Text

Description automatically generated

## 

## Table of Contents

Contents

[Table of Contents 3](#_Toc88352508)

[Preface 4](#_Toc88352509)

[Purpose of Document 4](#_Toc88352510)

[Documentation Standards 4](#_Toc88352511)

[User Requirements 5](#_Toc88352512)

[Project Overview 5](#_Toc88352513)

[Statement of the Problem 5](#_Toc88352514)

[Non-Functional Requirements 6](#_Toc88352515)

[System Interface Requirements 6](#_Toc88352516)

[Maintainability and Administration Requirements 6](#_Toc88352517)

[Usability Requirements 6](#_Toc88352518)

[System Design 7](#_Toc88352519)

[Project Management 9](#_Toc88352520)

[Team Configuration 9](#_Toc88352521)

[Project Standards and Procedures 9](#_Toc88352522)

[Glossary 10](#_Toc88352523)

[Index 11](#_Toc88352524)

## Preface

## Purpose of Document

* This document is to provide information and reference to “Hire Five Star” web application development procedure. Found in the following sections are purposes of developing the application and business requirements that it will fill. It contains Use case and design diagrams as well as system requirements. Application architecture are illustrated as well as hardware and software requirements for development and deployment. Finally, project timelines are described in detail. The intent of the project is to provide a web application to provide a form of brokerage for customers to find services and for businesses and workers to advertise and gain new clients and business.

## Documentation Standards

* The term “Service Provider” will refer to any business/individual worker who provides work/services to be advertised on Hire Five Star.
* Font used throughout is Calibri – size 12
* All diagrams that are embedded in document are made using Draw.io
* Figma used for all mockups
* Project timelines and Gantt Chart are displayed from Excel documentation

## User Requirements

**Business Overview and Objectives**

Hire Five Star seeks to provide a platform to solve the problem of finding the best quality workers/services/companies and to hire them. The company seeks to implement a web-based interface for potential customers to use for free to log in and search for companies or individuals that provide a service that they are looking for. The interface is meant to be easy to use and provide a reliable means of finding rated services for hire. Conversely, the interface would charge a fee for Service providers to use the platform as a means to advertise their business and reach potential customers. The solution will result in a safe and effective way for both customers and Service Providers to conduct business.

## Project Overview

## Statement of the Problem

* There currently is no easy method of finding the best quality of workers or companies that provide a service and hire them.
* Some workers aren’t finding work

**Project Scope**

* Provide web application that allows customers to search for services.
* Provide a means to compare different service providers.
* Assure the quality of work and payments for both sides
* Allow service providers to post relevant information about their services and pricing
* Ability to generate a contract between customer and service provider in order to ensure quality of work and agreed pricing is met
* Provide a reward system in the form of discounts for high volume users
* Application fee for service providers with monthly fees and percentage of commission from business obtained from application with 3 tier account system for service providers, Gold, silver, and bronze. Higher account tiers will show first in searches.

**System Environment**

* The application will be a web-based application providing and easy to use interface for both customer to seek out services and service provider to advertise and sell services.

**Current System**

* Current existing systems include web browser searching, word of mouth through personal connections and currently knowing a service provider.

**System Requirements Use Case Diagram**

*Diagram

Description automatically generated*

###### **Use Case Descriptions and Scenarios**

|  |  |  |
| --- | --- | --- |
| **Use Case 1: Customer interaction with application**  **Actor description**:Any citizen who signs up as a user of the application and that is searching for specific services to hire a provider of said service. | | |
| **Actor Action** | **System Action** | |
| **1.1 - Login** – Customer will log into app via HTML form with username and password | **1.2**. App will Authenticate provided username and password and if user exists and password is valid, log user into Customer interface. | |
| **1.3 - Search for Services –** Customer will have ability to use search function to enter service name. | **1.4 –** will perform filtered search of requested services by matching words. Will also filter results based on Service Provider tier level and ratings. | |
| **1.5 Contact Service Provider-** will have option and means to contact service provider to initiate negotiations | **1.6 –** Will open chat style messaging system to connect customer and Service Provider | |
| **1.7 – sign contract -** parties will have means to send electronic copy of service contract and add digital signature | **1.8 –** Contract can be sent between parties in e-mail form for digital signature. | |
| **1.9 - Submit Review of Service** Provider – customer will have option to submit review of Service provider upon contract completion. | **1.9.1** – Will post review contents to Service Provider profile as such that it is visible in review portion of profile for customers to view. | |
| **1.9.2 - Receive Incentive Discounts for High Usage -** Incentive discounts will be awarded for users who hire more service providers through app. | **1.9.3 –** track hiring history of Customers in profile. Award incentive discounts based on number of successful hires | |
| **1.9.4 – Report Scam companies to Admin** can select option on Service Provider profile to “Report user”. | **1.9.5 –** on “Report User” selection will provide checkbox form of reason for reporting and send request to admin for investigation | |
| **Use Case 2: Service Provider interaction with application**  Actor description - Any worker or business owner that provides a service and wanting to advertise and sell their services on an efficient platform. | | |
| **Actor Action** | **System Action** | |
| **2.1 - Login** – Service provider will log into app via HTML form with username and password. App will Authenticate and will log into service provider interface | **2.2 -** App will Authenticate provided username and password and if user exists and password is valid, log user into Customer interface. | |
| **2.3 - Create Profile –** will have means to create profile with description of services, bio. May request assistance in account creation | **2.4 –** on profile creation will send account creation request to admin for approval and if approved will create account for Service Provider. | |
| **2.5 - Choose Tier Level –** choose tier level to increase visibility in searches. Gold, Silver and Bronze | **2.6 –** Assign tier level to account to indicate visibility level during searches. | |
| **2.6 - Receive communications from customers –** means to receive messages from customers to provide more information. | **2.7 –** notification of new messages in interface and chat style messaging system will be used for communication with customer for negotiations. | |
| **2.8 - Sign contract –** parties will have means to send electronic copy of service contract and add digital signature | **2.9 -** Contract can be sent between parties in e-mail form for digital signature. | |
| **Use Case 3: Administrator interaction with application**  Actor description – A Designated operator of the application that will provide a form of control of the application use. | | |
| **3.1 – Receive request for account creation –** will approve/disapprove new account request sent when Service Provider creates profile. On request of Service Provider, will aid in the account creation process. | | **3.2 –** will send notification to admin profile with Service Provider profile details for approval. On approval will generate new profile for Service Provider and store in DB. |
| **3.3 = Monitor accounts –** view Service Provider accounts andwill receive reports from customers reporting scam companies and will have means to delete accounts. | | **3.4 –** search function for finding Service Provider accounts and ability to delete fake accounts |

## Functional Requirements

* Allow users to create a profile as either a customer or a service provider
* One log in for all users but depending on profile status will direct user to appropriate interface (customer or service provider)
* Easy to use search option for customers to search for services
* AI virtual assistant (chatbot) to aid in customer user experience
* Direct messaging feature to message service provider for questions and to negotiate for services.
* Customer usage should be tracked to award discount incentives
* Service providers sign up should allow user to indicate tier level. Tier level upgrade at any time will be available. Tier level will dictate visibility.
* Customers will have ability to submit reviews of service providers if they have hired provider
* Service providers will be able to post description of their services and pictures of their work for advertising purposes
* Easy to use username and password reset should be implemented.

## Non-Functional Requirements

* Clean, easy to use Web application interface.
* User log in passwords will be stored in database using hash algorithm
* All HTML forms will be sanitized for exploitations including SQL injection and command injection attacks. Use known good libraries within development frameworks if necessary.
* Tier level of service providers will be a field in service provider table in data base. Search results will be sorted by tier level in the query.
* Reviews submitted by customers will be stored in own table with relationship to the service provider and review can be seen when viewing service provider profile in application. Reviews will be sorted by date submitted.
* Customer incentive discounts will be based on usage **(# of successful hires, not log ins!)** usage will be tracked in database table and incentive field will be calculated based on usage field.

## System Interface Requirements

* Use pre-built A.I virtual assistant to be embedded in application (to be chosen)
* Payment gateway service API (Paypal or similar) to handle transactions between customers and service providers

## Maintainability and Administration Requirements

* Payment methods will have to be monitored and modified on ad-hoc bases to meet PCI requirements, user/industry standards etc.
* U.I will be updated to continue to provide modern user experience based on current industry standards. A modern front-end framework with little chance of deprecation in near future should be chosen to increase lifespan of application without having to perform a complete re-write of codebase.
* Will have administrator log in to handle false profiles or misuse of platform.
* Easy to use username and password reset should be implemented.
* On-going maintenance and administration after handover will be conducted by Application Support Engineer employed by “Hire Five Star”.

## Usability Requirements

* Expected to have varying levels of experience of users
* Focus will be to have a clean and intuitive user interface. Styles and fonts should be easy to see. All functions should be accessible with minimal “clicks”
* For inexperienced users, the AI chatbot should be very intuitive and easy to use to aid in use of the platform.

## Interaction Sequence Diagrams

Service Provider Sequence Diagram

*A picture containing timeline

Description automatically generated*

Customer Sequence Diagram

A picture containing table

Description automatically generated

Admin Sequence Diagram

**A picture containing table

Description automatically generated**

## State Machine Diagram

**Diagram

Description automatically generated**

## Activity Diagrams

Customer Activity Diagram

Diagram

Description automatically generated

Service Provider Activity Diagram

Diagram

Description automatically generated

Admin Activity Diagram

Diagram

Description automatically generated

## System Design

Layered Architecture

*Diagram

Description automatically generated*

Hardware Architecture

* The application frontend will run on client’s personal computer in the browser using modern frontend framework. The application backend will run on webserver with a database configured locally on same server. The application frontend will send http requests to backend on web server and receive a json response with requested data.

*Diagram

Description automatically generated*

**Hardware Platform**

* Development - X64, 2.8Ghz 4 core processor, 8 GB Ram 250GB SSD storage
* Production – 1.6 GHz CPU, 1.75 GM RAM, 1 x 40 GB HDD

Software Platform

* VS Code (or similar IDE)
* Nodejs
* ExpressJS
* react v 17.0.2
* react-dom v 17.0.2
* Babel
* MongoDB
* Bootstrap for React (CDN) – (May also use Semantic U.I and/or Material U.I)

Interaction Model

Log in and sign-up page

A picture containing text, businesscard

Description automatically generated

Graphical user interface

Description automatically generated

Customer and Service Provider Home pages

Graphical user interface, website

Description automatically generated

A picture containing table

Description automatically generated

Service Provider Profile creation/Update profile page and contact page

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Font – Montserrat

**Colour scheme:**

Login/sign-up/contact – Left div – Linear gradient #57A9C2 - #329F3D

NAV Bar - Linear gradient #329F3D - #57A9C2

Customer and Service Provider Home pages – body background - #F2F6F6

Service Provider Profile form - #57A9C2

Buttons - #EA6C6C

Font color (buttons,Nav,Forms) – white

Font Color(Home page Body) – Black

Login/Sign-up/contact page picture – Right Div

- Text, letter

Description automatically generated

Persistence Model

* Will utilize ORM in backend framework or if using NoSQL type database (i.e., MongoDB) will use appropriate query structure as per database.*Diagram

  Description automatically generated*

## Project Management Schedule

###### Project start meeting 11/21/21 11/22/21

###### Project Capstone Document 22/11/2021 30/11/2021

###### Use Case Diagram 1/12/2021 3/12/2021

###### Interaction Sequence Diagrams 2/12/2021 4/12/2021

###### State Machine Diagrams 3/12/2021 5/12/2021

###### Activity Diagrams 4/12/2021 6/12/2021

###### Interaction Sequence Diagrams 5/12/2021 7/12/2021

###### AdminActivityDiagram 6/12/2021 8/12/2021

###### ServiceActivityDiagram 7/12/2021 9/12/2021

###### CustActivityDiagram 8/12/2021 10/12/2021

###### Deployment diagram 9/12/2021 11/12/2021

###### Coding meeting and minutes 10/12/2021 12/12/2021

###### start of coding 11/12/2021 13/12/2021

###### Management 13/12/2021 18/12/2021

###### Git and GitHub 18/12/2021 31/12/2021

###### Merv system research 31/12/2021 8/1/2022

###### Flask and django research 8/1/2022 12/1/2022

###### miscellaneous research 12/1/2022 25/1/2022

###### 2 weeks to close project 25/1/2022 7/2/2022

###### Testing 7/2/2022 12/2/2022

###### Presentation 12/2/2022 28/2/2022

*Chart

Description automatically generated*

## Team Configuration

* Stephen Church – Team Lead/Backend Developer
* Robert deVries – Frontend Developer
* Behnam Hossein – Frontend Developer

## Project Standards and Procedures

* Regular team meetings will occur via Slack group chat in 1–2-week intervals
* Project is stored on remote repository on GitHub
* All team members shall create local repositories on there own personal development computers and will link to remote repository. Team member is to ensure that regular Push/Pulls are conducted to ensure local and remote repositories are kept as up to date as possible.
* All team members are responsible for creating/working in local branches on new features. Once feature work is complete a pull request will be created in remote repository with remaining team members selected as reviewers for approval. Once all parties have approved of changes, the branch will be merged with main.

## Glossary

**Use case** - A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

**Interaction diagram** - An interaction diagram is used to show the interactive behavior of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system.

**Sequence Diagrams** – Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

**State Machine Diagram** – The behavior of an entity is not only a direct consequence of its inputs, but it also depends on its preceding state. The past history of an entity can best be modeled by a finite state machine diagram or traditionally called automata. State Machine Diagrams (or sometimes referred to as state diagram, state machine or state chart) show the different states of an entity. State machine diagrams can also show how an entity responds to various events by changing from one state to another. State machine diagram is a UML diagram used to model the dynamic nature of a system.

**Activity Diagrams** – An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

**Layered Architecture** – The layered architecture style is one of the most common architectural styles. The idea behind Layered Architecture is that modules or components with similar functionalities are organized into horizontal layers. As a result, each layer performs a specific role within the application.

**Package Diagrams** – Package diagram, a kind of structural diagram, shows the arrangement and

organization of model elements in middle to large scale project. Package diagram can show both structure and dependencies between sub-systems or modules, showing different views of a system, for example, as multi-layered (aka multi-tiered) application – multi-layered application model.

**frontend framework** - A front end framework is a platform for developing your front end. It generally comprises some way to construct your files, associate data with DOM elements, style your components and make AJAX requests.

**backend framework** - Backend frameworks are libraries of server-side programming languages that help build the backend structure of a website. Backend frameworks provide ready-made components for developing a dynamic web application.

**Mern stack** - MERN stands for MongoDB, Express, React, Node, after the four key technologies that make up the stack.

MongoDB - document database

Express(.js) - Node.js web framework

React(.js) - a client-side JavaScript framework

Node(.js) - the premier JavaScript web server

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js the popular and powerful JavaScript server platform.

**Nodejs** - Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

**ExpressJS** - Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js

**React** - React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications

**Babel** – Babel is a free and open-source JavaScript transcompiler that is mainly used to convert

ECMAScript 2015+ code into a backwards compatible version of JavaScript that can be run by older JavaScript engines. Babel is a popular tool for using the newest features of the JavaScript programming language

**MongoDB** – MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License.

**Bootstrap** – Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

**Semantic U.I** – Semantic UI is a front-end development framework similar to bootstrap designed for theming. It contains pre-built semantic components that helps create beautiful and responsive layouts using human-friendly HTML.

**Material U.I** – MUI Core (formerly Material UI) is the React UI library. Follow your own design system, or start with Material Design.