Exercise 1

Create a new project called *VacationLeaveRequest*. In this project create a new process called *Vacation Leave Request*, 1 lane corresponding to the actor *Employee*, 1 milestone *Register*, and 1 activity *Register Vacation Request* connected to a start and an end event.

Exercise 2

In the data model, in the main (Master) entity *VacationLeaveRequest* specify the attributes (right click on the entity and select option *edit attributes*):

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Date</td>
<td>Date - Time</td>
</tr>
<tr>
<td>Start Date</td>
<td>Date - Time</td>
</tr>
<tr>
<td>End Date</td>
<td>Date - Time</td>
</tr>
</tbody>
</table>
Exercise 3

Attach the following form to activity *Register Vacation Request* (using Controls -> Containers -> Group and Layout -> 50%-50%):
Specifying the following policies:

<table>
<thead>
<tr>
<th></th>
<th>Visible</th>
<th>Editable</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Date</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Applicant</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Start Date</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>End Date</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Number of Days</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

(display attribute -> fullname)

Run the application.

Exercise 4

RequestDate must be automatically specified by the system through the “On Enter” expression setDate:

```<VacationLeaveRequest.RequestDate>=DateTime.Today```

attached to activity Register Vacation Request.

Exercise 5

Applicant must be automatically specified by the system through the “On Enter” expression setApplicant:

```<VacationLeaveRequest.Applicant>=Me.Case.Creator.Id```

attached to activity Register Vacation Request.
Exercise 6

**NumberOfDays** must be automatically specified by the system through the “On Save” expression `setNumDays` attached to activity **Register Vacation Request** (Note: You must specify the additional variables used **EffectiveDurationMin** and **EffectiveDurationDay** in the list of variables):

```java
EffectiveDurationMin = CHelper.getEffectiveDuration(Me, <VacationLeaveRequest.StartDate>, <VacationLeaveRequest.EndDate>);

EffectiveDurationDay = ((EffectiveDurationMin / 60) / 8);
if (EffectiveDurationDay < 1){
   CHelper.ThrowValidationError(" You must request at least one vacation day");
} else{
   <VacationLeaveRequest.NumberofDays> = (EffectiveDurationMin / 60) / 8;
}
```

Run the application.

Exercise 7

**NumberOfDays** must be automatically specified by the system when Start Date or End Date change values.

Run the application.

Exercise 8

Create an organization hierarchy  Director -> Manager -> Developer
and specify the following performers policies on activity Register Vacation Request:

Create 5 users: 1 Director, 2 Managers, 1 Developer and 1 Human Resources Employee

Run the application.

Exercise 9

Add activity Specify Project. The project can be selected from a list of projects in the form. Update process model and data model and specify a form and the possible performers for this activity.
Run the application.

Exercise 10

Add a new activity Approve Leave Request to the model

Add to the data model a boolean attribute Approved to VacationLeaveRequest to specify if the request is approved or not.
Define the form for Approve Leave Request:

Assign Approve Leave Request to every manager
Run the application and check that activity *Approve Leave Request* is not assigned to all managers.

**Exercise 11**

*Assign Approve Leave Request* to a manager different from the one executing *Specify Project*. To do this add a WFUser attribute *ProjectManager* to *VerificationLeaveRequest*. 
Define an “On Exit” action on activity Specify Project
The performer of *Specify Project* should be different from *ProjectManager*

Run the application and verify that activities *Specify Project* and *Approve Leave Request* are executed by two different managers.
Exercise 12

Add activities **Notify Acceptance** and **Notify Rejection**:

Define the corresponding forms:

![Notify Acceptance Form]

![Notify Rejection Form]

and assign their performer to every Human Resources Employee.
Define the business rules connected to the decision point.

Run the application.

Exercise 13
Export your project to a .bex file