

Teemu Rantamäki

Varasto-ohjelmiston kehitysprojekti

Opinnäytetyö

Kevät 2019

SeAMK Tekniikka

2019 Tietotekniikka



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikka

Tutkinto-ohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Tietoverkkotekniikat

Tekijä: Teemu Rantamäki

Työn nimi: Varasto-ohjelmiston kehitysprojekti

Ohjaaja: Jyri Lehto

Vuosi: 2019

Sivumäärä: 41

SeAMK Liiketoiminta ja kulttuuri tarjosi opinnäytetyöaiheen varasto-ohjelman suunnittelusta ja kehityksestä, jolla kulttuurin opiskelijat voivat varata kuvauskalustoa omiin opiskeluprojekteihinsa internetin kautta. Varastonhoitajalle tulee oma sovellus ja kulttuurin opiskelijoille varausohjelma. Tuotteelle ei saa tulla päällekkäisiä varauksia samalle ajalle.

Molemmat sovellukset saatiin valmiiksi Windows-työpöytä-sovelluksina. Tietokantana on Microsoft Access Database, joka ei ole vielä palvelimella saatavilla. Työ jatkuu vielä opinnäytetyön jälkeen, jolloin muutetaan tietokanta SQL-muotoon palvelimelle ja tehdään verkkosivu opiskelijoille, mistä he voivat varata kuvauslaitteita.

Avainsanat: tietokantahallintajärjestelmä, varastokirjanpito, lainausjärjestelmät, ohjelmointi, käyttöliittymä

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Networking Technology

Author: Teemu Rantamäki

Title of thesis: Warehouse Program Development Project

Supervisor: Jyri Lehto

Year: 2019

Number of pages: 41

The School of Business and Culture in Seinäjoki University of Applied Sciences offered this thesis project of developing stock management software. This software is meant for a supply room, from which the students of culture and arts can reserve camera equipment for their own school projects. The manager of the supply room has his own program for managing inventory. It is important to develop the program so that multiple reservations for a single item at same time are not possible.

Both programs were written as Windows desktop programs and they both use the Microsoft Access Database as the inventory's database. This has not been published on the Internet yet. The development of the programs is still continuing. The database will be converted to an SQL server. Also a web page for students' supply reservations is still under development.

Keywords: database system, stock bookkeeping, reservation system, programming, user interface

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	1
Thesis abstract	2
SISÄLTÖ.....	3
Kuvaluettelo	5
Käytetyt termit ja lyhenteet	7
1 JOHDANTO.....	8
1.1 Työn tausta	8
1.2 Työn tavoite.....	8
1.3 Työn rakenne	8
1.4 Kulttuuriala SeAMKissa.....	9
2 TIETOKANNAT	10
2.1 Microsoft Access Database.....	10
2.2 SQL-tietokanta	10
3 MICROSOFT VISUAL STUDIO	12
3.1 Työpöytäsovelluksen käyttöliittymä	12
3.2 Yhteys tietokantaan.....	13
3.3 Web Service.....	16
3.4 Regex.....	16
3.5 Android Xamarin	16
4 SUUNNITTELU	18
4.1 Tietokannan suunnittelu	18
4.2 Varasto-ohjelman suunnittelu.....	19
4.3 Varausohjelman suunnittelu	19
5 TIETOKANNAN TULOS.....	21
5.1 Asiakkaat-taulu.....	21
5.2 Tuotteet-taulu	22
5.3 Varaukset-taulu	23
6 VARASTO-OHJELMAN TULOS	24
6.1 Tuotteet-ikkuna	29

6.2	Kategoriat-ikkuna	30
6.3	Asiakkaat-ikkuna	31
6.4	Varaukset-ikkuna	32
7	VARAUSOHJELMAN TULOS.....	33
7.1	Ensimmäinen välilehti.....	34
7.2	Toinen välilehti	35
8	YHTEENVETO JA POHDINTA.....	40
8.1	Tulos opinnäytetyön aikana.....	40
8.2	Varausohjelman kehittäminen opinnäytetyön jälkeen	40
	LÄHTEET.....	41

Kuvaluettelo

Kuva 1. using OleDb	13
Kuva 2. Yhteys tietokantaan	13
Kuva 3. Valitaan, mitä luetaan tietokannasta	13
Kuva 4. Otetaan tietoja lukijasta.....	14
Kuva 5. Mihinkä tauluun lisätään	14
Kuva 6. Mitä lisätään.....	14
Kuva 7. Tietueen päivitys.....	15
Kuva 8. Tietueen poistaminen.....	15
Kuva 9. Uuden tietueen ID-numeron haku.....	15
Kuva 10. Regexin using-komento	16
Kuva 11. Toimintokaavio varauksen tekemisestä	20
Kuva 12. Tietokannan relaatiot	21
Kuva 13. Asiakkaat-taulu	21
Kuva 14. Tuotteet-taulu.....	22
Kuva 15. Tuotteen varaukset	22
Kuva 16. Poistettava varaus	22
Kuva 17. Varaukset-taulu.....	23
Kuva 18. Varasto-ohjelman aloitusikkuna	24
Kuva 19. Viikonumeron hakeminen.....	24
Kuva 20. Viikon päivämäärät listaan	25

Kuva 21. Päivämäärä talteen	25
Kuva 22. Sarakkeisiin päivät	26
Kuva 23. Haetaan haku- ja palautusajat	26
Kuva 24. Syötetään haku- ja palautusajat taulukkoon	27
Kuva 25. Järjestetään kellonajan mukaan.....	28
Kuva 26. Varasto-ohjelman Tuotteet-ikkuna	29
Kuva 27. Varasto-ohjelman kategoriat-ikkuna.....	30
Kuva 28. Varasto-ohjelman asiakkaat-ikkuna	31
Kuva 29. Varasto-ohjelman varaukset-ikkuna.....	32
Kuva 30. Varausohjelman aloitusikkuna	33
Kuva 31. Varausohjelman ensimmäinen välilehti.....	34
Kuva 32. Varausohjelman toinen välilehti	35
Kuva 33. Kategoriat puuhun.....	36
Kuva 34. Tuotteet puuhun.....	37
Kuva 35. Checked-bool.....	37
Kuva 36. Puun tuotteet listaan	38
Kuva 37. Kutsutaan metodia	38
Kuva 38. Kellonajan valitsin	39

Käytetyt termit ja lyhenteet

Access Database	Microsoftin kehittämä tietokantatyyppe, joita voidaan tehdä Access 2016 –ohjelmalla.
ASP.NET	Ohjelma, joka toimii selaimella. Ulkonäkö on koodataan HTML-kielellä ja CSS-kielellä. Toimintoja voidaan ohjelmoida C#-kielellä.
C#	Microsoftin kehittämä olio-ohjelmointikieli.
CSS	Cascading Style Sheets. Voidaan muokata HTML-tiedoston rakenteiden tyyliä.
HTML	Hypertext Markup Language. Tällä voidaan rakentaa internet-sivuja.
JavaScript	Ohjelmointikieli, jolla voidaan ohjelmoida esim. HTML-tiedostoon toimintoja.
jQuery	Yksinkertaistettu JavaScript-kirjasto.
Kontrolli	Sovelluksen komponentti, johon voi kirjoittaa tai josta voi valita arvon. Näitä ovat esim. tekstikenttä, alavetovalikko, päivämäärän valitsin ja valintaruutu.
MS SQL	Microsoftin kehittämä SQL-palvelimen ylläpito-ohjelma.
MYSQL	Oraclen kehittämä SQL-pilvipalvelu.
SQL	SQL-lauseilla voidaan hakea, muokata, lisätä ja poistaa tietoja SQL-tietokannoista ja Access-tietokannoista.
Visual Studio	Microsoftin kehittämissympäristö, jolla voi tehdä monentyyppisiä sovelluksia ja eri ohjelmointikielillä.

1 JOHDANTO

1.1 Työn tausta

SeAMK Liiketoiminta ja kulttuuri tarjosi opinnäytetyöaiheen uuden varasto-ohjelman suunnittelusta ja tekemisestä. Aiemmassa kulttuurin yksikön käyttämässä varastohallintaohjelmassa oli se ongelma, että sillä pystyi varaamaan saman tuotteen monelle opiskelijalle samalle ajalle. Ohjelman suunnittelu ja toteutus aloitettiin täysin alusta. Varasto-ohjelmaa tulevat käyttämään kulttuurin opiskelijat ja varastonhoitajat. Kulttuurin yksikkö oli miettinyt ostavansa varastohallintaohjelman, mutta sen hinta koettiin liian kalliiksi.

1.2 Työn tavoite

Varastohallintaohjelman tavoitteena on, että kulttuurin opiskelijat voivat varata kuvaustarvikkeita omiin projekteihinsa internetin kautta. Yhtä tavaraa ei pidä pystyä varaamaan monelle opiskelijalle samalle ajalle. Varastonhoitajat voivat ylläpitää varastoa. Varastonhoitajalle ja opiskelijoille tehdään oma sovellus.

1.3 Työn rakenne

Luvussa kaksi kerrotaan Microsoft Access- ja SQL-tietokannoista.

Luvussa kolme kerrotaan käyttöliittymästä ja c#-ohjelmoinnista.

Luvussa neljä kerrotaan tietokannan ja sovellusten alkusuunnittelusta.

Luvussa viisi kerrotaan, minkälainen oli tietokannan taulujen lopputulos.

Luvussa kuusi kerrotaan varastonhoitajan sovelluksen tuloksesta.

Luvussa seitsemän kerrotaan varausohjelman tuloksesta.

Luvussa kahdeksan kerrotaan tuloksista yleisesti, ja kuinka kehittäminen jatkuu.

1.4 Kulttuuriala SeAMKissa

Seinäjoen Ammattikorkeakoulussa on mahdollista opiskella kulttuurituottajaksi. Opinnot kestävät noin neljä vuotta ja tutkinnon kokonaislaajuus on 240 opintopistettä. Kulttuurituottajat opiskelevat mm. tuotantotekniikkaa, tuottamista, audiovisuaalisia menetelmiä sekä markkinointia ja budjetointia. Kulttuurituottajat toimivat työelämässä kulttuuritapahtumien ja –tuotantojen asiantuntijoina. (SeAMK. [Viitattu 16.3.2019].)

2 TIETOKANNAT

Varasto-ohjelmaan tarvitaan tietokanta, johon varastonhoitaja ja opiskelija voivat ohjelmillaan muodostaa yhteyden. Tietokannan vaihtoehtoiksi valittiin Microsoftin Access Database ja SQL-tietokanta. Tunnetuimmat SQL-palvelimien tarjoajat ovat Microsoftin kehittämä MS SQL ja Oraclen kehittämä MYSQL. Tässä työssä käytetään ensin Access Databasea. Access-tietokannan voi myöhemmin muuntaa SQL-tietokannaksi. Kumpaakin tietokantaa luetaan ja päivitetään SQL-komennoilla: select, insert, update ja delete. Kummankin tietokannan rakenne on samanlainen, eli rivi on tietue ja sarakkeissa on tietueen tiedot.

Tietokannoissa Asiakkaan etunimi ja sukunimi kannattaa laittaa erikseen, jotta on helpompi lajitella etunimen tai sukunimen mukaan. Jokaisessa taulussa kannattaa aina ensimmäisen sarakkeen tyyppi olla AutoNumber. AutoNumberiin tulee seuraava vapaa numero, kun tehdään uusi tietue. Taulun tietueen voi laittaa relaatioon toisen taulun tietueeseen. Esimerkiksi asiakkaat-tilin ID-numero-tietue voi olla yksi-moneen-relaatioissa lainaukset-tilin asiakasnumero-tietueen kanssa. Näin asiakkaan tiedoista voi nähdä kaikki hänen lainauksensa ja lainauksesta näkee, kenenkä lainaus se on. (Bulleen 2012, 17.)

2.1 Microsoft Access Database

Microsoft Office -sovellukseen kuuluvalla Microsoft Accessillä voi tehdä tietokantoja. Tässä työssä on tehty tietokanta Access Database (2007) -muodossa, jonka tiedostomuoto on accdb. Tietokantaan voidaan ottaa yhteys c#-kielellä OleDb-komennoilla. Tietokannan maksimikoko on 2 gigabittia, maksimikäyttäjämäärä samaan aikaan on 255, ja kohteiden maksimimäärä on 32,768 (lan 2016).

2.2 SQL-tietokanta

SQL-tietokannan maksimikoko on 16 terabittia, samanaikaisia käyttäjiä voi olla maksimissaan 32 767 ja kohteiden maksimimäärä on 2 147 487 647 (lan 2016). Jos reitittimessä ei ole valmiiksi rakennettua palomuuria, niin kannattaa käyttää laitteen

omaa Access Control Listia (ACL) hallitakseen, mitä portin yhteyksiä saa mennä reitittimen läpi. Mutta ACL ei suojaa DDoS (Distributed Denial of Service) -hyökkäyksiä vastaan. DDoS-hyökkäyksessä tukitaan verkon liikenne lähettämällä erittäin paljon pyyntöjä samaan aikaan. (Cherry 2015, 8.)

3 MICROSOFT VISUAL STUDIO

Microsoft Visual Studiolla voi tehdä monentyyppisiä sovelluksia monilla eri ohjelmointikielillä. Esimerkiksi C#-kielellä voi tehdä konsoli, työpöytä-, .NET-, Android- ja Apple-ohjelmia.

3.1 Työpöytäsovelluksen käyttöliittymä

Käyttöliittymässä pitää käyttää tuttuja termejä, ja tekstijonoissa on hyvä olla mahdollisimman vähän tekstiä. Tekstien leveys kannattaa olla maksimissaan 65 merkkiä. Sinistä tekstiä ei kannata käyttää, koska ne ovat yleensä linkkejä. Tärkeimmissä teksteissä voi käyttää tummennettua fonttia. Tekstien pitää mahtua sarakkeisiin kokonaan ja sarakkeet eivät saa olla liian leveitä. (Kennedy & Satran 2018.)

Käyttöliittymän ikkunassa on hyvä olla yhtä paljon komponentteja joka puolella, jotta ohjelma näyttää tasaiselta. Komponentteja voi suurentaa tai siirtää, jos ohjelmassa on liikaa tyhjää tilaa. Kun ikkunan kokoa muutetaan, komponenttien koon pitää vaihtua. Ikkunan minimikoko määritellään sen mukaan, kuinka pienessä tilassa ikkunan komponentteja pystyy vielä käyttämään. (Kennedy & Satran 2018.)

Ohjelman käynnistyessä kontrolleissa kannattaa olla oletuksena kaikista turvallisin arvo. Jos mahdollista, on hyvä käyttää tekstikenttien sijasta rajoittavia kontrolleja, kuten listoja, alasvetovalikkoja ja päivämäärän valitsimia. Käytä-painikkeen klikkaamisen jälkeen ikkunan pitää jäädä auki. Tab-näppäintä painamalla hyppäämisen seuraavaan kontrolliin pitää tapahtua vasemmalta oikealle ja ylhäältä alaspäin. Tab-näppäimellä ja nuolinäppäimillä hyppäämisen pitää osua kaikkiin kontrolleihin. Toimintopainikkeet tulevat tähän järjestykseen: Kyllä/OK, Ei, Peruuta ja Käytä. (Kennedy & Satran 2018.)

3.2 Yhteys tietokantaan

Seuraavia ohjelmointiesimerkkejä on poimittu varasto-ohjelman eri kohdista, joten muuttujien nimet vaihtelevat. Accdb-tietokannan lukemiseen ja muokkaukseen tarvitaan using-komento, kuten kuvassa 1.

```
using System.Data.OleDb;
```

Kuva 1. using OleDb

Sitten määritellään OleDb-yhteys accdb-tietokantaan (kuva2).

```
OleDbConnection yhteys = new OleDbConnection(
    @"provider=Microsoft.ACE.OLEDB.12.0;
    data source = C:\Varastonhallinta\Varasto.accdb");
```

Kuva 2. Yhteys tietokantaan

Jos halutaan hakea tietoa, tehdään OleDb-lukija ja OleDb-komento, jossa on SQL-lause. SQL-lauseen jälkeen kerrotaan, mihin tietokantaan yhteys muodostetaan. Sitten aktivoidaan OleDb-lukija.

```
OleDbDataReader drAsiakas;
OleDbCommand HaeAsiakas = new OleDbCommand(
    "SELECT ID, Tunnus, Sähköposti, " +
    "Puhelin FROM Asiakkaat WHERE ID="
    + aID + "", yhteys);

yhteys.Open();

drAsiakas = HaeAsiakas.ExecuteReader();
```

Kuva 3. Valitaan, mitä luetaan tietokannasta

Seuraavaksi haetaan tietoja lukijasta ja tallennetaan niitä tekstikenttiin. Hakasulkeiden sisällä oleva numero tarkoittaa, kuinka mones tieto otetaan select-lauseesta. Numero nolla on tässä tapauksessa taulun ID-kenttä. Lopuksi pitää muistaa sulkea OleDb-yhteys. OleDb-lukija ja -komennot kannattaa myös tyhjentää.

```

while (drAsiakas.Read())
{
    tbID.Text = drAsiakas[0].ToString();
    tbTunnus.Text = drAsiakas[1].ToString();
    tbSahkoposti.Text = drAsiakas[2].ToString();
    tbPuhelin.Text = drAsiakas[3].ToString();
}

drAsiakas = null;
HaeAsiakas = null;
yhteys.Close();

```

Kuva 4. Otetaan tietoja lukijasta

Uuden tietueen lisäämisessä tarvitaan myös OleDb-yhteys ja -komento. Nyt käytetään OleDb-komennoissa insert-lausetta, kuva 5.

```

OleDbCommand cmd = new OleDbCommand("INSERT INTO Tuotteet " +
    "(Nimi, Kategoria, Lisaosat, Huomioitavaa) " +
    "VALUES(@nimi, @kategoria, @lisaosat, @huomioitavaa)", con);

con.Open();

```

Kuva 5. Mihinkä tauluun lisätään

Seuraavaksi valitaan, mitä tietoja lisätään uuteen tietueeseen. ExecuteNonQuery-komennoilla lisätään tietue tietokantaan, kuva 6.

```

cmd.Parameters.Add("@nimi", OleDbType.VarChar).Value = tbNimi.Text;
cmd.Parameters.Add("@kategoria", OleDbType.VarChar).Value = comboBox2.SelectedValue;
cmd.Parameters.Add("@lisaosat", OleDbType.VarChar).Value = tbOheis.Text;
cmd.Parameters.Add("@huomioitavaa", OleDbType.VarChar).Value = tbHuomioitavaa.Text;

cmd.ExecuteNonQuery();
con.Close();

```

Kuva 6. Mitä lisätään

Tietueen päivittämisessä käytetään update-lausetta, kuva 7.

```

using (OleDbCommand cmd2 = con2.CreateCommand())
{
    cmd2.CommandText = @"UPDATE Asiakkaat SET Tunnus=@tunnus,
Nimi=@nimi, Sähköposti=@sähköposti, Puhelin=@puhelin WHERE ID="
+ dataGridView1.SelectedCells[0].Value.ToString() + ";

    cmd2.Parameters.AddWithValue("@tunnus", tbTunnus.Text);
    cmd2.Parameters.AddWithValue("@nimi", tbNimi.Text);
    cmd2.Parameters.AddWithValue("@sähköposti", tbSähköposti.Text);
    cmd2.Parameters.AddWithValue("@puhelin", tbPuhelin.Text);

    con2.Open();
    cmd2.ExecuteNonQuery();
    con2.Close();
}

```

Kuva 7. Tietueen päivitys

Tietueen poistamisessa käytetään delete-lausetta, kuva 8.

```

using (var komento = yhteys.CreateCommand())
{
    var kategoriaID = new OleDbParameter(
        "@id", dataGridView2.SelectedCells[0].Value);

    komento.CommandText = "DELETE FROM Kategoriat " +
        "WHERE (ID) = @id";

    komento.Parameters.Add(kategoriaID);

    yhteys.Open();
    komento.ExecuteNonQuery();
    yhteys.Close();
}

LataaKategoriat();

```

Kuva 8. Tietueen poistaminen

Uuden tietueen lisäämisessä voidaan hakea uuden tietueen id-numeron, kuva 9.

```

public static int HaeViimeisinAutonumber(OleDbConnection yhteys)
{
    using (OleDbCommand komento = new OleDbCommand("SELECT @@IDENTITY;", yhteys))
    {
        return (int)komento.ExecuteScalar();
    }
}

```

Kuva 9. Uuden tietueen ID-numeron haku

3.3 Web Service

SQL-tietokantaan ei kannata ottaa suoraan yhteyttä, vaan yhteys kannattaa ottaa web servicen kautta, joka on turvallisempaa. Web servicellä ohjelma voi vedota toisen ohjelman metodiin joko samaan tietokoneeseen tai toiseen tietokoneeseen. Web servicet käyttävät HTTP-, XML- ja SOAP-protokollia, joita kaikki laitteet voivat käyttää. Ennen web servicen käyttämistä tarvitaan proxy. Proxy on rekisteröity asiakasohjelman kanssa, ja asiakasohjelma soittaa web serviceen. Proxy vastaanottaa puhelut, muuttaa ne oikeaan muotoon ja lähettää SOAP-pyyntöä palvelimeen. Sitteen SOAP välittää web servicen dataa. (Samantaray 2017.)

3.4 Regex

Regexista on apua tiettyjen merkkien etsimisessä, poimisessa ja/tai poistamisessa tekstijonosta. Ohjelman alkuun tarvitaan seuraava komento, joka on kuvassa 10.

```
using System.Text.RegularExpressions;
```

Kuva 10. Regexin using-komento

Haku määritellään erikoismerkeillä [], { }, (), *, +, \, ?, |, \$, . ja ^. { }-merkkien sisällä merkitään, kuinka monta kertaa tietyn merkkijonon pitää esiintyä. []-merkkien kaikkia sisällä olevia merkkejä etsitään erikseen. Jos [] -merkkien sisällä laitaa viivan kahden merkin välille, niin kaikki merkit näiden kahden väliltä etsitään. Kaikki merkit, jotka ovat ^-merkin jälkeen, ovat pois etsinnästä. \d-yhdistelmällä etsitään kaikki numeroarvot. \D-yhdistelmällä etsitään kaikki muut kuin numerot. (Strauss 2016, 291.)

3.5 Android Xamarin

Visual Studiolla voi käyttää C#-ohjelmointia Android Xamarin -ohjelman komentojen teossa. OleDb-komennot eivät enää toimi Android-ohjelmassa, joten jos tekee Androidille varauksen teon ohjelman, pitää muuttaa tietokanta SQL-muotoon.

Mobiililaitteissa on kaikenkokoisia näyttöjä, joten sovelluksen tila voidaan täyttää reunoille asti "match_parent"-komennolla. Minimikoko, johon mahtuu kaikki sisältö, määritellään komennolla "wrap_content". (Thornsby 2016, 35.) Tekstikenttään johon syötetään salasana, kannattaa laittaa: android:inputType="textPassword" (Thornsby 2016, 58).

4 SUUNNITTELU

Kulttuurilla oli aiemmin varasto-ohjelma, jolla pystyi varaamaan saman tuotteen monelle opiskelijalle samanaikaisesti. Vanha ohjelma oli yhteydessä SQL-tietokantaan, joka ei ole enää käytössä.

Tavoitteena oli tehdä varastonhoitajalle varastonhallintaohjelma, jolla voi ylläpitää tietokantaa tuotteista, opiskelijoista ja varauksista. Opiskelijoille pitää tehdä eri ohjelma, jolla voi tehdä pelkästään varauksia. Molemmat ohjelmat ovat internet-yhteydessä samaan tietokantaan. Samaa tuotetta ei pidä pystyä lainaamaan samalle ajalle monelle opiskelijalle. Varausohjelman käyttö pitää sallia vain Kulttuurin opiskelijoille.

Ohjelmien ja tietokannan suunnittelu ja toteutus aloitettiin täysin alusta. Ohjelmat tehtiin ensin tämän työn tekijälle tutuimmalla tavalla, eli Microsoft Visual Studiolla C#-ohjelmointikielellä Windows Formsille. Tietokantana käytetään aluksi Microsoft Access Databasea. Kun ohjelmat ja Access-tietokanta toimivat täysin offline-tilassa, voidaan toteuttaa tietokantaa internetiin, muuttamalla tietokanta SQL-muotoon koulun SQL-palvelimelle tai laittamalla Access Database koulun palvelimelle. Opiskelijoille voisi tehdä Android-ohjelman tai verkkosivun varauksille, sitten kun tietokanta on toteutettu internetiin.

4.1 Tietokannan suunnittelu

Tuotteille on eri kategorioita. Yhtä tuotetta ei ole kymmentä kappaletta enempää, joten kaikki tuotteet voivat olla yksilötuotteita. Tuotteelle ei tarvitse merkitä lukumäärää ja tehdä jonkinlaista laskuria, joka laskee, kuinka paljon tuotetta on saatavilla. Tuotteesta näkee lainauspäivät, kenellä se on tällä hetkellä lainassa ja keille se on menossa seuraavaksi. Tuotteen tiedoista näkee, onko se ehjä.

Asiakkaalle laitetaan opiskelijatunnus, salasana, nimi, sähköposti ja puhelinnumero.

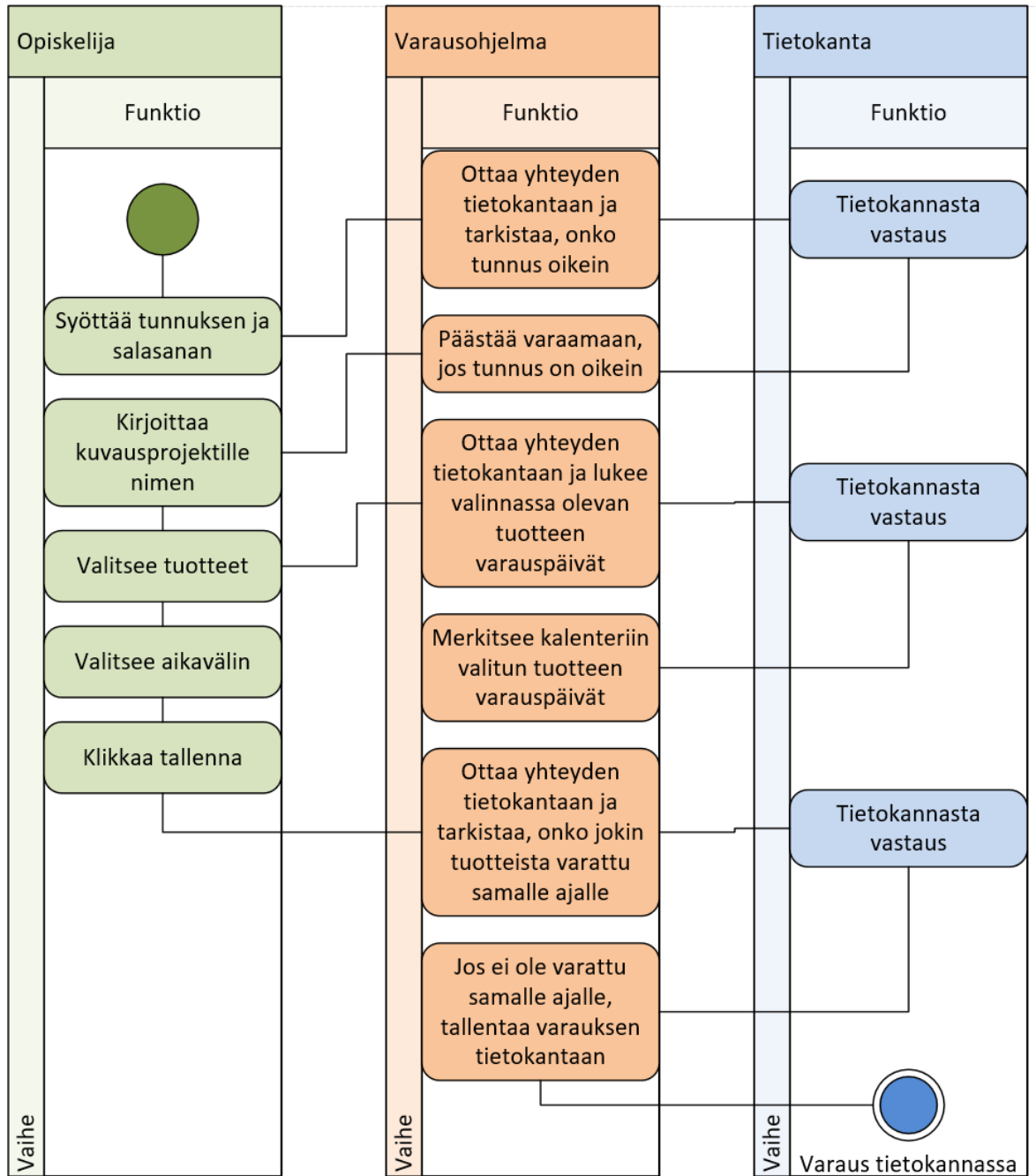
Varauksista näkee opiskelijan, joka on tehnyt varauksen, ja kaikki tuotteet, jotka kuuluvat samaan varaukseen. Varauksessa on varauksen alku- ja loppu-päivämäärät. Kummatkin ohjelmat ovat yhteydessä samaan tietokantaan.

4.2 Varasto-ohjelman suunnittelu

Varasto-ohjelma tehdään C#-kielellä Windows-työpöytä-sovelluksena. Varasto-ohjelmaa käytetään vain paikan päällä, joten siitä ei tarvitse tehdä verkkosivua tai kännykkäsovellusta. Varastonhoitajan ohjelmalla voidaan lisätä, muokata ja poistaa tuotteita ja kategorioita. Asiakkaiden ja varauksien tietoja voi myös tarkastella ja muokata. Kun varauksen tuotteiden palautus on myöhässä, ohjelma ilmoittaa siitä. Ohjelma voisi myös lähettää sähköpostiviestin opiskelijalle, jos varaus on myöhässä. Kun varastonhoitaja on kuitannut varauksen tuotteet palautetuksi, tuotteiden varaukset siirtyvät seuraaville varauksille.

4.3 Varausohjelman suunnittelu

Ensin tehdään C#-kielellä Windows-työpöytä-sovellus. Myöhemmin tehdään verkkosivu tai kännykkäsovellus. Varausohjelmassa opiskelija voi tarkastella omien varauksiensa tietoja tai tehdä uuden varauksen. Opiskelijalla voi olla maksimissaan kolme meneillään olevaa tai tulossa olevaa varausta samaan aikaan. Lainaukselle on pakko antaa projektinimi. Varaus pitää tehdä viikkoa ennen lainausajan alkua. Varauksen teossa opiskelija lisää haluamansa tuotteet ”ostoskoriin” ja valitsee aikavälin lainaukselle. Kuvassa 11 on askeleet varauksen teolle.

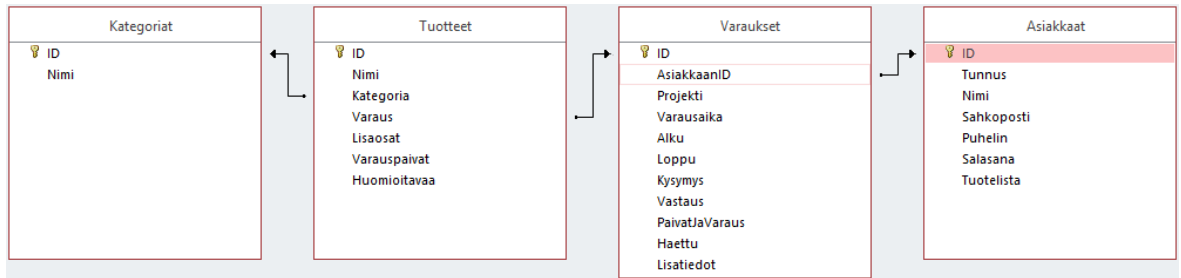


Kuva 11. Toimintokaavio varauksen tekemisestä

Jos varauksille tehdään oma verkkosivu, voisi samalla miettiä SeAMKin käyttämää kirjautumisen tunnistusmenetelmää käytettäväksi myös tässä. Samalla voitaisiin tehdä rajaus vain kulttuurin opiskelijoille. SeAMKin kirjautumista käytettäisiin vain päästääkseen verkkosivulle. Varausverkkosivulla varauksien tekoon käytetään varaston tietokantaan kuuluvaa tunnusta.

5 TIETOKANNAN TULOS

Tietokannassa on Kategoriat-, Tuotteet-, Asiakkaat- ja Varaukset-taulut. Kategoriat-taulussa on vain kentät ID ja Nimi. Kategoriat-taulun ID-kenttä on relaatiossa Tuotteet-taulun Katogoria-kentän kanssa, jonka tyyppi on myös kokonaisluku.



Kuva 12. Tietokannan relaatiot


5.1 Asiakkaat-taulu

Asiakkaat
ID
Tunnus
Nimi
Sahkoposti
Puhelin
Salasana
Tuotelista

Kuva 13. Asiakkaat-taulu

Kaikki kentät, paitsi ID, ovat lyhyttä tekstiä. Tunnus-kenttään syötetään opiskelija-tunnus. Asiakkaat voivat ohjelmassa painiketta klikkaamalla tallentaa tuotelistan muistiin. Tuotelista-kentässä on tuotteiden ID-numerot peräkkäin pilkulla eroteltuina ja tuotelista päällekirjoitetaan joka kerta.

5.2 Tuotteet-taulu

Tuotteet	
	ID
	Nimi
	Kategoria
	Varaus
	Lisaosat
	Varauspaivat
	Huomioitavaa

Kuva 14. Tuotteet-taulu

Nimi- ja Huomioitavaa-kentät ovat lyhyttä tekstiä. Kategoria- ja Varaus-kentät ovat kokonaislukuja. Lisaosat- ja Varauspaivat-kentät ovat pitkää tekstiä. Rikki-kentän tilalle on vaihdettu Huomioitavaa-kenttä.

Kategoria-kenttään tulee kategorian ID-numero. Kun tuotteet on palautettu, Varaus-kenttään ei tule seuraavan vuorossa olevan varauksen ID-numeroa, vaan siihen tulee sen varauksen ID-numero, jolla tuote on tällä hetkellä lainassa. Jos tuote ei ole kenelläkään tällä hetkellä lainassa, niin silloin Varaus-kentän numero on nolla. Lisaosat-kentässä on tuotteen mukaan kuuluvat oheislaitteet. Ne erotellaan pilkulla ja välilyönneillä. Ohjelma luettelee ne listana, jotta on helpompi lukea.

Varauspaivat-kenttään tallennetaan kaikkien tulevien varauksien ID-numerot ja varauspäivät. ID-numero on sulkeiden sisällä ja varauspäivät ovat ID-numeron jälkeen välilyönneillä eroteltuna. Varauksen lopetusmerkki on puolipiste. Kuvassa 15 esimerkki yhden tuotteen varauksista:

`(75) 30.3.2019 31.3.2019 ;(76) 1.4.2019 2.4.2019 3.4.2019 ;`

Kuva 15. Tuotteen varaukset


Kun varauksen tuote on palautettu, tuotteen varauspaivat-kentästä poistuu sen varauksen tekstijono. Kuvassa 16 näytetään, mikä tekstijono poistetaan, jos varauksen numero 75 tuote palautetaan.

`(75) 30.3.2019 31.3.2019 ;(76) 1.4.2019 2.4.2019 3.4.2019 ;`

Kuva 16. Poistettava varaus

C#-sovelluksessa komento `Regex.Replace(string-muuttujan nimi, @"\"(. *?)", "")`; poistaa tekstijonosta ()-merkit ja niiden sisällä olevat merkit. Jonosta poistetaan vielä `replace`-komennolla puolipisteet. Näin voidaan jonosta poimia päivämäärät, jotka näkyvät tummennettuina ohjelman kalenterissa.

5.3 Varaukset-taulu

Varaukset	
	ID
	AsiakkaanID
	Projekti
	Varausaika
	Alku
	Loppu
	Kysymys
	Vastaus
	PaivatJaVaraus
	Haettu
	Lisätiedot

Kuva 17. Varaukset-taulu

AsiakkaanID on kokonaisluku. Projekti on lyhyt teksti. Varausaika, Alku ja Loppu ovat tyyppiä päivämäärä/aika. Kysymys, Vastaus, PaivatJaVaraus ja Lisätiedot ovat pitkää tekstiä. Haettu on tyyppiä Kyllä/Ei.

Varausaika-kentässä on päivämäärä ja kellonaika, jolloin varaus on tehty. Alku- ja loppu-kentissä ovat varauksen aloitus- ja lopetuspäivämäärät ja kellonajat. PaivatJaVaraus-kenttään tallentuu varauksen ID-numero ja varauspäivät. Kun tuote palautetaan, ohjelma lukee Varaukset-taulun PaivatJaVaraus-kentästä tekstijonon, jonka pitää poistaa Tuotteet-taulun Varauspaivat-kentästä. Kun ohjelmassa laiteetaan rasti ruutuun Haettu-kenttään, niin Tuotteet-taulun Varaus-kentän numeroksi muuttuu kyseisen varauksen ID-numero. Opiskelija voi kirjoittaa kysymyksen, ja varastonhoitaja voi vastata kysymykseen. Varastonhoitaja voi kirjoittaa tekstin Lisätiedot-kenttään.

6 VARASTO-OHJELMAN TULOS

Valitun päivän viikkokalenteri näkyy alhaalla

Kategoriat Tuotteet Asiakkaat Varaukset

huhtikuu 2019

ma	ti	ke	to	pe	la	su
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

tänään: 17.4.2019

Viikon päivien tapahtumat

	Maanantai 22.4	Tiistai 23.4	Keskiviikko 24.4	Torstai 25.4	Perjantai 26.4
8 - 9					
9 - 10				Klo 9.0 asiakas2 hakee projektin projekti2 kamat.	
10 - 11			Klo 10.0 asiakas1 hakee projektin projekti1 kamat.		
11 - 12					
12 - 13				Klo 12.0 asiakas2 palauttaa projektin projekti2 kamat.	
13 - 14					
14 - 15				Klo 14.0 asiakas1 palauttaa projektin projekti1 kamat.	
15 - 16					
16 - 17					
17 - 18					
18 - 19					
19 - 20					

Kuva 18. Varasto-ohjelman aloitusikkuna

Ohjelman aloitusikkunassa on neljä painiketta, joista voi avata Kategoriat, Tuotteet, Asiakkaat ja Varaukset-ikkunat. Ensimmäisessä ikkunassa on myös kalenteri. Kun klikkaa kalenterin jotain päivää, kyseisen viikon tuotteiden haku- ja palautusajat näkyvät alapuolella olevassa taulukossa, joka on tehty näyttämään viikkokalenterilta. Kuvassa 19 esitetään, kuinka saadaan päivän viikkonumero.

```
public int WeekNumber(DateTime date)
{
    CultureInfo ciCurr = CultureInfo.CurrentCulture;
    int weekNum = ciCurr.Calendar.GetWeekOfYear(date,
        CalendarWeekRule.FirstFourDayWeek,
        DayOfWeek.Monday);
    return weekNum;
}
```

Kuva 19. Viikkonumeron hakeminen

Seuraavaksi tehdään lista viikonpäiville. Tällöin metodi, joka hakee valitun päivän viikon arkipäivien päivämäärät ja laittaa ne listaan.

```
List<DateTime> viikonpaivat = new List<DateTime>();

2 references | 0 changes | 0 authors, 0 changes
private void LataaHautJaPalautukset()
{
    dataGridView1.Columns.Clear();
    dataGridView1.Rows.Clear();
    viikonpaivat.Clear();

    // haetaan valitun päivän viikkonumero
    int viikko = WeekNumber(monthCalendar1.SelectionRange.Start);

    // viikkonumeron jokainen päivä listaan
    for (int i = 1; i < 8; i++)
    {
        DateTime jan1 = new DateTime(DateTime.Now.Year, 1, 1);

        int daysOffset = DayOfWeek.Monday - jan1.DayOfWeek;

        DateTime firstMonday = jan1.AddDays(daysOffset);

        var cal = CultureInfo.CurrentCulture.Calendar;

        int firstWeek = cal.GetWeekOfYear(
            jan1, CalendarWeekRule.FirstFourDayWeek,
            DayOfWeek.Sunday);

        var weekNum = viikko;

        if (firstWeek <= 1)
        {
            weekNum -= 1;
        }

        var result = firstMonday.AddDays(weekNum * 7 + i - 1);

        viikonpaivat.Add(result);
    }
}
```

Kuva 20. Viikon päivämäärät listaan

Kuvassa 21 otetaan ensimmäinen päivämäärä eli maanantai listasta. Sen eteen ja taakse lisätään sulut, jotta varmasti on oikea päivä kyseessä. Muut päivät otetaan samalla lailla.

```
string maanantai2 = "(" + viikonpaivat[0].Date.Day.ToString() +
    "." + viikonpaivat[0].Date.Month.ToString() +
    "." + viikonpaivat[0].Date.Year.ToString() + ")";
```

Kuva 21. Päivämäärä talteen

Kuvassa 22 lisätään taulukon sarakkeisiin päivät. Päivämäärien muuttujat, joissa ei ole numeroa kaksi perässä, ovat päivämääriä ilman vuotta. Pilkun takana on sarakkeen nimi, ja pilkun jälkeen on sarakkeen teksti.

```
dataGridView1.Columns.Add(maanantai2, "Maanantai " + maanantai);
dataGridView1.Columns.Add(tiistai2, "Tiistai " + tiistai);
dataGridView1.Columns.Add(keskiviikko2, "Keskiviikko " + keskiviikko);
dataGridView1.Columns.Add(torstai2, "Torstai " + torstai);
dataGridView1.Columns.Add(perjantai2, "Perjantai " + perjantai);
dataGridView1.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
dataGridView1.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
dataGridView1.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
dataGridView1.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
dataGridView1.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
```

Kuva 22. Sarakkeisiin päivät

Kuvassa 23 haetaan jokaiseen päivään varaukset, joissa tuotteiden haku aika on sinä päivänä. Tietokannan alkuajassa on myös kellonaika, joten pitää käyttää where like –lausetta.

```
foreach (DateTime item in viikonpaivat)
{
    DateTime paiva = Convert.ToDateTime(item, CultureInfo.CurrentCulture);
    string p = paiva.Date.Day.ToString() +
        "." + paiva.Date.Month.ToString() +
        "." + paiva.Date.Year.ToString();

    OleDbDataReader drAlut;
    OleDbCommand HaeAlut = new OleDbCommand(
        "SELECT Varaukset.Projekti, Varaukset.Alku, " +
        "Asiakkaat.Nimi, Varaukset.Lisatiedot " +
        "FROM Varaukset, Asiakkaat " +
        "WHERE Varaukset.Alku LIKE '" + p + "%' " +
        "AND Varaukset.AsiakkaanID = Asiakkaat.ID", yhteys);

    if ((yhteys.State != ConnectionState.Open))
    {
        yhteys.Open();
    }

    drAlut = HaeAlut.ExecuteReader();
```

Kuva 23. Haetaan haku- ja palautusajat

Kuvassa 24 tarkistetaan taulukon jokaisen rivin otsikosta, esiintyykö otsikon nimessä hakuajan tunti. Sitten syötetään hakemisen teksti sille rivin sarakkeelle, jonka nimenä on varauksen päivä. Tekstiin tehdään rivin vaihto, jos samalla tunnilla on enemmän, kuin kaksi aikaa.

```

while (drAlut.Read())
{
    DateTime alku = DateTime.Parse(drAlut[1].ToString());
    string a1 = alku.Hour.ToString();
    string a2 = alku.AddHours(1).Hour.ToString();
    string alkupaiva = "(" + alku.Date.Day.ToString() +
        "." + alku.Date.Month.ToString() +
        "." + alku.Date.Year.ToString() + ")";

    foreach (DataGridViewRow rivi in dataGridView1.Rows)
    {
        if (rivi.HeaderCell.Value.ToString() == a1 + " - " + a2)
        {
            if (rivi.Cells[alkupaiva].Value != null)
            {
                rivi.Cells[alkupaiva].Value += Environment.NewLine;
            }

            rivi.Cells[alkupaiva].Value += "Klo " + alku.Hour.ToString() +
                "." + alku.Minute.ToString() + " " + drAlut[2].ToString() +
                " hakee projektin " + drAlut[0].ToString() +
                " kamat. " + drAlut[3].ToString();
        }
    }
}

drAlut = null;
HaeAlut = null;
yhteys.Close();

```

Kuva 24. Syötetään haku- ja palautusajat taulukkoon

Varauksien tuotteiden palautusajat koodataan saman foreach-lauseen sisälle kuin aloitusajat, ja ne tehdään samalla lailla. Kuvassa 25 järjestetään kellonaikojen mukaan tunneissa olevat ajat. Taulukkoon laitetaan myös automaattinen skaalaus.

```

foreach (DataGridViewRow rivi in dataGridView1.Rows)
{
    for (int i = 0; i < rivi.Cells.Count; i++)
    {
        if (rivi.Cells[i].Value != null &&
            rivi.Cells[i].Value.ToString().Contains(Environment.NewLine))
        {
            string[] lauseet =
                rivi.Cells[i].Value.ToString().Split(
                    new[] { Environment.NewLine },
                    StringSplitOptions.None);

            var arr = lauseet.OrderBy(x => x).ToArray();

            rivi.Cells[i].Value = null;

            foreach (var item in arr)
            {
                if (rivi.Cells[i].Value != null)
                {
                    rivi.Cells[i].Value +=
                        Environment.NewLine + Environment.NewLine;
                }

                rivi.Cells[i].Value += item;
            }
        }
    }
}

dataGridView1.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;
dataGridView1.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCellsExceptHeaders;
dataGridView1.ClearSelection();
}

```

Kuva 25. Järjestetään kellonajan mukaan

6.1 Tuotteet-ikkuna

Form2

Etsi tuote Näytä kaikkien kategorioiden tuotteet Kategoria

ID	Nimi	Kategoria	Huomioitavaa
32	Blackmagic Ursa Mini	Kamerat	
33	Blackmagic Ursa Mini	Kamerat	
35	Canon 5D M3	Kamerat	
36	DJI Phantom 4 (Drone)	Kamerat	
37	GoPro Fusion 360	Kamerat	
38	GoPro Hero 4	Kamerat	

Tyhjennä valinta Kuinka monta uutta samannimistä tuotetta lisätään Muokkaa kaikkia samannimisiä tuotteita tai poista kaikki samannimiset tuotteet Poista valittu tuote

Tuotteen nimi

Tuotteen kategoria

Oheislaitteet

Huomioitavaa

Lisää uusi/Tallenna muutokset

Varaukset, joille valittu tuote on varattu ja kellä on lainassa

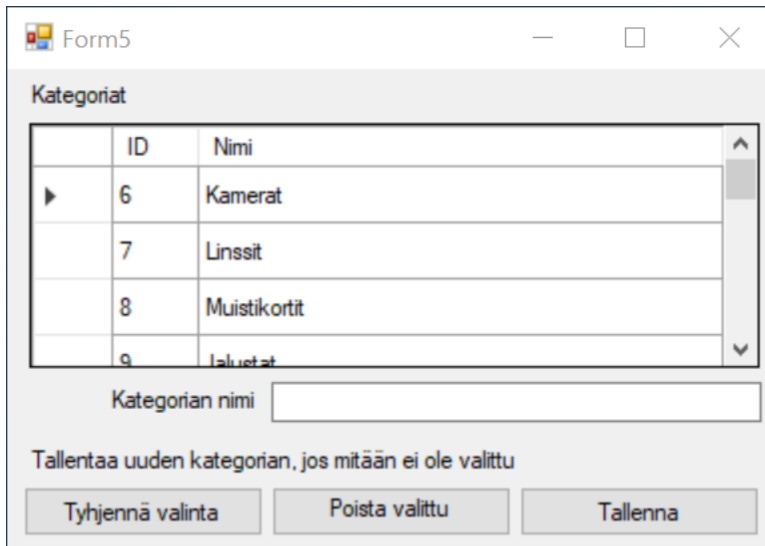
ID	Asiakas	Projekti	Alku	Loppu	Lainassa
102	asiakas2	projekti2	25.4.2019 9.00.00	25.4.2019 12.00.00	<input type="checkbox"/>
103	asiakas1	tapahtuma1	29.4.2019 10.00.00	29.4.2019 12.30.00	<input type="checkbox"/>

Kuva 26. Varasto-ohjelman Tuotteet-ikkuna

Varasto-ohjelman Tuotteet-ikkunassa Tuotteen nimi-, Oheislaitteet- ja Huomioitavaa-tekstikenttien tiedoilla voidaan lisätä tai muokata tuotteita. Tuotteelle voidaan valita alasvetovalikosta kategoria. Tuotteet-taulukon yläpuolella on toinen kategorian alasvetovalikko ja taulukossa näkyvät valitun kategorian tuotteet. Kaikkien kategorioiden tuotteet saadaan näkyviin vieressä olevasta valintaruudusta. Ylhäällä olevaan tekstikenttään voidaan kirjoittaa hakusana, ja Tuotteet-taulukko näyttää vain niitä tuotteita, joiden nimessä esiintyy haun kirjain-yhdistelmä. Kun klikataan taulukon rivin mitä tahansa solua, sen rivin Tuotteen nimi-, Tuotteen kategoria-, Oheislaitteet- ja Huomioitavaa-tiedot menevät alhaalla oleviin kenttiin. Valitun tuotteen varaajat ja lainaaja näkyvät toisessa taulukossa. Kun taulukossa ei ole valittuna mitään tuotetta, Tallenna-painikkeella voidaan tallentaa uusi tuote. Jos tuote on valittuna, samalla Tallenna-painikkeella voidaan tallentaa tuotteen muutokset. Kun klikataan Tyhjennä valinta -painiketta, taulukossa ei ole valittuna mitään tuotetta ja tekstikentät tyhjäntyvät. Kategorian vaihdon, tuotteen tallennuksen tai tuotteen poiston jälkeen taulukko latautuu uudestaan, taulukossa tuotteen valinta tyhjenee ja

tekstikentät tyhjäntyvät. Kummankin kategorian valinnan-alasvetovalikkoon jää valituksi aiemmin valittu kategoria. Näin estetään, ettei jokaisen lisäyksen tai muutoksen jälkeen jää valituksi aina ylin kategoria. Jos halutaan lisätä monta samannimistä tuotetta, voidaan syöttää tuotteiden lukumäärä tekstikenttään. Tähän kenttään vaihtuu jokaisen tallennuksen jälkeen takaisin numero yksi. Tämän lisäksi on valintaruutu, joka mahdollistaa kaikkien samannimisten tuotteiden tietojen muokkaamisen tai kaikkien samannimisten tuotteiden poiston.

6.2 Kategoriat-ikkuna



ID	Nimi
6	Kamerat
7	Linssit
8	Muistikortit
9	Jalustat

Kategorian nimi

Tallentaa uuden kategorian, jos mitään ei ole valittu

Kuva 27. Varasto-ohjelman kategoriat-ikkuna

Kategoriat-ikkuna on yksinkertainen, koska siinä on vain taulukko kategorioille, tekstikenttä kategorian nimelle ja painikkeet tallentamiselle, poistolle ja valinnan tyhjentämiseksi.

6.3 Asiakkaat-ikkuna

Form3

Etsi opiskelija

ID	Tunnus	Nimi	Sähköposti	Puhelin
▶ 19	igfkfgkh	asiakas 1	a1	123
20	gafawf	asiakas2	a2	321
18	t123	testiqwerty	asd	dsa

Tyhjennä valinta

Opiskelijatunnus

Nimi

Sähköposti

Puhelinnumero

Poista valittu opiskelija

Opiskelijan varaukset

ID	Projekti	Alku	Loppu	Haettu
▶ 101	projekti1	24.4.2019 10.00	25.4.2019 14.00	<input checked="" type="checkbox"/>
103	tapahtuma1	29.4.2019 10.00	29.4.2019 12.30	<input type="checkbox"/>

Varauksen tuotteet

ID	Nimi	Kategoria	Huomioitavaa
▶ 31	Blackmagic Ursa Mini	Kamerat	
116	Dyna DM-150S	Akut	

Tuotteen mukana kulkevat oheislaitteet

kahva
laukku
lintulauta
luuppi
olkatuki
virtajohto

Kuva 28. Varasto-ohjelman asiakkaat-ikkuna

Asiakkaat-ikkunassa hakukenttä toimii samalla lailla kuin Tuotteet-ikkunassa. Asiakkaan lisäys, muokkaus ja poisto tapahtuvat samalla lailla kuin Tuotteet-ikkunassa. Valitun asiakkaan varaukset näkyvät toisessa taulukossa, jossa näkyy varauksen alku- ja loppupäivämäärät sekä kellonajat. Valitun varauksen tuotteet ovat kolmannessa taulukossa. Valitun tuotteen oheislaitteet näkyvät isossa tekstikentässä.

6.4 Varaukset-ikkuna

Tuotteita ei ole palautettu sovittuun aikaan Tuotteita ei ole haettu sovittuun aikaan Etsi varaus projektin nimellä

ID	Asiakkaan nimi	Projekti	Varaus tehty	Varauksen alku	Varauksen loppu	Haettu	Lisätiedot	Kysymys	Vastaus
101	asiakas1	projekti1	17.4.2019 8.19	24.4.2019 10.00	25.4.2019 14.00	<input checked="" type="checkbox"/>	Muista muistaa	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
102	asiakas2	projekti2	17.4.2019 8.21	25.4.2019 9.00	25.4.2019 12.00	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
103	asiakas1	tapahtuma1	17.4.2019 9.06	29.4.2019 10.00	29.4.2019 12.30	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>

Tuotteiden hakemispäivä ja kellonaika: keskiviikko 24. huhtikuuta 2019 10.00.00

Tuotteiden palautuspäivä ja kellonaika: torstai 25. huhtikuuta 2019 14.00.00

Asiakas: asiakas1 ID: 19 Tunnus: igfkgkh Sähköposti: a1 Puhelin: 123

Kysymys lainauksesta: Kirjoita vastaus kysymykseen: Oma muistilappu varaukseen: Muista muistaa

Projektin nimi: projekti1 Tuotteet haettu Tallenna muutokset

Valitut varauksen tuotteet

ID	Tuotteet.Nimi	Kategoriat.Nimi	Huomioitavaa
31	Blackmagic Urna Mini	Kamerat	
116	Dyna DM-150S	Akut	

Kuittaa valittu tuote palautetuksi Poistaa valitun tuotteen varauksesta

Poista koko varaus Poistaa myös kaikki tuotteet varauksesta

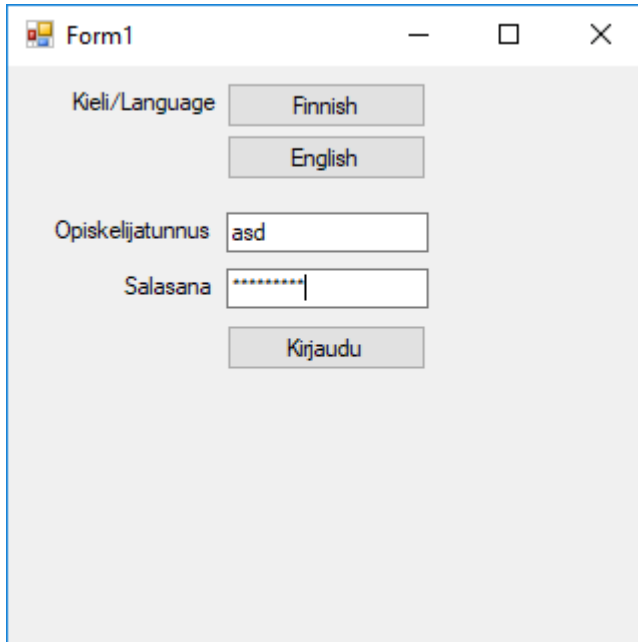
Tuotteen mukana kulkevat ohjeistukset: kahva, laukku, lintulauta, luuppi, olkatuki, virtajohto

Luo PDF

Kuva 29. Varasto-ohjelman varaukset-ikkuna

Varaukset-ikkunassa varauksen vastuuhenkilöä voidaan vaihtaa alasvetovalikosta. Varauksen aikoja ei voida muokata. Varausta klikkaamalla tulevat varauksen tiedot näkyviin alhaalla oleviin kenttiin. Valitusta varauksesta voi tallentaa pdf-tiedoston, jossa lukee varauksen ja asiakkaan tiedot sekä varauksen tuotelista. Varastonhoitaja voi kirjoittaa isolle tekstikentälle oman muistilapun valittuun varaukseen. Muistilappu näkyy myös ohjelman aloitusikkunan viikkokalenterissa varauksen perässä. Varaustaulukossa näkee kyllä/ei-kentistä, onko varauksella kysymys ja siihen vastaus. Kysymys näkyy isossa tekstikentässä ja vastauksen voi kirjoittaa toiseen isoon tekstikenttään. Varauksen rivi näkyy punaisella, jos sovittu palautusajan kellonaika on ylittynyt. Varauksen rivi näkyy violettina, jos sovittu hakemisaika ylittyy. Kun tavarat on haettu, voidaan valintaruudulla päivittää tietokantaan, että tavarat ovat tällä hetkellä sillä varauksella lainassa. Kun kaikki tuotteet on palautettu, koko varaus poistetaan. Tuotteita voidaan myös poistaa varauksen tuotelistasta yksitellen, jos osa tuotteista palautetaan etuajassa.

7 VARAUSOHJELMAN TULOS



The image shows a window titled "Form1" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a login form with the following elements:

- A label "Kieli/Language" followed by two buttons: "Finnish" and "English".
- A label "Opiskelijatunnus" followed by a text input field containing the text "asd".
- A label "Salasana" followed by a password input field containing ten asterisks "*****".
- A button labeled "Kirjaudu" (Login) positioned below the password field.

Kuva 30. Varausohjelman aloitusikkuna

Varausohjelman käynnistyessä aukeaa Kirjautumis-ikkuna, jossa voidaan vaihtaa kieltä suomen ja englannin välillä, mutta kielenvaihtoa ei ole vielä koodattu. Tunnuksen ja salasanan syöttämisen jälkeen ohjelma tarkistaa tietokannasta, ovatko tunnus ja salasana oikein. Sovellus näyttää virheilmoituksen, jos tunnus tai salasana on väärin. Jos syötetty tunnus on oikein, aukeaa toinen ikkuna, jossa on kaksi väli-
lehteä

7.1 Ensimmäinen välilehti

Form2

Henkilötiedot Uusi varaus

Tarkista tietosi ennenkuin jatkat varauksen tekoon

Varauksesi (varauksien maksimimäärä on 3)

Varaukset, joiden tuotteita ei ole haettu määräajassa, näkyvät violettilla
Varaukset, joiden tuotteita ei ole palautettu määräajassa, näkyvät punaisella

Luo pdf varauksesta

Varauksen aloituspäivämäärä ja kellonaika
keskiviikko 1. toukokuuta 2019 9.45.00

Varauksen palautuspäivämäärä ja kellonaika
perjantai 3. toukokuuta 2019 14.00.00

Vain yksi tuotelista muistissa. Tallentaa aina päälle

Tallenna tuotelista

Tuotteen mukana kulkevat oheislaitteet
akkulevy
Lupolaukku
virtajohto

Uusi varaus

Henkilötiedot

ID: 19

Tunnus: igfkfgh

Nimi: asiakas 1

Sähköposti: a1

Puhelin: 123

Päivitä henkilötiedot

ID	Projekti
101	projekti1
103	tapahtuma1
104	tapahtuma2

Varauksen tuotteet

ID	Nimi	Kategoria	Huomioitavaa
32	Blackmagic Ursa Mini	Kamerat	
117	Dyna DM-150S	Akut	
104	Easyrig	Gimballit ja rigit	
61	Lexar 256GB	Muistikortit	
88	Lupo Light	Valot	
197	Sennheiser 25-1 II 70 ohm	Kuulokkeet	

Kysymykseesi

Vastaus kysymykseesi

Kuva 31. Varausohjelman ensimmäinen välilehti

Ensimmäisellä välilehdellä pyydetään käyttäjää tarkistamaan, ovatko oma nimi, sähköposti ja puhelinnumero oikein. Muutokset voidaan tallentaa tietokantaan. Käyttäjä näkee omat olemassa olevat varauksensa taulukossa. Valitusta varauksesta voidaan painiketta klikkaamalla tallentaa pdf-tiedoston, kuten varasto-ohjelmassa. Varauksen riviä klikkaamalla tulevat varauksen tuotteet näkyviin toiseen taulukkoon. Toisen taulukon valitun tuotteen oheislaitteet näkyvät vieressä olevassa listassa. Käyttäjä voi tallentaa valitun varauksen tuotelistan muistiin, jos haluaa nopeasti tehdä uuden samanlaisen varauksen. Datetimepicker-komponentteja on neljä kappaletta, joista kaksi näyttää kellonajan. Alhaalla on vielä kaksi isoa tekstikenttää, joissa lukevat varauksen kysymys ja vastaus. Kummankin välilehden voi vaihtaa ylhäällä olevasta välilehden valitsimesta tai alhaalla olevasta painikkeesta.

7.2 Toinen välilehti

Henkilötiedot Uusi varaus

Checkkaa haluamasi tuotteet. Tuotteen tekstiä klikatessa näkyvät sen varausajat tummennettuna kalenterissa

Akut

- Dyna DM-150S
- Dyna DM-150S
- Dyna DM-150S
- Dyna DM-155S
- Dyna DM-155S
- Dyna DM-155S
- Endura 50
- Endura 50
- Endura 80
- Endura 80
- Endura Elite
- Endura Elite

Gimbalit ja rigit

- Easyrig
- Monopod
- Monopod
- Ronin
- Wooden Camera

Jalustat

Johdot

Kamerat

- Blackmagic Urusa
- Blackmagic Urusa
- Blackmagic Urusa Mini
- Blackmagic Urusa Mini
- Blackmagic Urusa Mini

Valitse aloituspäivämäärä ja kellonaika
keskiviikko 24. huhtikuuta 2019 09.36

Valitse palautuspäivämäärä ja kellonaika
keskiviikko 24. huhtikuuta 2019

Projektin nimi*

Kysymys

Hae tallennettu tuotelistasta Tallenna tuotelistasta

Valitut tuotteet

ID	Nimi	Kategoria	Huomioitavaa
116	Dyna DM-150S	Akut	
104	Easyrig	Gimbalit ja rigit	
31	Blackmagic Urusa Mini	Kamerat	
224	Lexar CFast	Kortinlukijat	
192	Beyerdynamic DT 770 Pro 80 ohm	Kuulokkeet	
68	Lexar 128GB	Muistikortit	
65	Lexar 256GB	Muistikortit	

Valitun tuotteen mukana kulkevat oheislaitteet

Takaisin Tallenna varaus

Kuva 32. Varausohjelman toinen välilehti

Toisessa välilehdessä on puunäkymä, jossa tuotteet ovat kategorioittain. Jos puussa klikataan tuotteen tekstiä tai valitaan tuote taulukosta, kalenterissa näkyy tummennettuina numeroina tuotteen varauspäivät. Tuotteita valitaan klikkaamalla nimien takana olevia valintaruutuja. Oikealla olevassa tuotetaulukossa näkyvät valitut tuotteet. Kuvassa 33 on metodi kategorioiden lisäyksestä puuhun. Kun tätä metodia käsketään, se lisää myös tuotteet puuhun, koska tässä metodissa kutsutaan tuotteiden täytön -metodia.

```

void KategoriatTaydenna()
{
    OleDbDataReader puuKategoria;
    OleDbCommand ValitseKategoriat = new OleDbCommand(
        "SELECT ID, Nimi FROM Kategoriat", yhteys);

    if ((yhteys.State != ConnectionState.Open))
    {
        yhteys.Open();
    }

    treeView1.Nodes.Clear();
    puuKategoria = ValitseKategoriat.ExecuteReader();
    while (puuKategoria.Read())
    {
        TreeNode nKategoria = new TreeNode();
        nKategoria.Text = puuKategoria["Nimi"].ToString();
        nKategoria.Tag = puuKategoria["ID"].ToString();
        this.treeView1.Nodes.Add(nKategoria);
        KategoriatLataaTuotteet(nKategoria);
        nKategoria = null;
    }

    puuKategoria = null;
    ValitseKategoriat = null;
    yhteys.Close();

    treeView1.Sort();

    dataGridView1.DataSource = null;
    dt.Clear();
    dataGridView1.Rows.Clear();
}

```

Kuva 33. Kategoriat puuhun

Kuvassa 34 on metodi tuotteiden lisäyksestä kategoriaksiin. Tuotteen perässä lu-
kee myös, jos tuotteessa on huomioitavaa.

```

void KategoriatLataaTuotteet(TreeNode nVanhempi)
{
    string arrVanhempiTyyppi;
    arrVanhempiTyyppi = nVanhempi.Tag.ToString();

    OleDbDataReader drTuotteet;
    OleDbCommand HaeTuotteet = new OleDbCommand(
        "SELECT ID, Nimi, Katgoria, " +
        "Huomioitavaa FROM Tuotteet WHERE Katgoria="
        + arrVanhempiTyyppi + "", yhteys);

    drTuotteet = HaeTuotteet.ExecuteReader();

    while (drTuotteet.Read())
    {
        string tuoteTeksti = "";
        if (drTuotteet[3].ToString() != "")
        {
            tuoteTeksti = drTuotteet[1].ToString()
                + drTuotteet[3].ToString();
        }

        if (drTuotteet[3].ToString() == "")
        {
            tuoteTeksti = drTuotteet[1].ToString();
        }

        string tuoteTagi = drTuotteet[0].ToString();
        TreeNode nTuote = new TreeNode();
        nTuote.Text = tuoteTeksti;
        nTuote.Tag = tuoteTagi;
        nVanhempi.Nodes.Add(nTuote);
        nTuote = null;
    }

    drTuotteet = null;
    HaeTuotteet = null;
    arrVanhempiTyyppi = null;
}

```

Kuva 34. Tuotteet puuhun

Tarvitaan Checked-bool, jotta voidaan tunnistaa, jos puussa on valittu jokin tuote.

```

public bool Checked { get; set; }

```

Kuva 35. Checked-bool

Kuvassa 36 on metodi, joka lisää puun valittujen tuotteiden ID-numerot listaan

```

public void HaeValitutTuotteet(TreeNodeCollection tuotteet)
{
    foreach (System.Windows.Forms.TreeNode tuote in tuotteet)
    {
        if (tuote.Checked)
        {
            if (tuotteetLista.Contains(Int32.Parse(tuote.Tag.ToString())) == false)
            {
                tuotteetLista.Add(Int32.Parse(tuote.Tag.ToString()));
            }
        }

        HaeValitutTuotteet(tuote.Nodes);
    }
}

```

Kuva 36. Puun tuotteet listaan

Joka kerta, kun klikataan tuotteen valintaruutua, kutsutaan metodia, joka lisää ID-numerot listaan.

```

private void treeView1_AfterCheck(object sender, TreeViewEventArgs e)
{
    tuotteetLista.Clear();
    HaeValitutTuotteet(treeView1.Nodes);
}

```

Kuva 37. Kutsutaan metodia

Taulukossa päivittyvät tuotteet samaan aikaan, kuin tuotelista. Hae tallennettu tuotelista-painiketta klikkaamalla opiskelija voi hakea oman tallennetun tuotelistan. Tuotteen oheislaitteet näkyvät samalla lailla kuin ensimmäisellä sivulla. Varaukselle on pakko laittaa projektinimi tekstikenttään. Jos on kysymyksiä, isoon tekstikenttään voi kirjoittaa kysymyksen. Datetimepicker-komponenteista voi valita varauksen aloitus- ja lopetuspäivät. Varauksen aloituspäivä voi olla minimissään viikon päästä. Kaksi Datetimepicker-komponenttia on muokattu kellonajan valitsimiksi, niistä voi valita ajat klo 9 ja 16 väliltä minuutin tarkkuudella.

```
dateTimePicker2.Format = DateTimePickerFormat.Custom;
dateTimePicker2.CustomFormat = "HH.mm";
dateTimePicker2.ShowUpDown = true;
dateTimePicker2.MinDate = DateTime.Parse("9:00:00");
dateTimePicker2.MaxDate= DateTime.Parse("16:00:00");
dateTimePicker4.Format = DateTimePickerFormat.Custom;
dateTimePicker4.CustomFormat = "HH.mm";
dateTimePicker4.ShowUpDown = true;
dateTimePicker4.MinDate = DateTime.Parse("9:00:00");
dateTimePicker4.MaxDate = DateTime.Parse("16:00:00");
```

Kuva 38. Kellonajan valitsin

Kun klikataan tallenna varaus –painiketta, ohjelma tarkistaa ennen tallennusta, jos joku tuote on juuri ehditty varata samalle ajalle. Jos on, tulee ilmoitus, jossa kerrotaan, mitkä tuotteet on varattu, ja pyydetään valitsemaan toinen aika tai toiset tuotteet. Onnistuneen varauksen jälkeen tulee ilmoitus, että varaus on tallennettu. Jos haluaa tehdä monia päivän mittaisia varauksia samoille tuotteille, sekin on mahdollista, koska kaikki valitut tuotteet jäävät valituksi tallennuksen jälkeen. Tälläkin välilehdellä voidaan tallentaa tuotelista muistiin, jos varauksen teko pitää keskeyttää. Jos opiskelijalla on jo kolme varausta, ohjelma estää uuden varauksen tekemisen. Ohjelma estää myös uuden varauksen tekemisen, jos varauksen pituus on enemmän kuin viikko. Viikonloppuja ei voi valita haku- ja palautusajoiksi. Kun varaus on tehty, ensimmäiselle sivulle päivittyy opiskelijan oma varauslista.

8 YHTEENVETO JA POHDINTA

Ohjelmien kehittäminen jatkuu opinnäytetyön jälkeen työharjoittelussa tämän työn tekijällä. Lopuksi pitää tehdä kunnon dokumentit sovelluksista ja lisätä selkeät kommentit sovelluksien koodeihin, jotta tiedetään, missä ollaan menossa, jos muu henkilö tulee myöhemmin jatkamaan tehtyjen sovelluksien kehittämistä.

8.1 Tulos opinnäytetyön aikana

Tämän opinnäytetyön kirjoittamisen aikana saatiin tehtyä valmiiksi varastohallinta-ohjelma ja varausohjelma Windows-työpöytä-sovelluksina. Ohjelmien ikkunoihin ei ole vielä suunniteltu sopivan näköistä käyttöliittymästandardia. Ikkunat eivät myöskään ole vielä responsiivisia. Verkkosivun ja Android-sovelluksen tekoa ehdittiin hie- man testata. Tällöin selvisi, että Microsoft Edgella, Google Chromella ja Mozilla Firefoxilla ei pysty enään muodostamaan yhteyttä Access Databaseen, joten tietokanta on pakko muuttaa SQL-muotoon. Androidillakaan ei pysty yhdistämään Access Databaseen. Internet Explorerilla toimii vielä yhteys Access Databaseen, mutta selainta ei kannata enään käyttää, koska sitä ei enään päivitetä.

8.2 Varausohjelman kehittäminen opinnäytetyön jälkeen

Seuraava vaihe on tietokannan laittaminen koulun SQL-palvelimelle. Ohjelmiin pitää toteuttaa yhteydet SQL-tietokantaan. Kun ohjelmat on todettu toimivaksi uuden tietokannan kanssa, aloitetaan varausverkkosivun tekeminen. Verkkosivulla asiakasta pyydetään tekemään tunnus varaston tietokantaan, vaikka käytössä olisikin SeAMKin kirjautumisikkuna. Verkkosivu voisi hyväksyä vain sellaisten tunnuksien luonnit, joissa sähköposti loppuu ”seamk.fi”-muotoon. Syötettyyn sähköpostiin lähetetään varmistusviesti, josta varmistetaan sähköposti, ennen kuin voi kirjautua tunnukselle. Opiskelijaa pyydetään tekemään kuvaussuunnitelma, ennen kuin varaa kalustoa. Opiskelija voi ladata kuvaussuunnitelmapohjan sivulta. Suunnitelman voi laittaa sitten liitteeksi, varausta ei pystyisi tekemään ilman dokumenttia. Verkkosivu voisi lähettää dokumentin automaattisesti palvelimelle.

LÄHTEET

- Bulleen, V. 2012. Operate Database Applications : ICAICT210A Microsoft Access 2010: Unit Workbook. [Verkkokirja]. Watsonia Publishing. [Viitattu 10.4.2019]. Saatavana Ebsco eBook Collection -tietokannasta. Vaatii käyttöoikeuden.
- Cherry, D. 2015. Securing SQL Server : Protecting Your Database from Attackers. [Verkkokirja]. William Andrew. [Viitattu 26.3.2019]. Saatavana ProQuest Ebook Central -tietokannasta. Vaatii käyttöoikeuden.
- IAN. 5.6.2016. Microsoft Access vs SQL Server. [Verkkosivu]. Database.Guide. [Viitattu 3.4.2019]. Saatavana: <https://database.guide/microsoft-access-vs-sql-server/>
- Kennedy, J. & Satran, M. 2018. UX checklist for desktop applications. [Verkkoartikkeli]. Microsoft Docs. [Viitattu 18.4.2019]. Saatavana: <https://docs.microsoft.com/fi-fi/windows/desktop/uxguide/top-violations>
- Samantaray, S. 2017. Generate JSON Data Using Web Service And SQL Server Stored Procedure. [Verkkoartikkeli]. C# Corner. [Viitattu 11.4.2019]. Saatavana: <https://www.c-sharpcorner.com/article/generate-json-data-using-web-service-and-sql-server-stored-procedure/>
- SeAMK. Ei päiväystä. Kulttuurituottaja (AMK). [Verkkosivu]. Seinäjoen Ammattikorkeakoulu. [Viitattu 16.3.2019]. Saatavana: <https://www.seamk.fi/kaikki-koulutukset/kulttuurituottaja-amk/>
- Strauss, D. 7.2016. C# Programming Cookbook. [Verkkokirja]. Packt Publishing Ltd. [Viitattu 9.4.2019]. Saatavana Ebsco eBook Collection -tietokannasta. Vaatii käyttöoikeuden.
- Thornsby, J. 5.2016. Android UI Design. [Verkkokirja]. Packt Publishing Ltd. [Viitattu 9.4.2019]. Saatavana Ebsco eBook Collection -tietokannasta. Vaatii käyttöoikeuden.