

Simo Pöllänen

**Mobilisovellusten hallintaohjelmiston suunnittelu työpöytäkäyttöjärjestelmille**

Opinnäytetyö  
Kajaanin ammattikorkeakoulu  
Luonnontieteiden ala  
Tietojenkäsittelyn Koulutusohjelma  
Syksy 2014



Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn Koulutusohjelma
Tekijä(t) Simo Pöllänen	
Työn nimi Mobiilisovellusten hallintaohjelmiston suunnittelu työpöytäkäyttöjärjestelmille	
Vaihtoehtoiset ammattiopinnot	Toimeksiantaja Rightware Oy
Aika Syksy 2014	Sivumäärä ja liitteet 19+0
<p>Tämä opinnäytetyö on tehty tilaustyönä Rightware Oy:lle. Työn tarkoituksena on tarjota yritykselle suunnitelma uutta ohjelmistoa varten. Työssä kuvattu ohjelmisto helpottaa yrityksen tuotteiden hallintaa. Ohjelmisto on tarkoitettu sekä yrityksen sisäiseen käyttöön että loppuasiakkaille.</p> <p>Asiakasyrityksen päätuotteita ovat kolmiulotteisten käyttöliittymien toteuttamiseen soveltuva Kanzi UI Solution ja suorituskykyä mittaavat sovellukset. Opinnäytteessä suunniteltu ohjelmisto mahdollistaa suorituskykymittauksista saadun datan helpomman visualisoinnin ja vertailun. Ohjelmistolla on tarkoitus myös hallita varsinaisia suorituskykyä mittaavia sovelluksia, jotka toimivat eri mobiilikäyttöjärjestelmillä.</p> <p>Opinnäytteessä käsitellään ohjelmiston toiminnallisuuksien aiheuttamat vaatimukset ja ohjelmistoon liittyviä teknologioita teoriatasolla ja tarjotaan kaavioin selvennetty suunnitelma ohjelmiston toteuttamista varten.</p>	
Kieli	Suomi
Asiasanat	Qt, sovelluskehitys, sovellussuunnittelu
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto



School Kajaani University of Applied Sciences	Degree Programme Business information technology
Author(s) Simo Pöllänen	
Title Designing an administrative desktop software to control mobile applications	
Optional Professional Studies	Commissioned by Rightware Oy
Date Autumn 2014	Total Number of Pages and Appendices 19+0
<p>This thesis was commissioned by Rightware Oy. The goal is to provide the company a plan for a new software. The software described in the thesis makes it easier to control the products made by the company. The software is meant for internal usage and for customers.</p> <p>The main products of the company are Kanzi UI Solution, a tool for creating 3D user interfaces, and different benchmarks. The software described in the thesis provides an easier way to visualize and compare the data received from benchmarking. The software is also intended to control the actual benchmarks that run on mobile platforms.</p> <p>The thesis deals with the requirements posed by the wanted functionality, theory related to the technologies used and provides a plan clarified with pictures to help with implementing the actual software.</p>	
Language of Thesis	Finnish
Keywords	Qt, software development, software design
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input type="checkbox"/> Library of Kajaani University of Applied Sciences

## ALKUSANAT

Kiitokset asiakkaana olevalle yritykselle oppimismahdollisuuden tarjoamisesta ja kollegoille, ohjaavalle opettajalle sekä puolisolleni rohkaisevista sanoista ja teoista.

## SISÄLLYS

1 JOHDANTO	1
2 RIGHTWARE OY	2
3 VAATIMUKSET	3
4 RATKAISU	5
4.1 Qt	5
4.2 Model-View-Controller –arkkitehtuuri	6
4.3 Vaihtoehtoisia arkkitehtuureita	8
5 SUUNNITELMA	9
5.1 Tulokset -näkyvä	9
5.2 Vertailu -näkyvä	11
5.3 Ohjelmiston käyttöliittymän komponenttikaavio	13
5.4 Ohjelmiston käyttöliittymän siirtymäkaavio	16
6 POHDINTA	18
LÄHTEET	19

## SYMBOLILUETTELO

Benchmark – Suorituskykyä mittaava sovellus/ohjelmisto

Javascript - Suurimmaksi osaksi verkko-ohjelmoinnissa käytetty ohjelmointikieli

## 1 JOHDANTO

Tässä opinnäytetyössä suunnitellaan mobiililaitteiden hallintaan tarkoitettun työpöytäohjelmiston ohjelmistoarkkitehtuuria. Opinnäytteessä on keskitytty käyttöliittymään ja siihen liittyviin tekniikoihin. Ohjelmiston tarpeet on kuvattu luvussa 3. Tarve kyseiselle ohjelmistolle ilmeni keskustellessani työnantajani kanssa tulevista ohjelmistokehitysprojekteista.

Opinnäyte on toteutettu esittelemällä projektiin liittyvät haasteet ja rajoitteet ja tarjoamalla niihin ratkaisu suunnitelman tasolla. Opinnäytettä tullaan käyttämään hyväksi ohjelmistoa toteutettaessa. Opinnäytteessä esillä olevien kaavioiden piirtämisessä on hyödynnetty ilmaista, verkossa toimivaa palvelua draw.io.

Hallintaohjelmiston ulkoasua on suunniteltu yhteistyössä yrityksen graafisen suunnittelijan kanssa. Ohjelmiston tarpeet määriteltiin yrityksen edustajien kanssa käydyissä palaverissa.

Loppukäyttäjä on ohjelmistosta riippumatta eniten tekemisissä sen käyttöliittymän kanssa. Tämän takia onkin tärkeää että käyttöliittymä on responsiivinen ja intuitiivinen. Usein ohjelmiston suunnittelu aloitetaan taustalla toimivan mekaniikan luomisesta, mutta tällöin käytettävyys voi kärsiä erilaisten arkkitehtuurissa olevien rajoitteiden takia.

## 2 RIGHTWARE OY

Tilajana toimiva yritys on Rightware Oy. Se on perustettu vuoden 2009 joulukuussa. Yrityksen pohjana toimii Futuremarkilta ostettu mobiiliyritystoimintayksikkö.

Rightwaren liiketoiminta jakautuu kahteen yksikköön:

- Käyttöliittymäkehitykseen keskittyvä Kanzi yksikkö
- Suorituskykymittaukseen keskittyvä Benchmark yksikkö

Kanzi yksikön päätuote on Kanzi UI Solution, joka mahdollistaa kolmiulotteisten käyttöliittymien helpon suunnittelemisen, kehittämisen ja kääntämisen eri alustoille. Tuote soveltuu niin mobiili-, työpöytä- kuin sulautetuille järjestelmille käyttöliittymäkehitykseen [1].

Benchmark yksikkö on kehittänyt lukuisia suorituskyvyn mittaamiseen soveltuvia sovelluksia eri alustoille, tätä nykyä kehitys keskittyy suurimmaksi osaksi mobiiliympäristöihin. Rightwaren tunnetuimmat suorituskykyä mittaavat ohjelmistot ovat:

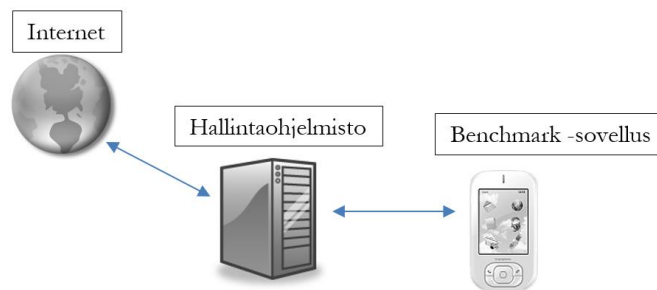
- Basemark ES 2.0
- Basemark ES 3.0
- Basemark CL
- Basemark GUI
- Basemark OS2
- Browsermark
- Basemark X

Monet ohjelmistoista toimivat useilla eri alustoilla, tehden alustojen välisen vertailun helpoksi [1].



### 3 VAATIMUKSET

Tilaaaja tarvitsee ohjelmiston tuotteidensa kehittyneempään hallitsemiseen. Tähän asti yrityksen on täytynyt jakaa käännettyt lopputuotteet manuaalisesti ja ohjeistaa asiakkaita benchmarkien asentamisessa, käyttämisessä ja suorituskykytuloksien vertailussa. Tämä on kuitenkin osoittautunut työlääksi ja huonoksi ratkaisuksi. Työkalu on suunnattu sekä yrityksen ja sen asiakkaiden käyttöön.



Kuvio 1. Hallintaohjelmiston asema testausketjussa.

Hallintaohjelmiston päätoiminnallisuudet ovat:

- Helpottaa benchmarkien jakelua.
- Helpottaa benchmarkien asentamista.
- Helpottaa benchmarkien konfigurointia ja ajamista.
- Tarjota mahdollisuus vertailla eri ajokertojen välisiä tuloksia mielekkäällä, graafisella tavalla.

Kyseiset toiminnallisuudet aiheuttavat hallintaohjelmistolle useita vaatimuksia ja rajoituksia:

- Hallintaohjelmiston täytyy olla helposti laajennettavissa.
  - Tulevaisuudessa tulee todennäköisesti uusia benchmarkteja jotka täytyy saada toimimaan hallintaohjelmiston kanssa.

- Uuden tuotteen / toiminnallisuuden lisääminen ei lähtökohtaisesti saa aiheuttaa tarvetta päivittää koko hallintaohjelmistoa.
- Hallintaohjelmiston täytyy olla helposti päivitettävissä.
  - Hallintaohjelmiston päivittäminen ei saa vaatia asiakasta kopioimaan tiedostoja manuaalisesti.
  - Hallintaohjelmiston tulee pystyä informoimaan käyttäjää saatavilla olevista päivityksistä ja luvan saatuaan pystyä päivittämään itsensä.
- Hallintaohjelmiston täytyy osata keskustella kohdelaitteiden kanssa.
  - Hallintaohjelmiston käyttäjä ei saa joutua itse kopioimaan ajettavia benchmarkteja kohdelaitteeseen tai hakemaan testitulokset käsin.
  - Tällä tavoin hallintaohjelmiston käyttäjän ei tarvitse huolehtia eikä tietää hallittavien benchmarkien toiminnasta vaan hän voi keskittyä testaamiseen.
- Hallintaohjelmiston täytyy pystyä hakemaan tarvittavat benchmarkit verkosta.
  - Helpottaa benchmarkien ajan tasalla pitämistä.
  - Vähentää benchmarkien levittämiseen vaadittavaa työtä.
- Hallintaohjelmiston täytyy pystyä ilmaisemaan testituloksia mielekkäällä, graafisella tavalla.
  - Helpottaa tuloksien ymmärtämistä.
- Hallintaohjelmiston täytyy tukea paikallista / verkossa olevaa tulostietokantaa.
- Hallintaohjelmistolla täytyy pystyä vertailemaan tuloksia.

Ohjelmisto siis helpottaa yrityksen tuotteiden käyttämistä ja nostaa jo olemassaolevien tuotteiden arvoa tarjoamalla niiden käyttäjille uusia tapoja tiedon käsittelyyn ja benchmarkien päivittämiseen ja koko testausprosessia yksinkertaistamalla.

## 4 RATKAISU

Yritys tarvitsee tehokkaan ja helposti laajennettavissa olevan työkalun. Monipuolisten lomaketoimintojensa, laajennettavuutensa ja alustariippumattomuutensa ansiosta rajapinnaksi valittiin Qt/QML.

### 4.1 Qt

Qt on useille alustoille saatavilla oleva ohjelmistorajapinta. Se mahdollistaa yhtenäisten käyttöliittymien kehittämisen useille eri alustoille samanaikaisesti vaatien useimmiten ainoastaan pieniä muutoksia ohjelmakoodiin alustojen välillä. Qt:llä luodut käyttöliittymät perivät ulkoasunsa käyttöjärjestelmältä, tästä johtuen eri alustoille käännetyt versiot näyttävät hieman erilaisilta mutta säilyttävät yhtenäisyyden käyttöjärjestelmän muiden ikkunoiden ja lomakkeiden kanssa. Qt:tä käyttävät ohjelmat eivät myöskään kärsi rajapintojen usein aiheuttamista suorituskykyongelmista, vaan niiden suorituskyky on lähes alustakohtaisesti kirjoitettujen ohjelmien tasolla [2].

Qt-ohjelmointi tapahtuu C++ -ohjelmointikielellä johon on lisätty erinäisiä toimintoja, joiden avulla tapahtumapohjainen ohjelmointi on yksinkertaisempaa. Qt toimii ohjelman ja käyttöjärjestelmän välisenä tulkkina [2].

Qt:n kolme pääkonseptia ovat:

- Käyttöliittymän täydellinen abstraktointi
  - Alussa Qt käytti omia piirtotoimintojaan eri alustojen ulkoasun emulointiin.
  - Nykyään Qt käyttää alustojen omia piirtotoimintoja käyttöliittymien esittämiseen.
- Signaalit ja vastaanottimet
  - Helpottavat olioiden välisten viestien välittämistä.

- Metaobjektikäytäjä
  - Tulkitsee C++ -koodia, generoi sen avulla ylimääräistä informaatiota joka tukee signaalien ja vastaanottimien toimintaa ja helpottaa asynkronisten funktiokutsujen käyttöä.

Qt tarjoaa sovelluskehittäjille myös Qt Quick –rajapinnan. Qt Quick on ohjelmistorajapinta, joka käyttää QML –ohjelmointikieltä javascriptin kanssa logiikan ja käyttöliittymäelementtien määrittelyyn. Qt Quickin toiminnan mahdollistaa Qt Declarative –niminen reaaliaikakääntäjä, joka käsittelee QML-määrittelyt ajon aikana ja rakentaa käyttöliittymän niiden perusteella. QML tarjoaa kehittäjälle välineet siirtymien ja monipuolisten käyttöliittymien hallintaan [3].

#### 4.2 Model-View-Controller –arkkitehtuuri

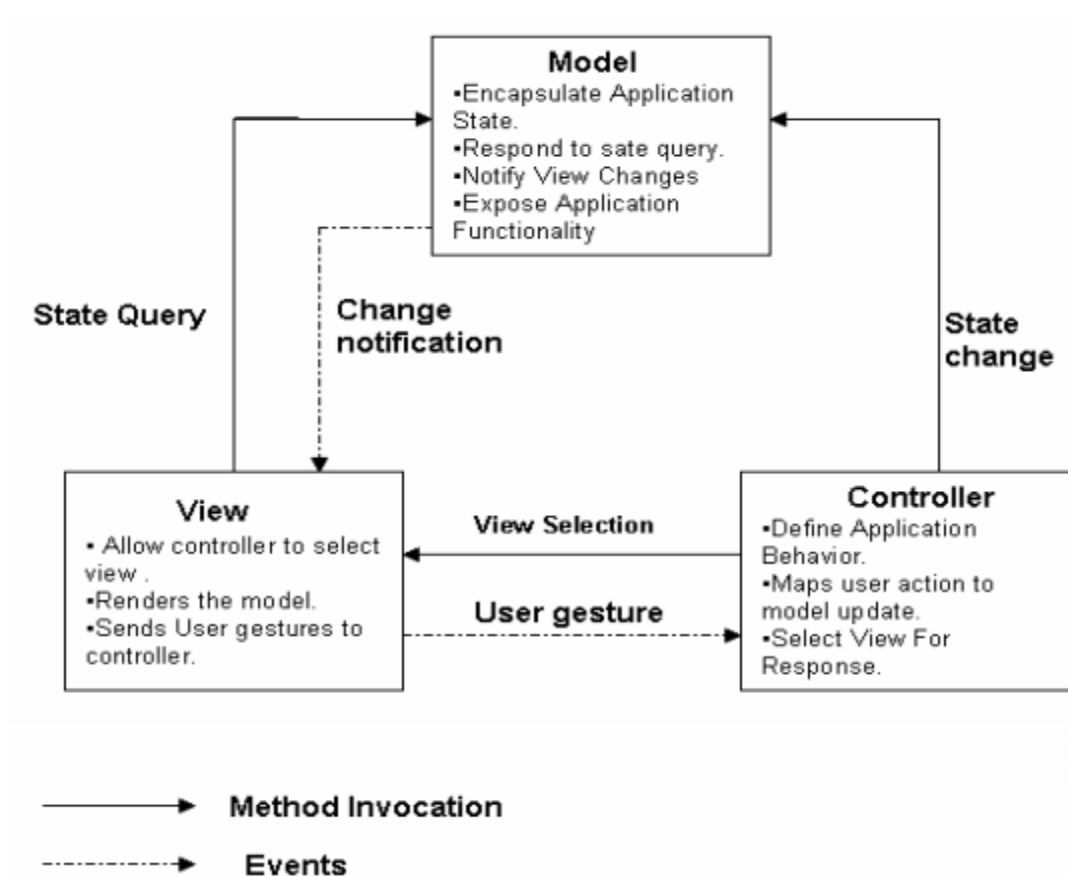
Model-View-Controller (lyhyemmin MVC) –arkkitehtuuri on eräs eniten käytetyimmistä tavoista rakentaa käyttöliittymäsovelluksia. Se oli myös ensimmäisten ratkaisumallien joukossa kun käyttöliittymiä ryhdyttiin suunnittelemaan 1970-luvulla [4].

MVC:n ytimessä on ajatus hajautetusta esittämisestä. Tässä toimintamallissa tärkeimpänä ajatuksena on käyttöliittymäelementeissä havaittavissa olevan ja alkuperäisen informaation erottaminen. Lähtökohtaisesti alkuperäistä informaatiota täytyy pystyä esittämään monin eri tavoin eri elementtien avulla. Alkuperäisen informaation täytyy olla mielekästä myös ilman ohjelmaa, joka sitä käyttää [4].

Jokainen MVC-arkkitehtuuriin pohjautuvassa ohjelmassa oleva käyttöliittymäelementti sisältää seuraavat kolme osa-aluetta:

- Model (malli)
  - Objekti, joka sisältää informaatiota. Malli ei riipu käyttöliittymästä. Malli huolehtii myös mahdollisen logiikan suorittamisesta. Opinnäytteessä kuvatussa käyttötarkoituksessa suorituskykyä mittaavista testeistä saatu tulosdata yhdessä sen käsittelylogiikan kanssa on esimerkki mallista.

- View (näköymä)
  - Objekti, joka sisältää tiedon siitä, miten vastaanotettu informaatio esitetään.
- Controller (ohjain)
  - Objekti, joka päättää kuinka elementin täytyy reagoida käyttäjän antamaan syötteeseen.



Kuvio 2. MVC-arkkitehtuurin looginen rakenne [5].

Model-View-Controller valittiin ohjelmiston lähtökohtaiseksi arkkitehtuuriksi koska sen data-malli sopii hyvin ohjelmiston tarpeisiin, se on tehokkaasti laajennettavissa ja yhdenmukaisuus Qt –rajapinnan kanssa vähentää yhteensopivuusongelmia [4]. MVC-arkkitehtuurin looginen rakenne on esitetty kuviossa 2.

### 4.3 Vaihtoehtoisia arkkitehtuureita

Valittaessa arkkitehtuuria ohjelmistolle harkittiin Model-View-Controller –arkkitehtuurin lisäksi myös muita vaihtoehtoja, mutta niissä olevat rajoitteet ja puutteet osoittivat ne epäsopiviksi käyttötarkoitukseen.

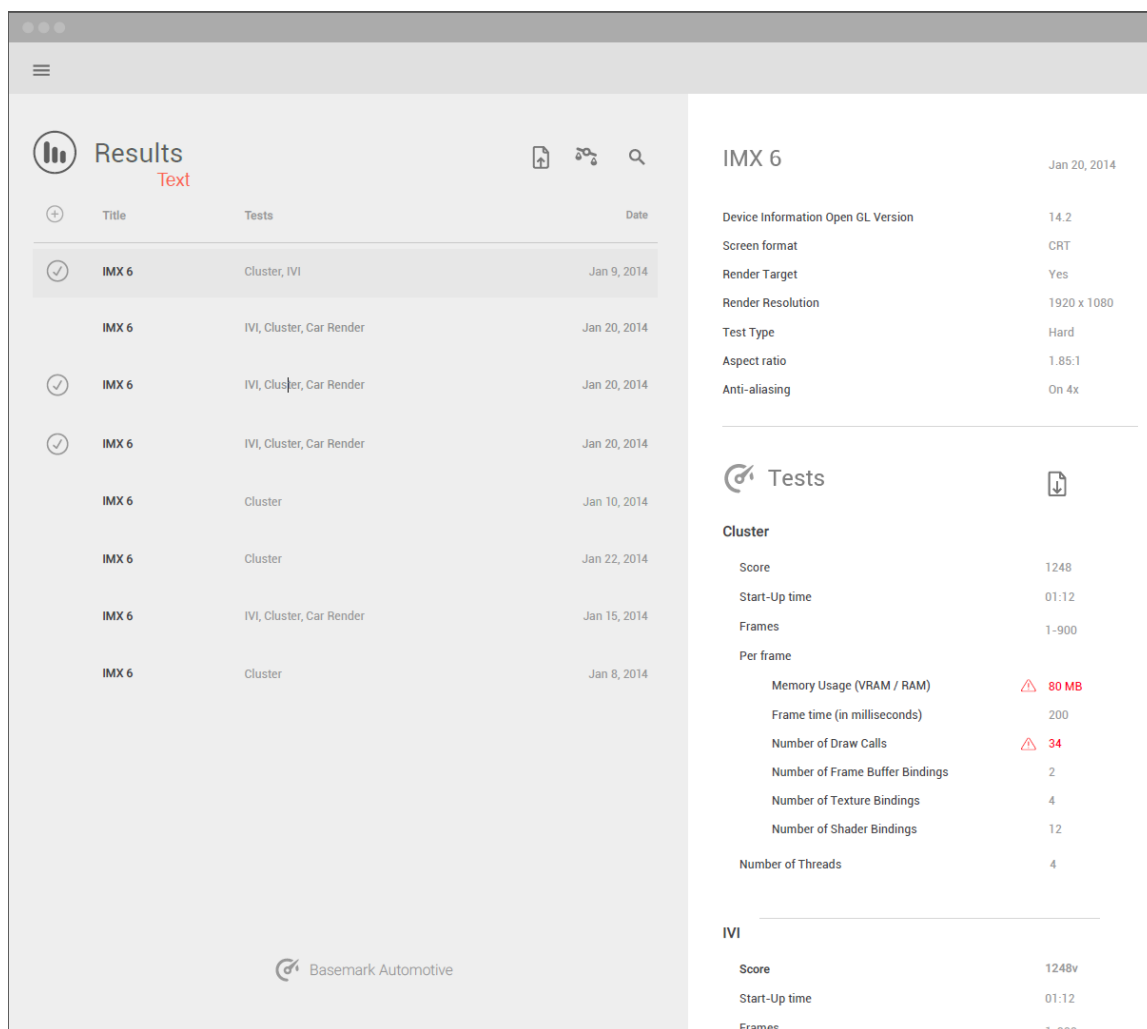
- Forms and Controls (Lomakkeet ja Ohjaimet)
  - Koostuu lomakkeista, jotka sisältävät erilaisia ohjaimia.
  - Lähtökohtaisesti ei tue eriytettyä esittämistä (tiedon esittämislogiikkaa ei ole eristetty tiedon käsittelylogiikasta).
- Model-View-Presenter (Malli-Näkymä-Esitin)
  - Koostuu widgeteistä (pienoisohjelmista).
  - Esitintä ja näkymää ei välttämättä eroteta tarkasti.

Esitettävä tieto on käyttötapauksessa tietyllä tavalla järjesteltyä ja luonteeltaan samanlaista. Tästä johtuen on viisainta käyttää rakennetta, jossa kerran tehtyjä komponentteja voi käyttää uudelleen eri tavoin [4].

## 5 SUUNNITELMA

Ohjelmiston suunnittelu aloitettiin käyttöliittymätasolta. Tällä tavoin varmistetaan että ohjelmisto on mahdollisimman käyttäjäystävällinen. Käyttöliittymää suunniteltaessa huomattiin myös mahdollisia ongelmia ohjelmiston toiminnoissa. Tästä johtuen ongelmille mietittiin mahdollisia ratkaisumalleja ja suunnitelmia muokattiin ottamaan nämä huomioon.

### 5.1 Tulokset -näkyvä



Kuvio 3. Tulokset -näkyvä [6].

Kuviossa 3 esitetyn näkymän vasemmassa lohossa esitetään tulokset yleisellä tasolla. Valittaessa yksi tulos tulosjoukosta oikealla oleviin lohkoihin tulee esille tarkempaa testiajokohtaista informaatiota.

Oikealla ylhäällä olevassa lohossa esitetään laitekohtaista informaatiota ja yleisellä tasolla tietoa ajettujen testien asetuksista.

Oikealla keskellä ja alhaalla olevissa lohkoissa esitetään testiajoon sisältyvien alitestien informaatiota.

The screenshot displays the Basemark Automotive software interface. On the left, a sidebar menu is open, showing options: RESULTS (selected), CONFIGURATIONS, RUN TEST, and TO MAIN MENU. The main area is divided into two columns. The left column shows a list of test results with columns for 'ts', 'Date', and 'Cluster, Car Render'. The right column displays detailed information for a specific test, 'IMX 6', dated Jan 20, 2014. This includes device information (Open GL Version 14.2, Screen format CRT, Render Target Yes, Render Resolution 1920 x 1080, Test Type Hard, Aspect ratio 1.85:1, Anti-aliasing On 4x). Below this, there is a 'Tests' section for 'Cluster' with metrics: Score (1248), Start-Up time (01:12), Frames (1-900), and Per frame details (Memory Usage (VRAM / RAM) 80 MB, Frame time (in milliseconds) 200, Number of Draw Calls 34, Number of Frame Buffer Bindings 2, Number of Texture Bindings 4, Number of Shader Bindings 12, Number of Threads 4). At the bottom, there is an 'IVI' section with metrics: Score (1248v), Start-Up time (01:12), and Frames (1-900).

Kuvio 4. Tulokset –näkyä sivupalkki avattuna [6].

Ikkunan vasemmassa reunassa oleva sivupalkki on avattu kuviossa 4. Sivupalkki tarjoaa käyttäjälle mahdollisuuden vaihtaa avointa sivua ohjelmistossa. Sivupalkki on olemassa ohjelmiston jokaisella sivulla.

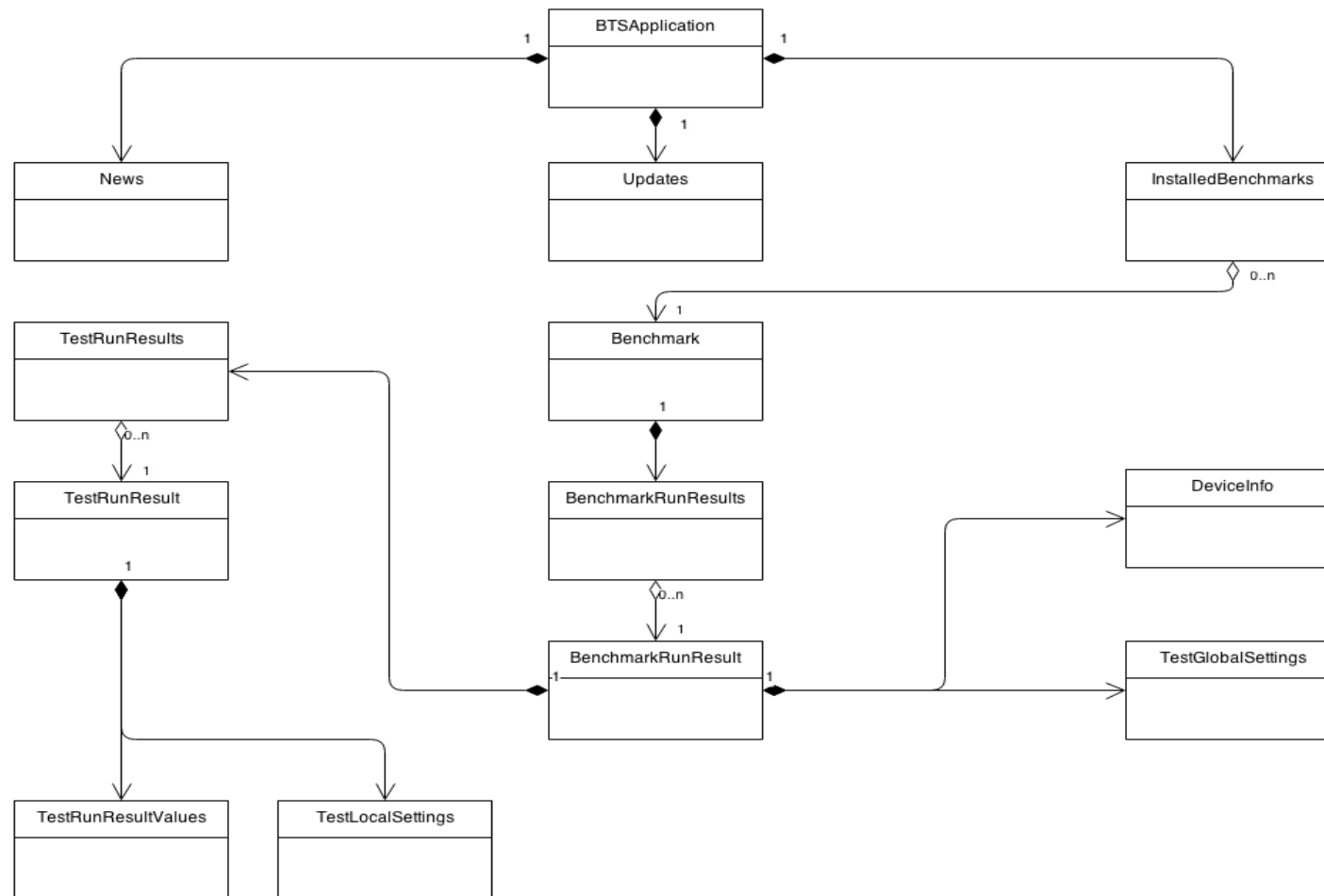


## 5.2 Vertailu -näkyvä

Areas	#98 IMX 6	#78 IMX 6	#75 IMX 6	#56 IMX 6
Overall Score	2536	2536	2536	2536
Star-Up Time	1230 ms	1230 ms	1230 ms	1230 ms
Memory Usage	58 MB	58 MB	58 MB	58 MB
Draw Calls	4	4	4	4

Kuvio 5. Vertailunäkymä [6].

Vertailunäkymä on esitetty kuviossa 5. Näkymässä käyttäjän on helppo vertailla tuloksia eri testiajojen välillä. Vasemmalla olevassa lohkossa esitetään vertailtavaksi valittu informaatio testikohtaisesti. Oikealla olevassa lohkossa käyttäjä pystyy valitsemaan vertailtavan datan, testin josta data haetaan ja testiajot joiden välillä vertailu suoritetaan.



Kuvio 6. Luokkatason suunnitelma.

### 5.3 Ohjelmiston käyttöliittymän komponenttikaavio

Kuviossa 6 on esitelty yleisellä tasolla ohjelmiston eri komponentit.

Ohjelmiston ytimessä on BTSApplication, joka toimii ohjelmiston kiinnekohtana käyttöjärjestelmään. BTSApplication ei näy loppukäyttäjälle, vaan se huolehtii ohjelmiston päivittämisestä Updates -moduulin kautta, yritystä koskevien uutisten näyttämisestä News -moduulin kautta ja benchmarkien sovitinosien tunnistamisesta InstalledBenchmarks -moduulin välityksellä.

Updates -moduuli huolehtii ohjelmiston päivittämisestä. Sen tehtävänä on myös keskustella yrityksen palvelimien kanssa ja ladata automaattisesti käyttäjälle saatavilla olevat benchmark-sovitinosat.

InstalledBenchmarks -moduuli tunnistaa lisäosina saatavilla olevat benchmark-sovitinosat. Eri benchmark-sovitinosat mahdollistavat yhteensopivuuden eri tuotteiden kanssa, näin koko sovellusta ei tarvitse päivittää toiminnallisuuden lisäämiseksi.

Benchmark-sovitinosa sisältää informaation benchmarkin nimestä, yksilöllisestä tunnuksesta, tuloksien dataformaattista ja olion joka sisältää testiajojen tulokset. Benchmark-sovitinosan avulla voidaan hakea testattavalta laitteelta (esimerkiksi älypuhelin tai tablettitietokone) testien tulokset, asentaa uusia tuotteita ja konfiguroida laitteelle jo asennettuja tuotteita yhteistyössä suunnitteilla olevan laiteläheisen komponentin kanssa.

BenchmarkRunResults on moduuli joka sisältää benchmarkkohtaisesti eri testiajokerroilla saadun informaation ja antaa sitä muille moduuleille tarpeen vaatiessa.

BenchmarkRunResult on moduuli joka sisältää yhden testiajokerran informaation ja palauttaa sitä BenchmarkRunResults -moduulille sen sitä tarvittaessa. Tämä moduuli sisältää myös kolme tukimoduulia:

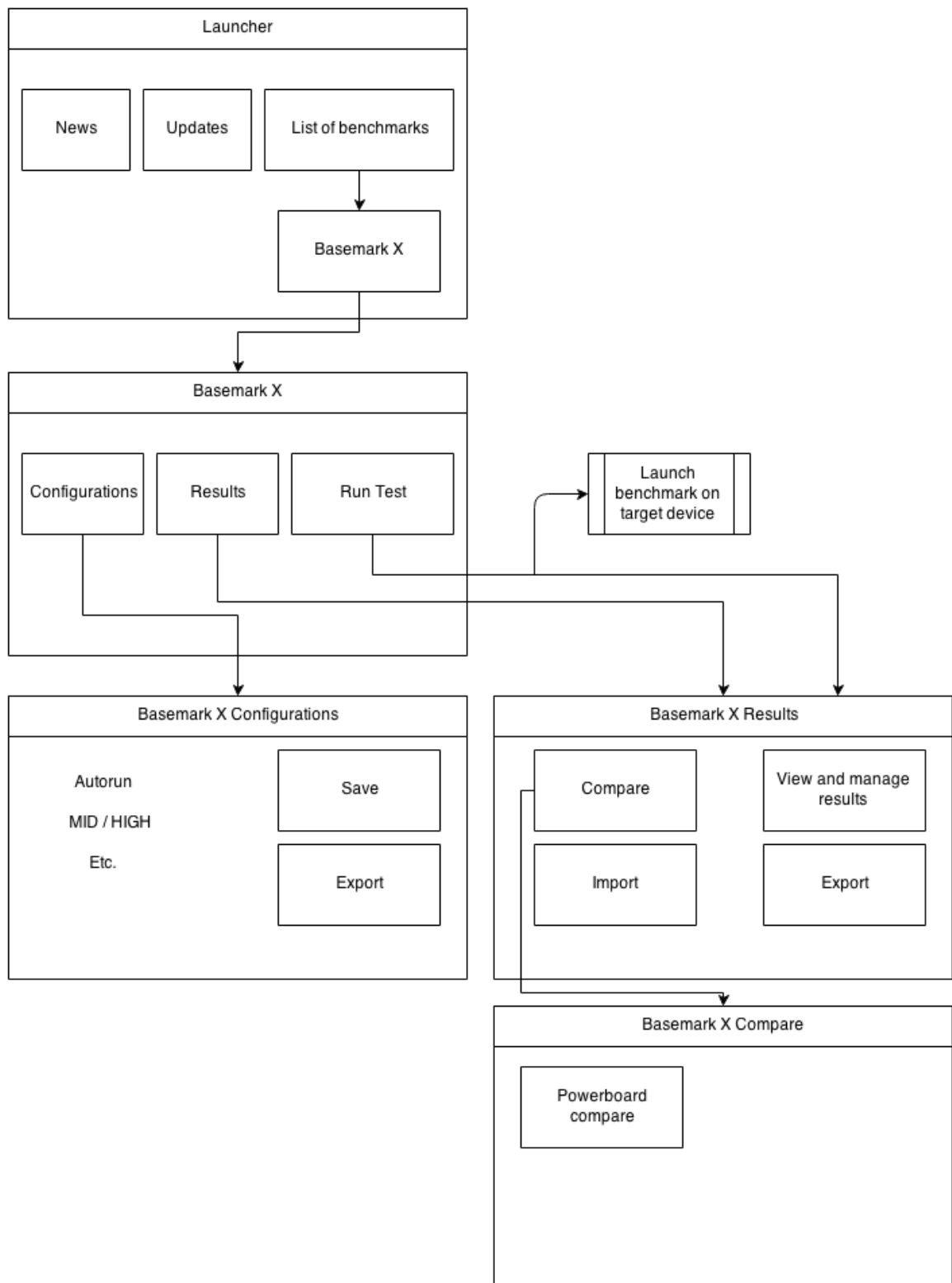
- TestGlobalSettings
  - Sisältää testiajokerran yleiset asetukset

- DeviceInfo
  - Sisältää laitekohtaista informaatiota
- TestRunResults

TestRunResults –moduuli sisältää yhden testiajokerran alitestit. Se myös tarvittaessa palauttaa sitä BenchmarkRunResult –moduulille.

TestRunResult –moduuli sisältää yhden testiajokerran yhden alitestin informaation. Se käyttää kahta tukimoduulia informaation hallintaan:

- TestRunResultValues
  - Sisältää yhden alitestin tulokset.
- TestRunLocalSettings
  - Sisältää yhden alitestin testikohtaiset asetukset.



Kuvio 6. Siirtymäkaavio.

#### 5.4 Ohjelmiston käyttöliittymän siirtymäkaavio

Kuviossa 6 on kuvattu hallintaohjelmiston looginen toiminta. Isot nimetyt lohkot ovat ohjelmiston käyttäjälle näyttämiä sivuja, jotka sisältävät painikkeita. Painikkeita on kaaviossa esitetty isojen lohkojen sisällä olevina pienempinä, nimettyinä lohkoina. Nuolet kaaviossa osoittavat siirtymiä eri sivujen välillä.

Ohjelmiston käynnistyessä käyttäjä näkee ensimmäisenä Launcher –sivun. Sieltä käyttäjä voi halutessaan siirtyä Updates –sivulle, News –sivulle tai valita käsiteltävän benchmarkin. Kaaviossa esitellyssä tilanteessa on valittu tarkasteluun Basemark X.

Käyttäjän valittua haluttu benchmark listalta, ohjelmisto siirtyy benchmarkkohtaiselle sivulle. Kyseiseltä sivulta käyttäjä voi valita halutun toiminnon seuraavista:

- Configuration (asetukset)
- Run Test (aja testi)
- Results (tulokset)

Configuration –sivulla käyttäjä voi määritellä benchmarkin ajoa koskevia asetuksia. Käyttäjällä on mahdollisuus tallentaa asetukset tiedostoon, jonka jälkeen hän voi käsin siirtää asetukset kohdelaitteelle tai vaihtoehtoisesti siirtää asetukset suoraan kohdelaitteelle. Tällöin ohjelmisto huolehtii tiedostojen kopioinnista oikeaan paikkaan.

Käyttäjän valitessa benchmarkkohtaiselta sivulta Run Test –painikkeen, käynnistää ohjelmisto suorituskykyä mittaavan ohjelmiston automaattisesti kohdelaitteella jos se on ohjelmissa käytävään tietokoneeseen sillä hetkellä yhdistettynä.

Results –sivulla käyttäjä voi tarkastella ja hallita benchmarkista saatuja tuloksia. Results –sivun kautta käyttäjällä on myös mahdollisuus tuoda ohjelmistoon kohdelaitteella saatuja tuloksia ja viedä tuloskokonaisuuksia erilliseen tietokantaan. Results –sivun kautta käyttäjä voi halutessaan päästä vertailusivulle.

Compare (vertailu) –sivulla käyttäjä voi vertailla eri ajokertojen välisiä tuloksia. Ohjelmistoon tullaan toteuttamaan erilaisia graafisia elementtejä, joiden avulla tulosten visuaalinen vertailu on helpompaa.

News (uutiset) –sivulla näytetään yritykseen ja sen tuotteisiin liittyviä uutisia.

Updates (päivitykset) –sivulla käyttäjä voi tarkistaa ohjelmiston päivitykset manuaalisesti ja tarvittaessa asentaa ne.

## 6 POHDINTA

Ohjelmiston suunnitteleminen käyttöliittymä edellä mahdollistaa käyttäjäkokemuksen huomioon ottamisen koko kehitysprosessin aikana. Aloitettaessa ohjelmiston suunnittelu käyttöliittymästä helpotetaan myös ohjelmiston muiden osa-alueiden suunnittelua, koska käyttöliittymää suunniteltaessa nousee esiin mahdollisia ongelmia ohjelman käytössä.

Teoriapohjaa opinnäytteelle kerättiin Qt –projektin kotisivuilta [7] ja aiheeseen liittyviltä verkkosivuilta.

Vaatimuksien listaaminen ennen varsinaisen suunnittelun aloittamista on hyödyllistä. Ohjelmisto koostuukin osasista, jotka yhdessä pystyvät tarjoamaan halutut toiminnallisuudet. Laajennettavuus on ollut eräänä tärkeimmistä suunnittelua ohjaavista elementeistä. Tästä on hyöttyä ohjelmistoa jatkokehittäessä.

Qt valittiin käytettäväksi ohjelmointirajapinnaksi helppokäyttöisyytensä ja MVC –arkkitehtuuriin pohjautuvan rakenteensa vuoksi. Qt on myös alustariippumaton, joten ohjelmisto on helppo tarvittaessa kääntää useille käyttöjärjestelmille. MVC –arkkitehtuuri mahdollistaa komponenttien luovan uudelleenikäytön mikä vuorostaan nopeuttaa kehitysprosessia.

Suunnitteluvaiheen aikana toteutettiin erinäisiä testejä Qt:n ja Qt Quickin toimintojen kokeilemiseksi. Testeistä oli hyötyä, sillä niiden avulla huomattiin mahdollinen ongelma alkuperäisessä suunnitelmassa.

Kokonaisuutena opinnäyte oli mielenkiintoinen oppimiskokemus. Ohjelmistoa suunniteltaessa nousi esille mahdollisia ongelmia ohjelman toiminnassa ja ne osattiin täten välttää.



## LÄHTEET

- (1) Rightware Available at <http://www.rightware.com/>. Accessed 10/30/2014
- (2) QtWhitepaper Available at <http://qt-project.org/wiki/QtWhitepaper>. Accessed 10/28/2014.
- (3) Introduction to Qt Quick Available at [http://qt-project.org/wiki/Introduction\\_to\\_Qt\\_Quick](http://qt-project.org/wiki/Introduction_to_Qt_Quick). Accessed 10/28/2014.
- (4) GUI Architectures Available at <http://martinfowler.com/eaDev/uiArchs.html>. Accessed 10/15/2014.
- (5) Study of J2EE Patterns Available at [http://www.it.iitb.ac.in/~kaushal/downloads/oos\\_project\\_report.pdf](http://www.it.iitb.ac.in/~kaushal/downloads/oos_project_report.pdf). Accessed 10/16/2014.
- (6) Aaltonen, Mikko (mikko.aaltonen@rightware.com) 2014
- (7) Qt Project Available at <http://qt-project.org/>. Accessed 10/28/2014.