NMCS Website & Contribution Management System

Product Requirements Document (PRD)

Backend: Django + Python

---

1. Project Proposal

The NMCS Website is a centralized digital platform designed to provide information about the group, showcase leadership and members, display events and activities, and enable secure financial contributions through digital payment integration. The system aims to improve transparency, accountability, and administrative efficiency within the organization.

Problem Statement

- Manual tracking of financial contributions.
- Limited transparency in total collections.
- Lack of centralized record for events and leadership.
- Administrative inefficiencies in communication and record keeping.

Objectives

- Implement user authentication (registration, login, logout).
- Display group leaders and members.
- Manage and display events.
- Integrate M-Pesa contributions via Daraja API.
- Provide personal and group contribution dashboards.
- Ensure security, scalability, and responsiveness.

2. System Architecture

The system follows a three-tier architecture:

1. Presentation Layer:
- HTML, CSS, JavaScript
- Django Templates

2. Application Layer:
- Django Framework
- Business logic handling
- Payment processing integration

3. Data Layer:
- PostgreSQL or MySQL
- Django ORM

External Integration:
User -> Website -> Daraja API -> M-Pesa STK Push -> Callback -> Django -> Database Update

3. UML Use Case Overview

Actors:
- Visitor
- Registered User
- Admin
- M-Pesa System

Visitor:
- View homepage
- View leaders and members
- View events
- Register account

Registered User:
- Login/Logout
- Make contributions
- View personal dashboard
- View contribution history

Admin:
- Manage leaders
- Manage events
- View all contributions
- Manage users

M-Pesa System:
- Receive payment request
- Send STK push
- Send transaction callback

4. Entity Relationship Diagram (ERD)

Entities:

User:
- id
- username
- email
- password

Profile:
- id
- user (OneToOne)
- phone_number
- role

Leader:
- id
- name
- position
- message
- image

Member:
- id
- name
- image

Event:
- id
- title
- description
- event_date
- location
- status

Contribution:
- id
- user (ForeignKey)
- amount
- transaction_code
- status
- created_at

Relationships:
- User has one Profile
- User has many Contributions
- Admin manages Leaders and Events

5. Two-Month Development Roadmap

Month 1:
Week 1: Project setup, Django configuration, database setup.
Week 2: Authentication system and role-based access.
Week 3: Leaders, Members, and Events models with CRUD operations.
Week 4: Contribution system (without payment integration) and dashboard.

Month 2:
Week 5: M-Pesa Daraja API integration (STK push and callback handling).
Week 6: Dashboard analytics and UI improvements.
Week 7: Security improvements and optimization.
Week 8: Testing, documentation, and deployment.