

Samu Vepsäläinen

# Pilvialustalle toteutettu tekstiviestipalvelu

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Pilvialustalle toteutettu tekstiviestipalvelu

4.4.2015

Tekijä(t) Otsikko	Samu Vepsäläinen Pilivialustalle toteutettu tekstiviestipalvelu
Sivumäärä Aika	38 sivua 4.4.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori Simo Silander Toimitusjohtaja Pekka Nurmela
<p>Opinnäytetyössä toteutettiin verkkoselaimella toimiva pilvipalvelu tekstiviestien lähettämiseen tilaajana toimivalle Positive Productions Oy:lle. Toteutuksessa käytetty Google App Engine -pilvialusta ja tekstiviestipalvelun arkkitehtuuri esitellään osana opinnäytetyötä.</p> <p>Toteutuksen lähtökohtana oli toteuttaa palvelu, jota tilaaja voi myydä verkossa asiakkaille edullisempaan hintaan kuin markkinoilla olevat toimijat. Positive Productionsin asiakkaan kannalta tavoite oli saavuttaa kustannussäästöjä palvelulla, joka ei kilpailijoista poiketen sisällä kiinteitä aloitus- tai kuukausimaksuja. Tavoitteenasettelu tehtiin yhdessä tilaajan kanssa.</p> <p>Opinnäytetyön lopputuloksena toteutettiin tekstiviestipalvelu. Positive Productions testaa tekstiviestipalvelua yhdessä asiakasyrityksen kanssa ennen palvelun markkinoinnin aloittamista. Asiakasyritys käyttää palvelua valmiiden töiden noutoilmoitusten lähettämiseen omille asiakkailleen.</p> <p>Tekstiviestipalvelua käytetään verkkoselaimella. Käyttöönotto, käyttö ja käytön maksaminen on mahdollista tehdä suoraan selaimessa. Palveluun integroidun verkkomaksamisen välityksellä loppukäyttäjä maksaa palvelun käytön suoraan selaimessa. Toteutettu ratkaisu minimoi palvelun asiakkuuksien ylläpitoon tarvittavan ajan.</p> <p>Opinnäytetyön kokemukset osoittavat, että Google App Engine -pilvialusta tarjoaa ohjelmistokehittäjälle ympäristön, joka nopeuttaa verkkopalvelun toteutusta perinteisiin virtuaalipalvelinympäristöön toteutettaviin ratkaisuihin nähden. Pilvipalvelujen kattavat ominaisuudet ja hinnoittelumalli nopeuttavat pilvipalvelujen yleistymistä.</p>	
Avainsanat	tekstiviestipalvelu, java, pilvialusta, google app engine

Author(s) Title	Samu Vepsäläinen Text Message Service Built on Cloud Platform
Number of Pages Date	38 pages 4 April 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Simo Silander, Principal Lecturer Pekka Nurmela, Managing Director
<p>For this thesis, a text message service operating on a cloud platform was built for Positive Productions Ltd. The Google App Engine cloud platform and the complete construction of the text message service is presented here.</p> <p>The core idea of this final project was to execute a service which Positive Productions can sell to its clients with more reasonable pricing than other actors in the field. In this specific case the client was looking for a cost efficient service that they could offer without fixed starting or monthly fees, unlike competing businesses at the moment. The project goals were set in co-operation with the client.</p> <p>In the end a text message service was built. Positive productions is currently testing the text message service with a client before starting to promote this service. The client is using the service in order to make pickup notifications to its own clients.</p> <p>The text message service is used on a web browser. Initialization, the use and paying for the use is possible directly on the browser. Through an online payment method integrated into the service the client is able to pay for the use directly on the browser. This solution minimizes the workload used for upkeeping the clientship for Positive Productions.</p> <p>This thesis shows that the Google App Engine cloud service offers to software engineers an environment that enables them to build online services faster compared to solutions built on traditional virtual environments. Today the cloud services are being utilized more because of their comprehensive features and pricing model.</p>	
Keywords	Text message service, Java, cloud platform, Google App Engine

## Sisällysluettelo

1	Johdanto	1
2	Tekstiviestipalvelu	2
2.1	Vaatimukset	2
2.2	Käyttötarkoitus	3
2.3	Ominaisuudet	4
3	Google Cloud	7
3.1	Google App Engine	8
3.2	Google Cloud Datastore	8
3.3	Google App Engine vrt virtuaalipalvelin	9
4	Tekstiviestipalvelun tekninen toteutus	10
4.1	Arkkitehtuuri	10
4.1.1	Malli	11
4.1.2	Näkymä	13
4.1.3	Kontrolleri	16
4.2	Integraatiot	17
4.2.1	Paytrail	17
4.2.2	Clickatell SMS-Gateway	23
4.3	Tekstiviestipalvelun toteuttaminen ja julkaisu	23
4.3.1	Projektin luonti Google App Engineen	24
4.3.2	Google App Engine SDK:n asennus	25
4.3.3	Projektin luonti	26
4.3.4	Viestin lähetys	29
4.3.5	Julkaisu Google App Engineen	33
5	Yhteenveto	36
	Lähteet	37

## Lyhenteet

ACID	Atomicity, Consistency, Isolation, Durability. Tietokantajärjestelmien periaate, jolla turvataan järjestelmän eheys kaikissa tilanteissa.
CSS	Cascading Style Sheets. WWW-dokumenteille kehitetty tyyliohjeiden laji.
DAO	Data access object. Luokka, joka tarjoaa rajapinnan tietokannan käytölle.
HTML	Hypertext Markup Language. Kuvauskieli, jota käytetään nettisivujen ohjelmointiin.
HTTP	Hypertext Transfer Protocol. Protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon.
JDO	Java Data Objects. Ohjelmointirajapinta, jonka välityksellä tietokantaan tallennettavaa tietoa voidaan käsitellä Java-olioina.
JSON	JavaScript Object Notation. JavaScriptistä riippumaton avoimen standardin mukainen tiedon esitysmuoto tiedonvälitykseen.
MVC	Model-view-controller. Ohjelmistoarkkitehtuurityyli, joka erottaa käyttöliittymän sovelluksen sovellusalueesta.
NoSQL	Not only SQL. Käytetään kuvaamaan perinteisestä relaatiomallista poikkeavia tietokantoja.
REST	Representational State Transfer. HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintojen toteuttamiseen.
SaaS	Software as a Service. Internet-selaimella käytettävä ohjelmisto, joka ei käyttöönottaessa vaadi asennustyötä ja jonka hinnoittelu on käyttöperusteinen.
SQL	Structured Query Language. Kyselykieli, jolla voidaan tehdä erilaisia operaatioita relaatiotietokantaan.

URL Uniform Resource Locator. Käytetään osoittamaan WWW-sivuja.

## 1 Johdanto

Tekstiviesti on varma tapa tavoittaa vastaanottajat. Tarve opinnäytetyössä esiteltävän tekstiviestipalvelun toteuttamiseen tuli tilaajana toimivan Positive Productions Oy:n asiakkaan kautta. Positive Productionsin asiakas oli etsinyt verkosta ratkaisua, jossa viestejä voi lähettää käyttöperusteisesti ilman kuukausimaksua tai muuta jatkuvaa kuluja. Käyttöperusteisessa hinnoittelussa ei ole aloitus- tai kuukausimaksuja, vaan käyttäjä maksaa vain lähetetyistä viesteistä. Päätös palvelun toteuttamiseen syntyi, koska verkosta ei löytynyt asiakkaan etsimää ratkaisua.

Tekstiviestipalvelun toiminnallinen tavoite on olla helppokäyttöisin, verkkoselaimella käytettävä, ajastettujen ja ryhmätekstiviestien lähetykseen sopiva ratkaisu. Käyttöönotto, viestien lähetykset ja käytön maksaminen tehdään helpoksi yksinkertaisen käyttöliittymän avulla, sekä osaksi palvelua toteutettavan verkkomaksun avulla. Palvelu toteutetaan ohjelmoimalla ja asennetaan ylläpitoon Google App Engine -pilvialustalle.

Opinnäytetyössä esitellään tekstiviestipalvelu ja toteutuksessa käytetyt teknologiat. Lisäksi työssä verrataan Google App Engine (GAE) -pilvialustan ja virtuaalipalvelimen käyttöä sovelluskehityksessä. Opinnäytetyö antaa lukijalle tietoa pilvipalveluista ja Software as a Service (SaaS) -palveluiden ohjelmoinnista Googlen alustan päälle. SaaS-palvelulla tarkoitetaan ohjelmistoa, joka ei käyttöönotettaessa vaadi asennustyötä ja jonka hinnoittelu on käyttöperusteinen ja sitä käytetään internetselaimella.

Opinnäytetyö toteutetaan tilaajana toimivalle Positive Productions Oy:lle. Positive Productions Oy on ohjelmistoyritys, joka tuottaa web-, mobiili- ja pilvipalveluita. Yritys innovoi ja kehittää ratkaisuja sähköiseen liiketoimintaan. Liiketoiminta koostuu ohjelmistoprojekteista, tuntilaskutettavasta kehitys- ja konsultointityöstä, tuki ja ylläpitomaksuista sekä SaaS-palveluista. Osana Positive Productionsin strategiaa on toteuttaa ja myydä palveluita, joissa käytön ja käyttöönoton helppous on avainasemassa. Opinnäytetyönä toteutettava tekstiviestipalvelu on Positive Productionsin uusi SaaS-palvelu. Positive Productions myy palvelua verkossa.

Työ on osa insinööritutkintoa, se toteutettiin Positive Productions Oy:lle. Työn ohjaajana toimi Metropolia Ammattikorkeakoulusta lehtori Simo Silander sekä Positive Productions Oy:stä toimitusjohtaja Pekka Nurmela.

## 2 Tekstiviestipalvelu

Tekstiviestipalvelu on verkkopalvelu, jonka välityksellä käyttäjät lähettävät tekstiviestejä verkkoselaimesta tietokoneella tai mobiililaitteella (kuva 1).

The screenshot shows the 'Tekstiviestipalvelu' web interface. At the top, it says '100 viestiä jäljellä. Osta lisää viestejä.' and 'Kirjaudu ulos'. The interface is divided into two main sections: a form for sending a message on the left and a list of sent messages on the right.

**Form for sending a message (Lähetä viesti):**

- Input field: Kirjoita viesti
- Text: Tero merkitsevää
- Input field: Lisää vastaanottajan nimi
- Input field: Lisää puhelinnumero
- Text: Esim. +358401393290 tai 0401353290
- Button: Lähetä
- Button: Tallenna luonnokseksi

**List of sent messages (Viestit):**

Viestit	Asiakkaat	Tilin hallinta	Luonnokset	Ajastetut	Lähetetyt	Arkistoidut
<b>Lähetetty</b>	22.03.2015 14:08	Tero Matilainen	0408097499			
Teille on varattu aika hammaslääkäriin huomenna maanantaina 23.3.2015 klo 12:45. Tervetuloa! T: Jani Makkonen, Hammasklinikka						
<b>Lähetetty</b>	22.03.2015 14:07	Janne Huuskonen	0401119934			
Teille on varattu aika hammaslääkäriin huomenna maanantaina 23.3.2015 klo 12:00. Tervetuloa! T: Jani Makkonen, Hammasklinikka						
<b>Lähetetty</b>	22.03.2015 14:07	Mari Turunen	0509337733			
Teille on varattu aika hammaslääkäriin huomenna maanantaina 23.3.2015 klo 10:30. Tervetuloa! T: Jani Makkonen, Hammasklinikka						
<b>Lähetetty</b>	22.03.2015 14:06	Aune Pentikäinen	0443887377			
Teille on varattu aika hammaslääkäriin huomenna maanantaina 23.3.2015 klo 09:45. Tervetuloa! T: Jani Makkonen, Hammasklinikka						
<b>Lähetetty</b>	22.03.2015 14:06	Jari-Pekka Aaltonen	0501403955			
Teille on varattu aika hammaslääkäriin huomenna maanantaina 23.3.2015 klo 09:00. Tervetuloa! T: Jani Makkonen, Hammasklinikka						

Kuva 1. Tekstiviestipalvelu

Palvelun käyttöönotto on asiakkaalle ilmaista, käyttäjä maksaa vain lähetetyistä viesteistä. Käyttöönotto tapahtuu rekisteröitymällä ilmaiseksi verkkopalveluun. Tekstiviestien lähetystä varten käyttäjä ostaa lähetettäviä viestejä verkkopankkimaksulla tai luottokortilla. Käytön maksaminen suoritetaan palveluun toteutetun verkkomaksamisen kautta.

### 2.1 Vaatimukset

Tekstiviestipalvelun toteutuksen lähtökohtana on ollut vaatimus toteuttaa kuukausimaksuton ratkaisu, jossa käyttöönotto ja käyttö on mahdollisimman helppoa. Käyttäjän tulee voida ottaa palvelu käyttöön verkkosivulla olevan rekisteröintilomakkeen ja ohjeiden perusteella ilman henkilökohtaista opastusta.

Lomakkeelle käyttäjä syöttää omat tietonsa ohjeiden perusteella, jonka jälkeen hänet kirjataan automaattisesti sisään. Viestien lähetystä varten toteutetaan tietokoneen ja



mobiililaitteen verkkoselaimella toimiva käyttöliittymä. Käytön maksaminen tehdään palveluun sisältyvän verkkomaksun avulla.

## 2.2 Käyttötarkoitus

Palvelun ensisijaisia käyttäjäkohderyhmiä ovat erilaiset yritykset ja yhteisöt. Alle on listattu esimerkkitapauksia, joihin tekstiviestipalvelu soveltuu.

- Ilmoitus viime hetken muutoksista tapahtumassa.
  - "Hei koulutuspaikka on vaihtunut. Nähdään auditoriossa klo 12:00. T: Martti Pitkänen, Excel-kouluttaja"
- Ilmoitus asiakkaalle noudettavissa olevasta tilauksesta.
  - "Hei, uudet verhot on valmiina. Valmiin työn voit noutaa ompelimosta Ma-Pe klo 12-18. Ystävällisin terveisin Susanna, Korjausompelu.net"
- Ajastettu muistutus sovitusta tapaamisesta.
  - "Teille on varattu aika hammaslääkäriin huomenna maanantaina 23.3.2015 klo 09.00. Tervetuloa! T: Jani Makkonen, Hammasklinikka"
- Vikailmoitus
  - "Hei, asiakas Jantunen osoitteessa Runeberginkatu 15 A 33 ilmoitti, että patterit eivät lämpiä. Voitteko käydä ilmaamassa patterit. T: Isännöitsijä Rahikainen"
- Tilannetiedotus
  - "Hyvä Asiakas, lento AY1923 Malagaan on myöhässä ja lähtee klo 23:00. Ystävällisin terveisin, Aurinkomatkat"

## 2.3 Ominaisuudet

Tekstiviestipalvelun ominaisuudet mahdollistavat lähetettävien tekstiviestien, asiakasrekisterin sekä omien tietojen ja tilausten hallinnan.

### Tekstiviestin lähetys

Tekstiviestipalvelussa viestin lähetys tapahtuu kirjoittamalla viesti, vastaanottajan nimi/numero ja painetaan Lähetä-painiketta. Lähetä-painikkeen painalluksen jälkeen palvelu lähettää ja tallentaa viestin. Jos vastaanottaja löytyy jo asiakasrekisteristä, palvelu ehdottaa olemassa olevan yhteystiedon valintaa. Jos numeroa vastaavaa yhteystietoa ei löydy, palvelu tallentaa uuden yhteystiedon automaattisesti asiakasrekisteriin.

Viestin vastaanottajat voi vaihtoehtoisesti valita asiakasrekisteristä lisäämällä yhteystietoja yksi kerrallaan tai valitsemalla kaikki asiakasryhmän yhteystiedot. Viesti ja siihen liittyvät vastaanottajat on mahdollista tallentaa luonnokseksi. Luonnos lähetetään vasta, kun käyttäjä käy painamassa Lähetä-painiketta.

### Ryhmäviesti

Ryhmäviestin avulla käyttäjä lähettää viestin usealle vastaanottajalle. Ryhmäviesti lähetetään lisäämällä viestiin useita vastaanottajia yksitellen tai valitsemalla asiakasryhmä. Asiakasryhmän valinta lisää ryhmässä olevat yhteystiedot viestin vastaanottajiksi.

Tekstiviestipalvelun asiakasryhmien hallinnassa käyttäjä luo, muokkaa ja poistaa ryhmiä sekä hallinnoi ryhmien yhteystietoja.

### Lähetyksen ajastaminen

Ajastus-toiminnon avulla käyttäjä ajastaa viestien lähetyksen. Ajastus tapahtuu syöttämällä viestin kirjoituksen yhteydessä lähetysajankohta ja tallentamalla viesti odottamaan lähetystä. Ajastetut viestit on mahdollista suodattaa käyttöliittymässä omaan näkymään. Palveluun toteutettu ajastustoiminto huolehtii, että ajastetut viestit lähetetään automaattisesti käyttäjän valitsemana lähetysajankohtana.

## Viestipohjat

Viestipohjien avulla tekstiviestipalvelun käyttäjä luo uuden viestin tallennetusta pohjasta. Uuden viestin lähetyksessä valitun viestipohjan teksti kopioituu viestin kirjoituskenttään. Tekstin kopioinnin jälkeen käyttäjän on mahdollista muokata viestiä ennen lähetystä. Viestipohjien hallinnassa käyttäjä voi tallentaa, muokkaa ja poistaa viestipohjia.

## Arkistointi

Arkistoidut viestit näkyvät käyttöliittymässä omassa näkymässä. Siirtämällä vanhat viestit arkistoon käyttäjä pitää lähetettyjen viestien listaa lyhyempänä. Toiminnon tarkoituksena on selkeyttää ajankohtaisten lähetysten käsittelyä.

## Asiakasrekisteri

Tekstiviestipalvelun asiakasrekisterin avulla käyttäjä hallitsee omia yhteystietoja (kuva 2). Lisäksi asiakasrekisteri sisältää asiakasryhmä toiminnon yhteystietojen ryhmiin jakamista varten.

Viestit Asiakkaat Tilin hallinta		
Haku		
Jari-Pekka Aaltonen	0501403955	<a href="#">✉</a> <a href="#">🗑</a>
Janne Huuskonen	0401119934	<a href="#">✉</a> <a href="#">🗑</a>
Aune Pentikäinen	0443887377	<a href="#">✉</a> <a href="#">🗑</a>
Mari Turunen	0509337733	<a href="#">✉</a> <a href="#">🗑</a>
Tero Matilainen	0408097499	<a href="#">✉</a> <a href="#">🗑</a>

Kuva 2. Asiakasrekisteri

Asiakasrekisteriin tallennetaan seuraavat tiedot yhteystiedosta:

- etunimi
- sukunimi
- sähköpostiosoite

- matkapuhelinnumero
- lisätiedot.

Yhteystietojen tallentaminen rekisteriin ei ole pakollista.

### Tilin hallinta ja verkkomaksut

Tekstiviestipalveluun on toteutettu Tilin hallinta ja viestien ostaminen verkkomaksujen avulla (kuva 3). Tilin hallinnassa käyttäjä muokkaa omia yhteystietojaan, selaa omia ostojaan tilaushistoriassa sekä ostaa uusia lähetettäviä viestejä verkkomaksun avulla.

Viestit Asiakkaat **Tilin hallinta**

99 viestiä jäljellä  
Osta lisää viestejä

**Valitse summa ja seuraa ohjeita**

10 € 20 € 50 €

**Hinnat**  
100 viestiä - 10 €  
220 viestiä - 20 €  
600 viestiä - 50 €

**Tilaushistoria**

22.03.2015	10 €	100 viestiä
------------	------	-------------

**Omat tiedot**

Samu

Vepsäläinen

Yrityksen tai organisaation nimi

samu.vepsalainen@positiveproductions.fi

0503276827

Salasana

Salasana uudelleen

Tallenna

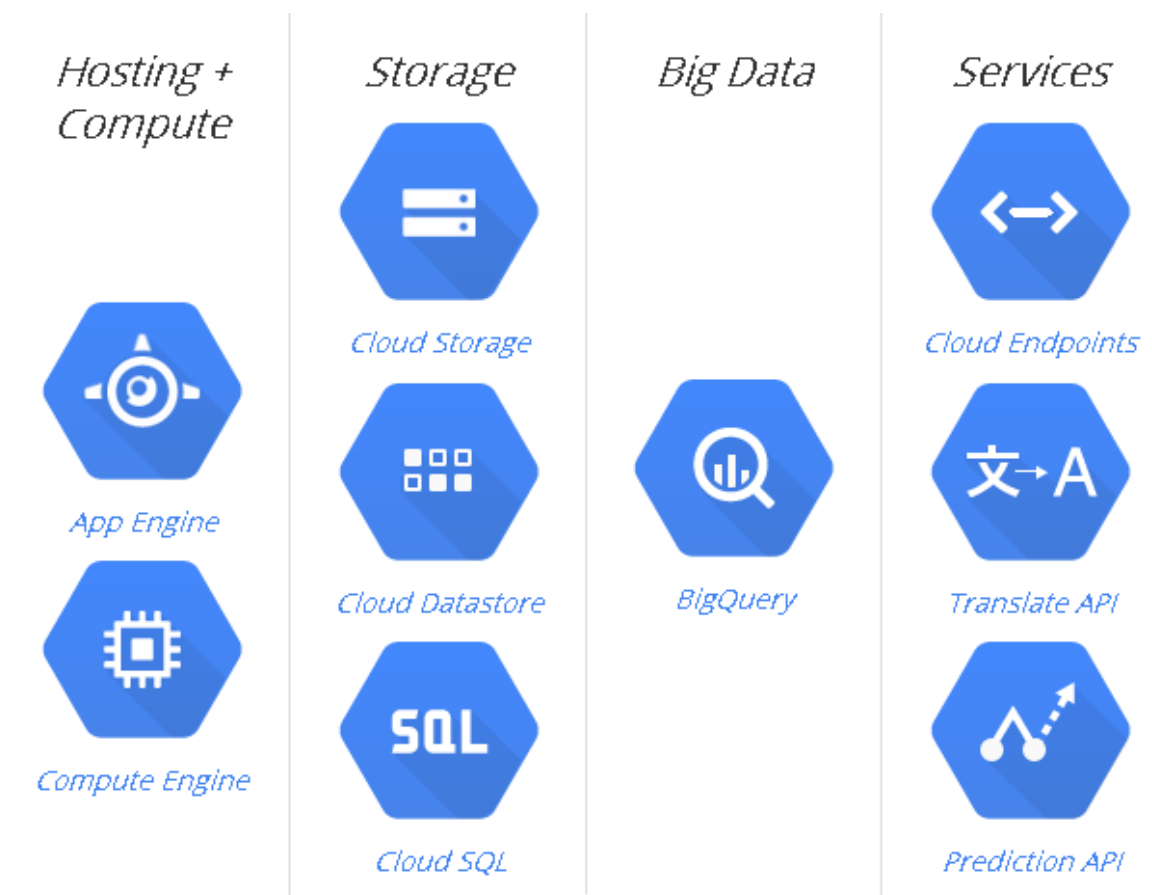
Kuva 3. Tilin hallinta

Viestien ostaminen tapahtuu siten, että käyttäjä valitsee haluamansa summan ja maksutavan, esim. 10 €, Osuuspankin verkkomaksu siirtyy maksutavan tarjoajan verkkopalveluun ja maksaa summan. Maksun jälkeen käyttäjä palaa takaisin tekstiviestipalvelun Tilin hallintaan. Ostetut viestit tulevat käyttöön ja maksetut ostokset näkyvät tilaushistoriassa. Käytettävissä olevien viestien määrä näkyy reaaliaikaisesti käyttöliittymässä.

### 3 Google Cloud

Tekstiviestipalvelu on toteutettu ohjelmoimalla Google App Engine -pilvialustan päälle Java-ohjelmointikielellä. Palvelun tietokannaksi on valittu Google Cloud Datastore. App Engine ja Cloud Datastore ovat osa Google Cloud -ympäristöä (kuva 4).

Google on kehittänyt useita suosittuja palveluita, joista useimmat käsittelevät suuria määriä tietoa globaalisti jopa miljoonille yhtäaikaistilaisille käyttäjille. Google Cloud -alusta avaa Googlen omien palveluiden taustalle kehitetyn teknologian avoimesti kehittäjien käytettäväksi. Google Cloud -ympäristön tarjotessa pitkälle kehittyneet työkalut, kehittäjät voivat keskittyä oman palvelun konseptin ja asiakaskokemuksen toteuttamiseen.[1.]



Kuva 4. Google Cloud [1]

Google Cloud -ympäristön tuotteista ja työkaluista on mahdollista yhdistellä tarvittava kokonaisuus kehitettävän pilvipalvelun luonteesta riippuen. Google laskuttaa

käytetyistä resursseista, alusta skaalautuu tehokkaasti käyttöasteen mukaisesti ja tarjoaa nopeat vasteajat palveluiden käyttäjille ympäri maailman.

Opinnäytetyössä Google Cloud -tuotteista on käytössä App Engine ja Cloud Datastore. Muita Google Cloud -tuotteita ei käsitellä tässä työssä.

### 3.1 Google App Engine

Google App Engine -pilvialusta mahdollistaa sovellusten ajon Google Cloud -ympäristössä. Google App Engine -alustan päälle voidaan toteuttaa ohjelmistoja Javan lisäksi Python-, PHP- ja GO -ohjelmointikielillä.

Google App Engine SDK:n avulla kehittäjät toteuttavat sovellukset ja palvelut Google App Engine -yhteensopiviksi sekä julkaisevat ne Googlen ympäristöön. SDK sisältää ohjelmistorajapinnat muiden Google Cloud -tuotteiden, kuten Google Cloud Datastore -tietokannan, käyttöön.[2.]

### 3.2 Google Cloud Datastore

Google Cloud Datastore on hallittu NoSQL-tietokanta, joka on suunniteltu tiedon tallentamiseen. Google Cloud Datastore skaalautuu käytön mukaan ja tukee transaktioita ja SQL:n kaltaisia kyselyitä.[3.]

Google Cloud Datastore huolehtii tietokannan rakenteesta ja tarjoaa kyselymoottorin tiedon hakuun ja järjestämiseen. Tietokannan tiedot on hajautettu ja monistettu useille palvelimille, joka varmistaa tiedon säilymisen ja saatavuuden.

NoSQL-tietokannat ovat perinteisestä relaatiomallista poikkeavia tietokantoja, jotka eivät seuraa mitään kiinteästi määriteltyä rakennetta ja sen ansiosta skaalautuvat horisontaalisesti. NoSQL-tietokanta pystyy käsittelemään paljon samanaikaisia luku- ja kirjoitusoperaatioita.[4.]

Cloud Datastore tukee ACID-transaktioita, jonka avulla tietokanta voi suorittaa useita tietokantaoperaatioita yhden transaktion sisällä. Tämän avulla kaikki transaktion sisällä

suoritetut operaatiot joko onnistuvat tai epäonnistuvat, mikä varmistaa tiedon eheyden.[5.]

### 3.3 Google App Engine vrt virtuaalipalvelin

Perinteisesti verkkopalveluita on kehitetty vuokraamalla virtuaalipalvelin, jossa kehittäjä vastaa verkkopalvelun ajoon vaadittavien palvelinohjelmistojen asennuksesta, konfiguroinnista ja päivityksistä. Palvelinohjelmistoja ovat esimerkiksi sovelluspalvelin, tietokantapalvelin ja sähköpostipalvelin.

Virtualisointi on tekniikka, joka mahdollistaa yksittäisen palvelimen jakamisen useisiin toisistaan eristettyihin virtuaalisiin palvelimiin, joita kutsutaan virtuaalipalvelimiksi. Yksi palvelin voi olla jaettu useiden asiakkaiden kesken. Virtuaalipalvelimia vuokraavat palveluntarjoajat vastaavat fyysisen palvelimen ylläpidosta, verkkoyhteydestä ja virtualisoinnista. Useimmiten palveluntarjoajat vastaavat myös virtuaalipalvelimen käyttöjärjestelmän asennuksesta. Vuokraaja käyttää virtuaalipalvelinympäristöä etäyhteyden välityksellä.

Google App Engine -alustaa käytettäessä kehittäjien ei tarvitse itse vastata ympäristön valvonnasta, turvallisuudesta ja huollosta. Virtuaalipalvelimille tyypilliset tehtävät ympäristön päivitysten, konfiguroinnin ja tietoturvan ylläpitämisen osalta jäävät Googlen vastuulle. Luotettavuus, vasteaikojen nopeus ja mukautuminen palvelun todelliseen käyttömäärään ovat Google App Engine -alustan etuja verrattuna aiemmin yleisesti käytettyihin hosting- ja palveluratkaisuihin.

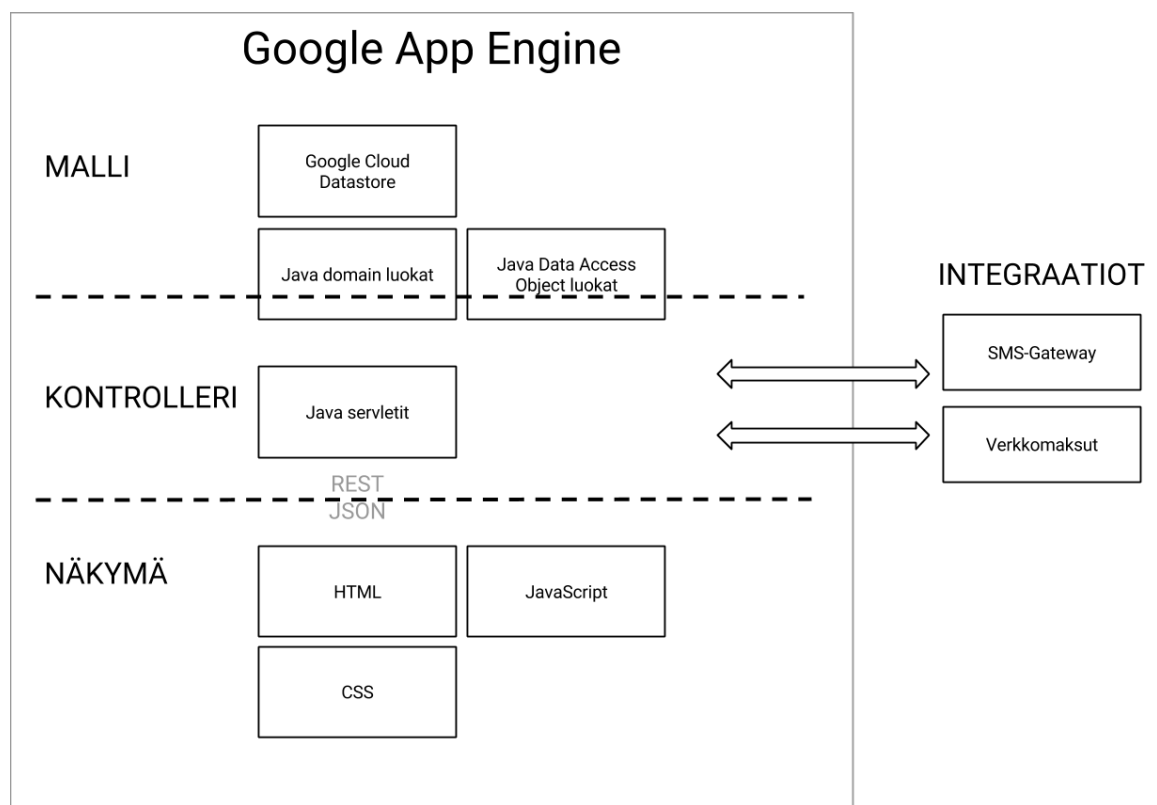
Perinteisen virtuaalipalvelinympäristön ongelma on, että joudutaan ostamaan lisää palvelimia, kun sivuston liikenne kasvaa. Kun liikenne laskee, ostetuista palvelimista ei ole hyötyä. Google App Engine -alustassa ei ole tätä ongelmaa, vaan maksaminen tapahtuu käytön mukaan ja palvelimen teho skaalautuu tarpeen mukaan eikä tarvitse lisätä tai poistaa uusia palvelimia.[6.]

## 4 Tekstiviestipalvelun tekninen toteutus

### 4.1 Arkkitehtuuri

Tekstiviestipalvelu on toteutettu noudattamaan MVC-arkkitehtuuria. MVC-arkkitehtuuri tarkoittaa ohjelmistoarkkitehtuuryliä, jossa erotetaan käyttöliittymä sovelluksen tiedonkäsittelyn toimintalogiikasta ja tietokannasta.[7.]

MVC-arkkitehtuuri koostuu kolmesta alueesta, jotka ovat malli, näkymä ja kontrolleri. Malli toteuttaa sovelluksen tietokannan, näkymä sovelluksen käyttöliittymän ja kontrolleri huolehtii tiedonvälityksestä mallin ja näkymän välillä (kuva 5).



Kuva 5. Tekstiviestipalvelun arkkitehtuuri

Arkkitehtuurin keskeisin etu on, että sen jokainen osa-alue voidaan toteuttaa uudestaan hyvin vähäisillä muutoksilla. Arkkitehtuurin muita etuja on, että malli on täysin riippumaton näkymästä ja kontrollerista. Käyttöliittymän muuttaminen on helppoa, koska kontrollerin ja näkymän riippuvuus toisistaan on hyvin vähäinen.



#### 4.1.1 Malli

Malli on sovelluksen tiedon käsittelyyn liittyvä kokonaisuus, joka pitää huolen tiedon tallennuksesta ja hakemisesta. Sovelluksen malli koostuu Google Cloud Datastore -tietokannasta, Java Data Objects (JDO) -rajapinnasta, Java Domain sekä Data Access Object (DAO) -luokista. Java Domain -luokiksi kutsutaan luokkia, joita käytetään tietokantaan tallennettavan tiedon käsittelyyn. DAO-luokat tarjoavat rajapinnan tietokannan käyttöön. Kuvassa 6 on esitetty osia tekstiviestipalvelun tietokantarakenteen toteuttavista Java Domain -luokista.



Kuva 6. Tekstiviestipalvelun Java Domain -luokat

Yhteys sovelluksen ja tietokannan välille muodostetaan JDO-rajapinnan välityksellä. JDO on ohjelmointirajapinta, jonka välityksellä tietokantaan tallennettavaa tietoa voidaan käsitellä Java-olioina. JDO muuntaa olioissa olevan tiedon automaattisesti tietokantaan tallennettavaan muotoon sekä muuntaa tietokannasta haettavan tiedon takaisin oliomuotoon.[8.]

Sovelluskoodissa JDO otetaan käyttöön luomalla instanssi PersistenceManagerFactory-luokasta (kuva 7). PM:n instanssi luodaan PMF:n avulla BaseDAOImpl-luokassa ja sitä käytetään tietokantayhteyden muodostamiseen.

```

public final class PMF {
    private static final PersistenceManagerFactory pmfInstance =
        JDOHelper.getPersistenceManagerFactory("transactions-optional");

    private PMF() {}

    public static PersistenceManagerFactory get() {
        return pmfInstance;
    }
}

```

Kuva 7. PersistenceManagerFactory-instanssin luonti

Tietokantaan tallennettavat tiedot määritetään sovelluskoodissa Java Domain -luokkien avulla. JDO Java -annotaatioiden avulla oliot merkitään tietokantaan tallennettaviksi. Kyselyiden avulla JDO mahdollistaa tiedon hakemisen tietokannasta ja muuttamisen oliomuotoon. @PersistenceCapable-annotaatiolla luokka määritetään Java Domain -luokaksi. @Persistent-annotaatiolla määritetään kentät, joiden arvot halutaan tallentaa tietokantaan. Yrityksen tietojen tallentamista ja käsittelyä varten on tekstiviestisovellukseen luotu Company Java Domain -luokka, jossa määritetään tallennettavaksi tiedoiksi yrityksen uniikki tunnus (key), luomispäivämäärä (created), muokkauspäivämäärä ja nimi (kuva 8).

```

@PersistenceCapable(identityType = IdentityType.APPLICATION)
public class Company {

    @PrimaryKey
    @Persistent(valueStrategy = IdGeneratorStrategy.IDENTITY)
    private Key key;

    @Persistent
    private Date created;

    @Persistent
    private Date updated;

    @Persistent
    private String name;
}

```

Kuva 8. Company Java Domain -luokka

Tiedon käsittely tapahtuu Java Domain -luokista luoduilla olioilla. Tiedon lisäys, muokkaus, haku ja poistaminen tietokannasta hoidetaan kutsumalla DAO-rajapinnan metodeja (kuva 9).[9.]

```

public interface CompanyDAO {

    List<Company> findCompanies();

    Company getCompany(Long id);

    void deleteCompany(Long id);

    Company updateCompany(Company company);

    Company saveCompany(Company company);

}

```

Kuva 9. CompanyDAO-rajapinta

DAO-metodissa avataan yhteys tietokantaan PersistenceManagerin (pm, peritty kantaluokasta) avulla ja tehdään haluttu toimenpide. Sovelluskoodissa Company-oliossa olevat yrityksen nimi ja muokkauspäivämäärä voidaan tallentaa kutsumalla updateCompany-metodia (kuva 10). Metodi hakee tiedot uniikin tunnuksen perusteella ja päivittää tiedot.

```

public class CompanyDAOImpl extends BaseDAOImpl implements CompanyDAO {

    @Override
    public Company updateCompany(Company company) {
        log.info("update company");

        Company updatedCompany = null;
        try{
            Date now = DateUtils.now();
            updatedCompany = pm.getObjectById(Company.class, company.getKey());
            updatedCompany.setUpdated(now);
            updatedCompany.setName(company.getName());

        }finally{
            pm.close();
        }
        return updatedCompany;
    }
    .
    .
    .
}

```

Kuva 10. updateCompany-metodi

#### 4.1.2 Näkymä

Näkymä määrittää käyttöliittymän ulkoasun ja tiedon näyttämisen sovelluksen käyttöliittymässä. Näkymä on toteutettu käyttöliittymässä HTML (Hypertext Markup

Language), CSS (Cascading Style Sheets) ja JavaScript-teknologioiden avulla. HTML-tiedosto kertoo selaimelle, miten sivun rakenne pitää muodostaa.

Ohjelmistokoodin HTML-tiedostossa kerrotaan, mitkä CSS- ja JavaScript-tiedostot ovat käytössä. Kuvassa 11 esimerkki CSS- ja Javascript-tiedostojen merkkeamisestä HTML:ssä.

```
<link href="/css/admin.css" rel="StyleSheet" type="text/css" />  
  
<script src="/js/app/views/MessagesView.js"></script>
```

Kuva 11. CSS- ja JavaScript-tiedostojen merkkeäminen HTML:ssä

Tekstiviestipalvelu näkymästä on tehty responsiivinen. Responsiivisuus tarkoittaa, että palvelua voidaan käyttää kaiken kokoisissa laitteissa ja näytöissä. Ulkoasun asettelu muuttuu käytettävän laitteen näytön koon mukaan. Responsiivisuuden saamiseksi apuna on käytetty Bootstrap-kirjastoa. Bootstrap tarjoaa rakenteen responsiiviseen käyttöliittymä kehittämiseen. Bootstrapin käyttöönotto tapahtuu lataamalla kirjastot kehittäjän verkkosivulta ja lisäämällä ne projektiin.[10.]

Kuvat 12 ja 13 havainnollistavat, miten responsiivisuus toteutuu käytännössä. Tekstiviestipalvelun etusivu kannettavan tietokoneen selaimessa (kuva 12).

Unohditko Salasanasi?
Anna sähköposti
Anna salasana
Kirjaudu

## Rekisteröidy

**Voit kokeilla palvelua ilmaiseksi**




Matkapuhelinnumero  
Esim. +358401393250 tai 0401393250

Painamalla Rekisteröidy -painiketta hyväksyt [käyttöehtomme](#).

Positive Productions Oy

Kuva 12. Tekstiviestipalvelun etusivu kannettavan tietokoneen selaimessa

Tekstiviestipalvelun etusivu skaalautuu pienempään tilaan mobiililaitteella käytettäessä (kuva 13).

Anna sähköposti
Anna salasana

## Rekisteröidy

**Voit kokeilla palvelua ilmaiseksi**




Matkapuhelinnumero

Positive Productions Oy

Kuva 13. Tekstiviestipalvelun etusivu mobiililaitteen selaimessa

### 4.1.3 Kontrolleri

Kontrolleri vastaanottaa käyttäjältä tulevat pyynnöt sekä muuttaa mallia ja näkymää vastauksena niihin. Tekstiviestipalvelussa kontrollerina toimii Java-Servletit, jotka on toteutettu käyttämään REST-arkkitehtuurimallia. Tieto näkymän ja kontrollerin välillä kulkee JSON-muodossa.

JSON (JavaScript Object Notation) on JavaScriptistä riippumaton avoimen standardin mukainen tiedon esitysmuoto tiedonvälitykseen. Kuvassa 14 on esimerkki viestin lähetyksessä käytetystä JSON-esityksestä (kuva 14).[11.]

```
{
  data: {
    message: "Teille on varattu aika hammaslääkäriin huomenna maanantaina 23.3.2015
            klo 12:45. Tervetuloa! T: Jani Makkonen, Hammasklinikka",
    phoneNumber: "0408097499",
    customerName: "Tero Matilainen"
  }
}
```

Kuva 14. Esimerkki viestin lähetyksessä käytetystä JSON-esityksestä

REST (Representational State Transfer) on HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintojen toteuttamiseen. REST määrittää HTTP-metodit kutsujen toteuttamiseen näkymän ja kontrollerin välillä.

Taulukko 1. REST rajapinnan tarjoamat HTTP-metodit.

GET	Käytetään yksittäisen tiedon lukemiseen tai kaikkien listaukseen.
PUT	Käytetään tiedon päivittämiseen.
POST	Käytetään uuden luomiseen.

DELETE	Käytetään tiedon poistamiseen.
--------	--------------------------------

Tekstiviestipalvelussa on käytetty POST- ja GET-metodeja, joista POST hoitaa tiedon lisäämisen ja päivittämisen riippuen, onko tiedolla yksittäinen tunniste määritelty. GET hoitaa joko tiedon listaamisen tai hakee yksittäisen tiedon tunnisteiden avulla.

REST-arkkitehtuurin käyttäminen parantaa rajapintojen

- suorituskkyä
- skaalautuvuutta
- yksinkertaisuutta
- muunneltavuutta
- siirrettävyyttä
- luotettavuutta.

[12]

## 4.2 Integraatiot

Tekstiviestipalveluun toteutettiin integraatiot Paytrail-verkkomaksupalveluun ja Clickatellin SMS-Gatewayhin.

### 4.2.1 Paytrail

Tekstiviestipalvelusta lähetettävät viestit ovat maksullisia, joten käyttäjän täytyy pystyä ostamaan viestejä. Viestien osto hoidetaan Paytrail-verkkomaksupalvelun välityksellä. Paytrail on suomalainen verkkomaksupalvelu, joka mahdollistaa turvallisen verkkomaksamisen lukuisilla eri tavoilla.

Maksutapahtuman vaiheet ovat:

1. Asiakas valitsee tuotteet ja täyttää yhteystietonsa verkkokaupassa.
2. Asiakas ohjataan Paytrailin maksunvalintasivulle valitsemaan maksutapa.
3. Asiakas vahvistaa maksun valitsemansa maksutavan sivulla.
4. Paytrail vastaanottaa maksun asiakasvaratilille.
5. Asiakas ohjataan automaattisesti takaisin verkkokauppaan ja hänelle lähetetään maksunvahvistusviesti sähköpostilla.
6. Verkkokauppa saa tiedon maksusta reaaliajassa ja tuotteet voidaan lähettää.
7. Paytrail tilittää maksun kauppiaille tilitysviiveen kuluttua.

Paytrailin käyttäminen on maksullista. Paytrailille maksetaan kuukausimaksua ja sen lisäksi jokaista maksutapahtumaa kohden veloitetaan transaktiokulu.

Paytrail tarjoaa ohjelmistokehittäjille rajapinnan maksupalvelun käyttöönottoon, jonka avulla verkkomaksaminen saadaan integroitua osaksi tekstiviestipalvelua.

Paytrailin käyttöönotto

1. Toteutetaan tekninen integraatio Paytrail Integraatio -oppaan ohjeiden mukaan.
2. Testataan tekninen integraatio Paytrailin tarjoamilla testitunnuksilla.
3. Tehdään sopimus Paytrailin kanssa ja otetaan yrityskohtaiset tunnukset käyttöön.

Paytrail-integraatio on toteutettu lomakerajapintaa käyttäen Paytrail Integraatio -oppaan mukaan. Lomakerajapintaa käytettäessä verkkosovelluksen sivulle muodostetaan maksutiedoista form-tietoryhmä, jonka käyttäjä lähettää Paytrail-maksupalveluun. Samalla hän siirtyy maksutavan valintaan. Käyttäjän suoritettua



maksun verkkopankissa ohjataan hänet takaisin onnistuneen maksun paluuosoitteeseen. Jos maksu on epäonnistunut tai peruuntunut, käyttäjä ohjataan sitä vastaavaan paluuosoitteeseen. Paluuosoitteet kulkevat maksutietojen mukana. Taulukossa 2 on esitetty lomakerajapinnan mukana lähetettävät maksutiedot.[13.]

Taulukko 2. Lomakerajapinnan mukana lähetettävät maksutiedot.[13.]

MERCHANT_ID	Kauppiastunnus on Paytrailin asiakkaalle antama asiakasnumero. Kauppias eli Positive Productions tekee sopimuksen Paytrailin kanssa.
AMOUNT	Maksun summa.
ORDER_NUMBER	Tilausnumero on asiakkaan ostoksen yksilöivä merkkijono, joka tehdään verkkokauppaohjelmistossa.
REFERENCE_NUMBER	Oletuksena pankkiin välitettävä viitenumero luodaan automaattisesti. Niille maksutavoille, joita käytetään rajapintana, voidaan tässä kentässä välittää oma viitenumero, joka välitetään pankin palveluun automaattisesti generoidun viitenumeron sijaan.
ORDER_DESCRIPTION	Tilauksesta voidaan lähettää mitä tahansa tekstimuotoista tietoa maksujärjestelmään. Yleisimmin käytetyt tiedot ovat asiakkaan nimi ja yhteystiedot sekä tilauksen tuotetiedot. Tiedot näytetään kauppiaspaneelissa maksun lisätiedoissa.

CURRENCY	Maksun valuutta.
RETURN_ADDRESS	Osoite, johon ohjataan sen jälkeen, kun maksu on suoritettu onnistuneesti Paytrailille.
CANCEL_ADDRESS	Osoite, johon ohjataan, jos maksu on peruuntunut tai epäonnistunut.
PENDING_ADDRESS	Ei käytössä.
NOTIFY_ADDRESS	Osoite, jota kutsutaan, kun maksu merkitään onnistuneeksi.
TYPE	Maksutavan versio.
CULTURE	Lokalisatio vaikuttaa maksutapavalintasivun oletuskieleen ja summien esitystapaan.
PRESELECTED_METHOD	Maksutapa.
MODE	Maksutavan valintasivun käytötapa.

VISIBLE_METHODS	Ei käytössä.
GROUP	Ei käytössä.
AUTHCODE	Maksun varmennus koodi.

Maksutiedot esitettynä HTML-koodissa (kuva 15).

```
<form action="https://payment.paytrail.com/" method="post">
  <input name="MERCHANT_ID" type="hidden" value="13466">
  <input name="AMOUNT" type="hidden" value="99.90">
  <input name="ORDER_NUMBER" type="hidden" value="123456">
  <input name="REFERENCE_NUMBER" type="hidden" value="">
  <input name="ORDER_DESCRIPTION" type="hidden" value="Testitilaus">
  <input name="CURRENCY" type="hidden" value="EUR">
  <input name="RETURN_ADDRESS" type="hidden" value="http://www.esimerkki.fi/success">
  <input name="CANCEL_ADDRESS" type="hidden" value="http://www.esimerkki.fi/cancel">
  <input name="PENDING_ADDRESS" type="hidden" value="">
  <input name="NOTIFY_ADDRESS" type="hidden" value="http://www.esimerkki.fi/notify">
  <input name="TYPE" type="hidden" value="SI">
  <input name="CULTURE" type="hidden" value="fi_FI">
  <input name="PRESELECTED_METHOD" type="hidden" value="">
  <input name="MODE" type="hidden" value="1">
  <input name="VISIBLE_METHODS" type="hidden" value="">
  <input name="GROUP" type="hidden" value="">
  <input name="AUTHCODE" type="hidden" value="270729B19016F94BE5263CA5DE95E330">
  <input type="submit" value="Siirry maksamaan">
</form>
```

Kuva 15. Esimerkki Payrail-lomarajapinnan kautta lähetettävistä tiedoista[13]

Paytrailin toiminta tekstiviestipalvelussa aloitetaan valitsemalla maksun määrä tilin hallinnasta (kuva 16).

ViestitAsiakkaatTilin hallinta

1962 viestiä jäljellä  
Osta lisää viestejä

Valitse summa ja seuraa ohjeita

10 €

20 €


50 €

**Hinnat**  
100 viestiä - 10 €  
220 viestiä - 20 €  
600 viestiä - 50 €

Kuva 16. Maksun määrän valinta tilin hallinnassa

Seuraavaksi käyttäjä valitsee halutun maksutavan (kuva 17). Maksutavan valinnan jälkeen käyttäjä ohjataan valitun pankin verkkopalveluun vahvistamaan maksu.






ViestitAsiakkaatTilin hallinta



Suomen Verkkomaksut on nyt Paytrail.

MAKSUN TIEDOT  
Maksun saaja / toimittaja Demo Yritys (Näytä tiedot)  
Tilausnumero 6CF9C490B8AF4F34B1B87C63C4D9A7D8  
Maksun summa 50,00 €



VALITSE MAKSUTAPA

Handelsbanken

S-Pankki

Aktia

Kuva 17. Tekstiviestipalveluun upotettu Paytrail-maksutavan valinta

Maksun suoritettua käyttäjä näkee tilinhallinnasta tilaushistorian. Suoritetut maksut näkyvät tilaushistoriassa (kuva 18).

Tilaushistoria		
26.02.2015	50 €	600 viestiä
26.02.2015	20 €	220 viestiä
23.02.2015	50 €	600 viestiä
23.02.2015	10 €	100 viestiä

Kuva 18. Tilaushistorian listaus tekstiviestipalvelussa

#### 4.2.2 Clickatell SMS-Gateway

Tekstiviestipalvelun viestien lähetys tapahtuu Clickatellin SMS-Gatewayn kautta. SMS-Gateway on sovellusalausta, johon on mahdollista liittää erilaisia järjestelmiä, sovelluksia tai palvelukokonaisuuksia ohjelmistorajapinnan kautta. SMS-Gateway yhdistää eri puhelinoperaattorien viestikeskukset ja tarjoaa rajapinnan tekstiviestien lähettämiseen ja vastaanottamiseen.

Clickatell SMS-Gatewayn hinnoittelu on käyttöperusteinen. Kehittäjä ostaa viestejä haluamallaan summalla, jonka jälkeen viestit voi lähettää rajapinnan kautta.[14.]

## Clickatell käyttöönotto

1. Rekisteröidytään luomalla maksuton käyttäjätunnus Clickatellin verkkosivulla. Rekisteröinnin yhteydessä Clickatell lähettää sähköpostilla tunnukset, jonka avulla yhteys rajapintaan muodostetaan.
2. Testataan omien tunnusten ja rajapinnan toiminta. Rekisteröinnin yhteydessä käyttäjä saa 10 ilmaista krediittiä rajapinnan testausta varten. Rajapintaa voi testata verkkoselaimella.
3. Luodaan yhteys Clickatellin rajapintaan ohjelmistokoodissa ja lähetetään viesti kutsumalla rajapinnan verkko-osoitetta `http://api.clickatell.com/http/sendmsg?user=xxxxx&password=xxxxx&api_id=xxxxx&to=448311234567&text=Terve+Pekka.`

### 4.3 Tekstiviestipilvipalvelun toteuttaminen ja julkaisu

Tekstiviestipalvelu toteutetaan ja julkaistaan pilvipalveluna Googlen ympäristöön.

### 4.3.1 Projektin luonti Google App Engineen

Pilvipalvelun toteuttaminen aloitetaan luomalla projekti Googlen App Engine -palveluun. Projektin luominen tapahtuu Google App Engine -verkkosivulla.

Googlen alustan käyttöönotto vaatii, että kehittäjällä on käytössä Google-tili. Google-tilin luonti tapahtuu maksutta Google Accounts -sivustolla. Kun Google-tili on olemassa, projektin luomisessa tulee antaa yksilöllinen tunniste ja otsikko palvelulle (kuva 19). Tunniste toimii palvelun osoitteena, ja sitä käyttämällä projekti julkaistaan Google App Engineen.

Google app engine vepsalainensamu@gmail.com | [My Account](#) | [Help](#) | [Sign out](#)

## Create an Application

You have 21 applications remaining.

**Application Identifier:**  
 .appspot.com  **Yes, "smspalvelu" is available!**

All Google account names and certain offensive or trademarked names may not be used as Application Identifiers. You can map this application to your own domain later. [Learn more](#)

**Application Title:**

Displayed when users access your application.

**Authentication Options (Advanced):** [Learn more](#)

Google App Engine provides an API for authenticating your users, including Google Accounts, Google Apps, and OpenID. If you choose to use this feature for some parts of your site, you'll need to specify now what type of users can sign in to your application.

☒ **Open to all Google Accounts users (default)**  
If your application uses authentication, anyone with a valid Google Account may sign in.

☐ **Restricted to the following Google Apps domain:**  
  
e.g. foo.com  
If your application uses authentication, only members of this Google Apps domain may sign in. If your organization uses Google Apps, use this option to create an application (e.g. an HR tracking tool) that is only accessible to accounts on your Google Apps domain. This option cannot be changed once it has been set.

☐ **(Experimental) Open to all users with an OpenID Provider**  
If your application uses authentication, anyone who has an account with an OpenID Provider may sign in.

© 2015 Google | [Terms of Service](#) | [Privacy Policy](#) | [Blog](#) | [Discussion Forums](#) | [Project](#) | [Docs](#)

Kuva 19. Projektin luonti Google App Engineen

Käyttäjän syötettyä tunnisteen, otsikon ja painamalla Create Application -painiketta on pilvialusta valmis käytettäväksi. Yhteenveto-sivu varmistaa projektin luonnin onnistuneen (kuva 20).

## Application Registered Successfully

The application will use **smspalvelu** as an identifier. This identifier belongs in your application's configuration as well. Note that this identifier cannot be changed. [Learn more](#)

The application uses the **High Replication** storage scheme. [Learn more](#)

If you use Google authentication for your application, **Tekstiviestipalvelu** will be displayed on Sign In pages when users access your application.

Choose an option below:

- View the [dashboard](#) for Tekstiviestipalvelu.
- Use [appcfg](#) to upload and deploy your application code.
- Add [administrators](#) to collaborate on this application.

Kuva 20. Yhteenveto onnistuneesta Google App Engine -projektin luonnista

### 4.3.2 Google App Engine SDK:n asennus

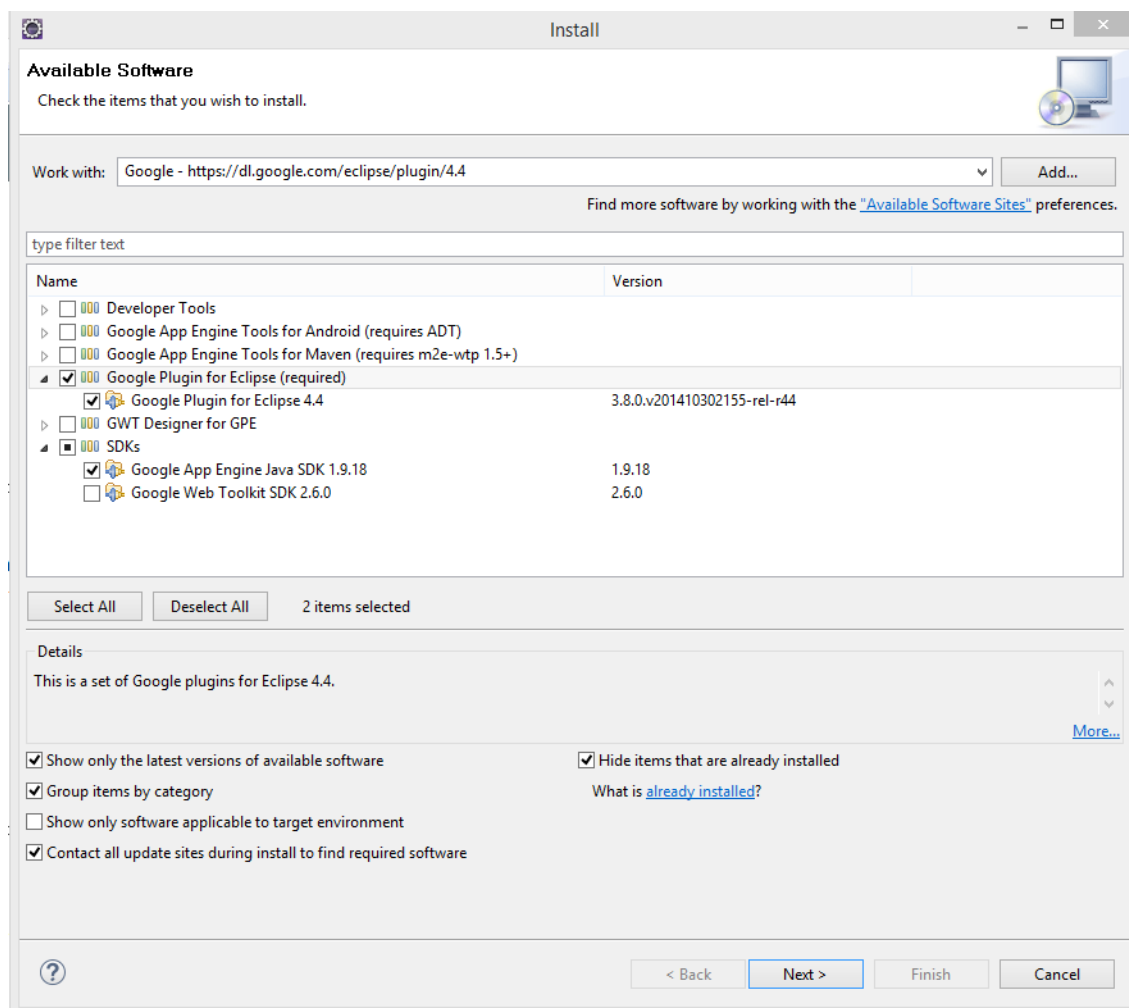
Google App Engine -sovelluksen kehittämiseen kehittäjä tarvitsee Eclipsen ja Google App Engine SDK:n. Eclipse on Java-sovelluskehittäjille suunnattu kehitysympäristö. Eclipsen asennuspaketti ladataan Eclipsen verkkosivuilta ja puretaan haluttuun kansioon.

SDK:n asentaminen tapahtuu Eclipsen Help -> Install New Software -valikon kautta. Google tarjoaa Eclipseen uusimmille versiolle oman pluginin ja SDK:n verkkosivuillaan (kuva 21).[15.] Google Plugin avulla asennetaan Eclipseen tarvittavat työkalut oman pilvipalvelusovelluksen kehittämiseen ja siirtämiseen pilveen. Google Plugin asennetaan Eclipsen Help -> Install New Software... -valikon kautta.

Eclipse version	Installation instructions	Direct plugin link
Eclipse 4.4 (Luna)	<a href="#">Plugin for Eclipse 4.4 (Luna)</a>	<a href="https://dl.google.com/eclipse/plugin/4.4">https://dl.google.com/eclipse/plugin/4.4</a>
Eclipse 4.3 (Kepler)	<a href="#">Plugin for Eclipse 4.3 (Kepler)</a>	<a href="https://dl.google.com/eclipse/plugin/4.3">https://dl.google.com/eclipse/plugin/4.3</a>
Eclipse 3.8/4.2 (Juno)	<a href="#">Plugin for Eclipse 3.8/4.2 (Juno)</a>	<a href="https://dl.google.com/eclipse/plugin/4.2">https://dl.google.com/eclipse/plugin/4.2</a>

Kuva 21. Google Eclipse -pluginin asennuslinkki Google App Engine -sivustolla

Asennus aloitetaan valitsemalla käytettävää versiota vastaava osoite ja syötetään se Eclipsen Work with: -kenttään (kuva 22). Eclipse lataa osoitteessa olevat kehitystyökalut, joista valitaan Google Plugin for Eclipse ja Google App Engine Java SDK.



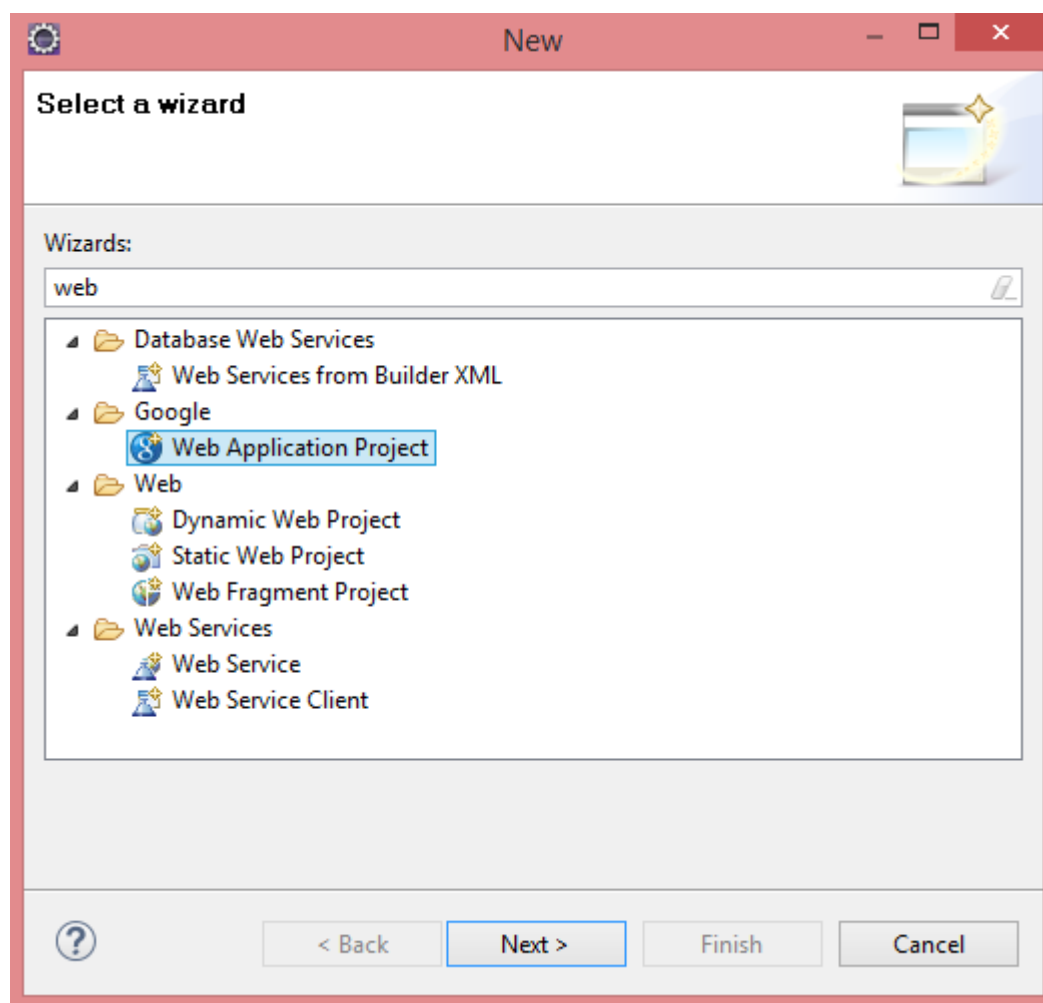
Kuva 22. Google Plugin ja Google App Engine SDK:n asennus

SDK:n asennuksen jälkeen ympäristö on valmis sovelluksen kehittämisen aloittamiseen.

#### 4.3.3 Projektin luonti

Kun Google App Enginen SDK ja Google Plugin on asennettu, voidaan luoda uusi App Engine -projekti. Uusi projekti luodaan Google Web Application Project Eclipsen wizard-toiminnon avulla (kuva 23).





Kuva 23. Projektin luonti Eclipse wizard -toiminnon avulla

Projektin konfiguroinnissa syötetään projektin nimi ja paketti. Lisäksi otetaan käyttöön Google App Engine SDK (kuva 24).

**Create a Web Application Project**  
Create a Web Application project in the workspace or in an external location

Project name:  
posp-sms

Package: (e.g. com.example.myproject)  
fi.posp.sms

Location  
☒ Create new project in workspace  
☐ Create new project in:  
 Directory: C:\Users\Samu\Oppari\workspace\posp-sms Browse...

Google SDKs  
☐ Use Google Web Toolkit  
     ☒ Use default SDK (none) [Configure SDKs...](#)  
     ☐ Use specific SDK:   
☒ Use Google App Engine  
     ☒ Use default SDK (App Engine - 1.9.18) [Configure SDKs...](#)  
     ☐ Use specific SDK: App Engine - 1.9.18  
 The project will use App Engine's [High Replication Datastore \(HRD\)](#) by default.

Identifiers for Google App Engine  
☐ Leave App Id field blank  
☒ Use App Id posp-sms Browse...

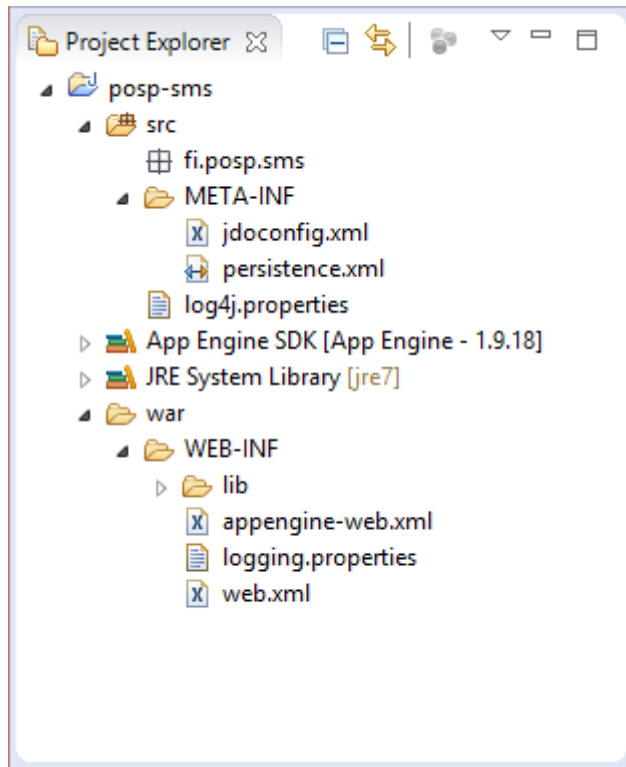
Your app will be deployed at:  
 - http://yourappid.appspot.com for regular applications  
 - http://yourappid.yourdomain.com for domain applications

Sample Code  
☒ Generate project sample code

? < Back Next > Finish Cancel

Kuva 24. Uuden Web Application Projektin luonti

Kun projektin konfigurointi on valmis, Eclipse luo pohjan projektille (kuva 25).



Kuva 25. Uuden projektin rakenne Eclipsessä

#### 4.3.4 Viestin lähetys

Projekti luonnin jälkeen voidaan aloittaa kehitystyö. Tässä luvussa esitellään, miten viestin lähettäminen on toteutettu pilvipalveluun.

Käyttöliittymässä käyttäjä täyttää viestin tiedot ja painaa Lähetä-painiketta (kuva 26).

**Lähetä viesti**

Hei Matti, uusi autonne on valmiina noudettavaksi  
Olarin liikkeestämme. Ystävällisin terveisin Delta Auto.

54 merkkiä jäljellä

Matti Pitkänen

+358123456789

Esim. +358401393250 tai 0401393250

Lähetä

Tallenna luonnokseksi

Kuva 26. Viestin lähetys

Lähetä-painike kutsuu SendMessageView-tiedoston JavaScript-metodia send, joka kerää datan formista ja lähettää datan post-kutsussa palvelimelle osoitteeseen /json/messages/add (kuva 27).

```
this.send = function() {
    var data = $("#sendMessageForm").serializeObject();
    data.status = "SENT";
    $.post("/json/messages/add", {data: JSON.stringify(data)}, function(result) {
        MESSAGES_VIEW.init();
    }, "json");
}
```

Kuva 27. send-metodin toteutus

Palvelimen saadessa kutsun se tunnistaa web.xml -tiedoston määrittymisen avulla, mihin luokkaan pyyntö ohjataan. Kuvasta 28 huomataan, että kaikki /json/messages/-alkuiset osoitteet ohjataan JSONMessagesServlet-kontrolleriin, joka sijaitsee fi.posp.sms.servlet.json-paketissa.

```

<servlet>
  <servlet-name>JSONMessagesServlet</servlet-name>
  <servlet-class>fi.posp.sms.servlet.json.JSONMessagesServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>JSONMessagesServlet</servlet-name>
  <url-pattern>/json/messages/*</url-pattern>
</servlet-mapping>

```

Kuva 28. JSONMessagesServlettiin ohjaus web.xml-tiedostossa

Seuraavaksi esitellään Message Domain -luokka (kuva 29). Message-oliota käytetään palvelun viestin tietojen kuljettamiseen ja tietokantaan tallentamiseen.

```

@PersistenceCapable(identityType = IdentityType.APPLICATION)
public class Message {

    @PrimaryKey
    @Persistent(valueStrategy = IdGeneratorStrategy.IDENTITY)
    private Key key;

    @Persistent
    private Date created;

    @Persistent
    private Date updated;

    @Persistent
    private Text message;

    @Persistent
    private String status;

    @Persistent
    private Date sentDate;

    @Persistent
    private String toPhoneNumber;

    @Persistent
    private String toFirstName;

    @Persistent
    private String toLastName;

    @Persistent
    private Boolean archived;

}

```

Kuva 29. Message domain -luokka

Pyynnön saapuessa kontrolleriin kerätään tiedot talteen ja muutetaan se JsonObject-olioksi. Seuraavaksi parsitaan JsonObjectista tiedot muuttujiin, jonka jälkeen luodaan

Message-olio ja asetetaan parsitut tiedot Message-olioon. Kun viestin tiedot on kerätty, on viesti valmis lähetettäväksi. Seuraavaksi viesti lähetetään SMS-Gatewayn kautta ja tallennetaan tietokantaan. Lopuksi palautetaan tieto onnistuneesta kutsusta näkymälle (kuva 30).

```
// Muunnetaan Näkymältä tullut pyyntö JsonObject olioksi
JsonObject data = JSONUtils.parseDataJsonObject(req);

// Parsitaan datasta viesti ja puhelinnumero
String messageString = JSONUtils.getString(data, "message");
String phoneNumber = JSONUtils.getString(data,
"phoneNumber");

// Luodaan uusi viesti olio
Message message = new Message();
message.setMessage(messageString);
message.setToPhoneNumber(phoneNumber);

// Lähetetään viesti SMS-Gatewayn kautta
sendSMS(messageString, phoneNumber);

// Tallennetaan lähetetty viesti tietokantaan
getMessageDAO().saveMessage(message);

// Palautetaan tieto lähetyksen onnitustumisesta Näkymälle
resp.getWriter().write(json);
```

Kuva 30. Kontrollerin toiminta viestin lähetyksessä

Kuvassa 31 on esitetty viestin lähetyks SMS-Gateway-rajapintaan. Koodissa lisätään puhelinnumero ja viesti URL-osoitteen parametreiksi ja lähetetään HTTP-kutsu SMS-Gateway -rajapinnan tarjoajalle.

```
// Lisätään puhelinnumero ja viesti parametreiksi
StringBuilder params = new StringBuilder();
params.append("to=").append(toPhoneNumber);
params.append("&text=").append(message);

// Luodaan yhteys SMS Gatewayhin ja lähetään viesti
URL url = new URL(SMSGATEWAY_API_URL);
URLConnection connection = (URLConnection) url.openConnection();
connection.setDoOutput(true);
connection.setRequestMethod("POST");
OutputStream outputStream = connection.getOutputStream();
OutputStreamWriter writer = new OutputStreamWriter(outputStream);
writer.write(params.toString());
```

Kuva 31. SMS-Gatewayn käyttö

Metodissa `saveMessage`, kontrolleri kutsuu viestin tallennusta. Samalla asetetaan viestille luonti- ja muokkausajankohdat. (Kuva 32).

```
@Override
public Message saveMessage(Message message) {
    log.info("save message");
    try{
        Date now = DateUtils.now();
        message.setCreated(now);
        message.setUpdated(now);

        pm.makePersistent(message);
    }finally{
        pm.close();
    }
    return message;
}
```

Kuva 32. `saveMessage`-metodin toteutus

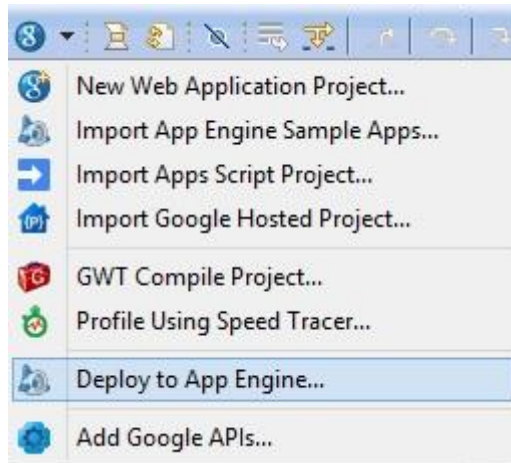
#### 4.3.5 Julkaisu Google App Engineen

Ennen julkaisua varmistetaan asetukset `appengine-web.xml`-tiedostosta.(kuva 33). Tiedostoon syötetään palvelun versionumero ja yksilöllinen tunniste, joka asennettiin vaiheessa 4.5.1.

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>smspalvelu</application>
  <version>1</version>
</appengine-web-app>
```

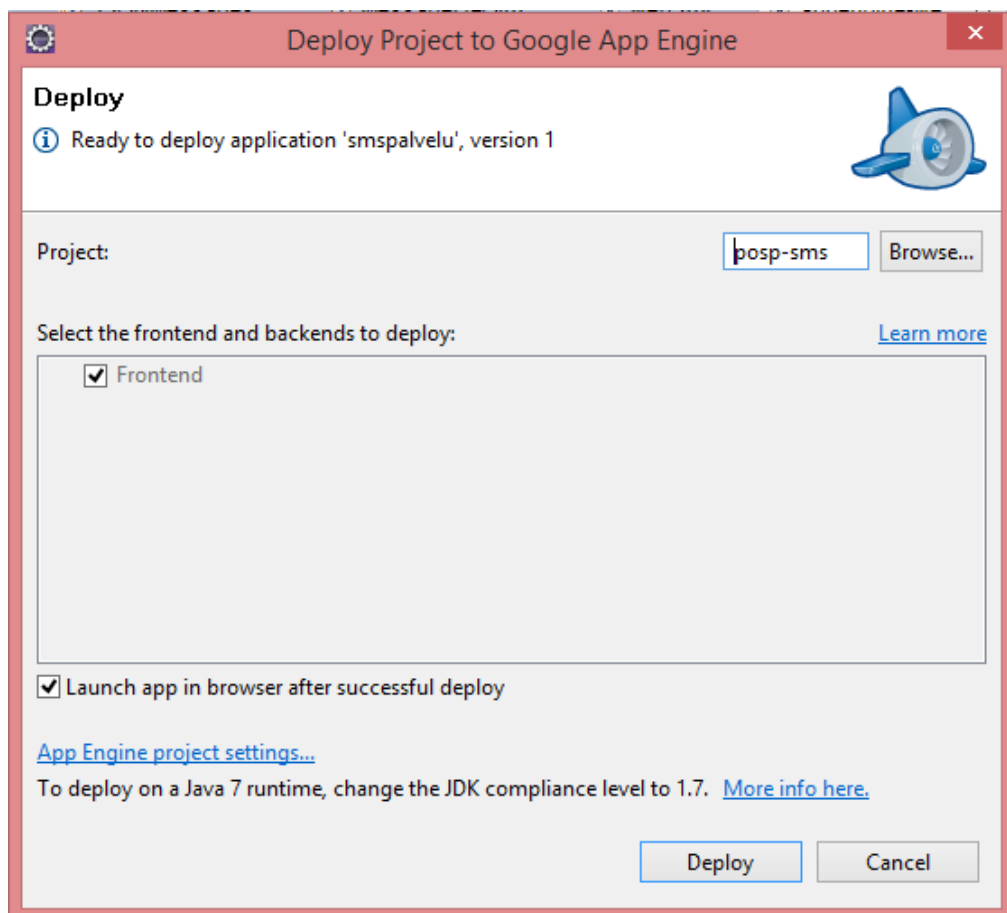
Kuva 33. `appengine-web.xml`

Palvelu julkaistaan Google Eclipse pluginin `Deploy` -toiminnon avulla (kuva 34).



Kuva 34. Deploy to App Engine -valinta Google pluginissa

Lopullinen julkaisun hyväksyntä Google Cloud -ympäristöön tapahtuu varmistusnäkyssä (kuva 35).



Kuva 35. Julkaisemisen varmistus



Julkaisu on valmis ja palvelua voi käyttää Google App Engine -pilvialustalla.

## 5 Yhteenveto

Opinnäytetyössä käytiin läpi pilvipalveluna toteutetun tekstiviestipalvelun vaatimukset, käyttötarkoitus ja ominaisuudet. Työssä esiteltiin toteutukseen käytetty Google App Engine -pilvialusta, alustan käytön aloittamiseen vaadittavat toimenpiteet sekä pilvipalvelun toteutus ja julkaisu Googlen Cloud -ympäristöön esimerkkien avulla. Lisäksi työssä pohdittiin verkkopalvelun kehityksen eroja pilvialustan ja perinteisen virtuaalipalvelinympäristön käytön välillä.

Esitettyjä tekniikoita käyttämällä onnistuttiin toteuttamaan tilaajana toimivalle Positive Productionsille pilvipalvelu tekstiviestien lähettämiseen. Käyttöperusteisella hinnoittelulla toteutetun tekstiviestipalvelun on tarkoitus pienentää asiakasyritysten kynnystä ottaa palvelu käyttöön omassa liiketoiminnassaan. Palvelun käyttö, jossa ei ole jatkuvaa kuluja, sopii hyvin myös pienille yrityksille, joilla kassavirta ei ole jatkuvaa.

Tekstiviestipalvelun toteutettiin Google App Engine -pilvialustalle MVC-arkkitehtuurin mukaan. Palveluun integroitiin Clickatellin SMS-Gateway ja Paytrail-verkkomaksut. Sovelluskehittäjille kirjoitettu opinnäytetyö antaa, esimerkkien avulla, kuvan verkkopalveluiden kehittämisestä ja julkaisusta Googlen ympäristöön.

Google App Engine ja muut -pilvialustat yleistyvät kovaa vauhtia. Kattavien ominaisuuksien, selkeiden käyttöohjeiden, helpon käytön ja joustavan hinnoittelun ansiosta ohjelmistokehittäjien on helppo ryhtyä kehittämään verkkopalveluita pilvialustalle.

Pilvialustojen skaalautuvuus, turvallisuus, käytön helppous sekä sisäänrakennetut rajapinnat ja työkalut nopeuttavat verkkopalveluiden kehitystä ja ylläpitoa. Joustava hinnoittelu, jossa pilvialustan käyttö maksaa vasta kuormituksen kasvaessa, mahdollistaa palvelun julkaisemisen verkkoon edullisesti ilman palvelimen hankintakuluja. Käytännössä hosting on maksutonta, jos käyttömäärät ovat pieniä. Hinnoittelumalli on hyvä etenkin tuotekehityksessä, jossa kehitettävä ratkaisu alkaa tuottaa rahaa vasta valmistuttuaan.

Verkkosovellusten kehityksessä pilvialustaa käytettäessä säästytään useilta toimenpiteiltä perinteiseen virtuaalipalvelimelle tehtävään sovelluskehitysmalliin verrattuna. Lisäksi aikaa ja rahaa säästyy.

## Lähteet

- 1 Google Cloud. Google. <<https://cloud.google.com>>. Luettu 13.04.2015.
- 2 Google App Engine. Google. <<https://cloud.google.com/appengine/docs>>. Luettu 13.04.2015.
- 3 Google Cloud Datastore. Google. <<https://cloud.google.com/datastore>>. Luettu 13.04.2015.
- 4 NoSQL. Verkkodokumentti. <<http://en.wikipedia.org/wiki/NoSQL>>. Luettu 13.04.2015.
- 5 ACID-transaktiot. Verkkodokumentti. <<http://en.wikipedia.org/wiki/ACID>>. Luettu 13.04.2015.
- 6 Google App Engine. Google. <<https://cloud.google.com/appengine>>. Luettu 13.04.2015.
- 7 MVC -arkkitehtuuri. Verkkodokumentti. <<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>>. Luettu 13.04.2015.
- 8 JDO. Google. <<https://cloud.google.com/appengine/docs/java/datastore/jdo/overview-dn2>>. Luettu 13.04.2015.
- 9 DAO. Verkkodokumentti. <[http://en.wikipedia.org/wiki/Data\\_access\\_object](http://en.wikipedia.org/wiki/Data_access_object)>. Luettu 13.04.2015.
- 10 Bootstrap. <<http://getbootstrap.com>>. Luettu 13.04.2015.
- 11 JSON. Verkkodokumentti. <<http://en.wikipedia.org/wiki/JSON>>. Luettu 13.04.2015.
- 12 REST. Verkkodokumentti. <[http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)>. Luettu 13.04.2015.
- 13 Paytrail. Verkkodokumentti. <<http://docs.paytrail.com/fi/index-all.html>>. Luettu 13.04.2015.
- 14 Clickatell SMS-Gateway. <<https://www.clickatell.com/>>. Luettu 13.04.2015.

- 15 Google Eclipse plugin.  
<<https://cloud.google.com/appengine/docs/java/tools/eclipse>>. Luettu  
13.04.2015.