



TAMPEREEN
AMMATTIKORKEAKOULU

MICROSOFT AZURE -RATKAISU TAPAHTU- MAILMOITTAUTUMISIA VARTEN

Ville Kärkkäinen

Ohjelmointiprojekti
Joulukuu 2015
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

KÄRKKÄINEN VILLE:

Microsoft Azure -ratkaisu tapahtumailmoittautumisia varten

Opinnäytetyö 29 sivua, joista liitteitä 0 sivua
Joulukuu 2015

Opinnäytetyö tehtiin yritykselle Mepco Oy. Työn tavoitteena oli luoda pohja kokonaisuudelle, jonka avulla asiakasyritykset voisivat vastaanottaa järjestämiensä tapahtumien ja kurssien ilmoittautumistietoja. Työssä tehtävän sivuston tuli saada tapahtumien tiedot Microsoft Dynamics CRM -asiakkuudenhallintaohjelmistosta, pyytää ilmoittautujan henkilötiedot, ohjata tämä maksamaan tapahtuma ja tallentaa tiedot maksetuista ilmoittautumisista takaisin CRM -ohjelmiston tietokantaan. Lisäksi tapahtumaan ilmoittautumisen tuli olla helposti jaettavan linkin takana.

Työn aluksi tutkittiin työhön kelpaavia tekniikoita ja toteutustapoja. Näiden valinnan jälkeen selvitettiin miten sivustosta saadaan riittävän luotettava että tiedot maksetuista maksuista tallennetaan aina oikein. Tietoturvasyistä työssä perehdyttiin myös erilaisiin saalausmenetelmiin. Työ toteutettiin pääasiassa C#-ohjelmointikielellä.

Työn tuloksena syntyi Microsoft Azure -pohjainen sivusto, joka lukee URL-parametrissa tapahtuman tiedot, antaa ilmoittautujan syöttää henkilötiedot ja ohjaa tämän maksamaan Checkout-verkkomaksupalvelun avulla. Linkki ilmoittautumissivustolle luodaan Microsoft Dynamics CRM -liitännäisen avulla. Liitännäinen lukee tapahtuman tiedot ja sisällyttää ne salattuna merkkijonona linkkiin. Onnistuneen maksun jälkeen sivusto lähettää CRM:lle tiedon ilmoittautumisesta.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software engineering

KÄRKKÄINEN VILLE:

Microsoft Azure Solution for Event Enrolment

Bachelor's thesis 29 pages, appendices 0 pages
December 2015

Bachelor's thesis was done for the company Mepco Oy. Goal of the thesis was to create a basis for a product with which client companies could handle the enrolments for courses and events they organize. The website created as part of the thesis was to receive the info on the events from Microsoft Dynamics CRM customer relationship management software, request the enrollee for his or her personal data, then guide him or her to pay for the enrolment and save the info back to the CRM database. In addition the enrolment had to be reachable by a link that could be easily shared.

The work started by researching technologies, techniques and methods suitable for it. Then it was investigated how to guarantee that info on paid enrolment fees would be saved reliably every time. Various cryptographic algorithms were studied for security reasons. Most of the work was done using the programming language C#.

A website based in Microsoft Azure cloud was create as a result. The website reads event information from the URL parameter, has the enrollee type his or her personal data and then guides the enrollee to pay with Checkout online payment service. The link to the enrolment website is created dynamically with a Microsoft Dynamics CRM plugin that encrypts all the necessary info on the event and includes them in the link. After a successful payment the website sends the enrolment info to CRM.

Key words: payment services, cryptography, databases

SISÄLLYS

JOHDANTO	6
1 TARVE.....	7
1.1 Käyttötapaukset	7
1.2 Työn rajaus	7
2 KÄYTETYT TEKNIIKAT JA PALVELUT	9
2.1 Microsoft Dynamics CRM.....	9
2.2 Microsoft Azure ja Azure Table	10
2.3 ASP.NET MVC	11
2.4 Checkout	12
3 ARKKITEHTUURI	13
3.1 Moduulit.....	13
3.1.1 Microsoft Dynamics CRM –plugin.....	13
3.1.2 Ilmoittautumissivusto.....	14
3.1.3 Maksutila-automaatio.....	14
3.1.4 Sovelluslaajennuskomponentit.....	15
3.2 Fyysinen sijainti ja kommunikointi	16
4 TEKNINEN TOTEUTUS	18
4.1 Plugin	18
4.2 Ilmoittautumissivusto.....	19
4.3 Maksutila-automaatio	21
5 KÄYTTÖ.....	23
5.1 Tapahtuman ja ilmoittautumislinkin luonti.....	23
5.2 Tapahtumaan ilmoittautuminen	24
5.3 Ilmoittautumisen näkeminen CRM:ssä.....	26
6 POHDINTA.....	28
LÄHTEET.....	29

LYHENTEET JA TERMIT

CRM	Microsoft Dynamics CRM -ohjelmisto
.NET Framework	Microsoftin luoma pääasiassa Windowsissa käytettävä ohjelmistokehys
C#	Oliopohjainen ohjelmointikieli, osa .NET Frameworkkia
Plugin	Ohjelmistoliitännäinen joka lisää ohjelmaan uusia toimintoja

JOHDANTO

Tämä insinöörityö on tehty yritykselle Mepco Oy ja työn aihe syntyi asiakasyritysten tarpeista. Mepco Oy on suomalainen ohjelmistoyritys joka tarjoaa yrityksille hallintaohjelmistoja ja -ratkaisuja.

Asiakasyritykset järjestävät tapahtumia ja kursseja, joihin halutaan kutsua ihmisiä. Osa ilmoittautumisista on maksullisia. Aiemmin tällaiset ilmoittautumiset on voitu hoitaa esimerkiksi pyytämällä ilmoittautujia lähettämään ilmoittautumistietonsa sähköpostilla ja lähettämällä heille sitten tiedot maksua varten. Tämä kuitenkin vaatii yrityksen työntekijöiltä paljon työtä erityisesti maksuja tarkistettaessa.

Tilannetta helpottamaan haluttiin palvelu, joka automatisoisi yrityksen työntekijöiden muute tekemästä työstä mahdollisimman suuren osan. Yritykselle jäisi tehtäväksi tallentaa tapahtuman tiedot asiakkuudenhallintaohjelmistoonsa ja lähettää kutsut. Lisäksi ilmoittautumisen tulisi olla riittävän helppoa että vaikeus tai vaadittu työmäärä eivät olisi syynä sille ettei kutsuttu henkilö ilmoittau.

Tässä työssä selvitetään mitä tekniikoita tarvetta täyttämään vaaditaan ja miten tarpeet täyttävä työ ohjelmoidaan. Maksuliikenteen takia erityistä huomiota kiinnitetään salausmenetelmiin ja välitettävien viestien aitouden todentamiseen.

1 TARVE

Työn aluksi pohdittiin ilmoittautumistapahtumaan liittyvää prosessia sekä yrityksen että ilmoittautujan osalta. Tämän perusteella pystyttiin tarkemmin selvittämään mitä työssä tehtävän kokonaisuuden täytyy tehdä, mitä osakokonaisuuksia se tulee sisältämään ja miten nämä osat liittyvät toisiinsa. Samalla päätettiin mitä rajattiin työn ulkopuolelle.

1.1 Käyttötapaukset

Kun yritys luo CRM–hallintaohjelmistoonsa tiedon tapahtumasta tai kurssista johon halutaan osallistujia, työ luo linkin jonka kautta tapahtumaan voidaan ilmoittautua. Tämä jälkeen yrityksen työntekijä lähettää linkin kaikille tapahtumaan kutsutuille.

Kun kutsuttu avaa linkin, hän päätyy sivulle jossa esitetään tapahtuman perustiedot ja pyydetään ilmoittautujan henkilötiedot. Henkilötiedot syötettyään ilmoittautujan tiedot tallennetaan ja hänet viedään sivulle jossa näytetään useita maksutapoja joista valita. Maksutapahtuman ajaksi käyttäjä ohjautuu ulkopuolisen palveluntarjoajan sivulle.

Onnistuneen maksutapahtuman jälkeen ilmoittautujan ja maksun tiedot tallennetaan yrityksen CRM:n tietokantaan jolloin yrityksen edustajat näkevät ilmoittautumiset CRM:n käyttöliittymästä käsin. Tämän tulee tapahtua vaikka asiakas sulkisi selaimen heti maksettuaan palaamatta takaisin ilmoittautumissivulle, tai vaikka esimerkiksi yhteysongelman takia tiedon tallentaminen ei heti onnistuisi.

1.2 Työn rajaus

Luodun linkin lähetys kutsutuille päätettiin jättää työn mittakaavan ulkopuolelle. Eri yritykset jotka ottavat valmiin työn käyttöön saattavat haluta jakaa linkkiä eri tavoin joten toteutus voidaan jättää asiakaskohtaiseksi räätälöinniksi. Lisäksi esimerkiksi linkin liittäminen sähköpostiin ei toteutuksena sisällä mitään haastavaa tai selvitystä vaativaa joten sen sisällyttämistä työhön ei pidetty mielekkäänä.

Työn tuloksena luodun verkkosivun ulkoasuun ei kiinnitetty työn aikana suurta huomiota. Lopulliseen ulkoasuun vaikuttaa esimerkiksi yrityksen markkinointiosaston vaatimukset ja yrityksen mahdollisesti päättämät ulkoasua ja logoja koskevat säännöt.

Vaikka lopullisessa käytössä osa tapahtumista joihin verkkosivun kautta voidaan ilmoitautua on ilmaisia, työhön sisällytettiin vain maksullisiin tapahtumiin ilmoittautuminen. Valintaan päädyttiin siksi että verkkomaksuihin liittyvät haasteet ovat merkittävä osa työtä ja ilmaiset tapahtumat eivät lisää kokonaisuuteen lainkaan uusia haasteita tai uutta pohdittavaa. Kun muu kokonaisuus toimii, ilmaisten tapahtumien mahdollistaminen on triviaalia.

Viimeisenä isona rajauksena päätettiin jättää työstä pois CRM:ään luotavat räätälöidyt ilmoittautumistietueet. Näiden sijaan työssä osoitetaan ilmoittautumistiedon lähetyksen toimivuus tallentamalla CRM-kantaan asiakastietue, jonka nimeksi asetetaan ilmoittautumistapahtuman yksilöivä tunnus. Syynä tähän on että lopullinen tapa miten ilmoittautuminen halutaan esittää CRM:ssä riippuu asiakasyritysten toiveista. Ilmoittautumisen esitystavan muokkaus on hyvin yksinkertaista sen jälkeen kun käytettävästä esitystavasta on sovittu, joten se ei olisi työn tutkimuksellisten tavoitteiden kannalta mielekäästä.

2 KÄYTETYT TEKNIIKAT JA PALVELUT

Mepco Oy:n yksi päätuote on yrityskohtaiset Microsoft Dynamics CRM-ratkaisut. Tämä osaltaan vaikutti toteutusmenetelmien valintaan. Alustaksi ilmoittautumissivustolle valittiin Microsoft Azure -pilvipalvelu, joka tarjoaa verkkosivutilan lisäksi paljon apuvälineitä tiedon tallentamiseen ja välittämiseen. Tämän lisäksi Azure-sivun toimintaa voidaan laajentaa samalla ohjelmointikielellä jolla tehdään CRM-plugineja.

2.1 Microsoft Dynamics CRM

Microsoft Dynamics CRM on Microsoftin kehittämä alun perin vuonna 2003 julkaistu asiakkuudenhallintaohjelmisto. CRM tarjoaa välineitä myynnin, markkinoinnin ja asiakaspalvelun edistämiseen ja sitä voidaan laajentaa sopimaan lähes minkä tahansa prosessien ja resurssien hallintaan. Uusin käytössä oleva versio ohjelmistosta on CRM 2015 joka julkaistiin loppuvuodesta 2014.

CRM:ää voidaan laajentaa useilla eri tavoilla. Käyttöliittymän kautta voidaan muokata ohjelmiston käyttämien lomakkeiden ulkoasua (kuva 1), luoda täysin uudenlaisia lomakkeita ja määrittää miten niiden suhteet toisiinsa. Lisäksi käyttöliittymän kautta voidaan säätää yksinkertaisia automaatioita jotka käsittelevät järjestelmässä olevia lomakkeita halutulla tapaa.

TALLENNA TALLENNA JA SULJE LOMAKE-EDITORI

TUOTE
Tuote: Uusi Tuote

YHTEENVETO

Nimi*

Tuotetunnus* --

Ylätaso --

Voimassa alkaen --

Voimassa asti --

Kuvaus --

Yksikköryhmä* --

Oletusyksikkö* --

Oletushinnasto+ --

Desimaaleja käytettä* --

Aihe --

TUOTTEEN OMINAISUUDET

Nimi ↑	Perusominaisuus	Tietotyyppi	Vain luku
Saat tämän sisällön käyttöön, kun luot tietueen.			

KUVA 1. Osa yhdestä CRM: vakiolomakkeesta.

Lomakkeille voidaan lisätä Javascript-funktioita ja itsetehtyjä HTML-elementtejä joiden avulla voidaan saada aikaan toimintoja joita vakiokäyttöliittymä ei mahdollista. Lisäksi CRM:ään voidaan ohjelmoida .NET –yhteensopivilla kielillä plugineja jotka käynnistyvät määrättyissä tilanteissa.

2.2 Microsoft Azure ja Azure Table

Microsoft Azure on Microsoftin pilvipalvelu joka tarjoaa alustan virtuaalipalvelimille ja kehitysalustan pilvisovelluksille. Azure tukee useita ohjelmointikieliä, ohjelmia ja ohjelmistokehyksiä ja kytkeytyy saumattomasti Microsoft Visual Studio -ohjelmankehitysympäristöön.

Azure Table on Microsoftin tietokantapalvelu joka noudattaa NoSQL-mallia. Tämä tarkoittaa että se ei säilytä tietoa relaationaalisesti kuten perinteiset tietokannat (Azure Storage Table Design Guide). Lähestymistavan etuja ovat yksinkertaisuus, helppo skaalautuvuus ja tehokkuus tilanteissa joissa kannan tietoja täytyy voida käsitellä reaaliaikaisesti. Tällainen lähestymistapa sopii parhaiten tilanteisiin joissa tallennettavat rivit eivät kytkeydy toisiinsa vaan ovat toisistaan irrallista tietoa.

Azure Table käyttää tallennettujen rivien tunnistamiseen kahta arvoa: partitionkey eli osioavain ja rowkey eli riviavain. Usealla eri rivillä voi olla sama osioavain mutta saman osioavaimen riveillä tulee olla kaikilla keskenään uniikit riviavaimet. Tehokkuuden ja tilankäytön optimoinnin vuoksi Azure Table voi jakaa kannan rivit useille eri koneille, mutta saman osioavaimen rivit pysyvät aina yhdessä. Tästä syystä kannasta tiedon hakeminen yhdellä osioavaimella on huomattavasti nopeampaa kuin usealla osioavaimella. Toisaalta jos rivejä ei ole jaettu riittävän moneen eri osioavaimeen, Azure ei voi yrittää jakaa kuormitusta useille eri koneille mistä voi seurata kannan hidastumista. Tehdyssä työssä täytyy tallentaa keskenään samankaltaista ja toisistaan täysin irrallista tietoa joten relaationaalisuudesta ei olisi hyötyä.

2.3 ASP.NET MVC

ASP.NET MVC on Microsoftin kehittämä avoimen lähdekoodin web-ohjelmistokehys joka hyödyntää MVC-ohjelmistoarkkitehtuurityyliä. ASP.NET MVC mahdollistaa web-sovellusten kirjoittamisen kielillä C# ja VB.NET. Ilmoittautumissivusto toteutettiin ASP.NET MVC:llä tämän C#-tuen takia ja koska MVC-arkkitehtuuri soveltui hyvin käyttötarkoitukseen.

MVC tulee sanoista model-view-controller eli malli-näkymä-ohjain. Arkkitehtuurin tarkoituksena on erottaa käyttöliittymä ja data toisistaan. Ohjelma jaetaan kolmeen osaan:

- Malli säilyttää ohjelmassa näytettävän ja manipuloitavan datan.
- Näkymä määrittää miten käyttäjälle esitettävä tieto näytetään.
- Ohjain reagoi käyttäjältä tuleviin komentoihin ja muuttaa mallia ja näkymää niiden mukaan.

Malli ei riipu näkymästä eikä ohjaimesta, joten se voidaan suunnitella ja toteuttaa täysin irrallaan muusta sovelluksesta. Tämä oli iso etu ilmoittautumissivuston toteutuksessa, koska samaa koodia voitiin käyttää sekä käyttäjälle esitettävällä sivulla, että taustalla pyörivissä automaatioissa jotka huolehtivat tilanteista joissa tieto maksutapahtumasta ei ole välittynyt sivustolle.

2.4 Checkout

Checkout on suomalainen verkkomaksupalvelu, jonka voi liittää verkkosivuun. Liittäminen voidaan tehdä joko upottamalla verkkosivuun suoraan Checkoutin tarjoamat maksutavat, tai lisäämällä sivulle linkin joka ohjaa Checkoutin sivuille maksutapavalintaan. Checkoutilla on valmiit ohjelmistopaketit useisiin web-sovelluskehyskehyksiin ja kirjastot esimerkiksi Pythonille ja PHP:lle. Työssä kuitenkin jouduttiin kirjoittamaan toimiva C#-rajapinta jotta ASP.NET MVC-sivu toimi Checkoutin kanssa.

Yksilö tai yritys joka tahtoo voida vastaanottaa maksuja Checkoutin kautta joutuu tekemään Checkoutin kanssa maksullisen sopimuksen jolloin palveluun saa kauppiastunnuksen. Tämän työn toteutuksessa käytettiin ilmaisia testaustunnuksia joilla palvelu toimii aivan kuin maksullisillakin, mutta rahaa ei siirretä. Kun työ siirtyy asiakaskäyttöön, jokaisen asiakasyrityksen on tarkoitus hankkia oma kauppiastunnus palvelua varten jolloin Mepco Oy:n ei tarvitse olla osana ilmoittautumisiin liittyvää rahaliikennettä.

3 ARKKITEHTUURI

Työn vaatimukseen sisältyy useita eri prosesseja kuten maksusivulinkin luonti, maksaminen ja maksutietojen vienti CRM:ään. Tämän lisäksi työhön liittyy useita eri ohjelmistoja, pilvipalveluita ja tietokantoja. Näiden vuoksi kokonaisuuden arkkitehtuuria on luonnollista lähestyä sekä ohjelmamoduulien näkökulmasta että työn osien fyysisen sijainnin näkökulmasta.

3.1 Moduulit

Työssä tehty kokonaisuus koostuu kolmesta päämoduulista: CRM-plugin, ilmoittautumissivusto ja maksujen tiloja päivittävä automaatio. Näitä pääkomponentteja varten tehtiin sovelluslaajennuskomponentteja, jotka toteuttavat yksittäisiä päämoduulien tarvitsemia toimintoja ja palveluita kuten integraatiot Checkout-palveluun ja CRM-järjestelmään sekä kryptografiset salausten menetelmät.

Toiminnallisuuksien jaon takana oli pyrkimys erottaa työn päämoduulien toimintalogiikka niiden teknisestä toteutuksesta ja pitää työn osat riittävän pieninä jotta niiden vastuut pysyvät selkeinä. Tämä helpottaa ylläpitoa ja jatkokehitystä ja auttaa välttämään kehäriippuvuuksien syntyä. Kehäriippuvuus tarkoittaa tilannetta jossa eri komponenttien keskinäinen riippuvuus muodostaa suljetun piirin eikä hierarkkista jonoa tai puuta. Tällöin mitään komponenttia ei voida ladata tai kääntää ensin eikä ohjelma toimi.

3.1.1 Microsoft Dynamics CRM -plugin

CRM-järjestelmää laajentamaan työn tarpeita varten kehitettiin plugin, joka luo tapahtumakohtaisen hyperlinkin kutsun mukana lähettämistä varten. Plugin sisällyttää linkkiin salatun merkkijonon jossa on tapahtumaan ilmoittautumista varten vaadittavat tiedot kuten tapahtuman nimi, hinta ja viimeinen ilmoittautumispäivä. Plugin käynnistyy ja luo tai päivittää linkin aina, kun CRM:ään luodaan uusi tapahtumatietue tai sellaisen linkissä välitettäviä tietoja muutetaan.

Luotu linkki salataan Rijndael-salausalgoritmilla. Linkin oikeellisuuden varmistamiseksi siihen sisällytetään SHA-256-algoritmilla tehty 32-bittinen tiiviste. Tiiviste on merkkijono joka luodaan muista linkkiin sisällytettävistä tiedoista. Mikäli linkin mukana välitettävä tiiviste ei vastaa muita tietoja, tiedetään että linkkiä on muutettu CRM-liitännäisen ulkopuolella.

3.1.2 Ilmoittautumissivusto

Ilmoittautumissivustolle pääsee vain kelvollisen linkin kautta, muissa tapauksissa käyttäjälle näytetään virheilmoitus. Linkki on kelvollinen vain, mikäli siinä tapahtuman pakolliset tiedot, oikea SHA-256-tiiviste ja se on salattu oikein AES-algoritmilla. Tapahtuman tiedot luetaan linkistä ja käyttäjälle esitetään lomake jossa pyydetään ilmoittautujan nimi ja osoite. Näiden jälkeen ilmoittautujan tiedot tallennetaan Azure Table -tietokantaan ja ilmoittautujalle tarjotaan maksuvaihtoehdot. Maksun ajaksi käyttäjä ohjataan maksutavasta riippuen joko Checkout-palvelun tai pankin sivuille ja lopuksi käyttäjä palautetaan ilmoittautumissivustolle jossa ilmoitetaan onnistuiko maksu.

Mikäli käyttäjä palaa ilmoittautumissivustolle maksun jälkeen, tieto maksun onnistumisesta tallennetaan ilmoittautujan tietoihin tietokantaan. Onnistuneen maksun tapauksessa tieto lähetetään myös suoraan CRM-ohjelmiston tietokantaan. Mikäli käyttäjä ei palaa sivulle, vastuu maksusta huolehtimisesta siirtyy maksutila-automaatiolle. Näin tapahtuu myös, mikäli käyttäjä palaa sivulle, mutta tiedon välitys CRM:ään ei onnistu.

3.1.3 Maksutila-automaatio

Maksutila-automaatio on Microsoft Azure -pilvipalvelussa ajastettuna ajettava ohjelma, joka käsittelee Azure Table -tietokantaan tallennettuja ilmoittautumistietoja joiden maksua ei ole vielä vahvistettu tai joita ei ole pystytty lähettämään CRM:ään. Ohjelma pyrkii vahvistamaan ovatko ilmoittautumiset maksettu ja lähettämään kaikki maksetut ilmoittautumiset CRM:ään.

Ohjelma ottaa aluksi yhteyden Azure Table -tietokantaan ja hakee kaikki ilmoittautumiset, joista on tallennettu tieto maksusta mutta joita ei ole vielä lähetetty CRM:ään. Ohjelma käy nämä ilmoittautumiset läpi yksi kerrallaan ja pyrkii viemään jokaisesta tiedot CRM:ään. Tietoja viedessä tarkistetaan onko kyseisen maksun tiedot viety jo jotta ei luoda kaksoiskappaleita. Onnistuneesti lähetetyille ilmoittautumistiedoille tallennetaan tieto lähetyksestä ilmoittautumissivuston Azure Table -tietokantaan jotta ohjelma ei käsittele niitä enää uudestaan.

Seuraavaksi ohjelma hakee ilmoittautumiset, joiden maksua ei ole vielä vahvistettu. Nämä sisältävät kaikki tapaukset, joissa käyttäjä ei ole palannut maksun jälkeen ilmoittautumissivulle, mutta lisäksi kaikki nekin tapaukset joissa käyttäjä on esimerkiksi syötännyt tietonsa mutta ei ole siirtynyt lainkaan maksamaan. Ohjelma käy läpi kaikki ilmoittautumistiedot yksi kerrallaan ja pyytää Checkout-palvelulta tietoa niiden maksujen tilasta. Mikäli maksu on maksettu, tieto maksusta merkitään Azure Table -tietokantaan ja ilmoittautuminen käsitellään kuten aiemmassa tapauksessa ilmoittautumiset joita ei ollut vielä viety CRM:ään. Mikäli Checkout vastaa että maksutapahtuma on vielä kesken, maksu jää odottamaan seuraavaa automaation ajoa. Jos yhteydenotto Checkoutiin ei onnistu, ohjelma siirtyy käsittelemään seuraavaa ilmoittautumista. Jos yhteydenotto epäonnistuu kolme kertaa peräkkäin, ohjelma lopettaa ajonsa yrittääkseen uudelleen seuraavalla ajokerralla.

3.1.4 Sovelluslaajennuskomponentit

Checkout-palvelun kanssa kommunikointia varten vaadittava toiminnallisuus on sijoitettu omaan C#-projektiin. Tämä sisältää tarvittavat luokat ja metodit Checkoutiin yhdistämistä ja saatujen vastausten käsittelyä varten. Lisäksi komponentti tarjoaa muille luokille rajapinnan jonka kautta ne voivat käyttää Checkoutia ottamatta kantaa palvelun toimintaan. Käytetyt HTTP-kutsut on eriytetty omaan pieneen projektiin sen takia, että jatkossa jokin toinenkin osa työstä saattaa käyttää niitä.

Azure Table -kannan käyttöön liittyvät luokat ja metodit ovat yhdessä projektissa. Komponentti tarjoaa muiden luokkien käyttöön rajapinnan, jonka avulla maksutietoja voidaan

hakea, tallentaa ja muokata ilman että tietokannan toiminnasta tarvitsee tietää. Komponentti myös tarjoaa poikkeusluokan tilanteisiin jolloin haettavaa maksutietoa ei löydetä kannasta.

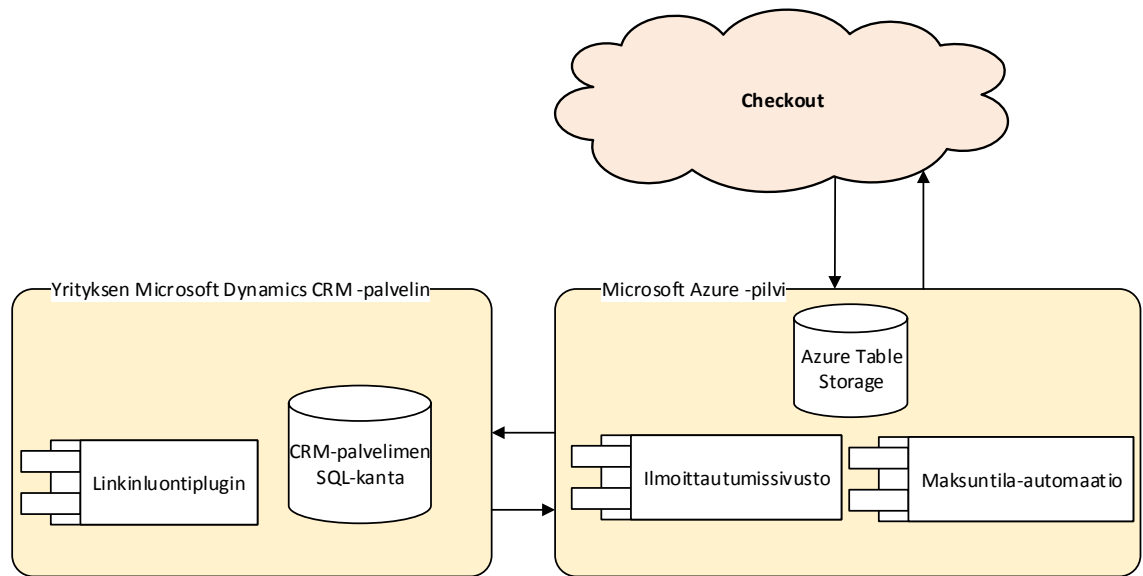
Salausmenetelmät, tiivisteiden luonti ja merkkijonojen pakkausmenetelmät on sijoitettu erilliseen kryptografiaprojektiin staattiseen luokkaan. Salausta tarvitaan linkin luontiin ja purkamiseen ja tiivisteitä Checkoutin kanssa kommunikointiin. Merkkijonon pakkausta käytetään maksutapahtuman yksilöivän tunnisteiden luontiin.

Omat pienet projektinsa on myös CRM:ään ilmoittautumistietojen lähetykseen, lokitukseen ja maksutapojen ja maksun tietojen säilytykseen. Maksun tietoja ja maksutapoja käytetään sekä ilmoittautumissivustolla, maksutila-automaatiossa että Checkout-integraatiossa joten niiden eriyttäminen omaan projektiinsa selkeytti riippuvuuksia.

3.2 Fyysinen sijainti ja kommunikointi

Linkinluontiplugin asennetaan asiakasyrityksen Microsoft Dynamics CRM-palvelimelle. Palvelin voi olla yrityksen oma, tai se voi olla Microsoftin pilvipalvelin mikäli yritys käyttää CRM:n Online-versiota. Ohjelman toiminnan kannalta tällä ei ole eroa. Sekä ilmoittautumissivusto että maksutila-automaatio sijaitsevat Microsoftin Azure-pilvessä. Samassa pilvessä on myös Azure Table -tietokanta. Checkout-palvelu sijaitsee palveluntarjoajan omalla palvelimella.

Azuressa sijaitsevat ilmoittautumissivusto ja maksutila-automaatio ovat yhteydessä sekä CRM-palvelimeen että Checkoutiin (kuvio 1). Sen sijaan Checkout ja CRM-palvelin eivät kommunikoi keskenään missään vaiheessa.



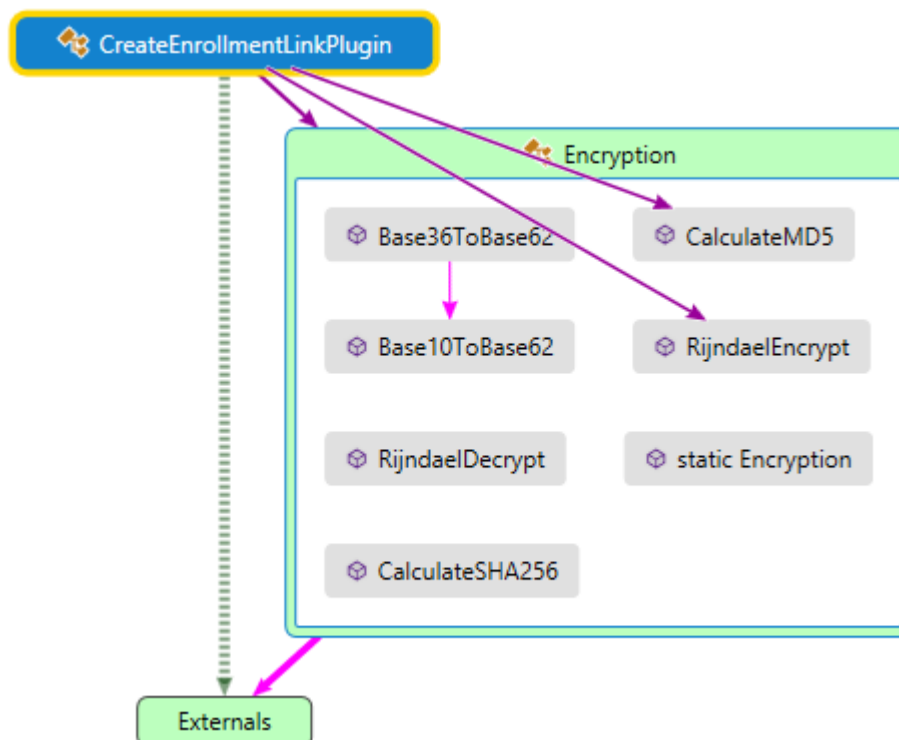
KUVIO 1. Komponenttien sijainnit ja yhteydet.

4 TEKNINEN TOTEUTUS

Lähes puolet työhön käytetystä ajasta kului tarvittavien komponenttien toteutukseen ja niiden toiminnan testaukseen. Erityistä haastetta aiheuttivat työn kannalta kriittiset yhteydet CRM-tietokantaan ja Checkout-palveluun. Koska tämä vaihe oli työn suurin yksittäinen osa, käydään pääkomponenttien tarkempi toteutus läpi.

4.1 Plugin

Ilmoittautumislinkin luovan pluginin toimintalogiikka tapahtuu yhden luokan sisällä. Luokka CreateEnrollmentLinkPlugin toteuttaa Microsoft Dynamics CRM:n kehitystyökalukirjaston tarjoaman plugin-rajapinnan IPlugin. Plugin-luokka käyttää linkin salaamiseen staattisen Encryption-luokan tarjoamia julkisia metodeja RijndaelEncrypt ja CalculateMD5 (kuvio 2).



KUVIO 2. CreateEnrollmentPlugin-luokan riippuvuudet. Externals viittaa työssä käytettyihin valmiskirjastoihin.

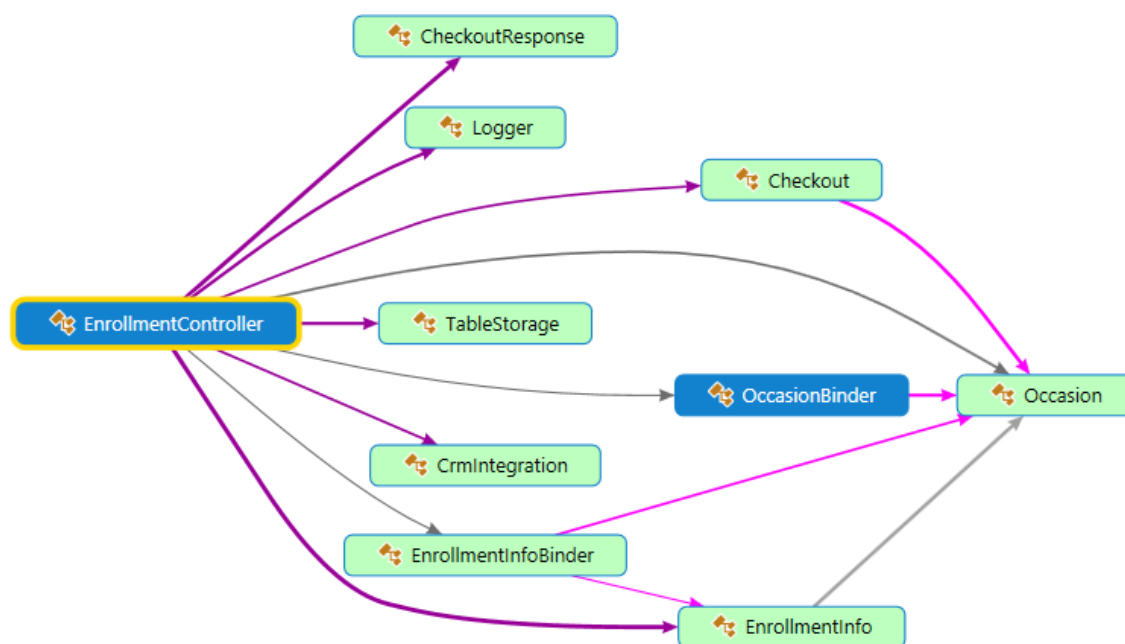
Rijndael on lohkosalausmenetelmä mikä tarkoittaa että se käsittelee yhden merkin sijaan kerrallaan tietyn kokoista lohkoa bittejä. Rijndaelin lohko- ja avainkoko voivat vaihdella 128, 192 ja 256 bitin välillä ja 128 bitin lohkoja käyttävä toteutus tunnetaan myös nimellä Advanced Encryption Standard eli AES (Paar & Pelzl 2010, 89). Tässä työssä käytettiin 128 bitin pituisia lohkoja ja 256 bitin pituisia avainta. Microsoftin .NET-kirjasto tarjoaa valmiin luokan ja metodit Rijndaelin käyttöön, mutta nämä ovat hyvin yleiskäyttöisiä. Työssä tehtäväksi jäi valmiiden metodien yhdistäminen käyttökelpoisiksi kokonaisuudeksi ja kapselointi muiden luokkien käyttöön.

MD5 on tiivistealgoritmi joka tuottaa minkä tahansa pituisesta syötteestä 128-bittisen tiivisteeseen joka voidaan esittää merkkijonona. Tiivisteestä ei voida päätellä alkuperäistä syötettä mitenkään ja pieninkin muutos syötteessä tuottaa täysin erilaisen tiivisteeseen. Tästä syystä MD5 soveltuu hyvin sähköisessä viestinnässä viestin oikeellisuuden tarkistamiseen. Kun viestin mukana lähetetään viestin tiiviste, vastaanottaja voi luoda viestistä uuden tiivisteeseen ja verrata sitä viestin mukana tulleeseen. Mikäli tiivisteet eivät ole identtiset, viesti on muuttunut. Teoriassa on mahdollista löytää loputtomasti erilaisia syötteitä joista tulee sama MD5-tiiviste. Käytännössä tämä on hyvin haastavaa sillä erilaisia mahdollisia MD5-tiivisteitä on 2^{128} eli n. $3,4 \cdot 10^{38}$ erilaista.

Plugin yhdistää tapahtuman tarvittavat tiedot yhdeksi merkkijonoksi erotinmerkkiä käyttäen. Mikäli esimerkiksi tapahtuman nimi sisältää myös tietojen erotukseen varatun merkin, merkki kaksinkertaistetaan. Tällöin myöhemmin merkkijonoa purkava metodi pysyy merkin nähdessään päättelemään onko sen rooli jakaa tietoa vai ei. Kun kaikki tarvittavat tiedot ovat merkkijonossa, merkkijonoon lisätään tiiviste ja merkkijono salataan Rijndael-algoritmillla. Lopuksi merkkijono lisätään ilmoittautumissivuston osoitteen perään ja tuloksena syntyvä linkki laitetaan näkyville tapahtuman lomakkeelle CRM:ään.

4.2 Ilmoittautumissivusto

ASP.NET-sovellusten toimintalogiikka sijoitetaan valmiin Controller-kantaluokan periviin ohjainluokkiin. Nämä käsittelevät dataa ja palauttavat käyttäjälle nähtäväksi erilaisia näkymiä. Näkymät ovat tässä työssä sekä HTML:ää että C#-kieltä sisältäviä CSHTML-sivuja. Ilmoittautumissivuston toimintalogiikka sisältyy kokonaan yhteen ohjaimeen, EnrollmentControlleriin (kuvio 3).



KUVIO 3. EnrollmentController-luokan riippuvuudet. Rajapintojen sijaan esitetään suoraan rajapinnat toteuttavat luokat.

EnrollmentController sisältää julkisia metodeja jotka kukin palauttavat yhden näkymän. Sivulle tullessa kutsuttava metodi riippuu tarkasta käytetystä osoitteesta. Kun sivulle saavutaan ilmoittautumislinkin kautta, EnrollmentController käyttää OccasionBinder-luokkaa purkamaan linkin sisällöstä tapahtuman tiedot jotka tallennetaan luokan Occasion olioon. Tämän jälkeen luokka palauttaa näkymän johon voidaan syöttää ilmoittautujan henkilötiedot.

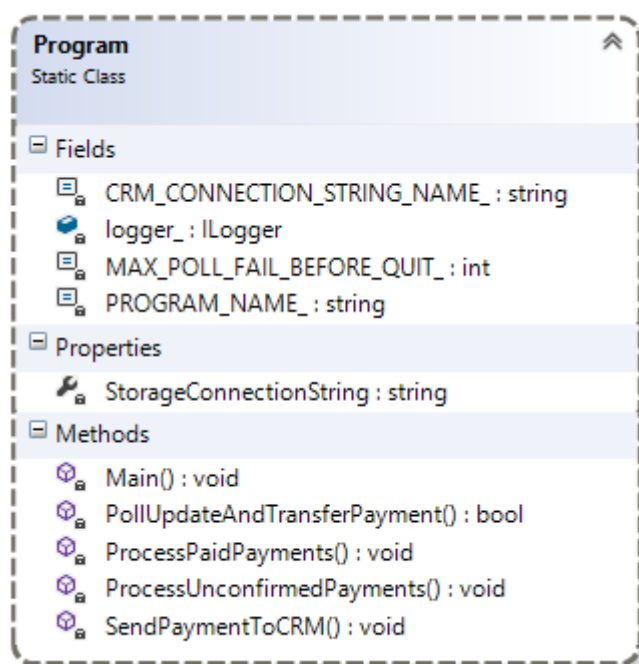
Kun ilmoittautuja on syöttänyt tietonsa, EnrollmentController käyttää EnrollmentInfoBinderiä sitomaan sivulla täytetyn lomakkeen sisällön EnrollmentInfo-luokan olioon. Tämän jälkeen ohjain tallentaa ilmoittautumisen tiedot Azure Table -kantaan TableStorage-luokan kautta ja kutsuu Checkout-luokan avulla käytettävissä olevat maksutavat. Lopuksi luokka palauttaa näkymän jossa esitetään maksutavat kuvallisina linkkeinä jotka johtavat Azuren ulkopuoliselle sivulle maksamaan ilmoittautumisen. Tähän maksutavat käsittelevään ohjainmetodiin päästään pelkästään HTTP POST -kutsulla. Tämä takaa että käyttäjä ei voi päästä näkemään maksusivuja mitenkään kirjoittamalla tietyn osoitteen selaimen osoiteriville.

Käyttäjän palatessa sivulle maksun jälkeen ohjain käsittelee Checkoutin lähettämän XML-muotoisen vastauksen luoden siitä CheckoutResponse-luokan olion. Vastauksen

oikeellisuus tarkastetaan käyttäen sen sisältämää 256-bittistä Secure Hash Algorithm -tiivistettä. Mikäli maksu epäonnistui tai keskeytettiin, ohjain palauttaa tästä kertovan näkymän. Muussa tapauksessa maksun tiedot yritetään lähettää staattisen CrmIntegration-luokan avulla CRM-ohjelmiston tietokantaan. Lopuksi Azure Table -kannassa sijaitsevaan ilmoittautumismerkintään lisätään tieto maksun suorituksesta ja siitä onnistuiko CRM:ään siirto vai ei, ja ohjain palauttaa näkymän jossa kerrotaan onnistuneesta maksusuorituksesta.

4.3 Maksutila-automaatio

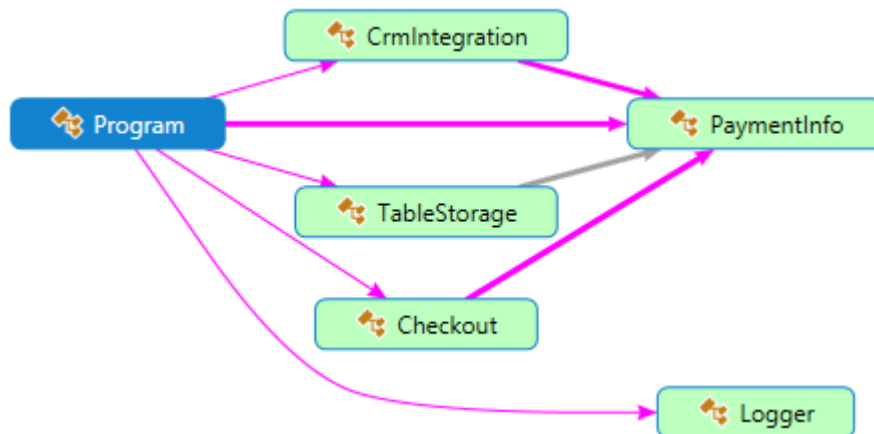
Maksutila-automaatio on komentoriviohjelma jonka toimintalogiikka on sijoitettu nimiavaruudessa PaymentStatusUpdater sijaitsevaan staattiseen luokkaan Program (kuvio 4). Ohjelman ajo alkaa Main-funktiosta joka avaa TableStorage-luokan avulla yhteyden Azure Tableen noutaakseen käsiteltäviä ilmoittautumistietoja.



KUVIO 4. Maksujen tiloja käsittelevän Program-luokan sisältö.

TableStorage palauttaa ohjelmalle käsiteltävien ilmoittautumisten tietoja PaymentInfo-luokan oliona (kuvio 5). Jos ilmoittautumisen maksusta ei ole tietoa, tilannetta kysytään Checkout-luokalta joka ottaa yhteyden Checkout-palveluun. Tämän jälkeen TableStorage

päivittää ilmoittautumisen maksun tilanteen kantaan. Kaikki maksetut maksut lähetetään CrmIntegration-luokan avulla CRM-tietokantaan.



KUVIO 5. Program-luokan riippuvuudet. Rajapintojen sijaan esitetään niitä toteuttavat luokat.

5 KÄYTTÖ

Työssä pyrittiin tekemään ilmoittautumislinkin luonnista ja ilmoittautumissivuston luonnista huomattavan helppoja käyttäjälle jotta käyttäjä ei voisi vahingossa toimia virheellisesti. Kuitenkin vaikka työn tarkoituksena on tulevaisuudessa olla myyntiin kelpaava tuote, työ ei vielä ole sellainen. Tästä syystä luodut moduulit tulevat jatkossa muuttumaan eikä tarkka toiminta todennäköisesti pysy sellaisena kuin se tässä esitetään.

5.1 Tapahtuman ja ilmoittautumislinkin luonti

Yrityksen työntekijä luo tapahtuma CRM:n käyttöliittymän kautta menemällä Myyntiosion tapahtumalistaan ja painamalla painiketta UUSI. Lomakkeella (kuva 2) on täytettävät kentät Nimi, Alkamisaika, Päätymisaika, Sijainti, Osallistumismaksu ja Ilmoittautuminen päättyy. Nimi on ainoa vaadittu tieto. Ilmoittautumislinkki on lukittu kenttä jonka sisältöä käyttäjä ei voi muokata.

Navigation: MYynti | Tapahtumat | Uusi Tapahtuma

Buttons: TALLENNA, TALLENNA JA SULJE, + UUSI, LOMAKE-EDITORI

TAPAHTUMA : TIEDOT

Uusi Tapahtuma

Yleiset

Nimi*	<input type="text"/>
Omistaja*	Etunimi Sukunimi
Alkamisaika	--
Päätymisaika	--
Sijainti	--
Osallistumismaksu	--
Ilmoittautuminen päättyy ⁺	--
Ilmoittautumislinkki	--


KUVA 2. Tyhjä tapahtumalomake CRM-käyttöliittymässä.

Kun työntekijä täyttää lomakkeen tiedot ja tallentaa, plugin käynnistyy ja luo ilmoittautumissivustolle luovan ja tapahtuman tiedot sisältävän linkin (kuva 3). Käyttäjä voi kopioida tämän esimerkiksi sähköpostiin ja lähettää kutsuille. Mikäli jo tallennetun tapahtuman tietoja muokataan lomakkeella, plugin käynnistyy uudelleen ja päivittää linkin.

TAPAHTUMA : TIEDOT

Testitapahtuma

Yleiset

Nimi*	Testitapahtuma
Omistaja*	 Etunimi Sukunimi
Alkamisaika	1.1.2016 16:30
Päättymisaika	2.1.2016 4:00
Sijainti	Turku
Osallistumismaksu	150,00€
Ilmoittautuminen päättyy ⁺	30.12.2015
Ilmoittautumislinkki	 http://mepcoenrollmenttesti.azurewebsites.net

KUVA 3. Tapahtumalomakkeen tiedot tallentamisen jälkeen.

5.2 Tapahtumaan ilmoittautuminen

Kutsutun avatessa ilmoittautumislinkin päätyy hän lomakkeelle johon hänen tulee täyttää tietonsa (kuva 4). Lomake on tällä hetkellä englanninkielinen eikä yhtäkään kenttää ole asetettu vielä pakolliseksi. Lomakkeen täytettyään käyttäjä siirtyy maksutapavalintaan painamalla painiketta Go to payment.

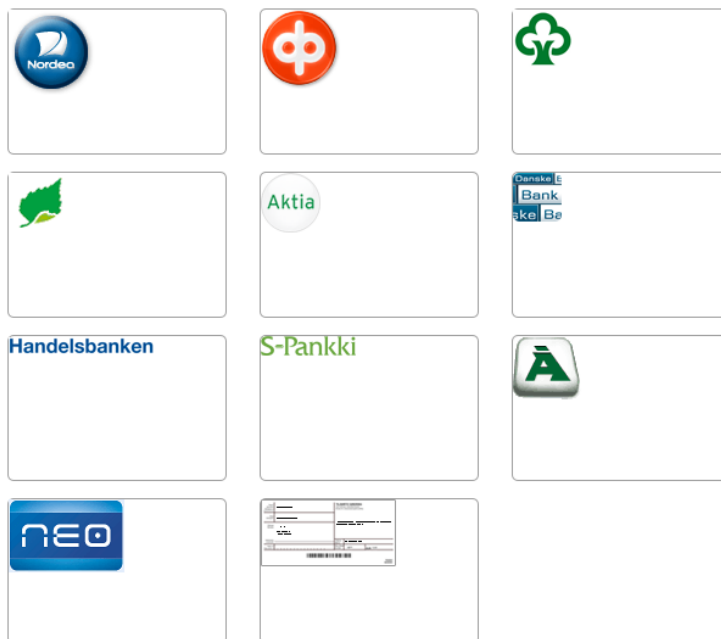
Personal information

First name
 Family name
 Street address
 Postcode
 Post office

KUVA 4. Täytetty henkilötietolomake ilmoittautumissivustolla.

Ilmoittautujan tietojen tallennuksen jälkeen hänelle näytetään maksutavat (kuva 5). Tällä hetkellä näytetyjä maksutapoja on rajallinen määrä sillä osa maksutavoista ei ole käytettävissä Checkoutin ilmaisilla testitunnuksilla. Käyttäjä voi valita haluamansa maksutavan klikkaamalla jotain kuvaketta.

Ilmoittautuminen



KUVA 5. Käyttäjälle näytetyt maksuvaihtoehdot sivulla.

Maksutavan valinnan jälkeen käyttäjä ohjataan maksamaan. Käyttäjä varmistaa henkilöllisyytensä maksutavasta riippuvalla tavalla ja vahvistaa maksun. Esimerkkitapauksessa on käytetty Nordean verkkomaksua (kuva 6).

Maksun tiedot	
Tilille:	FI74 2950 1800 0000 14
Saaja:	Checkout Oy
Viite:	3244 43602
Eräpäivä:	Heti
Määrä:	150,00 euro
Maksaja:	LAURA TESTI
Tililtä (Käytettävissä):	FI41 2950 1800 0205 90 (euro 9.107,45+) ▼
Viesti:	Testi Henkil__ ▲▼
Vahvista antamalla tunnuslukutaulukon alaosassa oleva vahvistustunnus. Vahvistustunnus H : ●●●●	

KUVA 6. Maksun vahvistus Nordean verkkomaksupalvelussa.

Tarkka maksuprosessi riippuu maksutavasta ja esimerkiksi pankista. Kuitenkin kun maksu on vahvistettu, onnistumisesta ilmoitetaan ja käyttäjä voi palata takaisin ilmoittautumissivustolle. Siellä hänelle näytetään viesti maksun onnistumisesta (kuva 7). Viesti on tällä hetkellä mahdollisimman minimalistinen.

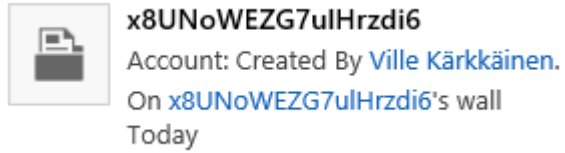
Ilmoittautuminen

OK.

KUVA 7. Vahvistusviesti ilmoittautumissivustolla onnistuneen maksun jälkeen.

5.3 Ilmoittautumisen näkeminen CRM:ssä

Koska ilmoittautumistietojen merkintätapaa ei ole vielä päätetty, ilmoittautumistiedoista tallennetaan CRM:ään ainoastaan ilmoittautumisen ja maksun yksilöivä tunniste. Onnistuneesti maksetun ilmoittautumisen näkee CRM:n käyttöliittymässä siirtymällä asiakaslistaan jonne on luotu uusi tietue (kuva 8) jonka nimenä on ilmoittautumisen tunniste.



KUVA 8. Ilmoitus uuden tietueen luonnista CRM:ssä.

Mikäli ilmoittautumisia on useita, jokaista ilmoittautumista kohti luodaan oma tietue ja jokaisella on eri nimi. Jos ilmoittautumistieto ei maksun jälkeen ole CRM:ssä, se tarkoittaa että siirto ei ole onnistunut. Tällöin siirto jää maksutila-automaation vastuulle.

6 POHDINTA

Työssä onnistuttiin täyttämään alussa asetetut tarpeet hyvin. Haastavin osuus liittyi Checkout-integraatioon sillä palvelun riittämättömän dokumentaation takia suurin osa toiminnasta täytyi selvittää työn aikana testaamalla. Lähtökohdista huolimatta lopputulos toimii luotettavasti.

ASP.NET MVC ja Azure Table –tietokanta osoittautuivat erittäin helppokäyttöisiksi ja tehokkaiksi työvälineiksi web-sovelluksen luontiin. Tietokannan relaatioton NoSQL-periaate sopi tarpeisiin siinä missä relaatiotietokannan tarjoamat lisäominaisuudet olisivat olleet hyödyttömiä. Lisäksi transaktiot relaatiotietokannassa ovat hitaampia.

Koska työn alussa määriteltiin selvästi asiat jotka rajattaisiin työn mittakaavan ulkopuolelle, jatkokehitys olisi selkeintä aloittaa näistä. Myyntikelpoisen tuotteen aikaansaamiseksi täytyy ilmoittautumissivun ulkoasua parantaa, tuki ilmaisille tapahtumille lisätä ja kehittää parempi menetelmä tallentaa ilmoittautumistiedot CRM:ään. Haluttaessa myös ilmoittautumislinkin lähetyksen automatisointi voitaisiin mahdollistaa.

Ilmiselvien kehitystarpeiden jälkeen olisi mahdollista laajentaa kokonaisuus tukemaan esimerkiksi toisia hallintaohjelmistoja ja maksupalveluita. Maksupalvelua käyttävä luokka on jo rajapinnan takana, mutta luokan olion luodakseen joudutaan kuitenkin käyttämään luokan rakentajaa. Mikäli rajapinta eriyttäisiin omaan moduuliinsa, voitaisiin tehdä tehdasluokka jonka kautta voitaisiin luoda maksupalveluolioita ilman että niitä käyttävä luokka riippuu niiden toteutuksesta millään tavalla. Tällöin erilaisten maksupalveluiden lisäys olisi helpompaa. Tuen lisäys erilaisille hallintaohjelmistoille vaatisi enemmän työtä ja saattaisi tämän takia olla taloudellisesti kannattamatonta.

LÄHTEET

Azure Storage Table Design Guide, Microsoft Azure. Viitattu 30.11.2015.

<https://azure.microsoft.com/en-us/documentation/articles/storage-table-design-guide/>

Paar, C & Pelzl, J.2010. Understanding Cryptography: A Textbook for Students and Practitioners. 1. painos. Berliini: Springer.