FRAPPER
Colin

Software technology
1DV600

Assignment 1 - Personal planning, Vision and Project Plan

01/29/2017

Note : I'm sorry about my english, I'm an Erasmus student.

Task 1 - Personal Planning

Task : 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 subtasks (A, B, C):

- *Plan*
- *Implement*
- *Reflect*

Each subtask should be documented with at least 100 words.

Subtask A – Books

Task : 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.

1. Plan

So we have the choice of the implementation, I choose to use Node.js because I never worked with Node before (I already worked in javascript), so it can be interesting to learn how to work with Node and I know that you can have multiple way to implement the functions in Node.

Create an object in the folder "dao" 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).

How planning this task :

First, I need 15 minutes to understand the relation between all the files and how import/export modules works in Node.js.

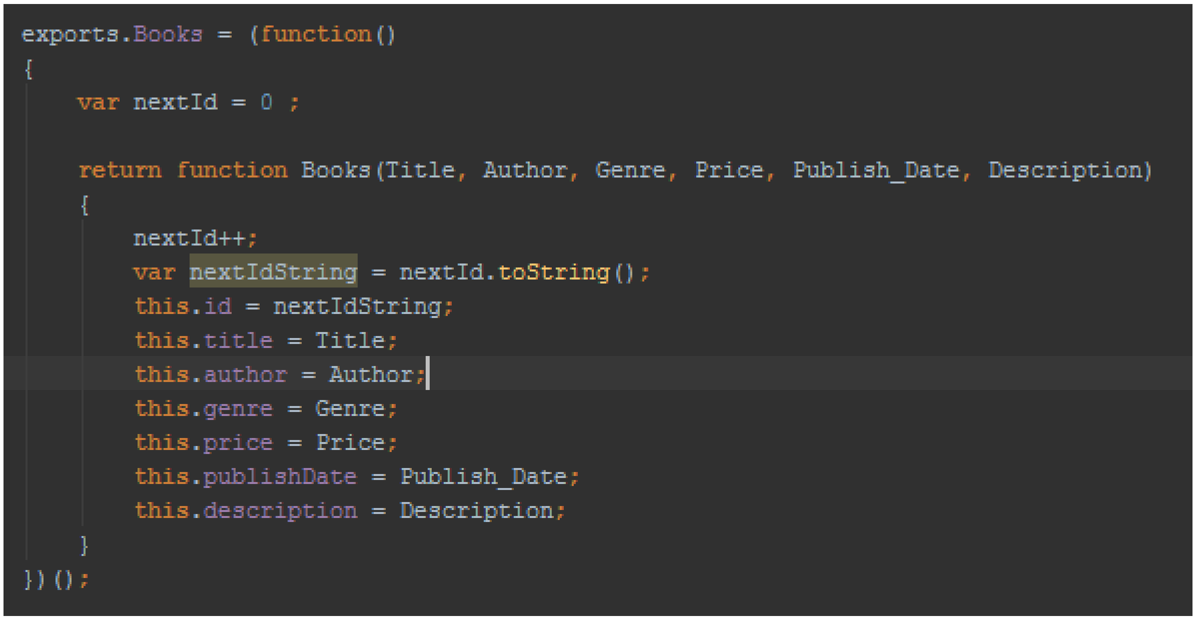
Then for create an object that represent a book, it only take few seconds, but there is multiple way to create an object and this is what I like in Node, so maybe I can take few minutes to think what is the best way to create an object and what is the best for this specially project.

The next step is to create a function getBooks available in GetBooksResource, where I can create a list of fictive objects (books). For this step, 5 minutes will be enough.

Total time for this subtask : 30 minutes.
I prefer say 30 minutes in case of some disagreement, it almost always happen and specially when you have never done this before, you must always think that it will not work perfectly the first time.

2. Implements

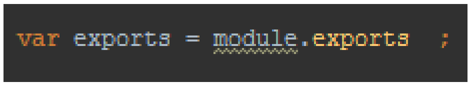
To create an object who represents a book, I created a function who work more or less like a constructor, here is the function :



```
exports.Books = (function()
{
    var nextId = 0 ;

    return function Books(Title, Author, Genre, Price, Publish_Date, Description)
    {
        nextId++;
        var nextIdString = nextId.toString();
        this.id = nextIdString;
        this.title = Title;
        this.author = Author;
        this.genre = Genre;
        this.price = Price;
        this.publishDate = Publish_Date;
        this.description = Description;
    }
})();
```

Explication : It's like a constructor in java, I also added something like the auto_increment for the database, every time there is a new book, it will automatically increment the id, I think it's very important especially later when people will want add a book or delete it. Even when the people will add a book they don't wanna put a special id himself. What I mean is that every task who can be automatic must be implemented.

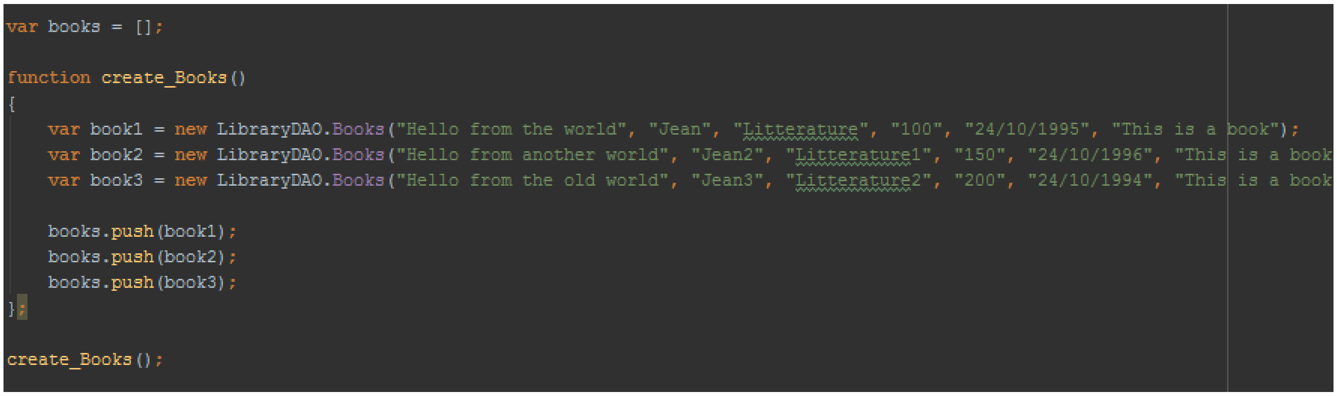
To understand why I have a export.Books, here is the explanation :



```
var exports = module.exports  ;
```

The think with the method I implemented is that it's gonna more easy and faster I just need to add “exports.TheNameOfTheMethod” and I will be able to use it in another file.

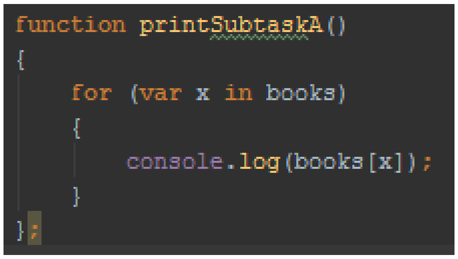
The next step was to create the function getBooks and create a list of books.
Here is the function :



```
var books = [];

function create_Books()
{
    var book1 = new LibraryDAO.Books("Hello from the world", "Jean", "Litterature", "100", "24/10/1995", "This is a book");
    var book2 = new LibraryDAO.Books("Hello from another world", "Jean2", "Litterature1", "150", "24/10/1996", "This is a book
    var book3 = new LibraryDAO.Books("Hello from the old world", "Jean3", "Litterature2", "200", "24/10/1994", "This is a book

    books.push(book1);
    books.push(book2);
    books.push(book3);
};

create_Books();
```

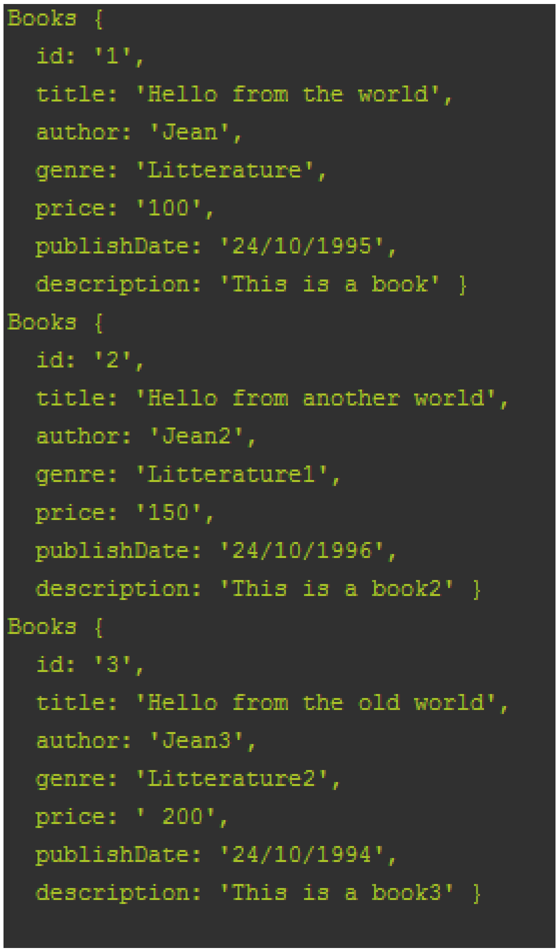
Explication : I created a list as a global variable. In the function create books, I can create many books as I want, then I insert them in the list books. I prefer to create a global variable because I will use it in other functions.

I also created another function to output the books with console.log



```
function printSubtaskA()
{
    for (var x in books)
    {
        console.log(books[x]);
    }
};
```

Here is the result in the terminal where vagrant is running :



```
Books {
  id: '1',
  title: 'Hello from the world',
  author: 'Jean',
  genre: 'Litterature',
  price: '100',
  publishDate: '24/10/1995',
  description: 'This is a book' }
Books {
  id: '2',
  title: 'Hello from another world',
  author: 'Jean2',
  genre: 'Litterature1',
  price: '150',
  publishDate: '24/10/1996',
  description: 'This is a book2' }
Books {
  id: '3',
  title: 'Hello from the old world',
  author: 'Jean3',
  genre: 'Litterature2',
  price: ' 200',
  publishDate: '24/10/1994',
  description: 'This is a book3' }
```

3. Reflect

When you finished the implementation, you must always look at the requirement specification that you made before (here is the plan), so it took me a little bit more than 15 minutes to understand the all code, like how export the method from one file to another, when methods/functions are called when you tape *http://localhost:9090/api/books* as an URL.

For the rest of the step, I was quite right in the requirement specification, so it took me a little bit more than 30 minutes, and this is the main “problem” when you making the requirement specification (plan) and you never did this kind of exercices before, you don't really know how long it will take, that's why for example, the manager or the person who made the requirement/plan must always ask to the people who already made this task because only them know how long it will take.

Subtask B – JSON

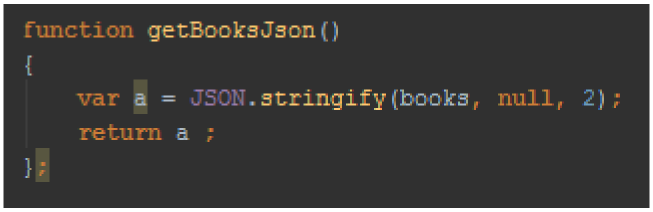
Task : Convert the objects created in subtask a into an JSON object and show it in the terminal using either System.out.println (Java) or console.log (Node.js).

1. Plan

To be honestn I suppose it will take 10 minutes, we have to convert all the books in a JSON format and there is already a function ho made this in javascript (JSON.stringtify).

2. Implement

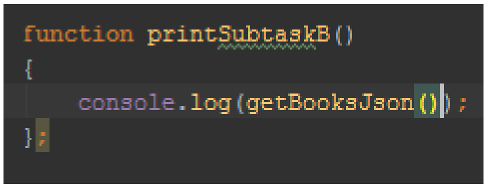
I also create a function getBooksJson, who gonna use the function getBooks and convert it into a JSON format, here is the function :



```
function getBooksJson()
{
    var a = JSON.stringify(books, null, 2);
    return a ;
};
```

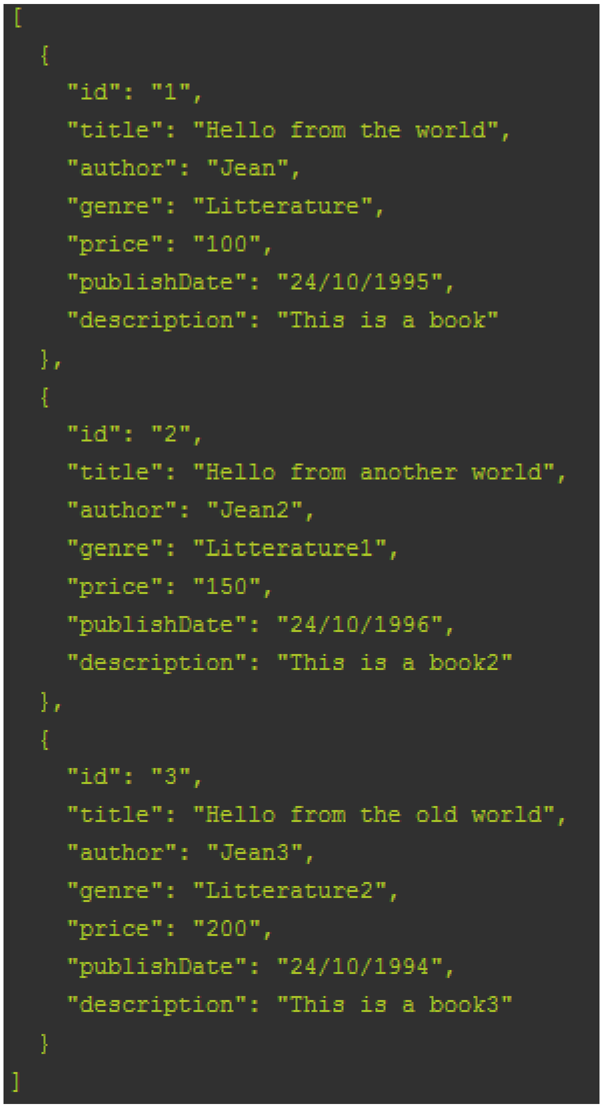
Explication : I use the method stringtify who convert an object into a JSON format, the paramater null and 2 is just to have a better looking when it's output to the console.

Same as the substackA, I have a method to output the books on JSON format.



```
function printSubtaskB()
{
    console.log(getBooksJson());
};
```

Here is the result in the terminal where vagrant is running :



```
[
  {
    "id": "1",
    "title": "Hello from the world",
    "author": "Jean",
    "genre": "Litterature",
    "price": "100",
    "publishDate": "24/10/1995",
    "description": "This is a book"
  },
  {
    "id": "2",
    "title": "Hello from another world",
    "author": "Jean2",
    "genre": "Litterature1",
    "price": "150",
    "publishDate": "24/10/1996",
    "description": "This is a book2"
  },
  {
    "id": "3",
    "title": "Hello from the old world",
    "author": "Jean3",
    "genre": "Litterature2",
    "price": "200",
    "publishDate": "24/10/1994",
    "description": "This is a book3"
  }
]
```

3. Reflect

There is not so much to say about this task, it took me few minutes to did it, I took maybe few minutes to think about if I will convert each books, or just the list who contains the books, I choose to convert the list himself because it's more “efficient coding”.

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.

First strategie : I wanna change the code about how the exports is made, with the code that I got you can only exports one function, so I will (and already) change it so I can export multiple function and then choose which function I will use.
Second strategie : I think the function to create the books doesn't need to be in the file, normally this

file (getBooksRessource) is only here to print and get the books, you don't have to create the books in this class, it's better to have another file more like a class call Books with getters and setters (I think it's possible in Node.js) who will return the list of the books.
I will not implement it in this assignment because I ran out of time for this one and I don't wanna take bad decision right now I need to think more about. (I will try to implement after the deadline of the first assignment).

Subtask C – Web

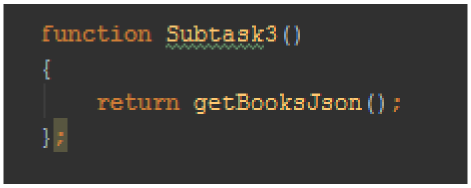
Task : In this subtask you are to answer the request in the web browser instead of printing it to the terminal. The subtask is done when you see the JSON object on screen. For inspiration, have a look at the Pingcode that you find in the same folder as the book resource. If you follow the API for the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

1. Plan

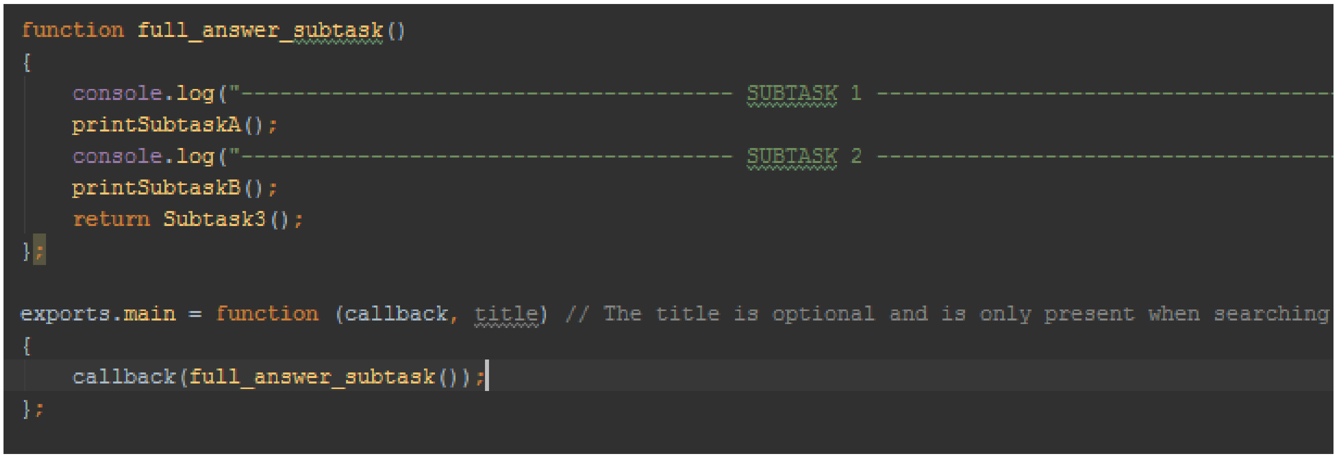
First in the task, it said that we need to have a look at the Ping code, and understand it. Then after we have to implement in sort of we can see JSON object on screen. I would say 10 minutes for task.

2. Implement

After looking the ping code, I saw that to print on the screen, you just need to use the function callback to print a JSON obejct on the screen, so I used the function getBooksJson who return the list of the book in a JSON format.

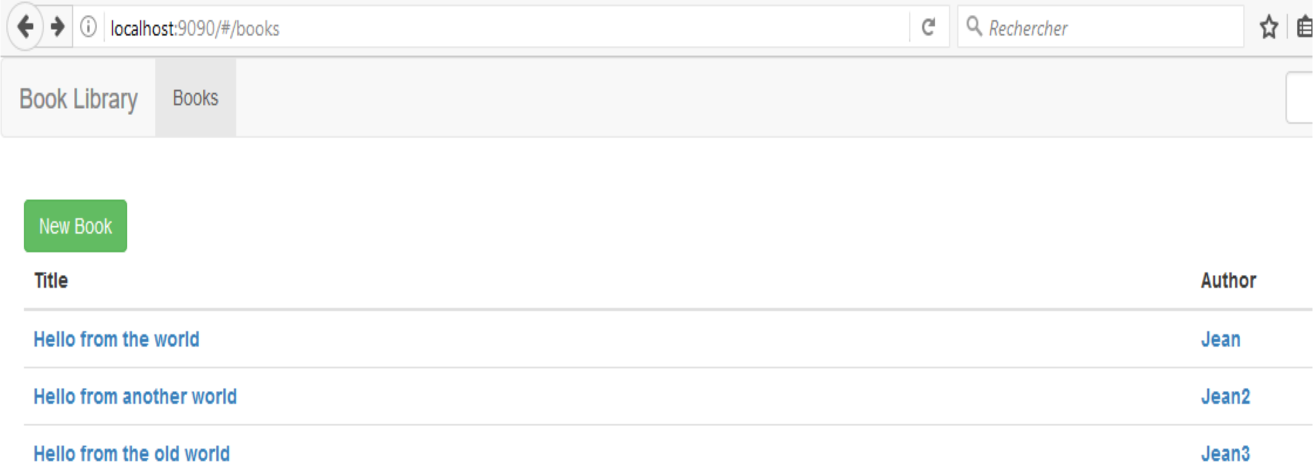


```
function Subtask3()
{
    return getBooksJson();
};
```



```
function full_answer_subtask()
{
    console.log("-------------------------------------- SUBTASK 1 -----------------------------------
    printSubtaskA();
    console.log("-------------------------------------- SUBTASK 2 -----------------------------------
    printSubtaskB();
    return Subtask3();
};

exports.main = function (callback, title) // The title is optional and is only present when searching
{
    callback(full_answer_subtask());
};
```

Here is the result :



Title	Author
Hello from the world	Jean
Hello from another world	Jean2
Hello from the old world	Jean3

3. Reflect

Again I don't really know what to say, the task took me 10 minutes to understand the ping code and test it on a web browser by taping *http://localhost:9090/api/ping* and see the JSON object. So I did the same by using the function callback , and when I tape *http://localhost:9090/api/books or* http://localhost:9090/#/books I am able to see the JSON object.

Task 2 - Vision

Task : 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.

You can find the Vision document separate from the report.

Reflect :
The question is why is good to make a Vision document and in what it will be useful ?

First it's always good to remember the basics and the objectives when you develop a project, it's the best way to lead to a great project, with the vision document you know where you're going, with who you are working, what goals the customers/end-users wants, so you can refer it when you are developping your code, it's like the beggining of the project. It's a big picture about where we should go, what you should do. With a good vision document, you should be able to see the constrain and try to avoid them. It communicates the fundamental "why and what" for the project and is a "test" against which all future decisions should be validated. When you wanna make some features to your project, you must look the vision document and see if the features correspond on what the end-user wants and if it's gonna be useful to them or for your team. You will be able to avoid any confusion. The goal of the vision document is to keep everyone who is involved in the development of the product focused on the direction that the product should be taken into.

To conclude, I would say that the vision document is a communication tool. The vision document is a reference point, something you can look back to and determine whether or not your efforts are going to the right place.

Task 2 – Project Plan

Task : 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.

You can find the project plan separate from the report.

Reflect :

It was extremly interresting to make a project plan, especially with argile. I learn a lot about this and I can understand how it can be very useful when you're building a project.
A project plan is here to write the step you need to follow to arrive to the final application/project. It's very important to have a good project plan so you know that you won't missunderstood somes task/requirement, and to be in accord with the customers and THIS is the more important, be in confidence with the customers. What I like with agile is that even if the requirements change (and they will, as I said in the report in the conclusion it is impossible to gather all the requirements at the beginning of a project), you will be able to make change those requirements. Requirements evolves but the timescale is fixed. Agile provide a lot of flexibility.

“Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.
It works by breaking projects down into little bits of user functionality called user stories, prioritizing them, and then continuously delivering them in short two week cycles called iterations.”.

Personnally I really liked to make this project plan but to be honest I don't think it's gonna fully exploited in this project because first I'm working alone, and I think agile is more efficient when you are working in a team, besides here we don't have customers we are ourself the developers/end users/customers. And I'm wondering if I'm gonna be able to see the changes/requirements I should make to lead this project and this is because as I said I'm alone and I'm in all the part of this project (developers, customers, ect). Anyway maybe it wasn't the best idea to choose agile as a software development, but I think I'm gonna learn a lot with this method and this is the main reason why I choose it.