Grand
Technology Solutions
INNOVATE.INSPIRE.TRANSFORM.

Tigray Youth Association Members Management System (TYAMMS)

Project Proposal

1. Executive Summary

The **Youth Association Members Management System (TYAMMS)** is an advanced digital solution designed to streamline the management of youth association members, their annual payments, and the administrative hierarchy across multiple organizational levels. It is a comprehensive web application designed for centralized management at the highest administrative level and a mobile application tailored for use across all levels, from regional to kebele. This system ensures efficient governance by integrating a tiered structure of administrators, including regional, zone, werreda, and kebele levels, with each level overseeing and managing its respective jurisdiction. By providing tailored functionality for each administrative tier, TYAMMS enables seamless member registration, payment tracking, receipt generation, and data reporting, fostering transparency and operational efficiency across the organization.

2. Introduction

2.1. Background

Youth associations play a pivotal role in empowering communities and fostering societal development. However, the manual management of association members, payments, and reports across complex administrative hierarchies creates significant inefficiencies. Current processes lack standardization, transparency, and scalability, which limits the association's ability to operate effectively.

2.2. Problem Statement

Managing members and payments across regions, zones, werreda, and kebeles presents the following challenges:

- Time-consuming and error-prone manual processes for member registration, payment tracking, and receipt issuance.
- A lack of centralized visibility into administrative and financial data across hierarchical levels.
- Inefficient workflows for submitting, reviewing, and approving reports between administrative levels.
- Limited tools to ensure accountability, transparency, and data accuracy.

These challenges hinder the operational efficiency of youth associations and limit their ability to serve their members effectively.

3. Objectives

3.1. Proposed Solution

The **Tigray Youth Association Members Management System (TYAMMS)** offers an integrated platform to address these challenges by providing:

- **Centralized Member Management**: A web and mobile application for registering members, tracking payments, and issuing receipts at the kebele level, with hierarchical reporting to werreda, zone, and regional levels.
- **Payment Tracking and Receipt Generation**: An automated system to record payments, generate digital receipts, and provide payment histories for members.
- **Report Approval Workflows**: A streamlined process for submitting, reviewing, and approving reports at every administrative level, ensuring transparency and accountability.
- **Role-Based Access Control**: Secure access for super admins, regional admins, zone admins, werreda admins, and kebele admins to manage their respective jurisdictions effectively.
- **Real-Time Monitoring**: Dashboards for super admins and higher-level administrators to view member statistics, payment data, and pending tasks in real time.

4. Scope of the Project

4.1. In-Scope

- **Member registration and management** at the kebele level.
- **Payment tracking and receipt generation for** annual dues.
- **Administrative hierarchy** for kebele, werreda, zone, region, and super admin levels.
- **Approval workflows for reports** submitted between administrative levels.
- **Web application** for super admin, regional, and zone-level users.
- **Mobile application** for kebele and werreda users.

4.2. Out-of-Scope

- **Integration with external financial systems (**planned as future enhancement).
- **Non-administrative user access** (e.g., members accessing their own profiles).

5. System Architecture

- **Frontend**: React-Js for the web app and Flutter for the mobile app, ensuring responsive and cross-platform support.
- **Backend**: Node.js with Express.js for robust and scalable server-side operations.
- **Database**:
  - Mongo DB: As a database for all transactions and data.
  - SQLite: For offline to manage client management and payment by kebele administrators.

- Use the following fields for designing the database.
  - **AdministratorsTable**

    Fields: userId, name, email, password, role, regionId, zoneId, weredaId, kebeleId (in some adminstrators some of them are not filled e.g. for regional administrators no need of the zoneId, werredaId, kebelleId),date of birth)

    Purpose: Manage user information, roles, and administrative hierarchy.

  - **Regions Table**

    Fields: regionId, regionName.

    Purpose: Store information about each region.

  - **Zones Table**

    Fields: zoneId, zoneName, regionId.

    Purpose: Store zone information and associate it with a region.

  - **Wereda Table**

    Fields: werreda, weredaName, zoneId.

    Purpose: Manage wereda details under respective zones.

  - **Kebele Table**

    Fields: kebeleId, kebeleName, weredaId.

    Purpose: Manage kebele-level information.

  - **Members Table**

    Fields: memberId, name, phoneNumber, kebeleId, Date of birth,address(regionId,zoneId,werredaId,kebelleId)

    Purpose: Store data on individual members.

  - **Payments Table**

    Fields: paymentId, memberId, paymentYear, amount, status, submittedDate.

    Purpose: Track member payments annually.

  - **Receipts Table(We will discuss further about this to decide what scenario to use to decide what to give ether receipt paper or code)**

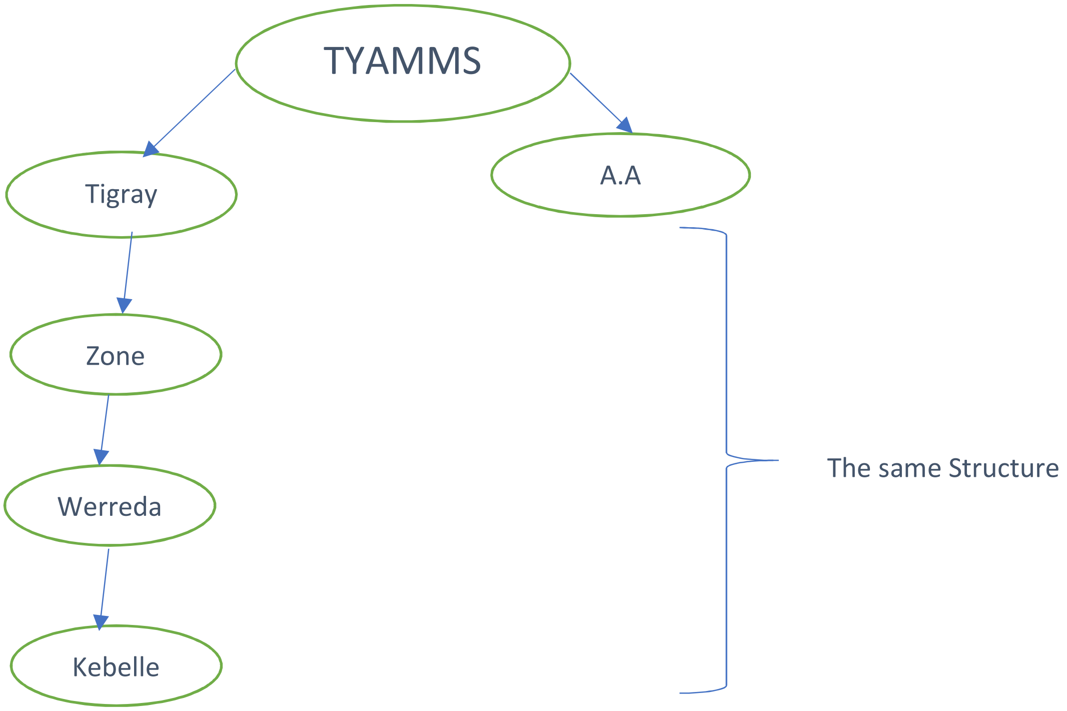
Fields: receiptId, paymentId, issueDate.

Purpose: Generate and store payment receipts.

- **Approval and disapproval table**

Fields: id,requestType,entityId,intiotorId,status(approved,pending),approverId, decisionReason(reason given by the approver)

NB. While developing you may have other fields and you can change anything based on your understanding and necessity of the system. And we can merge the administrators table and the members table



6. Functional Requirements

6.1. Member Management

- The system shall allow kebele administrators to:
  - Add members with fields (*Member Name, Member ID, Date of Birth, Gender, Contact Information, Address (Region, Zone, Werreda, Kebele), Membership Start Date, Membership Status (Active/Inactive*)
  - update, and validate member records (for the given fields).
  - Search and retrieve member details.

- The system shall allow werreda administrators to:
  - Monitor and validate member registration data submitted by kebeles.
- The system shall allow zone and regional administrators to:
  - View and oversee member data within their jurisdictions.
- The system shall allow the super admin to:
  - Access all member data across all administrative levels.
  - Manage policies regarding member registration fields.
- Manage Approval or disapproval(by top-lsevel) of low-level request for delate and update

6.2. User Access Control (Based on their role)

The system shall enforce role-based access control:

- Super admin shall manage access at all levels.
- Regional, zone, and werreda administrators shall manage access for their respective subordinates.
- Kebele administrators shall manage their own access credentials.
-

6.3. Payment Management

- The system shall allow kebele administrators to:
  - Record annual payments from members (*e.g. Payment ID, Member ID, Payment Date, Amount Paid, Receipt Number, Payment Status (Paid, Pending, Overdue*))
  - Generate and issue payment receipts.
  - Operate in offline mode using SQLite for data entry and storage.
  - Synchronize payment data to MongoDB when connectivity is restored.
- The system shall allow werreda administrators to:
  - Validate payment data submitted by kebeles.
  - Monitor payment collection status within their jurisdiction.
- The system shall allow zone and regional administrators to:
  - View and validate payment data submitted by lower levels.
  - Monitor payment collection progress.
- The system shall allow the super admin to:

  - Set payment deadlines.
  - Generate reports on payment collection across all levels.

- Manage Deadlines (for the kebele to submit their offline data)

6.4. Administrative Management

- The system shall allow administrators at all levels to manage accounts and settings for users below their level.
- The super admin shall:
  - Add and manage regional administrators
  - Oversee and configure the entire administrative hierarchy.
- The regional, zone, and werreda administrators shall:
  - Add and manage lower-level administrators within their jurisdictions.
  - View and validate administrative activities of lower levels.
- Manage everything through fields (*Admin Name, Admin ID, Role (Region, Zone, Werreda, Kebele), Contact Information, Assigned Area (Region, Zone, Werreda, Kebele*)

6.5. Report Management

- The system shall generate detailed reports on:
  - Member registrations at all administrative levels.
  - Payment collections and compliance with deadlines.
  - Administrative activities (e.g., account creation, synchronization logs).
- The system shall allow administrators to:
  - Filter reports by date, zone, werreda, and kebele.
  - Export reports to PDF or Excel for offline use.

6.6. Offline Data Handling

The system shall:

- Enable kebele administrators to work offline using SQLite for local data storage.
- Synchronize local data with MongoDB when connectivity is available.
- Alert kebele administrators when synchronization is required (e.g., approaching deadlines).

6.7. Notification and Alert Management

The system shall send alerts to administrators about:

- Pending or overdue tasks (e.g., payment submissions).
- Data synchronization issues for kebele administrators.

- Changes in policies or deadlines from the super admin.

6.8. Deadline Management

- The system shall allow the super admin to set and enforce deadlines for:
  - Payment collection.
  - Member registration data submission.

The system shall allow notifications to be sent via email or in-app messages.

7. Non-Functional Requirements

- **Scalability**: Support thousands of members across regions and hierarchical levels.
- **Performance**: Ensure fast response times for all user actions.
- **Usability**: Intuitive interfaces for both web and mobile applications.
- **Security**: Implement encryption, secure authentication, and data protection measures.

8. Project Deliverables

- Functional web and mobile applications.
- Documentation, including user guides and system manuals.
- Training sessions for administrative users at each level.
- Deployment on a secure hosting platform.

9. Conclusion

The **Youth Association Members Management System (YAMMS)** is a transformative solution that leverages modern technology to address the inefficiencies in managing youth associations. By combining web and mobile platforms, it ensures seamless member management, payment tracking, and hierarchical administration. With YAMMS, youth associations can operate with greater transparency, efficiency, and accountability, empowering their members and driving organizational success.