

# **VAATIMUSMÄÄRITTELY**

## **Vuokra-asuntojen hallinnan sovellus**

# HARJOITUSTYÖ OHJELMISTOTUOTANTO OHSU19S

## **Projektiryhmä**

Tapani Tarri projektipäällikkö ja suunnittelija

OHSU19S satunnaisesti valittu testausasiantuntija sovelluksen testaajaksi

## **Johto- ja ohjausryhmä**

Jyri Lehto & kump.

## **Asiakas**

Seinäjoen ammattikorkeakoulun asuntosijoittajat

# HARJOITUSTYÖ OHJELMISTOTUOTANTO OHSU19S

## Sisällysluettelo

Sisällysluettelo .....	3
VERSIONHISTORIA.....	4
1 Johdanto .....	5
1.1 Termit .....	5
2 Vaatimukset.....	5
2.1 Ohjelmalliset vaatimukset .....	5
2.2 Toiminnalliset vaatimukset.....	5
2.3 Laadulliset vaatimukset .....	6
3 Toiminnot .....	6
4 Käyttöliittymä .....	7
5 Vaaditut ulkoiset liittymät .....	7
6 Rajoitteet suunnittelulle ja toteutukselle.....	7
7 Validointi.....	7
8 Dokumentointi.....	7
9 Lähteet.....	8
LIITE: .....	8

**HARJOITUSTYÖ OHJELMISTOTUOTANTO OHSU19S****VERSIONHISTORIA**

<b>Versio</b>	<b>Päivämäärä</b>	<b>Laatija</b>	<b>Kuvaus</b>	<b>Hyväksyjä</b>
0.01	5.12.2019	T.Tarri	Keskeneräinen	
1.0	6.12.2019	T.Tarri	OHSU19S-versio	

# HARJOITUSTYÖ OHJELMISTOTUOTANTO OHSU19S

## 1 Johdanto

Harjoitustyöprojektin tarkoituksena on kehittää pilottisovellus vuokra-asuntojen hallintajärjestelmästä vastaamaan asuntosijoittajien ja vuokranantajien tarpeisiin. Tavoitteena on kehittää sovellus, mikä tietotekniikan ja –liikenteen keinoin mahdollistaa käyttäjäystävällisesti perustoimintojen osalta tiedon välittämisen toimintakentän eri osapuolten välillä ja ensisijaisesti vuokranantajan tarpeet huomioiden. Sovellus kehitetään OHSU19S olio-ohjelmointityön harjoitustyönä hyödyntäen ohjelmointiteknisesti olio-ohjelmoinnin mahdollisuuksia toteutuksessa

### 1.1 Termit

- *C sharp (C#)* Microsoftin .NET-alustalle kehitetty ohjelmointikieli.
- .NET-alusta (.NET Framework) Microsoftin kehittämä ohjelmistokomponenttikirjasto, jota Microsoft Visual Studio-ympäristössä kehitetyt ohjelmistot käyttävät.
- Visual Studio Microsoftin ohjelmankehitysympäristö.
- Microsoft yhdysvaltalainen ohjelmistoalan yritys.
- Olio-ohjelmointi on ohjelmoinnin teoreettinen malliratkaisu, jossa ohjelmointiongelmien ratkaisut jäsennetään olioiden yhteistoimintana ja jossa oliot sisältävät toisiinsa loogisesti liittyvää tietoa ja toiminnallisuutta.

## 2 Vaatimukset

### 2.1 Ohjelmalliset vaatimukset

- O1** Sovelluksen tulee hyödyntää ja noudattaa olio-ohjelmoinnin periaatteita
- O2** Käytettävä ohjelmointikieli on ensisijaisesti C sharp (C#).
- O3** Ohjelmointiratkaisussa toteutetaan esitetyssä osaamismatriisissa (LIITE) määritellyt piirteet, mitkä osaltaan laadullisesti osaamisen suhteen osoittavat sovelluksen laadinnassa käytettyä osaamistasoa

### 2.2 Toiminnalliset vaatimukset

- T1** Sovelluksen pääohjelmasta loogisesti edetään toteuttamaan luokissa mainittuja toimintoja/metodeja.
- T2** Käyttöliittymänä pilottisovelluksessa sovelluksen toimintojen ohjaamiseen käytetään tekstipohjaista komentoriviliittymää (konsoli).
- T3** Lisäksi toteutetaan yksinkertainen graafinen käyttöliittymä, mikä avaa komentoriviliittymän ikkunan sovelluksen käyttöä varten (tämä vaatimus ei ole pakollinen).
- T4** Käyttöliittymän pilottisovelluksessa tulee tekstikehotein ja -komennoin selkeästi suomenkielellä ohjata käyttäjää sovelluksen käytössä.
- T5** Käyttöliittymän kautta kyetään avaamaan laadittava sovelluksen tuotekuvausdokumentti ennen sovelluksen muuta käyttöä.
- T6** Pilottisovelluksessa ei vaadita käyttöliittymän kautta mahdollisuutta ylläpitää hallittavista asunnoista listaa (tämä ominaisuus voidaan lisätä myöhemmin tulevaan versioon), mutta luokat kahdesta asunnosta toteutetaan pilottisovellukseen ja käyttöliittymän kautta nähdään listaus näiden kahden asunnon nimistä, mitkä voivat olla pilotissa samat kuin luokkiensa nimet.
- T7** Käyttöliittymän kautta voidaan kirjoittaa haluttu määrä asukkaiden nimiä, mitkä ohjelmallisesti kirjoitetaan asukkaiden nimet sisältävään tekstitiedostoon.

## HARJOITUSTYÖ OHJELMISTOTUOTANTO OHSU19S

**T8** Käyttöliittymän kautta voidaan kirjoittaa omistajan nimi, mikä ohjelmallisesti kirjoitetaan omistajien nimet sisältävään tekstitiedostoon.

**T9** Käyttöliittymän kautta voidaan vielä muuttaa edellä kirjoitetun omistajan nimeä seuraavan komentoliittymän antaman ohjeen mukaan.

**T10** Käyttöliittymän kautta valitaan, kumman asunnon osalta maksetaan vuokra, jos kyseessä on vuokranmaksun maksaminen.

**T11** Käyttöliittymän kautta valitaan mahdollisuus maksun kirjaamiseksi ohjelmallisesti määritellylle omistajalle sitä varten olevaan suoritettujen maksujen tekstitiedostoon.

**T12** Käyttöliittymän kautta kirjataan vuokralaisen nimi, joka em. maksun suorittaa ja se kirjoitetaan ohjelmallisesti sitä varten olevaan tekstitiedostoon.

**T13** Käyttöliittymän kautta kirjoitetaan em. maksun määrä euroina ja se kirjoitetaan edelleen ohjelmallisesti sitä varten olevaan tekstitiedostoon.

**T14** Käyttöliittymän kautta näytetään edellä maksettu summa, sen maksaja, mitä asuntoa maksu koskee ja asunnon omistajan nimi. Näytetään myös tämän maksutapahtuman päivämäärä ja kellonaika.

**T15** Käyttöliittymän kautta edellisen lisäksi näytetään listaus myös aiemmista maksutapahtumista. Niiden maksutapahtumien osalta kirjataan tapahtuma-ajankohdat omistajaa kohden sitä varten luotuun tekstitiedostoon.

**T16** Käyttöliittymän kautta maksutapahtumatiedot voidaan ohjata ohjelmallisesti kirjoitettavaksi veroilmoitusta varten olevaan tekstitiedostoon.

**T17** Käyttöliittymän kautta voidaan ohjata maksutapahtumatietojen välittäminen sähköpostitse määritellyyn (esim. pankin) sähköpostiosoitteeseen (tämä vaatimus ei ole pakollinen pilottisovellutuksessa).

**T18** Tämän jälkeen sovellus sulkeutuu.

**T19** Mikäli edellä mainittuja toiminnallisuuksia ei jonkin toiminnallisuuden osalta valita käyttöliittymän kautta toteutettavaksi sallitaan toteuttamatta jätetyn toiminnallisuuden jälkeen sovelluksen sulkeutuminen pilottisovelluksessa ja siitä ei tarvitse antaa ohjelmallisesti käyttäjälle erillistä ilmoitusta.

### 2.3 Laadulliset vaatimukset

**L1** Osaamismatriisissa esitetyt pystyakselilla esitetyt kohdat (indeksit) ”Oliot, Luokkarakenne, Rajapinnat, Dokumentointi ja Tallentaminen” tulee toteutua sovelluksen ohjelmointiratkaisuissa vähintään vaaka-akselin indeksin ”1” osalta. Osaamisen laatua mittaa toteutuksessa se, miten laajasti osaamismatriisin alkioden määrittelyt toteutuvat ko. harjoitustyönä olleessa sovelluksessa.

**L2** Ohjelmallinen toiminnallisuus on kohtuullisen virheetöntä.

**L3** Pilottisovelluksen toteutus on ylläpidettävää ja jatkokehiteltävää sekä ominaisuuksiltaan laajennettavissa.

**L4** Suorituskykyyn ei tarvitse kiinnittää erityistä huomiota pilottisovelluksessa.

## 3 Toiminnot

Ohjelmallisia toimintoja ovat:

- Komentorivikäyttöliittymän kautta tietojen kirjoittaminen ja niiden esittäminen.
- Tekstitiedostoihin kirjoittaminen ja niistä lukeminen.
- Tekstitiedon sarjallistaminen tavuiksi ja niiden kirjoittaminen tiedostoon ja edelleen sarjallistetun tiedon konvertointi tiedostosta takaisin tekstiksi ja sen kirjoittaminen tekstitiedostoon.
- Tiedostojen sähköpostitus (optiona).

# HARJOITUSTYÖ OHJELMISTOTUOTANTO OHSU19S

## 4 Käyttöliittymä

Pilottisovelluksen käyttö voidaan aloittaa kehitysympäristönä käytetyn Visual Studion esim. 2017 tai 2019 versioista, mistä päästään komentorivikäyttöliittymään suoraan tai ensin graafiseen käyttöliittymään ja siitä edelleen komentorivikäyttöliittymään. Sovelluksen käynnistäminen voidaan suorittaa myös sovelluksen tai graafisen käyttöliittymän exe-tiedostoista. Käyttöliittymien ja pilottisovelluksen tulee toimia Windows 10 käyttöjärjestelmän ohessa.

## 5 Vaaditut ulkoiset liittymät

Pilottisovelluksen osalta ei vaadita ulkoisia liittymiä. Tekstitiedostoihin kirjoittaminen ja niistä lukeminen voidaan toteuttaa Internetin välityksellä myös ulkoisena tiedonsiirtona. Samoin tekstitiedostojen siirtoon voidaan toteuttaa ohjelmallisesti sähköpostitusta toteuttava ominaisuus.

## 6 Rajoitteet suunnittelulle ja toteutukselle

Pilottisovelluksen osalta rajoitteena on sen julkaiseminen OHSU193 Olio-ohjelmoinnin kurssin harjoitustyönä jo 29.11.2019, joten sen ajankohdan jälkeen em. julkaisuun ei tällä OHSU193 Ohjelmistotuotannon kurssin aikana tehtävällä vaatimusmäärittelyllä ole johdattavaa vaikutusta vaan mahdolliset vaikutukset kohdistuvat mahdollisiin seuraaviin ko. Olio-ohjelmoinnin kurssin harjoitustyön versioihin.

## 7 Validointi

Pilottisovelluksen validointi on ollut tarkoitus toteuttaa OHSU193 Ohjelmistotuotannon kurssin aikana määritellyin testaustoimenpitein. Tämä testaus kuitenkin todennäköisesti kurssin aikataulutuksen vuoksi jää toteuttamatta, joten sovelluksen kehittäjä Tapani Tarri on omakohtaisesti validoinnin toteuttamisesta vastuussa ja sen vuoksi on validointi Tapani Tarrin toimesta toteutettu 6.12.2019 vesiputousmallin periaatteen mukaan kartoitettujen tässä vaatimusmäärittelyssä esitettyjen vaatimusten suhteen.

## 8 Dokumentointi

- ohjelman koodi ja tuotekuvaus ovat tallennettu 5.12.2019 GitHub-verkkosivuston versionhallinnan tiedosto-/dokumentaatiokantaan, minne johtaa seuraava linkki  
<https://github.com/k1900295/Harjoitusty-olio-ohjelmointi-Tapani-Tarri> .
- Vaatimusmäärittely ”VAATIMUSMÄÄRITTELY\_vuokra-asuntojen hallinnan sovellus\_v.1.0.pdf” on tallennettu 6.12.2019 GitHub-verkkosivuston versionhallinnan tiedosto-/dokumentaatiokantaan, minne johtaa seuraava linkki  
<https://github.com/k1900295/Harjoitusty-olio-ohjelmointi-Tapani-Tarri>.
- Testaussuunnitelma tallennetaan GitHub-verkkosivuston versionhallinnan tiedosto-/dokumentaatiokantaan, minne johtaa seuraava linkki  
<https://github.com/k1900295/Harjoitusty-olio-ohjelmointi-Tapani-Tarri>.
- Testausraportti tallennetaan GitHub-verkkosivuston versionhallinnan tiedosto-/dokumentaatiokantaan, minne johtaa seuraava linkki  
<https://github.com/k1900295/Harjoitusty-olio-ohjelmointi-Tapani-Tarri>.

# HARJOITUSTYÖ OHJELMISTOTUOTANTO OHSU19S

## 9 Lähteet

- Wikipedia
- Internet
- OHSU19S Ohjelmistotuotantokurssin materiaali

## LIITE:

Osaamismatriisi:

	1	2	3	4	5
Oliot	osaa luoda luokasta olion	osaa luoda dynaamisen oliokokoelman	osaa määritellä oliolle kentät ja metodit	osaa määritellä näkyvyydet julkisiksi ja yksityisiksi	osaa määritellä oliolle property-kenttiä
Luokkarakenne	osaa määritellä luokan, jossa on konstruktori	osaa määritellä luokan, jossa on viite toiseen olioon	osaa määritellä luokan, jossa on viite oliokokoelmaan	osaa määritellä luokan, joka perii yläluokan	osaa määritellä luokan, jossa hyödynnetään metodin ylikirjoittamista
Rajapinnat	osaa määritellä rajapinnan	osaa määritellä rajapinnan toteuttavan luokan	osaa hyödyntää polymorfismia rajapinnan avulla	lähdekoodista on erotettu jokin osa rajapinnan avulla	lähdekoodista on erotettu useampi osa rajapintojen avulla
Dokumentointi	lähdekoodi on luettavaa	lähdekoodi on jäsennelty	lähdekoodista on tehty luokkakaavio	lähdekoodissa on kommentteja	lähdekoodi on itsekommentoiva
Tallentaminen	osaa tallentaa tekstiä tiedostoon	osaa ladata tekstiä tiedostosta	osaa tallentaa oliodatan tiedostoon	osaa ladata oliodatan tiedostosta	osaa hyödyntää oliodatan sarjallistamista