Daniel Hanna

Testing is an important aspect of application development. Below is an explanation of the approaches that will be used for the various testing types.

1. **Unit Testing:** Unit testing will be performed by testing each method to ensure that it can properly perform its desired function. More specifically, local data structures will be tested to ensure that local data is stored properly, and boundary conditions will be tested to ensure that the program does not fail in atypical situations. Error handling will also be tested (essentially by “purposefully breaking” the flow of typical operations to create an error situation).

2. **Integration Testing:** Integration testing would be performed in a bottom-up style, in which modules at lower levels would be tested with “higher up” modules. For example, testing to ensure that the link embedded in the site logo integrates appropriately with the correct route for the home page will be done.

3. **System Testing:** After successful unit and integration testing, system testing will be performed most effectively by presenting the finished product to testers, who would use the finished system and provide feedback on and determined issues. This would most likely be done near the end of the prototype stages, before deployment/hand off.

Testing Tools to be Used

There are several effective tools that are available to aid in the use of testing and debugging. One major consideration which must be made how to *manage* testing; this will be done via tracking issues and milestones through GitHub. This GitHub feature is an excellent way to organize tasks which must be completed and issues that are made apparent throughout the project’s development. Debugging using Git commands may also be used, particularly in the advent of a non bug-free commit. Debugging any Ruby code with tools such as Pry may or may not be used, as dependant on the severity of the bugs/errors and the extent to which they can be discovered during unit and integration testing. However, there is an increased probability of such a tool being used during the system testing phase, particularly because it may be difficult to locate such issues near the project’s delivery time. Of course, any used tools or changes will be appropriately documented during the growth of the application.

Test Cases

A table for test cases can be seen below.

Functionality Tested	**Inputs**	**Expected Output**	**Actual Output**
View table of articles	User request (HTML link selection)	Display table of articles	
No Error/crash for	Incorrectly formatted	Return homepage	

search	submission for search		
Add article to website	Title, text	Return submitted article with title and text	
Display Error for new article submission	Incorrectly formatted title/text	Return new article form and display error that stopped form from being submitted	
Delete article from website	Request in form of HTML link selection	Display confirmation confirmation box asking “are you sure?” with “yes” and “no” as options	
Confirm article deletion	User input (confirmation box) saying “yes”	Return the updated article list	
Confirm article deletion	User input (confirmation box) saying “no”	Return the article list with no items deleted	
Confirm article editing	User Request (HTML link selection)	Return the filled out article form that is editable	