How to Read this Document

This document is divided into sections that showcase the potential functions of the Time Machine system. This document is intended to showcase the functional specification of the system, and also provides some design directions for how the system might look to an end user.

All of the wireframes contain annotations that are found at the bottom of each page. These annotations describe the functions of the features diagramed.

Each of these designs is subject to change based on the technical needs and implementation details that will inevitably arise as a part of the development process. Further, designing an "ideal" system means that many of the features described may not be present in the final product. Features are categorized in the following way to provide guidance to the reader:

🟢 Will be present in the initial release

🟡 Will be present in a future release (phase 2)

🔴 Currently not planned for a release (phase 'next')

What is Time Machine?

Time machine is a form of content staging that blends into the publishing workflow. The system is intended for content and layout editors to be able to view a preview of the work that they're doing "in context" with other parts of the site. The project involves the following major parts:

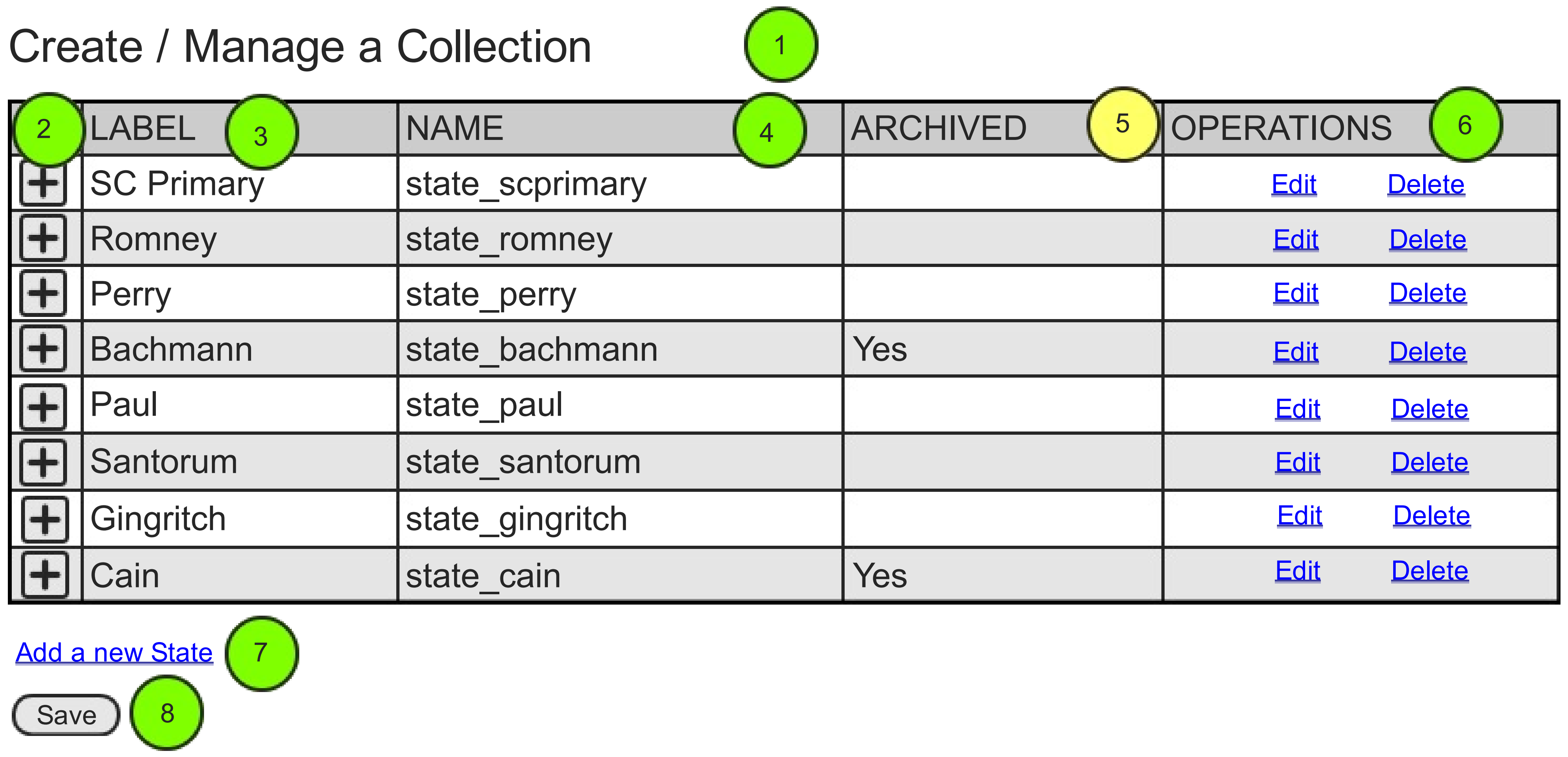
1. Collections and administration
2. Workflow administration
3. States UI for content entry
4. Time Machine UI

About this section:

This section describes how collections are defined within the system. Collections are a way of collecting content into a single "what-if" publishing scenario. By tagging a piece of content with a collection, a user is saying that they want to be able to stage that content and be able to view it as of a specific date.

The need for effective dating of collections seems reasonable, but more technical investigation and user stories need developed to explore this context. Regardless, when content is tagged with a collection, it is considered "staged", even if final publication remains unscheduled.

Section Contents:



Create / Manage a Collection

	LABEL	NAME	ARCHIVED	OPERATIONS	
+	SC Primary	state_scprimary		Edit	Delete
+	Romney	state_romney		Edit	Delete
+	Perry	state_perry		Edit	Delete
+	Bachmann	state_bachmann	Yes	Edit	Delete
+	Paul	state_paul		Edit	Delete
+	Santorum	state_santorum		Edit	Delete
+	Gingritch	state_gingritch		Edit	Delete
+	Cain	state_cain	Yes	Edit	Delete

Add a new State

Notes:

1. This admin screen allows you to define which collections are available in the system. This assumes a system similar to managing "fields" in Drupal 7.

2. Collections should be able to be reordered. This order should affect their order in lists and assign a "weight" to the collection.

3. Users should be able to define a label for a collection.

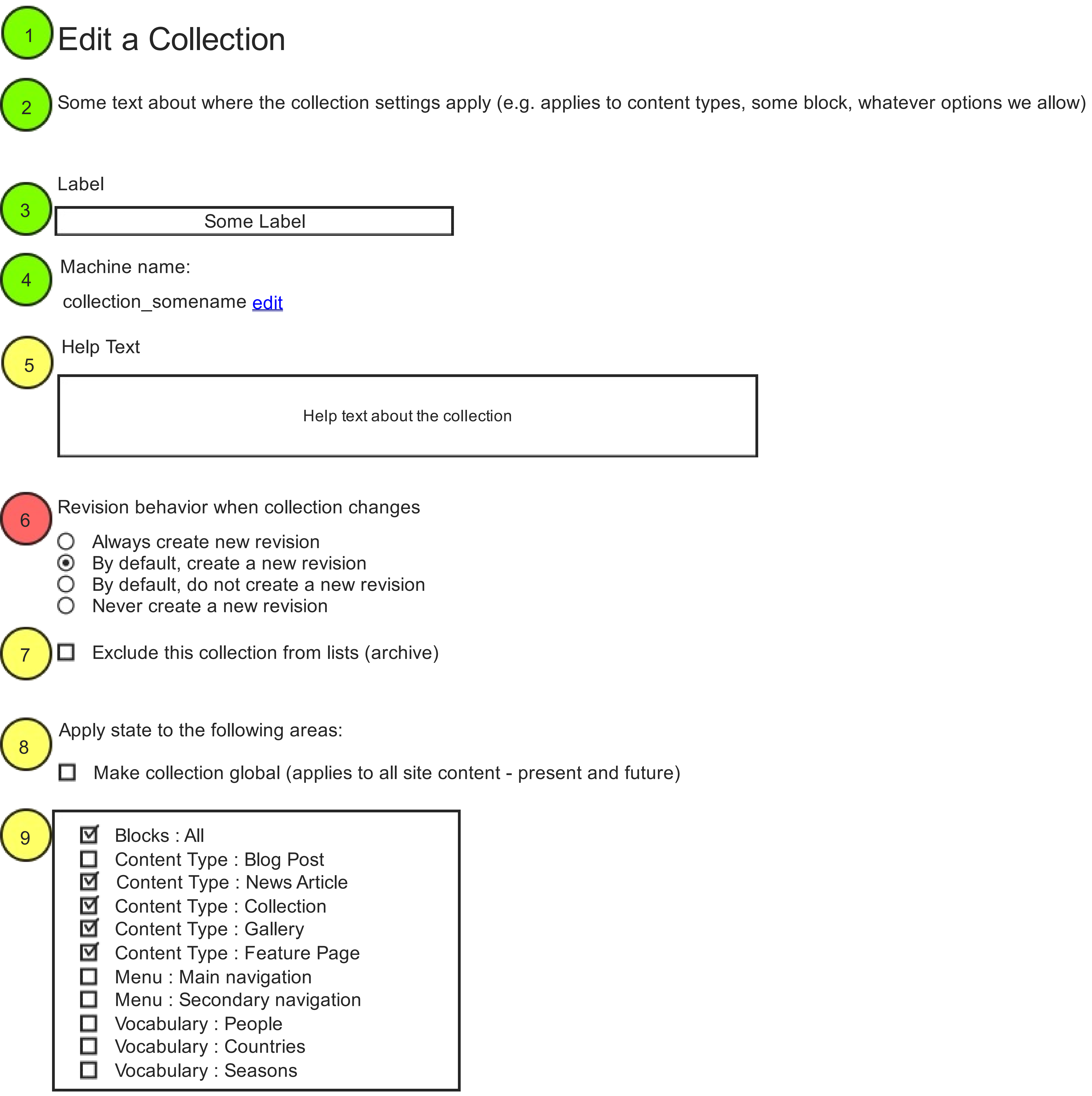
4. A machine name is auto-generated for a collection that can be edited if necessary.

5. Archive means that this collections wil no longer show up in lists. This is a way of preserving collections that are no longer valid.

6. Editing and deleting should function something like editing and deleting within the fields interface. Edit is diagramed in "Edit a State".

7. A link to adding / editing a collection. This is the same form as "Edit a Collection". Deleting removes the collection from the list and from the system.

8. A save button that saves changes to this form.



1 Edit a Collection

2 Some text about where the collection settings apply (e.g. applies to content types, some block, whatever options we allow)

3 Label

Some Label

4 Machine name:

collection_somename edit

5 Help Text

Help text about the collection

6 Revision behavior when collection changes

- ○ Always create new revision
- ● By default, create a new revision
- ○ By default, do not create a new revision
- ○ Never create a new revision

7 ☐ Exclude this collection from lists (archive)

8 Apply state to the following areas:

☐ Make collection global (applies to all site content - present and future)

9
- ☑ Blocks : All
- ☐ Content Type : Blog Post
- ☑ Content Type : News Article
- ☑ Content Type : Collection
- ☑ Content Type : Gallery
- ☑ Content Type : Feature Page
- ☐ Menu : Main navigation
- ☐ Menu : Secondary navigation
- ☐ Vocabulary : People
- ☐ Vocabulary : Countries
- ☐ Vocabulary : Seasons

Notes:

1. Any time a collection is added or edited a user is presented with this form. The only difference is that when editing an existing collection the form is pre-populated.

2. Basic text to guide the user through the form.

3. A user-defined label for the collection. This will be the label that appears throughout the system.

4. A machine-readable name for the collection. This is atuo-populated, but the user may change it if they click the edit link.

5. Help text that the user may define for the collection if desired. There are no current plans to have the help text appear elsewhere in the system at this time.

6. Revision options for the collection. Currently, Drupal has some difficult technical limitations around revisions, so each change will create a new revision of a node. Future functionality may include optional revision management if it is a desired feature.

7. This checkbox creates an "archive" collection. This means that the collection would no longer be available for addition to content, but content with the collection would stil remain tagged with it.

8. Each collection can be applied to various parts of the system. This includes different content types or site components (e.g. blocks and menus). By default, the collection will apply globally.

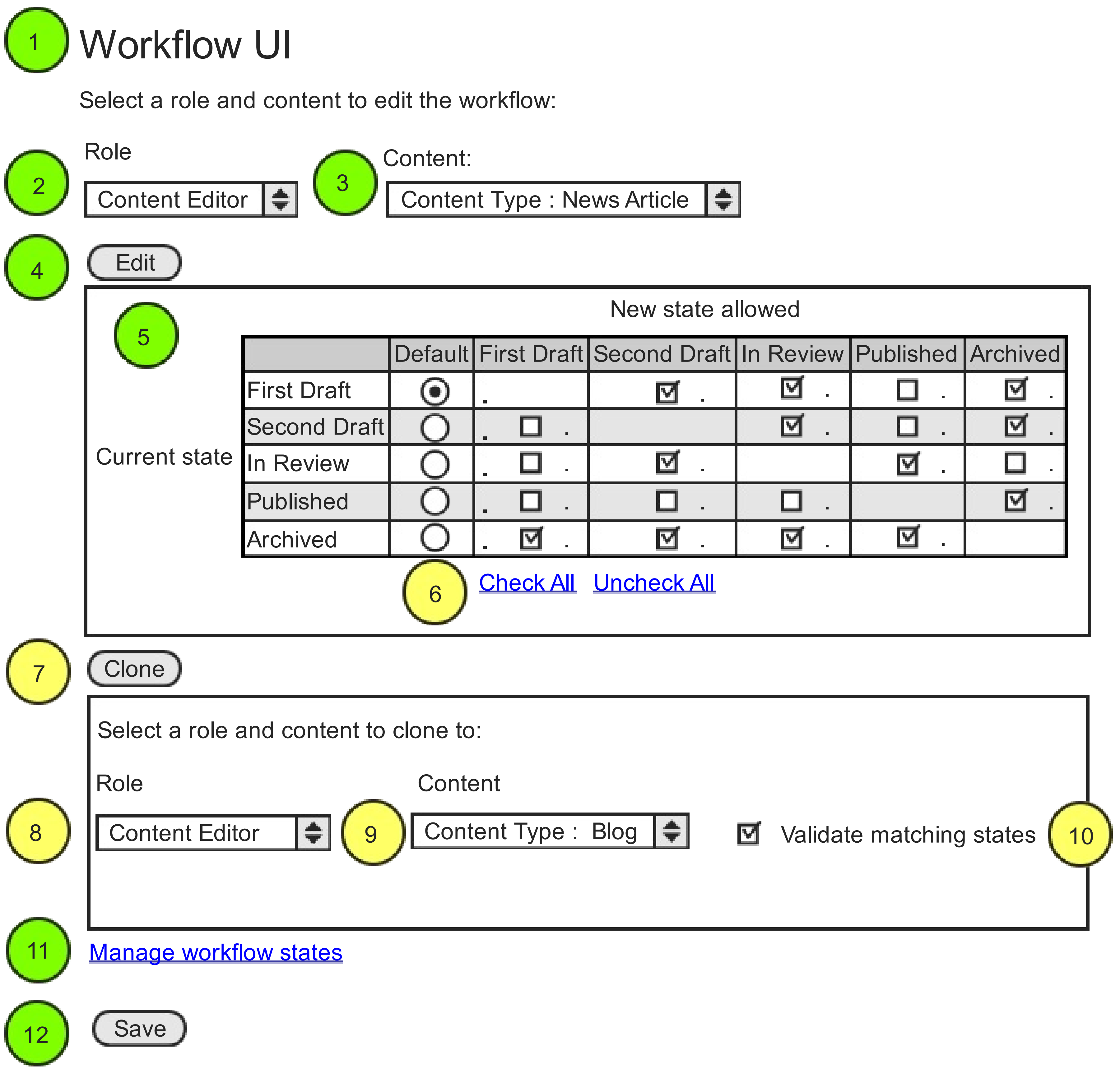
9. If the "make collection global" checkbox is un-checked, a list of available entities and site components are listed. Any of these that is checked will have that collection available to it.

About this section:

This section describes how workflow states are created and moderated by the system. Workflow defines the status of the content within the publication cycle and is able to be moderated on a type-by-type basis.

Defining a workflow takes into account both the role and the content for which the workflow is being defined. Further, moving through a workflow triggers actions that can be used to perform other tasks.

Section Contents:



Workflow UI

Select a role and content to edit the workflow:

2 Role: Content Editor

3 Content: Content Type : News Article

4 Edit

5

		New state allowed					
		Default	First Draft	Second Draft	In Review	Published	Archived
Current state	First Draft	●		☑	☑	☐	☑
	Second Draft	○	☐		☑	☐	☑
	In Review	○	☐	☑		☑	☐
	Published	○	☐	☐	☐		☑
	Archived	○	☑	☑	☑	☑	

6 Check All Uncheck All

7 Clone

Select a role and content to clone to:

8 Role: Content Editor

9 Content: Content Type : Blog

☑ Validate matching states 10

11 Manage workflow states

12 Save

Notes:

1. The workflow UI is similar to that of Workbench Moderation. Users should be able to determine how content moves through a publication workflow based on the role of a user and the content that they're moderating.

2. This is a drop-down of all roles in the system.

3. This is a drop down of all content and components in the system. It includes things like each content type, each vocabulary, each menu, blocks, etc

4. Clicking edit generates the table in the box below that shows the workflow options for the given role / content pair.

5. The table is a clickable listing of all valid content options. The "current state" show valid predecessors, the "new state allowed" shows valid successors. Users should also be able to choose one default workflow state that will be automatically assigned via the radio button list.

6. Check all / uncheck all controls should be provided for convienence to the user. Not essential to function.

7. The ability to clone workflows would be a useful feature for copying between different roles / content combinations. This button would bring up the information in the box below.

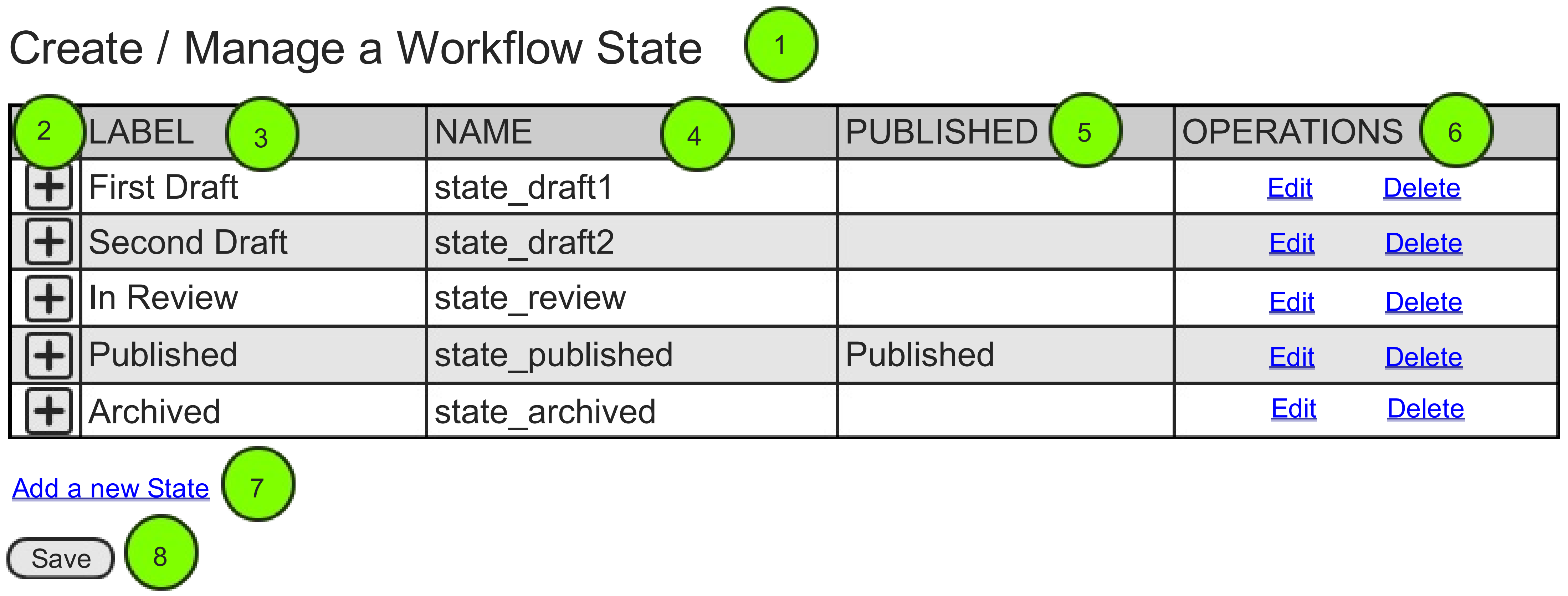
8. Similar to #2, this is a list of all roles in the system.

9. Similar to #3, this is a list of all content and components in the system.

10. This checkbox allows your to validate that the states allows in the destination role / content pair exactly match the states as defined in the source role / content pair (as you can define different valid states for different types of content and site components). Without an exact match the system will notify the user an not copy anything. If this box is unchecked the system will copy only those states that are valid in both the source and destination pairs.

11. Manage workflow states is a link to the screen diagramed in "State Admin (workflow).

12. Save finalizes any changes made on this page.



Create / Manage a Workflow State

	LABEL	NAME	PUBLISHED	OPERATIONS
⊞	First Draft	state_draft1		Edit Delete
⊞	Second Draft	state_draft2		Edit Delete
⊞	In Review	state_review		Edit Delete
⊞	Published	state_published	Published	Edit Delete
⊞	Archived	state_archived		Edit Delete

Notes:

1. This admin screen allows you to define which context states are available in the system. This would be available from a configuration menu for context states.

2. States should be able to be reordered. This order should affect their order in lists and assign a "weight" to the state.

3. Users should be able to define a label for a state.

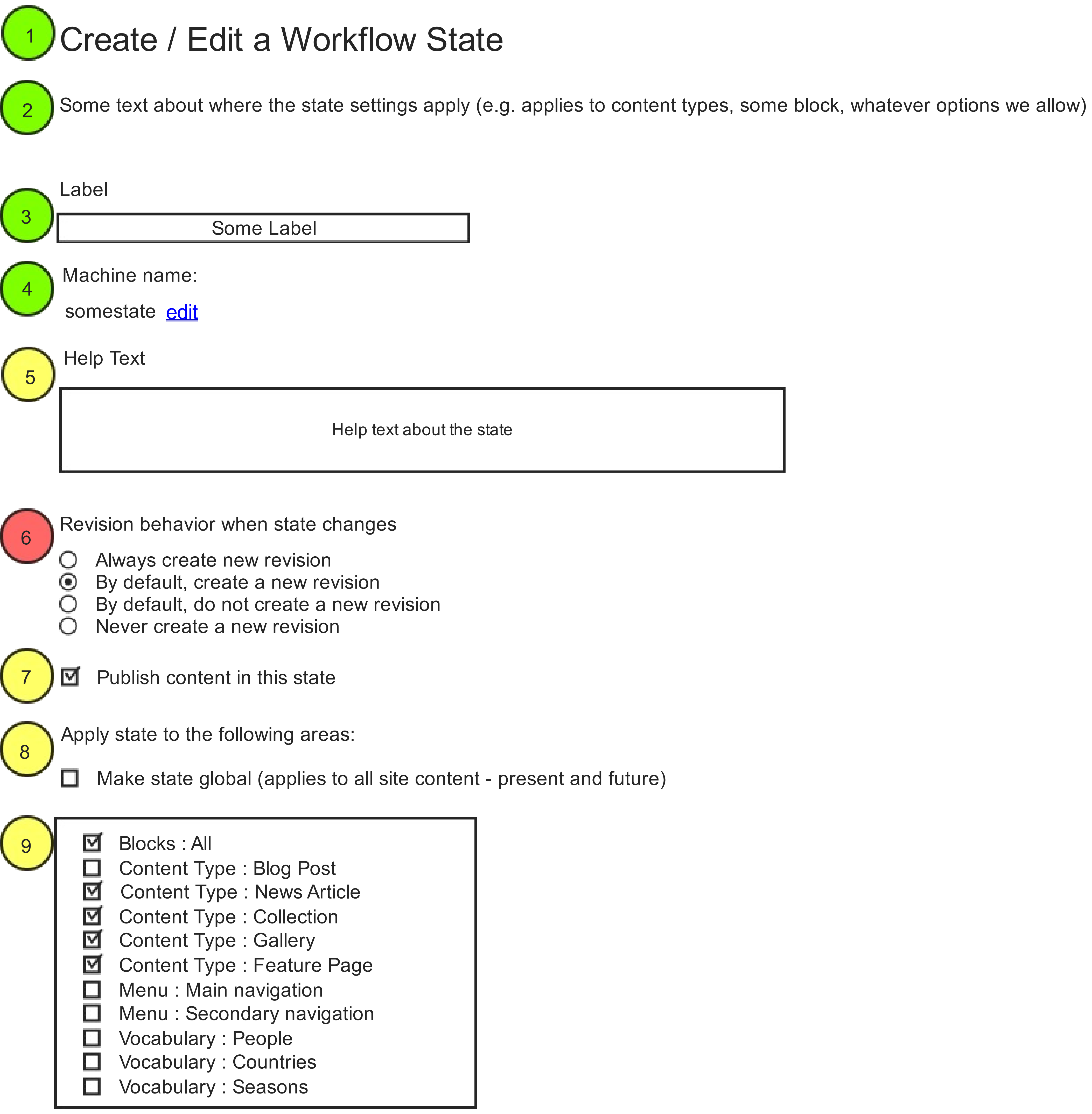
4. A machine name is auto-generated for a state that can be edited if necessary.

5. This column denotes if the state will cause a piece of content to be published. This option can be changed by editing the state.

6. Editing and deleting should function something like editing and deleting within the fields interface. Edit is diagramed in "Edit a State".

7. A link to adding / editing a state. This is the same form as "Edit a State". Deleting removes the state from the list and from the system.

8. A save button that saves changes to this form.



1 Create / Edit a Workflow State

2 Some text about where the state settings apply (e.g. applies to content types, some block, whatever options we allow)

3 Label

Some Label

4 Machine name:

somestate edit

5 Help Text

Help text about the state

6 Revision behavior when state changes

- Always create new revision
- [✓ By default, create a new revision]
- By default, do not create a new revision
- Never create a new revision

7 [✓ Publish content in this state]

8 Apply state to the following areas:

Make state global (applies to all site content - present and future)

9
- [✓ Blocks : All]
- Content Type : Blog Post
- [✓ Content Type : News Article]
- [✓ Content Type : Collection]
- [✓ Content Type : Gallery]
- [✓ Content Type : Feature Page]
- Menu : Main navigation
- Menu : Secondary navigation
- Vocabulary : People
- Vocabulary : Countries
- Vocabulary : Seasons

Notes:

1. Any time a context state is added or edited a user is presented with this form. The only difference is that when editing an existing state the form is pre-populated.

2. Basic text to guide the user through the form.

3. A user-defined label for the state. This will be the label that appears throughout the system.

4. A machine-readable name for the state. This is atuo-populated, but the user may change it if they click the edit link.

5. Help text that the user may define for the state if desired. There are no current plans to have the help text appear elsewhere in the system at this time.

6. Revision options for the state. Currently, Drupal has some difficult technical limitations around revisions, so each change will create a new revision of a node. Future functionality may include optional state management if it is a desired feature.

7. This checkbox determines if the workflow state will publish the content when that state is applied.

8. Each state can be applied to various parts of the system. This includes different content types or site components (e.g. blocks and menus). By default, the state will apply globally.

9. If the "make state global" checkbox is un-checked, a list of available entities and site components are listed. Any of these that is checked will have that state available to it.

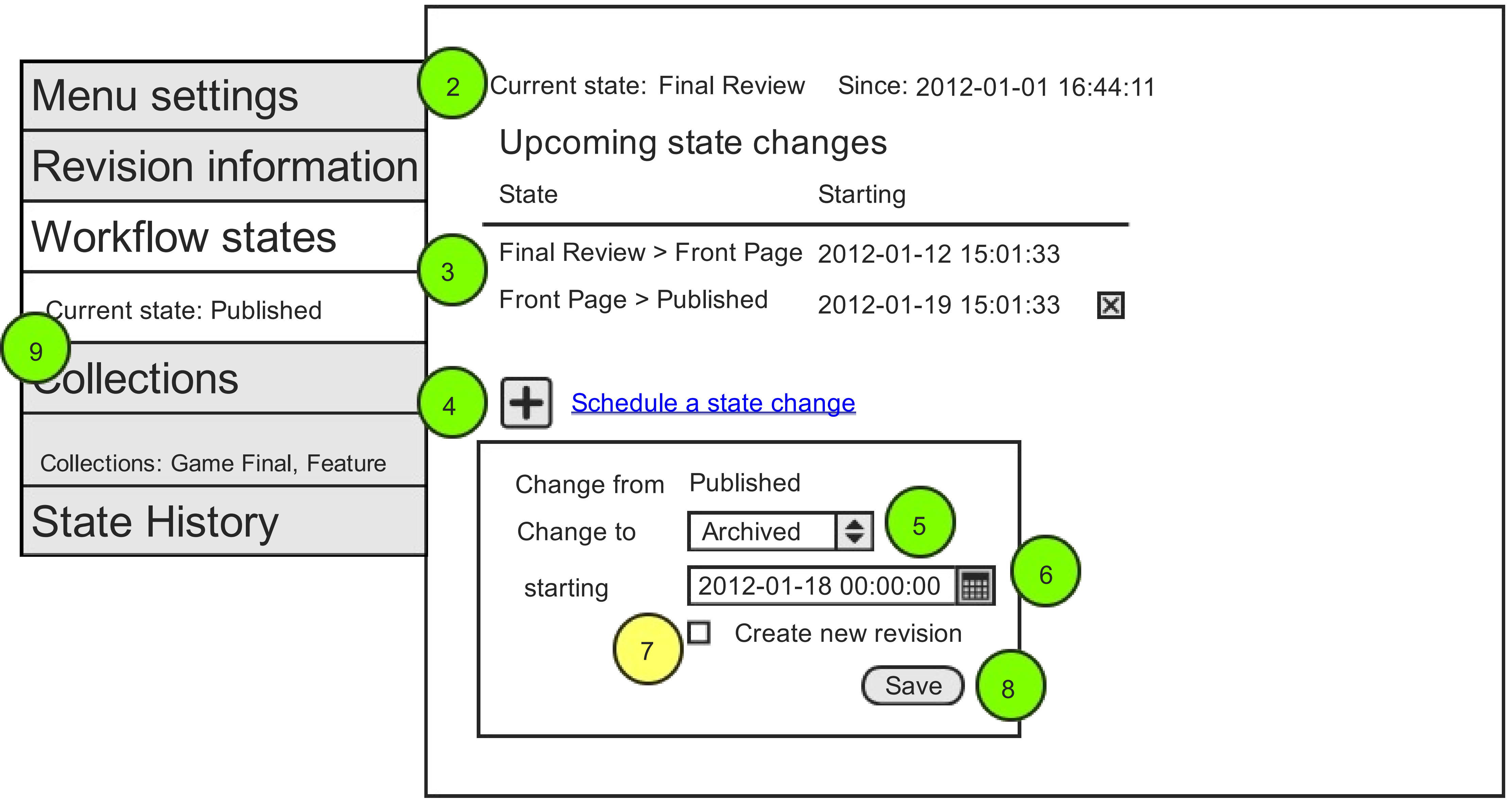
About this section:

This section describes how workflow states are created and moderated by the system. Workflow defines the status of the content within the publication cycle and is able to be moderated on a type-by-type basis.

Defining a workflow takes into account both the role and the content for which the workflow is being defined. Further, moving through a workflow triggers actions that can be used to perform other tasks.

Section Contents:

Editing states on entities / site components



Menu settings

Revision information

Workflow states

Current state: Published

Collections

Collections: Game Final, Feature

State History

Current state: Final Review Since: 2012-01-01 16:44:11

Upcoming state changes

State	Starting	
Final Review > Front Page	2012-01-12 15:01:33	
Front Page > Published	2012-01-19 15:01:33	☒

Schedule a state change

Change from	Published
Change to	Archived
starting	2012-01-18 00:00:00

☐ Create new revision

Save

Notes:

1. When editing / creating a piece of content (e.g. a node), one of the vertical tabs should be associated with a workflow state. If no vertical tabs exist this information should be embedded in the creation / edit screen for that component.

2. Information on the current workflow including the state that the content is in, and when that state began.

3. A listing of future scheduled states for the content. This includes the ability to remove the last scheduled state by clicking the "x". You have to remove states one at a time starting from the last one to preserve date integrity.

4. Clicking the plus icon or the link brings up the information in the box about scheduling a new state change.

5. A drop-down of valid states based on the current state as defined in the Workflow UI.

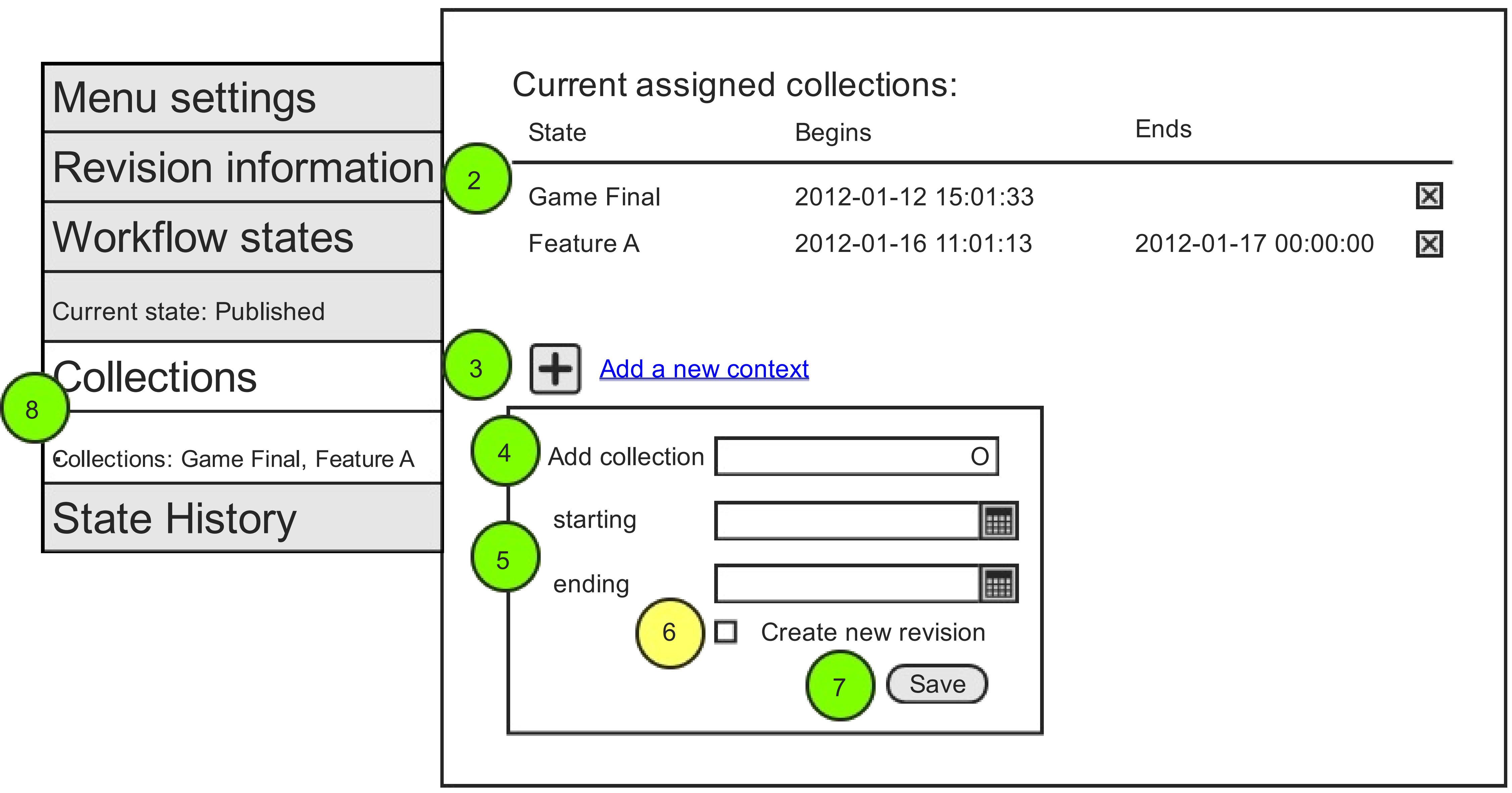
6. A date picker control determining when the next state will begin. The system must validate that the date is after the last scheduled workflow change (meaning the start date is last on the list).

7. Creating a new revision will happen automatically if this box is checked. The first phase of this project doesn't include this fine-grained revision management, so all state changes will create a new revision.

8. This button saves the scheduled state change and adds it to the bottom of the upcoming state changes list.

9. The vertical tab links should contain information about the current state of the node without requiring you to open the tab. This is similar to other Drupal functionality when using vertical tabs.

Editing states on entities / site components



Current assigned collections:

State	Begins	Ends	
Game Final	2012-01-12 15:01:33		☒
Feature A	2012-01-16 11:01:13	2012-01-17 00:00:00	☒

Add collection	
starting	
ending	
☐ Create new revision	

Notes:

1. When editing / creating a piece of content (e.g. a node), one of the vertical tabs should be associated with a collection. If no vertical tabs exist this information should be embedded in the creation / edit screen for that component.

2. Information on the current collections applied to the content should be displayed along with their effective dates. A user should be able to remove any state by clicking the "x" button.

3. Clicking the plus icon or the link brings up the information in the box about adding a new collection.

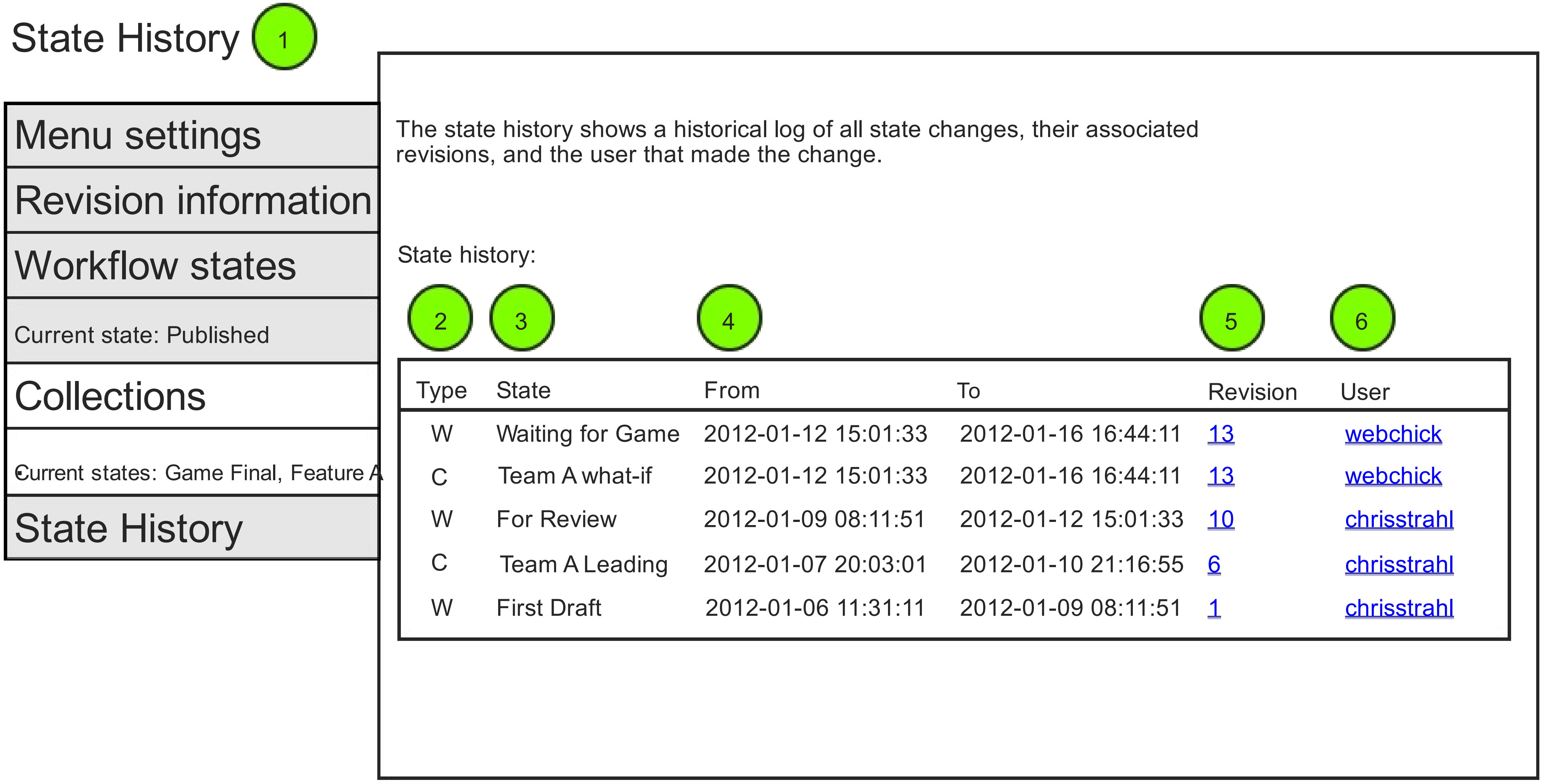
4. An autofilling text-box that looks up valid collections for this piece of content as you type. To create a new valid collection you can enter a collection that isn't found. This creates the collection with a set of basic default parameters that can be edited later if a user desires.

5. A date picker control determining the effective dates for the collection. The collection will take effect on the starting date and that effect will end on the ending date.

6. Creating a new revision will happen automatically when the collection is applied if this box is checked. The first phase of this project doesn't include this fine-grained revision management, so all collection changes will create a new revision.

7. This button saves the scheduled collection change and adds it to the assigned collection list.

8. The vertical tab links should contain information about the current collections of the node without requiring you to open the tab. This is similar to other Drupal functionality when using vertical tabs.



State History 1

Menu settings

Revision information

Workflow states

Current state: Published

Collections

Current states: Game Final, Feature A

State History

The state history shows a historical log of all state changes, their associated revisions, and the user that made the change.

State history:

2 3 4 5 6

Type	State	From	To	Revision	User
W	Waiting for Game	2012-01-12 15:01:33	2012-01-16 16:44:11	13	webchick
C	Team A what-if	2012-01-12 15:01:33	2012-01-16 16:44:11	13	webchick
W	For Review	2012-01-09 08:11:51	2012-01-12 15:01:33	10	chrisstrahl
C	Team A Leading	2012-01-07 20:03:01	2012-01-10 21:16:55	6	chrisstrahl
W	First Draft	2012-01-06 11:31:11	2012-01-09 08:11:51	1	chrisstrahl

Notes:

1. The state history is a way of viewing the historical states and associated information. It's a log of both the collections and the workflow states for reference purposes.

2. The type of state is listed in the table. "W" for workflow, "C" for collection.

3. The state label is displayed.

4. The effective dates of the state are displayed for all states but the current state

5. A link to the revision that was created when the state was changed.

6. A link to the user profile of the person that made the state change.

Editing states on nodes / entities

Current state: Published, Game Final, Featured

Edit workflow state   Edit context state

Workflow State:

Current state: Published   Started: 2012-01-16 16:44:11

Change to [Archived] starting [2012-01-18 00:00:00]

Schedule another state

Collections:

Game Final	starting	2012-01-16 16:44:11	ending	
Featured	starting	2012-01-17 00:00:00	ending	2012-01-18 00:00:00

Add another context state

Add a new state   Edit workflow

State history:

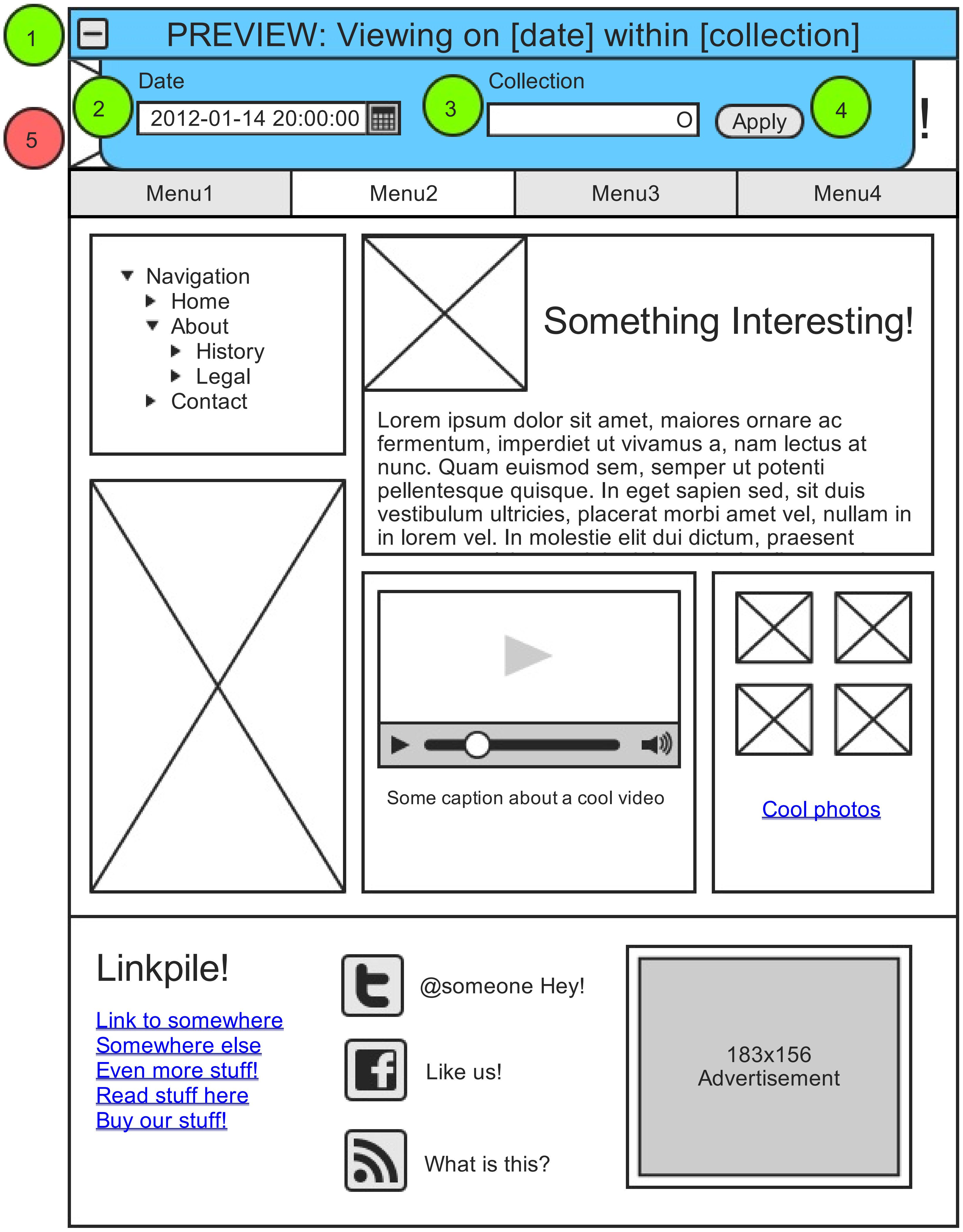
Type	State	From	To	Revision	User
W	Waiting for Game	2012-01-12 15:01:33	2012-01-16 16:44:11	13	webchick
C	Team A what-if	2012-01-12 15:01:33	2012-01-16 16:44:11	13	webchick
W	For Review	2012-01-09 08:11:51	2012-01-12 15:01:33	10	chrisstrahl
C	Team A Leading	2012-01-07 20:03:01	2012-01-10 21:16:55	None	chrisstrahl
W	First Draft	2012-01-06 11:31:11	2012-01-09 08:11:51	1	chrisstrahl

No notes for this page

About this section:

This section provides an example of how both collections and workflow states interact. It showcases an example from the South Carolina Republican Primary election. In this example, there are several pieces of content that are viewed at different dates and with different contexts.

Section Contents:


1
PREVIEW: Viewing on [date] within [collection]
5
2
Date
2012-01-14 20:00:00
3
Collection
Apply
4
!
Menu1
Menu2
Menu3
Menu4
Navigation
Home
About
History
Legal
Contact
Something Interesting!
Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit duis vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent
Some caption about a cool video
Cool photos
Linkpile!
Link to somewhere
Somewhere else
Even more stuff!
Read stuff here
Buy our stuff!
@someone Hey!
Like us!
What is this?
183x156
Advertisement

Notes:

*** Ignore page content - it's just a sample ***

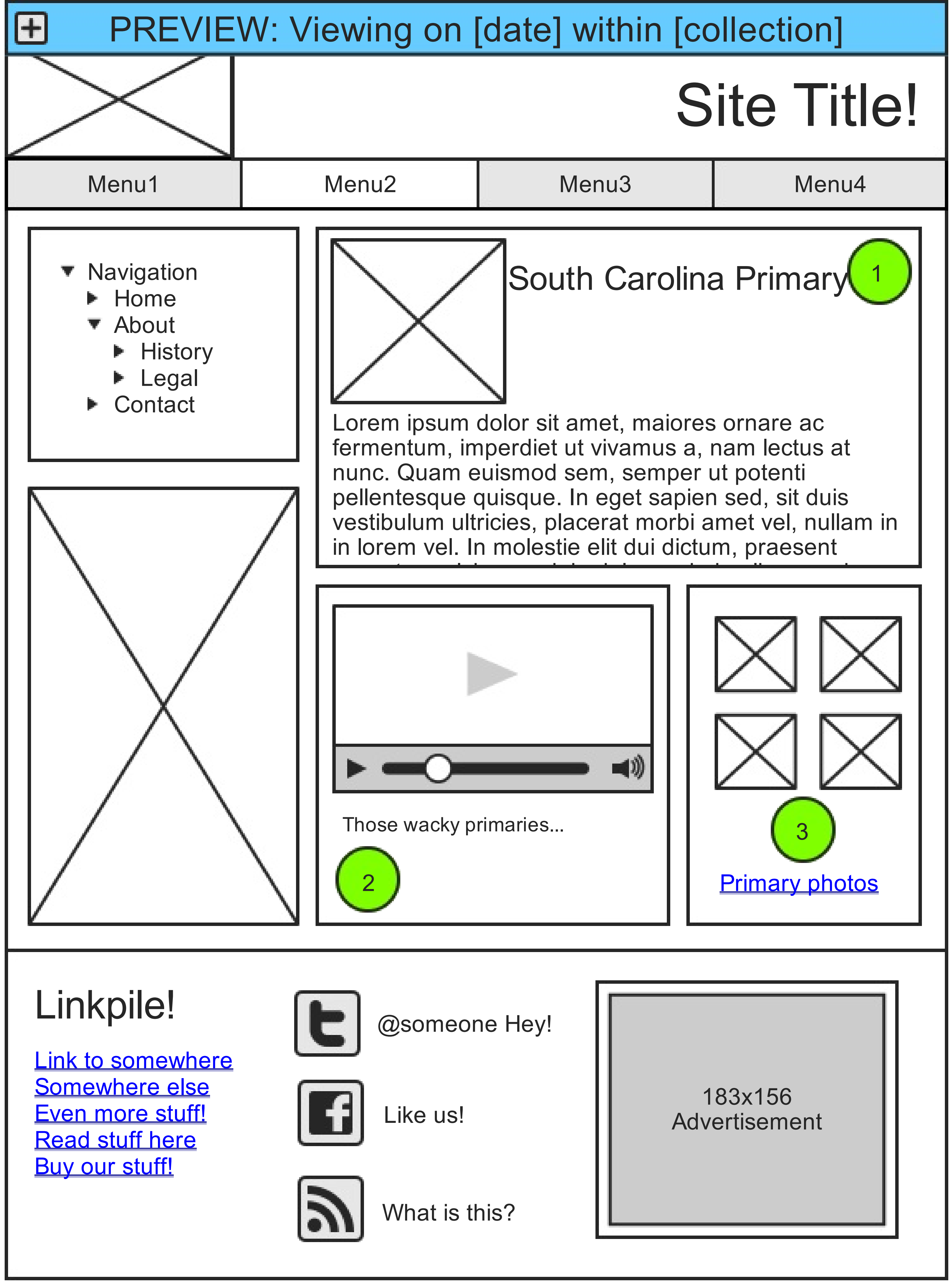
1. The time machine bar should appear at the top of the page and show the current date and state parameters being viewed. It should be able to expand to edit those parameters.

2. The date picker should allow a user to move the site to any point in time. Phase 1 will likely be future only, whereas future phases may allow you to move backward in time.

3. A user should be able to specify the context state to view the content. This is an autocomplete text field that selects from all of the available contexts.

4. The apply button applies the date and context parameters to the site changing the site to appear as it would on the appropriate date with the appropriate context.

5. This functionality could be extended to include things like language, location, user, or role. Having these options as a more flexible way of previewing content could expand the usefullness of the tool.



Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit duis vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent

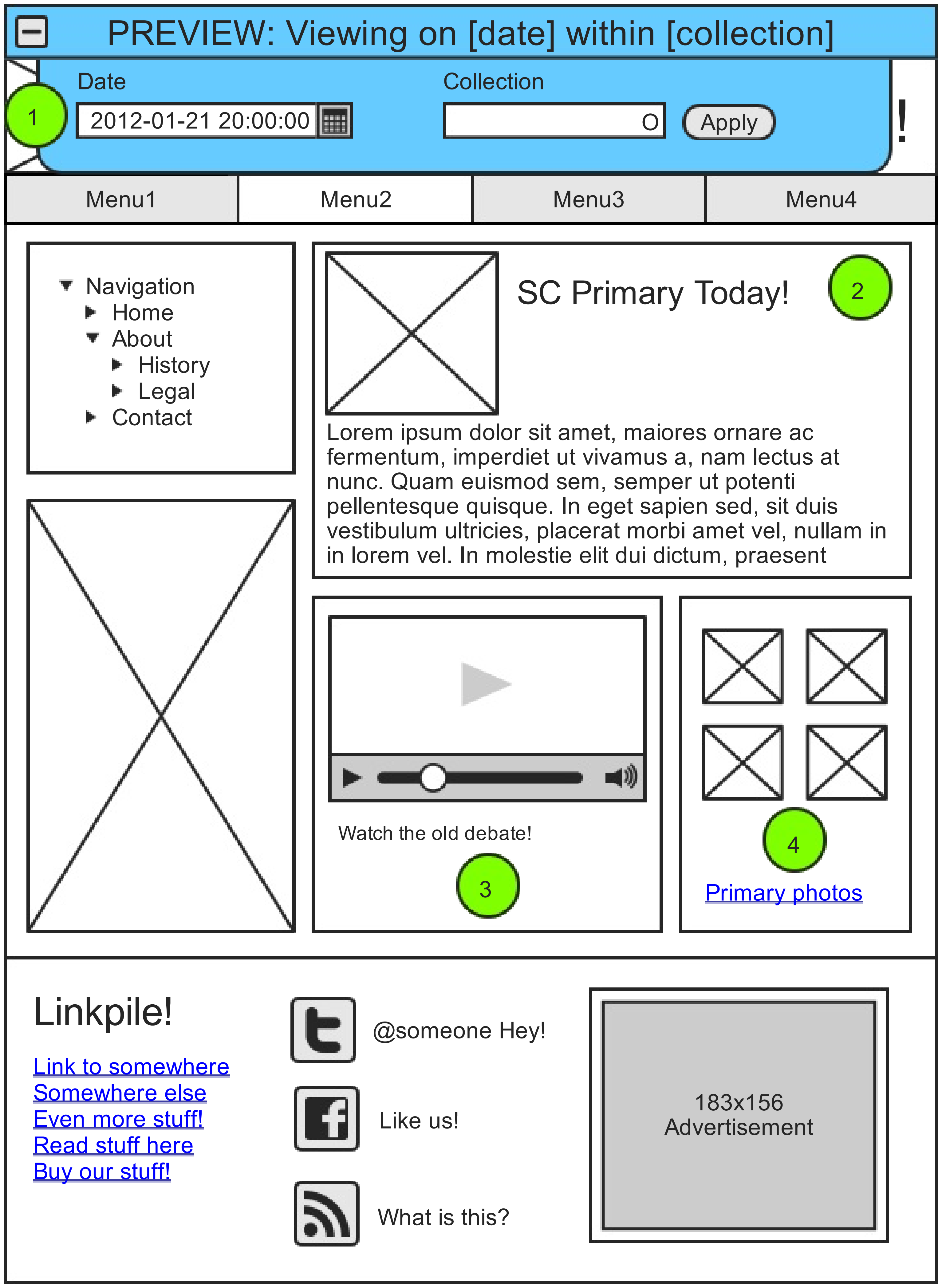
Notes:

*** This shows an example of what a current site looks like gearing up for the SC Republican Primary. The date parameter would be set for the current date / time, and the context parameter would be blank ***

1. This is a news article content that is currently published.

2. This is video content that is currently published.

3. This is a photo gallery that is currently published.



Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit duis vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent

Notes:

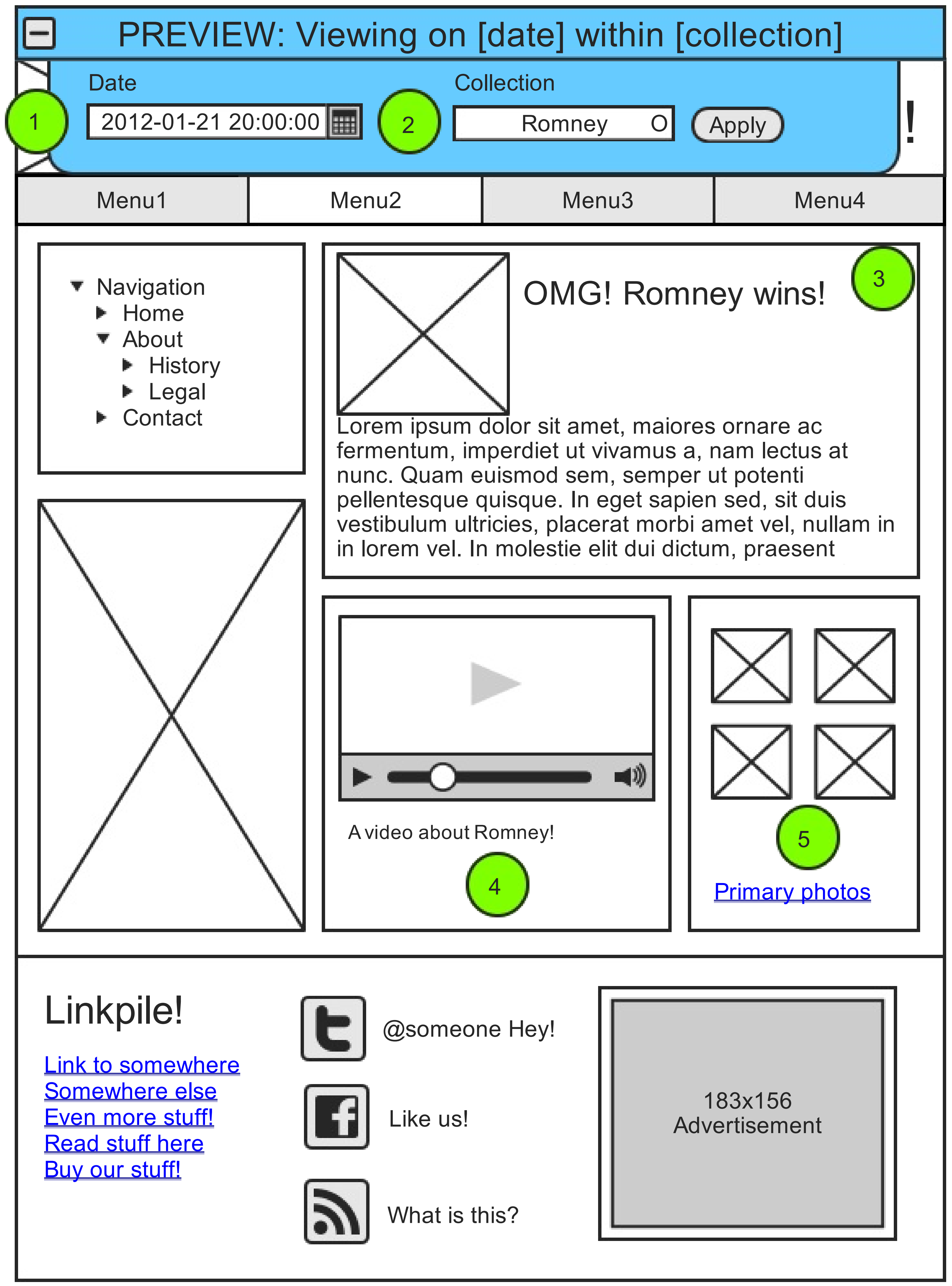
*** This demonstrates how content changes based on the date parameter being set ***

1. A date parameter for 8:00 PM on the day of the primary is selected. The system returns all of the content that is scheduled to be published at that particular time.

2. An article related to the SC Primary that is displayed that is scheduled to be published on 2012-01-21 00:00:00.

3. A video about an old debate is displayed that is scheduled to be published on 2012-01-20 08:00:00.

4. A gallery of primary photos that is currently published on the existing site is still published as it's not been scheduled to be removed from the page.


PREVIEW: Viewing on [date] within [collection]
Date
2012-01-21 20:00:00
Collection
Romney
Apply
!
1
2
Menu1
Menu2
Menu3
Menu4
Navigation
Home
About
History
Legal
Contact
OMG! Romney wins!
3
Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit duis vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent
A video about Romney!
4
5
Primary photos
Linkpile!
Link to somewhere
Somewhere else
Even more stuff!
Read stuff here
Buy our stuff!
@someone Hey!
Like us!
What is this?
183x156
Advertisement

Notes:

*** This demonstrates how content changes based on the date and collection parameter being set ***

1. A date parameter for 8:00 PM on the day of the primary is selected.

2. The collection parameter is set to "Romney" to simulate a Romney win. All content staged with the context of "Romney" is displayed on the page. If no tag is present, then the system defaults to whatever is "published" as of the selected date.

2. An article related Romney's win is published that has a start date of 2012-01-21 20:00:00 for the collection "Romney". This overrides the SC Primary content that was published there previously.

3. An video related Romney's win is published that has a start date of 2012-01-21 20:00:00 for the collection "Romney". This overrides the debate content that was there previously.

4. A gallery of primary photos that is currently published on the existing site is still published as it's not been scheduled to be removed from the page. No other content for this location has the context of "Romney".