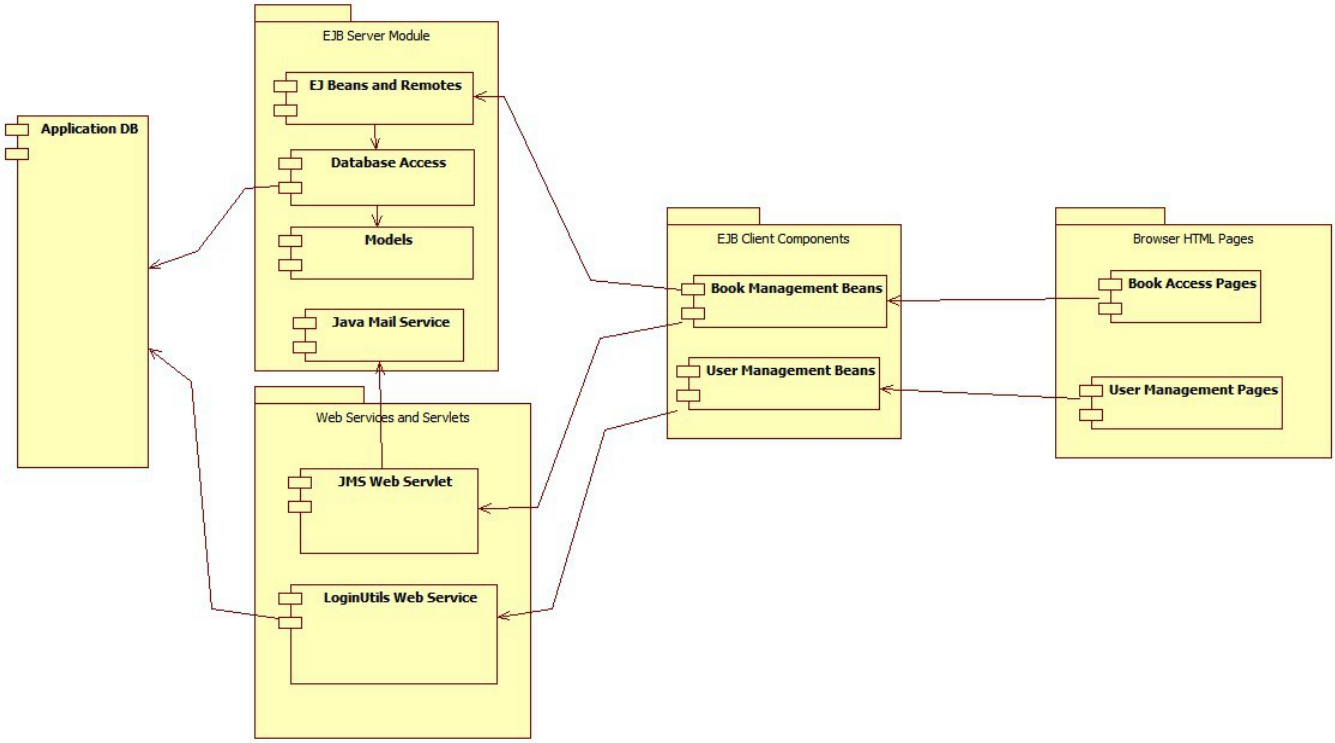
Distributed Systems – assign 4

-Stefan Prisca, group 30441-

Conceptual Diagram

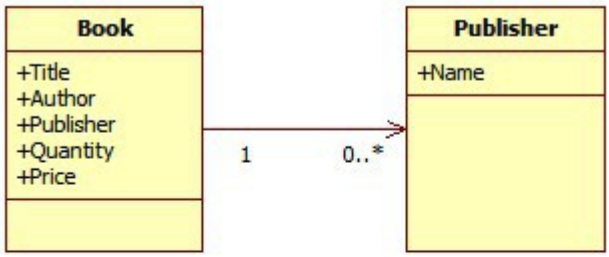


In the above diagram, you can clearly see all the important parts of this application and how they interect with each other:

1. The Applicaiton Database: is the database used to store information about books, which also includes book publishers, and about the application users.
2. The EJB Server Module component: This includes all components that run as part of the EJB server-side module:
    1. The DB Access Logic: this is the persistence API and extra bussines logic that is used to access the DB.
    For this application, Eclipse Link was chosen as the persistence API. It offers a wide range of possible database connections (including SQL Dbs and NonSql dbs). It is also easy to configure through xml files, and has a central maven repository that facilitates the maven build.
    2. Application Models: These are in-memory representations of the data models from the database.
    3. The EJB Bean and Remotes is responsible for providing bussines logic to the client. This contains the class that implements the common communication interface and the common remote interface that will be exported to the client at runtime.
    4. The Java Mail Service component will send mails to users when new books are added.
3. The EJB Client Components package includes the JSF beans that will act as the EJB client in this applicaiton, as well as the beans that will deal with user management.
4. The LoginUtils Web Service is a SOAP web service that handles user management requests like login validation, registering a new user, etc.
    This is separate from the rest of the EJB application, and is a stand-alone web service. Although it is stand-alone, it can be configured to use the same database as the application it is used for, by providing the db connection string to the LoginUtilsFactoryService class, which will create and return the login utils service using the provided db.

5. The JMS Web Servlet is a web servlet that is used to pass JMS messeges to the Java Mail Service. Basically, this will pass the mail that will be sent to all users when a new book is added.
6. Client-side HTML pages: contains the web pages that will be displayed in the browser. These were build using the JSF2 technologies.

Data Design



Books

The database entries that represent the books managed by the applicaiton:

- Title: the title of the book
- Author: the author of the book
- Publisher: the publisher of the book. This is a reference to the Publisher table
- Quantity: the number of books available.
- Price: the price of a book.

Publisher

The representation of book publishers:

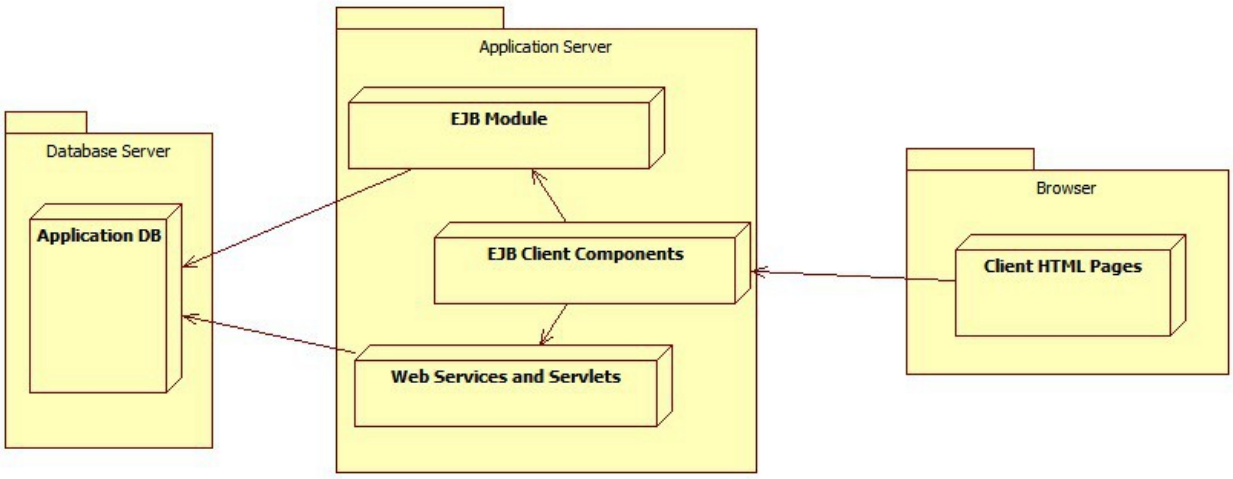
- Name: The name of this publisher.

Users

First data element consists of the table that contains all the user information that is required for the application to run:

- id: database identifier for the current user
- name: name of the user
- birth date: the birth date of the user
- email address: the email address of the user.
- LoginID: the id the user is using to log in the application
- LoginPW: the password used to log in to the application
- type: the type of the user: administrator or regular user.

Deployment Diagram



The above diagram shows how the components will be deployment into the running application:

1. The Application Database will be separate from the application, and can be hosted on any Micrisoft SQL server.
2. The Application Server will contain all bussines login required, including the data base access and models. This will also host the web services and servlets used for login utils and for JMS side of the application.
   Note that the web services and servlets can be deployed on any other AS or Container.
3. The EJB Client Beans (Including the UserManagement beans) will also be deployed in the same Application Serve. As these are developed as a web application, the logic will be available on the entire network.
4. The User HTML pages will be displayed in the user's browser.

Installing the application

It is recommended to use maven tool for building and deploying the project.
All the pom files are configured, so in order to run the build, do the following:

1. Clone the project git repository
2. Go to the App4 folder.
   There should be one project here:
   1. ro.stefanprisca.distsystems.app4.ejb
      This is the JSF RMI client project
3. Before installing the application, you'll have to install the mssql jdbc driver and the web service for user management:
   1. MSSQL JDBC driver:
      1. go to *git_root/utils/mssqljdbc* folder
      2. you should have the *sqljdbc4.jar* driver there.
      3. For windows systems, there is a *mvn_install.bat* file that will install this driver using maven.
      4. For other systems, use the command:
         mvn install:install-file -Dfile=sqljdbc4.jar -DgroupId=com.microsoft.sqlserver -DartifactId=sqljdbc4 -Dversion=4.0 -Dpackaging=jar
   2. UserManagement Web Service:
      You need this installed in the local maven repository because the EJB client project has a dependency on it, as it is required to use the objects provided by the service.
      1. go to *git_root/utils/* folder
      2. you should have the *ro.stefanprisca.distsystems.utils.login* project there.
      3. Also, there is a maven aggregator for installing all the utilities (if there will be more in the future). So, the only thing you need to do is run a *mvn clean install* on that aggregator.
4. After maven has installed the jdbc driver and the usermanagement utils in your local repository, go back to the application folder (App4) and run *mvn clean install*
5. This will install the application, and generate the *ro.stefanprisca.distsystems.app4.client* file in ..\App4\ro.stefanprisca.distsystems.app4.client\target

An alternative to steps 3-5 is using the common maven aggregator from the repository root folder. This will install the App4 project and the login utilities project. Also, you can add any other project to the module list in this aggregator, and it will install it.
Note: you will still need to manually install the jdbc driver as explined in step 3.1 from above.

Deploying the application using Jboss AS 8.1(Wildfly).

In order to deploy the application, you'll have to install Jboss AS 8.1 as a local service first.

After you did this, copy the above mentioned *ro.stefanprisca.distsystems.app4.client.war* and the *utils/usermanager/ro.stefanprisca.distsystems.utils.login.war* files in the standalone deployment folder of JBoss.

Now you just have to start Jboss and access the page:
http://localhost:xxxx/OnlineBookShop/home.xhtml
The mail webservlet is available at:
http://localhost:xxxx/JMSService
Further instructions on how to deploy applications with jboss can be found here.

Opening the application in Eclipse:

The App4 project:
If you have the Eclipse Maven plug-in installed, all you need to do is import the projects as Maven projects.
Note that this is the recommended way to import the projects in Eclipse.

If not, you'll have to select *Create New Project,* deselect the *Use default location* option, and set the project location to the *ro.stefanprisca.distsystems.app4.ejb* folder.

The Usermanagement web service:

This project can also be imported using Eclipse Maven.
You can also import this from eclipse, using the method presented above (by creating a new project in that file location).
Note that after you create the project, you'll have to make sure to import all the required libraries.

Further note: you don't really need to open this project. If you want to run the application, just copy the war file in Tomcat, and the service will perform as expected (tests have been made :-).

Running the application.

Before starting the application, you'll need to also configure the local database, for the users and the books. You will find the specifications for this data bases in the *persistence.xml* file:

- *\App4\ro.stefanprisca.distsystems.app4.ejb\src\main\resources\META-INF\persistence.xml*

Also, you might want to add some accounts in the Users database in order to log in. I usually use the following two:

1. administrator account: id = admin; pw = admin
   Use this account to test the administrator functions
2. regular user account: id = user; pw = user
   Use this acount to test the regular user functions.

Alternativly, you can use the Register service to register a new user.

After you deploy the application using jboss, and you access the application through the browser, you'll have to log in, or to register.

Depending on your log in credentials, you will be redirected to the administrator page or to the regular user page.
Note: If you are not logged in as an administrator, and you try to access the administrator page, you will be redirected to the log in page. Only logged in administrators can access this page!

1. Administrator page:
   Here you will see a table with all the book records from the database.
   You can edit books information, add new books, or add quantity to them.
   Note that when you edit book entries, you'll be redirected to the Edit Book page.

Book Name	Book Author	Book Publisher	Quantity	Price	Edit Book
Some User Edited book	New Test Author	New Publisher 144	83		Edit
Test Time	Stefan Test 2	Test Publisher	1	120$	Edit
UI Modified 1	User Admin	Stefan Pub New	1	1000$	Edit
UI Created	User	New Publisher 144	-1	10$	Edit
Some New Book	Some New Author	Some New Publisher	2	20$	Edit
			0		Edit
			0		Edit
sadf	asdf	sadf	0	sadf	Edit
			0		Edit
asf	asf	asf	0	asdf	Edit

Add New Book

LogOut

   Also, when adding new books, the email service will trigger automatically, and an email will be sent to all your users! So be sure if you want to add a new book or not.

2. Regular user page:
   Here you will find book offers and aditional informations.
   Furthermore, you can:
   1. Take a book that you are interested in by pressing the Take button on the far right of each book entry.
      Note that if you try to take a book with quantity < 1, then an error message will be displayed.
   2. Search your shopping cart to see what books were added to it.
      1. Here you can either confirm your command, go back to the book list, or remove books from your shopping cart.

There are less books than selected!!!!

Book Name	Book Author	Book Publisher	Quantity	Take book!
Some User Edited book	New Test Author	New Publisher 144	83	Take
Test Time	Stefan Test 2	Test Publisher	1	Take
UI Modified 1	User Admin	Stefan Pub New	1	Take
UI Created	User	New Publisher 144	-1	Take
Some New Book	Some New Author	Some New Publisher	2	Take
			0	Take
			0	Take
sadf	asdf	sadf	0	Take
			0	Take
asf	asf	asf	0	Take

See Shopping Cart
LogOut

Book Name	Book Author	Book Publisher	Quantity	Remove book
Some User Edited book	New Test Author	New Publisher 144	2	Remove
Test Time	Stefan Test 2	Test Publisher	1	Remove

Confirm Checkout
Back

References

- JSF2: <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>
- EclipseLink: <https://www.eclipse.org/eclipselink/>
- Jboss: <http://www.jboss.org/>
- Maven: <http://maven.apache.org/>
- JMS: <http://docs.oracle.com/javaee/6/tutorial/doc/bncdq.html>
- EJB: <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>
- Project github repo: <https://github.com/stefanprisca/FacultDistributedSystems>