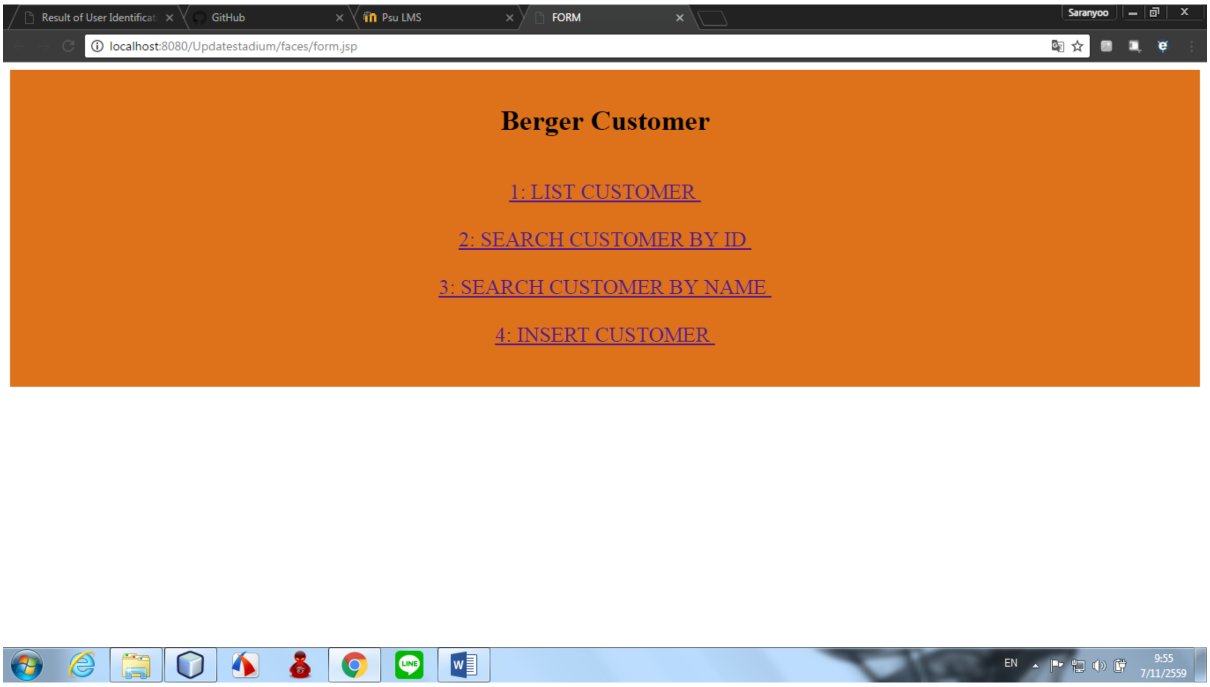
1. หน้าจอ

-แสดงรายการข้อมูลในตาราง


Result of User Identificat
GitHub
Psu LMS
FORM
Saranyoo
localhost:8080/Updatestadium/faces/form.jsp
EN
9:55
7/11/2559

Berger Customer

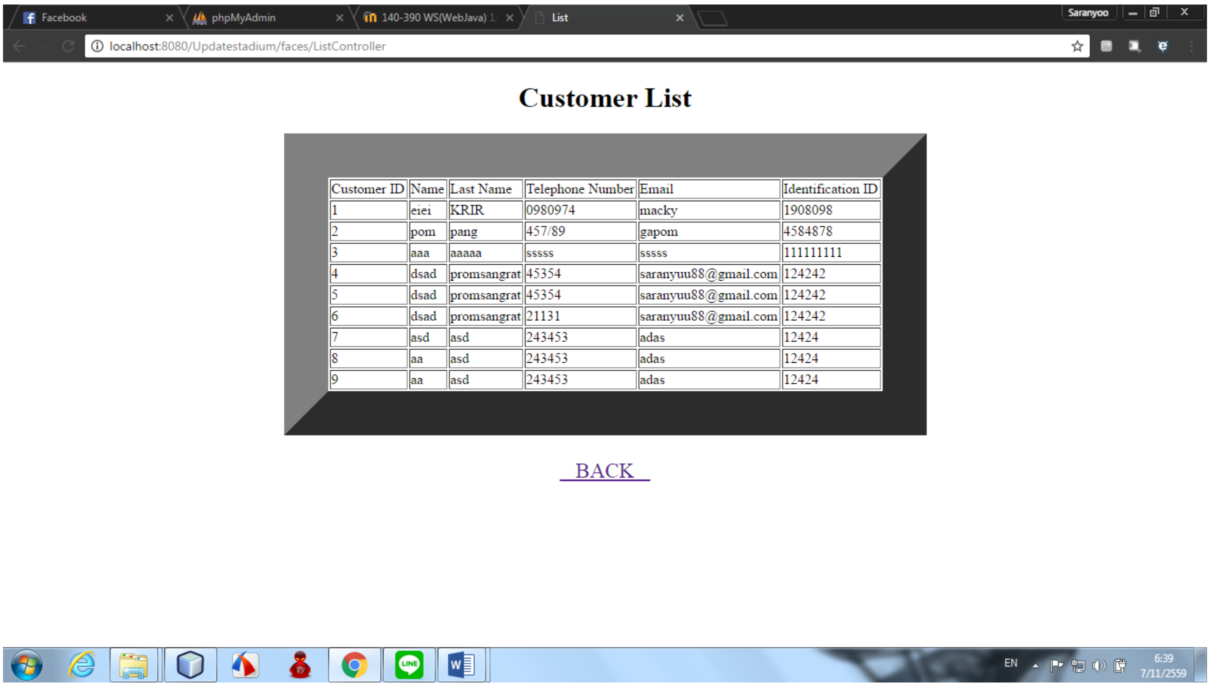
1: LIST CUSTOMER

2: SEARCH CUSTOMER BY ID

3: SEARCH CUSTOMER BY NAME

4: INSERT CUSTOMER

คลิกเข้าไปใน list customer



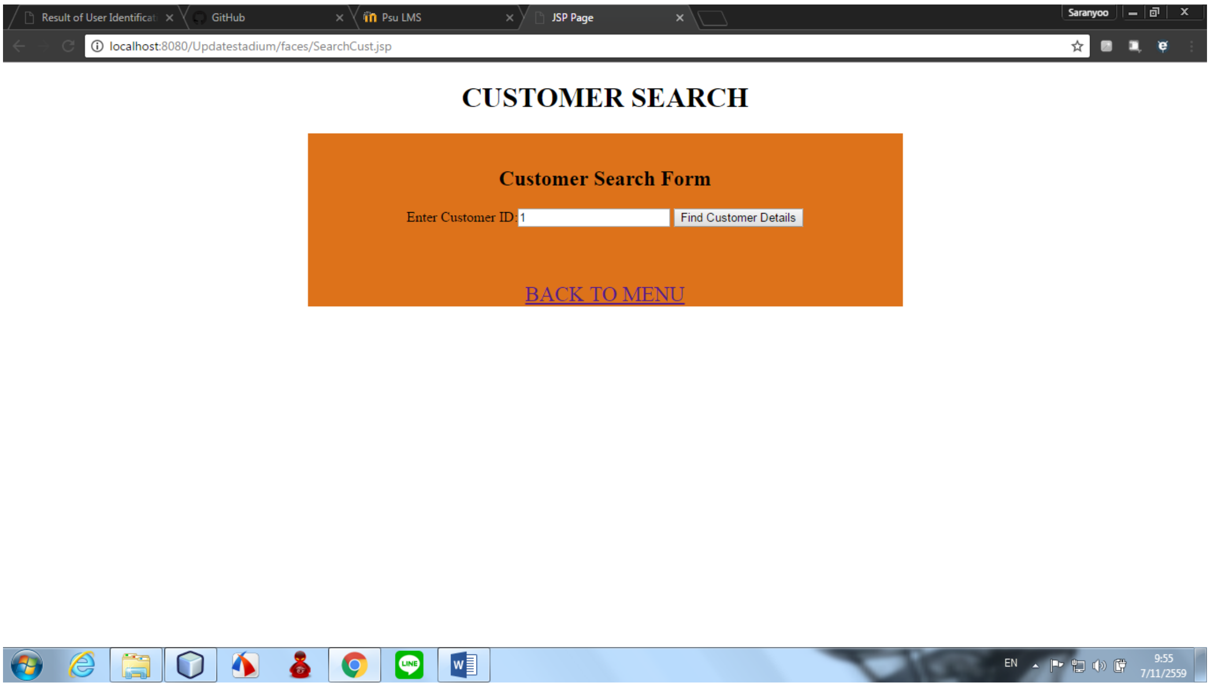
Customer List

Customer ID	Name	Last Name	Telephone Number	Email	Identification ID
1	eiei	KRIR	0980974	macky	1908098
2	pom	pang	457/89	gapom	4584878
3	aaa	aaaaa	sssss	sssss	111111111
4	dsad	promsangrat	45354	saranyuu88@gmail.com	124242
5	dsad	promsangrat	45354	saranyuu88@gmail.com	124242
6	dsad	promsangrat	21131	saranyuu88@gmail.com	124242
7	asd	asd	243453	adas	12424
8	aa	asd	243453	adas	12424
9	aa	asd	243453	adas	12424

BACK

-ฟอร์ม ค้นหา + ผลการค้นหา มี 2 แบบคือ

----- ค้นหาจาก id -> ได้ผลการค้นหารายการเดียว



localhost:8080/Updatestadium/faces/SearchCust.jsp

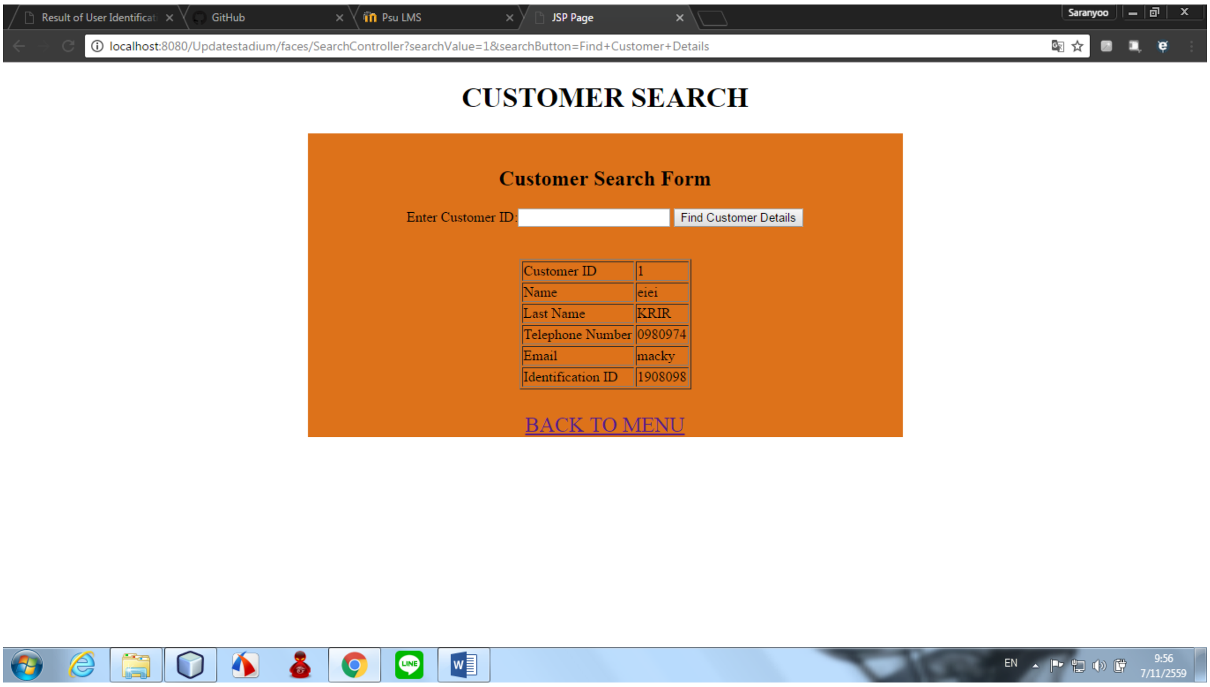
CUSTOMER SEARCH

Customer Search Form

Enter Customer ID: 1 Find Customer Details

BACK TO MENU

ผลการค้นหา



localhost:8080/Updatestadium/faces/SearchController?searchValue=1&searchButton=Find+Customer+Details

CUSTOMER SEARCH

Customer Search Form

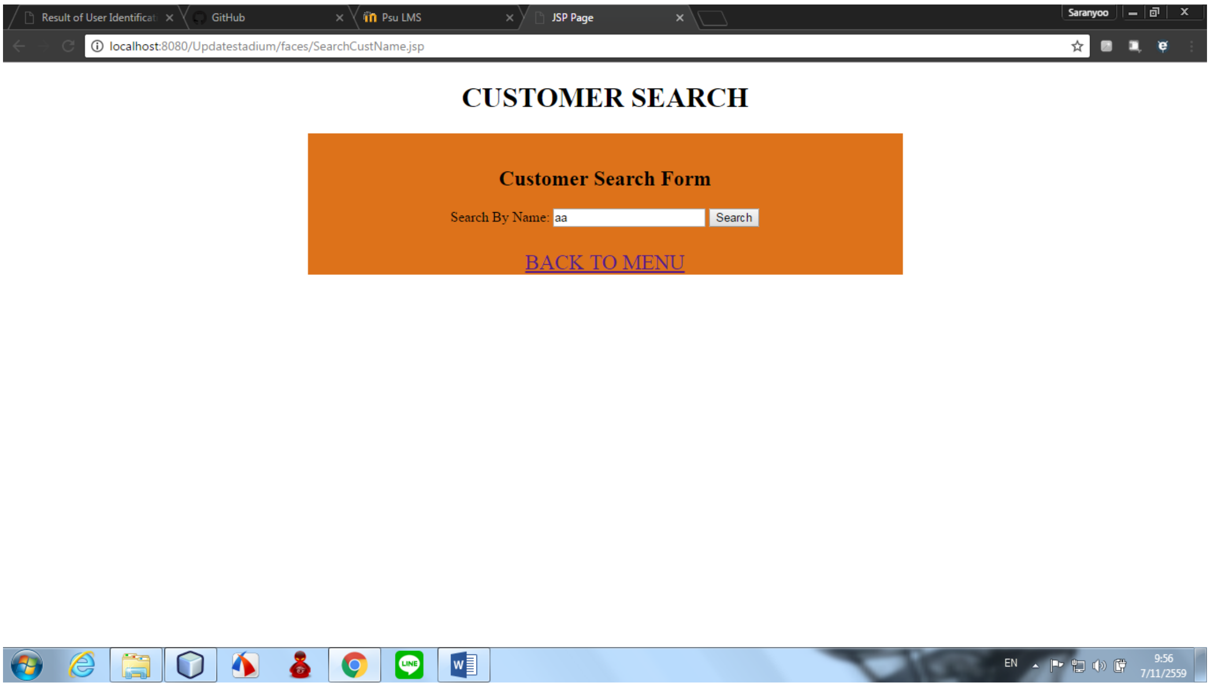
Enter Customer ID: Find Customer Details

Customer ID	1
Name	eiei
Last Name	KRIR
Telephone Number	0980974
Email	macky
Identification ID	1908098

BACK TO MENU

Mr.Saranyoo Promsangrat 5730213036 SE

----- ค้นหาจาก ข้อความ (ใช้ LIKE) -> ได้ผลการค้นหาเป็น List



localhost:8080/Updatestadium/faces/SearchCustName.jsp

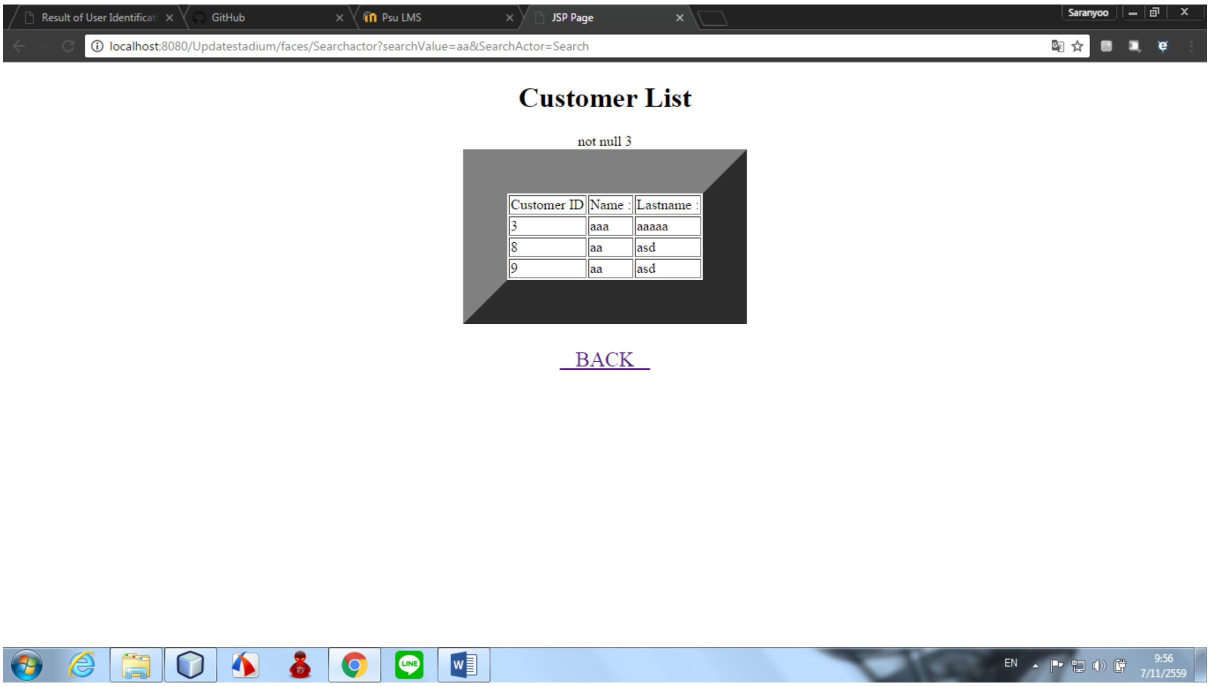
CUSTOMER SEARCH

Customer Search Form

Search By Name: aa Search

BACK TO MENU

ผลการค้นหา



localhost:8080/Updatestadium/faces/Searchactor?searchValue=aa&SearchActor=Search

Customer List

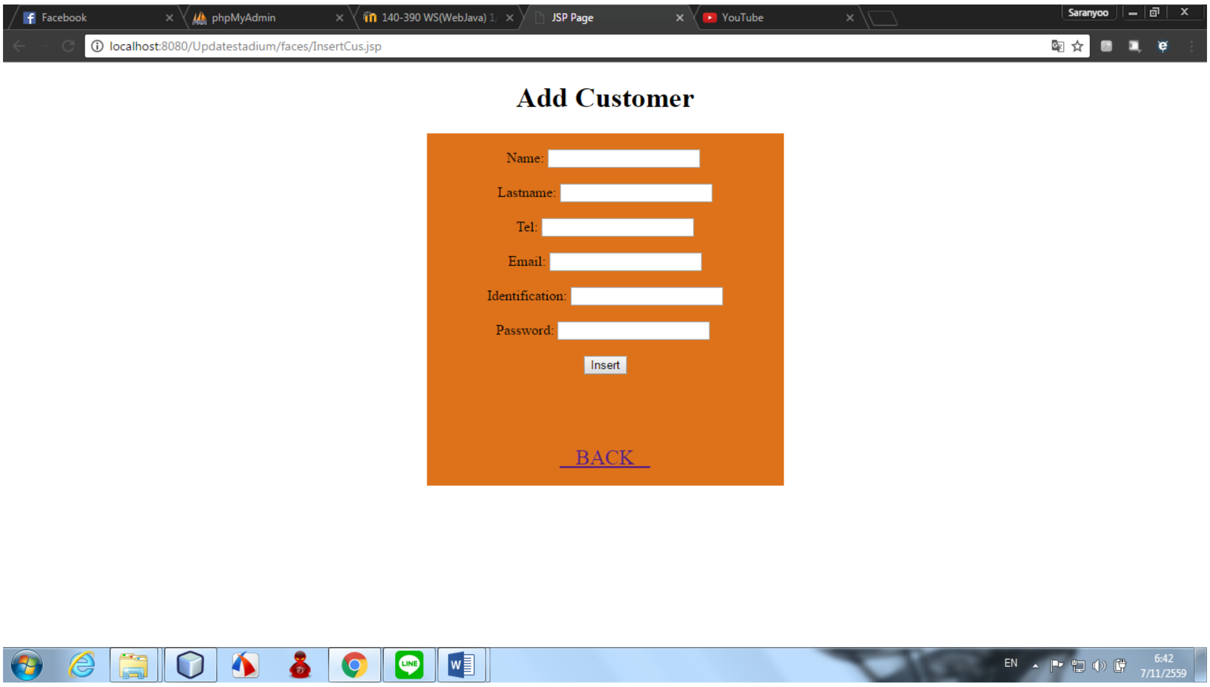
not null 3

Customer ID	Name :	Lastname :
3	aaa	aaaaa
8	aa	asd
9	aa	asd

BACK

Mr.Saranyoo Promsangrat 5730213036 SE

-ฟอร์มเพิ่มข้อมูล + ผลการเพิ่มข้อมูล



Add Customer

Name:

Lastname:

Tel:

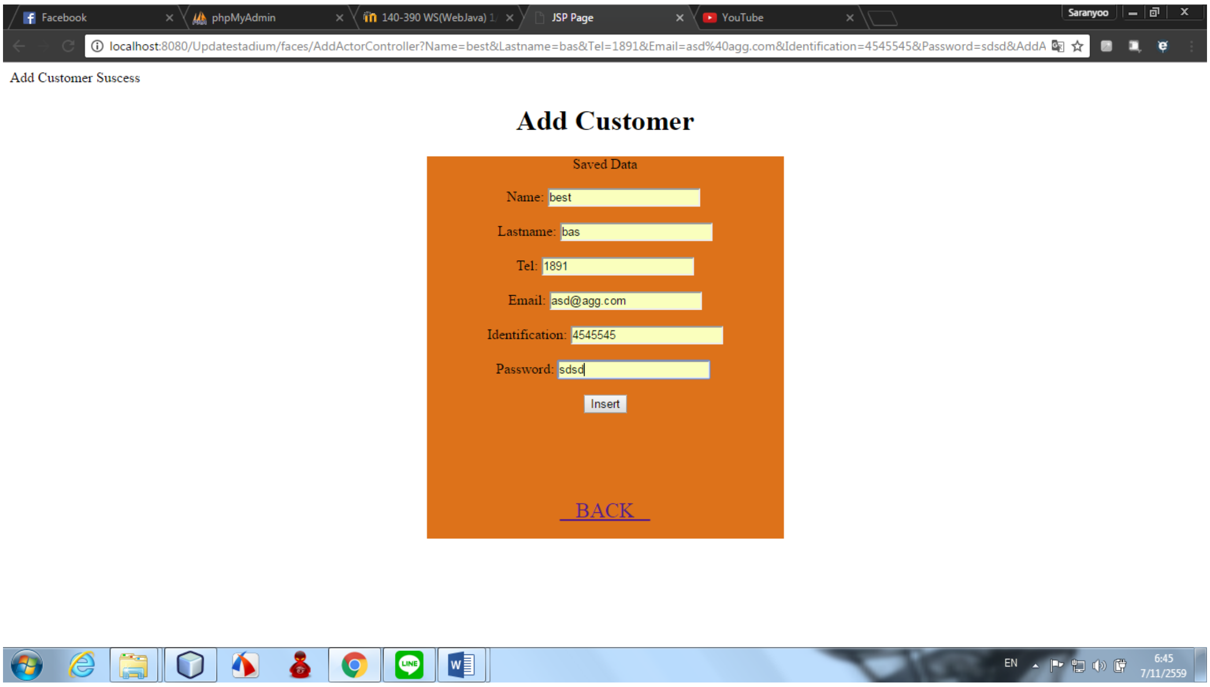
Email:

Identification:

Password:

Insert

BACK


Facebook
phpMyAdmin
140-390 WS(WebJava)
JSP Page
YouTube
Saranyoo
localhost:8080/Updatestadium/faces/AddActorController?Name=best&Lastname=bas&Tel=1891&Email=asd%40agg.com&Identification=4545545&Password=sdsd&AddA
EN
6:45
7/11/2559

Add Customer Sucess

Add Customer

Saved Data

Name: best

Lastname: bas

Tel: 1891

Email: asd@agg.com

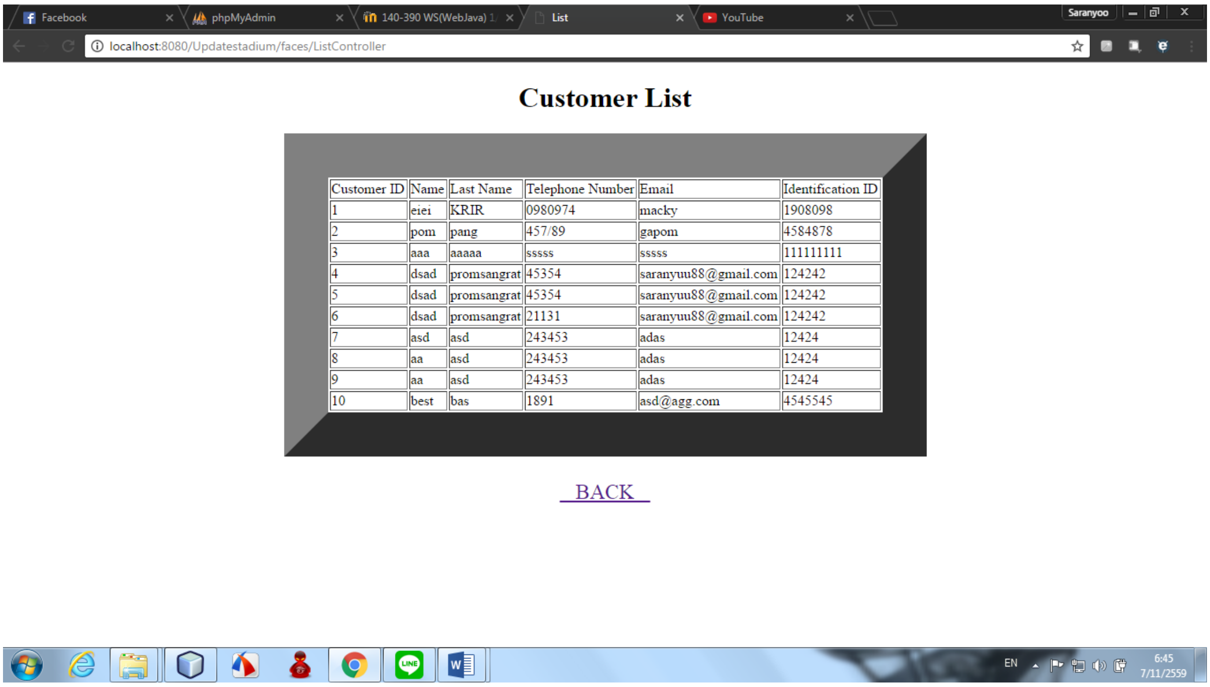
Identification: 4545545

Password: sdsd

Insert

BACK

ผลการเพิ่มข้อมูล

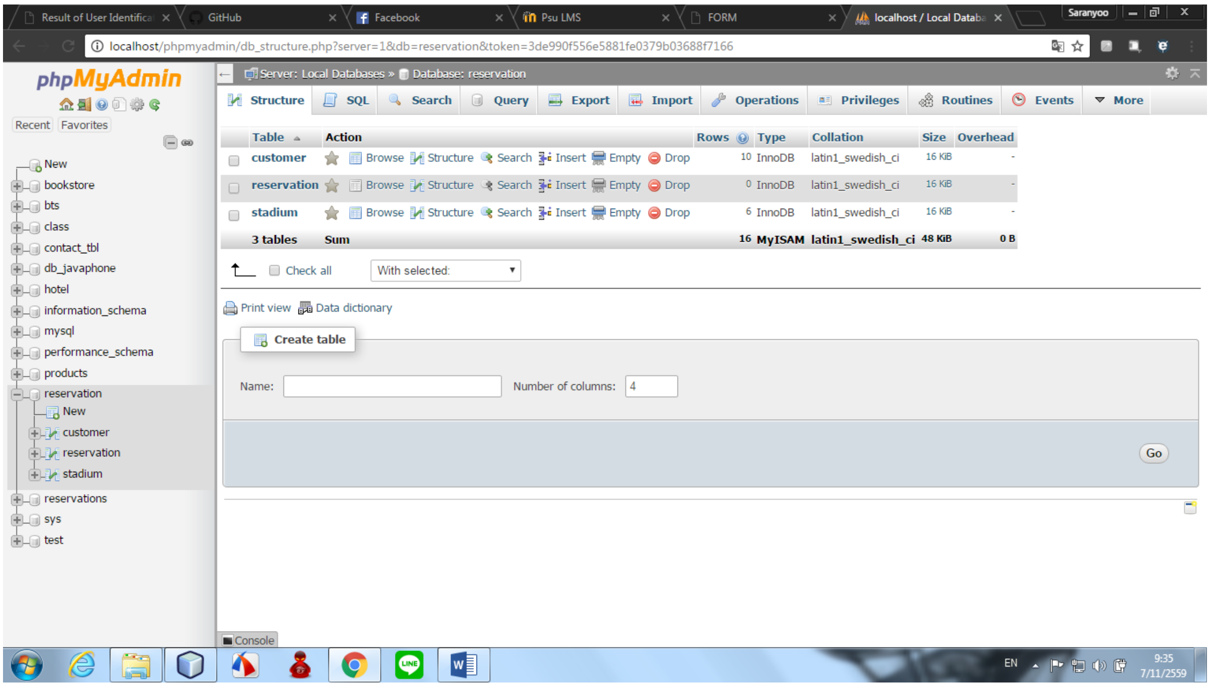


Customer List

Customer ID	Name	Last Name	Telephone Number	Email	Identification ID
1	eiei	KRIR	0980974	macky	1908098
2	pom	pang	457/89	gapom	4584878
3	aaa	aaaaa	sssss	sssss	111111111
4	dsad	promsangrat	45354	saranyuu88@gmail.com	124242
5	dsad	promsangrat	45354	saranyuu88@gmail.com	124242
6	dsad	promsangrat	21131	saranyuu88@gmail.com	124242
7	asd	asd	243453	adas	12424
8	aa	asd	243453	adas	12424
9	aa	asd	243453	adas	12424
10	best	bas	1891	asd@agg.com	4545545

BACK

2. โครงสร้างตาราง + ตัวอย่างข้อมูลในตาราง



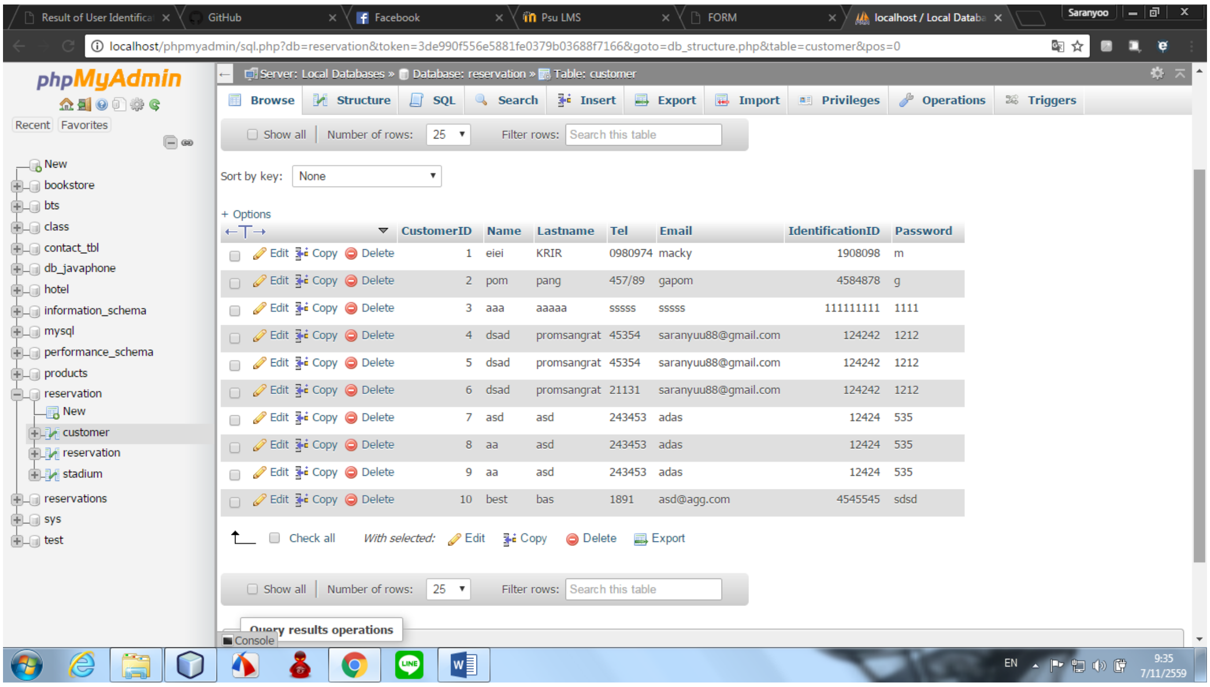
localhost/phpmyadmin/db_structure.php?server=1&db=reservation&token=3de990f556e5881fe0379b03688f7166

Server: Local Databases » Database: reservation

Table	Action	Rows	Type	Collation	Size	Overhead
customer	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 KiB	-
reservation	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
stadium	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 KiB	-
3 tables	Sum	16	MyISAM	latin1_swedish_ci	48 KiB	0 B

Create table

Name: Number of columns: 4



localhost/phpmyadmin/sql.php?db=reservation&token=3de990f556e5881fe0379b03688f7166&goto=db_structure.php&table=customer&pos=0

Server: Local Databases » Database: reservation » Table: customer

Number of rows: 25

Sort by key: None

	CustomerID	Name	Lastname	Tel	Email	IdentificationID	Password
Edit Copy Delete	1	eiei	KRIR	0980974	macky	1908098	m
Edit Copy Delete	2	pom	pang	457/89	gapom	4584878	g
Edit Copy Delete	3	aaa	aaaaa	sssss	sssss	111111111	1111
Edit Copy Delete	4	dsad	promsangrat	45354	saranyuu88@gmail.com	124242	1212
Edit Copy Delete	5	dsad	promsangrat	45354	saranyuu88@gmail.com	124242	1212
Edit Copy Delete	6	dsad	promsangrat	21131	saranyuu88@gmail.com	124242	1212
Edit Copy Delete	7	asd	asd	243453	adas	12424	535
Edit Copy Delete	8	aa	asd	243453	adas	12424	535
Edit Copy Delete	9	aa	asd	243453	adas	12424	535
Edit Copy Delete	10	best	bas	1891	asd@agg.com	4545545	sdsd

3. ไฟล์ที่เป็น model ของตาราง (ชื่อตาราง.java)

Customer.java

```
package stadium;

public class Customer  implements java.io.Serializable {



     private Integer customerId;
     private String name;
     private String lastname;
     private String tel;
     private String email;
     private int identificationId;
     private String password;

    public Customer() {
    }

    public Customer(String name, String lastname, String tel, String
email, int identificationId, String password) {
       this.name = name;
       this.lastname = lastname;
       this.tel = tel;
       this.email = email;
       this.identificationId = identificationId;
       this.password = password;
    }

    Customer(String customerId, String firstname, String lastname) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
```

```
    }

    Customer(int customeridd, String firstname, String lastname) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

    Customer(int customeridd, String Name, String Lastname, String
Tel, String Email, String Identification) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

    Customer(int customeridd, String Name, String Lastname, String
Tel, String Email, String Identification, String Password) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

   Customer(int customeridd, String Name, String Lastname, String
Tel, String Email, int IdentificationID, String Password) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

   Customer(String Name, String Lastname, String Tel, String Email,
int IdentificationID) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

    public Integer getCustomerId() {
        return this.customerId;
    }

    public void setCustomerId(Integer customerId) {
        this.customerId = customerId;
    }
    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }
    public String getLastname() {
        return this.lastname;
    }

    public void setLastname(String lastname) {
        this.lastname = lastname;
    }
    public String getTel() {
        return this.tel;
    }

    public void setTel(String tel) {
        this.tel = tel;
    }
    public String getEmail() {
        return this.email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
   public int getIdentificationId() {
        return this.identificationId;
    }

    public void setIdentificationId(int identificationId) {
        this.identificationId = identificationId;
    }
    public String getPassword() {
        return this.password;
    }

    public void setPassword(String password) {
        this.password = password;
    }




}
```

4. code ไฟล์ Hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/reservation?zeroDateTimeBehavior=convertToNull</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.show_sql">true</property>
    <property name="hibernate.current_session_context_class">thread</property>
    <property name="hibernate.query.factory_class">org.hibernate.hql.internal.classic.ClassicQueryTranslatorFactory</property>
    <mapping resource="stadium/Reservation.hbm.xml"/>
    <mapping resource="stadium/Stadium.hbm.xml"/>
    <mapping resource="stadium/Customer.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

5. code ไฟล์ Servlet ทั้งหมด

ActorService.java = สร้างไว้สำหรับเสิชข้อมูล ที่ได้รับค่ามาจาก หน้า SearchCustName.jsp

```
package stadium;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;


public class ActorService extends HttpServlet {

    public String addActor(Customer customer) {
        String message = null;
        Session session = null;
      try {
          session= HibernateUtill.getSessionFactory().openSession();
          Transaction tx = session.getTransaction();
          tx.begin();
          session.save(customer);
          tx.commit();
          message = "Saved Data";
        return message;
        } catch (Exception e) {
            return (e.getMessage());
        } finally {
            session.close();
        }
    }

    public List searchActorByName(String searchKeyword) {
        List EmployeeList = null;
        Session session = null;
        try {
            session = HibernateUtill.getSessionFactory().openSession();
            org.hibernate.Transaction tx = session.beginTransaction();
            Query query = session.createQuery("from Customer  where
Name like :keyword ");
            query.setParameter("keyword", "%" + searchKeyword + "%");
            EmployeeList = (List<Customer>) query.list();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return EmployeeList;
    }

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
         /* TODO output your page here. You may use following sample
code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ActorService</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet ActorService at " +
request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
      return "Short description";
    }// </editor-fold>

}
```

AddActorController.java = สร้างไว้การ insert ข้อมูลที่ได้รับมากจากหน้า insertCus.jsp เมื่อ insertข้อมูลสำเร็จ ระบบจะแสดงคำว่า add customer suscess

```
package stadium;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
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;
import org.hibernate.validator.constraints.Email;


public class AddActorController extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("Add Customer Suscess");
            HttpSession session = request.getSession(true);

        //String customerId = request.getParameter("customerId");
        //int customeridd = Integer.parseInt(customerId);
            String Name = request.getParameter("Name");
            String Lastname = request.getParameter("Lastname");
            String Tel = request.getParameter("Tel");
            String Email = request.getParameter("Email");
            String Identification = request.getParameter("Identification");
            int IdentificationID = Integer.parseInt(Identification );
           String Password = request.getParameter("Password");

           Customer ac = new Customer(Name,
Lastname,Tel,Email,IdentificationID,Password);
        //ac.setFirstName(firstname);
        //ac.setLastName(lastname);
        ActorService acs = new ActorService();
       String resultMessage = acs.addActor(ac);

        RequestDispatcher rd =
request.getRequestDispatcher("InsertCus.jsp?message=" +
resultMessage );
        request.setAttribute("searchResult", null);
        rd.forward(request, response);
      }
   }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
  /**
   * Handles the HTTP <code>GET</code> method.
   *
   * @param request servlet request
   * @param response servlet response
   * @throws ServletException if a servlet-specific error occurs
   * @throws IOException if an I/O error occurs
   */
   @Override
   protected void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
      processRequest(request, response);
   }

  /**
   * Handles the HTTP <code>POST</code> method.
   *
   * @param request servlet request
```

```
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

CustService.java = สร้างไว้สำหรับ เสิชข้อมูล ที่ได้รับค่ามาจากหน้า searchCust.jsp

```
package stadium;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;

public class CustService extends HttpServlet {

    public List getCustomer() {
        List CustomerList = null;
        Session session = null;
        try {
            session = HibernateUtill.getSessionFactory().openSession();
            org.hibernate.Transaction tx = session.beginTransaction();
            Query q = session.createQuery("from Customer");
            CustomerList = (List<Customer>) q.list();
        } catch (Exception e) {
            e.printStackTrace();
       }
        return CustomerList;
    }

    public Customer searchCustomer(short search_customerId)
```

```
   {
        String message=null;
      Session session=null;
     Transaction tx = null;
                         Customer customer = null;
                         try {
              session=HibernateUtill.getSessionFactory().openSession();
                                tx = session.getTransaction();
                                tx.begin();
                                 Query query = session.createQuery("from
Customer where customerid="+search_customerId);
                                 customer = (Customer)query.uniqueResult();
                                tx.commit();
                         } catch (Exception e) {
                                 if (tx != null) {
                                        tx.rollback();
                                 }
                                e.printStackTrace();
                         } finally {
                                session.close();
                         }
                         return customer;
   }

   protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
          throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
     try (PrintWriter out = response.getWriter()) {
       /* TODO output your page here. You may use following sample
code. */
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet CustomerServlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Servlet CustomerServlet at " +
request.getContextPath() + "</h1>");
out.println("</body>");
out.println("</html>");
        }
    }

   // <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
   /**
    * Handles the HTTP <code>GET</code> method.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
processRequest(request, response);
    }


   /**
    * Handles the HTTP <code>POST</code> method.
    *
```

```
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
processRequest(request, response);
    }

   /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Customer.java = เป็นไฟล์ model ของตาราง

```
package stadium;

public class Customer  implements java.io.Serializable {


    private Integer customerId;
    private String name;
    private String lastname;
    private String tel;
    private String email;
    private int identificationId;
   private String password;

   public Customer() {
   }

   public Customer(String name, String lastname, String tel, String email, int identificationId, String password) {
      this.name = name;
      this.lastname = lastname;
      this.tel = tel;
      this.email = email;
      this.identificationId = identificationId;
      this.password = password;
   }

   Customer(String customerId, String firstname, String lastname) {
       throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
   }
```

```
    Customer(int customeridd, String firstname, String lastname) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

    Customer(int customeridd, String Name, String Lastname, String
Tel, String Email, String Identification) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

    Customer(int customeridd, String Name, String Lastname, String
Tel, String Email, String Identification, String Password) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

    Customer(int customeridd, String Name, String Lastname, String
Tel, String Email, int IdentificationID, String Password) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }

    Customer(String Name, String Lastname, String Tel, String Email,
int IdentificationID) {
        throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
}

public Integer getCustomerId() {
    return this.customerId;
}

public void setCustomerId(Integer customerId) {
    this.customerId = customerId;
}
public String getName() {
    return this.name;
}

public void setName(String name) {
    this.name = name;
}
public String getLastname() {
    return this.lastname;
}

public void setLastname(String lastname) {
    this.lastname = lastname;
}
public String getTel() {
    return this.tel;
}

public void setTel(String tel) {
    this.tel = tel;
}
public String getEmail() {
    return this.email;
}
```

```
    public void setEmail(String email) {
        this.email = email;
    }
    public int getIdentificationId() {
        return this.identificationId;
    }

    public void setIdentificationId(int identificationId) {
        this.identificationId = identificationId;
    }
    public String getPassword() {
        return this.password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

CustomerService.java = เป็นการสร้างเพื่อให้ฟังก์ชั่น customer list ทำงาน ได้

```
package stadium;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.Transaction;

public class CustomerService extends HttpServlet {

  public List getName() {
    List CustomerList = null;
    Session session = null;
    try {
      session = HibernateUtill.getSessionFactory().openSession();
      org.hibernate.Transaction tx = session.beginTransaction();
      Query q = session.createQuery("from Customer ");
      CustomerList = (List<Customer>) q.list();
    } catch (Exception e) {
      e.printStackTrace();
    }
    return CustomerList;
  }
```

```
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample
code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
         out.println("<head>");
            out.println("<title>Servlet FilmServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet FilmServlet at " +
request.getContextPath() + "</h1>");
           out.println("</body>");
            out.println("</html>");
        }
    }

   // <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
   /**
    * Handles the HTTP <code>GET</code> method.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

   /**
    * Handles the HTTP <code>POST</code> method.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

   /**
    * Returns a short description of the servlet.
    *
    * @return a String containing servlet description
    */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

    Customer SearchCust(short search_CustomerId) {
        throw new UnsupportedOperationException("Not supported 
yet."); //To change body of generated methods, choose Tools | 
Templates.
    }

    Customer SearchCustomer(short search_CustomerId) {
        throw new UnsupportedOperationException("Not supported 
yet."); //To change body of generated methods, choose Tools | 
Templates.
    }



}
```

HibernateUtill.java

```

package stadium;

import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.SessionFactory;


public class HibernateUtill {

    private static final SessionFactory sessionFactory;

    static {
        try {

            sessionFactory = new 
AnnotationConfiguration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("Initial SessionFactory creation failed." +
ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

ListController.java = โชว์ list รายชื่อ customer ใน sql

```
package stadium;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
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 ListController extends HttpServlet {

protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

        // Create a session object if it is already not  created.
           HttpSession session = request.getSession(true);

           try {

               CustomerService acs = new CustomerService();
               List CustomerList = acs.getName();
           //   int num = filmList.size();
               session.setAttribute("listResult", CustomerList);


           } catch (Exception e) {
               e.printStackTrace();
           }
             RequestDispatcher rd =
request.getRequestDispatcher("CustomerList.jsp");
             rd.forward(request, response);

        }

    }


// <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@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);
   }

  /**
   * Returns a short description of the servlet.
   *
   * @return a String containing servlet description
   */
   @Override
      public String getServletInfo() {
      return "Short description";
   }// </editor-fold>

}
```

Reservation.java

```
package stadium;

public class Reservation  implements java.io.Serializable {



     private Integer reservId;
     private int customerId;
     private int price;

    public Reservation() {
    }

    public Reservation(int customerId, int price) {
       this.customerId = customerId;
       this.price = price;
    }

    public Integer getReservId() {
        return this.reservId;
    }

    public void setReservId(Integer reservId) {
       this.reservId = reservId;
    }
    public int getCustomerId() {
        return this.customerId;
    }

    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public int getPrice() {
        return this.price;
    }

    public void setPrice(int price) {
        this.price = price;
    }




}
```

SearchController.java = สร้างไว้สำหรับ เสิชข้อมูล โดยใช้ ID ที่ได้รับค่ามาจากหน้า searchCust.jsp

```
package stadium;

import java.io.IOException;
import java.io.PrintWriter;
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 SearchController extends HttpServlet {

protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
```

```
        // Create a session object if it is already not  created.
          HttpSession session = request.getSession(true);

         try {
             short search_customerId =
Short.parseShort(request.getParameter("searchValue"));

             CustService acs = new CustService();
          Customer customer_obj =
acs.searchCustomer(search_customerId);

          session.setAttribute("searchResult", customer_obj);

             RequestDispatcher rd =
request.getRequestDispatcher("SearchCust.jsp");
             rd.forward(request, response);

          } finally {
             out.close();
          }
      }
  }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. 
Click on the + sign on the left to edit the code.">
  /**
   * Handles the HTTP <code>GET</code> method.
   *
   * @param request servlet request
   * @param response servlet response
   * @throws ServletException if a servlet-specific error occurs
   * @throws IOException if an I/O error occurs
```

```
    */
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

   /**
    * Handles the HTTP <code>POST</code> method.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

   /**
    * Returns a short description of the servlet.
    *
    * @return a String containing servlet description
    */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
```

```
}
```

Searchactor.java = สร้างไว้เพื่อรับค่าจาก SearchCustName.jsp เพื่อไปแสดงค่า การค้นหาที่ได้รับ มาเป็น list ที่เราค้นหาในหน้า custList

```
package stadium;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
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 Searchactor extends HttpServlet {


    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
        // Create a session object if it is already not created.
            HttpSession session = request.getSession(true);

            try {
            String searchValue = request.getParameter("searchValue");
               ActorService acs = new ActorService();
```

```
                List CustomerList =
acs.searchActorByName(searchValue);
            int num = CustomerList.size();
              session.setAttribute("CustomerListResult", CustomerList);

            } catch (Exception e) {
                e.printStackTrace();
            }
            RequestDispatcher rd =
request.getRequestDispatcher("CustList.jsp");
            rd.forward(request, response);
        }
    }



    // <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
      @Override
      protected void doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
     }
    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Stadium.java

```
package stadium;
// Generated Oct 30, 2016 10:11:47 PM by Hibernate Tools 4.3.1
```

```
import java.util.Date;

/**
 * Stadium generated by hbm2java
 */
public class Stadium  implements java.io.Serializable {


     private Integer staduimId;
     private String type;
     private String status;
     private int pricePhour;
     private Date timeF;
     private Date timeT;

    public Stadium() {
    }

    public Stadium(String type, String status, int pricePhour, Date
timeF, Date timeT) {
       this.type = type;
       this.status = status;
       this.pricePhour = pricePhour;
       this.timeF = timeF;
       this.timeT = timeT;
    }

    public Integer getStaduimId() {
        return this.staduimId;
    }

    public void setStaduimId(Integer staduimId) {
        this.staduimId = staduimId;
    }
    public String getType() {
        return this.type;
    }

    public void setType(String type) {
        this.type = type;
    }
    public String getStatus() {
        return this.status;
    }

    public void setStatus(String status) {
        this.status = status;
    }
    public int getPricePhour() {
        return this.pricePhour;
    }

    public void setPricePhour(int pricePhour) {
        this.pricePhour = pricePhour;
    }
    public Date getTimeF() {
        return this.timeF;
    }

    public void setTimeF(Date timeF) {
        this.timeF = timeF;
    }
    public Date getTimeT() {
        return this.timeT;
    }
```

```
    public void setTimeT(Date timeT) {
        this.timeT = timeT;
    }


}
```

Customer.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate
Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated Oct 30, 2016 10:11:49 PM by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
    <class name="stadium.Customer" table="customer"
catalog="reservation" optimistic-lock="version">
        <id name="customerId" type="java.lang.Integer">
            <column name="CustomerID" />
            <generator class="identity" />
        </id>
        <property name="name" type="string">
            <column name="Name" length="30" not-null="true" />
        </property>
        <property name="lastname" type="string">
            <column name="Lastname" length="30" not-null="true" />
        </property>
        <property name="tel" type="string">
            <column name="Tel" length="11" not-null="true" />
        </property>
        <property name="email" type="string">
            <column name="Email" length="30" not-null="true" />
        </property>
        <property name="identificationId" type="int">
            <column name="IdentificationID" not-null="true" />
        </property>
        <property name="password" type="string">
            <column name="Password" length="16" not-null="true" />
        </property>
    </class>
</hibernate-mapping>
```

Reservation.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate
Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated Oct 30, 2016 10:11:49 PM by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
    <class name="stadium.Reservation" table="reservation"
catalog="reservation" optimistic-lock="version">
        <id name="reservId" type="java.lang.Integer">
            <column name="ReservID" />
            <generator class="identity" />
        </id>
        <property name="customerId" type="int">
            <column name="CustomerID" not-null="true" />
        </property>
        <property name="price" type="int">
            <column name="Price" not-null="true" />
        </property>
    </class>
</hibernate-mapping>
```

Stadium.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate
Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated Oct 30, 2016 10:11:49 PM by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
  <class name="stadium.Stadium" table="stadium"
catalog="reservation" optimistic-lock="version">
    <id name="staduimId" type="java.lang.Integer">
      <column name="StaduimID" />
      <generator class="identity" />
    </id>
   <property name="type" type="string">
      <column name="Type" length="50" not-null="true" />
    </property>
    <property name="status" type="string">
      <column name="Status" length="65535" not-null="true" />
    </property>
    <property name="pricePhour" type="int">
      <column name="PricePHour" not-null="true" />
    </property>
    <property name="timeF" type="time">
      <column name="timeF" length="8" not-null="true" />
    </property>
    <property name="timeT" type="time">
     <column name="timeT" length="8" not-null="true" />
    </property>
  </class>
</hibernate-mapping>
```

6. code ไฟล์ JSP ทั้งหมด

Form.jsp = เป็นการแสดงเมนู ต่างๆ มี4ฟังก์ชั่น คือ 1.แสดงตารางข้อมูล customer list

2.การเสิชข้อมูล โดย id

3. การเสิชข้อมูลโดย name

4.การเพิ่มข้อมูลเข้าไปในฐานข้อมูล

```
<%@page import="stadium.Customer"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>

   <center>
<body>
<h1>CUSTOMER SEARCH</h1>
<div style="background-color: #dd721a; width:50%;">
<br />
<h2>Customer Search Form</h2>
          <form name="formSearchActor"action="Searchactor" >

          Search By Name: <input type="text"name="searchValue" value="" />
          <input type="submit" value="Search" name="SearchActor" />
      </form>
```

```
<td><a href="form.jsp"><font size ="5" ><br>BACK TO
MENU</br></font></a></td>
</body>
</center>
</html>
```

CustList.jsp = หน้าแสดงผลที่การค้นหาของฟังก์ชั่น search customer by name

```
<%@page import="stadium.Customer"%>
<%@page import="java.util.List"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html;
           charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <center>
   <body>
      <h1>Customer List</h1>
      <%
         try {
            List<Customer> CustomerList = (List<Customer>)
session.getAttribute("CustomerListResult");
            if (CustomerList != null) {
               out.println("not null");
               out.println(CustomerList.size());
     %>
      <table border="50" bgcolor="">
         <tbody>
            <tr>
```

```
          <td>Customer ID</td>

          <td>Name :</td>
          <td>Lastname :</td>
        </tr>
        <% for (Customer acc : CustomerList) {
      %>
        <tr>

            <td><% out.println(acc.getCustomerId()); %></td>

            <td><% out.println(acc.getName()); %></td>

            <td><% out.println(acc.getLastname()); %></td>
          </tr>
        <% } %>
      </tbody>
    </table>
    <%
        } else {
          out.println("return null");
       //out.println(EmployeeList);
        }
      } catch (Exception e) {
        e.printStackTrace();
        out.println("error");
      }
   %>


    <td><a href="SearchCustName.jsp"><font size ="5"
><br>&nbsp;&nbsp; BACK &nbsp; </br></font></a></td>
  </body>
```

```
   </center>
</html>
```

CustomerList.jsp

```
<%@page import="stadium.Customer"%>
<%@page import="java.util.List"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
      <title>List</title>
   </head>

   <center>
      <body>
      <form>
      <h1>Customer List</h1>

      <%
         try {

            List<Customer> CustomerList = (List<Customer>)
session.getAttribute("listResult");

            if (CustomerList != null) {



    %>
      <table border="50" bgcolor="">

      <tbody>
        <tr>
          <td>Customer ID</td>
            <td>Name</td>
             <td>Last Name</td>
             <td>Telephone Number</td>
              <td>Email</td>
              <td> Identification ID</td>

        </tr>
        <%                  for (Customer acc : CustomerList) {
      %>
        <tr>
           <td><% out.println(acc.getCustomerId()); %></td>
           <td><% out.println(acc.getName()); %></td>
           <td><% out.println(acc.getLastname()); %></td>
           <td><% out.println(acc.getTel()); %></td>
            <td><% out.println(acc.getEmail()); %></td>
            <td><% out.println(acc.getIdentificationId()); %></td>
        </tr>
        <% } %>
      </tbody>
    </table>


    <%


        } else {
           out.println("return null");
        }
      } catch (Exception e) {
        e.printStackTrace();
```

```
            out.println("error");
         }

     %>
       <td><a href="form.jsp"><font size ="5" ><br>BACK TO
MENU</br></font></a></td>
   </body>
   </form>
   </center>
</html>
```

InsertCus.jsp = หน้าการเพิ่มข้อมูลใน ฐานข้อมูล

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html;
         charset=UTF-8">
      <title>JSP Page</title>
   </head>
      <body background = "">
   <center>
   <body>
      <h1>Add Customer</h1>
      <div style="background-color: #dd721a; width:30%;">

      <%
         String message = request.getParameter("message");
         if (message != null) {
            out.println(message);
         }
```

```
    %>

     <form name="formAddActor" action="AddActorController">

        <br>Name: <input type="text" name="Name"value="" />
<br><br>
        Lastname: <input type="text" name="Lastname"value=""
/><br><br>
        Tel: <input type="text" name="Tel"value="" /><br><br>
        Email: <input type="text" name="Email"value="" /><br><br>
        Identification: <input type="text"
name="Identification"value="" /><br><br>
        Password: <input type="text" name="Password"value="" />
<br><br>
        <input type="submit" value="Insert"name="AddActor"
/><br><br>
     </form>

     <br><br>



   <td><a href="form.jsp"><font size ="5" ><br>BACK TO
MENU</br></font></a></td>
   <br>
     </body>
   </center>
</html>
```

SearchCust.jsp = หน้าการเสิชข้อมูล โดยใช้ ID แล้วจะโชว์ผลการค้นหา

```
<%@page import="stadium.Customer"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
<title>JSP Page</title>
</head>

   <center>
<body>
<h1>CUSTOMER SEARCH</h1>
<div style="background-color: #dd721a; width:50%;">
<br />
<h2>Customer Search Form</h2>
         <form name="searchForm" action="SearchController">
            Enter Customer ID:<input type="text" name="searchValue"
value="" >
            <input type="submit" value="Find Customer Details"
name="searchButton" /><br><br><br>
         </form>


<%
            try {
           // Create a session object if it is already not  created.
               Customer acc = (Customer)
session.getAttribute("searchResult");

               if (acc != null) {
%>
```

```
<table border="1">

<tbody>
          <tr>
         <td>Customer ID</td>
            <td><% out.println(acc.getCustomerId()); %></td>
          </tr>
          <tr>
            <td>Name</td>
            <td><% out.println(acc.getName()); %></td>
          </tr>
          <tr>
            <td>Last Name</td>
            <td><% out.println(acc.getLastname()); %></td>
          </tr>
          <tr>
            <td>Telephone Number</td>
            <td><% out.println(acc.getTel()); %></td>
          </tr>
          <tr>
            <td>Email</td>
            <td><% out.println(acc.getEmail()); %></td>
          </tr>
          <tr>
            <td> Identification ID</td>
            <td><% out.println(acc.getIdentificationId()); %></td>
          </tr>
        </tbody>
      </table>
```

```
<%
            }

             } catch (Exception e) {
out.println(e.getMessage());
             }

       %>

<td><a href="form.jsp"><font size ="5" ><br>BACK TO
MENU</br></font></a></td>
</body>
</center>
</html>
```

SearchCustName.jsp = หน้าการเสิชข้อมูล โดยใช้ NAME แล้วจะโชว์ผลการค้นหา

```
<%@page import="stadium.Customer"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
<title>JSP Page</title>
</head>

   <center>
<body>
<h1>CUSTOMER SEARCH</h1>
<div style="background-color: #dd721a; width:50%;">
<br />
```

```
<h2>Customer Search Form</h2>
           <form name="formSearchActor"action="Searchactor" >

          Search By Name: <input type="text"name="searchValue"
value="" />
          <input type="submit" value="Search" name="SearchActor" />
       </form>


<td><a href="form.jsp"><font size ="5" ><br>BACK TO
MENU</br></font></a></td>
</body>
</center>
</html>
```