

Book Club

	1. Below Standard	2. Approaching Standard	3. Meets Standard	4. Above Standard
Feature Completeness	- Fewer than 12 user stories are complete.	- At least 12 user stories are complete but not all stories are 100% finished.	- All user stories are 100% complete.	- All user stories are 100% complete and additional extension work was attempted.
MVC Responsibilities	- Views, Controllers, and Models do not have a clear sense of responsibilities.	- Views contain logic that make significant additional calls to the database including Model class methods. - Controllers do a significant amount of processing of data that should exist at a model level. - Models process data in a way that is inefficient, or contains work that determines presentation logic (how data might get viewed).	- Views make some calls to the database for .each calls and branch logic. - Controllers push almost all work to the Models for reading/writing. - Models sufficiently process and retrieve data.	- Views only display data and do not make additional calls to the database; they may include branch logic. - Controllers request all data from Models except #all or #find calls and only do #create and #delete methods for writing data. - Models read/write all other data using branch logic or smaller methods.
ActiveRecord	- No ActiveRecord methods are used to read/write data; Ruby code was written to process data.	- Significant number of places where Ruby is used to process data that could use ActiveRecord instead.	- ActiveRecord is used in a clear and effective way to read/write data using no Ruby to process data.	- Highly effective and efficient use of ActiveRecord beyond what we've taught in class or by this point in the Obstacle Course. Even calling #each methods will not cause additional database lookups from the database.
Testing	- Testing is below 75% or not done.	- Testing is above 75% and below 98% - Tests are poorly written/unclear objective of the tests. - Hard coding values rather than using variable.	- 98% or better test coverage for features. - 100% model test coverage. - Using variable as opposed to hard coding values. - Meaningful written tests	- Very clear test-driven development. - Test files are extremely well organized and nested. - Utilize before(:each) blocks for sharing test setups. - 100% test coverage for features, and 100% of all model methods is tested.
UX & Styling/ Layout	- Little or no styling or layout. - User workflow is significantly problematic(unclear what routes are available or how to use the application).	- Styling is poor/incomplete. - Layout does not utilize good HTML patterns - Incomplete navigation for routes. (Must manually type URIs)	- Purposeful styling pattern and HTML layout using the application.html.erb and application.css - Links/Buttons to reach all areas of the site	- Extremely well styled and purposeful layout. - Excellent color scheme and font usage. - Easy to use and follow the application workflow. (User does not have to search for links/buttons for navigation) - Utilizes additional tooling, such as SCSS