## 07 Child Health Record Booklet

*Empowering Field Agents to Capture Critical Child Health Data Anywhere*

Design a desktop or mobile application that enables offline health data collection for children in remote areas, with secure local storage. The app must support authenticated data upload via eSignet once internet connectivity is restored.

**Complexity Level: Low**

### Overview

In many rural, underdeveloped, and disaster-affected regions, millions of children suffer from malnutrition. To combat this, UNICEF, in partnership with local governments, dispatches health workers and volunteers to remote areas—places where internet access is unreliable or nonexistent.

These field representatives must collect critical health data: children's demographics, nutrition indicators like visible symptoms of malnutrition, recent illness history, and parental consent. Often, this data must be gathered completely offline and stored locally until the representatives reach an area with internet connectivity. At that point, the representatives must authenticate using their national ID via eSignet (OTP-based verification) and upload the stored data securely.

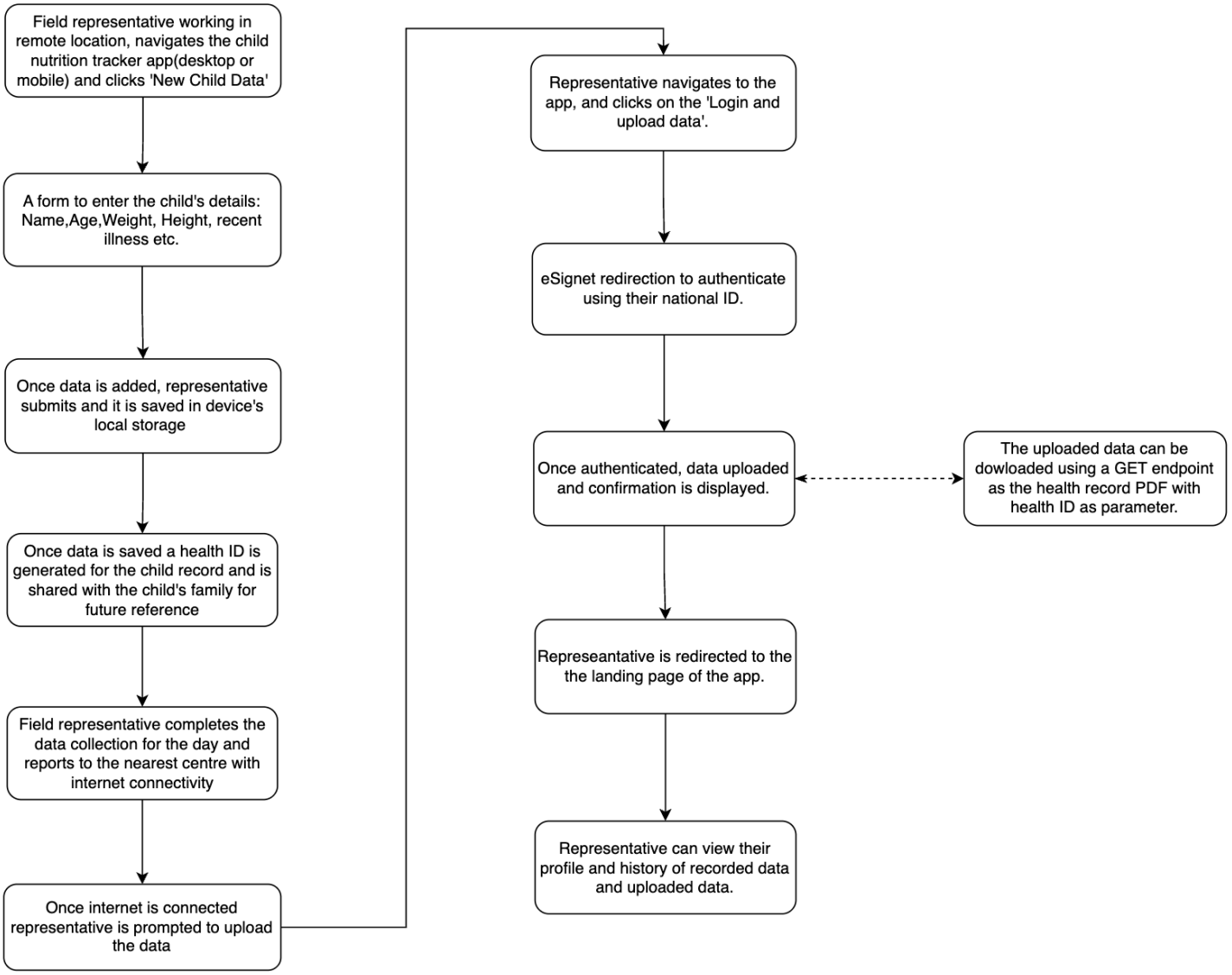
Your challenge is to design and develop a desktop or mobile application that supports this full workflow offline data collection, secure local storage, and authenticated data upload once the internet becomes available.

### Recommended Tech Stack

**For Web/Desktop App:**

1. **Frontend**: React / Angular
2. **PWA Support**: Workbox.js / IndexedDB
3. **Backend (for optional dashboard)**: Node.js / Django / Flask/ Java
4. **Database**: PostgreSQL / MongoDB
5. **Authentication**: eSignet integration (mock or real)

## User Journey



## Exact Tasks

**<u>Mandatory Tasks</u>**

**Mobile/Desktop Application:**

**1. New Child Data Collection Form**

a. Allow representatives to fill in a form for new child data collection.
b. Form Fields**:**
    i. Child's Name
    ii. Face Photo (Capture or Upload)
    iii. Age
    iv. ID (Unique identifier or local tracking)
    v. Child's Weight
    vi. Child's Height
    vii. Parents/Guardian's Name
    viii. Visible Signs of Malnutrition
    ix. Recent Illnesses
    x. Parental Consent (Checkbox: must be checked to proceed)
    xi. Fields like malnutrition, illnesses: Allow "N/A" or "Skip" if not applicable.

**Note:** The list of fields provided is suggestive, not exhaustive. Additional fields may be added or existing ones removed based on the specific context and use case requirements.

**2. Offline Data Collection Support**

a. Allow data collection even when the device has no internet access.
b. Locally store the collected child records securely.
c. Generate a unique **Health ID** for each child at the time of saving the record.
d. Share the generated Health ID with the child's family for future reference.

**3. Data Upload & Sync**

a. Detect when internet connectivity is restored.
b. Prompt the representative to upload all previously saved (offline) records.
c. Authenticate via eSignet using National ID and OTP before uploading.
d. Ensure secure transmission of all stored records to backend/simulated server.
e. Confirm and notify successful upload to the representative.

**4. Data History and Tracking**

a. Display list of all saved child records (whether uploaded or not).
b. Display status and confirmation of all uploaded records.
c. Notify and mark child records as "Uploaded" once syncing is successful.

**5. Representative Profile Section**

a. After successful authentication with eSignet, allow representatives to view their profile information.

**6. GET Endpoint for Child Health Booklet**

To be developed alongside the mobile/desktop app.

a. Create a new **GET API endpoint** to fetch the child's health record booklet in **PDF format**.
b. Use the **Health ID** (generated during offline collection) as a parameter in the request.

**Good-to-Have Tasks**

Participants are encouraged to extend their solution by developing:

1. **Multi-language Support**
   Multi-language support, ensuring the app is accessible to field representatives in different regions and languages.
2. **Retry Mechanism for Uploads**
   Implement a retry mechanism for failed uploads due to connectivity issues, ensuring no data is lost.
3. **Geo-location Tagging**
   Geo-location tagging, allowing survey data to be associated with a location using the device's GPS (if available).
4. **Local Data Encryption**
   Local data encryption, to enhance privacy and security of sensitive health information stored on the device.

**Bonus Tasks:**

**1. Admin Portal**

a. **View & Access Uploaded Data:**
Build an admin portal that allows admins to view the data uploaded by field representatives.
b. **Filter & Search Functionality:**
Provide filters (by date, location, representative ID, etc.) and search options to quickly access specific records.

**2. Informative Dashboards & Insights**

a. **Visual Dashboards:**
Create dashboards within the admin portal to visually represent trends and patterns using graphs, charts, and tables.
b. **Data-Driven Insights:**
Use the collected data to highlight key metrics such as malnutrition cases, regional health trends, and field activity levels.
c. **Child Health Record Access:**
Enable admins to download a child's health record booklet in PDF format by entering the child's Health ID.

## Deliverables

Participants are required to submit the following comprehensive deliverables to successfully complete the challenge:

**1. Mobile/Web Application**

a. **Fully Functional Web Application** – Provide a complete web application with working desktop and mobile support.

**2. API Endpoints**

b. **Download Child Health Booklet** – Provide a GET API endpoint to download the child health record booklet by passing the Health ID.

**3. Demonstration & Code**

a. **Demonstration Video** – Share a video demonstration that showcases the working solution end-to-end.
b. **Source Code Repository** – Provide access to the full source code through a public repository (GitHub/GitLab).
c. **Presentation Deck** – Include a PPT summarizing the approach, results, and potential impact of the solution.

**4. Documentation**

a. **Setup and Installation Guide** – Step-by-step instructions to build, deploy, and run the application locally or on supported devices.
b. **Architecture Overview** – Description of the overall system architecture, data flow, and storage strategy.
c. **Technology Stack** – Explanation of all technologies, frameworks, and tools used in the solution.
d. **Features Summary** – Concise summary of all implemented features, both required and additional.
e. **Assumptions** – Provide a detailed list of assumptions made during development.
f. **API Documentation** – Detailed description of the new API endpoints developed as part of the solution.

## Resources:

1. eSignet code base - GitHub Link
2. Sandbox environment:
    a. eSignet with mock ID
    b. eSignet with MOSIP ID
3. Self Registration Portal for generating UIN: Link
4. eSignet partner onboarding form: Link
5. eSignet partner integration guide: Link
6. Collab end user guide: Link