Stark Labs

weDev Software Requirements Specification For Simplified Turk System

Version 2.0

weDev	Version: 2.0
Data Structure and Logic Specification	Date: 10/11/17
Phase II Report	

Revision History

Date	Version	Description	Author
13/10/17	1.0	First project description of weDev, simplified marketplace Turk System.	Brandon Dinh-Le, Sebastian Henriquez Mikhail Kreytser, Michelle Uy
10/11/17	2.0	Second project description of weDev. More specifically the system's data structures, logic and functionalities	Brandon Dinh-Le, Sebastian Henriquez Mikhail Kreytser, Michelle Uy

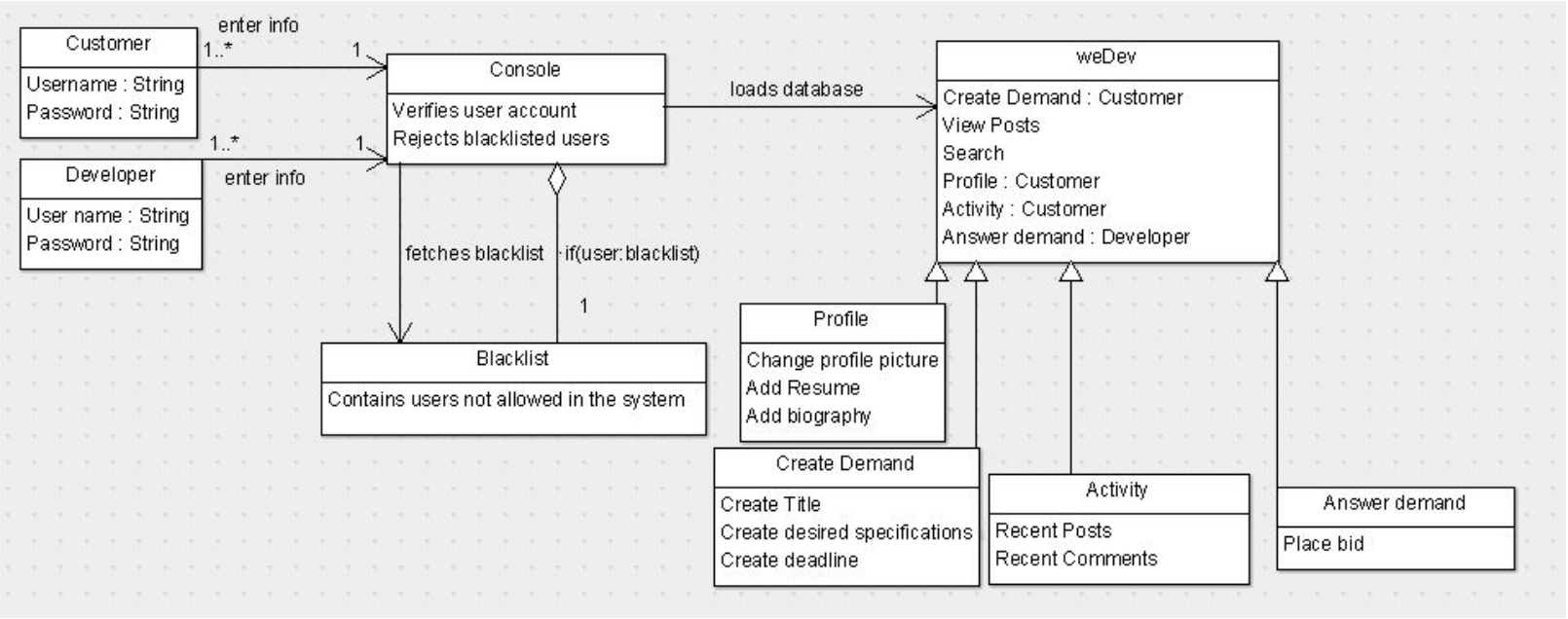
weDev	Version: 2.0
Data Structure and Logic Specification	Date: 10/11/17
Phase II Report	

Table of Contents

weDev	Version: 2.0
Data Structure and Logic Specification	Date: 10/11/17
Phase II Report	

Data Structure and Logic Specification

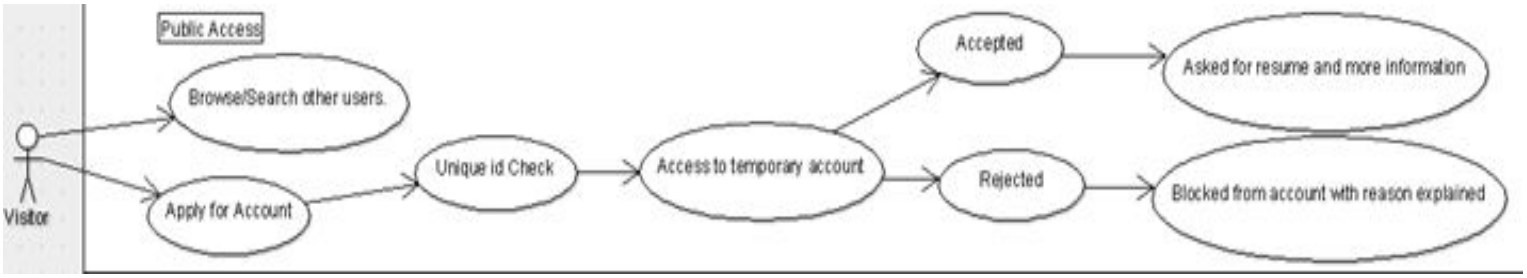
1. Introduction



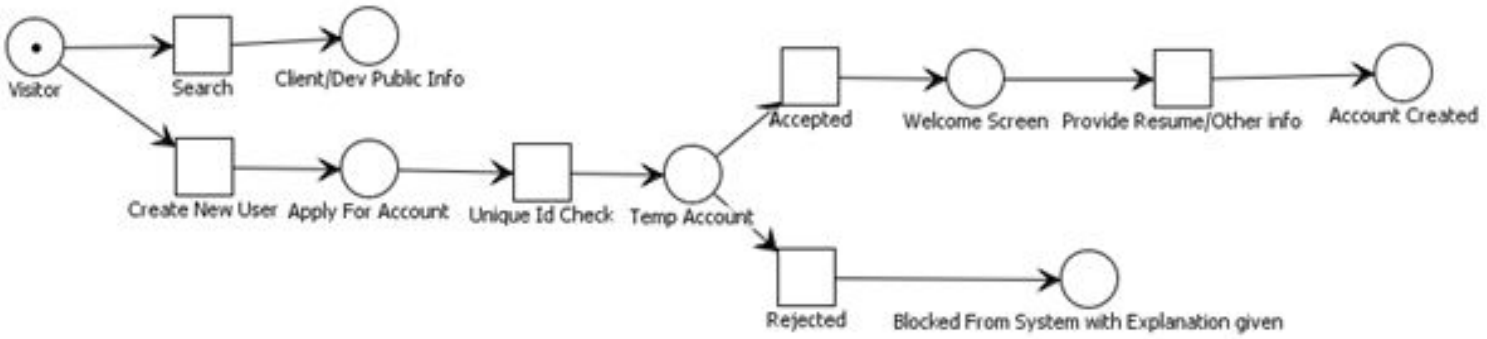
Here is a Class Diagram of the general overview of the system. This overview goes over the different activities registered user have accessed to once they log into the system and are approved. It in particular highlights the interaction that occurs between clients and developers.

2. Use Cases

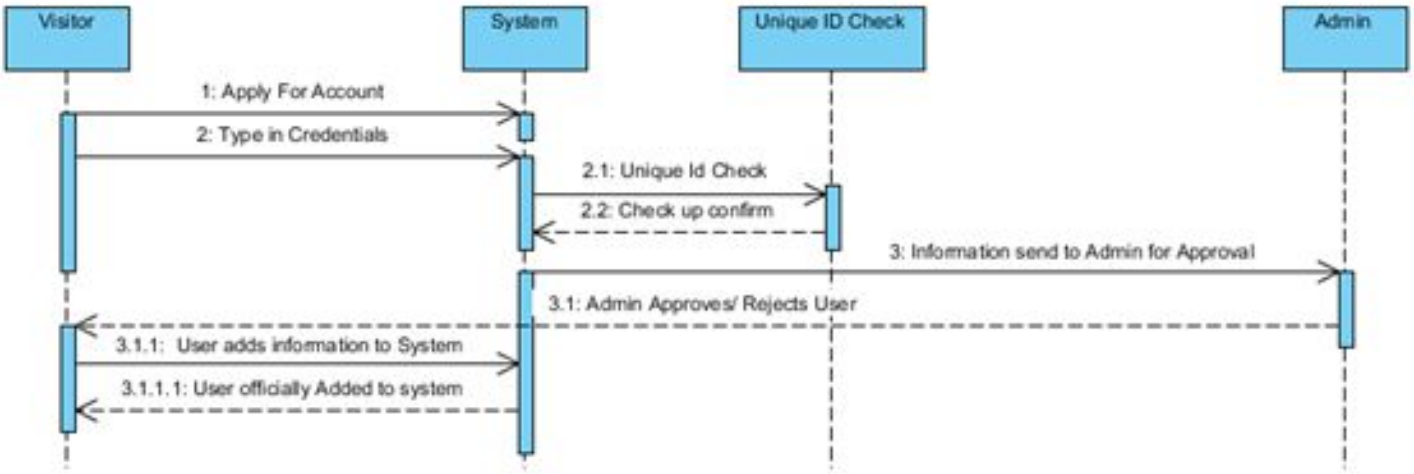
User Case: Public Access



Petri Net: Public Access



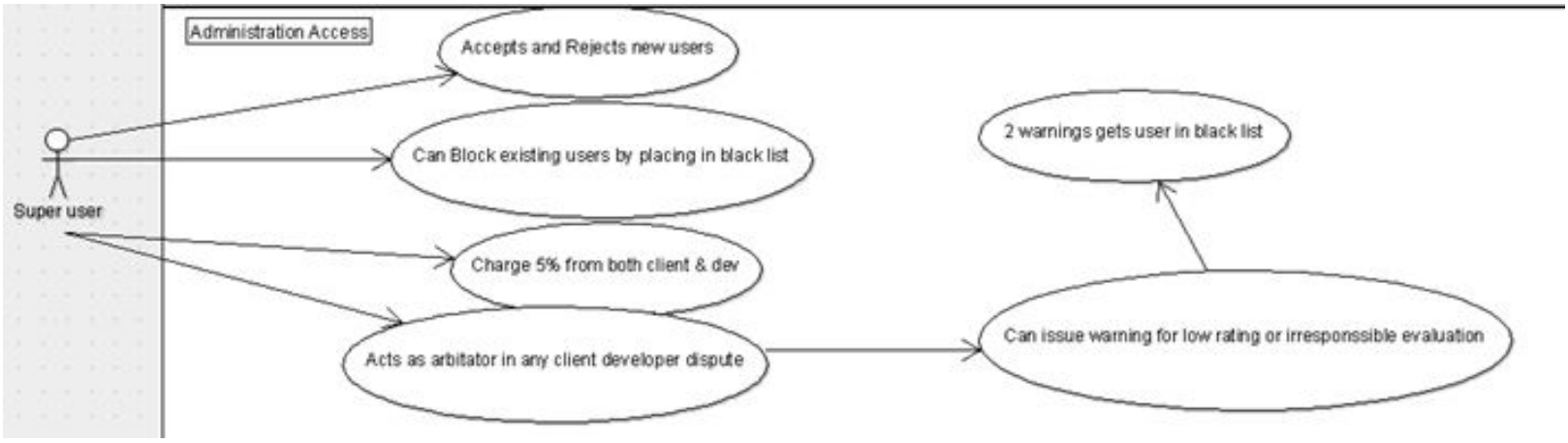
Sequence Diagram: Public Access



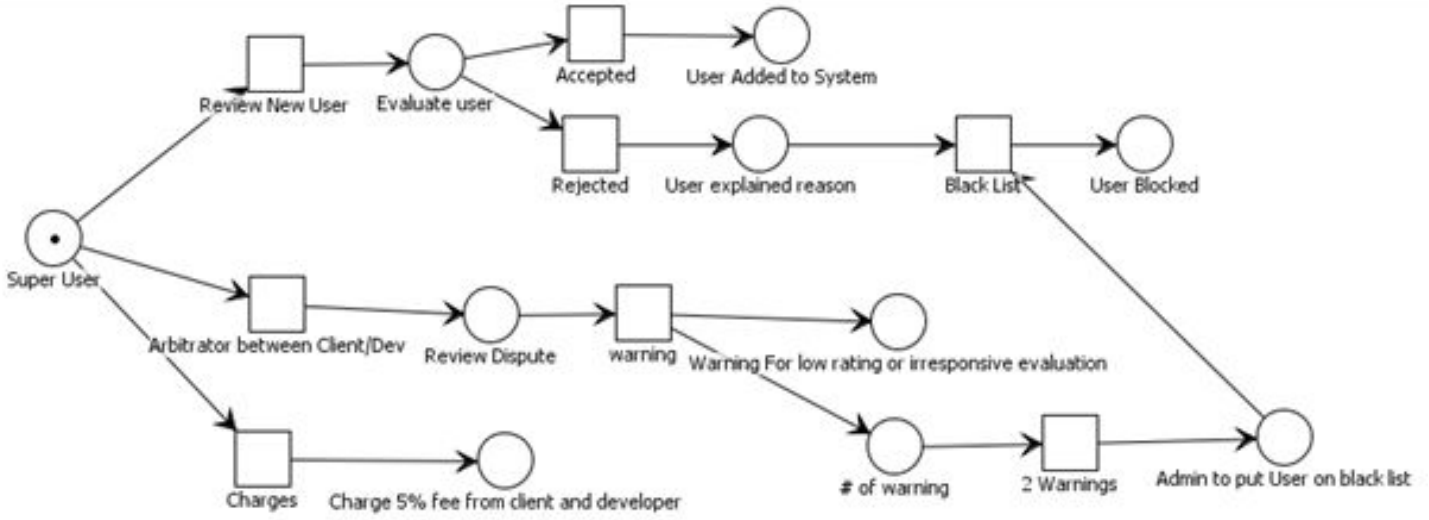
In this case, the consider what first time users and anyone without an account

will have Access to when first entering the system. Visitors have access to browse and search clients and developers. Visitors will have access to any public information made available by clients and developer. They will also have access to information such as If any visitor wants to hire a developer or work for a client, they will have access to create an account. To create an account, visitors will be allowed to create a temporary account, the system will check for the uniqueness of the visitor's temporary id and after that the visitor will have access to the temporary account. Once the visitor account is complete the system will decide if the visitor qualifies to become a member. If the visitor if accepted the system will welcome the newly registered user and ask him/her to provide more information such as resume, picture, line of work and any interests that can lure future clients or developers. If the visitor is rejected, then the user will be notify of the reason why they were rejected the next time the next time they log in the temporary account. The message explaining the reason will be shown on a screen which will not allow the user to access the account anymore.

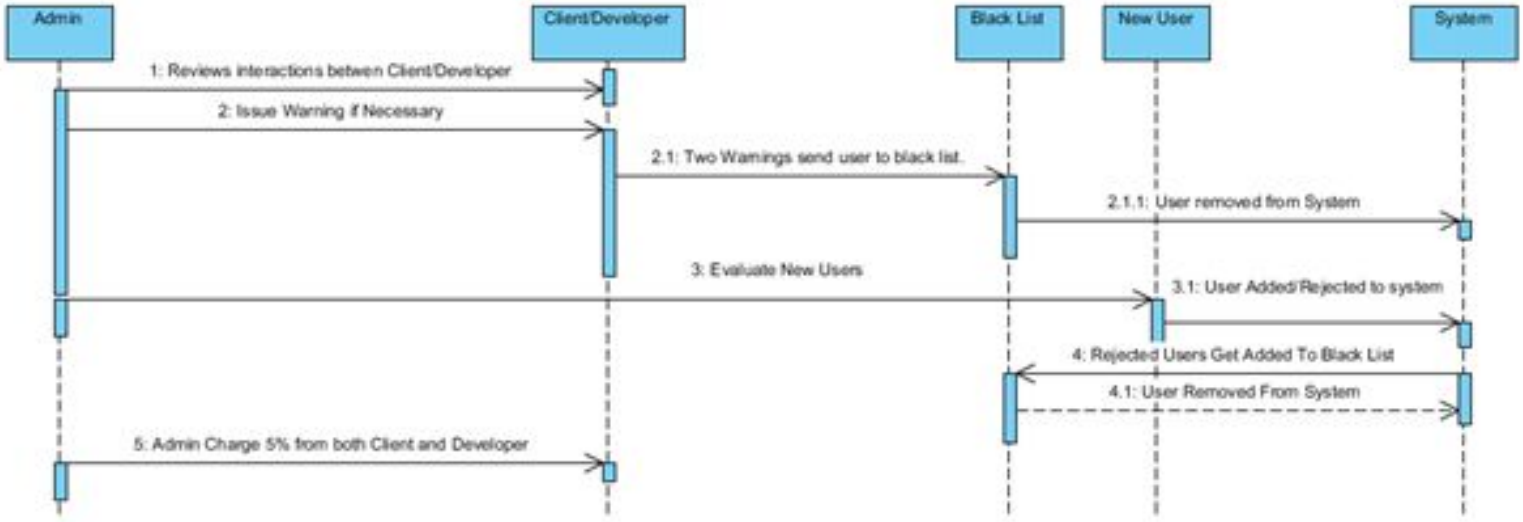
User Case: Administration Access



Petri Net: Administration Access

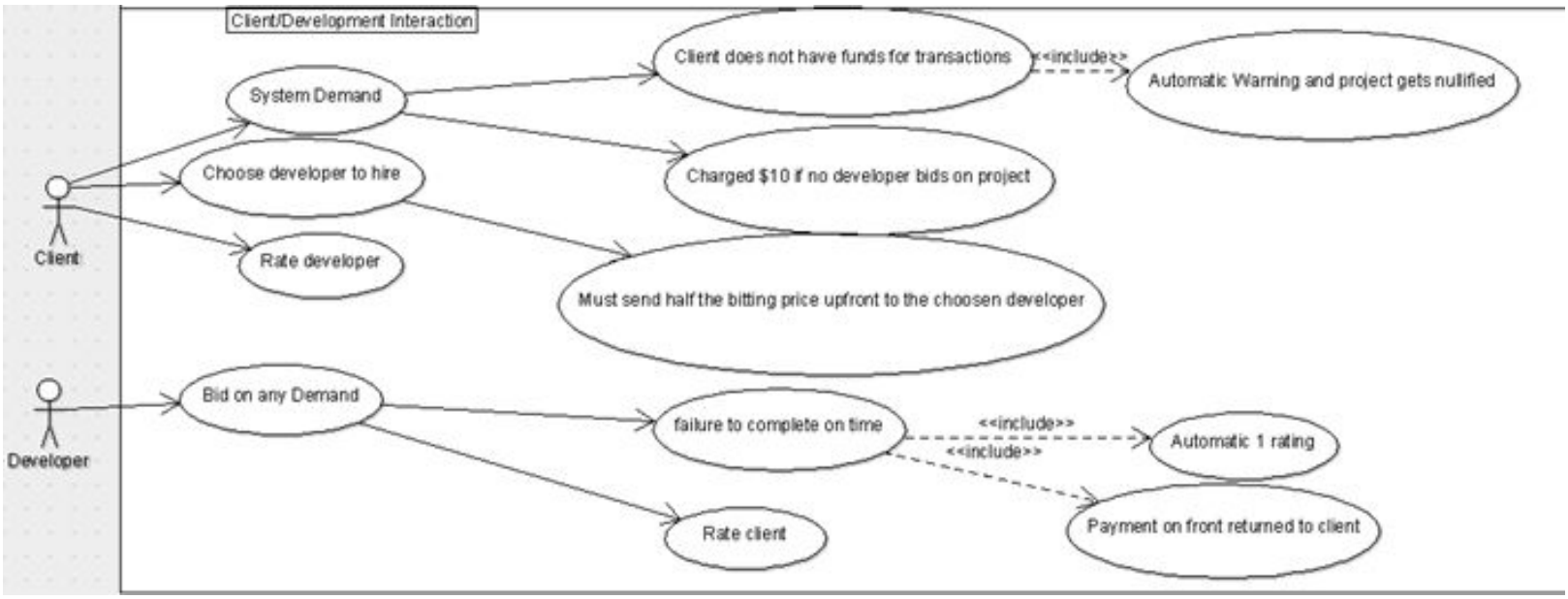


Sequence Diagram: Administration Access

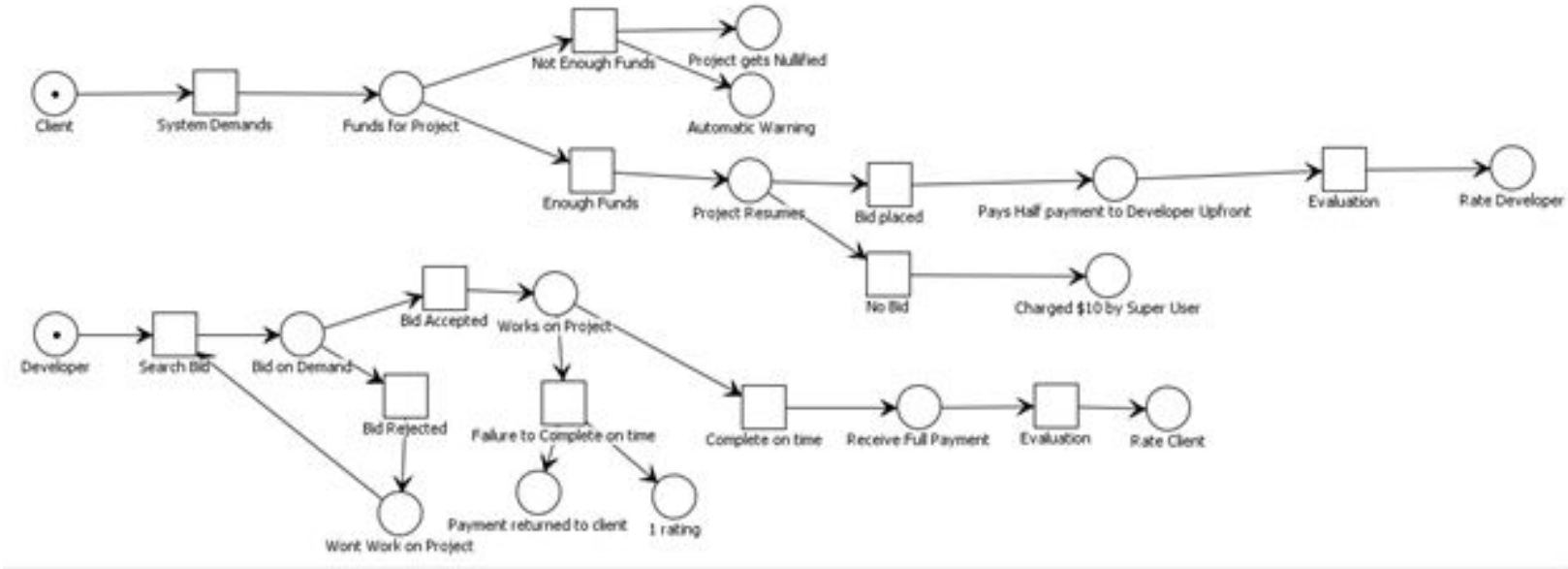


In this case we consider what the super-user will be able to do to moderate the system. The Super-User is able to accept or reject newly registered users. For those who are accepted, the system will welcome with a message and will ask the newly accepted user for more information. Those rejected by the Super-user will lose access to their temporary account and the next time they log in a message provided by the super-user will explain the reason behind the rejection. The super-user can also block current users by placing them in a blacklist. The users that are sent to the blacklist are not allowed to return to the system for one year. Anyone that attempts to register that is on the black list will be automatically rejected by the system. The Super-User also can serve as a moderator between clients and developers. Whenever there is disagreement between clients and development occur the super user will mediate the situation. An example of this will happen if say the client gives a bad rating to the developer and does not want to pay full price for the work done. In this scenario, the super-user will evaluate the case and determine if the rating was fair or not. If the rating is accurate, the Super-user can reduce the payment of the developer and issue a warning to the developer for low rating. If the client rated unfairly a warning will be issued to client for irresponsible evaluation. If either side gets two warnings they will automatically be placed in blacklist.

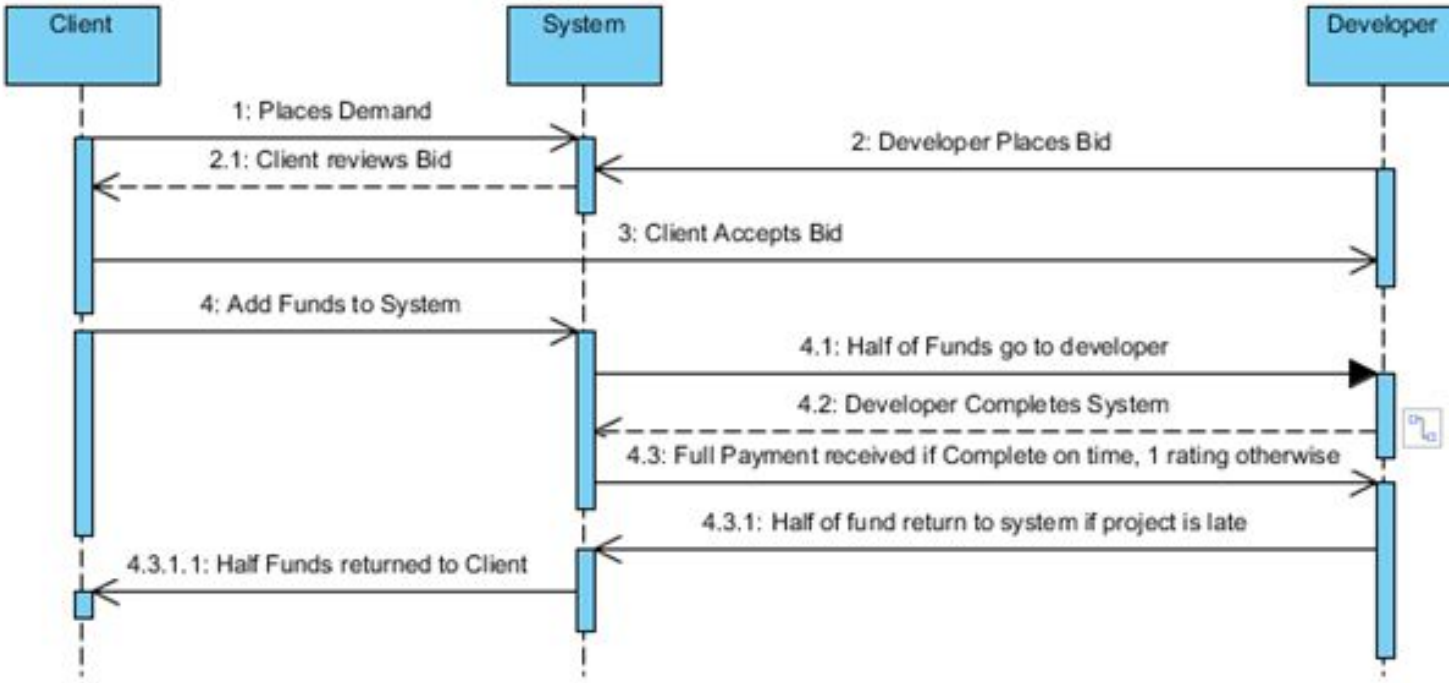
User Case: Client-Developer interaction



Petri Net: Client/Developer interaction

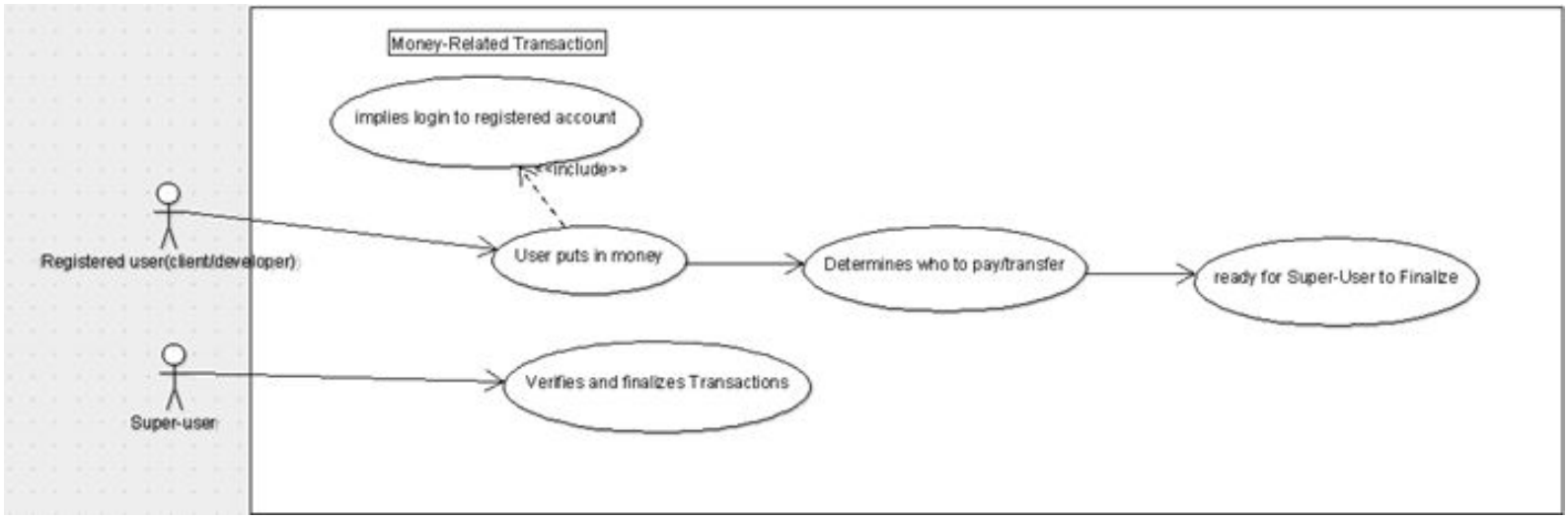


Sequence Diagram: Client/Developer Interaction

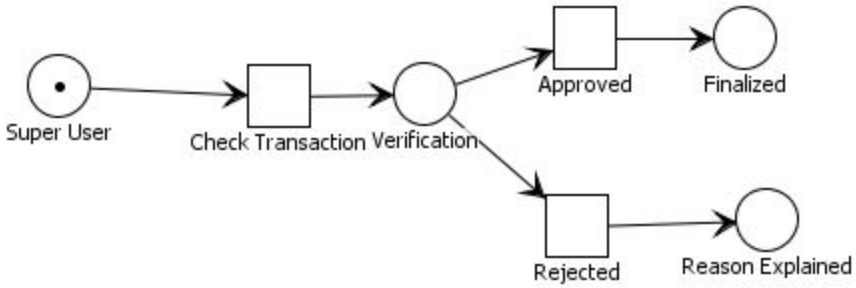


In this case we consider how the system interaction will work between clients and developers. The client will post a system demand and developers will have the opportunity to bid on that demand. If not bid is placed, the client will be charged $10. The client is not obligated to take the highest bid, however, if they do not choose the highest bid they must explain their reason for not choosing the highest bid. Once the developer is chosen, The client fill pay half the bidding price upfront. Developers are obligated to finish the system on the assigned deadline. If they do not meet the deadline, the must return their front payment back to the client. They also automatically receive a rating of 1 which will be followed by a warning. Both the client and the developer can evaluate each other through a rating system. If the client does not have enough funds, the project will be cancelled automatically and the client will receive a warning.

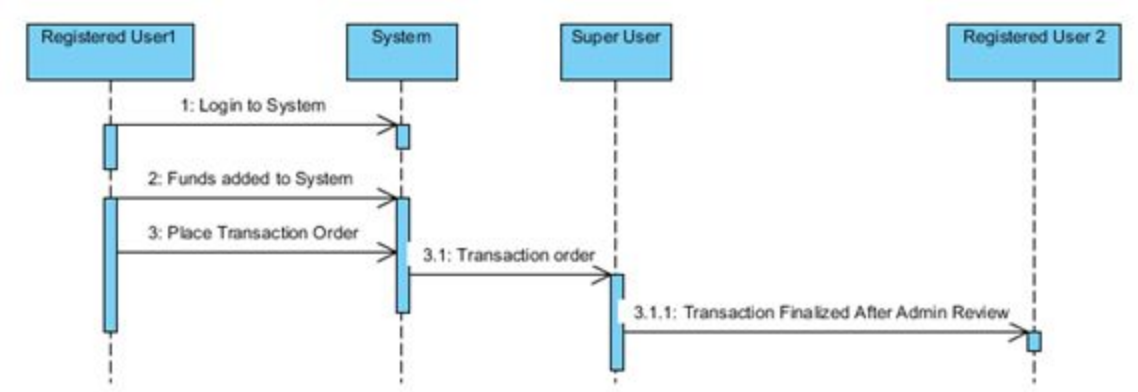
User Case: Money Transaction



Petri Net: Money Transaction

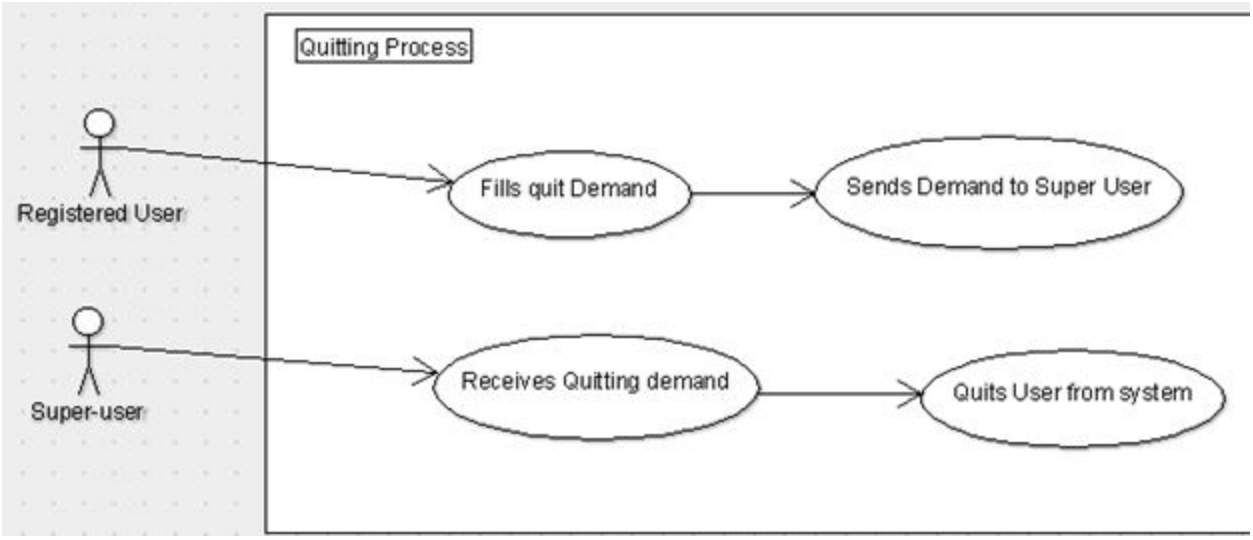


Sequence Diagram: Money Transaction

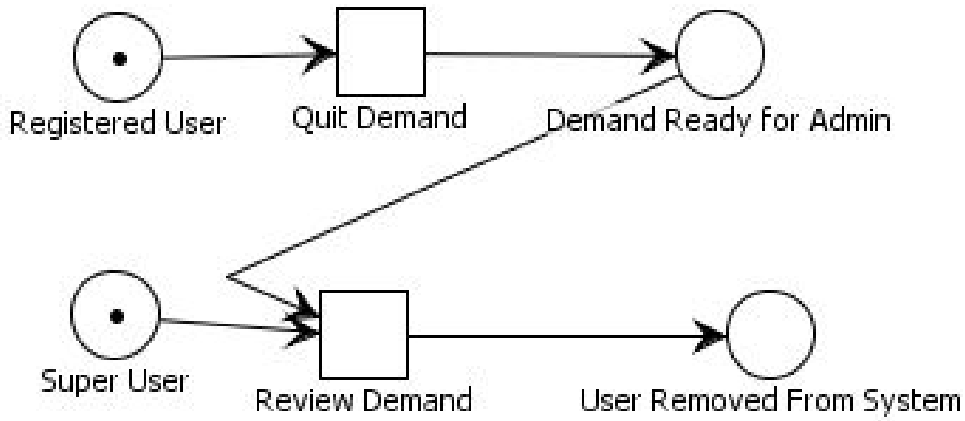


The user case diagram above shows how transactions occur in this system. The User enters the desired amount into the system. After the money and the transaction are read by the system, the system passes the information to the Super-user who will verify and finalize the transaction.

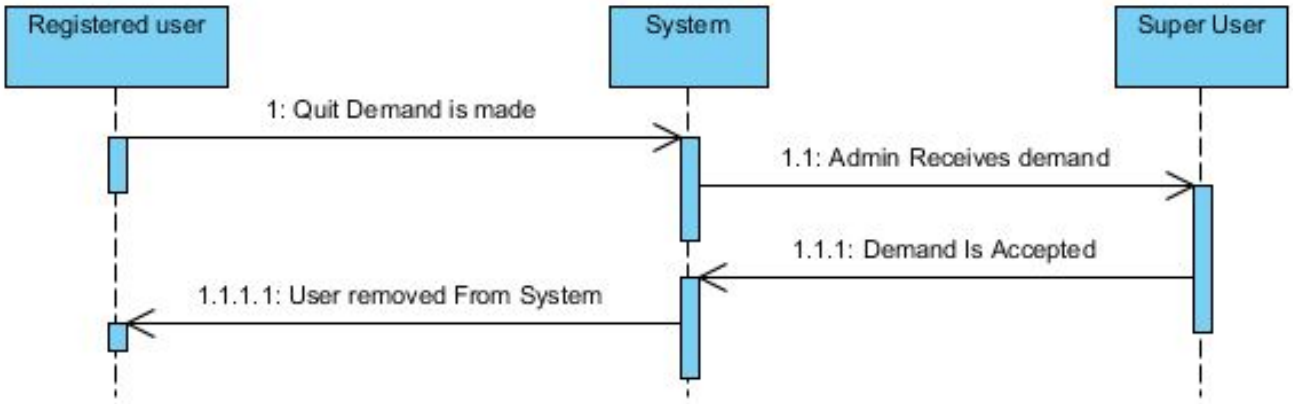
User Case: Quitting Process



Petri Net: Quitting Process

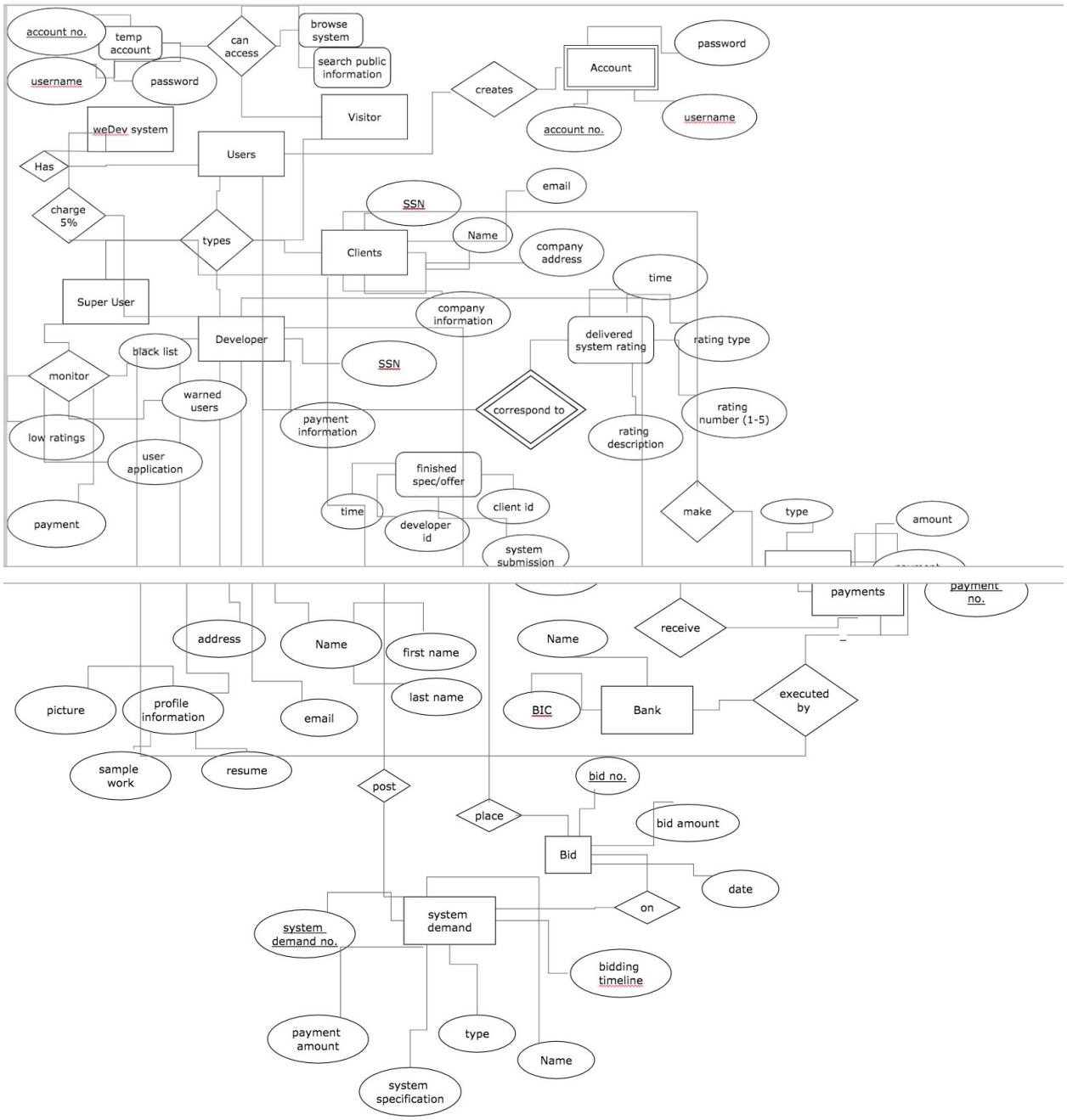


Sequence Diagram: Quitting Process



To quit the system, The User must fill a quit demand that will be send to the super-user. The super-user will receive the demand and take the user out of the system.

3. E-R Diagram


account no.
temp account
username
password
can access
browse system
search public information
Visitor
weDev system
Has
Users
creates
Account
password
account no.
username
charge 5%
types
SSN
email
Name
company address
Clients
time
Super User
company information
Developer
black list
monitor
SSN
delivered system rating
rating type
warned users
payment information
correspond to
rating description
rating number (1-5)
low ratings
user application
finished spec/offer
time
client id
developer id
system submission
payment
make
type
amount
payments
payment no.
receive
address
Name
first name
last name
Name
picture
profile information
email
BIC
Bank
executed by
sample work
resume
post
bid no.
place
bid amount
Bid
date
on
system demand
system demand no.
bidding timeline
payment amount
type
Name
system specification

4. Detailed Design

Create A User:

1. Redirect Function:

```
If user is logged in
        Redirect to Profile page;
Else
        Redirect to SignUp Page;
```

2. SignUp:

```
//User enters SignUp information.
//UserName, AccountType, Email, Password.
If Information is missing
        Render(signUp,ERROR:FILL IN ALL INFO);
If Email is already used
        Render(signUp,ERROR:EMAIL TAKEN);
If UserName is already used
        Render(signUp,ERROR:USERNAME  TAKEN);
Else
        HTML post request;
        // uploads data to database .
        Redirect  to Approval-Status page;
```

Log in:

1. Redirect Function:

```
If user is logged in
        Redirect to Profile page;
Else
        Redirect to Login Page;
```

2. LogIn:

```
//User enters information.
//Email, Password.
If Information is missing
        Render(signIn,ERROR:FILL IN ALL INFO);
If Email is wrong
        Render(signIN, ERROR:EMAIL IS WRONG);
If Password is wrong
        Render(signIN, ERROR:PASSWORD IS WRONG);
Else
        Redirect to Approval-Status page;
```

Approval-Status:

1. Redirect Function:

```
If user is logged in
        If Approved
                Redirect to Profile page;
        Else
                Render(Approval-Status page, STATUS/REASON);
Else
        Redirect to Login Page;
```

Admin Approve/Block Users:

1. Redirect Function:

```
If user is not logged in
        Redirect to Login page;
If Admin
        Render(Users-page);
        //This page displays “pending” or reason for rejection.
Else
        Redirect to Profile page;
```

2. Users:

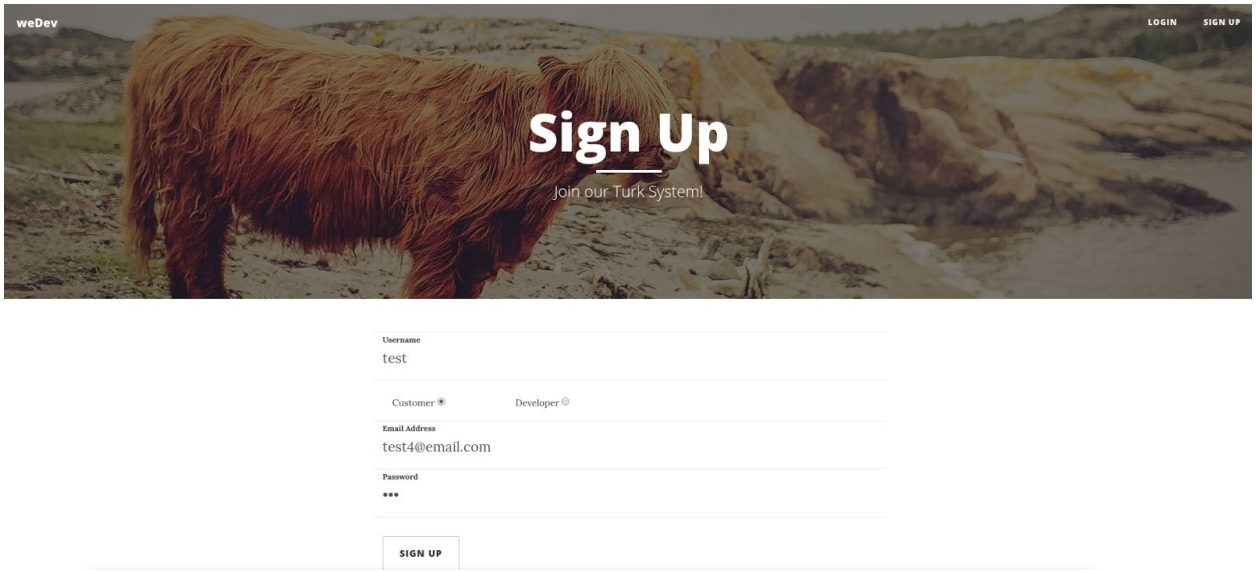
```
Users = GetAllUsers();
Render(usersPage,Users);
//When user is clicked on,
        editUser = GetUser(username);
        Render(SingleUserPage, editUser);
        //Admin can now edit the user information,
        //AccoutType, AccountStatus.
        //Admin can Approve or add a  reason for rejection.
        Html Put request;
        // updates the information in the database.
        Redirect  to(Users-page);
```

Client Posts Demand Create:

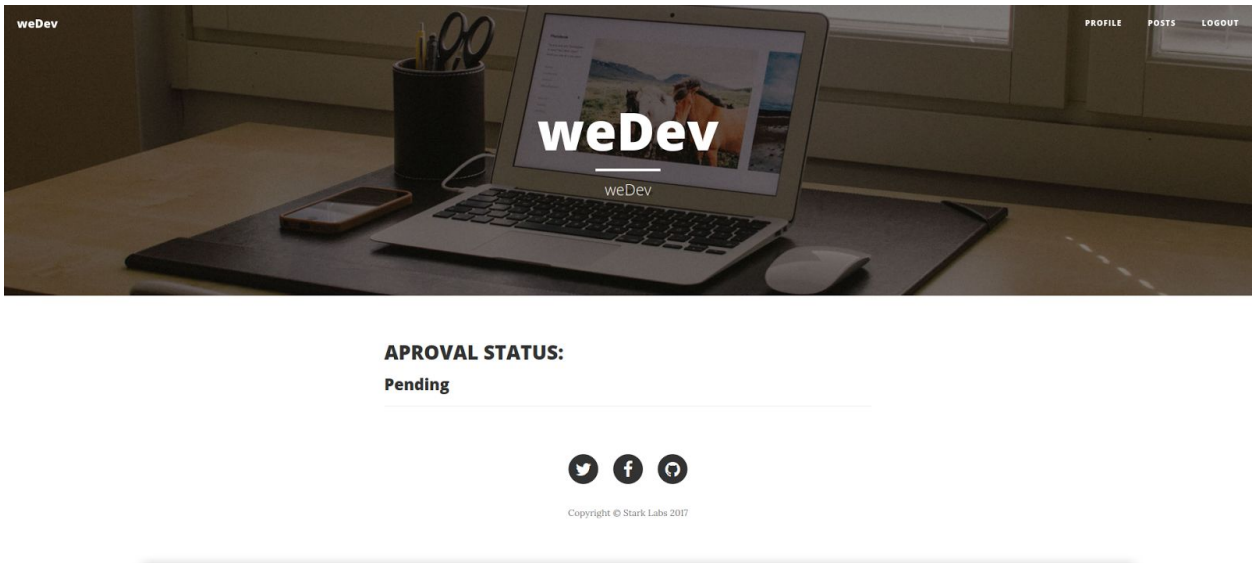
1. Redirect Function:
    If LoggedIn && Customer
        Continue to postCreate page
    Else
        Render(AllPosts);

2. CreatePost:
    //User enters information
    If Information is missing
        Render(CreatePost,ERROR:FILL IN ALL INFO);
    Else
        HTML Post request
        // sends data to the database

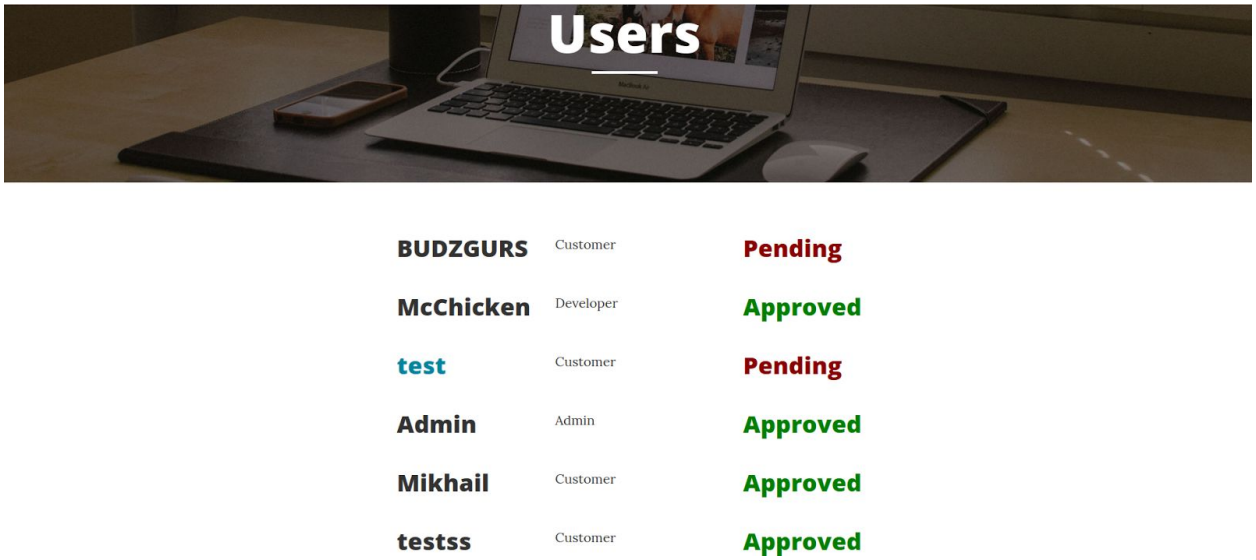
5. System Screens



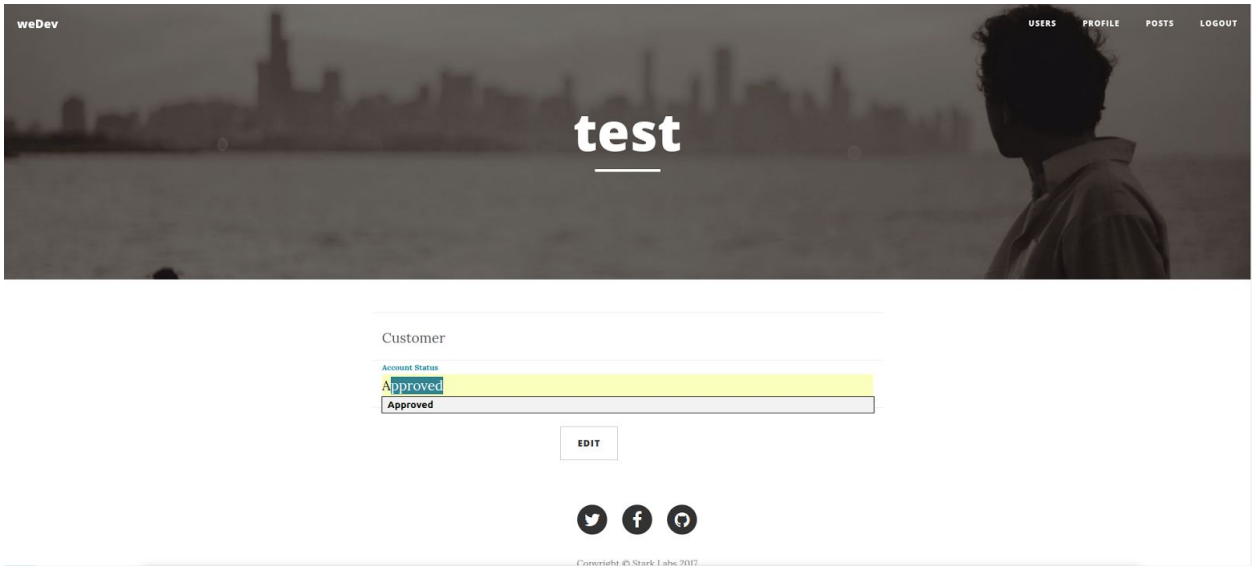
Sign Up page apply for a temporary account.



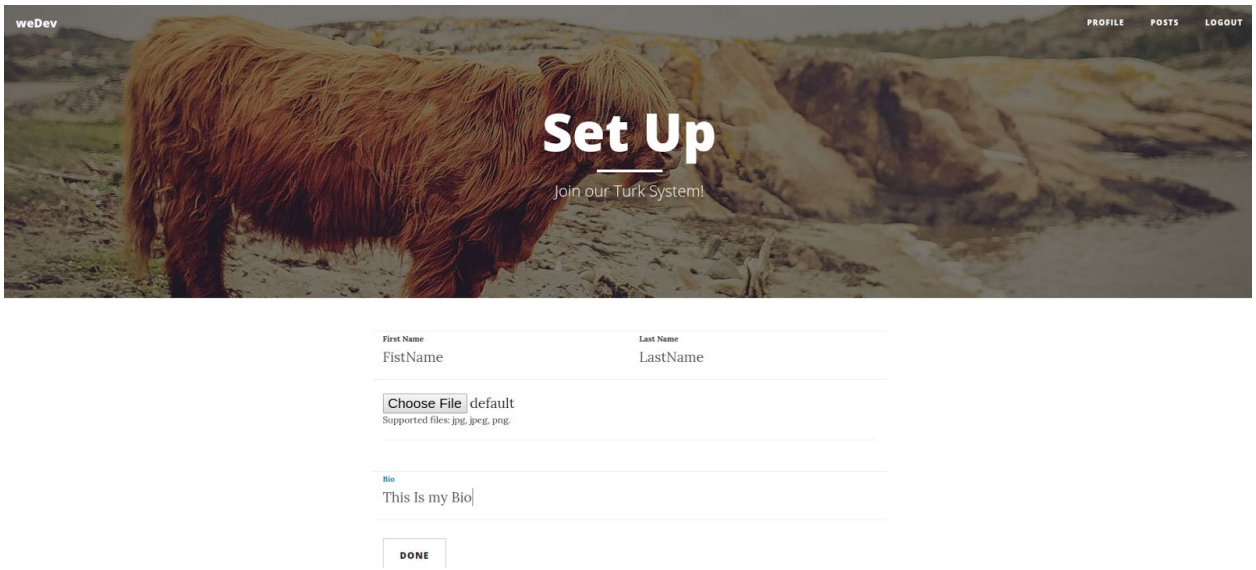
Wait for admin approval.



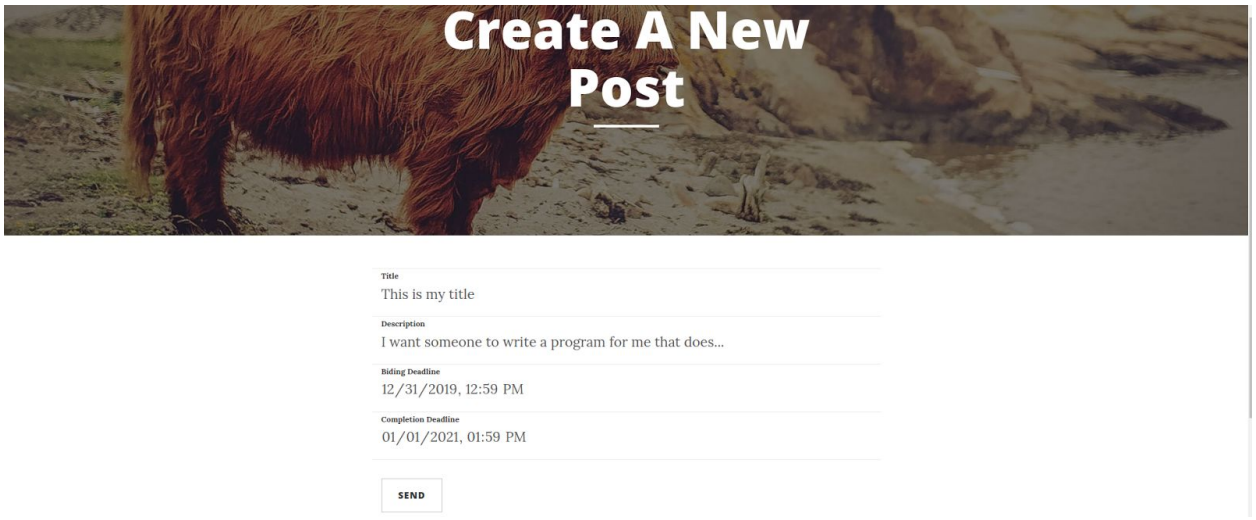
Admin can see all users.



Admin can change account status.



After being approved the customer can set up a profile.



Customer can can create a post.

6. Minutes of Group meetings

- 2 Hours each week during and after recitation class.
    - Discuss new functions and implementations that should be added.

- 50 Minutes every other week on Slack.
    - Discuss any error and updating the team on the overall progress.

7. 1st Phase report update

- We had one error in the User Case Diagrams. We have updated the diagram as shown in part 2.