# gig.io

## Product Backlog

Geoff Hackett, Sam Fellers, Rishabh Saxena, Andrew Houvenagle, Austin Wirth

## Problem Statement

Many people have menial jobs that they need done and lack the time or ability to do them. Our application aims to solve this problem by providing an environment for users to find cheap, reliable help for completing daily and simple tasks in an efficient manner. Gig.io allows users to post certain tasks they need completed and other users can accept these tasks for an agreed upon pay. A key feature of the application will be the ability for users to bid on the wage they are willing to accept for said task. In addition, a review system will also be available to make sure that users receive help from trustworthy and genuine people.

## Background Info

### Audience

Our audience consists of anyone looking for work without a major time commitment as well as people that have tasks that need done but do not want to hire a professional service. Specifically we believe members of an older generation who need manual labor done for them is a good target audience for the portion of users who need tasks done; and we believe students will make up a significant portion of users looking for tasks to do. Students typically have very difficult schedules to work around. Rather than looking for a job with hours flexible enough for them, they are able to decide when they work with Gig.io.

### Similar Platforms

There are a few applications that could be argued Gig.io may resemble. The first one is Craigslist. Craigslist is essentially a forum environment for people within a certain area, it handles everything from discussion forums to romantic meetups and includes a space for people to offer labor services. Gig.io is much more specific, it only allows legitimate users who need work done to post. It also offers a much safer way to handle the transactions, being done on the website instead of in person, anyone who violates terms will be banned from the website. The second similar platform is Fiver. Gig.io is different because it is not strictly online tasks and it allows the person who needs the work done to post rather than someone who is willing to do a certain task. Gig.io also offers a bidding feature, allowing you to get the best price for whatever you need done.

## Limitations

The main limitations of similar apps is the payment systems and the potential for abuse of the applications. The bidding system of our application will ensure competitive pricing for jobs. In addition, our built-in payment system will make transactions simple and secure. We will include rating and reporting systems to provide conflict resolution tools to users. A user will be able to report another user for failure to complete a task, not showing up, or not paying the agreed upon amount. The ratings will let users know that they can trust the job poster or the bidder and leave feedback on a completed task.

## Functional Requirements

| Number | Description |
|---|---|
| 1 | As a user, I would like to be able to create an account with a username and password. |
| 2 | As a user, I would like to be able to log in to my account. |
| 3 | As a user, I would like to be able to log out of my account. |
| 4 | As a user, I would like to be able to create a job post, including the description, the current bid, the location, who posted it, and the rating of the user who posted the job. |
| 5 | As a user, I would like to be able to change my username and password. |
| 6 | As a user, I would like to be able to participate in the bidding process for a job. |
| 7 | As a user, I would like to see all of the available job postings within a given distance. |
| 8 | As a user, I would like to be able to filter jobs by current bid, rating, etc. |
| 9 | As a user, I would like to be able to view my current job posts and see all bids and the ratings of the people who place the bids. |
| 10 | As a user, I would like to submit a review for another user based on their performance of a job. |
| 11 | As a user I would like to be able to view my ratings for posting jobs and for completing jobs. |
| 12 | As a user who completed a job for someone, I would like to be able to submit a review for the job poster. |
| 13 | As a user, I would like to perform a transaction, or pay another user for completing a task. |
| 14 | As an administrator, I would like to ban or manage users within our system. |

| | |
|---|---|
| 15 | As a user I would like to be able to report a user for not completing a job as specified or not showing up. |
| 16 | As a user I would like to be notified if a report is submitted against me. |
| 17 | As a user, I would like to be notified if I have won the bid for a job. |
| 18 | As a user who has posted a job, I would like to be able to end a bid whenever I want, or close it with no winner if I no longer need the job done. |
| 19 | As an administrator, I would like any job posting that has had no activity for 30 days to be automatically deleted. |
| 20 | As a user, I would like to receive notifications via email. |
| 21 | As a user, I would like to be able to track a job posting. (If time permits) |

## Non-Functional Requirements

### Security

The transaction system must be secure and reliable. Users must know them at they can trust our system with their money and payment information. The server should securely store user information in the database using modern cryptographic techniques.

### Usability

The interface must be user-friendly and intuitive to use. Because our application is not targeted specifically at the tech industry, our website must be usable by anyone. This means that the layout must be simple, consistent, and easy to learn. Since the job postings are user generated, the layout must be adaptable and responsive in order to display information in a clear, readable manner.

### System Design

We will use Node.js for the development of the backend. Node.js is advantageous for performance reasons and does not require developers to learn a new language in addition to javascript. A database will be used to store user information in a way that is both efficient to develop and efficient to use. The application must be able to integrate with PayPal. This will simplify development of the payment system and will provide developers with confidence that the payment system will function as intended. In addition, the application must be accessible from a variety of web browsers. This will ensure that users across a variety of platforms and browsers will be able to use Gig.io.

### Scalability

The system should able to handle a traffic load of at least 100 users and should be able to handle large loads of traffic gracefully. Modern web applications tend to receive a lot of traffic and must be equipped with the resources to handle the large amount of users. Failing to accomplish this task could result in a poor experience for users and potentially crash the application as well.