TrackMe project Julián Cuéllar Mangut
Javier Fernández Rodríguez

POLITECNICO
MILANO 1863

Acceptance Test deliverable

Deliverable:	ATD
Title:	Acceptance Test deliverable
Authors:	Julián Cuéllar Mangut, Javier Fernández Rodríguez
Version:	1.0
Date:	20-January-2019
Copyright:	Copyright © 2018, Julián Cuéllar, Javier Fernández

Revision history:

V 2.0:	After getting in touch with the developers.
V 1.0:	First version of the document.

Contents

1 Project

- **Authors of the project**: Aristide Bordoli and Juan Carlos Jaramillo.
- **Repository**: https://github.com/Jaywrkr/JaramilloBordoli

2 Installation Setup

2.1 Instructions

The project relies too much in NetBeans, the instructions included only mentions *You only need to open the Project in NetBeans and you can run it.*

Moreover, only the NetBeans versions with support for Java Web and EE are able to handle the project. Requirement which is not stated anywhere.

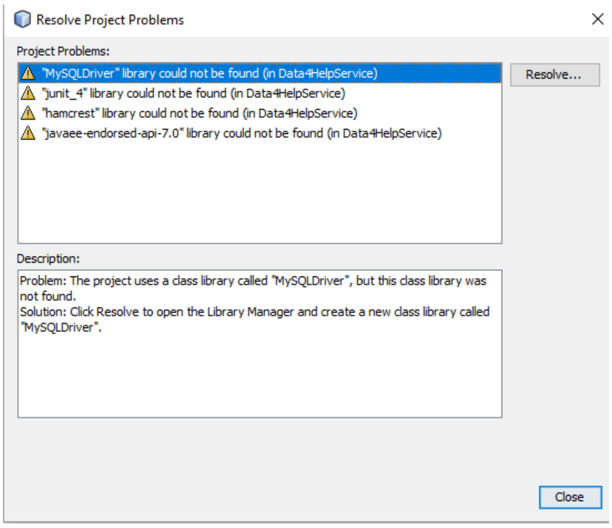
The system is implemented using servlets, but any instruction about setting up Tomcat or something similar is not included. The instructions also fails in mentioning the two servlets that are needed, and how to run them all together.

2.2 Errors found

After contacting the developers of the project, we have been told to follow the guide *Java Enteprise Edition, Toolkit installation and configuration manual* available at the beep page of the subject. In the following, the steps performed and the problems found are described.

MySQL and NetBeans with support for Web and EE were installed following closely the instructions.

The first problem found is the lack of several libraries, which are not included. Figure 1 shows the message displayed at the opening of Data4HelpService.

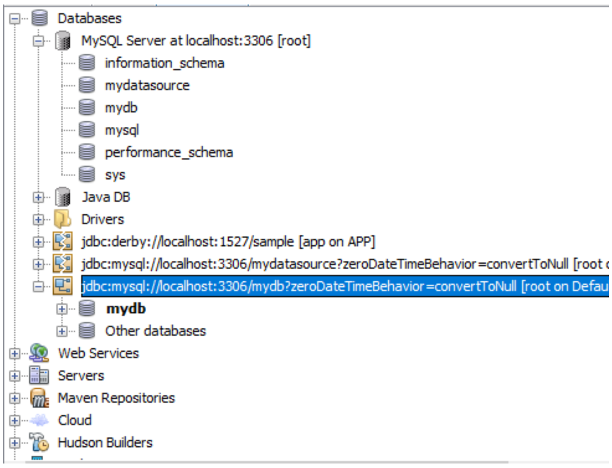


"MySQLDriver" library could not be found (in Data4HelpService)
"junit_4" library could not be found (in Data4HelpService)
"hamcrest" library could not be found (in Data4HelpService)
"javaee-endorsed-api-7.0" library could not be found (in Data4HelpService)

Problem: The project uses a class library called "MySQLDriver", but this class library was not found.
Solution: Click Resolve to open the Library Manager and create a new class library called "MySQLDriver".

Figure 1: Libraries missing

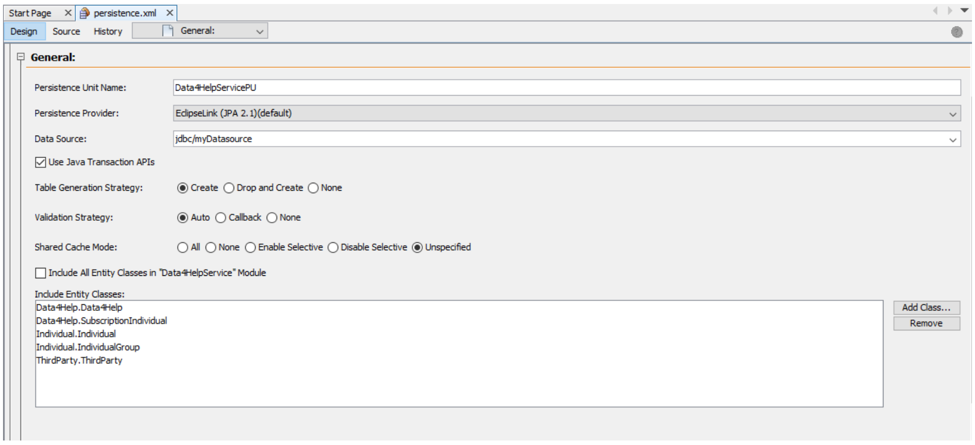
MySQL connector was added to NetBeans and the database myDatasource created, as can be seen in figure 2. Moreover, the connection with the databases was successful.



```
Databases
    MySQL Server at localhost:3306 [root]
        information_schema
        mydatasource
        mydb
        mysql
        performance_schema
        sys
    Java DB
    Drivers
    jdbc:derby://localhost:1527/sample [app on APP]
    jdbc:mysql://localhost:3306/mydatasource?zeroDateTimeBehavior=convertToNull [root o
    jdbc:mysql://localhost:3306/mydb?zeroDateTimeBehavior=convertToNull [root on Defau
        mydb
        Other databases
Web Services
Servers
Maven Repositories
Cloud
Hudson Builders
```

Figure 2: Databases

The only document in which a reference to a database was found was *persistance.xml*. Figure 3 shows the content of the file. *jdbc/myDatasource* is the source of the problems later described.



General:

Persistence Unit Name: Data4HelpServicePU

Persistence Provider: EclipseLink (JPA 2.1)(default)

Data Source: jdbc/myDatasource

☑ Use Java Transaction APIs

Table Generation Strategy: ● Create ○ Drop and Create ○ None

Validation Strategy: ● Auto ○ Callback ○ None

Shared Cache Mode: ○ All ○ None ○ Enable Selective ○ Disable Selective ● Unspecified

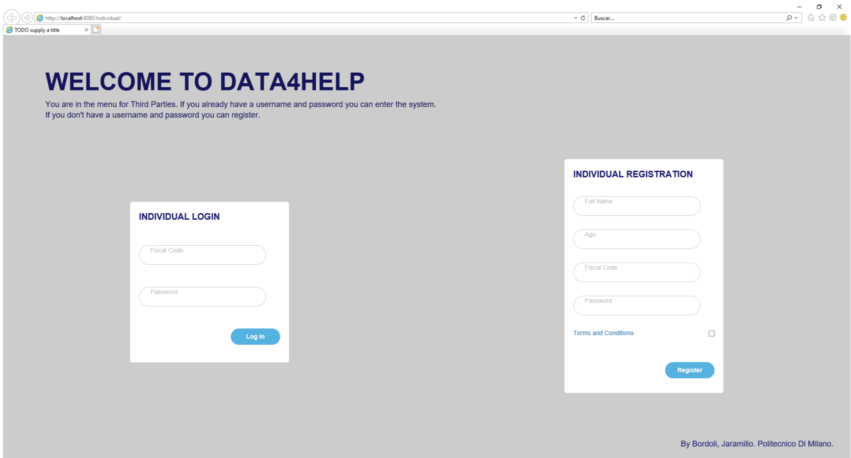
☐ Include All Entity Classes in "Data4HelpService" Module

Include Entity Classes:

Data4Help.Data4Help
Data4Help.SubscriptionIndividual
Individual.Individual
Individual.IndividualGroup
ThirdParty.ThirdParty

Figure 3: persistance.xml file

Running Individual servlet launches the page shown in figure 4.



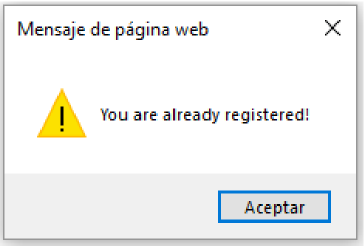
WELCOME TO DATA4HELP

You are in the menu for Third Parties. If you already have a username and password you can enter the system.
If you don't have a username and password you can register.

By Bordoli, Jaramillo. Politecnico Di Milano.

Figure 4: Individual welcome page

Filling the register form and clicking on *Register* leads to the alert shown in figure 5.

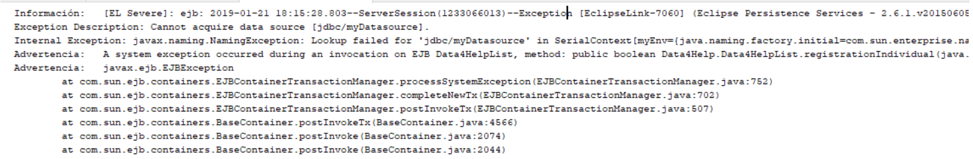


Mensaje de página web

You are already registered!

Figure 5: Alert shown after registering

Figure 6 shows the console of the server, which displays an exception related with the connection to the database *jdbc/myDatasource*. The servlet ThirdParty performs the same way.



```
Información:   [EL Severe]: ejb: 2019-01-21 18:15:28.803--ServerSession(1233066013)--Exception [EclipseLink-7060] (Eclipse Persistence Services - 2.6.1.v2015060
Exception Description: Cannot acquire data source [jdbc/myDatasource].
Internal Exception: javax.naming.NamingException: Lookup failed for 'jdbc/myDatasource' in SerialContext[myEnv={java.naming.factory.initial=com.sun.enterprise.na
Advertencia:   A system exception occurred during an invocation on EJB Data4HelpList, method: public boolean Data4Help.Data4HelpList.registrationIndividual(java.
Advertencia:   javax.ejb.EJBException
	at com.sun.ejb.containers.EJBContainerTransactionManager.processSystemException(EJBContainerTransactionManager.java:752)
	at com.sun.ejb.containers.EJBContainerTransactionManager.completeNewTx(EJBContainerTransactionManager.java:702)
	at com.sun.ejb.containers.EJBContainerTransactionManager.postInvokeTx(EJBContainerTransactionManager.java:507)
	at com.sun.ejb.containers.BaseContainer.postInvokeTx(BaseContainer.java:4566)
	at com.sun.ejb.containers.BaseContainer.postInvoke(BaseContainer.java:2074)
	at com.sun.ejb.containers.BaseContainer.postInvoke(BaseContainer.java:2044)
```

Figure 6: Console of the Glassfish server hosting Individual servlet

2.3 Attempts carried out

The following configurations have been tried, none of them solved the problem.

- Different databases, different prefix for the database
- Creation of JDBC Resource
- Creation of Connection Pool
- Adding the Connection Pool to *web.xml*
- Adding the JDBC Resource to *web.xml*
- Adding the missing libraries to the project ClassPath.

3 Acceptance Test Case

For the moment we was not able to try the application due to problems with database and libraries, if we are able to fix it we will updated this document.

Description requirement	**Status test case**
The system should require unique registration for each individual.	Pending
The system should require unique registration for each third party.	Pending
Third party should make a request to access to individual data.	Pending
Third party should make a request to access to group of individual data.	Pending
The access to individuals' data should be prevent if it is not related to an authorized request.	Pending
The access to group of individuals' data should be prevent if the number of individuals whose data satisfy the request is lower or equal to 1000.	Pending
Third party shall visualize only the individuals' data related to an authorized request.	Pending
Third party shall visualize only the group of individuals' data for which the number of individuals whose data satisfy the request is higher than 1000.	Pending
As soon as one or more parameter is below defined thresholds than the system send a request to the dispatcher in order to ask for an ambulance to be sent to the individual's location.	Pending
The system should define system registration for individuals.	Pending
The system should define system registration for third party.	Pending
The system should define system request for registered third party in order to request for individual's data if the individual is registered in the system.	Pending
The system should define system request for registered third party in order to request for anonymous group of individual's data if the individuals are registered in the system.	Pending
The criteria in order to select the individuals related to a group should be age range and geographical area.	Pending
The system should define system verification for registered individual in order to approve/reject request by the third party for individual's data.	Pending
The system should define system verification in order to approve request by the third party for anonymous group of individual's data if the number of individuals whose satisfy the request is higher than 1000.	Pending
The system should define system verification in order to reject request by the third party for anonymous group of individual's data if the number of individuals whose satisfy the request is lower or equal than 1000.	Pending
The system should define system subscription for automatic retrieve of individuals' data requested by a registered third party for an authorized request.	Pending
The system should define system subscription for automatic retrieve of anonymous group of individual's data requested by a registered third party for an authorized request.	Pending
The system should prohibit access by unauthorized users.	Pending
A minimum of 2 components (e.g., user ID and password) should be required for access.	Pending

Table 1: Requirements test case

4 Additional Point

4.1 Documentation quality

Most of the code comments are autogenerated by the IDE, adding noise and no clarity at all to the code. There are a few comments indicating things that are missing as validation of the user input. There are comments copied directly from StackOverflow.

Regarding the ATD document, there are elements placed in wrong documents (ex: The class diagram found in the ATD should be a reference since that element should be in the DD).

4.2 Tests quality

In the package related with the database operation, Data4HelpService, can be found several tests. These tests only checks the correct functioning of the insert and search in the database, which works well as are implemented by the persistence library.

The most interesting functions, the ones implemented by the system, are out of the scope of the tests.

4.3 Code quality

There are a few flaws in the code.

- Unused variabes
- Unused imports
- Exceptions used as control flow
- Mixed HTML with java code

The last one is related to the architecture of the whole system.

4.4 Notes about architecture

The architecture do not follow a clear structure. The bussiness logic is not contained in a single place and the view of the data is split between the java code and the JSP.

A better approach will be to delegate all the business logic to the servlets, leaving Data4HelpService only for the access to the data and confine the representation of the data (the so-called *view*) entirely to the JSP pages.