

**OOP SEMESTER PROJECT**

**INSTRUCTOR: MAAM HAREEM KIBRIYA**

|  |  |  |
| --- | --- | --- |
| Members Name: |  | ANSHARA ZAHOOR, AZLAN ZAFAR |
| Registration No: |  | 241802, 241820 |
| Department: |  | COMPUTER SCIENCE |
| Submission Date: |  | 1/06/25 |

**AIR UNIVERSI**TY ISLAMABAD

Project Name

Freelance Platform (OOP-based)

Introduction:

**This project is a console-based Freelance Platform built using Object-Oriented Programming (OOP) principles in C++ and a dummy front-end with java. It simulates an online marketplace where clients can post freelance gigs, and freelancers can apply to them. The system supports essential features such as user registration, login, gig management, applications, chat simulation, payments, and reviews.**

Designed for simplicity and clarity, this platform uses text files for persistent data storage and demonstrates real-world application of inheritance, polymorphism, abstraction, and file handling in C++.

****Problem Statement & Solution****

**Problem:  
Freelancers often struggle to find clients, while clients find it difficult to locate trustworthy service providers. Additionally, there is no centralized system for managing gig applications, payments, or reviews in a seamless way.**

**Solution:  
This Freelance Platform offers a minimal but functional solution:**

* Freelancers can post gigs.
* Clients can search and apply to gigs.
* Secure login and registration using text files.
* Review and payment tracking.
* Chat interaction after application.
* File-based record management system ensures persistent data storage.

****Classes and Their Purpose****

| **Class** | **Purpose** |
| --- | --- |
| Auth | Manages user login and registration |
| User | Abstract base class for Client and Freelancer |
| Client | Inherits from User, allows searching, applying for gigs, payments, reviews |
| Freelancer | Inherits from User, allows posting gigs, viewing applications, payments, and reviews |
| Gig | Handles gig posting, viewing, and searching |
| Review | Manages submission and viewing of reviews |
| Payment | Handles making and viewing payments |
| PaymentManager | Allows updating payment statuses |
| ApplicationManager | Views gig applications submitted by clients |
| Chat | Simulates chat between freelancers and clients post-application |
| GigFilter | Filters gigs based on a price range |

****Packages / Files Used****

**Since this is a simple C++ project, there are no external libraries or package managers used. It relies on the C++ Standard Library:**

* iostream, fstream, sstream, string

Text files used:

* users.txt — stores registered users
* gigs.txt — stores posted gigs
* applications.txt — stores gig applications
* reviews.txt — stores client reviews
* payments.txt — stores payment records

**Flow of Program (main function)**

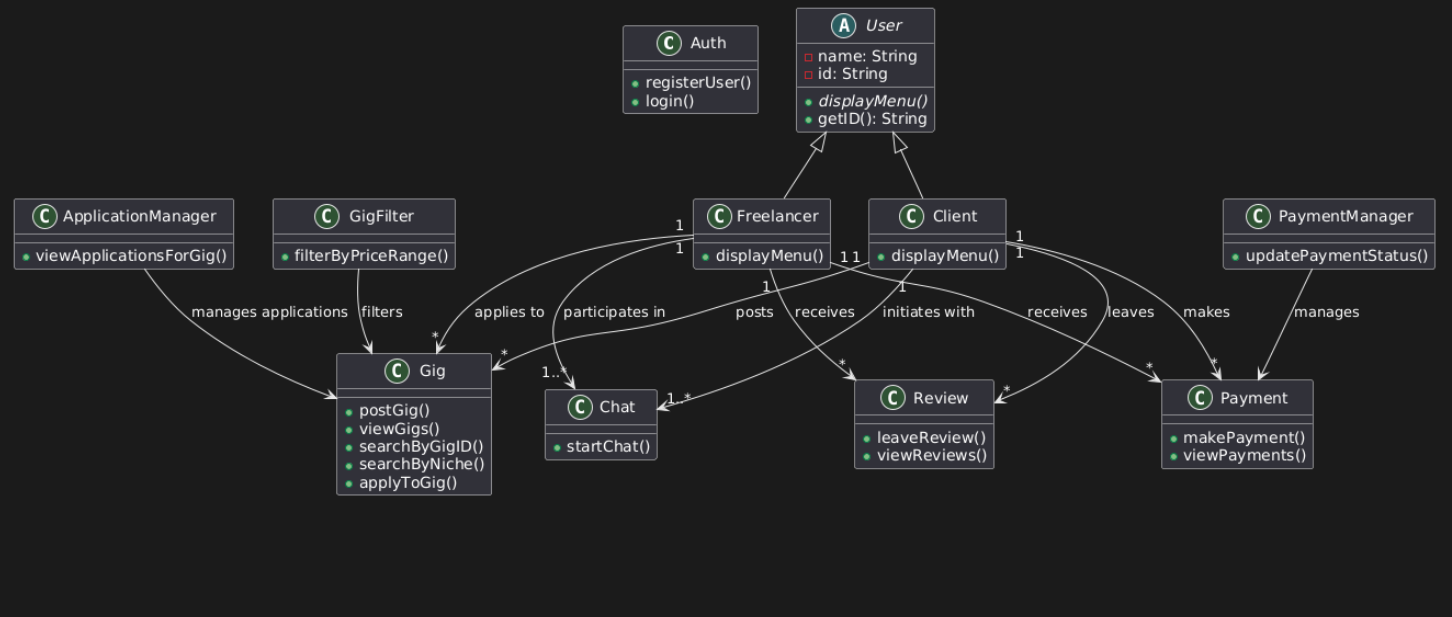
Shows Main Menu:

* Register
* Login
* Exit

On Login:

* Asks for role: Client or Freelancer.
* Authenticates using users.txt.
* If valid, creates either a Client or Freelancer object dynamically.
* Based on role, displays a custom menu:
* Client::displayMenu() — options to view/apply/pay/review.
* Freelancer::displayMenu() — options to post/view gigs, see reviews/payments.

UML Diagram

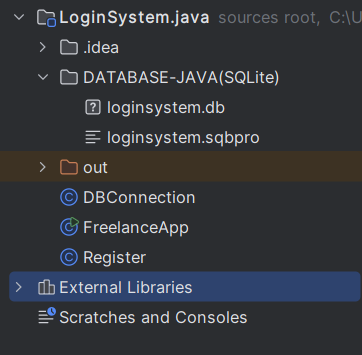


GUI

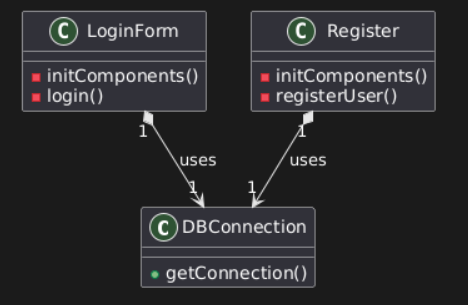
**This project contains a simple front-end GUI but functional Login and Registration form developed using Java Swing for GUI, SQLite for database integration, and JAVA Database Connectivity for connection. This form allows users to register with their credentials and later log in using those credentials, which are stored securely in the local database. The goal is to demonstrate the integration between Java applications and a lightweight database in a user-friendly environment.**

Packages and Classes:

| **Package** | **Class Name** | **Description** |  |
| --- | --- | --- | --- |
| LoginSystem.java | | LoginForm.java | |Main GUI and event handler for login |  |
| LoginSystem.java | | Register.java | |Handles user registration and data storage |  |
| LoginSystem.java | |DBConnection.java | |Provides methods to connect to SQLite DB |  |
|  |  |  |  |

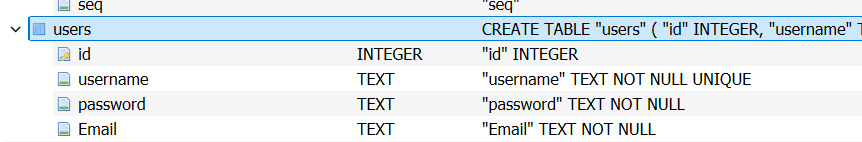


UML Diagram



SQLite DATABASE:

TABLE:



DATA:

A screenshot of a computer

AI-generated content may be incorrect.