

Project 3: Lifecycle Documents with Requirements Tracking

Software Requirements Specification

Version 1.0

11/27/2013

Kevin Palmer, Peter Magalhaes, Ari Ushani,
Jeremiah Butler

Revision History

Date	Description	Author	Comments
11/19/13	Version 1.0	Kevin	Created
12/9/13	Version 1.1	Jeremiah	Add Models
12/9/13	Version 1.2	Peter	Review

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Jeremiah Butler	Team Lead	Nov 27, 2013

Table of Contents

REVISION HISTORY

DOCUMENT APPROVAL

1. INTRODUCTION

- 1.1 PURPOSE
- 1.2 SCOPE
- 1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS
- 1.4 OVERVIEW

2. GENERAL DESCRIPTION

- 2.1 PRODUCT PERSPECTIVE
- 2.2 PRODUCT FUNCTIONS
- 2.3 USER CHARACTERISTICS
- 2.4 GENERAL CONSTRAINTS
- 2.5 ASSUMPTIONS AND DEPENDENCIES

3. SPECIFIC REQUIREMENTS

- 3.1 EXTERNAL INTERFACE REQUIREMENTS
 - 3.1.1 *User Interfaces*
 - 3.1.2 *Hardware Interfaces*
 - 3.1.3 *Software Interfaces*
 - 3.1.4 *Communications Interfaces*
- 3.2 FUNCTIONAL REQUIREMENTS
 - 3.2.1 *<Create Lifecycle Project>*
 - 3.2.2 *<Read Lifecycle Document>*
 - 3.2.3 *<Edit Lifecycle Document>*
 - 3.2.4 *<Export Lifecycle Document>*
 - 3.2.5 *<Generate RTM>*
- 3.3 USE CASES
 - 3.3.1 *Create Lifecycle Document*
 - 3.3.2 *Read Lifecycle Document*
 - 3.3.3 *Edit Lifecycle Document*
 - 3.3.4 *Export Lifecycle Document*
 - 3.3.5 *Generate RTM*
 - 3.3.6 *Quit Program*
- 3.4 CLASSES / OBJECTS
 - 3.4.1 *<Class / Object #1>*
- 3.5 NON-FUNCTIONAL REQUIREMENTS
 - 3.5.1 *Performance*
 - 3.5.2 *Reliability*
 - 3.5.3 *Availability*
 - 3.5.4 *Security*
 - 3.5.5 *Maintainability*
 - 3.5.6 *Portability*
- 3.6 INVERSE REQUIREMENTS
- 3.7 DESIGN CONSTRAINTS
- 3.8 LOGICAL DATABASE REQUIREMENTS
- 3.9 OTHER REQUIREMENTS

4. ANALYSIS MODELS

- 4.1 SEQUENCE DIAGRAMS
- 4.3 DATA FLOW DIAGRAMS (DFD)

4.2 STATE-TRANSITION DIAGRAMS (STD)

5. CHANGE MANAGEMENT PROCESS

A. APPENDICES

A.1 APPENDIX 1

A.2 APPENDIX 2

A.3 APPENDIX 3

A.4 APPENDIX 4

A.5 APPENDIX 5

A.6 APPENDIX 6

A.7 APPENDIX 7

A.8 APPENDIX 8

1. Introduction

The Software Requirements Specification details the extent of NUWC's Lifecycle Project Manager. The product's main feature is its ability to create and manage lifecycle documents using a graphical user interface. The lifecycle documents will be organized and exported using an XML Schema. This can be accomplished by a user who has no knowledge of the XML language. This document will review all the basic functionality required for a user to edit, create, and manage lifecycle projects.

1.1 Purpose

To provide a detailed description of how the product will produce its lifecycle documents as well as edit and export them. It also overviews the basic functional requirements requested by the customer.

1.2 Scope

The Lifecycle Project Manager will create and edit projects and lifecycle documents. It will serve as a cheap but easy to use software that allows users to input their own data and organize based on an XML schema. The product will export the lifecycle documents as PDF and HTML, as well as print out a requirements traceability matrix.

1.3 Definitions, Acronyms, and Abbreviations

API - Application Programming Interface

CSS - Cascading Style Sheet

CSV - Comma Separated Value

GUI - Graphical User Interface

HTML - HyperText Markup Language

JS - Javascript

LC - Lifecycle

NUWC - Naval Undersea Warfare Center Division Newport

PDF - Portable Document Format

RTM - Requirements Traceability Matrix

UMASSD - University of Massachusetts: Dartmouth

XML - Extensible Markup Language

1.4 Overview

The SRS contains explicit details describing how the product is designed and implemented. It is organized in a top-down layered approach. The goal of this project is to create a prototype application capable of allowing a user to productively manage lifecycle documentation in XML without an previous XML experience.

2. General Description

2.1 Product Perspective

Our project team design goal is to develop a software solution to the proposal offered by NUWC. The design will accommodate for previous user complaints and shortcoming. It will also replace manual spreadsheet entries into a database. The software will also automate the process of generating a RTM. It will be a new, self-contained, web-based solution that requires minimal constraints. The design goal is to provide a tabular method of viewing multiple documents simultaneously to promote user productivity.

2.2 Product Functions

The software will prompt the user to open a new project or create a new one. When the user selects new project it presents the user with a set default templates which have been checked against a defined XML schema. These templates will be saved with the filename specified by the user. The user can then edit these templates with their information. The user can define requirements which will be traced throughout other documents in the project. After the documents have been saved, the user can select to create a RTM (Requirement Traceability Matrix) from the XML templates which the user has created and modified.

2.3 User Characteristics

This product is intended to be used by a non-technical user who may have no knowledge of the XML language. There will be no user groups, or administrator authentications. The user will be assumed to have access once the user is able to log into the computer where the software is stored.

2.4 General Constraints

The software will run on a Microsoft Windows 7 platform within Microsoft Internet Explorer browser. The use of a database is prohibited and there is also no Internet connection available. The software must implement an XML schema developed by the team. The schema helps keep the document structured and search-able. By using an XML schema and an XSLT processor, the user will have the ability to export documents as either an HTML or PDF.

2.5 Assumptions and Dependencies

One dependency would be the available browser. If this were to change our application may break or provide an unintended appearance or functionality to the user. The product will create a directory for all of the created/edited LC documents, and the user will have permission on the local machine for the software to function as intended. The team must also assume the user is capable of creating and editing technical documentation. Lastly, one major assumption is user has no knowledge of XML and the user should have not have to see what is happening behind the scenes while creating the project and exporting the documents.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface should be easy to use and have simple text boxes and an active table of contents menu, and a preview of the document selected in a preview pane. Since the user is unaware of the XML back end implementation, a GUI must be driven from XML to allow for easy adaptation.

- 3.1.1.1 - User is unaware of XML back end implementation
- 3.1.1.2 - System shall present a GUI to users that allow them to develop and edit lifecycle documents (i.e Requirements, Design, and Test)
- 3.1.1.3 - Ability to view existing requirements, design elements and test cases shall be available at all times

3.1.2 Hardware Interfaces

The application will be able to operate on a modern Microsoft Windows operating computer, that has an active installation of Microsoft Internet Explorer. Since there will be no network connection available, there is no requirement to interface to a network, however, the software solution will be web-based and address some scalability issues. All other hardware interfacing will be handled by the operating system, or the browser.

- 3.1.2.1 - All hardware interfacing is handled by the browser

3.1.3 Software Interfaces

The XML Schema will be developed by the team, and the RTM will be auto-generated upon request. The XSLT processor will be used to convert the XML lifecycle documentation into HTML or PDF documents. The software will also use Internet Explorer as the main software interface.

- 3.1.3.1 - Must work with Internet Explorer 10
- 3.1.3.2 - Program should be capable of running XSLT processing for HTML and PDF output. An XML instance shall be transformed, via XSLT designed and developed by the team into an HTML-based interactive document
- 3.1.3.3 - System shall provide a mechanism to associate related parts of each document. (e.g. Requirement x is associated with design element y and test case z).
- 3.1.3.4 - System shall produce XML instances conformant with an XML schema designed and developed by the team.
- 3.1.3.5 - System shall be able to auto-generate a Requirements Traceability Matrix at any point during development

3.1.4 Communications Interfaces

The software solution will not need to communicate outside the local machine, since there will may be no network available.

- 3.1.4.1 - Must communicate with local file system to create and edit XML files

3.2 Functional Requirements

Important features of the software project include creation of a Requirement Traceability Matrix (RTM) with the option as exporting as a '.csv' file. Other functionality include user being able to add and implement their own templates for use.

3.2.1 <Create Lifecycle Project>

3.2.1.1 Introduction

Option to create a new LC project.

3.2.1.2 Inputs

Input project name and template selection.

3.2.1.3 Processing

Create project directory. Existing files may be added to project if needed.

3.2.1.4 Outputs

Project directory created. XML template is selected.

3.2.1.5 Error Handling

Javascript form checking.

3.2.2 <Read Lifecycle Document>

3.2.2.1 Introduction

Option to open a LC project.

3.2.2.2 Inputs

Browse for a project name.

3.2.2.3 Processing

Open project directory. Parse XML for requirements, create array of sections for Table of Contents and preview.

3.2.2.4 Outputs

Display Table of Contents and preview

3.2.2.5 Error Handling

Browsing keeps user from entering invalid file location.

XML schema ensures documents can be handled by our system.

3.2.3 <Edit Lifecycle Document>

3.2.3.1 Introduction

Option to edit an open LC document.

3.2.3.2 Inputs

Mouse click on section within Table of Contents.

3.2.3.3 Processing

Selected section is put into and editable text area.

3.2.3.4 Outputs

Section is focused within preview and the given area is visibly now editable.

3.2.3.5 Error Handling

If an error occurs during this processing, the user will be alerted and checked against the schema again.

3.2.4<Export Lifecycle Document>

3.2.4.1 Introduction

Existing LC documents are able to be exported as HTML or PDF file.

3.2.4.2 Inputs

Project where export button is selected will be input, user will select which document they wish to export.

3.2.4.3 Processing

XSLT processor uses XML and XSLT stylesheet to process HTML or PDF export.

3.2.4.4 Outputs

HTML file is exported based on LC document chosen.

3.2.4.5 Error Handling

Javascript form checking, user will be alerted to any errors in processing.

3.2.5 <Generate RTM>

3.2.5.1 Introduction

Requirements Traceability Matrix is generated based off XML requirements.

3.2.5.2 Inputs

Project selection from user.

3.2.5.3 Processing

Find LC documents from locations in project file. Parse requirements to create a HTML table.

3.2.5.4 Outputs

HTML table with down-loadable CSV.

3.2.5.5 Error Handling

User will be alerted to error, document will be rechecked against schema.

3.3 Use Cases

3.3.1 Create Lifecycle Document

3.3.1.1 Brief Description

User is presented with the option to create a Lifecycle project, or new document within an existing project. The user only needs to input a name, location, and template selection.

3.3.1.2 Preconditions

The user has started the program.

A valid template has been selected.

3.3.1.3 Basic Flow of Events

User selects create.

Name, location, and template are input.

Templates are copied to the given location using the name specified.

Read method is called

3.3.1.4 Alternate Flows

If creating a document within an existing project, location is already specified.

3.3.1.5 Sub Flows

Documents are checked against schema for initial integrity.

3.3.1.6 Post Conditions

Project directory is created

Project file is created containing meta-data and additional project file locations.

3.3.1.7 Specific Requirements

User is unaware of XML back end implementation

All hardware interfacing is handled by the browser

Must work with Internet Explorer 10

Must communicate with local file system to create and edit XML files

3.3.1.8 Additional Information

3.3.2 Read Lifecycle Document

3.3.2.1 Brief Description

User selects to open a LC document in a read-only format. A file system search is opened.

3.3.2.2 Preconditions

The document to be opened must already exist and match the XML schema.

3.3.2.3 Basic Flow of Events

Open LC document is clicked.

User is prompted with search of file system.

User selects LC project.

LC document is opened in a read-only mode.

3.3.2.4 Alternate Flows

User could open file which does not match XML schema →prompted with error.

User can click on the read-only document and enable an editing mode.

User can switch tabs which will open a different document within the project.

3.3.2.5 Sub Flows

Documents are checked against schema for integrity.

3.3.2.6 Post Conditions

Document is opened in read-only mode until user clicks on section to be edited.

3.3.2.7 Specific Requirements

User is unaware of XML back end implementation

System shall present a GUI to users that allow them to develop and edit lifecycle documents (i.e Requirements, Design, and Test)

Must work with Internet Explorer 10

System shall produce XML instances conformant with an XML schema designed and developed by the team.

Must communicate with local file system to create and edit XML files

3.3.2.8 Additional Information

Not applicable at this time.

3.3.3 Edit Lifecycle Document

3.3.3.1 Brief Description

User selects a section from the Table of Contents. This section is focused in the preview and put into an editable text area.

3.3.3.2 Preconditions

A LC project has been opened and checked against the given schema.

Table of Contents and preview have been generated and displayed

3.3.3.3 Basic Flow of Events

User selects a section from the Table of Contents

This section is focused and made editable

User makes changes and selects another section or exits

User is prompted to save any changes

3.3.3.4 Alternate Flows

No changes are made

3.3.3.5 Sub Flows

Changes are committed to local file

3.3.3.6 Post Conditions

If files are changed, they still match schema

3.3.3.7 Specific Requirements

User is unaware of XML back end implementation

System shall present a GUI to users that allow them to develop and edit lifecycle documents (i.e Requirements, Design, and Test)

Ability to view existing requirements, design elements and test cases shall be available at all times

All hardware interfacing is handled by the browser

Must work with Internet Explorer 10

Must communicate with local file system to create and edit XML files

3.3.3.8 Additional Information

3.3.4 Export Lifecycle Document

3.3.4.1 Brief Description

LC Document will be exported as an HTML file using the XSLT processor based on the defined XML schema.

3.3.4.2 Preconditions

XML Schema is defined.

LC Document contains text.

3.3.4.3 Basic Flow of Events

XML Schema is defined.

LC Document is opened and being viewed/edited.

User clicks Export LC Document.

User is prompted for name of file to be exported.

XSLT processor uses XML schema to create HTML file.

3.3.4.4 Alternate Flows

User can be asked which export file format is appropriate.

User can choose between HTML and PDF.

3.3.4.5 Sub Flows

Documents are checked against schema for integrity.

3.3.4.6 Post Conditions

HTML file is exported and viewable in browser.

3.3.4.7 Specific Requirements

System shall present a GUI to users that allow them to develop and edit lifecycle documents (i.e Requirements, Design, and Test)

All hardware interfacing is handled by the browser

Must work with Internet Explorer 10

Program should be capable of running XSLT processing for HTML and PDF output.

An XML instance shall be transformed, via XSLT designed and developed by the team into an HTML-based interactive document

Must communicate with local file system to create and edit XML files

3.3.4.8 Additional Information

Not applicable at this time.

3.3.5 Generate RTM

3.3.5.1 Brief Description

A RTM should be generated from the requirements, use cases, and test cases within a project. This will be displayed as an HTML table while giving the user the ability to download it as a CSV.

3.3.5.2 Preconditions

An project is open and the documents follow a given schema

3.3.5.3 Basic Flow of Events

User clicks generate RTM

Requirements document is scanned, assigning an ID to each requirement for easy navigation

Use and test case documents are scanned for these requirements

Found instances are noted on table

HTML and CSV are generated and displayed

3.3.5.4 Alternate Flows

If the desired document does not exist, those cells will be left blank

3.3.5.5 Sub Flows

CSV is generated from table

If button is pressed, download is presented to user

3.3.5.6 Post Conditions

None

3.3.5.7 Specific Requirements

User is unaware of XML back end implementation

System shall present a GUI to users that allow them to develop and edit lifecycle documents (i.e Requirements, Design, and Test)

Must work with Internet Explorer 10

System shall produce XML instances conformant with an XML schema designed and developed by the team.

System shall be able to auto-generate a Requirements Traceability Matrix at any point during development

Must communicate with local file system to create and edit XML files

3.3.5.8 Additional Information

3.3.6 Quit Program

3.3.6.1 Brief Description

Browser close button will exit the LC Project Manager.

3.3.6.2 Preconditions

Software is open.

3.3.6.3 Basic Flow of Events

User decides they wants to exit program.

User clicks close browser.

Browser unload prompts user to save

Program is terminated.

3.3.6.4 Alternate Flows

Not applicable.

3.3.6.5 Sub Flows

Not applicable.

3.3.6.6 Post Conditions

Program is terminated.

3.3.6.7 Specific Requirements

All hardware interfacing is handled by the browser

Must work with Internet Explorer 10

3.3.6.8 Additional Information

Not applicable at this time.

3.4 Classes / Objects

Will be included in next version.

3.4.1 <Class / Object #1>

3.4.1.1 Attributes

3.4.1.2 Functions

3.5 Non-Functional Requirements

3.5.1 Performance

No performance requirements listed. Reasonable access time is ideal: no greater than 10 seconds.

3.5.2 Reliability

Product must run locally. Internet access is not needed.

3.5.3 Availability

Product is available to run on any system with Internet Explorer 10

3.5.4 Security

Product is ran on local machine without Internet access. Prevents any external threats.

3.5.5 Maintainability

The system will be designed by our team and maintained by NUWC.

3.5.6 Portability

The system is a non-install and is portable to any machine running Windows 7 and using Internet Explorer 10.

3.6 Inverse Requirements

Product is not designed to function correctly in any browser other than Internet Explorer 10.

3.7 Design Constraints

Product will run on Windows 7 operating system using IE. Access to the Internet is prohibited.

3.8 Logical Database Requirements

Database access is prohibited. The system will run locally without Internet access.

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

4.1 Sequence Diagrams

See Appendices A1

4.3 Data Flow Diagrams (DFD)

See Appendices A2

4.2 State-Transition Diagrams (STD)

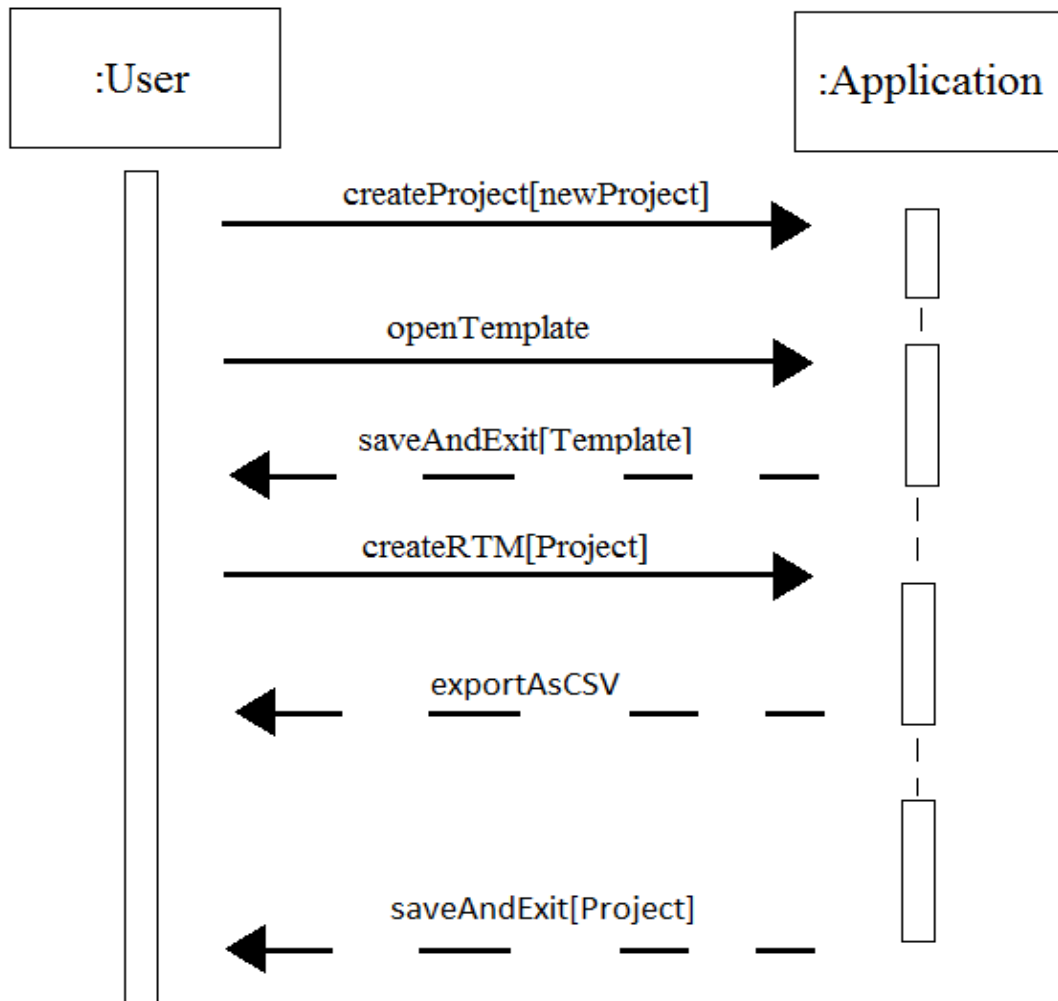
See Appendices A3 - A8.

5. Change Management Process

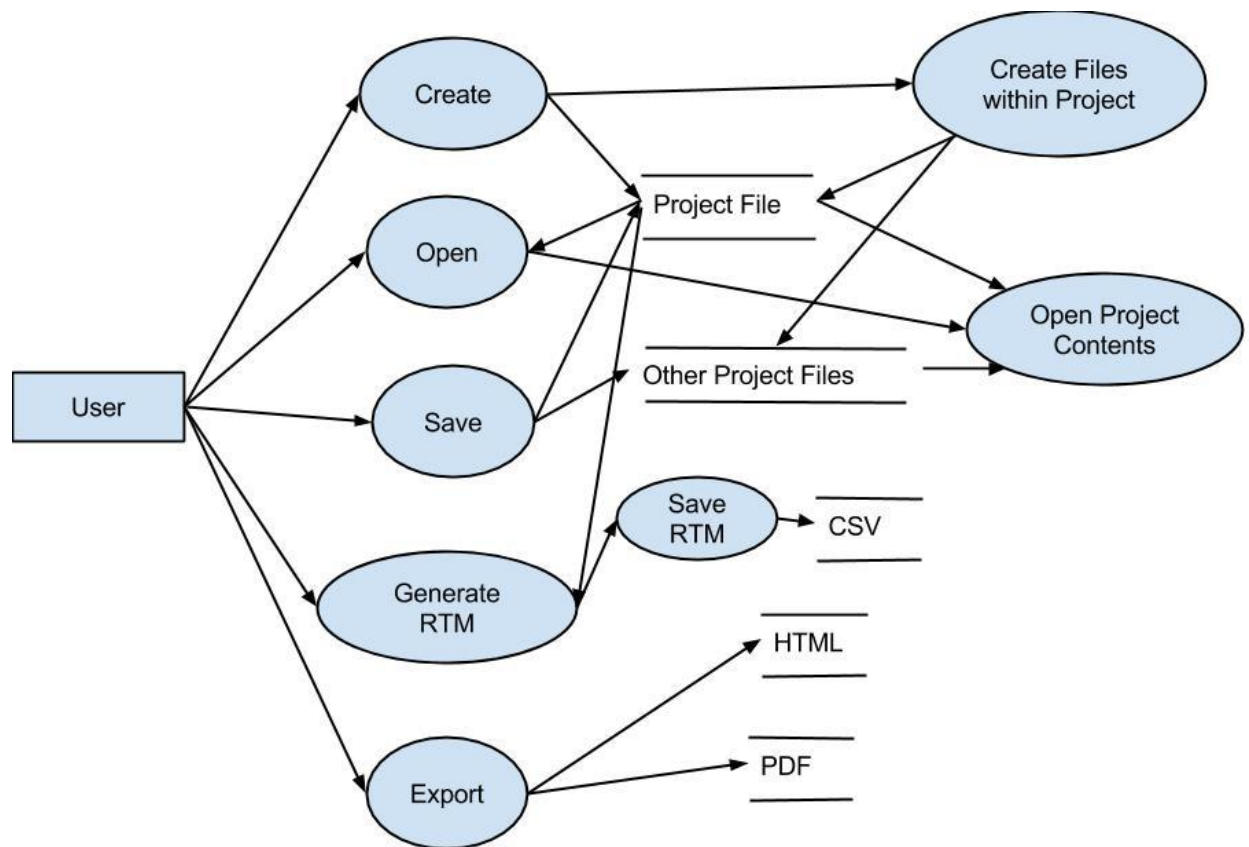
Changes will be acknowledged by changing the version number of the document. Changes will initially be marked in red until changes are approved by the entire team and the team supervisor, Mr. Grimley.

A. Appendices

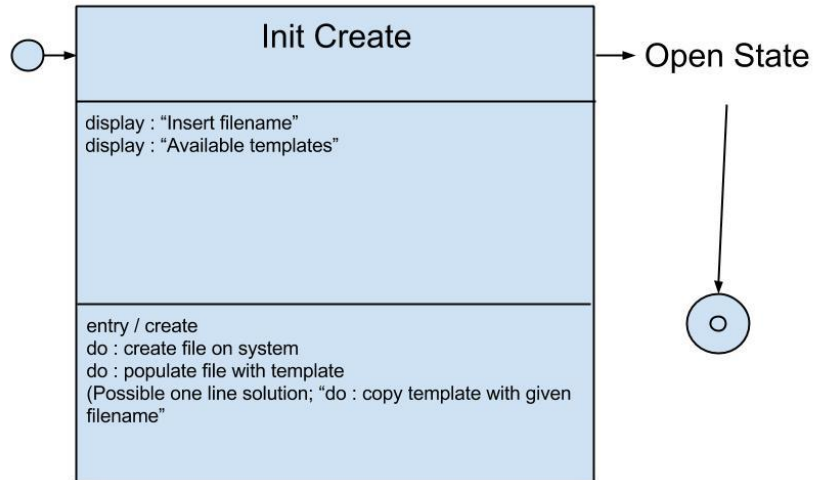
A.1 Appendix 1



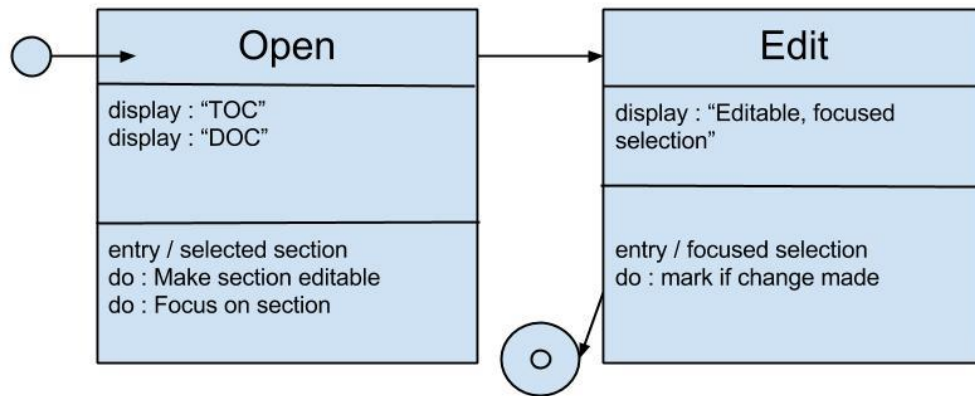
A.2 Appendix 2



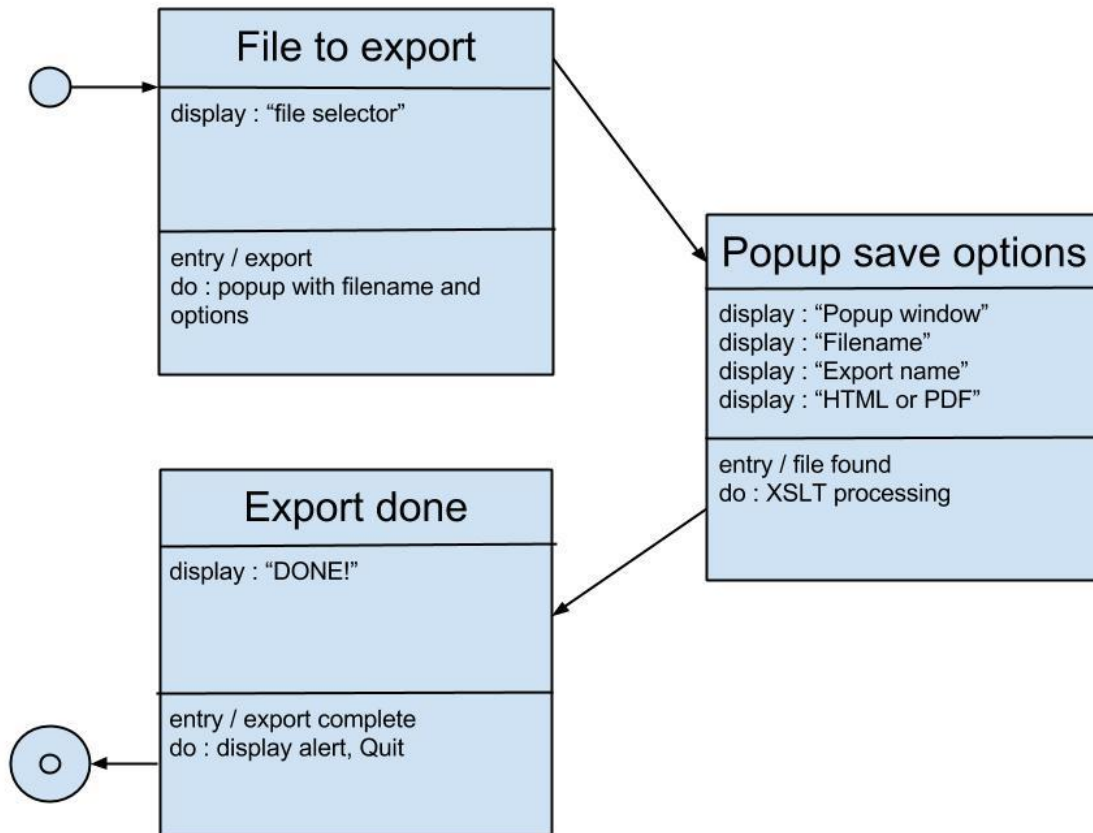
A.3 Appendix 3



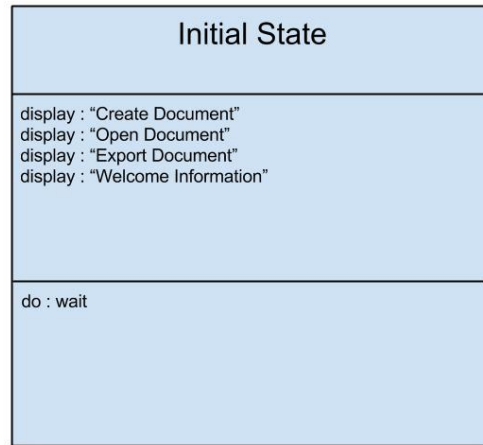
A.4 Appendix 4



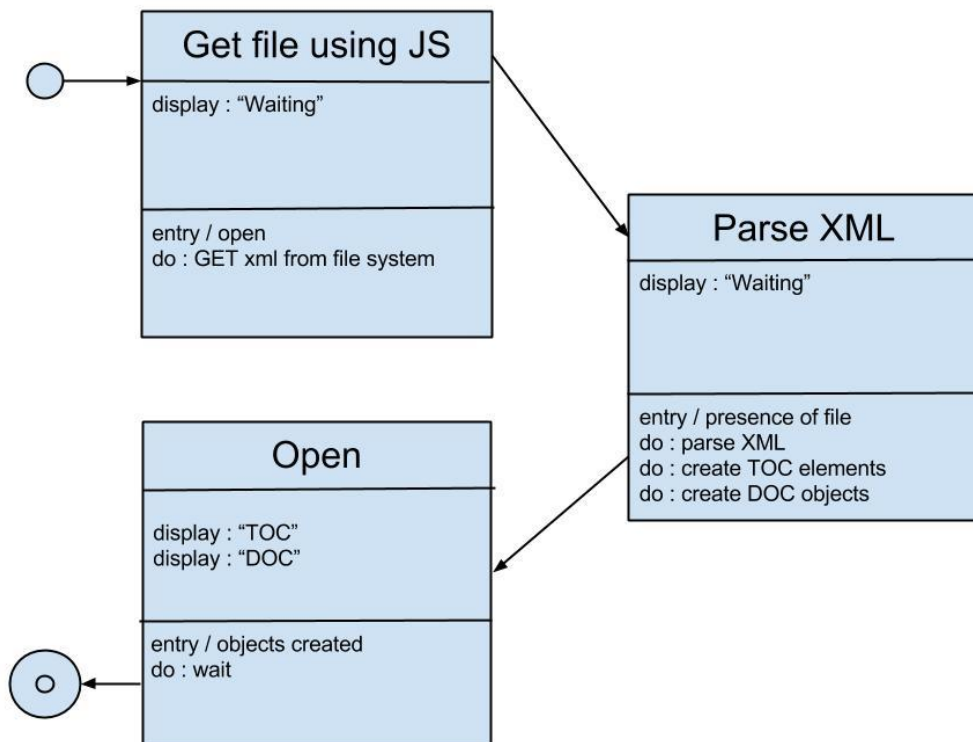
A.5 Appendix 5



A.6 Appendix 6



A.7 Appendix 7



A.8 Appendix 8

Quit
display : "Popup"
entry / browser unload do : check unsaved changes do : confirm quit