Ohjelmoinnin harjoitustyön

loppuraportti

Yauheni Baikou (K8792) & Roope Melasniemi (K8980)

Harjoitustyön loppuraportti

Huhtikuu 2017

Tekniikan ala

Insinööri (AMK), tieto- ja viestintätekniikka

Sisällys

[1.1 Linkki Githubiin 1](#_Toc480736391)

[1.2 Rakennekaavio ja Luokkien kuvaus 1](#_Toc480736392)

[1.3 Työaikaraportti 3](#_Toc480736393)

[1.4 Näyttökaappauksia projektista 3](#_Toc480736394)

[1.5 Ongelmat 3](#_Toc480736395)

[1.6 Testaus 3](#_Toc480736396)

[1.7 Itsearviointi 4](#_Toc480736397)

## Linkki Githubiin

<https://github.com/YauSollerS/TTOS0300S3-HARJOITUSTYO>

## Rakennekaavio ja Luokkien kuvaus

Lisäsimme alkuperäiseen suunnitelmaan vielä muutamia luokkia esim:

- **Tallennukset**, joka sisältää metodeja, jotka tallentavat tiedot JSON tiedostoon ja lataa tiedot sieltä.

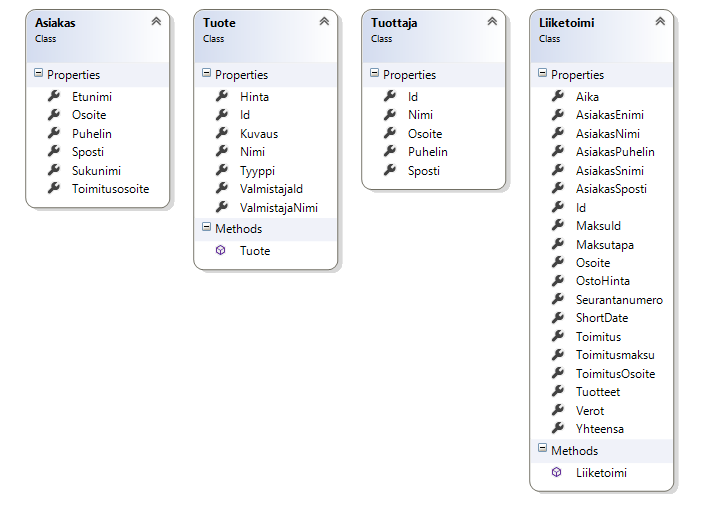
- **EnumBindingSourceExtension**- luokan, joka helpottaa Enum:in Data Bindaamista WPF:ssä.

- **ReplayCommand** luokan, joka toteuttaa ICommand rajapinnan. Sen tapaukset ovat komento kohteita, joiden kautta näkymä voi kutsua metodeja.

Käytimme **iCommandia** metodien ja komentojen luomiseen, sillä koimme, että se helpotti niiden luomista valtavasti.

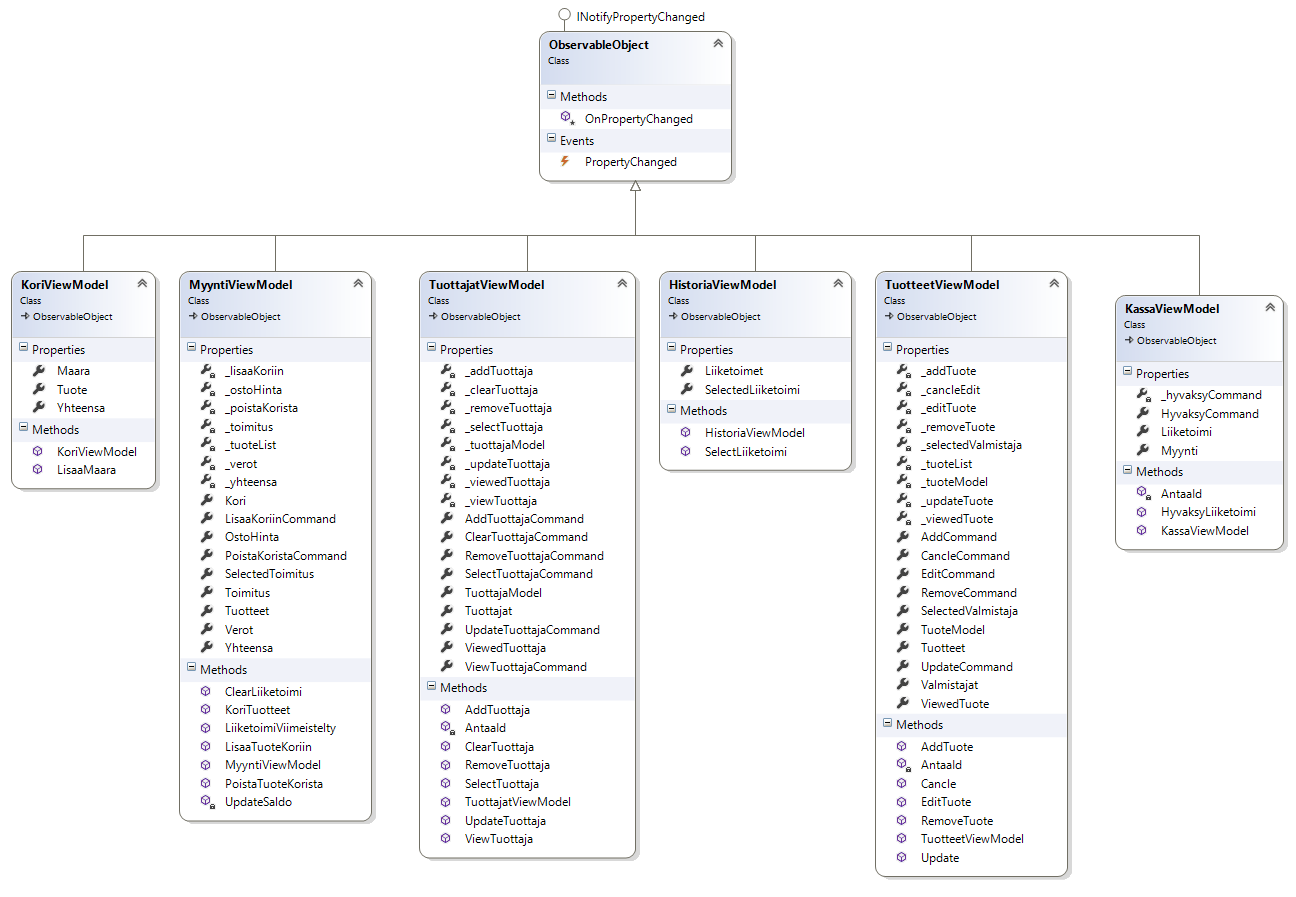
**Model -luokat (**Luokat, jotka sisältävät täkeitä tietoja, kuten ominaisuuksia. Esim. nimet, id:t, osoitteet jne.)**:**

* **Asiakas**
* **Liiketoimi**
* **Tuote**
* **Tuottaja**



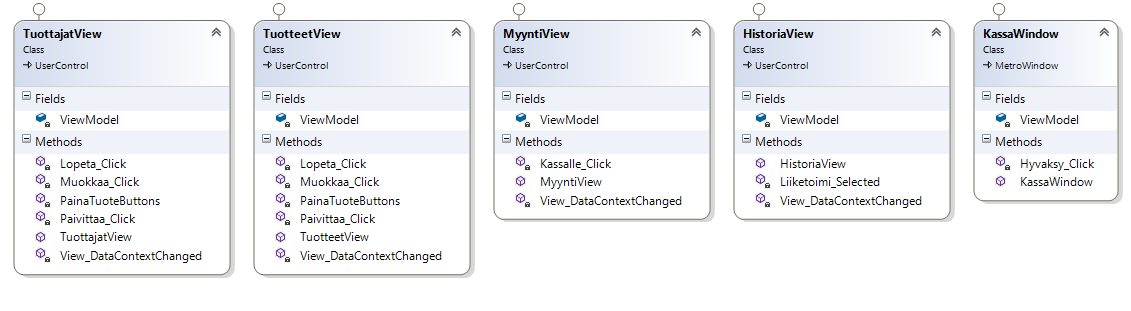
**ViewModels – luokat (**ViewModel(luokat) tarjoaa perustusmäärityksen View-elementin logiikan tukemiseksi. ObservabeObject mahdollistaa INotifyPropertyChanged- ra-japinnan toteuttamisen):

* **HistoriaViewModel**
* **KassaViewModel**
* **KoriViewModel**
* **MyyntiViewModel**
* **ObservableObject**
* **TuottajatViewModel**
* **TuotteetViewModel**



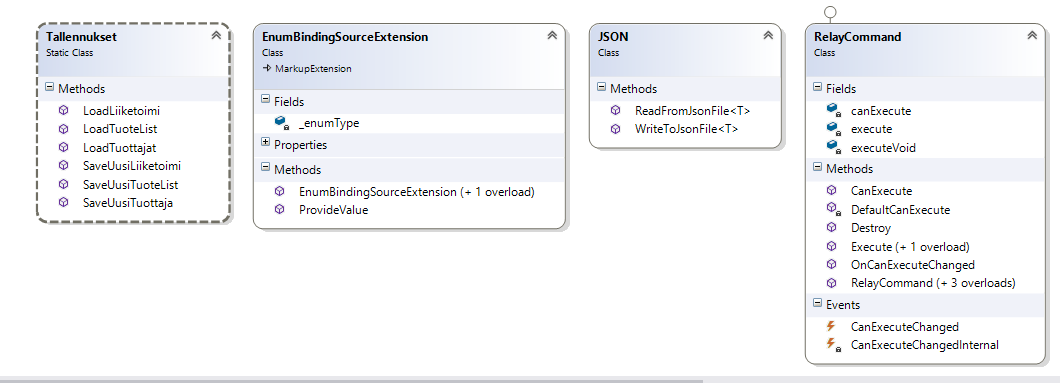
**View – luokat** (View luokat sisältää kontrolleja):

* **TuottajatView**
* **MyyntiView**
* **TuotteetView**
* **HistoriaView**
* **KassaWindow**



**Lisäluokat:**

* **JSON** - JSON luokka sisältää koodia, joka ohjaa JSONia.
* **EnumBindingSourceExtension** - EnumBindingSourceExtension- luokka helpottaa Enum:in Data Bindaa-mista WPF:ssä.
* **RelayCommand -** ReplayCommand- luokka toteuttaa ICommand rajapinnan. Sen tapauk-set ovat komento kohteita, joiden kautta näkymä voi kutsua metodeja.
* **Tallennukset -** Tallennukset- luokka sisältää metodeja, jotka tallentavat tiedot tiedostoi-hin ja tuo/lukee tiedot tiedostoista.

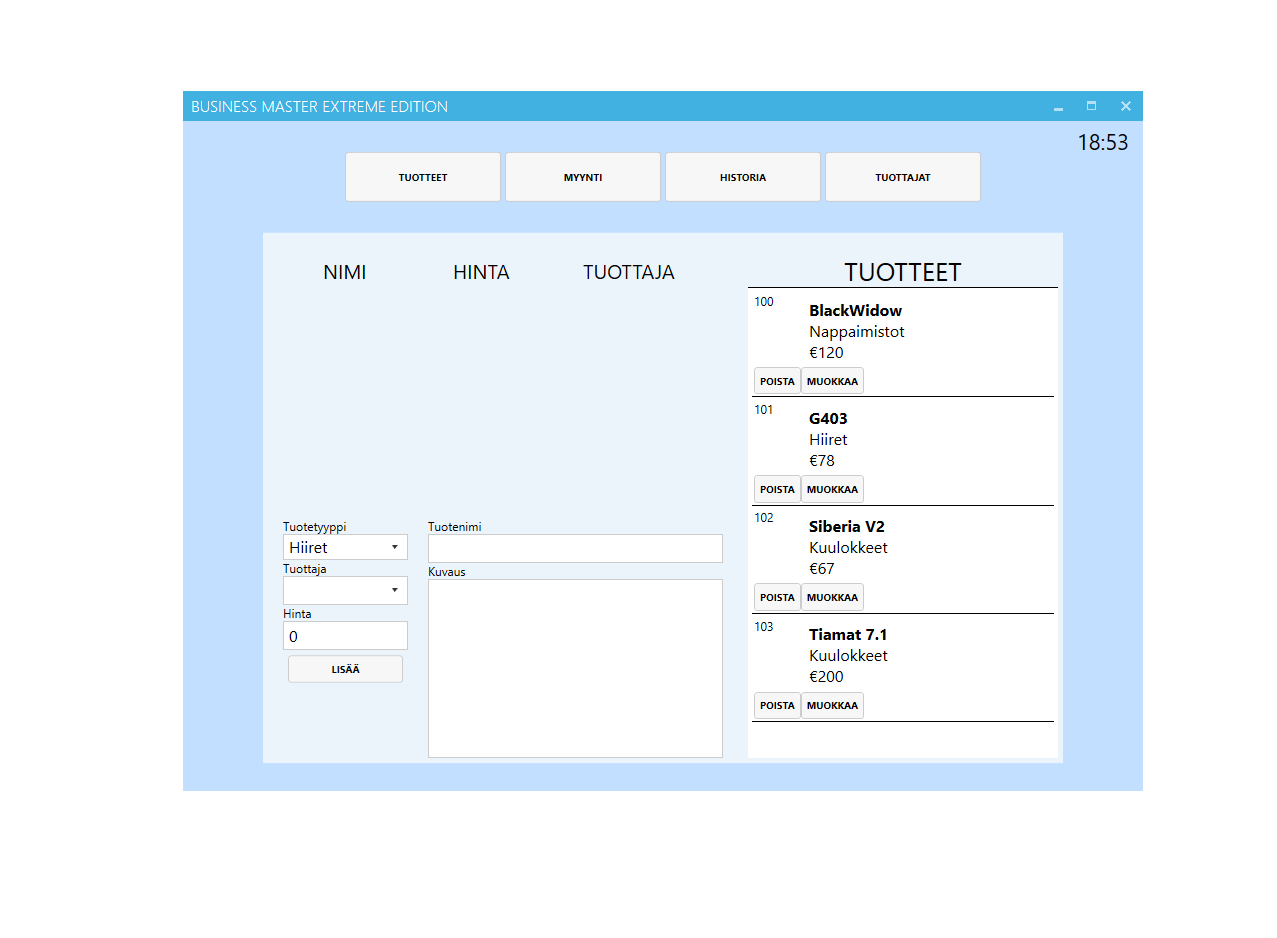


## Työaikaraportti

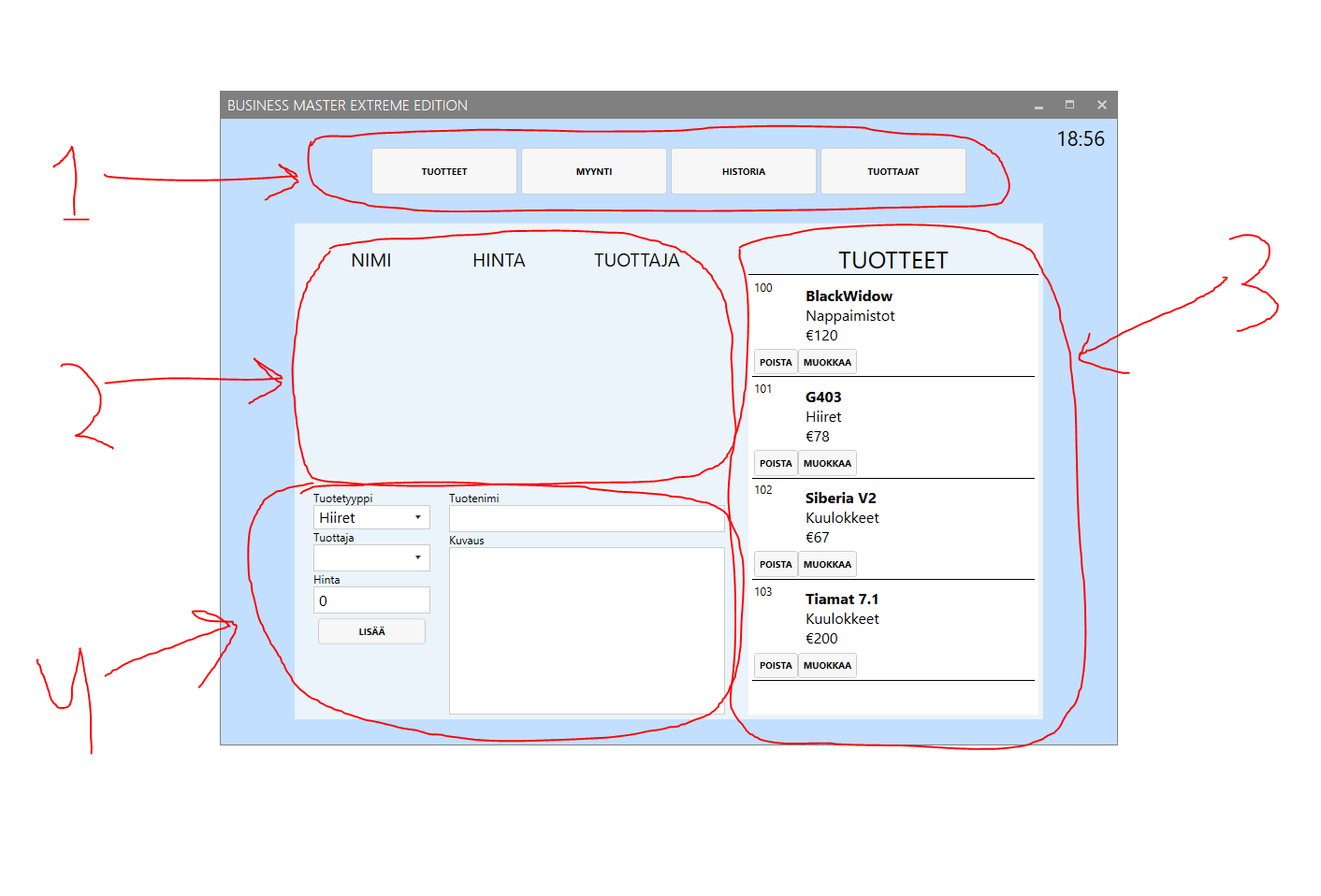
|  |  |  |  |
| --- | --- | --- | --- |
| Viikko | Tehtävä | Suun (h) | Vastuu |
| 9 | Suunnitelma | 5 | Kaikki |
| 10 | Prototyyppi | 5 | Kaikki |
| 11 | Model-luokat ja XAMLia | 10 | Roope |
|  | RelayCommand ja VIewModels | 10 | Yauheni |
| 12 | XAMLia | 10 | Roope |
|  | VIewModels + Tallennukset + JSON | 15 | Yauheni |
| 14 -15 | XAMLia + EnumBindingSourceExtension | 25 | Roope |
|  | ViewModels + XAMLia (Bindings) | 25 | Yauheni |
| 16 | Viimeinen testaus ja korjaus | 6 | Yauheni |
| 16 | Loppuraportti | 5 | Roope |
| 17 | Projektiesittely |  | Kaikki |

## Näyttökaappauksia projektista

**Yleisnäkymä ohjelmasta**

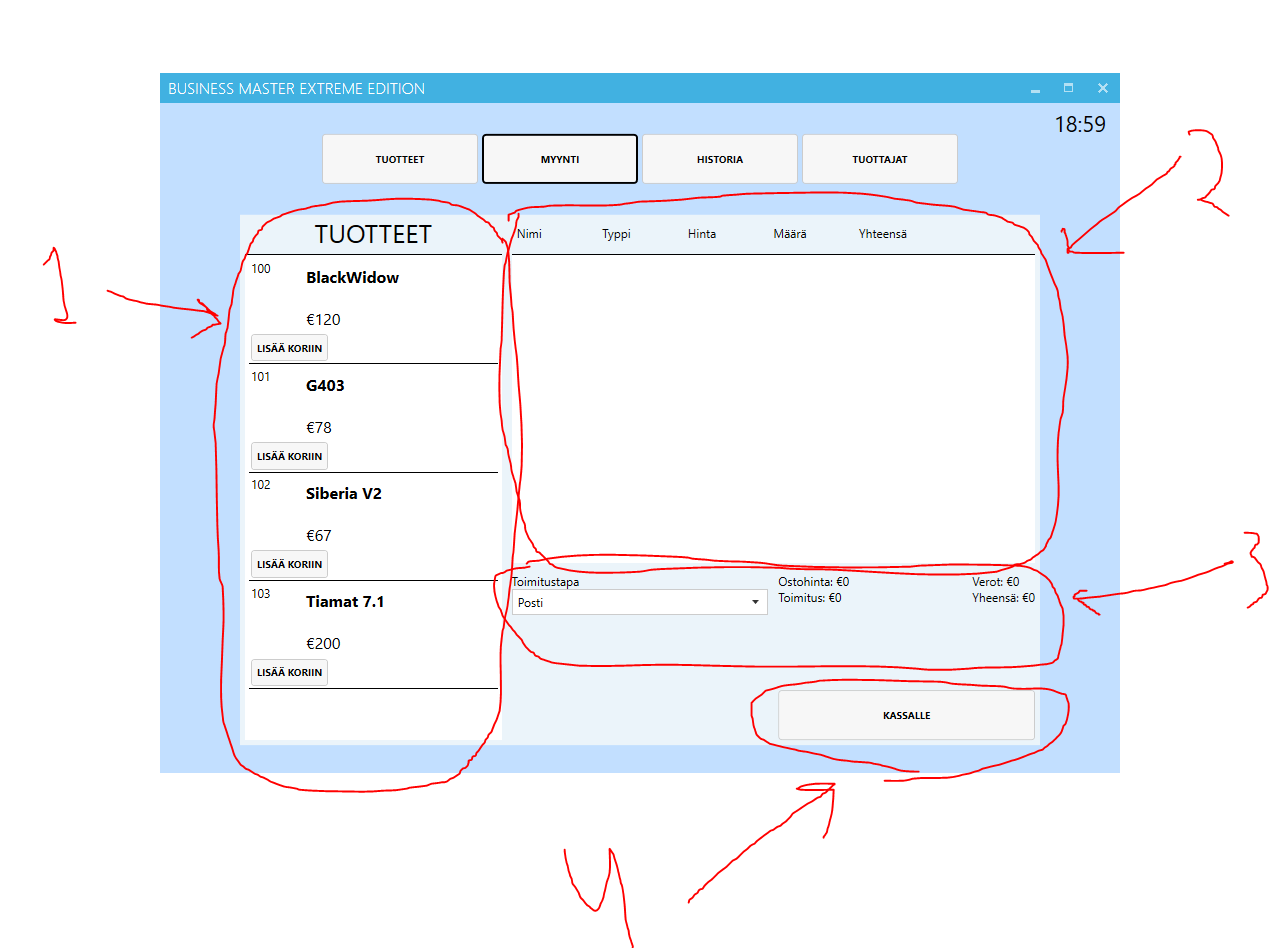


**Tuotteet-näkymä**



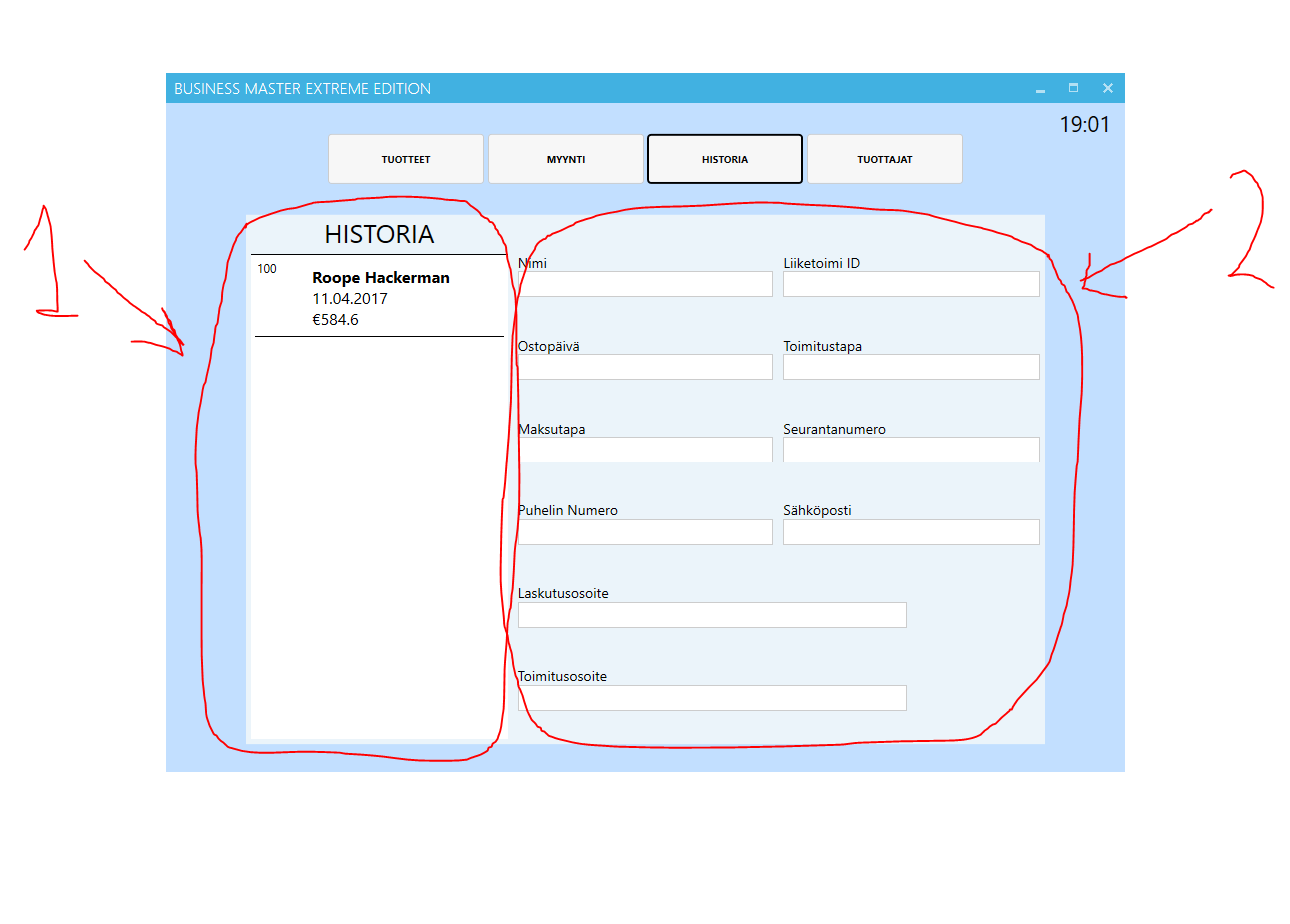
1. Ylhällä näkyvät painikkeita, joista voi siirtyä eri näkymiin
2. Tässä kohdassa näkyy valitun tuotteen tiedot, kun siitä clickaa.
3. Listaus kaikista tuotteista
4. Tuotteiden lisäys ja poisto kohta

**Myynti-näkymä**



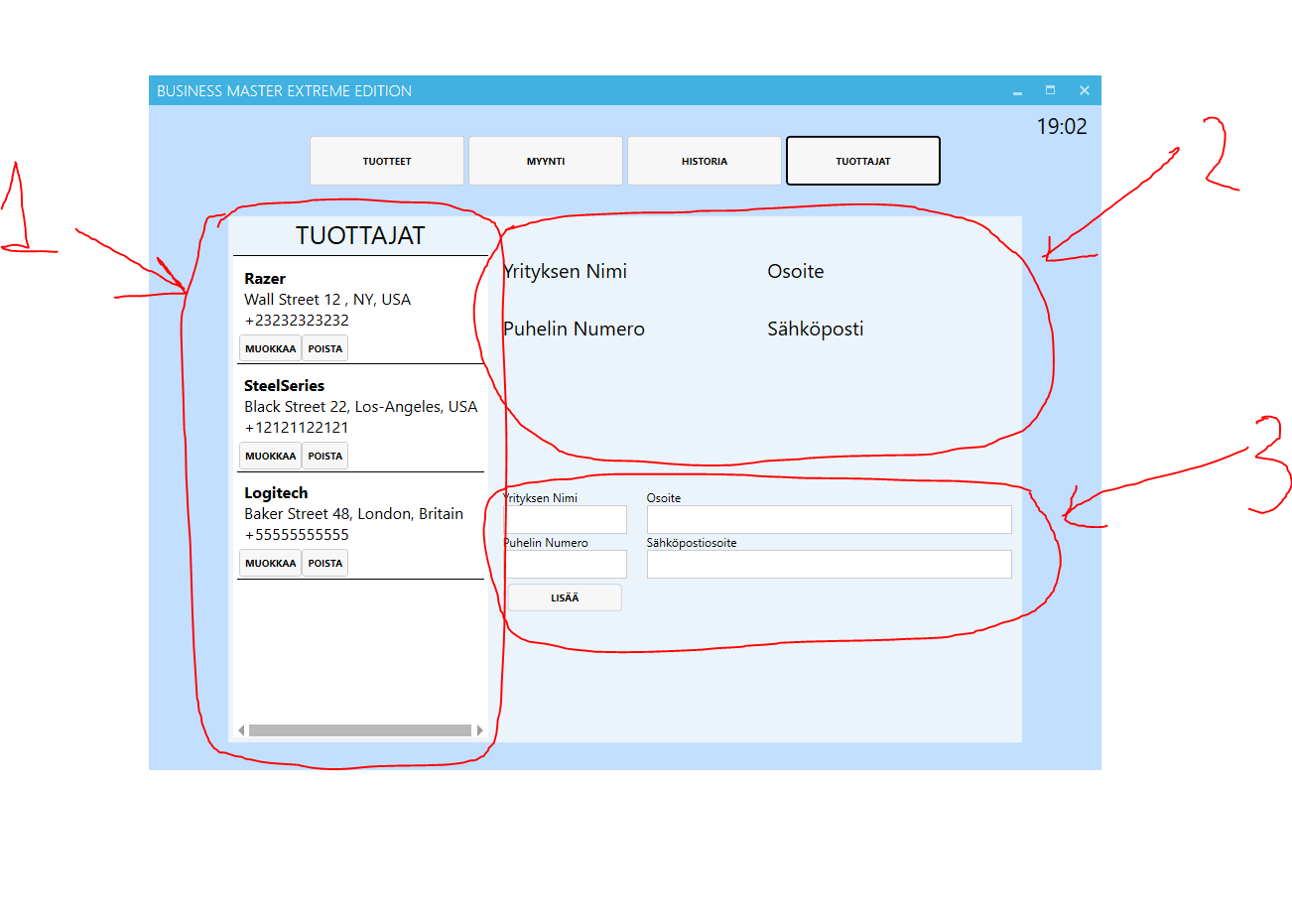
1. Listaus tuotteista
2. Ostoskori
3. Toimitustavan valinta ja ostoksen kustannus näkymä
4. Kassa painiketta

**Historia-näkymä**

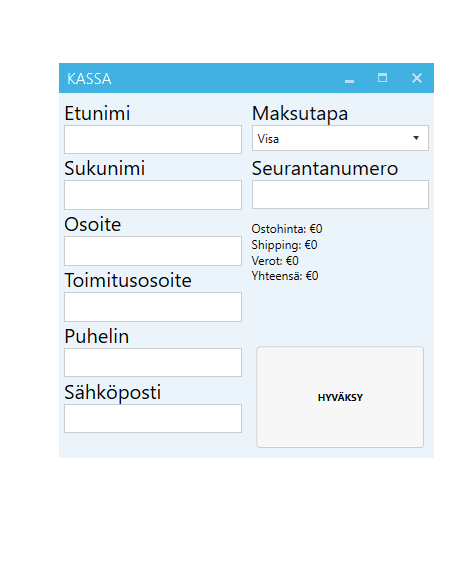


1. Ostohistoria
2. Tarkemmat tiedot osto tapahtumasta

**Tuottajat-näkymä**



1. Lista tuottajista
2. Näkymä tuottajan tiedoista
3. Tuottajien lisäys ja poisto

**Kassa-näkymä**

## Ongelmat

Projektin alussa aiheen keksiminen oli melko vaikeaa. Aluksi suunnittelimme ruokailua ja liikuntaa tarkkailevaa ohjelmaa, mutta se tuntui tuskalliselta tietokanta ohjelmalta, joten unohdimme sen. Sitten saimme ideaksi tehdä myynti sovelluksen, jota ryhdyimme kehittämään ja tämä idea tuntui mukavalta.

Seuraava ongelma meillä oli komentojen ja metodien lisäykseen liittyen. Googlaamisen jälkeen löysimme iCommandin, joka tuntui ratkaisulta ongelmaamme, sillä se helpotti komentojen ja metodien luomista valtavasti. Meillä meni hetki tuon iCommandin opetteluun, mutta koemme, että se silti nopeutti työtämme paljon.

## Testaus

Testasimme itse kaikkia ohjelmamme ominaisuuksia, emmekä löytäneet bugeja tai virhetoimintoja. Tuotteiden lisäys yms. yms. toimi kuten kuuluikin.

## Itsearviointi

Toivomme on arvosanaa 5. Ohjelmamme on moderni ja hyvin toimiva. Se sisältyy hyvin harjoitustyö aiheeseen. Näimme paljon aikaa ja vaivaa tämän työn eteen, jonka lopulta saimme toimimaan haluamallamme tavalla. Siinä on monia ominaisuuksia ja kaikki toimii hienosti.