

**Web-sovelluksen rakentaminen ja elinkaaren hallinta
OutSystemsissa**



Hämeen ammattikorkeakoulu

Tietojenkäsittelyn koulutusohjelma

Visamäki syksy 2017

Pekka Salo

Tietojenkäsittelyn koulutusohjelma
Hämeenlinna

Tekijä	Pekka Salo	Vuosi 2017
Työn nimi OutSystemsissa	Web-sovelluksen rakentaminen ja elinkaaren hallinta	
Työn ohjaaja	Tommi Lahti	

TIIVISTELMÄ

Opinnäytetyössä tutkittiin web-applikaation elinkaaren hallintaa, aina sen ideoinnista eläköitymiseen asti. Käytännöntyön tarkoituksena oli rakentaa yksinkertainen, helposti hallittava ja jatkokehitettävä applikaatio Hämeen ammattikorkeakoululle käyttäen OutSystems-työkalua. Opinnäytetyön toimeksiantajana oli Mika Rintala, Hämeen ammattikorkeakoulun tietohallinnon yksiköstä.

Kehitettävän web-applikaation tavoitteena oli rakentaa portaali yrityksiä ja yhteistyökumppaneita varten, johon he voivat ilmoittaa mahdollisista harjoittelu-, työ- ja opinnäytetyöpaikoista opiskelijoille. Applikaatio piti olla myös helposti integroitavissa muihin jo olemassa oleviin järjestelmiin. Lisäksi pohdittiin mitä applikaation elinkaaren hallinta sisältää, mitä eri tekniikoita nykyapplikaation kehityksessä tulisi ottaa huomioon ja miten applikaatiota tulee hallita myös sen käyttöönoton jälkeen.

Teoriaosuuteen kerättiin lähteitä kirjallisuudesta ja esimerkiksi OutSystemsin opetusmateriaaleista. Käytännön osuudessa esiteltiin OutSystemsin kehitystyökalu Service Studio. Lisäksi kehitettiin, julkaistiin ja hallinnoitiin HamkHub – applikaatiota.

Avainsanat Nopean kehityksen kehitys, applikaation elinkaari, OutSystems.

Sivut 28 sivua

Degree Programme in Business Information Technology
 Hämeenlinna

Author	Pekka Salo	Year 2017
Subject	Web Application Building and Lifecycle Management with OutSystems	
Supervisor	Tommi Lahti	

Abstract

This thesis researched the control of the Web -applications lifecycle, all the way from the idea down to the retirement period. The commissioner of this thesis was Mika Rintala from the Information management of Häme University of Applied Sciences.

The objective of the developed application is to build Internet portal for companies and partners where they can advertise internship , work and thesis-opportunities for the students of HAMK. The Application should be easily integrated for existing applications and software. In addition, it was contemplated what application lifecycle includes, what different techniques one should consider when developing application and how applications should be controlled after it has been deployed.

The theoretical information was collected from literature and from the Internet, such as OutSystems training material. In the practical part, the thesis introduces Service Studio, the developing tool of OutSystems Platforms. The HamkHub -application was developed, published and managed with it.

Keywords Rapid Application Development, lifecycle, OutSystems.

Pages 28 pages

SISÄLLYS

1	JOHDANTO.....	4
2	APPLIKAATION NOPEAN KEHITYKSEN MALLI JA LOW-CODE ALUSTAT	5
2.1	Low-code alustat	6
2.2	OutSystems	6
3	OUTSYSTEMS PLATFORMS	8
3.1	Platform Server	9
3.2	Service Studio	11
3.3	One-Click-Publish	12
3.4	Henkilökohtainen ympäristö.....	12
3.5	Mobiiliapplikaatio OutSystemsissä	13
3.6	Webapplikaatio OutSystemsissä	13
3.7	The Forge.....	14
4	APPLIKAATION ELINKAARI JA SEN HALLINTA	15
4.1	OutSystemsin tuki applikaation elinkaaren hallinnassa.....	16
4.1.1	Deployment	16
4.1.2	Operations and analytics.....	17
4.1.3	Management	18
5	APPLIKAATION KEHITYS OUTSYSTEMSIÄ KÄYTTÄEN.....	19
5.1	Service Studio	19
5.2	OutSystemsin dataobjektit.....	20
5.3	Moduulit.....	21
5.4	Applikaation luominen	22
5.5	Datan näyttäminen sivuilla	24
6	APPLIKAATION JULKAISU JA JATKOKEHITYS	26
7	YHTEENVETO	27
	Lähteet.....	28

1 JOHDANTO

Tässä opinnäytetyössä tutustutaan applikaatioiden kehitykseen, elinkaareen sekä kehitetään responsiivinen web-palvelu käyttäen OutSystems järjestelmää. Lisäksi avataan yleisiä applikaatioiden kehityksessä käytettäviä käsitteitä ja työkaluja kuten REST-rajapintaa, RAD-työkalujen hyötyjä sekä sitä, miten OutSystems helpottaa applikaatioiden kehitysprosessia perinteisiin malleihin verrattuna.

Opinnäytetyön tekijällä on aikaisempaa kokemusta niin mobiili- kuin web-sovellusten kehittämisestä useilla eri tavoilla, kuten Android Studiolla, Visual Studion .net-ympäristöllä sekä Xamarin-laajenuksella. Tekijän motivaationa toimii oma kiinnostus mobiiliapplikaatioiden sekä responsiivisten web-applikaatioiden luomista kohtaan, halu opetella käyttämään OutSystems-järjestelmää sekä halu oppia applikaation julkaisu kokonaisuudessaan. Aikaisempaa kokemusta applikaation julkaisusta tai päivittämisestä tekijällä ei ole.

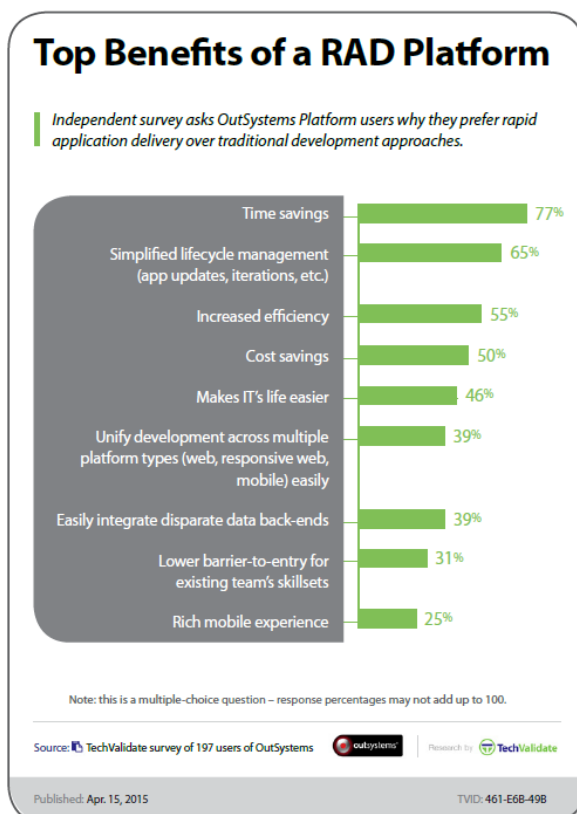
Työn lähtökohtana on muiden ohjelmointikielien ja tietokantojen perustaito ja ymmärrys. Käytännönvaiheessa käydään läpi sovelluksen kehitys ja julkaiseminen käyttäen OutSystemsiä. Osuudessa ei tulla käymään läpi julkaisun kannalta vähemmän merkittäviä аспектеja, kuten käytettävyyttä tai oman ulkoasun luomista. Opinnäytetyön lopussa kehitetty sovellus myös päivitetään.

Opinnäytetyön tutkimuskysymyksiä ovat: Mitä ovat RAD-työkalut? Miten OutSystems voi helpottaa kehittäjän työtä? Mitä kuuluu applikaation elinkaareen ja sen hallitsemiseen?

2 APPLIKAATION NOPEAN KEHITYKSEN MALLI JA LOW-CODE ALUSTAT

RAD (Rapid Application Delivery) pyrkii erityisesti nopeuttamaan applikaatioiden kehitysprosessia ja saamaan aikaan toimivia järjestelmiä lyhyessä ajassa. Tämän ohjelmistokehitysmallin kehitti James Martin kollegoineen vuonna 1991. (Naz & Khan, 2015).

Normaalin applikaation kehityksessä kehittäjät kirjoittavat kaiken koodin manuaalisesti, kun taas RAD-ohjelmistojen tarkoituksena on minimoida se. RAD-mallissa kehittäjät luovat visuaalisia malleja applikaation osista, kuten integraatioista tai tietokannoista, ja ohjelmisto luo taustalla kaiken tarvittavan koodin ja toiminnallisuuden. Tämä myös pienentää vaatimuksia siitä, mitä kehittäjiltä vaaditaan: samanlaisiin lopputuloksiin päästään myös niiden toimesta, jotka eivät ole kovin kauan ohjelmoinnin parissa työskennelleet. Alla on esitetty RAD:in edut OutSystemsin käyttäjien mielestä (Kuva 1). (Outsystems, n.d.a.)



Kuva 1. OutSystemsin käyttäjien eniten arvostamat ominaisuudet.

Yleisesti ajateltuna RAD sisältää neljä seuraavaa vaihetta: suunnitteluvaihe, käyttäjän suunnitteluvaihe, rakennusvaihe ja leikkausvaihe. RAD:issa tarkoitus on edelleen säilyttää tuotteen korkea laatutaso, mutta kulut ja kehitykseen käytetty aika minimoidaan. Tavoitteeseen pääsyyn auttaa tarkka ja laadukas suunnittelu. (Naz&Khan,2015.)

2.1 Low-code alustat

Low-code-alustoilla tarkoitetaan kehitysympäristöä, jossa RAD-ohjelmistokehitysmallia pyritään toteuttamaan. Toinen yleinen nimi niille onkin rapid application development platforms. Low-code-kehitysalustat tarjoavat mahdollisuuden mm. click and drag -ohjelmointiin, sisäänrakennettujen työkalujen sekä kehittämistä nopeuttavien ja helpottavien palveluiden käyttöön. Yleisesti kyseiset ohjelmat pyörivät web -selaimen avulla. (Ciot, 2016.)

Jokaisessa kehitettävässä sovelluksessa on tietyt samat elementit, jotka täytyy saada toimimaan. Esimerkiksi jokaisessa sovelluksessa sisäinen navigointi tai tietokantayhteydet täytyy toteuttaa jollain tapaa. Low-code-alustat pyrkivät ratkaisemaan nämä ongelmat kehittäjän puolesta, jotta kehittäjä voi itse keskittyä sovelluksen yksityiskohtaisiin ja uniikkeihin toimintoihin. Low-code-alusta sisältää yleensä taulukossa 3 esiteltyt ominaisuudet. (OutSystems, 2017.)

Taulukko 1. Low-code alustojen yleiset toiminnallisuudet. (OutSystems, 2017.)

Visuaalinen IDE:n	Kehitysympäristön visuaalinen osa, jossa pystytään mm. suunnittelemaan UI tai liittämään toiminnallisuudet toisiinsa.
Työkalu back end -yhteyksien hallinnoimiseen	Alusta pyrkii luomaan tarvittavat tietokantayhteydet ja niiden hallitsemiseen tarvittavat komponentit valmiiksi, ja käyttäjän ei tarvitse kuin asettaa yhteyksissä käytettävät parametrit.
Applikaation elinkaaren hallintatyökalut	Esimerkiksi koodin debuggaus-työkalu

2.2 OutSystems

Mobiili- ja web -applikaatioiden kehittäminen tunnetaan todella aikaa vievinä prosesseina, jotka vaativat tuntemusta monelta eri alalta ja useasta eri tekniikasta. Siinä täytyy onnistua yhdistämään muun muassa hyvä suunnittelu, kehitys ja markkinointi. (Rowles 2014, 119-121.) Lisäksi tarvitsee hallita julkaisuun, ohjelman päivittämiseen ja muuhun yleiseen hallintaan liittyvät tekniset asiat. OutSystems pyrkii ratkaisemaan kaikki nämä ongelmat yhdessä paketissa.

OutSystems on yrityksille suunnattu RAD-alusta, joka on nopea ja kattava kehitysympäristö, jolla voidaan kehittää, julkaista ja hallinnoida sekä natiiveja mobiili-että web -applikaatioita. Se on saatavilla PaaS (Platform as a Service) ratkaisuna, yritykselle asennettavana ratkaisuna tai näiden välisenä hybrid-ratkaisuna.

OutSystems Platformin tarkoituksena on olla kokonaisvaltainen työkalu, joka nopeuttaa applikaation kehityksen jokaista kehitysvaihetta markkinoiden vaativan nopeuden mukaan. Se pitää sisällään niin front - ja back -endin , tietokantayhteydet sekä muut isojenkin projektien tarvitsemat työkalut. Tämä saattaa säästää

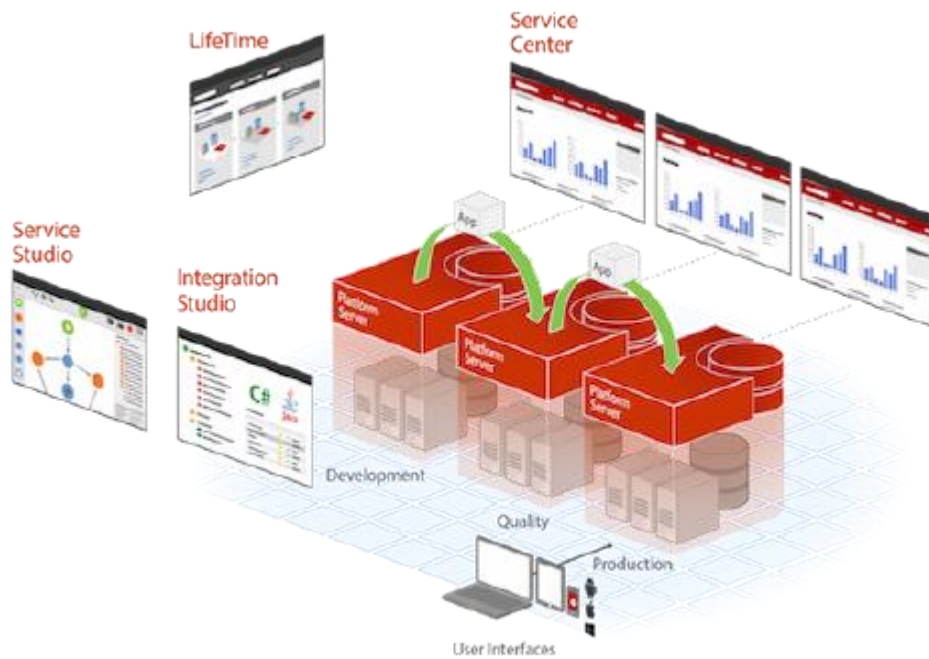
huomattaviakin summia sekä lisätä tuottavuutta, koska palveluita ei tarvitse ostaa eri tahoilta.

Ohjelman kaikki ominaisuudet pyrkivät edistämään mahdollisimman nopeaa applikaatioiden toimittamista. Kehittäjät luovat sovelluksiaan lähinnä visuaalisen käyttöliittymän kautta, joten itse koodin kirjoittamiseen menee mahdollisimman vähän aikaa.

3 OUTSYSTEMS PLATFORMS

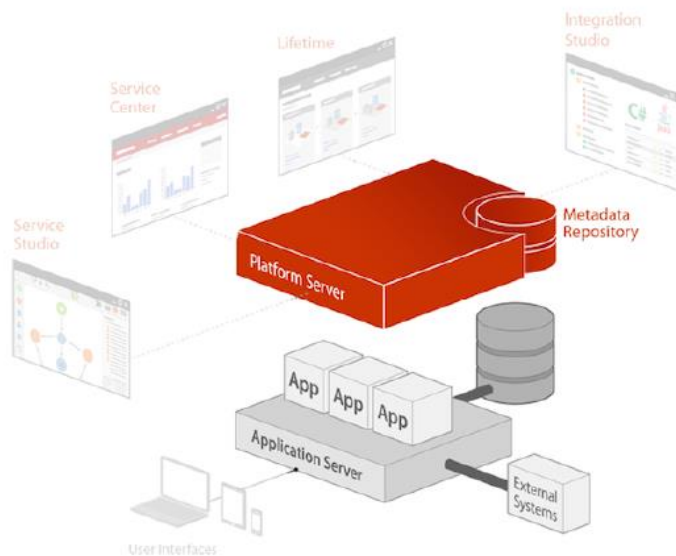
OutSystems Platforms (kuva 2) sisältää monia erilaisia ympäristöjä ja työkaluja, joilla pyritään sisällyttämään kaikki applikaation kehitykseen ja elinkaareen tarvittavat prosessit. Normaalin web-applikaation taustalla pyörii lukuisia toisiaan tukevia tekniikoita, jotka mahdollistavat monipuolisen kehityksen. Kehittäjän käytettävissä on muun muassa Microsoft .NET stack, Oracle WebLogic ja JBOSS. Lähdekoodi generoidaan joko C#- tai Java -ohjelmointikieliin.

OutSystems Platform tarjoaa työkaluja mm. käyttöliittymien, tietokantojen, SOAP- ja REST -web-palveluiden, ajastettujen toimintojen sekä monenlaisten integraatiomahdollisuuksien käyttämiseen sovelluksissa. Kaikkiin näihin työkaluihin pääsee käsiksi käyttäen OutSystemsin visuaalista käyttöliittymää, OutSystems Serviceä. (OutSystems. n.d.d.)



Kuva 2. OutSystems Platform kokonaisuutena. (OutSystems. n.d.d.)

3.1 Platform Server



Kuva 3. Platform Server. (OutSystems, n.d.i.)

Platform Serverillä (kuva 3) on tietty määrä servereitä, jotka hoitavat applikaatioiden rakentamisen, kokoamisen, ajamisen ja monitoroinnin. Ne on jaoteltu kolmeen tyyppiin servereihin: Front-End -serverit, käyttöönoton kontrolliserverit ja tietokanta serverit. (OutSystems, n.d.i.)

Front-End -serveri on standardisoitu IIS- tai Java-applikaatio -serveri, jota on laajennettu kolmella ylimääräisellä OutSystemsin omalla serverillä. Jokainen serveri voi sisältää useita eri front end -palveluita. Jokainen Front -end -serveri sisältää seuraavat komponentit:

Taulukko 2. Front end -serverin komponentit. (OutSystems, n.d.i.)

The Web Application Server (WAS)	Jokainen Front -End Service käyttää omaa WAS:ia pyörittämiseensä. .NET tapauksissa WAS:ina toimii Windowsin IIS, ja Javan tapauksissa WAS:ina toimii WebLogic tai JBOSS.
Deployment Service	Toimii läheisessä yhteistyössä Deployment Controller Servicen kanssa. Yhdessä ne varmistavat, että applikaatiot varmasti asentuvat ja käyttöönotetaan onnistuneesti Front -End Serverillä.
Scheduler Service	Service Studion kautta on mahdollista mallintaa erilaisia ajastettuja toimintoja ja aktiviteetteja, jotka Scheduler Service suorittaa
Log Service	Mahdollisten virheiden ja ongelmatilanteiden kannalta jokaisessa palvelussa on tärkeitä logittaa tieto mistä, milloin ja mahdollisesti miksi virhe on tapahtunut. OutSystemsissä tiedot tallentuvat Log Serviceen.

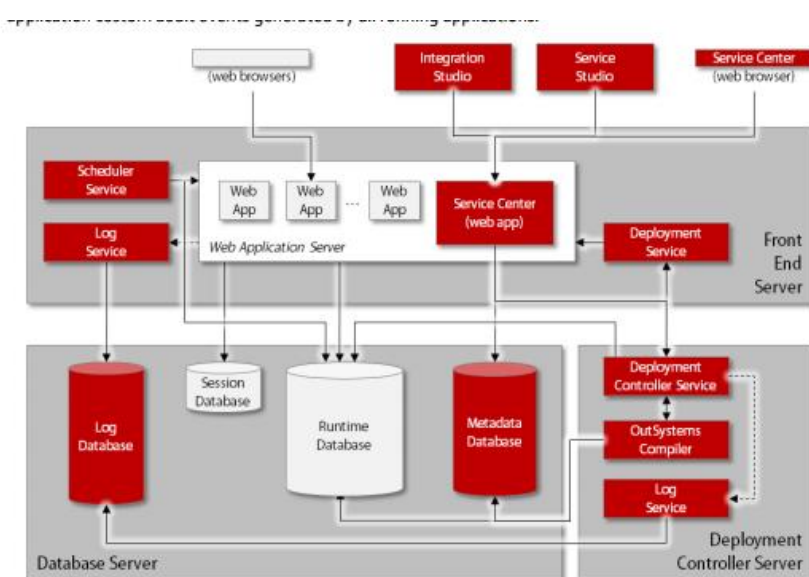


Figure 6. The Platform Server components and services are clustered in 3 types of "servers": the front-end servers, the deployment controller server and the database server. Each of these server clusters can be installed across multiple physical or virtual machines in a data center or cloud enabled environment.

Kuva 4. Platform Serverin komponentit ovat jaettu kolmeen eri palveluun, jotka keskustelevat keskenään. (OutSystems, n.d.i.)

Deployment Controller Server on vastuussa kaikesta rakentamisesta Front End -servereille. Jokaisella Platform Serverillä on vain yksi Controller Server, mutta se sisältää kaksi erilaista sisäistä komponenttia, OutSystems Compiler ja Deployment Controller Service (kuva 4). Compiler rakentaa applikaation lähdekoodin joko C#-tai Java-kielelle applikaation mallien mukaan. Lisäksi se luo tarvittavat tietokantascriptit mallien mukaan. Deployment Controller Service on enemmänkin yleinen koordinaattori OutSystems Controllerin ja muiden Front-Endin komponenttien kanssa. (OutSystems, n.d.i.)

Database Server jakautuu kolmeen eri osaan, joista jokaisella on oma tarkoituksensa:

Taulukko 3. Database Serverin osiot. (OutSystems, n.d.i.)

Runtime	Tänne varastoidaan kaikki applikaation data, kuten esimerkiksi käyttäjätiedot
Metadata	Metadataan tallennetaan esimerkiksi OutSystem Platformin applikaatioversiot, Platformin käyttäjät ja turvallisuusasetukset
Sessiot	Sessioihin tallennetaan käyttäjän sessiotiedot

Service Center: Service Center (kuva 5) on web-applikaatio, joka pystyy toimimaan Platform Serverin työkalujen kanssa. Sitä käytetään hallinnoimaan sovellukselle

vaadittavia asioita ja tarkastelemaan sen toimintaa sekä mahdollisia virhetilanteita. (OutSystems, n.d.i.)

The screenshot shows the Service Center monitoring dashboard. The top navigation bar includes 'Factory', 'Monitoring', 'Administration', and 'Analytics'. The 'Monitoring' section is active, showing 'Development' as the selected environment. The main content area is divided into several sections:

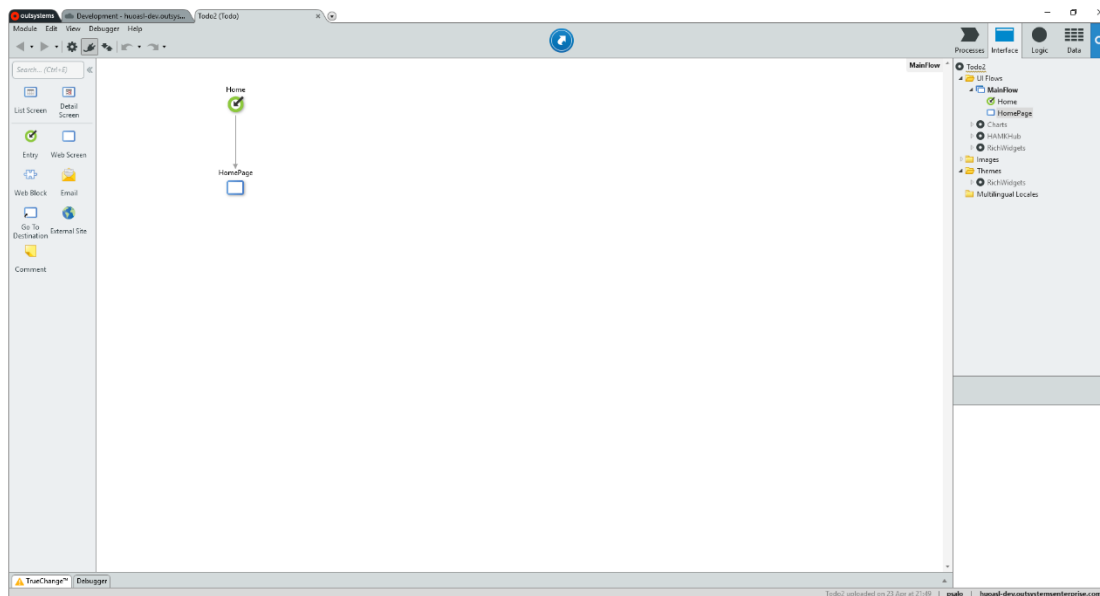
- Environment Health:** Displays the current status of the platform server. A green status indicator is shown.
- Front-end Servers:** Displays the IP address of the Front-end Servers and the status of the following services: Deployment Controller service, SMS Gateway, and Timers activity. A table lists the server 'jwdemo1dev (Controller)' with IP address '127.0.0.1' and green status indicators for all services.
- Processes:** Displays the current number of active process instances, suspended process instances, queued process flow activities, and queued process instances. A table shows 0 active, 0 suspended, 0 queued, and 0 with errors for all categories.
- Timers:** Lists which timers will be executed next, ordered by their Next Run time. A table lists various timers such as 'RetrievePageTimer', 'ErrorLogMonitor', and 'CheckSystemStatus' with their last run and next run times.
- LifeTime:** Shows the current version of the application (7.0.0.51) and the development mode.
- Recent items:** Lists recent items, including 'ServiceCenter eSpace' and 'Independent Modules Application'.

Kuva 5. Yleiskuva Service center:ista. (OutSystems, n.d.i.)

3.2 Service Studio

OutSystems Platformin visuaalinen kehitysympäristö on nimeltään OutSystems Service Studio (kuva 6). Service Studiota käytetään sekä mobiili- että web- applikaatioiden kehityksessä.

Jotta Service Studiolla voidaan aloittaa applikaation kehittäminen, täytyy ensin kirjautua kehitysympäristöön. Kehitysympäristöstä päästään hallinnoimaan kaikkia serverille luotuja applikaatioita, luomaan uusia, tai asentamaan Forgesta löytyviä muiden käyttäjien kehittämiä applikaatioita. (OutSystems. n.d.e.)



Kuva 6. Uuden moduulin Service Studio aloitusnäky. (OutSystems. n.d.e.)

3.3 One-Click-Publish



Kuva 7. One-click-publish.

Applikaatio julkaistaan OutSystems-serverille käyttäen one-click-publish-painiketta (kuva 7). Mahdolliset ympäristön antamat virheilmoitukset tosin estävät julkaisemisen. Julkaistaessa applikaatiota serverille, Service Studio lähettää kehitetyn applikaation Platform Serverille, joka kokoaa ja optimoi koodin, UI:n ja kaikki applikaation toimivuuteen tarvittavat ominaisuudet. Tämän jälkeen one-click-publish-painike muuttuu linkiksi kehitettyyn sivustoon, jossa applikaatiota voidaan testata. (OutSystems. n.d.e.)

3.4 Henkilökohtainen ympäristö

OutSystems tarjoaa ilmaisen, pilvipohjaisen version OutSystems Platform -kehitystyökalusta. Tämän palvelun nimi on OutSystems Personal environment. Henkilökohtaisella ympäristöllä pystyy kehittämään applikaatioita niin kauan kun kehittäjät ja loppukäyttäjät pysyvät aktiivisina. Jos tilistä tulee epäaktiivinen, ympäristö uudelleenkierrätetään seuraavalle käyttäjälle. Näin säästetään resursseja muille kehittäjille. OutSystems kuitenkin säilyttää kopion siitä tilasta, mihin ympäristö on jäänyt, ja se on mahdollista palauttaa tietyn ajan kuluessa.

Vaikka henkilökohtaisella ympäristöllä voi luoda täysin vastaavanlaisia mobiili- ja web-applikaatioita kuin OutSystems Platformilla, se on suunniteltu toimivaksi pienille projekteille. Skaalautuvuuden, rasituksen ja monien muiden tekijöiden johdosta yritysten on suotavaa käyttää niille suunnattua Enterprise -lisenssiä. (OutSystems. n.d.c.)

3.5 Mobiiliapplikaatio OutSystemsissä

Mobiiliapplikaation kehitys ja toiminta ovat OutSystemsissä toiminnallisuuksiltaan hyvinkin samanlaisia kuin missä tahansa muussa natiivin mobiilisovelluksen kehitystyökalussa. Ohjelma ladataan ja käytetään itse puhelimessa, joka mahdollistaa mm. lokaalin datantallennuksen tai laitteen muiden ominaisuuksien, kuten sormenjälkitunnistuksen käytön (kuva 8). (OutSystems. n.d.b.)



Kuva 8. Mobiili -applikaatio OutSystemsissä. (OutSystems, n.d.b.)

3.6 Webapplikaatio OutSystemsissä

OutSystemsissä web -applikaatio (kuva 9) on responsiivinen, selaimessa toimiva applikaatio, joka pystyy mukautumaan jokaisen käyttävän laitteen ominaisuuksien ja näytön koon mukaan. Normaalin internet web -sivun lailla käyttäjän ei tarvitse asentaa mitään erikseen laitteellensa, vaan applikaatioita käytetään internet-selaimen kautta. Applikaatio nojaa suuresti internet-yhteyden varaan, eikä applikaatiota pysty käyttämään offline-tilassa ilman kolmannen osapuolen komponentteja. (OutSystems. n.d.b.)



Kuva 9. Web-applikaatio OutSystemsissä. (OutSystems. n.d.b.)

3.7 The Forge

Forge (kuva 10) on OutSystemsin oma moduulivarasto, jossa käyttäjät voivat jakaa omia kehittämäänsä OutSystems-komponentteja ja kirjastoja muiden käytettäväksi. Komponentit voivat olla joko yksittäisiä toiminnallisia ratkaisuja, kuten tietynlaisia käyttöliittymien pohjia tai jopa kokonaisia applikaatiota. Kaikki Forgessa sijaitsevat projektit ovat open source -lisensoitu BSD-mallin mukaan. Kaikkia BSD:n alla olevia ohjelmistoja voidaan käyttää ja muokata käyttäjän halujen mukaan. Forgen tarkoituksena on nopeuttaa entisestään applikaatioiden kehitystä: käyttäjien ei tarvitse käyttää aikaa ongelmien ja logiikkaosuuden parissa, joihin on jo olemassa muiden tahojen kehittämä ratkaisu. (OutSystems. n.d.d.)

The screenshot displays the OutSystems Forge website. At the top, there is a navigation bar with the OutSystems logo and links for PLATFORM, LEARN, COMMUNITY, and SUPPORT. Below this, there are tabs for FORUMS, FORGE, and IDEAS. The main heading is "Forge", followed by a description: "OutSystems Forge is a repository of reusable application modules, connectors and components that speed application delivery." There are two buttons: "PUBLISH NEW PROJECT" and "FORGE FAQs". A featured project "Directory" is highlighted with a red icon and a description: "A full-featured responsive application that allows you to customize your employee contact details." Below this is a search bar labeled "Search Open Source Projects" and a "Sort by: Most Popular" dropdown. A grid of project cards is shown, including "Google Maps" (published by OutSystems Labs, 1745 stars), "Excel Report" (published by Miguel Mairnes, 221 stars), "Paypal connector" (published by Gonzalo Borrego), and "Oracle Connector P9" (published by Grazina). A "CATEGORIES" sidebar on the right lists various categories like Featured, Applications, Components, Widgets, Connectors, Reporting, Libraries, Themes, Plug-ins, and How-tos.

Kuva 10. The Forge (OutSystems. n.d.d.)

4 APPLIKAATION ELINKAARI JA SEN HALLINTA

Applikaation elinkaaren määrittäminen ei ole helppoa, koska eri tahot määrittelevät sen eri näkökulmista. On kuitenkin tärkeää muistaa, että puhuttaessa applikaation elinkaaresta, sillä tarkoitetaan sen koko olemassaoloa, ei vain aikaa mitä on käytetty sen kehitykseen. Myös kaikki julkaisun jälkeen tapahtuvat monitoroinnit, päivitykset ja muu toiminta sisältyvät applikaation elinkaareen.

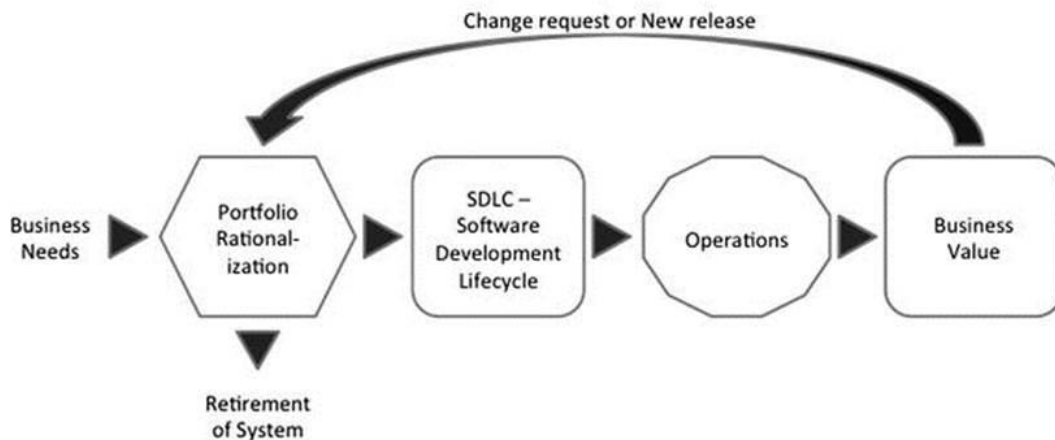
Applikaation elinkaari voidaan jakaa kolmeen eri aikajaksoon: ideaan, käyttöönottoon ja sen elinkaaren päätökseen. Näiden kolmen ajanjakson aikana tapahtuu kolme eripituista applikaation kehitykseen liittyvää toimintajaksoa: hallintotapa, kehitys ja toiminnot. Applikaation hallintotapaan kuuluu kaikki päätökset ja projektin hallintamuodot, jotka kestävät koko applikaation elinkaaren ajan. Kehitys on itse applikaation kehittämistä, ja sitä tapahtuu myös käyttöönoton jälkeen, sisältäen päivityksiä tai jopa täysin uusia versioita tuotteesta. Vähän ennen applikaation käyttöönottoa alkavat operaatiot, jotka sisältävät itse käyttämisen hallintaa, kuten asiakaspalvelua ja tuotteen viemistä maailmalle. Operaatiot kestävät aina applikaation käytön loppuun asti. (Chapell D, 2008, 2.)

Applikaation elinkaari ja sen hallinta (myöhemmin AEH) on lisääntyvässä määrin kasvattanut kiinnostustaan kehitys yhteisöissä. On todettu, että hyvällä applikaation elinkaaren hallintaprosessilla voi nopeuttaa kehitystä, pienentää kustannuksia ja nostaa laatutasoa todella suurissa määrin. (Rosseberg J. 2013. Chapter 1, 1.)

Kommunikaation muutos, globalisoituminen, älypuhelimet ja internet ovat muuttaneet suuresti kaupankäyntiä kaikkialla. Asiakaskunnat ovat laajempia, kaikki informaatio on helposti ja välittömästi tavoitettavissa ja tarve uusille ratkaisuille on kasvava. Tämä vuoksi yrityksen on oltava valmiina kohtaamaan uudet haasteet nopeallakin varoitusajalla. Jotta voidaan vastata kasvavaan ja alati muuttuvaan kysyntään, tulee AEH:sta todella tärkeä osa yrityksen kehityksessä. (Rosseberg J. 2013. Chapter 1, 2.)

Rosseberg määrittää (Kuva 11) AEH-prosessin sen syntymästä aina sen eläköitymiseen asti. Kehää voidaan kiertää niin kauan kuin se bisnesarvon kannalta on järkevää. Koska kysyntä kasvaa ja muuttuu, täytyy yrityksen olla varma siitä, että heidän applikaatio varmasti pystyy tuottamaan ja suorittamaan kaikki tarvittavat toimet.

ALM Process & Roles



Kuva 11. Rossebergin esittämä kaavio applikaation hallinnan elämänkaarelle.

Applikaation elinkaaren hallinnan prosessiin kuuluu useita erilaisia rooleja, riippuen tietysti applikaatiosta ja siitä, mitä palveluita sen kuuluu tarjota. Yleisesti applikaatiolla on kuitenkin rahoittajat, projektin omistaja ja / tai Scrum master, erilaiset kehitystiimit vastuualueineen, testaajat ja lopuksi operaatioiden ja asiakkaiden kanssa työskentelevät tiimit. (Rosseberg J. 2013. Chapter 2, 2.)

4.1 OutSystemsin tuki applikaation elinkaaren hallinnassa

OutSystems jakaa applikaation elinkaaren hallinnan omassa tuotteessaan seuraaviin kolmeen osioon: Deployment, Operations and Analytics ja Management. Näiden kolmen osa-alueen kautta on tarkoitus tarjota täydellinen tuki applikaatiolle, aina sen kehityksestä eläköitymiseen saakka. OutSystemsin yhtenä pääideana on applikaatioiden mahdollisimman nopea ja helppo kehitys. Tietokantayhteydet, back-endin optimoiminen ja muut operaatiot luodaan automaattisesti niin uusiin kuin olemassa oleviin kantoihin. (OutSystems, n.d.g.)

4.1.1 Deployment

OutSystemsissä voi siirtää koko kehityksessä olevan projektin tuotantoon yhdellä klikkauksella, "one-click-publish". OutSystems huolehtii riippuvuuksien tarkastamisesta, versionhallinnan ja muut tarvittavat toimenpiteet automaattisesti. OutSystems tarjoaa kehittäjälle kolme erilaista tilaa, applikaation eri versioille. Täten uuden version kehitystä voidaan hyvin jatkaa, kun asiakkailla käytössä oleva versio pysyy muuttumattomana. Kun kehittäjä yrittää siirtää tuotetta seuraavaan vaiheeseen, esimerkiksi tuotantoon, OutSystems tekee automaattisesti kaikki vaadittavat tarkistukset, jotta tuotantoon viety versio varmasti toimii, eikä esimerkiksi riippuvuuksia ole jäänyt matkan varrelle. Vaihteita voidaan hallita sille tarkoitettulla työkalulla (kuva 12). (OutSystems, n.d.h.)

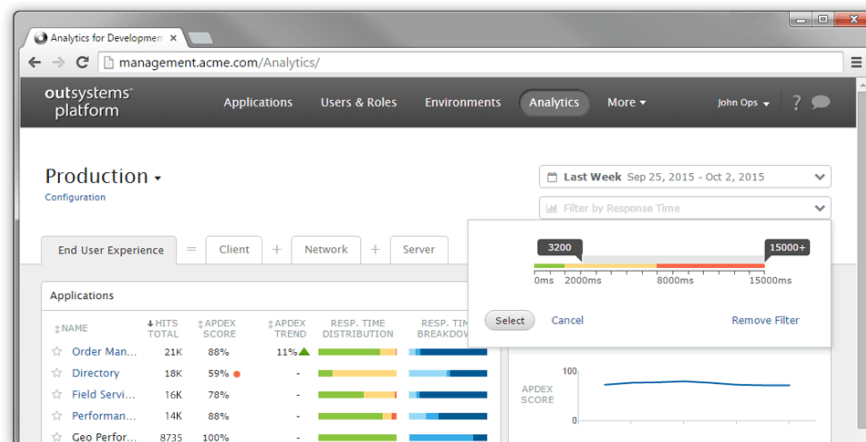
	Development	Quality Assurance	Production
Codes <small>Customer Portal team</small>	0,4 <small>Today, 09:47 Administrator</small>	0,4 <small>Today, 09:05 Camilla Malmelin</small>	
CRM Services <small>Customer Portal team</small>	1,3 <small>Feb 12, 17:26 Administrator</small>	1,3 <small>Today, 09:08 Camilla Malmelin</small>	
Customer Portal <small>Customer Portal team</small>	0,5 <small>Today, 09:51 Administrator</small>		
Directory <small>Directory team</small>	0,3 <small>Feb 10, 11:27 Administrator</small>	0,3 <small>Feb 10, 11:40 Administrator</small>	0,2 <small>Feb 10, 11:11 Camilla Malmelin</small>
Documents	0,1 <small>Feb 2, 15:23 António Pereira</small>	0,1 <small>Feb 10, 12:45 Audrey Tai</small>	
eCommerce <small>Dev eCommerce</small>	0,4+ <small>Feb 12, 16:42 Administrator</small>	0,4+ <small>Feb 10, 17:31 Administrator</small>	0,1 <small>Feb 10, 11:11 Camilla Malmelin</small>
Expenses	0,3 <small>Feb 12, 09:52 Administrator</small>	0,3 <small>Today, 09:51 Camilla Malmelin</small>	
Intranet Services	0,1 <small>Feb 2, 15:23 António Pereira</small>	0,1 <small>Feb 6, 18:27 Audrey Tai</small>	0,1 <small>Feb 10, 11:11 Camilla Malmelin</small>
JQuery Plugins	0,2 <small>Feb 4, 17:51 Timo Peromies</small>	0,2 <small>Feb 10, 12:47 Audrey Tai</small>	

Kuva 12. Applikaation eri vaiheiden hallintatyökalu OutSystemssissä. (OutSystems, n.d.h.)

4.1.2 Operations and analytics

Operaatioita ja analytiikkaa varten OutSystems tarjoaa oman hallintatyökalun (kuva 13), mutta pyrkii myös mahdollistamaan ulkoisten palveluiden helpon integraation mahdollisimman laajan ja laadukkaan analytiikan tarjoamiseksi. OutSystemsin oma työkalu monitoroi mm. tietokantakutsujen kestoa, integraatioiden toimivuutta ja kuvan piirtämiseen käytettyä aikaa varastoiden kaiken näistä saadun datan logitiedostoihin.

Lisäksi infrastruktuurin hallintatyökalu on jaettu asiakas-, tietoliikenne- ja serverilohkoihin, jotta hitauksien löytäminen on mahdollisimman helppoa ja nopeaa. Esimerkiksi tietoliikenteen lohkoista voidaan erikseen nähdä ja analysoida sovelluksen toimivuutta, kun loppukäyttäjä on yhteydessä 3G-verkkoon 4G verkon tai Wlanin sijaan. Samalla nähdään, olisiko mahdollista optimoida hitaampien yhteyksien toimivuuksia. (OutSystems. n.d.i.)



Kuva 13. Operaatioiden ja analytiikan hallintatyökalu. (OutSystems. n.d.j.)

4.1.3 Management

OutSystems tarjoaa todella laajan pohjan applikaation hallinnalle, kuten täyden tuen olemassa oleville tietokantayhteyksille ja tuotteen versioinnille. Applikaatiolle on myös mahdollista määrittää erilaisia ajastettuja toimintoja, jotka suorittavat niille määritetyt tehtävät itsenäisesti niille määritettyinä ajanjaksoina. Kaikki tämä hallinta tapahtuu OutSystemsin hallintakonsolista. (OutSystems. n.d.k.)

5 APPLIKAATION KEHITYS OUTSYSTEMSIÄ KÄYTTÄEN

Käytännön työn tarkoituksena on rakentaa responsiivinen internet-portaali, jonne yritykset voivat rekisteröityä ja ilmoittaa työ-, harjoittelu- ja opinnäytetyöpaikoista opiskelijoille. Portaalin työstenimi on HAMKHub. HAMKHubin suunnitellut ominaisuudet sekä näkymät ovat:

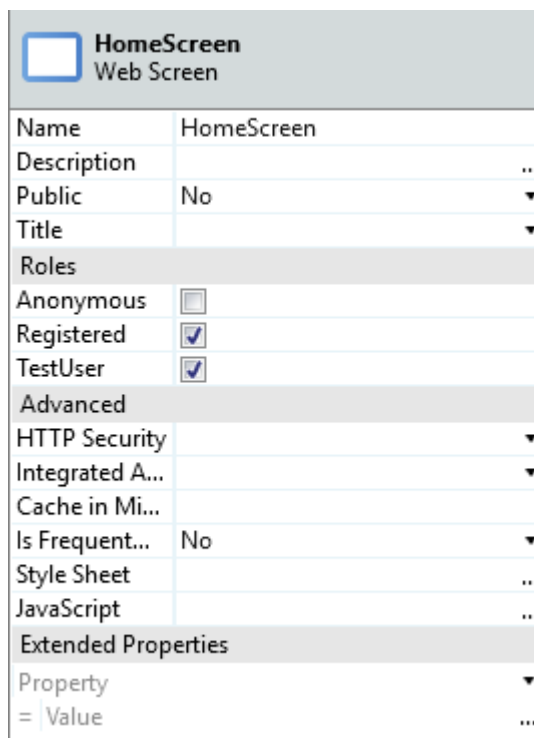
Rekisteröitymisnäky. Rekisteröityessään yrityksen tulee täyttää seuraavat kentät: Y-tunnus, nimi, osoite, sähköposti ja kontaktin puhelinnumero.

Kirjautumis näky. Kirjaututtuaan yritykselle avautuu näky sen omista tiedoista ja ilmoituksista. Näkyyn sisältyy muun muassa linkki omista ilmoituksista ja linkki uuden ilmoituksen jättämiseksi. Omia olemassa olevia ilmoituksia tulisi päästä muokkaamaan tai poistamaan selvällä paikalla olevan linkin kautta. Uutta ilmoitusta jättäessä tulisi täyttää tietyt ehdot, kuten osoite ja kontaktin yhteystiedot.

5.1 Service Studio

Service Studion editor area:ssa (kuva 6) kehittäjä rakentaa sovelluksensa ulkoasun ja logiikan. Kuten monien muidenkin ohjelmien editoreissa, saatavilla on monenlaisia erilaisia suunnittelua helpottavia työkaluja ja pohjia, joilla sivuja voi muokata haluamallaan tavalla sekä visuaalisesti että loogisesti.

Oikealta alakulmasta löytyvän asetukset -valikosta (kuva 14) muokataan valitun komponentin tietoja ja se, mitä tietoja komponentit lähettävät toisilleen.



Kuva 14. Asetukset -valikko.

Ylhäältä löytyvien Workspace-välilehtien avulla voi siirtyä serverin näkymien ja OutSystems Forgen välillä. Jokainen applikaatio koostuu monista erilaisista moduuleista, joista keskustellaan myöhemmin.

Oikealla ylhäällä sijaitsee layer-välilehdet (kuva 15). Layer-välilehdistä kukin sisältää oman puurakenteensa, josta hallinnoidaan applikaation rakennetta. Näiden välilehtien puurakenteista löytyy kaikki kehittäjälle tärkeimmät ominaisuudet ja toiminnallisuudet.



Kuva 15. WorkSpace-välilehdet.

Taulukko 4. Workspace välilehdiltä löytyvät toiminnallisuudet.

Data	Määritetään ja hallitaan kaikki tietokantoihin liittyvät toiminnallisuudet.
Logic	Kehitetään applikaation koodi, jota on mahdollista käyttää myös muiden moduulien toimivuuksissa. Lisäksi tällä välilehdellä määritetään datayhteydet, esimerkiksi voidaan toteuttaa ulkopuolinen REST API -rajapinta.
Interface	Tällä välilehdellä kehittäjä määrittelee applikaation ulkoasun, sekä sinne tallennetaan esimerkiksi tarvittavat kuvat ja teemat.
Process	Toteutetaan automaattiset ja synkronoidut työt kuten päivitykset.

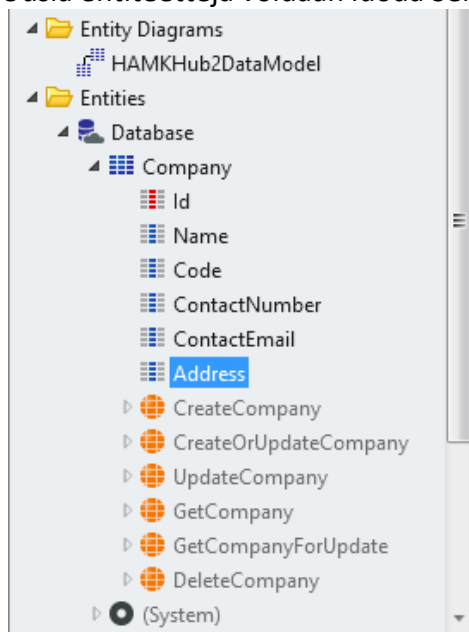
5.2 OutSystemsin dataobjektit

OutSystemsin käyttämä kieli on ”Strongly Type” -tyyppistä, joka tarkoittaa, että jokainen käytettävä datatyyppi tarvitsee määrittää yksilökohtaisesti, eikä sitä voida periä kuten esimerkiksi C#-kielessä. (OutSystems, n.d.j.)

Yleisin OutSystemes Platformissa käytetty datatyyppi on nimeltään entiteetti (Entity). Entiteetin voi käsittää tietokantatauluna, jolle asetetaan tiettyjä arvoja, eli attribuutteja. Esimerkiksi entiteetti voisi olla ”Asiakas”, jolla on attribuutit ”nimi” ja ”osoite”. Eli jokainen attribuutti on tietokantataulun ”Entity” kolumni ”Attribute”. Jokaiselle entiteetille alustuu automaattisesti entity identifier, joka on entiteetin primary key. Entiteetin primary key:ksi voi myös jälkepäin asettaa jonkun muun attribuutin halutessaan. (OutSystems, n.d.j.)

Attribuutille täytyy määrittää luodessa sen tyyppi ennalta määritetystä listasta. OutSystems tosin osaa myös ”arvata” attribuutin tyyppin valmiiksi sille annetun nimen perusteella. Esimerkiksi lisätessä entiteetille ”Company” attribuutti ”ContactEmail”, OutSystems osaa asettaa sen tyyppiä automaattisesti ”Email”. Attribuutille voidaan myös määrittää muun muassa oletusarvo, tai osoittaa onko se pakollinen tieto uutta entiteettiä luodessa (Kuva 16). (OutSystems, n.d.j.)

Uusia entiteettejä voidaan luoda Service Studion "Data" välilehdellä.



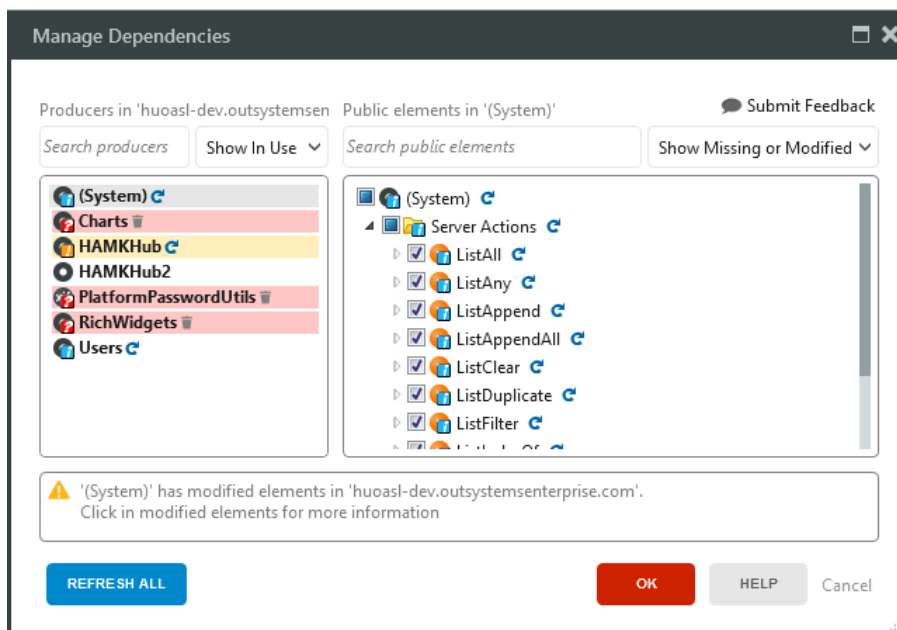
Kuva 16. Moduuliin on luotu entiteetti "Company" ja sille on määritetty tarvittavat attribuutit. (OutSystems. n.d.j.)

Kehittäjän on mahdollisuus luoda myös staattisia entiteettejä, joilla on etukäteen määritetyt arvot joita ei voida käyttäjän puolesta muokata.

Entiteetin luomisen jälkeen Service Studio luo automaattisesti sille perustoiminnot, joita entiteetin hallinnassa, tallennuksessa ja yleisessä käytössä yleensä tarvitaan. Entiteetti on oletusasetuksiltaan yksityinen ja kirjoitussuojattu, mutta nämä voidaan muuttaa entiteetin property-ikkunassa. (OutSystems, n.d.j.)

5.3 Moduulit

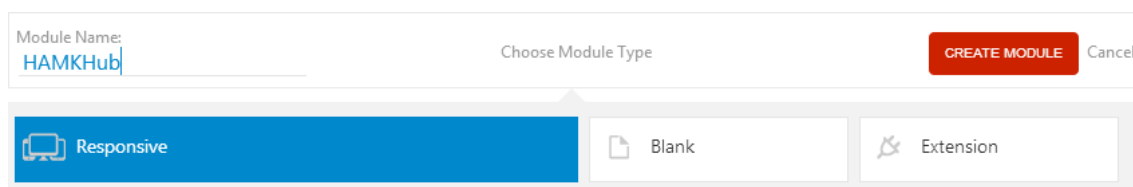
Kehitettäessä applikaatiota OutSystemsissa, voidaan sen eri toiminnallisuuksia jakaa moduuleihin, ja antaa niille sitten oikeuksia käyttää toistensa tietoja. Esimerkiksi kaikki dataan ja tietokantoihin liittyvät asiat voivat sijaita eri moduulissa kuin itse käyttöliittymä, ja käyttöliittymä voi hyödyntää taustalla olevaa datamoduulia. Tämä keventää applikaation käyttöä ja helpottaa suuresti sen ylläpitoa. Moduuleille voidaan antaa oikeuksia käyttää toistensa tietoja Manage Dependencies -valikosta (kuva 17).



Kuva 17. Manage Dependencies -valikosta voidaan antaa moduuleille oikeuksia käyttää toistensa tietoja.

5.4 Applikaation luominen

Aluksi luodaan uusi moduuli, jonka sisälle lähdetään rakentamaan web -sivustoa (kuva 18). Jos haluaa käyttäjille parhaan mahdollisen käyttökokemuksen myös mobiilin kautta, on hyvä valita sivuston tyyppi responsiivinen. Responsiivinen sivusto skaalautuu käytettävän laitteen näytön koon mukaan, joten se on käyttäjäystävällinen ja toimiva niin mobiilissa kuin tietokoneen kautta käytettäessä. Muita mahdollisia tyyppivalintoja ovat tyhjä sekä laajennus.

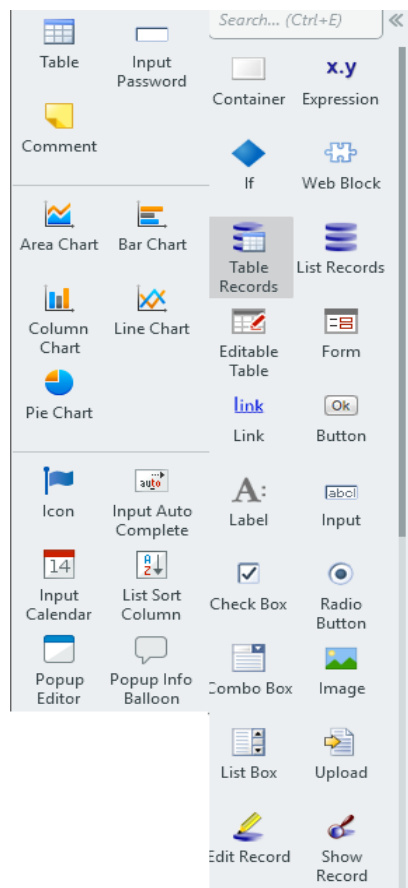


Kuva 18. Uuden moduulin luominen.

Aloittaessa uutta responsiivista web-moduulia OutSystems alustaa kehittäjälle kotisivun valmiiksi. Sivun on vielä tyhjä, mutta sitä päästään suunnittelemaan välittömästi.

Kehittäjän on mahdollista myös sekä käyttää ulkopuolisia pohjia, että rakentaa niitä itse tarpeensa mukaan. Myös esimerkiksi Forgesta voi saada käyttöönsä ulkopuolisten luomia pohjia, ja asettaa ne käyttöön läpi koko sovelluksen.

Sivun ulkoasun suunnitteluun OutSystems tarjoaa kaikki peruselementit valmiiksi, eikä niitä tarvitse tuoda vaivalloisesti moduulin sisään. Service Studio vasemmalta löytyy toolboksi, josta voidaan raahata haluttuja elementtejä sivulle, ja asetella ne kuten kehittäjä haluaa (Kuva 19).



Kuva 19. Service Studion tarjoaman toolboxin työkalut.

Kaikkien elementtien pitää sijaita containereissa, ja containereita voi olla myös sisäkkäin toistensa kanssa. Elementtejä voi kustomoida joko peruskustomointi-työkaluilla, esimerkiksi määrittämällä marginaaleja eri elementtien välille. Lisäksi Service Studio tarjoaa CSS (Cascading style sheets) -kustomointia sitä osaaville. CSS:ää käytetään perinteisesti kaikissa web-sivustoissa käyttöliittymän suunnittelussa, ja sillä saakin käyttöliittymän mukautumaan täysin käyttäjän haluamalla tavalla.

Service Studio tarjoaa todella helpon tavan linkittää sivustoja keskenään esimerkiksi napinpainallusten välille. Raahattuaan napin ikkunaan, hiiren oikealla painikkeella kehittäjä voi määrittää mihin sivustolle navigoidaan painalluksen jälkeen. Jos sivustoa ei ole vielä olemassa, tästä aktiviteetista voidaan se myös luoda suoraan, ja sivustot linkittyvät automaattisesti.

Lisäksi sivustoille on mahdollista määrittää yhteisiä ominaisuuksia, aivan kuten normaalienkin nettisivustojen kanssa. Sivustojen alta löytyvät footerit tai sivuston kattavat menut ovat tästä hyvä esimerkki. OutSystemsissä nämä sivustoille yhteiset elementit hoidetaan niin kutsutuilla web-blokeilla. Kun web-blokki liitetään UI Flowhun, se voidaan UI Flowhun suhteasetuksista määrittää näkyviin halutuille sivustoille.

Mahdolliset virhetilanteet ja niiden hallinta tapahtuvat OnException-elementeillä. Jokainen erilainen Exception Handler käsittelee omaa, tietyn tyyppistä virhetilannettaan, joten kehittäjä voi luoda monia eri Handlereita samaan projektiin ja jopa samaan UI Flowhun.

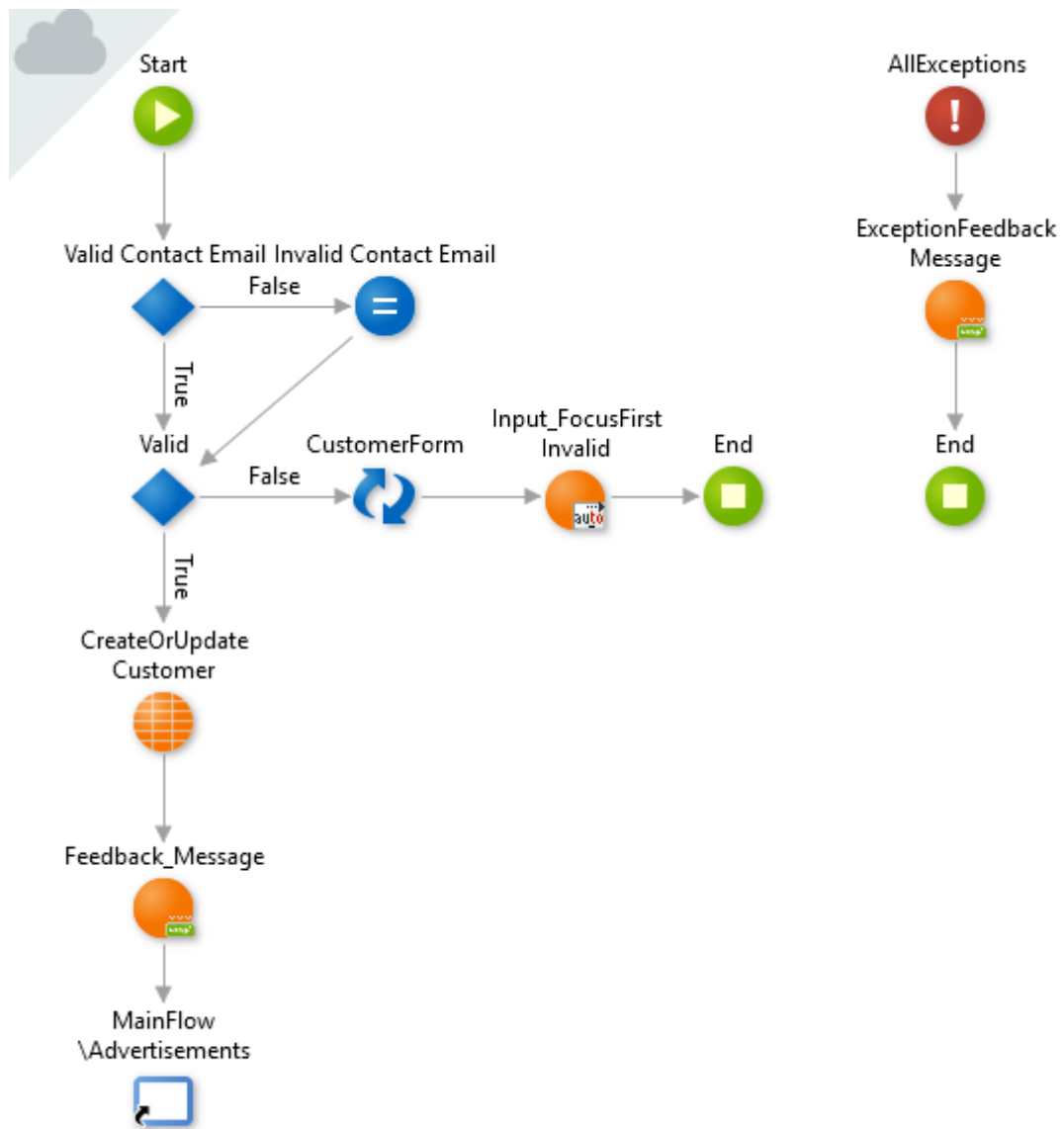
5.5 Datan näyttäminen sivuilla

Entiteettien kautta johdetuista tiedoista voidaan johtaa data käyttöliittymille monella eri tapaa. Usein nopein ja toimivin ratkaisu on kuitenkin antaa OutSystemsin rakentaa pohjat valmiiksi, taustalogiikan kera, ja kehittäjä voi keskittyä lähinnä ulkoasuun ja lisäominaisuuksien rakentamiseen. OutSystems osaa entiteettien pohjalta rakentaa valmiiksi sekä entiteetin yksityiskohtaisen esittely- ja muokkaussivun, tai listan, jossa kaikki tietokannasta löytyvät entiteetin yksilöt näytetään.

Käyttöliittymään voi määrittää tapahtumia, joiden avulla tietoa haetaan, luodaan ja poistetaan tietokannasta. Tietoa voidaan myös esiladata Preparaatioiden eli tietynlaisten alustusten kautta, jotka ajetaan aina samalla hetkellä, kun käyttöliittymää piirretään käyttäjälle. Esimerkiksi siirrettäessä objekti web-sivustolta toiselle, voidaan data vastaanottaa ja käsitellä valmiiksi, jotta se on käyttäjälle toimintakykyinen niin pian kuin mahdollista.

Kun verkkosivulle luodaan lomake, jossa halutaan näyttää tiettyjä entiteettejä, voidaan entiteetti raahata suoraan lomakkeelle Data layeristä, ja lomake osaa muodostaa oikeat kolumnit automaattisesti. Kehittäjän täytyy ainoastaan kertoa, mistä kyseinen tieto on saatavilla.

Erilaisia actioneita voidaan liittää verkkosivulle nappien kautta. Napit voivat liittyä johonkin esimerkiksi jonkin lomakkeen tietoihin, tai ihan vain uudelleenohjattakseen käyttäjän seuraavalle sivulle (kuva 20).



Kuva 20. Yrityksen rekisteröinti -ikkunan toiminnon "Register" toiminta. Kun napia painetaan, kutsutaan CreateOrUpdateCustomer-metodia, joka tallentaa uuden yrityksen tietokantaan. Tämän jälkeen siirrytään "Kirjautuneen yrityksen näkymä" -ikkunaan.

Entiteettien välille voi luoda viittauksia kuten normaaleissa yleisesti käytetyissä tietokannoissa. OutSystems luo viittaukset automaattisesti, ja helpoin tapa luoda viittaus on valitessa attribuutin tyyppiä asettaa se toisen entiteetin attribuutiksi, jolloin viittaus muodostetaan suoraan. Viittausta ei kuitenkaan voi tehdä muuhun attribuuttiin kuin entityn primary key:hin.

6 APPLIKAATION JULKAISU JA JATKOKEHITYS

Kehityksen jälkeen applikaation julkaisua ja jatkokehitystä hallinnoidaan OutSystems Platformin visuaalisella hallintatyökalulla. Hallintatyökalusta voidaan tarkastella eri ympäristöjä, joita kehittäjällä on käytettävissään. Jotta applikaatiota voidaan sujuvasti jatkokehittää, on suotavaa, että kehittäjällä on käytössään useampia ympäristöjä kuin pelkkä kehitysympäristö.

Kehitysympäristöstä applikaatio kopioidaan seuraavaksi testausympäristöön. Näin samasta applikaatiosta syntyy kaksi eri versiota: testausversio jonka testaaminen voidaan aloittaa ennen sen viemistä tuotantoon, ja kehitysversio, johon voidaan alkaa suunnitella ja rakentaa uusia ominaisuuksia. Kumpaankin versioon voidaan tehdä muutoksia toisesta tietämättä.

Testausvaiheessa applikaatioon ei enää kehitetä mitään uusia ominaisuuksia, mutta siihen korjataan mahdollisia vikoja (bugs) ja logiikkavirheitä. Kun applikaation on testausten ja korjausten jälkeen valmis, se voidaan siirtää viimeiseen vaiheeseen, eli tuotantoon. Tuotantoympäristö on jälleen täysin oma riippumaton ympäristö, ja sen versio on käytettävissä asiakkaille asti.

Applikaatio päivittää OutSystemsissä itsensä automaattisesti käyttäen One-click-publish -toimintoa. Jos kehitysympäristöstä julkaistaan suoraan asiakkaille tuleva versio, ei tarvita hallintaa OutSystems Platformin taholta, vaan kaikki päivittyvät asiakkaille automaattisesti. Muuten versionhallinta tapahtuu kuten aikaisemmin todettu OutSystemsin hallintatyökalua käyttäen (kuva 12).

7 YHTEENVETO

Opinnäytetyön tavoitteena oli toteuttaa HamkHub web-portaali, johon yritykset voivat rekisteröityä ja ilmoittaa vapaana olevista työ-, opinnäytetyö ja harjoittelu-paikoista Hämeen ammattikorkeakoulun opiskelijoille. Portaalin rakennus sujui hyvin, mutta jatkokehitettävää jäi vielä tulevaisuuden varalle. Esimerkiksi tietoturvalisua tai portaalin ulkoasullisia asioita ei tässä työssä otettu huomioon lähes ollenkaan. Opinnäytetyön aikataulu ja raportti myöhästyi noin puolella vuodella alkuperäisestä aikataulusta, mutta osapuolet ovat kuitenkin tyytyväisiä, että työ lopulta valmistui.

OutSystems on todella kehittäjän arkea helpottava työkalu, mutta jos kehittäjällä on yhtään enempää kokemusta ja taitoa normaalista kehitysmallista, on ohjelman helppouteen vaikea luottaa. Ohjelma osaa esimerkiksi itse rakentaa tarvittavat tietokantayhteydet, joiden toimintaan kehittäjän täytyy oppia luottamaan.

Suurimmat ongelmat portaalin kehityksessä tulivat joko liiallisesta manuaalisesta kehittämisestä, sekä välillä hieman epämääräisistä virheilmoituksista. Usein esimerkiksi kehittäjän oikeuksiin liittyvien ongelmien virheviesteissä ei itse ratkaisua tuotu esiin lainkaan, vaan niihin saatavia vastauksia joutui etsimään internetistä ja OutSystemsin virallisilta foorumeilta. Lisäksi OutSystemsin kertovien lähteiden löytäminen oli todella haastavaa, koska saatavilla olevaa dokumentaatiota ei yhtiön sivuja lukuun ottamatta juuri ole.

OutSystems on kuitenkin erittäin hyvä ohjelma etenkin yksinkertaisten applikaatioiden rakentamiseen ja hallintaan. Myös kehityksen jälkeisiin työkaluihin, kuten erilaisiin julkaisu- ja jatkokehitys on panostettu todella paljon.

LÄHTEET

Chappell D. 2008. What is application lifecycle management? Viitattu 22.10.2017 osoitteesta <http://www.davidchappell.com/WhatIsALM--Chappell.pdf>

Ciot T. 2016. What is a Low -Code Platform. Viitattu 24.9.2017 osoitteesta <https://www.progress.com/blogs/what-is-a-low-code-platform>

Fielding R. (2000). Viitattu 9.5.2017.
https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

Naz R. & Khan M. (2015). Rapid Applications Development Techniques: A Critical Review. International Journal of Software Engineering and Its Applications Vol. 9 No. 11. Viitattu 9.9.2017.
http://www.sersc.org/journals/IJSEIA/vol9_no11_2015/15.pdf

OutSystems (2017). What is low code. Viitattu 24.9.2017 osoitteesta <https://www.outsystems.com/blog/what-is-low-code.html>

OutSystems (n.d.a). Rapid Application Delivery Platforms. Viitattu 23.9.2017 osoitteesta <https://www.outsystems.com/res/outsystems-rapid-application-delivery-platforms>

OutSystems (n.d.b). Choose Between a Mobile or a Web App. Viitattu 11.5.2017 osoitteesta https://success.outsystems.com/Documentation/10/Getting_Started/Choose_Between_a_Mobile_or_a_Web_App

OutSystems (n.d.c). Personal Environment/What's an OutSystems personal environment. Viitattu 23.9.2017 osoitteesta https://success.outsystems.com/Support/Personal_Environment/What's_an_OutSystems_personal_environment

OutSystems (n.d.d). How is software in the Forge licensed. Viitattu 11.5.2017 osoitteesta https://success.outsystems.com/Evaluation/Architecture/1_OutSystems_Platform_tools_and_components/00_How_is_software_in_the_Forge_licensed%3F

OutSystems (n.d.e). Outsystems Overview. Viitattu 11.5.2017 osoitteesta <https://www.outsystems.com/learn/lesson/851/outsystems-overview/>

OutSystems (n.d.f). OutSystems Platform tools and components. Viitattu 11.5.2017 osoitteesta https://success.outsystems.com/Evaluation/Architecture/1_OutSystems_Platform_tools_and_components

OutSystems. (n.d.g) LifeCycle Management. Viitattu 22.10.2017 osoitteesta https://success.outsystems.com/Evaluation/Lifecycle_Management

OutSystems. (n.d.h) How does OutSystems handle deployment and staging. Viitattu 22.10.2017 osoitteesta https://success.outsystems.com/Evaluation/Lifecycle_Management/0_How_does_OutSystems_handle_deployment_and_staging

OutSystems. (n.d.i) What kind of monitoring and analytics does OutSystems offer. Viitattu 22.10.2017 osoitteesta https://success.outsystems.com/Evaluation/Lifecycle_Management/2_What_kind_of_monitoring_and_analytics_does_OutSystems_offer

OutSystems (n.d.j). OutSystems Platform architecture. Viitattu 13.5.2017 osoitteesta https://success.outsystems.com/Evaluation/Architecture/2_OutSystems_Platform_architecture

OutSystems (n.d.k). Configuration management for OutSystems applications. Viitattu 22.10.2017 osoitteesta https://success.outsystems.com/Evaluation/Lifecycle_Management/4_Configuration_management_for_OutSystems_applications

OutSystems (n.d.l). Modeling Data. Viitattu 21.10.2017 osoitteesta <https://www.outsystems.com/learn/lesson/874/modeling-data/>

OutSystems (n.d.m). Service Studio. Viitattu 21.10.2017 osoitteesta <https://www.outsystems.com/platform/diagrams/service-studio/>

OutSystems. (n.d.n). OutSystems Platform architecture. Viitattu 14.10.2017 osoitteesta https://success.outsystems.com/Evaluation/Architecture/2_OutSystems_

Rossberg J. 2013. Beginning Application Lifecycle Management. Viitattu 22.10.2017 osoitteesta <http://library.books24x7.com.ezproxy.jamk.fi:2048/asset-viewer.aspx?bookid=53246&chunkid=199118833¬eMenuToggle=0&left-MenuState=1>