Assignment 1

Personal planning, Vision and Project Plan

1DV600 Spring 2016

The main purpose of this assignment is to bootstrap the project. A good personal planning is vital in project and therefore you are going to do four smaller tasks and have the opportunity to reflect over you planning.

Remember that all assignments are individual!

Task 1 -- Personal Planning

When the client requests a list of books to present for the user it does the call **http://localhost:9090/api/books/** to the server and it expects the answer as a JSON object (an associative array). We are going to split the functionality into three tasks, but it is your task to plan these tasks. Take into account the time for learning and understanding of the problem when you plan the time. Make your planning with 15 minutes as the minimum unit.

Repeat the following pattern for all tasks (a, b, c):

- Plan
- Implement
- Reflect

Each subtask should be documented with at least 100 words.

Subtask A -- Books

The main objective of the subtask is to create a list of books and a function or method to get them. There are slight differences depending on which implementation you are using, either Java or Node.js but those differences will be clearly noted. Common for both is that they should handle books, and for each book we need the information *id*, *title*, *author*, *genre*, *publishDate* and *description*.

Java
Create a class in the package "models" that represents a book. After that, create a short list of fictive (or real) objects in the function **getBooks** that is available in **GetBooksResource**. When calling the URL http://localhost:9090/api/books the list of books should be outputted with **System.out.println**. The subtask is done when you see the objects in the terminal (where *vagrant* is run).

Node.js

Create an object in the folder “models” that represents a book. After that, create a short list of fictive (or real) objects in the anonymous function (or name it **getBooks**) that is available in **GetBooksResource**. When calling the URL http://localhost:9090/api/books the list of books should be outputted with **console.log**. The subtask is done when you see the objects in the terminal (where *vagrant* is run).

Subtask B -- JSON

Convert the objects created in subtask a into an JSON object and show it in the terminal using either **System.out.println** (Java) or **console.log** (Node.js).

Improvement Strategies

Choose two improvement strategies based on your reflections on subtask a and b. Describe what you have decided to improve and why. Implement your improvements in the next subtask.

Subtask C -- Web

In this subtask you are to answer the request in the web browser instead of printing it to the terminal. The subtask is done when you see the JSON object on screen. For inspiration, have a look at the **Ping** code that you find in the same folder as the book resource. If you follow the API for the model (as seen in **/api/books**), you will be able to show the books on the start page.

Task 2 -- Vision

Create a vision document for the system. This should be a document covering about half an A4 page describing the system. The purpose of the document is to make sure that everyone involved in the project has the same vision of what is to be created. Use the “Assignment Overview” and previous subtasks as your source for what to write.

In addition, write down your reflections on creating a vision document. This reflection should be about 100 words.

Task 3 -- Project Plan

Write a project plan for the project. This project plan should show the way to the complete and finished application, something that you should be able to follow. Write as much as possible in the project plan, add the milestones from Assignment Overview, and update the document throughout the course when you know more in the later assignments.

Again, as an addition, write down your reflections on creating a project plan. This reflection should be about 100 words.

Include in the submission

The following should be included in the submission that you hand in via Moodle:

- Vision document
- Project Plan
- Implementation (**/src** for Java and **/app** for Node.js)
- Report with all personal reflections
- Time log