

## Assignment – 1

Develop a Spring Boot application that uses an open-source engine (e.g. Camunda) to realize the following workflow

- 1) User X creates a purchase order and submit to user R for review
- 2) User R reviews the purchase order, either accept or reject
- 3) User A receives the purchase order on acceptance by user R and either accept or reject
- 4) User X updates the purchase order if user B or User A rejects

Endpoint	/list (GET with header containing role and username)
Precondition	User with role=PUBLISHER/REVIEWER/APPROVER
Normal flow	For role=PUBLISHER, list all the purchase orders with status=created or rework For role=REVIEWER, list all the purchase orders with status=created or reworked For role= APPROVER, list all the purchase orders with status=reviewed
Error handling	validation error (412), unauthorized owner (401) to client
Request Payload	None
Response Payload	200 with Array of Purchase orders or error

Endpoint	/txn (POST/PATCH with header containing role and username)
Precondition	User with role=PUBLISHER/REVIEWER/APPROVER
Normal flow	For role=PUBLISHER, create or update the purchase order For role=REVIEWER, reject or accept the given purchase order For role= APPROVER, , reject or accept the given purchase order
Error handling	validation error (412), unauthorized owner (401) to client
Request Payload	Purchase order data <pre>{   "recordId" : "03457:123456"   "taskId" : "replace pump"   "partsPrice": 150   "labourPrice": 50   "amount": 200 }</pre> OR <pre>{   "recordId" : "03457:123456"   "action" : accept or reject   "reviewSummary" : [{}] }</pre>
Response Payload	200 with updated purchase order

### Purchase order sample data model

```
//created purchaseOrder
{
  "recordId" : "03457:123456"
```

```
"taskId" : "replace pump"
"partsPrice": 150
"labourPrice": 50
"amount": 200
"status" : "created", → other values (rejected, accepted
"reviewSummary" : [] -> contains collection of review comments
"createdBy" : "user X"
"updatedBy" : ""
}
```

```
//accepted purchaseOrder by reviewer
{
  "recordId" : "03457:123456"
  "taskId" : "replace pump"
  "partsPrice": 150
  "labourPrice": 50
  "amount": 200
  "status" : "reviewed",
  "reviewSummary" : [{"Initial version"}]
  "createdBy" : "user X"
  "updatedBy" : "user R"
}
```

```
//rejected purchaseOrder by reviewer
{
  "recordId" : "03457:123456"
  "taskId" : "replace pump"
  "partsPrice": 150
  "labourPrice": 50
  "amount": 200
  "status" : "rework",
  "reviewSummary" : [{"Update the partsPrice to 175"}]
  "createdBy" : "user X"
  "updatedBy" : "user R"
}
```

```
//accepted purchaseOrder by approver
{
  "recordId" : "03457:123456"
  "taskId" : "replace pump"
  "partsPrice": 150
  "labourPrice": 50
  "amount": 200
  "status" : "approved",
  "reviewSummary" : [{"Initial version"}, {"Verified the cost"}]
  "createdBy" : "user X"
  "updatedBy" : "user A"
}
```

```
//rejected purchaseOrder by approver
```

```
{  
  "recordId" : "03457:123456"  
  "taskId" : "replace pump"  
  "partsPrice": 150  
  "labourPrice": 50  
  "amount": 200  
  "status" : "rework",  
  "reviewSummary" : [{"Initial version"}, {"Revise the labourPrice to 75"}]  
  "createdBy" : "user X"  
  "updatedBy" : "user A"  
}
```

**Also provide Junit tests**