

Your final report must include these things **(total 70 points)**:

1. Your database tables and attributes (2 tables minimum, 5 tables maximum, 3 attributes minimum per table, 15 attributes maximum per table). You need to explain each table, attribute, and PK-FK relationships. (0 pts for no submission, 3 pts for no explanation, 6 pts for missing explanation, 10 pts for full explanation) (10 points)

Freelancer table:

```
CREATE TABLE Freelancer (
    FreelancerID int Not Null PRIMARY KEY,
    Freelancer_First_Name VARCHAR(30) NOT NULL,
    Freelancer_Last_Name VARCHAR(30) NOT NULL,
    Skillset VARCHAR(20) NOT NULL,
    HourlyRate DECIMAL(10,2) NOT NULL
);
```

Output of freelancer table:

```
Freelance Job Management System

All Freelancers:
ID: 1, Name: Aryan Desai, Skillset: Coding, Hourly Rate: $25.0
ID: 2, Name: k h, Skillset: h, Hourly Rate: $8.0
ID: 3, Name: Eric Jose, Skillset: Java, Hourly Rate: $35.0
ID: 4, Name: Jon Jones, Skillset: python, Hourly Rate: $25.0
ID: 5, Name: Jon Burrows, Skillset: python, Hourly Rate: $25.0
ID: 6, Name: Joe Burrows, Skillset: python, Hourly Rate: $25.0
ID: 121, Name: hi t, Skillset: N/A, Hourly Rate: $8.0
```

Client:

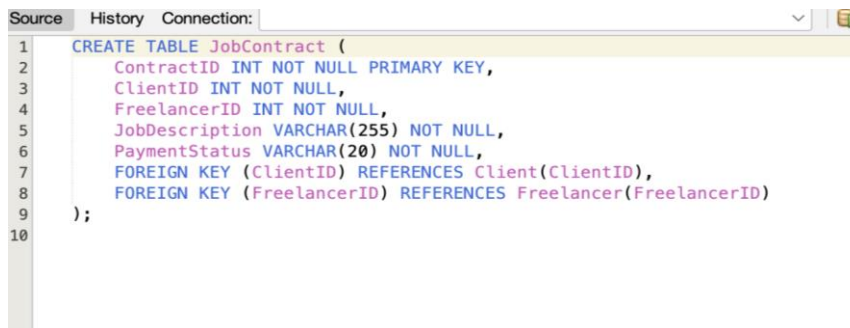
```
CREATE TABLE Client (
    ClientID int Not Null PRIMARY KEY,
    Client_First_Name VARCHAR(30) NOT NULL,
    Client_Last_Name VARCHAR(30) NOT NULL,
    CompanyName VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL
);
```

Output of Client:

```
MainGUI
Freelance Job Management System

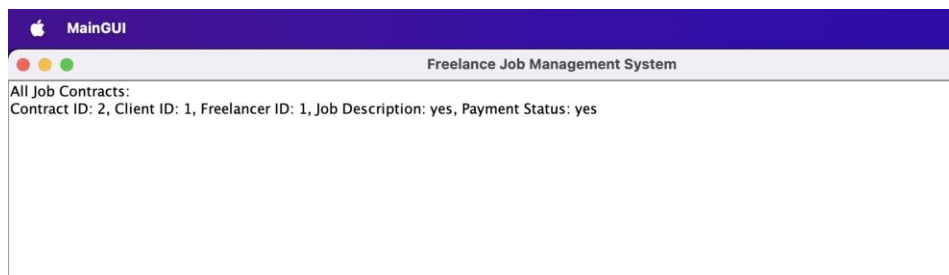
All Clients:
ID: 1, Name: h k, Company: pokoiru, Email: email@p.com
```

Job Contract table:



```
Source History Connection:
1 CREATE TABLE JobContract (
2     ContractID INT NOT NULL PRIMARY KEY,
3     ClientID INT NOT NULL,
4     FreelancerID INT NOT NULL,
5     JobDescription VARCHAR(255) NOT NULL,
6     PaymentStatus VARCHAR(20) NOT NULL,
7     FOREIGN KEY (ClientID) REFERENCES Client(ClientID),
8     FOREIGN KEY (FreelancerID) REFERENCES Freelancer(FreelancerID)
9 );
10
```

Output of job contract:

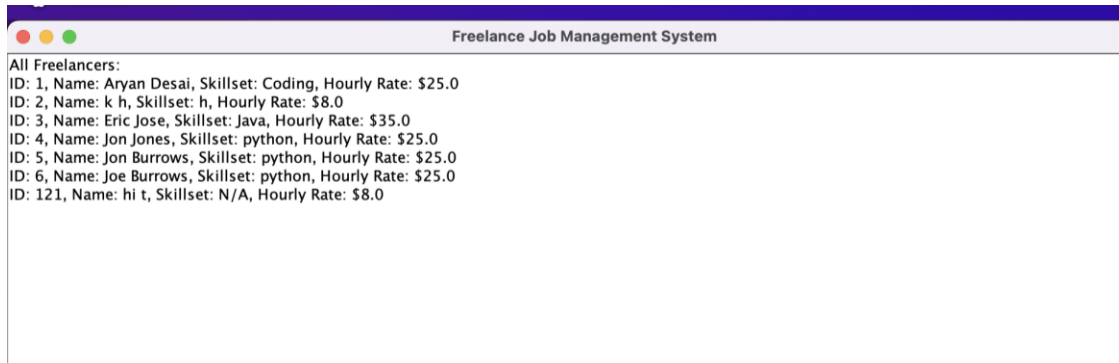


Explanation: This database is part of a freelance management system designed to facilitate the relationship between **clients**, **freelancers**, and **job contracts**.

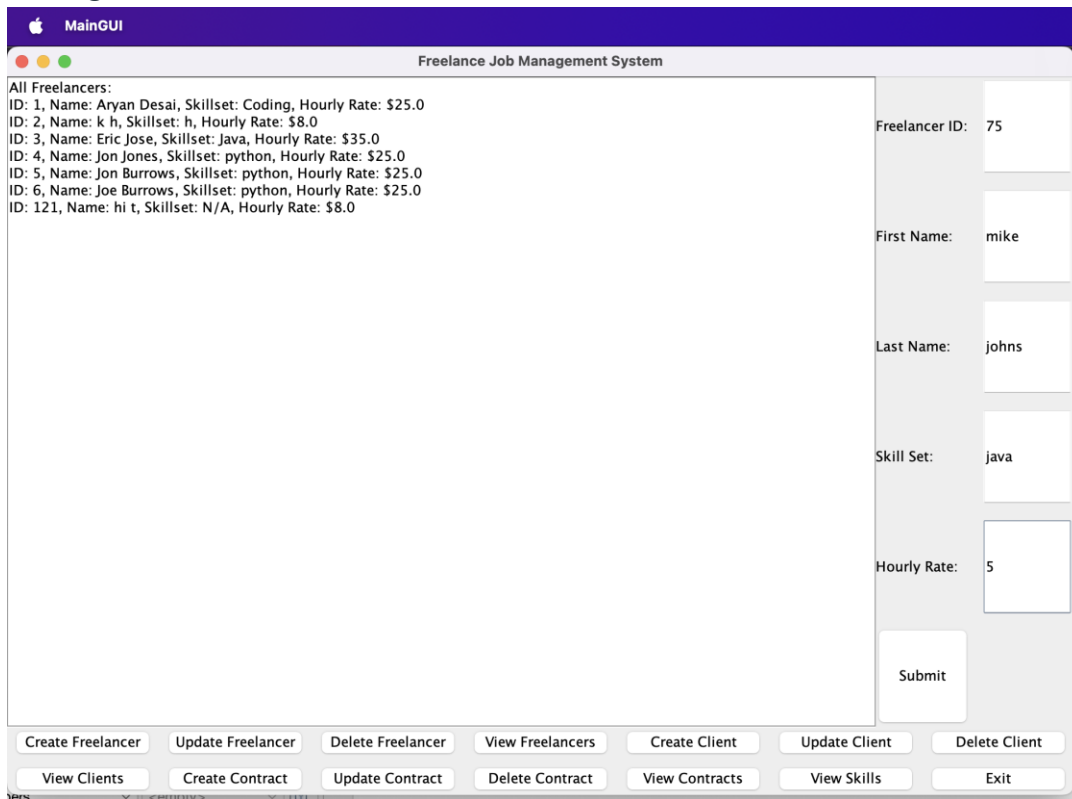
- **Clients**: Store information about clients, such as names, company names, and emails. This is linked to job contracts through a foreign key.
- **Freelancers**: Store information about freelancers, including their skills and hourly rate. They are also associated with job contracts through a foreign key.
- **Job Contracts**: Records of contracts between clients and freelancers, tracking job descriptions, payment statuses, and assigned freelancers.

This system helps in organizing and managing the relationships and transactions between clients and freelancers, providing functionalities like creating, updating, and deleting clients, freelancers and contract information.

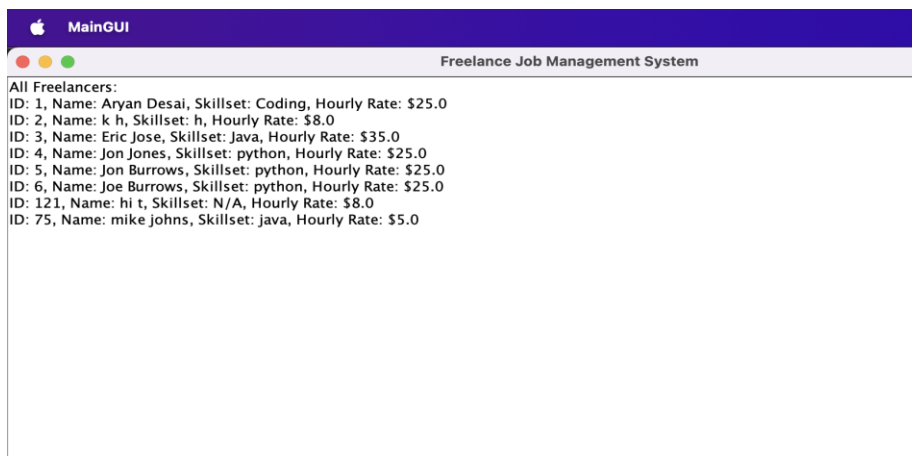
2. How to create a record for every table? Show your table before creating a record. Show your interface for creating a record (GUI, Command Line, etc.). Show your table after creating a record. (0 pts for no submission, 5 pts for showing the create interface for every table, 10 pts for showing the create interface for every table and showing your table before and after creating the record with screenshots) (10 points)  
Freelancer current:



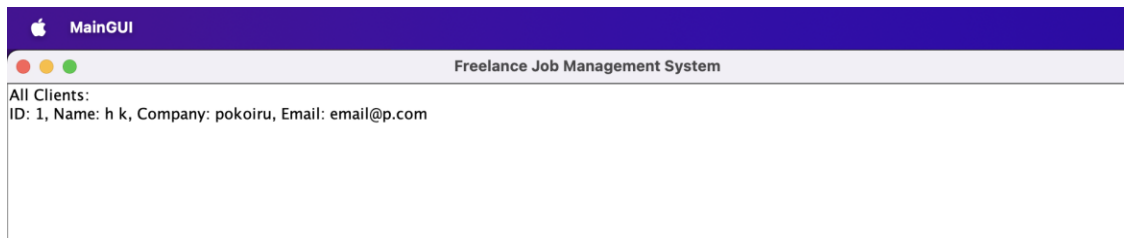
During freelancers add:



After:



Client before:

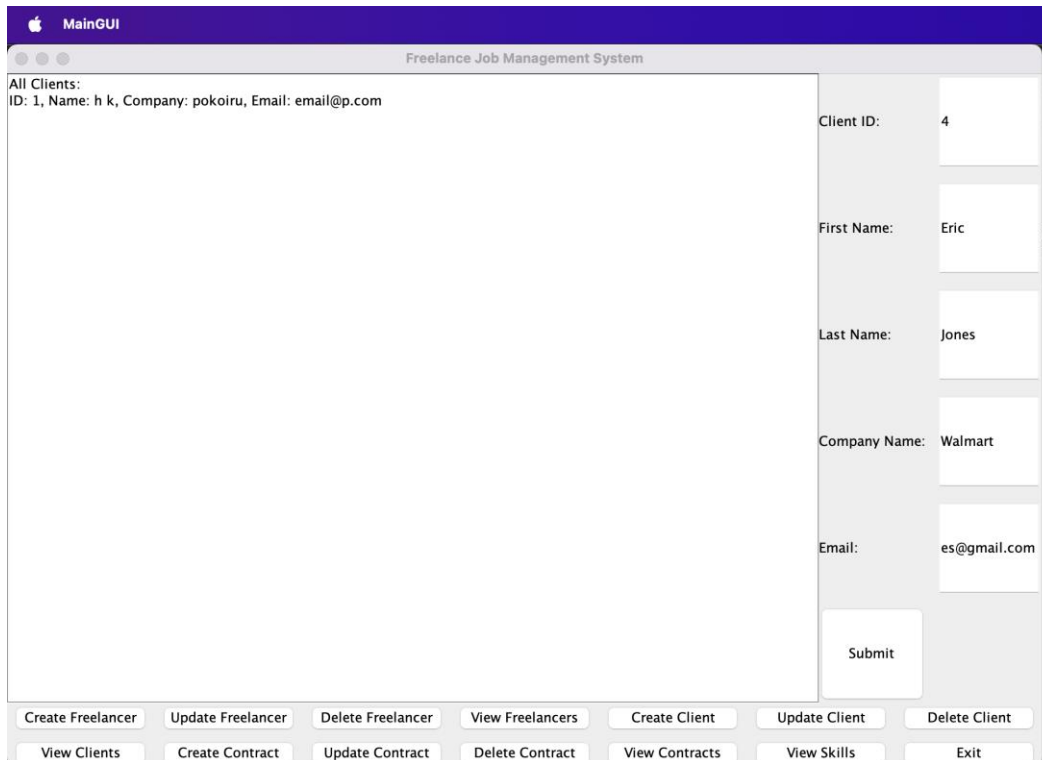


MainGUI

Freelance Job Management System

All Clients:  
ID: 1, Name: h k, Company: pokoiru, Email: email@p.com

Client during:



MainGUI

Freelance Job Management System

All Clients:  
ID: 1, Name: h k, Company: pokoiru, Email: email@p.com

Client ID: 4

First Name: Eric

Last Name: Jones

Company Name: Walmart

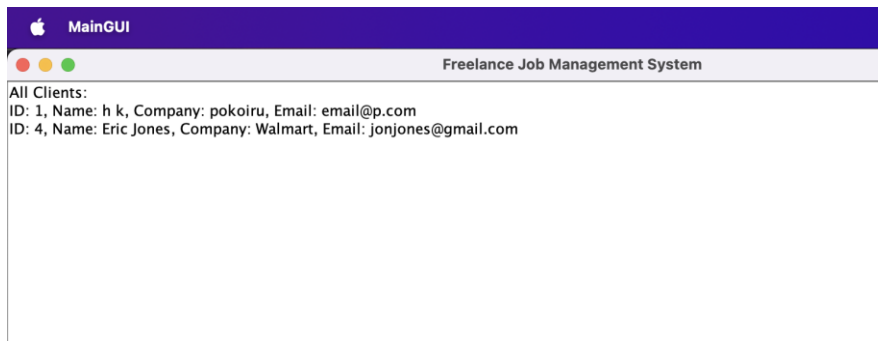
Email: es@gmail.com

Submit

Create Freelancer Update Freelancer Delete Freelancer View Freelancers Create Client Update Client Delete Client

View Clients Create Contract Update Contract Delete Contract View Contracts View Skills Exit

After:

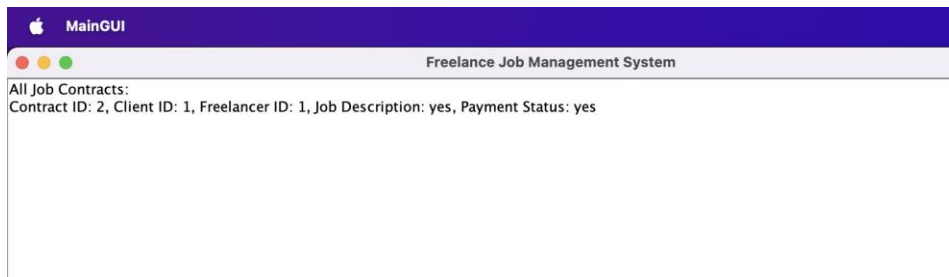


MainGUI

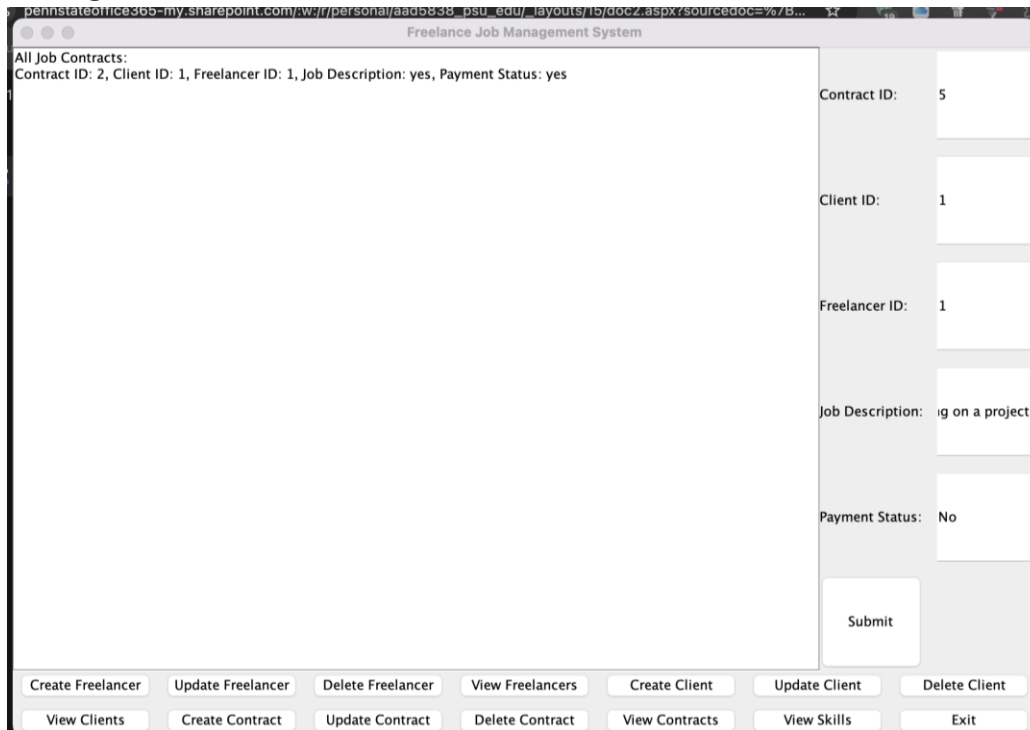
Freelance Job Management System

All Clients:  
ID: 1, Name: h k, Company: pokoiru, Email: email@p.com  
ID: 4, Name: Eric Jones, Company: Walmart, Email: jonjones@gmail.com

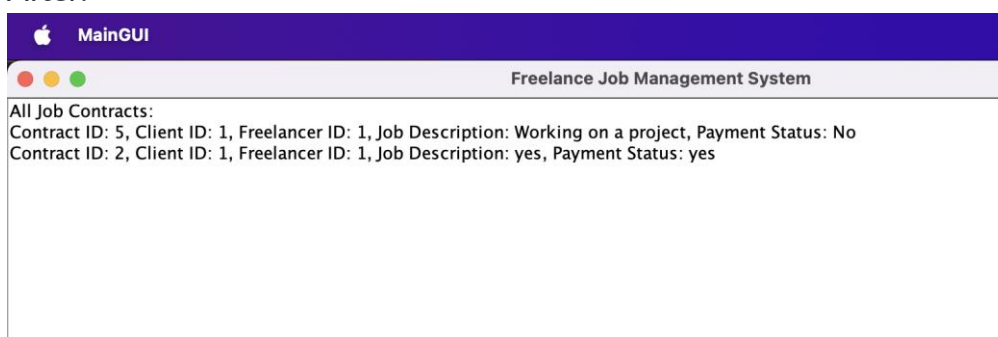
Job contract before:



During:

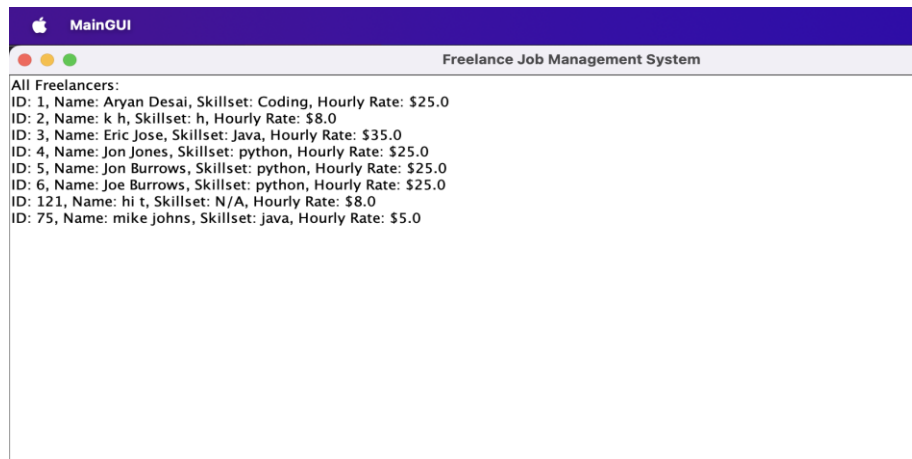


After:

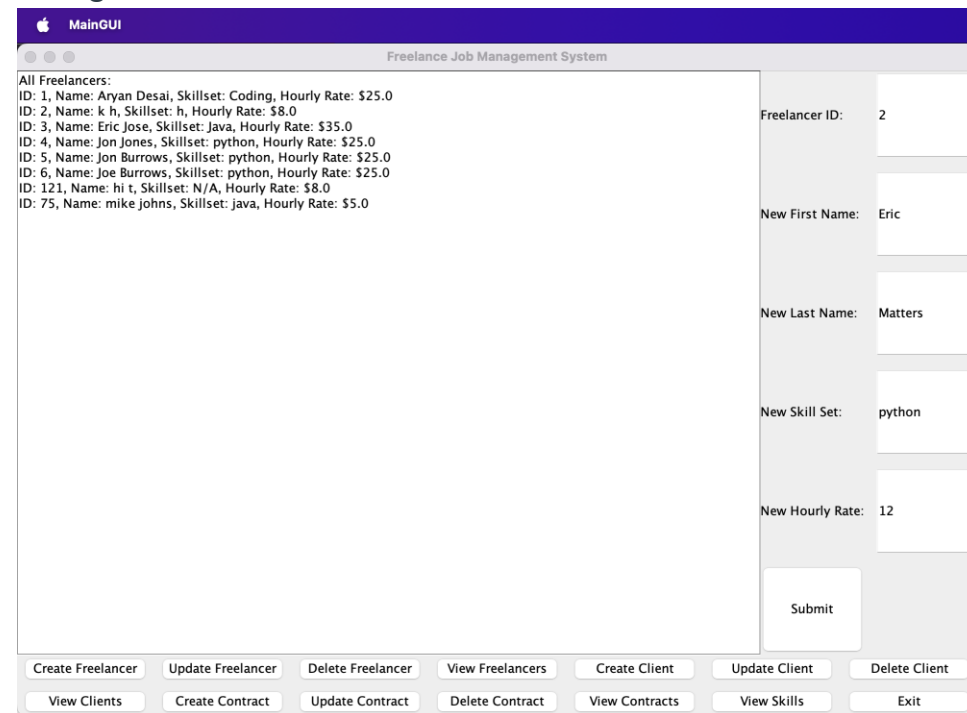


- How to update a record for every table? Show your table before updating a record. Show your interface for updating a record (GUI, Command Line, etc.). Show your table after updating a record. (0 pts for no submission, 5 pts for showing the update interface for every table, 10 pts for showing the update interface for every table and showing your table before and after updating the record with screenshots) (10 points)

## Freelancers before:



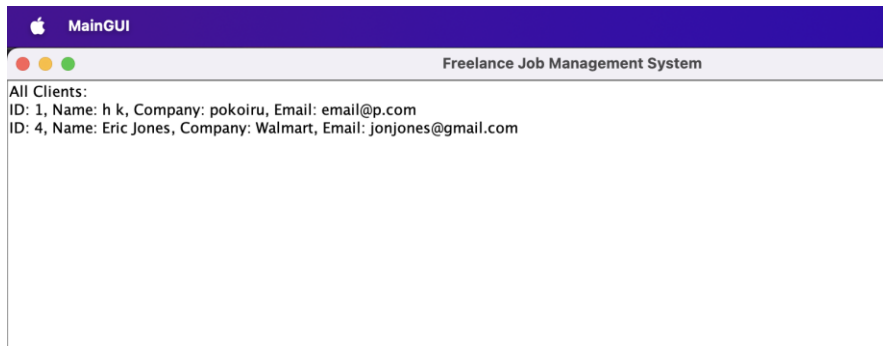
## During:



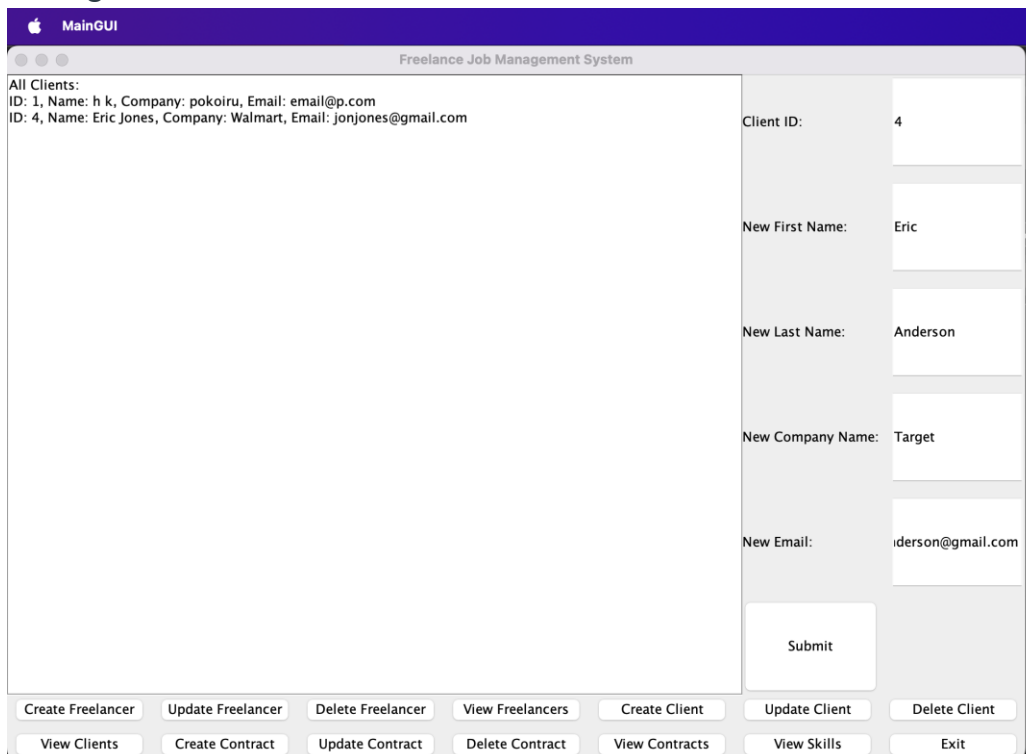
## After:



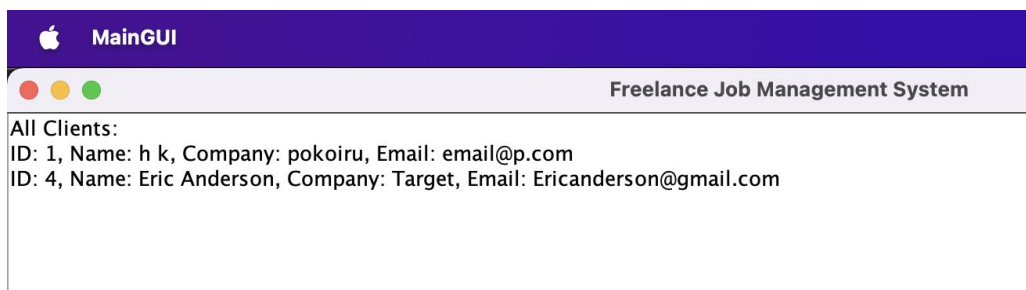
## Clients before:



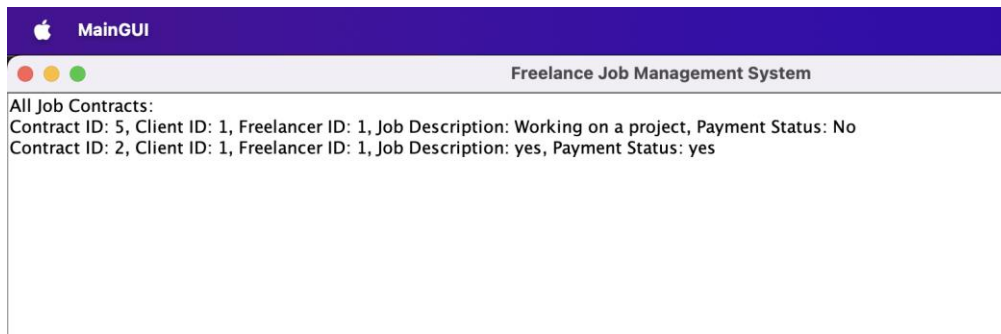
During:



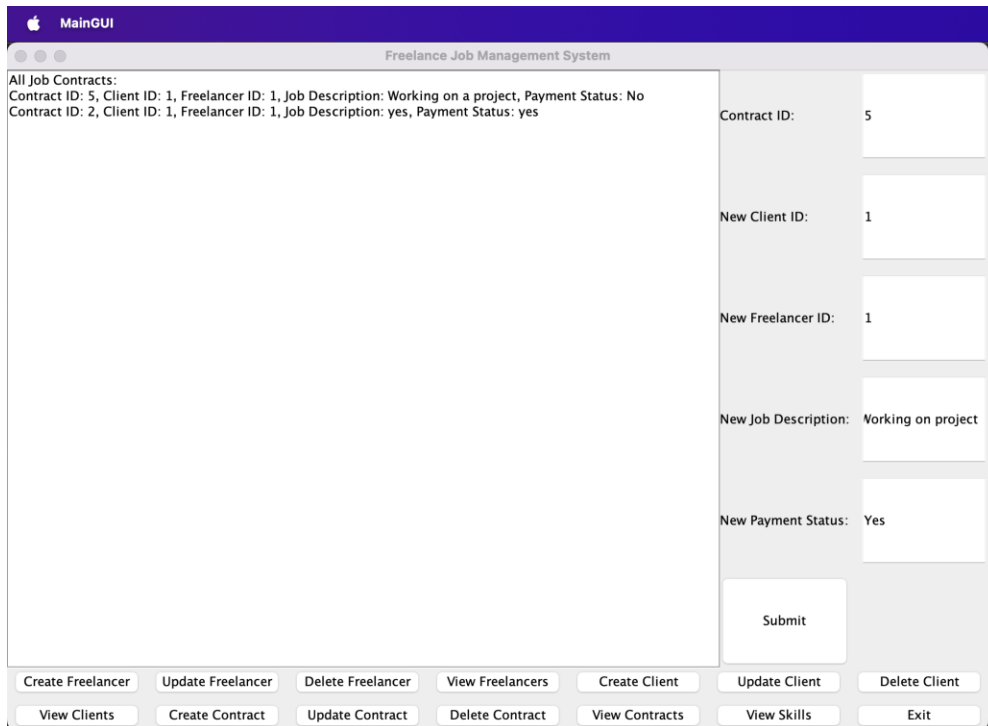
After:



Job contracts before:



During:



After:

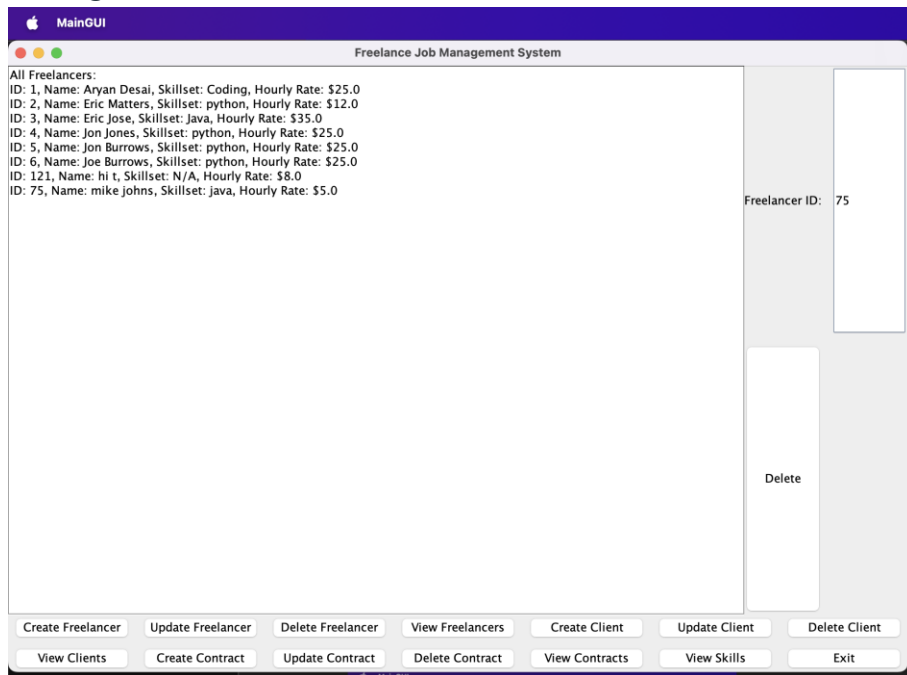


- How to delete a record for every table? Show your table before deleting a record. Show your interface for deleting a record (GUI, Command Line, etc.). Show your table after deleting a record. (0 pts for no submission, 5 pts for showing the delete interface for every table, 10 pts for showing the delete interface for every table and showing your table before and after deleting the record with screenshots) (10 points)
- Freelancer before:





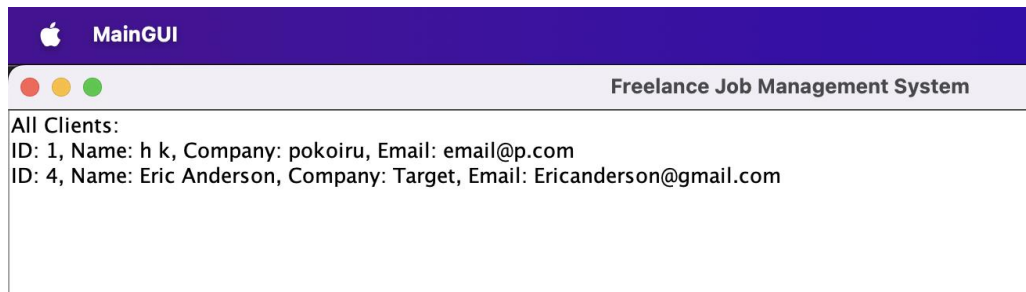
During:



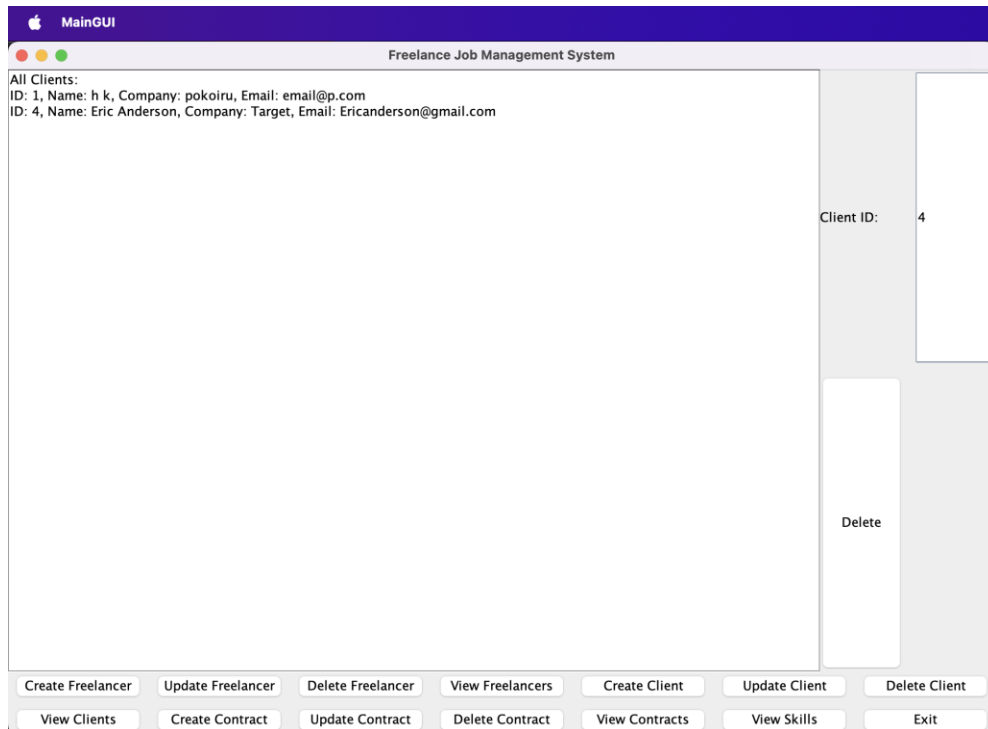
After:



Clients before:



During:



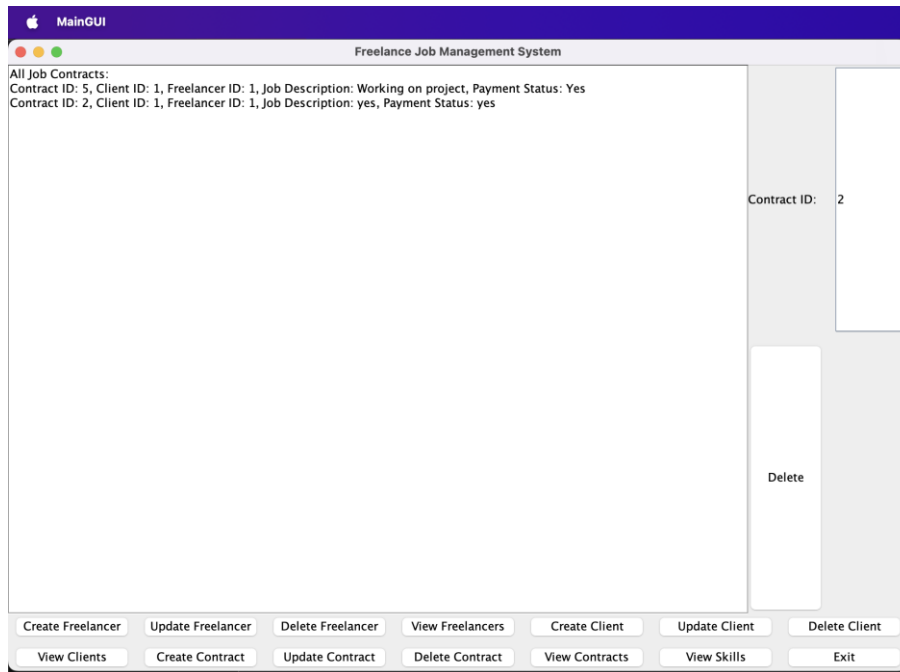
After:



Job Contracts before:



During:

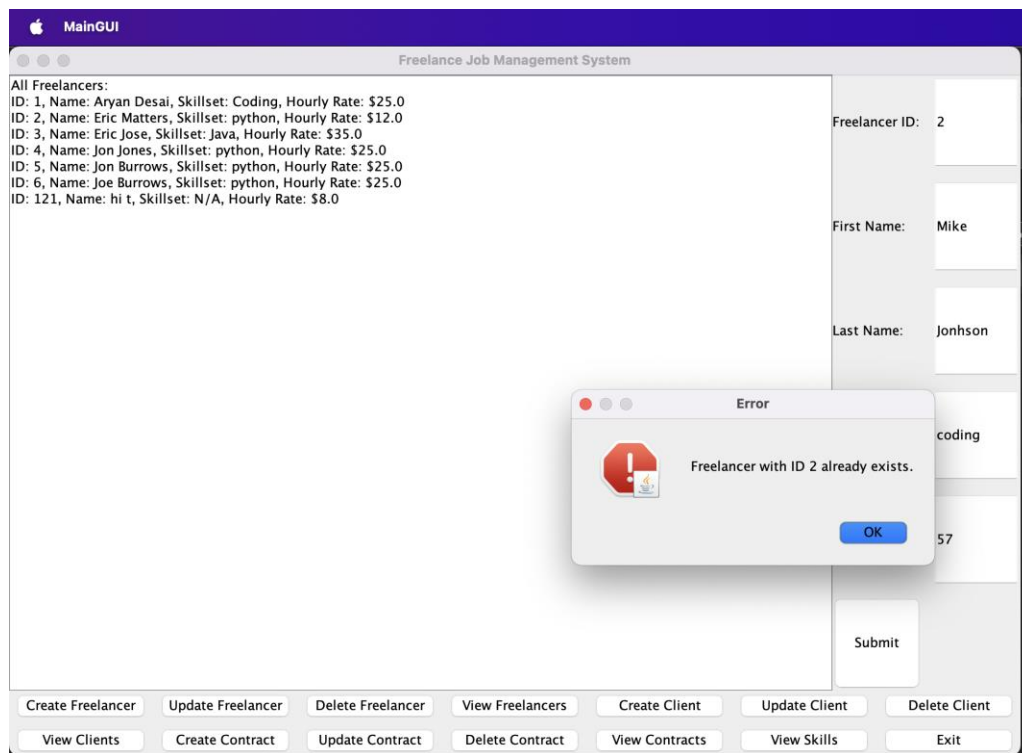


After:

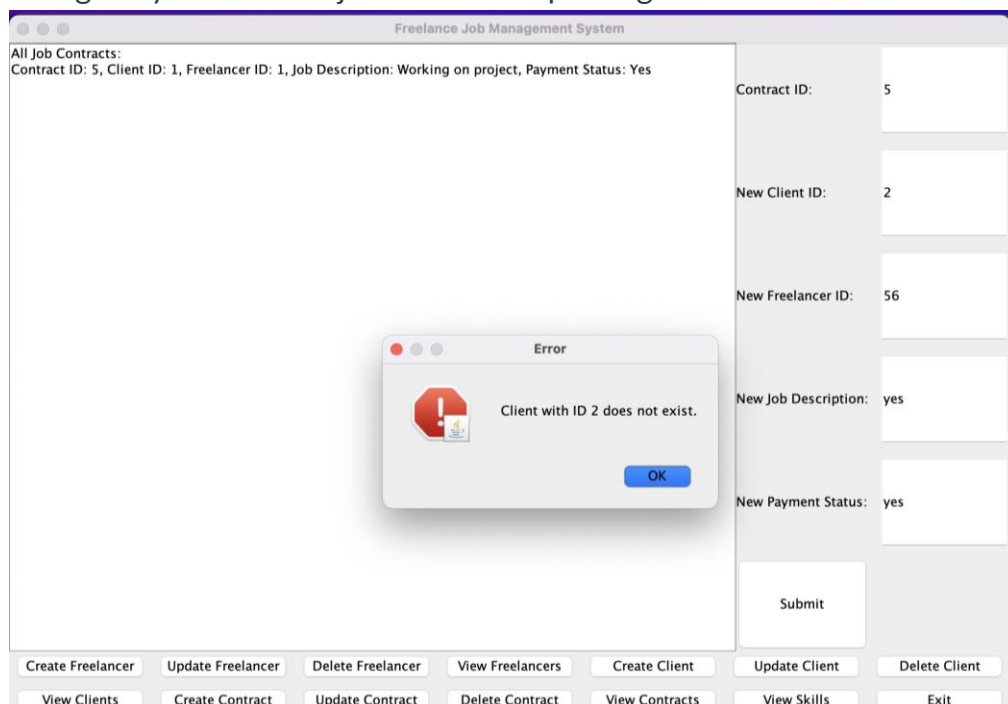


5. Show 5 different examples of error handling. Example errors: primary key violation, foreign key violation, empty field, data type error (writing a string text inside a field that only accepts integer), email validation (check if the data is actually in e-mail format), phone validation (check if the phone entered is in a correct format). (2 pts for each example, each example must be a different type of error handling, in other words if you show primary key violation error handling twice for two tables, it is still counted as only 2 pts, you must perform another error handling such as foreign key violation in order to get another 2 pts). (10 points)

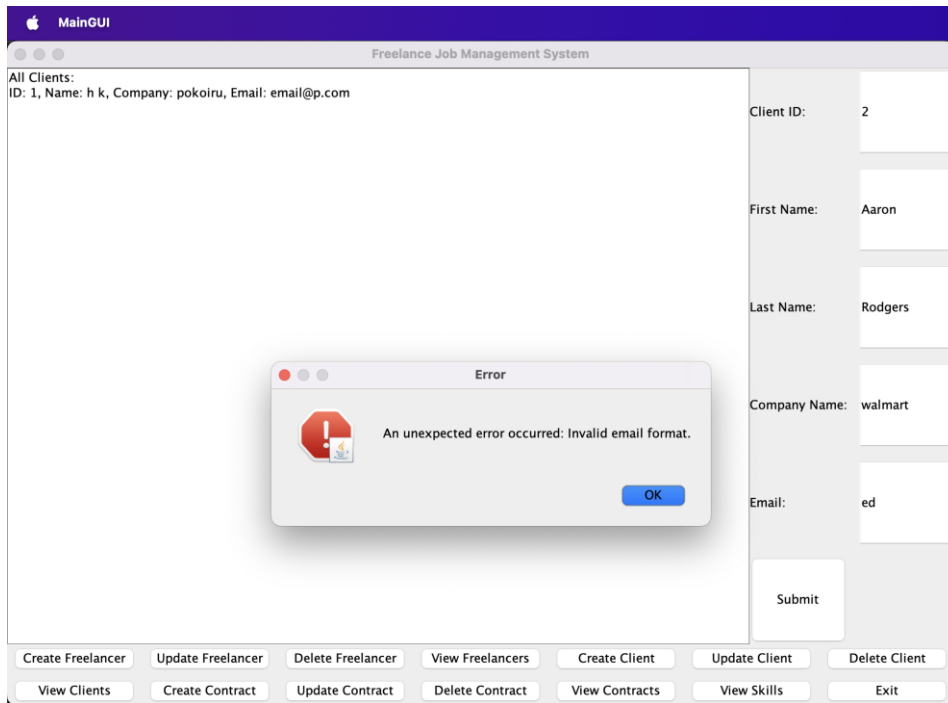
1. Pk violation in freelancers id exists but tried to add another:



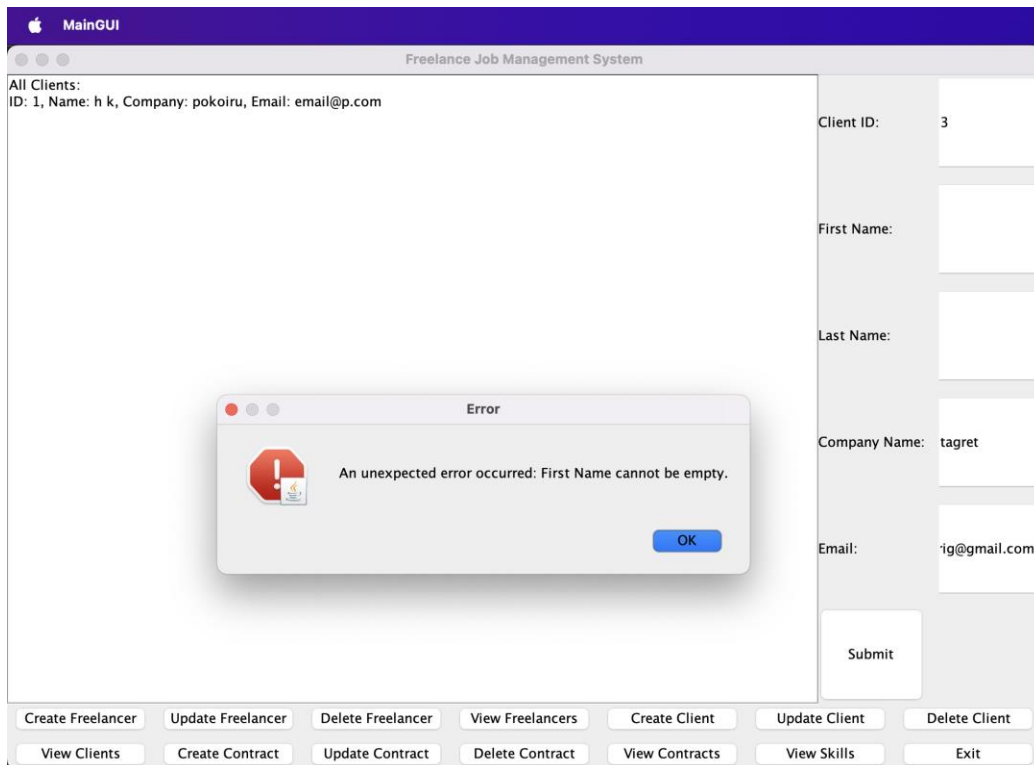
2. foreign key violation in job contracts updating an id that doesn't exist for client id:



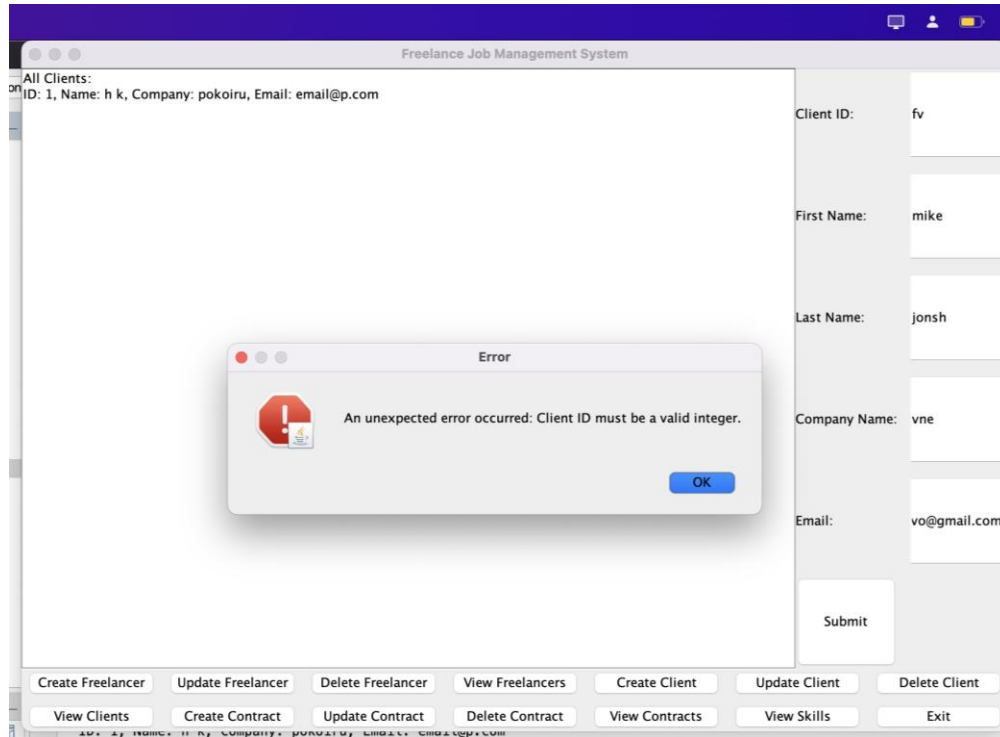
3. Email format:



#### 4. Blank entries error:

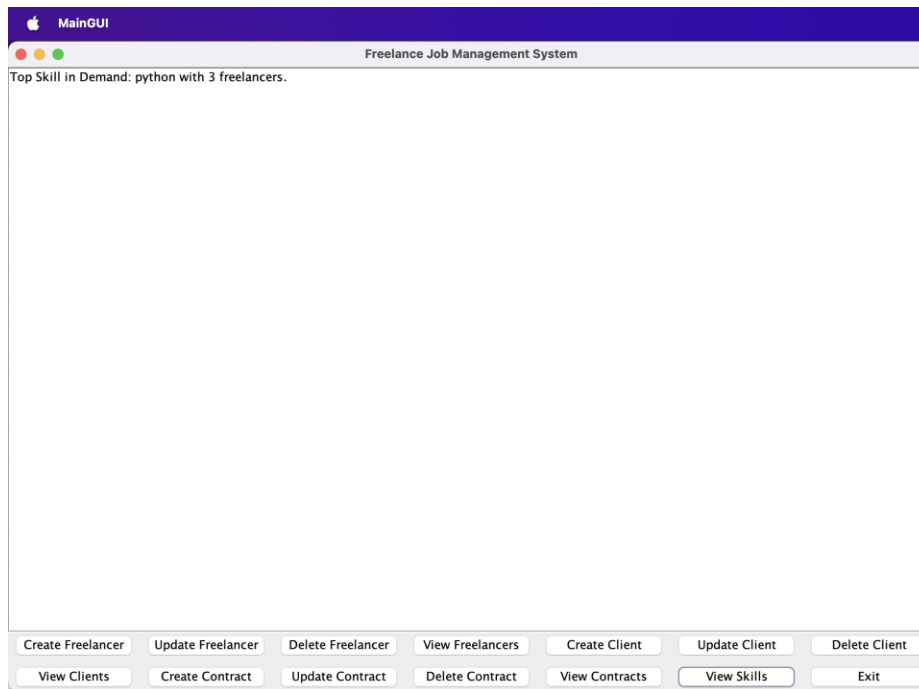


5. Ids must be positive integers error:



6. Show something unique about your project that does not exist in other student projects. This must be a unique thing for your application. You have to do a research and find an interesting thing to show in your application that makes your application unique. Some examples are using other JComponents (the ones we haven't discussed in class) in your application, using some charts or graphs in your application (showing demographics in pie chart, pie chart might be harder to do in a command line application), playing a video tutorial inside your application). You must ask me first before doing this unique aspect of your project so I can approve that if it is something unique. (10 points)

The unique feature in my project is the SkillMap functionality. It analyzes the skill sets of freelancers and identifies the most in-demand skills based on the frequency of their appearance in the freelancer profiles. This helps employers or project managers quickly spot key skills and trends in the freelancer pool.



7. Find a user to test your application. Ask her/him to try to break your program (offer him/her a coffee if s/he breaks it). Write a paragraph about his/her user experience. What were the problems? What kind of suggestions did s/he give? Which updates did you do after the test? (0 pts for no submission, 5 pts for mediocre writing, 10 pts for very well written detailed writing)

I had my grandmother test my code, and during her testing, she encountered several issues. She found errors when leaving fields blank, when entering non-existent IDs, and when attempting to delete IDs that didn't exist. I went back and fixed those problems, and after testing it again, she couldn't identify any other errors. Her feedback was that the GUI looked too simple and bland, and it would benefit from images and color to make it more visually appealing. If I had more time, I would add those features, improve error handling, and implement additional functionality, such as filtering.

Error fix from crash my code challenge:

## Fixed code:

```
// Validates that the input is a valid name (non-empty and only alphabetic characters)
private String validateName(String input, String fieldName) throws Exception {
    // Check for null or empty input
    if (input == null || input.trim().isEmpty()) {
        throw new Exception(fieldName + " cannot be empty.");
    }

    // Check for length constraint (e.g., max 50 characters)
    int maxLength = 50; // You can adjust the max length as needed
    if (input.length() > maxLength) {
        throw new Exception(fieldName + " cannot be longer than " + maxLength + " characters.");
    }

    // Check for only alphabetic characters
    if (!input.matches("[a-zA-Z]+")) {
        throw new Exception(fieldName + " must contain only alphabetic characters.");
    }

    return input;
}

// Validates that the input is not empty (used for skills or any non-numeric field)
```

## Fixed output:

The screenshot shows the 'MainGUI' application window titled 'Freelance Job Management System'. On the left, a list of freelancers is displayed:

- All Freelancers:
- ID: 1, Name: Aryan Desai, Skillset: Coding, Hourly Rate: \$25.0
- ID: 2, Name: Eric Matters, Skillset: python, Hourly Rate: \$12.0
- ID: 3, Name: Eric Jose, Skillset: Java, Hourly Rate: \$35.0
- ID: 4, Name: Jon Jones, Skillset: python, Hourly Rate: \$25.0
- ID: 5, Name: Jon Burrows, Skillset: python, Hourly Rate: \$25.0
- ID: 6, Name: Joe Burrows, Skillset: python, Hourly Rate: \$25.0
- ID: 121, Name: hi t, Skillset: N/A, Hourly Rate: \$8.0

On the right, there is a form for adding a new freelancer with the following fields:

- Freelancer ID: 495
- First Name: voerboren
- Last Name: davis
- Skill Set: python
- Hourly Rate: 5894

A 'Submit' button is located below the form. An 'Error' dialog box is open in the center, displaying a red octagonal warning icon and the message: 'First Name cannot be longer than 50 characters.' with an 'OK' button.

At the bottom of the application window, there is a row of buttons: Create Freelancer, Update Freelancer, Delete Freelancer, View Freelancers, Create Client, Update Client, Delete Client, View Clients, Create Contract, Update Contract, Delete Contract, View Contracts, View Skills, and Exit.