Requirements Definition
Group 1

Job Finder

Introduction and Context
The purpose of this project is to build a system for facilitating community yardwork job offers. It will serve as a profitable means to match young workers with customers offering job listings. The site owner will not employ anyone, rather act as a job finding service that receives a surcharge for making a connection between the customer and worker.

To populate the system with listings, customers will have the ability to sign up and post the type of job they have available, and when they are hoping to have it completed. They will also add a balance to conveniently handle payment to the worker that completes their job listings. Once they have listed a job, the system will match it with an available worker.

The system will allow for workers to create accounts and set their availability. Jobs will be rewarded to workers on a fair basis, with a slight preference system in place for those with higher ratings from customer feedback. They may specify the type of jobs they're willing to accept and will be notified when one is available. Once a job is accepted by a worker, they will receive contact information for the customer and be expected to complete the job.

Upon completion, both parties will notify the software it is complete, and money will be transferred accordingly. 90% of a listed job wage will be rewarded to the worker, and 10% to the site owner. Both parties will have the opportunity to leave feedback on each other.

This system will provide a convenient means for young people making money to be connect with customers needing work done around their yard without having to create an individual advertising scheme.

Users and Goals

The purpose of this section is to describe the system's actors and the way in which they aim to interact with the system using UML use case diagrams.

<Insert UML use case diagrams>

Functional Requirements

This section focuses on requirements related to how the system is expected to behave.

1. User Authentication and Account Management
    1.1. The initial page will ask the user to sign in or sign up.
    1.2. Registration
        1.2.1. If the user clicks on the signup button, they will be taken to a page with a form.
            1.2.1.1. A user can register as a customer or as a worker.
            1.2.1.2. Users must sign up with a username, password, and contact information
            1.2.1.3. Users will have the option to add a profile picture or graphic.
            1.2.1.4. There will be a button to confirm registration, which will either successfully sign the user up or display errors with instructions on how to fix them.
            1.2.1.5. ** add terms and conditions
    1.3. Logging in
        1.3.1. Users with active accounts will sign in with a username and password each session.
            1.3.1.1. There will be fields for the username and password on the initial page, along with a button to sign them in.
        1.3.2. Incorrect attempts will result in an error being displayed back to the user.
            1.3.2.1. The error will let the user know that the email or password was invalid, or that another error was encountered, prompting them to try again.
    1.4. Account Management (All Users)
        1.4.1. A user must be logged in to access account management features.
        1.4.2. Any type of user will be allowed to change any account details aside from the account type, including email, password, contact information, and profile pictures.
        1.4.3. A user should be able to view account balance and blacklisted users.
            1.4.3.1. Customer accounts can money into their account (see Customer Account Features).
2. Customer Account Features
    2.1. The key functionality of this user is the ability to post job offers using a built-in template accessible from a home page.
        2.1.1. They will be able to specify job type (e.g., lawn mowing)
        2.1.2. They will be able to specify a time frame for when the job is to be completed (e.g., Thursday morning).
        2.1.3. They will have the ability to specify the estimated time for completion.
        2.1.4. They will have the ability to specify payment.
    2.2. They will have the ability to add money into their account.
        2.2.1. Note that for the scope of this project, we will not be using real money. Instead, they will enter an arbitrary number in for the system to work with.
        2.2.2. They will click a button to add money, then be prompted to enter a number in. Once saved, their balance will update to reflect the action.
    2.3. They will have the ability to blacklist workers.
        2.3.1. Blacklisted workers will not be able to interact with the customers on the system.

2.3.2. Any job posted by the customer that has been assigned to a blacklisted worker will be automatically cancelled.
2.3.3. They can also the action, though this won't reassign cancelled jobs.
2.4. They can leave feedback on worker after job is complete
2.4.1. They will be provided with a template containing specific categories to rate. ******ADD CATEGORIES HERE
2.5. They will have the ability to contact the owner about discrepancies at any point by clicking on a link to a complaint form.

3. Worker Account Features
3.1. The key functionality to this type of user is the ability to accept jobs posted by customers.
3.2. They will be able to set their availability to take jobs, meaning they will only be assigned jobs that fit within their set availability.
3.3. They will receive 90% compensation for each completed job.
3.3.1. Each job will display the customer's offer along with the actual amount they will receive.
3.4. They can blacklist customers.
3.4.1. Blacklisted customers will not be able to interact with the workers on the system.
3.4.2. Any job currently assigned by a blacklisted customer will be automatically cancelled.
3.4.3. They can also undo the action, though this won't reassign cancelled jobs.
3.5. They can leave feedback on customers.
3.5.1. They will be provided with a template containing specific categories to rate.
3.5.2. ***** ADD CATEGORIES HERE
3.6. They can mark when a job has been completed.
3.6.1. They won't be compensated until the customer marks it as finished.

4. Owner Account Features
4.1. A special account will be created for the system owner, and an owner account cannot be created through the typical registration process.
4.2. They will receive 10% of compensation for each completed job.
4.3. They will have a page only accessible to them for system management purposes.
4.3.1. They can change account access types (e.g., customer and worker).
4.3.2. They can add and manage types of jobs.
4.3.2.1. This would involve adding and removing job types from the drop-down list customers see when posting a job offer.
4.3.3. They have privileges to move money between accounts for reimbursement purposes.
4.4. In addition to these special features, they will also have the same access to processes that customer and worker accounts have.

** give customer the option to send complaint to owner

** only worker marks completion

** automatic assignment, priority element with rating

**worker has option to accept or decline job

**worker can change availability

**no blacklisting

Non-Functional Requirements

This section focuses on requirements related to how the system will work.

1. The system must utilize a database.
    1.1. The database will store user information.
        1.1.1. This includes permissions, username, password, email, contact information, ratings, and account balances.
    1.2. The database will be used to store information on job postings.
        1.2.1. This includes job title, description, type, address, estimated completion time, compensation, desired completion window, customer, and customer contact information.
            1.2.1.1. The customer contact information should only be accessible by the customer who created the offer, the worker assigned to it, and the owner.
2. The system design must be data-driven and dynamic.
    2.1. Updates to the database should affect other areas of the system.
3. The team will use Git as a version control system, with GitHub.
    3.1. The team will use GitHub Project to manage backlogs.
4. The system must be deployable either locally or by a cloud service.
5. The system must support mobile devices.

Future Features

This section focuses on possible additions for future versions that aren't necessary to the MVP.

1. The system could implement a filter by location feature to account for distance between workers and customers.
   1.1. Workers would have the option to set maximum distances.
2. The system could implement a preference system where customers could add 'favorite' workers to give priority to for their job offers.
3. The system could implement the ability for a customer to set up a recurring job offer.
4. A complaint system could be implemented for customers and workers to interact with the owner. **********Future or functional?? If future, edit functional to reflect change.
5. An alert system via email/SMS/Push could be implemented to let users know when jobs are available and accepted.
6. The database could track the history of jobs and a report could be displayed back to users.
7. The system could implement a tip system to adjust wages.

Glossary

This section defines a list of terms relevant to the project.

Owner - a user with elevated privileges that profits from connections and can manage job types, refunds, and user conflicts

Worker - a user that is looking to complete job offers for compensation

Customer - a user that is looking to provide compensation for a job offer

Job - a specific task related to yard work managed by the owner and specified by the customer for a worker to complete

System - the software application this project aims to create; the product being built

User - an owner, customer, or worker with specific access according to the user type

Database - storage for system data to be stored (e.g., user and job details)