# *Processes Online* User Guide
## Process Automation Tutorial

## 1 Introduction

In this User Guide, we will show you how to automate a business process by way of an example, for a company called Pristine Consultants Pvt. Ltd.

The starting point is the State Transition model of the process. The example we take up is a Leave Application process, whose State Transition Model (Figure 1) was developed in *Processes Online User Guide – Process Modelling for Process Automation.*

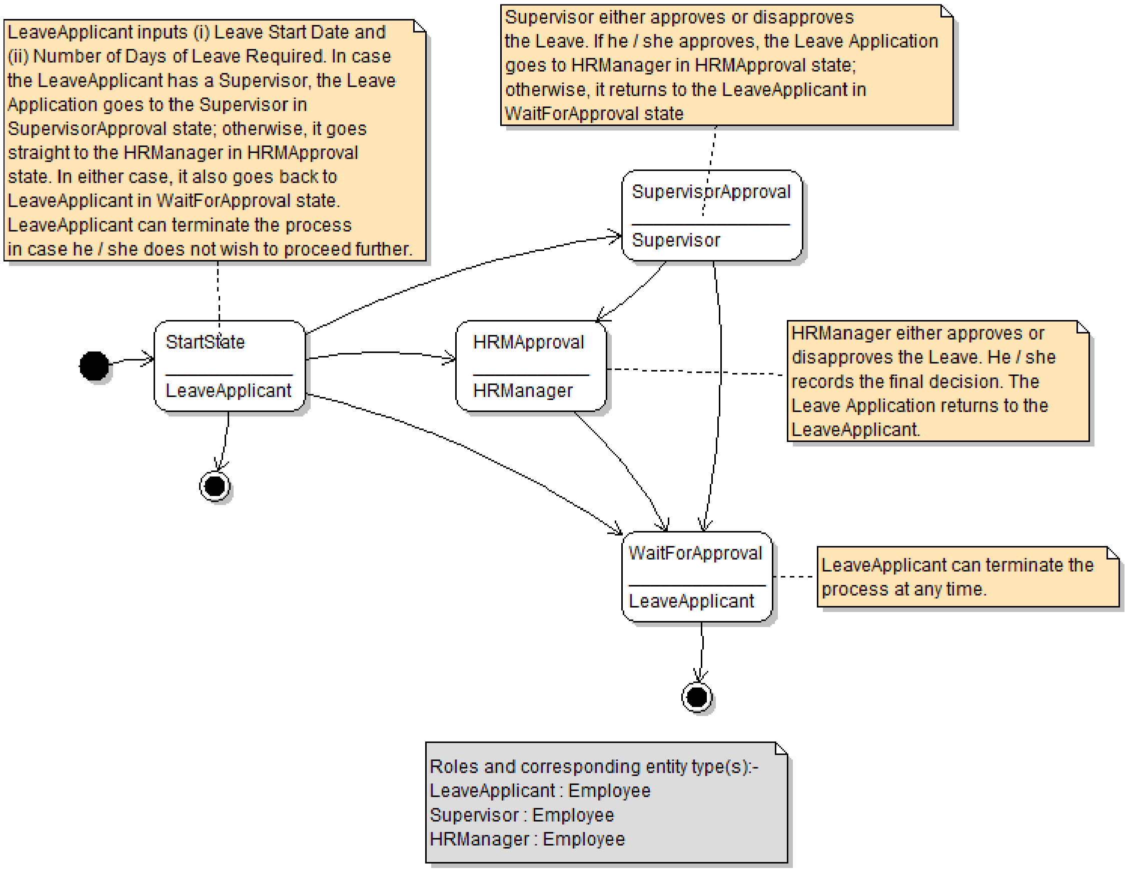
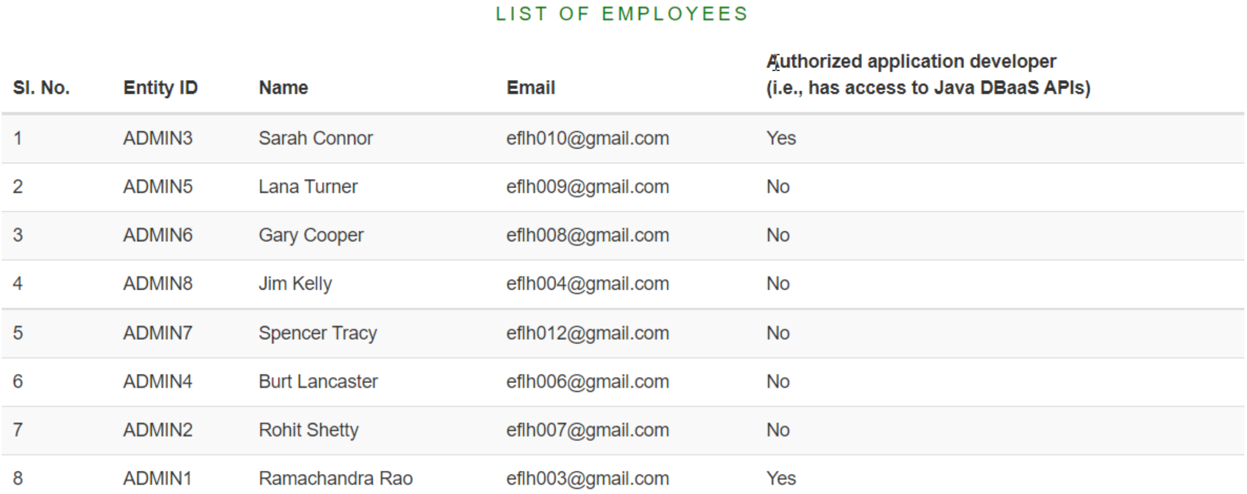

LeaveApplicant inputs (i) Leave Start Date and (ii) Number of Days of Leave Required. In case the LeaveApplicant has a Supervisor, the Leave Application goes to the Supervisor in SupervisorApproval state; otherwise, it goes straight to the HRManager in HRMApproval state. In either case, it also goes back to LeaveApplicant in WaitForApproval state. LeaveApplicant can terminate the process in case he / she does not wish to proceed further.
Supervisor either approves or disapproves the Leave. If he / she approves, the Leave Application goes to HRManager in HRMApproval state; otherwise, it returns to the LeaveApplicant in WaitForApproval state
SupervisorApproval
Supervisor
StartState
LeaveApplicant
HRMApproval
HRManager
HRManager either approves or disapproves the Leave. He / she records the final decision. The Leave Application returns to the LeaveApplicant.
WaitForApproval
LeaveApplicant
LeaveApplicant can terminate the process at any time.
Roles and corresponding entity type(s):-
LeaveApplicant : Employee
Supervisor : Employee
HRManager : Employee

Figure 1

The Administrator of Pristine Consultants Pvt. Ltd. has registered the following employees:-



LIST OF EMPLOYEES

| Sl. No. | Entity ID | Name | Email | Authorized application developer (i.e., has access to Java DBaaS APIs) |
|---|---|---|---|---|
| 1 | ADMIN3 | Sarah Connor | eflh010@gmail.com | Yes |
| 2 | ADMIN5 | Lana Turner | eflh009@gmail.com | No |
| 3 | ADMIN6 | Gary Cooper | eflh008@gmail.com | No |
| 4 | ADMIN8 | Jim Kelly | eflh004@gmail.com | No |
| 5 | ADMIN7 | Spencer Tracy | eflh012@gmail.com | No |
| 6 | ADMIN4 | Burt Lancaster | eflh006@gmail.com | No |
| 7 | ADMIN2 | Rohit Shetty | eflh007@gmail.com | No |
| 8 | ADMIN1 | Ramachandra Rao | eflh003@gmail.com | Yes |

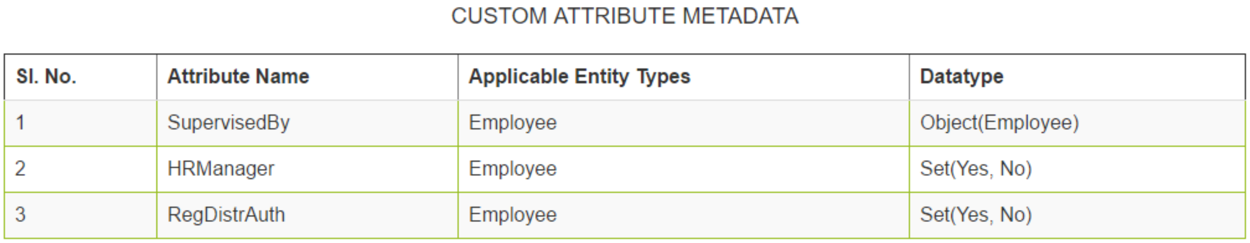
Figure 2

In the company, *any* employee can apply for leave, i.e., can play the LeaveApplicant role. Currently, Gary Cooper has been designated as HRManager. Also, currently, Burt Lancaster is the Supervisor of:-

- Gary Cooper
- Rohit Shetty
- Lana Turner
- Sarah Connor
- Spencer Tracy
- Jim Kelly

Ramachandra Rao and Burt Lancaster do not have Supervisors. (Each of them is his own Supervisor !!)

In order to reflect the above, Sarah Connor defines two additional custom attributes, viz., SupervisedBy and HRManager, with the following metadata:-



CUSTOM ATTRIBUTE METADATA

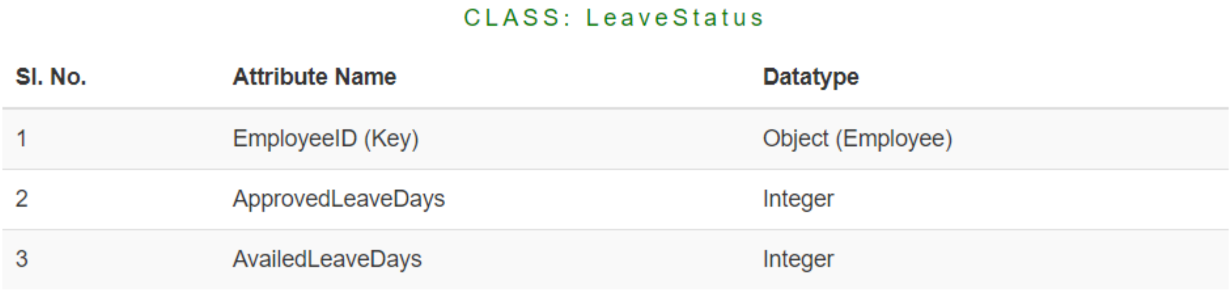
| Sl. No. | Attribute Name | Applicable Entity Types | Datatype |
|---|---|---|---|
| 1 | SupervisedBy | Employee | Object(Employee) |
| 2 | HRManager | Employee | Set(Yes, No) |
| 3 | RegDistrAuth | Employee | Set(Yes, No) |

Figure 3

She goes on to set HRManager to Yes in case of Gary Cooper. She also sets SupervisedBy to Burt Lancaster in case of Gary Cooper, Rohit Shetty, Lana Turner, Sarah Connor, Spencer Tracey, and Jim Kelly.

# 2 LEAVESTATUS CLASS

Each employee’s leave status for the current year is maintained by the company. The HRManager’s decision to approve or reject a leave application is based on viewing the applicant’s leave status. Accordingly, Sarah Connor creates the following **LeaveStatus** class:-



CLASS: LeaveStatus

| Sl. No. | Attribute Name | Datatype |
|---|---|---|
| 1 | EmployeeID (Key) | Object (Employee) |
| 2 | ApprovedLeaveDays | Integer |
| 3 | AvailedLeaveDays | Integer |

Figure 4

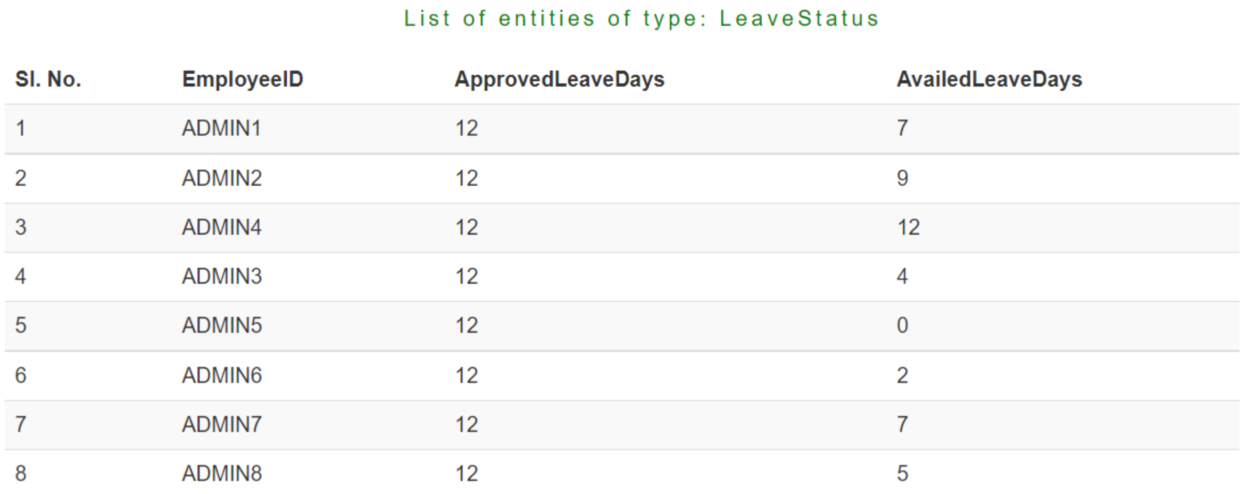
**Note**:- In the rest of this tutorial, we will consider the maintenance of LeaveStatus objects as being outside the scope of the Leave Application business process.

The meaning of each Attribute Name is shown in the table below:-

| Attribute | Meaning |
|---|---|
| EmployeeID | The Entity ID of an entity belonging to the Employee class |
| ApprovedLeaveDays | The number of days of leave approved for the employee for the current year |
| AvailedLeaveDays | The number of days of leave availed by the employee to date in the current year |

Figure 5

To facilitate testing, Sarah Connor goes on to create the following LeaveStatus objects:-



List of entities of type: LeaveStatus

| Sl. No. | EmployeeID | ApprovedLeaveDays | AvailedLeaveDays |
|---|---|---|---|
| 1 | ADMIN1 | 12 | 7 |
| 2 | ADMIN2 | 12 | 9 |
| 3 | ADMIN4 | 12 | 12 |
| 4 | ADMIN3 | 12 | 4 |
| 5 | ADMIN5 | 12 | 0 |
| 6 | ADMIN6 | 12 | 2 |
| 7 | ADMIN7 | 12 | 7 |
| 8 | ADMIN8 | 12 | 5 |

Figure 6

# 3 Leave Application process: Project setup in Eclipse

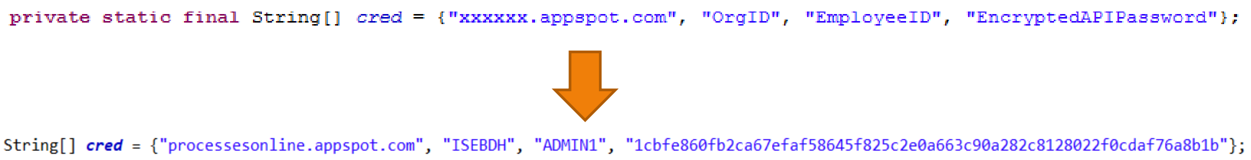
Start the Eclipse IDE. Click **File > New > Dynamic Web Project**. In the form that presents itself, enter the project name as **LeaveApplication**. (With no blank spaces). Click **Finish**.

Find a file, Enflow.jar, in the folder, **Processes Online_DBaaS_Kit**. Next, expand the LeaveApplication project in the Eclipse Project Explorer, to find the **lib** folder within the WebContent and WEB-INF folder. Now drop Enflow.jar into the lib folder.

Now right click Enflow.jar, and click **Build Path > Add To Build Path**.

Find a folder, com, in the folder, **Processes Online_DBaaS_Kit**. Next, expand the LeaveApplication project in the Eclipse Project Explorer, to find the src folder within the Java Resources folder. Now drop com into the src folder.

The next is to input Sarah Connor's credentials into the class, Cred.java, which is within the com.enterpriseflow.customapp package. Expand Cred.java to find the following definition of a String[] called cred. Input Sarah Connor's credentials therein (Recollect that Sarah Connor is an authorized application developer):-



```
private static final String[] cred = {"xxxxxx.appspot.com", "OrgID", "EmployeeID", "EncryptedAPIPassword"};
```

```
String[] cred = {"processesonline.appspot.com", "ISEBDH", "ADMIN1", "1cbfe860fb2ca67efaf58645f825c2e0a663c90a282c8128022f0cdaf76a8b1b"};
```

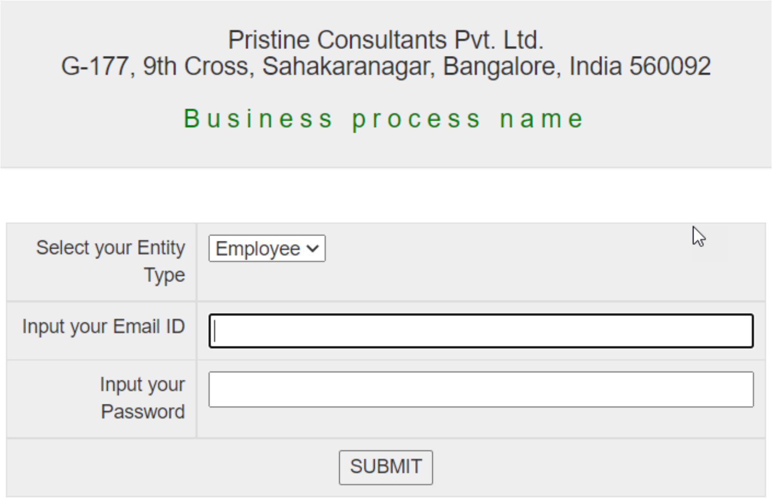
Figure 7

Find a folder, WebContent, in the folder, **Processes Online_DBaaS_Kit**. **This folder contains template JSPs, using which the core JSPs of the Leave Application process will be derived.** Drop the contents of this folder into the WebContent folder of the LeaveApplication project in Eclipse Project Explorer.

At this stage, select LeaveApplication project in Eclipse Project Explorer. Then click **Run -> Debug As > Debug on Server**.

Then, visit http://localhost:8080/LeaveApplication/.

You will now see the following screen, which enables an entity of any of the 8 entity types (Employee, Customer, Vendor, Member, Faculty, Student, Doctor, Patient) to login; this is nothing but the file, index.jsp, within the WebContent folder :-



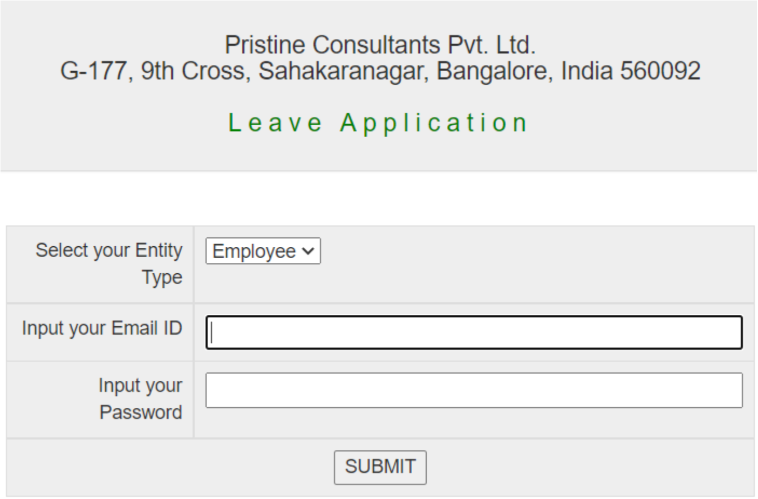
Pristine Consultants Pvt. Ltd.
G-177, 9th Cross, Sahakaranagar, Bangalore, India 560092

Business process name

| | |
|---|---|
| Select your Entity Type | Employee |
| Input your Email ID | |
| Input your Password | |

Figure 8

In the file, index.jsp, change the string “Business process name” to “Leave Application”, whereupon the above changes to the following:-



Pristine Consultants Pvt. Ltd.
G-177, 9th Cross, Sahakaranagar, Bangalore, India 560092

Leave Application

| | |
|---|---|
| Select your Entity Type | Employee |
| Input your Email ID | |
| Input your Password | |

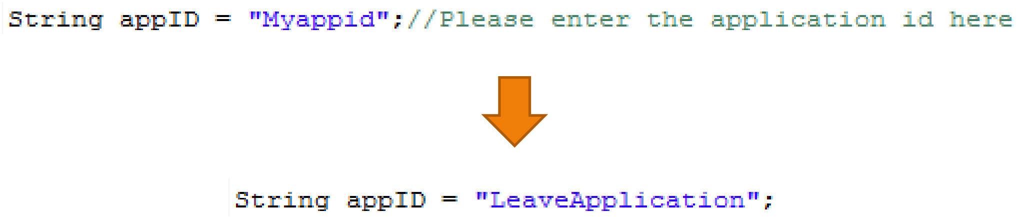
Figure 9

# 4 CUSTOMLOGIN.JSP

When an entity logs in, he / she is first taken to the page, customlogin.jsp, within the CustomApp folder.

Next, expand the customlogin.jsp file in Eclipse. Here again, replace the text, “Business process name” with the text, “Leave Application”.

You will also find a String called appID being defined. Now change the definition to **LeaveApplication**. (Please do not leave any blank space in between Leave and Application).



```
String appID = "Myappid";//Please enter the application id here
```

```
String appID = "LeaveApplication";
```

Figure 10

Notice that customlogin.jsp has a buttons, viz., **INITIATE THE PROCESS**.

The person who logs in will see the button, **INITIATE THE PROCESS**, only if he / she has been assigned StartState role of the process (in our example, the role, LeaveApplicant of the LeaveApplication process). Remember that we want *every* employee to be able to initiate the process, i.e., apply for leave. Therefore, change the line:-

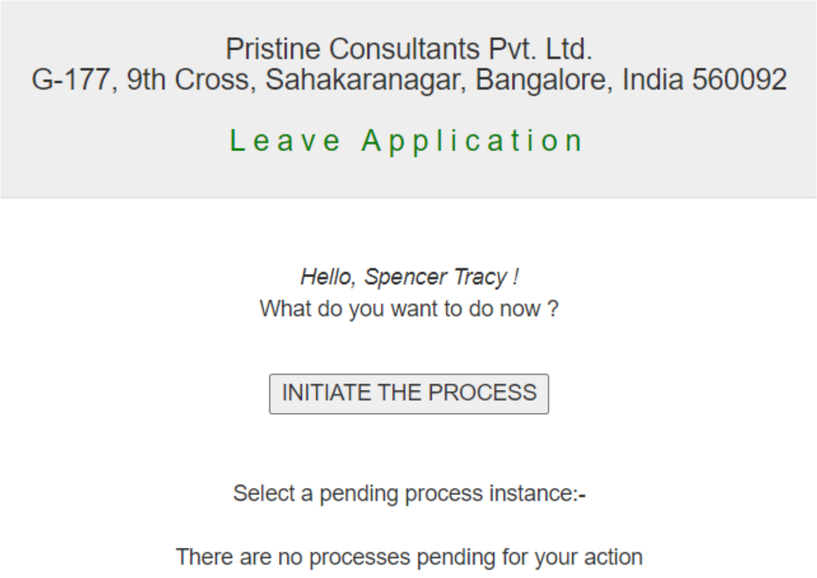
*boolean caninit = Bizproapi.roleAuthorized(entityType, email, appID);*

to:-

*boolean caninit = entityType.equals("Employee");*

The person who logs in will also see zero or more buttons, **SELECT**, to enable him / her to select the process instances awaiting his / her action.

If, now, Spencer Tracy were to login, then customlogin.jsp would present itself as follows:-



Pristine Consultants Pvt. Ltd.
G-177, 9th Cross, Sahakaranagar, Bangalore, India 560092

Leave Application

*Hello, Spencer Tracy !*
What do you want to do now ?

Select a pending process instance:-

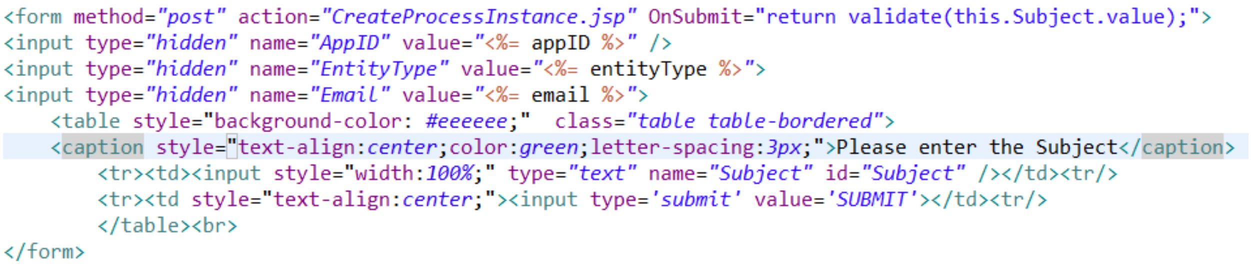
There are no processes pending for your action

Figure 11

If Spencer were to click **INITIATE THE PROCESS in** the above screen, he would be taken to Initiate.jsp.

## 5 INITIATE.JSP AND CREATEPROCESSINSTANCE.JSP

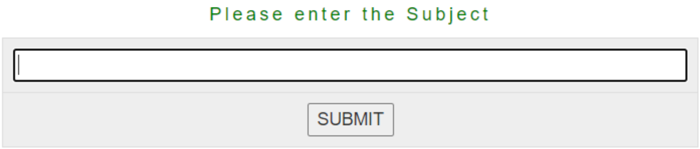
Initiate.jsp contains the **Initiate Form** shown below:-



```
<form method="post" action="CreateProcessInstance.jsp" OnSubmit="return validate(this.Subject.value);">
<input type="hidden" name="AppID" value="<%= appID %>" />
<input type="hidden" name="EntityType" value="<%= entityType %>">
<input type="hidden" name="Email" value="<%= email %>">
    <table style="background-color: #eeeeee;"  class="table table-bordered">
    <caption style="text-align:center;color:green;letter-spacing:3px;">Please enter the Subject</caption>
        <tr><td><input style="width:100%;" type="text" name="Subject" id="Subject" /></td><tr/>
        <tr><td style="text-align:center;"><input type='submit' value='SUBMIT'></td><tr/>
        </table><br>
</form>
```

Figure 12

Initiate.jsp presents itself to Spencer as follows:-



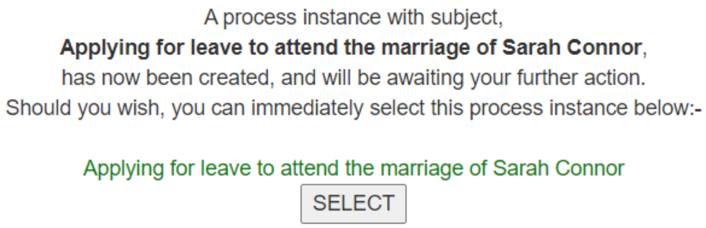
Please enter the Subject

Figure 13

Spencer can enter whatever he wishes to in the Subject field above, e.g., “Applying for leave to attend the marriage of Sarah Connor”.

Upon clicking SUBMIT, the value of the field, Subject, is taken to CreateProcessInstance.jsp, where a new instance of, in our example, the business process with ID, LeaveApplication, gets created with the Subject field value. The process instance will then be pending with the same actor in StartState.

You are encouraged to browse CreateProcessInstance.jsp to understand what is happening. Notice, in particular that a method called `Bizfloapi.initiateProcessInstance` is invoked, which indeed causes a new LeaveApplication process instance to be created; CreateProcessInstance.jsp presents itself to Spencer Tracy as follows:-



A process instance with subject,
**Applying for leave to attend the marriage of Sarah Connor**,
has now been created, and will be awaiting your further action.
Should you wish, you can immediately select this process instance below:-

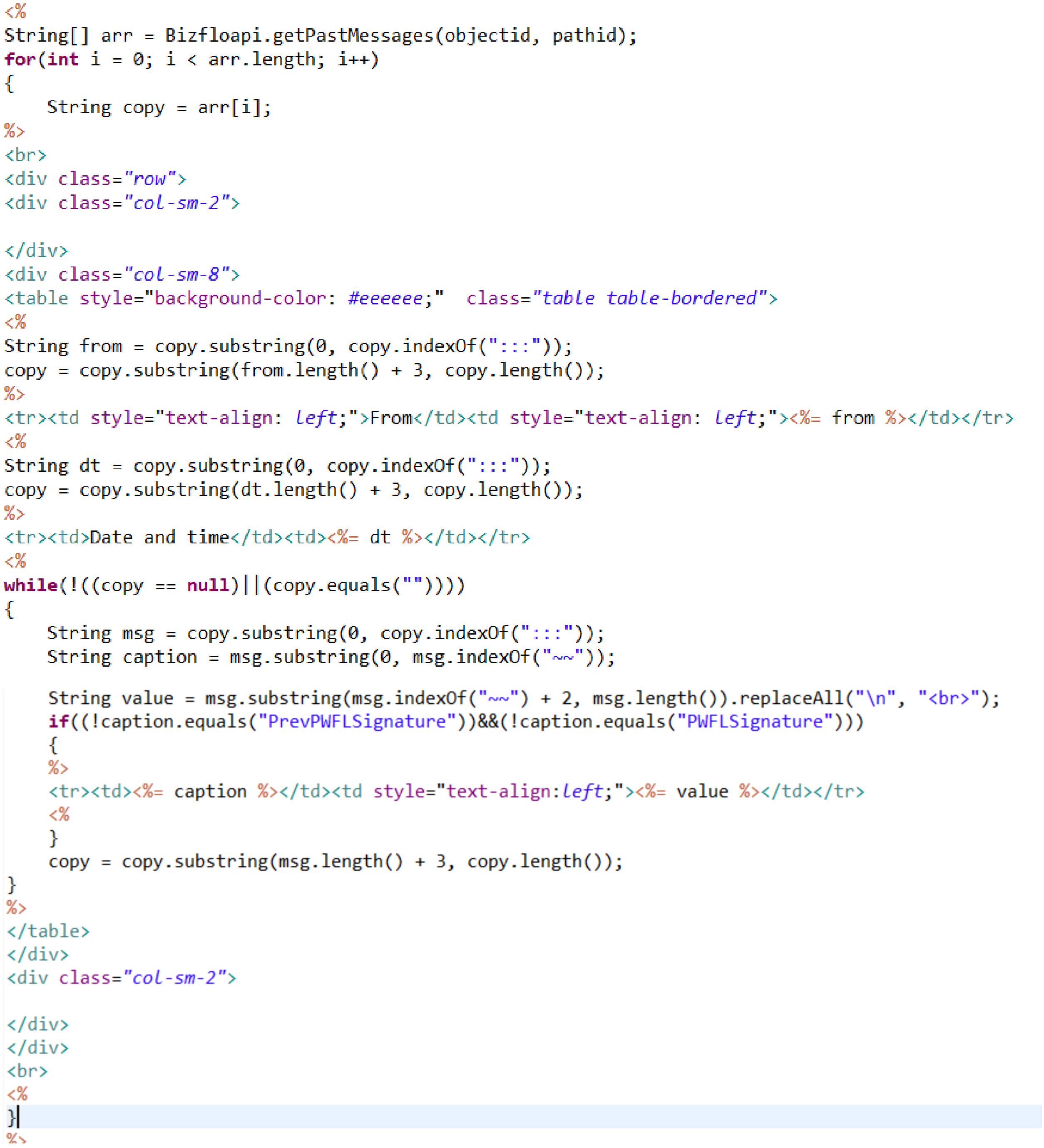
Applying for leave to attend the marriage of Sarah Connor

Figure 14

# 6 StartState.jsp: The state JSP template

Notice that within the appjsps folder there is a JSP called StartState.jsp. **We will be using this JSP as a template for creation of the JSPs corresponding to *all the states of a process*.** (In our example, viz., LeaveApplication process, there will be four state JSPs, viz., StartState.jsp, SupervisorApproval.jsp, HRMApproval.jsp and WaitForApproval.jsp).

The body of the template JSP contains the following code:-



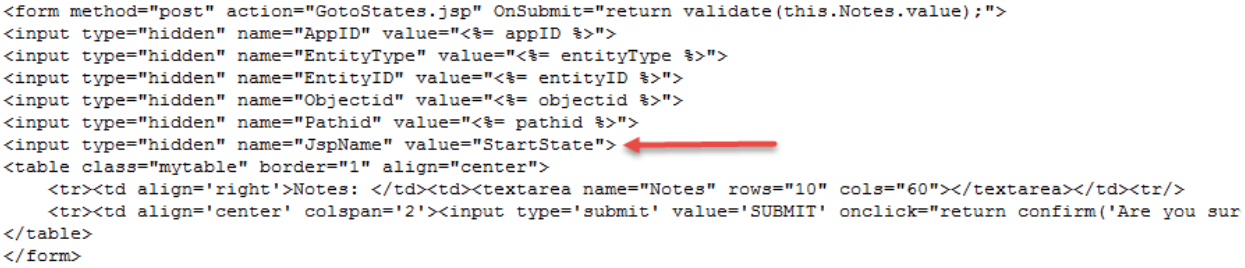
```
<%
String[] arr = Bizfloapi.getPastMessages(objectid, pathid);
for(int i = 0; i < arr.length; i++)
{
    String copy = arr[i];
%>
<br>
<div class="row">
<div class="col-sm-2">

</div>
<div class="col-sm-8">
<table style="background-color: #eeeeee;"  class="table table-bordered">
<%
String from = copy.substring(0, copy.indexOf(":::"));
copy = copy.substring(from.length() + 3, copy.length());
%>
<tr><td style="text-align: left;">From</td><td style="text-align: left;"><%= from %></td></tr>
<%
String dt = copy.substring(0, copy.indexOf(":::"));
copy = copy.substring(dt.length() + 3, copy.length());
%>
<tr><td>Date and time</td><td><%= dt %></td></tr>
<%
while(!((copy == null)||(copy.equals(""))))
{
    String msg = copy.substring(0, copy.indexOf(":::"));
    String caption = msg.substring(0, msg.indexOf("~~"));

    String value = msg.substring(msg.indexOf("~~") + 2, msg.length()).replaceAll("\n", "<br>");
    if((!caption.equals("PrevPWFLSignature"))&&(!caption.equals("PWFLSignature")))
    {
    %>
    <tr><td><%= caption %></td><td style="text-align:left;"><%= value %></td></tr>
    <%
    }
    copy = copy.substring(msg.length() + 3, copy.length());
}
%>
</table>
</div>
<div class="col-sm-2">

</div>
</div>
<br>
<%
}
%>
```

Figure 15

Here, **objected** is the unique id of the business process instance, while **pathid** is the unique id of the path along which the process instance has flowed to the current actor.

The above code pastes on the page, the messages from previous actors along the path, pathid.

The body of the template JSP (i.e., StartState.jsp) contains the following **State Transition Form**:-



```
<form method="post" action="GotoStates.jsp" OnSubmit="return validate(this.Notes.value);">
<input type="hidden" name="AppID" value="<%= appID %>">
<input type="hidden" name="EntityType" value="<%= entityType %>">
<input type="hidden" name="EntityID" value="<%= entityID %>">
<input type="hidden" name="Objectid" value="<%= objectid %>">
<input type="hidden" name="Pathid" value="<%= pathid %>">
<input type="hidden" name="JspName" value="StartState">
<table class="mytable" border="1" align="center">
    <tr><td align='right'>Notes: </td><td><textarea name="Notes" rows="10" cols="60"></textarea></td><tr/>
    <tr><td align='center' colspan='2'><input type='submit' value='SUBMIT' onclick="return confirm('Are you sur
</table>
</form>
```

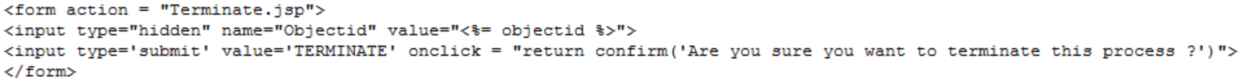
Figure 16

**appID** is the unique ID of the business process. (In our example, it will get set at run time to "LeaveApplication"). **entityType** and **entityID** are the EntityType and EntityID of the current actor.

**Important Note**:-The value of the **JspName** hidden field should be changed to the name of the state represented by the JSP.

Upon clicking the **SUBMIT** button, the data in the form is taken to **GotoStates.jsp**, indeed to a block of code therein corresponding to the state represented by the present JSP. It is indeed this block of code that causes state transitions from out of the present state.

The body of the template JSP contains the following **Terminate Button**:-

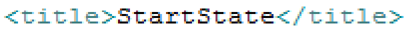


```
<form action = "Terminate.jsp">
<input type="hidden" name="Objectid" value="<%= objectid %>">
<input type='submit' value='TERMINATE' onclick = "return confirm('Are you sure you want to terminate this process ?')">
</form>
```

Figure 17

Upon clicking the **TERMINATE** button, Terminate.jsp gets accessed, whereupon the present business process instance, whose unique ID is **objected** gets archived.

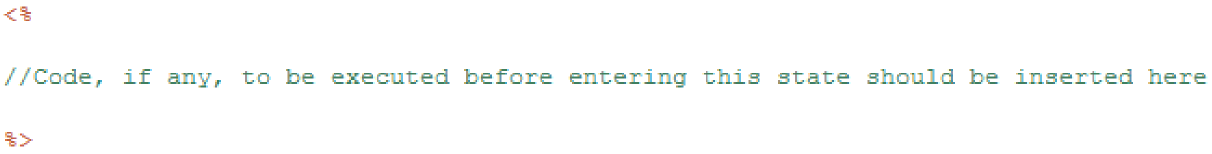
The body of the template JSP contains the following title tag:-



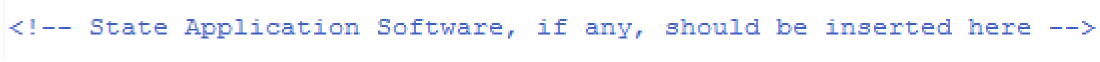
```
<title>StartState</title>
```

**Note**:-Preferably, change the text from **StartState** to something else that reflects the activities performed in the state represented by the JSP.

Notice also the following lines in the template JSP:-



```
<%

//Code, if any, to be executed before entering this state should be inserted here

%>
```

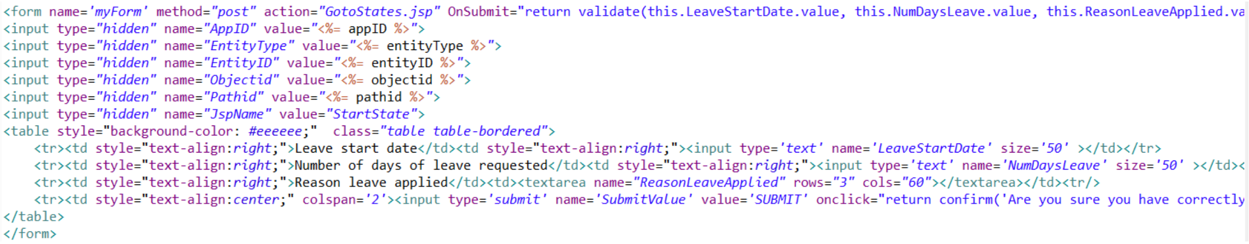


```
<!-- State Application Software, if any, should be inserted here -->
```

# 7 Creation of the State JSPs of the LeaveApplication Process

The LeaveApplication process has 3 more states, viz., SupervisorApproval, HRMApproval and WaitForApproval, in addition to StartState (Recollect this from Figure 1 above). Now create 3 more jsps within the same folder, corresponding to these 3 states, viz., SupervisorApproval.jsp, HRMApproval.jsp and WaitForApproval.jsp, using StartState.jsp as the template. In all these JSPs, replace the text, “Business process name” with the text, “Leave Application”.

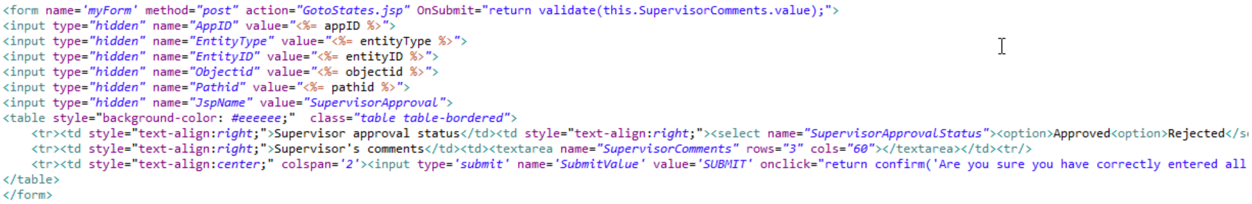
In StartState, LeaveApplicant inputs (i) Leave Start Date, (ii) Number of Days of Leave, and (iii) Reason for Leave. Accordingly edit the State Transition Form in StartState.jsp as shown below:-



```
<form name='myForm' method="post" action="GotoStates.jsp" OnSubmit="return validate(this.LeaveStartDate.value, this.NumDaysLeave.value, this.ReasonLeaveApplied.va
<input type="hidden" name="AppID" value="<%= appID %>">
<input type="hidden" name="EntityType" value="<%= entityType %>">
<input type="hidden" name="EntityID" value="<%= entityID %>">
<input type="hidden" name="Objectid" value="<%= objectid %>">
<input type="hidden" name="Pathid" value="<%= pathid %>">
<input type="hidden" name="JspName" value="StartState">
<table style="background-color: #eeeeee;"  class="table table-bordered">
    <tr><td style="text-align:right;">Leave start date</td><td style="text-align:right;"><input type='text' name='LeaveStartDate' size='50' ></td></tr>
    <tr><td style="text-align:right;">Number of days of leave requested</td><td style="text-align:right;"><input type='text' name='NumDaysLeave' size='50' ></td><
    <tr><td style="text-align:right;">Reason leave applied</td><td><textarea name="ReasonLeaveApplied" rows="3" cols="60"></textarea></td><tr/>
    <tr><td style="text-align:center;" colspan='2'><input type='submit' name='SubmitValue' value='SUBMIT' onclick="return confirm('Are you sure you have correctly
</table>
</form>
```

Figure 18

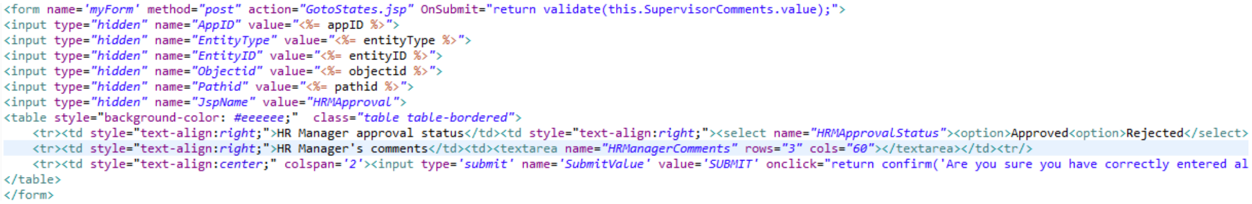
In SupervisorApproval state, Supervisor cannot archive, i.e., terminate the process. Hence, remove the Archive Button from SupervisorApproval.jsp. Next, edit the State Transition Form in SupervisorApproval.jsp as shown below:-



```
<form name='myForm' method="post" action="GotoStates.jsp" OnSubmit="return validate(this.SupervisorComments.value);">
<input type="hidden" name="AppID" value="<%= appID %>">
<input type="hidden" name="EntityType" value="<%= entityType %>">
<input type="hidden" name="EntityID" value="<%= entityID %>">
<input type="hidden" name="Objectid" value="<%= objectid %>">
<input type="hidden" name="Pathid" value="<%= pathid %>">
<input type="hidden" name="JspName" value="SupervisorApproval">
<table style="background-color: #eeeeee;"  class="table table-bordered">
    <tr><td style="text-align:right;">Supervisor approval status</td><td style="text-align:right;"><select name="SupervisorApprovalStatus"><option>Approved<option>Rejected</s
    <tr><td style="text-align:right;">Supervisor's comments</td><td><textarea name="SupervisorComments" rows="3" cols="60"></textarea></td><tr/>
    <tr><td style="text-align:center;" colspan='2'><input type='submit' name='SubmitValue' value='SUBMIT' onclick="return confirm('Are you sure you have correctly entered all
</table>
</form>
```

Figure 19

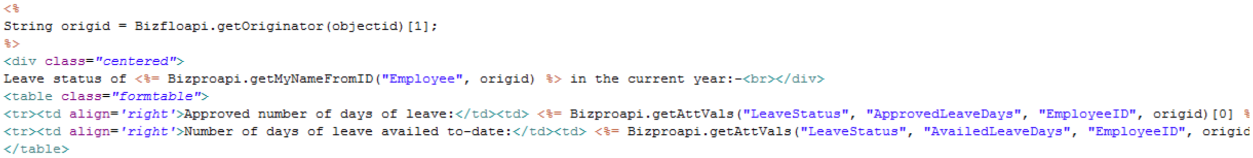
In HRMApproval state, HRManager cannot archive, i.e., terminate the process. Hence, remove the Archive Button from HRMApproval.jsp. Next, edit the State Transition Form in HRMApproval.jsp as shown below:-



```
<form name='myForm' method="post" action="GotoStates.jsp" OnSubmit="return validate(this.SupervisorComments.value);">
<input type="hidden" name="AppID" value="<%= appID %>">
<input type="hidden" name="EntityType" value="<%= entityType %>">
<input type="hidden" name="EntityID" value="<%= entityID %>">
<input type="hidden" name="Objectid" value="<%= objectid %>">
<input type="hidden" name="Pathid" value="<%= pathid %>">
<input type="hidden" name="JspName" value="HRMApproval">
<table style="background-color: #eeeeee;"  class="table table-bordered">
    <tr><td style="text-align:right;">HR Manager approval status</td><td style="text-align:right;"><select name="HRMApprovalStatus"><option>Approved<option>Rejected</select>
    <tr><td style="text-align:right;">HR Manager's comments</td><td><textarea name="HRManagerComments" rows="3" cols="60"></textarea></td><tr/>
    <tr><td style="text-align:center;" colspan='2'><input type='submit' name='SubmitValue' value='SUBMIT' onclick="return confirm('Are you sure you have correctly entered al
</table>
</form>
```

Figure 20

To help him decide on approval or rejection of the leave application, HRManager would like to view the leave status of the applicant. Hence, put the following code in the body of HRMApproval.jsp, in place of the line, *“<!--State Application Software, if any, should be inserted here -->”*:-



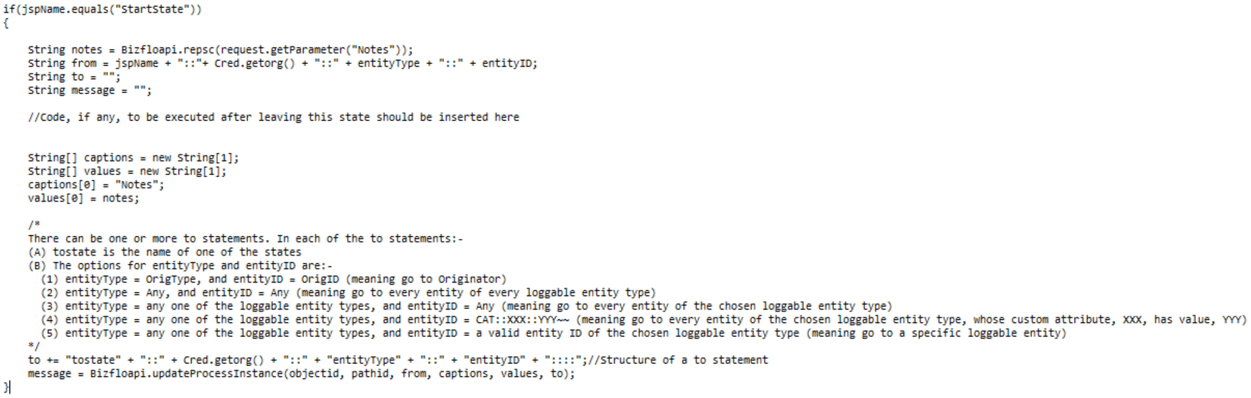
```
<%
String origid = Bizfloapi.getOriginator(objectid)[1];
%>
<div class="centered">
Leave status of <%= Bizproapi.getMyNameFromID("Employee", origid) %> in the current year:-<br></div>
<table class="formtable">
<tr><td align='right'>Approved number of days of leave:</td><td> <%= Bizproapi.getAttVals("LeaveStatus", "ApprovedLeaveDays", "EmployeeID", origid)[0] %
<tr><td align='right'>Number of days of leave availed to-date:</td><td> <%= Bizproapi.getAttVals("LeaveStatus", "AvailedLeaveDays", "EmployeeID", origid
</table>
```

Figure 21

No state transitions take place from WaitForApproval state. Hence, delete the State Transition Form from WaitForApproval.jsp. Since the LeaveApplicant can terminate the process at any time, we leave the Archive Button in place on this JSP.

# 8 GOTOSTATES.JSP

GotoStates.jsp contains the following template code block for the StartState, and for all other states of a business process:-



```
if(jspName.equals("StartState"))
{

    String notes = Bizfloapi.repsc(request.getParameter("Notes"));
    String from = jspName + "::"+ Cred.getorg() + "::" + entityType + "::" + entityID;
    String to = "";
    String message = "";

    //Code, if any, to be executed after leaving this state should be inserted here


    String[] captions = new String[1];
    String[] values = new String[1];
    captions[0] = "Notes";
    values[0] = notes;

    /*
    There can be one or more to statements. In each of the to statements:-
    (A) tostate is the name of one of the states
    (B) The options for entityType and entityID are:-
      (1) entityType = OrigType, and entityID = OrigID (meaning go to Originator)
      (2) entityType = Any, and entityID = Any (meaning go to every entity of every loggable entity type)
      (3) entityType = any one of the loggable entity types, and entityID = Any (meaning go to every entity of the chosen loggable entity type)
      (4) entityType = any one of the loggable entity types, and entityID = CAT::XXX::YYY~~ (meaning go to every entity of the chosen loggable entity type, whose custom attribute, XXX, has value, YYY)
      (5) entityType = any one of the loggable entity types, and entityID = a valid entity ID of the chosen loggable entity type (meaning go to a specific loggable entity)
    */
    to += "tostate" + "::" + Cred.getorg() + "::" + "entityType" + "::" + "entityID" + "::::";//Structure of a to statement
    message = Bizfloapi.updateProcessInstance(objectid, pathid, from, captions, values, to);
}
```

Figure 22

This code block decides the state transitions to certain to-states, as well as the messages that flow to these to-states. The messages are comprised of the **captions** and **values** arrays, which should have the same non-zero dimensions.

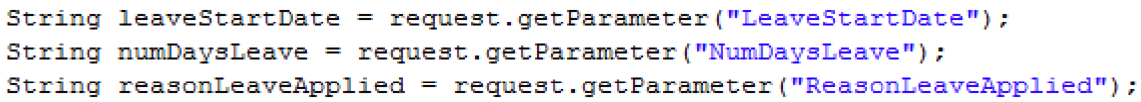
Now, replicate the template code block for two other states of LeaveApplication process, viz., SupervisorApproval and HRMApproval. We do not need to replicate it for the WaitForApproval state, since there are no state transitions from out of that state. The two additional code blocks will start with **if(jspName.equals("SupervisorApproval"))** and **if(jspName.equals(“HRMApproval"))**.

# 9 EDITING GOTOSTATES.JSP – STARTSTATE BLOCK

We first capture the data coming in from StartState.jsp. To this end we delete the statement:-

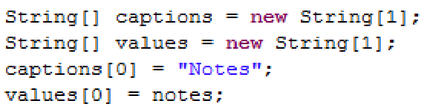
String notes = Bizfloapi.repsc(request.getParameter("Notes"));

We replace it with the following statements:-

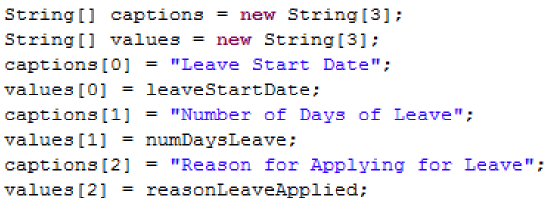


```
String leaveStartDate = request.getParameter("LeaveStartDate");
String numDaysLeave = request.getParameter("NumDaysLeave");
String reasonLeaveApplied = request.getParameter("ReasonLeaveApplied");
```

We next assemble the messages that go to the to-states. To this end, we replace code as shown below:-



```
String[] captions = new String[1];
String[] values = new String[1];
captions[0] = "Notes";
values[0] = notes;
```



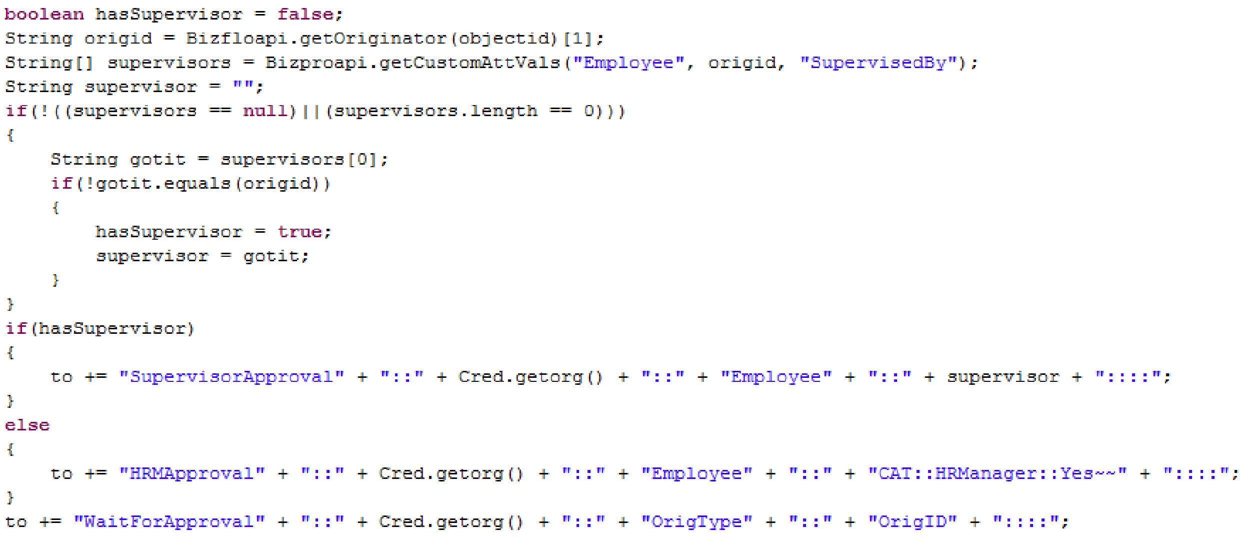
```
String[] captions = new String[3];
String[] values = new String[3];
captions[0] = "Leave Start Date";
values[0] = leaveStartDate;
captions[1] = "Number of Days of Leave";
values[1] = numDaysLeave;
captions[2] = "Reason for Applying for Leave";
values[2] = reasonLeaveApplied;
```

Figure 23

We next check if the LeaveApplicant has a Supervisor. If he has one, we send the process to the Supervisor in the SupervisorApproval state; else we send it straight to the HRManager in the HRMApproval state. In either case, we also send the process back to the LeaveApplicant in the WaitForApproval state. To this end, we delete the statement:-

to += "tostate" + "::" + Cred.getorg() + "::" + "entityType" + "::" + "entityID" + "::::";

and in its place, insert the following code:-



```
boolean hasSupervisor = false;
String origid = Bizfloapi.getOriginator(objectid)[1];
String[] supervisors = Bizproapi.getCustomAttVals("Employee", origid, "SupervisedBy");
String supervisor = "";
if(!((supervisors == null)||(supervisors.length == 0)))
{
    String gotit = supervisors[0];
    if(!gotit.equals(origid))
    {
        hasSupervisor = true;
        supervisor = gotit;
    }
}
if(hasSupervisor)
{
    to += "SupervisorApproval" + "::" + Cred.getorg() + "::" + "Employee" + "::" + supervisor + "::::";
}
else
{
    to += "HRMApproval" + "::" + Cred.getorg() + "::" + "Employee" + "::" + "CAT::HRManager::Yes~~" + "::::";
}
to += "WaitForApproval" + "::" + Cred.getorg() + "::" + "OrigType" + "::" + "OrigID" + "::::";
```

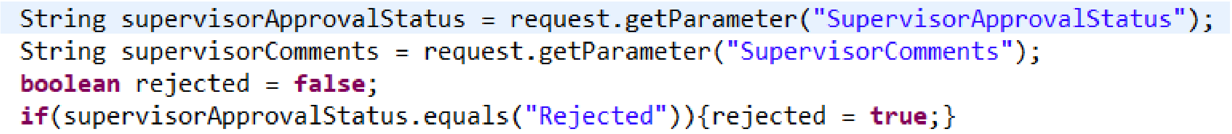
Figure 24

# 10 EDITING GOTOSTATES.JSP – SUPERVISORAPPROVAL BLOCK

We first capture the data coming in from SupervisorApproval.jsp. To this end we delete the statement:-

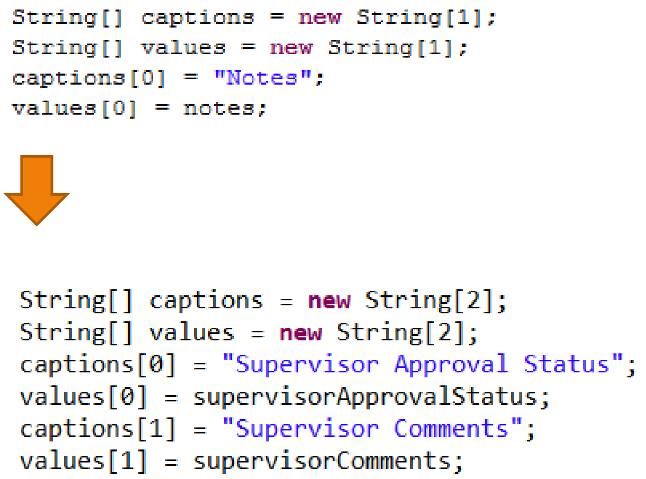
String notes = Bizfloapi.repsc(request.getParameter("Notes"));

We replace it with the following statements:-



```
String supervisorApprovalStatus = request.getParameter("SupervisorApprovalStatus");
String supervisorComments = request.getParameter("SupervisorComments");
boolean rejected = false;
if(supervisorApprovalStatus.equals("Rejected")){rejected = true;}
```

We next assemble the messages that go to the to-states. To this end, we replace code as shown below:-



```
String[] captions = new String[1];
String[] values = new String[1];
captions[0] = "Notes";
values[0] = notes;
```

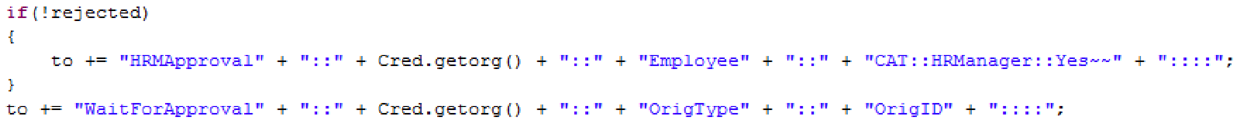
```
String[] captions = new String[2];
String[] values = new String[2];
captions[0] = "Supervisor Approval Status";
values[0] = supervisorApprovalStatus;
captions[1] = "Supervisor Comments";
values[1] = supervisorComments;
```

Figure 25

We next delete the statement:-

to += "tostate" + "::" + Cred.getorg() + "::" + "entityType" + "::" + "entityID" + "::::";

and in its place, insert the following code:-



```
if(!rejected)
{
    to += "HRMApproval" + "::" + Cred.getorg() + "::" + "Employee" + "::" + "CAT::HRManager::Yes~~" + "::::";
}
to += "WaitForApproval" + "::" + Cred.getorg() + "::" + "OrigType" + "::" + "OrigID" + "::::";
```

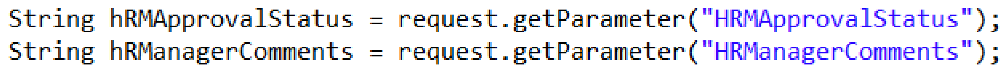
Figure 26

# 11 EDITING GOTOSTATES.JSP – HRMAPPROVAL BLOCK

We first capture the data coming in from HRMApproval.jsp. To this end we delete the statement:-

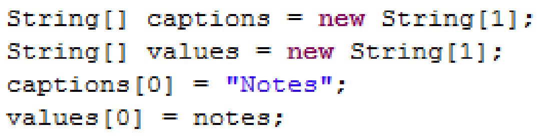
String notes = Bizfloapi.repsc(request.getParameter("Notes"));

We replace it with the following statements:-

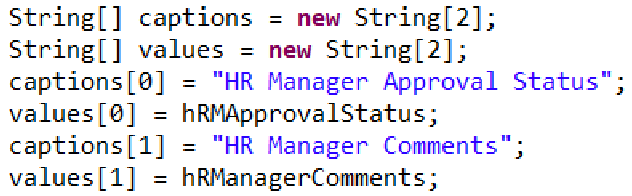


```
String hRMApprovalStatus = request.getParameter("HRMApprovalStatus");
String hRManagerComments = request.getParameter("HRManagerComments");
```

We next assemble the messages that go to the to-states. To this end, we replace code as shown below:-



```
String[] captions = new String[1];
String[] values = new String[1];
captions[0] = "Notes";
values[0] = notes;
```



```
String[] captions = new String[2];
String[] values = new String[2];
captions[0] = "HR Manager Approval Status";
values[0] = hRMApprovalStatus;
captions[1] = "HR Manager Comments";
values[1] = hRManagerComments;
```

Figure 27

We next delete the statement:-

to += "tostate" + "::" + Cred.getorg() + "::" + "entityType" + "::" + "entityID" + "::::";

and in its place, insert the following code:-

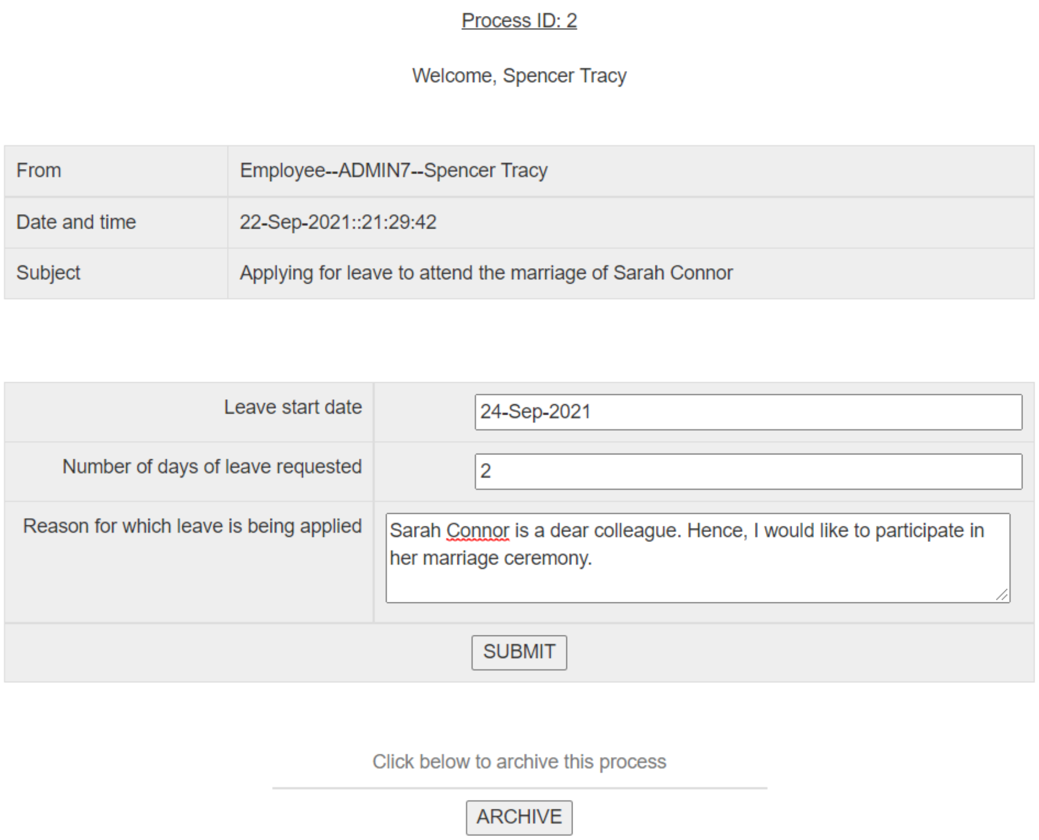
to += "WaitForApproval" + "::" + Cred.getorg() + "::" + "OrigType" + "::" + "OrigID" + "::::";

# 12 SOURCE CODE

The source code of the LeaveApplication business process will be found in the **sourcecode** folder.

# 13 SPENCER TRACY'S LEAVE APPLICATION

Go back to Figure 14. If Spencer were to click **SELECT**, he would land on StartState.jsp, which would appear as follows:-



Process ID: 2

Welcome, Spencer Tracy

| | |
|---|---|
| From | Employee--ADMIN7--Spencer Tracy |
| Date and time | 22-Sep-2021::21:29:42 |
| Subject | Applying for leave to attend the marriage of Sarah Connor |

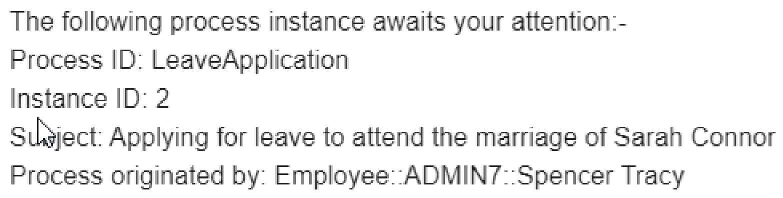
| | |
|---|---|
| Leave start date | 24-Sep-2021 |
| Number of days of leave requested | 2 |
| Reason for which leave is being applied | Sarah Connor is a dear colleague. Hence, I would like to participate in her marriage ceremony. |

Click below to archive this process

Figure 28

Notice the State Transition Form, and what data Spencer has filled. (Notice that Spencer has applied for 2 days' leave, beginning 24th September 2021).

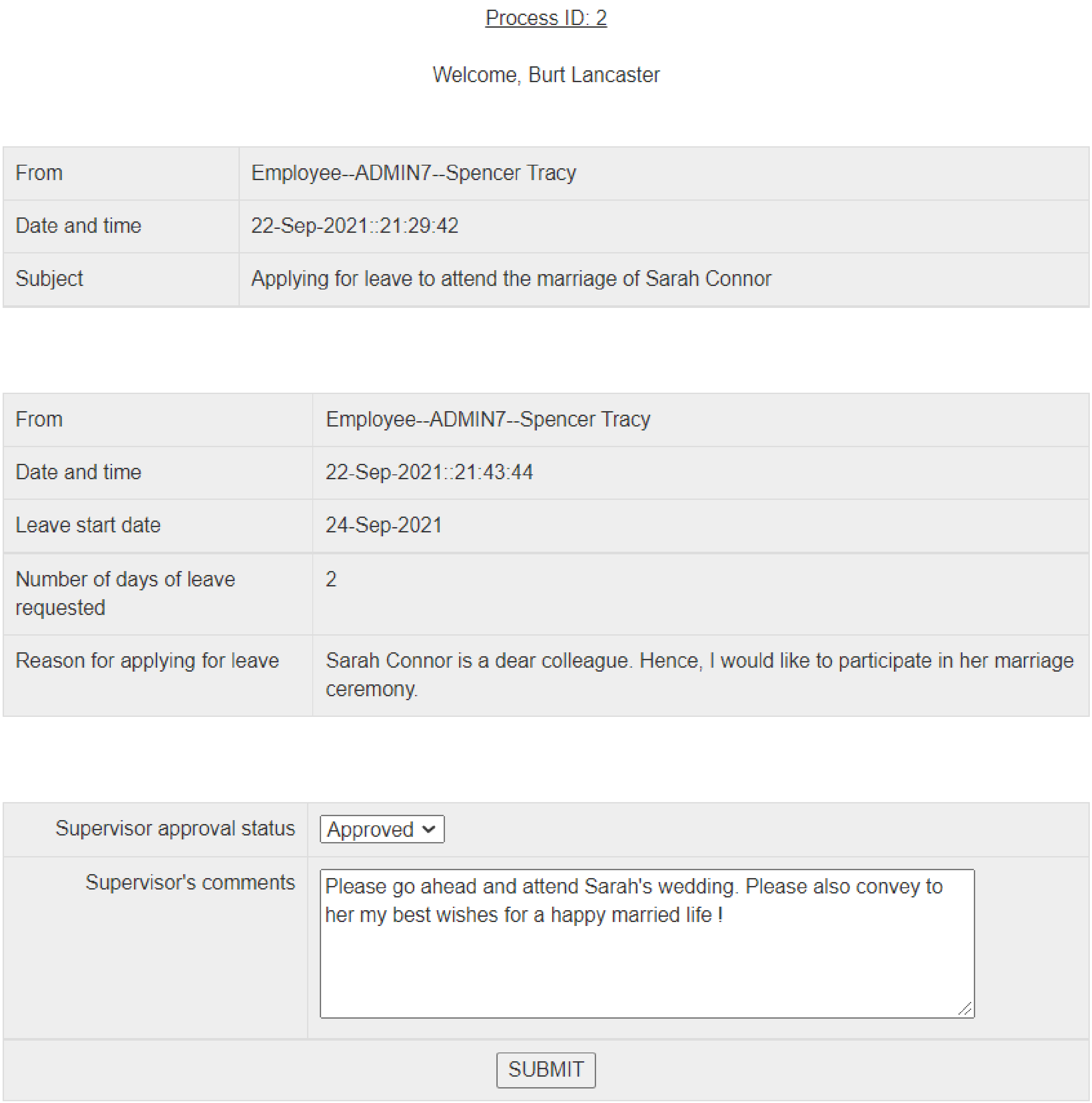
When Spencer clicks SUBMIT, the process flows to his supervisor, Burt Lancaster, who receives an email alert as follows:-



The following process instance awaits your attention:-
Process ID: LeaveApplication
Instance ID: 2
Subject: Applying for leave to attend the marriage of Sarah Connor
Process originated by: Employee::ADMIN7::Spencer Tracy

Figure 29

When Burt logs in, he finds that the above process is awaiting his attention. When he selects the process, he lands on SupervisorApproval.jsp, which presents itself to him as follows:-



Process ID: 2

Welcome, Burt Lancaster

| | |
|---|---|
| From | Employee--ADMIN7--Spencer Tracy |
| Date and time | 22-Sep-2021::21:29:42 |
| Subject | Applying for leave to attend the marriage of Sarah Connor |

| | |
|---|---|
| From | Employee--ADMIN7--Spencer Tracy |
| Date and time | 22-Sep-2021::21:43:44 |
| Leave start date | 24-Sep-2021 |
| Number of days of leave requested | 2 |
| Reason for applying for leave | Sarah Connor is a dear colleague. Hence, I would like to participate in her marriage ceremony. |

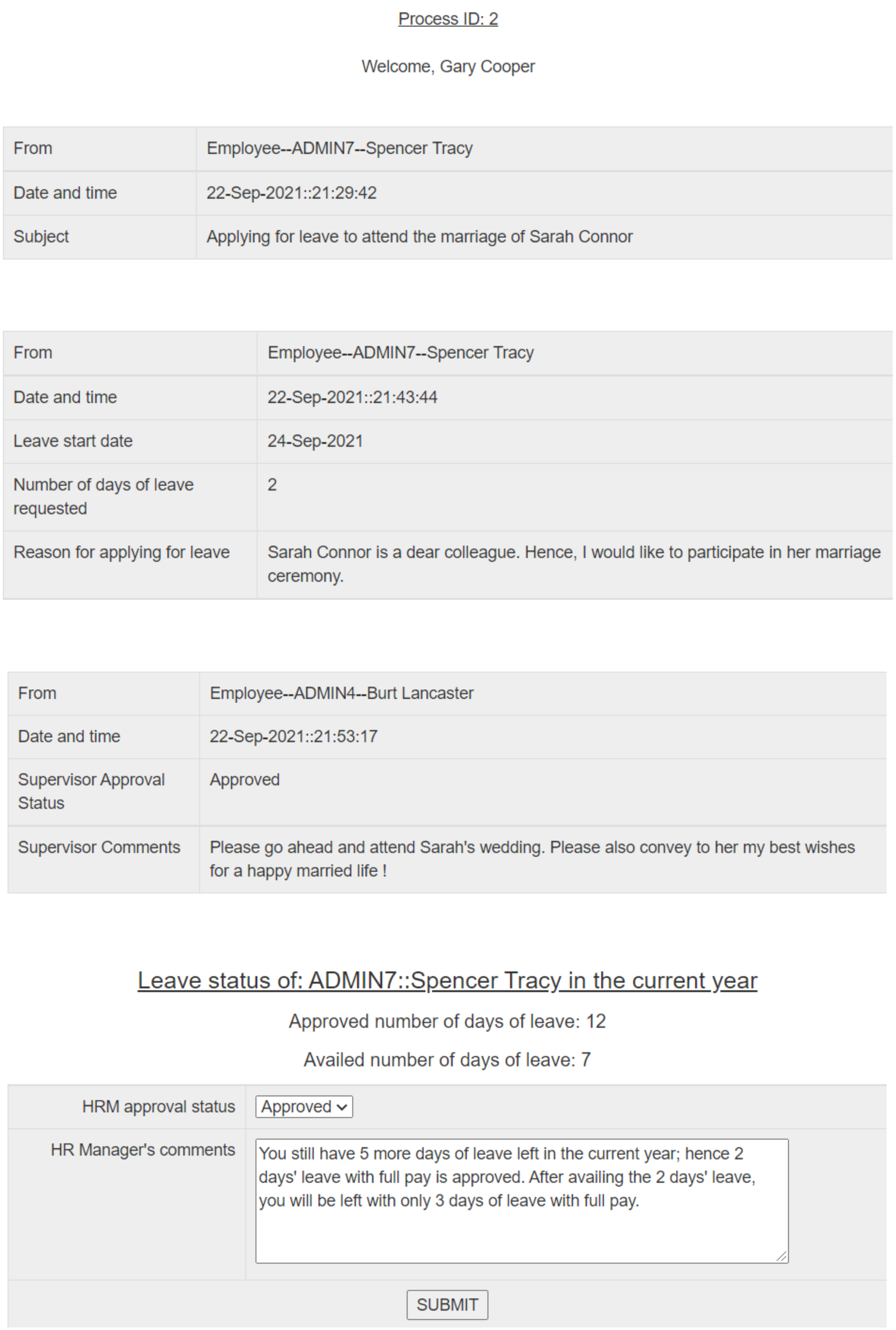
| | |
|---|---|
| Supervisor approval status | Approved |
| Supervisor's comments | Please go ahead and attend Sarah's wedding. Please also convey to her my best wishes for a happy married life ! |

Figure 30

Notice the State Transition Form, and what data Burt has filled in. (Notice that Burt has approved Spencer's leave proposal).

When Burt clicks **SUBMIT**, the process instance flows to the HR Manager, viz., Gary Cooper, as well as back to the originator of the process instance, viz., Spencer Tracy; both of them receive email alerts.

When Gary Cooper logs in, he finds that the above process is awaiting his attention. When he selects the process, he lands on HRMApproval.jsp, which presents itself to him as follows:-



Process ID: 2

Welcome, Gary Cooper

| | |
|---|---|
| From | Employee--ADMIN7--Spencer Tracy |
| Date and time | 22-Sep-2021::21:29:42 |
| Subject | Applying for leave to attend the marriage of Sarah Connor |

| | |
|---|---|
| From | Employee--ADMIN7--Spencer Tracy |
| Date and time | 22-Sep-2021::21:43:44 |
| Leave start date | 24-Sep-2021 |
| Number of days of leave requested | 2 |
| Reason for applying for leave | Sarah Connor is a dear colleague. Hence, I would like to participate in her marriage ceremony. |

| | |
|---|---|
| From | Employee--ADMIN4--Burt Lancaster |
| Date and time | 22-Sep-2021::21:53:17 |
| Supervisor Approval Status | Approved |
| Supervisor Comments | Please go ahead and attend Sarah's wedding. Please also convey to her my best wishes for a happy married life ! |

Leave status of: ADMIN7::Spencer Tracy in the current year

Approved number of days of leave: 12

Availed number of days of leave: 7

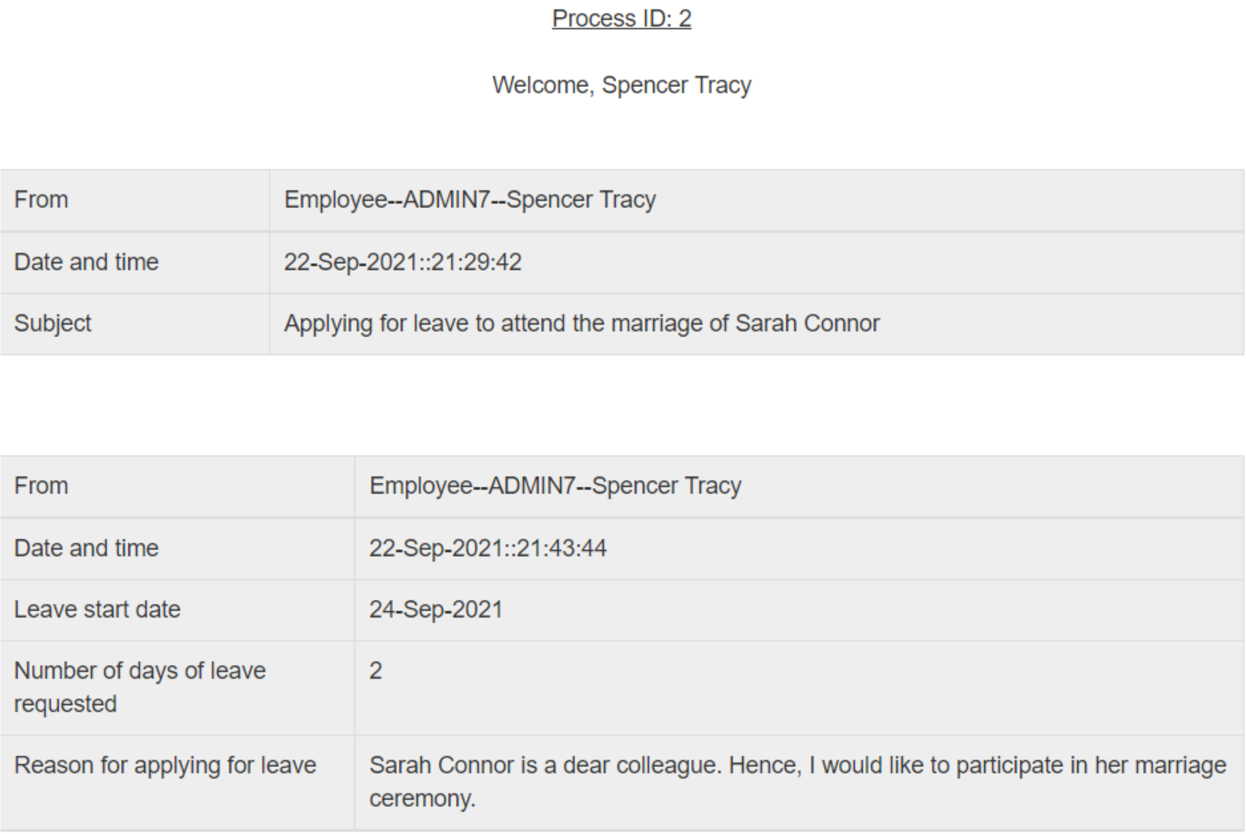
| | |
|---|---|
| HRM approval status | Approved |
| HR Manager's comments | You still have 5 more days of leave left in the current year; hence 2 days' leave with full pay is approved. After availing the 2 days' leave, you will be left with only 3 days of leave with full pay. |

Figure 31

Notice the State Transition Form, and what data Gary has filled in. (Notice that Gary too has approved Spencer's leave proposal; also notice his comments, which are based on the leave status of Spencer in the current year, *which he can see in the screen in Figure 31*).

When Gary clicks **SUBMIT**, the process instance flows back to the originator of the process instance, viz., Spencer Tracy, who receives an email alert.

When Spencer logs in, he finds that the above process instance is awaiting his attention. When he selects the process, he lands on WaitForApproval.jsp, which presents itself to him as follows:-

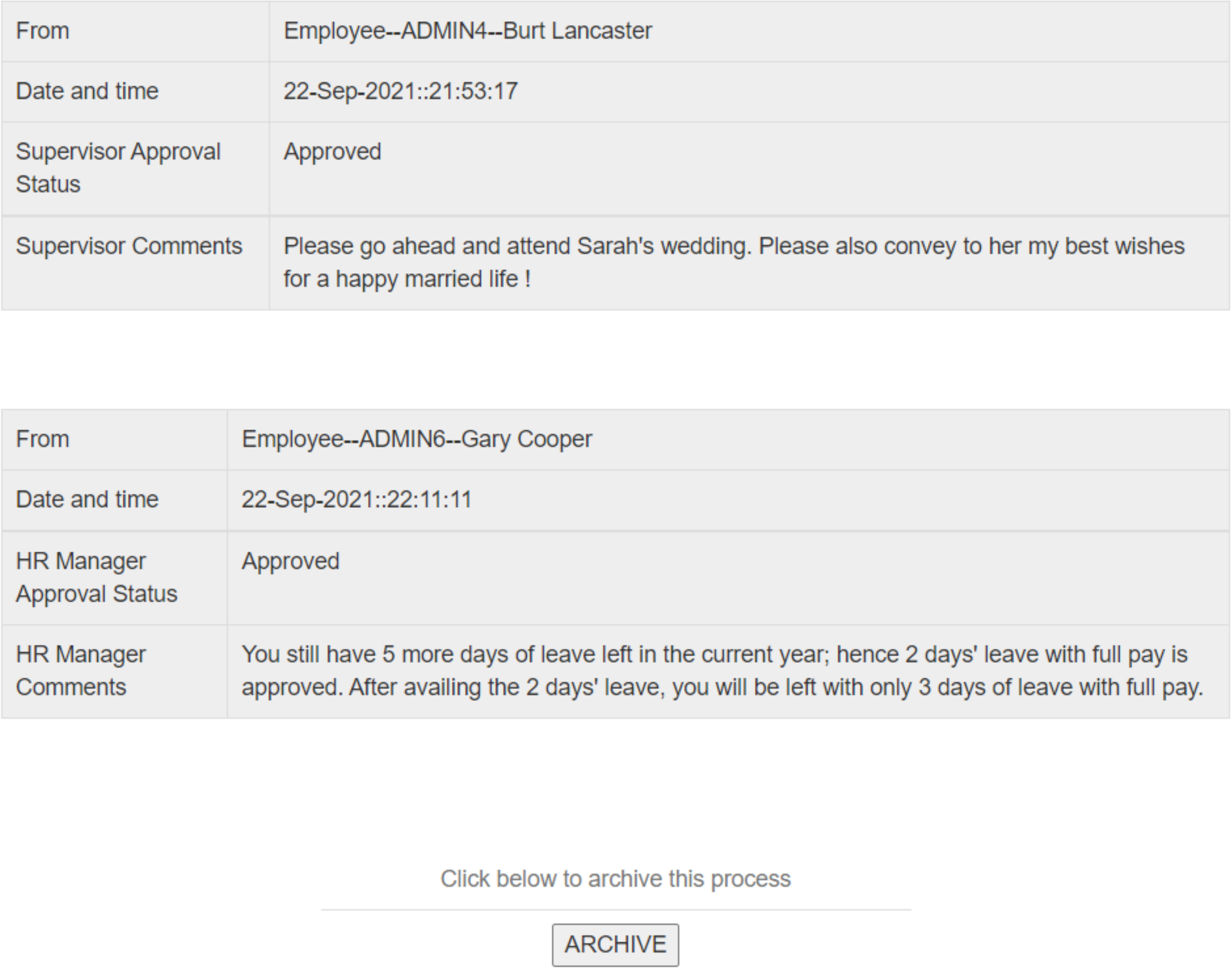


Process ID: 2

Welcome, Spencer Tracy

| | |
|---|---|
| From | Employee--ADMIN7--Spencer Tracy |
| Date and time | 22-Sep-2021::21:29:42 |
| Subject | Applying for leave to attend the marriage of Sarah Connor |

| | |
|---|---|
| From | Employee--ADMIN7--Spencer Tracy |
| Date and time | 22-Sep-2021::21:43:44 |
| Leave start date | 24-Sep-2021 |
| Number of days of leave requested | 2 |
| Reason for applying for leave | Sarah Connor is a dear colleague. Hence, I would like to participate in her marriage ceremony. |



| | |
|---|---|
| From | Employee--ADMIN4--Burt Lancaster |
| Date and time | 22-Sep-2021::21:53:17 |
| Supervisor Approval Status | Approved |
| Supervisor Comments | Please go ahead and attend Sarah's wedding. Please also convey to her my best wishes for a happy married life ! |

| | |
|---|---|
| From | Employee--ADMIN6--Gary Cooper |
| Date and time | 22-Sep-2021::22:11:11 |
| HR Manager Approval Status | Approved |
| HR Manager Comments | You still have 5 more days of leave left in the current year; hence 2 days' leave with full pay is approved. After availing the 2 days' leave, you will be left with only 3 days of leave with full pay. |

Click below to archive this process

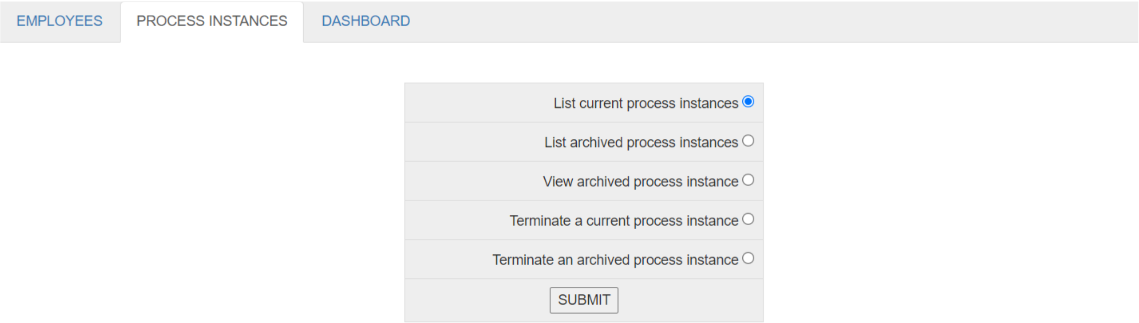
Figure 32

Spencer sees that his leave has been approved; accordingly he is happy. From the HR Manager's comments, he learns that after availing the approved leave, he will still be left with 3 days' leave with full pay during the current year.

Spencer now clicks **ARCHIVE**, so that the process instance goes into archive.

## 14 ARCHIVED PROCESS INSTANCES

Archived process instances can be accessed by the Administrator under the **Admin Main Menu PROCESS INSTANCES** tab:-



List current process instances ●

List archived process instances ○

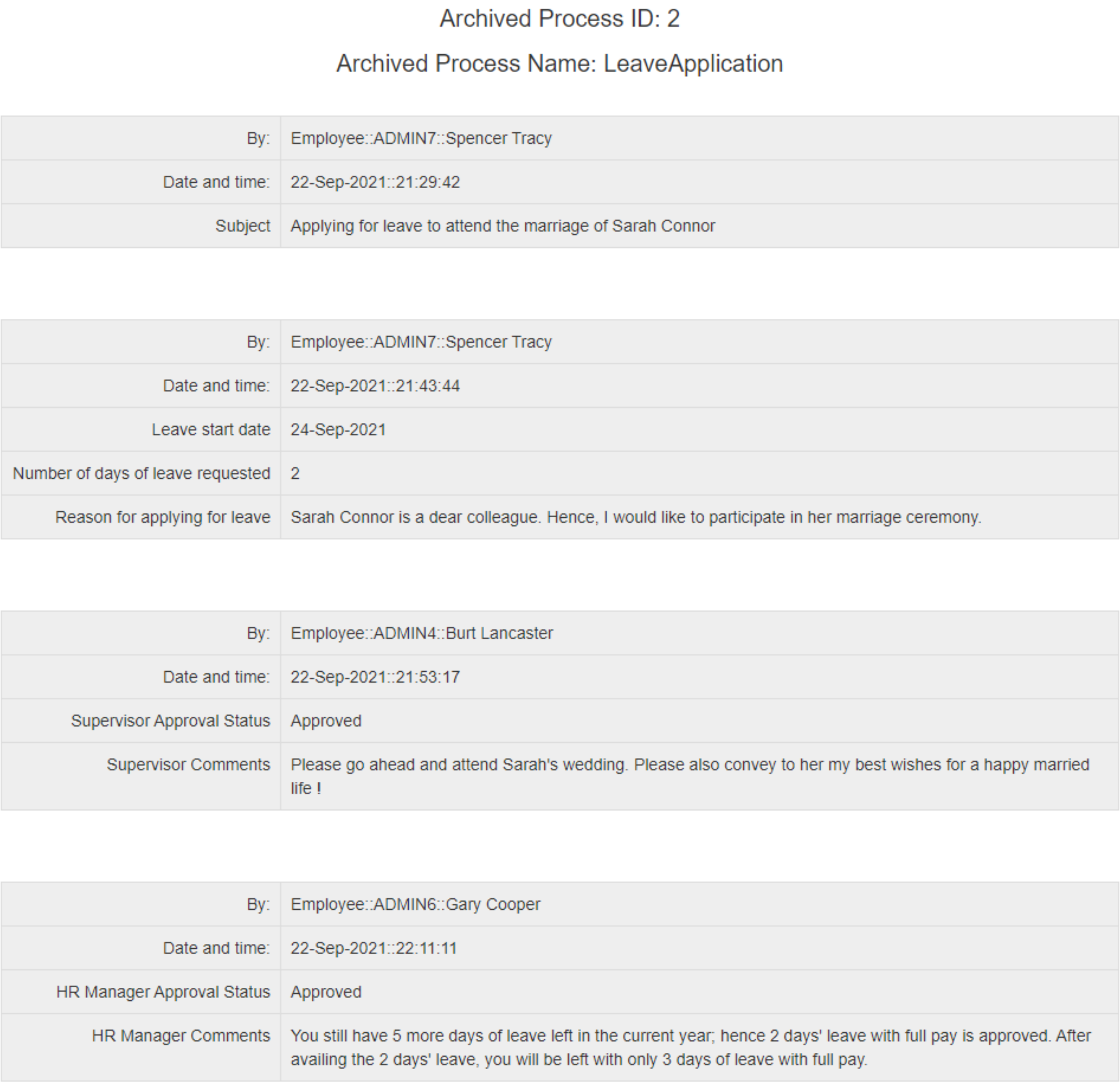
View archived process instance ○

Terminate a current process instance ○

Terminate an archived process instance ○

Figure 33

The process instance archived by Spencer would look to the Administrator like this:-



Archived Process ID: 2

Archived Process Name: LeaveApplication

| | |
|---|---|
| By: | Employee::ADMIN7::Spencer Tracy |
| Date and time: | 22-Sep-2021::21:29:42 |
| Subject | Applying for leave to attend the marriage of Sarah Connor |

| | |
|---|---|
| By: | Employee::ADMIN7::Spencer Tracy |
| Date and time: | 22-Sep-2021::21:43:44 |
| Leave start date | 24-Sep-2021 |
| Number of days of leave requested | 2 |
| Reason for applying for leave | Sarah Connor is a dear colleague. Hence, I would like to participate in her marriage ceremony. |

| | |
|---|---|
| By: | Employee::ADMIN4::Burt Lancaster |
| Date and time: | 22-Sep-2021::21:53:17 |
| Supervisor Approval Status | Approved |
| Supervisor Comments | Please go ahead and attend Sarah's wedding. Please also convey to her my best wishes for a happy married life ! |

| | |
|---|---|
| By: | Employee::ADMIN6::Gary Cooper |
| Date and time: | 22-Sep-2021::22:11:11 |
| HR Manager Approval Status | Approved |
| HR Manager Comments | You still have 5 more days of leave left in the current year; hence 2 days' leave with full pay is approved. After availing the 2 days' leave, you will be left with only 3 days of leave with full pay. |

Figure 34