

Content Management System (CMS)

Functional Design Specification (FDS)

1 Version Control

This section documents the changes made to the FDS document as well as version according to Author Name, Version, Descriptive document name and Signature.

INTERNAL SIGN-OFF					
Person	Name	Date	Document Description	Version	Signature
Author	Kris Mortensen	June 26	1 st draft	0.7	
Reviewed by	Alan Levin	August	Final draft	1.0	
Reviewed by	Chris Higgo	July	1 st draft	0.7	
Reviewed by	Fatima Boltman	July	1 st draft		
Reviewed by	William Sinclair	July	1 st draft		

2 Approval

This section documents the list of those that have approved the FDS according to Designation, Name, Approval Date and Signature.

Designation	Name	Approval Date	Signature
Project Sponsor (KEEG)	Alan Levin	Aug 2002	
Team Leader (KEEG)	Chris Higgo	Aug 2002	
Content Manager (KEEG)	Katherine de Tolly	Aug 2002	
MIS Manager (IT)	Wynand Vivier	n/a	
Project Manager (IT)	Fatima Boltman	July 2002	
Analyst/Developer (IT)	William Sinclair	July 2002	

Table of Contents

1	Version Control	1
2	Approval	1
	Table of Contents	2
3	Scope.....	4
3.1	Introduction	4
3.2	Inputs.....	4
3.3	CMS functionality	4
3.3.1	Capture and edit (content submission)	4
3.3.2	Authentication	4
3.3.3	Workflow	5
3.3.4	Administration of users.....	5
3.3.5	Tools – (available to specific levels of users)	5
3.3.6	General.....	5
3.4	Specification Deliverables	5
3.5	Timeline and resources.....	6
4	Glossary & Acronyms.....	6
5	Business Drivers.....	6
5.1	Policy	6
5.2	E-Government strategy	6
5.3	Content Policy and Guidelines	6
5.4	Roles and responsibilities	7
5.4.1	Owner - Portal Task Team (PTT).....	7
5.4.2	Lead supplier - KEEG	7
5.4.3	IT Services	7
5.5	Return on investment	7
6	Logical Data Models.....	8
6.1	Classes.....	8
6.2	Associations.....	8
6.3	Attributes.....	8
6.4	Packages.....	8
7	Functional Models	9
8	Functionality	11
8.1	General Principles.....	11
8.2	Language.....	12
8.3	User Profiles.....	12
8.3.1	Web Author.....	12
8.3.2	Custodian (Data Steward).....	12

8.3.3	Access control	13
8.4	CMS Menu.....	13
8.5	Content life cycle.....	13
8.5.1	Register content.....	14
8.5.2	Edit content.....	15
8.5.3	Re-assign editor.....	15
8.5.4	Request comment	15
8.5.5	Request authorisation.....	15
8.5.6	Authorise content.....	16
8.5.7	Promote content	16
8.5.8	Delete content	16
8.5.9	Suspend content	16
8.6	Bundled Content.....	16
8.7	Standard Content Functionality.....	17
8.7.1	Content Item fields.....	18
8.7.2	Taxonomy (Content Categorisation)	19
8.7.3	Information Documents (Info Docs).....	19
8.7.4	Comments.....	19
8.7.5	Content Version Log	19
8.7.6	Actions	19
8.8	File Upload	20
8.9	CMS Job Queues	20
8.9.1	Web Author queue.....	21
8.9.2	Custodian queue	21
8.9.3	Job Queue display	21
8.10	Change Control	21
8.11	Complex Selects	22
8.12	Reports.....	22
8.12.1	Tender Adverts.....	22
8.12.2	Repository content.....	25
8.12.3	CMS Usage.....	25
8.12.4	CMS User Tree	26
8.12.5	Multiple Concurrent Logins.....	26
8.12.6	Portal Usage	26
8.13	Text Formatting	26
9	Implementation and Operation	27
9.1	Help.....	27
9.2	Data Capture	27

9.3	Service level requirements	28
9.3.1	Accessibility	28
9.3.2	Uptime.....	28
9.3.3	Scheduled downtime	28
9.3.4	Unscheduled downtime	28
9.3.5	Noticeboard	28

3 Scope

3.1 Introduction

The Content Management System (CMS) is to be developed in order to facilitate the capturing of information into the Cape Gateway Data Repository as well as the approval of that content for publication on the Cape Gateway Portal. The call centre, web users and those that service the public at the Cape Gateway walk-in facility, will then access the data in this repository in order to retrieve government information.

The CMS must be simple to use, yet also allow for flexible workflows in the process of editing and approving content before it may be made publicly accessible.

The CMS development forms stage two of the Cape Gateway development project. It will be started immediately following the completion of the data model (stage 1) and precedes the development of the portal (stage 3).

It should be kept in mind that this system will have future development iterations as the requirements grow and the e-government strategy progresses. The current plan is to enhance the CMS every 12-18 months. Future versions of the portal will include more complex transactional capabilities.

3.2 Inputs

The data model reflects a broad range of structured information. It is the primary input although others - including but not restricted to all inputs used in the data modelling project - must be used and any gaps identified, investigated and included.

Other documents that provide valuable inputs into the CMS specification include: Portal Sections, the current website, user profiles, public web sites audit, department content outlines (PTT documents), Cape Gateway business plan and other government web sites.

3.3 CMS functionality

The Content management system must include the following functionality:

3.3.1 Capture and edit (content submission)

The elements of the data model that will be used must be defined and the database components that follow must be specified. Capture of data into each of these components must be specified.

3.3.2 Authentication

Each user must be authenticated via simple username and password. There will be various privileges associated with different users; the levels and details of these must all be defined.

3.3.3 Workflow

The rules for workflow will be different for each user, and each piece of content. These will be governed through a policy (described in chapter 5.3) distinct and separate from the system. As such the system must allow for different workflows for each user for each package of data. For example: one department may want any user to be able to automatically publish, others may want a more stringent process that involves checks by other users. Some may want no checks on editing but a full approval process on new additions.

3.3.4 Administration of users

Set up users, etc

3.3.5 Tools – (available to specific levels of users)

For example:

- a. edit, delete and expire content
- b. help (access to Content Policy, writing style guidelines, system help)
- c. text formatting tool (bold, hyperlink, etc)
- d. search for content (probably necessary for hyperlinking)
- e. create new content category/attribute/etc.
- f. assign a piece of content a value (e.g. today's highlighted service)
- g. attach comments to content for review
- h. view the audit trail of a particular piece of content
- i. Messaging & Notification – out of the workflow will come the need to message users of tasks they need to attend to (e.g. rewrite content, edit content, moderate content, etc.).
- j. Reporting – It is essential that blockages in the workflows are identified and reported. Various other aspects of reporting must be identified.
- k. Statistics – about:
 - content in repository
 - usage of CMS (must be logged to ensure an audit trail and accurate reporting)

3.3.6 General

The implementation of the above functionality must be balanced with usability. Ease of use must be implemented in the design, in order to encourage uptake of the product. For example, the flows (or route through screens) to input a particular content type must be kept to a minimum. The particular tasks that users will typically carry out on the system must be achievable quickly and easily. In this the design needs to balance flexibility and simplicity.

3.4 Specification Deliverables

This specification is purely a functional design specification and is not meant to be a detailed technology specification. It takes the form of a group of documents that cover the CMS functionality and processes, as well as the information content of each CMS user screen. These are developed in consultation with the project sponsor, Alan Levin. The final specification passes through a process of at least two iterations.

3.5 Timeline and resources

The entire specification process required a total of eight weeks. The project sponsor – Alan Levin, expert consultant – Kris Mortenson, Systems Analyst – William Sinclair, IT project manager – Fatima Boltman, Content Manager – Katherine de Tolly and Team Leader – Chris Higgo were resources involved in the development of this specification.

4 Glossary & Acronyms

Please refer to the data model for the definition and description of any content item.

CMS – Content Management System

Content Item – any piece of content that is maintained in the CMS

FDS – Functional Design Specification

Government Structure Component (GSC) - any organisational unit within government

5 Business Drivers

5.1 Policy

The White Paper “Preparing the Western Cape for the Knowledge economy of the 21st century” is the initiating business driver the Cape Gateway project. This paper identifies that

“in the knowledge economy, information is in many ways the key resource. The introduction of world-class systems for the collection, analysis, management and dissemination of information is widely seen as an indispensable precondition for the development and implementation of effective and competitive strategies for regional economic growth and development”

5.2 E-Government strategy

The need for a provincial government portal was initially recognised in the first quarter of 2000. A Business Plan¹ was developed and agreed in November 2000. This was further enhanced through a detailed process of establishing a broad e-government strategy: the Cape Online programme². Further business justifications and strategic details defining the background to this project may be identified through these documents.

5.3 Content Policy and Guidelines

The CMS is designed to be highly flexible in order to meet the demands of not only all the departments within Provincial Government, but also other government and external stakeholders. This need for flexibility means that relatively few rules are to be built into the CMS. The Content Policy and Guidelines (separate document) is thus an essential guide for behaviour on the CMS.

¹ Cape Gateway Business Plan – 2000 – available on request

² Cape Online strategy – August 2001 – available on request

5.4 Roles and responsibilities

5.4.1 Owner - Portal Task Team (PTT)

The ultimate ownership of Cape Gateway and the Content Management System resides with the Portal Task Team. This committee was established by a resolution made by the provincial top management in November 2001. The members of the PTT include representatives from each department as appointed by the heads of each department. In addition the PTT includes invited stakeholders and other e-champions as co-opted by the PTT from time to time.

5.4.2 Lead supplier - KEEG

The branch Knowledge Economy and E-Government (KEEG) currently resides in the Department of Economic Development and Tourism. KEEG is responsible for developing and implementing the provincial e-government strategy. The Cape Gateway project is its flagship project and currently of highest priority. KEEG includes new provincial competencies in the areas of usability and content management.

5.4.3 IT Services

The IT services branch currently resides in the Department of Provincial Administration. The branch is responsible for providing ICT services to the organisation. Cape Gateway ICT infrastructure and development are provided by IT services.

5.5 Return on investment

As per all core projects in the Cape Online e-government strategy, implementation of the Cape Gateway information products, including the CMS, will result in much improved business usage of ICTs. This will in turn result in greater efficiencies in all aspects of work and most importantly a significant increase in customer service levels. This will have a significant positive impact on all the cabinet objectives.

6 Logical Data Models

Two data models have been developed. The first is the logical data model of the content itself and the second is data model of the logical tables required to manage the CMS. Both models are developed and maintained in Rational Rose – file Data_Model_vn.n.mdl. An HTML version of the models is available on the web (<http://capeonline.org/datamodel>). The following aspects are covered by the logical data model:

6.1 Classes

These represent the information or content items within government on which the portal will provide information, e.g. legislation (acts, green / white papers) government departments, services, facilities, projects, commissions. Also referred to as objects or entities. A class is represented on the model as a box with a name in it. It is described by a Definition, Description and Examples.

6.2 Associations

Classes rarely exist in isolation. It is their relationship to other classes that is of interest and that needs to be recorded in the model, e.g. a standing committee monitors a particular ministry and has a number of government officials as members. Associations are represented on the model as lines between the classes, sometimes with a label to clarify the nature of the relationship where it may not be apparent.

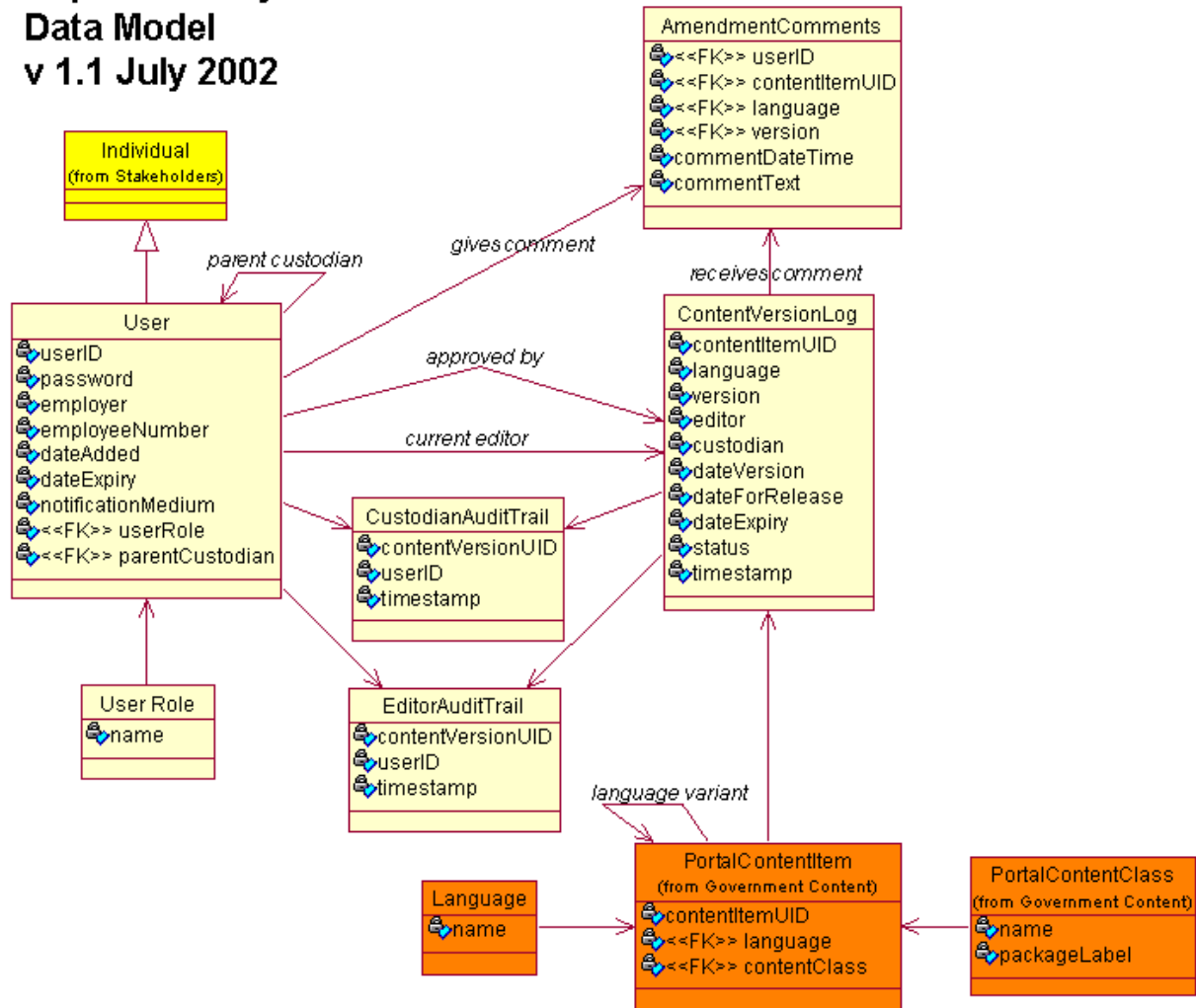
6.3 Attributes

Attributes are the detailed information that describe an instance of a class and distinguish it from other instances, e.g. the class Government Employee has the attributes: name, persal number, title, and telephone number(s). Attributes are listed in the middle section of the class.

6.4 Packages

A package is used to group classes that are representative of a common area. The aim is to assist in the simplification and understanding of the subject domain, e.g. government structure, which encompasses departments, Parliament, commissions, posts and employees. On the model packages are indicated by colour coding the constituent classes, again to enhance the readability of the model. The package to which a class belongs is written in brackets underneath the class name. In a higher level model, a package is represented as a tabbed rectangle.

Cape Gateway CMS Data Model v 1.1 July 2002

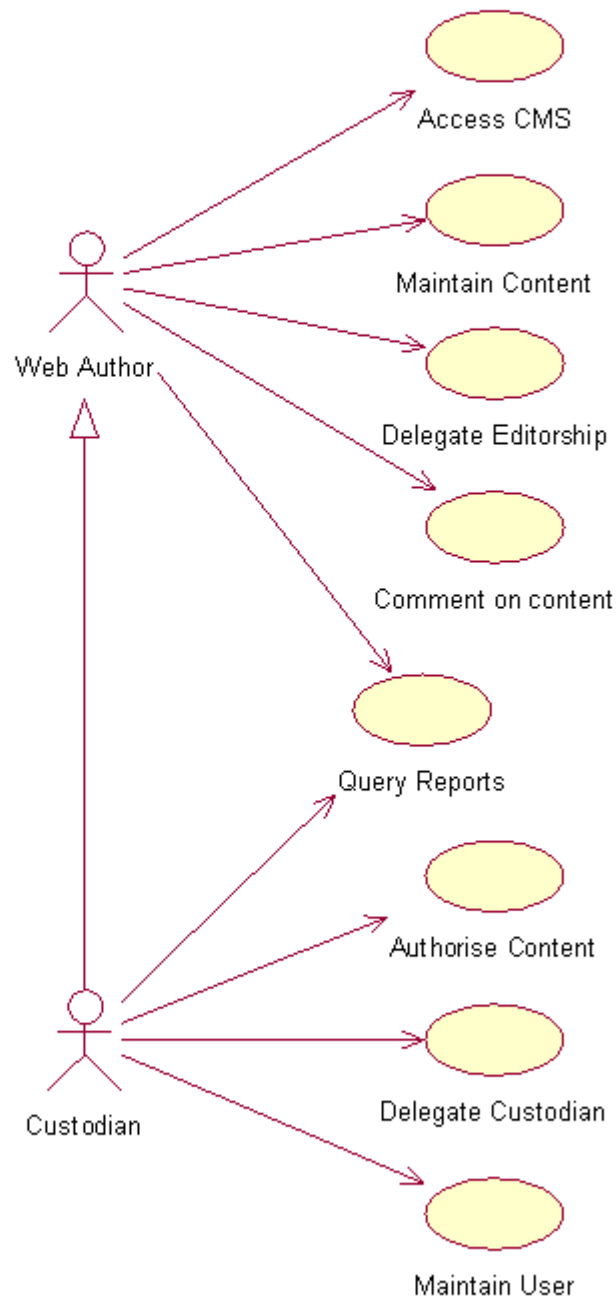


7 Functional Models

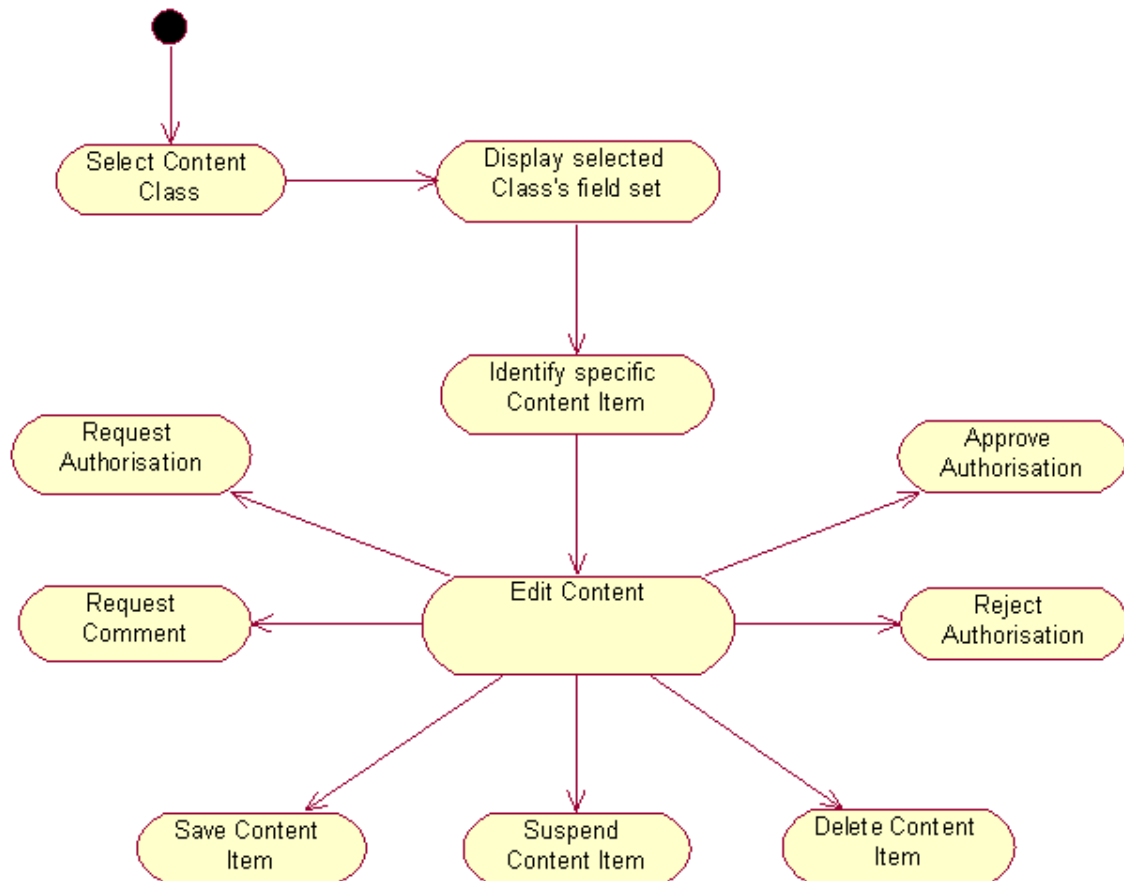
The functional models are also documented in Rational Rose in the same model (.mdl) as the data model (above) under the package CMS Behaviour Model. The following use case diagrams are extracted from there for overview purposes. A detailed functional decomposition can be found in User Requirements Specification.

Content Management System

Use Case - version 1.2



Maintain Content - version 1.0



8 Functionality

8.1 General Principles

Any user may add or modify content items. However all additions or changes must be authorised by a custodian. Hence the semantic integrity of the content is maintained by the custodians, guided by policy, as opposed to a complex, automated system of access control and content jurisdictions.

There must be a common look and feel for similar types of content across the CMS, e.g. all publications, whether legislative or informational must be maintained in the same way, with similar looking screens.

8.2 Language

The language requirements of the Portal are that a user can specify a language of preference and a second language. The portal will then provide content in the language of preference, if it exists. If a content item does not exist in the preferred language, then a version of the content item in the second language will be shown. If this in turn does not exist, then the third language version will be shown.

The practicality of managing the content to cater for this portal requirement has the following implications:

The language must be specified for every content item added or modified in the CMS; i.e. the main content item editing screen must have language as mandatory field for every content item, regardless of class.

All component content items of a bundled content item must be in the same language of the main content item; e.g. if a Post is entered in Xhosa, then the GSC that is to be selected must be in Xhosa. If the GSC does not exist in Xhosa then it must be added.

The contentItemUID of PortalContentItem must be "language neutral"; i.e. the unique key is not dependent on language. This means that every language variant of a content item will have the same contentItemUID.

The CMS must recognise when a content item is a language variant of an existing content item so that it may be given the same contentItemUID. This is easier said than done as some key identifying attributes have language dependent component attributes, such as name, which will make automatic matching impossible. It is therefore necessary to implement policies that will assist the web authors in ensuring they understand the concept of language variants and enter content items accordingly. The CMS must provide a function that will enable users to indicate that a content item is a language variant of another by assigning the same contentItemUID to both content items. This may be referred to as language synchronisation. The content richness of the Portal will inevitably differ between languages.

8.3 User Profiles

There are essentially two user profiles, web authors and custodians.

8.3.1 Web Author

Anyone, including a custodian, who has been granted access to the CMS will have web author rights, these rights include:

- The ability to add or modify any content and request its promotion onto the Portal. These requests must then be approved by the appropriate custodian before any content, new or modified, will be made available on the portal
- The ability to select any custodian as seen fit to approve new or modified content
- The ability to solicit comments from a specified list of people regarding a content item he/she has edited
- The ability to comment on any content that is open for comment or pending authorisation

8.3.2 Custodian (Data Steward)

A custodian is a CMS user with the following authorities and responsibilities:

- The responsibility to assess requests for approval of content to be published on the Portal and the authority to approve or reject such content
- The authority to create new CMS users and subsequently modify their user details as required and to allocate them the appropriate role – web author or custodian. A custodian may

maintain the details of users further down the “hierarchy”. A custodian may grant custodian rights to anyone else, but only within his/her jurisdiction (hierarchy branch)

- Has all the capabilities of a web author

The custodian that approves a content item is recorded against the relevant ContentVersionLog for audit trail and other purposes (see Request Authorisation).

There is an implied hierarchy of custodians and web authors created by recording the custodian who adds a user to the system as the “parent” custodian of that user. The custodian, or other parent further up the hierarchical branch, may change the parent custodian for a user at any point in time. The uses of the hierarchy are as follows:

- To assist web authors when they are adding content and want to submit the content for approval by defaulting in their immediate parent custodian. This applies only to adding content as the custodian who previously approved the content will be the default custodian for content modification. Note these are only a default as the web author may choose another (more appropriate) custodian to approve the content
- A web author may wish to “escalate” the approval request further up the custodian chain, for example if the original custodian is not available
- A custodian is responsible for the content he/she approves. However there is an implied responsibility up the custodian chain for having granted custodian rights to the said custodian.

8.3.3 Access control

Users must have access to the CMS via the internet and not just through the intranet or WAN.

Duplicate logins will not be allowed. However this is to be controlled through reporting the transgression and subsequent discipline as opposed to up front blocking of access (which would preclude possible identification of transgressors).

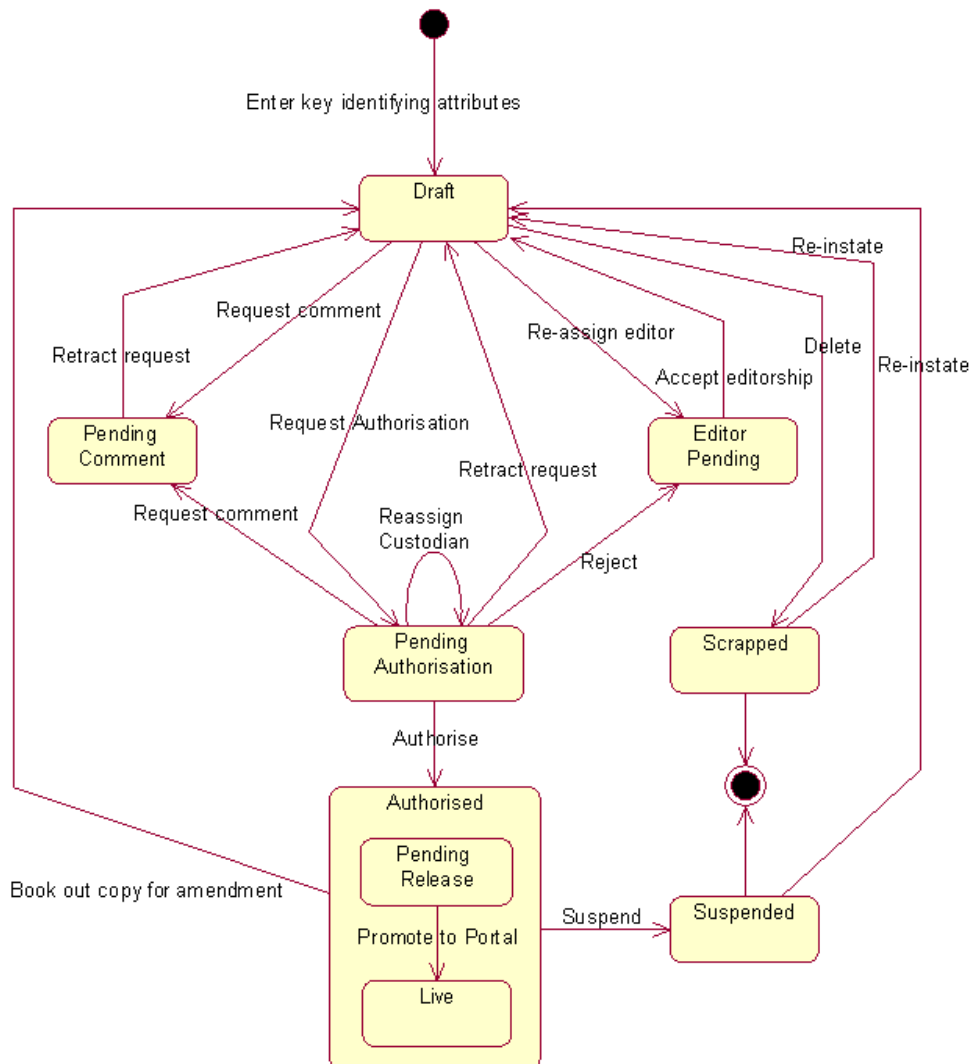
8.4 CMS Menu

The CMS menu provides a logical navigation path to the functions of the CMS. See User Requirements Specification for detail.

8.5 Content life cycle

All content added to or modified on the CMS will go through a standard life cycle: see the Portal Content Life Cycle state chart.

Portal Content Life Cycle State Chart - version 1.2



8.5.1 Register content

The process starts with a valid user selecting the appropriate content item maintenance screen, via the CMS menu. The user will then be required to enter the key identifying attributes for that type of content (see Field Specifications). At this time it will be determined if this is new or existing content. All existing content items will have a record in the PortalContentItem class and new content items will be inserted with a unique identifier (UID). The versioning process will provide a new version of the content item for modification. For a new content item the fields will be blank (except for the key ones already entered); otherwise they will be populated with the content of the previous version. The remaining process is then the same for modify or add.

Any web author may edit existing content. However, if the content item is currently being edited (i.e. in a state other than *authorised*, *live*, *suspended* or *scrapped*) then only the current editor may edit it. A custodian that is further up the custodian hierarchy may access a content item that

is in any state and re-allocate editorship, except if the content item is locked by another editor who is currently editing it. An error message should indicate this.

The latest version of a content item will always be brought up.

8.5.2 Edit content

Once the content item is made available to the user for update, it is deemed to be in a *draft* state and the user has the role of editor. Only one person (user) may have editorship of a content item at any point in time. At any point the user may explicitly **save** the content on the screen. **Save** will simply store the content on the database, without changing the content status. The user may exit the modify function at any point in time, in which case a prompt will ask the user if the content must be saved, if there are any unsaved changes. Save will be performed automatically when the user selects an option to progress the content item through its life cycle (i.e. Request Comment, Request Authorisation or transfer editorship). **Validation** of content item fields (attributes) will take place whenever the content item is expected to change status (i.e. Request Comment or Request Authorisation)

8.5.3 Re-assign editor

A user may re-assign editorship of a content item to another user at which time the state will change to *editor pending*. The content item will change back to a draft state once it is opened for modification (accept editorship) by either the proposed editor or the original editor; i.e. whichever user opens the content item for modification (from their job queue) accepts editorship.

8.5.4 Request comment

An editor may request that other users give input to or comment on a content item. The selected commentators will then be explicitly requested to give input to the proposed content. Any other valid users may also give comment at this stage. It will, however, have to be of their own initiative, facilitated by the browse function of all content submitted for comment (suggest that this is constrained by the government structure component of the user to reduce the size of the potential list). Editing of a content item will not be allowed while it is in this *awaiting comment* state, as it will be difficult to comment on something that may be changing at the same time. The way this will be managed is that the content item will return to a *draft* state when it is opened for modification by the editor, and thereby no longer be *awaiting comment*. A commentator should not be affected, or stopped from commenting when an editor withdraws a content item from an *awaiting comment* state. An editor may do this in order to add or remove commentators. Note: when the editor removes or adds a commentator after the original request for comment, the other commentators should not be affected – i.e. the content item will remain in their job queue, unless they have already commented, in which case it should not re-appear in their queue.

If editorship is required to be re-assigned while awaiting comment the editor will return the content item to a *draft* state by opening it for modification and then re-assigning editorship. The new editor may then request comment once he/she accepts editorship (open for modification). Once a commentator has entered a comment and pressed save, then the content item should no longer appear on his/her job queue, even if the editor (web author) leaves it in a *pending comment* state for longer.

8.5.5 Request authorisation

When the editor is happy that a content item is ready to be published on the portal, the content item is sent by the editor to a custodian for authorisation. The content is now in a *pending authorisation* state. If the editor is not a custodian and it is a new content item, then the custodian immediately above the web author in the user hierarchy will be assigned by default. In the case of a modified content item the custodian who authorised the previous version will be the default custodian for this version. The web author may however select a different custodian from the default.

In the case of the editor being a custodian the functionality to approve a content item must be streamlined in order to avoid inefficient promotion of content. Hence, if a custodian wishes to approve a content item (within his/her jurisdiction), this should be able to be done without having to request authorisation, which he/she will approve anyway. In effect if the content item custodian is the same as the user (user-ID) and the status is draft then the approve action may be pressed and the *pending authorisation* state will be suppressed and set to *authorised*. Unless, of course, another custodian is selected, then authorisation must be requested. The editorship role of a content item transfers from the requesting editor to the allocated custodian when it is in a *pending authorisation* state.

8.5.6 Authorise content

Accept: A piece of content will enter an *authorised* state once approved by a custodian. Once authorised, that version of the content item may no longer be edited. Any subsequent modification of the content item will result in a new version being created in the *draft* state.

Reject (Amendment required): A custodian may suggest or require further amendments to the content, in which case the custodian will fill in necessary comments and reject the request. The content item will then move to an *editor pending* state and editorship will return to the web author that requested the authorisation, unless the custodian explicitly allocates a different web author for editorship. Note that a content item can only be deleted from a *draft* state – see delete content.

8.5.7 Promote content

Once the content item is approved (authorised) it is assumed, from a logical perspective, to be live on the portal. The physical process of making the content available may not be immediate due to practical reasons or there may be a release date (in the future) specified for the content item. Once the physical process of moving the content item onto the live portal is complete, the content item is set to a *live* state. It is assumed that the promotion to the portal will take place at regular intervals, the duration of which, while dependent on physical / technical considerations, should be no longer than hourly and more frequently if feasible. Should the promotion not be immediate due to practical considerations, there must be a facility to effect changes immediately on the Portal (e.g. in the case of urgent withdrawal of content).

8.5.8 Delete content

A content item that is being edited (i.e. not *live*, *authorised*, *suspended* or *scrapped*) may be deleted; but only by the current editor. The content item will then be in a *scrapped* state. When a content item is pending authorisation and the custodian, who is a different person to the requesting web author, wishes to delete the content item, the custodian must **reject** the item with the appropriate comments for the web author (requesting editor) to take action.

8.5.9 Suspend content

A custodian may at any point withdraw content from the portal. The suspend function will be used to do this and the affected content will then have a *suspended* state. Suspended content may be reinstated, which is done by opening it for modification, whereby it will have a *draft* state and follow the standard process.

8.6 Bundled Content

When content is added or modified it may be necessary to add associated content items for completeness' sake. Where the addition or modification of a content item involves one or more associated content items, all the associated content items will be "bundled" with the main content item for the purposes of content editing, commenting and authorisation; i.e. bundled content must be authorised as a whole and will only be promoted to the portal when all the

component content items are authorised. For example: when adding a Job Advertisement a Post must be specified. The user may select an existing Post or add one if the required one does not already exist. In the latter case, both the Job Ad and the new Post must be authorised by a custodian. If, for example, the new Post was not approved, then the Job Ad will also be rejected.

When a content item (the primary content item) has one or more associated content item with it in a content bundle, these secondary content items may not have a further set of associated content items (tertiary content items) included in the bundle. For example: When adding a Post and a Government Structure Component (GSC) is to be selected, the editor may opt to add a new GSC. While adding the GSC the option to add Official Contacts to the new GSC may not be taken. This avoids the problem of the bundled content becoming too complex and cumbersome, particularly when having to approve, version control or language synchronise.

8.7 Standard Content Functionality

This is functionality that must be available on the main content editing screen, regardless of the class of the content item. Exceptions and deviations will be specified where appropriate. The main content editing screen will be opened in 2 modes:

By an editor (web author or custodian) in order to update any of the fields. This is done when a content item is opened for modification by a user that is equal to `ContentVersionLog.editor` or `ContentVersionLog.custodian`.

By a commentator to review the content item and make comments. In this mode the user may only update the comment text to add comments. All the other fields will be read only and the only available action will be save and exit.

Main Content Editing screen

Logical Spec. Version 0.3

Actions				
Save	Preview	Review Audit Trail	Request Comment	Approve
Delete	Print		Request Authorisation	Reject

Content Version Log				
Release Date	Editor	Commentators	Version	Date Last Modified
Expiry Date	Custodian		Status	

Category - Taxonomy	Information Document	Comments
---------------------	----------------------	----------

Content Item fields	
Key Identifying Attributes	
Language	search
Content Body	

8.7.1 Content Item fields

This section represents the actual fields which are specific to a content item based on its Portal Content Class. Refer to Appendix A – Field Specifications – for the detailed specifications.

Key Identifying attributes – represents the fields required to Register Content. See Content life cycle above. Common to all content items is the Language field, and as such it resides on the PortalContentItem class. This must be entered for every content item.

The search function will help the user to find the relevant content item. This will be sensitive to the particular class of the content item.

Content Body – this is where the bulk of the fields (attributes) of a content item are maintained. The specific fields and their business rules are specified in the Field Specifications

8.7.2 Taxonomy (Content Categorisation)

This option allows the content item that is currently open for modification to be categorised for portal navigational purposes. It will allow the currently available taxonomy items (content categories) to be browsed and selected (linked to), thus categorising the content accordingly. The browse functionality will start by displaying the top level taxonomy item in each available taxonomy hierarchy and allow drill down into each hierarchy (as per a folder/file structure). The appropriate category(ies) (taxonomy item) may then be selected and linked to the content item. The alternate screen from which content items may be categorised is the Taxonomy Item content item edit screen itself (see Field Specifications).

8.7.3 Information Documents (Info Docs)

Info docs are a specialisation of Publication that facilitates any publication to be associated to any other content item in the CMS. This enables any amount of additional information, including images or any other format of publication, to be accessed in conjunction with a given content item. This option on the standard add or modify screen for all content items will allow all publications to be browsed and the selected publication(s) to be linked to the current content item. The alternate screen from which content items may have Info Docs linked to them is in the Information Document content item edit screen itself (see Field Specifications).

8.7.4 Comments

This section can be updated by anyone, regardless of the state of the content item – given that the user already has the content item open for modification. This may be while editing the content item, reviewing it for authorisation or commenting on it as requested by the editor. Each time a comment is made by a user, a record is kept of the user, date/time and comment text in the AmendmentComments class. Furthermore the window must show all the previous comments made by anyone for the content item, showing the user, date/time and comment text – much like a “chat room” or discussion thread.

8.7.5 Content Version Log

Content Version log is the mechanism by which the editing and versioning of content is controlled; see Field Specifications - ContentVersionLog for the rules for each individual field. Release and expiry dates will determine when the content item will be available on the portal and when it should be removed. The user will be able to specify the editor and the custodian as required and when requiring comment to be made on the content item, the relevant web authors will be selected in the Commentators field. There will be ‘complex select’ functionality availability on these 3 fields that will help navigate the potentially numerous users to be chosen from. The status, version and date last modified fields are automatically updated by the system and are read only.

8.7.6 Actions

Save will store the content as it stands on the screen into the database. The user may explicitly Save at any point in time, allowing the content item to be stored for later editing. Save will also be automatically invoked with the following actions: Request Comment, Request Authorisation, Approve and Reject. Other actions will be implied when a save is done, such as: Assign a new editor will happen when a different editor is selected and the content item Saved. **Save** will update the timestamp on ContentVersionLog, even if the status has not changed.

Delete can be pressed when the user = ContentVersionLog.editor and ContentVersionLog.Status = draft. Delete will then change the status to *Scrapped* whereby the system will remove all records relating to that **version** of the content item. (How this is physically done is to be specified in the physical/technical specification.)

Preview will show the user what the content item will look like on the portal.

Print will print the current content item, regardless of its status.

Review Audit Trail will show a list of the previous editors and custodians, from the classes EditorAuditTrail and CustodianAuditTrail.

Request Comment can only be selected by the editor (web author or custodian) and only after one or more web authors have been selected in the commentators field. The status will then be set to *Awaiting Comment* and the content item will then appear in the relevant web authors' job queues. Selecting this action will automatically save the content as described above, but the status will only be changed once all the fields are entered correctly as specified in the Field Specifications.

Request Authorisation can only be selected by the editor (web author or custodian) and only once all the fields are entered correctly as specified in the Field Specifications. The status will then be set to Pending Authorisation and the content item will show in the relevant custodian's job queue for authorisation. It will also remain in the requesting editor's job queue.

Approve can only be selected by a custodian and when the status is *Pending Authorisation* or *Draft*. If in a Draft state a Save and Validation of all the fields will be performed before updating the status. Allowing custodians to Approve a content item in a *Draft* state saves them having to request authorisation when they have the authority anyway. However a custodian may wish to have another custodian approve the content item, in which case the Request Authorisation action must be selected.

Reject can only be selected by a custodian and when the status is *Pending Authorisation*. The custodian must enter a comment in the comment field before selecting **reject**, not only for the benefit of the requesting editor, but also for any other web authors that may review the content item at a later stage.

Retract Request does not have an explicit action option on the main editing screen as it is done implicitly from the CMS job queue screen. This happens when a requesting editor opens for modification a content item that is in an *Authorisation Pending* or *Awaiting Comment* state. Once open for modification the content item will revert to a *Draft* state.

8.8 File Upload

There must be functionality that will allow files (e.g. images) to be loaded onto the CMS (and hence be internet accessible). All upload files must be defined in the CMS as publications, therefore the upload function will then bring up the add publication screen and default the upload option into the content component section of publication. For uploaded documents the name should have no spaces, and the name length should be restricted. The upload functionality must cater for content which is uploaded, but which may be rejected by the custodian and therefore will need to be removed from the file storage system.

8.9 CMS Job Queues

CMS users will (usually) be required to review or authorise content that has been created or modified by other web authors. Likewise, they will request others to review or authorise their content editings. As the number of content items to be reviewed could be numerous, it is required that the user may view these in a list or queue, known as a Job Queue and then be able

to take the appropriate action for selected content items. The entries that appear in a user's job queue will be determined by the user's profile (web author or custodian). Therefore there are logically two queues, the Web Author queue and the Custodian queue.

8.9.1 Web Author queue

The job queue for a web author will list content items that:

- the web author is currently modifying: where status = draft and editor = user-id
- the web author has requested comment on from other web others. This row must also show the number of commentators requested, commented and awaiting comment.
- other web authors have requested the web author to comment on
- the web author has requested authorisation on
- require the web author to accept editorship on; either due to the previous editor wishing to pass the editorship on or because an authorisation request has been rejected (note: this may be for a request that the web author requested or one that a different web author requested, as the custodian has the liberty to change the editor at the time of rejecting the request)
- the web author has requested another web author to take on editorship – i.e. editorship pending

Note that content items that have had an authorisation request approved will no longer appear on the web author's job queue; therefore the web author may presume that a request has been approved if it no longer appears in the job queue. (This could always be checked by browsing the portal)

8.9.2 Custodian queue

A custodian is automatically a web author and therefore will have a job queue with all the content items mentioned above as well as content items that the custodian must review for approval.

8.9.3 Job Queue display

- The columns to be shown on the Job Queue include:
- Date and time of when the content item was last modified
- Status of the content item
- the editor and/or the custodian of the content item
- the Portal Content Class of the content item
- the display label of the content item (as specified in Field Specifications)

The user must be able to open any of the content items displayed for modification.

8.10 Change Control

The process of adding and modifying content is covered in the [Content life cycle](#). The CMS must allow for changes to be reviewed against the original content (current portal content). It is foreseen that the portal will be used for this. (see also Preview under actions in the Standard Content Functionality section).

8.11 Complex Selects

The Field Specifications (Appendix A) often refer to the screen object “complex select”. This is required when a user should select an already existing content item, but where the possible list to choose from could be long. The complex select is intended to make it simpler and more intuitive to select an appropriate option. This is facilitated by allowing the user to constrain on one or more appropriate attributes (which in turn may be related content classes). The field spec for each class will specify the fields by which the user may constrain the select list. For example the list of possible locations may be too long to scroll through in a reasonable time, therefore it would be useful if the user could constrain the list be the location category, e.g. ‘magisterial districts’.

8.12 Reports

This section specifies the reporting requirements identified to date. The visual design and layout of the reports will be defined in the User Requirement Specification of the CMS.

8.12.1 Tender Adverts

This section details the reports that the CMS should output so that PAWC tenders can be published online. Capturing the tenders on the CMS serves two purposes:

- It replaces the current capturing of tender ads in MSWord (or other applications)
- The ads are published via the CMS on the Portal.

All output files are to be in HTML format, including the table format, as in the current tender ads.

Report 1: Tender ads to be published in the national bulletin

This report allows the user to generate an electronic version of the ad(s) required. The address list report (Report 2) for the selected tender ads should be produced with this report.

Screen 1: How ads are selected:

Tickbox: “Ads I’ve entered” – ie ads that the user entered themselves. AND/OR

Date range: “Date ad(s) captured”. Use: contentVersionLog.timestamp. Default=from a week ago to today. AND/OR

Complex select: “Owner (eg department)” – ie select the GSC that owns the tender ad. AND/OR

Select: “Tender category”

Screen 2: results of select

Show all tender ads that fit the criteria as selected above. Sort them ascending by estimatedValue; then by Tender Category; then descending by contentVersionLog.timestamp. The output should show the different value bands in different sections. The following fields should appear on screen:

- Tickbox (to select whether you want the ad output to file)
- EstimatedValue
- Tender Category
- Description
- RequiredAt
- TenderNumber
- AdvertDate
- SiteObtainableFrom

- SiteDeliverTo

There should be a defined way to 'Output to file' to be specified in the User Requirements Specification.

Report 2: Address

This report allows the user to generate an electronic version of the necessary address info pertaining to Annexure 1 of the National Bulletin.

Screen 1: How addresses are selected:

Tickbox: "Show all addresses" OR

Select: "Site number" OR

Complex select: "Owner (eg department)" – ie select the GSC that owns the procurementSite.

Screen 2: results of select

Show all addresses that fit the criteria as selected above. Sort them ascending by Site number if the user selected Tickbox: "Show all addresses" or the complex select for GSC. The following fields should appear on screen:

- Tickbox (to select whether you want the address output to file)
- siteNumber
- Name of Department
- PhysicalAddress
- PostalAddress
- TenderBoxAddress
- EnquiryAppointment
- TelephoneNumber
- FaxNumber
- OfficeHours

There should be a defined way to 'Output to file' to be specified in the User Requirements Specification.

Report 3: Results of tender invitations

This report allows the user to generate an electronic version of the results, incl brand, price etc.

Screen 1: How tender results are selected:

Show status changed to 'awarded' (user doesn't need to select this) AND

Date range: "Date tender status changed to 'awarded'". Use: contentVersionLog.timestamp? default=from a week ago to today. AND/OR

Complex Select: "Tender category" AND/OR

Complex Select: GSC

Screen 2: results of select

Show all tender results that fit the criteria as selected above. Sort them ascending by contentVersionLog.timestamp? if that criterion was used, or by Tender Category is that criterion was used. (If both were used, sort ascending by contentVersionLog.timestamp then Tender Category.) The following fields should appear on screen:

- Tickbox (to select whether you want the address output to file)

- tenderCategory
- parentCategory
- tenderNumber
- Item No
- SuccessfulTenderer
- Price
- Brand
- BasisOfDelivery
- HDIpercentage

There should be a button at the bottom of the screen 'Output to file'

Output file:layout: Tender_Results_username_today'sdate.doc. The file should be laid out the same as the current tender results (see B.RESULTS OF TENDER INVITATIONS in any Government Tender Bulletin).

Report 4: Tenders finalised

This report allows the user to generate an electronic version of the tender invitations finalised.

Screen 1: How tender results are selected:

Show only status changed to 'finalised' (user doesn't need to select this) AND

Date range: "Date tender status changed to 'finalised'". Use: contentVersionLog.timestamp? default=from a week ago to today. AND/OR

Complex Select: "Tender category" AND/OR

Complex select: GSC

Screen 2: results of select

Show all tender results that fit the criteria as selected above. Sort them ascending by contentVersionLog.timestamp? if that criterion was used, or by Tender Category is that criterion was used. (If both were used, sort ascending by contentVersionLog.timestamp then Tender Category.) The following fields should appear on screen:

- Tickbox (to select whether you want the address output to file)
- parentCategory
- tenderNumber

There should be a button at the bottom of the screen 'Output to file'

Output file layout: Tenders_Finalised_username_today's date.doc. The file should be laid out the same as the current ads (see C.TENDER INVITATIONS FINALISED in any Government Tender Bulletin).

Report 5: Tenders cancelled

This report allows the user to generate an electronic version of the tender invitations that have been cancelled.

Screen 1: How cancelled tenders are selected:

Show status changed to 'cancelled' (user doesn't need to select this) AND

Date range: "Date tender status changed to 'cancelled'". Use: contentVersionLog.timestamp? default=from a week ago to today. AND/OR

Complex Select: "Tender category"

Screen 2: results of select

Show all tender results that fit the criteria as selected above. Sort them ascending by contentVersionLog.timestamp? The following fields should appear on screen:

- Tickbox (to select whether you want the address output to file)
- tenderNumber

There should be a button at the bottom of the screen 'Output to file'

Output file layout: Tenders_Cancelled_username_today's date.doc. The file should be laid out the same as the current ads (see C.TENDER INVITATIONS CANCELLED in any Government Tender Bulletin).

8.12.2 Repository content

This report provides an overview of the content in the repository. The primary class is ContentVersionLog.

Selection Parameters:

- Portal Content Class – default "All"
- Status– default "All"
- Date Range (timestamp) – default **to** today **from** one week back

Display Fields:

- Class, status, timestamp
- Key identifying attributes and Display label
- Editor and Custodian
- Release date and version

The result set must be sortable by any column. Any row (online) may be selected to preview the content item (double click?)

8.12.3 CMS Usage

This report allows great flexibility in viewing users and their usage of the CMS. The primary classes are User and ContentVersionLog. All current users are eligible; i.e. dateExpiry greater than current date.

Selection Parameters:

- User.dateAdded range – default all
- User.parent Custodian – default to current user's own custodian
- User.userRole – default to "all"
- ContentVersionLog.status – default to "all"
- Count of content items (greater than given count) – default to zero (i.e. all content items)
- GSC – default to blank (if this is selected it implies government employees only, other it will show all users including non-government organisations).

Display Fields:

- Name, User-Id, Parent Custodian
- Employer – if government, then show post and GSC
- Date added

- ContentVersionLog.status and the count of items in that status
- Notification Medium
- The date the last content item was modified

The result set must be sortable by any column. Any row (online) may be selected to drill to either the user's job queue or the user's full details.

8.12.4 CMS User Tree

This report will produce the whole user hierarchy. Primary class is User.

This will enable users to see where they fit in, in terms of custodians etc. Display the name, user role, and (if the user is a government employee) the post and GSC. Each new level (child) should be indented to indicate the hierarchy.

8.12.5 Multiple Concurrent Logins

This report will highlight the occurrences of any user-id attempting to sign-on more than once at a time – indicating than more than one person may have knowledge of the user-id and password, as well as access to the CMS.

8.12.6 Portal Usage

The portal usage statistics are seen to be part of the portal and will be specified in the Portal Architecture phase.

8.13 Text Formatting

The content that needs to be entered into the CMS as text will be done so through a standard text box. The initial implementation is expected to be a simple text entry using basic HTML tags and not a third party tool.

The text box must have help functionality that includes a description of the valid HTML tags, style guide and content policy. The following HTML tags are the recommended set to be displayed in the help:

Text formatting Help

Text within the CMS can be formatted by including HTML tags in the text.

The following tags may be useful:

HTML Tag	Sample HTML	Sample Result
Bold	one two three	one two three
Italic	one <i>two</i> three	one <i>two</i> three
Bold & Italic	one <i>two</i> three	one <i>two</i> three

Line Break	one two three	one two three
Paragraph Break	one two<p>three	one two three

A HTML tag Wizard or Toolbar may be considered to assist the users

The following are allowed, but not preferred, therefore omit from the help screen:

Embedded URL	one two three	one two three
Embedded E-mail address	one two three	one two three
Underline	one <u>two</u> three	one <u>two</u> three

Images should not be allowed – Information Source should be used for this.

Ordered and Unordered lists should be allowed, as well as line items (bullet points)

Allowing embedded URL introduces the risk of content being accessible through the portal that is not managed by the CMS and its associated content authorisation process. However this risk is currently perceived to be minimal and is outweighed by the benefits.

9 Implementation and Operation

9.1 Help

The integration of all system help and instructions are to be specified in the usability audit that follows this first phase specification. The design of the help section is to be developed in the Usability – CMS visual design sub projects. The content of the help will be determined by the Content –Trained web authors and custodians sub project.

9.2 Data Capture

It is currently expected that all data will be captured manually through the CMS, with one exception. We have been informed that the Data Warehouse contains locations base data that will need to be imported or integrated (contact Jasper Stupart).

9.3 Service level requirements

9.3.1 Accessibility

The CMS will need to be accessible via all WCPG client workstations as well as via the Internet using the standards as defined by the DPSA "Handbook on information inter-operability standards" policy (i.e. W3C browser compliant or otherwise).

9.3.2 Uptime

The CMS (database and application servers) must be available 99.5%x24x7x365.

9.3.3 Scheduled downtime

All scheduled downtime must be communicated to all users at least 48 hours in advance.

9.3.4 Unscheduled downtime

A contact person and associated 24x7x365 contact details must be provided in order for any custodian to report any unscheduled downtime.

9.3.5 Noticeboard

A web noticeboard must be established to report/indicate any downtime periods and the causes of the downtime.