**Harjoitustyön loppuraportti**

**Käyttöliittymien ohjelmointi**

Konsta Mustonen

**Sisältö**

[1 GitHub-linkki 2](#_Toc480670095)

[2 Lopullinen rakennekaavio ja perustelut muutoksille 2](#_Toc480670096)

[3 Projektissa toteutetut luokat 2](#_Toc480670097)

[4 Työaikaraportti 3](#_Toc480670098)

[5 Näyttökaappaukset sovelluksesta 4](#_Toc480670099)

[6 Ongelmat ja niiden ratkaisut 5](#_Toc480670100)

[7 Testaus 6](#_Toc480670101)

[8 Itsearviointi 6](#_Toc480670102)

[9 Kommentit kurssista 7](#_Toc480670103)

# GitHub-linkki

<https://github.com/TheKotti/GUI-HT>

# Lopullinen rakennekaavio ja perustelut muutoksille

Verrattuna alkuperäiseen luokkakaavioon, lopullinen työ päätyi hyvin suppeaksi. Luokkia oli tarkoitus olla kolme Hitman-pelin eri tehtävätätyypeille sekä yksi järjestelmän käyttäjille. Huonon ajankäytön vuoksi jouduin karsimaan työtä niin, että vain yhtä tehtävätyyppiä, Elusive Target, käsiteltäisiin. Kun käsiteltävien taulukoiden määrä putosi, poistui tarve useille luokille ja niiden periyttämiselle Mission-luokasta.

Alkuperäisiä mockupeja katsoessani en suoraan sanoen osannut sanoa, mikä niiden perimmäinen ajatus oli. Vaikka käsiteltävien taulukoiden karsinta tarkoitti, että toinen alkujaan suunnitelluista ikkunoista olisi joka tapauksessa turha, oli minusta silti hyvä, että jätin suunnitelman pääosin huomiotta. Mikäli taulukoita olisi enemmän, navigoisin niitä mieluummin Menusta löytyvän valikon kuin erillisen ikkunan avulla.

Tärkeimpiin toimintoihin pääsin mielestäni hyvin, laajuuden kutistumisesta huolimatta. Tietojen muokkaaminen ja lisääminen on vaivatonta, varmuuskopioiden hallinta toimii, ja tietojen poistaminen poistaa rivit myös taulukoista, joihin poistettavaan tietoon viitataan. Teknisesti ottaen ohjelma ei täytä vaatimusta tietokantayhteyden ottamisesta käynnistyessä, mutta nyt aiheeseen perehtyneempänä se ei tunnu muutenkaan hyvältä toimintaperiaatteelta.

# Projektissa toteutetut luokat

Projektin keskeinen luokka on Elusive, joka sisältää Hitman-pelin Elusive Target -tehtävien tietoja. Luokka itsessään on hyvin yksiselitteinen sen sisältäessä pääosin vain int-, string ja DateTime-tyyppistä tietoa. Näistä vain DateTime osoittautui hieman haastavaksi käsitellä, johtuen varmasti paljon siitä, että käsittelin sitä aluksi liian monimutkaisesti.

Tietokannasta haettavat Elusivet kootaan ElusiveList-kokoelmaluokkaan, johon on koottu Elusive-joukon käsittelyyn liittyviä metodeja. Näistä selkeästi oleellisin on metodi, jolla tietokannan rivit muutetaan olioiksi, muiden metodien lähinnä ajaessa muita metodeja. Suuri osa tietojenkäsittelystä tehdään suoraan tietokantaan DBElusives-luokassa, joka hoitaa tietokantayhteyden käytön ja kaikki lopulta tehtävät muutokset.

Tieto kulkee käytännössä hieman mutkan kautta, sillä ohjelman keskiössä oleva DataGrid, johon ElusiveListillä haettu lista olioita kootaan. Kun DataGridistä valitaan rivi, rivin tiedot haetaan TextBox-, ComboBox ja DatePicker-kontrolleihin. Update-nappia painettaessa DataGrid päivitetään vastaamaan kontrolleissa olevia tietoja, jonka jälkeen DataGridissä oleva tieto lähetetään tietokantaa kohti. Tietokantaan tehdyn muokkauksen jälkeen tietokannasta haetaan päivitetty lista ja kiinnitetään se DataGridiin. Tiedon toistuva hakeminen tietokannasta ei varmasti ole tehokkain mahdollinen ratkaisu, mutta olen havainnut sen silti hyvin toimivaksi, ainakin pienellä tietokannalla.

# Työaikaraportti

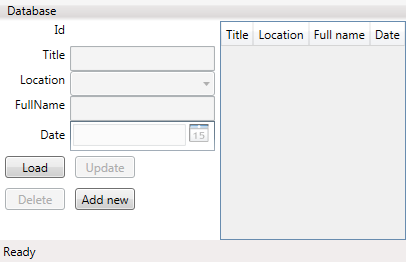
Minulla oli ajankäytössä suuria ongelmia. Aluksi suuri osa harjoitustyölle varatusta ajasta meni täysin hukkaan, kun Entity Framework ei osoittautunut yhteistyökykyiseksi. Kun työkalut tuottivat vaikeuksia, oli työssä alkuun pääseminenkään hyvin haastavaa eikä minkäänlaisesta flowsta ollut tietoakaan.

Lopulta hylkäsin Entityn ja yritin tehdä vanhaan harjoitukseen pohjautuen ohjelman ilman Entityä. Pääsin tällä tavoin alulle, mutta minulle tuli pian selväksi, että jäljellä oleva aika, alkuperäinen visioni ja tahti jolla työ eteni eivät muodostaneet toimivaa yhtälöä. Keuhkoputkentulehdus ei myöskään nopeuttanut asioita.

Lähellä opintojakson loppua kysyin aiheen mahdollisesta vaihtamisesta ja uuden aiheen sijaan sain object-relational mapping -pohjaisen mallin, jonka avulla pystyin ottamaan uuden alun samalle työlle. Malli kävi järkeeni paljon paremmin kuin aiemmat yritelmät ja pääsiäisen aikana pääsin vireeseen ja onnistuin tekemään valtaosan nyt järkevämpään mittaan karsitusta työstä muutamassa päivässä.

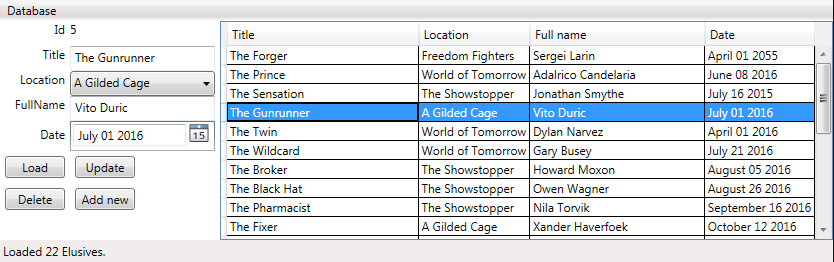
# Näyttökaappaukset sovelluksesta

Alusta lähtien halusin pitää eri näkymien määrän minimissään, ja kun työn laajuus karsiutui, ei näkymiä voinutkaan tehdä montaa. Lähes kaikki keskittyy yhteen ikkunaan, lukuun ottamatta erinäisiä varmistusviestejä ja uuden rivin lisäämistä tietokantaan.

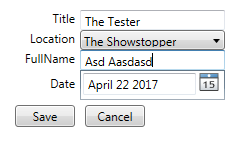


Kuvio 1: Ohjelma heti käynnistämisen jälkeen.

Virheiden ja väärinkäytön minimoinniksi, eri kontrollit ovat käytössä vain, kun niillä on mahdollista tehdä jotain. Välittömästi käynnistämisen jälkeen käyttäjä voi vain ladata tiedot näkyviin, lisätä uuden rivin tai Menua käyttäen ladata varmuuskopion tai palauttaa tietokannan varmuuskopiosta. Uuden rivin lisääminen olemassa olevia rivejä näkemättä ei ole minusta kovin intuitiivista, mutta en nähnyt hyvää syytä estää sitä. Napin painallusta seuraavan tietojen käsittelyn jälkeen mitään riviä ei valita oletuksena ja kontrollit poistuvat käytöstä, kunnes käyttäjä valitsee taas rivin.



Kuvio 2: Ohjelma, kun tiedot on haettu ja rivi valittu.



Kuvio 3: Uuden rivin lisäys.

Add new- nappia painettaessa aukeaa uusi ikkuna, johon täytetään uudelle riville tulevat tiedot, samaan tapaan kuin pääikkunassa päivitetään olemassa olevia tietoja. Pidin erillistä ikkunaa tässä tapauksessa käyttäjän kannalta selkeämpänä ratkaisuna kuin pääikkunan päivityskontrollien käyttämistä.

Muita aukeavia ikkunoita ovat vain tietokannan palautusta tehtäessä tiedostonvalintaikkuna sekä palautuksen ja rivin poiston yhteydessä varmistusviesti. Tietokantaa varmuuskopioidessa ohjelma ei kysy, minne tiedosto tallennetaan, koska pidän luontevana tallentaa se samaan kansioon ohjelman kanssa ja nimetä se automaattisesti päiväyksen mukaan.

# Ongelmat ja niiden ratkaisut

Todennäköisesti aiheutin itselleni huomattavan määrän tarpeettomia ongelmia olemalla perehtymättä liian tarkasti ORM-pohjaiseen malliin. Ensisilmäykseltä se vaikutti hyvin paljon sellaiselta, miltä olin ajatellut työni näyttävän lopulta. Halusin yrittää löytää ratkaisut itse kopioimatta liikaa suoraan mallista, ja sen seurauksena ratkaisut eivät todennäköisesti olleet tehokkaimpia tai käytännöllisimpiä mahdollisia.

Ongelmat olivat kuitenkin suhteellisen pieniä ja nopeasti ratkeavia, joskaan ratkaisut eivät välttämättä olleet erityisen hyviä. ObservableCollectionin käyttö olisi todennäköisesti välttänyt merkittävän osan niistä, sillä kulutin paljon aikaa yhdistääkseni DataGridin ja sitä ohjaavat eri syötteet haluamallani tavalla. En halunnut DataGridin olevan suoraan muokattavissa ja halusin sen päivittyvän täsmälleen nappia painettaessa. DataGrid ja esimerkiksi nappien aktivoimisen hallinnointi ja StatusBarin tekstin muokkaus toisesta ikkunasta vaativat kohtuullisen paljon hienosäätöä, mutta eivät keskeyttäneet työn etenemistä liiaksi.

Suurimmat ongelmat tulivat viime hetkellä lisätystä DatePickeristä ja sen hienosäätämisestä. Olin yllättynyt, miten haastavaa päiväyksen oletusformaatin muuttaminen oli. Onneksi joku muukin oli joskus yrittänyt muuttaa formaattia ja ongelmaan löytyi valmis, monirivinen ratkaisu XAMLin puolelle. Sen käyttö ratkaisi osaltaan myös ongelman, jossa käyttäjä pystyi kirjoittamaan DatePickeriin mitä tahansa. Myös DatePickeriin liittyvä pieni, mutta paljon aikaa syönyt ongelma liittyi DataGridin päivittämiseen DatePickerin perusteella. Tarkastamisesta huolimatta en aluksi huomannut SelectedDateProperty-kenttää, ja kulutin useita tunteja etsien vaihtoehtoista ratkaisua.

# Testaus

Ohjelmassa ei ole kovin montaa käyttäjälle näkyvää liikkuvaa osaa, mikä helpotti testausta paljon. Aina uuden ominaisuuden lisätessäni minun oli nopeaa käydä läpi kaikki aiemmat ominaisuudet ja varmistaa, ettei uusi ominaisuus häirinnyt vanhaa ominaisuutta.

Lisäsin ominaisuuksia myös testauksen kannalta järkevässä järjestyksessä, eli uuden ominaisuuden testaukseen pääsi loogisesti käymällä läpi liudan jo lisättyjä ominaisuuksia. Rivin poistoa testatessa kävin läpi myös rivien hakemisen sekä rivin lisäämisen ja muokkaamisen yhtä riviä käsitellen. Tietoa käsiteltäessä tarkastin myös ajoittain phpMyAdminin avulla, olivatko muutokset päivittyneet tietokantaan odotetulla tavalla.

Yksikkötestaamiselle en nähnyt tarvetta testitulosten tullessa näkyviin suoraan ohjelman käyttöliittymään, enkä oikeastaan tiedä kuinka yksikkötestaus edes tulisi suorittaa tällaisessa projektissa.

# Itsearviointi

Ajankäyttö oli surkeaa ja jouduin karsimaan alkuperäistä visiota melko paljon. Ohjelmalla pystyy mielestäni tekemään muokkauksia täsmälleen suunnitellulla tavalla, mutta valitettavasti muutostenteon mahdollisuus rajoittuu vain pieneen osaan tietokantaa. Muut taulukot, joita alun perin toivoin voivani muokata ovat melko samankaltaisia kuin ohjelman käsittelemä taulukko, mutta poikkeavat kuitenkin sen verran, ettei ajan puitteissa ollut järkevää sekaannuttaa niitä asiaan.

En ole törmännyt odottamattomiin virheisiin ohjelman viimeisimmän version käytössä, ainoat virheviestit, joita olen saanut ovat johtuneet tietokantayhteyden pettämisestä. Vaikka ohjelma tuntuisi toimivan teknisesti virheettömästi, en voi väittää, että koodi olisi erityisen laadukasta. Omintakeinen lienee osuvampi kuvaus.

Vaikka koin tehneeni töitä tehokkaasti loppuhetkillä ja ohjelma toimii kuten pitääkin, jäi tästä työn laajuuden ja laadukkuuden vuoksi sellainen kakkosen maku suuhun.

# Kommentit kurssista

Aihealue tuntui todella laajalta ja WPF:stä tuntui jatkuvasti löytyvän jotain uutta, joka vaatisi syventymistä, joten olisin suosinut enemmän kontaktiopetusta tähän eri muotoiluihin ja kontrolleihin liittyen ja harjoitustyön tekemistä siinä sivussa. Sama tunne jäi joskus kontaktiopetuksen jälkeenkin, kun tunnilla oltiin käyty läpi jonkinlaisen ohjelman teko. Vaikka tunnilla seuraaminen onnistui ja työskentely oli sillä hetkellä järkeenkäyvää, tuli asiaa välillä niin paljon, ettei läheskään kaikkea pystynyt sisäistämään.