|  |
| --- |
| Lawrence Technological University |
| Finite State Machine Online |
| Manual & Documentation |

|  |
| --- |
| Author: Dustin Hardin  Course: MCS5013-01 Web Server Programming  Date: 4/28/2014 |

Table of Contents

[Version History 2](#_Toc386839637)

[Instructions 3](#_Toc386839638)

[Application Overview 5](#_Toc386839639)

[User Interface 5](#_Toc386839640)

[Graphing 5](#_Toc386839641)

[Account Management 5](#_Toc386839642)

[Session Management 5](#_Toc386839643)

[Database 5](#_Toc386839644)

[Cross Browser Compatibility 5](#_Toc386839645)

[JavaScript 5](#_Toc386839646)

[CSS 5](#_Toc386839647)

[Technologies 5](#_Toc386839648)

[Front-End Application: HTML, JavaScript, CSS 5](#_Toc386839649)

[Additional CSS & Components: Twitter Bootstrap 5](#_Toc386839650)

[Server & Server Application: Node.js, JavaScript 6](#_Toc386839651)

[Additional Server Modules: Express and Mongoose 6](#_Toc386839652)

[Database: mongoDB 6](#_Toc386839653)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 4/26/2014 | Dustin Hardin | Initial outline of requirements. |
| 1.1 | 5/1/2014 | Dustin Hardin | Added sections on cross-browser compatibility and technologies implemented. |
| 1.2 | 5/3/2014 | Dustin Hardin | Added “Application Overview” section. |

# Instructions

1. Installation
   1. Node.js
      1. Download and Install Node.js.

Link: <http://nodejs.org/download/>

* 1. mongoDB
     1. Download and Install mongoDB.

Link: <https://www.mongodb.org/downloads>

* 1. Finite State Machine
     1. Download project files for Finite State Machine.
     2. Extract Finite State Machine to your local machine.
     3. File directory setup
        1. After extracting, your file director should be as displayed in Table 1 below.

|  |
| --- |
| Finite State Machine  +-lib  | +--crud.js  | +--routes.js  | `--user.js  +--app.js  `--package.json  +--public  | `--fsm.html  +--css  | +--fsm.css  | +--fsm.graph.css  | `--fsm.shell.css  +--bootstrap  | +--bootstrap.css  | +--bootstrap.css.map  | +--bootstrap.min.css  | +--bootstrap-theme.css  | +--bootstrap-theme.css.map  | `--bootstrap-theme.min.css  +--fonts  | +--glyphicons-halflings-regular.eot  | +--glyphicons-halflings-regular.svg  | +--glyphicons-halflings-regular.ttf  | `--glyphicons-halflings-regular.woff  +--js  | +--fsm.data.js  | +--fsm.fake.js  | +--fsm.graph.js  | +--fsm.js  | +--fsm.menu.js  | +--fsm.model.js  | `--fsm.shell.js  +--jq  | +--bootstrap.js  | +--bootstrap.min.js  | +--jquery.event.gevent.js  | +--jquery.uriAnchor.js  | `--jquery-1.10.2.js |

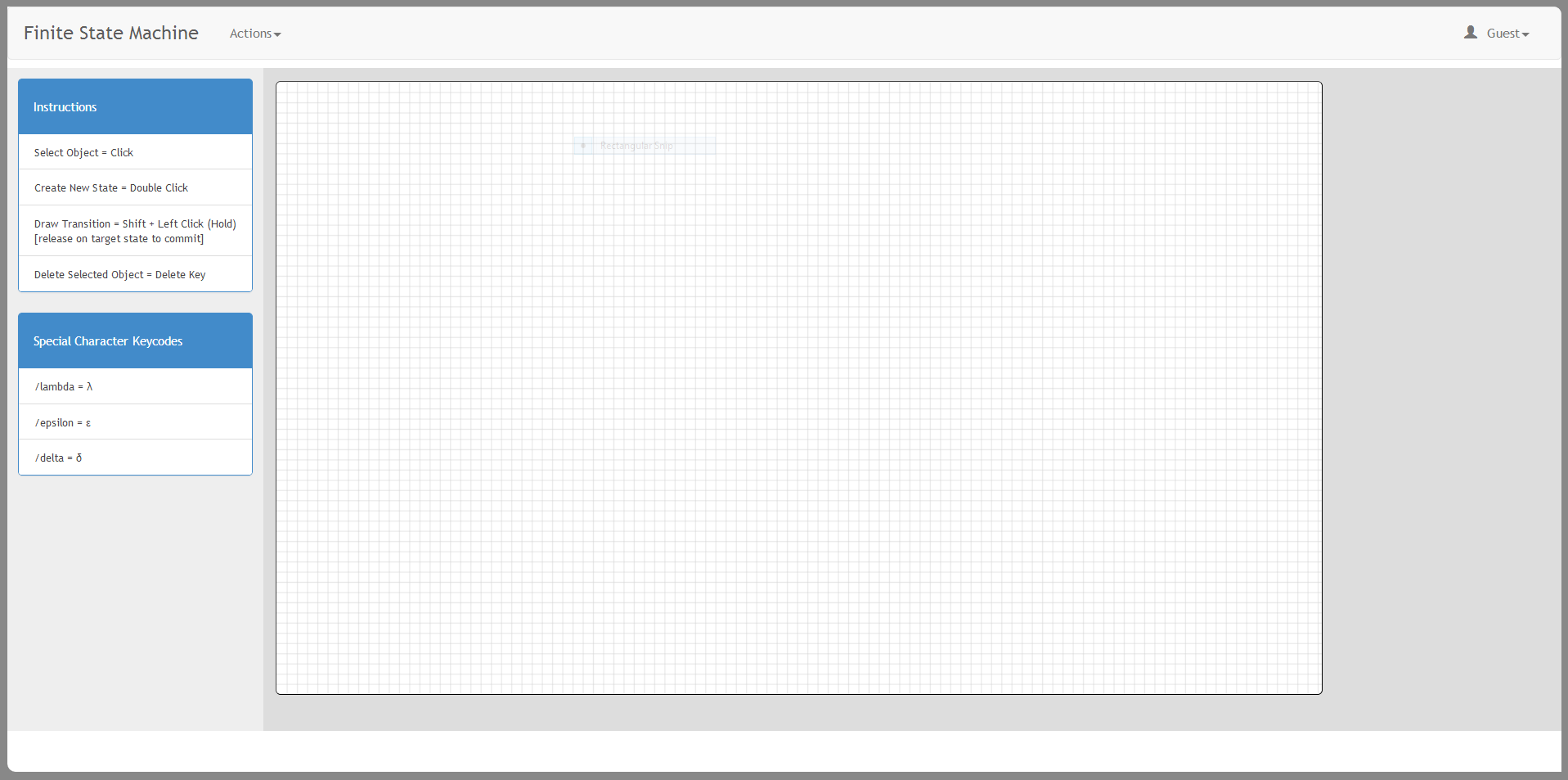
Table - Finite State Machine Online File Directory Setup

1. APM package install
   1. Open Command Prompt.
   2. Navigate to Finite State Machine top level directory.
   3. Type npm install and press Enter.
      1. This will install all of the server dependencies listed in the package.json file which resides in the top level directory listed in Table 1 above.
2. mongoDB Database Setup
   1. Create a directory on the server or local machine to store the database files for the users and graphs.
   2. Open Command Prompt.
      1. Type mongod --dbpath <directory created in previous step> and press Enter.
      2. You may now close the Command Prompt for mongod
   3. Open another Command Prompt.
      1. Type mongo and press Enter.
      2. Type use fsm and press Enter.
      3. Type db.user.insert({“name”:”<some name>”, “password”:”<some password>”}); and press Enter.
         1. Repeat this step for all of the various users you wish to have access to the system when using the Finite State Machine application. This is a workaround since user registration has not yet been implemented.
      4. You may now close the Command Prompt for mongo.
3. Application Setup
   1. Database
      1. Open Command Prompt (on Server).
      2. Type mongod --dbpath <directory created in previous step> and press Enter.
   2. Server
      1. Open Another Command Prompt (on Server).
      2. Navigate to Finite State Machine top level directory.
      3. Type node app.js and press Enter.
   3. Web Application
      1. Open a web browser.
      2. If web browser opened on same machine as server:
         1. Type <http://localhost:3000> and press Enter.
      3. If web browser opened on a different machine then server:
         1. Type IP address of server and press Enter (e.g., 192.168.1.59 if both the server and client machine are running in the same network).

# Application Overview

## User Interface

*User Login*



*User Instructions*

*Action Menu: New, Save, Load, Print functions*

*Editable Graph*

Figure - Finite State Machine User Interface

## Graphing

Users may interact with the graph using their mouse and keyboard. User instructions on the various commands are laid out in the *User Instructions* portion of the user interface.

## Account Management

Users are able to Login and Logout of the Finite State Machine application using the login at the top right hand corner of the user interface.

## Session Management

Session management has not yet been implemented in this version of the Finite State Machine application.

## Database

User interactions such as Login, Save, and Load emit events to the server, sending information regarding the current user and the current state of the graph.

{

\_id: <ObjectId2>,

user\_id: <ObjectId1>,

name: *String*,

states: [{

x: Number,

y: Number,

text: String,

startState: Boolean,

endState: Boolean,

transitions: [{

startState: { x: Number, y: Number },

endState: { x: Number, y: Number },

text: String

}]

}]}

Document 2 - Graph

Document 1 - User

{  
 \_id: <ObjectId1>,

username: *String*,

password: *String*

}

Items in the mongoDB database are referred to as *Documents* and are stored in JavaScript Object Notation (JSON) format. Both users and graphs have a unique ID and are paired by storing the user ID of the user who saved the graph in the graph document entry. When graphs are loaded, the user ID of the current user is used to query the graph document collection.

# Cross Browser Compatibility

## JavaScript

When designing a web application, developers have to decide on which browsers to support due to time constraints, as well as project funding. An approach to designing web applications is to make it usable to the three major browser vendor: Google Chrome, Internet Explorer, and Mozilla Firefox.

Backwards compatibility testing was done only on the latest iterations of each of these browsers due to time constraints and has been verified to work in each environment.

## CSS

The Finite State Machine web application was created using not only custom CSS for layout, but also with CSS styling scripts from the third-party vendor Twitter: Bootstrap.

The responsive design allows for cross-browser styling that remains consistent. While there are CSS styling *hacks* that exist for the major browsers, the application styling remains seamless between each one.

# Technologies

## Front-End Application: HTML, JavaScript, CSS

### Additional CSS & Components: Twitter Bootstrap

Links:

[Bootstrap](http://getbootstrap.com/getting-started/)

## Server & Server Application: Node.js, JavaScript

### Additional Server Modules: Express and Mongoose

Links:

[Node.js](http://nodejs.org/)

[Express](http://expressjs.com/)

[Mongoose](http://mongoosejs.com/)

## Database: mongoDB

Links:

[mongoDB](https://www.mongodb.org/)





