Toolkit: components

The following is a list of the expected software components that the toolkit will require.

Module name	Functionality	Priority / notes
Login	Main login system • Authentication, requires implementation • Authorisation	Essential
Login::Native	Native cookie implementation	Essential
Login::OpenId	OpenID implementation	Low
User	Notion of a user • Used on both the public and private sides of the site • Settings • Has a username, which is compatible with URLMoniker. • Can have a membership or memberships of a Group (which is the normal case). If more than one, they define their 'primary' Group. Upon login, they are taken to their Group's URL (or a chosen group, if more than one). If not grouped, www. is their subdomain. • User type (probably subclassed): Committee; Member; External • Shield privilege system • Has an avatar (picture) if they wish • Can choose to enable Thread->MailDelivery()	Essential
User::Page	The user's profile page. • Every user's page has a URL which is always available, even if it contains little information, at /users/<URLMoniker>/ (or similar) • Only information they have chosen to show will be shown. • Has an edit mode to edit the profile	

AccountInvitation	⅄ Sends out invites for Group::Member membership ⅄ Expires old invitations	
Issue	A submitted item ⅄ Can be a problem or good practice (or infrastructure/event/other) ⅄ May have a map location attached to it ⅄ Sometimes has a Issue::SphereOfInfluence attached to it rather than just being a point (or perhaps every Issue should have at least a small Issue::SphereOfInfluence for simplicity in implementation) ⅄ Each Issue becomes a Forum::Thread ⅄ Can be Geographical() ⅄ Can be Resolved()	Emulate the CycleStreets addphoto.xml API
Issue::Submission	Submission of an Issue • Mainly for the public side of the website • Also on the private side of the website • Can be submitted by e-mail using MailWebGateway (and is not geographical) • Requires a User account, to prevent spam • Can have ThumbsUp and thumbs-down added, which is one-per-User if logged in, or otherwise uses simple session/IP management to deter misuse • Encourages people to add a map location if relevant (‘becomes geographical’) • Can be become geographical later, if no map at start, by a User who is a Group::Member • Tag/Tag->Subtab() category • Has a title, summarising the problem/example • Has an optional description	
Issue::SphereOfInfluence	The sphere of influence of an Issue. Example use-case is that a particular path must be protected when planning applications for surrounding properties come along. Those planning applications show as points on the map, but the SphereOfInfluence of the path means that a ‘You should be aware of this’ notification appears. • This is effectively a circle drawn round a point, which is then used to see if it	

	intersects with a Boundary • Itself implements Boundary->Circular() • In time this could be expanded to implement GUI controls for Boundary->ThickenedLine() which might also be useful (e.g. so it influences planning applications along a path)	
Thread	Core container for discussions; pluggable – has no native implementation ⅄ Each represents an Issue ⅄ Has an Id() which relates to a public URL, e.g. /issues/<id>/ for a Forum::Thread ⅄ Implements each message/reply as a Thread::Element ⅄ Attachable to many types of objects (e.g. Forum, PrivateMessage ⅄ Web view ⅄ E-mail view, implemented using MailWebGateway, if the User has enabled MailDelivery() ⅄ Has a header which can contain various things ⅄ Implements AccessControlList ⅄ Implements MessageQuoter ⅄ Can set the Issue as Resolved()	
Thread::Header	Core container for summary metadata about a thread; pluggable – has no native implementation.	
Thread::Element	A reply within a thread ⅄ No native implementation – see the Thread::Element::* modules below ⅄ Has an Id() ⅄ Stores parent reply ID ⅄ Container for a set of objects (e.g. text, document with metadata) ⅄ Has a 'Strictly private' property option	ThreadReply – or is that confusing, as not everything is a reply?
Thread::Element::Simple	Standard, simple reply (e.g. text typed into webpage) ⅄ Must be able to handle quotations	
Thread::Element::Photo	Photograph, including metadata	

Thread::Element::StreetViewDisplay	An instance of a StreetViewDisplay	
Thread::Element::MapLocation	A location on a map - OpenStreetMap as the default map style - Probably using OpenLayers, or perhaps LeafletJS	
Thread::Element:GeolocatedImage	An image with an associated map location	Spec is still unclear whether this can be a primary ID or is effectively a thread-specific item
Thread::Element::ChangeNotification	A notification that an element of the Thread->Header() has been changed - At least title, e.g. “User X has changed the title to …”	
Thread::Element::Weblink	A link to an online newspaper or blog, or general web link - Should attempt to store some semantic structural information, e.g. could capture the favicon so that the external site is easily recognisable - Requires a description to be added - Can be multiple links, each with a description and associated metadata, using Thread::Element::BucketContent - Implements UrlDiscoverMetadata	
Thread::Element::Document	A document, or set of documents. - Implementation uses Thread::Element::BucketContent to enable more than one item in this bucket	
Thread::Element::Tweet	A tweet URL - Captures information via the Twitter API about this tweet and the user who posted it - Implements UrlDiscoverMetadata	
Thread::Element::AttachedEmail	An e-mail attached. - Could be multi-part, so again uses Thread::Element::BucketContent for	

	implementation	
Thread::Element::CommentFromPublic	A comment from a member of the public who has added a comment to an item on the public submission side. - These comments appear in a corner of the page and can be 'dragged into' the thread via some GUI means. If not dragged in, they stay in this corner of the page.	
Thread::Element::BucketContent	Implementation of a single item (e.g. link with metadata) that forms one of potentially several items in a budget such a Thread::Element::Weblink container.	
Thread::Element::ExternalPersonAddedNotification	A notification that an external person has been added to the thread.	
Thread::Element::InfoBlock	A reference (link / expandable box) to an info block. - References InfoBlock but does not modify it - Comment can be added by the User, e.g. “The second link is most relevant here” - Different types of info block (e.g. contact name, best practice resource, cost) may be displayed differently	
Thread::Element::PolicyStatement	Display of an official policy position of the group	Not clear yet if this should be implemented as an info block or just text that a member can type in
Thread::Element::CallToAction	A pre-formatted invitation (e.g. “Someone please agree to draft a letter”) that people can accept. - A set of pre-formatted texts should be available (i.e. implements AutoComplete)	
Thread::Element::EnsemblingDocument	Integration with www.ensembling.com - Finalisation of a letter, triggered by an API call (direction unclear at present) should result in the comments being imported into the toolkit. - Reminds people importance of seeking Committee approval for letters	API under discussion with Ensembling's creator

Thread::Element::GoogleDocument	Integration with Google Docs ⅄ Also reminds people importance of seeking Committee approval for letters	
Thread::Element::FinalisedLetter	A statement that a finalised letter has been sent, and the letter itself	
Thread::Element::ProposedCampaignObjective	A proposed campaign objective	Is this necessary – will people bother to use this rather than a standard text reply and how is it used?
Thread::Element::Deadline	A deadline or set of deadlines. ⅄ Each deadline has a time ⅄ Each deadline has a description associated with it	
Thread::Element::Poll	A reference to a Poll	
Thread::Element::Petition	A reference to a Petition	
Thread::Element::Questionnaire	A reference to a Questionnaire	
Thread::Element::AchievedObjective	A statement of achievement of a campaign objective, e.g. that a cycle parking location has been added. ⅄ This will also show as visible text on the public map view, where it was originally submitted. ⅄ Should ask whether the user wants to create a CaseStudy (Thread::Element::CaseStudy)	
Thread::Element::SupersededPhoto	A photo which officially supersedes the original photo in depicting the new situation on the ground.	Is this a subsection of a Thread::Element::AchievedObjective
Thread::Element::CaseStudy	Reference to a ‘Case study’ entry (which is a type of InfoBlock) for the information of other	

	groups.	
Forum	Core discussion area, which users will spend most time in ⅄ Has a top-level URL like /issues/ ⅄ Makes available the submitted issues within the Group’s Forum::DefaultListedItems->Problems() ⅄ Various filtering methods by Prioritisation or Tag/Tag->Subtab() category ⅄ Viewable as Forum::Map or Forum::Listing	
Forum::DefaultListedItems	The items that are ordinarily shown in the Forum to the Users of a Group. • Implements the Boundary in the Group’s Group::Settings • Includes other items that a Group::CommitteeMember has added in • Splittable as Problems()and GoodPractice()	
Forum::Map	Shows all threads that are Thread->Geographical() • Different icons for different categories • Also available on the public side – basically shows Issues submitted • Omits Issue->Resolved() items by default	Categories vs tags issue unresolved in the spec
Forum::Thread	Thread within a forum ⅄ Each thread Implements Thread ⅄ Has a Forum::Thread::URL ⅄ Has a Tag (and Subtag) ⅄ Thread->Header() implemented by Forum::Header	
Forum::Thread::URL	The URL of a thread. ⅄ Has a URL like /issues/<id>/. ⅄ This will display for any subdomain, even if not in the Group's Group::Boundary ⅄ The Group::Url determines the content of the page (e.g. the discussions in the Thread, or a statement that the group isn't working on it). ⅄ For www.<toolkitdomain> this show the issue, but instruct the user that they need to go to a local site, with their sites listed and an indicator showing whether the group is working on this issue (i.e. whether it contains an opening Thread::Element	Not clear yet whether the www. address version should show all groups working on the issue (which breaks group confidentiality) or just those groups that the

		User is a member of
Forum::Header	An implementation of Thread::Header. Contains key details about the Forum's thread ⅄ Implements Thread->Header() ⅄ Has a Title, description ⅄ Shows the Tag/Subtag ⅄ Time/date submitted ⅄ Button for subscribing/unsubscribing which implements Forum::Subscription ⅄ Button showing who is subscribed ⅄ A StreetViewDisplay ⅄ A display of Group(s) dealing with this thread, including their logo ⅄ The number of public ThumbsUps, showing thumbs up/down indications from public users ⅄ Prioritisation controls ⅄ A link to WriteToThem ⅄ A lead User for the issue	
Forum::Subscription	Subscriptions of users ⅄ Links a User or set of Users to a set of Forum->Id()s ⅄ Can subscribe or unsubscribe a User ⅄ Can return a set of subscribed Users ⅄ Can display which subscriptions are external users rather than members	
PrivateMessage	Implements Thread ⅄ Is between two Users	
Tag	Thematic category ⅄ Has a Tag (category) (e.g. cycle parking) ⅄ Optionally has a Subtag (sub-category) (e.g. sheffield stands) ⅄ Tag/Subtag means that a hierarchy can be created ⅄ Category and Subcategory have a URLMoniker and a visible name ⅄ Enforces uniqueness across URL moniker	

URLMoniker	Class to validate URL monikers. These include: ⅄ Category (e.g. 'cycleparking') ⅄ Username (e.g. 'Martin'), which uses the same rules as YouTube/CycleStreets ⅄ Group name (e.g. ‘camcycle’). ⅄ Monikers may have reserved names (e.g. www for a Group name) which prevents those being created by the calling class.	
Group	A campaign group, such as “Cambridge Cycling Campaign”, “CTC” [a national group] ⅄ Groups have a Group::Boundary ⅄ A group's boundary can overlap with another's – which is particularly likely in some cases ⅄ Have many Group::Member ⅄ Groups (or, strictly speaking, their members) cannot see any other group's data, with the exception of (i) InfoBlocks that are national and (ii) Threads that have been explicitly shared. The general concept is that the toolkit is an internal management tool for the group. ⅄ Have one or more Group::CommitteeMember, with at least one always existing ⅄ Has a Moniker (e.g. 'camcycle' for Cambridge Cycling Campaign, 'ctc', 'lcc', etc.) which is lower-case letters a–z only. (Possibly hyphen – characters may be acceptable). This is not editable by the group, but the system administrators should be able to change this if necessary. ⅄ Has a Group::Url ⅄ Has a Group::GuiStyle ⅄ Has a settings area implemented by Group::Settings ⅄ Have a logo ⅄ Have contact details, a main website URL, Twitter link, Facebook link, and other things that ordinary members will expect to have available for easy reference ⅄ Have a Group::AboutPage which shows the various details of the group ⅄ Committee members of the Group can issue an AccountInvitation	
Group::Member	A User who is a member of a Group	

	• Member approval policy being determined by the Group::Settings, with the User usually requiring approval by a Group::CommitteeMember	
Group::CommitteeMember	A committee member of a Group. ⅄ Is a User ⅄ Is associated with a specific Group ⅄ An active group always has at least one Committee member ⅄ Has rights to change settings and various other PrivilegedOperations ⅄ Committee members can approve new people subscribing. ⅄ Can add additional items into the Group’s Forum::DefaultListedItems	Question of managing an 'inactive' group.
Group::ApplicationForm	An application form to create a Group, which a group uses to register themselves ⅄ Wizard-like approach ⅄ Applications are checked by CycleStreets initially; later it is likely that there is no such approval stage ⅄ Form makes clear this is a system for campaign groups, and that others (e.g. cycle training providers) may find it unsuitable ⅄ Is created by a User, who becomes the first Committee member of the Group	
Group::Settings	The settings of a Group. ⅄ Various control panels for editing the Group's settings ⅄ Editable only by a User who is a committee member ⅄ Has privacy controls ⅄ Has a policy determining whether changes require the approval of a majority of Committee members or otherwise	Change policy system needs further explanation at spec level
Group::Url	The server name and base URL of a group. Examples are: ⅄ <Moniker>.<toolkitdomain> (the default) ⅄ toolkit.camcycle.org.uk (where the group has pointed a CNAME at the toolkit server) ⅄ www.camcycle.org.uk/toolkit/ (where the group has set up a reverse proxy to the server) ⅄ Only one of these exists at any one time.	CycleStreets has a solid implementation of this which could be translated.

	⅄ Where a group’s moniker has changed, or the moniker has been changed to e.g. a CNAME subdomain, the user will be redirected to the current one ⅄ www is a special-case which is a generic ungrouped view of the site. www will not be available as a group moniker.	
Group::GuiStyle	A header and footer which determines the style. ⅄ By default the native toolkit style will be used ⅄ Is editable by any Group::CommitteeMember	
Group::AboutPage	A page about the group. ⅄ Have contact details, a main website URL, Twitter link, Facebook link, and other things that ordinary members will expect to have available for easy reference ⅄ Shows the list of Committee members ⅄ Has a link to each User who is a member, showing their avatar if they have enabled it, and a link to their User::Page	
Group::Boundary	A boundary, or set of boundaries for the group. ⅄ Implemented using Boundary	
Group::MembershipImport	Imports a supplied membership list that a Group::CommitteeMember supplies ⅄ Webfors, accepting textarea text or a CSV ⅄ Triggers AccountInvitation to send out notifications ⅄ Stores hash of e-mails, which mean that a newly-registering member can get automatically approved into a Group rather than having to wait for post-approval ⅄ Can be re-uploaded, which replaces the previous list and locks an active User if they are no longer present in the membership database. ⅄ Originally-uploaded (non-hashed) data is discarded	
StreetViewDisplay	A Street View (3D) display ⅄ Implements the Google Maps Street View API ⅄ Could implement Bing's equivalent in future	
ThumbsUp	Capturing and display of the number of thumbs-ups and thumbs-downs by external users ⅄ Needs a session security implementation to deter multiple hits	

	⅄ Possibly may need to store the IP ⅄ Should reference a User account if the person is logged-in	
Prioritisation		
InfoBlock	Note national or local	
AutoComplete	A helper module which is aware of previously-entered texts in various contexts, and is able to auto-suggest to reduce the amount of typing	
Poll		
Petition		
Questionnaire		Low priority for implementation
UrlDiscoverMetadata	A helper class which us able to extract metadata from a given URL. ⅄ Will have a registry of known cases that can be easily added to. ⅄ Twitter example: Will obtain details of the Twitter user and their avatar via the Twitter API ⅄ General link example: Will extract the site's favicon	
Boundary	A module which enables the setting and querying of the boundary or boundaries of a group. • Types of boundary are: LocalAuthority(); Square(); Circular() • Also a type ThickenedLine() in future, which 'thickens' a route line, effectively creating a curvy box around it • Note that occasionally a Local Authority boundary can have a hole in it (e.g. Cambridge is completely surrounded by South Cambs) • Can determine whether a point is within a boundary	

	• Can determine whether an Issue::SphereOfInfluence intersects with a boundary	
PrivilegedOperations	Privileged operations are only available to Committee members. ⅄ Privileged operations are shown with a shield icon (or similar indicator) ⅄ Privileged operations generally trigger a Notification to other Committee members	
Notification		
AccessControlList	An important module which governs the accessibility of a Thread to a User, based on their membership (or otherwise) of a Group ⅄ Group members can see	
MailWebGateway	A component which enables a Thread, and specifically a Thread::Element to be received by e-mail in additional to web-based viewing. ⅄ Means that users who prefer e-mail can use the toolkit like a standard e-mail list (but actually being lots of mini, issue-specific e-mail lists rather than monolithic topic/discussion lists). ⅄ Works in both directions: mail-to-web and web-to-mail. ⅄ Uses Thread::Element::* implementations of e-mail equivalent content. For instance, display of a document and description on the web would have as its parallel an e-mail delivered as text with the document attached. ⅄ In mail-to-web direction, an incoming message becomes a forum posting, with the content being processed in the most suitable way ⅄ Uses MailWebGateway::EmailId ⅄ Uses Thread::Header (e.g. to generate the subject line)	High
MailWebGateway::EmailId	The e-mail ID of a message sent out by MailWebGateway. ⅄ Has an ID which is not guessable ⅄ Associated with a Thread::Element->Id() but is not the same as this ID ⅄ Similar to Facebook's 'Reply by e-mail to add a comment' system. ⅄ Enables a response to be unambiguously associated with a thread and which message it is in reply to	Spec question of allowing topic@domain to auto-create a thread. Use mail headers instead?
MessageQuoter	A helper module to work out which parts of a message are quoted, and which are new text.	

	⅄ Has the ability to reject a message where very large amounts of text are quoted. The rejection is thrown back to the calling class, so that the forum gives the user a warning that they need to quote better, or an e-mail with MailWebGateway is rejected with an explanatory message.	