SOFTWARE REQUIREMENT SPECIFICATION (SRS)

DATA STRUCTURE SIMULATOR

Version 1.0

Safiya Mehmood (CS-17006)

Fatima Azim (CS-17013)

Husun Bano (CS-17015)

Department of Computer and Information Systems
NED University of Engineering and Technology

Submitted to
Ms. Fakhra Aftab

Novemeber 21, 2019

Contents

Chapter 1

Introduction

The Data Structure Simulator is a simulation tool that provides insight to the students with a computer educational background. It basically stimulates the data structures in an animated way for better understanding. In this section, a brief overview of the software such as the scope of the product, intended audience, acronyms and an overall document overview is mentioned in detail.

1.1 Document Purpose

The document describes the requirements and specifications of the Data Structures Simulator. It explains the functional features of the three data structures that we will explore; stack, Linked list and Graphs, along with interface details, design constraints and related considerations. The main purpose of this web application is to help undergraduate students to understand the underlying principles of basic data structures. It enables them to visualize the several operations that can be performed on the data structures.

1.2 Product Scope

The tool will graphically represent Stack, Linked lists and Graphs. There will be animations that will show how objects are deleted, the manner of adding new objects, or how one can initialize any of the aforementioned data structures. The web app facilitates teachers in graphically demonstrating data structures and the different operations that can be performed on them. The tool can be used to show common mistakes students make and clear up any misconceptions or fallacies about the chosen data structure. Students can also independently use the tool to self-study data structures, where they may further clarify and solidify their concepts. It can be used to solve complex

homework problems. The tool can also be used without an instructor since it is user-friendly and easy to use. The main purpose for the development of DSS is to utilize the concept the visual learning to aid students to grasp the basic concepts in a better way.

1.3 Intended Audience and Document Overview

This document is intended to serve as a guide to course instructors, course students, independent (self-study) students and professors evaluating the system. It can be used as a guiding tool in schools, universities, any educational institution or even at home. The rest of this SRS is organized as follows:

- Section 2 provides an overview description of the software. It gives the proficiency level to be expected of the user, some general constraints, assumptions and dependencies that are presumed while making the software. It gives a basis to establish the technical requirements in the next chapter.
- Section 3 contains most important features presented with detailed description, and requirements. It gives specific requirements which the software is expected to deliver. Functional requirements are given in this section along with the External Interface Requirements. A Use case Diagram is also illustrated to give a clear idea of the software to be developed.
- Section 4 specifies the Non-Functional requirements. Performance, safety and security requirements are mentioned over here. In addition, Software Quality Attributes have been discussed in detail.

All sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

1.4 Definitions, Acronyms and Abbreviations

- **BFS:** Breadth First Search-Breadth-An algorithm for traversing or searching tree or graph data structures. It starts at the tree root, and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.

- **Course Student:** Students that are enrolled in a data structure or relevant course and are studying under a specific instructor that is using the tool.
- **DFD:** Data Flow Diagram.
- **DFS:** Depth First Search-An algorithm traverses a graph in a depthward motion and uses a stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration
- **DSS:** Data Structure Simulator.
- **Graph:** A non-linear data structure consisting of nodes and edges.
- **GUI:** Graphical User Interface.
- **Independent Student:** Students who are not enrolled in a course where the instructor is using the simulator.
- **JavaScript:** JavaScript, often abbreviated as JS, is a high-level, interpreted scripting language
- **Latex:** A document formatting tool to prepare documents.
- **Linked list:** A data structure consisting of a collection of nodes which together represent a sequence.
- **MySql:** MySQL is an open-source relational database management system.
- **Node:** A single item (object) in a Linked list or Graph.
- **OS:**Operating System.
- **Stack:** A linear data structure which follows a particular order in which the operations are performed. The order may be LIFO.
- **WebApp:** It is a client–server computer program that the client (including the user interface and client-side logic) runs in a web browser.
- **WYSIWYG:**What You See Is What You Get
- **Xampp:** XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages

1.5 Document Conventions

The SRS has been prepared using the Latex software which is a document preparation system. When writing, the writer uses plain text as opposed to the formatted text found in WYSIWYG word processors.

1.6 References and Acknowledgments

- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- Bruade: The principal source of textbook material is "Software Engineering: An Object-Oriented Perspective" by Eric J. Bruade (Wiley 2001).
- Reaves SPMP: "Software Project Management Plan Jacksonville State University Computing and Information Sciences Web Accessible Alumni Database." Jacksonville State University, 2003

Chapter 2

Overall Description

2.1 Product Perspective

The data structure simulator is a new, self-contained system which will be of great use to students and teachers alike which features the visualization of data structures. The below figures shows the demonstration of the simulator.

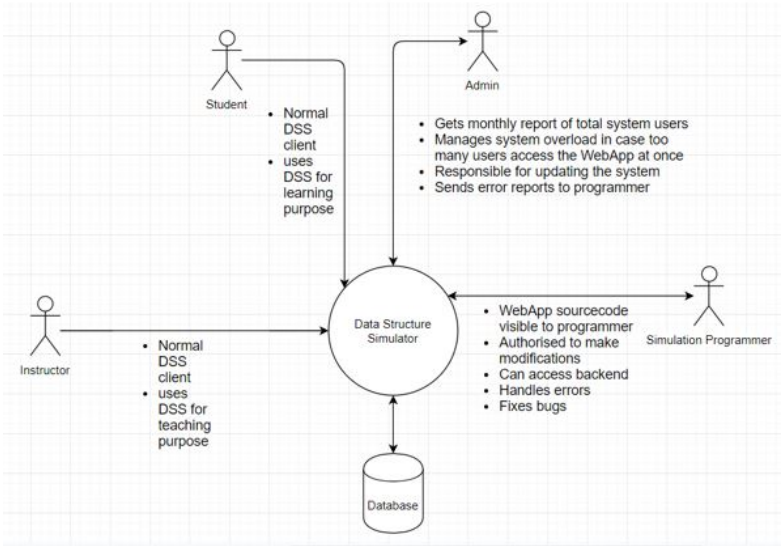


Figure 2.1: DIAGRAM

2.2 Product Functionality

Given below are the major functions that can be performed using DSS WebApp. Moreover, a Data Flow Diagram (DFD) for better understanding of the system is also given. The system will:

- Allow the users to add elements at the beginning or end of the linked list.
- Allow users to delete an element in linked lists.
- Allow users to delete a specific edge that is created.
- Allow users to insert an item onto stack.
- Users can delete an element from the top of the stack(TOS).
- Element at the top of the stack is shown to the user.
- Enter starting vertex: user has to enter a vertex present in the graph according to which searching will take place
- BFS: user can perform breadth-first search on the graph
- DFS: user can perform depth-search search on the graph
- A new graph can be selected to perform searching technique on.
- The user has an option to choose either directed or undirected graph.
- Next to the graphical representation of graph , a tabular view of parent and visited nodes is displayed which updates in real-time along with the animation.
- In addition, a log of each node visited and path of visit of displayed both graphically and in textual form, for better understanding.
- Animation can be paused/played by the user..
- 'Animation running' / 'animation completed' prompts at the corner of the screen.
- In the end, searching paths are highlighted with blue arrows for the user to view in graph.

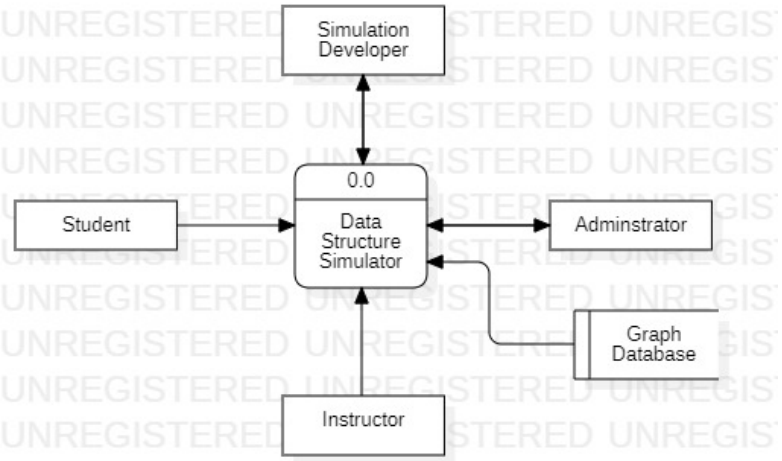


Figure 2.2: DATA FLOW DIAGRAM LEVEL0

2.3 Users and Characteristics

2.3.1 System Admin

The Administrator is the person or people who will have any and all the privileges of all other user types. They will be able to impersonate other users within the system. They will also have the authority to view and edit the WebApp. Also, He would be responsible for updating the system on a timely basis, manage any overloading on the system and report any bugs/errors to the simulation programmer.

2.3.2 Instructor

The Instructor is the person who would demonstrate the use of the simulator to his/her students.

2.3.3 Student

The students are the main users of the system as they would make a majority of the clientele. They would use the system most frequently as compared to all other users.

2.4 Operating environment

DSS is not dependent on geographical areas i.e. it would be able to function anywhere. The system shall operate in newest versions of all web browsers that work on windows operating systems (Java script). There should be no constraint on users being able to access the system at a given time. (24/7 availability) Continuous service is preferred, minor service interruptions can be tolerated.

2.5 Design and Implementation Constraints

2.5.1 Hardware Constraint

Hardware limitation of this software product can be unavailability of laptop or pc. If these devices are not available, then the end-user can't use the software to conduct a demonstration. Also, DSS would not work on mobile based browsers.

2.5.2 Browser Constraint

Installation of a web browser is necessary for the working of DSS. If a browser isn't installed, then a user must install it in his/her laptop. Internet Browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge etc.

2.5.3 JavaScript Constraint

Newest versions of the browser should be available which supports JavaScript

2.5.4 Internet Constraint

DSS requires the availability of Internet access at all time. Without connection to the Internet, the system would be futile.

2.5.5 Input Device Constraint

A mouse and keyboard are necessary to interact with the application. Touch screen mode is not available.

2.6 User Documentation

To run DSS, complex working knowledge of computers is not required, since it is targeted towards students who are just beginning to learn about computer-related concepts. The GUI is very user-friendly and can be understood easily by a novice learner. However, a basic understanding of data structures like stack, graph and linked lists is required for the user to properly grasp the concept and purpose of the simulator. Below is a link to a user manual online. https://pdfserv.maximintegrated.com/en/an/UG5861.pdf

2.7 Assumptions and Dependencies

2.7.1 Assumptions

- A major assumption is that the users of the data structure simulator are computer science/computer engineering/IT students/teachers who have a conceptual understanding of what data structures are and how they work.
- A web browser must be installed on the laptop or PC.
- If DSS is being used on a laptop then it must be charged.
- If DSS is being used on a PC then a stable power must be supplied.

2.7.2 Dependencies

This software is highly dependent on the availability of computer and internet connection where the website is going to be accessed.

Chapter 3

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The customer interface will contain three screens. All three screen will have a consistent layout.

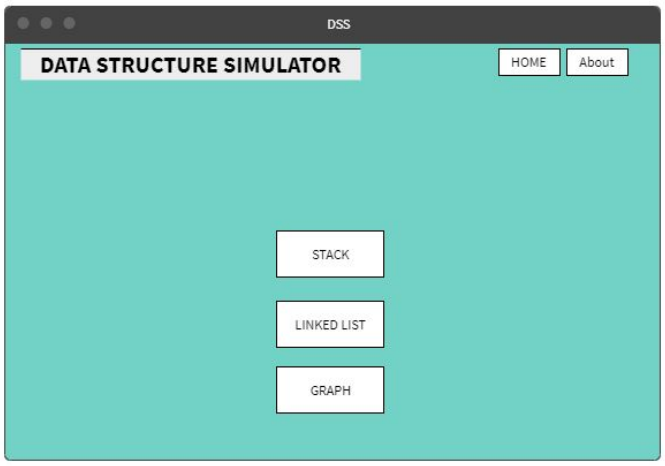
1. **Home Screen** This is the main navigation page. In this screen, system shows a list of options (UI Elements) of data structures. Each button will redirect the user to a new link to pages of Linked list, Stack and Graph.

2. **Linked list**Simulation of Linked list will be on this page, including insertion and deletion of items.

3. **Stack** Simulation of Stack will be on this page, including insertion and deletion of elements.

4. **Graph**Simulation of Graphs will be on this page, including option for BFS and DFS algorithms.

There will be an animation speed bar which can be used by user to adjust the speed of animation, for example item can be popped out of stack in 1 second, 5 seconds, 10 seconds and so forth. The size, height and width, of the simulation canvas can be altered. However, the height cannot be less than 200 and width cannot be less than 1000.



3.1.2 Hardware Interfaces

There are no hardware interfaces in our system as DSS is a Simulator and doesn't require the usage of any hardware component.

3.1.3 Software Interfaces

- For Database services system shall use xampp for MySql.
- System will run on Windows operating system such as Windows 10.
- The graph will be exported through the database into the Graph Animation page.

3.1.4 Communications Interfaces

DSS is an online simulator so the user needs a web browser and thus internet access is required for communication over the network and communication between the servers.

3.2 Functional requirements

3.2.1 Main Page

User Requirement

1. The user will access the front page of DSS.

System Requirement

1. The user will click on any of the three options given on main page.
    - option 1 is stack.
    - option 2 is linked list.
    - option 3 is graph.
2. After clicking on any of the aforementioned buttons, the user will be redirected to a new page as per the selection.

3.2.2 Animation speed bar

User Requirement

1. The user can toggle the animation speed bar.

System Requirement

1. A toggle bar is shown on the screen which changes the speed of the animation.
2. The speed bar slows down the animation if the user moves it to the left of the bar depending on the position up to 20 to 30 milliseconds.
3. The speed bar fastens the animation if the user moves it to the right of the bar depending on the position up to 2 to 3 milliseconds.

3.2.3 Change Canvas Size

User Requirement

1. The user can change the size of the canvas.

System Requirement

1. Two textboxes are displayed on the screen.
    - textbox 1 is height.
    - textbox 2 is width.
2. The user can interchange the values of height and width as per user requirement.
3. The height cannot be less than 200 and width not less than 1000.
4. Change Canvas button is given to alter the canvas.

3.2.4 Stack

Following are the functional requirements of the stack interface.

Function	Push
Description	Inserts an element onto stack.
Input	Any number, character or string.
Source of Input	An immediate value.
Output	Item is seen in the stack.
Action	Element pushed in stack can be observed.
Pre-condition	TOS is incremented.
Post-condition	Element is seen at TOS.
Exception	

Function	Pop
Description	Inserts an element onto stack.
Input	-
Source of Input	Pop button on the GUI.
Output	Item removed from the stack
Action	Element deleted from TOS can be observed.
Pre-condition	Stack contains elements i.e. it is not empty.
Post-condition	TOS is decremented.
Exception	TOS=BOS i.e. stack is empty.

Function	Peek
Description	Shows the element at the top of the stack.
Input	-
Source of Input	-
Output	Displays the element.
Action	An element
Pre-condition	Stack is not empty.
Post-condition	-
Exception	-

Function	Clear
Description	Erases the stack from the screen.
Input	-
Source of Input	-
Output	A blank canvas is shown on the screen.
Action	An element
Pre-condition	Stack is present.
Post-condition	-
Exception	

3.2.5 Graph

Following are the functional requirements of the graph interface.

Function:	Enter starting vertex
Description	A graph with multiple nodes will be displayed to the user. User would enter a specific vertex with respect to which searching will be performed.
Inputs	Vertex in graph
Source of input	Keyboard
Outputs	Program makes sure the entered vertex is present in the graph
Action	Entering a number in the input box
Precondition	User is on the graphs page
Postcondition	User clicks 'Run'
Exception Path	If user enters a vertex which not present in the graph then nothing happens.

2.

Function	Run BFS
Description	After entering a vertex the user clicks on 'Run BFS'. The program performs a breadth-first search on the graph which in an exhaustive process and explains each phase of the searching technique step by step. BFS is a traversing algorithm and the program starts traversing from a selected node (source or starting node) and traverses the graph layer wise thus exploring the neighbor nodes (nodes which are directly connected to source node). It then moves towards the next-level neighbor nodes.
Inputs	Mouse click
Source of inputs	Mouse
Outputs	BFS search performed
Action	Clicking on Run BFS button
Precondition	A vertex/node must be entered in the input box
Postcondition	User views BFS searching technique
Exception	If internet connection fails then searching halts.

Function	Run DFS
Description	The user also has an option to perform DFS on the graph. After entering a vertex the user clicks on 'Run DFS'. The program performs a depth-first search on the graph. The DFS algorithm is a recursive algorithm that uses the idea of backtracking. It involves exhaustive searches of all the nodes by going ahead, if possible, else by backtracking. Here, the word backtrack means that when you are moving forward and there are no more nodes along the current path, you move backwards on the same path to find nodes to traverse. All the nodes will be visited on the current path till all the unvisited nodes have been traversed after which the next path will be selected.
Inputs	Mouse click
Source of inputs	Mouse
Outputs	DFS search performed
Action	Clicking on Run DFS button
Precondition	A vertex/node must be entered in the input box
Postcondition	User views DFS searching technique
Exception	If internet connection fails then searching halts.

4.

Function	Display New graph
Description	User can click on 'new graph' button to access another of the ten graphs stored in the database
Inputs	Mouse click
Source of inputs	Mouse
Outputs	New graph displayed
Action	Clicking on 'new graph' button
Precondition	User must be on the graphs webpage
Postcondition	Input box cleared and new graph can be seen

Function	Choose one of directed/undirected graph option
Description	The user selects one of two press-selected radio buttons with labels 'directed graph' and 'undirected graph' respectively. a **directed graph** contains an ordered pair of vertices whereas an **undirected graph** contains an unordered pair of vertices. There are no arrowed lines in an undirected graph.
Inputs	Mouse click
Source of inputs	Mouse
Outputs	Directed/undirected graph displayed according to selection
Action	Selecting one of two radio buttons
Precondition	User must be on graph webpage
Postcondition	One of the two must be selected. **The default setting is directed graph.**

6.

Function	Pausing and resuming animation
Description	There is a pause button at the bottom of the screen which the user can press to pause animation if searching is too fast to understand. This pauses the traversal and display freezes. The pause button then changes to 'play' which the user can press to resume traversal.
Inputs	Mouse click on 'pause'/ 'play'
Source of inputs	Mouse
Outputs	Searching paused/resumed
Action	Clicking on button
Precondition	Animation must be running(searching being performed)
Postcondition	Searching paused/resumed
Exception	If the 'pause'/ 'play' button is pressed without running the search then nothing happens

3.2.6 Linked List

Following are the functional requirements of the linked list interface.

Function	Enter node value in textbox
Description	User will enter the desired value of the node to be inserted or deleted in input textbox
Inputs	Char
Source of inputs	Keyboard
Outputs	Display the entered value in the input textbox
Action	User has only entered text in the input textbox
Precondition	User must be on the Linked List page
Postcondition	None
Exception	None

Function	Insert Node
Description	User will click the insert button to add the node to the linked list. The value of the node will be acquired from the input textbox
Inputs	Mouse click
Source of inputs	Mouse
Outputs	The node will be placed in its appropriate place in the Linked List.
Action	User can observe the position where the new node has been inserted in Linked list.
Precondition	User must have entered text in the input textbox
Postcondition	The new node will point to the tail of the linked list
Exception	If input textbox contains no value, no node will be inserted.

Function	Delete Node
Description	User will click delete button. The node which corresponds to the value in the input box will be deleted. The linked list's pointers will be updated accordingly.
Inputs	Mouse click
Source of inputs	Mouse
Outputs	The node will be deleted
Action	Absence of the deleted node can be observed.
Precondition	User must have entered text in the input textbox
Postcondition	The direction of pointers will be adjusted accordingly
Exception	If input textbox contains no value, no node will be deleted.

3.3 Behaviour Requirements

3.3.1 Use case View

Main Page Use case

The Student/Instructor accesses the DSS and a variety of options are displayed. This use case extends into three more use cases including stack, LL and Graphs.

Stack Use case

The user is redirected to stack animation page where he/she can push, pop or peek elements. The aforementioned operations are a further extension of the stack use case.

LL Use case

The user is redirected to Linked list animation page where he/she can insert or delete items. The aforementioned operations are a further extension of the LL use case.

Graph Use case

The user is redirected to graph animation page where he/she can simulate graphs using BFS or DFS. The aforementioned operations are a further extension of the graph use case.

Maintenance Use case

This use case is further extended to manage overload and update system that are tasks for the admin

System Report Use case

A timely report is given to both the developer and admin.

Provide Repair Use case

This is divided into handle error and remove bugs in case any error or bug arises in the simulator.

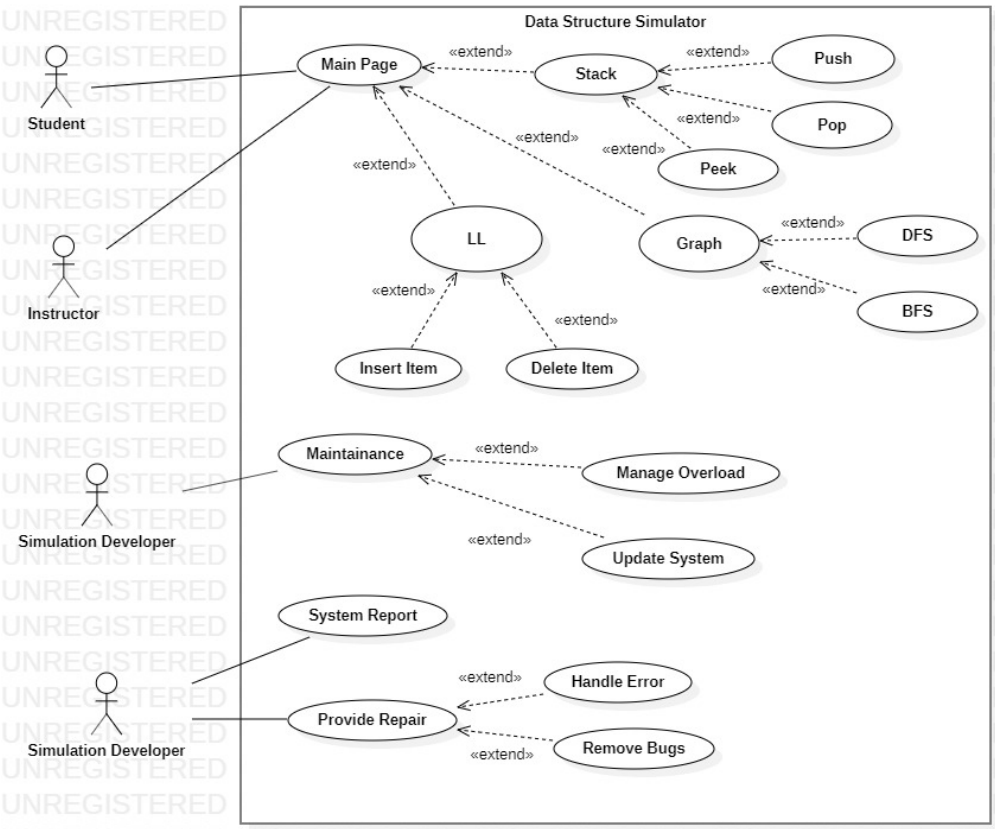


Figure 3.1: USE CASE DIAGRAM

Chapter 4

Other Non-Functional requirements

4.1 Performance Requirements

Since the DSS is an interactive tool, therefore the major performance requirement is low response time.

1. After clicking on any one of the data structures (stack, graph and linked list) on the main page, the structure view should not take more than 4 seconds to load.

2. When adding/deleting an item from stack/linked list, the updated view should be visible instantaneously.

3. When performing searches in the graph, the process should not take more than 10 seconds, although the speed can be adjusted in the tool since this feature is better understood at a slower speed so that the clients can grasp the searching technique properly.

4. In every action-response of the system, there are no immediate delays. In case of opening windows forms or sessions the delay is much below 2 seconds.

4.2 Safety and Security Requirements

Safety

There are no safety requirements for this tool other than the normal hazards of a computer system, like, using the tool for too long which might cause a

strain to the eyes.

Security

There aren't any security requirements for the DSS since it is an open application built for academic purposes and no personal data is being stored. Hence there is no need for a login id and password for every user and all clients can freely access its services. However, it is a closed source software therefore necessary actions need to be taken to ensure that the source code for the tool is secure from hackers since any change might hinder the working of the tool which can lead to a nuisance for the clients.

4.3 Software Quality Attribute

4.3.1 Adaptibility

There can be a change in the information stored in the database.

4.3.2 Availability

The system is up and running for most of the time and server is not down for more than a few minutes to avoid inconvenience of the clients.

4.3.3 Correctness

The techniques for the data structure operations must work accurately to build proper concepts of students.

4.3.4 Flexibility

If need arises in the future, software can be modified to change the requirements.

4.3.5 Interoperability

The data is transferred from the database to the web server and vice versa. This way data is transferred from one part of the system to another.

4.3.6 Maintainability

Software can be easily repaired if a fault occurs.

4.3.7 Portability

Software can be easily installed on devices and would run smoothly according to the requirement.

4.3.8 Reliability

No matter how many students access it, system must give the correct results.

4.3.9 Reuseability

Current version can be used in the future versions with more functionality added.

4.3.10 Robustness

Software must have checks to ensure that valid data items are entered.

4.3.11 Testability

All the requirements are fulfilled, response time is low, and all functions are working perfectly.

4.3.12 Useability

Interface of the software must be easy to use. It would not be complex since students have a view, so interface should be simple.

Index