Perfect! Now let's proceed with the **Use Case Document**, which outlines how users will interact with the system in different scenarios.

**📄 3. Use Case Document**

**Project:** *FreelancerConnect*

**📘 1. Overview**

The use case document describes the main interactions users (freelancers, clients, and admin) will have with the system. Each use case includes the goal, actors involved, flow of events, and any exceptions or alternative paths.

**🧑‍💼 Primary Actors**

* **Freelancer**
* **Client**
* **Admin**

**🧾 Use Case List**

| **ID** | **Use Case Title** | **Actor** |
| --- | --- | --- |
| UC01 | Register as Freelancer/Client | Freelancer/Client |
| UC02 | Log in to the platform | All users |
| UC03 | Edit profile | Freelancer/Client |
| UC04 | Post a project | Client |
| UC05 | Browse and filter projects | Freelancer |
| UC06 | Submit a proposal | Freelancer |
| UC07 | View proposals for a project | Client |
| UC08 | Hire a freelancer | Client |
| UC09 | Start messaging with freelancer | Client |
| UC10 | Receive project offer | Freelancer |
| UC11 | Mark project as completed | Freelancer/Client |
| UC12 | Leave rating and review | Freelancer/Client |
| UC13 | Request withdrawal of funds | Freelancer |
| UC14 | Resolve disputes | Admin |
| UC15 | Block user | Admin |
| UC16 | View dashboard | All users |
| UC17 | Log out | All users |

**📘 Sample Detailed Use Cases**

**✅ UC01 – Register as Freelancer/Client**

**Actor:** Freelancer / Client  
**Goal:** To register a new user account with role selection.  
**Precondition:** User is not logged in.  
**Postcondition:** Account is created and role is saved.

**Main Flow:**

1. User clicks “Sign Up”.
2. Selects role: Freelancer or Client.
3. Enters details (Name, Email, Password).
4. System validates and creates the account.
5. Redirects to dashboard.

**Alternate Flow:**

* If email is already used, show error.

**✅ UC04 – Post a Project**

**Actor:** Client  
**Goal:** To post a new job/project.  
**Precondition:** User must be logged in as a client.  
**Postcondition:** Project is listed on the platform.

**Main Flow:**

1. Client navigates to “Post a Project”.
2. Enters project title, description, budget, deadline, category.
3. Clicks submit.
4. System saves project and lists it for freelancers.

**Alternate Flow:**

* Budget not entered: show validation message.

**✅ UC06 – Submit a Proposal**

**Actor:** Freelancer  
**Goal:** To apply for a client’s project.  
**Precondition:** Freelancer is logged in.  
**Postcondition:** Proposal is sent to the client.

**Main Flow:**

1. Freelancer browses available projects.
2. Clicks “Submit Proposal”.
3. Enters bid amount, cover letter/message.
4. Clicks send.
5. Proposal appears under client’s project dashboard.

**✅ UC09 – Start Messaging with Freelancer**

**Actor:** Client  
**Goal:** To initiate a conversation with a freelancer.  
**Precondition:** Freelancer has submitted a proposal.  
**Postcondition:** Chat thread is created.

**Main Flow:**

1. Client opens project proposals.
2. Clicks on a freelancer’s name.
3. Sends a message.
4. Freelancer receives notification and can reply.

**✅ UC13 – Request Withdrawal of Funds**

**Actor:** Freelancer  
**Goal:** To withdraw earnings.  
**Precondition:** Freelancer has sufficient balance.  
**Postcondition:** Payment request is processed.

**Main Flow:**

1. Freelancer opens wallet/earnings section.
2. Clicks “Withdraw Funds”.
3. Enters bank or payment details.
4. Confirms.
5. Admin/payment gateway processes it.

This document ensures that all interactions are accounted for before we start designing the logic and UI.

Let me know when to move to the next one:  
➡️ **System Design Document (SDD)** (includes component design, high-level architecture, data flow, and tech stack).

Reply **continue** to proceed.

**Use Case Document for Freelancing Platform**

**1. User Registration**

* **Actor:** Visitor
* **Description:** A new user (freelancer or client) registers on the platform.
* **Preconditions:** Valid email and password.
* **Postconditions:** Account is created.
* **Flow:**
  1. User fills registration form.
  2. System validates inputs.
  3. System creates account.
  4. User receives confirmation.

**2. User Login**

* **Actor:** Registered User
* **Description:** User logs into the system.
* **Preconditions:** Registered account.
* **Postconditions:** User is authenticated.
* **Flow:**
  1. User submits credentials.
  2. System verifies.
  3. Redirects to dashboard.

**3. Create/Update Freelancer Profile**

* **Actor:** Freelancer
* **Description:** Freelancer completes profile with skills, experience, etc.
* **Preconditions:** Must be logged in as freelancer.
* **Postconditions:** Profile saved.
* **Flow:**
  1. User adds/edits profile fields.
  2. System saves updates.

**4. Create/Update Client Profile**

* **Actor:** Client
* **Description:** Client updates organization/personal info.
* **Preconditions:** Must be logged in as client.
* **Postconditions:** Profile saved.
* **Flow:**
  1. Client edits profile.
  2. System validates and saves.

**5. Post a Project**

* **Actor:** Client
* **Description:** Client posts a new project.
* **Preconditions:** Valid profile.
* **Postconditions:** Project is published.
* **Flow:**
  1. Client fills project form.
  2. System validates.
  3. Project is live.

**6. Search/Browse Projects**

* **Actor:** Freelancer
* **Description:** Freelancer views projects.
* **Preconditions:** Logged in as freelancer.
* **Postconditions:** N/A.
* **Flow:**
  1. Freelancer uses filters.
  2. Results are shown.

**7. Place a Bid on Project**

* **Actor:** Freelancer
* **Description:** Freelancer submits bid.
* **Preconditions:** Logged in + profile completed.
* **Postconditions:** Bid is submitted.
* **Flow:**
  1. Clicks on a project.
  2. Fills bid form.
  3. System validates and submits.

**8. View Freelancer Bids**

* **Actor:** Client
* **Description:** Client reviews bids for project.
* **Preconditions:** Must own the project.
* **Postconditions:** N/A.
* **Flow:**
  1. Client opens project.
  2. System displays bids.

**9. Accept Bid / Create Contract**

* **Actor:** Client
* **Description:** Client selects a freelancer.
* **Preconditions:** At least one bid exists.
* **Postconditions:** Contract is generated.
* **Flow:**
  1. Selects a bid.
  2. System creates contract.

**10. Start Work & Submit Deliverables**

* **Actor:** Freelancer
* **Description:** Freelancer submits work.
* **Preconditions:** Active contract.
* **Postconditions:** Work submitted.
* **Flow:**
  1. Freelancer uploads files.
  2. System saves submission.

**11. Approve Work & Release Payment**

* **Actor:** Client
* **Description:** Client reviews work and releases funds.
* **Preconditions:** Work submitted.
* **Postconditions:** Funds sent to wallet.
* **Flow:**
  1. Client approves.
  2. System releases payment.

**12. Wallet & Transaction History**

* **Actor:** All users
* **Description:** View wallet balance and transactions.
* **Preconditions:** Must be logged in.
* **Postconditions:** N/A.
* **Flow:**
  1. User navigates to wallet.
  2. Transactions are displayed.

**13. Leave a Review**

* **Actor:** Client or Freelancer
* **Description:** Submit a review after contract.
* **Preconditions:** Contract completed.
* **Postconditions:** Review saved.
* **Flow:**
  1. User fills rating/comment.
  2. System saves.

**14. Report/Dispute a Contract**

* **Actor:** Client or Freelancer
* **Description:** Raise an issue during/after work.
* **Preconditions:** Active or completed contract.
* **Postconditions:** Dispute raised.
* **Flow:**
  1. Selects contract.
  2. Fills dispute form.

**15. KYC Verification**

* **Actor:** Freelancer or Client
* **Description:** Upload identity documents.
* **Preconditions:** Logged in.
* **Postconditions:** Pending/approved.
* **Flow:**
  1. Upload ID.
  2. Admin reviews and verifies.

**16. Contact Support**

* **Actor:** All Users
* **Description:** Raise a ticket for help.
* **Preconditions:** Logged in.
* **Postconditions:** Ticket created.
* **Flow:**
  1. Fill form.
  2. System sends to support team.

**17. Change Password / Manage Account Settings**

* **Actor:** All Users
* **Description:** Update password, email, or delete account.
* **Preconditions:** Logged in.
* **Postconditions:** Changes applied.
* **Flow:**
  1. User updates settings.
  2. System saves changes.