Application Development MCOMD1ADC

Dr. Alexios Louridas

2021-01-14

Contents

Intorduction

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

Chapter 1

Recap

1.1 Module Structure

1.2 Revision

This lab will go back and revise on certain areas that would help us understand the module this year. Looking back to the

1.3 Software Development

1.3.1 Software Development Lifecycle

1.3.2 Intellectual Property

1.3.3 Referencing

1.4 Version Control

While developing software, development teams use tools to help them reduce development times and produce more stable and better solutions. The most popular of these are version control or source control tools. The aim of these tools is to keep a record of all changes that happen on the source code of an application or applications being developed.

A version control tool keeps a record of all changes while recording the person or team responsible, the time of a change and the content that has been altered. All of these are recorded in a specialised database usually called a repository or repo for short, that can be accessed at any point to review code and provide a fix or an addition to the source code by identifying the best place, time and/or person to do so.

Try and remember how many times have you lost work that you wish you have backed up or you have but you do not remember where it is located, or you have not backed up as frequently as you would have wanted with the specific change you are looking. All of these cases would disappear or be minimised by using a version control tool and having a repository to store all your changes.

In this module you will learn how to use one of the most popular version control tool. Before we go on to play and start creating or obtaining repositories let us see the types of repositories that exist.

1.4.1 Client based model

A single repository exists that users connect and exchange information. The repository is centralised and all repository functions are done centrally. The users connect and obtain a working copy of the repo. The users have the ability to change the source files but any changes to the repository would need to be communicated directly to the central repository.

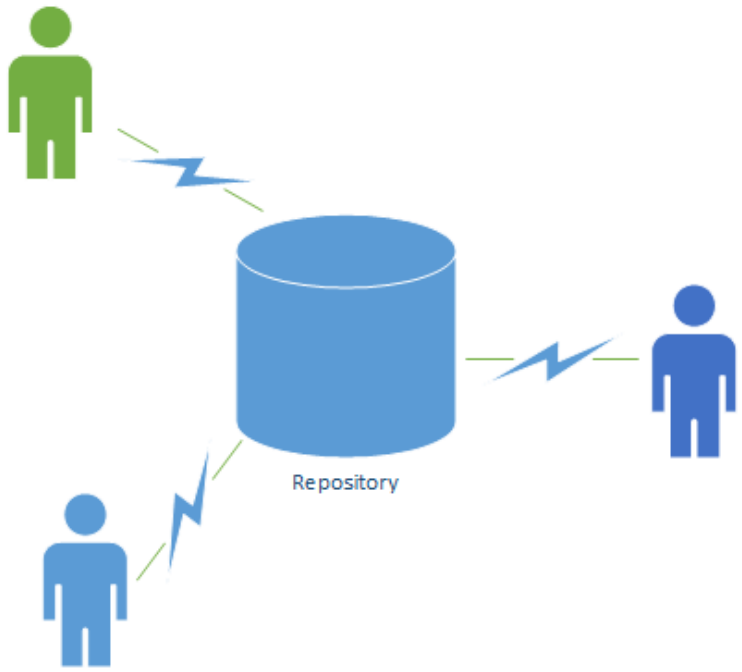

Figure 1.1: Client data model

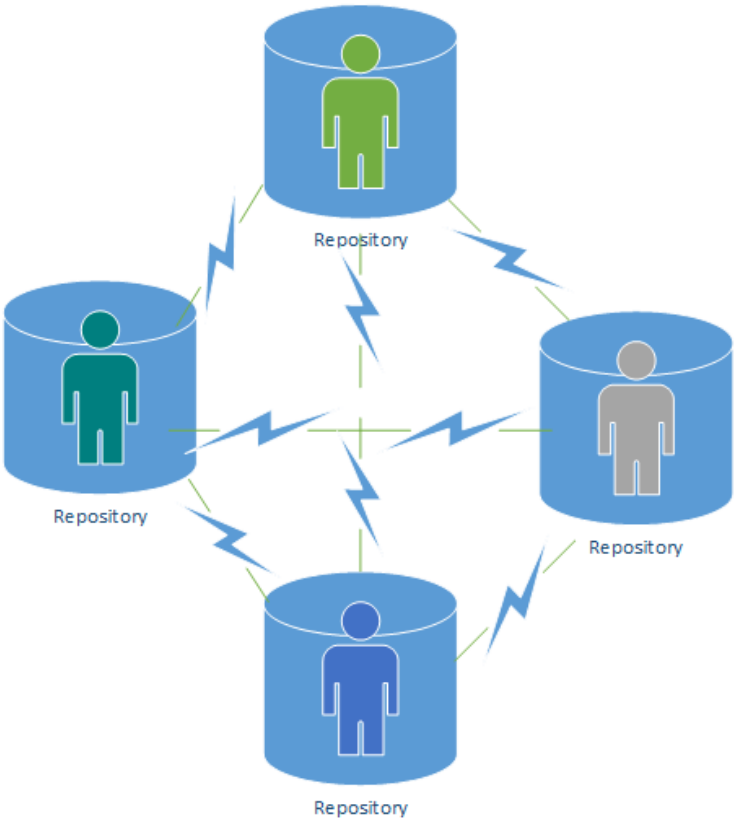

Figure 1.2: Distributed data model

1.4.2 Distributed based model

1.5 Git and GitHub

1.5.1 Beyond Source Code

Chapter 2

Memory and Intorduction to Classes

2.1 Heap and Stack

2.2 Classes

2.3 Reference types and Value types

2.4 More on Methods

Chapter 3

Arrays and Collections

3.1 Multi-Dimensional and Jagged Arrays

3.2 Lists

Chapter 4

Objects and Structures

4.1 Classes and Objects

4.2 Members

4.3 Structures

4.4 Methods and Properties

Chapter 5

Algorithms

Chapter 6

GUI and more on Classes

6.1 Introduction to access modifiers (Public and Private)

6.2 Overriding

6.3 User Interface Design Guidelines

6.3.1 General

6.3.2 Windows Forms Specific

6.4 Forms

6.4.1 Add controls in a form

6.4.2 Designer

6.4.3 Controls Types

6.4.4 Programmatically controlling them

Chapter 7

Testing and Data Validation

7.1 Introduction to Test Driven Development

7.2 Introduction to Unit Testing

Chapter 8

File System

8.1 Access files, directories and drives

8.2 Create files and directories

8.3 Write a simple

8.4 Introduction to object relationship