Software Requirements Specification (SRS)

Domain: Creator and gig economy platforms

Problem Statement: Task, Project & Engagement Management System

Team Members:

Tejas Nama - S20240010159

Rehan Shaik - S20240010221

T Neel - S20240010232

Mokshagna Manduva - S20240010138

Nenavath Jaswanth - S20240010161

Version: 1.0

Chapter 1: Preface

1.1 Expected Readership

This document is intended for the following stakeholders:

- **System Architects and Developers:** To understand the functional logic, data structures, and architectural constraints required for implementation.
- **Project Managers:** To track development progress against defined requirements and milestones.
- **Quality Assurance (QA) Teams:** To design test cases for functional verification, specifically focusing on the escrow and expert review workflows.
- **Client/Stakeholders:** To verify that the proposed solution meets the business needs regarding task management and quality assurance.

1.2 Version History

Version	Date	Description of Changes
1.0	3-2-2026	Initial release of the SRS. detailed definition of Client, Gig Worker , and Expert Reviewer roles, including the Technical Audit workflow and Escrow mechanics.

Chapter 2: Introduction

2.1 Need for the System

The current gig economy landscape lacks robust quality assurance mechanisms for technical deliverables. Clients often lack the technical expertise to evaluate code quality, leading to the acceptance of sub-par work. Furthermore, existing platforms often separate "hiring" from "project management." This system integrates hiring, detailed task management , and a unique Expert Reviewer ecosystem to mitigate risk and ensure high-quality deliverables.

2.2 System Functions

The system provides a unified platform for task execution and engagement management. Key functions include:

- **Role-Specific Onboarding:** Providing separate account creation paths for Clients and Gig Workers with role-specific features.
- **Task & Project Management:** Creation of tasks with milestones, sub-tasks, and progress tracking.
- **Escrow System:** Holding points/funds securely until milestone approval.
- **Expert Review (Conflict Resolution):** Mediation by Expert Reviewer during disputes.
- **Expert Review (Technical Audit):** Proactive code/quality review by Expert Reviewer before client acceptance.

2.3 Business Context

The system aligns with the strategic objective of creating a high-trust gig marketplace. It serves as an intermediary platform, integrating with external payment gateways and email notification services.

Chapter 3: Glossary

Term	Definition
Client	A user (Brand/Organization) who posts tasks, funds the escrow, and hires Gig Workers.
Gig Worker	A freelancer or gig worker who bids on tasks and submits deliverables via milestones.
Expert Reviewer	An internal platform employee or designated specialist responsible for conducting technical audits on deliverables and providing binding resolutions during disputes.
Escrow	A financial holding area where points are locked upon contract initiation and released only upon approval.
Milestone	A distinct sub-segment of a larger task with its own deliverable and assigned point value.
Technical Audit	An optional service where an Expert Reviewer evaluates work for hidden issues (e.g., security flaws) before Client acceptance.
Workroom	The collaborative space for messaging and file sharing between the Client and Gig Worker.

Chapter 4: User Requirements Definition

4.1 User Services

- **The Client** shall be able to post tasks, fund the escrow, and optionally select "Technical Audit" to ensure code quality.
- **The Gig Worker** shall be able to browse tasks, submit proposals, and manage milestones for payment.

- **The Expert Reviewer** shall be able to access flagged tasks to provide binding resolutions for conflicts or technical audits.

4.2 Non-functional Requirements

- **Usability:** The system shall clearly distinguish between Client and Worker accounts during the sign-up process.
- **Security:** The platform shall ensure all personal data and financial transactions are kept private and secure.
- **Reliability:** The escrow system shall ensure funds are never released without authorized approval or dispute resolution.
- **Performance:** The system shall provide fast loading times for task dashboards and ensure chat messages are delivered in real-time.

Chapter 5: System Requirements Specification

5.1 Functional Requirements

This section defines the specific behaviours and functions the system must perform to meet user needs.

5.1.1 Authentication & Onboarding

- The system shall allow users to register and verify their identity using email or social login.
- The system shall restrict users to a single role (Client or Gig Worker) per account upon registration.

5.1.2 Task Management & Hiring

- The Client shall be able to create a task that includes a Title, Description, Category, Budget (Points), and an option for "Technical Audit."
- The Client shall be able to divide a task into smaller Milestones, assigning a portion of the total budget to each.
- The system shall allow Clients to search for Gig Worker based on skills and invite them to apply.
- The Gig Worker shall be able to initiate a collaboration request for an open project by submitting a proposal.

5.1.3 Milestones and Workroom

- The system shall provide a visual board (Kanban or List) to track Milestone status (To Do, In Progress, Submitted, Done).
- The system shall allow Gig Worker to upload files and deliverables directly to a Milestone.
- The system shall provide a secure chat feature for communication between the Client, Gig Worker, and Expert Reviewer.

5.1.4 Escrow & Financial Logic

- The system shall deduct points from the Client's wallet and lock them in a secure "Escrow" holding area before work begins.
- The system shall automatically transfer points from Escrow to the Gig Worker immediately after the Client clicks "Approve."

- The system shall prevent funds from being released if a task is flagged for Conflict or fails an Audit.

5.1.5 Expert Review (Conflict & Audit)

- The system shall allow either the Client or Gig Worker to "Flag Conflict" if they disagree on the work, which notifies an Expert Reviewer.
- If the Client selected "Technical Audit," the system shall route the Gig Worker's work to an Expert Reviewer before the Client sees it.
- The system shall process the Expert's report: a "Pass" sends the work to the Client, while a "Fail" sends it back to the Gig Worker for fixes.
- The system shall give the Expert Reviewer temporary access to view the project code and chat history to perform their review.

5.2 Non-Functional Requirements

This section defines the quality standards and constraints of the system.

5.2.1 Usability

- The system shall provide **role-specific dashboards** that immediately present relevant workflows (e.g., "Post Task" for Clients vs. "Find Work" for Workers) upon login, without requiring manual configuration.
- The system shall be responsive and function correctly on both desktop and mobile web browsers.

5.2.2 Performance

- The system shall load key pages, such as the Task Dashboard, in under 3 seconds.
- Chat messages and status updates shall appear to users in real-time (within 1 second).

5.2.3 Security

- All sensitive user data (passwords, personal ID) and financial records shall be encrypted to prevent unauthorized access.
- The system shall ensure that Expert Reviewers can only see the specific projects they are assigned to review, not the entire database.

5.2.4 Reliability

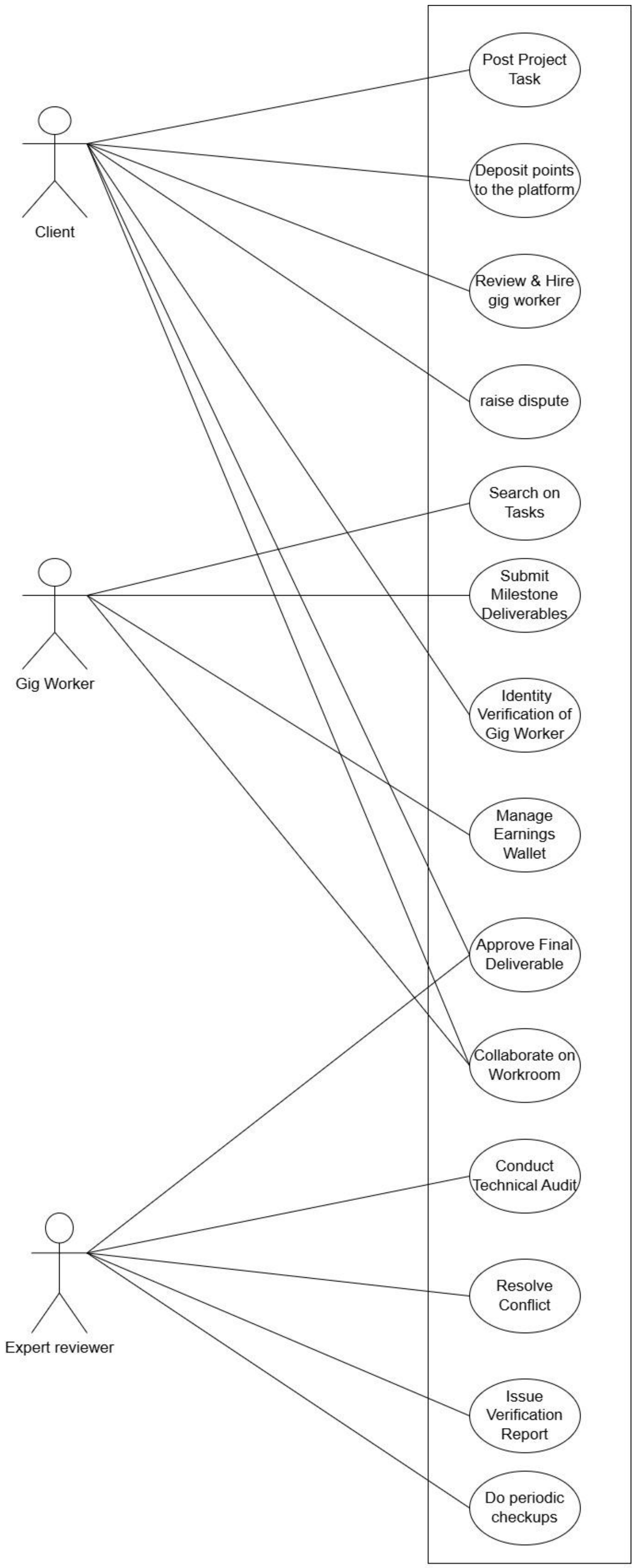
- The financial system must be 100% accurate; points must never be lost during a transfer between Client, Escrow, and Gig Worker.
- The system shall aim for high availability, ensuring the platform is accessible to users 99.9% of the time.

Chapter 6: System Models

This Section contains

- Use case diagram
- Activity diagrams
- Sequence diagram

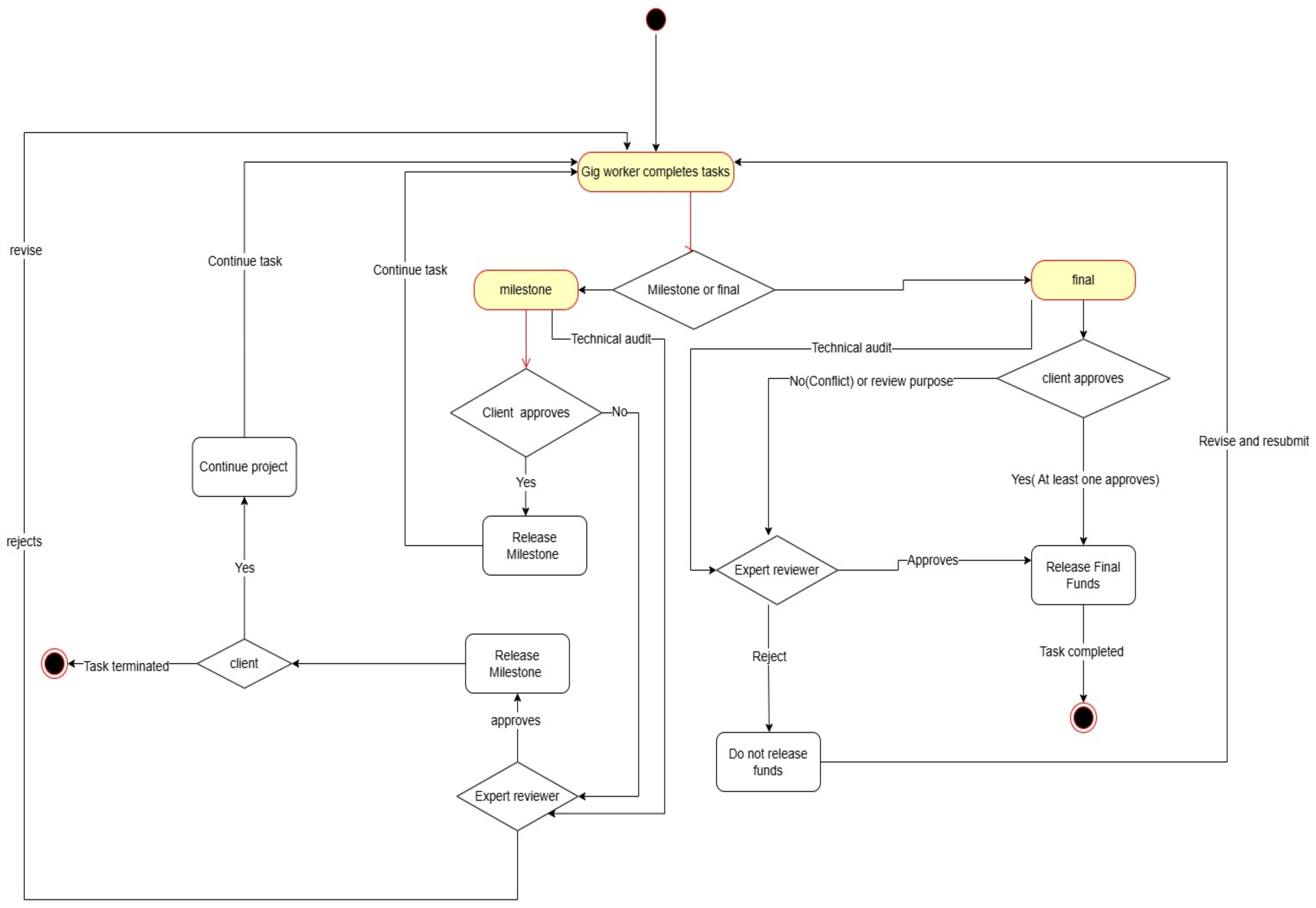
USE CASE DIAGRAM


Client
Gig Worker
Expert reviewer
Post Project Task
Deposit points to the platform
Review & Hire gig worker
raise dispute
Search on Tasks
Submit Milestone Deliverables
Identity Verification of Gig Worker
Manage Earnings Wallet
Approve Final Deliverable
Collaborate on Workroom
Conduct Technical Audit
Resolve Conflict
Issue Verification Report
Do periodic checkups

Activity Diagrams

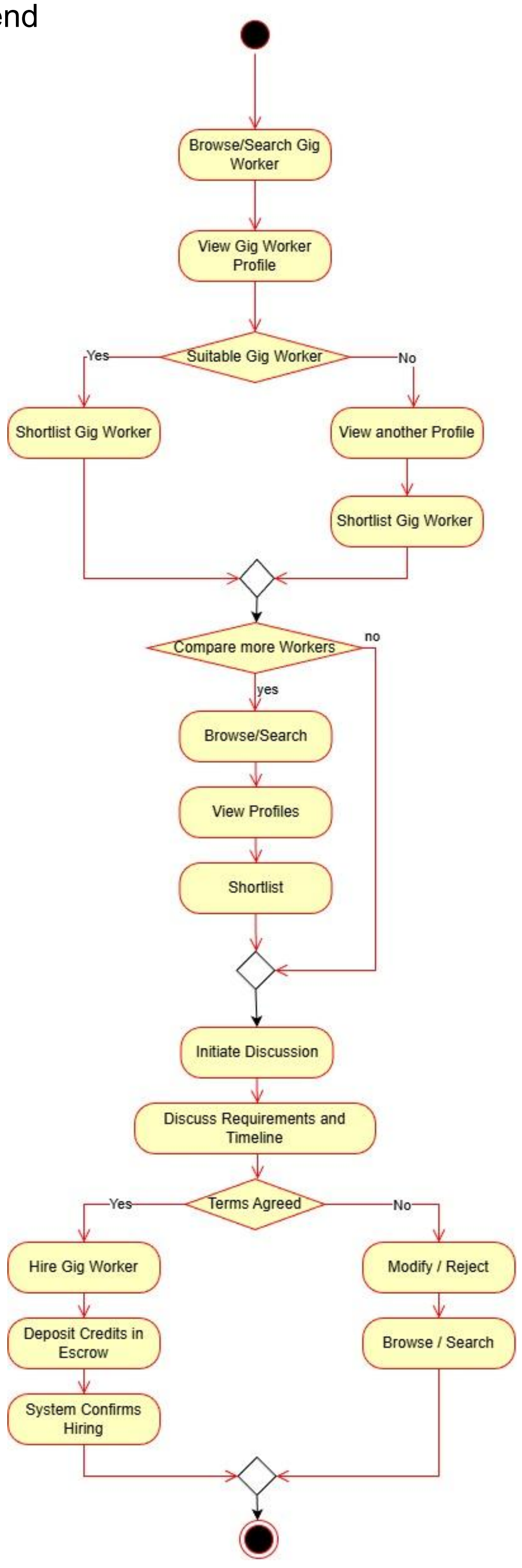
Approve Final Deliverables, Submit Funds

This activity diagram illustrates the iterative lifecycle of project deliverables, detailing the logic for both interim milestones and the final project handover.



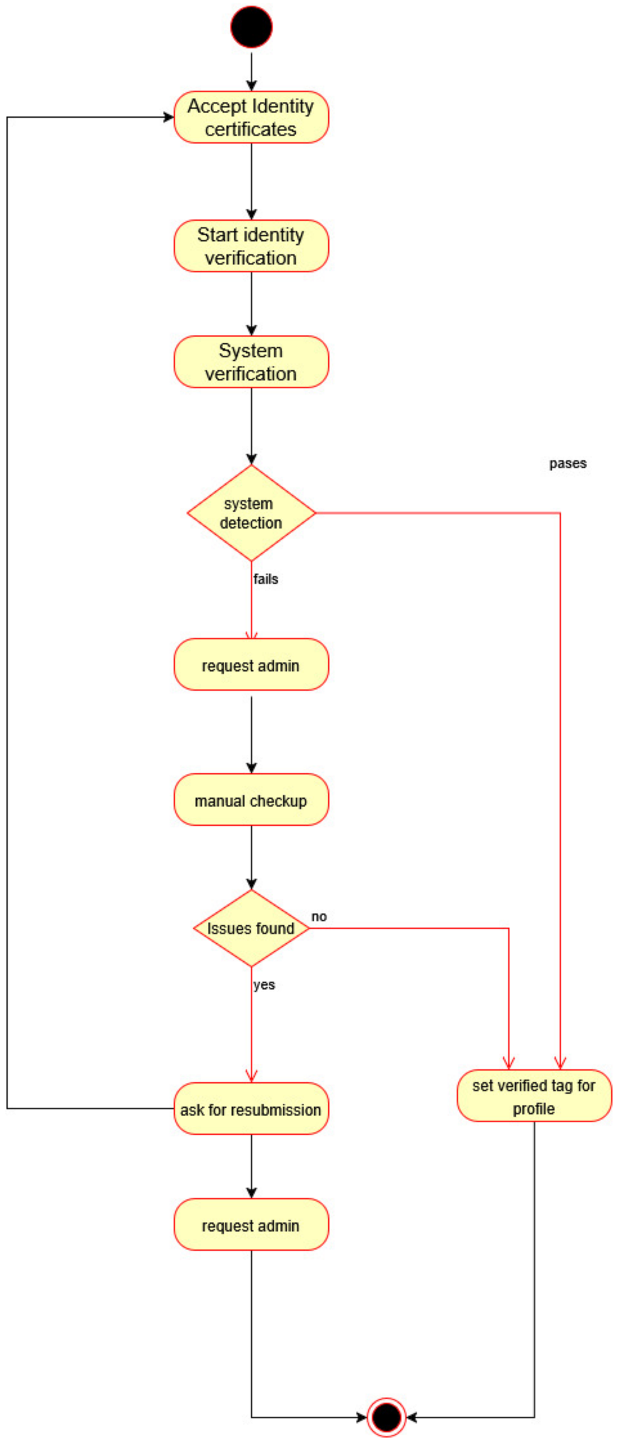
Review and Hire Gig Workers

This activity diagram outlines the end-to-end process for a Client to identify, evaluate, and engage a suitable Gig Worker.



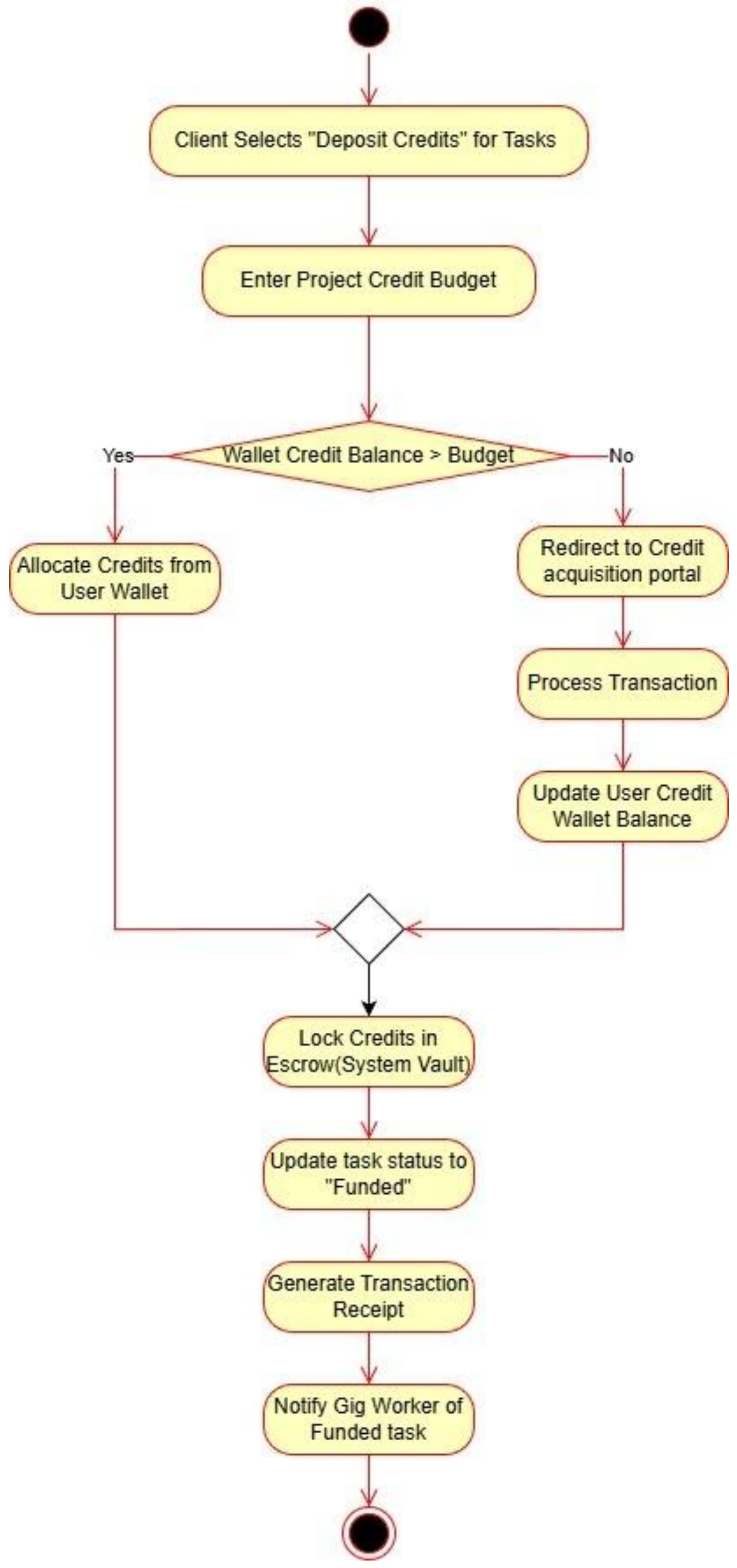
Identity Verification

This activity diagram illustrates the mandatory security protocols required to validate the authenticity of new users (both Clients and Gig Workers).



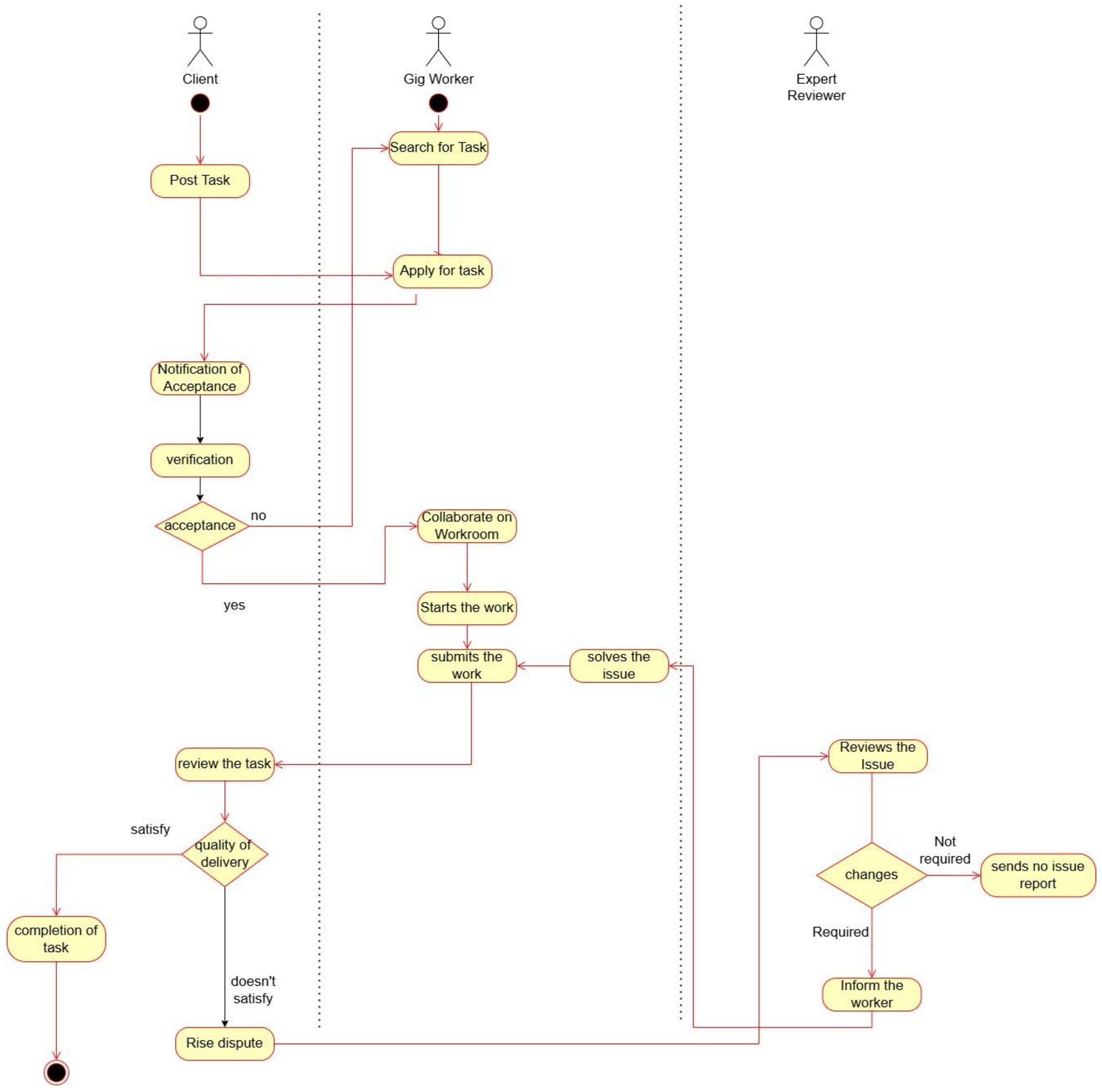
Deposit Funds to Platform

This activity diagram details the financial transaction process required for users (primarily Clients) to load credits into their platform wallet.

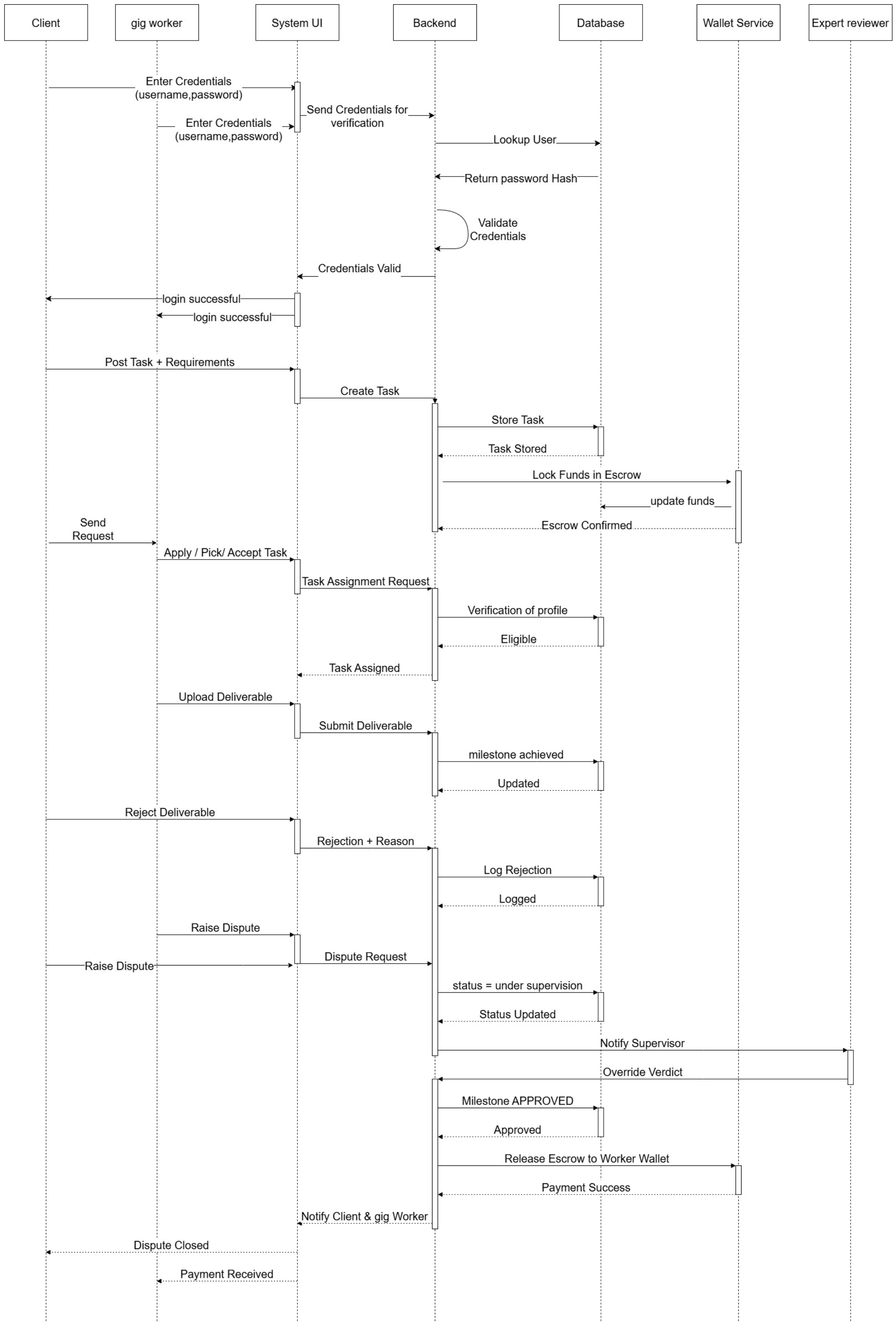

Client Selects "Deposit Credits" for Tasks
Enter Project Credit Budget
Wallet Credit Balance > Budget
Yes
No
Allocate Credits from User Wallet
Redirect to Credit acquisition portal
Process Transaction
Update User Credit Wallet Balance
Lock Credits in Escrow(System Vault)
Update task status to "Funded"
Generate Transaction Receipt
Notify Gig Worker of Funded task

SEARCH AND ASSIGN TASKS

This activity diagram details the core operational workflow for matching open work requirements with available talent.



Sequence Diagram


Client
gig worker
System UI
Backend
Database
Wallet Service
Expert reviewer
Enter Credentials (username,password)
Send Credentials for verification
Enter Credentials (username,password)
Lookup User
Return password Hash
Validate Credentials
Credentials Valid
login successful
login successful
Post Task + Requirements
Create Task
Store Task
Task Stored
Lock Funds in Escrow
update funds
Escrow Confirmed
Send Request
Apply / Pick/ Accept Task
Task Assignment Request
Verification of profile
Eligible
Task Assigned
Upload Deliverable
Submit Deliverable
milestone achieved
Updated
Reject Deliverable
Rejection + Reason
Log Rejection
Logged
Raise Dispute
Dispute Request
Raise Dispute
status = under supervision
Status Updated
Notify Supervisor
Override Verdict
Milestone APPROVED
Approved
Release Escrow to Worker Wallet
Payment Success
Notify Client & gig Worker
Dispute Closed
Payment Received

Chapter 7: Appendices

7.1 Hardware Requirements

- **Server:** The system will utilize scalable cloud infrastructure to ensure stable performance as user traffic increases.
- **Storage:** The system will use secure cloud storage to host all user-uploaded files and project deliverables.

7.2 Database Requirements

- **Logical Organization:** The database will be organized into three primary categories:
    - **User Data:** Stores identity verification, profiles, and role status (Client vs. Gig Worker).
    - **Project Data:** Stores tasks, detailed milestones, and uploaded deliverables.
    - **Financial Data:** Stores escrow ledgers, transaction history, and wallet balances.
- **Data Relationships:** The system will enforce strict connections between these categories to ensure integrity. Specifically:
    - Every **Task** must be linked to a specific **Client**.
    - Every **Financial Transaction** must be directly tied to a specific **Milestone** and **User**.
    - Every **Deliverable** must be associated with the Gig Worker who submitted it.

Chapter 8: Index