**Vaatimusmäärittelydokumentti**

Laskutusohjelma

HAMK, Ohjelmistokehitys

Lehtinen Jarno, Pusa Jaana

Sisällys

[1. Johdanto 4](#_Toc129340739)

[1.1 Tarkoitus ja kattavuus 4](#_Toc129340740)

[1.2 Tuote ja ympäristö 4](#_Toc129340741)

[1.3 Määritelmät, termit ja lyhenteet 4](#_Toc129340742)

[1.4 Viitteet 4](#_Toc129340743)

[1.5 Yleiskatsaus dokumenttiin 4](#_Toc129340744)

[2. Yleiskuvaus 5](#_Toc129340745)

[2.1 Ympäristö 5](#_Toc129340746)

[2.2 Toiminta 5](#_Toc129340747)

[2.4 Yleiset rajoitteet 5](#_Toc129340748)

[2.5 Oletukset ja riippuvuudet 5](#_Toc129340749)

[3. Tiedot ja tietokanta 5](#_Toc129340750)

[3.1 Tietosisältö 5](#_Toc129340751)

[3.2 Käyttöintensiteetti 6](#_Toc129340752)

[3.3 Kapasiteettivaatimukset 6](#_Toc129340753)

[3.4 Tiedostot ja asetustiedostot 6](#_Toc129340754)

[4. Toiminnot 6](#_Toc129340755)

[4.1 Asiakkaan tietojen syöttö sovellukseen 6](#_Toc129340756)

[4.2 Palvelutason tietojen muokkaus 6](#_Toc129340757)

[4.3 Maksulista 6](#_Toc129340758)

[4.4 Alennustietojen muokkaus 6](#_Toc129340759)

[4.5 Kokeilujakso tilaajille 6](#_Toc129340760)

[5. Ulkoiset liittymät 7](#_Toc129340761)

[5.1 Laitteistoliittymät 7](#_Toc129340762)

[5.2 Ohjelmistoliittymät 7](#_Toc129340763)

[5.3 Tietoliikenneliittymät 7](#_Toc129340764)

[6. Muut ominaisuudet 6.1 Suorituskyky ja vasteajat 7](#_Toc129340765)

[6.2 Saavutettavuus (availability), toipuminen, turvallisuus, suojaukset 7](#_Toc129340766)

[6.3 Ylläpidettävyys 7](#_Toc129340767)

[6.4 Siirrettävyys ja yhteensopivuus 7](#_Toc129340768)

[7. Suunnittelurajoitteet 8](#_Toc129340769)

[8. Hylätyt ratkaisuvaihtoehdot 8](#_Toc129340770)

[9. Jatkokehitysajatuksia 8](#_Toc129340771)

[Liitteet 8](#_Toc129340772)

# 1. Johdanto

## 1.1 Tarkoitus ja kattavuus

Tuotetaan laskutusohjelma asiakkaan X käyttöön. Asiakas x on verkkopalveluyritys, joka toteuttaa tilausmallista verkkopalvelua. Palvelussa on kolme erilaista kuukausihintaista palvelutasoa (Basic, Pro, Enterprise) mahdollisilla alennuksilla vähennettynä ja laskutus toteutuu 30 päivän välein.

## 1.2 Tuote ja ympäristö

Laskutusohjelma on WPF-sovellus eli windows-ympäristössä toimiva desktop-sovellus, joka tuottaa käyttäjälleen listan hänen asiakkailtaan tulevista maksuista seuraavilta kuukausilta.

Laskutusohjelmaan syötetään asiakkaan tiedot ja hänen palvelutasonsa. Asiakkaita on oltava mahdollista myös poistaa sovelluksesta. Asiakkaan palvelutasoa on mahdollista muuttaa ylemmälle tai alemmalle tasolle.

Lisäksi asiakkaalle annetaan automaattisesti alennus, jos hän tilaa palvelua maksamalla kerralla 6 tai 12 kuukautta. Ohjelmaan haluttaisiin myös mahdollisuus ottaa huomioon 30 päivän ilmainen kokeiluaika palvelun ostajalle.

Tuotetta käytetään asiakkaan päätelaitteelta (tietokone tai kannettava tietokone), mobiilisovellusta ei tarvita.

Sovellus

## 1.3 Määritelmät, termit ja lyhenteet

|  |  |
| --- | --- |
| *Termi* |  |
| *Palvelutaso* | *Asiakkaan käyttämän verkkopalvelun sisältämien palveluiden laajuus; määritellyt kiinteät palvelupaketit Basic, Pro ja Enterprise* |
| Tietokanta | *Sovelluksen tietokantana toimivat json-tiedostot, jotka tallentuvat automaattisesti ohjelmatiedostoihin.* |
|  |  |
|  |  |
|  |  |
|  |  |

## 1.5 Yleiskatsaus dokumenttiin

Tässä dokumentissa käsitellään projektiryhmän toteuttaman Laskutusohjelman toiminnallisuus käyttäen vaatimusten kuvauksia sekä sovelluksen ja sen toimintojen mallinnusta kaavioin ja sanallisin kuvauksin. Dokumenttia käytetään suunniteltaessa sovellukseen toteutettavia ominaisuuksia ja dokumentissa asetetut vaatimukset määrittävät suunnittelu- ja toteutusvaiheessa suoritettavien tehtävien järjestyksen.

Luvussa yksi kuvataan tämän dokumentin tarkoitusta sekä lyhyesti sovelluksen yleistä tarkoitusta ja käyttäjäryhmää. Lisäksi luku sisältää käytettyjen termien ja lyhenteiden selitykset. Luvussa kaksi kuvataan tarkemmin toteutettavaa sovellusta sekä sen toimintaympäristöä. Luvussa kolme kuvataan järjestelmän vaatima tietosisältö. Luku neljä sisältää sovelluksen tärkeimmät toiminnot ja niiden tarkemman läpikäynnin. Luvussa viisi kuvataan sovelluksen vaatimat laitteisto- ja ohjelmistoliittymät. Luvussa kuusi kuvataan muista sovelluksen ominaisuuksia, kuten ylläpidettävyyttä. Luku seitsemän sisältää ohjelmiston suunnitteluun vaikuttavat rajoitteet. Luvussa kahdeksan käydään läpi hylätyt ratkaisuvaihtoehdot sekä näiden perustelut ja luvussa yhdeksän tuodaan esiin kehitettyyn sovellukseen esiin nousseet jatkokehitysajatukset.

# 2. Yleiskuvaus

Sovellus toimii yrityksen X suunnitteilla olevan verkkopalvelun laskutuksen perustana. Sovelluksessa on asiakas(käyttäjä) sekä palvelutaso. Palvelutasoja on useita. Laskutus tapahtuu kuukauden välein. Sovelluksessa on lisäksi 6 ja 12 kk ostomahdollisuus tuotteelle. 30 päivän ilmainen kokeilu huomioidaan sovelluksessa myös. Sovelluksessa on näkymä millä saa tulevat maksut näkyviin.

## 2.1 Ympäristö

Tuotteen toimintaympäristönä toimii C# WPF sovellus. Sovelluksen kaikki toiminnallisuudet tehdään C# ja muita applikaatioita tai plugineja ei käytetä.

## 2.2 Toiminta

Sovelluksen käyttäjäryhmä/kohderyhmä on yrityksen X laskutustiimi. Sovellus toimii Yrityksen X suunnitteilla olevan verkkopalvelun tuotteiden laskutuksen apuna.

2.3 Käyttäjät  
Sovelluksen käyttäjäryhmä/kohderyhmä on yrityksen X laskutustiimi. Sovellus toimii Yrityksen X suunnitteilla olevan verkkopalvelun tuotteiden laskutuksen apuna.

## 2.4 Yleiset rajoitteet

Sovellusta saa käyttää yritys X Lisenssillä. Määritellään vuosisopimuksella.

## 2.5 Oletukset ja riippuvuudet

Sovellus tarvitsee toimiakseen vähintään WIN7 käyttöjärjestelmän. Muita vaatimuksia ei ole. Sovelluksen tarvitsee olla yhteydessä internetiin erillisen tietokannan takia.

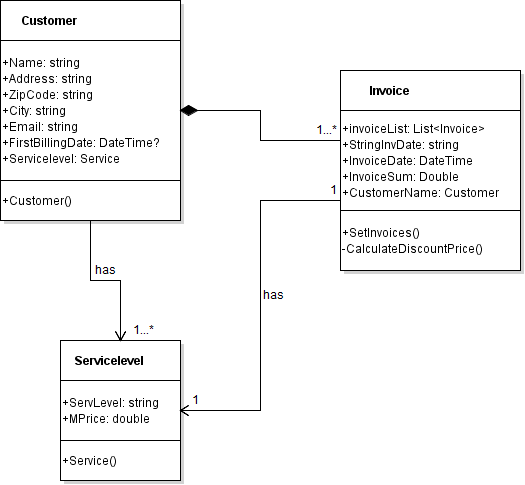
# 3. Käytettävä teknologia

## 3.1 Teknologian kuvaus

Sovellus toteutetaan WPF-sovelluksena C# olio-ohjelmointikielellä ilman ladattavia lisäkirjastoja. Sovellus sisältää kolme luokkaa, Customer, Invoice ja Servicelevel sekä kolme ikkunatasoa: MainWindow, CustomerWindow ja Servicewindow.

Servicelevel-luokan objektit luodaan ServiceWindown Tallenna-metodissa ja kuljetetaan servicelevels.json-tiedoston avulla CustomerWindown käyttöön. CustomerWindow.xaml.cs lukee serviceList-listaan sisään servicelevels.json -tiedoston ja populoi listan itemeilla CustomerWindow.xaml-tiedoston comboBoxServiceLevel-elementin.

CustomerWindow.xaml.cs luo Tallenna-metodissa Customer-objektin, joka ottaa vastaan Servicelevel-objektin comboBoxServiceLevel-elementistä. Samassa metodissa luodaan Invoice-objektit, jotka käyttävät sekä Customer- että Servicelevel-objekteja. Customer-objektien tiedot tallennetaan customerList-ObservableCollertioniin ja sieltä customerlist.json-tiedostoon. Invoice-objektien tiedot säilötään customerInvoiceList ObservableCollectioniin ja sieltä invoicelist.json-tiedostoon ja kuljetetaan sen avulla luettavaksi MainWindow.xaml.cs -tiedoston metodiin, josta ne siirretään näytettäväksi MainWindown listaelementtiin listViewInvoices. CustomerWindow.xaml.cs-tiedostossa on myös metodi, jolla voidaan poistaa CustomerWindown listViewCustomers-listaelementistä valittu kohde sekä asiakkaista että laskuista json-tiedostoja käyttämällä.



## 3.2 Tiedot ja tietokanta

Sovellus laskutus toimii tietopohjana. Asiakastietojen sekä laskuttajan tietojen tallentamiseen käytetään json-tiedostoja, joihin tallennetaan teknologian kuvauksessa mainittujen Invoice-, Customer- ja Service -luokista perustetut objektit. Tietojen lisäys ja poistaminen json-tiedostoihin tapahtuu tallentamalla muutokset kunkin luokan tyyppiseen objektilistaan ja tallentamalla lista muutoksineen json-tiedostoon, jonka avulla muutokset saadaan kuljetettua eri ikkunoihin.

### 3.1 Tietosisältö

Sovellukseen tarvitaan palvelutasojen perustietojen lisäksi asiakkaiden yhteys- ja osoitetietoja sekä päivämäärätietoja ja laskentaa laskulistojen muodostamiseksi. Tietohakemistoon on tallennettu sovelluksessa käytetyt tiedot tyyppeineen sekä WPF-projektin luokkien vastaavien muuttujien nimet.



#### 3.1.2 Customer-luokka

Customer-luokasta muodostetaan asiakas -objekteja. Luokan objektit sisältävät string-tyyppisenä datana asiakkaan nimen (Name), osoitteen (Address), postinumeron (ZipCode), kaupungin (City) ja sähköpostin (Email) sekä DateTime -tyyppisenä tiedon ensimmäisestä laskutuspäivästä (FirstBillingDate), jota on ajateltu hyödynnettävän jatkokehityksessä asiakkaan laskutustietojen muokkauksessa. Asiakkaan palvelutaso tallennetaan objektiin Service-luokan objektina Servicelevel.

#### 3.1.2 Service-luokka

Service-luokka kuvaa Laskutussovelluksen asiakkaalleen tarjoamia palvelutasoa. Siitä muodostettavia objekteja hyödynnetään sekä Invoice-luokan että Customer-luokan objekteissa. Service-luokan objekti sisältää palvelutason nimen eli string-tyyppisen ServLevel -muuttujan sekä double-tyyppisen MPrice-muuttujan, johon tallennetaan palvelun kuukausihinta.

#### 3.1.3 Invoice-luokka

Invoice-luokka tuottaa sovelluksen olennaisen tiedon eli laskutuslistat. Luokan objekteihin tallennetaan DateTime-tyyppinen InvoiceDate, jonka avulla muodostetaan sovelluksessa näytettäväksi string-tyyppinen StringInvDate. Lisäksi double-tyyppisenä tallennetaan laskun summa, jonka laskemisen apuna käytetään luokan metodille lähetettävää Service-luokan objektia. Lisäksi laskulle tallennetaan Customer-luokan tyyppinen CustomerName, jotta yhden asiakkaan laskuja listalla on mahdollista hallita Customer-luokan objektin kautta. Kaikki luodut Invoice-objektit tallennetaan luokan staattiseen invoiceListiin, jolta ne voidaan tallentaa invoicelist -json-tiedostoon ja hyödynnettäväksi muualla sovelluksessa.

### 3.2 Käyttöintensiteetti

Sovelluksen tiedot päivittyvät automaattisesti json-tiedostoihin kun muutoksia (lisäys/poisto) tehdään. Intensiteetti voi olla suuri ajoittain, mutta muutokset eivät vie paljon tilaa eikä kaistaa.

### 3.3 Kapasiteettivaatimukset

Tietokannan kokovaatimukset riippuvat asiakasmäärästä. Minimi Tietokannan koko on 10 GB. Kahdennettu tietokantapalvelin, jolloin toinen tietokanta toimii toisen tietokannan varmuuskopiona. Suorituskykyvaatimuksena tietokantapalvelin on oltava yhteydessä sovellukseen, jotta muokatut, poistetut tai uudet tiedot siirtyvät tietokantaan.

### 3.4 Tiedostot ja asetustiedostot

Sovelluksen asennustiedostot sekä varmuuskopiot pidetään erillään. Varmuuskopiot palauttavat 24 tunnin takaisen tilanteen.

# 4. Toiminnot

Sovelluksen tärkeimmät toiminnot ovat asiakkaan tietojen syöttö, laskujen muodostaminen alennuksineen, palvelutason tietojen syöttö sekä maksulistan näyttäminen asiakkaalle.

4.1 Asiakkaan tietojen lisäys ja käsittely

Käyttäjä voi lisätä sovellukseen asiakkaan sekä liittää tälle palvelutason. Asiakkaan tiedot ja palvelutaso lisätään samasta näkymästä, asiakastiedot syötetään tekstikenttiin ja palvelutaso valitaan avattavasta luettelosta. Käyttäjä voi myös poistaa asiakkaan, jolloin myös kaikki asiakkaan laskut poistuvat sovelluksesta. Tietojen muokkaus ei ole tässä julkaisussa mahdollista.

Asiakkaan tietojen muokkaus toteutetaan poistamalla asiakas ja syöttämällä hänen tietonsa uudestaan.

## 4.2 Palvelutason lisäys ja käsittely

Käyttäjä voi lisätä uuden palvelutason sovellukseen ja poistaa halutessaan palvelutasoja. Tietojen muokkaus ei ole tässä julkaisussa mahdollista.

## 4.3 Maksulista

Käyttäjä voi nähdä sovelluksesta listan asiakkaan tulevista maksuista seuraavilta kuudelta kuukaudelta. Listan voi tulostaa tulostimelle. Tietojen muokkaus ei ole tässä julkaisussa mahdollista.

## 4.4 Alennustiedot

Automaattisesti annettavien alennusten alennusprosentteja ei ole mahdollista muokata. Sovelluksen laatimishetkellä tiedossa olevat alennukset ovat:

* 6 kuukautta kerralla maksaville asiakkaille 10 % alennus, jonka jälkeen kuukausilaskutus normaalihinnalla
* 12 kuukautta kerralla maksaville asiakkaille 18 % alennus, jonka jälkeen kuukausilaskutus normaalihinnalla

## 4.5 Kokeilujakso tilaajille

Palvelun tilaajille voidaan valita annettavaksi ilmainen 30 päivän kokeilujakso.

# 5. Ulkoiset liittymät

Sovellus ei ole yhteydessä ulkoisiin laitteisiin, ohjelmistoliittymiin tai rajapintoihin.

## 5.1 Laitteistoliittymät

Sovellus ei ole yhteydessä laitteistoliittymiin tai rajapintoihin.

## 5.2 Ohjelmistoliittymät

Sovellus ei ole yhteydessä ulkoisiin ohjelmistoliittymiin tai rajapintoihin.

## 5.3 Tietoliikenneliittymät

Sovelluksella ei ole tarvetta tietoliikenneliittymälle. Laskujen muodostaminen sovelluksella ja niiden eteenpäin lähettäminen asiakkaille tapahtuu eri sovelluksilla.

# 6. Muut ominaisuudet 6.1 Suorituskyky ja vasteajat

Sovelluksen suorituskyky sekä vasteajat määräytyvät sovelluksen käytön mukaan. Laskutus sovelluksen käyttö on hetkittäistä joka ei vaadi vasteajalta ja suorituskyvyltä paljon.

## 6.2 Saavutettavuus (availability), toipuminen, turvallisuus, suojaukset

Sovellus tallentaa henkilötietoja. Henkilötietojen käyttöön liittyen noudatetaan GDPR määrityksiä. Tietokantapalvelin suojataan ensi tilassa palomuurilla. Sovellusta sekä tietokantapalvelinta ei määritetä käyttämään ulkoisia yhteyksiä.

## 6.3 Ylläpidettävyys

Palautustiedostot sekä varmuuskopioinnit sijoitetaan kahdennetuille palvelimille. Lisäksi sovellus sekä tietokanta tallennetaan ulkoiselle kovalevylle, jolloin palauttaminen onnistuu. Tietokannan päivittäminen varmuuskopioihin tapahtuu reaaliajassa. Ulkoiseen varmuuskopioon kerran kuukaudessa tai kun asiakkaiden määrä laskee tai nousee kymmenellä tai muissa tilanteissa, milloin huomataan, että siihen on tarvetta.

Versiopäivitykset suoritetaan ajanmukaisesti ja tietoturvapäivitysten osalta ne suoritetaan 24 tunnin kuluessa.

## 6.4 Siirrettävyys ja yhteensopivuus

Sovellus ei ole yhteensopiva muiden sovellusten kanssa tämänhetkisen tilanteen huomioiden.

Sovellusta pystyy käyttämään yleisimmät tietokonejärjestelmät.  
6.5 Operointi  
Sovellusta käytetään paikallisesti yrityksen tiloissa missä sijaitsee myös kahdennettu tietokantapalvelimet. Sovellusta ei ole tällä hetkellä mahdollista käyttää muualla.

6.6 Käytettävyys (Usability), käytön tehokkuus, käyttäjien tyytyväisyys  
Sovelluksen käyttökokemukset määritetään testijakson aikana (1kk). Käytön tehokkuus määräytyy yrityksen verkkopalvelun käytön mukaan.

# 7. Suunnittelurajoitteet

Sovelluksen suunnittelussa ei noussut esiin erityisiä rajoitteita muutoin kuin sovelluksen käyttöympäristöön liittyen. Koska kyseessä on .NET ympäristön Desktop-sovellus, se soveltuu käytettäväksi vain tietokoneella, jossa on Windows-käyttöjärjestelmä.

7.2 Laitteistorajoitteet

Sovellus ei aseta erityisiä vaatimuksia tai rajoitteita laitteiston suhteen, mutta koska kyseessä on .NET WPF-sovellus, se soveltuu käytettäväksi tietokoneella, jossa on alla täsmennetty käyttöjärjestelmä. Kuitenkin suositellaan vähintään 800MHz prosessoria, 512 MB RAM-muistia ja DirectX 9-yhteensopivaa näytönohjainta.

7.3 Ohjelmistorajoitteet

Sovellus on .NET WPF-sovellus, joka soveltuu käytettäväksi tietokoneella, jossa on vähintään Windows XP-käyttöjärjestelmä Service Pack 2:lla. Tämän vaatimusmäärittelydokumentin kuvaamassa sovellusversiossa ei käytetä rajapintoja muihin ohjelmiin, joten nämä eivät aiheuta rajoitteita sovelluksen suhteen. Mahdollisissa tulevissa kehitysversioissa asiaa on tarkasteltava uudestaan.

7.4 Muut rajoitteet  
Sovellus ei käytä verkkoyhteyttä.

# 8. Hylätyt ratkaisuvaihtoehdot

Sovelluksen toteuttaminen konsolisovelluksena päätettiin hylätä, koska se ei tarjoa riittävän helppokäyttöistä käyttöliittymää.

# 9. Jatkokehitysajatuksia

Asiakkaan tietojen muokkausmahdollisuus (tällä hetkellä täytyy poistaa asiakas ja lisätä uudet asiakastiedot).

Kaksoiskappaleiden tarkistus poisto listoilta/json-tiedostoista käsittelyvaiheessa.

Palvelutasojen listauksen automaattinen järjestäminen.

Asiakkaiden ja laskujen järjestämis- ja suodatusmahdollisuus listanäkymästä luokan ominaisuuksien mukaan (päiväys, nimi, summa etc).

Customer, Invoice ja Service-luokkien muokkaaminen tietokantavalmiuteen (käyttöön asiakasID, laskunumero, serviceID), normalisointi

Sovelluksen tietopohjan siirtäminen SQL-tietokantaan yllä olevien toteutuksen helpottamiseksi.

# Liitteet

Ulkoasun rautalankamalli, visuaalinen suunnitelma

Koodin rakenteen suunnitelma