Lok Lagbe?

# **1. Abstract**

As the name suggests, **“Lok Lagbe?”** is a mobile-first web application developed to manage the everyday task-posting and helping ecosystem in local communities. In recent years, the demand for informal help in urban and semi-urban areas has increased, but there is no organized digital platform to connect task posters with local helpers. Traditionally, such tasks are posted on social media or managed informally, leading to inefficiencies and trust issues.

This project addresses those gaps by offering a digital platform that facilitates task posting, browsing, and verifying users through email and NID submission. By removing unnecessary features like payment processing from the Minimum Viable Product (MVP), the project aims for fast, achievable delivery while establishing a strong user foundation and trust system.

## **2. Introduction**

### **2.1 Overview of the Project**

The **“Lok Lagbe?”** system is designed to provide a simple, secure, and mobile-optimized solution for connecting people who need help with those offering it. Currently, there is no structured way for users to easily find trusted local helpers for small, everyday tasks (such as cleaning, shifting, errands, tutoring, etc.). As a result, many people rely on unverified sources or word-of-mouth, which can be unreliable or unsafe.

This platform allows users to:

* Register and verify their identity using email and NID.
* Post tasks manually specifying category, description, and location.
* Browse available tasks based on location and category.
* Track task status from posting to completion.
* Rate and review each other after task fulfillment.

By digitizing and streamlining these interactions, **“Lok Lagbe?”** aims to reduce friction, improve trust, and enable safe collaboration in everyday work environments.

**2.2 Problem Definition**

#### **2.2.1 Task Posting Form**

Users can submit tasks by filling out a digital form. The form includes fields like task title, description, date, category, and location. Once submitted, it becomes available for browsing. Helpers can then review and accept tasks based on availability and interest.

#### **2.2.2 Notification and Reporting**

Users are notified about task status changes (e.g., accepted, completed) through the app interface. A reporting mechanism allows users to flag inappropriate behavior or fake profiles. Admins can review these reports and take necessary actions.

#### **Administrator Access**

Admins have special privileges and can:

1. View, edit, or delete user and task records.
2. Review reports and take action (e.g., block users).
3. Moderate content and manage trust/safety settings.

#### **2.2.3 Task Workflow Management**

The task lifecycle follows a clearly defined sequence:

* Posted → Accepted → Pending → Completed

Admins ensure that task flows are valid and intervene in case of abuse. Since payments are excluded in the MVP, the workflow remains lightweight and focused on matching and communication.

# 3. Feasibility Study

# 3.1 Operational Feasibility

Lok Lagbe? is designed for users seeking quick help with everyday tasks and for local helpers to find work.

Operationally, the app is minimal and mobile-first, supporting core interactions like task posting, task acceptance

* Easy onboarding via email verification.
* No dependency on payment gateways.
* Admin moderation ensures trust and reliability.
* Designed to scale for urban or semi-urban populations.

Conclusion: Operationally feasible and practical for rollout with a small team.

# 3.2 Technical Feasibility

The app uses proven technologies: Firebase for authentication, database; React Native for cross-platform support.

**Frontend**: React Native for cross-platform mobile app (Android & iOS).

**Authentication**: Firebase (preferred) — supports Email verification.

**Backend**:

**Simple app**: Firebase (Firestore rules).

**Database**:  
Firebase Firestore.

**NID Verification**: Use a 3rd-party API (via backend) to verify NID numbers.

**Location**: Manual entry — avoids need for map integration.

Conclusion: Technically feasible using modern frameworks and minimal resources.

# 3.3 Schedule Feasibility

Planned over 4 weeks:

* Week 1: User auth, task category, task posting, navigation.
* Week 2: Task browsing, filtering, status tracking.
* Week 3: Ratings, admin tools, profile section.
* Week 4: Testing, bug fixes, deployment.

Conclusion: Schedule is realistic for a small team with focused scope.

# 3.4 Risk Assessment

Risk - Mitigation

Spam/fake users: Email verification and admin moderation.

Scope creep: MVP boundaries fixed, optional features delayed.

Monetary Issue: Money trust issue.

# 4. Software Requirements and Specifications

# 4.1 Functional Requirements

User Categories:

* User (Poster/Helper)
* Admin

User Functionalities:

* FR1: Register with email verification.
* FR2: Upload NID before accessing platform.
* FR3: Browse tasks, filter by category and location.
* FR4: Post task (title, category, description, manual location).
* FR5: Accept tasks and track status (Posted → Accepted → Pending → Completed).
* FR7: Rate and review after task completion.

Admin Functionalities:

* FR8: View, edit, delete users/tasks.
* FR9: Handle reports and manage blocked users.

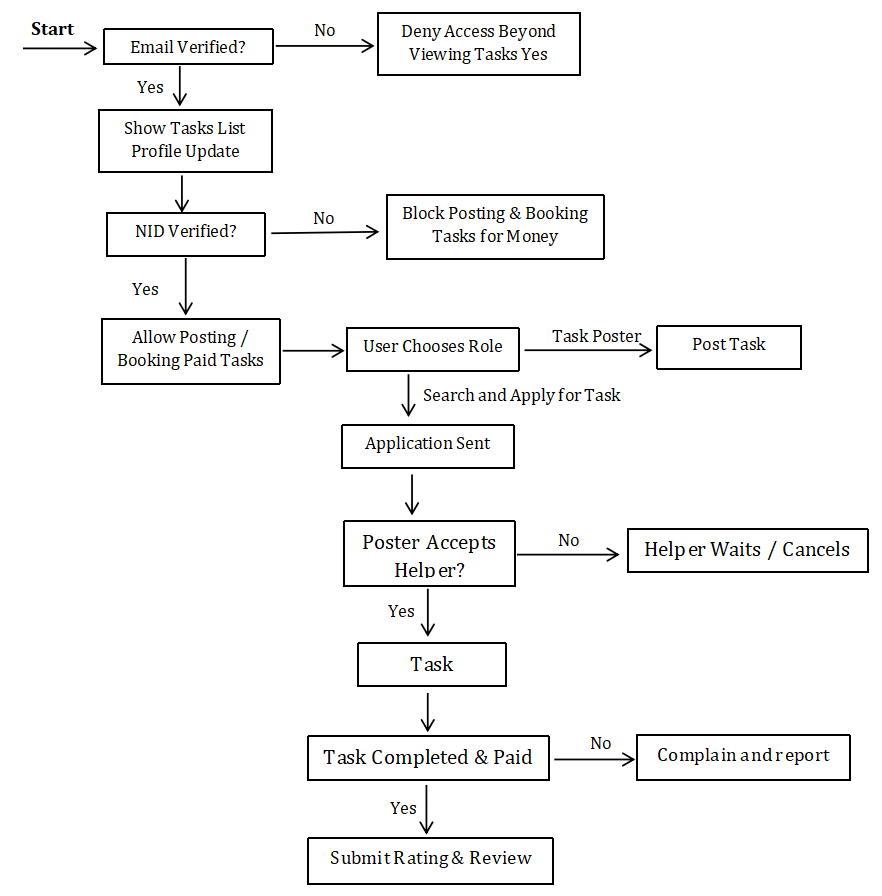
# 4.2 Non-Functional Requirements

Interface: Responsive mobile-first UI.

* Performance: <3s response time for common operations.
* Security: Email verification, NID check, admin moderation.
* Scalability: Handles 1K concurrent users in Firebase setup.
* Availability: Target 99% uptime.

# 5. Decision Tree and Decision Table

# 5.1 Decision Tree (User Perspective)



# 5.2. Decision Table

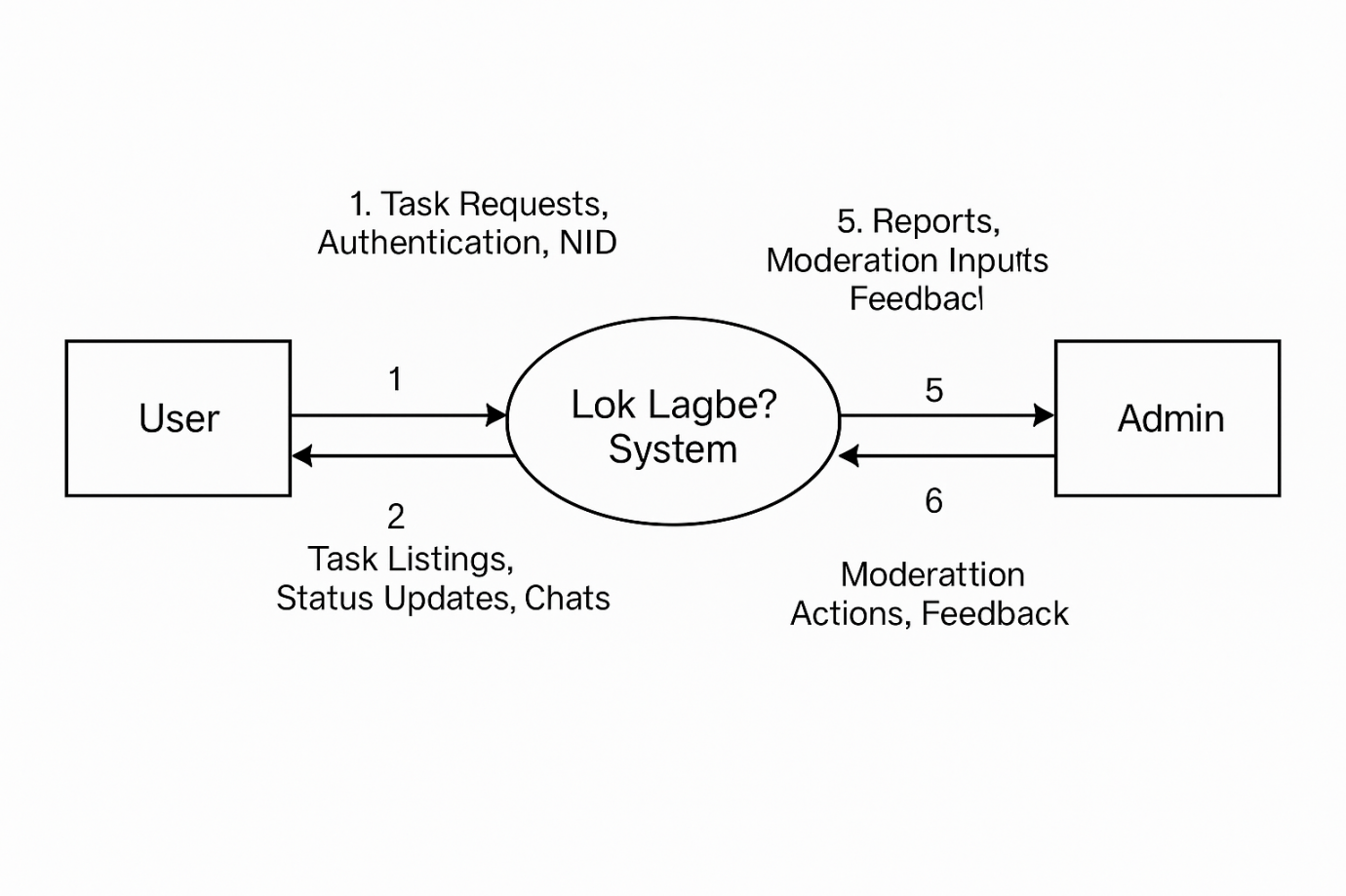
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condition | Email Verified | NID Verified | User Role | Action |
| User not logged in | No | N/A | N/A | N/A |
| Email verified, but no NID | Yes | No | N/A | Only View tasks, update profile |
| Email & NID verified | Yes | Yes | Task Poster | Can post-paid tasks |
| Email & NID verified | Yes | Yes | Helper | Can apply/book paid tasks |
| Task applied & accepted | - | - | - | Task status moves to Accepted |
| Poster confirms completion | - | - | - | Confirm completion |
| Ratings submitted | - | - | - | End process |

# 

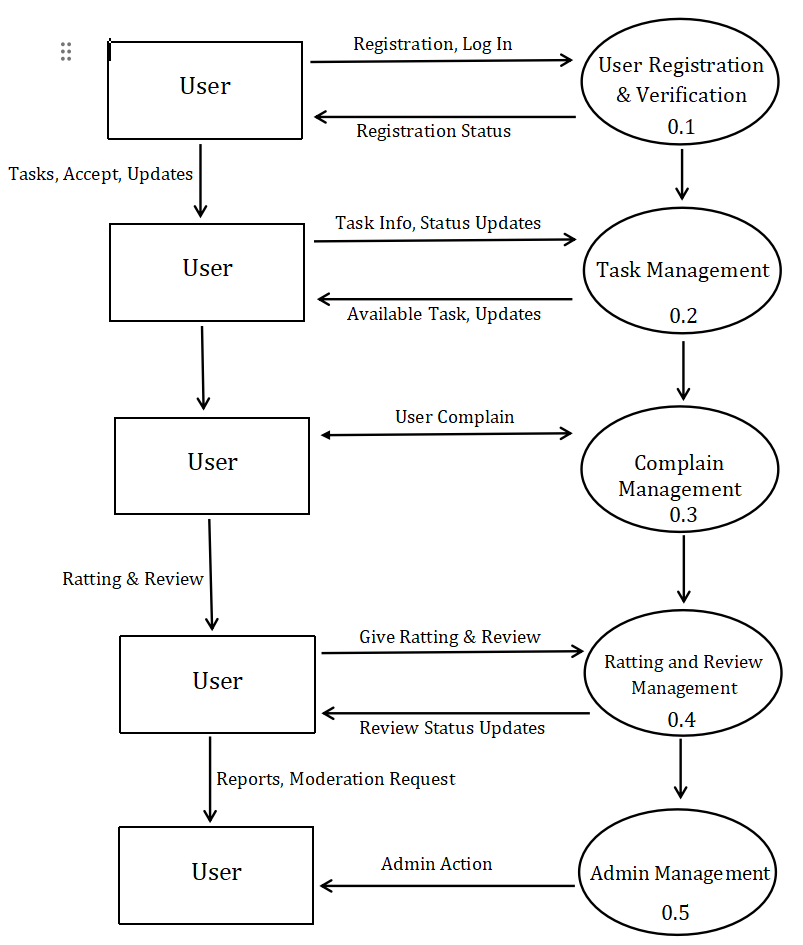
# 7. Design

## 7.1 Data Flow Diagram (DFD)

# Level 0 – Context Diagram



# Level 1 – Detailed DFD



**Use Case Diagram:**

****