**Requirements Definition**

**Got Work?**

**1. Introduction and Context**

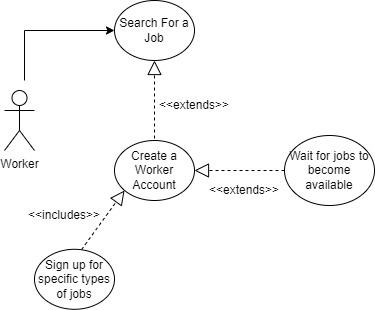
In years past, connecting people with tasks like lawn mowing and snow shoveling with willing workers has been primarily done via word of mouth. The goal of this system is to increase connectivity between workers and people with jobs.

This system will allow customers to sign up and record when and what jobs need done. It will then connect these customers to willing and available workers who will then come and perform the task. Upon the completion of a job, the worker will be paid. This system will allow customers to review the work performed, and request that they do not get a specific worker again. It will also allow the customer to request a refund from the system administrator.

Workers will be able to sign up and register times that they are willing to work along with how far they are willing to travel. When a customer posts a new job within their available time and area, they will receive a notification and will be able to opt in to doing the job. Upon opting in, a worker will then be shown how to get to the job. Upon completing a job, the worker will be able to leave a review on the customer for other workers to view.

This system aims to help more workers and customers connect. By providing a digital medium, people who would never hear of each other by word of mouth may now have the chance to connect.

**2. Users and their Goals**

*Figure 1 - Worker Looking for a Job*

Participating Actor: Worker

Entry Condition:

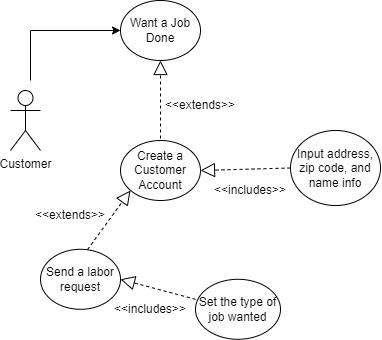
* Signed in as Worker

Exit Condition:

* Job’s are available and accepted
* No job is available

Event Flow:

1. Worker wants to work a job
2. Worker creates an account (further explained in figure 4)
3. Worker sets the types of jobs they want to work
4. Currently available jobs are searched for

*Figure 2 - Customer Looking For a Worker*

Participating Actor: Customer

Entry Condition:

* Signed in as Customer

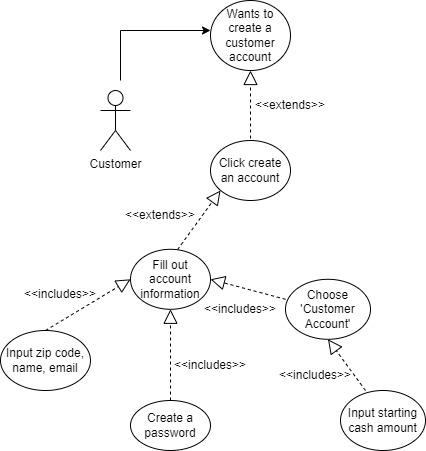
Exit Condition:

* Job is created

Event Flow:

1. Customer needs a job done
2. Customer creates an account (further explained in figure 3)
3. Customer sends a labor request
4. Workers with correct conditions are sent job

*Figure 3 - Customer Creating a New Account*

****

Participating Actor: Customer

Entry Condition:

* None

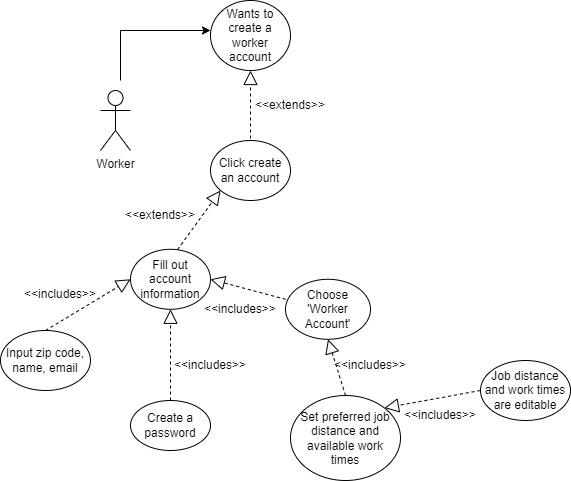
Exit Condition:

* Account is created
* Account creation is canceled

Event Flow:

1. Customer clicks ‘Sign Up’
2. Customer fills out account information
   1. Zip code, name, email, etc.
3. Customer creates password
4. Customer creates a ‘Customer Account’
   1. Deposit money (optional)

*Figure 4 - Worker Creating a New Account*

****

Participating Actor: Worker

Entry Condition:

* None

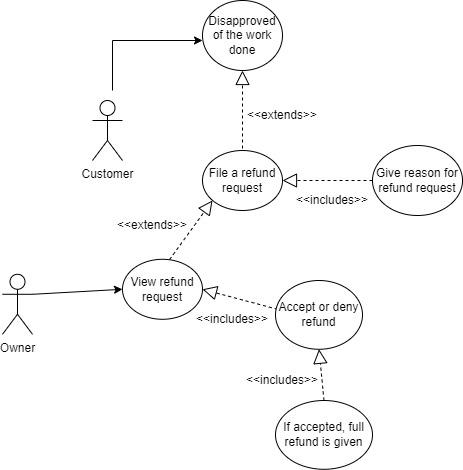
Exit Condition:

* Account is created
* Account creation is canceled

Event Flow:

1. Worker clicks ‘Sign Up’
2. Worker fills out account information
   1. Zip code, name, email, etc.
3. Worker creates password
4. Worker creates a ‘Worker Account’
   1. Inputs max job distance and available work times

*Figure 5 - Owner Reimbursement*

****

Participating Actor: Worker, Owner

Entry Condition:

* Customer is logged in
* Job was recently done for customer

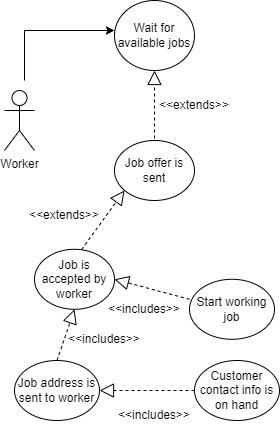
Exit Condition:

* Refund request is accepted
* Refund request is denied

Event Flow:

1. Customer did not like the work done
2. Customer files a refund request
3. Owner approves or denies refund request

*Figure 6 - Worker Getting Job*

****

Participating Actor: Worker

Entry Condition:

* Worker is logged in
* Worker’s available work times are set to their desired times

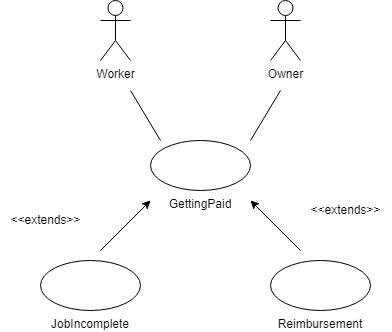
Exit Condition:

* Job is found and accepted
* Job is not found or declined

Event Flow:

1. Job offers are sent to worker
2. Worker accepts job offer
3. Address is sent to worker
4. Worker starts job
   1. Customer contact info is saved if needed

*Figure 7 - Worker and Owner get paid*



Participating Actor: Worker, Owner

Entry Condition:

* Worker has completed a job for Customer

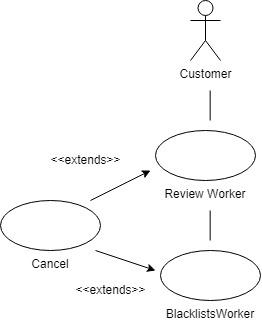
Exit Condition:

* Payment is processed successfully

Event flow:

1. Worker completes a job
2. Cost of job is taken from Customer’s balance
3. Worker receives 90% of job cost, Owner receives 10%

*Figure 8 - Customer Reviews and Blacklists Worker*



Customer Review

Participating Actor: Customer

Entry Condition:

* Worker has worked at least one job for Customer

Exit Condition:

* Worker submits review (positive or negative) or cancels

Event flow:

1. Worker works a job for Customer
2. Worker reviews or cancels
3. Customer receives feedback

Blacklists Worker

Participating actor: Customer

Entry Condition:

* Worker has worked at least one job for Customer
* Customer expresses dislike of worker through review process

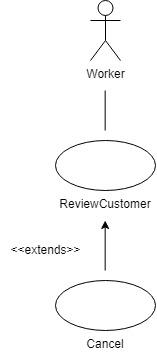
Exit Condition:

* Worker submits blacklist request OR cancels

Event flow:

1. Worker reviews customer
2. Worker blacklists customer
3. Worker cancels or submits form to blacklist

*Figure 9 - Worker Reviews Customer*



Participating Actor: Worker

Entry Condition:

* Worker completes a job for Customer

Exit Condition:

* Worker submits review OR cancels

Event Flow:

1. Worker completes a job for Customer
2. Worker rates customer positively or negatively
3. Worker submits the form OR cancels

**3. Functional Requirements**

1. User Authentication and Access

1.1. The system must require that all users sign into a login page before giving

access to the system.

1.1.1. On the login page, the system must provide a link to a page for the

new user to register

1.1.1.1. The system must get the user’s email.

1.1.1.2. The system must let the user register as either a Worker or a

Customer.

1.1.1.3. The system must get the user’s password.

1.1.2. The system must allow a registered user to login with their email

address and password.

1.1.2.1. On a failed attempt, the system must allow the user to try

again.

1.2. Users can have access to either Worker, Customer, or Owner features.

1.2.1. For Worker features see section Functional Requirements #3.

1.2.2. For Customer features see section Functional Requirements #4.

1.2.3. For Owner features see section Functional Requirements #5.

1.3. All user profiles have access to User Profile Features.

1.3.1. For User Profile Features see Functional Requirements #2.

1. User Profile Features

2.1. User profiles must have an email address.

2.1.1. This must not be able to change.

2.1.2. This must be able to be seen on the website.

2.2. User profiles must have a password.

2.2.1. This must be able to be changed on the website.

2.2.2. The password must not be visible anywhere on site.

2.3. User profiles must have a monetary balance.

2.3.1. Users must be able to increase the balance via a deposit.

2.3.1.1. The system must not require real currency to process a

deposit.

2.3.2. Users must be able to decrease the balance via a withdrawal

2.3.2.1. The system must not process a withdrawal using real

Currency.

2.4. User Profiles must have a zip code.

2.4.1. This zip code must be able to be changed.

2.4.2. This must be seen on the website.

2.6. User Profiles must have a name (first and last)

2.6.1. Names must not be able to be changed.

2.6.2. Names must be visible when connecting Customers and Workers.

2.5. Non-Owners must not be able to view another user’s profile information.

1. Worker Profile Features

3.1. Workers must be able to input what types of jobs they are willing to do.

3.1.1. This must be able to be changed at any time.

3.2. Workers must be able to input what times they are available for a job.

3.2.1. Workers must be able to choose to make an available time recurring.

3.2.2. Workers must be able to change their available times.

3.2.2.1. Workers must not lose any accepted jobs if they change an

availability time.

3.3. Workers must be able to set the area they are willing to work in.

3.3.1. Workers must be able to set and change their own zip code.

3.3.2. Workers must be able to set the maximum distance they are

willing to travel.

3.4. Workers must be able to accept a new job.

3.4.1. If a job matches the Worker’s willing job types, available time, and is

within the Worker’s area, the Worker must be provided the opportunity

to accept the job.

3.4.1.1. The Worker must be presented with jobs they may accept on

the website.

3.4.1.2. Upon accepting a job, another Worker must not be able to

accept the same job.

3.4.1.2.1. Jobs that have already been accepted by a Worker

must not appear as a notification.

3.5. Workers must be presented with jobs that they have already accepted.

3.5.1. Workers must not be able to drop jobs that they have already

accepted.

3.5.2. Workers must be shown jobs they have already accepted on the

website.

3.5.2.1. Contact information for the Customers of jobs must be

available to the Worker.

3.6. Workers must be paid after performing a job.

3.6.1. The Worker’s balance must increase by the ninety percent of the job’s

price when completed.

3.7. Workers must be able to rate a job after completion.

3.7.1. The option to like or dislike must be presented on the website the next

time the Worker logs in.

3.7.1.1. The form presented must provide a way to like.

3.7.1.2. The form presented must provide a way to dislike.

3.7.1.3. The form presented must provide a way to submit.

3.7.2. The Worker must be asked how long the job took.

3.7.2.1. For recurring jobs, this average should be displayed on job

requests.

1. Customer Profile Features

4.1. Customers must be able to request a job to be completed.

4.1.1. Customers must be able to set what time they want the job completed.

4.1.2. Customers must be able to set what kind of job they want completed.

4.1.2.1. Customers must be able to select a job type from a list of

allowed types.

4.1.3. Customers must be able to add where they want the job completed.

4.1.3.1. The Customer must be able to input the address.

4.1.3.2. The Customer must be able to put in a contact phone

number.

4.1.4. Customers must not be able to post a new job if they do not have a

high enough balance to pay for the job.

4.1.4.1. Customers must not be able to decrease their account

balance below what their current jobs require.

4.1.4.2. Recurring jobs must assess whether the Customer has the

required balance a week in advance.

4.1.5.2.1. If the Customer does not have a high enough

balance, the recurring job must be canceled.

4.2. Customers must be shown upcoming jobs.

4.2.1. Pending job requests should be able to be viewed.

4.2.1.1. Pending job requests must be able to be canceled.

4.2.2. Accepted job requests should be able to be viewed.

4.2.2.1. Accepted job requests must not be able to be canceled.

4.3. When a job is completed, the Customer’s account balance must be reduced

by the job’s cost.

4.4. When a job is completed, the Customer must be presented with the option to

like or dislike the Worker.

4.4.1. Upon disliking a Worker, the Worker could be added to the Customer’s

Blacklist by the Customer.

4.4.1.1. Blacklisted Workers must not have the option to accept this

Customer’s jobs.

4.4.2. Upon disliking a Worker, the Customer must be presented the option

to appeal for a refund.

4.4.2.1. Appeals for a refund are directed to the Owner.

4.4.2.1.1. Upon a successful appeal, the full cost is refunded

4.4.3. Customer must be presented with the option to not leave a review

1. Owner Profile Features

5.1. The Owner must be able to view other account’s information.

5.1.1. The Owner must not be able to see account passwords.

5.1.2. The Owner must be able to see all other account information.

5.2. The Owner must be able to add allowed job types.

5.3. The Owner must be able to see refund appeals.

5.3.1. The Owner must be able to accept or deny an appeal.

5.3.2. Upon a new appeal being created, the Owner should be notified via

email.

5.3.3. Resolved refund appeals must be visible to the Owner.

5.5. The Owner’s account balance must increase by ten percent of each job’s

price when a job is completed.

1. Job Features

6.1. Jobs created by the Customer must store data applicable to the Job.

6.1.1. Jobs must store the address of the Customer.

6.1.1.1. Jobs must store the zip code of the Customer.

6.1.2. Jobs must connect to the Customer.

6.1.3. When assigned, Jobs must connect to the Worker.

6.1.4. Jobs must store the cost of the Job.

6.1.5. Jobs must store the time and date of the Job.

6.1.6. Jobs must store the type of the Job.

**4. Non Functional Requirements**

1. The system must use a website.
2. The system must have different levels of Authorization.
3. The system must use a database.
4. The system must have a GUI.
5. The system must have cute little icons.

**5. Future Features**

This section contains features that will not be implemented during this project, but could be in the future.

1. Use of real money
2. The creation of a mobile application
3. Site being hosted on a server

**6. Glossary**

1. “Worker” refers to a user that will complete a job for a customer
2. “Customer” refers to a user that offers a job to a worker
3. “Owner” refers to the user that owns and runs the site.
4. “User” refers to any human using the site with a login.