

Báo cáo tiến độ

Ngày 23/11-29/11

GVHD: Huỳnh Xuân Phụng

Sinh Viên thực hiện:

17110213 Trần Cao Quyền

17110181 Nguyễn Trọng Luật

Kế Hoạch Báo Cáo:

1. Nội dung thực hiện trong tuần
 - Fetch data từ server về app
2. Nội dung thực hiện trong tuần tới

Nội dung báo cáo:

1/ Code gán đơn hàng:

- Với mỗi cụm đã được phân, cụm được chia làm 2 phần là dataInTime (những đơn hàng phù hợp với thời gian) và dataOutTime (những đơn hàng không phù hợp thời gian và cần xem xét lại)
- Sau khi sắp xếp các đơn hàng theo lộ trình thích hợp, dữ liệu sẽ được cập nhập về Database
- Duyệt qua các cụm:
 - a. DataInTime:
 - + Trong cụm dữ liệu được gán là dataInTime và dataOutTime, dataInTime được tính trước vì nó đã có lộ trình phù hợp, dataOutTime sẽ được tính sau khi xem xét các điều kiện
 - + Với dữ liệu trong dataInTime, nó sẽ được chuyển hóa thành dạng string (stringify) để gán vào cột Chuoi trong bảng ChuoiGiaoHang
 - + Tạo ChuoiGiaoHang với hàng creatChuoi

```
const promises2 = resultData.map(async (e) => {  
  
    let dataInTime = e.dataInTime  
    let dataOutTime = e.dataOutTime  
  
    let dataPostChuoiInTime = {  
        Chuoi: JSON.stringify({ chuoidonhang: dataInTime }),  
        SoLuong: dataInTime.length  
    }  
    let chuoiData = await createChuoi(dataPostChuoiInTime);  
    chuoiData = await chuoiData.json();  
});
```

- + Hàm createChuoi (data trả về là dữ liệu của chuỗi vừa được tạo nhằm lấy id của chuỗi gán vào đơn hàng):

```
function createChuoi(data) {  
    let dataPost = data;  
    let settings = {  
        method: "POST",  
        body: JSON.stringify(dataPost),  
        headers: { 'Content-Type': 'application/json' },  
    };  
    return fetch('https://servert1cn.herokuapp.com/chuoi GIAO HANG', settings);  
}
```

- + Sau khi tạo xong chuỗi, tiếp tục là phân tạo thông báo cho khách hàng và cập nhật lại thông tin đơn hàng:

```
const inTime = dataInTime.map(async (e) => {
  let dataThongbao = {
    NoiDung: 'Đơn hàng ' + e.donhang.id + ' sẽ được giao đến địa chỉ: ' +
      e.address.TenDiaChi + ' vào khung giờ (' + e.address.TimeRange + ')',
    Type: 'giao hang',
    NguoiDungId: e.reciver.id
  }
  await createThongBao(dataThongbao);
  await updateDonhang(e.donhang.id, e.address.id, chuoiData.data.id, e.price);
  await updateStatus(e.donhang.id);
})
const inTimePromises = await Promise.all(inTime);
```

- + Thông báo được tạo dựa trên dữ liệu trong dataInTime và tạo thông báo bằng hàm createThongBao:

```
async function createThongBao(data) {
  let dataPost = data;
  let settings = {
    method: "POST",
    body: JSON.stringify(dataPost),
    headers: { 'Content-Type': 'application/json' },
  };
  return fetch('https://servertlcn.herokuapp.com/thongbao', settings);
}
```

- + Khi đã tạo xong thông báo, tiếp đến là cập nhật thông tin đơn hàng (đơn hàng nằm trong chuỗi giao hàng nào, đơn hàng sẽ được gửi đến địa chỉ nào) bằng hàm updateDonhang:

```
function updateDonhang(iddonhang, iddiachi, idchuoi, price) {
  let dataPut = {
    TongTien: TongTien + price,
    DiaChiId: iddiachi,
    ChuoiGiaoHangId: idchuoi,
  }
  let settings = {
    method: "PUT",
    body: JSON.stringify(dataPut),
    headers: { 'Content-Type': 'application/json' },
  };
  return fetch('https://servertlcn.herokuapp.com/donhang/' + iddonhang, settings);
}
```

- + Mỗi đơn hàng khách hàng có thể lựa chọn nhiều địa chỉ và khung giờ khác nhau, vì thế khi đã chọn được địa chỉ và khung giờ phù hợp thì các địa chỉ còn lại là không cần thiết nên tui em chuyển trạng thái các địa chỉ đó thành false qua hàm updateStatus:

```
function updateStatus(iddonhang) {
  let dataPut = {
    laMacDinh: false
  }
  let settings = {
    method: "PUT",
    body: JSON.stringify(dataPut),
    headers: { 'Content-Type': 'application/json' },
  };
  return fetch('https://servertlcn.herokuapp.com/diachi/' + iddonhang + '/update', settings);
}
```

b. DataOutTime:

- + Với dataOutTime thì việc đầu tiên cần làm là kiểm tra xem các đơn hàng bị liệt kê trong dataOutTime có còn địa chỉ và khung giờ khác không, nếu còn thì đơn hàng sẽ được xóa khỏi dataOutTime:

```
const outTimeCheck = dataOutTime.map(async (e) => {
  let check = await checkOthersAddress(e.donhang.id, e.address.id)
  if (check) {
    let index = dataOutTime.indexOf(e)
    dataOutTime.splice(index, 1);
  }
})
const outTimeCheckPromises = await Promise.all(outTimeCheck);
```

- + Hàm checkOtherAddress: Kiểm tra xem đơn hàng còn địa chỉ và khung giờ sau khung giờ hiện tại đang xét hay không

```
async function checkOthersAddress(iddonhang, iddiachi) {
  let settings = {
    method: "GET",
  };
  let dataFetch = await fetch('https://servertlcn.herokuapp.com/diachi/' + iddonhang + '/donhang', settings)
  dataFetch = await dataFetch.json();
  var date1;
  var date2;
  var time1;
  var time2;
  var flag = false;
  if (dataFetch.data.length > 1) {
    for (var e in dataFetch.data) {
      if (dataFetch.data[e].id == iddiachi) {
        date1 = new Date(dataFetch.data[e].ThoiGianKetThuc).getDate()
        time1 = dataFetch.data[e].ThoiGianKetThuc.split("T")[1].replace("Z", "").substr(0, 5);
      }
      date2 = new Date(dataFetch.data[e].ThoiGianBatDau).getDate()
      time2 = dataFetch.data[e].ThoiGianBatDau.split("T")[1].replace("Z", "").substr(0, 5);
      if (date2 - date1 > 0) {
        flag = true; break;
      }
      if (timediff(time1, time2)) {
        flag = true; break;
      }
    }
  }
  return flag;
}
```

- + Sau cùng là làm lại các bước như dataInTime (nếu dataOutTime còn phần tử sau khi đã xem xét ở bước trên):

```

if (dataOutTime.length > 0) {
  let dataPostChuoiOutTime = {
    Chuoi: JSON.stringify({ chuoidonhang: dataPostChuoiOutTime }),
    SoLuong: dataOutTime.length
  }
  let chuoiDataOutTime = await createChuoi(dataPostChuoiOutTime);
  chuoiDataOutTime = await chuoiData.json();
  const outTime = dataInTime.map(async (e) => {
    let dataThongbao = {
      NoiDung: 'Đơn hàng ' + e.donhang.id + ' sẽ được giao đến địa chỉ: ' +
        e.address.TenDiaChi + ' vào khung giờ (' + e.address.TimeRange + ')',
      Type: 'giao hang',
      NguoiDungId: e.reciver.id
    }
    await createThongBao(dataThongbao);
    await updateDonhang(e.donhang.id, e.address.id, chuoiDataOutTime.data.id);
    await updateStatus(e.donhang.id);
  })
  const outTimePromises = await Promise.all(outTime);
}

```

2/ Kết quả:

a. Thông báo:

```

{
  "id": 3,
  "NoiDung": "Đơn hàng 6 sẽ được giao đến địa chỉ: 26 Chương Dương, Linh Chiểu, Thủ Đức, Thành phố Hồ Chí Minh, Vietnam vào khung giờ (07:00 - 11:00)",
  "Type": "giao hang",
  "isNew": true,
  "NguoiDungId": 2
},
{
  "id": 4,
  "NoiDung": "Đơn hàng 2 sẽ được giao đến địa chỉ: Võ Văn Ngân, Thành phố Hồ Chí Minh, Vietnam vào khung giờ (08:00 - 11:00)",
  "Type": "giao hang",
  "isNew": true,
  "NguoiDungId": 1
},

```

b. Đơn hàng:

+ Đơn hàng 6 (Người dùng 2):

```

{
  "result": "ok",
  "data": {
    "id": 6,
    "NgayDatHang": "2020-11-17T02:36:45.069Z",
    "TienVanChuyen": "100",
    "TongTien": "100100",
    "TinhTrangDon": null,
    "NguoiDungId": 2,
    "BuuCucId": null,
    "ChuoigiaoHangId": 1,
    "DiaChiId": 9,
    "GhiChu": "",
    "DanhGia": "",
    "daThanhToan": null
  },
  "message": "List DonHang successfully"
}

```

+ Đơn hàng 2 (Người dùng 1):

```

{
  "result": "ok",
  "data": {
    "id": 2,
    "NgayDatHang": "2020-11-17T02:09:32.368Z",
    "TienVanChuyen": "100",
    "TongTien": "30100",
    "TinhTrangDon": null,
    "NguoiDungId": 1,
    "BuuCucId": null,
    "ChuoigiaoHangId": 1,
    "DiaChiId": 4,
    "GhiChu": "",
    "DanhGia": "",
    "daThanhToan": null
  },
  "message": "List DonHang successfully"
}

```

c. Địa chỉ:

+ Địa chỉ 9 (Đơn hàng 6, Người dùng 2):

```
{
  "result": "ok",
  "data": {
    "id": 9,
    "TenDiaChi": "26 Chương Dương, Linh Chiểu, Thủ Đức, Thành phố Hồ Chí Minh, Vietnam",
    "KinhDo": "106.7603852",
    "ViDo": "10.8522347",
    "ThoiGianBatDau": "2020-11-20T07:00:00.000Z",
    "ThoiGianKetThuc": "2020-11-20T11:00:00.000Z",
    "laMacDinh": false,
    "NguoiDungId": 2,
    "DonhangId": "6"
  },
  "message": "List DiaChi successfully"
}
```

+ Địa chỉ 4 (Đơn hàng 2, Người dùng 1):

```
{
  "result": "ok",
  "data": {
    "id": 4,
    "TenDiaChi": "Võ Văn Ngân, Thành phố Hồ Chí Minh, Vietnam",
    "KinhDo": "106.7645936",
    "ViDo": "10.8507321",
    "ThoiGianBatDau": "2020-11-20T08:00:00.000Z",
    "ThoiGianKetThuc": "2020-11-20T11:00:00.000Z",
    "laMacDinh": false,
    "NguoiDungId": 1,
    "DonhangId": "2"
  },
  "message": "List DiaChi successfully"
}
```

3. Gán id Shipper vào chuỗi đơn hàng

- Fetch tất cả các chuỗi đơn hàng về app: fetch data từ api <https://servertlen.herokuapp.com/chuoiigiaohang/>

-

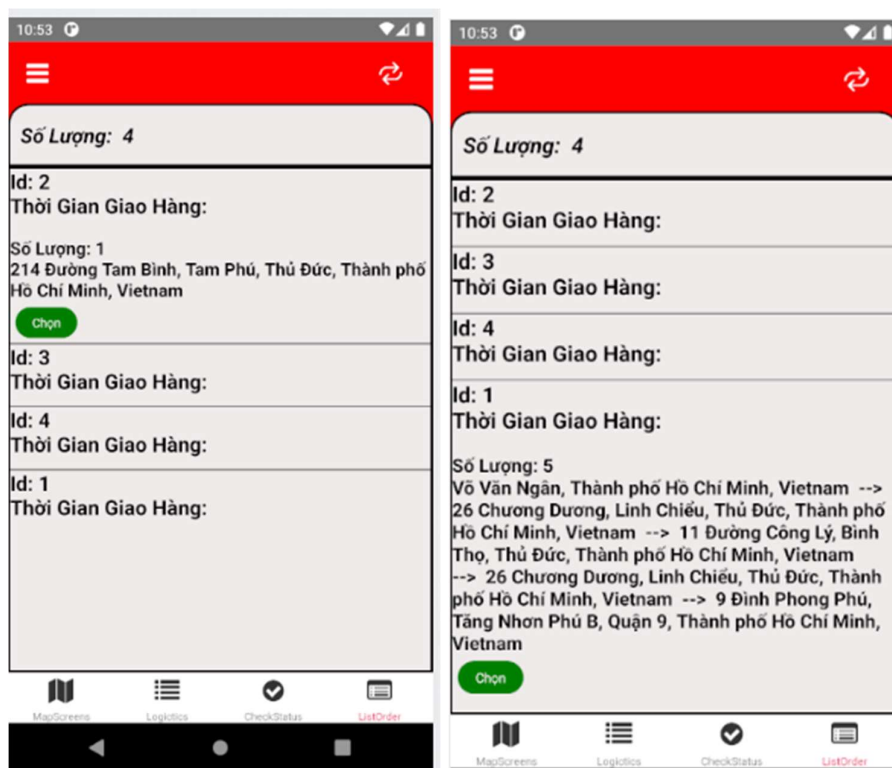

```

}
fetchData() {
  ...
  return fetch('https://server1cn.herokuapp.com/chuoigiaohang/',
    { method: 'GET' })
    .then(async (responseJson) => {
      responseJson = await responseJson.json()
      if (responseJson.result !== "failed") {
        let data = responseJson.data
        //data = await JSON.parse(data)
        //console.log(data)
        if (this._isMounted) {
          this.setState(
            {
              isLoading: false,
              dataSource: data,
              soluong: data.length
            })
        }
      }
      else {
        this.setState(
          {
            isLoading: false,
          })
      }
    })
    .catch((error) => {
      console.log(error);
    });
}
}

```



- Shipper dựa vào địa chỉ và số lượng đơn hàng để chọn chuỗi đơn hàng muốn giao



- Shipper ấn chọn để chọn đơn hàng, app check xem shipper đó đã đăng ký chuỗi nào trước đó mà chưa hoàn thành chưa, nếu có thì ko cho đăng ký tiếp, nếu chưa thì tiến hành put id shipper vào chuỗi đơn hàng đã chọn

```

let data = {
  ShipperId: idShipper
}
fetch('https://servertlcn.herokuapp.com/chuoigiaohang/' + idShipper + '/shipper',
  { method: 'GET' })
  .then(async (responseJson) => {
    responseJson = await responseJson.json()
    let a = responseJson.data
    if (responseJson.result !== "failed") {
      Alert.alert("Bạn Đã Đăng Ký Chuỗi Trước Đó")
    }
    else {
      fetch('https://servertlcn.herokuapp.com/chuoigiaohang/' + section.id,
        {
          method: 'PUT',
          body: JSON.stringify(data),
          headers: { 'Content-Type': 'application/json' }
        })
        .then((response) => response.json())
        .then((responseJson) => {
          console.log(responseJson)
          store.dispatch({
            type: 'ADDORDER',
            payload: chuoi
          })
          store.dispatch({
            type: 'ADDCHUOI',
            payload: section.id
          })
        })
      .catch((error) => {

```



- Sau khi shipper đã đăng ký thì trong danh sách địa chỉ sẽ có địa chỉ các đơn

