# **Air Via Implemented System**

|  |  |
| --- | --- |
|  | |
| **Document Intention:** | Report to Air Via on product implementation. |
| **Document Authors:** | Alfred Manville, Skyway Solutions |
| **Document Version Date-Stamp:** | 16/04/2023 |
| **Document Version:** | 1 |
| **Document Version History:** | |
| 1. Document created. | |

Table of Contents

[**Air Via Implemented System** 1](#_Toc132556110)

[**Introduction** 3](#_Toc132556111)

[**Compilation** 4](#_Toc132556112)

[**Architecture** 5](#_Toc132556113)

[**Source Files:** 6](#_Toc132556114)

[**Built Classes:** 9](#_Toc132556115)

[**Instructions** 12](#_Toc132556116)

[**Runtime** 13](#_Toc132556117)

[**Architecture** 15](#_Toc132556118)

[**Deployment** 16](#_Toc132556119)

[**Architecture** 16](#_Toc132556120)

[**SQL Schema (Auto-Generated by Program on Startup):** 17](#_Toc132556121)

[**Testing** 20](#_Toc132556122)

[**Functional Testing** 20](#_Toc132556123)

[**Non-Functional** 34](#_Toc132556124)

[**Review** 35](#_Toc132556125)

# **Introduction**

This document contains the implementation of the system describing its architecture, how to compile it using the required software and how it should be deployed on the target system. This document then follows with a set of testing plans with the results of each plan attached. Further, the java documentation for the implementation can be found at <https://github.com/karansambee/IN2018-Team-Project/tree/master/JavaDoc>, while a white paper on the way the database system is used is located at <https://cdn.captainalm.com/download/university/IN2018-DB-WhitePaper.docx> .

There is regret that not all features were implemented in time with GUIs but all backend packages were provided. The main source branch of this software can be found at <https://github.com/karansambee/IN2018-Team-Project/tree/master>.

# **Compilation**

This product has been developed using the java programming language (The Version 17 Standard Edition). When compiled, the produced data in non-native java byte code that must be executed by a Java Virtual Machine (JVM). During runtime, the JVM handles all memory allocation using a garbage collector to de-allocate memory. In addition, just-in-time (JIT) compilation of byte code to machine code can be done by the JVM to optimize the executed code to near native performance. The JVM implementation is free and the recommended one is called OpenJDK where you will need to use Adoptium on windows ( <https://adoptium.net/temurin/nightly/> ).

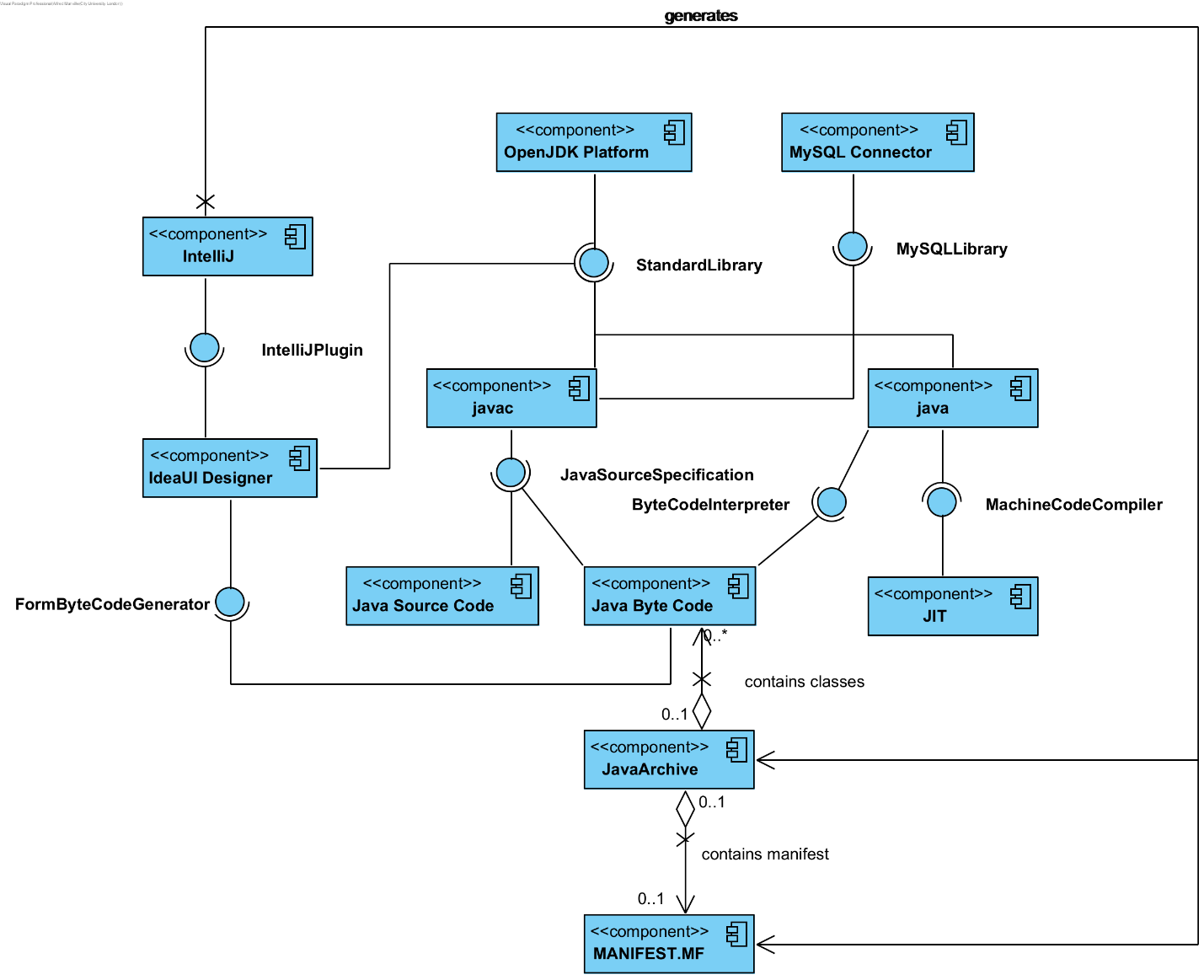
The was developed using IntelliJ and used to forms designer in order to create swing forms, the caveat to this is you must use IntelliJ for compilation as the forms designer currently has issues converting the XML form code to java source code (Duplicate symbols cause compilation to fail if the source code of the forms is generated). This means java swing is in use for all GUIs, this was chosen for the fact that swing is part of the Java SE specification and therefore must be supported by all interactive JVMs whereas its alternative, JavaFX is an extension part of the specification where support may not be included in a base install of java.

Dependencies for this project (Excluding the standard libraries provided by the JVM) include:

* mysql-connector-j (8.0.32) [GPLv2] which allows for connection to the database.
* forms\_rt [Apache 2] which is used as a utility library by swing forms produced by IntelliJ.

When being packed into a JAR file (Java Archive) a MANIFEST.MF file is used that contains information for execution class path (The paths of libraries / dependencies to load) and the class to execute when the JAR is executed.

## **Architecture**



### **Source Files:**

.\skywaysolutions

.\META-INF

.\Main.java

.\skywaysolutions\app

.\skywaysolutions\app\customers

.\skywaysolutions\app\database

.\skywaysolutions\app\gui

.\skywaysolutions\app\rates

.\skywaysolutions\app\reports

.\skywaysolutions\app\sales

.\skywaysolutions\app\staff

.\skywaysolutions\app\stock

.\skywaysolutions\app\utils

.\skywaysolutions\app\customers\CustomerType.java

.\skywaysolutions\app\customers\FlexiblePlanRange.java

.\skywaysolutions\app\customers\PlanType.java

.\skywaysolutions\app\customers\package-info.java

.\skywaysolutions\app\customers\CustomerTableAccessor.java

.\skywaysolutions\app\customers\Discount.java

.\skywaysolutions\app\customers\DiscountTableAccessor.java

.\skywaysolutions\app\customers\FlexibleDiscountEntriesTableAccessor.java

.\skywaysolutions\app\customers\FlexibleDiscountEntry.java

.\skywaysolutions\app\customers\Customer.java

.\skywaysolutions\app\customers\CustomerController.java

.\skywaysolutions\app\customers\ICustomerAccessor.java

.\skywaysolutions\app\database\IDB\_Connector.java

.\skywaysolutions\app\database\IFilterStatementCreator.java

.\skywaysolutions\app\database\MultiLoadSyncMode.java

.\skywaysolutions\app\database\package-info.java

.\skywaysolutions\app\database\DatabaseEntityBase.java

.\skywaysolutions\app\database\DatabaseTableBase.java

.\skywaysolutions\app\database\DB\_Connector.java

.\skywaysolutions\app\database\DatabaseBackupTable.java

.\skywaysolutions\app\gui\control

.\skywaysolutions\app\gui\tab

.\skywaysolutions\app\gui\hoster

.\skywaysolutions\app\gui\Login.form

.\skywaysolutions\app\gui\Main.form

.\skywaysolutions\app\gui\Prompt.form

.\skywaysolutions\app\gui\package-info.java

.\skywaysolutions\app\gui\About.form

.\skywaysolutions\app\gui\AccountEditor.form

.\skywaysolutions\app\gui\JDialogx.java

.\skywaysolutions\app\gui\BlankEditor.form

.\skywaysolutions\app\gui\BlankTypeEditor.form

.\skywaysolutions\app\gui\RateEditor.form

.\skywaysolutions\app\gui\About.java

.\skywaysolutions\app\gui\AccountEditor.java

.\skywaysolutions\app\gui\BlankEditor.java

.\skywaysolutions\app\gui\BlankTypeEditor.java

.\skywaysolutions\app\gui\CustomerEditor.form

.\skywaysolutions\app\gui\CustomerEditor.java

.\skywaysolutions\app\gui\Login.java

.\skywaysolutions\app\gui\Main.java

.\skywaysolutions\app\gui\Prompt.java

.\skywaysolutions\app\gui\RateEditor.java

.\skywaysolutions\app\gui\Seller.form

.\skywaysolutions\app\gui\Seller.java

.\skywaysolutions\app\gui\Transacter.form

.\skywaysolutions\app\gui\Transacter.java

.\skywaysolutions\app\gui\control\DateField.form

.\skywaysolutions\app\gui\control\PersonalInformationEditor.form

.\skywaysolutions\app\gui\control\package-info.java

.\skywaysolutions\app\gui\control\NonEditableDefaultTableModel.java

.\skywaysolutions\app\gui\control\StatusBar.form

.\skywaysolutions\app\gui\control\VTextField.java

.\skywaysolutions\app\gui\control\DateField.java

.\skywaysolutions\app\gui\control\PersonalInformationEditor.java

.\skywaysolutions\app\gui\control\StatusBar.java

.\skywaysolutions\app\gui\tab\DashboardTab.form

.\skywaysolutions\app\gui\tab\package-info.java

.\skywaysolutions\app\gui\tab\ITab.java

.\skywaysolutions\app\gui\tab\AccountsTab.form

.\skywaysolutions\app\gui\tab\DatabaseTab.form

.\skywaysolutions\app\gui\tab\RatesTab.form

.\skywaysolutions\app\gui\tab\StockTab.form

.\skywaysolutions\app\gui\tab\StockTypesTab.form

.\skywaysolutions\app\gui\tab\AccountsTab.java

.\skywaysolutions\app\gui\tab\CustomerTab.form

.\skywaysolutions\app\gui\tab\CustomerTab.java

.\skywaysolutions\app\gui\tab\DashboardTab.java

.\skywaysolutions\app\gui\tab\DatabaseTab.java

.\skywaysolutions\app\gui\tab\RatesTab.java

.\skywaysolutions\app\gui\tab\SaleTab.form

.\skywaysolutions\app\gui\tab\SaleTab.java

.\skywaysolutions\app\gui\tab\StockTab.java

.\skywaysolutions\app\gui\tab\StockTypesTab.java

.\skywaysolutions\app\gui\hoster\HostRunner.java

.\skywaysolutions\app\gui\hoster\IHostInvokable.java

.\skywaysolutions\app\gui\hoster\package-info.java

.\skywaysolutions\app\rates\package-info.java

.\skywaysolutions\app\rates\ConversionRate.java

.\skywaysolutions\app\rates\ExchangeRateTableAccessor.java

.\skywaysolutions\app\rates\IRateAccessor.java

.\skywaysolutions\app\rates\RateController.java

.\skywaysolutions\app\reports\IReportAccessor.java

.\skywaysolutions\app\reports\IReportGenerator.java

.\skywaysolutions\app\reports\TableCell.java

.\skywaysolutions\app\reports\TicketStockTurnover.java

.\skywaysolutions\app\reports\package-info.java

.\skywaysolutions\app\sales\Payment.java

.\skywaysolutions\app\sales\PaymentType.java

.\skywaysolutions\app\sales\SaleType.java

.\skywaysolutions\app\sales\package-info.java

.\skywaysolutions\app\sales\ISalesAccessor.java

.\skywaysolutions\app\sales\Refund.java

.\skywaysolutions\app\sales\RefundTableAccessor.java

.\skywaysolutions\app\sales\Sale.java

.\skywaysolutions\app\sales\SaleController.java

.\skywaysolutions\app\sales\SaleTableAccessor.java

.\skywaysolutions\app\sales\Transaction.java

.\skywaysolutions\app\sales\TransactionTableAccessor.java

.\skywaysolutions\app\staff\PasswordString.java

.\skywaysolutions\app\staff\StaffRole.java

.\skywaysolutions\app\staff\package-info.java

.\skywaysolutions\app\staff\Account.java

.\skywaysolutions\app\staff\StaffTableAccessor.java

.\skywaysolutions\app\staff\AccountController.java

.\skywaysolutions\app\staff\IStaffAccessor.java

.\skywaysolutions\app\stock\package-info.java

.\skywaysolutions\app\stock\BlankTableAccessor.java

.\skywaysolutions\app\stock\BlankType.java

.\skywaysolutions\app\stock\BlankTypeTableAccessor.java

.\skywaysolutions\app\stock\Blank.java

.\skywaysolutions\app\stock\BlankController.java

.\skywaysolutions\app\stock\IStockAccessor.java

.\skywaysolutions\app\utils\CheckedException.java

.\skywaysolutions\app\utils\MonthPeriod.java

.\skywaysolutions\app\utils\PersonalInformation.java

.\skywaysolutions\app\utils\Time.java

.\skywaysolutions\app\utils\package-info.java

.\skywaysolutions\app\utils\IRepairable.java

.\skywaysolutions\app\utils\ResultSetNullableReturners.java

.\skywaysolutions\app\utils\Stream.java

.\skywaysolutions\app\utils\AccessorManager.java

.\skywaysolutions\app\utils\Decimal.java

.\META-INF\MANIFEST.MF

### **Built Classes:**

.\com

.\skywaysolutions

.\META-INF

.\Main.class

.\com\intellij

.\com\intellij\uiDesigner

.\com\intellij\uiDesigner\core

.\com\intellij\uiDesigner\core\AbstractLayout.class

.\com\intellij\uiDesigner\core\DimensionInfo.class

.\com\intellij\uiDesigner\core\GridConstraints.class

.\com\intellij\uiDesigner\core\GridLayoutManager.class

.\com\intellij\uiDesigner\core\HorizontalInfo.class

.\com\intellij\uiDesigner\core\LayoutState.class

.\com\intellij\uiDesigner\core\Spacer.class

.\com\intellij\uiDesigner\core\SupportCode$TextWithMnemonic.class

.\com\intellij\uiDesigner\core\SupportCode.class

.\com\intellij\uiDesigner\core\Util.class

.\com\intellij\uiDesigner\core\VerticalInfo.class

.\skywaysolutions\app

.\skywaysolutions\app\customers

.\skywaysolutions\app\utils

.\skywaysolutions\app\database

.\skywaysolutions\app\gui

.\skywaysolutions\app\rates

.\skywaysolutions\app\staff

.\skywaysolutions\app\stock

.\skywaysolutions\app\sales

.\skywaysolutions\app\reports

.\skywaysolutions\app\customers\CustomerType.class

.\skywaysolutions\app\customers\FlexiblePlanRange.class

.\skywaysolutions\app\customers\PlanType.class

.\skywaysolutions\app\customers\CustomerTableAccessor.class

.\skywaysolutions\app\customers\Customer.class

.\skywaysolutions\app\customers\Discount.class

.\skywaysolutions\app\customers\DiscountTableAccessor.class

.\skywaysolutions\app\customers\FlexibleDiscountEntriesTableAccessor.class

.\skywaysolutions\app\customers\FlexibleDiscountEntry.class

.\skywaysolutions\app\customers\CustomerController$CustomerFilter.class

.\skywaysolutions\app\customers\CustomerController$DiscountFilter.class

.\skywaysolutions\app\customers\CustomerController$AliasFinder.class

.\skywaysolutions\app\customers\CustomerController$FlexiblePlanFilter.class

.\skywaysolutions\app\customers\CustomerController.class

.\skywaysolutions\app\customers\ICustomerAccessor.class

.\skywaysolutions\app\utils\Decimal.class

.\skywaysolutions\app\utils\CheckedException.class

.\skywaysolutions\app\utils\PersonalInformation.class

.\skywaysolutions\app\utils\IRepairable.class

.\skywaysolutions\app\utils\AccessorManager.class

.\skywaysolutions\app\utils\MonthPeriod.class

.\skywaysolutions\app\utils\Time.class

.\skywaysolutions\app\utils\ResultSetNullableReturners.class

.\skywaysolutions\app\utils\Stream.class

.\skywaysolutions\app\database\DatabaseTableBase.class

.\skywaysolutions\app\database\DatabaseEntityBase.class

.\skywaysolutions\app\database\IDB\_Connector.class

.\skywaysolutions\app\database\IFilterStatementCreator.class

.\skywaysolutions\app\database\MultiLoadSyncMode.class

.\skywaysolutions\app\database\DB\_Connector.class

.\skywaysolutions\app\database\DatabaseBackupTable.class

.\skywaysolutions\app\gui\control

.\skywaysolutions\app\gui\tab

.\skywaysolutions\app\gui\hoster

.\skywaysolutions\app\gui\Prompt$1.class

.\skywaysolutions\app\gui\Prompt.class

.\skywaysolutions\app\gui\JDialogx.class

.\skywaysolutions\app\gui\AccountEditor$1.class

.\skywaysolutions\app\gui\AccountEditor$2.class

.\skywaysolutions\app\gui\AccountEditor.class

.\skywaysolutions\app\gui\CustomerEditor$1.class

.\skywaysolutions\app\gui\CustomerEditor.class

.\skywaysolutions\app\gui\RateEditor$1.class

.\skywaysolutions\app\gui\RateEditor.class

.\skywaysolutions\app\gui\BlankEditor$1.class

.\skywaysolutions\app\gui\BlankEditor.class

.\skywaysolutions\app\gui\BlankTypeEditor$1.class

.\skywaysolutions\app\gui\BlankTypeEditor.class

.\skywaysolutions\app\gui\About$1.class

.\skywaysolutions\app\gui\About.class

.\skywaysolutions\app\gui\Login$1.class

.\skywaysolutions\app\gui\Login.class

.\skywaysolutions\app\gui\Main$1.class

.\skywaysolutions\app\gui\Main.class

.\skywaysolutions\app\gui\Seller.class

.\skywaysolutions\app\gui\Transacter.class

.\skywaysolutions\app\gui\control\NonEditableDefaultTableModel.class

.\skywaysolutions\app\gui\control\VTextField$1.class

.\skywaysolutions\app\gui\control\VTextField.class

.\skywaysolutions\app\gui\control\StatusBar$HelpActionHandler.class

.\skywaysolutions\app\gui\control\StatusBar$StatusClearRunner.class

.\skywaysolutions\app\gui\control\StatusBar.class

.\skywaysolutions\app\gui\control\DateField$1.class

.\skywaysolutions\app\gui\control\DateField.class

.\skywaysolutions\app\gui\control\PersonalInformationEditor$PIUpdater.class

.\skywaysolutions\app\gui\control\PersonalInformationEditor.class

.\skywaysolutions\app\gui\tab\ITab.class

.\skywaysolutions\app\gui\tab\AccountsTab.class

.\skywaysolutions\app\gui\tab\CustomerTab.class

.\skywaysolutions\app\gui\tab\DashboardTab.class

.\skywaysolutions\app\gui\tab\DatabaseTab.class

.\skywaysolutions\app\gui\tab\RatesTab.class

.\skywaysolutions\app\gui\tab\SaleTab.class

.\skywaysolutions\app\gui\tab\StockTab.class

.\skywaysolutions\app\gui\tab\StockTypesTab.class

.\skywaysolutions\app\gui\hoster\IHostInvokable.class

.\skywaysolutions\app\gui\hoster\HostRunner$EventStruct.class

.\skywaysolutions\app\gui\hoster\HostRunner.class

.\skywaysolutions\app\rates\IRateAccessor.class

.\skywaysolutions\app\rates\ConversionRate.class

.\skywaysolutions\app\rates\ExchangeRateTableAccessor.class

.\skywaysolutions\app\rates\RateController$AllFilter.class

.\skywaysolutions\app\rates\RateController.class

.\skywaysolutions\app\staff\IStaffAccessor.class

.\skywaysolutions\app\staff\StaffRole.class

.\skywaysolutions\app\staff\PasswordString.class

.\skywaysolutions\app\staff\Account.class

.\skywaysolutions\app\staff\StaffTableAccessor.class

.\skywaysolutions\app\staff\AccountController$EmailFinder.class

.\skywaysolutions\app\staff\AccountController.class

.\skywaysolutions\app\stock\IStockAccessor.class

.\skywaysolutions\app\stock\BlankTableAccessor.class

.\skywaysolutions\app\stock\Blank.class

.\skywaysolutions\app\stock\BlankType.class

.\skywaysolutions\app\stock\BlankTypeTableAccessor.class

.\skywaysolutions\app\stock\BlankController$BlankAssignmentFilter.class

.\skywaysolutions\app\stock\BlankController$AllFilter.class

.\skywaysolutions\app\stock\BlankController.class

.\skywaysolutions\app\sales\ISalesAccessor.class

.\skywaysolutions\app\sales\SaleType.class

.\skywaysolutions\app\sales\Payment.class

.\skywaysolutions\app\sales\PaymentType.class

.\skywaysolutions\app\sales\Sale.class

.\skywaysolutions\app\sales\Transaction$1.class

.\skywaysolutions\app\sales\Transaction.class

.\skywaysolutions\app\sales\Refund.class

.\skywaysolutions\app\sales\RefundTableAccessor.class

.\skywaysolutions\app\sales\SaleController$IDFilterer.class

.\skywaysolutions\app\sales\SaleController$SaleFilter.class

.\skywaysolutions\app\sales\SaleController.class

.\skywaysolutions\app\sales\SaleTableAccessor.class

.\skywaysolutions\app\sales\TransactionTableAccessor.class

.\skywaysolutions\app\reports\IReportAccessor.class

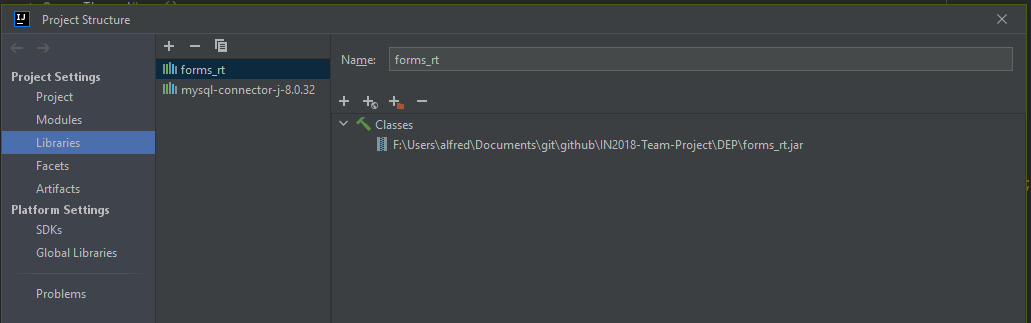
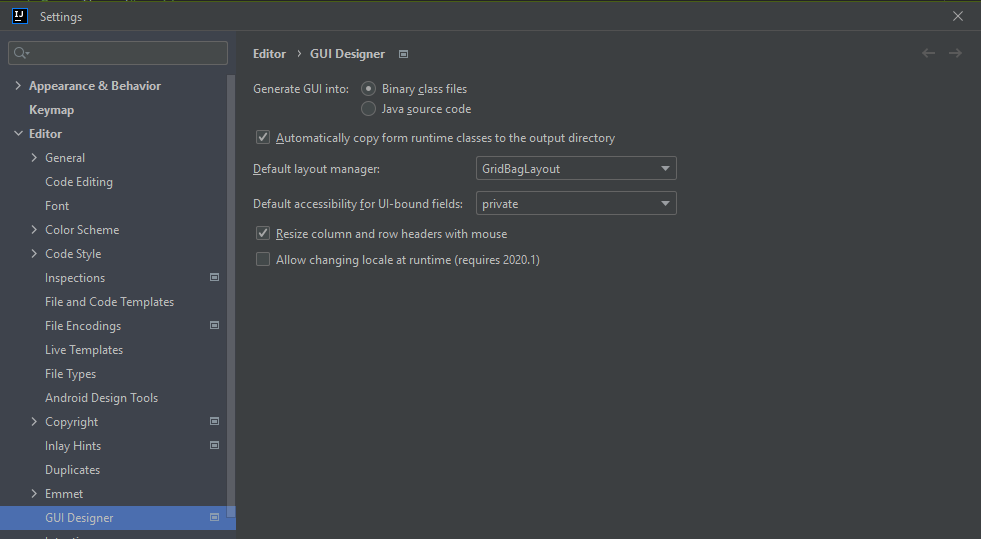
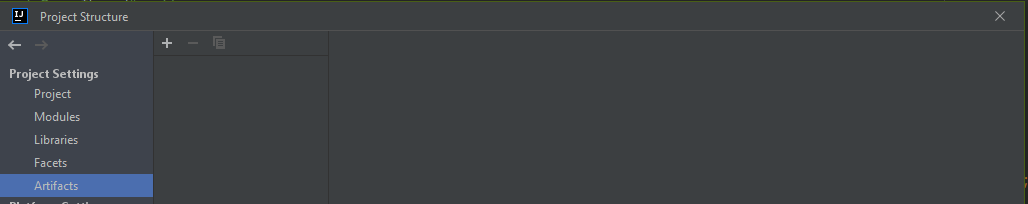
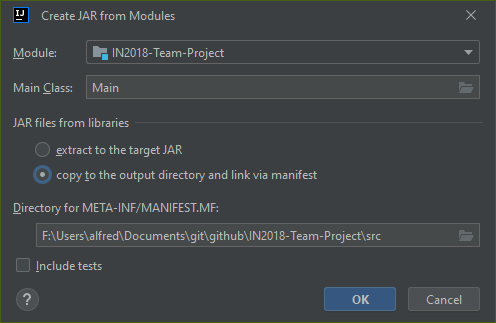
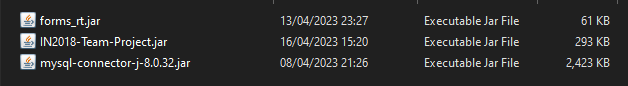
.\skywaysolutions\app\reports\IReportGenerator.class

.\skywaysolutions\app\reports\TableCell.class

.\skywaysolutions\app\reports\TicketStockTurnover.class

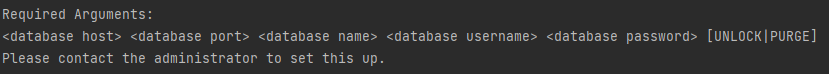
.\META-INF\MANIFEST.MF

## **Instructions**

1. Import the source code into IntelliJ (Copy the contents of the provided src folder into the empty src folder of a new IntelliJ project).
2. Obtain the mysql connector from <https://dev.mysql.com/downloads/connector/j/> and switch the combo box to “Platform Independent”. Once downloaded, extract the jar file and place it into a folder called DEP in the project.
3. Obtain the forms\_rt.jar from the lib folder within the IntelliJ installation director (Platform / installation dependent) and place this within the DEP folder as well.
4. Go to File -> Project Structure -> Libraries (In Project Settings): 
5. Now add the two JAR files from the DEP folder as shown above (Press the plus icon, then select the jar file).
6. Press Ok to close the dialog.
7. Go to File -> Settings -> Editor -> GUI Designer: 
8. Make sure the GUI is generated into binary class files and the default layout manager is GridBagLayout.
9. Press Ok to close the dialog.
10. Go to Build -> Build Project to build the project.
11. Go to File -> Project Structure -> Artifacts (In project Settings): 
12. Now press the + icon and select JAR -> From Modules with Dependencies: 
13. Make sure the main class is Main and that the libraries are copied to the output directory and linked via manifest, then press OK.
14. Then Go to Build -> Build Artifacts -> Select the artifact -> Build.
15. The artifacts have now been built and a folder with the results can be found in out/artifacts: 

# **Runtime**

This software starts off by invoking the Main class in which the arguments are checked for:



The database host is the IP Address or Hostname of the database server, the port is the port that database server is running on, the database name is the name of the database, the username is the name of the user, the password is the password used for authentication. This part is to be setup be the administrator who is responsible for creating shortcuts that launch the JAR with the correct parameters.

If a PURGE parameter was provided, the database will be cleared and then the tables will be regenerated (Which includes default entries such as the default administrator account and the USD rate).

If an UNLOCK parameter was provided, the database will be forced to ‘unlock’ records locked by the auxiliary table system.

The database functionality uses the MYSQL Connector library and the rendered GUI that appears next uses the forms\_rt library which provides utilities for swing.

At the bottom of all dialogs (Except about and prompt), you have a status bar:



Pressing the “?” button will allow you to select a component to get help information for it, it there is no help information, nothing happens. This mode can be exited by pressing the “x”.



Extended prompts are used for longer status messages and for exception information. These are accessed by pressing the “^” arrow.

The default created administrator account has an ID of 1, the email address of admin@localhost and a password of Administrator. This should not be removed (As it would be re-created) but instead can be disabled using the staff manager tab.

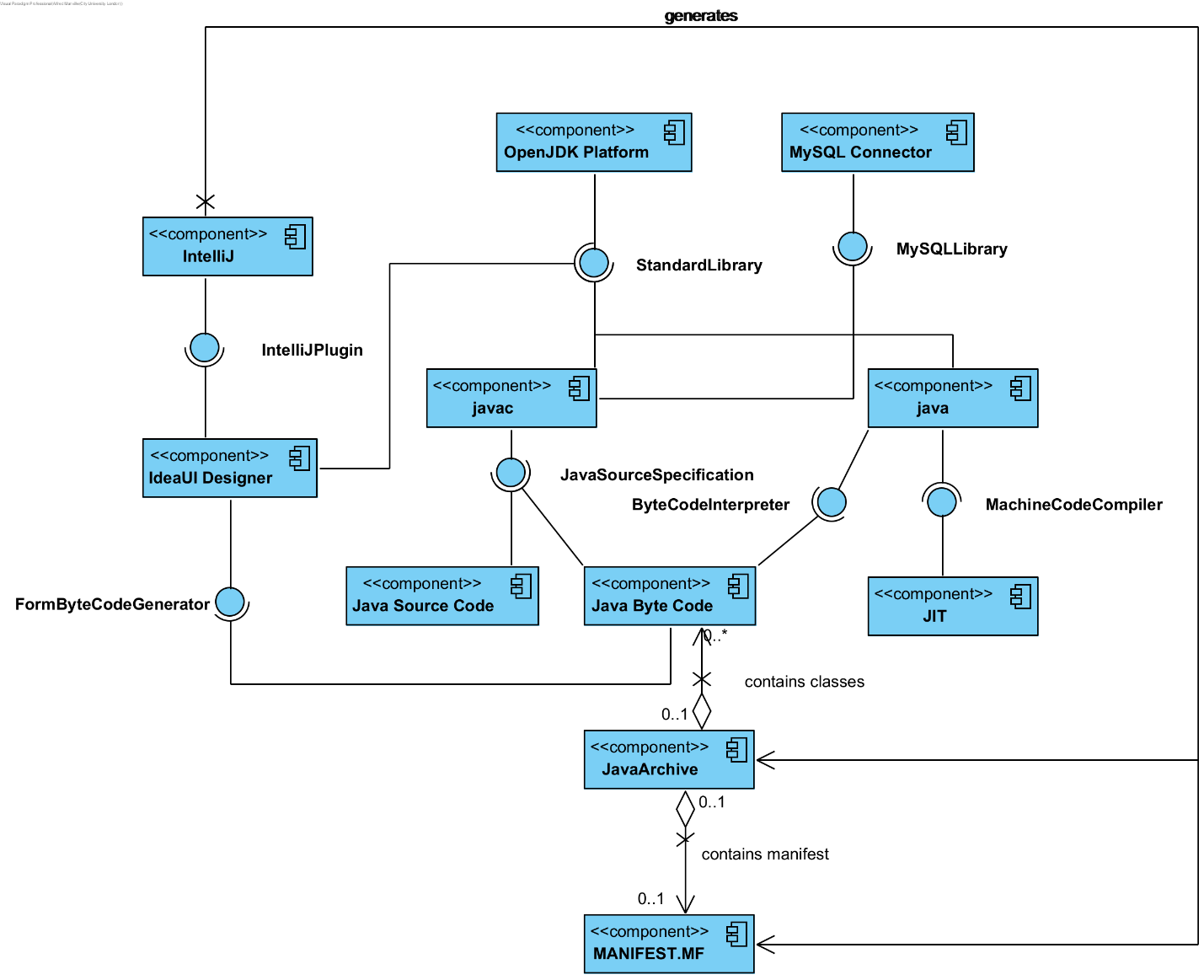
The default created rate is for USD and it is a 1:1 rate.

In order to create stock, you must create stock types first and at least 1 advisor account.

All passwords the software uses are hashed and salted as part of the process.

The database has 2 tables for every table provided by the given SQL schema, the table and an auxiliary table, which has an AUX\_ prefix. The AUX tables have the same primary key as the normal table but none of the other columns. When a corresponding record exists in the AUX table, the record is said to be unlocked, when the record does not exist, it is said to be locked. A record must be locked before it can be stored to and should be locked when loading from it. This system allows for concurrency to work.

## **Architecture**



# **Deployment**

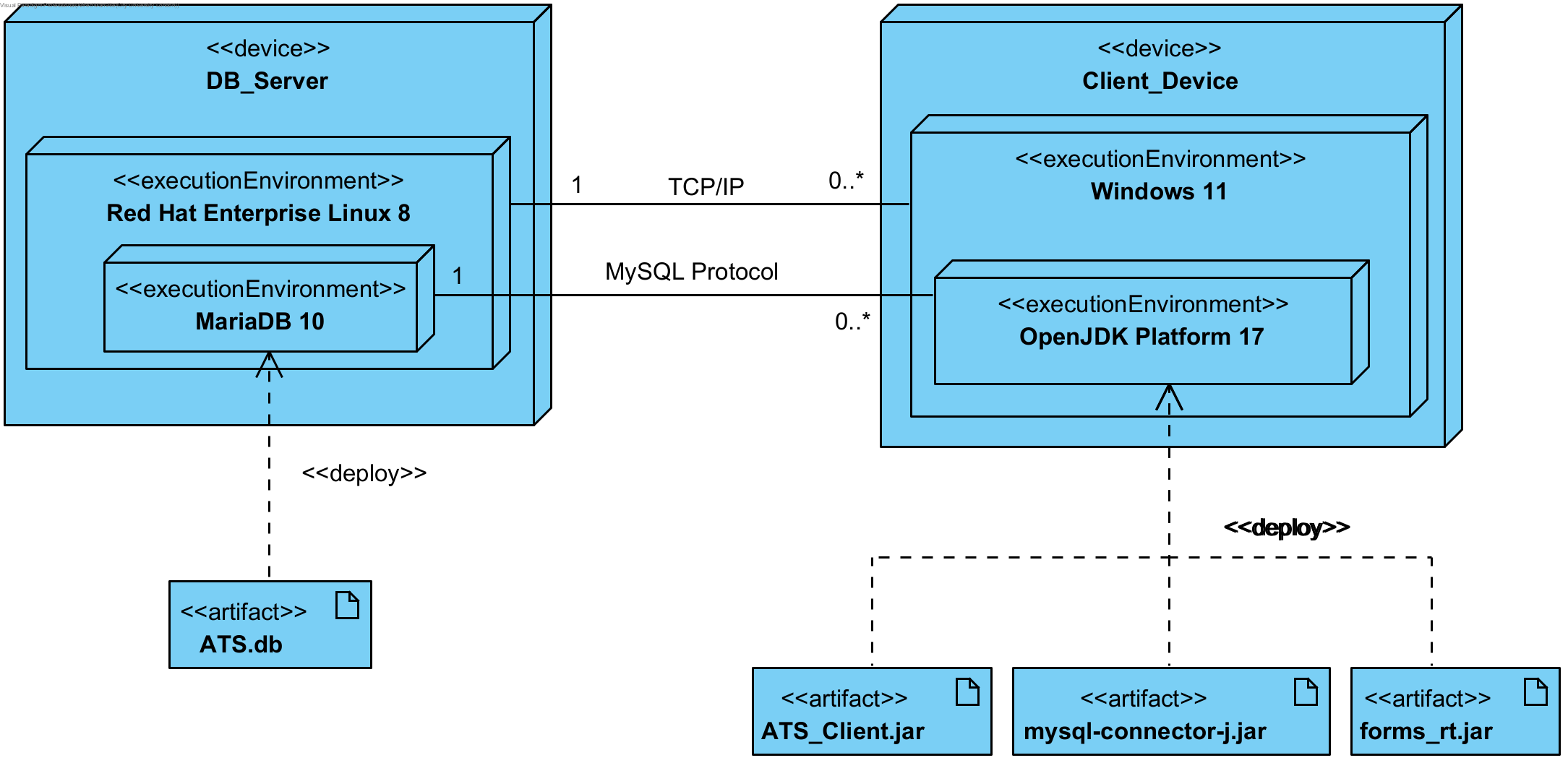
The software is to be deployed onto the system, once in artifact JAR form, in the same folder as its 2 dependencies (MySQL connector J and forms\_rt). The system should have a JVM installed with this JVM conforming to Java 17 language specification.

The database server must be running a MySQL protocol compatible server, such as MariaDB with a database setup and at least a single DDL user. This DDL user’s credentials are used by the administrator in accessing the database through the program for setup.

Once the administrator puts the required arguments for the JAR file (Using the DDL credentials), the program will create its tables automatically when started up for the first time. The administrator can log in with the default credentials to setup the system or restore a backup.

The software can have shortcuts created against it, EG: <Path to java executable> -jar <database host> <database port> <database name> <database username> <database password>

## **Architecture**



## **SQL Schema (Auto-Generated by Program on Startup):**

/\* Complete set of DDL statements (CREATE TABLE statements) \*/

CREATE TABLE ExchangeRate (

CurrencyName char(4) NOT NULL,

USDConversionRate numeric(8, 6) NOT NULL,

CurrencySymbol char(2) NOT NULL,

PRIMARY KEY (CurrencyName));

CREATE TABLE Staff (

StaffID bigint(19) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

CurrencyName char(4),

StaffRole integer(1) NOT NULL,

ComissionRate numeric(8, 6),

Firstname varchar(15) NOT NULL,

Surname varchar(15) NOT NULL,

PhoneNumber varchar(15) NOT NULL,

EmailAddress varchar(25) NOT NULL UNIQUE,

DateOfBirth date NOT NULL,

Postcode varchar(7) NOT NULL,

HouseNumber varchar(4) NOT NULL,

StreetName varchar(20) NOT NULL,

HashedPassword binary(32) NOT NULL,

PasswordSalt binary(32) NOT NULL,

FOREIGN KEY (CurrencyName) REFERENCES ExchangeRate(CurrencyName));

CREATE TABLE BlankType (

TypeNumber INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

TypeDescription varchar(255) NOT NULL);

CREATE TABLE Blank (

BlankNumber bigint(19) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

StaffID bigint(19) NOT NULL,

TypeNumber integer(3) NOT NULL,

BlankDescription varchar(255) NOT NULL,

Blacklisted boolean NOT NULL,

Void boolean NOT NULL,

ReceivedDate date NOT NULL,

AssignedDate date,

ReturnedDate date,

FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),

FOREIGN KEY (TypeNumber) REFERENCES BlankType(TypeNumber));

CREATE TABLE DiscountPlan (

DiscountPlanID bigint(19) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

DiscountType integer(1) NOT NULL,

DiscountPercentage numeric(8, 6));

CREATE TABLE FlexibleDiscountEntries (

EntryID bigint(19) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

DiscountPlanID bigint(19) NOT NULL,

AmountLowerBound numeric(12, 2) NOT NULL,

AmountUpperBound numeric(12, 2) NOT NULL,

DiscountPercentage numeric(8, 6) NOT NULL,

FOREIGN KEY (DiscountPlanID) REFERENCES DiscountPlan(DiscountPlanID));

CREATE TABLE Customer (

CustomerID bigint(19) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

DiscountPlanID bigint(19),

CurrencyName char(4) NOT NULL,

Firstname varchar(15) NOT NULL,

Surname varchar(15) NOT NULL,

PhoneNumber varchar(15) NOT NULL,

EmailAddress varchar(25),

DateOfBirth date NOT NULL,

Postcode varchar(7) NOT NULL,

HouseNumber varchar(4) NOT NULL,

StreetName varchar(20) NOT NULL,

AccountDiscountCredit numeric(12, 2),

PurchaseAccumulation numeric(10, 2) NOT NULL,

PurchaseMonthBeginning date NOT NULL,

Alias varchar(32) NOT NULL UNIQUE,

CustomerType integer(1) NOT NULL,

FOREIGN KEY (DiscountPlanID) REFERENCES DiscountPlan(DiscountPlanID),

FOREIGN KEY (CurrencyName) REFERENCES ExchangeRate(CurrencyName));

CREATE TABLE Sale (

BlankNumber bigint(19) NOT NULL,

CustomerID bigint(19) NOT NULL,

CurrencyName char(4) NOT NULL,

SaleType integer(1) NOT NULL,

CommissonRate numeric(8, 6) NOT NULL,

SaleDate date NOT NULL,

DueDate date NOT NULL,

Cost numeric(8, 2) NOT NULL,

CostInUSD numeric(8, 2),

Tax numeric(8, 2) NOT NULL,

AdditionalTax numeric(8, 2),

PreDiscountCost numeric(8, 2),

PRIMARY KEY (BlankNumber),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (BlankNumber) REFERENCES Blank(BlankNumber),

FOREIGN KEY (CurrencyName) REFERENCES ExchangeRate(CurrencyName));

CREATE TABLE Transcation (

TranscationID bigint(19) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

BlankNumber bigint(19) NOT NULL,

CurrencyName char(4) NOT NULL,

AmountPaid numeric(10, 2) NOT NULL,

AmountPaidInUSD numeric(10, 2),

TransactionDate date NOT NULL,

PaymentType integer(1) NOT NULL,

CardNumber bigint(19),

ChequeNumber bigint(19),

FOREIGN KEY (BlankNumber) REFERENCES Sale(BlankNumber),

FOREIGN KEY (CurrencyName) REFERENCES ExchangeRate(CurrencyName));

CREATE TABLE Refund (

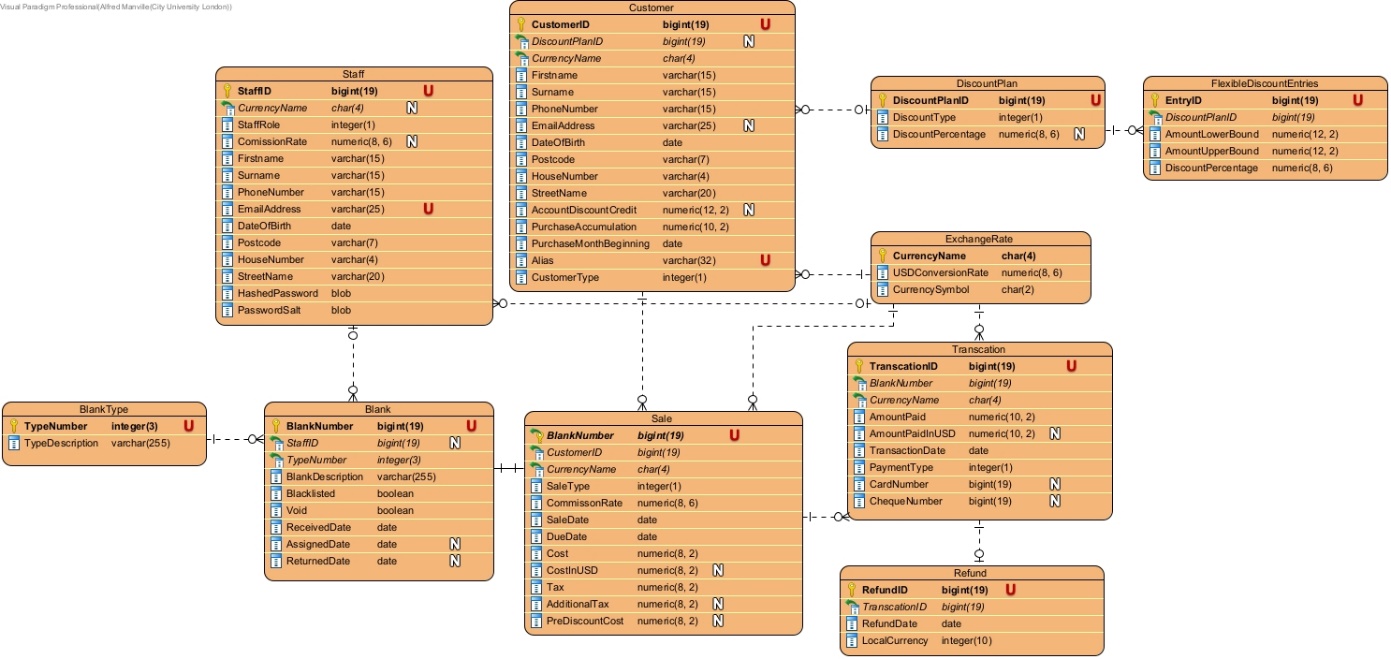
RefundID bigint(19) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

TranscationID bigint(19) NOT NULL,

RefundDate date NOT NULL,

LocalCurrency integer(10) NOT NULL,

FOREIGN KEY (TranscationID) REFERENCES Transcation(TranscationID));

****

# **Testing**

## **Functional Testing**

|  |  |
| --- | --- |
| **Use Case ID:** 1 | **Use Case Name:** Sign In |
| **Test Number:** 1.1 | |
| **Objective:** allow the user to sign in with the valid credentials | |
| **Set Up:** There is a stable connection, and the code is working | |
| Expected results:  1. The user puts their details in  2. The details are correct  3. Sign in is successful  4.Next screen is showed to the user | |
| **Test:**  1**.** The user is presented with the form  2. The user inputs their email  3. The user inputs the relevant password for that email account  4. This password is checked to see if it matches the relevant email account  5. If it does the next screen will be presented to the user | |
| **Test Record:** | |
| **Date:** 05-03-2023 | **Tester: Huzaifa Pathan** |
| **Result:** passed | |

|  |  |
| --- | --- |
| **Use Case ID:** 1 | **Use Case Name:** Sign In - LoginAttemptFailed |
| **Test Number:** 1.2 | |
| **Objective:** login attempt failed because the email was inputted incorrectly | |
| **Set Up:**   1. There is a stable connection to the database 2. The code is working and running 3. The sign in screen is displayed to the user | |
| **Expected results:**   1. An error message is displayed to the user indicating the error (wrong format) | |
| **Test:**   1. The user is presented with the sign in screen 2. The user inputs an email 3. The system alerts the user with an error message that the email has been inputted in the wrong format | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 1 | **Use Case Name:** Sign In - LoginAttemptFailed |
| **Test Number: 1.3** | |
| **Objective:** login attempt failed because the email was incorrect | |
| **Set Up:**   1. There is a stable connection to the database 2. The code is working and running 3. The sign in screen is displayed to the user | |
| **Expected results:**   1. An error message is displayed to the user indicating that the email inputted could not be found | |
| **Test:**   1. The user is presented with the sign in screen 2. The user inputs an email 3. The user presses login 4. An error message comes up alerting the user that the email they have inputted could not be found | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 1 | **Use Case Name:** Sign In - LoginAttemptFailed |
| **Test Number: 1.4** | |
| **Objective:** login attempt failed because the password was incorrect | |
| **Set Up:** There is a stable connection, and the code is working | |
| **Expected results:** the user puts a password which doesn’t match the relevant account and the sign in attempt fails | |
| **Test:**   1. The user inputs an email 2. The user inputs a password 3. The user presses the submit button 4. The password doesn’t match the password that maps to the email on the system 5. An error message is displayed to the user showing this | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 1 | **Use Case Name:** Sign In – Connection Unavailable |
| **Test Number: 1.5** | |
| **Objective:** login attempt fails because the connection to the database drops | |
| **Set Up:** The connection was initially there to get to the login screen but as the user inputs their details correctly and presses submit the connection to the database is lost | |
| **Expected results:** the system shows an error that the connection to the database has been lost | |
| **Test:**   1. The sign in screen is displayed to the user 2. The user inputs their email and password and presses submit 3. The password is checked against the account with that email 4. The password Is a match 5. The connection is lost 6. An error message is displayed informing the user of this | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 2 | **Use Case Name:** Sign Out |
| **Test Number: 2** | |
| **Objective:** the user signs out of the system | |
| **Set Up:**   1. There is a connection to the database 2. The user is signed in | |
| **Expected results:** the user can press the logout button and gets logged out | |
| **Test:**   1. User has logged in to the system 2. User presses the logout button 3. User has been logged out of the system 4. User is returned to the sign in page | |
| **Test Record:** | |
| **Date: 06-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account |
| **Test Number: 3.1** | |
| **Objective:** System administrator creates an account for a user | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** the system administrator creates an account for the user | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Fills the required fields in correctly 5. Presses OK 6. New account has been added | |
| **Test Record:** | |
| **Date: 06-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account – no staffID |
| **Test Number: 3.2** | |
| **Objective**: System administrator creates an account for a user- fails because there is no staff ID | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** error message comes up indicating staff ID is null | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Leaves staffID null 5. Presses OK 6. Error message says staffID is null | |
| **Test Record:** | |
| **Date: 06-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account – incorrect email format |
| **Test Number: 3.3** | |
| **Objective:** System administrator creates an account for a user- fails because the email format is incorrect | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** error message comes up indicating the email is in the wrong format | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Types email incorrectly 5. Presses OK 6. Error message says email is in an invalid format | |
| **Test Record:** | |
| **Date: 06-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account – email in use |
| **Test Number: 3.4** | |
| Objective: System administrator creates an account for a user- fails because the email is already in use | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** error message comes up indicating the email is already in use | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Uses an email that is already being used 5. Presses OK 6. Error message says email is already in use | |
| **Test Record:** | |
| **Date: 07-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account – staffID in use |
| **Test Number: 3.5** | |
| **Objective:** System administrator creates an account for a user- fails because the staffIDis already in use | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** error message comes up indicating the email is already in use | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Uses staff ID that is already being used 5. Presses OK 6. Error message says staffID is already in use | |
| **Test Record:** | |
| **Date: 07-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account – email field is null |
| **Test Number: 3.6** | |
| **Objective:** System administrator creates an account for a user- fails because the email field is null | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** error message comes up indicating the email field is null | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Leaves email filed null 5. Presses OK 6. Error message says email is null | |
| **Test Record:** | |
| **Date: 07-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account – password field is null |
| **Test Number: 3.7** | |
| **Objective:** System administrator creates an account for a user- fails because the password field is null | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** error message comes up indicating the password field is null | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Leaves password filed null 5. Presses OK 6. Error message says password is null | |
| **Test Record:** | |
| **Date: 07-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 3 | **Use Case Name:** Create Account – passwords don’t match |
| **Test Number: 3.8** | |
| **Objective:** System administrator creates an account for a user- fails because the passwords don’t match | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** error message comes up indicating the passwords don’t match | |
| **Test:**   1. System admin signs in 2. Clicks on staff manager 3. Clicks on add 4. Input the password and confirm password fields differently 5. Presses OK 6. Error message says password fields do not match | |
| **Test Record:** | |
| **Date: 07-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 4 | **Use Case Name:** Retrieve Accounts |
| **Test Number: 4.1** | |
| **Objective:** System administrator retrieves multiple accounts | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** the account in mind is retrieved from database | |
| **Test:**   1. Database is queried by the user 2. List of the accounts queried are displayed 3. Each account Is then stored as an object in the array | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 4 | **Use Case Name:** Retrieve Accounts – connection has been lost |
| **Test Number: 4.2** | |
| **Objective:** System administrator retrieves multiple accounts – the connection drops | |
| **Set Up:**   1. There is a connection to the database initially 2. System administrator has signed in | |
| **Expected results:** error occurs as connection has dropped | |
| **Test:**   1. Database is queried by the user 2. List of the accounts queried are displayed 3. Each account Is then stored as an object in the array 4. During any of these processes the connection may be lost, and the querying cannot happen | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 4 | **Use Case Name:** Retrieve Accounts – no accounts came up when querying |
| **Test Number: 4.3** | |
| **Objective:** System administrator retrieves multiple accounts – no accounts were found | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in | |
| **Expected results:** no accounts were shown when the query search happened | |
| **Test:**   1. Database is queried by the user 2. No accounts were displayed | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 5 | **Use Case Name:** Retrieve Account |
| **Test Number: 5** | |
| **Objective:** System administrator retrieves multiple accounts | |
| **Set Up:**   1. There is a connection to the database 2. System administrator has signed in 3. There are a list of accounts to be searched from | |
| **Expected results:** the account in mind is retrieved from database | |
| **Test:**   1. Use Case ID 4 2. For each account in this list if it’s the one being searched for then return it else go to next account 3. If no account match return null | |
| **Test Record:** | |
| **Date: 05-03-2023** | **Tester: Huzaifa Pathan** |
| **Result: passed** | |

|  |  |
| --- | --- |
| **Use Case ID:** 6 | **Use Case Name:** Modify Account |
| **Test Number:** 6.1 | |
| **Objective:**  Modifies an existing user account. | |
| **Set Up:**  The user is logged in, the user has selected an account to modify, the user has pressed the modify account button. | |
| **Expected results:**  The user should be able to modify the selected account. | |
| **Test:**  1.Account is retrieved from the database.  2. The user is shown an interface that allows them to enter all the personal information.  3. If the selected role is TravelAdvisor: commission percentage is allowed to be changed.  4. The user changes the information that they want to  5. The user presses the submit button  6. The information is updated | |
| **Test Record:**  The expected behaviour is observed. | |
| **Date:** 06/04/2023 | **Tester:** Samina Matin |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case ID:** 6 | **Use Case Name:** Modify Account: Password Not Matching |
| **Test Number:** 6.2 | |
| **Objective:**  The password and confirm password text boxes do not have matching contents. | |
| **Set Up:**  The modify account button was pressed, the text in the password and confirm password textboxes do not match. | |
| **Expected results:**  The should get an error message “The passwords do not match”. | |
| **Test:**  1. The user changes the information that they want to  2. The user doesn’t put the same password in both fields  3. The user presses the submit button  4. The information does not update, an error massage popped up “passwords do not match”.  5. The account is not modified. | |
| **Test Record:**  The expected behaviour is observed. | |
| **Date:** 06/04/2023 | **Tester:** Samina Matin |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case ID:** 6 | **Use Case Name:** Modify Account: Cancelled |
| **Test Number:** 6.3 | |
| **Objective:**  The modify account operation was cancelled. | |
| **Set Up:**   1. User has signed in 2. Account to modify has been selected | |
| **Expected results:**  Modifying the selected account should be cancelled. | |
| **Test:**  1. The user changes the information that they want to  2. The user presses the cancel button.  3. GUI is closed. | |
| **Test Record:**  The expected behaviour is observed. | |
| **Date:** 06/04/2023 | **Tester:** Samina Matin |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case ID:** 6 | **Use Case Name:** Modify Account: Connection Unavailable |
| **Test Number:** 6.4 | |
| **Objective:**  The database connection was unavailable for storage or querying. | |
| **Set Up:**  User is signed in | |
| **Expected results:**  The system should show an error on screen for “Database Unavailable.”. | |
| **Test:**  1. Account is retrieved from the database.  2. Update button is pressed  3. The system shows an error on screen for “Database Unavailable.”.  4. Modifying the selected account doesn’t work as the connection is unavailable. | |
| **Test Record:**  The expected behaviour is observed. | |
| **Date:** 07/04/2023 | **Tester:** Samina Matin |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case ID:** 7 | **Use Case Name:** Delete Account |
| **Test Number:** 1 | |
| **Objective:**  Deletes the user account that no longer works in the company. | |
| **Set Up:**  The user is logged in and the selected account to be deleted is selected | |
| **Expected results:**  The selected account should be deleted. | |
| **Test:**  1.Account is retrieved from the database.  2. The user is shown an interface that allows them to select which account to delete.  3. The user deletes the account that they want to.  4. The user presses the delete button.  5. The account is deleted. | |
| **Test Record:**  The expected behaviour is observed. | |
| **Date:** 07/04/2023 | **Tester:** Samina Matin |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case ID:** 7 | **Use Case Name:** Delete Account: Advisor Assigned To Blanks |
| **Test Number:** 7.1 | |
| **Objective:**  If advisor is attempting to blanks, deletion fails. | |
| **Set Up:**  The advisor is assigned to a blank in stock manager. | |
| **Expected results:**  Deletion is stopped. | |
| **Test:**  1.Account is retrieved from the database.  2. The user is shown an interface that allows them to select which account to delete.  3. The user deletes the account that they want to.  4. The user presses the delete button.  5. The account is not deleted because the advisor is assigned to blanks. | |
| **Test Record:**  The expected behaviour is observed. | |
| **Date:** 07/04/2023 | **Tester:** Samina Matin |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case ID:** 8 | **Use Case Name:** CreateBlankStock |
| **Test Number:** 8.1 | |
| **Objective:** Allows for blanks to be input into the system | |
| **Set Up:**  System has a stock type with a type code and a staff Advisor to be assigned to the blank. | |
| **Expected results:** A blank was successfully created with a blank type | |
| **Test:**   1. Press “Create” button in stock manager tab 2. Assign an advisor with ID 101 3. Assign stock type 100 4. The blank shows up in stock | |
| **Test Record:** Expected behaviour observed | |
| **Date: 14/04/2023** | **Tester: Maudud Ahmed** |
| **Result: Pass** | |

|  |  |
| --- | --- |
| **Use Case ID:** 8 | **Use Case Name:** CreateBlankStock - no staff advisor in staff list |
| **Test Number:** 8.2 | |
| **Objective:** Blanks cannot be created if there is no advisor available | |
| **Set Up:**  System has a stock type with a type code | |
| **Expected results:** Blank creator does not pop up | |
| **Test:**   1. Press “Create” button in stock manager 2. Popup to create blank does not appear | |
| **Test Record:** Expected behaviour observed | |
| **Date: 14/04/2023** | **Tester: Maudud Ahmed** |
| **Result: Pass** | |

|  |  |
| --- | --- |
| **Use Case ID:** 8 | **Use Case Name:** CreateBlankStock - no stock types available |
| **Test Number:** 8.3 | |
| **Objective:** Blanks cannot be created if there is no stock type available | |
| **Set Up:**   1. System does not have a stock type | |
| **Expected results:** Blank creator does not pop up | |
| **Test:**   1. Press “Create” button in stock manager tab 2. Popup to create blank does not appear | |
| **Test Record:** Expected behaviour observed | |
| **Date: 14/04/2023** | **Tester: Maudud Ahmed** |
| **Result: Pass** | |

|  |  |
| --- | --- |
| **Use Case ID:** 8 | **Use Case Name:** CreateBlankStock - no advisor assigned |
| **Test Number:** 8.4 | |
| **Objective:** Allows for blanks to be input into the system with unassigned state | |
| **Set Up:**  System has a stock type with a type code and staff advisor | |
| **Expected results:** A blank was successfully created with a blank type but with unassigned state | |
| **Test:**   1. Press “Create button” in stock manager tab 2. Assign stock type 100 3. The blank shows up in stock but with the state “unassigned” | |
| **Test Record:** Expected behaviour observed | |
| **Date: 14/04/2023** | **Tester: Maudud Ahmed** |
| **Result: Pass** | |

|  |  |
| --- | --- |
| **Use Case ID:** 9 | **Use Case Name:** AccessBlankStock |
| **Test Number: 9**.1 | |
| **Objective:** Access the stock of blanks in the database | |
| **Set Up:**  stock contains 10 blanks with 3 different stock types 100, 901, 553 | |
| **Expected results:** all blanks should appear in stock | |
| **Test:**   1. Created 10 blanks: 2 blanks stock type 100, 3 blank stock type 553, 5 blank stock type 901 and 2 blanks unassigned. 2. Pressed “refresh” button 3. All blanks appeared in stock with accurate information in the stock manager tab. | |
| **Test Record:** Expected behaviour observed | |
| **Date: 14/04/2023** | **Tester: Maudud Ahmed** |
| **Result: Pass** | |

|  |  |
| --- | --- |
| **Use Case ID:** 10 | **Use Case Name: RecordRevokeBlank** |
| **Test Number: 10**.1 | |
| **Objective: Revokes blanks to stop them being sold or used by the system** | |
| **Set Up:**  System has blanks to be blacklisted | |
| **Expected results:** Blank state is set to blacklisted | |
| **Test:**   1. Select blank with ID 901000002 in stock manager tab 2. Select blank list 3. Popup warning 4. Select yes 5. Blank state appears as blacklisted and can no longer be used or sold in the system | |
| **Test Record:** Expected behaviour observed | |
| **Date: 14/04/2023** | **Tester: Maudud Ahmed** |
| **Result: pass** | |

## **Non-Functional**

**Testing Security**

The objective of a security test would be to see if a user can access parts of the program they should not be able to access, through means of an error in what components have what permissions or through malicious entry of an account with the correct permissions.

Performing a check of which components should be accessible would be done with the design report’s Use Case Diagram and an account for each of these actors to check which use cases they have access to and whether those use cases, corresponding to the components, should be accessible. For example, managers can assign blanks but not change what kind of blanks are available nor add blanks.

Performing a brute force check would be done by modifying the source code to attempt to authenticate against an account through all known characters that can be in a password. This should take a very long time to get into accounts with secure passwords and allow the identifying of accounts with weak passwords. The AccountController provides the “authenticateAccount” function for logins which can be used for this task.

**Testing Reliability**

The objective of a reliability test would be to see if the program can handle concurrency and unstable connections.

Performing a concurrency check would require two instances of the program to be running with 2 different users doing similar tasks, if the tasks to not conflict and cause desynced data, this would mean the test succeeded. Of Note: Users should get used to pressing the “refresh” button.

Performing tests on unstable connections can be tried out by either sending large amounts of data to or from the machine the program is running on in order to test how it reacts to low throughput (It should still operate but slower) or by cutting the connection itself in and out (This may not work if the connection times out, as unfortunately, the software has no support to recreate a dropped connection that has timed out). After running these tests, the data can be checked for corruption or desync and if it does not have either of these issues, the software can be considered reliable for data storage.

# **Review**

In review, Skyway Solutions apologies for an incomplete prototype but you can be assured that if we were given the chance to complete the prototype, it can be done so quickly due to the nature of the codebase. The existing software should also show that validation can be implemented and when created in good time, the interfaces can be quite robust.