

```

import {
  Button,
  Checkbox,
  Col,
  DatePicker,
  Form,
  Input,
  InputNumber,
  Row,
  Select,
  Spin,
  notification,
} from "antd";
import { isEmpty } from "lodash";
import React, { useEffect, useRef } from "react";
import { useDispatch, useSelector } from "react-redux";

import SlipModalComponent from "components/SlipModalComponent";
import {
  GetPhuongAnKinhDoanh,
  UpsertPhuongAnKinhDoanh,
} from "store/slices/khamSuckhoeDoanSlice";
import { DOCUMENT_TYPE_ID, dateFormatList } from "constants";
import moment from "moment";
import ChiTietPhuongAn from "../ChiTietPhuongAn";
import TextArea from "antd/lib/input/TextArea";

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 ModalPhuongAn({
  visibleModal,
  setVisibleModal,
  loading,
  reloadData,
  phuongAnId,
  setPhuongAnId,
  permitList = [],
}) {
  const dispatch = useDispatch();
  const [form] = Form.useForm();

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

  const childRef = useRef(null);

  useEffect(() => {
    form.setFieldValue("ptTyLeChenh", 0);
    if (isEmpty(phuongAnId)) return;
    dispatch(GetPhuongAnKinhDoanh(phuongAnId));
  }, [dispatch, form, phuongAnId]);

  useEffect(() => {
    if (isEmpty(phuongAnId)) return;

```

```

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, phuongAnId, PhuongAnKinhDoanhDetail, form]));

const onFinish = (values) => {
  if (values.soLuotKham < 0) {
    notification.error({
      message: "Lưu phương án",
      description: "Số lượt khám không đúng",
    });
    return;
  }

  if (values.ptChietKhau < 0 || values.ptChietKhau > 100) {
    notification.error({
      message: "Lưu phương án",
      description: "Tỷ lệ chiết khấu không đúng",
    });
    return;
  }

  if (!isEmpty(phuongAnId)) {
    if (
      PhuongAnKinhDoanhDetail?.totalSigns !== null &&
      PhuongAnKinhDoanhDetail?.totalSigned !== null &&
      PhuongAnKinhDoanhDetail?.totalSigned ===
        PhuongAnKinhDoanhDetail?.totalSigns
    ) {
      notification.error({
        message: "Lưu phương án",
        description: "Phương án đã được ký duyệt, bạn không thể sửa",
      });
      return;
    }
  }

  if (!values.coSo && !values.khongCoSo) {
    notification.error({
      message: "Lưu phương án",
      description: "Vui lòng chọn có sổ hoặc không có sổ",
    });
    return;
  }

  const payload = {
    id: values.id,
    tenPhuongAn: values.tenPhuongAn,
    doanKhamId: values.doanKhamId,
    soLuotKham: values.soLuotKham,
    ptChietKhau: values.ptChietKhau,
    ptTyLeChenh: values.ptTyLeChenh,
  };

```

```

hangMucKhamId: values.hangMucKhamId,
mauKhamId: values.mauKhamId,
ngayKham: values.ngayKham,
ngayThanhToan: values.ngayThanhToan,
ngayTraCk: values.ngayTraCk,
ngayTraHs: values.ngayTraHs,
ghiChuKhachHang: values.ghiChuKhachHang || "",
coSo: values.coSo,
khongCoSo: values.khongCoSo,
onSuccess: ({ data }) => {
  // Refresh data after add/update table
  if (isEmpty(phuongAnId)) {
    setPhuongAnId(data.data.Id);
    if (data.data.Id && data.data.MaPhieu && childRef.current) {
      childRef.current.signedDocumentCreate(
        data.data.Id,
        data.data.MaPhieu,
        values.tenPhuongAn
      );
    }
  }
  if (!isEmpty(phuongAnId)) dispatch(GetPhuongAnKinhDoanh(phuongAnId));
  if (reloadData) reloadData();

  notification.info({
    message: "Lưu phương án kinh doanh",
    description: "Lưu thành công thông tin phương án kinh doanh",
  });
},
};

payload.id = phuongAnId;
dispatch(UpsertPhuongAnKinhDoanh(payload));
};

return (
  <SlipModalComponent
    ref={childRef}
    maskClosable={false}
    bodyStyle={{ overflowY: "auto", maxHeight: "calc(100vh - 300px)" }}
    width={1400}
    title={
      !isEmpty(phuongAnId)
        ? "Sửa phương án kinh doanh"
        : "Thêm phương án kinh doanh"
    }
    objectId={phuongAnId}
    objectType={"KskPhuongAnKinhDoanh"}
    documentTypeId={DOCUMENT_TYPE_ID.KskPhuongAnKinhDoanh}
    widthInfo={550}
    titleTLink="Phương án kinh doanh"
    descriptionTLink={` ${PhuongAnKinhDoanhDetail?.maPhuongAn} ${PhuongAnKinhDoanhDetail?.tenPhuongAn}`}
    isVisibleModal={visibleModal}
    onCancelModal={() => {
      setVisibleModal(!visibleModal);
      form.resetFields();
      if (!isEmpty(phuongAnId)) setPhuongAnId(null);
    }}
    onRenderTab={[
      {
        label: "Dịch vụ khám",
        children: (
          <ChiTietPhuongAn
            phuongAnId={phuongAnId}
            PhuongAnKinhDoanhDetail={PhuongAnKinhDoanhDetail}
          />
        ),
      },
    ]}
    onSignReloadForm={(data) => {
      if (data.totalSigned === data.totalSigns) {
        if (!isEmpty(phuongAnId)) {
          dispatch(GetPhuongAnKinhDoanh(phuongAnId));
          reloadData();
        }
      }
    }}
  >

```

```

    }
  }
  onRenderInfo={() => {
    return (
      <Spin tip="Đang tải..." spinning={loading}>
        <Form
          form={form}
          layout="vertical"
          onFinish={onFinish}
          className="p-3"
        >
          <Row gutter={24}>
            <Col sm={24}>
              <Form.Item
                name="tenPhuongAn"
                label="Tên phương án"
                rules={[
                  {
                    required: true,
                    message: "Vui lòng nhập tên phương án",
                  },
                ]}
              >
                <Input />
              </Form.Item>
            </Col>
          </Row>

          <Row gutter={24}>
            <Col sm={24}>
              <Form.Item
                name="doanKhamId"
                label="Đoàn khám"
                rules={[
                  {
                    required: true,
                    message: "Vui lòng nhập đoàn khám",
                  },
                ]}
              >
                <Select
                  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={24} md={24} lg={24}>
              <Form.Item
                name="hangMucKhamId"
                label="Hạng mục khám"
                rules={[
                  {
                    required: true,
                    message: "Vui lòng nhập hạng mục khám",
                  },
                ]}
              >
                <Select
                  showSearch
                  filterOption={(input, option) =>
                    (option?.label ?? "")
                      .toLowerCase()

```

```

        .includes(input.toLowerCase())
    }
    options={HangMucKhamList?.map((i) => ({
        label: i.tenThietLap,
        value: i.id,
    })))}
    />
</Form.Item>
</Col>
</Row>

<Row gutter={24}>
    <Col sm={24} md={24} lg={24}>
        <Form.Item
            name="mauKhamId"
            label="Mẫu khám"
            rules={[
                {
                    required: true,
                    message: "Vui lòng nhập mẫu khám",
                },
            ]}
        >
            <Select
                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"
            rules={[
                {
                    required: true,
                    message: "Vui lòng nhập số lượt khám",
                },
            ]}
        >
            <InputNumber style={{ width: "100%" }} />
        </Form.Item>
    </Col>
    <Col sm={8} md={8} lg={8}>
        <Form.Item
            name="ptChietKhau"
            label="chiết khấu (%)"
            rules={[
                {
                    required: true,
                    message: "Vui lòng nhập tỷ lệ chiết khấu",
                },
            ]}
        >
            <InputNumber style={{ width: "100%" }} />
        </Form.Item>
    </Col>
    <Col sm={8} md={8} lg={8}>
        <Form.Item
            name="ptTyLeChenh"
            label="Chênh HĐ (%)"
            rules={[
                {

```

```

        required: true,
        message: "Vui lòng % tỷ lệ chênh HĐ",
    },
  ]}
  >
  <InputNumber style={{ width: "100%" }} />
</Form.Item>
</Col>
</Row>

<Row gutter={24}>
  <Col sm={12} md={12} lg={12}>
    <Form.Item
      name="ngayKham"
      label="Ngày khám"
      rules={[
        {
          required: true,
          message: "Vui lòng nhập ngày khám",
        }
      ]}
    >
      <DatePicker
        format={dateFormatList}
        style={{ width: "100%" }}
        placeholder="Chọn ngày đánh giá"
      />
    </Form.Item>
  </Col>
  <Col sm={12} md={12} lg={12}>
    <Form.Item
      name="ngayThanhToan"
      label="Ngày thanh toán"
      rules={[
        {
          required: true,
          message: "Vui lòng nhập ngày thanh toán",
        }
      ]}
    >
      <DatePicker
        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="coSo"
      label="Đã có số"
      valuePropName="checked"
    >
      <Checkbox
        onChange={(e) => {
          if (e) form.setFieldValue("khongCoSo", false);
        }}
      />
    </Form.Item>
  </Col>
  <Col sm={12} md={12} lg={12}>
    <Form.Item
      name="khongCoSo"
      label="Chưa có số"
      valuePropName="checked"
    >
      <Checkbox
        onChange={(e) => {
          if (e) form.setFieldValue("coSo", false);
        }}
      />
    </Form.Item>
  </Col>
</Row>

```

```

        </Form.Item>
      </Col>

    </Row>
    <Row gutter={24}>
      <Col sm={24} md={24} lg={24}>
        <Form.Item
          name="ghiChuKhachHang"
          label="Ghi chú khách hàng"
          // rules={[
            // {
              //   required: true,
              //   message: "Vui lòng nhập ghi chú",
            // },
          // ]}
        >
          <TextArea rows={4} />
        </Form.Item>
      </Col>
    </Row>

    <Row gutter={24}>
      <Col sm={12} md={12} lg={12}>
        <Form.Item name="ngayTraCk" label="Ngày trả chiết khấu">
          <DatePicker
            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
            format={dateFormatList}
            style={{ width: "100%" }}
            placeholder="Chọn ngày trả hồ sơ"
          />
        </Form.Item>
      </Col>
    </Row>

    <Row>
      <Col sm={24}>
        <Button
          disabled={PhuongAnKinhDoanhDetail?.totalSigned > 1}
          type="primary"
          onClick={() => {
            form.submit();
          }}
        >
          Lưu
        </Button>
      </Col>
    </Row>
  </Form>
</Spin>
);
}}
></SlipModalComponent>
);
}

```