# Test and Development environment

## Operating System

The Operating System is Windows 10 Home Edition.

## IDE

Test and development took place in IntelliJ IDEA 2020.2.3 x64, with JDK 15.

The app is based on the Java EE template of IntelliJ according to the instructions of the link: <https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-ee-application.html#source_code>

## Tomcat

The Tomcat server was deployed with a Docker image according to the instructions of the link: <https://www.jetbrains.com/help/idea/deploying-a-web-app-into-an-app-server-container.html#Deploying_a_web_app_into_an_app_server_container-5-procedure>

In order to implement Docker Tomcat Image, Docker for Windows should be installed and connected with IntelliJ. In this case Docker was installed in Windows Home according to the instructions: <https://docs.docker.com/docker-for-windows/install-windows-home>

## MySQL

The app communicates with a MySQL database that was deployed with a WAMP server installed in windows. The WAMP server was downloaded from: <https://www.wampserver.com/en/>

The communication between the Tomcat Server and the MySQL server takes place through the ip of the pc and the port 3306.

In order to open the communication outside the localhost user should be able to receive connections from outside.

Database name : “database”

User: “root”

Password : “”

In order to create the tables please run the SQL script that is provided inside the zip file.

## App Run Environment

The app “runs” from a browser, using the URL: <http://127.0.0.1:8080/PrimerCRUD-1.0-SNAPSHOT/> where PrimerCRUD-1.0-SNAPSHOT stands for the war file in which the app is built.

## External libraries and Dependencies

The app uses the library mysql-connector-java-8.0.22, which is included in the zip file.

Right Click on Project ->Open Module Settings->Modules->Add-> (Select the MySQL jar file) mysql-connector-java-8.0.22 which is provided in the applications zip file.

App can be deployed either by the IntelliJ’s build or by Maven deployment for that reason the following lines wre added to the pom.xml

<properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
</properties>

and

<dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.22</version>  
</dependency>

# Code Small Documentation

Zip file includes:

Readme.doc that you are currently reading ☺

The database.sql file for creation of the database’s tables.

The gitignore file.

The project files for the IntelliJ deployment.

The source code of the app.

The main package is the gr.primer.PrimerCRUD, which is divided into the sub-packages “configuration”, “dataLayer”, “logicLayer” and “viewLayer”. The app resembles the three tier architecture. The index.jsp file is included in the “webapp” folder. The “index.jsp” connects with the “ViewLogic.java”. The “ViewLogic.java” connects with the “User.java”, the “Department.java”, the “index.jsp” and the “DaoImpl.java”. The “DaoImpl.java” connects with the “ViewLogic.java”, the “SQLImpl.java” and the “User.java” and the “Department.java”. The “SQLImpl.java” connects with the “DaoImpl.java”, the “User.java” and the “Department.java”.

# User Interface Small description

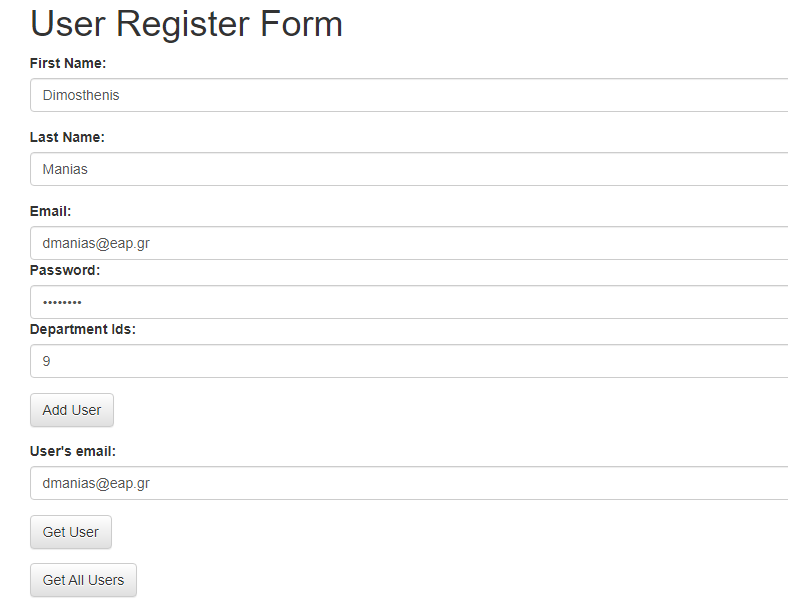


Figure 1. User's part of the interface

“Add User” adds a user in the database.

“Get User” brings the user from the email filled in the box above.

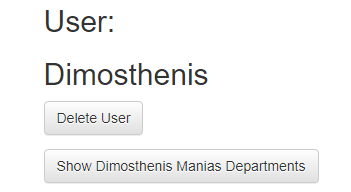


Figure 2. Get User Functionality

“Show … Departments” brings the Departments of the User and “Delete User” deletes the user from the database.

“Get All Users” brings all users that are stored in the database and the UI user can performs the “Delete User” and “Show…Departments” operations for each User.

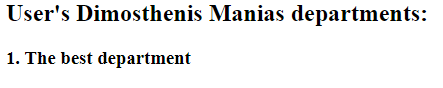


Figure .Show Dimosthenis Manias Departments functionality

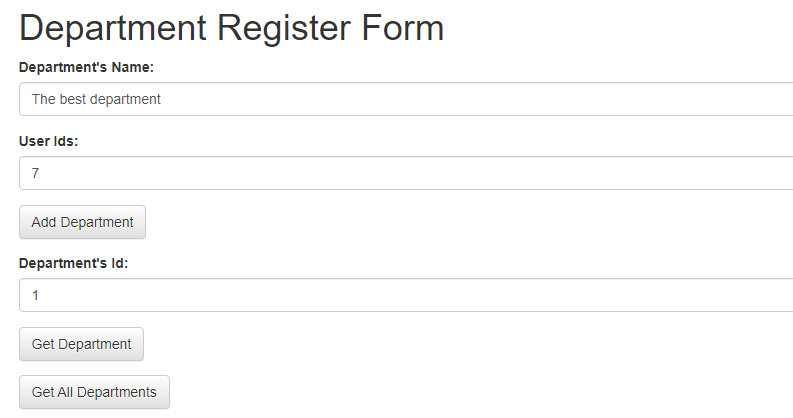


Figure 4. Department's part of the interface

Respectively,

“Add Department” stores a department in the database.

“Get Department” brings the department with the specific “Id” from the database and,

“Get All Departments” brings all departments from the database.

For each department can be performed the “Delete Department” and “Show Department Users” operations.

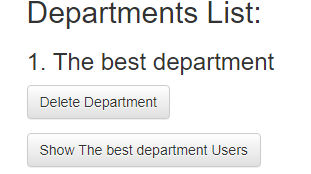


Figure 5. Get All Departments Functionality