Session-14

Validation

Contents

- Plug-in Mechanism
- Validation Framework

Plug-in Mechanism

What is a Plug-in?

- Any Java class that you want to initialize when Struts application starts up and destroy when the application shuts down.
  - Any Java class that implements org.apache.struts.action.Plugin interface

```
public interface PlugIn {
    public void init(ActionServlet servlet, ApplicationConfig config)
                                                   throws ServletException;
    public void destroy();
}
```

Why Plug-in?

- Before Struts 1.1 (in Struts 1.0), you had to subclass ActionServlet to initialize application resources at startup time.
- With plugin mechanism (in Struts 1.1), you create Plugin classes and configure them.
- Generic mechanism
  - Struts framework itself uses plugin mechanism for supporting Validator and Tiles

How do you configure Plug-in's?

- Must be declared in struts-config.xml via **<plug-in>** element : e.g.,

```
<plug-in className="org.apache.struts.plugins.ModuleConfigVerifier"/>


<plug-in className=
    "org.apache.struts.webapp.example.memory.MemoryDatabasePlugIn">
   <set-property property="pathname" value="/WEB-INF/database.xml"/>
</plug-in>


<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
  <set-property property="pathnames“ value="/WEB-INF/validator-rules.xml,
   /WEB-INF/validation.xml"/>
</plug-in>
```

How do Plug-in's get called?

- During startup of a Struts application, ActionServlet calls init() method of each Plug-in configured.
- Plug-ins are called in the order they are configured in struts-config.xml file.

Validation Framework

Struts Validation Framework

- Originated from “generic” Validator framework from Jakarta

- Struts 1.1 includes this by default
  - Allows **declarative validation** for many fields
  - **Formats**
    - Dates, Numbers, Email, Credit Card, Postal Codes
  - **Lengths**
    - Minimum, maximum
  - **Ranges**

Validation Rules

- Rules are tied to specific fields in a form.
- Basic Validation Rules are built-in in the Struts 1.1 core distribution
  - e.g. “required”, “minLength”, “maxLength”, etc.
- The built-in validation rules come with Javascript that allows you to do client side validation.
- Custom Validation rules can be created and added to the definition file.
  - Can also define regular expressions

Things to do in order to use **Validator framework**

- Configure Validator Plug-in
- Configure **validation.xml** file (and **validationrules.xml** file)
- Extend Validator ActionForms or Dynamic ActionForms
- Set **validate="true"** on Action Mappings
- Include the <html:javascript> tag for client side validation in the form's JSP page

How to Configure Validator Plugin

- Validator Plug-in should be configured in the Struts configuration file.
  - Just add the following <plug-in> element

```
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
        <set-property    property="pathnames"
                                 value="/WEB-INF/validator-rules.xml,
                                                         /WEB-INF/validation.xml"/>
</plug-in>
```

Two Validator Configuration Files

- **validation-rules.xml**
  - Contains global set of validation rules.
  - Provided by Struts framework.

- **validation.xml**
  - Application specific.
  - Provided by application developer.
  - It specifies which validation rules from validationrules.xml file are used by a particular ActionForm.
  - No need to write validate() code in ActionForm class.

validation-rules.xml

Built-in (basic) Validation Rules

- required, requiredif
- minlength
- maxlength
- mask
- byte, short, integer, long, float, double
- date, range, intRange, floatRange
- creditCard, email

validation-rules.xml: Built-in “required” validation rule

```
<form-validation>
    <global>
        <validator name="required"
            classname="org.apache.struts.validator.FieldChecks"
            method="validateRequired”
    methodParams="java.lang.Object,org.apache.commons.validator.ValidatorAction, org.apache.commons.validator.Field, org.apache.struts.action.ActionErrors, javax.servlet.http.HttpServletRequest"
    msg="errors.required">
    <javascript><![CDATA[
        function validateRequired(form) {
        var isValid = true;
            ...
    </javascript>
</validator>
```

Built-in Error Messages From Validator Framework

```
# Built-in error messages for validator framework checks
# You have to add the following to your application's resource bundle
        errors.required={0} is required.
        errors.minlength={0} cannot be less than {1} characters.
        errors.maxlength={0} cannot be greater than {2} characters.
        errors.invalid={0} is invalid.
        errors.byte={0} must be an byte.
        errors.short={0} must be an short.
        errors.integer={0} must be an integer.
        errors.long={0} must be an long.
        errors.float={0} must be an float.
        errors.double={0} must be an double.
        errors.date={0} is not a date.
        errors.range={0} is not in the range {1} through {2}.
        errors.creditcard={0} is not a valid credit card number.
        errors.email={0} is an invalid e-mail address.
```

validation.xml

- **E.g., LogonForm**

```
<form-validation>
<!-- ========== Default Language Form Definitions ================ -->
    <formset>
          <form name="logonForm">
                <field property="username"
                            depends="required, minlength,maxlength">
                    <arg key="prompt.username"/>
                    <arg key="${var:minlength}"    name="minlength"
                                     resource="false"/>
                    <arg key="${var:maxlength}"   name="maxlength"
                                     resource="false"/>
```

validation.xml : LogonForm

```
    <var>
        <var-name>maxlength</var-name>
        <var-value>16</var-value>
    </var>
    <var>
        <var-name>minlength</var-name>
        <var-value>3</var-value>
    </var>
</field>
```

validation.xml : LogonForm

```
<field      property="password"
            depends="required, minlength,maxlength“>
     <arg   key="prompt.password"/>
    <arg   key="${var:minlength}" name="minlength“       resource="false"/>
    <arg   key="${var:maxlength}" name="maxlength“       resource="false"/>
    <var>
            <var-name>maxlength</var-name>
            <var-value>16</var-value>
    </var>
    <var>
            <var-name>minlength</var-name>
            <var-value>3</var-value>
    </var>
</field>
</form>
```

struts-config.xml: logonForm

```
<form-beans>
    <!-- Logon form bean -->
    <form-bean     name="logonForm"
                   type="org.apache.struts.validator.DynaValidatorForm">
          <form-property name="username" type="java.lang.String"/>
          <form-property name="password" type="java.lang.String"/>
    </form-bean>

    <!-- Subscription form bean -->
    <form-bean name="subscriptionForm"
          type="org.apache.struts.webapp.example.SubscriptionForm"/>
</form-beans>
```

<field ...> Element

- Corresponds to a property in an **ActionForm**
  - <!ELEMENT field (msg?, arg0?, arg1?, arg2?, arg3?, var*)>

- Attributes
  - **property**: name of a property (in an ActionForm) that is to be validated
  - **depends**: comma-delimited list of validation rules to apply against this field
    - All validation rules have to succeed

<field ...> Element: **<msg ...> child element**

- Allows you to specify an alternate message for a field element

- <msg name=”..” key=”...” resource=”...”/>
  - **name**: specifies rule with which the msg is used, should be one of the rules in validation-rules.xml
  - **key**: specifies key from the resource bundle that should be added to the ActionError if validation fails
  - **resource**: if set to false, the value of key is taken as literal string

<field ...> Element: **<arg0 ...> child element**

- <arg0 key=”..” name=”..” resource=”..”>
  - Are used to pass additional values to the message
  - <arg0> is the **1st replacement** and <arg1> is the **2nd replacement,** and so on

<field ...> Element: **<var ...> child element**

```
<var>
    <var-name>...</var-name>
    <var-value>...</var-value>
</var>
```

- Set parameters that a field element may need to pass to one of its validation rules such as minimum and miximum values in a range validation.

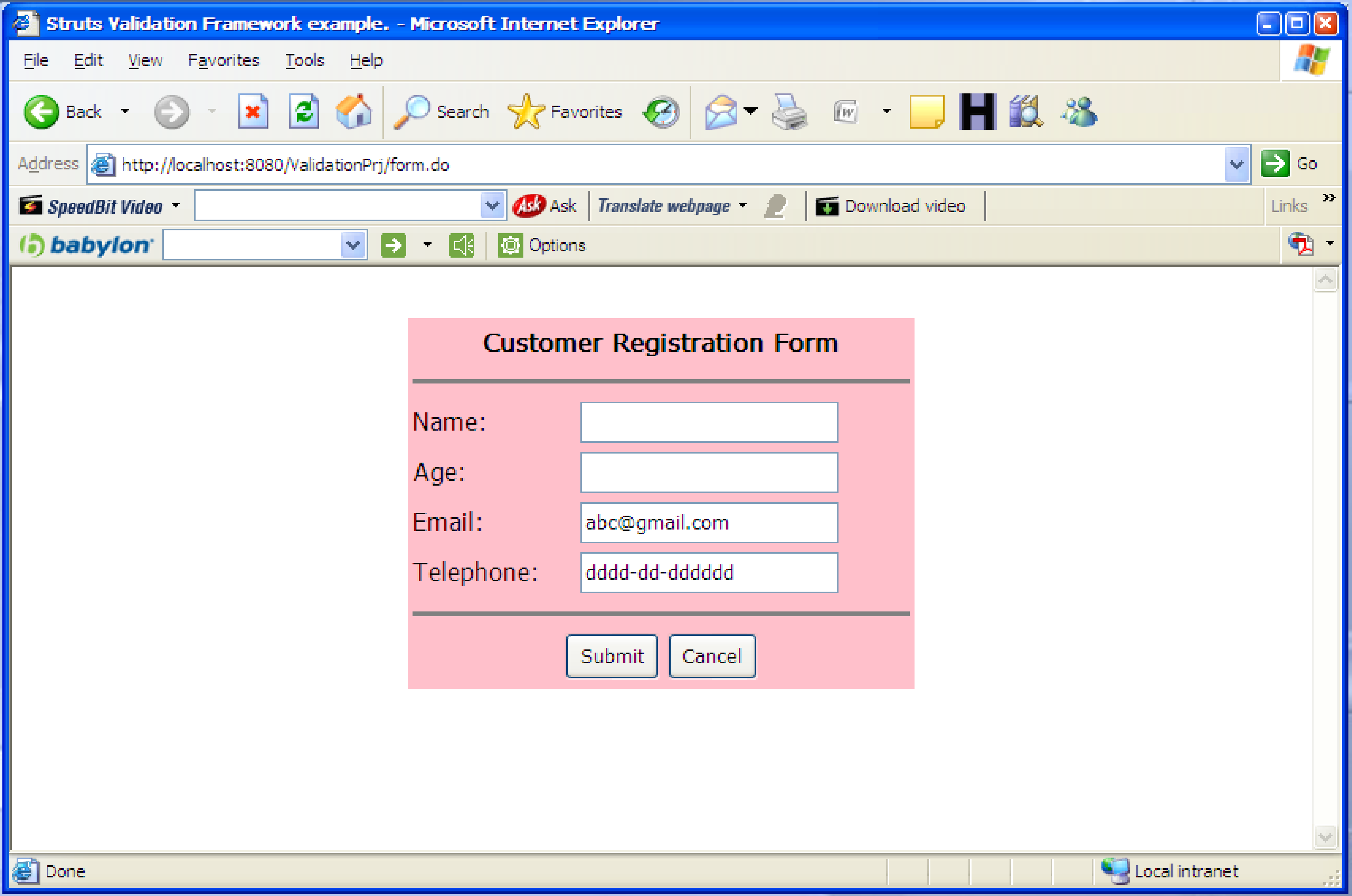
- Referenced by <argx> elements using ${var: var-name} syntax.

Validaton Notes:

- ✘ You cannot use **ActionForm** class with the Validator.
  - – Instead you need to use special subclasses of ActionForm class.

- ✓ Validator supports two types **special ActionForms**
  - – **Standard ActionForms**
    - ValidatorActionForm
    - ValidatorForm
  - – **Dynamic ActionForms**
    - DynaValidatorActionForm
    - DynaValidatorForm

Example


Struts Validation Framework example. - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
SpeedBit Video Ask Translate webpage Download video Links
babylon Options
Done
Local intranet

http://localhost:8080/ValidationPrj/form.do

Customer Registration Form

Name:	
Age:	
Email:	abc@gmail.com
Telephone:	dddd-dd-dddddd

Submit Cancel

Example : struts-config.xml

```
<form-bean name="userForm"
             type="org.apache.struts.validator.DynaValidatorForm" >

        <form-property  name="name"   type="java.lang.String" />
        <form-property  name="age"      type="java.lang.Integer" />
        <form-property  name="email"   type="java.lang.String"
                        initial="abc@gmail.com"/>
        <form-property  name="telephone"   type="java.lang.String"
                        initial="dddd-dd-dddddd"/>

</form-bean>
```

Example : struts-config.xml

```
<action  path="/userAction"
         type="user.data.UserAction"
         name="userForm"
         scope="request"
         validate="true"
         cancellable="true"
         input="/pages/form.jsp" >
   <forward name="success" path="/pages/success.jsp" />
</action>
```

Example: validation.xml

```
<formset>
   <form   name="userForm">
            <field   property="name"
                     depends="required , minlength">
                     <arg  key="label.name"/>
                     <arg   key="${var:minlength}"
                            name="minlength"
                            resource="false" />
                     <var>
                            <var-name> minlength</var-name>
                            <var-value>5</var-value>
                     </var>
            </field>
```

Example: validation.xml

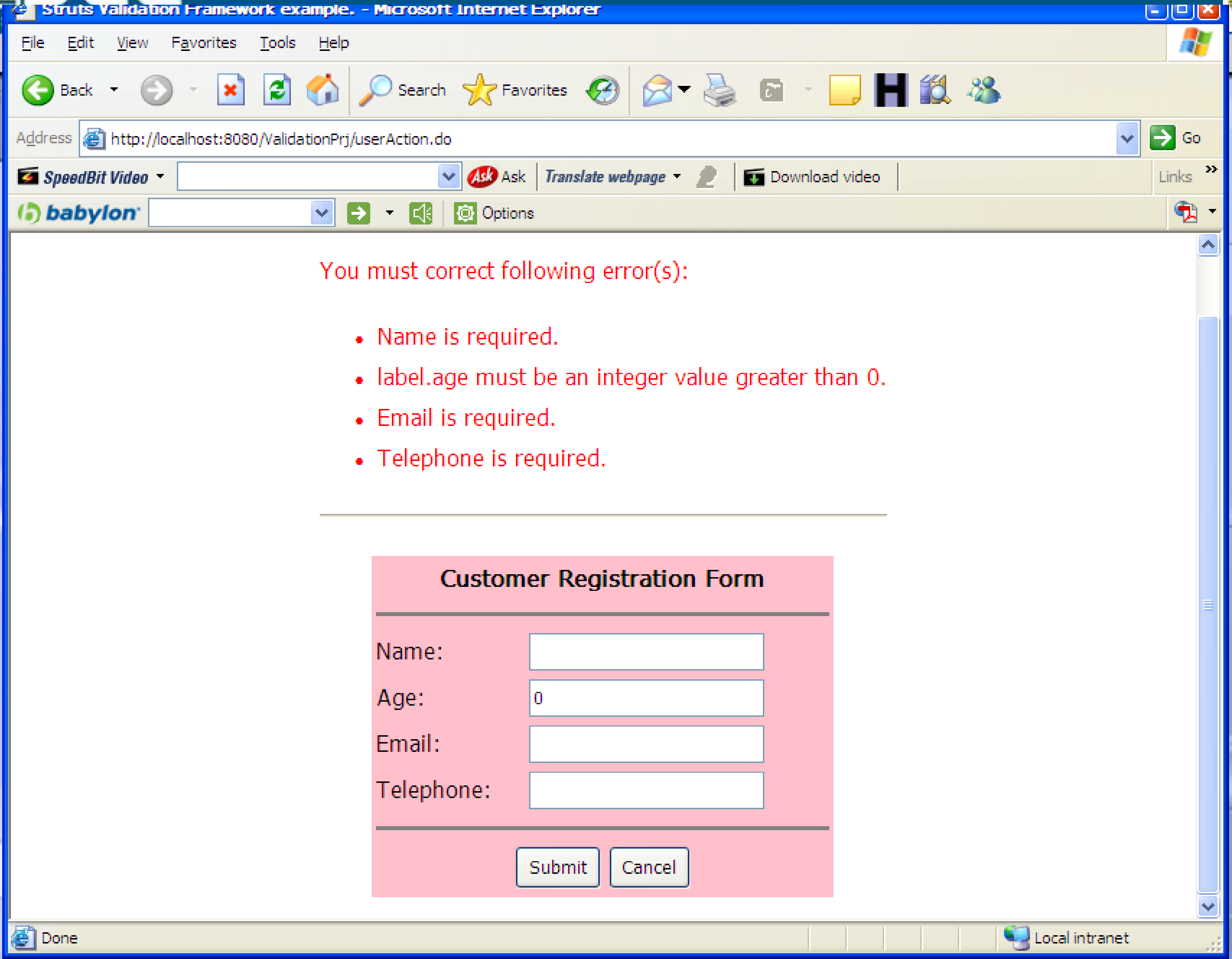
```
<field  property="age"
         depends="required , mask">
        <msg name="mask" key="userform.age" />
        <arg key="label.age" resource="false"/>
        <var>
                <var-name>mask</var-name>
                <var-value>^[1-9]{1}[0-9]*$</var-value>
        </var>
</field>
```

Example: validation.xml

```
   <field  property="email"
           depends="required ,email">
          <msg name="email" key="userform.email" />
          <arg key="label.email"/>
          <var>
                    <var-name>email</var-name>
                    <var-value>
                          ^[a-zA-Z]{1}[a-zA-Z0-9_]*@[a-zA-Z0-9]*.[a-zA-Z]*$
                    </var-value>
          </var>
 </field>
```

Example: validation.xml

```
    <field  property="telephone"
          depends="required,mask">
         <msg name="mask" key="userform.telephone" />
         <arg key="label.telephone"/>
         <var>
                 <var-name>mask</var-name>
                 <var-value>^00[1-9]{2}-[0-9]{2}-[0-9]{6}$</var-value>
         </var>
    </field>
  </form>
</formset>
```


Struts Validation Framework example. - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8080/ValidationPrj/userAction.do
Go
SpeedBit Video
Ask
Translate webpage
Download video
Links
babylon
Options

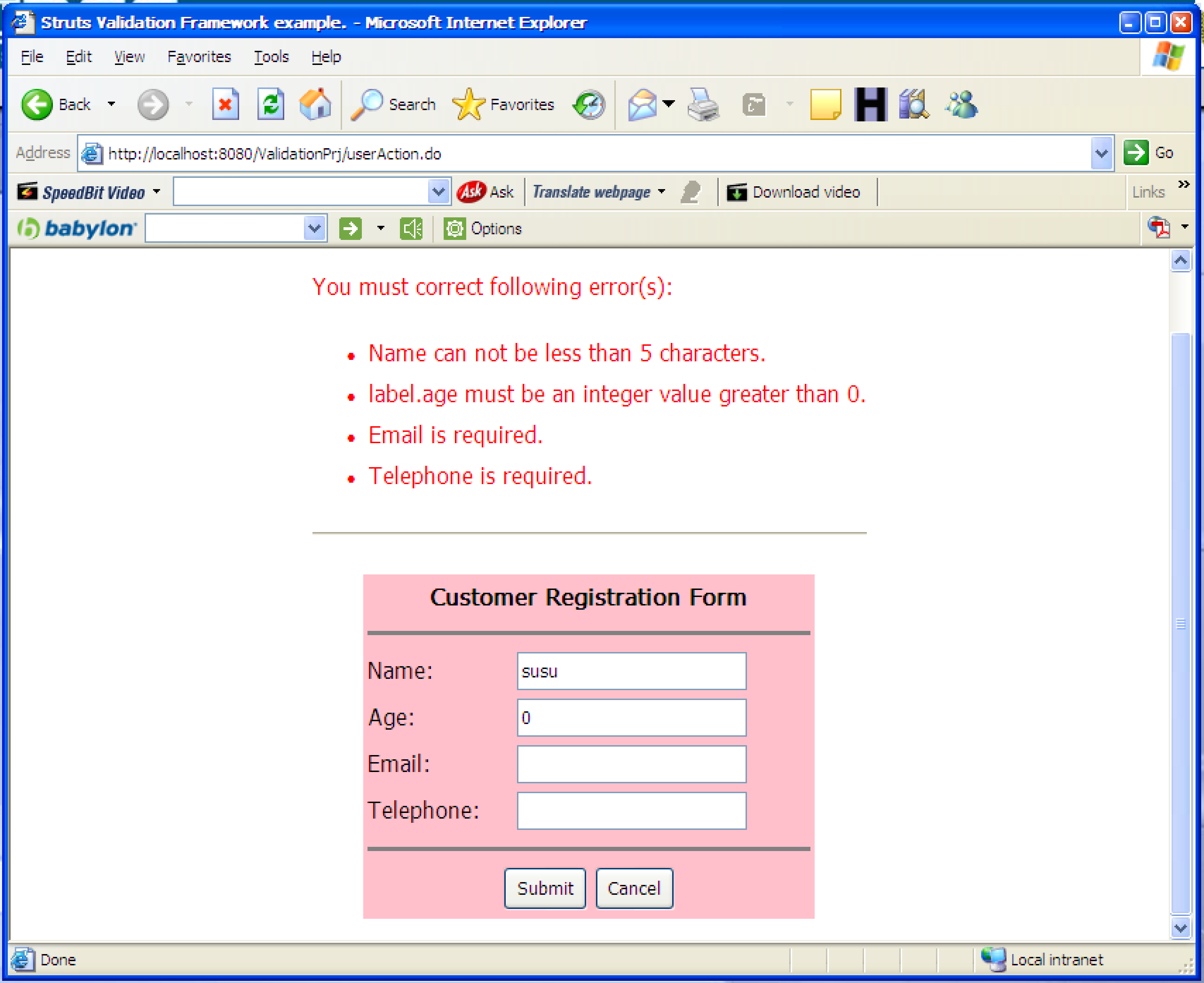
You must correct following error(s):

- Name is required.
- label.age must be an integer value greater than 0.
- Email is required.
- Telephone is required.

Customer Registration Form

Name:	
Age:	0
Email:	
Telephone:	

Submit Cancel



Struts Validation Framework example. - Microsoft Internet Explorer

http://localhost:8080/ValidationPrj/userAction.do

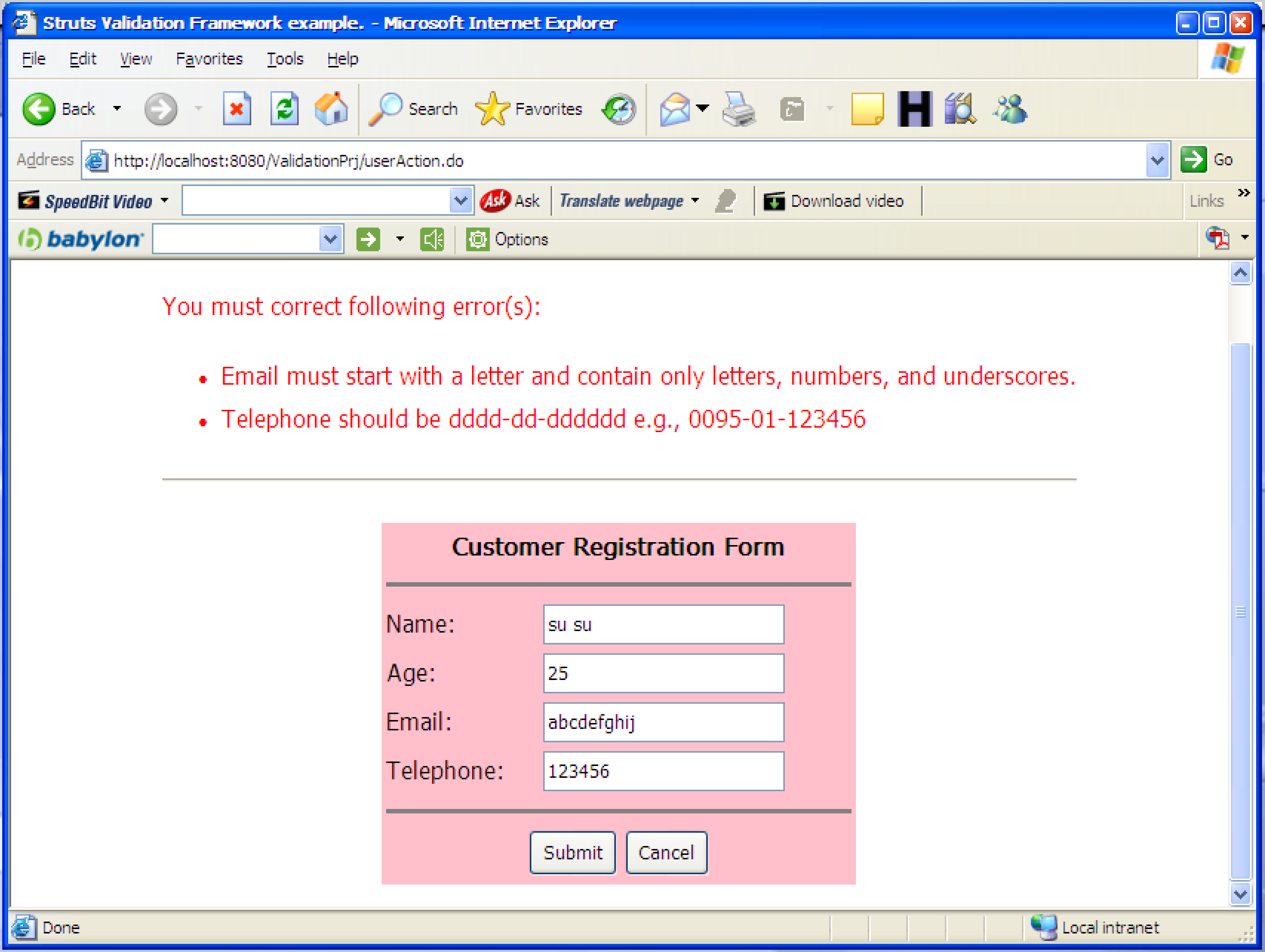
You must correct following error(s):

- Name can not be less than 5 characters.
- label.age must be an integer value greater than 0.
- Email is required.
- Telephone is required.

Customer Registration Form

Name:	susu
Age:	0
Email:	
Telephone:	

Submit Cancel



http://localhost:8080/ValidationPrj/userAction.do

You must correct following error(s):

- Email must start with a letter and contain only letters, numbers, and underscores.
- Telephone should be dddd-dd-dddddd e.g., 0095-01-123456

Customer Registration Form

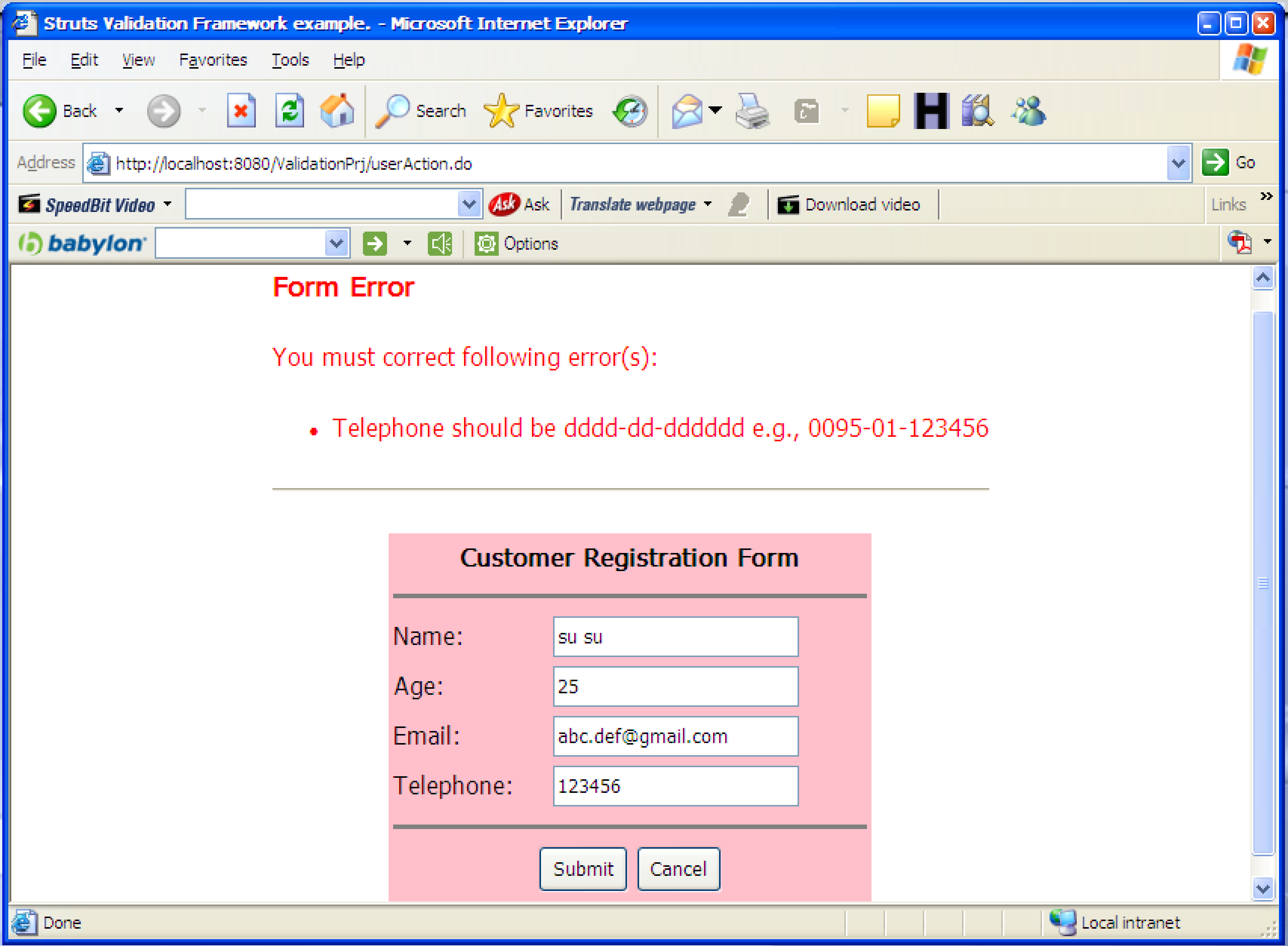
Name: su su

Age: 25

Email: abcdefghij

Telephone: 123456

Submit Cancel



Struts Validation Framework example. - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8080/ValidationPrj/userAction.do
Go
SpeedBit Video
Ask
Translate webpage
Download video
Links
babylon
Options
Done
Local intranet

Form Error

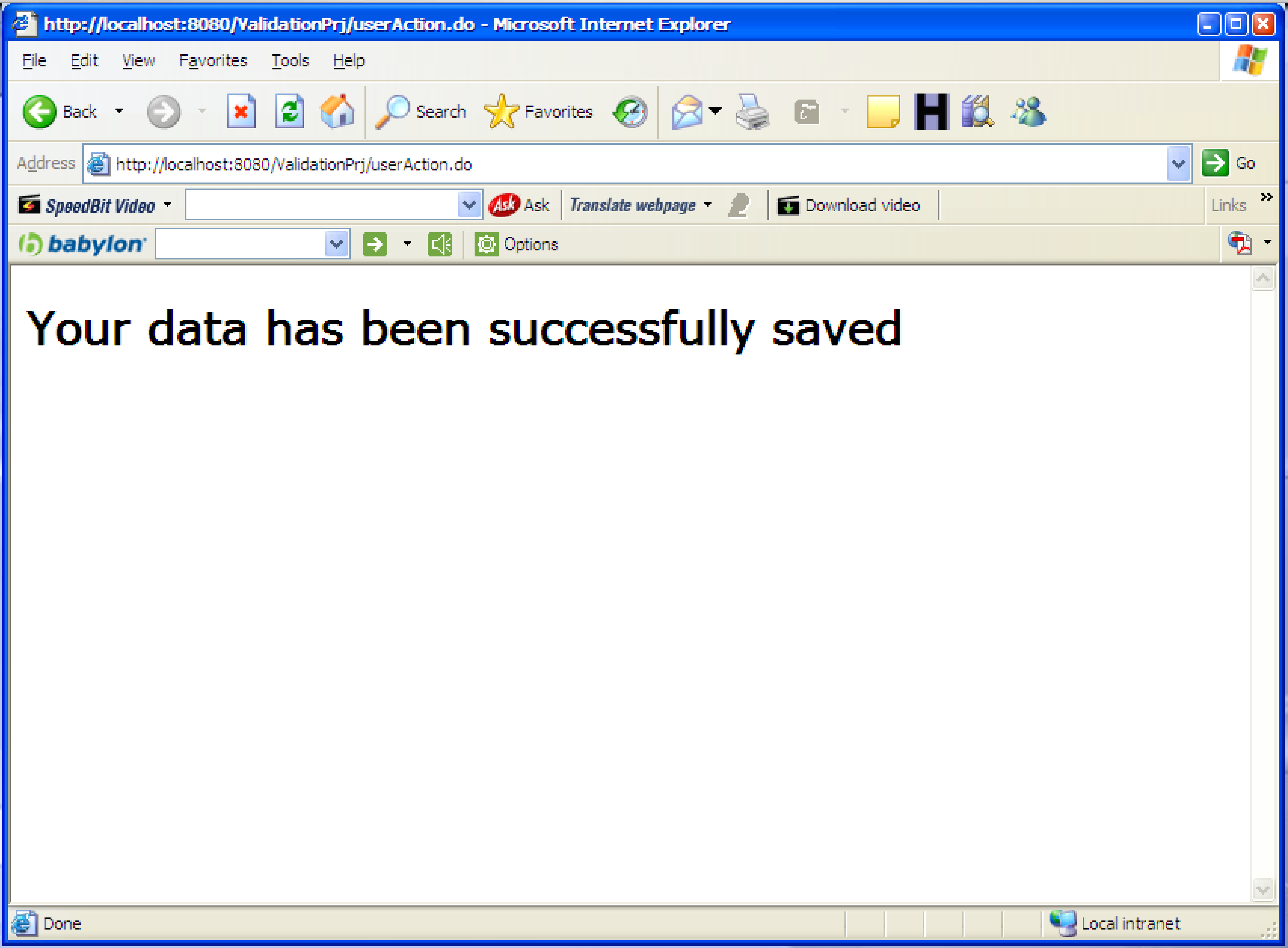
You must correct following error(s):

- Telephone should be dddd-dd-dddddd e.g., 0095-01-123456

Customer Registration Form

Name:	su su
Age:	25
Email:	abc.def@gmail.com
Telephone:	123456

Submit Cancel



Your data has been successfully saved

Thank you!