

```

import {
  Button,
  Col,
  DatePicker,
  Form,
  Input,
  InputNumber,
  Row,
  Select,
  Spin,
  notification,
} from "antd";
import { isEmpty } from "lodash";
import React, { useEffect, useMemo, useRef, useState } from "react";
import { useDispatch, useSelector } from "react-redux";
import { PrinterOutlined } from "@ant-design/icons";
import SlipModalComponent from "components/SlipModalComponent";
import {
  GetDonHangKhamSuckhoe,
  GetPhuongAnKinhDoanh,
  UpdateDonHangKhamSuckhoe,
  UpsertDonHangKhamSuckhoe,
} from "store/slices/khamSuckhoeDoanSlice";
import { dateFormatList } from "constants";
import moment from "moment";
import ChiTietPhuongAn from "../ChiTietPhuongAn";
import NhanSu from "../NhanSu";
import VatTu from "../VatTu";
import DocumentViewer from "constants/DocumentViewer";
import { DOCUMENT_TYPE_ID } from "constants";

const formater = new Intl.NumberFormat("vi-VN");

// formatter and parser input number
export const formatterNumber = (val) => {
  if (!val) return 0;
  return formater.format(val);
};

export const parserNumber = (val) => {
  if (!val) return 0;
  return Number.parseFloat(
    // eslint-disable-next-line no-useless-escape
    val.replace(/\$\$s?|(\.+)/g, "").replace(/(\.){1}/g, ".")
  ).toFixed(2);
};

export default function ModalDonHang({
  visibleModal,
  setVisibleModal,
  loading,
  reloadData,
  donHangId,
  setDonHangId,
  permitList = [],
}) {
  const dispatch = useDispatch();
  const [form] = Form.useForm();

  const {
    DonHangKhamSuckhoeDetail,
    HangMucKhamList,
    MauKhamList,
    DoanKhamList,
    PhuongAnKinhDoanhDetail,
    PhuongAnKinhDoanhList,
  } = useSelector((state) => state.ksk);

  const [phuongAnDetailOld, setPhuongAnDetailOld] = useState([]);
  const [isPrinter, setIsPrinter] = useState(false);
  const [printData, setPrintData] = useState({});
  const childRef = useRef(null);

```

```

// eslint-disable-next-line react-hooks/exhaustive-deps
const onChangePhuongAn = (e) => {
  if (e === undefined || e === null) return;
  dispatch(GetPhuongAnKinhDoanh(e));
};

useEffect(() => {
  form.resetFields();
  if (isEmpty(donHangId)) return;
  dispatch(GetDonHangKhamSucKhoe(donHangId));
}, [dispatch, donHangId, form]);

useEffect(() => {
  if (isEmpty(donHangId)) return;
  if (DonHangKhamSucKhoeDetail === phuongAnDetailOld) return;
  setPhuongAnDetailOld(DonHangKhamSucKhoeDetail);

  form.setFieldsValue(DonHangKhamSucKhoeDetail);
  onChangePhuongAn(DonHangKhamSucKhoeDetail?.phuongAnId);

  const ngayKhamThucTe = !isEmpty(DonHangKhamSucKhoeDetail?.ngayKhamThucTe)
    ? moment(DonHangKhamSucKhoeDetail?.ngayKhamThucTe)
    : null;

  form.setFieldValue("ngayKhamThucTe", ngayKhamThucTe);
  const ngayKhamThucTeTu = !isEmpty(
    DonHangKhamSucKhoeDetail?.ngayKhamThucTeTu
  )
    ? moment(DonHangKhamSucKhoeDetail?.ngayKhamThucTeTu)
    : null;

  form.setFieldValue("ngayKhamThucTeTu", ngayKhamThucTeTu);
  // eslint-disable-next-line no-use-before-define
}, [
  dispatch,
  donHangId,
  DonHangKhamSucKhoeDetail,
  form,
  onChangePhuongAn,
  phuongAnDetailOld,
]);

useEffect(() => {
  // Lay ten don hang mac dinh theo phuong an kinh doanh
  if (isEmpty(donHangId) && PhuongAnKinhDoanhDetail && PhuongAnKinhDoanhDetail?.tenPhuongAn) {
    form.setFieldValue("tenDonHang", `Đơn hàng ${PhuongAnKinhDoanhDetail?.tenPhuongAn}` );
  }
  form.setFieldsValue(PhuongAnKinhDoanhDetail);

  const ngayKham = !isEmpty(PhuongAnKinhDoanhDetail?.ngayKham)
    ? moment(PhuongAnKinhDoanhDetail?.ngayKham)
    : null;

  form.setFieldValue("ngayKham", ngayKham);

  const ngayThanhToan = !isEmpty(PhuongAnKinhDoanhDetail?.ngayThanhToan)
    ? moment(PhuongAnKinhDoanhDetail?.ngayThanhToan)
    : null;

  form.setFieldValue("ngayThanhToan", ngayThanhToan);

  const ngayTraCk = !isEmpty(PhuongAnKinhDoanhDetail?.ngayTraCk)
    ? moment(PhuongAnKinhDoanhDetail?.ngayTraCk)
    : null;

  form.setFieldValue("ngayTraCk", ngayTraCk);

  const ngayTraHs = !isEmpty(PhuongAnKinhDoanhDetail?.ngayTraHs)
    ? moment(PhuongAnKinhDoanhDetail?.ngayTraHs)
    : null;

  form.setFieldValue("ngayTraHs", ngayTraHs);
}, [dispatch, PhuongAnKinhDoanhDetail, donHangId, form]);

const onFinish = (values) => {

```

```

    if (DonHangKhamSucKhoeDetail?.trangThai === 2) {
      notification.error({
        message: "Lưu đơn",
        description: "Đơn hàng đã chuyển để hoàn thiện!",
      });
      return;
    }

    if (
      values.totalSigned &&
      (values.totalSigned > 1 || values.totalSigned === values.totalSigns)
    ) {
      notification.error({
        message: "Lưu đơn",
        description: "Đơn hàng đã được ký",
      });
      return;
    }

    const payload = {
      id: values.id,
      tenDonHang: values.tenDonHang,
      phuongAnId: values.phuongAnId,
      ngayKhamThucTe: values.ngayKhamThucTe,
      ngayKhamThucTeTu: values.ngayKhamThucTeTu,
      onSuccess: ({ data }) => {
        // Refresh data after add/update table
        if (isEmpty(donHangId)) {
          setDonHangId(data.data.Id);
          if (data.data.Id && data.data.MaPhieu && childRef.current) {
            childRef.current.signedDocumentCreate(
              data.data.Id,
              data.data.MaPhieu,
              values.tenDonHang
            );
          }
        }

        //if (!isEmpty(donHangId)) dispatch(GetDonHangKhamSucKhoe(donHangId));
        if (reloadData) reloadData();

        notification.info({
          message: "Lưu đơn hàng",
          description: "Lưu thành công thông tin đơn hàng",
        });
      },
    };

    payload.id = donHangId;
    dispatch(UpsertDonHangKhamSucKhoe(payload));
  };

  const renderedTabs = useMemo(
    () => [
      {
        label: "Dịch vụ khám",
        children: (
          <ChiTietPhuongAn
            phuongAnId={PhuongAnKinhDoanhDetail?.id}
            PhuongAnKinhDoanhDetail={PhuongAnKinhDoanhDetail}
          />
        ),
      },
      {
        label: "Nhân sự tham gia",
        children: <NhanSu donHangId={DonHangKhamSucKhoeDetail?.id} />,
      },
      {
        label: "Danh sách vật tư",
        children: <VatTu donHangId={DonHangKhamSucKhoeDetail?.id} />,
      },
    ],
    [DonHangKhamSucKhoeDetail?.id, PhuongAnKinhDoanhDetail]
  );

  return (

```

```

<SlipModalComponent
  ref={childRef}
  maskClosable={false}
  bodyStyle={{ overflowY: "auto", maxHeight: "calc(100vh - 300px)" }}
  width={1400}
  title={!isEmpty(donHangId) ? "Sửa đơn hàng" : "Thêm đơn hàng"}
  objectId={donHangId}
  objectType={"DonHangKSK"}
  documentTypeId={DOCUMENT_TYPE_ID.DonHangKsk}
  widthInfo={550}
  titleTLink="Đơn hàng KSK"
  descriptionTLink={DonHangKhamSucKhoeDetail?.maDonHang}
  isVisibleModal={visibleModal}
  onCancelModal={() => {
    setVisibleModal(!visibleModal);
    form.resetFields();
    if (!isEmpty(donHangId)) setDonHangId(null);
  }}
  onRenderTab={renderedTabs}
  // onSignReloadForm={({data}) => {
  //   if (data.totalSigned === data.totalSigns) {
  //     if (!isEmpty(donHangId)) dispatch(GetDonHangKhamSucKhoe(donHangId));
  //   }
  // }}
  onRenderInfo={() => {
    return (
      <Spin tip="Đang tải..." spinning={loading}>
        <Form
          form={form}
          layout="vertical"
          onFinish={onFinish}
          className="p-3"
        >
          <Row gutter={24}>
            <Col sm={8}>
              <Form.Item name="maDonHang" label="Mã đơn hàng">
                <Input disabled />
              </Form.Item>
            </Col>
            <Col sm={16}>
              <Form.Item
                name="phuongAnId"
                label="Phương án"
                rules={[
                  {
                    required: true,
                    message: "Vui lòng nhập phương án",
                  },
                ]}
              >
                <Select
                  showSearch
                  filterOption={({input, option}) =>
                    (option?.label ?? "").
                      .toLowerCase()
                      .includes(input.toLowerCase())
                  }
                  options={PhuongAnKinhDoanhList?.map((i) => ({
                    label: i.tenPhuongAn,
                    value: i.id,
                  })))}
                  onChange={(e) => onChangePhuongAn(e)}
                />
              </Form.Item>
            </Col>
          </Row>

          <Row gutter={24}>
            <Col sm={24}>
              <Form.Item
                name="tenDonHang"
                label="Tên đơn hàng"

```

```

        rules={[
          {
            required: true,
            message: "Vui lòng nhập tên đơn hàng",
          },
        ]}
      >
      <Input />
    </Form.Item>
  </Col>
</Row>

<Row gutter={24}>
  <Col sm={24}>
    <Form.Item name="doanKhamId" label="Đoàn khám">
      <Select
        disabled
        showSearch
        filterOption={(input, option) =>
          (option?.label ?? "")
            .toLowerCase()
            .includes(input.toLowerCase())
        }
        options={DoanKhamList?.map((i) => ({
          label: i.tenDoan,
          value: i.id,
        })))}
      />
    </Form.Item>
  </Col>
</Row>

<Row gutter={24}>
  <Col sm={12} md={12} lg={12}>
    <Form.Item name="hangMucKhamId" label="Hạng mục khám">
      <Select
        disabled
        showSearch
        filterOption={(input, option) =>
          (option?.label ?? "")
            .toLowerCase()
            .includes(input.toLowerCase())
        }
        options={HangMucKhamList?.map((i) => ({
          label: i.tenThietLap,
          value: i.id,
        })))}
      />
    </Form.Item>
  </Col>
  <Col sm={12} md={12} lg={12}>
    <Form.Item name="mauKhamId" label="Mẫu khám">
      <Select
        disabled
        showSearch
        filterOption={(input, option) =>
          (option?.label ?? "")
            .toLowerCase()
            .includes(input.toLowerCase())
        }
        options={MauKhamList?.map((i) => ({
          label: i.tenThietLap,
          value: i.id,
        })))}
      />
    </Form.Item>
  </Col>
</Row>

<Row gutter={24}>
  <Col sm={8} md={8} lg={8}>
    <Form.Item name="soLuotKham" label="Số lượt khám">
      <InputNumber style={{ width: "100%" }} disabled />
    </Form.Item>
  </Col>
</Row>

```

```

</Col>
<Col sm={8} md={8} lg={8}>
  <Form.Item name="ptChietKhau" label="Chiết khấu (%)">
    <InputNumber style={{ width: "100%" }} disabled />
  </Form.Item>
</Col>
<Col sm={8} md={8} lg={8}>
  <Form.Item name="ptTyLeChenh" label="Chênh HĐ (%)">
    <InputNumber style={{ width: "100%" }} disabled />
  </Form.Item>
</Col>
</Row>

<Row gutter={24}>
  <Col sm={12} md={12} lg={12}>
    <Form.Item name="ngayKham" label="Ngày khám dự kiến">
      <DatePicker
        disabled
        format={dateFormatList}
        style={{ width: "100%" }}
        placeholder="Chọn ngày khám dự kiến"
      />
    </Form.Item>
  </Col>
  <Col sm={12} md={12} lg={12}>
    <Form.Item name="ngayThanhToan" label="Ngày thanh toán">
      <DatePicker
        disabled
        format={dateFormatList}
        style={{ width: "100%" }}
        placeholder="Chọn ngày thanh toán"
      />
    </Form.Item>
  </Col>
</Row>

<Row gutter={24}>
  <Col sm={12} md={12} lg={12}>
    <Form.Item
      name="ngayKhamThucTeTu"
      label="Ngày khám thực tế từ"
    >
      <DatePicker
        showTime={{
          format: "HH:mm",
        }}
        format="DD/MM/YYYY HH:mm"
        //format={dateFormatList}
        style={{ width: "100%" }}
        placeholder="Chọn ngày khám từ giờ"
      />
    </Form.Item>
  </Col>
  <Col sm={12} md={12} lg={12}>
    <Form.Item name="ngayKhamThucTe" label="Đến">
      <DatePicker
        showTime={{
          format: "HH:mm",
        }}
        format="DD/MM/YYYY HH:mm"
        //format={dateFormatList}
        style={{ width: "100%" }}
        placeholder="Chọn ngày khám đến giờ"
      />
    </Form.Item>
  </Col>
</Row>

<Row gutter={24}>
  <Col sm={12} md={12} lg={12}>
    <Form.Item name="ngayTraCk" label="Ngày trả chiết khấu">
      <DatePicker
        disabled
        format={dateFormatList}

```

```

                style={{ width: "100%" }}
                placeholder="Chọn ngày trả chiết khấu"
            />
        </Form.Item>
    </Col>
    <Col sm={12} md={12} lg={12}>
        <Form.Item name="ngayTraHs" label="Ngày trả hồ sơ">
            <DatePicker
                disabled
                format={dateFormatList}
                style={{ width: "100%" }}
                placeholder="Chọn ngày trả hồ sơ"
            />
        </Form.Item>
    </Col>
</Row>

<Row>
    <Col sm={6}>
        <Button
            type="primary"
            onClick={() => {
                form.submit();
            }}
        >
            Lưu
        </Button>
    </Col>
    <Col sm={6}>
        <Button
            type="primary"
            onClick={() => {
                setIsPrinter(true);
                setPrintData({
                    id: DonHangKhamSucKhoeDetail?.id,
                    sophieu: DonHangKhamSucKhoeDetail?.maDonHang,
                });
            }}
            icon={<PrinterOutlined type="primary" />}
        >
            In đơn
        </Button>
    </Col>
    <Col sm={12}>
        {DonHangKhamSucKhoeDetail?.trangThai <= 2 && (
            <Button
                type="primary"
                onClick={() => {
                    dispatch(
                        UpdateDonHangKhamSucKhoe({
                            id: donHangId,
                            trangThai:
                                DonHangKhamSucKhoeDetail?.trangThai === 0
                                    ? 1
                                    : DonHangKhamSucKhoeDetail?.trangThai === 1
                                        ? 2
                                        : 3,
                            onSuccess: () => {
                                reloadData();
                                dispatch(GetDonHangKhamSucKhoe(donHangId));
                            },
                        })
                    );
                }}
            >
                {DonHangKhamSucKhoeDetail?.trangThai === 0
                    ? "Chuyển đang khám"
                    : DonHangKhamSucKhoeDetail?.trangThai === 1
                        ? "Chuyển hoàn thiện hồ sơ"
                        : "Thanh lý hợp đồng"}
            </Button>
        )}
    </Col>
</Row>

```

```
</Form>

<DocumentViewer
  openModel={isPrinter}
  setIsPrinter={setIsPrinter}
  soPhieu={printData.sophieu}
  idPhieu={DonHangKhamSucKhoeDetail?.id}
  setPrintData={setPrintData}
  tenMauIn={"DonKSK.rptDonKSK"}
  loaiPhieu={"DonKsk"}
  tieuDe={`In đơn KSK ${printData.sophieu}`}
  param={{ id: DonHangKhamSucKhoeDetail?.id }}
/>
</Spin>
);
}}
></SlipModalComponent>
);
}
```