# CST8218 Assignment 2

This assignment is to be done in groups of 3 to 4 members all from the same lab section. If your group has 1 (just you) or 2 members, then send an email to your Lab Professor with the names of your members, and your Lab Professor will combine the smaller groups to form groups of 3 to 4.

Objective: Using a team member’s Assignment 1 solution as a starting point, implement the following requirements to achieve a full-featured mature solution. Note it is expected that you will make changes to the Entity class (Sprite.java), as well as any other project files needed to satisfy the requirements.

## Requirements:

* The RESTful interface works as specified in Assignment 1.
* Ensure your code has appropriate implementation comments.
* The project should be properly tested (unit and integration tests)
* Properly handle the updating of Sprites through the RESTful interface, such that a Sprite’s previous attribute values do not get zeroed-out when only other values are updated. It is expected you will change the Sprite Entity for this.
* Properly produce and consume both XML and JSON formats with the RESTful interface, including the **color** field (see hints below).
* Use appropriate HTTP response codes for REST calls. For example, if a sub-resource (**Sprite**) is created, a 201 HTTP response code (created) is appropriate. See the documentation for the **Response** and **Response.ResponseBuilder** classes and note you’ll need to use **Response** as the return type for your service methods.
* Add AJAX capability to appropriate input fields on JSF pages.
* Add users and authentication to the application, so that in order to use the JSF pages (not the RESTful interface), the user must have been added to the database with the right group. It’s preferable to merge an adapted solution to Lab4 into the project, but before you start any merging, make sure you store a copy of your project so you can revert back to the pre-merging version if the merging doesn’t go well. A fallback strategy is to leave Lab4 as a separate project, and simply adapt it and/or use it to authenticate for the Sprite application.
* As the **lowest** priority requirement, make your project unique by enhancing the program to make it behave, in any creative and small way, more like a game.

## Recommendations

* I recommend that you do your development in a virtual machine that you can take snapshots of, often, at least between steps. If your laptop is slow or for whatever reason you don’t want to work with virtual machine snapshots, be sure to make commits to a source code repository often, whether or not you’re using virtual machine snapshots.
* Dividing the tasks amongst the team is normal, but INTEGRATE your changes as EARLY as you can, and OFTEN.
* **Hints for dealing with Color objects in JSON and XML**: this is a similar concept to the Converter for Color in the JSF pages. Given enough time, you are (or should be) able to research this on your own, but the following information should speed up your research. We will use elements of the JAX-RS API.
  + We cannot add dependencies to a pom.xml file unless we rebuild the project as a Maven project, so instead of rebuilding we will simply add dependency jar files as libraries to the projects that need them. You can expect to need these jar files, which can be downloaded from a Maven repository
    - jackson-databind-2.9.8.jar
    - jackson-core-2.9.8.jar
    - jackson-annotations-2.9.8.jar
    - jersey-media-json-jackson-2.28.jar
  + JSON
    - You have an ApplicationConfig.java file in your project. Find that file, and compare it to the **Registering JackSon Modules** section of this web page: <https://cassiomolin.com/2016/08/10/using-jackson-as-json-provider-in-jersey-2x/>.
    - The annotations @JsonSerialize, @JsonDeserialize will end up in your Entity class. Study the API documentation for those annotations.
    - You’ll need to create some classes to implement @JsonSerializer<T> and @JsonDeserializer<T> (those classes you create need to be added to ApplicationConfig.java also).
  + XML
    - This page should help you a great deal <https://stackoverflow.com/questions/33055349/jaxb-xml-to-java-awt>
    - Note that **FontStyle** is their Entity class, and yours is **Sprite**
    - Note that you can test this REST work without creating something like the **App** class at the bottom of that answer.

## Submission

With all group members present and prepared to answer questions, demonstrate your final enhanced Sprite program to your Lab Instructor, and submit a zipped archive of the Netbeans project folder(s) using the Brightspace link provided. Please use only zip, and do NOT use .7zip, or .rar formats for the submitted archive.