# openIMIS StarterKit

# Requirements Analysis

# Defining your digital system user requirements

Defining your business requirements through consultation with key stakeholders is an important step to ensuring that the technology solution will meet those needs. It is critical to document and gain consensus on what the priority needs are. These requirements will be used to guide the process of evaluating possible solutions and inform the decision as to “adopt, adapt or develop”.

**Business Requirements** are high-level descriptions of what the health financing sector requires in terms of a digital solution and information in order to meet the needs of the beneficiaries and fulfil its mandate.

**User requirements** are clearly articulated statements of what a digital system must be able to do in order to satisfy the needs of users and beneficiaries. These should be derived from business requirements. They should be defined in two clear categories, functional and non-functional. Functional requirements describe the required **behaviour and functions** of the system. Non-functional requirements describe specific criteria that can be used to judge the **operation of a system** e.g. performance, security, availability as well as additional considerations such as data migration requirements, and requirements to integrate or interoperate with other systems.



## Business Process Modelling

For each main business process it is useful to document the process using a standard business process modelling technique. This can assist with identifying issues and gaps in the current process to assess where a digitised solution can add the most value.

* Use a business process modelling tool and the **Business Process Modelling Guide** to develop a visual model of your processes flow
* (free business process modelling tools are available to download e.g. Bizagi <https://www.bizagi.com/>).
* For each of the business processes modelled, document additional information for each step as shown in the **Business Process Modelling Guide. [additional document to download]**

## Requirements Gathering Guidelines & Templates

###

There are a number of tools and techniques that can be used to gather and document this information. Remember that these artifacts are **a means of communicating to different stakeholders** i.e. end users, project sponsors, system architects, developers, testers, monitoring and evaluation experts, etc.

● Keep them simple

● Keep them consistent

● Make sure terminology is defined, simple and consistently used throughout; for example: Provider, Practitioner, HCW, Clinician may all refer to the same type of user - choose one and stick to it

Correct notation, such as Business Process Modelling Notation (BPMN) is important if you are communicating with peers who already understand it but for general audiences the aim is **a** **clear and unambiguous shared understanding** of what you are trying to build and the format is less important.

###

### Gathering Requirements

The most important factor is to have access to the right people. Be sure to include the people who will be working with the system on a day to day basis, not just the senior management team or decision-makers. They should include data capturers, claims processing clerks, system administrators, etc.

There are various ways to elicit and define the requirements, including:

● Brainstorming

● JAD (Joint Application Design) workshops

● Small focus groups

● One on one interviews

● Questionnaires

● Observations

● Document analysis

Be aware of the **power dynamics** in the room during large group discussions

Ideally you need to be able to speak to the actual end users of the proposed solution, preferably one on one or in small groups without supervisors/manager present

**Don’t over-promise** : make it clear you are there to understand what is needed and that many other factors influence the final scope

Ensure you have participants’ consent to record sessions or to take photographs

Try to have one or more scribes to assist with taking notes

The **Requirements Template** can be used as a simple tool to define high-level **functional requirements** and **non-functional requirements.**

###

### User Personas

###

A user persona is a representative for each type of user that may interact with the system, such as a beneficiary, a claims processor or a manager.

The **User Personas Template**­ can be used to identify key characteristics of the user and describing them in such a way allows a better understanding of their wants, needs and common frustrations so that these can be addressed effectively. Using input from user research at the focal point of design decisions ensures that the system works in such a way that fulfils user needs.

**

***Template link:*** <https://docs.google.com/document/d/1QDmdUpl1-WmBLxuuphXI7y8hz1-GDFB2SkVFuAqxvZw/edit>

 Types of Requirements

● Functional i.e. what must the system do

*Example of a HIGH-LEVEL functional requirement:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number** | **Requirement** | **Role** | **Requirement Description** | **Process Map** | **Priority****(High,Medium,Low)** | **Complexity (High,Medium,Low)** |
| **Add Family/Group** |  |
| **FR1.1** | Create a Family/Group | Enrolment Officer | The User is able to create a family/group by entering a list of relevant details on the form. (**S*ee data dictionary section*** *for the details displayed with respective formats.)*  | I1. |  High  | High |

A good practice would be to document the functional requirements in a user story.

### User Stories

For User Stories, it is useful to follow a Behaviour Driven Design approach eg:

**AS A**  <specific user/persona/role>

**I WANT**  <desired feature/issue that needs to be solved>

**SO THAT**  <benefit from implementing feature>

The story should include, where appropriate:

○ Acceptance criteria - these form the basis for the test cases

○ UX mock-ups / UI designs

○ Logging and auditing

○ Performance criteria

○ How to demo

**AS AN**  Enrolment Officer

**I WANT**  to be able to create a family/group for a patient/member

**SO THAT**  All relevant information pertaining to the family can be captured, such as;

* Region
* District
* Municipality
* Village
* Poverty Status
* Confirmation Type
* Family/Group Type
* Permanent Address Type
* Insurance Number
* Other Names
* Last Name
* Birth Date
* Gender etc…

And once this has been captured, Insurees, Policies & Contributions can be assigned to the family/group created.

### Define the Information Requirements

Before being able to define what digital systems are required to support the health insurance business need, it is necessary to understand what information requirements exist i.e. what data is collected, stored and put to use within the existing system.

The type of questions to consider are:

* What type of information is required, at the operational , management and reporting levels?
* Who owns the data?
* Where will master data be stored?
* How and where will each type of data be created, stored, transported and reported?
* What data transformation is required to support the information exchange between solution components?

Informational requirements refers to reports, data visualisation, dashboards. These requirements are often gathered from stakeholders rather than users. It is also more demanding and difficult for the analyst to obtain these requirements accurately, because it invariably requires much more in depth business knowledge. Reports may include, but not limited to:

* 1. Operational reports e.g. Primary Operational Indicators-claims
	2. Management reports or dashboards e.g. Product Sales
	3. Exception reports e.g. Rejected Photos
	4. Control reports e.g. Matching Funds
	5. System Monitoring e.g. User activity Reports, Status of Registers

*EXAMPLE USING TEMPLATE*

|  |  |
| --- | --- |
| **IR03** | **Detailed Report Requirements** |
| **Report Category** | **Management Reports** |
| **Report Name** | **Product Sales** |
| **Report Description** | *Brief descriptions of the contents of the report*This report contains the value of sales of insurance policies by insurance sales agents/enrollment officers. |
| **Purpose** | *Describe the purpose of the report*The report gives an overview of the state of the sales activities. It shows within the chosen time frame the sales volumes (financial collections from sales) which helps analyze the financial performance of different regions and districts. Each financial transaction related to policy sales are listed in this report which can be used to cross verify payments received by the scheme for the chosen time period. |
| **Audience** | Scheme Manager, Accountant and Enrolment Officer |
| **Triggers** | *Accountant requires sales report to cross verify payments received in the insurance office**Management required Product Sales reports on a monthly or ad hoc basis.* |
| **Input parameters** | *The user can select to filter the report using the following Parameters:*

|  |  |
| --- | --- |
| **Parameter** | **Options** |
| Date From | Date picker |
| Date To | Date picker |
| Region | List of regions |
| District | List of districts in the selected region |
| Product | List of products or services |

 |
| **Sort Sequence** | **Default sort**: *Other sort options e.g. allow the user to sort by any of the columns* |
| **Report Headers** | **Report Name**: **Report Parameters**:  |
| **Report Content** |  *Describe the columns and expected data per column* |
| **Report layout** | *Add an example of what the report will look like with column headers and example data / add a wireframe* |
| **Report Footer for printed reports** | *User (that ran the report)**Report Name -* ***Product Sales****Date Created -* ***9/12/2019****Date and Time Printed (if printed) -* ***9/12/2019 12:15:28 PM****Number of pages -* ***1 of 1*** |
| **Export** | *Allow user to export reports to .xls or csv, PDF or Word* |

### Define the Non-Functional Requirements

Define the full list of **non-functional requirements** considering required operational standards and non-functional standards provided below. Defining a comprehensive list of non-functional requirements mitigates the risk of the system not performing as expected, allowing you to define performance standards.These may include:-

**Performance related observable requirements** -These requirements allow you to define how you want and need the system to perform within defined parameters to ensure high quality performance, minimise down-time and fulfil user needs.

* Reliability
* Availability
* Usability
* Security

**Requirements that support system evolution over time**: These requirements allow you to define ways in which the system can be adapted and evolve as the number of users and amount of data in the system increases and requirements further develop.

* Scalability
* Adaptability
* Maintainability
* Extensibility

*Example of a Non-Functional Requirement for Performance:
NFR01 Allow users to find features within 3 clicks or less.*

*Example of a Security Matrix for a Non-functional Requirement that defines which users will be able to perform certain functions within the system.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Permissions | Enrolment Officer | Manager | Accountant | Clerk | Medical Officer | Scheme Administrator | IMIS Administrator | Claim Administrator |
| Add Family/Group | X | X |  |  |  |  | X |  |
| Edit Family/Group | X | X |  |  | X | X | X |  |
| Delete Family/Group | X | X |  |  |  |  | X |  |
| Add Insurees | X | X |  | X |  |  | X |  |
| Edit Insurees | X | X |  | X |  |  | X |  |
| Delete Insurees | X | X |  | X |  |  | X |  |
| Add Policies | X | X | X | X |  | X | X |  |
| Edit Policies | X | X | X | X |  | X | X |  |
| Delete Policies | X | X | X | X |  | X | X |  |
| Renew Policies | X | X | X | X |  | X | X | X |
| Add Contributions |  | X | X |  |  | X | X | X |
| Edit Contributions |  | X | X |  |  | X | X | X |
| Delete Contributions |  | X | X |  |  | X | X | X |

### Define data migration requirements

## What data will need to be migrated to the new system?

## What level of data cleaning and transformation is required to ensure that the data meets the requirements and constraints of the new system?

*Example of Transitional Requirements for Data Migration:
NFR03 The system must provide data migration functionality through the scanning of existing paper based records.
NFR04 by providing the ability to back capture existing paper records through manual data entry and flag as legacy records. The system must have the ability to enter existing insurance numbers to historical birth, death, marriage and divorce records that are migrated from existing paper-based and electronic records.*

### Define Integration and Interoperability requirements

* What existing systems does the new system need to integrate with?
* What data needs to be exchanged?

*Example Integration / Interoperability requirements:*

*NFR05 The system must have the ability to securely interconnect with the central vital events registry, NIA, master person index etc. via a secure web-based interface or mobile application. The system must ensure that incoming data input is validated, evaluated for expected size, format and type before acceptance.
NFR06 The system must use open standards to promote interoperability.
NFR07 The system must support standard messaging protocols.
NFR08 The system must provide real-time response to mobile transactions submitted to the central database.
NFR09 The system must be able to interface with open source or existing third party reporting tools.
NFR10 The system must provide the capability for integration with other systems through an API.*

### Develop a Data Dictionary

The Data Dictionary is a centralised repository of information about data such as meaning, relationships to other data, origin, usage, and format. A Data Dictionary is a set of information describing the contents, format and structure of a database and the relationship between its elements used to control access to and manipulation of the database.

*Example of a data dictionary using the template*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Entry Options** | **Data Type****(Numeric, text, date, alphanumeric, Y/N, Length of field)** | **Rules** |
| **Region** | Region to be selected by user | Dropdown  | Text(50) | Display a dropdown list of regions that the user will select |
| **District** | District to be selected by user | Dropdown | Test(50) | Display a dropdown list of districts only associated with the selected region for the user to select |
| **Municipality** | Municipality to be selected by user | Dropdown | Text(50) | Display a dropdown list of municipalities associated with the selected district for the user to select |
| **Village** | Village to be selected by user | Dropdown | Text(50) | Display a dropdown list of villages associated with the selected municipality for the user to select |
| **Poverty Status** | Poverty status of patient | Y/N | Y/N |  |
| **Confirmation Type** |  | Dropdown | Text(50) |  |
| **Family/Group Type** |  | Dropdown | Text(50)  |  |
| **Permanent Address Details** | Address of patient | Free Text | Text(Unlimited) |  |
| **Confirmation No.** |  | Auto Generate | Numeric(11) |  |

##

# Finding the right system for you

##

## Evaluating potential solutions

1. First decide what type of criteria you are going to use to evaluate possible solutions. For example, some common criteria are:
	1. Financial sustainability - consider both initial and long-term costs
	2. Ease of implementation
	3. Ease of maintenance and support
	4. Accuracy of data
	5. Timeliness and relevance of data
	6. System reliability
	7. System performance
	8. System security
	9. System recoverability

Make sure that key stakeholders who will be using and managing the system on an operational basis are included in this evaluation process.

1. Prioritise or weight these criteria as it is very unlikely that any solution will meet all criteria fully and choices will need to be made according to which are more important.
2. Investigate the possible solutions and rate them according to how well they meet your criteria
3. Consider whether you want to develop and maintain the system internally or whether you should procure the system from an external vendor. These are the type of options you could consider:

|  |  |
| --- | --- |
| **PROS** | **CONS** |
| **Out-of-the-box software** |
| * Lower up-front costs
* Know what you’re getting
* Shorter delivery timescale
* Support often included
* Upgrades often free/at a reduced cost
* Already tested/refined through other implementations
* Community support available (through forums & expert users)
 | * May have to adjust processes to meet software limitations
* Feature requests ignored if larger customer base do not demand it
* High customisation fees
* If costs are charged per user, costs can be very high
 |
| **Custom-developed software** |
| * Get what you need/want
* Freedom to change the software to align with business needs
* Built with your business and employees in mind
* Potential to engage local IT industry
* No licensing costs
* Ability to brand the software
* Specific application support from people who know the platform
 | * High up-front costs
* All changes to the software come with an associated cost
* Software might still not fulfil all needs/wants
* Dependent on technical capabilities of the team hired to develop
* Support dependent on availability of developers and people who know the custom software
 |
| **Open Source software** |
| * Few, if any, licensing fees.
* Easy to manage due to the absence of licensing requirements
* Continually evolving as developer add and modify it
* Ability to update the software to meet the needs of your business
* Not tied to a particular vendor’s platform that only works with their other systems
 | * No guaranteed support, dependent on community of users to respond to and fix problems
* Software can be orphaned when developers stop updating it
* Evolves with developer’s wishes rather than user/business needs
* Malicious users could negatively update the software
 |
| **Cloud Hosted Solution** |
| * Cost-effective – lower up-front costs, removes need to buy expensive software and pay for licensing and lower traditional server costs;
* Reduces the need for specialised skills to maintain the service;
* Accessibility – allows access from multiple platforms;
* Adaptability – enables almost immediate use without application setup and installation;
* Data centralisation – all your data in one place that can be accessed remotely
* Scalability – allow for easier and more flexibility scalability to cope with increased transaction loads as and when needed
* Cloud security
* Provides a flexible testing environment
 | * Low bandwidth will negatively affect performance;
* Lack of insight into your network – difficult to resolve bugs;
* Data protection legislation and/or government policies may prohibit the use of cloud-based data storage
 |