**Phase III: Software Design and Modeling**

**Deadline: April 1st, 2024, 23:59**

Dear Students,

In Phase III of our project, we will focus on software design and modeling. Below are the requirements for this phase, explained in an easy-to-understand format:

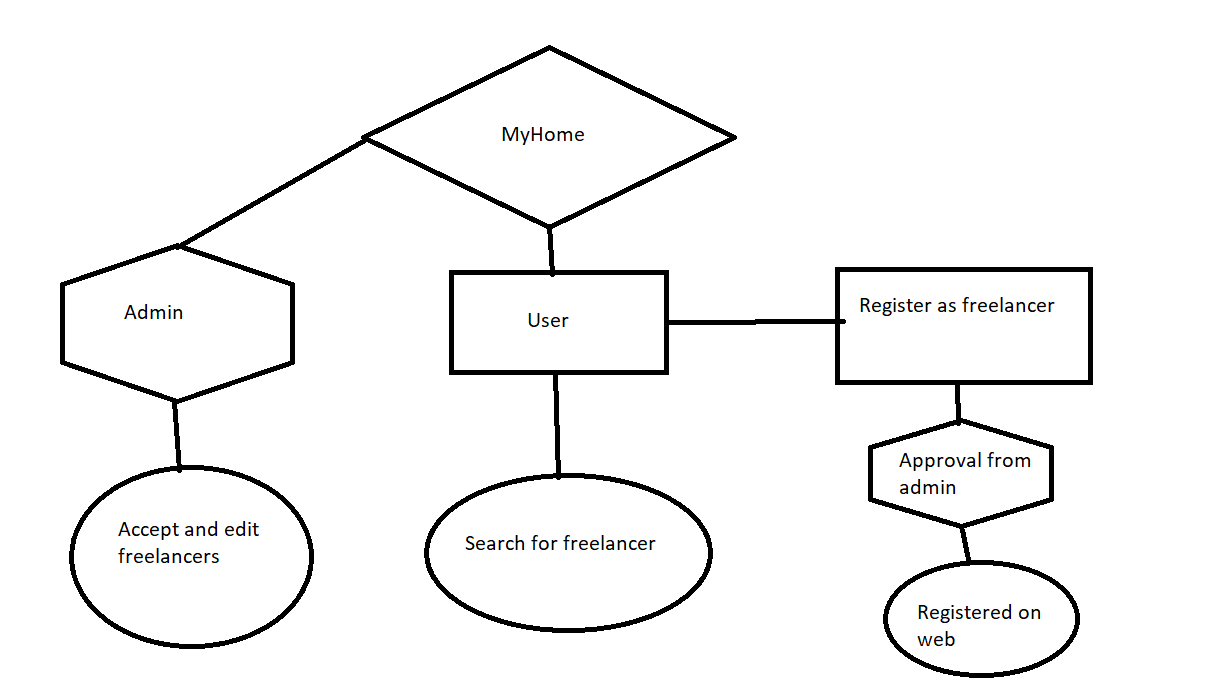
**Software Design and Modeling**

**Group Name: My home**

**Software Architecture**

System Architecture:

Explain how different parts of the system work together. Think of it as describing the big picture of your application - what it does and how it does it.

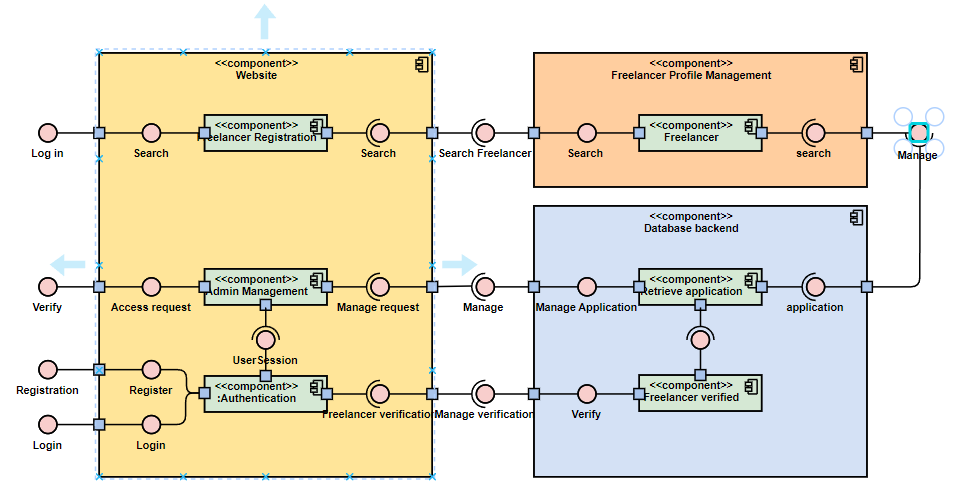


* The website acts as a bridge between clients seeking home maintenance services and freelancers offering such services.
* Components include user registration, job posting, freelancer profiles, messaging, and admin management.
* These components interact seamlessly to facilitate communication and task management between clients and freelancers.
* Freelancer Profile Management Module:
  + Freelancers can create and manage their profiles, including personal information, skills, experience, portfolio, and payment preferences.
  + This module enables freelancers to showcase their expertise and attract potential clients.
* Admin Management Module:
  + Admins oversee the overall operation of the platform, managing user accounts, handling disputes, and ensuring compliance with platform policies.
  + They have access to verification systems to review and verify freelancer profiles for authenticity, accuracy, and compliance.
* Database Backend:
  + The database stores user information, job postings, messages, and other relevant data.
  + It facilitates data retrieval and management, ensuring the integrity and reliability of information across the platform.

Component Diagram:

Draw a picture showing the different parts (components) of your application and how they interact with each other. For example, if your application has a login feature, a component diagram would show how the login component talks to other parts of the system.

This diagram will illustrate how the different components of the website interact with each other. It will depict the flow of information and actions between users, and administrative controls.



User Interface (UI):

* + This component represents the user interface through which users interact with the application.
  + It includes web pages, forms, and graphical elements displayed to users.
* User Registration and Authentication Module:
  + Handles user registration and authentication processes.
  + Allows users to register, log in, and manage their accounts.
* Freelancer Profile Management Module:
  + Manages freelancer profiles, including personal information, skills, and portfolio.
* Admin Management Module:
  + Administers the overall operation of the platform.
  + Manages user accounts, verifies freelancer profiles, and resolves disputes.
* Database Backend:
  + Stores and manages data related to users, job postings, messages, and other application entities.
  + Facilitates data retrieval and persistence for the application.

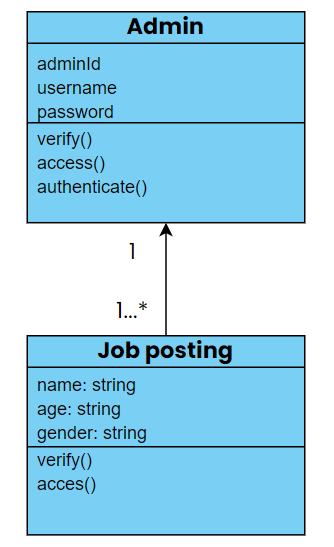
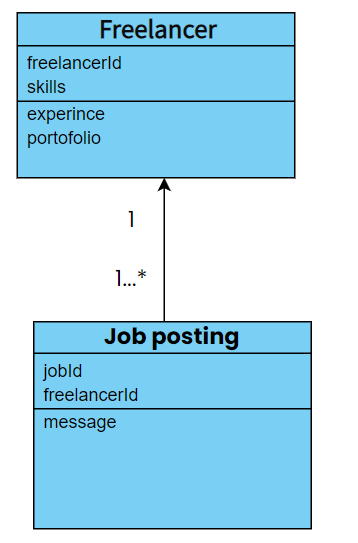
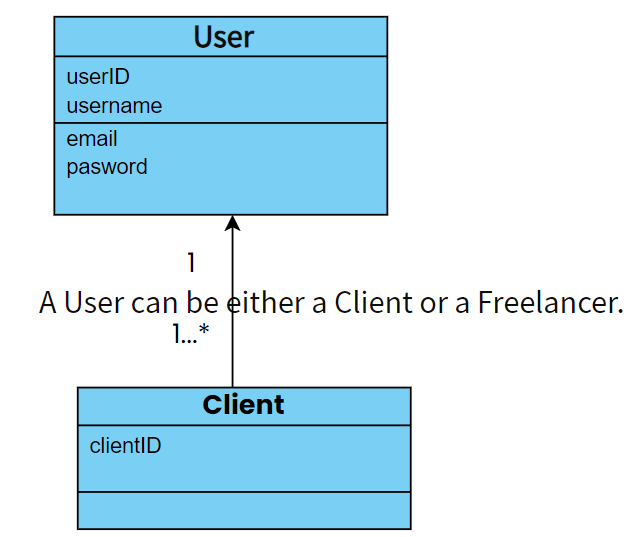
The diagram illustrates how these components interact with each other to provide the functionality required by the application. The admin module oversees the entire operation, while external services enhance the functionality of the application.

**Detailed Design**

Class Diagram:

Think of a class diagram as a family tree for your application. It shows the different types of "things" in your application (called classes) and how they relate to each other. For example, if your application deals with cars, a class diagram would show that a Car class might have attributes like color and model, and methods like drive() and park().

This diagram will showcase the structure of the system's classes and their relationships. It will include classes for users, job postings, messages, and administrative functions, illustrating how they are interconnected.

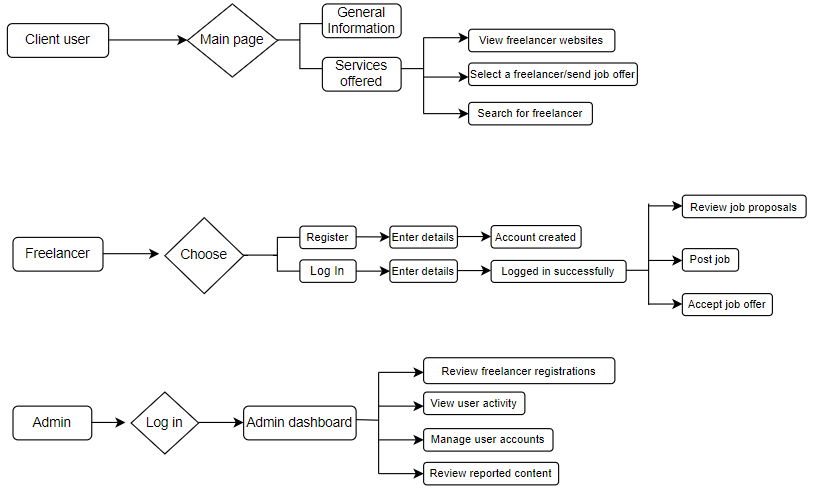


Sequence Diagrams:

Sequence diagrams show the order in which things happen in your application. They're like step-by-step instructions for how different parts of your application interact with each other to accomplish a task. For example, a sequence diagram for ordering food

online would show the steps involved, like selecting items, adding them to the cart, and checking out.

These diagrams will outline the step-by-step processes involved in actions such as user registration, job posting, proposal submission, and hiring. They will illustrate the interactions between various components of the system during these processes.



Database Design:

Explain how you've organized the data in your application. This includes things like what tables you have in your database, how they're related to each other, and how you've made sure your data is organized efficiently.

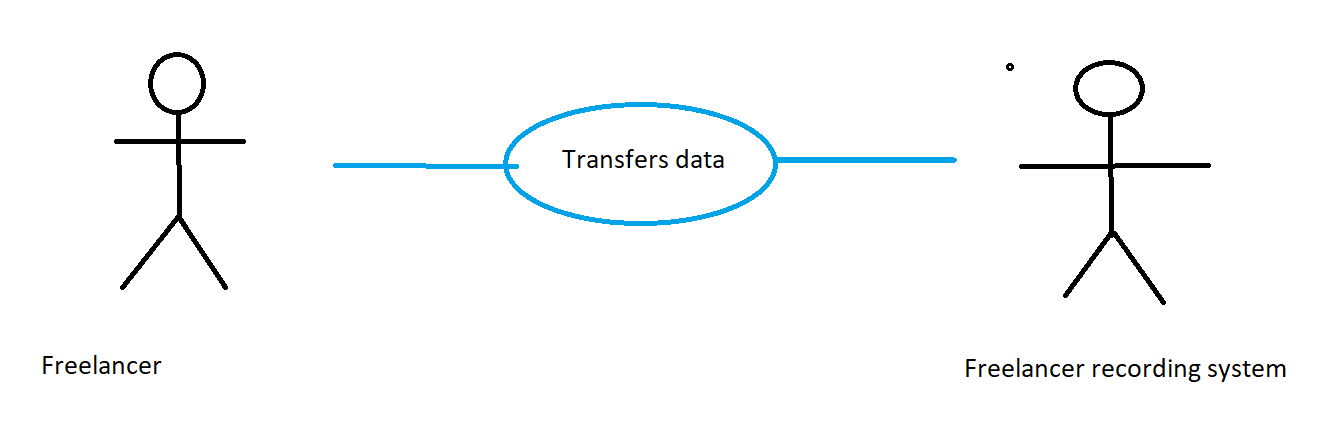
Explain the structure of the database, including the table for storing user information, job details, messages, and administrative data.

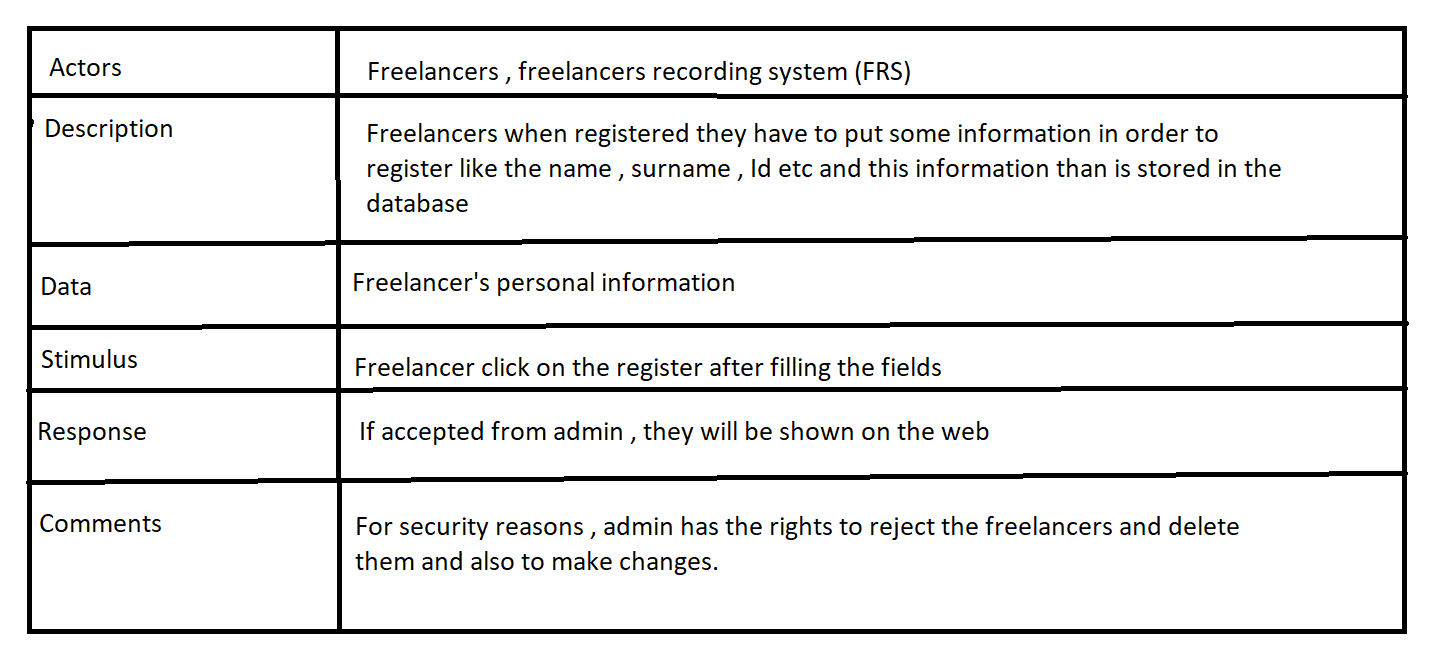


**Modeling**

Use Case Diagram:

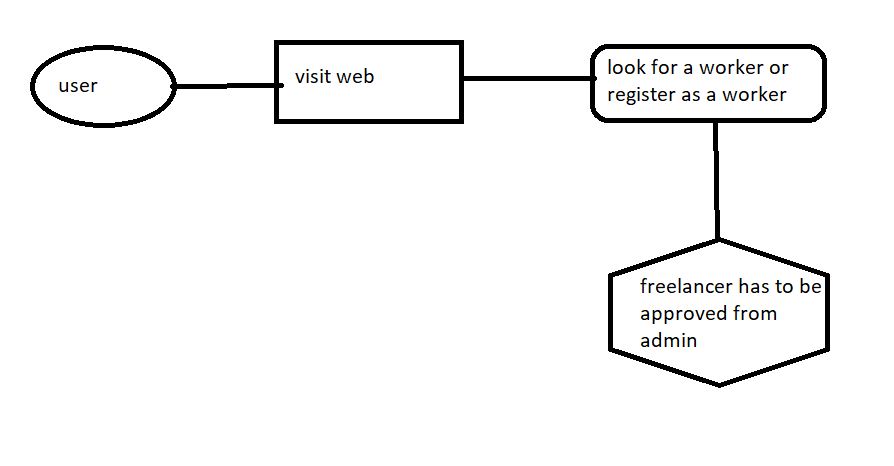
A use case diagram shows the different ways people (or other systems) can use your application. It's like a map of all the different things your application can do. For example, a use case diagram for a music streaming app might show that users can search for songs, create playlists, and listen to music.

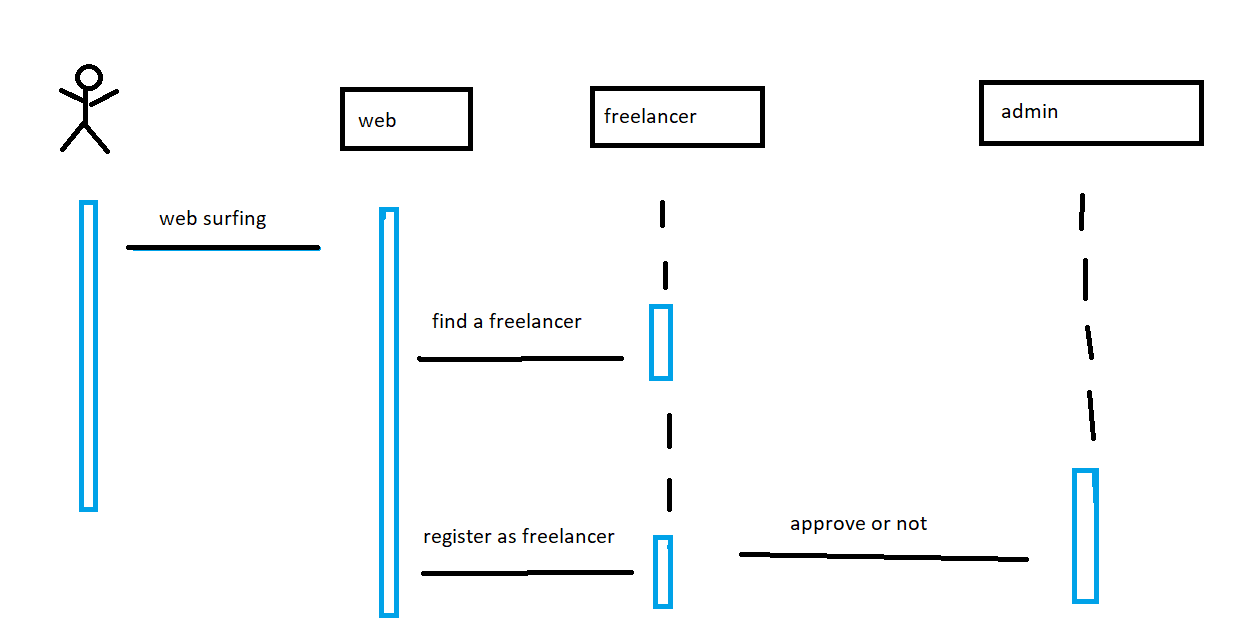




Activity Diagrams:

Activity diagrams show the flow of activities in your application. They're like flowcharts that show the steps involved in completing a task. For example, an activity diagram for booking a flight might show the steps involved, like searching for flights, selecting one, and entering passenger information.





State Diagrams:

State diagrams show the different states that an object in your application can be in, and how it transitions between those states. They're like maps of all the possible "statuses" your application can be in. For example, a state diagram for a light switch might show that the switch can be in the "on" or "off" state, and how it transitions between them.

We do not have this.

**Additional Notes**

Feel free to include detailed explanations and annotations alongside your diagrams to ensure clarity.

Make sure all your diagrams are properly labeled and formatted for easy understanding. Include any additional notes or instructions for the submission.

If you have any questions or need clarification, don't hesitate to ask.