

# PROJECT ZERO

## JUnit Testing

### Overall

Covered 4,674 Instructions of 12,211 (38.3%)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
PhilipLawrence0Bank-Maven	38.3 %	4,674	7,537	12,211
src/main/java	30.4 %	3,177	7,290	10,467
com.revature.bank.dao.implementations	33.0 %	1,263	2,566	3,829
com.revature.bank.model	15.5 %	334	1,816	2,150
com.revature.bank.services.menus	13.0 %	203	1,362	1,565
com.revature.bank.resources	0.0 %	0	594	594
com.revature.bank.services.handlers	33.5 %	189	375	564
com.revature.bank.services.buisness.logic	69.7 %	657	285	942
com.revature.bank.exceptions	15.5 %	32	175	207
com.revature.bank.dao	54.1 %	112	95	207
com.revature.bank	0.0 %	0	15	15
com.revature.bank.services.helpers	98.2 %	387	7	394
src/test/java	85.8 %	1,497	247	1,744

## Services

### Critical

Methods where errors will cause security vulnerabilities or cascading null reference pointers throughout the code.

Covered 387 of 394 (98.2%)
















com.revature.bank.services.helpers	98.2 %	387	7	394
LoggerSingleton.java	66.7 %	14	7	21
AccountTypeHelper.java	100.0 %	77	0	77
MathHelper.java	100.0 %	49	0	49
MenuHelper.java	100.0 %	186	0	186
PermissionRankHelper.java	100.0 %	52	0	52
ScannerSingleton.java	100.0 %	9	0	9

The missing commands of the LoggerSingleton come from the Access Denied log added the final day of the sprint. As this is mission critical (an error here would leave us blind to an employee trying to force their way past our security), it needs to be tested in the next sprint, even though it is very simple and durable.

### Important

Methods that perform important business logic and need to be properly deployed as failure here can allow invalid data to be entered into the database.

Covered 657 of 942 instructions (69.7%)






























com.revature.bank.services.buisness.logic		69.7 %	657	285	942
>  ProcessAccountRequest.java		0.0 %	0	124	124
>  AccountService.java		0.0 %	0	94	94
>  AssociatePersonService.java		0.0 %	0	45	45
>  PermissionRankService.java		53.2 %	25	22	47
>  AccountRequestUserService.java		100.0 %	35	0	35
>  AccountTransactionService.java		100.0 %	486	0	486
>  PersonService.java		100.0 %	111	0	111

These functions are important and need full testing before deployment. The one hiccup during presentation would have been detected had ProcessAccountRequest been fully tested, however extreme time constraints led this function to being written functional first (with a small error) instead of TDD.






















## Luxury

Methods that improve user experience. Mistakes here will make sections of the program unavailable, but will not compromise data integrity

392 of 2129 instructions (18.4%)

com.revature.bank.services.menus		13.0 %	203	1,362	1,565
>  AccountMenu.java		0.0 %	0	315	315
>  RequestAccountsMenu.java		11.9 %	21	155	176
>  LoginMenu.java		0.0 %	0	137	137
>  MainMenu.java		4.5 %	6	127	133
>  ModifyUserMenu.java		8.3 %	11	122	133
>  ListAccountsMenu.java		15.0 %	18	102	120
>  NewPersonMenu.java		11.5 %	13	100	113
>  EmployeeMenu.java		0.0 %	0	84	84
>  ManageRequestedAccounts.java		0.0 %	0	72	72
>  ManageRequestedAccount.java		0.0 %	0	66	66
>  ListUsers.java		0.0 %	0	30	30
>  AbstractMenu.java		52.6 %	30	27	57
>  ListAssociatedAccountsMenu.java		19.4 %	6	25	31
>  CustomerMenu.java		100.0 %	98	0	98


Menu Functions are AbstractMenus that contain factories returning new menus used for navigation within the program.

com.revature.bank.services.handlers		33.5 %	189	375	564
>  AccountOwnershipHandler.java		28.8 %	21	52	73
>  PermissionRankHandler.java		34.2 %	25	48	73
>  AccountRequestUserHandler.java		28.8 %	17	42	59
>  TransactionHandler.java		0.0 %	0	41	41
>  AccountHandler.java		40.0 %	24	36	60
>  AccountRequestHandler.java		25.0 %	11	33	44
>  AccountTransactionHandler.java		46.7 %	28	32	60
>  AccountTypeHandler.java		37.3 %	19	32	51
>  AssociatedPersonHandler.java		36.7 %	18	31	49
>  PersonHandler.java		48.1 %	26	28	54



































Handler functions exclusively forward data back and forth to the DAO allowing a place where the code can choose to implement a handler with a different DAO if some data needs to be persisted in a separate database.

## DAO

1375 of 4036 Instructions (34.1%)

 com.revature.bank.dao	 54.1 %	112	95	207
>  DAOUtilities.java	 54.1 %	112	95	207

DAO utilities hold the connection method as well as lazy singleton getters for each DAOImpl to provide faster implementation.

 com.revature.bank.dao.implementations	 33.0 %	1,263	2,566	3,829
>  AccountDAOImpl.java	 33.6 %	182	360	542
>  PermissionRankDAOImpl.java	 30.0 %	148	346	494
>  AccountOwnershipDAOImpl.java	 27.8 %	133	345	478
>  AccountTransactionDAOImpl.java	 27.8 %	110	285	395
>  AccountRequestDAOImpl.java	 20.8 %	65	248	313
>  AccountRequestUsersDAOImpl.java	 31.9 %	115	245	360
>  PersonDAOImpl.java	 52.6 %	252	227	479
>  AssociatedPeopleDAOImpl.java	 41.2 %	141	201	342
>  AccountTypeDAOImpl.java	 38.1 %	117	190	307
>  AccountTransactionStatusDAOImpl.java	 0.0 %	0	17	17
>  AccountTransactionTypeDAOImpl.java	 0.0 %	0	17	17
>  AddressDAOImpl.java	 0.0 %	0	17	17
>  CityDAOImpl.java	 0.0 %	0	17	17
>  EmailDAOImpl.java	 0.0 %	0	17	17
>  PersonStandingDAOImpl.java	 0.0 %	0	17	17
>  PhoneNumberDAOImpl.java	 0.0 %	0	17	17

The DAOImpl at the bottom with 0% coverage are for stretch goals that will add additional functionality that goes beyond the scope of this sprint.











## Models

334 of 2150 Implementations (15.5%)

com.revature.bank.model		15.5 %	334	1,816	2,150
> AccountTransaction.java		0.0 %	0	235	235
> Person.java		27.5 %	75	198	273
> City.java		0.0 %	0	147	147
> Account.java		37.1 %	72	122	194
> Email.java		0.0 %	0	122	122
> PhoneNumber.java		0.0 %	0	122	122
> AccountTransactionStatus.java		0.0 %	0	108	108
> AccountTransactionType.java		0.0 %	0	108	108
> AccountOwnership.java		25.7 %	35	101	136
> Address.java		0.0 %	0	96	96
> PersonStanding.java		0.0 %	0	96	96
> AccountType.java		29.9 %	40	94	134
> AccountRequest.java		20.9 %	24	91	115
> AssociatedPeople.java		21.9 %	23	82	105
> AccountRequestUsers.java		28.0 %	23	59	82
> PermissionRank.java		54.5 %	42	35	77



## Exceptions

32 of 207 Instructions (15.5%)

com.revature.bank.exceptions		15.5 %	32	175	207
> AccessDenied.java		0.0 %	0	23	23
> ForceCloseThread.java		17.4 %	4	19	23
> InsufficientFunds.java		17.4 %	4	19	23
> InsufficientLineOfCredit.java		17.4 %	4	19	23
> InvalidNegativeValue.java		17.4 %	4	19	23
> InvalidPasswordChoice.java		17.4 %	4	19	23
> Logout.java		17.4 %	4	19	23
> ReturnMainMenu.java		17.4 %	4	19	23
> UnsupportedInteger.java		17.4 %	4	19	23


## Unimplemented Stretch Goals

0 of 594 Instructions (0%)

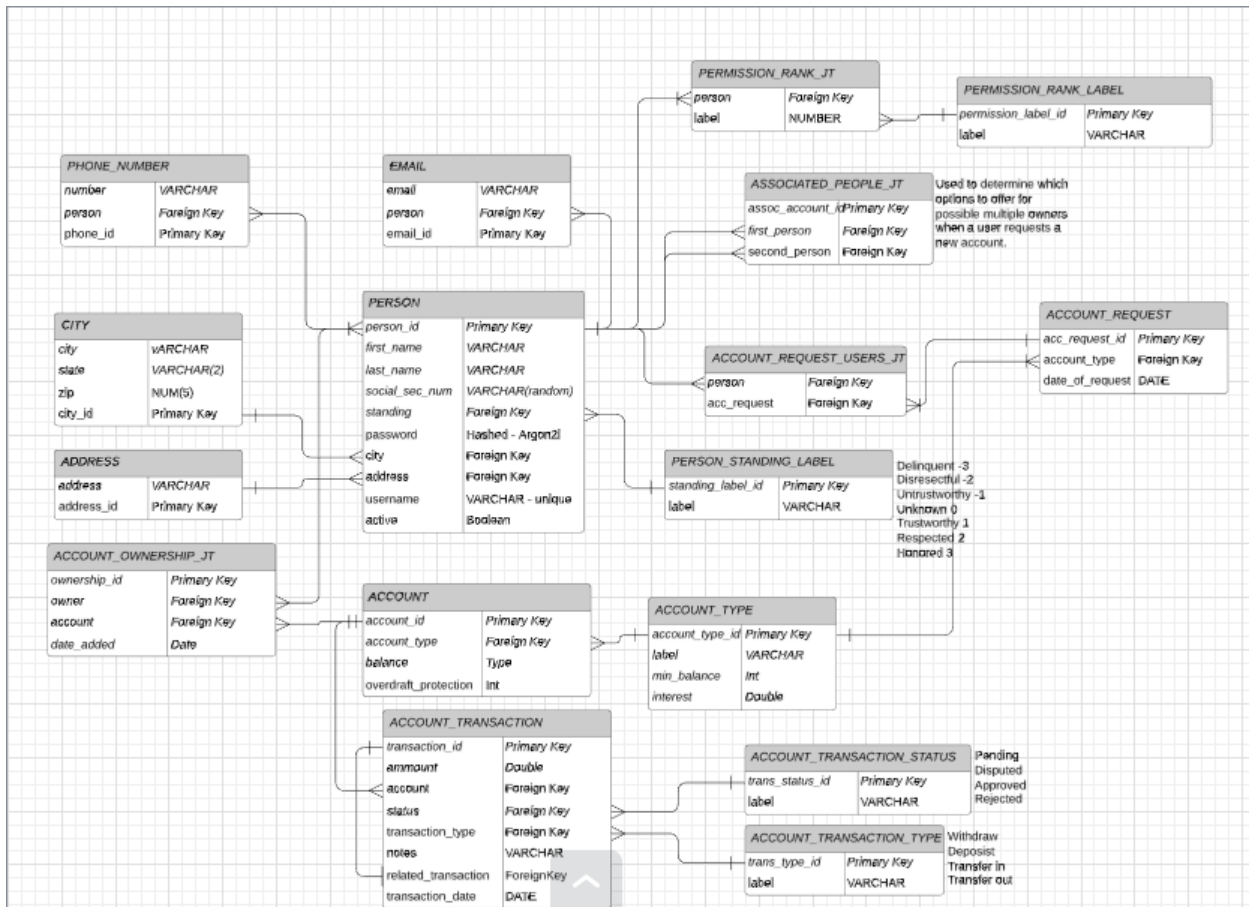
com.revature.bank.resources		0.0 %	0	594	594
> State.java		0.0 %	0	594	594

## Incidental Coverage of Test Classes

1497 of 1744 Instructions (85.8%)

src/test/java		85.8 %	1,497	247	1,744
---------------	---	--------	-------	-----	-------

# SQL ERD



Full Version: <https://www.lucidchart.com/invitations/accept/dbc26236-oc8b-4574-b6a3-bdf2ddoe3b96>

## Tables

1. ACCOUNT
2. ACCOUNT\_OWNERSHIP\_JT
3. ACCOUNT\_REQUEST
4. ACCOUNT\_REQUEST\_USERS\_JT
5. ACCOUNT\_TRANSACTION
6. ACCOUNT\_TRANSACTION\_STATUS
7. ACCOUNT\_TRANSACTION\_TYPE
8. ACCOUNT\_TYPE
9. ADDRESS – only in SQL, not implemented in Application
10. ASSOCIATED\_PEOPLE\_JT
11. CITY – only in SQL, not implemented in Application
12. EMAIL – only in SQL, not implemented in Application

13. PERMISSION\_RANK\_JT
14. PERMISSION\_RANK\_LABEL
15. PERSON
16. PERSON\_STANDING – only in SQL, not implemented in Application
17. PHONE\_NUMBER – only in SQL, not implemented in Application

## Procedures

1. create\_account
2. create\_account\_ownership\_jt
3. create\_account\_request
4. create\_acc\_tran
5. create\_acc\_tran\_status – only in SQL, not called in Application
6. create\_acc\_tran\_type – only in SQL, not called in Application
7. create\_account\_type
8. create\_account\_req\_users\_jt
9. create\_address – only in SQL, not called in Application
10. create\_associated\_people\_jt
11. create\_city – only in SQL, not called in Application
12. create\_deposit
13. create\_withdraw
14. create\_transfer
15. create\_email – only in SQL, not called in Application
16. create\_person
17. create\_standing – only in SQL, not called in Application
18. create\_rank – only in SQL, not called in Application
19. create\_phone – only in SQL, not called in Application

```

    usecaseDiagram
        participant Customer
        participant BankUser as Bank User
        participant AdminUser as Admin User

        Customer -->|1| Withdraw
        Customer -->|2| Deposit
        Customer -->|1| Transfer
        Customer -->|2| CheckAccount[Check Account]
        Customer -->|2| OpenAccount[Open Account]
        Customer -->|2| CloseAccount[Close Account]

        BankUser -->|2| Withdraw
        BankUser -->|2| Deposit
        BankUser -->|2| Transfer
        BankUser -->|2| CheckAccount
        BankUser -->|2| OpenAccount
        BankUser -->|2| CloseAccount
        BankUser -->|2| ManageEmployees[Manage Employees]

        AdminUser -->|2| ManageEmployees

        Withdraw -->|2| Withdraw
        Deposit -->|2| Deposit
        Transfer -->|2| Transfer
        CheckAccount -->|2| CheckAccount
        OpenAccount -->|2| OpenAccount
        CloseAccount -->|2| CloseAccount
        ManageEmployees -->|2| ManageEmployees

        Note over Customer, BankUser, AdminUser: Swimlanes
    
```

# Customers

- Create login user
- Set username (on creation)
- Set password (on creation)
- Set first name
- Set last name
- Set random SSN
- Request new account
- Deposit into owned account

- Withdraw from owned account
- Transfer from owned account
- Associate login user with another login user (to allow requesting joint account)
- Request joint account
- Add user to existing owned account

## Employees

- View user information
- View account information
- Approve pending accounts

## Admins

- View user information
- View account information
- Withdraw from any account
- Deposit into any account
- Transfer from any account
- Add any user to any account
- Deactivate any account
- Activate any account

## Validation

- Full sanitization of user input to either
  - Integer
  - Positive Integer
  - Token (one word alphanumeric)
  - Positive Double (two-digit decimal) – used for user views and transactions
  - Positive Double (five-digit decimal) – used for internal tracking of fractional cents for interest
- Withdraw and Transfer limitation
  - Limited by account\_type's minimum balance
  - Limitation expanded by account's overdraft protection