The Smart Genealogist: Project Vision & Architecture

Project Vision Statement

Core Objective: Develop an AI-powered system for detecting and explaining anomalies in genealogical forests (multiple interconnected family trees) with high precision and genealogist-friendly interface.

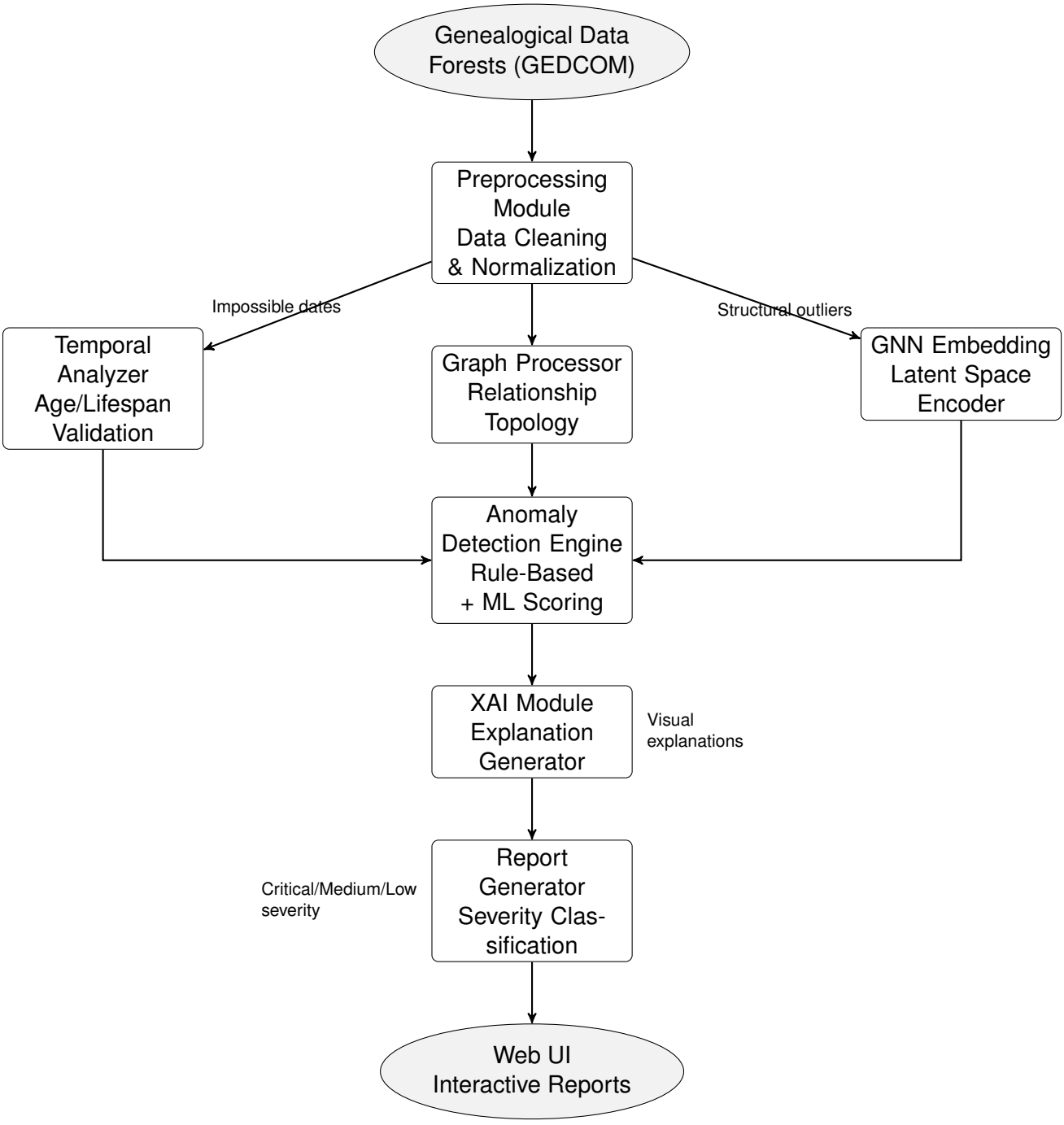
Key Capabilities

- **Anomaly Detection (Precision $> 95\%$)**
  - *Temporal Checks:* Impossible age gaps, lifespan overlaps
  - *Graph-Based Outliers:* High-degree nodes, circular relationships
  - *Embedding-Based:* GNN-encoded latent space anomalies
- **Explainable AI**
  - Visual timeline conflict displays
  - Natural language explanations
  - Severity-based prioritization
- **User Interface**
  - Web-based interactive visualization
  - Filterable anomaly reports
  - Collaborative workspace

Implementation Goals

Performance	$> 95\%$ precision on labeled anomaly dataset
Usability	Intuitive UI with transparent reasoning paths
Scalability	Handle forests with > 100k individuals
Security	GDPR-compliant data processing

System Architecture Flowchart



Architecture Components:

- **Data Ingestion:** Supports GEDCOM standard + API integrations
- **Processing Pipeline:** Parallel anomaly detection pathways
- **Hybrid Detection:** Combines rules-based checks ML-
- **Explanation Layer:** Generates human-interpretable justifications
- **Web Interface:** React.js frontend D3.js

Service Layer Diagram for Anomaly Detector

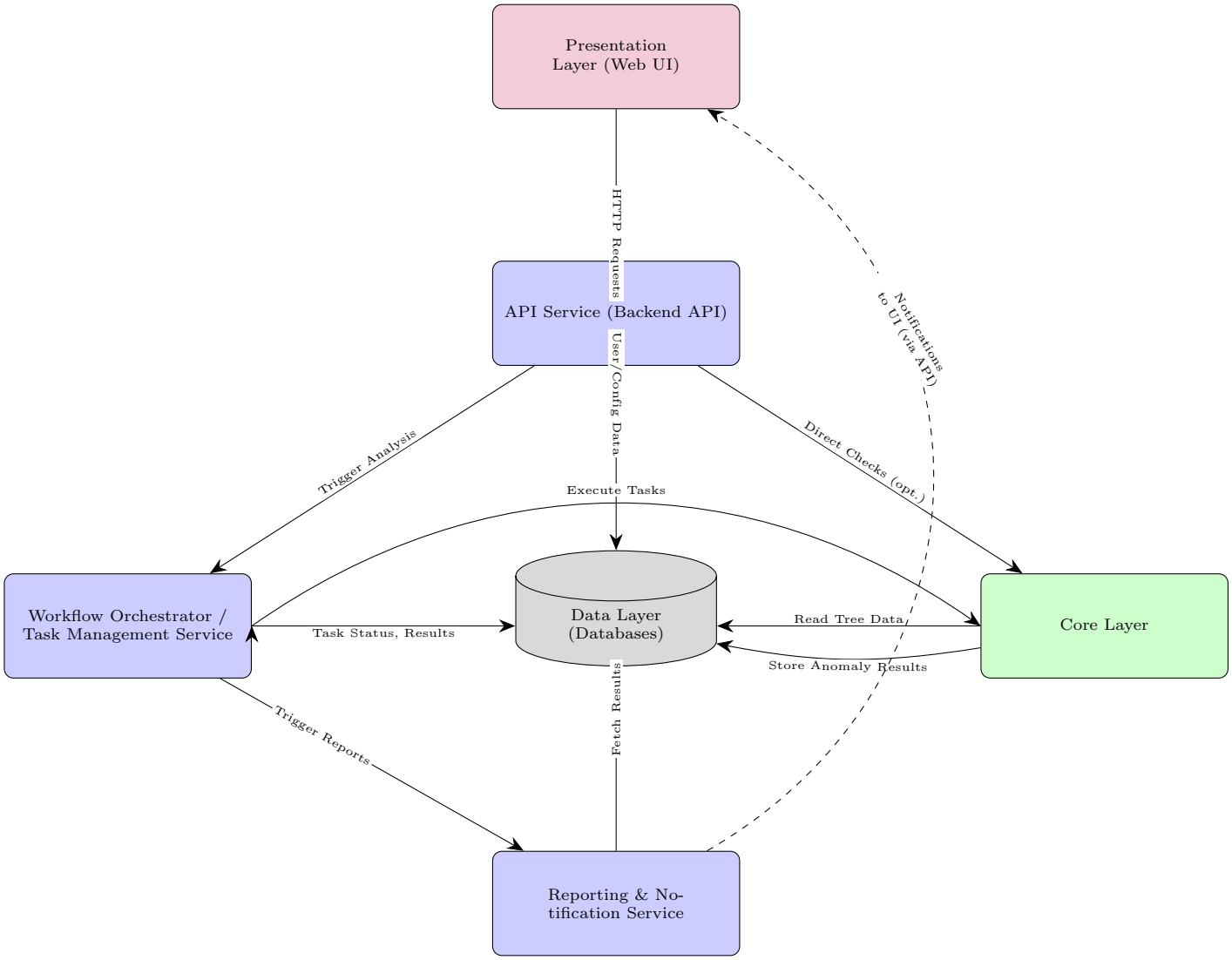


Figure 1: Revised High-Level Diagram with alternative layout and database icon. Only 'Notifications to UI' is dashed.

Service Layer Component Details

Table 1: Detailed Description of Service Layer Components for "Genealogical Trees Scanner"

Service Component	Description & Responsibilities	Key Interactions	Alignment with Project Requirements
API Service (Backend API)	Provides a secure and well-defined interface (e.g., RESTful or GraphQL) for the Presentation Layer (Frontend) to interact with the system's backend functionalities. **Responsibilities:** • User authentication and authorization. • Handling requests for genealogical data upload and management (CRUD operations on trees/individuals, if applicable through this system). • Initiating anomaly detection tasks by invoking the Workflow Orchestrator. • Retrieving anomaly reports, individual anomaly details, and their explanations for UI display. • Managing system configurations accessible to users (e.g., thresholds, active checks). • Relaying notifications from the Reporting Service to the Frontend.	• Presentation Layer (Frontend) • Workflow Orchestrator • Core Layer • Data Layer (for user profiles, configurations, direct data queries) • Reporting & Notification Service (to receive notification data for UI)	Supports "Web interface (backend + frontend)" (Task 3). Enables the "Genealogist-friendly UI" (Expected Result 3) by serving data and functionality. Crucial for a "scalable...tool" (Objective).
Workflow Orchestrator / Task Management Service	Manages and coordinates complex, potentially long-running, asynchronous tasks, particularly the multi-step anomaly detection process across forests of trees. **Responsibilities:** • Receiving analysis requests from the API Service. • Deconstructing analysis into sub-tasks (e.g., Temporal Checks, Graph-Based Outliers, Embedding-Based Detection for each tree or batch of trees). • Scheduling, queuing, and managing the execution of these sub-tasks on the Anomaly Detection Engine. • Monitoring task progress, handling retries on failure, and detailed error logging. • Triggering the Reporting & Notification Service upon successful completion or significant events. • Ensuring analysis results and metadata are persisted to the Data Layer.	• API Service • Core Layer • Reporting & Notification Service • Data Layer (for storing task states, results, logs)	Key to a "scalable, AI-driven tool" (Objective) by managing resource-intensive detection processes efficiently. Ensures reliable execution of "Anomaly Detection" tasks (Task 1). Supports building a "Smart system" (Expected Result 1).

Continued on next page

Table 1 – continued from previous page

Service Component	Description & Responsibilities	Key Interactions	Alignment with Project Requirements
Reporting & Notification Service	Responsible for compiling and formatting analysis results into structured reports and for notifying users about relevant system events or detected anomalies. **Responsibilities:** • Aggregating raw anomaly data, severity levels, and explanations from the Data Layer or directly from the Anomaly Detection Engine's outputs. • Generating "Anomaly Reports (with severity levels)" (Task 3) in formats suitable for UI consumption (e.g., JSON for dynamic tables/visualizations). • Facilitating the "Visualize anomalies" (Task 2) requirement by preparing data for graphical representation on the frontend. • Managing and dispatching notification data (to be relayed by API Service to Frontend) and/or direct notifications (e.g., email via an external gateway) about scan completion, newly detected critical anomalies, or system status.	• Workflow Orchestrator (trigger) • Data Layer (for fetching analysis results) • API Service (to channel notification messages/data to Frontend) • (Potentially) External Notification Gateways (e.g., SMTP server for emails)	Directly fulfills "Anomaly Reports (with severity levels)" (Task 3). Contributes to "transparent reasoning" (Expected Result 3) by delivering structured insights. Enhances the "Genealogist-friendly UI" (Expected Result 3) by providing timely feedback and actionable information.

Presentation Layer

Table 1: Web UI Enhancements for Genealogical Trees Scanner (Anomaly Detector)

UI Section / Component	Feature Description	Purpose / Alignment with Project Requirements
Main Genealogical Tree View	**Node Anomaly Indicators** Visual cues (e.g., colored border and a small icon like [!] or a specific symbol) on person nodes and relationship lines. Color-coding indicates anomaly severity (e.g., Red: Critical, Yellow: Warning, Blue: Informational).	Directly flags anomalies on the tree (Project Task 1: Flag anomalies). Provides immediate visual feedback, enhancing genealogist-friendly UI (Expected Result 3).
Main Genealogical Tree View	**Anomaly Indicator Tooltip** On hovering over an anomaly indicator, a tooltip appears displaying a concise summary (e.g., "2 Anomalies: 1 Temporal, 1 Structural").	Offers quick, non-intrusive information. Improves user experience and genealogist-friendliness (Expected Result 3).
Main Genealogical Tree View (Toolbar)	**Anomaly Display Filters** Dedicated filter options presented as dropdowns or toggle buttons to: • Show/hide all anomaly indicators. • Filter by severity level (Critical, Warning, Informational). • Filter by anomaly category (e.g., Temporal, Structural, Embedding-based).	Allows users to focus on specific types or severities of anomalies. Supports "Anomaly Reports (with severity levels)" (Project Task 3).
Person Detail Panel (Sidebar)	**Dedicated "Anomalies" Tab** A new tab alongside "Overview", "Photos", etc. Lists all anomalies specifically related to the selected individual. Each entry includes severity icon, a brief description, and an "Explain" button.	Organizes anomaly information per person. Directly supports displaying "Anomaly Reports" (Project Task 3) and facilitates focused investigation.
Main Navigation Menu	**"Anomaly Analysis" Section Link** A new top-level menu item, e.g., "Anomaly Analysis" or "Tree Health Check", leading to a dedicated page.	Provides a centralized hub for comprehensive anomaly management and review, key for a "Smart system" (Expected Result 1).
"Anomaly Analysis" Page	**Anomaly Dashboard & Scan Control** Top section of the page displaying: • Summary statistics (total anomalies, count by severity, count by type). • A "Run Full Scan / Re-scan Tree" button to initiate or update the anomaly detection process.	Offers an overview of tree data quality and control over the analysis process. Supports "Anomaly Reports" (Project Task 3).

Continued on next page

Table 1 – continued from previous page

UI Section / Component	Feature Description	Purpose / Alignment with Project Requirements
"Anomaly Analysis" Page	**Comprehensive Anomaly Report Table** A detailed, sortable, and filterable table listing all detected anomalies across the tree. Columns include: • Severity (icon/text). • Anomaly Type (e.g., Impossible Age Gap, Circular Relationship). • Concise Description. • Involved Individuals (links to their profiles). • Data Points (e.g., conflicting dates). • "Explain" button/link. • Status field (e.g. New, Investigating, Ignored, Resolved) - (Optional, for advanced workflows).	Delivers "Anomaly Reports (with severity levels)" (Project Task 3). Core for a "Genealogist-friendly UI with transparent reasoning" (Expected Result 3). Enables detection precision assessment (Expected Result 2).
Modal / Dedicated View (from "Explain" button)	**Anomaly Explanation View** Presents a detailed explanation for a selected anomaly: • **Clear Textual Explanation:** What the anomaly is, why it's flagged (e.g., "Parent's birth date is after child's birth date"). • **Involved Data:** Highlights the specific data points causing the anomaly. • **Visualization (XAI):** Contextual visualization (e.g., mini-timeline for temporal conflicts, highlighted path for circular relationships). • **Suggested Actions:** Guidance on how to verify or correct the data.	Fulfills "Detects and explains anomalies" (Objective) and "Explainable AI (XAI): Visualize anomalies" (Project Task 2). Ensures "transparent reasoning" (Expected Result 3) and supports building a "Smart system" (Expected Result 1).
Data Editing Interface	**Post-Edit Anomaly Re-check Indication** After a user edits data relevant to an anomaly, provide an immediate visual cue or a subtle notification if the change resolved an existing anomaly or, conversely, introduced a new one for the edited person and their immediate relations.	Provides instant feedback, making the UI more dynamic and "genealogist-friendly" (Expected Result 3). Helps maintain data quality proactively.