VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Joonas Kevari

INTERNETKAUPPA WEB-SOVELLUKSELLE

Tekniikka
2017

VAASAN AMMATTIKORKEAKOULU
Tietotekniikka

TIIVISTELMÄ

Tekijä	Joonas Kevari
Opinnäytetyön nimi	Internetkauppa Web sovellukselle
Vuosi	2017
Kieli	suomi
Sivumäärä	41
Ohjaaja	Pirjo Prosi

Taktiko Solutions Oy, joka toimii tuotteen lopullisena tilaajana, on kehittänyt taktisen valmennuksen web-sovelluksen. Sen tarkoituksena on toimia valmentajan ja pelaajan apuvälineenä taktisen valmennuksen osa-alueella. Tälle sovellukselle toteutettiin työssä esitelty verkkokauppa.

Opinnäytetyö toteutettiin Devatus Oy:lle. Työssä valmistettiin kauppa kaikkine toimintoineen Taktiko nimiselle web-sovellukselle. Tuote toimii internetissä, jolloin kaupan integroiminen tuotteeseen oli yksi vaatimuksista.

Tuote toimii Meteor Frameworkin pohjalla, joten kauppa tuotettiin sen teknologioita hyödyntäen. Muina riippuvuuksina oli tilaajan käyttämä palvelu, jonne kaikki maksut kirjattiin. Palvelun nimi oli Checkout.fi, joten kaupan tuli myös hyödyntää Checkout.fi:n tuottamaa maksunvälitysrajapintaa.

Valmiissa kaupassa ostajan onnistunut tai epäonnistunut maksusuoritus palautetaan GET -metodilla alkuperäiselle sivulle, josta parametrien lukemisen jälkeen voidaan joko evätä tai lisätä käyttöoikeuksia. Kaupan tuli olla myös turvallinen sekä ostajan että tuotteen myyjän osalta. Tästä syystä kauppaan toteutettiin monimutkainen maksunvarmistusprosessi, jotta maksun kirjautumista ei voida huijata. Näin tuotetta ei saada käyttöön, ilman että maksu on täysin oikein kirjautunut järjestelmään.

Taktiko solutions Oy on myös kasvuopen finalisti vuodelta 2016.

Avainsanat Meteor Framework, Web-sovellus, Taktiko

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietotekniikka

ABSTRACT

Author	Joonas Kevari
Title	Webshop for web application
Year	2017
Language	Finnish
Pages	41
Name of Supervisor	Pirjo Prosi

Taktiko Solutions Oy, which is also the final client of the work, has developed a web application for tactical training. Taktiko Solutions is also a KasvuOpen finalist from the year 2016. The main objective of the application is to be a tool for players and coaches to help both to understand the area of tactical coaching. This work presents a web shop which was built into the web application.

This Bachelor's thesis was made for Devatus Oy and a full featured web shop was created into a Taktiko web application. The application operates on the internet and is the reason why the web

shop integration into the web application was one of the most important requirements for the web shop. The application uses Meteor Framework and therefore the web shop uses the same technology as the application itself. The other dependency was API, which was used by Taktiko Solutions. All payment transactions go through a service called Checkout API and that is also the way the application registers all payment transactions.

In the web shop, both successful and unsuccessful payment transactions will be returned to the front page of the application with POST -method. After the parameters analysis the product automatically sets new license for the customer or sends an error code onto the screen. For security reasons the web shop has a complex verification and validation process for avoiding abuses. This prevents application usage without payment.

Keywords Meteor Framework, Web Application, Taktiko

SISÄLLYS

KUVA- JA TAULUKKO LUETTELO

LYHENTEET JA TERMIT

METEOR	Meteor Framework, täydenpinon sovellusalusta
JS	Javascript ja javascript ohjelmointikielen tiedostoformaatti
CSS	Cascading Style Sheets tyylittelydokumentti
CLIENT	Asiakas, käyttäjä
MD-5	Tiivistealgoritmi
MDG	Meteor Development Group, meteorin kehittäjäryhmä
NODE	V8 JavaScript moottoriin pohjautuva kehitysalusta
DDP	Distributed Data Protocol, tiedonvälitysprotokolla
JSON	JavaScript Object Notation, avoimenstandardin tiedostomuoto tiedon välityksessä
DOM	Document Object Model, dokumenttioliomalli
E2E	End-to-end, palvelimelta käyttäjälle
NPM	Node Package Manager, pakettien hallintatyökalu
NoSQL	Not only SQL tietokanta, perinteisestä relaatiotietokannasta poikkeava tietokanta
HTTP	Hypertext Transfer Protocol, hypertekstin siirtoprotokolla
ECMA	Kansainvälinen standardointijärjestö
W3C	World Wide Web Consortium, WWW standardeja ja suosituksia toteuttava yhteisö

1 JOHDANTO

Tässä työssä työn tekijä tuottaa toimeksiantajalleen web-kaupan. Web-kauppa integroidaan osaksi uutta ja innovatiivista valmennustyökalua, jonka Taktiko Solutions on kehittänyt. Web-kauppa mahdollistaa tuotteen myynnin suoraan sen asiakkaille ilman erillistä yhteydenottoa.

Työn tekijän toimeksiantajana toimii Devatus Oy, joka puolestaan on saanut toimeksiannon Taktiko Solutions Oy:ltä.

1.1 Taktiko Solutions Oy

Taktiko Solutions Oy on yritys, joka perustettiin vuonna 2015 tuotteistamaan, sekä kehittämään palloilulajeille tarkoitettua taktisen valmennuksen työkalua. Tuotteen nimi on Taktiko ja se mahdollistaa peliymmärryksen oppimisen interaktiivisen vuorovaikutuksen avulla. /1/

Tuote pyrkii hyödyntämään markkinarakoa, joka taktisen valmennuksen työkaluille on olemassa. Tuote on kehittyneempi versio ”fläppitaulusta” ja mukana on paljon ominaisuuksia, joita fläppitaulu ei pysty tuottamaan. /1/

Taktikossa on mahdollista luoda erilaisia harjoitteita, taktiikoita tai pelata livetilassa yhdessä muun joukkueen kanssa. Taktiko mahdollistaa myös harjoitteiden nauhoittamisen. Kaikki harjoitteet, taktiikat ja nauhoitukset on mahdollista jakaa yhdelle tai useammalla henkilölle sosiaalisen median tai sähköpostin avulla. Tuote sopii niin yksittäiselle valmentajalle kuin suurille seuroille.

Taktiko Solutions Oy menestyi myös kasvuopenissa ja pääsi aina finalistiksi saakka.

1.2 Devatus Oy

Työn tekijän toimeksiantaja, Devatus Oy, on nopeasti kasvava ohjelmistokehitystalo, joka palvelee asiakkaidensa muuttuvia tarpeita joustavasti. Yrityksellä on vahva osaaminen ohjelmistokehityksestä, mobiili- ja verkkosovellus kehityksestä,

sekä integrointi- ja projektinjohtamisesta. Yrityksen henkilöstömäärä on tällä hetkellä alle 20 ja se on täysin työntekijöidensä omistuksessa. /2/

2 TEKNOLOGIAT

Tässä osassa esitellään työssä käytettyjä teknologioita. Työssä käytetyt teknologiat ovat välttämättömiä, jotta voidaan saavuttaa haluttu tulos sekä web-kaupan että itse tuotteen osalta. Työssä esitellyt web-teknologiat ovat nykyaikaisia ja niitä ylläpidetään jatkuvasti. Ylläpidettävyys on tärkeä osa tuotteen elinkaarta.

2.1 Meteor.js

Meteor on täydenpinon (full-stack) web-sovelluskehys, joka toimii sekä palvelimena että rajapintana sovellukselle ja sisältää myös käyttöliittymän. Meteorin kehittäjänä toimii Meteor Development Group (MDG). MeteorJS:n avulla on mahdollista kehittää web- ja mobiilisovelluksia. /3/

Meteor on kehitetty Node.js -pohjaisen palvelimen päälle, joka antaa mahdollisuuden toteuttaa asiakas- ja palvelinpuolen samalla ohjelmointikielellä, JavaScriptillä. Toteutus näkyy myös kooditasolla saakka, sillä Meteorin ratkaisuissa on mahdollista kirjoittaa asiakas- ja palvelinpuolen koodia samaan tiedostoon. Ne voidaan erottaa isClient ja isServer -muuttujien avulla. /3/

Käyttöliittymän toteutuksessa käytetään Meteorin omaa Blaze -sivupohjamoottoria, joka on suunniteltu hyödyntämään reaktiivisuutta. Käyttöliittymä on mahdollista toteuttaa myös Angular- tai React-sovelluskehysten sivupohjilla. /3/

Meteorilla on myös aktiivinen yhteisö, joka kehittää omatoimisesti Meteorin liitännäisiä kirjastoja ja osia. /21/

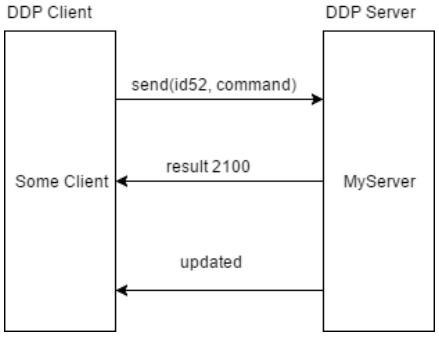
2.1.1 Reaktiivisuus

Etuna Meteorissa on sen reaktiivisuus, eli reaaliaikaisuus, joka tarkoittaa tehtyjen muutosten näkymistä välittömästi selaimessa. Reaktiivisuudella tarkoitetaan, että selaimen pitää pystyä näyttämään erilaista dataa, mutta samalla myös kuuntelemaan mahdollisia muutoksia. /3/

Reaktiivisuus on yksi tuotteen tärkeimmistä vaatimuksista, sillä tuotteessa tulee pystyä tuottamaan animaatioita ja esittelemään pelaajien liikkeitä, sekä piirtämään erilaisilla piirtotyökaluilla. Ilman reaktiivista käyttöliittymää, tuotteen tärkeimmät ominaisuudet ja vaatimukset jäävät toteuttamatta. Jotta useamman samanaikaisen käyttäjän toteuttamat liikkeet näkyvät muille, on käytettävä reaktiivisuutta tukevaa teknologiaa, jonka tässä tapauksessa Meteor tarjoaa.

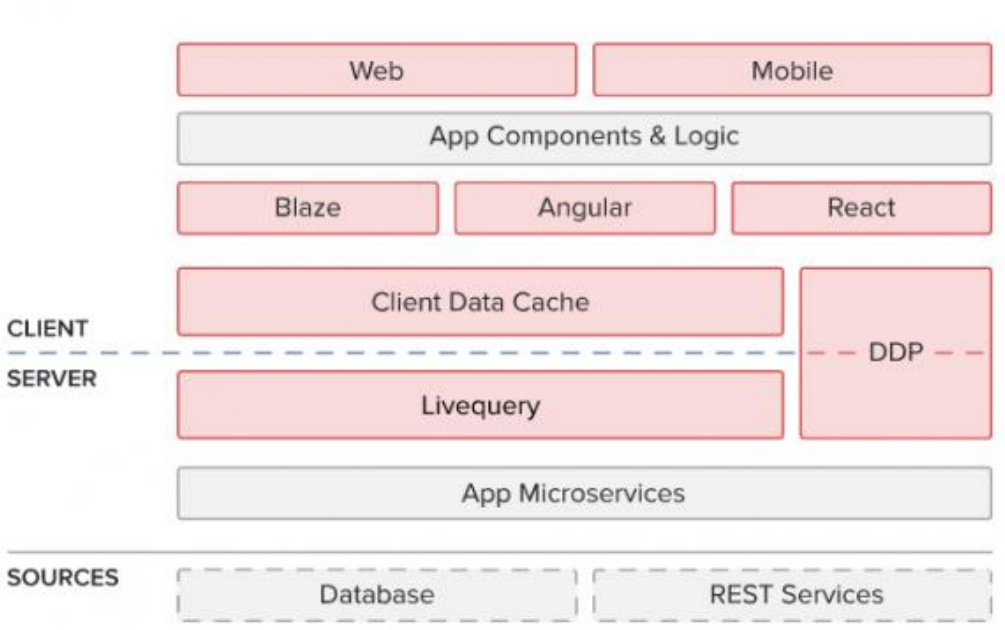
2.1.2 Rakenne

Kuvassa 1 esitellään meteorin tiedonvälityksessä käytettyä DDP-protokollaa (Distributed Data Protocol), joka on luotu meteoria varten. Se perustuu Web Sockets rajapintaa toteuttavaan SockJS -kirjastoon. Protokolla käyttää määrätyn muotoista JSON-viestiä, jotta se voi välittää tietoja asiakkaan ja palvelimen välillä. Protokollassa on myös se etu, että DDP ilmoittaa kutsujalle, kun kaikki kirjoitusoperaatiot on heijastettu muille yhdistetyille asiakkaille. /5/



Kuva 1. DDP-protokolla /20/

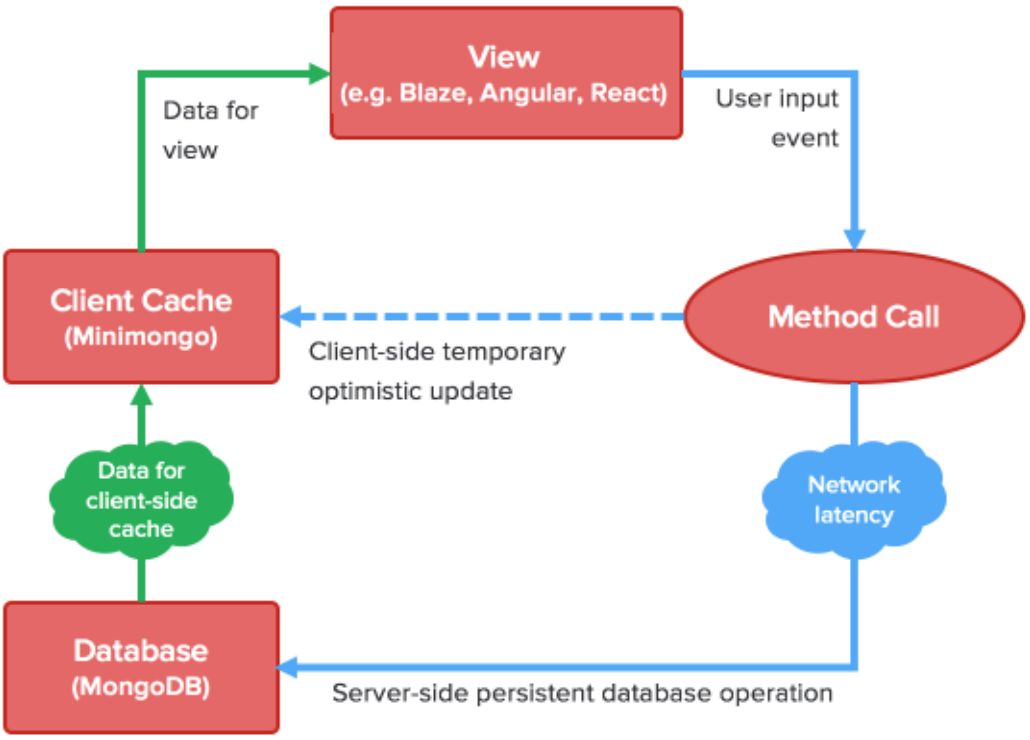
Tietojen synkronointi puolestaan perustuu julkaise tila-periaatteeseen. Se tarkoittaa, että asiakas kuuntelee Meteorin tietokantana käytettyä MongoDB -tietokantaa. Kuvattu toiminnallisuus on nimeltään Livequery. Jotta tietoa voidaan nähdä käyttöliittymässä, on Meteor Tracker -komponentin päivitettävä muutokset DOMiin. /5/



Kuva 2. Meteor.js kuvaus /4/

Meteor on suunniteltu reagoimaan nopeasti muutoksiin myös tilanteissa, joissa verkon latenssi olisikin suuri. Tämä on toteutettu siten, että muutos tehdään ensin selaimen paikalliseen JavaScript -pohjaiseen Minimongo -tietokantaan. Tätä myös kutsutaan nimellä optimistinen käyttöliittymä (Optimistic UI). /5/

Käyttäjän tekemä muutos tallennetaan kahteen kertaan, joista ensimmäinen tallennetaan paikalliseen minimongo-tietokantaan ja toinen lähetetään palvelimelle. Päivitys käyttöliittymään tapahtuu vertaamalla näitä kahta aiemmin mainittua tietokantaa keskenään, jonka jälkeen korvataan asiakkaan puolelta tiedot vastaamaan palvelimelta saatavaa arvoa. /5/



Kuva 3. Optimistinen käyttöliittymä /5/

2.1.3 Node.js

Node.js (myöhemmin Node) on suunniteltu web-sovellusten kehitykseen ja sen moottorina toimii JavaScriptin virtuaalikone. Node pohjautuu Googlen V8 JavaScript-moottoriin. Koodi kirjoitetaan JavaScriptillä, jonka jälkeen V8 kääntäjä kääntää koodin koneelle ymmärrettäväksi. Noden palvelinosan ohjelmointi voidaan kirjoittaa kokonaan JavaScriptillä. /6/

Kehitysalustana Nodea voidaan kehua hyvänä monestakin syystä, mutta suurimpina etuina ovat erityisesti jo aiemmin mainittu JavaScriptin laaja käyttömahdollisuus. Tämän lisäksi Nodella on toimiva logiikka käsitellä erilaisia pyyntöjä. Nodessa ei käsitellä pyyntöjä useassa säikeessä, vaan ne usein käsitellään samassa säikeessä. Tämä antaa Node-palvelimelle mahdollisuuden mukautua hyvin erilaisiin tilanteisiin, joissa pyyntöjen määrä ja koko vaihtelevat. /6/

2.1.4 MongoDB

MongoDB on nopea ja mukautuva NoSQL-tietokanta. Mongo sana tulee englanninkielisestä sanasta ”humongous” (valtava/suunnaton), joka kuvaa myös sen mukautuvuutta ja tehokkuutta. Se pohjautuu NoSQL -pohjaiseen dokumenttien säilytysmalliin, joka tarkoittaa, että tieto on tallennettu tietokantaan JSON muodossa, kun perinteisessä tietokannassa tiedot on tallennettu riveihin ja sarakkeisiin. /7/

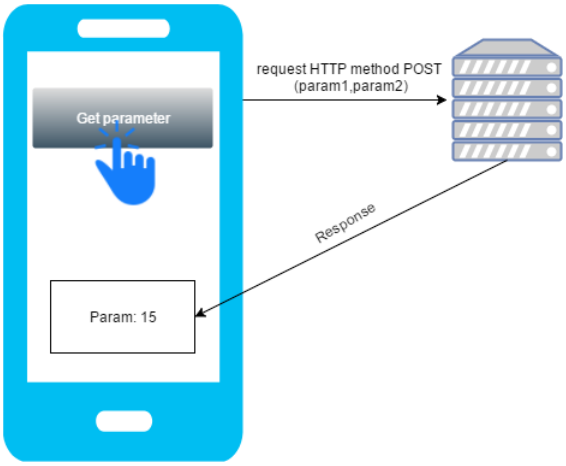
MongoDB tarjoaa hyvän tietokannan web-sivuille, joissa on kovaa liikennettä, koska se kykenee mukautumaan nopeasti muuttuviin tilanteisiin ja käsittelemään tietoa myös nopeasti. Se sopii myös erinomaisesti sivuille, joissa pitää tallettaa käyttäjien kommentteja, blogeja tai muita tietoja. Se on myös helppo implementoida web-sovellukseen. MongoDB on yksi tehokkaimmista tietokannoista. /7/

2.2 HTTP metodit

Tässä osassa esitellään HTTP:n (Hypertext Transfer Protocol) GET- ja POST-pyynnöt ja niiden toiminta.

HTTP on suunniteltu mahdollistamaan kommunikaatio asiakas- ja palvelinpuolten välillä. Se toimii myös pyyntö-vastausprotokollana näiden asiakas-ja palvelinpuolten välillä. POST metodilla viedään tietoa palvelinsovellukselle ja GET puolestaan tuo tietoja palvelinsovellukselta.

Esimerkkinä voidaan pitää tilannetta, jossa asiakaspää voi lähettää HTTP pyynnön toiselle palvelimelle POST-metodilla. On tärkeää muistaa, että POST-metodin tiedot eivät jää selainhistoriaan ja niitä ei voi erikseen tallentaa kirjamerkkeihin. Hyvä puoli POST-metodissa on se, että sillä ei ole rajoituksia datan pituuden osalta. /9/



Kuva 4. HTTP POST

GET-metodissa puolestaan voidaan tietty polku laittaa kirjanmerkkeihin ja tallentaa vaikkapa koko GET-metodi kirjanmerkkeihin. Suurin ero GET- ja POST-metodilla on näkyvyyden osalta se, että POST-metodin dataa ei näytetä, mutta GET-metodin data on näkyvissä URL:ssa. Palvelin voi GET-metodilla palauttaa asiakasselaimelle takaisin erilaisia parametreja. Tämä saattaa aiheuttaa tietoturvariskejä ja tämän vuoksi ei koskaan pidä käyttää GET-metodia lähetettäessä tärkeää informaatiota, kuten käyttäjän kirjautumistietoja. /9/

2.3 MD5

Ronal Rivest on kehittänyt useita erilaisia tiivistealgoritmeja, joita ovat mm. MD4 ja tässä osassa esitelty MD5. Rivest on tietotekniikan professori, joka oli kehittämässä useita erilaisia salausmenetelmiä (mm. RC2,RC4, RSA ja MD5). /18/

MD5 algoritmi tuottaa tuloksena 128 bittisen tiivisteen, jota voidaan kutsua ”sormenjäljeksi”, koska on lähes mahdotonta tuottaa kahta samanlaista sormenjälkeä. MD5 algoritmia pyritään yleensä käyttämään eräänlaisena digitaalisena allekirjoituksena sovelluksille, joissa suuri määrä tietoa pitää tiivistää ennen sen salaamista.

MD5:lla on heikkouksia verrattuna MD4:ään. MD5 on hieman hitaampi kuin aiempi versio (MD4), mutta se on todettu olevan turvallisempi. Vaikka MD5 voidaan pitää vanhana, on se silti edelleen toimiva ratkaisu tiivistämään tärkeää tietoa. /10/

2.4 Ohjelmointikielet

Ideaalinen ohjelmointikieli on sellainen, joka mahdollistaa kehittäjän toteuttaa ohjelmia selkeästi ja rakenteellisesti. Koska ohjelmat on tarkoitettu ymmärrettäviksi, muuteltaviksi ja ylläpidettäviksi koko niiden elämänkaaren, hyvä ohjelmointikieli auttaa muita lukemaan koodia ja ymmärtämään sen toimintaa.

Mietittäessä ohjelmointikieltä ohjelmistoa varten, tulee myös miettiä sen suunnittelua, implementointia ja ylläpitoa, unohtamatta kuitenkaan testausta. Näin valitut kielet tukevat koodin kirjoittamista ymmärrettäväksi, muuteltavaksi ja ylläpidettäväksi.

Tässä työssä on käytetty ohjelmointikielinä pääasiassa JavaScript-, HTML5- ja CSS-kieliä, jotka ovat asiakaspään kolme pääkieltä.

2.4.1 JavaScript

JavaScript on ohjelmointikieli, jota eri selaimet tukevat. Sen avulla voidaan toteuttaa interaktiivisuutta HTML-elementteihin. Erilaisten web-teknologioiden kehittyessä, JavaScriptin rooli on kasvanut jatkuvasti ja sen kehitystarve jatkuu edelleen. ECMAScript ja DOM ovat omalta osaltaan merkittävästi parantaneet JavaScriptin yhdenmukaisuutta eri selainten välillä. /3/

Itsessään ECMAScript ei ole miltään osa-alueeltaan sidottu internetiin, vaan se on kuvaus scriptikielestä, sillä ECMA on standardoinut sen. Sen ei myöskään ole tarkoitus auttaa ohjelmoijia koodin rakentamisessa, vaan tällöin ohjelmoijien on syytä käyttää JavaScript-dokumentteja. ECMAScript on dokumentoitu ECMA-262-dokumentissa. /13/

BOM (Browser Object Model) on osa selainta, mutta sillä ei ole taustalla yhtä selkeää standardointiorganisaatiota. Ilman DOM- ja BOM- rajapintaa, ei ole JavaScriptiä /12/. BOM rakentuu suuresta kokoelmasta erilaisia objekteja ja antaa JavaScript-ohjelmalle mahdollisuuden käyttää selaimen ikkunan alla sen jokaista osaa. /13/

ECMAScriptin valmistumisen kanssa samoihin aikoihin, myös W3C sai valmiiksi dokumenttioliomallinsa (DOM). Siinä HTML-elementit ja niiden erilaiset attribuutit on määritelty puurakenteeseen. Se mahdollistaa HTML-dokumenttien muokkaamisen DOM-rajapinnan kautta. /11/

2.4.2 HTML

HTML eli Hypertext Markup Language on yhdessä CSS:n kanssa kaksi ydinteknologiaa internetsivujen rakentamisessa. HTML määrittää elementtien avulla koko verkkoselaimella esitetyn sivun rakenteen. Näitä molempia ylläpitää W3C organisaatio. HTML tarjoaa sivuille struktuurin ja se myös mahdollistaa, esimerkiksi erilaisten dokumenttien julkaisun, johon pystytään määrittämään kuvia, listoja ja muita elementtejä. Sen avulla pystytään liittämään internetsivuun myös videoita ja kuvia. /14/

HTML5 on uuden sukupolven versio, jossa on uudelleen määritelty mm. HTML 4.0.1, XHTML 1.0 ja XHTML 1.1. Se tarjoaa käyttäjälleen mahdollisuuden tuottaa ominaisuuksia, joita nykypäivän kehitystarpeet määrittävät, kuten mobiilituki ja responsiivisuus. Uuden version (HTML5) erikoisuutena on se, että se on suunniteltu käyttöjärjestelmäriippumattomaksi. Se on suunniteltu tukemaan kaikkia olemassa olevia web-selaimia. /15/

2.4.3 CSS

Verkkosivustojen ja internetin yleistyessä verkkosivujen määrä kasvoi räjähdysmäisesti, mutta niiden ulkoasut olivat yksinkertaisia. W3C (Word Wide Web Consortium) alkoi pikaisesti etsiä yksinkertaisille sivuille ratkaisua vuonna 1995. Noin vuoden päästä (1996), W3C julkaisi ensimmäisen kerran CSS-suositukset. Se mahdollisti rikkaamman esitystavan tavallisille HTML-sivuille. CSS mahdollisti HTML-elementtien yksittäisen, sekä moninaisen muokkaamisen. /16/

Cascada Style Sheets on myös W3C: n standardoima dokumenttipohjainen kieli, jota käytetään HTML:n kanssa. Ilman HTML-dokumenttia, CSS on turha, sillä sen tarkoituksena on määrittää ja muokata ulkoasuja. /16/

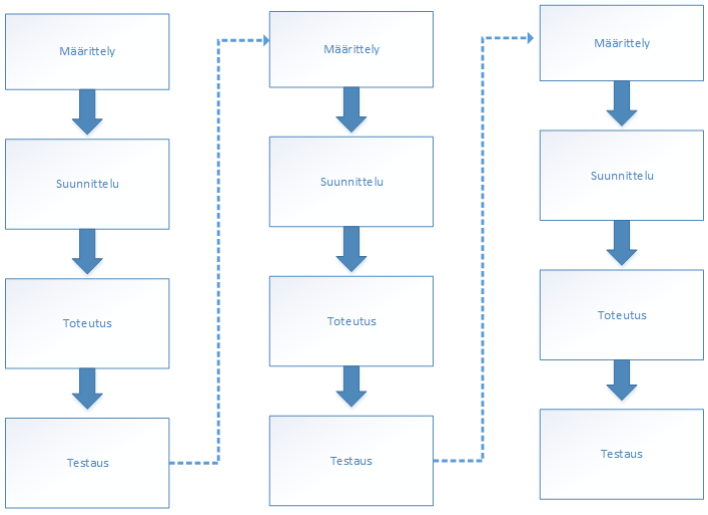
CSS:n toinen versio, CSS2 julkaistiin vuonna 1998 ja se toi paljon ominaisuuksia, joita oli odoteltu. Myöhemmin kävi kuitenkin ilmi, että sen osat oli osittain vaikea implementoida ja osa niistä poistettiin. Nykyisin CSS3 on jaettu erilaisiin moduuleihin, joiden selaintuki vaihtelee. /16/

3 OHJELMISTON MÄÄRITTELY

Järjestelmä toteutetaan Devatus Oy:n projektinjohdon asettamien tavoitteiden ja määritysten mukaisesti. Tarkoituksena on integroida web-kauppa jo olemassa olevaan web-sovellukseen. Tuotetta tulee pystyä ostamaan web-sovelluksen sivuilta ilman, että käyttäjän siirtyy itsenäisesti verkkopankkiin. Kaupan tulee olla selkeä ja helposti ymmärrettävä, koska asiakkaiden ikä ja tietotekniset taidot vaihtelevat suuresti.

3.1 Tuotekehitys

Monien tuotekehityshankkeiden läpivienti tapahtuu EVO-mallin mukaisesti. EVO tulee Evolutionary Delivery sanoista ja tarkoittaa vesiputousmallia (**Kuva 5.**). EVO-mallissa julkaistaan esimerkiksi kerran vuodessa uusilla ominaisuuksilla täydennetty versio. Tuotekehitys lähtee liikkeelle määrittelystä, josta jatketaan määrittelyiden pohjalta arkkitehtuuriin ja suunnitteluun. Toteutusosassa pyritään toteuttamaan määritysten ja arkkitehtuurin suunnittelemat toiminnot. Testaus puolestaan tarkoittaa mahdollisten virheiden karsimista, jotta tuote olisi mahdollisimman laadukas, eikä asiakas joudu tekemisiin virheiden kanssa. /17/



Kuva 5. EVO-malli /17/

3.2 Verkkokaupan määrittely

Verkkokaupan määrittely -osassa sovelluksen määrittelyllä tarkoitetaan web-kaupan määrittelyä. Kauppa on osa sovellusta, mutta tässä työssä se käsitellään erillisenä osana projektia, jolla on omat määritelmät ja vaatimukset.

Vaatimusmäärittely (Requirement Specification) nimetään ja osittain ymmärretään eri tavalla. Haikala & Märijärvi (2002) määrittelevät synonyymeiksi vaatimusmäärittely käsitteelle analyysi, määrittely ja vaatimusmäärittely. Pääasiallinen tavoite vaatimusmäärittelyllä on selvittää sovelluksen toteutuskelpoisuus, tavoitteiden ja vaatimusten määritteleminen ja ratkaisumallin laatiminen.

Tärkein määrittelyprosessin tuloksena syntyvä dokumentti on kuitenkin toiminnallinen määrittely (functional specification). Tähän dokumenttiin yleensä kuvataan vaatimukset ja järjestelmän koko kuvaus. Vaatimuksia kartoitettaessa ohjelmistolle annetaan yleensä tavoitteita. Tämän työn tavoitteet on esitetty taulukossa 1. /17/

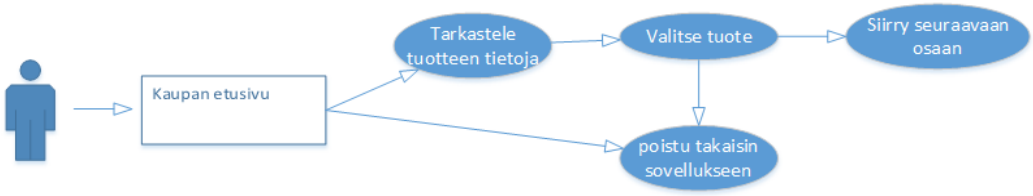
Taulukko 1. Kaupan vaatimusmäärittely

Vaaditut ominaisuudet	**Tärkeys**
Tietoturvallinen yhteys maksunsuorituksen yhteydessä	1
Tuotteiden selkeä esittely ja hinnoittelu	1
Selkeä ja helposti ymmärrettävä ostoprosessi	1
Tarpeellisten henkilötietojen täydentäminen ja validointi	1
Tietokanta- ja validointiprosessin turvaaminen	1
Tuotesisällön kuvaus	2
Automaattinen käyttöoikeushallinta	2
Tyylikäs ulkoasu	2

Taulukossa 1 esitetyt vaatimukset on jaettu kolmeen eri kategoriaan. Näillä kategorioilla on selkeä merkitys ja tärkeysjärjestys tuotteen kannalta. Vaatimusmäärittelyssä 1 tarkoittaa ”tulee olla” (Must-have), 2 – ”hyvä jos olisi” (Good-to-have) ja 3 - ”kiva jos olisi” (nice-to-have).

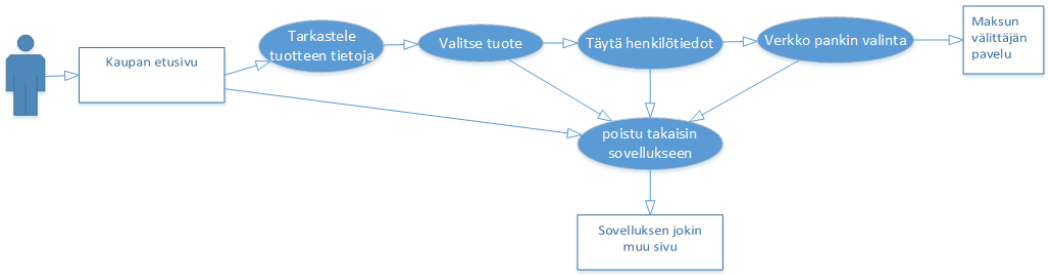
3.3 Käyttöprosessi

Kaupan ensimmäisellä sivulla käyttäjän tulee voida tarkastella tuotetta, valita tuote tai poistua kaupasta sovelluksen toiselle sivulle.



Kuva 6. Käyttöprosessi

Kun käyttäjä etenee normaalin maksuprosessin mukaisesti, hän ensin tarkastelee tuotteen tietoja ja valitsee miellyttävimmän tuotteen omien tarpeidensa mukaan. Käyttäjä voi siirtyä toiselle sivulle, eli poistua kaupasta, missä vaiheessa tahansa maksuprosessissa. Henkilötietojen keräyksen osalta prosessi tehdään yksinkertaiseksi siten, että siinä mahdollisuuksina on palata edelliseen osaan tai täyttää henkilötiedot ja siirtyä seuraavaan osaan, tai palata edelliseen näkymään, joka tässä tapauksessa on kaupan etusivu.

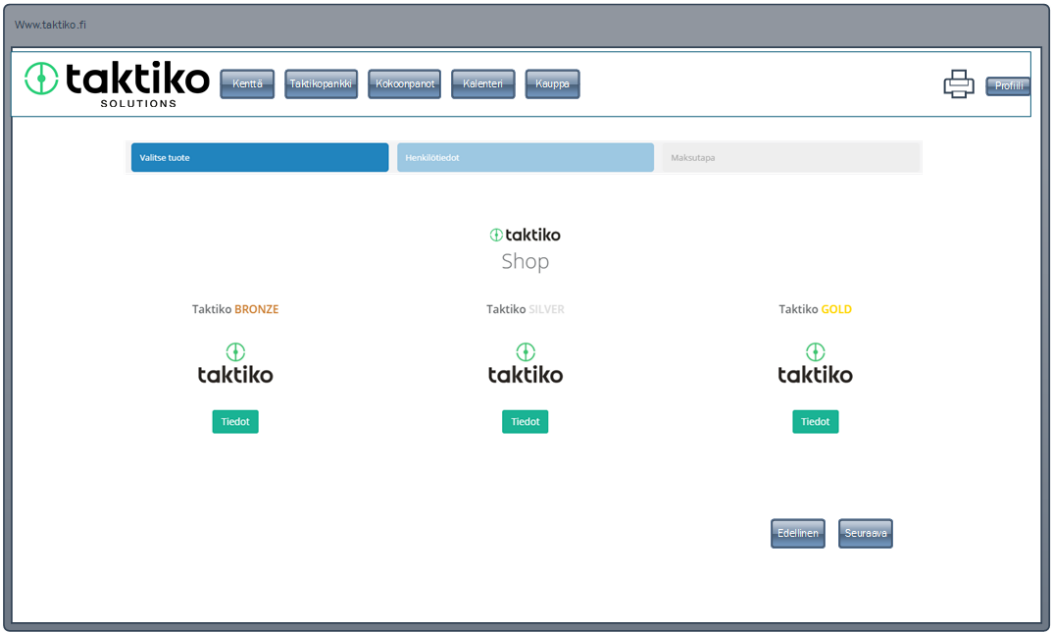


Kuva 7. Käyttäjän ostoprosessi

Kuvassa 9 esitellyssä prosessissa käydään läpi käyttäjän vaihtoehdot tulevassa kaupassa. Kuva noudattaa aiemmin 3-vaiheista mallia, joka myös tulee olemaan käytössä kaupan ostoprosessissa. Tähän kuuluvat tuotteen valinta, henkilötietojen täyttäminen ja verkkopankin valinta. Tämän jälkeen maksuun liittyvät tiedot annetaan maksun välittäjälle, joka ohjaa käyttäjää. Kun käyttäjä keskeyttää tai vie loppuun saakka maksuprosessin, maksunvälityspalvelu palauttaa käyttäjän takaisin Taktiko-sovelluksen etusivulle.

3.4 Kaupan ulkoasu

Ulkoasun suunnittelussa huomioitiin käyttäjäystävällisyys kiinnittämällä huomioita sen selkeyteen ja käytettävyyteen. Käyttöliittymä (UI) pyritään luomaan mahdollisimman helpoksi käyttäjälleen ja tämän vuoksi myös mahdollisimman hyvin automatisoiduksi. Kaupan osalta automatisointi tarkoittaa automaattisesti siirtymistä pankkipalveluista takaisin tuotteen sivuille ja toisinpäin.



Kuva 8. Kaupan etusivu

Kaupassa tulee näkyä myös Taktiko Solutions -logo yläpalkissa. Meteorin avulla on määritelty sivuston perusnäkymä, joka tarkoittaa yläpalkin ja alapalkin käyttöä jokaisessa näkymässä.

Koska tuote haluttiin näyttävän mahdollisimman yksinkertaiselta ja helppokäyttöiseltä, suunniteltiin kaupan käyttävän jQueryn Steps -käyttöliittymäkomponenttia, jotta maksun rekisteröintiprosessi olisi johdonmukainen. Askelten ensimmäinen osa toimii samalla myös kaupan etusivuna, jossa esitellään tuotteet, tuotteiden tarkemmat tiedot sekä niiden hinnat.

Toinen sivu on henkilötietojen keräämistä varten tarkoitettu osa, se kerää tarvittavat henkilötiedot, joita käytetään maksun eteenpäin saattamisessa. Näitä varten on varattu tekstikenttä jokaiselle tiedolle ja käyttäjä kirjoittaa näihin kenttiin tarvittavat henkilötiedot. Henkilötiedon pakollisuus merkataan tähdellä (*).

Kuva 9. Henkilötietojen keräys

Henkilötiedot tulee tarkastaa ennen siirtymistä seuraavaan osaan, jotta vääriä tietoja tai väärässä formaatissa esiintyviä tietoja ei pääse turhaan kuormittamaan maksunvälittäjän palvelimia. Jokainen tekstikehys tarkastetaan, sillä maksun välityksessä käytetään näitä tietoja ja niiden on oltava täysin oikein kirjoitettu. Tämän vuoksi kirjoitusasuista poistetaan kaikki turhat välilyönnit, erikoismerkit ja tarkastetaan, että sähköposti on oikean tyyppisesti kirjoitettu.

Kun henkilötiedot on oikein kirjoitettu ja ne ovat oikeassa muodossa, voidaan maksun tiedot lähettää maksunvälittäjälle POST-metodilla, joka puolestaan tulee palauttamaan pankkien logot. Näitä painamalla siirrytään nettimaksamiseen.

Kuva 10. Siirtyminen nettipankkiin

3.5 Maksunvälitys

Maksun välittäjänä toimii checkout.fi -palvelu. Sen toiminta perustuu http POST- ja GET-pyyntöihin. Välityspalvelun rajapintana toimii sen oma palvelu, johon POST-metodilla lähetetään ostajan yksilölliset tiedot, joista luodaan MD5-tarkiste. Tarkiste pitää sisällään 24 erilaista parametria. Taulukossa 2 esitellään parametrit, jotka muunnetaan MD5 muotoon.

Taulukko 2. MD5-tarkisteen parametrit /19/

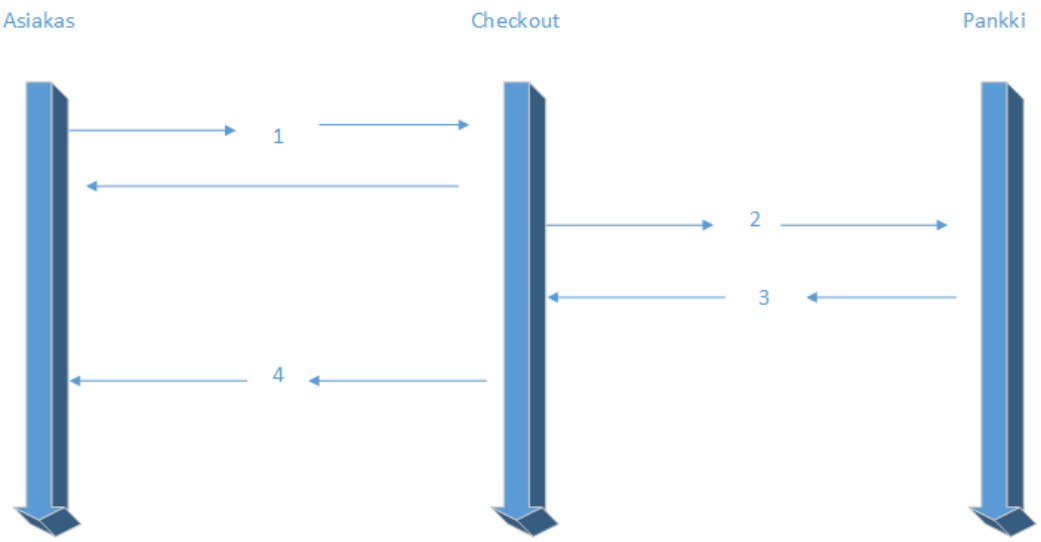
Kuvaus	**Muuttuja**	**Arvo**	**Tarvittava tieto**
Maksu versio	VERSION	0001	KYLLÄ
Yksilöivä tunnus maksulle	STAMP	13468	KYLLÄ
Maksun määrä sentteinä	AMOUNT	2000	KYLLÄ
Standardoitu viite	REFERENCE	43321792	KYLLÄ
Viesti	MESSAGE	”hei sinne”	EI
Maksun kieli (FI, EN,SE)	LANGUAGE	FI	EI
Kauppiaan tunnus	MERCHANT	375917	KYLLÄ
Palautus takaisin sivulle	RETURN	”http://osoite.fi”	KYLLÄ
Peruutuksen jälkeinen palautus	CANCEL	”http://osoite.fi”	KYLLÄ
Keskytyksen jälkeinen palautus	REJECT	”osoite”	EI
Maksun viivästys	DELAYED	”osoite”	EI
Maa (ISO-3166-1 Apha-3)	COUNTRY	”FIN”	EI
Valuutta, EUR	CURRENCY	EUR	KYLLÄ
Laitteen tyyppi (1=html)	DEVICE	0	KYLLÄ
Sisällön tyyppi	CONTENT	1	KYLLÄ

Maksun tyyppi	TYPE	0	KYLLÄ
Tarkastus summan tarkastus	ALGORITHM	3	KYLLÄ
Toimitus päivä	DELIVERY_DATE	20171231	KYLLÄ
Etunimi (vaaditaan lainapalveuissa)	FIRSTNAME	”Matti”	EI
Sukunimi (vaaditaan lainapalveluissa)	FAMILYNAME	”Meikäläinen”	EI
Osoite (vaaditaan lainapalveluissa)	ADDRESS	”Jokusentie 14”	EI
Postitomisto(vaaditaan lainapalveluissa)	POSTOFFICE	”Vaasa”	EI
Tarkastus summa, joka on summa kaikista kentistä	MAC	33	KYLLÄ
Sähköposti	EMAIL	joku.joku@joku.fi	EI
Puhelinnumero	PHONE	040123456	EI

Taulukossa 2 esitetyt parametrit lasketaan MD5 muotoon kaavalla

$$MD5(VERSION + STAMP + AMOUNT + REFERENCE + MESSAGE + LANGUAG + MERCHANT + RETURN + CANCEL + REJECT + DELAYED + COUNTRY + CURRENCY + DEVICE + CONTENT + TYPE + ALGORITHM + DELIVERY\_DATE + FIRSTNAME + FAMILYNAME + ADDRESS + POSTCODE + POSTOFFICE + SECRET\_KEY)$$

Kyseinen välityspalvelu vaatii maan, valuutan ja maksutavan lisäksi myös ostajan etunimen, sukunimen ja paljon muita henkilökohtaisia tietoja, jotka on esitelty taulukossa 2 ”tarvittava tieto” -sarakkeessa, jossa tekstinä on ”kyllä”. Tarvittavia tietoja on yhteensä 14. Maksunvälityspalveluun on mahdollista asettaa myös muita tietoja, mutta ne ovat valinnaisia.



Kuva 11. Maksun eteneminen sekvenssikaaviona

Kuvassa 11 esitellään tuotteessa käytetyn maksun eteneminen sekvenssikaaviona. Kuvassa asiakas on tuotteen ostaja, joka käyttää verkkokauppaa, Checkout maksun välittäjä ja sen tarjoama rajapinta. Pankki kuvaa verkkopankkia, jossa ostaja maksaa tuotteen.

Kuvan kohdassa 1 lähetetään POST-metodilla Checkout-palveluun asiakkaan tiedot ja maksun määrä parametreina. Rajapinta tarkastaa niiden oikean muodon. Mikäli parametrit eivät kelpaa, palauttaa Checkout-palvelu sivulle virheilmoituksen. Mikäli maksun tiedot ovat oikein, palauttaa maksunvälitys palvelupankkien linkit asiakkaan valittavaksi. Näistä vaihtoehdoista klikkaamalla, asiakas ohjataan automaattisesti kirjautumaan nettipankkitunnuksilla omalle pankkitililleen. tätä kuvataan kohdassa 2. Mikäli viestin kulku on estynyt tai metodia ei voida onnistuneesti toteuttaa, palauttaa rajapinta virheilmoituksen asiakkaan selaimeen.

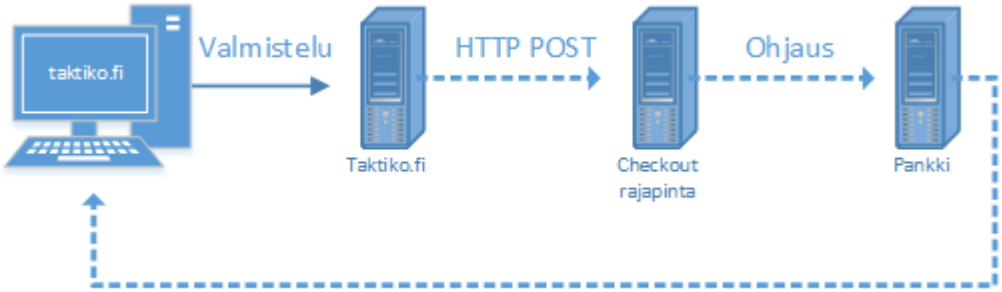
Pankkipalvelussa voidaan keskeyttää maksun suoritus tai suorittaa maksu. Maksun suorituksen tai keskeytyksen jälkeen palautetaan asiakas takaisin tuotteen etusivulle. Tuotteen sivulle palattaessa, ovat mukana myös parametrit, joiden mukaan tuote pystyy tunnistamaan, onko maksu suoritettu onnistuneesti.

Onnistuneen maksusuorituksen jälkeen parametrit luetaan ja niiden pohjalta lisätään käyttäjälle käyttöoikeutta tuotteen hinnassa annetun määrän mukaisesti.

Maksun kirjaaminen tuotteen järjestelmään toteutetaan täysin turvallisesti siten, että tuotetta ei pystytä käyttämään luvattomasti ilman maksua.

4 KAUPAN TOTEUTUS

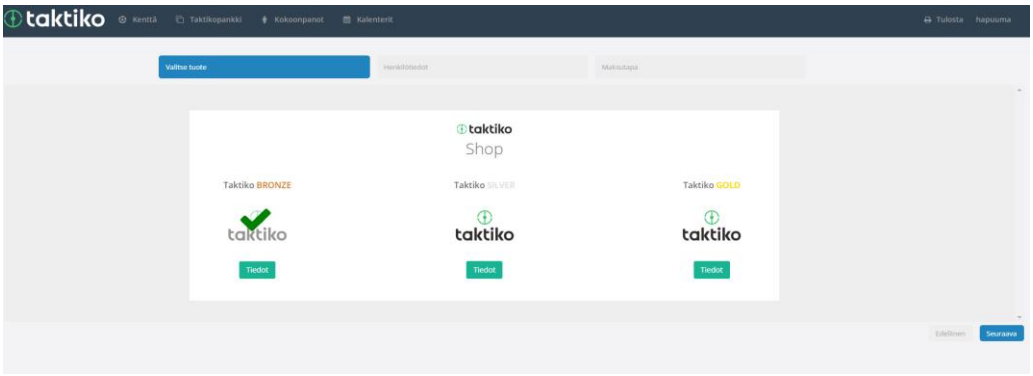
Toteutettu kauppa käyttää hyödykseen asiakkaan selainta, omaa palvelinta ja ulkopuolisia maksujärjestelmärajapintoja ja -palvelimia. Asiakas ohjataan kerran pois tuotteen sivuilta, mutta palautetaan sinne takaisin maksun keskeytyksen tai suorituksen jälkeen.



Kuva 12. Kaupan toiminta

4.1 Käyttöliittymä

Käyttöliittymä toteutettiin suunnitelmien mukaisesti ja tässä hyödynnettiin erityisesti Meteorin tuomia etuja, jolla voitiin hyödyntää muualla käytettyä ulkoasua kaupan pohjana. Kaupan askeleet puolestaan toteutettiin JQueryn Stepsillä.

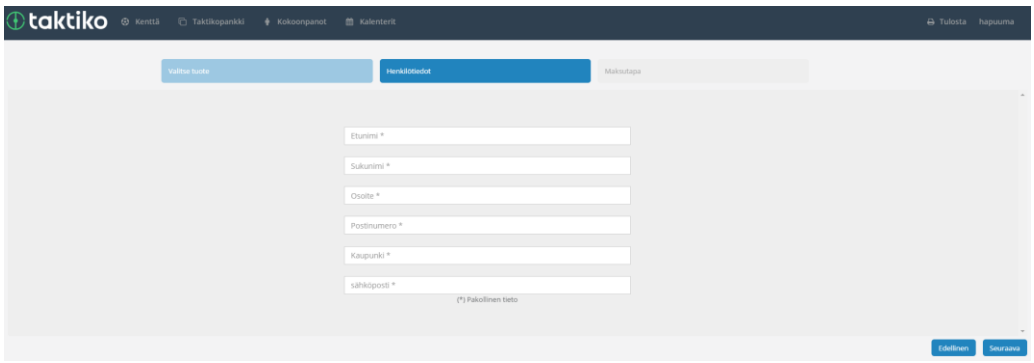


Kuva 13. Kaupan etusivu

Kuvassa 13 esitellään kaupan ulkoasua. Kaupan ensimmäinen askel toimii etusivuna, jossa käyttäjän on tarkoitus valita ostettava tuote. Tuotevaihtoehtoja kaupan toteuttamishetkellä oli kolme: Taktiko Bronze, Taktiko Silver ja Taktiko Gold. Tuotteiden tarkemmat tiedot saadaan näkyviin painamalla ”tiedot” nappulaa, joka

avaa näkymään tuotteen tarkemmat tiedot. Kaupan käyttöliittymä on yhteensopiva myös tableteille, koska tuotetta on pystyttävä käyttämään esimerkiksi harjoituksissa. Työkalupalkki ylhäällä mahdollistaa nopean siirtymisen takasin halutulle sivulle tuotteen sisällä.

Toinen askel, eli henkilötietojen keräys, on tuotteen ostamisen osalta kriittisin, sillä näiden tietojen pohjalta luodaan POST-metodi, jolla ollaan yhteydessä Checkout rajapintaan. Henkilötiedot-askeleessa kerätään henkilön tärkeimmät tiedot tekstikenttien avulla.



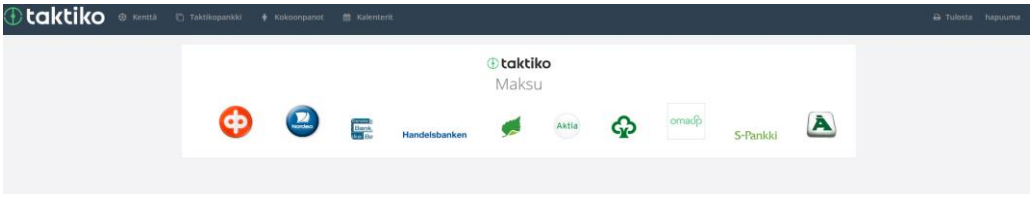
Kuva 14. Henkilötiedot -askel

Henkilötiedot-askel on hyvin yksinkertainen, sillä tiedot joudutaan tarkastamaan omalla palvelimella ennen pyynnön valmistelua eteenpäin. Myös asiakkaan mukavuuden kannalta on tärkeää, että henkilötietojen keräämisessä ei pyydetä arkaluontoisia tai luottamuksellisia tietoja. Lisäksi liiallinen määrä täytettäviä kenttiä saattaa heikentää tuotteen menekkiä, koska se saattaisi vaatia ponnisteluita henkilöiltä, jotka eivät ole tottuneet web-kauppojen toimintoihin. Askeleen yläosassa, suoraan navigointi palkin alla, on askelia kuvaavat elementit, joiden avulla käyttäjä näkee ostoprosessin vaiheen. Näitä elementtejä painamalla pystyy myös siirtymään prosessissa taaksepäin, mutta ei eteenpäin.

Kuva 15. Tarvittavat henkilötiedot

Henkilötiedot kerätään käyttämällä yksinkertaisia tekstikenttiä. Vaikka kaikkien henkilötietojen täyttäminen on ehtona kaupan syntymiselle, on se erikseen esitetty käyttämällä tähteä (*) esittämään tarvittava tieto. Tekstikentissä näkyvä teksti on ”placeholder”, joka poistuu näkyvistä käyttäjän klikattua kyseessä olevaa tekstikenttää. Henkilötietojen validointi- ja varmistusprosessi esitellään myöhemmin kohdassa ”Validointi-, valmistelu- ja ohjausprosessi”.

Kun henkilötiedot on onnistuneesti varmistettu ja validoitu, siirtyy käyttäjä maksun valintasivulle, joka toimii kaupan viimeisenä sivuna. Tämän jälkeen asiakas ohjataan valitsemansa pankin sivulle jatkamaan maksusuoritukseen.



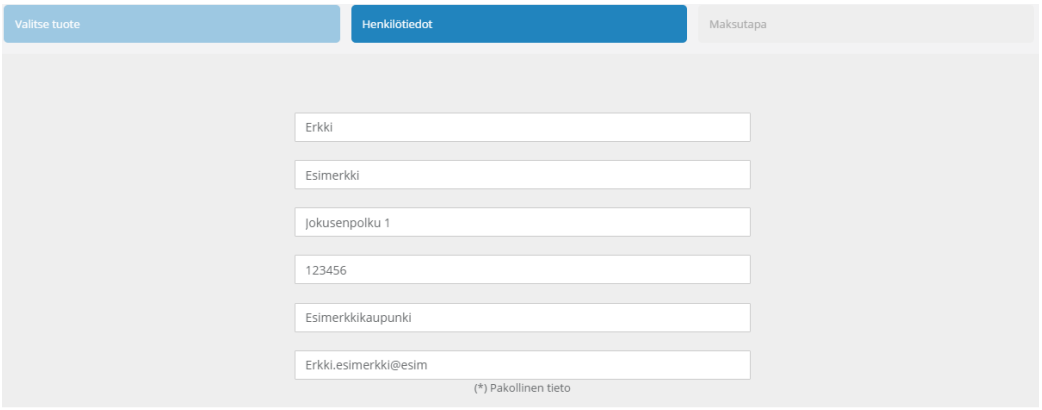
Kuva 16. Kaupan viimeinen askel

Kaupan maksuosasta on mahdollisuus edelleen keskeyttää maksun suoritus ilman, että siirrytään maksun suoritukseen. Jokainen pankin logo sisältää parametrit ja linkit, joita käyttäjä tarvitsee siirtyessään pankin maksupalveluun, näin käyttäjän ei itse tarvitse kirjoittaa tai tallentaa tietoja omasta toiminnastaan suorittaakseen ostoprosessi loppuun saakka.

4.2 Validointi-, valmistelu- ja ohjausprosessi

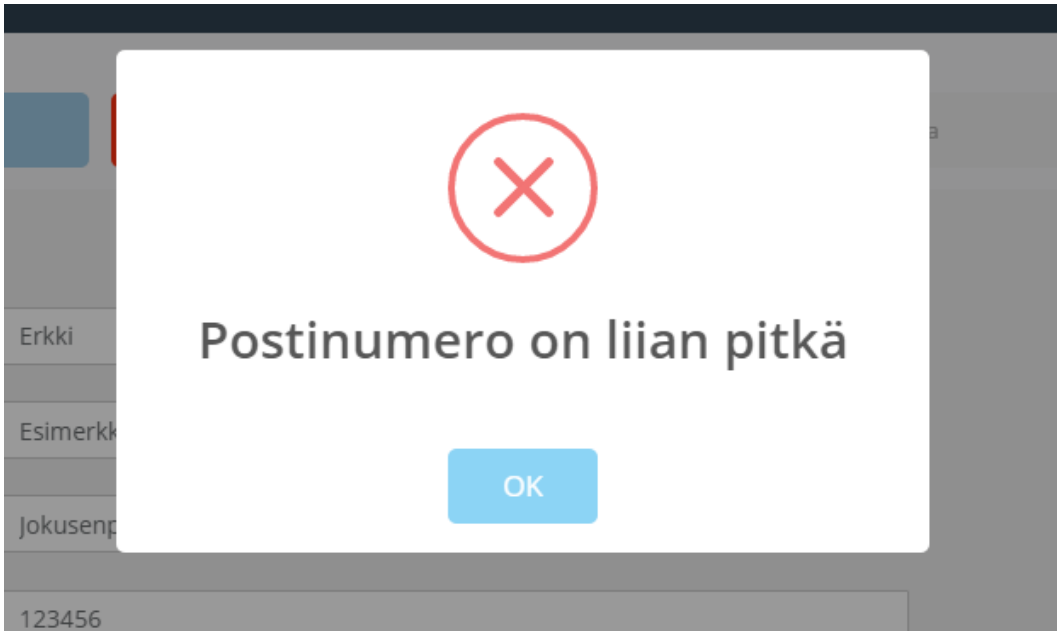
Kun henkilötiedot on saatu kerättyä, valmistelee palvelin valmiiksi Checkout rajapinnalle oleelliset tiedot. Valmisteluprosessi alkaa oleellisten tietojen yhteenvedolla, jossa tiedot validoidaan. Nämä tiedot kerätään ostoprosessin edetessä ja sisältää henkilötiedot, hinnan ja muita tärkeitä tietoja. Osa tiedoista on ennalta määritettyjä. Näitä ovat esimerkiksi tuotteen myyjän tiedot, joiden pohjalta Checkout-palvelu kirjaa maksun myyjän tilille.

Henkilötietojen validointiprosessi toteutetaan kerralla henkilötiedot -askeleessa ja asiakas ei pääse etenemään ostoprosessissa, ellei henkilötiedot ole oikein kirjoitettu. Validointiprosessiin kuuluu myös postinumeron, sähköpostin ja osoitteen oikeinkirjoituksen lisäksi tarvittavien välilyöntien ja erikoismerkkien karsiminen tai virheen ilmoittaminen näiden esiintyessä. Henkilötietojen validointiprosessi etenee systemaattisesti ylhäältä alas. Kuvassa 17 on kirjoitettu väärin postinumero sekä sähköpostiosoite. Postinumero on liian pitkä ja sähköpostiosoitteesta puuttuu osia, jolloin virheilmoitus tulee näkyä ruudulla.



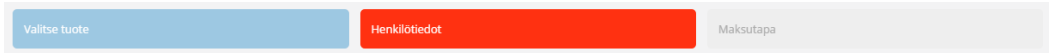
Kuva 17. Virheelliset henkilötiedot

Henkilötietojen virheilmoituksen voi kuitata painamalla ”ok” nappulaa.



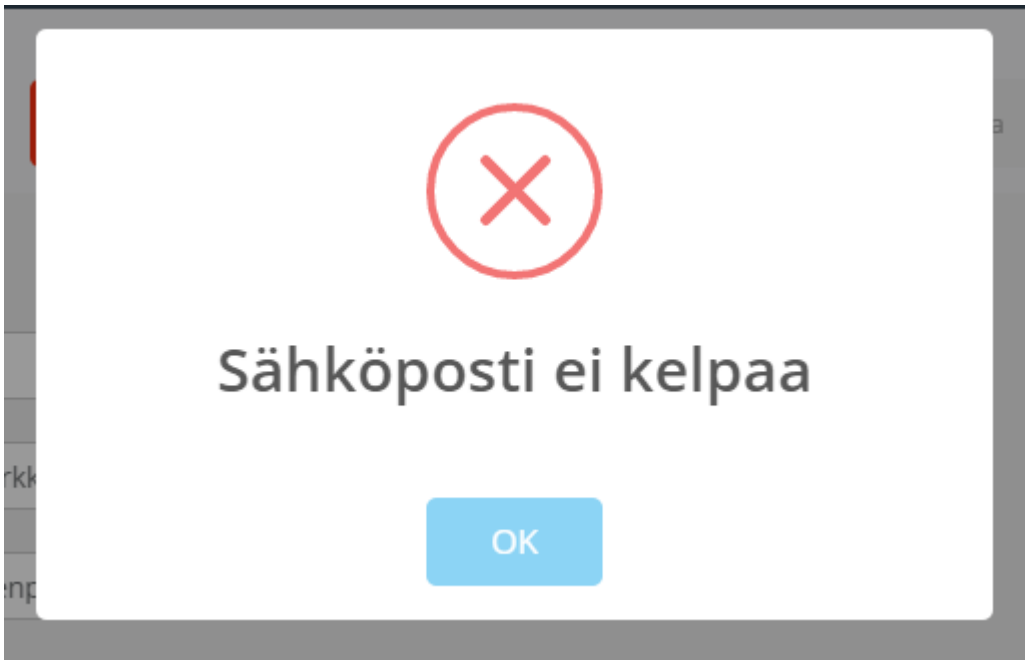
Kuva 18. Virheilmoitus

Virheilmoituksen jälkeen käyttäjä ei poistu kaupasta, vaan jää nykyiseen askeleensa. Askel ilmoittaa myös värien avulla missä askeleessa virhe ilmenee.



Kuva 19. Askeleen virheen indikointi

Näiden ilmoitusten ja indikaatioiden jälkeen käyttäjälle ei jää epäselväksi, missä virhe sijaitsee ja käyttäjän on helppo korjata virheellinen henkilötieto. Esitellyssä esimerkissä kuitenkin oli myös toinen virheellinen tieto: sähköposti ei kelvannut, koska sen muoto ei täsmää ehtoihin. Lopputuloksena seuraa uusi virheilmoitus.



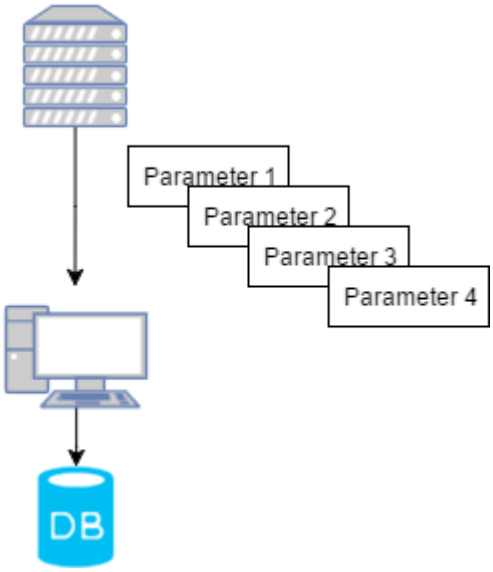
Kuva 20. Virheellinen sähköpostiosoite

Kun virheellinen sähköpostiosoite on korjattu, voi käyttäjä siirtyä eteenpäin ostoprosessissa pankin valintaan.

Tiedot on tässä vaiheessa valmisteltu oikeaan muotoon ja validoitu, ettei mahdollisia kirjoitusvirheitä esiinny. Kauppa on valmis ohjaamaan käyttäjän tietoineen eteenpäin maksun suoritukseen.

4.3 Oston rekisteröinti

Ostoprosessin viimeistely alkaa, kun asiakas ohjataan takaisin tuotteen sivustolle. Verkkopankki, jonka asiakas on valinnut, palauttaa asiakkaan takaisin rajapinnan kautta parametrit mukanaan. Palvelin lukee parametrit automaattisesti ja aloittaa monimutkaisen varmistusprosessin. Varmistusprosessin tuloksena asiakas saa käyttöönsä lisenssin tai virheilmoituksen selaimeensa, riippuen onko varmistusprosessi mennyt onnistuneesti loppuun saakka, vai onko kyseessä mahdollisesti tietoja koskeva virhe. Varmistusprosessin monimutkaisuus johtuu käyttäjäoikeuksien hallinnasta ja tietoturvasyistä. Varmistusprosessin ansiosta asiakas ei pysty saamaan käyttöönsä maksamatonta, täydet oikeudet antavaa lisenssiä.



Kuva 21. Oston rekisteröinti

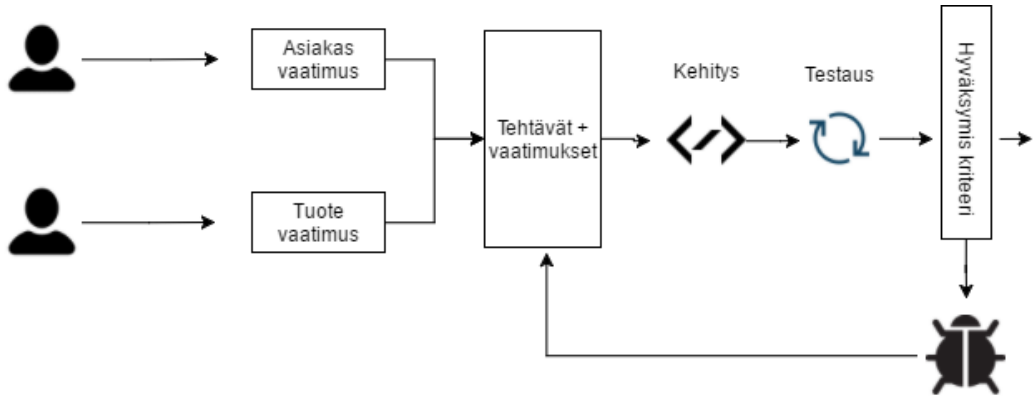
Kuvassa 21 kuvataan verkkopankin palautustoimintoa, jossa rajapinta palauttaa asiakkaan parametreineen takaisin tuotteen sivulle. Tuotteen sivustolla varmistusprosessin käsittely käyttää hyödykseen tietokantaa, sekä asiakkaan antamia tietoja ja rajapinnan palauttamia parametreja.

5 TESTAUS

Tuotteen testauksessa käytettiin ”Black-box” -testaustekniikoita ja kaikki testaukset toteutettiin manuaalisesti. Testausautomaatiota ei tuotteen testaukseen ollut käytössä, mutta sen käyttöön ottaminen myöhemmin on mahdollista. Tuotteen testaus keskittyi pääasiassa yrityksen sisällä toteutettavaan käytettävyystestaukseen ja käyttöliittymätestaukseen. Organisoimatonta testausta tapahtui testauskäyttäjien toimesta jatkuvasti. Testauksen avulla pyrittiin täyttämään hyväksymiskriteerit. Näiden pohjalta saatiin arvokasta palautetta tuotteen käytettävyydestä ja käyttöliittymästä.

5.1 Hyväksymiskriteerit

Tässä osassa esitellään verkkokaupalle asetettuja hyväksymiskriteereitä. Hyväksymiskriteerit ovat asiakkaan tai tuotteen omistajan toteuttamia kriteereitä, jotka pyritään täyttämään. Lopullinen kriteerien hyväksyntä tullaan määrittämään vasta testauksen jälkeen.



Kuva 22. Hyväksymiskriteerit osana sovelluskehitysprosessia

Kuvassa 22 kuvataan tarkemmin miten hyväksymiskriteerit rakentuvat. Jokaiselle tehtävälle on määritetty hyväksymiskriteerit, joita arvioidaan manuaalisen testauksen aikana. Tuotevaatimus tulee tuotteen tilaajalta, joka kertoo tuotteen ominaisuudet ja vaatimukset niihin. Asiakas puolestaan asettaa palautteen osalta omia vaatimuksia. Näiden osalta tehdään erilaisia tehtäviä, joita kehittäjät, tässä tapauksessa työn tekijät, toteuttivat projektin aikana. Testausta suoritetaan usein kehityk-

sen aikana, mutta varinaista hyväksymiskriteerien täyttymistä tarkkaileva testaus suoritetaan kehityksen jälkeen. Mikäli tämän jälkeen ilmenee vikoja, ne usein asetetaan takaisin tehtäväjonoon virheenä, koska hyväksymiskriteeri ei täyty.

5.2 Kaupan testitapaukset

Taulukko 3. Hyväksymiskriteerit

ID	Vaatimus	Testattava toiminto	Odotettu tulos
T67-1	Tuotteen valinta on helppoa ja yksinkertaista	Tietojen näkyminen, etusivun toiminnot, siirtyminen yhteystietolomakkeeseen	Etusivun toiminnot toimivat ja siirtyminen yhteystietolomakkeeseen onnistuu, kun tuote on valittu.
T67-2	Asiakkaan virheen korjaus ja yhteystietojen validointi	Validointiprosessi jokaisessa yhteystietokohdassa toimii	Yhteystietolomakkeesta ei pääse etenemään, mikäli havaitaan erikoismerkkejä, syntaksivirhe
T67-3	Kauppa tuo esiin verkkopankkien logot	Verkkokauppojen logot tulevat esiin ja niitä on mahdollisuus klikata	Verkkokauppojen logoja pystyy klikkaamaan ja ohjaus lähtee liikkeelle
T83-1	Kauppa käyttää hyödyksi maksurajapintaa	Kauppa ohjaa automaattisesti asiakkaan valitsemaansa verkkopankkiin	Asiakas ohjataan verkkopankkiin
T83-2	Verkkopankki ohjaa asiakkaan takaisin oikealle sivustolle	Maksun jälkeen asiakas ohjataan takaisin sivustolle	Asiakas ohjataan takaisin tuotteen sivustolle, mukanaan tarvittavat parametrit
T83-3	Tuotteen sivuilla palvelin käsittelee automaattisesti käyttäjän maksun	Tuote rekisteröi käyttäjälle 10 lisenssiä vuodeksi	Käyttäjä saa 10 x vuoden lisenssin
T83-4	Maksu on rekisteröity oikein tietokantaan	Lisenssi, käyttöoikeuksien määrä ja kesto, viite, yhteystiedot ja asiakasnumero	Tietokanta on päivittynyt oikein

5.3 Kaupan testaus

Kaupan testaus toteutettiin hyväksymistestauksella, jolloin tarkasteltiin asiakasvaatimuksia suhteessa toteutettuun toimintoon.

Taulukko 4. Testien tulokset

ID	Testattava toiminto	Odotettu tulos	Tulos
T67-1	Tietojen näkyminen, etusivun toiminnot, siirtyminen yhteystietolomakkeeseen	Etusivun toiminnot toimivat ja siirtyminen yhteystietolomakkeeseen onnistuu, kun tuote on valittu.	☑ etusivulla on nähtävissä tuotekohtaiset tiedot ja hinnat. Siirtyminen yhteystietolomakkeelle onnistuu
T67-2	Validointiprosessi jokaisessa yhteystietokohdassa toimii	Yhteystietolomakkeesta ei pääse etenemään, mikäli havaitaan erikoismerkkejä, syntaksivirhe	☑ Yhteystietolomakkeelta ei pääse seuraavaan, elleivät kaikki tiedot ole oikein
T67-3	Verkkokauppojen logot tulevat esiin ja niitä on mahdollisuus klikata	Verkkokauppojen logoja pystyy klikkaamaan ja ohjaus lähtee liikkeelle	☑ Logot ilmestyvät ja klikkauksen jälkeen automaattinen ohjaus verkkopankkiin alkaa
T83-1	Kauppa ohjaa automaattisesti asiakkaan valitsemaansa verkkopankkiin	Asiakas ohjataan verkkopankkiin	☑ Asiakas ohjataan automaattisesti asiakkaan verkkopankkiin
T83-2	Maksun jälkeen asiakas ohjataan takaisin sivustolle	Asiakas ohjataan takaisin tuotteen sivustolle, mukanaan tarvittavat parametrit	☑ Asiakas ohjataan takaisin tuotteen sivustolle, mukana varmennusparametrit
T83-3	Tuote rekisteröi käyttäjälle 10 lisenssiä vuodeksi	Käyttäjä saa 10 x vuoden lisenssin	☑ Käyttäjä saa 10 x vuoden lisenssin
T83-4	Lisenssi, käyttöoikeuksien määrä ja kesto, viite, yhteystiedot ja asiakasnumero	Tietokanta on päivittynyt oikein	☑ Tietokanta on päivittynyt oikein

6 YHTEENVETO

Opinnäytetyön tuloksena syntyi toimiva verkkokauppa, joka rakennettiin web-sovelluksen yhteyteen. Se toimii eniten käytetyillä verkkoselaimilla. Verkkokauppa käyttää maksunvälitysrajapintaa ja ohjaa asiakkaan automaattisesti asiakkaan valitsemaan verkkokauppaan. Verkkokauppa analysoi rajapinnan palauttamat parametrit ja suorittaa automaattisesti käyttäjätietojen hallintaa, tunnistaa virheet ja ilmoittaa asiakkaalle, mikäli maksu on keskeytynyt.

Työn suurimmat ongelmat olivat validoinnissa ja rajapinnan käytön yhteydessä. Usein testauksissa esille tulleissa ongelmissa kyseessä oli riittämätön varmistus ja henkilötietojen validointi. Tämä ilmentyi monenlaisina virheinä ja usein kyseessä oli välilyönnin tuottama syntaksivirhe, jonka rajapinta ilmoitti ennen verkkopankkiin siirtymistä. Tämä saatiin korjattua, mutta itse ongelman paikantaminen oli alkuun haasteellista, sillä rajapinnan antama syntaksi ja virheilmoitus eivät osanneet kertoa tarkempaa tietoa virheestä.

Responsiivinen ja reaktiivinen käyttöliittymä soveltuu hyvin käytettäväksi myös tablettiversiona. Koko sovellus on myös suunniteltu niin, että käyttöliittymää voidaan käyttää vaikkapa jalkapallokentän reunalla. Näin lisenssin uusiminen kentän reunalta ei tuo ongelmia asiakkaalle.

Verkkokauppa on toimiva, mutta sen saattaminen nykyaikaiseksi ja tyylikkääksi vaatii käyttöliittymän uudelleentoteuttamista. Tähän käyttöön on olemassa useita avoimen lähdekoodin tarjoamia kirjastoja ja malleja, joita voisi hyödyntää esimerkiksi tämän verkkokaupan käyttöliittymän parantamiseen. Kun taustalla on valmis ja toimiva rajapintaratkaisu, vaatimukset uudelleentoteuttamisen osalta ovat pienet.

LÄHTEET

/1/ Kasvuopen. 2017. Viitattu 1.2.2017. https://www.kasvuopen.fi/ yrityk-set/taktiko-solutions-oy.

/2/ Devatus Oy. 2017. Viitattu 1.2.2017. http://www.devatus.fi/

/3/ Tuomisto, H. 2016. Reaaliaikaisen Tilannekuvan jakamisen suorituskyky. Viitattu 8.2.2017. https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/24460/Tuomisto.pdf?sequence=1&isAllowed=y

/4/ Dajaeger, G. Comparing Angular, Aurelia and React: Is there a next-gen JS framework that rules them all?. Viitattu 4.2.2017 http://www.ae.be/blog-en/comparing-angular-aurelia-react-js-framework/.

/5/ Stubailo, S. Optimistic UI with Meteor, Viitattu 4.2.2017 https://blog.meteor.com/optimistic-ui-with-meteor-67b5a78c3fcf#.26wzbqnh2

/6/ Herron, D. Node Web Development. 2011. Birmingham. Packt Publishing Ltd.

/7/ Dayley, B. Node.js, MongoDB, and AngularJS Web Development. 2014. Addison-Wesley.

/8/ Cimpanu, C. 2017. MongoDB Databases Held for Ransom by Mysterious Attacker. Viitattu 4.2.2017. https://www.bleepingcomputer.com/ news/security/mongodb-databases-held-for-ransom-by-mysterious-attacker/

/9/ W3C. HTML & CSS. Viitattu 4.2.2017. http://www.w3schools.com/Tags/ref_httpmethods.asp

/10/ Rivest, R. 1992. The MD5 Message-Digest Algorithm. http://www.ietf.org/rfc/rfc1321.txt

/11/ N. Zakas, Web definitions: DOM, Ajax and more. Viitattu 4.2.2017. https://www.nczonline.net/blog/2009/09/29/web-definitions-dom-ajax-andmore/

/12/ Mozilla Developer Network. 2017. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction

/13/ Bhardwaj, S. 2013. Browser Object Model. Viitattu 4.2.2017. http://bay-six.blogspot.fi/2013/01/javascript-browser-object-model.html

/14/ W3C. HTML & CSS. Viitattu 8.2.2017. http://www.w3.org/standards/webdesign/htmlcss.html

/15/ Pilgrim, M. HTML5: Up and running. 2010. O'Reilly Media, inc.

/16/ Eric, A. M. 2010. CSS: The Definitive Guide: The Definitive Guide. O'Reilly Media, inc.

/17/ Haikala, I & Märijärvi, J. 2002. Ohjelmistotuotanto. Talentum Media Oy.

/18/ Association for Computing Machinery. 2012. Viitattu 3.5.2017.
http://amturing.acm.org/award_winners/rivest_1403005.cfm

/19/ Checkout. Tekninen materiaali. Julkinen rajapintakuvaus. Viitattu 3.5.2017.
https://checkoutfinland.github.io/#payment

/20/ MeteorHacks. 8.4.2014. Introduction to DDP. Viitattu 10.5.2017.
https://meteorhacks.com/introduction-to-ddp.html

/21/ Meteor Developers. Meteor Guide. Viitattu 24.5.2014.
https://guide.meteor.com/