



**Linnéuniversitetet**

Kalmar Vaxjö

Report

# Vision Document

*Refactored*



*Author:* Helena Tevar Hernandez

*Handledare:* Jesper Andersson

*Semester:* Spring 2017

*Discipline:* Software Technologies

*Course code:* DV600

# Table of Contents

<b>1 Vision</b>	<b>2</b>
1.1 Purpose	2
1.2 References	2
1.3 Overview	2
1.4 Stakeholders	2
<b>2 Project Plan</b>	<b>4</b>
2.1 Assignment 1	4
2.1.1 Goals	4
2.1.2 Planning	4
2.1.3 Time log	5
2.1.4 Reflections	5
2.2 Assignment 2	6
2.2.1 goals	6
2.2.2 Planning	6
2.2.3 Time log	6
2.2.4 Reflections	6
2.3 Assignment 3	7
2.3.1 Goals	7
2.3.2 Planning	7
2.3.3 Time log	8
2.3.4 Reflections	8
2.4 Assignment 4	8
2.4.1 Goals	8
2.4.2 Planning	8
2.4.3 Time Log	9
2.4.4 Reflections	10

# 1 Vision

The purpose of this document is to provide an overview of the project included in the subject “Software Technologies”.

## 1.1 Purpose

The purpose of the project is to implement a half-finished back-end library to complete a group of user cases and the tests needed to ensure the functionality of the system. The project will include documentation and UML diagrams to explain the process of developing.

## 1.2 References

This document has been modified and based in the vision document template gave by IBM Knowledge Center.

## 1.3 Overview

The purpose of this document is to define the features of the assignment for the subject “Software Technologies”, included in the Linné University program also called “Software Technology”. The main assignment is divided in four assignments that are worked one over each other. The objective of this assignment is to learn the process of a software project.

The assignment consists on completing a half-finished library web application: a finished front end that students can use to verify that everything done during the assignments works, and a module partially finished that needs to be completed. In this case, the module will be back-end code written in Java. The completed web will need to be able to add, modify and delete books, view the data of a book, view a list of books and search books by author or title. These features will include documentation and implementation. To accomplish these goals the students will learn through lectures, workshops, Q/A sessions and will be guided by the teacher’s assistants.

The first assignment consists of four subtasks that will help the students to understand the basics of project planning and documentation while starting the Java to Json implementation. The second assignment will work with analysis, design and implementation using diagrams from UML, primarily use case and sequence diagrams. The third assignment will teach students to perform test on systems. The fourth and last assignment will complete the system, repeating the previous processes with the remaining use cases.

## 1.4 Stakeholders

This assignment is single based and for this reason it only has a stakeholder.

Name: student Helena Tevar.

Responsibilities: the student has all the responsibility over this assignment.

Type: Student first year.

Success criteria: Receiving a grade of 2 or higher.

Needs and risks:

1. This stakeholder has a high risk of not being able of complete the project because of health problems.

- The student is pregnant during the course and that may lead to possible health difficulties that make this project to fail.
2. This stakeholder has little or none experience with the environment or languages.
- The student has not programmed in this level before, so she will need time to learn and it may affect the schedule.

## 2 Project Plan

Each assignment of the course would be an iteration of the project. It will include the goals or requirements of each assignment, planning and a time log. A personal reflection will be included at the end of each iteration. The time log will be presented by hours. This type of planning gives freedom to organize the time during the day, expecting difficulties with other subjects and personal problems that would affect a fixed schedule.

### 2.1 Assignment 1

During this assignment, we estimate the risks associated with the people in the development team as well as the risk they come from the software tools and other support software used to develop the system. The students may be unavailable or not enough trained for the software used during this assignment.

#### 2.1.1 Goals

The goals of this assignment are:

- Answer a JSON string from a list of books when requested from <http://localhost:9090/api/books/>
- Documentation. Create a vision document, project plan, time log and personal reflections.

#### 2.1.2 Planning

This assignment is divided in three subtask and documentation. Each subtask works in a short feature that should be planned, implemented (in the /src folder) and documented.

##### 2.1.2.1 Subtask A

The objective of this subtask is to modify the class “book” given in the package “models”, create a book list in the function “getBooks” inside “GetBooksResource”. When calling the URL <http://localhost:9090/api/books> the list should be outputted in the PowerShell console.

To complete this subtask, it is needed a basic knowledge in Java, to modify and create the variables needed as well as the function.

This subtask can be summarized in the next steps.

- Check how many variables are needed by the front-end.
- Create the variables inside object package `lnu.models – book.java`.
- Create or modify the method `toString()` if needed to output a string.
- Declare several books inside `GetBooksResource ()`.
- Declare a list and add to the list the books created before.
- Print the book’s list through the PowerShell console.

##### 2.1.2.2 Subtask B

The objective of this subtask is to convert the books created the step before into an JSON object and print them in the terminal.

This subtask can be summarized in the next steps.

- Search information about Java object to JSON and learn the ways of implement JSON in our system.

- Create a JSON object for each book in the list.
- Print out the JSON strings created in the console.

### 2.1.2.3 Subtask C

This subtask objective is to show the JSON objects created before printed in the web browser instead of in the terminal.

This subtask would take two steps: check the api specification about `get/api/books` and return a JSON string in `getBooks()`. This last part of the assignment should be easier due to the extend documentation given in the subtask B.

### 2.1.3 Time log

	Planned hours	Real hours	Difference
<i>Assignment 2 weeks 50%</i>	<i>40</i>		
Lectures and Workshops	6	6	0
Lecture: Installation	2	4	2
Subtask A: Learning	1	2	1
Subtask A: Code + Test	1	1	0
Total S.A.	2	3	1
Subtask B: Learning	2	8	6
Subtask B: Code + Test	3	8	5
Total S.B.	5	16	11
Subtask C: Learning	2	2	0
Subtask C: Code + Test	3	1	-2
Total S.C.	5	3	-2
Vision document	4	6	2
Total	24	38	14

### 2.1.4 Reflections

The lack of experience with JSON and Java made continuous loops of mistakes that were difficult to understand and solve. During the subtask, A, an error when the variables were declared was made that broke the possibility of continuing this subtask. During this time, the console would show different errors but only after talking with the teacher's assistants Tobias, the subtask A error was noticed. The exercise had three possible solutions: to make the ObjectMapper able to reach private variables from the class books, add getters to the class book or change the variables to public (not recommended, because it would break the encapsulation). After adding getters to the class book, the subtask worked without know problems, thanks to the Java to JSON code was fine.

During the process of subtask A and B was clear that improvements needed to be done in relation of testing. Independent testing is not reliable. The tendency of test what is needed and not a white-box test, where every part of the code is tested, may throw

errors. In this case, a simple basic error was not checked because the situation did not require it, so it went to the next phase unnoticed.

## 2.2 Assignment 2

This assignment goes further in the planning phases analysis, design and implementation. The UML gains more weight in this assignment and it will be presented in the appendix.

### 2.2.1 goals

The goals of this assignment are:

- UML (Identify use cases, use cases realization and robustness diagrams)
- Fetch books from the XML file supplied.

### 2.2.2 Planning

Steps to follow during this assignment:

- Research and learn about UML way of documenting<sup>1</sup>
- Identify and document the use cases used in the system
- Gather information about XML to Java objects to JSON
- Show the solution in UML diagrams
- Implement it with a given XML file
- Implement “Delete Book” without design.

### 2.2.3 Time log

	Planned hours	Real hours	Difference
<i>Assignment 2 weeks 50%</i>	<i>40</i>		
Lectures and Workshops	8	8	0
Task 1: Learning	4	6	2
Subtask A: Identify Use Cases	3	3	0
Subtask B: Robustness Diagrams	3	3	0
Subtask C: Use Case Realization	3	3	0
Task 2: Design	5	5	0
Task 3: Implementation	4	6	2
Total	30	34	4

### 2.2.4 Reflections

The robustness diagrams help to understand the sequence of the use case in a simple way, that will be expanded in the sequence diagram. This was useful when working with the use case realization, because the objects, boundaries and processes were

---

<sup>1</sup> See document “UML”

already explained in this diagram. As a personal preference, I would like to add arrows to clarify the direction of the action.

The sequence diagram is useful before programming. It is clear and concise about all the objects and methods that may be used. This diagram helps to see the order of the interactions that a programmer needs to code. Working with sequence diagrams may be useful when working in groups, because it documents step by step the process of the use case.

The implementation of task 2 was completed with delays but successfully. After finishing task 2, I began with task 3 without planning or creating any kind of diagram. My opinion after finishing task 3 is that diagrams improve the job. During this task, I ended up writing my own diagrams in paper to divide the user case in smaller pieces of code.

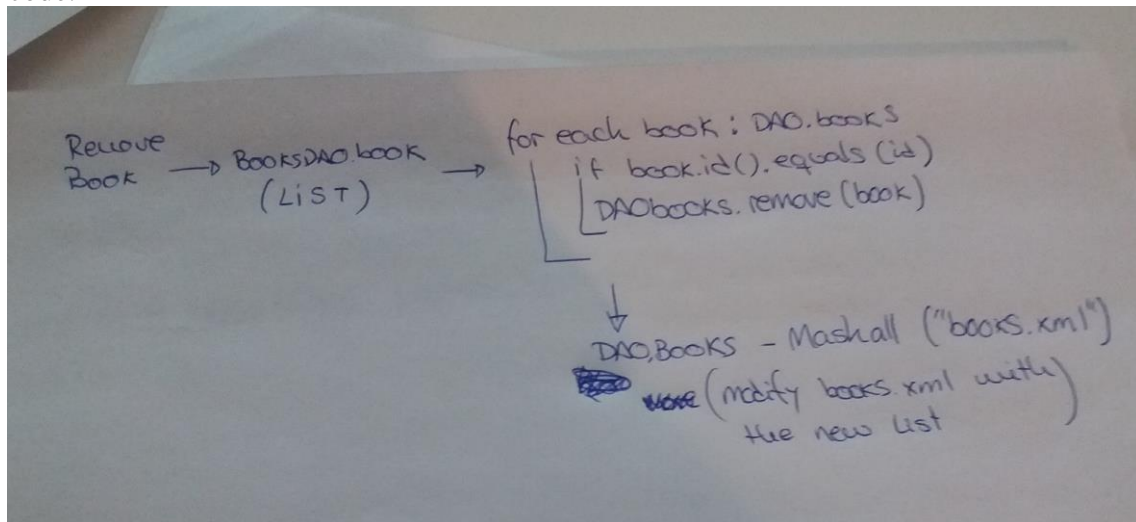


Figure 1- Handmade diagram

## 2.3 Assignment 3

This assignment focusses on performing tests to ensure the reliability of the system.

### 2.3.1 Goals

The goals of this assignment are:

- Create a test plan.
- Test two use cases.
- Create two Junit tests and eventually to all the API resources.

### 2.3.2 Planning

Steps to follow during this assignment:

- Research and learn more about Junit and Mockito.
- Create a test plan for each use case.
- Implement the Junit tests.



### 2.3.3 Time log

This assignment was not delivered during the course, and so, the time log failed. It is explained during the reflection

	Planned hours	Real hours	Difference
<i>Assignment 3 weeks 50%</i>	<i>40</i>		
Lectures and Workshops	8		8
Learning about Junit	4		
Test cases two use cases	4		5
Implement unit test	6		8
Reflection	2		
Total	22		

### 2.3.4 Reflections

During the vision document, I specified the possibility that the stockholder, or in other words myself, may become sick. Unfortunately, that risk happened and I was unable to finish this assignment. After being sick for two weeks and away from any kind of work or study by medical recommendation, I was forced to give birth to help me heal. So, when I recovered, I had to take parental leave for six months. During the summer, with the motivation of having pass the home test and that the library worked in the use case “get books” and “add books”, I decided that I will try to finish the project in the best way I could if with that I still had a chance to passing the subject.

From now on until the assignment 4, I have not planned hours of work because it would be useless but registered the ones I did work. I will keep working in the project when I the new CEO of the family allows me until the last day to upload the assignment.

## 2.4 Assignment 4

This assignment works the iterative process of the project. The aim is to iterate through the rest of the use cases left and repeat the UML documentation, implementation and testing.

### 2.4.1 Goals

The goals of this assignment are implement and test the requirements “add book” and “modify book” following the steps used during the assignments 1, 2 and 3.

### 2.4.2 Planning

During this assignment, we will repeat the process done before with the use cases “add book” and “modify book”.

- Add book
  - Use UML to plan the implementation.

- Implement in the project.
- Design test.
- Implement test.
- Modify book
  - Use UML to plan the implementation.
  - Implement in the project.
  - Design test.
  - Implement test.
- Reflections

### 2.4.3 Time Log

	Planned hours	Real hours	Difference
<i>Assignment 4</i>			
<b>“Addbook” case</b>			
Create diagrams		3	3
Implement		12	12
Create test plan		2	2
Implement test		5	5
<b>“EditBook” case</b>			
Create diagrams		3	3
Implement		10	10
Create test plan		2	2
Implement test		5	5
<b>Resources</b>			
Create test plan		4	4
Implement		20	20
<b>Total</b>			<b>66</b>

#### 2.4.4 Reflections

During this summer, I was not able to work properly because I was full time in parental leave. It was impossible to spend time in front of a computer but I had time to read about programming<sup>2</sup>. Because of that I decided to reformat a little bit of the code when I could, specially the catalog. I choosed to give more responsibility to the catalog so it would handle internally all the use cases. The biggest problem I got was when trying to test exceptions. I used a response 404 for the resources so it was clear, but I got a lot of problems when testing exceptions in models and DAO and an unexpected action with the XML.

The perfect project will need a new iteration to work with:

- More refactoring.
  - At some point, the backend rewrites the XML two times. It happens each time we try to make an XML from a catalog.
- Finish the Junit API test.
  - Finish the junit tests with exceptions.

---

<sup>2</sup> Books I could check during summer:  
Jeff Atwood, Effective Programming.  
Robin C. Martin, Clean Code.

