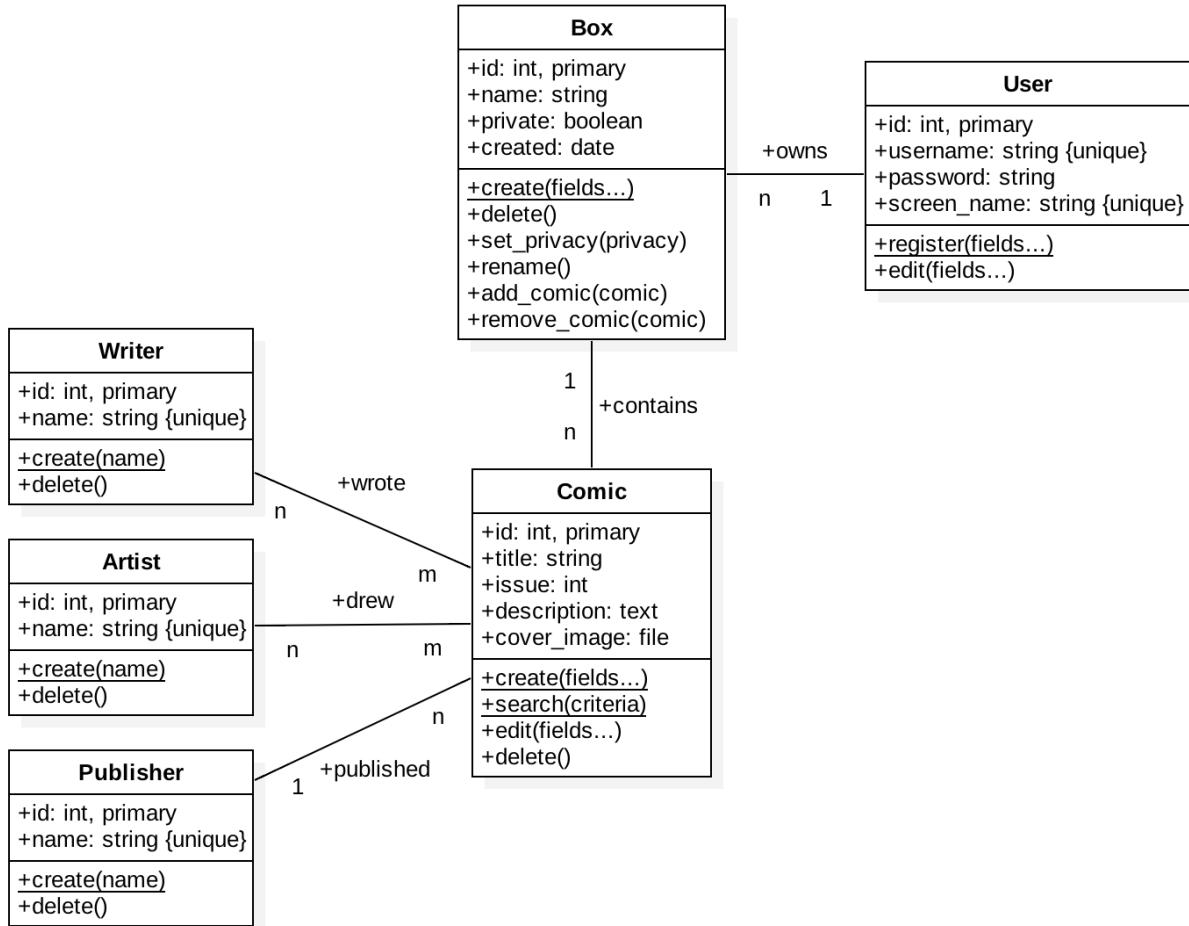


IAPT Longboxes Design Document

Yoo70944

27th November 2015

1 Data Model



Legend

- Primary keys are denoted by a primary tag after the attribute name.
- Underlined transactions denote that they are static, i.e., do not act upon a specific instance of the entity.

Assumptions

- Artist and Writer entities were separated from a single Person entity to increase extensibility and maintainability, as in the future fields specific to artists or writers may need to be added to the model.
- Although not noted in the diagram, Box names are case-insensitively unique to a User to prevent the potential confusion of two boxes owned by the same User with identical names.
- Integer primary keys are used in the model as they are best supported by *web2py*. Alternatively, unique fields are viable candidates for being the primary key.

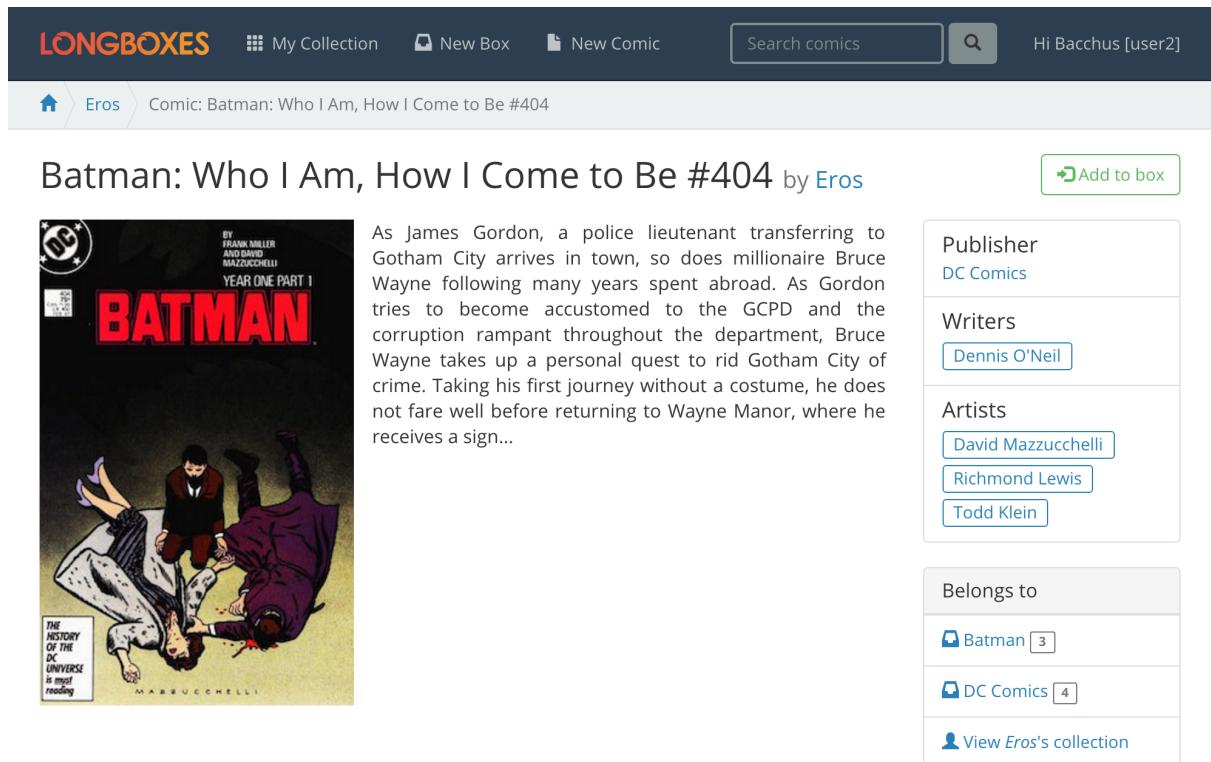
1.1 Transaction to controller function mapping

Model	Transaction	Controller function	Comments
User	register	default/user/register	
User	edit	default/user/profile	
Box	create	box/create	
Box	delete	box/delete	
Box	set_privacy	box/set_privacy	
Box	rename	box/view	[1]
Box	add_comic	box/add_comic	
Box	remove_comic	box/remove_comic	
Comic	create	comic/create	
Comic	search	comic/search	
Comic	edit	comic/edit	
Comic	delete	comic/delete	
Writer	create	comic/create	
Artist	delete	comic/edit	[2]
Publisher			

[1] This controller function handles box renames from a form in a modal dialog on the view page.

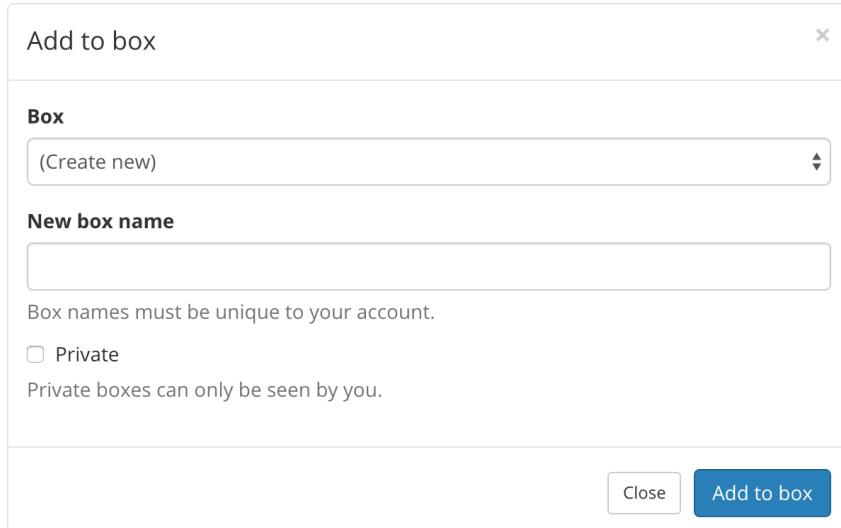
[2] Writer, Artist and Publisher entities are created on-demand as and when comics require them. When one of aforementioned entities is no longer referenced by any comic, it is deleted.

2 Design Rationale



The screenshot shows a web-based comic book viewer. At the top, there's a dark header bar with the "LONGBOXES" logo, navigation links for "My Collection", "New Box", "New Comic", a search bar, and a user profile for "Hi Bacchus [user2]". Below the header, a breadcrumb trail shows the user has navigated to "Eros / Comic: Batman: Who I Am, How I Come to Be #404". The main content area displays the comic cover for "Batman: Who I Am, How I Come to Be #404" by Eros. The cover art features a dramatic scene with two figures, one in a suit and another in a purple cape, set against a dark background. To the right of the cover is a detailed description of the comic: "As James Gordon, a police lieutenant transferring to Gotham City arrives in town, so does millionaire Bruce Wayne following many years spent abroad. As Gordon tries to become accustomed to the GCPD and the corruption rampant throughout the department, Bruce Wayne takes up a personal quest to rid Gotham City of crime. Taking his first journey without a costume, he does not fare well before returning to Wayne Manor, where he receives a sign...". To the right of the description are several boxes listing the comic's metadata: Publisher (DC Comics), Writers (Dennis O'Neil), Artists (David Mazzucchelli, Richmond Lewis, Todd Klein), Belongs to (Batman 3, DC Comics 4), and a link to "View Eros's collection". A green "Add to box" button is located at the top right of the comic's detail area.

Figure 2.1: Viewing a comic within another user's collection. (</comic/view/19>)



A modal dialog titled "Add to box" is shown. It has a "Box" section with a dropdown menu set to "(Create new)". Below it is a "New box name" input field. A note says "Box names must be unique to your account." There is a "Private" checkbox with the explanatory text "Private boxes can only be seen by you." At the bottom are "Close" and "Add to box" buttons.

Figure 2.2: The modal dialog that appears when selecting Add to box in Figure 2.1. Unfiled is selected in the drop-down by default, but here (Create new) has been selected to demonstrate the extra fields.

2.1 Supporting users through feedforward and feedback

To demonstrate how the interface supports users through good feedforward and feedback, I will explain the interface design rationale when a user wishes to add another user's comic to their collection.

Figure 2.1 contains several design decisions that support the user through this journey. All widgets that can be clicked (with the exception of text in the navigation bar) are coloured. For example, links in the 'breadcrumb' bar, the link in the main page header, and the links on the right-side of the page all have a bright colours. In addition, the widget to add the comic to their collection (the one the user is looking for to proceed on their journey), is coloured in a green hue to stand out and to indicate that this action performs a different type of action than the others. This distinguishable green hue, and the 'arrow into box' icon (used as a metaphor for placing an item into a box), is consistent with other 'add comic to box' operations across the application. Note that other controls on the page are grouped logically (such as the writers, artists and containing boxes), allowing the user to gloss over groups of widgets which are not relevant to their current intention.

Upon clicking the Add to box button, a small modal dialog appears over the current content, as shown in Figure 2.2. It contains a small form with a dropdown to select the box they wish to add the comic to. At this stage the user may realise that they want to put the comic in a new box that doesn't exist yet, and by selecting the (Create new) option, extra fields appear at the end of the form to create a new box. This enables the user to perform the task of putting a comic in a new box without having to context switch to the New Box journey. The fields themselves have instructions immediately below them to explain the constraints and ramifications of that input, allowing the user to make informed decisions and reduce erroneous inputs.

The user can clearly find the button to proceed by its relatively large size and bright, filled background colour, contrasting it from the rest of the form. Through the use of the required attribute on the applicable `<input>` elements, the web browser prevents the form from submitting without the required fields being populated, and prompts the user in the case of an error.

Upon submitting the form, the user is redirected to the newly created comic page, and is prompted by a green banner spanning the width of the grid explaining that the transaction was successful, as shown in Figure 2.3. Thus completing the user journey with clear signposting and feedforward, and clear feedback on their actions whether they be valid or erroneous.



Added comic to New Name Here.



Figure 2.3: Banner displayed at the top of the page after adding a comic to a new box.

2 Design Rationale

Figure 2.4: Search results for the `frank` query. Red lines superimposed to emphasise grid structure and the alignment of related control edges and text.
(/comic/search?search=frank)

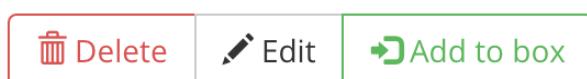


Figure 2.5: Command buttons on the comic detail page. Delete and Edit are only visible when the logged in user owns the comic under inspection.

2.2 Supporting users through visual layout

To demonstrate how the interface supports users through visual layout, I will explain the interface design rationale when a user wishes to delete a specific comic in their collection, but cannot remember its name. For the sake of example, in this scenario the user only knows an artist of the comic and not its title.

The most common tasks in the *Longboxes* application have been placed in the floating navigation bar to ensure they dominate the interface. The floating position of the navigation bar ensures high visibility by always being on screen regardless of scroll position, allowing for quick access to the most common tasks. Note that apart from the logo image, no links have a high saturation in the navigation bar. This is because all of these items can be clicked, and high contrast links in this bar could distract from the main interface. However, the links' background brightens when underneath the cursor to indicate that they are interactive. The links to the left-side and centre are accompanied by icons. These icons are used across the application to refer to collections, boxes and comics to allow for immediate recognition of what a control refers to before reading the text, thereby reducing cognitive load.

For the first step in the journey, the user notes the search box in the navigation bar by its contrasting edge, muted instruction text Search comics (as to not be distracting), and the magnifying glass icon, which is a ubiquitous metaphor across user interfaces to represent a search operation. The user enters the name of the artist of the comic, frank, and presses the Enter key to execute the search.

As shown in Figure 2.4, the comics are displayed in a grid formation below the page header. The spacing between the comic previews is consistent and spacious enough to demonstrate clear logical separation between them. The user notices that the query returns comics of which Frank Mastromauro is a writer, however they are only interested in a specific comic drawn by Frank Quitely. As is consistent across the application, the buttons for the most common actions for that page are grouped in the top-right corner, in line with the page header. The user clicks the Artist only button to exclude irrelevant results.

The user finds the comic they intend to delete. To view the comic's detail page, they click the More information link, which has a blue hue to be consistent with other links across the application. Figure 2.1 shows how this page looks when viewing a comic within another user's collection. In this case, as the user is viewing a comic they own, the command buttons appear as shown in Figure 2.5. The user double-checks that the comic they are about to delete is in fact the correct one, which is easy due to the prevalence of important information and grouping of the comic properties on the right-side, and the large type of the comic title and owner in the page header.

The red hue of the Delete button distinguishes it from the other widgets, and is consistent with other destructive operations across the application. This differentiation supports the user on this journey by making it easier to find the button. Clicking the button prompts the user to confirm that they intend to irreversably delete the comic, ensuring the click was purposeful. Upon confirmation, the comic is deleted and the user is redirected to their collection with feedback notifying them that the operation completely successfully.