Cashflow Management System: Functional Specification

Table of Contents

1. Introduction

This document describes the functionality of the proposed Cashflow Management System (CMS). It is designed to provide a reference for the development of the software, to agree details of the implementation of the software and to form the basis for the costing of the project.

The CMS is modelled on an existing spreadsheet-based system already in use by Top Security and related companies. The aim is to recreate the same functionality and reporting in a way that will be immediately useful to Top Security, but also of general applicability to a wide range of companies. In other words, the aim is to productize the existing system so that it could be sold as a service to other businesses. This proposal should be viewed as a first phase in moving towards this goal - not as the end product.

The system will be modular and cloud-based, which can be easily scaled from low-level usage to high-level usage, without incurring exceptional costs relating to hosting/hardware etc. It is a key part of the proposal that the Developer configure and maintain the hosting directly, via Amazon Web Services. This allows us to stand over the availability and maintainability of the system.

The system will be an HTML5/Javascript Web Application, which will be viewable in a web-browser or on nearly all mobile devices (iPad, iPhone, Android Phone and Tablet etc…). This will save greatly on development costs, as Native App development would mean doing new work for each platform.

The system will be built using entirely open-source and Linux-based software for the development of the system. This means we will not be tied into any 3rd party proprietary technology.

2. Technology

Every technology choice made below aims to adhere to the following criteria:

- Stability and longevity of technology support
- Speed of development
- Flexibility and control of features
- Not getting locked into a specific vendor
- Application performance
- Ease of hiring suitable developers in the future

2.1. PostgreSQL + Redis

Most data will be stored in a PostgreSQL Database. Postgres (for short) is an open source, relational database. It is widely used by large scale enterprises such as Sony, Yahoo and Instagram. The database will be on an AWS hosted instance providing automatic backup and multi-zone availability (i.e. zero downtime).

Data such as system settings, localisation data and rules which the system may have to read constantly, but write rarely and which, therefore, will require very rapid access will be stored in Redis. Redis is an in memory Key-Value store, or dictionary. It allows very rapid access to this kind of basic data.

2.2. Node.js / Expressjs / sequelizejs

Access to data and execution of core business logic, will take place via an HTTP REST API.

By constructing the system around an API, we will keep the system flexible and also provide a uniform interface for future development. This will make the development of native applications (iOS/Android) much easier going forward. RESTful API's are widely used and understood by millions of developers, so they are the most sensible choice when considering future development. Using an API also opens the system to possible 3rd party integration in the future which is increasingly considered industry-standard on the web.

The development of the API will be done with the Node.js programming language, using the Expressjs Framework and Sequelizejs as an ORM.

Node.js is a well established development language with thousands of open source extensions, meaning developers can leverage code already written by others to quickly build working systems. At the same time when complex changes are needed they are free to modify everything they have worked with. Expressjs provides a framework for simply and quickly building web applications and REST APIs. Using Node.js also means that we will be using a single language, Javascript, for the entire system. This makes maintenance and hiring easier in future.

Sequelizejs is an ORM which provides uniform access to a range of different Databases. This will allow us to maintain a level of independence from any specific Database as development continues.

2.3. Docker / AWS

The Docker platform is a container which can run an App bundled with all its dependencies on any Hardware/OS combination. This makes scaling, deployment and testing much easier. It makes it possible to deploy new application servers in minutes. It will also allow us to run the App on any hosting service we wish and to quickly redeploy it to another if we should change providers.

At present it is intended to deploy the Dockerized App(s) onto AWS Elastic Beanstalk. This service allows high availability, automatic scaling (load-balancing) and automatic backup for a minimal cost.

2.4. HTML5/Javascript/AngularJS

The user interface will be a web application built almost entirely in HTML5 and Javascript. This app will be able to run in any modern web browser, on most tablets and mobile phone devices.

The development will leverage the AngularJS framework which was developed by Google to create responsive interactive web applications. This will save a great deal of development time, while still maintaining maximum flexibility.

There will also be a thin layer of server-side code for managing secure authentication.

2.5. LogEntries

We will use the LogEntries system to maintain debugging, security and audit trail logs for the entire system.

2.6. Github

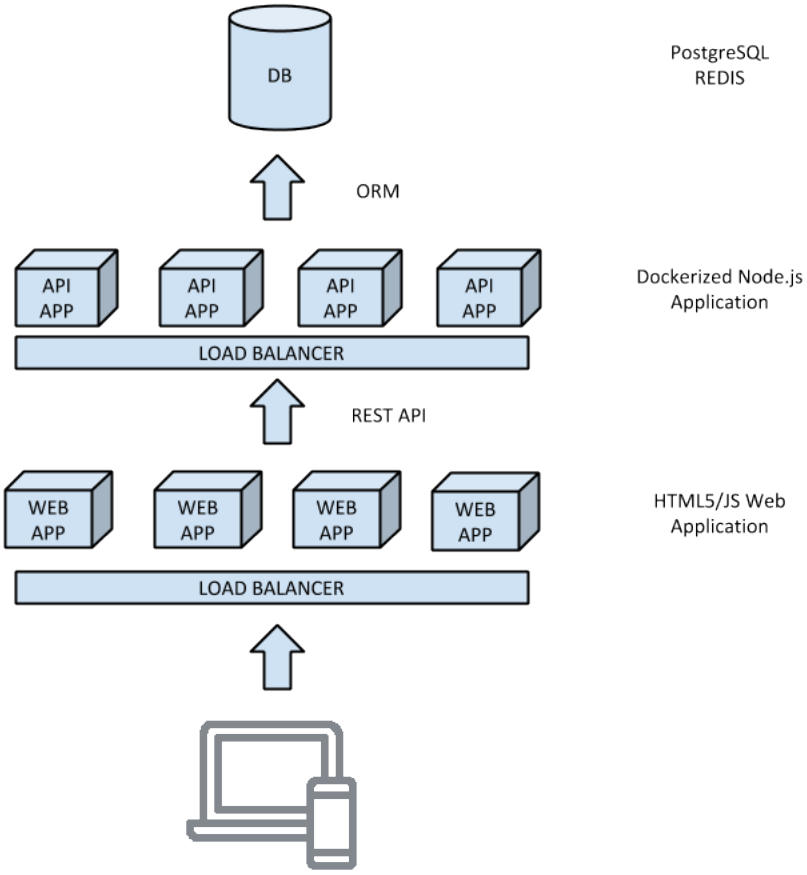
The entire source code of the project will be committed to Github on an ongoing basis. Git provides version control and acts as an effective remote backup. It is also useful as part of the deployment process in a scaleable application.

2.7. Scheduled Tasks

All tasks requiring scheduled execution will be designed as Node.js scripts and scheduled to run with cron. The cron daemon will be configured to run in the Docker container. This means each deployment will have the power to run scheduled jobs across the entire system. However we will use rcron to ensure that only one machine at a time takes the actual responsibility for scheduling execution.

3. System Design

The overall system will look something like this:



3.1. DB

The Database will store data for:

- *Users*
- *Companies / Departments*
- *Flows (In/Out)*: a list of either positive or negative flows (i.e. income streams and or expenditure types)
- *Budgets*: budgeted figures for each flow for each day
- *Dailies*: actual figures for each flow for each day

- *Balances*: Stored balance at end of every day
- *Notes*: notes made by Managers
- *Messages / Notifications*: a queue of notifications sent by system (Email, SMS etc)

The Redis Key-Value store will retain:

- *System Settings*: System wide variables
- *Localisation Data*: Locale specific text, currency and datetime info
- *Rules*: company specific rule definitions for visual highlighting or notification of problems

For detailed information on data structure see below.

3.2. API

The API provides:

- Secure Authentication
- Access to data in DB
- Business Logic

3.3. Web Application

The Web Application provides:

- Operator Interface
- Manager Interface
- Administrator Interface
- Super Administrator Interface

3.4. Security / Authentication

Both the API and Web Application will communicate solely over HTTPS. HTTPS (secure) connections will be enforced for all users connecting to the service. A wildcard SSL certificate will be obtained from a trusted authority and used to verify the identity of all servers within the system.

All connections to the Database will be locked down by IP Address. Only our own API Servers will be able to make a direct connection to the Database. Web clients will only access data via the API.

Login to the system will be via Email and Password. Strong passwords will be enforced. Once logged in the user will be cookied. The cookie will persist for one week and will allow automatic entry to the system without re-entry of credentials. On each re-entry the cookie will be refreshed. There will be an option to disable this feature, if the system is being accessed from a non-private computer. This is approximately similar to the mechanism for a service like GMAIL.

Every time a user logs in, or is passed through based on a cookie the following data will be recorded in the Audit Trail:

- Username
- IP Address
- User-Agent Data (i.e. browser, device etc)
- Success/Failure

In the first phase registration for accounts will be by invitation only (see Registration).

3.5. Localisation

The system will be built with full Localisation. In other words all messages, captions and other text will be stored in a Localisation Database that can then be used to substitute alternatives in other languages. Localisation will only extend to Latin Alphabet languages in the first phase.

Additionally currency, datetime and other locally changeable matters will be made capable of substitution.

There will be no interface for management of Localisation in the first phase. We are simply ensuring that the system can be localised in later iterations.

4. Data Structure

By default all tables will employ Unique ID, Date Created, Date Deleted, Date Modified columns. I don't specify these in detail for each entry.

4.1. Periods / Date Management

The default settings will be Month long periods, divided into 5 day weeks. The average user will be unaware of any greater complexity than that.

However, the day will be the actual core unit of time in the system. Everything will be divisible to a single day period if needed. This will allow for maximum flexibility in setting the budget periods, dealing with weeks that contain holidays or 6 day weeks (eg. Israel) etc.

All settings configured for periods will be at Company/Department level. This allows flexibility across multi-company organisations.

Whatever settings are configured the system will define the user interfaces based on individual tabs for week long periods, with the correct number of tabs for the number of weeks (or partial weeks) contained within the overall period (usually a month).

By default weekends will considered inactive and hidden from display. It will be possible to reactivate them, but this will be considered a rarely needed function and hidden away in the Administrator's interface.

There will be a mechanism for users to specify when a particular day is inactive (i.e. Bank Holidays etc). It will also be possible for an Administrator to specify inactive days in the Calendar ahead to any future date. They will also be able to select a template of inactive days from a country-specific list. The purpose of deactivating a day is to prevent any notifications or warnings relating to the filling in of daily figures. Also we will visually highlight inactive days. Reactivation of inactive days will be possible for Operators, Managers or Administrators. This is convenient, but may need future review.

Budgets, Dailies and Balances will all be stored on an individual day basis in the Database.

4.2. Companies / Departments

A company or department is the basic unit of organisation for a single budget. In other words what is currently contained in a specific spreadsheet file will be contained within a company/ department. We will only differentiate companies and departments by name. This allows us to sell the product to SME's and Enterprise without changes to the underlying functionality.

Each company/department will comprise:

- *Name*
- *Address*
- *Operator*: the user with responsibility for budgets and daily figures
- *Manager*: the user with oversight of budgets and daily figures
- *Credit Limit / Overdraft*: the level below zero at which a balance should trigger a warning

The *Operator* will always be the user assigned the responsibility for budgeting and filling in the daily figures. If no user is available the company will not be active. The system must prevent any user in the Operator role from being removed from the system.

4.3. Users

A user is always a single, real person. That person may occupy many different roles within the system, but they will always have only one login. The system will be responsible for ensuring they have access to the correct interfaces and permissions.

A user will comprise:

- *Email*: choosing an email for user identity guarantees global uniqueness
- *Password*
- *First Name*
- *Surname*
- *Address*: Country and City will be specifically obtained
- *Phone*

A user will able to occupy one or more of the following basic roles:

- *Operator*: this represents the person setting budgets and filling in daily figures.
- *Viewer*: this represents a person with the ability to merely view the daily figures. This role does not receive notifications, enter notes or data. In the real world this will simply be an additional person the Manager has invited to have oversight of one or more budgets.
- *Manager*: this represents the person viewing the data input.
- *Administrator*: this represents a billable entity, who has the power to create users and companies.
- *Super Administrator*: this represents the Administrator of the Cashflow System

In a conventional scenario an Administrator will create a Company and a user. They will then assign that user to the Operator role for that Company. The Operator will then set budgets and input data. The Manager will monitor the data. The Manager will often be the same actual person as the Administrator.

However, to maintain flexibility, the system will allow any user to occupy different roles across different Companies. The system will not allow the user to occupy two roles in relation to the same company. This means:

- A user CAN be the Operator for Company A and the Manager for Company B
- A user CANNOT be the Operator for Company A and the Manager for the same Company (i.e. Company A). It makes no sense for the system to allow the same person to both input figures and provide oversight. This enforces our philosophy of connecting people.
- A user CAN be both Manager and Administrator. Usually this will be the case.

4.4. Flows (In/Out)

A flow is simply a category, which expects either a positive or a negative value. Although it may represent very different types of income or expenditure, all it needs to be, for the system's sake is a number.

Some rules:

- A flow's value must match its expectation. i.e. an inflow cannot be negative, an outflow cannot be positive. Any flow can be zero.

- By default all flows will be optional, i.e. they may be filled in or left empty (default to zero).
- It will be possible to set a flow to be mandatory, but this functionality will be de-emphasised. It will only enforce manual entry, which could still be a zero.

A flow will comprise the following:

- *Company/Department*: the entity that the flow belongs to. [strongest argument yet for use of a JSON Document store - should these elements be stored simply as an array of objects below each company - would simplify model].
- *Name*: description of flow
- *Direction Flag*: expects positive or negative?
- *Mandatory Flag*: must always be entered? Default: no.
- *Order*: position relative to other flows. This is because the users may wish to give priority to certain Inflows or Outflows in the display.

4.5. Budgets

The user will always experience a budget as a series of flows and values for a specific period. However, in the data structure a budget is always a set of flows and values for a specific day. These figures, once set, cannot be changed. In the database these figures will exist unconfirmed. Once confirmed they will be *frozen*. It is only once they are frozen that they can be viewed by anyone except the Operator who is entering them. The purpose of this storage-then-freezing mechanism is to allow the Operator to stop and start during the entry process without losing data.

Although, for the user, the entire set of days in the budget are frozen after entry, in the system Budget freezing is done per day. That means it freezes the budget for each day entered. Then if period is adjusted later the budget is still frozen but can constitute a different period.

When budget is frozen by user we must ensure that the operation is Atomic for the whole set of days which they have budgeted for. In other words there must be no chance of partial completion of the freezing of a budget from a user's perspective.

A budget comprises:

- *Company/Department*: who the budget is for
- *Date*: indicates the specific day the budget is for
- *Frozen*: flag indicating if budget is confirmed. Default: no.
- *FrozenDateTimeStamp*: precise timestamp when budget was frozen
- *Flows*: associative array of Flow IDs and values

4.6. Dailies

Dailies are the actual figures input every day. Once entered in the DB these figures cannot be modified.

A daily comprises:

- *Company/Department*: who the daily is for
- *Date*: indicates the specific day the figures are for
- *Flows*: associative array of Flow IDs and values

4.7. Balances

We don't want to have to recalculate opening balances going back to dawn of time - every time we load. So we save a balance every day when the dailies are entered. Then we can load it for whatever view we need starting after that day.

There are two types of opening balances:

- *Budget Balances*: calculated and stored every time a budget is frozen
- *Actual Balances*: calculated and stored every time a daily is entered

A balance comprises:

- *Company/Department*: who the balance is for
- *Type*: actual or budget
- *Date*: the specific day the balance is for
- *Balance*: the value

In theory an opening balance for any day is yesterday. In practise, of course, there are inactive days (weekends etc), so the opening balance must be the last balance recorded.

The system must ensure that a balance can never fail to be recorded for any day on which a budget or a daily has been entered.

4.8. Notes

A note is simply a block of text that belongs to a specific Manager and relates to a specific day and a specific Company. This allows Managers to attach notes to a specific day's figures. We don't tie it down to the actual or budget figures as observations will probably apply to both.

A note comprises:

- *Manager*: who made the note
- *Company/Department*: which entity it relates to
- *Date*: the day it relates to
- *Text*: the content of the note

4.9. Messages / Notifications

The system will (initially) send the following different kinds of notifications / messages:

- Warnings to Manager about missed budgets deadlines
- Warnings to Manager about missed daily deadlines
- Forgotten Password reminders
- User Invitations (see Registration section)

The system will store a queue of messages to be sent. Once a message is sent (by a scheduled task) it will be marked as such. After 60 days these messages will be removed.

A message will comprise:

- *Message Media*: Email, SMS etc. Initially all messages will be emails.
- *Message Subject*: Plain text summary of Message (i.e. email subject)
- *Message Content*: Actual content of Message
- *Message Content-Type*: The encoding of the Message (like text/html etc)
- *Recipients*: List of user ids
- *Sent Flag*: whether sent or unsent. Default: unsent.

- *Date Sent*: datetimestamp of when message was sent

4.10. System Settings

The system as a whole will have a variety of configurable factors. These may include things like:

- System Level Notification Email Addresses
- Content Distribution Network URIs
- API Endpoint URIs
- DB Endpoint URIs
- Maximum, Minimum and Default values for a variety of features
- Schedules for periodic tasks

This data will be stored in the Key-Value store.

4.11. Rules

Rules are criteria for determining when to provide visual highlighting or sending notifications. These are company / department specific.

These will be stored in Key-Value store. *In the first phase these rules will not be manageable via any user interface. However, they will be designed in a such a way that future versions will be able to implement such an interface.*

[Format for rules not yet precisely determined].

4.12. Localisation Data

Locale (Language/Region) specific data will be stored in the Key-Value store. This data is likely to be referenced a great deal. Note: on the frontend it will probably make sense to load localisation data in JSON at beginning of each session, rather than consulting it at every turn.

[Format of locale data not yet precisely determined.]

4.13. Audit Trail

We will log normal user events within the system to provide an Audit Trail. So for example when someone makes an entry of some data this will be recorded alongside their username. This is good for security. In future versions it may also provide the basis for allowing Administrators to track who did what, when.

5. Interface

The site design will be based on a Bootstrap template, defined by Twitter engineers and which has become a de-facto standard for modern website design.

The web application will have a single public-facing page, describing the product and providing contact information. It will also provide a login interface. There will also be a link to Terms.

Once the user is logged in, there will be differing navigations for different types of users.

The interface descriptions below refer to the Desktop based view. Mobile views will differ automatically. More work will need to be done to define those layouts.

5.1. User Registration

There will be no open registration process for users in the first phase.

There will be an invitation based Registration system. The Super-Administrator will be able to send invitations to Administrator level users by email. These will contain a unique link, allowing the invited person to signup providing their basic identity and contact information and to set a password.

Once this account is configured, the Administrator will themselves be able to issue invitations to lower-level users and they will sign-up in a similar way.

The User Registration interface will comprise:

- *Email*: Read only value. This will simply confirm to the User that they have been invited with the correct email address.
- *First Name*
- *Surname*
- *Address*
- *City*
- *Country*
- *Phone*

- *Password*: a strong password will be enforced
- *Join Button*

5.2. Login

The login interface will comprise:

- *Email*
- *Password*
- *Remember Me*: checkbox
- *Login*
- *Forgotten Password*: link

There will be validation for email address. If the *Remember Me* checkbox is checked then the system will apply an appropriate cookie on the user's browser.

The *Forgotten Password* link will bring the user to an interface comprising:

- *Email*
- *Reset Password*: button

If they input a valid email then they will be sent an email allowing them to reset their password. Password reset will enforce a strong password.

5.3. Operator Interface

The Operator Interface is where reporting units (Accountants, Financial Controllers etc...) will set budgets and input dailies.

The main navigation elements will be:

- *Company Selection*: this allows the operator to move between the companies they have responsibility for.
- *Period Selection*: this will default to the current period. The user will be able to select and view earlier periods. [Should we allow them to view historical periods?]
- *Profile*: allows the user to view and/or edit their contact data. They will be able to edit their email address.

- *Logout*: allows the user to explicitly logout. If they choose this option then they will have to manually login next time, regardless of previously set cookies etc.

Since, in theory, an Operator may work on one or more Companies, there will be a large clear title bar indicating the Company currently in view.

5.3.1. Company View

The following information will be clearly indicated and will remain in view when the Operator changes tabs:

- *Period*: the month and year of the period being viewed
- *Balances*: the opening balance for Budget and Actual for the period
- *Credit Limit*: the overdraft limit for the Company

Beneath the navigation and information display will be a tabbed view, one tab for each week within the period and an additional monthly totals tab. The number of tabs will vary obviously. Each tab will be titled Week1, Week2 etc.

On selecting a particular weekly tab (see monthly tab below) a series of columns will be displayed:

- *Flows*: Inflow and Outflows grouped and ordered (order is set by Manager).
- *Days*: two columns for each day in the week, one for budget figures and one for actual. Each column will be titled with the day of the week and date.
- *Weekly Totals*: two columns showing budget totals and actual totals

For each Day there will be the following calculated values (budget and actual):

- *Inflow Subtotal*: sum of inflows
- *Outflow Subtotal*: sum of outflows
- *Net Inflow/Outflow*: Inflow Subtotal less Outflow Subtotal
- *Opening Balance*: For Day 1 of the period this is the carried forward opening balance. Every other Day it is the Closing Balance of the previous *active* day.
- *Closing Balance*: It is the Opening Balance added to the Net Inflow/(Outflow).

For the Weekly Totals there will be the following calculated values (budget and actual):

- *Inflow Totals*: a sum of each inflow for each day

- *Outflow Totals*: a sum of each outflow for each day
- *Weekly Inflow Subtotal*: a sum of the total inflows across the week
- *Weekly Outflow Subtotal*: a sum of the total outflows across the week
- *Weekly Net Inflow/Outflow*: Weekly Inflow Subtotal less Weekly Outflow Subtotal
- *Opening Balance*: the balance of the first day in the week period
- *Closing Balance*: It is the Opening Balance added to the Net Inflow/(Outflow).

In a separate display there will be two additional tables of calculated values:

- *Cumulative Cash Inflow*: showing for each day the Inflows, budget and actual next to each other. Calculated by accumulating each Daily Inflow Subtotal.
- *Cumulative Cash Outflow*: showing for each day the Outflows, budget and actual next to each other. Calculated by accumulating each Daily Outflow Subtotal.

If the Operator selects the Period Totals Tab they will see a similar view which summarises the totals for each Week within the period and also totals them.

- *Flows*: Inflow and Outflows grouped and ordered (order is set by Manager).
- *Weeks*: two columns for each week in the month, one for budget figures and one for actual. Each column will be titled with the number of the week. The current Month will display above this.
- *Monthly Totals*: two columns showing budget totals and actual totals

There will also be a separate display showing the Cumulative totals (as above). All weekly figures have already been calculated in previous views and therefore we simply need to redisplay them in this view.

The calculated Monthly totals (budget and actual) are:

- *Inflow Totals*: a sum of each inflow for each week
- *Outflow Totals*: a sum of each outflow for each week
- *Monthly Inflow Subtotal*: a sum of the total inflows across the month
- *Monthly Outflow Subtotal*: a sum of the total outflows across the month
- *Monthly Net Inflow/Outflow*: Monthly Inflow Subtotal less Monthly Outflow Subtotal
- *Opening Balance*: the balance of the first day in the Month period
- *Closing Balance*: It is the Opening Balance added to the Monthly Net Inflow/(Outflow).

5.3.2. Operator Workflow

In a sense the Workflow is more significant than the display. The workflow determines how the Operator uses the software. The workflow must be strongly enforced by the interface. It is difficult to summarise all aspects of the workflow (without prototyping) but these are the key points:

- The Operator will be sent an email on the 1st day of the period reminding them of the requirement to enter a budget
- When the Operator logs in on the 1st day of the period, the system will prompt them to immediately begin entering their budget
- If the Operator has not entered a budget by a preset time during the 1st day of the period, then they will receive a reminder
- Until budgeting is done, the Operator cannot enter any daily figures for the period
- When budget is being confirmed, there will be a double-confirmation procedure as figures can never be changed
- Every other day the Operator logs in the system will prompt them to enter the daily figures. Confirmation will be required before figures are committed as these also cannot be changed.
- As soon as the budget is set a notification will be sent to the Manager
- As soon as the daily figures are entered an email will be sent to the Manager

5.4. Viewer Interface

The Viewer interface is simply a read-only version of the Operator interface. It allows a thrid person to have oversight of the same data as the Manager, without the power to make/view notes. The Viewer does not receive notifications.

5.5. Manager Interface

The Manager interface allows the Manager to view budgets and daily figures for all Companies / Departments.

The main navigation elements will be:

- *Company Selection*: this allows the Manager to move between the companies / departments they have responsibility for.
- *Period Selection*: this will default to the current period. The user will be able to select and view earlier periods.
- *Profile*: allows the user to view and/or edit their contact data. They cannot change email.
- *Logout*: allows the user to explicitly logout. If they choose this option then they will have to manually login next time, regardless of previously set cookies etc.

Beneath this they will have a read-only version of the Company View (as per section XXX). In addition they will have a *Notes* option. This will open a floating text entry box allowing them to make and save notes relating to a specific day's figures (budget and actual).

5.6. Administrative Interface

The Administrator interface allows the Administrator to:

- Create and modify Users
- Create and modify Companies / Departments
- Manage Period Settings for Companies / Departments
- Manage other settings for Companies / Departments

The main navigation elements will be:

- *Users*: Menu heading. Displays list of all Users.
- *Invite User*: Displays New User interface
- *Companies / Departments*: Menu heading. Displays list of all Companies / Departments
- *Create Company / Department*: Displays New Company / Department interface
- *Profile*: allows the user to view and/or edit their contact data.
- *Logout*: allows the user to explicitly logout. If they choose this option then they will have to manually login next time, regardless of previously set cookies etc.

Often the Administrator and Manager users will be the same person. When this is the case their navigation options will be integrated together.

5.6.1. User Administration

5.6.1.1. User List Interface

A tabulated list of Users, comprising:

- *Email*
- *Name*
- *Last Login Date*
- *View Link*

The list will be searchable and sortable. On clicking the *View* link the User will be brought to the View User interface.

5.6.1.2. New User Interface

A form comprising the following fields:

- *Email*
- *Invite button*

The idea behind this system is make the new User do their own work entering their data. This saves the Administrator time.

When the Invite button is pressed an email will be sent to the specified email address. The email will contain a unique and self-expiring link (24 hours). When the User clicks the link they will be brought to the User Registration interface (see above).

If the User has already signed-up with the system (in any organisation whatsoever) then their invitation will not bring them to the User Registration interface. Instead they will be invited to Login and *Accept* this new invitation. Once they have done so the Administrator will see them added to his User list.

5.6.1.3. View User Interface

This interface will provide the following features:

- *User Information*: the User's full information (see above). This will include a list of all Companies / Departments for which they are Operator. Also a list of all Companies / Departments for which they are Viewer.
- *Remove User*: User is removed from the Administrator's control panel, if they are not currently the Operator assigned to any Company / Department. Otherwise they cannot be removed. In reality the User is not deleted from the system, just hidden.

The Administrator will not be able to modify the User's email or contact information. Each user manages this for themselves. Again the philosophy is to minimise any data entry done by the Administrator.

5.6.2. Company / Department Administration

5.6.2.1. Company / Department List

This interface will show a tabulated list of Companies / Departments showing:

- *Company / Department Name*
- *Company / Department Address*
- *Company / Department Balance*: the latest available actual balance
- *Company / Department Operator*: the person assigned to enter Budgets etc
- *Company / Department Manager*: the person assigned to Manage the Operator
- *Company / Department Credit*: the overdraft / credit limit for the Company
- *View Link*

The list will be searchable and sortable. On clicking the *View* link the User will be brought to the View Company interface.

5.6.2.2. View Company / Department Interface

This interface will provide the following features:

- *Company / Department Information*: the Company / Department's full profile. This data will be modifiable in place.
- *Users*: this will allow the Administrator to assign the Operator and Manager for the Company / Department from a list. It will also allow assignment of Viewer users.

- *Inflows*: this will allow the Administrator to add and remove Inflows.
- *Outflows*: this will allow the Administrator to add and remove Outflows.
- *Remove Company / Department*: if the Company / Department has no current Operator and no data attached to it, then it will be removed from the Administrator's interface. In reality it will just be hidden from view.

Notes on Flows:

- Removed Flows that have no data attached to them in the system will be completely removed
- Removed Flows that have no data in the currently open period will be hidden immediately
- Removed Flows that have data in the currently open period will be flagged as to be hidden from the start of the next period
- Added Flows will be shown immediately for the current period. But if a budget has already been set then all their budget values will be zero. [needs discussion].
- Flow names will not be editable. An error will simply have to be deleted immediately and re-added with a new name. This will save us from a great deal of backend work and also prevent retroactive confusion where a renamed item may not map correctly to its historical identity.

5.7. Super-Administrator Interface

This interface is designed to manage the system as a whole. Although it will act as a template for many future features, it will have a very limited set in the first phase:

- *View Users*: This will show a list of all Administrator users (i.e. customers)
- *View Companies / Departments*: This will show a list of all Companies / Departments for each Administrator User
- *Add New User*: This will allow the Super Administrator to issue an invitation to a new User. The process will be similar to that described above in both User Registration and New User Interface.
- *Disable User*: This will allow the Super Administrator to freeze a User and all their Companies / Departments. This will not prevent the User logging in, or

even their Operators and Managers from logging in. However, it will prevent data entry.

6. Development

6.1. Timeline

Project development is likely to take 40 working days. During that time the key developer will require 10 days parental leave. Assuming a start date of 5th January we could aim for completion of initial build on 13th March.

I would then suggest a likely 2 to 3 week process, for testing and feedback. At the end of this process, depending on the level of the issues raised, there would be another period of bug-fixing and adjustment.

6.2. Costs

Although the overall complexity of the project was greater than initially expected, there appear to be fewer working days required on the design side than estimated (based on discussions with Designer). Therefore, overall, I have been able to maintain the original price.

Item	Price
Database design and implementation.	1,000
Data API Development. RESTful HTTP interface. The API is the core data and business logic layer for the system. It provides a universal interface for any system that wants to communicate with the data. This is key for long term flexibility and modularity.	4,500
HTML5/Javascript Web App development. Development of a Web App sitting on top of the Data API. The Web App will provide access to the	12,500

cash flow data across web browsers and most mobile devices. It will be accessed via SSL and secure authentication.	
Notification System. Development of a notification system to send emails based on agreed criteria (such as budget's not being met etc).	950
Sub-total:	18,950
VAT (@23%):	4,358.50
Total:	23,308.50

6.3. Terms

After sign off of specification:

40% Payment in advance
30% Payment on completion of initial build
30% Payment on satisfactory completion of testing/feedback