FINITE STATE MACHINE WEB APPLICAITON

Software Requirements Specification

4/27/2014

Author: Dustin Hardin

Lawrence Technological University

MCS5013 Web Server Programming

Table of Contents

[Version History 2](#_Toc386373701)

[Introduction 2](#_Toc386373702)

[Description 2](#_Toc386373703)

[Specific Requirements 2](#_Toc386373704)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 0.1 | 4/26/2014 | Dustin Hardin | Initial outline of requirements. |
| 0.1.2 | 4/26/2014 | Dustin Hardin | Review and edit. |
| 1.0 | 4/27/2014 | Dustin Hardin | Revision. |
| 1.1 | 5/20/2014 | Dustin Hardin | Modified SRS priority and layout. Added additional requirements. |

# Introduction

Software developers have difficulties designing finite state machines on the fly due to not only funding constraints to purchase the required software, but having the software actually implement the design for them accurately. The provided program will be able to not only make designing finite state machines for convenient for developers, but the tool will also provide the following benefits:

* Runs in all major browsers (i.e., Chrome, Firefox, Internet Explorer v9+)
* Quick and intuitive graph creation
* Save and load graphs from a server
* Special character code support via string commands

Developers are attracted to tools which increase their productivity and do not have a dollar amount attached. The problem with the tools available today is that they require expensive licensing and often times, do not provide quick and intuitive results.

# Description

The FSM Online web application will allow users to quickly create finite state machines within a web browser. The web application will employ various third-party plugins to assist and expedite the development process. Not all functionalities will require a plugin and will be implemented using proprietary methods within the applications namespace.

# Specific Requirements

1. Draw [Priority = High]
   1. States
      1. The System shall represent States as circles.
      2. When the User double clicks on the graph, the System shall create a State at the location of the click event.
      3. When the User presses and holds the Mouse 1 button over a State, the System shall allow the User to move the State on the graph to the location of the Mouse 1 release.
   2. Transitions
      1. The System shall represent Transitions as directed arrows from the current State to the destination State.
      2. When the User holds the Shift key and presses the Mouse 1 button, the System shall draw a directed Transition to the destination State that Mouse 1 button was release on.
      3. The System shall draw Transitions in an arched directed arrow to prevent Transition overlap.
   3. Labeling
      1. When the User presses any printable characters on the keyboard, the System shall insert the corresponding text into the selected object.
      2. When the User presses the Backspace key on the keyboard, the System shall delete the last character of the selected object’s text.
      3. The System shall center all labeling in the middle of objects.
      4. The System shall replace all string combinations of /<Greek letter spelled out> with its corresponding Greek letter (e.g., /lambda shall be replaced with λ).
   4. Selecting
      1. When the User presses the Mouse 1 button when the cursor is over a State, the System shall select that State.
      2. When the User presses the Delete key, the System shall delete the selected State.
      3. When the User presses the Mouse 1 button when the cursor is over a Transition, the System shall select that Transition.
      4. When the User presses the Delete key, the System shall delete the selected Transition.
      5. The System shall only select one object at a time.
      6. The System shall outline selected States green.
      7. The System shall outline selected Transitions blue.
   5. Pan & Zoom
      1. The System shall allow the user to *pan* *& zoom* over the graph.
      2. The System shall represent the *pan & zoom* with a mini-map of the graph.
   6. Resizing
      1. The System shall dynamically resize the graph to accommodate the size of the finite state machine graph.
   7. Collision Detection
      1. The System shall not allow the User to move States on top of other States.
2. Login [Priority = High]
   1. Validation
      1. The System shall allow the User to login with their Facebook or Google account.
3. Logout [Priority = High]
   1. The System shall allow the logged in User to logout.
4. Save [Priority = High]
   1. JSON
      1. The System shall allow the User to save the graph to the server.
5. Load [Priority = High]
   1. JSON
      1. The System shall allow the User to load any graphs saved under their username into the application.
6. Export [Priority = Medium]
   1. PDF
      1. The System shall allow the User to export their graph to a PDF file.
   2. PNG
      1. The System shall allow the User to export their graph to a PNG file.
   3. JSON
      1. The System shall allow the User to export their graph to a JSON file.
7. Import [Priority = Medium]
   * 1. The System shall allow the User to import a JSON file.
     2. The System shall draw the graph based on the imported JSON file.
8. Undo/Redo [Priority = Low]
   * 1. The System shall allow the User to Undo and Redo changes made to the graph in the current session.