Visual Development Platform for Ruby on Rails

Anmol Desai*, Nicholas Molloy*, Jing Sun† and Gillian Dobbie†

*Department of Electrical and Computer Engineering
†Department of Computer Science, The University of Auckland, New Zealand
Emails: *{ades597, nmol886}@aucklanduni.ac.nz, †{jing, gill}@cs.auckland.ac.nz

***Abstract*—This paper briefly reviews the construction of Ruby on Rails applications, identifies the pitfalls of existing tools, and proposes the design and development of a lean cross-platform desktop application, along with an evaluation of the prototype.**

***Index Terms*—Ruby on Rails, Visual tool, Web Applications**

I. Introduction

Ruby on Rails is a framework that runs on the Ruby language, and provides developers the tools necessary to build modern web services [1]. While rails simplifies the creation of web-services, it can be found to be difficult for a beginner to not only install the framework, but to also learn and use it. This in part is due to the fact that the developer interacts with rails predominantly through the command line. This paper presents the initial research on existing tools, the design decisions, the development and implementation of a visual development platform for Ruby on Rails, namely the *Rails Editor*.

The tool provides the developers with many features that aim at enhancing the productivity of a web application in Ruby on Rails [2]. The developers are able to execute many Ruby on Rails commands visually, clone from a git repository, create model/controller/routes diagram, visualise database schemas, chat and use a native inbuilt terminal. The collection of all these features in the editor makes the development of Ruby on Rails projects much more convenient as well as efficient.

II. Design and Implementation

The *Rails Editor* was developed as a cross-platform desktop application, which utilises and extends the Electron framework, Chromium, Node.js, HTML, CSS, etc. Figure 1 below shows the overall architecture of our solution.

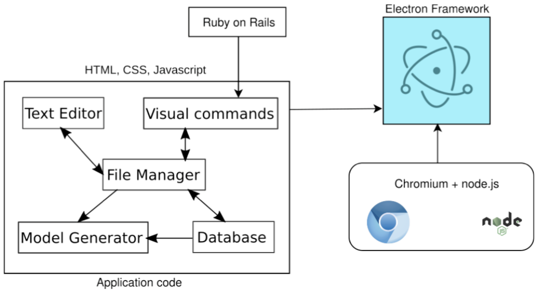

Fig. 1. Overall architecture of the tool

DOI reference number: 10.18293/SEKE2017-047

In order to aid in developer's understanding of Rails projects that they are working on, several diagrams, e.g., the model, controller and route diagrams, can be generated and viewed in the tool. These diagrams illustrate the important aspects of a Rails project in a visual manner. For example, Figure 2 shows an example of the generated route diagram.

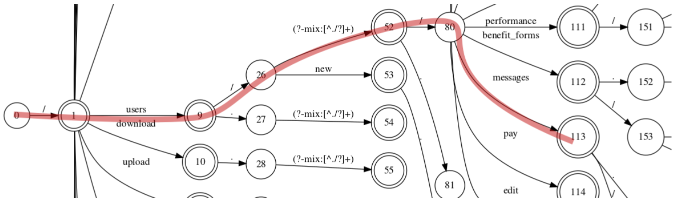

Fig. 2. An example of a route diagram generated by the tool

III. Evaluation and Conclusion

Three types of evaluations were conducted. A user survey showed that all of the features implemented were received positively by the participants. A comparison with existing tools showed that the tool developed implements many features that are not existent in currently available tools. A small performance test showed that the tool is able to perform within reasonable limits. Finally, it is worth mentioning that the *Rails Editor* project won the Final Year Research Project Prize in the Software Tools category by a panel of judges from ICT industries in 2016.

In the future, we plan to make the following improvements. The chat feature could be extended to support a group of developers. Currently chat only allows two users to communicate at a time. It could be extended to allow team programming features, in a similar way to products such as Skype or Google Hangouts. Since the tool is implemented using web technologies, a large portion of it can be run within a standard web browser. The application could be modified and extended in such a way that it runs on a "cloud" infrastructure. Features such as running rails commands and the terminal would be executed remotely, allowing a user to access their development environment from any where using a web browser.

References

[1] B. Tate and C. Hibbs, *Ruby on Rails: Up and Running, 2nd Edition*. O'Reilly Media, October 2008.

[2] C. Jones, *Applied Software Measurement (2Nd Ed.): Assuring Productivity and Quality*. Hightstown, NJ, USA: McGraw-Hill, Inc., 1997.