

**Project Description :**

To appreciate the architecture of Persistence in Java called JPA which was achieved using EclipseLink Which is the JPA provider & Java Server Faces. In this project we have implemented the JPA, JSF & EJB bean. we have managed to Electronic patient record as Hospital Management application.

This project will be definitely help for hospital as to maintain electronic record of patients as it easily maps with database tables & populated the table with help of Object Relational Mapping called ORM. This project can be extended & made more robust suiting for the commercial & analysis purpose. This architecture also allows many other features such caching, pagination, managing many other entities.

**Installation , Compile and Runtime Requirements**

- Microsoft Windows 7 Version 7.0.1 running on x32
- 2GB RAM
- 320 GB Hard Disk
- Intel Core to Duo Processor
- JDK 1.7.0
- Net Beans IDE 7.0.1
- Glass fish web server
- MySql 5.2
- Browser Mozilla Firefox / Chrome
- JPA Provider EclipseLink & JSF libraries

**Database**

This project is already configured to JNDI datasource called ***DataSource*** & database ***itm4515db*** as mention as the requirement for both the JPA providers i.e. JPA & Hibernate ORM.

**Insights and expected results**

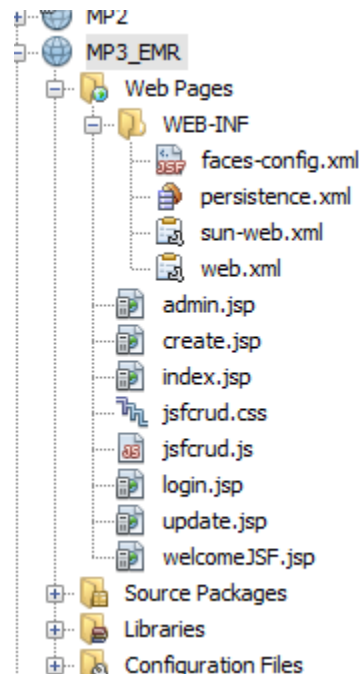
- **JSF** Java Server Faces Java-based web application framework intended to simplify development integration[clarify] of web-based user interfaces.
- JSF is a request-driven MVC web framework for constructing user interfaces using components. As a display technology, JSF 2 uses Facelets. Other view technologies such as XUL can also be employed. JSF 1.x uses JavaServer Pages (JSP) for its display technology. JavaServer Faces is a standardized technology which was formalized in a specification through the Java Community Process. It is part of the Java Platform, Enterprise Edition.
- **JPA** Java Persistence Architecture provided by EclipseLink & Hibernate help in Object relation mapping either by the creating the class of the required columns or the by mapping the database's tables there by its mapping the corresponding columns. The entity are highlight with help of annotation such

@entity denotes entity class,  
@id denotes primary  
@table denotes table name ... & So on

This results in success mapping & fetching the results. Also provides caching.  
Entity is dependent on persistence.xml which holds the JNDI mapping as well  
JPQL & Native queries are used to fetch the results.

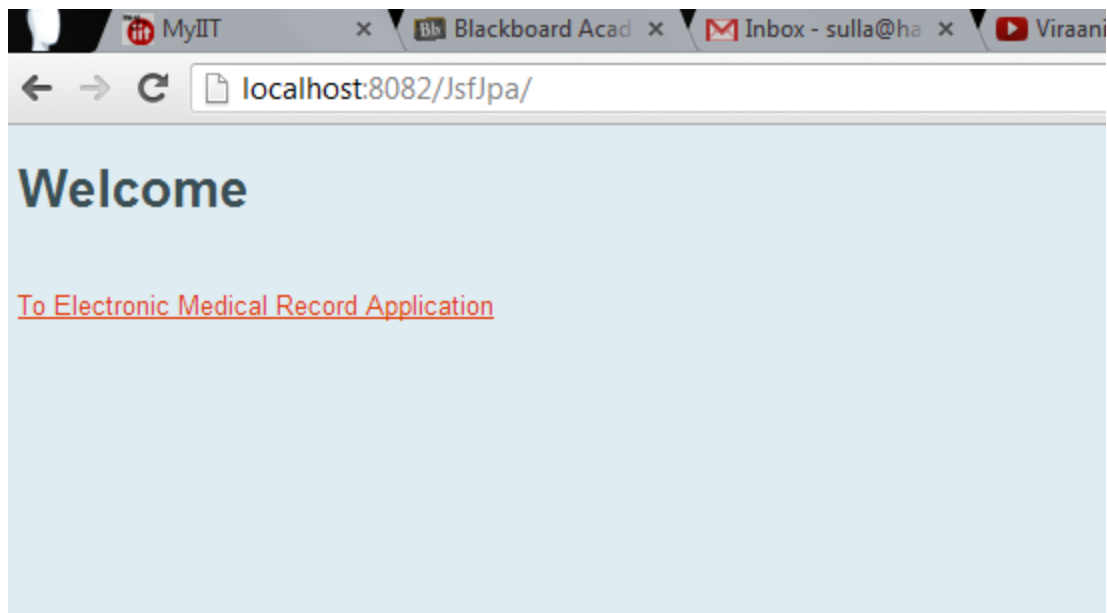
- **JSTL** tags are used to check the conditions. Tag lib like core & function TagLibs are used.
- **HTML5 & CSS** is used to maintained the web pages which makes easier to maintain Layout & web components which are required to displayed the data analytics related to this project
- **JSF & AJAX support** : JSF is often used together with [Ajax](#), a [Rich Internet application](#) technology. Ajax is a combination of technologies that make it possible to create rich user interfaces. The user interface components in Mojarra (the JSF reference implementation[3]) and Apache MyFaces were originally developed for HTML only, and Ajax had to be added via JavaScript. This has changed, however: Because JSF supports multiple output formats, Ajax-enabled components can easily be added to enrich JSF-based user interfaces. The JSF 2.0 specification provides built in support for Ajax by standardizing the Ajax request lifecycle, and providing simple development interfaces to Ajax events, allowing any event triggered by the client to go through proper validation, conversion, and finally method invocation, before returning the result to the browser via an XML DOM update

### Screen captures



**Fig : Project Hierarchy**

**Note : JFT paging standard used here is in JSP  
This makes user to use even JSP tag if needed.**



**Fig : Welcome Page**

```

<phase-listener>
    src.mp3.AuthenticationPhaseListener
</phase-listener>
</lifecycle>
<managed-bean>
    <managed-bean-name>userManager</managed-bean-name>
    <managed-bean-class>src.mp3.UserManager</managed-bean-class>

    <managed-bean-scope>request</managed-bean-scope></managed-bean>

<navigation-rule>
    <navigation-case>
        <from-outcome>login</from-outcome>
        <to-view-id>/login.jsp</to-view-id>
        <redirect/>
    </navigation-case>
    <navigation-case>
        <from-outcome>create</from-outcome>
        <to-view-id>/create.jsp</to-view-id>
    </navigation-case>
    <navigation-case>
        <from-outcome>app-main</from-outcome>
        <to-view-id>/welcomeJSF.jsp</to-view-id>
        <redirect/>
    </navigation-case>
    <navigation-case>
        <from-outcome>app-admin</from-outcome>
        <to-view-id>/admin.jsp</to-view-id>
        <redirect/>
    </navigation-case>

```

**Fig : Faces-config.xml**

## Login

Username:

Password:

[Create New Account](#)

**Fig : Login Page**

## Login

- Login Failed! Username 'sdf' does not exist.

Username:

Password:

[Create New Account](#)

**Fig : Attempting to login with invalid username & password**

## Create a New Account

First Name:	<input type="text" value="shafi"/>	
Last Name:	<input type="text" value="ulla"/>	
Username:	<input type="text" value="shafi"/>	
Password:	<input type="password"/>	
Password (verify):	<input type="password"/>	
Contact Info:	<input type="text" value="iasd"/>	
Current or Prev Employer:	<input type="text" value="dsfsd"/>	
Care Provider:	<input type="text"/>	
Patient Image:	<input type="text" value="sadasd"/>	
Emergency Info:	<input type="text"/>	
Patient History:	<input type="text"/>	
Insurance ID:	<input type="text" value="sadasd"/>	create:InsID: 'sadasd' must be a number between -2147483648 and 2147483647 Example: 9346

**Fig : User or Patient Record creation doing JSF validation on various fields**

First Name:	shafi
Last Name:	ulla
Username:	shafi
Password:	
Password (verify):	
Contact Info:	723129382
Current or Prev Employer:	pepsi co
Care Provider:	mercy hospital
Patient Image:	E:/image.jpg
Emergency Info:	343242343
Patient History:	1. Cought
Insurance ID:	232323

Create

• Username 'shafi' already exists!

**Fig : User already existing**

localhost:8082/JsfJpa/faces/welcomeJSF.jsp

Most Visited Getting Started ITM 422 Career Management C... Sign up :: Splitwise VP Database Connection ... Data

## Listing your record

Welcome shafi ulla! You've been registered since Sat Dec 08 19:18:59 CST 2012.

ID	First Name	Last Name	First Name	Contact Info	Employer	Care Provider	Emergency Info	Image
6	shafi	ulla	shafi	723129382	pepsi co	mercy hospital	343242343	

Logout update

**Fig : Showing Patients record**

localhost:8082/JsfJpa/faces/admin.jsp

Most Visited ☐ Getting Started ☐ ITM 422 ☒ Career Management C... ☐ Sign up :: Splitwise ☒ Database Con

## ADMIN PAGE

Hello Admin, Welcome admin admin! Registered admin since Sat Dec 08 02:45:06 CST 2012.

### List of the Patients details :

ID	First Name	Last Name	First Name	Contact Info	Employer	Care Provider	Emergency Info
1	shafi	shafi	shafi	shafi	shafi	shafi	shafi
4	admin	admin	admin	admin	admin	admin	admin
6	shafi	ulla	shafi	723129382	pepsi co	mercy hospital	343242343

Logout EditByID DeleteALL

**Fig : As Admin, admin can view all the patients record, Edit by ID & also Delete all the records from the entity. The ID is unique here, generated automatically by AUTO INC feature in mysql**

## Welcome shafi shafi!

you can now update your record & save

ID	First Name	Last Name	First Name	Contact Info	Employer	Care Provider	Emergency Info
1			shafi	shafi	shafi	shafi	shafi

First Name:

Last Name:

Contact Info:

Current or Prev Employer:

Care Provider:

Emergency Info:

Patient History:

Insurance ID:

Logout save

**Fig : user able to edit this details & save**

### **Conclusion & Learning :**

- ✓ **Implementing Entity for JPA Eclipse Link & JSF**
- ✓ **JSF field validation**
- ✓ **Creation of facelets**
- ✓ **Using facelets with JPA & JTSL**
- ✓ **Using this architecture the Object Mapping becomes very easy.**
- ✓ **The Web component & database linking becomes easier.**
- ✓ **JSTL tag help in maintained MVC & avoid scriplets**
- ✓ **Less coding with help of mapping & automatic designing in Netbeans in case of JPA**