Eleedom IMF System Architecture and Data Flow Documentation

System Overview

```
graph TD
    A[Frontend] --> B[Backend API]
    B --> C[(MongoDB Database)]
    B --> D[Email Service]
    B --> E[File Storage]

    subgraph Frontend Applications
        A1[Admin Portal]
        A2[Employee Portal]
        A3[HR Portal]
        A4[Branch Portal]
        A5[Ops Portal]
        A6[Advisor Portal]
        A7[Finance Portal]
        A8[CIC Portal]
    end

    A --> A1 & A2 & A3 & A4 & A5 & A6 & A7 & A8
```

Authentication Flow

```
sequenceDiagram
    participant User
    participant Frontend
    participant Backend
    participant Database
    participant EmailService

    User->>Frontend: Enter Credentials
    Frontend->>Backend: POST /login
    Backend->>Database: Validate Credentials
    Database-->>Backend: User Data
    Backend->>Backend: Generate JWT
    Backend-->>Frontend: Return Token & User Info
    Frontend->>Frontend: Store in SessionStorage
```

User Management Flow

```
flowchart TD
    A[User Registration] --> B{User Type}
    B -->|Admin| C[Admin Registration]
    B -->|Employee| D[Employee Registration]
    B -->|HR| E[HR Registration]
    B -->|Branch| F[Branch Registration]
    B -->|Ops| G[Ops Registration]
    B -->|Advisor| H[Advisor Registration]
    B -->|Finance| I[Finance Registration]
    B -->|CIC| J[CIC Registration]

    C & D & E & F & G & H & I & J --> K[Save to Database]
    K --> L[Send Welcome Email]
```

Password Reset Flow

```
sequenceDiagram
    participant User
    participant System
    participant Database
    participant Email

    User->>System: Request Password Reset
    System->>Database: Verify User
    System->>System: Generate Reset Token
    System->>Email: Send Reset Link
    User->>System: Access Reset Link
    System->>System: Verify Token
    User->>System: Enter New Password
    System->>Database: Update Password
    System->>Email: Send Confirmation
```

Policy Management Flow

```
graph TD
    subgraph Policy Creation
        A[Advisor/Branch] -->|Creates| B[New Policy]
        B -->|Validates| C[Policy Details]
        C -->|Saves| D[(Policy Collection)]
    end

    subgraph Policy Processing
        D -->|Retrieved by| E[OPS Admin]
        E -->|Verifies| F[Policy Details]
        F -->|Updates| G[Policy Status]
        G -->|Notifies| H[Finance Team]
    end

    subgraph Policy Types
        PT[Policy Types] -->|Includes| MT[Motor]
        PT -->|Includes| HT[Health]
        PT -->|Includes| LT[Life]
        PT -->|Includes| GT[General]
    end

    subgraph Policy Components
        PC[Policy Details] -->|Contains| NC[NCB Details]
        PC -->|Contains| PM[Payment Mode]
        PC -->|Contains| PO[Payout Options]
        PC -->|Contains| CS[Commission Slab]
    end

    subgraph Related Collections
        D <-->|References| CO[(Company Collection)]
        D <-->|References| BR[(Branch Collection)]
        D <-->|References| AD[(Advisor Collection)]
        D <-->|References| CU[(Customer Collection)]
    end
```

Policy Data Flow

```
sequenceDiagram
    participant Advisor
    participant Branch
    participant OPS
    participant Finance
    participant Database
    participant Customer

    Advisor->>Branch: Create Policy Request
    Branch->>Database: Save Policy Details
    Database-->>OPS: Notify New Policy
    OPS->>Database: Verify & Update Status
    Database-->>Finance: Policy Ready for Processing
    Finance->>Database: Update Payment Status
    Database-->>Customer: Policy Confirmation
    Database-->>Advisor: Commission Update
```

Policy Collection Schema Overview

```
classDiagram
    class Policy {
        +String policyId
        +String policyType
        +String companyId
        +String branchId
        +String advisorId
        +String customerId
        +Date startDate
        +Date endDate
        +Number premium
        +String status
        +Object paymentDetails
        +Object commissionDetails
    }

    Policy --> Company
    Policy --> Branch
    Policy --> Advisor
    Policy --> Customer
    Policy --> PaymentMode
    Policy --> CommissionSlab
```

Data Models

User Authentication

- Admin
- Employee
- HR Admin
- Branch
- Ops Admin
- Advisor
- Finance
- CIC

Core Entities

- Company
- Branch
- Policy
- Employee
- Attendance
- Salary
- Leave Balance

API Endpoints

Authentication Routes

- POST /loginadmin - Admin login
- POST /login/employee - Employee login
- POST /hradmin/login - HR login
- POST /branches/loginbranch - Branch login
- POST /ops/login - Ops login
- POST /advisor/login - Advisor login
- POST /finance/login - Finance login
- POST /cic/login - CIC login

Password Management

- POST /forgot/{user-type}/pass - Initiate password reset
- POST /{user-type}/pass/:id/:token - Reset password

Employee Management

- POST /dashboard/addemployee - Add new employee
- GET /employees/data - List employees
- PUT /api/employee/update/:id - Update employee
- DELETE /employee/data/:id - Delete employee

Branch Management

Security Implementation

1. Authentication
   - JWT based authentication
   - Token expiration: 8-24 hours
   - Session storage for token management
2. Password Security
   - Bcrypt password hashing
   - Salt rounds: 10
   - Password reset tokens with 15-minute expiration
3. Email Notifications
   - Welcome emails
   - Password reset links
   - Account updates

Database Collections

1. Users Collections:
   - admins
   - employees
   - hradmins
   - branches
   - opsadmins
   - advisors
   - finance
   - cic
2. Operational Collections:
   - policies
   - attendance
   - salaries
   - leaves
   - branches
   - companies

Component Details Documentation

Portal Components

```
graph TD
    subgraph Admin Portal
        AD[Admin Dashboard] --> AUM[User Management]
        AD --> ABM[Branch Management]
        AD --> ACM[Company Management]
        AD --> APM[Policy Management]
        AD --> ASM[System Settings]
    end

    subgraph Employee Portal
        ED[Employee Dashboard] --> EP[Profile]
        ED --> EA[Attendance]
        ED --> EL[Leave Management]
        ED --> ES[Salary Details]
        ED --> EDR[Daily Reports]
    end

    subgraph HR Portal
        HR[HR Dashboard] --> HEM[Employee Management]
        HR --> HAM[Attendance Management]
        HR --> HSM[Salary Management]
        HR --> HLM[Leave Management]
        HR --> HHM[Holiday Management]
        HR --> HOL[Offer Letters]
    end

    subgraph Branch Portal
        BD[Branch Dashboard] --> BPM[Policy Management]
        BD --> BAM[Advisor Management]
        BD --> BCM[Customer Management]
        BD --> BDR[Daily Reports]
        BD --> BFM[Financial Reports]
    end

    subgraph Ops Portal
        OD[Ops Dashboard] --> OPV[Policy Verification]
        OD --> OCF[Claim Forms]
        OD --> OIF[Indorsement Forms]
        OD --> ODV[Daily Visits]
        OD --> OCA[Cancellation Forms]
    end

    subgraph Advisor Portal
        ADV[Advisor Dashboard] --> ADVP[Policy Creation]
        ADV --> ADVC[Customer Management]
        ADV --> ADVR[Reports]
        ADV --> ADVCOM[Commission Details]
        ADV --> ADVT[Targets]
    end

    subgraph Finance Portal
        FD[Finance Dashboard] --> FPP[Policy Processing]
        FD --> FCP[Commission Processing]
        FD --> FDL[Daily Ledger]
        FD --> FFR[Financial Reports]
        FD --> FPM[Payment Management]
    end

    subgraph CIC Portal
        CIC[CIC Dashboard] --> CICP[Policy Processing]
        CIC --> CICF[Form Management]
        CIC --> CICR[Reports Generation]
        CIC --> CICD[Document Verification]
    end
```

Master Components

```
graph TD
    subgraph Policy Related
        PT[Policy Types] --> MT[Motor Insurance]
        PT --> HT[Health Insurance]
        PT --> LT[Life Insurance]
        PT --> GT[General Insurance]

        PD[Policy Details] --> NC[NCB Details]
        PD --> PM[Payment Modes]
        PD --> PO[Payout Options]
        PD --> CS[Commission Slabs]
    end

    subgraph Employee Related
        EM[Employee Management] --> AT[Attendance Tracking]
        EM --> LM[Leave Management]
        EM --> SM[Salary Management]
        EM --> DV[Daily Visits]
        EM --> HR[HR Admin]
    end

    subgraph Finance Related
        FM[Finance Management] --> DL[Daily Ledger]
        FM --> CP[Commission Processing]
        FM --> PP[Payment Processing]
        FM --> FR[Financial Reports]
    end

    subgraph Document Related
        DM[Document Management] --> CF[Claim Forms]
        DM --> IF[Indorsement Forms]
        DM --> OF[Offer Letters]
        DM --> CN[Cancellation Forms]
    end
```

Component Relationships

```
graph TD
    subgraph Core Components
        CC[Company] --> BC[Branch]
        BC --> AD[Advisor]
        AD --> PL[Policy]
        PL --> CU[Customer]
    end

    subgraph Support Components
        SC[Staff] --> AT[Attendance]
        SC --> SL[Salary]
        SC --> LV[Leave]
        SC --> DV[Daily Visit]
    end

    subgraph Financial Components
        FC[Finance] --> DL[Daily Ledger]
        FC --> CS[Commission Slab]
        FC --> PY[Payment]
        FC --> CM[Claims]
    end

    subgraph Document Components
        DC[Documents] --> CF[Claim Forms]
        DC --> IF[Indorsement Forms]
        DC --> OF[Offer Letters]
        DC --> CA[Cancellation]
    end
```

Data Flow Between Components

```
sequenceDiagram
    participant UI as User Interface
    participant API as API Gateway
    participant AUTH as Authentication
    participant BL as Business Logic
    participant DB as Database
    participant NS as Notification Service

    UI->>API: Request
    API->>AUTH: Validate Token
    AUTH-->>API: Token Valid
    API->>BL: Process Request
    BL->>DB: Data Operation
    DB-->>BL: Response
    NS-->>UI: Send Update
    BL-->>API: Operation Result
    API-->>UI: Final Response
```

Component Access Control

```
graph TD
    subgraph Access Levels
        L2[Admin]
        L3[HR Admin]
        L4[Branch Admin]
        L5[Ops Admin]
        L6[Finance Admin]
        L7[Advisor]
        L8[Employee]
    end

    subgraph Access Rights
        AR1[Full System Access]
        AR2[Company Management]
        AR3[Employee Management]
        AR4[Policy Management]
        AR5[Financial Management]
        AR6[Report Generation]
    end

    L1 -->|Has| AR1
    L2 -->|Has| AR2
    L3 -->|Has| AR3
    L4 -->|Has| AR4
    L5 -->|Has| AR4
    L6 -->|Has| AR5
    L7 -->|Has| AR4
    L8 -->|Has| AR6
```

Component Integration

Frontend-Backend Integration

- RESTful API endpoints for each component
- JWT-based authentication
- Real-time updates using WebSocket
- File upload/download capabilities
- State management using Redux

Database Integration

- MongoDB collections for each component
- Relationship mapping between collections
- Indexing for performance optimization
- Data validation at schema level
- Automated backup system

External Service Integration

- Email service for notifications
- SMS gateway for alerts
- Payment gateway integration
- Document storage service
- Report generation service

API Component Flow Architecture

```
graph TD
    subgraph Client Side
        FE[Frontend Application]
        API[API Calls]
    end

    subgraph Server Architecture
        RT[Routes Layer]
        CT[Controllers Layer]
        MD[Models Layer]
        DB[(MongoDB Database)]
    end

    %% Main Flow
    FE -->|HTTP Request| API
    API -->|Request| RT
    RT -->|Route Handler| CT
    CT -->|Data Operations| MD
    MD -->|Query/Update| DB
    DB -->|Response| MD
    MD -->|Data| CT
    CT -->|Response| RT
    RT -->|HTTP Response| API
    API -->|Update UI| FE

    %% Routes Folder Structure
    subgraph Routes Directory
        MR["/routes/routes.js"]
        AR["/routes/announcement/*"]
        CR["/routes/cancelForm/*"]
        DR["/routes/dailyVisits/*"]
        HR["/routes/hrsalary/*"]
        LR["/routes/letters/*"]
        UR["/routes/user_routes/*"]
    end

    %% Controllers Folder Structure
    subgraph Controllers Directory
        MC["/controller/*.controller.js"]
        AC["/controller/advisor/*"]
        CC["/controller/CC/*"]
        FC["/controller/finance/*"]
        HC["/controller/hradmin/*"]
        NC["/controller/ncb/*"]
        UC["/controller/user_controller/*"]
    end

    %% Models Folder Structure
    subgraph Models Directory
        SM["/models/*.Schema.js"]
        AM["/models/advisor/*"]
        CM["/models/cc/*"]
        FM["/models/finance/*"]
        HM["/models/hr/*"]
        NM["/models/ncb/*"]
        UM["/models/user_models/*"]
    end

    %% Example Flow for Employee Operations
    subgraph Example Flow
        ER["/routes/routes.js"]
        EC["/controller/addemp.controller.js"]
        EM["/models/addempSchema.js"]
    end

    %% Connections
    ER -->|Routes to| EC
    EC -->|Uses| EM
    EM -->|Interacts with| DB
```

API Flow Examples

Employee Management Flow

```
sequenceDiagram
    participant Client
    participant Route as routes.js
    participant Controller as addemp.controller.js
    participant Model as addempSchema.js
    participant DB as MongoDB

    Client->>Route: POST /dashboard/addemployee
    Route->>Controller: addEmployee()
    Controller->>Model: create()
    Model->>DB: Insert Document
    DB-->>Model: Success/Error
    Model-->>Controller: Result
    Controller-->>Route: Response
    Route-->>Client: HTTP Response
```

Policy Management Flow

```
sequenceDiagram
    participant Client
    participant Route as routes.js
    participant Controller as addpolicy.controller.js
    participant Model as addpolicySchema.js
    participant DB as MongoDB

    Client->>Route: POST /dashboard/addpolicy
    Route->>Controller: createPolicy()
    Controller->>Model: create()
    Model->>DB: Insert Document
    DB-->>Model: Success/Error
    Model-->>Controller: Result
```

Detailed API Documentation

Authentication APIs

```
sequenceDiagram
    participant Client
    participant Auth Controller
    participant JWT Service
    participant Database
    participant Email Service

    Client->>Auth Controller: Login Request
    Auth Controller->>Database: Validate Credentials
    Database-->>Auth Controller: User Data
    Auth Controller->>JWT Service: Generate Token
    JWT Service-->>Auth Controller: JWT Token
    Auth Controller->>Email Service: Login Notification
    Auth Controller-->>Client: Token & User Data
```

Login Endpoints

Endpoint	Method	Description	Request Body	Response
/loginadmin	POST	Admin login	{email, password}	{token, userData}
/login/employee	POST	Employee login	{email, password}	{token, userData}
/hradmin/login	POST	HR login	{email, password}	{token, userData}
/branches/loginbranch	POST	Branch login	{email, password}	{token, userData}

Policy Management APIs