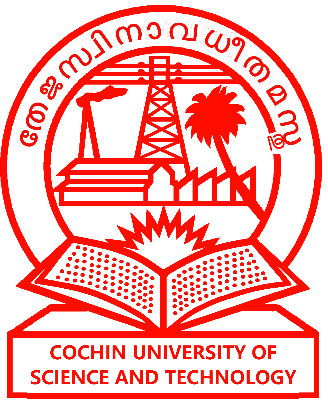
**COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY**



**ANDROID BASED PROJECT REPORT**

*On*

**ANDROID FITNESS APP**

Submitted in partial fulfilment for the award of degree

*Of*

***Bachelor of Technology***

*In*

***INFORMATION TECHNOLOGY***

*Of*

**COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY**

Submitted by

**NIKHIL KJ (20422516)**

**JUSTIN CS (20422511)**

**SREESHANTH CV (20422521)**

**A ABINAYA (20422501)**

*Under the guidance*

*Of*

**Dr. Jabir K V T**

**Associate Professor**

**Head of Department**

**Ms. Jesna**

**Asst. Professor**

**Ms. Athira K R**

**Asst. Professor**

**Division of IT**

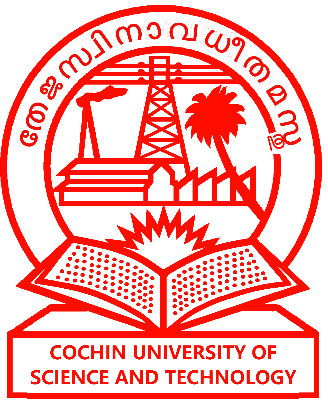
**Cochin University College of Engineering Kuttanadu**

**April, 2025**

**Department of Information Technology**

**COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**Department of Information Technology**



**CERTIFICATE**

This is to certify that the project report entitled “**ANDROID FITNESS APP**” submitted by **NIKHIL KJ (20422516), JUSTIN CS (20422511), SREESHANTH CV (20422521),**

**A ABINAYA (20422501)** of Cochin University of Science and Technology towards partial fulfilment of requirements for the award of the degree of **Bachelor of Technology in Information Technology** is a bonafide record work carried out by them under our supervision and guidance during the academic year 2025.

**Ms. Athira K R Dr. Jabir K V T**

**Asst. Professor Head Of Department**

**Ms. Jesna**

**Asst. Professor**

**DECLARATION**

We undersigned hereby declare that the project report "*Android Fittness App*" submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the Cochin University of Science and Technology, Kerala is a bonafide work done by us under supervision of Ms. Jesna and Ms. Athira K R, This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kuttanad **NIKHIL KJ (20422516)**

7/04/2025 **JUSTIN CS (20422511)**

**SREESHANTH CV (20422521)**

**A ABINAYA (20422501)**

**ACKNOWLEDGMENT**

We are pleased to present the “Android Fitness App” project and take this opportunity to express our profound gratitude to all those people who helped us in the completion of this project.

We are grateful to our principal, Dr. Asaletha R for the facilities provided by her for the preparation of this major project.

We are thankful to The Head of Department, Information Technology, Dr. Jabir KVT, for giving us all the support.

We express our sincere thanks to our Project Coordinator, Ms. Jesna, Assistant Professor and Ms. Athira K R, Assistant Professor, Department of Information Technology, for giving us innovative suggestions, timely advice, and suggestions during this endeavour.

Last, but not least, we express our gratitude to all the staff members of the Department of Information Technology for their valuable advice and timely suggestions. We would also like to thank the faculties of Cochin University College of Engineering Kuttanadu, CUSAT for their encouragement.

**ABSTRACT**

Android Fitness is a platform designed to provide users with convenient access to tailored fitness solutions through a user-friendly mobile application. The platform features two primary roles: Admin and User. Users can register by providing their personal details and explore a variety of workout plans. Each workout plan is designed to cater to different fitness levels, ensuring inclusivity and effectiveness. Users can track their progress by calculating BMI.The Admin oversees the platform, managing users, curating workout plans, and ensuring the quality and reliability of the content.The platform allows users to update their profiles and preferences to personalize their fitness experience, making it a dynamic and engaging environment.Android Fitness aims to create a secure and efficient fitness ecosystem where users can achieve their health goals conveniently and effectively, empowering them to lead healthier lives.

**CONTENTS**

1. INTRODUCTION

1.1 GENERAL INTRODUCTION ……………………………………... 2

1.2 OBJECTIVE ……………………………………………………..…. 2

1.3 SCOPE ……………………………………………………………… 2

2. REQUIREMENT ANALYSIS

2.1 FUNCTIONAL REQUIREMENT ………………………………… 4

2.2 NON-FUNCTIONAL REQUIREMENT ………………………….. 8

3. SYSTEM DESIGN

3.1 USECASE DIAGRAM …………………………………………… 10

3.2 USECASE SPECIFICATION …………………………………….. 12

3.3 CLASS DIAGRAM ……….………………………………………. 32

3.4 FLOW DIAGRAM ………………...…….…….………………….. 33

3.5 ENTITY RELATIONSHIP DIAGRAM…..………..…………....… 35

3.6 TABLE DESIGN………………………….……………………….. 36

4. IMPLEMENTATION

4.1 SYSTEM ARCHITECTURE ……………………………………... 39

4.2 TOOLS USED …………….………………………………………. 40

4.2.1 FRONTEND TOOL……………..…...………………………. 40

4.2.2 BACKEND TOOLS…………………..……………...………. 40

4.3 MODULES ……………………………………………………..… 40

4.3.1 ADMIN MODULE…………………………….…….………. 40

4.3.2 ARTIST MODULE ……....……...…….…………………….. 40

4.3.3 CUSTOMER MODULE …...…….…….………………….… 40

5. CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION…………………………………………………….. 42

5.2 FUTURE SCOPE ………………………………………………...…42

6. REFERENCES

6. REFERENCES ...……….….…………………………...….. 44

7. APPENDICES

7. APPENDICES ……….….……………………………….….46

7.1 DATABASE……….……………...……………………………... 46

7.2 ADMIN FUNCTIONS……….…...……………………………... 46

7.3 ARTIST FUNCTIONS …….….…….……………………………47

7.4 CUSTOMER FUNCTIONS ……….…...…….…………...……...49

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Description** | **Page Number** |
| **1** | Usecase Diagram | 11 |
| **2** | Class Diagram | 32 |
| **3** | DFD Level 0 | 33 |
| **4** | DFD Level 1 | 33 |
| **5** | DFD Level 2 | 34 |
| **6** | ER Diagram | 35 |
| **7** | System Architecture | 40 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Description** | **Page Number** |
| **1** | Admin | 36 |
| **2** | User | 36 |
| **3** | Workouts | 37 |
| **4** | Exercise | 37 |
| **5** | Workouts | 38 |

**CHAPTER 1**

**INTRODUCTION**

1. **INTRODUCTION**

**1.1 GENERAL INTRODUCTION**

The **Android Fitness App** aims to create a seamless and secure platform that helps users improve their fitness journey. It provides a user-friendly interface where users can log workouts, set fitness goals, and stay motivated. The app includes an admin panel that allows administrators to manage users and workout content while also having the authority to remove inappropriate material. Additionally, the application features a search function where users can input their preferred workout type, and the system will suggest relevant exercises accordingly. By integrating these features, the app ensures a smooth and personalized fitness experience for all users.

**1.2 OBJECTIVE**

The **objective** of the Android Fitness App is to provide a seamless and interactive platform that helps users improve their health and achieve their fitness goals.

**1.3 SCOPE**

The **Android Fitness App** encompasses key functionalities such as user registration, profile management, workout viewing, and real-time progress monitoring by calculating BMI. The app also includes a search feature that suggests relevant exercises based on user preferences. Admins play a crucial role in managing users, monitoring workout content, and ensuring a secure and engaging platform. By integrating these features into a well-structured system, the app provides a comprehensive fitness solution, empowering users to achieve their health goals while maintaining an interactive and user-friendly experience.

**CHAPTER 2**

**REQUIREMENT ANALYSIS**

1. **REQUIREMENT ANALYSIS**

Requirement analysis for the Android Fitness App was carried out by identifying the core functionalities needed to make the platform efficient and user-friendly. A detailed list of features was crafted, prioritizing them based on project constraints. The goal was to define essential features that would ensure the platform serves both Admin and Users effectively.

**2.1 FUNCTIONAL REQUIREMENTS**

**R.1 ADMIN MANAGEMENT**

R.1.1. Add Workouts

Input: Workout plan details including exercise,name,duration,difficulty level.

Output: Response message indicating success or failure.

Process: -The details are added to the database.

R.1.2. Delete Workouts

Input: Selects the workout to be deleted.

Output: Response message indicating success or failure.

Process: -The details are deleted from the database.

R.1.3. View Users

Input: Select view user button.

Output: User accounts displayed successfully.

R.1.4. Delete Users

Input: Select delete user button.

Output: User account deleted successfully.

Process: -The details are deleted from the database.

**R.2 USER REGISTRATION**

**R.2.1 Registration**

R.2.1.1 Selection of “Register” option

Input: “Register” option is selected.

Output: The artist is prompted to enter details like username, email, age, password.

R.2.1.2 Registration

Input: Entering details and pressing submit button

Output: The user is successfully registered.

**R.3 LOGIN**

**R.3.1 User Login**

R.3.1.1 Selection of “Login” button

Input: “Login” is selected.

Output: The user is prompted to enter username and password.

R.3.1.2 Login

Input: Entering username and password.

Output: The user is redirected to corresponding page.

**R.3.2 Admin Login**

R.3.2.1 Selection of “Admin” button

Input: “Admin” is selected.

Output: The admin is prompted to enter email id and password.

R.3.2.2. Login

Input: Admin LogIn to the application by giving username and password.

Output: Admin logged in to the system.

**R.4 PROFILE**

R.4.1. View profile

Input: “Profile” option is selected.

Output: Displays the user’s profile with details entered during registration.

R.4.2. Update profile

Input: User updates their profile details and presses “Save profile” button.

Output: Profile is updated successfully.

Process: -The details are updated in the database.

**R.5 WORKOUTS**

**R.5.1 Search Workout**

R.5.1.1. Selection of search box

Input: “Search” option is selected.

Output: User is prompted to enter the product name.

R.5.1.2. Search

Input: User can enter the workouts details.

Output: Display relevant workouts based on search query.

Process: -The details are fetched from the database based on query.

**R.5.2 View Workout**

R.5.2.1. View

Input: login to the system.

Output: User homepage along with the workouts will be displayed.

**R.6 BMI CALCULATION**

R.6.1. Selection of “BMI” option

Input: User selects “BMI” option.

Output: User is prompted to enter height and weight.

R.8.2. Calculation

Input: Calculates BMI with provided weight and height.

Output: BMI of the user will be successfully displayed.

**2.2 NON-FUNCTIONAL REQUIREMENTS**

### 2.2.1. PERFORMANCE

The performance of system must be fast. It should be able to handle a large number of users without significant delays.

### 2.2.2 SECURITY

The user should keep their email id and password safe and should not share it with anyone. The user data such as password will be encrypted by the system which increases the security.

### 2.2.3 USABILITY

The system should be user-friendly, requiring minimal training for users to navigate and use the system.

### 2.2.4 SCALABILITY

The system should be able to accommodate future growth and increased demand without sacrificing performance.

### 2.2.5 RELIABILITY

The system should be reliable, minimizing the downtime and ensuring data integrity.

In conclusion, the requirement analysis has successfully defined the key functional and non-functional requirements that will guide the development of the app. By identifying and documenting these requirements, it has established a clear understanding of the app's essential features, which include user registration, profile management, admin management, view workouts, search workouts, and BMI calculation. Additionally, the non-functional requirements such as system performance, security, usability, reliability and scalability ensure that the app operates efficiently and meets user expectations.

**CHAPTER 3**

**SYSTEM DESIGN**

1. **SYSTEM DESIGN**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to meet the specified requirements. It serves as the blueprint for the entire system, translating the functional and non-functional requirements into a detailed framework for development. It involves creating a robust and scalable architecture that satisfies the platform's various user needs, such as those of the admin and user.

The high-level design identifies the key components of the system, providing an overview of how different parts of the application interact and function together. This design phase ensures that the system is efficient, user-friendly, and capable of handling the required operations like user registration, profile management, admin management, view workouts, search workouts, and BMI calculations.

**3.1 USECASE DIAGRAM**

The Usecase diagram is the interactions between different users and the system, capturing its functional requirements. It helps in understanding how various actors, including admin, art and user, interact with the system to perform different tasks. The admin is responsible for managing users, viewing users, and deleting users when necessary. The user plays a key role in the system by searching and viewing workouts, calculating BMI. The system also includes profile management functionalities, allowing users to register, log in, and update their profiles. By defining these interactions, the use case diagram provides a clear structure of the system’s functionality, ensuring a seamless experience for all users.



Fig.3.1.1 Use-case Diagram

**3.2 USECASE SPECIFICATION**

The Use Case Specification provides a detailed description of each use case, defining its purpose, flow, actors, and system interactions. It serves as a structured document that outlines how users interact with the system to accomplish specific tasks.

|  |
| --- |
| Use case : SignUp |
| ID: 1 |
| Brief Description:  User SignUp into the system. |
| Primary actor:  User. |
| Secondary actor:  None. |
| Precondition:  User should not have been registered to the system previously. |
| Main flow : 1. Use case starts when user selects registration.  2. User enters the details like email, password, name, mobile number.  3. Input details will be stored in the database.  4. Customer register into the system. |
| Postcondition:  Customer successfully registered. |
| Alternative flow:  Customer login. |

|  |
| --- |
| Alternative flow : Login : SignUp |
| ID : 1.1 |
| Brief description:  User login to the system. |
| Primary actor:  User. |
| Secondary actor:  None. |
| Precondition:  User have been registered to the system. |
| Alternative flow : 1. The alternative flow starts when user selects login.  2. User enters the details like User name and password.  3. User logged into the system. |
| Postcondition:  User successfully logged in. |

|  |
| --- |
| Use case : Logout |
| ID: 1 |
| Brief Description:  User LogOut from the system. |
| Primary actor:  User. |
| Secondary actor:  None. |
| Precondition:  User should have been logged into the system. |
| Main flow : 1. Use case starts when user clicks logout.  2. User logged out from the system. |
| Postcondition:  Customer successfully registered. |
| Alternative flow:  Customer login. |

|  |
| --- |
| Use case : View User |
| ID : 2 |
| Brief description :  User views the profile. |
| Primary actor :  User. |
| Secondary actor :  None |
| Precondition :  User should login. |
| Main flow : 1. Use case starts when user selects view profile.  2. User is able to view profile. |
| Postcondition :  User view profile successfully. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : User edit profile |
| ID : 3 |
| Brief description :  User edit the profile. |
| Primary actor :  User. |
| Secondary actor :  None |
| Precondition :  User should login. |
| Main flow : 1. Use case starts when user selects edit profile.  2. User can edit the details like email, age, username, weight, height.  3. Edited details will be stored in the database.  4. The database is updated. |
| Postcondition :  User successfully edited the profile. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : BMI calculation |
| ID : 4 |
| Brief description :  User calculate the BMI. |
| Primary actor :  User. |
| Secondary actor :  None |
| Precondition :  User should login. |
| Main flow : 1. Use case starts when user selects BMI calculator.  2. User enter the details like weight, height.  3. According to the details calculate the BMI.  4. The BMI is calculated. |
| Postcondition :  User successfully calculated the BMI. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : View Workout |
| ID : 5 |
| Brief description :  User view the workout. |
| Primary actor :  User. |
| Secondary actor :  None |
| Precondition :  User should login. |
| Main flow : 1. Use case starts when user logged into the system.  2. User can view the list of workouts. |
| Postcondition :  User successfully viewed the workouts. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : Search Workout |
| ID : 6 |
| Brief description :  User searches the workout. |
| Primary actor :  User. |
| Secondary actor :  None |
| Precondition :  User should login. |
| Main flow : 1. User Clicks search workout.  2. Enters the workout details. |
| Postcondition :  User successfully searches the workout. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : Admin login |
| ID: 1 |
| Brief Description:  Admin login into the system. |
| Primary actor:  Admin. |
| Secondary actor:  None. |
| Precondition :  Admin should be have been registered to the system previously. |
| Main flow : 1. Use case starts when Admin selects login.  2. Admin enters the details like username and password.  3. Admin logged into the system. |
| Postcondition :  Admin successfully logged in. |
| Alternative flow:  None. |

|  |
| --- |
| Use case : View Exercise |
| ID : 5 |
| Brief description :  Admin view Exercise. |
| Primary actor :  Admin. |
| Secondary actor :  None |
| Precondition :  Admin should login. |
| Main flow : 1. Use case starts when admin selects view Exercise.  2. Admin can view the list of Exercise. |
| Postcondition :  Admin viewed the list of Exercise. |
| Alternative flow :  None. |
|  |

|  |
| --- |
| Use case : Add Exercise |
| ID : 7 |
| Brief description :  Admin adds the Exercise. |
| Primary actor :  Admin. |
| Secondary actor :  None |
| Precondition :  Admin should login. |
| Main flow : 1. Admin Clicks Add Exercise.  2. Admin enters the Exercise details. |
| Postcondition :  Admin successfully Added the Exercise. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : Delete Exercise |
| ID : 4 |
| Brief description :  Admin Deletes the Exercise. |
| Primary actor :  Admin. |
| Secondary actor :  None |
| Precondition :  Admin should login. |
| Main flow : 1. Use case start when the Admin Clicks delete Exercise.  3. Admin selects the Exercise which want to be delete.  2. Admin deletes the selected Exercise. |
| Postcondition :  Admin successfully deleted the selected Exercise. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : View User |
| ID : 3 |
| Brief description :  Admin view the user. |
| Primary actor :  Admin. |
| Secondary actor :  None |
| Precondition :  Admin should login. |
| Main flow : 1. Use case start when the Admin Clicks view user.  3. Admin view the list of users. |
| Postcondition :  Admin successfully viewed the users. |
| Alternative flow :  None. |

|  |
| --- |
| Use case : Delete User |
| ID : 3 |
| Brief description :  Admin delete the user. |
| Primary actor :  Admin. |
| Secondary actor :  None |
| Precondition :  Admin should login. |
| Main flow : 1. Use case start when the Admin Clicks delete user.  3. Admin selects the user which want to be delete.  2. Admin deletes the selected user. |
| Postcondition :  Admin successfully deleted the users. |
| Alternative flow :  None. |

**3.3 CLASS DIAGRAM**

The class diagram represents the key entities and their relationships within the system. The main classes include Admin, Artist, Customer, Artwork, Order, and Feedback.

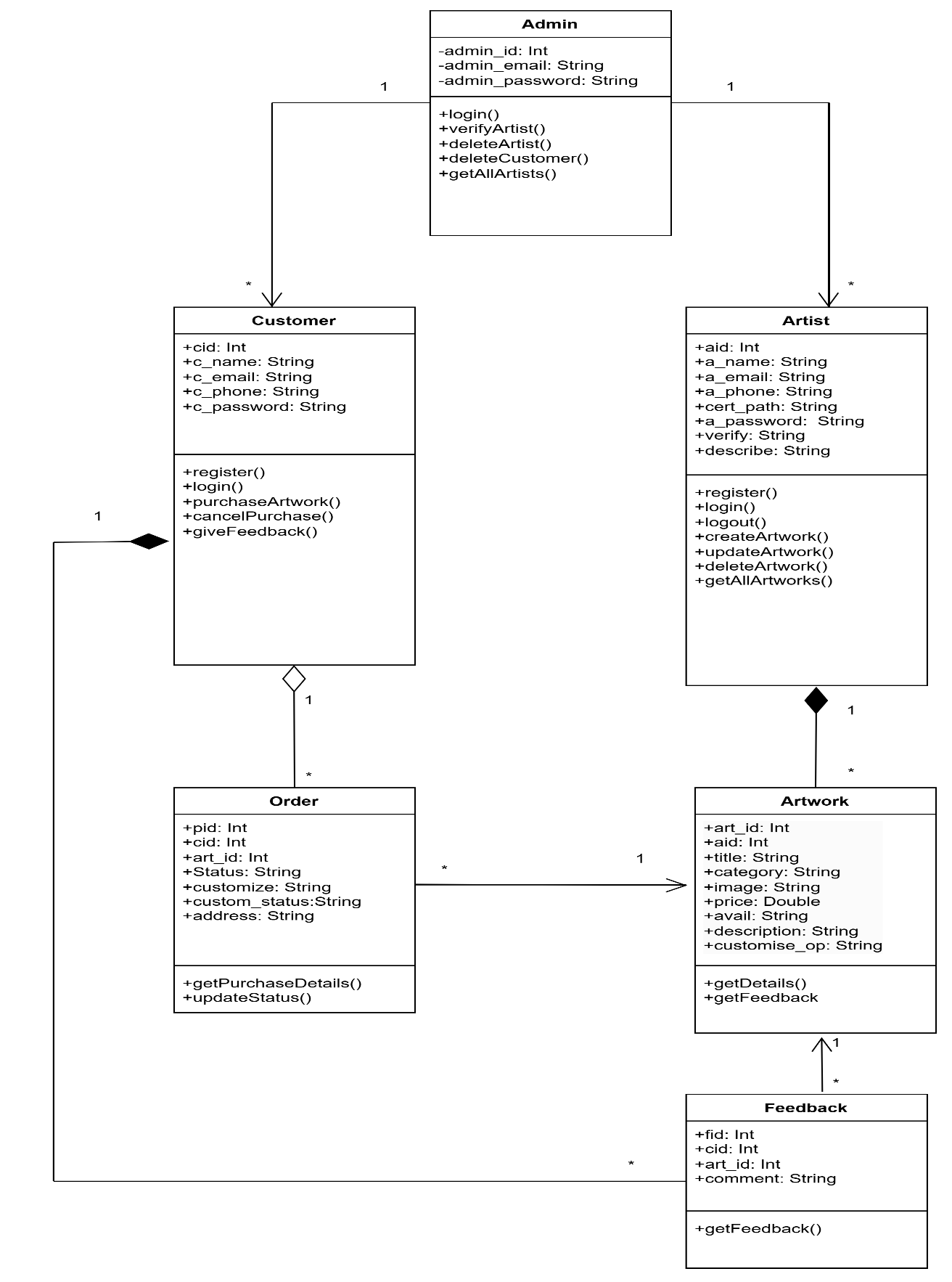
****

Fig.3.3.1 Class Diagram

**3.4 FLOW DIAGRAM**

A Flow diagram is a way of representing a flow of data through a process or a system (usually an information system). It also provides information about the outputs and inputs of each entity and the process itself.

A flow diagram has no control flow — there are no decision rules and no loops.

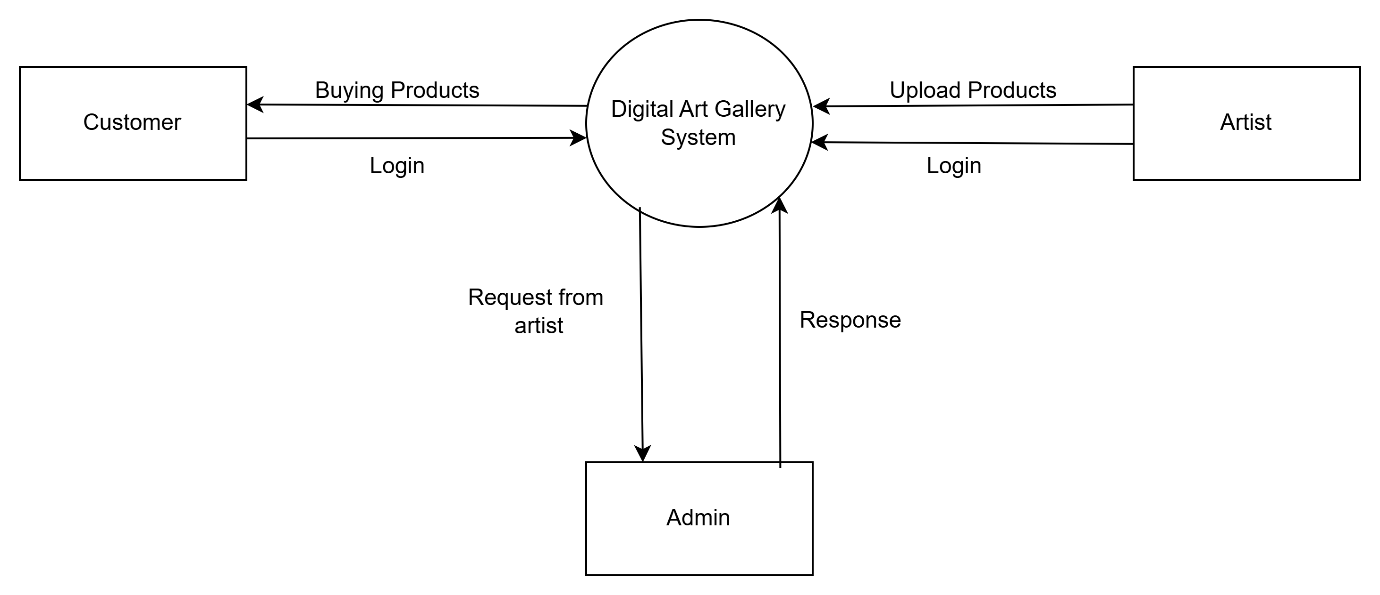


Fig.3.4.1 DFD Level 0

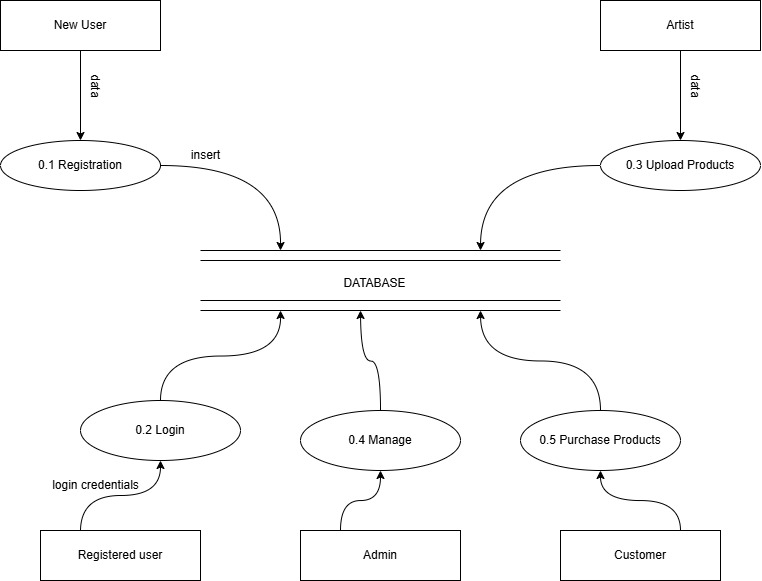


Fig.3.4.2 DFD Level 1

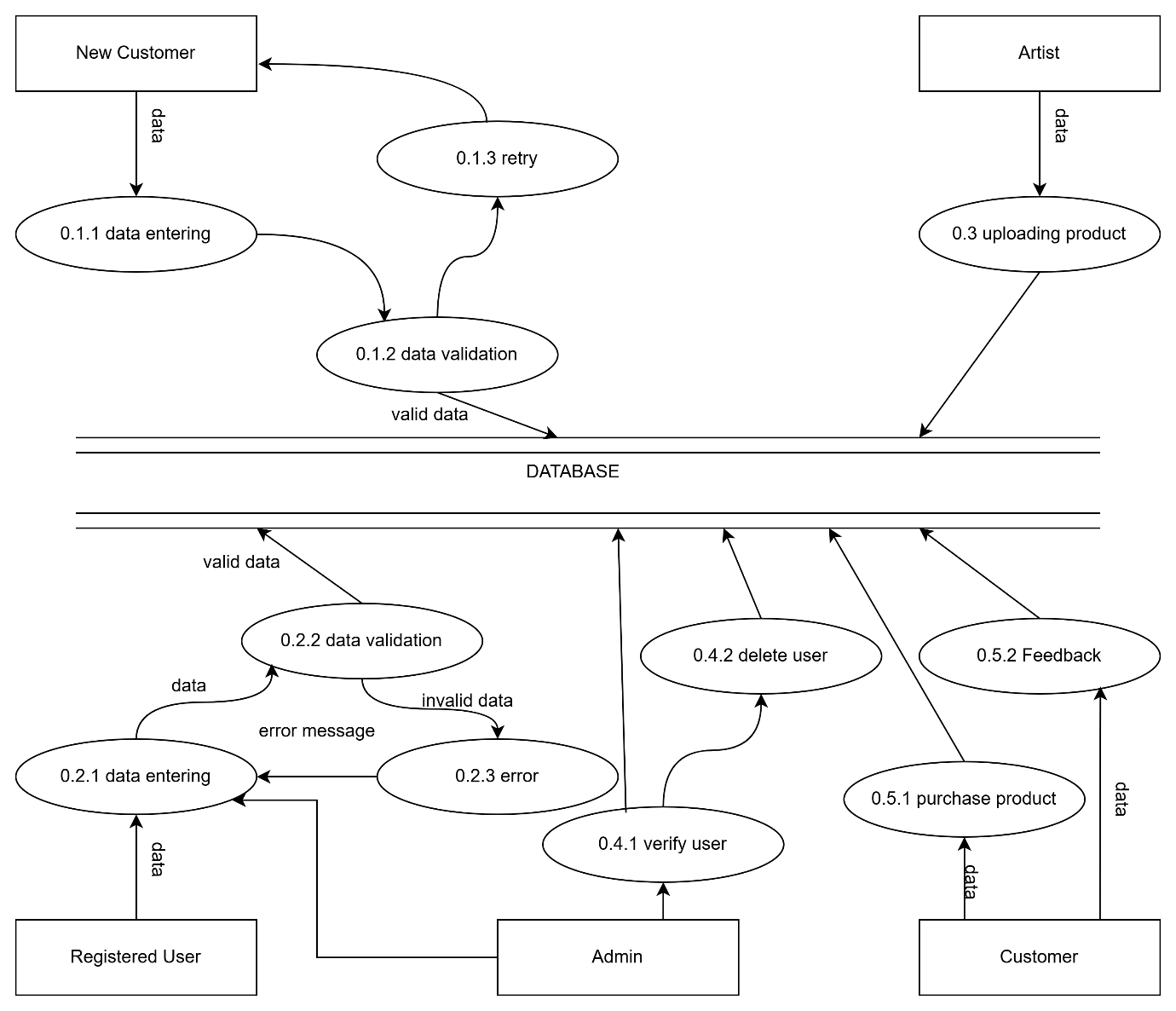


Fig.3.4.3 DFD Level 2

**3.5 ENTITY RELATIONSHIP DIAGRAM**

The Entity Relationship Model (ER Model) provides a conceptual representation of the database structure, defining how different entities interact. The ER diagram consists of six key entities: Admin, Artist, Customer, Artwork, Order, and Feedback. Each entity contains specific attributes that store relevant information, and relationships between them ensure efficient data management.

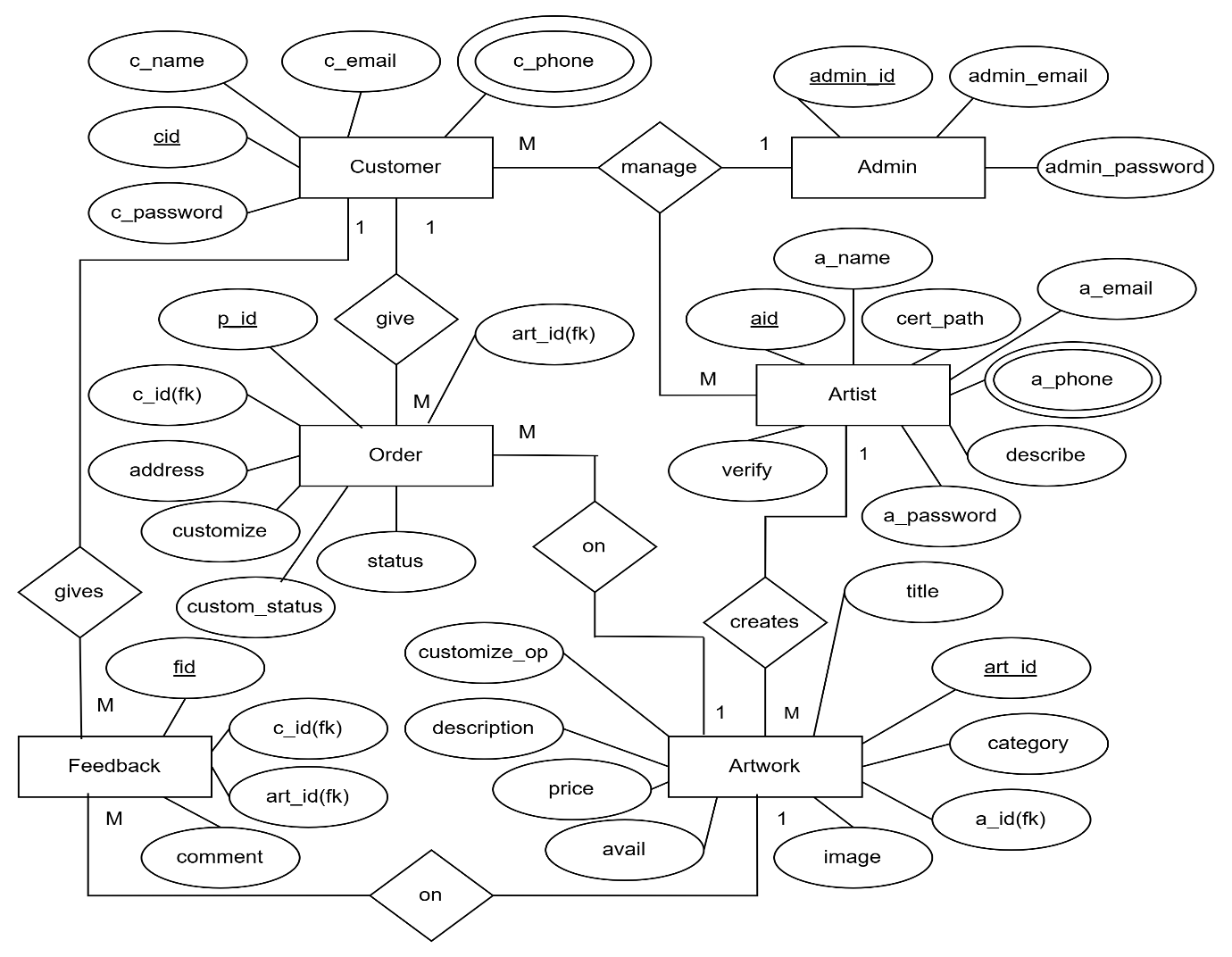
The admin entity includes attributes like admin\_id (primary key), along with admin\_email and admin\_password. The artist entity includes attributes like aid (primary key), a\_name, a\_email, a\_phone, cert\_path, a\_password, verify, and describe.The customer entity includes attributes like cid(primary key), along with c\_name, c\_email, c\_phone, and c\_password. The Artwork entity stores details of artworks uploaded by artists, including art\_id (primary key), aid (foreign key), title, category, image, price, avail, description, and customize\_op. The order entity handles artwork purchases and customization requests. It consists of pid (primary key), cid (foreign key), art\_id (foreign key), status, customize, custom\_status, and address.Lastly, the Feedback entity allows customers to share reviews about purchased artwork. It includes fid (primary key), cid (foreign key), art\_id (foreign key), and comment. 

Fig.3.5.1 ER Diagram

**3.6 TABLE DESIGN**

Table design refers to the organization of data in a structured manner according to a database model. The tables in our app are in Third Normal Form (3NF) to minimize redundancy and avoid anomalies during insertion, deletion, and updates. Normalization is essential to reduce data duplication and ensure data integrity. It ensures that all non-key attributes are directly dependent on the primary key and not on any other non-key attributes, thus eliminating transitive dependency. For example, in the feedback table, the attributes include fid (primary key), cid (foreign key to customer), art\_id (foreign key to artwork), and comment. Each attribute is directly related to fid and does not depend on any other non-key attributes, ensuring the table adheres to 3NF.

ADMIN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | NAME | DATA TYPE | NULL | DEFAULT | KEY |
| 1 | admin\_id | int | No | None | Primary |
| 2 | admin\_email | text | No | None |  |
| 3 | admin\_password | text | No | None |  |

USERS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | NAME | DATA TYPE | NULL | DEFAULT | KEY |
| 1 | ID | int | No | None | Primary |
| 2 | username | text | No | None |  |
| 3 | age | text | No | None |  |
| 4 | email | text | No | None |  |
| 5 | password | text | No | None |  |
| 6 | gender | text | No | None |  |
| 7 | bodyType | text | No | None |  |
| 8 | height | real | Yes | None |  |
| 9 | weight | real |  |  |  |

WORKOUTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | NAME | DATA TYPE | NULL | DEFAULT | KEY |
| 1 | Workout\_id | int | No | None | Primary |
| 2 | Workout\_name | text | No | None |  |

EXCRCISES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | NAME | DATA TYPE | NULL | DEFAULT | KEY |
| 1 | exercise\_id | int | No | None | Primary |
| 2 | exercise\_name | text | No | None | Foreign |
| 3 | exercise\_image | text | No | None |  |
| 4 | exercise\_description | text | No | None |  |
| 5 | exercise\_video\_url | text | No | None |  |
| 6 | Workout\_category | text | No | None |  |

ORDER

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | NAME | DATA TYPE | NULL | DEFAULT | KEY |
| 1 | pid | int | No | None | Primary |
| 2 | cid | int | No | None | Foreign |
| 3 | art\_id | int | No | None | Foreign |
| 4 | status | text | No | None |  |
| 5 | custom\_status | text | No | Pending |  |
| 6 | customize | text | Yes | None |  |
| 7 | address | text | No | None |  |
|  |  |  |  |  |  |

FEEDBACK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | NAME | DATA TYPE | NULL | DEFAULT | KEY |
| 1 | fid | int | No | None | Primary |
| 2 | cid | int | No | None | Foreign |
| 3 | art\_id | int | No | None | Foreign |
| 4 | comment | text | No | None |  |

In conclusion, we designed the usecase diagram to define user interactions, while the class diagram outlined the structural relationships between entities. The flow diagram detailed the process flow, ensuring logical execution of operations. The ER diagram represented the database schema, and the table design followed 3NF principles to minimize redundancy. Together, these diagrams provide a well-structured foundation for the system's implementation.

**CHAPTER 4**

**IMPLEMENTATION**

1. **IMPLEMENTATION**

The implementation phase focuses on transforming the system design into a functional application. It begins with defining the system architecture, which outlines the overall structure and interactions between different components. Various tools are utilized for development, categorized into front-end and back-end technologies, each playing a crucial role in ensuring a seamless user experience and efficient data handling. The system is divided into multiple modules, including the admin module, artist module and customer module, each responsible for specific functionalities.

**4.1 SYSTEM ARCHITECTURE**

System Architecture defines the overall structure and behavior of the app, outlining how different components interact to achieve system functionality. It consists of various layers, including the User Interface, Application Logic, and Database, ensuring smooth data flow and user interactions.

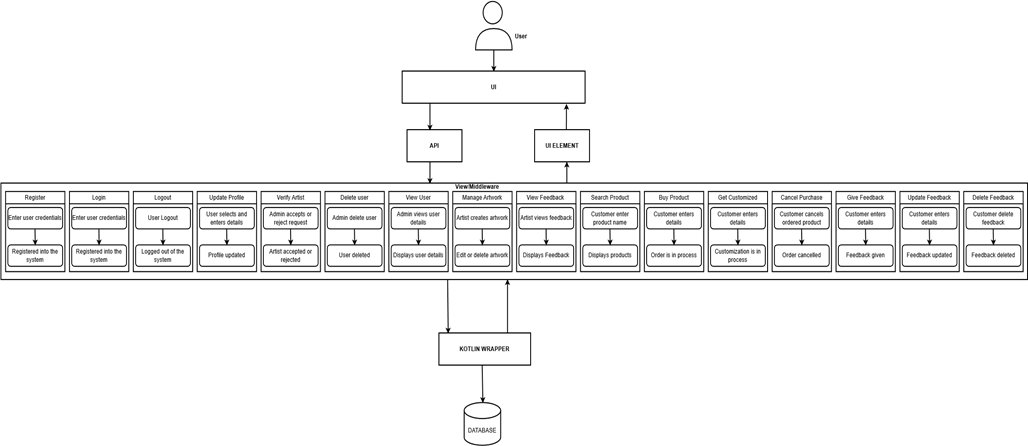


Fig.4.1.1 System Architecture

**4.2 TOOLS USED**

The app is built using various tools that help in front-end and back-end development, database management, and overall system functionality. These tools ensure smooth performance, user-friendly interaction, and efficient data handling.

**4.2.1 FRONTEND TOOLS**

(i) XML**:** XML (Extensible Markup Language) is used to define and structure data in a hierarchical format. In Android development, XML is primarily used for designing user interfaces by defining layouts, styles, and attributes. It is lightweight, scalable, and easy to read, making it an efficient choice for UI implementation. Unlike programming languages, XML only contains structured tags and does not include logic or functionality.

**4.2.2 BACKEND TOOLS**

(i) Kotlin: Kotlin is the backend programming language used in Android Studio for developing robust applications. It provides essential libraries and frameworks to handle data processing, user interactions, and business logic efficiently. With strong community support, enhanced security, and compatibility with modern development practices, Kotlin enables developers to create responsive and scalable Android applications.

(ii) SQLite: SQLite is a lightweight relational database management system embedded within Android applications. It supports a wide range of SQL queries, allowing efficient data storage and retrieval. SQLite ensures data integrity, supports encryption for secure storage, and provides built-in security features to protect sensitive user information. Since it does not require a separate database server, it is highly optimized for mobile applications.

**4.3 MODULES**

**4.3.1 ADMIN MODULE**

The Admin module serves as the central control system of the platform, ensuring smooth operations and maintaining the integrity of the digital art gallery. Admins are responsible for verifying artists' by reviewing qualification certificates before granting access to the platform. They can also manage user accounts by deleting fraudulent or inactive users.

**4.3.2 ARTIST MODULE**

The Artist module empowers creators to showcase and sell their artwork on the platform. Artists can register, submit their qualification certificates for verification, and manage their profiles. Once verified, they can upload, update, or delete their artworks,and enable customization options for customers. The module also allows artists to view customer orders, approve or reject customization requests, and respond to feedback.

**4.3.3 CUSTOMER MODULE**

The Customer module is designed to provide a seamless experience for users interested in browsing and purchasing artwork. Customers can register, log in, and explore various artworks listed by artists. This module enables them to place orders, customize artwork based on preferences, track their purchases, and provide feedback after receiving their orders.

**CHAPTER 5**

**CONCLUSION AND FUTURE SCOPE**

1. **CONCLUSION AND FUTURE SCOPE**

**5.1 CONCLUSION**

The Digital Art Gallery App successfully provides a structured and user-friendly platform for artists to showcase their digital artwork and for customers to browse, purchase, and customize products. The system incorporates key functionalities such as user registration, profile management, artwork listing, real-time product availability updates, and a feedback mechanism. The admin role ensures platform authenticity by verifying artist credentials and managing users. Through a streamlined and intuitive interface, the app enhances user experience, making digital art more accessible and interactive.

One of the major advantages of the platform is its structured verification process, which ensures that only qualified artists can sell their artwork. The customization feature allows customers to personalize their purchases, adding a unique touch to their selected pieces. Additionally, the system also ensures accurate availability of artwork.The inclusion of a feedback system fosters transparency, helping both artists and customers make informed decisions.

Despite its benefits, the app has certain limitations. The reliance on admin verification may introduce delays in artist approvals.

**5.2 FUTURE SCOPE**

The Digital Art Gallery App has significant potential for future enhancements that can improve user experience, expand accessibility, and optimize platform functionality. Several key improvements can be explored to enhance the app’s capabilities:

**(i)** **AI-Based Artwork Recommendations**: Implementing artificial intelligence to analyze user preferences and browsing history to provide personalized artwork recommendations.

**(ii) Collaboration and Auction Features:** Enabling artists to collaborate on artwork and introducing an auction system for limited-edition or exclusive pieces.

**(iii) Social Media Integration:** Allowing artists to share their work directly on social media platforms to increase visibility and engagement.

**(iv) Enhanced Search and Filter Options**: Improving search functionalities with AI-powered filters to help customers find artwork based on style, theme, and artist preferences.

**CHAPTER 6**

**REFERENCES**

1. **REFERENCES**

**Web References:**

**CHAPTER 7**

**APPENDICES**

1. **APPENDICES**