Case Study: Virtual Art Gallery

Name: Maddaka Tejaswini

Batch: Python-1

Key Functionalities:

Artwork management :

The Virtual Art Gallery System aims to provide an immersive and interactive experience for art enthusiasts to explore, view, and appreciate a diverse collection of artworks online.

Personal Galleries: Enable users to create their virtual galleries and curate their collections.

Schema design:

Entities:

• Designing the schema for a Virtual Art Gallery involves creating a structured representation of the database that will store information about artworks, artists, users, galleries, and various relationships between them. Below is a schema design for a Virtual Art Gallery database:

• Entities and Attributes:

• Artwork

ArtworkID (Primary Key)

Title

Description

CreationDate

Medium

ImageURL (or any reference to the digital representation)

• Artist

ArtistID (Primary Key)

Name Biography

BirthDate

Nationality

Website

Contact Information

• User

UserID (Primary Key)

Username

Password

Email

First Name

Last Name

Date of Birth

Profile Picture

FavoriteArtworks (a list of references to ArtworkIDs)

• Gallery

GalleryID (Primary Key)

Name

Description

Location

Curator (Reference to ArtistID)

OpeningHours

• Relationships:

• Artwork - Artist (Many-to-One)

An artwork is created by one artist.

Artwork.ArtistID (Foreign Key) references Artist.ArtistID.

• User - Favorite Artwork (Many-to-Many)

A user can have many favorite artworks, and an artwork can be a favorite of multiple users.

User_Favorite_Artwork (junction table):

UserID (Foreign Key) references User.UserID.

ArtworkID (Foreign Key) references Artwork.ArtworkID.

• Artist - Gallery (One-to-Many)

An artist can be associated with multiple galleries, but a gallery can have only one curator (artist).
Gallery.ArtistID (Foreign Key) references Artist.ArtistID.

• Artwork - Gallery (Many-to-Many)

An artwork can be displayed in multiple galleries, and a gallery can have multiple artworks.

Artwork_Gallery (junction table):

ArtworkID (Foreign Key) references Artwork.ArtworkID.

GalleryID (Foreign Key) references Gallery.GalleryID.

Artist.py

```
class Artist:
    def __init__(self, artistId,
name,biography,birthDate,nationality,website,contactInformation):
        self.artistId = artistId
        self.name = name
        self.biography = biography
        self.birthDate = birthDate
        self.nationality = nationality
        self.website = website
        self.contactInformation = contactInformation

    #Setters

    def set_artistId(self,artistId):
        self.artistId = artistId
    def set_name(self,name):
        self.name = name
    def set_biography(self,biography):
        self.biography = biography
    def set_birthDate(self,birthDate):
        self.birthDate = birthDate
    def set_nationality(self,nationality):
        self.nationality = nationality
    def set_website(self,website):
        self.website = website
    def set_contactInformation(self,contactInformation):
        self.contactInformation = contactInformation

    #Getters

    def get_artistId(self):
        return self.artistId
    def get_name(self):
        return self.name
    def get_biography(self):
        return self.biography
    def get_birthDate(self):
        return self.birthDate
    def get_nationality(self):
        return self.nationality
    def get_website(self):
        return self.website
    def get_contactInformation(self):
        return self.contactInformation
```

artwork.py

```
class Artwork:
    def
__init__(self,artworkId,title,description,creationDate,medium,imageURL):
        self.artworkId = artworkId
        self.title = title
        self.description = description
        self.creationDate = creationDate
        self.medium = medium
        self.imageURL = imageURL

    #Setters

    def set_artworkId(self,artworkId):
        self.artworkId = artworkId
    def set_title(self,title):
        self.title = title
    def set_description(self,description):
        self.description = description
    def set_creationDate(self,creationDate):
        self.creationDate = creationDate
    def set_medium(self,medium):
        self.medium = medium
    def set_imageURL(self,imageURL):
        self.imageURL = imageURL

    #Getters

    def get_artworkId(self):
        return self.artworkId
    def get_title(self):
        return self.title
    def get_description(self):
        return self.description
    def get_creationDate(self):
        return self.creationDate
    def get_medium(self):
        return self.medium
    def get_imageURL(self):
        return self.imageURL
```

gallery.py

```
class Gallery:
    def
__init__(self,galleryId,name,description,location,curator,openingHours):
        self.galleryId = galleryId
        self.name = name
        self.description = description
        self.location = location
```

```
        self.curator = curator
        self.openingHours = openingHours
    #Setters

    def set_galleryId(self,galleryId):
        self.galleryId = galleryId
    def set_name(self,name):
        self.name = name
    def set_description(self,description):
        self.description = description
    def set_location(self,location):
        self.location = location
    def set_curator(self,curator):
        self.curator = curator
    def set_openingHours(self,openingHours):
        self.openingHours = openingHours

    #Getters


      def get_galleryId(self):
          return self.galleryId
      def get_name(self):
          return self.name
      def get_description(self):
          return self.description
      def get_location(self):
          return self.location
      def get_curator(self):
          return self.curator
      def get_openingHours(self):
          return self.openingHours
```

user.py

```
class User:
    def __init__(self,
userId,userName,password,email,firstName,lastName,dateOfBirth,profilePicture,f
avouriteArtworks):
        self.userId = userId
        self.userName = userName
        self.password = password
        self.email = email
        self.firstName = firstName
        self.lastName = lastName
        self.dateOfBirth = dateOfBirth
        self.profilePicture = profilePicture
        self.favouriteArtworks = favouriteArtworks

    #Setters
```

```
    def set_userId(self,userId):
        self.userId = userId
    def set_username(self,userName):
        self.userName = userName
    def set_password(self,password):
        self.password = password
    def set_email(self,email):
        self.email = email
    def set_firstName(self,firstName):
        self.firstName = firstName
    def set_lastName(self,lastName):
        self.lastName = lastName
    def set_dateOfBirth(self,dateOfBirth):
        self.dateOfBirth = dateOfBirth
    def set_profilePicture(self,profilePicture):
        self.profilePicture = profilePicture
    def set_favouriteArtworks(self,favouriteArtworks):
        self.favouriteArtworks = favouriteArtworks

    #Getters

    def get_userId(self):
        return self.userId
    def get_username(self):
        return self.userName
    def get_password(self):
        return self.password
    def get_email(self):
        return self.email
    def get_firstName(self):
        return self.firstName
    def get_lastName(self):
        return self.lastName
    def get_dateOfBirth(self):
        return self.dateOfBirth
    def get_profilePicture(self):
        return self.profilePicture
    def get_favouriteArtworks(self):
        return self.favouriteArtworks
```

Coding

Create the model/entity classes corresponding to the schema within package entity with variables declared private, constructors(default and parametrized) and getters,setters)

Service Provider Interface/Abstract class

Keep the interfaces and implementation classes in package dao

Create IVirtualArtGallery Interface/abstract class with the following methods

// Artwork Management

addArtwork();

parameters- Artwork object

return type Boolean

updateArtwork();

parameters- Artwork object

return type Boolean

removeArtwork()

parameters-artworkID

return type Boolean

getArtworkById();

parameters-artworkID

return type Artwork

searchArtworks()

searchArtworks();

parameters- keyword

return type list of Artwork Object

// User Favorites

addArtworkToFavorite();

parameters- userId, artworkId

return type boolean

removeArtworkFromFavorite()

parameters- userId, artworkId

return type boolean

getUserFavoriteArtworks()

parameters- userId

return type boolean

}

IvirtualArtGallery.py

```python
from abc import ABC, abstractmethod
from typing import List
from entity.artwork import Artwork

class IVirtualArtGallery(ABC):
    @abstractmethod
    def get_next_artworkID(self):
        pass

    @abstractmethod
    def addArtwork(self, artwork: Artwork)-> bool:
        pass

    @abstractmethod
    def add_gallery(self, gallery):
        pass

    @abstractmethod
    def updateArtwork(self, artwork: Artwork)-> bool:
        pass

    @abstractmethod
    def removeArtwork(self, artworkId:int)-> bool:
        pass

    @abstractmethod
    def getArtworkById(self, artworkId: int) -> Artwork:
        pass

    @abstractmethod
    def searchArtworks(self,search_object:str)-> List[Artwork]:
        pass

    @abstractmethod
    def addArtworkToFavorite(self,userId,artworkId)-> bool:
        pass

    @abstractmethod
    def removeArtworkFromFavorite(self,userId,artworkId)-> bool:
        pass

    @abstractmethod
    def getUserFavoriteArtworks(self,userId)-> List[Artwork]:
        pass
```

7: Connect your application to the SQL database:

1. Write code to establish a connection to your SQL database.

Create a utility class DBConnection in a package util with a static variable connection of Type Connection and a static method getConnection() which returns connection.

Connection properties supplied in the connection string should be read from a property file.

Create a utility class PropertyUtil which contains a static method named getPropertyString() which reads a property fie containing connection details like hostname, dbname, username, password, port number and returns a connection string.

DBConnection.py

```
import pyodbc
from util.PropertyUtil import PropertyUtil

class DBConnection:

    @staticmethod
    def getConnection():
        try:
            connection_string=PropertyUtil.getPropertyString()
            connection=pyodbc.connect(connection_string)
            print("Connected successfully")
            return connection
        except Exception as e:
            print(str(e) + '--Database is not connected--')
            return None
```

PropertyUtil.py

```
class PropertyUtil:

    @staticmethod
    def
getPropertyString(property_file_path=r"C:\Users\Asus\OneDrive\Desktop\VirtualA
rtGallery\util\PropertyFile.txt"):
        try:
            with open(property_file_path, 'r') as file:
                properties = {}
                for line in file:
                    if '=' in line:
                        key, value = line.strip().split('=', 1)  # Split by
'=' only on the first occurrence
                        properties[key.strip()] = value.strip()
```

```
                # Create the connection string
                connection_string = f"DRIVER={{ODBC Driver 17 for SQL
Server}};" \
                                    f"SERVER={properties['server']};" \
                                    f"DATABASE={properties['dbname']};" \
                                    f"Trusted_Connection={properties['trusted_
connection']};"
                return connection_string
        except ValueError as ve:
            print('db is missing',ve)
        except Exception as e:
            print(f"Error reading property file: {e}")
            return None
```

PropertyFile.txt

driver = {SQL Server}

server = LAPTOP-Q72Q77L5\SQLEXPRESS

dbname = VirtualArtGallery

trusted_connection = yes

8: Service implementation

1. Create a Service class CrimeAnalysisServiceImpl in dao with a static variable named connection of type Connection which can be assigned in the constructor by invoking the getConnection() method in DBConnection class

2. Provide implementation for all the methods in the interface.

VirtualArtGalleryImpl.py

```
from typing import List

from dao.IvirtualArtGallery import IVirtualArtGallery
from entity.artwork import Artwork
from entity.gallery import Gallery
from exception.ArtWorkNotFoundException import ArtWorkNotFoundException
from exception.UserNotFoundException import UserNotFoundException
from util.DBConnection import DBConnection
from tabulate import tabulate


class VirtualArtGalleryImpl(IVirtualArtGallery):
    connection=None
```

```
    def __init__(self):
        self.connection=DBConnection.getConnection()

    def get_next_artworkID(self):
        conn = DBConnection.getConnection()
        cursor = conn.cursor()
        try:
            cursor.execute("SELECT Max(ArtworkID) FROM Artwork")
            max_id = cursor.fetchone()[0]
            return (max_id + 1) if max_id is not None else 1
        except Exception as e:
            print(e)
            return 1
        finally:
            cursor.close()

    def addArtwork(self, artwork):
        cursor = self.connection.cursor()
        try:
            query = "INSERT INTO Artwork
(ArtworkID,Title,Description,CreationDate,Medium,ImageURL) VALUES
(?,?,?,?,?,?)"
            cursor.execute(query,(self.get_next_artworkID(),artwork.get_title(
),artwork.get_description(),artwork.get_creationDate(),artwork.get_medium(),ar
twork.get_imageURL()))
            self.connection.commit()
            print("------Artwork added------")
            return True
        except Exception as e:
            print("------Error in adding Artwork------",e)
            self.connection.rollback()
            return False
        finally:
            cursor.close()


    def updateArtwork(self, artwork, artworkId):
        cursor = self.connection.cursor()
        try:

            query = "UPDATE Artwork SET "
            params = []

            if artwork.get_title():
                query += "Title=?, "
                params.append(artwork.get_title())
            if artwork.get_description():
                query += "Description=?, "
                params.append(artwork.get_description())
```

```
            if artwork.get_creationDate():
                query += "CreationDate=?, "
                params.append(artwork.get_creationDate())
            if artwork.get_medium():
                query += "Medium=?, "
                params.append(artwork.get_medium())
            if artwork.get_imageURL():
                query += "ImageURL=?, "
                params.append(artwork.get_imageURL())


            query = query.rstrip(", ")
            query += " WHERE ArtworkID=?"
            params.append(artworkId)

            cursor.execute(query, tuple(params))
            self.connection.commit()
            print("------Artwork updated------")
            return True

        except Exception as e:
            print("------Error in updating Artwork------", e)
            return False
        finally:
            cursor.close()

    def removeArtwork(self, artworkId):
        cursor = self.connection.cursor()
        try:
            query = "SELECT Count(*) FROM Artwork WHERE ArtworkID=?"
            cursor.execute(query,(artworkId,))
            count = cursor.fetchone()[0]
            if count == 0:
                raise ArtWorkNotFoundException(artworkId)

            # Remove any references in User_Favorite_Artwork table
            delete_favorites_query = "DELETE FROM User_Favorite_Artwork WHERE
ArtworkID=?"
            cursor.execute(delete_favorites_query, (artworkId,))

            # Remove any references in Artwork_Gallery table
            delete_gallery_query = "DELETE FROM Artwork_Gallery WHERE
ArtworkID=?"
            cursor.execute(delete_gallery_query, (artworkId,))


            query='DELETE FROM Artwork WHERE ArtworkID=?'
            cursor.execute(query,(artworkId,))
```

```
            self.connection.commit()
            print("------Artwork removed------")
            return True


        except ArtWorkNotFoundException as e:
            print(e)
            return False

        except Exception as e:
            print("------Error in removing Artwork------",e)
            self.connection.rollback()
            return False
        finally:
            cursor.close()


    def updateGallery(self, gallery):
        try:
            cursor = self.connection.cursor()
            query = "UPDATE Gallery SET name = %s, description = %s, location
= %s, curator = %s, openinghours = %s WHERE GalleryID = %s"
            cursor.execute(query, (
                gallery.get_name(), gallery.get_description(),
gallery.get_location(), gallery.get_curator(),
                gallery.get_opening_hours(), gallery.get_gallery_id()))
            self.connection.commit()
            print("Gallery updated")
            return True
        except Exception as e:
            print("Error updating gallery", e)
            self.connection.rollback()
            return False
        finally:
            cursor.close()

    def getArtworkById(self, artworkId: int) -> Artwork:
        try:
            cursor = self.connection.cursor()
            query = 'SELECT * FROM Artwork WHERE ArtworkID=?'
            cursor.execute(query, (artworkId,))
            artwork = cursor.fetchone()
            if artwork is None:
                raise ArtWorkNotFoundException(artworkId)
            else:
                artwork_details = [
                    ['ArtworkID', artwork[0]],
                    ['Title', artwork[1]],
                    ['Description', artwork[2]],
```

```
                    ['CreationDate', artwork[3]],
                    ['Medium', artwork[4]],
                    ['ImageURL', artwork[5]],
                ]
                print("Artwork details")
                print(tabulate(artwork_details, tablefmt="grid"))
                return artwork
        except ArtWorkNotFoundException as e:
            print(e)
        except Exception as e:
            print("Error in getting the details:",e)
        finally:
            cursor.close()


    def searchArtworks(self, search_object) -> list[Artwork]:
        cursor = self.connection.cursor()
        artworks = []
        try:
            query = 'SELECT * FROM Artwork WHERE Title LIKE ? OR Medium LIKE ?
OR Description LIKE ?'
            cursor.execute(query, (f'%{search_object}%', f'%{search_object}%',
f'%{search_object}%'))
            artwork_data = cursor.fetchall()

            if artwork_data:
                artwork_table = []
                for artwork in artwork_data:
                    artwork_details = Artwork(
                        artworkId=artwork[0],     # Assuming ArtworkID is the
first column
                        title=artwork[1],         # Title is the second column
                        description=artwork[2],   # Description is the third
column
                        creationDate=artwork[3],  # CreationDate is the fourth
column
                        medium=artwork[4],        # Medium is the fifth column
                        imageURL=artwork[5]       # ImageURL is the sixth
column
                    )
                    artworks.append(artwork_details)
                    # Prepare a list of artwork details for printing
                    artwork_table.append([
                        artwork_details.artworkId,
                        artwork_details.title,
                        artwork_details.description,
                        artwork_details.creationDate,
                        artwork_details.medium,
                        artwork_details.imageURL
```

```
                    ])

                # Print the artwork details using tabulate
                print("Artworks found:")
                print(tabulate(artwork_table, headers=["Artwork ID", "Title",
                                                       "Description", "Creation
Date",
                                                       "Medium", "Image URL"],
tablefmt="grid"))



                return artworks
            else:
                print("No artwork found matching the search term")
                return artworks
        except Exception as e:
            print("Error in searching artworks:",e)
            self.connection.rollback()
            return []
        finally:
            cursor.close()

    def addArtworkToFavorite(self, userId, artworkId) -> bool:
        cursor = self.connection.cursor()
        try:

            query = "SELECT * FROM Artwork WHERE ArtworkID=?"
            cursor.execute(query, (artworkId,))
            if cursor.fetchone() is None:
                raise ArtWorkNotFoundException(artworkId)

            query = 'INSERT INTO User_Favorite_Artwork(UserID,ArtworkID)
VALUES (?,?)'
            cursor.execute(query, (userId, artworkId))
            self.connection.commit()
            print("Added to favorites")
            return True


        except ArtWorkNotFoundException as e:
            print(e)
            return False

        finally:
            cursor.close()

    def removeArtworkFromFavorite(self, userId, artworkId) -> bool:
```

```
        cursor = self.connection.cursor()
        try:

            query = "SELECT * FROM Artwork WHERE ArtworkID=?"
            cursor.execute(query, (artworkId,))
            if cursor.fetchone() is None:
                raise ArtWorkNotFoundException(artworkId)

            query = "DELETE FROM User_Favorite_Artwork WHERE UserID=? AND
ArtworkID=?"
            cursor.execute(query, (userId, artworkId))
            self.connection.commit()
            print("Removed from favorites")
            return True


        except ArtWorkNotFoundException as e:
            print(e)
            return False

        finally:
            cursor.close()

    def getUserFavoriteArtworks(self, userId):
        cursor = self.connection.cursor()
        try:
            query = 'SELECT a.* FROM Artwork a JOIN User_Favorite_Artwork u on
a.ArtworkID=u.ArtworkID WHERE UserID=?'
            cursor.execute(query, (userId,))
            artwork_data = cursor.fetchall()
            if artwork_data:
                for artwork in artwork_data:
                    artwork_details = [
                        ['Artwork ID', artwork[0]],
                        ['Title', artwork[1]],
                        ['Description', artwork[2]],
                        ['CreationDate', artwork[3]],
                        ['Medium', artwork[4]],
                        ['ImageURL', artwork[5]]
                    ]
                    print(tabulate(artwork_details, tablefmt="grid"))
            else:
                raise UserNotFoundException(userId)
        except Exception as e:
            print(e)
        finally:
            cursor.close()
```

```
    def add_gallery(self, gallery):
        try:
            cursor = self.connection.cursor()
            query = "INSERT INTO Gallery (name, description, location,
curator, openinghours) VALUES (%s, %s, %s, %s, %s)"
            cursor.execute(query, (
            gallery.get_name(), gallery.get_description(),
gallery.get_location(), gallery.get_curator(),
            gallery.get_openingHours()))
            self.connection.commit()
            print("Gallery added")
            return True
        except Exception as e:
            print("Error adding gallery", e)
            self.connection.rollback()
            return False
        finally:
            cursor.close()

    def update_gallery(self, gallery):
        try:
            cursor = self.connection.cursor()
            query = "UPDATE Gallery SET name = %s, description = %s, location
= %s, curator = %s, openinghours = %s WHERE GalleryID = %s"
            cursor.execute(query, (
                gallery.get_name(), gallery.get_description(),
gallery.get_location(), gallery.get_curator(),
                gallery.get_openingHours(), gallery.get_galleryId()))
            self.connection.commit()
            print("Gallery updated")
            return True
        except Exception as e:
            print("Error updating gallery", e)
            self.connection.rollback()
            return False
        finally:
            cursor.close()

    def delete_gallery(self, galleryId):
        try:
            cursor = self.connection.cursor()
            query = "DELETE FROM Gallery WHERE GalleryID = %s"
            cursor.execute(query, (galleryId,))
            self.connection.commit()
            print("Gallery deleted")
            return True
        except Exception as e:
            print("Error deleting gallery", e)
```

```
            self.connection.rollback()
            return False
        finally:
            cursor.close()

    def search_galleries(self, search_term: str) ->list[Gallery]:
        cursor=None
        try:
            cursor = self.connection.cursor(dictionary=True)
            query = "SELECT * FROM Gallery WHERE Name LIKE %s OR Description
LIKE %s"
            cursor.execute(query, (f"%{search_term}%", f"%{search_term}%"))
            galleries_data = cursor.fetchall()
            galleries = []
            for gallery_data in galleries_data:
                gallery = Gallery(
                    gallery_data['GalleryID'],
                    gallery_data['Name'],
                    gallery_data['Description'],
                    gallery_data['Location'],
                    gallery_data['Curator'],
                    gallery_data['OpeningHours']
                )
                galleries.append(gallery)
            return galleries
        except Exception as e:
            print("Error searching galleries:", e)
            return []
        finally:
            cursor.close()
```

9: Exception Handling

Create the exceptions in package exceptions

Define the following custom exceptions and throw them in methods whenever needed. Handle all the

exceptions in main method,

1. ArtWorkNotFoundException :throw this exception when user enters an invalid id which doesn't

exist in db

2. UserNotFoundException :throw this exception when user enters an invalid id which doesn't

exist in db

9. Main Method

Create class named MainModule with main method in main package.

Trigger all the methods in service implementation class.

ArtWorkNotFoundException.py

```
class ArtWorkNotFoundException(Exception):
    """Raised when an artwork is not found in the database."""
    def __init__(self, artworkId):
        message = f"Artwork with ID '{artworkId}' not found in the database."
        super().__init__(message)
```

UserNotFoundException.py

```
class UserNotFoundException(Exception):
    """Raised when a user is not found in the database."""
    def __init__(self, userId):
        message = f"User with ID '{userId}' not found in the database."
        super().__init__(message)
```

Main.py

```
import sys
import os

base_dir = os.path.abspath(os.path.join(os.path.dirname(__file__), ".."))
sys.path.append(base_dir)
from tabulate import tabulate
from exception.UserNotFoundException import UserNotFoundException
from exception.ArtWorkNotFoundException import ArtWorkNotFoundException
from dao.VirtualArtGalleryImpl import VirtualArtGalleryImpl
from entity.artwork import Artwork


class MainModule:
    def __init__(self):
        self.service=VirtualArtGalleryImpl()
    def menu(self):
        print("*"*50)
        print("Welcome to Virtual Art Gallery")
        print("*"*50)
        while True:
            menu=[
                ["1.","Add artwork"],
                ["2.","Update artwork"],
                ["3.","Remove artwork"],
                ["4.","Get Artwork By ID"],
                ["5.","Search artworks"],
                ["6.","Add Artwork To Favorite"],
                ["7.","Remove Artwork From Favorite"],
                ["8","Get User Favorite Artworks"],
```

```
            ["9","Exit"]
        ]

        print(tabulate(menu,headers=["Option","Description"],tablefmt="gri
d"))

        choice=input("Enter your choice: ")

        if choice=="1":
            artworkId=self.service.get_next_artworkID()
            title=input("Enter artwork title: ")
            description=input("Enter artwork description: ")
            creationDate=input("Enter artwork creation date: ")
            medium=input("Enter artwork medium: ")
            imageUrl=input("Enter artwork image url: ")
            if artworkId is None:
                print("-------Artwork ID cannot be null------")
            else:
                artwork=Artwork(
                    artworkId=artworkId,
                    title=title,
                    description=description,
                    creationDate=creationDate,
                    medium=medium,
                    imageURL=imageUrl
                )
                self.service.addArtwork(artwork)


        elif choice == "2":
            try:
                artworkId = int(input("Enter Artwork ID to update: "))
                cursor = self.service.connection.cursor()
                query = "SELECT * FROM Artwork WHERE ArtworkID=?"
                cursor.execute(query, (artworkId,))
                if cursor.fetchone() is None:
                    raise ArtWorkNotFoundException(artworkId)


                newTitle = input("Enter new artwork title (leave blank to
skip): ")
                newDescription = input("Enter new artwork description
(leave blank to skip): ")
                newCreationDate = input("Enter new artwork creation date
(leave blank to skip): ")
                newMedium = input("Enter new artwork medium (leave blank
to skip): ")
```

```
                    newImageUrl = input("Enter new artwork image URL (leave
blank to skip): ")

                    artwork = Artwork(
                        artworkId=artworkId,
                        title=newTitle if newTitle else None,
                        description=newDescription if newDescription else
None,
                        creationDate=newCreationDate if newCreationDate else
None,
                        medium=newMedium if newMedium else None,
                        imageURL=newImageUrl if newImageUrl else None
                    )

                    self.service.updateArtwork(artwork, artworkId)
                except ArtWorkNotFoundException as e:
                    print("Error:", e)
                except Exception as e:
                    print("An unexpected error occurred:", e)



            elif choice=="3":
                try:
                    artworkId = int(input("Enter Artwork ID to remove: "))
                except Exception as e:
                    print("Invalid input for Artwork ID.",e)
                    continue
                self.service.removeArtwork(artworkId)

            elif choice=="4":
                try:
                    artworkId=int(input("Enter Artwork ID to get details: "))
                    self.service.getArtworkById(artworkId)
                except Exception as e:
                    print("Invalid input:",e)


            elif choice=="5":
                search_object=input("Enter Search Object: ")
                self.service.searchArtworks(search_object)

            elif choice=="6":
                try:
                    userID=int(input("Enter User ID: "))
                    cursor = self.service.connection.cursor()
                    query = "SELECT * FROM Users WHERE UserID=?"
                    cursor.execute(query, (userID,))
                    if cursor.fetchone() is None:
```

```
                        raise UserNotFoundException(userID)
                    artworkId=int(input("Enter Artwork ID to add to
favourites: "))
                    self.service.addArtworkToFavorite(userID,artworkId)
                except UserNotFoundException as e:
                    print("Error:", e)
                except Exception as e:
                    print("Invalid input:",e)


            elif choice=="7":
                try:
                    userID=int(input("Enter User ID: "))
                    cursor = self.service.connection.cursor()
                    query = "SELECT * FROM Users WHERE UserID=?"
                    cursor.execute(query, (userID,))
                    if cursor.fetchone() is None:
                        raise UserNotFoundException(userID)  # Raise exception
if user is not found

                    artworkId=int(input("Enter Artwork ID to remove
favourites: "))
                    self.service.removeArtworkFromFavorite(userID,artworkId)
                except UserNotFoundException as ue:
                    print("Error:",ue)  # Handle the exception here
                except Exception as e:
                    print('Invalid input:',e)


            elif choice=="8":
                try:
                    userID=int(input("Enter User ID: "))
                    self.service.getUserFavoriteArtworks(userID)
                except Exception as e:
                    print("Invalid input:",e)


            elif choice=="9":
                print("Thank you for using Virtual Art Gallery!")
                break
if __name__ == '__main__':
    MainModule().menu()
```

10. Unit Testing

Creating Unit test cases for a Virtual Art Gallery system is essential to ensure that the system functions correctly. Below are sample test case questions that can serve as a starting point for your JUnit test suite:

1. Artwork Management:

a. Test the ability to upload a new artwork to the gallery.

b. Verify that updating artwork details works correctly.

c. Test removing an artwork from the gallery.

d. Check if searching for artworks returns the expected results

2. Gallery Management:

a. Test creating a new gallery.

b. Verify that updating gallery information works correctly.

c. Test removing a gallery from the system.

d. Check if searching for galleries returns the expected results.

ArtworkTest.py

```
import sys
import os
base_dir = os.path.abspath(os.path.join(os.path.dirname(__file__), ".."))
sys.path.append(base_dir)

import unittest
from unittest.mock import MagicMock
from dao.VirtualArtGalleryImpl import VirtualArtGalleryImpl
from entity.artwork import Artwork

class TestArtworkManagement(unittest.TestCase):
    def setUp(self):
        self.service = VirtualArtGalleryImpl()

    def test_add_artwork_success(self):
        self.service.connection = MagicMock()
        cursor_mock = self.service.connection.cursor.return_value
        cursor_mock.execute.return_value = None
        artwork = Artwork(artworkId="101",
                          title="Testing",
                          description="Test Description",
                          creationDate="2024-03-31",
```

```
                            medium="english",
                            imageURL="http://test.jpg")
        result = self.service.addArtwork(artwork)
        self.assertTrue(result)
        cursor_mock.execute.assert_called_once()

    def test_add_artwork_failure(self):
        self.service.connection = MagicMock()
        cursor_mock = self.service.connection.cursor.return_value
        cursor_mock.execute.side_effect = Exception("Mocked DB Error")
        artwork = Artwork(artworkId="101",
                            title="Testing",
                            description="Test Description",
                            creationDate="2024-03-31",
                            medium="english",
                            imageURL="http://test.jpg")
        result = self.service.addArtwork(artwork)
        self.assertFalse(result)
        cursor_mock.execute.assert_called_once()

    def test_update_artwork_success(self):
        self.service.connection = MagicMock()
        mock_cursor = MagicMock()
        self.service.connection.cursor.return_value = mock_cursor
        mock_cursor.execute.return_value = None
        updated_artwork = Artwork(artworkId="1",
                                  title="Updated Title",
                                  description="Updated Description",
                                  creationDate="2024-05-13",
                                  medium="french",
                                  imageURL="http://updated_artwork.jpg")
        result =
self.service.updateArtwork(updated_artwork,updated_artwork.artworkId)
        self.assertTrue(result)
        mock_cursor.execute.assert_called_once()

    def test_update_artwork_failure(self):
        self.service.connection = MagicMock()
        cursor_mock = self.service.connection.cursor.return_value
        cursor_mock.execute.side_effect = Exception("Mocked DB Error")
        updated_artwork = Artwork(artworkId="1",
                                  title="Updated Title",
                                  description="Updated Description",
                                  creationDate="2024-05-13",
                                  medium="french",
                                  imageURL="http://updated_artwork.jpg")
        result =
self.service.updateArtwork(updated_artwork,updated_artwork.artworkId)
```

```
        self.assertFalse(result)
        cursor_mock.execute.assert_called_once()


    def test_remove_artwork_success(self):
        self.service.connection = MagicMock()
        cursor_mock = self.service.connection.cursor.return_value
        cursor_mock.execute.return_value = None
        artwork_id = "1"

        result = self.service.removeArtwork(artwork_id)
        self.assertTrue(result)

        # Assert that execute was called 4 times
        self.assertEqual(cursor_mock.execute.call_count, 4)

        # Verify the calls were made with the correct arguments
        cursor_mock.execute.assert_any_call("SELECT Count(*) FROM Artwork
WHERE ArtworkID=?", (artwork_id,))
        cursor_mock.execute.assert_any_call("DELETE FROM User_Favorite_Artwork
WHERE ArtworkID=?", (artwork_id,))
        cursor_mock.execute.assert_any_call("DELETE FROM Artwork_Gallery WHERE
ArtworkID=?", (artwork_id,))
        cursor_mock.execute.assert_any_call("DELETE FROM Artwork WHERE
ArtworkID=?", (artwork_id,))


    def test_remove_artwork_failure(self):
        self.service.connection = MagicMock()
        cursor_mock = self.service.connection.cursor.return_value
        cursor_mock.execute.side_effect = Exception("Mocked DB Error")
        artwork_id = "1"
        result = self.service.removeArtwork(artwork_id)
        self.assertFalse(result)
        cursor_mock.execute.assert_called_once()

    def test_search_artworks_success(self):
        self.service.connection = MagicMock()
        mock_cursor = MagicMock()
        self.service.connection.cursor.return_value = mock_cursor

        # Mock the fetchall return value to match the database schema
        mock_cursor.fetchall.return_value = [
        (1, 'Artwork 1', 'Description 1', '2024-05-15', 'Medium 1',
'image1.jpg'),
        (2, 'Artwork 2', 'Description 2', '2024-05-16', 'Medium 2',
'image2.jpg')
        ]
```

```
        search_term = "elephant"
        artworks = self.service.searchArtworks(search_term)

        # Debug print to check the result of searchArtworks
        #print("Artworks returned from search:", artworks)

        self.assertIsInstance(artworks, list)
        self.assertEqual(len(artworks), 2)
        self.assertEqual(artworks[0].get_title(), 'Artwork 1')
        self.assertEqual(artworks[1].get_title(), 'Artwork 2')



if __name__ == '__main__':
    unittest.main()
```

GalleryTest.py

```
import sys
import os
base_dir = os.path.abspath(os.path.join(os.path.dirname(__file__), ".."))
sys.path.append(base_dir)

import unittest
from unittest.mock import MagicMock
from dao.VirtualArtGalleryImpl import VirtualArtGalleryImpl
from entity.gallery import Gallery

class TestGalleryManagement(unittest.TestCase):
    def setUp(self):
        self.service = VirtualArtGalleryImpl()

    def test_create_gallery_success(self):
        self.service.connection = MagicMock()
        cursor_mock = self.service.connection.cursor.return_value
        cursor_mock.execute.return_value = None
        gallery = Gallery(galleryId="Test Gallery Id",
                          name="Test Gallery",
                          description="Test Description",
                          location="Test Location",
                          curator="Test Curator",
                          openingHours="Test Opening Hours")
        result = self.service.add_gallery(gallery)
        self.assertTrue(result)
        cursor_mock.execute.assert_called_once()
```

```
def test_create_gallery_failure(self):
    self.service.connection = MagicMock()
    cursor_mock = self.service.connection.cursor.return_value
    cursor_mock.execute.side_effect = Exception("Mocked DB Error")
    gallery = Gallery(galleryId="Test Gallery Id",
                      name="Test Gallery",
                      description="Test Description",
                      location="Test Location",
                      curator="Test Curator",
                      openingHours="Test Opening Hours")
    result = self.service.add_gallery(gallery)
    self.assertFalse(result)
    cursor_mock.execute.assert_called_once()

def test_update_gallery_success(self):
    self.service.connection = MagicMock()
    cursor_mock = self.service.connection.cursor.return_value
    cursor_mock.execute.return_value = None
    gallery_id = 1
    updated_gallery = Gallery(galleryId=gallery_id,
                              name="Updated Gallery",
                              description="Updated Description",
                              location="Updated Location",
                              curator="Updated Curator",
                              openingHours="Updated Opening Hours")
    result = self.service.update_gallery(updated_gallery)
    self.assertTrue(result)
    cursor_mock.execute.assert_called_once()

def test_update_gallery_failure(self):
    self.service.connection = MagicMock()
    cursor_mock = self.service.connection.cursor.return_value
    cursor_mock.execute.side_effect = Exception("Mocked DB Error")
    gallery_id = 1
    updated_gallery = Gallery(galleryId=gallery_id,
                              name="Updated Gallery",
                              description="Updated Description",
                              location="Updated Location",
                              curator="Updated Curator",
                              openingHours="Updated Opening Hours")
    result = self.service.update_gallery(updated_gallery)
    self.assertFalse(result)
    cursor_mock.execute.assert_called_once()

def test_remove_gallery_success(self):
    self.service.connection = MagicMock()
    cursor_mock = self.service.connection.cursor.return_value
    cursor_mock.execute.return_value = None
```

```
        gallery_id = 1
        result = self.service.delete_gallery(gallery_id)
        self.assertTrue(result)
        cursor_mock.execute.assert_called_once()

    def test_remove_gallery_failure(self):
        self.service.connection = MagicMock()
        cursor_mock = self.service.connection.cursor.return_value
        cursor_mock.execute.side_effect = Exception("Mocked DB Error")
        gallery_id = 1
        result = self.service.delete_gallery(gallery_id)
        self.assertFalse(result)
        cursor_mock.execute.assert_called_once()

    def test_search_galleries_success(self):
        self.service.connection = MagicMock()
        mock_cursor = MagicMock()
        self.service.connection.cursor.return_value = mock_cursor

        mock_cursor.fetchall.return_value = [
            {'GalleryID': 1, 'Name': 'Gallery',
             'Description': 'A gallery featuring impressionist artworks',
'Location': '1st Floor', 'Curator': 1,
             'OpeningHours': '9 AM - 5 PM'},
            {'GalleryID': 2, 'Name': 'Renaissance', 'Description': 'A gallery
featuring renaissance artworks',
             'Location': '2nd Floor', 'Curator': 2, 'OpeningHours': '10 AM - 6
PM'}
        ]
        search_term = "Art"
        galleries = self.service.search_galleries(search_term)
        self.assertEqual(len(galleries), 2)
        self.assertEqual(galleries[0].get_name(), 'Gallery')
        self.assertEqual(galleries[1].get_name(), 'Renaissance')

if __name__ == '__main__':
    unittest.main()
```

Outputs:

```
select * from Artwork;
```

Results Messages

	ArtworkID	Title	Description	CreationDate	Medium	ImageURL
1	101	The Starry Night	Famous painting by Vincent van Gogh	1889-06-01	Oil on Canvas	https://example.com/starry_night.jpg
2	102	The Kiss	Symbolist painting by Gustav Klimt	1907-01-01	Oil on Canvas	https://example.com/the_kiss.jpg
3	103	The Thinker	Bronze sculpture by Auguste Rodin	1904-01-01	Bronze	https://example.com/the_thinker.jpg
4	104	Guernica	Famous anti-war painting by Pablo Picasso	1937-01-01	Oil on Canvas	https://example.com/guernica.jpg
5	105	Water Lilies	Series of paintings by Claude Monet	1899-01-01	Oil on Canvas	https://example.com/water_lilies.jpg

```
select * from Artist;
```

Results Messages

	ArtistID	Name	Biography	BirthDate	Nationality	Website	ContactInfomation
1	201	Vincent van Gogh	Dutch post-impressionist painter	1853-03-30	Dutch	https://vangogh.com	contact@vangogh.com
2	202	Gustav Klimt	Austrian symbolist painter	1862-07-14	Austrian	https://klimt.com	contact@klimt.com
3	203	Auguste Rodin	French sculptor	1840-11-12	French	https://rodin.com	contact@rodin.com
4	204	Pablo Picasso	Spanish painter and sculptor	1881-10-25	Spanish	https://picasso.com	contact@picasso.com
5	205	Claude Monet	French impressionist painter	1840-11-14	French	https://monet.com	contact@monet.com

```
select * from Users;
```

Results Messages

	UserID	Username	Password	Email	FirstName	LastName	DateOfBirth	ProfilePicture
1	301	artfanatic23	pass123	user1@example.com	Emily	Johnson	1995-09-20	https://example.com/emily_profile.jpg
2	302	paintinglover78	iloveart	user2@example.com	Michael	Smith	1988-04-12	https://example.com/michael_profile.jpg
3	303	artcollector1	art123	user3@example.com	Sophia	Anderson	1976-12-05	https://example.com/sophia_profile.jpg

```
select * from Gallery;
```

Results Messages

	GalleryID	Name	Description	Location	Curator	OpeningHours
1	401	National Gallery of Art	Art museum in Washington D.C.	Washington D.C., USA	201	10 AM - 5 PM, Monday to Sunday
2	402	Tate Britain	Art gallery in London	London, UK	205	9:30 AM - 6 PM, Monday to Saturday
3	403	Museum of Modern Art	Modern art museum in New York City	New York, USA	204	10:30 AM - 5:30 PM, Thursday to Tuesday

```
select * from User_Favorite_Artwork;
```

Results Messages

	UserID	ArtworkID
1	301	101
2	302	102
3	303	104
4	303	105

```
select * from Artwork_Gallery;
```

Results Messages

	ArtworkID	GalleryID
1	101	401
2	102	402
3	103	401
4	104	403
5	105	402

```
Connected successfully
**************************************************
Welcome to Virtual Art Gallery
**************************************************
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice:
```

```
Connected successfully
**************************************************
Welcome to Virtual Art Gallery
**************************************************
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 1
Connected successfully
Enter artwork title: The moon
Enter artwork description: Series of paintings of moon
Enter artwork creation date: 2024-11-02
Enter artwork medium: Oil on canvas
Enter artwork image url: https://example.com/moon.jpg
Connected successfully
------Artwork added------
```

Results | Messages

	ArtworkID	Title	Description	CreationDate	Medium	ImageURL
1	101	The Starry Night	Famous painting by Vincent van Gogh	1889-06-01	Oil on Canvas	https://example.com/starry_night.jpg
2	102	The Kiss	Symbolist painting by Gustav Klimt	1907-01-01	Oil on Canvas	https://example.com/the_kiss.jpg
3	103	The Thinker	Bronze sculpture by Auguste Rodin	1904-01-01	Bronze	https://example.com/the_thinker.jpg
4	104	Guernica	Famous anti-war painting by Pablo Picasso	1937-01-01	Oil on Canvas	https://example.com/guernica.jpg
5	105	Water Lilies	Series of paintings by Claude Monet	1899-01-01	Oil on Canvas	https://example.com/water_lilies.jpg
6	106	The moon	Series of paintings of moon	2024-11-02	Oil on canvas	https://example.com/moon.jpg

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 2
Enter Artwork ID to update: 202
Error: Artwork with ID '202' not found in the database.
```

Results Messages

	ArtworkID	Title	Description	CreationDate	Medium	ImageURL
1	101	The Starry Night	Famous painting by Vincent van Gogh	1889-06-01	Oil on Canvas	https://example.com/starry_night.jpg
2	102	The Kiss	Symbolist painting by Gustav Klimt	1907-01-01	Oil on Canvas	https://example.com/the_kiss.jpg
3	103	The Thinker	Bronze sculpture by Auguste Rodin	1904-01-01	Bronze	https://example.com/the_thinker.jpg
4	104	Guernica	Famous anti-war painting by Pablo Picasso	1937-01-01	Oil on Canvas	https://example.com/guernica.jpg
5	105	Water Lilies	Series of paintings by Claude Monet	1899-01-01	Oil on Canvas	https://example.com/water_lilies.jpg
6	106	The moon	Series of paintings of moon	2024-11-02	Oil on canvas	https://example.com/moon.jpg

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 2
Enter Artwork ID to update: 105
Enter new artwork title (leave blank to skip):
Enter new artwork description (leave blank to skip):
Enter new artwork creation date (leave blank to skip):
Enter new artwork medium (leave blank to skip): Bronze
Enter new artwork image URL (leave blank to skip):
------Artwork updated------
```

Results Messages

	ArtworkID	Title	Description	CreationDate	Medium	ImageURL
1	101	The Starry Night	Famous painting by Vincent van Gogh	1889-06-01	Oil on Canvas	https://example.com/starry_night.jpg
2	102	The Kiss	Symbolist painting by Gustav Klimt	1907-01-01	Oil on Canvas	https://example.com/the_kiss.jpg
3	103	The Thinker	Bronze sculpture by Auguste Rodin	1904-01-01	Bronze	https://example.com/the_thinker.jpg
4	104	Guernica	Famous anti-war painting by Pablo Picasso	1937-01-01	Oil on Canvas	https://example.com/guernica.jpg
5	105	Water Lilies	Series of paintings by Claude Monet	1899-01-01	Bronze	https://example.com/water_lilies.jpg
6	106	The moon	Series of paintings of moon	2024-11-02	Oil on canvas	Https://example.com/moon.jpg

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 3
Enter Artwork ID to remove: 108
Artwork with ID '108' not found in the database.
```

Results Messages

	ArtworkID	Title	Description	CreationDate	Medium	ImageURL
1	101	The Starry Night	Famous painting by Vincent van Gogh	1889-06-01	Oil on Canvas	https://example.com/starry_night.jpg
2	102	The Kiss	Symbolist painting by Gustav Klimt	1907-01-01	Oil on Canvas	https://example.com/the_kiss.jpg
3	103	The Thinker	Bronze sculpture by Auguste Rodin	1904-01-01	Bronze	https://example.com/the_thinker.jpg
4	104	Guernica	Famous anti-war painting by Pablo Picasso	1937-01-01	Oil on Canvas	https://example.com/guernica.jpg
5	105	Water lillies	Series of arts of lillies	1990-11-01	Oil on canvas	https://example.com/lillies.jpg
6	106	The moon	Series of paintings of moon	2024-11-02	Oil on canvas	https://example.com/moon.jpg

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 3
Enter Artwork ID to remove: 106
------Artwork removed------
```

Results Messages

	ArtworkID	Title	Description	CreationDate	Medium	ImageURL
1	101	The Starry Night	Famous painting by Vincent van Gogh	1889-06-01	Oil on Canvas	https://example.com/starry_night.jpg
2	102	The Kiss	Symbolist painting by Gustav Klimt	1907-01-01	Oil on Canvas	https://example.com/the_kiss.jpg
3	103	The Thinker	Bronze sculpture by Auguste Rodin	1904-01-01	Bronze	https://example.com/the_thinker.jpg
4	104	Guernica	Famous anti-war painting by Pablo Picasso	1937-01-01	Oil on Canvas	https://example.com/guernica.jpg
5	105	Water lillies	Series of arts of lillies	1990-11-01	Oil on canvas	https://example.com/lillies.jpg

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 4
Enter Artwork ID to get details: 110
Artwork with ID '110' not found in the database.
```

```
|   Option | Description                    |
+==========+================================+
|        1 | Add artwork                    |
+----------+--------------------------------+
|        2 | Update artwork                 |
+----------+--------------------------------+
|        3 | Remove artwork                 |
+----------+--------------------------------+
|        4 | Get Artwork By ID              |
+----------+--------------------------------+
|        5 | Search artworks                |
+----------+--------------------------------+
|        6 | Add Artwork To Favorite        |
+----------+--------------------------------+
|        7 | Remove Artwork From Favorite   |
+----------+--------------------------------+
|        8 | Get User Favorite Artworks     |
+----------+--------------------------------+
|        9 | Exit                           |
+----------+--------------------------------+
Enter your choice: 4
Enter Artwork ID to get details: 102
Artwork details
+--------------+-----------------------------------+
| ArtworkID    | 102                               |
+--------------+-----------------------------------+
| Title        | The Kiss                          |
+--------------+-----------------------------------+
| Description  | Symbolist painting by Gustav Klimt |
+--------------+-----------------------------------+
| CreationDate | 1907-01-01                        |
+--------------+-----------------------------------+
| Medium       | Oil on Canvas                     |
+--------------+-----------------------------------+
| ImageURL     | https://example.com/the_kiss.jpg  |
+--------------+-----------------------------------+
```

Results Messages

	ArtworkID	Title	Description	CreationDate	Medium	ImageURL
1	101	The Starry Night	Famous painting by Vincent van Gogh	1889-06-01	Oil on Canvas	https://example.com/starry_night.jpg
2	102	The Kiss	Symbolist painting by Gustav Klimt	1907-01-01	Oil on Canvas	https://example.com/the_kiss.jpg
3	103	The Thinker	Bronze sculpture by Auguste Rodin	1904-01-01	Bronze	https://example.com/the_thinker.jpg
4	104	Guernica	Famous anti-war painting by Pablo Picasso	1937-01-01	Oil on Canvas	https://example.com/guernica.jpg
5	105	Water lillies	Series of arts of lillies	1990-11-01	Oil on canvas	https://example.com/lillies.jpg

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 5
Enter Search Object: Rose
No artwork found matching the search term

+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 5
Enter Search Object: Water
Artworks found:
+--------------+---------------+---------------------------+-----------------+---------------+--------------------------------+
|   Artwork ID | Title         | Description               | Creation Date   | Medium        | Image URL                      |
+==============+===============+===========================+=================+===============+================================+
|          105 | Water lillies | Series of arts of lillies | 1990-11-01      | Oil on canvas | https://example.com/lillies.jpg |
+--------------+---------------+---------------------------+-----------------+---------------+--------------------------------+
```

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 6
Enter User ID: 100
Error: User with ID '100' not found in the database.
```

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 6
Enter User ID: 301
Enter Artwork ID to add to favourites: 200
Artwork with ID '200' not found in the database.
```

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 6
Enter User ID: 301
Enter Artwork ID to add to favourites: 103
Added to favorites
```

Results | Messages

	UserID	ArtworkID
1	301	101
2	301	103
3	302	102
4	303	104
5	303	105

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 7
Enter User ID: 301
Enter Artwork ID to remove favourites: 103
Removed from favorites
```

Results | Messages

	UserID	ArtworkID
1	301	101
2	302	102
3	303	104
4	303	105

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 8
Enter User ID: 301
+--------------+--------------------------------------+
| Artwork ID   | 101                                  |
+--------------+--------------------------------------+
| Title        | The Starry Night                     |
+--------------+--------------------------------------+
| Description  | Famous painting by Vincent van Gogh  |
+--------------+--------------------------------------+
| CreationDate | 1889-06-01                           |
+--------------+--------------------------------------+
| Medium       | Oil on Canvas                        |
+--------------+--------------------------------------+
| ImageURL     | https://example.com/starry_night.jpg |
+--------------+--------------------------------------+
```

```
+----------+------------------------------+
|   Option | Description                  |
+==========+==============================+
|        1 | Add artwork                  |
+----------+------------------------------+
|        2 | Update artwork               |
+----------+------------------------------+
|        3 | Remove artwork               |
+----------+------------------------------+
|        4 | Get Artwork By ID            |
+----------+------------------------------+
|        5 | Search artworks              |
+----------+------------------------------+
|        6 | Add Artwork To Favorite      |
+----------+------------------------------+
|        7 | Remove Artwork From Favorite |
+----------+------------------------------+
|        8 | Get User Favorite Artworks   |
+----------+------------------------------+
|        9 | Exit                         |
+----------+------------------------------+
Enter your choice: 8
Enter User ID: 305
User with ID '305' not found in the database.
```

```
+-----------+-------------------------------+
|   Option | Description                    |
+===========+===============================+
|        1 | Add artwork                    |
+-----------+-------------------------------+
|        2 | Update artwork                 |
+-----------+-------------------------------+
|        3 | Remove artwork                 |
+-----------+-------------------------------+
|        4 | Get Artwork By ID              |
+-----------+-------------------------------+
|        5 | Search artworks                |
+-----------+-------------------------------+
|        6 | Add Artwork To Favorite        |
+-----------+-------------------------------+
|        7 | Remove Artwork From Favorite   |
+-----------+-------------------------------+
|        8 | Get User Favorite Artworks     |
+-----------+-------------------------------+
|        9 | Exit                           |
+-----------+-------------------------------+
Enter your choice: 9
Thank you for using Virtual Art Gallery!

Process finished with exit code 0
```

ArtworkTest.py

```
Testing started at 10:50 ...
Launching unittests with arguments python -m unittest C:\Users\chand\PycharmProjects\VirtualArtGallery\test\ArtworkTest.py in C:\Users\chand\PycharmProjects\VirtualArtGallery\test

Connected successfully
Connected successfully
------Error in adding Artwork------ Mocked DB Error
Connected successfully
Connected successfully
------Artwork added------
Connected successfully
------Error in removing Artwork------ Mocked DB Error
Connected successfully
------Artwork removed------
Connected successfully
Artworks found:
+--------------+-----------+---------------+-----------------+----------+-------------+
|   Artwork ID | Title     | Description   | Creation Date   | Medium   | Image URL   |
+==============+===========+===============+=================+==========+=============+
|            1 | Artwork 1 | Description 1 | 2024-05-15      | Medium 1 | image1.jpg  |
+--------------+-----------+---------------+-----------------+----------+-------------+
|            2 | Artwork 2 | Description 2 | 2024-05-16      | Medium 2 | image2.jpg  |
+--------------+-----------+---------------+-----------------+----------+-------------+
Connected successfully
------Error in updating Artwork------ Mocked DB Error
Connected successfully
------Artwork updated------


Ran 7 tests in 0.151s

OK

Process finished with exit code 0
```

GalleryTest.py

```
Testing started at 10:52 ...
Launching unittests with arguments python -m unittest C:\Users\chand\PycharmProjects\VirtualArtGallery\test\GalleryTest.py in C:\Users\chand\PycharmProjects\VirtualArtGallery\test

Connected successfully
Error adding gallery Mocked DB Error
Connected successfully
Gallery added
Connected successfully
Error deleting gallery Mocked DB Error
Connected successfully
Gallery deleted
Connected successfully
Connected successfully
Error updating gallery Mocked DB Error
Connected successfully


Ran 7 tests in 0.125s

OK
Gallery updated

Process finished with exit code 0
```