Software Requirements Specification
for
Freelancer Mediator

202203034 - Parth Sorathiya
202203065 - Paawan Vala

1 Introduction

The Freelancer Mediator platform is a database-backed system designed to connect clients and freelancers. It allows clients to post projects, specify required skills and budgets, and hire freelancers who can submit proposals. Freelancers maintain profiles and can track contracts, payments, and ratings. This project was developed for the DBMS course to demonstrate understanding of relational schema design, query formulation, and data manipulation using SQL.

2 Purpose

The purpose of this project is to design and implement a fully functional relational database system that supports the core functionality of an online freelancing platform. It aims to facilitate matching of project requirements to freelancer skills, handle contracts and payments, and provide analytical insights.

3 Scope

The system supports:

- User management for both clients and freelancers
- Skills management and assignment
- Project creation and proposal submission
- Contract and payment lifecycle management
- Ratings and feedback mechanism
- Advanced queries for analytics and reporting

4 System Working and Flow

1. A user registers and chooses to act as either a client or freelancer.
2. Clients post projects, specifying required skills and budget.
3. Freelancers update their profiles and list their skills.
4. Matching projects are shown to freelancers based on skill sets.
5. Freelancers submit proposals, which clients can accept or reject.
6. On acceptance, a contract is created and work begins.
7. After completion, payments are processed and reviews are exchanged.

5 Functional Requirements

- CRUD operations on users, freelancers, clients, projects, proposals, contracts, payments, and reviews
- ENUM-based status tracking for projects, contracts, and proposals
- Skill-based matching between projects and freelancers
- Aggregation queries for earnings and ratings
- Analytics such as top skills and freelancer leaderboards

6 User Classes and Characteristics

6.1 Clients

- Can create and manage projects
- Can view proposals and initiate contracts
- Can make payments and submit reviews

6.2 Freelancers

- Can maintain profile and skills
- Can search and bid for projects
- Can track contracts and payments
- Can receive feedback and ratings

7 User Privileges

- **Admin**: Full control of DB and user monitoring
- **Clients**: Insert projects, accept/reject proposals, create contracts
- **Freelancers**: Submit proposals, view project listings, receive contracts and payments

8 Assumptions

- Each contract results in exactly one payment
- Each user is either a client or freelancer (but can be extended)
- All projects require at least one skill
- Reviews are only submitted post contract completion

9 Queries and Project Behavior

The following SQL queries demonstrate key functionalities:

- Freelancer skill matching for project recommendations
- Age calculation from DOB using AGE() function
- Review aggregation and average rating computation
- Leaderboard generation based on monthly earnings
- Identification of most common and most earning skills
- Pending payment detection for freelancers
- Client-side project tracking, proposal status checking

These queries ensure that the system meets both transactional and analytical needs of users, providing both interaction and insight.

10 Conclusion

This project successfully models a freelancing platform from a database perspective. It leverages PostgreSQL features like ENUMs, joins, aggregates, and views to demonstrate complex real-world DBMS capabilities.