SE 3XA3: Software Requirements Specification
Finite State Machine Simulator

Team #16, NonDeterministic Key
Anhao Jiao (jiaoa3)
Kehao Huang (huangk53)
Xunzhou Ye (yex33)

8 February 2022

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
8 February 2022	1.0	First version
21 March 2022	2.0	Revision 0
12 April 2022	3.0	Revision 1

This document describes the requirements for Finite State Machine Simulator. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (**?**). If you make further modifications to the template, you should explicity state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

The purpose of our project is to build a software system called Finite State Machine Simulator that provides a convenient way to model finite state machines and simulate transitions between states. The system is also able to visualize finite state machines by generating their corresponding LaTeX snippets. The system is intended to be used both as a Python library and as standalone application running in the Python interactive shell.

1.2 The Stakeholders

1.2.1 The Clients

The clients for this project are Professor Asghar Bokhari of McMaster University and TAs of the course SFWRENG 3XA3.

1.2.2 The Customers

The customers for this projects are students, educators, and people who work with finite state machines in an academic setting.

1.2.3 Other Stakeholders

Not applicable. The software system is intended to provide academic support and requires the users to posses basic knowledge of finite state machines and Python programming language. Therefore, members of the public is not included in our consumer base.

1.3 Mandated Constraints

- The product must be completed by April 12, 2022.

1.4 Naming Conventions and Terminology

- **FSM**: Short form of finite state machine.
- **FSMS**: Short form of Finite State Machine Simulator, which refers to this software system.
- **SRS**: Short form of Software Requirement Specification, which refers to this document.
- **CLI**: Short form of Command-Line Interface.

1.5 Relevant Facts and Assumptions

1.5.1 Relevant Facts

This project is a reimplementation of an open source project, pytransitions/transitions. The original code base is under MIT open source license and has around 3000 lines of codes.

1.5.2 Assumptions

- Users are familiar with Python programming language.
- Users have basic knowledge of FSM.
- Users have experience with CLI.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

Work is assigned and organized into deliverables listed in Table 2.

Table 2: **Deliverables**

Item	Due Date
Requirements Document (Revision 0)	February 11
Proof of Concept Demonstration	Week of February 28
Test Plan (Revision 0)	March 11
Design and Document (Revision 0)	March 18
Revision 0 Demonstration	Week of March 21
Final Demonstration (Revision 1)	Week of April 4
Peer Evaluation	Week of April 4
Final Documentation (Revision 1)	April 12

2.1.2 Work Partitioning

Partition 1: Core function of FSMS, including FSM creation, modification, trigger declaration

Partition 2: LaTeX snippet exportation

2.1.3 Individual Product Use Cases

UC1. To define FSMs

UC2. To modify existing FSMs

UC3. To fire triggers (transitions) and inspect the working state

UC4. To export a defined FSM to external sources

2.2 Functional Requirements

FR1. The user must be able to define a FSM by specifying relevant states and an initial state.

Rationale A minimum well-defined FSM is consists of a set of states and a initial state as the entry point. This is the most basic requirement for any FSM simulating system.

Fit Criterion The user is able to instantiate a FSM object with all specified states.

Priority High

FR2. The user must be able to add, remove, modify transitions between states (triggers).

Rationale Having triggers makes a FSM non-trivial, and allows simulations to be performed.

Fit Criterion The user is able to update triggers in a FSM as they wish. The effect of changes to triggers takes place as soon as the user completes the update transaction.

Priority High

FR3. The user should be able to add, remove, modify states in a FSM.

Rationale It is always welcome to provide additional flexibility of modifying FSMs on-the-fly to the user. There is no need to make states constant for any FSM.

Fit Criterion The user is able to modify state as they wish.

Priority Medium

FR4. The user must be able to fire triggers they specified in prior

Rationale If the user is not able to invoke transitions, it defeats the purpose of specifying transitions for a FSM.

Fit Criterion The system record and processes the user's transaction and transition the state of the FSM accordingly.

Priority High

FR5. The user must be provided a mean to inspect the state of a FSM.

Rationale If the user is not able to inspect the active state of a FSM, the user has no way of know the result of any performed transition. Simulation on FSMs is all about observing state changes as the result of triggers firing.

Fit Criterion The user can access the active state of a FSM at any time.

Priority High

FR6. The user must be able to export an existing FSM in the form of LaTeX snippets.

Rationale The user can use the exported snippets in combination with a TeX rendering engine for a graphical visualization of an existing FSM, or as a figure to be referenced in their academic materials.

Fit Criterion The rendered digram clearly displays all essential information describing the FSM.

Priority High

3 Non-functional Requirements

3.1 Look and Feel Requirements

3.1.1 Appearance Requirements

NF1. The system shall display user interface in the command line

Rationale This is the main use case of the system.

Fit Criterion Users shall be able to use the system at Python interactive shell.

Priority High

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use Requirements

NF2. The system shall provide an interface that is easy to understand and use

Rationale Intended users are familiar with FSM and Python.

Fit Criterion 90% of software engineering student at McMaster University can use the product.

Priority High

3.2.2 Learning Requirements

NF3. The system shall allow users to be able to use quickly.

Rationale Intended users are familiar with Python and CLI.

Fit Criterion 90% of users can learn to use this system within 1 hour.

Priority High

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements

NF4. The system shall respond and execute user's operation within 2.0 seconds

Rationale The product shall not create stalls.

Fit Criterion All outputs for users' operation are generated within 2.0 seconds.

Priority High

3.3.2 Precision or Accuracy Requirements

NF5. The system shall display the same FSM as user's inputs

Rationale The system should display the correct FSMs.

Fit Criterion The displayed FSMs include correct states and transitions as users defined.

Priority High

3.3.3 Robustness or Fault-Tolerance Requirements

NF6. The system shall ignore incorrect/unknown input from the user

Rationale The system must prevent the occurrence of fault.

Fit Criterion Invalid inputs from users are not accepted by the system.

Priority High

NF7. The system shall have zero fault-tolerant

Rationale Users must know when they made an error.

Fit Criterion A warning message is displayed when there is an error.

Priority High

3.3.4 Scalability or Extensibility Requirements

NF8. The system shall be able to add new extensions

Rationale Further improvements and extensions of the system is needed.

Fit Criterion Not applicable.

Priority High

3.3.5 Longevity Requirements

NF9. The system shall be able to run till April 30, 2022

Rationale This is the minimum live span of the system.

Fit Criterion The system is able to run without being maintained.

Priority Medium

3.4 Operational and Environmental Requirements

3.4.1 Expected Physical Environment

NF10. The system shall run on any laptop/desktop with Linux/Windows systems

Rationale Users are able to use the system on these devices and systems.

Fit Criterion The system operates on these devices and systems.

Priority Medium

3.4.2 Release Environment

NF11. The system shall be available before April 1, 2022

Rationale This is the deadline of the project set by the client.

Fit Criterion The system is able to operate no later than April 1, 2022.

Priority Low

3.5 Maintainability and Support Requirements

3.5.1 Maintenance Requirements

NF12. The system shall take no longer than 4 hours to be maintained/updated

Rationale Not applicable.

Fit Criterion Not applicable.

Priority Low

3.5.2 Supportability Requirements

NF13. The system shall be able to run on most of the laptop/desktop

Rationale The system should have a low hardware requirement.

Fit Criterion 98% of students at McMaster University are able to run the system on their laptops and PCs.

Priority Low

3.6 Security Requirements

3.6.1 Integrity Requirements

NF14. The system shall not be able to be modified by unauthorized users

Rationale This prevents the system from malfunctions.

Fit Criterion Any updates/modifications to the system shall only be made and reviewed by development team.

Priority High

3.6.2 Privacy Requirements

NF15. The system shall not share users' account information

Rationale Users account information are credential.

Fit Criterion Not applicable.

Priority High

NF16. The system shall not collect any non-essential information from users

Rationale This prevents unauthorized disclosure of users information.

Fit Criterion Not applicable.

Priority High

3.6.3 Audit Requirements

NF17. An audit is required before releasing an update of the system

Rationale Not applicable.

Fit Criterion Not applicable.

Priority Low

NF18. Errors should be fixed after an audit

Rationale Not applicable.

Fit Criterion 80% of Errors are fixed after an audit.

Priority High

3.7 Cultural Requirements

3.7.1 Cultural Requirements

NF19. The system shall not display any offensive cultural symbol/language

Rationale The intended users of the system are considered to have various cultural and religious backgrounds.

Fit Criterion Not applicable.

Priority High

3.7.2 Political Requirements

NF20. The system shall not be associated with any political opinions

Rationale The Not applicable.

Fit Criterion Not applicable.

Priority High

3.8 Legal Requirements

3.8.1 Standards Requirements

NF21. The system shall meet the ISO/IEC 12207

Rationale ISO/IEC 12207 is an international standard for software lifecycle processes.

Fit Criterion The system meet the standard of ISO/IEC 12207.

Priority Medium

3.9 Health and Safety Requirements

NF22. The system shall not cause the computers to overload and explode

Rationale The system has a low hardware requirement.

Fit Criterion The hardware running the system is under normal temperature.

Priority Low

NF23. The system shall not affect user's physical and mental health

Rationale The system must not harm users' health and safety.

Fit Criterion Users feel comfortable using the system in various situations.

Priority High

4 Project Issues

4.1 Open Issues

- The most difficult part of developing the system is to implement the LaTeX exportation feature. Developing a system to mechanically generate LaTeX snippets require extensive knowledge of both the LaTeX mark-up language as well as the graphing library, `tkiz`.
- Testing for the LaTeX exportation feature is considered to be difficult.

4.2 Off-the-Shelf Solutions

- There is no existing software that can provide a solution to the implementation of the LaTeX exportation feature. This issue is planned to be resolved within the project development team.
- To test the exportation output, an additional dependency, the LaTeX rendering back-end would be needed. However, manual inspection of the rendered result would also be necessary.

4.3 New Problems

Some of the users may be not familiar with command line interface. In this case, it would be difficult for those to use the software.

4.4 Tasks

~~See Gantt Project file under ProjectSchedule directory.~~
Gantt Project on GitLab

4.5 Migration to the New Product

Not Applicable

4.6 Risks

~~Not Applicable~~
One of the potential risks associated with the project is that the testing for

the diagram and LaTeX Exportation feature can be complicated and time-consuming. Although the correctness of diagrams is easy to be validated, their appearance can hardly be evaluated by any automation tools. Therefore, we expect to use a large amount of manual testing and visual inspection of the output.

4.7 Costs

Not Applicable

4.8 User Documentation and Training

The software will include a user manual which contains instructions on how to use the software.

4.9 Waiting Room

Provide a GUI for the software. Users could choose between a command-line interface and graphical user interface to use the software.

4.10 Ideas for Solutions

Each transition could be represented by arrows, each state could be represented by circles. During simulation, the software would highlight the current state/transitions.

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.