

CERTIFICATE OF INTERNSHIP



Certificate of Internship

**This is to certify that
Mr. Wieslaw Samushonga has successfully
completed the online
Java MasterClass Internship program
from 20-07-2020 to 12-10-2020 at
AllSoft Solutions and developed the project
" Ace Academy Equipment Logging System" .**

A handwritten signature in black ink, appearing to read "Ashwani Kumar".

**Ashwani Kumar
Supervisor**



12-10-2020

**Issued On
ID: ASS289ba1**

ABOUT THE COMPANY

Allsoft Solutions & Services Pvt Ltd

Allsoft Solutions and Services Pvt Ltd, IBM Business Partner was started in 2000 having its headquarters in the USA, with the establishment of three branches in India. Allsoft has been committed to providing the highest quality and needs-based services to its clients, both locally and internationally. They offer IT consulting, Industrial Training Program in various technologies, Web Development and Cloud services as well. They believe in “Making of the era with professionals” in this highly competitive industry. Their policy towards introducing new, on demand and emerging technology makes them competitive in the market as their goals are to provide the best possible value and to lead in the markets through service and innovation. Being one of IBM’s trusted and certified business partner, they offer training in IBM technologies to provide students an exposure to the IBM tools also.

Admittance

As per the emergence of the digital era, the needs of the corporate world are changing. So, they made an oath to provide the technologies that are insanely changing our world.

IBM certified trainers

To provide the best training experience we are having our finest IBM certified trainers who will fill up brains with the knowledge of today’s growing technologies and endow for the industrial world requirements.

Smart coursework

Our coursework is carefully planned to provide the real essence of knowledge to our trainees. Courses they train include **Java Programming (Advanced classes included)**, Python, Big Data, Web Development, Cloud Computing, Apple Development, Software Testing, BlockChain, AI, Ethical Hacking, PHP, Network Training, IOT, Android App Development and Machine Learning.

ABSTRACT

Due to ongoing concerns on how equipment and stocks at sports clubs are managed, it’s been suggested that a system that oversees the flow of all items has to be put in place. This is why the Ace Tennis Academy **Head Coach approached AllSoft Solutions** for a solution to this issue. The problem was passed down to me and after a few months of development, a desktop application has been tailor-made to their needs. During this report I shall thoroughly go through all the concepts and dependencies that were used to develop the “**Ace Academy Equipment Logging System**”.

The project titled “Ace Academy Equipment Logging System” is a **sports stock management software** for monitoring and controlling the handling of equipment at an academy .The project is developed in Java and mainly focuses on basic operations at an academy like **adding new members, new equipment, and updating information, searching equipment and members and facility to borrow and return equipment**. "Ace Academy Equipment Logging System" is a **desktop application** written for both 32 and 64-bit Windows operating systems, designed to help specific personnel maintain and organize the academy's equipment. **This software is easy to use for both beginners and advanced users.**

The system as of now will only contain the **administrator's access and priviledged members** who have paid a nominal fee. This is so that there is improved security of the academy's equipment and giving members' access as well will open doors to **heavy breaches on the system** and false issuing and returns. **Java** was ultimately decided as the **primary back-end language** with **JavaFX used for the front-end** as it gives **beautiful, modern and styled** visuals. It was an added advantage for me as well as I had to go through numerous learning curves and it improved my coding skills in a period of less than 4 months.

TABLE OF CONTENTS

CERTIFICATE OF INTERNSHIP	i
DECLARATION.....	Error! Bookmark not defined.
ACKNOWLEDGEMENT.....	Error! Bookmark not defined.
ABOUT THE COMPANY.....	ii
ABSTRACT	ii
LIST OF FIGURES	3
CHAPTER 1: INTRODUCTION.....	4
1.1. Introduction	4
1.2 System analysis existing system:.....	6
1.3 Proposed system:.....	7
CHAPTER 2:TOOLS AND TECHNOLOGY USED	8
2.1 Requirements.....	8
2.2 JavaFX Applications	8
Key Features	9
CHAPTER 3.....	11
3.1 Images on Views	11
3.2 Database	11
3.21 DB.java.....	12
3.22 DBConnection.java	14
3.3 Execute.java.....	17
3.4 Login Screen.....	18
3.41LoginScreenView1.fxml	19
3.42 LoginScreenController.java.....	23
3.5 Home View.....	25
3.51 AceHomeView.fxml.....	26
3.52 AceHomeController.java.....	29
3.6 Equipment Issuing to Members.....	32

3.61 EquipmentBorrowedView.fxml	33
3.62 EquipmentBorrowedController.java	36
3.63 EquipmentBorrowedModel.java	45
3.7 Equipment Returned back by Members to the academy	46
3.71 EquipmentReturnedView.fxml.....	47
3.72 EquipmentReturnedController.java.....	49
3.73 EquipmentReturnedModel.java.....	56
3.8 Equipment Searching to identify Stock Available	58
3.81 EquipmentSearchView.fxml	58
3.82 EquipmentSearchController.java.....	60
3.83 EquipmentModel.java	64
3.9 Equipment View to see the detailed list and status of Equipment	66
3.91 EquipmentView.fxml	66
3.92 EquipmentController.java	69
3.10 MembershipView which shows the List of Registered Members	77
3.10.1 MembershipView.fxml.....	78
3.10.2 MembershipController.java.....	80
3.10.3 MembershipModel.java.....	89
CHAPTER 4: RESULTS AND DISCUSSIONS	91
4.1 The Flow of Control and Information	91
Figure 4.1 DFD for Model-View-Controller Pattern used in the project.	91
4.2 Level of Accomplishment.	92
CHAPTER 5.....	94
5.1 Conclusions	94
5.2 Future Scope.....	94
REFERENCES	95

LIST OF FIGURES

Figure 1.1 Screenshot of All Packages and Files Used.....	5
Figure 2.2 Illustration of modelling of data for MVC concepts.....	5
Figure 3.01: ace.png.....	12
Figure 3.02: back.png.....	12
Figure 3.03: equipmentreturned.png.....	12
Figure 3.04: Equipmentsearch.png	12
Figure 3.05 equipment.png.....	12
Figure 3.06: membership.png.....	12
Figure 3.07: equipmentborrowed.png.....	12
Figure 3.1 Screenshot of Xampp Control Panel for phpMyAdmin.....	13
Figure 3.2 Screenshot of EquipmentBorrowed Table on phpMyAdmin.....	13
Figure 3.2 Screenshot of Equipmentdetail Table on phpMyAdmin.....	15
Figure 3.4 Screenshot of EquipmentReturn Table on phpMyAdmin.....	15
Figure 3.5 Screenshot of MembershipDetail Table on phpMyAdmin.....	15
Figure 3.5 Screenshot of Login UserData Table on phpMyAdmin.....	15
Figure 3.6 a) Screenshot of Log In Screen before credentials are authenticated.....	16
b) After Log In credentials are approved.....	16
c) Screenshot of Login Screen after Exit button is clicked.....	17
Figure 3.7 a) Screenshot of Home View just After Logging In.....	22
b) When Logout Button is pressed.....	22
Figure 3.8 Screenshot of Equipment Borrowed View.....	29
Figure 3.9 Screenshot of Equipment Returned View.....	43
Figure 3.10 Screenshot of Equipment Search View.....	54
Figure 3.11 Screenshot of Equipment Stock View.....	62
Figure 3.12 Screenshot of Membership View.....	73
Figure 4.1 DFD for Model-View-Controller pattern used in the project.....	86
Figure 4.2 a) Screenshot: Clicking + New for user input.....	86
b) Clicking Add puts the Member on record into the Database.....	86

CHAPTER 1: INTRODUCTION

1.1. Introduction

The purpose of the application is automation of the sports academy ,it provide facilities to administrators and privileged personnel to search for the required equipment and it allows the administrator to Issue to & collect equipment from members and can create & delete memberships.

Packages:

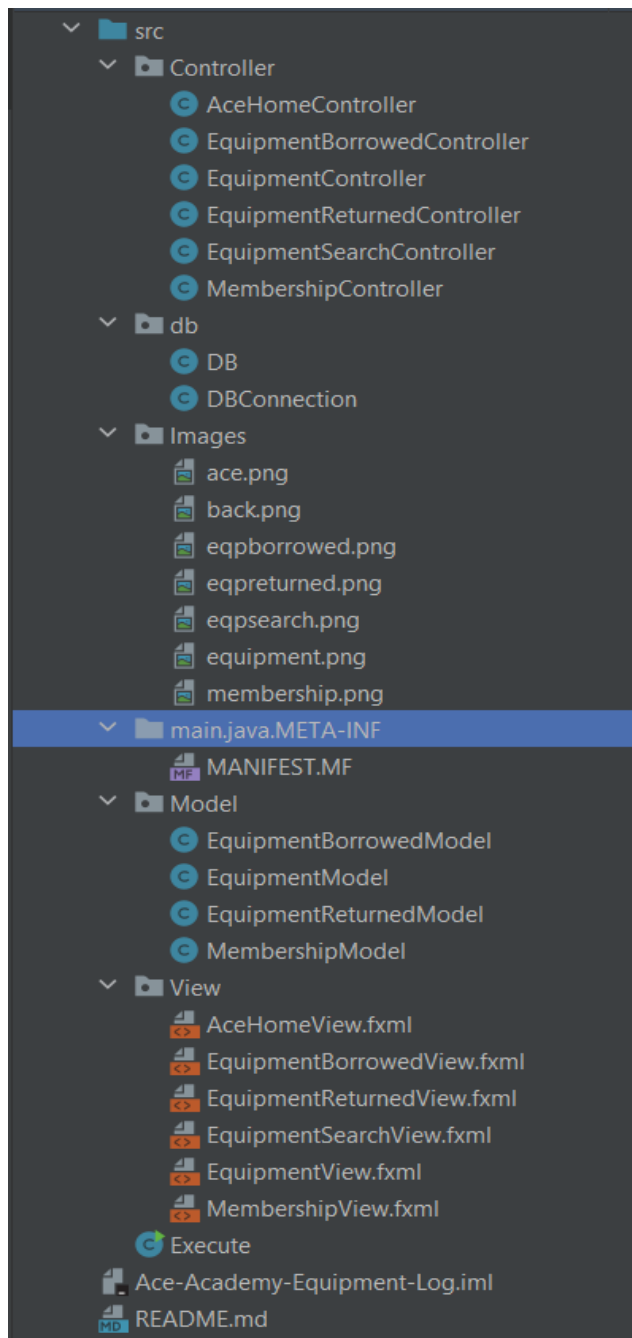


Figure 1.1 Screenshot of All Packages and Files Used

As shown on the pictures to your above, there are 6 packages apart from the ‘**Execute**’ Main Class for the program.

Controller

The controller is the like the housekeeper of the application – it performs coordination between model and view to entertain a user request. The user requests are received - for example, when the user clicks on any GUI elements to perform any action.

The primary function of a controller is to call and coordinate with the model to fetch any necessary resources required to act. Usually, on receiving a user request, the controller calls the appropriate model for the task at hand.

Model

The model is quite simply the data for our application. The data is “*modelled*” in a way it’s easy to store, retrieve, and edit. The model is how we apply rules to our data, which eventually represents the concepts our application manages.

For any software application, everything is modelled as data that can be handled easily. What is a user, a book, or a message for an app? Nothing really, only data that must be processed according to specific rules. Like, the date must not be higher than the current date, the email must be in the correct format, the name mustn’t be more than “x” characters long, etc.

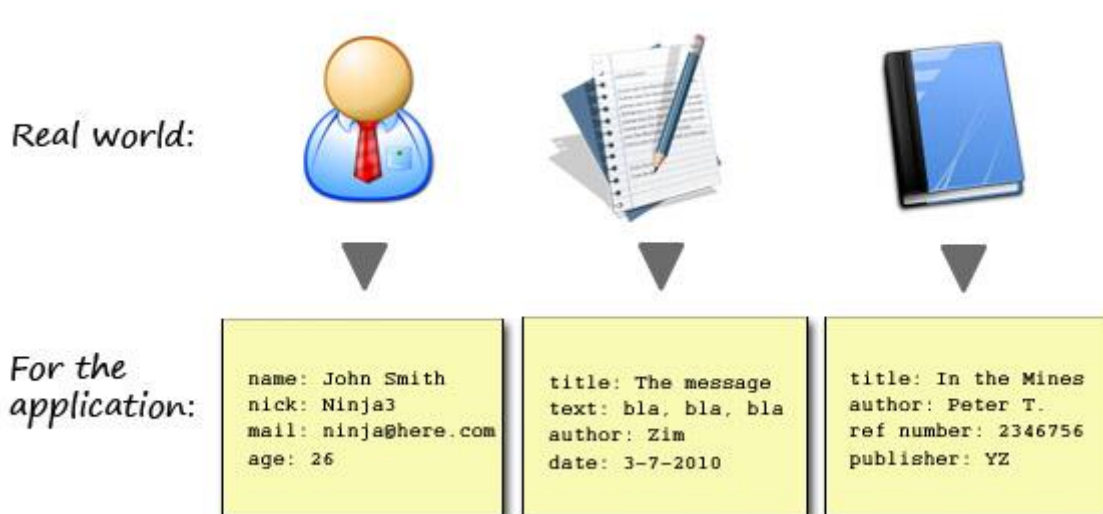


Figure 2.2 Illustration of modelling of data for MVC concepts.

Whenever a user makes any request from the controller, it contacts the appropriate model which returns a data representation of whatever the user requested. This model will be the same for a particular work, irrespective of how we wish to display it to the user. That is why we can choose any available view to render

the model data.

Additionally, a model also contains the logic to update the relevant controller whenever there is any change in the model's data.

View

As the name suggests, the view is responsible for rendering the data received from the model. There may be pre-designed templates where you can fit the data, and there may even be several different views per model depending on the requirements.

Images

All the images used in graphics enhanced will be stored in their own package. This ensures comprehensibility for the compilation of the whole application. Images are necessary components of the JavaFX graphics.

DB

The DB package focuses mainly on the database end of application. Being linked to a model via the ArrayList makes the control and flow of records so much easier. The DB packages uses objects to access to access the information on the database server i.e MongoDB via the Client i.e phpMyAdmin.

1.2 System analysis existing system:

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what curb all the problems in the current system. Analysis began when the Head Coach user noticed that the manual and in person management or lack thereof of a system in place is the problem. In the existing system all the management of equipment stocks is done manually, So taking more time for a process like borrowing balls or returning a racket and also for

searching of members and equipment, which by the way depended largely on the memory of the Head Coach. Another major disadvantage is that to preparing the list of equipment borrowed and the available equipment at the academy will take more time, currently it is done as a one day process for verifying all records. So after conducting the feasibility study we decided to make the manual Equipment Logging to be computerized.

1.3 Proposed system:

Proposed system is an automated Equipment Logging System. Through the software, the administrator can add members, add equipment, search members, search equipment, update information, edit information, borrow and return equipment in real time.

Our proposed system has the following advantages.

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility
- Reliable and Safe Environment to Issue and Collect equipment

All the manual difficulties in managing the equipment have been rectified by implementing computerization.

CHAPTER 2: TOOLS AND TECHNOLOGY USED

2.1 Requirements

Software Requirements:

Operating System: Windows 7, 8, 10 (All editions)

Languages: Java 1.6 or above

Tools: NetBeans /Eclipse/IntelliJ Idea

Scene Builder within JavaFX

Java JDK-14 and JRE1.8

Database: Server-**MariaDB via XAMPP**

Client –**phpMyAdmin**

Web Server: **Apache 2.4.46**

Hardware Requirements:

Processor: 800 MHz or above

RAM (SD/DDR): 1GB

HDD/ SSD: 30GB

Dependences:

JavaFX API v10.0.1 and Runtime V8.0.202-ea

2.2 JavaFX Applications

Written as a Java API, JavaFX application code can reference APIs from any Java library. For example, JavaFX applications can use Java API libraries to access native system capabilities and connect to server-based middleware applications.

The look and feel of JavaFX applications can be customized. Cascading Style Sheets (CSS) separate appearance and style from implementation so that developers can concentrate on coding. Graphic designers can easily customize the appearance and style of the application through the CSS. If you have a web design background, or if you would like to separate the user interface (UI) and the back-end logic, then you can develop the presentation aspects of the UI in the FXML scripting language and use Java code for the application logic. If you prefer to design UIs without writing code, then use JavaFX Scene Builder. As you

design the UI, Scene Builder creates FXML markup that can be ported to an Integrated Development Environment (IDE) so that developers can add the business logic.

Key Features

JavaFX 2.2 and later releases have the following features:

- **Java APIs.** JavaFX is a Java library that consists of classes and interfaces that are written in native Java code. The APIs are designed to be a friendly alternative to Java Virtual Machine (Java VM) languages, such as JRuby and Scala.
- **FXML and Scene Builder.** FXML is an XML-based declarative markup language for constructing a JavaFX application user interface. A designer can code in FXML or use JavaFX Scene Builder to interactively design the graphical user interface (GUI). Scene Builder generates FXML markup that can be ported to an IDE where a developer can add the business logic.
- **WebView.** A web component that uses WebKitHTML technology to make it possible to embed web pages within a JavaFX application. JavaScript running in WebView can call Java APIs, and Java APIs can call JavaScript running in WebView.
- **Swing interoperability.** Existing Swing applications can be updated with new JavaFX features, such as rich graphics media playback and embedded Web content.
- **Built-in UI controls and CSS.** JavaFX provides all the major UI controls required to develop a full-featured application. Components can be skinned with standard Web technologies such as CSS
- **Canvas API.** The Canvas API enables drawing directly within an area of the JavaFX scene that consists of one graphical element (node).
- **Multitouch Support.** JavaFX provides support for multitouch operations, based on the capabilities of the underlying platform.
- **Hardware-accelerated graphics pipeline.** JavaFX graphics are based on the graphics rendering pipeline (Prism). JavaFX offers smooth graphics that render quickly through Prism when it is used with a supported graphics card or graphics processing unit (GPU). If a system does not feature one of the recommended GPUs supported by JavaFX, then Prism defaults to the Java 2D software stack.
- **High-performance media engine.** The media pipeline supports the playback of web multimedia content. It provides a stable, low-latency media framework that is based on the GStreamer multimedia framework.
- **Self-contained application deployment model.** Self-contained application packages have all of the application resources and a private copy of the Java and JavaFX runtimes. They are distributed as

native installable packages and provide the same installation and launch experience as native applications for that operating system.

***The whole application in a summary is an infusion of Array Lists, JavaFX MVC (model-view-controller and FXML View Graphics) and Database (JDBC) into Java Core Concepts.**

CHAPTER 3

3.1 Images on Views



Figure 3.01: ace.png



Figure 3.02: back.png

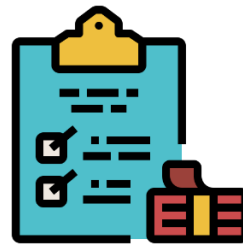


Figure 3.03: equipmentreturned.png

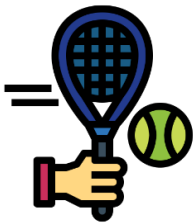


Figure 3.04: Equipmentsearch.png



Figure 3.05 equipment.png

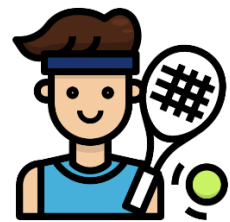


Figure 3.06: membership.png



Figure 3.07: equipmentborrowed.png

3.2 Database

The database of the whole project is managed using MongoDB which has a server of phpMyAdmin via XAMPP Control Panel. There are 5 tables in total and these are imaged below.

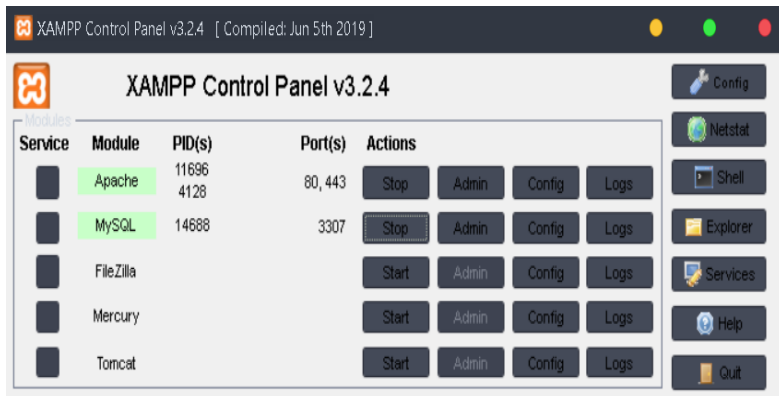


Figure 3.1 Screenshot of Xampp Control Panel for phpMyAdmin.

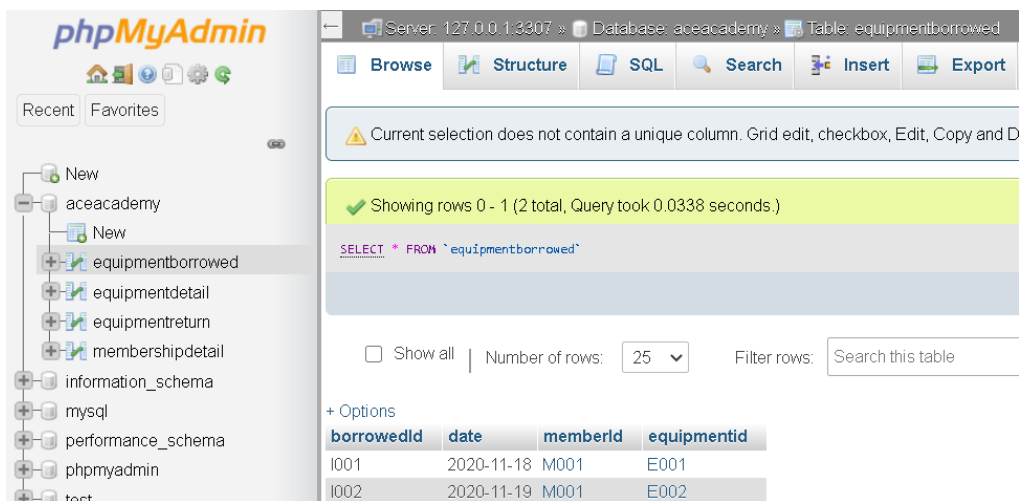


Figure 3.2 Screenshot of EquipmentBorrowed Table on phpMyAdmin

3.21 DB.java

```
package db;
```

```
import Model.EquipmentBorrowedModel;
```

```
import Model.EquipmentReturnedModel;
```

```
import Model.EquipmentModel;
```

```
import Model.MembershipModel;
```

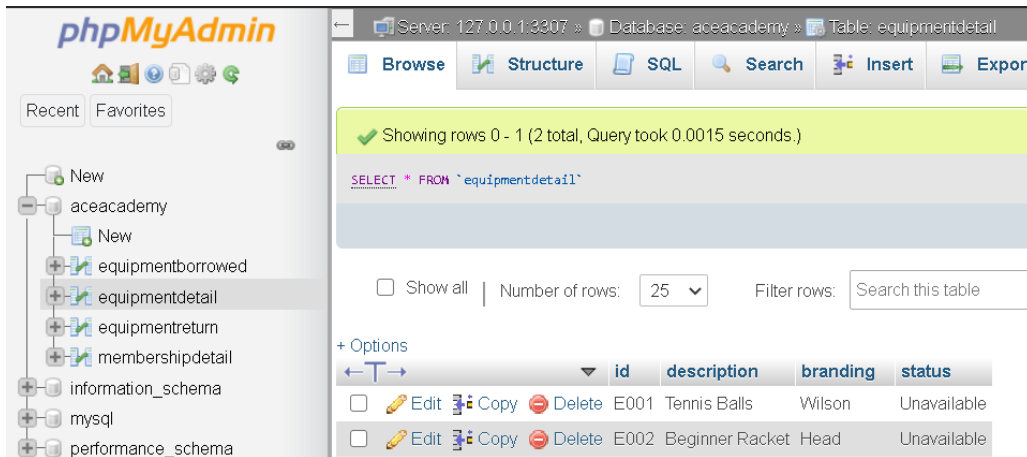
```
import java.util.ArrayList;
```

```
public class DB {
```

```

public static ArrayList<MembershipModel> members = new ArrayList<>();
public static ArrayList<EquipmentModel> equipment = new ArrayList<>();
public static ArrayList<EquipmentBorrowedModel> borrowed = new ArrayList<>();
public static ArrayList<EquipmentReturnedModel> returned = new ArrayList<>();
}

```



Server: 127.0.0.1:3307 » Database: aceacademy » Table: equipmentdetail

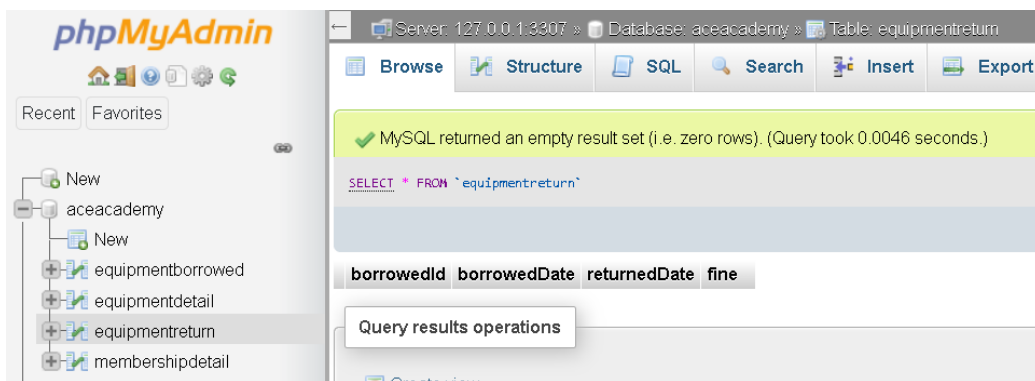
Showing rows 0 - 1 (2 total, Query took 0.0015 seconds.)

SELECT * FROM `equipmentdetail`

Number of rows: 25 Filter rows: Search this table

	id	description	branding	status
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E001	Tennis Balls	Wilson	Unavailable
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E002	Beginner Racket	Head	Unavailable

Figure 3.2 Screenshot of Equipmentdetail Table on phpMyAdmin



Server: 127.0.0.1:3307 » Database: aceacademy » Table: equipmentreturn

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0046 seconds.)

SELECT * FROM `equipmentreturn`

borrowedId	borrowedDate	returnedDate	fine
------------	--------------	--------------	------

Query results operations

Figure 3.4 Screenshot of EquipmentReturn Table on phpMyAdmin

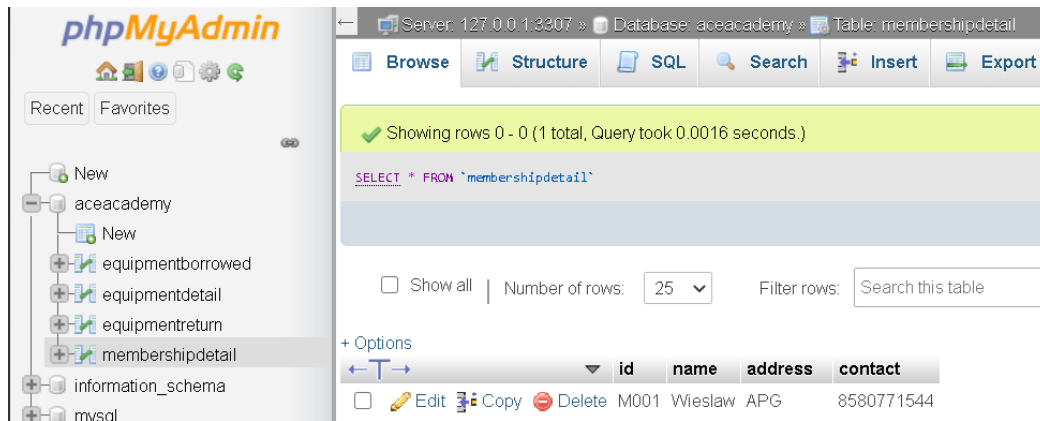


Figure 3.5 Screenshot of MembershipDetail Table on phpMyAdmin

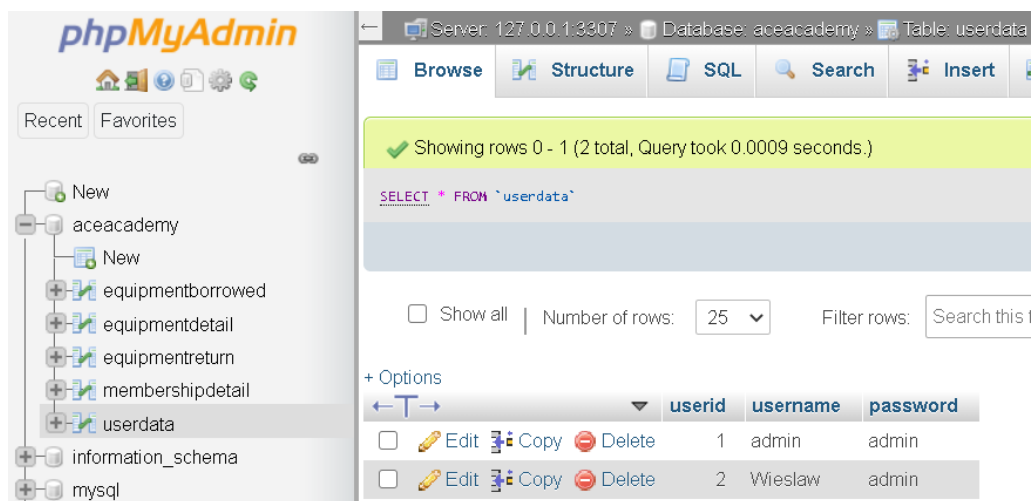


Figure 3.5 Screenshot of Login UserData Table on phpMyAdmin

3.22 DBConnection.java

```
package db;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

```
public class DBConnection {
```

```
    private static DBConnection dbConnection;//
```

```
    private Connection connection;// creating connection object
```

```

private DBConnection() {
    try {
        Class.forName("com.mysql.jdbc.Driver");// method used to call class containing external
configuration file to connect to database. Afterwards
        connection =
DriverManager.getConnection("jdbc:mysql://localhost:3307/aceacademy?createDatabaseIfNotExist=true&allowMultiQueries=true", "root","");
        //linking to database online via xampp server.
        PreparedStatement pstmt = connection.prepareStatement("SHOW TABLES");// Interface used to run
many Sql queries.Dynamically.Right now it will show all non-temporary tables
        ResultSet resultSet = pstmt.executeQuery();//The ResultSet interface provides getter methods for
retrieving column values from the current row.
        if (!resultSet.next()) {
            String sql = "\n" +
                "CREATE TABLE `equipmentdetail` (\n" +
                " `id` varchar(10) NOT NULL,\n" +
                " `description` varchar(15) DEFAULT NULL,\n" +
                " `branding` varchar(20) DEFAULT NULL,\n" +
                " `status` varchar(20) DEFAULT NULL,\n" +
                " PRIMARY KEY (`id`)\n" +
                ") ENGINE=InnoDB DEFAULT CHARSET=latin1;\n" +
                "\n" +
                "CREATE TABLE `equipmentborrowed` (\n" +
                " `borrowedId` varchar(10) NOT NULL,\n" +
                " `date` date DEFAULT NULL,\n" +
                " `memberId` varchar(10) DEFAULT NULL,\n" +
                " `equipmentid` varchar(10) DEFAULT NULL,\n" +
                " PRIMARY KEY (`borrowedId`)\n" +
                " CONSTRAINT FOREIGN KEY (`memberId`) REFERENCES `membershipdetail`
(id`),\n" +
                " CONSTRAINT FOREIGN KEY (`equipmentid`) REFERENCES `equipmentdetail`
(id`)\n" +
                ") ENGINE=InnoDB DEFAULT CHARSET=latin1;\n" //InnoDB is a database storage
engine. database storage engine is by which tables are stored, retrieved and handled.

```

```

"\n" +
"CREATE TABLE `membershipdetail` (\n" +
" `id` varchar(11) NOT NULL,\n" +
" `name` varchar(50) DEFAULT NULL,\n" +
" `address` varchar(50) DEFAULT NULL,\n" +
" `contact` varchar(12) DEFAULT NULL,\n" +
" PRIMARY KEY (`id`),\n" +
") ENGINE=InnoDB DEFAULT CHARSET=latin1;\n" +
"\n" +
"CREATE TABLE `equipmentreturn` (\n" +
" `id` int(11) NOT NULL,\n" +
" `borrowedDate` date NOT NULL,\n" +
" `returnedDate` date DEFAULT NULL,\n" +
" `fine` int(10) DEFAULT NULL,\n" +
" `borrowedId` int(10) DEFAULT NULL,\n" +
" PRIMARY KEY (`id`),\n" +
" CONSTRAINT FOREIGN KEY (`borrowedId`) REFERENCES `equipmentborrowed`
(`borrowedId`),\n" +
") ENGINE=InnoDB DEFAULT CHARSET=latin1;\n";

pstmt = connection.prepareStatement(sql);
pstmt.execute();
}
} catch (Exception e) {
    throw new RuntimeException(e); // Callers of that method aren't required to catch the exception, or
acknowledge it in any way. It's an unchecked exception
}

}

public static DBConnection getInstance() {
    return dbConnection = ((dbConnection == null) ? new DBConnection() : dbConnection);
}

public Connection getConnection() {

```

```
        return connection;
    }
}
```

3.3 Execute.java

//The Execute.java file contains the first code that will call onto all the other files.

```
import db.DBConnection;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;
import java.sql.SQLException;

public class Execute extends Application {

    public static void main(String[] args) {
        launch(args);
        try {
            DBConnection.getInstance().getConnection().close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void start(Stage primaryStage) throws IOException {
        primaryStage.setScene(new
Scene(FXMLLoader.load(this.getClass().getResource("/View/LoginScreenView1.fxml"))));
```

```
    primaryStage.show();  
  }  
}
```

3.4 Login Screen

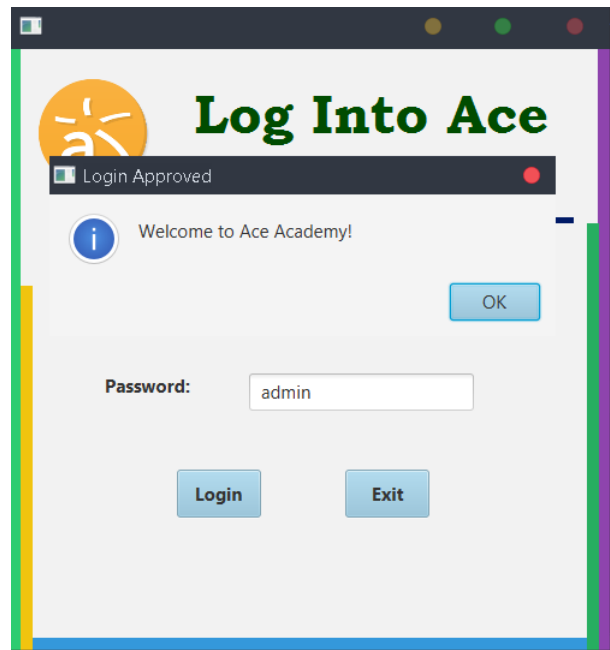
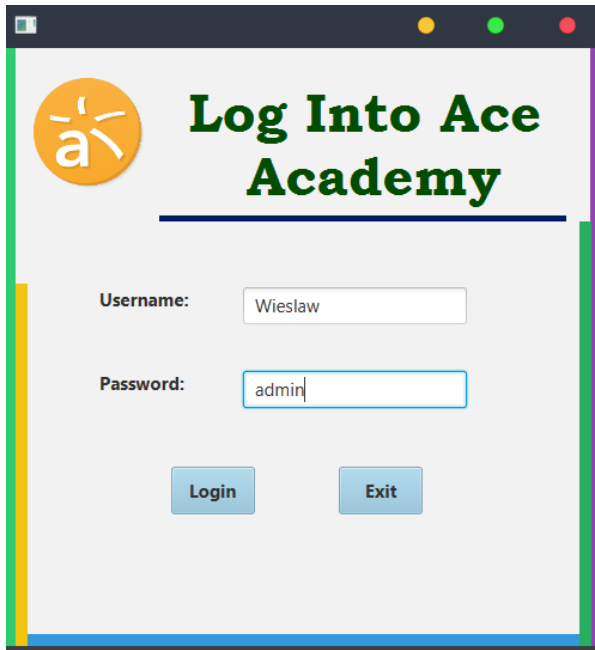


Figure 3.6a Screenshot of Log In Screen b)After Log In credentials are approved Before credentials are authenticated.

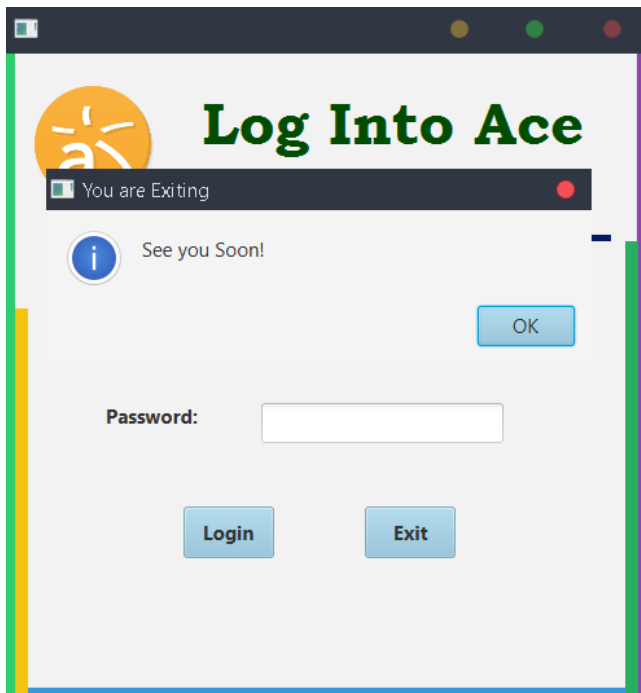


Figure 3.6c) Screenshot of Login Screen after Exit button is clicked.

3.41LoginScreenView1.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.image.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.control.Button?>
<AnchorPane xmlns:fx="http://javafx.com/fxml/1" fx:id="root" maxHeight="-Infinity" maxWidth="-Infinity"
    minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="600.0"
    style="-fx-background-color: #ecf0f1; " xmlns="http://javafx.com/javafx/10.0.1"
    fx:controller="Controller.AceHomeController" >
    <children>
        <ImageView fx:id="membership" fitHeight="100.0" fitWidth="100.0" layoutX="333.0"
layoutY="191.0"
            onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
            onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@../Images/membership.png"/>
            </image>
        </ImageView>
        <ImageView fx:id="equipment" fitHeight="100.0" fitWidth="100.0" layoutX="61.0" layoutY="379.0"
            onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
            onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@../Images/equipment.png"/>
            </image>
        </ImageView>
        <ImageView fx:id="eqp_borrowed" fitHeight="116.0" fitWidth="127.0" layoutX="233.0"
layoutY="368.0"
            onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
            onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@../Images/eqpborrowed.png"/>
            </image>
        </ImageView>
    </children>
</AnchorPane>

```

```

<ImageView fx:id="ace_tennis" fitHeight="116.0" fitWidth="127.0" layoutX="20.0" layoutY="20.0"
    onMouseClicked="#navigate" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/ace.png"/>
    </image>
</ImageView>
<ImageView fx:id="eqp_returned" fitHeight="100.0" fitWidth="100.0" layoutX="415.0"
layoutY="379.0"
    onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
    onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/eqpreturned.png"/>
    </image>
</ImageView>
<ImageView fx:id="eqp_search" fitHeight="100.0" fitWidth="100.0" layoutX="150.0"
layoutY="191.0"
    onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
    onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/eqpsearch.png"/>
    </image>
</ImageView>
<Text fill="#141452" fontSmoothingType="LCD" layoutX="55.0" layoutY="73.0"
strokeType="OUTSIDE"
    strokeWidth="0.0" text="Ace Academy Equipment Log" textAlignment="CENTER"
wrappingWidth="521.140625">
    <font>
        <Font name="Bookman Old Style Bold" size="45.0"/>
    </font>
</Text>
<Text fill="#1a1aff" layoutX="335.0" layoutY="321.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Membership">
    <font>
        <Font name="Britannic Bold" size="16.0"/>
    </font>

```



```

</Text>
<Text fill="#1a1aff" layoutX="80.0" layoutY="518.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment"
    textAlignment="CENTER">
    <font>
        <Font name="Britannic Bold" size="16.0"/>
    </font>
</Text>
<Text fill="#1a1aff" layoutX="210.0" layoutY="518.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Borrowed">
    <font>
        <Font name="Britannic Bold" size="16.0"/>
    </font>
</Text>
<Text fill="#1a1aff" layoutX="395.0" layoutY="518.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Returned">
    <font>
        <Font name="Britannic Bold" size="16.0"/>
    </font>
</Text>
<Text fill="#1a1aff" layoutX="140.0" layoutY="321.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Search">
    <font>
        <Font name="Britannic Bold" size="16.0"/>
    </font>
</Text>
<Button fx:id="logoutButton" layoutX=" 240" layoutY="540" alignment="CENTER"
defaultButton="true" mnemonicParsing="false" text="Logout"
    style="-fx-font-weight:bold;" prefHeight="40.0" prefWidth="80"
    onAction="#logout"/>
<Pane prefHeight="600.0" prefWidth="10.0" style="-fx-background-color: #2ecc71;"/>
<Pane layoutX="10.0" layoutY="197.0" prefHeight="403.0" prefWidth="10.0"
    style="-fx-background-color: #f1c40f;"/>
<Pane layoutX="590.0" prefHeight="600.0" prefWidth="10.0" style="-fx-background-color:
#8e44ad;"/>

```

```

<Pane layoutX="20.0" layoutY="590.0" prefHeight="10.0" prefWidth="561.0" style="-fx-background-
color: #3498db;"/>
<Pane layoutX="581.0" layoutY="145.0" prefHeight="455.0" prefWidth="10.0"
style="-fx-background-color: #27ae60;"/>
<Pane layoutX="135.0" layoutY="140.0" prefHeight="5.0" prefWidth="360.0"
style="-fx-background-color: #b30000;"/>
</children>
</AnchorPane>

```

3.42 LoginScreenController.java

```

package Controller;

import javafx.animation.TranslateTransition;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.stage.Stage;
import javafx.util.Duration;

import java.io.IOException;
import java.sql.*;
import java.util.Optional;

public class LoginScreenController{
    @FXML
    private TextField user;
    @FXML private TextField password;
    @FXML private Button loginButton;
    @FXML
    public Button exitButton;

```

@FXML

```
public void exitAction(ActionEvent closeEvent) throws SQLException, ClassNotFoundException {
    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle("You are Exiting");
    alert.setHeaderText(null);
    alert.setContentText("See you Soon!");
    Optional<ButtonType> result = alert.showAndWait();
    Stage closeStage = (Stage) exitButton.getScene().getWindow();
    closeStage.close();
}

public void loginAction(javafx.event.ActionEvent actionEvent) throws SQLException,
ClassNotFoundException {
    Connection con=null;
    PreparedStatement ps=null;
    ResultSet rs=null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        con= DriverManager.getConnection("jdbc:mysql://localhost:3307/aceacademy", "root", "");
        ps= con.prepareStatement("SELECT * FROM userdata WHERE username=? and password=?");
        ps.setString(1,user.getText());
        ps.setString(2,password.getText());
        rs=ps.executeQuery();
        if (rs.next()){
            Alert alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("Login Approved");
            alert.setHeaderText(null);
            alert.setContentText("Welcome to Ace Academy!");
            alert.showAndWait();
        }
        Parent root;
        try
        {
            root = FXMLLoader.load(getClass().getResource("/View/AceHomeView.fxml"));
            Stage stage = new Stage();
            Scene scene = new Scene(root);
```

```

        stage.setScene(scene);
        stage.show();
        TranslateTransition tt = new TranslateTransition(Duration.millis(350), scene.getRoot());
        tt.setFromX(-scene.getWidth());
        tt.setToX(0);
        tt.play();
        ((Node)(actionEvent.getSource())).getScene().getWindow().hide();
    } catch (IOException ioe)
    {
        ioe.printStackTrace(System.out);
    }
}
else{
    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle("Login Failed");
    alert.setHeaderText(null);
    alert.setContentText("Please Try Again!");
    alert.showAndWait();
}
} catch (ClassNotFoundException | SQLException e) {
    e.printStackTrace();
}

}

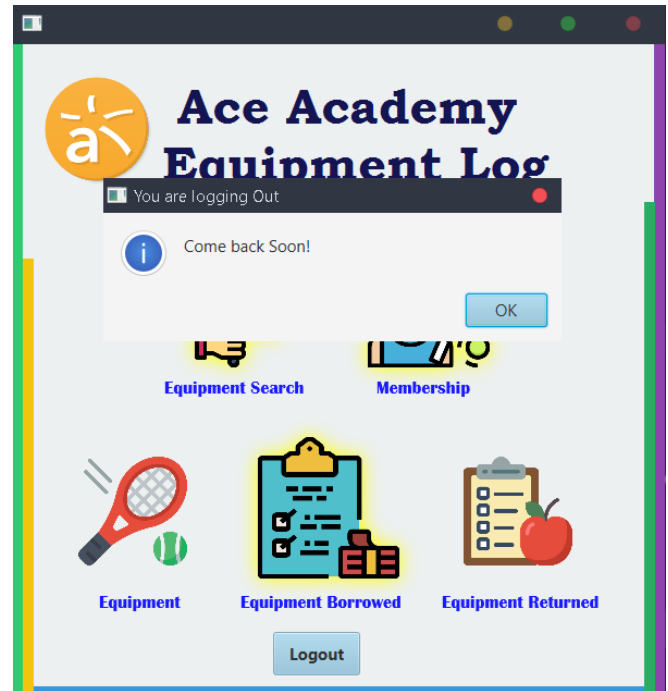
}

```

3.5 Home View



**Figure 3.7 a) Screenshot of Home View
Just After Logging In**



b) When Logout Button is Pressed

3.51 AceHomeView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.image.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>
<AnchorPane xmlns:fx="http://javafx.com/fxml/1" fx:id="root" maxHeight="-Infinity" maxWidth="-Infinity"
    minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="600.0"
    style="-fx-background-color: #ecf0f1; " xmlns="http://javafx.com/javafx/10.0.1"
    fx:controller="Controller.AceHomeController" >
    <children>
        <ImageView fx:id="membership" fitHeight="100.0" fitWidth="100.0" layoutX="333.0"
layoutY="191.0"
            onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
            onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@../Images/membership.png"/>
            </image>
        </ImageView>
    </children>
</AnchorPane>
```

```

    </image>
</ImageView>
<ImageView fx:id="equipment" fitHeight="100.0" fitWidth="100.0" layoutX="61.0" layoutY="379.0"
    onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
    onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/equipment.png"/>
    </image>
</ImageView>
<ImageView fx:id="eqp_borrowed" fitHeight="116.0" fitWidth="127.0" layoutX="233.0"
layoutY="368.0"
    onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
    onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/eqpborrowed.png"/>
    </image>
</ImageView>
<ImageView fx:id="ace_tennis" fitHeight="116.0" fitWidth="127.0" layoutX="20.0" layoutY="20.0"
    onMouseClicked="#navigate" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/ace.png"/>
    </image>
</ImageView>
<ImageView fx:id="eqp_returned" fitHeight="100.0" fitWidth="100.0" layoutX="415.0"
layoutY="379.0"
    onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
    onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/eqpreturned.png"/>
    </image>
</ImageView>
<ImageView fx:id="eqp_search" fitHeight="100.0" fitWidth="100.0" layoutX="150.0"
layoutY="191.0"
    onMouseClicked="#navigate" onMouseEntered="#playMouseEnterAnimation"
    onMouseExited="#playMouseExitAnimation" pickOnBounds="true" preserveRatio="true">

```

```

    <image>
      <Image url="@../Images/eqpsearch.png"/>
    </image>
  </ImageView>
  <Text fill="#141452" fontSmoothingType="LCD" layoutX="55.0" layoutY="73.0"
strokeType="OUTSIDE"
      strokeWidth="0.0" text="Ace Academy Equipment Log" textAlignment="CENTER"
wrappingWidth="521.140625">
    <font>
      <Font name="Bookman Old Style Bold" size="45.0"/>
    </font>
  </Text>
  <Text fill="#1a1aff" layoutX="335.0" layoutY="321.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Membership">
    <font>
      <Font name="Britannic Bold" size="16.0"/>
    </font>
  </Text>
  <Text fill="#1a1aff" layoutX="80.0" layoutY="518.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment"
      textAlignment="CENTER">
    <font>
      <Font name="Britannic Bold" size="16.0"/>
    </font>
  </Text>
  <Text fill="#1a1aff" layoutX="210.0" layoutY="518.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Borrowed">
    <font>
      <Font name="Britannic Bold" size="16.0"/>
    </font>
  </Text>
  <Text fill="#1a1aff" layoutX="395.0" layoutY="518.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Returned">
    <font>
      <Font name="Britannic Bold" size="16.0"/>

```

```

        </font>
    </Text>
    <Text fill="#1a1aff" layoutX="140.0" layoutY="321.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Search">
        <font>
            <Font name="Britannic Bold" size="16.0"/>
        </font>
    </Text>
    <Pane prefHeight="600.0" prefWidth="10.0" style="-fx-background-color: #2ecc71;"/>
    <Pane layoutX="10.0" layoutY="197.0" prefHeight="403.0" prefWidth="10.0"
        style="-fx-background-color: #f1c40f;"/>
    <Pane layoutX="590.0" prefHeight="600.0" prefWidth="10.0" style="-fx-background-color:
#8e44ad;"/>
    <Pane layoutX="20.0" layoutY="590.0" prefHeight="10.0" prefWidth="561.0" style="-fx-background-
color: #3498db;"/>
    <Pane layoutX="581.0" layoutY="145.0" prefHeight="455.0" prefWidth="10.0"
        style="-fx-background-color: #27ae60;"/>
    <Pane layoutX="135.0" layoutY="140.0" prefHeight="5.0" prefWidth="360.0"
        style="-fx-background-color: #b30000;"/>
</children>
</AnchorPane>

```

3.52 AceHomeController.java

```

package Controller;

import javafx.animation.ScaleTransition;
import javafx.animation.TranslateTransition;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.effect.DropShadow;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;

```



```

import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.util.Duration;
import java.io.IOException;

public class AceHomeController {

    public AnchorPane root;
    public ImageView membership;
    public ImageView equipment;
    public ImageView eqp_borrowed;
    public ImageView eqp_returned;
    public ImageView eqp_search;

    //navigate windows
    public void navigate(MouseEvent event) throws IOException {

        if (event.getSource() instanceof ImageView) {
            ImageView icon = (ImageView) event.getSource();

            Parent root = null;

            switch (icon.getId()) {
                case "membership":
                    root = FXMLLoader.load(this.getClass().getResource("/View/MembershipView.fxml"));
                    break;
                case "equipment":
                    root = FXMLLoader.load(this.getClass().getResource("/View/EquipmentView.fxml"));
                    break;
                case "eqp_borrowed":
                    root =
FXMLLoader.load(this.getClass().getResource("/View/EquipmentBorrowedView.fxml"));
                    break;

```

```

        case "eqp_returned":
            root =
FXMLLoader.load(this.getClass().getResource("/View/EquipmentReturnedView.fxml"));
            break;
        case "eqp_search":
            root = FXMLLoader.load(this.getClass().getResource("/View/EquipmentSearchView.fxml"));
            break;
    }

    if (root != null) {
        Scene subScene = new Scene(root);
        Stage primaryStage = (Stage) this.root.getScene().getWindow();
        primaryStage.setScene(subScene);
        primaryStage.centerOnScreen();

        TranslateTransition tt = new TranslateTransition(Duration.millis(350), subScene.getRoot());
        tt.setFromX(-subScene.getWidth());
        tt.setToX(0);
        tt.play();

    }
}

//mouse hover animations
public void playMouseEnterAnimation(MouseEvent mouseEvent) {
    if (mouseEvent.getSource() instanceof ImageView) {
        ImageView icon = (ImageView) mouseEvent.getSource();
        ScaleTransition scaleT = new ScaleTransition(Duration.millis(200), icon);
        scaleT.setToX(1);
        scaleT.setToY(1);
        scaleT.play();
        icon.setEffect(null);

        DropShadow glow = new DropShadow();
    }
}

```

```

        glow.setColor(Color.RED);
        glow.setWidth(20);
        glow.setHeight(20);
        glow.setRadius(20);
        icon.setEffect(glow);
    }
}

public void playMouseExitAnimation(MouseEvent mouseEvent) {
    if (mouseEvent.getSource() instanceof ImageView) {
        ImageView icon = (ImageView) mouseEvent.getSource();

        ScaleTransition scaleT = new ScaleTransition(Duration.millis(200), icon);
        scaleT.setToX(1.2);
        scaleT.setToY(1.2);
        scaleT.play();

        DropShadow glow = new DropShadow();
        glow.setColor(Color.YELLOW);
        glow.setWidth(20);
        glow.setHeight(20);
        glow.setRadius(20);
        icon.setEffect(glow);
    }
}
}

```

3.6 Equipment Issuing to Members

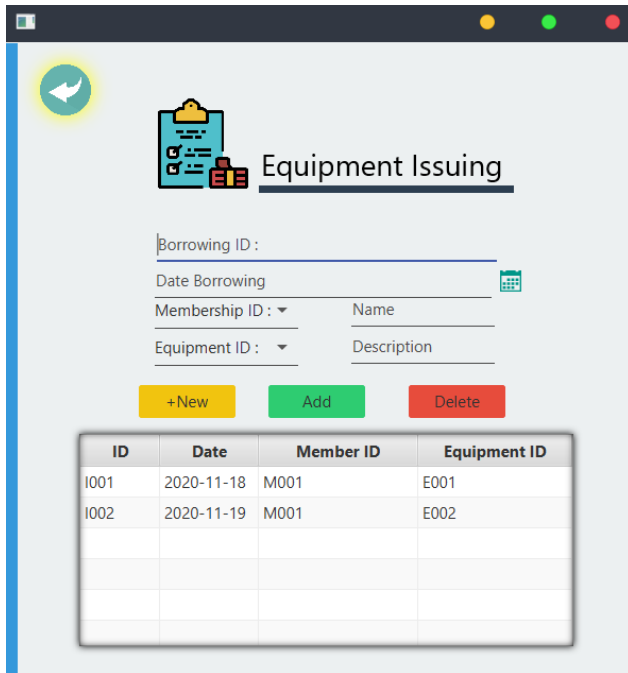


Figure 3.8 Screenshot of Equipment Borrowed View

3.61 EquipmentBorrowedView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import com.jfoenix.controls.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.effect.InnerShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>
<?import javafx.scene.text.Text?>
<AnchorPane xmlns:fx="http://javafx.com/fxml/1" fx:id="eqp_brw" maxHeight="-Infinity" maxWidth="-Infinity"
    minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="600.0"
    style="-fx-background-color: #ecf0f1;" xmlns="http://javafx.com/javafx/10.0.1"
    fx:controller="Controller.EquipmentBorrowedController">
    <children>
```

```

<ImageView fitHeight="40.0" fitWidth="40.0" layoutX="36.0" layoutY="25.0"
onMouseClicked="#back_click"
    onMouseExited="#playMouseEnterAnimation" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@../Images/back.png"/>
    </image>
</ImageView>
<Text layoutX="240.0" layoutY="126.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Issuing">
    <font>
        <Font size="28.0"/>
    </font>
</Text>
<ImageView fitHeight="91.0" fitWidth="98.0" layoutX="140.0" layoutY="50.0"
pickOnBounds="true"
    preserveRatio="true">
    <image>
        <Image url="@../Images/eqpborrowed.png"/>
    </image>
</ImageView>
<Pane layoutX="-1.0" layoutY="1.0" prefHeight="600.0" prefWidth="12.0" style="-fx-background-
color: #3498db;"/>
    <Pane layoutX="589.0" prefHeight="600.0" prefWidth="12.0" style="-fx-background-color:
#2c3e50;"/>
    <JFXTextField fx:id="txt_brw_id" layoutX="142.0" layoutY="176.0" prefHeight="25.0"
prefWidth="320.0"
        promptText="Borrowing ID :"/>
    <JFXDatePicker fx:id="txt_brw_date" layoutX="140.0" layoutY="210.0" prefHeight="28.0"
prefWidth="353.0"
        promptText="Date Borrowing"/>
    <JFXTextField fx:id="txt_name" layoutX="325.0" layoutY="237.0" prefHeight="25.0"
prefWidth="135.0"
        promptText="Name"/>
    <JFXTextField fx:id="txt_descpn" layoutX="325.0" layoutY="272.0" prefHeight="25.0"
prefWidth="135.0"

```

```

        promptText="Description"/>
    <JFXComboBox fx:id="member_brw_id" layoutX="140.0" layoutY="237.0" prefHeight="25.0"
prefWidth="135.0"
        promptText="Membership ID :"/>
    <JFXComboBox fx:id="equipment_id" layoutX="140.0" layoutY="272.0" prefHeight="25.0"
prefWidth="135.0"
        promptText="Equipment ID :"/>
    <JFXButton layoutX="125.0" layoutY="323.0" onAction="#new_action" prefHeight="25.0"
prefWidth="91.0"
        style="-fx-background-color: #f1c40f;" text="+New"/>
    <JFXButton layoutX="247.0" layoutY="323.0" onAction="#add_Action" prefHeight="25.0"
prefWidth="91.0"
        style="-fx-background-color: #2ecc71;" text="Add"/>
    <JFXButton layoutX="379.0" layoutY="323.0" onAction="#delete_Action" prefHeight="25.0"
prefWidth="91.0"
        style="-fx-background-color: #e74c3c;" text="Delete"/>
    <TableView fx:id="eqp_brw_tbl" layoutX="69.0" layoutY="369.0" prefHeight="200.0"
prefWidth="465.0">
        <columns>
            <TableColumn fx:id="brw_id" prefWidth="75.0" text="ID"/>
            <TableColumn fx:id="brw_date" prefWidth="93.0" text="Date"/>
            <TableColumn fx:id="mem_name" prefWidth="150.0" text="Member ID"/>
            <TableColumn fx:id="eqpp_id" prefWidth="145.0" text="Equipment ID"/>
        </columns>
        <effect>
            <InnerShadow>
                <input>
                    <DropShadow/>
                </input>
            </InnerShadow>
        </effect>
    </TableView>
    <Pane layoutX="238.0" layoutY="135.0" prefHeight="7.0" prefWidth="240.0"
        style="-fx-background-color: #2c3e50;"/>
</children>

```

3.62 EquipmentBorrowedController.java

```
package Controller;

import Model.EquipmentBorrowedModel;
import Model.EquipmentModel;
import com.jfoenix.controls.JFXComboBox;
import com.jfoenix.controls.JFXDatePicker;
import com.jfoenix.controls.JFXTextField;
import db.DB;
import db.DBConnection;
import javafx.animation.ScaleTransition;
import javafx.animation.TranslateTransition;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.effect.DropShadow;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.util.Duration;
```

```

import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Optional;

public class EquipmentBorrowedController {
    public JFXTextField txt_brw_id;
    public JFXDatePicker txt_brw_date;
    public JFXTextField txt_name;
    public JFXTextField txt_descpn;
    public JFXComboBox member_brw_id;
    public JFXComboBox equipment_id;
    public TableView<EquipmentBorrowedModel> eqp_brw_tbl;
    public AnchorPane eqp_brw;
    private Connection connection;

    //JDBC
    private PreparedStatement selectAll;
    private PreparedStatement selectmemID;
    private PreparedStatement selecteqdtl;
    private PreparedStatement table;
    private PreparedStatement delete;

    public void initialize() throws ClassNotFoundException {
        Class.forName("com.mysql.jdbc.Driver");

        eqp_brw_tbl.getColumns().get(0).setCellValueFactory(new PropertyValueFactory<>("brwId"));
        eqp_brw_tbl.getColumns().get(1).setCellValueFactory(new PropertyValueFactory<>("date"));
        eqp_brw_tbl.getColumns().get(2).setCellValueFactory(new PropertyValueFactory<>("memberId"));
        eqp_brw_tbl.getColumns().get(3).setCellValueFactory(new PropertyValueFactory<>("eqpId"));
    }
}

```



```

try {
    connection = DBConnection.getInstance().getConnection();
    ObservableList<EquipmentBorrowedModel> issued = eqp_brw_tbl.getItems();

    selectAll = connection.prepareStatement("SELECT * FROM equipmentborrowed");
    selectmemID = connection.prepareStatement("select name from membershipdetail where id=?");
    selecteqdtl = connection.prepareStatement("select description,status from equipmentdetail where
id=?");
    table = connection.prepareStatement("INSERT INTO equipmentborrowed values(?,?,?,?)");
    delete = connection.prepareStatement("DELETE FROM equipmentborrowed WHERE
borrowedId=?");
    ResultSet rst = selectAll.executeQuery();

    while (rst.next()) {
        System.out.println("load");
        issued.add(new EquipmentBorrowedModel(rst.getString(1),
            rst.getString(2),
            rst.getString(3),
            rst.getString(4)));
    }

    eqp_brw_tbl.setItems(issued);
    member_brw_id.getItems().clear();
    ObservableList cmbmembers = member_brw_id.getItems();
    String sql2 = "select id from membershipdetail";
    PreparedStatement pstmt1 = connection.prepareStatement(sql2);
    ResultSet rst1 = pstmt1.executeQuery();

    while (rst1.next()) {
        cmbmembers.add(rst1.getString(1));
    }

    equipment_id.getItems().clear();
    ObservableList cmbequipment = equipment_id.getItems();

```

```

String sql3 = "select id from equipmentdetail";
PreparedStatement pstmt2 = connection.prepareStatement(sql3);
ResultSet rst2 = pstmt2.executeQuery();
while (rst2.next()) {
    cmbequipment.add(rst2.getString(1));
}
} catch (SQLException e) {
    e.printStackTrace();
}
member_brw_id.getSelectionModel().selectedItemProperty().addListener(new ChangeListener() {
    @Override
    public void changed(ObservableValue observable, Object oldValue, Object newValue) {

        if (member_brw_id.getSelectionModel().getSelectedItem() != null) {
            Object selectedItem = member_brw_id.getSelectionModel().getSelectedItem();
            if (selectedItem.equals(null) || member_brw_id.getSelectionModel().isEmpty()) {
                return;
            }
            try {
                selectmemID.setString(1, selectedItem.toString());
                ResultSet rst = selectmemID.executeQuery();

                if (rst.next()) {
                    txt_name.setText(rst.getString(1));
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
});

```

```

equipment_id.getSelectionModel().selectedItemProperty().addListener(new ChangeListener() {
    @Override
    public void changed(ObservableValue observable, Object oldValue, Object newValue) {

```

```

if (equipment_id.getSelectionModel().getSelectedItem() != null) {
    Object selectedItem = equipment_id.getSelectionModel().getSelectedItem();

    try {
        txt_descpn.clear();
        selecteqdtl.setString(1, selectedItem.toString());
        ResultSet rst = selecteqdtl.executeQuery();

        if (rst.next()) {
            if (rst.getString(2).equals("Available")) {
                txt_descpn.setText(rst.getString(1));
            } else {
                Alert alert = new Alert(Alert.AlertType.ERROR,
                    "The equipment isn't available!",
                    ButtonType.OK);
                Optional<ButtonType> buttonType = alert.showAndWait();
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
});
}

```

//button new action

```

public void new_action(ActionEvent actionEvent) throws SQLException {
    txt_descpn.clear();
    txt_name.clear();
    member_brw_id.getSelectionModel().clearSelection();
    equipment_id.getSelectionModel().clearSelection();
    txt_brw_date.setPromptText("Date Borrowing");

```

```

String sql = "Select borrowedId from equipmentborrowed";

```

```

PreparedStatement pstmt = connection.prepareStatement(sql);
ResultSet rst = pstmt.executeQuery();

String ids = null;
int maxId = 0;

while (rst.next()) {
    ids = rst.getString(1);

    int id = Integer.parseInt(ids.replace("I", ""));
    if (id > maxId) {
        maxId = id;
    }
}
maxId = maxId + 1;
String id = "";
if (maxId < 10) {
    id = "I00" + maxId;
} else if (maxId < 100) {
    id = "I0" + maxId;
} else {
    id = "I" + maxId;
}
txt_brw_id.setText(id);
}

//button add action
public void add_Action(ActionEvent actionEvent) throws SQLException {

    ObservableList<EquipmentBorrowedModel> issued = FXCollections.observableList(DB.borrowed);
    ObservableList<EquipmentModel> books = FXCollections.observableList(DB.equipment);

    if (txt_brw_id.getText().isEmpty() ||
        equipment_id.getSelectionModel().getSelectedItem().equals(null) ||
        member_brw_id.getSelectionModel().getSelectedItem().equals(null)

```

```

        || txt_brw_date.getValue().toString().equals(null)) {
Alert alert = new Alert(Alert.AlertType.ERROR,
        "Please fill your details.",
        ButtonType.OK);
Optional<ButtonType> buttonType = alert.showAndWait();
return;
} else {
    String memberId = (String) member_brw_id.getSelectionModel().getSelectedItem();
    String eqpId = (String) equipment_id.getSelectionModel().getSelectedItem();
    issued.add(new EquipmentBorrowedModel(txt_brw_id.getText(),
txt_brw_date.getValue().toString(), memberId, eqpId));

    try {
        table.setString(1, txt_brw_id.getText());
        table.setString(2, txt_brw_date.getValue().toString());
        table.setString(3, (String) member_brw_id.getSelectionModel().getSelectedItem());
        table.setString(4, (String) equipment_id.getSelectionModel().getSelectedItem());
        int affectedRows = table.executeUpdate();

        if (affectedRows > 0) {
            System.out.println("Data insertion successfull");
            String sql2 = "Update equipmentdetail SET status=? where id=?";
            PreparedStatement pstm2 = connection.prepareStatement(sql2);
            String id = (String) equipment_id.getSelectionModel().getSelectedItem();
            pstm2.setString(1, "Unavailable");
            pstm2.setString(2, id);
            int affected = pstm2.executeUpdate();

            if (affected > 0) {
                Alert alert = new Alert(Alert.AlertType.INFORMATION,
                "Status updated.",
                ButtonType.OK);
                Optional<ButtonType> buttonType = alert.showAndWait();
            } else {
                System.out.println("ERROR");
            }
        }
    }
}

```

```

        }
    }
} catch (SQLException e) {
    e.printStackTrace();
}
}

try {
    eqp_brw_tbl.getItems().clear();
    initialize();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}

//button delete action
public void delete_Action(ActionEvent actionEvent) throws SQLException {
    //BookIssueTM selectedItem = (BookIssueTM) FXCollections.observableList(DB.issued);
    EquipmentBorrowedModel selectedItem = eqp_brw_tbl.getSelectionModel().getSelectedItem();
    if (eqp_brw_tbl.getSelectionModel().isEmpty()) {
        Alert alert = new Alert(Alert.AlertType.ERROR,
            "Please select a row.",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
        return;
    } else {
        try {
            delete.setString(1, selectedItem.getBrwId());
            delete.executeUpdate();

            String sql2 = "Update equipmentdetail SET status=? where id=?";
            PreparedStatement pstm2 = connection.prepareStatement(sql2);
            String id = (String) equipment_id.getSelectionModel().getSelectedItem();
            pstm2.setString(1, "Available");
            pstm2.setString(2, id);

```

```

        pstmt2.executeUpdate();

        Alert alert = new Alert(Alert.AlertType.INFORMATION,
            "Record deleted!",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
    } catch (SQLException e) {
        e.printStackTrace();
    }

}

try {
    eqp_brw_tbl.getItems().clear();
    initialize();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}

public void back_click(MouseEvent event) throws IOException {
    URL resource = this.getClass().getResource("/View/AceHomeView.fxml");
    Parent root = FXMLLoader.load(resource);
    Scene scene = new Scene(root);
    Stage primaryStage = (Stage) this.eqp_brw.getScene().getWindow();
    primaryStage.setScene(scene);

    TranslateTransition tt = new TranslateTransition(Duration.millis(350), scene.getRoot());
    tt.setFromX(-scene.getWidth());
    tt.setToX(0);
    tt.play();
}

public void playMouseEnterAnimation(MouseEvent event) {
    if (event.getSource() instanceof ImageView) {
        ImageView icon = (ImageView) event.getSource();
    }
}

```

```

ScaleTransition scaleT = new ScaleTransition(Duration.millis(200), icon);
scaleT.setToX(1.2);
scaleT.setToY(1.2);
scaleT.play();

DropShadow glow = new DropShadow();
glow.setColor(Color.YELLOW);
glow.setWidth(20);
glow.setHeight(20);
glow.setRadius(20);
icon.setEffect(glow);
}
}
}

```

3.63 EquipmentBorrowedModel.java

```

package Model;

public class EquipmentBorrowedModel {
    private String brwId;
    private String date;
    private String memberId;
    private String eqpId;

    public EquipmentBorrowedModel(String brwId, String date, String memberId, String eqpId) {
        this.brwId = brwId;
        this.date = date;
        this.memberId = memberId;
        this.eqpId = eqpId;
    }

    public String getBrwId() {
        return brwId;
    }
}

```



```

    }

    public void setBrwId(String brwId) {
        this.brwId = brwId;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public String getMemberId() {
        return memberId;
    }

    public void setMemberId(String memberId) {
        this.memberId = memberId;
    }

    public String getEqpId() {
        return eqpId;
    }

    public void setEqpId(String eqpId) {
        this.eqpId = eqpId;
    }
}

```

3.7 Equipment Returned back by Members to the academy

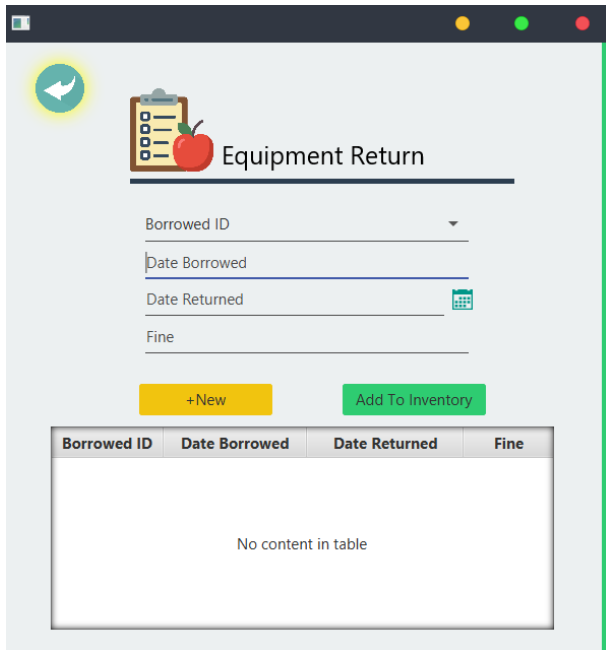


Figure 3.9 Screenshot of Equipment Returned View

3.71 EquipmentReturnedView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import com.jfoenix.controls.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.effect.InnerShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>
<?import javafx.scene.text.Text?>
<AnchorPane xmlns:fx="http://javafx.com/fxml/1" fx:id="Returnroot" maxHeight="-Infinity" maxWidth="-Infinity"
    minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="600.0"
    style="-fx-background-color: #ecf0f1;" xmlns="http://javafx.com/javafx/10.0.1"
    fx:controller="Controller.EquipmentReturnedController">
    <children>
        <ImageView fitHeight="40.0" fitWidth="40.0" layoutX="37.0" layoutY="25.0"
onMouseClicked="#img_back"
            onMouseEntered="#playMouseEnterAnimation" pickOnBounds="true" preserveRatio="true">
```

```

    <image>
      <Image url="@../Images/back.png"/>
    </image>
  </ImageView>
  <ImageView fitHeight="82.0" fitWidth="87.0" layoutX="126.0" layoutY="45.0"
pickOnBounds="true"
      preserveRatio="true">
    <image>
      <Image url="@../Images/eqpreturned.png"/>
    </image>
  </ImageView>
  <Text layoutX="216.0" layoutY="120.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Return"
      wrappingWidth="291.29541015625">
    <font>
      <Font size="25.0"/>
    </font>
  </Text>
  <Pane layoutX="590.0" prefHeight="600.0" prefWidth="11.0" style="-fx-background-color:
#2ecc71;"/>
    <JFXTextField fx:id="txt_brw_date" layoutX="141.0" layoutY="202.0" prefHeight="25.0"
prefWidth="318.0"
      promptText="Date Borrowed"/>
    <JFXTextField fx:id="txt_fine" layoutX="141.0" layoutY="275.0" prefHeight="25.0"
prefWidth="318.0"
      promptText="Fine"/>
    <JFXDatePicker fx:id="txt_rtn_date" layoutX="141.0" layoutY="238.0" prefHeight="28.0"
prefWidth="330.0"
      promptText="Date Returned"/>
    <JFXButton layoutX="135.0" layoutY="336.0" onAction="#btn_new" prefHeight="25.0"
prefWidth="131.0"
      style="-fx-background-color: #f1c40f; text="+New"/>
    <JFXButton layoutX="335.0" layoutY="336.0" onAction="#btn_add_inveb" prefHeight="25.0"
prefWidth="141.0"
      style="-fx-background-color: #2ecc71; text="Add To Inventory"/>

```

```

<TableView fx:id="rt_tbl" layoutX="48.0" layoutY="378.0" prefHeight="200.0" prefWidth="495.0">
    <columns>
        <TableColumn fx:id="iss_id_col" prefWidth="110.0" text="Borrowed ID"/>
        <TableColumn fx:id="iss_date_col" prefWidth="141.0" text="Date Borrowed"/>
        <TableColumn fx:id="ret_date_col" prefWidth="154.0" text="Date Returned"/>
        <TableColumn fx:id="fine_col" minWidth="9.0" prefWidth="90.0" text="Fine"/>
    </columns>
    <effect>
        <InnerShadow/>
    </effect>
</TableView>

<JFXComboBox fx:id="eqp_brwd_id" layoutX="141.0" layoutY="163.0" prefHeight="25.0"
prefWidth="318.0"
    promptText="Borrowed ID"/>

<Pane layoutX="126.0" layoutY="134.0" prefHeight="5.0" prefWidth="378.0"
    style="-fx-background-color: #2c3e50;"/>
</children>
</AnchorPane>

```

3.72 EquipmentReturnedController.java

```

package Controller;

import Model.EquipmentReturnedModel;
import com.jfoenix.controls.JFXComboBox;
import com.jfoenix.controls.JFXDatePicker;
import com.jfoenix.controls.JFXTextField;
import db.DBConnection;
import javafx.animation.ScaleTransition;
import javafx.animation.TranslateTransition;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;

```

```

import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.effect.DropShadow;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.util.Duration;

import java.io.IOException;
import java.net.URL;
import java.sql.*;
import java.time.LocalDate;
import java.util.Optional;
import java.util.concurrent.TimeUnit;

public class EquipmentReturnedController {
    public AnchorPane Returnroot;
    public JFXTextField txt_brw_date;
    public JFXTextField txt_fine;
    public JFXDatePicker txt_rtn_date;
    public TableView<EquipmentReturnedModel> rt_tbl;
    public JFXComboBox eqp_brwd_id;
    private Connection connection;

    public void initialize() throws ClassNotFoundException {
        Class.forName("com.mysql.jdbc.Driver");

        rt_tbl.getColumns().get(0).setCellValueFactory(new PropertyValueFactory<>("id"));
        rt_tbl.getColumns().get(1).setCellValueFactory(new PropertyValueFactory<>("issuedDate"));
    }

```

```

rt_tbl.getColumns().get(2).setCellValueFactory(new PropertyValueFactory<>("returnedDate"));
rt_tbl.getColumns().get(3).setCellValueFactory(new PropertyValueFactory<>("fine"));

try {
    connection = DBConnection.getInstance().getConnection();
    ObservableList<EquipmentReturnedModel> returns = rt_tbl.getItems();

    String sql = "SELECT * from equipmentreturn";
    PreparedStatement pstmt = connection.prepareStatement(sql);
    ResultSet rst = pstmt.executeQuery();
    while (rst.next()) {
        float fine = Float.parseFloat(rst.getString(4));
        returns.add(new EquipmentReturnedModel(rst.getString(1), rst.getString(2), rst.getString(3),
fine));
    }
    rt_tbl.setItems(returns);

    eqp_brwd_id.getItems().clear();
    ObservableList cmbIssue = eqp_brwd_id.getItems();
    String sql2 = "select borrowedId from equipmentborrowed";
    PreparedStatement pstmt1 = connection.prepareStatement(sql2);
    ResultSet rst1 = pstmt1.executeQuery();

    while (rst1.next()) {
        cmbIssue.add(rst1.getString(1));
    }
} catch (SQLException e) {
    e.printStackTrace();
}

eqp_brwd_id.getSelectionModel().selectedItemProperty().addListener(new ChangeListener() {
    @Override
    public void changed(ObservableValue observable, Object oldValue, Object newValue) {
        if (eqp_brwd_id.getSelectionModel().getSelectedItem() == null) {
            return;

```

```

    }
    try {
        String sql = "select date from equipmentborrowed where borrowedId =?";
        PreparedStatement pstmt = connection.prepareStatement(sql);
        pstmt.setString(1, (String) eqp_brwd_id.getSelectionModel().getSelectedItem());
        ResultSet rst = pstmt.executeQuery();
        if (rst.next()) {
            txt_brw_date.setText(rst.getString(1));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

});

txt_rtn_date.valueProperty().addListener(new ChangeListener<LocalDate>() {
    @Override
    public void changed(ObservableValue<? extends LocalDate> observable, LocalDate oldValue,
LocalDate newValue) {

        if (txt_rtn_date.getValue() == null) {
            return;
        }

        LocalDate returned = txt_rtn_date.getValue();
        LocalDate issued = LocalDate.parse(txt_brw_date.getText());

        Date date1 = Date.valueOf(issued);
        Date date2 = Date.valueOf(returned);

        long diff = date2.getTime() - date1.getTime();

        System.out.println(TimeUnit.MILLISECONDS.toDays(diff));
        int dateCount = (int) TimeUnit.MILLISECONDS.toDays(diff);
        float fine = 0;

```

```

        if (dateCount > 14) {
            fine = dateCount * 15;
        }
        txt_fine.setText(Float.toString(fine));
    }
});
}

```

//btn new action

```

public void btn_new(ActionEvent actionEvent) {
    txt_fine.clear();
    txt_brw_date.setPromptText("Date Borrowed");
    eqp_brwd_id.getSelectionModel().clearSelection();
    txt_rtn_date.setPromptText("Date Returned");
}

```

//btn add action

```

public void btn_add_inveb(ActionEvent actionEvent) throws SQLException {
    if (eqp_brwd_id.getSelectionModel().isEmpty() ||
        txt_brw_date.getText().isEmpty() ||
        txt_rtn_date.getValue() == null ||
        txt_fine.getText().isEmpty()
    ) {
        Alert alert = new Alert(Alert.AlertType.ERROR,
            "Please fill details.",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
        System.out.println("Fill Your empty fields!");
        return;
    }
}

```

```

String issueID = (String) eqp_brwd_id.getSelectionModel().getSelectedItem();
String sql = "INSERT INTO equipmentreturn VALUES (?, ?, ?, ?)";
PreparedStatement pstm = connection.prepareStatement(sql);
pstm.setString(1, (String) eqp_brwd_id.getSelectionModel().getSelectedItem());

```



```

pstm.setString(2, txt_brw_date.getText());
pstm.setString(3, txt_rtn_date.getValue().toString());
pstm.setString(4, txt_fine.getText());
int affectedRows = pstm.executeUpdate();

if (affectedRows > 0) {
    System.out.println("successful");
    String sql4 = "Update equipmentdetail SET status=? where id=?";
    PreparedStatement pstm2 = connection.prepareStatement(sql4);

    String sql3 = "select equipmentid from equipmentborrowed where borrowedId=?";
    PreparedStatement pstm3 = connection.prepareStatement(sql3);
    pstm3.setString(1, (String) eqp_brwd_id.getSelectionModel().getSelectedItem());
    ResultSet rst3 = pstm3.executeQuery();
    String id = null;

    if (rst3.next()) {
        id = rst3.getString(1);
    }
    pstm2.setString(1, "Available");
    pstm2.setString(2, id);
    int affected = pstm2.executeUpdate();
    if (affected > 0) {
        Alert alert = new Alert(Alert.AlertType.INFORMATION,
            "Status updated.",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
    }
} else {
    System.out.println("Something went wrong");
}
try {
    rt_tbl.getItems().clear();
    initialize();
} catch (ClassNotFoundException e) {

```

```

        e.printStackTrace();
    }
    connection.close();
    eqp_brwd_id.getItems().remove(issueID);
    eqp_brwd_id.getSelectionModel().clearSelection();
    txt_brw_date.clear();
    txt_fine.clear();
    txt_rtn_date.getEditor().clear();
}

public void img_back(MouseEvent event) throws IOException {
    URL resource = this.getClass().getResource("/View/AceHomeView.fxml");
    Parent root = FXMLLoader.load(resource);
    Scene scene = new Scene(root);
    Stage primaryStage = (Stage) this.Returnroot.getScene().getWindow();
    primaryStage.setScene(scene);

    TranslateTransition tt = new TranslateTransition(Duration.millis(350), scene.getRoot());
    tt.setFromX(-scene.getWidth());
    tt.setToX(0);
    tt.play();
}

public void playMouseEnterAnimation(MouseEvent event) {
    if (event.getSource() instanceof ImageView) {
        ImageView icon = (ImageView) event.getSource();

        ScaleTransition scaleT = new ScaleTransition(Duration.millis(200), icon);
        scaleT.setToX(1.2);
        scaleT.setToY(1.2);
        scaleT.play();

        DropShadow glow = new DropShadow();
        glow.setColor(Color.YELLOW);
        glow.setWidth(20);
    }
}

```

```
        glow.setHeight(20);
        glow.setRadius(20);
        icon.setEffect(glow);
    }
}
```

3.73 EquipmentReturnedModel.java

```
package Model;
```

```
public class EquipmentReturnedModel {
    private String id;
    private String borrowedDate;
    private String returnedDate;
    private float fine;
```

```
    public EquipmentReturnedModel(String id, String borrowedDate, String returnedDate, float fine) {
        this.id = id;
        this.borrowedDate = borrowedDate;
        this.returnedDate = returnedDate;
        this.fine = fine;
    }
```

```
    public String getId() {
        return id;
    }
```

```
    public void setId(String id) {
        this.id = id;
    }
```

```
    public String getBorrowedDate() {
        return borrowedDate;
```

```
}

public void setBorrowedDate(String borrowedDate) {
    this.borrowedDate = borrowedDate;
}

public String getReturnedDate() {
    return returnedDate;
}

public void setReturnedDate(String returnedDate) {
    this.returnedDate = returnedDate;
}

public float getFine() {
    return fine;
}

public void setFine(float fine) {
    this.fine = fine;
}
}
```

3.8 Equipment Searching to identify Stock Available

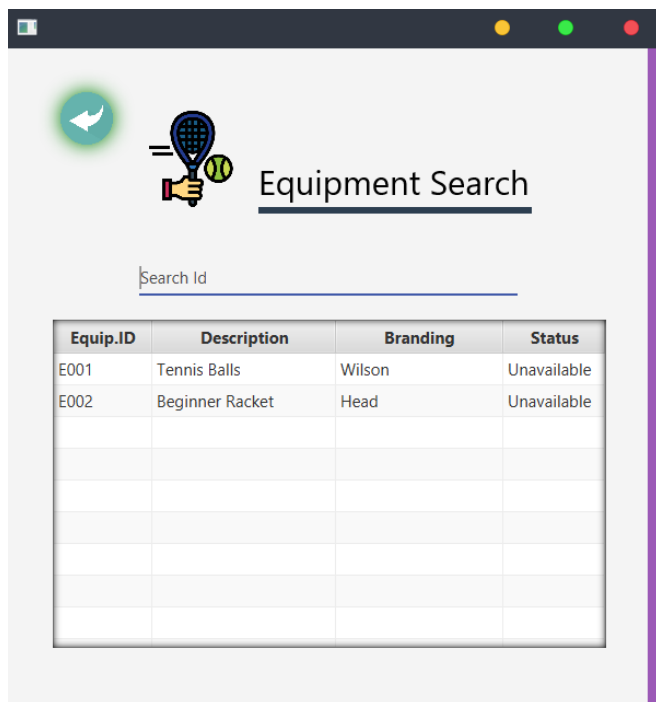


Figure 3.10 Screenshot of Equipment Search View

3.81 EquipmentSearchView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import com.jfoenix.controls.JFXTextField?>
<?import javafx.scene.control.*?>
<?import javafx.scene.effect.InnerShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
```

```

<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>
<?import javafx.scene.text.Text?>
<AnchorPane xmlns:fx="http://javafx.com/fxml/1" fx:id="search_root" maxHeight="-Infinity"
maxWidth="-Infinity"
    minHeight="-Infinity" prefHeight="600.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/10.0.1"
    fx:controller="Controller.EquipmentSearchController">
<children>
    <ImageView fitHeight="40.0" fitWidth="40.0" layoutX="53.0" layoutY="44.0"
onMouseClicked="#img_bk"
        onMouseExited="#playMouseEnterAnimation" pickOnBounds="true" preserveRatio="true">
        <image>
            <Image url="@../Images/back.png"/>
        </image>
    </ImageView>
    <ImageView fitHeight="94.0" fitWidth="101.0" layoutX="120.0" layoutY="54.0"
pickOnBounds="true"
        preserveRatio="true">
        <image>
            <Image url="@../Images/eqpsearch.png"/>
        </image>
    </ImageView>
    <Text layoutX="230.0" layoutY="134.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Search">
        <font>
            <Font size="31.0"/>
        </font>
    </Text>
    <JFXTextField fx:id="eqp_search" layoutX="121.0" layoutY="195.0" prefHeight="25.0"
prefWidth="346.0"
        promptText="Search Id"/>
    <TableView fx:id="eqp_table" layoutX="42.0" layoutY="248.0" maxHeight="-Infinity" maxWidth="-
Infinity"
        minHeight="-Infinity" minWidth="-Infinity" prefHeight="300.0" prefWidth="506.0">

```

```

<columns>
  <TableColumn fx:id="eqp_id_col" prefWidth="90.0" text="Equip.ID"/>
  <TableColumn fx:id="eqp_dscpn_col" prefWidth="168.0" text="Description"/>
  <TableColumn fx:id="eqp_brand_col" prefWidth="153.0" text="Branding"/>
  <TableColumn fx:id="eqp_sts_col" prefWidth="93.0" text="Status"/>
</columns>
<effect>
  <InnerShadow/>
</effect>
</TableView>
<Pane layoutX="586.0" prefHeight="600.0" prefWidth="14.0" style="-fx-background-color:
#9b59b6;"/>
  <Pane layoutX="230.0" layoutY="145.0" prefHeight="6.0" prefWidth="250.0"
    style="-fx-background-color: #2c3e50;"/>
</children>
</AnchorPane>

```

3.82 EquipmentSearchController.java

```

package Controller;

import Model.EquipmentModel;
import com.jfoenix.controls.JFXTextField;
import javafx.animation.ScaleTransition;
import javafx.animation.TranslateTransition;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.ObservableList;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.effect.DropShadow;

```

```

import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.util.Duration;

import java.io.IOException;
import java.net.URL;
import java.sql.*;

public class EquipmentSearchController{
    public JFXTextField eqp_search;
    public TableView<EquipmentModel> eqp_table;
    public AnchorPane search_root;
    private Connection connection;

    public void initialize() {

        eqp_table.getColumns().get(0).setCellValueFactory(new PropertyValueFactory<>("id"));
        eqp_table.getColumns().get(1).setCellValueFactory(new PropertyValueFactory<>("description"));
        eqp_table.getColumns().get(2).setCellValueFactory(new PropertyValueFactory<>("branding"));
        eqp_table.getColumns().get(3).setCellValueFactory(new PropertyValueFactory<>("status"));

        try {
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3307/aceacademy", "root", "");
            ObservableList<EquipmentModel> equipment = eqp_table.getItems();

            String sql = "SELECT * from equipmentdetail";
            PreparedStatement pstmt = connection.prepareStatement(sql);
            ResultSet rst = pstmt.executeQuery();
            while (rst.next()) {
                System.out.println("Initializer");
                equipment.add(new EquipmentModel(rst.getString(1), rst.getString(2), rst.getString(3),
rst.getString(4)));
            }
        }
    }
}

```



```

    }
    eqp_table.setItems(equipment);
    //connection.close();
} catch (SQLException e) {
    e.printStackTrace();
}

eqp_search.textProperty().addListener(new ChangeListener<String>() {
    @Override
    public void changed(ObservableValue<? extends String> observable, String oldValue, String
newValue) {
        String searchText = eqp_search.getText();

        try {
            eqp_table.getItems().clear();

            try {
                Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3307/aceacademy", "root", "");
                String sql = "Select * FROM equipmentdetail where id like ? OR description like ? OR
branding like ?";
                PreparedStatement pstmt = connection.prepareStatement(sql);
                String like = "%" + searchText + "%";
                pstmt.setString(1, like);
                pstmt.setString(2, like);
                pstmt.setString(3, like);

                ResultSet rst = pstmt.executeQuery();

                ObservableList tbl = eqp_table.getItems();
                tbl.clear();

                while (rst.next()) {

```

```

        tbl.add(new EquipmentModel(rst.getString(1), rst.getString(2), rst.getString(3),
rst.getString(4)));
    }

    connection.close();
} catch (SQLException e) {
    e.printStackTrace();
}

} catch (NullPointerException n) {
    return;
}

}
});

```

```

}

```

```

public void img_bk(MouseEvent event) throws IOException {
    URL resource = this.getClass().getResource("/View/AceHomeView.fxml");
    Parent root = FXMLLoader.load(resource);
    Scene scene = new Scene(root);
    Stage primaryStage = (Stage) this.search_root.getScene().getWindow();
    primaryStage.setScene(scene);

    TranslateTransition tt = new TranslateTransition(Duration.millis(350), scene.getRoot());
    tt.setFromX(-scene.getWidth());
    tt.setToX(0);
    tt.play();
}

```

```

public void playMouseEnterAnimation(MouseEvent event) {
    if (event.getSource() instanceof ImageView) {

```

```

    ImageView icon = (ImageView) event.getSource();

    ScaleTransition scaleT = new ScaleTransition(Duration.millis(200), icon);
    scaleT.setToX(1.2);
    scaleT.setToY(1.2);
    scaleT.play();

    DropShadow glow = new DropShadow();
    glow.setColor(Color.GREEN);
    glow.setWidth(20);
    glow.setHeight(20);
    glow.setRadius(20);
    icon.setEffect(glow);
}
}
}

```

3.83 EquipmentModel.java

```

package Model;

public class EquipmentModel {
    private String id;
    private String description;
    private String branding;
    private String status;

    public EquipmentModel(String id, String description, String branding, String status) {
        this.id = id;
        this.description = description;
        this.branding = branding;
        this.status = status;
    }
}

```

```
public String getId() {  
    return id;  
}
```

```
public void setId(String id) {  
    this.id = id;  
}
```

```
public String getDescription() {  
    return description;  
}
```

```
public void setDescription(String description) {  
    this.description = description;  
}
```

```
public String getBranding() {  
    return branding;  
}
```

```
public void setBranding(String branding) {  
    this.branding = branding;  
}
```

```
public String getStatus() {  
    return status;  
}
```

```
public void setStatus(String status) {  
    this.status = status;  
}
```

```
@Override  
public String toString() {  
    return "EquipmentModel{" +
```

```

        "id='" + id + "'" +
        ", description='" + description + "'" +
        ", branding='" + branding + "'" +
        ", status='" + status + "'" +
        '}';
    }
}

```

3.9 Equipment View to see the detailed list and status of Equipment

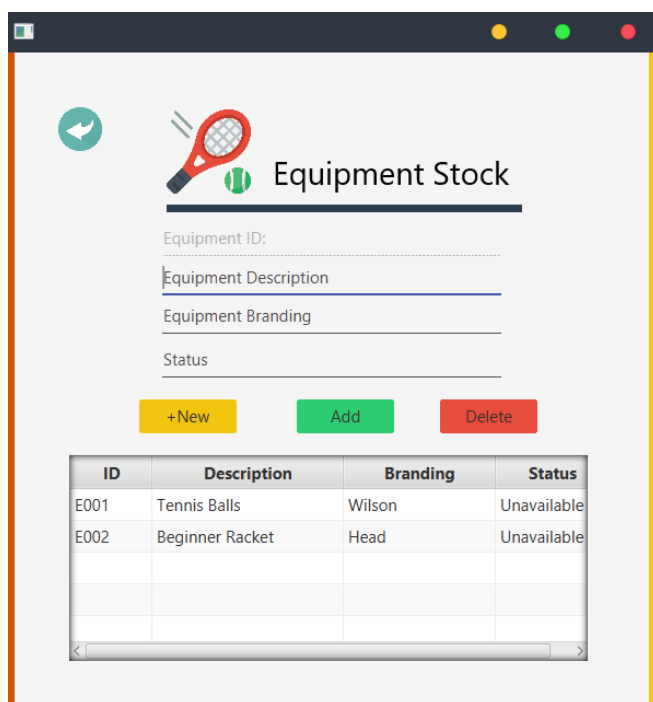


Figure 3.11 Screenshot of Equipment Stock View

3.91 EquipmentView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>
<?import com.jfoenix.controls.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.effect.InnerShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>

```

```

<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>
<?import javafx.scene.text.Text?>
<AnchorPane xmlns:fx="http://javafx.com/fxml/1" fx:id="eqp_root" maxHeight="-Infinity" maxWidth="-Infinity"
    minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="600.0"
    xmlns="http://javafx.com/javafx/10.0.1" fx:controller="Controller.EquipmentController">
    <children>
        <ImageView fx:id="img_back" fitHeight="40.0" fitWidth="40.0" layoutX="50.0" layoutY="50.0"
            onMouseClicked="#img_back" onMouseExited="#playMouseEnterAnimation"
pickOnBounds="true"
            preserveRatio="true">
            <image>
                <Image url="@../Images/back.png"/>
            </image>
        </ImageView>
        <ImageView fitHeight="77.0" fitWidth="89.0" layoutX="149.0" layoutY="52.0"
pickOnBounds="true"
            preserveRatio="true">
            <image>
                <Image url="@../Images/equipment.png"/>
            </image>
        </ImageView>
        <Text layoutX="246.0" layoutY="121.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Equipment Stock">
            <font>
                <Font size="29.0"/>
            </font>
        </Text>
        <JFXTextField fx:id="eqp_id" layoutX="145.0" layoutY="155.0" prefHeight="25.0"
prefWidth="310.0"
            promptText="Equipment ID:"/>
        <JFXTextField fx:id="eqp_dscpn" layoutX="145.0" layoutY="191.0" prefHeight="25.0"
prefWidth="310.0"

```

```

        promptText="Equipment Description"/>
        <JFXTextField fx:id="eqp_brand" layoutX="145.0" layoutY="226.0" prefHeight="25.0"
prefWidth="310.0"
        promptText="Equipment Branding"/>
        <JFXTextField fx:id="eqp_st" layoutX="145.0" layoutY="266.0" prefHeight="25.0"
prefWidth="310.0"
        promptText="Status"/>
        <JFXButton fx:id="btn_new" layoutX="124.0" layoutY="317.0" onAction="#btn_new"
prefHeight="25.0"
        prefWidth="89.0" style="-fx-background-color: #f1c40f;" text="+New"/>
        <JFXButton fx:id="btn_add" layoutX="268.0" layoutY="317.0" onAction="#btn_Add"
prefHeight="25.0"
        prefWidth="89.0" style="-fx-background-color: #2ecc71;" text="Add"/>
        <JFXButton fx:id="btn_dlt" layoutX="399.0" layoutY="317.0" onAction="#btn_dlt"
prefHeight="25.0"
        prefWidth="89.0" style="-fx-background-color: #e74c3c;" text="Delete"/>
        <TableView fx:id="eqp_table" layoutX="60.0" layoutY="368.0" prefHeight="188.0"
prefWidth="474.0">
        <columns>
        <TableColumn fx:id="eqp_id_col" prefWidth="75.0" text="ID"/>
        <TableColumn fx:id="eqp_dscpn_col" prefWidth="175.0" text="Description"/>
        <TableColumn fx:id="eqp_brand_col" prefWidth="139.0" text="Branding"/>
        <TableColumn fx:id="eqp_st_col" minWidth="0.0" prefWidth="104.0" text="Status"/>
        </columns>
        <effect>
        <InnerShadow/>
        </effect>
        </TableView>
        <Pane layoutX="590.0" prefHeight="600.0" prefWidth="10.0" style="-fx-background-color:
#f1c40f;"/>
        <Pane prefHeight="600.0" prefWidth="10.0" style="-fx-background-color: #d35400;"/>
        <Pane layoutX="149.0" layoutY="139.0" prefHeight="7.0" prefWidth="325.0"
        style="-fx-background-color: #2c3e50;"/>
        </children>
</AnchorPane>

```

3.92 EquipmentController.java

```
package Controller;

import Model.EquipmentModel;
import com.jfoenix.controls.JFXButton;
import com.jfoenix.controls.JFXTextField;
import db.DB;
import db.DBConnection;
import javafx.animation.ScaleTransition;
import javafx.animation.TranslateTransition;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.effect.DropShadow;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.util.Duration;

import java.io.IOException;
import java.net.URL;
```



```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Optional;

public class EquipmentController {
    public JFXTextField eqp_id;
    public JFXTextField eqp_dscpn;
    public JFXTextField eqp_brand;
    public JFXTextField eqp_st;
    public TableView<EquipmentModel> eqp_table;
    public AnchorPane eqp_root;
    public JFXButton btn_add;
    private Connection connection;

    //JDBC
    private PreparedStatement selectall;
    private PreparedStatement selectID;
    private PreparedStatement newIdQuery;
    private PreparedStatement addToTable;
    private PreparedStatement updateQuarary;
    private PreparedStatement deleteQuarary;

    public void initialize() throws ClassNotFoundException {
        //disable id field
        eqp_id.setDisable(true);

        //load table
        eqp_table.getColumns().get(0).setCellValueFactory(new PropertyValueFactory<>("id"));
        eqp_table.getColumns().get(1).setCellValueFactory(new PropertyValueFactory<>("description"));
        eqp_table.getColumns().get(2).setCellValueFactory(new PropertyValueFactory<>("branding"));
        eqp_table.getColumns().get(3).setCellValueFactory(new PropertyValueFactory<>("status"));

        Class.forName("com.mysql.jdbc.Driver");
    }
}

```

```

try {
    connection = DBConnection.getInstance().getConnection();
    selectall = connection.prepareStatement("SELECT * from equipmentdetail");
    updateQuarary = connection.prepareStatement("UPDATE equipmentdetail SET description=? ,
branding=? , status=? where id=?");
    selectID = connection.prepareStatement("select * from equipmentdetail where id=?");
    addToTable = connection.prepareStatement("INSERT INTO equipmentdetail values(?,?,?,?)");
    newIdQuery = connection.prepareStatement("SELECT id from equipmentdetail");
    deleteQuarary = connection.prepareStatement("DELETE from equipmentdetail where id=?");
    ObservableList<EquipmentModel> members = eqp_table.getItems();
    ResultSet rst = selectall.executeQuery();
    while (rst.next()) {
        System.out.println("load");
        members.add(new EquipmentModel(
            rst.getString(1),
            rst.getString(2),
            rst.getString(3),
            rst.getString(4)
        ));
    }
    eqp_table.setItems(members);
} catch (SQLException e) {
    e.printStackTrace();
}

eqp_table.getSelectionModel().selectedItemProperty().addListener(new
ChangeListener<EquipmentModel>() {
    @Override
    public void changed(ObservableValue<? extends EquipmentModel> observable, EquipmentModel
oldValue, EquipmentModel newValue) {
        EquipmentModel selectedItem = eqp_table.getSelectionModel().getSelectedItem();
        try {
            connection = null;
            try {

```

```

        selectID.setString(1, selectedItem.getId());
        ResultSet rst = selectID.executeQuery();

        if (rst.next()) {
            eqp_id.setText(rst.getString(1));
            eqp_dscpn.setText(rst.getString(2));
            eqp_brand.setText(rst.getString(3));
            eqp_st.setText(rst.getString(4));
            eqp_id.setDisable(true);
            btn_add.setText("Update");
        }
        btn_add.setText("Update");
    } catch (SQLException e) {
        e.printStackTrace();
    }
    } catch (NullPointerException n) {
        return;
    }
    });
}

//button new action
public void btn_new(ActionEvent actionEvent) throws SQLException {
    btn_add.setText("Add");
    eqp_st.setText("Available");
    eqp_st.setDisable(true);
    eqp_id.setDisable(false);
    eqp_brand.clear();
    eqp_dscpn.clear();
    eqp_dscpn.requestFocus();

    ResultSet rst = newIdQuery.executeQuery();

    String ids = null;

```

```

int maxId = 0;

while (rst.next()) {
    ids = rst.getString(1);

    int id = Integer.parseInt(ids.replace("E", ""));
    if (id > maxId) {
        maxId = id;
    }
}
maxId = maxId + 1;
String id = "";
if (maxId < 10) {
    id = "E00" + maxId;
} else if (maxId < 100) {
    id = "E0" + maxId;
} else {
    id = "E" + maxId;
}
eqp_id.setText(id);
}

//button add action
public void btn_Add(ActionEvent actionEvent) throws SQLException {
    ObservableList<EquipmentModel> books = FXCollections.observableList(DB.equipment);

    if (eqp_id.getText().isEmpty() || eqp_dscpn.getText().isEmpty() || eqp_brand.getText().isEmpty()) {
        Alert alert = new Alert(Alert.AlertType.ERROR,
            "Please fill your details.",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
        return;
    }

    //reg ex

```

```

        if (!(eqp_dscpn.getText().matches("^\\b([A-Za-z.]+\\s?)+$") &&
eqp_brand.getText().matches("^\\b([A-Za-z.]+\\s?)+$"))) {
            new Alert(Alert.AlertType.ERROR, "Enter Valid Name").show();
            return;
        }
        if (btn_add.getText().equals("Add")) {
            addToTable.setString(1, eqp_id.getText());
            addToTable.setString(2, eqp_dscpn.getText());
            addToTable.setString(3, eqp_brand.getText());
            addToTable.setString(4, eqp_st.getText());

            int affectedRows = addToTable.executeUpdate();

            if (affectedRows > 0) {
                System.out.println("Data insertion successfull");
            } else {
                System.out.println("ERROR");
            }
        } else {
            if (btn_add.getText().equals("Update")) {
                for (int i = 0; i < books.size(); i++) {
                    if (eqp_id.getText().equals(books.get(i).getId())) {
                        updateQuarary.setString(1, eqp_dscpn.getText());
                        updateQuarary.setString(2, eqp_brand.getText());
                        updateQuarary.setString(3, eqp_st.getText());
                        updateQuarary.setString(4, eqp_id.getText());
                        int affected = updateQuarary.executeUpdate();

                        if (affected > 0) {
                            Alert alert = new Alert(Alert.AlertType.INFORMATION,
                                "Record updated!",
                                ButtonType.OK);
                            Optional<ButtonType> buttonType = alert.showAndWait();
                        } else {
                            Alert alert = new Alert(Alert.AlertType.ERROR,

```

```

        "Update error!",
        ButtonType.OK);
    Optional<ButtonType> buttonType = alert.showAndWait();
    }
    }
    }
    eqp_table.setItems(books);
    }
}
try {
    eqp_table.getItems().clear();
    initialize();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}

//button delete action
public void btn_dlt(ActionEvent actionEvent) throws SQLException {
    EquipmentModel selectedItem = eqp_table.getSelectionModel().getSelectedItem();
    if (eqp_table.getSelectionModel().isEmpty()) {
        Alert alert = new Alert(Alert.AlertType.ERROR,
            "Please select a member.",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
        return;
    } else {
        deleteQuarary.setString(1, selectedItem.getId());
        int affected = deleteQuarary.executeUpdate();

        if (affected > 0) {
            Alert alert = new Alert(Alert.AlertType.INFORMATION,
                "Record deleted!",
                ButtonType.OK);
            Optional<ButtonType> buttonType = alert.showAndWait();

```

```

    }
}
try {
    eqp_table.getItems().clear();
    initialize();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}

```

```

public void img_back(MouseEvent event) throws IOException {

```

```

    URL resource = this.getClass().getResource("/View/AceHomeView.fxml");
    Parent root = FXMLLoader.load(resource);
    Scene scene = new Scene(root);
    Stage primaryStage = (Stage) this.eqp_root.getScene().getWindow();
    primaryStage.setScene(scene);

```

```

    TranslateTransition tt = new TranslateTransition(Duration.millis(350), scene.getRoot());
    tt.setFromX(-scene.getWidth());
    tt.setToX(0);
    tt.play();
}

```

```

public void playMouseEnterAnimation(MouseEvent event) {

```

```

    if (event.getSource() instanceof ImageView) {
        ImageView icon = (ImageView) event.getSource();

        ScaleTransition scaleT = new ScaleTransition(Duration.millis(200), icon);
        scaleT.setToX(1.2);
        scaleT.setToY(1.2);
        scaleT.play();

```

```

        DropShadow glow = new DropShadow();
        glow.setColor(Color.YELLOW);

```

```
        glow.setWidth(20);
        glow.setHeight(20);
        glow.setRadius(20);
        icon.setEffect(glow);
    }
}
```

3.10 MembershipView which shows the List of Registered Members

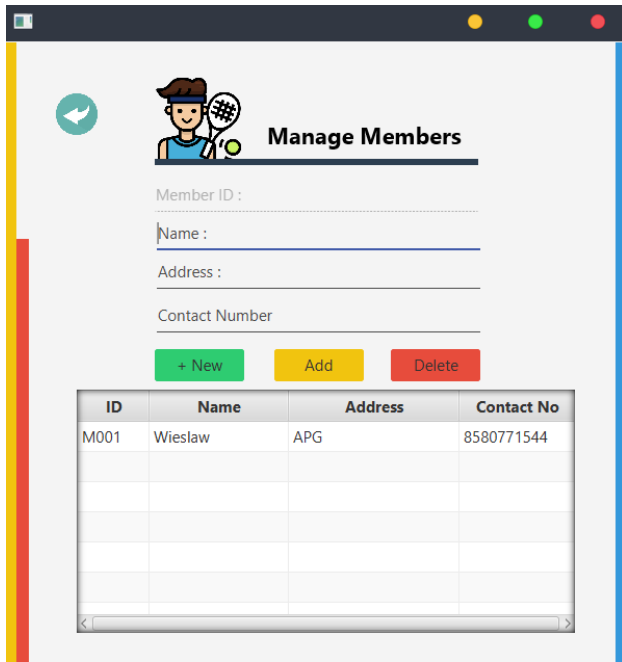


Figure 3.12 Screenshot of Membership View

3.10.1 MembershipView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?import com.jfoenix.controls.*?>
```

```
<?import javafx.scene.control.*?>
```

```
<?import javafx.scene.effect.InnerShadow?>
```

```
<?import javafx.scene.image.Image?>
```

```
<?import javafx.scene.image.ImageView?>
```

```
<?import javafx.scene.layout.*?>
```

```
<?import javafx.scene.text.Font?>
```

```
<?import javafx.scene.text.Text?>
```

```
<AnchorPane xmlns:fx="http://javafx.com/fxml/1" fx:id="root" prefHeight="400.0" prefWidth="600.0"
    xmlns="http://javafx.com/javafx/10.0.1" fx:controller="Controller.MembershipController">
```

```
<children>
```

```
<Pane layoutY="1.0" prefHeight="600.0" prefWidth="12.0" style="-fx-background-color: #f1c40f;"/>
```

```
<Pane layoutX="12.0" layoutY="190.0" prefHeight="411.0" prefWidth="12.0"
    style="-fx-background-color: #e74c3c;"/>
```

```
<Pane layoutX="588.0" layoutY="1.0" prefHeight="600.0" prefWidth="12.0" style="-fx-background-
color: #3498db;"/>
```

```

<ImageView fitHeight="86.0" fitWidth="92.0" layoutX="145.0" layoutY="32.0"
pickOnBounds="true"
    preserveRatio="true">
    <image>
        <Image url="@../Images/membership.png"/>
    </image>
</ImageView>
<Text layoutX="253.0" layoutY="99.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Manage
Members">
    <font>
        <Font name="System Bold" size="22.0"/>
    </font>
</Text>
<JFXTextField fx:id="mem_id" layoutX="145.0" layoutY="133.0" prefHeight="25.0"
prefWidth="311.0"
    promptText="Member ID :"/>
<JFXTextField fx:id="mem_nme" layoutX="147.0" layoutY="170.0" prefHeight="25.0"
prefWidth="311.0"
    promptText="Name :"/>
<JFXTextField fx:id="mem_addss" layoutX="147.0" layoutY="207.0" prefHeight="25.0"
prefWidth="311.0"
    promptText="Address :"/>
<JFXTextField fx:id="mem_num" layoutX="147.0" layoutY="249.0" prefHeight="25.0"
prefWidth="311.0"
    promptText="Contact Number"/>
<JFXButton fx:id="btn_new" layoutX="145.0" layoutY="296.0" onAction="#btn_new"
prefHeight="25.0"
    prefWidth="86.0" style="-fx-background-color: #2ecc71;" text="+ New"/>
<JFXButton fx:id="btn_add" layoutX="260.0" layoutY="296.0" onAction="#btn_add"
prefHeight="25.0"
    prefWidth="86.0" style="-fx-background-color: #f1c40f;" text="Add"/>
<JFXButton layoutX="372.0" layoutY="296.0" onAction="#btn_dtl" prefHeight="25.0"
prefWidth="86.0"
    style="-fx-background-color: #e74c3c;" text="Delete" textAlignment="CENTER"/>

```

```

<TableView fx:id="mem_tbl" layoutX="70.0" layoutY="335.0" prefHeight="234.0"
prefWidth="479.0">
    <columns>
        <TableColumn fx:id="mem_id_col" prefWidth="69.0" text="ID"/>
        <TableColumn fx:id="mem_name_col" prefWidth="134.0" text="Name"/>
        <TableColumn fx:id="mem_add_col" prefWidth="164.0" text="Address"/>
        <TableColumn fx:id="mem_no_col" prefWidth="111.0" text="Contact No"/>
    </columns>
    <effect>
        <InnerShadow/>
    </effect>
</TableView>
<ImageView fx:id="img_bk" fitHeight="40.0" fitWidth="40.0" layoutX="50.0" layoutY="50.0"
    onMouseClicked="#img_back" onMouseEntered="#playMouseEnterAnimation"
pickOnBounds="true"
    preserveRatio="true">
    <image>
        <Image url="@../Images/back.png"/>
    </image>
</ImageView>
<Label alignment="CENTER" layoutX="171.0" layoutY="569.0" prefHeight="17.0"
prefWidth="264.0"
    textAlignment="CENTER" textFill="#e74c3c"/>
<Pane layoutX="145.0" layoutY="113.0" prefHeight="6.0" prefWidth="311.0"
    style="-fx-background-color: #2c3e50;"/>
</children>

</AnchorPane>

```

3.10.2 MembershipController.java

```

package Controller;

import Model.MembershipModel;
import com.jfoenix.controls.JFXButton;

```

```

import com.jfoenix.controls.JFXTextField;
import db.DB;
import db.DBConnection;
import javafx.animation.ScaleTransition;
import javafx.animation.TranslateTransition;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.effect.DropShadow;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.util.Duration;

import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Optional;

public class MembershipController {

```

```

public JFXTextField mem_id;
public JFXTextField mem_nme;
public JFXTextField mem_addss;
public JFXTextField mem_num;
public TableView<MembershipModel> mem_tbl;
public ImageView img_bk;
public AnchorPane root;
public JFXButton btn_new;
public JFXButton btn_add;

//JDBC
private Connection connection;
private PreparedStatement selectall;
private PreparedStatement newIdQuery;
private PreparedStatement addToTable;
private PreparedStatement updateQuarary;
private PreparedStatement deleteQuarary;
private PreparedStatement slectmemID;

public void initialize() throws ClassNotFoundException {
    //disable id field
    mem_id.setDisable(true);
    //set colman
    mem_tbl.getColumns().get(0).setCellValueFactory(new PropertyValueFactory<>("id"));
    mem_tbl.getColumns().get(1).setCellValueFactory(new PropertyValueFactory<>("name"));
    mem_tbl.getColumns().get(2).setCellValueFactory(new PropertyValueFactory<>("address"));
    mem_tbl.getColumns().get(3).setCellValueFactory(new PropertyValueFactory<>("contact"));
    Class.forName("com.mysql.jdbc.Driver");

    try {
        connection = DBConnection.getInstance().getConnection();
        selectall = connection.prepareStatement("SELECT * from membershipdetail");
        slectmemID = connection.prepareStatement("select * from membershipdetail where id=?");
        newIdQuery = connection.prepareStatement("SELECT id FROM membershipdetail");
        addToTable = connection.prepareStatement("INSERT INTO membershipdetail values(?,?,?,?)");
    }
}

```

```

        updateQuarary = connection.prepareStatement("UPDATE membershipdetail SET name=? ,
address=? , contact=? where id=?");
        ObservableList<MembershipModel> members = mem_tbl.getItems();
        ResultSet rst = selectall.executeQuery();
        while (rst.next()) {
            System.out.println("load");
            members.add(new MembershipModel(
                rst.getString(1),
                rst.getString(2),
                rst.getString(3),
                rst.getString(4)
            ));
        }
        mem_tbl.setItems(members);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

        mem_tbl.getSelectionModel().selectedItemProperty().addListener(new
ChangeListener<MembershipModel>() {
            @Override
            public void changed(ObservableValue<? extends MembershipModel> observable,
MembershipModel oldValue, MembershipModel newValue) {
                MembershipModel selectedItem = mem_tbl.getSelectionModel().getSelectedItem();
                try {
                    connection = null;
                    try {
                        slectmemID.setString(1, selectedItem.getId());
                        ResultSet rst = slectmemID.executeQuery();
                        if (rst.next()) {
                            mem_id.setText(rst.getString(1));
                            mem_nme.setText(rst.getString(2));
                            mem_addss.setText(rst.getString(3));
                            mem_num.setText(rst.getString(4));
                            mem_id.setDisable(true);
                        }
                    }
                }
            }
        });
    }
}

```

```

        btn_add.setText("Update");
    }
} catch (SQLException e) {
    e.printStackTrace();
}
} catch (NullPointerException n) {
    return;
}
}
});
}

```

//button new action

```

public void btn_new(ActionEvent actionEvent) throws SQLException {
    //txt_valid.setText("");
    mem_nme.clear();
    mem_addss.clear();
    mem_num.clear();
    btn_add.setText("Add");
    mem_id.setDisable(false);

```

```

    ResultSet rst = newIdQuery.executeQuery();

```

```

    String ids = null;

```

```

    int maxId = 0;

```

```

    while (rst.next()) {

```

```

        ids = rst.getString(1);

```

```

        int id = Integer.parseInt(ids.replace("M", ""));

```

```

        if (id > maxId) {

```

```

            maxId = id;

```

```

        }

```

```

    }

```

```

    maxId = maxId + 1;

```

```

String id = "";
if (maxId < 10) {
    id = "M00" + maxId;
} else if (maxId < 100) {
    id = "M0" + maxId;
} else {
    id = "M" + maxId;
}
mem_id.setText(id);
}

//button add action
public void btn_add(ActionEvent actionEvent) throws SQLException {
    ObservableList<MembershipModel> members = FXCollections.observableList(DB.members);
    //is empty
    if (mem_id.getText().isEmpty() ||
        mem_nme.getText().isEmpty() ||
        mem_addss.getText().isEmpty() ||
        mem_num.getText().isEmpty()) {
        Alert alert = new Alert(Alert.AlertType.ERROR,
            "Please fill your details.",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
        return;
    }
    //reg-ex
    if (!(mem_nme.getText().matches("^\\b([A-Za-z.]\\s?)+$") &&
mem_addss.getText().matches("^\\b[A-Za-z0-9/\\s]+.$") && mem_num.getText().matches("\\d{10}"))) {
        new Alert(Alert.AlertType.ERROR, "Entered detail Invalid").show();
        return;
    }

    //save & update
    if (btn_add.getText().equals("Add")) {
        addToTable.setString(1, mem_id.getText());
    }
}

```



```

addToTable.setString(2, mem_nme.getText());
addToTable.setString(3, mem_addss.getText());
addToTable.setString(4, mem_num.getText());
int affectedRows = addToTable.executeUpdate();

if (affectedRows > 0) {
    System.out.println("Data load successful");
} else {
    System.out.println("Something went wrong");
}
} else {
    if (btn_add.getText().equals("Update")) {
        for (int i = 0; i < members.size(); i++) {
            if (mem_id.getText().equals(members.get(i).getId())) {
                try {
                    updateQuarary.setString(1, mem_nme.getText());
                    updateQuarary.setString(2, mem_addss.getText());
                    updateQuarary.setString(3, mem_num.getText());
                    updateQuarary.setString(4, mem_id.getText());
                    int affected = updateQuarary.executeUpdate();

                    if (affected > 0) {
                        Alert alert = new Alert(Alert.AlertType.INFORMATION,
                            "Record updated!!!",
                            ButtonType.OK);
                        Optional<ButtonType> buttonType = alert.showAndWait();
                    } else {
                        Alert alert = new Alert(Alert.AlertType.ERROR,
                            "Update error!",
                            ButtonType.OK);
                        Optional<ButtonType> buttonType = alert.showAndWait();
                    }
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

```

        }
    }
}
mem_tbl.setItems(members);
}
try {
    mem_tbl.getItems().clear();
    initialize();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}

//btn add delete
public void btn_dtl(ActionEvent actionEvent) throws SQLException {
    MembershipModel selectedItem = mem_tbl.getSelectionModel().getSelectedItem();
    if (mem_tbl.getSelectionModel().isEmpty()) {
        Alert alert = new Alert(Alert.AlertType.ERROR,
            "Please select a member.",
            ButtonType.OK);
        Optional<ButtonType> buttonType = alert.showAndWait();
        return;
    } else {
        deleteQuarary = connection.prepareStatement("DELETE from memberdetail where id=?");
        deleteQuarary.setString(1, selectedItem.getId());

        int affected = deleteQuarary.executeUpdate();
        if (affected > 0) {
            Alert alert = new Alert(Alert.AlertType.INFORMATION,
                "Record deleted!!",
                ButtonType.OK);
            Optional<ButtonType> buttonType = alert.showAndWait();
        }
    }
}
try {

```

```

        mem_tbl.getItems().clear();
        initialize();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}

```

```

public void playMouseEnterAnimation(MouseEvent event) {
    if (event.getSource() instanceof ImageView) {
        ImageView icon = (ImageView) event.getSource();

        ScaleTransition scaleT = new ScaleTransition(Duration.millis(200), icon);
        scaleT.setToX(1.2);
        scaleT.setToY(1.2);
        scaleT.play();

        DropShadow glow = new DropShadow();
        glow.setColor(Color.YELLOW);
        glow.setWidth(20);
        glow.setHeight(20);
        glow.setRadius(20);
        icon.setEffect(glow);
    }
}

```

```

public void img_back(MouseEvent event) throws IOException {
    URL resource = this.getClass().getResource("/View/AceHomeView.fxml");
    Parent root = FXMLLoader.load(resource);
    Scene scene = new Scene(root);
    Stage primaryStage = (Stage) this.root.getScene().getWindow();
    primaryStage.setScene(scene);

    TranslateTransition tt = new TranslateTransition(Duration.millis(350), scene.getRoot());
    tt.setFromX(-scene.getWidth());
    tt.setToX(0);
}

```

```
        tt.play();
    }
}
```

3.10.3 MembershipModel.java

```
package Model;
```

```
public class MembershipModel {
```

```
    private String id;
    private String name;
    private String address;
    private String contact;
```

```
    public MembershipModel(String id, String name, String address, String contact) {
        this.id = id;
        this.name = name;
        this.address = address;
        this.contact = contact;
    }
```

```
    public String getId() {
        return id;
    }
```

```
    public void setId(String id) {
        this.id = id;
    }
```

```
    public String getName() {
        return name;
    }
```

```
    public void setName(String name) {
        this.name = name;
```

```
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getContact() {
    return contact;
}

public void setContact(String contact) {
    this.contact = contact;
}
}
```

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1 The Flow of Control and Information

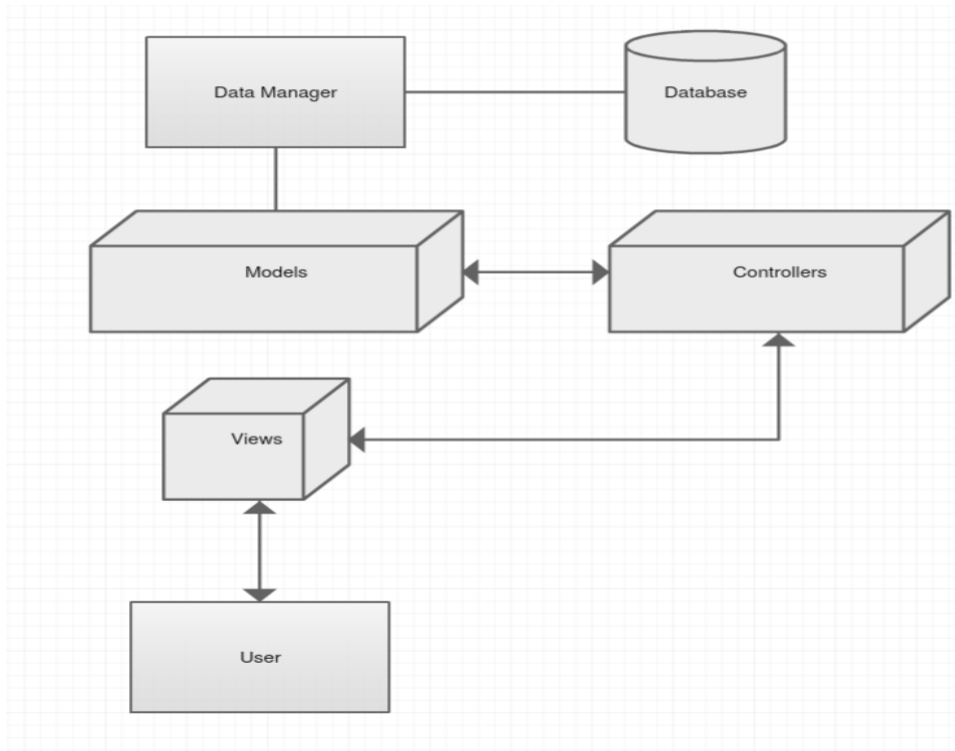


Figure 4.1 DFD for Model-View-Controller Pattern used in the project.

An example of efficient use of the concept is addition of a new member:

The left screenshot shows the 'Manage Members' form with the following data:

ID	Name	Address	Contact No
M001	Wieslaw	APG	8580771544
M002	Vicky	Kalkaji	8590123456

The right screenshot shows the 'Manage Members' form with the following data:

ID	Name	Address	Contact No
M001	Wieslaw	APG	8580771544
M002	Vicky	Kalkaji	8590123456

For user input **into the Database**

- **Users Input** Leads us to Login Screen (**View**) and credentials are authenticated then Membership Screen (**View**) is opened.
- Manipulations such as logging in, +New, Add (**Buttons on Views**) of Login Screen upto Membersip Screen are programmed in LoginScreenController.java, AceHomeController.java and MembershipController.java (**Controllers**). Clicking on **+ New** autogenerates a membership ID of M002.
- After Member's (**Vicky**) info is recorded into the table MembershipModel.java (**Model**) is called up as it control the arrays used to store information in a table format. The information is the transferred via phpMyAdmin (**Database Manager**) into MongoDB (**Database**).
- The two edged arrows between Model, Controller and View define data flow whereby the Controller file is used to interlink Views to Models. The “middle-man” component are Controllers.

One of the underlying reasons for this software is to effectively utilise performance of resources at the Academy, which in turn helps grow the academy. The headache of not accessing a certain resource for a member without knowledge of even its whereabouts has now been reduced. This helps maintain a direct connection between the members and the Academy. The features to Add and remove (Delete), Search, Borrow, Return allows priviledged users with other logs to add and easily search for equipment as they deem fit.

- **Programming.** Through design of the system, I learnt more about object-oriented languages, such as C++, Java. **Learning about Specific technologies used in Java especially made it worthwhile.**
- **Project and Technical Skills.** Through plenty researching and analysing of the different styles of coding, it has taught me to appreciate the process more.
- **Remotely accessing the Database:** Though it is still in beta testing, it has proven to be used especially in situations where an administration is not at their work station.
- **Technical Support.** Providing online support has been the norm since the period of training itself was online.

4.3 Interaction with experienced people in the field

I also had the opportunity to network with potential future employers and gain insight into the types of employees they look for. This made me realize the greatest value of Internship which is providing a unique and exciting experience that is unparalleled in the classroom and to coordinate job experience with academic.

4.4 Learning IT area of interest

Internship offered me the opportunity to essentially focus more on Specialising in a certain field in order to gain valuable work experience in my areas of interest. Communicating with some of the best developers, designers and managers in information technology department, solving problems that you can never face in a school environment, bonding with other interns that make up a very diverse demographic.

CHAPTER 5

5.1 Conclusions

The main objective of the project is to develop a Logging Application. I took a wide range of scenarios and possibilities that might happen at a sports club where I came to know more about their need. I made a detailed research in that path to cover the loop holes that the existing system is facing and to eradicate them in our application. In the process of research I came to know about the latest technologies and different algorithms.

The portability of the application has been achieved by using some of the latest JAVA technologies. I implemented these functionalities using JAVA API's. I had gone through core and security concepts of java (JSSE, JCA) packages and for developing GUI I had to implement JAVAFX.

As a result, the product has been successfully developed in terms of extendability, portability, and maintainability and tested in order to meet all requirements that are:

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility
- Reliable and Safe Environment to Issue and Collect equipment

5.2 Future Scope

With the knowledge I have gained by developing this application, I am confident that in the future I can make the application more effective by adding these services.

- Extending this application by providing Authorisation service.
- Increasing the effectiveness of the application by providing Voice Commanding.
- Extending it to Web Support.
- Implementing Encryption algorithms e.g AES and Integrity algorithms e.g MD5.
- Extending it for Remote Access.

REFERENCES

- ✓ **MySQL**. (2014, September). MySQL Tutorial. Retrieved June 23, 2016, from `
https://www.ntu.edu.sg/home/ehchua/programming/sql/MySQL_Beginner.html.
- ✓ <https://stackoverflow.com/> for Problem Solving
- ✓ **Database Programming** with JDBC and Java by O'Reilly
- ✓ **Head First Java** 2nd Edition
- ✓ *(80 hours video content)*

<https://www.udemy.com/course/java-the-complete-java-developer-course/>
- ✓ <https://docs.oracle.com/javase/tutorial/>
- ✓ <http://javatpoint.com/java-tutorial>
- ✓ <http://www.tutorialspoint.com/mysql/>
- ✓ [httpd.pache.org/docs/2.0/misc/tutorials.html](http://d.pache.org/docs/2.0/misc/tutorials.html)