Graded Unit 2 dEVELOPMENT

Max Ritchie - 545737

BCSC2/FK

Forth Valley College

Contents

[1. Application 2](#_Toc8164138)

[1.2. Libraries and Constructs 3](#_Toc8164139)

[1.2.1. Unfamiliar Libraries 3](#_Toc8164140)

[1.2.2. Constructs 5](#_Toc8164141)

[1.3. Error Handling 6](#_Toc8164142)

[2. Testing 7](#_Toc8164143)

[2.1. Test Plan 7](#_Toc8164144)

[2.1.1. Test Strategies 7](#_Toc8164145)

[2.1.1.1. Business Model 7](#_Toc8164146)

[2.1.1.2. View Model 7](#_Toc8164147)

[2.1.2. Tests To be Run 7](#_Toc8164148)

[2.1.2.1. Test Plan Tables 8](#_Toc8164149)

[2.1.2.2. Blank Test Logs 14](#_Toc8164150)

[2.1. Test Runs 17](#_Toc8164151)

[2.1.1. Test Run One 17](#_Toc8164152)

[2.1.2. Test Run Two 20](#_Toc8164153)

[3. User Documentation 23](#_Toc8164154)

[3.1. User Manual 23](#_Toc8164155)

[3.1.1. Installation Process 23](#_Toc8164156)

[3.1.2. User Navigation 23](#_Toc8164157)

[3.1.3. Login Process 24](#_Toc8164158)

[3.1.4. Displaying Reports 25](#_Toc8164159)

[3.1.4. Printing Reports 26](#_Toc8164160)

[3.1.5. Logging Out 27](#_Toc8164161)

[3.1.6. Closing the Application 28](#_Toc8164162)

[4. Information Sources 29](#_Toc8164163)

# 1. Application

In the inception phase of the project I proposed the idea of a fully working application, with the time frame that I had I was only able to produce a prototype of the application with the main functions of the application working with pulling information out of the database and displaying it. Having to carry-out a lot of research which I referenced in previous reports, found in section *Graded Unit - Inception Phase Report – 1.7.3. Other Resource References* and in section *4. Information Sources*, a lot of time was spend learning new software and libraries which I discuss in section *1.2.1. Unfamiliar Libraries*. Not knowing the amount of learning and research I had to undertake I was unable to provide what I planned on making but what I have produced is the main functionality of the application is to display reports and invoices to the user with the option to print them out.

## 1.2. Libraries and Constructs

### 1.2.1. Unfamiliar Libraries

In the development and the inception phases of my project I used new techniques and software to plan out and create my application. Down below is all the new software along with the new libraries and constructs that I had to use in order to complete my application:

JavaFX  
The programming language that I chose to create my application in was JavaFX. The reason that I chose this language was that I has a good knowledge and understanding of programming in regular Java, also I know that Java contains a wide range and variety of libraries that I can pick and choose from. JavaFX is a software platform that is used to create and develop desktop applications. The intent of JavaFX is to replace Swing (another Java software platform) as the Standard GUI (Graphical User Interface) library for Java development. In the past I have previously used Swing to create Java applications, so I decided to chose JavaFX as the platform I would use. As I mention in my previous documentation: *Graded Unit - Inception Phase Report – 1.6.2. Software Requirements*, by referencing different resources that I would be using to learn how to use the new libraries, I managed to carry-out further research and with the aid of online forums and video tutorials, I was able to have a greater understanding and have the ability to carry-out new tasks that I would previously not have been able to do. With the research that I carried out I was able to create that main parts of my application that I wanted to achieve, even though I did not complete my application fully, I was able to complete a fully functioning prototype.

Using JavaFX, I had the ability to use a variety of different and new libraries that I have never used before. Creating the UI for my application I had to import each item/element into the source code. For each FX element (variables and methods) a @FXML tag had to assigned to it. With use of adding elements from a database, I needed somewhere to store all entities. To do this I had to use **javafx.collections.ObseravableList**, which allows me to create a list to store all entities held in a selected table in the database. I also had to implement more libraries to take and display data from the database. The other libraries I used were **javafx.scene.control.TableColumn** and **javafx.scene.control.TableView**. TableView creates a table view with the use of TableColumn which display different columns inside the table view. When created the ObservableList can be added into the table and display all data held in the list that has been pulled from the database.All these libraries are used and can be used in **Graded Unit/src/account/AccountPageController.java**.

Asana  
At the beginning of the project (The Inception Phase), I was required to create a project plan to monitor my timing and progress with each stage of the project. To start with I wanted to create a plan using a new online application: Asana. Asana is a web-based application which is designed to track and manage work, teams in organizations would tend to use Asana as their planning application. Carrying-out research on how to use the new online software, I managed to create headings for each section and assigned all required tasks, but this was only the basic functions that I was able to use. I could not generate a required Gantt chart using the free version of Asana. Having emailed Asana stating that I am a student and can be intitled to the student version of Asana, which would grant me access to the required features that I needed to complete of fully required project plan. Unfortunately, I was unable to gain access to the student account and had to resort to another resource to create my project plan. Fortunately, in the past I had previously used Microsoft Project to create similar project plans and I was able to use my previous knowledge and skills to create a fully finished project plan along with the required Gantt chart that I was not able to produce before.

SQLite  
My initial plan was to use UwAmp and phpMyAdmin to create and manage my database. Before I started the creation of my database, I was shown SQLite and I decided to change my plan and use it instead. SQLite is a relational database management system, the difference with SQLite and other database management systems is that SQLite is not a client-side database engine, nothing needs to be run at the same time as the application for the database to work. The database is embedded into the program and gets called by one line of code. Being a software that I have never used before, I had to carryout research by reading online help forums and watching tutorials on how to create and implement the database into my application. Having researched how to use SQLite I managed to create my required database with ease as it was straight forward to use and implement into the application.

Using SQLite was totally new to me, I have used similar software in the past, but they all operated differently than SQLite, also I have never implemented a database into a Java program. This means that I had to use new libraries that I have never used before. I decided to create and connect a database to the application over different storage methods because I wanted to try something new and I thought it would work better to store large amounts of data over other storage methods. Using the database means that updates will be applied a lot quicker when data is added or modified.

I had to use a few new SQL libraries in order to have to database connect to my application and view the data within the database. The first and most important library that I had to use was **java.sql.Connection**. This library must be used in order for the application to connect to the database, otherwise the application will not work. Another library that was used was **java.sql.DriverManager**, this library works along-side connecting the application to the database. This allows the database to find the required connection of where the jdbc driver is located and establishes the connected between the application and the database. These libraries are used and can be found in **Graded Unit/src/database/DatabaseConnection.java**.

In Order to pull information from the database I had to use queries to find the location of the required information in the database. One of the new libraries that I used was **java.sql.sqlPreparedStatement**. This library was needed for holding the required SQL query that would be carried out. After executing the query I needed to use another library called **java.sql.sqlResultSet**. This library got used to capture and hold the results from the query that was run. Being stored in the result sets allows the application to use the data from the database. The last final library that I needed to use for all the SQL parts of the application was **java.SQLException**. Each sections of code that held any SQL code is surrounded by try catches. Included in each catch there is a SQLException that will be catch any invalid SQL commands or errors that occur. All of these SQL libraries that I have discussed can be found in **Graded Unit/src/account/AccountPageController.java**.  
  
Scene Builder  
As I am using JavaFX, to create the UI (User Interface) I decided to use Scene Builder as it works with JavaFX and in the past I had already worked with Swing so I had chose a new library to use that I haven’t previously used. Scene Builder is a visual layout tool that allows user to quickly design and create JavaFX application UI. Scene Builder produces FXML files that can be included into Java projects by binding the UI to the applications logic code. Having used Swing in past, I was familiar with designing and creating UI in Java as JavaFX and Swing use similar ways of creating and design interfaces. Being new to me, I had to go online and watch tutorials on how use Scene Builder and how to correctly design and layout everything to have it working properly. I also had to research how to implement the interfaces into the application which was very straight forward with the use of online forums and video tutorials.

### 1.2.2. Constructs

With all the new libraries and software that I used to create the application and database, I had to learn how to do use and understand all the new features that I needed. I carried out a lot of research by visiting online help forums and watching tutorials on various websites and YouTube, all resources that I have used are listed in section *4. Information Sources*.

The first construct that I am going to discuss is the login process of the application. Having never created or done anything like this before I had to start from scratch by looking online for resources to help me with the process. I had to take in login credentials from the user and accept it if the credentials are valid and match what is stored within the database, this is done by using created queries that take in and use the inputted user credentials. This process can be found in the code in **Graded Unit/src/database/DatabaseManager.java – isLogin on Line 67**.

The next construct that I am discussing plays a big part in the main feature of the application and that is the ability to display information from the database to the application. Having using JavaFX this meant that I had to create and use queries to search and pull required information out of the database. Once found, the data is then held into a list which is then used on a table view that display all information in columns which are also added into the table view by a table column which is created and initialised with all required column headings. In **Graded Unit/src/account/AccountController.java** is where I fill all my tables with the required information from the database, when called the selected table will be filled, ready to be displayed to the screen when called upon.

## 1.3. Error Handling

Within the application I used a wide range of different and new error handling exceptions that I have not used before. I created a short list of the different types of error handling that I used along with a short explanation of why I used and where about in the application:

1. Exception  
This is used to stop the application from crashing at any point, I have also included to display the stack trace in the console to see what the error is and where it is getting caused. This can be found in any catch statement in the source code.

2. NullPointerExecption  
This error handling exception deals with any errors that fail to find anything that is required to be displayed or shown to the screen. I have used this exception in multiple places in the application byt I am going to discuss the one that is located at **Graded Unit/src/applicaiton/Driver.java– Line 50**. This is included here to display an error message to the screen if there is an error with finding the correct page when loading up the application.

3. LoadExecption  
This was used to display error messages if a page could not load correctly, it will display an error message to the screen informing the user there is a problem and tell them how to resolve it. I have used this exception throughout my application, but I am going to talk about the one located in **Graded Unit/src/applicaiton/Driver.java Graded Unit/src/database/DatabaseManager.java – Line 61**. This is here to display an error message to the screen to inform the user that the main page cannot be displayed and will tell the user how to fix the problem and what actions to carry-out.

4. SQLExecption  
I was required to use this error handling exception for all the methods that housed the SQL query statements. If any invalid queries are found or any issues arise, this exception will be produced. I have use this exception in each catch statement in **Graded Unit/src/account/AccountController.java** for the display methods that hold any SQL query statements. If an error did occur, an error message if displayed to the screen informing the user of the error that has occurred.

# 2. Testing

## 2.1. Test Plan

2.1.1. Test Strategies  
For the testing of the application, I will be carrying out two different types of testing – black box and white box testing. Black box testing will allow me to test the application how any user would use it, it only includes the testing of the functionality of the application without looking at the internal code and structure. White box testing allows me to test the internal structure of the application, I must have a good knowledge and understanding of what everything does to properly carryout testing. For me to carry out these tests, I have created a test plan of everything I am going to test and what the excepted results should be, these test plans are shown in section *2.1.2.1. Test Plan Tables*.

#### 2.1.1.1. Business Model

The way that I am going to test the business model of the application is to go with the created test plans in section *2.1.2.1. Test Plan Tables*. I am going to be thoroughly testing where user input is required and how the application will respond with different input. I am carryout integration testing to make sure that all classes that are used within the application works properly together when run. All results of testing will be shown in the section *2.1. Test Runs*.

#### 2.1.1.2. View Model

With the testing of the business model, I am also going to be testing the view model of the application to make sure that the UI (User Interface) works as intended. I am testing the interface to ensure that all functional aspects and features works as they are meant to. Within this stage of testing I am going to be running similar tests as the business model, but they are more aimed towards on the user interface and how the application acts on how certain actions are done and carried-out. I will be testing all available buttons that are supposed to work and the navigation around the application.

### 2.1.2. Tests To be Run

In this section I am creating and showing the test that will be run and tested on the application. I am going to be splitting all tests into different sections to show which class/page is being tested. I am going to create a table of my test plan, what is going to be tested and the purpose of the test along with the action to be taken and the excepted result. I am also going to create another table with will show a blank test which will be used and filled out in the next section *2.1. Test Runs.* I have also created a blank template of a test log (found in section *2.1.2.2. Blank Test Logs*) which will be used and filled in when the application is being tested. The filled out and completed test logs are found in section *2.1. Test Runs*. I am also going to be including all test runs including failed tests, the outcomes and how I fix the issues and errors then re running the tests until they all work correctly.

Being a prototype, not everything in the application works so I have only included and created tests for the working functions of the application prototype. I am including the view model and business model together as they both test key features in the application. All tables that I have created are shown below.

#### 2.1.2.1. Test Plan Tables

|  |  |  |  |
| --- | --- | --- | --- |
| **Start Up** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 1. | Check if the applications runs | Clicked on the runnable jar file | Application runs correctly and homepage displays to the screen |
| 2. | Check that the database is connected properly | Clicked on the runnable jar file | Application runs correctly with no error messages being displayed |

|  |  |  |  |
| --- | --- | --- | --- |
| **Homepage** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 3. | Check if the dropdown menu works correctly on the music tab | Clicked on the music tab in the menu | Drop down menu appears displaying the CDs and Vinyl tabs |
| 4. | Check if the home button works correctly | Clicked on the home button in the menu | Refreshes the homepage and continues to display the homepage to the screen |
| 5. | Check if the login/register button works correctly | Clicked on the login/register button in the menu | Closes the homepage and displays the login page to the screen |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Page** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 6. | Check if the home button works correctly | Clicked on the home button | Closes the login page and displays the homepage to the screen |
| 7. | Check if the login function works correctly | Entered username (Admin) and password (Admin) into the username and password fields and clicked the login button | Closes the login page, logins the user in and displays the users account page to the screen |
| 8. | Check if the register link works correctly | Clicked on the register link | Closes the login page and displays the register page to the screen |
| 9. | Check if the username field takes in input | Clicked on the username input field and typed in credentials | Username field takes in input from the user |
| 10. | Check if the password field takes in input and hides the password | Clicked in the password input field and typed in credentials | Password field takes in input from the user and is hidden, unable to see what characters are entered |

|  |  |  |  |
| --- | --- | --- | --- |
| **Register Page** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 11. | Check if the home button works correctly | Clicked on the home button | Closes the register page and displays the homepage to the screen |
| 12. | Check if the given name field takes in input | Clicked on the given name input field and typed in credentials | Given name field takes in input from the user |
| 13. | Check if the last name field takes in input | Clicked on the last name input field and typed in credentials | Last name field takes in input from the user |
| 14. | Check if the email address field takes in input | Clicked on the email address input field and typed in credentials | Email address field takes in input from the user |
| 15. | Check if the phone number field takes in input | Clicked on the phone number input field and typed in credentials | Phone number field takes in input from the user |
| 16. | Check if the username field takes in input | Clicked on the username input field and typed in credentials | Username field takes in input from the user |
| 17. | Check if the password field takes in input and hides the password | Clicked on the password input field and typed in credentials | Password field takes in input from the user and is hidden, unable to see what characters are entered |

|  |  |  |  |
| --- | --- | --- | --- |
| **Account Page** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 18. | Check if the home button in the menu works correctly | Clicked on the home button | Closes the account page and displays the homepage to the screen |
| 19. | Check if the home button in the account details box works correctly | Clicked on the home button | Closes the account page and displays the homepage to the screen |
| 20. | Check if the combo box ‘Select Table’ drops down and displays all available tables | Clicked on the select table combo box | Combo box menu drops down and displays all the available tables |
| 21. | Check if the ‘Employee Information’ table gets displayed with the correct headings in the database viewer area | Selected the ‘Employee Information’ option in the select table combo box and clicked the search button | Employee Information table displays with the correct headings to the screen in the data base viewer area |
| 22. | Check if the ‘Customer Information’ table gets displayed with the correct headings in the database viewer area | Selected the ‘Customer Information’ option in the select table combo box and clicked the search button | Customer Information table displays with the correct headings to the screen in the data base viewer area |
| 23. | Check if the Invoices’ table gets displayed with the correct headings in the database viewer area | Selected the ‘Invoices’ option in the select table combo box and clicked the search button | Invoices table displays with the correct headings to the screen in the data base viewer area |
| 24. | Check if the ‘Suppliers Information’ table gets displayed with the correct headings in the database viewer area | Selected the ‘Supplier Information’ option in the select table combo box and clicked the search button | Suppliers Information table displays with the correct headings to the screen in the data base viewer area |
| 25. | Check if the update table button works correctly | Clicked the update table button | Closes the account page and displays the update info pop up page to the screen |
| 26. | Check if the print button works when employee information has been selected and displayed | Clicked the print button when employee information is being displayed to the screen | Print pop up displays to the screen ready for print confirmation, prints out the last entry in employee information table in a customized format |
| 27. | Check if the print button works when customer information has been selected and displayed | Clicked the print button when customer information is being displayed to the screen | Print pop up displays to the screen ready for print confirmation, prints out the last entry in customer information table in a customized format |
| 28. | Check if the print button works when invoices has been selected and displayed | Clicked the print button when invoices is being displayed to the screen | Print pop up displays to the screen ready for print confirmation, prints out the last entry in invoices table in a customized format |
| 29. | Check if the print button works when suppliers information has been selected and displayed | Clicked the print button when suppliers information is being displayed to the screen | Print pop up displays to the screen ready for print confirmation, prints out the last entry in suppliers information table in a customized format |
| 30. | Check if the correct error message displays when no table has been selected when searching | Clicked the search button without selecting a table from the select table combo box | No Report Selected error message is displayed to the screen |
| 31. | Check if the correct error message displays when no table has been selected when printing | Clicked the print button without selecting a table from the select table combo box | No Report Selected error message is displayed to the screen |
| 32. | Check if the logout button works correctly | Clicked the logout button in the menu | Logs the user out and closes the account page then displays the homepage to the screen |

|  |  |  |  |
| --- | --- | --- | --- |
| **Update Pop Up Page** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 33. | Check if the back button works correctly | Clicked on the back button | Closes the update pop up page and displays the users account page to the screen |

|  |  |  |  |
| --- | --- | --- | --- |
| **Additional Features** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 34. | Check if the X exit button works correctly | Clicked the exit X button on the window in the trop right-hand corner | Closes and shuts down the application |
| 35. | Check if the maximize button doesn’t do anything | Clicked the maximize button on the window in the trop right-hand corner | Nothing happens, the application stays the same size |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Entries** | | | |
| **Test No.** | **Purpose of Test** | **Action/Input** | **Expected Result** |
| 36. | Check if the login displays an error message | Entered 123 in the username field and 123 in the password field and click the login button | Error message displays to the screen |
| 37. | Check if the login displays an error message | Leave the username and password field blank and clicked the login button | Error message displays to the screen |

#### 2.1.2.2. Blank Test Logs

Below, I have created a blank test log which will be used to document all testing that I have planned out in the test plan. All test shown in the test log are crosses referenced with the test plan above in section *2.1.2.1. Test Plan Tables* to show what is being tested and why the test is being carried, also to compare expected results with the actual results.

|  |  |  |  |
| --- | --- | --- | --- |
| **Start Up** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 1. |  |  |  |
| 2. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Homepage** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 3. |  |  |  |
| 4. |  |  |  |
| 5. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Page** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 6. |  |  |  |
| 7. |  |  |  |
| 8. |  |  |  |
| 9. |  |  |  |
| 10. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Register Page** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 11. |  |  |  |
| 12. |  |  |  |
| 13. |  |  |  |
| 14. |  |  |  |
| 15. |  |  |  |
| 16. |  |  |  |
| 17. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Account Page** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 18. |  |  |  |
| 19. |  |  |  |
| 20. |  |  |  |
| 21. |  |  |  |
| 22. |  |  |  |
| 23. |  |  |  |
| 24. |  |  |  |
| 25. |  |  |  |
| 26. |  |  |  |
| 27. |  |  |  |
| 28. |  |  |  |
| 29. |  |  |  |
| 30. |  |  |  |
| 31. |  |  |  |
| 32. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Update Pop Up Page** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 33. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Additional Features** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 34. |  |  |  |
| 35. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Entries** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 36. |  |  |  |
| 37. |  |  |  |

## 2.1. Test Runs

I am going to be toughly testing all aspects, features and functions of the application. Each test run will be displayed below until all issues and problems that occur are fixed and the everything listed works as it’s supposed to:

### 2.1.1. Test Run One

My first test that I ran failed at the start. I was unable to run all tests bar one as the first test when trying to start the application. I document this below and discuss how I fixed the issue in my next test run:

|  |  |  |  |
| --- | --- | --- | --- |
| **Start Up** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 1. | Application does not run at all | Fail | Nothing is displayed to the screen when trying to run the jar file |
| 2. | N/A |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Homepage** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 3. | N/A |  |  |
| 4. | N/A |  |  |
| 5. | N/A |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Page** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 6. | N/A |  |  |
| 7. | N/A |  |  |
| 8. | N/A |  |  |
| 9. | N/A |  |  |
| 10. | N/A |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Register Page** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 11. | N/A |  |  |
| 12. | N/A |  |  |
| 13. | N/A |  |  |
| 14. | N/A |  |  |
| 15. | N/A |  |  |
| 16. | N/A |  |  |
| 17. | N/A |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Account Page** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 18. | N/A |  |  |
| 19. | N/A |  |  |
| 20. | N/A |  |  |
| 21. | N/A |  |  |
| 22. | N/A |  |  |
| 23. | N/A |  |  |
| 24. | N/A |  |  |
| 25. | N/A |  |  |
| 26. | N/A |  |  |
| 27. | N/A |  |  |
| 28. | N/A |  |  |
| 29. | N/A |  |  |
| 30. | N/A |  |  |
| 31. | N/A |  |  |
| 32. | N/A |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Update Pop Up Page** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 33. | N/A |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Additional Features** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 34. | N/A |  |  |
| 35. | N/A |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Entries** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 36. | N/A |  |  |
| 37. | N/A |  |  |

### 2.1.2. Test Run Two

To fix the problem that I encountered in my first test run was I had to change the run preferences of the application in eclipse. The application could not find the correct class name as it was spelled incorrectly. After that I had to install a newer version of Java to allow me to gain access to a run time environment that I did not have before. I searched online for help and tutorials which are found in section *4. Information Sources*. After carryout those two tasks I ran the tests again and below are the results:

|  |  |  |  |
| --- | --- | --- | --- |
| **Start Up** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 1. | Application runs | Pass |  |
| 2. | Database gets connected | Pass |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Homepage** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 3. | Dropdown menu displays | Pass |  |
| 4. | Homepage gets refreshed and displayed | Pass |  |
| 5. | Displays the login page | Pass |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Page** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 6. | Homepage displays | Pass |  |
| 7. | Logins into the application | Pass |  |
| 8. | Opens the registration page | Pass |  |
| 9. | Username field allowed me to type and takes in input | Pass |  |
| 10. | Password field allowed me to type and takes in input | Pass |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Register Page** | | | |
| **Test No.** | **Actual Result** | **Pass/Fail** | **Additional Comments** |
| 11. | Displays homepage | Pass |  |
| 12. | Given name field allowed me to type and takes in input | Pass |  |
| 13. | Last name field allowed me to type and takes in input | Pass |  |
| 14. | Email address field allowed me to type and takes in input | Pass |  |
| 15. | Phone number field allowed me to type and takes in input | Pass |  |
| 16. | Username field allowed me to type and takes in input | Pass |  |
| 17. | Password field allowed me to type and takes in input |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Account Page** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 18. | Homepage displays | Pass |  |
| 19. | Homepage displays | Pass |  |
| 20. | Drop down menu displays all available tables | Pass |  |
| 21. | Employees Information table displayed | Pass |  |
| 22. | Customer Information table displayed | Pass |  |
| 23. | Invoices table displayed | Pass |  |
| 24. | Suppliers Information table displayed | Pass |  |
| 25. | Update page displayed | Pass |  |
| 26. | Print pop up displayed | Pass |  |
| 27. | Print pop up displayed | Pass |  |
| 28. | Print pop up displayed | Pass |  |
| 29. | Print pop up displayed | Pass |  |
| 30. | No table selected error message displayed | Pass |  |
| 31. | No table selected error message displayed | Pass |  |
| 32. | Logouts out of the application | Pass |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Update Pop Up Page** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 33. | Displays the account page | Pass |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Additional Features** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 34. | Closes the application | Pass |  |
| 35. | Nothing happens | Pass |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Entries** | | | |
| **Test No.** | **Purpose of Test** | **Pass/Fail** | **Additional Comments** |
| 36. | Error message displayed | Pass |  |
| 37. | Error message displayed | Pass |  |

# 3. User Documentation

## 3.1. User Manual

I have created different sections for the user manual, each section will display a step by step walk through for the user to follow and have a greater understanding of how to use the application. Being a prototype not everything in the application works, only the main features that make the application function work so the manuals I have created with only show the user how to use the features that currently work. Below are the different sections of the user manual:

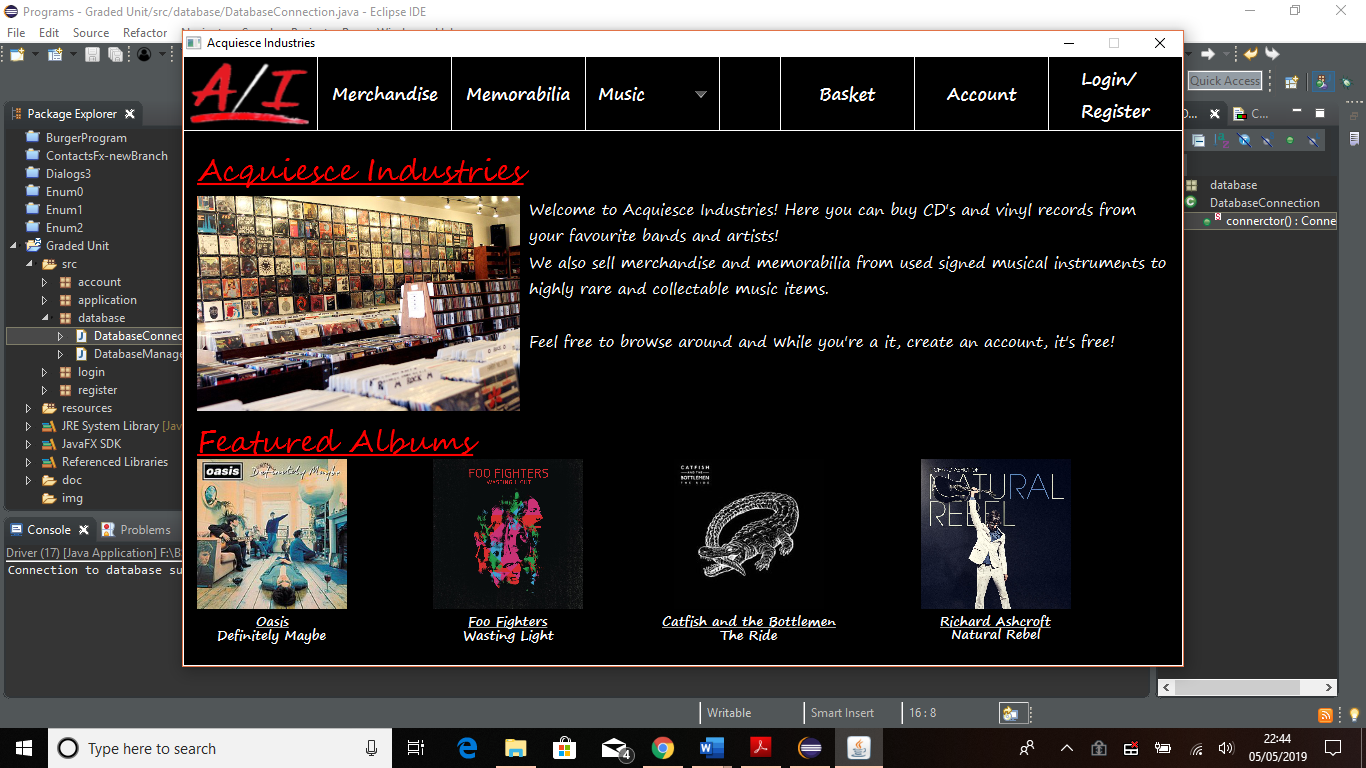
### 3.1.1. Installation Process

Step 1.   
The first step to installing the application correctly is to have the applications folder (Graded Unit) saved into your C: drive of the PC. The reasoning for this is to establish the connection with database as the database file directory goes through the C: drive to establish the connection.

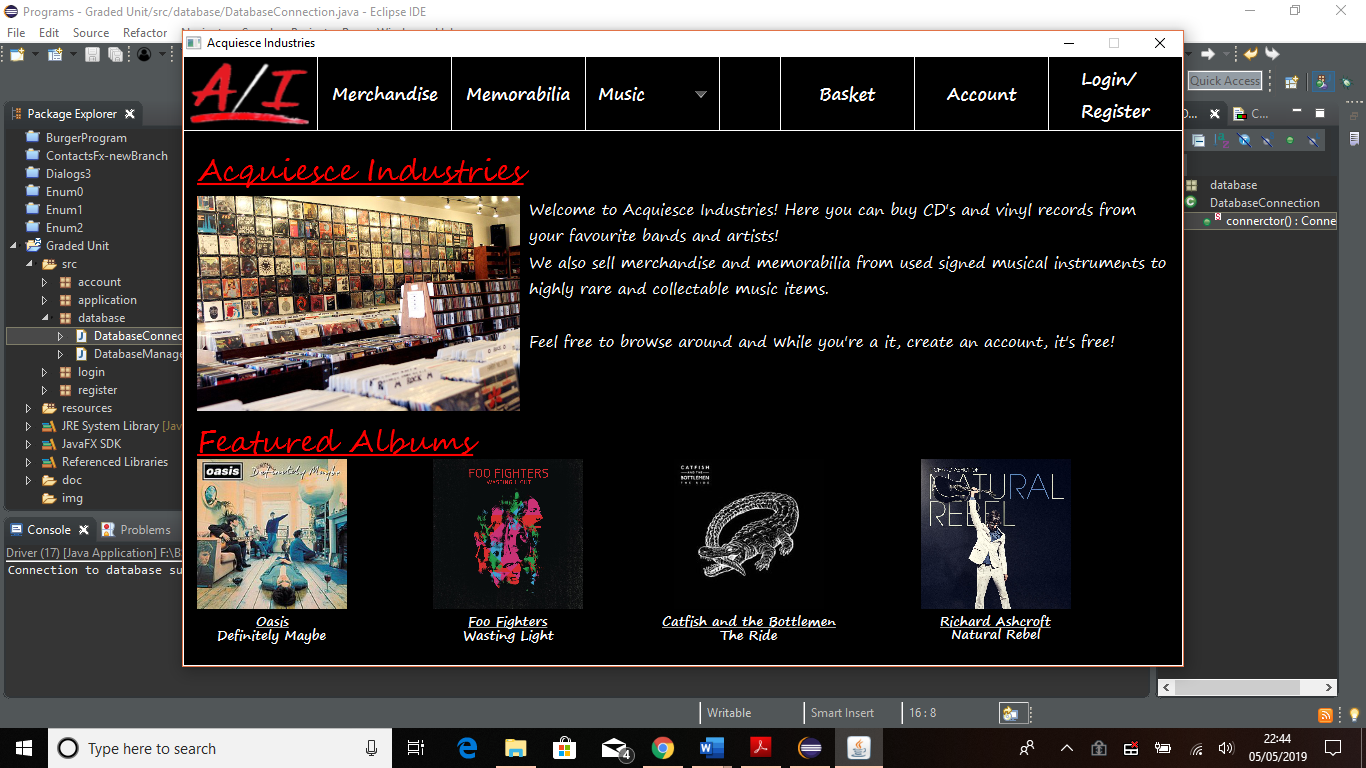
Step 2.  
After installing the application onto the C: drive, you can now open the open the folder and start the application and move onto the next sections.

### 3.1.2. User Navigation

Step 1. – Refreshing the Homepage  
After opening and loading the application, the homepage will display to the screen. To refresh the homepage, all that needs to be done is to click the homepage button (the company logo on left hand-side of the menu bar at the top of the screen is the home button). An image of the homepage and the location of the homepage button is located below.

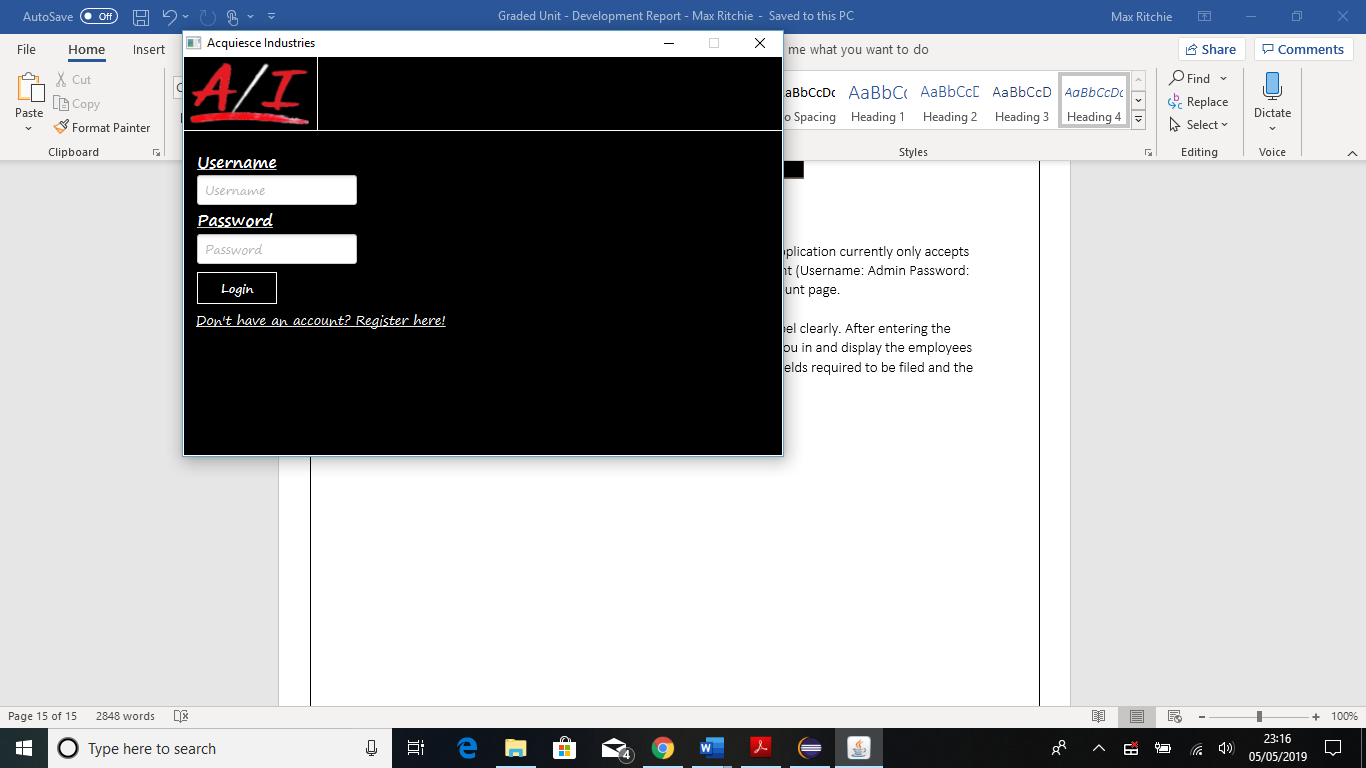
The Homepage   
Button

Step 2. – Accessing the Login Page  
In order to gain access to the account page and to the database, the user must login to the application. To access the login page, you must click on the **Login/Register** button which is located on the right-hand side of the menu bar at the top of the of the homepage. An image of the homepage and the location of the login button is shown below.

 Login/Register Button

### 3.1.3. Login Process

Step 1.  
The login process includes entering a username and password. The application currently only accepts employee accounts so that to login you must use an employee account (Username: Admin Password: Admin will log you into the application) to display the employees account page.  
  
Enter the username and password into the required fields that are label clearly. After entering the login credentials click the **Login** button and the application will login you in and display the employees account page. Below is an image of the login page with labels to the fields required to be filed and the button that needs to be clicked to login.



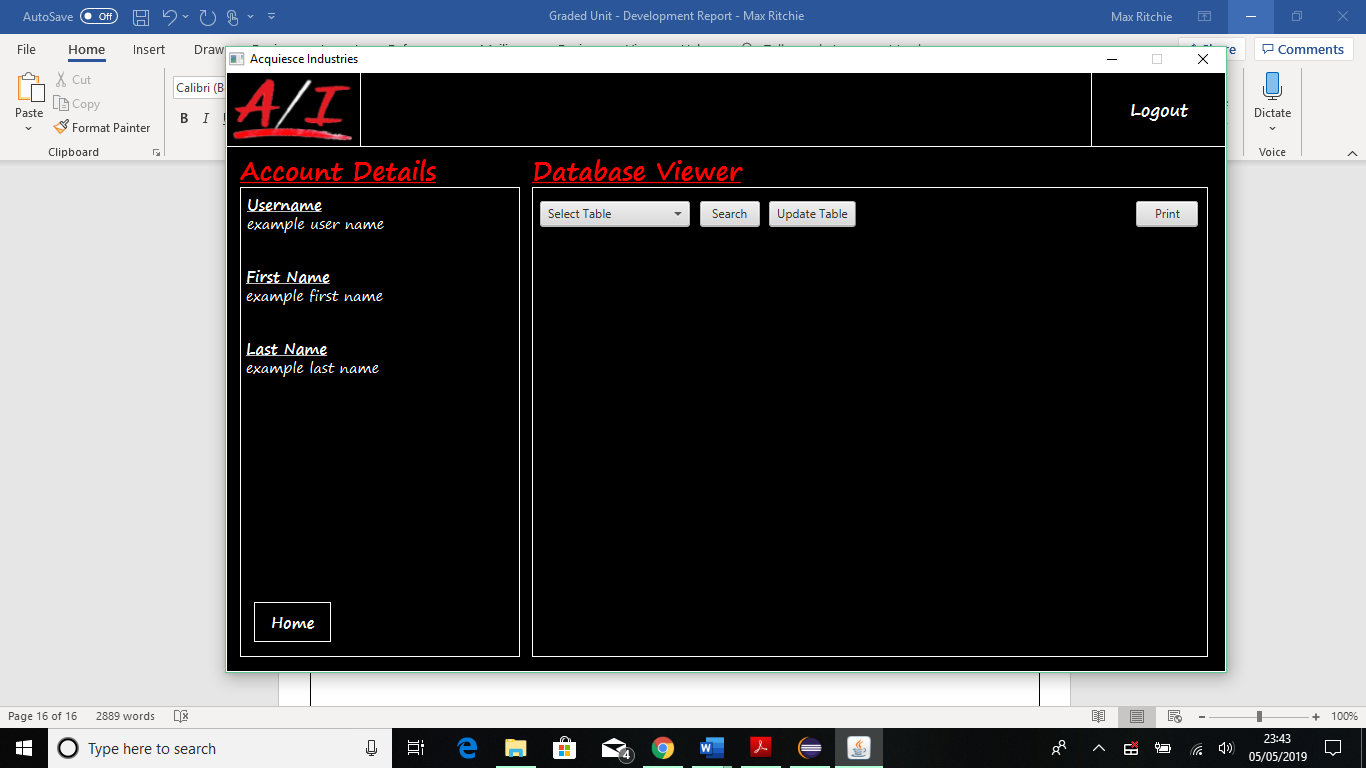
Username Field

Password Field

Login Button

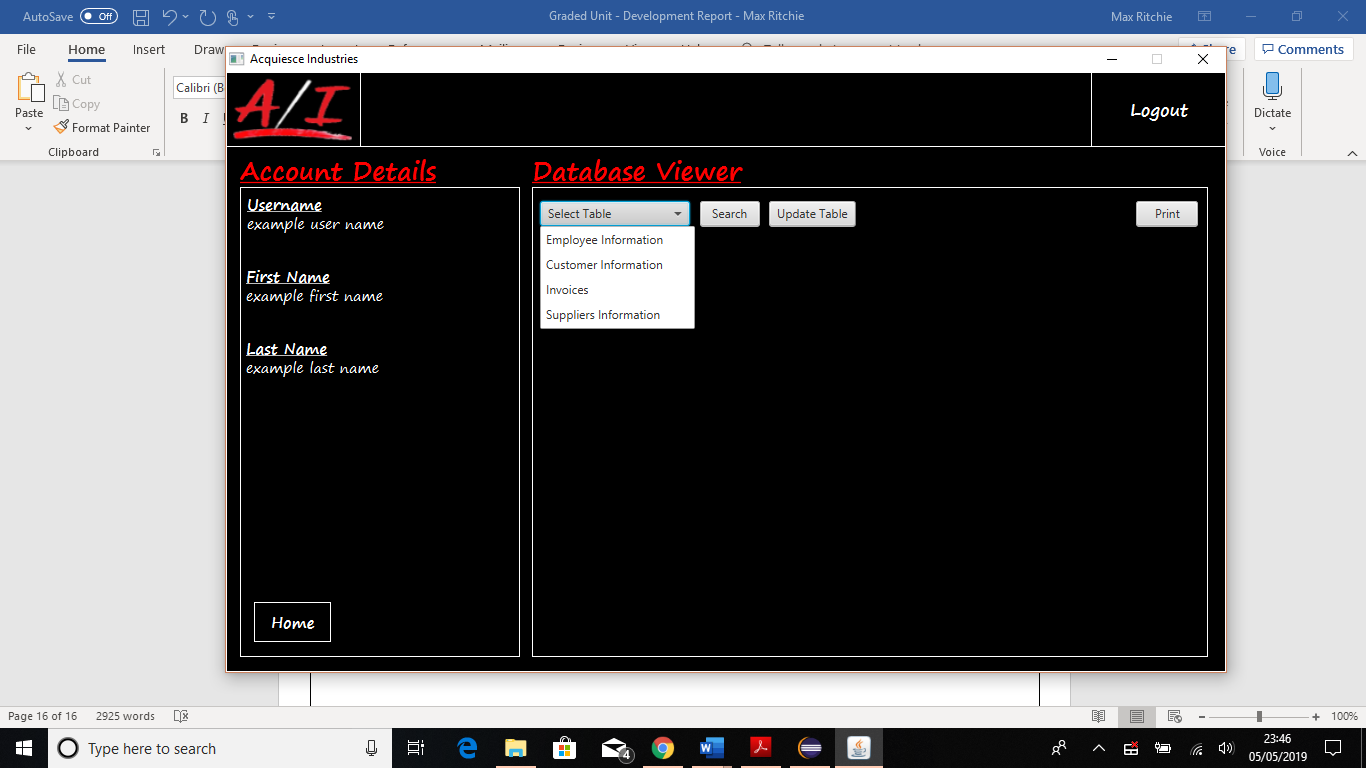
### 3.1.4. Displaying Reports

Step 1.  
After logging into the application, the account page will be displayed. The first step to displaying a report is to select a report from the **Select Table** drop-down menu.



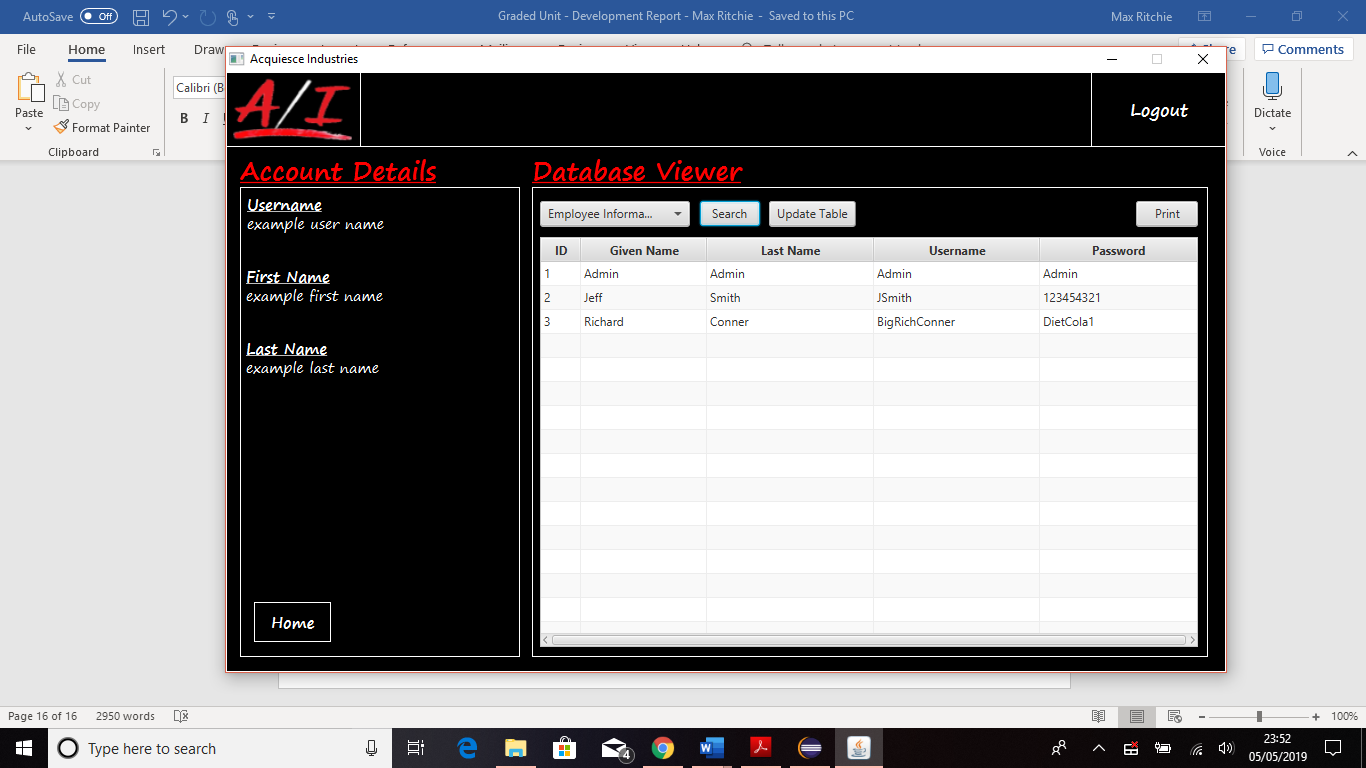
The dropdown that holds all available reports. When clicked the menu will drop down and show all options.

Step 2.  
When the drop-down menu is displayed, click on the required report that you wish to display.



Select required report to be displayed

Step 3.  
After selecting the required report, click the search button to display the requested report in a table view.

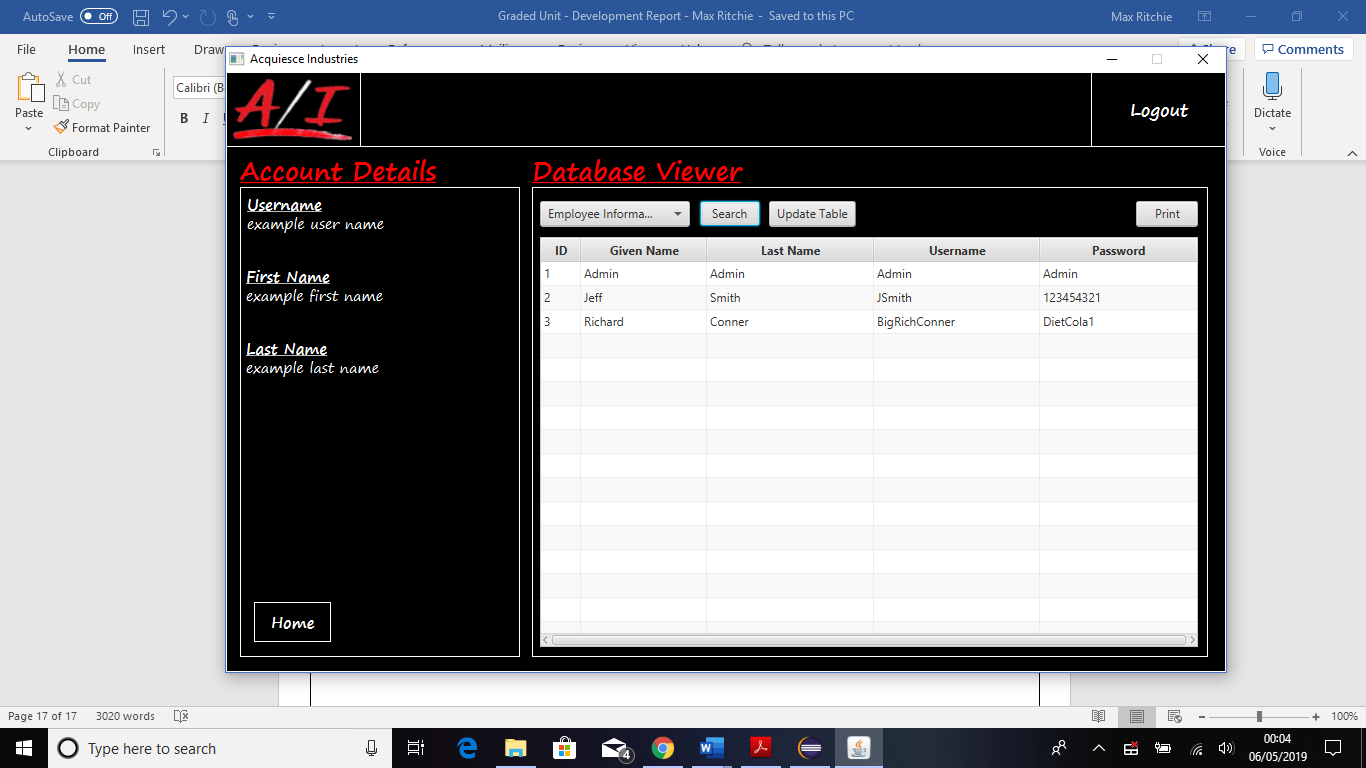


A report **MUST** be selected in order for the table view to be displayed to the screen, otherwise an error message will occur.

Other reports can be displayed to the screen if you repeat the process from *Step 2.*

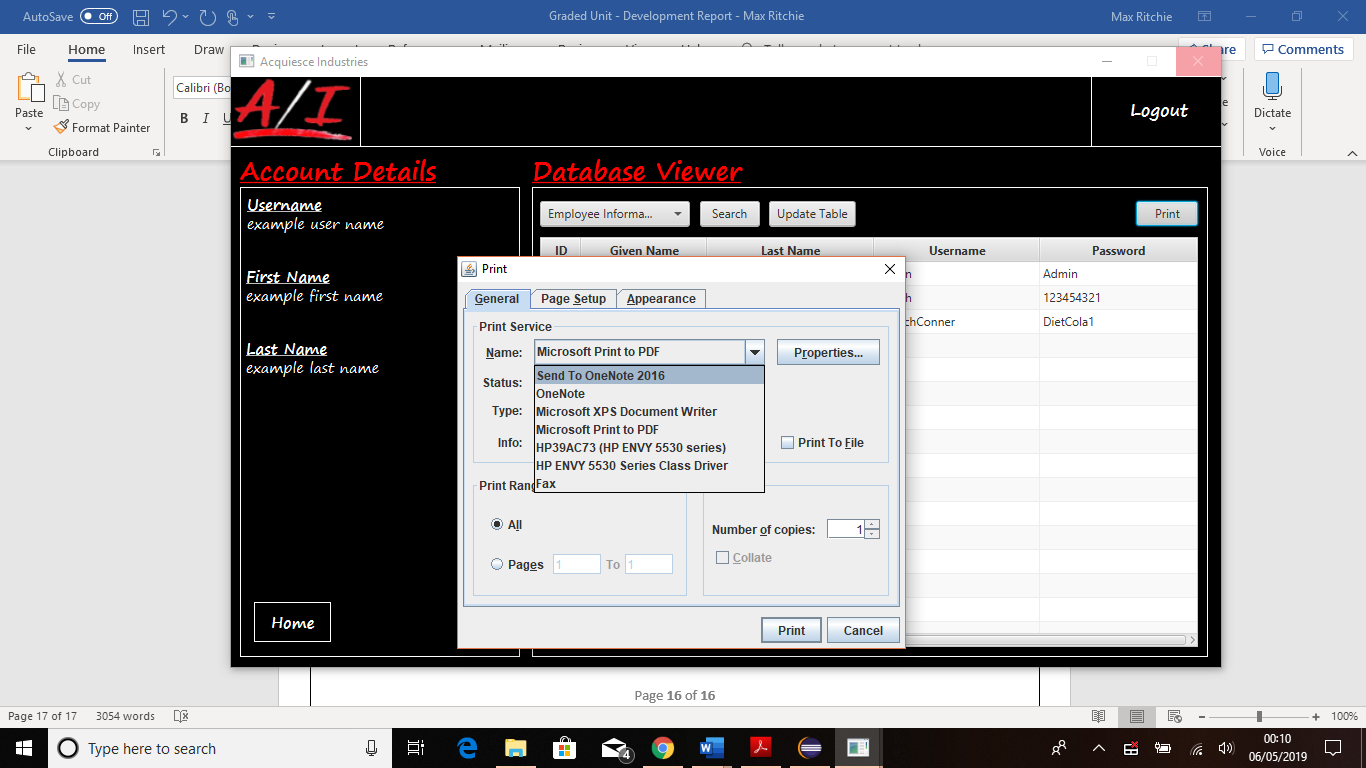
### 3.1.4. Printing Reports

Step 1.  
When having a report displayed to the screen, the print button is now functional. You can click the print button to print out your requested report.



Just like the search button, a report **MUST** be selected and displayed to the screen in order for the print function to work, otherwise an error will occur.

Step 2.  
When the print button has been clicked and the print function has been initialized, the print screen will be displayed to the screen. When displayed you can select your printer from the dropdown menu and click print when ready.

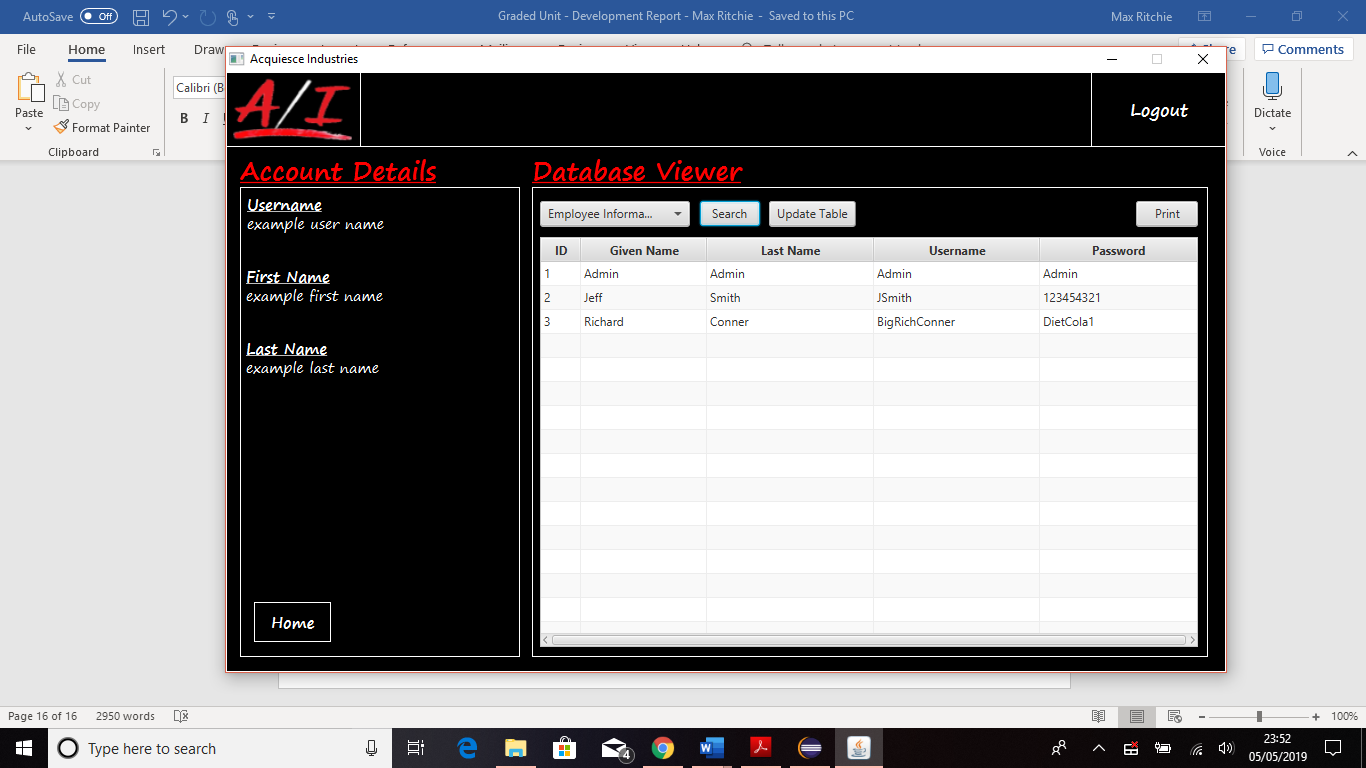


Dropdown menu allows you to select your printer.

When your printer had been slected, click the print button to print out your required report.

### 3.1.5. Logging Out

Step 1.  
To log out of the application all you need to do is click the logout button in the top right-hand side of the page. You can also click the either of the home buttons which also log you out of the application and display the homepage to the screen.

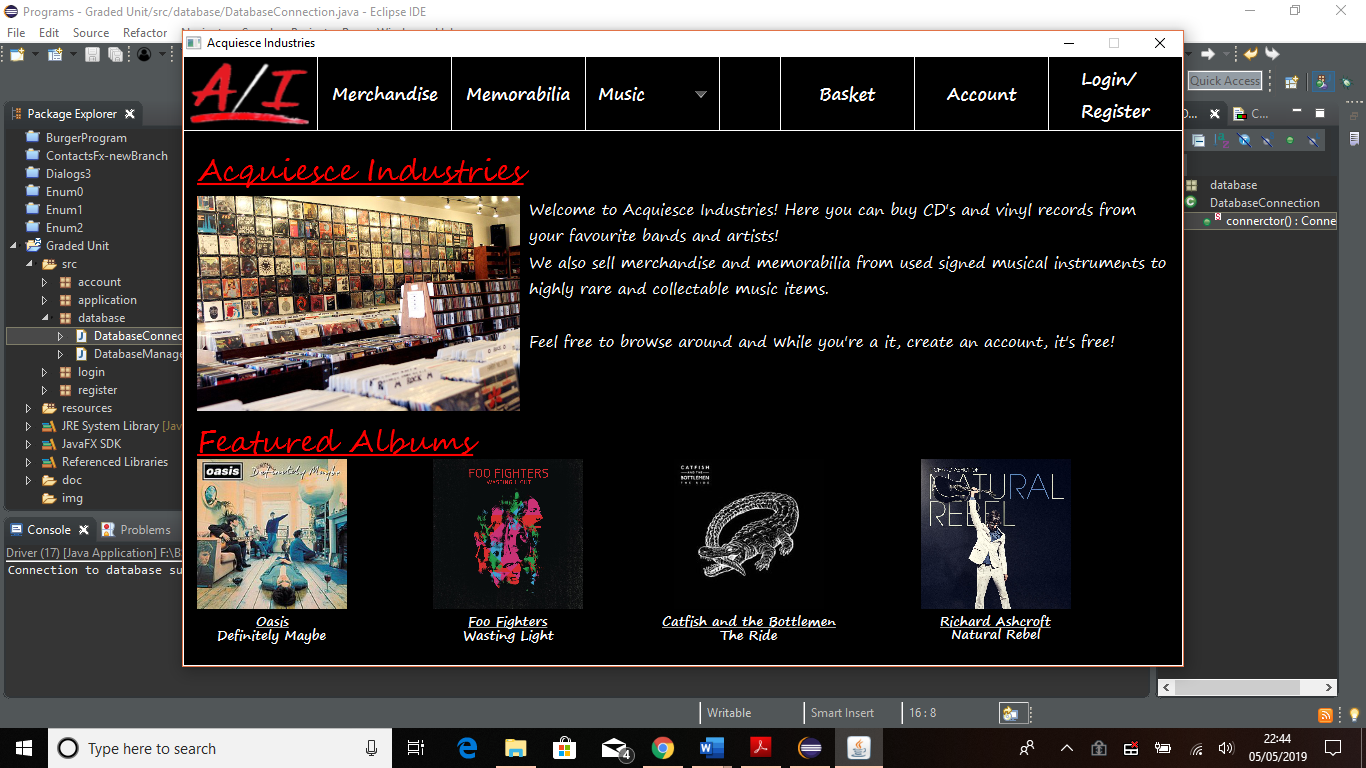
Home button that will log you out.

Second home   
button that will also log you out.

The main logout button that will log you out of the application and take you back to the homepage.

### 3.1.6. Closing the Application

Step 1.  
To close the application you click the X in the top right-hand corner of the window that application is held in. This can be done anywhere in the application and will also log you out if signed in at the time.

 On every page, the X button closes and shuts down the application.

# 4. Information Sources

Corporation, O., 2019. *java.com - Java Download.* [Online]   
Available at: https://www.java.com/en/download/  
[Accessed 01 05 2019].

Overflow, S., 2019. *stackoverflow.com - JAR file: Could not find main class.* [Online]   
Available at: https://stackoverflow.com/questions/10149463/jar-file-could-not-find-main-class  
[Accessed 01 05 2019].

ProgrammingKnowledge, 2015. *YouTube.com - JavaFX SQLite Database Tutorial.* [Online]   
Available at: https://www.youtube.com/watch?v=NWcFTTbKbLs&list=PLS1QulWo1RIbqOzdu1jShN0xah41S6lLP  
[Accessed 15 04 2019].