Group 3  
CSE 404: Software Engineering

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

**Introduction**

“*done”* is a web based platform that gives freelancers the opportunity to be employed in the field of their choice; it allows employers to choose and communicate with the freelancer they want to hire to complete the task/job at hand. There are multiple web applications on the world wide web that offers the same functionality, however the functionalities that make *“done”* stand out from the rest are its payment method, its employment method specifically designed to help employers find the best fit person for their tasks and lastly the fields that are available for freelancers and employers to choose from.

**Functionalities**

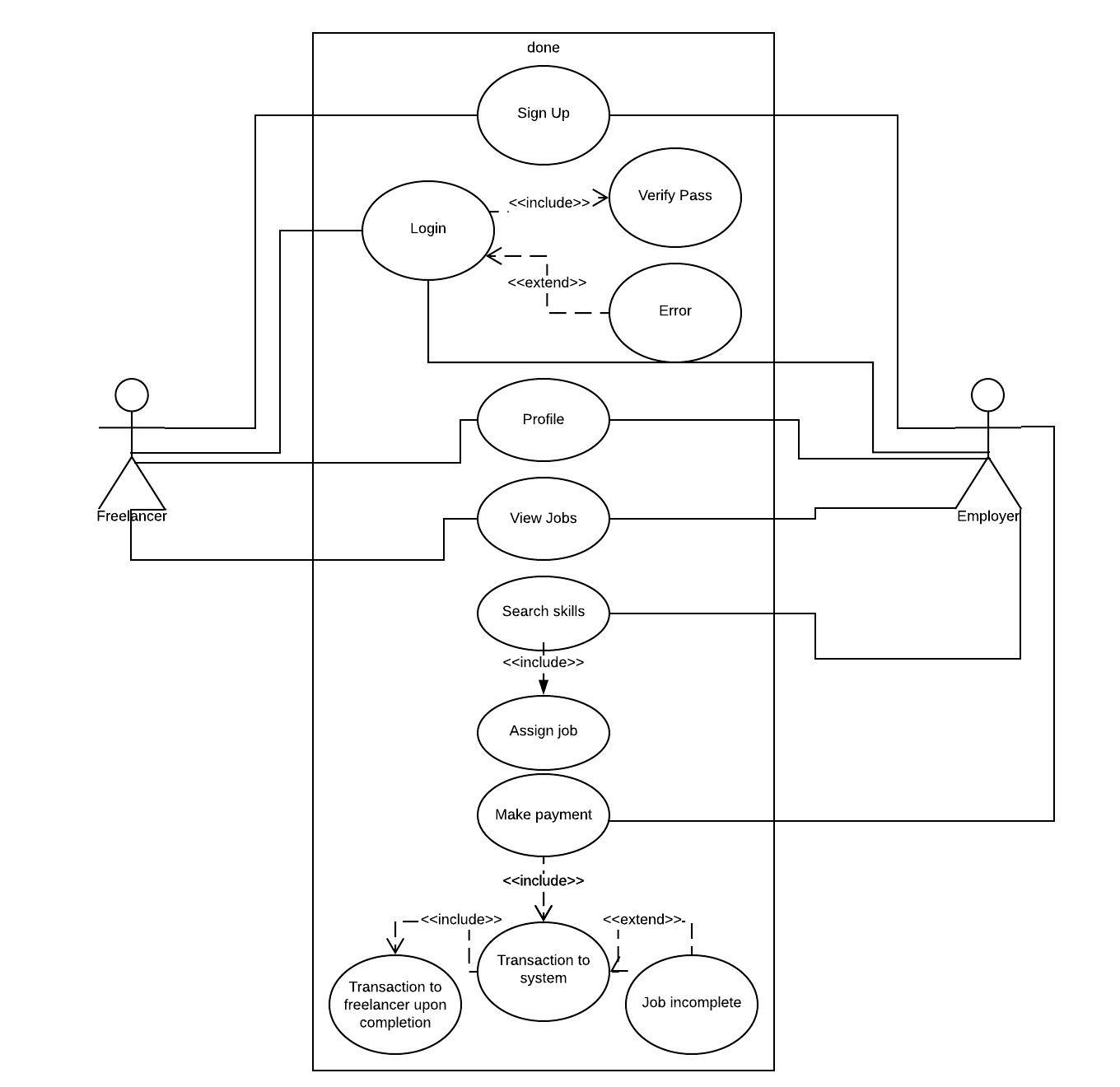
The application will allow freelancers from a variety of backgrounds/fields to join and exhibit their talents for employers to choose from. Instead of freelancers “*bidding”* for a job they will have the opportunity to apply for a job however it is the employer that will have the freedom to choose who gets to get the job done. Once the employer decides on who to employ for their task at hand they will be given the option to communicate via the web application or via other methods that they may prefer. During the process of communication the freelancer and employer will decide the terms of their agreement and proceed with the next step. After the terms are set the employer will complete his/her payment for the task; in this case *“done”* will act as a middle man who will hold on to the payment till the task is complete, once complete payment to the freelancer will be carried out by “*done”*. The employer as well as the freelancer will be made aware of the charges that will be deducted by the application as platform/company fees. Employers will be able to rate and comment about the freelancers they employed and based on these ratings others will have the knowledge about who to employ as well as get an assurance of the work quality that they will deliver.

A separate section will be included for tutors who are looking for tuitions. This particular section will be specifically developed for tutors and parents to interact with each other and find their perfect match. The payment system for tutors will be a bit different with the parents paying the site it’s fees before assigning a tutor and being able to communicate with them; all payments made afterwards will be made directly by the parents to the tutors.

All in all the *“done”* will be a platform via which freelancers will be able to exhibit their skills and portfolios to the thousands of employers looking for an easy solution to their tasks at hand.

**Use Case Diagram**

The simplest description of what a use case diagram is that it provides a representation of the user’s interaction with the system. It shows the relationship between the user and the different cases that the user is involved in.

****

***Fig 1.1 Use Case Diagram***

In the case of our system the users have been divided into two categories; the Freelancer and the Employer. Initially when a user views the website they view it as a guest and has minimal to no interaction with the system, after viewing its features he/she is prompted to sign up as a freelancer or as an employer. The actors in the system are **Freelancer** and **Employer.**

**Sign Up**

Users can either register themselves as a freelancer or an employer on the done platform. According to their choice, they will be presented with a form to complete that requires mandatory fields of information to get them registered on the system. On submission of the form, the input gets stored in the database, but before doing so the password of the user is hashed for security purposes. Even the administrators of the system cannot view the password that is entered by the user.

**Login**

Both the actors of the system are directed towards this use case when they want to interact with done. A form asking for their credentials is presented when they want to login. The username and password are matched with the ones stored in the database. If the user enters any incorrect information, the system sends back an error message.

**Search Skills**

This use case is only exposed to the employer actor. An employer can enter the skills they are searching for to complete their task. Freelancers with matched skills are returned in a list from where they can view the freelancer’s details.

**Assign Job**

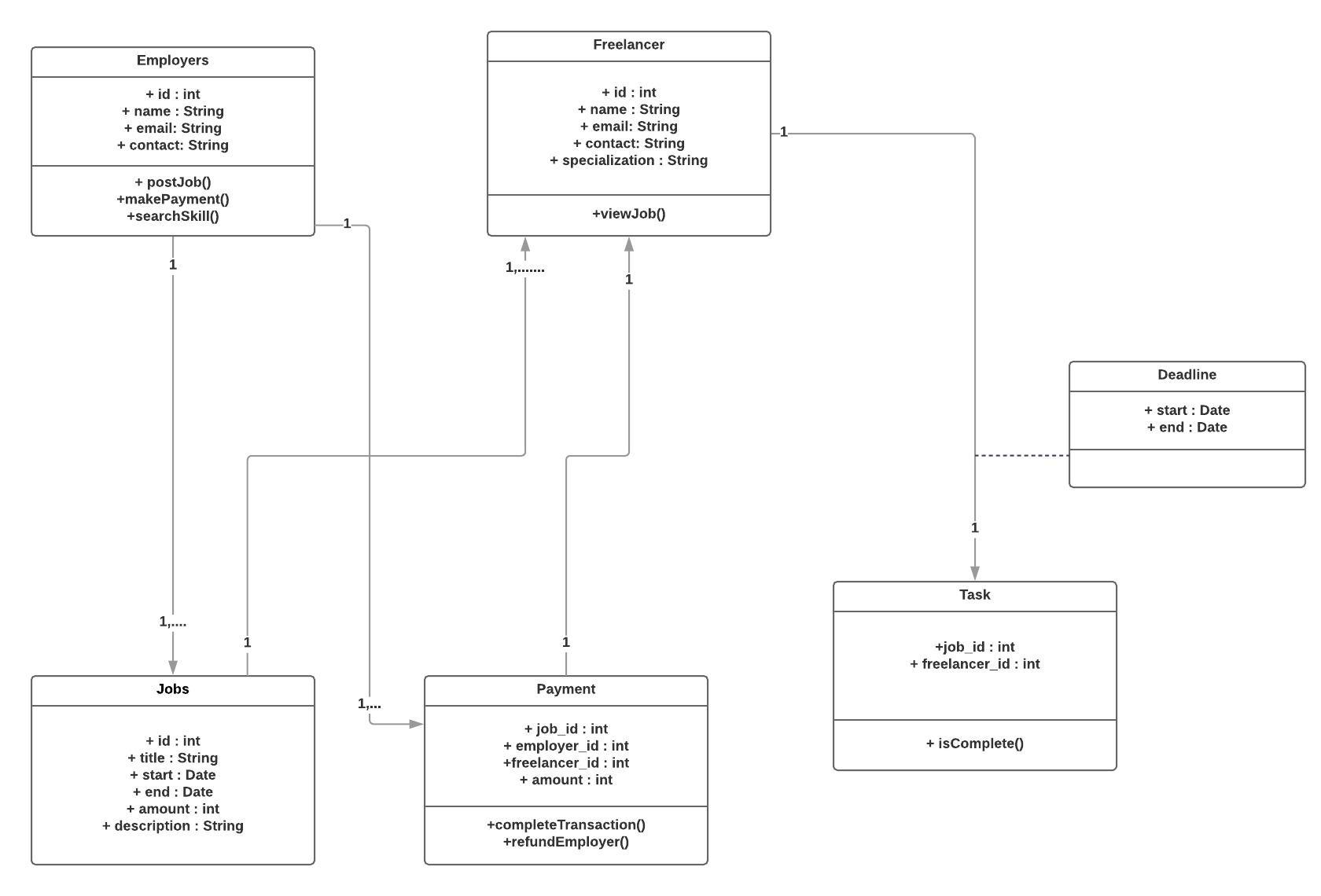
An extension of the previous use case is the Assign Job use case. Upon selecting a freelancer from the list, an employer can assign the task to the freelancer which would establish a connection between the two actors.

**Make Payment**

The employer actor makes the payment once the job is assigned and started. The payment is first made to the system. Upon completion of the job, within the deadline, the payment is forwarded towards the freelancer’s account. If the freelancer fails to complete the task, within the deadline, the employer will be refunded with the payment they made earlier.

**Class Diagram**

A class diagram is a static structure diagram in the unified modeling language (UML) that describes the structure of a system by showing the systems classes, their attributes, operations and the relationship between the objects.



***Fig 2.1 Class Diagram***

**Employer**Employers in the system are the ones who serve as the job -givers to the freelancers. Basic attributes they have are a unique id, their name, email and contact. An employer can post more than one job, which have a set of its own attributes, which will be discussed later. Employers are also able to makePayment and searchSkill. The latter provides the employers with a list of freelancers who match the skillset the employer is searching for.

**Freelancer**

The freelancers are the job-seekers in the system. Their basic attributes also consist of a unique id, and other fields such as their name, email, contact, specialization. We have also kept a field to know the freelancer’s location so that we can add a locale extended search engine in the future. viewJob is one of the tasks that they can perform which includes them applying for jobs that haven’t been assigned to any freelancer yet.

**Jobs**

Jobs are posted/assigned by the Employer. One employer may post multiple jobs. Each job has a unique id, a title, starting and ending dates, the remuneration and a brief description of the job. The job, if posted, can be viewed by all the freelancers on the system and will be subject to an apply option which the freelancer can use.

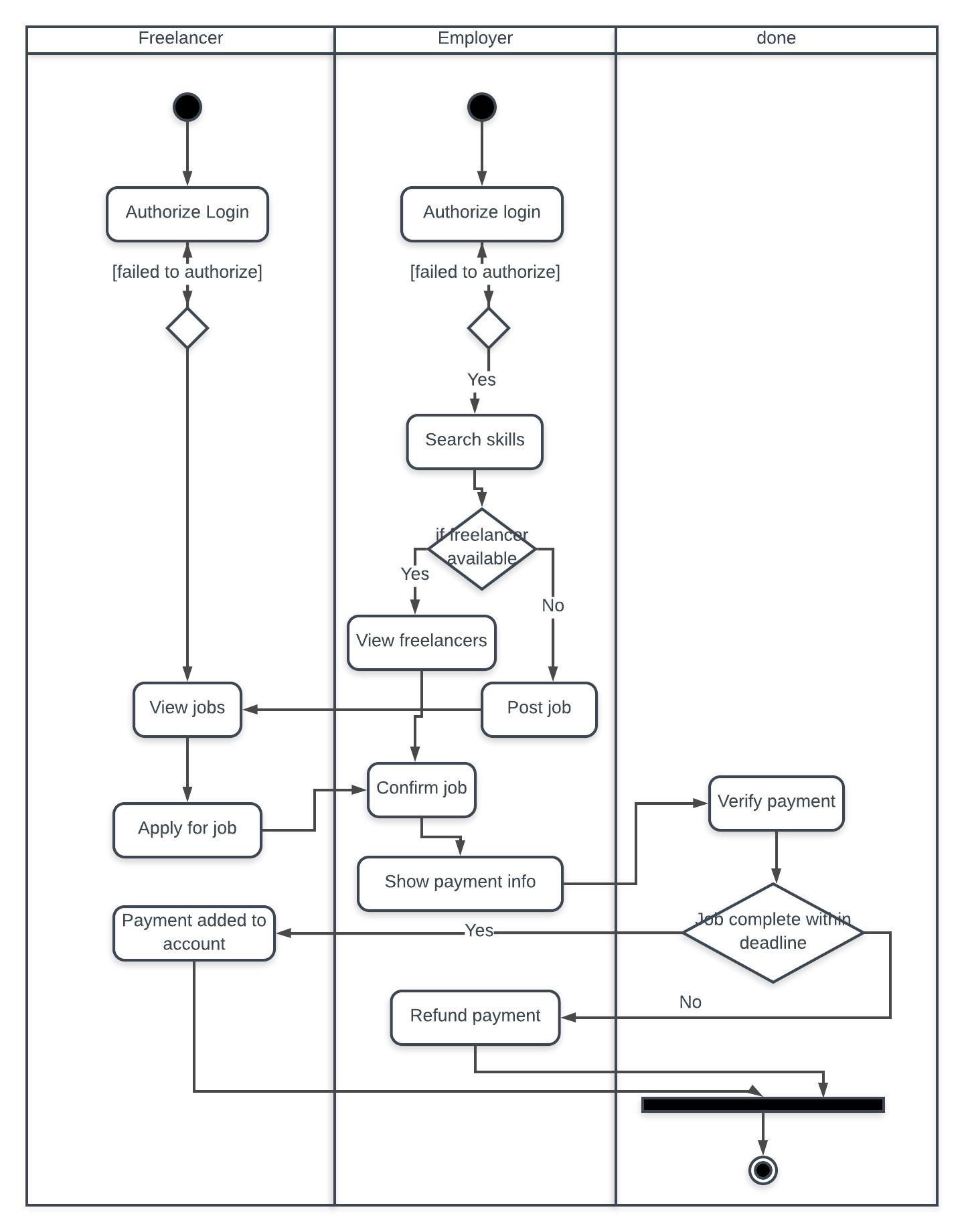
**Payment**  
An Employer will make a payment to the system when a job is assigned to a freelancer. The payment will consist of the job id, the employer id, the freelancer id and the amount that is to be paid. The payment can go either to the freelancer’s account through the completeTransaction action or be refunded to the Employers account. This depends on the task being completed within the time frame that is set for the job. An employer can make multiple payments, each for a job that is posted by them. A payment can only be made to one freelancer for a job.

**Task**

To check if an assigned job is complete or not the Task class returns the result. It consists of the job id and the freelancer id for checking the details. The job id is used to reference the start and end dates of the job.

**Activity Diagram**

An activity diagram is a representation of a set of procedural activities of the system. They describe parallel and conditional activities, use cases in a much more detailed way.



**Figure 3.1 Activity Diagram**

The activity diagram consists of three swimlanes. Freelancer, Employer and the system itself, done. The activity diagram has two start states; one for the freelancer and one for the employer.

Employers are subject to a login form which authorizes their session and allows them more options to interact with the system. Once logged in, an employer can search for skills that they require to complete their task. A list of freelancers with the matched skill set are returned. If the search returns nothing, then the employer can post the job which will be enlisted in the sites view jobs page. Employers can select a freelancer and assign them with the job after which they will be making the transaction to the system.

Freelancer are subject to a login form as well which authorizes their session and lets them continue if they have entered the correct credentials. Freelancers may view jobs that are posted by the employers and apply for any, which fit their requisites.   
The system plays an integral part in the payment system. Once the transaction is made by the employer, the system keeps track of the assigned job. If the job is not complete within the time frame that has been decided, the payment is refunded to the employer. If the freelancer successfully completes the task, the system forwards the payment to the freelancers account.

The end state is in the done swimlane in which the session terminates when the employer or the freelancer logs out.