|  |
| --- |
| Team Azure |
| Toteutussuunnitelma |
| TosiSopiva-laskutusohjelma |

|  |
| --- |
| Aarni Moisio |
| Andrei Murashkin |
| Jussi Rantapere |

# JOHDANTO

Tämä dokumentti käsittelee TosiSopiva- laskutusohjelman toteutusta, testausta ja jatkokehityssuunnitelmaa.

Team Azuren tarkoitus on kehittää laskutusohjelma, joka kykenee suoriutumaan pienyrityksen laskutusmyynnin työkaluna automaattisesti poistaen tarpeen laskujen manuaaliselta kirjoittamiselta ja manuaaliselta reskontran seuraamiselta.

Ohjelma tarvitsee käyttöliittymän, joka toteutetaan python ohjelmointikielellä. Python ohjelmointikieli tarjoaa eri vaihtoehtoja käyttöliittymän toteuttamiseen, kuten tkinter ja pyqt. Näiden tekniikoiden välillä ei ole vielä tehty valintaa. Valinta tehdään, kun tekniikoita on ehditty paremmin arvioida. Käyttöliittymän täytyy olla käyttäjän kannalta selkeä ja niin yksinkertainen, että ohjelman käyttöönotto sujuu nopeasti ja pienellä vaivannäöllä.

Koska tällä ohjelmointikurssilla on opiskeltu myös C- kieltä, tehdään tietokannan ohjaus C- kielellä, vaikka se voitaisiin yhtä hyvin tehdä myös python kielellä. Sekä pdbc ja odbc ovat toimivia vaihtoehtoja. Lisäksi kannan ohjaus voitaisiin tehdä myös toteuttamalla tietokantaoperaatiot hoitava backend ohjelma pilvipalveluun. Tässä tapauksessa C- kieli ei kuitenkaan olisi järkevä vaihtoehto, vaan silloin esimerkiksi C# tai Nodejs olisivat ratkaisuna parempia, koska backend ohjelman käyttöliittymä olisi internetissä, jolloin autentikointi ja authorisointi olisi pakko toteuttaa käyttöliittymän suojaksi.

Python- ja C- kielen yhteensovittaminen vaatii C- koodin kääntämistä dll- kirjastoksi, jonka funktioita kutsutaan python ohjelmasta ctypes python kirjaston avulla. C- koodin muistinhallinnan suhteen täytyy noudattaa erityistä huolellisuutta, jotta kaikki dynaamisesti allokoitu muisti vastaavasti vapautetaan. Muistinhallinnan avuksi löytyy työkaluja, kuten Visual Studio 2022:n Diagnostic tools, joka sisältää Memory usage näkymän.

## Työnjako

Aarni Moisio kehittää python koodia ja jonkin verran C- koodia, Andrei Murashkin ja Jussi Rantapere kehittävät sekä C- että python koodia.

Työnjako tarkentuu vielä myöhemmin.

# TOTEUTUS

## Käyttöliittymä

Ohjelmaan rakennetaan graafinen käyttöliittymä. Toimintaan ohjataan valikoiden kautta. Tietojen syöttämistä varten kehitetään lomakenäkymä, josta ’tallenna’- näppäimellä tallennetaan tiedot tietokantaan.

Käyttöliittymällä täytyy myös voida esittää näytöllä aikaisemmin tehtyjä laskuja. Tätä varten tarvitaan hakutoiminto, johon syötetään hakukriteereitä.

Lisäksi toteutetaan yleisempi näyttö, jolla oleellisimmat tiedot voidaan tulostaa ruudulle, kun käyttäjän täytyy löytää tietoja isommasta laskujoukosta, eikä ole täysin selvää, mikä on juuri se etsittävä lasku.

## Tietokanta dll- kirjasto

ODBC ajuria käyttävä C- kielinen dll- kirjasto sisältää pääosin funktioita, jotka kutsuvat tietokannassa olevia ’stored procedure’ kyselyitä, jotka suorittavat varsinaiset SQL kyselyt. Microsoft Azure Sql Database PaaS tuote vaatii Microsoft ODBC ajuria versio 18.

Käytännössä Sql Server voi sijaita muuallakin, kuin pilvipalvelussa, vaikkapa paikallisesti tietokoneella, jossa TosiSopiva laskutusohjelmaa käytetään. Pilvipalvelu mahdollistaa kuitenkin helpon tavan tarjota sama data kaikille käyttäjille.

Tietoa käyttöliittymän ja tietokanta dll- kirjaston välillä vaihdetaan json muodossa. Tämä on hyvä tapa varmistaa, että oikea tieto esitetään oikeassa paikassa.

## Käyttöliittymän ohjaus

Käyttöliittymän tehtävänä on varmistaa, että puutteellista ja/tai virheellistä tietoa ei tallenneta tietokantaan.

A diagram of a flowchart

Description automatically generated

Kuva 1Ohjelma varmistaa syötettyjen tietojen oikeellisuuden

## Tietokanta

Azure SQL Database palvelun käyttöönotto tarjoaa käyttöön tyhjän tietokannan. Tämä dokumentti ei ota kantaa mahdollisiin rajoituksiin, mitä Azure Sql Database palveluun liittyy, kuten esimerkiksi tietokannan maksimikokoon jne. Tässä yhteydessä riittää, kun kuvataan kannassa tarvittavat taulut.

Asiakas- tauluun kerätään asiakkaan henkilö- ja osoite- ja muut yhteystiedot. Laskut- tauluun sijoitetaan kaikki laskun yhteiset tiedot:

* Laskun numero
* Asiakas numero
* Laskun päivämäärä
* Pankin viite
* Laskun kokonaissumma, veroton
* Vero
* Laskun kokonaissumma, verollinen
* Laskun eräpäivä
* Asiakkaan maksama määrä

Laskurivi- tauluun taas tallennetaan laskuihin liittyvät yksittäisten tuotteiden tiedot, kuten tuotteen nimi, ostettu määrä ja yksikköhinta. Mahdollisesti tuotteiden tiedot sijoitetaan kuitenkin tuote- tauluun, jolloin tilausrivi- taulun tiedot muokataan siten, että yksittäiseen tuotteeseen viitataan tuotetunnuksella.

* Laskurivin numero
* Laskun numero
* Kappalemäärä
* Hinta
* Tuotteen kuvaus
* Tuoteyksikön numero

Tietokannan taulujen, asiakas ja lasku sekä lasku ja laskurivi, välillä vallitsee yksi-moneen relaatioita, mikä tarkoittaa sitä, että yksi asiakas yhdistetään aina tiettyyn laskuun ja tietyt laskurivit yhdistetään aina tiettyyn laskuun. Kaikki taulut ja niiden relaatiot luodaan ja ylläpidetään sql- kielellä. Nämä toimenpiteet tehdään suoraan tietokantaan ilman niiden yhdistämistä TosiSopiva- laskutusohjelmaan.

Tietokanta tarvitsee myös muita tauluja, koska laskusta täytyy ilmetä, mitä tuotteita on myyty. Tuotteilla on käytännössä erilaisia variaatioita ja nämä variaatiot on oleellista huomioida laskuissa. Tästä syystä tuotevariaatiot on pystyttävä kuvaamaan tietokannassa. Muuten niiden hallinta manuaalisesti tekisi ohjelman käyttämisestä huomattavasti hankalampaa ja automatisoinnin aste laskisi oleellisesti.

# TESTAUS

Ohjelmiston testaus tehdään yksikkötestauksena. Tarvittavat testikeissit muodostetaan projektin edetessä. Testaamiseen käytetään Visual Studio 2022:n tarjoamia työkaluja.

Testaus rakennetaan siten, että kaikki C- funktiot testataan python ctypes- kirjaston avulla. Jokainen funktio testaan erikseen. Testaus on mahdollista myös automatisoida erilaisten valmiiden saatavilla olevien testausautomatisointityökalujen avulla. Tällöin testit voidaan ajastaa tapahtuvaksi tietyin välein. Näin mahdolliset ohjelmointivirheet löydetään ja pystytään eliminoimaan mahdollisimman aikaisessa vaiheessa.

# JATKOKEHITYS

## Ohjelman käyttäjäkohtaiset käyttöoikeudet ja käyttäjän todentaminen

Ohjelman on alkuvaiheessa tarkoitus olla vain tyypillisen pienyrityksen käyttöön soveltuva työväline, jota käyttää yksi laskutuksesta vastaava henkilö. Tietoturvan varmistamiseksi ohjelmistoon voitaisiin rakentaa kirjautusmistoiminto, jolloin ainoastaan kirjautuneet autentikoidut henkilöt voivat nähdä tiedot ja tehdä niihin muutoksia. Autentikoiduilla käyttäjillä voi olla myös toisistaan poikkeavia oikeuksia nähdä sekä muuttaa tietoja. Tällaisia toimintoja ohjelmaan saa mukaan esimerkiksi Microsoft Azuren tarjoaman Microsoft Entra ID:n avulla. Tällöin tietokantaoperaatioiden suorittaminen pitäisi toteuttaa backend ohjelmaan, jossa käyttäjän henkilöllisyys ja käyttöoikeudet tarkistettaisiin.

Isompien yritysten käyttöön jatkokehitysaihe olisi kehittää mahdollisuus useammille yhtäaikaisille käyttäjille, jotka käyttäisivät tietokantaa toisistaan riippumatta eri työpisteissä.

Tällöin ohjelman tulisi kyetä suoriutumaan tilanteesta, joissa ohjelmaa käyttää laskuja kirjoittava henkilö. Hänellä täytyy olla oikeus kirjautua sisään ohjelmaan ja kirjoittaa esimerkiksi uusia laskuja. Mutta ohjelman on kyettävä myös näyttämään reskontratietoja, koska laskutusreskontra on oleellinen osa laskutuksen seurantaa. Kenties laskuja kirjoittavan henkilön ei tule voida nähdä laskutusreskontraa. Tällöin, vaikka laskuja kirjoittava henkilö kykenee kirjautumaan itse ohjelmaan, ei hän ole oikeutettu pääsemään näkemään reskontratietoja.

Samalla tavalla laskutusohjelmasta voidaan tulostaa tilastotietoja kuukausimyynnistä, tai vaikkapa myynnistä vuositasolla. Ja reskontraa hoitavat tai uusi laskuja kirjoittavat henkilöt eivät ole välttämättä oikeutettuja näkemään tietoja koko laskutusmyynnistä.

## Tilaustietojen yhdistäminen osaksi järjestelmää

Yritys, joka myy tuotteita ennakkoon toisille yrityksille, tarvitsee ennakkotilausten hallintaan oman ennakkotilausrekisterin, joka voidaan integroida TosiSopiva ohjelman tietokantaan. Ennakkotilauksia käsitellään varsinaisesta laskutuksesta erillisenä osana. Tilaukset muodostavat kuitenkin tulevaa laskutusta muodostavan osan ja on samankaltaisesti rakennettava osa järjestelmää.

Tilausten hallinta kertoo yrityksen tulevasta laskutuksesta. Kun tuotteiden toimituksen aika tulee, muodostetaan tilauksesta lasku, kun tavara sovittuna aikana toimitetaan asiakkaalle laskua vastaan. Tilaustiedot ja sitä vastaava laskutus pidetään kuitenkin toisistaan erillisinä, koska edellisvuosien tilaus- ja laskutustiedot sisältävät yritykselle arvokasta dataa. Tällaista dataa voidaan käyttää liiketoiminnan ylläpitoon ja kehittämiseen, joten tiedon täytyy tarvittaessa olla saatavilla sopivassa muodossa.

## Tietokannan rakenteen kehittäminen

Yrityksen täytyy seurata laskutusreskontraa. Avointen erääntyneiden laskujen poimiminen täytyy olla helposti toteutettu järjestelmään. Tällä hetkellä tietokannassa on laskurivi taulussa sarake, johon asiakkaan jo maksama osa laskusta summataan. Tällainen ratkaisu on vielä puutteellinen, koska asiakas saattaa maksaa laskua pienemmissä osissa ja tällöin maksettu summa koostuu eristä, jotka on maksettu tiettyinä päivinä. Tämä tämänhetkinen puute voidaan ratkaista lisäämällä tietokantaan taulu, joka kerää tiedot päivämääristä, maksetusta summasta ja vaikkapa pankista, jonka kautta maksu on tullut. Ideaalinen ratkaisu kuitenkin olisi se, että tietoja ei tarvitsisi syöttää ohjelmaan käsin, vaan tiedot saataisiin suoraan pankin järjestelmästä.