**PROJECT TITLE**

**(Font: Times New Roman, 18, Bold, Centre Alignment)**

Thesis Submitted in

partial fulfillment for the award of

**(Font: Times New Roman, 14, Centre Alignment)**

## Bachelor of Computer Application

**(Font: Times New Roman, 16, Bold, Centre Alignment)**

### Submitted by

**(Font: Times New Roman, 14, Bold, Centre Alignment)**

1. **Enrollment No.: Name**
2. **Enrollment No.: Name**
3. **Enrollment No.: Name**
4. **Enrollment No.: Name**
5. **Enrollment No.: Name**

**(Font: Times New Roman, 16, Bold, Centre Alignment)**

## Department of Computer Application Under the supervision of

**Name of Guide Designation and Department**

**(Font: Times New Roman, 14, Bold, Centre Alignment)**

****

## Silver Oak University Gota, Ahmedabad-382481,

**Gujarat, India**

**(Font: Times New Roman, 16, Bold, Centre Alignment)**

## MONTH & YEAR

**(Font: Times New Roman, 16, Bold, Centre Alignment**

# CERTIFICATE

It is certified that the project titled **[Project Name]** has been carried in group are bonafide students of BCA VI semester, Department of Computer Application, Silver Oak University, Ahmedabad, Gujarat. The project is a record of the work accomplished during the sixth semester of BCA as a partial fulfillment of the requirement for the award of degree in Bachelor of Computer Applications (BCA), under my guidance.

**Name of students Enrollment of student**

Name-1 Name-2 Name-3 Name-4

(Guide Name) Designation, Department of Computer Application, Silver Oak University, Ahmedabad, Gujarat.

# DECLARATION

We hereby declare that project entitled **“<TITLE OF PROJECT>>”** in partial fulfillment for the degree of **Bachelor of Computer Application** to **Silver Oak University**, Ahmedabad, is bonafide record of the project work carried out at **SILVER OAK COLLEGE OF COMPUTER APPLICATION** under the supervision of **<<Name of Guide>>** and that no part of any of these reports has been directly copied from any student’s report or taken from any other sources, without providing due reference.

### Name of students Sign of student

Name-1 Name-2 Name-3 Name-4

# ACKNOWLEDGEMENT

We would like to express our profound sense of deepest gratitude to our guide **<<Name of Guide>>**, Silver Oak College of Computer Application (SOCCA), Ahmedabad for valuable guidance, sympathy and co-operation for providing necessary facilities and sources during the entire period of this project.

We wish to convey our sincere gratitude to all the faculties of Department of Computer Application who have enlightened us during our studies. The facilities and cooperation received from the technical staff of department is thankfully acknowledged.

The main credit for the successful completion of this project can be given to Principal of our college Dr. Hemal Patel who supported us throughout this enterprise.

We express our thanks to all those who helped us in one way or other.

Last, but not least, we would like to thank the authors of various research articles and books that were referred to.

Name of students Enrollment of student Name-1

Name-2 Name-3 Name-4

## ABSTRACT

Fiber reinforced polymer composites, developed primarily for the aerospace and defense industries, are a class of materials with great potential to use in civil infrastructure. Since the construction of the first all-composite bridge superstructure in Miyun, China, in 1982, they have been gradually gaining acceptance from civil engineers as a new construction material.

### FIGURES/DIAGRAMS:

Pg. No

Figure 1: Flow Chart…………………………………………… Figure 2: Diagram……………………………………………….

Figure 4: System…………………………………………………

Figure 5

Figure 6

Figure 7

## TABLE OF CONTENTS

Certificate from the Guide i

Declaration from the Student ii

[Acknowledgement… iii](#_TOC_250002)

[Abstract… iv](#_TOC_250001)

List of Figures… v

[Table of Contents… vi](#_TOC_250000)

1. Introduction
   1. Project Profile . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   2. Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   3. Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
   4. Goal . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   5. Organization of the Report/Thesis . . . . . . . . . . . . . . . . . . . . . . . .
2. **Initial System Study**
   1. Chapter Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. About Organization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. Drawbacks of the existing system . . . . . . . . . . . . . . . . . . . . . . . .
   4. Problem definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   5. The proposed system . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   6. Scope of the system . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   7. Scope of this project . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   8. System development approach . . . . . . . . . . . . . . . . . . . . . . . . . .
3. **Feasibility Analysis**
   1. Section Name . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
4. **System Analysis**
   1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. Data flow diagram
   3. Entity relationship diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   4. Physical and behavioral aspects of the system . . . . . . . . . . . . . . . . .
      1. Sub section here . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
5. **Software Requirements Specifications**
   1. General Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      1. Product Perspective . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      2. Product Functions . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      3. User Characteristics . . . . . . . . . . . . . . . . . . . . . . . . . . .
      4. General Constraints . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      5. Assumptions and Dependencies . . . . . . . . . . . . . . . . . . . . .
      6. Functional Requirements . . . . . . . . . . . . . . . . . . . . . . . . .
      7. Functional Requirement . . . . . . . . . . . . . . . . . . . . . . . .
   2. External Interface Requirements . . . . . . . . . . . . . . . . . . . . . . . . .
      1. User Interfaces . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      2. Hardware Interfaces . . . . . . . . . . . . . . . . . . . . . . . . . . .
      3. Software Interfaces . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. Performance Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   4. Design Constraints . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      1. Standard Compliance . . . . . . . . . . . . . . . . . . . . . . . . . . .
      2. Hardware Constraints . . . . . . . . . . . . . . . . . . . . . . . . . . .
      3. Other Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . .
      4. Scope of this project . . . . . . . . . . . . . . . . . . . . . . . . . . .
6. **System design**
   1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. System Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. Module Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   4. Database Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   5. Input Output Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   6. Algorithm design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   7. Electronic Data Communication Design . . . . . . . . . . . . . . . . . . . . .
   8. System Maintenance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   9. Other Alternatives Considered . . . . . . . . . . . . . . . . . . . . . . . . . .
7. **System Implementation**
   1. Hardware Components . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. Software Environment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. System Development Platform . . . . . . . . . . . . . . . . . . . . . . . . . .
   4. Project Accomplishment Status . . . . . . . . . . . . . . . . . . . . . . . . . .
   5. Guidelines for Continuation . . . . . . . . . . . . . . . . . . . . . . . . . . .
   6. Hardware Components . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   7. Software Environment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
8. **System Testing**
   1. Test Plan . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. Test Cases . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
9. **Conclusion & Future Direction of Work**
   1. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. Future Direction of work . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**Bibliography/Reference**

**Chapter 1**

**Introduction**

**1.1 Project Profile**

**Project Title**: SkillSphere – A Freelance Services Platform

* SkillSphere is a web-based platform that connects clients with freelancers for a variety of projects, both technical and non-technical. The platform provides a seamless experience for posting tasks, assigning them to freelancers, tracking progress, and handling payments. The system supports two user roles: Clients and Freelancers.
  1. **Overview**
* The gig economy is growing rapidly, and many people prefer working as freelancers. However, it’s often difficult for clients to find skilled freelancers, and for freelancers to find genuine opportunities. SkillSphere aims to solve this by offering a centralized platform where clients can post their projects and freelancers can apply, get selected, and complete tasks in an organized environment. The platform also features profile ratings, secure communication, and task management tools.
  1. **Motivation**
* Our motivation stems from personal experiences and current market trends. Many of our peers and professionals struggle to find freelancing work due to a lack of proper platforms that serve regional or niche categories. Additionally, businesses also struggle to find trustworthy, skilled freelancers without going through lengthy verification processes. SkillSphere seeks to provide a trustworthy, user-friendly solution to both.
  1. **Goal**

The main goal of SkillSphere is to:

* Simplify the process of hiring freelancers
* Provide a verified environment for both clients and freelancers
* Enable smooth task tracking and management
* Promote regional and entry-level freelancers by giving them a platform to showcase their skills
  1. **Organization of the Report/Thesis**

This report is organized as follows:

* **Chapter 2** discusses the current systems and introduces the proposed solution.
* **Chapter 3** outlines the feasibility of the project.
* **Chapter 4** presents system analysis with diagrams.
* **Chapter 5** provides software requirement specifications.
* **Chapter 6** focuses on system design, architecture, and modules.
* **Chapter 7** explains implementation details.
* **Chapter 8** includes system testing and validation.
* **Chapter 9** concludes the report and describes possible future enhancements.

**Chapter 2**

**Initial System Study**

**2.1 Chapter Introduction**

* This chapter highlights the background of the project, studies the drawbacks of the existing solutions, and introduces our proposed system—SkillSphere. It also defines the problem we aim to solve and the scope and methodology adopted for development.

**2.2 About Organization**

* SkillSphere acts as a virtual freelance marketplace that connects clients with skilled freelancers. It is designed to streamline project collaboration by allowing clients to post tasks and freelancers to apply, complete, and deliver work through an organized platform.

The system replicates the structure of a freelancing service organization, focusing on efficiency, transparency, and accessibility for both parties.

**2.3 Drawbacks of the Existing System**

Most existing freelancing platforms like Fiverr, Upwork, and Freelancer have the following drawbacks:

* High service fees for both freelancers and clients.
* Complex UI, especially for new users.
* Overcrowded marketplace, making it hard for new freelancers to stand out.
* Limited regional opportunities and language support.
* Lack of transparency in communication and project status.

**2.4 Problem Definition**

* Clients often struggle to find reliable freelancers and track their task progress. Freelancers, especially beginners, face difficulty in getting hired and maintaining trust. Our system aims to resolve these issues by providing a platform with simplified task posting, two-way communication, and a transparent workflow between clients and freelancers.

**2.5 The Proposed System**

SkillSphere is a freelancing platform where:

* Clients can register and post jobs under different categories.
* Freelancers can register, browse posted jobs, and apply for relevant tasks.
* Clients can assign tasks, communicate, and review submissions.
* Admins oversee the platform to ensure smooth functioning.

**2.6 Scope of the System**

The platform is designed for web use and focuses on:

* Individual freelancers and small businesses
* Regional and entry-level job listings
* General and tech-specific tasks (e.g., logo design, content writing, app development)

**2.7 Scope of This Project**

This version of SkillSphere includes:

* Role-based login for Clients and Freelancers
* Task posting, bidding, assignment, and completion flow
* Admin panel for user and task management
* Email-based communication notifications

**2.8 System Development Approach**

We followed an **Iterative Development Model** using Agile methodology:

* Planning and requirement gathering
* UI/UX designing in Figma
* Frontend development using React.js
* Backend development using Node.js and MongoDB
* Testing and bug fixes after every sprint

**Chapter 3**

**Feasibility Analysis**

**3.1 Chapter Introduction**

Before proceeding with development, it is essential to analyze the feasibility of the project from multiple perspectives. This ensures that the system can be successfully built, deployed, and maintained within available resources and constraints.

**3.2 Technical Feasibility**

SkillSphere is technically feasible as it is built using widely adopted technologies such as:

* **Frontend**: React.js
* **Backend**: Node.js with Express
* **Database**: MongoDB
* **Hosting**: Cloud platforms like Vercel or Render

These technologies are scalable, well-documented, and supported by large communities, making them suitable for this platform.

**3.3 Operational Feasibility**

The system is simple and user-friendly, designed for users with basic computer and internet skills. The role-based design (Client and Freelancer) ensures ease of use and smooth operation. Admin-level monitoring adds additional control and ensures effective platform management.

**3.4 Economic Feasibility**

As a student project, SkillSphere was developed using free and open-source tools, minimizing development costs. In a real-world deployment, monetization could be achieved via:

* Service fees or commissions
* Premium subscriptions for additional features

The return on investment (ROI) potential is high, especially if targeted toward niche or regional markets.

**3.5 Schedule Feasibility**

The project was developed within the academic semester timeline. Proper planning, time allocation, and the use of agile methodology helped ensure that key milestones were completed on time without overburdening the team.

**Chapter 4**

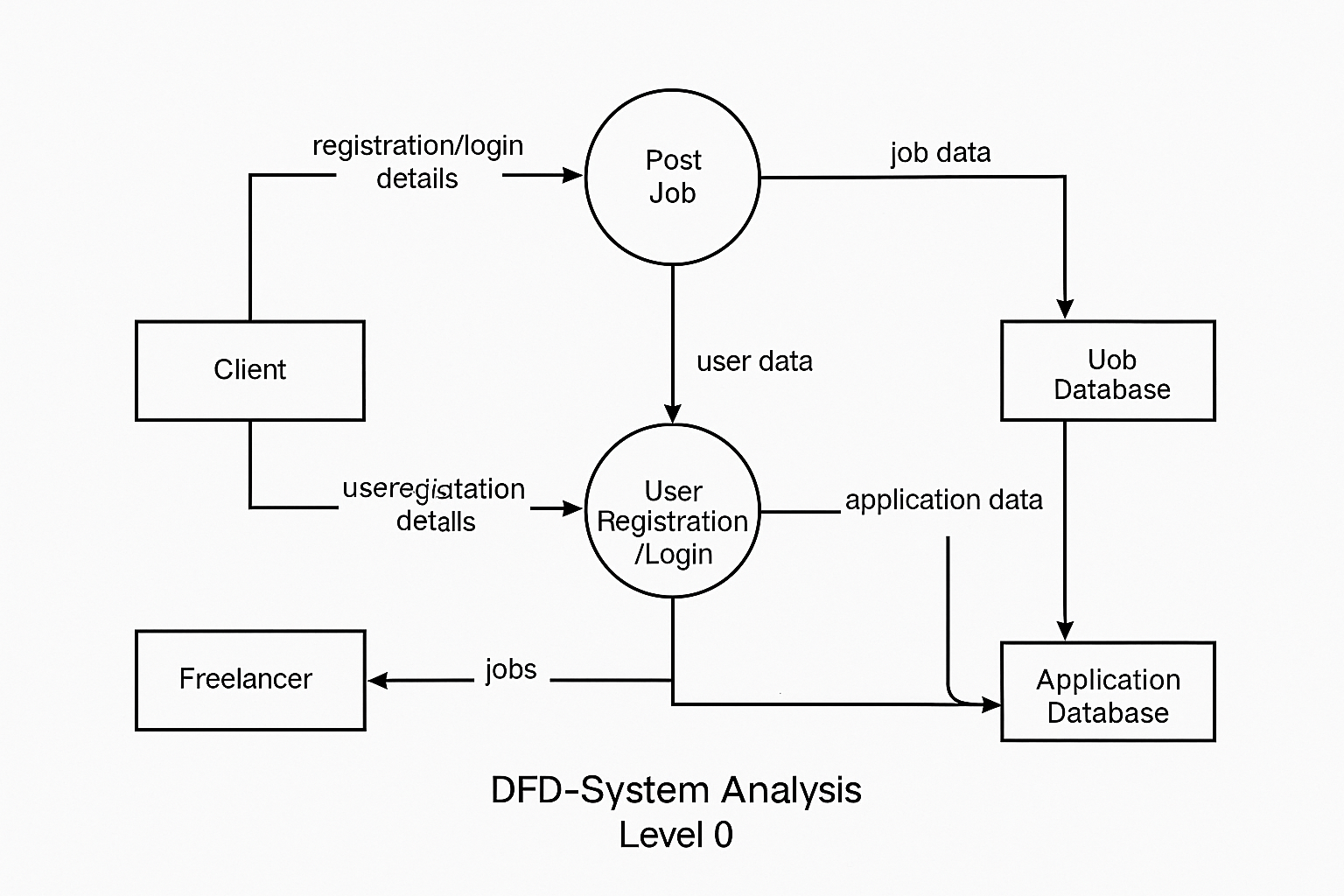
**System Analysis**

**4.1 Introduction**

System analysis involves understanding and modeling how SkillSphere functions in terms of data movement, system components, and user interaction. It helps identify what the system should do, what entities are involved, and how they interact.

**4.2 Data Flow Diagram (DFD)**

The Data Flow Diagram (DFD) represents the flow of data within the SkillSphere system. It visualizes how users interact with the system and how data is processed and stored.

****

**4.2.1 Explanation of Level 1 DFD:**

1. **User Registration/Login:**
   * **Both Clients and Freelancers interact with the system to register or log in.**
   * **The system checks user credentials and stores/retrieves data from the User Database.**
2. **Post Job:**
   * **A Client can post a job by entering job details such as title, description, budget, and deadline.**
   * **The job information is stored in the Job Database.**
3. **Apply for Job:**
   * **A Freelancer views available jobs and applies by submitting a proposal.**
   * **The system stores application details in the Application Database, linked to both the freelancer and the job.**

**External Entities:**

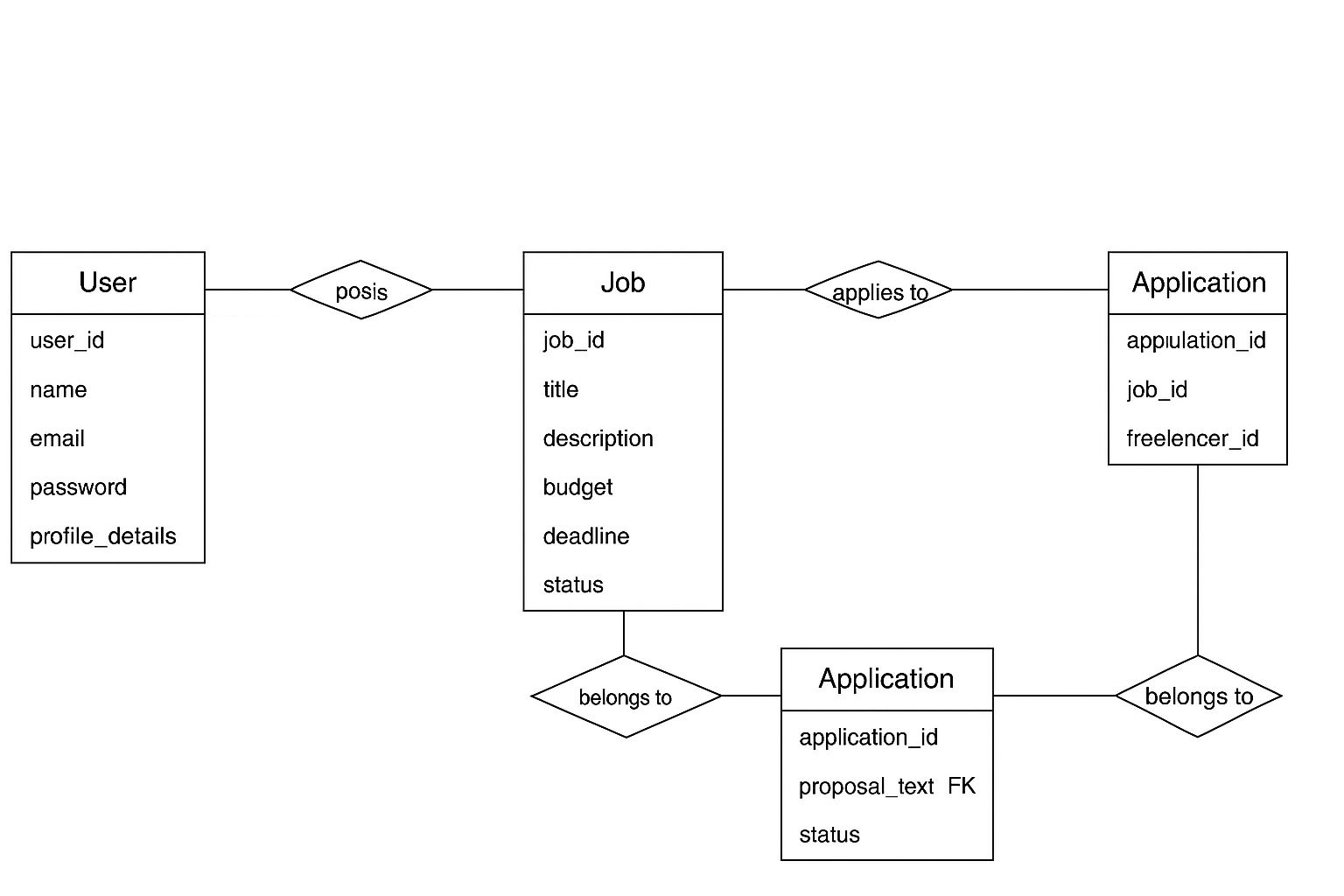
* **Client and Freelancer are the primary external entities that interact with the system.**

**Data Stores:**

* **User Database: Stores all registered user details.**
* **Job Database: Stores information about posted jobs.**
* **Application Database: Stores job applications made by freelancers.**

**4.3 Entity Relationship Diagram (ERD)**

The Entity Relationship Diagram (ERD) of SkillSphere represents the logical structure of the database system and shows how different entities in the system are related to each other.



**4.3.1 Explanation of the ERD:**

1. **User Entity:**
   * This entity stores information about all users of the system, including both **Clients** and **Freelancers**.
   * Attributes include: user\_id, name, email, password, role, and profile\_details.
   * The role attribute helps distinguish between Clients and Freelancers.
2. **Job Entity:**
   * This entity stores all job postings created by Clients.
   * Attributes include: job\_id, title, description, budget, deadline, status, and posted\_by (foreign key referencing the User table).
   * A **User** with the Client role can post multiple jobs.
3. **Application Entity:**
   * This table tracks applications submitted by Freelancers for jobs.
   * Attributes include: application\_id, job\_id, freelancer\_id, proposal\_text, and status.
   * job\_id and freelancer\_id are foreign keys referencing the Job and User tables respectively.
   * One Freelancer can apply to multiple jobs, and one job can have many applications.
4. **Category Entity:**
   * Represents different categories (like Web Development, Graphic Design, Writing, etc.) that a job can belong to.
   * Attributes: category\_id and name.
5. **JobCategory Entity (Many-to-Many Relationship Table):**
   * Since a job can belong to multiple categories and a category can include multiple jobs, this linking table connects Jobs and Categories.
   * Attributes: job\_id and category\_id (both foreign keys).

**Relationships Summary:**

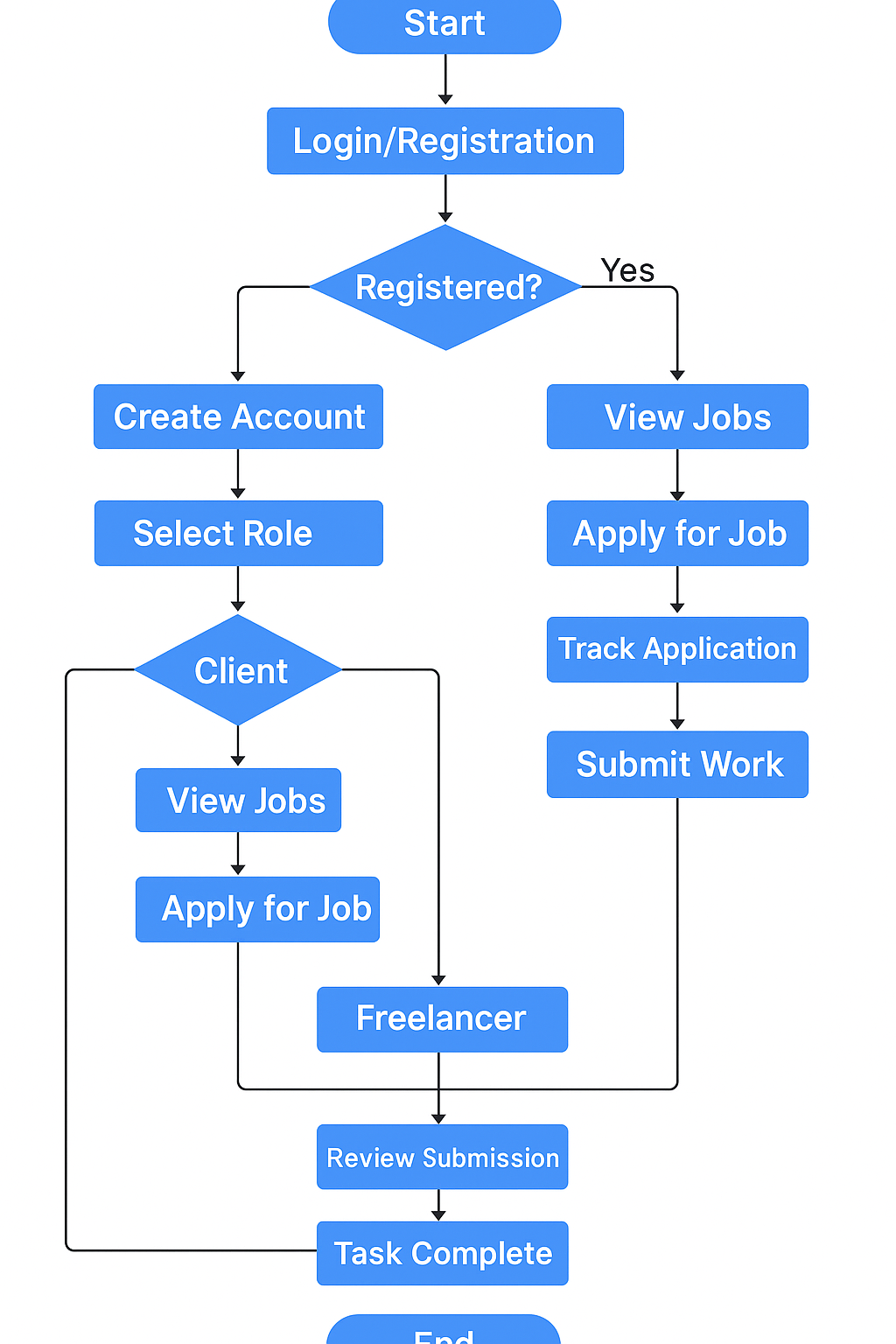
* One **Client** can post many **Jobs**.
* One **Freelancer** can apply to many **Jobs**, and one **Job** can receive many **Applications**.
* One **Job** can be linked to multiple **Categories** via the JobCategory table.

**4.4 Flow Chart**

A flow chart represents the sequence of steps in the system workflow. It visually shows the decision points and the flow of control based on different user actions in the platform. The SkillSphere platform supports two major roles: **Client** and **Freelancer**.

**4.4.1 Explanation of the Flow Chart:**

1. **Start:**
   * The user opens the platform.
2. **Login/Registration:**
   * User chooses to either register or log in.
   * If not registered, they create a new account and select a role (Client or Freelancer).
   * If already registered, login is validated.



1. **Role-Based Navigation:**
   * If **Client**:
     + Can post a job
     + View applications from freelancers
     + Assign job to selected freelancer
   * If **Freelancer**:
     + Can view job listings
     + Apply for jobs
     + Track application status
2. **Task Execution:**
   * Once a job is assigned, the freelancer works on it and submits the work.
3. **Job Completion & Review:**
   * Client reviews the submission.
   * Once approved, the task is marked as complete.
4. **End**

**4.5 Physical and Behavioral Aspects of the System**

**4.5.1 Physical Aspects**

The physical aspects describe the hardware and software environment where the SkillSphere system operates.

**Hardware Requirements:**

* Processor: Intel i3 or higher
* RAM: Minimum 4 GB
* Storage: Minimum 100 MB of free disk space
* Display: Minimum 1024×768 resolution

**Software Requirements:**

* Operating System: Windows/Linux/macOS
* Browser: Chrome, Firefox, or Edge (latest versions)
* Backend: Node.js / Express.js
* Frontend: React.js
* Database: MongoDB
* Other Tools: Git, VS Code

**4.5.2 Behavioral Aspects**

The behavioral aspects focus on how users interact with the system and how the system reacts.

* The system distinguishes users based on their role: **Client** or **Freelancer**.
* Clients can post jobs and view applicants.
* Freelancers can view available jobs and apply to them.
* After a freelancer is selected, the system tracks job status and completion.
* Users receive feedback and updates at each step (e.g., success messages, error alerts).
* Simple and intuitive interface ensures a smooth user experience.

**Chapter 5**

**Software Requirements Specification (SRS)**

The **Software Requirements Specification (SRS)** provides a complete description of the behavior, functionalities, and constraints of the SkillSphere system. It acts as a blueprint for developers and stakeholders.

**5.1 Introduction**

**Purpose:**

The purpose of this document is to define the functional and non-functional requirements of the SkillSphere freelance service platform. It serves as a guide for the design, development, and deployment phases.

**Scope:**

SkillSphere is a freelance marketplace platform that connects **Clients** who need work done with **Freelancers** who offer various services. It enables clients to post jobs and freelancers to apply. The system manages users, job listings, and applications in a streamlined and secure way.

**Definitions, Acronyms, and Abbreviations**

* **Client** – A user who posts jobs
* **Freelancer** – A user who applies for jobs
* **SRS** – Software Requirements Specification
* **UI** – User Interface
* **DB** – Database

**5.2 Overall Description**

**Product Perspective**

SkillSphere is a standalone web-based application with a frontend built in **React.js** and backend APIs powered by **Node.js**. It interacts with a **MongoDB** database for data persistence.

**Product Functions**

* User registration and authentication
* Role-based access (Client/Freelancer)
* Job posting and management
* Job browsing and application
* Application tracking and status updates
* Notification and feedback system

**User Characteristics**

* Basic understanding of using web applications
* Clients should be able to describe their requirements clearly
* Freelancers should be able to create convincing proposals

**Assumptions and Dependencies**

* Stable internet connection is available
* Users access the system through modern web browsers
* Data security and privacy policies are followed

**5.3 Specific Requirements**

**5.3.1 Functional Requirements**

1. The system shall allow new users to register with email and password.
2. The system shall authenticate users on login.
3. The system shall allow Clients to post new jobs.
4. The system shall allow Freelancers to browse and apply to jobs.
5. The system shall let Clients view and manage applications.
6. The system shall allow status updates for job applications (e.g., accepted, rejected).

**5.3.2 Non-Functional Requirements**

1. The system shall respond within 3 seconds for 95% of the operations.
2. The system shall be available 24/7 with 99% uptime.
3. The system shall protect user data using secure authentication and encrypted storage.
4. The system shall have a responsive UI compatible with all major devices and browsers.