Team 8 (Local Link)

1. **Tristen Seng**
   * **Student ID**: 025683201
   * **Email**: tristen.seng@student.csulb.edu
2. **Nathan Wakefield**
   * **Student ID**: 025628445
   * **Email**: nathan.wakefield@student.csulb.edu
3. **Frederick Peralta**
   * **Student ID**: 005739394
   * **Email**: frederick.peralta01@student.csulb.edu
4. **Francisco Lopez-Morelos**
   * **Student ID**: 027045708
   * **Email**: francisco.lopez-morelos01@student.csulb.edu

## Preface

The purpose of this document is to be able to describe the web-application “Local Link” in detail. We will review specifics on how the application works, the intended audience, and use cases and user stories.

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 0.0 | 3/15/2024 | Initial Documents |
|  |  |  |

### Purpose

This document serves as a comprehensive guide for the development and understanding of the software project titled "Local Link"

### Audience

The intended audience of this document includes project stakeholders, developers, testers, and anyone involved in the project lifecycle.

## Introduction

“Local Link” will be a web application used for connecting communities together. Workers will be able to list their own qualifications for home improvement (upon verification), while users can list their needs. The workers will then be able to accept jobs based on their qualifications and the app will facilitate communication between the two parties.

### Project Overview

"Local Link" is a web-based home improvement job listing system designed for small businesses and people in need of home improvement. It streamlines job postings, communication, and payment between parties.

### Project Goals

* Host job postings for home improvement.
* Host job listings for home improvement needs.
* Provide transactions.
* Have the ability to rate the performance of the worker who completed the job.

## Glossary

(Define key terms and acronyms used throughout the document, unless they are commonly known to each possible stakeholder (e.g., “Cell phone”) AND used with their common meaning. Do not expect your stakeholders to be experts. If in doubt, define a term.)

* **Employer:**  A service seeker that hires workers to complete a job.
* **Worker:** A service provider with required skills hired by an employer.

## User Requirements and Use Cases

(Outline what the system must do from the user's perspective. User stories need to use the format discussed in class and on our slides. Use cases provide detailed scenarios of system interactions.)

### User Stories

(A collection of user stories that apply to the project.)

1. As a registered user, I want to list a job for a plumber and find qualified individuals in my area that are highly rated and can do the job well.
2. As a professional in home design, I want to be able to list my home design services to individuals who are in need.
3. As a plumber, I want to excel and get high ratings to get more jobs in my local area.
4. As a registered employee, I want to view worker postings so that I can choose the worker with the skillset that I need.
5. As a registered worker, I want to view job requests so that I can choose which job to accept.
6. As a registered worker, I want to be able to reject job requests so that my job list is not big,
7. As a registered employee, I want to rate and review workers so that other employers can make good hiring decisions.
8. As a registered worker, I want to add skills to my profile so that they can be advertised in job postings.
9. As a registered employee, I want to filter job postings so that I can select the correct worker.

### Use Case: Worker Accepts a Job

|  |  |
| --- | --- |
| Identifier | UC-5 View Job Request |
| Purpose | Remove job posting from list of available jobs­­­­­­ |
| Requirements | User Story #5 |
| Development Risks | None |
| Pre-conditions | Worker is on “available jobs,” page |
| Post-conditions | Job is accepted |

**Table 1: Typical Course of Action**

|  |  |  |
| --- | --- | --- |
| Seq# | Actor’s Action | System’s Response |
| 1 | Worker navigates through “jobs available,” page | System provides all job requests |
| 2 | Worker can look through any requests and can accept them | System provides worker information about the employers needs |
| 3 | Worker can accept any request by clicking the “accept” button | System notifies employer that request was accepted |
| 4 |  | System removes job from “jobs available,” page |
| 5 |  | System pops up an option to automatically deny all other jobs |

**Table 2: Alternate Course of Action**

|  |  |  |
| --- | --- | --- |
| Seq# | Actor’s Action | System’s Response |
| 1 | Worker looks through the “jobs available,” page | System provides all job requests |
| 2 | Worker looks though all the job requests | System provides worker information about the employers needs |
| 3 | Worker accepts the request by personally contacting the employer |  |
| 4 |  | After 30 days all job requests are automatically removed |

**Table 3: Exceptional Course of Action**

|  |  |  |
| --- | --- | --- |
| Seq# | Actor’s Action | System’s Response |
| 1 | Worker ignores any request notifications |  |
| 2 | Worker stops using the app but doesn’t disable account |  |
| 3 |  | After 30 days all job requests are automatically removed |

## System Architecture

(Describe the high-level design of the software.)

### Components

1. **Frontend**: Web-based user interface (built with React).
2. **Backend**: RESTful API (built with Node.js and Express).
3. **Database**: MySQL for data storage.
4. **Authentication**: OAuth 2.0 for user authentication.

### Deployment Diagram