Toni Laitinen

Muistelusovelluksen jatkokehitys

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

28.8.2017

Metropolia

Tekijä(t) Otsikko Sivumäärä Aika	Toni Laitinen Muistelusovelluksen jatkokehitys 25 sivua 28.8.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikan koulutusohjelma
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori, Ilpo kuivanen Lehtori, Jussi Alhorinne Toimitusjohtaja, Ilkka Tiainen

Insinöörityön tarkoituksena oli jatkaa verkkosovelluksen kehittämistä. Sovellus on ikäihmisille suunnattu muistelupalvelu, johon voi tallentaa tarinoita. Pääasiallisena tehtävänä oli luoda uusi käyttöliittymä sekä verkkokauppa sovellukselle.

Aluksi insinöörityössä esitellään sovelluksen toimintaa. Käydään läpi työympäristön asennusta ja palvelimen pystyttämistä, jonka jälkeen tutustutaan jatkokehityksen aikana tehtyihin muutoksiin.

Jatkokehityksessä tehty uusi käyttöliittymä oli muutamien testikäyttäjien mielestä parempi kuin vanha. Ohjelman kehitys tulee jatkumaan tämän dokumentoinnin jälkeen.

Avainsanat	Java, Javascript, HTML, Red5, Spring, MySQL

Author(s) Title Number of Pages Date	Toni Laitinen Further Development of Recollection Application 25 pages 28 August 2017
Degree	Bachelor of Engineering
Degree Programme	Information technology
Specialisation option	Software engineering
Instructor(s)	Ilpo Kuivanen, Senior Lecturer Jussi Alhorinne, Senior Lecturer

Purpose of this thesis is to continue development of internet application. Application is targeted towards elderly people, where they can record stories. Main task was to create new user interface and online store.

In the beginning of thesis will be introduction of application. After that we go through installation of working environment and setting up server. At last we look at new changes made to application during further development.

According to some test users, new user interface made during development was better than old one. Development of this application will continue after this thesis.

Keywords	Java, Javascript, HTML, Red5, Spring, MySQL

Sisällys

Lyhenteet

1 Johdanto 1

2 Sovellus 1

3 Työympäristön asennus 4

3.1 JDK:n asennus 4
3.2 Projekti Eclipseen 5
3.3 Koontiversion luonti 6

4 Testipalvelimen pystyttäminen 8

4.1 Etäkoneen pystyttäminen 8
4.2 Red5:n asennus 8
4.3 MySQL:n asennus 8
4.4 IIS:n asennus 9
4.5 Palvelimen käynnistäminen 10

5 Jatkokehitys 10

5.1 Käyttöliittymä 10
5.2 Koodi muutokset 12
5.2.1 Clouds.js 12
5.2.2 Genericpage.tag 18
5.2.3 Home.jsp 18
5.2.4 Muut muutokset 19
5.3 Verkkokauppa 21
5.4 Tietokantojen yhdistäminen 23

6 Yhteenveto 25

Lähteet 26

Lyhenteet

JDK	Java Development Kit. Javan ohjelmistokehityspaketti.
Red5	Red5 on mediapalvelin, joka on tarkoitettu videon suoratoistoa varten.
Pilvipalvelu	Palvelu joka tarjotaan etäkoneelta.
SaaS	Software as a Service. Pilvilaskennanmalli jossa ohjelmisto tarjotaan palveluna
Instanssi	Pilvipalvelussa instanssi tarkoittaa yksittäistä virtuaalista palvelinta pilviverkossa.
IIS	Internet Information Services. Web-palvelinohjelmisto, jota käytetään Windows-pohjaisissa palvelimissa.
DOM	Document Object Model. Järjestelmäriippumaton ohjelmointirajapinta, jolla voidaan kuvata eri dokumenttien rakenne puuna, jossa jokainen solmu esitetään oliona.
JSP	Java Server Pages. Javaan perustuva teknologia, jonka avulla luodaan dynaamisia web-sivuja käyttäen mm. HTML:ää ja XML:ää.
CSS	Cascading Style Sheets. Kieli joka kuvaa tyylin jolla HTML-elementit esitetään
Paytrail	Verkkomaksamiseen käytetty palvelu
Bootstrap	Verkkosivujen suunnitteluun tarkoitettu kehys, jonka on suunnattu pääasiallisesti mobiililaitteiden skaalautuvaan näkymään.
MD5	Kryptografiassa (salakirjoitustekniikka) käytetty algoritmi, jolla syötteestä tehdään tiiviste.
MySQL	Relaatiotietokantaohjelmisto

1 Johdanto

Työ sisältää kuvauksen ohjelman toiminnasta, ohjeet työympäristön asentamiseen sekä testipalvelimen pystyttämiseen. Lopuksi käydään läpi jatkokehityksen aikana tehtyjä muutoksia.

Sovellus on ikäihmisille suunnattu muistojen tallennusverkkopalvelu. Palvelua voidaan käyttää tietokoneella sekä tabletilla. Se on SaaS-pohjainen sovellus, jossa on käytetty Spring MVC-sovelluskehystä sekä Red5-mediapalvelinta.

Alun perin ohjelma on teetetty intialaisella ohjelmistoalihankkijalla. Ohjelmistotyötä ovat aiemmin jatkaneet useat eri ohjelmoijat. Minut palkattiin tekemään uusi käyttöliittymä sekä verkkokauppa sovellukseen. Sovelluksen nimeä ei mainita työssä, koska se on pyydetty pitämään salassa.

Uusi käyttöliittymä tehtiin yhteistyössä graafisen suunnittelijan kanssa. Saadakseni käyttöliittymästä halutun näköisen jouduin muuttamaan logiikkaa, jolla tarinat piirrettiin näkymään. Verkkokaupan tekeminen jäi kesken, mutta siitä on tehty suunnitelma. Verkkomaksupalveluna tullaan käyttämään Paytrail:ia. Työssä käydään läpi, miten Paytrail-maksutapahtuma toteutetaan.

2 Sovellus

Palvelu on verkkopalvelu muistojen tallentamista varten. Tarinoita voidaan tallentaa teksti-, kuva-, ääni- sekä videomuodossa. Video- ja äänitarinat voidaan nauhoittaa suoraan palveluun käyttäen web-kameraa ja mikrofonilla, tai vaihtoehtoisesti ladata tiedostot tietokoneelta. Käyttäjä voi jakaa tarinansa julkisesti kaikkien nähtäväksi. Halutessaan käyttäjä voi kuitenkin pitää tarinansa yksityisenä tai erikseen luodun yhteisön kesken.

Tarinoiden selaamiseen on olemassa kaksi erilaista näkymää: Aikajananäkymä, jossa tarinoita selataan aikajanaa pitkin kolmiulotteisessa näkymässä, sekä kartta-näkymä jossa tarinat on sijoitettu kartalle.

Sovellus on suunnattu pääasiallisesti ikäihmisille, joten ohjelman toiminta on pyritty pitämään mahdollisimman selkeänä ja fonttikoko suurena.

Taulukko 1. Käytetyt kolmannen osapuolen tekniikat

Teknologia	Kuvaus
Java	Ohjelmointikieli. Sovelluksessa käytetty pääasiallinen ohjelmointikieli.
Java EE	Java Platform, Enterprice Edition. Alusta java-kehitystä varten, joka tarjoaa verkkoon liittyviä työkaluja.
HTML5	Hypertext Markup Language. Käytetään internetsivujen luontiin.
CSS	Cascading Style Sheet. HTML-sivujen tyylin muotoilukieli.
Javascript	Ohjelmointikieli jolla voidaan lisätä dynaamista toiminnallisuutta web-sivuihin
jQuery	Javascript-kirjasto, jota käytetään mm. DOM-elementtien käsittelyssä.

three.js	Javascript-kirjasto, jota käytetään 3D-animaation esittämisessä. Sovelluksen tarinaelementit piirrettiin käyttäen three.js-kirjastoa.
Bootstrap	Sovelluskehys jota käytetään responsiivisien sivujen luonnissa. Monissa sovelluksen painikkeissa käytettiin tätä tekniikkaa, jotta ne saataisiin uudelleen järjesteltyä erikokoisissa näkymissä.
Spring	Spring on sovelluskehys, jonka avulla voidaan yhdistää eri systeemejä toisiinsa.
Hibernate	Hibernate on java-pohjainen sovelluskehys, jota käytetään olioiden kartoittamiseen
JSP	Java Server Pages on teknologia, jolla web-sivu yhdistetään tietokantaan.
JSTL	JSTL on JSP:n avainsana kirjasto, joka sisältää mm. XML ja SQL tageja.
Red5	Red5 on median suoratoisto palvelin. Tätä palvelinta käytettiin sovelluksen käynnistämiseen.

Maven	Maven on työkalu, jolla projektin rakentaminen automatisoidaan.
HDFVR	Verkossa sivulla toimiva sovellus, jolla voidaan nauhoittaa video web-kameraa käyttäen suoraan palvelimelle.

Sovelluksen rakenne on kuvattu tarkemmin edeltävissä dokumenteissa. [1.] [2.]

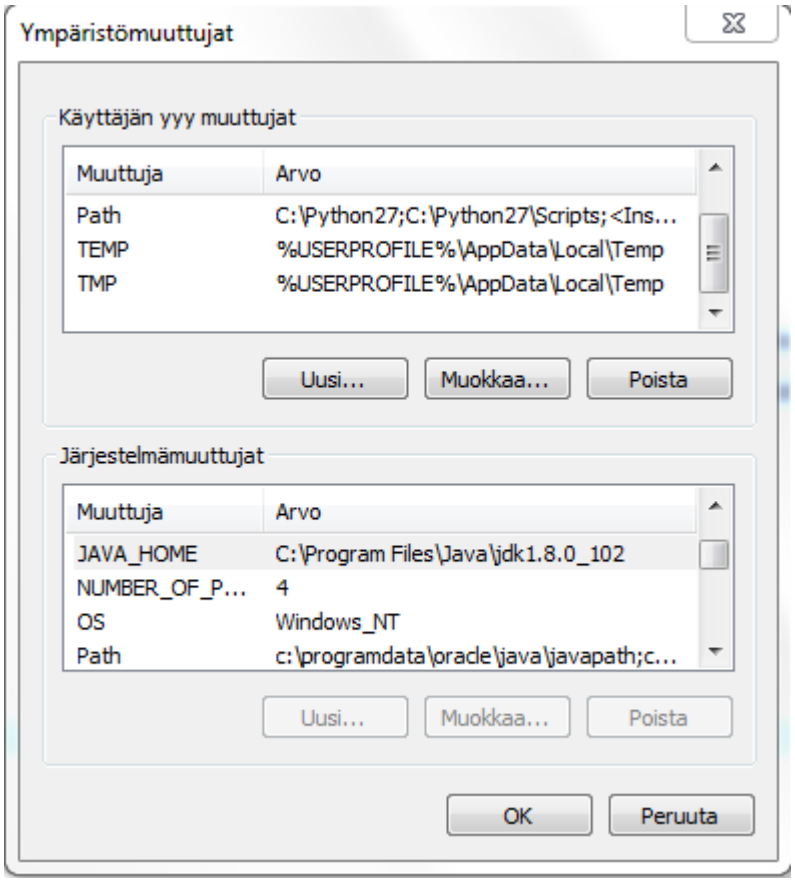
3 Työympäristön asennus

3.1 JDK:n asennus

Työympäristön asennus aloitettiin lataamalla JDK (Java Development Kit). Sen jälkeen ympäristömuuttujiin lisättiin JAVA_HOME ja määriteltiin sen arvoksi hakemisto, johon JDK asennettiin.

1. Hiiren oikealla Tietokoneesta > Ominaisuudet > Järjestelmän lisäasetukset > Ympäristömuuttujat.

2. Lisätään järjestelmämuuttujiin uusi muuttuja nimeltä JAVA_HOME ja määritetään sen arvoksi polku hakemistoon, johon JDK on asennettu.



Kuva 1 Ympäristömuuttujat

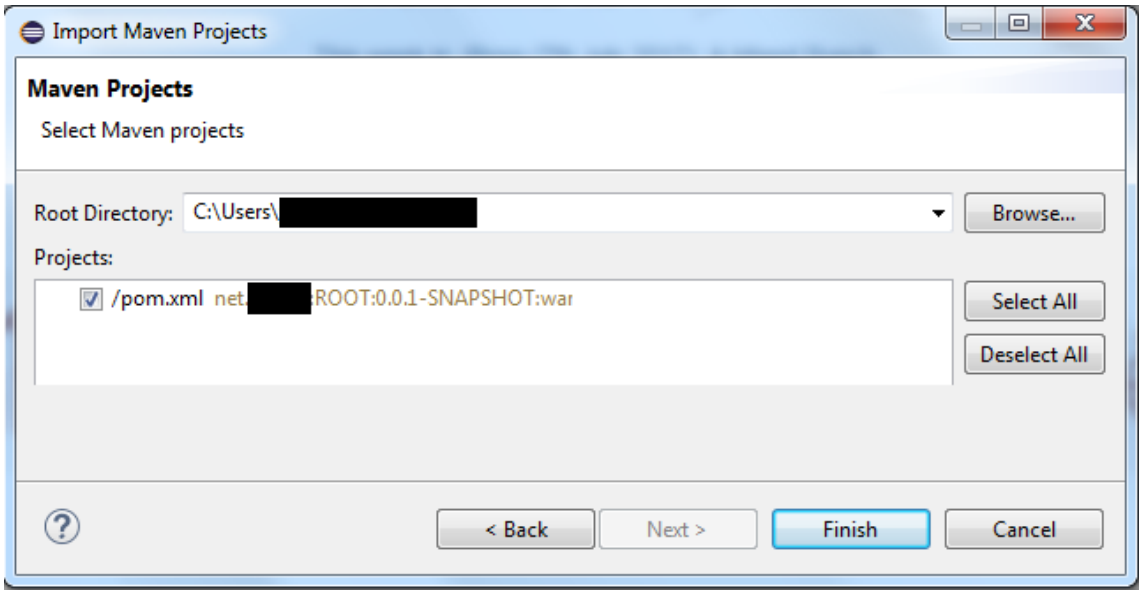
3. Sovelluksen ohjelmointi tapahtui käyttäen Eclipse-IDE työympäristöä. Ladataan Eclipse ja käynnistetään se.

4. Asetetaan java-kääntäjäksi juuri ladattu JDK. Window>Preferences>Java>Installed JREs> JDK

3.2 Projekti Eclipseen

1. Projekti ladattiin gitlabista ”git clone” -komennon avulla. Sen jälkeen projekti tuotiin Eclipseen.

2. File > Import…> Existing Maven Projects



Kuva 2 Projektin tuonti Eclipseen

3. Juurihakemistoksi valittiin gitlabista ladattu projekti, jonka jälkeen pom.xml-niminen tiedosto ruksattiin ja lopuksi painettiin ”Finish”-painiketta.

4. Projektia varten tarvittiin myös muutamia kirjastoja, joita varten projektin juureen lisättiin ”lib”-kansio.

5. Project > Properties > Java Build Path. Jossa sitten lisättiin tarvittavat kirjastot ja poistettiin virheelliset kirjastot.

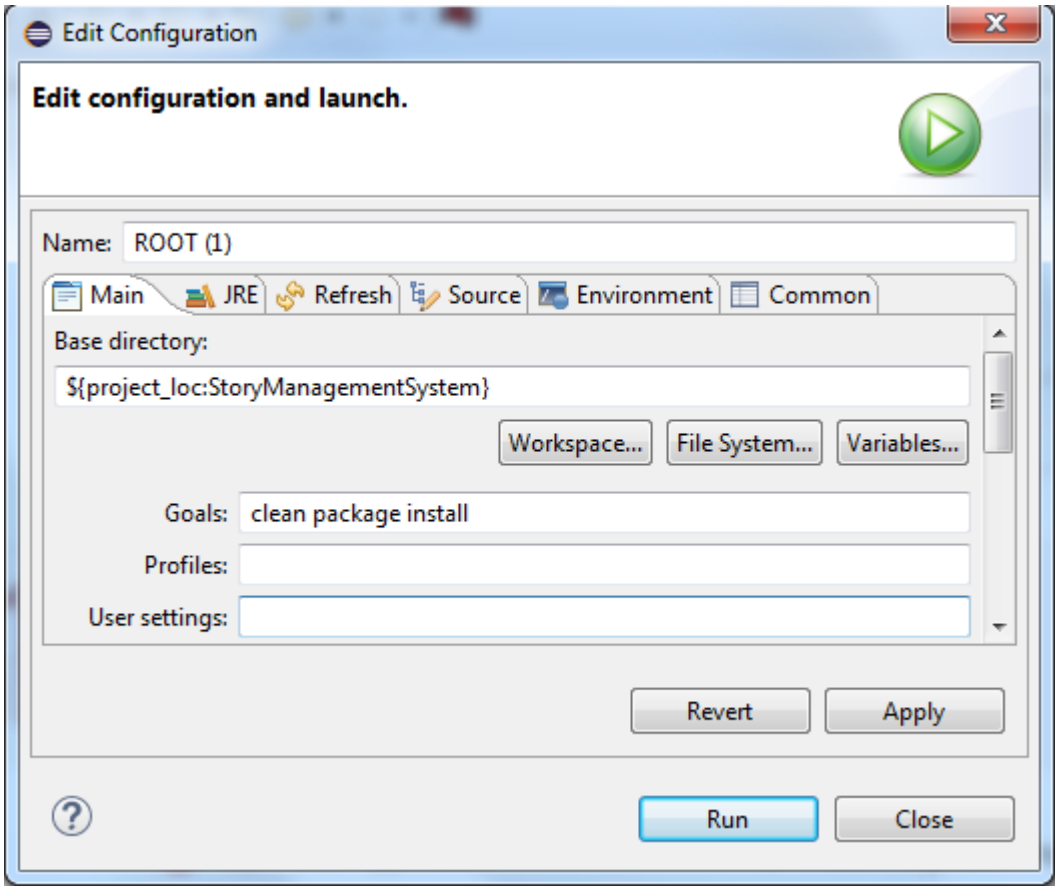
6. Mikäli projektissa esiintyi ”Maven depencies”-virheitä, niin nämä saatiin poistettua päivittämällä.

7. Hiiren oikealla projektista > Maven > Update project…> Ok

3.3 Koontiversion luonti

Ohjelmistosta luotiin koontiversio, jotta projekti saatiin toimimaan Red5-palvelimella.

1. Hiiren oikealla projektista > Run As > 6 Maven Build



Kuva 3 Koontiversion luonti Eclipsessä

2. Tavoitteeksi asetettiin ”clean package install” ja painettiin ”Run”.

3. Projektin target kansioon luotiin WAR-tiedosto sekä projektin ROOT-kansio, joka sitten vietiin Red5-palvelimelle.

4. Red5-serverin webapps-kansiosta poistettiin valmiina oleva ROOT-kansio ja korvattiin se aikaisemmin luodulla ROOT-kansiolla.

5. Palvelin käynnistettiin red5-juurihakemistossa sijaitsevasta red5.bat-tiedostosta. Sovelluksen näkyy osoitteessa http://localhost:5080, mikäli portti 5080 on avattu palomuurin asetuksista.

4 Testipalvelimen pystyttäminen

Ohjelmiston kehittämistä ja testaamista varten luotiin uusi testipalvelin. Uudella testipalvelimella on tarkoitus testata sovelluksen uuden version testaamista, ennen kuin se viedään julkiselle palvelimelle. Edeltävällä testipalvelimella oleva versio on jatkuvan kehityksen alaisena, joten se ei ollut optimaalinen testaamiseen.

4.1 Etäkoneen pystyttäminen

Aluksi Amazonin pilvipalveluun luotiin uusi instanssi. Instanssin luonnin yhteydessä ladattiin tiedosto, joka sisälsi etäkoneen ip-osoitteen ja salasanan. Tarvittavat portit lisättiin saapuviin sääntöihin (inbound rules), jotta etäkoneelle päästään kirjautumaan omalta koneelta etätyöpöytäyhteyden avulla.

4.2 Red5:n asennus

Red5-mediapalvelimesta ladattiin versio 1.07, sillä se muissa versioissa näytti olevan ongelmia sovelluksen kanssa. Red5:n webapps-kansioon lisättiin web-kameravideon nauhoitussovellus HDFVR. Olemassa oleva ROOT-kansio korvattiin Eclipsessä luodulla ROOT-kansiolla.

4.3 MySQL:n asennus

Tietokantaa varten ladattiin MySQL-Installer, josta saatiin helposti asennettua MySQL-toimintaa varten tarvittavat komponentit. Asennettaviin komponentteihin kuuluivat MySQL Server, MySQL Connector ja MySQL Workbench.

Asennuksen yhteydessä pyydettiin määrittelemään root-tunnuksen salasana. Red5-palvelimella sijaitsevassa dispatcher-servlet.xml-nimisessä tiedostossa määritellään tietokannan tunnukset. Mikäli tiedostossa on eri tunnukset kuin ne, jotka määriteltiin aiemmin, niin tulee ne muuttaa.

Tietokannan luontiin käytettiin skriptitiedostoa.

1. Avataan MySQL Workbench.

2. Luodaan yhteys tietokantaan. Database > Connect to Database > Syötä käyttäjätunnus ja salasana.

3. Ajetaan skriptitiedosto. File > Run SQL Script… > Valitse skripti tiedosto > Run.

4.4 IIS:n asennus

Palomuurin asetuksista avattiin tarvittavat portit sovelluksen käyttöä varten. Tämän jälkeen asennettiin IIS (Internet Information Services) manager.

1. Ohjauspaneeli > Ohjelmat > Ota Windows ominaisuuksia käyttöön tai poista niitä käytöstä > Valitse IIS > OK.

2. Hiiren oikealla tietokoneesta > hallitse > Tools > IIS-palveluiden hallinta

Palvelua varten tarvittiin varmenne. Testipalvelimen käyttöä varten riitti, että luotiin itse allekirjoitettu varmenne.

3. Kun ollaan IIS-palveluiden hallinnassa, kaksoisnäpäytetään localhost > Palvelinvarmenteet > Luodaan itse allekirjoitettu varmenne… > Annetaan varmenteelle nimi ja OK

4. Sivustot > Hiiren oikealla Default Web Site > Sidonnat…

   Lisätään sivun sidontoihin http- ja https-portit. Aiemmin luotua varmennetta tarvitaan https-portin sidonnan yhteydessä.

5. Sivustot > Hiiren oikealla Default Web Site > Perusasetukset…

   Asetetaan fyysiseksi poluksi Red5-palvelimessa sijaitseva ROOT-kansio.

Yhteys nimellä… > Määritelty käyttäjä > Määritä > Aseta käyttäjänimeksi ”IUSR” ja jätä salasana tyhjäksi > OK.

4.5 Palvelimen käynnistäminen

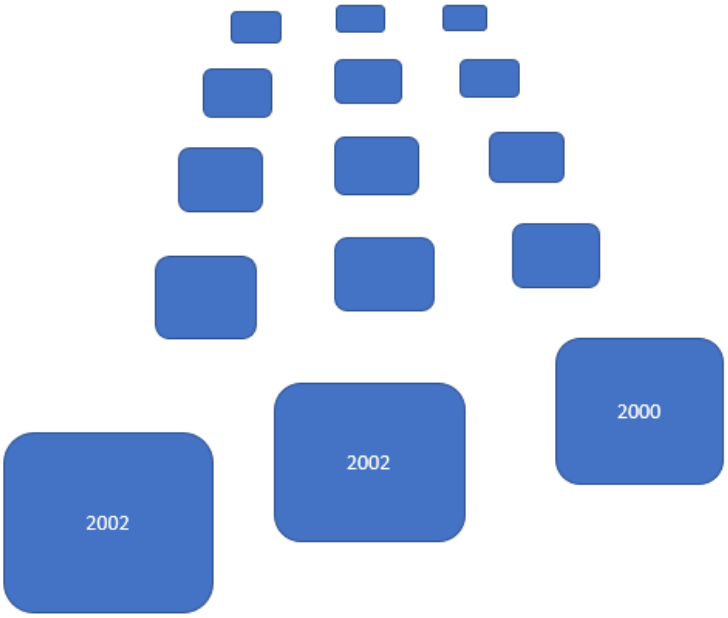
Red5-palvelin käynnistettiin juurihakemistossa sijaitsevasta red5.bat-tiedostosta. IIS-palvelin käynnistettiin IIS-palveluiden hallinnasta. Tämän jälkeen Sovelluksen tulisi näkyä selaimessa etäkoneen ip-osoitteessa. Mikäli yhteys ei toimi, kannattaa käynnistää uudelleen Red5- ja IIS-palvelimet.

5 Jatkokehitys

Jatkokehityksen tärkeimpänä tarkoituksena oli luoda uusi käyttöliittymä. Suurin osa muutoksista tapahtui clouds.js-nimisessä javascript-tiedostossa. Muihin jatkokehityksen tehtäviin kuului verkkokaupan luonti sekä tietokantojen yhdistäminen.

5.1 Käyttöliittymä

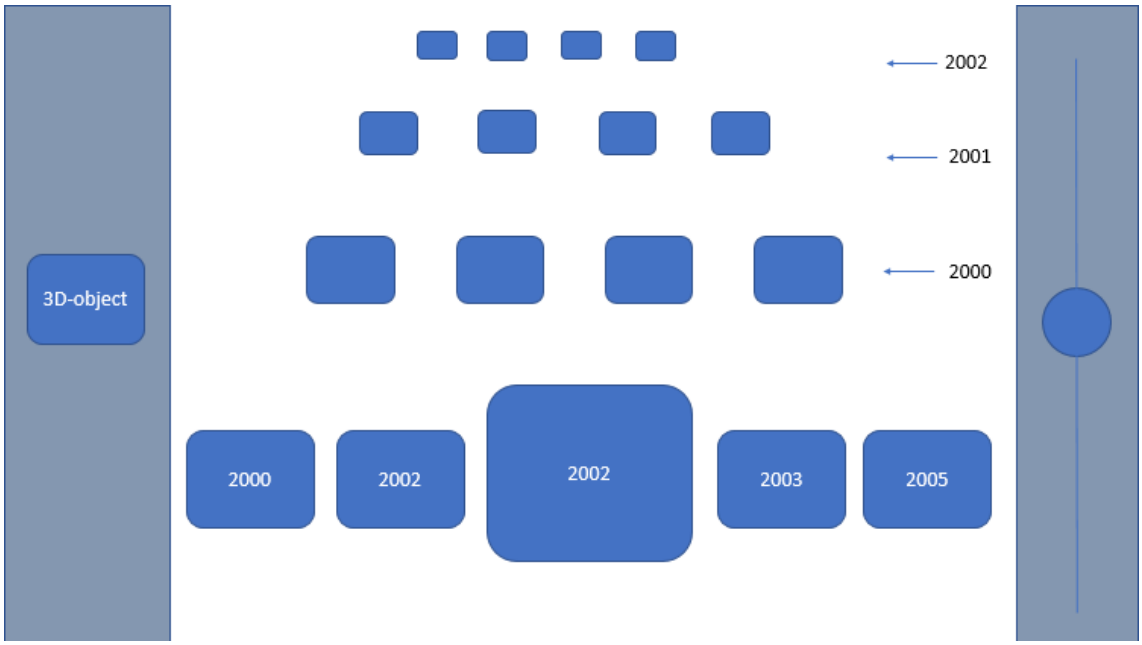
Vanhassa näkymässä tarinat kulkivat jonossa. Ongelma vanhassa näkymässä tuli esille tarinoiden lisääntyessä. Kaikki tarinat kulkivat yhdessä kasassa, eikä eri vuosien tarinoilla ollut selkeää erottelua. Tiettyyn vuoteen päästäkseen saattoi joutua etenemään aikajanalla pitkään.



Kuva 4 Vanha näkymä

Uudessa näkymässä viidestä lähimmästä tarinasta näytetään esikatselukuva tarinan sisällöstä ja keskimmäinen tarina on suurennettu. Vanhemmat tarinat on lajiteltu vuoden perusteella omille riveille. Mikäli samalla vuodella on paljon tarinoita, niin tietyn rajan ylimenevät tarinat muuttuvat läpinäkyviksi ja lopulta näkymättömiksi. Mitä kauempana tarinat ovat, sitä läpinäkyvämmäksi ne muuttuvat.

3D-objektit ja ankkuritarinat (esim. tärkeä tapahtuma kyseisenä vuotena) näytettiin erikseen vasemmalla omalla kaistallaan. Vanhassa näkymässä nämä objektit kulkivat muitten tarinoiden joukossa.



Kuva 5 Uusi näkymä

Uuteen näkymään tehtiin uusi vierityspalkki, joka skaalautui näytön mukaan. Vierityspalkissa näytettiin myös kunkin tarinarivin vuosiluvut.

5.2 Koodi muutokset

5.2.1 Clouds.js

Näkymä luotiin javascript-tiedostossa, jossa käytettiin WebGL-kirjastoa grafiikoiden luontiin. Vanhassa versiossa kameran positiota liikutettiin pitkin linjaa, jolle tarinat oli asetettu. Logiikan muuttaminen osoittautui hankalaksi vanhan koodin päälle. Poistin suuren osan vanhasta koodista ja vaihdoin logiikkaa, jolla näkymä toimii. Uudessa versiossa kamera pysyy paikoillaan ja tarinoiden sijaintia muutetaan.

```
for(var i = 0; i<storiesList.length;i++){
storiesList[i].meshIndex = i;
if(storiesList[i].year == xYear){
    if(storiesList[i].storyType=="story" && i > chestCounter+(displayStories-1)){
        xCounter++;
    }else if(i > chestCounter+(displayStories-1)){
        objectYearCounter++;
        storiesList[i].objectYear = objectYearCounter;
    }

}else{
    for(var j = 0; j<storiesList.length;j++){
        if(storiesList[j].year==xYear){
            storiesList[j].sameYears = xCounter;
        }
    }
    xCounter=1;
    objectYearCounter=0;
    xYear = storiesList[i].year;
    }
}
```

Kuva 6 Lasketaan eri vuosien lukumäärät

Kuvan 6 esimerkissä tarina-objekteihin lisätään muuttuja, jossa on lukumäärä tarinoista, joissa on sama vuosi. Tätä muuttujaa käytetään, jotta saadaan asetettua taustalla olevat tarinat oikeisiin paikkoihin. Laskuista jätettiin pois etuosan tarinat, koska niissä saattoi olla eri vuosien tarinoita.

```
if(i > chestCounter+displayStories-1+lisays){

                storyImageMesh[i].position.y= 100+((yearDifference)*100);
                storyTextMesh[i].position.y = 100+((yearDifference)*100);
                storyDateMesh[i].position.y = 100+((yearDifference)*100);

                storyImageMesh[i].position.z= -251-((yearDifference)*300);
                storyTextMesh[i].position.z = -250-((yearDifference)*300);
                storyDateMesh[i].position.z = -250-((yearDifference)*300);

                if(storiesList[i].year == xYear){
                    if(storiesList[i].storyType == "story"){
                        xCounter++;
                    }
                }else{
                    xCounter=1;
                    xYear = storiesList[i].year
                }

                storyImageMesh[i].position.x= -65+(xCounter*50)-((storiesList[i].sameYears-1)*25);
                storyTextMesh[i].position.x = -65+(xCounter*50)-((storiesList[i].sameYears-1)*25);
                storyDateMesh[i].position.x = -65+(xCounter*50)-((storiesList[i].sameYears-1)*25);

                if(storyImageMesh[i].position.x>150*(0.85+yearDifference)*(kuvasuhde/2) ||
                storyImageMesh[i].position.x<-200*(0.85+yearDifference)*(kuvasuhde/2)){

                    storyImageMesh[i].material.visible=false;
                }else if(storyImageMesh[i].position.x>100*(0.85+yearDifference)*(kuvasuhde/2) ||
                storyImageMesh[i].position.x<-150*(0.85+yearDifference)*(kuvasuhde/2)){
                    storyImageMesh[i].material.opacity=0.15;
                }else{
                    storyImageMesh[i].material.opacity=0.8-(yearDifference*0.15);

                }
}
```

Kuva 7 Lasketaan tarinoiden sijainti

Tarinat lajiteltiin vuosien mukaan riveihin. Mitä suurempi ero etuosan tarinan vuodesta, sitä ylemmäksi sekä kauemmaksi tarinat asetettiin näkymään. Tarinat järjestetään keskitetysti näytölle. Reunojen ylimenevät tarinat muuttuvat näkymättömiksi. Kaukana näkyvät tarinat muuttuvat läpinäkyviksi, jotta näyttäisi, että tarinat katoavat sumuun.

```
if(displayStories==5){
    if(enlargeCounter==2){
    storyWriterMesh[i].material.visible=true;
    storyImageMesh[i].scale.x = $("#threeChestTextSize").val()*1.5;
    storyImageMesh[i].scale.y = $("#threeChestTextSize").val()*2.25;
    storyTextMesh[i].scale.x = storyTextMesh[i].scale.y = $("#threeChestTextSize").val()*1.5;
    storyDateMesh[i].scale.x = storyDateMesh[i].scale.y = $("#threeChestDateSize").val()*1.5;
    storyWriterMesh[i].scale.x = storyWriterMesh[i].scale.y = $("#threeChestDateSize").val()*1.5;
    storyDateMesh[i].position.y = -25;
    storyTextMesh[i].position.x = -55;
    storyTextMesh[i].position.z = -100;

    }else{
    storyImageMesh[i].scale.x = $("#threeChestTextSize").val()*1;
    storyImageMesh[i].scale.y = $("#threeChestTextSize").val()*1.5;
    storyTextMesh[i].scale.x = storyTextMesh[i].scale.y = $("#threeChestTextSize").val();
    storyDateMesh[i].scale.x = storyDateMesh[i].scale.y = $("#threeChestDateSize").val();

    }
```

Kuva 8 Etuosan tarinat suurennetaan

Kuvassa 8 muutetaan etuosan tarinaobjektien kokoa ja sijaintia. Normaalinäkymässä etuosan tarinoita on viisi, mutta kuvasuhteesta riippuen tarinoita voi olla myös kolme tai yksi. Kaikkia näitä tarinoita suurennetaan, mutta keskimmäinen tarina on suurennettu muista isommaksi. Tarina-komponenttien kokoa pystyi muuttamaan admin-työkaluilla jQuery:n DOM-muuttujien avulla. [6.]

```
if(storiesList[i].isClose == false){

    var esikatselu = false;

    for(var j =0 ;j < storiesList[i].storyPages.length; j++){
            if(storiesList[i].storyPages[j].storyContent.type=="image"){
                storyImageMesh[i].material.map = THREE.ImageUtils.loadTexture(storiesList[i].
                storyPages[j].storyContent.content);
                storyImageMesh[i].material.needsUpdate = true;
                esikatselu = true;
                j = storiesList[i].storyPages.length;
            }
    }

    if(esikatselu == false){

        if(storiesList[i].storyPages[0].storyContent.type=="TEXT"){
        storyImageMesh[i].material.map = THREE.ImageUtils.loadTexture('resources/img/text_preview2.jpg');
        storyImageMesh[i].material.needsUpdate = true;
    }else if(storiesList[i].storyPages[0].storyContent.type=="audio"){
        storyImageMesh[i].material.map = THREE.ImageUtils.loadTexture('resources/img/audio_preview.jpg');
        storyImageMesh[i].material.needsUpdate = true;
    }else if(storiesList[i].storyPages[0].storyContent.type=="video"){
        storyImageMesh[i].material.map = THREE.ImageUtils.loadTexture('resources/img/video_preview.png');
        storyImageMesh[i].material.needsUpdate = true;
    }

    }
}
        storiesList[i].isClose = true;
```

Kuva 9 Esikatselukuvien lataaminen

Kuvassa 9 on esimerkki tarinaobjektin tekstuurin muutoksesta. Lähellä oleviin tarinoihin lisättiin esikatselukuva, mikäli tarinasta löytyi kuva joltakin sivulta. Muutoin esikatselukuvaksi asetetaan tarinan tyyppiä kuvaava kuva, jotka olivat teksti-, audio- sekä videotarina. Kuvatarinoista löytyi aina kuva, joten niitä ei tarvinnut lisätä.

```
function renderObject(x, y, z, width, height, src){

    objectGeometry = new THREE.Geometry();
    var objectTexture = THREE.ImageUtils.loadTexture(src, new THREE.UVMapping());

        var objectMaterial = new THREE.MeshBasicMaterial({
                map : objectTexture,
                transparent: true,
                opacity:1,
                depthTest: false
            });

            var shape = new THREE.PlaneGeometry(width,height);
            objectMesh[objectCounter] = new THREE.Mesh(shape, objectMaterial);
            objectMesh[objectCounter].position.x = x;
            objectMesh[objectCounter].position.y = y;
            objectMesh[objectCounter].position.z = z;

            THREE.GeometryUtils.merge(objectGeometry, objectMesh[objectCounter]);
            objectGroup.add(objectMesh[objectCounter]);
            scene.add(objectGroup);
            objectCounter++;
}
```

Kuva 10 Objektien luontifunktio

Etuosan tarinoita varten luotiin kehykset käyttäen kuvan 10 renderObject-funktiota. Kaikissa kehyksissä oli valkoiset reunat. Vasemmilla olevissa kehyksissä oli tumma linssi ja oikealla vaalea linssi joiden läpi näkyi esikatselukuva. Keskimmäisessä kehyksessä suurin osa esikatseluvasta näkyi selkeästi, mutta kehyksen alareunassa oli pieni tummennettu osio, johon päivämäärä oli asetettu. Tarinan otsikko nostettiin ylös, jolloin se ei enää ollut esikatselukuvan päällä.

```
function changeChestOnMouseHover()
{
    var vector = new THREE.Vector3( mouse.x, mouse.y, 1 );
    projector.unprojectVector( vector, camera );
    var ray = new THREE.Raycaster( camera.position, vector.sub( camera.position ).normalize() );
    var intersects = ray.intersectObjects( objects );

    if ( intersects.length > 0 )
    {
        if ( intersects[ 0 ].object != INTERSECTED )
        {
            INTERSECTED = intersects[ 0 ].object;
            INTERSECTED.currentHex = INTERSECTED.material.color.getHex();
            var id = intersects[ 0 ].object.index;

                    for(var j=0;j<storyImageMesh.length;j++){
                        if(j==id){
                            storyImageMesh[id].material.color.setHex( 0xA5C2D0 );
                            storyDateMesh[id].material.color.setHex( 0xA5C2D0 );
                            storyTextMesh[id].material.color.setHex( 0xA5C2D0 );
                        }else{

                            storyImageMesh[j].material.color.setHex(0xffffff);
                        if(storyTextMesh[j].position.x<-60){
                            storyDateMesh[j].material.color.setHex( 0xffffff );
                            storyTextMesh[j].material.color.setHex( 0xffffff );
                            }else if(storyTextMesh[j].position.x<0){
                            storyDateMesh[j].material.color.setHex( 0xffffff );
                            storyTextMesh[j].material.color.setHex( 0x000000 );
                            }else{
                            storyDateMesh[j].material.color.setHex( 0x000000 );
                            storyTextMesh[j].material.color.setHex( 0x000000 );
                            }
                        }

                    }

            $("#canvas").css("cursor","pointer");
        }
    }
```

Kuva 11 Tarinaobjektin värin muutos

Kuvassa 11 tarinaobjektin väri muuttui vaalean sinertäväksi, kun kursoria pidettiin sen päällä. Muussa tapauksessa tarinaobjekti palautettiin alkuperäiseen väriin. Vasemmanpuoleisissa etuosan teksteissä käytettiin valkoista väriä, kun taas oikeanpuoleisissa mustaa. Keskimmäisen tarinaobjektin otsikon väri oli musta ja päivämäärä valkoinen. Värit olivat erilaiset, koska niitten tuli näkyä hyvin aiemmin mainittujen kehyksien päällä.

5.2.2 Genericpage.tag

Genericpage.tag-niminen tiedosto sisälsi ylä- ja alatunnisteen sisällöt, jotka näkyivät aina huolimatta siitä, mikä palvelun sivuista oli auki. Sivun taustakuva määriteltiin myös tässä tiedostossa.

```
<c:if test="${sessionScope.userSession != null}">
document.body.style.backgroundImage =
"url('http://[redacted]/resources/img/skybox_2.png')";
document.getElementById("header").style.backgroundImage =
"url('http://[redacted]/resources/img/top-repeat2.png')";
document.getElementById("footer").style.backgroundImage =
"url('http://[redacted]/resources/img/alapalkki3.png')";
</c:if>
```

Kuva 12 Taustakuvan vaihto JSTL:llä

Taustakuva sekä ylä- ja aulatunnisteen tausta vaihtuivat erilaiseksi, mikäli käyttäjä oli kirjautuneena palveluun. Istunnon tarkistamiseen käytetään JSTL:ää (JavaServer Pages Standard Tag Library), kuten kuvassa 12. [5.]

5.2.3 Home.jsp

Home.jsp-tiedostossa määriteltiin aikajananäkymä, joka oli myös ensimmäisenä aukeava perusnäkymä. Tiedosto sisälsi vierityspalkin, jota liikuttamalla näkymän vuosilukua muutettiin. Sivun oikeaan luotiin välilehdet, joista pääsi muuttamaan näkymää aikajananäkymän ja karttanäkymän välillä.

```
<div id="navslider" style="height:100%">
    <div id="nav-slider" style="max-height:100%;cursor: pointer; float:left;
    display:block; margin:auto"></div>
</div>

<div id="navkartta">
    <img id="map-button" href="#" onclick=""src="/resources/img/aikajana-yläosa.png"
    style="object-fit:cover; height:45vh; width: 5vw; cursor:pointer" >
</div>

<div id="navaika">
    <img id="map-button" href="#" onclick=""src="/resources/img/aikajana-alaosa.png"
    style="object-fit:cover; height:45vh; width:5vw" >
</div>
```

Kuva 13 Skaalautuvat painonapit

Sivun laidassa olevat välilehdet sekä vierityspalkki skaalautuvat ikkunan koon suhteen. Skaalautuva tyyli saadaan aikaan käyttäen viewport-yksiköitä vh ja vw. [7.] Muut div-osioiden ominaisuudet määritellään erillisessä css-tiedostossa, kuten kuvan 14 esimerkissä.

```
#navkartta {
    position: absolute;
    right: 0px;
    top: 5%;
    bottom: 45%;
    left: 95%;
    text-align: right;
    z-index: 0;
    padding:0;
    marigin:0;
}
```

Kuva 14 navkartta css-tyyli

5.2.4 Muut muutokset

mapView.jsp-tiedostossa määriteltiin karttanäkymä. Tarina objektit sijoitettiin Google-Maps:n kartalle. Koodi muutokset tähän tiedostoon olivat samanlaisia kuin aikanäkymässä. Karttanäkymän vierityspalkki toimi zoomaus-toimintona vuosien sijasta.

```
.ui-slider-handle {
    background-image: url(../img/sliderpalkki5.png) !important;
    background-repeat: no-repeat !important;
    width: 175px !important;
    height: 175px !important;
    line-height: 175px;
    vertical-align: middle;
    background-color: transparent !important;
    border: 0px !important;
    margin-left: -5px !important;
    font-size: 22px;
    text-align: center;
    font-weight: bold !important;
    outline: none;
    font-family: "Courier New";
}
```

Kuva 15 Vierityspalkin css-tyyli

Sivuston tyylit saatettiin määritellä useassa eri paikassa, mutta suurin osa tyyleistä määriteltiin Style.css-tiedostossa. ”!important”-merkinnällä varmistetaan, että kyseistä ominaisuutta käytetään, mikäli kyseiselle kohdalle on olemassa useita tyylivaihtoehtoja.

```
#footer {
    width: 100%;
    background-image: url(../img/alapalkki2.png);
    background-repeat: repeat-x;
    height: 100px; /*115px*/
    text-align: center;
    position: absolute;
    bottom: 0px;
    left: 0px;
    z-index: 10;
}
```

Kuva 16 Alatunnisteen css-tyyli

Ylä- ja alatunnisteen taustoja monistetaan background-repeat-muuttujalla. Käytetty kuva oli hyvin ohut, jota sitten monistettiin vaakasuunnassa repeat-x muuttujan avulla, kuten kuvan 16 esimerkissä.

5.3 Verkkokauppa

Verkkokaupan ohjelmointi jäi kesken, mutta sain tehtyä jonkinlaisen pohjan verkkokaupalle. Tutustuin myös Paytrail-maksupalvelun toimintaan.

```
<div class="row">
<div class="col-md-3" ><img src="/resources/img/[redacted]_vip.png" height="100" width="100"></img></div>
<div class="col-md-7">VIP [redacted] tarjoaa käyttäjälle:<br>
                1.Rajattoman määrän yksityisiä tarinoita.<br>
                2.Elinikäisen käyttöoikeuden.<br>
                3.Osallisuuden [redacted]-käyttäjien ydinryhmässä(Kannatusjäsenet)</div>
<div class="col-md-2 hd-btns home_button"><a href="buyingForm2.jsp">100€</a></div>
</div>
```

Kuva 17 Ostettava tuote

Tuotevalikon ohjelmoinnissa käytettiin bootstrap-kehystä. Kuvan 17 koodiesimerkin painonappiin ei ole lisätty vielä Paytrail-linkkiä.

Maksun suorittamiseen käytetään Paytrail-verkkomaksupalvelua. Paytrail on suomalainen maksulaitos, joka kuuluu osaksi pohjoismaista Nets-perhettä.

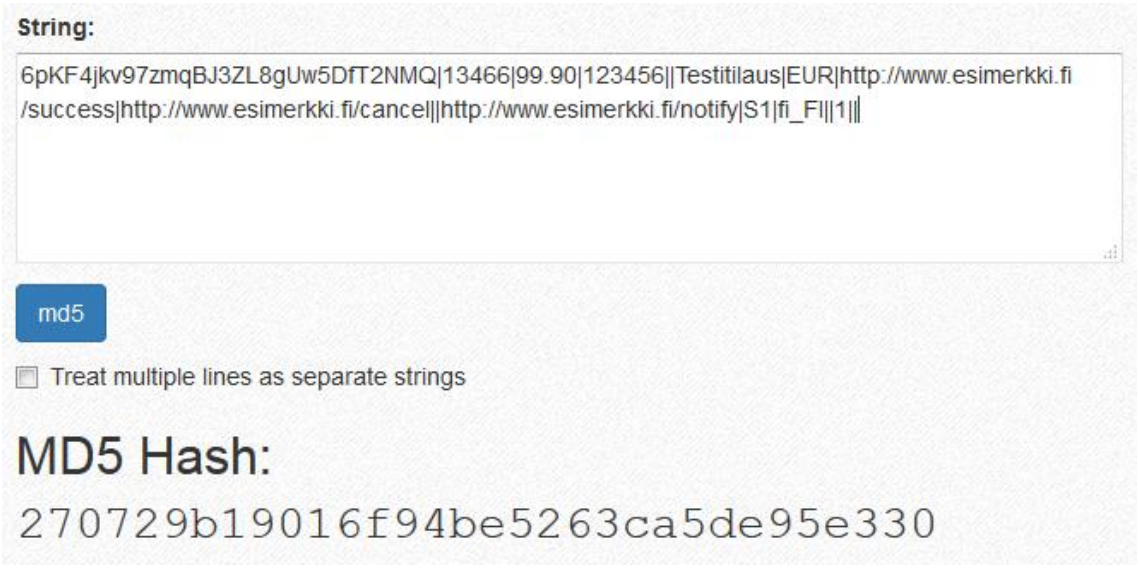
Paytrail valittiin maksutavaksi koska se sisältää suuren valikoiman maksutapoja. Se sisältää kaikkien kotimaisten pankkien verkkomaksupainikkeet, joihin kuukuu: Nordea, Osuuspankki, Danske Bank, Säästöpankki, Oma Säästöpankki, POP Pankki, Aktia, Handelsbanken, Ålandsbanken ja S-Pankki, sekä korttimaksu palveluita: Visa, Visa Electron, Mastercard, American Express, Eurocard, Diners Club ja JCB. Suurin osa palvelumme käyttäjistä on suomalaisia, joten kotimaisuus oli myös yksi peruste verkkomaksupalvelun valintaan, sillä se lisää luotettavuutta.

Maksupainike määriteltiin kuvan 18 mukaisesti. [3.] Nappia painamalla käyttäjä siirtyi Paytrailin sivulle, jossa maksu voitiin suorittaa. Maksun jälkeen käyttäjä siirtyi lomakkeessa määritetylle paluusivulle tai maksun peruutussivulle. Maksutapahtuman toiminnan kannalta tärkein muuttuja on AUTHCODE, eli koodi joka todistaa maksutapahtuman oikeaksi.

```
<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="S1">
  <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>
```

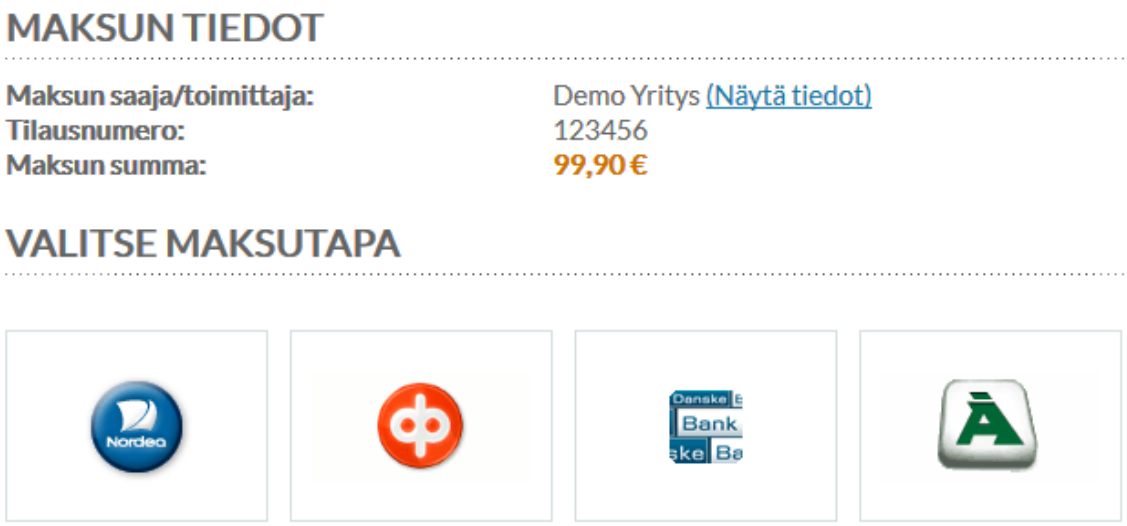
Kuva 18 Paytrail maksupainikkeen esimerkki

AUTHCODE saatiin käyttämällä md5 Hash kääntäjää. [4.] Kääntäjään syötettiin merkkijono, joka koostui kauppiassalaisuudesta (merchant secret) mikä oli Paytrail:ltä saatu henkilökohtainen koodi, joka yhdisti maksutapahtuman omaan tiliin, sekä lomakkeessa olevista arvoista, jotka oli erotettu pysty viivalla. Koodin kirjaimet piti muuttaa isoiksi, jotta linkki toimisi.



Kuva 19 MD5 Hash kääntäjä

Painiketta painettaessa siirrytään Paytrailin sivulle. Tällä sivulla näkyy maksun saaja, tilausnumero sekä maksun summa. Sivulla tulee valita maksutapa, jonka jälkeen siirrytään valitsemasi verkkopankin sivulle maksamaan tilausta.



Kuva 20 Paytrailin maksutapahtuma

Maksun jälkeen siirrytään takaisin lomakkeessa määritetylle paluusivulle. Tilauksen tiedot näkyvät URL:issa kyselyinä (Query String).

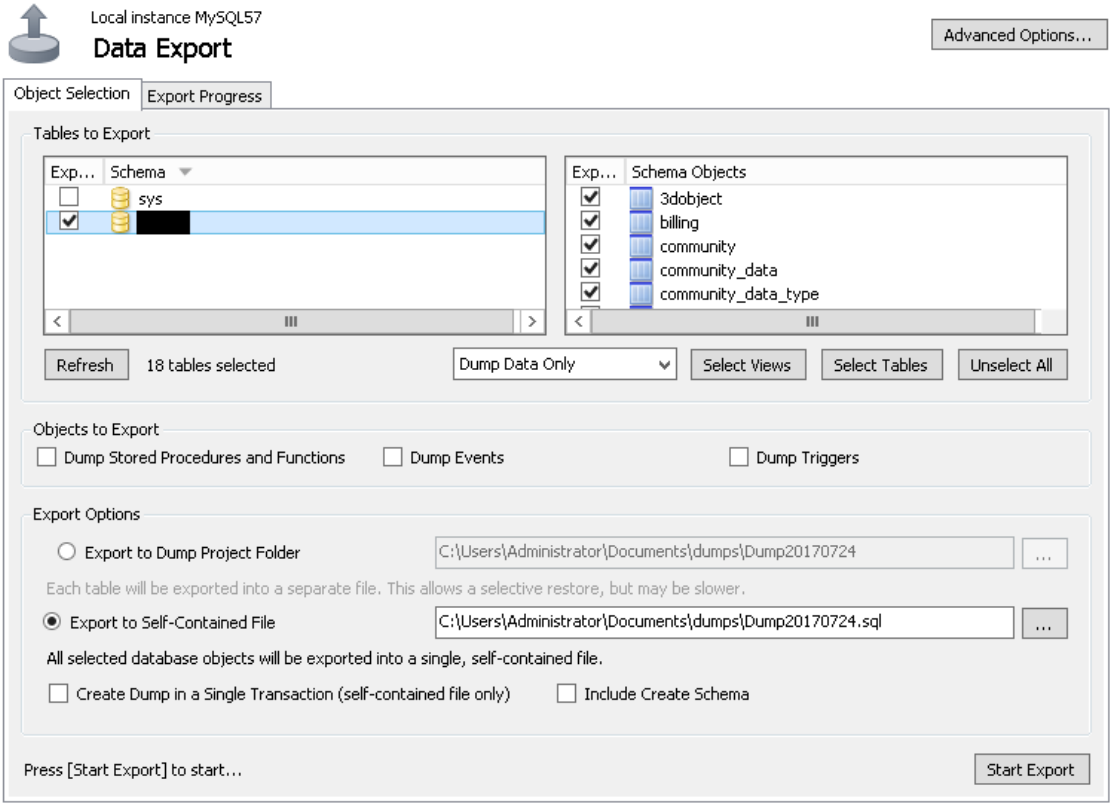
5.4 Tietokantojen yhdistäminen

Palvelusta oli olemassa kaksi eri versiota. Vanhempi versio sisälsi tarinoita, jotka oli tarkoitus siirtää uuteen versioon. Tietokannat kuitenkin erosivat hiukan toisistaan ja tiedostoilla saattoi olla sama id, joka aiheutti ongelmia.

```
SET FOREIGN_KEY_CHECKS = 0;
UPDATE story  SET id = id + 100 where id >= 1204 and id <= 1226;
SET FOREIGN_KEY_CHECKS = 1;
```

Kuva 21 MySQL-tietokannan ID-muunnos

Tietokantojen id:ien ollessa samoja kasvatettiin niitten arvoa riittävän suuresti kuvan 21 kaltaisella lauseella.



Kuva 22 MySQL-tietokannan kopioiminen

Kuvan 22 ”Data Export” toiminnolla tietokannasta voitiin luoda sql-tiedosto. Ennen tietokantojen yhdistämistä alkuperäisestä tietokannasta otettiin talteen sql-tiedosto, jotta tietokanta voidaan palauttaa vahingon sattuessa.

Lisättävästä tietokannasta otettiin vain data valitsemalla ”Dump Data Only”. Valitsemalla ”Export to Dump Project Folder” tietokannan jokaisesta taulukosta luotiin erikseen omat tiedostot kansioon. Tämä vaihtoehto saatettiin valita, mikäli kaikkia taulukoita ei haluttu lisätä. Tietokanta voitiin tallentaa kokonaisuudessaan yhteen tiedostoon valitsemalla ”Export to Self Contained File”. Ajan säästämiseksi käytettiin tätä vaihtoehtoa. Tästä luotu skriptitiedosto ajettiin alkuperäisessä tietokannassa.

6 Yhteenveto

Tavoitteena oli luoda uusi käyttöliittymä sekä verkkokauppa sovellukselle. Sain luotua uuden käyttöliittymän. Uutta versiota ei ole kuitenkaan vielä viety julkiseen versioon muutamien skaalauksesta aiheutuvien bugien vuoksi. Verkkokaupan ohjelmointi jäi kesken, mutta sain tehtyä alustavaa suunnittelua jatkoa varten.

Työn aikana opin pystyttämään pilvipalvelun, käsittelemään tietokantoja sekä tekemään skaalautuvia web-sivuja. Kehityin toisen tekemän koodin tulkitsemisessa ja bugien etsimisessä.

Ohjelman kehitystä tullaan jatkamaan tämän dokumentoinnin jälkeen. Tämä opinnäytetyö toimii myös ohjeena palvelun kehittämistä jatkaville henkilöille. Jatkokehityksessä tullaan keskittymään verkkosovelluksen kaupallistamiseen ja käytettävyyden parantamiseen.

Työskentely tämän sovelluksen parissa vaatii kykyä tulkita toisten tekemää koodia sekä kykyä työskennellä itsenäisesti. Sovelluksen toimintaan tutustuminen saattaa viedä jonkin verran aikaa, mutta työskentely on joustavaa.

Lähteet

1 Auvinen, Risto. 2016. Verkkosovelluksen jatkokehitys. Insinöörityö. Metropolia Ammattikorkeakoulu.

2 Lähteenmaa, Juha. 2016. Verkkosovelluksen ohjelmistoteknologiat. Insinöörityö. Metropolia Ammattikorkeakoulu.

3 Paytrail – Integration guide. Verkkodokumentti. http://docs.paytrail.com/en/index.html. Luettu 14.8.2017

4 md5 Hash Generator. Verkkodokumentti. http://www.miraclesalad.com/webtools/md5.php. Luettu 14.8.2017

5 JSP – Standrad Tag Library (JSTL) Tutorial. Verkkodokumentti. https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.html. Luettu 14.8.2017

6 jQuery Tutorial. Verkkodokumentti. https://www.w3schools.com/jquery/default.asp. Luettu 14.8.2017

7 CSS Units. Verkkodokumentti. https://www.w3schools.com/cssref/css_units.asp. Luettu 14.8.20