

Peltihamsteri-sovellusprojekti

**Mari Kasanen
Leevi Liimatainen
Marina Mustonen
Juhani Sundell
Arttu Ylä-Sahra**

Projektisuunnitelma

Versio 1.0.0

Julkinen

4.4.2019

**Jyväskylän yliopisto
Informaatioteknologian tiedekunta**

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.20__		
Tilaaaja	__.__.20__		
Ohjaaja	__.__.20__		

Tietoja dokumentista

Tekijät:

Marina Mustonen marina.s.mustonen@gmail.com 044-9733582

Dokumentin nimi: Peltihamsteri-projekti, Projektisuunnitelma

Sivumäärä: 44

Tiedosto: Peltihamsteri_Projektisuunnitelma_1.0.0.doc

Tiivistelmä: Peltihamsteri-sovellusprojekti kehittää sovelluksen Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriossa kerättävän datan synkronointiin ja hallintaan. Projektisuunnitelma kuvaa Peltihamsteri-projektin suunnitellun läpiviennin. Suunnitelmassa kuvataan projektin termistöä, tavoitteita ja tarpeita, resursseja, käytänteitä, ryhmän jäsenten vastuualueita, tehtävien työmääriä, prosessimallia ja aikataulua sekä mahdollisia projektiin vaikuttavia riskejä ja niiden hallintaa.

Avainsanat: Aikataulu, käytänteet, projektihallinta, projektiorganisaatio, projektisuunnitelma, prosessimalli, resurssit, riskit, sovellusprojekti, taustaa, tavoitteet, tehtäväjako, tehtävät, työmäärät, vastuualueet.

Muutoshistoria

Versio	Päivä	Muutokset	Tekijä
0.0.1	8.2.2019	Projektisuunnitelman laatiminen aloitettiin.	Marina Mustonen
0.0.2	10.2.2019	Kirjoitettiin taustaa ja tarpeita, hahmotettiin yleiskuva projektisuunnitelmasta.	Marina Mustonen
0.0.3	12.2.2019	Kirjoitettiin tarkennuksia taustaan.	Marina Mustonen
0.0.4	16.2.2019	Hahmoteltiin työjakoa.	Marina Mustonen
0.0.5	17.2.2019	Hiottiin työnjakoa ja kirjoitettiin riskeistä.	Marina Mustonen
0.0.6	24.2.2019	Hahmoteltiin aikataulu ja hiottiin eri kohtia.	Marina Mustonen
0.0.7	25.2.2019	Tehtiin tarkennuksia ja korjauksia eri kohtiin.	Marina Mustonen
0.0.8	26.2.2019	Kirjoitettiin toteutettavasta sovelluksesta ja sen tavoitteista ja toiminnallisuudesta.	Marina Mustonen
0.1.0	4.3.2019	Tehtiin tarkennuksia ryhmän kommenttien perusteella.	Marina Mustonen
0.1.5	12.3.2019	Tehtiin korjauksia ohjaajien kommenttien perusteella.	Marina Mustonen
0.2.0	15.3.2019	Tehtiin lisää korjauksia ohjaajien kommenttien perusteella.	Marina Mustonen
0.2.5	21.3.2019	Tehtiin korjauksia ohjaajien ja tilaajan kommenttien perusteella.	Marina Mustonen
0.3.0	26.3.2019	Hiottiin ulkoasua.	Marina Mustonen
1.0.0	4.4.2019	Hiottiin kirjoitusasua.	Marina Mustonen

Tietoja projektista

Peltihamsteri-sovellusprojekti kehittää sovelluksen Jyväskylän yliopiston kognitiivisen ajosimulaatiolaboratoriossa kerättävän datan synkronointiin ja hallintaan. Ajosimulaatiokokeessa saadaan erilaisia mittaustuloksia useammalta laitteelta ja laitteista saatava data tulee synkronoida samaan tiedostoon. Ohjelma kehitetään C#-kielellä Microsoft Windows -ympäristöön.

Tekijät:

Mari Kasanen	<code>r.mari.s.kasanen@student.jyu.fi</code>	044-2359811
Leevi Liimatainen	<code>leevi.m.m.liimatainen@student.jyu.fi</code>	050-5052731
Marina Mustonen	<code>marina.s.mustonen@gmail.com</code>	044-9733582
Juhani Sundell	<code>juhani.k.sundell@student.jyu.fi</code>	044-2666773
Arttu Ylä-Sahra	<code>arttu.e.yla-sahra@student.jyu.fi</code>	044-0668949

Tilaaajan edustajat:

Hilkka Grahm	<code>hilkka.grahn@jyu.fi</code>	040-8053342
Tuomo Kujala	<code>tuomo.kujala@jyu.fi</code>	0400-247392

Ohjaajat:

Jonne Itkonen	<code>jonne.itkonen@jyu.fi</code>	050-4432381
Jukka-Pekka Santanen	<code>santanen@jyu.fi</code>	040-8053299

Yhteystiedot:

- Sähköpostilistat peltihamsteri@korppi.fi,
peltihamsteri_opetus@korppi.fi
- Sähköpostiarkistot [https://korppi.jyu.fi/kotka/servlet/
list-archive/peltihamsteri/](https://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri/),
[https://korppi.jyu.fi/kotka/servlet/
list-archive/peltihamsteri_opetus/](https://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri_opetus/)
- Työhuone AgC 226.4, puh. 040-3573826

Sisältö

1	Johdanto	1
2	Termit	2
3	Taustaa ja tavoitteita	6
3.1	Taustaa ja tarpeita	6
3.2	Toteutettavan sovelluksen rakenne	8
3.3	Sovelluksen työnkulku ja toteutusratkaisut	10
3.4	Tulokset.....	12
3.5	Oppimistavoitteet.....	13
4	Projektiorganisaatio ja resurssit	15
4.1	Projektiorganisaatio	15
4.2	Tilat ja laitteet	16
4.3	Kehitys- ja dokumentointityökalut.....	17
4.4	Luennot ja perehdytykset	17
5	Käytänteet	19
5.1	Palaverit.....	19
5.2	Tiedotus.....	20
5.3	Tiedostojen nimeäminen	20
5.4	Hakemistorakenne	21
5.5	Lähdekoodi	21
5.6	Testaus.....	22
5.7	Versiohallinta ja -numerointi.....	22
5.8	Katselmoinnit ja tulosten hyväksyminen	23
5.9	Tulosten koostaminen ja toimittaminen	23
6	Roolit, vastualueet ja tehtävät	24
6.1	Roolit ja vastualueet	24
6.2	Tehtävät ja niiden työmäärät.....	25

7	Prosessi ja aikataulu	28
7.1	Prosessi	28
7.2	Aikataulu	30
8	Riskien hallinta	31
8.1	Riskien todennäköisyydet ja haittavaikutukset	31
8.2	Laitteiden erilaisuus ja tietojen puute	32
8.3	Sovelluksen joustavuuden toteuttamisen vaikeus	32
8.4	Ominaisuuksien toteuttamisen vaikeus	33
8.5	Ryhmän tietotaitojen puutteet	33
8.6	Ryhmän jäsenten muut sitoumukset	34
9	Yhteenveto	35
	Lähteet	36

1 Johdanto

Peltihamsteri-projekti kehittää Sovellusprojekti-kurssilla keväällä 2019 Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriolle sovelluksen tutkimuksissa kerättävän datan synkronointiin ja hallintaan. Ajolaboratoriossa kehitetään testausmenetelmiä teollisuuden ajoneuvokäyttöliittymiä varten, sekä tehdään perustutkimusta kuljettajien tarkkaavaisuudesta, visuaalisesta havainnoinnista ja liikennekäyttäytymisestä. Ajosimulaatiokokeessa saadaan dataa useasta eri laitteesta. Ajosimulaatiolaboratoriolla on tarvetta sovellukselle, jolla näistä eri laitteista saatava data saadaan synkronoitua keskenään sen käsittelyn ja analysoinnin helpottamiseksi.

Projektisuunnitelma kuvaa projektin suunniteltua läpivientä. Sen laatimisessa on hyödynnetty Monisiro-projektin projektisuunnitelmaa [1], Isäxi-projektin projektisuunnitelmaa [2] ja sovellusprojektien ohjetta [3]. Dokumentin pohjaksi on otettu Moveatis-projektin projektisuunnitelma [4]. Projektisuunnitelman lisäksi projektiryhmä laatii ohjelmistolle vaatimusmäärittelyn, arkkitehtuurikuvausten sekä järjestelmätestaussuunnitelman ja -raportin. Projektin lopuksi ryhmä laatii projektiraportin, jossa tarkastellaan projektin toteutunutta läpivientä suhteessa projektisuunnitelmaan. Ryhmä laatii myös sovellusraportin, jossa kuvataan sovelluksen toteutusratkaisuja sekä mahdollisia puutteita ja jatkokkehitysideoita. Kaikista projektin palavereista laaditaan pöytäkirjat.

Projektisuunnitelman luvussa 2 esitellään projektissa käytettäviä termejä. Luvussa 3 kuvataan projektin taustoja ja tavoitteita. Luvussa 4 esitellään projektiin osallistuvat henkilöt ja projektiryhmän resurssit. Luku 5 kuvaa projektin käytänteitä. Luku 6 kuvaa projektin jäsenten roolit, vastuualueet, tehtävät ja arvioidut työmäärät. Luvussa 7 kuvataan sovelluskehitysprosessia ja aikataulua. Luku 8 kuvaa projektin läpivientiin mahdollisesti vaikuttavia riskejä ja niiden hallintaa.

2 Termit

Luvussa kuvataan Peltihamsteri-projektiin liittyviä termejä. Dokumentissa esiintyvät aihealueen termit ovat seuraavat:

AOI	eli Area of Interest on silmänliikekameran ohjelmistoon määritettävät alueet, joista mitataan kuinka usein ja kuinka kauan koehenkilön katse kohdistuu niihin.
Asetustiedosto	(eng. <i>Study settings</i>) on tiedosto, johon tallennetaan kokeen asetukset, jolloin samoja asetuksia voi käyttää useamman eri koehenkilön kanssa.
CSV	eli Comma-Separated Values on tiedostomuoto, jolla tallennetaan yksinkertaista taulukkomuotoista tietoa tekstitiedostoon.
Datan jäsentäminen	on arvojen erottelua ja tunnistamista raakadatasta.
Katsedata	sisältää tiedon kuinka usein ja kuinka kauan koehenkilön katse kohdistuu AOI:ksi (kts. yllä) määritettyyn kohteeseen.
Koe	(eng. <i>Study</i>) on yhden koehenkilön kanssa suoritettava 1–2 h kestävä ajosimulaatiokerta, jossa koehenkilö suorittaa tehtäviä ja suorituksesta kerätään dataa.
Lua-skripti	on tilaajan käyttämä kooditiedosto ajosimulaattoridatan mittaustulosten valikointiin.

R	on ohjelmointikieli tilastolliseen laskentaan ja grafiikan muodostamiseen.
Tehtävä	(eng. <i>Task</i>) on kokeen aikana koehenkilön suorittama yksi ajosuorite.
Toissijainen tehtävä	on tehtävän aikana koehenkilön suorittama muu toiminto kuin ajaminen, esimerkiksi sovelluksen käyttäminen testilaitteella.
Tulostiedosto	on kehitettävän sovelluksen muodostama CSV-tiedosto, johon on yhdistetty ja synkronoitu valituilta laitteilta saatavat valitut dataosiot yhden koehenkilön yhdestä tehtävästä.

Dokumentissa esiintyvät kohdealueen laitteiden ja ohjelmistojen termit ovat seuraavat:

Ajosimulaattori	on laite, jolla simuloidaan oikean ajoneuvon ajamista virtuaalitodellisuudessa.
Android	on Linux-pohjainen käyttöjärjestelmä, jota käytetään yleisimmin älypuhelimissa ja tableteissa.
D-Lab	on tilaajan käyttämän silmänliikekameran ohjelmisto.
DSI-Streamer	on tilaajan käyttämän EEG-laitteen ohjelmisto.
EEG	eli elektroenkefalografia tarkoittaa aivosähkökäyrää.

EepSoft	on tilaajan käyttämän ajosimulaattorin ohjelmisto.
Ergoneers Dikablis Professional	on tilaajan käyttämä uusi silmänliikekamera.
Noldus Observer XT 12	on videopisteytys- ja datan synkronointiohjelmisto.
QStates	on EEG-laitteen kanssa käytettävä ohjelmisto, jolla laskettavia tunnuslukuja käytetään muun muassa koehenkilön kuormituksen arvioimiseen.
SEER	on tilaajan aiemmin käyttämän silmänliikekameran ja ajosimulaattorin datan synkronointiin kehitetty ohjelmisto.
Silmänliikekamera	on laite silmän aseman ja liikkeiden mittaamiseen.
VisGuard Observer	on taustasovellus, jolla kognitiotieteen tutkimusryhmä on kerännyt syötetietoja koehenkilön käyttämästä laitteesta.
Wearable Sensing DSI-24	on tilaajan EEG-laite.

Dokumentissa esiintyvät kehitysvälineisiin ja -tekniikoihin, sekä projektin hallintaan liittyvät termit ovat seuraavat:

C#	on Microsoftin .NET-alustalle kehitetty ohjelmointikieli.
Git	on hajautettu versiohallintaohjelmisto lähdekoodien ja dokumenttien hallintaan.

Hyväksymistestaus	on tilaajien suorittama testaus sovellukselle, jolla varmistetaan sovelluksen täyttävän tilaajien asettamat vaatimukset.
Katselmointi	on silmäkkäin tapahtuva tulosten tarkastaminen, jossa tarkastetaan toteutukset ja niiden erot alkuperäiseen suunnitelmaan, sekä esitetään parannusehdotuksia.
Käyttöliittymä	on ohjelman ihmiselle näkyvä osa, jonka avulla käyttäjä ja ohjelma ovat vuorovaikutuksessa keskenään.
Lähdekoodi	on tietokoneohjelman tekstimuotoinen ohjelmointikielinen listaus.
Tilakatsaus	on projektipäällikön tekemä viikottainen katsaus, jossa käydään läpi projektin eteneminen.
WPF	eli Windows Presentation Foundation on .NET-kirjasto, joka muodostaa rajapinnan ja toimii käyttöliittymän pohjana.
Xamarin	on natiivien sovellusten kehittämiseen tarkoitettu kehitystyökalu, jolla koodin saa helposti jaettua eri käyttöjärjestelmien välillä.
YouSource	on Gitorius-pohjainen paikallinen järjestelmä lähdekoodien ja materiaalien hallintaan ja julkistamiseen.

3 Taustaa ja tavoitteita

Luvussa käsitellään projektissa toteutettavalle sovellukselle, ryhmän oppimiselle ja muille tuloksille asetettuja tavoitteita.

3.1 Taustaa ja tarpeita

Projektin tilaajana toimii Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratorio. Ajosimulaatiolaboratoriossa on vuosina 2007–2019 tehty kokeellisia tutkimuksia, joihin on osallistunut yli 500 koehenkilöä. Kokeissa on keskitytty mittaamaan kuljettajan tarkkaavaisuutta ja liikennekäyttäytymistä, sekä esimerkiksi teollisuuden ajoneuvokäyttöliittymien ja muiden laitteiden käytön vaikutusta niihin. Kokeissa kerätään dataa tyypillisesti vähintään kolmelta eri laitteelta (yleensä ajosimulaattori, silmänliikekamera ja toissijaisen tehtävän laite). Uutena hankintana laboratoriolle on EEG-laite, jota on testattu, mutta ei vielä käytetty ajosimulaatiokokeissa. Myöskään uutta silmänliikekameraa ei ole vielä käytetty kokeissa.

Tällä hetkellä datan käsittelyt ja analyysit sisältävät paljon virhealtista manuaalista työtä, kun dataa kerätään usean ohjelmiston kautta. Ajosimulaattorista dataa kerätään EepSoft-ohjelmiston kautta ja silmänliikekamerassa D-Lab-ohjelmiston kautta. Silmänliikekamerasta saatava katsedata määrittää kuinka usein ja kauan koehenkilön katse kohdistuu esimerkiksi toissijaisen tehtävän laitteeseen tai muuhun AOI:ksi määritellyyn kohteeseen. Katsedatan käsinpisteitys on tehty Noldus Observer XT -ohjelmistolla. Kognitiotieteen tutkimusryhmä on käyttänyt joissain tutkimuksissaan VisGuard Observer -taustasovellusta toissijaisen tehtävän laitteen syötetietojen hankintaan, mutta ajosimulaatiokokeissa sitä ei ole käytetty. EEG-laitteella on käytössä kaksi eri ohjelmistoa, sen oma ohjelmisto DSI-Streamer sekä koehenkilön kuormituksen arviointiin tarkoitettu QStates. Kehitettävän sovelluksen tarkoitus on vähentää laitteilta saatavan datan yhdistämiseen ja synkronointiin liittyvää manuaalista työtä, mutta ei korvata laitteiden omia ohjelmistoja.

Silmänliikekameran ja ajosimulaattorin datan synkronointiin on kehitetty SEER-ohjelmisto vanhalle silmänliikekameralle. Ajosimulaatiolaboratoriolla on tarvetta uudelle ohjelmistolle, joka toimisi uuden silmänliikekameran kanssa, ja jolla saisi datan synkronoitua myös EEG-laitteesta. Lisäksi tavoitteena on synkronoida muun datan kanssa ajosimulaatiokokeen toissijaisesta tehtävästä saatava data, joita ovat koehenkilön kokeen aikana käyttämästä laitteesta, kuten älypuhelimesta tai tabletista, saatavat syötetiedot. Syötetietoja kerätään siitä, kun koehenkilö tekee jotain laitteella. Esimerkiksi koehenkilö etsii ja laittaa soimaan musiikkikappaleen. Syötetietojen keräämiseen ryhmä kehittää sovelluksen käyttäen apuna VisGuard Observer -taustasovellusta.

Sovelluksen tulee olla joustava siinä, että uusia laitteita voi tulevaisuudessa liittää sovellukseen lisäämällä uusia laitemoduuleja. Käyttäjän tulee voida myös määritellä ennen kokeen alkua, mitä laitteita ja mitä mittaustuloksia halutaan mukaan tulostiedostoon. Haluttujen mittaustulosten valikointiin ajosimulaattorista on käytetty Lua-skriptiä, jota ryhmä jatkokehittää niin, että valittavissa ovat keskusteluissa tilaajan kanssa sovitut mittaustulokset.

Alun perin tilaajan tavoite helppokäyttöisestä käyttöliittymästä sovellukselle oli prioriteeteissa viimeisenä, mutta keskusteluissa tilaajan kanssa käyttöliittymän tärkeys on noussut korkeammalle. Ryhmä kehittää vähintään yksinkertaisen käyttöliittymän sovellukselle, ja ominaisuuksia lisätään ajan salliessa.

Toissijaisena tavoitteena tilaajalla oli alun perin katsedatan pisteytykseen liittyvä ominaisuus, joka korvaisi heidän käyttämänsä Noldus Observer XT -ohjelmiston. Ohjelmisto ei ole optimaalinen, ja sen käyttö on vaatinut paljon videokuvan tarkastamista jälkikäteen pisteytyksen varmistamiseksi. Katsedatan saaminen ja pisteyttäminen ovat helpompia uudella silmänliikekameralla, joten varsinaista pisteyttämistä kehitettävän sovelluksen ei tarvitse tehdä. Silmänliikekameran ohjelmisto ei kuitenkaan tue reaaliaikaista katsedatan pisteytystä, joten katsedatan synkronoiminen muun datan kanssa on edelleen tärkeä

tavoite. Kehitettävän sovelluksen tulee voida yhdistää ja synkronoida katsepisteytysdata jälkikäteen muun datan kanssa.

Kolmanneksi tavoitteeksi tilaaja oli listannut tunnuslukujen automaattisen laskemisen raakadatasta. Heillä on ollut käytössä R-kielinen ohjelma joidenkin tunnuslukujen laskemiseen. Kehitettävässä sovelluksessa voisi olla ominaisuus, jolla tunnusluvut saadaan laskettua automaattisesti. Keskusteluissa tilaajan kanssa tämän ominaisuuden prioriteetti on sovittu alhaiseksi. Projektiryhmä toteuttaa sen, jos jää aikaa muun sovelluksen ollessa valmis, mutta ominaisuus on myös mahdollista rajata projektin ulkopuolelle.

3.2 Toteutettavan sovelluksen rakenne

Tavoitteena on, että sovellus synkronoi datan eri laitteista. Alun perin tilaaja toivoi, että synkronointi tapahtuisi reaaliajassa. Ryhmän perehdyttyä kokeissa käytettäviin laitteisiin on ilmennyt, että reaaliaikainen synkronointi ei ole käytännöllinen ratkaisu. Tilaajalle sopii myös ratkaisu, jolla data saadaan synkronoitua jälkikäteen aikaleimojen avulla. Data kerätään, jäsennetään ja kirjoitetaan väliaikaisiin tiedostoihin reaaliaikaisesti niiltä osin kuin se on mahdollista. Data yhdistetään ja synkronoidaan datan keräyksen jälkeen.

Ajosimulaattoriohjelmisto on hyvin joustava ja räätälöitävissä erilaisiin tarpeisiin. Myös uusi silmäliikekamera (Ergoneers Dikablis Professional) ja EEG-laite (Wearable Sensing DSI-24) tukevat joiltain osin reaaliaikaisuutta. Toissijaisen tehtävän laitteen syötetietojen reaaliaikaiseen keräämiseen tilaaja toivoo projektin jäseniltä sovellusta, jonka toteutusratkaisua valittaessa voi käyttää apuna kognitiotieteen tutkimusryhmän muissa tutkimuksissa käyttämää VisGuard Observer -taustasovellusta. Silmäliikekameran katsedatan pisteytystä ei voi tehdä reaaliajassa, joten katsepisteytysdata yhdistetään muuhun dataan jälkikäteen.

Toteutettavan sovelluksen olennaisimmat ominaisuudet ovat seuraavat:

- Sovellus toimii käyttöjärjestelmässä Microsoft Windows 10.

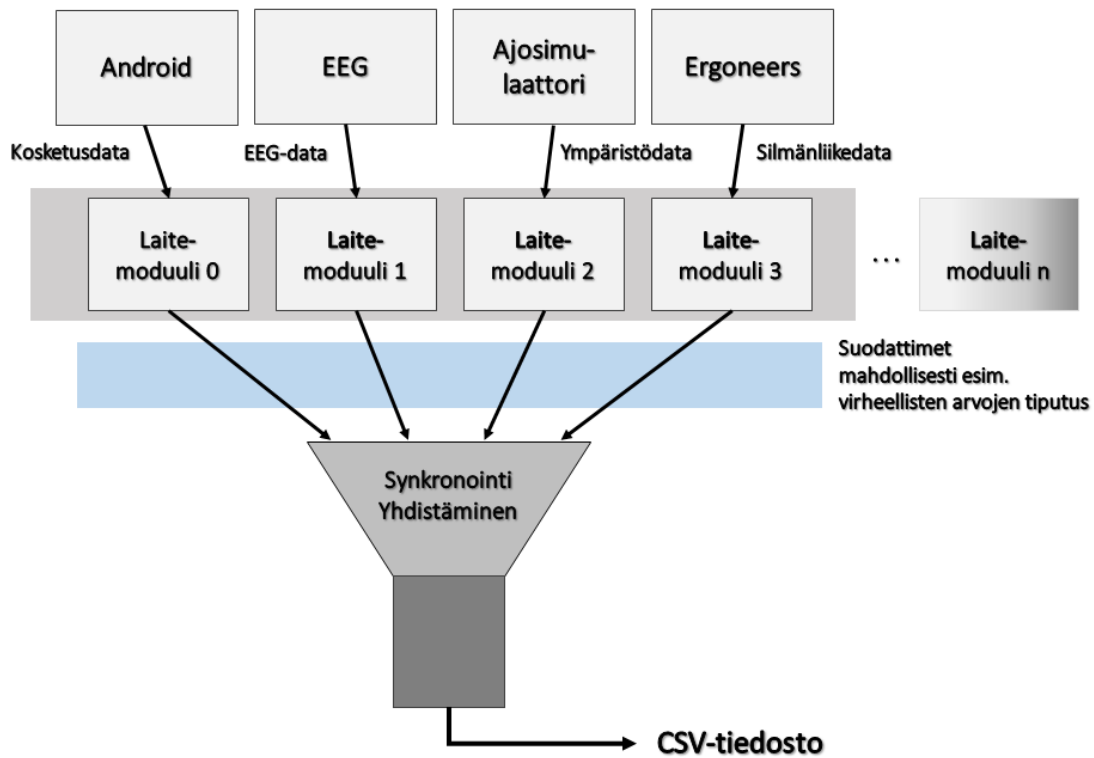
- Kokeessa käytettävät laitteet voidaan valita.
- Käytettävien laitteiden tarjoamasta datasta voidaan valita halutut dataosiot.
- Sovelluksen muodostama tulostiedosto on CSV-muotoinen.
- Tulostiedoston tiedostonimi sisältää tiedot kokeesta, koehenkilöstä, tehtävästä ja tehtävän suoritusajasta (aikaleima).
- Tulostiedosto sisältää yhden koehenkilön yhden tehtävän valittujen laitteiden valitut dataosiot synkronoituna.
- Tulostiedosto voidaan muodostaa niin, että se sisältää myös jälkikäteen lisättyä dataa.

Eri laitteilla on erilaisia viiveitä ja tiedostomuotoja. Ryhmä kehittää sovelluksen niin, että eri laitteille on omat moduulit, jotka lukevat datan laitteilta ja kirjoittavat ne väliaikaisesti omiin tiedostoihinsa jäsenneyssä ja yhtenäisessä muodossa. Dataan kirjataan aikaleimat, jolloin data voidaan yhdistää ja synkronoida yhteen tulostiedostoon. Uusien laitteiden lisääminen sovellukseen tehdään ohjelmointitaitoiselle käyttäjälle mahdolliseksi valmiskomponentteja hyödyntäen. Sovelluksen käyttöä helpottamaan kehitetään graafinen käyttöliittymä.

Graafisen käyttöliittymän (GUI) olennaisimmat ominaisuudet ovat seuraavat:

- Tallennus saadaan päälle GUI:sta.
- Tallennus saadaan pois päältä GUI:sta.
- GUI:ssa käytettävä kieli on englanti.
- GUI:sta voidaan käskä sovellusta muodostamaan tulostiedosto.
- GUI ilmaisee selkeästi, tuleeko laitteilta dataa ja tallentuuko se.

Kuvassa 1 on esitelty toteutettavan sovelluksen yleisrakenteen hahmotelma.



Kuva 1: Sovelluksen yleisen rakenteen hahmotelma.

3.3 Sovelluksen työnkulku ja toteutusratkaisut

Kehitettävä sovellus synkronoi eri laitteilta tulevan datan jatkokäsittelyn ja analyysien helpottamiseksi. Data kerätään joko TCP- tai UDP-yhteydellä ajosimulaattorista, silmänliikekamerasta ja EEG-laitteesta. Toissijaisen tehtävän datankeruutapa tarkentuu projektin edetessä. Laitekohtaisesti joko kerätään kaikki data ja suodatetaan sitten halutut osiot tai pyydetään vain halutut osiot.

Data muutetaan ohjelmointikielen tukemiksi tietotyypeiksi ja tallennetaan väliaikaisesti tiedostoihin. Niistä data välitetään datan yhdistämisestä vastaavalle komponentille. Jokaisella tietotyypillä on yhtenäinen rajapinta merkkijonoksi muuntamiseen, jolloin CSV-tiedoston muodostaminen onnistuu. Synkronoinnissa hyödynnetään eri laitteiden datalle annettuja yhtenäisiä aikaleimoja, mutta tarkempi synkronointitapa ei ole vielä selvillä. Ryhmä selvittää eri vaihtoehtoja datan synkronoinnille.

Kehitettävä sovellus tallentaa jokaisen yksittäisen koehenkilön yksittäisestä koetehtävästä eri laitteista kerätyt dataosiot yhteen tulostiedostoon. Laitteista kerätään numeerista tietoa tulostiedostoon, jonka tulee olla selkeä ja helposti käsiteltävä. Yhden koehenkilön eri koetehtävien datan tulee olla helposti yhdistettävissä. Lisäksi tiedostomuodon tulee olla sellainen, että sen voi sellaisenaan jakaa avoimena datana, eli tiedoston otsikkorivistä tulee selvittää mitä dataa sarake sisältää.

Sovellus kehitetään työasemalle C#-kielellä, ja se toimii Microsoft Windows-ympäristössä. Sovelluksen käytön helpottamiseksi kirjoitetaan käyttöohjeet. Sovellus kommentoidaan huolella, jotta käyttäjät voivat helposti kehittää sovellusta eteenpäin. Jatkokehityksen mahdollistamiseksi kehitetään myös tiedossa olevien laitteiden valmiskomponentteja, joita voi hyödyntää uusien laitteiden lisäämisessä sovellukseen. Laitemoduulien lisäämiseen laaditaan ohjeet, mutta lisääminen tulee luultavasti vaatimaan jatkokehittäjältä ohjelmointitaitoja, sillä lisättävien laitteiden ominaisuudet pitää ottaa huomioon laitemoduulia lisättäessä.

Sovellukselle kehitetään graafinen käyttöliittymä, jolla sovellusta voidaan käyttää. Vähintään käyttäjän tulee voida käyttöliittymän kautta määrittää kokeen nimi, koehenkilön koodi ja tehtävän numero. Näiden tietojen sekä aikaleiman perusteella generoidaan tulostiedoston nimi. Käyttöliittymässä tulee voida laittaa tallennus päälle ja pois. Siinä tulee olemaan yleinen visuaalinen indikaattori, osoittaen tuleeko laitteilta dataa ja tallentuuko se. Ajan salliessa käyttöliittymään kehitetään ominaisuudet, joiden avulla käyttäjä voi myös määrittää, miltä laitteilta dataa halutaan ja mitä mittaustuloksia miltäkin laitteelta halutaan mukaan tulostiedostoon. Koasetukset voi tallentaa asetustiedostoon, jolloin kokeen asetuksia voi hyödyntää eri koehenkilöiden kokeissa. Laitekohtaiset visuaaliset indikaattorit osoittavat, tuleeko valituilta laitteilta dataa ja tallentuuko se.

3.4 Tulokset

Sovelluksen lisäksi projektiryhmä toteuttaa seuraavat tulokset:

- **Ajankäyttöraportti** sisältää ryhmän jäsenten työtunnit, sekä niiden jakautumisen eri tehtäväkokonaisuuksille ja tehtäville.
- **Arkkitehtuurikuvaus** sisältää rinnakkaiset kuvaukset sovelluksen arkkitehtuurista eri näkymien mukaisesti.
- **Esittelymateriaali** sisältää väli- ja loppuesittelyjen esitysgrafiikat ja muistiot.
- **Itsearvioinnit** sisältävät ryhmän jäsenten arviot omasta toiminnastaan, kokemuksistaan ja oppimisestaan projektin aikana.
- **Kehitysympäristön pystytysohjeet** sisältävät ohjeet ryhmän käyttämän kehitysympäristön muodostamiseen jatkokehityksen helpottamiseksi.
- **Käyttöohjeet** neuvovat, miten sovellusta käytetään.
- **Laitemoduulin muodostamisen ohjeet** sisältävät ohjeet, joiden avulla käyttäjä voi lisätä uusia laitteita sovellukseen valmiskomponentteja hyödyntäen.
- **Lisenssisitoumuksella** projektiryhmän jäsenet sitoutuvat sijoittamaan toteuttamansa lähdekoodin valitun avoimen lähdekoodin lisenssin alaisuuteen.
- **Lähdekoodi** sisältää sovelluksen lähdekoodin kommentteineen.
- **Palaverien dokumentit** sisältävät palaverien esityslistat, tilakatsaukset ja pöytäkirjat.
- **Projektiraportti** kuvaa projektin toteutunutta läpivientiä sekä vertaateumaa suunnitelmaan kuvaten mm. syitä ja vaikutuksia.
- **Projektisuunnitelma** on projektin läpivientiä kuvaava suunnitelma, jossa kuvataan muun muassa projektin termistö, tavoitteet, projektiorganisaatio ja resurssit, projektiryhmän jäsenten roolit, vastualueet, tehtävät ja työmäärät sekä projektissa käytettävä prosessimalli, aikataulu ja riskien hallintaa.

- **Sovellusraportti** kuvaa toteutetun sovelluksen oleellimmat toteutusratkaisut ja toiminnot, tavoitteiden toteutumista sekä mahdolliset puutteet, heikot toteutusratkaisut ja jatkokehitysideat.
- **Sähköpostiarkistot** sisältävät kaikki projektin kahdelle sähköpostilistalle lähetetyt sähköpostiviestit.
- **Testausraportit** kuvaavat kullakin testauskerralla suoritettujen testitapausten tulokset sekä havaitut virheet ja puutteet.
- **Testaussuunnitelma** kuvaa suoritettavat testitapaukset ja testausympäristön kokoonpanon.
- **Vaatimukset** sisältävät sovelluksen käyttäjilleen tarjoamat tiedot ja toiminnot.

3.5 Oppimistavoitteet

Sovellusprojektissa projektiryhmän jäsenet oppivat ohjelmistokehityksen tietotaitoja ja saavat kokemusta projektityöskentelystä. Peltihamsteri-projektin jäsenet saavat projektissa kattavan kuvan ohjelmistokehitysprojektin läpiviennistä aina aihealueeseen tutustumisesta tulosten toimittamiseen saakka. Projektissa oleellista on kokonaiskuvan saamisen lisäksi ryhmässä työskentelyn taitojen kehittyminen, oman roolin ja vastuun ymmärtäminen, ajankäytön suunnittelu sekä aiemmilla kursseilla opittujen taitojen hyödyntäminen käytännössä. Projekti voi myös antaa selkeyttä siihen, millaisissa työtehtävissä haluaa tulevaisuudessa toimia.

Projektissa on myös tärkeää viestintätaitojen kehittäminen. Projektiryhmä oppii viestimään niin sisäisesti jäsenten kesken kuin ulkoisesti tilaajan edustajien ja ohjaajien kanssa. Jäsenet myös oppivat laatimaan kirjallisia dokumentteja.

Edellisten tavoitteiden lisäksi jäsenet ovat asettaneet seuraavia henkilökohtaisia tavoitteita:

- Mari Kasasen tavoitteena on oppia projektityötapoja, jotka soveltuvat useamman kuukauden pituiseen työskentelyyn. Pienemmillä kursseilla ei ole mahdollista oppia pitkäkestoista projektityöskentelyä.
- Leevi Liimataisen tavoitteena on saada kokemusta projektitoiminnan työtavoista ja parantaa osaamistaan ohjelmistokehittämisessä.
- Marina Mustosen tavoitteena on oppia projektin suunnittelua ja hallintaa, viestintätaitoja ja ohjelmistokehitystyökalujen käyttöä sekä johtamista.
- Juhani Sundellin tavoitteena on saada aidonoloista kokemusta ohjelmistokehityksestä. Hän on aiemmin tehnyt ohjelmointitöitä joko omaksi ilokseen tai opettajan vaatimusten mukaan, mutta ei koskaan tilaajan tarpeiden mukaan. Lisäksi hänen viime vuosien harjoitustyönsä ovat pääasiassa olleet WWW-sovelluksia, joten sovellusprojektin aihe tarjoaa mahdollisuuden oppia uusia asioita.
- Arttu Ylä-Sahran tavoitteena on kehittää projektitaitojaan ml. viestintää ja vuorovaikutusta, jotka hän kokee heikommiksi kohdikseen. Hän haluaa myös kehittää ohjelmoinnissa tarvittavia teknisiä yhteistyötaitoja, kuten Git-versiohallinnan käyttöä.

4 Projektiorganisaatio ja resurssit

Luku esittelee sovellusprojektin jäsenet ja ohjaajat sekä tilaajan edustajat. Luvussa kuvataan myös projektissa käytettävät tilat, laitteet ja kehitystyökalut.

4.1 Projektiorganisaatio

Projektiryhmään kuuluvat Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra. Kaikki jäsenet ovat tietotekniikan maisteriopiskelijoita, Ylä-Sahra on ohjelmointikielten periaatteiden maisteriohjelmassa ja muut ovat ohjelmistotekniikan maisteriohjelmassa.

Kasasella on ohjelmointikokemusta lähinnä muutamalta kurssilta sekä ohjelmointityöstä. Vahvuudekseen hän kokee ohjelmien suunnittelun ja analysoinnin.

Liimataisen aiempi ohjelmointikokemus sisältää kursseilla ja vapaa-ajalla tehdyjä pieniä ohjelmia. Vahvuutena hänellä on visuaalinen puoli ja suunnittelu.

Mustonen ei ole aiemmin toiminut työtehtävissä tai projektipäällikkönä tietotekniikan alalla, mutta hän oli oman biologian alan väitöskirjaprojektinsa projektipäällikkö. Vahvuudekseen hän kokee tehtävien jakamisen osiin, aikatauluissa pysymisestä huolehtimisen ja tieteellisen tekstin laatimisen.

Sundellin aiempi ohjelmointikokemus koostuu opintojen aikana tehdyistä ohjelmista sekä omaksi ilokseen tehdyistä pienimuotoisista ohjelmista. Hän on enimmäkseen ohjelmoinut Pythonilla ja Javalla, joten projektin ohjelmointikieliksi valittu C# vaatii jonkin verran totuttelua. Hänellä on hyvä perustietämys useilta tietotekniikan osa-alueilta, mutta ei varsinaista erikoisosaamista. Hän kuvailee itseään yhteistyökykyiseksi, oppimishaluiseksi ja sinnikkääksi yksityiskohtien viilaajaksi.

Ylä-Sahra on tehnyt pienempiä projekteja ja töitä pääosin itsenäisesti käyttäen usein erikoisempia ratkaisuja ja ideoita. Hän on esimerkiksi käyttänyt funktionaalisia tai muuten kokeellisia kieliä, projekteina muun muassa äänentunnistustyökalu Golla, salasanan vahvuustarkastin Elixirillä ja Boggle-ratkaisin Haskellilla. Sovellusprojektista hän saa kokemusta perinteisemmästä sovelluksen kehittämisestä.

Tilaaajaorganisaatio on Jyväskylän yliopiston informaatioteknologian tiedekunnan kognitiotieteen ajosimulaatiolaboratorio. Tilaajan edustajina toimivat tutkijatohtori Tuomo Kujala ja tohtorikoulutettava Hilikka Grahn.

Projektin vastaavana ohjaajana toimii tietotekniikan laitoksen Jonne Itkonen. Itkonen toimii myös projektin teknisenä ohjaajana. Toisena ohjaajana toimii Jukka-Pekka Santanen. Jyväskylän yliopiston IT-palvelut vastaa ryhmän jäsenten tietokoneiden ja ohjelmistojen ylläpidosta.

Sovellusprojekti-kurssin yhteydessä järjestettävän *Projektiviestintä IT-alalla* -kurssin opettajina ovat Hanna Kivimäki ja Kati Rantala-Lehtola. *Sovellusprojektin hallintaa, viestintää ja työkaluja* -kurssin pääasiallisena opettajana ja yhteyshenkilönä toimii Jukka-Pekka Santanen.

4.2 Tilat ja laitteet

Ryhmän käyttöön on annettu työhuone Ag C226.4 Jyväskylän yliopiston Agora-rakennuksesta. Lisäksi ryhmän varattavissa on Agorassa sijaitseva tietotekniikan sovellusprojektin kokoushuone Ag C418.1, jota pääasiassa käytetään palavereihin tilaajan kanssa. Projektin kaksi ensimmäistä palaveria pidettiin tilaajan tiloissa ajosimulaatiolaboratoriossa VS3. Projektiryhmän käytössä on myös tietotekniikan projektioipintojen yhteistila Ag C225.2. Kyseinen taukahuone sisältää mm. välineet kahvin ja teen keittämiseen.

Työhuone on varustettu viidellä pöytämallisella tietokoneella, joista kaikkiin on liitetty kaksi näyttöä. Kaikkiin työasemiin on asennettu Microsoft Windows

10 -käyttöjärjestelmä. Työpisteisiin on asennettu tarvittavat kehitysympäristöt ja ohjelmistot. Projektiryhmän käytössä on kaksi verkkolevyä, joilla voidaan säilyttää projektiin liittyviä dokumentteja. Projektiryhmän varattavissa on digitaalisanelin palaverien nauhoittamiseen.

4.3 Kehitys- ja dokumentointityökalut

Sovelluskehitykseen valittiin pääasialliseksi ohjelmointikieleksi C# ja ohjelmointiympäristöksi Visual Studio 2017. Graafinen käyttöliittymä kehitetään WPF-kirjastolla (Windows Presentation Foundation). Sovellus syötetietojen keräämiseen Android-laitteilta kehitetään käyttäen apuna Xamarin-kehitystyökalua. Versiohallintaan käytetään YouSource-versiohallintajärjestelmää.

Dokumentointiin käytetään Microsoftin Office-perheen ohjelmistoja ja LaTeX-ladontajärjestelmää MiKTeX-ohjelmistolla. Osa kaavioista tehdään FreeMind-ohjelmistolla. Työajanseurantaan käytetään Petri Heinosen kehittämää Exceltyökirjaa. Työvaiheiden aikataulutusta tehdään GanttProject-sovelluksella.

4.4 Luennot ja perehdytykset

Kaikki projektiryhmän jäsenet suorittavat sovellusprojektin ohessa suositellut oheiskurssit. *Projektiviestintä IT-alalla* -kurssi (3 op) pitää sisällään seuraavat kokonaisuudet:

- vuorovaikutusosaaminen, ryhmäilmiöt, yhteistyötaidot ja ryhmän tehokkuus,
- projektiviestinnän tavoitteet ja muodot,
- kirjoittamisprosessi,
- palautetaidot sekä
- viestinnän ymmärrettävyys ja tilanteenmukaisuus.

Lisäksi kurssi *Sovellusprojektin hallintaa, viestintää ja työkaluja* (1 op) sisältää seuraavat luennot:

- aloitusluento,
- vaatimusmäärittely,
- versiohallinta,
- projektin hallinta ja johtaminen,
- käytettävyyssluento sekä
- tekijänoikeusluento.

Myös oheiskursseihin käytettävä aika kirjataan sovellusprojektin työajanseurantaan.

5 Käytänteet

Luvussa kuvataan projektin käytänteitä. Käytänteiden noudattaminen edesauttaa asetettujen tavoitteiden saavuttamista, tulosten kehittämistä ja projektin hallittua läpivientiä aikataulun mukaisesti.

5.1 Palaverit

Projektiorganisaatio pyrkii pitämään helmikuussa yhteisen palaverin joka viikko ja muina kuukausina vähintään joka toinen viikko. Jos käsiteltäviä asioita ei katsota olevan, voidaan yksittäisiä palavereja perua. Palavereihin kutsutaan kaikki projektiryhmän jäsenet, ohjaajat ja tilaajan edustajat. Projektiorganisaation palaverien lisäksi projektiryhmä pitää sisäisiä palavereja viikoittain.

Jotta projektiorganisaation palaveri on laillinen ja päätösvaltainen, tulee projektiryhmän toimittaa sen esityslista projektiorganisaatiolle vähintään vuorokausi ennen palaveria. Päätösvaltaisessa palaverissa tulee olla paikalla vähintään yksi projektiryhmän edustaja, yksi tilaajan edustaja ja vastaava ohjaaja.

Palavereissa käydään aina läpi edellisen palaverin pöytäkirja, projektipäällikön tilakatsaus projektista, seuraavan palaverin ajankohta, jatkotoimenpiteet ja mahdolliset muut asiat. Edellä mainittujen kohtien lisäksi palavereissa keskustellaan kehitettävän sovelluksen tavoitteista ja toteutusratkaisuista, esitellään projektiryhmän tuloksia sekä tehdään projektin etenemistä koskevia päätöksiä.

Jokaisesta palaverista toimitetaan projektiorganisaatiolle myös pöytäkirja, jonka palaverissa sihteerinä toiminut projektiryhmän jäsen laatii. Lisäksi projektiorganisaatiolle toimitetaan palavereissa käsitellyt materiaalit. Puheenjohtaja johtaa keskustelua ja huolehtii palaverin etenemisestä esityslistan mukaisesti. Jokaisen projektiryhmän jäsenen tulee toimia vähintään kahdesti sekä puheenjohtajana että sihteerinä.

5.2 Tiedotus

Projektin aikana projektiorganisaation tiedotuksesta vastaa ensisijaisesti projektipäällikkö. Projektiryhmän ulkoisessa viestinnässä jäsenet vastaavat omien vastuualueidensa viestinnästä tarvittaville sidosryhmille. Projektiryhmän sisäisessä viestinnässä jäsenet vastaavat työmäärien, töiden etenemisen ja mahdollisten ongelmien raportoinnista projektipäällikölle.

Projektiorganisaation tiedottamista varten on perustettu kaksi sähköpostilistaa. Sähköpostilistaa `peltihamsteri@korppi.fi` käytetään projektiryhmän, tilaajan edustajien ja ohjaajien tiedotukseen. Sen sähköpostiarkisto on osoitteessa `https://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri`. Lisäksi projektiryhmän jäsenten ja ohjaajien käytössä on sähköpostilista `peltihamsteri_opetus@korppi.fi`. Sen sähköpostiarkisto on osoitteessa `https://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri_opetus`.

Projektiryhmän sisäinen tiedotus tapahtuu suullisen viestinnän lisäksi Slack- ja Whatsapp-pikaviestimillä.

5.3 Tiedostojen nimeäminen

Lähdekooditiedostojen nimeämisessä käytetään C#-ohjelmointikielen yleisiä käytänteitä. Tiedostojen nimet ja hakemistojen nimet kirjoitetaan englanniksi.

Dokumenttitiedostot nimetään sisältöä kuvaavilla nimillä ja dokumentin kielen mukaisesti. Kussakin tiedostonimessä on ensiksi projektin nimi ja sen jälkeen dokumentin nimi. Juokseva numero ja päivämäärä lisätään toistuvien tapahtumien tiedostoihin, kuten palaverien pöytäkirjoihin. Jos nimi koostuu useammasta kuin yhdestä sanasta, käytetään välimerkinä alaviivaa.

5.4 Hakemistorakenne

Projektiryhmän laatimat tulokset tallennetaan sekä CD-levylle että julkiseen WWW-hakemistoon seuraavan hakemistorakenteen mukaisesti:

```
sovellus
  lähdekoodi
  asennuspaketti
dokumentit
  ajankaytto
  esittelyt
  itsearviointit
  lisenssisitoumus
  ohjeet
  projektiraportti
  projektisuunnitelma
  sovellusraportti
  suunnitteludokumentit
  vaatimukset
palaverit
  esityslista
  poytakirjat
  tilakatsaukset
sahkopostiarkistot
  peltihamsteri
  peltihamsteri_opetus
testaus
  testausraportit
  testaussuunnitelmat
```

5.5 Lähdekoodi

Sovelluksen lähdekoodi kirjoitetaan ja kommentoidaan englanniksi. Aliohjelmat, luokat ja muuttujat nimetään kuvaavilla englanninkielisillä nimillä C# -kielen käytänteiden mukaisesti [6].

5.6 Testaus

Ohjelmoija suorittaa tarvittavat yksikkötestaukset kirjoittamalleen lähdekoodille. Sovelluksen käyttöliittymää testataan kehityksen aikana koko ajan jäsenen toimesta, jotta varmistutaan sen toiminnasta ja laadusta. Toteutetun sovelluksen laadunvarmistus tapahtuu järjestelmätestauksella, joka suoritetaan kaksi kertaa. Projektin loppupuolella järjestetään hyväksymistestaus tilaajan kanssa, jolla varmistetaan vaatimusmäärittelyssä asetettujen vaatimusten täyttyminen.

Järjestelmätestauksesta vastaava henkilö laatii testaussuunnitelman sekä vastaa testauksen läpiviennistä. Vastaavan henkilön lisäksi muut projektiryhmän jäsenet ovat mukana järjestelmätestauksen läpiviennissä. Testaussuunnitelma sisältää testauskerroilla suoritettavat testitapaukset kuvattuina vaihe vaiheelta. Järjestelmää testataan eri laiteyhdistelmillä ja keräten eri dataosioiden dataa laitteilta. Testaukserran suorittaneen jäsenen laatimassa testausraportissa kuvataan testaukserran havainnot sekä mahdolliset havaitut virhetilanteet ja puutteet.

Hyväksymistestaukseen osallistuu hyväksymistestauksen vastuuhenkilön lisäksi tilaajat, jotka tulevat olemaan sovelluksen pääkäyttäjät. Hyväksymistestaus suoritetaan tilaajan tiloissa ajosimulaatiolaboratoriossa. Hyväksymistestauksen vastuuhenkilö laatii testaussuunnitelman ja testausraportin tuloksista yhdessä tilaajan edustajan kanssa. Suunnitelma sisältää tilaajien toimesta suoritettavat toimenpiteet. Hyväksymistestauksen vastuuhenkilön laatima testausraportti kuvaa tilaajien huomiot sekä tilaajien arviot vaatimusten täyttymisestä.

5.7 Versiohallinta ja -numerointi

Sovelluksen lähdekoodin versiohallintaan käytetään Git-versiohallintaohjelmistoa. Lähdekoodi sijoitetaan YouSource-palveluun, jossa se on projektiryhmän ja ohjaajien sekä tilaajan edustajien nähtävissä projektin ajan.

Julkistettavissa dokumenteissa käytetään kolmiportaista numerointia (i.j.k). Projektiryhmän sisäisiä versioita numeroidaan pienimmällä merkitsevällä numerolla paikassa k, kuten i.j.1. Projektioorganisaatiolle julkistettavat versiot numeroidaan toisella merkitsevällä numerolla paikassa j, kuten i.1.k. Ensimmäisen koko projektioorganisaation hyväksymän dokumentin versio on 1.0.0, ja tätä seuraavien hyväksytyjen versioiden numerointia kasvatetaan toisen merkitsevän numeron osalta.

5.8 Katselmoinnit ja tulosten hyväksyminen

Projektin aikana sovelluksen lähdekoodi katselmoidaan teknisen ohjaajan toimesta vähintään kaksi kertaa. Katselmoinneissa ohjaaja antaa ehdotuksia lähdekoodin parantamiseen. Katselmointiin osallistuu teknisen ohjaajan lisäksi koko projektiryhmä. Tilaajan edustaja osallistuu katselmointiin halutessaan.

Tekninen ohjaaja hyväksyy lähdekoodin viimeisen katselmoinnin yhteydessä tai jälkeinpäin sähköpostitse. Vastaava ohjaaja ja tilaajan edustajat hyväksyvät keskeiset laaditut dokumentit, joita ovat ainakin projektisuunnitelma, projektiraportti, vaatimukset, käyttöohjeet ja sovellusraportti.

5.9 Tulosten koostaminen ja toimittaminen

Projektiryhmä kokoaa luvussa 3.4 kuvatut projektin tulokset sekä tulostettuina projektikansioon että tallennettuina CD-levylle luvussa 5.4 kuvatun hakemistorakenteen mukaisesti. Tulokset toimitetaan tilaajalle CD-levyllä. Informaatioteknologian tiedekunnalle toimitetaan tuloksista koostettu projektikansio ja CD-levy. Tiedekunnan arkistoon toimitetaan lisäksi yksi CD-levy.

6 Roolit, vastualueet ja tehtävät

Luvussa määritellään projektiryhmän jäsenten roolit, vastualueet ja tehtävät. Lisäksi arvioidaan projektin tehtävien työmääriä ja esitellään tehtävien jakautumista projektiryhmän jäsenille.

6.1 Roolit ja vastualueet

Projektipäällikkönä toimii Marina Mustonen. Projektipäällikön vastuulla on projektin suunnittelu, hallinta ja raportointi sekä sisäinen että ulkoinen viestintä. Projektipäällikön vastuulla on myös tehtävien jakaminen projektiryhmän jäsenille ja projektin etenemisen valvominen. Jos projektipäällikkö on jostain syystä estynyt hoitamaan omia vastualueitaan, siirtyy vastuu varapäällikkönä toimivalle Mari Kasaselle.

Taulukossa 1 on määriteltynä jäsenten vastualueita projektin tuloksiin liittyen. Vastuuhenkilö vastaa pääosin kyseisen tuloksen saattamisesta valmiiksi, sekä siihen liittyvästä viestinnästä projektiorganisaatioon kuuluvien henkilöiden ja sidosryhmien kanssa. Vastuualueen vastuuhenkilö ei välttämättä toteuta kyseistä vastuualuetta yksin.

Vastuualue	Vastuuhenkilö
Projektisuunnitelma	Marina Mustonen
Projektiraportti	Marina Mustonen
Vaatimusmäärittely	Mari Kasanen
Ajosimulaattori	Arttu Ylä-Sahra
EEG	Mari Kasanen
Silmänliikekamera	Juhani Sundell
Toissijaisen tehtävän laite (Android)	Leevi Liimatainen
Datan keruukomponentit	Juhani Sundell
Datan synkronointikomponentit	Mari Kasanen
Käyttöliittymä	Leevi Liimatainen
Järjestelmätestaus	Arttu Ylä-Sahra
Hyväksymistestaus	Marina Mustonen
Sovellusraportti	Juhani Sundell
Ohjeet	Leevi Liimatainen
Yhteenvetosivu	Marina Mustonen

Taulukko 1: Vastuualueet ja vastuuhenkilöt.

6.2 Tehtävät ja niiden työmäärät

Taulukossa 2 on arvioitu eri tehtävien työmääriä. Arvioita on tehty aikaisempien projektien ja tämän projektin jo toteutuneiden tuntien lisäksi valistuneiden arvioiden pohjalta.

Projektiryhmän jokaisen jäsenen tavoitteena on käyttää Sovellusprojektiin 20 työtuntia viikossa. Työtuntien arvioinnin mukaan Sovellusprojektin läpivientiin tarvitaan yhteensä noin 1350 tuntia. Projektin ohella suoritettaviin oheiskursseihin on laskettu työtunnit erikseen.

Tehtäväkokonaisuus - Tehtävä	MK	LL	MM	JS	AY	Kaikki
Projektin hallinta						
Projektisuunnitelma	1	1	50	1	1	54
Projektiraportti	2	2	32	2	2	40
Seuranta ja hallinta	0	0	15	0	0	15
Tiedotus	2	2	6	2	2	14
Tulosten viimeistely ja kokoaminen	6	5	14	5	5	35
Yhteensä	11	10	117	10	10	158
Palaverit						
Esityslistat	2	2	2	2	2	10
Palaverit	20	20	20	20	20	100
Pöytäkirjat	9	9	9	9	9	45
Tilakatsaukset	0	0	8	0	0	8
Yhteensä	31	31	39	31	31	163
Esitutkimus						
Aihealueeseen tutustuminen	15	15	15	15	15	75
Laitteisiin tutustuminen	20	20	5	30	15	90
Työkaluihin tutustuminen	10	10	5	10	10	45
Yhteensä	45	45	25	55	40	210
Vaatimusmäärittely						
Suunnittelu	5	2	2	2	2	13
Raportointi	10	0	0	0	0	10
Yhteensä	15	2	2	2	2	23
Suunnittelu						
Tiimitapaaminen	25	25	25	25	25	125
Ajosimulaattorimoduuli	1	1	1	1	10	14
EEG-moduuli	10	1	1	1	1	14
Silmänliikekameramoduuli	1	1	1	30	1	34
Android-moduuli	1	15	1	1	1	19
Datan keruukomponentit	6	6	4	12	12	40
Datan synkronointikomponentit	8	6	6	4	15	39
Käyttöliittymä	15	20	3	3	3	44
Yhteensä	67	75	42	77	68	329
Toteutus						
Ajosimulaattorimoduuli	1	1	1	1	10	14
EEG-moduuli	10	1	1	1	1	14
Silmänliikekameramoduuli	1	1	1	10	1	14
Android-moduuli	1	10	1	1	1	14
Datan keruukomponentit	6	6	4	12	20	48
Datan synkronointikomponentit	9	6	6	9	21	51
Käyttöliittymä	40	44	1	1	1	87
Yhteensä	68	69	15	35	55	242

Järjestelmätestaus						
Suunnittelu	2	2	2	2	24	32
Testaus	10	10	2	10	10	42
Raportointi	0	0	1	0	10	11
Yhteensä	12	12	5	12	44	85
Hyväksymistestaus						
Suunnittelu	1	1	2	1	1	6
Testaus	0	0	3	0	0	3
Raportointi	0	0	4	0	0	4
Yhteensä	1	1	9	1	1	13
Viimeistely						
Sovellusraportti	2	2	4	30	2	40
Lähdekoodin viimeistely	4	4	1	4	4	17
Katselmoinnit	4	4	4	4	4	20
Kehitysympäristön pystytysohjeet	1	1	0	1	1	4
Käyttöohjeet	5	10	4	2	2	23
Laitemuodulin lisäämisohjeet	2	2	1	4	4	13
Sovelluksen luovutus	2	2	2	2	2	10
Yhteensä	20	25	16	47	19	127
Projektin tunnit yhteensä	270	270	270	270	270	1350
Oheiskurssit						
Projektiviestintä IT-alalla (XYHI004)	50	50	50	50	50	250
Sovellusprojektin hallinta (TIES412)	24	24	24	24	24	120
Yhteensä	74	74	74	74	74	370
Projektin ja oheiskurssien tunnit yhteensä	344	344	344	344	344	1720

Taulukko 2: Tehtävien työmäärät ja tehtäväjako.

7 Prosessi ja aikataulu

Luvussa kuvataan sovelluksen kehitystyön tueksi valittua prosessia. Lisäksi esitellään projektin aikataulusuunnitelma, joka toimii työkaluna projektin hallinnassa.

7.1 Prosessi

Suunnitteluvaiheessa kartoitetaan sovelluksen vaatimuksia tilaajan edustajien kanssa keskustellen. Keskustelujen pohjalta laaditaan sovellusta ja projektin läpivientiä kuvaavia määrittely- ja suunnitteludokumentteja.

Sovellusprojektissa noudatettava sovelluksen kehitysprosessi mukailee spiraalimallia [5], jota räätälöidään tarvittaessa projektin tarpeisiin sopivalla tavalla. Spiraalimalli on riskien arviointiin pohjaava iteratiivinen prosessimalli, johon voi ottaa ominaisuuksia muista malleista projektin tarpeista riippuen. Ohjelmiston kehittäminen tapahtuu sykleissä, joissa toistuvat samat askeleet. Alussa syklit ovat nopeita ja pitenevät projektin edetessä. Sykli alkaa ohjelmiston tavoitteiden, vaihtoehtojen ja rajoitusten kartoittamisella. Seuraavaksi tehdään riskianalyysi, jossa kartoitetaan projektiin liittyviä epävarmuuksia ja eri vaihtoehtojen riskejä. Kolmannessa askeleessa voidaan riskianalyysin tulosten ja projektin tarpeiden perusteella tehdä esimerkiksi prototyyppejä, simulaatioita, analyttistä mallinnusta tai näiden yhdistelmiä, sekä projektin loppupuolella varsinaisen ohjelman kehittelyä. Neljännessä askeleessa arvioidaan aikaansaatuja tuloksia ja suunnitellaan seuraavaa sykliä, joka aloitetaan jälleen tavoitteiden ja vaihtoehtojen kartoittamisella.

Peltihamsteri-projektissa aloitetaan vaatimusmäärittelyn laatiminen heti ensimmäisen palaverin (1.2.2019) jälkeen. Vaatimusmäärittely laaditaan tilaajan kanssa käytyjen keskustelujen pohjalta, ja siinä priorisoidaan sovelluksen vaatimukset. Vaatimukset tarkentuvat projektin edetessä, mutta ensimmäinen versio toimitetaan tilaajille ensimmäisten seitsemän viikon aikana.

Ryhmä aloittaa ensimmäisen syklin kaksi viikkoa projektin alun jälkeen, jolloin on saatu alustava käsitys sovelluksen vaatimuksista. Ryhmä aloittaa syklin miettimällä vaihtoehtoja, joilla voidaan päästä tavoitteisiin ja täyttää tilaajan asettamat vaatimukset. Ryhmä pohtii eri vaihtoehtoja ja niiden rajoitteita. Seuraavaksi ryhmä kartoittaa vaihtoehtojen riskit. Kolmanteen askeleeseen sopii kahdessa ensimmäisessä syklissä prototyypilähtöinen lähestyminen. Kehitettävä sovellus on uusi (eli kyseessä ei ole olemassa olevan sovelluksen jatkokehitys), ja kaikkia sovellukseen liitettäviä laitteita ei ole vielä testattu käytössä, mikä tuo epävarmuutta niiden toiminnasta ja ominaisuuksista. Ensimmäisessä syklissä kehitettävillä prototyypeillä kokeillaan, mikä on toimivaa ja mikä ei. Syklin neljännessä askeleessa arvioidaan kehitettyjä prototyyppejä ja päätetään, otetaanko ne jatkokehitykseen seuraavaan sykliin vai hylätäänkö ne. Ensimmäinen sykli on luonteeltaan kokeilullinen, joten se pidetään lyhyenä (vajaa kaksi viikkoa).

Toinen sykli kestää kaksi viikkoa. Se aloitetaan käymällä läpi tarkentuneita vaatimuksia ja pohtimalla vaihtoehtoja. Ryhmä jälleen kartoittaa eri vaihtoehtojen riskejä ja kolmannessa askeleessa kehittää prototyyppejä. Neljännessä askeleessa arvioidaan prototyyppien toimivuus ja jatkokehittelypotentiaali.

Ensimmäisessä ja toisessa syklissä keskitytään kehittämään käyttöliittymän lisäksi datan keräämiseen liittyviä komponentteja. Kolmannessa ja neljännessä kehitetään myös datan synkronointiin liittyviä komponentteja.

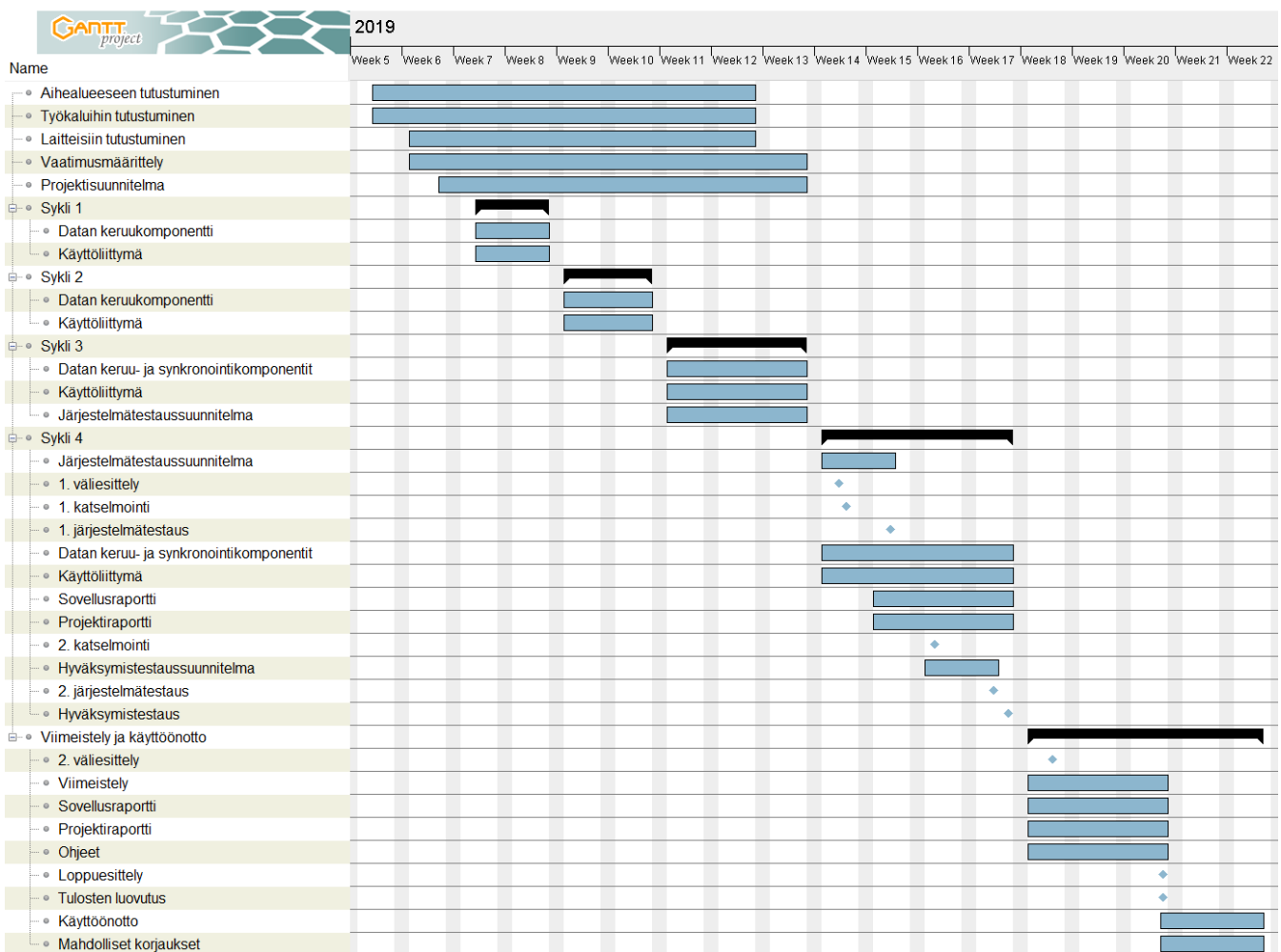
Kolmannessa ja neljännessä syklissä ensimmäisessä askeleessa käydään läpi ja tarvittaessa tarkennetaan vaatimuksia sekä pohditaan vaihtoehtoja. Riskit kartoitetaan toisessa askeleessa ja kolmannessa askeleessa kehitetään sovellusta. Neljännessä askeleessa arvioidaan tuloksia. Kolmas sykli kestää kolme viikkoa ja neljäs sykli neljä viikkoa. Neljännen syklin päätteeksi tehdään hyväksymistestaus, jonka jälkeen sovellukseen tehdään tarvittaessa vielä korjauksia. Tarvittaessa korjausten jälkeen tehdään toinen hyväksymistestaus.

Projektiryhmän sisäisiä palavereja pidetään vähintään viikottain. Niissä käydään läpi syklin silloisen askeleen mukaisia asioita, ml. tarpeiden ja vaihtoehtojen kartoittaminen, riskianalyysi tai tulosten pohdinta ja seuraavan syklin suunnittelu.

Viimeistelyvaiheessa viimeistellään sovellus, lähdekoodi ja dokumentit tilaajalle luovutusta varten.

7.2 Aikataulu

Kuvassa 2 esitellään projektin suunniteltu aikataulu. Projektin tulokset luovutetaan tilaajalle toukokuun 2019 puoliväliin mennessä.



Kuva 2: Projektin suunniteltu aikataulu.

8 Riskien hallinta

Luvussa kuvataan projektin mahdollisia riskejä, sekä arvioidaan niiden vakavuuksia ja vaikutuksia projektin läpivientiin ja tuloksiin. Lisäksi esitellään tavat riskien ehkäisemiseen ja riskin toteutuessa niistä toipumiseen.

8.1 Riskien todennäköisyydet ja haittavaikutukset

Taulukon 3 ennakoitavat riskit voivat sovellusprojektin aikana toteutua ja toteutuessaan vaikuttaa projektin tuloksiin tai läpivientiin. Lisäksi taulukossa esitetään arvio riskin todennäköisyydestä ja haittavaikutusten vakavuudesta riskin toteutuessa. Asteikkona todennäköisyyden ja haittavaikutusten arvioinnissa on käytetty kolmiportaista jaottelua: korkea, keskitaso ja matala.

Riski	Todennäköisyys	Vakavuus
Laitteiden erilaisuus ja tietojen puute	Korkea	Korkea
Sovelluksen joustavuuden toteuttamisen vaikeus	Korkea	Matala
Ominaisuuksien toteuttamisen vaikeus	Korkea	Matala
Ryhmän tietotaitojen puutteet	Keskitaso	Keskitaso
Ryhmän jäsenten muut sitoumukset	Matala	Keskitaso

Taulukko 3: Projektin riskit.

Luvuissa 8.2–8.7 esitellään ennakoituja riskejä sekä niiden vaikutuksia projektin läpivientiin ja tuloksiin. Luvuissa esitellään lisäksi ehkäisykeinot riskien välttämiseksi ja toipumiskeinot riskistä palautumiseen sen toteutuessa.

8.2 Laitteiden erilaisuus ja tietojen puute

Kehitettävällä sovelluksella tullaan käsittelemään dataa useasta eri laitteesta, ml. ajosimulaattori, silmänliikekamera, EEG-laite ja toissijaisessa tehtävässä käytettävä laite. Laitteissa on eri ohjelmistot, eri viiveet datan saamisessa ja pääosin eri tiedostomuodot. Sovelluksella on tarkoitus synkronoida ja tallentaa data niistä yhteen tulostiedostoon. Laitteista uutta silmänliikekameraa ja EEG-laitetta ei ole vielä käytetty ajosimulaatiokokeissa ollenkaan, joten niiden kaikki ominaisuudet ja käyttömahdollisuudet eivät ole täysin tilaajien tiedossa.

Eri ohjelmistojen ja tiedostomuotojen yhteensovittaminen ja viiveen huomioiminen synkronisoinnissa sekä epäselvyys laitteiden ominaisuuksista tuovat haasteita sovelluksen onnistumiselle ja projektin läpiviennille. Ryhmä minimoi riskiä tutustumalla laitteisiin hyvin ja kehittämällä projektin alussa prototyyppinä nähdäkseen, mikä on mahdollista ja toimivaa ja mikä ei. Jos kaikkien laitteiden kaikkia ominaisuuksia ei saada projektin aikana selville, tulostiedosto jää vaillinaiseksi joidenkin haluttujen mittaustulosten osalta, jolloin saattaa mittaustuloksia joutua lisäämään tiedostoon jälkikäteen. Tämä on tilaajasta hyväksyttävä ratkaisu.

8.3 Sovelluksen joustavuuden toteuttamisen vaikeus

Tilaaja toivoo kehitettävältä sovellukselta joustavuutta siinä, että uusia laitteita voidaan lisätä sovellukseen myöhemmin. Mahdollisista uusista laitteista ei ole tietoja, joten niiden ominaisuuksia ja ohjelmistoja ei voi ottaa huomioon sovelluksen kehittämisessä. Riski vaikuttaa sovelluksen jatkokehityskelpoisuuteen, joten siitä toipumiseen ei voida vaikuttaa projektissa.

Ryhmä minimoi riskiä niin, että mahdollisille uusille laitteille voi tehdä valmiita komponentteja nyt tiedossa olevien laitteiden pohjalta. Jatkokehittäjä voi hyödyntää komponentteja uusien laitemoduulien lisäämisessä. Lisättävän laitteen ominaisuuksista riippuen lisääminen vaatinee ohjelmointitaitoja, mutta tilaajalla on itselläänkin ohjelmointitaitoja, joten tämän pitäisi olla mahdollista.

Jos riski toteutuu, eli uusia laitteita ei voi lisätä sovellukseen, se tulee ilmi vasta projektin päätyttyä, jolloin ryhmä ei voi enää vaikuttaa asiaan.

8.4 Ominaisuuksien toteuttamisen vaikeus

Erityisesti tilaajan alkuperäisessä aihe-ehdotuksessa sovellukselle oli lueteltu useita haluttuja ominaisuuksia. Niistä katsedatan pisteytys ja tunnuslukujen laskeminen datasta ovat haastavia toteuttaa. Osittain nämä asiat on jo saatu ratkaistua keskusteluissa tilaajan kanssa, kun osa ominaisuuksista on osoittautunut tarpeettomiksi tai hyvin matalan tärkeysasteen ominaisuuksiksi. Jotkut ominaisuudet, kuten datan reaaliaikainen synkronointi ja tarpeettoman datan suodattaminen tuovat silti haasteita sovelluksen kehittämiseen ja voivat vaikuttaa sovelluksen toteuttamiseen kaikkien vaatimusten osalta.

Ryhmä ehkäisee em. ongelmia tutustumalla laitteiden viiveisiin ja eri synkronointimahdollisuuksiin sekä kehittämällä prototyyppejä datan suodattamisesta. Datan reaaliaikainen synkronointi ei näytä käytännölliseltä vaihtoehdolta, mutta tilaajalle käy myös ratkaisu, jolla saa datan synkronoitua jälkikäteen. Jos dataa ei saada suodatettua niin, että tulostiedostoon tulevat vain halutut mittaustulokset, niin hyväksyttävä ratkaisu on, että kaikki mittaustulokset tulevat tiedostoon, josta käyttäjä voi valikoida haluamansa mittaustulokset.

8.5 Ryhmän tietotaitojen puutteet

Projektin jäsenillä ei ole työelämäkokemusta IT-alalta, ja tämä projekti on ensimmäinen näin ison kokoluokan ohjelmointityö jäsenille. Jäsenillä ei ole paljon kokemusta projektiluontoisesta työskentelystä, joten uutta opittavaa on paljon, ja se saattaa vaikuttaa projektin läpivientiin.

Suurimmalla osalla jäsenistä on kuitenkin hyvät ohjelmointitaidot, joten projektissa tarvittavien taitojen oppimisen pitäisi olla melko nopeaa. Tilaaja on myös antanut vapaat kädet ohjelmointikielen ja työkalujen valinnassa, joten

ryhmä voi valita ohjelmointikielen ja työkalut, joista on jo kokemusta. Lisäksi ohjaaja auttaa tarvittaessa.

8.6 Ryhmän jäsenten muut sitoumukset

Mustosella on opintojen lisäksi osa-aikatyö (15h/viikko), mutta hän pystyy sen ohella sitoutumaan projektiin 20h/viikko kuten muutkin ryhmän jäsenet. Hän myös on lopettamassa työn maaliskuun puolella välissä. Mustosella alkaa kokopäiväinen harjoittelu 20.5., mutta siinä vaiheessa projektissa on suunnitelman mukaan vain dokumenttien ja sovelluksen hiomista, jota Mustonen pystyy tekemään harjoittelun ohella. Jäsenillä ei ole tiedossa matkoja tai muita esteitä, mutta yllättävät tapaukset kuten sairaudet saattavat aiheuttaa poissaoloja. Jäsenten muut sitoumukset saattavat viivästyttää projektin läpivientiä. Riskiä voidaan ehkäistä sillä, että ryhmän jäsen ilmoittaa muusta sitoumuksestaan muulle ryhmälle heti saatuaan tietää siitä. Riskin toteutuessa muu ryhmä edistää poissa olevan jäsenen vastuualuetta, jotta projekti saadaan valmiiksi ajallaan.

9 Yhteenveto

Peltihamsteri-projekti kehittää kevään 2019 Sovellusprojekti-kurssilla Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriolle sovelluksen, jonka avulla synkronoidaan ajosimulaatiokokeen eri laitteista saatava data ja tulostetaan ne tulostiedostoon. Tulostiedostossa dataa on helppo käsitellä ja analysoida.

Projektin tulokset luovutetaan tilaajalle toukokuun puolella välissä. Projektin haasteina ovat riittävä tutustuminen sovelluksen hyödyntämiin laitteisiin sekä sovelluksen joustavuuden ja tiettyjen ominaisuuksien toteuttaminen. Projektin läpiviennin suurimmat riskit liittyvät laitteiden tietojen puutteeseen sekä eri ominaisuuksien ja joustavuuden toteuttamisen vaikeuteen.

Sovellusprojekti-kurssi tarjoaa projektiryhmän jäsenille kokemusta projektimuotoisesta työskentelystä, projektin hallinnasta, tilaajan kanssa toimimisesta, johtamisesta ja ohjelmistokehityksestä. Sovelluksen kehittämisessä jäsenet saavat tilaisuuden hyödyntää aiemmissa opinnoissa omaksumiaan taitoja.

Lähteet

- [1] Severi Jääskeläinen, Samuel Kaiponen, Heta Rekilä ja Sinikka Siironen, "Monisiro-projekti, Projektisuunnitelma", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2.5.2018.
- [2] Sauli Flinkman, Jere Juntila, Tuukka Jurvakainen ja Anette Karhu, "Isäxi-sovellusprojekti, Projektisuunnitelma", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 27.3.2017.
- [3] Jukka-Pekka Santanen, "Tietotekniikan sovellusprojektien ohje", saatavilla PDF-muodossa, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 30.1.2017.
- [4] Jarmo Juujärvi, Sami Kallio, Kai Korhonen, Juha Moisio ja Ilari Paananen, "Moveatis-sovellusprojekti, Projektisuunnitelma", Jyväskylän yliopisto, tietotekniikan laitos, 11.4.2016.
- [5] B. W. Boehm, "A Spiral Model of Software Development and Enhancement", Computer 21(5): 61-72. 1988.
- [6] Anthony D. Green, "C# Language Specifications" <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/introduction>, Microsoft 7.1.2017.