

AKVR

Junatietopalvelu

Ahonen Valtteri
Kukko Joni

Harjoitustyö
12/2015

Ohjelmistotekniikan koulutusohjelma
Tekniikan ja liikenteen ala



Sisällys

1	Asennus	2
2	Tietoa ohjelmasta	3
2.1	Toteutetut toiminnalliset vaatimukset	3
2.1.1	Junien aikataulut ja tiedot	3
2.1.2	Asemien aikataulut	4
2.1.3	Myöhästymistilastot	5
2.1.4	Myöhästymisien syy-tilastot	6
2.2	Toteuttamatta jääneet toiminnot	7
2.2.1	Junat kartalla	7
2.2.2	Metatietojen esittäminen	7
2.2.3	Juna- ja asemahakujen päivämäärät	7
3	Käyttöliittymä	9
3.1	Bootstrap	9
3.2	Ulkoasuvalinnat	9
3.3	Master Page	9
4	Ohjelman riippuvuudet	10
4.1	Ulkoiset kirjastot	10
4.2	Tiedostot ja paikallinen data	10
5	Ongelmat ja jatkokehitys	11
5.1	Ongelmat	11
5.2	Jatkokehitys	11
6	Opittua	12
7	Tekijät	13
7.1	Front end – Ahonen	13
7.2	Back end – Kukko	14

1 Asennus

Sovellus ei tarvitse erillistä asentamista vaan sen voi kääntää lähdekoodeista suoraan toimivaksi kokonaisuudeksi. Tietokantana toimii VR:n avoin rajapinta, ja testidata on valmiina App_Data kansiossa.

Sovelluksen ainoa kirjastoriippuvuus on Newtonsoft.Json json-kirjasto, jota tarvitaan VR:ltä tulevan datan helppoon deserialisointiin.

Konfiguroitavia asioita web.config-tiedostossa ovat VRpath, jolla merkataan VR:n avoimen rajapinnan osoitteeseen, ja AKVRproduction, jolla kerrotaan ohjelmalle käytetäänkö VR:n rajapintaa vai paikallista datavarastoa.

2 Tietoa ohjelmasta

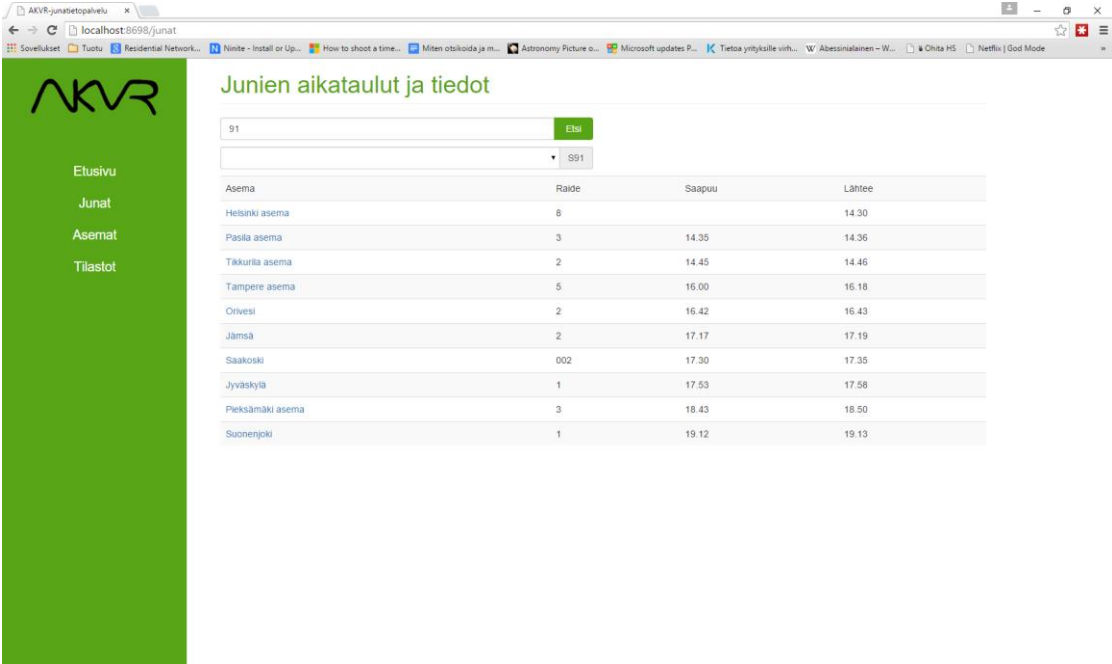
2.1 Toteutetut toiminnalliset vaatimukset

2.1.1 Junien aikataulut ja tiedot

Junat-sivulla pystytään hakemaan annetun junatunnuksen avulla tiedot kyseisen junan aikatauluista ja asemista, joilla juna pysähtyy. Näkyvillä on myös aseman raide, jolta asiakkaat voivat nousta kyytiin. Aikataulutiedoista näkyy asemakohtaisesti saapumis- ja lähtöaika.

Samalle sivulle toteutettiin myös aseman nimellä junien hakeminen. On todennäköistä, ettei käyttäjä muista junatunnusta, mutta tietää aseman, jolta nousee kyytiin. Jos kirjoituskenttään antaa aseman nimen tai nimen alun, niin järjestelmä hakee listan sillä asemalla pysähtyvistä junista. Käyttäjä voi tästä kyseisestä listasta valita haluamansa junan ja järjestelmä toimii taas kuten aikaisemmin kuvattu.

Asemien nimet taulukossa toimivat samalla linkkinä asemat-sivulle suorittaen samalla automaattisesti kyselyn klikatun aseman nimellä.



Asema	Raide	Saapuu	Lähtee
Helsinki asema	8		14.30
Pasila asema	3	14.35	14.36
Tikkurila asema	2	14.45	14.46
Tampere asema	5	16.00	16.18
Orivesi	2	16.42	16.43
Jämsä	2	17.17	17.19
Saakoski	002	17.30	17.35
Jyväskylä	1	17.53	17.58
Pekämäki asema	3	18.43	18.50
Suonenjoki	1	19.12	19.13

2.1.2 Asemien aikataulut

Asemat-sivulla käyttäjä pystyy hakemaan aseman nimen tai nimen alun perusteella kyseisellä asemalla pysähtyvät junat aikatauluineen ja ennusteineen. Haetusta datasta näytetään junan päivämäärä, määränpää, raide millä juna pysähtyy, aikataulu ja aikataulun paikkaansa pitävyyden ennuste sekä junanumero. Nämä tiedot ovat näkyvillä kahdessa eri listassa joista toinen näyttää asemalle saapuvat ja toinen lähtevät junat.

Määränpäänä oleva aseman nimi toimii myös linkkinä tälle samalle sivulle ja sen klikkaaminen suorittaa automaattisesti saman haun kyseistä asemaa kohti. Junan numero toimii linkkinä junat-sivulle, missä suoritetaan automaattisesti haku kyseistä junaa kohti.

Asemien aikataulut

Yliv Etsi

Saapuvat

Pvm	Määränpää	Raide	Aikataulu	Ennuste	Junan numero
1.12.2015	Rovaniemi	002	14.01		IC49
1.12.2015	Vaasa	002	17.55		S57
1.12.2015	Hämeenlinna	001	22.28		T5012
2.12.2015	Helsinki asema	002	0.26		IC266
2.12.2015	Rovaniemi	002	14.01	14.01	IC49
2.12.2015	Vaasa	002	17.55		S57
2.12.2015	Tampere tavara		19.51		T5086
2.12.2015	Hämeenlinna	001	22.28		T5012
3.12.2015	Helsinki asema	002	0.26		IC266

Lähtevät

Pvm	Määränpää	Raide	Aikataulu	Ennuste	Junan numero
1.12.2015	Rovaniemi	002	14.05		IC49
1.12.2015	Vaasa	002	17.59		S57
1.12.2015	Hämeenlinna	001	22.41		T5012
2.12.2015	Helsinki asema	002	0.35		IC266
2.12.2015	Rovaniemi	002	14.05	14.05	IC49
2.12.2015	Vaasa	002	17.59		S57
2.12.2015	Tampere tavara		20.10		T5086

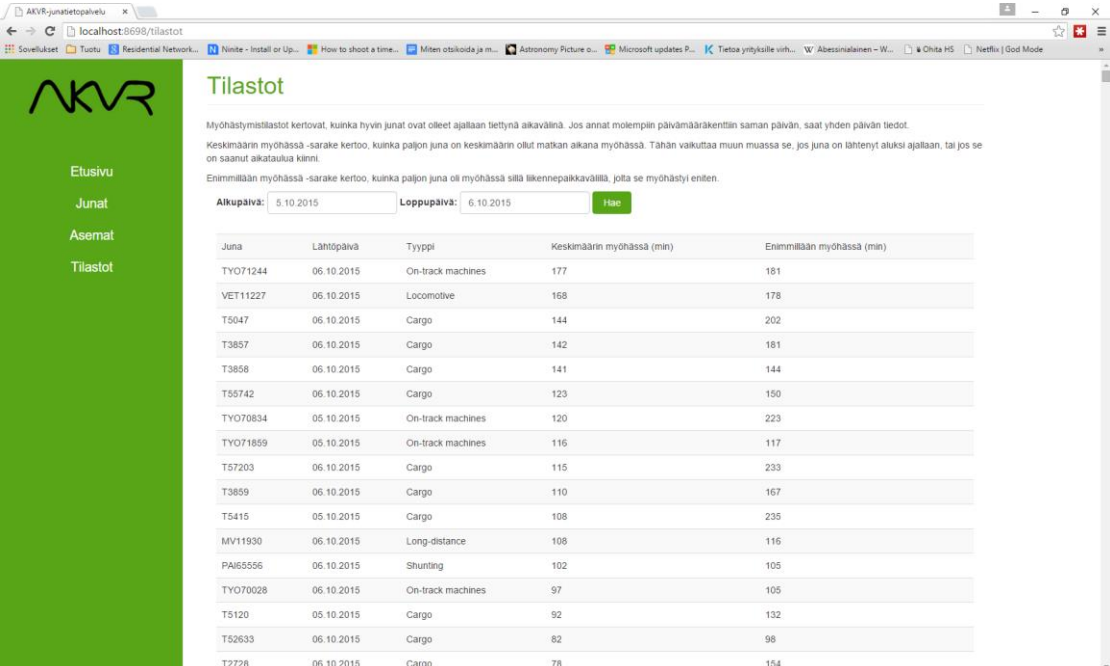
2.1.3 Myöhästymistilastot

Tilastot-sivulla käyttäjän on mahdollista hakea aikavälin avulla tietoa junien myöhästymisistä. Listassa näytetään junan numero, lähtöpäivä, junan tyyppi ja minuutteina kyseisen junan keskimäärin myöhässä ja enimmillään myöhässä tiedot.

VR tarjoaa dataa vain junan asemakohtaisesta myöhästymisestä, joten koko junaa koskevat tilastot joudutaan laskemaan huomattavasti suuremmasta data määrästä. Tästä johtuen pitkän aikavälin hakemisessa saattaa kestää huomattava tovi.

Keskimäärin myöhässä -sarake kertoo, kuinka paljon juna on keskimäärin ollut matkan aikana myöhässä. Tähän vaikuttaa muun muassa se, jos juna on lähtenyt aluksi ajallaan, tai jos se on saanut aikataulua kiinni.

Enimmillään myöhässä -sarake kertoo, kuinka paljon juna oli myöhässä sillä liikennepaikkavälillä, jolta se myöhästyi eniten.



AKVR

Tilastot

Myöhästymistilastot kertovat, kuinka hyvin junat ovat olleet ajallaan liitynä aikavälinä. Jos annat molempien päivämääräkenttien saman päivän, saat yhden päivän tiedot. Keskimäärin myöhässä -sarake kertoo, kuinka paljon juna on keskimäärin ollut matkan aikana myöhässä. Tähän vaikuttaa muun muassa se, jos juna on lähtenyt aluksi ajallaan, tai jos se on saanut aikataulua kiinni.

Enimmillään myöhässä -sarake kertoo, kuinka paljon juna oli myöhässä sillä liikennepaikkavälillä, jolta se myöhästyi eniten.

Aikupäivä: 6.10.2015 Loppupäivä: 6.10.2015 **Hae**


Juna	Lähtöpäivä	Tyyppi	Keskimäärin myöhässä (min)	Enimmillään myöhässä (min)
TYO71244	06.10.2015	On-track machines	177	181
VET11227	06.10.2015	Locomotive	168	178
T5047	06.10.2015	Cargo	144	202
T3857	06.10.2015	Cargo	142	181
T3858	06.10.2015	Cargo	141	144
T55742	06.10.2015	Cargo	123	150
TYO70834	05.10.2015	On-track machines	120	223
TYO71859	05.10.2015	On-track machines	116	117
T57203	06.10.2015	Cargo	115	233
T3859	06.10.2015	Cargo	110	167
T5415	05.10.2015	Cargo	108	235
MV11930	06.10.2015	Long-distance	108	116
PAI5556	06.10.2015	Shunting	102	105
TYO70028	06.10.2015	On-track machines	97	105
T5120	05.10.2015	Cargo	92	132
T52633	06.10.2015	Cargo	82	98
T2728	06.10.2015	Cargo	78	154

2.1.4 Myöhästymisien syy-tilastot

Myöhästymisien syyt-sivu oli viime hetken lisäys toiminnallisuuteen. Tällä sivulla käyttäjältä kysytään päivämäärä, ja järjestelmä hakee kyseisen päivän kaikki syyt mitkä johtivat junan myöhästymiseen asemakohtaisesti.

Listauksessa näytetään tieto junasta, junan tyypistä, asemasta jolla myöhästyminen tapahtui ja lista syistä mitkä johtivat junan myöhästymiseen tällä asemalla.

Vr ei suoraan tarjoa tämän tyylistä listaa vaan kyseinen data joudutaan parsimaan huomattavasti suuremmasta datamäärästä ja tästä syystä joidenkin kiireisten päivien hakeminen saattaa kestää huomattavan ajan. Järjestelmä hakee junien kaikki historiatiedot kyseiseltä päivältä ja hakee listasta junat, joilla on ainakin yksi asema, jolle on merkitty ainakin yksi myöhästymissy poistaen samalla kaiken muun ylimääräisen tiedon.



Etusivu

Junat

Asemat

Tilastot

Myöhästymis syyt

Tilastot Viimeisimmät syyt

Paivamaara 27.11.2015 Hae

marraskuu 2015

ma

ti

ke

to

pe

la

su

26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Junat		Asema	Syy
IC1		Parikkala	Liikennetekniset syyt - Junakohtaus, edellä kulkeva juna tai sivuutus
IC1		Kitee	Matkustajapalvelu - Pysähtymisajan ylitys
S2		Kouvola asema	Matkustajapalvelu - Aikataulusta poikkeava pysähdys
IC4		Imatra asema	Liikennetekniset syyt - Junakohtaus, edellä kulkeva juna tai sivuutus
IO6	Long-distance	Joensuu asema	Moottorijunat ja vaunut -
S7	Long-distance	Imatra asema	Liikennetekniset syyt - Junakohtaus, edellä kulkeva juna tai sivuutus
IC9	Long-distance	Kouvola asema	Liikennetekniset syyt - Tulojuna myöhässä
PVV32	Long-distance	Vainikkala asema	Matkustajapalvelu - Passi- ja tultarkastus
PVV32	Long-distance	Kouvola asema	Muut syyt - Muut häiriöt
PVV32	Long-distance	Lahti	Rata - Radan kunnossapito- ja rakennustyöt
IC40	Long-distance	Parkano	Junan kokoonpano - Kaluston allennettu Sn
S57	Long-distance	Seinäjoki asema	Matkustajapalvelu - Pysähtymisajan ylitys
IO66	Long-distance	Mikkeli	Liikennetekniset syyt - Junakohtaus, edellä kulkeva juna tai sivuutus
IO66	Long-distance	Kouvola asema	Liikennetekniset syyt - Junakohtaus, edellä kulkeva juna tai sivuutus

2.2 Toteuttamatta jääneet toiminnot

2.2.1 Junat kartalla

Yksi ensimmäisistä ideoista harjoitustyötä varten oli kopioida jo olemassa olevan junat kartalla –sovelluksen päätoiminto. VR:ltä tulevaa dataa tutkittaessa paljastui kuitenkin se, että datasta on saatavilla irti vain radan osa millä juna sijaitsee.

Vaikka nämä radan osat ovat vain muutamia satoja metrejä pitkiä ja sen puolesta sijainti voitaisiinkin määrittää suhteellisen tarkkaan, niin radan osien maantieteellisestä sijainnista oli hyvin huonosti tietoa. Radan osaan oli merkitty etäisyys ratajärjestelmän alkupisteestä metreinä, joten tämän datan tulkkaminen kartalle olisi ollut suuri työ ja mennyt ohi harjoitustyön tarkoituksen. Tästä syystä ominaisuus päätettiin jättää ehkä-listalle.

2.2.2 Metatietojen esittäminen

VR tarjoaa suhteellisen paljon yksinkertaista metadataa kuten ratajärjestelmään hyväksytyt operaattorit (Destia ja muut vastaavat) ja asemien karttakoordinaatit.

Tarkoituksena oli listata kaikki tämä data jossain luettavassa muodossa. Aikataulu ei kuitenkaan antanut tarpeeksi periksi, joten näistä itseään toistavista ominaisuuksista päätettiin luopua.

2.2.3 Juna- ja asemahakujen päivämäärät

Alun perin kyseisillä sivuilla piti olla myös päivämäärät joiden avulla olisi kyennyt hakemaan sivun dataa myös historiasta tai tulevasta. Dataa tutkittaessa kuitenkin huomattiin, ettei VR tarjonnut suoraa mahdollisuutta päivämäärien käyttämiseen, joten data olisi pitänyt parsia itse.

Junien kohdalla ongelmaksi muodostui se, ettei VR tarjonnut dataa kuin neljä tuntia taaksepäin ja 16 tuntia eteenpäin junan kulkemiselle. Tämän ominaisuuden saattaminen toimivaksi olisi vaatinut useita ”trial and error” -tyylisiä hakuja VR:n rajapintaa vasten, eikä tämä kehittäjiä mielestä vastannut harjoitustyön tarkoitusta.

Asemien kohdalla ongelma oli se, että junia pystyi hakemaan tietokannasta vain, jos tiesi mihin juna olisi jatkanut matkaa siltä asemalta. Tämän olisi voinut kiertää hakemalla kaikki junat ja etsimällä haettua asemaa edeltävän aseman ja suorittaen oikean haun näiden kahden tiedon avulla. Tämän olisi voinut saada toimimaan suhteellisen vaivattomasti, mutta aikataulusta loppui päivät kesken.

3 Käyttöliittymä

3.1 Bootstrap

Sovelluksessa valittiin käytettäväksi Bootstrap-tyylikirjasto, jotta ulkoasun toteuttaminen olisi mahdollisimman nopeaa. Tämän lisäksi käytettiin kuitenkin myös jonkin verran omia CSS-tyylejä.

3.2 Ulkoasuvalinnat

Ulkoasu ja värimaailma jäljittelevät jossain määrin VR:n omaa designia. Tällä haluttiin hyödyntää jo olemassa olevia assosiaatioita tiettyjen elementtien ja junien ja aikataulujen kanssa. Ainakin tekijöiden itsensä mielestä sovellus onkin helposti tunnistettavissa juniin liittyväksi.

3.3 Master Page

Päätimme käyttää tunnilla käsiteltyä Master Pagea kokeillaksemme sen toimivuutta. Master Pagen avulla voitiin luoda valmis pohja, jota muut sivut noudattelivat. Master Pagen avulla esimerkiksi vasemman laidan navigointipalkin toteuttaminen kävi helposti.

4 Ohjelman riippuvuudet

4.1 Ulkoiset kirjastot

Ohjelma tarvitsee toimiakseen peruskirjastojen lisäksi vain Newtonsoft.json-kirjaston ja se löytyy suoraan Visual Studio "Add Reference"-ominaisuuden kautta.

Kyseisen kirjaston avulla mapper-luokat pystyvät deserialisoimaan VR:ltä tulevan datan tehokkaasti suoraan domain-luokiksi.

4.2 Tiedostot ja paikallinen data

Osa VR:ltä tulevasta datasta on suhteellisen suurta. Esimerkiksi junahistorian hakemisen yhteydessä palautettava json-teksti saattaa ylittää 15 megatavun rajan. Tästä syystä järjestelmää nopeuttamaan VR:ltä tallennettiin useita datamalleja käytettäväksi paikallisesti.

Nämä datamallit löytyvät App_Data/JSON kansioista. Paikalliset tiedostot saadaan käyttöön automaattisesti kun web.configin AKVRproduction-tietueen laittaa epätodeksi. Tämän tietueen avulla Basemapper huolehtii automaattisesti siitä mistä data milloinkin haetaan joten muita muutoksia järjestelmään ei tarvita.

5 Ongelmat ja jatkokehitys

5.1 Ongelmat

Yhdeksi ongelmaksi muodostui se, ettei mvc-arkkitehtuurin hyödyntäminen kannattanut lainkaan tämän kaltaisessa asp.net-projektissa. Arkkitehtuuri oli jäykkä ja resurssien suhteen tuhlaileva. Back code -tiedostoista kasvoi suuria niiden toimiessa datan esittämisen lisäksi myös controller-tyylisesti.

Pieni haaste oli myös junadatan esittäminen sopivassa muodossa, kun palautuvan datan mukana oli sisäkkäisiä taulukkorakenteita.

5.2 Jatkokehitys

Järjestelmään alun perin suunniteltu junat kartalla -ominaisuus jäi sen aikavaativuuden takia pois. Ominaisuus on kuitenkin mahdollista toteuttaa, kunhan siihen on valmis näkemään vaivaa, joten siinä on yksi jatkokehityksen kannalta varma kohde.

Jatkokehityksen kannalta oleellisia kohteita olisivat myös päivämäärien käyttö junat- ja asemat-sivuilla.

6 Opittua

Arkkitehtuurin osalta opittiin ainakin se, ettei mvc-malli ole välttämättä parhain ratkaisu asp.net-projekteissa. Tulevaisuudessa osaa kääntää katseensa esimerkiksi mvvm-arkkitehtuuriin mikä tuntuisi taipuvan tämän kaltaisiin järjestelmiin huomattavasti paremmin.

ASP .NET –tekniikat olivat kokonaisuudessaan molemmille tekijöille uusi aluevaltaus ja näin ollen koko koodaaminen oli yhdenlaista opettelua. C# kielenä vakuutti monipuolisuudellaan, mutta web-tekniikoissa riitti kummasteltavaa.

7 Tekijät

Tekijöinä toimivat Valtteri Ahonen ja Joni Kukko.

7.1 Front end – Ahonen

Ahosen vastuualueisiin kuuluivat näkymien ja tyylien tekeminen sekä back-code-tiedostojen kirjoittaminen mitkä toimittivat myös kontrollerien virkaa mvc-arkkitehtuurissa.

Itse näkymäsivut toteutettiin web formeina eli .aspx-tiedostoina. Sivuilla käytetyt elementit ja rakenteet pysyivät yksinkertaisina osin tarkoituksenhakuisesti, sillä ulkoasu ei ollut harjoitustyön pääpainona.

Kontrollereiden varsinainen C#-osuus hoitui hyvän työnjaon vuoksi helposti. Etukäteen oli jo sovittu, mitä toiminnallisuuksia toteutetaan, ja näkymien ja kontrollereiden vaatimat taustapalvelut olivat usein jo valmiiksi tehtyinä, kun niitä alettiin kutsua. Data tuli hyvin rakenteellisessa muodossa, joten myös sen käsittely kontrolli-tasolla oli suhteellisen helppoa. Jonkin verran päänsäivä aiheuttivat sisäkkäiset listat, joiden parsiminen johti lopulta hiukan palikkamaisiin ratkaisuihin. Näitä voidaan kuitenkin myös perustella hienostuneempien tekniikoiden kankeudella tilanteessa, sekä aikataulun asettamilla rajoituksilla. Pääasia oli, että ohjelma toimi oikein, eikä tehnyt kooditasolla mitään kohtuuttoman typerää. Esimerkiksi Repeater-elementtejä käytettiin siellä, missä niitä helposti voitiin hyödyntää.

Käyttäjän antamia syötteitä joudutaan tarkistamaan jo sen vuoksi, että niitä lähetetään tietyn muotoisina eteenpäin ohjelmassa.

VR:n kanta asetti rajoituksia esimerkiksi tiettyihin hakuihin, eikä aivan kaikkia ominaisuuksia pystytty toteuttamaan. Toteutetut ominaisuudet saatiin kuitenkin toimimaan hyvin ja alkuperäisen suunnitelman mukaista arkkitehtuuria seurattiin kuuliaisesti loppuun saakka havaituista haasteista huolimatta.

Ahosen itselleen ehdottama arvosana on neljä.

7.2 Back end – Kukko

Kukon vastuisiin kuuluivat arkkitehtuurin suunnittelu ja järjestelmän taustarakenteet kuten servicet, mapperit ja domain-mallit.

Osa serviceistä pysyi pieninä koska dataa ei juuri tarvinnut käsitellä, mutta esimerkiksi juna-service kasvoi muihin nähden selkeästi. Kyseiseen palveluun pääsi suunnittelemaan ja toteuttamaan muutamia tehokkaita LINQ-kyselyitä käsiteltävää dataa kohti.

Mapperit pysyivät suhteellisen yksinkertaisena VR:n selkeän manuaalin ansioista. VR:n avoimen rajapinnan avulla saatiin useat tiedot suoraan halutussa muodossa oman järjestelmän sisään.

Basemapper ja sen clientit Kukko kirjoitti yhdenmukaistamaan datan hakemista useasta eri lähteestä. Eri lähteitä ovat VR:n avoin rajapinta, välimuistiin tallennettu istunto ja paikalliset tiedostot. Basemapperin avulla mappereiden ei tarvinnut välittää mistä data tulee vaan se tulee aina oikeassa muodossa riippumatta lähteestä.

Domain-mallit kirjoitettiin suoraan VR:ltä tulevan DTO:n (data transfer object) mukaisiksi lisäten niihin tarvittaessa lisää parametreja tiedon esittämisen helpottamiseksi.

Back end pysyi siistinä ja toimivana sekä sen toimintaperiaate on selkeä ja jatkokehitys helppoa. Selkeä miinus back endissä on sen resurssien käyttö. Kaikki haetaan erikseen, eikä back end osaa yhdistellä tietoa suoraan vaan useamman lähteen tiedot front end joutuu hakemaan erikseen. Tähän tilanteeseen päädyttiin kankean mvc-arkkitehtuurin vuoksi. Samasta syystä front end joutuu tekemään enemmän töitä mikä johti front endin monimutkaistumiseen.

Kokonaisuus on kuitenkin toimiva ja pysyttelee suhteellisen hyvin arkkitehtuurinsa sisällä. Jos jotain arkkitehtuurissa muuttaisi niin back-code-tiedostot ja controller-tyyliset ominaisuudet kannattaisi erotella toisistaan kokonaan omiin luokkiinsa. Jos projektin aloittaisi vasta nyt, niin harjoitustyöstä opittujen asioiden pohjalta mvc-arkkitehtuurin sijaan valittaisiin varmasti mvvm-arkkitehtuuri.

Näillä perusteilla Kukko ehdottaa arvosanakseen nelosta.