Software Requirements Specification

Functional Requirements

1. User Registration & Authentication

The system shall allow users to register, log in, log out, and reset passwords as:

- Creator / Gig Worker
- Client
- Organization

2. Creator/Gig Profile Management

The system shall allow creators to:

- Create and update professional profiles.
- Add skills, tools, experience, and portfolio.
- Create and publish gig offerings describing services they provide.

3. Client – Gig Discovery

The system shall allow clients to search and browse gig offerings published by creators, including the ability to:

- View all available gig posts.
- Filter gigs by category, skills, pricing, and delivery time.
- View detailed gig information and creator profiles.

4. Task Posting

The system shall allow organizations to:

- Create tasks with title, description, budget, deadline, and required skills.
- Edit and delete tasks.

5. Task Discovery & Application

The system shall allow creators to discover and apply for tasks, including the ability to:

- Browse all open tasks.
- Search and filter tasks by skill, category, and budget.
- Apply to tasks with a proposal message.

6. Creator Selection & Assignment

The system shall allow clients/organizations to:

- View applications.
- Accept or reject proposals.
- Assign a creator to a task.

7. In App Communication

The system shall allow:

- Real-time messaging between clients and creators.
- File sharing within tasks.

8. Ratings & Reviews

The system shall allow:

- Mutual ratings after task completion.
- Public display of reputation on profiles.

Non-Functional Requirements

1. Performance Requirements

- The system shall respond to user requests (page loads, searches, contract viewing) within **3 seconds** under normal conditions.
- The system shall support processing of at least **200 gig searches per minute** without delay.
- The system shall handle simultaneous gig posting and proposal submissions without affecting response time.

2. Reliability Requirements

- The system shall maintain an uptime of **99% or higher** during operational hours.
- The system shall ensure that contract and payment **data are not lost** in the event of system failure.
- The system shall automatically **retry failed transactions** related to work submissions and payments.

3. Scalability Requirements

- The system shall support up to **5,000 concurrent users** without performance degradation.
- The platform shall allow future addition of more **gig categories and users** without system redesign.

4. Availability Requirements

- The system shall be **available 24/7**, except during scheduled maintenance periods.
- Scheduled **maintenance** downtime shall not exceed **2 hours per month**.

5. Usability Requirements

- **New users** shall be able to complete the **gig posting or service requesting process** within **5 minutes**.
- The system shall provide **clear error message**s when invalid data is entered.
- The user interface shall be responsive and usable on **mobile and desktop browsers.**

6. Security Requirements

- The system shall use **multi-factor authentication** for all user logins.
- The system shall implement **role-based authorization** for Clients, Gig Workers, and Managers.
- All payment-related data shall be transmitted using **HTTPS encryption.**

7. Maintainability Requirements

- The system shall be developed using a **modular architecture** to allow future updates.
- Code shall follow **standard coding** practices and documentation.
- **System logs shall be maintained** to help in debugging and issue tracking.

8. Efficiency Requirements

- The system shall **not exceed 70% CPU utilization** during peak usage.
- Database operations such as **gig search and profile retrieval shall be optimized** to avoid resource wastage.

9. Portability Requirements

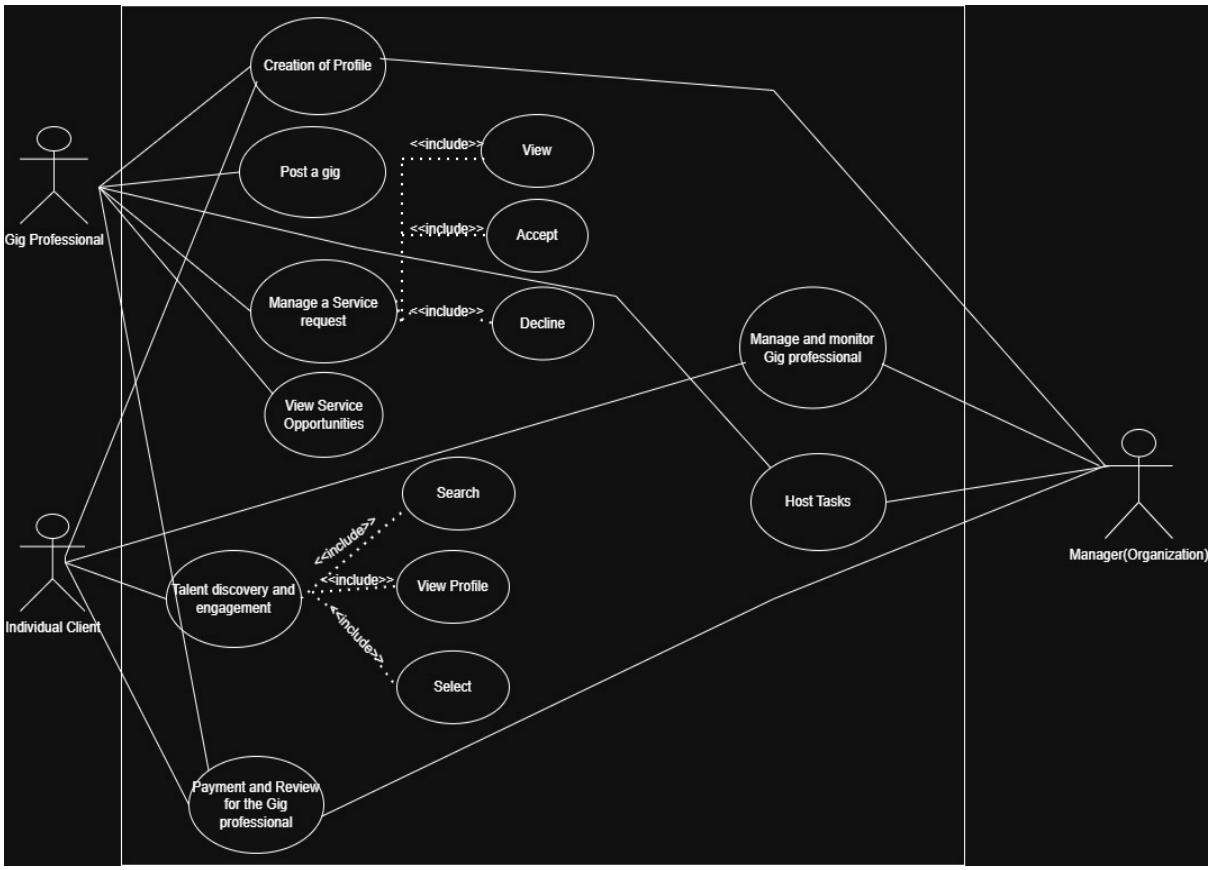
- The application shall run on **Windows, macOS, and Linux servers**.
- The web platform shall function correctly on **major browsers (Chrome, Firefox, Edge, Safari)**.

10. Reusability Requirements

- The user **authentication** module shall be **reusable** for future web modules.
- The **payment processing** component shall be designed for reuse in other transaction-based features.

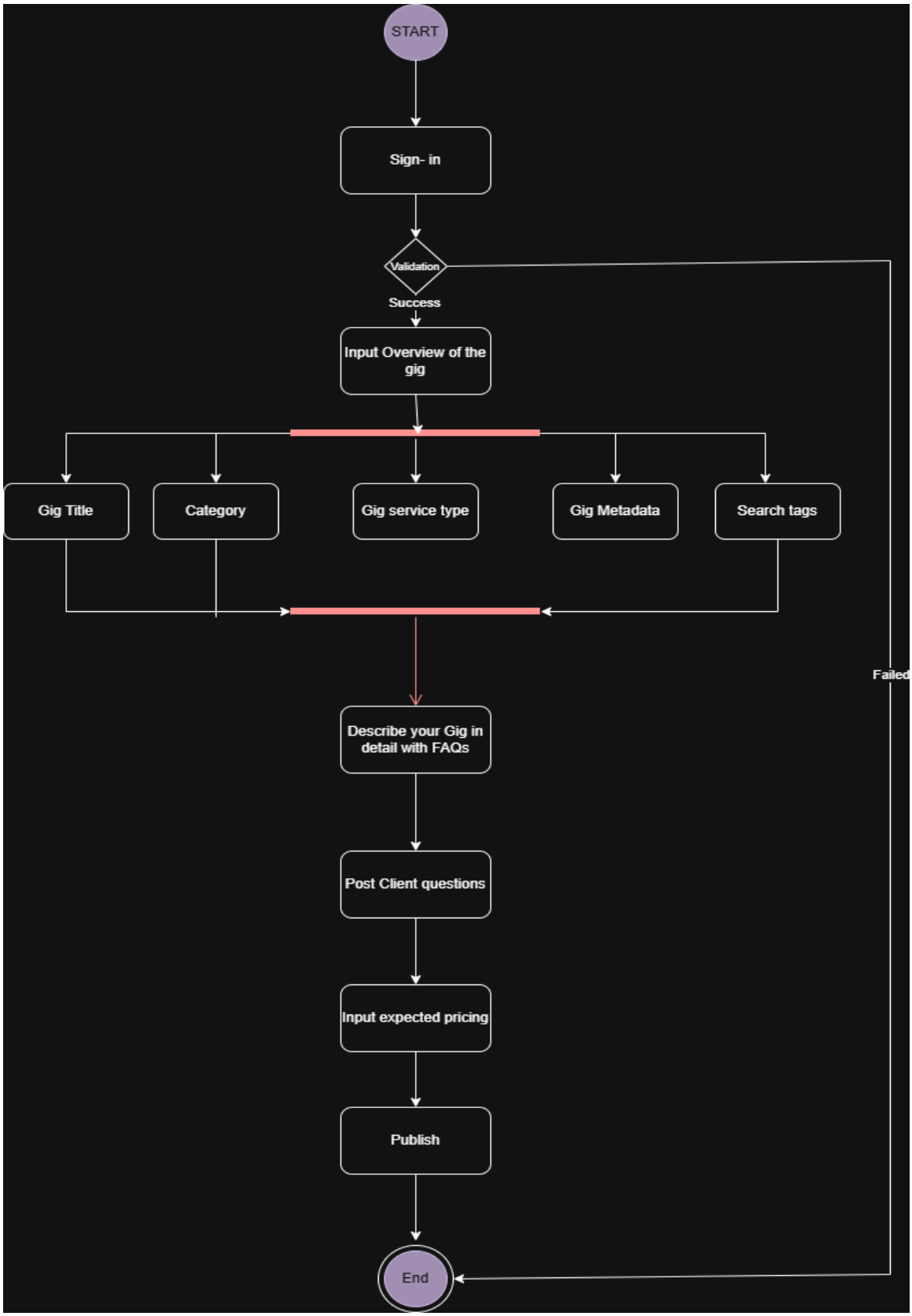
UML Diagrams

1. Use Case Diagram

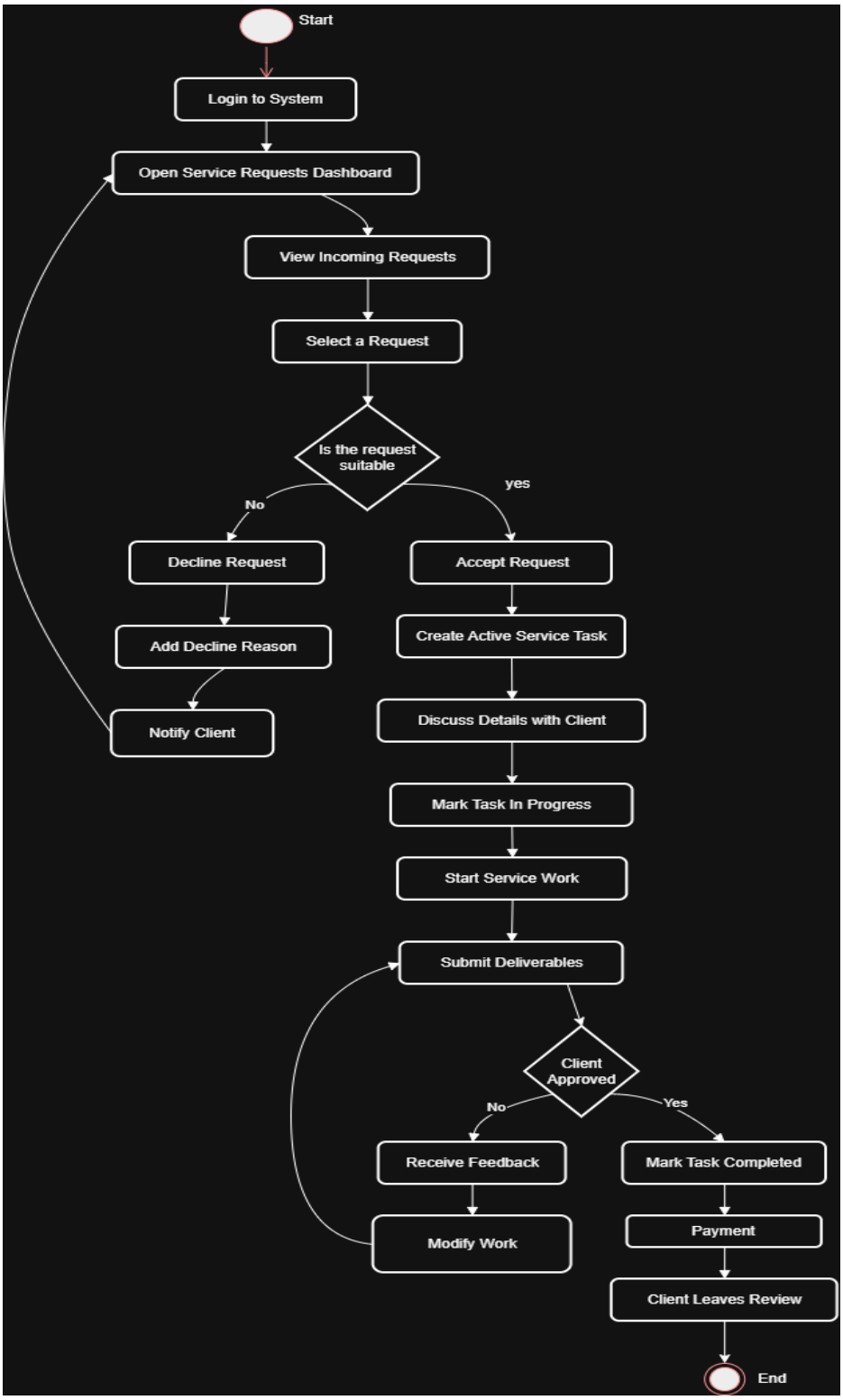


2. Activity Diagrams

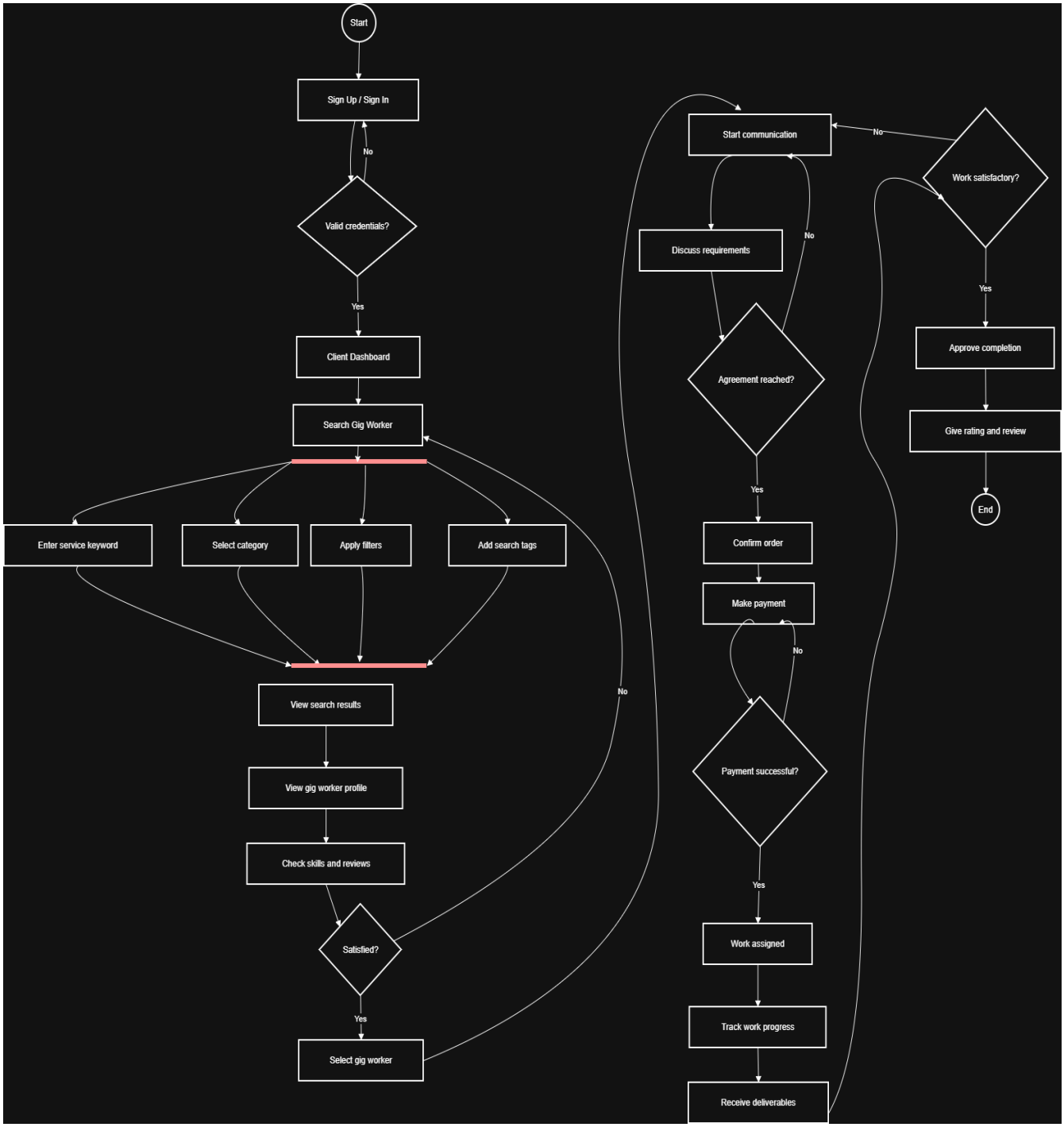
- **Post a gig**



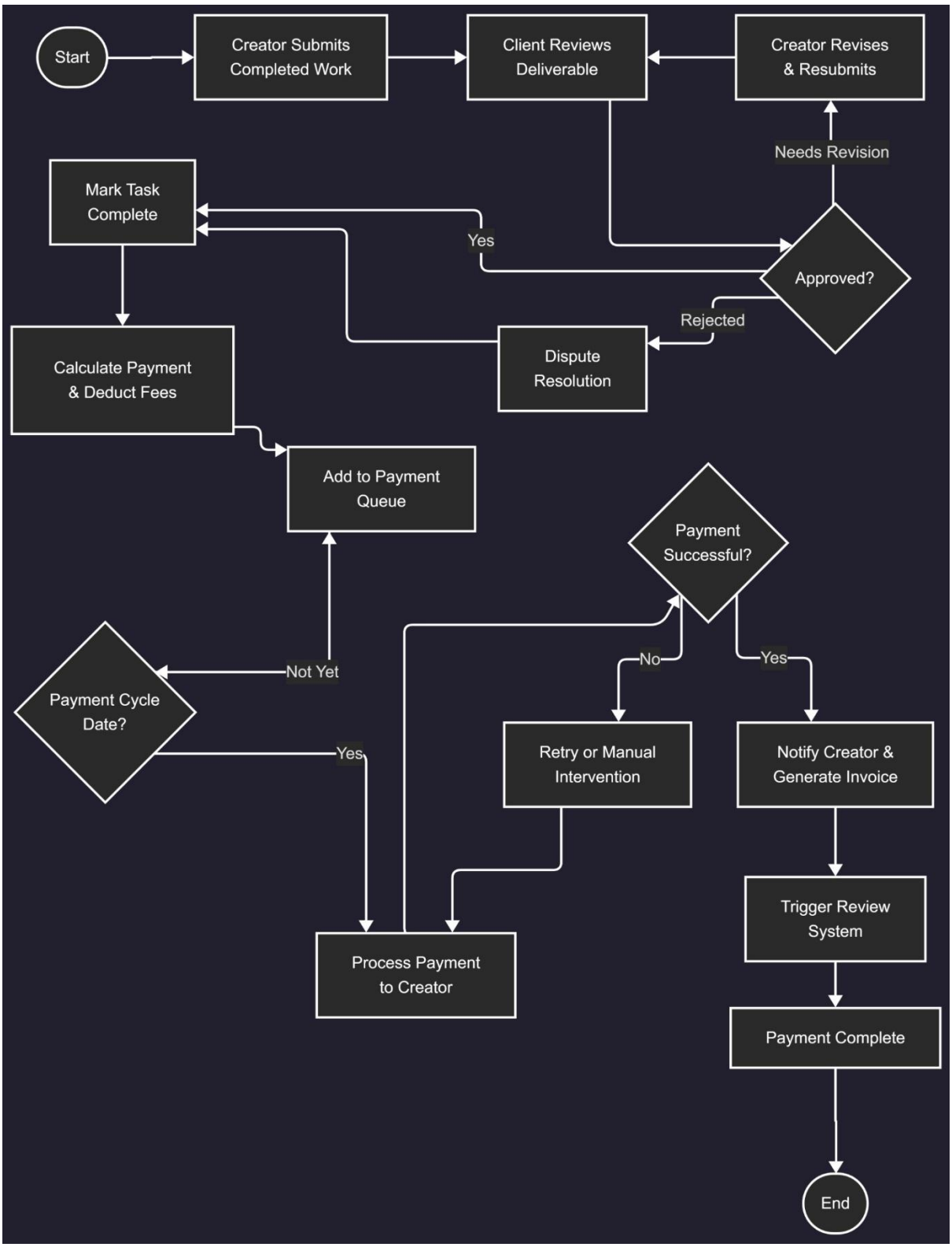
- Manage a service request



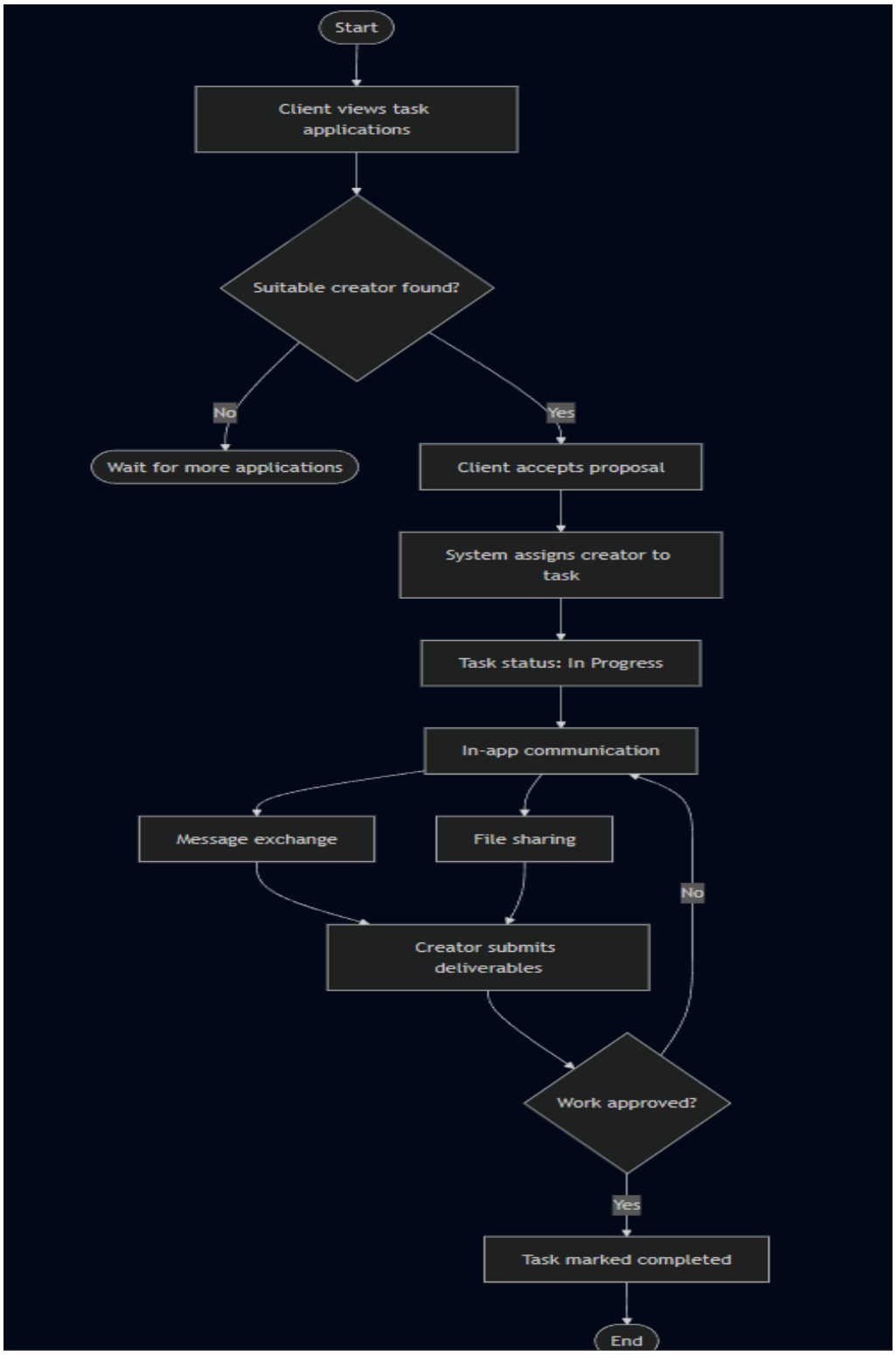
- Talent discovery and engagement



- Payment and Review the gig professional



- Monitor and manage the gig professional



Sequence Diagram