International University, HCMC National University

School of Computer Science and Engineering

A blue and white logo

Description automatically generated**PROPOSAL GROUP 5**

**Online Art Gallery Database**

|  |  |
| --- | --- |
| Nguyễn Hữu Minh Nguyễn | ITITIU21089 |
| Tô Anh Khoa | **ITITIU21225** |
| Nguyễn Duy Phúc | **ITDSIU21030** |
| Trần Nguyễn Nam Anh | **ITITIU20162** |
| Phạm Đăng Quang | **ITCSIU21147** |
| Phạm Minh Nguyên | **ITITUN21033** |

Course name: Principles of Database Management

Advisor: Assoc. Prof. Dr. Nguyễn Thị Thúy Loan

TABLES OF CONTENTS

[I. Introduction 4](#_Toc155280652)

[1. Abstract 4](#_Toc155280653)

[2. Goal 4](#_Toc155280654)

[3. The technique and & tools usedT 4](#_Toc155280655)

[II. Implementation plan & Contributions 5](#_Toc155280656)

[III. PROJECT ANALYST 7](#_Toc155280657)

[1. Database Design 7](#_Toc155280658)

[1.1 Entity Relationship Diagram 7](#_Toc155280659)

[1.2 Database Diagram 8](#_Toc155280660)

[1.3 Normalization 8](#_Toc155280661)

[1.4 Schemas and Tables 9](#_Toc155280662)

[2. Database creation 11](#_Toc155280663)

[3. Data insertion 16](#_Toc155280664)

[4.1 Update users information 17](#_Toc155280665)

[4.2 Get account information 18](#_Toc155280666)

[4.3 Change password 18](#_Toc155280667)

[4.4 Get items in exhibition 18](#_Toc155280668)

[4.5 Get list of items from a gallery which were not already in an exhibition 18](#_Toc155280669)

[4.6 Remove items from an exhibition 18](#_Toc155280670)

[Finding exhibition within next 14 days: 18](#_Toc155280671)

[4.8 Find items in an exhibition. 19](#_Toc155280672)

[4.9 List all artworks belong to a gallery 19](#_Toc155280673)

[4.10 Finding information of an order 19](#_Toc155280674)

[5. Application: 20](#_Toc155280675)

[5.1 Overview 20](#_Toc155280676)

[5.2 Hosting tools. 21](#_Toc155280677)

[5.3 Setting up Java connection 22](#_Toc155280678)

[5.4 Running the application 23](#_Toc155280679)

[6. Application Demo: 24](#_Toc155280680)

[6.1 Login page 24](#_Toc155280681)

[6.2 Admin 25](#_Toc155280682)

[6.3 Users 29](#_Toc155280683)

[6.4 Gallery 34](#_Toc155280684)

[6.5 Shipper 36](#_Toc155280685)

[IV. Conclusion: 37](#_Toc155280686)

[1. Achieved Goals 37](#_Toc155280687)

[2. Future work 37](#_Toc155280688)

[3. Conclusion 37](#_Toc155280689)

# [Introduction](https://docs.google.com/document/d/1wI0ZiNaUhDhZFthKvX-7Ra6g1SQUBww7zeE0nm_uC6A/edit?fbclid=IwAR1E8dJQ5sGZaBGTVfZ8APUBe2MatPiomLOG2OWEcxFda497k4-j82mxN_w#heading=h.qf23fpoxtki)

## Abstract

The Online Art Gallery Database is an innovative platform that revolutionizes the way artists and art lovers connect. It provides a virtual space for artists to showcase their work and for art enthusiasts to discover and purchase their favorite pieces. With its user-friendly interface and limitless potential for creativity, the Online Art Gallery Database is an excellent option for anyone interested in SQL database projects.

This platform streamlines the process of finding, displaying, and buying art. Artists can easily upload their artwork with detailed descriptions and high-quality images, while art lovers can search through a vast collection using various filters. The Online Art Gallery Database not only helps artists gain exposure, but also provides art enthusiasts with a seamless and enjoyable experience, making it a valuable tool for both groups. With secure payment options and efficient shipping arrangements, the Online Art Gallery Database ensures a smooth and hassle-free transaction. Overall, this project offers an excellent opportunity to showcase your skills and creativity while making a significant impact in the art community.

## Goal

To make it possible for inexperienced readers to understand the problem the group wants to convey.

To learn and master SQL.

To improve the ability in data research and data mining

## The technique and & tools used

Main tool: Database & Server : MySQL, XAMPP.

GUI: JAVA (Netbeans Jpanel Form)

ERD: Draw.io

Database Diagram: Drawsql.com

Version control: Github.com

# Implementation plan & Contributions

|  |  |  |  |
| --- | --- | --- | --- |
| Time | No. | Tasks | Member |
| **Week 1 - 4**  **Proposal** | 1 | Choosing topic | All members |
| 2 | Write an introduction and general information  about the topic. | Phúc/ Khoa |
| 3 | Write a timeline for the proposal and edit the  Word file. | Anh/ M.Nguyễn |
| 4 | Check for mistakes to complete the proposal. | All members |
| **Week 4-5**  **Work with data** | 7 | Do research, find data sources | All members |
| 8 | Classify information, find commonalities and influences | Quang/Nguyên |
| 9 | Analyze information,  filter information | M.Nguyễn/Phúc |
| 10 | Attend pair meetings for discussion | All members |
| **Week 5**  **Apply SQL and Java with database** | 11 | Use SQL to work with filtered data | All members |
| 12 | Trying to apply java code |
| 13 | Running code/debug |
| 14 | Attend pair meetings for discussion |
| **Week 6**  **Complete the project and test.** | 15 | Note down the main results after test  Identify key ideas from the database project | All members |
| 16 | Attend pair meetings for discussion |
| **Week 7 & 8**  **Report Preparation and overview** | 17 | Prepare for final report | All members |
| 18 | All slide will be completed |
| **Week 9**  **Final recap** | 19 | Prepare for presentation | All members |
| 20 | Attend pair meeting for discussion |

**Table 1 : Timeline planning and task**

|  |  |  |
| --- | --- | --- |
| **Name** | **Responsibility** | **Contributions** |
| Nguyễn Hữu Minh Nguyễn | Organization, Version Controller | 16.6 |
| Nguyễn Duy Phúc | Interface Developer | 16.6 |
| Tô Anh Khoa | Database Developer | 16.6 |
| Trần Nguyễn Nam Anh | Database Developer | 16.6 |
| Phạm Đăng Quang | Database Developer | 16.6 |
| Phạm Minh Nguyên | Database Developer | 16.6 |

**Table 2 : Contributions**

# PROJECT ANALYST

# Database Design

### 1.1 Entity Relationship Diagram

**A diagram of a diagram

Description automatically generated**

**Figure 1: ERD Diagram for Online Gallery Database**

### 1.2 Database Diagram

**A screenshot of a computer

Description automatically generated**

**Figure 2: Database Diagram for Online Gallery Database**

### 1.3 Normalization

Our database has achieved Boyce-Codd NF which is illustrated below:

|  |  |
| --- | --- |
| Normal Form | Description |
| 1 N.F | Each table cell should contain a single value.  Ever table has primary key. |
| 2 N.F | Each non-key attribute in the table must be dependent on the entire primary key. |
| 3 N.F | Every non-key attribute is nontransitively dependent on every key |
| BC N.F. | Each attribute in the table must depend on the key, the whole key, and nothing but the key |

**Table 3. Normal form achievement**

### 1.4 Schemas and Tables

**Table 4. An overview of all database entities and their attributes**

|  |  |
| --- | --- |
| **Table** | **attributes** |
| Artworks | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **TITLE** varchar(255) NOT NULL  **ABOUT** text NOT NULL  **CREATED\_YEAR** int NOT NULL  **IS\_FOR\_SALE** Boolean NOT NULL  **PRICE** decimal(8, 2) NOT NULL  **GALLERY\_ID** int UNSIGNED NOT NULL  **ARTIST\_ID** int UNSIGNED NOT NULL |
| Shippers | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **SHIPPER\_NAME** varchar(255) NOT NULL  **PHONE** varchar(20) NOT NULL  **ACCOUNT\_ID** int unsigned NOT NULL |
| Exhibitions | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **EXHIBITION\_NAME** varchar(255) NOT NULL  **DETAIL** text NOT NULL  **START\_DATE** date NOT NULL  **END\_DATE** date NOT NULL  **GALLERY\_ID** int unsigned NOT NULL |
| Accounts | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **USERNAME** varchar(255) NOT NULL UNIQUE  **PASSWORD** varchar(255) NOT NULL  **ROLE\_ID** int unsigned NOT NULL |
| Roles | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **ROLE\_NAME** varchar(20) NOT NULL UNIQUE |
| Artists | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **FULLNAME** varchar(255) NOT NULL  **ABOUT** text NOT NULL  **BIRTH\_YEAR** year NOT NULL  **DEATH\_YEAR** year NOT NULL |
| Orders | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **USER\_ID** int unsigned NOT NULL  **SHIPPER\_ID** int unsigned NOT NULL  **ARTWORK\_ID** int unsigned NOT NULL  **ORDER\_TIME** datetime NOT NULL |
| Galleries | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **GALLERY\_NAME** varchar(255) NOT NULL  **ADDRESS TEXT** NOT NULL  **ACCOUNT\_ID** int unsigned NOT NULL |
| Interactions | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **USER\_ID** int unsigned NOT NULL  **ARTWORK\_ID** int unsigned NOT NULL  **INTERACTION\_TIME** datetime NOT NULL |
| Users | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **FIRST\_NAME** varchar(20) NOT NULL  **LAST\_NAME** varchar(20) NOT NULL  **PHONE** varchar(20) NOT NULL UNIQUE  **ADDRESS** text NOT NULL  **ACCOUNT\_ID** int unsigned NOT NULL |
| Items | **ID** int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY  **EXHIBITION\_ID** int unsigned NOT NULL  **ARTWORK\_ID** int unsigned NOT NULL |

From the above table, there are some requirements go with the table characteristics:

1. The accounts table contains important information and can only be seen or altered by authorized account which is dependent on the role id with the role table.
2. Artworks, Shippers, Exhibitions, Artists, Galleries, Interactions and Items tables contain information equivalent to the art in the databases and can be searched by user.
3. Users and Orders tables are responsible for primary information of the customer.

## Database creation

Database:

A black and white text

Description automatically generated

Roles Table: A close-up of a text

Description automatically generated

Accounts Table:

A close up of text

Description automatically generated

Artists Table:

A computer code with text

Description automatically generated with medium confidence

Shippers Table:

A close-up of text

Description automatically generated

Users Table:

A white background with blue text

Description automatically generated

Galleries Table:

A close up of text

Description automatically generated

Artworks Table:

A computer code with text

Description automatically generated

Interactions Table:

A close-up of a computer code

Description automatically generated

Orders Table:

A close-up of text

Description automatically generated

Exhibitions Table:

A white background with text

Description automatically generated

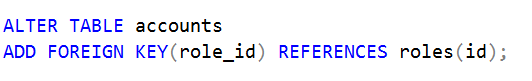
Item Table:

A close-up of a sign

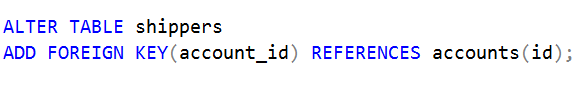
Description automatically generated

Alter Table – Foreign Key:

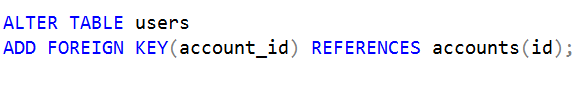
Account Alter Table – Foreign key:



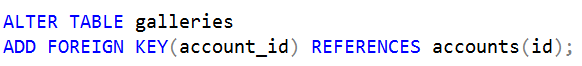
Shipper Alter Table – Foreign key:



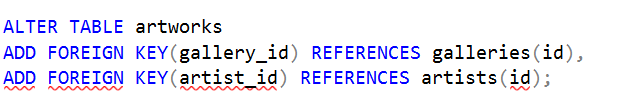
User Alter Table – Foreign Key:



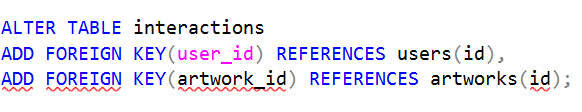
Gallery Alter Table – Foreign key:



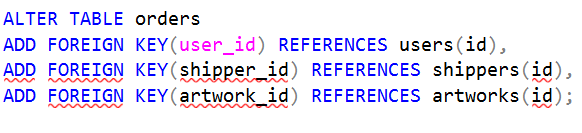
Artwork Alter Table – Foreign Key:



Interaction Alter Table – Foreign Key:



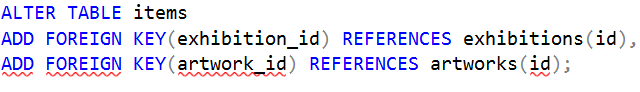
Order Alter Table – Foreign Key:



Exhibition Alter Table – Foreign Key:



Item Alter Table – Foreign Key:



## Data insertion

These data are to demonstrated to preview the data insert only, and the complete data can be found using the backup file attached to this report.

Roles table:A white background with black and white clouds

Description automatically generated

Accounts table:A white background with black and white clouds

Description automatically generated  
Artists table:

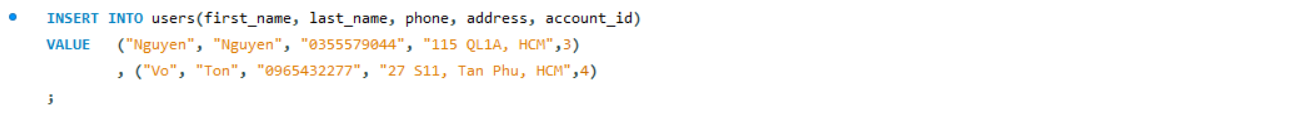
A close-up of a text

Description automatically generated

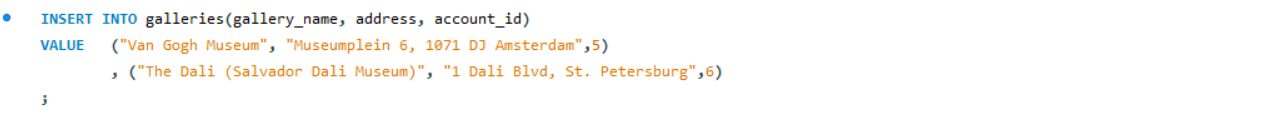
Shippers tables:



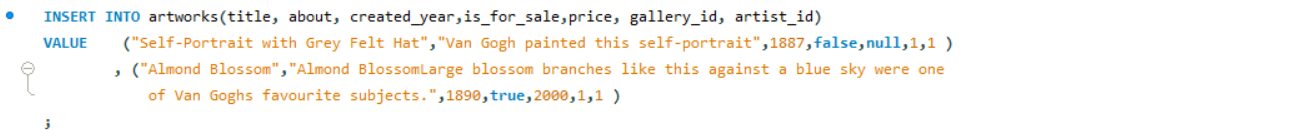
Users table:



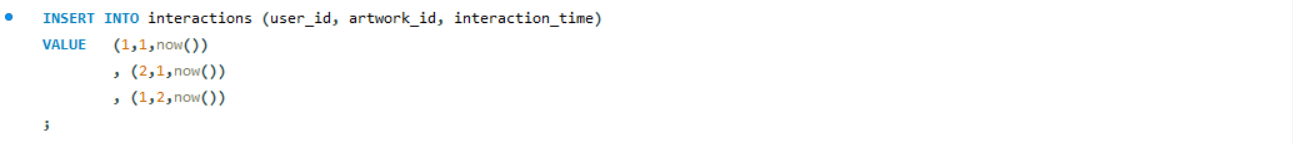
Galleries table:



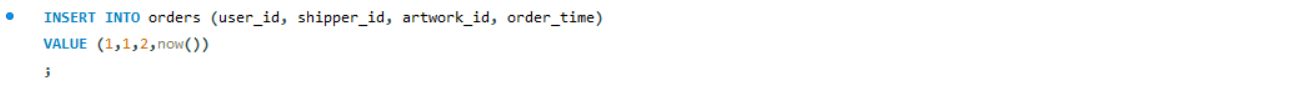
Artworks table:



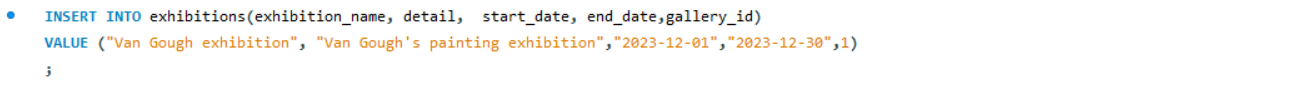
Interactions table:



Orders table:



Exhibitions table:



Items table:



1. **Data Queries:**

These are important queries used in the project with pseudo-data and it use. These data can be changed in used with Java GUI Input. The code is attached in folder “dataqueries”:

### 4.1 Update users information

UPDATE users

SET first\_name = ?, last\_name = ?, phone = ?, address = ?

WHERE id = userID;

### 4.2 Get account information

SELECT \* FROM accounts WHERE id = (SELECT account\_ID FROM users WHERE ID = ?)

### 4.3 Change password

UPDATE accounts SET password = ? WHERE id = (SELECT account\_ID FROM users WHERE ID = ?)

### 4.4 Get items in exhibition

SELECT \* FROM ( items INNER JOIN artworks on artworks.id =items.artwork\_id) WHERE items.exhibition\_id = 1;

### 4.5 Get list of items from a gallery which were not already in an exhibition

SELECT \* FROM artworks WHERE artworks.id NOT IN(SELECT artwork\_id FROM items WHERE exhibition\_id = 1);

### 4.6 Remove items from an exhibition

DELETE FROM items WHERE exhibition\_id = ? AND artwork\_id = ?;

### Finding exhibition within next 14 days:

A close-up of a computer screen

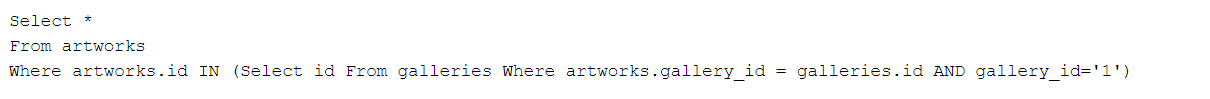
Description automatically generated

### 4.8 Find items in an exhibition.

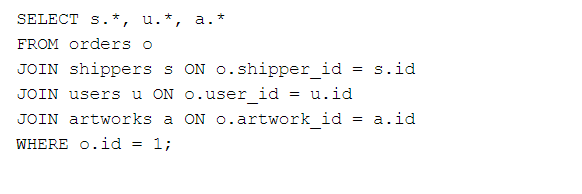
A white background with black text

Description automatically generated

### 4.9 List all artworks belong to a gallery



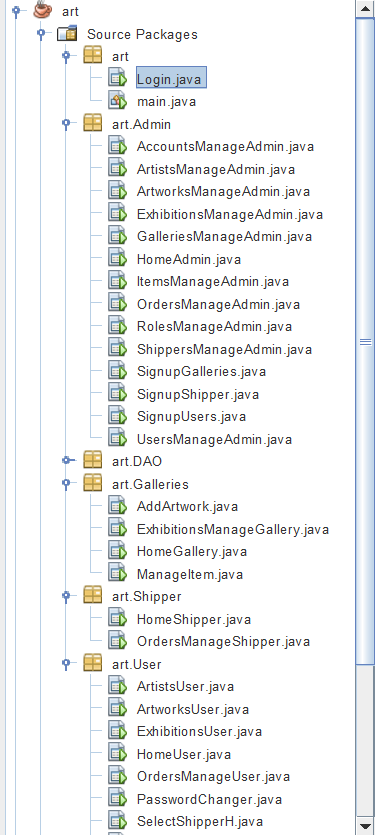
### 4.10 Finding information of an order



## Application:

### 5.1 Overview

In this project, our team decided to use Java programming language to develop an application GUI that can interact with the database.



**Firgure 3: Project file directory**

### 5.2 Hosting tools.

To connect the database with the application, a database host server is needed. We use XAMPP to set up a simple local host server with database.

A screenshot of a computer

Description automatically generated

**Figure 4: XAMPP Control Panel**

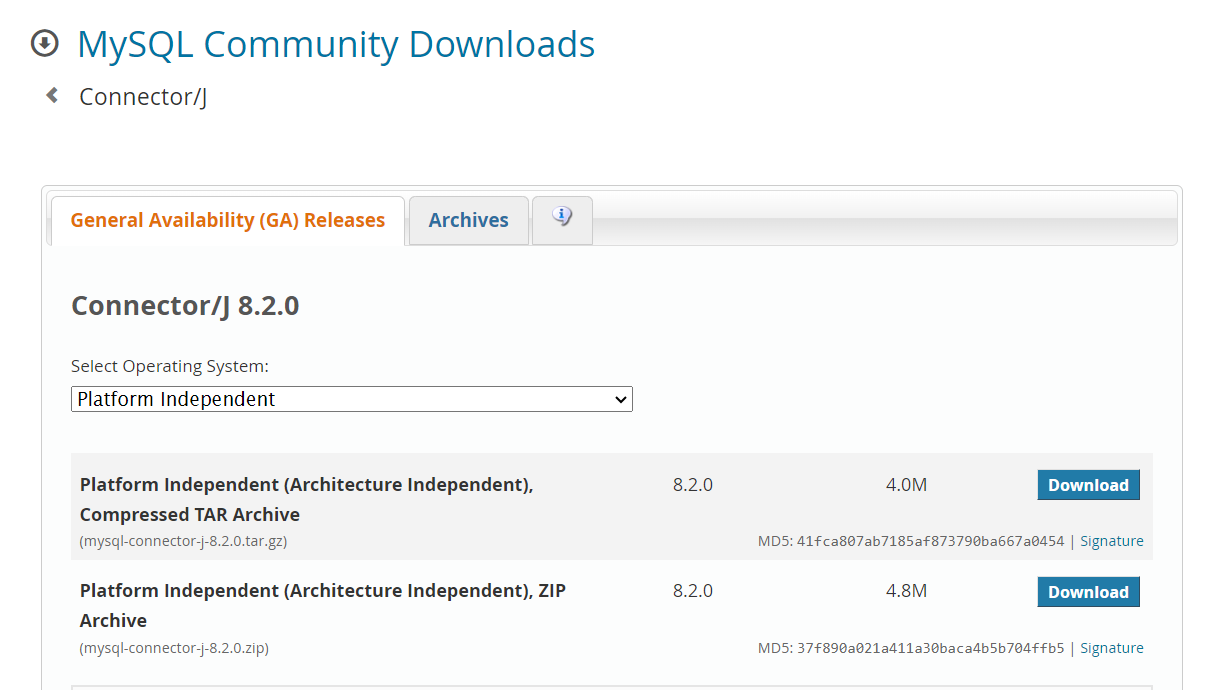
A screenshot of a computer

Description automatically generated

**Figure 5: phpMyAdmin – MySQL database comes with XAMPP**

### 5.3 Setting up Java connection

After creating artgallery database on phpMyAdmin – XAMPP local host server. We need to add MySQL Connector library for Java, which can be downloaded from MySQL official website: <https://dev.mysql.com/downloads/connector/j/5.1.html>



**Figure 6: Connector/J 8.2.0 Library**

After downloading, the library must be added to our project

A screenshot of a computer

Description automatically generated

**Figure 7: Adding the .jar library**

ConnectionProvider class was created to handle connection between our application and database using the library including above.

**A screenshot of a computer

Description automatically generated**

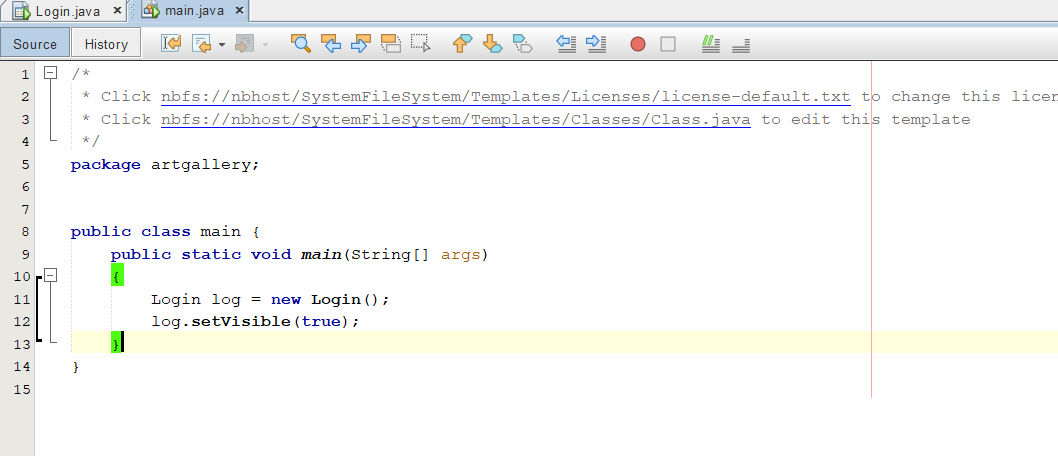
**Figure 9: ConnectionProvider.java class**

Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/artgallery?useSSL=false","root","");

The above ("jdbc:mysql://localhost:3306/artgallery?useSSL=false","root","") path can be change for different host configuration at your own machine.

### 5.4 Running the application

The application can be started by running main.java file in artgallery project



**Figure 10: main.java**

## Application Demo:

### 6.1 Login page

**A screenshot of a login form

Description automatically generated**

**Figure 11: Login page when run the project**

### 6.2 Admin

By login to admin account, Admin can create accounts with different roles from “users”

**A screenshot of a computer

Description automatically generated**

**Figure 12: Admin homepage allow viewing/editing each table data**

**A screenshot of a computer

Description automatically generated**

**Figure 13: After clicking to Accounts button**

**A screenshot of a computer

Description automatically generated**

**Figure 14: Added new data**

**A screenshot of a computer

Description automatically generated**

**Figure 15: After added**

All of the table in the schema can be controlled by the same way as above

A screenshot of a login form

Description automatically generated

**Figure 16: User sign up form**

**A screenshot of a login form

Description automatically generated**

**Figure 17: Gallery sign up form**

**A screenshot of a login form

Description automatically generated**

**Figure 18: Gallery sign up form**

### 6.3 Users

**A screenshot of a computer

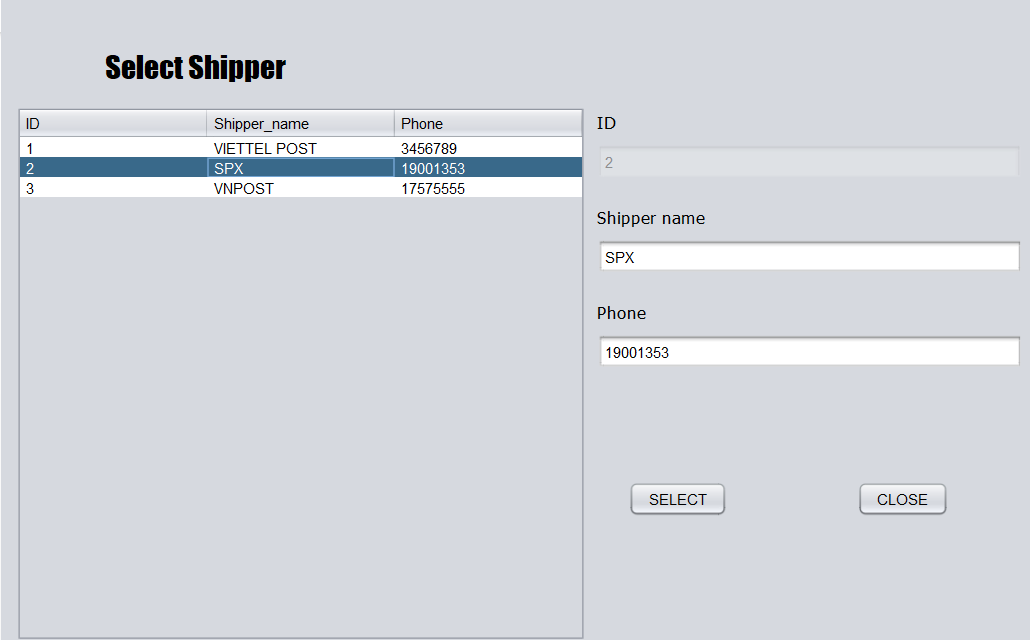
Description automatically generated**

**Figure 19: Users homepage**

**A screenshot of a computer

Description automatically generated**

**Figure 20: Browse artworks information**

****

**Figure 21: Select delivery options when buy**

**A screenshot of a computer

Description automatically generated**

**Figure 22: Reviewing orders information**

**A screenshot of a computer

Description automatically generated**

**Figure 23: Browse artists information**

**A screenshot of a computer

Description automatically generated**

**Figure 24: Browse artist’s artwork**

**A screenshot of a computer

Description automatically generated**

**Figure 25: Browse exhibition’s information**

**A screenshot of a computer

Description automatically generated**

**Figure 26: View list of artwork featured in exhibition.**

**A screen shot of a login form

Description automatically generated**

**Figure 27: Change current users information.**

### 6.4 Gallery

**A screenshot of a computer

Description automatically generated**

**Figure 28: Gallery homepage.**

**A screenshot of a computer

Description automatically generated**



**Figure 29: Add Artworks to the database.**

**A screenshot of a computer

Description automatically generated**

**Figure 30: Manage exhibition hold by the gallery.**

**A screenshot of a computer

Description automatically generated**

**Figure 31: Add and remove artworks from the exhibition.**

### 6.5 Shipper

**A screenshot of a computer

Description automatically generated**

**Figure 32: Shipper homepage.**

# Conclusion:

## Achieved Goals

After completing this project, we have successfully created a database project by utilizing our knowledge gained from PDM classes and online tutorials. We have developed a Java application that can connect and interact with a database from scratch. Our project can create a basic art gallery database that includes login, browsing, and buying artwork from different galleries. It also displays information about the galleries, artworks, artists, and exhibitions.

The project has taught us how to work as a team, develop a working plan, and conduct proper research to complete a project. Firstly, we researched the appropriate ways to complete the project. Secondly, we designed a database that achieved Boyce Codd normal form and was suitable for our needs. Thirdly, we created a database using MySQL. Lastly, we developed a GUI application using Java and connected it to the database.

## Future work

There are several ways in which we can improve this project. Firstly, we want to find a proper way to store and display images of the artwork, which we cannot display in the current project. Secondly, we want to improve the GUI and develop an online web application based on this project.

## Conclusion

This project simplifies the process of discovering, displaying, and purchasing art. After completing this project, we have gained knowledge about database management, including how to create ERD, database diagrams, normalization, and develop a Java application that can connect to the database.