This document describes the **teamwork assignment** for Telerik Academy students **Desktop Applications with XAML and WPF** course.

# Project Description

Design and implement a desktop application using WPF or a RIA application using Silverlight. The application must have a server **REST API** written in **ASP.NET WebAPI** or **Telerik Everlive**.

## General Requirements

Please define and implement the following assets in your project:

### Requirements for the *REST API*

* Use **ASP.NET WebAPI or Telerik Everlive**
  + Expose **RESTful** endpoints
  + Host the application in **AppHarbor,** if using WebAPI
* The REST API should expose **only CRUD** operations over the database and **user login/register**
  + The business logic of the application **should be implemented on the SPA applications**

### Requirements for the *Desktop application*

* Use either **WPF** or **Silverlight**
* Use the **MVVM design pattern**
  + The application must be separated into consistent layers – **Views**, **ViewModels** and **Models**
* The application must have a usable UI
  + Not too much pretty, yet intuitive
* Use the **command pattern** for behavior binding
  + Event handlers for the UI controls must not be present
* The **application** must have **at** **least 5 different views**
  + Excluding **login**, **register** and **home** views
* The application must have at least 10 different ViewModels

## Additional Requirements

* Follow the **best practices for OO design**: use data encapsulation, use exception handling properly, use inheritance, abstraction and polymorphism properly and follow the principles of strong cohesion and loose coupling.
* Use a source control system by choice.

## Optional Requirements

If you have a chance, time and a suitable situation, you might add some of the following to your project:

* Usage of **message queues**
* **Unit** and **integration** testing

## Non-Required Work

* **Completely finished project** is not obligatory required. It will not be a big problem if your project is not completely finished or is not working greatly. This team work project is for educational purpose. Its main purpose it to experience **object-oriented modeling** and **OOP** in a real-world project and to get some experience in **team working** and team collaboration with TFS.

## Deliverables

Put the following in a **ZIP archive** and submit it (each team member submits the same file):

* The complete **source code**.
* Brief **documentation** of your project (2-3 pages). It should provide the following information (in brief):
  + Team name and list of team members
  + Project purpose – what problem do you solve?
  + Class diagram of your types
  + The URL of your SVN repository
  + Any other information (optionally)
* Optionally provide a **PowerPoint presentation** designed for the project defense.

## Public Project Defense

Each team will have to deliver a **public defense** of its work to the other students and trainers. You will have **only 5 minutes** for the following:

* **Demonstrate** the application (very shortly).
* Show the **class diagram** (just a glance).
* Show the **source code** in the **SVN-**web-based source code browser.
* Show the **commits logs** to confirm that team member have contributed.
* Optionally you might prepare a PowerPoint presentation (3-4 slides).

Please be **strict in timing**! Be **well prepared** for presenting maximum of your work for minimum time. Bring your own laptop. Test it preliminary with the multimedia projector. Open the project assets beforehand to save time. You have **5 minutes**, no more.

## Give Feedback about Your Teammates

You will be invited to **provide feedback** about all your teammates, their attitude to this project, their technical skills, their team working skills, their contribution to the project, etc. The feedback is important part of the project evaluation so **take it seriously** and be honest.