
HL7® FHIR®
FOUNDATION

# Gantt Documentation including sub-tasks

1. **Define the FHIR resources:**
   The task includes defining the different IMNA's resources that are specific to diabetes and obesity patients, such as self-measured vital signs, home glucose tests, etc.

   Subtasks:
   1.1 Understanding and selecting the metadata and content for exchange.
   1.2.Understand the connections between each resource and resource, for example, refer an observation resource such as fasting blood test to a medicationRequest resource such as Metformin prescription and so on.

2. **FHIR Interface Architecture Design:**
   The task includes designing the appropriate architecture for the FHIR interface

   Subtasks:
   2.1 Define the interactions to be supported (e.g. posting bundles as collections of patients and observations, or searching and reading patient observations, etc.
   2.2 Define the data formats (JSON, BSON, etc).
   2.3 Define IMNA's API Gateway and Map FHIR resource types to the corresponding API

Gateway resource types. Also, define how to handle and implement URL mappings and HTTP methods such as GET., POST, etc, e.g. with Lambda functions.

2.4 Choose a database and indexing - Create FHIR data repository and store it in a NoSQL database which best provides the input data representation and query patterns necessary for a FHIR data repository such as MongoDB, DynamoDB, etc. Also, choose the right indexing (e.g. global secondary indexes) in order to perform searches and retrieve observations for a patient.

2.5 Choose the storage infrastructure in order to store archived FHIR messages, with low cost but also high durability.

2.6 Processing FHIR messages - Choose how to handle API Gateway requests in the Architecture, e.g. using Lambda function with a RESTful framework, serverless framework, HAPI FHIR, etc.

2.7 Deploying the FHIR Interface - Choose how to deploy the FHIR resources for this Architecture,

2.8 Building the client application and creating a secured access to the API Gateway - Choose an appropriate client application that calls the APIs on the FHIR server, and design the application. E.g., it can be a simple Node.JS app that gets a list of patients and related observations, or a more advanced app such as a patient-focused application that displays vitals and immunization charts.
Also, define how to generate the authorization tokens for a secured access to our API Gateway, such as Okta, Auth0, Cognito, etc.

## 3. Profiles Design:

Design appropriate profiling/extensions for the various resources. Choose from various implementation guides such as PRO or in SIMPLIFIER.NET such as the Israeli Core project.

## 4. Test the FHIR server:

Provide QA to the FHIR server, to make sure that all components are running smoothly and efficiently.

Subtasks:

3.1 Execution setup - Establish the testing environment by preloading data required for the execution of the tests against the FHIR system under test.

3.1 Test execution - Test API endpoints, evaluate secure access, and test typical operations such as create, update, and read.

3.2 Creating a QA report.

## 5. Deploy a working, in production, FHIR interface:

Upon successfully completing the FHIR server tests, deploy the FHIR interface as stated in the architecture. Maintain a continuous test and prod environments.

6. **Create a summary report on the project deliverables which includes:**
   5.1 The project Summary
   5.2 Methods and Deliverables - Critical meetings attended, design and implementation documents, publications for the FHIR community, and next steps.

**Important remark** - For every task, document all meetings agendas and recaps, as the part of the deliverables.