

Team 6 (Local Services Market)

1. **Hamzah Qutob**
 - o **Student ID:** 025591616
 - o **Email:** Hamzah.Qutob@student.csulb.edu
2. **Toan Tran**
 - o **Student ID:** 027297970
 - o **Email:** Toan.Tran02@student.csulb.edu
3. **Ivy Ly**
 - o **Student ID:** 026840279
 - o **Email:** Ivy.Ly02@student.csulb.edu
4. **Eren Yolcuoglu**
 - o **Student ID:** 028994899
 - o **Email:** eren.yolcuoglu01@student.csulb.edu
5. **Elias Woldie**
 - o **Student ID:** 027805724
 - o **Email:** elias.woldie01@student.csulb.edu

Preface

Version	Date	Changes
1.0	3/14/2024	Initial Version

Purpose

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

Audience

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

Introduction

The demand for accessible and reliable local services has never been higher. Whether its individuals or businesses, people are seeking efficient ways to find or hire skilled service providers in their communities. However, the process of connecting these service providers with the seekers comes with challenges, including trust issues, unskilled workers, inefficient communication, and payment systems. However, "SkillMatch" is a platform designed to bridge the gap between service providers and those who seek their services.

The goals of this project firstly is to create a vibrant marketplace where service providers are able to show their skills, set the hours they want to work at, and find work opportunities at what they are good at. Secondly is to offer the users of the communities a platform to find, hire, and review local talent that match their specific service needs. Whether its home repairs, cutting hair, or any other service, SkillMatch aims to be the go-to service for all local service needs

With SkillMatch, hopefully we are able to build a community where there is trust, quality, and convenience for all the users. We hope to keep evolving and adapt to the needs of our users and make sure that SkillMatch is the best place to go for any local service market.

In this document, we outline a guide for the development and deployment of "SkillMatch." The user requirements and use cases, the System Architecture, and the Deployment Diagram. We will also dive into what goes into SkillMatch, we have stakeholders, developers, testers, and the end-users.

Project Overview

SkillMatch is a web application that allows service providers ("workers") and service seekers ("employers") to connect, meet, and provide/ receive a service.

Project Goals

- Create a platform that facilitates services for service seekers
- Helps service providers find work and grow their business
- Expand the range of services offered to cater to diverse needs

Glossary

- **Employer:** A user that's seeking for a service provided by a worker.
- **Worker:** A user that's providing a service for an employer.
- **Service:** The type of work or help (i.e., housekeeping, plumbing, gardening, etc.) provided or received.
- **API:** Application Programming Interface.

User Requirements and Use Cases

User Stories

1. As a service-seeking user, I want the option to create an account as an employer so that I can view worker postings to find services to meet my needs.
2. As a service-provider user, I want the option to create an account as a worker so that I can advertise my skills on my profile and get hired by a potential employer.
3. As a worker, I want to be able to reject or accept incoming requests for my service so that they won't overlap in my availability schedule.
4. As an employer, I want the option to filter worker postings by location, rate, and reviews so that I can find good workers within my local area.
5. As an employer, I want the option to request for a worker's service after reviewing their profile so that a certain task can be completed that meets my needs.
6. As an employer, I want the option to rate/review a worker's service so that other users are aware of the type of service that will be provided by the worker whether good or bad.
7. As a Service provider, I want to set my availability schedule so that I can plan my work efficiently and manage my time effectively.
8. As an employer, I want to receive timely updates on the progress of the services I've hired so that I know when a service is complete.
9. As a Service provider, I want to have a secure payment process with transparent pricing details so that I can ensure a trustworthy transaction.
10. As an employer, I want to be able to communicate with service providers through an integrated chat or phone call so that I can answer or ask any clarifying questions.

Use Case: View Worker Profile

Identifier	View Worker Profile
Purpose	See worker's information and credentials to verify if the worker's skills are fit for the needed service by the potential employer.
Requirements	Worker account must exist to create a worker profile/posting
Development Risks	None
Pre-conditions	Employer is on the worker posting page.

Post-conditions	Worker profile is visible for the potential employer.
-----------------	---

Table 1: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Employer views worker postings	
2	Employer selects a worker posting	
		Upon selection, the system will look up worker's information that has been saved in the worker's information database.
		A pop-up window will appear displaying the worker's information

Table 2: Alternate Course of Action

Seq#	Actor's Action	System's Response
1	Employer views worker postings	
2	Employer filters postings by location, rate, or reviews	
3	Employer selects a worker posting	
		Upon selection, the system will look up worker's information that has been saved in the worker's information database.
		A pop-up window will appear displaying the worker's information

Table 3: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	Employer views worker postings	
2	Employer filters postings by location, rate, or reviews	
		System is unable to find information in the database based on filter selection.
		No worker posting is displayed.

System Architecture

Components

11. **Frontend:** Web-based user interface (built with React).
12. **Backend:** RESTful API (built with Node.js and Express).
13. **Database:** MySQL for data storage.
14. **Authentication:** OAuth 2.0 for user authentication.

Deployment Diagram