# Annex 1: Terms of Reference\_Deliverables

Beneficiary Management System

## Overview

This document will outline the functional and technical specifications for the development of a Beneficiary Management System (BMS). The purpose of the BMS is to enable the registration, import, storage and management of beneficiary information through a hosted web application. This system will provide an intuitive, easy to use platform for managing humanitarian and development projects, including the provision of selective distribution lists for transferring money and goods in humanitarian relief scenarios.

## Example Scenario

### Unconditional Cash Distribution, Cambodia

Following on from a disaster in which the lives of tens of thousands of people are affected, it is determined that a direct injection of cash will serve the dual purpose of providing food security for affected households, while also providing a measure of stability for the local economy. A donor funded project has been set up to distribute cash to particularly vulnerable groups in an socio-economically depressed region of the country, with a significant percentage of vulnerable households. The project requires the distribution of $120 to 10,000 households that meet a specific list of criteria in a defined period of time and requires reporting on the success of the distribution.

## Functional Overview

“The Beneficiary Management System” or “BMS” is the place name of a new product to be developed for People in Need in early 2018; the actual name of the end product will be different.

These Terms of Reference are a starting point for the design of the Beneficiary Management System, not a final blueprint. Although we will endeavour to include as much detail as possible in this outline, the system must be built as a collaborative project between People in Need and the developers. It is expected that through a consultative process, minor features and changes will be made over the course of the development period, using an iterative approach to delivery, meeting the needs and expectations of PIN with regular feedback gathered dynamically over the course of the project.

For the purpose of this documentation, we will define a ‘user’ as a member of People in Need staff who will be operating the platform. And a ‘beneficiary’ in this instance, will be the recipient of the distributions and the people whose demographic data we will be storing.

## High Level Requirements

* The BMS architectural design and technology needs to be robust, scalable, modular and adaptable, supporting future development and growth.
* The platform should be built using open source tools and technologies where possible, and built with the Principles for Digital Development in mind. https://digitalprinciples.org/
* The platform must offer an intuitive, user friendly experience and provide a functional, fast and efficient way to manage beneficiary data.
* The development of the BMS must be conducted in a collaborative and transparent manner, with good communication between PIN working group and developers, using a continually updated online code base and implemented with a flexible project management methodology (for example Agile).

## Major Components

### Browser Based, Responsive Web Application

An easy to use, private, mobile compatible, web based application is required to manage the system overall and provide an intuitive and easy to use interface for all features of the platform, including beneficiary registration and management, cash transfer functionality, reporting tools and administrative functions. The web application UI should be adaptive to three major platforms, Mobile, Tablet and Computer (anything sized 1280x1024 and above).

Including;

* Email based username and password login for users.
* A side menu (or charm bar), providing easy access to all program elements.
* A search bar on every page for finding beneficiary, project, distribution and transaction data.
* The front page (or dashboard) must display at a minimum, the following mission and project summary information.
	+ Number of active projects.
	+ Number active beneficiaries.
	+ List of active distributions.
	+ Total Budget being distributed.

### Beneficiary Information Management

A secure, robust database must be built, allowing users of the web application to create, view, edit, delete, bulk import and export (based on privilege level) beneficiary data.

In more detail;

* Registration creates a global digital record for every beneficiary, with unique ID, including personal details and a defined number of vulnerability criteria (outlined in Annex A).
* Beneficiary records must be viewable in both table form as a list, or in individual form as a separate record with photos/id scans etc. if available.
* Manual registration of individual beneficiary data can be done via the application, either on a desktop or compatible mobile device with browser or;
* Users can import beneficiary data via CSV or API request from external source.
	+ De-duplication functionality must be provided. (Developers to provide concept in proposal. Ideally duplication checks can be done )
	+ Field matching for imported information must be provided. (Developers to detail in proposal.)
* Basic filter and search functionality.
* Version / Control tracking – tracks list of changes made by different users.
* Database must be divided by individual and household record sets. A household must contain one individual (Household Head) as a minimum. An individual must be a member of a household (and can only be a member of one household).
* Datasets for beneficiary data are fixed for this stage of the software build, but must be expandable and customisable in future.
* Users must be able to add and replace images for beneficiary data. For example, for National ID scans or other identifying information.

### Project Management

An easy to use project management application to create, view and edit projects.

Including;

* Unique Project ID.
* Donor information.
* Total Project Allocated Funds for Distribution.
* Individuals or households are enrolled in projects defined in the project setup.
* Households or individuals can be enrolled in multiple projects, but must be enrolled in a minimum of one. So either via manual registration, or for bulk beneficiary imports, a project code is applied.
* A summary page containing basic tabular data.
* Filter and search functionality.

### Distribution Function

Create, view, edit and delete separate ‘Distributions’, which provide for the division of the project into distribution points, defined by geographic location, and vulnerability qualification.

Including;

* Automated decision tree for determining qualifications for cash distributions based on selected vulnerability criteria.
* Vulnerability definition and assessment.
* Generate a distribution group, based on user defined selection criteria calculated from the stored beneficiary data (collected manually or imported). i.e. Cash distribution to IDPoor Cat 1 households with single parent and at least one female dependent aged 13 – 18.
* Define the total amount of cash to be distributed, and define the cash transfer methodology.
* Allow for manual ‘double checks’ of beneficiary data and then the addition or removal of individual households from each ‘distribution’.
* Display of final Distribution List, which can then be used to send to the Cash Transfer Application.

### Cash Transfer Application

The system must be able to connect via API to WING mobile banking services in order to transfer money via mobile phone to beneficiaries.

Once a final distribution list has been created in the Distribution function of the application.

* Connect via a RESTful API to send bulk transaction requests to existing WING accounts (these will be pre-registered accounts for all beneficiaries in the project).
* Using beneficiary mobile phone number, send cash directly via WING to beneficiaries from connected Organisation account with existing balance.
* Authorization approval for distribution of cash to mobile banking service.
* Transaction monitoring and reporting.

### Reports

A basic set of default reports must be provided. Can be tabular and text based, but preferably simple visual reporting as well.

* Project Summary
* Household Summary
* Country Overview
* Transaction Summary

### Export Function

Export reports and beneficiary data.

* Individual and household data in CSV or JSON format.
* Basic reports as outlined above.

### Administrative Settings

An administrative settings page must allow for the following functionality for privileged users;

* Extensive user management, including password control.
* Beneficiary database field management
* Financial Service Provider settings
* Donor table
* Sectors table
* Server Settings

## Functional Workflow

##

## Technical Specifications

### Overview

**Cloud Hosted:** In order to make the BMS easy to deploy and administer anywhere in the world, the system should be hosted on a ‘cloud’ solution, providing auto scaling to deal with sudden increases in capacity requirements.

**HTTP REST API:** The system must be built with a robust, custom API which can integrate easily with a variety of 3rd party applications, including cash transfer platforms and The Hub API integration tool for polling IDPoor data.

**Flexibility:** By leveraging the latest tools available in cloud hosting platforms like ‘Docker’ containerisation, should provide for the modular build that we require.

**Server & Client Architecture:** The server and client tools must be determined by the developers and allow for further development in future for the open source community.

### Hosting

The developers will design and build the front end web application for the BMS based on the requirements specified in this documentation. The website will:

* Have a response web design (Bootstrap UI framework for example)
* Be developed using a HTML5 Web Application Framework
* Be able to consume RESTful web services from external applications (Cambodia IDPoor API for example. http://www.idpoor.gov.kh/en/data-users/8/8)

### Hosting

Amazon Web Services Hosting. As we already have an existing partnership with AWS and free hosting currently, it would make financial and administrative sense to continue to use their services. However, if a cost effective alternative can be proposed, then it should be included.

This means a combination of:

* + Elastic Beanstalk Instances - For instant scaling and flexibility when required.
	+ Docker – Containerisation allowing ease of extension and standalone modular construction.

### Database

Stable, secure, scalable, fast, cloud hosted database to store all required beneficiary information. A full database schema must be included in the proposal.

* PostgreSQL or equivalent
	+ AWS DynamoDB or equivalent (To be specified by the developer).

### HTTP REST API

Robust REST API to connect to application which can allow for management of beneficiary data, projects and money transfers.

* Provide a stable, HTTP REST API which means that the system can be more easily integrated with third party programs, which provides the ability to securely share specific data across agencies and allows for a more modular design, creating flexibility and a stage installation approach.
* Connect to mobile banking services to transfer money
	+ The system must provide functionality to connect to WINGs transaction API to directly transfer money to beneficiaries via the BMS application.

Authenticated API: The API Will use basic authentication token based authentication, ensuring user and application data remains secure. Specifically, API authentication should be split into 4 groups

* Unauthenticated: API calls that generate page elements ALL users can see, via the dashboard, regardless of login status.
* User APIs: APIs that *only* authenticated users can access, using authentication over HTTPS.
* Admin APIs: APIs that ONLY authenticated admins can access (via the client, for rendering admin pages and for admin only tasks e.g. user addition / deletion) using HTTP basic authentication over HTTPS
* Token Based APIs: The BMS needs to integrate with other platforms, including mobile data collection tools, IVR platforms, finance applications and other beneficiary management systems. Therefore, the system must allow for token based APIs.

Fully Documented API: Swagger UI (or similar) should be used to generate working requests from the code, allowing for developers and third parties to integrate with the BMS API.

### Languages

Server Side

* To be determined by developers (Python 3.6 or similar)

Client Side

* To be determined by developers (Angular.js or similar)

### UX and UI Design

Mobile and Web Application. Preferably HTML5 (to be determined by developer).

For this first stage of the project, a simple bootstrapped UI can be implemented, in order to ensure that the program functionality can be developed within the budget of the project. Basic wireframes have been created (see here: <https://www.fluidui.com/editor/live/preview/cF93Y1JoTEVhWlg2WnBtQTl3ZHJzOGRQUXhXNlVTdkJhWg>==) , but further consultation with the working groups and User Acceptance Testing throughout the iterative development process will be required in order to create a simple, flair free, but intuitive and easy to use interface.

Must be compatible with all modern browser versions of Google Chrome, Apple Safari, MS Edge and Mozilla Firefox and backwards compatible to at least 2015.

### Search & Filter Functionality

Primary database tables including individual and household data and project and distribution data should offer search functionality and filter functionality by key attributes.

### Security

Security of beneficiary data is critical to the functionality of the BMS.

* No passwords should be stored in plain text.
* All form data should be validated.
* Hosted database must be encrypted.
* REST API secured via OAuth2 Standard over TLS, including function to generate shared key.
* Application served over HTTPS with Trusted Certificate (PIN to provide).
* Application should be protected against risks listed in OWASP TOP 10 (XSS, CSRF, …)
* All passwords must be stored with a one-way hash for extra security (preferably Bcrypt with random salt or similar). People in Need will be responsible for organising the required certificates specified by the developers.
	+ Strong password based login.
	+ Strict password controls.
	+ Users must sign SOP agreement.
* Role based Access Control (RBAC) Limited user access must be the default for most standard users. Only small number of ‘admin’ accounts.
* Access delineated by country – project view.
* User activity and interactions should be logged.

### Beneficiary Information Security

* Unique Beneficiary ID
* OAuth Security for each Unique ID

### User Management

Customisable user access to allow for highest possible level of user data security and separation of processes and responsibilities (Described below).

* Global Administrator
* Regional Manager
* Mission Manager (or Country Director)
* Project Manager
* Project Officer
* Field Officer
* Read Only

### RBAC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Region | Access | Permissions | Capability |
| Global Admin | Global  | Admin | Full Administrative Rights | RW |
| Regional Manager | Multiple Countries | Manager |  |  |
|  |  |  | Project Management | R |
|  |  |  | Reporting | R |
|  |  |  | Beneficiary Management | R |
|  |  |  | Authorise Payments | Y |
| Country Manager | Country | Manager |  |  |
|  |  |  | User Management | RW |
|  |  |  | Project Management | RW |
|  |  |  | Reporting | RW |
|  |  |  | Beneficiary Management | RW |
|  |  |  | Authorise Payments | Y |
| Project Manager | Project | Manager |  |  |
|  |  |  | Project Management | RW |
|  |  |  | Reporting | RW |
|  |  |  | Beneficiary Management | RW |
|  |  |  | Authorise Payments | Y |
| Project Officer | Project | Officer |  |  |
|  |  |  | Project Management | R |
|  |  |  | Reporting | RW |
|  |  |  | Beneficiary Management | RW |
| Field Officer | Project |  |  |  |
|  |  |  | Project Management | R |
|  |  |  | Reporting | R |
|  |  |  | Beneficiary Management | RW |
| Read Only | Country | RO |  |  |
|  |  |  | Reporting | R |
|  |  |  | Beneficiary Information | R |

### Performance

* Site must be functional in low bandwidth locations, where often only GPRS mobile internet may be available.
* Low average site load time where 3G and above connections are available.
	+ Reduced HTTPS requests where possible.
	+ Optimizing file size, reducing compression
	+ Leveraging browser cache and cookies where possible

### Documentation

* A concise commented source code should be maintained in an online code repository for the duration of the project in order to track progress and monitor features and issues.
* A basic FAQ should be provided to guide users through the use of the system, once the system has been fully developed.
* A wiki should be written to guide future development of the system, once the system has been fully developed.

## Hardware Compatibility Requirements

#### (Potential Options for Trial Period)

### Laptop Device

Lenovo ThinkPad T470

* Long battery life, solid build.

### Scanner (ID or other)

3M QS1000 Full Page Reader

* Reasonable pricing with acceptable functionality.

### Mobile Device

Samsung Galaxy S8 Active

* Long battery life, shatter, water and dust resistant.

### Out of Scope

Business Model – These terms of reference do not cover the business model that will be used to support the software ongoing.

Funding – These specifications do not outlay how the money to fund this project will be sourced.

Standard Operating Procedures - The specifications do not specify the set of procedures that will need to define the procedures for staff, management and beneficiaries, while using the new system.

Training – These specifications do not detail the training required for staff and managers of the system.

Cloud Hosting Environment – PIN has an established AWS environment, which can be used for the production application (and staging if required).

Certificates – PIN will organise and pay for any security certificates when required.

### Future Features

Out of scope, but to be kept in mind.

* Local hardware installation capacity
* Other distribution modalities, including in-kind distributions, vouchers, hard currency, etc.
* Mobile application to provide offline functionality.
* More API functionality, including support for larger number of 3rd party applications and connections, like mobile data collection tools and other mobile banking services.
* More comprehensive identification checks, including support for bar code reading, smartcards, biometric authentication
* More detailed and visually representative reporting functionality, including GIS mapping
* A complaint response mechanism
* Vendor support and connectivity
* Stock management features
* Cash for Work support
* Financial System Integration (Navision, Agresso, SAP, etc)
* SMS/IVR integration for beneficiary communication
* Market monitoring tools
* Inter-Agency Data Sharing functionality.
* Help Button with Chat bot

## Deliverables

Please provide in the description field below, how you will incorporate each deliverable into the end product.

|  |  |
| --- | --- |
| Outputs | Description |
| Mobile Compatible Web Application* Email based username and password login
* Intuitive graphic user interface
* Side menu Bar
* Front page summary data
 |  |
| Encrypted, cloud hosted database* Beneficiary Information
* Projects
* Distributions (or Transfers), including cash transactions
* Registration functionality for
* Individuals & Households
 |  |
| Project Management Functionality* Create, edit and delete projects.
* Total Project Allocated Funds for Distribution.
* Individuals or households are enrolled in projects defined in the project setup
* A summary page containing basic tabular data
* Filter and search functionality
 |  |
| Distribution Function* Beneficiary Information
* Projects
* Automated decision tree
* Vulnerability definition and assessment.
* Distribution lists
* Allow for manual ‘double checks’ of beneficiary data and then the addition or removal of individual households from each ‘distribution’.
 |  |
| Cash Transfer Application* Automated Cash Transfer functionality via the WING API to send payments direct to user accounts
* A transaction summary and report function
 |  |
| Administrative Settings* Extensive user management, including password control.
* Beneficiary database field management
* Financial Service Provider settings
* Donor table
* Sectors table
* Server Settings
 |  |
| Reports & Exporting* Beneficiary Information
* Projects
* Distributions (or Transfers), including cash transactions
* Registration functionality for
* Individuals and/or Households
 |  |
| Documentation* Basic User Guide
* Online code repository, to be updated
 |  |

## Selection Criteria

Each vendor can receive up to 10 points. The relative importance of each criterion is indicated by approximate weight by points.

## Development Timeline

Developer to provide.

## Budget Estimate

Developer to provide.

|  |  |
| --- | --- |
| Evaluation Criteria | Points |
| Technical Approach* Understanding of the subject matter and issues.
* Comprehensiveness of proposed approach and technical soundness of technical approach and reasoning why the technical approach was chosen.
* Implementation plan and proposed timeline are realistic and include all proposed elements of the activity.
 | **2 points** |
| Proposed Personnel* Personnel key proposed for this work must have a provable track record in Database Design, Application Design and Web-Portal Functionality in Cambodia.
* Personnel should also be familiar with and, ideally, have experience working with the international open data community.
* Proposed personnel who will be responsible for the implementation have the relevant skills and past experience to successfully complete the assignment, both in terms of conducting the electronic data collection, analysis and training and also in designing a comprehensive common data structure that meets of needs of stakeholders.
* Personnel should have excellent written and oral communication skills in English.
 | **1 points** |
| Past Experience* Experience and capacity of the organization(s) and adequacy of resources.
* The organization shows past experience and credible reputation in the area of Database Design, Android App Design and Web-Portal Functionality in Developing Countries (networks, publications, presentations, etc. should be referenced).
 | **2 points** |
| Budget* A detailed, realistic, deliverable based budget is required.
* Proposed budget represents the best value, and all costs included are allocable to this activity, reasonable and allowable under PIN and ECHO internal procurement rules.
 | **3 points** |
| Proposed Timeline* The timeline is realistic and include all proposed elements of the activity.
* Includes the project management methodology being used.
 | **2 points** |
| TOTAL | **10 points** |

# Annex A

### Beneficiary Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Format | Required | Source | Restrictions |
| Household Head ID | Select One | Y | - | - |
| Given Name | Text | Y | - | - |
| Family Name | Text | Y | - | - |
| Gender | Select One | Y | M/F | - |
| DOB | Number | N | - | - |
| Mobile phone number | Phone | Y | - | - |

### Beneficiary Historical Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Format | Required | Source | Restrictions |
| Occupation | Occupation | N | National Registration API |  |
| Pregnant | Select One | N | - | Only F |
| Lactating | Select One | N | - | Only F |
| Disability | Select One | N | - | - |
| National ID | Photo | N | - | - |
| Location | Select One | Y | From YAML File | - |
| IDPoor | Number | Y | IDPoor API via The Hub | - |
| Address | Alphanumeric | N | - | - |
| Income source | Select Multiple | N | - | - |
| Number of dependents | Number | N | IDPoor API via The Hub | - |
| HH children with nutritional issue (SAM/MAM) | Number | N | - | - |
| HH w children < 5 Yrs old | Y/N | N | - | - |
| Solo Parent | Y/N | N | - | - |
| Female Solo Parent | Y/N | N | - | Only F |
| Notes | Text | - | - | - |

Import Process



On import:

* Match fields
* Check for duplicates

## Projects

Project Data Fields

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Type | Required? | Source |
| Name | Text | Y | - |
| Project ID | AlphaNumeric | Y | - |
| Donor | Select One | Y | (Donors dB) |
| Start Date | Date | Y | - |
| End Date | Date | Y | - |
| Number of Households | Number | Y | - |
| Value | Number (USD) | Y | - |
| Sector | Select Multiple | N | (Sectors dB) |
| Notes | Text | N | - |

## Distribution Data Fields

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Type | Required? | Source |
| Name | Text | Y | - |
| Location | Select One | Y | Locations Table |
| Project ID | Select One | Y | Projects dB |
| Selection Criteria1 | Choose Value | Y | Individual or Household dB |
| Selection Criteria2 (Can add multiple if required) | Choose Value | N | Individual or Household dB |

## Cash Transaction Data Fields

Transaction Fields

|  |
| --- |
| Transaction ID |
| Household ID |
| Distribution ID |
| Money Sent - Wing to Wing |
| Agent Location |
| Money Received |

## Mission Configuration Tables

## Locations

To be derived from universal YAML file.

<https://github.com/hexorx/countries/tree/master/lib/countries/data/subdivisions>

## Financial Services Configuration

|  |
| --- |
| Field |
| Service |
| Account ID |
| Balance |
| Users |

## Global Properties

## Donors

|  |
| --- |
| Fields |
| Donor ID |
| Full Name |
| Short Name |
| Date Added |
| Notes |

## Sectors

|  |
| --- |
| Fields |
| Food Security |
| Health |
| Shelter |
| WASH |
| Multi-Sector |

## Database List, Search and Filter Structure Example



### Project Procedure

Select Project

Location

Define Criteria

Date – Define Quantities of Cash/Goods – Select Payment/Distribution Method – Select Users by Location – The Filter by Selection Criteria – Have they received assistance before?

Re-validation Process

* Either Randomised phone checks or list posted at Community Centre – See SOP

Generate Final List

Open Distribution

Confirm payment? Y/N

Confirm Distribution Completion.