

**United Nations Development Programme**

**May 2018**

**Development and Implementation of an Aid Information Management System (AIMS) for Somalia**

**Functional and Technical Specifications**

Contents

[Introduction 4](#_Toc514173659)

[1.1 Timings and Deliverables 5](#_Toc514173660)

[1.2 Management and responsibility 6](#_Toc514173661)

[1.3 Sustainability and training 6](#_Toc514173662)

[1.4 Development requirements 6](#_Toc514173663)

[1.6 Development Sprints 7](#_Toc514173664)

[1.7 Deployment and launch process 8](#_Toc514173665)

[1.8 Support, feedback and testing process 8](#_Toc514173666)

[1.9 Approval and reporting 8](#_Toc514173667)

[1.10 Travel and Meetings 9](#_Toc514173668)

[1.10 Functional specifications 9](#_Toc514173669)

[Control 10](#_Toc514173670)

[2.1 Open source 10](#_Toc514173671)

[2.2 Hosting 10](#_Toc514173672)

[2.3 Deployment 11](#_Toc514173673)

[2.4 Backup and restoration 12](#_Toc514173674)

[2.5 Security 12](#_Toc514173675)

[Data input and storage 14](#_Toc514173676)

[3.1 Data input permissions 14](#_Toc514173677)

[3.2 Language translation 14](#_Toc514173678)

[3.3 Mandatory fields 14](#_Toc514173679)

[3.4 Validation 14](#_Toc514173680)

[3.5 Saving 14](#_Toc514173681)

[3.6 Mapping 14](#_Toc514173682)

[3.7 Updating codelists/schemas 15](#_Toc514173683)

[3.8 Geocoding 15](#_Toc514173684)

[3.9 Financial Data 15](#_Toc514173685)

[3.10 Exchange rates 16](#_Toc514173686)

[3.11 Double counting 16](#_Toc514173687)

[3.12 Merging projects 16](#_Toc514173688)

[3.13 Data entry interface 17](#_Toc514173689)

[3.14 Core fields 17](#_Toc514173690)

[3.15 Custom fields 19](#_Toc514173691)

[3.16 Envelope data 19](#_Toc514173692)

[3.17 Financial Year (FY) rollover 20](#_Toc514173693)

[3.18 IATI (International Aid Transparency Initiative) 20](#_Toc514173694)

[3.19 Access data from IATI 20](#_Toc514173695)

[3.20 Dynamic IATI data 22](#_Toc514173696)

[3.21 Storing data in IATI format 22](#_Toc514173697)

[3.22 Publish data to IATI registry 22](#_Toc514173698)

[3.23 Further IATI links (as of late 2016) 22](#_Toc514173699)

[Data output 24](#_Toc514173700)

[4.1 Premade reports 24](#_Toc514173701)

[4.2 Likely initial reports 24](#_Toc514173702)

[4.3 Report plugins and dependencies 25](#_Toc514173703)

[4.4 Data export 25](#_Toc514173704)

[4.5 Subscriptions 25](#_Toc514173705)

[Users 26](#_Toc514173706)

[5.1 Public access 26](#_Toc514173707)

[5.2 Subscribers 26](#_Toc514173708)

[5.3 User registration 26](#_Toc514173709)

[5.4 Organisations 26](#_Toc514173710)

[5.5 Permissions 26](#_Toc514173711)

[5.6 Notification 27](#_Toc514173712)

[5.7 Logging 27](#_Toc514173713)

[Design 28](#_Toc514173714)

[6.1 Modular nature 28](#_Toc514173715)

[6.2 Multi language 28](#_Toc514173716)

[6.3 Useability 28](#_Toc514173717)

[6.4 Scale and concurrent users 29](#_Toc514173718)

[6.5 User help and guidance 29](#_Toc514173719)

[6.6 Contact page 29](#_Toc514173720)

[Site structure 30](#_Toc514173721)

[7.1 Core pages 30](#_Toc514173722)

[7.1.1 Main template 30](#_Toc514173723)

[7.1.2 Login 30](#_Toc514173724)

[7.1.3 Homepage 30](#_Toc514173725)

[7.1.4 Contact page 30](#_Toc514173726)

[7.2 Reports menu 30](#_Toc514173727)

[7.2.1 List of projects 31](#_Toc514173728)

[7.2.2 Example report page 31](#_Toc514173729)

[7.3 Data entry menu 31](#_Toc514173730)

[7.3.1 Filter and match pre-entry page 31](#_Toc514173731)

[7.3.2 Project data entry page 32](#_Toc514173732)

[7.4 User pages 33](#_Toc514173733)

[7.4.1 Registration page 33](#_Toc514173734)

[7.4.2 Subscription page 33](#_Toc514173735)

[7.4.3 Password reset page 33](#_Toc514173736)

[7.4.4 Notifications and approval page 33](#_Toc514173737)

[7.4.5 Envelope data entry page 33](#_Toc514173738)

[7.5 Management pages 33](#_Toc514173739)

[7.5.1 Reports management page 33](#_Toc514173740)

[7.5.2 Mailout page 33](#_Toc514173741)

[7.5.3 Email settings management page 34](#_Toc514173742)

[7.5.4 IATI settings management page 34](#_Toc514173743)

[7.5.5 Storage management page 34](#_Toc514173744)

[7.5.6 Custom fields management page 34](#_Toc514173745)

[7.5.7 Organisation management page 34](#_Toc514173746)

[7.5.8 Exchange rates management page 34](#_Toc514173747)

[7.5.9 Codelist/Mapping rates management page 34](#_Toc514173748)

# Introduction

UNDP is supporting the Federal Government of Somalia to develop and implement a new Aid Information Management System (AIMS). The successful applicant will, under an UNDP contract, develop and support the implementation of the new AIMS. The AIMS will be managed by the Ministry of Planning, Investment and Economic Development of the Federal Government of Somalia.

Somalia receives large volumes of aid from foreign donors. It is useful to track these flows so that the Government of Somalia, donors and civil society can coordinate and share information. Data users need to know how much money flows, to which regions, for what purpose, from who, implemented by who, and between what dates.

This tracking is currently being done by exchanging and compiling data in MS Excel sheets but a review of the needs and processes in Somalia and experience from other countries shows that while core processes and stakeholders will remain the same, a web app would make the collection simpler and enable more people to access the data.

This tracking is done by the Federal Government of Somalia (FGS) with data provided by the various donors. Involved stakeholders include:

* MoPIED - the Ministry of Planning, Investment and Economic Development, (especially the units for Project Management, M&E, Statistics and IT), ACU - the Aid Coordination Unit of Office of the Prime Minister, the Ministry of Finance, and the Ministry of Foreign Affairs and International Cooperation
* The Federal Member States, their respective Ministries, and the FGS Line Ministries, as well as the Region of Banadir
* Multilateral and Bilateral Donors, Pooled Funds, Pillar and Sector Working Groups, the New Deal Architecture, covering development, humanitarian and peacebuilding support, as well as the implementing organisations
* Civil Society including NGOs/CSOs, researchers and academia, and the public.

An AIMS (Aid Information Management System) is a tool which collects data on aid projects, both the financial amounts, and associated data such as descriptions of the purpose, the location, and the organisations giving and receiving the funds. This information is typically provided by the organisation funding the project (the donor). By collecting all this information in one place, and making it available, an AIMS allows different stakeholders to track what funds are flowing where, from who/to who, at what time, and for what purpose. Without an AIMS, this process of compiling information so that a full picture can be seen in one place is typically done by circulating and compiling MS Excel/Word templates. An AIMS is therefore a comparatively simple system. A database holds data entered by users through a form and presents it on request. The main goals of an AIMS are therefore to:

* Simplify data entry and data management
* Allow more complex data relationships e.g multi-year data, or multi-funder projects
* Allow greater public access to data

Somalia has previously had an expensive commercial AIMS system which failed. Despite not being very technically complex, if not appropriately designed and implemented, an AIMS will not work. Many of these issues are documented in the Somali AIMS Review Report carried out in 2015 <http://somaliaims.github.io/files/Somali%20AIMS%20Review%20Report%20(19Aug).pdf> which, for example, identifies that:

* Tailoring - the AIMS must collect the data needed and available in Somalia and store it in the data formats that are specific to Somalia.
* Sustainability – the AIMS must be designed and developed to be Somali owned, operated, adjusted and hosted.
* Ease of use – accessing data in the AIMS must be possible for users who have not received training.

To develop the AIMS, it is important to understand how it will be managed. The AIMS responds to the needs of a wide group of stakeholders, including multiple Ministries and departments of the Federal Government of Somalia, the Federal Member States’ governments, donor agencies, implementing organisations, CSOs and the public. Of these, the donor and implementing agencies involved in each project will provide the data, and the FGS, Ministry of Planning, Investment and Economic Development will manage the AIMS system. This diverse set of users is necessary to ensure maximum benefit, best data quality and sustainability of the AIMS. This is not unique to Somalia. However, unlike many other AIMS systems, the management of the AIMS in Somalia will reflect the diverse group of stakeholders and therefore be responsive to all their needs.

There will be a management committee which should represent most stakeholders of the AIMS, the members of which will have the greatest level of permissions (management users). Between these management users, they will administer the AIMS, e.g. keeping the schemas up to date, approving new organisations, etc. This is a deliberate move away from a model where a single government unit has one or two staff who are ‘admin users’. In other countries, this has caused the AIMS to collapse as it becomes responsive only to the needs of one stakeholder and therefore cooperation from other stakeholders falls away. There is also a significant risk that the loss of one or two key staff eliminates most of the administrative capacity and institutional knowledge.

It is also important that all AIMS administration is therefore performed through the single AIMS interface. No separate ‘admin’ interface should be available and therefore no data or setting should be modified outside of the normal data workflow.

Users will also perform many tasks previously requiring an administrative response themselves e.g. resetting passwords. The task of authorising a new user will be able to be performed by other users of that organisation, and the wider management group. This will reduce the reliance of the AIMS on a single admin user.

The rest of this document lays out specifications and a development approach for an online AIMS for Somalia, designed to respond to these specific requirements and therefore to overcome those problems.

## 1.1 Timings and Deliverables

While the respondents to this Request for Proposals are encouraged to develop a more detailed time-planning and are free to propose a faster track, the initially estimated time planning is as follows:

* Month 1: Setup of open source software project, review of requirements, and consultation with stakeholders, gathering of open source components and data on e.g. MoPIED hosting setup.
* Month 2 to 6: The development and testing phase. Development will take place through a series of sprints, each with testing and adjustment phases with the stakeholders;
* Months 7 and 8: At the end of the primary development phase, all features will be complete, deployments should be handled by MoPIED IT unit, and the system is expected to go live for selected users. These users should receive training in using the system, and will give feedback which will then be implemented;
* Months 8 and 9: Entry of actual data from previous years, and further testing, feedback, training and refinement of the system by a limited group;
* Month 10: The system will go publicly live;
* Months 11 and 12: Final period of testing, feedback, training and handover of the system.

## 1.2 Management and responsibility

The entire process will be managed by the International Developer who will carry out the work and will be responsible for all aspects of the contract to UNDP who is the signoff client. UNDP, however, will closely cooperate with the functional client, the Ministry of Planning, Investment and Economic Development. The successful bidder will be supported by the UNDP AIMS Consultant and is expected to work in close collaboration with the MoPIED IT unit to ensure their contribution to the code, and ability to manage the system once deployed.

## 1.3 Sustainability and training

The sustainability of the AIMS depends on the FGS being able to easily modify and redeploy the AIMS, both during and after development. A key part of this TOR is therefore to provide the training and engagement possibilities so that MoPIED staff must be involved in the AIMS development, not just in deployment, training and client relations. The Internatonal Developer will be responsible for:

* Establishing the development capacity of the MoPIED IT unit, and what skill gaps need filling for them to be able to contribute to the AIMS development, and sustainably host and modiy the AIMS in the future, sharing a report with UNDP.
* Developing a training programme comprised of self-study resources for the MoPIED IT unit to fill these gaps.
* Overseeing and coaching MoPIED IT unit code contributions to the AIMS.
* Ensuring that MoPIED IT unit staff full understand the workings of all components of the AIMS code.
* Assessing the status of MoPIED IT unit development, and hosting capacity and sharing a brief report at the mid and end points of the project.

## 1.4 Development requirements

* The International Developer will be responsible for supplying their own staging, testing, build servers and similar, during the initial development process.
* The FGS will be responsible for sourcing either a server or renting a cloud-based virtual server, a domain, and connectivity for the initial, all testing deployments, and the final deployment.
* The resulting AIMS code will be licensed under the Affero General Public License (AGPL, see [https://www.gnu.org/licenses/agpl-3.0.en.html).](https://www.gnu.org/licenses/agpl-3.0.en.html)
* The AIMS should be developed using .NET/ASP.NET/dotnetcore/C# as this is the language /framework in which the MoPIED IT unit have the most experience.
* All tools, languages and components required to develop, deploy and run the resulting AIMS must be available under similar copyleft licenses. The FGS, UNDP or any other interested party must be able to fork, adjust and run the code without further restriction.
* The AIMS code will be managed on a public Github repository of the user ‘somaliaims’ (<https://github.com/somaliaims/>) to which the winning bidder will be added.
* All code must be developed on the Somaliaims public Github repository from the contract signing onwards. Use of a private repository with periodic releases is not acceptable.
* All code must be published in commented form to the Somaliaims Github repository before a deliverable will be regarded as completed.
* Contributions made to the code by other developers (e.g. the MoPIED IT unit, or the UNDP AIMS consultant) should be strongly considered for inclusion. The developers will have control of the repository necessary to accept or reject these contributions during the contract period; however, reasons for any rejection will be required.
* All code will be extensively commented in English.
* Developer documentation should be provided for each major portion of code.
* Complete, tested and fully functioning deployment instructions must be provided.
* The system should be created using industry-standard development tools and programming languages available without cost under similarly permissive licenses.
* All components needed to run the AIMS must be ‘cross-platform’ i.e. able to run under a minimum of both Windows and Linux server operating systems, either natively, or via a virtualisation/containerisation solution, e.g. Docker which must also be available under similarly non-restrictive licenses.
* The development should endeavour to reuse existing AIMS/IATI components where possible to place the AIMS within an already tested and functioning ecosystem.

## 1.6 Development Sprints

The International Developer is required to develop the AIMS through an iterative process of developing a feature/component, demonstrating the feature/component to the stakeholders, receiving feedback, and tweaking the feature/component, etc. Previous experience developing AIMS suggests that a series of two-week sprints gives the best results in cases where neither the clients nor the developers have developed an AIMS before, and therefore both require the opportunity to provide and receive regular feedback so that decisions on e.g. the data structure, and its impact on e.g. the types of reports that can be produced, can be given at the earliest opportunity. Delivering a finished product without evidence of systematic user acceptance will not be deemed acceptable. The feedback will be provided by the stakeholders group, accessible via the google group email: ‘[somali-aims-stakeholders@googlegroups.com](mailto:somali-aims-stakeholders@googlegroups.com)’, and the feedback process will be supported by the UNDP AIMS consultant.

A suggested set/order of sprints is given below, focusing on delivering the core functionality first, before extending each aspect to have the full features. Several sprints may take place in parallel:

1. Data structure and basic frontend with storage of Somali specific fields in IATI format and test data, plus temporary data output page/table
2. Backup and (re)deployment system for MoPIED IT unit use i.e. including config scripts, instructions, backup/restore system.
3. IATI format data, locating and parsing and local storage (Datastore, and extra files) including humanitarian data published by UN-OCHA
4. Available projects (local and IATI) display, sorting and matching, merging and writing
5. Manual data entry, and mapping of data from other coding systems
6. Dynamic IATI data entry and introduction of edge-case test data
7. Initial reports including charts – reading from internal API in IATI format
8. User registration, link to organisations, permissions and requests to join projects
9. Finalisation of reporting system including upload of externally developed reports and approx. 7 basic pre-made reports, project profiles, export to excel, and pdf, embedded links, etc.
10. User/organisation management and other management pages, notifications and emails etc
11. Final features e.g. custom fields, management of financial year rollover including additional data entry assistance pages and similar

## 1.7 Deployment and launch process

* During the development process, the MoPIED IT Unit should be trained to learn how to backup and redeploy the AIMS.
* Where possible, all user testing should be done on releases deployed to the MoPIED server to ensure that MoPIED IT unit can sustainably deploy the AIMS.
* Before launch, data from the 2015-2018 Interim Aid Mappings should be entered into the system by the developers so that the system is launched with live data. This data will be supplied by the FGS/UNDP in a series of Excel sheets.
* The International Developer will support the AIMS through the launch process.

## 1.8 Support, feedback and testing process

* During the development, testing and launch phases, the International Developer should reply within 1 working day to a support request and answer a phone call/whatsapp/skype communication immediately.
* After the end of development and the limited launch the developers will need to collect written/verbal feedback from all the main user groups including the FGS, donors (including some based in Nairobi), and a selection of other users and then implement a round of adjustments, testing and redeployment.
* After the public launch, a second round of feedback, adjustment, testing and redeployment will be needed.
* Bugfixing and the support of the MoPIED IT Unit will be ongoing throughout the entire contract.

## 1.9 Approval and reporting

The Proposer is expected to send brief updates to the stakeholders email group on a weekly basis, identifying progress during the week, the status and an updated schedule for delivery, and identifying where input is required from the stakeholders.

The development process, meeting of specifications, and achievement of deliverables will be overseen by UNDP Somalia in cooperation with the International AIMS consultant. UNDP Somalia will give final acceptance once all features have been satisfactorily completed, once the FGS have successfully deployed and setup a working AIMS instance without outside assistance, and once all source code has been published to Github.

The contract will end after the completion of the second round of feedback and adjustment. At this stage, the lead company will pass all passwords required to operate all AIMS software/image repositories to UNDP. The FGS will remain responsible for their own installation, data, and contracting any further support they require. The FGS and any other potential user countries, will remain able to download copies of the latest (and previous) AIMS images from the software and image repositories.

## 1.10 Travel and Meetings

The International Developer must physically attend the MoPIED building in Mogadishu multiple times during the contract to:

* Meet the MoPIED IT unit in person during the initiation of the project to discuss how they plan to work together, and to assess the IT environment (servers, connectivity etc) and MoPIED IT unit level of capacity
* During the contract, to carry out training for the MoPIED IT unit
* Once every two months to provide updates, demonstrate the latest developments and to gather feedback on the results of the sprints in person.

At other times, the International Developer may meet stakeholders within Mogadishu Airport (MIA):

* The developers must meet the rest of the FGS and UNDP during the initiation of the project.
* The developers must meet at least 5 of the donor (data input) stakeholders for at least 2 hours, at least twice, during the development stage of the project and at least one of these meetings must be in Nairobi (the base for many donor data input staff).

## 1.10 Functional specifications

The remainder of this document sets out the functional, and some technical specifications for the AIMS developed during the review of the previous AIMS and aid data processes in Somalia. The International Developer will need to take decisions on how best to achieve these requirements in consultation with the users and based on the outcomes and feasibility of different approaches and tools available. It is not expected for the functional requirements to change significantly. However, should adjustments to the functional specification be required during the development process, the developers agree to incorporate these adjustments at no additional cost, provided that the overall workload does not increase.

# Control

The FGS needs to have full control and ownership over the functioning of the AIMS. With the previous system, they were not able to independently host it, back it up, or redeploy it, nor to independently hire developers to fix bugs or add features. The software was owned and controlled by the supplier whose business model meant that regular adjustments to the specific needs of Somalia were not cost effective or possible.

## 2.1 Open source

The new AIMS will be released under the Affero General Public License (AGPL, see <https://www.gnu.org/licenses/agpl-3.0.en.html>).

This means that all code used in developing the AIMS must be available for license under the AGPL. There are a range of code pieces relevant to an AIMS that are available to use under various permissive licenses and these are identified in the specifications below.

Releasing the code under the AGPL license means that it will be available for other countries to use. They will be able to download the AIMS code, deploy it, and by entering their own data in the initial config file, run their own AIMS. It is expected that as other countries use the AIMS, they may develop and contribute additional features to the software that Somalia can also therefore benefit from.

The aim is to develop the AIMS as a project, where the current contractor provides the initial development, but which in future may be built on and extended by a wider community.

## 2.2 Hosting

The Government of Somalia wish to host the AIMS themselves. The options for hosting the AIMS therefore need to be designed according to the skills and capacities already available. The hosting will be managed by the Ministry of Planning, Investment and Economic Development (MoPIED) IT unit which has the following skills and capacities:

* Five staff including a Head of Division, IT advisors, Web editors and IT officers.
* The IT Unit currently manage the MoPIED database management system, ministry website & social media, Electronic Document Management system, National Data Archive system, Somali financial information management system, and Employee Information Management system.
* The IT Unit staff all have a Bachelor’s degree in IT, some have Master’s degrees. Several also have certificates for Cisco and Oracle services.
* The IT Unit is familiar with basic remote server management

The hosting requirements are therefore:

* The AIMS must run on technology/servers that is already available to the FGS
* The AIMS installation must be possible on both remote and local destinations, dedicated and shared servers
* The AIMS code must be independent of server type and host operating system
* The hosting solution must be possible without requiring the FGS to pay license fees.

## 2.3 Deployment

The Somali AIMS Review Report identified that many AIMS fail because their data structure/features cannot easily be updated to keep with changes. In the case of Somalia, the previous AIMS data structure included many fields that had fallen out of use e.g. old National Priority Sectors. These increased the data entry workload for no benefit, and the lack of the most recent National Plan Pillars meant that the data could not be output using a relevant breakdown. Therefore, the ability to easily redeploy the AIMS with an updated version of the software (whilst keeping the existing data and configuration) is essential to being able to keep it valid. This leads to the following functional requirements for deployment and backup:

* Release of new versions of the AIMS available for download from the/a repository
* Minimal installation e.g. run one install script (which also sets up the initial management user), and no requirement to install and configure multiple dependencies, separately configure certificates, updates etc.
* All country specific data provided in a single configuration file for upload on deployment (as other countries may use the same AIMS code) to be imported after deployment or, upon initial view of a new installation, a form requesting the initial user to enter all data needed for the configuration file e.g. initial management user email, country name, storage locations, email service etc.
* Import of backed-up data from within the AIMS – not via external SQL scripts, etc. Pulling from connected storage locations where available.

The proposed approach is therefore:

* Developers provide instructions and training to install the relevant hosting setup e.g. Docker/VMWare to MoPIED
* MoPIED IT install Docker on existing server infrastructure that they are familiar with
* Developers make Docker images/containers available for download over the internet (noting that the internet connection in Mogadishu may be slow and more prone to disconnection)
* Developers provide training, instructions and installation scripts to MoPIED IT unit to download and deploy AIMS code and all necessary components
* On initial connection to the server with a new instance, the AIMS should then present a form asking for the configurations needed e.g. title/domain, admin email, etc. Alternatively, the backed-up configuration file can be uploaded, or the AIMS pointed to a storage space (could be a local folder, network folder, USB stick, FTP or other internet folder, or a cloud storage server e.g. Dropbox) holding the backup files to restore. New setting required for new features will still require user input
* There should be a feature within the AIMS to alert AIMS managers to availability of updated version on the repository

Examples of items to be included in the (text editable) configuration file are:

* Domain
* Title
* Backup storage location (type, address, username, password)
* Plugins in use (will download from repo)
* Repository to check for updates
* External data sources
* Email server/service to use for mailouts including any logins
* Initial manager email address
* Custom field details
* Users and associated fields including passwords, emails, organisations, etc.

## 2.4 Backup and restoration

It is important that the data in the AIMS can be backed up for safekeeping off-site. A data backup system should also provide the ability to roll back the data to previous point in time by importing a past backup. The backup system should facilitate easy re-deployment and upgrading through the ability to easily backup the data and restore it. Data backup:

* This should be a non-technical process with the AIMS displaying available backups in the storage location for the user to select and import, alternatively allowing upload of an offline backup file
* For one-off backups, a button in the AIMS should save a backup which is then available for download
* The AIMS should do automated backups on a schedule, and allow the users to limit the storage size used, and which deletes old backups appropriately.
* There should be a choice (including multiple) of storage locations to include local folders, network storage, emailed backups, and cloud providers e.g. Dropbox
* The AIMS should also backup the configuration file where necessary
* The AIMS should implement an appropriate naming process
* The backups module should obtain data from the IATI xml format AIMS data access API
* The backups should be in IATI XML format
* On restoring backed-up data where there is existing newer data, the AIMS log should display which rows/fields have changed between backups
* Whenever restoring a backup, the AIMS should take a backup of the pre-restore data in case the user wishes to revert the backup
* The backup should be written using the IATI xml format AIMS data API

## 2.5 Security

The security of the AIMS is important and concerns both security of the host container/VM, of the application, and of the data. The largest risks are expected to be attacks aiming to misuse of the host. The MOPIED IT unit can be expected to maintain the security of the host upon which they install the container/virtualisation software. However, they cannot be expected to maintain the application server, particularly if it uses technologies that they are not familiar with. The data security is probably low risk, especially as it is also being made available publicly. The following should apply:

* The system must take steps to implement multiple levels of security: network level, database level, application level, API level, record level, and field level.
* The AIMS should take recognised steps to prevent code-injection, cross-site-scripting, SQL injection and similar attacks and should use an established user authentication and authorisation modules and all passwords should be hashed. Inputs should be validated and sessions managed using established components. The application should run at the lowest possible level of privilege and all application administration should be done either within the application, or by the setup script. No administrative users, or other management features (e.g. admin consoles, or SQL table management interfaces) should be exposed.
* The container/VM holding the AIMS (as stored on the repository) should tracking and make use of regularly updated components/images. It should also be selected and customised to minimise attack vectors e.g. by only opening strictly necessary ports, only running necessary services, running a firewall, etc. By redeploying the application, the FGS should therefore receive the latest and most secure version.
* Where required, HTTPS should be implemented through a free CA e.g. ‘letsencrypt’ and the process of obtaining and installing a certificate automated during the install process and when updated certificates are required.
* Backups should not be encrypted.

# Data input and storage

Data input is at the heart of the AIMS. The specifications below are the result of several consultations covering the key fields, data availability and each of data entry and embody many compromises and trade-off’s e.g. reducing the number/complexity of fields to bring the time taken to enter a project to a minimum. The Somali AIMS has some specific needs that will make it vary from most other AIMS.

## 3.1 Data input permissions

Not all users should be able to edit project data. Only registered users can log in and add data on aid projects. Projects will be editable by any user associated with any organisation that is added to the project as either an implementer or a funder. Projects should not be editable by any management or administrative user accounts, who should instead contact the donor responsible for the project to request an edit.

## 3.2 Language translation

The AIMS should operate in English.

## 3.3 Mandatory fields

The mandatory fields required for a project to be saved should be minimised to include only those necessary for the functioning of the data structure e.g. this is likely to be limited to an organisation (and this can be auto-filled based on how the contributor is organised) and a project name.

## 3.4 Validation

The system must have methods of validating data entry. As well as standard approaches such as validating input characters, this should include checking the logic against other fields e.g. spend to date should not be greater than project value, start dates should not be after end dates. The validation should also use searchable dropdowns where possible from the embedded codelists, and also cover validation for e.g. email addresses, urls and similar as well as providing widgets for dates, and limits for character length, etc.

## 3.5 Saving

The quality of internet connections in Somalia can be poor. The AIMS requires users to fill out multiple 'tabs' with multiple fields. In the previous AIMS this caused data loss as connections broke before the user clicked a 'save' button. The new AIMS should not have a 'save/publish' button and therefore everything should be written to the database as soon as it is typed into the form. If the save is unsuccessful (e.g. because the connection dropped), then the AIMS should keep attempting to save again.

## 3.6 Mapping

In many cases, the coding scheme of data in the donor database that is the source of the data to be entered into the AIMS is not the same as is needed in Somalia. For example, many donors use the OECD CRS purpose-codes, or Humanitarian cluster names/codes to define sectors, whereas the scheme relevant for Somalia is the Somali Chart of Accounts i.e. the PSG (Peacebuilding & Statebuilding Goals) codes, or by the time of launch, NDP (National Development Plan) Pillars/sub-pillars. Manually translating from one scheme to another takes a lot of time (especially if the contributor is not familiar with the Somali schema) and is a barrier to data entry. The AIMS therefore needs to allow automatic mapping from one schema to another. The mapping between schemas can be added in advance (a mapping management page accessible to management users will be needed) and therefore, for a given field, the contributor can be offered different schemas (the destination scheme, plus all mapped schemas) for data entry. Not all mappings will be perfect, for example there will be some 1-n relationships, or cases where the user disagrees with the automatic mapping. In this case, the user must be able to manually select the option (perhaps filtered down to likely options where there are several choices) required from the destination schema. At all times, the entry selection, and the result of the mapping should be visible to the contributor.

These mappings and other codelists/schemas will need to be stored in the configuration file, not hardcoded into the AIMS. The IATI data format can store data in multiple schema for some fields, and for the Somali specific codelists these will need setting up as vocabularies, with appropriate publishing of the codelist.

## 3.7 Updating codelists/schemas

A frequent problem affecting AIMS is that the codelists are subject to variation. For example, a new Ministry may be created, or a new sub-national region made. The AIMS needs to be able to handle these cases. There needs to be a management page within the AIMS to manage the current codelists. This should be linked to/integrated with the screen for managing the mappings (as the codelists change, the mappings will need updating). When a codelist is changed, the owners of projects need to be notified and it should be highlighted to them which fields have changed (and which need to be re-entered if there is not an automatic 1-1 mapping) as a result of the codelist change.

## 3.8 Geocoding

Geodata (adding the locations of projects) will benefit from a different interface as entering multiple locations via a dropdown menu can be time consuming and lead to errors. Examples from other AIMS include:

* <https://github.com/devgateway/open-aid-geocoder>
* <https://github.com/markbrough/maedi-projects>

## 3.9 Financial Data

The Somali AIMS deliberately collects a very restricted set of financial data compared to most AIMS. This is based on the demand for financial data not requiring more detail, and financial data being particularly burdensome to enter e.g. if adding and converting each individual disbursement.

Each project has a ‘project value’ given in the project currency, encompassing all financial contributions to the project from all funders. When combined with the project start and end date, as well as the disbursements to date, this should be used by the AIMS to estimate and prefill (allowing user modification, which must still total 100%) disbursements for future years. The prefill should be based on equal expected disbursements of the remaining funds across all future years.

A major difficulty many donors have with entering financial data is that their own data systems use a different financial year. The AIMS needs to allow a donor to select their own financial year when entering data, and the AIMS then converts this to Somali financial years for the donor e.g. if the donor and Somali financial years are 6 months apart, to get Somali FY2016 actual disbursements, the AIMS could take 50% of donor 2015/16 data and 50% of donor 2016/17 data.

## 3.10 Exchange rates

In several places, the data entry form will require exchange rates to convert the currency of data entry into a common currency (USD/SOM) for reporting. The form should allow the contributor to enter an exchange rate (also showing the result to prevent errors). However, in many cases, the contributor is either not sure of the exchange rate, or is happy to be provided with an exchange rate for speed. In these cases, the AIMS system should provide an exchange rate. The AIMS should pull exchange rates, based on the currency pair/date from various online sources and present them to the user to select from. Possibilities include:

* <https://github.com/markbrough/exchangerates>
* https://openexchangerates.org/
* <https://data.oecd.org/conversion/exchange-rates.htm>
* <http://databank.worldbank.org/data/reports.aspx?source=2&series=PA.NUS.FCRF&country=#advancedDownloadOptions> and <https://datahelpdesk.worldbank.org/knowledgebase/articles/902049-data-catalog-api>

There will need to be a management page to manage the exchange rate sources (these should not be hardcoded but allow plugins enabling various sources to be added) and set a default. The exchange rate options should be developed as an API within the AIMS system as it will be called by several different AIMS components e.g. data entry, reporting, backup, etc.

In addition, the FGS also has official exchange rates as used in the FMIS (Financial Management Information System). The exchange rate management page should allow management users to enter the official FMIS exchange rates for data entry users to choose from alongside the other options.

## 3.11 Double counting

One of the key advantages of an AIMS over manually collated excel sheets (the common alternative) is to avoid double counting, where the same project (or different projects, but the same financial flows) are entered twice (normally by different users/organisations who are both participating on a project together). To minimise the likelihood of this occurring, when a user clicks ‘add a new project’, the initial screen should ask the user to fill out a few basic fields (for example, project name, funding organisations, implementing organisations, and sector). From these initial data points, the AIMS should be filtering (dynamically, realtime, with fuzzy logic on the project name, etc.) and show users the similar existing projects already in the system, to reduce the likelihood that a project is entered twice. If the list of similar projects does show a project already entered, the user may click to ‘edit’ the existing project. Alternatively, the user may click to ‘add a new project’ and then be transported to the manual data entry form, with the data already entered pre-filled. Below, the specifications will discuss filtering projects from IATI data as well as those found to suggest similar projects – and if one of those is selected to edit, the AIMS will import the data to make a local project.

## 3.12 Merging projects

When a double counted project is identified, the AIMS users need a way to merge projects. It is likely that they will want to keep some parts of each project, and discard others so this option must exist. If the user’s organisation is not a member of one or either of the projects, then they will first need to request to be added to one/both projects to perform the merge, so this should also be an option. Once added, they will then have sufficient permissions to perform the merge.

This may be best implemented as a button available on each project data entry page titled ‘merge with another project’ which would then present a (filterable) list of other projects before switching the user into the data import interface (discussed later) with the duplicate project as the data source. After completing the merge, the user should be required to delete the two source projects. It should not be possible to remove organisations during the merge of a project. Note (also discussed later) that as IATI is the data storage format in all cases, merging two projects, is just a special case of importing external data to fill out a project, with the data from one project to be merged pre-entered into the data entry form, and the other made available for import on a field by field basis.

When merging two or more projects, the interface should remove all duplicate data where possible e.g. if there are 3 source projects, and all have the same sector, only one option should be shown.

A second likely situation is where several projects have been entered (without double counting) but a decision is taken to combine them into a single project. The interface should allow more than one project to be combined at the same time, i.e. allow multiple external sources to be presented at once. One approach is seen here: <http://bd-iati.github.io/documentation/merging-updating-cofinanced-projects/>.

## 3.13 Data entry interface

* The specifications above suggest a data entry interface that also does the following:
* Presents the data entry fields in as simple a way as possible and following good design practice e.g. lining up and sizing entry boxes accordingly, selecting appropriate entry widgets
* Use grouping of similar fields where necessary to guide the user
* Minimise vertical scrolling where possible. No horizontal scrolling.
* Pre-fills any data passed by previous pages e.g. title, organisations, sector from the page which checks for duplicates, or data from an existing project if a merge operation is being performed
* Next to this, the interface should present potential data from the various sources:
* Merge candidate projects
* Data being pulled from the locally stored version of IATI
* For each field, allow the user either to manually enter data, or to insert the data from one of the internal/external sources
* Where the external source is IATI, to provide an option to have the field follow that data source (automatically, or notify – discussed further below)
* Where multiple sources are available, aggregate or sort the options e.g. if there are 5 source projects, but all have the same sector, only present each sector once. Or if there are multiple financial amounts, also offer to aggregate several amounts.
* There should also be an option to ‘take all’ e.g. a shortcut to accept all the external source data i.e. when importing an external project onto an otherwise blank form.
* Provide ‘help’ text where possible

## 3.14 Core fields

The core fields (those visible to contributors) are:

| Data need | Detail | Fields | Comment |
| --- | --- | --- | --- |
| Who | Who is providing funding?  Who is involved in implementation? | Funder(s)  Implementer(s) | Organisations drawn from list of organisations already in AIMS, or ‘add new organisation’. At least one organisation is mandatory, but can be drawn from organisation of the contributor user. Translation of organisation names to be provided by management users, or user when ‘adding a new organisation’. |
| What | What is the project name?  What is its objective?  Which sectors does it target? | Project title  Project objectives/purpose  Sector(s) i.e. PSG, or NDP Pillar | Title and objectives are translatable text fields. Sectors involves mapping between OECD DAC sectors, Humanitarian sectors and the Somali CoA codes (i.e. those for PSG or by the time of launch, NDP Pillar, the current Somali interpretation of Sector). Allow multiple sectors. Sectors translated by management users. |
| Where | Which projects are active in Somalia? Which projects target a specific state or region? | Location(s) (federal state or similarly sized entity)  Shares of project value by location(s) | Allow multiple, assist users e.g. by offering a button to enter equal splits of project value across locations, and allowing entering either percentages of values. |
| When | When will the project be implemented? | Start date  End date | Validation prevents end date before start date, etc. |
| How Much | What is the overall budget of the project? How much has been spent to date?  How much was spent last year?  How much will likely be spent next year? | Funder,  Funder amount,  Amount currency, Amount exchange rate to USD, Amount grant/loan status  Project total spend to end of last Somali FY  Amount/share of project total value expected to spend in each future Somali FY covered by project | See notes on financial data entry, and assistance provided by the AIMS i.e. Financial Years, different currencies. |
| Additional information | Where can I find out more information? | Document uploads or links  Project contact details | Project contact details involve name, organisation, email, phone number, location. Documents involve document title only. |

Management users should be able to provide alternative ‘visible names’ for these core fields via the custom fields page. For example, to rename ‘sector’ to ‘NDP pillar’.

Each project will necessarily have a unique ID (probably a hash, non-sequential) - not exposed anywhere publicly, except being visible in the URL when producing project profile reports e.g.: [http://somaliaims.so/reports/project-profile/28d6928ef62/pdf/](http://somaliaims.so/reports/project-profile/28d6928ef62). Project names should not be used for URLs as they are not unique.

## 3.15 Custom fields

As well as the core fields, it is likely that the Somali data users will need a few other fields to collect data for special purposes e.g. surveys. These fields will also operate on a ‘per project basis’. The AIMS managers will need an interface to add/remove/adjust these fields based on a predefined set of field types to choose from (checkbox, dropdown, text, etc.). Any custom field codelists should appear in the normal codelist management interface, and the fields should be available in the reports. These custom fields should appear on their own group in the data entry interface. These fields should be able to interact with the core fields. The number of custom fields should be limited to 5.

When adding a custom field it should be possible to set a condition (e.g. selected sector is ‘Education’) which controls whether the field appears. For custom fields to be able to import data from IATI, the management interface will need a field (ideally pulling from a dropdown of all elements) to allow management users to identify the specific IATI xml element of interest.

## 3.16 Envelope data

A key use of the AIMS is to predict what aid is coming in the future therefore the expected disbursements in the upcoming financial year (FY) for each project is a key field. However, aggregating the project level expected disbursement fields will not give the full answer. Donors have not always decided the projects for 100% of their support in advance, i.e. at the time of data collection therefore they have no project names and descriptions to enter those funds against. The AIMS therefore also needs to collect the ‘Envelope’ data. This is data that represents each donor’s expected spending per sector, per Somali FY. This will be an amount larger or equal to the combined total of the individual projects’ expected spending for that FY, with the difference representing unprogrammed amounts. Collecting envelope data will therefore require a separate data entry interface from the project level data (although with the ability to pull from the IATI data ‘recipient country budget’ data field in IATI Organisation files (select IATI source, select organisation, preset to Somalia, import data, then allocate across sectors). This interface should allow a user to identify the amount their organisation will provide, broken down by the Somali sectors, and translated into Somali FY (should allow entry in donor FY and suggest conversion – this implies envelope data can be entered for unlimited future years – but the interface should default to show the upcoming 3 years). Once entered, this data should be dynamically added to the project level data as ‘un-programmed funds’. When additional projects are then entered, the AIMS will need to adjust the amount of the ‘unprogrammed funds’ project so that the sum of the expected spending of the projects is equal to that of the envelope data. In this way, the envelope data will be available for use in all reports. Users should not be able to edit these special ‘unprogrammed funds’ projects through the data entry interfaces.

## 3.17 Financial Year (FY) rollover

The AIMS (like most budgetary processes) operates on an annual cycle. Each year, it is important to confirm the spending from the most recently completed FY, update the spending in the current FY, and collect data on the expected spending in the upcoming FY (plus potentially the two after that). This presents a moving target of three (or five) years. Whilst the project data entry will not be affected (start and end dates will naturally track this cycle, and be set any number of years into the future), it is useful to be able to set some defaults for the rest of the interface. These will affect the default year shown for reports, data entry tables, defaults for dropdowns and similar. There will need to be a management interface to control the default FY.

## 3.18 IATI (International Aid Transparency Initiative)

IATI is a global aid transparency standard and it makes information about aid spending easier to access, re-use and understand the underlying data using a unified open standard. You can find more about the IATI XML standard at: [www.iatistandard.org](http://www.iatistandard.org/)

The AIMS will make use of IATI formats and data for importing data, accessing dynamically updated data, storing data, and publishing data. See Codelists: 2.01 example: <http://iatistandard.org/201/codelists/downloads/clv1/codelist/> and <https://github.com/IATI/IATI-Codelists-NonEmbedded/tree/master/xml>

Datastore: <http://datastore.iatistandard.org/docs/> and the datastore API <http://datastore.iatistandard.org/docs/api/>

## 3.19 Access data from IATI

The IATI Standard is a common data format for machine readable data. As this format is used by many donors who register the location of their datafiles on the IATI registry (who in turn make it available through the IATI Datastore). IATI is therefore an important source of data for the AIMS. This has the potential to save considerable time for contributors as they can import existing data rather than manually typing it out.

To do so, several steps need to take place:

The AIMS needs to know the location(s) of IATI data and access that data. It then needs to download all potentially relevant data (all IATI activities marked as relevant for Somalia) and store it locally, before making it available as an API. This should be the same API as is used to access manually entered AIMS project (as they will also be stored in IATI format).

This will require a management interface to enter the appropriate address. This is likely to include the IATI registry, but also the IATI Datastore (<http://bd-iati.github.io/documentation/retrieving-data/> and <http://datastore.iatistandard.org/docs/api/>) and potentially other web accessible locations for specific IATI files not included in the registry. Here is the official datastore query builder: <http://datastore.iatistandard.org/query/> and known issues: <https://github.com/IATI/IATI-Query-Builder/issues> and how it works: <https://github.com/IATI/IATI-Query-Builder>. Forward looking project data will come from activity files, but forward looking envelope data may be sourced from the Organisation files: <https://discuss.iatistandard.org/t/using-forward-data/783/10.>

Accessing IATI data will require a thorough understanding of the IATI standard. For example, the IATI standard specifies that when a single sector is given, and not percentage allocation is given, that the sector defaults to 100%. The parser needs to follow these various logics to supply accurate data to the interface.

Plugins will need writing to parse the different versions of the IATI standard that are in use. See e.g. <http://bd-iati.github.io/development/sprint-1/> although it would be preferable to have one plugin for each IATI version (noting: <http://bd-iati.github.io/documentation/versions/>) so that future IATI versions can be handled by uploading a new plugin (not requiring a redeployment, and ideally with the option to download directly from the repository as well as manually upload) by a management user via the IATI data interface. This interface should alert management users that there is IATI format data available in a version of the standard for which they do not have the plugin installed.

An IATI compatible project unique ID scheme should be used, in the knowledge that the Somali AIMS will republish the projects to IATI and therefore the project ID should follow the relevant rules for both organisation and activity identifiers: <http://iatistandard.org/202/activity-standard/overview/iati-identifier/>.

When a user is adding a new project (or editing an existing project), as well as showing similar projects already in the AIMS, the pre-entry interface also needs to display potentially matching projects from IATI sourced data.

If the user finds that their HQ has published the project that they would like to add in IATI, they should be able to select that project (or group of projects) and be taken to the data entry interface with the data on offer/ prefilled.

Care will be needed to help the user to manage this import process. For example, automatically importing all the organisations listed in the IATI data as implementing organisations will lead to a huge list of organisations which may contain many duplicates. Instead the interface should require some minor manual input to match the organisation names given in the IATI data with those organisations that are already setup in the AIMS. In this case, for those that are not already setup, they must be added as a part of the process in the same way a user doing manual data entry would select ‘my organisation is not listed’.

IATI projects that are already linked to an AIMS project should be identified differently. By screening these out, and by selecting their organisation, it should be possible for a contributor to search for and identify IATI projects from their organisation which are not currently in the AIMS to add them. This might involve a checkbox on the pre-entry interface to only show IATI projects that are not already linked to an AIMS project. Entering the user organisation should therefore display all unmatched projects available in IATI data.

It is also worth noting that due to the large volume (and short duration) of humanitarian projects, it is likely that entering them manually will not happen and that IATI import will be the main source. For most humanitarian data, this will happen by importing humanitarian data published by UN OCHA from their FTS database to IATI. It should therefore be possible to import multiple projects at once from IATI source data without having to access the detailed project data entry page. This would be on a restricted basis i.e. that all the required fields can be imported, or mapped automatically, that these are fresh imports, rather than merges between and AIMS project and an IATI project. The ‘direct import’ button should appear accordingly.

## 3.20 Dynamic IATI data

One significant advantage of IATI data, is that it is not static. This gives the possibility that when contributor links (by selecting it in the filtering/matching interface) an AIMS project to a set of IATI activities (activity codes) then they can also set certain fields to dynamically follow the IATI data and be continuously updated in the AIMS whenever they are updated in the IATI data. For the approach in Bangladesh, see: <http://bd-iati.github.io/development/sprint-6/> . An example would be setting the AIMS project to track the financial data in IATI so that the IATI disbursement transactions keep the AIMS figure up to date.

## 3.21 Storing data in IATI format

Given the potential to make use of, and the requirement to be able to publish IATI format data, it seems easiest to store the data in a compatible structure e.g. noting how the standard is structured and making use of IATI codelists internally.

Several of the fields in use in the AIMS are not available in the IATI standard. Whilst they may be in future, until they are, the data in these fields should be published in IATI compatible format through the use of IATI extensions e.g. <https://discuss.iatistandard.org/t/examples-of-well-implemented-iati-extensions/770/4> as well as 'user defined vocabularies' <https://discuss.iatistandard.org/t/specification-of-user-defined-vocabularies/788>.

IATI xml should be the flat file backup format as this will be easiest to import, and possible to read outside the AIMS by users without SQL tools, with the various IATI xml to spreadsheet tools.

## 3.22 Publish data to IATI registry

The AIMS should also publish data to IATI so that the Somali AIMS data becomes available in the many tools being developed to use IATI data and to support general transparency. This same file will also be the backup file format. In order to do this, the FGS needs to be registered as a publisher on the IATI registry and make an IATI xml format file available (again, using the same plugins). The IATI file should be generated live from the database when users request the data. This also means that all data entered by users is public and this should be clear to users in the relevant interfaces.

When filtering and matching projects, there will therefore be a choice whether to include local AIMS projects separately, or import them from IATI (when the Somali AIMS publishes, they will be visible in the registry and Datastore) along with all other IATI data (they will be identifiable by the publisher (Somali AIMS, secondary publisher)). However, because users may want/expect to see their projects reflected immediately in the filtering/matching interface, it seems likely that the data used will be a mix of local data, and IATI data (filtering out the Somali AIMS as a publisher).

## 3.23 Further IATI links (as of late 2016)

Several other AIMS and similar applications make heavy use of the IATI standard, and IATI data:

* IATI data import module for Bangladesh: <https://github.com/BD-IATI/edi> already implements many of the features that the Somali AIMS will need
* Mohinga, the Myanmar AIMS with significant IATI use: <https://github.com/catalpainternational/MohingaV1>
* Development Gateway IATI importer: <https://github.com/devgateway/iatiimport>
* OPIA – <https://www.oipa.nl/home> - but note that this IATI parser implements choices that may not be preferable for the Somali AIMS e.g. if there are three sectors in IATI data, but percentage splits are not given, it assumes equal splits or disbursements. This introduces made up data that is not authorised by the user importing the data. The AIMS parser could import the sectors, but would then need to request the user to provide the percentage splits (of which one option may well be a ‘take equal splits button’. See also OPIA Parser: <http://docs.oipa.nl/en/latest/parser/> and <https://github.com/zimmerman-zimmerman/OIPA/tree/master/OIPA/iati/parser>
* The IATI Secretariat has recently introduced and IATI python module, although it is unclear how/if this will be supported in the longer term. See <http://discuss.iatistandard.org/t/iati-systems-design-architecture-proposal/709> and <http://discuss.iatistandard.org/t/introducing-the-iati-python-library/720>
* ODCS have provided an IATI flatten tool. See <https://discuss.iatistandard.org/t/flatten-tool-new-approach-for-spreadsheet-to-iati-conversion/777> and <https://github.com/OpenDataServices/flatten-tool>
* Aidstream is a web application to convert spreadsheets to IATI xml: <https://github.com/younginnovations/aidstream> and there are discussions around other spreadsheet to IATI tools: <https://discuss.iatistandard.org/t/challenge-create-super-simple-conversion-from-excel-to-iati-xml/630/13>
* There are some more advanced (e.g. with live preview) IATI data filter and select interfaces: <https://discuss.iatistandard.org/t/show-iati-client-side-only-datastore-query-builder/778> and <http://please.treadsoft.ly/query/> which would also share some of the live filter features that the AIMS will require.
* SQL to IATI: <https://discuss.iatistandard.org/t/dfid-sql-to-iati-tool-easy-to-integrate/822/2>

# Data output

Achieving simple data output is vital to the AIMS’ success as without data output, there is no incentive for donors to input data. In addition, data output and subsequent use, is the most effective means of verifying and improving the data quality.

## 4.1 Premade reports

Very few users can make use of the seemingly simple (but in reality, complex) table and chart construction interfaces built into most AIMS. There are too many options, it takes too long, it is never quite configurable enough, and it requires too much technical knowledge. The Somali AIMS review identified that the previous systems took a minimum of 16 clicks, and probably around 20 minutes to make a single table or chart. The Somali AIMS Review also identified that most data requests could be handled by around 7 premade reports (see list below).

Premade reports are those where the data to include, tables and charts required are all predefined, leaving the user just to do some basic filtering e.g. by date, by sector, state or organisation. Basic examples can be seen here: http://www.somaliaaidflows.so/locations/Southwest. In this way, a user without any technical knowledge can produce a report that is likely to answer the majority of their queries within about 60 seconds and just a few ‘clicks’. Requirements for the premade reports include:

* Export in MS Word and PDF formats – fully printable
* Export of tables, and data used for charts in MS Excel format
* Embedded url link to the report (with the latest data) on the AIMS
* ‘Subscribe to this report’ link to subscription page with report (and email if logged in) preselected
* Embed the date the report was produced
* Have full titles including the title of the AIMS, and the report as well as showing the report name and what filters were applied
* Filters to include
* Organisation(s)
* Sector(s)
* Dates
* Location(s)
* Value
* Potentially others as discussed with FGS
* Financial data in Somali Shillings (SOS), EUR or USD

## 4.2 Likely initial reports

* Filterable list of all projects
* Individual project profile – e.g. <http://devtracker.dfid.gov.uk/projects/GB-1-203742/>
* Sector report - <https://mohinga.info/en/dashboard/donor/>
* Location report – e.g. <http://www.southsudanaims.org/reports/location_report/> but including maps
* Organisation report - showing projects involved in and funding provided
* Aid overview/dashboard – e.g. <http://amis.mof.gov.np/TEMPLATE/ampTemplate/dashboard/build/index.html>
* Aid over time report
* Export for budget process including relevant CoA coding and filtered by on/off-treasury

## 4.3 Report plugins and dependencies

Over time, the premade reports will need to change. There will be new data needs, and new fields in the AIMS. This means that the reports will need to be adjusted regularly. The AIMS will therefore need a system where different reports can be developed and then added/removed without redeploying the AIMS. This suggests an architecture where the core technologies of export/import, and industry standard charting and table libraries are built into the main application and then the specific reports describe the fields and tables/charts which makes use of those libraries, and which can be added/removed via a management page. These should be (text, xml files) editable by technical (but not programmer) users e.g. they would follow a set format including the dependencies required, and then describe the ‘view’ in terms of tables, charts, styles and fields that the AIMS would supply to the charting and table libraries (likely JavaScript) needed to produce the output. Upon upload of a new report, the AIMS should check and reject it when the fields it requires are not available in the AIMS e.g. if written to use a custom field. In this way, it should be very simple for non-programmers to add new reports to the AIMS.

## 4.4 Data export

Although most queries are expected to be catered for by the premade reports, not all will be. To assist in these cases, the AIMS needs to offer the ability for users to download raw data. Whilst to some extent this is catered for by the IATI xml format data, this will not be useable by many users and so an MS Excel file download where the user can select which fields they would like to include, should be available. This export will be based on one project per row e.g. as in <http://spreadsheets.aidonbudget.org/> and code: <https://github.com/markbrough/flatiati>. This will provide advanced users the possibility to download basic project data, extend it with their own fields, and produce complex reports in MS Excel.

## 4.5 Subscriptions

There should be a page for users to provide an email address and then select which premade reports they would like to subscribe to, and with what frequency. The AIMS should then email them pdf exports of the relevant reports when required. Users wishing to adjust their subscription should return to the subscription page, re-enter the same email address, and provide different responses, including ‘unsubscribe from all’. Users should also be able to subscribe to be notified of any change in a report e.g. new data. The same subscription page should also allow users to unsubscribe from any updates.

# Users

The Somali AIMS will have several types of users. These will need registering, giving appropriate and differential permissions, notifying of events, and their activities logging. There will be four user types: the public, subscribers, data entry users (contributors) and managers.

## 5.1 Public access

There will be full public access to all areas except the data entry, and management pages. No login or registration will be required to view data, make reports, etc.

## 5.2 Subscribers

Subscribers can subscribe to regular reports (in addition to receiving specific mailings sent by management users).

## 5.3 User registration

All users will self-register (name, email address, password organisation – incl. adding new organisations when not already setup), and selecting the options that they require (contributor, manager) and this will be sent for approval. When registering as a contributor, the request will be sent to all other registered users of that organisation (as well as to the managers group), allowing one of them to approve the registration. If they are the first user for that organisation, the registration request will just be sent to the managers group. Users requesting manager privileges will have their request sent to the managers group and need approval from an existing manager. Registered users wishing to change their privileges should access a page to delete their accounts, or to add/remove privileges and/or organisations. This page should also show a user all the other registered users in their organisation, and if a manager, all other managers. If no management users remain, the AIMS should revert to the initial setup/walkthrough screen requesting the email of the initial management user.

All users not active for more than 2 years should be emailed three reminders to log in, otherwise their account will be deleted. If there is no login within 3 further months, the account should be deleted and the user emailed. These emails should be copied to the other users in the same organisation. The log will still record the original user details.

## 5.4 Organisations

Management users will need to be able to manage the organisations in the AIMS, for example, merging several organisations together (e.g. for naming differences), and seeing which organisations have no users (e.g. when added to a project by another organisation) to go and find users.

## 5.5 Permissions

Anyone who belongs to one of the organisations associated with a project (funder, implementer) can add new projects, and edit existing projects where that organisation is involved. This principle is designed to minimise the chance of orphaned projects due to high turnover of contributor, and allow several organisations to collaborate on reporting joint projects to reduce double entry. To edit a project, a user will have to request their organisation to be added by a user of an organisation already part of the project.

Management users have no special rights to edit projects apart from those of their organisation.

## 5.6 Notification

At several stages, the AIMS will need to notify data entry and management users of events and changes. This notification should take place by email, and by an indicator visible when logged into the AIMS. Where there are multiple notifications, emails should be limited to one per day. Notifications should not resend. Examples of notifications will include:

* A user would like to add their organisation to a project your organisation is involved in
* Availability of new external IATI data in one of your organisation’s projects
* A user from your organisation has joined the AIMS
* A user from your organisation is inactive and will be deleted shortly
* A change in a mapping has affected one of your organisation’s projects

The user will need to login to the AIMS to perform any relevant actions. The email should contain a link directly to the appropriate page and to the general notifications page. Where possible (e.g. not requiring a specific user input, e.g. just an approval, this action should be possible on the notifications page). The notification should remain until the user marks it complete.

Users who are logged in should also have a notifications indicator visible in the AIMS which leads to a page listing outstanding notifications. These should be removed when actioned. As some notifications will go to multiple users, once actioned, a notification will need to disappear for all users who received it. Notifications will need to be backed up alongside the data and configuration files but this can also be as plain text.

## 5.7 Logging

The AIMS will have no approval process when users edit project data. The model is that users (organisations) are responsible for what they have entered. Any users who are unhappy with a data change should either edit it themselves if they have permission, or contact the relevant organisations. However, to be able to resolve disputes, or to track changes, the AIMS will need to have a log of all changes to project data visible to all data entry and management users. This should specify the user involved (including changes because of a change in remappings), give a timestamp, a link to the project, and the before/after values. The AIMS does not need to log non-project data changes e.g. actions of management users, production of reports etc.

# Design

Several aspects of the design are worth comment.

## 6.1 Modular nature

To ease upgrading and adding additional features in the future, the AIMS application should be modular. Different components should interact using standard APIs.

For example, the AIMS should expose the both the entered projects, and the locally stored IATI projects via an identical API which returns IATI format data. This can be used by many different parts of the system including the data entry module to show the current and available projects, by the backup module to make a backup and by the reporting module to generate reports. All the different data entry/edit options should use the same API to write data. This (and other) APIs should be made as future-proof as possible e.g. by allowing for the various fields to change. Specific documentation for the various APIs should be provided on the repository. Likely APIs include:

* Database reading and writing
* Requesting exchange rates (with plugins for the different sources)
* User authorisation and authentication
* Writing to storage locations (e.g. local, Dropbox, Google Drive, OneDrive, FTP)
* Emailing users – (with plugins for Mailchimp, Gmail, local server etc)
* The notifications system

It may be decided that some of these APIs will be exposed publicly to allow interaction with other FGS systems.

## 6.2 Multi language

The AIMS should be useable in English but may be extended to Somali at a future data so considerations should be given during development.

## 6.3 Useability

There are a number of usability requirements:

* It is rare that decimal places are required, all instances should be checked if necessary and removed where possible to clean up the page.
* Currency should not use symbols e.g. ‘$’, instead using e.g. ‘USD’ or ‘SOS’ as the currency codes - and for all other currencies, follow ISO codes: <https://www.oanda.com/currency/iso-currency-codes/>
* Links should be clearly defined i.e. not clickable images, no animations.
* The style should be as simple as possible e.g. white background, standard fonts, minimal graphics e.g. [http://www.southsudanaims.org](http://www.southsudanaims.org/)
* Use of the screen should be maximised e.g. avoiding multiple layers of titles, borders, etc. Similarly, there is unlikely to be need for a visible footer.
* The need for scrolling should be minimised.
* User Interface elements performing similar functions should maintain their approximate positions across multiple pages e.g. filters should always be in the same place.
* The AIMS should be optimised to work on screen sizes of 1024\*768 and upwards, including a responsive to adjust for viewing the AIMS (but not data entry) on mobile devices.
* The AIMS should scale for use on mobile devices with a reduction in functionality e.g. showing the reporting views only.
* The minimum data required should be requested. For example, a typical user registration form will include: title, separate first and last names, role, organisation, street address, telephone number(s), fax numbers, email address, etc. Many of these are not possible or relevant in Somalia and the AIMS requires a single name field, and email, and perhaps a phone number.
* Many users will have very low bandwidth connections. Reducing the number of images, and the number of data exchanges and browser dependencies will be key, as will be steps to improve the resilience of connections to interruptions.
* The main menu should always be visible, including the option for return to the home screen.
* No ‘hover’ menu’s, rather ‘click’ which must persist for a period i.e. not disappear as soon as the user’s mouse moves away.
* The AIMS must have a consistent/single user interface for all components
* The AIMS must be compliant with the appropriate web standards e.g. w3c and function on multiple modern browsers both on MS Windows and on other platforms. The AIMS should not require browser plugins such as Flash and Java.
* The AIMS must have a consistent and meaningful set of well-documented error codes and facilities for error analysis and reporting.

## 6.4 Scale and concurrent users

The AIMS should be scalable to add up to 1,000 projects a year for 10 years (averaging 500 per year), allowing up to 10 concurrent users (5 entering data, 5 getting reports). Space should be allowed for attachments averaging 2 per project.

## 6.5 User help and guidance

Manuals for AIMS systems are rarely a success. By the time a user has found and accessed the manual, they have lost interest. Therefore, all necessary user support text should be built into the interface. A consistent way of indicating help text (e.g. an icon) should be developed and in each case located directly by the relevant field or button.

## 6.6 Contact page

This page should present a form (requesting a name and email address) that will email all managers detailing the user issue. With Captcha to prevent spamming.

# Site structure

This section contains some additional guidance about key components of the various AIMS pages discussed above. While these items are expected to be included, it is understood that the development process may reveal alternative and improved approaches that may call for a different structure and design.

## 7.1 Core pages

### 7.1.1 Main template

The main template should maintain the same layout for all pages. It should define content blocks for the title, menubar, filters, and content. The menu-bar should:

* Highlight the selected option - no on-hover, on-click to show submenu's where necessary. The submenu’s should persist until another click is made.
* ‘Menu' which returns user to homepage
* ‘Reports’ which presents the list of premade and other reports
* ‘Data entry’ which takes the user to the filtering/matching page after a login
* ‘User management’ which allows the user to select registration, subscription, etc.
* ‘Management’ which allows access to management pages
* The menu bar should only have two levels, the initially visible top headings, and then sub headings linking directly to the specific pages.

### 7.1.2 Login

Users will need to login to access management and data input pages. This should not be a popup as many uses have popup blockers, ideally a ‘mini form’ that expands from the menu to take username and password if not already logged in. Users should be auto-logged out after a period of 1 day.

### 7.1.3 Homepage

A brief description of the purpose of the AIMS, including:

* that all data is public and free to access, description and links to premade reports
* that it is managed by the Ministry of Planning, Investment and Economic Development, FGS, and that the data is entered by donors and implementers
* some key data e.g. welcome to new organisations/users W, there were X new projects, projects Y and Z closed.
* Options/links to subscribe, and to register to add data
* Link to contact page for further enquiries

### 7.1.4 Contact page

The contact page will have a form to contact managers – emailed to all email addresses in the management group.

## 7.2 Reports menu

The reports sub-menu will list the various custom reports available starting with the list of projects.

### 7.2.1 List of projects

This page should show a table containing all the projects and showing basic data. The table columns should be appropriately filterable e.g. time, sector, location, text search, value, funder/implementer, etc. Each project name should be a link to the relevant project profile page. The filters should be persisted in the URL, so that the page with filtered results can easily be shared with others.

### 7.2.2 Example report page

* Report title
* All necessary filters including currency
* Options to export to MS Word, pdf, or export data to excel (xlsx, not csv)
* Report area showing live report content, typically a table, and some charts
* Link to report query e.g. <http://somaliaims.org/reports/sector/pdf/?sector=education-startdate=xyz>

## 7.3 Data entry menu

There are two pages concerned with data entry. Users will initially access a pre-filtering page which is designed to guide the user to making use of and checking against data available through IATI, and data already available in the AIMS, before starting manual data entry. The second stage is to manually enter any further data required.

### 7.3.1 Filter and match pre-entry page

Selecting ‘data entry’ will trigger a login and if authenticated, will lead users to the initial data entry page which requests basic information in order to present a filtered list of projects from both the AIMS and from IATI data. From there, users will either choose to enter a new project, edit an existing AIMS project, or import IATI data into a new project:

Can pre-fill/filter for organisation. This can be unselected to show projects from all/other organisations where the user’s organisation is not a member.

Has data entry fields for project Name, Sector, Organisations involved (and others if necessary to get sufficient filtering). These should be dropdowns or text search / autocomplete / autosuggest as appropriate. For fields with multiple schema, the user should choose schema first. Care (and some manual input) may be needed to help link the user organisation to the right IATI Organisations/publishers

As basic data is entered, the table below filters (on-the-fly) and a method for ordering (ranking) the projects shown by likelihood of matching needs to be applied.

AIMS projects and IATI source projects need to be differentiated e.g. by background colour. IATI projects already linked to other projects should be differentiated and grouped under the AIMS project that they are already linked to.

Filtered projects need checkboxes or similar, to allow the user to select them (including multiple).

The user can then choose to continue with the selected projects (i.e. edit or import depending on the source) or alternatively enter new project based on blank data entry form.

Continuing with the selected projects transfers the user to the project data entry page with as much information prefilled as possible, either by containing the details of the existing project, or of the IATI activities selected. Where there are multiple pieces of data, or fields that require user input, the AIMS will highlight the areas where attention is needed.

For groups of projects which include AIMS projects where the user’s organisation is not a member, instead of ‘continue with these project(s)’ the button should instead say ‘request to add my organisation to this project’. This will notify the users of the organisations who are already involved in the project to approve (or not) the adding of a new organisation.

### 7.3.2 Project data entry page

This is the main page for entering project data.

Group 1 should contain the basic project data e.g. name, description, dates, organisations, sectors

Start and end date, and project name are mandatory

Users cannot remove all organisations, one must remain. Users cannot remove their own organisation.

Group 2 should contain the financial data

Users can select from the listed funders and add the amount provided (and exchange rate to allow conversion to USD) and grant/loan status of the amount. The total project value should calculate from the amounts accorded to different funders (multiple amounts from the same funder should be possible e.g. a grant and a loan).

This page should allow the update by Somali Financial Year of actual disbursement totals, and the calculation and adjustment of annual expected disbursement amounts by Somali Financial Year. This should be achieved through offering the user tools to enter data in other Financial Year’s and then have the AIMS make a conversion, as well as calculating remaining funds, and remaining years based on the project total value, disbursements to date, and years passed/remaining. In all cases, the user should be able to adjust the amounts/percentages, and the AIMS to compensate fields based on expected disbursements to keep the total of actual and expected disbursements equal to the project value.

Group 3 should contain the geographic/location data

The AIMS will record data at two levels, Federal Member State/Region level, and a sub-level for districts.

This needs to be done via a more user friendly method than repeatedly clicking and selecting from a dropdown as users will often have to add multiple regions. A suggested (equal) split of the project value should be calculated by the AIMS on the fly, but with the ability for users to adjust the split across locations and the AIMS to maintain the total across locations as equal to the total value by adjusting any fields that the user has not individually adjusted.

Group 4 should contain the custom data fields

Custom fields should be visible when appropriate triggers in the core (or other custom) fields are triggered.

In addition, the following features should be available:

* Autosave - no need for a save button, auto publish
* Button to ‘work on another project’ which returns user to the pre-entry page
* A button to delete the project
* A button to take the user to the log prefiltered for the specific project

## 7.4 User pages

### 7.4.1 Registration page

This page will allow users to register as contributor or management users. Users should also be able to add a new organisation on this page if their organisation is not already in the AIMS.

### 7.4.2 Subscription page

This page will allow users to subscribe to (and receive via an email address they submit) various reports, according to a schedule and frequency they select. It will also allow users to unsubscribe.

### 7.4.3 Password reset page

All users need to be able to access a page to enter their email address to reset their password. This should not require any management input.

### 7.4.4 Notifications and approval page

This page should display all notifications, grouped appropriately. Giving options to silence them (in which case they are hidden – with a checkbox to unhide, or mark them as done – with another checkbox to unhide). Notifications will include notification of a new user joining the user’s organisation, the need to approve a new user to join the user’s organisation, a request to add an organisation to a project that the user’s organisation is a part of, and changes to project data stemming from new data from linked IATI activities, or changes do to a remapping of an automapping.

### 7.4.5 Envelope data entry page

This page should allow data entry users to update the funding envelope data for their organisation.

## 7.5 Management pages

These pages allow management users to manage the AIMS.

### 7.5.1 Reports management page

This page will allow management users to upload new reports, and remove unused reports. It should also show how many times each report has been accessed in various time periods.

### 7.5.2 Mailout page

The mailout page allows management users to send emails to all the users. This page should have text boxes for the email subject, content, attachments, and checkboxes for the different user groups e.g. subscribers, contributors, and managers.

### 7.5.3 Email settings management page

This page should allow managers to manage the settings necessary for the AIMS to send emails e.g. entering mailserver connection details, or details of a mailchimp or gmail account.

### 7.5.4 IATI settings management page

This page should allow management users to enter settings for connecting to and publishing IATI data e.g. the publisher details, the location of the registry and any other IATI data sources. This page should also remind users of the location of the public IATI data file.

### 7.5.5 Storage management page

This page should allow management users to enter details of and select the storage options (including duplication) for the AIMS e.g. local folders, or cloud storage accounts. This storage will cover the backup files (appropriately named) and any uploaded documents, sorted into a folder for each project.

### 7.5.6 Custom fields management page

This page should allow management users to edit the custom fields available, and change their order.

### 7.5.7 Organisation management page

This page should show a list of organisations and their users, allow grouping similarly named organisations and combining under one name, as well as showing organisations without users.

### 7.5.8 Exchange rates management page

This page should allow management users to manage the plugins which provide exchange rates for data entry, enter data for the official FGS FMIS exchange rates, and set a default exchange rate source.

### 7.5.9 Codelist/Mapping rates management page

This page should allow management users to manage the codelists and associated mappings. Changing a mapping should not affect already entered projects. Codelists of the input side of an automapping will also need to be possible to edit e.g. to add new donor classifications, or adjust to a change in IATI data e.g. <https://discuss.iatistandard.org/t/align-sector-codelist-with-the-latest-version-published-the-dac/771>.