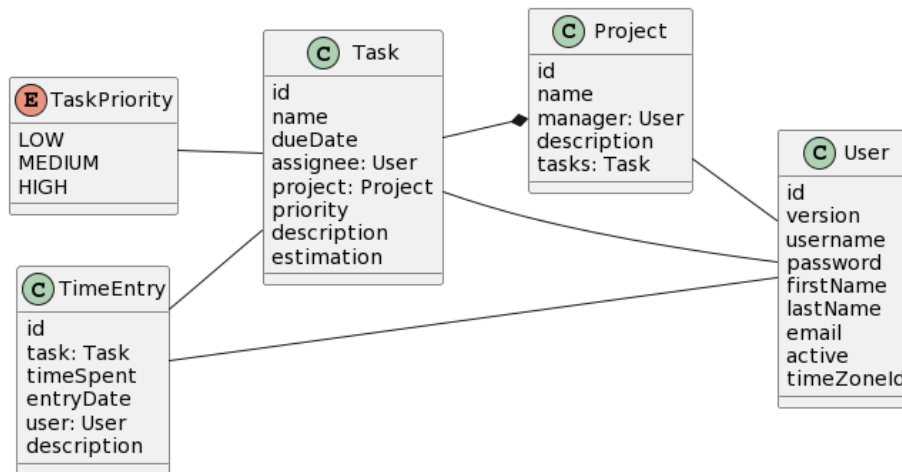




Assignment 2

Create your app's user interface

For this assignment, you'll be adding views for the entities in the data model: **Project**, **Task**, and **TimeEntry**.



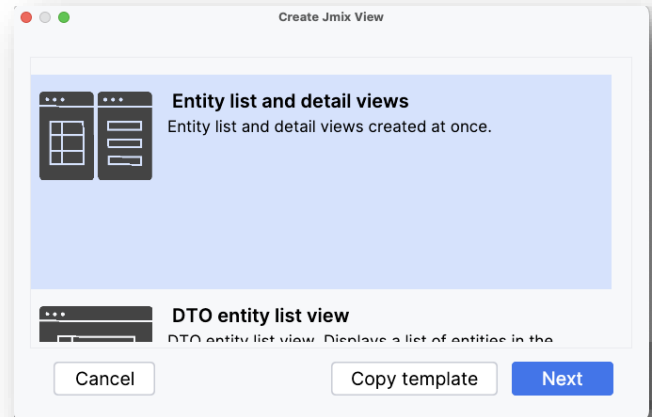
One of these views will also have an event handler.

Step 1: Create views for the Project entity

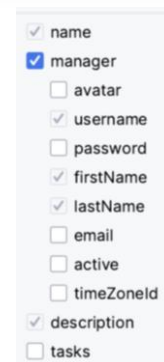
The views for **Project** entity have been created in the course videos. If your project already have both **Project.list** and **Project.detail**, proceed to the next step.

To create views for the **Project** entity:

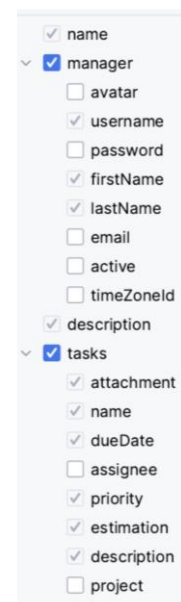
1. Select the **Entity list and detail views** template and follow the steps up to fetch plan definition.



2. Specify the fetch plan for the list view



3. Define the following fetch plan for the detail view

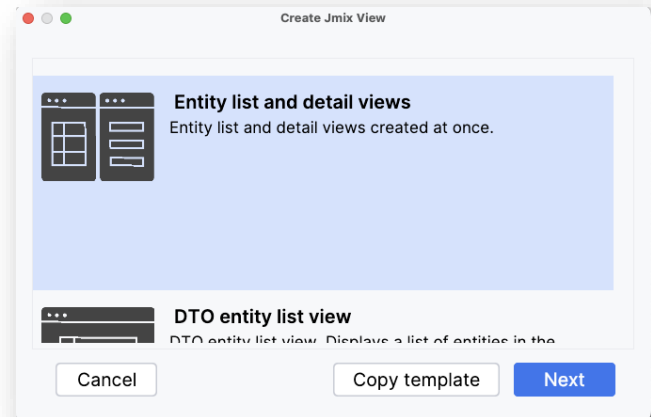


Step 2: Create views for the Task entity

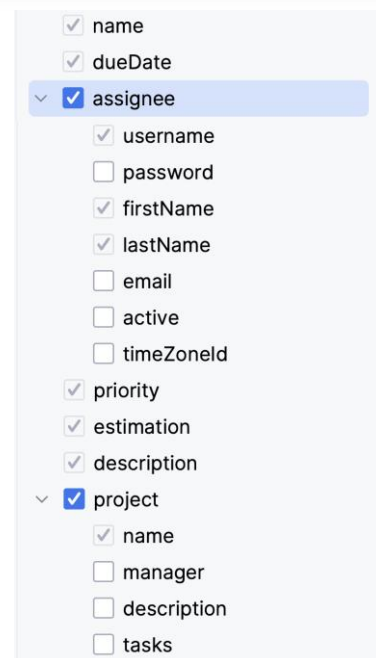
The **Task.detail** view has been created in one of the course videos. In that video we created it from scratch. This time, we will create both **Task.list** and **Task.detail** using Studio's template.

To create views for the **Task** entity:

1. Select the **Entity list and detail views** template and follow the steps up to fetch plan definition.



2. Specify the fetch plan for the list view

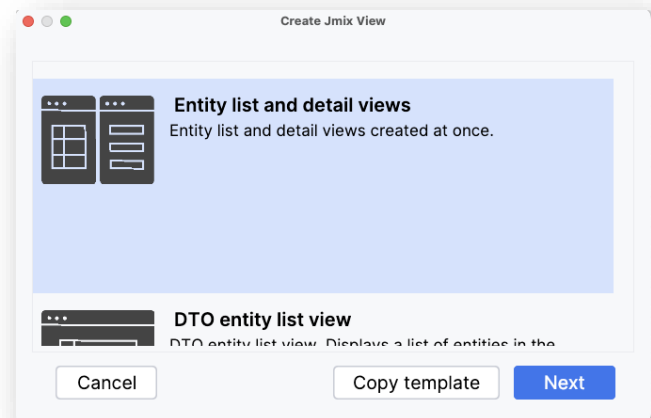


3. Use the same fetch plan for the detail view.

Step 3: Create views for the TimeEntry entity

To create views for the **TimeEntry** entity:

1. Select the **Entity list and detail** views template and follow the steps up to fetch plan definition.



2. Specify the fetch plan for the list view

- ☒ task
 - ☒ name
 - ☐ dueDate
 - ☐ assignee
 - ☐ priority
 - ☐ estimation
 - ☐ description
 - ☐ project
 - ☒ timeSpent
 - ☒ entryDate
- ☒ user
 - ☒ username
 - ☐ password
 - ☒ firstName
 - ☒ lastName
 - ☐ email
 - ☐ active
 - ☐ timeZoneId
 - ☒ description

3. Use the same fetch plan for the detail view

Step 4: Create a handler

To speed up the completion of forms, some fields can be filled in automatically. For example, when creating a new **TimeEntry** instance, you can automatically initialize its `entryDate` attribute with the the current date.

To do this, follow these steps:

1. Navigate to the `TimeEntry.detail` view's controller.
2. Add an `InitEntityEvent` event handler.
3. Add the logic to set the current date as `entryDate`'s value.

Tip: Use tools in Studio to generate the handler and a code snippet to get the current date and time.



Generate Handler



Code Snippets