Name: Muhammad Aun

A logo of a company

Description automatically generated

Reg#: 22-NTU-CS-1280

BSSE-A-5

SDA Project: Freelancer App

Sir. Umar

**Overview:**

Freelancer app is a platform, as we can understand through its name “Freelancer”, designed for the selling and buying the professional services. It is works as a digital workplace where the business client can post a job and freelancers can show there expertise and can bid on the jobs and then can give the description about their bids and the time under which he/her can do that job. The client chooses any of the bids of the freelancers of like it and that job will be assigned to that freelancer (just like winning the tenders).

**Scope:**

Freelancer has a very wider scope all over the world to connect the businesses with the skilled and expert freelancers. It covers many skills to serve and buy i.e. programming, graphics design, writing, marketing, video editing. It supports the multiple languages and currencies. It can access all over the world and can expand with new features by simplifying how business and freelancers work together. It supports flexible and remote working opportunities.

**Brief highlight:**

Founded in: 2009

Headquarter in: Sydney Australia

Founder & CEO: Matt Barrie

Supported Platforms: Android, IOS, Web, and Desktop (Windows, Linux, and mac)

Target Users: Individual Freelancers, Businesses looking work, Clients who wants the services

**Core Functionalities of Freelancer.com:**

**User Management:**

* + - User Registration for Clients and Freelancers.
    - Profile creation involves Resume, skills, portfolio, Certifications.
    - Users’ roles (Clients who post jobs, Freelancers who bid on them).
    - Security and Authentication via email verification, two factor authentication.

**Project & Job Management:**

* Posting Job: Clients can post projects with detailed description, budget, and timeline.
* Bidding: Freelancers add proposals against the projects and also quote Price for that job.
* Awarding Project: After reviewing the bids, Clients assign the projects to freelancers.
* Milestones: For phased payment clients can create and approve milestones.
* Progress Tracking: There are available updates on the project and timelines to show the progress of a the project.

**Communication & Collaboration:**

* Real-time chat is available between clients and freelancers.
* Files related to the projects can be uploaded and shared.
* Notifications for project updates, messages and milestones are another feature of this platform.

**Payment:**

* Secured payment through Escrow system
* Integrated with Stripe, PayPal, credit/debit card.
* Saves all of the projects costs and payments.

**Browsing & Discovery:**

* Allows clients to filter the freelancers by their skills, rating and location.
* Freelancers can search jobs using categories and different keywords.
* Personalize the recommendations for the freelancers according to their skills and expertise.

**Rating, feedback & Reviews:**

* After project Completion the client and freelancer both can give rating and feedbacks.
* They can display reviews on their profiles for trustability and credibility.

**Premium Features:**

* There are subscription plans available for higher bidding limits and boosting profile.
* Visible priority for client job posting.
* There are tests for freelancers to validate their expertise and get certifications.

**Multiple languages and Currency Support:**

* It supports multiple languages and currencies all over the world.

**Analytics & Insights:**

* Allow clients to track the projects costs and performance of freelancers.
* Freelancers can monitor their bids, statistics of their projects and earnings.

**Modules in Freelancer.com:**

**User Management & Authentication Module:**

* Users Registration i.e.; Client or Freelancer.
* Login, Logout with password management.
* Two-Factor verification and social login.
* Creating profile and updating.
* Users’ roles.

**Job Managing Module**

* Posting jobs via Clients.
* Listing & searching jobs via freelancers.
* Job categorization and filtration.
* Posting Featured jobs.

**Biddings and Proposals Module:**

* Bidding System for freelancers.
* Submitting Proposals with timelines and prices.
* Client Reviews and Approvals for Biddings.

**Project Managing Module:**

* Assigning Projects to freelancers and updating statuses.
* Milestones creation and their approvals.
* Tracking progress and notifications.

**Payment Management Module:**

* Escrow system for payments.
* Integration with Payment gateways i.e.; PayPal, Stripe, etc.
* Payments based on milestones.
* History of transactions and automated invoice generation.

**Communication & collaboration Module:**

* Real-time messaging system.
* Attachments for project i.e.; sharing files.
* Notifications for messages, bids, milestones.

**Review & rating Module:**

* Rating system on project completion both for freelancer and client.
* Displaying reviews and feedbacks on the profiles.

**Admin & Moderation Modules:**

* Moderation for users and content.
* Conflict management and dispute Resolution.
* Dashboard to monitor analytics of platform activities.

**Premium features & Subscription Module:**

* Subscription plans for freelancers for higher bidding limits and boosting profile.
* Posting featured jobs for the clients.
* Certifications for freelancers’ skills

**Multiple languages and currency Modules:**

* Localization of languages for global users.
* Support & conversions of currency.

**Customer help center & support Module:**

* 24/7 customer support vie email, and live chat.
* FAQs.
* Ticketing system to resolve the issues.

**Algorithms of Freelancer.com:**

1. User Registration Algorithm

Purpose: To register a new user (client or freelancer).

Input: User details (name, email, password, role).

Output: Successful registration or error message.

plaintext

Step 1: Ask the user to provide their details (name, email, password, and role).

Step 2: Check the provided details for validity (e.g., confirm the email format is correct and the password meets the required criteria).

Step 3: Verify if the email is already registered in the system.

If it is, show an error message: "Email already in use."

If it isn't, move on to the next step.

Step 4: Store the user's details in the database.

Step 5: Send a confirmation email to the user containing a verification link.

Step 6: Let the user know that registration was successful and that they need to verify their email.

2. Job Posting Algorithm

Purpose: To enable clients to post jobs.

Input: Job details (title, description, budget, deadline).

Output: Job is posted or an error message is displayed.

plaintext

Step 1: Ask the client to enter the job details (title, description, budget, and deadline).

Step 2: Validate the job details provided.

If the details are invalid, show an error message: "Please provide valid information."

If the details are valid, proceed to the next step.

Step 3: Save the job details in the system.

Step 4: Notify the admin to review the job post.

Step 5: The admin evaluates the job post.

If approved, make the job post visible to freelancers.

If rejected, inform the client with the reason for rejection.

Step 6: Notify the client that their job post has been successfully published (if approved).

3. Freelancer Bidding Algorithm

Purpose: To enable freelancers to bid on jobs.

Input: Freelancer details (bid amount, time required, proposal).

Output: Bid is submitted or an error message is displayed.

Step 1: Show available jobs to the freelancer.

Step 2: Allow the freelancer to choose a job and enter their bid details (amount, time, and proposal).

Step 3: Validate the bid details.

If invalid, display an error message: "Bid details are incorrect. Please try again."

If valid, move to the next step.

Step 4: Save the bid details in the system under the chosen job.

Step 5: Notify the client about the new bid submission.

Step 6: Inform the freelancer that their bid has been submitted successfully.

4. Bid Acceptance Algorithm

Purpose: To enable clients to accept a freelancer's bid.

Input: Selected bid.

Output: The project is assigned to the freelancer.

Step 1: Display all submitted bids for the job to the client.

Step 2: Allow the client to review and choose a bid.

Step 3: Confirm the client's choice.

If the client confirms, assign the project to the freelancer.

If the client cancels, return to the bid review step.

Step 4: Update the project's status in the system to "Assigned."

Step 5: Notify the chosen freelancer about the project assignment.

Step 6: Inform other freelancers that their bids were not accepted.

5. Payment Processing Algorithm

Purpose: To handle milestone payments between clients and freelancers.

Input: Payment details (amount, milestone description).

Output: Payment is processed or an error message is shown.

Step 1: Allow the client to create a milestone payment.

Step 2: Validate the milestone details (amount and description).

If invalid, display an error message: "Invalid milestone details."

If valid, move to the next step.

Step 3: Connect to the payment gateway.

If payment fails, notify the client: "Payment could not be processed. Please try again."

If payment succeeds, proceed to the next step.

Step 4: Record the milestone payment in the system.

Step 5: Notify the freelancer about the secured milestone payment.

6. Dispute Resolution Algorithm

Purpose: To address conflicts between clients and freelancers.

Input: Details of the dispute (reason, supporting documents).

Output: The dispute is either resolved or escalated.

plaintext

Step 1: Allow the user (either client or freelancer) to initiate a dispute by providing the necessary details and supporting documents.

Step 2: Validate the provided dispute details.

If the details are invalid, show an error message: "Invalid dispute details. Please check your submission."

If the details are valid, move on to the next step.

Step 3: Forward the dispute to the admin for evaluation.

Step 4: The admin reviews the dispute and may request additional documents if needed.

Step 5: The admin investigates and resolves the dispute.

If resolved, notify both parties of the outcome.

If not resolved, escalate the dispute to a higher authority.

Step 6: Record the dispute resolution status in the system.

7. User Review Algorithm

Purpose: To enable users to review one another after completing a project.

Input: Review details (rating, comments).

Output: The review is submitted or an error message is shown.

plaintext

Step 1: Allow the user (client or freelancer) to choose a completed project.

Step 2: Prompt the user to provide review details (rating and comments).

Step 3: Validate the review details.

If the details are invalid, display an error message: "Please provide valid review details."

If the details are valid, proceed to the next step.

Step 4: Save the review in the system under the profile of the user being reviewed.

Step 5: Notify the reviewed user about the new review.

Step 6: Update the average rating of the reviewed user based on the new review.

**Diagrams**

**Use Case Diagram:**

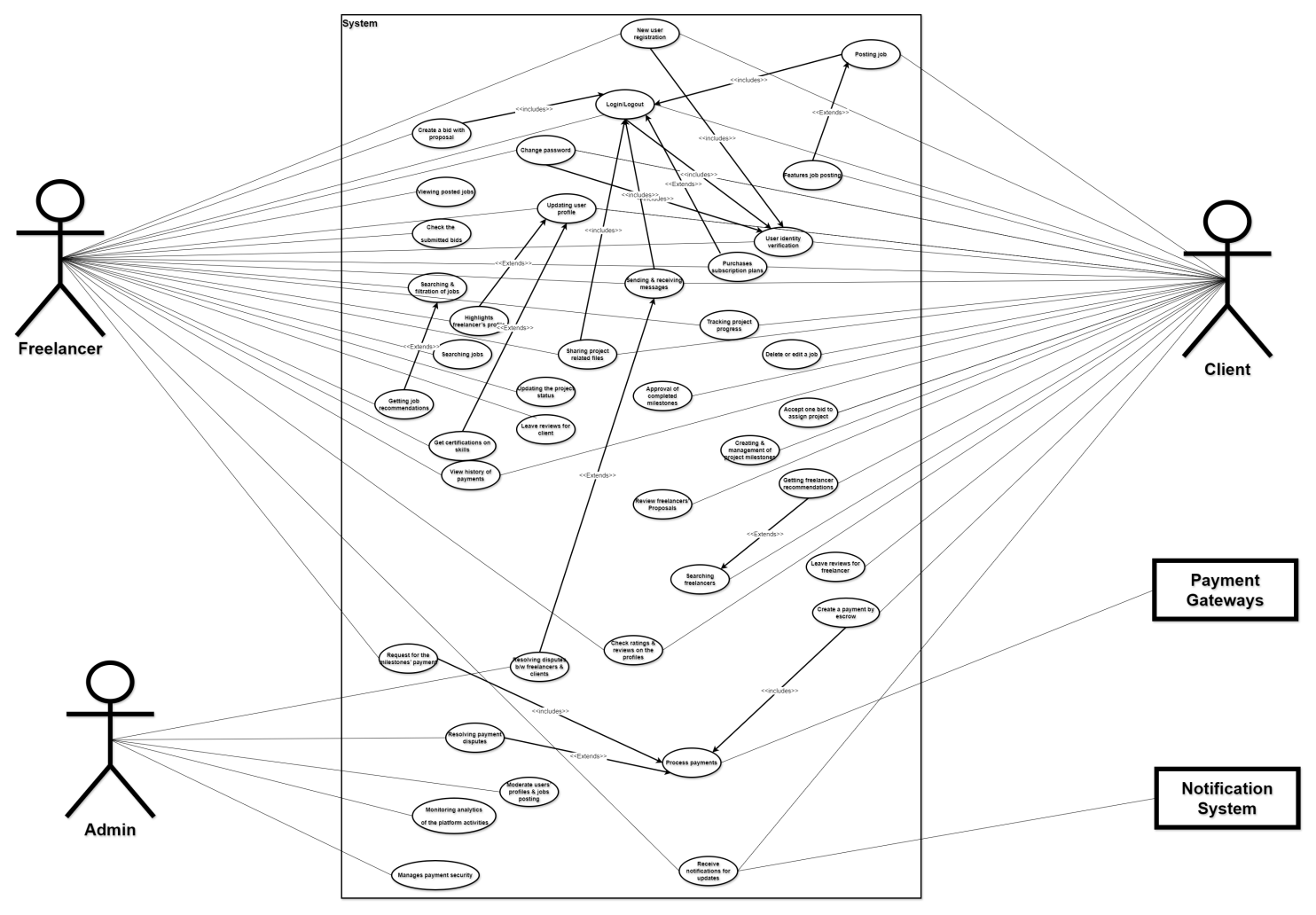
**Actors Identification**

* Client
* Freelancer
* Admin
* Payment Gateways
* Notification System

**Use Cases**

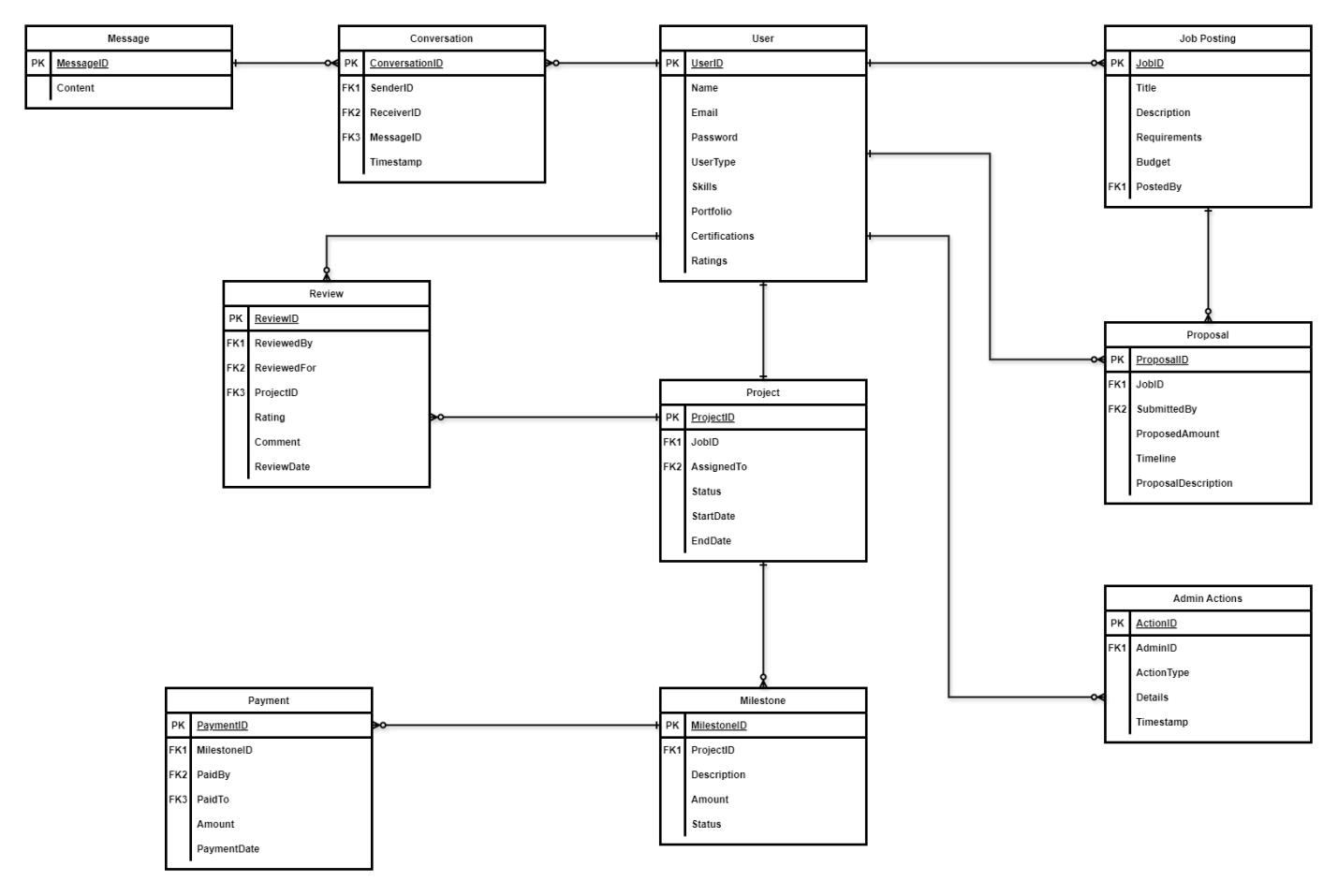
* New user registration
* Login/Logout
* Updating user profile
* User identity verification
* Change password
* Posting job
* Viewing posted jobs
* Searching & filtration of jobs
* Delete or edit a job
* Create a bid with proposal
* Check the submitted bids
* Review freelancers’ Proposals
* Accept one bid to assign project
* Creating & management of project milestones
* Updating the project status
* Approval of completed milestones
* Tracking project progress
* Create a payment by escrow
* Request for the milestones’ payment
* Process payments
* View history of payments
* Resolving payment disputes
* Sending & receiving messages
* Sharing project related files
* Receive notifications for further updates
* Searching freelancers
* Searching jobs
* Getting job recommendations
* Getting freelancer recommendations
* Leave reviews for freelancer
* Leave reviews for client
* Check ratings & reviews on the profiles
* Moderate users’ profiles & jobs posting
* Resolving disputes b/w freelancers & clients
* Monitoring analytics of the platform activities
* Manages payment security
* Purchases subscription plans
* Features job posting
* Highlights freelancer’s profile
* Get certifications on skills

**Diagram**



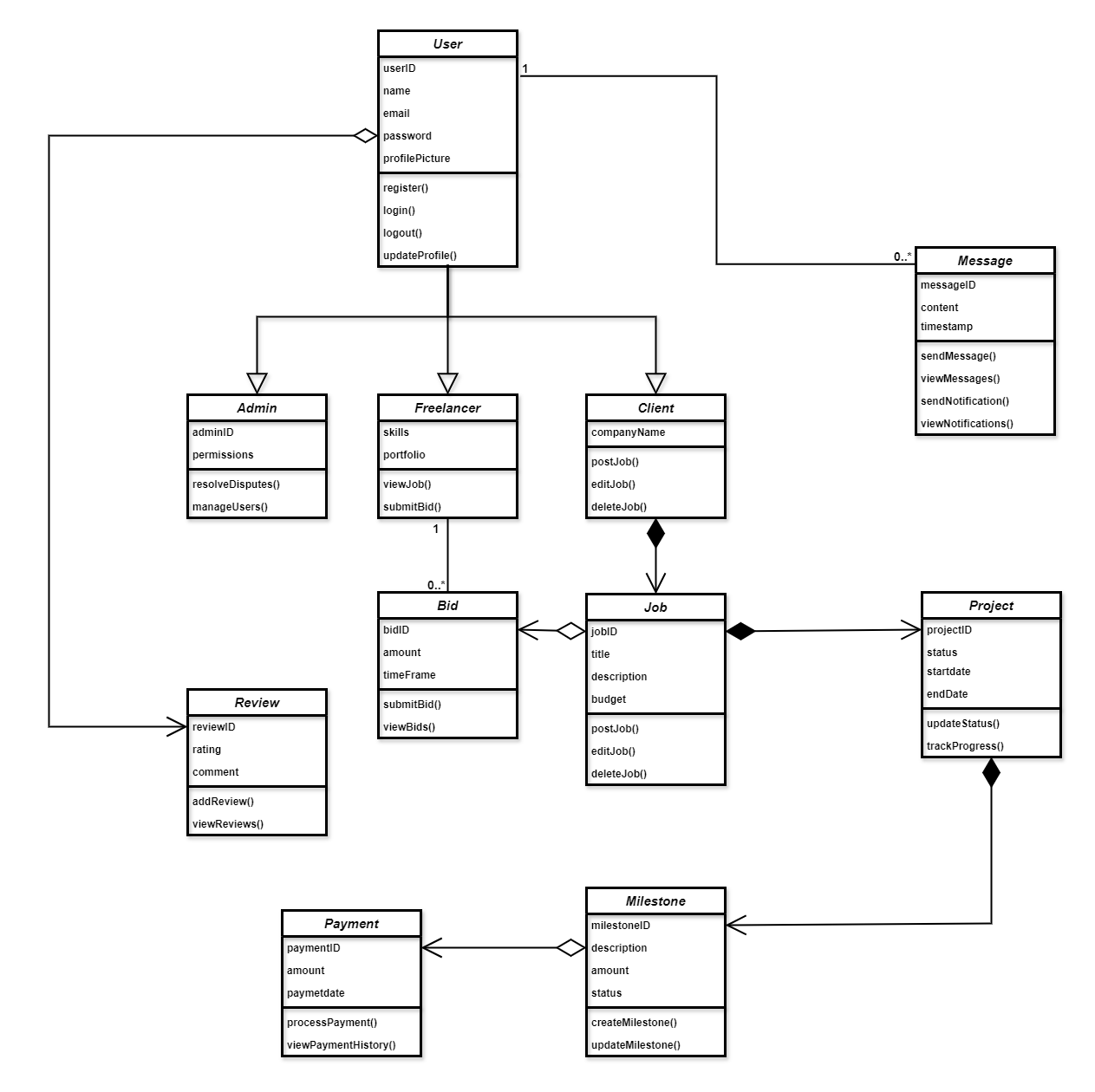
**Entity Relationship Diagram (ERD):**

Following is the Entity Relationship Diagram (ERD) for the Database creation for this project “Freelancer” app.

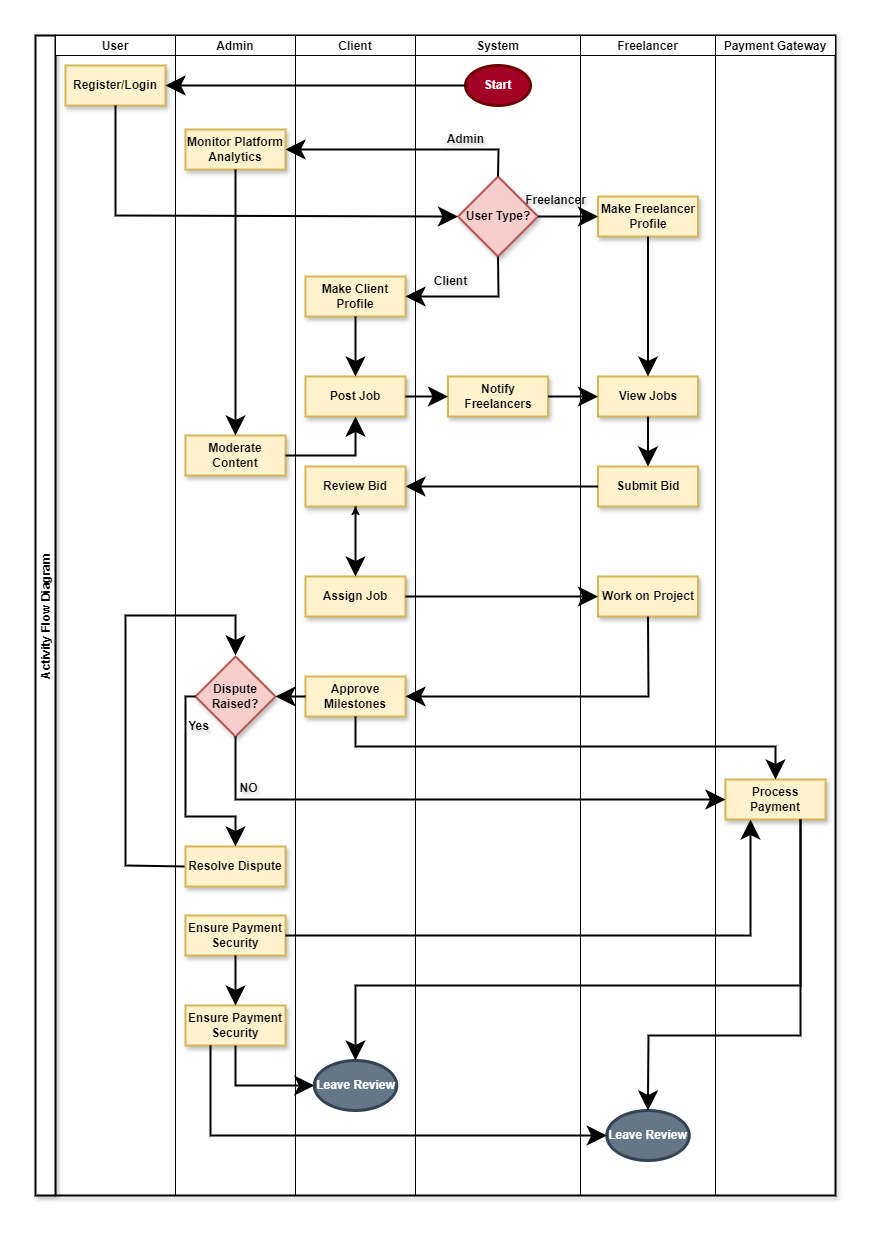


**Class Diagram:**

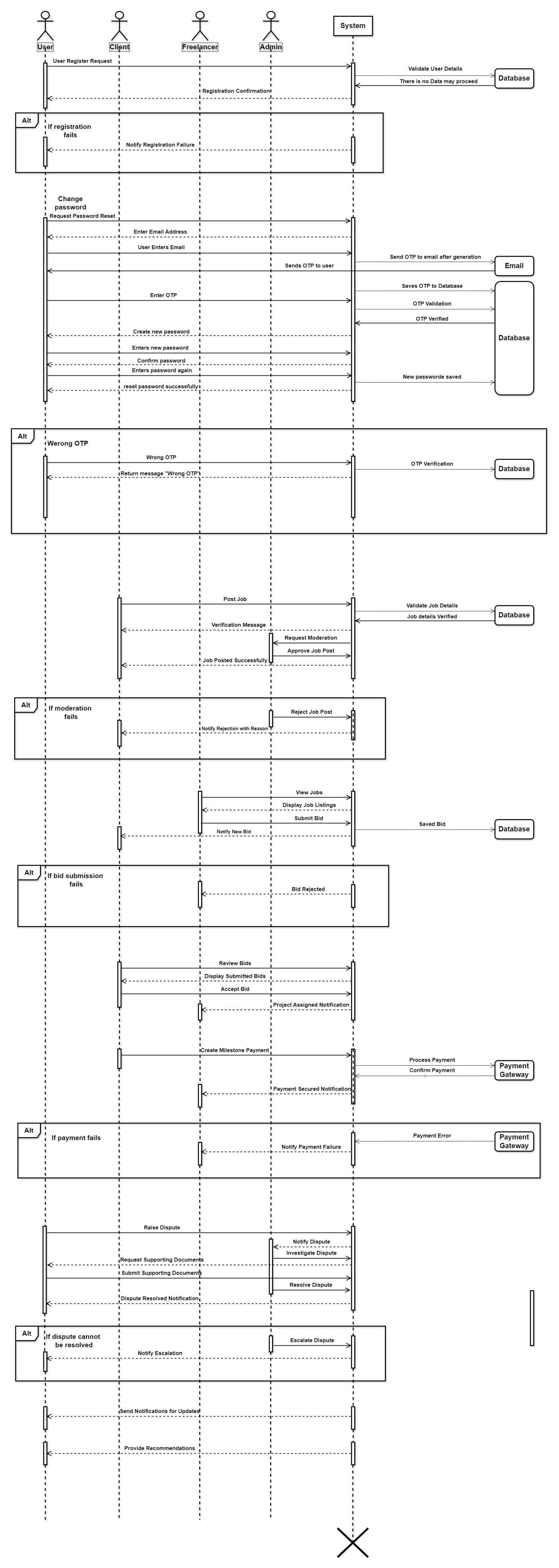
Here is the Class Diagram for this project “Freelancer” app.

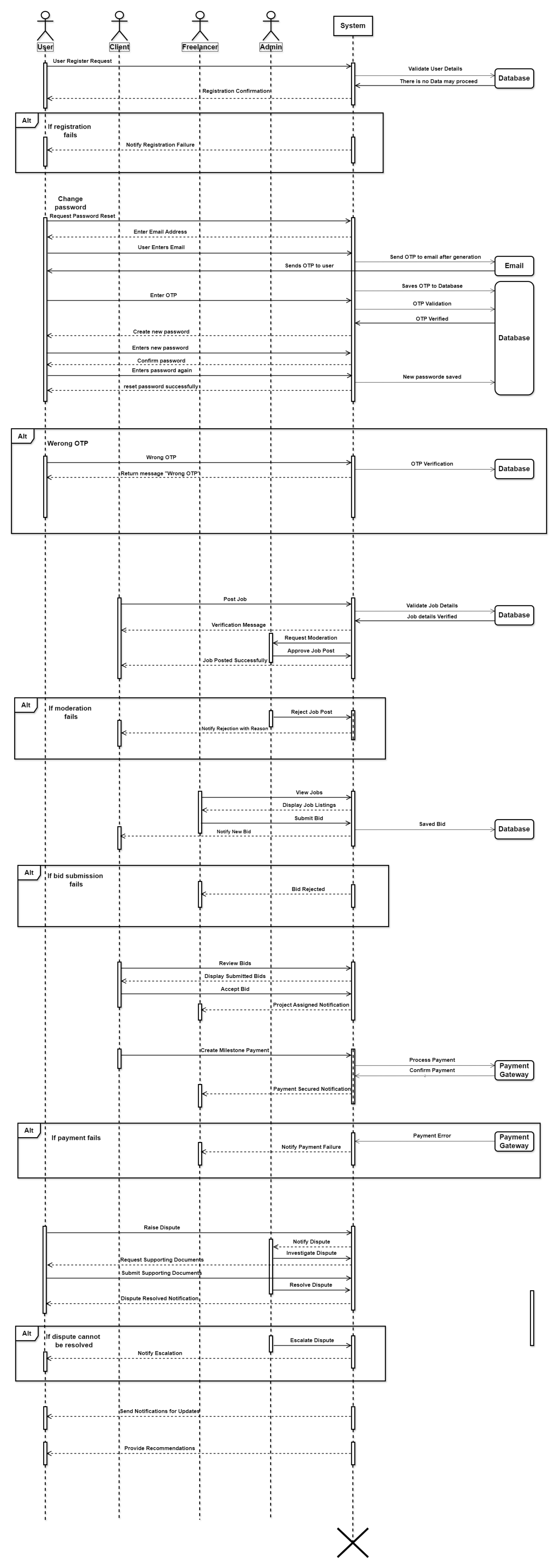


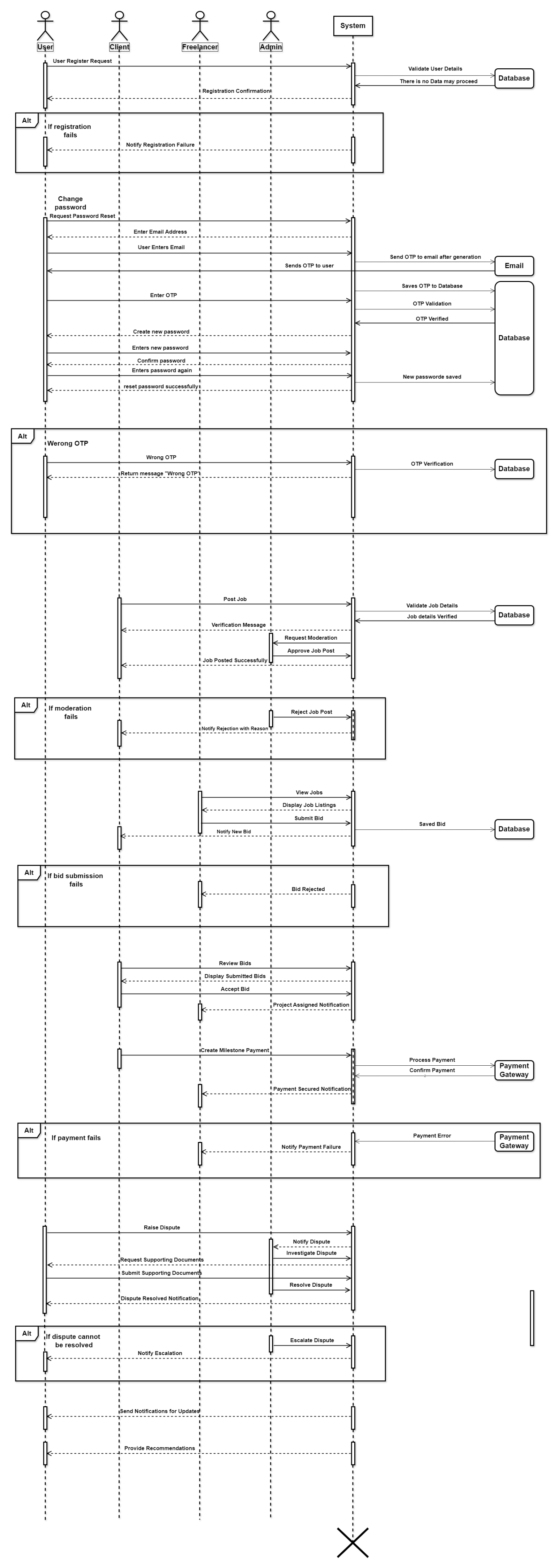
**Diagram Activity Flow Diagram:**



**Sequence Diagram:**







**Conclusion:**

The Freelancer App is a revolutionary platform that connects businesses with skilled professionals, enabling flexible and remote work opportunities. With a user-friendly interface and robust features, the app streamlines the hiring process, facilitates seamless communication, and ensures secure payments.

The architecture used in this Application is “Client Server” and “layered”.