Requirements

This document defines the requirements for the required solution, however, it will not provide a complete picture of the solution without supporting documents as well as demonstrations of our current solution as most of the functionality will be derived from our current solution and then iterated upon.

Also, this is not the final version of this document. This document will be iterated upon after discussions, clarifications, and negotiations with the development team.

Accounting module features

A transaction entry stores a debit and credit towards an account. Transactions are always stored in vouchers. There can be a large amount of transactions and vouchers serve to categorize, more importantly, group all of them. A transaction cannot exist outside of a voucher. A voucher always originates from an office, and it always has a currency type.

Budget lines are a fixed amount of budget that are designated for different purposes mainly to be utilized for projects. Budget lines serve to track different income sources in for the organization.

A trial balance is a view and a sum of all the credits and debits attributed to all accounts via transactions. I.e., all the credits and debits done by transactions (not just the current amount in an accounts debits and credits). The **sum** is just a sum of all the debits and credits separately in the view. The objective for this view is to check if all the transactions balance out.

Notes are a data construct primarily used to group different account codes together for viewing/analysis/review purposes. Generally, notes are used to create consolidated reports of large number of accounts.

Exchange rates. An exchange rate for a currency must be added every day by each office. The organization uses local exchange rates due to geographical limitations, and therefore in order to be up-to-date with the local currency exchange rate, the exchange rate must be stored manually every day. All calculations that involve currency exchange will be using the daily exchange rates that have been entered into the system. If an exchange-rate for a day is not available, the system will use the previous day's exchange-rate.

Opening and Closing vouchers. At the end of the year, a closing voucher is created that is generated by subtracting the current credit and current debit from each other for each account and the value that is higher will be the one that is subtracted from and the outcome of that calculation will be what is used for the opening voucher for the account for the next year.

Exchange gains and losses are also calculated in the opening and closing voucher generation process. Effectively exchange rates from throughout the year are compared to the exchange rate for a currency for the 31st of December every year. If the calculation determines that the value of the currency is more at the end of the year than what it was valued at the rest of the year (overall, considering exchanges that have taken place throughout) then we have made an exchange gain.

The Financial Report is a group of multiple consolidated reports that are themselves grouped using notes. The following are all the reports:

- **Notes -** This page defines functions like SUM, CR, DR for notes
- **Notes in COA** - This page specifies notes for different accounts. The selections made in this view will determine the results of the Balance Sheet and Details of Notes views. Effectively, all sub-accounts of accounts assigned to notes in this view will be used to do a sum in the Details of Notes (in a grouped manner) and then the totals are displayed in the balance sheet. The description field of notes in the Balance Sheet is also taken from note narrations in Notes view.
- **Balance Sheet** - The balance sheet view displays the totals of grouped sums done in the Details of Notes for their corresponding notes. The notes here are taken from the Notes view.
- **Income/Expense Ac -** Only a specific set of notes appear in this view. Namely those assigned CR and DR values. In this view, for each entry the note must be assigned a DR and CR account in the Notes view otherwise it is not valid. If the CR value for a note is higher than the DR value then the balance will be in positive, otherwise it will be negative.
- **Details of Notes -** This report shows the debit and credit amounts of all accounts. In the report, all accounts are grouped under different notes and are displayed as such. Multiple accounts assigned to a single note will be displayed under the note the the report and then summed at the end.
- **Donor Expenditure Report -** This report provides a summary for all debits and credits done for different budget lines and their current balance.

Functionally from a user perspective they need to be able to do the following:

- Create and submit transactions (clerks)
- Reject and approve transactions (control users)
- View a journal of transactions. The journal displays transaction information using vouchers as a basis.
- View a ledger of transactions. The ledger displays purely transaction information and only provides a reference to the vouchers that they belong to.
- View trial balance. Export trial balance sheet to an excel file.
- View Budget Balance. View paper-based report on budget balance. Export the results of budget balance sheet to excel.

System based functional requirements:

- Users do not interact directly with transactions and vouchers within the system. When they create a transaction, they are shown a form that lets them describe the type of transaction via a dynamic category system. The system will define which vouchers and accounts the transactions will land in.
- Create opening and closing balances/vouchers at the end of the year.
- Calculate exchange gains and losses at the end of the month and place them in transactions for appropriate accounts.
- Calculate all payroll at the end of the month, generate the relevant payroll entries in the payroll module.

Project Pipelining Requirements

This module is a state-based task management system and the flow of the states will be described in the Visio diagram that is provided alongside this document. Please refer to that for the flow definition. Keep in mind to go through the comments in the Visio diagram as they additional information on corner cases, requirements, technical details etc. Following are some high-level points on the project pipelining module:

1. A single place/dashboard where all flow activities are handled.
2. We create an activity data model to integrate "activities" from different modules into the projects module. Effectively users can specify a "reason" for each activity that they are doing in a module. E.g. Control users can specify that the account voucher entries (activity) that they are doing is for a specific task/state in a specific project (reason). If a user specifies the reason for an activity to be a project task then they must be able to also specify when a task is complete so that a director can approve it.
3. Each document can have multiple attachments and each attachment has collaboration/versioning.
4. Flow will be represented in a single place, and we will be eliminating situations where items have to go through several different views to represent its state in the flow. All people who are involved in the flow will have access to a newly created project. The flow itself will be represented by a list of tasks that appear as they are needed. The tasks themselves are pre-determined either by admin, or hard-coded if there isn't enough time to implement (develop) a flow manager module. Each state may require a single task to take place or multiple. Flow tasks that have to do with other modules will need to be tracked via the activity system.
5. Once a user flags a task for completion, it will need to be approved by the supervisor/control user for the task before the next tasks in the flow are queued.
6. Projects and tasks both need to track time in order to enforce certain time-sensitive policies. Users will need to be notified once a certain time threshold has passed for a task or a project. There will be color coding based on time passed on a task or project.
7. There is not only one user who approves/rejects tasks in all the steps in the workflow. We should be able to assign different users to approve/reject/finalize different tasks/activities/steps in the workflow.
8. There might need to be a way for "someone" to specify the requirements for different activities/tasks in the flow. Maybe the person who is supervising the activity will need to specify it, they will also need to be able to delegate tasks to other users.

A meta state is a high-level workflow state that each involve mini-workflows inside them. We can have it where a list of parameters/specifications is set for the meta-state (maybe at the first stage of the meta-state workflow) and once all the specifications are met, the meta-state will be flagged as complete by itself. The alternative is to have one person given the rights to flag a meta state as complete whenever they see it fit. I think we need to have a combination of both, where we are both tracking pre-requisites and waiting a person to track the task as complete, it might help to have some room for error here because the person who set the pre-requisites might have missed a few things out. Effectively, a meta-state can have all of its pre-requisites met and still not be complete because someone deemed it necessary to perform a few more tasks/activities. The first place that meta-state will be used is in the logistics and control department has to work together to make sure a list of items to be bought is approved.

- Different users will need to perform different tasks within the same meta-state. We want to reduce clicks-to-destination, so how do we represent user-specific information/functionality to different users within the same state. One way might be to present users with a link or button that re-directs them to their part of the logistics module dashboard.
- There needs to be a way to represent to a user that there is something new for them to do specifically (not that some new activity has taken place within that domain). Mostly, users don't care unless there is something they need to do themselves.

Fields Grouping

These are the fields that every project will have, grouped into logical field sets. Field sets will be used to display fields in different tabs in the project pipelining dashboard.

- Project Details
  - Opportunity No. Type
  - Opportunity #
  - Opportunity Description
  - Project Goal
  - Project Objective
  - Main Activities
- Project
  - Province
  - District
  - Sector
  - Office
  - Start Date
  - End Date
- Budget
  - Budget
  - Direct Beneficiary Male
  - Direct Beneficiary Female
  - Indirect Beneficiary Male
  - Indirect Beneficiary Female
- Donor
  - Donor Name
  - Donor Contact Person
  - Donor Contact Person Cell#
  - Donor Contact Person Email
- Documents
  - Concept
  - EOI
  - Proposal
  - Budget

   - Presentation
   - Contract
   - REOI Receive Date
   - Submission Date

Administration Functionalities:

1. Assign permission-sets to users
2. Define organization hierarchy
3. Assign users to different roles in the hierarchy
4. The organization will have multiple offices, each office will have multiple departments. Each department has a supervisor and employees.
5. The organizational hierarchy will come into play in the project management pipeline because supervisors will need to assign task ("activities") to their subordinates. Supervisors can only assign tasks to users in their department.
6. If a supervisor wants to assign an activity to a person that does not have the rights to be able to perform the activity, the supervisor needs to be prompted and given the ability to assign the relevant permissions to the user. These kinds of ad-hoc permissions should be temporary by default and the supervisor should be able to choose to assign the permission permanently.
7. Meta permissions. These are high level permissions that are assigned to a user upon being assigned to a department. A meta permission will mainly have two types, one for supervisor permissions and other for normal employees. The reason behind meta permissions is for a department to have a logical connection to a specific module. So, a finance department might need to have permissions to the accounting module by default. Meta permissions are optional for a department mostly because this is a loose connection, they assign a permission set to a user when they are added to a department and upon removal from a department, it ensures that those permissions are removed. It is a simple way of saying:

```
User.OnEnterDepartment(int depId){

      if(depId == 3){

      permissions.Add(User, permId);

      permissions.Add(User, permId);

      }

}
```

Organizational hierarchy will come into play mostly during data creation, for example when vouchers are created in the accounting module, the office that the voucher was created in will automatically be entered. In the old system, it would have to manually be entered.

System-based administrator functionalities:

1. There are some permissions that can only be attributed to users with specific roles in the organization hierarchy such as access to specific departments, department based notifications etc.

Notifications

Users only need to be notified for things that are relevant for them. However, the user should be able to subscribe to more specific notifications e.g., department supervisors may want to be notified of the activities of their department employees.

Along with notifications, users will need to have a live feed of notifications. This will especially be useful when it comes to the project management and project pipelining.

Project Management Module (PMU)

This module involves having a GANTT based project manager for each project that has been finalized in project pipelining.

- There are business specific fields that need to be added for each project activity which will be showing via demonstration of our current solution.
- The PMU will need to be tracked as an activity in the project pipelining system as well so that involved members can have access to the project at any time.
- There is a review process that also takes place on every project task. Users need to be able to communicate on each project task in the project manager.