

Classes and interfaces :-

- 1) Bank
- 2) Branch
- 3) BankAccount
- 4) Savings Account
- 5) CurrentAccount
- (6) Customer
- (7) Address
- (8) Transaction

Relationships between Entities

Branch — HAS-A — Bank

SavingsAccount — IS-A — BankAccount

CurrentAccount — IS-A — BankAccount

BankAccount — HAS-A — Customer

Transaction — HAS-A — BankAccount

[Optional] Relationships of Bank, Branch & Customer with
the Address class — HAS-A

Class Descriptions

Bank: `<optional List<Branch>>`

Bank:

- ✓ `Integer bankCode;`
- `String bankName;`
- `Address mainOfficeAddr;`
- `//constructors`
- `//getters & setters`
- `//override toString`
- `//override compareTo`

Branch:

- `4 Integer branchCode;`
- ✓ `4) Integer bankCode;`
`or`
`Bank bank;`
- `3) String manager;`
- `4) Address address;`
- `//constructors, getters, setters,`
- `toString, compareTo, equals,`
- `hashCode.`

Address:

String street;

String city;

String state;

String country;

Integer pincode.

/ other statutories

* can be a string too

Customer:

String name;

* Address address;

Integer customerId;

//other statutories

Transaction:

```
Integer accountNo;  
OR  
BankAccount account;  
enum("credit", "debit") type;  
Date date;  
double amount;  
// other statutories
```

BankAccount (abstract):

```
1) Integer accountNo;  
2) Integer customerId;  
OR  
Customer customer;  
3) Integer branchCode;  
OR  
Branch branch;  
enum("saving", "current") type;  
// other statutories  
// methods from diagram
```

SavingsAccount

and CurrentAccount

Refer to Core Java discussions #14.

Additional fields to consider:-

- i) In BankAccount, a List<Transaction>
- ii) In Customer, a List<BankAccount>
- iii) In Branch, a List<BankAccount>
- iv) In Bank, a List<Branch>

Controllers

1) HomeController :-

index() - displays the landing page with a navigation bar.

Shows a list of all banks in a sidebar (clickable)

The navbar will have a "Customer Portal" button which will take you to CustomerController#index

2) Transaction Controller

deposit() - GET (accountNo(fv), amt(qP))

withdrawl() - POST (accountNo, amt)

admin - showTransactions(optional<startDate>, optional<endDate>)

CustomerController

index() - shows a form to enter customerId
welcome() - GET (customerId(PV))



- Welcome message with <name>
<options> including

viewProfile() - GET (customerId(PV))

updateProfile() - PATCH (name and/or address from a form)

editProfileForm() - GET (show a form populated with
customer details)
└ c/o/e

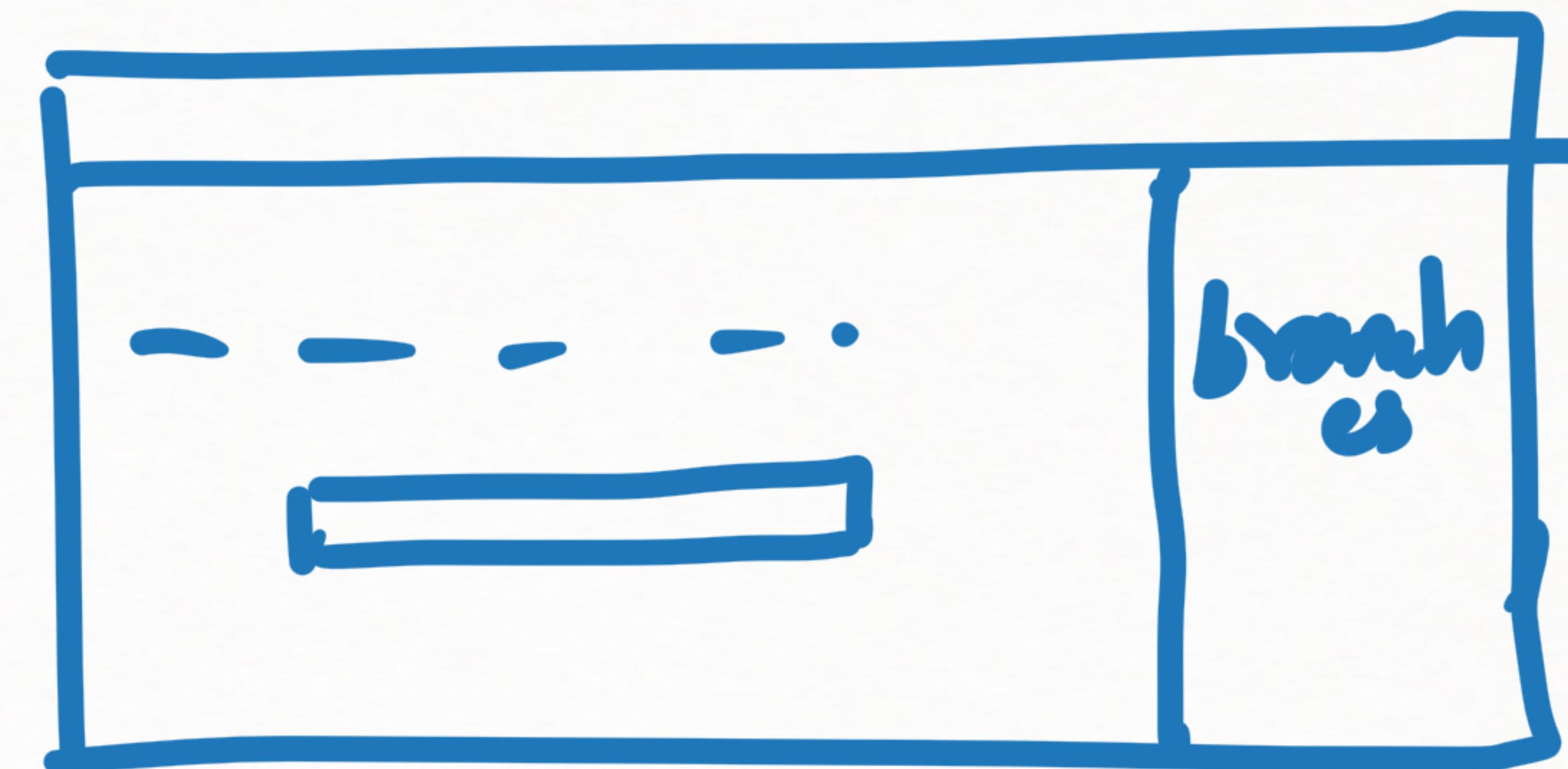
`create()` - admin/employee

`delete()` - DELETE - `(customerId(PV))` - c/a

`showTransactions()` - (optional <startDate>,
optional <endDate>)

Bank Controller

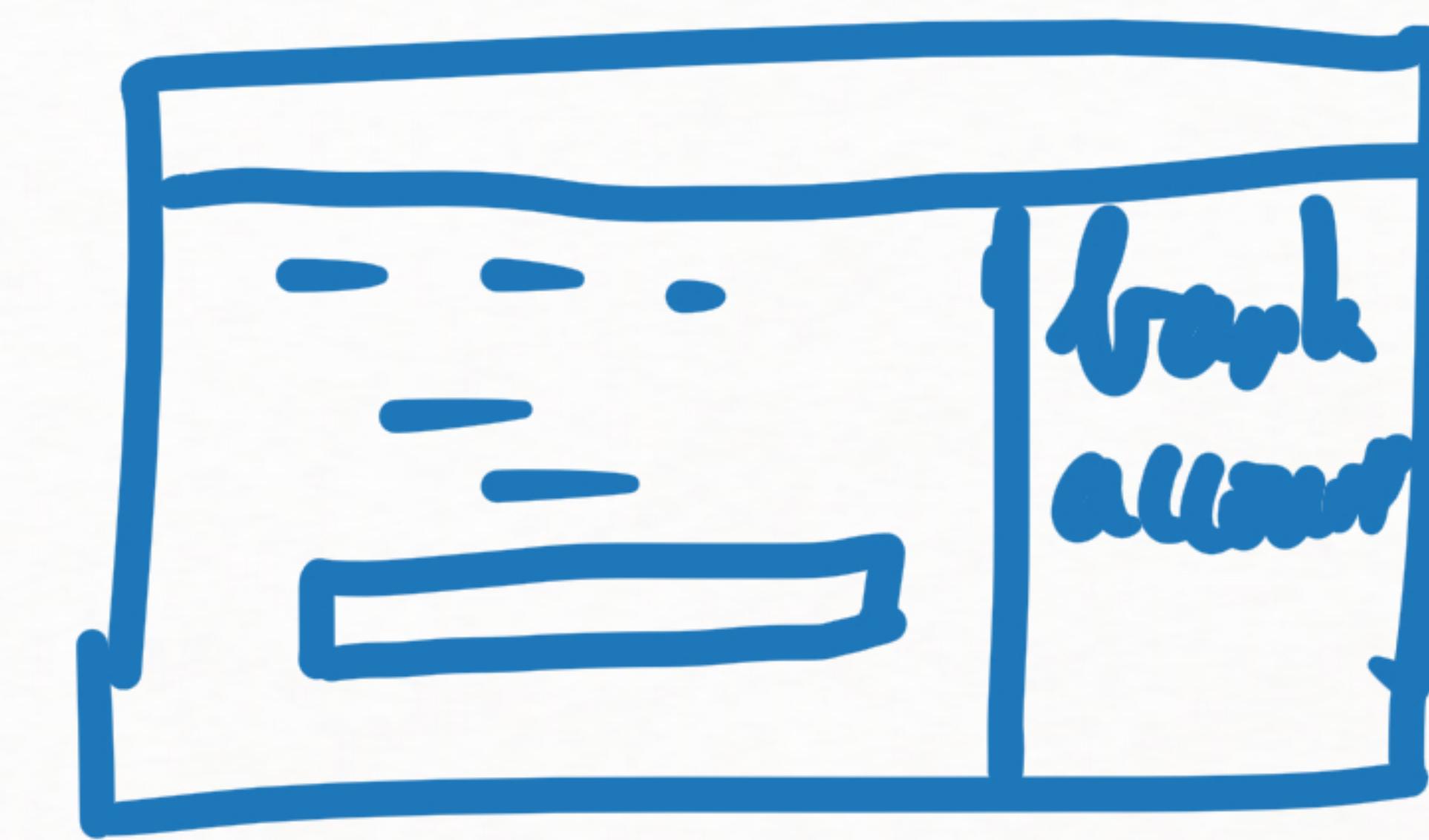
create()
update() < GET
delete()
viewDetails()



BranchController

both - view()
admin {
 delete()
 create()
 update()
}

can be done by both admin
& employees of that branch
(branchrole)



both - showTransactions (optional<startDate>, optional<endDate>)
both - showCustomers (optional<accountType>)

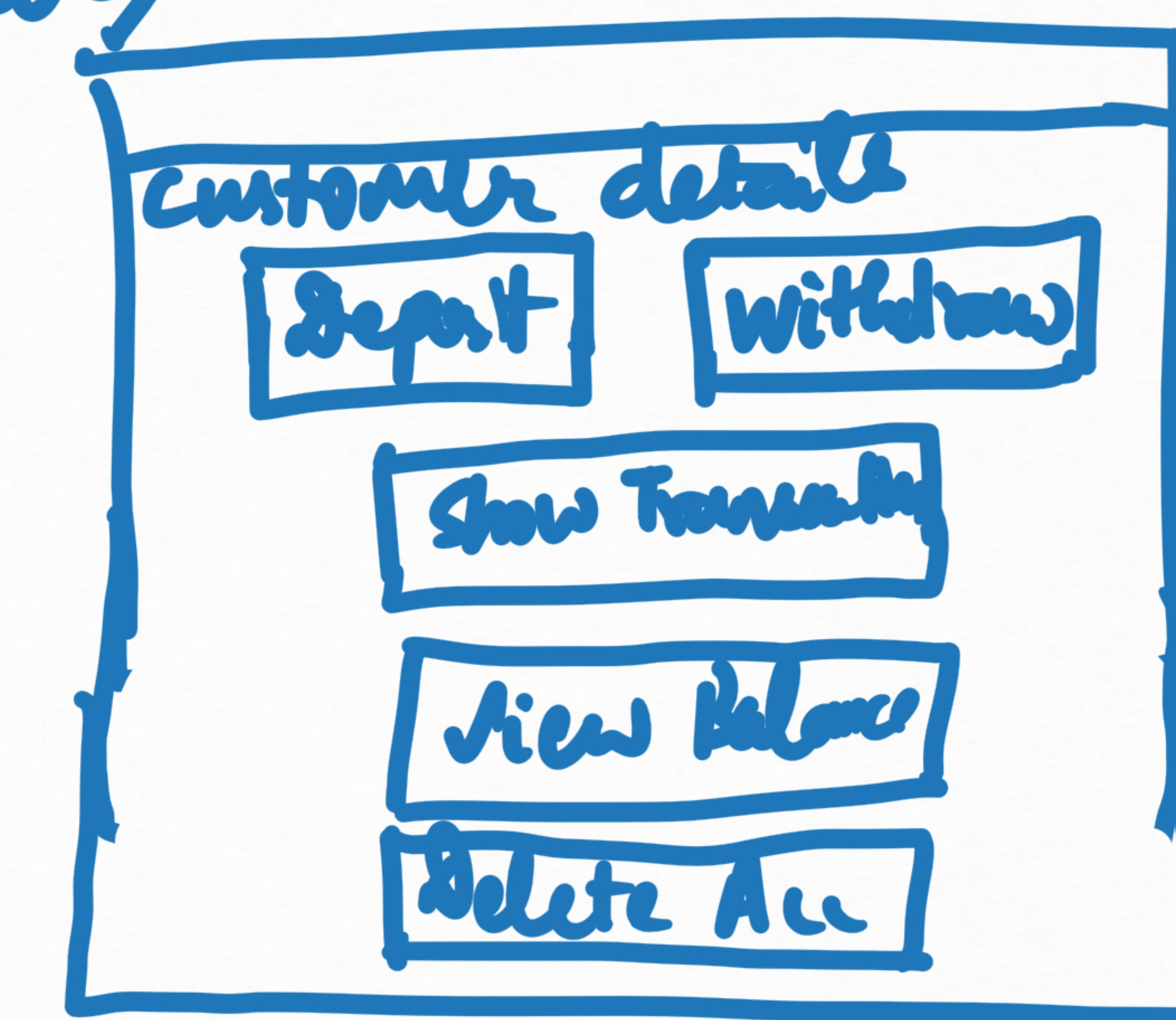
Bank Account Controller

L (employee)

R - viewDetails() - details

U (employee)

D (employee/customer)



Reversal Request Controller

user - create() - (transactionId)

admin - showAll()

admin - delete()

admin - approve(): it creates an opposite type
of transaction with the same amt
to the same account (calls
methods of Transaction Controller)

EmployeeController

AdminController

- View Profile (Admin) - showProfile()
- showTransactions()
 - calls the showAll() method of TransactionService with the bankCode
 - ✓ redirect the ^{OR} request to bank/{bankCode}/transactions/all

→ showCustomers()

↳ calls the showCustomers(bankCode) method of the BankService class with the bankCode of the admin, which in turn calls showAll(bankCode) of the CustomerService class.

or
redirects the request to :-

/banks/{bankCode}/customers/all

→ same for showBranches

approveReversal) - calls 'approve' method of RRC

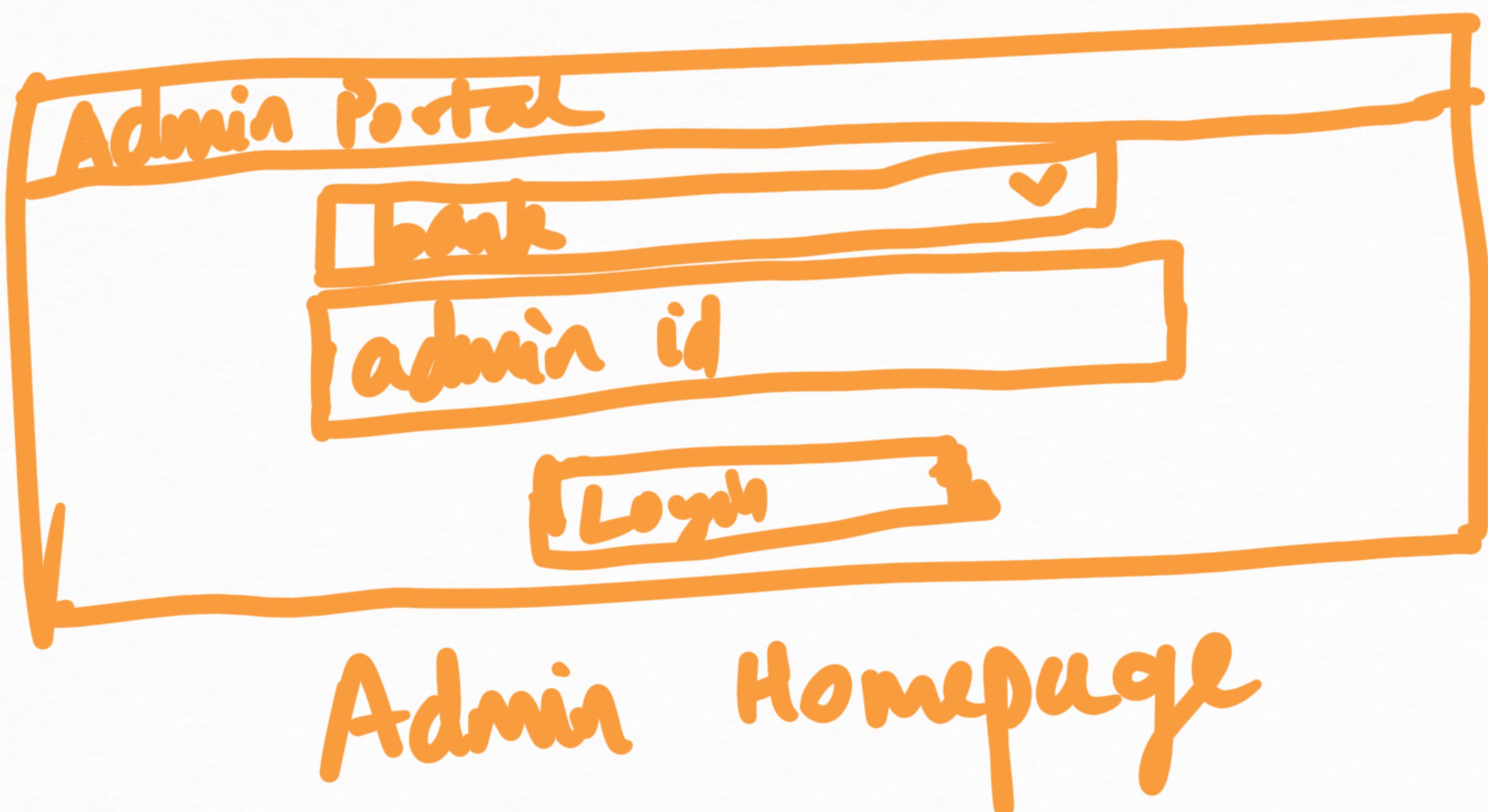
(rejectReversal) - calls 'delete()' method of RRC .

Frontend Pages

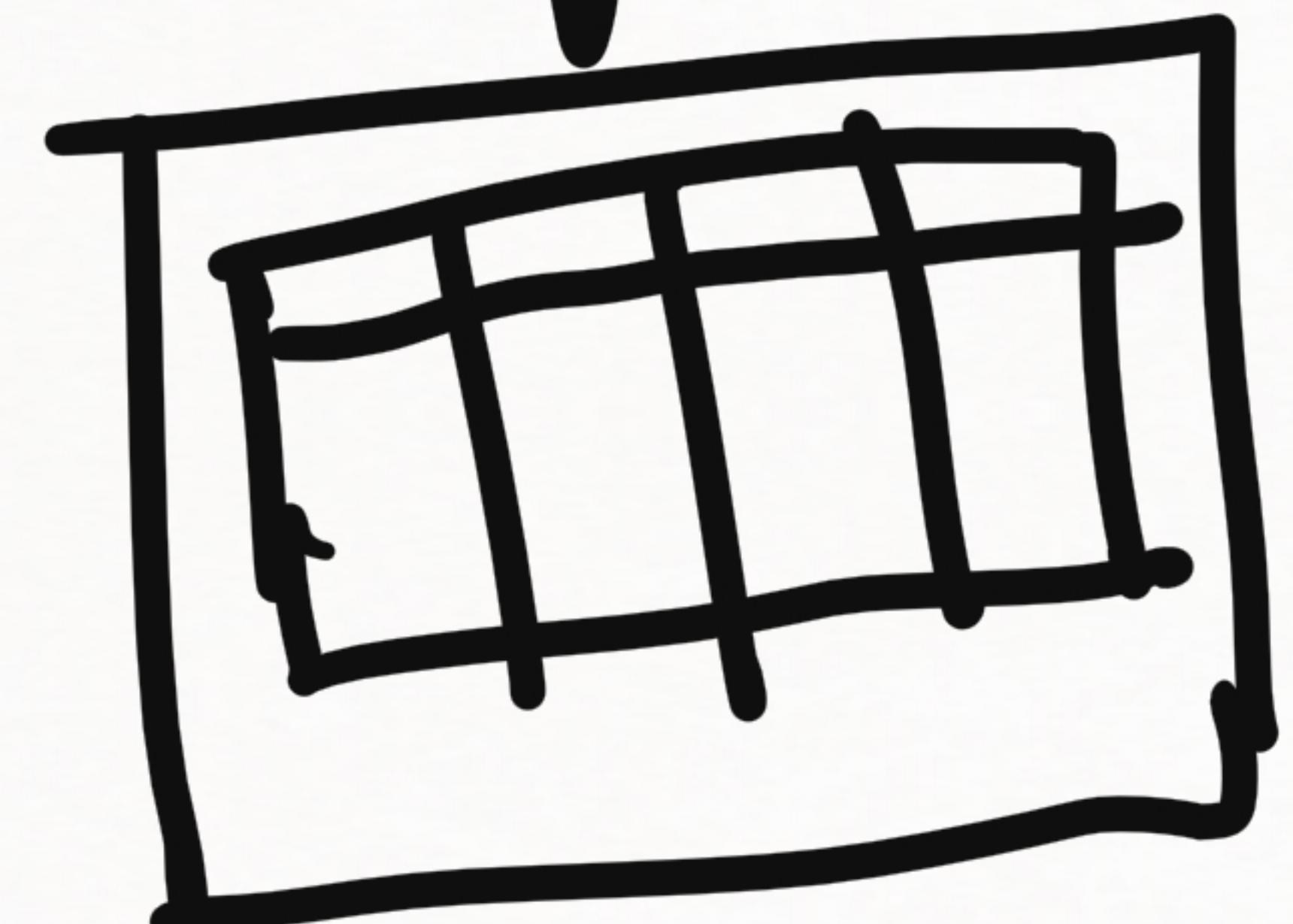
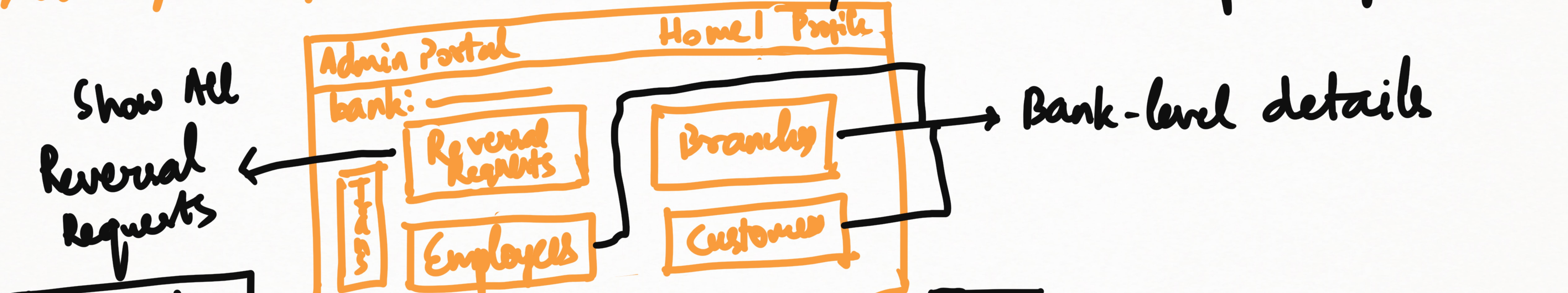
1) Admin Pages

at localhost:8080/admin/index

/admin



/bank/{code}/admin/{id}/dashboard



(datatable)

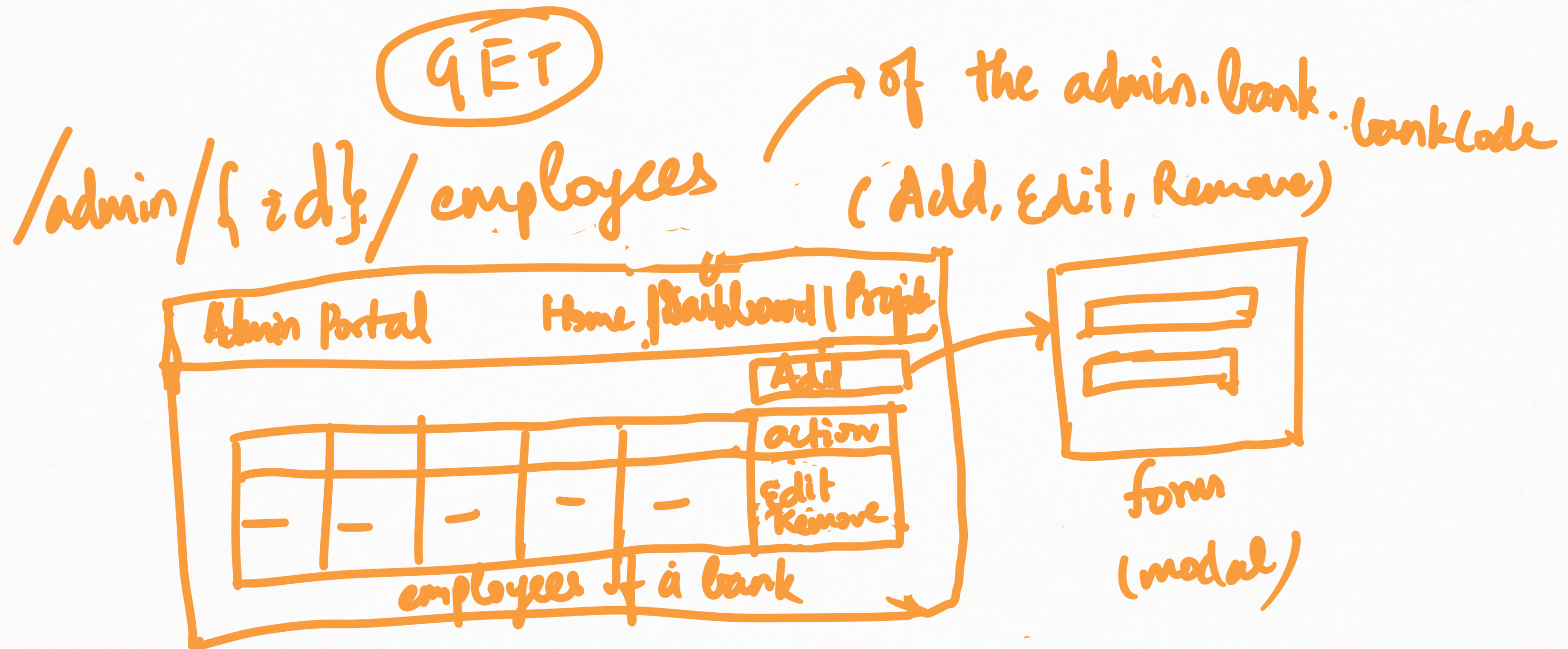
Admin Dashboard

Deny(X) - remove the request from the table

Approve(V) = make a
opposite transaction to the account

req date	transactionNo	date	type	action
-	-	-	-	<input checked="" type="checkbox"/>

Reversal Requests



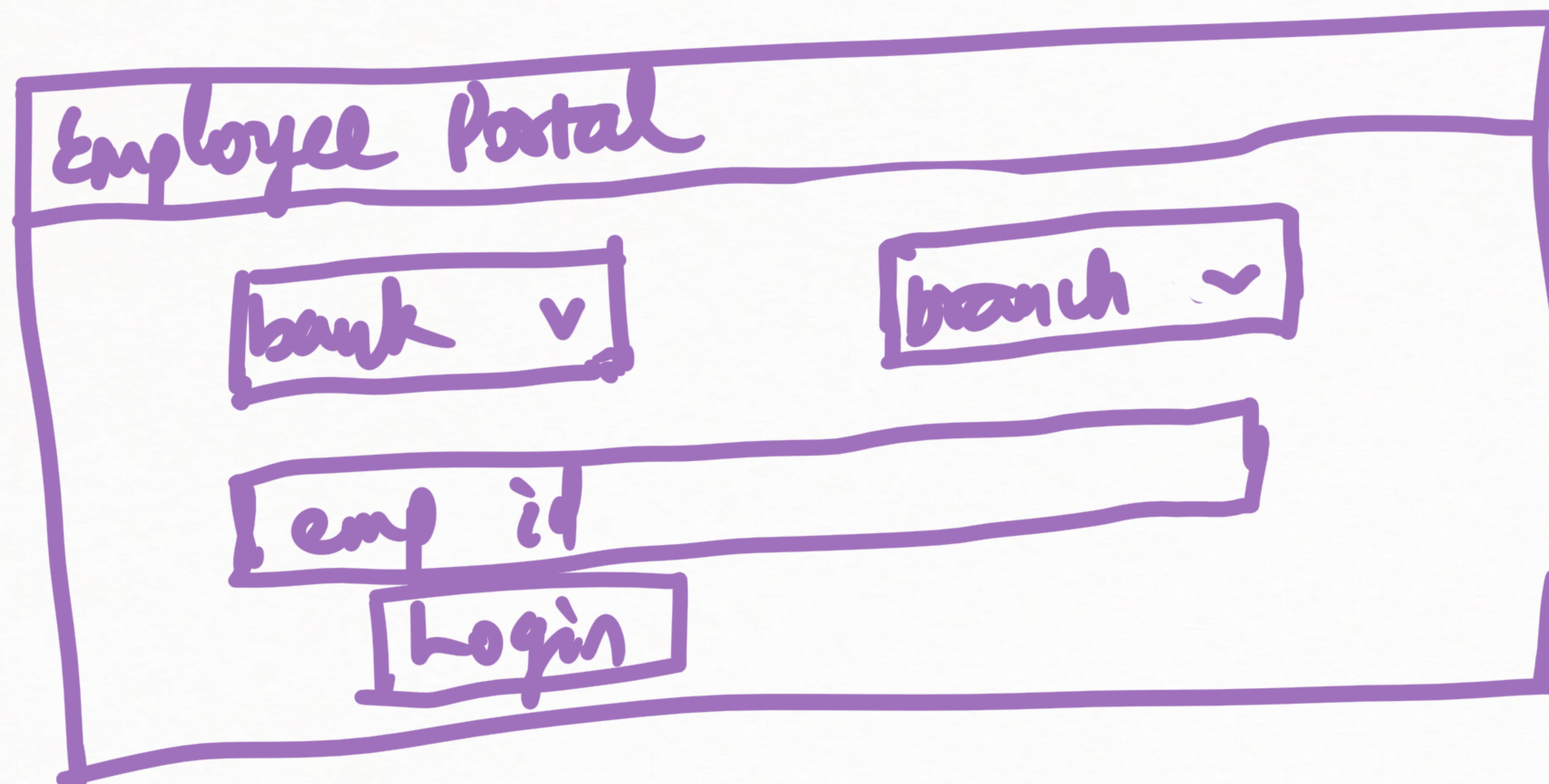
/admin/{id}/customers

/admin/{id}/branches **(Add, Edit, Remove)**

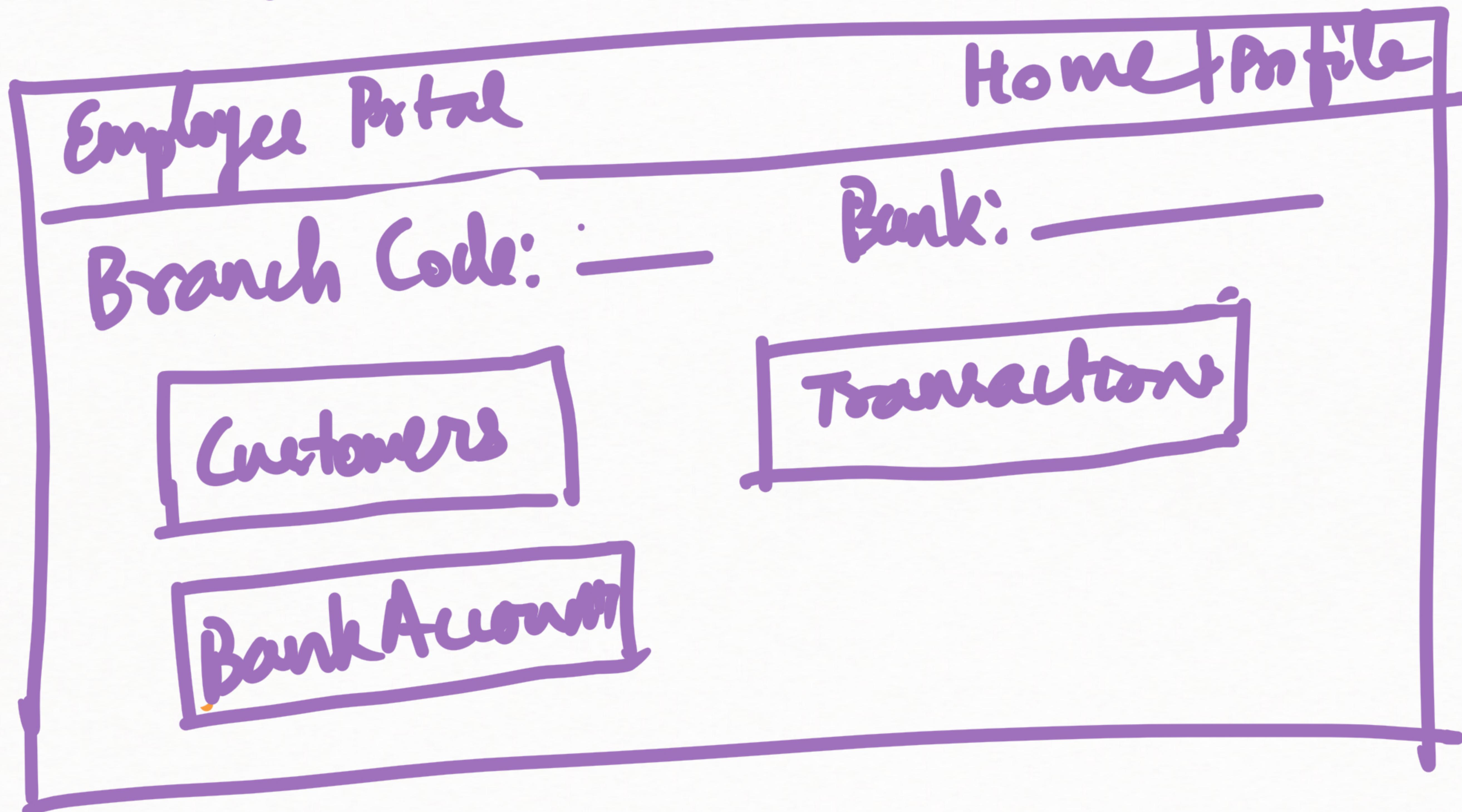
/admin/{id}/transactions


2) Employee Pages

localhost : 4040 / employee / index



/branch/{url}/employee/{id}/dashboard





Same as previous for

Bank Accounts (Add, Edit, Remove, View)

Transaction (View)

View Profile/Edit Profile

3) Customer Pages

/customer/index

bank v

accNb

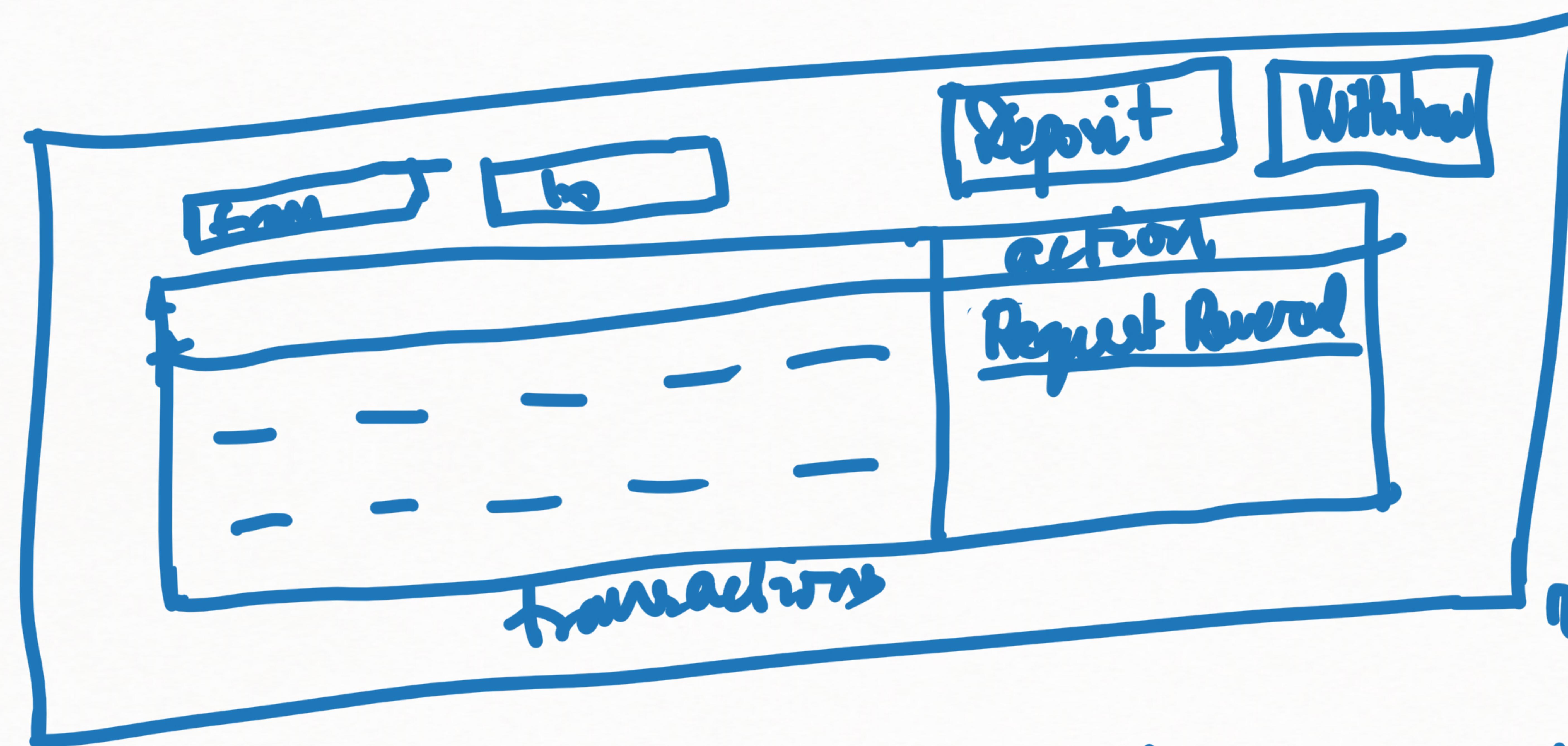
Login

/customer/{id}/dashboard

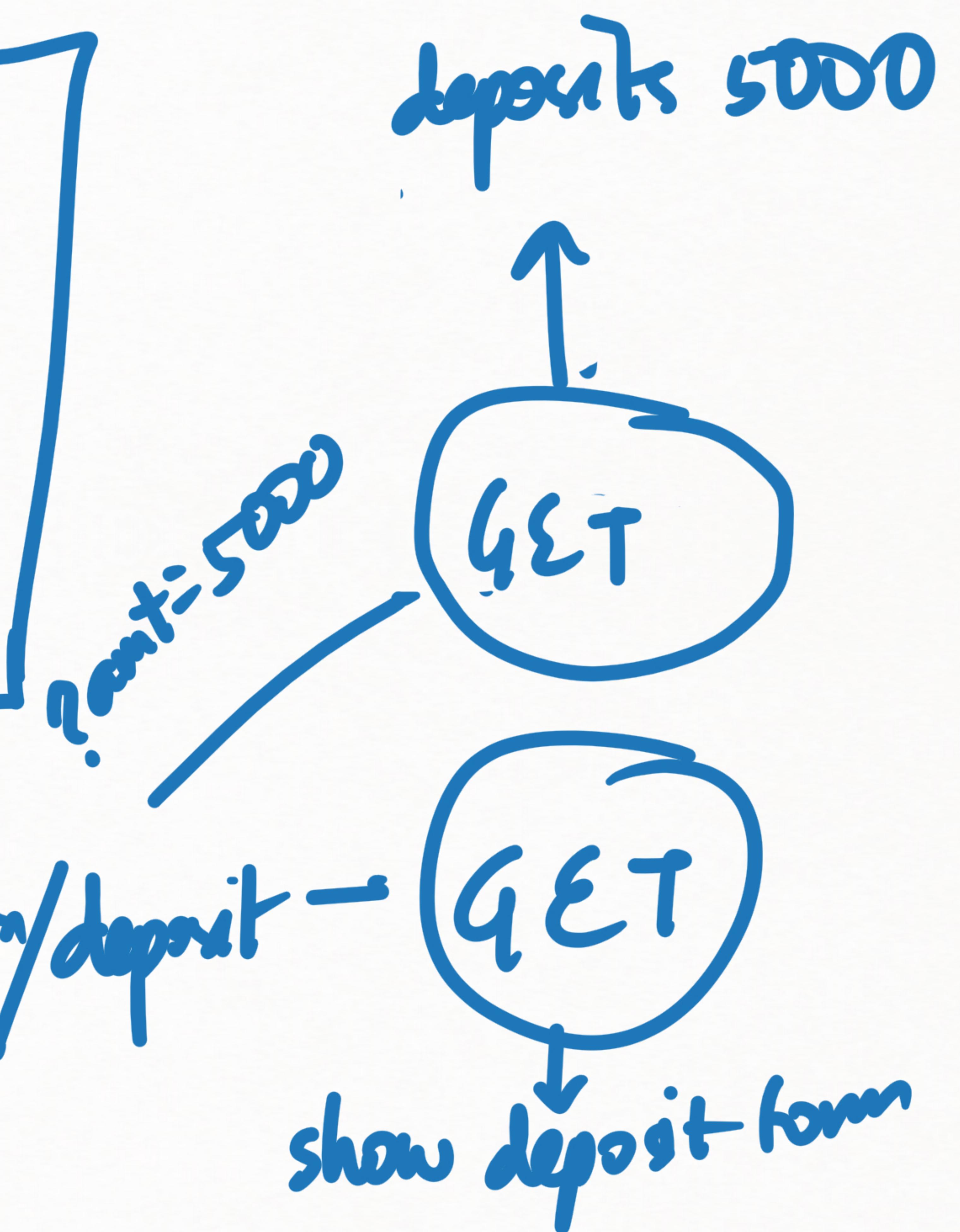
- View Transactions
- Profile (Editable, navbar)
- Deposit
- Withdraw

customer

/customer/{id}/transactions



/customer/{id}/acc/{accNo}/transaction/deposit - GET



/customer/{id}/bankAccount/{accNo}/withdraw

↳ GET :- shows withdrawal form

↳ POST :- {amt = 5000}) - withdraws 5000