1. **Đăng ký chuyến đi**

**\*Đồ thị dòng điều khiển**

const idneed = req.body.tripId;

const testTrip = await Trip.findOne({ where: { id: idneed } });

let testAccountId = req.body.accountId;

if (!testTrip || testAccountId === null) {

res.send(404);

} else {

let valueRegisterTrip = new RegisterTrip({

nameRegister: req.body.nameRegister,

...

});

let soLuongDangKi = valueRegisterTrip.adults + valueRegisterTrip.children + valueRegisterTrip.baby;

if (testTrip.theRemainingAmount >= soLuongDangKi) {

valueRegisterTrip.save()

.then((data) => {

Trip.update({

theRemainingAmount: testTrip.theRemainingAmount - soLuongDangKi

}, { where: { id: testTrip.id } });

res.json(data);

})

.catch((error) => res.status(400).send({

error: error.message

}));

} else {

res.status(400).send({

error: 'Số lượng không đủ để đăng ký.'

})

}

}

Đồ thị bên có 3 nút quyết định nhị phân nên có độ phức tạp C = 3+1 = 4.

4 đường thi hành tuyến tính độc lập cơ bản là :

1. 1→2→3

2. 1→2→4→5→6→7

3. 1→2→4→5→6→8

4. 1→2→4→5→9

* Test case cho đường 1:

Kết quả kỳ vọng : testTrip = null;

* Test case cho đường 2:

Kết quả kỳ vọng : testTrip != null;

testTrip.theRemainingAmount >= soLuongDangKi

Save RegisterTrip thành công.

* Test case cho đường 3:

Kết quả kỳ vọng : testTrip != null;

testTrip.theRemainingAmount >= soLuongDangKi

Save RegisterTrip thất bại.

* Test case cho đường 4:

Kết quả kỳ vọng : testTrip != null;

testTrip.theRemainingAmount < soLuongDangKi

* Thông báo: 'Số lượng không đủ để đăng ký.'

**\*Đồ thị dòng dữ liệu**

1. **Login**

**\*Đồ thị dòng điều khiển**

const {

username,

password

} = req.body;

const account = await Account.findOne({

where: {

username: username

}

});

if (account && comparePassword(password, account.password)) {

const accountCustomer = await Account.findOne({

include: [{

model: models.Customer,

where: { accountId: account.id }

}]

});

const accountTourGuide = await Account.findOne({

include: [{

model: models.TourGuide,

where: { accountId: account.id }

}]

});

if (accountCustomer) {

res.status(200).send(accountCustomer);

}

if (accountTourGuide) {

res.status(200).send(accountTourGuide);

}

else {

res.status(404).send('Not Found');

}

}

else {

res.status(404).send('Not Found');

}

Đồ thị bên có 3 nút quyết định nhị phân nên có độ phức tạp C = 4+1 = 5.

5 đường thi hành tuyến tính độc lập cơ bản là :

1. 1→2→3→4→5→6

2. 1→2→10

3. 1→2→3→10

4. 1→2→3→4→5→7→8

5. 1→2→3→4→5→7→9

* Test case cho đường 1:

Kết quả kỳ vọng : account != null && comparePassword(password, account.password) = 1;

accountCustomer = true;

* Test case cho đường 2:

Kết quả kỳ vọng : account = null;

* Test case cho đường 3:

Kết quả kỳ vọng : account != null && comparePassword(password, account.password) = 0;

* Test case cho đường 4:

Kết quả kỳ vọng : account != null && comparePassword(password, account.password) = 1;

accountCustomer = false;

accountTourGuide = true;

* Test case cho đường 5:

Kết quả kỳ vọng : account != null && comparePassword(password, account.password) = 1;

accountCustomer = false;

accountTourGuide = false;

* Thông báo: ‘Not Found’

Quy ước tên biến

|  |  |
| --- | --- |
| a | username |
| b | password |
| c | account |
| d | comparePassword(password, account.password) |
| e | accountCustomer |
| f | accountTourGuide |

1. **Đăng ký**

**\*Đồ thị dòng điều khiển**

let valueAccount = new Account({

username: req.body.username,

… });

var salt = bcrypt.genSaltSync(10);

var hash = bcrypt.hashSync(valueAccount.password, salt);

valueAccount.password = hash;

valueAccount.save().then((data) => {

res.status(200).send(data);

let valueCustomer = new Customer({

name: req.body.fullname,

… });

let valueTourGuide = new TourGuide({

name: req.body.fullname,

…. });

if (valueAccount.isCustomer == true) {

valueCustomer.save().then(async listDocCustomer => {

res.status(200).send(listDocCustomer);

console.log('Customer');

var req = {

username: valueAccount.username,

…

};

await sendEmail(req, res, next);

});

}

if (valueAccount.isTourGuide == true) {

valueTourGuide.save().then(async listDocTourGuide => {

res.status(200).send(listDocTourGuide);

console.log('TourGuide');

var req = {

username: valueAccount.username,

…

};

await sendEmail(req, res, next);

});

} else {

res.send(error);

}

});

Đồ thị bên có 3 nút quyết định nhị phân nên có độ phức tạp C = 2+1 = 3.

3 đường thi hành tuyến tính độc lập cơ bản là :

1. 1→2→3→4→5→6

2. 1→2→3→4→7→8→9

3. 1→2→3→4→7→10

* Test case cho đường 1:

Kết quả kỳ vọng :

valueAccount.isCustomer = true;

* Test case cho đường 2:

Kết quả kỳ vọng :

valueAccount.isCustomer = false;

valueAccount. isTourGuide = true;

* Test case cho đường 3:

Kết quả kỳ vọng :

valueAccount.isCustomer = false;

valueAccount. isTourGuide = false;

Quy ước tên biến

|  |  |
| --- | --- |
| a | valueAccount |
| b | salt |
| c | hash |
| d | valueCustomer |
| e | valueTourGuide |