Software Requirements Specification

for

LaTeX Code Generator

Government Engineering College, Wayanad

Nisanth Komal P.C.
Amal Wilson

October 12, 2022

Contents

1 Introduction 3
1.1 Purpose 3
1.2 Scope 3
1.3 Document Conventions and Acronyms 3
1.4 References 3
1.5 Overview 4

2 Overall Description 4
2.1 Product Perspective 4
2.1.1 User Interfaces 4
2.1.2 Software Interfaces 4
2.2 Product Functions 5
2.2.1 Create New Report 5
2.2.2 Opening an Existing Document 5
2.2.3 Saving a Document 5
2.2.4 Insertion of Mathematical Content 5
2.2.5 Inserting Graphical Content 5
2.2.6 Aligning the Content 5
2.2.7 Formatting Text 5
2.2.8 Insertion of Environments 6
2.2.9 Completion of Commands 6
2.3 User Characteristics 6
2.4 Constraints 6
2.5 Assumptions and Dependencies 6
2.6 Apportioning of requirements 6

3 Specific Requirements 7
3.1 Functional Requirement Specification 7
3.1.1 Creation of New Document 7
3.1.2 Open File 8
3.1.3 Save Document 9
3.1.4 Insertion of Mathematical Symbols 9
3.1.5 Insertion of Mathematical Equations 10
3.1.6 Insertion of Mathematical Constructs 10
3.1.7 Insertion of Matrices 11
3.1.8 Insertion of Derivatives 11
3.1.9 Code Completion for Escape Sequences 12
3.1.10 Closing an Environment 12
3.1.11 Formatting Text 12
3.1.12 Insertion of Numbering or Bulletting environments 13
3.1.13 Insert Chapter 13
3.1.14 Insertion of Sections 14
3.1.15 Insertion of Footnote 14
3.1.16 Insertion of Images 15
3.1.17 Insertion of Tables 15
3.1.18 Insertion of Hyperlinks 16
3.1.19 Insertion of Title Page 16
3.1.20 Insertion of Abstract Page 17
3.1.21 Insertion of Table of Contents Page 17
3.2 Non-functional Requirements 17

3.2.1 Performance Requirements 17
3.2.2 Portablitiy 17
3.2.3 Security Requirements 18
3.2.4 Software Quality Attributes 18

4 Additional models and reqirements based on them 18

1 Introduction

Keep this Wiki link as a reference on a tab in your browser to add more content
link
or refer to the set SRS of IEEE full set appropriate to your project. The version below has a selected sections from that document which suit this project's requirements; may be you don't need all of them; may be you need items not included here. Wherever possible use UML diagrams.
Delete this note in final document

1.1 Purpose

The purpose of this document is to give a detailed description of LaTeX Code Generator. This document will explain purpose and features of the system and identifies its users. It is intended for both the designers, developers as well as end-users of the system.

1.2 Scope

The LaTeX Code Generator will be a platform used for generating LaTeX source code for preparing reports. This product creates .tex file for generating the report, rather than generating the report itself . The user should compile the code to get the report. The system will be designed to :

- preparing technical reports with mathematical content.
- provide platform independency and maintainability.

The product allows the user to concentrate on the content instead of spending time on commands and its syntax.

1.3 Document Conventions and Acronyms

Term	**Defenition**
Typesetting	The process of putting text into the right style and size of type and the desired arrangement on the page
LaTeX	A document preparation system for high-quality typesetting. It is a markup/programming language
Product / Software	Product and software is used interchangeably to denote the LaTeX Code Generator
User	A person who writes a technical documents or technical reports
LaTeX Environment	A LaTeX code segment of the form \begin{environment-name} ... \end{environment-name}
Preamble	The section before the \begin{document} command in a LaTeX document. Contents in this section will not appear in the document. This section holds the configuration commands of the document.

1.4 References

1. LaTeX Tutorials A primer-Indian TEX Users Group,Trivandrum, India 2003 September
2. http://en.wikibooks.org/wiki/LaTeX
3. IEEE Recommended Practice for Software Requirements Specifications IEEE-SA Standards Board.
4. http://dev.midnightcoding.org/projects/gummi

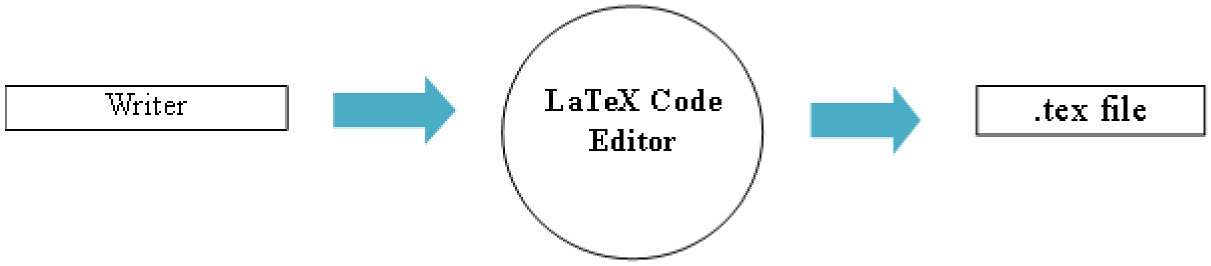
1.5 Overview

Section 2 of this document gives an overall description about the product.It describes the general functionalities,expected user groups and constraints of the product. Section 3 gives more specific information about the functionalities specified in section 2 . Section 2 is a general view of the system and should be used as a guidance to section 3. Section 3 is intended for developers and testers and may be skipped by end users.

2 Overall Description

2.1 Product Perspective

This product is an editor and code generator for LaTeX typesetting system. This is a alternative for the existing system Gummi - a source code editor for LaTeX. It wll add support for mathematical content, which is not present in the existing product. The product will help user with code completion for commonly used commands.



2.1.1 User Interfaces

The application will have a plain text editor interface. LaTeX will be highlighted with a different color. Buttons and dialog boxes for insertion of LaTeX code segments will be available. The user can start a new LaTeX report document from the menu or using a keyboard shortcut. The editor allows opening an existing .tex file for editing. Menus will be provided for insertion of code segments for title page, chapters and sections within the started/opened document. Code segments for tables and images can be inserted using dialog boxes. Code for mathematical symbols and constructs can be inserted with the help of buttons. The document can be saved as a .tex file.

2.1.2 Software Interfaces

LaTeX Code Generator rquires the following software products Java Runtime Environment Version 7 or Higher. Which can be obtained from https://java.com/getjava

This product uses APIs from the following libraries

1. jsyntaxpane
   jsyntaxpane will provide syntax highlighting and line numbering functionalities.
   Source : https://code.google.com/p/jsyntaxpane/

2. jortho
   jortho is a spellchecker with suggestion for correct words.
   Source : http://jortho.sourceforge.net/

2.2 Product Functions

This section outlines the use cases of the user. Specific functionalities will be discussed in section 3.

2.2.1 Create New Report

The user can create a new technical report. When creating the new document, user can select options like paper size, font size etc.. All the commands required for starting the new LaTeX report document with the selected options will be inserted into the editor. The user can work with atmost one document at a time. The current document - if any, will be closed after saving.

2.2.2 Opening an Existing Document

User can open any existing LaTeX document and can continue edit it. Opening a document closes the current document in the editor.

2.2.3 Saving a Document

When the user makes a change in a document the save menu will become enabled. User can save current file in .tex format.

2.2.4 Insertion of Mathematical Content

Mathematical content includes equations, mathematical symbols and mathematical structures and constructs like integrals, derivatives, fractions, matrices etc. The user can insert any of the provided mathematical content by clicking the corresponding button in a toolbar. Dialog boxes will be provided if the inserted content has optional parameters or if additional information is required. The code segment will be inserted by the software at the current cursor position

2.2.5 Inserting Graphical Content

The user can insert code for insertion of images. The image can be selected with the help of a file browser. $\LaTeX$ enviornment for inserting the selected image wll be inserted by the software at the cursor position.

2.2.6 Aligning the Content

Alignment includes aligning the text left, right or centered. User can select the text need to be aligned and click on an alignment button. The software will surround the text with proper commands.

2.2.7 Formatting Text

The following formatting operations can be performed on selected text.

- changing the size of selected text
- making text bold, italic, emphasized or underlined
- making text superscript or subscript

each operation can be performed by the user by selecting the text and clicking the corresponding button. The text will be surrounded by proper LaTeX commands.

2.2.8 Insertion of Environments

User can insert any of the following environments and the code will be generated by the editor

- insertion of tables : user will be asked for additional parameters like number of columns and rows
- insertion of framed boxes : user will be asked for additional parameters like dimensions of the box
- insertion of hyperlinks

2.2.9 Completion of Commands

After the user types the beginning of a LaTeX environment the user can use a key combination to automatically insert the end statement of that environment. The use can replace a character with its escape command using the same key combination.

2.3 User Characteristics

The users of this system are those who writes technical reports or technical documents. The user must be familiar with any text editing software like gedit or notepad. Two types of users are

1. Beginners in LaTex
   These are the major user group for LaTeX code generator. User under this category will be new to the LaTeX typesetting system or those who started to learn LaTeX. They must be able to compile a .tex file using compilation commands. They need not be aware about the basic structure of a LaTeX file. The knowledge is advised for better use of the product. They are not expected to be knowing about the commands used for styling or alignment of text, insertion of various symbols, insertion of mathematical structures etc. Knowledge about LaTeX packages or optional parameters of commands are also not required.

2. Experienced Users in LaTeX
   Users in this category must have written documents in LaTeX without the help of any LaTeX code generators. They are expected to write commands manually.

2.4 Constraints

Every component of the system should be written in Java. The source code must follow Java 7 API and should not use any depecrated APIs.

2.5 Assumptions and Dependencies

None

2.6 Apportioning of requirements

The following requirements shuld be provided as increments on future versions and need not be included in the first release. This is to ensure the delivery requirements of the product. No more specific details for these requirements will be provided in section 3.

1. Preview of the Code
   The user can write LaTeX code in the editor manually or by using the tools provided by the software. The preview shows the user how the document will look after compilation.

Changes in preview should be made available as the user makes changes in the LaTeX code. This feature will be called live/realtime preview. This feature depends on a LaTeX compiler.

2. Exporting as PDF
   The source code can be exported to PDF or DVI or Post Script formats. This feature depends on a LaTeX compiler.

3. Error detection
   Errors present in the source code should be identified and should be marked in the source code with inforamative message. This feature depends on a LaTeX compiler.

3 Specific Requirements

3.1 Functional Requirement Specification

This section describes each functionalities and their flow of events in detail. The actor in each use case is always the 'User', unless specified explicitily.

3.1.1 Creation of New Document

Use case name : Start a new report document

Objective : The user selects new document option and the software inserts the LaTeX commands for the new document into the editor

Priority : High

Precondition : Editor window is empty

PostConditions : LaTeX commands inserted in the editor with cursor positioned at content section of the LaTeX document.

Flow of Events :

1. Basic Flow
   - 1.1. Customer selects the new report option from file menu or clicks on the new file button in toolbar
   - 1.2. The system shows new report dialog box with options : paper size, font size, pagenumbering style, number of columns, title page placement, headers & margins and two buttons create new document and create blank document
   - 1.3. User enters the options for the new document
   - 1.4. User clicks 'Create new document'
   - 1.5. Software disoposes the dialog box
   - 1.6. Software inserts latex commands into the editor
   - 1.7. Software places cursor between \begin{document} and \end{document} commands
2. Alternate Flow 1
   - 2.1. After step 1.2 user selects 'Create blank document'
   - 2.2. Software disposes the dialog box
   - 2.3. Software places cursor at beginning position of the editor

3.1.2 Open File

Use case name	:	Opening a LaTeX file
Objective	:	The user opens the document and the file contents appear in the editor
Priority	:	High
Precondition	:	Editor window is empty
PostConditions	:	The document is loaded in the editor.
Flow of Events	:	

1. Basic Flow
    1.1. User selects open from file menu or click on open button in the toolbar
    1.2. Software shows a open file dialog
    1.3. User selects a file
    1.4. User click on open
    1.5. Software reads file
    1.6. Software writes read content to editor
2. Alternate Flow 1
    2.1. After step 1.4 the selected file is not a .tex file
    2.2. An error message is displayed
    2.3. Software proceeds to 1.2

3.1.3 Save Document

Use case name : Save Active Document

Objective : The user saves the current document with a .tex extension

Priority : High

Precondition : Editor contains the document to be saved

PostConditions : The document is saved into disk and the window content remains unchanged

Flow of Events :

1. Basic Flow
   - 1.1. User selects save from file menu or clicks save button in toolbar
   - 1.2. Software shows save dialog box
   - 1.3. User enters filename
   - 1.4. Usr clicks save
   - 1.5. Software creates a file with entered filename
   - 1.6. Software writes contens to the created file
2. Alternate Flow 1
   - 2.1. After step 1.4 the user entered file already exists
   - 2.2. A confirmation box for overwriting the file is selected
   - 2.3. User clicks Yes
   - 2.4. Software overwrites contents in the file in the disk with new contents
3. Alternate Flow 2
   - 3.1. After step 1.1 file is already created or the file is an existing one opened by the software
   - 3.2. Software rewrites existing file with new contents

3.1.4 Insertion of Mathematical Symbols

Use case name : Insert Mathematical Symbol

Objective : The user clicks on the symbol in the toolbar and the software inserts the LaTeX command for clicked symbol

Priority : High

Precondition : Cursor is placed at the desired position

PostConditions : LaTeX command for selected symbol is inserted at the current cursor position

Flow of Events :

1. Basic Flow
   - 1.1. User places cursor at a position to insert the symbol
   - 1.2. User clicks the 'Symbols' button or selects 'Insert Mathematical Symbol' option from menu
   - 1.3. Software displays a toolbar with available symbols
   - 1.4. User clicks on a symbol
   - 1.5. Software inserts the code for the clicked symbol at cursor position

3.1.5 Insertion of Mathematical Equations

Use case name : Insert Mathematical Equation Environment

Objective : The user selects insert equation option from menu and the software inserts the LaTeX command for equation environment

Priority : High

Precondition : Cursor is placed at the desired position

PostConditions : LaTeX command for the equation environment is inserted at the current cursor position

Flow of Events :

1. Basic Flow

   1.1. User places cursor at the desired position

   1.2. User selects the insert equation option from menu

   1.3. Software displays a dialog box for enabling or disabling equation numbering

   1.4. User selects selects an option

   1.5. User clicks insert

   1.6. Software inserts the code for selected options at the current cursor position

3.1.6 Insertion of Mathematical Constructs

Use case name : Insert Mathematical Construct

Objective : user clicks on a construct button from toolbar and the software inserts the code segment for the construct at the cursor position.

Priority : High

Precondition : Cursor is placed at the desired position

PostConditions : LaTeX command for the selected construct is inserted at the current cursor position

Flow of Events :

1. Basic Flow

   1.1. User clicks on a construct button in the constructs toolbar

   1.2. Software inserts the command for the selected construct at cursor position

3.1.7 Insertion of Matrices

Use case name : Insert Matrix Environment

Objective : user clicks matrix button from constructs toolbar and the software inserts the code segment for the matrix environment at the cursor position.

Priority : High

Precondition : Cursor is placed at the desired position

PostConditions : LaTeX command for matrix environment is inserted at the current cursor position. A 'usepackage' command is inserted in the preamble

Flow of Events :

1. Basic Flow
   - 1.1. User clicks the insert matrix button in the mathematical constructs toolbar
   - 1.2. Software shows insert matrix dialog box
   - 1.3. User enters number of rows columns and selects the delimiter style
   - 1.4. User clicks insert button.
   - 1.5. Software inserts the usepackage command in the preamble
   - 1.6. Software generates the command for matrix environment
   - 1.7. Software inserts the commands at cursor position

3.1.8 Insertion of Derivatives

Use case name : Insert Command for Differential Equation

Objective : user clicks derivative button from constructs toolbar and the software inserts the code segment for derivative at the cursor position.

Priority : High

Precondition : Cursor is placed at the desired position

PostConditions : LaTeX command for displaying a differential equation is inserted at the current cursor position

Flow of Events :

1. Basic Flow
   - 1.1. User clicks on the insert derivative button in the constructs toolbar
   - 1.2. Software shows insert derivative dialog box
   - 1.3. User selects type of derivative and enters the variable with respect to derivation happens
   - 1.4. User clicks insert button
   - 1.5. Code for inserting a derivative is inserted at the current cursor position

3.1.9 Code Completion for Escape Sequences

Use case name : Replace Characters with Escape Sequences

Objective : user presses the key combination control + enter after placing the cursor after the characters character if necessary is replaced with the corresponding escape sequence code. Characters requiring escaping are ˜,ˆ, &, $, %, {, } and \

Priority : Normal

Precondition : Cursor is placed after the character to be escaped

PostConditions : The charecter before cursor is replaced with corresponding escape sequence if the character need to be escaped

: 1. Basic Flow

   1.1. User places the cursor after a character

   1.2. User presses control + enter

   1.3. The character before cursor with is replaced with its escape sequence code

3.1.10 Closing an Environment

Use case name : Close a LaTeX Environment

Objective : User types the places the cursor after the \begin{environment} statement and presses ctrl + enter and the \end{environment} statements is entered by the software

Priority : Normal

Precondition : Cursor is placed after a \begin{environment} statement

PostConditions : \end{environment} statement is inserted after the \begin{environment} statement. Cursor is placed between \{begin} and \{end} statements

: 1. Basic Flow

   1.1. User places the cursor after the \begin{environment} statement

   1.2. User presses ctrl + enter

   1.3. Software inserts \end{environment} statement on next line

   1.4. Software places cursor after \begin{environment} command.

3.1.11 Formatting Text

Use case name : Code Generation for Formatting Commands

Objective : User selects the text to be formatted and clicks on a formatting button and the selected text is surrounded by LaTeX code for formatting the text

Priority : Normal

Precondition : Text to be formatted is selected in the editor

PostConditions : Selected text is surrounded by formatting command

: 1. Basic Flow

   1.1. User selects the text to be formatted

   1.2. User clicks on a formatting button

   1.3. Software replaces selected text with text & code for formatting

3.1.12 Insertion of Numbering or Bulletting environments

Use case name	:	Insertion of Numbering or Bulletting Environment
Objective	:	User clicks on the numbering or bulletting button and the corresponding environment is inserted at cursor position
Priority	:	High
Precondition	:	Cursor is placed at the desired position or the text is selected
PostConditions	:	Code for the selected environment is inserted at the cursor position
	:	

1. Basic Flow
   - 1.1. User places cursor at desired position
   - 1.2. User clicks on a numbering or bulletting environment
   - 1.3. Software inserts the code for the selected environment at cursor position
2. Alternate flow
   - 2.1. User selects the text
   - 2.2. User clicks on a numbering or bulleting environment
   - 2.3. Software replaces the text with the code for the selected environment with text placed inside the environment

3.1.13 Insert Chapter

Use case name	:	Insert Code for Starting a New Chapter
Objective	:	User selects insert chapter from menu and the code for starting the chapter is inserted at cursor position
Priority	:	Normal
Precondition	:	Cursor is placed at desired position
PostConditions	:	Code for chapter is inserted at cursor postion
	:	

1. Basic Flow
   - 1.1. User places the cursor at desired position
   - 1.2. User selects insert chapter from menu
   - 1.3. Software shows insert chapter dialog box
   - 1.4. User enters chapter name
   - 1.5. User clicks insert
   - 1.6. Software inserts code for chapter at cursor position

3.1.14 Insertion of Sections

Use case name	:	Insert Code for Starting a New Section
Objective	:	User selects insert section from menu and the code for section is inserted at cursor position
Priority	:	Normal
Precondition	:	Cursor is placed at desired position
PostConditions	:	Code for chapter is inserted at cursor postion
	:	1. Basic Flow

1. Basic Flow
   - 1.1. User places the cursor at desired position
   - 1.2. User selects insert section from menu
   - 1.3. Software shows insert section dialog box
   - 1.4. User enters section name
   - 1.5. User selects section depth section, subsection or subsubsection, enable or disable numbering and add/remove in table of contents options
   - 1.6. User clicks insert
   - 1.7. Software inserts code for chapter at cursor position

3.1.15 Insertion of Footnote

Use case name	:	Insert Code for Footnotes
Objective	:	User selects insert footnote from menu and the code for footnote is inserted at cursor position
Priority	:	Normal
Precondition	:	Cursor is placed at desired position
PostConditions	:	Code for footnote is inserted at cursor postion
	:	1. Basic Flow

1. Basic Flow
   - 1.1. User places the cursor at desired position
   - 1.2. User selects insert footnote from menu
   - 1.3. Software shows insert footnote dialog box
   - 1.4. User enters footnote content
   - 1.5. User clicks insert
   - 1.6. Software inserts code for footnote at cursor position

3.1.16 Insertion of Images

Use case name : Insert Code for Insertion of Images

Objective : User selects insert image from menu and the code for insertion of image is inserted at cursor position

Priority : Normal

Precondition : Cursor is placed at desired position

PostConditions : Code for insertion of image is inserted at cursor postion. The ’usepackage’ command is inserted in the preamble

:

1. Basic Flow

   1.1. ser places the cursor at desired position

   1.2. User selects insert image from menu

   1.3. Software shows insert image dialog box

   1.4. User clicks browse image button

   1.5. Software shows image browse dialog

   1.6. User selects image

   1.7. User enters options of images in insert image dialog box

   1.8. User clicks insert

   1.9. Software inserts code usepackage in the preamble

   1.10. Software inserts code for image at cursor position

3.1.17 Insertion of Tables

Use case name : Insert Code for Table

Objective : User selects insert table from menu and the code for table is inserted at cursor position

Priority : Normal

Precondition : Cursor is placed at desired position

PostConditions : Code for table is inserted at cursor postion

:

1. Basic Flow

   1.1. User places the cursor at desired position

   1.2. User selects insert table from menu

   1.3. Software shows insert table dialog box

   1.4. User enters number of rows, columns and selects the borders of the table

   1.5. User clicks insert

   1.6. Software inserts code for table at cursor position

3.1.18 Insertion of Hyperlinks

Use case name : Insert Code for Inserting Hyperlinks

Objective : User selects insert hyperlink from menu and the code for hyperlink is inserted at cursor position

Priority : Normal

Precondition : Cursor is placed at desired position

PostConditions : Code for hyperlink is inserted at cursor postion

: 1. Basic Flow

   1.1. User places the cursor at desired position

   1.2. User selects insert hyperlink from menu

   1.3. Software shows insert hyperlink dialog box

   1.4. User selects the type of link email or web link

   1.5. User enters the link address

   1.6. User clicks insert

   1.7. Software inserts code for hyperlink at cursor position

3.1.19 Insertion of Title Page

Use case name : Insert Code for Making a Titile Page

Objective : User selects insert titlepage from menu and the code for titlepage is inserted at the beginning of content section

Priority : Normal

Precondition : Document with all proper LaTeX document structure in editor

PostConditions : Code for titlepage is inserted at beginning of the content section

: 1. Basic Flow

   1.1. User selects insert title page from menu

   1.2. Software shows insert title page dialog box

   1.3. User enters title, author, and date

   1.4. User clicks insert

   1.5. Software inserts code for title page at the beginning of content section

3.1.20 Insertion of Abstract Page

Use case name	:	Insert Code for Inserting an Abstract Page
Objective	:	User selects insert abstract page from menu and the code for abstract page is inserted at the cursor position
Priority	:	Normal
Precondition	:	Cursor placed at the desired position
PostConditions	:	LaTeX code for abstract page is inserted at the cursor position
	:	1. Basic Flow 1.1. User places the cursor at desired position 1.2. User selects insert abstract page from menu 1.3. Software shows insert abstract page dialog box 1.4. User enters the abstract content 1.5. User clicks insert 1.6. Software inserts code for abstract page at cursor position

3.1.21 Insertion of Table of Contents Page

Use case name	:	Insert Code for Creating Table of Contents Page
Objective	:	User selects insert table of contents page from menu and the code for table of contents page is inserted at the beginning of content section of the document
Priority	:	Normal
Precondition	:	Document with proper LaTeX document structure in editor
PostConditions	:	Code for table of contens is inserted at the beginning of content section
	:	1. Basic Flow 1.1. User selects insert title page from menu 1.2. Software shows insert table of contents page dialog 1.3. User clicks insert 1.4. Software inserts code for table of contents page at the beginning of content section 1.5. Software inserts code for footnote at cursor position

3.2 Non-functional Requirements

3.2.1 Performance Requirements

The system should generate and insert the command within 2 seconds after user clicks on a button. An existing file with a size of 1 KB or less should be opened and loaded into the editor within 3 seconds after clicking the open command. No more than 2seconds should be taken for each additional 1 KB in the file size.

3.2.2 Portablitiy

The product should not require any change in source code while porting. Every function of software should behave exactly the same, independent of the platform. To ensure this Java must be selected as the development language.

3.2.3 Security Requirements

3.2.4 Software Quality Attributes

4 Additional models and reqirements based on them

These were sample information

Add more based on chapter 7,8,9,10 of Pressman 8th edn; chapter 3 of text by Deiter and Schmidt; chapter 4,5 of Sommervillie

Based on the examples in text, decide what to do with the work in hand; find how new information(say the type of modeling you do - above content is from the stand point of modeling of the functional/non-functional aspects) can be incorporated to this standard document and then give it

The more you see it, better will be your design