän luomista ja aikarajoitteen poistamista tapauskohtaisesti harkita. Koska funktio toimii aikarajoitteen puitteessa, tulee pitkiin mittauksiin tällä hetkellä asettaa korkeita arvoja. Funktiota voidaan myös yksinkertaistaa, sillä mittaustulosten näkyvyys samanaikaisesti MATLAB:ssa ja iMotions:ssa ei ole hyödyllistä kuin kehitysvaiheessa. Samanaikaisesti näkyvällä datalla voidaan varmistaa kummankin sovelluksen toimivan ja datan olevan vertailtavissa. Kuitenkin mittaustilanteissa tällä ei ole varsinaista hyötyä. Käyttöliittymän luominen loisi paljon enemmän käyttäjäystävällisyyttä, ja automatisaatio olisi toimivampi ratkaisu. Tämän hetken kahden sovelluksen välillä toimiminen ei ole käyttäjälle sujuvin vaihtoehto.

Hienosäätönä voitaisiin data myös formatoida tarkemmin. Jatkokehityksessä voitaisiin esimerkiksi koodata funktio, joka muuttaisi EKG:n mittausdatan sykkeeksi, ja lähettäisi iMotions sovellukselle sykkeen yhtenä arvona. Käyttöliittymään voitaisiin esimerkiksi asettaa eri toimivuuksia, kuten EKG:n raakadata, EKG-funktion, josta saadaan syke, tai eri Mittaus-Shimmereitä. Funktion tallentamaa datan määrää voitaisiin myös jatkokehittää vastaamaan kunkin tilanteiden tarpeisiin. Tällöin dataa ei tarvitsisi tallentaa niin isoja määriä. Esimerkistä voitaisiin tehdä vaihtoehto, joka ei tallenna dataa, vaan esimerkiksi pelkästään lähettäisi sitä. Tällä vaihtoehdolla olisi mahdollista saada tallennettua dataa iMotions:n kautta tarvittaessa siellä tehtävän tutkimuksen yhteyteen. Käyttäjällä voisi myös olla valintavaihtoehtona esimerkiksi Shimmerien määrä, ja asettaa arvot, joita Shimmerit mittaisivat. Jatkokehityksessä Shimmereitä yhdistellessä tulee huomioida kuitenkin XML-tiedoston muokkatarpeen, jokaista mittaustilannetta vastaavaksi. Näitä voitaisiin tehdä etukäteen, esimerkiksi yksi EKG:lle ja toinen GSR:lle.

LÄHTEET

- [1] Mulesoft 2020. What is and API? (Application Programming interface). Viitattu 01.10.2020 <https://www.mulesoft.com/resources/api/what-is-an-api>
- [2] Avoin rajapinta 2014. Viitattu 01.10.2020. <http://avoinrajapinta.fi/>
- [3] What is an API? .Viitattu 01.10.2020 <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>
- [4] Chris Hoffman. 2018 What is an API? Viitattu 01.10.2020 <https://www.howtogeek.com/343877/what-is-an-api/>
- [5] Margaret Rouse 2019 API security. Viitattu 05.10.2020. <https://searcharchitecture.techtarget.com/definition/API-security>
- [6] Mozilla developers MDN 2020 Introduction to web APIs. Viitattu 05.10.2020 https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction
- [7] Spasovski, Martin, Packt Publishing 2013: OAuth 2.0 identity and access management patterns. Viitattu 05.10.2020 https://turkuamk.finna.fi/Record/turkuamk_electronic.994159317205970
- [8] REST vs. Soap RedHat. n.d. viitattu 06.10.2020 <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest>
- [9] SmartBear 02.01.2020 What's the Difference? Viitattu 07.10.2020 SOAP vs REST. <https://smartbear.com/blog/test-and-monitor/soap-vs-rest-whats-the-difference/>
- [10] Shubhang Dixit. 30.01.2019. Beginners Guide to Client Server Communication. Viitattu 08.10.2020 <https://medium.com/@subhangdxt/beginners-guide-to-client-server-communication-8099cf0ac3af>
- [11] Shimmer RealtimeTechnologies 2018 Shimmer GSR+ manual. Viitattu 19.10.2020 http://www.shimmersensing.com/images/uploads/docs/GSR_User_Guide_rev1.13.pdf
- [12] Galvanic Skin Response (GSR). Viitattu 20.10.2020 <https://www.brainsigns.com/en/science/s2/technologies/gsr>
- [13] iMotions platform 2020. Viitattu 21.10.2020 <https://imotions.com/platform/>
- [14] iMotions 2020. Viitattu 22.10.2020 <https://imotions.com/>
- [15] iMotions iMotions 8.1 2020 Sovellus. Viitattu 22.10.2020
- [16] Wikipedia 2020. MATLAB. Viitattu 22.10.2020 <https://fi.wikipedia.org/wiki/MATLAB>
- [17] Wikipedia 2020. MATLAB. Viitattu 22.10.2020 <https://en.wikipedia.org/wiki/MATLAB>
- [18] MathWorks 2020. Supported File Formats for Import and Export Viitattu 22.10.2020. https://se.mathworks.com/help/matlab/import_export/supported-file-formats.html
- [19] Turun Ammattikorkeakoulu. 2020 Terveysteknologia. Viitattu 23.10.2020. <https://www.turkuamk.fi/fi/tutkimus-kehitys-ja-innovaatiot/tutkimusryhmat/terveysteknologia/>
- [20] Turun Ammattikorkeakoulu 2020. Health Tech. Viitattu 23.10.2020. <https://healthtech.turkuamk.fi/healthtechlab/>
- [21] Turun Ammattikorkeakoulu 2020. Health Tech Koulutus Viitattu 23.10.2020. <https://healthtech.turkuamk.fi/koulutus/>
- [22] Turun Ammattikorkeakoulu 2020. Health Tech Tutkimus. Viitattu 23.10.2020 <https://healthtech.turkuamk.fi/tutkimus/>

[23]Duodecim. Mustajoki Pertti. Jarmo Kaukua. 09.07.2008. EKG (sydänfilmi) Viitattu 27.10.2020
https://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=snk03210

Kuvien lähteet Iiro Koskialho 2020.

