Main Application Structure on SERVER SIDE

App: This is the main structure that has the public static void main.

> Controller Package Folder

AuthenticationController

AccountController

TransferController

UserController

> DAO Package Folder

AccountDAO: Interface for accessing account data

JdbcAccountDAO: accessing postgreSQL

TransferDAO

JdbcTransferDAO

UserDAO

JdbcUserDAO

> Exception Package Folder

DAO Exception

> Model Package Folder

> Main

Account: Contains Account ID, User ID and Balance

Transfer: Contains Transfer ID, Transfer Type, Status, From account, To account, Amount

> Security

Authority

LoginDto: This is a login endpoint for user

LoginResponseDto: Allows to verify if user has token

RegisterUserDto

User: username, password, role

> Security

> jwt

JWTConfigurer

JWTFilter

TokenProvider

JwtAccessDeniedHandler

JwtAuthenticationEntryPoint

SecurityUtils

UserModelDetailsService

UserNotActivatedException

WebSecurityConfig

> Test Folder

> java

>BaseDaoTests

>JbdcUserDaoTests

>JbdcAccountDaoTests

>JbdcTransferDaoTests

>TestingDatabaseConfig (edited)

**# Outline for Client Side**

> App : main application (public static void main(String[] args))

> Model Package Folder

> Security

> AuthenticatedUser

> User

> UserCredentials

> Account

> Transfer

> Services Package Folder

> Security

> AuthenticatedApiService

> AuthenticatedService

> UserService: username, password, role

> Console Service:

> Account Service: account\_id, balance,

> Transfer Service: Transfer type, transfer status,

>Test Package Folder

> AccountServiceTests

> TransferServiceTests

> AuthenticationServiceTests

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Service Design Outline

1. **Controller Package**
2. **Service Package**: It interacts with DAOs to perform operations on the database and processes the data received from the controllers.
3. **DAO Package**
4. **Model Package**: Represents the data structure of the application. Models are used to transfer data between different layers.
5. **Security Package**: Manages authentication and authorization. Includes JWT components for token management, user details services, and security configuration.
6. **Exception Handling**: (Makes it easier instead of retyping DAO exception code)
7. **Testing**: Includes unit tests for DAO and service package

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To-Do List to adjust code

1. Current account balance shows null when it should be $1000.00 (or a number)
   1. I ran through debug, and I realized that when we declared Acount balance = new Account(); It’s empty and has no information to pull from. We declare Account balance as the current user with their id.
   2. I was able to update AccountService to get current balance. But now I am getting an error that is logged. I think our endpoints aren’t correct and we aren’t getting the right information.
2. Add a GetAllTransfers to TransferControlle
3. View pending requests gives this error
4. When sending or requesting money, it should list users and their id. I got it to print that, but now it just says no other users.
5. When sending money, it requested the amount and we got an error message