

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

SAPUI5 skatu veikspējas analīzes rīks

Kvalifikācijas darbs

Autors: Romāns Kolduns

Studenta apliecības numurs: rk13025

Vadītājs: dipl.mat., vecākais projektu pārvaldnieks, Imants Pops

Rīga, 2015

Anotācija

Kvalifikācijas darba mērķis ir izveidot rīku, kurš spēs noteikt, cik lielā mērā XML skatu izmantošana SAPUI5 rīkkopā ietekmē šo skatu izpildes laiku interneta pārlūkā, salīdzinājumā ar Javascript skatu paradigmas pielietošanu ekvivalentu problēmu risināšanai.

Šis dokuments satur izstrādājamā produkta tehnisko dokumentāciju, kas sevī ietver programmatūras prasību specifikāciju, programmatūras projektējuma aprakstu un testēšanas dokumentāciju.

Atslēgas vārdi:

SAPUI5, veiktspēja, Javascript, XML, C#, .NET, MVC

Abstract

The purpose of this qualification work is to develop a tool to determine, how much use of XML views in SAPUI5 toolkit would affect their execution time in a web browser, compared to Javascript views designed for solving equivalent tasks.

This document contains technical documentation of the product being developed, which consists of software requirements specification, software design and testing documentation.

Keywords:

SAPUI5, performance, Javascript, XML, C#, .NET, MVC

Satura rādītājs

Vārdnīca.....	6
Ievads.....	7
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	8
1.1. Ievads.....	8
1.1.1. Dokumenta nolūks.....	8
1.1.2. Darbības sfēra	8
1.2. Vispārējs apraksts	8
1.2.1. Situācijas apraksts	8
1.2.2. Vispārēji ierobežojumi	9
1.3. Funkcionālās prasības.....	10
1.4. Lietotāja saskarne	20
1.5. Nefunkcionālās prasības	23
2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	24
2.1. Ievads.....	24
2.1.1. Dokumenta nolūks.....	24
2.2. Tehniskie risinājumi	24
2.3. Dekompozīcijas apraksts	24
2.3.1. Moduļu apraksts	25
2.3.2. Prasību trasējamības matrica	28
2.4. Klašu diagramma.....	30
2.5. Algoritmu apraksts	31
2.5.1. SAPUI5 XML skata konvertēšana par tam ekvivalentu Javascript skatu	31
2.5.2. SAPUI5 skatu izpildes laika mērīšana.....	35
3. TESTĒŠANAS DOKUMENTĀCIJA	37
3.1. Testēšanas plāns	37
3.2. Testēšanas žurnāls	38

4. PROJEKTA ORGANIZĀCIJA	40
5. KONFIGURĀCIJU PĀRVALDĪBA	41
6. KVALITĀTES NODROŠINĀŠANA	42
7. DARBIETILPĪBAS NOVĒRTĒJUMS	43
7.1. Sākotnējais darbietilpības novērtējums	43
7.2. Darbietilpības novērtējums pēc produkta nodošanas ekspluatācijā	43
8. LIETOTĀJA CEĻVEDIS.....	44
8.1. Instalēšanas un palaišanas pamācība	44
8.2. SAPUI5 skatu konversija	45
NOBEIGUMS	46
IZMANTOTĀ LITERATŪRA	47
PROGRAMMATŪRAS PIRMKODS	48

Vārdnīca

Agregācijas – visi viena objekta apakšobjekti.

Asociācijas – visi viena objekta saistītie objekti.

C# – *Microsoft* izstrādāta programmēšanas valoda.

DRY (Don't repeat yourself) – programmatūras izstrādes princips, kas vērsts uz to, lai programmas pirmkodā neatkārtotos vieni un tie paši koda fragmenti.

Javascript – klienta puses skriptu valoda, ko izmanto tīmekļa mājaslapās.

JIRA – veicamo darbu uzskaites un kļūdu atsekošanas sistēma [5].

JSHint – rīks, kurš paredzēts Javascript koda kļūdu un potenciālo problēmu atklāšanai un labošanai. [6]

MVC (Model-View-controller) – projektēšanas veids, kas tiek lietots, lai sadalītu lietojumprogrammu trīs pastāvīgās komponentēs - modelī, skatos un kontrolleros.

Parsēšana – ievaddatu analīze atbilstoši iepriekš zināmiem nosacījumiem.

PPA – programmatūras projektējuma apraksts.

PPS – programmatūras prasību specifikācija.

SAPUI5 – rīkkopa tīmekļa mājaslapu saskarnes izstrādei saskaņā ar MVC pieeju.

SAPUI5 skats (arī “derīgs” SAPUI5 skats) – noteiktam formātam atbilstoša teksta datne (.html, .json, .js, .xml), kas atbilst SAPUI5 skatu specifikācijai.

SVN (Subversion) – atklāta pirmkoda versiju pārvaldības sistēma.

XML (*eXtensible Markup Language*) ir plaši izmantota iezīmēšanas valoda.

XPath – vaicājumu valoda, kas paredzēta, lai noteiktu ceļu līdz XML mezglam.

Ievads

Pastāv 4 SAPUI5 skatu paradigmas: Javascript, XML, JSON un HTML. Katrai no tām ir savas priekšrocības un trūkumi, tomēr šāda dažādība veido arī atšķirības skatu veikspējā. Tāpēc ir vajadzīgs kāds rīks, kurš spētu analizēt skatus un noteikt, kura paradigma izrādīsies piemērotāka konkrēta uzdevuma veikšanai. Ir jānoskaidro, kura skata izpilde aizņems vismazāk laika un patērēs vismazāk datora resursu.

Šajā darbā no 4 SAPUI5 skatu paradigmām tiek izmantotas tikai divas – XML un Javascript. Tas skaidrojams ar to, ka programmaprodukta pasūtītājs ikdienā izmanto tikai šīs skatu paradigmas.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Dokumenta nolūks

Programmatūras prasību specifikācijā formulētas funkcionālās un nefunkcionālās prasības SAPUI5 skatu veiktspējas analīzes rīkam.

Dokuments paredzēts programmatūras izstrādātājam un produkta pasūtītājam. Balstoties uz to, tiks realizēta produkta funkcionalitāte.

1.1.2. Darbības sfēra

Produkts paredzēts SAPUI5 rīkkopas XML skatu tūlītējai konvertēšanai par identiskiem Javascript skatiem un to veiktspējas analīzei.

1.2. Vispārējs apraksts

1.2.1. Situācijas apraksts

Pašlaik SAPUI5 lietotņu izstrādātājiem nākas izšķirties – rakstīt statiskas saskarnes objektu deklarācijas, izmantojot XML skatus, vai izmantot dinamisku pieeju, rakstot Javascript skatus.

Javascript skatos ir atļauts definēt funkcijas skata kodā, kamēr XML skatos tas nav pieļaujams. XML skatu paradigma paredz klasiskās MVC pieejas pielietošanu, kur skats ir pilnībā atdalīts no kontrollera, bet Javascript skatos ir iespējams iekļaut daļu no kontrollera koda. Tāpat, atšķirībā no Javascript skatiem, XML skatos izmantotās statiskās deklarācijas ir nepieciešams atbilstoši interpretēt, pirms notiek tālākā apstrāde. Šis process savukārt paildzina XML skatu izpildes laiku.

Diemžēl šobrīd nav pieejams neviens rīks, kas spētu gatavu XML skatu konvertēt uz ekvivalentai objektu struktūrai atbilstošu SAPUI5 Javascript skatu un salīdzināt to veiktspēju, tāpēc vajadzīgs rīks, kas to spēs paveikt.

1.2.2. *Vispārēji ierobežojumi*

Lai arī produkts spēj darboties autonomi un pārveidot skatus, neizmantojot ārējās bibliotēkas, servisu vai programmatūru, jebkurš SAPUI5 formātam atbilstošs skats var tikt interpretēts tikai interneta pārlūkā. Lietotāja pārlūkam jāatbalsta HTML5 un Javascript un tam obligāti jābūt iespējamam pat izpildot XML skatus. Pārbaudīts, ka SAPUI5 skati darbojas šādās pārlūkprogrammās:

- Microsoft Internet Explorer 9+,
- Google Chrome 40+,
- Mozilla Firefox 36+

Produkts darbojas tikai Microsoft Windows operētājsistēmā, sākot ar Windows Vista.

Tabulā 1.2.2.1 nosauktas pamatprasības, kuras nepieciešamas produkta darbināšanai:

1.2.2.1. tabula

	Minimālās prasības	Vēlamās prasības
Operētājsistēma	Windows Vista vai jaunāks, .NET Framework 4.0	Windows 7 vai jaunāks, .NET Framework 4.0
RAM (brīvā vieta)	16 MB	32 MB
Brīvā vieta uz datu nesēja	22 MB	22 MB + vieta SAPUI5 skatiem
Ekrāna izšķirtspēja	1280x1024 px	1280x1024 px
Interneta savienojums	Obligāts	
Interneta pārlūks	Pārlūks ar Javascript un HTML5 atbalstu	

Datorā jābūt instalētam .NET Framework 4.0 vai jaunākai versijai. Ja tāda nav, tas tiks pieinstalēts automātiski.

Interneta savienojums ir obligāts.

1.3. Funkcionālās prasības

1.3.1. Datņu atvēršana un to satura apskate

Identifikators	Open_File
Nosaukums	Datņu atvēršana un to satura apskate
Ievads	<p>Funkcija nodrošina iespēju atvērt XML formāta datnes un apskatīt to saturu teksta laukā. Datņu atvēršana iespējama gan, galvenajā izvēlnē izvēloties File --> Open, bet arī, izmantojot <i>Drag and drop</i> mehānismu.</p> <p>Šī funkcija ir paredzēta tikai, lai nolasītu lietotāja izvēlētās datnes saturu un parādītu to uz ekrāna. Funkcija netiek izsaukta gadījumos, kad citu funkciju izpildes laikā jānolasa kādas datnes saturs.</p>
Ievaddati	.txt, .xsd vai .xml formāta datnes.
Apstrāde	<p>Datne tiek atvērta un tās saturs tiek parādīts, neveicot nekādu datu apstrādi.</p> <p>Uz ekrāna teksta laukā tiek attēlots izvēlētās datnes tekstuālais saturs.</p>
Izvaddati	nav
Kļūdu u.c. paziņojumi	1) Nepareizs datnes formāts. Izvēlieties XML, XSD vai .txt datnes.

1.3.2. SAPUI5 skata validācija

Identifikators	Validate_View
Nosaukums	SAPUI5 skata validācija
Ievads	<p>Pirms sākt darbības ar kārtējo skatu, tā saturs vispirms tiek pārbaudīts atbilstībai XML formāta sintaksei un tam jābūt derīgam SAPUI5 skatam. Tikai tad, ja datnes saturs atbilst XML formātam, tad tālāk tiek pārbaudīts, vai tas atbilst SAPUI5 XML skatu formātam.</p> <p>Funkcija pati nodrošina norādītās datnes satura nolasīšanu.</p>
Ievaddati	Validējamās XML datnes pilns ceļš un nosaukums

Apstrāde
1) Tiek pārbaudīts, vai dotās datnes saturs atbilst XML formātam. 2) Tiek pārbaudīts, vai datnes saturs ir derīgs SAPUI5 skats.
Izvaddati
1) Dotais kods ir / nav SAPUI5 skats 2) Kļūdas paziņojuma teksts, ja ievaddati nav SAPUI5 skats
Kļūdu u.c. paziņojumi
1) Nepareizs datnes formāts. 2) Dotā datne nav derīgs SAPUI5 skats.

1.3.3. XML parsēšana

Identifikators	Parse_XML_File
Nosaukums	XML parsēšana
Ievads	
<p>Parsēšana nepieciešama ievaddatu analizēšanai un to tālākai sagrupēšanai pa daļām – atverošie un aizverošie tagi, atribūti, vērtības, priedēkļi (<i>prefix</i>), vārdkopas (<i>namespace</i>) u.c. No iegūtajiem fragmentiem var izveidot SAPUI5 Javascript skatu, saliekot šos fragmentus vajadzīgajā kārtībā.</p>	
Ievaddati	
XML datne .xml formātā.	
Apstrāde	
<p>Vispirms tiek izsaukta funkcija <code>Validate_View</code>, kas pārbauda, vai dotā datne satur derīgu SAPUI5 XML skatu. Ja pārbaude nav izieta, tiek parādīts kļūdas paziņojums.</p> <p>XML koda saturs tiek sadalīts šādās daļās:</p> <ul style="list-style-type: none"> • Mezgla nosaukums, • Mezgla tips, • Priedēklis (<i>prefix</i>), • Vārdkopas URI saite (<i>namespace URI</i>), • Vecāka mezgls, • Mezgla dziļums (<i>depth</i>), • Mezgla pozīcija vecāka mezgla bērnu sarakstā, • Bēni, • Ceļš (<i>XPath</i>), • Atribūti <ul style="list-style-type: none"> ○ Atribūta nosaukums ○ Atribūta vērtība 	
Izvaddati	

XML mezgli
Kļūdu u.c. paziņojumi
1) Dotā datne nesatur derīgu XML skatu.

1.3.4. XML struktūras koka attēlošana

Identifikators	Draw_XML_Tree
Nosaukums	XML struktūras koka attēlošana
Ievads	
	<p>Funkcija nepieciešama, lai uzskatāmi varētu redzēt XML koda struktūru un objektu hierarhiju. Šī funkcija nav paredzēta Javascript koda izveidei, tā vajadzīga tikai lietotājam, lai vizualizētu XML kodu.</p> <p>Jābūt iespējai pielāgot struktūras izskatu:</p> <ul style="list-style-type: none"> • Var rādīt XML mezgla saturu kā parastu tekstu (<i>outer XML</i>), • Objektu nosaukumus var aizstāt ar ID atribūtu vērtībām, piemēram, „Button” aizvietot ar „Submit”, • Var izvērst / sakļaut visu koka struktūru vai jebkuru atsevišķu mezglu.
Ievaddati	
	XML datne .xml formātā
Apstrāde	
	<p>Tiek izsaukta funkcija Validate_View. Ja validācija nav izdevusies, tad funkcija Validate_View parāda atbilstošu kļūdas paziņojumu. Ja validācija ir veiksmīgi izieta, tad tiek izveidots saknes mezgls, kuram tiek pievienoti bērni. Bērniem savukārt tiek pievienoti to bērni un tā process turpinās, kamēr mezgliem vairs nav bērnu.</p> <p>Ja mezglam ir bērni, tad uz ekrāna tiek rādīts tā nosaukums ar priedēkli, bet ja bērnu nav, tad var tikt rādīts tā nosaukums vai ID atribūta vērtība, ja eksistē tāds atribūts.</p>
Izvaddati	
	Uz ekrāna koka izskatā tiek attēlota XML datnes satura struktūra.
Kļūdu u.c. paziņojumi	
nav	

1.3.5. XML un Javascript koda saspiešana

Identifikators	Compress_Code
Nosaukums	XML un Javascript koda saspiešana
Ievads	
Funkcija dzēš rindu pārtraukumus, atkāpes un liekās tukšumzīmes ar mērķi samazināt skata koda simbolu skaitu un attiecīgi arī tā izmēru. Šāds apstrādāts kods aizņem mazāk vietas un to iespējams ātrāk pārsūtīt uz klienta datoru. Paredzams, ka funkcija tiks izmantota tikai XML un Javascript koda saspiešanai. Funkcijas izpildei nav nepieciešama ievaddatu validācija vai cita veida pārbaudes.	
Ievaddati	
XML vai Javascript kods	
Apstrāde	
1) Dzēš „tab”simbolus 2) Dzēš rindu pārtraukumus Funkcijas atgrieztais kods tiek parādīts uz ekrāna.	
Izvaddati	
Javascript vai XML kods saspiestā veidā.	
Kļūdu u.c. paziņojumi	
nav	

1.3.6. Javascript skata izveidošana no XML skata

Identifikators	Create_Javascript_View
Nosaukums	Javascript skata izveidošana no XML skata
Ievads	
Funkcija nodrošina sintaktiski pareiza un darbotiespējīga Javascript skata izveidi, kuram turklāt jādarbojas precīzi tāpat, kā attiecīgais XML skats.	
Ievaddati	
XML datne .xml formātā.	
Apstrāde	
Tiek izsaukta funkcija Parse_XML_File, kura savukārt izsauc funkciju Validate_View. Kļūdu gadījumā tiek izvadīts kļūdas paziņojums. Javascript kods tiek ģenerēts atbilstoši algoritma aprakstam 2.5.1. nodaļā.	
Izvaddati	
Javascript kods, kuru var saglabāt un izpildīt.	

Kļūdu u.c. paziņojumi
nav

1.3.7. Javascript koda lasāmības uzlabošana

Identifikators	Beautify_Javascript
Nosaukums	Javascript koda lasāmības uzlabošana
Ievads	
Funkcija sadala Javascript kodu pa rindiņām, saliek sākuma atkāpes un veic citas izmaiņas, lai padarītu kodu lasāmu un pārskatāmu.	
Ievaddati	
Javascript kods bez sākuma atkāpēm un rindu pārnesumiem.	
Apstrāde	
Kods tiek noformēts atbilstoši JSHint [6] ieteikumiem.	
Izvaddati	
Javascript kods ar sākuma atkāpēm un rindu pārnesumiem.	
Kļūdu u.c. paziņojumi	
nav	

1.3.8. Darbības izpildes laika noteikšana

Identifikators	Measure_Time
Nosaukums	Darbības izpildes laika noteikšana
Ievads	
Funkcija izmēra, cik ilgi izpildījās pēdējā veiktā darbība. Laiks tiek mērīts mikrosekundēs.	
Ievaddati	
Nav.	
Apstrāde	
Tiek mērīts, cik daudz laika pagājis no darbības izpildes sākuma līdz beigām. Laiks tiek mērīts šādām funkcijām: <ol style="list-style-type: none"> 1) Validate_View, 2) Parse_XML_File, 3) Create_Javascript_View. Iegūtais rezultāts tiek parādīts statusa joslā kā darbības izpildes laiks (skat. 1.4. nodaļu "Lietotāja saskarne").	
Izvaddati	

Darbības izpildes laiks mikrosekundēs.
Kļūdu u.c. paziņojumi
Nav.

1.3.9. SAPUI5 skatu izpilde iebūvētajā interneta pārlūkā

Identifikators	Run_View
Nosaukums	SAPUI5 skatu izpilde iebūvētajā interneta pārlūkā
Ievads	
Funkcija nodrošina izvēlētajā SAPUI5 skata izpildi programmā iebūvētajā interneta pārlūkā.	
Ievaddati	
.xml vai .js formāta datne.	
Apstrāde	
Ja funkcijai jāizpilda XML skats, tā izsauc funkciju Validate_View. Ja konstatēts, ka dotajam skatam nevar atrast kontrollera datni, tad tiek izsaukta funkcija Create_Controller_File, kas šo datni izveido. Tiek nolasīts skata datnes saturs un noskaidrots skata nosaukums, tips un mape, kurā tas atrodas. Kad tas ir zināms, tiek ģenerēta index.html datne, kas satur norādes uz sākotnējo skatu. Ja kontrollera vai index.html datni nav iespējams saglabāt, tad funkcija izvada atbilstošu kļūdas paziņojumu un pārtrauc savu darbību, pretējā gadījumā izveidotā index.html datne tiek atvērta interneta pārlūkā un izpildās arī izvēlētais skats. Skata izpildes laikā darbojas Measure_View_Execution_Time funkcija.	
Izvaddati	
nav	
Kļūdu u.c. paziņojumi	
1) Nevar izveidot index.html datni. Nav administratora tiesību datu rakstīšanai.	

1.3.10. SAPUI5 skatu izpildes laika mērīšana

Identifikators	Measure_View_Execution_Time
Nosaukums	SAPUI5 skatu izpildes laika mērīšana
Ievads	
Funkcija apkopo informāciju par SAPUI5 skata izpildes laiku programmā iebūvētajā interneta pārlūkā.	
Ievaddati	
Nav	
Apstrāde	
Skata ielādes laikā tiek atzīmēts, cik daudz laika ir pagājis kopš satura ielādes sākuma. Kad skats ir attēlots uz ekrāna, tiek apkopota informācija par visiem skata ielādes posmiem.	
Izvaddati	
Objekts, kas satur šādu informāciju: 1) Satura lejupielādes laiks 2) Satura attēlošanas laiks 3) Kopējais skata izpildes laiks 4) Izpildītā skata datnes nosaukums 5) Skata datnes izmērs 6) Interneta pārlūka versija 7) Skata izpildei patērētais procesora laiks 8) Izmantotais RAM apjoms	
Kļūdu u.c. paziņojumi	
Nav	

1.3.11. SAPUI5 skatu izpildes laika salīdzināšana

Identifikators	Compare_View_Execution_Time
Nosaukums	SAPUI5 skatu izpildes laika salīdzināšana
Ievads	
Funkcija parāda uz ekrāna tabulu, kurā redzams lietotāja izvēlētajā XML skata un programmas automātiski konvertētā Javascript skata izpildes laiks un informācija par patērētajiem datora resursiem. Tiek parādīts arī, kura skata izpilde aizņēma mazāk laika un cik liela ir izpildes laika starpība procentuālā izteiksmē.	
Ievaddati	
Datu struktūra, kas satur izpildīto SAPUI5 skatu izpildes laika informāciju.	
Apstrāde	
No saņemtajiem skatu izpildes datiem tiek veidots HTML kods, kurā definēta tabula ar skatu	

izpildes laika kopsavilkumu.
Izvaddati
HTML kods ar SAPUI5 skatu izpildes laika kopsavilkuma tabulu.
Kļūdu u.c. paziņojumi
nav

1.3.12. SAPUI5 skatu izpildes laika rezultātu eksportēšana HTML formāta datnē

Identifikators	Export_Execution_Time_Results
Nosaukums	SAPUI5 skatu izpildes laika rezultātu eksportēšana HTML formāta datnē.
Ievads	
Funkcija nodrošina iespēju saglabāt datorā pēdējā izpildītā SAPUI5 skata izpildes laika rezultātus HTML formāta datnē.	
Ievaddati	
HTML kods ar SAPUI5 skatu izpildes laika kopsavilkumu.	
Apstrāde	
Lietotāja izvēlētā mapē tiek saglabāta .html datne.	
Izvaddati	
.html formāta datne.	
Kļūdu u.c. paziņojumi	
nav	

1.3.13. SAPUI5 skata kontrollera datnes izveidošana

Identifikators	Create_Controller_File
Nosaukums	SAPUI5 skata kontrollera datnes izveidošana
Ievads	
Funkcija pārbauda, vai dotajam SAPUI5 skatam eksistē kontrollera datne. Ja tādas nav, tad tā tiek izveidota. Funkcija pati nenodrošina skata validāciju, jo tā tiek izsaukta no citām funkcijām, piemēram, Run_View.	
Ievaddati	
Kontrollera nosaukums un mape, kurā tam jāatrodas.	
Apstrāde	

Ja mapē, kurā atrodas SAPUI5 skats, neatrodas kontrollera datne, uz kuru norādīta saite <i>controllerName</i> , tad tiek izveidota šāda tukša datne. Tā ietver sevī kontrollera nosaukumu. Datnes nosaukums tiek veidots pēc šāda parauga: “ <i>controllerName.controller.js</i> ”.
Izvaddati
.js formāta datne.
Kļūdu u.c. paziņojumi
nav

Kontrollera datni ir nepieciešams izveidot, lai būtu iespējams izpildīt SAPUI5 skatu. Ja skatā ir norāde *controllerName*, bet kontrollera datne nav atrodama, tad skats netiks izpildīts, bet interneta pārlūka konsolē būs redzama “datne nav atrasta” kļūda nr. 404.

1.3.14. SAPUI5 skatu izpilde ārējā interneta pārlūkā

Identifikators	Run_View_In_External_Browser
Nosaukums	SAPUI5 skatu izpilde ārējā interneta pārlūkā
Ievads	
Funkcija nodrošina izvēlētā SAPUI5 skata izpildi ārējā interneta pārlūkā, kas izvēlēts kā noklusētais pārlūks.	
Ievaddati	
.xml vai .js formāta datne.	
Apstrāde	
Skatu validācija, kontrollera un index.html datņu saglabāšana notiek tāpat kā funkcijā Run_View (skat. 1.3.9. nodaļu). Tiek noteikts noklusētais interneta pārlūks lietotāja datorā un šis pārlūks tiek atvērts, lai tajā izpildītu SAPUI5 skatu. Skata izpildes laiks netiek mērīts un rādīts uz ekrāna.	
Izvaddati	
nav	
Kļūdu u.c. paziņojumi	
1) Nevar izveidot index.html datni. Nav administratora tiesību datu rakstīšanai.	

1.3.15. Visu SAPUI5 objektu noklusētā XML un Javascript koda apskate

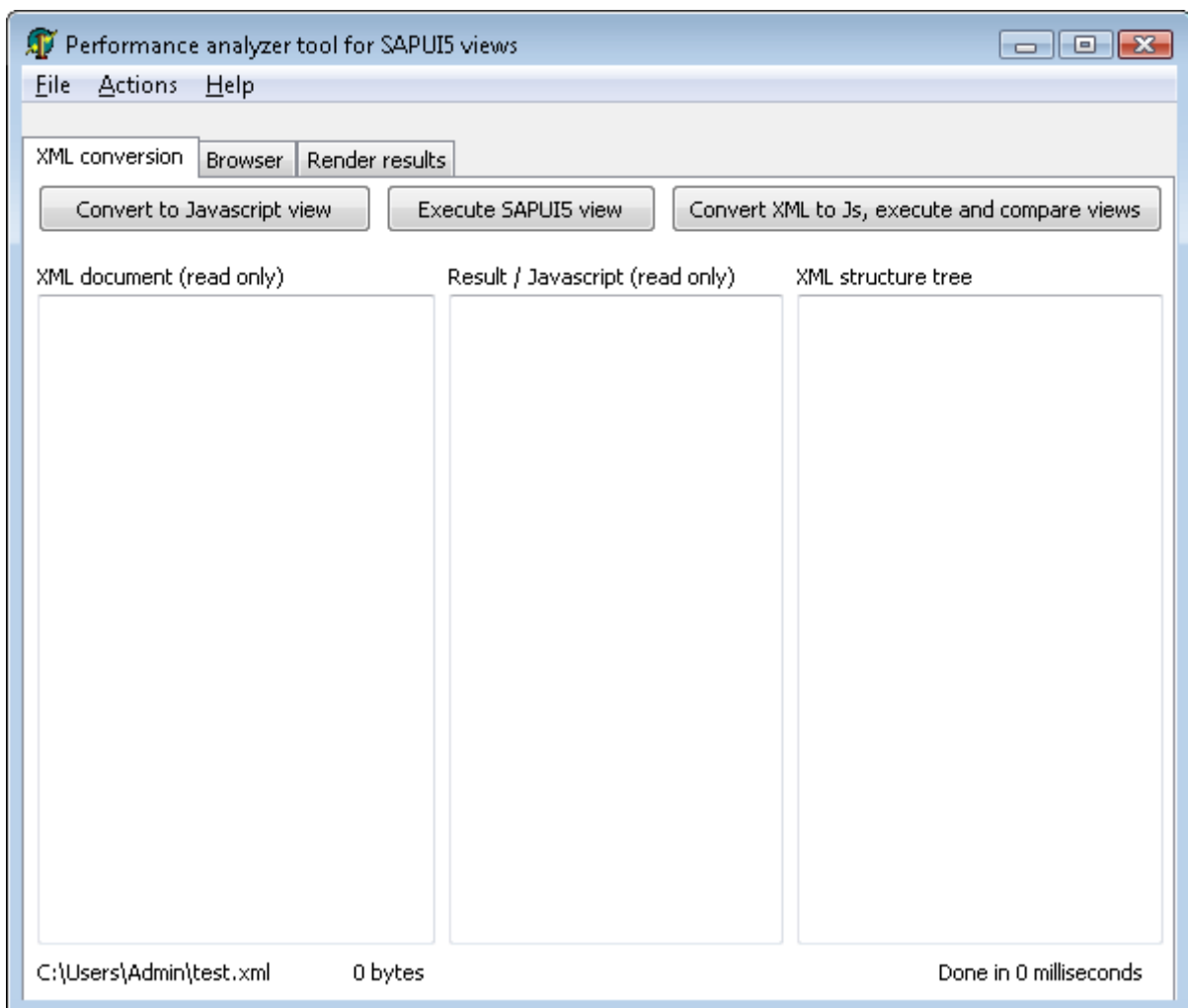
Identifikators	View_Default_Code
Nosaukums	Visu SAPUI5 objektu noklusētā XML un Javascript koda apskate
Ievads	
	Funkcija nodrošina iespēju aplūkot un vajadzības gadījumā arī kopēt jebkura SAPUI5 objekta noklusēto XML un Javascript pirmkodu. Par noklusēto kodu tiek uzskatīts tāds objekta kods, kurā ir ietvertas visas pieejamās īpašības (XML gadījumā – atribūti) ar to noklusētajām vērtībām un komentāri.
Ievaddati	
	Programmas resurss ar SAPUI5 objektu koda paraugiem.
Apstrāde	
	Pēc programmas resursa nolasīšanas nolasītais saturs tiek sadalīts pa daļām, kur katrai daļai atbilst cits SAPUI5 objekts. Uz ekrāna ir redzams SAPUI5 objektu saraksts, kur, noklikšķinot uz izvēlēta objekta nosaukuma, varēs redzēt tā XML un Javascript koda paraugus.
Izvaddati	
	nav
Kļūdu u.c. paziņojumi	
	nav

1.4. Lietotāja saskarne

Lietotāja saskarnei jābūt realizētai kā standarta Windows lietotnei ar noklusēto komponentu izskatu, bez animācijām un citiem vizuāliem efektiem.. Pamatprasības programmas lietotāja saskarnei:

- Palaižot programmu, vispirms jāparāda sākuma ekrāns. Jāpiedāvā lietotājam atvērt jebkuru SAPUI5 XML vai Javascript skatu. Tikai tad, kad vēlamais skats ir izvēlēts, starta ekrānam jāpazūd un jābūt redzamam programmas galvenajam logam. Turpmāk visā programmas darbības laikā starta ekrānam vairs nav jābūt redzamam.
- Programmas logā un jebkurā izvēlnē jābūt redzamām tikai tām pogām un izvēlnes elementiem, ar kuriem ir atļauts veikt darbības. Nav pieļaujami gadījumi, kad lietotājam ir redzamas liekas pogas, kuras neveiks nekādas darbības. Ja kādu darbību dotajā brīdī nevarēs vai nebūs iespējams veikt, tad attiecīgajiem saskarnes elementiem ir jāpazūd.
- Programmas loga augšpusē jābūt izvēlnei, kurā pieejamas visas pamatdarbības. Izvēlnes struktūrai jābūt šādai:
 - Datne
 - Atvērt, aizvērt datni,
 - Saspiest XML kodu,
 - Kopēt XML kodu,
 - Iziet
 - Darbības
 - Parsēt XML kodu,
 - Validēt XML datni,
 - Konvertēt SAPUI5 XML skatu uz ekvivalentu Javascript skatu,
 - Izpildīt atvērto SAPUI5 skatu
 - Palīdzība
 - Palīdzība par programmu
- Mainot programmas loga izmērus, tā elementiem jāpielāgojas jaunajam izmēram un attiecīgi jāspēj izplesties vai sarauties, lai logā nebūtu tukšu laukumu un nekas nelien pāri malām.

- Programmas loga apakšdaļā jābūt statusa joslai. Tajā jāparāda šāda informācija:
 - Datnes ceļš un nosaukums,
 - Datnes izmērs,
 - Skata kontrollera nosaukums,
 - Skata objektu skaits,
 - Pēdējās darbības veikšanai patērētais laiks.
- Galvenā loga saturam jābūt sadalītam 3 cilnēs:
 - Skatu konvertēšana un citas darbības,
 - XML koda priekšskatījums *read only* režīmā,
 - Programmas ģenerētā Javascript koda priekšskatījums *read only* režīmā,
 - XML struktūras koks
 - Interneta pārlūks,
 - Skatu izpildes laika informācija.



1.4.1. attēls. Programmas galvenā loga makets.

- Jābūt logam, kurā var apskatīt visus SAPUI5 objektu koda paraugus. Logs paredzēts funkcijai View_Default_Code. Logā jābūt šādiem saskarnes elementiem:
 - Objektu saraksts
 - Teksta lauks XML koda paraugam
 - Teksta lauks Javascript koda paraugam
 - Izvēlne
 - Kopēt XML kodu
 - Kopēt Javascript kodu
 - Meklēšanas lauks

Meklēšanas laukam jābūt taustiņjutīgam.

Objektu sarakstam jāatrodas loga kreisajā pusē, Javascript kodam – loga vidusdaļā, bet XML kodam – labajā pusē.

- Jābūt palīdzības logam, kurā jāparāda programmas nosaukums un informācija par tās versiju.

1.5. Nefunkcionālās prasības

- 1) Produktam jādarbojas Windows Vista vai jaunākā operētājsistēmā, 32 un 64 bitu versijās.
- 2) Jābūt instalācijas pakotnei (*installer, uninstaller*) 32 un 64 bitu versijās. Tāpat jābūt “pārnēsājamaī” (portable) versijai, kuru var izplatīt kā .zip arhīvu un lietot bez administratora tiesībām. Arī šajā gadījumā jābūt 32 bitu un 64 bitu programmas versijai.
- 3) Programmai jābūt lietojamai uzreiz pēc tās instalēšanas. Lietotājam pašam nav jānodrošina papildu bibliotēkas un servisi.
- 4) Jābūt Unikoda atbalstam.
- 5) Jādarbojas īsinājumaustiņiem (*keyboard shortcuts*). Jābūt šādām kombinācijām:
 - Ctrl+O : Atvērt datni,
 - Ctrl+S: Saglabāt uzģenerēto Javascript skatu,
 - Ctrl+R : Kopēt apstrādes rezultātu (XML analīzi vai Javascript skata kodu),
 - F2 : Apskatīt jebkura SAPUI5 objekta Javascript un XML kodu,
 - F3: Izpildīt šobrīd atvērto SAPUI5 skatu interneta pārlūkā,
 - Esc : Iziēt.
- 6) Jādarbojas *Drag and Drop* mehānismam. Jābūt iespējai atvērt datnes, ievielkot tās ar peli programmas galvenajā logā. Līdzīgi, ievielkot SAPUI5 skata datni interneta pārlūkā, tas nekavējoties jāizpilda.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Dokumenta nolūks

Programmatūras projektējuma aprakstā (PPA) aprakstīts, kādā veidā tiks realizētas izstrādājamā produkta funkcijas. Šis dokuments paredzēts produkta izstrādātājam.

2.2. Tehniskie risinājumi

Produkts tiek izstrādāts *Microsoft Visual Studio Express 2013 for Windows Desktop* vidē, C# programmēšanas valodā.

SAPUI5 skatu paraugi tiek izstrādāti, izmantojot *Eclipse IDE for Java EE developers*. Skati sagatavoti XML formātā un kā Javascript skripti.

2.3. Dekompozīcijas apraksts

Rīks tiks izstrādāts .NET tehnoloģijā, bet SAPUI5 skati – pēc MVC (Model-View-Controller) pieejas. Tādā veidā ar SAPUI5 skatiem saistītais kods tiek sadalīts 3 daļās:

- Modelis (Model) ir *XML schema* (.xsd) datne, kurai jāatbilst jebkuram SAPUI5 XML skatam. Javascript gadījumā var pieņemt, ka modelis ir index.html datne, kas satur saites uz skatiem un kontrolleriem.
- Skats (View) satur informāciju par saskarnes objektiem, kurus paredzēts attēlot uz ekrāna,
- Kontrolleris (Controller) sastāv no funkcijām, kuras tiek izsauktas pie noteiktiem nosacījumiem un spēj mainīt skata objektu izskatu vai uzvedību.

2.3.1. Moduļu apraksts

2.3.1.1. „Main_form” modulis

Datnes nosaukums	Form1.cs
Nolūks	Nodrošināt programmas galvenā loga attēlošanu.
Metodes	<ol style="list-style-type: none">1) SAPUI5 skatu izpildes laika kopsavilkuma attēlošana (Compare_View_Execution_Time)2) Index.html datnes izveidošana (Run_View)3) SAPUI5 skatu kontrollera izveidošana (Create_Controller_File)4) SAPUI5 skatu izpilde ārējā interneta pārlūkā (Run_View_In_External_Browser)5) SAPUI5 skatu izpilde iebūvētajā interneta pārlūkā (Run_View)6) SAPUI5 skatu izpildes laika kopsavilkuma ģenerēšana (Compare_View_Execution_Time)7) SAPUI5 skatu izpildes laika kopsavilkuma eksportēšana (Export_Execution_Time_Results)8) Lietotāja noklusētā interneta pārlūka noteikšana (izsauc funkcija Run_View_In_External_Browser)9) Datnes atvēršana (Open_File)10) XML struktūras koka zīmēšana (Draw_XML_Tree)

2.3.1.2. „JsBeautify” modulis

Datnes nosaukums	JSBeautify.cs
Nolūks	Padarīt iepriekš saspiestu Javascript kodu lasāmu un pārskatāmu, izmantojot <i>jsbeautifylib</i> [1] bibliotēku.

2.3.1.3. „Form_all_controls” modulis

Datnes nosaukums	FormAllControls.cs
Nolūks	Nodrošināt loga attēlošanu, kurā katram SAPUI5 objektam var apskatīt noklusēto kodu XML un Javascript formātā. Par “noklusēto” uzskatāms tāds objekta kods, kurā ir ietvertas visas pieejamās īpašības (XML gadījumā – atribūti) ar to noklusētajām vērtībām.
Metodes	1) Datnes, kas satur visu objektu koda paraugus, ielāde (View_Default_Code)

2.3.1.4. „Js_view” modulis

Datnes nosaukums	JsView.cs
Nolūks	Nodrošināt Javascript skata izveidošanu no XML skata.
Metodes	<ol style="list-style-type: none">1) Javascript skata izveidošana (Create_Javascript_View)<ol style="list-style-type: none">a. Mezgla parsēšanab. Atribūtu parsēšanac. Agregāciju analizēšana un Javascript koda ģenerēšanad. Atribūta vērtības datu tipa pieraksta noteikšanae. Atribūta vērtības pieraksta noteikšanaf. <i>Formatter</i> un <i>Factory</i> parsēšana2) XML datnes parsēšana (Parse_XML_File)<ol style="list-style-type: none">a. Mezgla parsēšanab. Mezgla pozīcijas noteikšanac. Mezgla dziļuma noteikšanad. Īsā ceļa noteikšana līdz mezglame. XPath noteikšana līdz mezglamf. XPath ar pozīcijas numuru noteikšana

2.3.1.5. „XML_functions” modulis

Datnes nosaukums	XMLFunctions.cs
Nolūks	Nodrošināt XML datņu validāciju atbilstoši XML formātam un SAPUI5 skatu formai, kā arī saspiest skata kodu.
Metodes	1) XML datnes validācija (Validate_View)

	<ol style="list-style-type: none"> 2) SAPUI5 XML skata validācija (Validate_View) 3) XML un Javascript koda saspiešana 4) Skata izpildes laika informācijas formatēšana 5) Datnes izmēra informācijas formatēšana
--	---

2.3.1.6. „Browser_Info” modulis

Datnes nosaukums	BrowserInfo.cs
Nolūks	Apkopot un apstrādāt informāciju par SAPUI5 skatu izpildes laiku.
Metodes	<ol style="list-style-type: none"> 1) Konstruktors 2) <i>IsEmpty</i> pārbaude 3) Skata ielādes laika aprēķināšana 4) Izmantoto procesora resursu aprēķināšana 5) Iepriekšējā skata informācijas iztīrīšana

2.3.1.7. „Form_Help” modulis

Datnes nosaukums	FormHelp.cs
Nolūks	Parādīt informāciju par programmu.

2.3.2. Prasību trasējamības matrica

Trasējamības matricā tabulas veidā parādīts, kā paredzēts realizēt PPS minētās prasības. Metožu pierakstā pirms punkta rakstīts klases nosaukums, bet pēc punkta ir pašas metodes nosaukums.

Tabulā nav uzskaitītas dažas palīgmetodes, kuras izpilda tikai nefunkcionālās prasības un metodes, kuras ir saistītas tikai ar lietotāja saskarni, piemēram, dažādas OnResize tipa metodes, DragEnter, DragDrop u.c. Tāpat netiek pieminētas OnClick metodes, kuras faktiski satur tikai izsaukumu uz metodi, kas tad arī izpilda vēlamās darbības.

Funkcija *Measure_Time* ir iebūvēta vairākās metodēs. Tas nozīmē, ka šai funkcijai nav paredzēta atsevišķa metode un tā automātiski izpildās un statusa joslā izveda darbības rezultātus, izsaucot jebkuru no metodēm, kuras attiecas uz *Measure_Time* funkciju.

Funkcionālās prasības identifikators	Pielietojamās metodes
Open_File	FormMain.OpenFile
Validate_View	XMLFunctions.IsValidXml XMLFunctions.IsValidSAPView
Parse_XML_File	XMLFunctions.IsValidXml XMLFunctions.IsValidSAPView JsView.ParseXml JsView.ParseCurrentNode JsView.GetNodePosition JsView.GetNodeDepth JsView.GetShortXPathToNode JsView.GetXPathToNode JsView.GetFullXPathToNode
Draw_XML_Tree	FormMain.DrawXmlTree
Compress_Code	XMLFunctions.CompressJavascript
Create_Javascript_View	JsView.CreateJavascriptView JsView.ParseCurrentJsNode JsView.ParseNodeAttributes JsView.ParseAggregationCode JsView.ParseAttribute

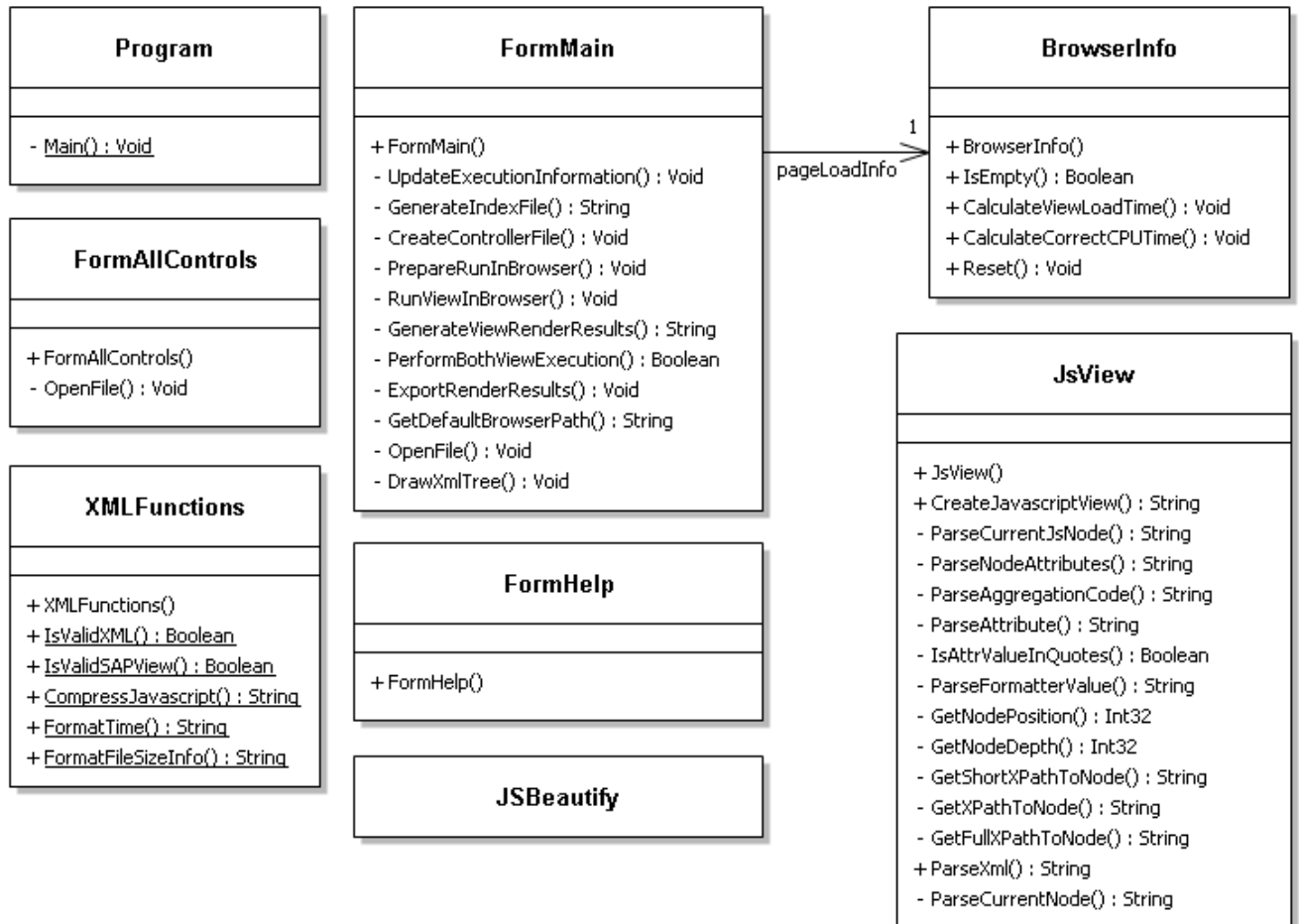
	JsView.IsAttrValueInQuotes JsView.ParseFormatterValue
Beautify_Javascript	<i>Jsbeautifylib</i> bibliotēka [1]
Measure_Time	Iebūvēts šādās metodēs: XMLFunctions.IsValidSAPView JsView.ParseXml JsView.CreateJavascriptView
Run_View	FormMain.RunViewInBrowser FormMain.CreateControllerFile FormMain.GenerateIndexFile
Measure_View_Execution_Time	Iebūvēts šādās metodēs: FormMain.RunViewInBrowser BrowserInfo.CalculateViewLoadTime BrowserInfo.CalculateCorrectCPUtime FormMain.UpdateExecutionInformation
Compare_View_Execution_Time	FormMain.PerformBothViewExecution FormMain.GenerateViewRenderResults FormMain.GenerateRow FormMain.UpdateExecutionInformation
Export_Execution_Time_Results	FormMain.ExportRenderResults
Create_Controller_File	FormMain.CreateControllerFile
Run_View_In_External_Browser	FormMain.PrepareRunInBrowser FormMain.GetDefaultBrowserPath FormMain.RunViewInBrowser FormMain.CreateControllerFile FormMain.GenerateIndexFile
View_Default_Code	FormAllControls.OpenFile

2.3.2.1. tabula. Programmatūras prasību trasējamības matrica.

2.4. Klašu diagramma

Klašu diagrammā redzamas visas programmā esošās klases. Diagrammā netiek rādīti mainīgie. Publiskās metodes atzīmētas ar “+”, bet privātās – ar “-”.

JSBeautify klases saturs nav uzrādīts, jo tā tiek izmantota kā papildu bibliotēka, ko darba autors pats nav izstrādājis vai modificējis [1].



2.4.1. attēls. Projekta klašu diagramma [8]

2.5. Algoritmu apraksts

2.5.1. SAPUI5 XML skata konvertēšana par tam ekvivalentu Javascript skatu

Vispirms tiek nolasīts XML dokumenta saturs un par saknes mezglu kļūst *View* mezgls. Ja tāda nav, tad tiek pieņemts, ka dotais saturs nav SAPUI5 skats pat, ja atbilst XML formāta sintaksei, vai tika dots XML fragments. Saknes mezglam tiek nolasīta atribūta *controllerName* vērtība un tā tiek piešķirta mainīgajam, kurš atbild par kontrollera nosaukumu. Parsējot saknes atribūtus, tiek apkopota informācija par visām tajos nosauktajām vārdkopām, lai vēlāk tās varētu ierakstīt *index.html* datnē, kas tiks ģenerēta kopā ar Javascript skatu. Kad atribūti ir apstrādāti, sākas mezgla (šajā gadījumā saknes) parsēšana uz agregācijām.

Ja, parsējot mezglu, tiek konstatēts, ka tas satur apakšmezglus, tad katrs no šiem apakšmezgiem tiks atsevišķi parsēts jau kā pašreizējā mezgla agregācija. Mezgliem, kuru vecāka mezgls ir sakne, tiek piešķirti arī objektu mainīgo nosaukumi, kuri vēlāk tiks ierakstīti Javascript kodā. Uzreiz tiek ģenerēts arī attiecīgais Javascript gatavā koda fragments. Ja konstatēts, ka pašreizējais mezgls atrodas zem vecāka mezgla *content*, *items* vai tamlīdzīga bloka, tad tam netiek piešķirts mainīgā nosaukums, jo šāds mezgls atradīsies attiecīgajā *content* masīvā.

Ja mezglam ir agregācijas (bērni), tad katrs no tiem tiek parsēts, sākot ar atribūtiem un beidzot ar agregācijām. Tā process var turpināties vairākos līmeņos, kamēr būs iziets cauri visu elementu visām agregācijām. Visbeidzot tiek pabeigta Javascript koda ģenerācija. Javascript kods tiek papildināts ar *sap.ui.jsView* definīciju un *createContent* funkciju. Kad tas ir izdarīts, SAPUI5 Javascript koda ģenerēšana ir pabeigta un iegūto kodu var saglabāt uz diska vai parādīt uz ekrāna.

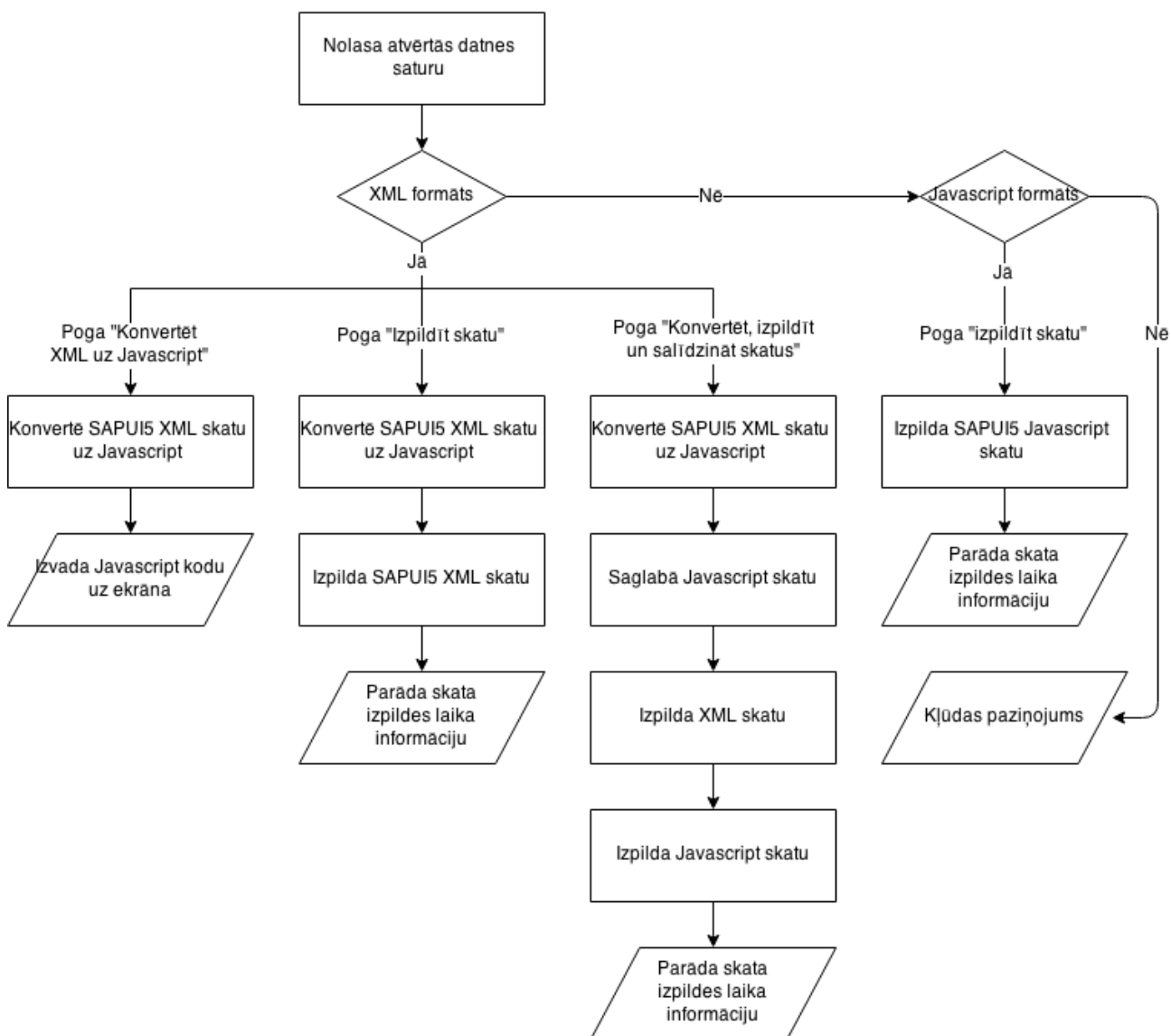
Atribūtu parsēšana

Ja mezgls satur atribūtus, katrs no tiem secīgi tiek parsēts. No atribūta iegūst tā nosaukumu un vērtību. Ja nosaukums ir "id", tad figūriekavās tiek ģenerēts Javascript koda fragments `this.createId(id)`, pretējā gadījumā tiek izsaukta metode *ParseAttribute*, kura veic atribūtu vērtības analīzi. Ja vērtība satur saiti uz *formatter* funkciju vai ceļu (*path*), tad tas tiek attiecīgi apstrādāts. Tiek noteikts kārtējā atribūta vērtības datu tips. Ja tas ir *string*, tad vērtība Javascript kodā būs liekama pēdiņās, bet ja tas ir, piemēram, *int*, tad bez pēdiņām.

Notiek pārbaude, vai atribūta nosaukums ir interpretējams kā notikums (*event*). Ja tas tā ir, tad atribūta vērtība tiek apstrādāta kā funkcijas nosaukums. Tādā gadījumā Javascript kodā šis gadījums tiks noformēts kā funkcijas izsaukums pie noteiktiem apstākļiem, piemēram, `press: oController.hideToolBar.`

1. līmeņa shēma

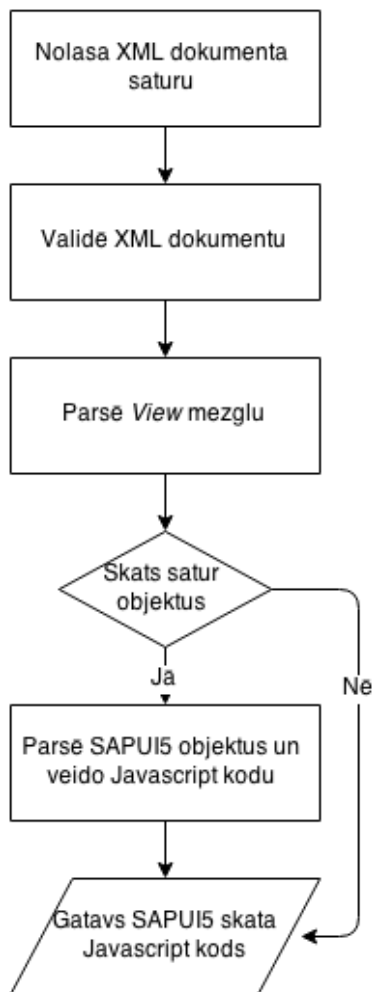
Shēmā redzamas visas darbības, kuras atļauts veikt ar SAPUI5 XML un Javascript skatiem.



2.5.1.1. attēls. Darbības ar SAPUI5 skatiem.

Šajā blokskārtā redzams, ka, veicot jebkuru shēmā attēloto darbību ar XML skatu, tas vispirms jākonvertē Javascript formātā. XML skata konversijas process parādīts 2. līmeņa shēmā.

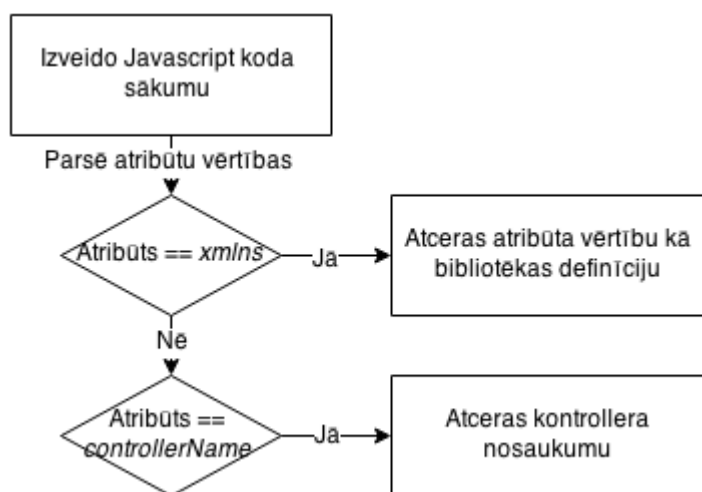
2. līmeņa shēma



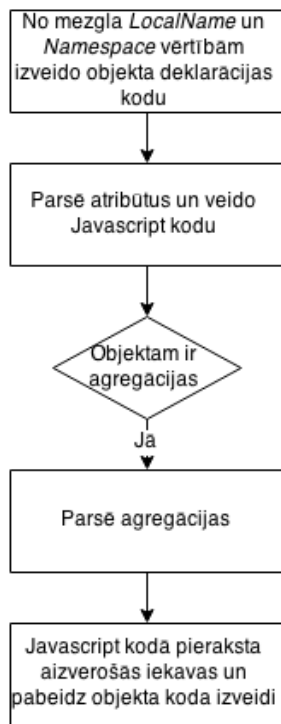
2.5.1.2. attēls. SAPUI5 XML skata konvertēšanas algoritma vienkāršota shēma.

3. līmeņa shēmas

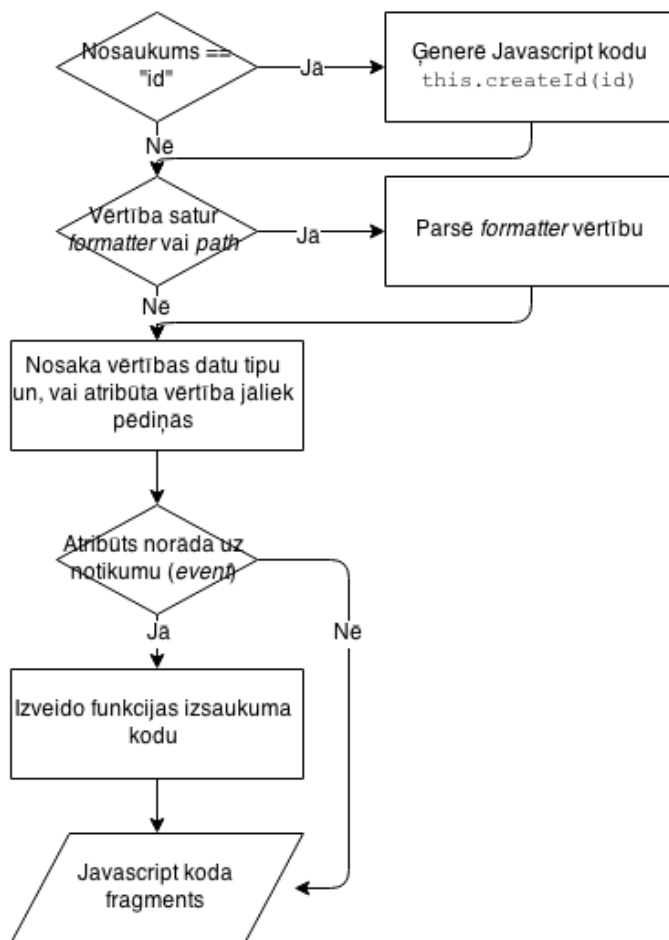
3. līmeņa shēmās sīkāk parādīti 2. līmeņa shēmā redzamie XML skata konvertēšanas posmi.



2.5.1.3. attēls. View mezgla parsēšana.



2.5.1.4. attēls. SAPUI5 objekta parsēšana.



2.5.1.5. attēls. Objekta atribūtu parsēšana.

2.5.2. SAPUI5 skatu izpildes laika mērīšana

Jebkuram programmā izpildītajam SAPUI5 skatam tā izpildes gaitā tiek mērīts skata izpildes laiks, bet pēc tā izpildes tiek apkopota informācija par datora resursu patēriņu. Tiek mērīts šāds laiks:

- Sagatavošanas darbu izpildes laiks. Sagatavošanas darbos ietilpst `index.html` atrašanās vietas noteikšana un šīs datnes satura ģenerēšana un saglabāšana. Ja nav atrasts kontrolleris, tad tam tiek izveidota datne ar tukšu kontrollera saturu.
- Datu lejupielādes laiks. Šeit tiek pieskaitīts arī laiks, kurā dati tiek nolasīti no diska.
- Skata attēlošanas laiks. Lai uz ekrāna attēlotu SAPUI5 objektus, nepieciešams izpildīt šo objektu attēlošanas skriptus (*renderer*). Tiek noteikts, cik ilgi notiek elementu attēlošana pārlūka logā.

Interneta pārlūkam izpildot satura ielādi, izmantojot *Navigate* metodi, nostrādā notikumi, pēc kuriem tiek fiksēts laiks, kad tas ir noticis. Tādā veidā pēc *Navigated* notikuma ir ielādēti SAPUI5 skripti un paša skata pirmkods. Pēc *DocumentCompleted* ir pilnībā pabeigta skatu izpilde, jo viss saturs ir parādīts uz ekrāna. Par skata attēlošanas (*render*) laiku tiek uzskatīts laiks no brīža, kad ir ielādēts viss nepieciešamais pirmkods izņemot objektu *renderer* skriptus, līdz brīdim, kad satura zīmēšana pārlūka logā ir pabeigta. To, ka skata izpilde ir pabeigta un vairs nekas netiek darīts, nosaka arī pēc pazīmes *WebBrowserReadyState*, kura kļūst par *Complete*.



2.5.2.1. attēls. SAPUI5 skata izpildes laika mērīšanas shēma.

3. TESTĒŠANAS DOKUMENTĀCIJA

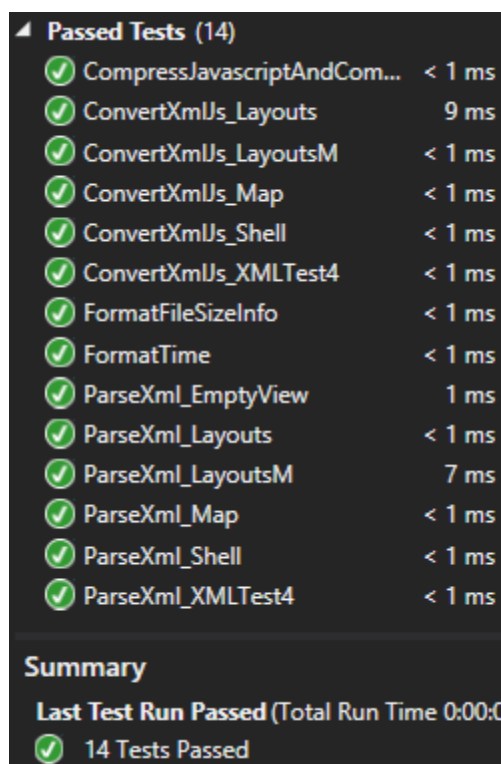
3.1. Testēšanas plāns

Daļa no produkta funkcionalitātes tiek automātiski testēta, izmantojot vienībtestus. Vienībtestiem ir izveidots savs projekts un tie tiek izpildīti *Microsoft Visual Studio* vidē. Vienībtesti veidoti, lai produkta izstrādātājs pārbaudītu, vai veicot izmaiņas programmatūras pirmkodā, nav ieviesis jaunas kļūdas.

Vienībtesti galvenokārt tiek izpildīti pēc šāda scenārija:

1. Tiek nolasīta datne, kurā atrodas sagaidāmie metodes darbības rezultāti,
2. Tiek nolasīta datne, kas satur apstrādājamus datus,
3. Tiek apstrādāti nolasītie dati,
4. Tiek salīdzināts, vai apstrādes rezultāts atbilst sagaidāmajam,
5. Ja rezultāts sakrīt ar sagaidāmo rezultātu, tad kārtējais tests ir iziets, bet ja tas nesakrīt, tad tests nav iziets.

Testu izpildes laikā nekāda informācija netiek saglabāta datorā. Neviena datne netiek modificēta. Testu datnēm ir izveidota mape „Resources”, kas ir iekļauta testu projektā.



3.1. attēls. Vienībtestu izpildes rezultāti.

3.2. Testēšanas žurnāls

Nr.	Testa nosaukums	Parametri	Rezultāts
1	CompressJavascriptAndCompare (XMLFunctions.CompressJavascript)	LayoutsM.compressed.js LayoutsM.uncompressed.view.js	Izpildās
2	ConvertXmlJs_LayoutsM (JsView.CreateJavascriptView)	LayoutsM.view.xml LayoutsM.view.js	Izpildās
3	ConvertXmlJs_Map (JsView.CreateJavascriptView)	Map.view.xml Map.view.js	Izpildās
4	ConvertXmlJs_Shell (JsView.CreateJavascriptView)	Shell.view.xml Shell.view.js	Izpildās
5	ConvertXmlJs_XMLTest4 (JsView.CreateJavascriptView)	XMLTest4.view.xml XMLTest4.view.js	Izpildās
6	ConvertXmlJs_Layouts (JsView.CreateJavascriptView)	Layouts.view.xml Layouts.view.js	Izpildās
7	FormatFileSizeInfo (FormMain klases metode)	Datnes izmērs baitos	Izpildās
8	FormatTime (FormMain klases metode)	Laiks milisekundēs, sekunžu pieraksts, milisekunžu pieraksts	Izpildās
9	ParseXml_EmptyView (JsView.ParseXml)	EmptyView.xml EmptyView.txt	Izpildās
10	ParseXml_Layouts (JsView.ParseXml)	Layouts.view.xml Layouts.txt	Izpildās
11	ParseXml_LayoutsM (JsView.ParseXml)	LayoutsM.view.xml LayoutsM.txt	Izpildās
12	ParseXml_Map (JsView.ParseXml)	Map.view.xml Map.txt	Izpildās
13	ParseXml_Shell (JsView.ParseXml)	Shell.view.xml Shell.txt	Izpildās
14	ParseXml_XMLTest4 (JsView.ParseXml)	XMLTest4.view.xml XMLTest4.txt	Izpildās

3.2.1. tabula. Vienībtestu izpildes rezultāti.

3.2.1. tabulā redzami visi vienībtesti un to izpildes rezultāti.

1. testā testēta `Compress_Code` funkcija, kurai tiek padota Javascript datne. Funkcija izpilda Javascript koda saspiešanu (skat. 1.3.5. nodaļu). Iegūtais rezultāts tiek salīdzināts ar vēlamu rezultātu. Ja tie sakrīt, tad tests ir izziets.

No 2. līdz 6. testam tiek testēta `Create_Javascript_View` funkcija (skat. 1.3.6. nodaļu). Kā parametri ir norādīti datņu nosaukumi, kuras izmantotas šajos testos.

7. testā pārbaudītā `FormatFileSizeInfo` funkcija ir pielietota gadījumos, kad statusa joslā, SAPUI5 skatu izpildes rezultātu tabulās vai citās vietās jāpārveido informācija par datnes izmēru, piemēram, "100" jāpārveido par "100 bytes", bet "6071" – par "5,92 kB".

8. testā pārbaudītā `FormatTime` funkcija līdzīgi, kā `FormatFileSizeInfo`, pārveido mikrosekunžu pierakstu par laika pierakstu sekundēs, veicot arī rezultāta noapaļošanu.

No 9. līdz 14. testam ir testēta `Parse_XML_File` funkcija (skat. 1.3.3. nodaļu). Kā parametri ir norādīti datņu nosaukumi, kuras izmantotas šajos testos.

Citas PPS aprakstītās funkcijas tika testētas manuāli. Izstrādātājs nolēma, ka automatizēti testēt tādas funkcijas, kā piemēram, skatu izpildes laika mērīšanu, skatu pareizu izpildi programmā iebūvētajā un ārējā interneta pārlūkā, skatu izpildes laika salīdzināšanu u.c., nevar testēt ar vienībtestiem.

4. PROJEKTA ORGANIZĀCIJA

Produkts tika izstrādāts *Microsoft Visual Studio* vidē, *C#* programmēšanas valodā.

Vispirms notika darbs pie XML skatu parsēšanas funkcijas izstrādes. Kad no XML mezgliem kļuva iespējams iegūt visu nepieciešamo informāciju un sadalīt to vajadzīgajās daļās, tika uzsākts darbs pie SAPUI5 skatu konversijas. Tikai tad, kad bija iespējams arī konvertēt skatus, tika uzsākta pārējās funkcionalitātes izstrāde.

Darbs tika organizēts pēc iteratīvās pieejas. Sākumā bija definētas prasības un izstrādāts projektējums, tomēr darba izstrādes gaitā radās nepieciešamība gan ieviest jaunu funkcionalitāti, gan labot dažu jau pabeigto funkciju aprakstus. Tas daļēji skaidrojams arī ar faktu, ka nebija skaidri zināms, vai sākotnēji plānotais SAPUI5 skatu konversijas algoritms patiešām darbosies korekti un nodrošinās pareizu agregāciju apstrādi. Tāpat sākumā nebija paredzēts, ka skatu izpildei būs obligāti jāģenerē *index.html* un kontrolleru datnes, ja tādas neeksistē.

Pirmos SAPUI5 skatus kļuva iespējams konvertēt un izpildīt no 2015.g. 26. marta, tomēr uzlabojumi notika arī vēlāk, jo sākotnēji netika atsevišķi apstrādātas agregācijas un dažādas skatu papildu iespējas.

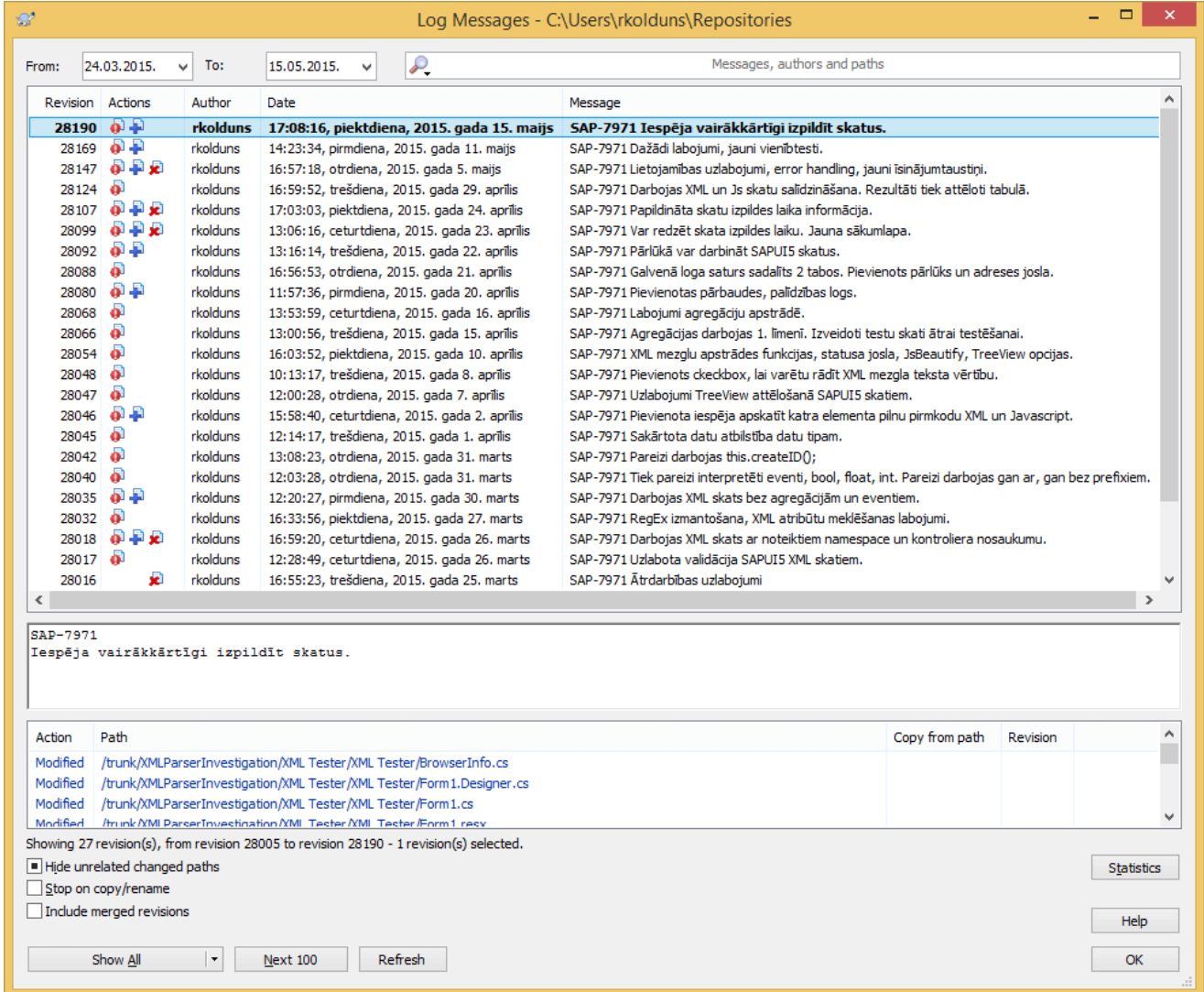
SAPUI5 skati, kas paredzēti testēšanas nolūkiem, izstrādāti *Eclipse IDE for Java EE Developers* izstrādes vidē.

Projekta dokumentācijas automātiskai ģenerēšanai tika izmantots *doxygen* [4]. Tas izveido tīmekļa dokumentus, kuros var apskatīt klašu hierarhiju, izmantotās metodes un mainīgos. Ir redzama projekta struktūra.

Veicamo darbu uzskaitē ir izmantota Atlassian JIRA sistēma [5]. *Jiras* tika veidotas svarīgāko uzdevumu izpildei. Katrā *Jirā* bija prognozēta veicamā uzdevuma darbietilpība un atzīmēts faktiskais patērētais laiks. Kopumā var secināt, ka vairumā gadījumu prognozētā darbietilpība pārsniedza faktisko uzdevuma veikšanai patērēto laiku, tomēr izstrādātājs uzskata prognozes par precīzām un pamatotām.

5. KONFIGURĀCIJU PĀRVALDĪBA

Versiju kontrolei ir izmantots *SVN*. Regulāri tika augšupielādēts programmas kods un izpildāmā datne. Tādā veidā tika nodrošināta iespēja vajadzības gadījumā ne tikai atgūt programmas pirmkodu, bet arī pašu programmu un salīdzināt tās darbību ar jaunāko versiju.



Log Messages - C:\Users\rkolduns\Repositories

From: 24.03.2015. To: 15.05.2015. Messages, authors and paths

Revision	Actions	Author	Date	Message
28190		rkolduns	17:08:16, piektdiena, 2015. gada 15. maijs	SAP-7971 Iespēja vairākkārtīgi izpildīt skatus.
28169		rkolduns	14:23:34, pirmdiena, 2015. gada 11. maijs	SAP-7971 Dažādi labojumi, jauni vienbtesti.
28147		rkolduns	16:57:18, otrdiena, 2015. gada 5. maijs	SAP-7971 Lietojamības uzlabojumi, error handling, jauni īsinājumaustīpi.
28124		rkolduns	16:59:52, trešdiena, 2015. gada 29. aprīlis	SAP-7971 Darbojas XML un Js skatu salīdzināšana. Rezultāti tiek attēloti tabulā.
28107		rkolduns	17:03:03, piektdiena, 2015. gada 24. aprīlis	SAP-7971 Papildināta skatu izpildes laika informācija.
28099		rkolduns	13:06:16, ceturtdiena, 2015. gada 23. aprīlis	SAP-7971 Var redzēt skata izpildes laiku. Jauna sākumlapa.
28092		rkolduns	13:16:14, trešdiena, 2015. gada 22. aprīlis	SAP-7971 Pārlikā var darbināt SAPUI5 skatus.
28088		rkolduns	16:56:53, otrdiena, 2015. gada 21. aprīlis	SAP-7971 Galvenā loga saturs sadalīts 2 tabos. Pievienots pārliks un adreses josla.
28080		rkolduns	11:57:36, pirmdiena, 2015. gada 20. aprīlis	SAP-7971 Pievienotas pārbaudes, palīdzības logs.
28068		rkolduns	13:53:59, ceturtdiena, 2015. gada 16. aprīlis	SAP-7971 Labojumi agregāciju apstrādē.
28066		rkolduns	13:00:56, trešdiena, 2015. gada 15. aprīlis	SAP-7971 Agregācijas darbojas 1. līmenī. Izveidoti testu skati ātrai testēšanai.
28054		rkolduns	16:03:52, piektdiena, 2015. gada 10. aprīlis	SAP-7971 XML mezglu apstrādes funkcijas, statusa josla, JsBeautify, TreeView opcijas.
28048		rkolduns	10:13:17, trešdiena, 2015. gada 8. aprīlis	SAP-7971 Pievienots ckeckbox, lai varētu rādīt XML mezgla teksta vērtību.
28047		rkolduns	12:00:28, otrdiena, 2015. gada 7. aprīlis	SAP-7971 Uzlabojumi TreeView attēlošanā SAPUI5 skatiem.
28046		rkolduns	15:58:40, ceturtdiena, 2015. gada 2. aprīlis	SAP-7971 Pievienota iespēja apskatīt katra elementa pilnu pirmkodu XML un Javascript.
28045		rkolduns	12:14:17, trešdiena, 2015. gada 1. aprīlis	SAP-7971 Sakārtota datu atbilstība datu tipam.
28042		rkolduns	13:08:23, otrdiena, 2015. gada 31. marts	SAP-7971 Pareizi darbojas this.createID();
28040		rkolduns	12:03:28, otrdiena, 2015. gada 31. marts	SAP-7971 Tiek pareizi interpretēti eventī, bool, float, int. Pareizi darbojas gan ar, gan bez prefixiem.
28035		rkolduns	12:20:27, pirmdiena, 2015. gada 30. marts	SAP-7971 Darbojas XML skats bez agregācijām un eventiem.
28032		rkolduns	16:33:56, piektdiena, 2015. gada 27. marts	SAP-7971 RegEx izmantošana, XML atribūtu meklēšanas labojumi.
28018		rkolduns	16:59:20, ceturtdiena, 2015. gada 26. marts	SAP-7971 Darbojas XML skats ar noteiktiem namespace un kontroliera nosaukumu.
28017		rkolduns	12:28:49, ceturtdiena, 2015. gada 26. marts	SAP-7971 Uzlabota validācija SAPUI5 XML skatiem.
28016		rkolduns	16:55:23, trešdiena, 2015. gada 25. marts	SAP-7971 Ātrdarbības uzlabojumi

SAP-7971
Iespēja vairākkārtīgi izpildīt skatus.

Action	Path	Copy from path	Revision
Modified	/trunk/XMLParserInvestigation/XML Tester/XML Tester/BrowserInfo.cs		
Modified	/trunk/XMLParserInvestigation/XML Tester/XML Tester/Form1.Designer.cs		
Modified	/trunk/XMLParserInvestigation/XML Tester/XML Tester/Form1.cs		
Modified	/trunk/XMLParserInvestigation/XML Tester/XML Tester/Form1.resx		

Showing 27 revision(s), from revision 28005 to revision 28190 - 1 revision(s) selected.

Hide unrelated changed paths
 Stop on copy/rename
 Include merged revisions

Show All Next 100 Refresh

Statistics Help OK

5.1. attēls. Repozitorija pārlika logs.

6. KVALITĀTES NODROŠINĀŠANA

Produkta kvalitātes nodrošināšanas aktivitāšu galvenais mērķis ir panākt visu pasūtītāja prasību īstenošanu un realizēšanu atbilstoši programmatūras prasību specifikācijai.

Pirms produkta izstrādes uzsākšanas notika iepazīšanās ar SAPUI5 skatu darbības principiem. Tika noskaidrots, ar ko savā starpā atšķiras katrs no šiem četriem veidiem un kādas ir to priekšrocības un trūkumi. Tika izstudēti arī svarīgi termini, piemēram, agregācijas, asociācijas, *formatter* un *factory* funkcijas u.c.

Notika konsultācijas ar projekta vadītāju un projekta komandu, lai saskaņotu izstrādes gaitu un vienotos par atsevišķām niansēm, kuras nevarēja paredzēt pašā sākumā.

Lai nodrošinātu izstrādājamā produkta koda kvalitāti, tika izmantots *StyleCop* [2] rīks, kas paredzēts *Visual Studio* videi. Galvenās programmēšanas stila nepilnības, ko izdevās novērst, izmantojot *StyleCop*, ir:

- Neatbilstoši mainīgo un metožu nosaukumi (nepareizi lielie un mazie sākuma burti),
- Atsevišķās vietās nav lietots *this* rādītājs,
- Nav tukšu rindu pēc *for* un *if* blokiem,
- Nav atstarpju matemātiskajās izteiksmēs un formulās,
- „Using” nav sašķiroti alfabētiskā kārtībā,
- Nav izmantoti iebūvētie mainīgie, piemēram, *string.Empty* vai *Environment.NewLine*.

Rakstot programmas kodu, ir ievērots DRY princips.

Veicamo uzdevumu uzskaitēi izmantota Atlassian JIRA [5] sistēma.

Produkta izstrādes laikā tika izstrādāti un regulāri izpildīti vienībtesti (skat. nodaļu “*Testēšanas dokumentācija*”). Tas ļāva ātri atklāt un labot kļūdas. Ir veikti arī manuālie testi, ar kuriem tika pārbaudīta programmas uzvedība un darbība kopumā.

Izmantojot *doxygen* rīku [4], programmatūras produkta izstrādes beigās tika veikta projektējuma dokumentācijas automatizēta ģenerēšana un salīdzināšana ar sākotnēji izstrādāto programmatūras projektējuma aprakstu.

Rakstot dokumentāciju, izmantoti Akadēmiskās terminu datubāzes ieteiktie svešvārdu tulkojumi [3].

Produkta izstrādes noslēguma stadijā vairākiem ar SAPUI5 pazīstamiem cilvēkiem tika piedāvāts pirmo reizi izstrādātāja klātbūtnē izmēģināt izstrādāto programmu. Tādā veidā tika atklātas un izlabotas lietotāja saskarnes nepilnības un uzlabota lietojamība.

7. DARBIETILPĪBAS NOVĒRTĒJUMS

7.1. Sākotnējais darbietilpības novērtējums

Ņemot vērā, ka izstrādātājam ir visai ierobežotas zināšanas un maza pieredze darbā ar SAPUI5 rīkkopu, tika nolemts, ka tradicionālās metodes, piemēram, COCOMO II, nevarēs sniegt ticamu darbietilpības aprēķinu.

Konsultējoties ar projekta vadītāju pēc „eksperta metodes”, noskaidrojās, ka projekta izstrāde varētu aizņemt vismaz 3 personmēnešus, turklāt ir plašas iespējas paplašināt pamata funkcionalitāti pat gadījumā, ja tā tiks izstrādāta ātrāk nekā 3 mēnešos.

7.2. Darbietilpības novērtējums pēc produkta nodošanas ekspluatācijā

Pieņemot, ka 1 persondiena atbilst 8 stundām, bet 1 personmēnesī ir 176 stundas, programmprodukts faktiski tika izstrādāts aptuveni 3 personmēnešos. Šajā laikā gan ir ietverta tikai programmatūras izstrāde un testēšana, un dokumentācijas izstrāde. Pirms programmatūras izstrādes sākuma autors apguva SAPUI5 skatu paradigmas; šis process aizņēma vēl apmēram 3 personnedēļas.

Programmatūras izstrādes gaitā izstrādātājs ir uzrakstījis aptuveni 3200 pirmkoda rindiņu. Papildus tam, ir izstrādāti vienībtesti un SAPUI5 testa XML skati.

8. LIETOTĀJA CEĻVEDIS

Šis ceļvedis ir veidots kā īsa pamācība, kā sagatavot darbam datorsistēmu, kurā plānots strādāt ar SAPUI5 skatiem. Tajā netiek aprakstīta katra programmas funkcija.

8.1. Instalēšanas un palaišanas pamācība

Lai darbinātu SAPUI5 skatu veikspējas analīzes rīku, lietotāja datorā jābūt instalētam .NET Framework 4.0 vai jaunākai ietvara versijai. Pretējā gadījumā lietotājam tiks palūgta atļauja to automātiski pieinstalēt.

Programma darbojas visās Windows operētājsistēmās, sākot ar Windows Vista. Tai ir 32 bitu un 64 bitu versijas.

Programmai ir pieejama instalācijas pakotne. Instalēšanas procedūra notiek standarta secībā – pēc *Setup.exe* palaišanas ir jāpiekrīt licences līgumam, jāizvēlas instalācijas mape, instalācijas tips un pēc vēlēšanās var izveidot darbvirsma ikonu.

Ja instalēšanas laikā ir izvēlēts “*Full installation*” tips, tad *Program Files* mapē kopā ar programmas izpildāmo datni atradīsies arī *resources.zip* arhīvs, kas satur SAPUI5 skatu izpildei nepieciešamo *resources* mapi ar visiem skriptiem. Šo arhīvu vajag pašrocīgi atarhivēt tajā pašā vietā, kur jāatrodas SAPUI5 skatiem. Minēto soli var izlaist, taču tādā gadījumā, izpildot skatus, visi izpildei nepieciešamie skripti tiks lejupielādēti no interneta resursiem.

Ja ir iegūta (lejupielādēta) programmas pārnēsājamā versija .zip arhīvā, tad šo arhīvu nepieciešams tikai atarhivēt. Nekādas papildu darbības nav obligāti jāveic.

Interneta pieslēgums ir obligāts, tomēr programmas darbības laikā tas tiks izmantots tikai gadījumos, ja SAPUI5 skati tiek izpildīti apstākļos, kad datorā blakus SAPUI5 skatiem neatrodas “*resources*” mape, kas satur visus SAPUI5 skriptus. Tādā gadījumā SAPUI5 skripti, piemēram, *core.js*, tiks lejupielādēti no SAP interneta vietnes. Ja resursu mape ir atrasta, tad visi skripti tiks iegūti no šīs mapes.

Lai izpildītu SAPUI5 skatus, tiek ģenerēta *index.html* datne. Tā kā pēc noklusējuma SAPUI5 kods tiek iegūts no interneta resursiem, *index.html* datnē tiek norādīta šāda adrese:

```
<script src="https://sapui5.hana.ondemand.com/resources/sap-ui-core.js"
```

Ar to pietiek, lai jebkurā datorā būtu iespējams nekavējoties izpildīt skatus, tomēr jāņem vērā, ka šajā gadījumā visi SAPUI resursi (*resources* mapes saturs) tiek lejupielādēti no interneta katru reizi, kad tiek darbināts izvēlētais skats.

Lai, darbinot izvēlētos skatus, SAPUI skripti netiktu ielādēti no interneta, `index.html` datnē `script src` avots jāmaina uz šādu:

```
<script src="resources/sap-ui-core.js"
```

Tas nozīmē, ka turpmāk visi skripti tiks ielādēti tieši no norādītās mapes, nevis lejupielādēti no interneta resursiem. Tomēr šajā gadījumā būs nepieciešams nodrošināt *resources* mapes klātbūtni vienā mapē ar `index.html` datni.

8.2. SAPUI5 skatu konversija

Lai arī programma piedāvā iespēju konvertēt SAPUI5 XML skatu par tam ekvivalentu SAPUI5 Javascript skatu, ir jāņem vērā, ka Javascript kods tiek ģenerēts automatizēti un līdz ar to koda kvalitāte nav augsta. Zemāk tiek uzskaitīti apstākļi, kas jāņem vērā, konvertējot skatus:

- Mainīgo nosaukumi nesatur semantisko jēgu. Tā kā tie tiek automātiski ģenerēti, tie tiek izvēlēti pēc principa: objekta tips + objekta kārtas numurs, piemēram, “Button1”, “TextField20” utml. Nav paredzēta iespēja piešķirt mainīgo nosaukumus atbilstoši “id”atribūta vērtībai, jo šis atribūts nav obligāts.
- Pastāv iespēja, ka pat pēc Javascript koda lasāmības uzlabošanas, tajā būs atrodami koda fragmenti, kur vienā rindā atrodas 2 funkciju izsaukumi vai vairākas dažāda veida iekavas, jauna rinda var sāktie ar aizverošajām iekavām u.c. Piemēram, ir iespējams iegūt rindas ar šādu saturu:
 - `}), new sap.m.Panel({`
 - vai
 - `}), 1,`
- Ja XML skatā kāds atribūts nesatur vērtību, tad Javascript skatā šis atribūts netiks iekļauts. Piemēram, Javascript kodā netiks iekļauts `text=""`.
- Ģenerētais Javascript kods nesatur komentārus. Ja XML kodā bijuši komentāri, tie tiek pazaudēti.

NOBEIGUMS

Kvalifikācijas darba rezultāts ir pilnībā funkcionējošs un sākotnēji definētajām prasībām atbilstošs produkts.

Var secināt, ka darba ietvaros izstrādātais programmaprodukts sevi attaisno un pierāda, ka Javascript skatu lietošana uzlabo veiktspēju, salīdzinājumā ar ekvivalentiem XML skatiem. Vairumam skatu veiktspējas starpība veido apmēram 5-10 %, tomēr gadījumos, kad skatā ir daudzi un savā starpā atšķirīgi maza izmēra vizuālie elementi, Javascript skati kļūst pat līdz 30..35 % efektīvāki par līdzvērtīgiem XML skatiem.

Darba izstrādes gaitā tika gūta pieredze darbā ar *Visual Studio* vidi un būtiski paplašinātas C# valodas zināšanas. Autors ir iepazinis vienībtestēšanas papildu iespējas, piemēram, ir nodrošināta vienībtestu izpildes iespēja, kuros no projekta resursiem jānolasa teksta datnes.

Lai arī kvalifikācijas darba ietvaros izstrādātais programmatūras produkts ir uzskatāms par gatavu nodošanai ekspluatācijā, to var turpināt uzlabot.

IZMANTOTĀ LITERATŪRA

1. Jsbeautifylib bibliotēka. [tiešsaiste] Pieejams: <http://jsbeautifylib.codeplex.com/>
2. StyleCop rīks. [tiešsaiste] Pieejams: <http://stylecop.codeplex.com/>
3. Akadēmiskā terminu datubāze. [tiešsaiste] Pieejams: <http://termini.lza.lv/term.php>
4. Doxygen. [tiešsaiste] Pieejams: <http://www.stack.nl/~dimitri/doxygen/index.html>
5. Atlassian JIRA. [tiešsaiste] Pieejams: <https://www.atlassian.com/software/jira>
6. JSHint. [tiešsaiste] Pieejams: <http://jshint.com/>
7. SAPUI5 dokumentācija. [tiešsaiste] Pieejams:
<https://sapui5.netweaver.ondemand.com/sdk/#docs/api/symbols/sap.ui.html>
8. NClass. [tiešsaiste] Pieejams: <http://nclass.sourceforge.net/>

PROGRAMMATŪRAS PIRMKODS

XML testa skats LayoutsM.view.xml

```
<core:View xmlns:core="sap.ui.core" xmlns:mvc="sap.ui.core.mvc"
xmlns="sap.ui.commons" xmlns:tbl="sap.ui.table" xmlns:lyt="sap.ui.layout"
xmlns:clyt="sap.ui.commons.layout" xmlns:m="sap.m"
controllerName="xmlviewtest.LayoutsM"
xmlns:html="http://www.w3.org/1999/xhtml">

    <lyt:HorizontalLayout visible="true" enabled="true">
        <lyt:content>

            <lyt:VerticalLayout visible="true">
                <lyt:content>
                    <Label id="LblChooseContry1" design="Bold"
text="Izvēlies valsti..."/>
                    <Label id="LblChosenContry2" text=""/>
                    <ListBox visibleItems="3" select="">
                        <items>
                            <core:Item text="Igaunija"/>
                            <core:Item text="Latvija"/>
                            <core:Item text="Lietuva"/>
                        </items>
                    </ListBox>
                </lyt:content>
            </lyt:VerticalLayout>

            <lyt:VerticalLayout visible="true" enabled="true" width="80%">
                <lyt:content>
                    <lyt:HorizontalLayout visible="true"
enabled="true">
                        <lyt:content>
                            <Button id="MyButton1" text="Button
1" press="btnTestJs" visible="true" enabled="true"/>
                            <TextField id="MyTextField"
value="Some text" editable="true" maxLength="20"/>
                        </lyt:content>
                    </lyt:HorizontalLayout>

                    <lyt:HorizontalLayout>
                        <lyt:content>
                            <CheckBox visible="true"
checked="false" text="Sample checkbox" enabled="true"/>
                            <DropDownBox xmlns="sap.ui.commons"
visible="true" value="(Not selected)" enabled="true" editable="true" name=""
maxPopupItems="10"/>
                            <RadioButton xmlns="sap.ui.commons"
text="Yes" valueState="None" groupName="sapUiRbDefaultGroup"/>
                        </lyt:content>
                    </lyt:HorizontalLayout>

                    <Panel xmlns="sap.m" headerText="" width="100%"
height="auto" expandable="false" expanded="false" expandAnimation="true">
                        <content>
                            <Text xmlns="sap.m" text="This is a
panel"/>
                        </content>
                    </Panel>

                    <Panel xmlns="sap.m" headerText="" width="100%"
height="auto" expandable="false" expanded="false" expandAnimation="true">
                        <content>
                            <Text xmlns="sap.m" text="Lorem ipsum
dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt
ut labore et dolore magna aliquyam erat, sed diam voluptua."/>
                        </content>
                    </Panel>
                </lyt:content>
            </lyt:VerticalLayout>
        </lyt:content>
    </lyt:HorizontalLayout>
</core:View>
```

```

        </content>
    </Panel>
</lyt:content>
</lyt:VerticalLayout>

</lyt:content>
</lyt:HorizontalLayout>

<tbl:Table title="Test table with columns" width="1200px"
columnHeaderVisible="true" visibleRowCount="10" firstVisibleRow="0"
selectedIndex="-1" minAutoRowCount="5" fixedBottomRowCount="0">
    <tbl:columns>
        <tbl:Column name="C1"/>
        <tbl:Column name="C2"></tbl:Column>
        <tbl:Column flexible="true" resizable="true"
visible="true" name="C3"/>
    </tbl:columns>
</tbl:Table>

</core:View>

```

Javascript testa skats LayoutsM.view.js

```

sap.ui.jsview("xmlviewtest.LayoutsM", {
    getControllerName: function () {
        return "xmlviewtest.LayoutsM";
    },
    createContent: function (oController) {
        var oContent = [];
        var oHorizontalLayout1 = new sap.ui.layout.HorizontalLayout({
            visible: true,
            enabled: true,
            content: [new sap.ui.layout.VerticalLayout({
                visible: true,
                content: [new
sap.ui.commons.Label(this.createId("LblChooseContry1"), {
                    design: "Bold",
                    text: "Izvēlies valsti...",
                }), new sap.ui.commons.Label(this.createId("LblChosenContry2"),
{}), new sap.ui.commons.ListBox({
                    visibleItems: 3,
                    items: [new sap.ui.core.Item({
                        text: "Igaunija",
                    }), new sap.ui.core.Item({
                        text: "Latvija",
                    }), new sap.ui.core.Item({
                        text: "Lietuva",
                    })], 1,
                )], 1,
            )], new sap.ui.layout.VerticalLayout({
                visible: true,
                enabled: true,
                width: "80%",
                content: [new sap.ui.layout.HorizontalLayout({
                    visible: true,
                    enabled: true,
                    content: [new sap.ui.commons.Button(this.createId("MyButton1"),
{
                        text: "Button 1",
                        press: oController.btnTestJs,
                        visible: true,
                        enabled: true,
                    }), new sap.ui.commons.TextField(this.createId("MyTextField"),
{
                        value: "Some text",
                        editable: true,
                        maxLength: 20,
                    }

```

```

    }}, 1,
  )), new sap.ui.layout.HorizontalLayout({
    content: [new sap.ui.commons.CheckBox({
      visible: true,
      checked: false,
      text: "Sample checkbox",
      enabled: true,
    }), new sap.ui.commons.DropdownBox({
      visible: true,
      value: "(Not selected)",
      enabled: true,
      editable: true,
      maxPopupItems: 10,
    }), new sap.ui.commons.RadioButton({
      text: "Yes",
      valueState: "None",
      groupName: "sapUiRbDefaultGroup",
    })], 1,
  )), new sap.m.Panel({
    width: "100%",
    height: "auto",
    expandable: false,
    expanded: false,
    expandAnimation: true,
    content: [new sap.m.Text({
      text: "This is a panel",
    })], 1,
  )), new sap.m.Panel({
    width: "100%",
    height: "auto",
    expandable: false,
    expanded: false,
    expandAnimation: true,
    content: [new sap.m.Text({
      text: "Lorem ipsum dolor sit amet, consetetur sadipscing
elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam
erat, sed diam voluptua.",
    })], 1,
  )), 1,
  )), 1,
});
oContent.push(oHorizontalLayout1);
var oTable3 = new sap.ui.table.Table({
  title: "Test table with columns",
  width: "1200px",
  columnHeaderVisible: true,
  visibleRowCount: 10,
  firstVisibleRow: 0,
  selectedIndex: -1,
  minAutoRowCount: 5,
  fixedBottomRowCount: 0,
  columns: [new sap.ui.table.Column({
    name: "C1",
  }), new sap.ui.table.Column({
    name: "C2",
  }), new sap.ui.table.Column({
    flexible: true,
    resizable: true,
    visible: true,
    name: "C3",
  })], 1,
});
oContent.push(oTable3);
return oContent;
}
});

```

Javascript skata izveides koda fragmenti

```
/// <created>
/// <author>Romāns Kolduns</author>
/// <date>2015.03.30</date>
/// </created>
/// <summary>
/// Modulis nodrošina Javascript skata izveidošanu no XML skata.
/// </summary>

using System;
using System.Collections.Generic;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Xml;
using System.Xml.Schema;
using System.Xml.XPath;

namespace XML_Tester
{
    public class JsView
    {
        public List<string> XmlNamespaces = new List<string>(new string[] { });
        public string ControllerName = string.Empty;
        public string CurrentFile = string.Empty;
        public int ControlsCount = 0;
        public int TotalControlsCount = 0;
        private string readerName = string.Empty;
        private string js = string.Empty;
        private string javascriptFirstLine = string.Empty;
        private string javascriptCurrentLine = string.Empty;
        private string qt = "\u0022";
        private List<string> objectNames = new List<string>(new string[] { });
        private List<string> objectXPaths = new List<string>(new string[] { });

        private List<string> libraries = new List<string>(new string[]
        {
            "sap.m", "sap.makit", "sap.me", "sap.ui.comp", "sap.ui.layout",
            "sap.ui.richtexteditor", "sap.ui.unified", "sap.ui.vbm", "sap.ushell", "sap.uxap",
            "sap.viz"
        });

        private List<string> events = new List<string>(new string[]
        {
            "afterInit", "afterRendering", "beforeExit", "beforeOpen",
            "beforeRendering",
            "change", "close", "itemSelect", "logout", "open",
            "press", "scroll", "search",
            "sectionClose", "sectionOpen", "sectionsReorder", "select"
        });

        private List<string> attributeValuesAsString = new List<string>(new
        string[]
        {
            "additionalText", "ariaDescription", "htmlText", "key", "name", "text",
            "title", "value"
        });

        private List<string> aggregationNodeNames = new List<string>(new string[]
        {
            "buttons", "columns", "content", "detailPages", "items", "masterPages",
            "nodes", "resources", "sections", "tabs", "tokens"
        });

        public JsView(string fileName)
        {
            this.CurrentFile = fileName;
        }
    }
}
```

```

//// <summary>
//// Izveido Javascript skatu no dotā XML koda.
//// </summary>
public string CreateJavascriptView(string xml)
{
    this.ControllerName = string.Empty;
    this.XmlNamespaces.Clear();
    this.ObjectNames.Clear();
    this.ObjectXPaths.Clear();

    try
    {
        XmlDocument xdoc = new XmlDocument();
        xdoc.LoadXml(xml);
        XmlNode root = xdoc.FirstChild;

        if (root.Name == "core:View" || root.Name == "mvc:View")
        {
            if (root.Attributes["controllerName"] != null)
            {
                this.ControllerName =
root.Attributes["controllerName"].Value;
            }
            else
            {
                return "// Javascript fragment";
            }

            if (root.Attributes.Count > 0) // Saknes atribūtu parsēšana.
            {
                for (int i = 0; i < root.Attributes.Count; i++)
                {
                    if (root.Attributes[i].Value != string.Empty &&
root.Attributes[i].Name.StartsWith("xmlns:"))
                    {
                        if (this.libraries.Contains(root.Attributes[i].Value))
                        {
                            this.XmlNamespaces.Add(root.Attributes[i].Value);
                        }
                    }
                }
            }

            this.js += this.ParseCurrentJsNode(root);
            if (!root.HasChildNodes)
            {
                this.js = string.Empty;
            }
        }
        catch
        {
            return "XML error";
        }

        this.javascriptFirstLine = "sap.ui.jsview(" + this.qt +
this.ControllerName + this.qt + ", { getControllerName : function() { return " +
this.qt +
                this.ControllerName + this.qt + "; }, createContent :
function(oController) {var oContent = []; ";
                this.js = this.javascriptFirstLine + this.js + " return oContent;});";
        return this.js;
    }

    /// <summary>
    /// Analizē doto XML mezglu.
    /// </summary>
    private string ParseCurrentJsNode(XmlNode currentNode)
    {

```

```

        string result = string.Empty;

        if (currentNode.NodeType == XmlNodeType.Comment || currentNode.NodeType
== XmlNodeType.Notiation || currentNode.NodeType == XmlNodeType.Whitespace)
        {
            return string.Empty;
        }

        if (currentNode.Name != "core:View" && currentNode.Name != "mvc:View")
        {
            this.ControlsCount++;
            if (!this.aggregationNodeNames.Contains(currentNode.LocalName))
            {
                this.TotalControlsCount++;
            }

            this.objectNames.Add("o" + currentNode.LocalName +
this.ControlsCount);
            this.objectXPaths.Add(this.GetXPathToNode(currentNode));
            // Veido objekta deficīciju Javascript kodā.
            result = "var o" + currentNode.LocalName + this.ControlsCount + " =
new " + currentNode.NamespaceURI + "." + currentNode.LocalName + "{";
            result = this.ParseNodeAttributes(currentNode, result);
        }

        if (this.aggregationNodeNames.Contains(currentNode.LocalName))
        {
            return this.ParseAggregationCode(currentNode);
        }

        if (currentNode.HasChildNodes)
        {
            for (int i = 0; i < currentNode.ChildNodes.Count; i++)
            {
                result += this.ParseCurrentJsNode(currentNode.ChildNodes[i]);
            }

            if (currentNode.Name != "core:View" && (currentNode.Name !=
"mvc:View"))
            {
                result += "});" + "oContent.push(o" + currentNode.LocalName +
(this.ControlsCount - 1).ToString() + ");";
            }
            else
            {
                if (currentNode.ParentNode.Name == "core:View" ||
currentNode.ParentNode.Name == "mvc:View")
                {
                    result += "});" + "oContent.push(" + "o" +
currentNode.LocalName + this.ControlsCount + ");";
                }
                else
                {
                    result += "});" + "o" + currentNode.ParentNode.LocalName +
(this.ControlsCount - this.GetNodePosition(currentNode)).ToString() +
".push(" + "o" + currentNode.LocalName + this.ControlsCount
+ ");";
                }
            }

            return result;
        }

        /// <summary>
        /// Analizē visus mezgla atribūtus un izveido gatavu koda fragmentu visiem
        atribūtiem uzreiz.
        /// </summary>
        private string ParseNodeAttributes(XmlNode currentNode, string result)
        {

```

```

        if (currentNode.Attributes.Count > 0)
        {
            for (int i = 0; i < currentNode.Attributes.Count; i++)
            {
                if (currentNode.Attributes[i].Name == "id")
                {
                    result = result.Remove(result.Length - 1);
                    result += "this.createId(" + this.qt +
currentNode.Attributes[i].Value + this.qt + "),{";
                }
                else
                {
                    if ((currentNode.Attributes[i].Value != string.Empty) &&
(currentNode.Attributes[i].Name != "xmlns"))
                    {
                        result +=
this.ParseAttribute(currentNode.Attributes[i].Name,
currentNode.Attributes[i].Value);
                    }
                }
            }
        }
        return result;
    }

    #region XML parsing
    /// <summary>
    /// Izveido koda fragmentu agregācijām. Agregācijas tiek realizētas kā
    apakšobjektu masīvs.
    /// </summary>
    private string ParseAggregationCode(XmlNode node)
    {
        string result = string.Empty;
        result = node.LocalName + ": [";
        for (int i = 0; i < node.ChildNodes.Count; i++)
        {
            result += "new " + node.ChildNodes[i].NamespaceURI + "." +
node.ChildNodes[i].LocalName + "({";
            result = this.ParseNodeAttributes(node.ChildNodes[i], result);
            this.TotalControlsCount++;
            if (node.ChildNodes[i].HasChildNodes)
            {
                if
(this.aggregationNodeNames.Contains(node.ChildNodes[i].LocalName))
                {
                    result += this.ParseAggregationCode(node.ChildNodes[i]);
                }
                else
                {
                    result += this.ParseCurrentJsSubNode(node.ChildNodes[i]);
                }
            }

            result += "}),";
        }

        result += "],";
        return result;
    }

    /// <summary>
    /// Analizē agregācijas, kas var saturēt apakšobjektus.
    /// </summary>
    private string ParseCurrentJsSubNode(XmlNode currentNode)
    {
        string result = string.Empty;
        this.ControlsCount--;
    }

```

```

    if (this.aggregationNodeNames.Contains(currentNode.LocalName))
    {
        return this.ParseAggregationCode(currentNode);
    }

    if (currentNode.HasChildNodes)
    {
        for (int i = 0; i < currentNode.ChildNodes.Count; i++)
        {
            result += this.ParseCurrentJsNode(currentNode.ChildNodes[i]);
        }
    }

    return result;
}

/// <summary>
/// Analizē XML mezgla atribūtu un izveido tam atbilstošu Javascript koda
fragmentu.
/// </summary>
private string ParseAttribute(string name, string value)
{
    string result = string.Empty;

    if (value.Contains("formatter:") || value.Contains("path:"))
    {
        return this.ParseFormatterValue(name, value);
    }

    if (this.IsAttrValueInQuotes(name, value))
    {
        if (this.events.Contains(name))
        {
            result += name + " : " + "oController." + value + ",";
        }
        else
        {
            result += name + " : " + this.qt + value + this.qt + ",";
        }
    }
    else
    {
        if (this.events.Contains(name))
        {
            result += name + " : " + "oController." + value + ",";
        }
        else
        {
            result += name + " : " + value + ",";
        }
    }

    return result;
}

#endregion
}
}

```

Kvalifikācijas darbs „*SAPUI5 skatu veikspējas analīzes rīks*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Romāns Kolduns* _____ .06.2015.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs/a: *dipl. mat., Imants Pops* _____ .06.2015.

Recenzents: *M. dat., Dace Gobleja*

Darbs iesniegts 01.06.2015.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2015. prot. Nr. _____

Komisijas sekretārs(-e): _____