



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

RALLIKILPAILUJEN TULOSPALVELUOHJELMAT

Marko Koponen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä Marko Koponen	
Työn nimi Rallikilpailujen tulospalveluohjelmat	
Päiväys 13.5.14	Sivumäärä/Liitteet 35 / 2
Ohjaaja lehtori Keijo Kuosmanen	
Toimeksiantaja/Yhteistyökumppani(t) projektivastaava Mauri Kotala, Lap Data System Oy	
Tiivistelmä <p>Tämän opinnäytetyön tavoitteena oli toteuttaa selainpohjainen järjestelmä ajanottolaitteilta saatavien aikatietojen tallentamiseen. Järjestelmän kehityksen aikana tavoitteena oli tallentaa erilaisten rallikilpailujen tuloksia, mutta järjestelmää voidaan hyödyntää myös mm. moottorikelkkakilpailuissa.</p> <p>Työn tilaajalla on jo ennestään käytössä sarjaporttiliitännäiset ajanmittauslaitteet, joita projektin määrittelyvaiheessa päätettiin käyttää. Myöskään muita liitännätapoja ei haluttu unohtaa, vaan järjestelmä pyrittiin toteuttamaan mahdollisimman modulaariseksi. Projekti sisälsi kolme kehitysvaihetta: aikatiedon lukeminen sarjaportin kautta, tuloksen lähettäminen palvelimelle ja tuloksen tallentaminen tietokantaan. Käyttöliittymän ulkoasuun ei vielä kehitysvaiheessa panostettu, vaan siitä tehtiin hyvin yksinkertainen ja informatiivinen järjestelmän testauksen helpottamiseksi.</p> <p>Järjestelmän toteuttamiseen käytettiin avoimen lähdekoodin ohjelmistoja. Java-sovelma ohjelmointiin NetBeans-ohjelmointiympäristössä ja Eclipse-ympäristöä käytettiin Web-sivun, JavaScriptien ja PHP-kirjaston toteuttamiseen. Järjestelmän testaaminen suoritettiin XAMMP-ohjelmistokokoelmalla.</p> <p>Opinnäytetyön tuloksena syntyi järjestelmä, jolla voidaan tallentaa kellolaitteen lähettämä aikatieto palvelimella olevaan tietokantaan. Järjestelmä toimii lähes kaikilla käyttöjärjestelmillä, sillä sen käyttöliittymä on Web-sivu ja sarjaportin luku on toteutettu Java-sovelman avulla.</p>	
Avainsanat ralli, Java-sovelma, JavaScript, PHP, HTML	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author Marko Koponen			
Title of Thesis Result System for Rally Competitions			
Date	13 May 2014	Pages/Appendices	35 / 2
Supervisor Mr. Keijo Kuosmanen, Lecturer			
Client Organisation /Partners Mr. Mauri Kotala, Project Manager of Lap Data System Oy			
<p>Abstract</p> <p>The goal of this thesis was to develop a complete system for storing measured time data from measuring devices to the database in a server. During development the aim was time measurement for rally competitions, but this system can also be used to measure time in for example snowmobile competitions. The most important thing in this thesis was to develop functionalities for acquiring measured time data and storing that data to the database.</p> <p>The client company already has in use time measuring devices which use the serial port. In the definition part of the project it was desired to use them also with this new system. During development it was also kept in mind that the connection type can change in future. To support changes, the system was done as modular as possible. This project consists of three parts. The first part handles the acquiring of measured time data from the serial port to the Web browser. The second part sends data to the server and the third part stores data to the database. The interface of the solution was done to be simple and informative for testing purposes.</p> <p>The complete system was done with open source software packets which were already well-known. The Java-applet was programmed using NetBeans as the software development environment. The Web page, Java scripts and the PHP library was developed using Eclipse. For testing of the system, the XAMMP package was used as a testing environment.</p> <p>As a result of this thesis a system for storing measured time data was done. The system can read measured time from the serial port and send it to the server and store it to a database. This result system can be used on almost every platform, because its user interface is a Web page which uses Java-Applet.</p>			
Keywords Rally, Java applet, JavaScript, PHP, HTML			

ESIPUHE

Tulostentallennusjärjestelmän suunnittelu ja toteutus on ollut mielenkiintoinen ja opettavainen tehtävä. Tässä opinnäytetyössä opin paljon selainohjelmointia, joka vielä projektin alkaessa oli minulle melko tuntematonta. Aihealueena ralli on mielenkiintoinen ja projektin aikana opin paljon uutta rallikisoista ja niissä tapahtuvasta aikatiedon käsittelystä.

Kiitän opinnäytetyön aiheesta projektivastaava Mauri Kotalaa Lap Data System Oy:stä. Häneltä olen saanut tähän opinnäytetyöhön runsaasti harvinaista tietoa rallikisojen järjestämisestä ja aikatietojen tallentamisesta. Suuri kiitos myös Lap Data System Oy:n toimitusjohtaja Pasi Ojaniemelle, jolta olen saanut kaiken tarvitsemani teknisen tuen niin selainohjelmointiin kuin palvelinpuolen toteutukseen. Haluan kiittää myös lehtori Keijo Kuosmasta opinnäytetyön kärsivällisestä ohjaamisesta. Lisäksi suuri kiitos läheisille ja ystäville siitä runsaasta näkyvästä ja näkymättömästä tuesta, jota olen saanut koko opiskeluni ajan.

Kuopiossa 13.5.2014

Marko Koponen

SISÄLTÖ

1	JOHDANTO	8
2	RALLIN HISTORIAA JA KILPAILUTAPAHTUMAN KUVAUS.....	9
2.1	Ralliautojen ryhmäjako.....	9
2.2	Erlaisia ralleja	9
2.3	Ralli-sprint ja kelkka-sprint	10
2.4	Rallin ajanoton kehittyminen vuosien saatossa.....	10
2.5	Yleiskuvaus kilpailun järjestelyistä	12
2.5.1	Rallikilpailun tulospalveluhenkilöstö	13
2.5.2	Kellojen synkronointi ja tulosten toimittaminen tulospalveluhenkilöstölle.....	13
2.6	Yhden erikoiskokeen kuvaus	14
2.6.1	Erikoiskokeen lähtöalue	14
2.6.2	Erikoiskokeen maalialue.....	14
2.7	Tuloslaskennan periaatteet.....	15
3	TYÖKALUJEN VALINTA JA JAVA-SOVELMA.....	16
3.1	NetBeans ohjelmointityökaluna	16
3.2	Java-sovelma.....	16
3.3	Java-sovelman allekirjoitus, avaimen luonti.....	16
3.4	Java-sovelman allekirjoitus, avaimen käyttöönotto NetBeans:ssa	17
3.5	TestNG-testin lisääminen NetBeans-projektiin	18
3.6	Eclipse ohjelmointityökaluna	20
4	AJANOTTOJÄRJESTELMÄN TOTEUTUS	21
4.1	Tavoitteet.....	21
4.2	Yleiskuvaus järjestelmästä	23
4.3	Sarjaporttiyhteys jSSC-kirjastolla.....	24
4.3.1	Sarjaporttien listaaminen getPortNames-metodilla.....	24
4.3.2	Sarjaportin alustus ja käyttöönotto	24
4.3.3	Kuvaus Java-sovelman rakenteesta	25
4.4	JavaScript-funktiot	26
4.5	PHP-kirjasto.....	27
4.6	MySQL-tietokanta	29
4.7	Käyttöliittymä ja tietoturvaravitukset	29

5 JATKOKEHITYSAJATUKSIA	32
LÄHTEET	33
LIITE 1. ARCTIC LAPLAND RALLY 2014 -KILPAILUN AIKATAULU	34

LYHENTEET JA MÄÄRITELMÄT

FIA	Federation Internationale de l'Automobile, kansainvälinen autoliitto, joka antaa määräykset autokilpailuihin ja valvoo niiden noudattamista.
ek	Erikoiskoe, suljettu osuus tietä, joka pyritään ajamaan mahdollisimman nopeasti.
ek-henkilö	Henkilö, joka aikatarkistuspisteellä merkitsee ajoneuvon numeron ja ajan pöytäkirjaan.
aikatarkastuspiste	Paikka, johon kuljettajan on saavuttava suunnitelmassa ilmoitettuna ajan-kohtana.
Ajax	Asynchronous JavaScript And XML, joukko Web-sovelluskehityksen tekniikoita, joiden avulla Web-sovelluksista voidaan tehdä vuorovaikutteisempia.
jQuery	Avoimen lähdekoodin JavaScript-kirjasto, jonka syntaksi on tehty mahdollisimman helposti ymmärrettäväksi.
JSObject	Luokka, jonka avulla Java-koodista voidaan käsitellä JavaScript-objectteja.
jSSC	Java Simple Serial Connector, Java-kirjasto sarjaporttien käyttöön.

1 JOHDANTO

Lähes kaikissa rallikisoissa ajanotto toimii automatisoidusti, niin että saavutetaan puolueeton ja tarkka aikatieto. MM-rallissa käytettävät ajanottolaitteiston kaltaiset ratkaisut ovat liian kalliita käytettäväksi Suomen rallikilpailuissa, joten tarvitaan halvempi ratkaisu erilaisten kilpailujen tulosten keräämiseen. Järjestelmää käytetään ralli-sprint- ja kelkka-sprint-tapahtumissa.

Tämän opinnäytetyön tilaaja on rovaniemeläinen Lap Data System Oy. Yritys on tarjonnut moottorieurheilukilpailujen ajanmittaus- ja raportointipalveluita 90-luvun alkupuolelta lähtien. Käytössä yrityksellä on omaan suunnitteluun ja toteutukseen perustuvat ohjelmistot. Vielä toistaiseksi aikatiedot kerätään paikalliselle palvelimelle, josta ne lähetetään FTP-ohjelmalla erilliselle julkaisupalvelimelle.

Tämän opinnäytetyön lähtökohtana käytettiin olemassa olevaa mittalaitteistoa, jolle haluttiin toteuttaa automatisoitu aikatietojen tallennusohjelmisto. Projektin tavoitteena on toteuttaa selainpohjainen järjestelmä, jolla tallennetaan rallin aikatiedot palvelimelle käsiteltäväksi ja koostettavaksi.

2 RALLIN HISTORIAA JA KILPAILUTAPAHTUMAN KUVAUS

2.1 Ralliautojen ryhmäjako

Kilpailuissa käytettävät autot jaetaan luokkiin ja ryhmiin. Alla (taulukko 1) on lyhyesti kerrottu yleisimmistä luokista ja niiden ominaisuuksista. Ero A-ryhmän ja N-ryhmän välillä on tekniikassa. Esimerkiksi Mitsubishi Lancer Evolution (N, A 5660) voi kuulua joko A- tai N-ryhmään. Ryhmä selviää auton katsastuspapereista. F-ryhmään kuuluvat N- ja A-ryhmän autot, joiden luokitus on vanhentunut. (Kotala 26.4.2014.)

TAULUKKO 1. Ralliautojen luokkajako.

Ryhmä	Kuvaus
Historic	1972 ja sitä vanhemmat autot. Historic-luokka on jaettu eri kategorioihin auton iän ja moottorin tilavuuden mukaan. Luokan tavoitteena on säilyttää autot ikäkausiensa mukaisina ja kunnioittaa näiden autojen historiaa.
V1600	Vakioautojen luokka. Moottorin kuutiotilavuus ei saa ylittää 1600 cc.
Trophy	A-ryhmän autot, joiden kuutiotilavuus on alle 1400 cc.
N-ryhmä	FIA:n luokittelemat autot, joiden luokitus on vielä voimassa.
A-ryhmä	FIA:n luokittelemat autot, joiden luokitus on voimassa.
F-ryhmä	Sisältää vanhentuneet N- ja A-ryhmän autot.

Jokaiselle automerkille ja mallille pitää olla FIA:n luokitus. Kun auto on luokiteltu, määritellään samalla luokituksen päättymisvuosi. Merkintä f vuosiluvun perässä tarkoittaa, että auto voi osallistua luokittelun päättymisvuoden jälkeen ryhmän F kilpailuihin, mikäli se rakennetaan täyttämään F-ryhmän vaatimukset. Esimerkkinä alla (taulukko 2) on osa Lanciaan malleista ja niiden ryhmistä. (Autourheilu.fi.)

TAULUKKO 2. Autojen luokittelu.

Tunnus	Automalli	Päättymisvuosi
N,A-5214	Delta HF (1585*1.7= 2695 cc) (920kg)	91f
N,A-5324	Delta HF 4WD (1995*1.7= 3392 cc) (1095kg)	92f
N,A-5355	Delta HF Integrale (1995*1.7= 3392 cc) (1116kg)	94f
N,A-5394	Delta HF Integrale 16V (1995*1.7= 3392 cc) (1165kg)	96f

2.2 Erilaisia ralleja

Suomessa ajetaan useita eritasoisia kilpailuja niin talvella kuin kesällä. Suomen suurimpana rallitapahtumana tunnetaan MM-sarjan osakilpailu Neste Oil Rally (Jyväskylän Suurajot). Kotimaisessa SM-rallissa ajetaan 4 tai 5 kilpailua vuosittain ja Suomenmestaruudesta kilpaillaan sekä kaksi- että nelivetoisten luokissa. Lisäksi järjestetään lyhyitä 3 - 12 erikoiskokeen pituisia kansallisia ralleja ja kolmen erikoiskokeen pituisia Rallispecial-kilpailuja. Myös nuorille kuljettajille on oma sarjansa, jossa

kilpaillaan kolmessa eri luokassa. Näiden kilpailujen yhteydessä kilpaillaan myös Rallin Naisten Suomenmestaruudesta. (Kotala 26.4.2014.)

F-ryhmän autoille järjestettävä F-Cup ja vanhemmille autoille järjestettävä Historic Rally Trophy ovat harrastajamääriltään Suomen suosituimmat rallisarjat. Kyseisissä sarjoissa reittiin ei saa tutustua ennakoon, vaan ne ajetaan ns. pimeinä kilpailuina. (Autourheilu.fi.)

2.3 Ralli-sprint ja kelkka-sprint

Ralli-sprintissä ajetaan samalla kalustolla kuin rallissakin. Merkittävin ero on se, että autokuntaan kuuluu ainoastaan kuljettaja (I-ohjaaja); ralleissa on lisäksi pakollinen kartanlukija (II-ohjaaja). Ralli-sprint ajetaan 2 - 5 km pitkällä radalla siten, että kukin kuljettaja ajaa kaksi eri lähtöä. Kelkka-sprint on samankaltainen kuin autolla ajettava sprint-kilpailu. (Kotala 26.4.2014.)

2.4 Rallin ajanoton kehittyminen vuosien saatossa

1970-luvulla ajanotto hoidettiin käsiajanotolla. Erikoiskoehenkilösto nouti ek-materiaalin kilpailukeskuksesta. Materiaaliin sisältyi mm. kaksi kelloa ja runsaasti paperimateriaalia. Kellot olivat sekuntikellotyyppisiä, ja mistä oli teipattu ne napit piiloon, joita ei saanut missään nimessä painaa. Kilpailujen järjestäjä ajasti kellot yleensä silmämääräisesti ja ek-henkilöstö laittoi omat kellonsa kilpailuikaan, koska kisajärjestäjän kellojen rikkoutuessa käytettäisiin ek-henkilöstön kelloja. Ek-materiaalit palautettiin ek:n päättymisen jälkeen kilpailutoimistoon, jossa sihteerit alkoivat laskea tuloksia. Tuloslaskenta kilpailupaikalla toimi niin, että sihteerit käsityönä ruutupaperia apuna käyttäen laskivat tulokset. Muun muassa Tunturirallin tulokset julkaistiin ja palkinnot jaettiin vasta kilpailupäivän jälkeisenä päivänä. Sihteereillä meni useita tunteja tulosten aikaansaamiseksi. Ja kun tulokset oli saatu järjestykseen, ne kirjoitettiin kirjoituskoneella ns. vahalle ja niitä pyöritettiin sitten vahakoneella tarvittava määrä. Myöhemmin kopiokone mullisti tulosten kopioimisen. (Kotala 26.4.2014.)

Esimerkki käsiajanoton ajoilta on Mauri Kotalan kertomus Tunturirallista ajalta, jolloin ralli kiersi koko Lapin läänin alueella. Sevettijärvellä ajettiin erikoiskoe yöllä. Kun tulokset eivät olisi ehtineet paperina ajoissa tuloslaskentaa varten, ne soitettiin lankapuhelimella Rovaniemelle. Yöllä mentiin paikallisen poromiehen taloon, kerrottiin keitä ollaan ja pyydettiin lupaa soittoon. Luvan jälkeen puhelu tilitettiin keskuksesta ja odotettiin, joskus pitkiäkin aikoja, puhelun yhdistymistä. Kilpailutulokset saatiin kuitenkin lopulta viestitettyä kilpailukeskukseen. (Kotala 26.4.2014.)

Kellojärjestelmiin saatiin 1980-luvun loppupuolella merkittävä muutos, kun käyttöön otettiin tulostavat kellolaitteet. Auton ylitettyä maalilinjan painettiin nappia ja kello tulosti nauhalle kelloajan. Tämä ei poistanut vanhaa käsiajanottoa, vaan toimi rinnalla varmistuksena. Samoihin aikoihin tulosviestintään tuli merkittävä muutos ARP-tukiasemien rakentamisen myötä. Autoradiopuhelimilla viestitettiin tulokset kilpailukeskukseen tuloslaskentahenkilöstölle. Tuolloin kilpailun sihteerit eivät enää käsitel-

leet tuloksia, vaan niitä hoitivat asiaan varta vasten nimetyt henkilöt. Näin toimitaan vielä edelleenkin. ARP-tukiasemien kautta saatiin kaikki samaan verkkoon YRP-puhelujen (yritysradiopuhelujen) kautta. Kilpailukeskukset toimivat johtokeskuksena ja kaikki erikoiskokeet olivat samassa verkossa. Eli kaikki kuulivat kaikkia. Tuolloin oli tärkeää radioverkon johtoaseman työskentely. Se johti radioliikennettä ja otti mm. ajat maastosta jokaiselta erikoiskokeelta suoraan kilpailukeskukseen. (Kotala 26.4.2014.)

Samoihin aikoihin käsiteltiin ensimmäiset tuloslaskennat tietokoneella. Levykkeellä käynnistettävät tietokoneet olivat käytössä ja niillä saatiin kilpailijat ja heidän aikansa lajiteltua järjestykseen. Kilpailijan ajat laskettiin aina käsin vähentäen maaliajasta lähtöaika. Tuolloin aikoja ei laskettu vielä tietokoneella ja käsityön vuoksi tuli virheitä. Tietokone toimi siis vain lajittelijana. (Kotala 26.4.2014.)

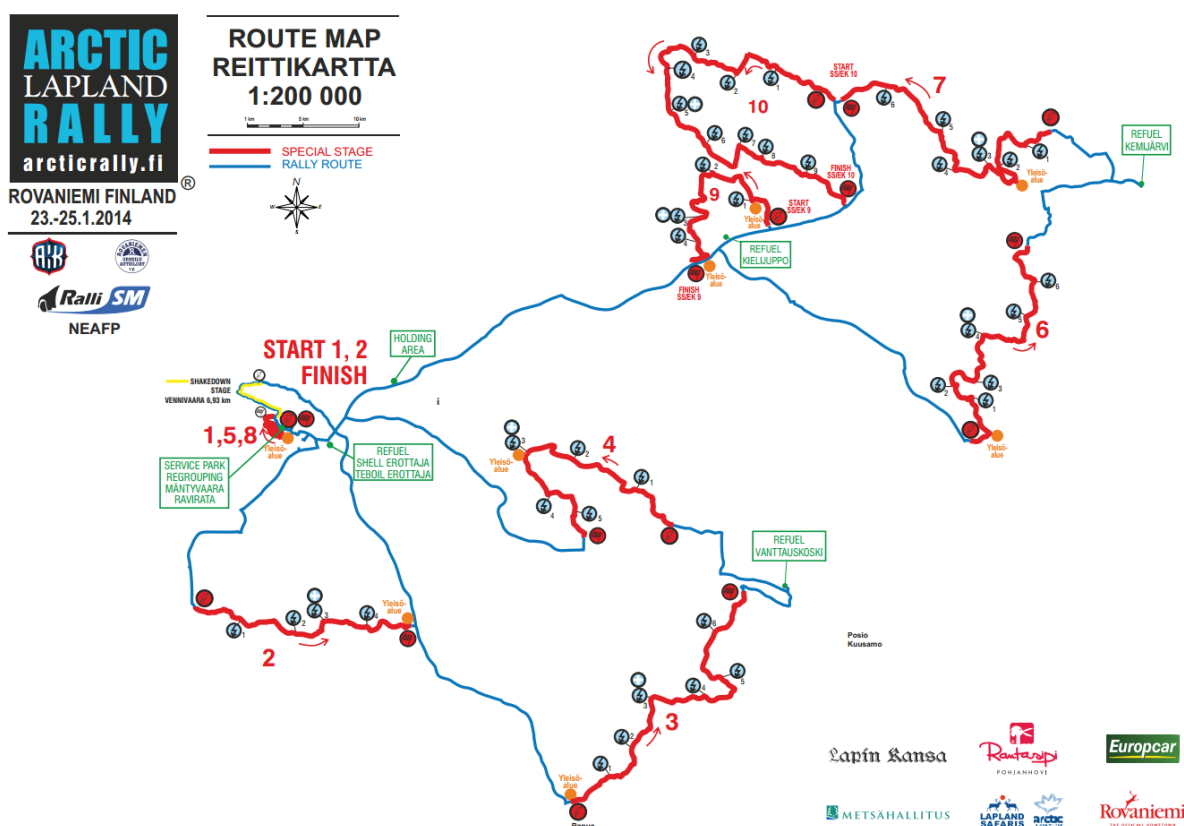
1990-luvun alkupuolella oli käytössä ensimmäiset tietokantaohjelmat, joilla tehtiin tuloslaskenta. Aikojen laskenta siirrettiin maastoon erikoiskoehenkilöstön tehtäväksi, koska kilpailijalle täytyi ilmoittaa erikoiskokeelle käytetty aika. Edelleenkin ajat laskettiin käsin ja sama aika syötettiin myös tuloslaskentaohjelmaan. Tässä järjestelmässä oli joskus paljonkin virheitä, koska ajan laskeminen ei ollut helppoa. Myöhemmin tuloslaskentaohjelmaan tehtiin muutos ja sinne syötettiin kilpailijan lähtö- ja maaliaika ja järjestelmä laski em. tiedoilla käytetyn ek-ajan. Sama menetelmä on käytössä edelleenkin. (Kotala 26.4.2014.)

Itse kellolaitteisiin tuli seuraavat isot muutokset 2000-luvun alkupuolella, kun lähtövaloja ohjasi kello. Aina tasaminuutilla valo vaihtui vihreäksi, muuten valo oli punainen. Viisi sekuntia ennen lähtöaikaa punainen valo alkoi vilkkua ja vihreällä valolla lähdettiin liikkeelle. Nämä kellojärjestelmät mahdollistivat ns. vilppilähtöjen kontrolloinnin, kun se aikaisemmin tapahtui vain ja ainoastaan ek-henkilöstön aistihavaintona. Lisäksi kellolaitteen avulla saadut reaktioaikatiedot olivat tärkeää informaatiota kilpailijoille. Myös kilpailun maaliin tuli sähköinen ajanotto, joten aika saatiin suoraan tekstiviestilaitteelle. Ek-henkilöstö syötti niin lähdössä kuin maalissakin kilpailijanumeron ja valitsi "Lähetä"-painikkeen. Kyseiset aikatiedot lähetettiin tekstiviesteinä tuloslaskentaan ja tuloslaskentahenkilö syötti ajat tuloslaskentaohjelmaan. Niissä kilpailuissa, joissa ei ole tekstiviestilaitetta, ajat otetaan puhelimitse tuloslaskentaan. (Kotala 26.4.2014.)

Internetin myötä tuloksia pystyttiin ja pystytään edelleen seuraamaan lähes reaaliajassa. Nykyisin viiveen aiheuttaa käsityönä tapahtuva tulosten tietokantaan syöttäminen. Virheiden määrä on nykyisessä tuloslaskennassa vähentynyt lähes olemattomiin, vain promillen luokkaan. Suurehkoissa rallissa syötetään käsin tuhansia aikoja, joista korjattavia aikoja on yleensä vain 0 - 10. (Kotala 26.4.2014.)

2.5 Yleiskuvaus kilpailun järjestelyistä

Rallikilpailujen järjestäminen on suuri projekti, johon kuuluu paljon suunnittelua. Alla (Kuva 1) on Rovaniemellä järjestetyn Arctic Lapland Rallyn reittikartta. Reittikartassa punaisella paksulla viivalla on merkittynä erikoiskokeet ja ohuemmat siniset viivat kuvaavat siirtymiä. Lisäksi kuvassa on merkittynä suorakaiteenmuotoisilla tekstikentillä tankkauspisteet, huoltopisteet ja odotusalue.



KUVA 1. Arctic Lapland Rallyn reittikartta (Arctic Lapland Rally.)

Rallireitille laaditaan aikataulus, jota pyritään noudattamaan tarkasti. Alla (taulukko 3) on ote samasta Arctic Lapland Rallyn aikataulusta ja liitteessä 1 on aikataulu kokonaisuudessaan. Aikataulun mukaisesti ensimmäinen kilpailijan on oltava lähtöpaikalla 12:00. Lähtöpaikalta alkaa "Mäntyvaara 1" -niminen siirtymä, jonka pituus on 18,35 km. Siirtymään on varattu aikaa 30 min, joten ensimmäisen kilpailijan on oltava aikataulukastuspisteellä 12:30. Tästä hänellä on 3 min aikaa siirtyä ek 1:n lähtöpaikalle ja keskittyä lähtöön. Ek 1 (Rantasipi Pohjanhovi / Mäntyvaara 1) on pituudeltaan 3,7 km ja ensimmäinen kilpailija lähtee erikoiskokeelle 12:33. Ek 1:n jälkeen on siirtymä "Aittajärvi", jonka pituus on 25,47 km ja sille on varattu 50 min aikaa. Siirtymän jälkeen on samanniminen ek 2, jonka pituus on 24,18 km. Näiden kahden ek:n jälkeen kokoontutaan tekniselle alueelle.

Huoltoon A on varattu aikaa 30 min. Tässä vaiheessa erikoiskokeita on ajettu yhteensä 27,88 km ja siirtymiä yhteensä 72,89 km. Huollon jälkeen auto tankataan ja aloitetaan siirtymä seuraavalle erikoiskokeelle. Tällä samaisella toimintaperiaatteella jatketaan kahden päivän ajan, joiden aikana ajetaan yhteensä 244,32 km erikoiskokeita ja 552,34 km siirtymiä.

TAULUKKO 3. Ote Arctic Lapland Rallyn aikataulusta (Arctic Lapland Rally.)

ARCTIC LAPLAND RALLY 2014 ITINERARY						
LEG 1 FRIDAY 24.1.2014						
TC	Location	SS	Liaison	Total	Target	First
SS		dist.	dist.	dist.	time	car due
0	Start Leg 1					12:00
1	Mäntyvaara 1		18,35	18,35	0:30	12:30
SS 1	Rantasipi Pohjanhovi / Mäntyvaara 1	3,7			0:03	12:33
2	Aittajärvi		25,47	29,17	0:50	13:23
SS 2	Aittajärvi	24,18			0:03	13:26
2a	Regrouping In / Tecnical Area		29,07	53,25	1:00	14:26
	Regrouping Mäntyvaara ravirata				0:10	
2b	Regrouping Out / Service In					14:36
SPA	Service A Mäntyvaara ravirata	27,88	72,89	100,77	0:30	
2c	Service Out / Tecnical Area					15:06
	Refuel, Shell Erottaja / Teboil Erottaja	(27.88)	(79.13)	(107.01)		
3	Siikakämä		52,55	52,55	1:10	16:16
SS 3	Hanaa! / Siikakämä	38			0:03	16:19
	Refuel,Vanttauskoski	(38.00)	(53.09)	(91.09)		
4	Jyrhämäjärvi		22,62	60,62	1:10	17:29

2.5.1 Rallikilpailun tulospalveluhenkilöstö

Tuloslaskennassa on yleensä tuloslaskentapäällikkö, joka kuuluu kilpailuorganisaatioon. Hänen tehtävänä on hoitaa kaikki tulostaulut paikalleen. Lisäksi hän huolehtii siitä, että paikalla on riittävä henkilöstö käsittelemään maastosta tulevia aikakortteja ja pöytäkirjoja. Varsinaisessa tuloslaskennassa tarvitaan 3 tai 4 henkilöä, jotka soittavat aikoja maastosta ja syöttävät ne tuloslaskentajärjestelmään. Lisäksi tarvitaan henkilö, joka huolehtii Internetiin menevistä tuloksista. Henkilöstön määrä on riippuvainen erikoiskokeiden määrästä. Ek:n lähdössä ja maalissa tarvitaan 1 tai 2 henkilöä, jotka kirjaavat ajat pöytäkirjaan ja syöttävät ajat tekstiviestilaitteelle tai antavat puhelimitse aikoja tuloslaskennalle. (Kotala 26.4.2014.)

2.5.2 Kellojen synkronointi ja tulosten toimittaminen tulospalveluhenkilöstölle

Kellot synkronoidaan ennen kisaa synkronointilaitteella. Kaikki kellot kytketään laitteeseen kiinni ja yhdellä ns. "master" -koneella käynnistetään kaikki samanaikaisesti. Nykyisin tulokset toimitetaan tulospalveluhenkilöstölle tekstiviestilaitteella. Mikäli tekstiviestilaitteita ei voida käyttää, tuloslaskenta-henkilöstö soittaa erikoiskokeen maaliin ja kysyy sieltä kilpailijan numeron, lähtöajan ja maalijan. (Kotala 26.4.2014.)

2.6 Yhden erikoiskokeen kuvaus

Erikoiskokeet ovat tieosuuksia, jotka on suljettu yleiseltä liikenteeltä. Kilpailun järjestäjä laittaa reitin varrelle tarvittavan määrän henkilöstöä, jotta reitti voidaan ajaa turvallisesti. Ennen erikoiskokeen alkua kilpailun etuautot tarkistavat toiminnan erikoiskokeiden lähdöissä ja maaleissa. Etuautoja on useita, ja jokaisen tehtävä on tarkastaa oman sektorinsa toiminta. Samalla tarkastus on myös koulutustilaisuus. Ennen varsinaisia kilpa-autoja ajavat nolla-autot: 000-auto, 00-auto ja 0-auto. Nämä autot ovat kilpailulisenssillä olevia kuljettajia, joilla on pitkä kokemus kilpailuista. He tekevät omat arvionsa erikoiskokeesta ja sen ajettavuudesta. Kun nolla-autot ovat päässeet erikoiskokeen maaliin, kilpa-autot päästetään erikoiskokeelle. (Kotala 26.4.2014.)

2.6.1 Erikoiskokeen lähtöalue

Kilpailija saapuu aikatarkistusasemalle aikakortissa olevalla ajalla. Aikatarkistusasemalla kilpailijalle merkitään asemalle saapumisaika ja erikoiskokeelle lähtöaika, joka on yleensä kolme minuuttia aikatarkistusasemalle saapumisen jälkeen. Kyseinen aika on ns. keskittymisaika erikoiskokeelle. Kilpailija siirtyy lähtöviivalle ja hänen aikakorttinsa tiedot tarkastetaan. Ek-henkilöstö merkitsee pöytäkirjaan kilpailijan numeron ja lähtöajan. (Kotala 26.4.2014.)

Lähtöajan lähestyessä punainen lähtövalo alkaa vilkkua. Kilpailija näkee myös isosta kellosta lähtöajan ja pystyy valmistautumaan lähtöön. Hänellä on lupa lähteä erikoiskokeelle, kun vihreä lähtövalo syttyy. Kilpailijan lähdettyä ek-henkilöstö syöttää tekstiviestilaitteeseen kilpailijan numeron ja valitsee lähetä-painikkeen, jolloin laite lähettää tiedot tekstiviestinä tuloslaskentaan. (Kotala 26.4.2014.)

Ek-henkilöstö ohjaa seuraavan kilpailijan lähtöviivalle ja hänen aikakorttinsa tiedot tarkastetaan. Seuraavalla täydellä minuutilla kilpailijalla on lupa lähteä erikoiskokeelle. Näin menetellään niin kauan, kunnes kaikki kilpailijat ovat lähteneet erikoiskokeelle. (Kotala 26.4.2014.)

2.6.2 Erikoiskokeen maalialue

Kilpailija saapuu maaliin ja maalilinjalla sähköinen ajanottolaite ottaa kilpailijalle ajan. Kilpailija ajaa maalilinjalta ns. stoppiautolle, joka on jarrutusmatkan päässä. Yleensä matka maalilinjalta stoppiautolle on noin 200 metriä. Stoppiautolla olevaan tekstiviestilaitteeseen on automaattisesti välitetty maalilinjasta ylityksestä aika. Kilpailijan numero syötetään tekstiviestilaitteeseen ja valitaan Lähetä-painike, jolloin aika lähetetään tuloslaskennalle. Samalla ek-henkilöstö täyttää kilpailijan aikakortin ja merkitsee pöytäkirjaan kilpailijan numeron ja maalijan. Maalilinjalla on myös kilpailijan aikatiedot paperille tulostava varajärjestelmä automaattisen järjestelmän lisäksi. Automaattisen järjestelmän vikaantua käytetään varajärjestelmän tulostamia tietoja ja ne välitetään tuloslaskennalle puhelimitse. (Kotala 26.4.2014.)

2.7 Tuloslaskennan periaatteet

Ek-henkilöstö tallentaa lähdöstä ja maalista aikatiedot pöytäkirjaan. Nämä samat aikatiedot välitetään puhelimitse myös henkilöstölle, joka syöttää ne tuloslaskentajärjestelmään. Tuloslaskentajärjestelmästä aikatiedot siirtyvät julkaisupalvelimen internetsivulle. Tulosten siirto on automatisoitu tapahtuvaksi tietyin väliajoin, jolloin viivettä voi tulla noin minuutin verran. (Giant.)

Erikoiskokeen tulossivu päivittyy niin kauan, kunnes kaikki kyseisen erikoiskokeen aikatiedot on otettu ja ajanottohenkilöstö on ne tarkastanut. Tulossivun ajat ovat epävirallisia niin kauan, kunnes tuomaristo on hyväksynyt tulokset. Tulosten hyväksymisen jälkeen koostetaan erilliset lopputulokset ja ne lajitellaan luokittain. Lopputuloksista julkaistaan myös dokumentit esim. PDF-muodossa. Dokumentteissa on kerrottu lopputulokset, erikoiskokeiden tulokset, keskeyttäneet, peruneet, suljetut, rangaistusajat ja tilastot. (Giant.)

Epävirallisissa tuloksissa voi olla virheitä monista syistä. Virhemahdollisuuksia ovat esim:

- Puhelinyhteydet voivat olla heikot, jolloin tuloslaskenta vastaanottopäässä voi kuulla aikatiedon väärin.
- Maastosta tuloksia antava henkilö voi tulkita aikakortissa olevia käsin kirjoitettuja numeroita väärin (vertaa esim. 1 ja 7).
- Tuloslaskenta voi syöttää aikatiedon väärin tuloslaskentajärjestelmään.

Kun kaikki kilpailijat ovat saapuneet erikoiskokeen maaliin, toimihenkilöt tarkistavat asiapaperit ajanottopäällikön johdolla. Aikatiedoista löytyneet virheet korjataan ja dokumentteihin lisätään mahdolliset rangaistusajat ja tarkistetut tulokset viedään tulostaululle. On äärimmäisen tärkeää, että kilpailijat käyvät tarkistamassa omat aikansa tulostaululta. Virheen havaitessaan kilpailija voi ottaa yhteyden kilpailun sihteeriin tai ajanottopäällikköön tulosten korjaamiseksi. (Giant.)

3 TYÖKALUJEN VALINTA JA JAVA-SOVELMA

3.1 NetBeans ohjelmointityökaluna

Toimeksiantajalla ei ollut erityisiä vaatimuksia käytettävistä kehitystyökaluista. Haluttiin kuitenkin käyttää avoimen lähdekoodin ohjelmistoja, koska todettiin niiden täyttävän kehitystyön edellytykset. Aiemmat hyvät kokemukset NetBeans:sta Java-kehityksessä puolsivat sen valintaa tähänkin projektiin. Tutun työkalun valinta nopeuttaa sovelluskehitystä, ja tulosta saadaan aikaan suhteellisen nopeasti.

Opinnäytetyön toteutuksen aikana Javan päivityksen yhteydessä sen tietoturvaan tuli muutoksia. Näiden muutosten myötä Java-sovelman allekirjoitus osoittautui pakolliseksi; luvussa 3.3 on kuvaus allekirjoituksen toteuttamisesta. Java-sovelman kehitystyön aikana toteutettiin automatisoitu testi tiedoston käsittelyyn. Testi lisättiin projektiin, jotta voitiin muutosten jälkeen varmistua tulosten tallennuksen toimivuudesta. Kyseinen testi toimii myös esimerkkinä mahdollisesti muille testeille, joten testin lisäämisestäkin (luku 3.5) on sisällytetty kuvaus tähän dokumenttiin.

3.2 Java-sovelma

Java-sovelma (applet) on Web-sivulle liitettävä ohjelma, joka ohjelmoidaan Java-ohjelmointikielellä. Ohjelmointikoodi käännetään sopivalla Java-kääntäjällä ns. luokkatiedostoksi ja liitetään osaksi Web-sivua. Java-sovelmat jaetaan käyttäjille yleensä palvelimen kautta ja yleensä käyttäjä käynnistää Java-sovelman selaimen kautta, mutta itse Java-sovelma suoritetaan Java-virtuaalikoneessa (JVM). Java-sovelman vahvuuksia ovat sen kohtuullisen helppo integroiminen Web-sivulle ja mahdollisuus käyttää laitteen resursseja. Lisäksi Java-sovelma on käyttöjärjestelmäriippumaton, ja juuri käyttöjärjestelmäriippumattomuus oli yksi tämän projektin tavoitteista. (Korpela.)

3.3 Java-sovelman allekirjoitus, avaimen luonti

Huhtikuussa 2013 Oracle julkaisi Javaan päivityksen, jonka myötä kaikkien Java-sovelmien tulee olla allekirjoitettuja. Aiemminkin Java-sovelmat on voitu allekirjoittaa, mutta se ei ollut pakollista. Päivityksen myötä Java Control Panel -sovelluksen tietoturva-asetuksiin tehtiin muutoksia ja nykyisin käyttäjä voi asetuksista säätää liukusäätimellä halutun tietoturvatason. Tässä projektissa Java-sovelman kehitystyön aikana käytettiin ns. self-signed-sertifikaattia, niin että välttyttiin sertifikaatin ostamiselta. (Oracle. Java-sovelman allekirjoitus.)

Työkalu Java-sovelman allekirjoitukseen tulee Java Runtime Environment (JRE) -paketin mukana ja se löytyy oletusasennuksen jälkeen "C:\Program Files\Java\jre7\bin" -hakemistosta. Komentokehoite käynnistetään administrator-oikeuksin ja avain luodaan seuraavan komennon mukaisesti:

```
keytool -genkey -keyalg RSA -alias selfsigned -keystore keystorefile.jks  
-storepass password -validity 360 -keysize 2048
```


Alla on käyttäjältä komennon suorituksen yhteydessä kysyttävät kysymykset esimerkkipastauksineen:

```

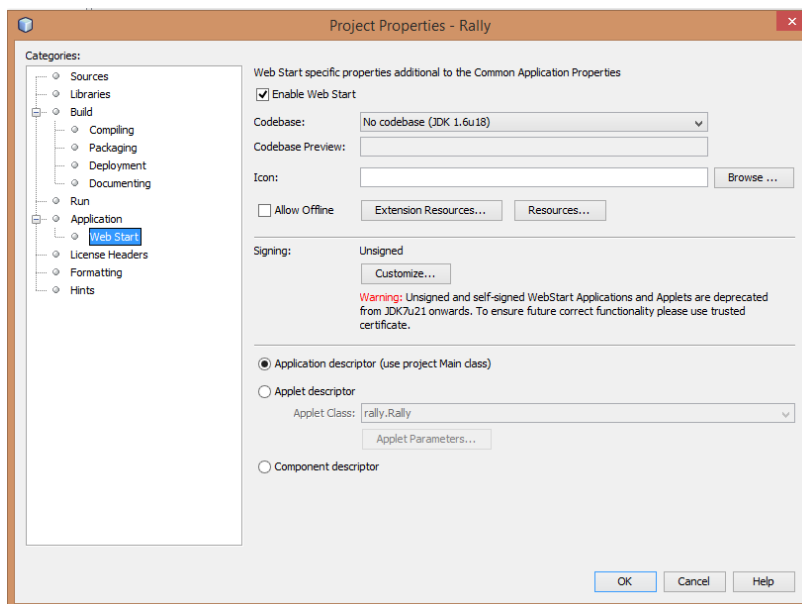
What is your first and last name?
    [Unknown]: Marko
What is the name of your organizational unit?
    [Unknown]: Student
What is the name of your organization?
    [Unknown]: Student
What is the name of your City or Locality?
    [Unknown]: Kuopio
What is the name of your State or Province?
    [Unknown]: Kuopio
What is the two-letter country code for this unit?
    [Unknown]: FI
Is CN=Marko, OU=Student, O=Student, L=Kuopio, ST=Kuopio, C=FI correct?
    [no]: yes
Enter key password for <selfsigned>
    (RETURN if same as keystore password):

```

Komennon parametreistä huomattavia ovat **–storepass**, jota tarvitaan, kun avainta otetaan käyttöön ja **–keystore**, jonka mukaan avaintiedosto nimetään. Komennon parametreistä löytyy tarkempaa tietoa Oraclen Web-sivuilta. (Keytool.)

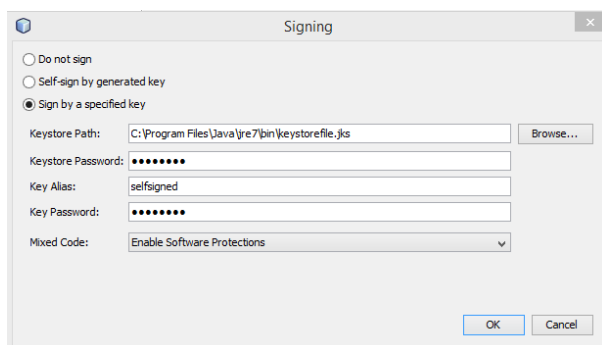
3.4 Java-sovelman allekirjoitus, avaimen käyttöönotto NetBeans:ssa

NetBeans:ssä allekirjoitusavain lisätään projektiin allekirjoitettavan projektin ominaisuuksia (Properties) esittelevästä näkymästä (Kuva 2). Toiminnot avaimen lisäämiseksi projektiin on sijoitettu Web Start -osioon. Näkymän yläreunassa on valintaruutu Enable Web Start, jonka valinta mahdollistaa asetuksien muokkaamisen. Näkymän alareunassa on kohta Applet desciptor, jolla valitaan se luokka, joka halutaan suorittaa Java-sovelman käynnistyessä. Valintaruudun Allow Offline valinta mahdollistaa Java-sovelman käyttämisen ilman jatkuvaa yhteyttä palvelimeen. (Signed applets.)



KUVA 2. Netbeans-projektin ominaisuudet.

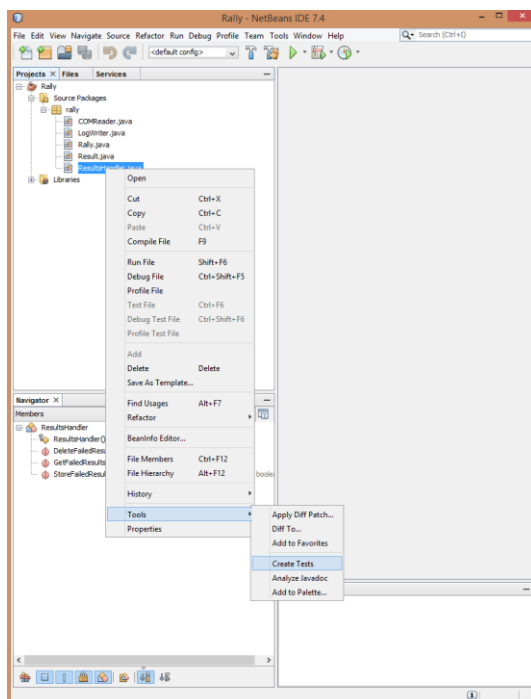
Näkymän (Kuva 2) keskellä on osio Signing, jonka Customize-painikkeella siirrytään toteuttamaan Java-sovelman allekirjoitus (Kuva 3). Allekirjoituksessa käytetään aiemmassa luvussa 3.3 keytool-työkalun yhteydessä syötettyjä tietoja. Projektin käännöksen yhteydessä allekirjoitus lisätään käännettyihin tiedostoihin ja siitä tiedotetaan NetBeans:n Output-ikkunassa.



KUVA 3. Allekirjoituksen lisääminen.

3.5 TestNG-testin lisääminen NetBeans-projektiin

Testaamiseen on olemassa useita erilaisia viitekehyksiä (framework), joilla on omat vahvuutensa. JUnit-kirjasto tulee NetBeans IDE:n mukana ja sen käyttö on tullut tutuksi jo aiempien projektien myötä. Se tarjoaa useimpiin tapauksiin riittävät ominaisuudet, mutta ei mahdollista testien suoritusjärjestyksen asettamista. Yksi suoritusjärjestyksen mahdollistava viitekehys testaukseen on TestNG, jota tässä projektissa käytettiin. TestNG:n ja JUnit:n koodien ulkoasu muistuttaa hyvin paljon toisiinsa, joten TestNG:n valinta tähän projektiin on perusteltua. (TestNG.)

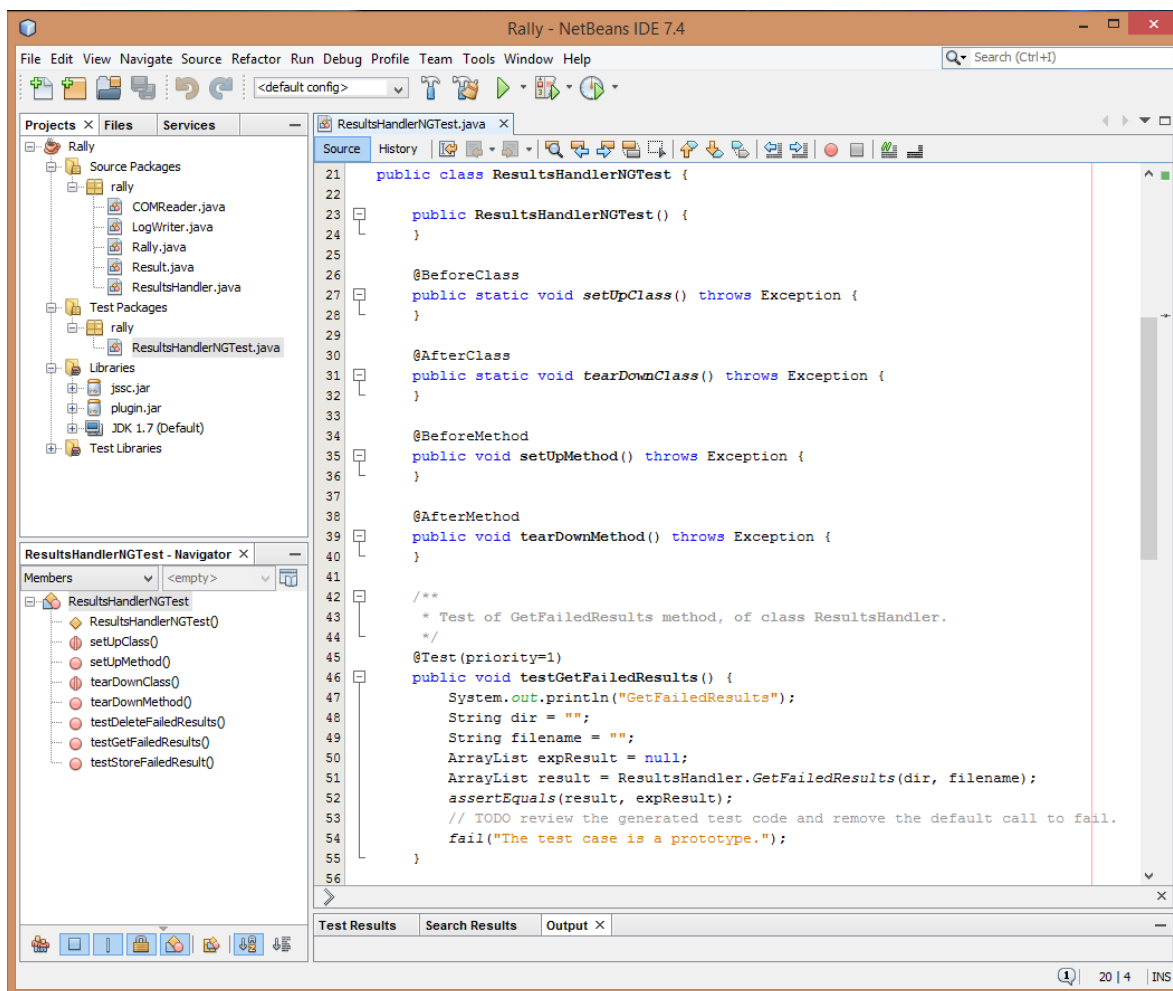


KUVA 4. Testien lisäys.

Testien lisääminen NetBeans:ssa tapahtuu työkalu-valikon (Tools) kautta (Kuva 4) Create Tests -toiminnon avulla. Seuraavassa vaiheessa käyttäjällä on mahdollisuus valita käytettävä viitekehys ja automaattisesti generoitava koodi. TestNG generoi oletusasetuksilla valmiiksi koodirungon testien toteuttamiseen (Kuva 5).

@BeforeClass-kommentin määrittelemä metodi suoritetaan tämän testiluokan aluksi. Tässä osiossa valmistaudutaan testien suorittamiseen eli alustetaan testiympäristö tunnettuun tilaan. Vastaavasti @AfterClass-kommentin määrittelemässä metodissa palautetaan järjestelmä tilaan, jossa se oli ennen tämän testiluokan suorittamista. Näiden kahden funktion tarkoitus on, että testaus ei sotke järjestelmää. @BeforeMethod-kommentin määrittelemä metodi suoritetaan jokaisen yksittäisen testin aluksi ja @AfterMethod-kommentin määrittelemä metodi jokaisen testin lopuksi.

@Test-kommentilla määritellään testit ja niille voidaan määritellä suoritusjärjestys (Kuva 5) lisäämällä kommentin perään "*priority=1*", jossa numero on testinsuorituksen järjestysnumero. Tässä projektissa priorisoinnilla pyrittiin välttämään testien vaatimaa valmistelua. Testit toteutettiin niin, että ensimmäinen testi kirjoittaa tiedostoon, toinen lukee sieltä ja kolmas poistaa tietoa. Tällaisessa testaustavassa on otettava huomioon testien epäonnistuminen. Tämän projektin kohdalla näissä testeissä luodaan uusi erikseen nimetty tiedosto, joten testien epäonnistuminen ei vaikuta järjestelmän toimintaan.



KUVA 5. Näkymä testien luonnin jälkeen.

3.6 Eclipse ohjelmointityökaluna

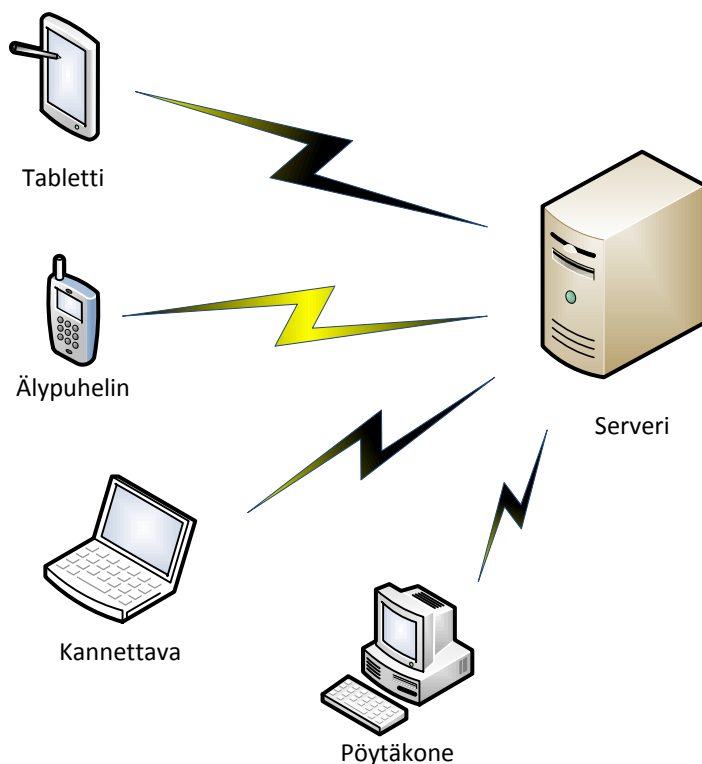
Tässä projektissa käytettiin Eclipseä työkaluna Web-sivun tekemiseen HTML-kielellä, PHP-kirjaston toteuttamiseen ja JavaScripteihin. Näissä osissa valmiiksi käännettävää koodia ei ole, joten Eclipse toimi käytännössä tekstieditorina. Eclipseen on saatavilla hyvin erilaisia teemoja, joilla eri koodikielten ulkoasua saadaan muokattua. Koodin luettavuus näytöltä paranee huomattavasti, kun esimerkiksi kommentit ovat eri värillä kuin toiminnallinen koodi. Nämä ominaisuudet löytynevät myös muista sovelluskehittimistä, mutta aiempi tuntemus Eclipsestä puolsi sen valintaa.

4 AJANOTTOJÄRJESTELMÄN TOTEUTUS

4.1 Tavoitteet

Projektin tavoitteena oli lukea sarjaportista kellolaitteen lähettämä aikatieta ja lähettää se palvelimelle tietokantaan tallennettavaksi. Työn tilaajalla on jo olemassa toimivat sarjaporttiin kytkettävät kellolaitteet, joiden mittaama aika toimitetaan nykyisellään osittain käsityönä tietokantaan. Opinnäytetyön tavoitteena oli siis lisätä järjestelmän automatisointia ja siten vähentää inhimillisten virheiden määrää ja nopeuttaa tulosten tallennusta tietokantaan. Samalla tulosten reaaliaikainen seuranta mahdollistui, niin että selostajien ja yleisön on mielekkäämpää seurata kilpailua.

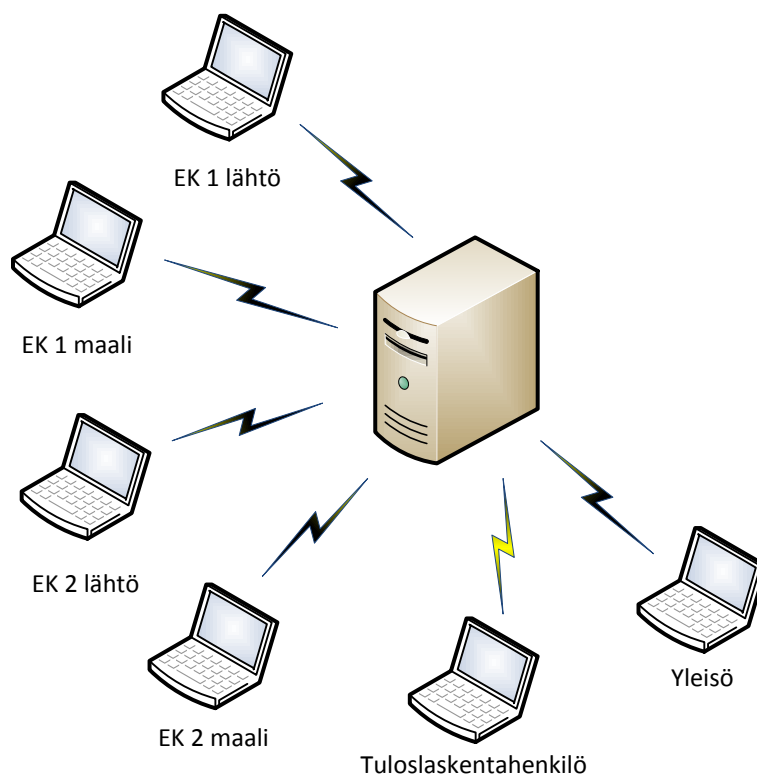
Sovellus haluttiin toteuttaa selainpohjaisena ylläpidettävyyden ja käyttöliittymän ulkoasun muokkaamisen vuoksi. Selainpohjaisuutta tuki myös tavoite tehdä ratkaisusta alustariippumaton, jotta järjestelmää voidaan käyttää erilaisilla laitteilla (Kuva 6). Haluttiin myös toteuttaa sovellus niin, että ohjelmistoja ei tarvitse asentaa kilpailupaikoilla käytettäville tietokoneille tai muille vaihtoehtoisille laitteille.



KUVA 6. Mahdollisuus käyttää erilaisia laitteita järjestelmän kanssa.

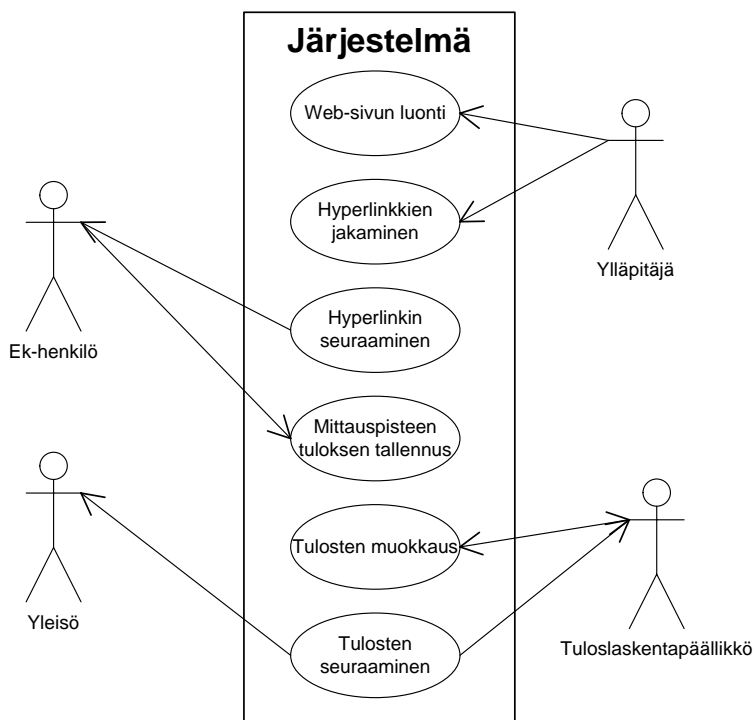
Jatkossa voi tulla tarve käyttää muitakin kuin sarjaporttiliitännäisiä mittalaitteita, joten selaintoteutuksessa tällöin tarvitsee korvata vain Java-sovelman osuus. Myös itse Java-sovelman toteutuksessa on otettu huomioon mahdolliset muut liitännät jakamalla Java-sovelman toiminnot eri luokkiin.

Kilpailupaikalla on monia mittauspisteitä (Kuva 7), joista jokaisesta lähetetään mittausdataa palvelimelle. Erikoiskokeilta tarvitaan mittaustieto vähintään lähdöstä ja maalista, jotta voidaan laskea erikoiskokeelle käytetty ajoaika. Tuloslaskentahenkilöt ja yleisö tulevat näkemään ajat reaaliaikaisesti omilla laitteillaan. Lisäksi esimerkiksi tuloslaskentapäällikölle voidaan antaa järjestelmään muokausmahdollisuus, jolloin hän voi korjata virheelliset tulokset. Mahdollisia muutoksia vaativia tapauksia voivat olla esimerkiksi väärän auton numeron syöttäminen mittauspisteellä tai kellolaitteen hajoaminen, jolloin turvaudutaan käsin tehdyn mittauksen tulokseen.



KUVA 7. Palvelin ja käyttäjät.

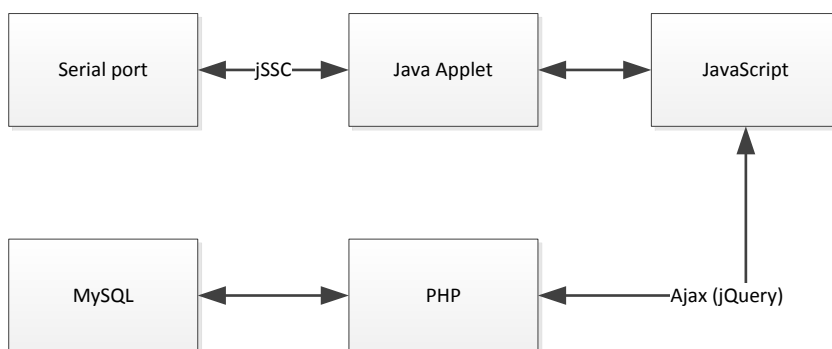
Järjestelmää voi käyttää monet henkilöt, joilla on erilaiset oikeudet ja toiminnot (Kuva 8) järjestelmän käytössä. Järjestelmän ideana on, että ylläpitäjä muodostaa tarvittavat Web-sivut jokaiselle mittauspisteelle ja toimittaa hyperlinkin esimerkiksi sähköpostilla ek-henkilölle. Tällöin ek-henkilö seuraa hyperlinkkiä ja hänelle avautuu käyttöliittymä. Sähköpostin mukana toimitettavan linkin yhteydessä määritellään käyttöoikeudet. Käyttötapauskaaviossa yleisö kuvaa niin kilpailupaikalla olevaa kuin kotona kilpailua seuraavaa henkilöä. Reaaliaikaisten tulosten myötä kuka tahansa voi seurata kilpailun tuloksia Web-sivun kautta.



KUVA 8. Järjestelmän käyttötapauskavio.

4.2 Yleiskuvaus järjestelmästä

Edellä mainitut tavoitteet huomioiden päädyttiin toteuttamaan järjestelmä, jossa Java-sovelma käyttää (Kuva 9) jSSC-kirjastoa sarjaportin lukemiseen. Sarjaportin lukuun valittiin jSSC-kirjasto, koska se on ilmainen ja monipuolinen kokonaisuus. Valintaa puolsi myös laaja käyttöjärjestelmä tuki, sillä niin Windows ja Linux kuin myös Mac OS X ja Solaris kuuluvat tuettujen käyttöjärjestelmien joukkoon. (Sokolov.)



KUVA 9. Kuvaus sovelluksen rakenteesta.

Java-sovelma käyttää JavaScriptin objekteja (tässä sovelluksessa funktioita) JObject-luokan avulla ja välittää siten mittaus tuloksen selaimen käyttöön. Mittaus tuloksen lähetyks palvelimelle tapahtuu jQuery-kirjaston metodeilla Ajax-tekniikkaa käyttäen. Palvelimella on PHP-kirjasto, joka huolehtii mittaus datan käsittelystä, kuten esimerkiksi aikojen muuttamisesta haluttuun muotoon. Lisäksi PHP-kirjasto tallentaa tiedot MySQL-tietokantaan.

4.3 Sarjaporttiyhteys jSSC-kirjastolla

Nykyisin käytössä olevat ajanottolaitteet kytketään sarjaporttiin, joten heti projektin alussa oli selvää, että sarjaporttiyhteyttä tarvitaan. Selaimella ei kuitenkaan ole suoraa yhteyttä tiedostojärjestelmään tai laitteisiin, joten ei myöskään sarjaportin käsittelyyn.

Asian selvittelyn yhteydessä löydettiin tieto, että Javalle saatavilla oleva jSSC-kirjasto mahdollistaa Java-sovelman ja sarjaportin kommunikoinnin. JSSC-kirjaston testaukseen toteutettiin yksinkertainen Java-sovelma ja testauksen yhteydessä todettiin jSSC-kirjaston täyttävän projektin sarjaporttiyhteyden vaatimukset. Java-sovelmaan lisättiin toiminnallisuuksia projektin edetessä tarpeen mukaan. Tässä projektissa jSSC-kirjaston tarjoamasta laajasta metodijoukosta käytettiin vain muutamaa metodia sarjaportin käsittelyyn.

4.3.1 Sarjaporttien listaaminen getPortNames-metodilla

Sarjaporttiin yhdistämiseen jSSC-kirjastolla tarvitaan sarjaportin nimeä. Etukäteen ei voida tietää erilaitteiden käyttämää sarjaporttien nimiä, joten getPortNames-metodia tarvitaan porttien nimien etsimiseen. Metodi palauttaa merkkijonotaulukon porttien nimistä, mutta koska Javan ja Javascriptin välisessä kommunikoinnissa taulukoiden välittäminen ei ole suositeltua, merkkijonot yhdistetään sovitulla erottimella yhdeksi merkkijonoksi. Aikaansaatu yksi merkkijono välitetään sitten JavaScriptille, joka erottelee merkkijonosta sarjaporttien nimet ja listaa ne selaimella näytettäväksi.

4.3.2 Sarjaportin alustus ja käyttöönotto

Sarjaportin käyttöönotossa sille (tässä kappaleessa sarjaportilla tarkoitetaan jSSC-kirjaston SerialPort-tyyppistä oliota) asetetaan tiedonsiirtoasetukset ja jSSC-kirjaston tapauksessa se tapahtuu siten, että ensin avataan sarjaportti openPort-metodilla. Seuraavaksi sarjaportille asetetaan yhteysparametrit setParams-metodilla. Metodi tarvitsee seuraavat parametrit:

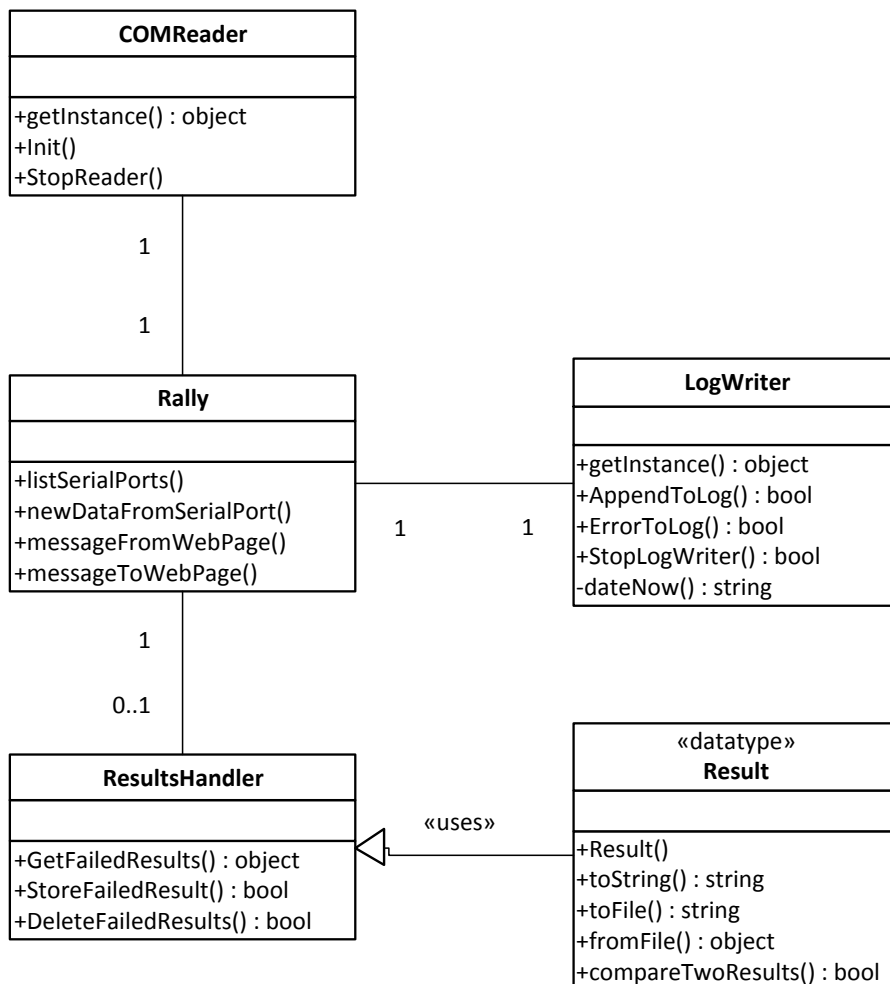
- baudRate 9 600 kbit
- dataBits 8
- stopBits 1
- parity 0

Lisäksi sarjaportille pitää kertoa, mitä kaikkia tapahtumia tarkkaillaan. Tässä projektissa ollaan kiinnostuneita vain saapuvasta datasta, joten setEventsMask-metodille annetaan parametrina jSSC-kirjaston MASK_RXCHAR-vakio.

Seuraavaksi sarjaporttiin liitetään addEventListener-metodin avulla jSSC-kirjaston SerialPortEventListener, joka ottaa vastaan sarjaportin tapahtumat. Tähän projektiin toteutettiin SerialPortEventListeneriin toiminto, joka välittää kaikki sarjaporttiin tulleet merkit Java-sovelman Rally-luokalle (Kuva 10).

4.3.3 Kuvaus Java-sovelman rakenteesta

Tässä projektissa Java-sovelma jaettiin viiteen luokkaan (Kuva 10). Rally-luokka perii JApplet:n ja toimii tässä projektissa ns. pääluokkana. Rally-luokan tehtävänä on hallita muita luokkia ja käsitellä datan välittämistä niiden välillä. COMReader-luokan tehtävänä on lukea sarjaporttiin saapuneet merkit jSSC-kirjaston avulla ja välittää ne Rally-luokalle. Sekä COMReader-luokka että LogWriter-luokka on toteutettu singleton-suunnittelumallin mukaisesti siten, että ne voidaan toteuttaa vain kerran.



KUVA 10. Java-sovelman luokkakaavio.

ResultsHandler-luokka ja sen käyttämä tietotyyppi toteutettiin valmiiksi myöhempää käyttöä varten. Tästä luokasta tehtiin staattinen, joten sitä ei tarvitse toteuttaa, vaan sitä voidaan käyttää suoraan. ResultsHandler-luokan tehtäviin kaavailtiin toiminnot, joiden avulla voidaan tallentaa mittaustulokset väliaikaisesti tiedostoon ja lukea ne tiedostosta. Näille toiminnoille voi olla tarvetta, jos tietoliikenneyhteys katkeaa ja mittaustuloksia ei saada lähetettyä palvelimelle. Tällainen tilanne on hyvinkin todennäköinen, koska kilpailupaikat ovat usein syrjäseuduilla.

Mittauslaitteistojen yhteys palvelimeen on suunniteltu toteutettavaksi USB-modeemien (ns. mokkuloiden) avulla, joten myös katsojien viestiliikennelaitteet käyttävät mahdollisesti samaa televerkkoa. Kilpailujen aikana GSM-verkko kuormittuu tavallista enemmän suuren käyttäjämäärän takia eivätkä teleoperaattorit todennäköisesti ole varautuneet tällaisiin satunnaisiin kuormituksiin.

4.4 JavaScript-funktiot

JavaScript-funktiot toimivat Java-sovelman ja palvelinyhteyden yhdistävänä osana (Kuva 9). Lisäksi JavaScriptien tehtävänä on päivittää käyttöliittymän kenttien tiedot, esimerkiksi sarjaportista saatu lähtöaika. Kuitenkin yksi JavaScriptien tärkeimmistä tehtävistä on Java-sovelman käyttöönotto.

NetBeans generoi valmiiksi Web-sivun ja sinne koodin, jolla Java-sovelma otetaan käyttöön. Tässä projektissa keskitettiin JavaScript-koodit yhteen paikkaan hallinnan helpottamiseksi, joten NetBeans:n generoima koodi kopioitiin keskitettyyn paikkaan. Koodin kopiointin vuoksi tiedostoviittaukset täytyi korjata osoittamaan oikeaan paikkaan WWW-palvelimella. "archive"-attribuutin tiedostopolut kertovat Java-sovelman tiedostojen sijainnin WWW-palvelimella. Seuraavassa generoidussa koodissa on tiedostopolut korjattu viittaamaan oikeaan paikkaan.

```
var attributes = {
    code: "rally.Rally",
    archive: "java/Rally.jar, java/lib/jssc.jar, java/lib/plugin.jar",
    name: "jsap",
    id: "jsap",
    width: 15,
    height: 15};
var parameters = {jnlp_href: "launch.jnlp"};
var version = "1.7";
deployJava.runApplet(attributes, parameters, version);
```

Alla (Taulukko 4) kuvataan lyhyesti toteutettujen JavaScript-funktioiden tehtävät. Skripteihin toteutettiin testaukselle välttämättömät toiminnot, jotta saatiin varmuus järjestelmän soveltuvuudesta suunniteltuun tarkoitukseen. Myöhemässä vaiheessa esimerkiksi eri mittauspisteitä varten tarvitaan omanlaisensa funktiot.

TAULUKKO 4. Javascript-funktioiden tehtävät.

Funktio	Tehtävä
saveStartDetails	Kerää tiedot Web-sivulta ja lähettää ne palvelimelle tallennettavaksi tietokantaan.
openSelectedPort	Tarkistaa käyttöliittymästä käyttäjän valitseman sarjaportin ja lähettää tiedon siitä Java-sovelmalle. Käytetään siinä tapauksessa, että laitteesta löytyy useita sarjaportteja.
updateCOMPorts	Java-sovelma lähettää viestissä tiedon laitteesta löytyneistä sarjaporteista JavaScriptille, joka tässä funktiossa käsittelee viestin ja päivittää käyttöliittymään löytyneet sarjaportit.
messageFromApplet	Käsittelee Java-sovelman lähettämät viestit. Pyrittiin keskittämään Java-sovelman ja JavaScriptien kommunikointi ylläpidettävyyden helpottamiseksi.

Tässä projektissa JavaScript keskustelee palvelimen kanssa AJAX-tekniikkaa käyttäen POST-metodilla. Seuraava esimerkkikoodi lähettää palvelimelle kahden muuttujan (startCarNumber ja startTime) arvot. AJAX-metodille annetaan parametreina nimi-arvo pareja, joiden avulla määritellään AJAX-pyyntöasetukset. Esimerkissä onnistuneen pyynnön päätteeksi näytetään onnistumisilmoitus ja sen lisäksi palvelimen lähettämä viesti. Vastaavasti virhetilanteessa näytetään virheviesti.

```
$.ajax({
  type: "POST",
  url: "php/saveTime.php",
  data: {"startCarNumber":startCarNumber,
        "startTime":startTime},
  success: function (msg) {
    alert("OK: " + msg.msg);
  },
  error: function (msg) {
    alert("Error: " + msg.msg);
  }
});
```

4.5 PHP-kirjasto

Palvelimelle valittiin PHP-ohjelmointikieli sen todella laajan tuettavuuden vuoksi. Lisäksi valintaa puolsi aiempi kokemus PHP-ohjelmointikielestä. Alla (Taulukko 5) on lista PHP-kirjastoon toteutetuista tiedostoista ja lyhyt kuvaus niiden tehtävistä.

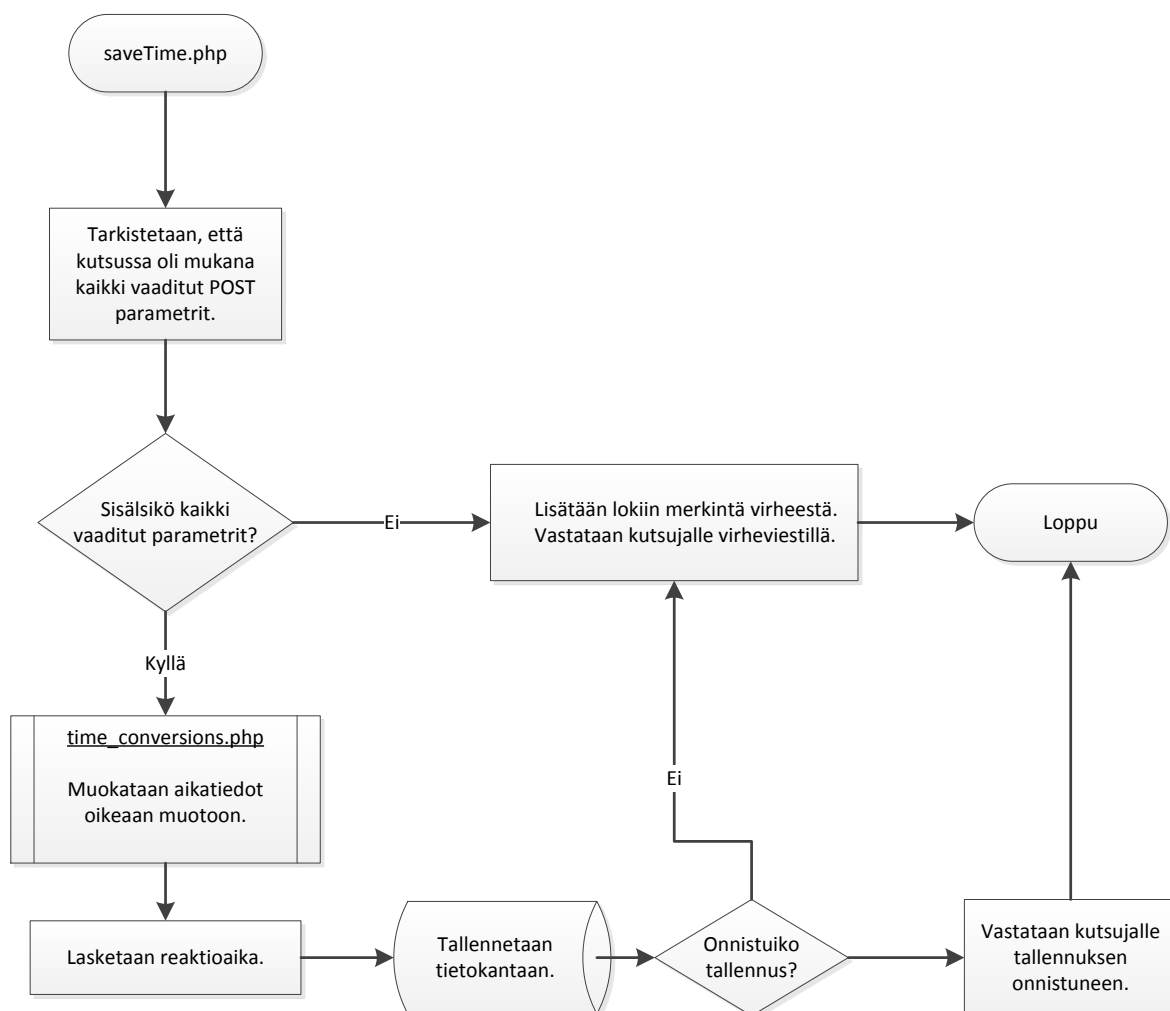
TAULUKKO 5. PHP-kirjaston tiedostot ja niiden tehtävät.

Tiedosto	Tehtävä
db_ajat.php	Nämä tiedostot on nimetty tietokannan taulujen mukaan. Näiden pääasiallinen tehtävä tässä vaiheessa on tallentaa tietokantaan uusia rivejä. Lisäksi raportointia ajatellen on jo valmiiksi toteutettu funktiot, joilla voidaan hakea tietoa kannasta.
db_kesk.php	
db_kilpailijat.php	
db_kilpailut.php	
db_lahdot.php	
db_luokat.php	
db_sarjat.php	
db_connection.php	Avaa ja sulkee yhteyden tietokantaan. Projektin kehitysvaiheessa myös tietokannan käyttäjätunnukset säilytettiin tässä tiedostossa.
Logging.php	Tallentaa loki-tiedostoon PHP-kirjaston tapahtumia. Jokaiselle päiväykselle muodostetaan oma loki-tiedosto, johon lisätään kellonajan ja viestin sisältävä merkintä.
saveTime.php	Tallentaa tässä vaiheessa lähtöpaikalta lähetetyt tiedot ja lisäksi laskee reaktioajan.
time_conversions.php	Muokkaa aikatiedot sopivaan muotoon. Tietokantaan tallennuksessa ja reaktioajan laskemisessa aikojen täytyy olla double-tyyppisinä.

Alla esitellään lohkokaaviokuvaus (Kuva 11) lähtöajan tallentamisesta. Aluksi kaikki parametrit tarkastetaan ja lasketaan puuttuvien parametrien määrä. Mikäli havaitaan puutteita, niin puutteet merkitään lokiin ja vastataan kutsujalle virheviestillä.

Käyttöliittymästä palvelimelle tuodut aikatiedot ovat tekstimuotoisia ja sisältävät ajan lisäksi ajanotolaitteen tunnusteen. Edellä mainitusta syystä johtuen aikatiedot täytyy muokata tietokantaan double-tyypisiksi määriteltyyn muotoon. Aikatietojen käsittely toteutettiin erilliseen tiedostoon ylläpidettävyyden selkeyttämiseksi.

Reaktioajan laskemisen jälkeen tiedot tallennetaan tietokantaan ja tallennuksen onnistumisen tilasta lähetetään käyttäjälle viesti. Tietokantaan tallennuksen yhteydessä käytetään transaktiota, jolloin tallennuksen epäonnistuessa tapahtuma perutaan. Mahdollisesta epäonnistumisesta merkitään lokiin virhe ja lähetetään kutsujalle virheviesti.



KUVA 11. Lohkokaaviokuvaus ajan tallennuksesta tietokantaan.

4.6 MySQL-tietokanta

Toimeksiantajalla oli jo projektin alkaessa valmiina tietokannan runko, johon projektin edetessä tehtiin tarvittaessa muutoksia. Aiempi tietokanta on toteutettu MySQL-relaatiotietokantaohjelmistolla, joten myös tähän projektiin se oli luonnollinen valinta. Tietokannan ja PHP-kirjaston kehittämiseen käytettiin XAMPP-ympäristöä, joka on saatavilla niin Windows kuin Linux- ja Apple-ympäristöihin. XAMPP-paketti sisältää mm. MySQL-tietokantaohjelmiston, PHP:n ja Apache HTTP-palvelimen. Näiden ominaisuuksien avulla palvelimen sovelluskehitysympäristö oli helppo pystyttää ja kehitystyöhön päästiin nopeasti. (XAMMP.)

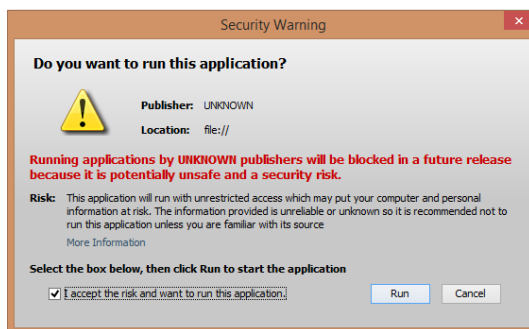
Ensin tietokanta suunniteltiin MySQL Workbench -työkalulla, josta tallennettiin tietokannan luontilauseet tiedostoon. Sen jälkeen luontilauseet ajettiin MySQL-kantaan ja lisättiin kantaan vähän dataa testausta varten. Näiden toimien jälkeen tehtiin PHP-kirjastoon tietokantayhteyteen tarvittavat osat ja testattiin taulujen lukemista. Tietokantaa muutettiin useaan kertaan ja samalla PHP-kirjasto päivitettiin vastaamaan uutta tietokannan rakennetta. Lähinnä muutokset liittyivät taulujen sarakkeisiin, joten ne olivat kohtuullisen helppoja muokata myös PHP-kirjastoon.

Tämän opinnäytetyön lopullisessa versiossa tietokantaan tallennetaan seuraavat asiat:

- kilpailujen tiedot mm. päivä ja nimi
- kilpailuun kuuluvien sarjojen tiedot mm. nimi ja lähtöjen määrä
- sarjoihin kuuluvien luokkien tiedot mm. nimi
- tiedot sarjojen lähdöistä mm. nimi ja pituus
- kilpailijoiden tiedot
- kilpailijoiden ajat lähdöillä
- tiedot keskeyttäneistä kilpailijoista.

4.7 Käyttöliittymä ja tietoturvaravitukset

Työn toimeksiantajan toiveena oli toteuttaa käyttöliittymä HTML-pohjaisena, jolloin se voidaan muokata kulloiseenkin tapahtumaan sopivaksi. Tästä syystä tässä opinnäytetyössä käyttöliittymän ulkoasuun ei panostettu, vaan toteutettiin yksinkertainen ja järjestelmän testaukseen soveltuva käyttöliittymä. Kun käyttäjä avaa linkin avulla Web-sivun, hänelle näytetään tietoturvaravitukset (Kuva 12). Tämän varoitus näytetään, koska käytössä on ns. self-signed-allekirjoitus.



KUVA 12. Java-sovelman tietoturvaroitus.

Tietoturvaroituksen jälkeen käyttäjälle näytetään käyttöliittymän Web-sivu. Käyttöliittymän vasemmassa yläreunassa listataan laitteesta löydetty sarjaportit. Mikäli laitteesta löytyy vain yksi sarjaportti, niin se valitaan automaattisesti. Käyttäjä valitsee sarjaportin, johon ajanotto-laite on kytketty, kuvan (Kuva 13) vasemmasta ylänurkasta.

EK1 lähtö

Serial ports [COM1 is selected]

COM1
 COM3

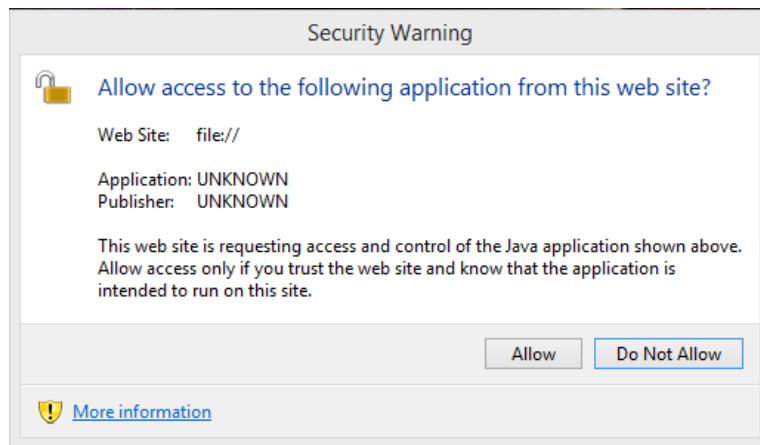
Serial port messages will be stored to: c:\temp\Log_20140404024540.txt

Seuraava			Historia	
Käpailu	ID <input type="text"/>	Käpailun nimi	14:04:00	64
Sarja	ID <input type="text"/>	Sarjan nimi	14:03:00	63
Lähtö	ID <input type="text"/>	Lähdön nimi		
Lahtöaika	<input type="text"/>			
Numero	<input type="text"/>			
Lahtöaika COM portista	<input type="text"/>			
<input type="button" value="Tallenna"/>				

KUVA 13. Käyttöliittymä.

Sarjaporttilistan alla kerrotaan käyttäjälle loki-tiedoston sijainti ja nimi. Alaosassa on kentät tallennettaville tiedoille. ID-kentät ovat kehitysvaiheessa esillä vain sovelluksen testausta helpottamaan, sillä ne ovat pakollisia tietokantaan tallennuksessa. Ek-henkilön syötettäväksi jäävät ihannelähtöaika ja auton numero. Kellolaitteelta tulee todellinen lähtöaika alimmaiseen kenttään heti kun kellolaite havaitsee kilpailijan lähteneen erikoiskokeelle. Täytettävien kenttien oikealla puolella on esimerkkinä sopivat arvot. Lisäksi käyttöliittymään on suunniteltu kerättäväksi tieto viimeisimmistä erikoiskokeelle lähteneistä kilpailijoista.

Sarjaportin valinnan yhteydessä käyttäjälle esitetään vielä yksi varoitus (Kuva 14), jonka hyväksynnän jälkeen sarjaportti on valittu ja mittaustulosten kerääminen voidaan aloittaa.



KUVA 14. Varoitus JavaScriptin yhteydestä Java-sovelmaan.

5 JATKOKEHITYSAJATUKSIA

Projektin alussa ideoitiin kokonainen järjestelmä, mutta kaikkea ei ehditty toteuttamaan. Tässä esitellään projektissa toteuttamattomia suunnitelmia ja pohditaan mahdollisia jatkokehitysideoita. Projektin edetessä varmistui, että järjestelmän kokonaisvaltainen kehittäminen on mahdollista annetuilla ehdoilla.

Java-sovelmaan liittyvien tiukkojen tietoturva-asetusten mukana seurasi kohtalaisen paljon uutta ja selvitystä vaativaa asiaa. Joidenkin asioiden selvittämiseen meni yllättävän paljon aikaa, ja projektin aikana eri Java-versioillakin ilmaantui uusia virheilmoituksia. Osan ongelmista ja virheilmoituksista ratkaisee todennäköisesti ostettu allekirjoitus Java-sovelmalle.

Määrittelyvaiheessa suunniteltiin toiminto, jonka avulla tuloslaskentahenkilö voisi seurata kilpailijoiden etenemistä erikoiskokeella. Suunnitelman mukaisesti käyttäjälle näytettäisiin kilpailijalista ja erikoiskokeiden numerot. Erikoiskokeiden numeroiden taustaväri vaihtuisi kilpailijan edetessä erikoiskokeilla, jolloin tulospalveluhenkilöstö voisi jo tämän toiminnon avulla arvioida mahdollisia ulosajoja. Taustaväri olisi punainen, kun kilpailijalla ei ole lähtöaikaa, eli ennen erikoiskokeelle lähtöä. Taustaväri muutettaisiin keltaiseksi, kun kilpailijalla on lähtöaika ja hän on lähtenyt erikoiskokeelle. Taustaväri muutettaisiin vihreäksi, kun kilpailijalla on lähtöaika ja maaliintuloaika eli kun hän on saapunut maaliin.

Myös kotona kisoja seuraavalle yleisölle mietittiin erilaisia raportteja, joiden ulkoasu tai toteutusta ei vielä kovin pitkälle suunniteltu. Alla olevassa raporttimallissa (Kuva 15) tulokset päivitetäisiin automaattisesti esim. yhden sekunnin välein, jolloin kilpailutilanteen seuraaminen olisi sujuvaa.

Ektulokset							Tilannetulokset						
Sja	No	I-Ohjaaja Auto	Lk	Lk Sja	Ek Aika	Ero 1.:een	Sja	No	I-Ohjaaja Auto	Lk	Lk Sja	Yhteensä Aika	Ero 1.:een
1.	8	RUOKANEN V Mitsubishi	1	1	17.03,6	**,**	1.	8	RUOKANEN V Mitsubishi	1	1	17.03,6	**,** (**,**)
2.	2	HYTTINEN K Mitsubishi	1	2	17.18,6	-15,0	2.	2	HYTTINEN K Mitsubishi	1	2	17.18,6	-15,0 (-15,0)
3.	6	VAINIONPÄÄ J Mitsubishi	1	3	17.24,5	-20,9	3.	6	VAINIONPÄÄ J Mitsubishi	1	3	17.24,5	-20,9 (-05,9)
4.	1	KLASILA T Mitsubishi	1	4	17.37,1	-33,5	4.	1	KLASILA T Mitsubishi	1	4	17.37,1	-33,5 (-12,6)
5.	7	VIITANEN M Mitsubishi	1	5	17.38,1	-34,5	5.	7	VIITANEN M Mitsubishi	1	5	17.38,1	-34,5 (-01,0)
6.	14	TEPPO J Honda	2	1	18.16,9	-01.13,3	6.	14	TEPPO J Honda	2	1	18.16,9	-01.13,3 (-38,8)
7.	11	KORVOLA S Ford	2	2	18.30,8	-01.27,2	7.	11	KORVOLA S Ford	2	2	18.30,8	-01.27,2 (-13,9)
8.	13	RINTAMÄKI A Honda	2	3	18.42,3	-01.38,7	8.	13	RINTAMÄKI A Honda	2	3	18.42,3	-01.38,7 (-11,5)

KUVA 15. Raportti erikoiskokeelta. (Lap Data System Oy.)

LÄHTEET

ARCTIC LAPLAND RALLY. Arctic Lapland Rally 2014 aikataulu [verkkajulkaisu]. [Viitattu 2014-04-29.] Saatavissa: <http://www.arcticrally.fi/index.php/kilpailijat/aikataulu>

AUTOURHEILU.FI. Autojen luokitus [verkkajulkaisu]. [Viitattu 2014-04-30.] Saatavissa: <http://www.autourheilu.fi/attachements/2012-12-17T14-01-29175.pdf>

GIANT. Näin tulostenlaskenta etenee [verkkajulkaisu]. [Viitattu 2014-04-29.] Saatavissa: <http://www.giant.fi/giant2014/tulosten%20tarkastusohje.pdf>

KEYTOOL. Java-sovelman allekirjoittaminen [verkkajulkaisu]. [Viitattu 2014-05-06.] Saatavissa: <http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html>

KORPELA, Jukka. Java-sovelmat ja Flash-tekniikka [verkkajulkaisu]. [Viitattu 2014-05-09.] Saatavissa: <https://www.cs.tut.fi/~jkorpela/webjulk/3.4.html>

KOTALA, Mauri 2014-04-26. Projektivastaava. [Haastattelu.] Oulu.

LAP DATA SYSTEM OY. Rallien tulosraportteja [verkkajulkaisu]. [Viitattu 2014-04-29.] Saatavissa: <http://www.lapdatasystem.com/>

ORACLE. Java-sovelman allekirjoitus [verkkajulkaisu]. [Viitattu 2014-04-29.] Saatavissa: <http://www.oracle.com/technetwork/java/javase/tech/java-code-signing-1915323.html>

SIGNED APPLETS. Allekirjoituksen käyttöönotto [verkkajulkaisu]. [Viitattu 2014-05-06.] Saatavissa: <http://forums.netbeans.org/ptopic38680.html>

SOKOLOV, Alexey. jSSC-kirjasto. [verkkajulkaisu]. [Viitattu 2014-04-29.] Saatavissa: <https://code.google.com/p/java-simple-serial-connector/>

TESTNG. Testaus-viitekehys [verkkajulkaisu]. [Viitattu 2014-05-04.] Saatavissa: <http://testng.org/doc/index.html>

XAMMP. Kehitysympäristö MySQL:lle ja PHP:lle [verkkajulkaisu]. [Viitattu 2014-05-06.] Saatavissa: <https://www.apachefriends.org/index.html>

LIITTEET

Liite 1. Arctic Lapland Rally 2014 -kilpailun aikataulu

ARCTIC LAPLAND RALLY 2014 ITINERARY						
LEG 1 FRIDAY 24.1.2014						
TC	Location	SS	Liaison	Total	Target	First
SS		dist.	dist.	dist.	time	car due
0	Start Leg 1					12:00
1	Mäntyvaara 1		18,35	18,35	0:30	12:30
SS 1	Rantasipi Pohjanhovi / Mäntyvaara 1	3,7			0:03	12:33
2	Aittajärvi		25,47	29,17	0:50	13:23
SS 2	Aittajärvi	24,18			0:03	13:26
2a	Regrouping In / Technical Area		29,07	53,25	1:00	14:26
	Regrouping Mäntyvaara ravirata				0:10	
2b	Regrouping Out / Service In					14:36
SPA	Service A Mäntyvaara ravirata	27,88	72,89	100,77	0:30	
2c	Service Out / Technical Area					15:06
	Refuel, Shell Erottaja / Teboil Erottaja	(27.88)	(79.13)	(107.01)		
3	Siikakämä		52,55	52,55	1:10	16:16
SS 3	Hanaa! / Siikakämä	38			0:03	16:19
	Refuel, Vanntauskoski	(38.00)	(53.09)	(91.09)		
4	Jyrhämäjärvi		22,62	60,62	1:10	17:29
SS 4	Lapin Safarit / Jyrhämäjärvi	29,89			0:03	17:32
5	Mäntyvaara 2		50,36	80,25	2:00	19:32
SS 5	Europcar / Mäntyvaara 2	3,7			0:13	19:45
5a	Technical Area In		1,69	5,39	0:10	19:55
	Technical Area				0:04	
5b	Technical Area Out / Service In					19:59
SPB	Service B Mäntyvaara ravirata	71,59	127,22	198,81	0:45	
5c	Service Out / Technical Area					20:44
5d	Finish Leg 1 Lordiaukio		7,21	7,21	0:20	21:04
5e	Parc Ferme 1 Sampokeskus		0,35	0,35	0:05	21:09
	TOTAL LEG 1	99,47	207,67	307,14		

LEG 2 SATURDAY 25.1.2014						
TC	Location	SS	Liaison	Total	Target	First
SS		dist.	dist.	dist.	time	car due
5f	Start Leg 2					7:00
5g	Tecnical Area In		7	7	0:15	7:15
	Tecnical Area				0:04	
5h	Tecnical Area Out / Service In					7:19
SPC	Service C Mäntyvaara ravirata		(14.56)	(14.56)	0:10	
5i	Service Out / Tecnical Area					7:29
	Refuel, Shell Erottaja / Teboil Erottaja	(33.59)	(88.85)	(122.44)		
6	Hanhikoski		82,26	82,26	1:20	8:49
SS 6	Hanhikoski	32,4			0:03	8:52
	Refuel, Kemijärvi	(32.40)	(93.22)	(125.62)		
7	Sarriojärvi		29,28	61,68	1:10	10:02
SS 7	Sarriojärvi	36,35			0:03	10:05
	Refuel, Kielijuppo	(36.35)	(38.66)	(75.01)		
8	Mäntyvaara 3		84,6	120,95	2:00	12:05
SS 8	Arina / Mäntyvaara 3	3,7			0:03	12:08
8a	Regrouping In / Tecnical Area		1,69	5,39	0:10	12:18
	Regrouping Mäntyvaara ravirata				0:10	
8b	Regrouping Out / Service In					12:28
SPD	Service D Mäntyvaara ravirata	72,45	197,83	270,28	0:30	
8c	Service Out / Tecnical Area					12:58
	Refuel, Shell Erottaja / Teboil Erottaja	(3.70)	(69.87)	(73.57)		
9	Heinunkierto		55,62	55,62	1:00	13:58
SS 9	City Hotel / Heinunkierto	23,4			0:03	14:01
	Refuel, Kielijuppo	(23.40)	(54.27)	(77.67)		
10	Ristilampi		27,97	51,37	1:00	15:01
SS 10	Europcar / Ristilampi	49			0:03	15:04
	Refuel, Kielijuppo	(49.00)	(33.38)	(82.38)		
10a	Finish Leg 2 Shell Napapiiri		48,61	97,61	1:30	16:34
HA	Holding Area				1:00	
HA1	Holding Area Out					17:34
POD	Finish Podium Lordiaukio		7,64	7,64	0:15	17:49
	TOTAL LEG 2	144,85	344,67	489,52		
	TOTAL LEG 1	99,47	207,67	307,14		
	TOTAL LEG1+LEG2	244,32	552,34	796,66		