SOFTWARE REQUIREMENT SPECIFICATION

Of Website HandyHire – Hire as You Desire

Submitted By:

Team SUST_404BrainNotFound

Executive Summary

Introducing **HandyHire** - Hire as You Desire. HandyHire envisions a world in which individuals are able to explore their innate household and handyman skills, thereby eradicating classism. Our website HandyHire intends to provide a virtual platform that connects people who are willing to provide odd-job or handyman assignments with others who are willing to accept them as part-time, full-time, or even contract work. Unique to our platform is the fact that we enable people to hire others for customized odd jobs, i.e., jobs that typically lack a label. By this they can earn some extra cash, regardless of their age, size, gender, educational background, experience, or skill set. As we provide positions that may require little to no experience (e.g., line-standers, workout assistants, etc.), anyone with a willingness to work can join. For instance, you enjoy gardening but require assistance with that task. So, you can post on our website in the customized odd-jobs section what you want one of our employees to perform for you. Whoever is willing to complete the task will react. Based on their profiles, you can choose any of the respondents who appeal to your preferences. In addition, connection fees, premium service fees, and Google AdSense would be significant sources of income. We propose to directly engage workers for premium services; district supervisors, divisional supervisors, state supervisors; technical teams, marketing teams, grooming teams, etc., using mass office personnel and on-field supervisors. We seek to safeguard the safety of both our employees and consumers by maintaining their authentic information. The concept of HandyHire adheres precisely to SDG-8 as it meets all three requirements.

Table Of Contents

Table Of Figures

Introduction

This chapter describes the whole overview, objectives of the project “REACH” for the audience.

Purpose

To Create a virtual platform where people can find workers to do their odd jobs easily and the people of minimum skill can gain some extra cash doing some odd jobs.

Identifying Stakeholders

Stakeholders are the list of people who will contribute the input as requirements. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”. They have a positive or negative influence on the completion of this project.

For the webApp ‘HandyHire”, we’ve identified our stakeholders.

General Users: General Users are the heart of this project as this is turning out to be a public service app.

There will be two kinds of general users in this app.

- **The Customer:** They will be using this app for getting the solution of their problem by hiring workers or freelancers for part-time or fulltime job.

- **The Workers:** Those who are looking for part-time jobs for earning extra-cash will use the app

Service Provider: They will ensure the security and quality of the worker using the application, they will update workers info, their active status, order details, payment details, rating etc.

Recognizing Multiple Viewpoints

After going through a market survey analysis for general users, we've found their viewpoint of them

General user's viewpoints:

- Easy to use the map
- Make connection in minimum possible time
- Getting order from the nearest locality
- Jobs that don't require specific skills or experience
- Improvement of base asking price after good job and review
- Security of the both workers and customers
- Getting notification of job on their connection (of job)
- Getting permanently hired by the authority
- Getting payment receipt to customers

The HandyHire's View Points:

- Providing the solution of customer's any kind of day-to-day life problems keeping a section to describe their problem, having some specific job also.
- Ensuring the security of both workers and customers by authentic details on opening a new account
- Giving notification to both client and workers for job post or search or comment

- Chat through messaging system
- Tracking the workers to find the locality of them
- Updating the ratings and payment details

- No intention to hire the workers as they will individually connect with each other
- Update after successful service

Analyzing Requirements:

Common requirements:

- Getting order from the nearest locality
- Jobs that don't require specific skills or experience
- Improvement of base asking price after good job and review
- Security of the both workers and customers
- Getting notification of job on their connection (of job)

Conflicting requirements:

- Get permanently hired by the authority
- No intention to hire the workers as they will individually connect with each other

We've finalized the requirements in basis of priority

Final requirements

- Getting order from the nearest locality
- Easy to use the map
- Make connection in minimum possible time
- Jobs that don’t require specific skills or experience
- Improvement of base asking price after good job and review
- Security of the both workers and customers
- Getting notification of job on their connection (of job)
- Update after successful service
- Give payment receipt to customers
- Chat through messaging system

Scenario Based Model

Introduction

Scenario based modeling is more understandable than anything. So, developing such prototype helps to involve all of the developers effectively. Our primary focus was to develop the model describing 'What?' not 'How?'. We've confined our models within the business domain of the customers and tried to keep the models as simple as possible.

For this purpose, we need to define two things as these will be used on a regular basis on the upcoming scenario-based models –

Primary Actor:

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor:

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

Use Case Diagram

Actually, it describes how the end users will interact with the software and is expressed with the primary and the secondary actors of the software. To begin developing a set of use cases, list the functions or activities performed by a specific actor.

Use case: REACH

Primary actor: Customers and Workers

Goal in context: To Connect the workers and the customers

Preconditions: App should be installed in android phone with active internet connection

Trigger: The Customer login into the app and search and connect a worker to do a job

Scenario:

For Customer:

1. login into the app
2. turn on GPS
3. Search for desired service
4. Selects work
5. Choose a worker
6. Contacts with him and hire him
7. Pay through the app
8. Give rating / review
9. Subscribe to Premium

For Workers:

1 login into the app

2 Turn on GPS.

3. wait for notification for work or search for post of work

4. Response to customer

5. Get hired and earn point by completing that task

Exceptions:

1. User forgot the password
2. Wrong account detail during opening a account
3. User don't turn on the location
4. Customer choose wrong payment gateway
5. Customer payment account is empty

Priority: Essential, must be implemented

When available: First increment

Frequency of use: Many times, per day

Channel to actor: via Webisite browser and internet connection

Secondary actors: Support technician, Service provider

Channels to secondary actors:

Support technician: Internet and computer network server

Use Case Diagram

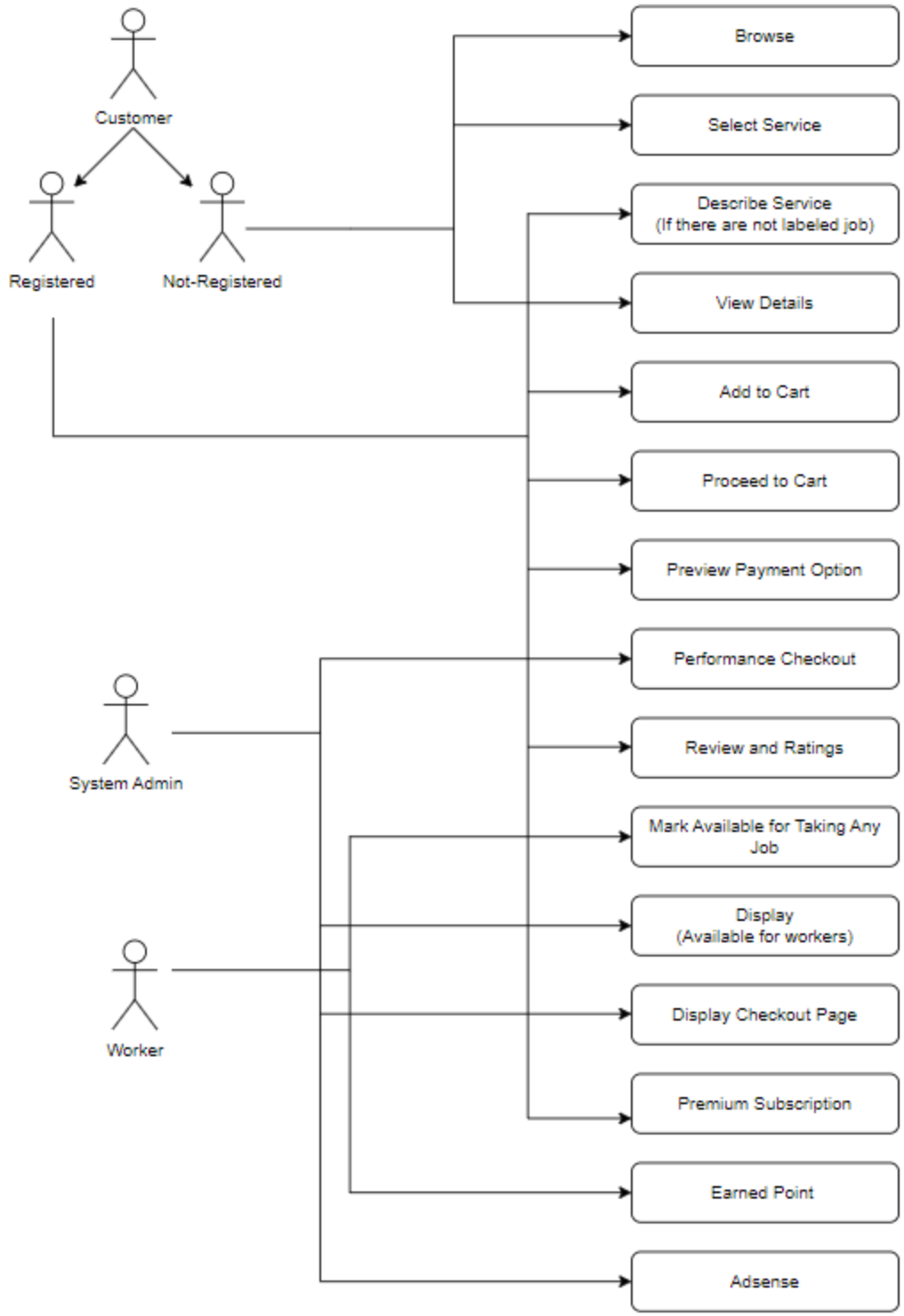

Figure 1

: Use Case Diagram of REACH

Ativity diagram:

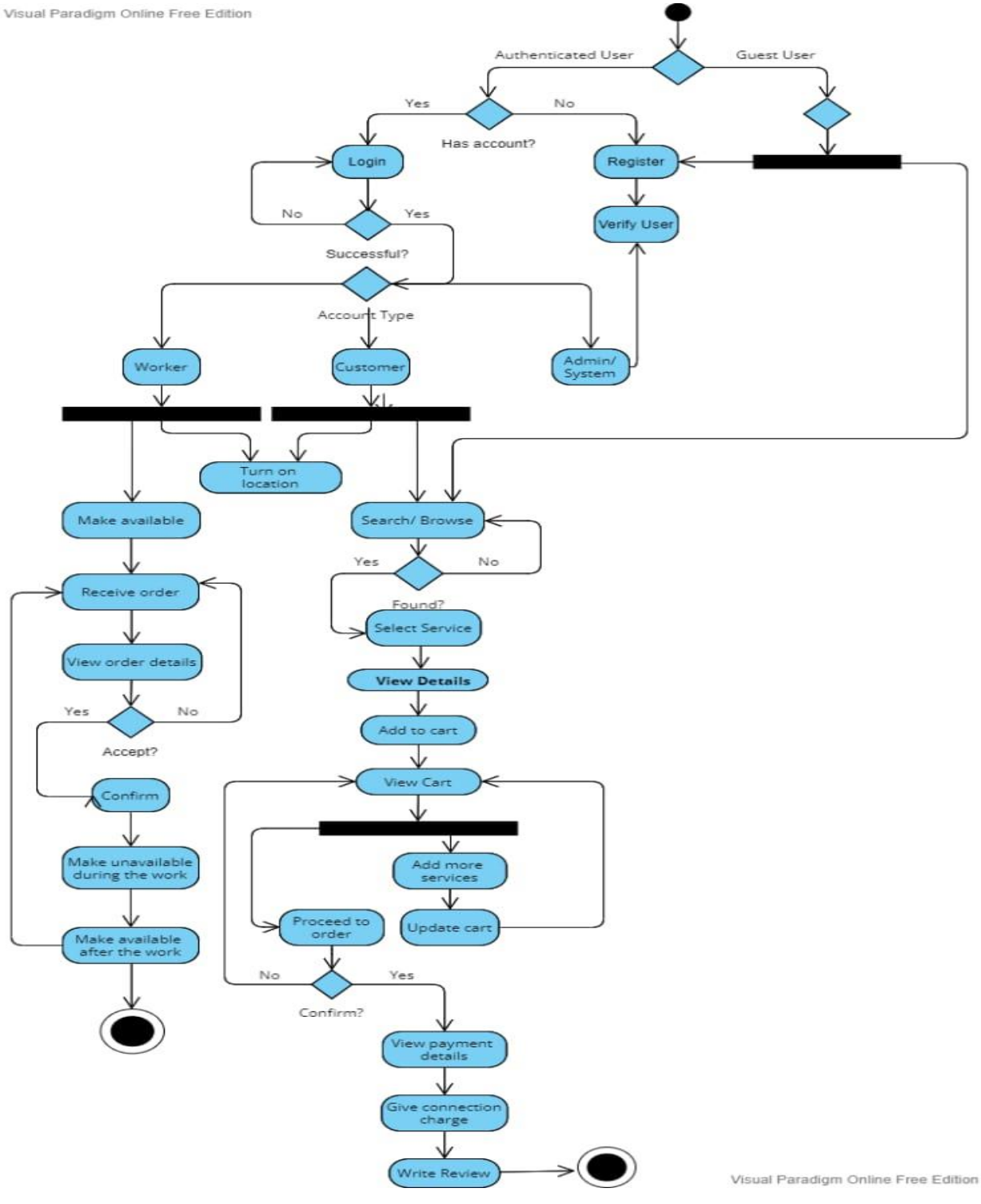

Figure 2 : Activity Diagram of REACH

Swim Lane Diagram

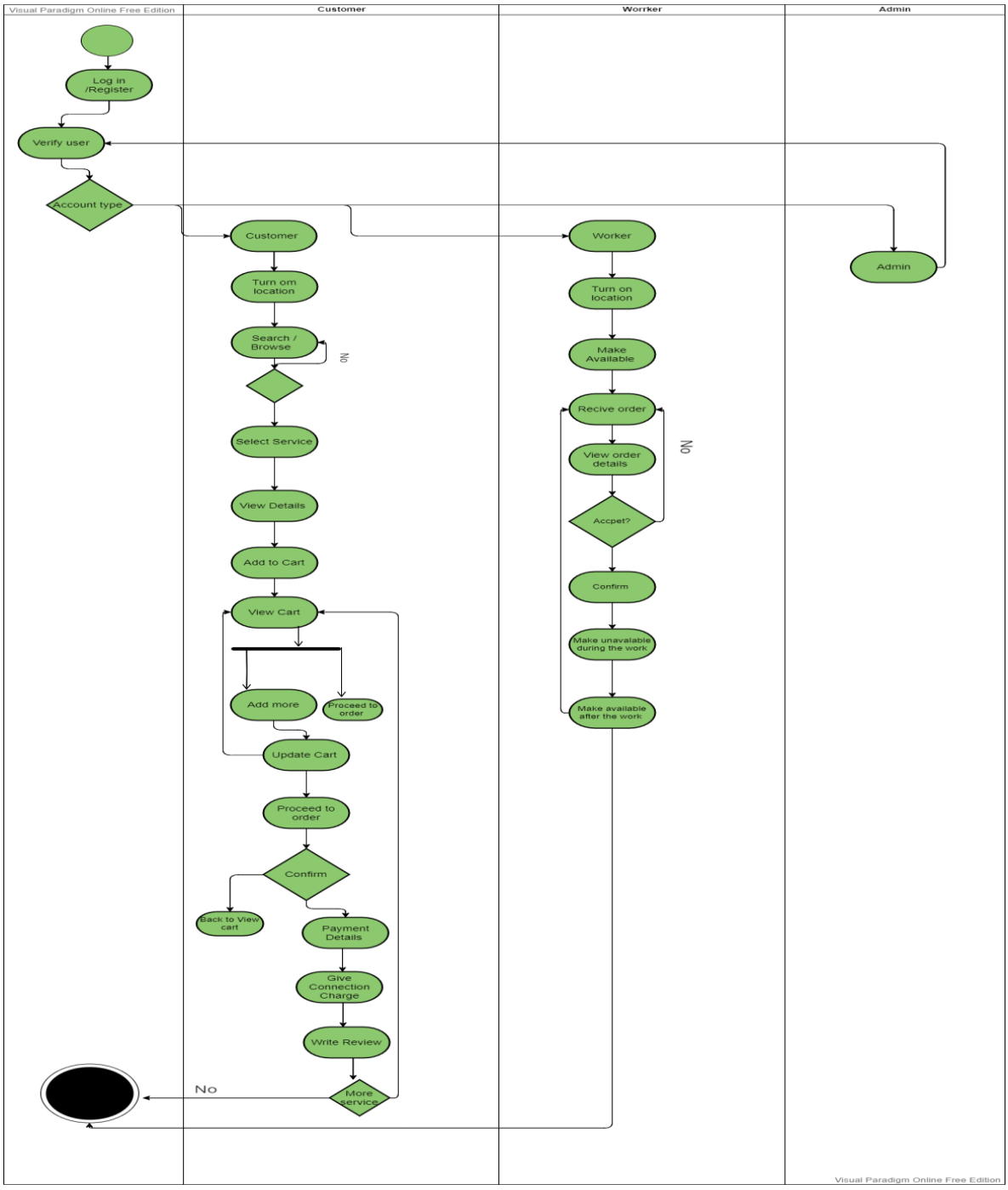

Figure 3 : Swim Lane diagram for the REACH

Data Model

ER Diagram

After finding the Data Object we've drawn Entity – Relationship Diagram (ER diagram) for better understanding of data flow and storing to database-

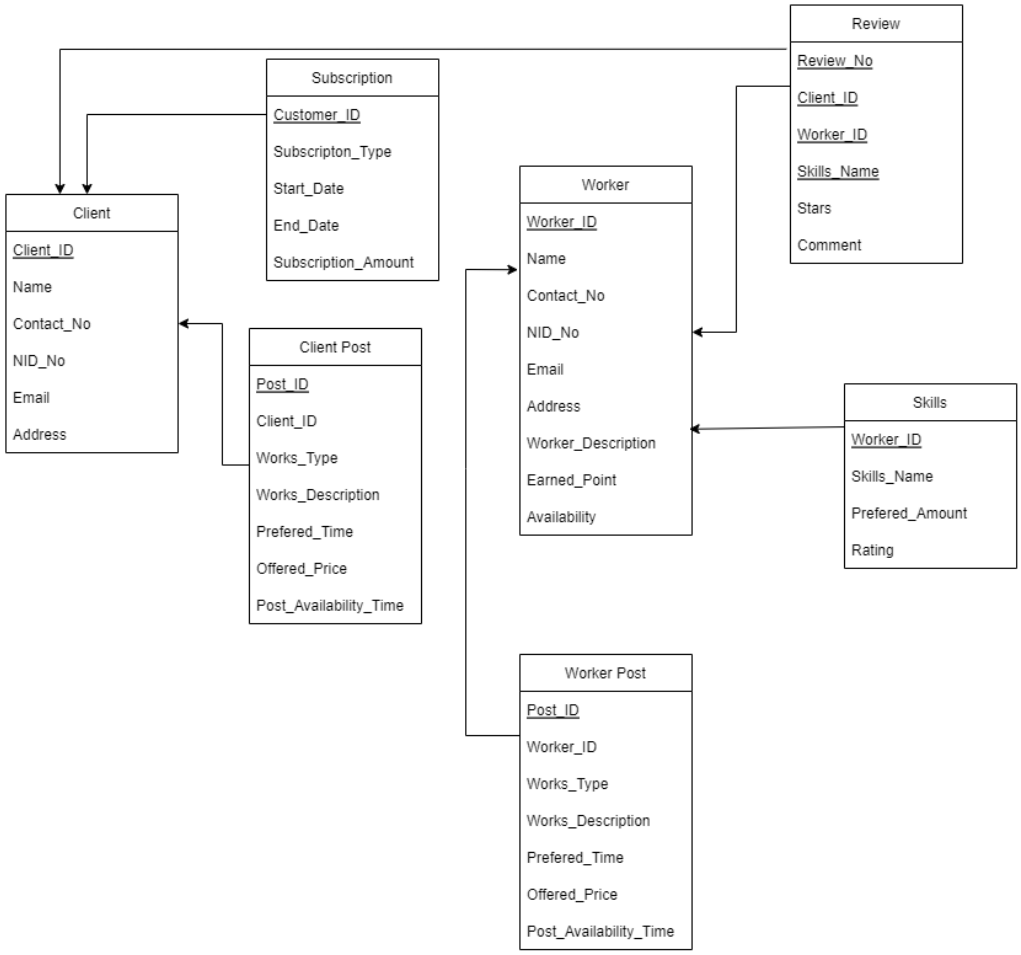

Figure 4 : ER diagram

Database Schema Tables:

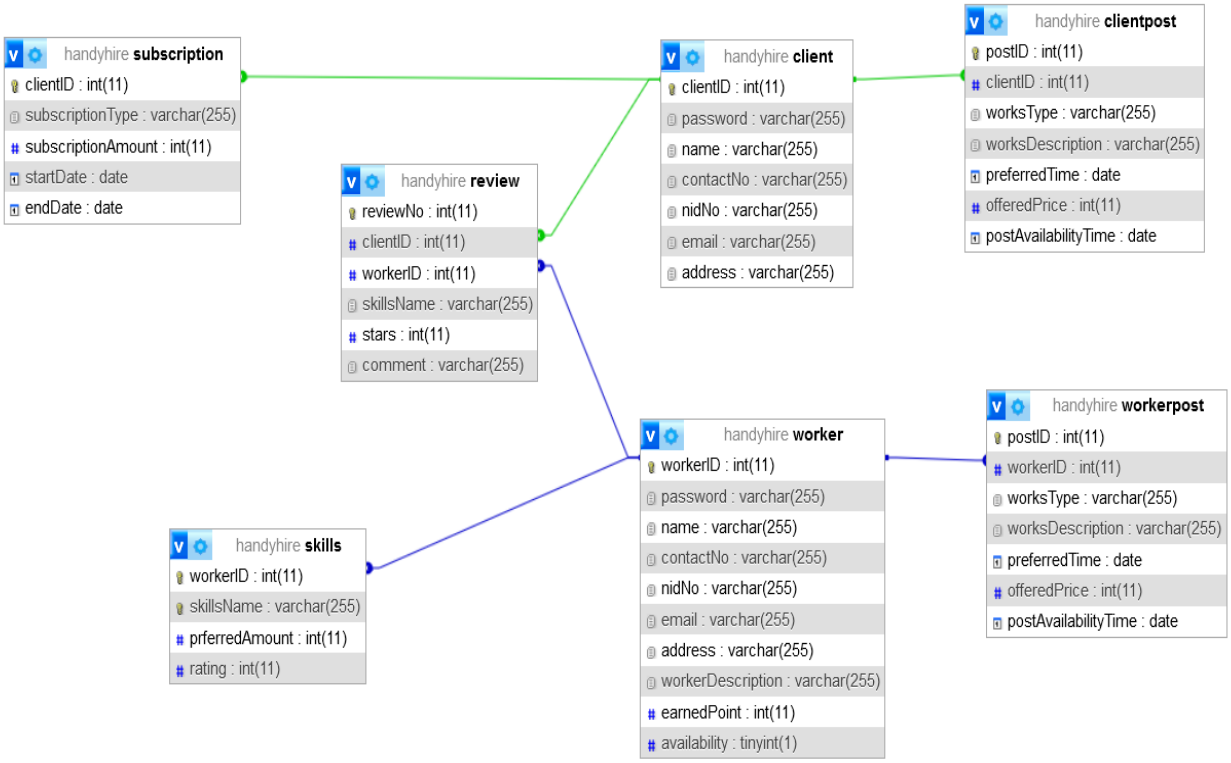

Figure 5: Database Schema

Flow Oriented Model

Data Flow Diagram of Level: **0** of " HandyHire"

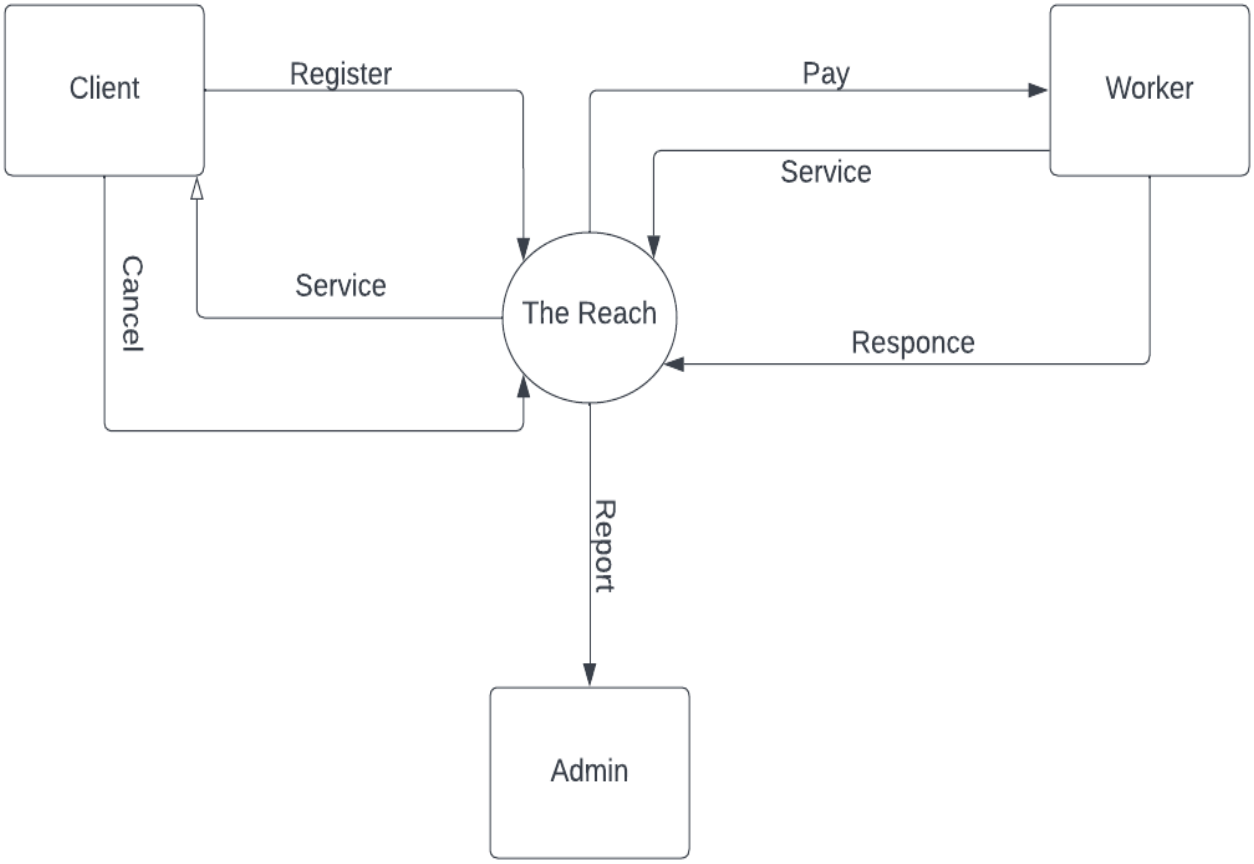

Figure 6 : Leve: 0 DFD of REACH

DFD of Level: 1 of “HandyHire”

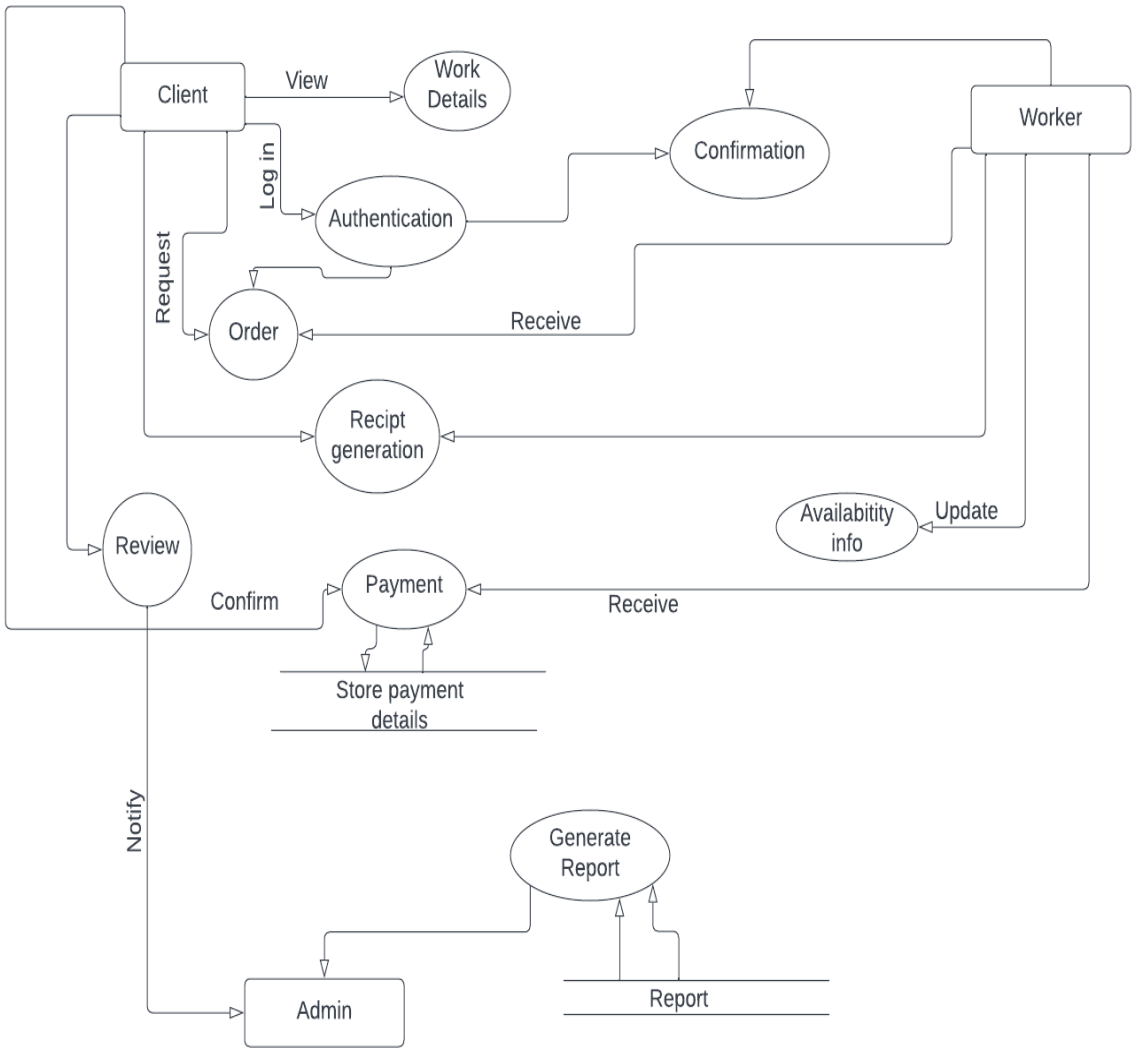

Figure 7 : Level: 1 DFD of REACH

DFD of Level: 2 of “HandyHire”

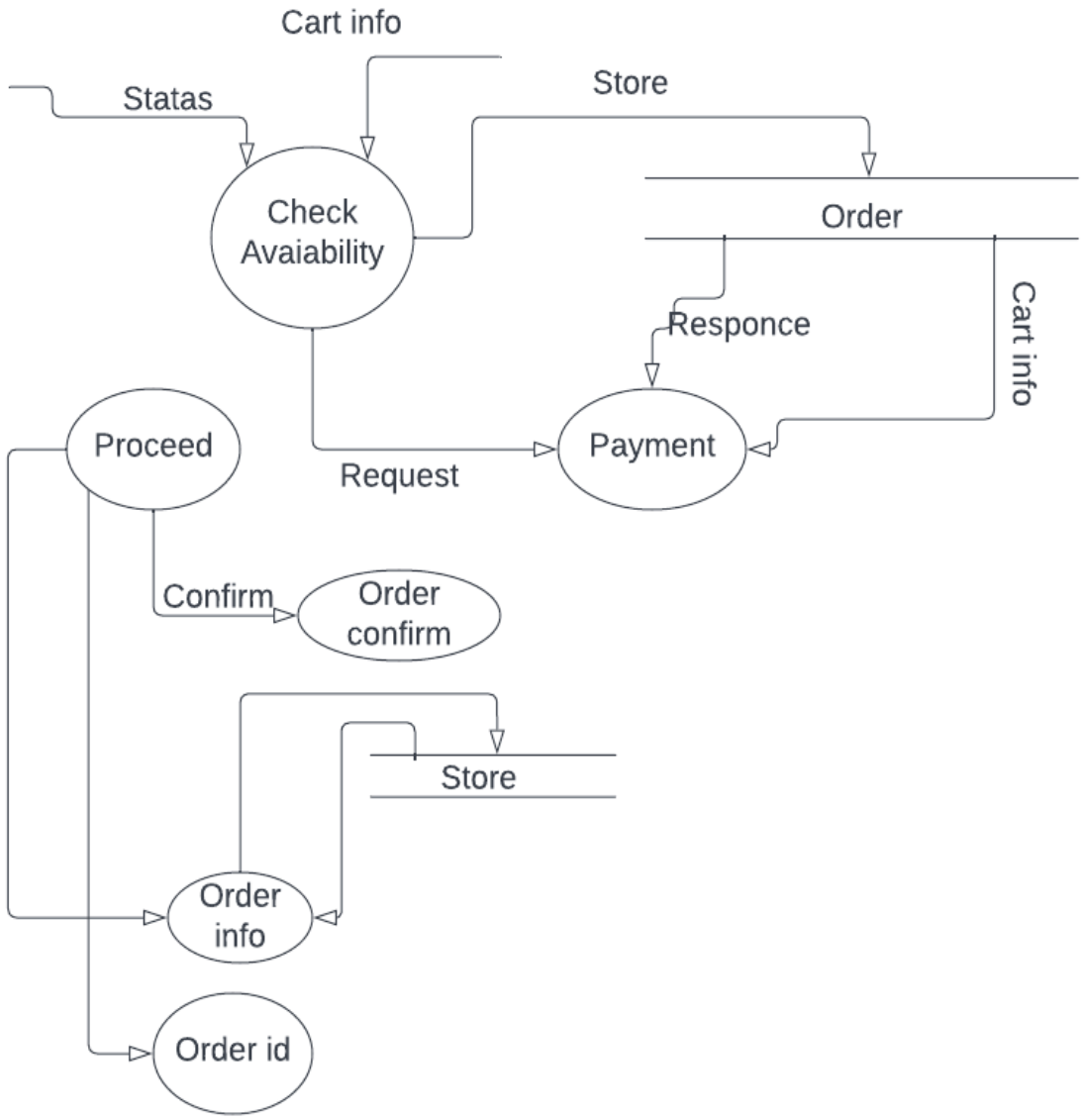

Figure 8 : Level 2: DFD diagram of REACH

Conclusion

It is extremely difficult to represent something real on paper. We are glad that we have been able to complete the requirements analysis of 'REACH' app and publish the SRS of the project. I believe that this report has been written as easy as possible manner and contains all the material necessary to have a complete understanding of the concept. I hope that any reader who pursues this document may quickly grasp the concept behind the 'REACH' application. I also hope that it will be an easy-to-follow document for implementing the application!