This document describes the **teamwork assignment** for Telerik Academy students **Web Services and Cloud Technologies** course.

# Project Description

Design and implement a web service application in the cloud and a JavaScript client application.

It could be one of the following:

* **Web chat** application
  + Users send messages between each other
  + Users can send files
  + Users can have a profile picture
  + Users receive notifications when another user sends them a message
* **Foursquare-like** application
  + Users can see a set of predefined places with coordinates
  + Users can check-in at a place near them
  + Users can post a comment about a place
  + Users can upload an image of the place
  + Users can create a place
  + Users receive notifications about people, checking in the place they are in
* **Image gallery** application
  + Users can own a gallery
  + The gallery can have albums
  + The albums can have sub albums
  + Users can upload images in the gallery or in any of the albums
  + Images have title
  + Users can leave a comment about an image
  + Users receive notifications when somebody comments an image of theirs
* **Chess** game
  + Users can join a random game
    - The engine decides which two players to start the game
  + Users can perform moves in a started game
  + Users can have a profile picture
  + Users receive notifications when a user in a game of theirs has made their move
* **Crowd-sourced** **news** application
  + Users can publish a news article containing images
  + Users can comment news articles
    - Comments can be nested
  + Users can vote for and against news articles
  + Users receive notifications when a new news article is published
* **Recipe** application
  + Users can upload recipes containing images and preparation steps
    - Preparation steps have completion time (e.g. bake potatoes for 5 minutes at 200 degrees)
  + Users can like and comment a recipe
  + Users can start cooking a recipe
    - Notifications are delivered when a preparation step's time has elapsed
* Another application by your choice
  + The only condition is to follow the Requirements

## General Requirements

Please define and implement the following assets in your project:

### Requirements for the *Web service application*

* Use **ASP.NET WebAPI**
  + Your application must be implemented using ASP.NET WebAPI
* Create **RESTful web services**
* Host the application in **AppHarbor**
* Use a **file storage cloud API**
  + **Dropbox**, **Google Drive** or other
* Use a **cloud database**
  + **MS SQL**, **MySQL**, **MongoDB**, **Redis** or other
* Implement push notifications
  + Use **PubNub**, **Urban Airship** or other

### Requirements for the *JavaScript client application*

* **Use jQuery** (DOM manipulation, AJAX, animations, etc…)
* **Implement a UI for your application** (use KendoUI, jQueryUI or implement your own UI logic)
* The **client application** should communicate with the **services application** using **HTTP** requests
* **The application must work in Google Chrome 26, Mozilla Firefox 21, Internet Explorer 9, Internet Explorer 10, Opera 12**

## 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
* **Backward compatibility** (make the application usable on browsers like IE8, IE7 and IE6)
* **Usage of a structural JavaScript framework**
  + Backbone.js, Ember.js, AngularJS or Knockout

## 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.

## Deliverables

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

* The complete **source code**.
* Add **webservices.teamwork@gmail.com** as a collaborator to your project.
* 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?
  + 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 **commits logs** to confirm that each team member has 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.