

# Christ Embassy Ireland Zone — Church Reporting Platform

Product Requirements Document (PRD)

Prepared by: Agbodesi Imoagene

October 30, 2025

Version 0.1

Confidential — For Internal Use Only

## Document Control

<b>Document Title</b>	Christ Embassy Ireland Zone Church Reporting Platform — Product Requirements Document
<b>Version</b>	0.1
<b>Author(s)</b>	Agbodesi Imoagene
<b>Date</b>	October 30, 2025
<b>Status</b>	Draft
<b>Confidentiality</b>	Confidential — For Internal Use Only

## Revision History

Version	Date	Author	Description of Change
0.1	October 30, 2025	A. Imoagene	Initial draft created

## Approvals

Name	Role	Date Approved
	Zonal Pastor	
	Group Pastor Representative	
	Technical Lead	

# 1 Executive Summary

The Christ Embassy Ireland Zone Church Reporting Platform is a unified web-based solution designed to streamline record-keeping, reporting, and oversight across key aspects of ministry administration. The platform will consist of four dedicated sub-domains:

- **Registry** — for managing member information, first-timer registrations, service attendance, and departmental assignments.
- **Finance** — for logging monetary giving such as offerings, tithes, partnerships, and other contributions, including support for partnership arms like Rhapsody of Realities, The Healing School, the InnerCity Mission for Children, and Loveworld Television Ministry.
- **Cells** — for the reporting of cell ministry activities, including attendance, testimonies, outreach results, and cell offerings.
- **Reports** — a centralised reporting portal for pastors and leaders to view dashboards, generate reports, and monitor ministry performance across the organisational hierarchy (regions, zones, groups, local churches, and outreaches).

The system is designed to replace fragmented manual processes and legacy spreadsheets with a single, reliable, and secure platform. It will improve the accuracy of records, enhance transparency in financial reporting, and provide pastors with timely insights for effective decision-making.

The platform will feature a unified backend and a restrictive role-based access control model. User accounts will be created top-down, with the Zonal Pastor able to provision accounts for Group Pastors, who can in turn provision for Local Church Pastors or Coordinators. Pastors can also create portal-specific roles such as Finance Officers, Church Administrators, or Cell Leaders. Each user's permissions are limited to their assigned scope within the church hierarchy.

Key objectives of the platform include:

- Establishing a single source of truth for membership and financial data.
- Ensuring accountability and auditability of records.
- Providing actionable insights through dashboards and KPIs on attendance, giving, assimilation, and cell ministry effectiveness.
- Reducing administrative overhead and reliance on spreadsheets.
- Supporting growth and scalability of ministry operations within the zone.

This PRD defines the requirements, scope, and success criteria for the platform. It is intended as a reference for stakeholders, developers, and technical teams responsible for its implementation.

## 2 Background and Context

The Christ Embassy Ireland Zone operates a large network of churches, groups, and cells spread across multiple counties. Administrative record-keeping and reporting have historically relied on manual spreadsheets, making it difficult to consolidate accurate data on membership, attendance, and financial partnerships.

To improve data integrity, visibility, and coordination across the zone, a unified digital platform is required. The Christ Embassy Ireland Zone Church Reporting Platform provides a centralised system for recording, verifying, and reporting key operational metrics while preserving the church’s hierarchical structure.

The platform streamlines four existing workflows—membership management, finance reporting, cell ministry coordination, and leadership dashboards—into a single integrated solution optimised for accuracy, accountability, and ease of use.

## 3 Objectives and Goals

The primary objective of the Christ Embassy Ireland Zone Church Management Platform is to provide a unified, secure, and scalable backend for managing people, finance, cells, and reports across the zone. The system is designed to replace manual processes and fragmented spreadsheets with structured data entry, controlled access, and automated reporting.

### Goals

- Establish a single source of truth for membership, attendance, financial contributions, and cell ministry reports.
- Ensure data integrity and accountability through role-based access control and audit logging.
- Provide timely and accurate dashboards and KPIs for pastors at different levels of oversight (zone, group, church, cell).
- Standardise data entry formats to reduce duplication and errors.
- Enable efficient migration of legacy records into a unified database.
- Support growth through modular design and scalability.

### Non-Goals

- Payment processing — the platform records giving, but does not process transactions directly.
- Event scheduling, media management, or volunteer rostering — these are out of scope for the MVP.
- Replacing KingsPay or other existing giving platforms — the focus is on record-keeping and reporting.
- Advanced CRM or mass communication features — only essential follow-up fields and contact flags will be included.

## 4 Project Scope

The Christ Embassy Zone Church Management Platform will deliver a unified backend with four dedicated sub-domains, each optimised for a specific operational area. All modules share a common data model and a single access control system.

### In Scope

- **Registry Portal** (`registry.<zone>.ce.church`)
  - Member information management (personal details, contact info, marital status, foundation school completion, baptism date).
  - First-timer registration and follow-up status tracking.
  - Service attendance records (men, women, teens, kids, first timers, new converts).
  - Departmental assignments and leadership roles.
- **Finance Portal** (`finance.<zone>.ce.church`)
  - Recording of offerings, tithes, seeds, first-fruits, and other giving types.
  - Partnership arm support (Rhapsody of Realities, Healing School, InnerCity Mission, Loveworld TV, plus custom campaigns).
  - Linking transactions to members, cells, or external givers.
  - Batch locking and verification for service-linked entries.
  - Simple reconciliation and reporting views.
- **Cells Portal** (`cells.<zone>.ce.church`)
  - Cell leader reporting (attendance, first timers, new converts, testimonies, offerings, meeting type).
  - Linking members to cells and designating leadership.
  - Escalation flags for pastoral follow-up.
- **Reports Portal** (`reports.<zone>.ce.church`)
  - Dashboards and KPIs covering membership, finance, cells, and attendance.
  - Drill-down and roll-up across the organisational hierarchy (region, zone, group, church, cell).
  - CSV, Excel, and PDF exports of standard reports.
  - Scheduled email reports.

### Out of Scope

- Direct payment processing — giving is recorded, not transacted.
- Event scheduling, volunteer management, and media handling.
- Advanced CRM features (mass messaging, marketing automation).

- Integration with external accounting software (exports only in MVP).
- Mobile native applications — all portals will be responsive web apps.

## 5 Roles & Permissions

This section defines the access-control model for the platform. It specifies the organisational hierarchy, scope semantics, role definitions, permission catalogue, delegation rules, high-risk actions (dual control), audit requirements, and reference enforcement patterns (API checks and Postgres RLS).

### 5.1 Access Control Model Overview

- **RBAC with Scopes:** A role grants a set of permissions (the “what”). Each user also has one or more *organisational assignments* that bound their access (the “where”): a specific org unit and a scope type.
- **Org Hierarchy:** `region`  $\rightarrow$  `zone`  $\rightarrow$  `group`  $\rightarrow$  `church`  $\rightarrow$  `outreach`. Cells belong to a church (not an org unit type).
- **Enforcement:** Every request must satisfy both:
  1. the user holds the *required permission*, and
  2. the *target record’s org location* lies within at least one of the user’s assigned scopes.

### 5.2 Organisational Entities & Scope Semantics

#### Org Units

- Table: `org_units(id, name, type, parent_id)` where `type`  $\in$  `{region, zone, group, church, outreach}`.

#### Assignments

- Table: `org_assignments(id, user_id, org_unit_id, role_id, scope_type)` where `scope_type`  $\in$  `{self, subtree, custom_set}`.
- For `custom_set`, a join table `org_assignment_units(assignment_id, org_unit_id)` enumerates allowed units.

**Record Localisation** Every domain record carries `tenant_id` and an org reference:

- **People/Members:** `people.org_unit_id` (their primary church).
- **Services & Attendance:** `services.org_unit_id`.
- **Cells & Cell Reports:** `cells.org_unit_id` (church), `cell_reports.cell_id`.
- **Finance Entries:** `finance_entries.org_unit_id` (church owning the record); may also link `service_id` or `cell_id`.

## 5.3 Roles (Templates)

Roles are templates; they can be adjusted by the Zonal Pastor (Tenant Admin) if needed. Default roles:

- **Zonal Pastor (Tenant Admin):** Full zone oversight; manage users, roles, lookups, and all reports (scope typically `subtree` at zone level).
- **Group Pastor:** Manage churches within assigned group(s); create church pastors/coordinators; view group reports.
- **Church Pastor / Coordinator:** Manage their local church and assign portal-specific users; limited reporting.
- **Church Administrator:** Registry portal; capture and update people, first-timers, departments, attendance for assigned churches.
- **Finance Officer:** Finance portal; create/verify/lock batches and entries; view finance reports for assigned churches.
- **Cell Leader:** Cells portal; submit cell reports and view own cell history; no wider access.
- **Reports Viewer (Pastoral):** Read-only dashboards/reports within assigned scope.
- **Technical Lead (Optional):** For zone-level setup; manage lookups (funds, partnership arms), not PII beyond necessity.

## 5.4 Permission Catalogue (Dot-Notation)

Permissions are grouped by portal. CRUD verbs are explicit. “Export” and “Approve/Lock” are separate privileges.

### Registry

- `registry.people.read/create/update/delete`
- `registry.people.merge` (deduplicate/merge)
- `registry.people.export`
- `registry.firsttimers.read/create/update/delete/export`
- `registry.attendance.read/create/update/delete/export`
- `registry.departments.read/create/update/delete`
- `registry.cells.assign` (assign member to cell; convenience overlap)
- `registry.admin_notes.read/create/update/delete` (pastoral notes; restricted)

## Finance

- `finance.entries.read/create/update/delete/export`
- `finance.batches.read/create/update/delete`
- `finance.batches.lock` (dual control)
- `finance.batches.unlock` (dual control; higher bar)
- `finance.verify` (mark entries verified/reconciled)
- `finance.lookups.manage` (funds, partnership arms, payment methods)

## Cells

- `cells.reports.read/create/update/delete/export`
- `cells.manage` (create/update cells, leaders, venues)
- `cells.reports.approve` (optional approval workflow)

## Reports

- `reports.view` (dashboards, drill-down)
- `reports.export` (CSV/Excel/PDF)
- `reports.schedule` (scheduled email reports)

## System / Administration

- `system.users.create/disable/reset_password`
- `system.roles.assign` (assign role templates to users)
- `system.scopes.assign` (assign org scopes to users)
- `system.audit.view`
- `system.settings.manage` (zone settings, consent texts)
- `system.exports.full_pii` (extra-protected export)

## 5.5 Permission Matrix (Default)

Role	Key Permissions (all within assigned scope)
Zonal Pastor (Tenant Admin)	<code>reports.view/export/schedule</code> , <code>system.users.create</code> , <code>system.roles.assign</code> , <code>system.scopes.assign</code> , <code>system.audit.view</code> , <code>system.settings.manage</code> , all Registry/Finance/Cells read, Finance lock/unlock (dual-control partner required), limited <code>system.exports.full_pii</code> .



Group Pastor	<code>reports.view/export</code> , Registry read, Cells read, Finance read, <code>system.users.create</code> (within group), <code>system.roles.assign</code> (church-level roles), <code>system.scopes.assign</code> (within group). No finance lock/unlock by default (optional).
Church Pastor / Coordinator	<code>reports.view/export</code> (church), Registry read/create/update, Cells manage (church), Finance read, <code>finance.verify</code> (optional), <code>system.users.create</code> (portal users for church), <code>system.roles.assign</code> (portal roles).
Church Administrator	<code>registry.people.*</code> (no delete/merge unless granted), <code>registry.firsttimers.*</code> , <code>registry.attendance.*</code> , <code>registry.departments.*</code> . No reports export beyond Registry, no Finance.
Finance Officer	<code>finance.entries.*</code> , <code>finance.verify</code> , <code>finance.batches.*</code> , <code>finance.batches.lock</code> (dual), read-only <code>reports.view</code> for Finance, <code>finance.entries.export</code> . No Registry/Cells write.
Cell Leader	<code>cells.reports.create/update</code> (own cell only), <code>cells.reports.read</code> (own cell), no exports by default.
Reports Viewer (Pastoral)	<code>reports.view/export</code> (assigned scope only).
Technical Lead (Optional)	<code>system.settings.manage</code> , <code>finance.lookups.manage</code> , read-only <code>reports.view</code> , no PII exports.

## 5.6 Delegation & Provisioning Rules

- **Top-down provisioning:** Zonal Pastor can create Group Pastors and assign group scopes. Group Pastors can create Church Pastors/Coordinators within their group(s). Church Pastors can create portal users (Church Administrator, Finance Officer, Cell Leader) for their churches/cells.
- **Scope assignment:** Each user may have multiple assignments (e.g., two churches). The union of assignments defines *effective scope*.
- **Impersonation:** Not allowed in MVP. If added later, must be audited and time-limited.

## 5.7 High-Risk Actions (Dual Control & Step-Up Auth)

- **Finance Batch Lock/Unlock:** Requires *two distinct users*:
  1. Locker: holds `finance.batches.lock`
  2. Approver: holds `finance.verify` or `finance.batches.lock` and is not the Locker

Unlock requires `finance.batches.unlock` + approval by Zonal/Group Pastor or designated approver.

- **Full-PII Export:** Requires `system.exports.full_pii` and step-up 2FA confirmation.
- **Delete/Merge People:** `registry.people.delete` and `registry.people.merge` gated by step-up 2FA and audit reason.

## 5.8 PII & Data Masking

- **Masking by Portal:** Finance screens show masked contact info by default (e.g., `joexample.com`, `085***321`); full reveal requires `registry.people.read` and explicit click-to-reveal with audit trail.
- **Pastoral Notes:** `registry.admin.notes.*` restricted to pastoral roles; excluded from standard exports.

## 5.9 Audit Logging (Immutable)

Audit *all writes* and the following reads/exports:

- Login, logout, 2FA enable/disable; password resets.
- **Reads:** click-to-reveal of masked PII; full-PII exports; audit-log access.
- **Writes:** create/update/delete across Registry/Finance/Cells; merge people; batch lock/unlock; verify/reconcile; role/scope assignments; lookup changes.

Audit records: (`actor_id`, `action`, `entity_type`, `entity_id`, `before_json`, `after_json`, `ip`, `user_agent`, `occurred_at`).

## 5.10 Session Security

- **2FA:** Required for Zonal/Group/Church Pastors and Finance Officers.
- **Session Timeout:** 30 min idle (portal configurable), absolute 12 h.
- **Password Policy:** 12+ chars, breach-checked; rotation not required but compromised detection enforced.
- **SSO (Optional):** Google/Microsoft SSO may be enabled at tenant level.

## 5.11 API Enforcement Pattern

Every API endpoint declares a required permission *p* and resolves a target `org.unit_id`. The request is allowed iff the user holds *p* and at least one assignment covers the target org.

## Effective-Access Pseudocode

```
function can(user, permission, target_org_id):
    if not user.permissions.contains(permission):
        return false
    for a in user.assignments:
        if a.scope_type == 'self' and a.org_id == target_org_id:
            return true
        if a.scope_type == 'subtree' and is_descendant(target_org_id, a.org_id):
            return true
        if a.scope_type == 'custom_set' and target_org_id in a.allowed_org_ids:
            return true
    return false
```

## 5.12 Database Enforcement (Postgres RLS)

All tenant tables include `tenant_id` and an org reference column. Row-Level Security (RLS) policies enforce per-row visibility.

### Helper Functions (Sketch)

```
-- Returns true if current_user has any assignment covering target org
CREATE FUNCTION has_org_access(target_org_id uuid) RETURNS boolean AS $$
    SELECT EXISTS (
        SELECT 1
        FROM org_assignments oa
        JOIN users u ON u.id = oa.user_id
        WHERE u.id = current_setting('app.user_id')::uuid
        AND (
            (oa.scope_type = 'self' AND oa.org_unit_id = target_org_id) OR
            (oa.scope_type = 'subtree' AND
              is_descendant_org(target_org_id, oa.org_unit_id)) OR
            (oa.scope_type = 'custom_set' AND EXISTS (
                SELECT 1 FROM org_assignment_units oau
                WHERE oau.assignment_id = oa.id AND oau.org_unit_id = target_org_id))
        )
    );
$$ LANGUAGE sql STABLE;

-- Permissions are cached into current_setting('app.perms') as a text[] at login.
CREATE FUNCTION has_perm(p text) RETURNS boolean AS $$
    SELECT p = ANY (current_setting('app.perms')::text[]);
$$ LANGUAGE sql STABLE;
```

### Example Policy

```
ALTER TABLE finance_entries ENABLE ROW LEVEL SECURITY;

CREATE POLICY finance_read_policy ON finance_entries
```

```

FOR SELECT
USING (
    current_setting('app.tenant_id')::uuid = tenant_id
    AND has_perm('finance.entries.read')
    AND has_org_access(org_unit_id)
);

CREATE POLICY finance_write_policy ON finance_entries
FOR INSERT WITH CHECK (
    current_setting('app.tenant_id')::uuid = tenant_id
    AND has_perm('finance.entries.create')
    AND has_org_access(org_unit_id)
);

-- Similarly for UPDATE/DELETE with corresponding permissions.

```

### 5.13 Field-Level Constraints

- **Portal Affinity:** Even if a user holds a broad role, UI and API endpoints must only expose data relevant to the current portal unless the user holds `reports.view` or system-level privileges.
- **Cell Leader Boundaries:** Cell Leaders can only access `cells.reports.*` for their own `cell_id`; API validates `author/owner`.

### 5.14 Lookups & Managed Lists

- **Funds and Partnership Arms** are managed by users with `finance.lookups.manage`. Custom campaigns are added with active date ranges. Historical entries preserve referential integrity.
- **Payment Methods** list (e.g., cash, KingsPay, bank transfer, POS, cheque) is also managed here.

### 5.15 Examples (Worked)

#### Example 1: Finance Officer locks Sunday batch

1. Officer has: `finance.entries.*,finance.batches.read/create/update,finance.batches.lock`
2. Scope: `subtree` for Church A.
3. Action: Lock requires counter-approval by another user (Finance Officer or Pastor) with overlapping scope. Both events are audited.

#### Example 2: Group Pastor views reports for entire group

1. Holds: `reports.view/export`.
2. Scope: `subtree` at Group G.

3. Dashboards include all churches in G; drill-down stops at church/cell records within G.

### Example 3: Cell Leader submits report

1. Holds: `cells.reports.create/update`.
2. API validates `cell_id` is the leader's assigned cell and the cell's `org_unit_id` is within scope (church).

## 5.16 Permissions Matrix (CSV for Reference)

For importer/testing, an equivalent CSV (not part of the PDF) can define role-to-permission mappings for seed data. Example columns:

`role_name, permission, default_granted[bool]`.

## 5.17 Operational Controls

- **Exports:** All exports include a footer with generator, timestamp, and scope; full-PII exports require elevated permission and step-up 2FA.
- **Change Windows:** Unlocking a locked batch is time-limited (e.g., within 7 days) unless Zonal Pastor approves an exception.
- **Rate Limits:** Apply rate limits to high-volume reads/exports per user to protect performance.

# 6 Functional Requirements

This section defines the functional requirements for the Christ Embassy Ireland Zone Church Reporting Platform. Requirements are grouped by portal (Registry, Finance, Cells, Reports) and describe system behaviour, data relationships, and user interactions. Each requirement is uniquely identified for reference.

## 6.1 Registry Portal (`registry.<zone>.ce.church`)

**Purpose** Provides authorised church administrators with facilities to manage member records, first-timers, service attendance, and departmental assignments.

### Core Entities

- **Member (Person):** uniquely identified individual with demographic and contact details, membership status, and church affiliation.
- **First Timer:** a visitor who attended for the first time and may later be linked to a Member record.
- **Service Attendance:** a record of attendance counts for a specific service date and church.
- **Department:** organisational unit within a church with members and leaders.

## Requirements

- **FR-R1** — The system shall allow Church Administrators and authorised pastors to create, edit, and search Member records containing: name, aliases, gender, date of birth, contact information, address, town, county, Eircode, marital status, foundation school completion, baptism date, membership status, and associated cell.
- **FR-R2** — The system shall validate mandatory fields (first name, last name, gender, contact number or email, and assigned church).
- **FR-R3** — Each Member record shall have a unique internal identifier and audit trail of changes.
- **FR-R4** — Church Administrators shall be able to register first-timers and associate them with the service they attended, capturing name, contact, inviter/source, and notes.
- **FR-R5** — The system shall maintain a pipeline of first-timers with statuses: New → Contacted → Returned → Member.
- **FR-R6** — The system shall store attendance summaries for each service, including men, women, teens, kids, total attendance, first timers, new converts, and notes. Each attendance entry is linked to a Service object (`service_id`) identifying the date, time, and church.
- **FR-R7** — The system shall provide validation to ensure only one attendance record per service per church.
- **FR-R8** — Departmental assignments shall link Members to one or more departments and optionally mark a leader.
- **FR-R9** — Registry users shall be able to export membership or attendance data as CSV/Excel, limited by their scope.
- **FR-R10** — All Registry actions (create, update, delete, merge) shall be recorded in the audit log.

## 6.2 Finance Portal (`finance.<zone>.ce.church`)

**Purpose** Captures and verifies records of all forms of monetary giving for accounting and reporting purposes. The system does *not* process payments but maintains a structured record of financial transactions and partnerships.

### Core Entities

- **Fund:** category of giving (tithe, offering, seed, first fruit, partnership, etc.).
- **Partnership Arm:** specific ministry initiative such as Rhapsody of Realities, The Healing School, InnerCity Mission for Children, Loveworld Television Ministry, or any custom campaign.
- **Finance Entry:** individual record of a contribution linked to a member, cell, or external giver.
- **Batch:** collection of finance entries for a service; may be locked and verified.

## Requirements

- **FR-F1** — Finance Officers shall be able to record new finance entries with: transaction date, amount, fund, partnership arm, payment method (cash, KingsPay, bank transfer, cheque, POS, other), reference, member/cell linkage, and comment.
- **FR-F2** — Each finance entry shall reference the associated service where applicable (`service_id`).
- **FR-F3** — The system shall permit association of entries to either a Member (`person_id`), a Cell (`cell_id`), or an external entity (text name).
- **FR-F4** — Entries shall include a `verified_status` field with states: Draft → Verified → Reconciled → Locked.
- **FR-F5** — Each batch shall be uniquely identified by (`church`, `service_date`, `service_name`) and can be locked only after dual approval.
- **FR-F6** — Locked batches are immutable; unlock requires dual authorisation and is fully audited.
- **FR-F7** — The system shall allow recording of partnerships with amount, start date, frequency (weekly, monthly, quarterly, annual), and fulfilment percentage.
- **FR-F8** — Finance users shall be able to generate summaries per fund, partnership arm, church, or service.
- **FR-F9** — All finance data exports must include generation timestamp, scope, and user metadata.
- **FR-F10** — Only authorised users can edit or delete verified entries, and all such actions require justification text.

### 6.3 Cells Portal (`cells.<zone>.ce.church`)

**Purpose** Allows Cell Leaders to submit weekly cell reports and track their own cell activities, while allowing pastors to monitor participation and outreach.

#### Core Entities

- **Cell:** a small fellowship group linked to a church with leader, assistant, meeting day, time, and venue.
- **Cell Report:** a weekly or ad-hoc meeting report containing attendance counts, first timers, new converts, testimonies, offerings, and meeting type.

## Requirements

- **FR-C1** — Cell Leaders shall be able to submit cell reports for their assigned cell only.
- **FR-C2** — Each cell report shall include date, time, meeting type, attendance, first timers, new converts, testimonies, and offerings total.

- **FR-C3** — Each cell report shall automatically reference its parent cell (`cell_id`) and the church through that cell's `org_unit_id`.
- **FR-C4** — The system shall validate that only one report per cell per date can be submitted.
- **FR-C5** — Cell offerings shall be automatically included in the Finance reports under the corresponding fund when applicable.
- **FR-C6** — Pastors may view submission status and overdue reports across cells they oversee.
- **FR-C7** — The system shall support optional approval workflow for cell reports (submitted → reviewed → approved).

## 6.4 Reports Portal (`reports.<zone>.ce.church`)

**Purpose** Provides zonal, group, and local pastors with dashboards and drill-down reports spanning membership, finance, attendance, and cell activities.

### Requirements

- **FR-P1** — The system shall display KPIs and charts based on the user's scope:
  - Attendance trends (weekly, monthly, demographic splits)
  - First-timer assimilation funnel
  - Foundation school and baptism progress
  - Department participation rate
  - Giving trends by fund and partnership arm
  - Per-capita giving (total giving / average attendance)
  - Cell reporting compliance and outreach outcomes
- **FR-P2** — Dashboards shall provide drill-down from zone → group → church → cell → individual.
- **FR-P3** — Reports shall support filters by date range, fund, church, cell, and member status.
- **FR-P4** — Reports shall be exportable as CSV, Excel, or PDF.
- **FR-P5** — Users with appropriate permissions may schedule recurring email reports.

## 6.5 Data Import & Migration

**FR-I1** The system shall support import of legacy spreadsheets (Members, Services, Attendance, Finance, Cells, Cell Reports) through a guided mapping interface with preview and validation.



**FR-I2** Validation errors shall be reported with line numbers and descriptions. Users may download a corrected template for re-upload.

**FR-I3** Imports shall be performed per portal according to role permissions; only Zonal or Group Pastors can perform zone-wide imports.

## 6.6 Audit & Logging Integration

**FR-A1** All create, update, delete, merge, lock/unlock, verify, and export actions shall be written to an immutable audit log with timestamp, actor, entity type, entity id, and IP address.

**FR-A2** System shall expose audit entries to users with `system.audit.view` permission filtered by their scope.

**FR-A3** Audit logs shall be retained for a minimum of seven years and included in backups.

## 7 Non-Functional Requirements

This section defines the non-functional requirements (NFRs) that govern the quality attributes of the Christ Embassy Ireland Zone Church Reporting Platform. These requirements ensure that the system operates securely, reliably, and efficiently across all sub-domains and user roles.

### 7.1 Security

- **NFR-S1** — All user authentication and session management shall follow industry-standard security practices using HTTPS, salted password hashing (Argon2 or bcrypt), and optional Single Sign-On (SSO) for pastors.
- **NFR-S2** — Role-Based Access Control (RBAC) and organisational scopes shall be enforced both at the API layer and the database layer via Row-Level Security (RLS).
- **NFR-S3** — All data in transit shall be encrypted using TLS 1.2 or higher; data at rest shall be encrypted using AES-256 or equivalent.
- **NFR-S4** — Multi-factor authentication (2FA) shall be required for pastors, finance officers, and users with elevated permissions.
- **NFR-S5** — The platform shall maintain immutable audit logs for all sensitive actions including data exports, deletions, financial verifications, and role assignments.
- **NFR-S6** — Session timeouts shall occur after 30 minutes of inactivity or 12 hours of continuous login, whichever comes first.
- **NFR-S7** — Password reset tokens shall expire within 15 minutes and be single-use.

## 7.2 Privacy & Compliance

- **NFR-P1** — The system shall comply with the General Data Protection Regulation (GDPR) as applicable in Ireland and the EU.
- **NFR-P2** — All personally identifiable information (PII) shall be encrypted at rest and masked where not essential to a user’s role.
- **NFR-P3** — Consent for storing contact information and follow-up communication shall be explicitly captured during registration.
- **NFR-P4** — Users with `system.exports.full_pii` must complete step-up 2FA confirmation before export.
- **NFR-P5** — Members shall have the right to request data correction or deletion, subject to church policy and record-keeping obligations.
- **NFR-P6** — Data retention periods shall be configurable, with a default of 7 years for audit and financial records.

## 7.3 Reliability & Availability

- **NFR-R1** — The system shall achieve a minimum of 99.9% uptime, excluding scheduled maintenance.
- **NFR-R2** — The platform shall degrade gracefully in the event of partial component failure (e.g., temporary loss of reporting service shall not affect data entry in Registry or Finance portals).
- **NFR-R3** — All user operations shall be idempotent where applicable (e.g., duplicate uploads or repeated submissions shall not produce duplicate records).
- **NFR-R4** — The system shall perform daily backups of the database and retain rolling backups for a minimum of 30 days.
- **NFR-R5** — Backups shall be encrypted and stored in geographically redundant locations.
- **NFR-R6** — Recovery Point Objective (RPO): 1 hour; Recovery Time Objective (RTO): 2 hours.
- **NFR-R7** — A backup restore test shall be successfully executed at least once per quarter.

## 7.4 Performance & Scalability

- **NFR-PS1** — API response times shall not exceed 300 ms for 95% of requests under normal load (P95 latency).
- **NFR-PS2** — The system shall support at least 1,000 concurrent active users within a single zone tenant.

- **NFR-PS3** — Data queries and reports shall use indexed aggregations to return results within 2 seconds for standard filters.
- **NFR-PS4** — The system shall scale horizontally via container replication (Docker/Kubernetes) and vertically via resource provisioning.
- **NFR-PS5** — Background jobs such as imports, exports, and scheduled report generation shall be handled asynchronously via a job queue (e.g., Celery or RQ) to avoid blocking requests.
- **NFR-PS6** — File uploads and large exports shall use presigned URLs or streamed downloads to optimise throughput.

## 7.5 Maintainability & Extensibility

- **NFR-M1** — The codebase shall follow a modular architecture, separating backend services, shared components, and frontend portals.
- **NFR-M2** — The backend shall expose a documented REST API with OpenAPI/Swagger specification.
- **NFR-M3** — All configuration values (e.g., database URIs, email credentials) shall be injected via environment variables or secure parameter storage.
- **NFR-M4** — New portals or modules (e.g., Events, Outreach) can be added without modifying existing schemas or permissions, using the same tenant-based architecture.
- **NFR-M5** — Automated testing (unit, integration, and API-level) shall be maintained with a minimum 80% coverage target.
- **NFR-M6** — Continuous Integration/Deployment (CI/CD) shall be implemented with code linting, schema migrations, and automated rollback.
- **NFR-M7** — All dependencies shall be pinned to versions and checked for security vulnerabilities regularly.

## 7.6 Usability & Accessibility

- **NFR-U1** — All portals shall be responsive and usable on desktop, tablet, and mobile screens.
- **NFR-U2** — Interfaces shall use clear typography and layout with consistent navigation across portals.
- **NFR-U3** — Data-entry forms shall include validation feedback and contextual error messages.
- **NFR-U4** — Colour contrast and text size shall meet WCAG 2.1 AA accessibility guidelines.
- **NFR-U5** — The system shall provide user feedback for all asynchronous operations (e.g., import progress, batch locking).

## 7.7 Monitoring & Observability

- **NFR-O1** — Application logs shall be structured (JSON) and centralised for analysis.
- **NFR-O2** — Metrics such as API latency, job queue depth, and failed logins shall be captured and visualised on dashboards.
- **NFR-O3** — Alerts shall be configured for availability, performance degradation, and authentication anomalies.
- **NFR-O4** — Audit and monitoring data shall be retained for at least 12 months.

## 8 KPIs and Success Metrics

This section defines the Key Performance Indicators (KPIs) and success metrics for the Christ Embassy Ireland Zone Church Reporting Platform. These metrics will guide ongoing performance monitoring and confirm whether the system achieves its intended operational and technical objectives.

### 8.1 Overview

KPIs are grouped into four primary domains aligned with the platform's portals: **Registry**, **Finance**, **Cells**, and **Reports**. Additionally, cross-cutting technical KPIs measure data integrity, uptime, and user engagement.

### 8.2 Registry KPIs

- **KPI-R1: Membership Data Completeness** — percentage of member records with all required demographic and contact fields completed. *Target:  $\geq 95\%$ .*
- **KPI-R2: Duplicate Records Rate** — percentage of duplicate or merged records over total members. *Target:  $\leq 2\%$ .*
- **KPI-R3: First-Timer Registration Timeliness** — proportion of first-timer entries captured within 24 hours of service attendance. *Target:  $\geq 90\%$ .*
- **KPI-R4: First-Timer Assimilation Rate** — percentage of first-timers who transition to full membership within 60 days. *Target:  $\geq 70\%$ .*
- **KPI-R5: Attendance Reporting Compliance** — proportion of scheduled services with attendance data recorded and approved. *Target: 100%.*
- **KPI-R6: Foundation School Completion Rate** — percentage of members who complete foundation school within one year of joining. *Target:  $\geq 80\%$ .*
- **KPI-R7: Baptism Tracking Rate** — proportion of baptized members with corresponding date recorded in the system. *Target: 100%.*

### 8.3 Finance KPIs

- **KPI-F1: Data Entry Timeliness** — percentage of service offerings entered within 48 hours of collection. *Target:  $\geq 95\%$ .*
- **KPI-F2: Verification Timeliness** — percentage of batches verified and locked within one week of entry. *Target:  $\geq 95\%$ .*
- **KPI-F3: Reconciliation Accuracy** — difference between total logged offerings and verified deposits per reporting period. *Target:  $\leq 1\%$  variance.*
- **KPI-F4: Partnership Fulfilment Rate** — proportion of active partners meeting or exceeding their pledged amounts. *Target:  $\geq 85\%$ .*
- **KPI-F5: Fund Mix Visibility** — availability of clear fund distribution reports (tithes, offerings, partnerships, first fruits). *Target: 100% availability.*
- **KPI-F6: Cell Offering Reconciliation Rate** — proportion of cell-reported offerings successfully reflected in finance summaries. *Target:  $\geq 95\%$ .*

### 8.4 Cells KPIs

- **KPI-C1: Reporting Compliance** — percentage of active cells that submit at least one report per week. *Target:  $\geq 95\%$ .*
- **KPI-C2: Attendance Accuracy** — variance between average cell attendance and verified church service attendance ratios. *Target:  $\leq 10\%$ .*
- **KPI-C3: Outreach Effectiveness** — number of first-timers and new converts reported per 100 active members. *Target: growth trend  $\geq +10\%$  quarter-over-quarter.*
- **KPI-C4: Testimony Capture Rate** — proportion of cell reports containing at least one recorded testimony. *Target:  $\geq 50\%$ .*
- **KPI-C5: Leader Submission Timeliness** — percentage of cell reports submitted within 24 hours of meeting. *Target:  $\geq 90\%$ .*

### 8.5 Reports & Oversight KPIs

- **KPI-P1: Dashboard Availability** — uptime of the reports portal and dashboard services. *Target: 99.9%.*
- **KPI-P2: Report Generation Speed** — average generation time for standard dashboards and exports. *Target:  $\leq 2$  seconds for dashboards,  $\leq 10$  seconds for exports.*
- **KPI-P3: Hierarchical Accuracy** — correctness of aggregated metrics when drilling down from zone to church level. *Target: 100% consistency.*
- **KPI-P4: Data Refresh Latency** — delay between entry or update in sub-portals and reflection in dashboards. *Target:  $\leq 5$  minutes.*
- **KPI-P5: Scheduled Report Reliability** — success rate of automated scheduled report deliveries. *Target:  $\geq 99\%$ .*

## 8.6 Cross-Domain Technical KPIs

- **KPI-T1: System Uptime** — total platform uptime excluding maintenance windows. *Target:* 99.9%.
- **KPI-T2: API Latency** — median and 95th percentile latency of API endpoints under nominal load. *Target:*  $P95 \leq 300$  ms.
- **KPI-T3: Error Rate** — percentage of failed API requests (5xx or unhandled 4xx) over total requests. *Target:*  $\leq 0.5\%$ .
- **KPI-T4: Data Integrity Score** — proportion of records passing consistency checks and referential integrity validation. *Target:*  $\geq 99.5\%$ .
- **KPI-T5: Backup Validation Success** — successful restore verification rate per quarter. *Target:* 100%.
- **KPI-T6: Audit Log Coverage** — proportion of sensitive actions successfully captured in audit logs. *Target:* 100%.

## 8.7 Measurement & Monitoring

- KPI metrics shall be automatically collected and displayed in the administrative reporting dashboard for zone leadership and technical staff.
- Historical KPI data shall be stored for at least 12 months for trend analysis.
- Alerts shall be configured for threshold breaches (e.g., missing service reports, delayed batch verification).
- KPI data shall also be exportable in CSV format for external analysis.

# 9 System Architecture Overview

This section describes the architectural design of the Christ Embassy Ireland Zone Church Reporting Platform. It defines the overall system topology, core modules, data flow, and design principles that ensure scalability, reliability, and maintainability.

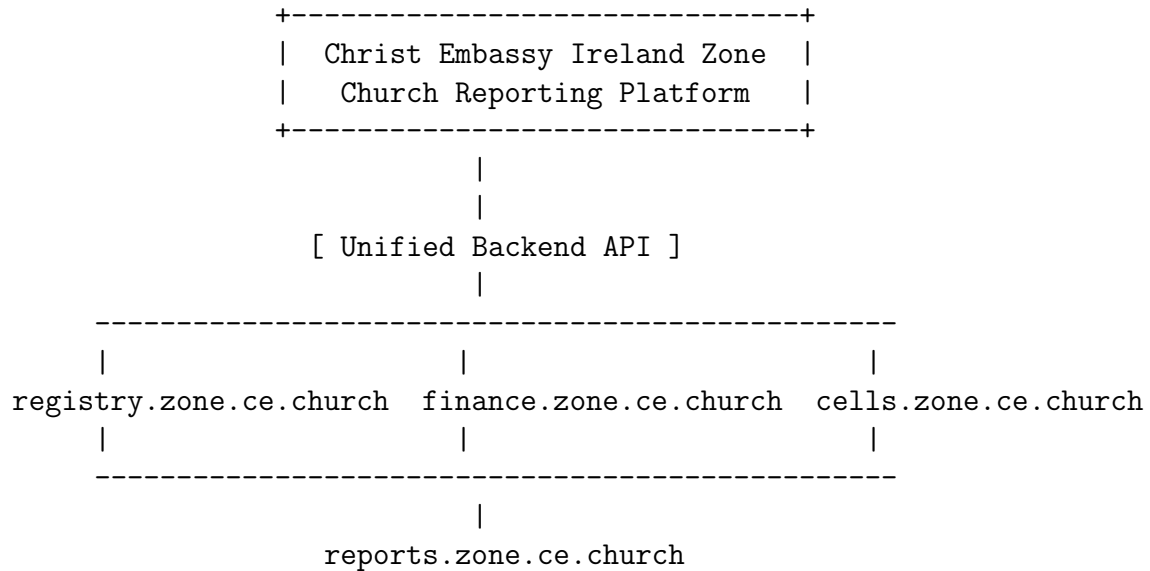
## 9.1 Architecture Summary

The platform follows a **modular monolith** architecture, with a unified backend serving four specialised web portals:

- **Registry Portal** — membership, first-timers, attendance, departments.
- **Finance Portal** — offerings, tithes, partnerships, reconciliation.
- **Cells Portal** — cell reports, attendance, testimonies, offerings.
- **Reports Portal** — dashboards and analytics for all zones, groups, churches, and cells.

Each portal is deployed under its own sub-domain and connects to a single, multi-tenant backend via authenticated API endpoints. The architecture supports horizontal scaling and future evolution into microservices without significant redesign.

## 9.2 High-Level Topology



Each frontend communicates with the same backend via REST APIs protected by role-based authentication. All data is stored in a shared PostgreSQL database with strict tenant and scope enforcement.

## 9.3 Backend Components

### Technology Stack

- **Framework:** Python & FastAPI
- **Database:** PostgreSQL with Row-Level Security (RLS)
- **Cache / Queue:** Redis
- **Storage:** S3-compatible object store for exports and receipts
- **Worker:** Celery or RQ for background jobs (imports, scheduled reports, roll-ups)
- **Frontend:** Next.js + React with Tailwind CSS and shadcn/ui
- **Deployment:** Docker containers orchestrated on ECS, Fly.io, or Kubernetes

### Backend Modules

1. **Auth & Users** — Authentication, 2FA, RBAC, and session management. Hydrates user permissions and scope context into API sessions.
2. **Registry** — Manages members, first-timers, departments, services, and attendance. One attendance record per service per church.

3. **Finance** — Handles funds, partnership arms, finance entries, batches, and verification workflows.
4. **Cells** — Records cells, leadership, meeting reports, testimonies, and offerings (mirrored into Finance for aggregation).
5. **Reports** — Serves pre-aggregated dashboards and analytics via materialized views and roll-up tables.
6. **Imports/Exports** — Processes legacy data uploads and report exports asynchronously.
7. **Audit** — Immutable audit logging and query interface.

## 9.4 Data and Tenancy Model

- All tables include `tenant_id` and `org_unit_id` fields.
- Organisational hierarchy: `region` → `zone` → `group` → `church` → `outreach`.
- Cells belong to a specific church and inherit its organisational scope.
- Row-Level Security (RLS) enforces visibility by tenant and assigned organisational scope (self, subtree, or custom set).

### Key Tables

- `people`, `memberships`, `first_timers`, `services`, `attendance`
- `departments`, `department_roles`
- `funds`, `partnership_arms`, `finance_entries`, `batches`, `partnerships`
- `cells`, `cell_reports`
- `audit_logs`, `imports`, `exports`

## 9.5 Data Flows

### Attendance Recording

1. Church Administrator records attendance for a `service_id` tied to a church and date.
2. The worker updates summary tables and dashboards in the Reports module.

### Cell Reports and Offerings

1. Cell Leader submits a `cell_report` with attendance, first-timers, and offerings.
2. The worker mirrors the offerings amount into a linked `finance_entry` (source: `cell_report`), tagged by `cell_id` for consolidated finance reporting.



## Finance Entry and Batch Verification

1. Finance Officer records service offerings, tithes, and partnerships.
2. When reviewed, a second authorised user locks the batch. Once locked, it becomes immutable and triggers reconciliation updates.

## Data Import

1. User uploads a spreadsheet through an import interface.
2. Mapping UI validates columns and previews data.
3. Import job is queued for processing and audit-tracked.

## 9.6 Reporting Architecture

- Pre-aggregated summaries are stored in `summary_*` tables and materialized views for high-performance dashboards.
- Updates are triggered by event hooks (e.g., new attendance record, verified batch, cell report submission).
- Reports use filters for date range, fund, partnership arm, and organisational scope.
- Aggregations are consistent across hierarchy levels (zone → group → church → cell).

## 9.7 Scalability & Deployment

- Stateless application containers; scale horizontally by increasing replicas.
- Shared PostgreSQL instance; read replicas for analytics load.
- Redis used for job queue, caching, and rate limiting.
- Static assets and exports served via CDN or object storage.
- Infrastructure defined using Terraform for reproducible deployment.

## 9.8 Observability

- Structured JSON logging with correlation IDs per request.
- Metrics: request latency, queue depth, cache hit ratio, failed logins.
- Alerting for downtime, slow queries, job failures, and high error rates.
- Dashboards for uptime, throughput, and data integrity monitoring.

## 9.9 Evolution Path

The architecture is designed to evolve gracefully:

- Reports may be separated into a dedicated analytics service backed by a read-replica or OLAP database (e.g., ClickHouse or BigQuery).
- An event bus (e.g., Redis Streams or NATS) can be introduced to manage asynchronous updates between Registry, Finance, and Reports modules.
- A federated search engine (e.g., OpenSearch) may be added for member and transaction lookup if performance requires.

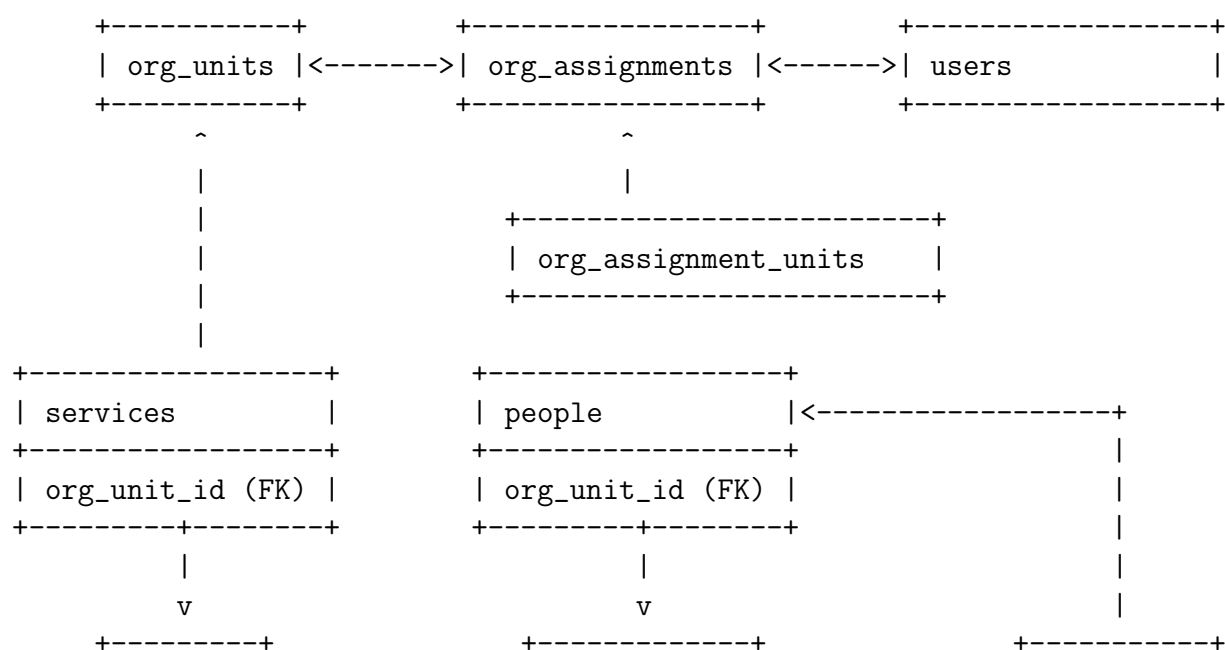
## 9.10 Key Design Decisions

- Modular monolith selected for simplicity, lower cost, and easier maintainability.
- Data aggregation through materialized views for near real-time reports.
- Unified access control model across all portals for consistent permissions enforcement.
- Event-driven updates between modules to maintain data consistency.

## 10 Data Model

This section defines the core entities, relationships, and constraints for the Christ Embassy Ireland Zone Church Reporting Platform. All tables include `tenant_id` UUID NOT NULL and standard audit columns (`created_at`, `created_by`, `updated_at`, `updated_by`). Organisational scope is enforced with `org_unit_id` and Postgres Row-Level Security.

## 10.1 High-level ERD (ASCII)





- **org\_assignment\_units:** assignment\_id FK, org\_unit\_id FK (for custom sets).

## 10.3 Registry Domain

### people

- **Columns:** id PK, tenant\_id, org\_unit\_id FK(church), member\_code UNIQUE per tenant, names (title, first, last, alias), dob DATE, gender ENUM, contact (email, phone), address (line1, line2, town, county, eircode), marital\_status, consent flags.
- **Indexes:** (tenant\_id, org\_unit\_id), trigram on (first\_name, last\_name, email, phone) for dedupe/search.

### memberships

- **Columns:** person\_id PK/FK, status ENUM(visitor, regular, member, partner), join\_date DATE, foundation\_completed BOOL, baptism\_date DATE NULL, cell\_id FK NULL.

### departments, department\_roles

- **departments:** id PK, tenant\_id, org\_unit\_id FK, name, status.
- **department\_roles:** id PK, dept\_id FK, person\_id FK, role ENUM(leader, member), start\_date, end\_date.

### services

- **Columns:** id PK, tenant\_id, org\_unit\_id FK, name ENUM(Sunday, Midweek, Special) or TEXT, service\_date DATE, service\_time TIME.
- **Uniqueness:** UNIQUE(tenant\_id, org\_unit\_id, service\_date, name).

### attendance

- **Columns:** id PK, tenant\_id, service\_id FK, men\_count, women\_count, teens\_count, kids\_count, first\_timers\_count, new\_converters\_count, total\_attendance, notes.
- **Uniqueness:** UNIQUE(tenant\_id, service\_id) (one attendance per service).
- **Note:** service carries org and date; linking via service\_id keeps attendance *service-aware*.

### first\_timers

- **Columns:** id PK, tenant\_id, person\_id FK NULL, service\_id FK, source/inviter, status ENUM(New, Contacted, Returned, Member), notes.

## 10.4 Cells Domain

### cells

- **Columns:** id PK, tenant\_id, org\_unit\_id FK(church), name, leader\_id FK(people) NULL, assistant\_leader\_id FK NULL, venue, meeting\_day ENUM, meeting\_time TIME, status ENUM(active,inactive).

### cell\_reports

- **Columns:** id PK, tenant\_id, cell\_id FK, report\_date DATE, report\_time TIME, attendance, first\_timers, new\_convert, testimonies TEXT, offerings\_total NUMERIC(12,2), meeting\_type ENUM(prayer\_planning, bible\_study, outreach), status ENUM(submitted, reviewed, approved), notes.
- **Uniqueness:** UNIQUE(tenant\_id, cell\_id, report\_date).

## 10.5 Finance Domain

### funds

- **Columns:** id PK, tenant\_id, name (e.g., Tithe, Offering, Seed, First Fruit, Partnership), is\_partnership BOOL, active BOOL.

### partnership\_arms

- **Columns:** id PK, tenant\_id, name (e.g., Rhapsody of Realities, Healing School, InnerCity Mission, Loveworld TV), active\_from DATE, active\_to DATE NULL, active BOOL.

### batches

- **Columns:** id PK, tenant\_id, org\_unit\_id FK, service\_id FK NULL, status ENUM(draft, locked), locked\_by, locked\_at.
- **Uniqueness:** recommend UNIQUE(tenant\_id, org\_unit\_id, service\_id) when service-linked.

### finance\_entries

- **Columns:** id PK, tenant\_id, org\_unit\_id FK, batch\_id FK NULL, service\_id FK NULL, fund\_id FK, partnership\_arm\_id FK NULL, amount NUMERIC(12,2), currency CHAR(3) DEFAULT 'EUR', method ENUM(cash, kingspay, bank\_transfer, pos, cheque, other), person\_id FK NULL, cell\_id FK NULL, external\_giver\_name TEXT NULL, reference TEXT NULL, comment TEXT NULL, verified\_status ENUM(draft, verified, reconciled, locked) DEFAULT draft, source\_type ENUM(manual, cell\_report) source\_id UUID NULL, transaction\_date DATE.
- **Indexes:** (tenant\_id, org\_unit\_id, transaction\_date), (fund\_id), (partnership\_arm\_id), (person\_id), (cell\_id).

## partnerships

- **Columns:** `id` PK, `tenant_id`, `person_id` FK, `fund_id` FK, `partnership_arm_id` FK NULL, `cadence` ENUM(`weekly`, `monthly`, `quarterly`, `annual`), `start_date` DATE, `end_date` DATE NULL, `target_amount` NUMERIC(12,2) NULL, `status` ENUM(`active`, `paused`, `ended`).
- **Fulfilment:** computed as sum of linked `finance_entries` for the person/fund/arm over the cadence window.

## 10.6 Audit, Imports, Exports

### audit\_logs

- **Columns:** `id` PK, `tenant_id`, `actor_id` FK(`users`), `action` TEXT, `entity_type` TEXT, `entity_id` UUID, `before` JSONB, `after` JSONB, `ip` INET, `user_agent` TEXT, `occurred_at` TIMESTAMPTZ.
- **Indexes:** (`tenant_id`, `entity_type`, `entity_id`), (`occurred_at` DESC).

### imports, import\_errors

- **imports:** job tracking with `status`, `type`, `payload` JSONB, `stats` JSONB.
- **import\_errors:** `import_id` FK, `row_number`, `column`, `message`.

### exports

- **Columns:** job tracking with `scope` JSONB, `format`, `file_url`, `created_by`, `created_at`.

## 10.7 Lookup & Enumerations

- **payment\_methods:** seed values — `cash`, `kingspay`, `bank_transfer`, `pos`, `cheque`, `other`.
- **meeting\_types:** `prayer_planning`, `bible_study`, `outreach`.
- **service\_names:** `Sunday`, `Midweek`, `Special` (or free text).
- **verification\_status:** `draft`, `verified`, `reconciled`, `locked`.

## 10.8 Reporting Summaries (for Performance)

Materialized views / summary tables to back dashboards:

- **summary\_attendance\_daily**(`org_unit_id`, `date`, `men`, `women`, `teens`, `kids`, `first_timers`, `new_converters`, `total`)
- **summary\_giving\_daily**(`org_unit_id`, `date`, `fund_id`, `partnership_arm_id`, `method`, `gross_amount`)
- **summary\_cells\_weekly**(`org_unit_id`, `week`, `reports_submitted`, `avg_attendance`, `first_timers`, `new_converters`, `offerings_total`)
- **summary\_members\_status**(`org_unit_id`, `status`, `count`)

These are refreshed on schedule and/or on write via worker jobs.

## 10.9 Indexing & Constraints Guidelines

- All FK columns indexed: `org_unit_id`, `service_id`, `person_id`, `cell_id`, `fund_id`.
- Time-series access paths: composite indexes on (`tenant_id`, `org_unit_id`, `date`) families.
- Uniqueness constraints: one attendance per service; one cell report per (cell, date); batch uniqueness per (church, service) when linked.
- Soft deletes avoided for finance and attendance; use immutable lock state + audit trail instead.

## 10.10 Data Lineage for Cell Offerings

- When a `cell_report` with `offerings_total` is approved, the worker inserts a `finance_entry` with `source_type='cell_report'` and `source_id = cell_report.id`, linked to the cell's church (`org_unit_id`) and `cell_id`. This ensures Finance and Reports reflect totals without double entry.

## 10.11 RLS Enforcement Note

All SELECT/INSERT/UPDATE/DELETE policies check:

1. `current_setting('app.tenant_id') = tenant_id`
2. **Permission Gate:** `has_perm('<domain>.<noun>.<verb>')`
3. **Scope Gate:** `has_org_access(org_unit_id)`

Sensitive fields (e.g., email, phone) are masked in views unless `registry.people.read` is present.

## 10.12 Notes and Extensions

- **Multi-currency:** keep `currency` on finance entries (default EUR), with future rate tables if needed.
- **Households:** optionally add `households` and `household_members` if family grouping becomes important.
- **Search:** consider trigram indexes for names/phones and partial indexes for active records.

# 11 Reporting Requirements

This section defines the functional and technical reporting requirements for the Christ Embassy Ireland Zone Church Reporting Platform. Reports are generated from verified data across the Registry, Finance, and Cells modules, with access and visibility determined by each user's organisational scope and permissions.

## 11.1 Purpose

The Reports Portal (`reports.<zone>.ce.church`) provides a unified, real-time view of church operations and performance metrics across all levels of the hierarchy—zone, group, church, and cell. It consolidates data from the Registry, Finance, and Cells modules, producing dashboards and detailed reports for pastors and authorised leaders.

## 11.2 General Reporting Principles

- Reports must reflect only **verified and locked** data from Finance, **approved** Cell Reports, and **submitted and approved** Attendance records.
- All data must be aggregated according to organisational hierarchy: Region → Zone → Group → Church → Cell.
- Access to reports is restricted to a user's defined scope. Users can only view or export data for churches, groups, or cells they oversee.
- Every report shall include:
  1. Scope (zone, group, church, or cell)
  2. Report period (start date – end date)
  3. Generation timestamp
  4. Generating user and role
- All reports shall be exportable in CSV, Excel, and PDF formats.

## 11.3 Dashboards Overview

### 1. Membership Dashboard

- Total registered members by church, group, and zone.
- Membership growth over time (weekly and monthly deltas).
- Demographics: gender, age groups, marital status.
- Foundation School completion rate.
- Baptism tracking: completed vs pending.
- Department participation and leadership distribution.

### 2. First-Timers & Assimilation Dashboard

- First-timer registrations by service and date.
- First-timer follow-up pipeline: New → Contacted → Returned → Member.
- Conversion rates over rolling 30-day and 60-day windows.
- Assimilation performance by church and group.
- First-timer retention trends and percentage of visitors who become regular attendees.



### **3. Attendance Dashboard**

- Attendance trends per service (Sunday, Midweek, Special).
- Demographic splits (men, women, teens, kids).
- First-timer and new convert counts per service.
- Comparative attendance growth by group or church.
- Rolling averages and month-on-month percentage change.

### **4. Finance Dashboard**

- Offerings, tithes, seeds, and first-fruits by service and date.
- Partnership contributions by arm (Rhapsody of Realities, Healing School, InnerCity Mission, Loveworld TV, custom campaigns).
- Total giving by fund and by payment method.
- Per-capita giving (total giving  $\div$  average attendance).
- Partnership fulfilment tracking by member and arm.
- Cell offering reconciliation totals (summed from cell reports).

### **5. Cells Dashboard**

- Active vs inactive cells by church and group.
- Weekly report submission compliance rate.
- Average cell attendance and outreach results.
- First-timers and new converts from cell meetings.
- Testimony and offering summaries.

### **6. Combined Zone Overview**

- Consolidated zone-level view of membership, attendance, giving, and cell participation.
- Comparative ranking of groups or churches based on growth, giving, and assimilation performance.
- KPI summaries for leadership reports and quarterly reviews.

## 11.4 Standard Reports (Tabular Exports)

- **RPT-01: Member Directory** — Export of all members with contact details, foundation/baptism status, and department.
- **RPT-02: First-Timer Log** — Detailed list of first-timers, visit date, inviter, contact status, and assimilation outcome.
- **RPT-03: Attendance Register** — Attendance summaries by service, including totals and demographic breakdown.
- **RPT-04: Finance Summary** — Totals by fund, partnership arm, and payment method, aggregated by date range and church.
- **RPT-05: Partnership Fulfilment Report** — Partner list with pledge, fulfilment progress, and outstanding balance.
- **RPT-06: Cell Meeting Summary** — Cell attendance, first timers, testimonies, offerings, and meeting types.
- **RPT-07: Church Growth Summary** — Combined metrics across membership, attendance, finance, and cells for longitudinal analysis.

## 11.5 Drill-Down Hierarchies

Dashboards and reports shall support the following hierarchical navigation:

- Zone → Group → Church → Cell → Individual Member
- Zone → Group → Church → Service

Drill-down capability must respect access scope and permissions, ensuring users cannot traverse below their authorised level. Each layer should display both aggregated metrics and access to the next level for detailed insights.

## 11.6 Scheduled & Automated Reports

- Pastors and administrators may configure recurring reports to be sent daily, weekly, or monthly via email.
- Supported schedules:
  - Weekly Summary (attendance, finance, cell compliance)
  - Monthly Church Overview
  - Quarterly Zone KPI Report
- All scheduled reports shall be generated by background jobs and delivered as downloadable PDF or Excel files.
- System logs all automated deliveries in the **exports** table with metadata: recipient, scope, schedule, and delivery timestamp.

## 11.7 Data Sources and Validation Rules

- Attendance data drawn from `attendance` table joined to `services` and `org_units`.
- Finance data drawn from `finance_entries` where `verified_status` IN ('verified','reconciled').
- Cell metrics drawn from approved `cell_reports`.
- Membership counts and demographics derived from `people` and `memberships` tables.
- All aggregates are filtered by `tenant_id` and limited by organisational scope.

## 11.8 Performance Requirements for Reports

- Dashboard load time (P95)  $\leq 2$  seconds using materialized views.
- Large exports ( $\geq 10k$  rows) generated asynchronously with email notification when ready.
- Aggregations precomputed nightly or on data-change events.
- All exports tagged with tenant, scope, and timestamp for audit.

## 11.9 Report Access & Audit Logging

- Every report view, export, or scheduled delivery shall log: user ID, scope, report type, filters, and timestamp.
- Full-PII exports require `system.exports.full_pii` permission and step-up authentication.
- Audit entries stored in `audit_logs` with action `report.export` or `report.view`.

## 11.10 Future Extensions

- Comparative cross-zone reporting for multi-zone leadership.
- Integration with KingsPay transaction summaries for automated import.
- Predictive dashboards estimating growth or giving trends using historical data.
- Optional mobile dashboard app for pastors with read-only access.

# 12 User Journeys and Workflows

This section outlines the key user workflows within the Christ Embassy Ireland Zone Church Reporting Platform. Each workflow describes how a typical task is performed from start to finish, highlighting role interactions, system validations, and audit behaviour.

## 12.1 Overview

Workflows are grouped by portal and role. Each represents a real-world ministry process captured within the system and emphasises data integrity, accountability, and reporting traceability.

## 12.2 Registry Workflows

### 1. First-Timer Registration and Assimilation

1. A Church Administrator logs in to the Registry Portal and selects the relevant church.
2. After a service, the clerk records first-timers:
  - Full name, gender, phone/email, inviter/source.
  - Service attended (linked via `service_id`).
  - Initial status: **New**.
3. The first-timer record appears in the “Follow-Up Pipeline” dashboard.
4. When follow-up occurs, a pastoral staff member updates status to **Contacted** or **Returned**.
5. If the person joins a foundation class, the system allows conversion to a Member record:
  - The user clicks “Convert to Member.”
  - A new **people** record is created, copying contact info and linking to the original first-timer entry.
6. The workflow ends when status becomes **Member** and is reflected in assimilation KPIs.

### 2. Service Attendance Recording

1. Church Administrator selects a `service_date` and `service_name` (Sunday, Mid-week, or Special).
2. Attendance counts are entered for men, women, teens, and children, as well as totals for first-timers and new converts.
3. The system validates one attendance record per service per church.
4. On submission, an attendance summary is written to the **attendance** table.
5. Background job updates `summary_attendance_daily(org_unit_id, date)`.
6. Zonal and Group Pastors immediately see updated totals in their dashboards.

### 3. Member Record Update

1. Clerks or pastors edit member details (address, marital status, foundation school completion, baptism date, etc.).
2. All edits are logged to `audit_logs` with before/after snapshots.
3. Updates trigger recalculation of membership demographics and KPIs.

## 12.3 Finance Workflows

### 1. Offering Entry and Batch Locking

1. Finance Officer creates a new batch for a church and service.
2. Within the batch, each entry records:
  - Transaction date, amount, fund, partnership arm, method, member or cell reference, and comment.
3. Entries remain in `draft` status until verified.
4. When verification is complete:
  - The first authorised user with `finance.batches.lock` attempts to lock the batch.
  - The system requests a second approver (dual control).
  - Upon dual confirmation, the batch status updates to `locked`.
5. Once locked:
  - Entries become immutable.
  - Dashboards and summaries refresh automatically.
  - Audit trail logs both approvers' actions.

### 2. Partnership Tracking

1. Pastor or Finance Officer creates partnership records for members, specifying: `fund_id`, `partnership_arm_id`, pledge amount, and cadence.
2. Finance entries linked to the same member and partnership arm increment fulfilment totals automatically.
3. The partnership dashboard displays current fulfilment percentage, outstanding balance, and due dates.

### 3. Reconciliation Workflow

1. Zonal or Group Finance Team exports a verification summary by fund and church.
2. After confirming bank deposits, the team marks verified batches as `reconciled`.
3. Locked and reconciled batches contribute to financial KPIs.

## 12.4 Cells Workflows

### 1. Weekly Cell Report Submission

1. Cell Leader logs in to the Cells Portal and selects their assigned cell.
2. Fills in:
  - Date and meeting time.
  - Attendance, first-timers, new converts.
  - Testimonies and notes.
  - Offerings total.
  - Meeting type (prayer, Bible study, outreach).
3. On submission:
  - Report status = `submitted`.
  - Offerings total triggers an automated creation of a `finance_entry` with `source.type='cell_report'` and `source_id=cell_report.id`.
  - The finance entry appears in Finance Portal under the respective church for verification.
4. Pastors can review and approve reports (optional workflow step), changing status to `approved`.

### 2. Cell Reporting Compliance

1. System checks each active cell weekly.
2. Cells without reports are marked “missing submission.”
3. Reports Portal dashboard visualises submission compliance by church and group.

## 12.5 Reports Portal Workflows

### 1. Dashboard Interaction and Drill-Down

1. User logs in and the system determines accessible scope via assigned roles.
2. Dashboard displays summary KPIs: attendance, giving, membership, cell compliance.
3. User drills down: `Zone` → `Group` → `Church` → `Cell` → `Member`.
4. Filters: date range, fund, service, or partnership arm.
5. Data sourced from materialized summary tables.

## 2. Report Export

1. User applies filters and clicks “Export.”
2. If small ( $\leq 10,000$  rows), export is synchronous; otherwise, a background job is queued.
3. Export file (CSV, Excel, or PDF) generated and stored in S3.
4. User notified with download link; audit log entry created.

## 3. Scheduled Reporting

1. Pastor configures recurring reports (e.g., Weekly or Monthly Summary).
2. Job scheduler triggers export and sends email via background worker.
3. The `exports` table logs delivery time, recipient, and success status.

# 12.6 Administrative Workflows

## 1. User and Role Provisioning

1. Zonal Pastor creates Group Pastors and assigns group scopes.
2. Group Pastors create Church Pastors/Coordinators.
3. Church Pastors create Church Administrators, Finance Officers, and Cell Leaders.
4. All actions audited with user, scope, and timestamp.

## 2. Importing Legacy Data

1. Admin uploads legacy spreadsheet (e.g., membership list).
2. System displays field mapping preview.
3. Validation engine checks for missing or invalid data.
4. User corrects and re-uploads if errors exist.
5. Upon successful import:
  - Records inserted under correct church/org scope.
  - Audit log created with import job reference.

# 12.7 Workflow Consistency Rules

- Every workflow action must be traceable in `audit_logs`.
- Data transitions follow strict states:
  - **Finance:** draft  $\rightarrow$  verified  $\rightarrow$  reconciled  $\rightarrow$  locked
  - **Cells:** submitted  $\rightarrow$  reviewed  $\rightarrow$  approved
  - **First-timers:** new  $\rightarrow$  contacted  $\rightarrow$  returned  $\rightarrow$  member
- State changes are immutable in reverse unless unlocked by authorised personnel with dual control.

## 13 Risks and Assumptions

This section identifies the major risks, dependencies, and underlying assumptions associated with the Christ Embassy Ireland Zone Church Reporting Platform. Each risk includes an impact assessment and mitigation strategy.

### 13.1 Project Assumptions

1. The system will initially be deployed for the **Ireland Zone** and may later be replicated for other zones.
2. All participating churches, groups, and cells will maintain stable internet access sufficient for web-based data entry.
3. Existing spreadsheets and records are sufficiently structured to support automated import after field mapping and validation.
4. The zone has designated personnel for each portal: Church Administrators, Finance Officers, Cell Leaders, and Report Viewers.
5. All users will undergo basic onboarding and training before launch.
6. Zone leadership will define approved organisational hierarchies and naming conventions prior to data migration.
7. Financial records are entered manually for reporting; no real payment processing is performed on the platform.
8. Access control will follow the hierarchical structure (Zonal → Group → Church → Cell) without exceptions.
9. A central technical administrator or IT officer will manage backups, infrastructure, and access provisioning at the zone level.

### 13.2 Project Risks and Mitigation Strategies

#### 1. Data Quality Risk

- **Description:** Legacy spreadsheets may contain inconsistent or incomplete data.
- **Impact:** Duplicate or invalid records could distort reporting accuracy.
- **Mitigation:** Implement pre-import validation, deduplication tools, and review dashboards for anomalies.

#### 2. User Adoption Risk

- **Description:** Transitioning from manual processes to a digital system may face initial resistance from some users.
- **Impact:** Incomplete reporting or low data entry compliance.
- **Mitigation:** Provide short training sessions, in-portal help, and simplified mobile access for clerks and cell leaders.



### 3. Connectivity Risk

- **Description:** Some outreaches or churches may experience poor network connectivity.
- **Impact:** Delayed data entry or missed attendance submissions.
- **Mitigation:** Support offline entry templates or lightweight mobile interfaces that sync when connectivity is restored.

### 4. Access Control Misconfiguration

- **Description:** Incorrect assignment of scopes or roles may expose data beyond intended boundaries.
- **Impact:** Breach of data privacy or loss of trust in system security.
- **Mitigation:** Use guided user provisioning wizards, predefined role templates, and periodic permission audits.

### 5. Financial Data Entry Errors

- **Description:** Manual entry of offerings or partnerships may introduce typos or misallocations.
- **Impact:** Reporting inaccuracies or reconciliation variance.
- **Mitigation:** Dual control for batch locking, field-level validation (numeric ranges, required fund), and audit trails.

### 6. Delayed Report Generation

- **Description:** Large reports may take long to generate during peak hours.
- **Impact:** Poor user experience and delayed insights for pastors.
- **Mitigation:** Use pre-aggregated summary tables, background job queues, and read replicas for analytics.

### 7. Data Loss or Corruption

- **Description:** Accidental deletion, database error, or hardware failure could compromise stored data.
- **Impact:** High; loss of financial or membership records.
- **Mitigation:** Daily encrypted backups, multi-region storage, and quarterly restore tests.

## 8. Audit Log Overflow or Mismanagement

- **Description:** Unbounded audit log growth may degrade performance or consume storage.
- **Impact:** Reduced database efficiency.
- **Mitigation:** Partition audit tables by date, archive older records to cold storage after 12 months.

## 9. Version Drift Between Portals

- **Description:** If portal frontends evolve asynchronously, UI inconsistencies or incompatible API calls may occur.
- **Impact:** Broken functionality or reporting errors.
- **Mitigation:** Maintain shared component libraries and strict version tagging across all sub-domains.

## 10. Leadership Reporting Delays

- **Description:** Pastors may not receive timely insights if lower-level data entry is delayed.
- **Impact:** Leadership decisions based on incomplete data.
- **Mitigation:** Scheduled reporting reminders, compliance KPIs, and automated notifications for missing submissions.

## 13.3 Technical Risks

### 1. Scalability Constraints

- **Description:** Rapid growth in churches or records may outpace single-database performance.
- **Mitigation:** Deploy read replicas for reporting, introduce partitioning on date/org\_unit\_id, and plan migration path to distributed analytics store if necessary.

### 2. Background Job Failures

- **Description:** Import, export, or scheduled jobs may fail silently due to worker errors.
- **Mitigation:** Implement job retry policies, error queues, and alerting dashboards for failed tasks.

### 3. API Performance Degradation

- **Description:** Inefficient queries or unindexed filters may degrade API latency under load.
- **Mitigation:** Continuous monitoring of query plans and use of cached summary tables for heavy endpoints.

## 13.4 Operational Risks

### 1. Staff Turnover

- **Description:** Departure of trained users may reduce data entry consistency.
- **Mitigation:** Maintain user manuals, assign backup clerks, and provide refresher sessions.

### 2. Misinterpretation of Reports

- **Description:** Pastors or finance teams may misread dashboard data if untrained in metric definitions.
- **Mitigation:** Include tooltips and embedded data definitions on dashboards; provide glossary in appendix.

### 3. Governance Risk

- **Description:** Absence of clear data ownership could lead to disputes or inconsistent updates.
- **Mitigation:** Define roles and responsibilities clearly within each church and group; enforce audit-based accountability.

## 13.5 Residual Risk Statement

While technical measures and governance processes significantly reduce likelihood and impact, residual risk remains in user compliance and operational discipline. Continuous monitoring, periodic training, and consistent leadership oversight will be essential to maintain data quality and trust in the system.

## A Appendices

### A.1 A. Glossary of Terms

Term	Definition
Zone	A major administrative division within Christ Embassy comprising multiple groups and churches.
Group	A cluster of churches within a zone supervised by a Group Pastor.
Church	A local assembly or parish within a group.
Outreach	A smaller church unit or fellowship group connected to a parent church.
Cell	The smallest fellowship unit within the ministry, typically meeting weekly in homes or venues.
Service	A worship gathering (e.g., Sunday or Midweek service).
First-timer	A visitor attending a service for the first time.

<b>Partner</b>	A member committed to regular financial support through a specific partnership arm.
<b>Batch</b>	A collection of financial entries recorded for a given service or period before verification.
<b>Verified Batch</b>	A batch reviewed and locked by authorised finance personnel.
<b>Foundation School</b>	A series of classes introducing new converts to the ministry's core teachings.
<b>Tenant</b>	Logical boundary representing a zone within the system's multi-tenant database.
<b>RLS</b>	Row-Level Security, used to enforce data access restrictions by organisation scope.
<b>PII</b>	Personally Identifiable Information such as name, email, phone number, or address.
<b>Dashboard</b>	A visual interface showing KPIs and trends from aggregated data.
<b>Dual Control</b>	Security process requiring two authorised users to approve a financial lock action.
<b>Cell Report</b>	A weekly submission detailing attendance, new converts, and offerings from a cell meeting.
<b>KingsPay</b>	Christ Embassy's internal electronic giving platform, referenced for record-keeping only.

## A.2 B. Enumerations and Lookup Tables

Enumeration	Allowed Values / Description
<b>Service Type</b>	Sunday, Midweek, Special
<b>Meeting Type</b>	Prayer and Planning, Bible Study, Outreach
<b>Payment Method</b>	Cash, KingsPay, Bank Transfer, POS, Cheque, Other
<b>Verification Status</b>	Draft, Verified, Reconciled, Locked
<b>Cell Report Status</b>	Submitted, Reviewed, Approved
<b>First-timer Status</b>	New, Contacted, Returned, Member
<b>Partnership Cadence</b>	Weekly, Monthly, Quarterly, Annual
<b>Org Unit Type</b>	Region, Zone, Group, Church, Outreach
<b>User Role Template</b>	Zonal Pastor, Group Pastor, Church Pastor, Finance Officer, Church Administrator, Cell Leader, Reports Viewer

## A.3 C. Example Import Template Fields

### Membership Import

- Full Name, Gender, Date of Birth, Email, Phone
- Address, County, Town, Postcode
- Date Joined, Foundation School Completed, Baptism Date

- Marital Status, Department, Cell

### Finance Import

- Transaction Date, Fund Type, Partnership Arm, Amount
- Payment Method, Giver Name or Member Code
- Service Date, Batch Reference, Comment

### Cell Report Import

- Cell Name, Meeting Date, Time, Attendance, First-timers, New Converts, Offerings Total, Meeting Type

## A.4 D. Key Performance Indicator Reference

KPI	Definition / Calculation
Membership Growth Rate	$\frac{\text{New Members in Period}}{\text{Members at Start of Period}} \times 100\%$
Attendance Growth Rate	$\frac{\text{Current Attendance} - \text{Previous Attendance}}{\text{Previous Attendance}} \times 100\%$
First-Timer Retention	$\frac{\text{First-timers who Returned}}{\text{Total First-timers}} \times 100\%$
Partnership Fulfilment	$\frac{\text{Total Given}}{\text{Pledged Amount}} \times 100\%$
Per-Capita Giving	$\frac{\text{Average Attendance}}{\text{Reports Submitted}} \times 100\%$
Cell Compliance Rate	$\frac{\text{Expected Reports}}{\text{Verified Entries}} \times 100\%$
Verified Finance Ratio	$\frac{\text{Total Finance Entries}}{\text{Total Finance Entries}} \times 100\%$

## A.5 E. Reference Links (External)

- Christ Embassy Official Website: <https://www.christembassy.org>
- Loveworld Inc. Partnership Arms Overview: <https://loveworldinc.org/partnership-arms>
- Data Protection Guidelines (Ireland): <https://www.dataprotection.ie>