

## Document describing the system installation

### 1) Downloads

#### 1. Eclipse for Java enterprise developer

All our java, jsp, servlet files have been implemented from the eclipse IDE. Indeed, this one allows us to easily create and develop a web application via our J2EE application server (java enterprise developer). For that we had to download from the site.

<https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-java-ee-developers>.

#### 2. Java, Jdk and JRE

Our project require the version 12 or more of java. If you don't have it, go to download it (here I choose the version 13) at:

<https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>

#### 3. Apache Tomcat server

The Tomcat web container, is composed of a jsp engine, a servlet engine and a deployment descriptor for web type war modules. These engines are in reality APIs which are implemented in the Tomcat server, and which make it possible to deploy only Java web applications of the war type.

It's the Apache Tomcat application server that will allow us to execute our code in addition to doing the HTTP work. Thus, such a server includes an HTTP server and adds to it the management of objects of various natures through the container (JSP, servlet, war, deployment descriptor).

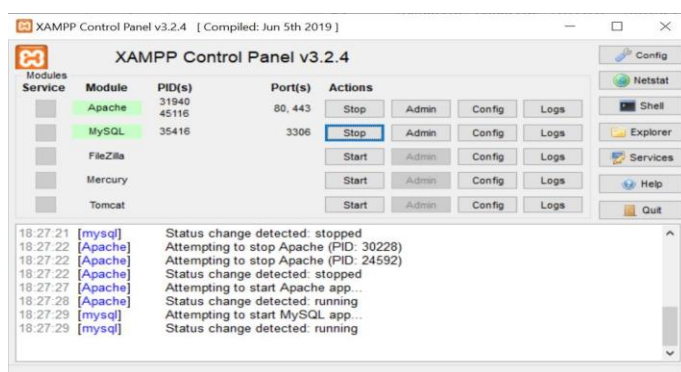
Downloading Tomcat: we have downloaded the binary version of Apache Tomcat 7.0 from the site:

<https://tomcat.apache.org/download-70.cgi>. In our case, we decided to connect the tomcat server to port 8000 on the computer. Just open the tomcat server from eclipse and change the HTTP / 1.1 port to 8000.

#### 4. XAMPP

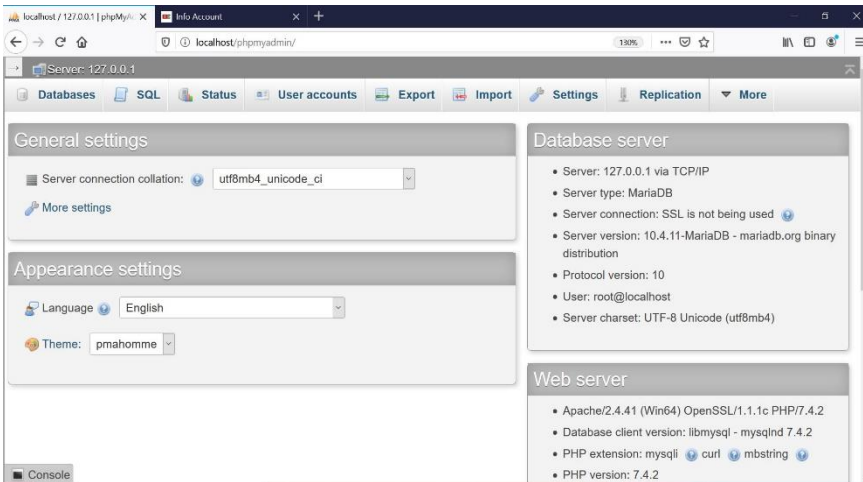
Each web application, simple or complicated, requires a database to store the data. MySQL, which is open source, is the database management system (DBMS) we have chosen, and it is one of the main components of XAMPP. Thus, our database is hosted in the XAMPP server (this server is a couple apache server with a php web engine). XAMPP allows us to configure a local test server before the implementation of the website. We have downloaded XAMPP from the site:

<https://www.apachefriends.org/fr/download.html>. After downloading XAMPP, check that MySQL and Apache are in "running" state:



By default, as we can see, MySQL is connected to port 3306 on the computer. In order to use this tool we must connect to a browser at the url address <http://localhost/phpmyadmin/>.

This url display the follow page :



## 5. MySQL connector

Java uses an API called JDBC (Java Database Connectivity) to connect to the database. JDBC (Java Database Connectivity) is a connection interface between the java program and the SQL database. This technology is a driver that defines how the client has access to the MySQL database.

The JDBC API classes are in the java.sql package. The 4 important classes that we use in our project are:

- DriverManager: responsible for loading and configuring the database.
- Connection: connect by authenticating to the database.
- Statement: send the sql request to the database.
- ResultSet: browse the results retrieved from the database in the case of a data selection.

We use these classes as you can see in the Database.java and DatabaseFoundations.java files in the database package. Thus, in order to communicate with the database from our system, it is necessary to access it, and then exchange data with the database. And that thanks to these functions.

- DatabaseFoundations.java: this file contains the creation of the database as well as that of the tables in case it does not exist, as requested by the professor.
- Database: this file contains all the functions necessary for communication with the database, such as adding a new user or adding a new command.

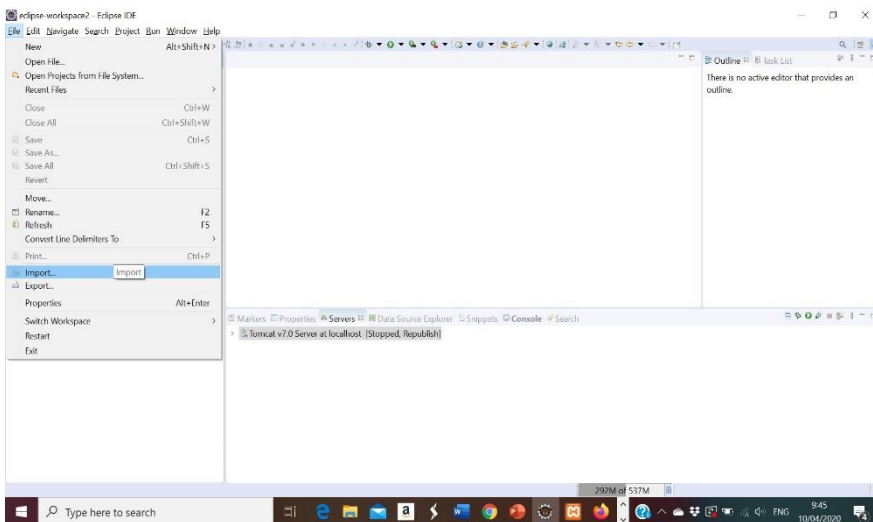
The driver we use for MySQL is "com.mysql.jdbc.Driver". We have downloaded the mysql-connector-java-5.1.14.jar driver from the site: <https://dev.mysql.com/downloads/connector/j/5.1.html>. added in the classpath of our project.

## 2) installation of our project as well as all the files necessary for its start-up

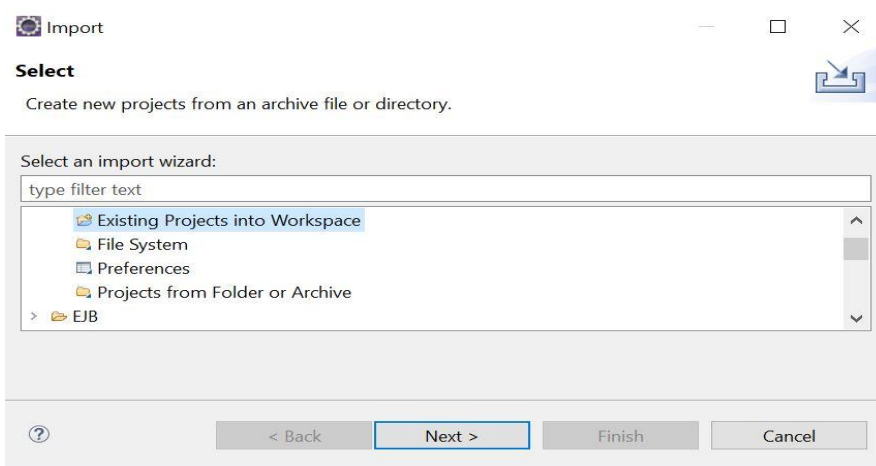
Our source files are in the GameWarrior.zip file. Download it and extract it. after extraction, you should have a GameWarrior file. For now, open eclipse.

follow the instructions to be able to operate our application:

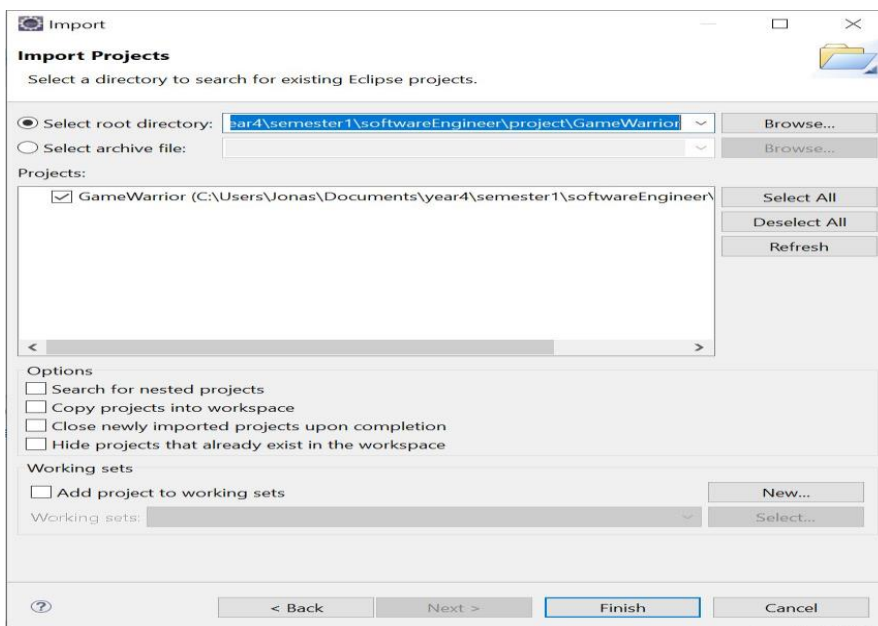
1. click on file, then on import like this:



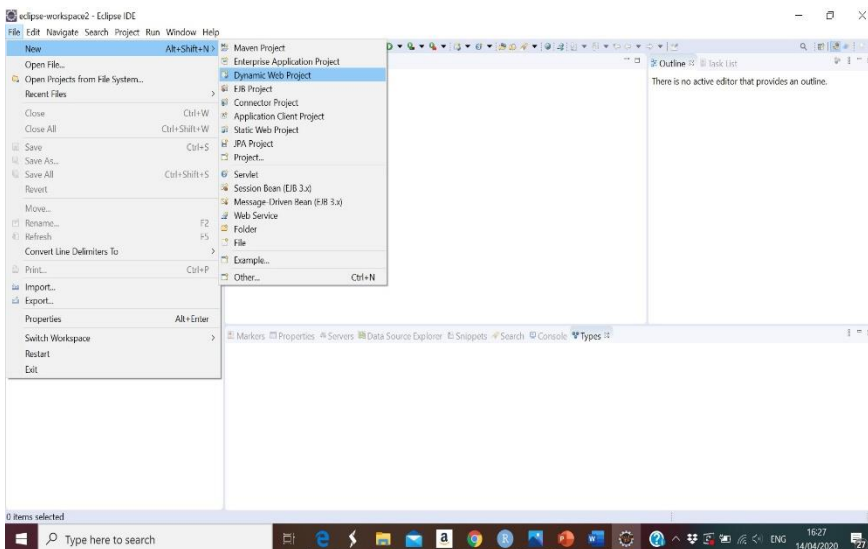
click on existing project into Workspace, and click “Next” like this:



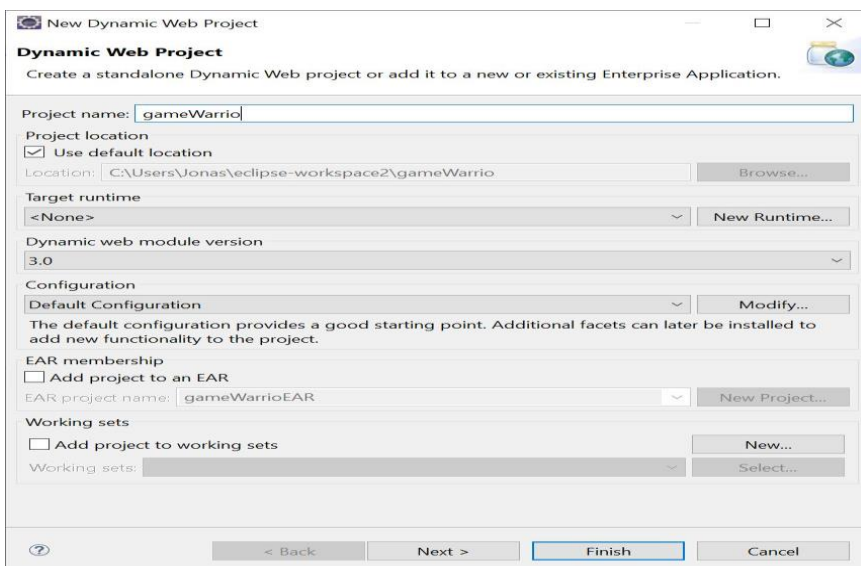
search your file GameWarrior on your computer and open, and after click on “Finish” it like this:



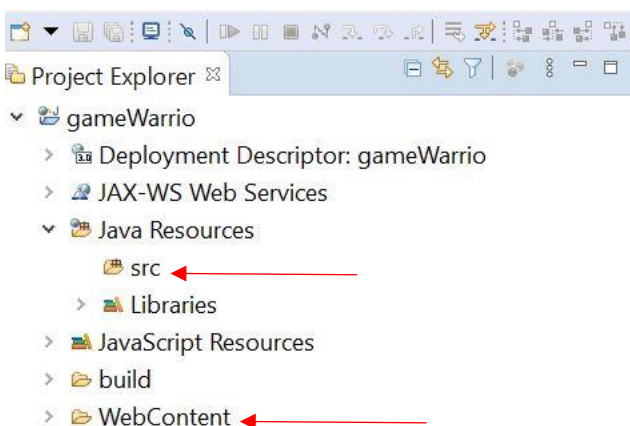
2. If you have only the webContent file and src file that is the case for us (so you don't need the first step), you have to open a new Web dynamic project like this:



And the follow click on “Finish”:

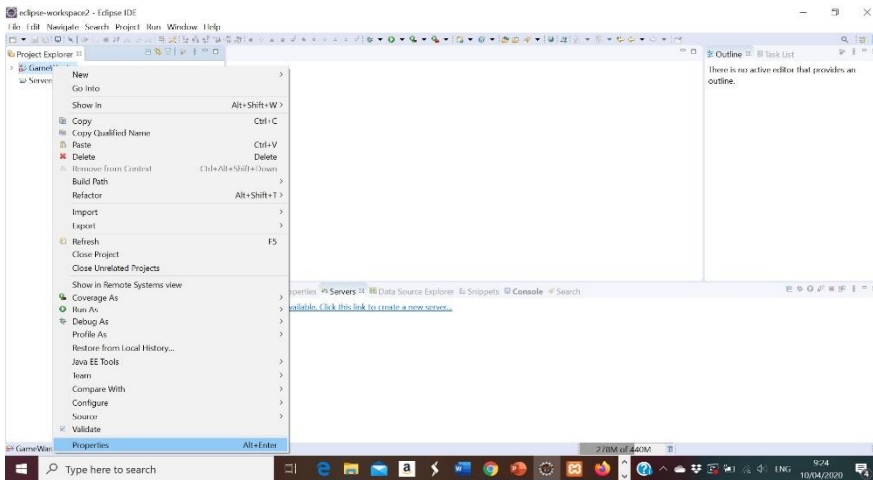


And you can to copy and paste the WebContent file and src file their right place here

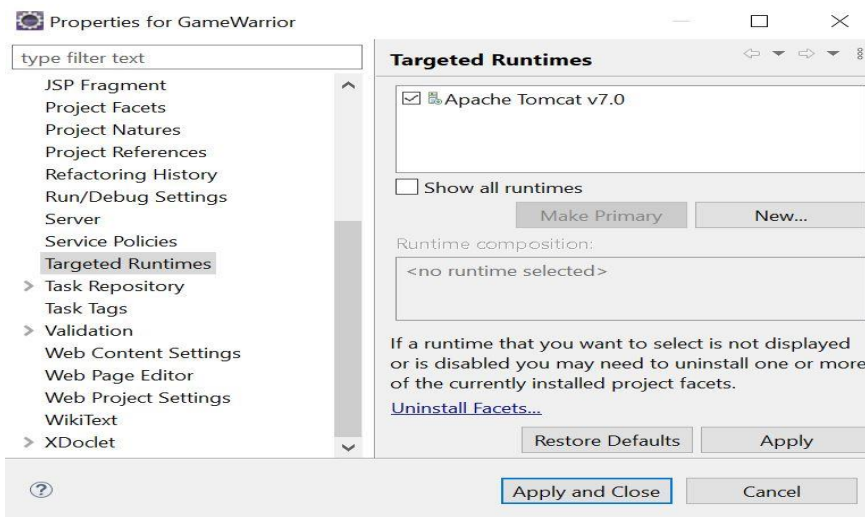


In order to copy src files, you need to open 4 new package in src file : controller, database, modle, view. In each package, copy the relevant files from our src file accordingly.

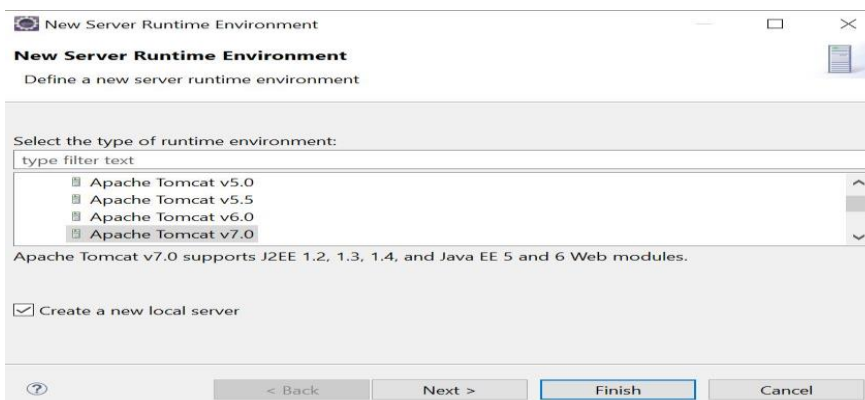
3. now, go into properties of the project like this:



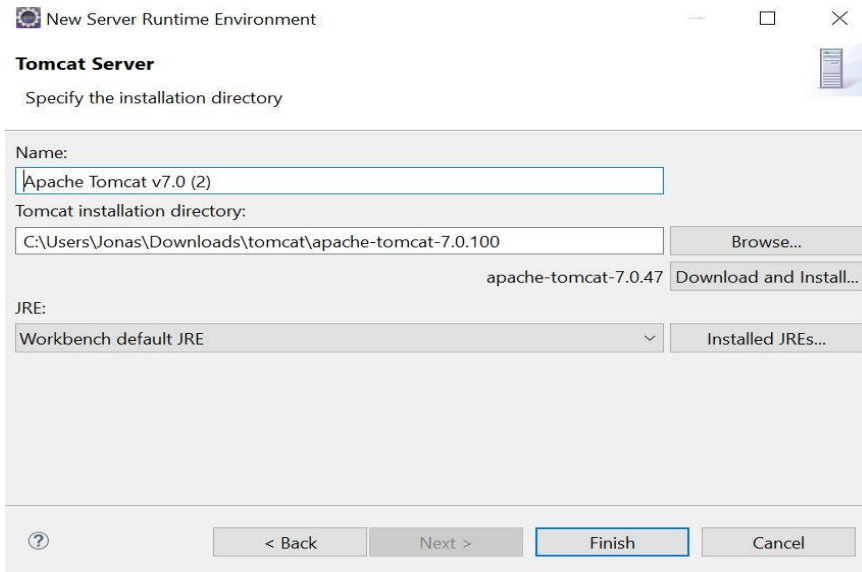
click on “Targeted Runtimes”. In my case here, I have already an Runtime environment “Apache Tomcat v7.0”. It is the server we downloaded earlier in this document. If you still have it, select it and click on “Apply and close”. If you didn’t install it yet, click on “New” and the follow will be at the next step.



select “Apache Tomcat v7.0”, and select “Create a new local server”, and click on the button “Next” like this :

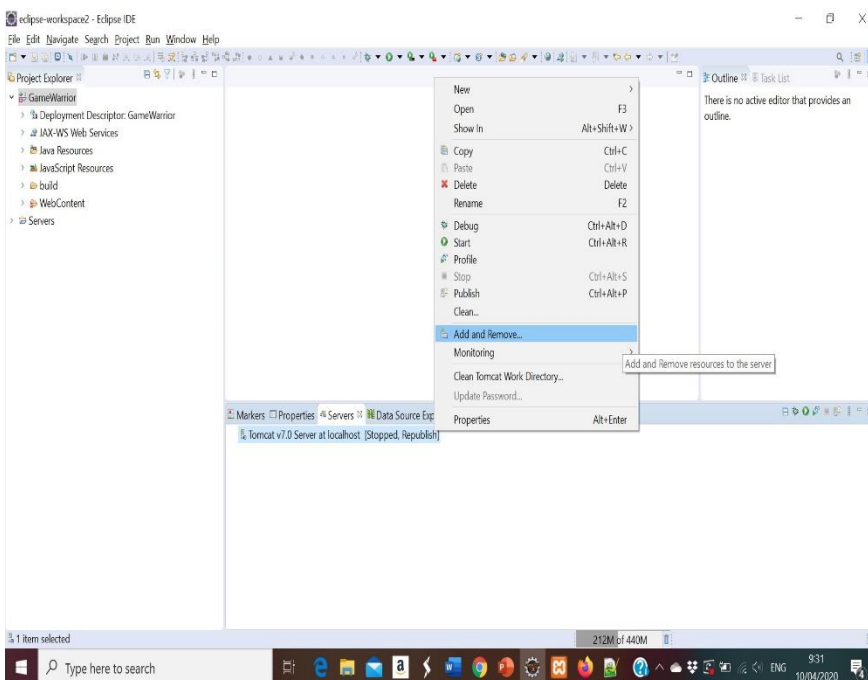


Go in your Browser, and select the file that you have downloaded Apache Tomcat server. We downloaded this file earlier in this document, in the first part. After selecting the file, click on “Finish”.

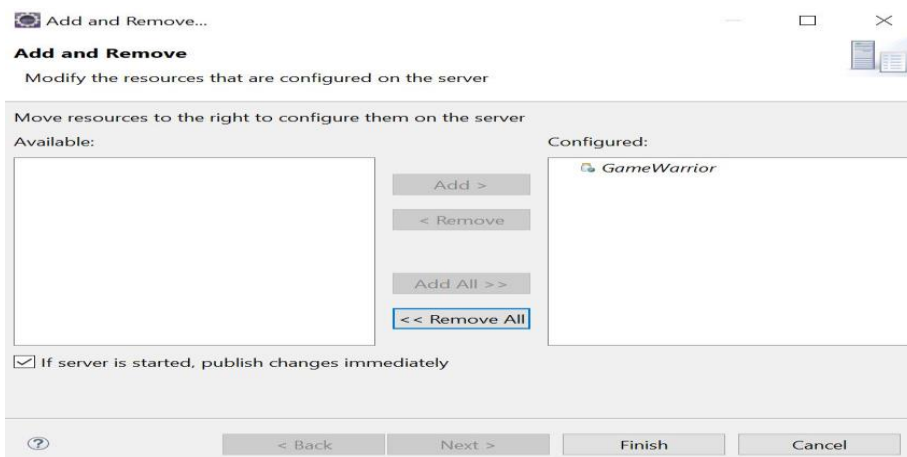


4. now go to the server window, then do the following:  
on Tomcat -> Add and Remove" like this:

"right click

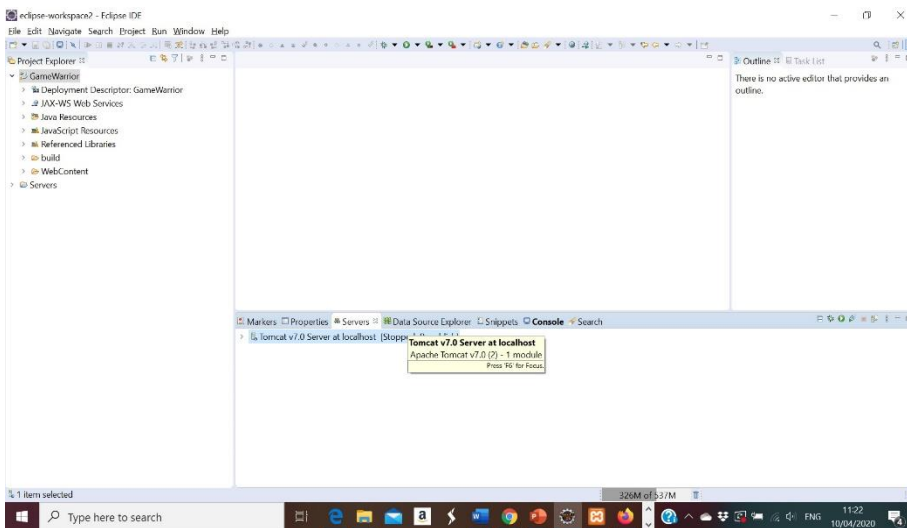


And the following window will be open: On the left, in Available, click select GameWarrior, then click on "Add" and then click on "Finish" like this:

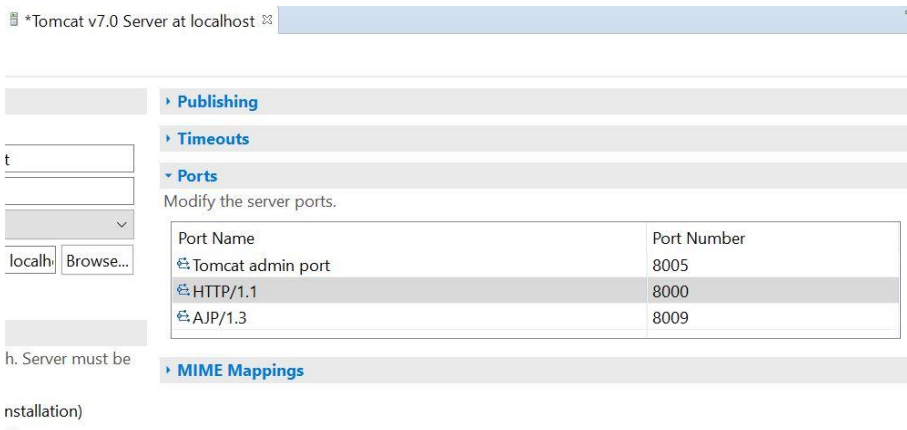




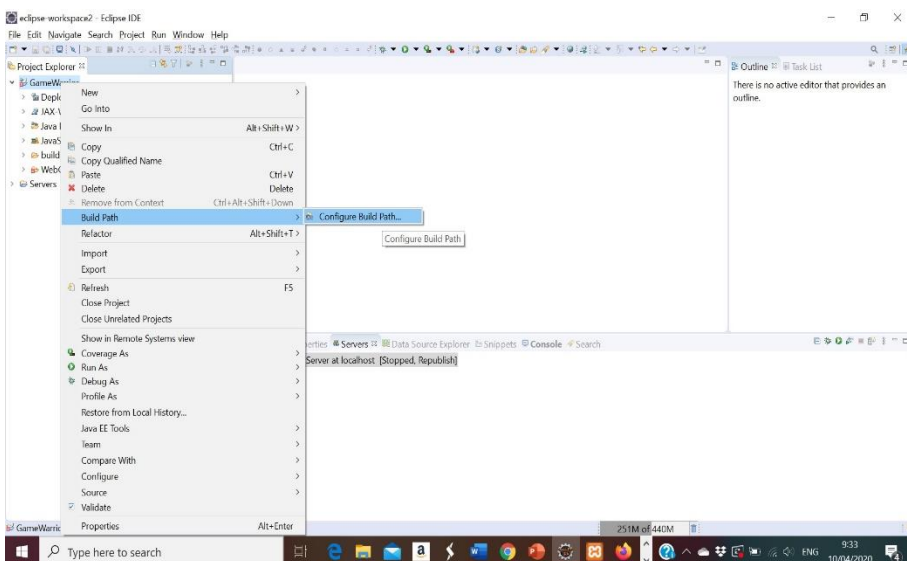
5. Now double click on Tomcat from the window server like this:



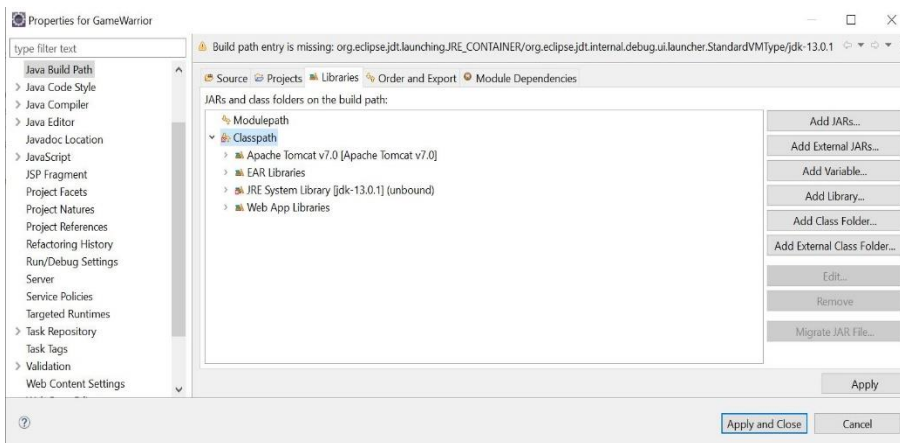
And change the port number of HTTP to 8000, in order to work with our application, like this:



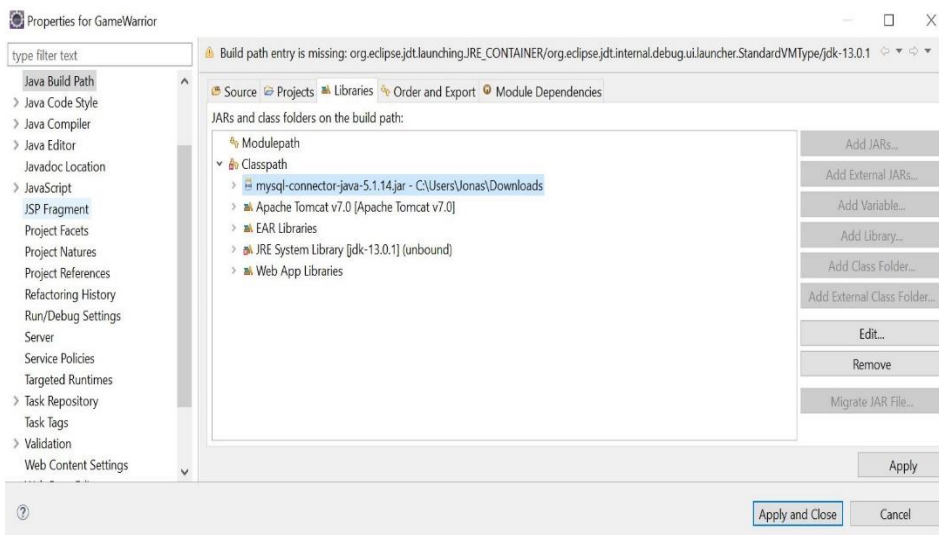
6. Now, right click on GameWarrior and select: "File -> Build Path -> Configure Build Path" like this:



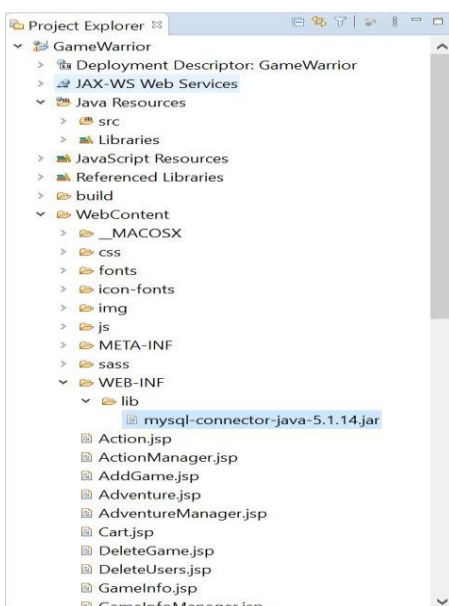
Select Classpath, and click on "Add External JARs..." like this:



And select mysql-connector-java-5.1.1.14.jar. It is the file that we downloaded from the first part. This file allows us to connect to the database.

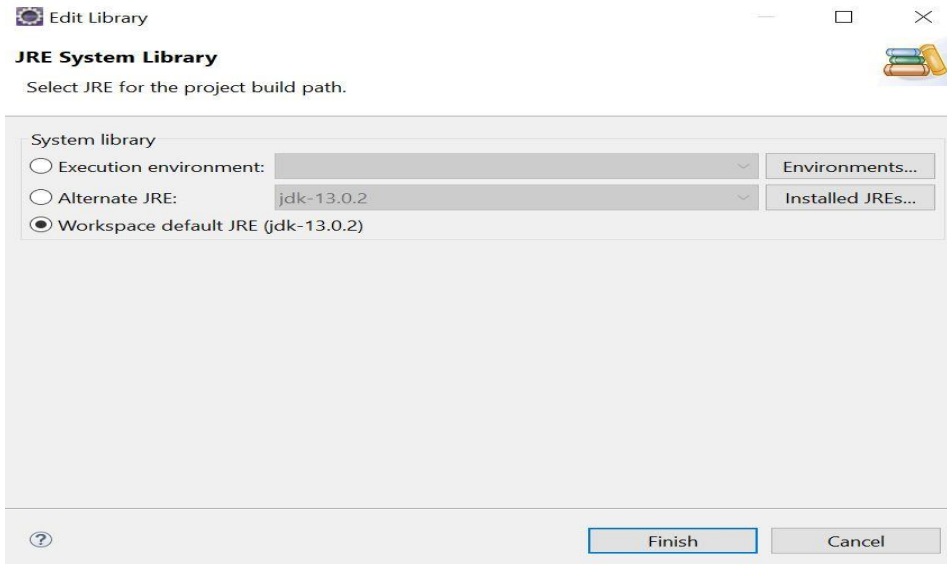


secondly, copy the mysql-connector file also in the WEB-INF - lib folder of your project. Indeed, the sub-folder named lib, contains the libraries necessary for the project (.jar archives). See where you need to copy mysql-connector:

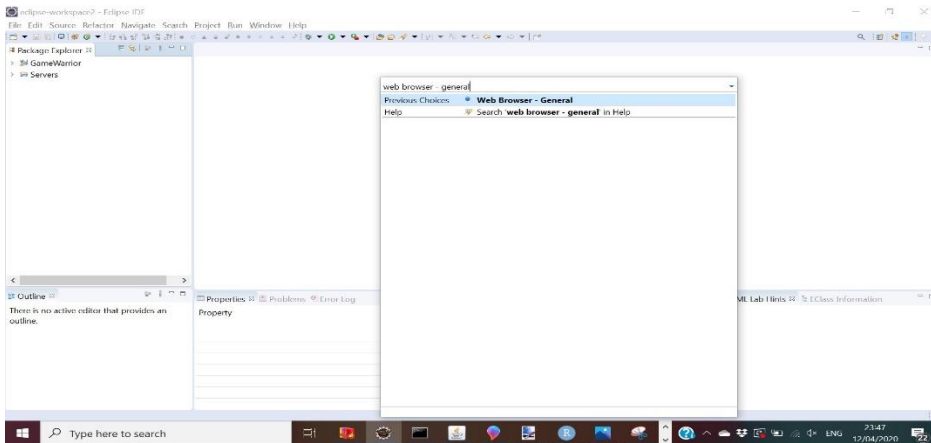




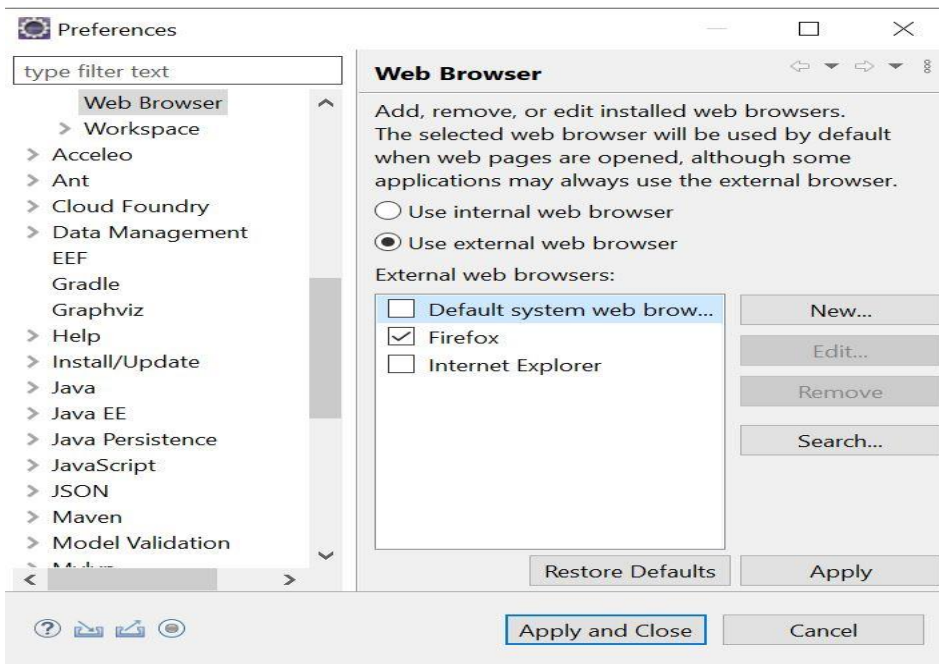
7. For now, in the same window (see the last picture) select JRE System Library, and click on “Edit...”. This window will be open. Choose the JRE (or from “Alternate JRE”, either select “Workspace default JRE” if you have), in our case we choose jdk -13.0.2 , that we downloaded in the first part. After that , select “Finish” and after “Apply and close”



8. Change the browser that allow you to open the website when you run the it on server. First you need to go to eclipse -> preferences , and after to go to General -> web browser. But it's quickly to do it like this. First search here:



Then change the web browser of your choice and click on “Apply and close”, like this:

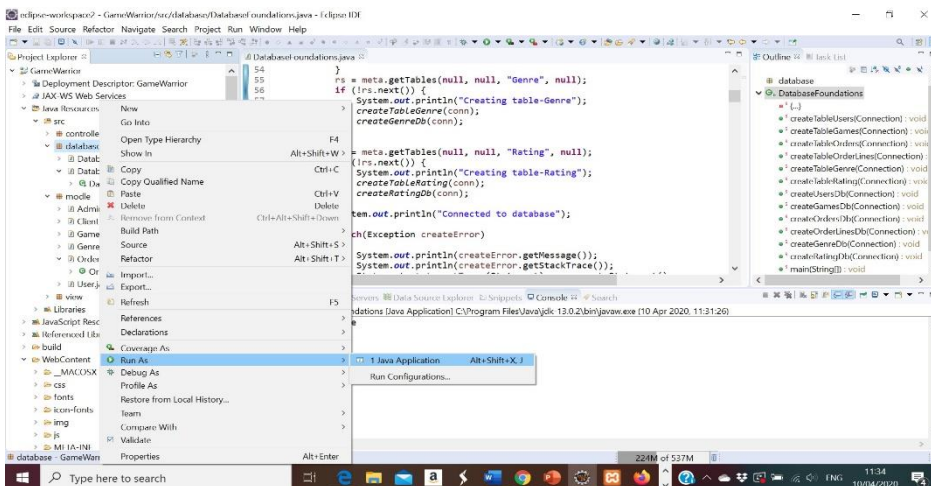


### 3) Connection to the database and start of our project

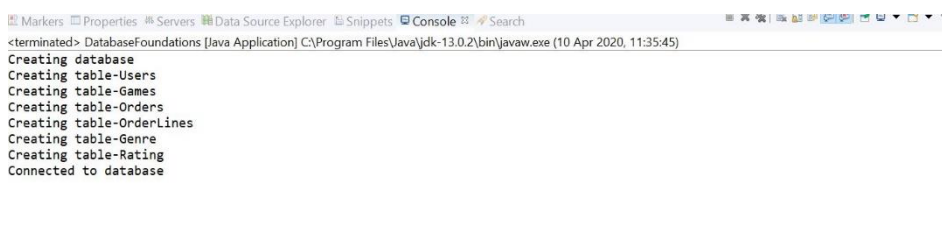
#### 1. Creating our database and connect to it

In the DatabaseFoundations.java file (it is in the package database), we connect to our server, then we create our "gamewarrior" database as requested by the professor, directly from our system. If there is an error during the creation of our database, we return an error.

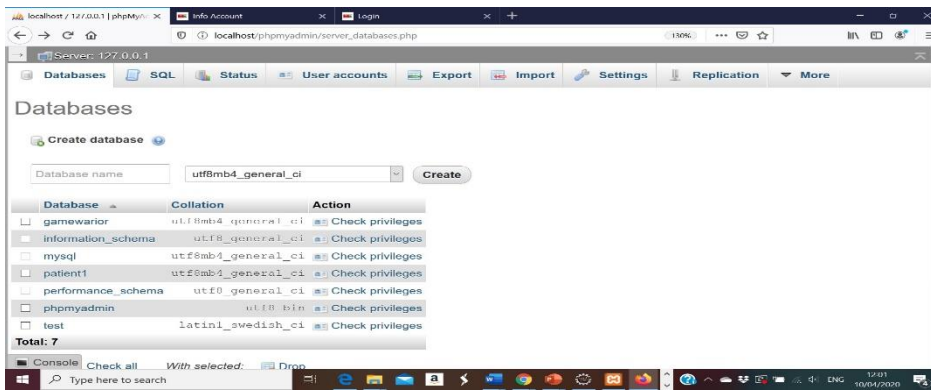
Run our database package in order to create for the first time our database (the main function is in DatabaseFoundations.java). You need to run it as Java application like this:



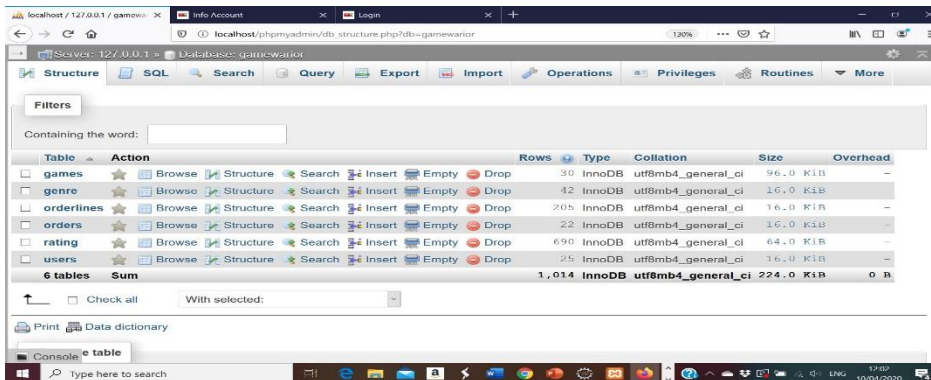
The console should display:



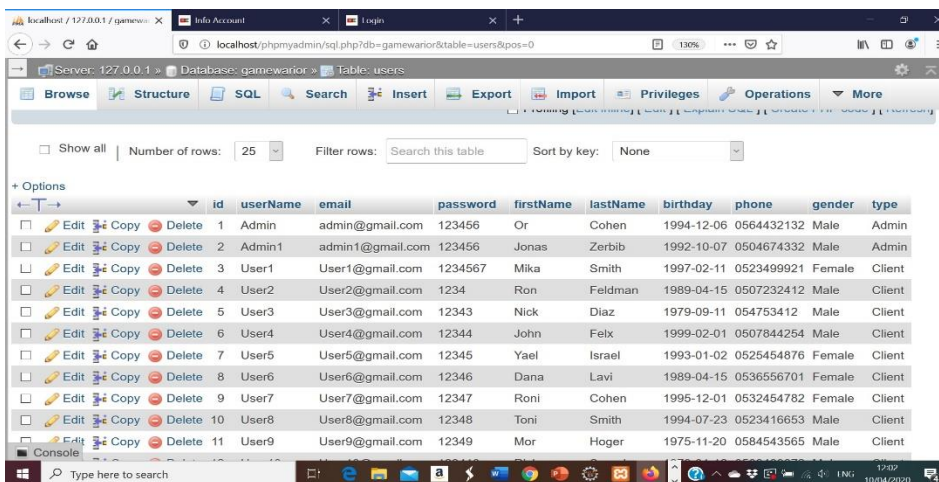
Now remember we had opened a url which connects us to the xampp server. Let's go then select "Databases" like this:



And click on “gamewarrior”, and you will see:



Let me show you the user table. click on users, you will see:

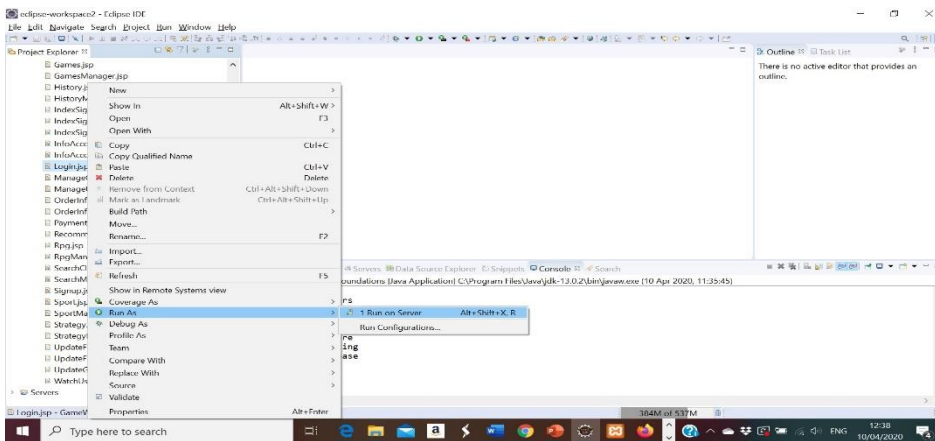


you can see that the table corresponds exactly to the data entered in the users table from our file DatabaseFundations.java

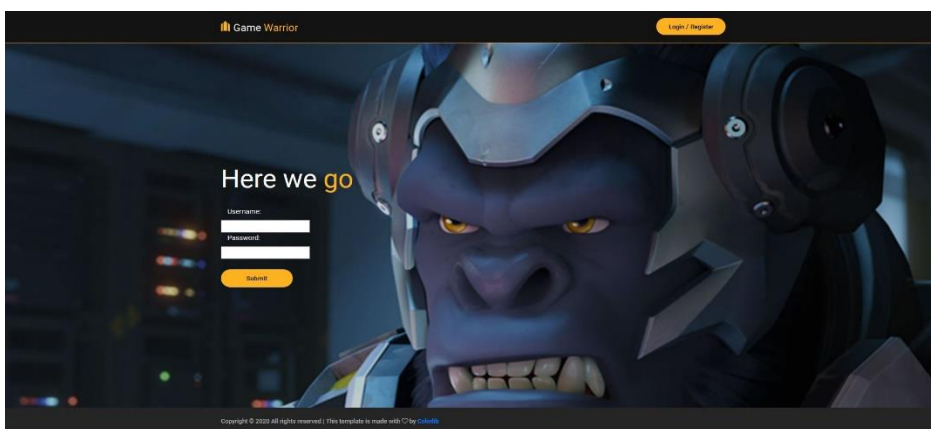
The advantage of the xampp server, is that you can modify directly your database, add, remove, or modify your data as you can see in this table.

## 2. Start our project:

You can run the project from the page Login.jsp for example. You need to run this page as “Run on server” like this:



And your browser will open the internet page:



From there, log in from an administrator account like: username: Admin1, password 123456 .Or from a customer account like: username: User1, password: 1234567. You can surf the site freely, create a new account, place an order, and more. The presentation of the website is in the document PresentationGameWarrior.pptx.