
Tekninen määrittely

Ravitsemusterapeuttipalvelu

Larate

versio 1.0

VERSIONHISTORIA

Versio	Päiväys	Tekijät	Selite
1.0	10.2.2020	Jenni Kortelainen Laura Huikko	Alkuperäinen

SISÄLLYSLUETTELO

1.	JOHDANTO	4
1.1	TARKOITUS	4
1.2	DOKUMENTIN KATTAVUUS	4
1.3	MÄÄRITELMÄT, TERMIT JA LYHENTEET	4
1.4	VIITTEET	6
1.5	YLEISKATSAUS DOKUMENTTIIN	6
2.	YLEISKUVAUS.....	7
2.1	JÄRJESTELMÄ	7
2.2	LAITE- JA OHJELMISTOYMPÄRISTÖ	7
2.3	KÄYTTÖLIITTYMÄ	7
2.4	REUNAEDOT	7
3.	ARKKITEHTUURIN KUVAUS	8
3.1	RATKAISUPERIAATTEET	8
3.2	TIETOKANTA-ARKKITEHTUURI.....	9
3.3	MODUULIT JA PROSESSIT	9
3.4	UUELLEENKÄYTETTÄVÄT KOMPONENTIT	9
4.	MODUULI (TAI PROSESSI)	10
4.1	KÄYTTÖLIITTYMÄN LUOMINEN.....	10
4.2	TIETOKANNAN LUOMINEN	10
4.3	BACK-END	11
4.4	MODUULIEN YHDISTÄMINEN	12
4.5	TESTAUS	12
5.	MUITA MAHDOLLISIA KOHTIA:	13

1. JOHDANTO

1.1 Tarkoitus

Tämä dokumentti on Itä-Suomen yliopiston tietojenkäsittelytieteen kandidaatin vaiheen pakollisen Ohjelmistotuotanto II (OTII, ETA7800, 3621416) kurssin projektityötä varten tehty. Kurssi on Savonia ammattikorkeakoulun järjestämä ja kurssi on toteutettu yhteistyönä ICT-polun kautta. Projektissa toteutetaan ravitsemustieteen opiskelijoiden tilaama ravitsemusterapeuttien verkkopalvelu ja kyseinen dokumentti on projektin tekninen määrittely.

Dokumentti toteutetaan projektiryhmän jäsenille projektin toteuttamisen ja suunnittelun tueksi, asiakkaalle sekä projektin ohjaajien eli kurssin opettajien tarkasteltaviksi. Dokumentti määrittää vaatimusmäärittelyssä esille tuotujen toiminnallisuuden teknistä puolta ja täydentää näin toiminnallista määrittelyä.

1.2 Dokumentin kattavuus

Tässä dokumentissa määritellään ja kuvataan projektiin ja dokumenttiin liittyvät määritelmät, termit ja lyhenteet sekä luodaan yleiskatsaus dokumenttiin. Dokumentista käy esille projektityön sovellusalue, järjestelmän osuus sovellusalueesta, sovelluksen toimintaa tukevat laite- ja ohjelmistoympäristöt, toteutuksen keskeiset reunaehdot sekä se, miten järjestelmä liittyy ympäristöönsä. Dokumentissa tarkastellaan myös projektin arkkitehtuurikuvausta. Arkkitehtuurikuvauksesta käy ilmi tietokanta-arkkitehtuuri, ohjelmistoarkkitehtuuri, moduulit ja prosessit sekä projektin uudelleenkäytettävät komponentit.

1.3 Määritelmät, termit ja lyhenteet

1.3.1 Alabanneri

Alabanneri on verkkosivuston alaosassa oleva pysyvä osa, jossa voi olla esimerkiksi yhteystiedot. Asiakas ei ole määritellyt alabannerin sisältöä.

1.3.2 Azure DevOps

Azure DevOps on Microsoftin tuottama verkkosivu, joka tarjoaa ohjelmistotuottajille palveluita tukemaan ryhmätyöskentelyä, jakamaan ohjelmakoodia ja rakentamaan applikaatioita sekä palvelinpalveluita. Palvelinpalvelu oli ennen nimeltään Visual Studio Team Foundation Server (TFS).

1.3.3 Back-End

Ohjelmistotuotannossa verkkoprojekteissa ohjelmat koostuvat yleisimmin kahdesta komponentista, Back-Endistä ja Front-Endistä. Back-End tarkoittaa käytännössä ohjelmakoodin osaa, jossa käsitellään talouslogiikka (= ohjelmaosa, jossa koodataan oikean maailman säännökset siitä, kuinka dataa voidaan luoda, tallettaa ja muuttaa) ja tiedon tallennus.

1.3.4 Front-End

Front-End on käytännössä ohjelmakoodin osa, jolla luodaan abstraktio, eli ohjelman komponentit, joista koostuu käyttäjäystävällinen käyttöliittymä.

1.3.5 Git

Git on versionhallintaohjelmisto, joka tarkkailee lähdekoodiin tehtäviä muutoksia. Versionhallintaohjelmiston avulla lähdekoodi, joka sisältää sen kaiken historian, voidaan peilata kaikille ohjelmistokehittäjille.

1.3.6 MariaDB

MySQL relational database management system (RDBMS) eli relaatiotietokantojen hallintajärjestelmä.

1.3.7 Moduuli

Ohjelmiston osa.

1.3.8 PHP

Alun perin verkkosivujen ja -ohjelmistojen kehitykseen luotu ohjelmointikieli.

1.3.9 SQL

Structured Query Language eli strukturoitu tiedustelukieli, jota käytetään hallitsemaan relaatiotietokannoissa sisältyvää dataa.

1.3.10 VS Code

Visual Studio Code on koodieditori. Projektiryhmä on päättänyt käyttämään kyseistä työkalua projektissaan.

1.3.11 Yläbanneri

Yläbanneri on verkkosivuston yläosassa oleva pysyvä osa, jossa voi olla esimerkiksi kuva tai kuvaus verkkosivusta. Asiakas on määritellyt kuvailevan tekstisisällön yläbanneria varten.

1.3.12 #C1727B

Heksadesimaali värikoodi keskivaalealle pinkinpunaiselle sävyille.

1.3.13 #5A9089

Heksadesimaali värikoodi syaanin sävyille.

1.3.14 #89BBB6

Heksadesimaali värikoodi keskivaalealle syaanin sävyille.

1.4 Viitteet

Dokumentissa viitataan projektin vaatimusmäärittely- sekä toiminnallinen määrittely dokumentteihin. Näiden lisäksi projektissa tullaan luomaan muitakin dokumentteja, joista saatetaan viitata tähän tai tästä niihin. Dokumentit löytyvät Savonia ammattikorkeakoulun Moodle-sivuilta Group R:n palautuksista ja projektijäsenten tietokoneilta.

1.5 Yleiskatsaus dokumenttiin

Dokumentin kohdassa yksi tarkastellaan dokumentin yleisilmettä ja sisältöä sekä määritellään termejä, jotka projektin dokumentaatiota tutkiessa saattavat tulla vastaan.

Luvussa kaksi käsitellään järjestelmää ja sen toimintaa ja teknisyyttä yleisellä tasolla.

Luvussa kolme käsitellään järjestelmän arkkitehtuuria. Lukuun sisältyy ratkaisuperiaatteet, tietokanta-arkkitehtuuri, moduulit ja prosessit sekä uudelleenkäytettävät komponentit.

Luvussa neljä kuvaillaan järjestelmän moduuleja tarkemmin ja tarkastellaan niiden rakennetta ja käyttäytymistä.

Luvussa viisi tarkastellaan muita esille tulevia asioita, joita dokumentissa ei ole vielä käsitelty. Näitä ovat esimerkiksi virhetilanteissa toimiminen sekä käyttöliittymään ja testattavuuteen liittyvät huomion arvoiset kohdat.

2. YLEISKUVAUS

2.1 Järjestelmä

Tilaaja toivoo sovellusta, jossa laillistetut ravitsemusterapeutit voivat maksua vastaan luoda profiilin/”jäsenyyden”. Ravitsemusterapeutti saa tällöin omia tietojansa kantavan profiilin/”kortin” sovellukseen, jonka avulla hän saa näkyvyyttä ja asiakkaita. Profiiliin lisättävissä esim. yhteystiedot, ajanvaraus.

Asiakkaat voivat etsiä palvelusta toimipaikan ja suodatinhaun perusteella tarpeisiinsa sopivaa ravitsemusterapeuttia. Suodatinhaussa ovat valittavina ravitsemusterapeuttien erityisosaamiset, kuten esim. liikuntaravitseminen, tyypin 1 diabetes, jne. Asiakkaalle esitetään haun suodattimiin sopivat ravitsemusterapeutit.

Projektissa luodaan asiakkaan toivomasta sovelluksesta prototyyppi. Prototyyppiin ei tule kaikkia toiminnallisuuksia, mitä asiakas lopulliselta sovellukselta toivoo (esim. jäsenmaksut).

2.2 Laite- ja ohjelmistoympäristö

Prototyyppi toteutetaan projektiryhmän jäsenten tietokoneilla. Jäsenet ovat valinneet tämän toteutustavan, jotta he voivat ladata tarvittavat ohjelmat prototyypin luomiseen (Git, jne.). Prototyypin luomiseen käytetään erinäisiä ohjelmia, muun muassa Azure DevOpsia projektin etenemisen seurantaan, Visual Studio Codea ohjelmointiin ja MariaDB:tä tietokannan luomiseen

2.3 Käyttöliittymä

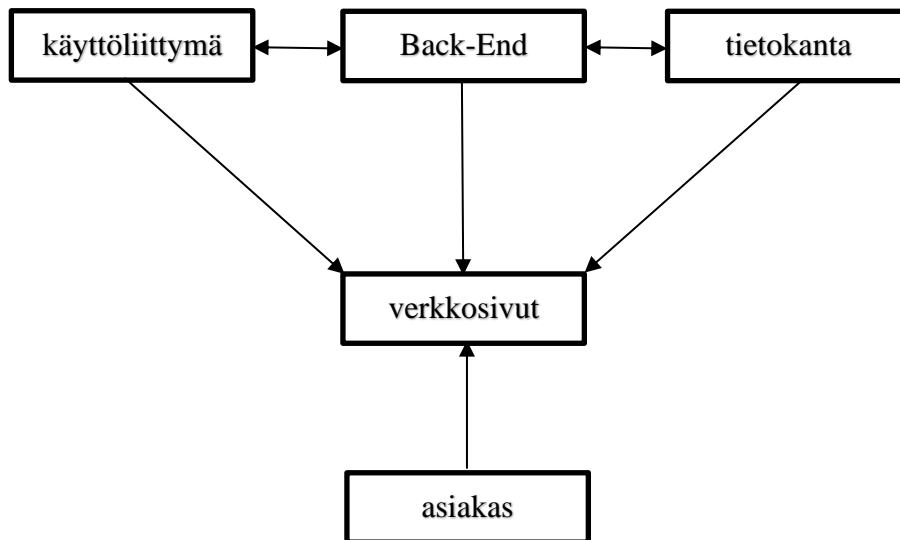
Sovelluksen etusivulle tulee siistit, käyttäjäystävälliset sivut, joista asiakas pystyy valitsemaan mitä haluaa tehdä (haku, tietoa, jne.). Sivustolle tulee myös yleistä tietoa sovelluksesta yläbanneriin ja yleistä tietoa ravitsemuksesta ja ravitsemusterapeuteista erilliselle sivulle.

2.4 Reunaehdot

Projektiryhmän tulee voida luoda prototyyppi käytössä olevien ohjelmien avulla 0€ budjetilla.

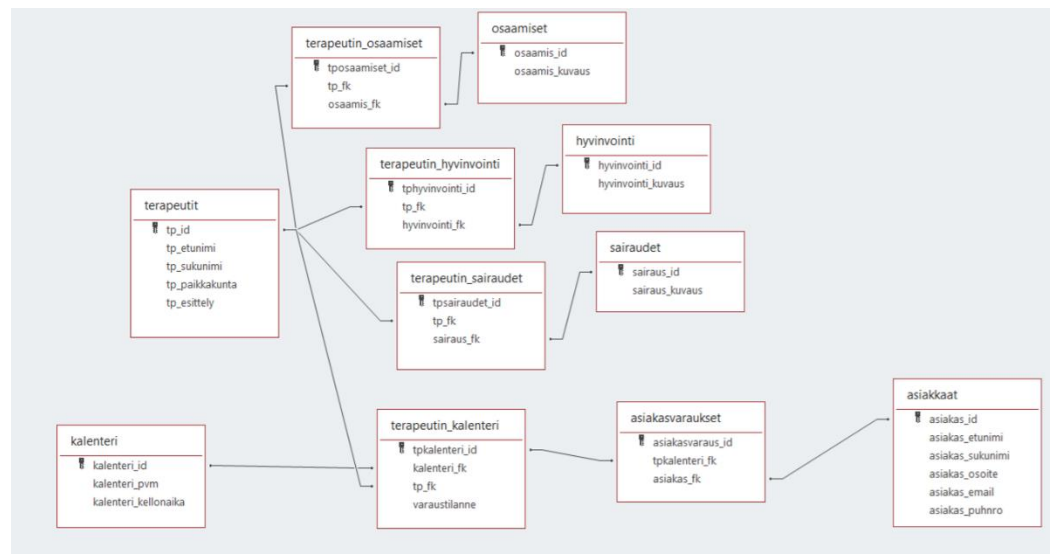
3. ARKKITEHTUURIN KUVAUS

3.1 Ratkaisuperiaatteet



Käyttöliittymä luodaan Front-Endillä ja se on yhteydessä Back-Endillä luotaviin toiminnallisuuksiin. Back-End on yhteydessä tietokantaan, josta tarvittavat tiedot saadaan. Front-End on yhteydessä tietokantaan ainoastaan Back-Endin kautta. Yhdessä Front- ja Back-End sekä tietokanta muodostavat verkkosivut, joista näkyvä osuus on Front-Endillä luotu käyttöliittymä. Asiakkaalla on pääsy verkkosivuille käyttöliittymän kautta.

3.2 Tietokanta-arkkitehtuuri



3.3 Moduulit ja prosessit

3.1 Käyttöliittymän luominen

3.2 Tietokannan luominen

3.3 Back-End

3.3.1 Ravitsemusterapeutin profiilin luonti

3.3.2 Ravitsemusterapeutin profiilin muokkaus

3.3.3 Ravitsemusterapeutin profiilin poisto

3.3.4 Haku suodatinten perusteella

3.3.5 Haku paikkakunnan perusteella

3.3.6 Ajanvaraus

3.4 Moduulien yhdistäminen

3.5 Testaus

3.4 Uudelleenkäytettävät komponentit

Projekti on kokonaisuudessaan täysin uudelleenkäytettävissä, mikäli prototyyppi halutaan viedä jatko tuotantoon.

4. MODUULI (TAI PROSESSI)

4.1 Käyttöliittymän luominen

Etusivu, palvelut, ravitsemusterapia ja ota yhteyttä ovat kaikki omia sivuja. Värimaailmaan viitteitä antavia värikoodeja #C1727B, #89BBB6, #5A9089. Yläbannerissa fonttikoko 14 otsikolle, tekstille 12, alabannereissa/sisällöissä: 12 (ei pakollisia, kunhan noudattaa samaa kaavaa). Etusivulle tulee karusellinäköymä profiileista. Tärkeintä kuitenkin on, että sivustojen tulee olla käyttäjäystävälliset. Yllä mainitut ehdotuksia, joihin voin käyttää luovaa ajattelua prototyypin käyttöliittymää tehdessä.

Projektin moduulienrunгон tarkoituksena on yhdistää mm. luokat ja tiedostot halutuiksi kokonaisuuksiksi. Moduulien rungon tehtävänä on myös huolehtia tiedonvälityksestä, jotta esim. tietokannasta saatava tieto välittyy oikein ja oikeaan osoitteeseen. Projektimme tärkein attribuutti on tietokanta, johon asiakkaat ja terapeutit voivat tallettaa käyttäjätietonsa ja tarkastella niitä. Attribuutteihin sisältyy myös asiakkaiden valitsemat profiilikuvat, jotka tallennetaan tietokantaan. Poikkeus ja virhetilanteiden käsittelyyn on luotu virheentarkistukset, jotka ilmoittavat käyttäjälle esimerkiksi virheellisten tietojen syötöstä tekstikenttiin. Projektissa olemme käyttäneet PHP-kutsuja, joilla ohjaamme käyttäjän halutuille sivuille esimerkiksi ”kirjaudu.php”, jonka takana on kirjautumissivu ja ”config.php”, joka hakee tiedot tietokannasta.

4.2 Tietokannan luominen

```
CREATE DATABASE IF NOT EXISTS RTP_db; USE RTP_db;
```

```
CREATE TABLE terapeutit ( tp_id int(7) PRIMARY KEY,  
tp_etunimi varchar(25), tp_sukunimi varchar(50), tp_paikkakunta  
varchar(50), tp_esittelyteksti varchar(666) );
```

```
CREATE TABLE osaamiset ( osaamis_id int(2) PRIMARY KEY,  
osaamis_kuvaus varchar(75) );
```

```
CREATE TABLE hyvinvointi ( hyvinvointi_id int(2) PRIMARY  
KEY, hyvinvointi_kuvaus varchar(75) );
```

```
CREATE TABLE sairaudet ( sairaus_id int(2) PRIMARY KEY,  
sairaus_kuvaus varchar(75) );
```

```
CREATE TABLE terapeutin_osaamiset ( tposaamiset_id int(7)  
PRIMARY KEY, tp_fk int(7), osaamis_fk int(2), CONSTRAINT  
terapeutit_osaaminen_fk FOREIGN KEY (tp_fk)
```

```
REFERENCES terapeutit(tp_id), CONSTRAINT osaaminen_fk  
FOREIGN KEY (osaamis_fk) REFERENCES osaamiset(osaamis_id)  
);
```

```
CREATE TABLE terapeutin_hyvinvointi ( tphyvinvointi_id int(7)  
PRIMARY KEY, tp_fk int(7), hyvinvointi_fk int(2), CONSTRAINT  
terapeutit_hyvinvointi_fk FOREIGN KEY (tp_fk) REFERENCES  
terapeutit(tp_id), CONSTRAINT hyvinvointi_fk FOREIGN KEY  
(hyvinvointi_fk) REFERENCES hyvinvointi(hyvinvointi_id) );
```

```
CREATE TABLE terapeutin_sairaudet ( tpsairaudet_id int(7)  
PRIMARY KEY, tp_fk int(7), sairaus_fk int(2), CONSTRAINT  
terapeutit_sairaus_fk FOREIGN KEY (tp_fk) REFERENCES  
terapeutit(tp_id), CONSTRAINT sairaus_fk FOREIGN KEY  
(sairaus_fk) REFERENCES sairaudet(sairaus_id) );
```

```
CREATE TABLE asiakkaat ( asiakas_id int(7) PRIMARY KEY,  
asiakas_etunimi varchar(25), asiakas_sukunimi varchar(50),  
asiakas_osoite varchar(75), asiakas_email varchar(75),  
asiakas_puhnr varchar(25) );
```

```
CREATE TABLE kalenteri ( kalenteri_id int(5) PRIMARY KEY,  
kalenteri_pvm DATE, kalenteri_kellonaika varchar(10) );
```

```
CREATE TABLE terapeutin_kalenteri ( tpkalenteri_id int(7)  
PRIMARY KEY, kalenteri_fk int(5), tp_fk int(7),
```

```
varaustilanne int(1) DEFAULT 1, CONSTRAINT  
terapeutit_kalenteri_fk FOREIGN KEY (tp_fk) REFERENCES  
terapeutit(tp_id), CONSTRAINT kalenteri_fk FOREIGN KEY  
(kalenteri_fk) REFERENCES kalenteri(kalenteri_id) );
```

```
CREATE TABLE asiakasvaraukset ( asiakasvaraus_id int(7)  
PRIMARY KEY, tpkalenteri_fk int(7), asiakas_fk int(7),  
CONSTRAINT asiakkaat_kalenteri_fk FOREIGN KEY  
(tpkalenteri_fk) REFERENCES terapeutin_kalenteri(tpkalenteri_id),  
CONSTRAINT asiakkaat_fk FOREIGN KEY (asiakas_fk)  
REFERENCES asiakkaat(asiakas_id) );
```

4.3 Back-End

4.3.1 Ravitsemusterapeutin profiilin luonti

Ravitsemusterapeutin tulee pystyä luomaan profiili palveluun. Prototyyppiin luotava mahdollisuus profiilin luontiin, eli kirjautumis-/rekisteröitymismahdollisuudet. Profiili päivitettävä tietokantaan ja tietojen tulee olla sieltä haettavissa.

4.3.2 Ravitsemusterapeutin profiilin muokkaus

Ravitsemusterapeutin tulee pystyä muokkaamaan luomaansa profiilia. Ravitsemusterapeutin kirjaututtua profiililleen hänelle luotava mahdollisuus käyttöliittymän kautta valita profiilin muokkaustoiminto. Tiedot kirjaututtaessa haettava tietokannasta.

4.3.3 Ravitsemusterapeutin profiilin poisto

Ravitsemusterapeutin tulee halutessaan voida poistaa palvelusta profiilinsa. Kirjaututtuaan ravitsemusterapeutille tarjottava käyttöliittymässä mahdollisuus profiilin poistamiseen. Tiedot kirjaututtaessa haettava tietokannasta.

4.3.4 Haku suodatinten perusteella

Asiakkaan tulee pystyä hakemaan palvelusta erillisten prototyypin tilaajien määrittämien suodatinten avulla hänelle sopivaa ravitsemusterapeuttia. Terapeutit haetaan tietokannasta.

4.3.6 Haku paikkakunnan perusteella

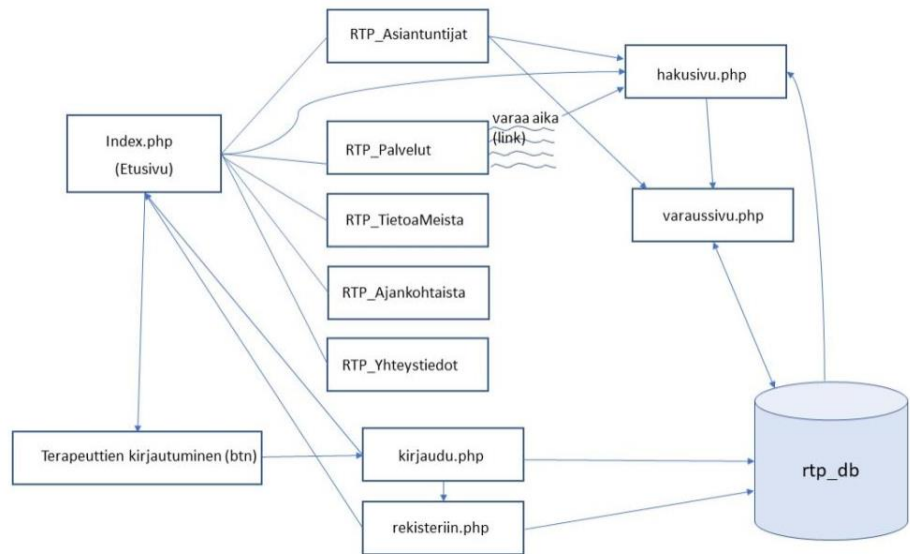
Asiakkaan tulee pystyä hakemaan palvelusta paikkakuntasuodattimen avulla hänelle sopivaa ravitsemusterapeuttia valitsemaltaan paikkakunnalta. Terapeutit haetaan tietokannasta.

4.3.7 Ajanvaraus

Mikäli ravitsemusterapeutti on lisännyt sivustolle vapaat aikansa, tulee asiakkaan kyetä tarkastelemaan ajanvarauskalenteria ja varata sieltä itselleen sopiva aika. Asiakkaan varatessa aika hän täyttää lomakkeen tyypillisesti omat tietonsa ja varaus viedään tietokantaan.

4.4 Moduulien yhdistäminen

Kun projektin moduulit on luotu erillisinä osina, tulee ne yhdistää yhdeksi kokonaisuudeksi. Moduulit yhdistetään WAMP-serverin, GIT:n ja Azure DevOps:n yhteiskäytön avulla. Täten saadaan yhdistettyä tietokanta MySQL:Stä ja koodillinen osuus VisualStudiosta. Kantaan päästään WAMP-serverin kautta PHPmyadmin:iin, josta käsin kanta voidaan muokata. Back-Endin toiminnallisuudet on saatava yhdistettyä FrontEndin käyttöliittymään ja tietokanta on saatava yhdistettyä Back-Endin toiminnallisuuksiin, jotta hakuja ja kyselyitä voidaan suorittaa reaaliajassa.



4.5 Testaus

Testausta suoritetaan jatkuvasti projektin edetessä niin uuden koodin kuin tietokannankin luomisprosessissa. Testaus on erittäin tärkeää, jotta työmäärä pysyy minimissään ja mahdolliset virheet ja poikkeamat havaitaan mahdollisimman aikaisessa vaiheessa. Kun moduulit on saatu yhdistettyä, alkaa projektin virallinen testausosio, jossa varmistetaan prototyypin toimivuus myös poikkeustilanteissa. Testauksessa tulee ottaa huomioon käyttöliittymän toiminta, sivuston varsinaiset toiminnallisuudet ja tietokannan käyttämisen onnistuminen. Tietokantaan tulee sivun kautta pystyä luomaan, hakemaan, muokkaamaan ja poistamaan dataa.

5. MUITA MAHDOLLISIA KOHTIA:

Prototyyppi ei vaadi varsinaista ylläpitoa projektin jälkeen. Prototyyppi ei mene viralliseen käyttöön, vaan tarkoituksena on luoda visuaalinen ja toiminnallinen havainnollinen palvelu, jonka avulla tilaaja näkee tuotteesta konkreettisen version ja saada kokemusta ohjelmistotuotannosta projektiryhmän jäsenille.

Prototyypin siirrettävyydestä on vaikeaa sanoa etukäteen mitään. Aikaisemmassa projektissaan jäsenet ovat huomanneet rajallisten työkalujen aiheuttamia haasteita projektin siirrettävyyden kannalta. Tarkemmat tiedot selviävät siinä vaiheessa, kun moduuleita on yhdistelty ja havaittu kuinka ne käyttäytyvät keskenään.

Virhetilanteita varten on luotava varasuunnitelmia ja ohjelma luotava niin, ettei se kaadu kesken ajan. Tätä varten käytettävä konkreettisia ratkaisuja, kuten esimerkiksi Front- ja Back-Endin puolella ”TryParse” -tyyppisiä luontirivejä.

Ratkaisuihin vaikuttanee projektin budjetti, joka on 0 €. Tästä syystä kaikki työkalut eivät esimerkiksi ole käytettävissä ja projektiryhmä joutunee käyttämään heille tuntemattomia työkaluja.