

**CCP**

**Submitted By:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Sheeraz Ali** | **Hooria** | **Sadia** | **Zain Ali** |
| Roll # | FL-23724 | FL-23752 | FL-23745 | FL-23740 |

**Course:** Software Design and Architecture

**Department:** Software Engineering

**Semester:** 4th **A** (Morning)

**Submitted To:**

**Dr. Huma Hayat Khan**

Table of Contents

[Project Description: 3](#_Toc19359)

[Functional Requirement: 3](#_Toc27677)

[Non-Functional Requirements: 4](#_Toc12513)

[Use Case Model 4](#_Toc19857)

[Full-Dress Use cases: 5](#_Toc23929)

[1. Register User Account: 5](#_Toc18181)

[2. Post Project: 6](#_Toc7428)

[3. Register Company for Bidding: 7](#_Toc2003)

[4. Conduct Real-Time Bidding: 8](#_Toc21235)

[5. Submit Proposal for Bidding: 9](#_Toc6429)

[6. Monitor Bidding Activities: 10](#_Toc12657)

[7. Process Payment: 11](#_Toc17324)

[Class Diagram 12](#_Toc26697)

[Activity Diagram 13](#_Toc25019)

[Sequence Diagram 14](#_Toc5108)

[State Machine Diagrams 21](#_Toc25671)

[1- State machine diagram for Client 21](#_Toc13895)

[2- State machine diagram for company 22](#_Toc20367)

[3- State machine diagram for Admin 23](#_Toc3408)

[Component Diagram 24](#_Toc20716)

[Deployment Diagram 25](#_Toc617)

# Project Description:

Our system design and architecture project involve developing a Tender Management Platform that connects clients (who post projects for bidding) and companies (who register and bid for projects). The system will facilitate bid management, real-time & scheduled bidding, admin moderation, and payment processing.

This platform ensures secure interactions between tender companies and clients by allowing companies to verify their credentials and certifications. Clients can choose a suitable bidding format, and the system will manage successful and unsuccessful bids efficiently. Additionally, admins will monitor the process, approve/reject projects, and track bid progress.

The system also incorporates commission-based payments and user rating mechanisms, where clients and companies can rate each other post-project completion.

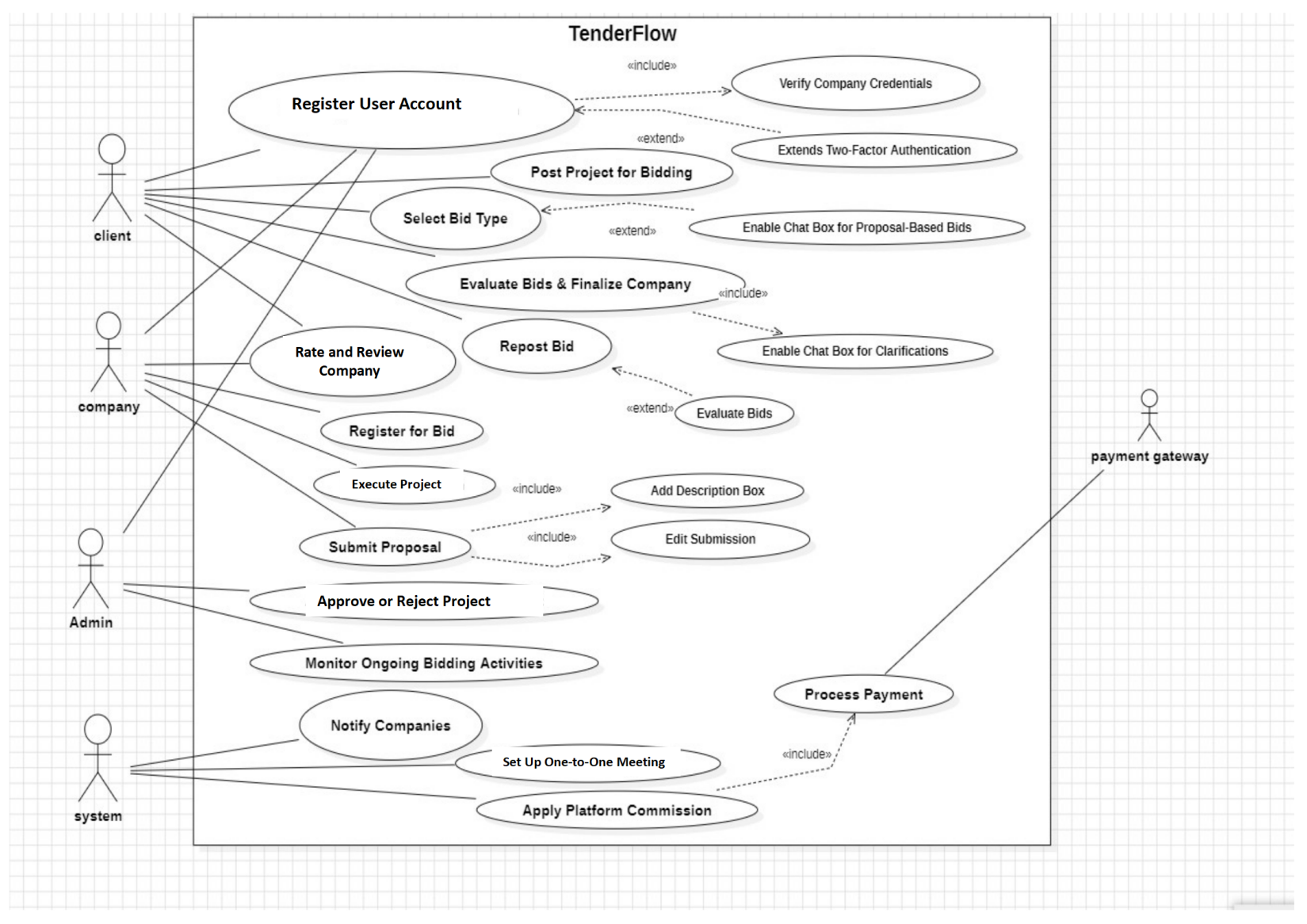
# Functional Requirement:

* System should allow clients to register and post projects for bidding.
* System should enable clients to select bid type (real-time or proposal-based) when posting a project.
* System should notify companies, allow them to register for bids, and set up one-to-one meetings for real-time bidding.
* System should notify companies, allow proposal submissions, enable optional description boxes, chat boxes, and submission edits.
* System should allow clients to evaluate proposals and select a suitable company.
* System should allow clients to repost unsuccessful real-time bids, excluding past companies.
* System should let admins monitor bidding activities and approve or reject project posts.
* System should apply a commission fee upon successful project completion.
* System should facilitate project execution, client ratings for companies, and company ratings for clients.
* System should handle payments via bank transfer or cheque upon project completion.

# Non-Functional Requirements:

* The platform should support thousands of clients and companies bidding simultaneously without performance degradation.
* The system should ensure secure login, encrypt user data, and require verified certifications from companies to prevent fraud.
* Real-time bidding should be optimized for low latency and quick response times.
* The system should handle failures gracefully, including automatic retries for failed transactions and bid submissions.
* The platform should offer an intuitive, clean UI/UX with easy navigation for clients, companies, and admins.
* The system should be modular and easily upgradable to support future enhancements and new features.

# Use Case Model



# Full-Dress Use cases:

## 1. Register User Account:

|  |  |
| --- | --- |
| Use Case ID | UC-01 |
| Use Case Name | Register User Account |
| Primary Actor | Client, Company |
| Secondary Actor | Admin |
| Preconditions | User must not be already registered. |
| Triggers | User initiates the registration process. |
| Normal Flow | 1. User accesses the registration page. 2. User provides credentials (email, password, company details if applicable). 3. System validates inputs and sends a verification email. 4. User verifies the email and activates the account. 5. User logs in with valid credentials. |
| Alternate Flow | **3a.** If verification email is not received, the user requests a resend.  **4a.** If verification fails, the system allows the user to re-enter details.  **E1.** If the email is already registered, the system notifies the user.  **E2.** If the password does not meet security requirements, an error is displayed. |
| Postconditions | User account is created and ready for use. |
| Includes | None |
| Extends | None |

## 2. Post Project:

|  |  |
| --- | --- |
| Use Case ID | UC-02 |
| Use Case Name | Post Project |
| Primary Actor | Client |
| Secondary Actor | Admin |
| Preconditions | Client must be registered and logged in. |
| Triggers | Client initiates a project posting request. |
| Normal Flow | 1. Client selects "Post a Project." 2. Provides project details (budget, category, requirements). 3. Selects bid type (Real-Time or Proposal-Based). 4. System validates inputs and submits the project. 5. Admin reviews and approves/rejects the project. 6. If approved, the project is published for bidding. |
| Alternate Flow | **5a.** If the admin rejects the project, the client is notified and can revise details.  **E1:** If mandatory details are missing, the system prompts for corrections. |
| Postconditions | The project is successfully listed for bidding. |
| Includes | Admin Approval Process |
| Extends | None |

## 3. Register Company for Bidding:

|  |  |
| --- | --- |
| Use Case ID | UC-03 |
| Use Case Name | Register Company for Bidding |
| Primary Actor | Company |
| Secondary Actor | None |
| Preconditions | Company must be registered and logged in. |
| Triggers | Company selects a project and opts to participate. |
| Normal Flow | 1. Company browses open projects. 2. Selects a project and registers for bidding. 3. System verifies company credentials and eligibility. 4. If eligible, company is registered for bidding. 5. System notifies the client about the new bidder. |
| Alternate Flow | **3a.** If the company is ineligible, a rejection message is displayed.  **E1:** If the company is blacklisted, they are not allowed to register for bids. |
| Postconditions | The company is successfully registered for bidding. |
| Includes | None |
| Extends | None |

## 4. Conduct Real-Time Bidding:

|  |  |
| --- | --- |
| Use Case ID | UC-04 |
| Use Case Name | Conduct Real-Time Bidding |
| Primary Actor | Client, Company |
| Secondary Actor | None |
| Preconditions | The project must be posted with a real-time bidding type. |
| Triggers | A company joins a real-time bidding session. |
| Normal Flow | 1. Client initiates a real-time bidding session. 2. Companies receive notifications and join the session. 3. Bidders place real-time offers. 4. If required, the client schedules one-on-one meetings. 5. Client evaluates bids and finalizes a company. |
| Alternate Flow | **5a.** If no suitable bid is found, the client reposts the bid, excluding past bidders.  **E1:** If a company disconnects, the system allows rejoining within a time limit. |
| Postconditions | A company is selected for the project, or the bid is reposted. |
| Includes | One-on-One Meeting |
| Extends | None |

## 5. Submit Proposal for Bidding:

|  |  |
| --- | --- |
| Use Case ID | UC-05 |
| Use Case Name | Submit Proposal for Bidding |
| Primary Actor | Client, Company |
| Secondary Actor | None |
| Preconditions | The project must be posted with a proposal-based bidding type. |
| Triggers | A company submits a proposal. |
| Normal Flow | 1. Client posts a project with a proposal-based bid type. 2. Companies submit proposals before the deadline. 3. Client reviews proposals. 4. Client selects the best proposal. |
| Alternate Flow | **4a.** If no proposal is selected, the client can reopen bidding.  **E1:** If a company submits an incomplete proposal, the system notifies them for revision. |
| Postconditions | A company is selected for the project, or the bid is reopened. |
| Includes | Proposal Evaluation |
| Extends | None |

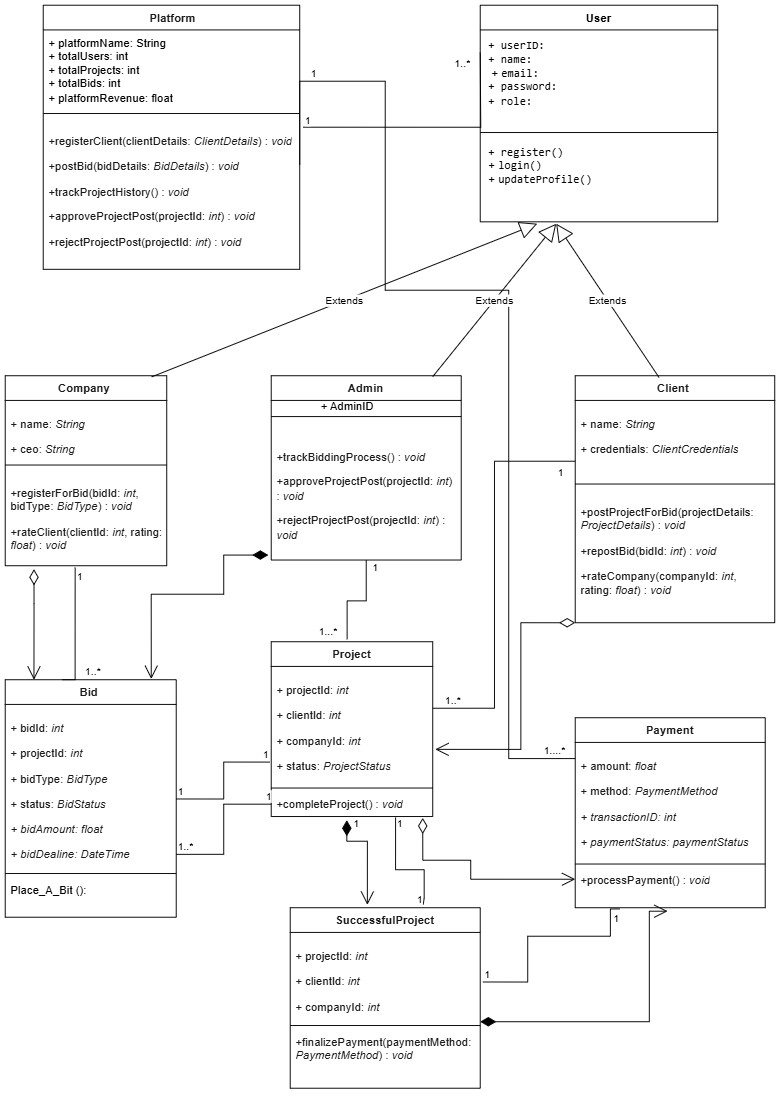
## 6. Monitor Bidding Activities:

|  |  |
| --- | --- |
| Use Case ID | UC-06 |
| Use Case Name | Monitor Bidding Activities |
| Primary Actor | Admin |
| Secondary Actor | None |
| Preconditions | Active bidding sessions must be ongoing. |
| Triggers | An admin logs into the system. |
| Normal Flow | 1. Admin accesses the monitoring dashboard. 2. Views active projects, bids, and proposal submissions. 3. Flags suspicious activity (if any). 4. Takes necessary actions (e.g., warning, bidder suspension). |
| Alternate Flow | **4a.** If a flagged bid is found valid, admin removes the warning.  **E1:** If an admin casssnnot access the dashboard, an error log is generated. |
| Postconditions | The bidding environment remains monitored and secured. |
| Includes | None |
| Extends | None |

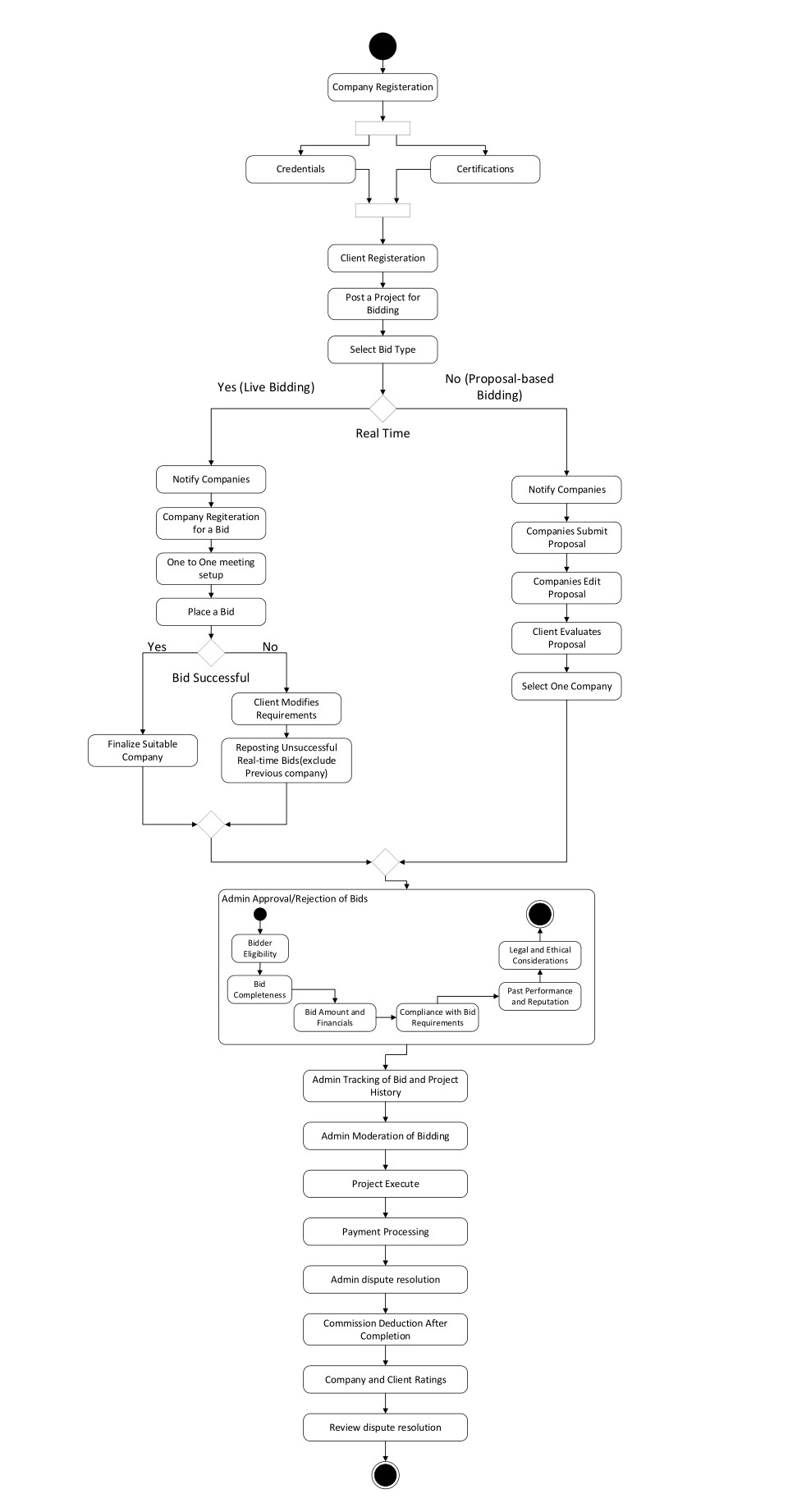
## 7. Process Payment:

|  |  |
| --- | --- |
| Use Case ID | UC-07 |
| Use Case Name | Process Payment |
| Primary Actor | Client, Company |
| Secondary Actor | Bank |
| Preconditions | The project must be completed and approved. |
| Triggers | A client initiates payment for a completed project. |
| Normal Flow | 1. Client selects payment method (Bank Transfer or Cheque). 2. System generates an invoice. 3. Payment request is sent to the bank. 4. Bank confirms transaction status. 5. System updates payment records and notifies stakeholders. |
| Alternate Flow | **4a.** If payment fails, the client is prompted to retry or use an alternate method.  **E1:** If bank servers are down, the transaction is queued for retry. |
| Postconditions | The payment is successfully processed. |
| Includes | None |
| Extends | None |

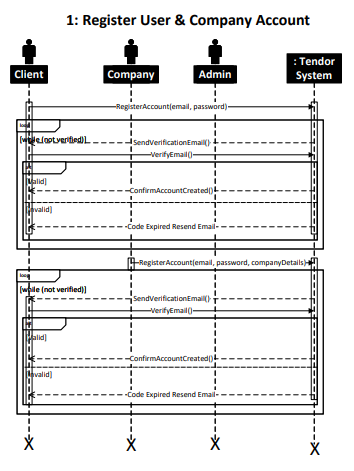
# Class Diagram

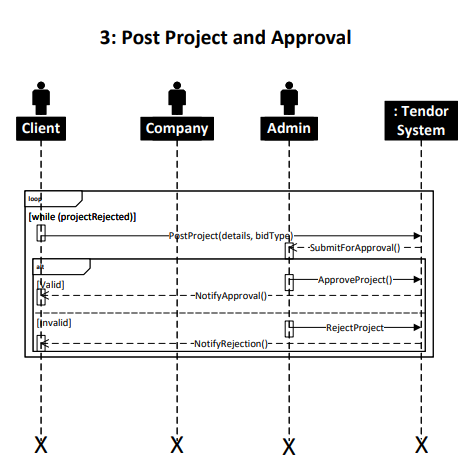
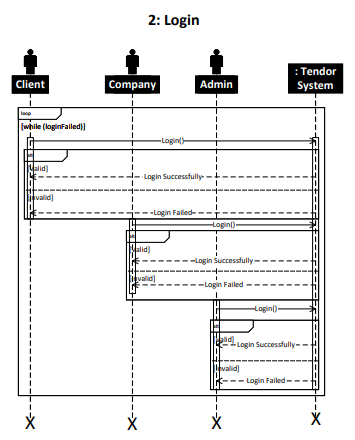


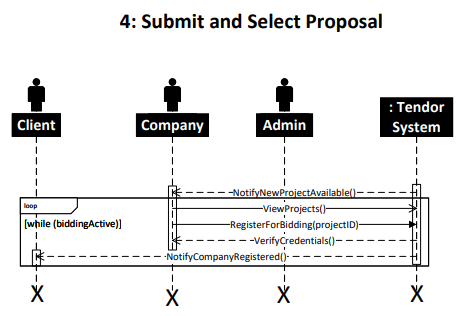
# Activity Diagram

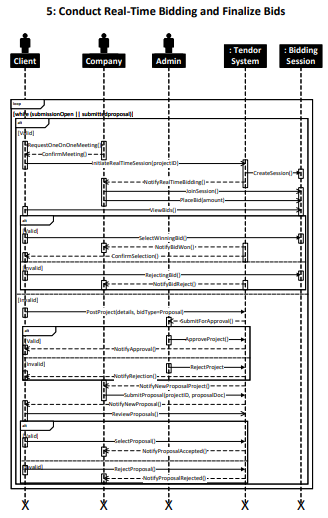


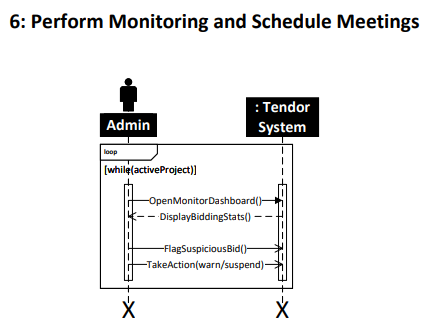
# Sequence Diagram

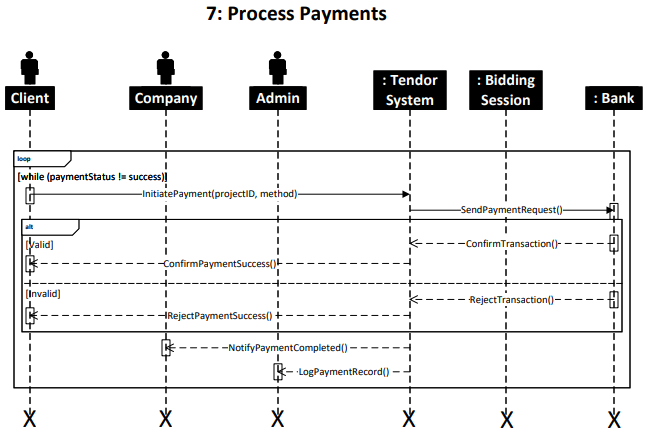




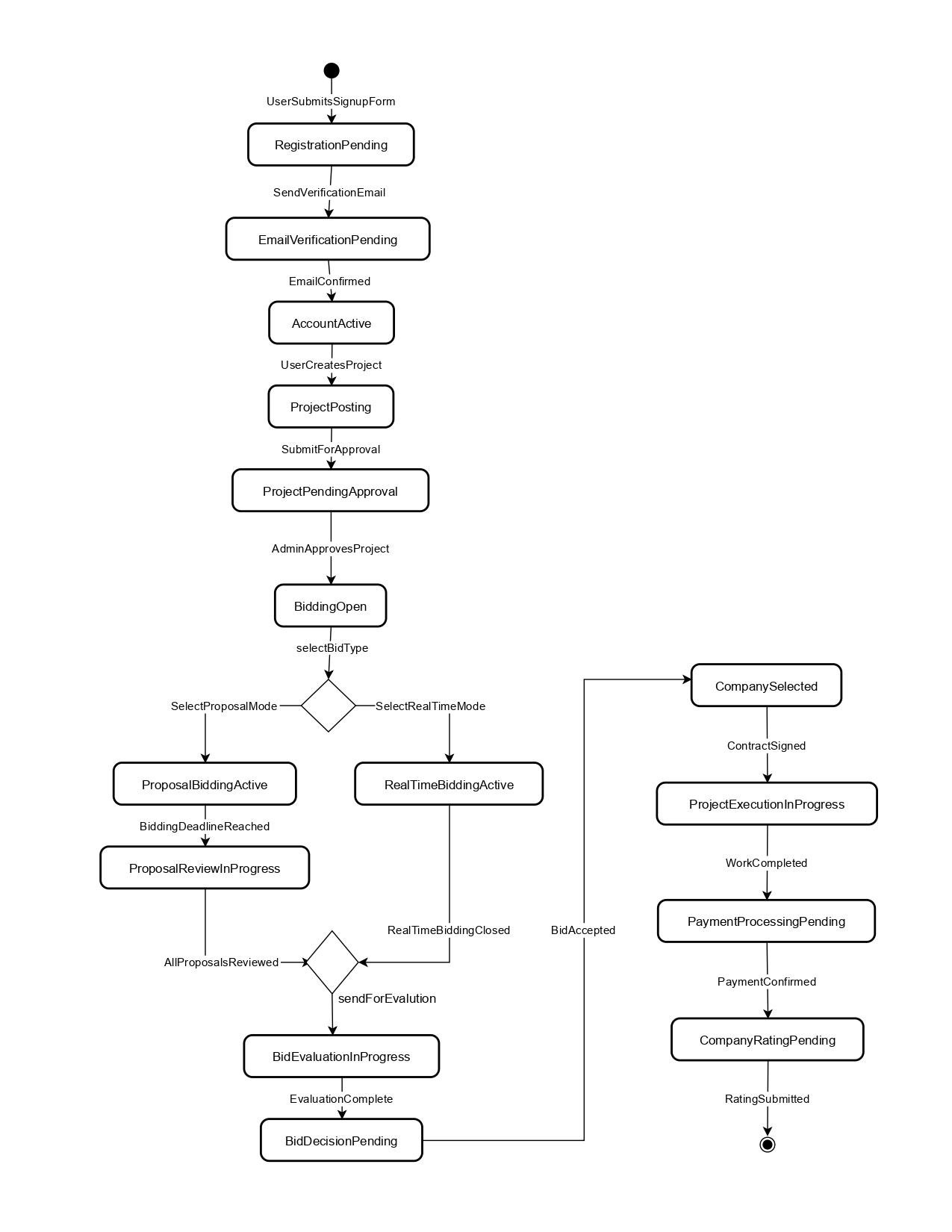
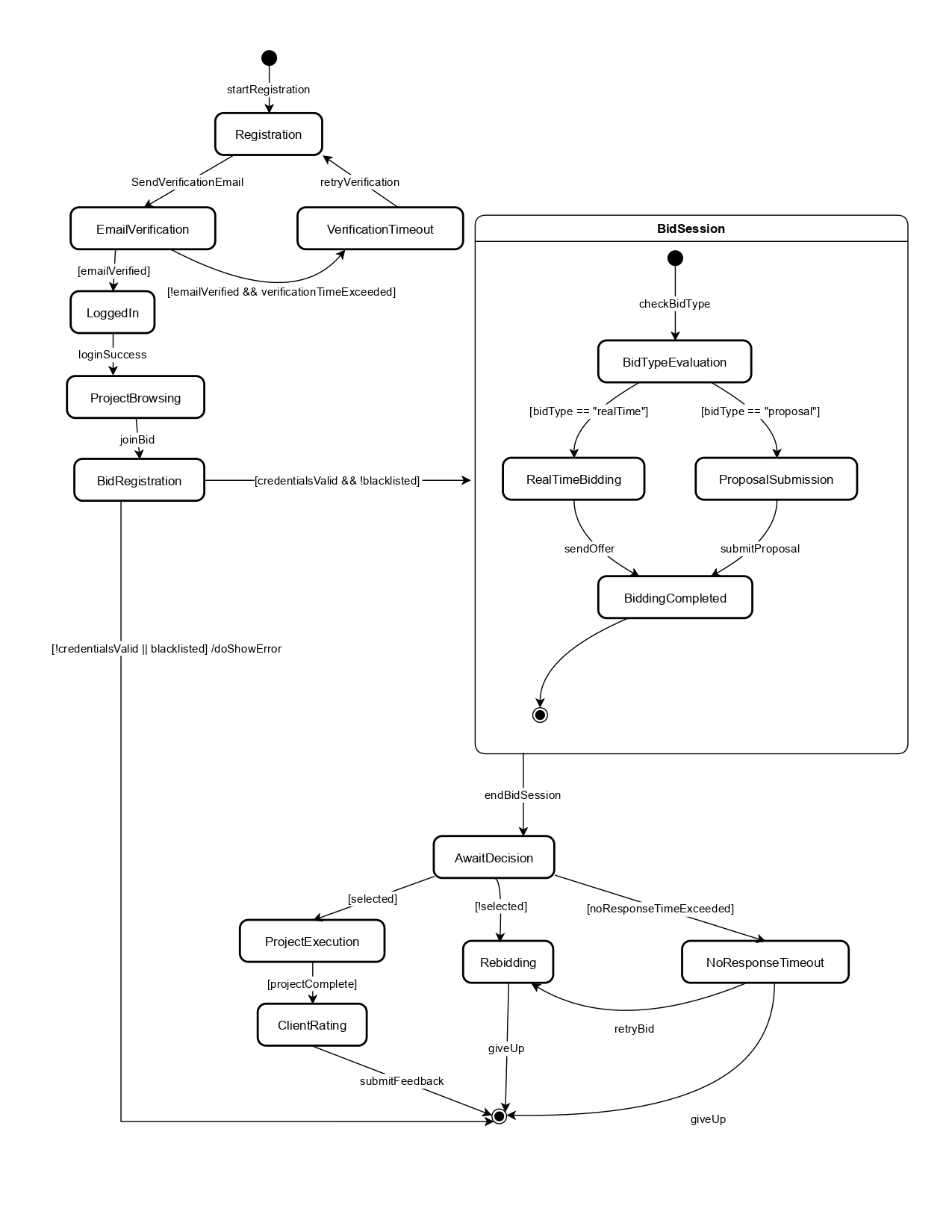








# State Machine Diagrams

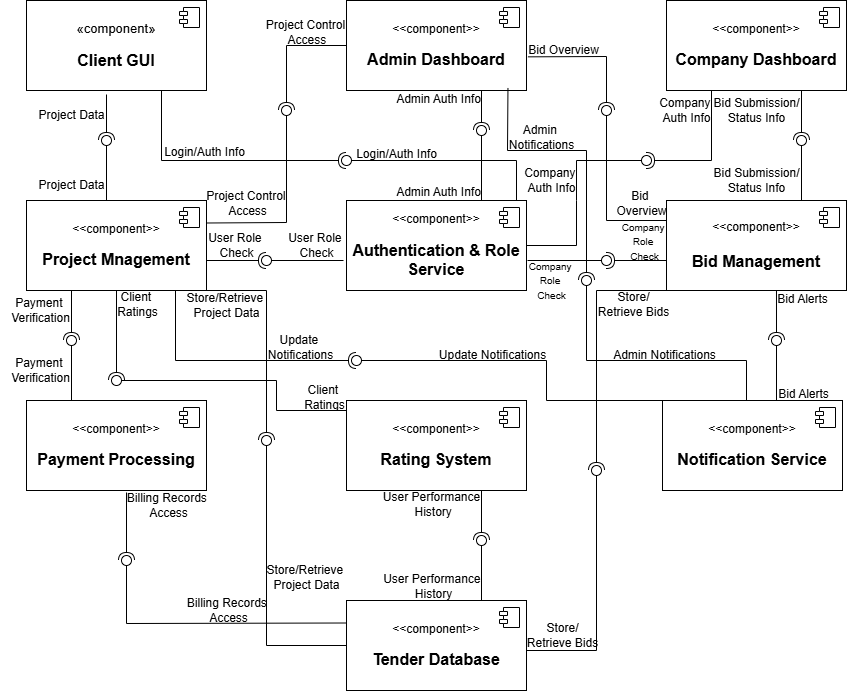
1. ****State machine diagram for Client
2. ****State machine diagram for company

## State machine diagram for Admin

**A diagram of a flowchart

AI-generated content may be incorrect.**

# Component Diagram



# Deployment Diagram

