

**BTech/II Year CSE/III Semester**

**19CSE202/Database Management Systems**

**Project Review**

|  |  |
| --- | --- |
| STUDENT NAME | STUDENT ROLL NUMBER |
| M. Thanusha Reddy | CB.EN.U4CSE22137 |
| M. Harini Krishna | CB.EN.U4CSE22138 |
| N. Ragavarshitha | CB.EN.U4CSE22139 |
| Prachi kumari | CB.EN.U4CSE22145 |
| R. Gowthamee Chandra | CB.EN.U4CSE22158 |

REGISTRATION CENTER FOR UNORGANISED WORKERS

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Tittle | Page no |
| ABSTRACTION | 01 |
| FUNCTIONALITIES | 02-03 |
| MODULES | 04-05 |
| ER-DIAGRAM | 06-07 |
| SCHEMA DIAGRAM | 08 |
| TABLES CREATION AND INSERTIONS | 09-11 |

01

ABSTRACTION

The "Registration System for Unorganized Workers" is a website designed to manage the registrations of workers in the unorganized sector. The unorganized workforce constitutes a significant portion of the labour market, often facing challenges due to lack of formal recognition and access to social benefits. This project aims to bridge the gap by providing a centralized platform to collect, organize, and analyze data related to unorganized workers' registrations.

The DBMS architecture encompasses a well-structured database schema that captures essential details about individual workers. This includes personal information, contact details, occupation, skills, and other pertinent data. The system offers an intuitive user interface, facilitating seamless data entry during the registration process.

This project aligns with the objective of promoting equity and inclusivity in labor markets. By harnessing the capabilities of modern technology, the Registration System for Unorganized Workers takes a significant step towards recognizing, supporting, and empowering workers who form an essential but often overlooked part of our economy.

02

FUNCTIONALITIES

Key Functionalities :

1. **User Registration and Authentication:**

* Allow workers and clients to register in the system with their personal details.
* Implement user authentication and role-based access control to ensure data security.

1. **Worker Management:**

* Store and manage information about workers, including their personal details, skills, contact information, and location.
* Allow workers to update their information as needed.

1. **Client Management:**

* Store and manage information about clients, including their personal details, contact information, and location.
* Track the services requested by clients and their preferences.

1. **Skill Management:**

* Maintain a database of different skills that workers possess.
* Allow for the addition, modification, and removal of skills.

**03**

**5.Service Request and Assignment:**

* Enable clients to request specific services.
* Match clients with suitable workers based on skills and availability.
* Track service assignments and status.

**6.Payment Processing:**

* Record payment information for completedservices.
* Support various payment methods, such as cash, digital payments, or other options.
* Generate invoices and receipts for clients and workers.

**7.Search and Filter:**

* Implement search and filter options for clients to find suitable workers and for workers to locate potential clients.

04

MODULES

**1.User Management Module:**

* User Registration
* User Authentication
* User Roles and Permissions

**2.Worker Management Module:**

* Worker Registration
* Worker Profile Management
* Skill Information
* Availability and Schedule

**3.Client Management Module:**

* Client Registration
* Client Profile Management
* Service Request History

**4. Skill Management Module:**

* Skill Database
* Skill Search and Filtering

**5.Service Request and Assignment Module:**

* Service Request Submission
* Matching Algorithm for Worker-Client Pairing
* Service Assignment Tracking

05

**6. Payment Processing Module:**

* Payment Recording
* Payment Method Handling
* Invoicing and Receipt Generation

**7.Feedback and Ratings Module:**

* Feedback Submission
* Rating System
* Review and Analysis

06

**ENTITY-RELATIONSHIP DIAGRAM**

ENTITIES:

* Registration\_Center
* Worker
* Skill
* Client
* Payment
* Receive
* Pay

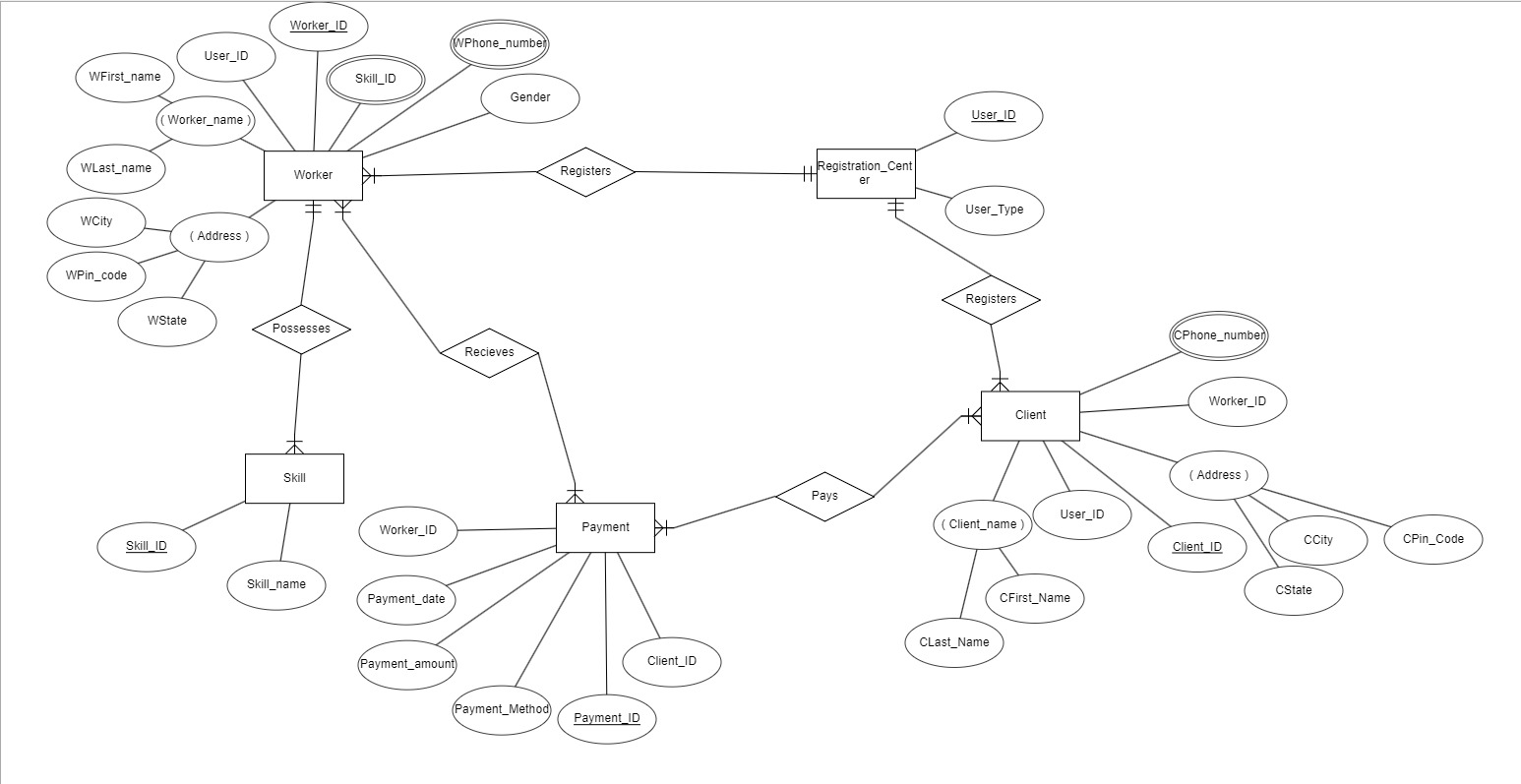
FOREIGN KEY RELATIONSHIPS:

* Registration\_Center.User\_ID and Worker.User\_ID
* Skill.Skill\_ID and Worker.Skill\_ID
* Registration\_Center.User\_ID and Client.User\_ID
* Worker.Worker\_ID and Client.Worker\_ID
* Worker.Worker\_ID and Payment.Worker\_ID
* Client.Client\_ID and Payment.Client\_ID
* Payment.Payment\_ID and Receive.Payment\_ID
* Client.Client\_ID and Pay.Client\_ID
* Payment.Payment\_ID and Pay.Payment\_ID
* Receive.Worker\_ID and Payment.Payment\_ID
* Pay.Client\_ID and Payment.Payment\_ID

07

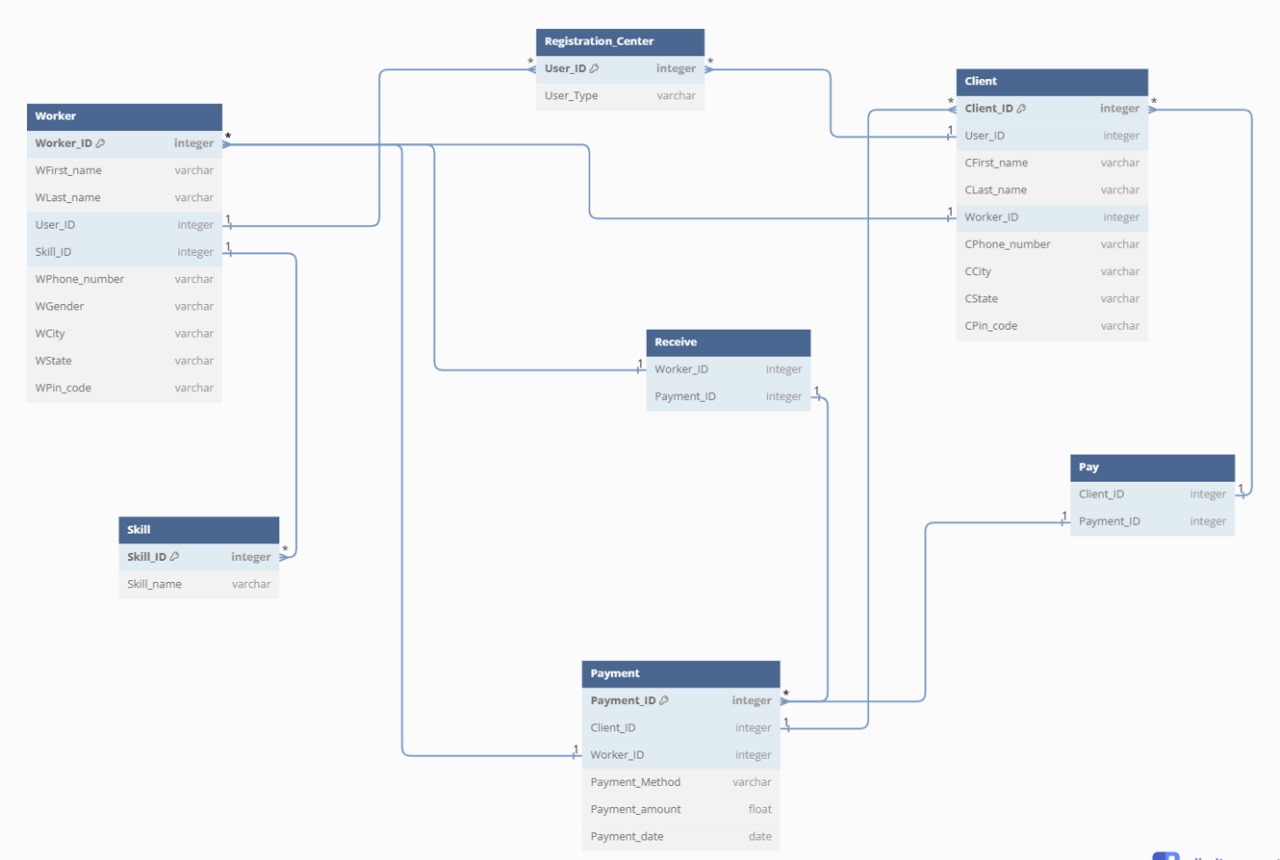
CARDINALITY:

* ONE TO MANY(Registration\_Center to Worker)
* ONE TO MANY(Registration\_Center to Client)
* MANY TO MANY(Client to Payment)
* MANY TO MANY(Worker to Payment)
* ONE TO MANY(Worker to Skill)



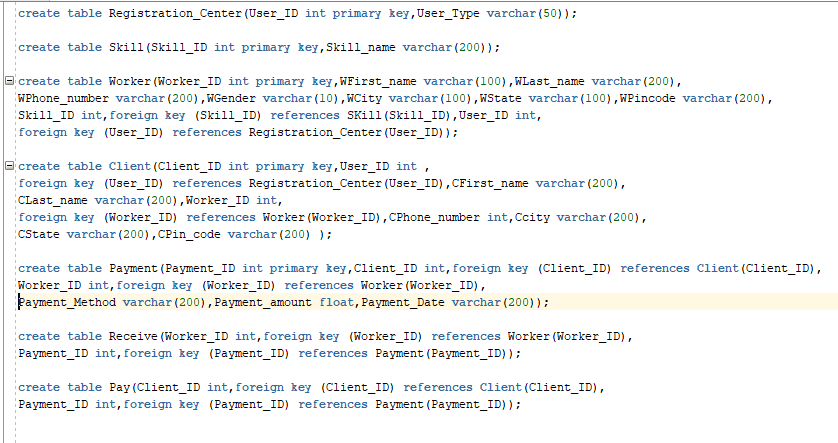
08

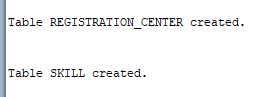
SCHEMA DIAGRAM:



09

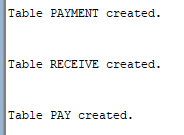
TABLES CREATION:





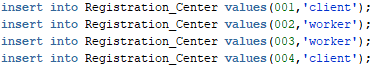






INSERTION:

Registration\_Center :



10

Skill:



Worker:



Client:



Payment:



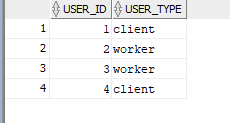
Receive:



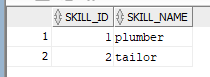
Pays:



Registration\_Center:

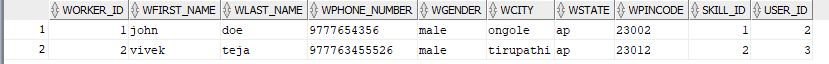


Skill:

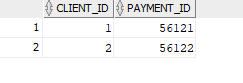


11

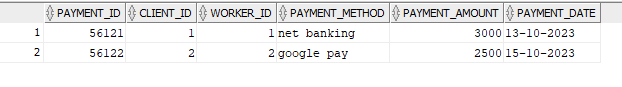
Worker:



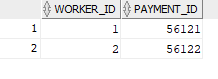
Pay:



Payment:



Receive:



Client:

