|  |  |  |  |
| --- | --- | --- | --- |
| **Ex. No. 07** | **DESIGNING JAVA PERSISTANCE API APPLICATION – BANKING SYSTEM** | | |
| Date of Exercise |  | Date of Output Verification |  |

**Aim**

To develop and deploy the enterprise java persistence API for banking system using J2EE programming.

**Description**

Java Persistence API (JPA) provides POJO (Plain Old Java Object) standard and object relational mapping (OR mapping) for data persistence among applications. Persistence, which deals with storing and retrieving of application data, can now be programmed with Java Persistence API starting from EJB 3.0 as a result of JSR 220. This API has borrowed many of the concepts and standards from leading persistence frameworks like Toplink (from Oracle) and Hibernate (from JBoss). One of the great benefits of JPA is that it is an independent API and can nicely integrate with J2EE as well as J2SE applications. [POJO – Plain Old Java Object is a term used to refer Java objects that do not extend or implement some specialized classes. Therefore, all normal Java objects are POJO’s only. The following classes are not POJO classes.

**Procedure**

1. Create a Java EE Project in NetBeans IDE
2. Create a table with a primary key
3. Create a new entity bean for database for that table
   1. Give package name
   2. Add new connection
   3. Give the connection name
   4. Select the table and import
   5. Make sure that it the persistence is checked
4. Create session bean or that entity class
5. Create a java class in the ejb module to handle all the business logic
6. In the web module, create a servlet to send and receive the requests from jsp and servlet

**Program**

Header.jsp

<nav class="navbar navbar-default ">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="#">WS | Banking System</a>

</div>

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav">

<li class="active"><a href="index.jsp">Home <span class="sr-only">(current)</span></a></li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"> Extras <span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="#">Action</a></li>

<li><a href="#">Another action</a></li>

<li role="separator" class="divider"></li>

<li><a href="#">Separated link</a></li>

<li role="separator" class="divider"></li>

<li><a href="#">One more separated link</a></li>

</ul>

</li>

</ul>

<form class="navbar-form navbar-left" role="search">

<div class="form-group">

<input type="text" class="form-control" placeholder="Search">

</div>

<button type="submit" class="btn btn-default">Submit</button>

</form>

<ul class="nav navbar-nav navbar-right">

<li><a href="#">Options<span class=""></span></a></li>

<%

if (session.getAttribute("name") == null) {

%>

<li><a href="login.jsp">Login <span class=""></span></a></li>

<%

} else if (session.getAttribute("name") == "failed") {

%>

<li><a href="login.jsp">Login <span class=""></span></a></li>

<%

} else {

%>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"> <%=session.getAttribute("name")%> <span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="#">Account Settings</a></li>

<li><a href="#">Privacy Settings</a></li>

<li role="separator" class="divider"></li>

<li><a href="#" onclick="checkBalance()">Check Balance</a></li>

<li role="separator" class="divider"></li>

<li><a href="logout.jsp">Logout</a></li>

</ul>

</li>

<%

}

%>

</ul>

</div>

</div>

</nav>

Footer.jsp

<style>

#footer{

position: fixed;

bottom: 0px;

text-align: center;

line-height: 1.2em;

background: #357ebd;

color: whitesmoke;

font-size: 1.2em;

right: 0px;

left: 0px;

letter-spacing: 1.5px;

padding-top: 12px;

}

</style>

<div id="footer">

<p class="footer-block">Developed by WS</p>

</div>

Index.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>WS | Banking System</title>

<script src="js/jquery-1.11.3.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/script.js"></script>

<link rel="stylesheet" href="css/bootstrap.min.css" />

<link rel="stylesheet" href="css/styles.css" />

</head>

<body>

<jsp:include flush="true" page="header.jsp"></jsp:include>

<jsp:include flush="true" page="index\_body.jsp"></jsp:include>

<div style="margin-bottom: 50px;"></div>

<jsp:include flush="true" page="footer.jsp"></jsp:include>

</body>

</html>

Index\_body.jsp

<script>

$(document).ready(function () {

$("#errormessage").hide();

});

function checkBalance() {

$("#errormessage").slideUp();

$.ajax({

type: "get",

url: "Action?Balance=true",

data: "name=test",

success: function (data) {

$("#balance").html("Balance in your Account is: " + data + " Rs/-");

}

});

}

function depositMoney() {

$("#errormessage").slideUp();

var form1 = $('#depositForm').serialize();

$.ajax({

type: "post",

url: "Action?Deposit=true",

data: form1,

success: function (data) {

$("#balance").html("Balance in your Account is: " + data + " Rs/-");

}

});

}

function transferMoney() {

$("#errormessage").slideUp();

var form1 = $('#transferForm').serialize();

$.ajax({

type: "post",

url: "Action?Transfer=true",

data: form1,

success: function (data) {

if (data != "false") {

$("#balance").html("Balance in your Account is: " + data + " Rs/-");

}

else {

$("#errormessage").slideDown();

}

}

});

}

function withdrawMoney() {

$("#errormessage").slideUp();

var form1 = $('#withdrawForm').serialize();

$.ajax({

type: "post",

url: "Action?Withdraw=true",

data: form1,

success: function (data) {

if (data != "false") {

$("#balance").html("Balance in your Account is: " + data + " Rs/-");

} else {

$("#errormessage").slideDown();

}

}

});

}

</script>

<div class='container alert alert-danger' id="errormessage">Error, Try Again!....</div>

<%

if (session.getAttribute("name") == null || session.getAttribute("name") == "failed") {

%>

<div class="panel col-sm-4 col-sm-offset-4 panel-default">

<div class="panel-heading panel-success">Note</div>

<div class="panel-body">

To Access your Bank Facilities Please <a class="btn btn-success" href="login.jsp">Signin</a>

</div>

</div>

<%

} else {

%>

<div class="panel col-sm-4 col-sm-offset-4 panel-default">

<div class="panel-heading panel-success">Check Balance</div>

<div class="panel-body" id="balance">

To Check your Balance <a class="btn btn-success" onclick="checkBalance()">Click Here</a>

</div>

</div>

<div class="panel col-sm-4 col-sm-offset-4 panel-default">

<div class="panel-heading panel-success">Deposit Money</div>

<div class="panel-body">

<form class="form-horizontal" role="form" id="depositForm">

<div class="form-group">

<label class="control-label col-sm-3" for="depositamount">Amount</label>

<div class="col-sm-9">

<input type="text" class="form-control" id="depositamount" name="depositamount" placeholder="Enter Amount to Deposit">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8 col-sm-10">

<button type="button" class="btn btn-default btn-primary" onclick="depositMoney()">Deposit</button>

</div>

</div>

</form>

</div>

</div>

<div class="panel col-sm-4 col-sm-offset-4 panel-default">

<div class="panel-heading panel-success">Withdraw Money</div>

<div class="panel-body">

<form class="form-horizontal" role="form" id="withdrawForm">

<div class="form-group">

<label class="control-label col-sm-3" for="withdrawamount">Amount</label>

<div class="col-sm-9">

<input type="text" class="form-control" id="withdrawamount" name="withdrawamount" placeholder="Enter Amount to Deposit">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8 col-sm-10">

<button type="button" class="btn btn-default btn-primary" onclick="withdrawMoney()">Withdraw</button>

</div>

</div>

</form>

</div>

</div>

<div class="panel col-sm-4 col-sm-offset-4 panel-default">

<div class="panel-heading panel-success">Transfer Money</div>

<div class="panel-body">

<form class="form-horizontal" role="form" id="transferForm">

<div class="form-group">

<label class="control-label col-sm-3" for="transferusername">Username:</label>

<div class="col-sm-9">

<input type="text" class="form-control" id="transferusername" name="transferusername" placeholder="Enter Transfer Username">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-3" for="amount">Amount: </label>

<div class="col-sm-9">

<input type="text" class="form-control" id="amount" name="amount" placeholder="Enter Transfer Amount">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8 col-sm-10">

<button type="button" class="btn btn-default btn-primary" onclick="transferMoney()">Transfer</button>

</div>

</div>

</form>

</div>

</div>

<%

}

%>

Login.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>WS | Login</title>

<script src="js/jquery-1.11.3.min.js"> </script>

<script src="js/bootstrap.min.js"> </script>

<script src="js/script.js"> </script>

<link rel="stylesheet" href="css/bootstrap.min.css" />

<link rel="stylesheet" href="css/styles.css" />

</head>

<body>

<jsp:include flush="true" page="header.jsp"></jsp:include>

<jsp:include flush="true" page="login\_body.jsp"></jsp:include>

<div style="margin-bottom: 30px;"></div>

<jsp:include flush="true" page="footer.jsp"></jsp:include>

</body>

</html>

Login\_body.jsp

<script>

$(document).ready(function () {

$("#signupform").hide();

$("#signinclick").click(function () {

$("#signupform").slideUp();

$("#signinform").slideDown();

});

$("#signupclick").click(function () {

$("#signinform").slideUp();

$("#signupform").slideDown();

});

});

</script>

<style>

body{

background-image: url('img/login-background.jpg');

overflow-x: hidden;

}

#body form, #body{

width: 450px;

margin: 0px auto;

padding: 0px;

}

#body form{

background: rgba(0,0,0,0.4);

padding: 7px;

margin-top: 40px;

border-radius: 2px;

color: white;

border:1px white solid;

box-shadow: 0 0 5px black;

transition: box-shadow 1s ease;

}

#body form:hover{

box-shadow: 0 0 15px black;

}

#body{

margin-top: 50px;

}

#body .heading{

font-size: 2em;

margin:0 auto;

}

#body ul{

margin-left: 70px

}

#body ul li{

display: inline;

padding: 10px 15px;

cursor: pointer;

background: rgba(0,0,0,0.3);

transition: background 0.3s ease;

margin-left: 20px;

border-radius: 2px;

color: white;

}

#body ul li:hover{

background: rgba(0,0,255,0.2);

color: black;

}

</style>

<%

if (session.getAttribute("name") != null) {

if (session.getAttribute("name") == "failed") {

%>

<div class='container alert alert-danger'>Error, Try Again!....</div>

<%

}

}

%>

<div id="body">

<ul id="tabs">

<li id="signinclick">SignIn</li>

<li id="signupclick">SignUp</li>

</ul>

<form id="signinform" class="form-horizontal" action="LoginSignup" method="post">

<p class="heading">Login</p>

<div class="form-group">

<label for="signin\_username" class="control-label col-sm-3">Username</label>

<div class="col-sm-8">

<input type="text" name="signin\_username" id="signin\_username" placeholder="Enter Username" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="signin\_password" class="control-label col-sm-3">Password</label>

<div class="col-sm-8">

<input type="password" name="signin\_password" id="signin\_password" placeholder="Enter Password" class="form-control" required>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8">

<input type="submit" name="submit\_button" class="btn btn-default" value="SignIn" />

</div>

</div>

</form>

<form id="signupform" class="form-horizontal" action="LoginSignup" method="post">

<p class="heading">Sign Up</p>

<div class="form-group">

<label for="signup\_fullname" class="control-label col-sm-3">Full Name</label>

<div class="col-sm-8">

<input type="text" name="signup\_fullname" id="signup\_fullname" placeholder="Enter Full Name" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="signup\_username" class="control-label col-sm-3">Username</label>

<div class="col-sm-8">

<input type="text" name="signup\_username" id="signup\_username" placeholder="Enter Username" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="signup\_email" class="control-label col-sm-3">Email</label>

<div class="col-sm-8">

<input type="email" name="signup\_email" id="signup\_email" placeholder="Enter Email" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="signup\_dob" class="control-label col-sm-3">Date of Birth</label>

<div class="col-sm-8">

<input type="date" name="signup\_dob" id="signup\_dob" class="form-control">

</div>

</div>

<div class="form-group">

<label for="signup\_Password" class="control-label col-sm-3">Password</label>

<div class="col-sm-8">

<input type="password" name="signup\_Password" id="signup\_Password" placeholder="Enter Password" class="form-control" required>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8">

<input type="submit" name="submit\_button" class="btn btn-default" value="SignUp" />

</div>

</div>

</form>

</div>

Logout.jsp

<%

session.invalidate();

response.sendRedirect("index.jsp");

%>

LoginSignup.java

import bankAction.JavaClass;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class LoginSignup extends HttpServlet {

JavaClass jc;

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

try {

response.setContentType("text/html;charset=UTF-8");

HttpSession session = request.getSession();

jc = new JavaClass();

String sbutton = request.getParameter("submit\_button");

switch (sbutton) {

case "SignIn": {

String username = request.getParameter("signin\_username");

String password = request.getParameter("signin\_password");

String name = jc.signin(username, password);

session.setAttribute("name", name);

session.setAttribute("username", username);

if (!"failed".equals(name)) {

RequestDispatcher rd = request.getRequestDispatcher("index.jsp");

rd.forward(request, response);

} else {

RequestDispatcher rd = request.getRequestDispatcher("login.jsp");

rd.forward(request, response);

}

break;

}

case "SignUp": {

String name = request.getParameter("signup\_fullname");

String username = request.getParameter("signup\_username");

String email = request.getParameter("signup\_email");

String dob = request.getParameter("signup\_dob");

String pwd = request.getParameter("signup\_Password");

String return\_value = jc.signup(name, username, email, dob, pwd);

session.setAttribute("name", return\_value);

session.setAttribute("username", username);

if (!"failed".equals(return\_value)) {

RequestDispatcher rd = request.getRequestDispatcher("index.jsp");

rd.forward(request, response);

} else {

RequestDispatcher rd = request.getRequestDispatcher("login.jsp");

rd.include(request, response);

}

break;

}

default:

session.setAttribute("name", "failed");

RequestDispatcher rd = request.getRequestDispatcher("login.jsp");

rd.forward(request, response);

break;

}

} catch (IOException | ServletException ex) {

System.out.println(ex);

}

}

}

Action.java

import bankAction.JavaClass;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class Action extends HttpServlet {

JavaClass jc;

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

jc = new JavaClass();

response.setContentType("text/html;charset=UTF-8");

response.setCharacterEncoding("utf-8");

try {

HttpSession session = request.getSession();

if (request.getParameter("Balance") != null) {

String bal = jc.Balance((String) session.getAttribute("username")) + "";

response.getWriter().write(bal);

} else if (request.getParameter("Transfer") != null) {

String flag = "false";

if (request.getParameter("amount").length() >= 1) {

flag = jc.transfer((String) session.getAttribute("username"), request.getParameter("transferusername"), Integer.parseInt(request.getParameter("amount")));

}

if ("true".equals(flag)) {

String bal = jc.Balance((String) session.getAttribute("username")) + "";

response.getWriter().write(bal);

} else {

response.getWriter().write(flag);

}

} else if (request.getParameter("Deposit") != null) {

jc.deposit((String) session.getAttribute("username"), Integer.parseInt(request.getParameter("depositamount")));

String bal = jc.Balance((String) session.getAttribute("username")) + "";

response.getWriter().write(bal);

} else if (request.getParameter("Withdraw") != null) {

String flag = jc.withdraw((String) session.getAttribute("username"), Integer.parseInt(request.getParameter("withdrawamount")));

if ("true".equals(flag)) {

String bal = jc.Balance((String) session.getAttribute("username")) + "";

response.getWriter().write(bal);

} else {

response.getWriter().write(flag);

}

}

} catch (IOException | NumberFormatException e) {

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

processRequest(request, response);

}

}

AbstractFacade.java

package bankAction;

import java.util.List;

import javax.persistence.EntityManager;

public abstract class AbstractFacade<T> {

private Class<T> entityClass;

public AbstractFacade(Class<T> entityClass) {

this.entityClass = entityClass;

}

protected abstract EntityManager getEntityManager();

public void create(T entity) {

getEntityManager().persist(entity);

}

public void edit(T entity) {

getEntityManager().merge(entity);

}

public void remove(T entity) {

getEntityManager().remove(getEntityManager().merge(entity));

}

public T find(Object id) {

return getEntityManager().find(entityClass, id);

}

public List<T> findAll() {

javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();

cq.select(cq.from(entityClass));

return getEntityManager().createQuery(cq).getResultList();

}

public List<T> findRange(int[] range) {

javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();

cq.select(cq.from(entityClass));

javax.persistence.Query q = getEntityManager().createQuery(cq);

q.setMaxResults(range[1] - range[0] + 1);

q.setFirstResult(range[0]);

return q.getResultList();

}

public int count() {

javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();

javax.persistence.criteria.Root<T> rt = cq.from(entityClass);

cq.select(getEntityManager().getCriteriaBuilder().count(rt));

javax.persistence.Query q = getEntityManager().createQuery(cq);

return ((Long) q.getSingleResult()).intValue();

}

}

Bankaccount.java

package bankAction;

import java.io.Serializable;

import javax.persistence.Basic;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.Table;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.Size;

import javax.xml.bind.annotation.XmlRootElement;

@Entity

@Table(name = "BANKACCOUNT")

@XmlRootElement

@NamedQueries({

@NamedQuery(name = "Bankaccount.findAll", query = "SELECT b FROM Bankaccount b"),

@NamedQuery(name = "Bankaccount.findByName", query = "SELECT b FROM Bankaccount b WHERE b.name = :name"),

@NamedQuery(name = "Bankaccount.findByUsername", query = "SELECT b FROM Bankaccount b WHERE b.username = :username"),

@NamedQuery(name = "Bankaccount.findByEmail", query = "SELECT b FROM Bankaccount b WHERE b.email = :email"),

@NamedQuery(name = "Bankaccount.findByDob", query = "SELECT b FROM Bankaccount b WHERE b.dob = :dob"),

@NamedQuery(name = "Bankaccount.findByPassword", query = "SELECT b FROM Bankaccount b WHERE b.password = :password"),

@NamedQuery(name = "Bankaccount.findByBalance", query = "SELECT b FROM Bankaccount b WHERE b.balance = :balance")})

public class Bankaccount implements Serializable {

private static final long serialVersionUID = 1L;

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 50)

@Column(name = "NAME")

private String name;

@Id

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 70)

@Column(name = "USERNAME")

private String username;

// @Pattern(regexp="[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+(?:\\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*@(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?", message="Invalid email")//if the field contains email address consider using this annotation to enforce field validation

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 220)

@Column(name = "EMAIL")

private String email;

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 20)

@Column(name = "DOB")

private String dob;

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 220)

@Column(name = "PASSWORD")

private String password;

@Basic(optional = false)

@NotNull

@Column(name = "BALANCE")

private int balance;

public Bankaccount() {

}

public Bankaccount(String username) {

this.username = username;

}

public Bankaccount(String username, String name, String email, String dob, String password, int balance) {

this.username = username;

this.name = name;

this.email = email;

this.dob = dob;

this.password = password;

this.balance = balance;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getDob() {

return dob;

}

public void setDob(String dob) {

this.dob = dob;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public int getBalance() {

return balance;

}

public void setBalance(int balance) {

this.balance = balance;

}

@Override

public int hashCode() {

int hash = 0;

hash += (username != null ? username.hashCode() : 0);

return hash;

}

@Override

public boolean equals(Object object) {

// TODO: Warning - this method won't work in the case the id fields are not set

if (!(object instanceof Bankaccount)) {

return false;

}

Bankaccount other = (Bankaccount) object;

if ((this.username == null && other.username != null) || (this.username != null && !this.username.equals(other.username))) {

return false;

}

return true;

}

@Override

public String toString() {

return "bankAction.Bankaccount[ username=" + username + " ]";

}

}

BankaccountFacade.java

package bankAction;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

@Stateless

public class BankaccountFacade extends AbstractFacade<Bankaccount> implements BankaccountFacadeLocal {

@PersistenceContext(unitName = "Exp\_7\_-\_Banking\_System\_using\_Java\_Persistence\_API-ejbPU")

private EntityManager em;

@Override

protected EntityManager getEntityManager() {

return em;

}

public BankaccountFacade() {

super(Bankaccount.class);

}

}

BankaccountFacadeLocal.java

package bankAction;

import java.util.List;

import javax.ejb.Local;

@Local

public interface BankaccountFacadeLocal {

void create(Bankaccount bankaccount);

void edit(Bankaccount bankaccount);

void remove(Bankaccount bankaccount);

Bankaccount find(Object id);

List<Bankaccount> findAll();

List<Bankaccount> findRange(int[] range);

int count();

}

JavaClass.java

package bankAction;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

public class JavaClass {

BankaccountFacadeLocal bankaccountFacade = lookupBankaccountFacadeLocal();

public String signin(String username, String password) {

if (bankaccountFacade.find(username) == null) {

return "failed";

} else {

return bankaccountFacade.find(username).getName();

}

}

public String signup(String name, String username, String email, String dob, String password) {

Bankaccount a = new Bankaccount();

a.setName(name);

a.setUsername(username);

a.setEmail(email);

a.setDob(dob);

a.setPassword(password);

try {

bankaccountFacade.create(a);

} catch (Exception e) {

return "failed";

}

return name;

}

public int Balance(String username) {

if (bankaccountFacade.find(username) == null) {

return -1;

}

return bankaccountFacade.find(username).getBalance();

}

public String transfer(String username, String transfername, int amount) {

if (bankaccountFacade.find(transfername) != null) {

Bankaccount a = bankaccountFacade.find(username);

if (a.getBalance() >= amount) {

a.setBalance(a.getBalance() - amount);

bankaccountFacade.edit(a);

a = bankaccountFacade.find(transfername);

a.setBalance(a.getBalance() + amount);

bankaccountFacade.edit(a);

return "true";

} else {

return "false";

}

}

return "false";

}

public void deposit(String username, int amount) {

Bankaccount a = bankaccountFacade.find(username);

a.setBalance(a.getBalance() + amount);

bankaccountFacade.edit(a);

}

public String withdraw(String username, int amount) {

Bankaccount a = bankaccountFacade.find(username);

if (a.getBalance() >= amount) {

a.setBalance(a.getBalance() - amount);

bankaccountFacade.edit(a);

return "true";

}

return "false";

}

private BankaccountFacadeLocal lookupBankaccountFacadeLocal() {

try {

Context c = new InitialContext();

return (BankaccountFacadeLocal) c.lookup("java:global/Exp\_7\_-\_Banking\_System\_using\_Java\_Persistence\_API/Exp\_7\_-\_Banking\_System\_using\_Java\_Persistence\_API-ejb/BankaccountFacade!bankAction.BankaccountFacadeLocal");

} catch (NamingException ne) {

Logger.getLogger(getClass().getName()).log(Level.SEVERE, "exception caught", ne);

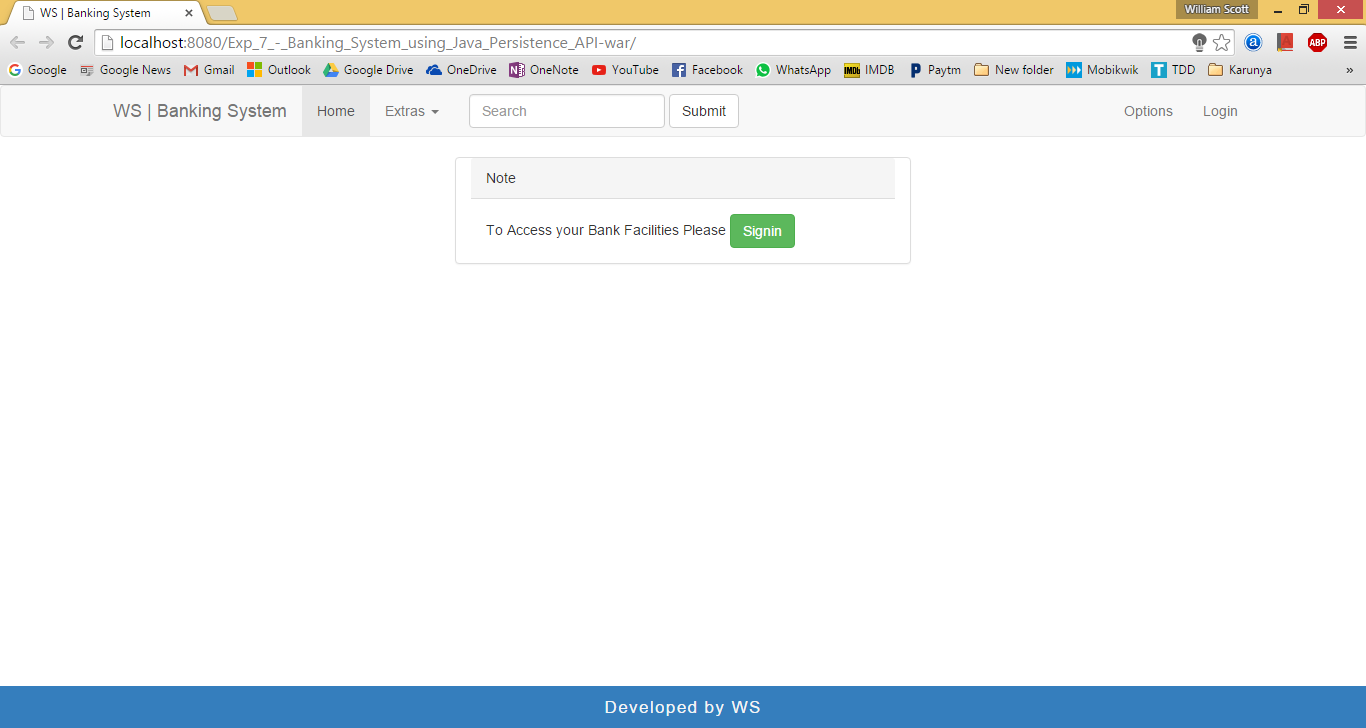
throw new RuntimeException(ne);

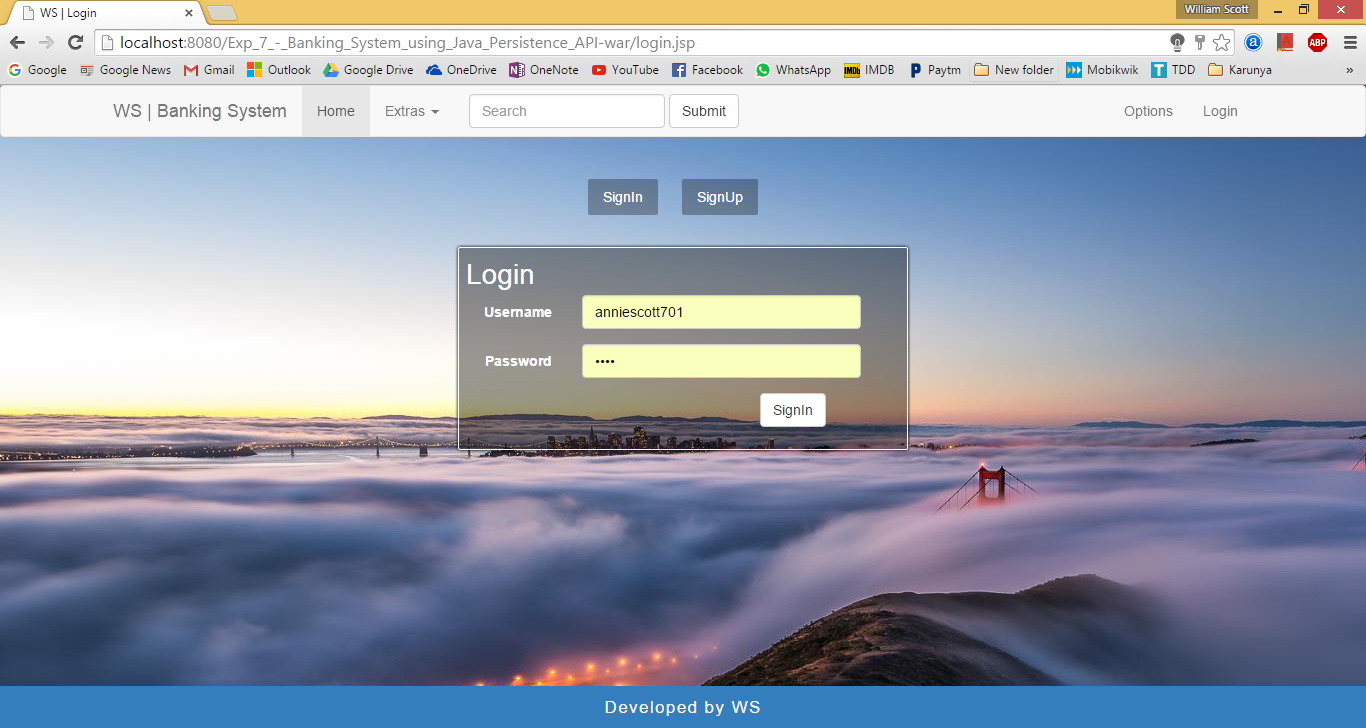
}

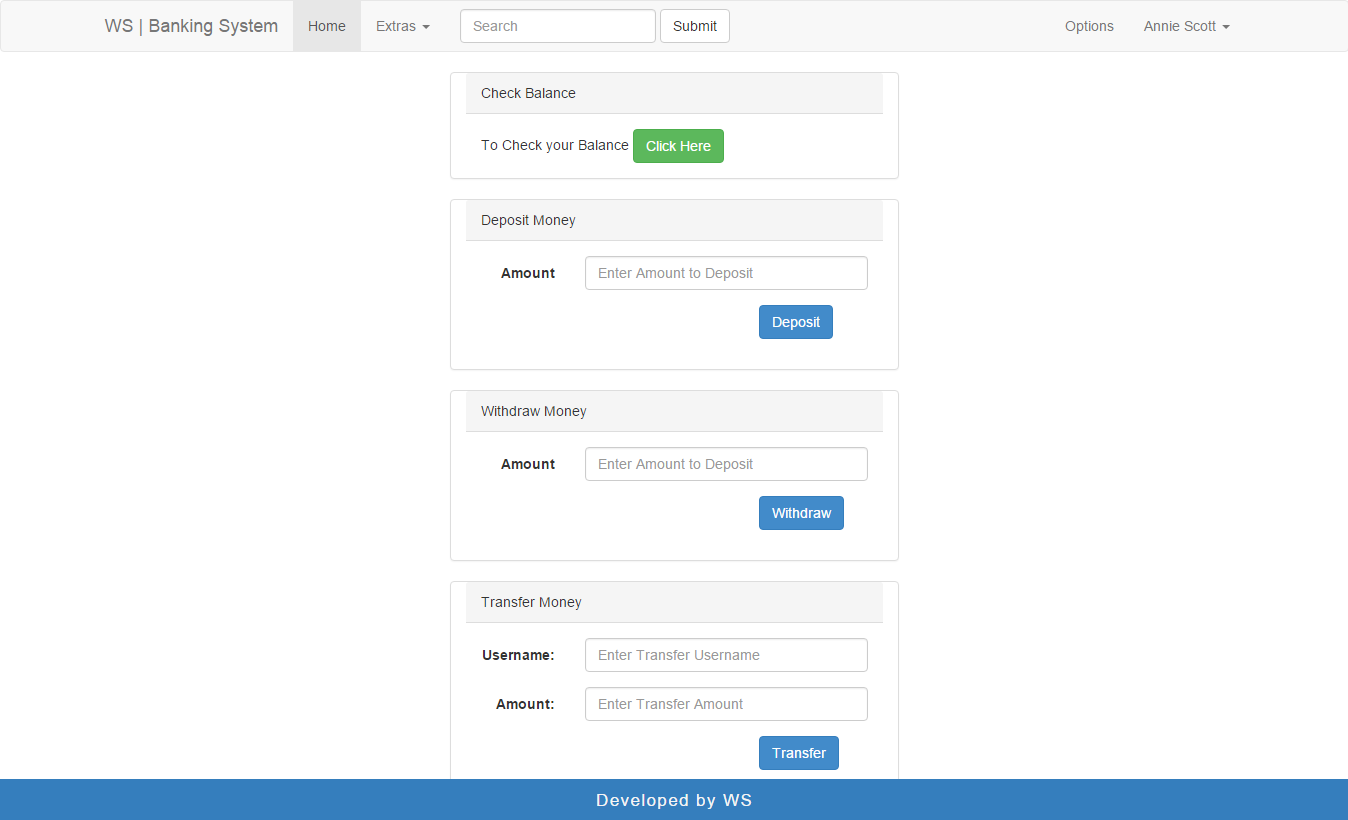
}

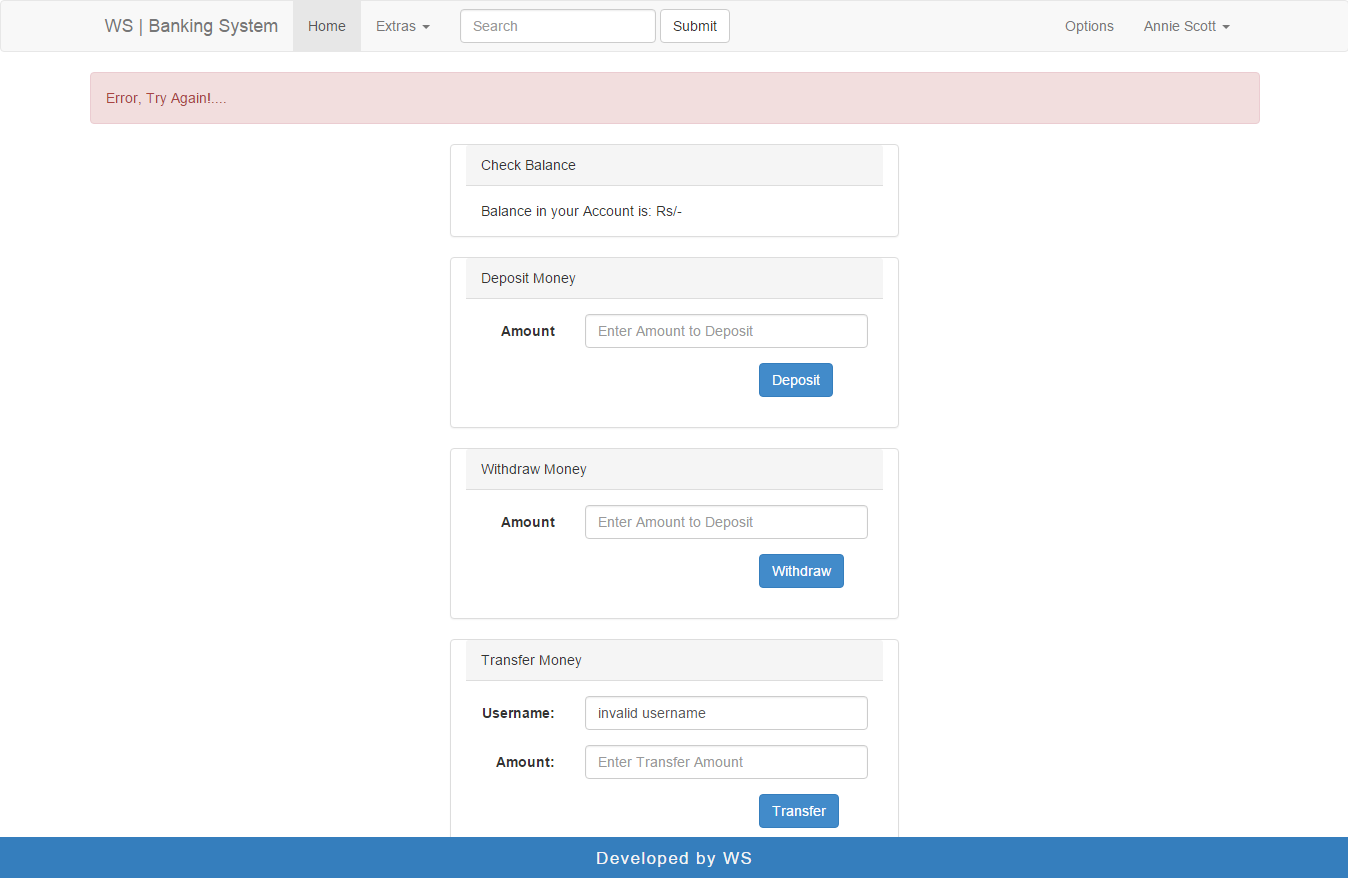
}

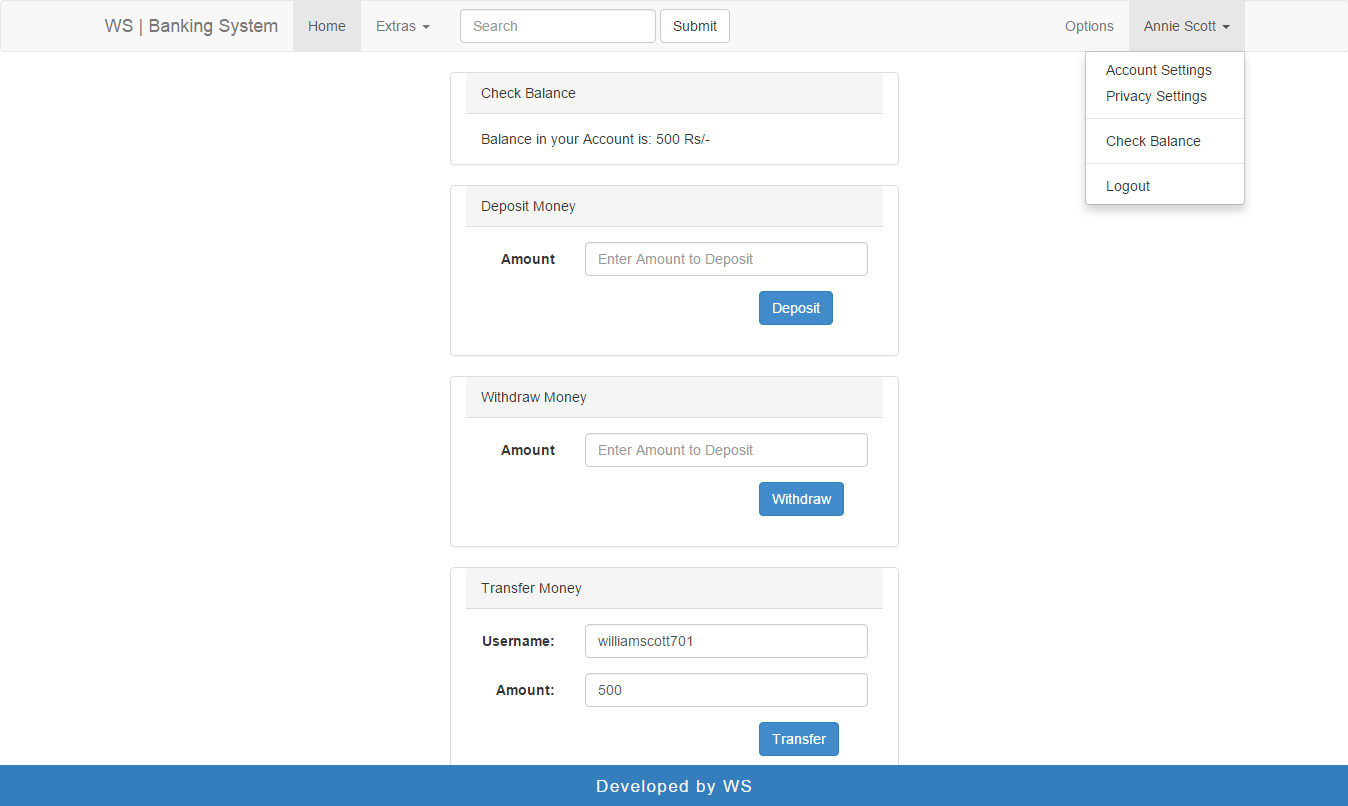
**Output**

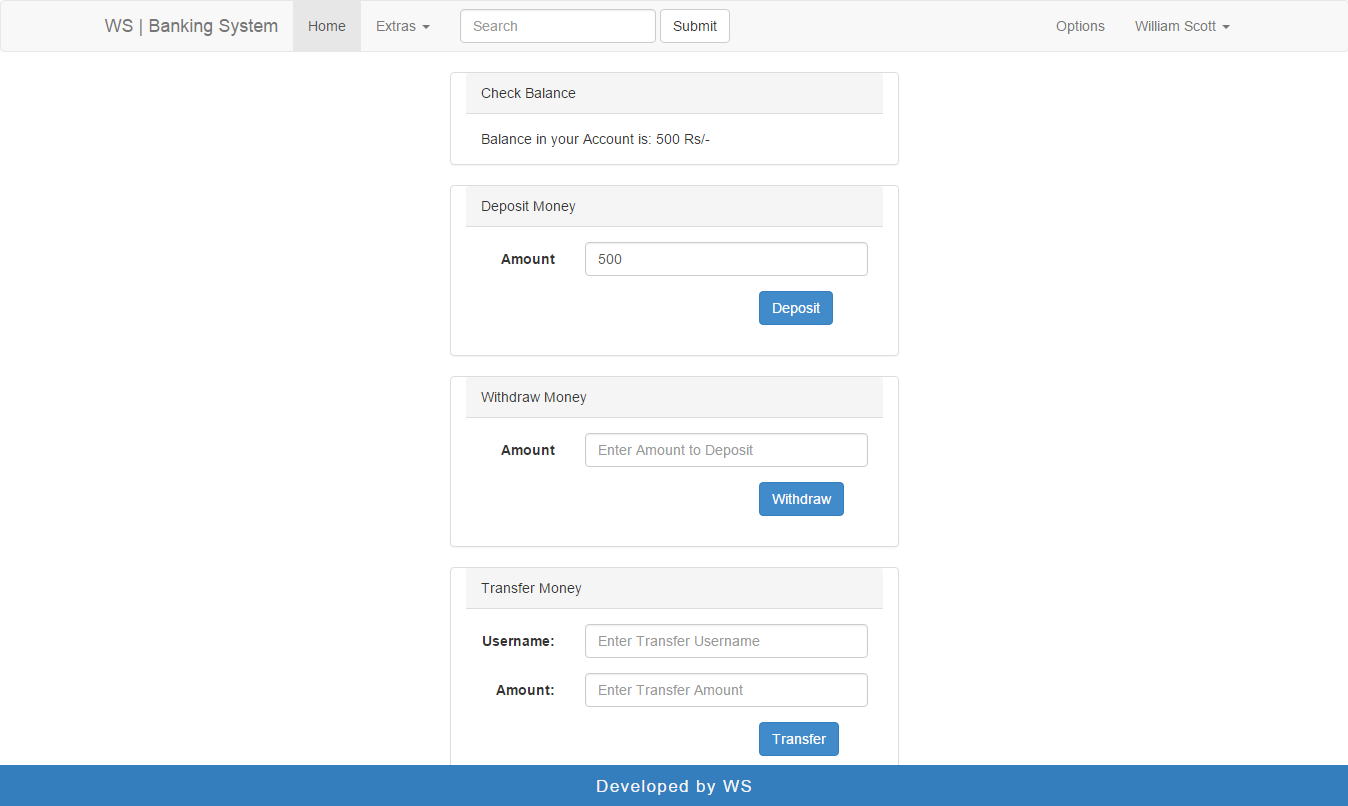












**Result**

Thus, the enterprise java persistence API for banking system was successfully developed and deployed

[Signature of the Staff In-charge]

[Name of the Staff In – charge]