

---

**<RST & Maybe G>**

---

**Online Turk System  
Software Design Report  
For Web Application**

**Created by:  
Rod Ta, Sunny Liang, Timothy Lee**

# Table of Contents

I.	Introduction	
	A. Purpose	2
	B. Scope	2
	C. Definitions	2
II.	Use Cases	
	A. Overall Collaboration Class Diagram	3
	B. Sign In	3
	C. Browse and Search Information	4
	D. Protest and Review Warning	5
	E. Post Demand and Hire Developer	6
	F. Bid	7
	G. Deliver System	8
	H. Manage Account	10
	I. Provide Rating	11
	J. Change or leave rating	12
	K. Account Creation	14
III.	E-R Diagrams	14
IV.	Detailed Design	
	A. Visitor Methods	15
	B. Client Methods	15
	C. Developer Methods	16
	D. Super-User Methods	17
	E. Client, Developer, and Super-User Methods	17
	F. Visitor, Client, Developer,Super-User Methods	17
	G. Client and Developer Methods	18
	H. System Methods	18
V.	System Screens	19
VI.	Group Meetings	42
VII.	1st Phase Report Responses to comments	43

## I. Introduction

### A. Purpose

1. The purpose of this Design Report is to present the data structure and logic of the functionalities described within our Software Requirement Specification report based off the online turk system.

### B. Scope

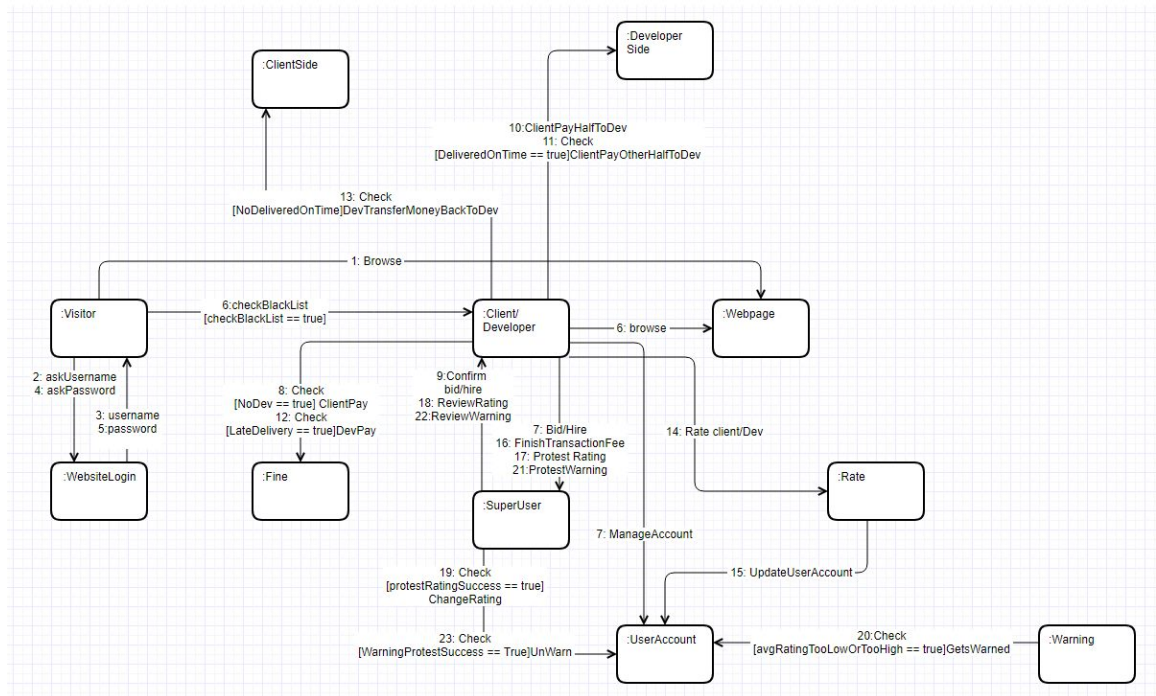
1. This software we will be developing will be an online bidding system for hiring developers. Members on the website can choose to be clients or developers. Visitors can just browse around and search public information made available by clients/developers. Developers seeking for work can bid their price for the specific task to a specific client's system demand. Clients then can choose which developer s/he wants to hire from all the current bidding developers. There will be a rating system applied to both the clients and developers to inform others how reliable the client/developer is. If the client/developer's rating are too low, there will be warnings and consequences handled by the super-user.

### C. Definitions, Acronyms, Abbreviations

Terms	Definition
Visitors	Anyone who wishes to browse or search public information made available by the clients/developers.
Clients	A user who can post system demands and show bidding timeline made available to developers who may be interested in the job.
Developer	A user who can bid on a client's system demand
Super-user	Administrator of the website who handles user accounts, money related issues and proctor user activities.

## II. Use Cases

## A. Overall Collaboration Class Diagram

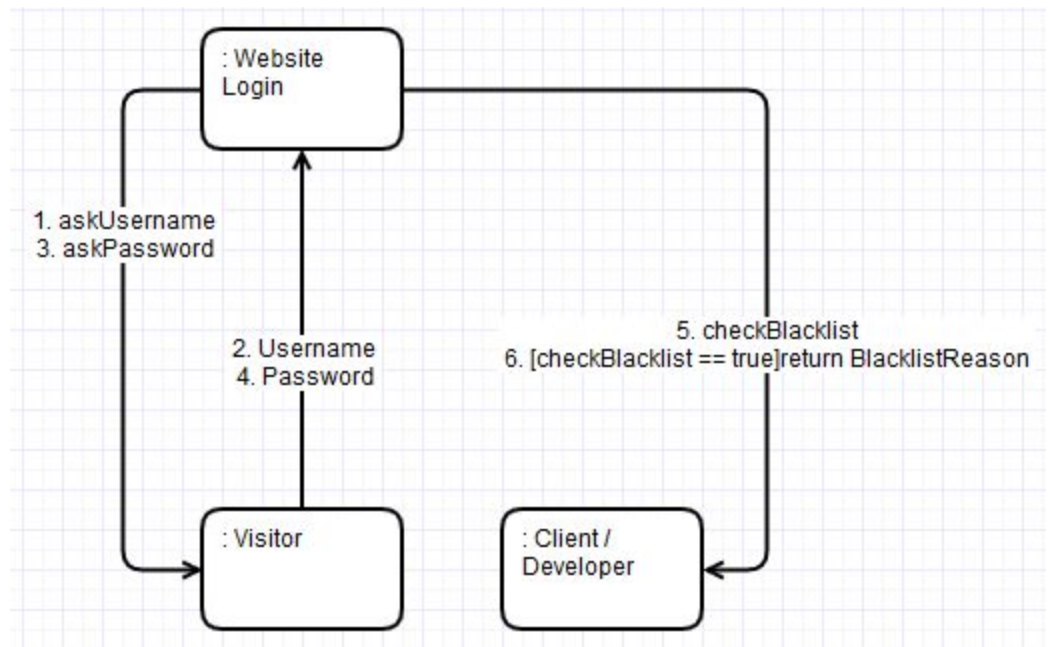


## B. Sign In

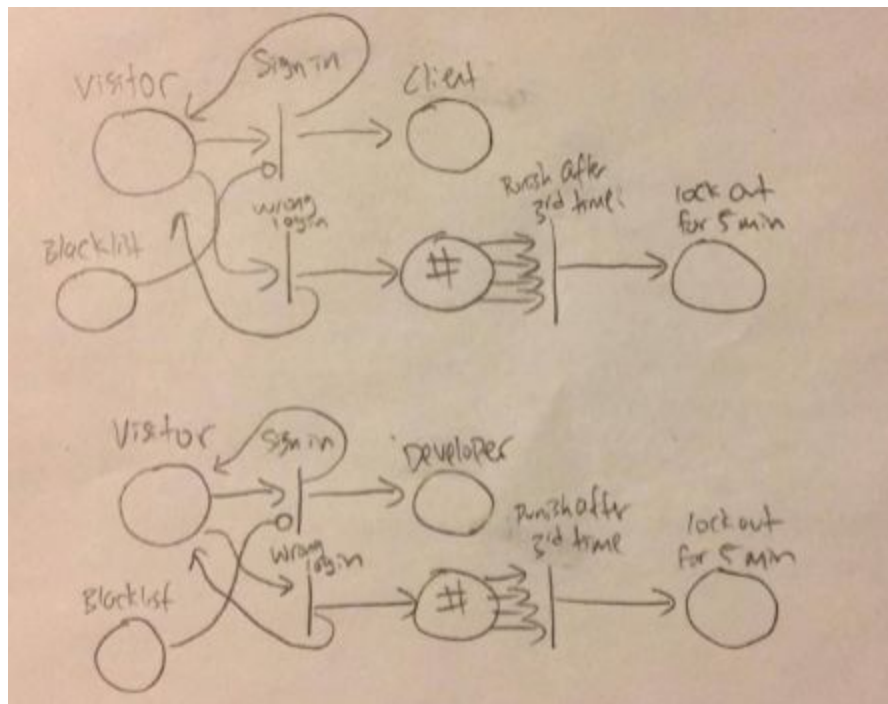
### 1. Description

- a) Clients and developers are required to login to perform specific tasks. When signing in, the system will check to see if the username entered is in the system database. If the username is not in the database, then the user will be required to signup in the signup page. The username is correct, but the password is not then the page will be redirected to the login screen, which will require the user to try logging in again. If both username and password are correct then the user is now logged in.

## 2. Collaboration Diagram



## 3. Petri - Net



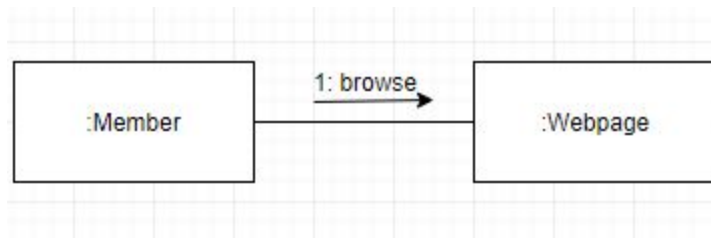
## C. Browse and Search Information

### 1. Description

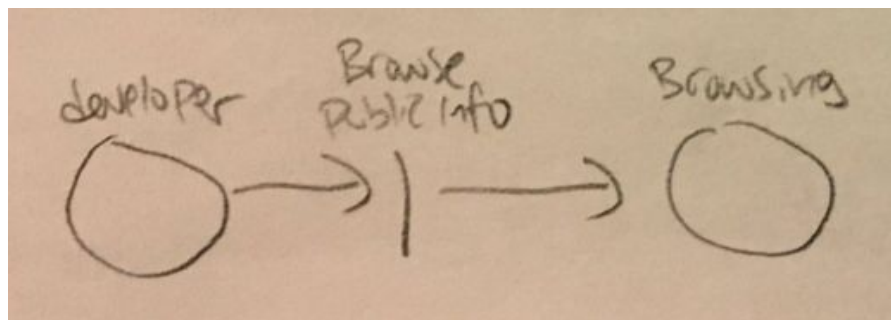
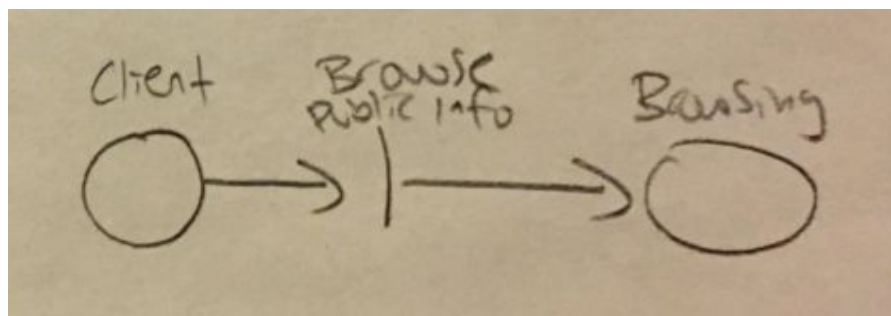
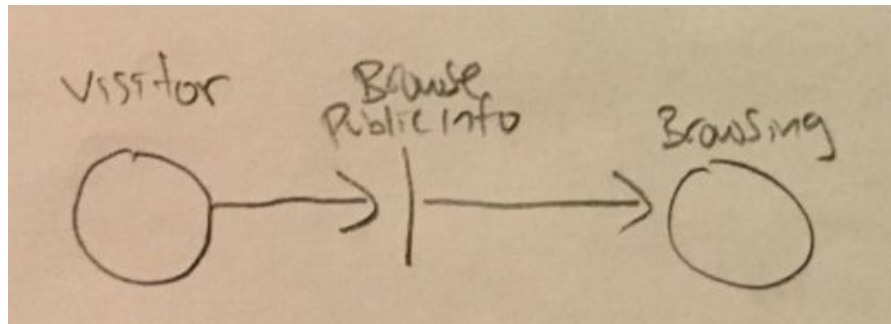
- All visitors, clients, super-user, and developers are able to browse and search through demands and user information. They will

have options to search through existing and expired demands, as well as look up a specific demand.

## 2. Collaboration Diagram



## 3. Petri-Net

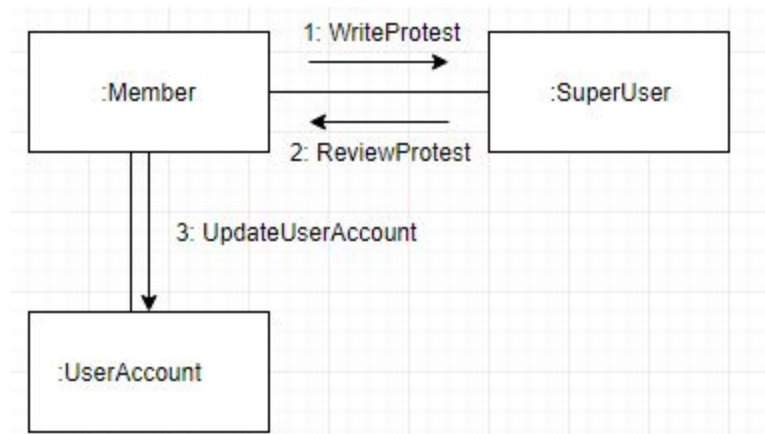


## D. Protest and Review Warning

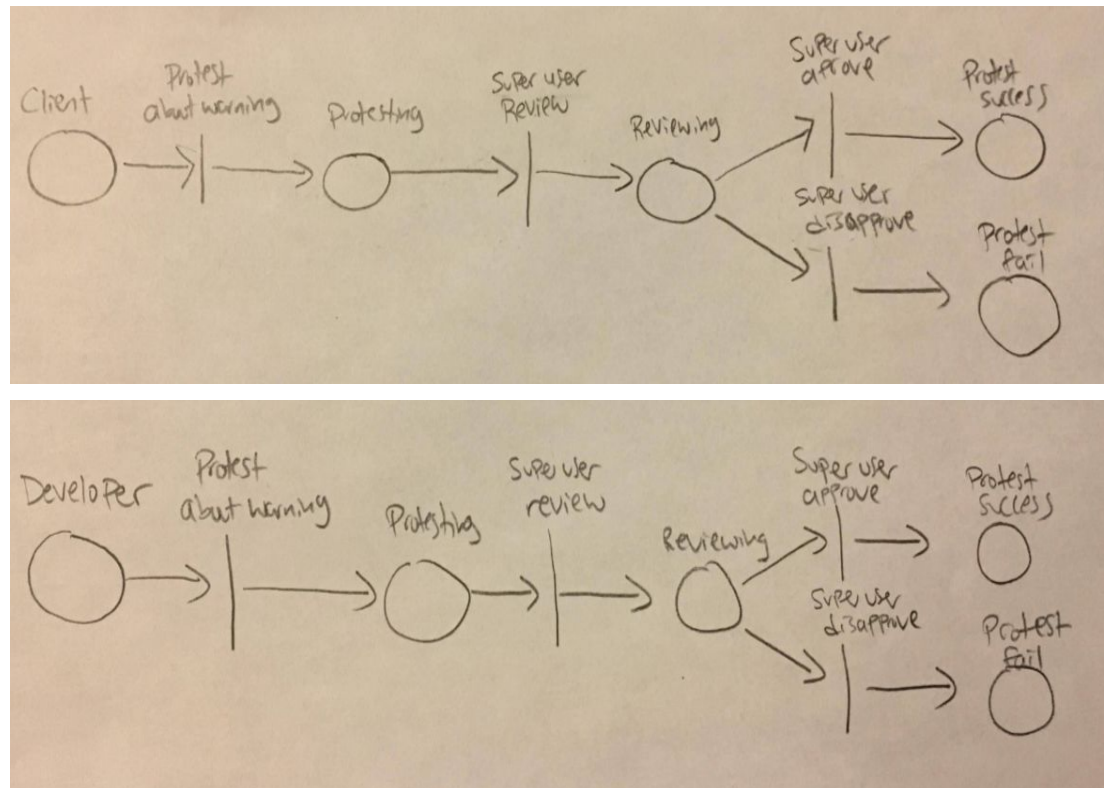
### 1. Description

- a) Clients and developers are able to protest about warnings received by the system, which will be reviewed by the super-user. Clients and developers will need to send an appeal to the super-user with their reasoning.

## 2. Collaboration Diagram



## 3. Petri-Net

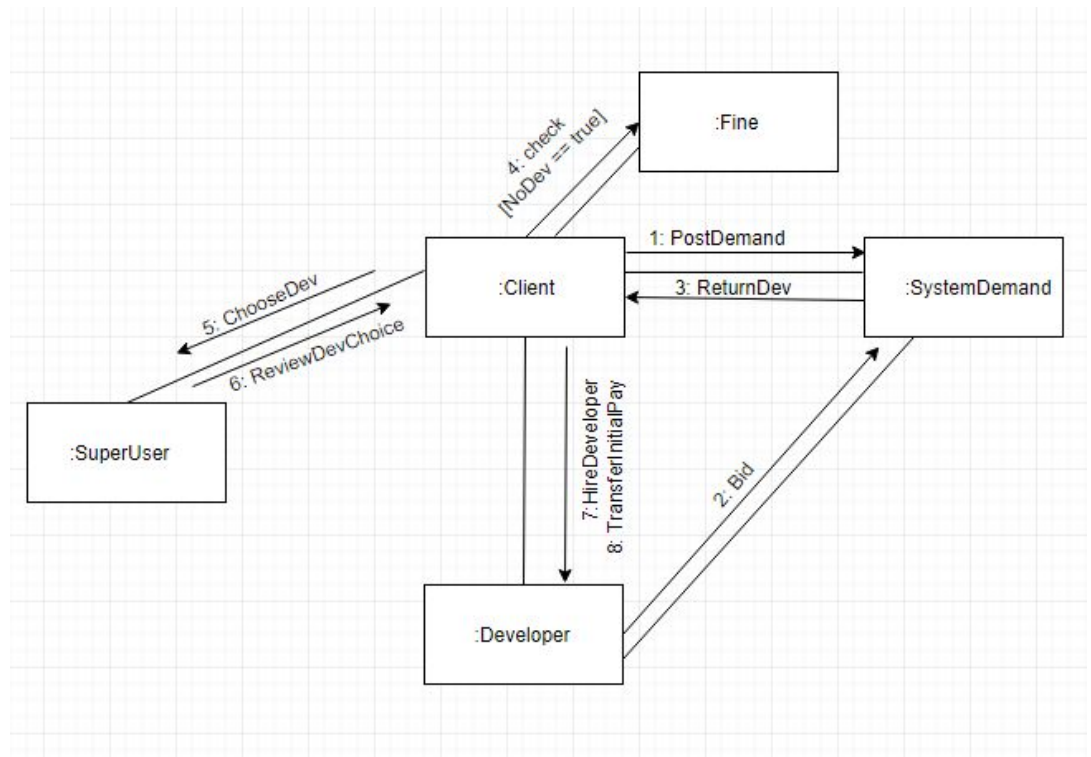


## E. Post Demand and Hire Developer

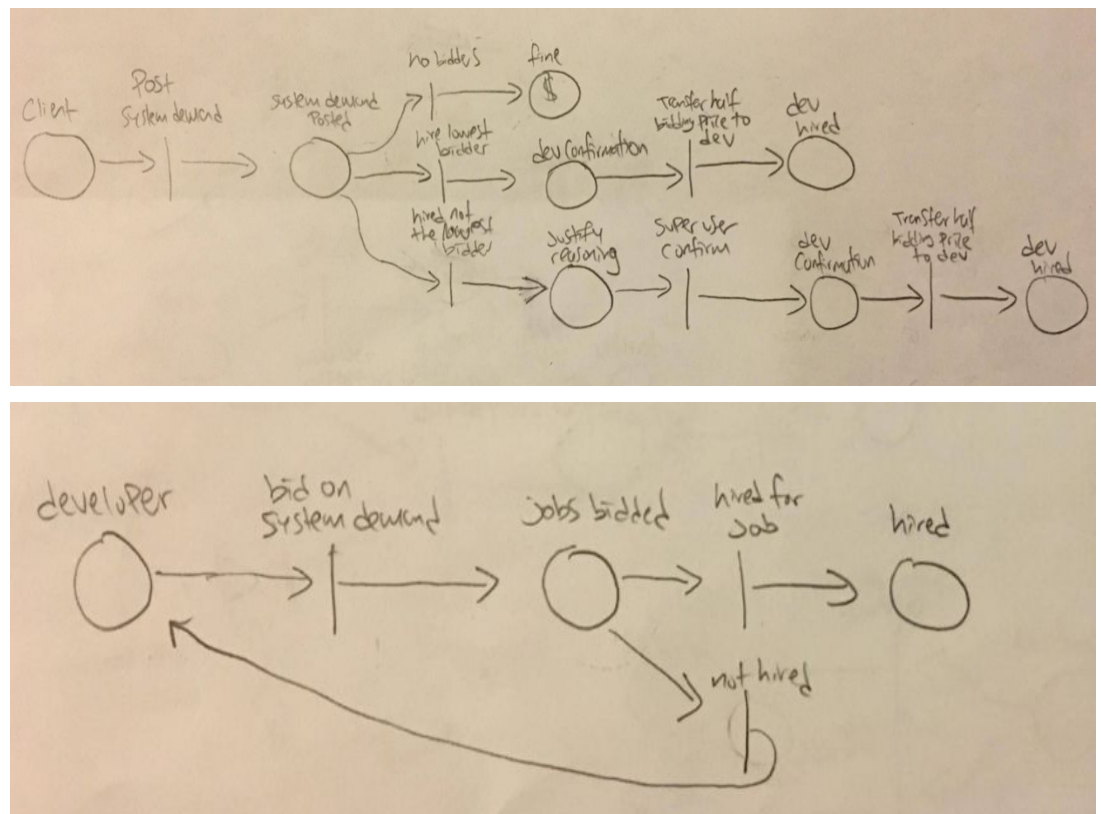
### 1. Description

- a) Clients are able to post their demands for a developer, which will include the initial pay and project description. To hire, clients will higher the lowest bidding developer or choose which developer they want. The super-user will have to review the client's choice of developer.

## 2. Collaboration Diagram



## 3. Petri-Net



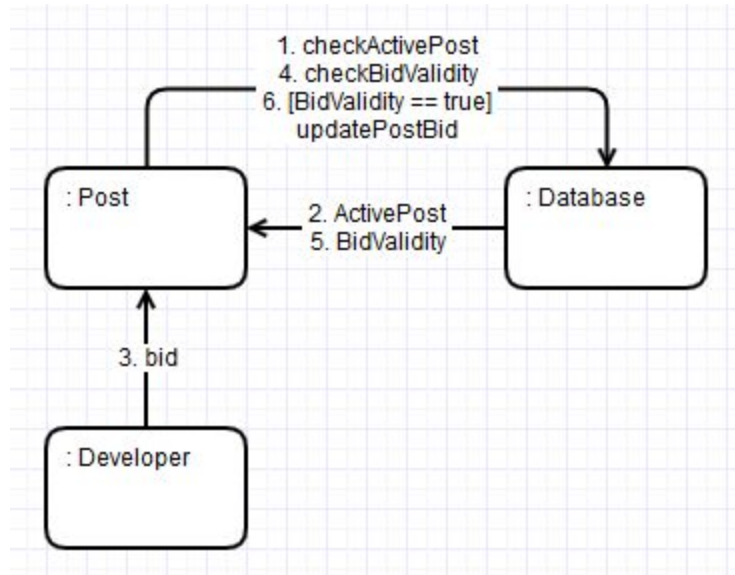
## F. Bid



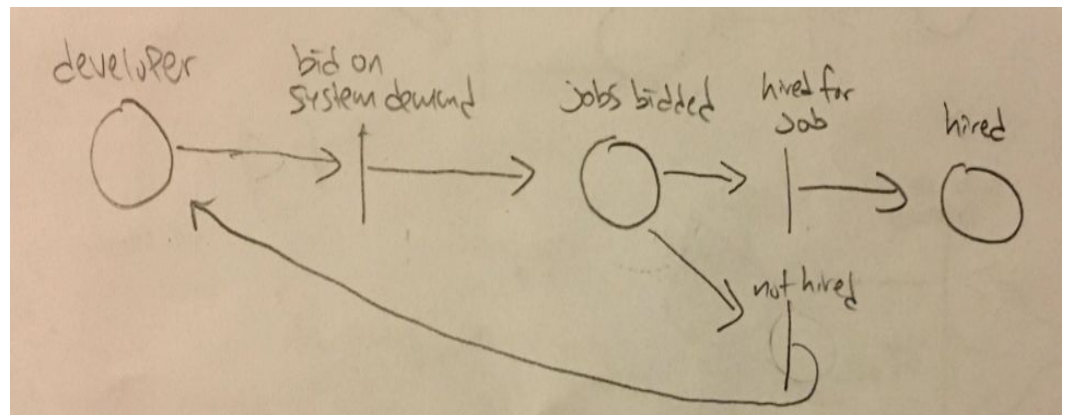
1. Description

- a) Developers are able to bid on client's demand, which will be entered into a database. They will be able to bid multiple times.

2. Collaboration Diagram



3. Petri-Net

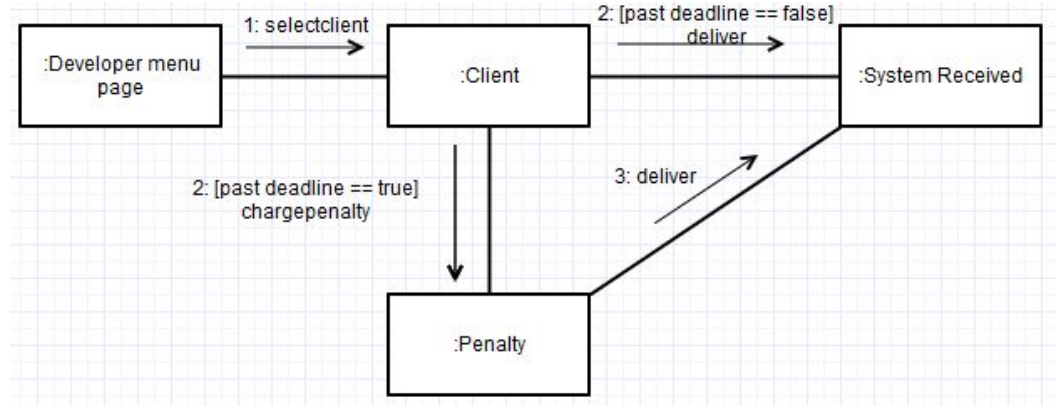


**G. Deliver System**

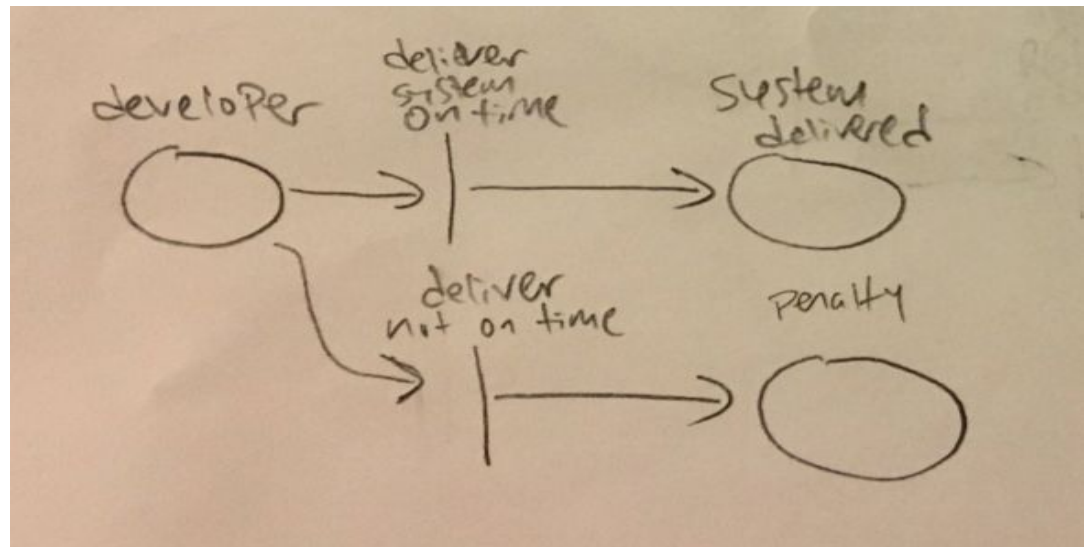
1. Description

- a) Developers are expected to deliver the system on time. If the system is not delivered on time a fixed penalty will be charged to them.

2. Collaboration Diagram



### 3. Petri-Net

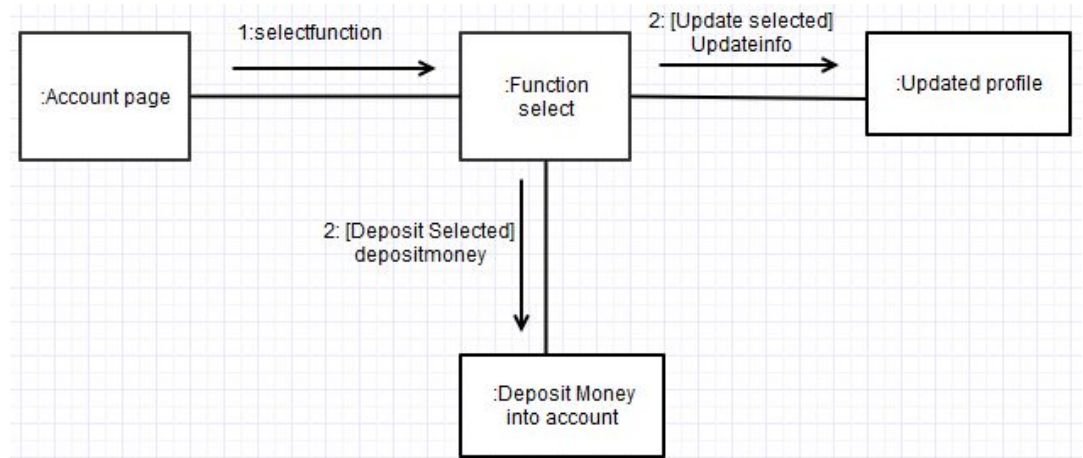


## H. Manage Account

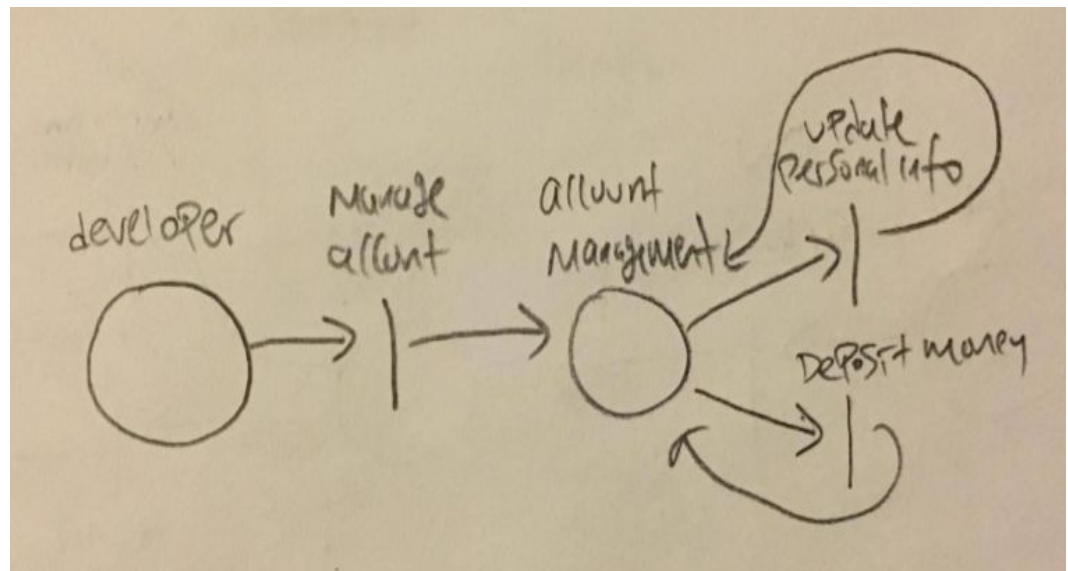
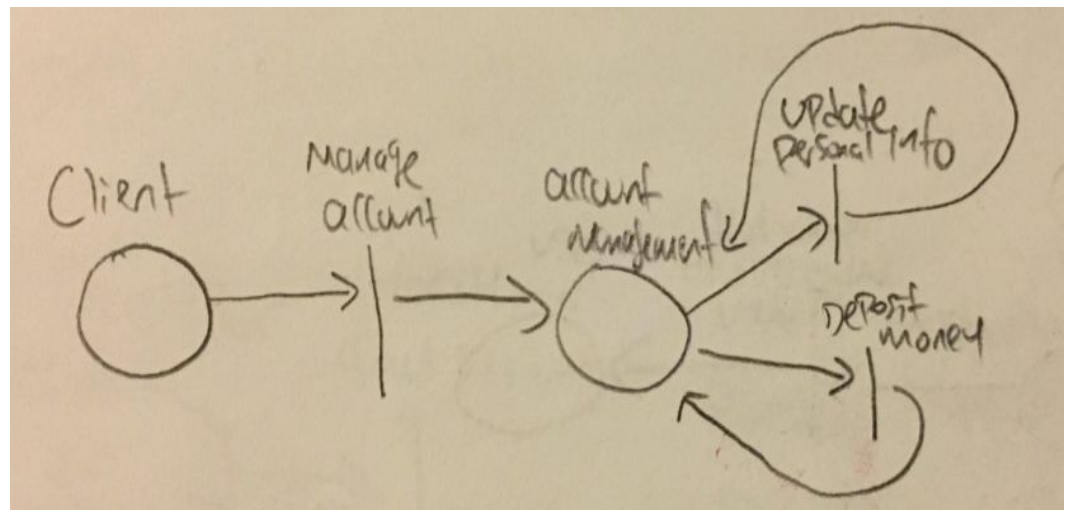
### 1. Description

- a) Clients and developers are able to manage their own accounts. The super-user will be able to manage all client and developer accounts.

## 2. Collaboration Diagram



## 3. Petri-Net

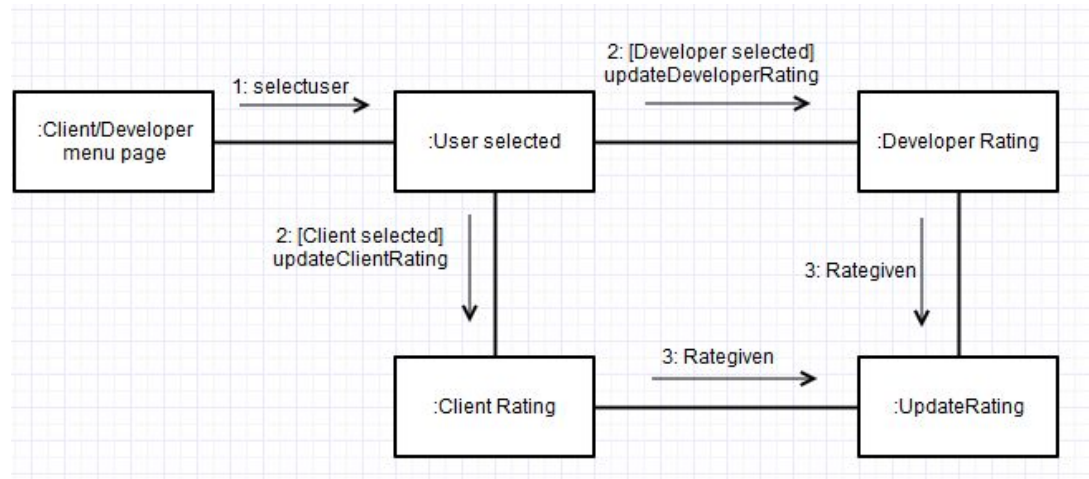


## I. Provide Rating

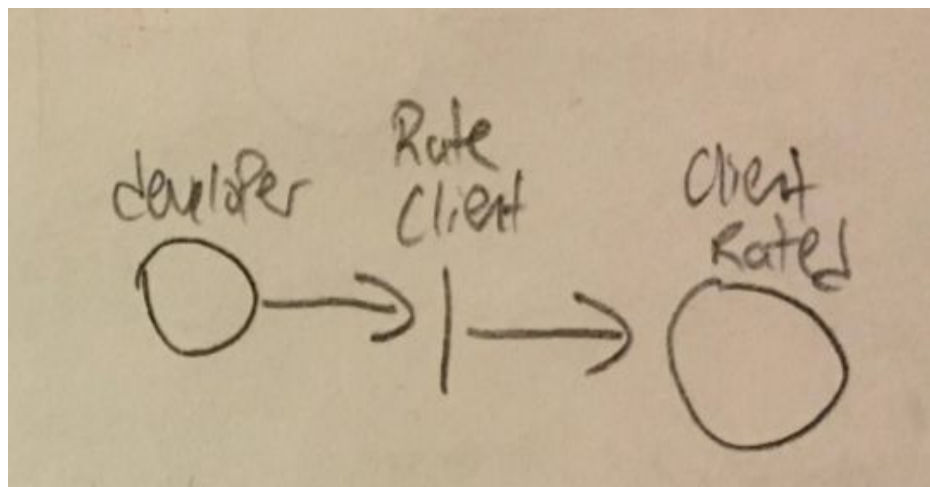
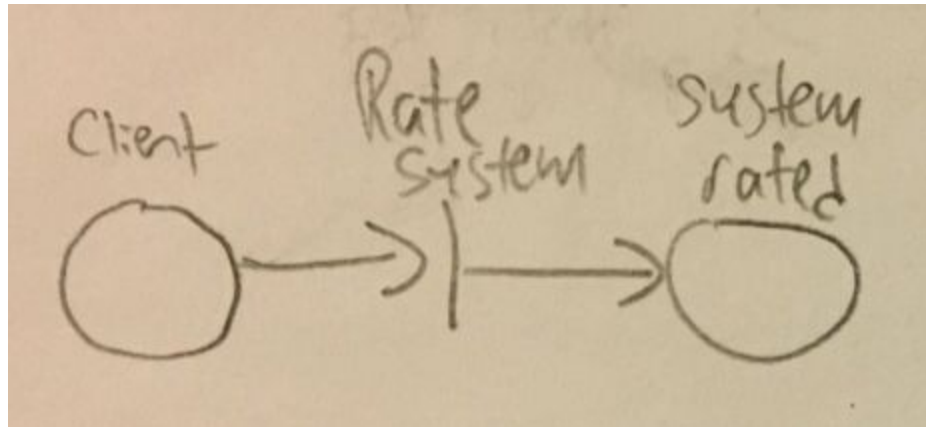
### 1. Description

- a) Clients and developers are able to provide ratings to each other, after a finished transaction. The rating will be recorded in a database.

### 2. Collaboration Diagram



### 3. Petri-Net

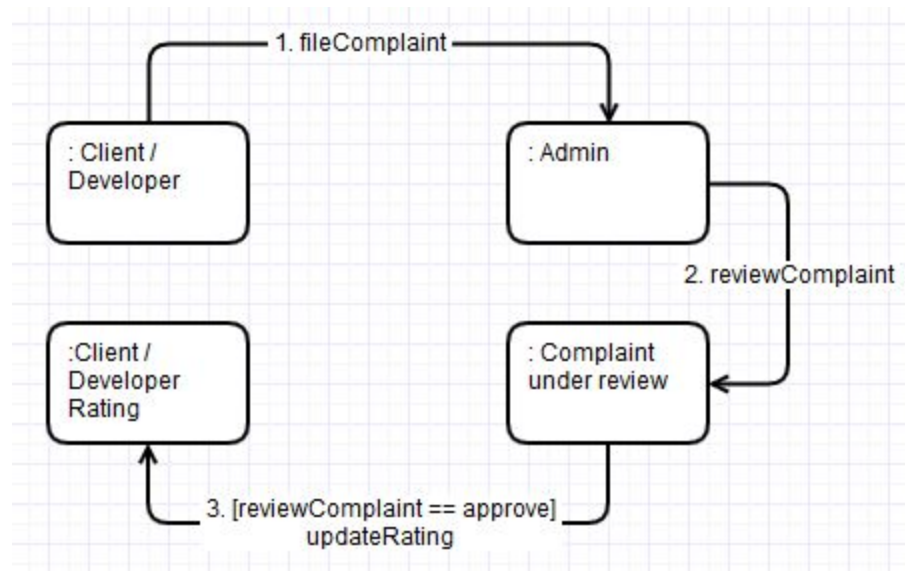


#### J. Change or leave rating

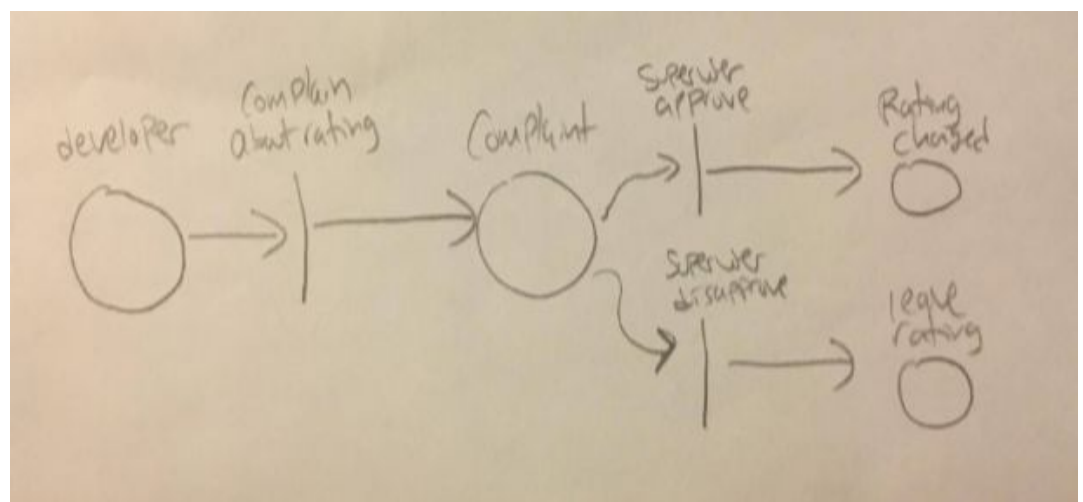
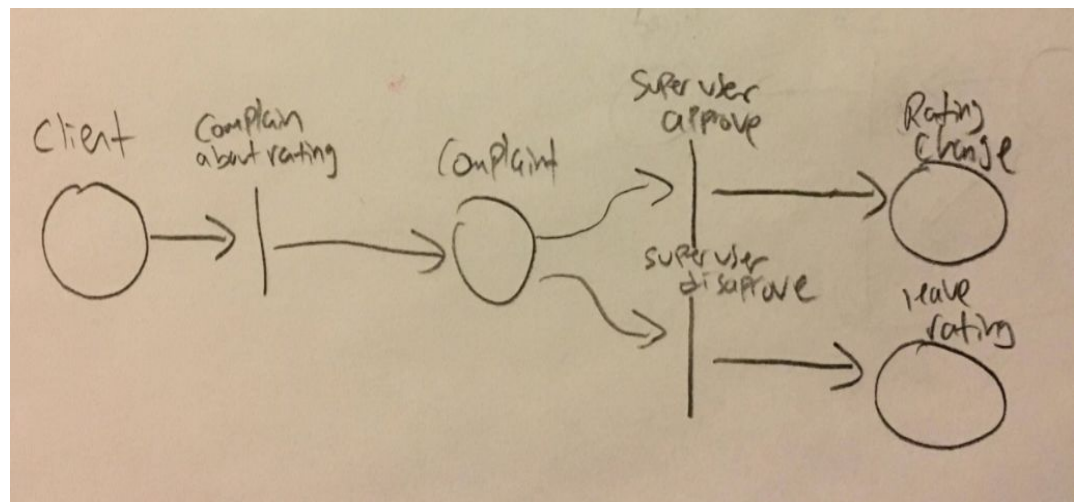
##### 1. Description

- a) The super-user has the ability to change or keep a rating based on the client's or developer's reasoning. If the super-user decides to change the rating, it will be updated accordingly in the database.

## 2. Collaboration



## 3. Petri-Net



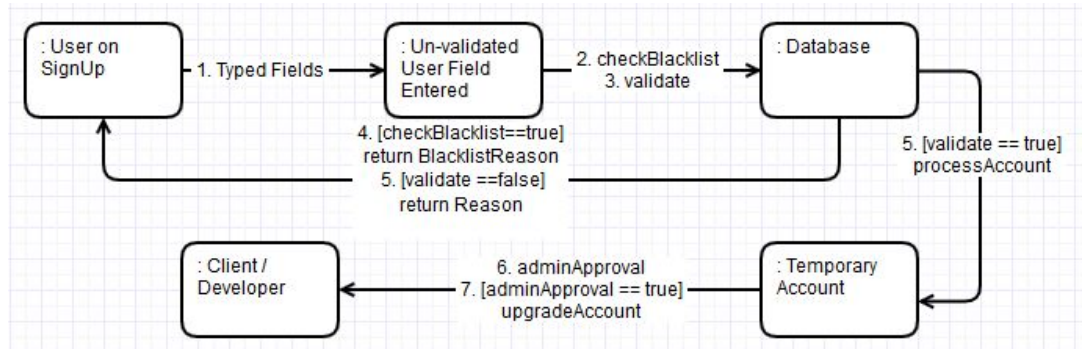


## K. Account Creation

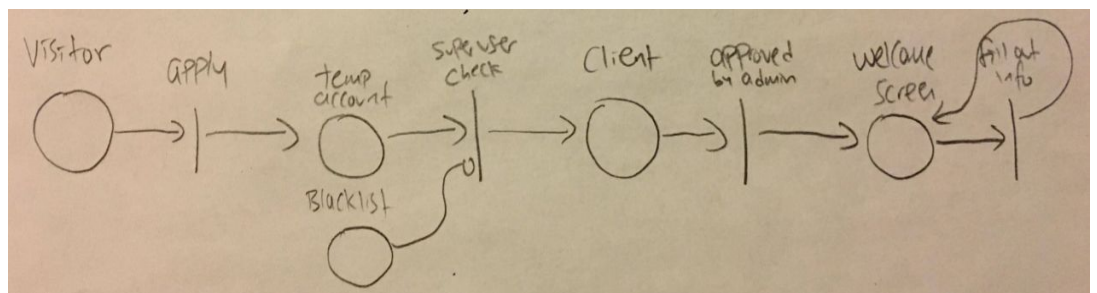
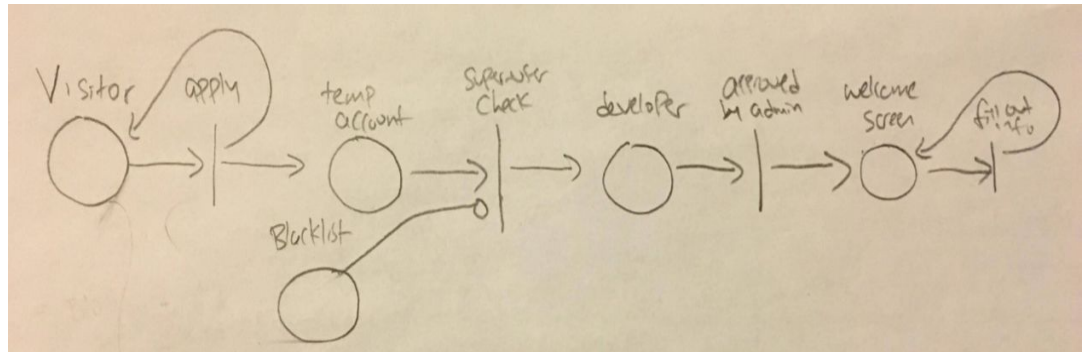
### 1. Description

- a) Visitors can sign up and apply to become either a client or developer. They must provide their basic information about themselves and deposit money to complete the application. However, if the username is created already the applicant must choose a different username. The super-user will check the applicants and promote them based on their choice of becoming a client or developer.

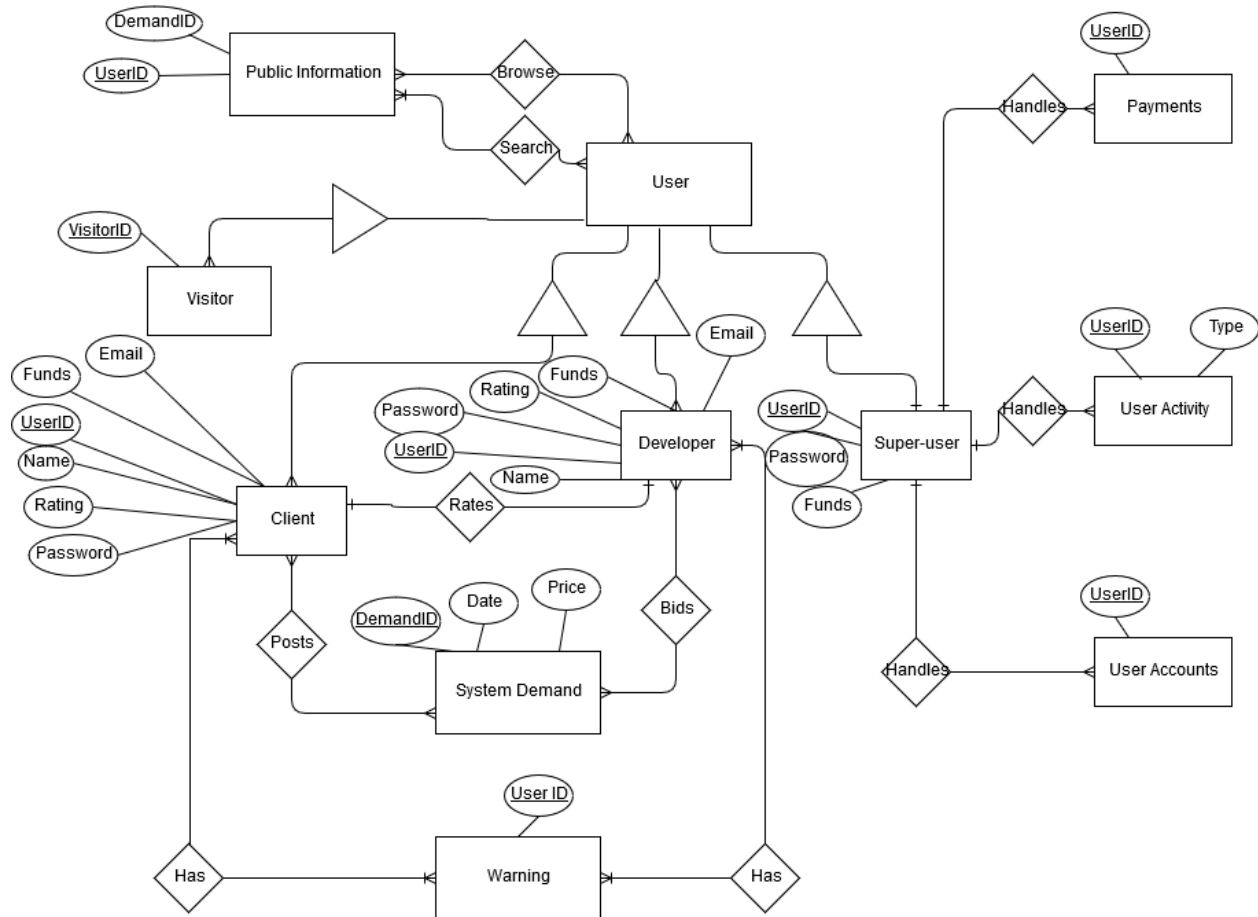
### 2. Collaboration Diagram



### 3. Petri-Net



## III. E-R Diagrams



## IV. Detailed Design

### A. Visitor Methods

#### 1. Create Account(info)

Check database(info)

If info is invalid

Send message "Enter in valid info"

Else if info exists

Send message "User already exists"

Else if info in blacklist

Send message "You are banned"

Else if info is valid and super-user approval

Insert user info into database

Render user homepage

Else

Send message ("Wait for super-user approval")

### B. Client Methods

#### 1. CreateDemand(demandinfo)



- If demandinfo is valid
    - Insert demandinfo into database
    - Render post page
  - Else
    - Return "Invalid demand info"
    - Render demand form page
- 2. RemoveDemand(userID,demandID,bid)
  - If no bid by deadline
    - Locate demandID in database and remove
    - Set user's current funds to current funds - 10 in database
- 3. Hire(userID,amount,projectID,bid)
  - If clients chooses developer and super-user approval
    - Insert user and projectID into database
    - Set client funds to current funds -  $\frac{1}{2}(\text{bid})$  in database
    - Set developer funds to current funds +  $\frac{1}{2}(\text{bid})$  in database
  - Else if lowest bidder
    - Insert user and projectID into database
    - Set client funds to current funds -  $\frac{1}{2}(\text{bid})$  in database
    - Set developer funds to current funds +  $\frac{1}{2}(\text{bid})$  in database
  - Else
    - Return message "Waiting for super-user approval"
- 4. Pay(userID, amount)
  - Set user accounts funds to current funds - amount in database

### C. Developer Methods

- 1. Bid(demandID,bid)
  - If bid is valid
    - Insert bid into database
    - Render post page
  - Else
    - Return "Try again"
- 2. RemoveBid(bid,demandID)
  - Delete bid from database
- 3. Delivery(projectID,bid)
  - If project delivered after deadline
    - Set developer funds to current funds -  $\frac{1}{2}(\text{bid})$  - fixed penalty
    - Set client funds to current funds +  $\frac{1}{2}(\text{bid})$

Set rating to 1  
Else  
Set super-user funds to current funds +  $\frac{1}{2}(\text{bid})$

#### **D. Super-User Methods**

1. ReviewWarningAppeal(appealID,userID)  
If super-user reverse warning  
Decrease user's warning by 1  
Else  
User's warning count remains the same
2. ReviewHire(userID, demandID,projectID)  
If super-user approves developer  
Insert userID and projectID into database  
Else  
Send message to client "Sorry choose another developer"
3. ChangeRating(projectID,userID)  
If super-user decides to change rating  
Update rating in database  
Else  
Do not change rating in database

#### **E. Client, Developer, Super-User Methods**

1. Login(user,password)  
Check database(user and password)  
If username and password exists  
Render homepage  
Else  
Return message "Retry login"  
Render login page
2. Logout(user)  
Login session set to false  
Current user session is set to none  
Render homepage

#### **F. Visitor, Client, Developer,Super-User Methods**

1. Browse(demands)  
Return all demands
2. Search(demandID)

If demand found in database  
     Return demand  
 Else  
     Return message "not found"

3. SearchUser(userID)  
     If userid found in database  
         Return user info, ratings, bids, demand  
 Else  
     Return "user not found"

### **G. Client and Developer Methods**

1. ManageAccount(userID)  
     Update user info in database

2. Deposit(userID, amount)  
     Set user accounts funds to current funds + amount in database

3. GiveRating(userID)  
     If client gives rating  $\geq 3$   
         Add rating to developer in database  
         Set super-user funds to current funds -  $\frac{1}{2}(\text{bid})$  in database  
         Set developer funds to current funds +  $\frac{1}{2}(\text{bid})$  in database  
 Else  
         Add reasoning to database  
         Send message to super-user

    If developer gives rating  $\leq 2$   
         Add rating and reasoning to database  
 Else  
         Add rating to database  
         Send message to super-user

4. Appeal(Warning.userID)  
     Add users appeal to database  
     Send message to super-user

5. CloseAccount(userID)  
     Remove all userID info from database

### **H. System Methods**

1. Warning(userID)  
     If average project rating  $\leq 2$  when projects  $\geq 5$

Increase user's warning by 1

2. WarningRatingstoOthers(UserID, RatingsGiven)

If average of RatingsGiven <2 or >4 when projects >=8

Increase user's warning by 1

3. ThrowUser(UserID)

If user warnings equal 2

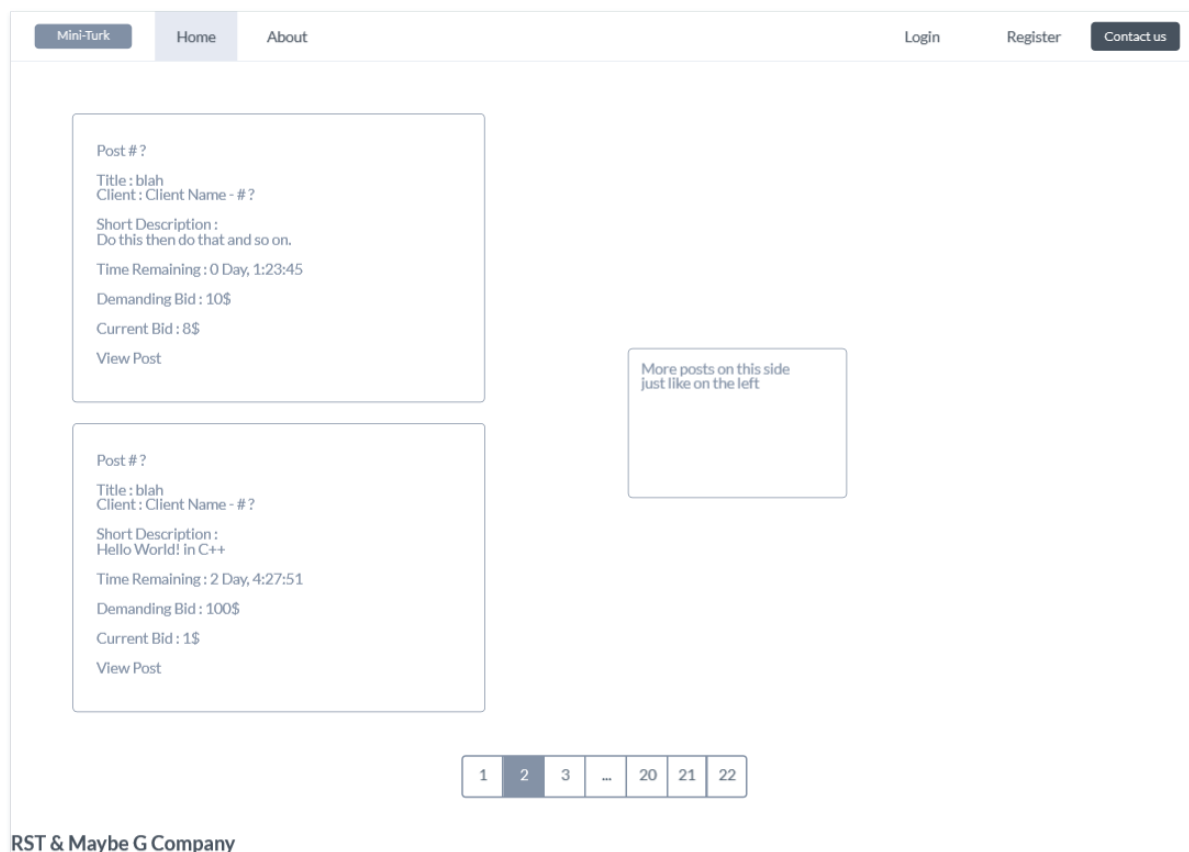
Send message to user

"You are now banned from the system"

Insert userID into blacklist in database

## V. System Screens

### A. Main Page



## B. Register

Logo Home About Features Pricing Contact us

### Sign up

**Name**  
Your name

**Email**  
Your email

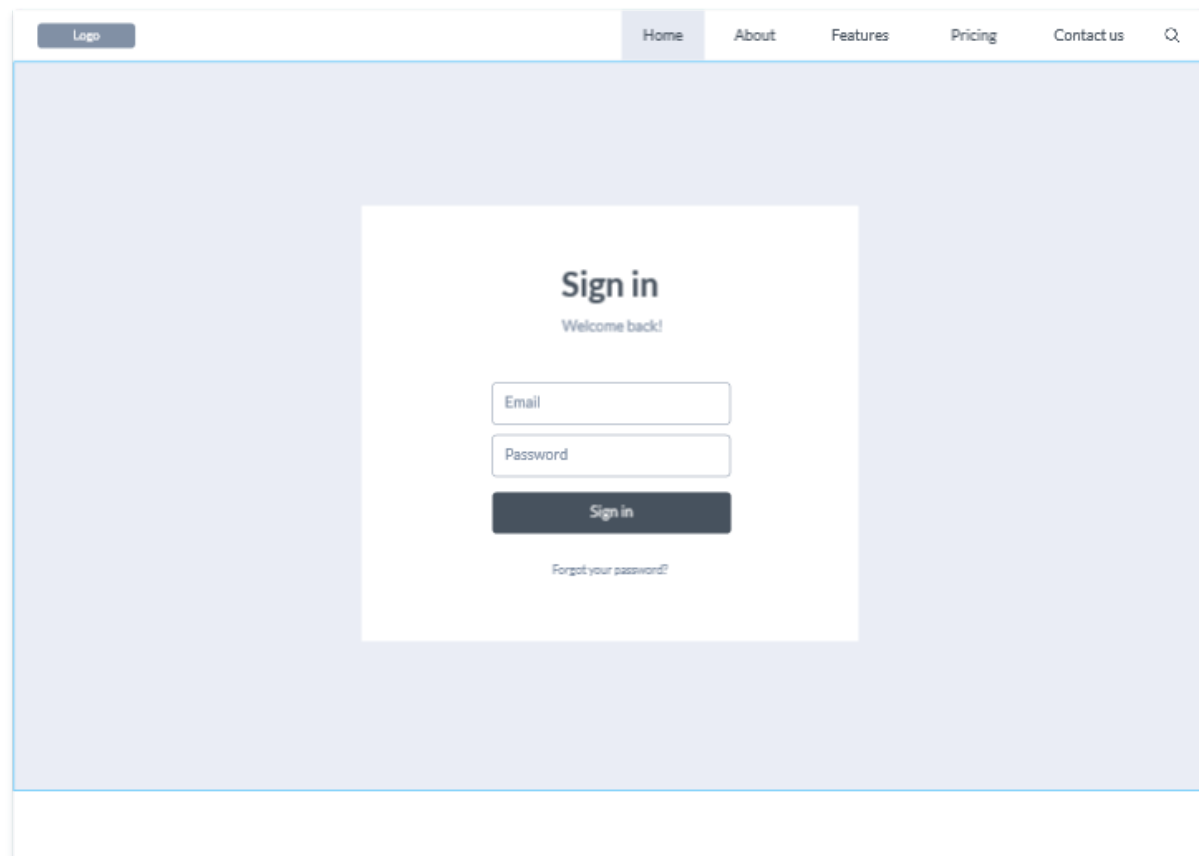
**Password**  
Your password

**Credit Card**  
Your credit card #

Pick Developer or Client

Create my account

## C. User Login



The image shows a user login form within a web application interface. The form is centered on a light blue background. At the top of the page, there is a navigation bar with a 'Logo' button and links for 'Home', 'About', 'Features', 'Pricing', and 'Contact us'. The 'Home' link is currently selected. The login form itself is a white card with the title 'Sign in' and a subtitle 'Welcome back!'. It contains two input fields: 'Email' and 'Password'. Below these fields is a dark blue 'Sign in' button. At the bottom of the form, there is a link that says 'Forgot your password?'. The entire form is enclosed in a thin blue border.

Logo Home About Features Pricing Contact us

### Sign in

Welcome back!

Email

Password

Sign in

[Forgot your password?](#)

## D. Sample Post

[Login](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[🔍](#)

### Post # 1337

Title : Quick Select  
Client : {name} - #  
Time Remaining : 0 Day, 1:00:00  
Demanding Bid : 203  
Current Bid : 53  
Current Bidder : Dev{Name} - # ?

Client's Basic Info  
Name : John Doe  
Email : johndoe@example.com  
Rating : 4.3 / 5

[View Profile](#)

[Place Bid](#)

Description  
Create a Python module that does Quick Select  
Input : List of numbers, nth smallest of list  
Output : Number for nth smallest of list  
You decide how it should work if list has even elements.

## E. Customer Profile

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Welcome to Client # ? Profile Page


Client's Name :  
Client's Email :  
Clients Rating :

**Posts**

Post #?  
Title : blah  
Short Description :  
Do this then do that and so on.  
Time Remaining : 0 Day, 1-23:45  
Demanding Bid : 10\$  
Current Bid : 8\$  
[View Post](#)

**In-Progress or Completed Posts**

Post #?  
Title : blahblah  
Demanding Bid : 10\$  
Winning Bid : 4\$  
[View Post](#)



**Resume**

Resume here

**Business Credential**

Credentials here

**Interests**

Interests here

Confidential

©<RST & Maybe G>, 2017

23



## F. Developer Profile


[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Welcome to Developer # ? Profile Page

Developer's Name :  
Developer's Email :  
Developer's Rating :

**Currently Bidding**  
Post # ?  
Title : blah  
Short Description :  
Do this then do that and so on.  
Time Remaining : 0 Day, 1:23:45  
Demanding Bid : 10\$  
Current Bid : 8\$  
[View Post](#)

**In-Progress or Completed Posts**  
Post # ?  
Title : blahblah  
Demanding Bid : 10\$  
Winning Bid : 4\$  
[View Post](#)



#### Resume

Resume here

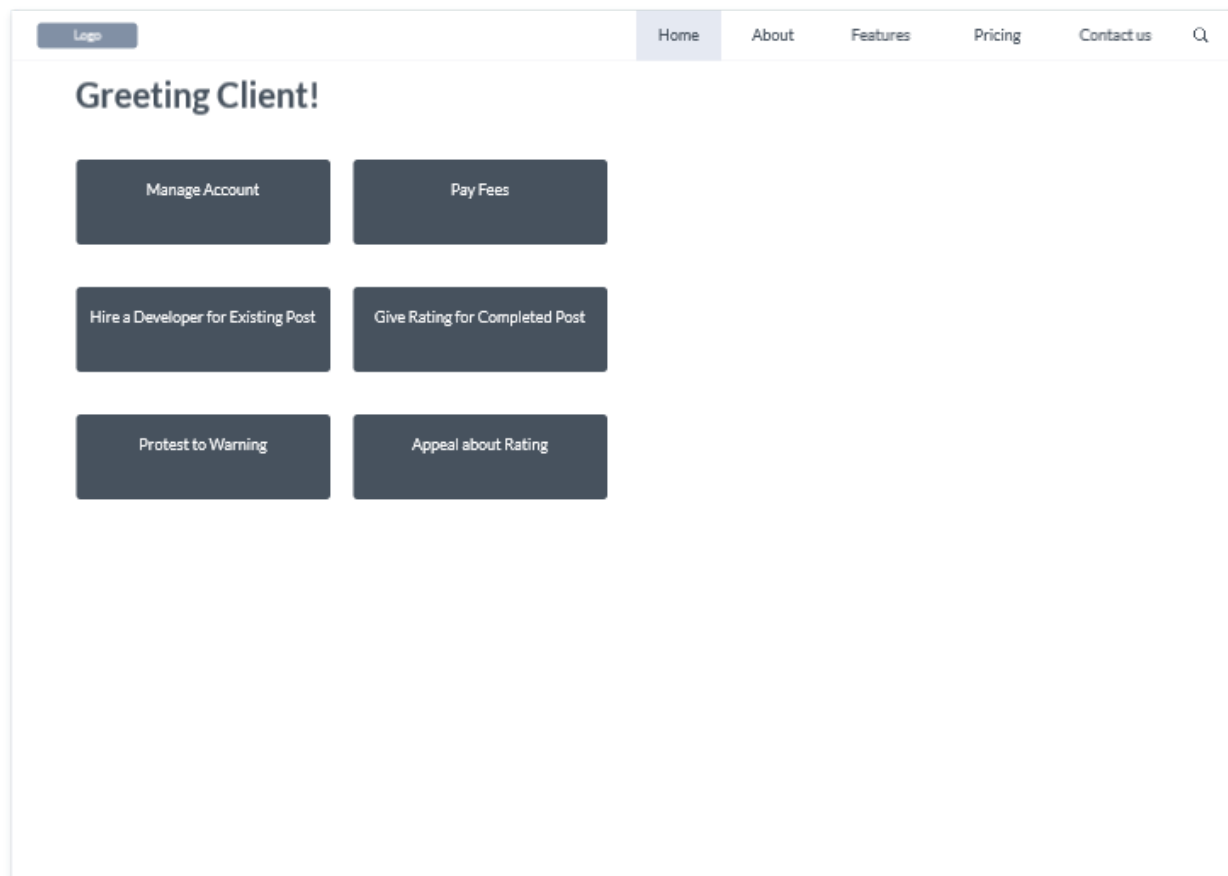
#### Sample Work

Sample Work Details here

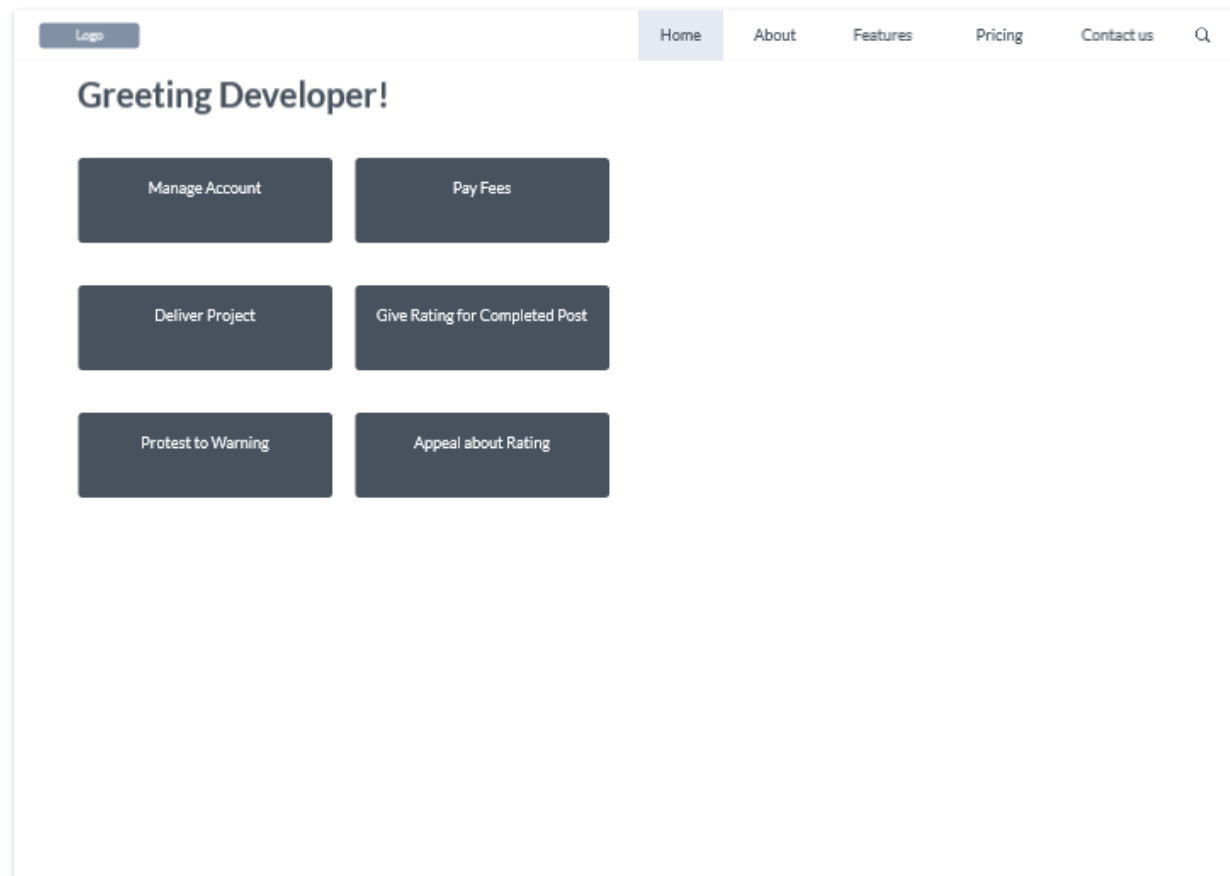
#### Interests

Interests here

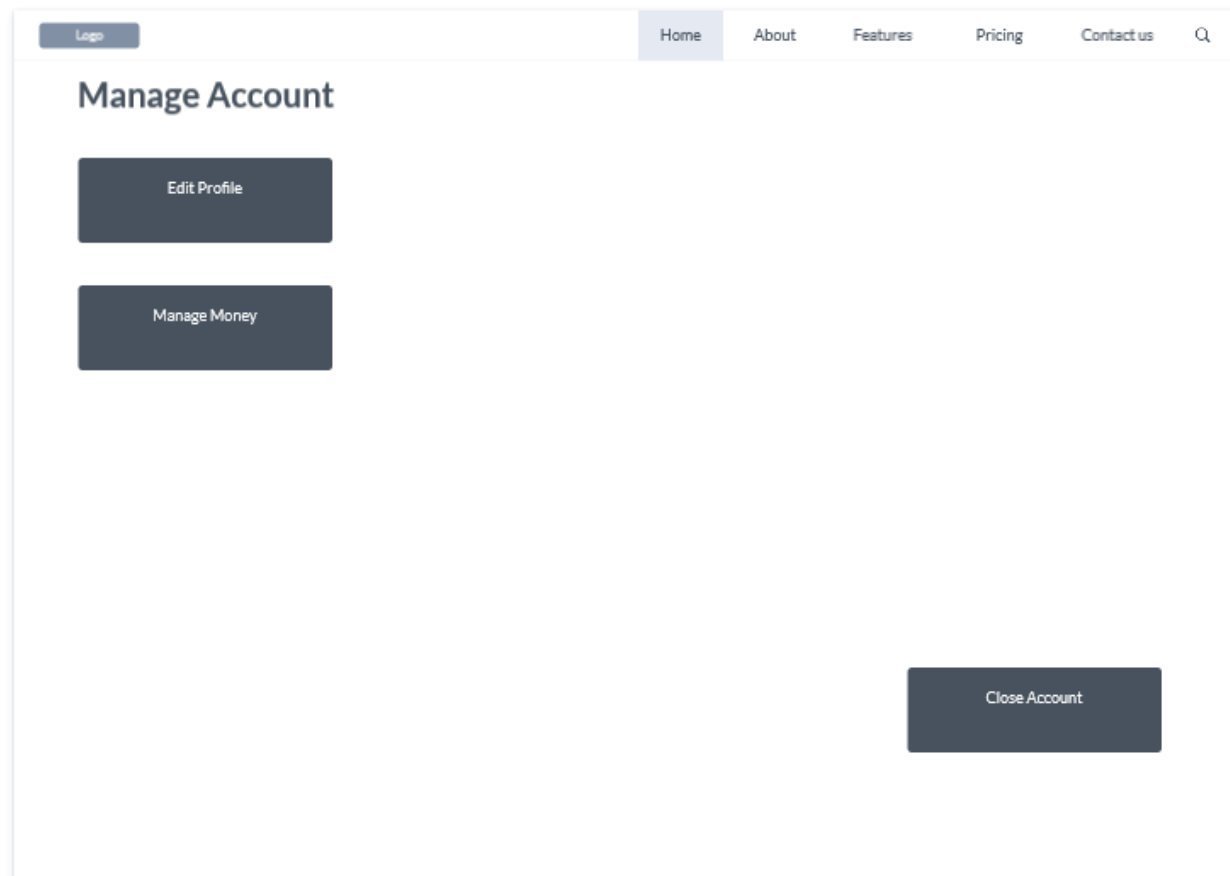
## G. Client Menu



## H. Developer Menu



## I. Client / Dev Manage Account



## J. Client Manage Profile

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Edit Profile

**Name:**

Your name

**Email:**

Your Email

**Resume:**

Resume.pdf

Browse

**Business Credential:**

Credentials

**Interests:**

Interests

blah

blahblah

Monkey

Update

## K. Developer Manage Profile

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Edit Profile

**Name:**

**Email:**

**Resume:**

[Browse](#)

**Sample Work:**

[Browse](#)

**Interests:**

Interests

blah

blahblah

blah

[Update](#)

## L. Client / Developer Manage Money

The screenshot shows a web application interface for managing money. At the top, there is a navigation bar with a 'Login' button on the left and links for 'Home', 'About', 'Features', 'Pricing', and 'Contact us' on the right, followed by a search icon. The main heading is 'Manage Money'. Below this, a box displays the current balance: 'You currently have \$ 123.45'. There are two input fields for transactions: 'Enter Deposit Amount' and 'Enter Withdraw Amount'. Each input field is paired with a corresponding action button: 'Deposit' for the deposit field and 'Withdraw' for the withdrawal field. The buttons are dark grey, while the input fields are light grey with thin borders.

## M. Client / Developer Pay Fee

The screenshot displays a web application interface for paying fees. At the top, a navigation bar includes a 'Logo' button and links for 'Home', 'About', 'Features', 'Pricing', and 'Contact us', along with a search icon. The main heading is 'Pay Fees'. Below this, a light gray box contains the text 'You currently owe \$ 5.00'. At the bottom of the main content area, there is a dark gray button labeled 'Pay Fee'.



## N. Client Hire

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Hire a Developer

Pick a Post # ▾

Post # :  
Title :  
Demanding Bid : \$ 10  
Dev # 1 Bid : \$ 10  
Dev # 2 Bid : \$ 8  
Dev # 3 Bid : \$ 5

Pick a Developer ▾

View Profile

Hire

## O. Developer Deliver

Logo

HomeAboutFeaturesPricingContact us

### Deliver a project

Pick a Post #

View Post

View Client

Post #:  
Title:  
Client: Name - #  
Demanding Bid: \$ 10  
Your Bid: \$ 5  
Description:  
blahblah

File.py

Browse File

Deliver

## P. Client / Developer Rate

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Rate your Experience

Pick a Post # ▾

View Post

View Client

View Developer

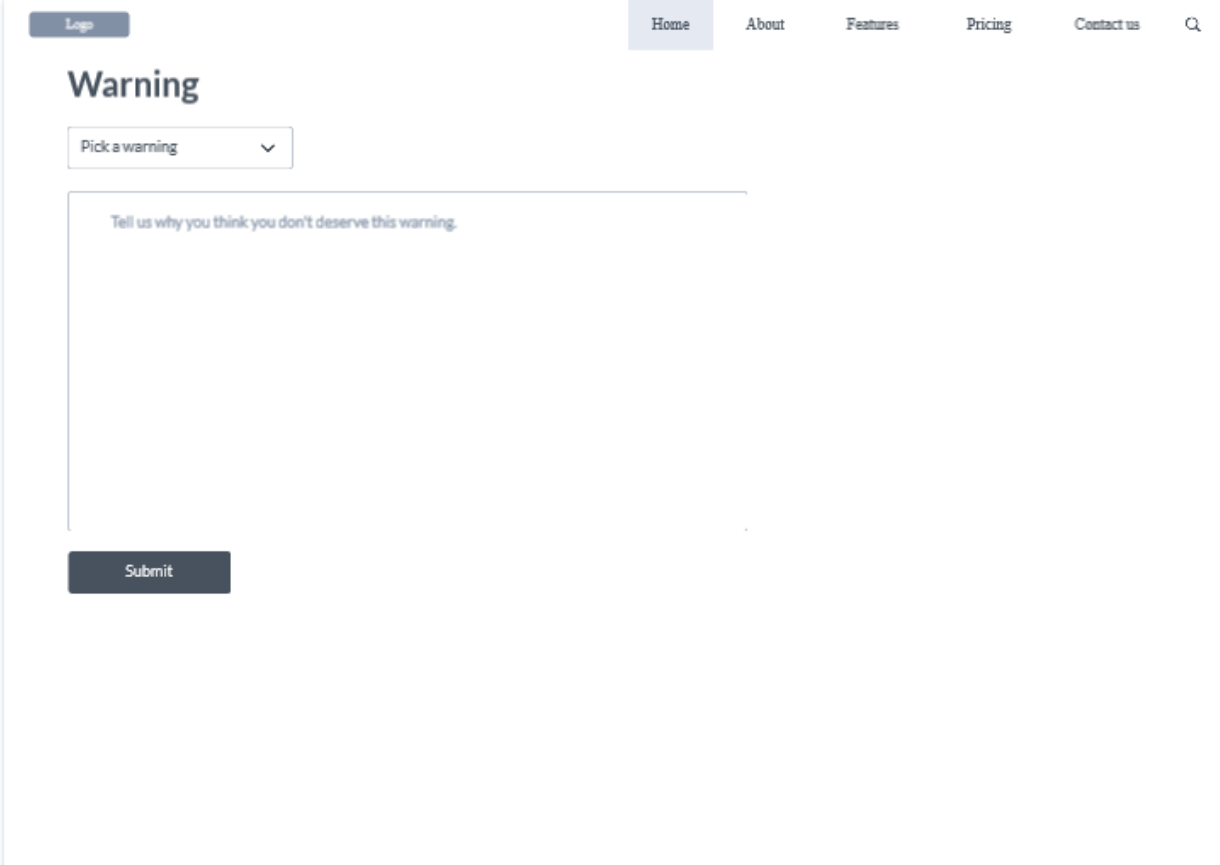
Post # :  
Title :  
Demanding Bid : \$ 10  
Winning Bid : \$ 10  
  
Client : Name - #  
Developer : Name - #

Pick a Rating from 1-5 ▾

1 is Worst  
5 is Best

Rate

## Q. Client / Developer Warning



The image shows a web form titled "Warning". At the top, there is a navigation bar with links: "Logo", "Home", "About", "Features", "Pricing", "Contact us", and a search icon. The "Warning" form itself has a dropdown menu labeled "Pick a warning" with a downward arrow. Below this is a large text area with the placeholder text "Tell us why you think you don't deserve this warning.". At the bottom of the form is a dark "Submit" button.

Warning

Pick a warning ▼

Tell us why you think you don't deserve this warning.

Submit

## R. Client / Developer Appeal Rating

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Appeal Rating

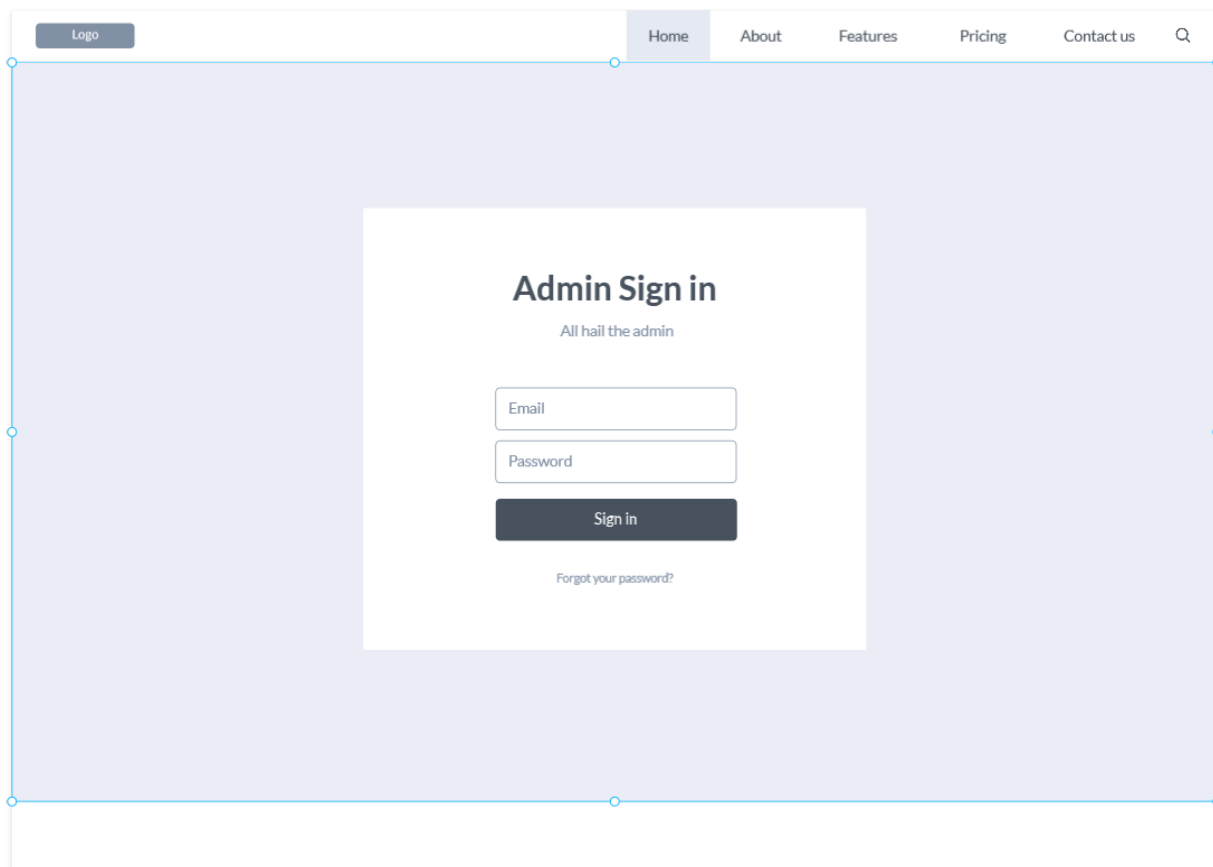
Pick a Post #

You got a rating of : #

Tell us why you think you don't deserve this rating.

Submit

## S. Admin Login



The image shows a web browser window with a navigation bar at the top. The navigation bar includes a 'Logo' button on the left and a list of links: 'Home', 'About', 'Features', 'Pricing', 'Contact us', and a search icon 'Q' on the right. The 'Home' link is highlighted. The main content area has a light blue background. In the center, there is a white rectangular box containing the 'Admin Sign in' form. The form has the title 'Admin Sign in' in bold, followed by the subtitle 'All hail the admin'. Below the subtitle are two input fields: 'Email' and 'Password'. Under the 'Password' field is a dark blue 'Sign in' button. At the bottom of the form box is a link that says 'Forgot your password?'. The entire browser window is outlined with a blue border and has small blue circles at the corners and midpoints of the edges, suggesting it's a design tool interface.

Logo

Home About Features Pricing Contact us Q

### Admin Sign in

All hail the admin

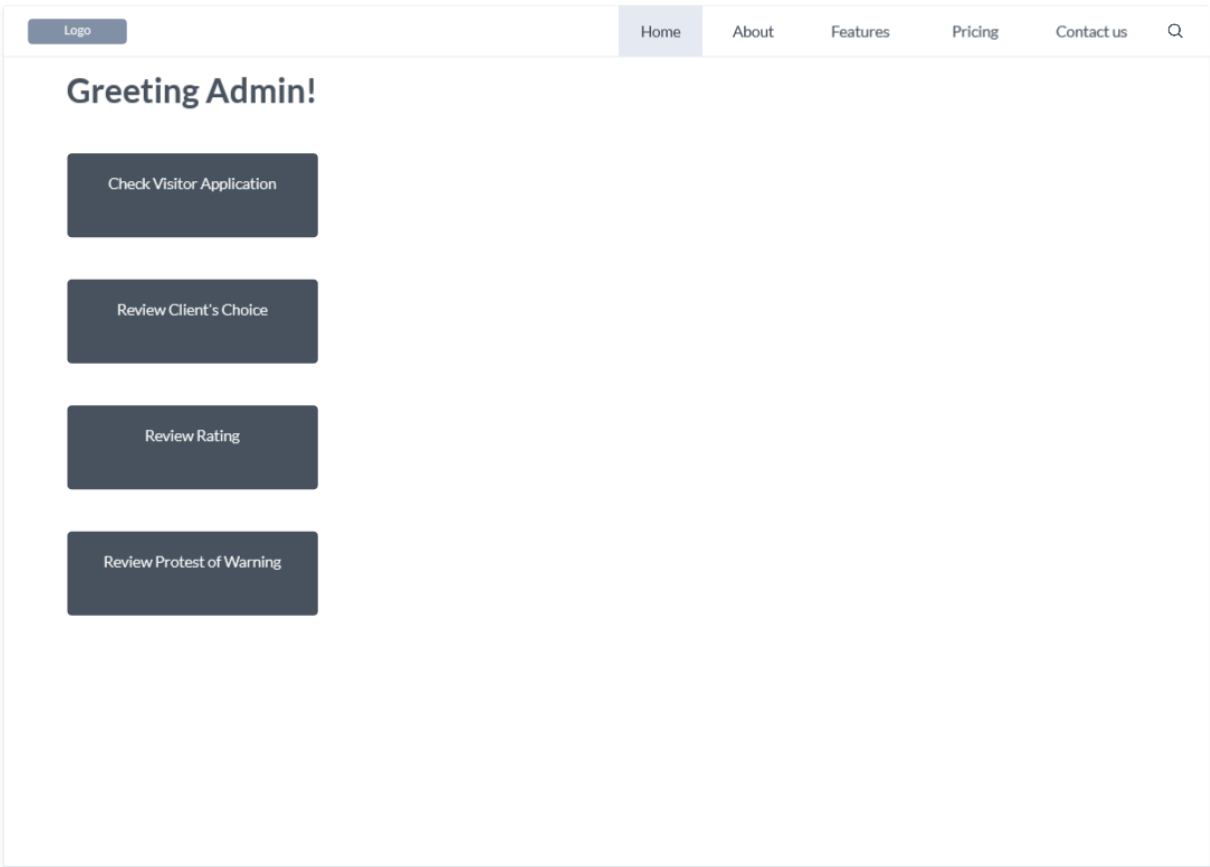
Email

Password

Sign in

[Forgot your password?](#)

T. Admin Main Page



## U. Admin - Check Visitor Application

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Check Visitor's Application

Application # ? ▾

Visitor's application info

Name

Email

Type

\$\$\$

Other stuff

More Info

Blahblah

Blah

Approve

Decline

Go Back



V. Admin - Review Client's Choice of Developer

Logo

HomeAboutFeaturesPricingContact us

# Review Client's Choice

Demand # ?

System Demand # ? :

Client # ? picked Dev # ?

Reason :

blah blah blah

Other Bidders:

Please enter message for declined choices

Approve

Decline

Go Back

## W. Admin - Review Rating

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Review Rating

Rating # ?

▼

Rating # ?  
Client # ? gave Dev # ? (or reverse) a rating of 1  
Reason:  
blah blah  
Dev # ? or Client # ? says they don't deserve it.  
Reason:  
blah blah

Enter reason for changing or leaving rating.

Pick a rating from 1-5

▼

Change Rating

Leave Rating

Go Back

## X. Admin - Review Protest of Warning

[Logo](#)[Home](#)[About](#)[Features](#)[Pricing](#)[Contact us](#)[Q](#)

### Review Protest of Warning

Protest # ?

Protest # ?  
Client # ? or Dev # ?

Reason:  
I don't deserve this warning :(

Enter reason for denying appeal.

## VI. Group Meetings

Meeting #	Type	Date	Minutes	About
1	Physical - Class	20-Oct	100	Discuss: What language and framework we are using.
2	Virtual - Skype, Youtube	21-Oct	60	Learn: Flask
3	Virtual - Skype, Youtube	22-Oct	120	Scrap: Flask Learn: Django
4	Physical - Class	27-Oct	100	Create: Scenarios for use cases
5	Physical - Class	3-Nov	100	Create: Pseudocode for main functions
6	Virtual - Skype, Youtube	4-Nov	120	Learn: More Django

7	Virtual - Skype	5-Nov	60	Draft: Sketch of Collaboration Diagram
8	Physical - Library	8-Nov	60	Review: Pseudocode
9	Physical - Library	10-Nov	60	Draft: Paper sketch of wire frame
10	Physical - Library Virtual - Skype	15-Nov	240	Create: Petri-net, E-R Diagram, Collaboration Diagram
11	Virtual - Skype	16-Nov	240	Create: Wire frames, Collaboration Clean: Petri-net, E-R Diagram, Collaboration Diagram
12	Virtual - Skype	17-Nov	240	Create: Wire frames Polish: Check Grammar and Check for completion

## VII. 1st Phase Report Responses to comments

1st Phase Report Grade : 97

Response to comments : Changed include into extend for parts that you mentioned.

Revised Use-Case diagram

