



# DRAFT TECHNICAL DESIGN DOCUMENT

---

## OVERVIEW OF THE ACTORS, SYSTEMS, FEATURES, FUNCTIONS AND DEVELOPMENT STRATEGIES FOR GELC 2.0

Version	Date	Author	Description
.1	5/30/2006	Joshua Marks	Draft - outline level
.2	6/1/2006	Joshua Marks	Draft outline with Bobbi notes + more
.3	6/7/2006	Joshua Marks	Began architecture section
.4	6/10/2006	Joshua Marks	Module descriptions added
.5	6/13/2006	Joshua Marks	Module descriptions updated + Shared Services descriptions
.6	6/22/2006	Joshua Marks	Bobbi edits plus content model
.7	6/30/2006	Joshua Marks	Draft Roadmap and Non-functional requirements added
.8	6/10/2006	Joshua Marks	Started draft of Phase 1 site design and map section added
.9	6/20/2006	Joshua Marks	Completed draft of Phase 1 site design and map section added

TABLE OF CONTENTS:

<b>OVERVIEW .....</b>	<b>3</b>
<b>DESIGN GUIDELINES AND CONSTRAINTS .....</b>	<b>3</b>
<b>USER CLASSES.....</b>	<b>4</b>
<b>DATA OBJECTS, CONTENT AND METADATA .....</b>	<b>5</b>
<b>USE CASE CLASSES.....</b>	<b>8</b>
<b>SYSTEMS, FUNCTIONS AND PROCESS MODELS .....</b>	<b>13</b>
<b>DESCRIPTION OF FUNCTIONAL MODULES .....</b>	<b>15</b>
USER AND GROUP ADMINISTRATION .....	16
COMMUNITY AND ORGANIZATION TOOLS .....	16
INSTRUCTIONAL DESIGN WIKI (TOOLS).....	16
TEXTBOOK DESIGN WIKI (TOOLS) .....	17
INSTRUCTIONAL MEDIA WIKI (TOOLS).....	17
CONTENT TRANSFORMATION AND RENDITIONING SERVICES .....	18
ASSESSMENT DEVELOPMENT TOOLS .....	18
ASSET REPOSITORY AND SEARCH SERVICES.....	18
LEARNING MANAGEMENT (LMS) AND ASSESSMENT ADMINISTRATION SERVICES .....	19
<b>DESCRIPTION OF SHARED COMPONENTS.....</b>	<b>20</b>
PORTAL SERVICES .....	20
SECURITY & AUTHENTICATION .....	20
RIGHTS AND ENTITLEMENTS .....	21
WEB SERVICERS INTERFACES AND SIF AGENTS .....	21
WORKFLOW AND BUSINESS PROCESS MANAGEMENT SERVICES .....	21
CONTENT AND CHANGE MANAGEMENT SERVICES.....	21
<b>FUNTIONAL AND NON-FUNTIONAL REQUIREMENTS.....</b>	<b>22</b>
<b>PROPOSED RELEASE PLAN AND TECNOLOGY ROADMAP.....</b>	<b>22</b>
PHASE 1- TRANSITION AND START-UP .....	23
PHASE 2- DEPLOY ENTERPRISE INTEGRATION FRAMEWORK .....	24
PHASE 3- EXTEND AND INTEGRATE NEW COMPONENTS .....	25
<b>PHASE ONE WEB SITE DESIGN AND MAP .....</b>	<b>25</b>
THE HIGH LEVEL SITE MAP AND FLOW .....	26
THE FRONT DOOR AND ENTRYWAY .....	27
THE MAIN HOME PAGE.....	30
GENERAL WIKI AND SITE EDITING INTERFACE RULES .....	32
GLOBAL SEARCH INTERFACE RULES .....	33
THE CONTRIBUTE CONTENT SECTION.....	34
THE MAIN USERS AND GROUPS AREA .....	37
THE COURSES AREA .....	41
SYSTEM ADMINISTRATION FUNCTIONS .....	47
<b>APPENDIX .....</b>	<b>47</b>
A. WIKI PLATFORM SCORE CARD .....	47
B. PORTAL FRAMEWORK TECHNOLOGY SCORE CARD .....	47
C. PARTNER AND COMPONENT TECHNOLOGY SCORE CARD .....	47
D. PHASE ONE DRAFT DEVELOPMENT SCHEDULE .....	47
<b>PARKING LOT.....</b>	<b>47</b>

---

## OVERVIEW

---

GELC seeks to build and operate a web delivered hosted service to enable the creation, discovery and use of Open Source Curriculum (OSC) by developers, educators and students of all ages, in all subject areas and languages, world wide. In the process GELC will foster and support highly engaged learning communities, both in terms of curriculum development and instructional delivery and management. This document will list and organize a set of use cases into logical use case and functional groupings. It will also outline the design constraints and guidelines and describe a service oriented architecture (SOA) suitable for meeting the requirements and use cases listed. This design document and its attachments are intended to be a “living document” that evolves and is to serve as an internal planning and project scoping document.

Part of the technology strategy is to utilize and integrate content, data, applications, systems and technologies from a wide range of sources and existing community projects. This requires a well considered systems integration strategy and technology roadmap both of which are also presented here. A component description section and “Scorecard” is included to map a number of identified systems and platform options to the proposed architecture at the feature and function level. Once a complete set of use cases and enabling technologies is selected for the first and each successive development effort and release cycle, these use cases will be extended and elaborated with detail as part of the planning and scheduling process, or ‘inception phase’.

Also included here, towards the end of the document, is a draft functional web site design for the first “Transition and Start-up” phase of development in four planned releases. This is a prospective design that will be greatly modified based on platform choices, user and stakeholder input and the practical limitations of time, budget and resources. However, this design should set the basic strategy and approach for delivering tools capable of meeting GELC’s important goals.

---

## DESIGN GUIDELINES AND CONSTRAINTS

---

As GELC is predicated on community development and an open source philosophy, the following are design and technology principles that will help direct the systems development and integration strategy:

- 1) Leverage open source tools and systems and existing projects where appropriate.
- 2) Create a consistent and simple user interface suitable to the needs of all users.
- 3) Integrate partner systems and content to the greatest degree possible.
- 4) Enable and motivate participation and collaboration across institutional and geographic boundaries through leveraging common needs and sharing assets, and creative and human resources.
- 5) Through the delivery of exceptionally capable and easy to use tools for content development and delivery, help create a globally accepted consistent materials form factor based on open standards and formats like DocBook, OpenDocument, PDF, MPEG/MP3, SCORM, IEEE LOM, IMS, Dublin Core and SIF.
- 6) Multi-lingual and accessible to the greatest degree possible
- 7) Modular, service oriented architecture for systems integration

- 8) Common or shared data model where feasible
- 9) Highly secure in terms of student and sensitive personal information
- 10) Support and enable the inclusion of commercial assets where appropriate for an open source community. Provide and enforce a spectrum of Creative Commons like and perhaps commercial licensing policies.
- 11) Support DRM for content as appropriate and necessary

---

## **USER CLASSES**

---

Users of the system are defined by what groups they join and what roles they assume in those groups. Which features and use cases apply to a user is described in terms of user classification or type. However, any individual user can simultaneously belong to many groups and in those groups have very different roles. There are five core user classes, each with potentially several and extensible sub-classes:

- 1) Administrators: Users with access to create and manage groups, organizational structures, user accounts, roles, rights, privileges, tasks, and to edit or remove approved/published content from the repository.
  - a) System admin – Administer at a system level with access to manage all users and content
  - b) Group admin – Administer at the group level, including approving/inviting new members of the group, assigning and changing roles and role privileges for that group, and manage task assignments for members of the group.
  - c) Project admin – Project administration is a function within a group for sharing the management of task assignment and collaboration tools and schedules (e.g. setting-up conference calls, assigning or editing tasks, setting policy like peer review and voting processes.)
- 2) Contributors: SMEs, Curriculum Developers, Designers, Editors, Writers and Reviewers – Those who create or contribute to content, as part of a Project Group. Any user, even visitors can contribute by adding notes or suggesting edits for review and approval, but only contributors with the appropriate role in the project group (Like Editor) can approve those suggested changes in the official publication (e.g. Drafts are open like a Wiki, but the master and all published renditions have an approval process.)
- 3) Teachers/Parents – Those who select, modify and/or assemble content into components or whole curricula into presentation collections (Publications) for their students, and utilize learning management functionality (Teachers themselves can be Students for professional development, Contributors as part of a Project Group, and Admins for their Student Groups). These users will be able to select complete curricula ready for delivery or assumed and or modify asset to build a custom curricular package for delivery.
- 4) Students – Those who use/consume content either individually (Self directed) or as part of a group lead by a teacher, as in a group course.
- 5) Visitors (Non-registered users)

- 6) Members (Registered users of type 1-4 above)

## DATA OBJECTS, CONTENT AND METADATA

Fundamental to any eLearning or repository system is the data model, which includes the content model, the metadata model and, in our case the user data model as well. Each of the data objects described below fit into one of these buckets, but also overlap and are interrelated. For example, part of the metadata model are curricular or 'Learning Objectives' frameworks which include the hierarchy of skills in a course and descriptions of specific teachable and measurable skills and knowledge. This "Instructional Metadata" relates to both content (e.g. 'this lesson covers this skill' or 'this assessment measures proficiency of this set of skills') and students and contributors (e.g. 'This student has demonstrated proficiency in this skill' or 'this contributor is a subject matter expert in remediation of this skill.') The content metadata model is based on the [IEEE LOM](#) and is compatible with it and other Metadata standards and controlled vocabularies such as [Dublin CORE](#) and [SIF](#). The content is treated in a hierarchical object oriented way, just as a textbook is made of units which in turn are made of chapters comprised of sections containing lessons, exercises, illustrations, activities, and tests with related curriculum guides, supplemental materials, worksheets, etc. The following diagram illustrated the relationship between assets, collections and curricula:

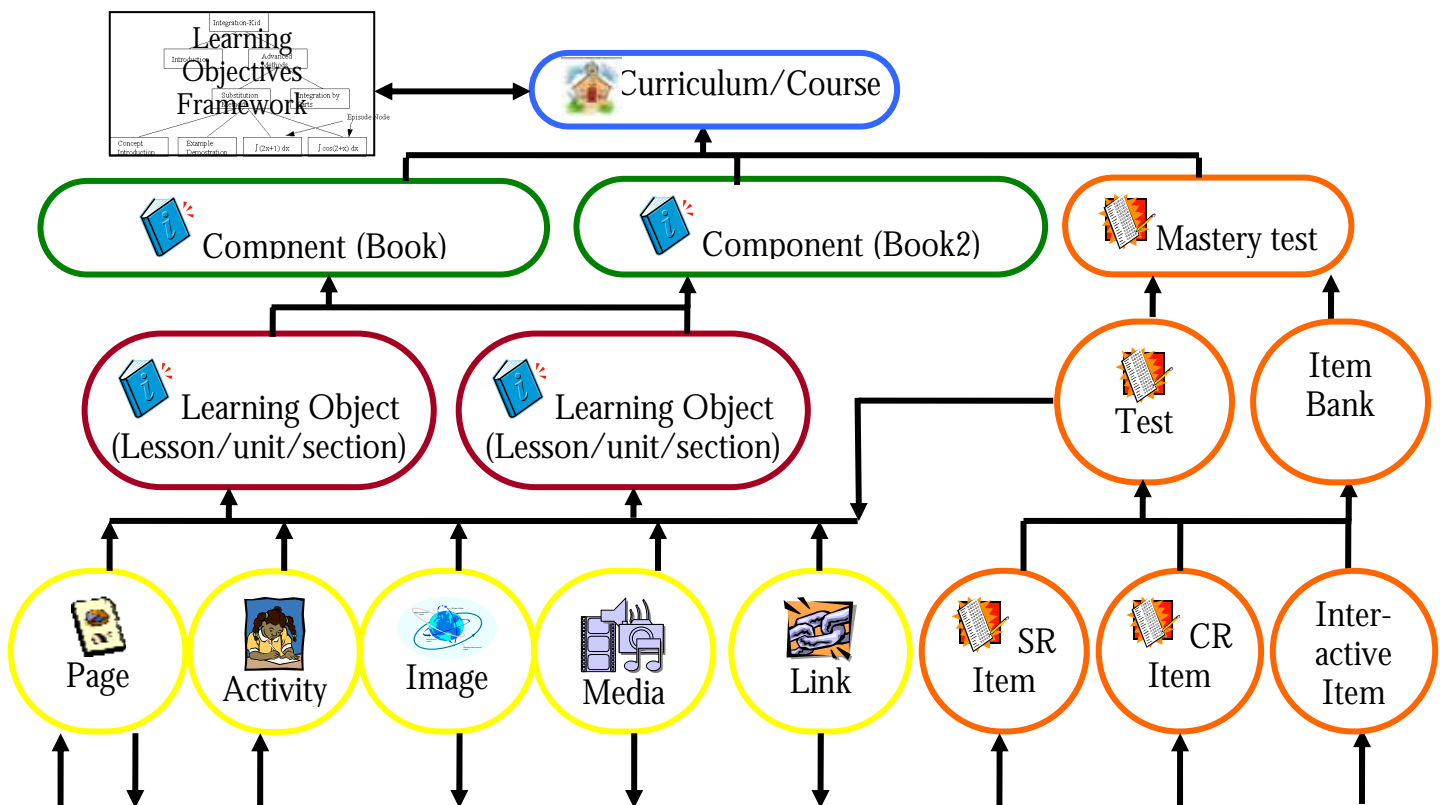


Figure 1- Conceptual Content Model

Each of the content objects in the diagram above has extensive metadata associated with it. For example each asset may be aligned to one or many curricular frameworks and skills. Each type of asset has a format, and may have several versions and exist in several languages. For example a “Page” might be a lesson in English as a web page, or part of a curriculum and instruction guide in Chinese as a PDF document. Similarly, a media asset might be a classroom lecture as a “Podcast”, or an instructional video in an MPEG format on how to score a specific test. Each asset also has a creator, contributors, creation & revision data, lifecycle state, and a license that it is made available under, all of which is considered related metadata.

There are many different data objects referenced in the use cases including (This is just a start and this will need a lot of refinement and eventually will be expressed as a logical data model):

- 1) Asset (Can be stored as files, hosted pages, external links or references to physical objects. All assets can be aligned directly to one or more curricular frameworks, associated with group organizations and individual contributors, have extensible metadata attached including lifecycle states and versions, and can be associated to parent/child assets and collections that contain the asset or a child/derivative of the asset. If assets are not directly aligned a framework, then they may be indirectly aligned by being included in a learning object that is.):
  - a) Learning Object Type (All have common metadata like creator, version, license, languages, lifecycle state, framework alignment, and many have specific metadata including: format(s), duration/time needed, pre-requisite skills, technical requirements, etc.). Learning objects are content packages of other assets.
    - i) Lesson/Lesson Plan
    - ii) Text Book
    - iii) Curriculum guide
    - iv) Course/program
    - v) Activity (Individual or Group)
    - vi) Assessment
    - vii) Exercise
    - viii) Reference/Resource
    - ix) Presentation/Lecture
    - x) Project
    - xi) ... (Game, Competition or Contest?)
  - b) Image/Media asset including videos, podcasts, illustrations, etc. (Potentially part of a learning object)
  - c) Interactive or Multi-media asset including Flash Animations and activities, Java Applet Simulations, 3-D models, and the like (Hosted or stand alone, potentially part of a Learning Object. One example may be an on-line multi-player activity.)

- d) Document asset (In a number of supported formats TBD.)
- e) Reference or Link (External) asset (Including external applications)
- f) Assessment assets (Including item bank and appropriate item metadata, like framework alignments.)
  - i) Item (With parameters and framework alignments.)
    - (1) Rubric
      - (a) Scoring Guide
      - (b) Response Evaluation Process (Processor)
    - ii) Form/test
      - (1) Section
      - (2) Answer Sheet
      - (3) Report/scale score
    - iii) Response/response set (The output of an assessment instance- not really an asset, more of an LMS or Workflow data element)
    - iv) Score/Assessment Report (Scale and raw- The processed output of a response set.)
    - v) Shared Stimulus (Perhaps reference to media asset)
- 2) Workflow and Process descriptions and associated assets. These process objects might be used for both the editorial/curriculum development processes, peer review processes, and for learning management processes and workflows (Or each might have its own set of processes objects.)
  - a) Task/Assignment to group, role or member (Including task direction, assets included and expected output with optional due date.)
    - i) Task Status /outcome (Query and Update)- Includes notes, responses, votes, edits, history etc.
  - b) Workflow process task sequence (The order of tasks for a workflow, like “Create first draft” or “Peer review”)
    - i) Conditional process branching (Rules)
- 3) Curriculum Framework (Course organizational structure and measurable outcomes)
  - a) Master or Global Curriculum Framework for a subject
  - b) Master subject taxonomy
  - c) Local (Group) Curricular Framework

- 4) Organizational/Group hierarchy (Group organizational structure)
  - a) Role within a group (And rights and privileges associated with the role in the group)
- 5) Event Log – List of user's interactions in time
- 6) Collaboration Data: Comments, Group Discussion Threads and Messages
- 7) User Record and ACL (Use data, Access Control List, access rights and privileges)
  - a) ID/Profile: Screen name, password, e-mail, contact info, language(s), image, etc.
  - b) Groups affiliations and roles in those groups
    - i) Tasks assigned and task history
      - (1) Assessment Result and Response Sets(?)
      - (2) Links to all contributions
  - c) GUID (To link to all work and contributions)
- 8) License or Policy

---

#### **USE CASE CLASSES**

---

Below is an organized outline of the use case classes (With some description added where needed)

- 1) Entry
  - a) Non-registered user (Visitor) navigation
  - b) Registered user (Member) navigation, including log-in (Enable Single Sign-on with Java.net, and a long term federation strategy with partner organizations.)
- 2) Register New User
  - a) Via a member invitation (To whole site or specific group(s))
  - b) Via random entry
  - c) Bulk load form organization
  - d) Express group affiliation and role within a group
    - i) By Invite form group admin
    - ii) By request to join group
    - iii) Define group specific roles, rights and access privileges for member



- e) Validate user type/class (Password reset, etc.)
  - f) Required user data
  - g) Optional user data (With consideration for COPPA and FERPA constraints)
- 3) Manage Users and Groups
- a) Administer user accounts, included banning bad actor.,
  - b) Member self edit profile (Change password, image, group affiliations)
  - c) Create Group
    - i) Define Role within A Group
    - ii) Define Privileges for a Role within a group
    - iii) Create/Delete/Edit a group workflow, process or policy
    - iv) Create an organizational hierarchy for a group (optional, more related to LMS and group reporting, but potentially useful for curriculum development organizations as well, particularly to support local editions of a national curriculum, for example), such as State -> District -> School -> Teacher -> Student
- 4) Create/Edit Asset/Object (Contributors only) –
- a) Create/edit in response to an assigned task,
  - b) Propose edit or new asset and send to group for approval (For all members and visitors via group submission workflow process)
  - c) Build content map/outline
    - i) Build object/asset specification and metadata (Based on curriculum framework)
    - ii) Align resource to curriculum framework
  - d) Start/end/edit development workflow or task instance
  - e) Assign task to contributor, role our group (Create, edit, review, etc.)
    - i) Cancel Task
    - ii) Re-assign task
  - f) Accept or reject completed task (As defied by workflow)
  - g) Create or add item to discussion about asset (Any member can post, visitors are reviewed before posting by board owner)

- h) Upload/import assets or Learning Object/Asset (Including all required and optional metadata)
  - i) Individual asset
  - ii) Bulk import SCO or Learning Object Content Package
  - iii) Bulk import entire collection (Like complete textbook; many special cases)
  - iv) Set appropriate license wrapper for asset
  - v) Align asset(s) to framework(s)
- i) Edit an asset (Create and publish derivative/modified asset)
- j) Create a derivative asset from an existing asset, with version control.
- k) Assessment objects special cases (e.g. Item creation tool and publishing to assessment delivery engine. Support IMS QTI as native structure for assessment assets.)
- l) “Link-out” or “External Object” special cases (e.g. send user to hosted assessment service that returns a score, or to an external LMS or web site where an asset is hosted. May require some form of sign-in passing, affiliate ID structure or single-sign-on functionality.)
  - i) As part of a partnership deal integrating two or more offerings
  - ii) As part of an individual member contribution (e.g. SME links to an externally hosted flash based simulation or application.)
- m) Choose and manager license wrappers for assets and collections
- 5) Framework Management- Build and edit master (Global) subject taxonomy and curricular framework (Extending a hierarchal structure from main topic down to measurable skill). The master framework will be used to create course and local curricular frameworks, and will be extended organically in this process.
  - a) Build, edit and extend master framework.
  - b) Import SIF LearningStandardDoc (And other framework and ontology structures like OWL, ODF, DublinCore, etc)
    - i) Import and merge arbitrary curriculum and/or Learning Object metadata structure and related Controlled Vocabularies into master taxonomy.
  - c) Export a framework as SIF LearningStandardDoc/LearningStandardItem XML structure (And other defined structures like OWL, ODF, DublinCore, etc).
  - d) Associate a PDF or document asset with a framework (As a reference to an official published document describing an adopted curricular framework, e.g. the Sunshine State Standards, or the NCTM standards, etc.)

- e) Individual or group create and/or compile a custom curriculum framework (e.g. Group course framework, book map, local learning standards, etc.).
    - i) Create new master framework node while building course framework.
    - ii) Define new term for master framework controlled vocabulary.
  - f) Align custom or imported framework with master framework.
- 6) Create/compile and edit collection (Curricula or course), with within licensing constraints.
- i) Select Course Curriculum Framework from Framework Management module (Assemble and extend form Master Curriculum Framework and reorganize, extend or revise.) can convert to content map/outline for new collection.
- b) Select Assets for collection (Scope) - Same as Search
- i) Use content map/outline to identify existing assets that are applicable to the collection form both the internal and external repositories.
    - (1) Select identified assets for inclusion in the collection/curricula.
  - ii) Identify new assets/objects to be created (Request for asset/object task from contributor group or groups)
- c) Create sequence for course (Sequence of presentation and assessment, including rule based branching logic/simple sequencing) – This is the LMS case for workflow
- d) Export collection
- i) As SCO (SCORM Content Package)
  - ii) As Book (DocBook or PDF)
  - iii) As assignment link to hosted LMS
  - iv) As a custom export object (TBD)
  - v) As a “Stand alone” course that can be run off-line
- 7) Search repository for content (By user class and role- Many different search modes)
- a) Member (See WIP- Work in progress, for restricted projects if role and group affiliation permits)
  - b) Visitor (Only sees published content and unrestricted WIP assets)
  - c) Web Service interface for external apps to search GELC repository
  - d) External search interfaces to query other repositories (ERIC, MELOT, NSDL, CNX, edumcommons, Flickr, etc.)

- 8) Search for member or community (Including TBD “Social networking” functionality)
- 9) Teacher manages students (Group LMS functionality)
  - a) Teacher Manages students (Class)
  - b) Assign task to student (Lesson, activity or assessment)
  - c) Student complete assignment (Track progress through course)
  - d) Reports (Many)
  - e) (Private/enhanced LMS capabilities for membership fee?)
- 10) Student driven interaction (Self directed study LMS functionality)
  - a) Student searches for asset, assessment our course
  - b) Student uses asset (“Checks-out” book, views content/course or takes an assessment) and the activity is tracked in student profile and transcripts.
  - c) View transcripts/student planner
    - i) Student (And/or teacher) gets score and progress reports, including prescriptive remediation assignments based on test taken and content consumed.
    - ii) Student is provided suggestions for materials to use next based on profile and transcripts. (Note: initially, for each topic, course or area of interest, students will be provided placement tests and introductory modules for courses.)
    - iii) Student sees open groups s/he might join based on interests and skills observed.
- 11) GELC and Group level policy discussions
  - a) Create policy process (Propose, comment, review, vote)
  - b) Define new licensing class and wrapper
  - c) Policy enforcement processes
- 12) All user activities
  - a) TBD, Mostly from above (e.g. Rate or comment on an asset, search, etc.)
- 13) Group Specific Activities
  - a) TBD (e.g. co-brand group area or create custom group page/area with its own look)
  - b) Create Organizational Hierarchy
  - c) Create roles and privileges

- d) Invite/manage group members
- e) Create group workflow processes
- f) Manage group tasks and assignment
- g) Group collaborative tools:
  - i) Discussion board (Read and comment) about group, section or individual asset.
  - ii) Document repository for group
  - iii) Group Whiteboard and Conferencing (Voice or Video)
  - iv) Calendar/Events
  - v) Messaging (e-mail, IM, etc.)
  - vi) Voting

---

#### **SYSTEMS, FUNCTIONS AND PROCESS MODELS**

---

To address the requirements as described in the use case classes listed here, common features must be identified and the core functional elements of the integrated system should be defined in a conceptual architecture. Considering the design guidelines and the general need to be flexible and extensible over time, a modular service oriented architecture (SOA) is proposed here. The intent is to add modules and features within modules in a set of iterative development or systems integration projects. What is described here is the basic framework and organization of functional modules. This framework is suitable for guiding this iterative design process for new development, or to develop integration strategies for existing applications and tools.

The functional modules will be described in terms of the use cases they support and their inputs, outputs and potential interactions with each other. Most use cases are primarily related to one functional module; like create content is mostly done in the Wiki modules. However, all modules will have interdependency with others, and in particular the community tools and shared services like authentication and discussion boards. This set of functions will then be aligned with the use cases and with the identified list of enabling technologies. For example, we will evaluate a number of Wiki platforms that might be selected and find that some can enforce role based conditional access, and others can't. Or some might be able to support math with acceptable formatting and others might not. We may find no single Wiki suitable for all our needs and have to extend one or build one from scratch. The main objective of the conceptual architecture is to guide the technology and development roadmap for the GELC infrastructure.

In the following diagram, users of different classes enter the system through the GELC portal, where security and authentication services are used to find the roles and groups associated with the user. These role and group affiliations are used by the rights and entitlements shared services to expose the appropriate modules, and functions within modules to the user. The functional modules (Together, the application layer), should utilize a shared workflow, process, version and configuration management services, at least to the greatest degree feasible.

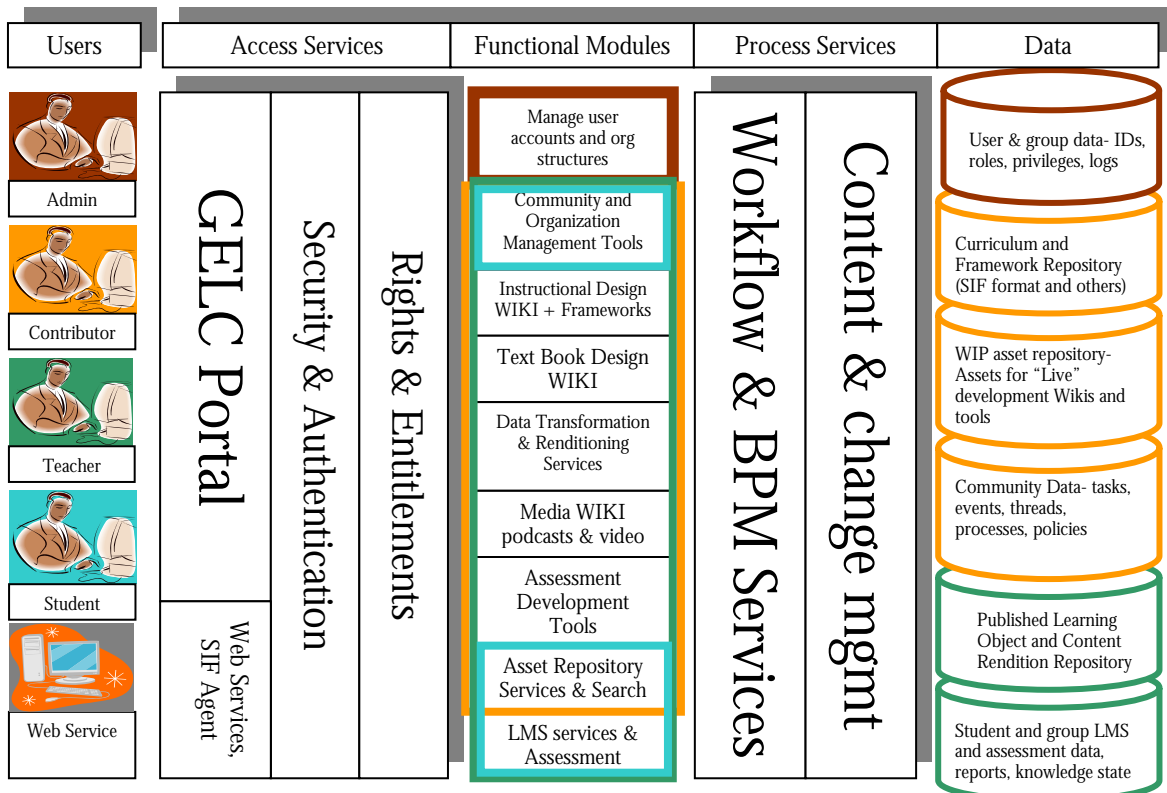


Figure 2- Conceptual Component Architecture

An SOA such as this can support a wide verity of collaborative development and instructional processes. The intent is to support a range of processes from collaborative textbook development, to peer review journals editorial processes, to course development using published materials, and on to managed instruction for a class or a school. Most of the content development processes would follow a dynamic as illustrated by Dr. Derek Keats in his 2003 paper [Collaborative Development of Open Content](#) :

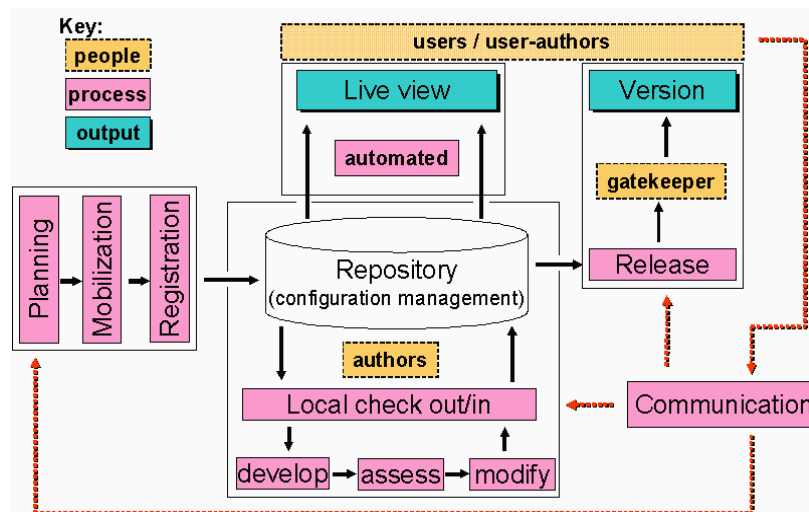


Figure 3- Content Development Process Model from [Keats Paper](#)

Over time, content development efforts from a range of groups would present a broad spectrum of assets available to users of GELC's repository. The use cases related to selecting and modifying assets for inclusion in a course would need its own business process and set of workflows. Once a collection of content suitable for delivery to a population of students the assets must be "Rendered and Packaged" using the Transformation and Rendition tools in its own publishing and export process that could be similar to this approach from the [Australian Learning Federation](#) (Particularly if some of the assets selected require some form of rights approval or license for some intended use and the LMS is external to GELCs):

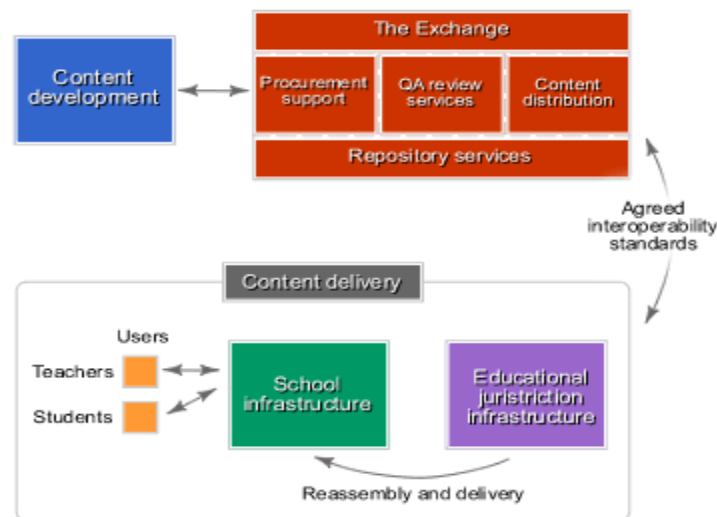


Figure 4- [The Learning Federation](#) Development and Delivery of Online Curriculum

---

## DESCRIPTION OF FUNCTIONAL MODULES

---

As illustrated in the conceptual architecture, the GELC site is really a composite of a number of components that could be implemented in one single application framework or where each has its own or even several applications running in parallel and interoperating. For example, the three Wiki based modules, the 'Instructional Design', 'Textbook' and 'Media' Wikis could be three instances of one Wiki platform (Each configured for the specific use), much as [WikiMedia](#) has done with its set of sites, or there could be three different Wiki platforms each suited to its purpose. Perhaps one or all of the Wikis are not Wikis at all but content management systems that allow contributors to check-out, update and check in document files while visitors can comment on but not change a current version of a document. This description of the functional modules is intended to be used for evaluating (On a separate technology "Score Card") a list of prospective enabling technologies, platforms, applications and tools that might become part of GELC in some iteration, such as: MediaWiki, the popular platform used by WikiMedia; xWiki, a next generation Java based wiki and application platform; Moodle, the popular open source LMS; Plone and Alfresco, the open source CMSs; The work of many regional projects and initiatives in this area (Like the South African project and Active Math project); open standards like SCORM, IMS, SIF, BPEL and SOAP; as well as commercial platforms and applications.

## **USER AND GROUP ADMINISTRATION**

Description: Manage User Accounts and Org Structures. Create, Update, Rename, Modify and Delete any user, role, relationship, affiliation, privilege of any user (As Administrator rights and admin role allows). Also includes content and repository asset administration.

Primary Use Cases: 3) Manage Users and Groups

Related Use Cases: 1) Entry, 2) Register New User, 8) Search Members or Community, 11) Policies, 13) Group Activities (Particularly 13b) – With all modules dependent on user roles and group affiliations, then all use cases have some relationship to this module.

Shared Service and Other Module interactions: This is a function that is germane to all modules and shared services

Special Consideration: In enterprise systems, a single directory service (Like LDAP and Active Directory) is often used to allow multiple applications to use one set of user records. However, most if not all of the existing systems that we integrate use their own or perhaps different directory and access control solutions. Getting a number of systems to share the same user and organizational data may be a long term and continue maintenance and update project.

## **COMMUNITY AND ORGANIZATION TOOLS**

Description: Messaging (e-mail), Discussion Boards, Events Calendar, Conferencing and White Boarding, Shared Documents, Process Management, and Task Management. The implementation of each can be simple and loosely coupled with other modules or complex and tightly coupled.

Primary Use Cases: 4e) assign task, 4f) Accept, reject task, 4g) asset discussion, 8) Search for member, 13) Group Specific Activities

Related Use Cases: 2e) Validate user/Role 3) Manage Users and Groups, 9b&c) LMS tasks

Shared Service and Other Module interactions: Mainly Rights and Entitlements

## **INSTRUCTIONAL DESIGN WIKI (TOOLS)**

Description: Framework Creation & Management, including the Master Skills Framework. Share and create lesson plans, course syllabi, activities, scope and sequence, align assets, compile assets into collections, courses and learning objects. Group activities, discussions, processes and workflows related to instructional design (The delivery of materials to a population and the measurement of progress of that population over time). Just one example, albeit sophisticated, of an instructional design process and workflow might go as follows:

A group (Formed in the community module) of education ministry officials in country X develops their curriculum guidelines using the GELC tools. They do this using their national community of teachers and parents to collaborate on the development of the list of skills to be taught and performance expectations to be measured and met. They then create or select and revise source materials into a course collection of learning objects aligned to their curriculum framework (Using repository, search and renditioning services). This “Package” is to be pilot tested in two schools as a controlled study and improvements made in line. Revised assets are re-packed the following year and the test expanded to 10 schools. This process continues until all of country X has migrated to the new and now ever improving materials.



On the other end of the spectrum, any individual teacher looking to create a elective course for his or her school might create a course framework by selecting a subset of skills in that topic from the master framework, and then using those to select assets to combine into a curriculum package, or even select from a and modify or localize a number of already packaged curricula on that topic.

Primary Use Cases: 5) Master Skills Framework, 6) Create/Compile collection (And particularly 6a) Course Curricular framework, 7) Search for content, 13) Group Activities

Related Use Cases: 2)e) Validate user/role, 3) Users & Groups, 4) Create/edit asset/object, 8) Search for member, 10) LMS, 11) Policy, 12) All user

Shared Service and other Module interactions: As with all the Wiki tools, the shared services are all pertinent control access and manage process. In fact, this module has significant interactions with all of the functional modules. Transformation Services is largely a “Vendor” to this organization and packaging process. Similarly the LMS module is largely a client of this one. Assessment and content assets from the Wikis and the Assessment tool are sources for content.

### **TEXTBOOK DESIGN WIKI (TOOLS)**

Description: This Wiki is focused on the group collaborative development of textbook assets. It can take a curriculum framework as an input, and has tools to create and edit a book map, sections and pages of a instructional textbook using on-line real-time editing tools. Version control (Content Management) and editorial workflows via tasks are used to manage the collective effort of the community and enforce editorial intent, process and schedules.

Primary Use Cases: 4) Create/edit asset/object, 13) Group management

Related Use Cases: 2)e) Validate user/role, 3) Users & Groups, 5) Master Framework, 7) Search Repository, 8) Search for member, 10) LMS, 11) Policy, 12) All user

Shared Service and other Module interactions: All shared services are needed to manage process and access. The community and instructional design modules are most related. However, the transformation services are also closely tied as the Wiki's content must be transformed to PDF, and other formats for delivery. The Assessment tool is a important source of assets for textbooks.

### **INSTRUCTIONAL MEDIA WIKI (TOOLS)**

Description: This Wiki is focused on media based instructional assets. It has tools for uploading or linking to external host locations (Like Flickr or the Internet Archives) for a wide spectrum of media and interactive assets, from podcasts and videos of lectures, to interactive activities, simulations and demonstrations. These assets will have and need rich metadata and may need to be converted into a set of master source formats on import. Version control (Content Management) and editorial workflows via tasks are used to manage the collective effort of the community and enforce editorial intent, process and schedules. In the case of linear media assets and images, rich tagging capabilities are desired to support search and discovery of the assets. Assets posted via the media Wiki can be linked and included in instructional design and textbook wiki pages as well. Its main intent is to support media assets and capture rich media metadata.

Primary Use Cases: 4) Create/edit asset/object, 13) Group management

Related Use Cases: 2)e) Validate user/role, 3) Users & Groups, 5) Master Framework, 7) Search Repository, 8) Search for member, 10) LMS, 11) Policy, 12) All user

Shared Service and other Module interactions: All shared services are needed to manage process and access. The community and instructional design modules are most related. However, the transformation services are also closely tied as the Wiki's content must be transformed to PDF, and other formats for delivery.

### **CONTENT TRANSFORMATION AND RENDITIONING SERVICES**

Description: Takes assets and changes format and structure. For example, it can take a Textbook Wiki document and convert it to a DocBook XML format suitable as a portable eBook, or a PDF file suitable for printing. It can also package assets into portable learning objects in the form of IMS/SCORM compliant content packages. It includes tools for configuring the transformation of assets (Form to form, like XML to PDF or ODF) and packaging of assets (For specific LMSs, or with specific metadata). Ideally it has a modular architecture itself so new transformations, rending and packaging processes can be developed and deployed. At some point in the future it might also become a broker with other repository and delivery services.

Primary Use Cases: 6d) Export Collections (And in particular 6d)iv) custom export object)

Related Use Cases: 4) Create/edit asset/object

Shared Service and Other Module interactions: This is a vendor to instructional management (Mostly), and a client of the Wiki's and Assessment tools. This is the key input to the hosted LMS functionality.

### **ASSESSMENT DEVELOPMENT TOOLS**

Description: Used to create assessment items, scoring rules and processes and collections of items as test forms, using IMS QTI as the item bank structure (Or some other structure that can be transformed to it). Items are aligned to the Master Skill Framework and have extensive and extensible metadata including psychometric parameters and statistics. Users can take a framework, and use this to create a test blueprint. The blueprint is used to select items from the item bank, or identify items to be created, in order to assemble a test form which can be saved or exported. The item creation tool should support interactive item types as well as a wide variety of selected and constructed response item types. The tool should be easy to use and WYSIWYG in design (Like a word processor for QTI items and tests). It also applies workflow processes to item development. You can think of it as the "Assessment Development Wiki" as an analog and companion to the Textbook Development Wiki.

Primary Use Cases: 4) Create Asset/item, (Particularly 4k- assessment special cases), 5) Master Framework

Related Use Cases: 2)e) Validate user/role, 3) Users & Groups, 7) Search Repository, 8) Search for member, 10) LMS, 11) Policy, 12) All user

Shared Service and Other Module interactions: All shared services apply as with the other content development tools. The LMS must deliver all items created here, so the LMS module is a large client of this one. The Transformation module must be able to render and package the items and tests. Instructional Design is closely tied, and textbooks use items.

### **ASSET REPOSITORY AND SEARCH SERVICES**

Description: All assets must be stored and retrieved. This includes assets used in Wiki resources, published renditions and their components, as well as externally linked or stored assets (such as web

pages and physical objects like manipulatives). The repository should enforce role based access rights and restrictions (e.g. people can not discover assets they do not have the right to view.) The repository module has four main components:

- A file based storage and retrieval infrastructure for assets and learning objects.
- A comprehensive, extensible and searchable metadata store (Relational database) exposing and containing the relationships between assets, frameworks and curricula as well as other important descriptive information for assets like the format, rights status, authorship, history, and version information.
- A powerful search capability that should include a full text and metadata filtered search (Or even semantic search) where assets of a certain type can be located (e.g. “Lessons” aligned with “Elementary Math” in “Spanish” that contains “Regrouping”)
- Search communities and people

Primary Use Cases: 6b) Select Assets for collection, 7) Search for content, 8) Search for Member or Community, 9d) Reports (In particular prescriptive and diagnostic tests) 10a) Student searches for asset

Related Use Cases: 2)e) Validate user/role, 3) Users & Groups, 4) Create/edit asset/object, 5) Frameworks, 8) Search for member, 10) LMS, 11) Policy, 12) All user

Shared Service and other Module interactions: All shared services are relevant. Search has deep interacts with all components that touch assets. This includes the Instructional Design Wiki, Textbook Wiki, Media Wiki and assessment dev tools. With regard to assessment, there is special metadata associated with “Items, banks and forms” that does not apply to other assets and resources. Community assets like files stored and discussion boards’ text should also be searchable. To enable this centralized and metadata rich search will required specialized tools (Crawlers, indexers, categorization tools, Xpath and Xquery systems, etc.) and may take quite while to implement well.

### **LEARNING MANAGEMENT (LMS) AND ASSESSMENT ADMINISTRATION SERVICES**

Description: The LMS platform is used to manage and monitor the delivery of instruction to a population of students. It should be SCORM compliant and be able to import IMS content packages, but might extend SCORM with additional features and capabilities. All assets delivered via the hosted LMS should be linked to their source form in the repository for comment and improvement in the source asset base. The delivery of materials is task and assignment base, and the LMS might include basic “Grade book” capabilities. Security is very important as student performance and personal information are highly sensitive. There are basically two modes of the LMS, teacher led course delivery and individual self paced course taking. Group distance learning capabilities, like a virtual classroom would desirable but probably not feasible.

The special capabilities related to assessment and reporting should enable both formative and summative use, and individual and group administration in both on-line and paper formats. Automated prescriptive remediation from the course materials or the whole repository is also needed. The instructional management tools should support both group and self-paced and individualized instruction. The community and collaboration tools modules should integrate the LMS content and enable group projects, homework help and general peer and mentor support, which should be as robust as feasible. Ideally, the LMS uses the same group and user management structure as all the

other modules, a class and a school being just another form of organization hierarchy and community.

Primary Use Cases: 3) Manage User and Groups (Students and teachers), 4e,f & g) Assign and approve tasks (Assignments), 7) Search repository (For prescriptive remediation) , 9) LMS functionality (May be special case of 4e & f), 10) self directed student, 11) Class and school policies, 13) Group activities (Like group writing project).

Related Use Cases: 2)e) Validate user/role, 3) Users & Groups, 4) Create/edit asset/object, 5) Frameworks, 8) Search for member, 10) LMS, 11) Policy, 12) All user

Shared Service and other Module interactions: All shared services apply, in particular authentication and security. The LMS takes content packaged by the Renditioning services. It uses common use and group structures and community tools. Assessment tools are closely aligned with the LMS module as this is the primary delivery environment for assessment assets created in the assessment module.

---

## **DESCRIPTION OF SHARED COMPONENTS**

---

There are two classes of shared services; access services and process services. Access services enforce authentication and manage user and group privileges and constraints with regard to any module, component or function of the system. Process services deal with human workflows and automated business processes both within and across functional modules.

### **PORTAL SERVICES**

Portal services are the “Container” for all of the system’s functional modules. The portal exposes the user’s “View” as a composite application of the applicable modules and functions based on that user’s group affiliations and roles in those groups. This includes a unified “Task panel” or “Inbox” containing all assigned or accepted tasks (For example a user who is a contributor and student would see tasks associates with both roles, like “Review this new article for Spelling and grammar” and “Take the next section test for Instructional Methodology 202”). Another is a link to their groups and courses and files. The portal services are tightly integrated with the other access services as described below. The portal might also include a way to personalize the user interface across all modules and create a personalized GELC desktop of sorts. The solutions should be standards compliant including JSR-168 and WSRP portlets.

### **SECURITY & AUTHENTICATION**

This module maintains SSL connections as needed and authenticates users as necessary (Thorough interrogation of soft or hard credentials.) It associates a log-in session with a specific user’s GUID, and logs all activity. It supports encryption of communications between the server and the user’s client (with 128 bit encryption keys?). Ideally all functional modules call the single set of authentication and security services. One approach might have all user records, keys, and group and role associations might be kept in an LDAP server or in a database. Once authenticated, a user session carries group affiliation and role assignment with a group to enable role base conditional access. Typically Portal Server solutions include this set of services and capabilities but assume that all functionality delivered relies on the Portal for these services. This service should also enable single sign-on and federation with partners and other external systems such as external repositories and services and local school and community portals.

## **RIGHTS AND ENTITLEMENTS**

The rights and entitlements system uses user role assignments and group affiliations to determine the access pillages to all modules and functions and assets. Modern portal server platforms and CMS systems also include group and role based conditional access as a service component, but again the embedded functional modules must relay on these services by design. Otherwise, each module must enforce its own user and role based conditional access, but share a single view of user, group and roles data across all modules.

## **WEB SERVICERS INTERFACES AND SIF AGENTS**

All data exchange functions (Like put and get assets or data, and in particular search) should be able to be exposed as a web service (Machine to machine interaction.) For example, an external LMS might query our repository to find assets suitable for a specific student. Also, a publisher might automate the posting of new assets from their publishing and editorial workflow system. With SIF interfaces, school administrative data and reports can be imported or exported with school partners in an automated fashion. With SIF agents and our own SIF zone, GELC's hosted services can be an integrated part of a school's own infrastructure. For example, when a school adds a student in the student information system (SIS), that student can be automatically registered with GELC and affiliated with the school's group in the system.

## **WORKFLOW AND BUSINESS PROCESS MANAGEMENT SERVICES**

Workflow is human interactions and processes, and Business Processes (BPM as used here) are machine processes. These two services are essentially the same thing, which is the management of task states, with consideration of lifecycle states. What is needed is an abstracted mechanism for managing and enforcing human and machine processes. A simple but power implementation is desired. On the machine processes side, a BEPL (Business process execution language) compliant (When the spec is complete) solution is desired. For human workflows, some simple but elegant method of assigning and tracking tasks is needed, crossing content development and instructional management uses form assigning writing or editing tasks to a contributor to assigning worksheets, assessment and other work to students in the LMS.

## **CONTENT AND CHANGE MANAGEMENT SERVICES**

Description: Content and change management enforces version control on file and web page based assets, enforces access control privileges and restrictions. For example, only a user with the right role can check-out and modify a master source file, and even when they do the change becomes a new file with an incremented version number in its file name. Change management is the same but also includes configurations and relational data changes (e.g. changing the alignment of some asset to the master framework should be reversible and the "Change" logged and auditable). If all asset and data interactions are channeled though this service, there will be no loss of data and all changes will be reversible. In addition all changes can be replayed in sequence to evaluate and correct process problems. It will be very challenging to have all modules share a single CMS service, but this is a essential element of the service oriented architecture described here.

---

## FUNTIONAL AND NON-FUNTIONAL REQUIREMENTS

---

The exact functional requirements for a given development and release iteration will be driven by the use cases at issue and the capabilities and limitations of the technologies and platforms selected. This is largely to be determined based on the phased development plan as it evolves. However, on the non-functional requirements side, GELC must be easy-to-use and appropriate in interface complexity for all users, yet as consistent as possible in UI. This may be a challenge when integration several impendent applications within a portal framework and attempting to support users ranging from young learners to experienced curriculum developers and educators, and in many languages and widely different connectivity and client environments.

**Scalability and Performance:** The system must be designed to scale gracefully and on-demand. However, it must also be functional in the resource constrained environment we will be starting with. As one strategic objective is to connect many existing and evolving projects and platforms, common integration patterns will be developed and leveraged to enable GELC to “Adopt,” integrate and operate systems and applications developed under R&D projects for which funding has expired. Each new system integration effort will require its own project and custom integration plan, but hopefully common web service, single sign-on and portal interfaces will be adopted by the development communities we collaborate and partner with.

**Quality of Service and Availability:** Both of these operational metrics will be initially be of a “Best effort” call. Standard measure like up-time and average response time will be taken and reported. Techniques like fail-over and clustering will be deployed as use demands and support and resources allow.

**Support, maintenance, and operations:** The systems should be as easy to support and maintain as possible, and must be able to operate with a minimum of system administrator, DBA or Developer intervention. Initially, the intent is to avoid custom software development and use “Out of the box” functionality for each of the components selected and integrated. This presents a set of tradeoffs and increases the need to have the communities and groups utilize and share best practices for getting the most out of the released capabilities. All efforts should be made to provide robust user help resources and tutorials on how to build and manage an effective development or learning community.

---

## PROPOSED RELEASE PLAN AND TECNOLOGY ROADMAP

---

The development plan and technology roadmap proposed here is designed to enable a natural evolution of the hosted services offering. The many iterative releases in the technology roadmap and release plan will be grouped into three long term phases: ***Transition and Start-up, Deploy Enterprise Integration Framework***, and ***Extend and Integrate New Components***. Each of these three planned phases involve several iterative release cycles based on RUP and Agile Iterative development and systems integration methodologies. Further more, each release cycle will include the four RUP development phases of Inceptions, Elaboration, Construction and Release. The timing and resources required for each release are expressed in the production schedule and budget which is completed at the end of the Inception phase for each release project. The target is to have a series of 30 to 90 day projects that overlap in that the Inception phase of the next release project starts during the transition phase of the prior one.

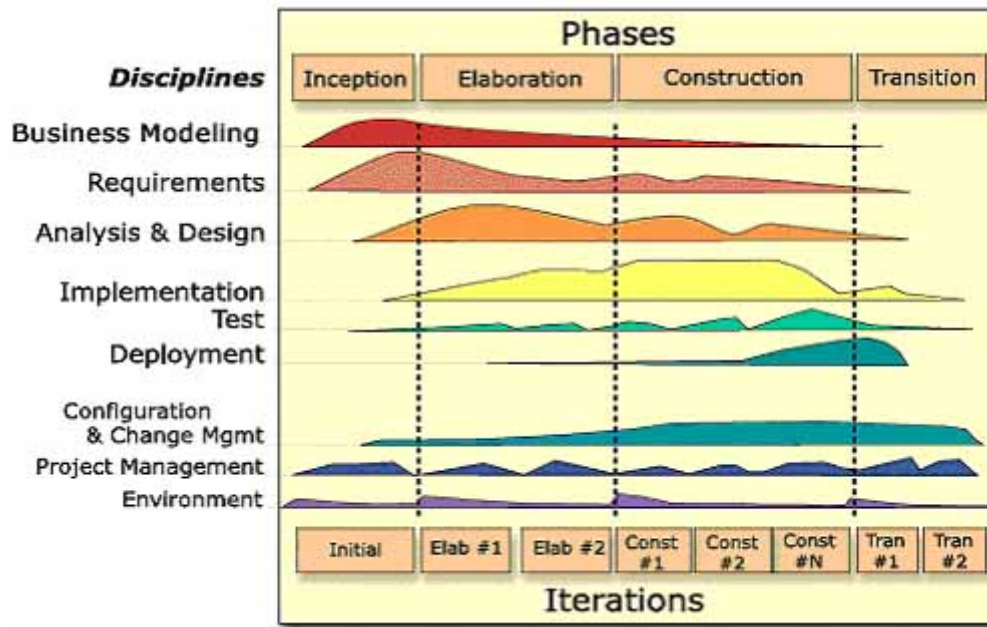


Figure 7- New Iterative Development Phases- [IBM](#)

#### PHASE 1- TRANSITION AND START-UP

This first GELC release is focused on deploying a dedicated infrastructure and web presence for the organization. It will clearly communicate the mission and value proposition of GELC to all stake holders. A primary focus is on soliciting organizations and individuals to contribute any and all existing resources with a strong preference for complete curriculum packages (e.g. complete learning objects, entire textbooks, etc. in whatever form they are available in). It will also be the place where exiting GELC groups, projects and assets will be ported, including vetting and organizing the assets and working with the community leaders to migrate all users to the new site. This content and user migration process will be used to develop best practices including a common tagging syntax and vocabulary, and a unified set of document and page templates.

In the process of facilitating the import of new contributions, much information of significant value to the next phases of development will be gained. They main objective of the end of this phase is to collect a critical mass of content and design the detailed repository data model for future phases. As there are MANY unknowns that are central to designing and deploying a robust set of repository services, this first release must allow for simple and flexible capturing of content metadata, and support a wide variety of content types and forms. This information will be used to create the full repository services implementation in the subsequent development phases. Until there is a strong body of resources suitable for delivery, the consumption capabilities and repository services will be quite limited to discovering published curricular packages with no LMS or customization capabilities. As this is a start-up environment, it must run on our limited initial infrastructure of three SunFire x4100 servers, and must be able to be operated with the small team that is in place. At the same time, this first capability must set the groundwork and effectively deliver and support the call to action to participate and share.

The four releases in this first ***Transition and Start-Up Phase*** are:

**Release 1- *New Web Presence, Front Door, Enable New Contribution and Porting of Existing Assets and Groups.*** This will be a Wiki based implementation where individuals can register and assets can be uploaded or entered into pages which contain organized Metadata and keyword tags. These pages will be organized around a Table of Contents/Subject Taxonomy that will act as a simple content index hierarchy. The implementation of this basic “Contributor” Wiki will utilize the capabilities of the selected Wiki Platform to the greatest degree possible, but will not seek to extend or build new functionality at this point, unless requirements dictate. This first release will be positioned as a ‘Public Beta’, and we can continue that designation until the completion of the transition and start-up phase, or longer.

The Wiki platform to be used will be evaluated and selected based on a number of requirements. These include the ability to appropriately tag and search for assets, manage and export/publish the content in suitable forms (PDF, HTML, Portable Media Files and even SCORM learning objects.) It will also be selected on the ability to be extended and integrate into the planned **Enterprise Integration Framework** in the next phase. There are a large number of Wiki platforms to consider, but based on the evaluation criteria the open source Java based [xWiki](#) is a leading contender as it is feature rich, supports a portal (JSR-168) integration and can run in the same environment as the SUN Portal Server. The very popular and capable [MediaWiki](#) (Used by Wikipedia and all other WikiMeida sites) is also a strong option in terms of stability, scalability, open source community, support and exiting add-ins, though it does not support portal integration. [Plone](#) has been used in several similar OSC projects including [www.cnx.org](#) and [EduCommons](#).

**Release 2- *Port and organize assets and collections as seed content for repository-*** This will be a design and content iteration where the style sheets, templates and tagging methods will be revised and content porting and import projects undertaken. Extensions and modifications to the Wiki platform and the system configuration will be addressed.

**Release 3- *Enable WIKI/CMS based community development of new content-*** This release will use the Wiki tools to allow groups to develop original content and curricula as Wiki pages and asset links. This content, through an editorial and publishing process will be pushed to the ‘consumer portal’ in the last release iteration of this development phase.

**Release 4- *Provide published curriculum “Consumer” portal-*** Implement and release a “Published curriculum” portal and repository where educators and students can search for and find complete vetted curricula and learning objects (Of any granularity from a lesson, to a complete course materials.) The interface for this portal will be a subject and level index with site search and tag filtered search capabilities. Content can be viewed on-line or downloaded and printed. In addition, users can add comments and discussion threads about the materials, including suggestions for improvement.

(The draft phase 1 development schedule is attached as Appendix D, but is preliminary until completion of inception)

## **PHASE 2- DEPLOY ENTERPRISE INTEGRATION FRAMEWORK**

In this next phase GELC will deploy its portal framework and shared services infrastructure. This framework will determine the basic integration patterns and services interfaces available for new components. This will create both significant opportunities and constraints in selecting new components to integrate in Phase 3. As with Wiki tools selection, there is a set of requirements and evaluation criterion for selection of this core platforms and framework. Paramount among the requirements is the need to have a robust vertically and horizontally scalable infrastructure that enables the continued addition and integration of new functional modules to effect large scale



composite applications. In general, this means that platforms will be evaluated in terms of their similarity to the Component Architecture described here. The likely approach will be an integrated portal and java application server platform such as [Exo](#) and the [Sun Java System Portal Server](#) which supports JSR-168 portlets (Such as the xWiki application), other web services interfaces and provides a wide set of core collaborative and community features (Content Management, Full text search, Discussion boards, group calendars, and role based security and condition access services, federated identity management and single sign on).

These enterprise portal systems can be complicated and difficult to manage and master. To help smooth the transition and implementation, Phase 2 will also be broken into a set of release cycles as Phase 1 was.

### **PHASE 3- EXTEND AND INTEGRATE NEW COMPONENTS**

This is the ongoing state of GELC where new functional components are integrated in successive releases, starting with content management and Assessment creation and delivery components, the hosted LMS and federated search and web services interfaces with other repositories and systems and on to DRM and other priorities. This part of the roadmap will be built as systems are selected and community input prioritizes their implementation.

---

### **PHASE ONE WEB SITE DESIGN AND MAP**

---

The phase one implementation represents the first dedicated infrastructure and web presence for GELC. As the mission has evolved since its inception within the 'Java.Net' software developer community, there is a need to offer a site that is considerably more "Educator Accessible" than the present CollabNet based environment. What is presented below is a "Wireframe" layout design and draft site map along with a narrative description of the main pages and interfaces. This draft design will be revised with input from stakeholders, educators, graphic designers and user interface experts and refined extensively through the four iterative releases of phase one. This includes the integration of any new name, logo and brand images selected by the executive committee over this same timeframe. The specific layout and aspects of the design will be limited or enabled by the chosen platform for this first phase. Through the four releases of phase one, pages and features described here will be turned-on in groups as indicated.

Although the site will be built and served using a Wiki/CMS platform, the experience is more like a web site where access to editing is limited to blogs, comments, discussions and content to which the user has the appropriate group and role assignments to modify. However, every page of the site is editable, including globally locked pages like the site welcome page, which are only editable by root system admins (e.g. GELC staff). However, when the user views a page that they do have rights to modify (Like their own blog or their own profile page) the page menu option to edit becomes available as do edit icons within the page. Unlike many Wikis, pages will be built using a number of pre-configured page templates (Which groups can hopefully create and modify themselves) with specific content and metadata fields. Most importantly the focus is on simplicity and utility. The initial experience in particular must be clear and direct and lead the visitor to a level of comfort with how the site works. To help this along, the site should utilize a familiar and internationally understood metaphor, such as a Library, Classroom or Textbook when creating the user interface and site structure. In fact, a sense physicality and location may also be helpful in making users comfortable with the site, particularly when it gets large and integrates many components and communities (Though this will not be an issue for Phase 1.)

## THE HIGH LEVEL SITE MAP AND FLOW

The following illustration describes the page sets to be created and enhanced in the four release of phase 1. This picture also implies some of the page flow of the site:

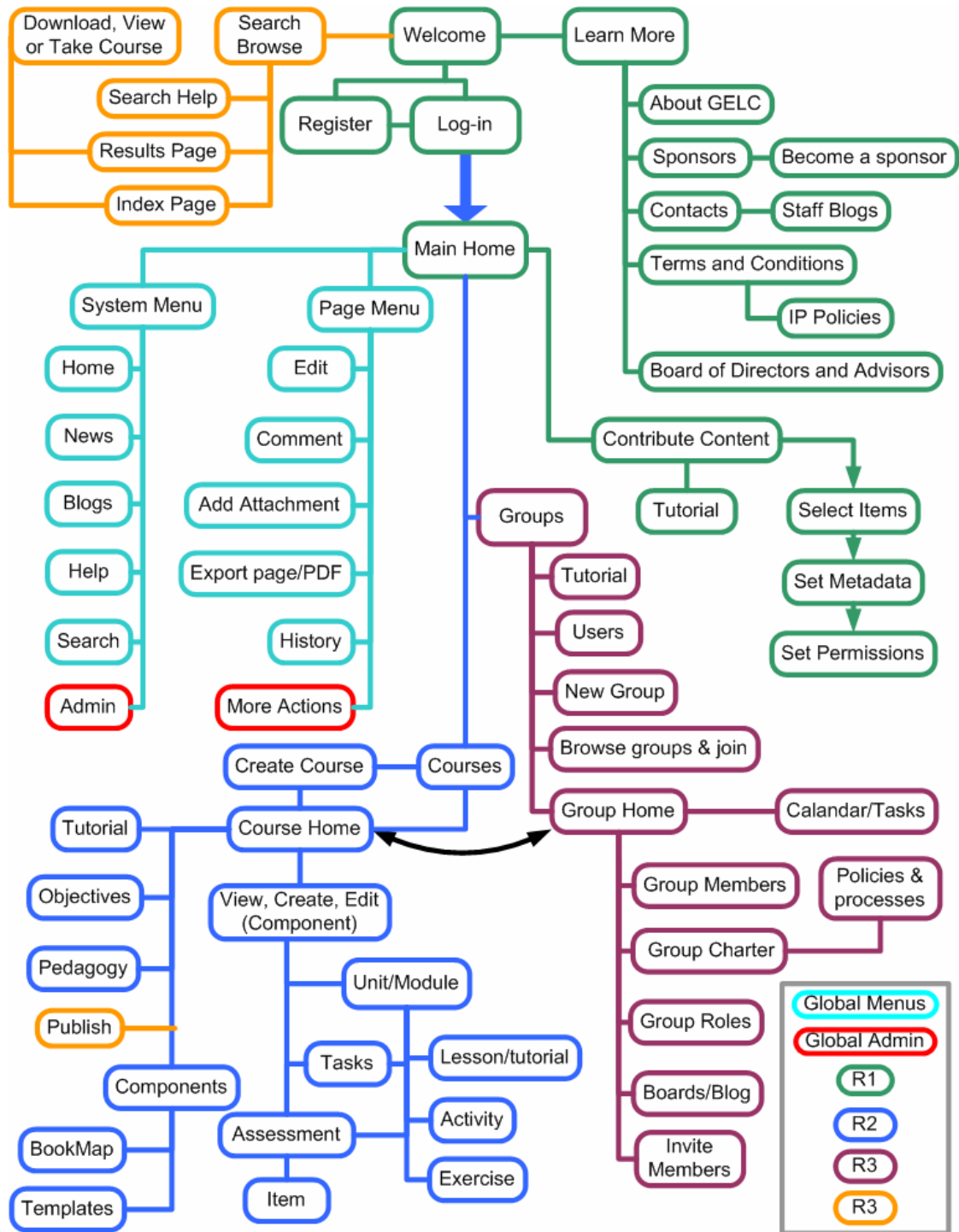


Figure 8- Phase 1 site map in four releases

## THE FRONT DOOR AND ENTRYWAY

This is the first set of pages someone sees when entering the site at [www.gelc.org](http://www.gelc.org). The intent is to create static entry process and highly focused first interaction. There is no global navigation in this first set of pages. The first page must communicate the mission and value of the site and motivate users to actions (Learn more, join the community and browse published courses). If the user is a registered user and has allowed auto log-in, they will be taken directly to the Main Home Page.

Graphically, this page is to be straightforward and clean, but also impactful and engaging. It might use a simple layout treatment with few ornaments, or might some how incorporate a graphical theme, such a library (Perhaps even the Library at Alexandria) or Classroom. Either way, the user should be able to quickly understand what GELC is about and why they should become a member.


		Masthead		Log-in Status	
<div>Welcome to GELC</div> <div>Mission Paragraph/What is OSC</div> <div>Benefits Paragraph</div> <div>Call to action paragraph</div>					
<div>Search For a Course or Resource</div>		<div>Description. Ipsum lorem feddi foop... (more)</div>		<div>Already a registered user? Log in here</div>	
<div>Learn More about GELC</div>		<div>Description. Ipsum lorem feddi foop...</div>		<div>User Name: <input type="text"/></div>	
<div>Register and Join the Community</div>		<div>Description. Ipsum lorem feddi foop... -Policies -Licenses</div>		<div>Password: <input type="text"/></div>	
				<div>Forget Password? <input type="button" value="Log-in"/></div>	
				<div>Sponsors and Logo</div>	
Footer					

Figure 9- Front door page elements

The front door page has three areas, the **Header**, the **Footer** and the **Body**.

The **Header**: This area contains the 'Home logo' in the upper left corner. It has the log-in link and/or user status in the upper right corner. Between the two is the Masthead or Section Head, which displays the section or area of the site the user is currently in (e.g. 'Welcome' or 'Main Home Page' or 'Advanced Linear Algebra Group Forum' etc.) This section head area might use a "Breadcrumb" structure to assist in navigation. The header (With changing Section Head text or graphic) stays consistent everywhere in the site. When multiple languages or regions are implemented, links to those sub-domains should be added to the Masthead.

The **Footer**: This area contains all global attribution and legal reference links. It contains terms of use, privacy policy, copyright policy, licenses, 'Powered by' and 'partner' links, as well as "About" and other links we want to be globally present. Localized attributions might also be placed in the

footer. In Wiki mode, this area might also include the 'created' and 'modified' legends. The footer is also globally present and consistent.

The **Body**: This is the main area of the page and for the **Front Door** layout, and is designed to ground the user and have them quickly select one of four actions.

- In release 1 of phase 1, the options available will be “Learn More” “Join” and “Log-in”. If the user selects “Learn More” the body is replaced with the **Learn More Entry Section**. The “Join” option takes the user to the **Resister** section. “Log-in” follows a standard log-in process and when completed takes the user to the **Main Home Page**.
- In release 4 the “Search for Curricula or Learning Resource” option will be added

The **Learn More Entry Section** contains all GELC information generally grouped into a few (Five or six) sub-pages, all of which appear in the **Front Door** layout by replacing what is in the body. There is a tabbed menu which is used to navigate the ‘learn more’ options. These sections include: About, Sponsors, Contacts, Terms and Conditions, Board of Directors and Advisors. In these main areas the sub topics might include: Staff Blogs, Contribute, and ‘IP policies.’ Other sections and info can and will be added over time.

The **Register** section of the entryway will require significant refinement over time as the full set of user properties is identified, defined and implemented. Initially, the amount of information we ask of users will be limited, and more open in structure. Later we will want more structured user metadata. For example, initially we may ask if the user is a teacher, student or content developer and then provide an open field for them to elaborate. Ideally we would have extensible structured ‘pick lists’ for many classes of user metadata such as teacher sub-classifications like level, subject, region, language, years of experience, etc. The initial registration process is presented in the body area of the entryway and should be similar to the following:

The Site Registration Form will appear in the body, and should be one page. The submission page should be secure (HTTPS connection), once certificates are registered. Depending on our ability to customize registration data in the selected platform, the form fields should include (\* denotes required field):

- \* Community/Screen name (Globally unique but not e-mail address. It will be important to note that this name will be visible to all users and will link to the user’s profile page.)
- \* E-mail address (Valid form, and ideally validated, perhaps with generated password sent to e-mail address)
- Type of member (Pick list: Teacher, Student, SME, Developer, Artist, Illustrator etc.)
- Country (Pick list)
- State/Province
- Topics of interest (Text field)
- Background/Organizational affiliation (Text Field)
- Request to join the following communities : (Multi-select list)
- Do not show my contact info to other members (Check box) (This may not be feasible)

- I am not a minor in their US (Or similar COPPA approach)
- \* “I have read, understand and agree to the privacy policy, terms of use and intellectual property policy” (All policies are linked to open in a new window), and the check box must be selected to register.

If we do not/can not send the user their first time log-in password via e-mail, we will need to add, “Enter password” and “Confirm Password” which must match. Ideally we get them to log-in, change their password, and go to and update their profile page, as a form of tutorial familiarize them with the site and how to edit a page. This page also has a “Forget password” features (Which should reset and send the user a new password). A “Password Hint” might also be added.

The **Search Section** of the Entryway will need some significant work, but will have three modes; simple search, advanced search and subject and level index browsing. A form of Metasearch might also be implemented where external repositories are queried (Like GEM, OpenDirectory, Connections, and others) and the results presented in a single form in some future release. The search section for visitors (As opposed to members) will be presented in the body of the entry layout. This is in the last release of phase one as it can not be finalized until there is a sufficient body of content to search. However, the search experience everywhere should look at feel similar.

In general content and metadata in the site is segmented. There are assets associated with groups and users, which, when “Published” become part of the “Published asset repository.” This is explained further in the **Contribute Content** section, and might represent physical segmentation (Making a copy to a different server) or a logical one (e.g. simply setting the view rights for the present version of a course to “Public”). This will depend largely on the CMS features, search capabilities and access control capabilities of the platform selected. The front door “Search for a Course or Resource” feature may require its own search engine and some non-trivial amount of development. Initially, in Phase 1, it will be best to do as much as we can be with the built-in search capability of the Wiki/CMS platform chosen.

Figure 10- Search for Content Page (FPO)

In this search page, the exact design will require prototyping and testing, but in general, users can select a set of filters, like course on a specific topic and level, and then further target by entering keywords. In the results page/area, the user will have three actions for published courses found: 'View' (or take the course), 'e-mail', and 'save'. View takes the user to the **View Course** page(s), which is just the published home page and all sub pages for the course. This interface should have a book like feel and structure with the Table of Contents as the main navigation, but also a linear way to move through the material. The 'e-mail' option bring up a dialog to enter the 'to' and 'from' addresses and note to the recipient, which send an e-mail containing a link and message to the target email address that points to that course home page. The 'Save' option downloads a PDF of the course. As some course may have media or interactive comments, the 'Save' option may not be available for courses that can not be rendered well as a PDF file.

### THE MAIN HOME PAGE

After a user logs-in, the main homepage is presented. This is the basic layout for the rest of the pages. It contains the same Header and Footer persistent elements as in the Entryway. However it adds two more global components, the **System Menu Bar**, and the **Page Menu Bar**. These two elements are highly dependent on the specific Wiki platform selected. Some implement these using a drop-down menu and some as tabs, and each has a different mix of features available. However it is important that these be consistent and easy to understand but also adhere to user and group access controls. The **Main Home Page Body** is where the features and main options are presented to the member user. The final element of the **Main Home Page** layout is the **Left Side Navigation Bar**, which is an optional and evolutionary design element. It is used to present a quick set of links to any page or function of the site. Ideally it can be made to present user specific links, such as links to group home pages for groups a the current user is a member of, and hopefully also links to task lists assigned to the user, and links to the user's profile. It should also contain popular "Tool" links like a search box, announcements, the **Contribute Content** page, and dynamic links (Such as pages recently changed or visited or pages the user created.)



Figure 12- The Main Home Page layout

The **System Menu Bar**: This menu set is for global navigation to main areas of the site and system functions. The main option here (Again this is highly dependent on the platform selected and its capabilities):

- Home- Takes the user to the Main Home Page
- News- The news area of the GELC site (One page initially)
- Users and Groups- Takes the user to main group page and a list of public blogs/forums.
- Help- take the user to the main help page and FAQ area, which includes a link to ask for more help via e-mail from the staff.
- Search- Takes the user to the site search page (Which is different the curriculum search) and is the basic keyword search for all site pages. The search results page will be tuned as feasible and necessary.
- Admin- This option is only available to users in the system administrator group. It provides full access to all system administration functions including adding, deleting and modify user account, roles and group assignments. It also provides access to create new groups and roles, pages, templates and all other aspects of the site. These capabilities are entirely defined by the selected platform and strong capabilities in this regard are a critical selection criteria for the Wiki platform.

The **Page Menu Bar**: This menu set is for operations related to the current page being viewed, regardless of whether it is the user's profile page (Including password changes) or the main home page or any other page. However, this is user role and group aware and will not allow users to change things they do not have the explicit right to. Group administration rights are hopefully independent from system administration privileges, though most platforms do not support this. The options here are also Wiki platform specific, though increasingly standard across platforms. One key feature is "Export as PDF" which will need to be added in if the platform does not support this natively. The page menu options should include:

- Edit- Take the user to the edit interface for this page (Should be as WYSIWYG as possible)
- Comment – Allows the user to add a comment to the page (Same a discussion in some Wiki's)
- Add attachment- Takes the user to the interface where they can attach an image or document to a page (This is a global tool in some Wikis. Some wikis have global files while others have files associated individual pages.) Ideally (But not likely), when an attachment is added it can be targeted at just the page or a group or the entire site. (This form might be the same as the "**Contribute Content**" pages, or a simplified version meant for use on the current page only.)
- PDF export- Renders the Wiki page being viewed to a PDF document well formatted for printing. Ideally, pagination, headers and footers and captions are all well styled. Even more challenging, and likely not feasible but highly desirable will be allowing groups to create their own PDF rendition styling and layout.
- History- Presents the version history, version control, compare and roll-back features for the present page.

- More actions- This is a catch-all for other available page level functions. For example, in xWiki, this contains a long list of options if you are an administrator, including the ability edit the page form, access rights to the page, preferences and other functions. Other wiki's, such as MediaWiki, have no such set of features, but it does have a "Watch" feature and user preferences page. This is the menu where page level access controls are available only to system and group administrators.


The **Left Side Navigation Bar**: As mentioned above, this navigation bar is a flexible design element that can contain links and functions that appear in other menus. It should also be site area specific, e.g. the options in the Groups area and the Courses area should be used to facilitate local navigation in that part of the site. Some of the elements that should be included in this menu in the **Main Home** Page are:

- Welcome
  - Feature 1 (Tutorial- How to use this site)
  - Feature 2 (Edit Your Blog)
  - Feature 3 (Feature Article)
- My Groups
  - Group 1 home page
  - Group 2 home page
  - (more)
- My Profile (edit)
- **Contribute Content**
- **Groups**
- **Courses**
- **Search** (The small tool version)

The **Main Home Page Body**: This is the doorway to the communities and content development areas. It will contain some of the elements of the left side menu in addition to its X primary sections:

- Featured Group- (Link to specific group home)
- Featured Course- (Link to specific course home)
- **Contribute Content**
- 'Join or create a group'- Links the user to the **Main Groups** page
- Browse and Create courses – Links to **Main Courses** page
- Topic of the week...

#### GENERAL WIKI AND SITE EDITING INTERFACE RULES

Every page and element of the site is an editable element of the Wiki/CMS. However, most users will only be able to edit pages (And page elements) that they themselves create (Like their profile page, or pages owned by a group that the user is a member of such as a set of textbook pages). To facilitate editing and navigation, all editable elements (Things illustrated in the wireframes in a 'boxed group' like a feature groups, or even the **Left Navigation Bar** itself) should have an edit icon visible () if the user has the rights to edit that page or element. When the edit icon is selected (Just as the edit this page option in the **Page Menu Bar**), the user is presented with the WYSIWYG editor for that element or page. Similarly, the other basic Wiki/CMS functions (History, Rollback, Compare Version, Attachments and Comments) are ubiquitous and follow role based conditional access rules, but only 'edit' is integrated at the page element level.



Group administrators will have limited access to manage groups and roles and control page access to groups and roles. Also, the group administrator hopefully will be able to allow or prevent non-members from viewing all or a subset of the group's pages and files. Ideally, in the "More Actions" menu of the **Page Menu Bar**, administrators will have access to the 'page access rights' function to allow or deny 'view' and 'edit' access to any group, role or member, but only for those pages the group administrator has administrative rights to (e.g. Group administrators can only control pages belonging to the group.)

All new pages will use one of a small set of templates, like "User Profile," "Blog" or "Activity/Type." (Note: the user profile page is automatically generated when the user registers and acts as that user's main home page on the site as it links to all their groups and assets.) These templates should include "Fill me in" instructions and each component of the page should be independently editable (So as to avoid finding a small bit in a large and complicated page.) Even these templates can be edited, but only by system administrators. Creating new templates is likely to be a vary technical and challenging task. Group administration may well also require a certain level of technical ability.

### GLOBAL SEARCH INTERFACE RULES

Search is a core feature of the offering, and one that presents a very significant technical challenge. It is unlikely that we will find the level of search ability in any of the Wiki platforms being considered to meet this goal, which may require custom search integration in the release 4 timeframe. All efforts should be made to have one global **Search Interface** that is used contextually and is both flexible and powerful in its use of metadata. For example, searching for users selects the appropriate metadata filters (Group and role affiliations) while searching for content can be targeted to courses, or images, or published assets only, etc. What is proposed here is a "Metadata grid layout" which would be used in the body of the Main Page, or in a pup-up window depending on where search was called. In this example, the user might have been building a content page and wanted to find an image of an apple:

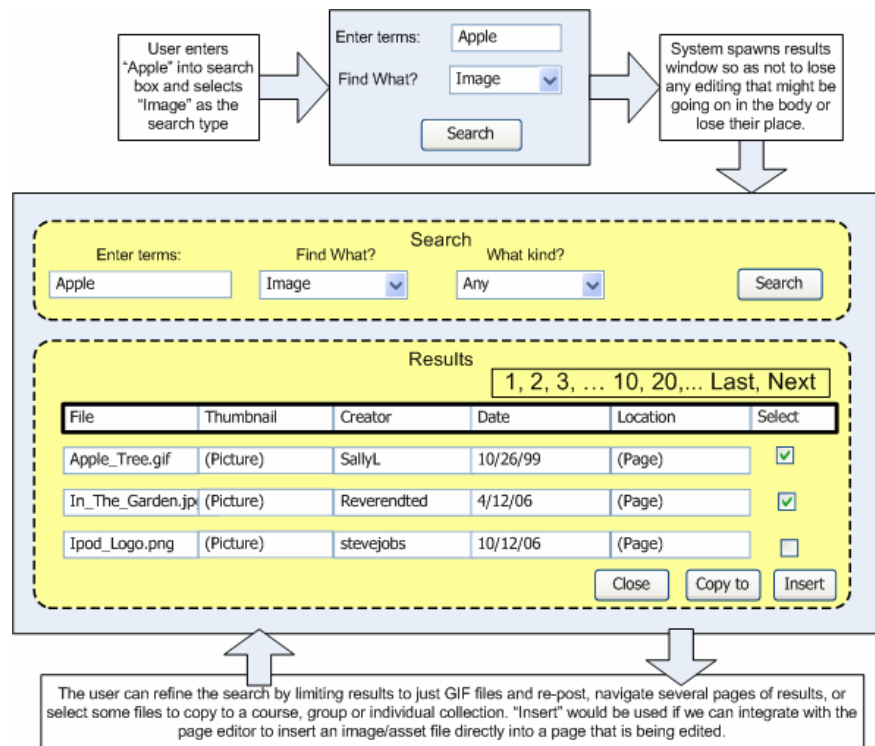


Figure 13- The Global Search Page "Metadata Grid" layout

## THE CONTRIBUTE CONTENT SECTION

This is the area where file based assets and of all types can be contributed to the community. It is intended to be easy to use and capture needed metadata to make the assets appropriately discoverable in the repository. It is only intended for file based assets (PDFs, office documents, text files, Flash files, MP3/Podcast files and SCORM Content packages, for example.) All attempts will be made to gracefully identify and present icons or thumbnails where feasible and potentially extract any metadata that we can from the file itself or the context (Like file size, and contributor, contribution date, etc.) The contribution should follow a 3 step process: 1) Select item(s) to upload, 2) Set Metadata, and 3) Set permissions. After contributing, a user can modify any of this information by going to their personal profile and selecting the “View my contributions and collections” link (Which may not be implemented in the first few releases). On that page they can edit any of their contributions and metadata. Hopefully they can also see where those contributions have been used. The pages might look something link this (Or perhaps they can be consolidated into a single page):

The diagram illustrates the layout of the 'Contribute Content to the Community' page. It includes a masthead with the GELC logo, a system menu bar, and a page menu bar. A left sidebar contains navigation links like 'News', 'My Groups', and a search box. The main content area features a dashed box for 'Contribute Content to the Community' with instructions and a 'Tutorial: (link)' section showing a three-step process: 1. Select Item(s), 2. Describe Items, and 3. Set Permissions. A 'Contribute' button is at the bottom of the main content area, and a footer is at the very bottom.

Figure 14- The Contribute Content First Page

The **Contribute Content Tutorial** link takes the user to a page with step-by-step instructions with illustrations of the entire process plus FAQs related to contribution.

When the user selects the Contribute Button, they are present with the **Select Items** page.

The 'Browse' option brings-up the clients file browser to select files. If multiple files are allowed to be selected in the client file browser, then they should all be uploaded at once, and all receive the same metadata and permissions in the next two steps.

'Select file type' is optional. Ideally the upload script will check that the file extension and mime types match before allowing the upload. Also, ideally there would be some ability to implement server side processing (Indexing the text, generating thumbnails, extracting metadata, transforming to some common format) in future releases. The end state objective is to be able to import and transform any file based content into a common source format directly editable in the WIKI in WYSIWYG fashion. Virus and malicious code checks would be very important to add early on.

The options in the yellow box, 'Select, View and Create Collections' is a way for a user to have several 'directories' to put files in, which is particularly useful for folks contributing resources in different areas or of different types. However, this may need to wait for a future iteration as it may be significant work to implement.

Figure 15- The Select Items contribute page

Selecting the 'Contribute' button either brings-up an error message (With tips on correcting) or move to the **Set Metadata** page. Ideally any format metadata that can be determined by interrogating the files should be automatically pre-filled-in in the form:

Figure 16- Set Metadata for Contribution page

This page is for setting all the needed metadata. The set of information presented here in the yellow areas (Format Metadata and Instructional Metadata) are FPO. The actual fields and lists will be determined with input from the CAO, educators and the technical sub-committee. Some important metadata is largely subjective, like completeness or quality of the asset. The potential breadth of instructional metadata can be staggering, ranging from reading level to learning style and sensory modality in addition to granular skill based framework alignments. Initially the metadata will be minimal and somewhat free-form (More text entry boxes than pick lists) and will be expanded and codified over time. Ideally, the pick lists for frameworks will come as part of a collection affiliation. Something along the lines illustrated here will be in place at the end of phase 1 IF the platform is capable of encoding such structured metadata in a form like this and use this data as a search filter.

When the user clicks the “Set metadata for...” button, they are taken to the last page, ***Set Permissions*** form page:

*Figure 17- Set Permissions for Contribution page*

Again, this will be significantly refined and the fields are for basic illustration. For example, we may want people to be able to select multiple licenses which would have the drop box pick list with a set of check boxes. Or, perhaps we might decide to enforce one single license which would remove the need for the selection option. Specifically which licenses will be supported and tracked is a policy issue for the Executive Director. The group list might include all active groups or just groups the user is a member of as well. Similarly the rights might be expanded to roles within groups as well in a multi select menu. Also, content contributed directly to a course collection may have default access privileges.

Upon completion of the process, by selecting the “Set permissions” button, the user is taken to a confirmation and “Thank you” screen and then returned to the contribute content first page.

## THE MAIN USERS AND GROUPS AREA

The section of the site represents the bulk of release 3, and the area where users can find groups to join or start their own group. The main page has five options in the body:

The ***Tutorial*** option presents a page detailing groups and how to use them. The ***Users*** option brings up the ***Find User's Page***. The ***Browse and Join Groups*** option shows a similar page for groups. The ***Create New Group*** is the set of pages where a member (Perhaps a special class of member) can charter a new group. Finally this page contains a list of “***My Groups***” which are all the groups the current user is a member of as a way to link directly to each group's ***Group Home*** page.

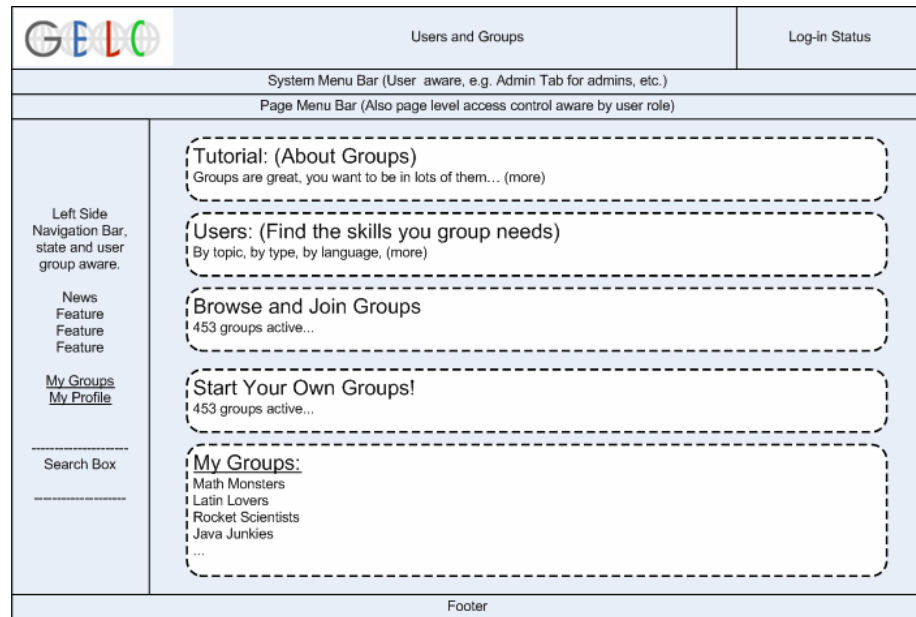


Figure 18- Users and Groups main page

The Find Users page only lists user's screen names. All user's screen names link to the User's Profile Page. However, if the user has selected to remain private, their profile page appears blank. A user's profile page is created when they log-in and can be modified by the user (Or a system admin) at any time. User's contributions, group affiliations, roles and other metadata are available in their profile page.

This search page, which shares basic UI design with the content and other search features, has the user entering the search string and all metadata filter parameters in one query input area with a search button. The exact user metadata pick lists here are based on the required and optional fields in the user registration and user profile pages. There can be a number of actions taken from viewing the User's Profile Page (If public) sending the user an e-mail (if the user chooses to post their e-mail address.) Security concerns may preclude direct e-mailing between users without the recipients prior consent. Alternately this might be a “Message” (e.g. comment) that is posted to the user's profile page or even a separate “My messages” page. This Invite option is only available if the Find User page was reached from the Invite Member page. If the action is to invite the user, the Invite Member to join group page is re-presented with the target user already filled in.

Find Users

Log-in Status

System Menu Bar (User aware, e.g. Admin Tab for admins, etc.)

Page Menu Bar (Also page level access control aware by user role)

Find user's tips.....

Search

name Type Language Subject Search

Results

1, 2, 3, ... 10, 20, ... Last, Next

UserName	Type	Languages	Subjects	Levels	Select
Bob	Teacher	En,Fr,Gr	Science	K-8	<input checked="" type="checkbox"/>
Sally	Teacher	En	Literture	Colledge	<input checked="" type="checkbox"/>
Ted	Designer	En,JP	any	any	<input type="checkbox"/>

E-mail Invite

Footer

Figure 19- Find Users page

**The User Profile Page** serves two purposes; it is the home base for the user where they can link to their groups, tasks and contributions, and their identity within the community. It should allow the user to personalize the page, almost like their “My space” page, with a picture, personal Blogs, and links to what is important to them and what they have contributed. They also use this page to edit their system profile include password, e-mail, and remove themselves from groups they have joined.

{Need wireframe for User Profile page, with “My” links here}

The **Browse and Find Groups** page is yet another search page where the search is restricted to group, which are listed alphabetically or can be filtered by group type (Professional organization, education institution, certificate authority, etc.). Each group name in this listing links to that group’s **Group Home** page.

The **Create New Group** page is where a user (Perhaps only users that have been vetted by GELC will have access to create groups, as it requires a significant commitment) can become the group administrator and found a new collective. Groups may be affiliated with an educational organization (Like UC Santa Cruz or LAUSD), a professional organization (like NCTM or Associated of Independent Guitar Teachers), or act as a certification organization (e.g. a peer review credential group who can put their seal of approval on a curriculum). When a new group is created, a set of group pages are automatically generated from base templates for the group home page, group charter and policies and a few default roles for the group are created. The founder is granted limited admin (editing) rights to the pages owned by the group, the ability to add and remove members and create new roles within the group and assign members to those roles (Including the group admin role itself). (Extra bonus if we can generate a sub-domain ‘www.group.gelc.org’ to point to this home page and all pages owned by the group.)




		Create New Group	Log-in Status
System Menu Bar (User aware, e.g. Admin Tab for admins, etc.)			
Page Menu Bar (Also page level access control aware by user role)			
<div> Left Side Navigation Bar, state and user group aware. </div> <div> News Feature Feature Feature </div> <div> <a href="#">My Groups</a> <a href="#">My Profile</a> </div> <div> Search Box </div>	<div> Create Group tips (More- Tutorial) </div> <div> <div> Set-up new Group </div> <div> Name </div> <div> Create Group e-mail list name @gelc.org </div> <div> Enter long description of the group </div> <div> Group logo image </div> <div> Browse </div> <div> User name of charter member </div> <div> More </div> <div> <input type="checkbox"/> Make group home private to group </div> <div> Create -&gt; </div> </div>		
	Footer		

Figure 20- Create New Group

The “logo” is optional and can be added or changed latter. “Charter Members” are also optional. The “Make group home private” option may not be desirable, but is an example of the type of set-up option that might be added or included, similarly this set-up might include a group content license that applies to all group assets. Selecting the Create button creates the group and the group home and related pages. The user gets a confirmation screen with instructions about next steps, namely edit and create the **Group Home Page**, which is the next page presented.



		Group Home	Log-in Status
System Menu Bar (User aware, e.g. Admin Tab for admins, etc.)			
Page Menu Bar (Also page level access control aware by user role)			
<div> Left Side Navigation Bar, state and user group aware. </div> <div> News Feature Feature Feature </div> <div> <a href="#">My Groups</a> <a href="#">My Profile</a> </div> <div> Search Box </div>	<div> <div>  Group Logo </div> <div> <div> Group Mission and Charter </div> <div> Short Description </div> <div> Charter and Policies </div> <div> Courses and Publications </div> </div> <div> <div> Group News and Announcements </div> <div> Item ..... </div> <div> Item ..... </div> <div> New Item </div> </div> <div> <div> This Week's Events and Tasks </div> <div> Event </div> <div> Event </div> <div> View Tasks </div> <div> View Calendar </div> <div> New Event or Task </div> </div> <div> <div> Group Blog and Discussion </div> <div> Feature Item... </div> <div> Feature Item </div> <div> Blogs and Discussion </div> </div> <div> <div> Group Members and Roles </div> <div> Feature Member... (Name) </div> <div> Accolade for feature member </div> <div> Member List </div> <div> Invite new Member </div> <div> Roles </div> </div> </div>		
	Footer		

Figure 21- Group Home

The group home page might be better set-up as a set of Tabs, in which case the features are on the “Home” tab and the other tabs would be “News,” “Tasks”, “Calendar”, “Discussions” and “Members.” If the tabs interface can not be implemented easily, then the **Left Side Navigation Bar** should have a Group navigation menu added with the same links. The group admins are the only users who can edit these group pages, though any member can add comments, events, tasks, invite new group members, and members can take on as many roles as they like. Editing group home allows for changing of features. Charter and Policies button brings-up a simple page with that content and related comments and discussions. Again only the group admins can change the policy and charter statements, which might be as formal as bi-laws. The ‘New Item’ button is a simple template for new news items and announcements.

The Tasks area of the **Group Home** is where the group organizes and tracks its work and many priorities. The New Task/Event button brings up a pop-up form where any member can fill-in Due Date, Priority, Title, Description of the task and select a role or individual members to which the task is to be assigned. If they select “Event”, the due date is the event date, and the event appears in the **Calendar** rather than the **Task List**. All users of that assigned roll (Or the individuals assigned the task) will see this task in the “My tasks” area of their **User Profile** page, and might also get an e-mail with the task. They can also view the tasks by selecting the ‘View Tasks’ button, which brings up a list of tasks grouped by roles, and sorted by due date and priority. If a user has completed a task, they can use this task list to mark a task completed (And by whom). (There should be an edit task button by each task in the list and event in the calendar to use to update the state of the task, which can be to change the task, mark it as some % complete or finished, or delete the task.) The exact look of the task list and calendar is TBD.

The **Blogs and Discussions** area contains open topical forums for the group leveraging the chosen platform’s available interface for these common features. The **Members List** screen is essentially the **Find Users** screen filtering users by affiliation with the current group, with an added field of ‘Role’ and an “Assign Task” button in place of the “Invite” button. However, the group administrators have one added button which is “Remove user from group.” This should bring-up a pop-up where the admin can write an explanation of why the user(s) selected (vis the action check box) have been removed from the group, which is sent as an e-mail to the unwanted user(s).

The ‘Roles’ button presents the **Roles** page which is a simple list of existing roles for the group with a description of the role. (Clicking on a role might lists all members of that role.) The default roles are Writer, Editor, Reviewer and Group Admin. If the viewer is a member, then there is a button next to each role that says “Accept” or “Decline” depending of which roles they currently have in the group. The one exception is the Group Admin role, which has a “Request” and “Decline” option. The request sends an e-mail to the group admin(s) with a link to a page where they can accept or reject the request. The Group Admin(s) can edit the page and add, modify or delete roles (Expect the group admin role). These changes appear in the appropriate pick lists, like assign task to role. Any tasks assigned to a deleted role are also removed with the role, and a warning informs the Group Admin of this.

The last option, available to any group member, is the “Invite New Member” button. This brings up a form where the member user names can be added (Plus a search button which brings up the **Find User** page with the ‘Invite’ option.). Invited members get an e-mail with a link to accept the invitation or reject it (There might also be a “My invitations” area in their personal profile page.) There is another box where an e-mail address of a non-member can be added. This will send a pre-formatted solicitation to join with a link to a special register page that will automatically add the member to the group (This may not be fully feasible). If the user viewing the Group home page is not a member of the group, the “Invite Member” button becomes a “Join group” button, which automatically adds the user to the group, though without a role. Any user can leave a group by



editing their “My groups” section of their **User Profile** page and selecting “Leave group.” If the last Group Admin leaves the group, then the group is disbanded and removed from the system, and a warning as such should be provided in this case. (Alternately, the group goes dormant and the next person to join the group gets to be the group admin and adopts the group.)

## THE COURSES AREA

From the Home page you can reach the **Courses** in development area where you can browse and view public courses or create a new course. The interface has two elements, ‘Search’ and ‘Create’. The page contains a “Search for course” box (Like all the other searches on the site), where the user can filter by subject, level, group, and perhaps other pertinent metadata. If a user selects a course from the list, they are taken to the **Course Home Page** for that course (Assuming the user has rights to view the course in development, and if not a “Private” message is display and the user can choose to join the group to view it and participate.) The ‘Create Course’ button brings up the **Create Course** page, which is a web form similar to the **Create Group** page:

Figure 22- Create Course Form

In this form the user names the course, select its availability, create its course description, and sets any and all pertinent metadata. Through an “Include/exclude” multi-list interface (Or something equivalent), the creator of the course can give owner rights to any group he is a member of, or leave himself as the only owner. There must be at least one owner to create the group. Any of this course header information can be updated, modified or extended by any owner by selecting the “Edit” icon in the Course Info area of the **Course Home** page, which brings up this same **Create Course** page but the “Create Course” button is replaced with a “Update Course” button. Do keep in mind that group administrators and the creator can limit access to modify pages to certain individuals or roles (There may be some cross group complexity to deal with, or this may be simply limited to the owner.) Selecting ‘Crate Course’ has the system generate all the default Course pages.

Once the Course is created, the user is taken to a blank **Course Home** page based on the default Course page template (e.g. Logo in the left, and features as described.) From there, the user and all

owners can edit and update any part of the course page set (With group and role page access controls as previously described). This includes the **Course Home**, **Course Components**, **Units/Modules/LOs (Lessons/Tutorials, Activities, Exercises, Assessments, and Items)**, **Course Objectives/Framework**, **Pedagogy**, **Components**, **Component Book Maps**, and **Unit/Module/LO Templates** pages. Each page has a set of tasks associate with it that might be a linked window of “Tasks for that page” or a **Left Navigation Bar** element. The **Course Home** page also contains Features and News related areas. The main part of the course home is a link to the course components. As described in the Conceptual Content Model, Courses are made up of Components (e.g, the different books used in a course, like a textbook student edition, a curriculum delivery and teachers guide, a workbook, a mastery test, reference resources, etc.). Each component is constructed of Units (aka Modules), which are of one of the known types, or some new type that is created for the course. At any place within the units, a link to other units in this or other courses can be entered as “Related links.”

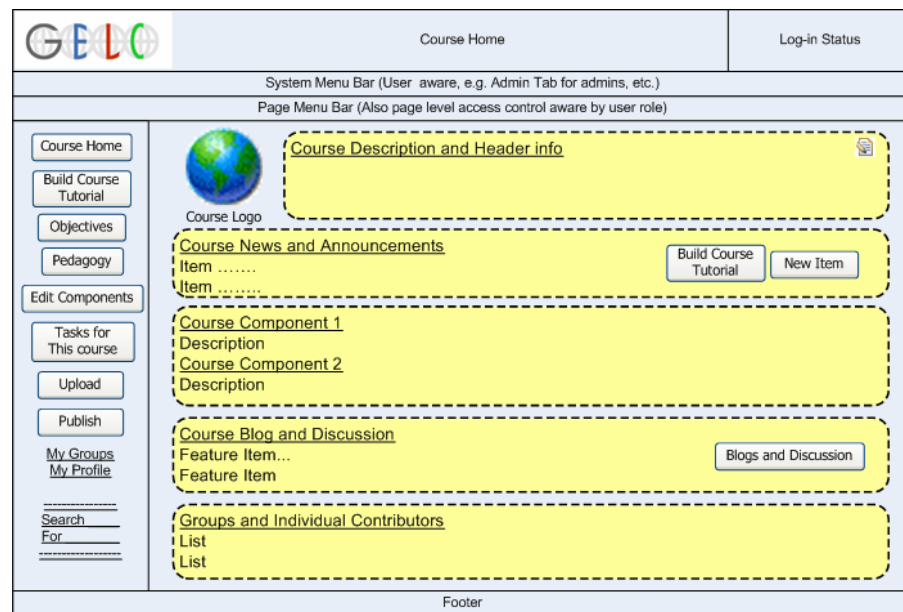


Figure 23- Course in development home

Some issues related to content in courses are important to keep in mind: Content is contributed to individual user's collections or to group collections (As described in the contribute content). To use an asset in a course by a user, it must exist in the course collection. Assets can be added to the course collection by contributing them to that collection directly, or by selecting and copying them from one from the user's or a group's collections to the course collection. Using the Search, as illustrated before, any asset, unit or even entire course can be discovered and copied to one of those course available collections. This affects a branching of that asset, and any future modification of that asset within the context of the present course is made on that course's copy. However, attribution to the original source and all those who modified it along the line is maintained by reference to its parent copy (As long as it exists.) If this level of CMS capability is not available in the selected platform, modifications should be considered as this flexible sharing, re-use and medication capability is central to the Open Source Curriculum value proposition.

As with the Groups area, each of the main functions of the Courses area should be presented within a persistent Tabs interface or the Left Navigation Bar should contain those links. Several of the Course Home page elements are similar or identical to other area in the site. The 'Build Course Tutorial' takes the user to a course on building courses. The 'New Item' button brings up a page to

create a new news item or announcement. The “Blogs and Discussions” button take the user to the ***Course Discussion Threads and Blogs*** page. The “Upload” button takes the user to a slightly modified version of the ***Contribute Content*** page where the uploaded asset goes to the course collection directly. Similarly, the search tool, when it bring-up the results page defaults to copy the selected assets to the current Course Collection.

The Course ‘Objectives’ button takes the user to the ***Course Objectives*** page. This page is where the contributors construct the “Learning Objectives Framework” for the course, which is, more formally, the instructional metadata controlled vocabulary and hierarchy. Functionally the framework is presented as a hierarchical list (Outline) of defined content tags for the course, where the root is the topic that was selected for the course from the ***Master Curriculum Framework***. The master curricular framework being a similar hierarchal structure maintained by a special group who is the one and only owner of the Master Curriculum Framework, which might also be called the Master Subject Taxonomy for GELC. If someone wants to build a course, but does not find an appropriate Master Framework Node to connect to, then they can work with the ***Master Framework Group*** to build out an appropriate branch.)

These Course Framework lists are stored as data objects and as such can be used in data element pick lists in the forms based headers for all course components, units and assets (As described below). Hopefully these frameworks are as easy to edit as an outline document, but more likely a specific interface is required to help maintain the structure of the data (e.g. the skill name, its parent and child relationships and its long description. Also, it would be good to be able to view a list of assets and units aligned with a specific node of a framework from the framework list itself.) The exact interface for this page is to be developed though prototyping strategies supported by the selected platform and with significant input from users and the CAO of GELC. The ability to handle this framework interface well is a key platform selection criterion.

The structure of the framework when viewing and editing might look like this:

*(Master Subject Taxonomy)*

*Subject (Math) – Description*

*Topic (Elementary Math) – Description*

*Sub-Topic (Arithmetic) – Description*

*Sub-Sub-topic (Long division) – Description*

The course in question would be aligned with the Master Subject Taxonomy at any level, perhaps the leaf node in this case of “Long division.” Any Course would then be tagged with a appropriate topical tag such as “*Math:ElementaryMath:Arithmetic:LongDivision*”. The Course Objectives Framework would then extend that structure down to the skill level for the course and at the appropriate granularity. In this way, each and every course added to the system effectively extends the master taxonomy down to the course skill level. These hierarchal framework structures appear in all subject pick lists in the forms for new units and assets added to the course collection. Now, any asset should be able to have alignments to many frameworks and many nodes. In this way, even though a course is meant to be about one topic form the master framework, it can contain cross curricular components and links to related topics, materials and courses thought the course’s materials!

The extended framework for this Long Division course might look something like this:

(Course Root- Math:ElementaryMath:Arithmetic:LongDivision')

Terms

Numerator

Denominator

Quotient

Reminder

Identify parts in vertical notation

Identify Parts in Horizontal Notation

Identify Parts in Long Division Notation

...

Long Division Notation

Format

Presentation of Place Value

....

Sub-Sub-Skill

The Course 'Pedagogy' button takes the user to the **Course Pedagogy** page. This page is a simple open Wiki document where the issues of instructional design, style, and links of topics related to the learning theories at play in the curriculum/course are described and refined by the contributors through the development process. It is an asset that can be published with the curriculum.

The 'Edit Components' button takes the user to the **Edit Components** page is where the basic structure of the course is created. Each component is treated as an independent "Book" or linear sequence or organization of units. For example, a course may have just one "Course" or "Book" component with a single TOC intended for all users of the curriculum. However, a curriculum might also include several parts intended for different uses or users. For example a comprehensive course might included 12 components: a 'Student Book', a 'Work Book', a 'Teacher's Edition' containing lesson plans and group instructional strategies, a 'Reference Book,' seven "Literature Selections" and a 'Mastery Assessment' (Which is not to be made available to the students until they have completed the course materials and offers a psychometrically valid scale score.) Another scenario is that the Course has no structure at all, but is just a collection of units or modules that can be used by the student in any order at their own pace, in which case each unit would be its own component of the course. The **Edit Components** page also contains a description of how the components are to be used. All new courses get one default component call "Book". Please note, any complete components copied into the Course Collection are automatically added to the course components lists.

In the Wireframe below, the "Select Component Type" pick list relates to the list of available Templates as described n the **Create Template** page. Initially, Templates may not be user editable, but this function would be available to system admins or at least system developers who will create new Component Templates and new Unit Templates as the community needs. In this case the "Create New Template" button becomes a "Request New Template" button. After a component is created, the Book Map for that component should be created. Selecting the "View and Edit Book Map" button for a component brings-up the **Edit BookMap** page for that component, which is really the table of contents of units (Pages) be created or assembled and their organization in the book. The interface should be similar to the edit framework interface. The elements of the TOC create a set of linked pages in the component, one page of the appropriate template per entry. (Please note, adding a page to a component while viewing and editing also modifies this book map.) It is also this screen where users can select existing Units in the Course Collection as elements of the Book

Map, allowing someone to compile a complete book of existing units. The page templates used for each component page are the ones available for the related component template (e.g. one Component Type “Book A” template might include unit templates for ‘Tutorial’, ‘Exercise’ and ‘Assessment’ templates styled to work together, while another might have pages for ‘Video Lecture/Presentation’, ‘Group Activity’, and ‘Report Assignment’). Any units imported into a course will bring their own template, and may need to be re-styled some to fit visually and structurally in the component to which it was added.

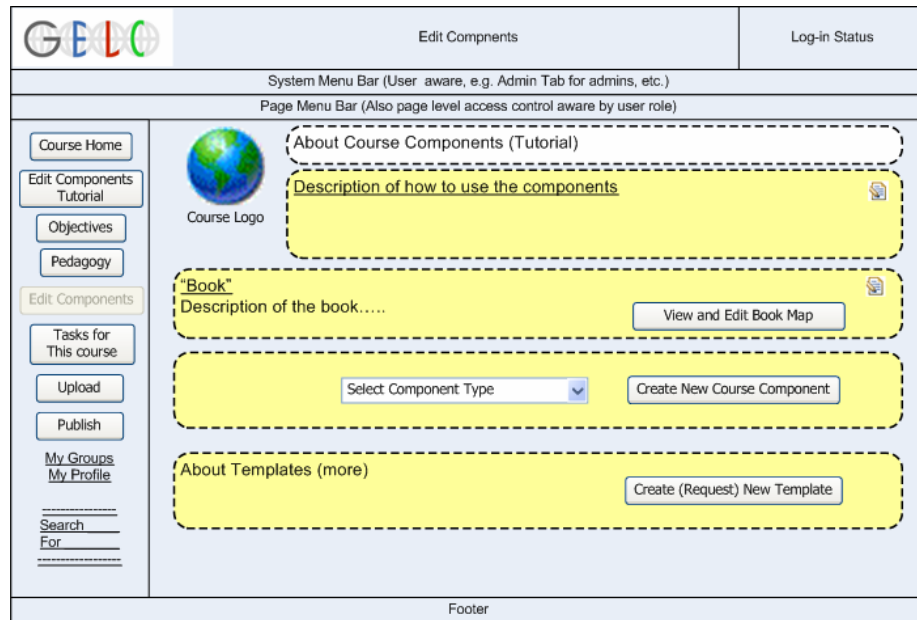


Figure 23- Edit Components page

One other additional point related to deleting components from the course- This is done by editing the Course Components page itself, which might be limited to course owners.

Search and Upload, as mention above, are contextual to the course currently being viewed. As such, any assets chosen from a search result page are by default targeted to be added to the current course collection (Or in the case of assets already in the course collection while editing a unit, should be inserted into the current Unit). Similarly, the “Upload” button brings up a version of the **Contribute Content** pages that assumes that the content is going to the Course Collection so as to be immediately usable in a unit of that course.

The “Tasks for ...” button across all course pages the in **Left Navigating Bar** is also contextual and related to the page of the Course area being viewed (e.g. “Course Tasks”, “Component X Tasks” “Course Framework Tasks” or “Unit Tasks”) and when selected brings-up a pop-up **Task** page with a list of tasks, sorted by Role for the course element in question. Users can add new tasks, check-out tasks (By editing the task and selecting “In Process” and signing it), or complete a task by editing the task and checking it as “Completed.” This task management might be done simply by editing a list page of tasks. This is the rudimentary workflow management capability for this first phase.

When the user selects a component from the **Course Home** page, the user sees the TOC for that component, just like in the Book Map page for that component (It is essentially the same page, but formatter as a sort of home or title page for the component.) However instead of controlling what Units and Templates are included, clicking on a unit brings-up that unit’s page (Which might

contain nested Sub-units). This is the **View, Create & Edit (Component, Unit, Module, and Element)** page. Each unit has its own template and structure (Like an assessment unit is a collection of items as described above.) However all unit templates have two components a Unit Header, where the unit metadata is managed in a form interface, and the Unit Body, which is the content of the unit as it will appear on screen to the end user. Each element and asset in the Unit Body is editable (With Role based access control.) The first set of unit templates should include “Lesson/Tutorial,” “Activity”, “Exercise”, “Assessment”, and “Assessment Item”. The Unit pages templates might look something like this:

Figure 24- View, Create and Edit Module page

This **View, Create and Edit** page is the part of the site most contributors will spend the majority of their time on. One scenario is that the unit was comprised of elements and assets collected from other courses and collections. In this case, the contributors would edit each element to bring the new unit, and its metadata, into a coherent whole by assigning tasks to contributors and progressively editing the assets and adding elements as needed. Perhaps the project is to translate and localize and update a science course from English to Portuguese? In another scenario, an individual has created a new course that is one Component, a Book. They have created a book map, and then upload assets and fill in the resulting blank Unit pages until the new book is complete.

The very last element of the Course area is the “Publish” button, which brings up the **Publish Course** Page. This option may only be available to course owners to avoid unwanted release. Eventually this will provide a wide variety of publishing options, but initially there are two publishing options, “Create PDF” and “Release This Course Version to the Public”. Create PDF creates one PDF rendition (Well styled based on the templates) for each component in the course and stores them in the public repository. The “Release” option copies all the course pages (Current Version) to the public area so users can view the entire course from the web. This process creates a Public Course Home Page, which is a truncated version of the in production course home with the tasks and other development information removed. This will be the page visitors see if they select the course from the **Download, View or Take Course** page in the Entryway. It basically contains the

course description and metadata, and a list of components with descriptions, and a list of contributors and groups (Including any “Peer review and Certification authority” group affiliations.

### **SYSTEM ADMINISTRATION FUNCTIONS**

The design of the global system administration functions will be determined by the capabilities of the platform chosen. However, they must be sufficient to meet the majority of functionality described here including the user, group and role management abilities and the page level conditional access described.

---

## **APPENDIX**

---

### **A. WIKI PLATFORM SCORE CARD**

### **B. PORTAL FRAMEWORK TECHNOLOGY SCORE CARD**

### **C. PARTNER AND COMPONENT TECHNOLOGY SCORE CARD**

### **D. PHASE ONE DRAFT DEVELOPMENT SCHEDULE**

---

## **PARKING LOT**

---

Safety and reporting abuse

Detecting Plagiarism

Copyright take-down notice procedures

Prioritization of features and the overall scope