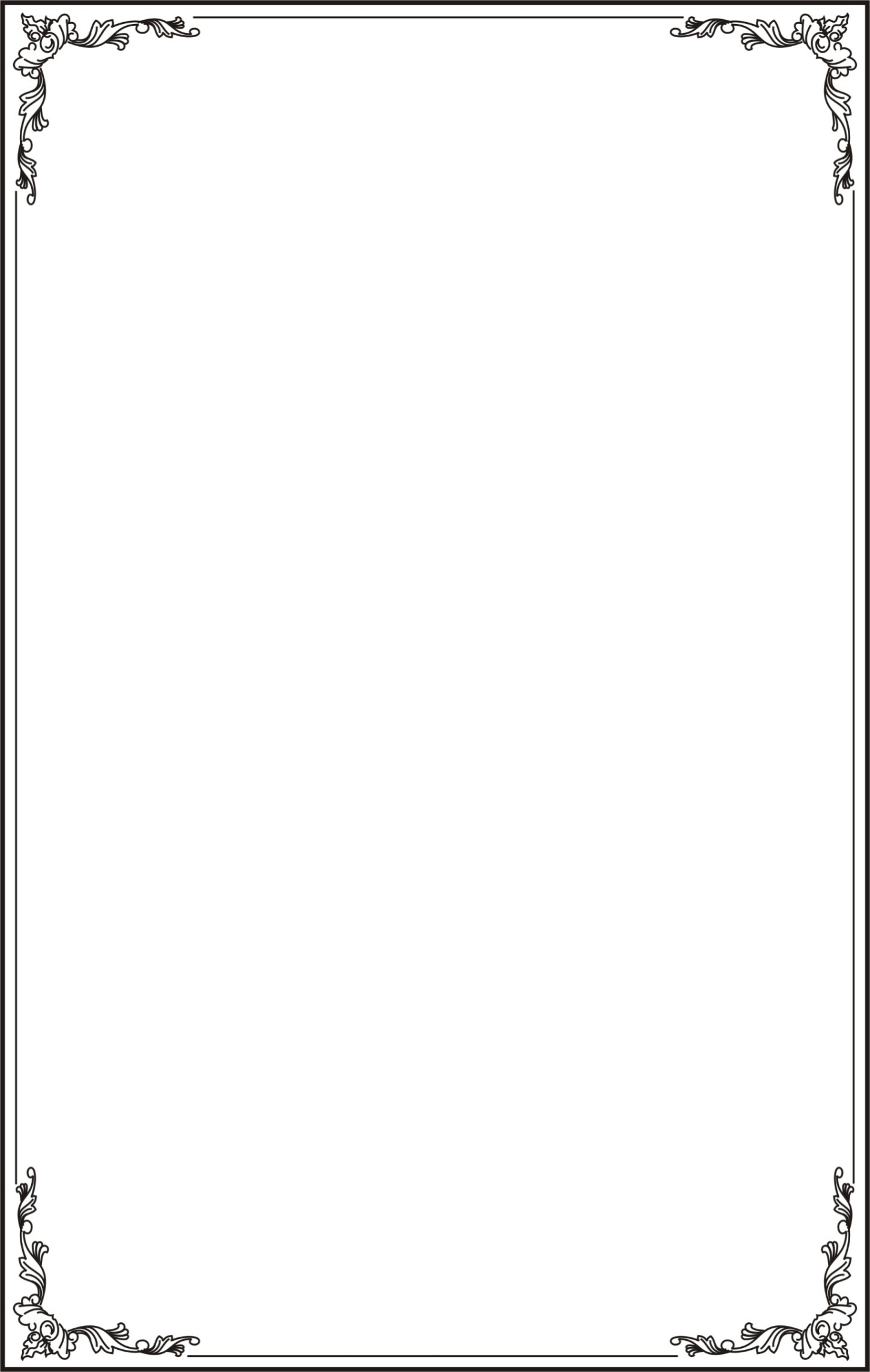
**CÔNG TY CỔ PHẦN**

**VIỄN THÔNG FPT**

****

**BÀI THU HOẠCH**

**ĐÀO TẠO TÂN BINH R&D TUẦN 9 & 10**

|  |  |
| --- | --- |
| Người hướng dẫn: | **MR. TRƯƠNG TẤN SANG** |
| Người thực hiện: | **TRẦN KHÁNH GIA UY** |
| Đơn vị công tác: | **Trung tâm giám sát và đảm bảo dịch vụ (SCC)** |
| Vị trí: | **Open Source Web Software Developer** |
| Bộ phận: | **SCC** |

***Thành phố Hồ Chí Minh, tháng 07 năm 2021***

# MỤC LỤC

[MỤC LỤC 2](#_Toc78130045)

[CHƯƠNG 1: TÌM HIỂU ANT DESIGN PRO 3](#_Toc78130046)

[1.1 Ant design pro 3](#_Toc78130047)

[1.2 Demo bizchart và tạo table chi tiết số lượng chat trên Ant design pro 4](#_Toc78130048)

[1.3 Source code 10](#_Toc78130049)

[1.4 Demo 10](#_Toc78130050)

[CHƯƠNG 2: TÌM HIỂU VỀ HỆ THỐNG INCIDENT MANAGEMENT 11](#_Toc78130051)

[2.1 Giới thiệu 11](#_Toc78130052)

[2.1.1 Giới thiệu nghiệp vụ 11](#_Toc78130053)

[2.1.2 Cài đặt cơ bản 11](#_Toc78130054)

[2.2 Cài đặt hiển thị thông tin sự cố 12](#_Toc78130055)

[2.2.1 Vấn đề 12](#_Toc78130056)

[2.2.2 Xử lý vấn đề 12](#_Toc78130057)

[2.3 Cài đặt hiển thị thông tin khi check incident 14](#_Toc78130058)

[2.3.1 Vấn đề 14](#_Toc78130059)

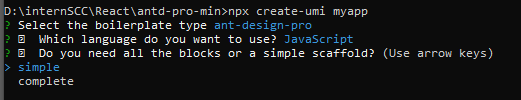
[2.3.2 Xử lý vấn đề 14](#_Toc78130060)

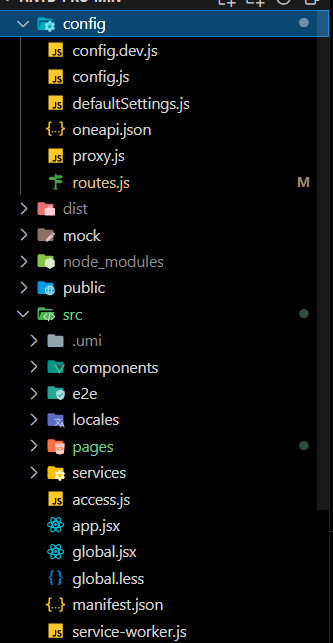
[2.4 Source code 15](#_Toc78130061)

[2.5 Demo 15](#_Toc78130062)

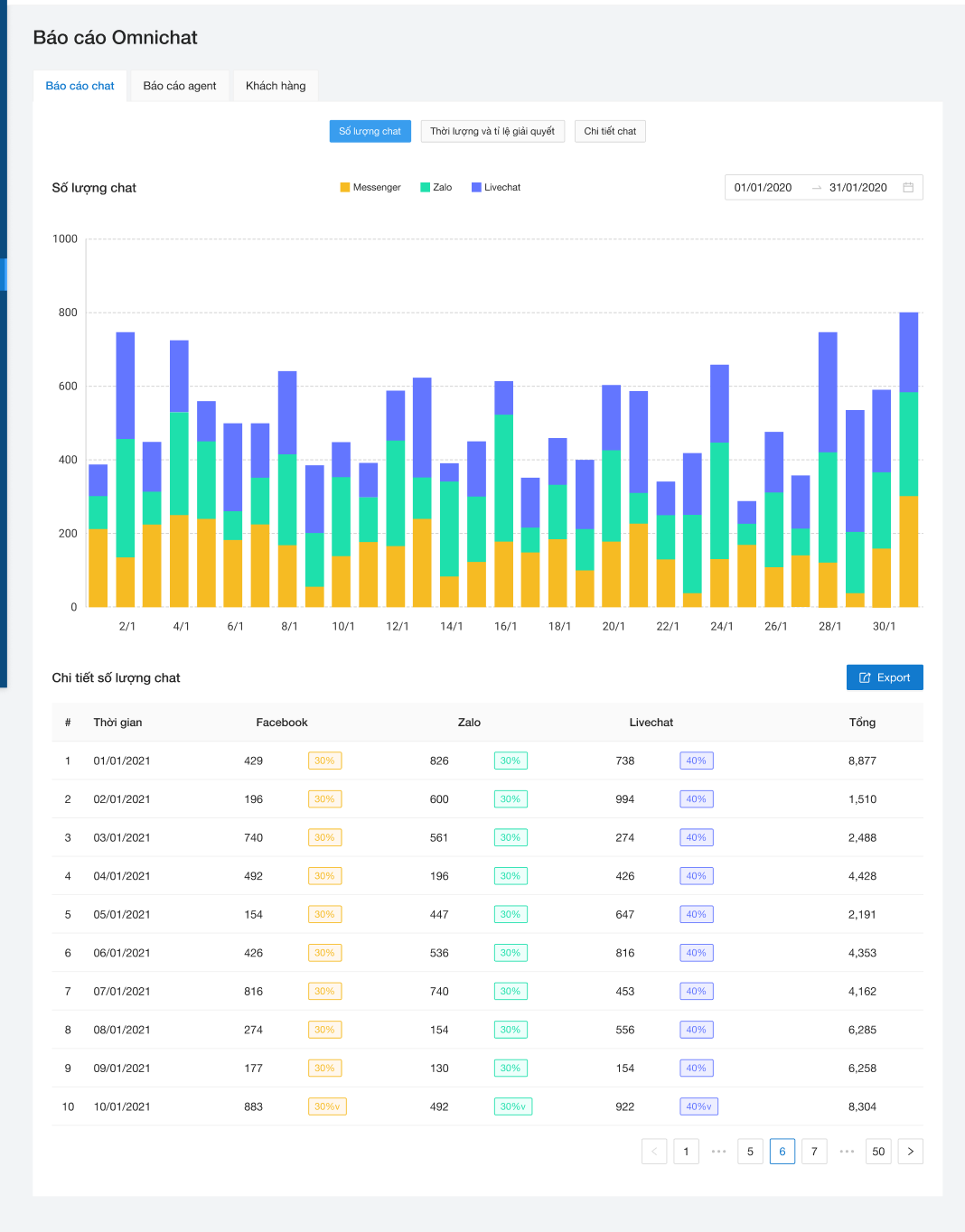
# CHƯƠNG 1: TÌM HIỂU ANT DESIGN PRO

## Ant design pro

* + 1. **Cài đặt**
* ****Cài đặt Ant design pro khá đơn giản với lệnh npx create-umi myapp và em chọn cài đặt bản single sử dụng ngôn ngữ JavaScript
  + 1. **Cấu trúc thư mục**
* Cấu trúc thư mục của project sau khi cài đặt như sau:

****

* Ant design pro sử dụng UmiJS và DvaJS. Vì thế, việc routing sẽ được cài đặt trong file ***routes.js*** và config những thứ khác ở trong file ***config.js***.
* Ở đây em sẽ quan tâm đến 3 folder là *components,* *pages* và *services  :*
  + Folder *components* sẽ chứa các components sử dụng chung
  + Folder *pages* sẽ chứa các trang trong project
  + Folder *services* sẽ chứa các thủ tục gọi api
  1. **Demo bizchart và tạo table chi tiết số lượng chat trên Ant design pro**
* Em sẽ thử tạo 1 trang sử dụng bizchart và table như sau:



* + 1. **Cài đặt chung**
* Đầu tiên em sẽ tạo ra một trang báo cáo omni có các thẻ như *Báo cáo chat, Báo cáo agent, Khách hàng* để chứa chart và table
* Trong file ***/reports/index.js*** :

import { Tabs } from 'antd';

import ReportChat from './reportChat';

import './index.less';

const { TabPane } = Tabs;

const reportOmni = () => {

  function callback(key) {

    console.log(key);

  }

  return (

    <div>

      <h3>Báo cáo Omnichat</h3>

      <div className="card-container">

        <Tabs onChange={callback} type="card">

          <TabPane tab="Báo cáo chat" key="1">

            <ReportChat />

          </TabPane>

          <TabPane tab="Báo cáo agent" key="2">

            Content of Tab Pane 2

          </TabPane>

          <TabPane tab="Khách hàng" key="3">

            Content of Tab Pane 3

          </TabPane>

        </Tabs>

      </div>

    </div>

  );

};

export default reportOmni;

* Sau đó em sẽ thêm các thẻ *Số lượng chat, Tỉ lệ và thời lượng giải quyết* và C*hi tiết chat* trong thẻ *Báo cáo chat*
* Trong file ***/reports/reportChat/index.js***:

import { Tabs } from 'antd';

import CountChat from './countChat';

const { TabPane } = Tabs;

const ReportChat = () => {

  function callback(key) {

    console.log(key);

  }

  return (

    <div>

      <Tabs defaultActiveKey="1" type="card" centered onChange={callback}>

        <TabPane tab="Số lượng chat" key="1">

          <CountChat />

        </TabPane>

        <TabPane tab="Thời lượng và tỉ lệ giải quyết" key="2">

          Content of Tab Pane 2

        </TabPane>

        <TabPane tab="Chi tiết chat" key="3">

          Content of Tab Pane 3

        </TabPane>

      </Tabs>

    </div>

  );

};

export default ReportChat;

* Sau đó, em sẽ tạo một component để chứa bizchart và table chi tiết số lượng chat
* Trong file ***/reports/reportChat/countChat/index.js***:

import Chart from './components/chart';

import Table from './components/table';

const CountChat = () => {

  return (

    <div>

      <Chart />

      <Table />

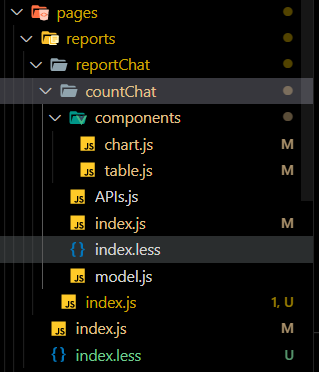
    </div>

  );

};

export default CountChat;

* Bây giờ em sẽ tạo chart và table
  + 1. **Demo bizchart**
* Trước tiên, em sẽ chia folder của mình như sau

****

* Vì ở đây em không có API cụ thể để lấy data nên em sẽ tạo data giả định với cấu trúc như sau :

const data = [

  {

    id: 1,

    date: '18/04/2021',

    type: 'Zalo',

    amount: 100,

  },

  {

    id: 2,

    date: '19/04/2021',

    type: 'Zalo',

    amount: 200,

  },

  {

    id: 3,

    date: '20/04/2021',

    type: 'Zalo',

    amount: 300,

  }, …

* Trong file ***chart.js*** em sẽ tạo chart sử dụng bizchart 4.0:

registerInteraction('element-link', {

  start: [{ trigger: 'interval:mouseenter', action: 'element-link-by-color:link' }],

  end: [{ trigger: 'interval:mouseleave', action: 'element-link-by-color:unlink' }],

});

const ChartComponent = () => {

  return (

    <div>

      <Space direction="vertical" size={12}>

        <RangePicker />

      </Space>

      <Chart padding="auto" autoFit height={500} data={data}>

        <Tooltip shared showMarkers={false} />

        <Interval position="date\*amount" color="type" adjust="stack" />

        <Legend position="top" />

        <Interaction type="element-highlight" />

        <Interaction type="element-link" />

      </Chart>

    </div>

  );

};

export default ChartComponent;

* + 1. **Table**
* Tương tự với chart em cũng sẽ tạo data giả định như sau :

const data = [

  {

    id: 1,

    date: '01/01/2021',

    mess: 315,

    zalo: 123,

    livechat: 645,

    sum: 1083,

  },

  {

    id: 2,

    date: '02/01/2021',

    mess: 334,

    zalo: 625,

    livechat: 244,

    sum: 1203,

  },

];

* Trong file ***Table.js***em sẽ quy định table có các cột như sau:

const columns = [

  {

    title: '#',

    dataIndex: 'id ',

    key: 'id',

  },

  {

    title: 'Thời gian',

    dataIndex: 'date',

    key: 'date',

  },

  {

    title: 'Facebook',

    dataIndex: 'mess',

    key: 'mess',

    render: (text, record) => (

      <div>

        <Row>

          <Col span="12"> {text}</Col>

          <Col span="12">

            <Tag color="volcano">{Math.round((text / record.sum) \* 100)}%</Tag>

          </Col>

        </Row>

      </div>

    ),

  },

  {

    title: 'Zalo',

    key: 'zalo',

    dataIndex: 'zalo',

    render: (text, record) => (

      <div>

        <Row>

          <Col span="12"> {text}</Col>

          <Col span="12">

            <Tag color="green">{Math.round((text / record.sum) \* 100)}%</Tag>

          </Col>

        </Row>

      </div>

    ),

  },

  {

    title: 'Livechat',

    key: 'livechat',

    dataIndex: 'livechat',

    render: (text, record) => (

      <div>

        <Row>

          <Col span="12"> {text}</Col>

          <Col span="12">

            <Tag color="geekblue">{Math.round((text / record.sum) \* 100)}%</Tag>

          </Col>

        </Row>

      </div>

    ),

  },

  {

    title: 'Tổng',

    key: 'sum',

    dataIndex: 'sum',

  },

];

const TableComponent = () => {

  return (

    <div>

      <div>

        <Row>

          <Col flex="300px">

            <b>Chi tiết số lượng chat</b>

          </Col>

          <Col flex="1100px"></Col>

          <Col flex="auto">

         <Button type="primary" icon={<ExportOutlined />}>

           Export

         </Button>

          </Col>

        </Row>

      </div>

      <Table columns={columns} dataSource={data} />

    </div>

  );

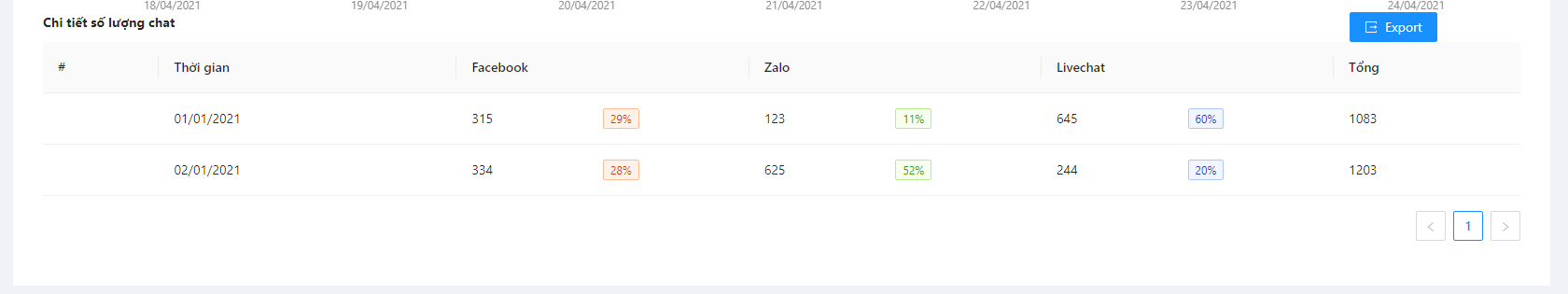
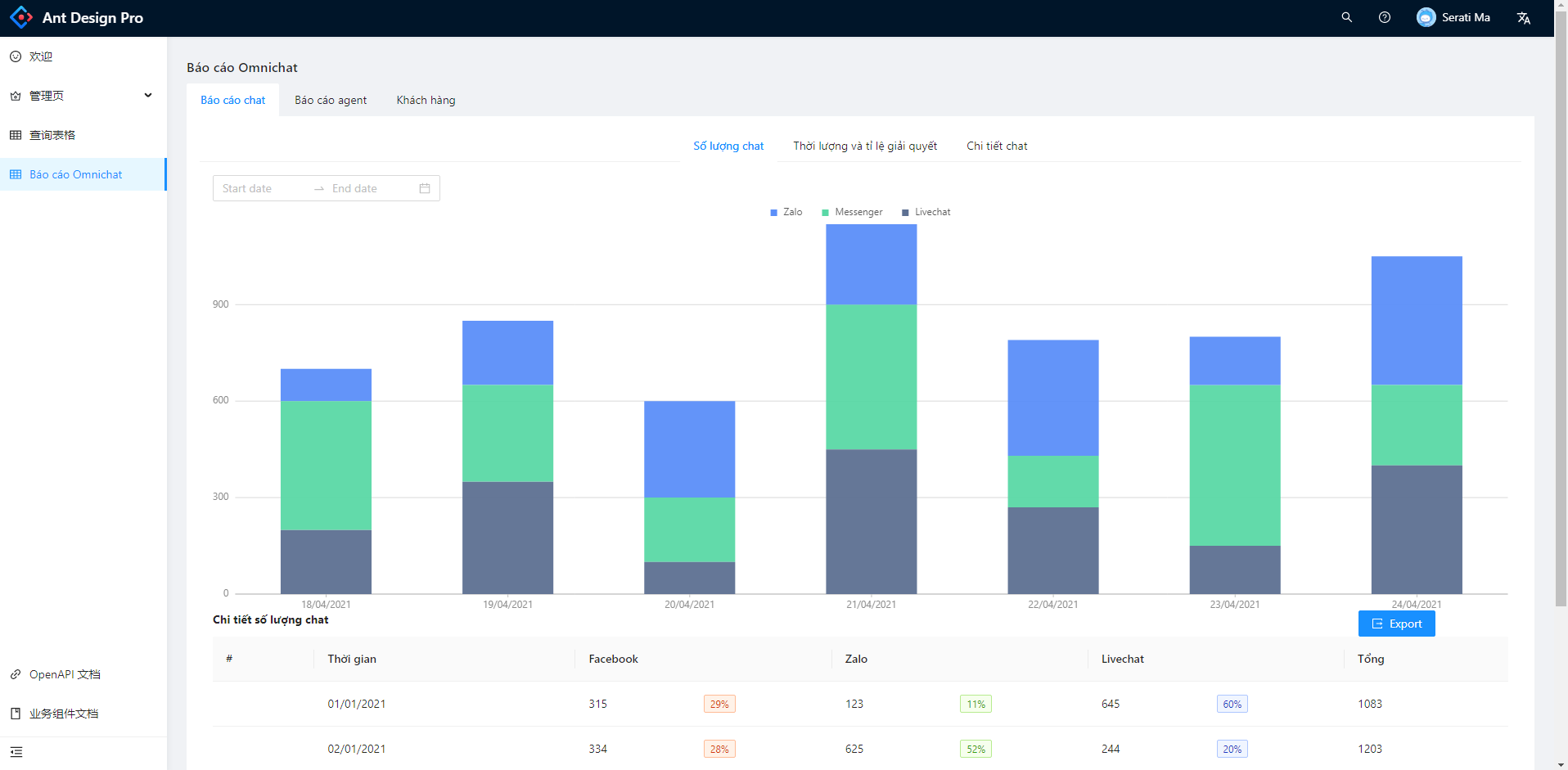
};

export default TableComponent;

* 1. **Source code**
* Tất cả source code nằm trong link github sau:

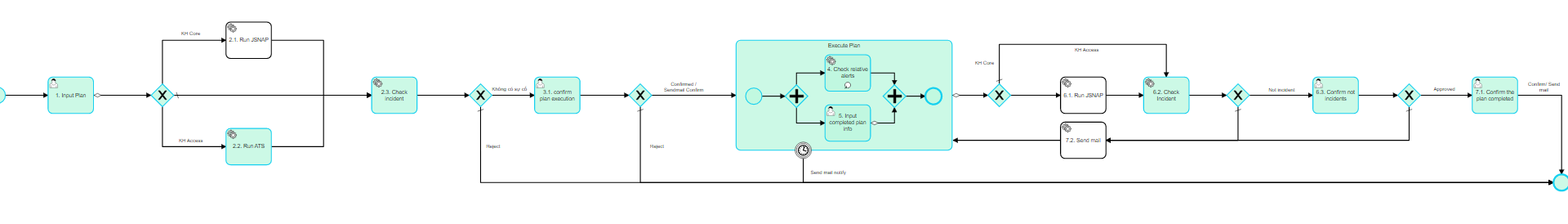
[internSCC/React/antd-pro-min at main · uytran36/internSCC (github.com)](https://github.com/uytran36/internSCC/tree/main/React/antd-pro-min)

* 1. **Demo**

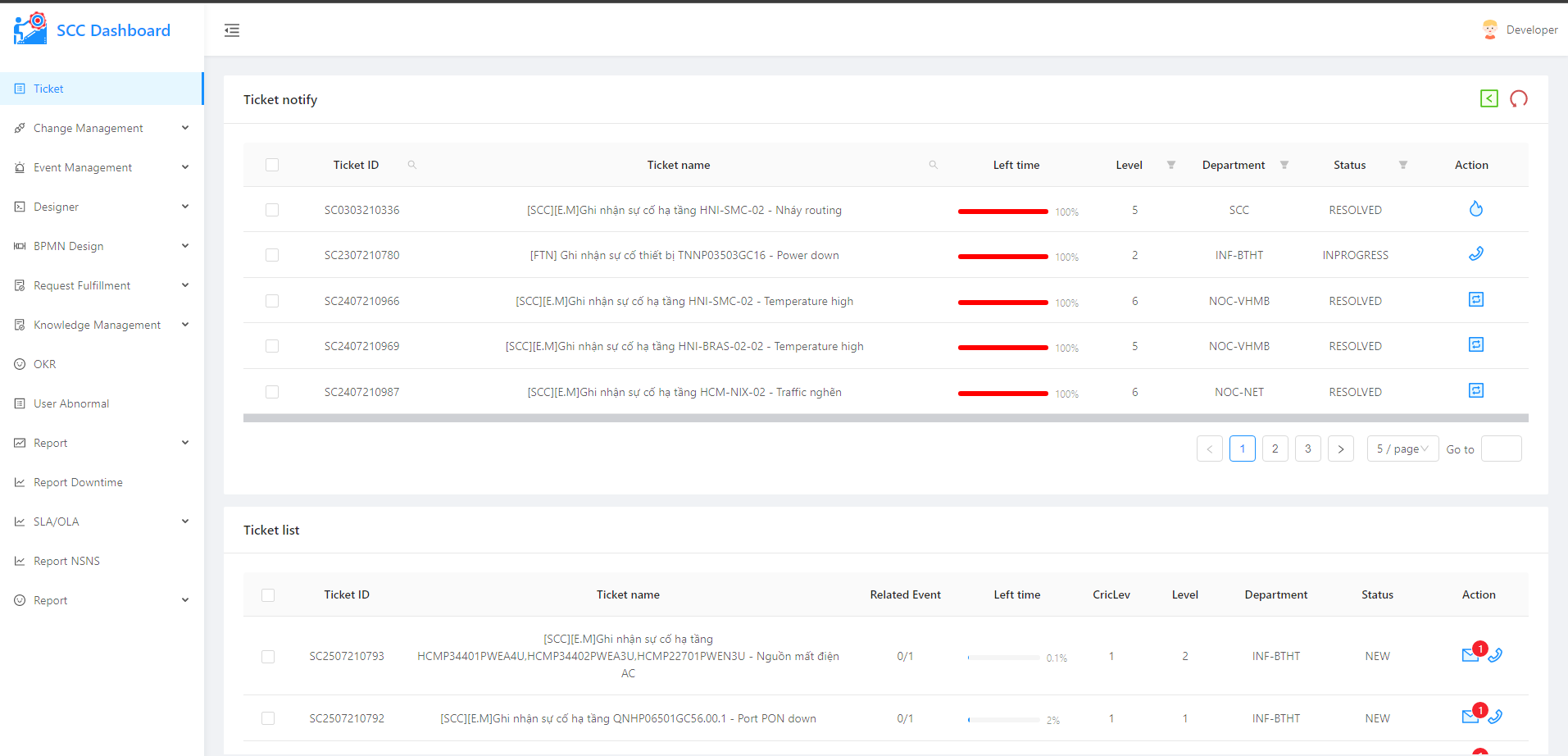
****

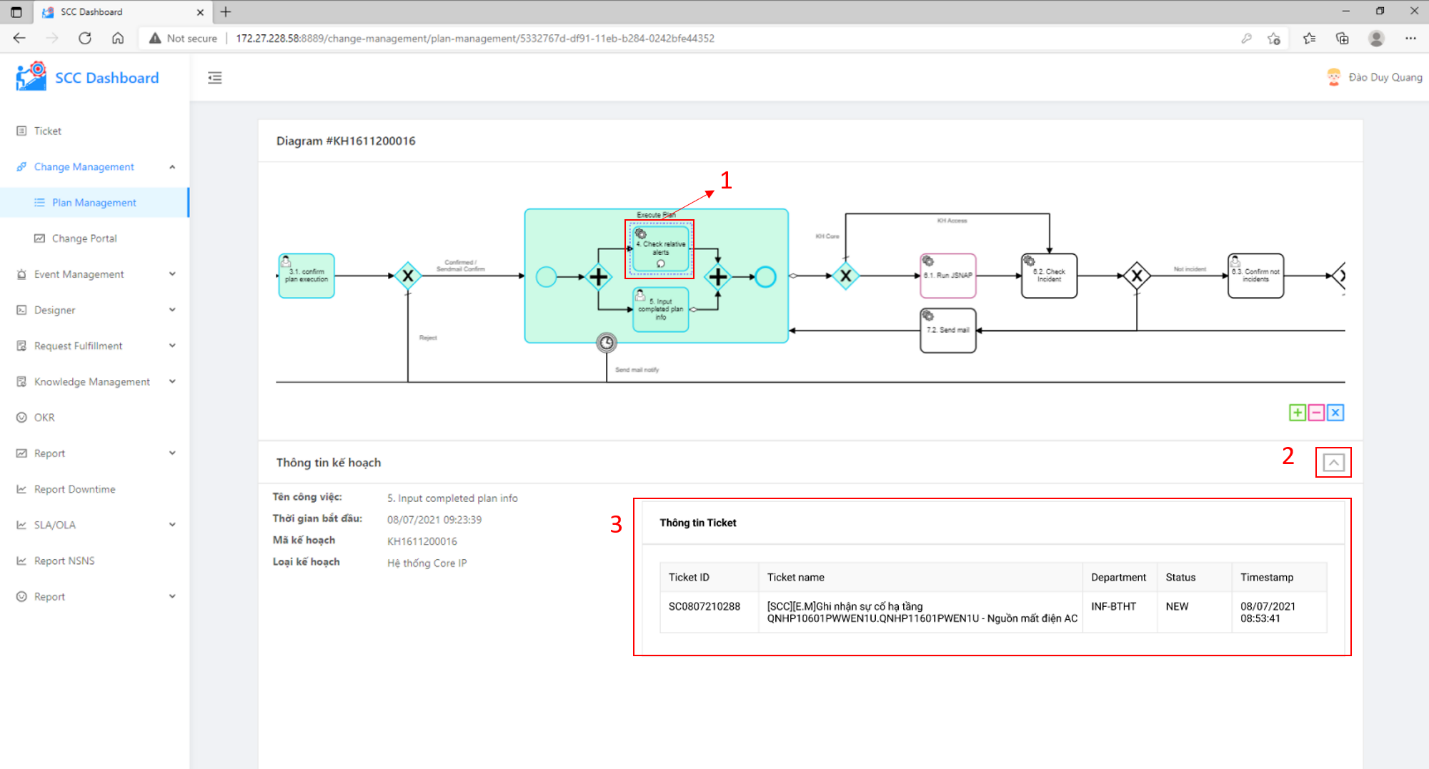
# CHƯƠNG 2: TÌM HIỂU VỀ HỆ THỐNG INCIDENT MANAGEMENT

* 1. **Giới thiệu**
     1. **Giới thiệu nghiệp vụ**
* Hằng đêm (từ 23h đến 5h sáng) đội ngũ sẽ kiểm tra các thiết bị hạ tầng và thực hiện các kế hoạch như nâng cấp, bảo trì, …
* Diagram này sẽ giúp đội ngũ có thể thao tác dễ dàng hơn trong việc kiểm tra, quản lý và thực hiện các kế hoạch.



* Bước *1. Input plan*: Đây là bước nhập kế hoạch
* Sau khi đã nhập kế hoạch xong qua bước *2.1. Run JSNAP*: đây là bước chạy các ip của các thiết bị trong hạ tầng, sau đó qua bước kiểm tra incident
* Bước *2.2. Check incident:* đây là bước kiểm tra có sự cố phát sinh trong thiết bị hay không nếu có thì reject còn nếu không qua bước *3.1. Confirm plan* execution để xác nhận kế hoạch
* *3.1. Confirm plan execution:* đây là bước xác nhận kế hoạch, sau khi đã xác nhận, thì chuyển qua bước thực hiện kế hoạch, nếu không xác nhận thì reject
* Ở bước thực hiện kế hoạch, sẽ có 2 nhiệm vụ:
  + Bước *4. Check relative alerts*: kiểm tra các cảnh báo liên quan đến sự cố
  + Bước *5. Input completed plan info*: Nhập đầy đủ thông tin kế hoạch
* Trong bước thực hiện kế hoạch sẽ gửi mail thông báo
* Sau khi thực hiện kế hoạch, phải thực hiện lại bước *6.1 Run JSNAP* và kiểm tra incident 1 lần nữa tại bước *6.2. Check incident.*
* Nếu có lỗi thì sẽ gửi mail tại bước *7.2. Send mail* và quay lại bước thực hiện kế hoạch. Nếu như không có lỗi thì qua bước xác nhận bước *6.3. Confirm not incidents,* sau đó sẽ được phê duyệt là đã hoàn thành xong kế hoạch và gửi mail thông báo.
  + 1. **Cài đặt cơ bản**
* Em sẽ clone theo repo sau: [Files · sangtt9/dev · SangTT9 / ant\_template · GitLab](http://172.27.228.161/sangtt9/ant_template/-/tree/sangtt9/dev) và đăng nhập. Như vậy em đã có thể sử dụng được hệ thống

****

* 1. **Cài đặt hiển thị thông tin sự cố**
     1. **Vấn đề**
* Hiện tại, ở bước *4. Check relative alerts* không hiển thị ra thông tin công việc chính vì thế, em sẽ thêm thông tin vào bước này như sau:
* Chú thích:

• Số 1: Thao tác click chọn để xem được thông tin Ticket

• Số 2: Icon thả xuống để mở rộng danh sách thông tin Ticket

• Số 3: Danh sách thông tin Ticket sự cố liên quan đến kế hoạch

* + 1. **Xử lý vấn đề**
* Đầu tiên em sẽ viết hàm gọi api để lấy thông tin, trong file ***services.js***:

export async function getTicketById(data) {

  return request(`${URL\_API\_SCCD\_PRODUCTION}/get-ticket-by-ip`, {

    method: 'POST',

    params: {

      api\_key: API\_KEY\_GET\_TICKET,

    },

    headers: {

      'Content-Type': 'application/json',

    },

    data: {

      ...data,

    },

  });

}

* Cấu trúc gọi api có body như sau :

{

    "deviceIp": "HNIB22802ME35,BGGM001-IPMS,BDHP03802GC57",

    "startDate": "2020-12-28 00:03:00",

    "endDate": "2020-12-31 00:50:00"

}

* Trong đó :
  + deviceIp : tên các thiết bị
  + startDate : ngày bắt đầu
  + endDate : ngày kết thúc
* Để lấy được tên các thiết bị, em sẽ phải truy cập đến state detailPlan/processInfo/listDevice trong Redux.
* Với ngày bắt đầu em sẽ lấy *endTime* trong bước *2.3. check incident* đã được lưu trong Redux.
* Với ngày kết thúc em sẽ lấy thời gian hiện tại.
* Như vậy logic để xử lý cấu trúc trên như sau, trong file ***model.js***:

devices = yield select(state => state.detailPlan.processInfo.listDevice);

listDeviceIp = yield devices

.map(function f(device) {

   return device.deviceName;

  })

  .join(',');

listActivityInstance = yield select(state => state.diagram.historyActivityInstance);

const cmTask23 = yield listActivityInstance.find(

item => item.activityId === 'cm\_task\_2\_3',

);

data = {

deviceIp: listDeviceIp,

  startDate: moment(cmTask23.endTime).format('YYYY-MM-DD HH:mm:ss'),

  endDate: moment().format('YYYY-MM-DD HH:mm:ss'),

};

* Sau đó, em sẽ gọi API vừa tạo mỗi khi nhấn vào bước 4, trong file ***model.js*** :

taskInfo = yield call(getTicketById, data);

const taskInfoCmTask4 = listActivityInstance.find(

item => item.activityId === 'cm\_task\_4',

);

const fakeTaskInfo = [

[

{

    activityId: taskInfoCmTask4.activityId,

     activityName: taskInfoCmTask4.activityName,

     activityType: taskInfoCmTask4.activityType,

     id: taskInfoCmTask4.id,

     startTime: taskInfoCmTask4.startTime,

   },

 ],

 { result: { info: { listTicket: taskInfo.data } } },

];

yield put({

type: 'save',

  payload: {

   taskInfo: fakeTaskInfo,

   taskId: payload.taskId,

  },

});

* Cuối cùng, em sẽ sửa lại các component sao cho khớp với data trả về khi gọi api.
* Trong file ***infoPlanCommon.jsx***em sẽ thêm vào trường hợp sau:

case 'cm\_task\_4':

return <ListTicket data={task[1]?.result} />;

* Và trong file ***ListTicket.jsx*** em sẽ sửa lại các column trong table sao cho khớp với data.
  1. **Cài đặt hiển thị thông tin khi check incident**
     1. **Vấn đề**
* Hiện tại khi nhấn vào bước *6.2. Check incident* vẫn chưa có data cho Thông tin ticket. Vì vậy em sẽ lấy dữ liệu từ bước *4. Check relative alerts* để hiển thị thông tin ticket
  + 1. **Xử lý vấn đề**
* Tương tự như ở mục **II.**em sẽ gọi lại API getTicketById tuy nhiên trong body, startDate sẽ là endTime của cm\_start\_event.
* Như vậy trong file ***model.js***, em sẽ sử dụng switch case để với trường hợp người dùng nhấn vào bước 4 hay 6.2 hay các bước khác sẽ hiển thị tương ứng dữ liệu:

case 'cm\_task\_6\_2': {

devices = yield select(state => state.detailPlan.processInfo.listDevice);

listDeviceIp = devices

.map(function f(device) {

return device.deviceName;

  })

  .join(',');

listActivityInstance = yield select(state => state.diagram.historyActivityInstance);

const cmStartEvent = listActivityInstance.find(

item => item.activityId === 'cm\_start\_event',

);

data = {

deviceIp: listDeviceIp,

  startDate: moment(cmStartEvent.endTime).format('YYYY-MM-DD HH:mm:ss'),

  endDate: moment().format('YYYY-MM-DD HH:mm:ss'),

};

const serviceTask = yield call(requestServiceTask, body);

const ticket = yield call(getTicketById, data);

taskInfo = [serviceTask[0], { result: { info: { listTicket: ticket.data } } }];

yield put({

type: 'save',

  payload: {

   taskInfo,

    taskId: payload.taskId,

  },

});

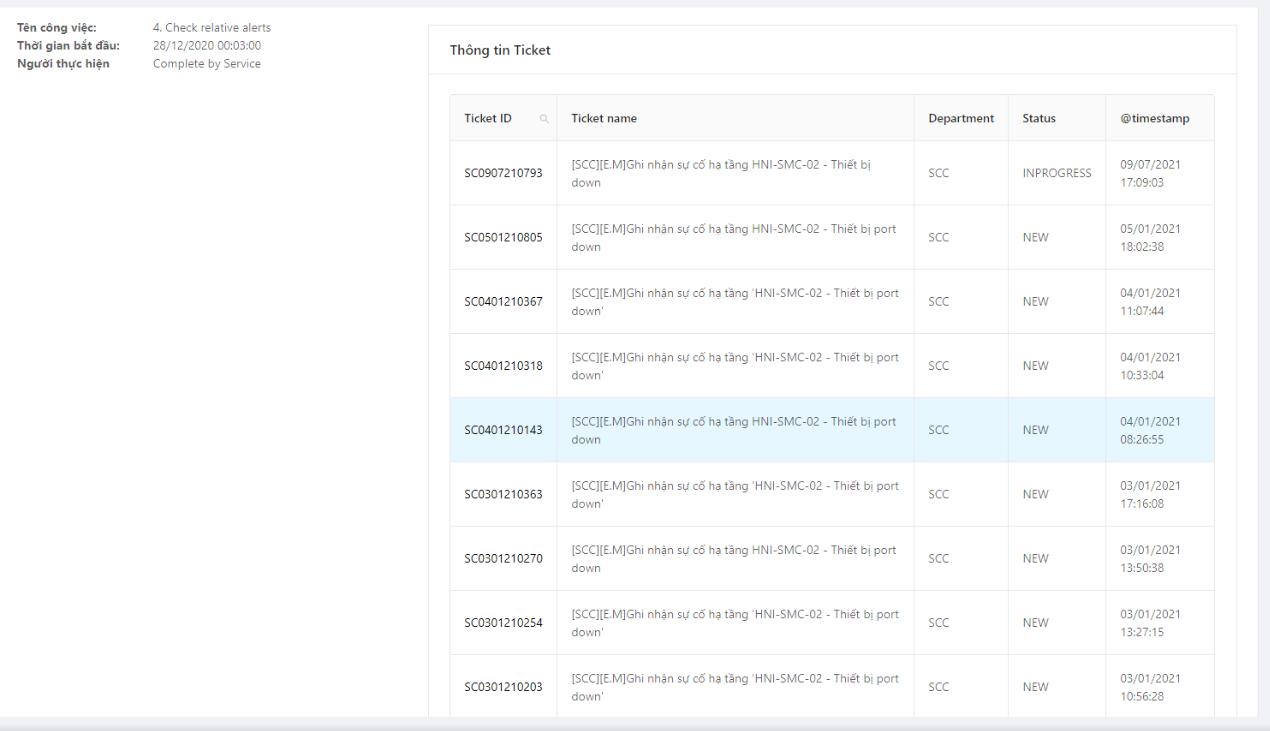
break;

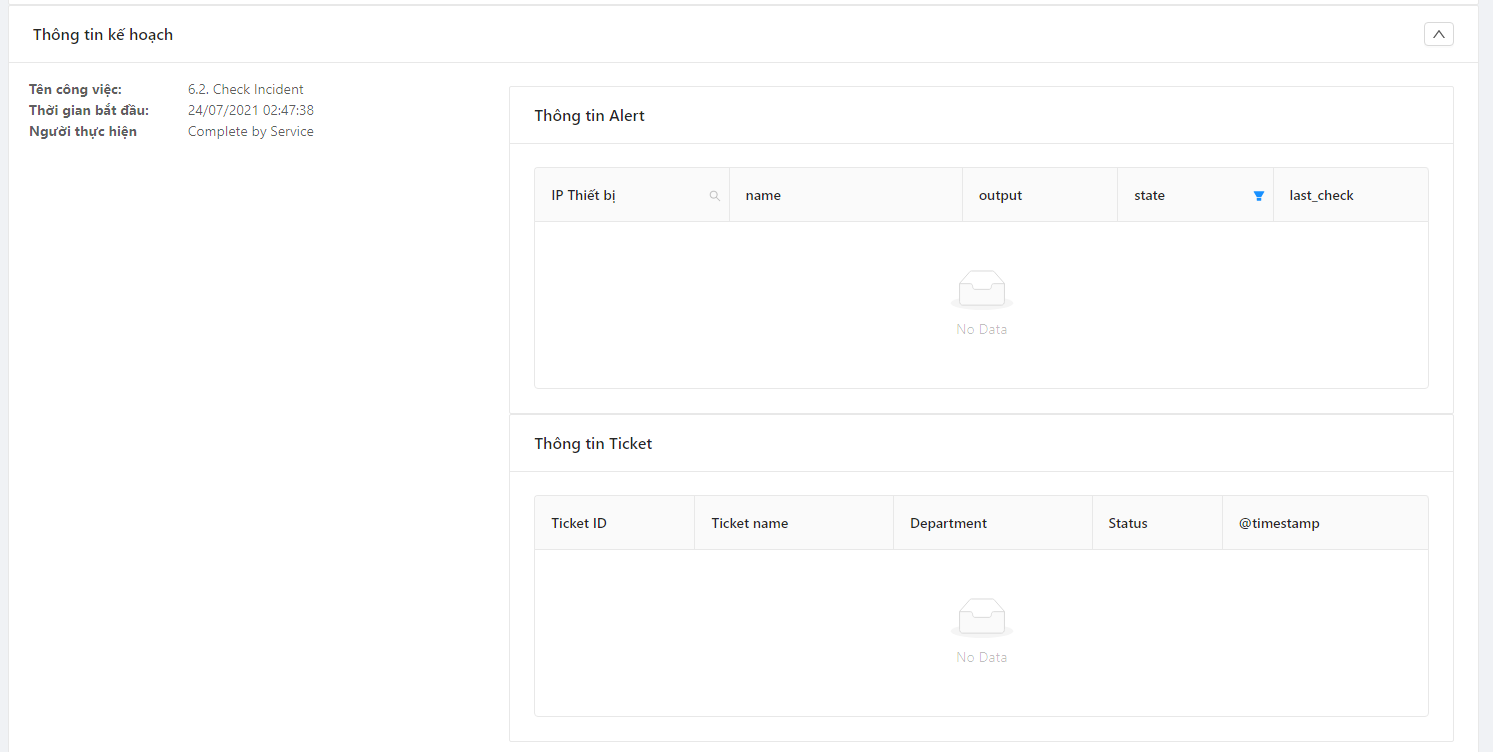
}

* 1. **Source code**
* Tất cả source code nằm trong link github sau:

[Files · uy/dev · SangTT9 / ant\_template · GitLab](http://172.27.228.161/sangtt9/ant_template/-/tree/uy/dev)

* 1. **Demo**

****

****