**BÁO CÁO 15: TÌM HIỂU CAMUNDA**

1. **Sơ lược về Camunda**

* Camunda là một nền tảng được phát triển bằng Java, mục đích xử lý tiến trình tự động cho các mô hình và ký hiệu quy trình nghiệp vụ (BPMN 2.0).
* Một mô hình BPMN thường sẽ bao gồm:

****

* + Sự kiện bắt đầu và sự kiện kết thúc

**A picture containing icon

Description automatically generatedA picture containing shape

Description automatically generated**

* + Các cấu trúc rẽ nhánh

**Shape, rectangle

Description automatically generatedShape, rectangle

Description automatically generated**

* + Service task và Usertask
* **Graphical user interface, text, application

  Description automatically generated**Ở trong các cấu trúc rẽ nhánh, ta có thể thêm vào các điều kiện bằng cách sử dụng các biến ví dụ:

1. **Tạo một process trong modeler**

* Để tạo được một definition, ta sử dụng phần mềm modeler và thực hiện vẽ mô hình.

Diagram

Description automatically generated

* Sau đó ta sẽ deploy mô hình này nên localhost
* Để tiến hành deploy definition, start các process instance cũng như complete chúng. Ta sẽ gọi các api đã được cài đặt sẵn.

1. **Apply sơ đồ BPMN vào React**

* Đầu tiên, ta lấy string xml trong modeler bằng cách gọi API có sẵn

axios

      .get(

        "http://localhost:8080/engine-rest/process-definition/key/" +

          id +

          "/xml"

      )

      .then((res) => {

        setXml(res.data);

      });

* Sau đó, import thư viện BpmnViewer để render xml

import BpmnViewer from "bpmn-js/dist/bpmn-navigated-viewer.production.min.js";

this.viewer = new BpmnViewer({

      container,

      height: 500,

    });

* Để bắt sự kiện khi nhấn vào một component trong mô hình ta sử dụng eventBus:

const eventBus = this.viewer.get("eventBus");

    eventBus.on("element.click", (e) => {

      console.log(e);

      if (e.element.type === "bpmn:CallActivity") {

        this.props.setDiagram(e.element.businessObject.calledElement);

      } else {

        axios

          .post(

            "http://localhost:8080/engine-rest/process-instance",

            JSON.stringify({ activityIdIn: [e.element.id] }),

            { headers: { "Content-Type": "application/json" } }

          )

          .then((res) => {

            this.props.setListProcessInstances(res.data);

          });

      }

    });

* Để có thể phóng to thu nhỏ mô hình, sử dụng canvas của Viewer, sau đó sử dụng hàm để phóng to thu nhỏ theo ý muốn

const canvas = Viewer.get("canvas");

canvas.zoom("fit-viewport", "auto");

zoomInOut = (number, viewer) => {

    viewer.get("zoomScroll").stepZoom(number);

    return null;

  };

1. **Tạo form và render form trên react.**

* Khi