

DATA Flow Diagram

LogicQ

Bootstrap+JSF



Index.xhtml

Component bind to bean

Bean bind by annotation

```
<td><h:outputText value="User Id : " /></td>
<td>
<h:inputText value="#{userMB.id}" required="true" label="User ID :"/>
</td>
```

```
@ManagedBean(name = "userMB")
@RequestScoped
public class UserBean implements Serializable {

    private static final long serialVersionUID = 1L;

    private int id;
    private String name;
    private String surname;
    private String addUser;
    private String reset;
```

This use only for page + component binding
No business logic here

```
@ManagedBean(name = "userMB")
@RequestScoped
public class UserBean implements Serializable {

    private static final long serialVersionUID = 1L;

    private int id;
    private String name;
    private String surname;
    private String addUser;
    private String reset;
```

It use VO object for holding values related to UI

UserVO

```
I_user=LogicqContextProvider.getApplicationContext().getBean(UserVO.class);
```

Use for UI business logic

1. Add() business logic will handle by UserManagedBean.java

```
~/
public class UserManagedBean {

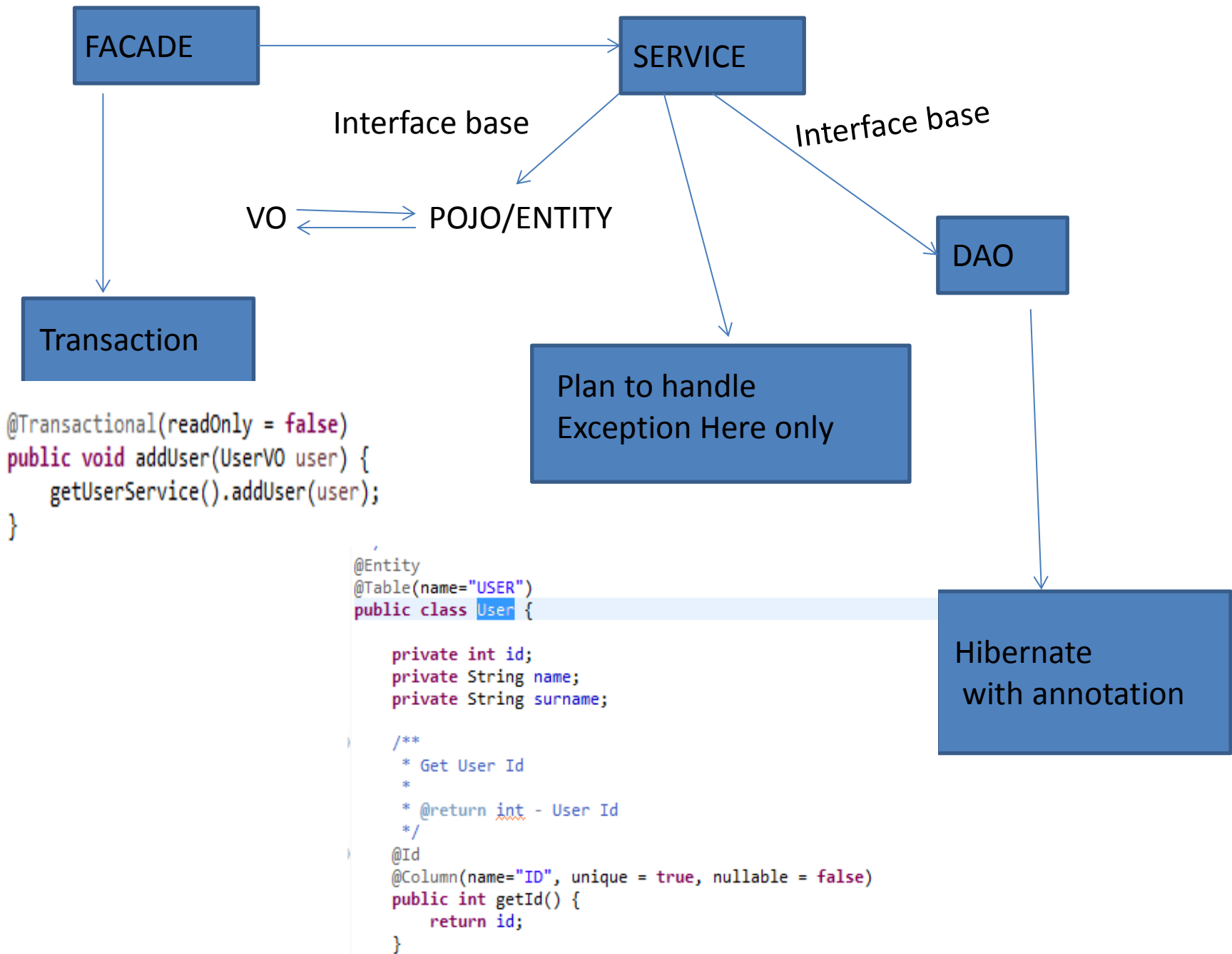
    public String addUser(UserVO p_user) {
        UserBeanHelper userbeanhelper=LogicqContextProvider.getApplicationContext().getBean(UserBeanHelper.class);
        userbeanhelper.addUser(p_user);
        return "sucess";
    }
}
```

UserManagedBean.java

UserBeanHelper.java

1. Call to Façade layer of Host and façade will be interface base implimentation
2. Conversion of Ulbean to VO Object and vise-versa
3. It will send VO /Httprequest and HttpResponse to Host

```
public class UserBeanHelper {  
  
    @Autowired  
    IUserFacade userFacade;  
  
    public IUserFacade getUserFacade() {  
        return userFacade;  
    }  
  
    public void setUserFacade(IUserFacade userFacade) {  
        this.userFacade = userFacade;  
    }  
  
    public void addUser(UserVO p_user) {  
        userFacade.addUser(p_user);  
    }  
  
    public UserVO conversionUItoVO(UserBean p_userbean, UserVO p_user) {  
        p_user.setId(p_userbean.getId());  
        p_user.setName(p_userbean.getName());  
        p_user.setSurname(p_userbean.getSurname());  
        return p_user;  
    }  
}
```



DAO



```
public class AbstractDAO<T> {  
  
    @Autowired  
    private SessionFactory sessionFactory;  
  
    /**  
     * Get Hibernate Session Factory  
     *  
     * @return SessionFactory - Hibernate Session Factory  
     */  
    public SessionFactory getSessionFactory() {  
        return sessionFactory;  
    }  
  
    /**  
     * Set Hibernate Session Factory  
     *  
     * @param SessionFactory - Hibernate Session Factory  
     */  
    public void setSessionFactory(SessionFactory sessionFactory) {  
        this.sessionFactory = sessionFactory;  
    }  
}
```

Please check details with source code related to user