## 🔑 Models (with polymorphic support)

Include these models in `app/Models`:

1. ProcessFlow.php

- Tracks definition of a flow and the model it is linked to.  
- Uses morphTo() for `model()` and hasMany for steps and instances.

2. ProcessStep.php

- Belongs to ProcessFlow. Has many actions and conditions.

3. ProcessCondition.php

- Belongs to ProcessStep. Has type and parameter fields.

4. ProcessAction.php

- Belongs to ProcessStep. Similar to condition.

5. ProcessInstance.php

- Tracks current step and completion. Morph to `model()` and belongs to flow and step.

6. ProcessLog.php

- Logs step completions with optional notes and metadata.

## 🧠 Services (in app/Services/ProcessFlow)

1. ProcessConditionService.php – Evaluates all conditions for a step.

2. ProcessActionService.php – Executes configured actions.

3. ProcessEngineService.php – Orchestrates advancing a step (condition check → action → update step).

## 🧭 Controller

ProcessFlowController.php – Handles advancing a process instance through a secured POST route.

## 🛡️ Middleware

Uses dynamic route-based permission check middleware to protect routes, e.g. `process.advance`.

## 📌 Route

POST /process/advance/{instance}  
→ ProcessFlowController@advance  
→ Route name: 'process.advance'  
→ Middleware: permission (Spatie + route name)

## 🎛️ Power User Interface (for Admins and Planners)

Role-based UI for `admin` and `planning` users to:  
- Create new process flows  
- Add and order steps  
- Attach conditions and actions to each step  
- View flow-to-model linkages

## 🧩 Example Blade UI Snippet

<!-- Create/Edit Flow -->  
<form method="POST" action="{{ route('process\_flows.store') }}">  
 @csrf  
 <input type="text" name="name" placeholder="Flow Name" required>  
 <textarea name="description"></textarea>  
 <select name="model\_type">  
 <option value="App\Models\Activity">Activity</option>  
 <option value="App\Models\PsipName">PsipName</option>  
 </select>  
 <input type="number" name="model\_id" placeholder="Model ID" required>  
 <button type="submit">Save</button>  
</form>  
  
<!-- Steps Table -->  
@foreach($flow->steps as $step)  
 <tr>  
 <td>{{ $step->order }}</td>  
 <td>{{ $step->name }}</td>  
 <td><a href="{{ route('steps.edit', $step) }}">Edit</a></td>  
 </tr>  
@endforeach

The full UI can be built using Laravel Blade + JS modals with bootstrap 5 for dynamic linking of actions and conditions.