Shopping Cart Project

By Alex Kohatzki

Price Compare – Final Project – CodeValue

Before I started work on this project, I thought how it will look, Design first. Finally, I decided using IDesign method. Main Idea:

UI 🡪 Managers 🡪 Engines 🡪 Accessors 🡪 xmlFiles/WebSite

all data passing by models and EventsArgs.

In **ShoppingCart** solution you can see **ShoppingCart** project, this project contains actually all business logic level, divided to parts:

* **Accessors**: this lower level in my project, all accessors responsible to access to outside world. in this project this world is .xml files and website (download file).
* **Engines**: this level is between Accessors and Managers. This is folder contains all engines, actually the idea that this files will do all dirty work. Folder contains:
  + RepositoryEngine – work on all data in the project, contains all action on the data.
  + UsersEngine – work on users, contains all action on users.
* **Managers**: this is top level in my project, that level connect between UI and logic.
* **Models**: to communicate between levels I used with models, kind of packages of data that I can set and send to other place in levels (or in same level).
* **Eventsrgs** : contains arguments to events that used in manager level (exposed to UI).
* **Interfaces**:
  + IManager – interface to manager’s level.
  + IRepositoryEngine – interface to engine that works with data.
  + IUserEngine - interface to engine that manage users.
  + IXmlAccessor – interface to accessors that works with xml files.
* **Resources**:
  + PathsInfo – contain static info: XmlPath, ImagePath
    - I wanted place all dynamic info in one place, that if some path will change in future its will be easy change it in this project.

I mapped products in xml file (ItemsList.xml), the program building list of products by taking keys(ItemKey) from this list and go to chain’s xml files and take actually info.

My ambition was to communicate(data) between logical and level and UI level just by events and events arguments, but finally I am decided to expose some properties too.

I also used serialization to save users and shopping cart in external files, I used .dat files, that can contains data in binary form or data form, I used binary.

Additional project in this solution it **ShoppingCart.WinFormUI**, this it UI level.

Bonus parts that I did:

* UI interface.
* Save and Load shopping cart in external file.
* Update data automatically in background, download and unzip files.
* Manage users and save their shopping cart.
* Export excel data file to shopping cart.