



*Department of*  
*Computer Science and Engineering*

## ***MeteoCal***

*Project development for the 2014 Software Engineering 2 Course*

*Supervisor:*     **Prof. Di Nitto**

### ***Authors***

**Andrea Bignoli**

Matr. 837493

*andrea.bignoli@gmail.com*

**Leonardo Cella**

Matr. 838074

*leonardocella@gmail.com*

**Installation**

**Guide**

## ***System Description***

MeteoCal is a web application, that provides his user with a lot of services related to personal event management. Users will need only a browser and no plugin installation is required.

In our design we dedicated particular attention to the friendliness of usage of our service. Our first purpose is to provide the user with the easiest and lightest interaction possible considered the complexity of the functionalities we decided to offer.

## ***Table of Contents***

1. Required Software	1
2. Installation Tutorial	1
3. Using The Source Code	9

## **1. Required software**

To install our application the user will need the following software on his machine:

- JDK 1.8.31+ (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)
- Glassfish 4.1 (<https://glassfish.java.net/download.html>)
- MySql Community Server (<http://dev.mysql.com/downloads/mysql/>)
- MySql Java connector (<http://dev.mysql.com/downloads/connector/j/>)

Optional, but suggested:

- Netbeans 8 (<https://netbeans.org/downloads/>)

## **2 Installation tutorial**

Once the components have been installed execute mysql from terminal. (On Windows system we suggest to add %MySql Install Folder%\MySQL Server 5.6\bin to the PATH system variable)

If the PATH variable has been added you can execute MySql from terminal by typing:

```
mysql -u root -p
```

Enter your root password and write the following commands:

```
create database meteocaldb;
```

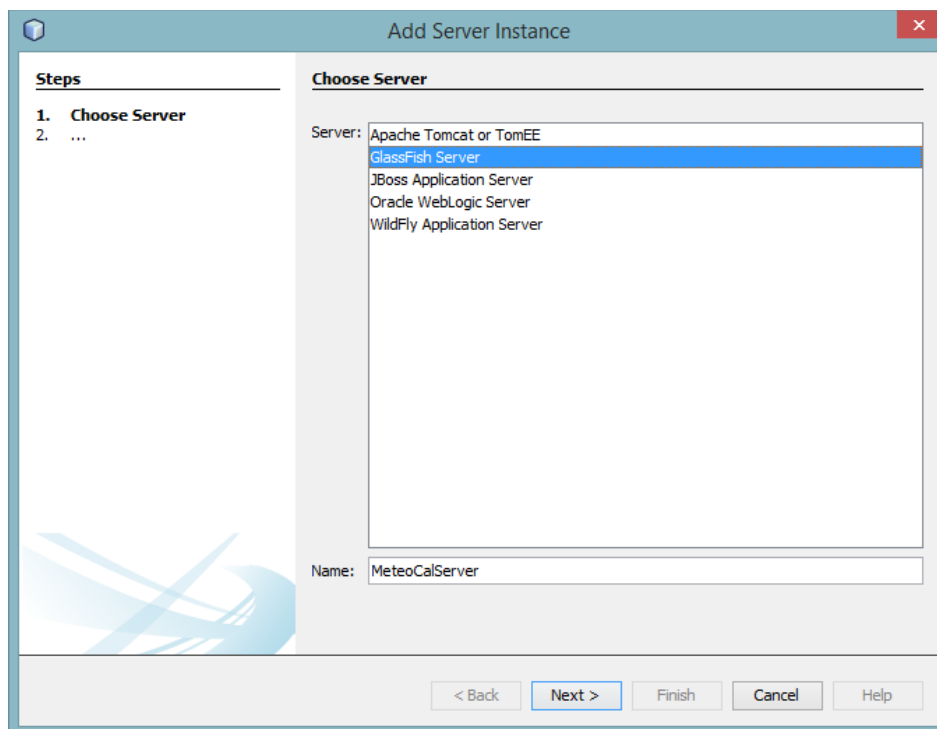
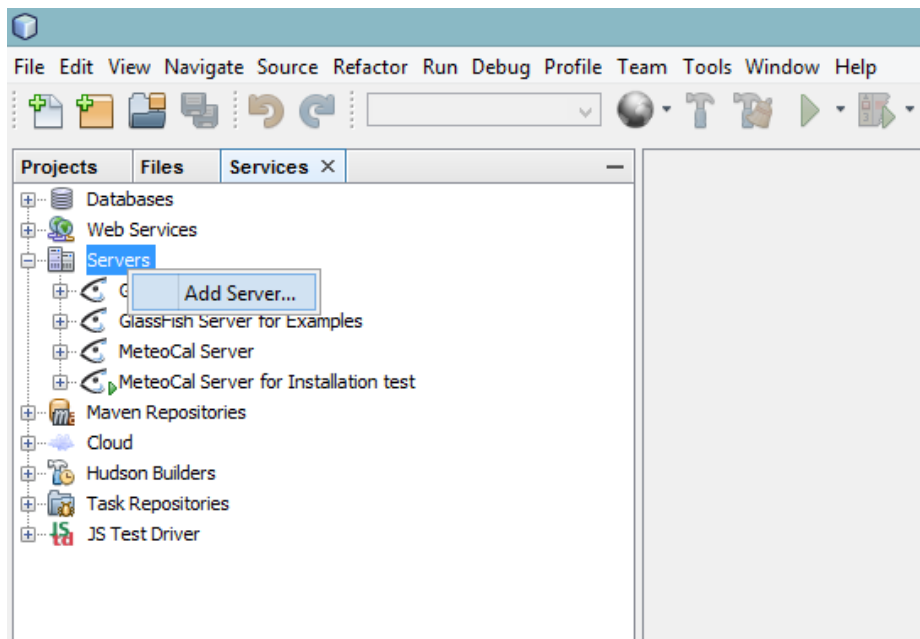
```
create user 'meteocal_user'@'localhost' identified by 'password'
```

```
grant all on meteocaldb.* to 'meteocal_user'@'localhost'
```

Then place mysql-connector-java-5.1.34-bin.jar (extracted from the MySql Java connector) in the folder %Glassfish installation folder%\glassfish\modules

Then start the Glassfish server. This can be done by command line or in GUI environments. One of these is Netbeans, that we will use in this tutorial.

Go to the Services tab, under the voice Servers and follow the visual instruction from these screens:



## Add Server Instance

### Steps

1. Choose Server
2. Server Location
3. **Domain Name/Location**

### Domain Location

Domain:

Host:  ☒ Loopback

DAS Port:  HTTP Port:  ☒ Use default ports (disabled due to port conflicts)

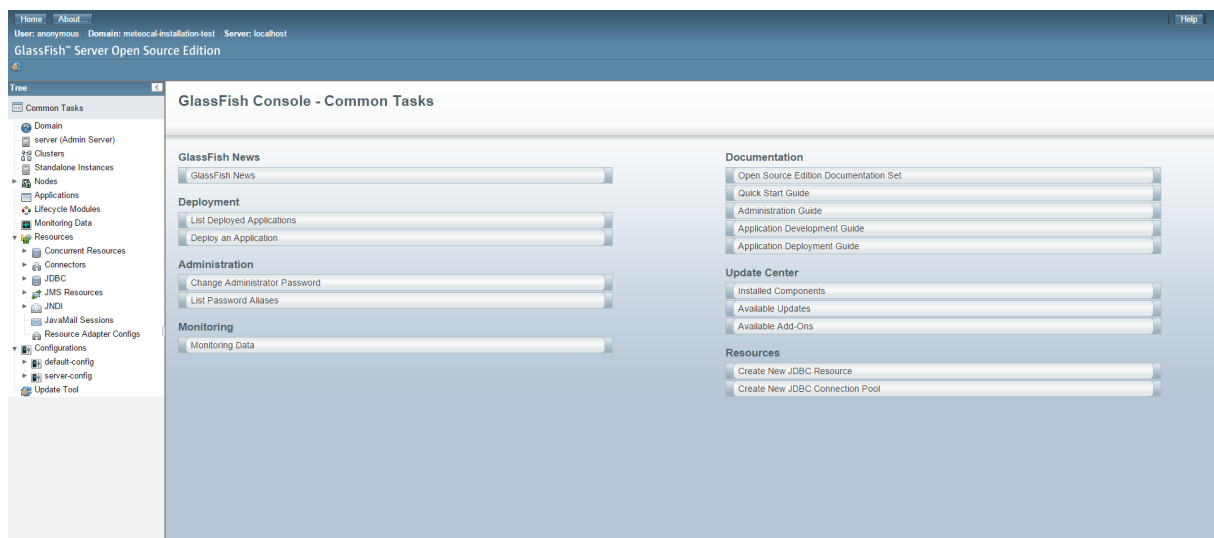
Target:

User Name:

Password:

Create new embedded domain: com-meteocal-domain

Confirm, wait for the domain creation. When this operation is finished right click on the new server and choose “View Domain Admin Console”. In the browser, you should see a page like this:



Then Resources/JDBC Connection Pools, choose to create a new one and enter the following data:

Home About...

User: anonymous Domain: meteocal-installation-test Server: localhost

GlassFish™ Server Open Source Edition

Tree

- Common Tasks
- Domain
  - server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
- Lifecycle Modules
- Monitoring Data
- Resources
  - Concurrent Resources
  - Connectors
  - JDBC
    - JDBC Resources
    - JDBC Connection Pools
  - JMS Resources
  - JNDI

### New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

**General Settings**

Pool Name: \*

Resource Type:  Must be specified if the datasource class implements more than 1 of the interface

Database Driver Vendor:  Select or enter a database driver vendor

Introspect: ☐ Enabled If enabled, data source or driver implementation class names will enable introspection

Next

Additional Properties (5)		
<input type="checkbox"/> <input type="checkbox"/>   <input type="button" value="Add Property"/> <input type="button" value="Delete Properties"/>		
Select	Name	Value
<input type="checkbox"/>	ServerName	<input type="text" value="localhost"/>
<input type="checkbox"/>	Port	<input type="text" value="3306"/>
<input type="checkbox"/>	DatabaseName	<input type="text" value="meteocaldb"/>
<input type="checkbox"/>	User	<input type="text" value="meteocal_user"/>
<input type="checkbox"/>	Password	<input type="text" value="password"/>

Finish.



Click on the new pool from the list and press Ping to verify that the connection works.

The screenshot shows the GlassFish Server Open Source Edition interface. The top bar includes 'Home' and 'About...' buttons, and user information: 'User: anonymous', 'Domain: meteocal-installation-test', and 'Server: localhost'. The left sidebar shows a tree view with 'Resources' expanded, and 'JDBC' > 'JDBC Connection Pools' > 'connectionToMeteocalDB' selected. The main panel has tabs for 'General', 'Advanced', and 'Additional Properties', with 'General' active. A yellow banner at the top of the main panel says 'Ping Succeeded'. Below it is the title 'Edit JDBC Connection Pool' and a description: 'Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a'. There are three buttons: 'Load Defaults', 'Flush', and 'Ping'. The 'General Settings' section contains the following fields:

- Pool Name:** connectionToMeteocalDB
- Resource Type:** javax.sql.DataSource (dropdown menu)
- Datasource Classname:** com.mysql.jdbc.jdbc2.optional.MysqlDataSource (text field)
- Driver Classname:** (empty text field)
- Ping:** ☒ Enabled
- Deployment Order:** 100 (text field)

Additional text for some fields: 'Must be specified if the datasource class implements more than 1 of the interfa' and 'Vendor-specific classname that implements the DataSource and/or XADatSou'.

Then create a new JDBC resource and enter the following data:

The screenshot shows the GlassFish Server Open Source Edition interface. The top bar includes 'Home' and 'About...' buttons, and user information: 'User: anonymous', 'Domain: meteocal-installation-test', and 'Server: localhost'. The left sidebar shows a tree view with 'Resources' expanded, and 'JDBC' > 'JDBC Resources' selected. The main panel has the title 'New JDBC Resource' and a description: 'Specify a unique JNDI name that identifies the JDBC resource you want to create. The nam'. The configuration fields are:

- JNDI Name: \*** jdbc/connectionToMeteoCalDB (text field)
- Pool Name:** connectionToMeteocalDB (dropdown menu)
- Description:** (empty text field)
- Status:** ☒ Enabled

Below these fields is a section titled 'Additional Properties (0)' with 'Add Property' and 'Delete Properties' buttons. A table with columns 'Select', 'Name', and 'Value' is shown, containing the text 'No items found.'

Then, under Configurations > server-config > Security > Realms, create a new Realm with the following data:

The screenshot shows the GlassFish Server Open Source Edition administration console. The left sidebar contains a 'Tree' view with the following structure:

- Common Tasks
- Domain
  - server (Admin Server)
  - Clusters
  - Standalone Instances
  - Nodes
  - Applications
  - Lifecycle Modules
  - Monitoring Data
  - Resources
    - Concurrent Resources
    - Connectors
    - JDBC
    - JMS Resources
    - JNDI
    - JavaMail Sessions
    - Resource Adapter Configs
- Configurations
  - default-config
  - server-config
    - Admin Service
    - Connector Service
    - EJB Container
    - HTTP Service
    - JVM Settings
    - Java Message Service
    - Logger Settings
    - Monitoring
    - Network Config
    - ORB
    - Security
      - Realms**
      - Audit Modules
      - JACC Providers

The main content area is titled 'New Realm' and includes the following configuration fields:

**Configuration Name:** server-config

**Name:** \* meteocal-authentication-realm

**Class Name:** com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm

Choose a realm class name from the drop-down list or specify a custom class

**Properties specific to this Class**

<b>JAAS Context:</b> *	jdbcRealm Identifier for the login module to use for this realm
<b>JNDI:</b> *	jdbc/connectionToMeteoCalDB JNDI name of the JDBC resource used by this realm
<b>User Table:</b> *	user_table Name of the database table that contains the list of authorized users for this realm
<b>User Name Column:</b> *	username Name of the column in the user table that contains the list of user names
<b>Password Column:</b> *	password Name of the column in the user table that contains the user passwords
<b>Group Table:</b> *	user_table Name of the database table that contains the list of groups for this realm
<b>Group Table User Name Column:</b>	 Name of the column in the user group table that contains the list of groups for this realm
<b>Group Name Column:</b> *	groupName Name of the column in the group table that contains the list of group names
<b>Password Encryption Algorithm:</b> *	SHA-256 This denotes the algorithm for encrypting the passwords in the database. It is a security risk to use weak algorithms.
<b>Assign Groups:</b>	 Comma-separated list of group names

You should now be ready to deploy the provided war file and test our platform.

To deploy the application go under Application and choose “Deploy”:

The screenshot shows the GlassFish Server Open Source Edition web console. The top navigation bar includes 'Home' and 'About...' buttons. Below the navigation bar, the user is 'anonymous', the domain is 'meteocal-installation-test', and the server is 'localhost'. The main title is 'GlassFish™ Server Open Source Edition'. On the left, a 'Tree' view shows the navigation structure: Common Tasks, Domain (server (Admin Server)), Clusters, Standalone Instances, Nodes, Applications (selected), Lifecycle Modules, Monitoring Data, and Resources. The main content area is titled 'Applications' and contains the text 'Applications can be enterprise or web applications, or various kinds of modul'. Below this, there is a section 'Deployed Applications (1)' with a table. The table has columns 'Select' and 'Name'. There is one entry: 'meteocal-web-install-test'. Above the table, there are buttons for 'Deploy...', 'Undeploy', 'Enable', and 'Disable', along with a 'Filter:' dropdown.

Then upload the provided meteocal.war file and set the following data:

The screenshot shows the 'Deploy Applications or Modules' page in the GlassFish Server Open Source Edition web console. The top navigation bar is the same as the previous screenshot. The left 'Tree' view shows the navigation structure: Common Tasks, Domain (server (Admin Server)), Clusters, Standalone Instances, Nodes, Applications (selected), Lifecycle Modules, Monitoring Data, Resources, and Configurations. The main content area is titled 'Deploy Applications or Modules' and contains the text 'Specify the location of the application or module to deploy. An application can be in a package'. Below this, there are two radio buttons for 'Location': 'Packaged File to Be Uploaded to the Server' (selected) and 'Local Packaged File or Directory That Is Accessible from GlassFish'. The 'Packaged File' option has a 'Choose File' button and the file name 'meteocal-web...ll-test.war'. The 'Local Packaged File' option has a text input field and a 'Browse' button. Below the 'Location' section, there is a 'Type' dropdown menu set to 'Web Application'. The 'Context Root' is a text input field with the value 'meteocal-web-install-test' and a note 'Path relative to server's base URL.'. The 'Application Name' is a text input field with the value 'meteocal-web-install-test'. The 'Virtual Servers' section has a dropdown menu set to 'server' and a note 'Associates an Internet domain name with a physical server.'. The 'Status' section has a checkbox for 'Enabled' which is checked, with a note 'Allows users to access the application.'. The 'Implicit CDI' section has a checkbox for 'Enabled' which is checked, with a note 'Implicit discovery of CDI beans'. The 'Precompile JSPs' section has a checkbox which is unchecked, with a note 'Precompiles JSP pages during deployment.'. The 'Run Verifier' section has a checkbox which is unchecked, with a note 'Verifies the syntax and semantics of the deployment descriptor. Vi'. The 'Force Redeploy' section has a checkbox which is unchecked, with a note 'Forces redeployment even if this application has already been dep'. The 'Keep State' section has a checkbox which is unchecked, with a note 'Retains web sessions, SFSB instances, and persistently created t'. The 'Deployment Order' section has a text input field and a note 'A number that determines the loading order of the application at se'. The 'Libraries' section has a text input field and a note 'A comma-separated list of library JAR files. Specify the library JAF'. The 'Description' section has a text input field.

Then, on the applications page you should be able to Launch the application and use it successfully.

## Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

Select	Name	Deployment Order	Enabled	Engines	Action
<input type="checkbox"/>	meteoal-web-install-test	100	✓	ejb, web	<a href="#">Launch</a>   <a href="#">Redeploy</a>   <a href="#">Reload</a>

This will bring you to a new page where there are two links, press the first one to enter.

## Web Application Links

If the server or listener is not running, the link may not work. In this event, check the status of the server instance. After launching the web application, use the browser's Back button to return to this screen.

**Application Name:** meteocal-web-install-test

**Links:**

- [server] <http://Z3570k:8080/meteocal-web-install-test>
- [server] <https://Z3570k:8181/meteocal-web-install-test>

Then you should see the MeteoCal Index page and be able to access every functionality we offer.

Registration

Username \*

Password \*

Password Confirmation \*

Signup

Login

Username \*

Password \*

Login

### **3. Using The Source Code**

To open the source code provided in the meteocal.zip file, extract the zip file and, in Netbeans, choose “Open Project”, navigate to the extraction folder and select the project folder. Make sure to have “Open Required Projects” selected.

