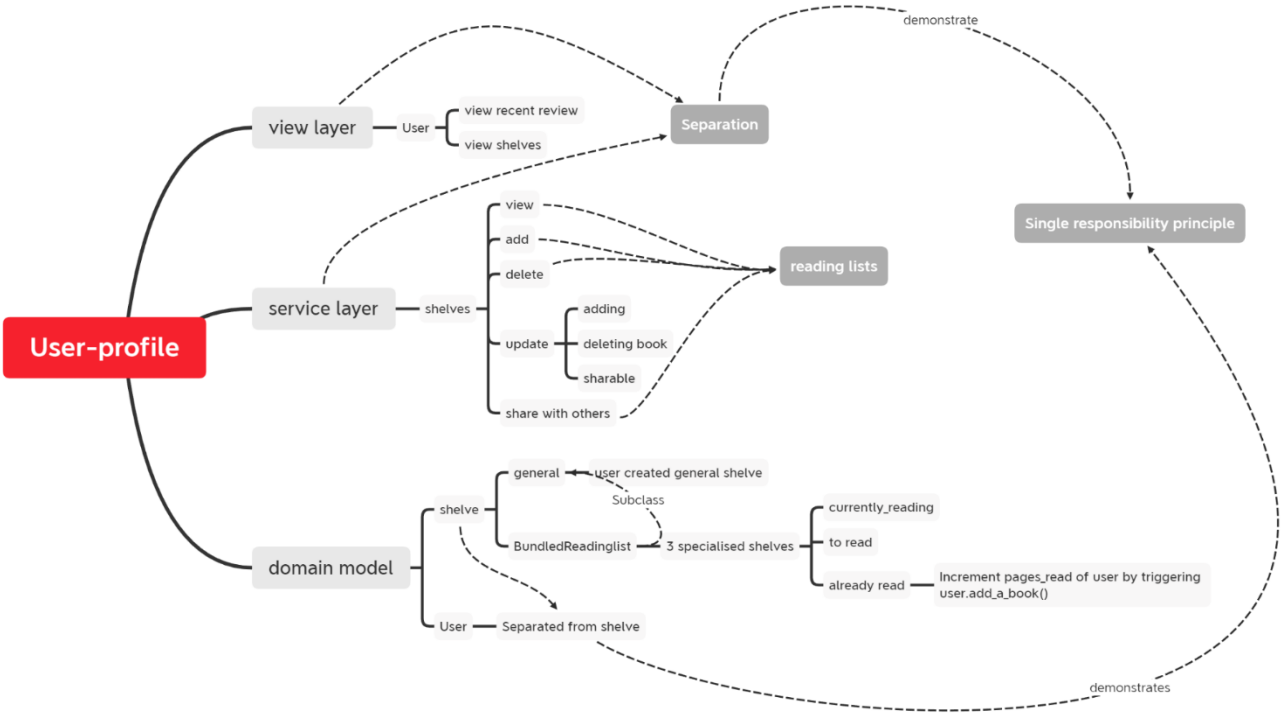
Assignment 2: User Profile Feature Report

Members: Kevin Hu and Darren Joe

Introduction: For our website Book zone, our cool new feature introduces a user profile section. Which gives users the ability of viewing their recent reviews made on any book and giving the user a way to create new shelves which can record certain books into their appropriate reading list such as to read, reading, and read.

New features:

Besides basic functionality of create, read, update, and delete (CRUD), We implemented a new feature which allows users to share their existing shelves with other users on the websit. When a user sets the public attribute to true, other users, including guest users without logging in are given access to view the appropriate reading list provided by the user, an example being “/reading_list?id=3”



Form usage:

We designed 3 forms using WTForms to allow the user to maintain the data inside the user profile shelves which include, add_to_list, delete_list, and delete_book. Each form utilizes the single responsibility principle in altering the information available in the user profile. Add_to_list allows for the user to add new books to their corresponding list and have it updated onto their user profile page. Delete_list allows

for the user to remove from the already existing lists and delete_book which allows for the removal of books from the existing lists.

Blueprints:

Throughout the implementation of the user profile page, we extended the functionality of various blueprints to implement the user profile such as book service blueprint which was implemented to be able to supply review data made by the current user. We also made use of the authentication blueprint which manages the authentication of new and existing users on the website, and load user which was implemented to help us obtain cookies of the user and use that information to display the appropriate information corresponding to the user logged in.

Layers:

The user profile in terms of displaying the content utilizes the single responsibility principle, utilizing various html files within the user and book files in templates. These html files were all designed to display the correct information and not rely on other files to show the information required in user profiles. This made it easier for us to maintain and decide on key design features that would benefit the user experience, we utilized a grid layout to keep information compacted but not overly cluttered on each front-end page of the website keeping it consistent. Overall making it more manageable in implementing the design aspect of the user profile.

When the user intends to add a new book a function call is made to the service layer which binds it to a shelf and adds it to the memory repository; this separates the view layer from the backend logic and binds the new reading list to existing resources.

Domain model:

Each user can have a variable number of reading lists which is aggregated to a shelve object rather than being directly attached to a user. This demonstrates the single responsibility principle to do more complicated handling in the shelf class than relying on other classes. For our model design we created a subclass called bundled reading list, what is included into these reading lists is excluded from each other. The to read, currently reading and have read reading lists are all instances of BundledReadingList, that is when a book is added to a reading list it is removed from the to read list and later added into the read list upon completion of reading the book. We have defined the ‘already read’ reading list to be of class ReadReadingList which is subclass of BundledReadingList, because it inherits all the behaviour of a BundledReadingList and will increase the number of pages read by a user