BPMN(Business Process Modeling Notation) :

Coordinate or orchestrate the behavior of people, system, information, and things to produce business outcomes. Processes are typically structured and repeatable.

Basic POC to Understand the functionality of BPMN 2.0

1. Add Flowable Dependency

<dependency>

<groupId>org.flowable</groupId>

<artifactId>flowable-engine</artifactId>

<version>6.6.0</version>

</dependency>

1. write bpmn20.xml file

circle – start event. It’s the starting point of a process instance.

rectangle – user task. User has to perform.

diamond shape with cross – exclusive gateway, based on the user’s decision will route the process instance to either the approval or the reject path.

1. Instantiate the instance of ProcessEngine via Programatically Using ProcessEngineConfiguration.
2. Provide the process variable to perform the process, in this case we take variables using scanner class. But in reality we will get these variables using form submission of via rest call/ api call.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Pools and Lanes:**

Pools:

1. Each Pool can contain a maximum of one process.
2. Pools represent: Companies, Customer or departments

Lanes:  Each lane represents one parallel of the process and shows both the exact tasks and responsibilities each department has. And the interplay between them.

**Understanding basic notations:**

1. Rectangle: Rectangle depicts Activity/Task is work that is performed with in business process.
2. Circle: Event is something that happens during course of Process.
3. Diamond: Gateway is used to control divergence and convergence of Sequence Flows in a Process.
4. Arrow: Two major flow elements are core to BPM:
5. A Sequence Flow is used to show the order that Activities will be performed in a process.

A Message Flow is used to show the flow of messages between two participants of a Process. Message Flow is used when different departments and organizations send information between each other.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Steps to create Process Mapping Path:**

1. Always map happy path first.
2. Create Pool and lanes as required in process.

Video steps:

1. Install Java, Maven, Tomcat, and Flowable.
2. Flowable with green icon is open source, with red icon enterprise version

Enterprise comes with inbuild Tomcat, whereas Open source Flowable comes without Tomcat we need to externally setup Tomcat for Open source Flowable .

Three Core Offerings of Flowable:

1. CMMN(Case Management Modelling Notation): we know about the result/outcome but not sure about the path.
2. BPMN(Business Process Modeling Notation): here we are sure about the path and outcome as well.
3. DMN(Decision Modeling Notation): Decision making a inbuild table which is used to store decision data(metadata) based on that in a process decisions happened.

Flowable has following as we can say components:

1. Flowable Engage
2. Flowable Design
3. Flowable Control

Dynamic Injection:

As we create any process in that if task is there that remains same (task we can say any manual interaction like approved or reject) till end. But here flowable provides the feature that we can replace that task with any sub-process at run time.

If we think any additional check is required then we go for it.

Reporting tools:

Flowable provides the data visualization feature, in form of bars graphs etc. only Flowable provide these features not others BPMNs.

Events in Flowable:

Event listeners are supported by flowable as inbuild feature, there is no need to integrate manually.

In Open Source Flowable

1. Add/Manage user -> IDM App
2. Process (BPMN) creation 🡪 Modeler App

**Flowable Integration:**

* 1. Add spring boot starter dependency “flowable-spring-boot-starter-basic”.
  2. Create folders under resource to keep specific files.

processes, cases, dmn, forms folders

* 1. Create bpmn file.
  2. For rest flavor need to add “flowable-spring-boot-starter-rest” dependency.

**Flowable Database Schema:**

The database table names associated with the Flowable Opensource code base start with **ACT\_TBL\_NAME**

Specific to Flowable work or Engage start with the **FLW\_prefix**

**The Second part of the table after first “\_” is a two character identification specifying the specific use case of the table.**

**GE\_:** General Data**,** which is used for various use cases.

**RU\_:** Runtime tables, contain the runtime data of processes, cases etc. That are not yet finished. Flowable only stores the runtime data during execution and removes the records after the instances ends. This keeps the runtime tables small and fast.

**HI\_:** Historical table, These tables contains the historical data when the data is removed from the runtime tables then. These tables contains all the information for these finished instances.

**RE\_:** Repository table, contains static information around models and definitions.