

Hälyri-Sovellusprojekti

Niko Mononen
Veli-Mikko Puupponen
Ilkka Rautiainen
Atte Söderlund

Projektiraportti

Versio: 0.2.0
Julkinen
8. kesäkuuta 2014

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2014		
Tilaaaja	__.__.2014		
Ohjaaja	__.__.2014		

Tietoa dokumentista

Tekijät:

- Niko Mononen (NM) niko.m.mononen@student.jyu.fi
- Veli-Mikko Puupponen (VMP) veli-mikko.a.puupponen@student.jyu.fi
- Ilkka Rautiainen (IR) ilkka.t.rautiainen@student.jyu.fi
- Atte Söderlund (AS) atte.a.soderlund@student.jyu.fi

Dokumentin nimi: Hälyri-projekti, Projektiraportti

Sivumäärä: 39

Tiedosto: Halyri_projektiraportti_0.2.0.tex

Tiivistelmä: Hälyri-projekti suoritti alustavan toteutettavuuskartoituksen ja kehitti hätäilmoitusjärjestelmän prototyypin Sovellusprojekti-kurssilla keväällä 2014. Projektiraportissa kuvataan projektin suunniteltua ja toteutunutta läpivientiä käsitellen muun muassa tavoitteita, resursseja ja käytänteitä. Lisäksi projektiraportissa verrataan projektin toteutumaa suunniteltuun mm. työnjaon ja työmäärien, aikataulun sekä arvioitujen riskien hallinnan osalta ottaen kantaa erojen syihin ja vaikutuksiin. **Avainsanat:** Aikataulu, EKG, haasteet, hätäkeskus, hätäpuhelu, kokemuksia, käytänteet, matkapuhelin, ohjelmistoprojekti, opittua, projektiraportti, prosessi, resursit, riskien hallinta, sovellus, tavoitteet, taustaa, tehtävät, työmäärät, vastuualueet, videopuhelu, Windows Phone 8.

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	6.5.2014	Projektiraportin laatiminen on aloitettu.	NM
0.0.2	10.5.2014	Kirjoitettu projektin taustasta, tavoitteista ja projektiorganisaatiosta.	NM
0.0.3	18.5.2014	Kirjoitettu käytänteistä, työmääristä, työnjaosta ja riskeistä.	NM
0.0.4	22.5.2014	Kirjoitettu tietojärjestelmälle asetettujen tavoitteiden toteutumisesta.	NM
0.1.0	24.5.2014	Korjattu kirjoitusvirheitä.	NM
0.1.1	27.5.2014	Raporttia on korjattu saadun palautteen mukaisesti.	NM
0.2.0	2.6.2014	Analysoitu tavoitteiden saavuttamista.	NM
0.2.1	8.6.2014	Raporttia on korjattu saadun palautteen mukaisesti.	NM

Tietoa projektista

Projekti kehitti prototyypin uudenlaisesta tietojärjestelmästä, joka mahdollistaa yhteyden hädässä olevien ihmisten ja hätäkeskusten välillä. Keskeinen ajatus on hyödyntää matkapuhelinten merkittävästi lisääntyntä datan lähetys- ja vastaanotto-kykyä osana auttamisprosessia. Tavoitteena kehitetyllä tietojärjestelmällä on luoda aikaisempaa tarkempi kuva hätätilanteesta entistä lyhyemmässä ajassa. Tämä auttaa pelastamaan ihmishenkiä, parantamaan palvelun laatua ja tehostamaan resursien käyttöä. Projektissa kehitetyn hätäilmoitusjärjestelmän prototyypin avulla pyritään osoittamaan teknologiset mahdollisuudet ja haasteet sekä havainnollistamaan käyttöliittymillä avuntarvitsijan ja hätäkeskuksen vuorovaikutusta.

Tekijät:

- Niko Mononen (NM) `niko.m.mononen@student.jyu.fi`
- Veli-Mikko Puupponen (VMP) `veli-mikko.a.puupponen@student.jyu.fi`
- Ilkka Rautiainen (IR) `ilkka.t.rautiainen@student.jyu.fi`
- Atte Söderlund (AS) `atte.a.soderlund@student.jyu.fi`

Tilaja:

- Vesa Lappalainen `vesa.t.lappalainen@jyu.fi`
- Tero Tuovinen `tero.tuovinen@jyu.fi`

Ohjaajat:

- Jaakko Kosonen `jaakko.j.kosonen@gmail.com`
- Olli Kauppinen `olli.p.kauppinen@student.jyu.fi`
- Jukka-Pekka Santanen `santanen@mit.jyu.fi`

Asiantuntijat:

- Helena Jantti `helena.jantti@kuh.fi`

Yhteystiedot:

- Sähköpostilistat: halyri@korppi.jyu.fi,
halyri_opetus@korppi.jyu.fi
- Sähköpostiarkistot: <https://korppi.jyu.fi/kotka/servlet/list-archive/halyri/>,
https://korppi.jyu.fi/kotka/servlet/list-archive/halyri_opetus/
- Projektitila: AgC 223.2, puh. 040-3573796

Sisältö

1	Johdanto	1
2	Termit	2
3	Taustaa ja tavoitteiden toteutuminen	4
3.1	Taustaa	4
3.2	Prototyypille asetettujen tavoitteiden toteutuminen	5
3.3	Oppimistavoitteiden toteutuminen	7
3.4	Tulokset ja niiden toteutuminen	8
4	Organisaatio ja resurssit	10
4.1	Projektorganisaatio	10
4.2	Projektin tilat ja laitteet	11
4.3	Dokumentointityökalut	11
4.4	Ohjelmointityökalut	12
4.5	Luennot ja perehdytykset	12
5	Projektin käytänteet	14
5.1	Tiedotus	14
5.2	Palaverit	15
5.3	Tiedostojen nimeäminen	16
5.4	Hakemistorakenne	16
5.5	Lähdekoodi	17
5.6	Versiointi ja versiohallinta	20
5.7	Testaus	20
5.8	Tulosten katselmoinnit ja hyväksyminen	21
5.9	Tulosten koostaminen ja toimittaminen	21
6	Tehtävät, työmäärät ja työnjako	23
6.1	Vastuualueet	23
6.2	Työnositus ja työmäärä	24
6.3	Ryhmän työtunnit tehtäväkokonaisuuksittain	27
6.4	Niko Monosen työtunnit tehtäväkokonaisuuksittain	27
6.5	Veli-Mikko Puupposen työtunnit tehtäväkokonaisuuksittain	27
6.6	Ilkka Rautiaisen työtunnit tehtäväkokonaisuuksittain	27
6.7	Atte Söderlundin työtunnit tehtäväkokonaisuuksittain	27

7 Ohjelmistokehitysprosessi ja aikataulu	28
7.1 Prosessi	28
7.2 Aikataulu	28
7.3 Ryhmän työtunnit viikoittain	29
7.4 Niko Monosen työtunnit viikoittain	29
7.5 Veli-Mikko Puupposen työtunnit viikoittain	29
7.6 Ilkka Rautiaisen työtunnit viikoittain	29
7.7 Atter Söderlundin työtunnit viikoittain	29
8 Riskit ja niiden hallinta	31
8.1 Riskien todennäköisyydet ja haitat	31
8.2 Alustan rajoitukset	32
8.3 Kehittäjien tietotaidon puutteet	32
8.4 Tavoitteiden rajaaminen	33
8.5 Tiedotuksen puutteet	34
8.6 Jäsenten poissaolot	34
8.7 Kokemattomuus projektihallinnassa	35
9 Jäsenten kokemuksia ja oppimaa	36
9.1 Ryhmän kokemuksia	36
9.2 Niko Monosen kokemuksia ja oppimaa	36
9.3 Veli-Mikko Puupposen kokemuksia ja oppimaa	37
9.4 Ilkka Rautiaisen kokemuksia ja oppimaa	37
9.5 Atte Söderlundin kokemuksia ja oppimaa	37
10 Yhteenveto	38
11 Lähteet	39

1 Johdanto

Projekti suoritti Sovellusprojekti-kurssilla keväällä 2014 alustavan tarvekartoituksen ja kehitti prototyypin uudenlaisesta hätäilmoitusjärjestelmästä, joka mahdollistaa yhteyden hädässä olevien ihmisten ja hätäkeskusten välillä. Keskeinen ajatus on hyödyntää matkapuhelinten merkittävästi lisääntyntä datan lähetyksen ja vastaanotokykyä osana auttamisprosessia. Tavoitteena kehitetyllä tietojärjestelmällä oli luoda aikaisempaa tarkempi kuva hätätilanteesta entistä lyhyemmässä ajassa. Tämä auttaa pelastamaan ihmishenkiä, parantamaan palvelun laatua ja tehostamaan resurssien käyttöä. Projektissa kehitetyn hätäilmoitusjärjestelmän prototyypin avulla pyritään osoittamaan teknologiset mahdollisuudet ja haasteet sekä havainnollistamaan käyttöliittymällä avuntarvitsijan ja hätäkeskuksen vuorovaikutusta.

Projektiraportissa kuvataan projektin toteutunutta läpivientiä esitellen sen tulokset, resurssit, käytänteet, prosessimallia, aikataulua sekä riskien hallintaa. Lisäksi käsitellään projektiryhmän jäsenten tehtäviä, työmääriä ja tehtäväjakoja projektissa. Projektiraportin laatimisessa on hyödynnetty Paatti-projektin projektiraporttia [2] ja Sovellusprojektien ohjetta [8]. Projektissa laaditussa vaatimusmäärittelyssä [6] on kuvattu kehitetylle sovellukselle asetettuja vaatimuksia ja niiden toteumaa. Sovellusraportissa [7] on kuvattu sovelluksen käyttöliittymää, toteutusratkaisuja, havaittuja ongelmia ja jatkokehitysideoita. Projektin muut tulokset on esitelty luvussa 3.4.

Luvussa 2 kuvataan dokumentissa käytetyt termit ja niiden merkitys. Luvussa 3 kuvataan projektin taustoja sekä tavoitteiden, tulosten ja projektiryhmän jäsenten oppimistavoitteiden toteutumista. Luvussa 4 esitellään projektiorganisaatio ja resurssit. Luvussa 5 kuvataan projektin käytänteet. Luvussa 6 tarkastellaan projektiryhmän tehtäviä sekä tehtävien työmääriä ja työnjakoa. Luvussa 7 kuvataan projektin ohjelmistokehitys prosessia ja aikataulua. Luvussa 8 kuvataan projektiin liittyneiden riskien hallintaa sekä käsitellään niiden syitä ja vaikutusta projektin läpivientiin ja tuloksiin. Luvussa 9 esitellään ryhmän jäsenten kokemuksia ja oppimaa projektista.

2 Termit

Dokumentin aihealueen termejä ovat seuraavat:

EKG	on elektrokardiogrammi eli sydänsähkökäyrä.
Hätäkeskus	vastaanottaa ja käsittelee yleiseurooppalaiseen hätänumeroon 112 soitetut hätäpuhelut.
Käyttöliittymä	on se ohjelmiston, laitteen tai minkä tahansa muun tuotteen osa, jonka kautta käyttäjä käyttää tuotetta.
Mobiilisovellus	on sovellus, joka on suunniteltu mobiilipäätelaitteisiin eli matkapuhelimiin, älypuhelimiin, kämmentietokoneisiin ja ammattikäyttöön tarkoitettuihin mobiilipäätelaitteisiin.
Palvelin	on palvelinsovellusta suorittava tietokone.
Palvelinsovellus	on ohjelma, joka tarjoaa erilaisia palveluja muille ohjelmille joko tietokoneverkon välityksellä tai paikallisesti samassa tietokoneessa.
Prototyyppi	tarkoittaa sovelluksen tai tietojärjestelmän versiota, jolla on tarkoitus varmistaa konseptin tekninen toimivuus ja/tai kartoittaa käyttäjien tarpeita.
Tietojärjestelmä	on ihmisistä, tietojenkäsittelylaitteista, tiedonsiirtolaitteista ja ohjelmistoista koostuva järjestelmä, jolla tehostetaan tai helpotetaan jotain tietojen käsittelyn toimenpiteitä tai tehdään se ylipäättään mahdolliseksi.
Tietokanta	on tietotekniikassa käytetty termi tietovarastolle. Se on koelma tietoja, joilla on yhteys toisiinsa.

Dokumentissa esiintyviä teknisiä termejä ovat seuraavat:

Alusta	koostuu käyttöjärjestelmästä ja muista tietojärjestelmän käyttämistä varusohjelmistoista.
Doxygen	on luokkadokumentointien generointiohjelma C#-lähdekooditiedostoille.

FreeMind	on ajatuskarttasovellus.
GanttProject	on ajan- ja resurssienhallintaohjelma projekteille.
Git	on hajautettu versiohallintajärjestelmä.
LaTeX	on ladontaohjelmisto.
Lähdekoodi	on tekstimuotoista ohjelmointikielistä listausta.
Trac	on hyvin yksinkertainen WWW-pohjainen alun perin ohjelmistoprojektien tehtävien hallintaan tarkoitettu tietojärjestelmä.
WP8	on lyhenne sanasta Windows Phone 8.
YouSource	on Git-versiohallintaohjelmistoa tukeva lähdekoodien julkistusjärjestelmä.

3 Taustaa ja tavoitteiden toteutuminen

Luvussa käsitellään projektin taustatietoja, projektissa toteutetulle tietojärjestelmän prototyypille ja muille tuloksille asetettujen tavoitteiden toteutumista sekä ryhmän jäsenille asetettuja oppimistavoitteiden toteutumista. Projektin tilaajan prototyypille asettamat välttämättömät ja tärkeät vaatimukset saatiin toteutettua. Projektin vaatimusmäärittelyssä[6] projektissa toteutettaviksi vaatimuksiksi rajatuista 142 vaatimuksista kaksi jäi keskeneräiseksi ja yhtä ei toteutettu. Nämä kolme vaatimusta sovittiin jatkokehitykseen. Toteutetulla prototyypillä pystytään osoittamaan teknologiset mahdollisuudet uudelleenlaiselle hätäilmoitusjärjestelmälle ja sillä voidaan havainnollistaa järjestelmän toimintaa.

Projektissa kohdatut haasteet, niiden ratkaisut sekä projektiryhmän laatimat sovel-lusraportti ja vaatimusmäärittely antavat hyvät valmiudet järjestelmän jatkokehi-tykseen.

3.1 Taustaa

Matkapuhelinteknologia on kehittynyt huomasti kuluneen vuosikymmenen aikana. Nykyisin Suomessa jokaiseen puhelinliittymään kytketään automaattisesti mahdol-lisuus tietoliikennedatan siirtämiseen, ja käytännössä suurin osa tänä päivänä myy-tävistä matkapuhelimista on varustettu "älyominaisuuksilla". Älypuhelimien laaja kattavuus tarjoaa luonnollisen ja tehokkaan tavan kehittää perinteistä hätäpuhelua nykyaikaan.

Hätäkeskuslaitos on ensimmäinen viranomaislenkki auttamisen ketjussa. Hätäkes-kuksen tehtävänä on puhelun aikana luoda vakioidulla menettelyllä riskiarvio ti-lanteesta ja välittää tehtävä asianomaiselle viranomaiselle. Tänä päivänä hätäpuhe-lu on rajoittunut puheyhteyden hätäkeskuksen päivystäjän kanssa. Pelkän puheyh-tyden avulla kokonaisarvion tekeminen tilanteesta voi olla joskus erittäin haasta-va, joten virhearvioita ja niiden seurauksia voidaan lukea päivälehdistä aika-ajoin.

Virhearviot voivat aiheuttaa seuraavanlaisia ongelmia:

- Tilannetta tai riskiä ei tunnisteta, jolloin viranomaisapu viivästyy tai jää saa-pumatta, ja pahimmillaan menehtyy ihmishenkiä.
- Tilanne tai riski yliarvioidaan, josta seuraa viranomaisresurssien epätarkoituk-sen mukainen käyttö johtaen mm. seuraaviin ongelmiin:

- Vaste on liian suuri tehtävän todelliseen riskiin nähden.
- Hälytysajojen lisääntyminen lisää liikenne-riskiä.

Hälyri-projektina aloitetun hankkeen tavoitteena on osoittaa uudenmallisten datayhteyksien laaja ja innovatiivinen hyödynnettävyys hätätilanteiden arvioinnissa ja hallinnassa. Hätäpuhelun aikana älypuhelimesta voitaisiin lähettää hätäkeskukseen ääni- ja videokuvaa, paikkatietoja sekä EKG- ja muuta anturidataa tukemaan riskiarvion tekemistä. Hätäkeskus voisi lähettää toimintaohjeita esimerkiksi videoon tai kuvina hädässä olevalle. Hätäkeskus voisi myös siirtää yhteyden suoraan auttajille.

Jyväskylän yliopiston tietotekniikan laitoksella on Sapporo-projektin kautta laaja kokemus julkisen hätäviestijärjestelmän kehittämisestä ja pilotoinnista. Siitä saadun tiedon perusteella vastaava hätäilmoitusjärjestelmän pilotointi voitaisiin luotettavasti suorittaa yliopiston hankkeena. Hankkeessa kehitettävä tietojärjestelmä voisi toimia hätätilanteessa lisäapuna perinteisen 112-hätäpuhelun rinnalla.

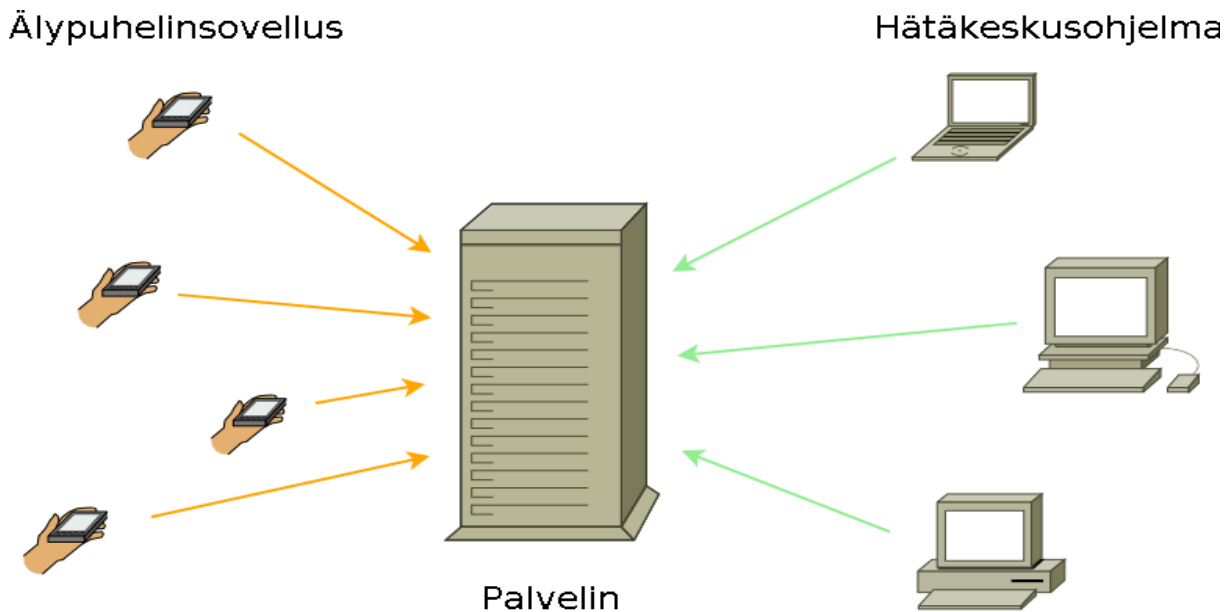
Hankkeessa kehitettävän tietojärjestelmän valmistuttua jokainen kansalainen voisi ilmaiseksi ladata siihen sisältyvän mobiilisovelluksen omalle puhelimelleen niin halutessaan. Toiminta perustuu siis keskeisesti vapaaehtoisuuteen, eikä sovellus ensimmäisessä vaiheessa tule vaikuttamaan perinteisen 112-hätäpuhelun toimintaan ja siihen liittyviin säädöksiin.

3.2 Prototyypille asetettujen tavoitteiden toteutuminen

Tilaaajan edustajat olivat määritelleet projektin alussa prototyypin kokonaistavoitteet, mutta tarkemmat vaatimukset selvitti projektiryhmä laatimalla kattavan vaatimusmäärittelyn. Vaatimusmäärittelyssä [6] vaatimukset priorisoitiin kuusiportaiselle asteikolle, jotka olivat välttämätön, tärkeä, projektissa tärkeä järjestelmässä mahdollinen, mahdollinen, idea ja ei toteuteta. Projektiryhmä kirjasi 105 vaatimusta hätäkeskusohjelmalle, 61 vaatimusta palvelimelle ja 115 vaatimusta älypuhelinsovellukselle. Projektissa toteutettaviksi vaatimuksiksi rajattiin vain tasojen välttämätön, tärkeä ja projektissa tärkeät vaatimukset. Toteutettavia vaatimuksia oli siten 25 hätäkeskusohjelmalle, 19 palvelinsovellukselle ja 98 älypuhelinsovellukselle.

Projektiryhmä täydensi vaatimusmäärittelyä uusilla vaatimuksilla, mutta projektissa toteutettaviksi rajattujen vaatimusten prioriteetteja ei jouduttu muuttamaan. Projektin aikana toteutettiin lähes kaikki projektissa toteutettavaksi rajatut vaatimukset.

set. Vain kaksi vaatimusta jäi keskeneräiseksi ja yksi toteuttamatta. Projektissa sovellukseen kehitetyistä vaatimuksista toteutuneiksi on laskettu vaatimuskokonaisuudet, joiden kaikki välttämättömät vaatimukset on toteutettu.



Kuva 3.1: Projektissa kehitetyn hätäilmoitusjärjestelmän prototyyppi.

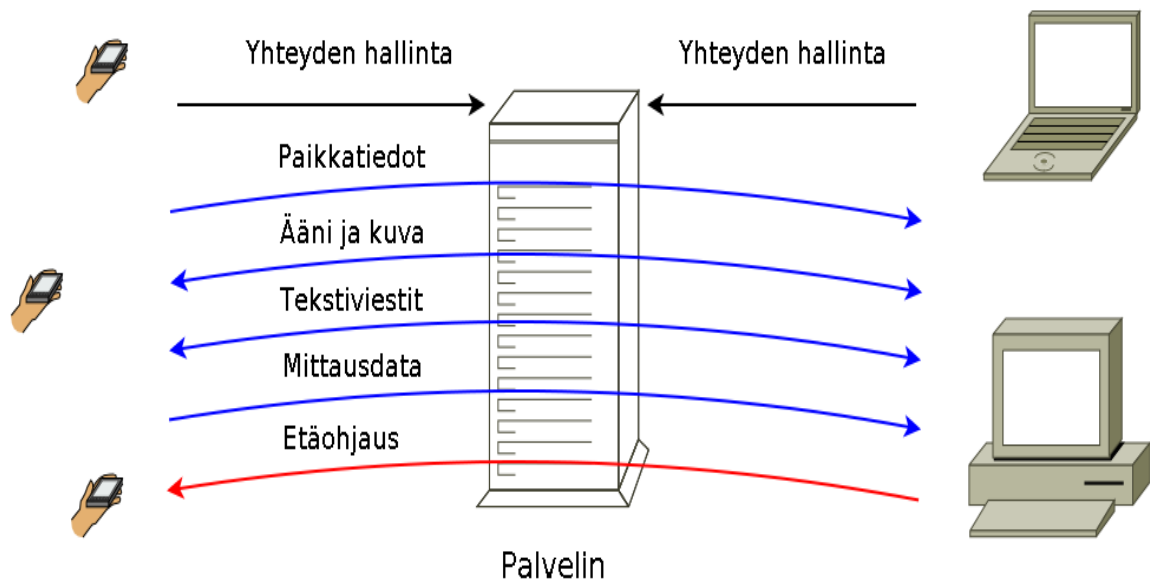
Toteutetun tietojärjestelmän prototyypin älypuhelinsovelluksella voidaan välittää ääntä ja videokuvaa. Lähettää automaattisesti paikka- ja henkilötietoja, sekä välittää muuta tietoa kuten esimerkiksi fysiologista mittausdataa (EKG) hätäkeskussovellukseen palvelimen kautta. Älypuhelinsovellus huolehtii myös tarvittavasta kaistastaan huonoilla yhteyksillä.

Hätäkeskuksen henkilökunnan käyttöön kehitetyllä sovelluksella voidaan ottaa hätäilmoituksia käsiteltäväksi, jolloin sovelluksen käyttöliittymässä esitetään älypuhelimesta lähetetyt tiedot. Hätäilmoitussovelluksella voidaan pyytää mobiilisovelluksesta tietoja, kuten esimerkiksi fysiologista mittausdataa. Kehitetyllä prototyypillä voidaan siirtää vain generoitua fysiologista mittausdataa, koska projektiryhmä ei saannut käyttöönsä EKG-mittauslaitetta. Hätäkeskussovelluksella voidaan ohjata älypuhelinsovelluksen toimintoja, kuten esimerkiksi pyytää videokuvaa ja tarkentamaan sijainti.

Hätäkeskussovellukselle toivottua ominaisuutta lähettää toimintaohjeita, kuvia ja videokuvaa mobiilisovellukselle ei toteutettu. Tietojärjestelmään toteutettu palve-

Älypuhelinsovellus

Hätäkeskusohjelma



Kuva 3.2: Prototyypin välittämät tiedot.

linsovellus huolehtii hätäpuheluiden käsittelystä ja välittämisestä, muttei tallentamisesta tietokantaan tai tiedostoon.

Toteutetulla prototyypillä pystytään osoittamaan teknologiset mahdollisuudet uudennlaiselle hätäilmoitusjärjestelmälle ja sillä voidaan havainnollistaa järjestelmän toimintaa. Projektissa kohdatut haasteet ja niiden ratkaisut antavat hyvät lähtökohdat jatkokehitykselle. Projektissa kehitetty tietojärjestelmän prototyyppi on suunniteltu jatkokehitystä silmälläpitäen. Sovellusraportissa [7] on kuvattu tarkemmin prototyyppiä ja käytettyjä toteutusratkaisuja.

3.3 Oppimistavoitteiden toteutuminen

Ryhmän jäsenet olivat asettaneet seuraavia henkilökohtaisia oppimistavoitteita:

- Niko Mononen halusi oppia ohjelmistoprojektin johtamista.
- Veli-Mikko Puupponen halusi kokemusta projektimuotoisesta työtavasta ja ryhmätyöskentelystä.
- Ilkka Rautiainen halusi kehittää ohjelmointiosaamistaan.
- Atte Söderlund halusi oppia projektimuotoista työtapaa kokonaisuudessa.

Henkilökohtaiset oppimistavoitteet toteutuivat jokaisen ryhmän jäsenen osalta. Ryhmän jäsenet ymmärsivät hyvin oman roolin ja vastuun merkityksen sekä ajankäytön hallinnan merkityksen. Projektin varsinaisen läpiviennin lisäksi opiskelijat oppivat yhteistyötä ryhmän sisällä. Työskenteleminen ryhmässä vaati vastuunottamista ja -antamista. Pienryhmässä toimiessa myös erilaisiin ongelma- ja ristiriitatilanteisiin oli varauduttava. Näitä ongelmatilanteita ratkoessaan ryhmän jäsenet oppivat myös työelämässä tarvittavia kykyjä. Kukin ryhmän jäsen on analysoinut oppimaansa tarkemmin luvussa 9.

3.4 Tulokset ja niiden toteutuminen

Tietojärjestelmän ohella projektiryhmä toteutti seuraavat dokumentit:

- **Ajankäyttöraportti** sisältää ryhmän jäsenten kirjaamat työtunnit sekä niiden jakautumisen eri tehtäväkokonaisuuksille ja tehtäville.
- **Asennusohjeet** sisältää tietojärjestelmän eri sovellusten asennusohjeet.
- **Esittelymateriaali** sisältää väli- ja loppuesittelyn materiaalit ja muistiot.
- **Esittelyviesti** pelastusalan asiantuntijoille. Viesti sisälsi lyhyen kuvauksen projektista ja siinä kehitetystä prototyypistä.
- **Esittelyvideo** kehitetystä prototyypistä.
- **Itsearvioinnit** sisältävät ryhmän jäsenten arvioinnit omasta toiminnasta, onnistumisesta, kokemuksista ja oppimisesta.
- **Lisenssisitoumuksella** ryhmän jäsenet vahvistavat sitoutuvansa sijoittamaan toteuttamansa lähdekoodin valitun avoimen lähdekoodin lisenssin alaisuuteen.
- **Luokkadokumentaatio** sisältää lähdekoodista automaattisesti generoidut luokkia kuvaavat dokumentit.
- **Lähdekoodi** sisältää lähdekoodin kommentteineen.
- **Palaverien dokumentit** sisältävät palaverien esityslistat, pöytäkirjat ja tilakatsaukset.
- **Projektiraportti** kuvaa projektin toteutunutta läpivientiä ja asetettujen tavoitteiden saavuttamista sekä jäsenten kokemuksia ja oppimaa.
- **Projektisuunnitelma** kuvaa projektin suunniteltuja tavoitteita, resursseja, käytänteitä, tehtävien työmääriä ja työnjakoa, aikataulua sekä riskien hallintaa.

- **Sovellusraportti** kuvaa toteutetun sovelluksen kokonaisrakenteen ja toiminnot, puutteelliset ja heikot toteutusratkaisut sekä jatkokehitysideat.
- **Sähköpostiarkistot** sisältävät kaikki projektin sähköpostilistoilla käytyt keskustelut.
- **Vaatimusmäärittely** kuvaa projektin toiminnalliset ja tekniset vaatimukset.

Projektiryhmä laati kaikki suunnitellut dokumentit. Laadituista tuloksista esittelyviesti ja esittelyvideo eivät sisältyneet alunperin suunniteltuihin tuloksiin.

4 Organisaatio ja resurssit

Luvussa esitellään projektiorganisaatio, käytössä olleet resurssit sekä projektiin liittyvät oheiskurssit ja perehdytykset. Projektisuunnitelmaan kirjattu projektiorganisaatio ja resurssit toteutuivat, eikä merkittäviä muutoksia tullut.

4.1 Projektiorganisaatio

Projektiryhmän muodostivat Niko Mononen, Veli-Mikko Puupponen, Ilkka Rautiainen ja Atte Söderlund. Niko Monosella oli aiempaa ohjelmointikokemusta työelämästä ja omien ohjelmistoprojektien kautta. Veli-Mikko Puupposella oli myös ohjelmointikokemusta yliopiston ohjelmointikurssien ja omien ohjelmistoprojektien kautta. Hänellä oli myös paljon tietämystä erilaisista tietojärjestelmistä, mistä oli hyötyä sovelluksen suunnittelu- ja toteutusvaiheessa. Ilkka Rautiaisella oli ohjelmointitaitoja yliopiston ohjelmointikursseilta. Atte Söderlundilla oli aiempaa kokemusta mobiiliohjelmoinnista, mikä osoittautui hyödylliseksi projektissa.

Tilaaajan edustajina toimivat Tero Tuovinen ja Vesa Lappalainen tietotekniikan laitokselta. Jukka-Pekka Santanen toimi projektin vastaava ohjaaja. Teknisiä ohjaajia olivat Olli Kauppinen ja Jaakko Kosonen. Kosonen toimi ensisijaisena ohjaajana tietojärjestelmän toteutusratkaisuihin liittyvissä asioissa. Kauppisella oli tietämystä kohdealueesta ja älypuhelinsovellusten kehittämiseen liittyvää aiempaa kokemusta. Projektin sidosryhmiin kuului myös hätäkeskusten, pelastusalan ja ensihoidon asiantuntijoita, joista projektiin osallistuivat Helena Jäntti Kuopion yliopistollisesta sairaalasta ja Tapio Kettunen Kuopion hätäkeskuksesta.

Jyväskylän yliopiston ATK-lähituki vastasi ryhmän käytössä olevista laitteista ja ohjelmistoista. Projektiin kuuluivat viestintäkurssin kirjoitusviestinnän opettajaista Timo Nurmi ja puheviestinnän opettajista Hanna Kivimäki. Johanna Silvennoinen piti projektiryhmälle käytettävyysspäivän ja antoi palautetta prototyypin käyttöliittymistä.

Projektisuunnitelmaan kirjattu projektiorganisaatio toteutui. Helena Jäntin osallistuminen projektin läpivientiin oli tosin vähäistä. Tekniset ohjaajat ja tilaaajan edustajat olivat kuitenkin hyvin tavoitettavissa. Projektiryhmä olisi kenties voinut tukeutua hieman enemmän teknisiin ohjaajiin projektin aikana.

4.2 Projektin tilat ja laitteet

Projektiryhmä sai käyttöönsä tietotekniikan laitokselta Agorasta huoneen C223.2, jossa oli asennettuina neljä Windows 8 -työasemaa. Projektiryhmällä oli käytössä myös palvelin ja Lappalaisen projektiryhmälle lainaama Windows Phone 8 -puhelin.

Sovellusprojektien avotilassa oli ryhmän käytössä monitoimitulostin. Ryhmän jäsenet pystyvät tulostamaan projektiin liittyvät dokumentit ilman maksuja. Ryhmällä oli mahdollisuus varata käyttöönsä videoprojektori, kannettava PC ja digitaalisanelin. Projektin aikana näistä tarvittiin vain kannettavaa tietokonetta. Projektiryhmän jäsenillä oli käytössä myös omia Windows Phone -puhelimia ja Veli-Mikko Puuppon digitaalisanelin. Projektiryhmällä oli käytettävissä virkistystila, jossa oli vedenkeitin ja kahvinkeitin. Tietotekniikan laitos tarjosi ryhmälle kahvit, teet ja mehut.

Projektilla oli käytössään yhteinen verkkolevy ja WWW-sivusto projektin tiedostojen säilytystä varten. Verkkolevy oli hakemistossa `\\sovpa7.cc.jyu.fi\halyri` ja sivusto osoitteessa `http://sovellusprojektit.it.jyu.fi/halyri`.

Projektin julkistetut dokumentit sijoitettiin trac-wikiin URL-osoitteeseen `https://trac.cc.jyu.fi/projects/sovproj/wiki/Halyri`. Trac on hyvin yksinkertainen WWW-pohjainen, alun perin ohjelmistoprojektien tehtävien hallintaan tarkoitettu järjestelmä. Lähdekoodi sijoitettiin Git-pohjaiseen YouSource-julkistusjärjestelmään osoitteeseen `https://yousource.it.jyu.fi/halyri/`.

Julkistetut dokumentit sijoitettiin pääosin suunnitellusti, vaikka joitakin dokumentteja sijoitettiin myös Google Drive -palveluun.

Projektilla oli käytössä sähköpostilistat `halyri@korppi.jyu.fi` ja `halyri_opetus@korppi.jyu.fi`. Sähköpostilistojen arkistoista `https://korppi.jyu.fi/kotka/servlet/list-archive/halyri/` ja `https://korppi.jyu.fi/kotka/servlet/list-archive/halyri_opetus/` keskustelua voi lukea WWW-selaimella kirjautumalla Korppi-opintotietojärjestelmään.

4.3 Dokumentointityökalut

Projektisuunnitelma ja projektiraportti laadittiin \LaTeX -ladontaohjelmistolla. Microsoft Office -toimisto-ohjelmistolla laadittiin palaverien esitysgrafiikat ja lisenssisi-
tous. Vaatimusmäärittelyn tekemisessä käytettiin FreeMind-ajatuskarttaohjelmaa.

Palaverien pöytäkirjat laadittiin joko Microsoft Word- ohjelmistolla tai \LaTeX -ladontaohjelmistolla. Projektin aikataulu laadittiin GanttProject-ohjelmalla ja tehtävien työtunnit Excel-taulukkolaskentaohjelmalla. Raakatekstimuodossa laadittiin muut tekstidokumentit, kuten esityslistat. Luokkadokumentit generoitiin lähdekoodista Doxygen-työkalulla.

Ajankäytön kirjaamiseen ja seurantaan ryhmällä oli käytössään Petri Heinosen sovellusprojekteille toteuttama Excel-sovellus [1]. Lisäksi kyseisestä sovelluksesta laadittiin tilakatsauksissa vaadittavat esitysgrafiikat.

Dokumentointityökalujen käyttö toteutui suunnitellusti. Ajankäyttöseurantasovelluksen kanssa oli aluksi hieman hankaluuksia, ja se vaikeutti hieman projektin seurantaa, mutta ei vaikuttanut projektin tuloksiin.

4.4 Ohjelmointityökalut

Sovellukset toteutettiin Microsoftin Visual Studio 2013 -ohjelmalla ja Windows Phone SDK -kehitystyökaluilla [4]. Käytettyjä kirjastoja ovat speex, SignalR sekä Microsoftin ASP.NET Framework ja Entity Framework. Ohjelmointikielenä käytettiin Microsoftin C#:a.

Suunniteltua tietokannanhallintajärjestelmää ei käytetty projektissa, koska prototyypille ei kehitetty tietokantaa. Käytetyt ohjelmointityökalut eivät olleet ryhmän jäsenille ennestään tuttuja, joten niihin tutustumiseen käytettävä aika oli huomioitu projektisuunitelmassa. Työkaluja opittiin kuitenkin käyttämään hakemalla tietoa Internetistä ja teknisten ohjaajien avustuksella.

4.5 Luennot ja perehdytykset

Projektin rinnalla järjestettiin kaksi oheiskurssia: *Sovellusprojektin hallintaa, viestintää ja työkaluja* sekä *Projektiviestintä IT-alalla*. Oheiskurssien työtunnit kirjattiin omalle tehtäväkokonaisuudelle työajanseurantasovelluksessa.

Kurssiin *Sovellusprojektin hallintaa, viestintää ja työkaluja* sisältyivät seuraavat luennot:

- aloitusluento,
- vaatimusmäärittely,

- projektin johtaminen ja hallinta,
- käytettävyysspäivä,
- tekijänoikeus ja sopimukset sekä
- versiohallinta.

Kurssiin *Projektiviestintä IT-alalla* kuuluivat puhe- ja kirjoitusviestinnän luentojen ja ryhmätöiden ohella projektissa laadittujen dokumenttien kirjoitusasun ja rakenteen muokkauksen työtunnit. Projektin aikana järjestetyt kaksi väliesittelyä kuuluivat myös viestintäkurssiin.

Projektin aikana pidettiin kaikki projektisuunnitelmaan kirjatut luennot ja opetus-tapaamiset.

5 Projektin käytänteet

Luvussa kuvataan projektissa noudatettuja käytänteitä. Nämä käytänteet edesauttoivat projektin läpivientiä, asetettujen tavoitteiden saavuttamista ja tulosten toteuttamista. Käytänteillä varmistettiin, että projekti etenee aikataulussa sekä projektin aikana toteutetut tulokset ovat korkealaatuisia ja keskenään yhteneviä.

Käytänteet poikkesivat suunnitellusta tiedostojen nimeämisen, testauksen ja tulosten luovuttamisen kohdalta. Tiedostojen nimeämisen käytänteet eivät toteutuneet projektiryhmän sisäisissä versioissa. Suunniteltua lähdekoodiin kirjoitettua yksikötestausta tehtiin hyvin vähän sillä testattavia funktioita oli vain muutamia. Testaussuunnitelma oli samalla testausraportin dokumenttipohja ja tilaajan edustaja halusi tulokset USB-tikulla CD-levyn sijaan.

5.1 Tiedotus

Projektin tiedotuksesta projektiorganisaation sisällä vastasi ensisijaisesti projektipäällikkö. Ryhmän jäsenet kommunikoivat ensisijaisesti projektipäällikön kanssa, mutta välittivät kuitenkin vastuualueillaan oleelliset asiat eteenpäin koko projektiorganisaatiolle. Projektiorganisaatiolle ryhmä tiedotti vähintään projektissa kohdatuista haasteista ja vaihtoehtoista sekä tehdyistä valinnoista, ratkaisuista ja päätöksistä.

Projektiorganisaation sisäinen sähköpostilista `halyri@korppi.jyu.fi` oli tarkoitettu tilaajien edustajille ja ohjaajille suunnattuun tiedotukseen. Sähköpostilistalle kuuluvat kaikki projektiorganisaation jäsenet. Postilistalle lähetetyt viestit tallentuivat sähköpostiarkistoon, joka löytyy osoitteesta `https://korppi.jyu.fi/kotka/servlet/list-archive/halyri`.

Projektiryhmän jäsenten ja ohjaajien käytössä oli myös sähköpostilista `halyri_opetus@korppi.jyu.fi`. Sen sähköpostiarkisto sijaitsee osoitteessa `https://korppi.jyu.fi/kotka/servlet/list-archive/halyri_opetus/`. Sähköpostilistalla käsiteltiin sellaisia asioita, jotka eivät olleet merkityksellisiä tilaajalle. Tällaisia asioita olivat esimerkiksi projektiryhmän sisäiset palaverit sekä erilaiset opetukseen ja koulutukseen liittyvät asiat.

Ryhmän sisäinen tiedotus tapahtui pääosin suullisesti, sillä ryhmän jäsenet työsken-

telivät samassa tilassa ja olivat yhteydessä toisiinsa. Tarvittaessa tiedotus ja keskustelu voitiin ryhmän sisällä hoitaa sähköpostitse.

Toteutunut tiedotus vastasi suunnitelmaa. Projektiryhmä lähetti kysymyksiä ja tarkennuksia sähköpostilistoille. Projektiryhmä keskusteli projektihuoneessa projektin läpivientiin sekä sovelluksen kehitykseen ja suunnitteluun liittyvistä asioista. Projektiryhmän jäsenillä oli eriävät työajat, joten keskustelua käytiin myös paljon sähköpostilla ja pikaviestimillä.

Projektin vastaava ohjaaja olisi toivonut säännöllisempää ja aktiivisempaa tiedotusta palaverien välillä. Valituista toteutusratkaisuistakin olisi voinut tiedottaa enemmän palaverien ulkopuolella.

5.2 Palaverit

Projektiorganisaatio pyrki pitämään yhteisen palaverin kerran kahdessa viikossa. Seuraavan palaverin ajankohta päätettiin aina edellisessä palaverissa.

Palavereissa käsiteltiin edellisen palaverin jälkeen tapahtuneita projektin etenemiseen vaikuttaneita asioita sekä tulevia toimenpiteitä ja tarvittavia päätöksiä. Jokaisessa palaverissa käytiin läpi edellisen palaverin pöytäkirjaan merkityt päätökset sekä osallistujille osoitetut toimenpiteet ja niiden tila. Palavereissa projektipäällikkö esitti tilakatsauksen, jossa esitettiin projektiryhmän viikoittainen ajankäyttö eri tehtäväkonaisuuksittain.

Palavereissa käsiteltiin myös toteutettavan sovelluksen ominaisuuksia ja vaatimuksia, sekä niiden toteutusratkaisuja. Palavereissa käsitellyt asiat pyrittiin käymään läpi niin perusteellisesti, että asiakkaan edustajat ja projektiryhmän jäsenet ymmärsivät asiat samalla tavalla, eikä väärinymmärryksiä päässyt syntymään. Jos projektiryhmällä oli esittää sovelluksesta konkreettisia käyttöliittymään liittyviä demoja tai prototyyppejä, ne esiteltiin palavereissa. Palavereissa sovittiin myös projektiin liittyvistä käytänteistä.

Ryhmän jäsenet toimivat vuorollaan palavereissa sihteerinä tai puheenjohtajana siten, että kyseiset tehtävät kiersivät ryhmän keskenään sopimassa järjestyksessä. Puheenjohtaja johti keskustelua ja piti huolen siitä, että palaveri eteni esityslistan osoittamalla tavalla. Sihteerin laati palaverista pöytäkirjan, jonka hän toimitti myöhemmin palaverin puheenjohtajalle tarkastettavaksi. Kun pöytäkirja oli puheenjohtajan

osalta hyväksytty, puheenjohtaja tai sihteeri toimitti sen koko projektiorganisaatiolle. Tämän lisäksi jokaisen ryhmän jäsenen laatima ensimmäinen pöytäkirja toimitettiin myös vastaavan ohjaajan ja kirjoitusviestinnän opettajan esitarkastettavaksi. Pöytäkirja hyväksyttiin seuraavassa palaverissa, ja siihen voitiin tällöin esittää muutoksia.

Palaverien suunnitellut käytänteet toteutuivat projektissa. Muutamien palaverien pöytäkirjojen laatiminen ja tarkastaminen viivästy useita päiviä. Tämä johtui ryhmän jäsenten kokemattomuudesta laatia pöytäkirjoja ja muiden töiden kasautumisesta.

5.3 Tiedostojen nimeäminen

Lähdekooditiedostojen nimeämisessä käytettiin C#-kielen yleisiä käytänteitä. Tiedostojen ja hakemistojen nimet kirjoitettiin englanniksi. Lisäksi tiedostonimet kirjoitettiin aina pienillä kirjaimilla ja välilyönnit korvattiin alaviivoilla (_). Lähdekoodissa ja dokumentoinnissa käytettyjen tekstitiedostojen tallennusmerkistönä käytettiin ensisijaisesti UTF-8-koodausta.

Dokumenttitiedostot nimettiin projektin nimen ohella niiden sisältöä kuvaavilla nimillä. Julkistettaessa nimiin lisättiin versionumero luvussa 5.6 esitettyjen käytänteiden mukaisesti. Esimerkiksi projektisuunnitelman pdf-tiedosto nimettiin `halyri_projektisuunnitelma_[numero].[numero].[numero].pdf`

Projektin aikana julkaistut tiedostot nimettiin pääsääntöisesti suunnitellun mukaisesti. Ryhmän sisäisessä nimeämisessä ja numeroinnissa poikettiin kuitenkin suunnitellusta.

5.4 Hakemistorakenne

Projektin tulokset tallennettiin CD-levylle ja projektin WWW-hakemistoon päähakemiston alle seuraavan hakemistorakenteen mukaisesti:

```
dokumentit
  ajankaytto
  esittelyt
```



```
    itsearvioinnit
    projektiraportti
    projektisuunnitelma
    sovellusraportti
    vaatimusmaarittely
    lisenssisitoumus
palaverit
    esityslistat
    materiaalit
    poytakirjat
    tilakatsaukset
    katselmoinnit
testaus
    testausraportit
    testaussuunnitelmat
sahkopostiarkistot
    halyri
    halyri_opetus
software
    class_documentation
    installation_documents
    installation_files
    source_code
esittelyviesti
esittelyvideo
```

Hakemistorakenne toteutui suunnitellusti.

5.5 Lähdekoodi

Projektissa toteutettu lähdekoodi sijoitettiin YouSource-julkistusjärjestelmään. Sovelluksen lähdekoodi kirjoitettiin noudattaen yleisiä C# -koodin käytänteitä [5]. Lähdekoodin kommentoinnissa käytettiin C#:n XML-tyylistä käytäntöä. Kommentoinnissa käytettiin XML-tyyliä, jotta lähdekoodista voitiin generoida luokkadokumentaatio Doxygen-ohjelmalla. Lähdekoodissa käytetyt aliohjelmat, luokat ja muuttujat nimettiin mahdollisimman kuvaavilla englanninkielisillä nimillä. Myös koko lähde-

koodin kommentointi toteutettiin englanniksi. Usean sanan nimet kirjoitettiin CamelCasena ja lähdekoodiin sallittiin enintään 80 merkkiä riviä kohden.

Seuraavassa on esimerkki edellä esitettyjen käytänteiden mukaisesta C# -koodista.

```
/// Copyright (C) 2014 Veli-Mikko Puupponen, Ilkka Rautiainen
///
/// The MIT License (MIT)
///
/// Permission is hereby granted, free of charge, to any person obtaining a copy
/// of this software and associated documentation files (the "Software"), to
/// deal in the Software without restriction, including without limitation the
/// rights to use, copy, modify, merge, publish, distribute, sublicense,
/// and/or sell copies of the Software, and to permit persons to whom the
/// Software is furnished to do so, subject to the following conditions:
///
/// The above copyright notice and this permission notice shall be included in
/// all copies or substantial portions of the Software.
///
/// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
/// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
/// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
/// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
/// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
/// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
/// IN THE SOFTWARE.

using System;
using HalyriServer.Model.Transfer;
using HalyriServer.Services.Exceptions;

namespace HalyriServer.Controllers
{
    /// <author>Veli-Mikko Puupponen, Ilkka Rautiainen</author>
    /// <summary>
    /// The singleton class manages active emergency connections (Connection).
    /// It contains a dictionary for active connections and their
    /// identifying GUIDs. All emergency connection states and life cycle
    /// methods are handled by the class. It functions as a broker between
    /// the mobile emergency connection client and the call center client.
    /// </summary>
    public class ConnectionController
    {
        private static readonly Lazy<ConnectionController> instance =
```

```
        new Lazy<ConnectionController>(() => new ConnectionController());
private DatabaseController databaseController;

/// <summary>
/// Returns the ConnectionController instance.
/// </summary>
public static ConnectionController Instance
{
    get { return instance.Value; }
}

/// <summary>
/// The method generates a new GUID. It currently uses
/// Guid.NewGuid() of .NET framework. No collision checking
/// is performed. All-zero GUIDs are not returned.
/// </summary>
/// <returns>A new GUID formatted as a string.</returns>
/// <example>
/// <pre name="test">
/// GetNewGuidStrig().length() > 0 === true;
/// GetNewGuidStrig().length() < 25 === true;
/// GetNewGuidStrig().length() == 25 === false;
/// </pre>
/// </example>
public string GetNewGuidStrig()
{
    Guid guid;
    do
        guid = Guid.NewGuid();
    while (guid == Guid.Empty);
    return guid.ToString();
}
}
}
```

Lähdekoodin ja kommentoinnin käytänteet toteutuivat suunnitelman mukaisesti. Kommentoinneissa oli kuitenkin puutteita.

5.6 Versiointi ja versiohallinta

Sovelluksen lähdekoodi sijoitettiin Git-pohjaiseen YouSource-julkistusjärjestelmään, josta se on myös tilaajan edustajien ja ohjaajien saatavilla. Kaikki sovellukseen ja projektiin liittyvä julkistettu dokumentaatio säilytetään trac-wikissä.

Dokumenttien versioissa käytettiin kolmiportaista versionumerointia. Ryhmän sisäiset versiot aloitettiin versionumerosta 0.0.1, ja kunkin uuden version osalta kasvatettiin vähiten merkitsevää numeroa yhdellä. Tällöin toinen versio oli versionumeroltaan 0.0.2. Projektioorganisaatiolle julkistettujen versioiden numerointi aloitettiin versionumerosta 0.1.0. Seuraavat versiot numeroitiin kasvattamalla toisen tason numeroa yhdellä. Ensimmäisen hyväksytyyn version numero oli 1.0.0, ja sitä seuraavissa hyväksytyissä versioissa kasvatettiin toisen tason numeroa yhdellä.

Suunnittelussa versiohallinnassa pitäydyttiin ja julkistettujen dokumenttien versionumerointi vastasi suunniteltua.

5.7 Testaus

Toteutettujen sovellusten toiminnan verifiointi tapahtui yksikkö- ja järjestelmätestauksella. Yksikkötestaus toteutettiin testaamalla yksittäiset luokat ja metodit Com-Testin C#-versiolla [3], mikäli se oli mahdollista. Yksikkötestit laati ja suoritti aliohjelman tekijä.

Järjestelmätestaukseen osallistuivat kaikki projektiryhmän jäsenet. Testausdokumenttien vastuuhenkilö laati ryhmän kanssa järjestelmän testaus suunnitelman vaatimusmäärittelyn pohjalta. Testaus suunnitelma oli samalla testustraportin dokumenttipohja. Testaus suunnitelma sisältää testauskerroilla suoritettavat testitapaukset ja testauskerralla suoritettujen testitapausten tulokset, virheet ja mahdolliset puutteet.

Kahdella järjestelmätestauskerralla havaitut virheet ja puutteet raportoitiin testustraportteihin[?] ja [?] sekä korjattiin. Testauksissa havaituista virheistä ja puutteista sekä niiden ratkaisusta keskusteltiin tilaajan edustajien kanssa viimeisessä palaverissa. Korjausten jälkeen suoritettiin regressiotestaus.

Vastaava ohjaaja koekäytti tietojärjestelmän molempia käytöliittymiä yhden kerran projektihuoneessa. Vastaava ohjaaja antoi projektiryhmälle palautetta[?] koekäytössään huomaamista puutteista ja kehitysideoista. Suunniteltua useamman henkilön koekäyttöä ei suoritettu.

Suunnitellusta poiketen testaus suunnitelma oli samalla testausraporttipohja ja lähdekoodiin kirjoitettua yksikkötestausta tehtiin hyvin vähän. Muilta osin sovellusten testaus toteutui suunnitellusti.

5.8 Tulosten katselmoinnit ja hyväksyminen

Ryhmän projektissa kirjoittama lähdekoodi katselmoitiin kaksi kertaa projektin aikana. Ensimmäiseen katselmointiin osallistui teknisten ohjaajien lisäksi koko projektiryhmä, vastaava ohjaaja ja tilaajan edustajista Vesa Lappalainen. Toiseen katselmointiin osallistui tilaajan edustajista Lappalainen, teknisistä ohjaajista Kosonen ja vastaava ohjaaja, sekä projektiryhmän jäsenistä Puupponen, Rautiainen ja Söderlund. Katselmoinneissa tekninen ohjaaja kommentoi lähdekoodia antaen vinkkejä ja parannusehdotuksia. Tekninen ohjaaja myös hyväksyi lähdekoodin viimeisessä katselmoinnissa. Projektiryhmä kirjasi katselmointien havainnot muistioiksi.

Projektin lopussa tulokset kokonaisuutena hyväksyttiin projektin ohjaajilla ja tilaajan edustajilla. Yksittäisistä tuloksista tilaajan edustajan hyväksyntä tarvittiin vähintään toteutetuille sovelluksille, sovellusraportille ja vaatimusmäärittelylle. Tekninen ohjaaja hyväksyi lähdekoodin yhdessä tilaajan edustajan kanssa. Vastaava ohjaaja hyväksyi projektin keskeisimmät dokumentit, joita ovat projektisuunnitelma, projektiraportti, sovellusraportti, vaatimusmäärittely ja luokkadokumentit. Projektisuunnitelma ja projektiraportti, sovellusraportti sekä vaatimusmäärittely hyväksyttiin projektipäällikön, tilaajan edustajan ja projektin vastaavan ohjaajan allekirjoituksilla.

Katselmoinnit ja tulosten hyväksyminen toteutuivat pääosin suunnitelman mukaisesti. Mononen ei päässyt osallistumaan toiseen lähdekoodin katselmointiin. Katselmoinnit auttoivat ryhmän jäseniä kehittämään ohjelmointitaitojaan saatuaan rakentavaa palautetta kirjoittamistaan koodista.

5.9 Tulosten koostaminen ja toimittaminen

Projektiryhmä kokosi projektin tulokset projektikansioon ja CD-levylle. Projektikansioon kerättiin kaikki projektissa laaditut dokumentit ja lähdekoodi. Lisäksi sähköpostiarkistot, tiivistetty projektin kuvaus ja jäsenten itsearviointit liitettiin projektikansioon ja CD-levylle. CD-levylle tallennettiin edellisten lisäksi myös kehitetty

ajettava sovellus ml. asennuspaketti ja asennusohjeet.

Tulokset toimitettiin tilaajalle USB-tikulla. Laitokselle toimitettiin projektikansio, joka sisältää projekti-CD:n. Toinen CD-levy toimitettiin laitoksen arkistoon. CD-levyt ja USB-tikut koostettiin vasta, kun kaikki projektin tulokset oli hyväksytty. Projektikansio sijoitettiin projektitilan kokoushuoneessa olevaan kirjahyllyyn.

Projektissa käytetty palvelin siirrettiin projektin päätyttyä tilaajan haltuun. Tietojärjestelmän viimeisimmät versiot ovat saatavilla YouSourcesta osoitteesta <https://yousource.it.jyu.fi/halyri/>.

Tulosten koostaminen ja toimittaminen toteutuivat suunnitelman mukaisesti paitsi, että tilaajan edustajat halusivat tulokset USB-tikulla CD-levyn sijaan.

6 Tehtävät, työmäärät ja työnjako

Luvussa esitetään projektiryhmän jäsenten suunnitellut ja toteutuneet vastualueet, tehtävät ja työnjako. Lisäksi luvussa esitetään ryhmän jäsenten arvioidut ja toteutuneet työtunnit tehtäväkokonaisuuksittain.

Projektiin käytettiin lähes suunniteltu määrä työtunteja. Työtunnit eivät kuitenkaan jakautuneet tehtäväkokonaisuuksittain täsmälleen suunnitelman mukaan. Varsinkin jäsenten välinen työtehtävien jako ei vastannut suunnitelmaa. Vähän tai ei lainkaan ohjelmoineet henkilöt keskittyivät projektin hallintaan sekä ongelmien tutkimiseen ja järjestelmätestaukseen. //TODO toteuma

6.1 Vastualueet

Projektipäällikkönä toimi Niko Mononen ja varapäällikkönä Atte Söderlund. Varapäällikkö vastasi projektipäällikön tehtävistä silloin, kun projektipäällikkö ei jostain syystä pystynyt hoitamaan jotain edellä mainituista tehtävistä. Varapäällikkö joutui muutaman kerran vastaamaan projektipäällikön tehtävistä hänen menojen ja kiireiden vuoksi.

Projektipäällikön vastuulle kuuluivat projektin suunnittelu ja hallinta sekä ajankäytön seuranta. Projektipäällikkö vastasi myös projektisuunnitelman ja projektiraportin laatimisesta. Projektipäällikön tehtäviin kuului lisäksi projektin tilan seuranta ja poikkeamiin reagoiminen.

Ryhmä jakoi tehtäviä monipuolisesti siten, että kaikki tekivät ainakin jossain vaiheessa projektia suunnittelua, toteutusta ja testausta. Tällöin kukin projektiryhmän jäsen sai kokemusta em. ohjelmistokehitysprosessin tehtäväkokonaisuuksista. Tuloksille nimetyt vastuuhenkilöt on esitetty taulukossa 6.1.

Vastuuhenkilö vastasi kyseisen tuloksen etenemisestä ja laadusta. Dokumentointia toteutettiin koko projektin ajan tasaisesti, jotta kaikki oleellinen saatiin kirjattua. Dokumentit laadittiin koko projektiryhmän voimin siten, että kukin laati vähintään yhden dokumentin.

Tulos	Suunniteltu	Toteutunut
Hätäkeskussovellus	Ilkka Rautiainen	Atte Söderlund
Käyttöliittymät	Atte Söderlund	Atte Söderlund
Mobiilisovellus	Atte Söderlund	Atte Söderlund
Palvelinsovellus	Veli-Mikko Puupponen	Veli-Mikko Puupponen
Projektisuunnitelma	Niko Mononen	Niko Mononen
Projektiraportti	Niko Mononen	Niko Mononen
Sovellusraportti	Atte Söderlund	Ilkka Rautiainen
Testausdokumentit	Ilkka Rautiainen	Ilkka Rautiainen
Tietokanta	Veli-Mikko Puupponen	
Vaatimusmäärittely	Veli-Mikko Puupponen	Veli-Mikko Puupponen

Taulukko 6.1: Olennaisten tulosten suunnitellut ja toteutuneet vastuuhenkilöt.

Suunniteltu vastuuhenkilö vastasi omasta vastuualueestaan koko projektin ajan. Suunniteltu vastuuhenkilöiden jako ei toteutunut. Tietokantaa ei prototyyppeihin suunniteltu eikä toteutettu. Hätäkeskussovelluksen ja sovellusraportin vastuuhenkilöiden vaihtamisen syynä on Rautiaisen keskittyminen dokumentointiin ja hätäkeskussovelluksen sekä älypuhelinsovelluksen riippuvuus toisistaan. Vastuuhenkilöiden poikkeaminen suunnitellusta ei vaikuttanut projektin tuloksiin.

6.2 Työnositus ja työmäärä

Projektiryhmän suunniteltiin käyttävän Sovellusprojektin osalta projektiin 250 tuntia kutakin jäsentä kohden, jolloin koko ryhmän työtunnit olivat yhteensä 1000 tuntia. Työmäärä vastaisi kymmentä opintopistettä ja olisi noin 14 tuntia viikossa jäsentä kohden. Lisäksi kunkin projektiryhmän jäsenen suunniteltiin käyttävän 70 tuntia oheiskursseihin. Projektin työmäärän suunniteltiin jakautuvan tehtävittäin ja jäsenten kesken kuvien 6.1 ja 6.2 mukaisesti.

Suunniteltu työnositus perustui kehitettävän tietojärjestelmän kokonaisrakenteeseen. Tehtävien jaossa huomioitiin jokaisen projektin jäsenen vastuualue. Työmäärien suunnittelussa oli huomioitu kehitettävän tietojärjestelmän rakenne, projektin jäsenten taidot ja aikataulut sekä projektin aikataulu. Työmäärien järkevyyttä oli pohdittu vertailemalla Paatti-projektin [2] suunniteltuja ja toteutuneita työmääriä.

Tehtäväkokonaisuus	Tehtävä	NM	VMP	IR	AS	YHT
Projektin hallinta						
	Projektsuunitelma	40	0	0	0	40
	Seuranta ja hallinta	19	2	2	2	25
	suunnittelu ja raportointi	20	2	2	2	26
	Projektiraportti	40	0	0	0	40
	Lisenssisitoumus	1	1	1	1	4
	Tiedotus	1	1	1	1	4
	Viimeistely ja kokoaminen	20	0	0	0	20
	Yhteensä	141	6	6	6	159
Palaverit						
	Esityslistat	1	1	1	1	4
	Palaverit	20	20	20	20	80
	Pöytäkirjat	10	10	10	10	40
	Valmistelu ja tutustuminen	8	8	8	8	32
	Katselmoinnit	4	4	4	4	16
	Yhteensä	43	43	43	43	172
Perehtyminen						
	Toteutukseen perehtyminen	20	20	20	20	80
	Aihealueeseen perehtyminen	5	5	5	5	20
	Yhteensä	25	25	25	25	100
Määrittely						
	Alustava vaatimusmäärittely	2	20	2	2	26
	Vaatimusmäärittelyn päivitys	1	10	1	1	13
	Vaatimusmäärittelyn viimeistely	0	2	0	0	2
	Yhteensä	3	32	3	3	41
Suunnittelu						
	Tietojärjestelmän rakenne	2	10	2	0	14
	Tietokannat	1	10	0	0	11
	Asetusten hallinta	1	10	1	1	13
	Mobiilikäyttöliittymä	1	2	2	20	25
	Hätäkeskuskäyttöliittymä	1	2	2	20	25
	Palvelinsovellus	2	20	10	0	32
	Rajapinnat	1	10	10	0	21
	Yhteensä	9	64	27	41	141

Kuva 6.1: Työajankäyttösuunnitelma tehtäväkokonaisuuksittain 1/2.

Tehtäväkokonaisuus Tehtävä		NM	VMP	IR	AS	YHT
Toteutus						
	Tietojärjestelmän rakenne	0	10	5	0	15
	Tietokannat	1	10	0	0	11
	Asetusten hallinta	1	10	1	1	13
	Mobiilikäyttöliittymä	4	2	15	40	61
	Hätäkeskuskäyttöliittymä	4	2	15	40	61
	Palvelinsovellus	4	20	20	0	44
	Yhteensä	18	64	76	86	244
Testaus						
	Suunnittelu	1	2	20	5	28
	Yksikkötestaus	2	5	5	5	17
	Järjestelmätestaus	2	2	10	5	19
	Yhteensä	5	9	35	15	64
Viimeistely						
	Sovellusraportti	3	2	20	11	36
	Lähdekoodin viimeistely	3	5	15	20	43
	Yhteensä	6	7	35	31	79
Tehtäväkokonaisuus Tehtävä		NM	VMP	IR	AS	YHT
Sovellusprojekti yhteensä		250	250	250	250	1000
Oheiskurssit						
	Puheviestintä	25	25	25	25	100
	Kirjoitusviestintä	25	25	25	25	100
	Sovellusprojektin hallinta	20	20	20	20	80
	Yhteensä	70	70	70	70	280
Sovellusprojektit ja oheiskurssit yhteensä		320	320	320	320	1280

Kuva 6.2: Työajankäyttösuunnitelma tehtäväkokonaisuuksittain 2/2.

Työnjaossa onnistuttiin melko hyvin huomioimaan jäsenten poissaolot sekä muiden kurssien vaatima työmäärä. Tehtäviin tarvittavia työtunteja ei osattu arvioida riittävän tarkasti, joten joidenkin tehtäväkokonaisuuksien toteutunut työtuntimäärä poikkesi selvästi suunnitellusta. Toteutuneita työtunteja kertyi sovelluksen vastuuhenkilölle huomattavasti suunniteltua enemmän. Tämä johtui siitä, että toteutettavia sovelluksia oli monta ja jokaisen sovelluksen rakenteen ymmärtäminen vaati huomattavan määrän työtunteja. Tämän lisäksi palvelinsovellus sekä videon ja äänen siirto osoittautui hankalaksi suunnitella ja toteuttaa. Puupponen käytti lähes tuplasti enemmän työtunteja suunnittelun ja toteutuksen osalta muihin jäseniin verrattuna. Hankaluudesta johtuen työtä ei voinut helposti siirtää muille jäsenille, koska toteutuksen ymmärtäminen olisi vaatinut huomattavan määrän työtunteja.

6.3 Ryhmän työtunnit tehtäväkokonaisuuksittain**6.4 Niko Monosen työtunnit tehtäväkokonaisuuksittain****6.5 Veli-Mikko Puupposen työtunnit tehtäväkokonaisuuksittain****6.6 Ilkka Rautiaisen työtunnit tehtäväkokonaisuuksittain****6.7 Atte Söderlundin työtunnit tehtäväkokonaisuuksittain**

7 Ohjelmistokehitysprosessi ja aikataulu

Luvussa kuvataan projektissa noudatettua prosessia ja aikataulua. //TODO olennaiset havainnot ja polleamat syyt seuraukset.

7.1 Prosessi

Projektin alussa oli määrittelyn ja suunnittelun vaihe. Siinä suunniteltiin projektin läpivienti, kartoitettiin järjestelmän vaatimukset ja laadittiin vaatimusmäärittely sekä hahmoteltiin sovelluksen käyttöliittymä, rakenne ja arkkitehtuuri.

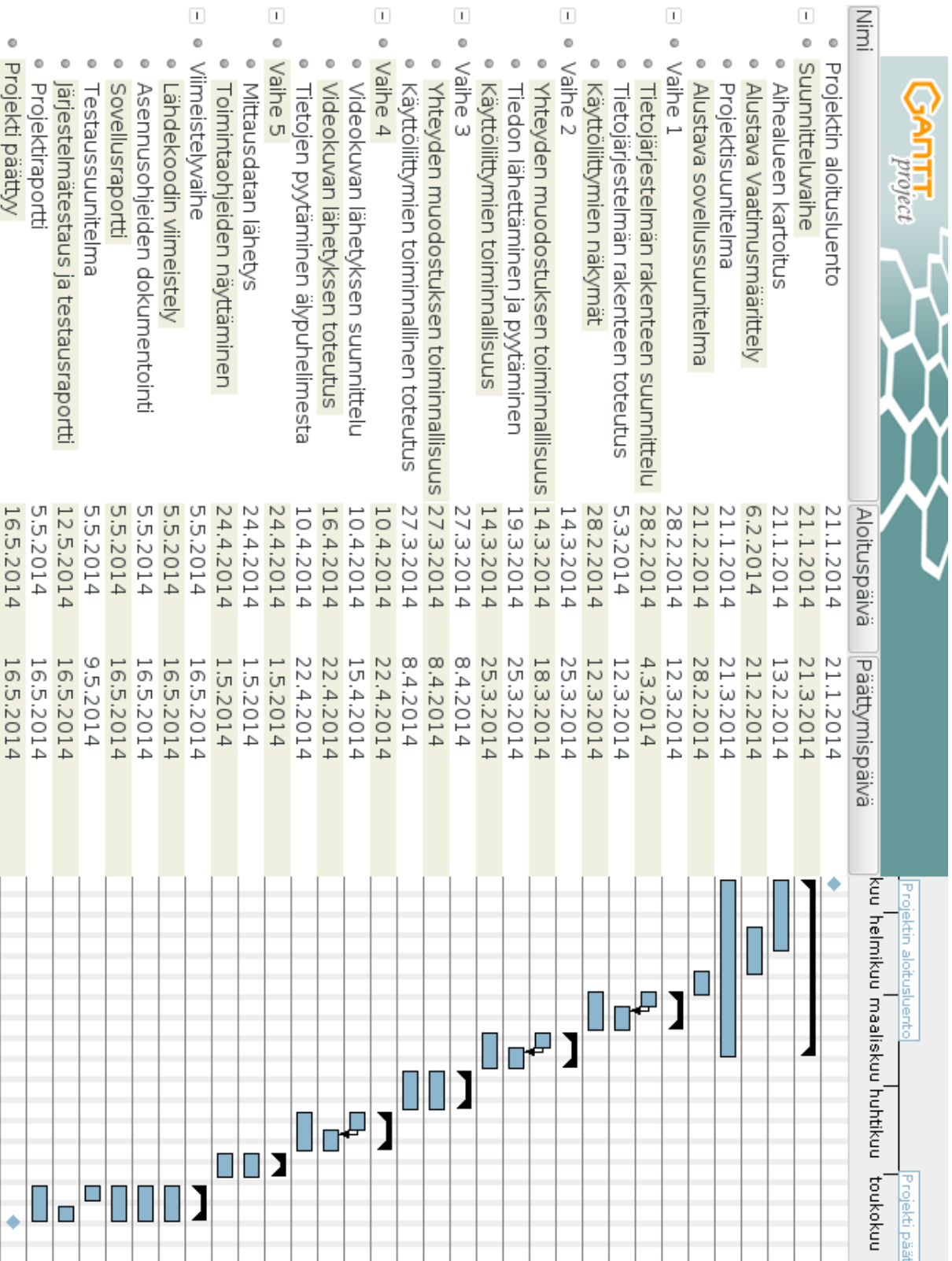
Ensimmäisessä kehitysvaiheessa toteutettiin mobiilisovelluksen, palvelimen ja hätäkeskuksen käyttöliittymän runko hyvin pienellä määrällä toivotuista toiminnoista. Kun runko oli kasassa ja todettu toimivaksi, alettiin siihen toteuttamaan lisää ominaisuuksia vaatimusmäärittelyn pohjalta kahden viikon vaiheissa. Ohjelmistokehitysvaiheita oli yhteensä viisi. Vaiheen aloittavassa palaverissa esiteltiin sen hetkistä sovellusta ja päätettiin tilaajan kanssa, mitä uusia ominaisuuksia kehitettiin seuraavaksi. Jokaiseen vaiheeseen sisältyi ohjelmiston suunnittelua, toteutusta ja testausta. Projektin lopuksi oli tulosten viimeistelyn ja raportoinnin vaihe.

7.2 Aikataulu

Kuvassa 7.1 on esitelty projektin suunniteltu aikataulu. Projekti alkoi 21.1.2014, ja se päättyi kesäkuun puolessa välissä. Ensimmäinen yksinkertainen prototyyppi oli valmis 10.4. Tämän jälkeen kahden viikon vaiheissa toteutettiin lisää ominaisuuksia palaverissa sovitun mukaisesti. Toukokuussa ei toteutettu uusia ominaisuuksia, vaan keskityttiin ohjelmiston, lähdekoodin ja dokumentaation viimeistelyyn.

Toteutuneen aikataulun rakenne vastaa hyvin pitkälti suunniteltua aikataulua. Toiseksi viimeisessä kehitysvaiheessa kohdattujen ongelmien vuoksi aikataulu venyi. Myös projektisuunitelman laatiminen venyi kuukauden yli suunnitellun. Projekti päättyi kesäkuun puolessa välissä, eikä 16.5. niin kuin oli alunperin suunniteltu. Projektille oli varattu pelivaraa toukokuun loppuun, joten projektin aikataulu viivästyi 4 viikkkoa.

7.3 Ryhmän työtunnit viikoittain**7.4 Niko Monosen työtunnit viikoittain****7.5 Veli-Mikko Puupposen työtunnit viikoittain****7.6 Ilkka Rautiaisen työtunnit viikoittain****7.7 Atter Söderlundin työtunnit viikoittain**



Kuva 7.1: Projektin suunniteltu aikataulu.

8 Riskit ja niiden hallinta

Luvussa kuvataan projektin ennakoitujen riskien toteutumista, ehkäisemistä ja käsittelyä sekä niiden vaikutusta projektin läpivientiin ja tuloksiin.

8.1 Riskien todennäköisyydet ja haitat

Riskien todennäköisyydet sekä niiden arvioidut ja toteutuneet haittavaikutukset on esitetty taulukossa 8.1. Todennäköisyyttä ja haittavaikutusta arvioitiin asteikolla pieni, keskinkertainen ja suuri.

Riski	Arvioitu todennäköisyys	Arvioitu haittavaikutus	Toteutunut haittavaikutus
Alustan rajoitukset	keskinkertainen	suuri	suuri
Kehittäjien tietotaidon puutteet	keskinkertainen	keskinkertainen	suuri
Tavoitteiden rajaaminen	keskinkertainen	keskinkertainen	pieni
Tiedotuksen puutteet	pieni	keskinkertainen	keskinkertainen
Jäsenten poissaolot	keskinkertainen	pieni	pieni
Kokemattomuus projektihallinnassa	keskinkertainen	pieni	keskinkertainen

Taulukko 8.1: Riskien todennäköisyys ja haittavaikutus.

Projektin aikana ei ilmennyt ennakoimattomia riskejä. Suurimmat erot arvioitujen ja toteutuneiden haittavaikutusten välillä oli kehittäjien tietotaitojen puutteissa ja kokemattomuudessa projektinhallinnassa. Haittavaikutukset tavoitteiden rajaamisessa jäivät odotettua pienemmiksi.

8.2 Alustan rajoitukset

Projektiryhmä joutui päättämään yhdessä tilaajan edustajien kanssa, mille alustoille tietojärjestelmään kuuluvat sovellukset toteutettiin projektin aikana. Koska alustat ovat hyvin erilaisia, olisi voinut jonkin ominaisuuden toteuttaminen vaatia huomattavasti enemmän resursseja kuin jollakin toisella alustalla. Alustan valinnasta riippuen joitakin tietojärjestelmältä vaadittuja ominaisuuksia oltaisiin voitu joutua rajaamaan projektin ulkopuolelle.

Riskiä ehkäistiin tutustumalla alustavasti etukäteen eri alustoihin. Riskin toteutuksessa oli mahdollista vaihtaa käytettävää alustaa tai rajata joitakin toteutettavia ominaisuuksia projektin ulkopuolelle.

Valittu alusta osoittautui ennakoidusti haasteelliseksi. Tämä johtui WP8-alustan valmiskomponenttien keskeneräisyydestä sekä puuttellisista ja heikosta dokumentaatiosta. Projektiryhmä joutui useasti päättämään yhdessä tilaajan edustajien ja teknisten ohjaajien kanssa, miten kohdattu ongelma ratkaistaan. Alustaa ei kuitenkaan jouduttu vaihtamaan, ja projektille asetettuja vaatimuksia ei muutettu.

Riskistä johtuen toteutettuun prototyyppiin jouduttiin toteuttamaan paljon komponentteja itse. Näitä komponentteja ei ole optimoitu eikä testattu kovinkaan perusteellisesti. Nämä komponentit kuitenkin osoittavat tekniikan olevan mahdollista toteuttaa. Prototyypin toteuttaminen toisella alustalla olisi vaatinut vähemmän työtunteja.

8.3 Kehittäjien tietotaidon puutteet

Ohjelmiston toteutuksessa käytettiin monia eri työkaluja ja tekniikoita, joista projektiryhmän jäsenillä ei ollut aikaisempaa kokemusta. Näin ollen ryhmä ei aina osannut ennakoida mahdollisia ongelmakohtia.

Riskin ehkäisemiseksi ryhmän jäsenet tutustuivat käytettäviin työkaluihin ja tekniikoihin. Ongelmatilanteissa projektiryhmällä oli myös käytettävissään kaksi teknistä ohjaajaa ja asiantuntijoita.

Projektisuunnitelmassa ennakoidusti ohjelmistojen toimintaan tutustumiseen jouduttiin käyttämään paljon aikaa, joka taas vei aikaa itse sovelluksen kehittämisestä. Lisäksi työkalujen ja ohjelmointikielen hyviä ominaisuuksia ei välttämättä heti osat-

tu hyödyntää, vaan asiat tehtiin niin sanotusti "vaikeamman kautta". Lisäksi WP8-alustan valmiskomponentit olivat keskeneräisiä sekä dokumentaatio oli puuttellisista ja heikkoa.

Projektiryhmän olisi ehkä pitänyt olla aktiivisemmin yhteydessä teknisiin ohjaajiin liittyen sovelluksen toteutukseen, jotta riskiä toteutuksen tietotaidon puutteissa olisi saatu entisestään ehkäistyä. Joihinkin asioihin edes tekninen ohjaaja ei osannut vastata, vaan asia jouduttiin selvittämään itse käyttämällä siihen paljon työtunteja.

Riskistä johtuen toteutettuun prototyyppiin jouduttiin toteuttamaan paljon komponentteja itse. Näitä komponentteja ei ole optimoitu eikä testattu kovinkaan perusteellisesti. Nämä komponentit kuitenkin osoittavat tekniikan olevan mahdollista toteuttaa.

Riskin toteutumisesta seurasi ryhmän jäsenille arvioitua suuremmat työtunnit ja projektin viivästyminen tavoitteesta.

8.4 Tavoitteiden rajaaminen

Tilaaajan edustajien tavoitteet olivat enimmäkseen hyvin yleisen tason kokonaistavoitteita, joten vaatimusmäärittelyn laatimiseen oli kiinnitettävä paljon huomiota. Ryhmän muodostama kuva ohjelmiston tavoitteista olisi voinut erota tilaaajan toiveista ja tarpeista. Tällöin tietojärjestelmä ei olisi tullut projektin päättyessä sisältämään kaikkia vaatimusmäärittelyssä kuvattuja ominaisuuksia.

Riskin ehkäisemiseksi projektiryhmä joutui projektin alussa käyttämään paljon aikaa sovittaakseen tilaaajan näkemykset yhtenäiseksi vaatimusmäärittelyksi [6]. Ennen kuin ohjelmiston toteutus aloitettiin, projektiryhmällä ja tilaajalla oli melko yhtenäinen kuva siitä, mitä ohjelmiston osia ja toimintoja oli mahdollista saada projektin aikana valmiiksi. Vaatimusmäärittelyyn kirjattiin projektissa toteutettavat vaatimukset ja priorisoitiin ne.

Riski ei juurikaan toteutunut ehkäisin onnistumisesta johtuen. Tilaaajan edustajat olivat joustavia tavoitteiden suhteen, ja projektille asetettuja tavoitteita ei jouduttu muuttamaan.

8.5 Tiedotuksen puutteet

Säännöllisesti järjestetyissä palavereissa tilaajan edustajat saivat hyvin tietoa projektin tilasta. Palaverien ulkopuolella tuloksista tiedotettiin, mutta tiedotus suorite-
tuista tehtävistä tai projektin etenemisestä oli olematonta, mikä laskee projektin läpiviennin läpinäkyvyyttä.

Ryhmän sisäisessä tiedotuksessa ei ollut ongelmia. Ryhmä oli usein samassa tilassa työskentelemässä, sekä kaikki muut saatiin hyvin nopeasti tavoitettua sähköpostien ja pikaviestimien avulla. Kommunikaatio sujui ongelmitta, ja tiedotus tuntui toimivan.

Ennakoitu riski tiedotuksen puutteesta toteutui osittain, mutta sen haittavaikutukset jäivät keskinkertaiseksi. Projektiryhmän aktiivisemmalla tiedottamisella olisi ollut mahdollista saada enemmän palautetta tilaajalta. Joissakin tilanteissa tiedotus viivästyi, mikä johtui projektin hallinnan puutteista. Tiedotuksen viivästyminen vaikutti erityisesti aikataulujen sopimiseen.

8.6 Jäsenten poissaolot

Projektin aikana ryhmän jäsenille ei ilmaantunut merkittäviä odottamattomia poissaoloja. Kaikilla ryhmän jäsenillä oli velvoitteita muillakin kursseilla ja työelämässä, mutta mitkään niistä eivät varsinaisesti olleet odottamattomia. Joihinkin toisten kurssien työtehtäviin meni odotettua enemmän aikaa, mutta se ei vaikuttanut projektin läpivientiin.

Riskiä hallittiin projektin aikana siten, että ryhmän jäsenet keskustelivat projektin aikana työtehtävistään. He myös pitivät huolen siitä, että myös muut tiesivät, mitä kukin jäsen on projektin aikana tehnyt. Täten yllättävän poissaolon toteutuessa ryhmän jäsenet olisivat tienneet poissaolijan työtehtävät, ja niitä olisi voinut jakaa muiden jäsenten kesken.

Vaikka ryhmän jäsenet olivat enimmäkseen tietoisia toistensa tekemisistä ja projektia edistävästä toimenpiteistä sekä hidastavista puutteista, ryhmän jäsenet eivät aina toimineet asian hyväksi muistuttamalla tai ottamalla asian hoidettavaksi.

Poissaolot vaikuttivat suunniteltuun aikatauluun viivästyttämällä tuloksia ja täten projektin läpivientä.

8.7 Kokemattomuus projektihallinnassa

Projektiryhmän jäsenillä ei ollut aiempaa kokemusta projektin suunnittelusta, läpiviennistä tai hallinnasta. Myös projektipäällikkö vastasi projektin läpiviennistä ensimmäistä kertaa. Vaikka riski ennakoitiin toteutuvaksi, suoriuduttiin projektin läpiviennissä ilman suurempia ongelmia.

Projektipäällikkö pyrki pitämään itsensä tietoisena siitä, mitä kukin ryhmän jäsen on tehnyt, tekee tällä hetkellä ja aikoo tehdä seuraavaksi. Ryhmän jäsenet kirjasivat työtunnit hyvin, täsmällisesti ja ajallaan. Kaikki ryhmän jäsenet olivat yleensä tietoisia siitä, missä vaiheessa projekti eteni. Lisäksi projektiryhmä auttoi projektipäällikköä. Ryhmän jäsenet muistuttivat projektipäällikköä, jos jokin asia uhkasi jäädä huomiotta. Myös vastaava ohjaaja auttoi projektipäällikköä projektin läpiviennissä.

Ryhmän jäsenet ja etenkin projektipäällikkö ei osannut arvioida vaadittavia työmääriä riittävällä tarkkuudella. Riskin toteutuminen keskinkertaisena aiheutti projektin viivästymistä suunnitellusta aikataulusta.

9 Jäsenten kokemuksia ja oppimaa

Projekti oli kaikille ryhmän jäsenille ensimmäinen oikea ohjelmistokehitysprojekti, joka sisälsi projektin kaikki kehitysvaiheet aina vaatimusmäärittelystä projektin tulosten luovutukseen. Jäsenet saivat kattavan kuvan projektin läpiviennistä ja ymmärsivät, että sovelluksen kehittäminen on kaikkea muutakin kuin pelkkää ohjelmointia.

Luvuissa 9.2 – 9.5 projektiryhmän jäsenet kuvaavat kokemuksiaan ja oppimiaan asioita

9.1 Ryhmän kokemuksia

9.2 Niko Monosen kokemuksia ja oppimaa

Projektin kohdealue vaikutti todella haastavalta ja mielenkiintoiselta. Tietojärjestelmäkokonaisuudet ovat aina olleet sydäntäni lähellä, ja päätin ottaa projektipäällikön tehtävän vastuulleni.

Sain selkää ohjausta ja tukea muilta ryhmän jäseniltä. Nämä asiat helpottivat huomattavasti työtäni ja tehostivat oppimistani merkittävästi. Ryhmän jäsenet tekivät myös paljon pyytämättäni, kun en osannut ottaa kaikkea huomioon. Projektipäällikön tehtävät osoittautuivat melko haastaviksi ja yllättävän paljon aikaa vieviksi.

Erilaisten dokumenttien kirjoittaminen tuntui turhauttavalta, mutta ymmärrän kyllä, että ilman suunnittelua ja raportointia projektin läpiviennistä ei tulisi mitään. Jouduin kuitenkin käyttämään huomattavan paljon aikaa niiden laatimiseen, jotta niistä tuli selkeitä, ymmärrettäviä ja kieliasultaan riittävän oikeita. Sain paljon palautetta dokumenteista myös vastaavalta ohjaajalta, mikä vaikutti dokumenttien laatuun.

Projektin aikana pyrin pitämään itseäni koko ajan tietoisena siitä, mitä ryhmän jäsenet tekevät ja mitä he tulevat tekemään. Mielestäni onnistuin tässä hyvin. Olisin kaivannut ryhmän jäseniltä hieman enemmän oma-aloitteisuutta tekemisiensä raportoinnissa. Tuntui jotenkin turhauttavalta kysellä jatkuvasti, että mitä olet tehnyt ja mitä teet seuraavaksi.

Projektipäällikön tehtävät kuitenkin veivät työaikaani varsinaiselta ohjelmoinnilta ja se harmitti minua paljon. Projektin aikana toteutin vain muutaman pienen toiminnallisuuden. Tämä ei kuitenkaan varsinaisesti haittaa, sillä sain tärkeää kokemusta projektin johtamisesta.

Projektiryhmä oli erinomainen. Kaikki ymmärsivät vastuunsa ja tekivät töitä todella tunnollisesti. Välillä tuntui, että muilla jäsenillä oli paljon aikaa tehdä projektia, enkä itse pysynyt lainkaan mukana työtunneissa. Tämä johtui siitä, että olin puoli-päiväisenä töissä ja minulla oli muitakin työläitä kursseja. Projektin käytössä oleva työaikani oli yleensä vasta klo 16 jälkeen, mikä selvästi haittasi yhteistoimintaa muun ryhmän kanssa.

Kokonaisuutena sovellusprojekti oli positiivinen kokemus, vaikka toukokuussa meinasin uupua kaikesta itselleni hankkimasta työmäärästä. Koen saaneeni todella tärkeää kokemusta projektimuotoisesta työtavasta tällä kurssilla.

9.3 Veli-Mikko Puupposen kokemuksia ja oppimaa

9.4 Ilkka Rautiaisen kokemuksia ja oppimaa

9.5 Atte Söderlundin kokemuksia ja oppimaa

10 Yhteenveto

Hälyri-projekti suoritti toteutettavuuskartoituksen ja kehitti hätäilmoitusjärjestelmän prototyypin Sovellusprojekti-kurssilla keväällä 2014. Projektin tulokset luovutettiin tilaajalle kesäkuun puolessa välissä.

Projektin suurimpina riskeinä toteutuivat alustan haasteellisuus ja kehittäjien tietotaidon puutteet. Kyseiset riskit saatiin kuitenkin hallittua projektin aikana, joten ne eivät estäneet projektin läpivientiä, mutta vaikuttivat hieman tuloksiin. Toteutuneista rikseistä johtuneiden ongelmien ratkaisemiseen jouduttiin kuitenkin käyttämään suunniteltua enemmän työtunteja.

Projektissa kehitetyn prototyypin pohjalta tullaan kehittämään tietojärjestelmästä pilottiversio, joten projektissa laadittuja dokumentteja tullaan hyödyntämään myöhemmin. Kehitettävä prototyyppi oli laaja ja monimutkainen, joten tietojärjestelmän sovellusten ominaisuuksien priorisointiin oli kiinnitettävä erityisen paljon huomiota.

Tietojärjestelmän toteutuksen kokemuksen lisäksi sovellusprojekti antoi projektiryhmän jäsenille kattavan käsityksen ohjelmistoprojektissa työskentelystä sekä sen vaatimuksista ja työtavoista. Ryhmän jäsenet oppivat asioita käytännön tekemisen kautta sekä sovelsivat aikaisemmilta kursseilta oppimaansa teorian tietoja ja taitoja.

Projektiin käytettyjen työtuntien toteuma ylittyi hieman. Jokainen jäsen teki projektissa reilut 250 tuntia töitä, joten projektiryhmän kokonaispanokseksi muodostui noin 1400 tuntia. Projektin läpivienti venyi neljällä viikolla, eikä päättynyt aikataulullisesti suunnitellun pelivaran puitteissa.

Vaatimusmäärittelyn laatimisessa ja prototyypin kehittämisessä projektiryhmä onnistui suunnitellusti. Vaikka tietojärjestelmän prototyyppi ei vielä sisällä kaikkia vaadittuja ominaisuuksia, tilaajan edustajat voivat käyttää sitä edistääkseen hanketta. Projektin aihealue ja käytetyt kehitysvälineet olivat vaativia ja kehitettävä tietojärjestelmä oli hyvin laaja. Projektin läpivienti onnistui erinomaisesti huomioiden, että se oli jäsenille ensimmäinen laajempi ohjelmistokehitysprojekti.

11 Lähteet

- [1] Petri Heinonen, "Ajankäytönseurantasovellus", saatavilla Excel-muodossa <URL: <http://appro.mit.jyu.fi/tools/ajankaytto/ajankaytonseuranta.xls>>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, viitattu 23.2.2014.
- [2] Tapio Keränen, Toni Salminen, Jari Salokangas ja Lauri Satokangas, "Paatti-projekti", saatavilla PDF-muodossa <URL: <http://sovellusprojektit.it.jyu.fi/paatti/>>, Jyväskylän yliopisto, tietotekniikan laitos, 2012.
- [3] Tomi Karppinen, "ComTest for C#", saatavilla WWW-osoitteessa <URL: <https://trac.cc.jyu.fi/projects/comtest/wiki/ComTestCsharp>>, Jyväskylän yliopisto, tietotekniikan laitos, 14.2.2014.
- [4] Microsoft, "Windows Phone SDK", saatavilla WWW-osoitteessa <URL: <http://dev.windowsphone.com/en-us/>>, Microsoft, 2014.
- [5] Microsoft, "C# Programming Guide", saatavilla WWW-osoitteessa <URL: <http://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>>, Microsoft, 2014.
- [6] Niko Mononen, Veli-Mikko Puupponen, Ilkka Rautiainen ja Atte Söderlund, "Hätäilmoitustietojärjestelmän vaatimusmäärittely", saatavilla WWW-osoitteessa <URL: <https://trac.cc.jyu.fi/projects/sovproj/wiki/Halyri/Vaatimusmaarittely>>, Jyväskylän yliopisto, tietotekniikan laitos, 2014.
- [7] Ilkka Rautiainen "Hälyri-projektin sovellusraportti", saatavilla WWW-osoitteessa <URL: <https://trac.cc.jyu.fi/projects/sovproj/wiki/Halyri/Sovellusraportti>>, Jyväskylän yliopisto, tietotekniikan laitos, 2014.
- [8] Jukka-Pekka Santanen, "Tietotekniikan Sovellusprojektien ohje", saatavilla WWW-osoitteessa <URL: <http://www.mit.jyu.fi/opetus/sovellusprojektit/projohje.html>>, Jyväskylän yliopisto, tietotekniikan laitos, 29.1.2013.