

**A VISUALIZATION SYSTEM FOR INFORMATION SECURITY  
STANDARDS ON LARGE MULTIPLE DISPLAY WALLS**

ระบบแสดงผลข้อมูลสำหรับมาตรฐานความปลอดภัยด้านสารสนเทศบน  
จอแสดงผลขนาดใหญ่

**BY**

**MISS BOONRAK KAEWNOPPARAT 5988124**  
**MR. NASORN NIAMPRADIT 5988159**  
**MR. CHAYAPON PUAKALONG 5988185**

**ADVISOR**

**DR. ASSADARAT KHURAD**

**CO-ADVISOR**

**DR. WURHICHART SAWANGPHOL**

**A Senior Project Submitted in Partial Fulfillment of  
the Requirement for**

**THE DEGREE OF BACHELOR OF SCIENCE  
(INFORMATION AND COMMUNICATION TECHNOLOGY)**

**Faculty of Information and Communication Technology  
Mahidol University  
2019**

## **ACKNOWLEDGEMENTS**

We would like to express our deepest gratitude for all the support that we have received since we started this project. Firstly, we would like to express our appreciation to our senior project advisor and co-advisor, Dr. Assadarat Khurat, and Dr. Wudhichart Sawangphol, who have helped us since our internship in Japan, chose the topic of our project, and provided much great advice for improving our project until the end of the senior project. Secondly, we would like to express thanks to our parents who have supported us since the three of us started to study at the Faculty of ICT. Also, they have provided much great encouragement to us so that we could surpass tough times and they always stayed at our side. Finally, we would like to thank all the people who supported us in our project by providing comments for us to improve our project, and by participating in the test that we held which are priceless experiences for us to improve our senior project and ourselves in the real world.

Mrs Boonrak Kaewnopparat

Mr. Nasorn Niampradit

Mr. Chayapon Puakalong

## A VISUALIZATION SYSTEM FOR INFORMATION SECURITY STANDARDS ON LARGE MULTIPLE DISPLAY WALLS

MISS BOONRAK KAEWNOPPARAT      5988124 ITCS/B  
MR. NASORN NIAMPRADIT            5988159 ITCS/B  
MR. CHAYAPON PUAKALONG            5988185 ITCS/B

B.Sc.(INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: DR. ASSADARAT KHURAT

### ABSTRACT

In the midyear of 2019, it is reported that there were over 3,800 data breaches exposing over 4.1 billion records. This leads to being concerned about information security management systems (ISMS) inside organizations to protect their assets based on the international security standards, namely ISO/IEC 27001:2013 and ISO/IEC 27002:2013 which provide guidelines to organizations for implementing an ISMS. However, there is a need to form a collaboration of information security experts to build an ISMS for an organization while getting the same idea of implementation processes. Consequently, we decided to develop an interactive visualization being able to represent the overview of ISO information security standards on a large multiple display wall to help the IT staff and the experts with illustrating the entire range of issues of information security standards and developing them in coherent manner.

The proposed application is the visualization that works on two platforms including SAGE2 and a web application by using data from Natural Language Processing which is a process using Python, sklearn and spaCy library to find the relationship between ISO/IEC 27001:2013 and ISO/IEC 27002:2013. The visualization applications from both platforms are developed by using JavaScript, Node.js, and D3.js libraries.

In conclusion, the visualization application provides information for users about the ISO security standards and methods to implement ISMS and enhance the collaboration using large multiple display walls. In the future, the visualization could be further developed to be compatible with other security standards as well.

KEYWORDS: ISO SECURITY STANDARD, SAGE2 APPLICATION, VISUALIZATION, NATURAL LANGUAGE PROCESSING

## ระบบแสดงผลข้อมูลสำหรับมาตรฐานความปลอดภัยด้านสารสนเทศบนจอแสดงผลขนาดใหญ่

นางสาว บุญรักย์ แก้วนพรัตน์ 5988124 ITCS/B

นาย นสรณ์ เนียมประดิษฐ์ 5988159 ITCS/B

นาย ชยพล พวากาลง 5988185 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: ดร. อัมภารัตน์ คูรัตน์

บทคัดย่อ

ในช่วงปีคริสต์ศักราช 2019 นี้ ได้มีเหตุการณ์มากกว่า 3,800 เหตุการณ์ที่เกี่ยวข้องกับการจารกรรมทางข้อมูลสารสนเทศจากมากกว่า 4.1 พันล้านบันทึก ซึ่งสอดคล้องกับ คำเตือนที่ได้นำไปสู่การตั้ง คำถามเกี่ยวกับระบบการจัดการความปลอดภัยของข้อมูล หรือ ISMS ขององค์กรต่างๆ ในการรับมือ กับความเสี่ยงและปักป้องสินทรัพย์ขององค์กร ว่า ได้มีมาตรฐานเทียบเท่ากับระดับมาตรฐานความปลอดภัยสารสนเทศของ ISO/IEC 27001:2013 และ ISO/IEC 27002:2013 ซึ่งช่วยเป็นข้อแนะนำ ในการวางแผน ISMS สำหรับองค์กรต่างๆ ได้อย่างไร ตาม ที่จะ วางแผน ISMS ได้นั้น จำเป็นที่จะต้องมีการร่วมมือกันระหว่างผู้เชี่ยวชาญทางด้านความปลอดภัยสารสนเทศ และแนวทางในการจัดการ กีติกรรม ที่จะไปในทิศทางเดียวกัน ดังนั้น ทางเรา ได้ตัดสินใจที่จะพัฒนาโปรแกรม แสดงผลข้อมูล ที่สามารถใช้หน้าจอแสดงผลขนาดใหญ่ในจอแสดงข้อมูลด้านความปลอดภัยสารสนเทศอย่างมีประสิทธิภาพ

โดย โปรแกรมแสดงผลข้อมูลนี้ สามารถแบ่งออก ได้เป็น 2 ส่วน ส่วนแรกคือ แอปพลิเคชันแสดงผลข้อมูลบนแพลตฟอร์ม SAGE2 และ เว็บแอปพลิเคชัน ซึ่งใช้ข้อมูลจากการประมวลภาษาธรรมชาติหรือ NLP ซึ่งใช้ภาษา Python, spaCy และ sklearn ในการพัฒนา เพื่อที่จะหาความสัมพันธ์ระหว่างมาตรฐานความปลอดภัยของ ISO/IEC 27001:2013 และ ISO/IEC 27002:2013 โดยแอพพลิเคชันแสดงผลข้อมูล ทั้งสองแพลตฟอร์ม พัฒนาโดยใช้ภาษา JavaScript, Node.js, D3.js และ jQuery

- โดยส่วน โปรแกรมแสดงผลข้อมูลสามารถแสดงข้อมูลที่เป็นประโยชน์เกี่ยวกับมาตรฐานความปลอดภัยสารสนเทศ ISO และการวางแผน ISMS รวมไปถึงส่งเสริมการทำงานร่วมกันบนจอแสดงผลขนาดใหญ่ ได้อย่างมีประสิทธิภาพ โดยในภายภาคหน้านี้ เราเลือกเห็นว่าระบบแสดงผลข้อมูลนี้สามารถพัฒนาต่อยอดเพื่อรองรับมาตรฐานความปลอดภัยอื่นๆ

## CONTENTS

	Page
ACKNOWLEDGMENTS	ii
ABSTRACT	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF LISTINGS	xii
<b>1 INTRODUCTION .....</b>	<b>1</b>
1.1 PROBLEM STATEMENT .....	1
1.2 MOTIVATION .....	2
1.3 OBJECTIVE OF THE PROJECT .....	2
1.4 EXPECTED BENEFITS .....	3
1.5 ORGANIZATION OF THE DOCUMENT .....	3
<b>2 BACKGROUND .....</b>	<b>4</b>
2.1 BACKGROUND KNOWLEDGE .....	4
2.1.1 ISO/IEC 27001:2013 AND ISO/IEC 27002:2013 .....	4
2.1.2 NATURAL LANGUAGE PROCESSING .....	4
2.1.3 TONIC MODELING .....	9
2.1.4 VISUALIZATION TECHNIQUE .....	10
2.1.5 MULTIPLE DISPLAY WALLS .....	12
2.2 LITERATURE REVIEW .....	14
2.2.1 CRUMBS: A CYBER SECURITY FRAMEWORK BROWSER .....	15
2.2.2 VISUALIZING CYBER SECURITY: USABLE WORKSPACES .....	17
2.2.3 SAGE2: A NEW APPROACH FOR DATA INTENSIVE COLLABORATION USING SCALABLE RESOLUTION SHARED DISPLAYS .....	18
2.2.4 DOITREES REVISITED: SCALABLE, SPACE-CONSTRAINED VISUALIZATION OF HIERARCHICAL DATA .....	19

<b>3 ANALYSIS AND DESIGN .....</b>	22
3.1 SYSTEM ARCHITECTURE OVERVIEW .....	22
3.2 SYSTEM STRUCTURE CHART .....	22
3.3 DATA FLOW DIAGRAM .....	24
3.3.1 DATA FLOW DIAGRAM LEVEL 0 .....	24
3.3.2 DATA FLOW DIAGRAM LEVEL 1 .....	24
3.3.3 DATA FLOW DIAGRAM LEVEL 2 .....	27
3.4 VISUALIZATION .....	28
3.4.1 SUNBURST VISUALIZATION .....	28
3.4.2 ZOOMABLE CIRCLE PACKING VISUALIZATION .....	28
<b>4 IMPLEMENTATION .....</b>	31
4.1 SYSTEM ENVIRONMENT .....	31
4.2 IMPLEMENTATION GUIDE AND TECHNIQUES .....	31
4.2.1 PROCESSING DATA .....	31
4.2.2 SAGE2 APPLICATION.....	39
4.2.3 WEB APPLICATION.....	56
<b>5 TESTING AND EVALUATION .....</b>	59
5.1 UNIT TESTS.....	59
5.2 USABILITY TEST PROCEDURE.....	59
5.3 EVALUATION .....	62
<b>6 IMPROVEMENT .....</b>	67
6.1 SUNBURST VISUALIZATION .....	67
6.2 ZOOMABLE CIRCLE PACKING VISUALIZATION .....	72
6.3 NATURAL LANGUAGE PROCESSING .....	79
<b>7 CONCLUSIONS .....</b>	82
APPENDIX A .....	84
APPENDIX B .....	98
REFERENCES .....	162
BIOGRAPHIES .....	165

**LIST OF TABLES**

	Page
Table 2.1: The content of Clauses in ISO/IEC 27001:2013 .....	5
Table 2.2: The content of Annexes in ISO/IEC 27002 (1).....	6
Table 2.3: The content of Annexes in ISO/IEC 27002:2013 (2)	7

## LIST OF FIGURES

	Page
Figure 2.1: The stages of the NLP process. ....	8
Figure 2.2: Example of a graph structure. ....	11
Figure 2.3: Example of a tree structure. ....	12
Figure 2.4: Example of a histogram. ....	12
Figure 2.5: Example of the Nureva Wall system. ....	13
Figure 2.6: The Example Prism Display platform. ....	14
Figure 2.7: The Example SAGE2 software. ....	14
Figure 2.8: Contextualization overview of NIST cybersecurity framework layer and Priority level, and Mandatory items as columns. ....	15
Figure 2.9: Current profile detail. Mouse-hovering on controls and subcategories provides full details about inspected items. ....	16
Figure 2.10: CRUMBS system configured for Gap analysis. The bottommost thinner horizontal bar provides (colors) the information about the function the controls belong to. All the controls are partially ordered with respect to a suitable implementation strategy. ....	17
Figure 2.11: The security manager inspects a gap bar for a subcategory of the Protect function; the gap bar shows a not so big distance between the current and target statuses. ....	17
Figure 2.12: History trees, a mockup visual workspace showing history and allowing workflow traceability through the use of keyframes, resulting in the final report on the far right. ....	18
Figure 2.13: Example of using SAGE2 in the conference. ....	19
Figure 2.14: DOITree visualization of the Open Directory Project, containing over 600,000 nodes. ....	20
Figure 3.1: The system architecture of the ISO visualization program. ....	22
Figure 3.2: The structure chart of the ISO visualization program. ....	23
Figure 3.3: The level 0 data flow diagram of the ISO visualization system. ....	24

Figure 3.4: The example of structure from the create JSON process .....	26
Figure 3.5: The level 1 data flow diagram of the ISO visualization system.....	27
Figure 3.6: The level 2 data flow diagram of ISO visualization system.....	28
Figure 3.7: The overview of Sunburst visualization that represents the ISO/IEC 27001:2013 information. It can be seen that all the ISO information is represented in the tree structure with the root node as a center circle. .	29
Figure 3.8: The overview of the Zoomable Circle Packing visualization that represents the ISO/IEC 27001:2013 information. This visualization represents the ISO information with each big circle as a subtree for each topic in the document. It can make users easily see the overview of the ISO information structure. ....	30
Figure 4.1: The example for <i>mapClause</i> and <i>mapDetail</i> variables .....	35
Figure 4.2: The idf table result .....	37
Figure 4.3: The example of dictionary variable to JSON file.....	38
Figure 4.4: The example of the result JSON file.....	39
Figure 4.5: The search bar and other buttons in the Sunburst visualization SAGE2 application .....	39
Figure 4.6: The nodes in the Sunburst visualization SAGE2 application.....	42
Figure 4.7: New application that shows the information of Clause 5: Leadership .	43
Figure 4.8: The transition process in the Sunburst visualization when user clicked on a node. ....	44
Figure 4.9: The pop-up application that shows the information of Annex 12.5.1 and 12.6.1 and their references. ....	46
Figure 4.10. The nodes in the Zoomable Circle Packing visualization SAGE2 application.....	53
Figure 4.11: Clause 4, Context of the organization, highlighted with yellow stroke as it is matched with search query.....	53
Figure 4.12: The pop-up window that contains information about Annex 15.2.2 in ISO:IEC 27002:2013 and its references. ....	54
Figure 4.13: The transition in the Zoomable Circle Packing visualization when the user clicks on each node.....	56

Figure 4.14: The pop-up window that contains information of Annex 9.2.6 and 11.2.8 in the Sunburst visualization on the web application .....	57
Figure 5.1: The procedure in usability testing. ....	60
Figure 5.2: Average time - train question.....	63
Figure 5.3: Average time - test question .....	63
Figure 5.4: Average time - test question section 1: Manual document .....	64
Figure 5.5: Average time - test question section 2: Sunburst visualization.....	64
Figure 5.6: Average time - test question section 3: Zoomable circle packing visualization.....	65
Figure 5.7: The correctness from the test separated by tester type .....	65
Figure 5.8: Effectiveness of the visualization from tester's opinion .....	66
Figure 5.9: The user satisfaction for the visualization .....	66
Figure 6.1: The color version of the Sunburst visualization....	68
Figure 6.2: The greyscale version of the Sunburst visualization .....	69
Figure 6.3: The help button next to the blind color button in the Sunburst visualization .....	69
Figure 6.4: The user manual of the Sunburst visualization on the SAGE2 application.....	70
Figure 6.5: The pop-up window with new design and its related subclauses .....	71
Figure 6.6: The text of the next two layers will appear after zooming to a node ...	72
Figure 6.7: The text of the upper layer is overlapped with the text of the lower layer	72
Figure 6.8: The text is cut in order to fit within the circle size .....	73
Figure 6.9: Shifting the text of the upper layer to the top of the text of the lower layer .....	73
Figure 6.10: The text is separated into several lines .....	74
Figure 6.11: The visualization of text separation that does not zoom to any layer...	74
Figure 6.12: The text is moved to the top of the circles.....	75
Figure 6.13: The text is moved to the top of the circles after zooming one level....	75
Figure 6.14: Displaying the text of the next two layers .....	76
Figure 6.15: The size of the text of the below level is decreased to make it be more readable.....	76

Figure 6.16: The text of the clause is moved to the center of the circle.....	77
Figure 6.17: The text size of the subclause is increased when zooming .....	77
Figure 6.18: Add the related subclauses into popups.....	78
Figure 6.19: Add help button and manual to help users using the program.....	78
Figure 6.20: The idf table result .....	80
Figure A.1: Launcher.bat in the SAGE2 folder .....	84
Figure A.2: The SAGE2 control window .....	85
Figure A.3: The SAGE2 console .....	85
Figure A.4: When the user drops a Zip file in the SAGE2 console .....	86
Figure A.5: List of applications that have already been installed on SAGE2.....	86
Figure A.6: The example of structure from the create JSON process .....	87
Figure A.7: The XAMPP folder .....	87
Figure A.8: Move the application folder to htdocs in the XAMPP folder.....	88
Figure A.9: Launch the XAMPP application .....	88
Figure A.10: Start the Apache module.....	89
Figure A.11: The URL for the Zoomable circle packing visualization .....	89
Figure A.12: The console of the SAGE2 application .....	90
Figure A.13: SAGE2 screen.....	90
Figure A.14: Interact with the SAGE2 application using the SAGE2 pointer .....	91
Figure A.15: The main page of the Sunburst visualization .....	91
Figure A.16: The difference between each focus area of the Sunburst visualization ..	92
Figure A.17: The Sunburst visualization with the result from the search function ...	93
Figure A.18: Pop-up window that contains the information from the ISO/IEC 27002:2013 document .....	93
Figure A.19: The main page of the Zoomable Circle Packing visualization .....	94
Figure A.20: The difference between each focus area of the Zoomable Circle Packing visualization .....	95
Figure A.21: The Zoomable Circle Packing visualization with the result from the search function .....	96
Figure A.22: Pop-up window that contains the information from the ISO/IEC 27002:2013 document .....	97

## LIST OF LISTINGS

	Page
Listing 4.1: Code for text preprocessing of ISO/IEC 27001:2013 .....	33
Listing 4.2: Code for text preprocessing of ISO/IEC 27002:2013.....	34
Listing 4.3: Creating sparse matrix from <i>TfidfVectorizer</i> library.....	35
Listing 4.4: Computing similarity scores from every corpus .....	36
Listing 4.5: Filtering similarity scores with its average scores .....	36
Listing 4.6: Code for creating JSON file.....	38
Listing 4.7: Create element for the Sunburst Visaulization application .....	39
Listing 4.9: Transform data into a node for the visualization.....	40
Listing 4.8: Create dictionary from ISO/IEC 27001 :2013 and 27002 :2013 .....	41
Listing 4.10: Create node for visualization .....	42
Listing 4.11: Click function in each node for main application.....	43
Listing 4.12: Click function in each node for child application.....	44
Listing 4.13: Popup function for each node in the visualization (SAGE2 version) ..	46
Listing 4.14: Initialize pop-up application in SAGE2 .....	47
Listing 4.15: Functions in Zoomable visualization.....	47
Listing 4.16: Overall in createDict function .....	48
Listing 4.17: Create Annex Object.....	48
Listing 4.18: Create Clause object.....	49
Listing 4.19: The function for set highlight value in the clause object.....	50
Listing 4.20: Overall visualization function .....	51
Listing 4.21: Converting to d3.pack .....	51
Listing 4.22: Creating the main container.....	52
Listing 4.23: Creating nodes (circles) .....	54
Listing 4.24: A function to zoom in on a node.....	56
Listing 4.25: Read JSON file of the ISO/IEC 27001:2013 and ISO/IEC 27002:2013 document (Web application version) .....	57

Listing 4.26: Popup function for each node in the visualization (Web application version) .....	58
Listing 6.1: Change the salutation and lightness color to be greyscale .....	67

## CHAPTER 1

### INTRODUCTION

#### 1.1 Problem Statement

In the 21<sup>st</sup> century, data has grown exponentially [1], and insightful data is limited to use due to the amount of knowledge and effort required for understanding the enormous amount of data. Conversely, the number of threats that have malicious intent with data has increased as well. According to the 2019 MidYear QuickView Data Breach Report, 3,813 breaches are exposing over 4.1 billion records [2]. One of those breaches was the Verifications.io data breach case which impacted more than 900 million user accounts and their private information was exposed on the Internet. This is due to the misconfiguration of their database. It is a significant issue for every organization to make use of data while keeping their information assets secured. The security of information is a crucial factor for most organizations. However, without knowledge and expertise in the information security field, it is impossible for them to implement, manage, and maintain their Information Security Management System efficiently.

The Information Security Management System is a set of policies and procedures for systematically managing an organization's sensitive data [3]. These system objectives are to minimize risk and ensure business continuity. Therefore, there are the security standards that organizations can use as a guideline and reference to implement their Information Security Management System such as NIST SP 800-53 Rev. 4 [4], and CCITT 2019 [5]. Each standard was published by the security organization. However, one of the most well known standards in information technology is ISO/IEC 27001 and 27002 that was published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). ISO/IEC 27001 provides best practice recommendations on information security management for use by those responsible for initiating, implementing, or maintaining an Information Security Management System [6] while ISO/IEC 27002 specifies a management system that is intended to

bring information security under explicit management control [7]. However, there is a lot of information in the context of ISO/IEC 27001 and ISO/IEC 27002. It requires lots of time and cost to understand and follow these standard guidelines in order to implement and manage an Information Security Management System without knowledge in the security field.

## 1.2 Motivation

Since ISO/IEC27001 and ISO/IEC27002 are significant guidance to improve the Information Security Management System of an organization, visualization acts as an important role in making them easier to understand. There is some research related to visualization whether their purpose is to evaluate the visualization or create it [8]. Moreover, there is research that contextualizes the NIST security standard to make it more understandable for security managers and helps in making a strategy to develop the Information Security Management System [1]. However, this research does not provide any standard information. To the best of our knowledge, there is no research that actually visualizes the ISO/IEC information security standard for the purpose of helping organizations.

In response to these problems, we have developed a visualization of ISO/IEC 27001 and 27002 to make it easier for users with limited knowledge in information security to understand its context. Furthermore, we have applied Natural Language Processing (NLP) to help in extracting and finding the essence for each annex and clause of the ISO/IEC security standard and use that information to build a hierarchy of relationships within the ISO standard. With the help of visualization techniques and SAGE2 software, we can represent connections within the ISO security standard structure, and thereby provide more insight and be easier to follow. Managing an Information Security Management System would be easier for an organization through our visualization program.

## 1.3 Objective of the project

1. To provide an interactive visualization of ISO/IEC 27001 and 27002.
2. To use NLP and Topic modeling to find the existing relationship between clauses

and annexes in ISO/IEC 27001 and 27002.

3. To enhance the collaborative working environment with the use of multiple-display walls.

#### **1.4 Expected Benefits**

1. The Organization's security managers are able to understand the context of ISO/IEC 27001 and ISO/IEC 27002, and to implement and maintain the Information Security Management System.
2. Users receive knowledge and get a better understanding of the Information Security Management System and ISO security standard.

#### **1.5 Organization of the Document**

This document consists of seven chapters. The first chapter proposes the project foundation including problem statement, motivation, objective, and expected benefits. The second chapter talks about technical knowledge that is used in developing the project and consists of two sections including background knowledge and literature review. The third chapter is about the system architecture overview including the system structure chart, data flow diagram, and visualization. The fourth chapter shows the implementation process of the visualization containing two sections which are the system environment, and the implementation guide and techniques. The fifth chapter represents the testing section and evaluation of the visualization. This chapter contains three sections which are unit tests, the usability test procedure, and evaluation. The sixth chapter shows the process of improvement from the previous chapter according to the evaluation. Finally, the seventh chapter represents the conclusion of the project which provides the summary and future development of the project.

## CHAPTER 2

### BACKGROUND

#### 2.1 Background Knowledge

##### 2.1.1 ISO/IEC 27001:2013 and ISO/IEC 27002:2013

ISO/IEC 27001:2013 and ISO/IEC 27002:2013 are standards for information security management published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) [6, 7]. These standards provide requirements for organizations to develop an information security management system (ISMS) in order to protect their information assets. In ISO/IEC 27001:2013, it begins the introduction, scope, references and terms and definitions in Chapter 0 to Chapter 3; and it defines processes to develop an ISMS in Chapter 4 to Chapter 10, or called Clause 4 to Clause 10 in this project. The content in each clause of ISO/IEC 27001:2013 is described in Table 2.1. Moreover, in the Information Security Management Systems, there are annexes in ISO/IEC 27002:2013 that are related with the controls for clauses in ISO/IEC 27001:2013. The content of each annex is provided in Table 2.2 and Table 2.3.

##### 2.1.2 Natural Language Processing

Natural Language Processing (NLP) is a technique in the field of Artificial Intelligence (AI) to deal with interaction from human languages to computer understanding. There are several uses of this technique including email classification which can detect whether the emails are spam or not, or even fake news identification from the NLP group at MIT which can detect the credibility of a news source. The common approaches used in processing natural languages include 5 processes which are Text Preprocessing, Lexical Analysis, Syntactic Parsing, Semantic Analysis, and Natural Language Generation, respectively as shown in Figure 2.1 [9].

Table 2.1: The content of clauses in ISO/IEC 27001:2013

Clause	Description
0 - Introduction	<ul style="list-style-type: none"> <li>Importance of Information Security Management System (ISMS)</li> <li>Compatibility with other management system standards</li> </ul>
1 - Scope	<ul style="list-style-type: none"> <li>Scope of ISO/IEC 27001:2013</li> </ul>
2 - Normative References	<ul style="list-style-type: none"> <li>The following documents are normatively referenced</li> </ul>
3 - Terms and Definitions	<ul style="list-style-type: none"> <li>Terms and definitions are given in ISO/IEC 27001:2013</li> </ul>
4 - Context of the Organization	<ul style="list-style-type: none"> <li>Organization contexts: issues, ability to achieve goals</li> <li>Interested parties needs and expectations</li> </ul>
5 - Leadership	<ul style="list-style-type: none"> <li>To ensure that everything is prepared for doing ISMS</li> <li>There are policies to support ISMS</li> <li>Roles, Authorities, Responsibilities are assigned for ISMS</li> </ul>
6 - Planning	<ul style="list-style-type: none"> <li>Determine risks and opportunities</li> <li>Risk Assessment including Identify, Analysis, and Evaluate risks</li> <li>Risk Treatments - Select suitable options and controls to formulate the treatment plan.</li> </ul>
7 - Support	<ul style="list-style-type: none"> <li>Staff who are related to ISMS are competent to carry out the work</li> <li>Resources are provided for establishment, implementation, maintenance, and continual improvement of ISMS.</li> <li>Documented information is controlled.</li> </ul>
8 - Operation	<ul style="list-style-type: none"> <li>Performing actions follows the planning.</li> </ul>
9 - Performance Evaluation	<ul style="list-style-type: none"> <li>Processes' monitoring and measurement</li> <li>Analysis and evaluation of the results after monitoring and measurement</li> <li>Internal Audit including define the criteria and scope of the audit, and report results to management</li> <li>Review feedback, results, and opportunities after evaluation</li> </ul>
10 - Improvement	<ul style="list-style-type: none"> <li>Take actions to nonconformities</li> <li>Evaluate and review corrective actions</li> <li>Make changes to ISMS</li> <li>Continual improvement</li> </ul>

Table 2.2: The content of Annexes in ISO/IEC 27002 (1)

<b>A.5 Information security policies</b>	
A.5.1 Management direction for information security	Focusing on publishing and using policies for information security with relevant regulations and supported business requirements
<b>A.6 Organization of information security</b>	
A.6.1 Interval organization	Management framework for control operation of information security within the organization
A.6.2 Mobile devices and teleworking	Use of teleworking and mobile devices are secure.
<b>A.7 Human resource security</b>	
A.7.1 Prior to employment	Employees and contractors understand their responsibilities and roles.
A.7.2 During employment	Employees and contractors are aware of information security responsibilities.
A.7.3 Termination and change of employment	The organization is protected after employment is changed or terminated.
<b>A.8 Asset management</b>	
A.8.1 Responsibility for assets	Assets identification and protection
A.8.2 Information classification	The level of information depends on its importance to the organization
A.8.3 Media handling	The information that is stored on media is protected from unauthorized changing.
<b>A.9 Access control</b>	
A.9.1 Business requirements of access control	Information access is limited.
A.9.2 User access management	The systems and services must be accessed from authorized users only.
A.9.3 User responsibilities	Users must protect their authentication information.
A.9.4 System and application access control	The systems and applications are protected from unauthorized access.
<b>A.10 Cryptography</b>	
A.10.1 Cryptographic controls	Cryptography should protect the confidentiality, authenticity, and integrity effectively.
<b>A.11 Physical and environmental security</b>	
A.11.1 Secure areas	Information and information processing facilities are protected from unauthorized physical access.
A.11.2 Equipment	Assets are protected from loss, damage, theft or compromise and the organization's operations interruption.

Table 2.3: The content of Annexes in ISO/IEC 27002:2013 (2)

A.12 Operations security	
A. 12.1 Operational procedures and responsibilities	Information processing facilities operate correctly and securely.
A.12.2 Protection from malware	Information processing facilities are protected against malware.
A.12.3 Backup	Data is backed up to prevent loss.
A.12.4 Logging and Monitoring	Events are recorded to generate evidence.
A.12.5 Control of operational software	Operational systems must have integrity.
A.12.6 Technical vulnerability management	To protect from the exploitation of technical vulnerabilities
A. 12.7 Information systems audit considerations	Impact of audit activities must be minimized on operational systems
A.13 Communications security	
A. 13.1 Network security management	Information in networks is protected and supports information processing facilities.
A.13.2 Information transfer	Information transfer must be secure
A.14 System acquisition, development, and maintenance	
A.14.1 Security requirements of information systems	Information security is a part of the information systems across the entire lifecycle.
A. 14.2 Security in development and support processes	Information security is designed and implemented within the development life cycle of information systems.
A.14.3 Test data	Test data should be protected.
A.15 Supplier relationships	
A. 15.1 Information security in supplier relationships	Organizations' assets that can be accessed from suppliers should be protected.
A.15.2 Supplier service delivery management	Maintenance of suitable information security level and service delivery is in line with supplier agreements.
A.16 Information security incident management	
A.16.1 Management of information security incidents and improvements	The management of information security incidents is consistent and effective.
A.17 Information security aspects of business continuity management	
A. 17.1 Information security continuity	The organization's business continuity management systems should be embedded with information security continuity.
A.17.2 Redundancies	Information processing facilities should be available.
A.18 Compliance	
A.18.1 Compliance with legal and contractual requirements	Breach of legal, regulatory, security requirements avoidance
A. 18.2 Information security reviews	Information security is implemented and operated to follow the policies and procedures.

## Text Preprocessing

Text Preprocessing is a process to convert a raw text into a well-defined sequence of words for analysis in NLP systems. To be specific, Text Preprocessing is split into 2 parts. The first part is Document triage which converts a file into a well-defined document. The second part is Text segmentation which is a process to segment a text into words and sentences. The Text segmentation is divided into 2 parts: word segmentation and sentence segmentation. The word segmentation process is to get the raw texts and uses text normalization techniques to make those raw texts become their normal forms. Another part of text segmentation is sentence segmentation. This process is to split sentences using sentence boundaries including punctuation in order to find contexts of each sentence.

## Lexical Analysis

Lexical Analysis is a mechanism to map the string to their lemma which is contained in a lemma dictionary.

## Syntactic Parsing

Syntactic parsing is a method to perform syntactic analysis of a sentence. In other words, it is a method to put together words to form components of sentences and to put together these components to form sentences following rules or ‘syntax’.

## Semantic Analysis

Semantic analysis refers to analyzing the meanings of words or finding contexts from whole sentences.

## Natural Language Generation

Natural Language Generation is an approach to defragment and complete sentences of input texts using unsupervised learning models. In this research, we will im-

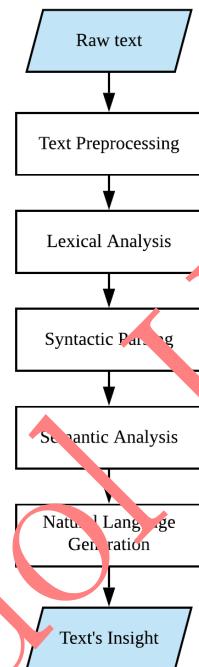


Figure 2.1: The stages of the NLP process.

plement 3 stages of the Classical approach of NLP which are the Text Preprocessing stage to segment texts into words for dealing with the other stages, the Lexical Analysis stage to normalize the segmented words into its normal form using lemma dictionaries, and the Semantic Analysis stage to analyze the meaning of the texts using word frequencies together with modeling topic to get its abstract topic. The reason why we skip the stages of Syntactic Parsing and Natural Language Generation (NLG) is that we do not need to put words together to form a sentence by using the Syntactic Parsing stage or to correct the sentence by using the NLG stage since we will use the actual text content to present in our visualization after we get its abstract topics to perform a relationship hierarchy for the visualization.

### 2.1.3 Topic Modeling

Topic modeling is a technique used in Natural Language Processing (NLP) as the Semantic Analysis stage and uses Text Mining to extract the meaning of the given contents to define an abstract topic of the contents. The most common approaches used for Topic Modeling include Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), and Correlated Topic Models (CTM).

#### **Latent Semantic Analysis (LSA)**

Latent Semantic Analysis (LSA) is a method to compute similarities between vector representations of words and documents in order to find the hidden topics of the whole contents. The LSA approach is to weight the related words which are represented as keys of the document with the vectors using Singular Value Decomposition (SVD) to reconfigure vector space from the most to the least significant in LSA for computing the similarities of texts [10]. However, with the process to find the semantic topics by the probabilities computed from the similarities of word vectors, it will be hard to measure the number of topics related to the contents. [11, 12].

#### **Latent Dirichlet Allocation (LDA)**

Latent Dirichlet Allocation (LDA) is a generative model based on the Bayesian model to define the probabilistic distribution of mixture models of topics, and with the topic probabilities, it will scope down the area of the abstract topics of the given doc-

ument to extract the meaning of the context. LDA concepts include finding words distribution in a topic, finding each topic proportion of a document and finding the related topics to the words in a document; thus, with these concepts computed with the Dirichlet distribution, it will give the most related topics to the input document as an output [13]. However, LDA still has a limitation in that it is unable to find the correlation among topics since the components of the proportions vector sampled by Dirichlet distribution are almost separated [11, 12].

### **Correlated Topic Model (CTM)**

Correlated Topic Model is a topic model proposed with the logistic normal distribution that allows a covariance structure among text components to model the correlation among topics. CTM uses the advantages of a hierarchical model of logistic normal to define the associations of the latent composition of topics with each document. CTM uses the generative process of LDA, but the topic proportions will be sampled from a logistic normal instead of a Dirichlet distribution. As a result, with CTM, there is less uncertainty about the remaining words than with LDA since the CTM uses the correlation of the composition to predict where the remaining words should rely on. The LDA is unable to do this. However, it requires lots of calculations for the logistic normal distribution in the CTM, and in the CTM, it can lead to handling a number of general words inside a topic by predicting the remaining words using the correlations [11, 12, 14].

#### **2.1.4 Visualization Technique**

The visualization process is a process of transformation of content into knowledge by creating an appropriate presentation. The visualization normally begins its processes with raw data and completes the processes with the knowledge obtained by humans. There are five processes in data transformation including filtering, mapping, layout, layer fusion, and rendering, and one interaction stage which is the visualization experience. There are many visualization techniques including Graph, Tree, and Histogram, etc. [15]. The Graph visualization technique is a way of representing structural information as diagrams of graphs data structure. A graph is a structure of objects in which some of them are related. The objects are called nodes, and their relationships are

called edges. An example of a graph can be seen in Figure 2.2. The Tree visualization technique is similar to the Tree data structure which is one type of graph. It has a root node to be the first node of the tree. Then this root node branches to the other nodes and so on. Therefore, the node that branches to other nodes is called the parent node while the nodes that are branches of the parent node are called child node. The node that has no child is called a leaf node. An example of a tree structure can be seen in Figure 2.3. Finally, a Histogram is a plot that visualizes a set of continuous data. It allows the inspection to see the underlying distribution, outliers, and skewness, etc. An example of a histogram can be seen in Figure 2.4. The visualization technique that we choose to apply is the Tree visualization technique. The reason that we do not choose a Histogram is that a Histogram does not suit the ISO security standard which is not continuous data. Moreover, a Graph implementation requires more expertise than implementing a Tree. The Tree visualization technique is appropriate with the information that is full of text and has a hierarchical relationship [16].

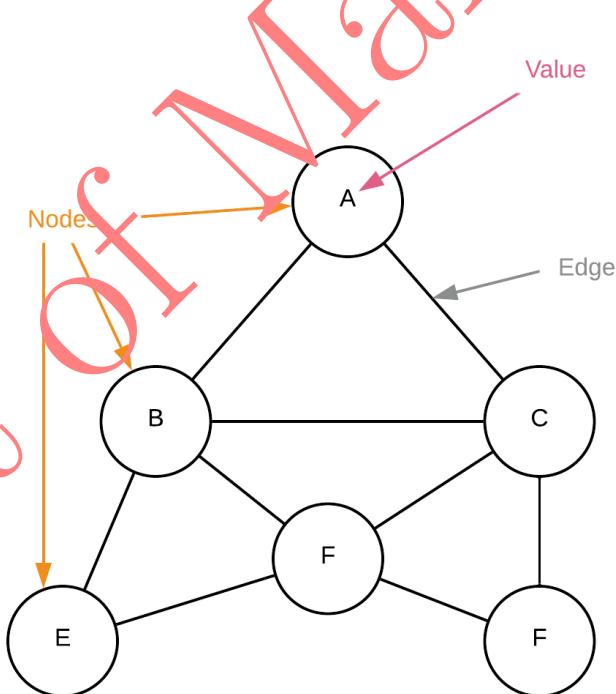


Figure 2.2: Example of a graph structure.

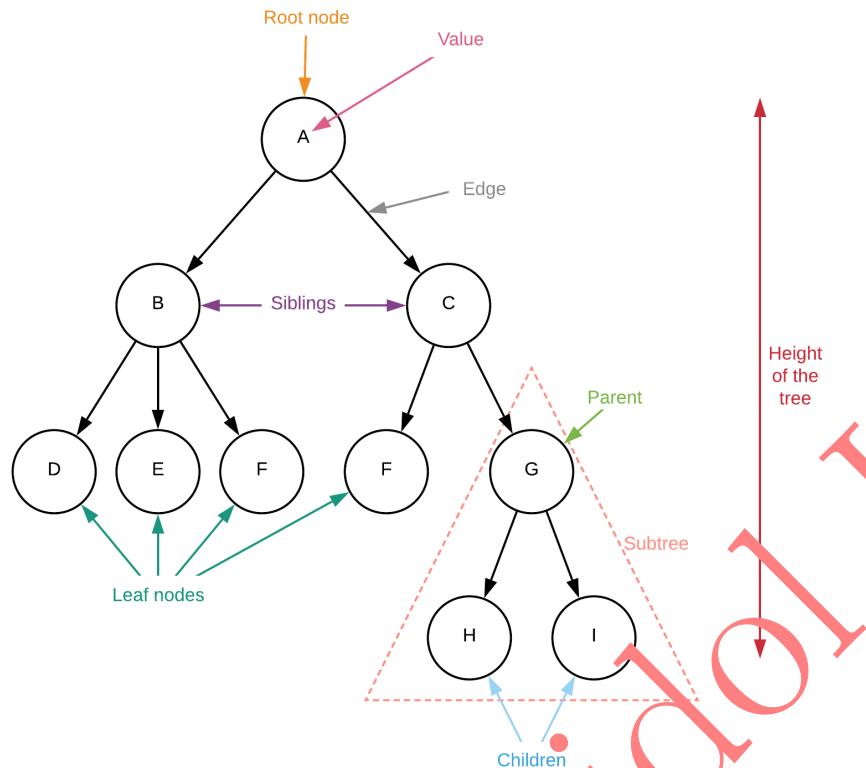


Figure 2.3: Example of a tree structure.

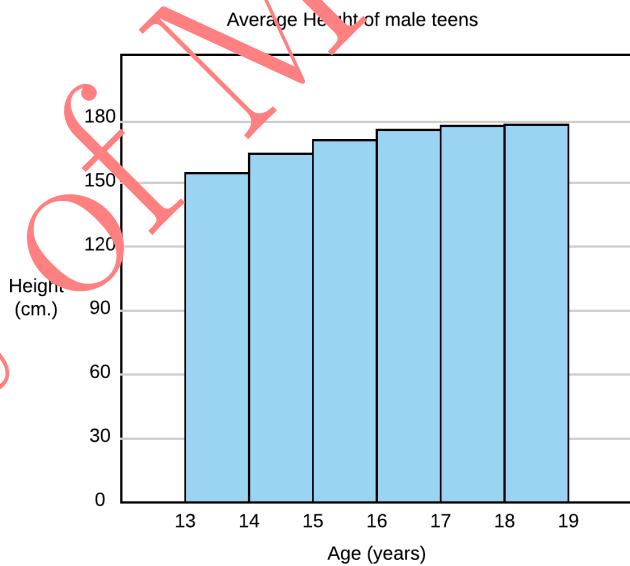


Figure 2.4: Example of a histogram

### 2.1.5 Multiple Display Walls

The multiple-display wall is a special monitor that consists of multiple computer monitors, projectors, or television sets tiled together contiguously or overlapped in order to form one large screen. The combination of multiple displays means that the resolution of each display would be combined with other displays and result in a higher resolution of the screen. The multiple-display wall can be used to share and display results in a collaborated working environment, and aid in representing large complex datasets. The multiple-display wall usually plays a role in collaborated working or a meeting because of the number of participants, and it is easier to manage visual information across the meeting in a highly effective manner. There are many platforms for a display wall that enhance the collaborated working including Nureva Wall, Prism Display, and SAGE2. Nureva Wall, as shown in Figure 2.5, is a visual collaboration system that enhances collaborative working and operates with a variety of applications [17]. On the other hand, Prism Display in Figure 2.6 is a visual collaboration software platform that enhances the workspace with its interactive displays and video walls [18]. Finally, SAGE2 software is a middleware for collaborated working with Scalable Resolution Shared Display as shown in Figure A.2 which means SAGE2 can be accessed with any resolution screen [19]. We choose to develop on the SAGE2 platform because SAGE2 supports many types of applications including custom applications. It also works with many resolutions as well, and SAGE2 is a free platform.

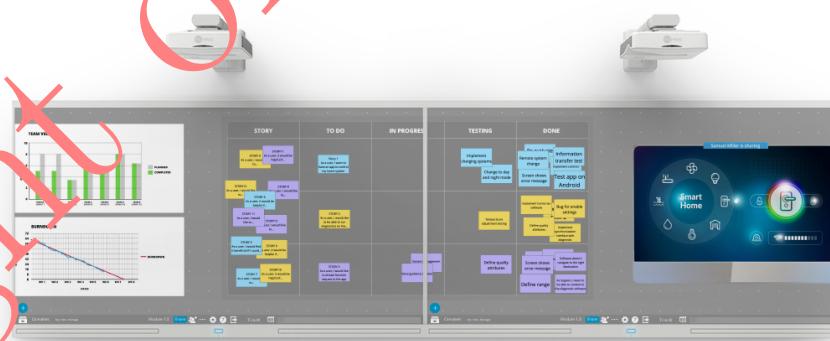


Figure 2.5: Example of the Nureva Wall system



Figure 2.6: The Example Prysm Display platform

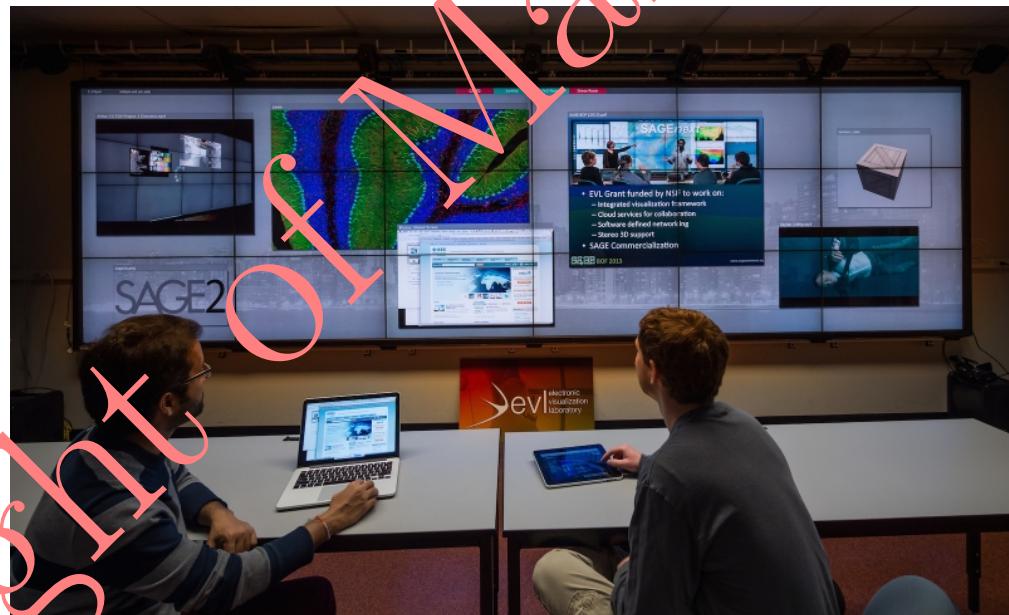


Figure 2.7: The Example SAGE2 software

## 2.2 Literature Review

### 2.2.1 CRUMBS: a Cyber Security Framework Browser

This research is about framework contextualization. It focuses on the Italian Adaptation of the Cyber Security Framework (IACSF), and the framework that is used in this paper is the NIST Cybersecurity Framework. The contents inside NIST are divided into a hierarchy from functions to subcategories. In order to make it easier to understand, IACSF creates a priority and mandatory items for each subcategory. Priorities help in category implementation. Users will know what category they should implement first. Mandatory items help in balancing their budgets and requirements. Each subcategory has four mandatory levels from the least complex to the most complex in order to make it suitable for its organizations. This browser contextualizes from the current profile (as is) and the target profile (to be) of the organizations.



Figure 2.8: Contextualization overview of NIST cybersecurity framework layer and Priority level, and Mandatory items as columns

Figure 2.8 demonstrates a context that contains the framework, and priority (PR) and maturities (M1 to M4) of each subcategory. The leftmost bar is used for zooming to a specific period that allows users to see the information easily. Moreover, the users can choose only objects they want to see, and the leftover will be changed to grey color to emphasize wanted content.



Figure 2.9: Current profile detail. Mouse-hoovering on controls and subcategories provides full details about inspected items.

After the users select their current profile and target profile, the context will be changed to demonstrate differences between them. In the framework, it contextualizes the ratio of maturity implementation completion of each function, category, and subcategory. In addition, maturities will be classified into three colors which are full green, half green-red, and full red according to maturity implementation. For example, the full green means already implemented, the half green-red means partially implemented.

This context demonstrates a gap analysis between two profiles. The left side of the gap analysis shows the ratio between unimplemented maturity and implemented maturity that is necessary to reach the target profile. The right side of the gap analysis shows the ratio between unimplemented maturity and implemented maturity that is higher than the target profile. Moreover, if the users want to see only a maturity, they just hover the cursor on the part that they want to see. Then, the maturities related to the part will be highlighted to emphasize it.

We may apply this paper to our project in three parts. First of all, because NIST and ISO/IEC27001 are similar with regard to hierarchy, we can try using this visualization with ISO/IEC27001. The next part is the statistical view. According to Figure 2.9 and Figure 2.10, there is a statistical view in the rightmost part. The statistical view demonstrates the summary of the information in histograms. For example, the number



Figure 2.10: CRUMBS system configured for Gap analysis. The bottom-most thinner horizontal bar provides (colors) the information about the function the controls belong to. All the controls are partially ordered with respect to a suitable implementation strategy.

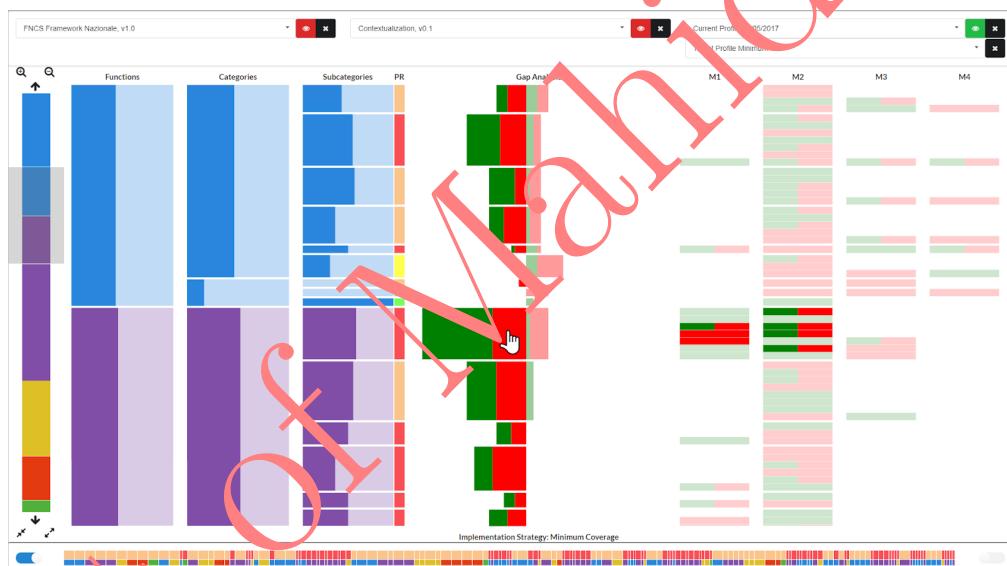


Figure 2.11: The security manager inspects a gap bar for a subcategory of the Protect function, the gap bar shows a not so big distance between the current and target statuses.

of selected subcategories, maturities, etc. help the users understand it easier. The final part is highlighting. Because of the amount of content in a standard, the users may be confused, so highlighting will be an important role for emphasis on content to make it clearer as in Figure 2.11.

### 2.2.2 Visualizing Cyber Security: Usable Workspaces

This research is about the cyber analytic tasks and visualization in a large high-resolution display work environment. Their study consisted of cybersecurity analysts being given the VAST 2009 challenge dataset to analyze and determine whether or not there was a malicious insider exfiltrating information in the prepared work environment. The researchers interviewed the cybersecurity analysts about the use of high-resolution displays and created a set of visual prototypes. Figure 2.12 shows a PowerPoint prototype of a history tree workspace that they offer. It provides a means for easily retracing the process when it comes to writing a report. Furthermore, they used these prototypes to gain further feedback from cyber analysts. As a result, they claim that additional display space and resolution is beneficial. However, the tools, input devices, and window managers currently used in this field do not make good use of the added display space.

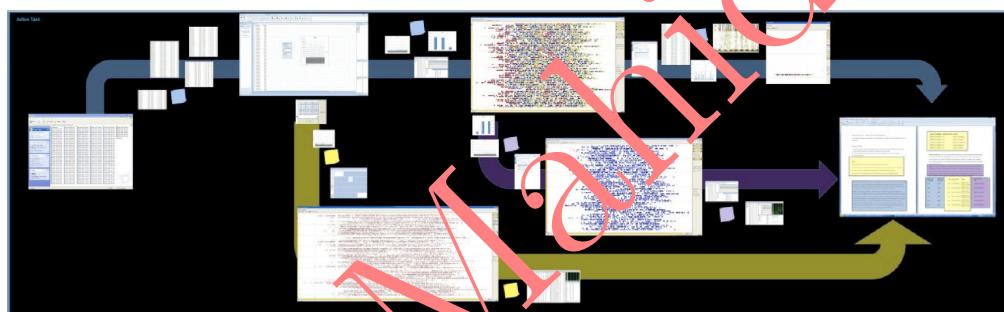


Figure 2.12: History trees, a mockup visual workspace showing history and allowing workflow traceability through the use of keyframes, resulting in the final report on the far right

We may apply this research in our project in a way to apply high resolution displays to our visualization. According to the research, with higher resolution, the information about ISO security standards can be displayed in more detail. Moreover, users can see the overview of ISO security standards clearly so that they would get a better understanding of ISO security standards and be able to apply it with their Information Security Management System.

### 2.2.3 SAGE2: A New Approach for Data Intensive Collaboration Using Scal-

### able Resolution Shared Displays

This research is about the implementation of SAGE2 and its benefits. SAGE2 is an open-source middleware that provides multiple users with a common operating environment to access, display, and share an assortment of data-intensive information as shown in Figure 2.13. The SAGE2 was a redesign and development of SAGE software by the University of Illinois Department of Computer Science using cloud-based and web-browser technologies in order to enhance data intensive co-located and remote collaboration with Scalable Resolution Shared Displays. Using SAGE2 as a platform to develop The Scalable Resolution Shared Displays enables SAGE2 to support content with unlimited resolution. In other words, SAGE2 supports any resolution of the display screen, while a display with higher resolution make it easier to visualize large volumes of data in a collaborative environment. Moreover, SAGE2 supports a wide bundle of content including images, videos, PDFs, 2D and 3D custom applications, and other applications from a remote source. SAGE2 supports many Operating Systems including Windows, macOS, and Linux. We decided to develop our visualization on the SAGE2 platform because of the benefit of SAGE2, as stated in the research, that allows users to collaborate working with each other from anywhere. Furthermore, SAGE2 can be accessed on many platforms with any resolution as well. Moreover, SAGE2 is a free platform. We also could develop an application and use it on SAGE2 without further installation. The applications on SAGE2 are based on JavaScript and Node.JS language, so we would build our application using these programming languages.

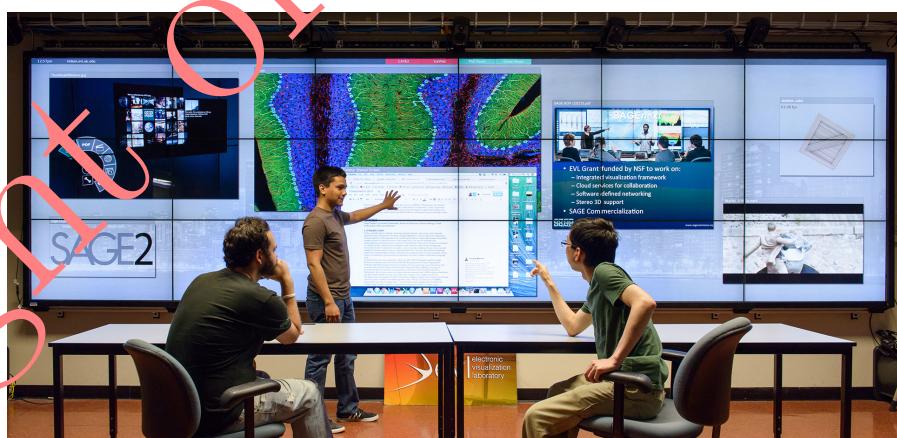


Figure 2.13: Example of using SAGE2 in the conference.

### 2.2.4 DOI Trees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data

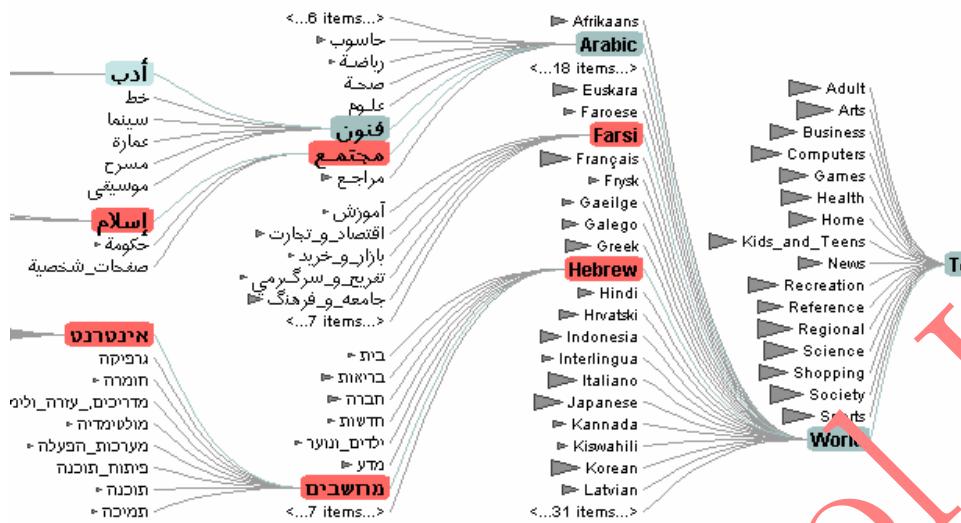


Figure 2.14: DOI Tree visualization of the Open Directory Project, containing over 600,000 nodes.

This research is about the extension of the Degree-of-Interest Tree browser (DOI Tree), Figure 2.14. The tree visualization technique is good because it is very useful and easy to control. Moreover, since the amount of data grows rapidly, but the resolution to display cannot follow that growth, Tree visualization improvement is desirable. This research improves DOI Tree by using a DOI calculation and TreeBlock, a space-constrained multi-focal tree layout technique. The visibility of each node depends on the interest of the users. When the users input a query to the system, it will use a Degree-of-Interest calculation to calculate the interest value of each node. Then, the system will have a threshold to decrease the size of the tree. The nodes that have the value less than the threshold will be hidden. For the layout, there are three steps to create a tree. The first step is space computation. In this step, the tree will divide each subtree into a block and calculate the width and the height of the block in pixels. The second step is a comparison between the area of the block and available space. If the area exceeds the threshold value, the size will be reduced. The last step is the assignment of the position of the blocks. The system must ensure that the parent block will be assigned before its children blocks.

This research is evaluated by comparison computation times for three approaches

of DOI which are a naïve tree traversal, a pruned tree traversal, and a disinterest thresholding approach. The experiment runs on a 1GHz Pentium III IBM ThinkPad T23 with 256MB RAM. The result is that, while the number of nodes increase, the disinterest thresholding approach has almost linear running time in FISHEYE view which is better than CRUMBs and usable workspaces which have been mentioned before.

We can apply this research to our project by using a focus node feature to emphasize the node that users want to see. Moreover, we can apply certain features of this project to eliminate the unimportant components from tree levels.

Copyright of Mahidol University

## CHAPTER 3

### ANALYSIS AND DESIGN

#### 3.1 System Architecture Overview

The conceptual diagram in Figure 3.1 provides an overview of our visualization program processes. The process starts with Natural Language Processing. In the Natural Language Processing process, the raw text would be used in data preprocessing and topic modeling. As a result, we would get the context of annexes and clauses of the ISO security standard including their relationship as well. The context and relationship would be kept in a JSON file so that we could retrieve the information whenever it was needed. However, the output from the NLP process will be checked with the manual mapping between ISO/IEC 27001:2013 and ISO/IEC 27002:2013. Then, the JavaScript and Node.js code would take the information from the JSON file to construct a visualization and display it on the multiple-display walls through the SAGE2 platform. In the visualization, the used JSON files will be the manually mapped ones until the result from the NLP process reaches sufficient correctness.

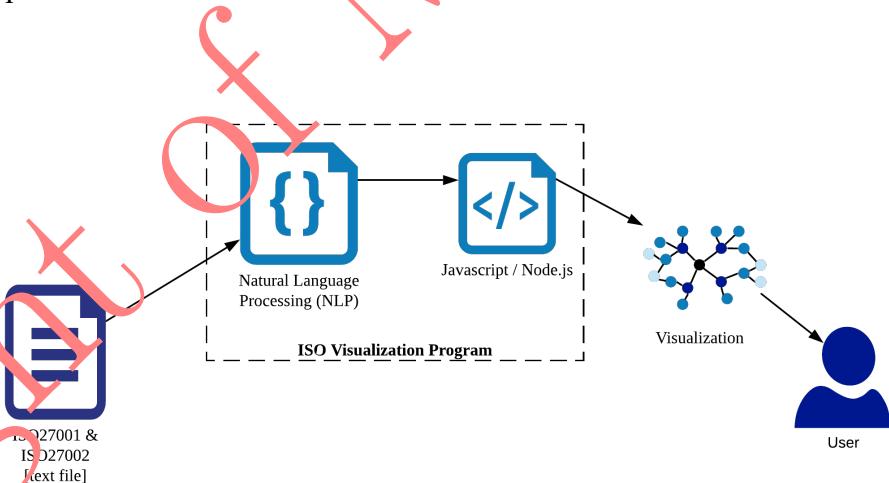


Figure 3.1: The system architecture of the ISO visualization program

### 3.2 System Structure Chart

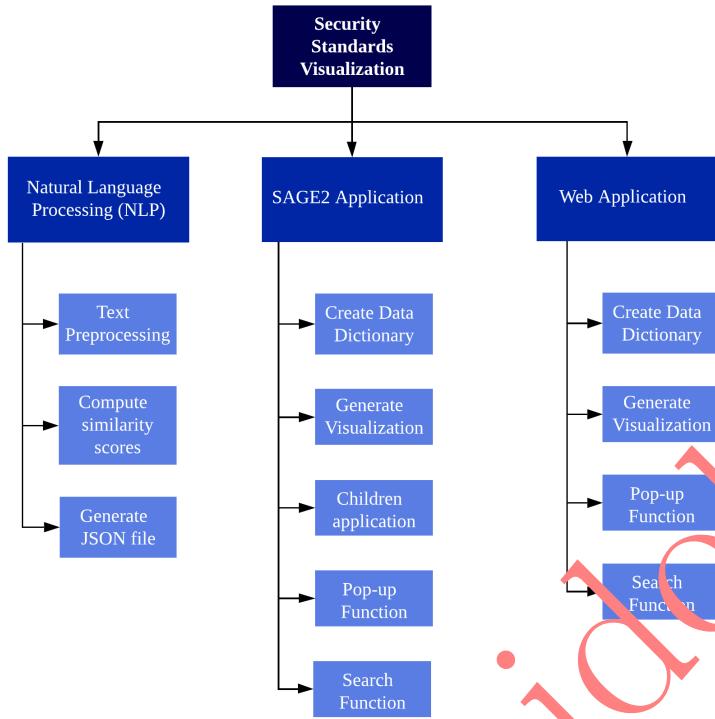


Figure 3.2: The structure chart of the ISO visualization program

The structure chart in Figure 3.2 provides an overview of our visualization program processes. There are three main functions of the visualization. The first function is Natural Language Processing. In this function, there are three processes including *Text preprocessing, computing similarity scores, and generating JSON files*. The text preprocessing process is to prepare the data from ISO/IEC 27001:2013 and ISO/IEC 27002:2013 before using those data to find the similarity between them. Then the result from the similarity score will be used to create JSON files. These JSON files from the NLP function will be used in the second and third function if the result provides enough correctness, or the second and third functions will use the JSON file that has been manually mapped between ISO/IEC 27001:2013 and ISO/IEC 27002:2013 instead. In the second function, the SAGE2 application contains five processes. Those processes are *creating data dictionary, generating visualization, search function, pop-up function, and children application function*. The *Create data dictionary function* retrieves the JSON files and uses them to create a tree data structure that will be used to generate the vi-

sualization. *The Search function* allows users to search for the term that is included in the topic of ISO documents. *The Children application function* is the function that opens the child application when users interact with the visualization and the interacted node is not the leaf of a tree data structure. Otherwise, *the pop-up function* will be performed instead. The final function is a web application with similar processes as the second function which includes *creating data dictionary, generate visualization, search function, and pop-up function*. The open child application is the exclusive process in SAGE2. The overview of the structure chart can be viewed in Figure 3.2.

### 3.3 Data Flow Diagram

#### 3.3.1 Data Flow Diagram Level 0

The Data Flow Diagram Level 0 shown in Figure 3.3 represents the communication between input data, our visualization program, and users. The input data are ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents. Our program will receive the raw text from those two documents as input data to generate a visualization and store it inside our system; then, it will provide the visualization to the user and react to the queries that a user has input.

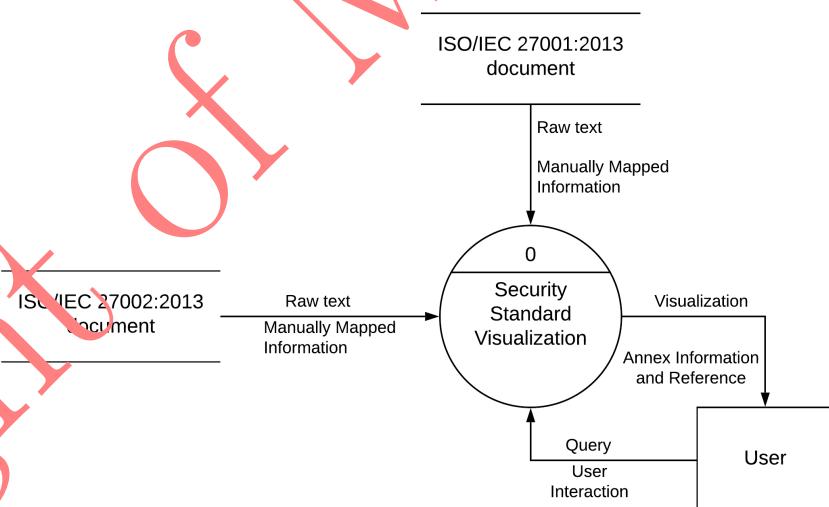


Figure 3.3: The level 0 data flow diagram of the ISO visualization system.

### 3.3.2 Data Flow Diagram Level 1

In the Data Flow Diagram Level 1, there are three functions in the system which are the NLP system, the SAGE2 application, and the web application. The input from ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents are processed in the Natural Language Processing system. The output from this process will be stored in the system as JSON files. The example result of output JSON files can be seen in Figure 3.4. In the SAGE2 application process, data from the JSON files are pulled and used in this process as well as in web application process. Both of the applications also receive input from users such as queries and mouse input while showing a visualization to the users. The overview of dataflow diagram level 1 can be seen in Figure 3.5.

```
{
  "name" : "Context of the organization",
  "clause" : "4",
  "content" : [
    {
      "scname" : "Understanding the organization and its context",
      "sclause" : "4.1",
      "annex" : ["A.7.1.1", "A.8.2.1", "A.9.1.1", "A.9.2.2", "A.9.4.1", "A.14.1.1"]
    },
    {
      "scname" : "Understanding the needs and expectations of interested parties",
      "sclause" : "4.2",
      "annex" : ["A.6.2.2", "A.7.1.1", "A.7.2.2", "A.7.3.1", "A.8.2.1", "A.8.3.3", "A.10.1.2", "A.11.2.2", "A.11.2.4", "A.11.2.5", "A.15.1.1", "A.15.1.2", "A.15.1.3", "A.15.2.2"]
    },
    {
      "scname" : "Determining the scope of the information security management system",
      "sclause" : "4.3",
      "annex" : ["A.9.1.1", "A.11.1.1"]
    },
    {
      "scname" : "Information security management system",
      "sclause" : "4.4",
      "annex" : ["A.14.2.6", "A.17.2.1"]
    }
  ]
},
```

(a) The structure of ISO/IEC 27001:2013

```
{
  "name" : "Information Security Policies",
  "annex" : "5",
  "subtopic" : [
    {
      "name" : "Management Direction for Information Security",
      "annex" : "5.1",
      "subtopic" : [
        {
          "name" : "Policies for Information Security",
          "annex" : "5.1.1",
          "control" : ["4.1", "4.2", "4.4", "5.2", "6.2"],
          "page" : "2"
        },
        {
          "name" : "Review of the Policies for Information Security",
          "annex" : "5.1.2",
          "control" : ["9.1.1"],
          "page" : "3"
        }
      ]
    }
  ],
}
```

(b) The structure of ISO/IEC 27002:2013

Figure 3.4: The example of structure from the create JSON process

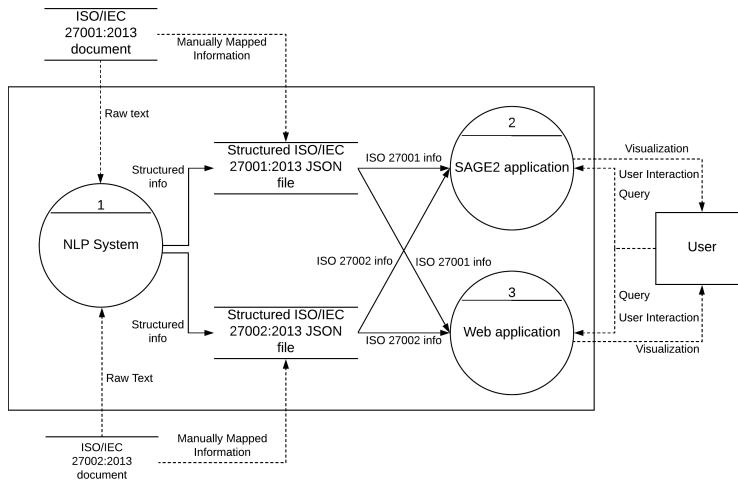


Figure 3.5: The level 1 data flow diagram of the ISO visualization system.

### 3.3.3 Data Flow Diagram Level 2

In the Data Flow Diagram Level 2, the NLP system contains<sup>2</sup> processes including *performing text preprocessing*, *computing similarity scores*, and *creating a JSON file*. The text preprocessing process receives raw text from ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents and produces results of a group of corpuses. The next process receives a group of corpuses and computes the similarity scores from those corpuses. The similarity results will be used to create JSON files which will be used in the SAGE2 and web applications if the similarity score is high enough. Otherwise, the SAGE2 application and web application will use the manually mapped JSON files instead. The process in the SAGE2 application starts with using data from JSON files to create data dictionaries. Then a data dictionary will be used to generate a visualization. After the visualization is created, the search function will be available. In the search function, it starts when a user inputs a search query. Then, the query will be used for creating data dictionaries. Users can interact with the system by sending mouse input which can result in two processes. The first one is the open child application which opens a new application that contains the information of the node that a user clicked. The other function is when a user clicked on an annex node. The visualization will open a pop-up application that contains information about the clicked annex and its reference in the document. For the web application part, most of the function is similar to the SAGE2

application. The different part is that the web application does not contain open the child application process. The overview of dataflow diagram level 2 can be seen in Figure 3.6.

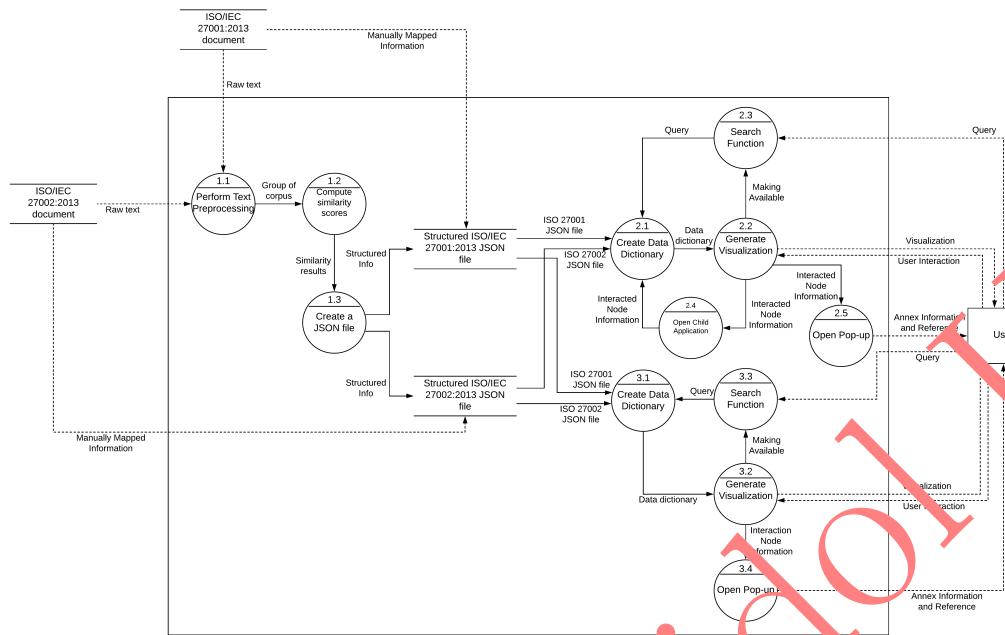


Figure 3.6: The level 2 data flow diagram of ISO visualization system.

### 3.4 Visualization

The visualization styles that will be used to implement the application consisted of two visualization styles.

#### 3.4.1 Sunburst Visualization

Sunburst visualization is one of the visualization styles in D3.js [20] that is suitable to represent the hierarchy of data. It represents the data as a circle with the main topic in the middle and shows the cumulative values of subtrees which are suitable with the ISO/IEC 27001:2013 structure. The example of Sunburst visualization can be seen in Figure 3.7.

#### 3.4.2 Zoomable Circle Packing Visualization

Zoomable Circle Packing visualization is one of the visualization styles in D3.js [20] that is suitable for representing the hierarchy of data. The reason to choose Zoomable Circle Packing visualization is because this visualization can represent the tree data struc-



Figure 3.7: The overview of Sunburst visualization that represents the ISO/IEC 27001:2013 information. It can be seen that all the ISO information is represented in the tree structure with the root node as a center circle.

ture with the circles for each subtree. Moreover, each circle contains smaller circles as its children which can represent the ISO/IEC 27001:2013 information. The example of a Zoomable Circle Packing visualization can be seen in Figure 3.8.

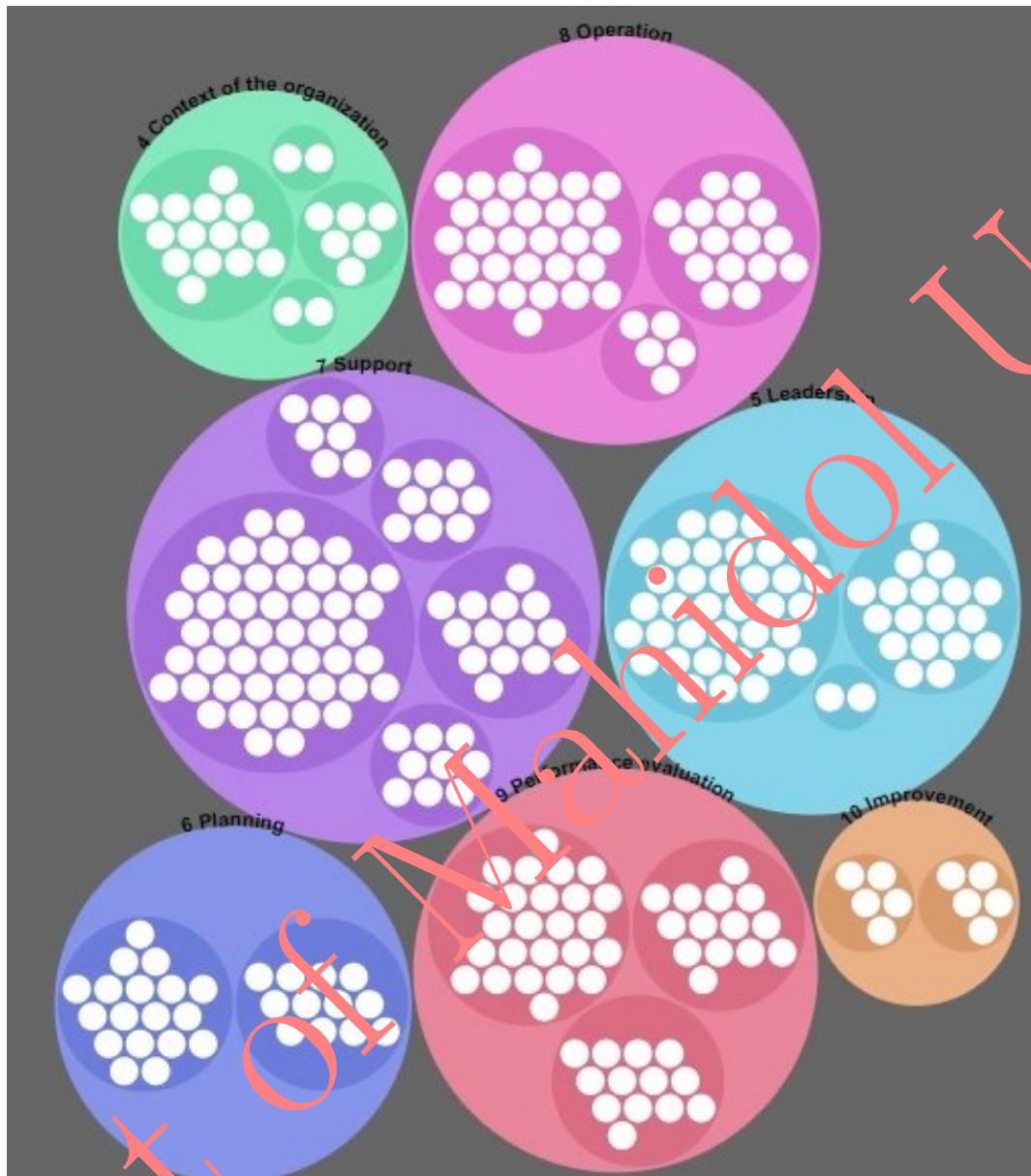


Figure 3.8: The overview of the Zoomable Circle Packing visualization that represents the ISO/IEC 27001:2013 information. This visualization represents the ISO information with each big circle as a subtree for each topic in the document. It can make users easily see the overview of the ISO information structure.

## CHAPTER 4

### IMPLEMENTATION

This chapter contains 2 topics which include Hardware & System Environment and Implementation guide & Techniques. The hardware and system environment are the required hardware system and applications for developing the application. On the other hand, the implementation guide and techniques are the processes of developing the visualization application.

#### 4.1 System Environment

- Operating System and Utilities Application
  - Microsoft Windows 10 - To operate the computer
  - SAGE2 version 4.0.12 - To provide the platform for the application
- Web Server Software
  - Web hosting domain - To host the web application
- Editor
  - Microsoft Visual Studio - To develop Python projects for Natural Language Processing (NLP)
  - Brackets - To develop web-based and SAGE2 application
- Component
  - SAGE2 Display wall - To test and display the application on SAGE2

#### 4.2 Implementation Guide and Techniques

The visualization application is divided into 3 parts which are processing data, SAGE2 application, and Web application. This section presents briefly about how the code has been implemented. For the full version of the codes, see Appendix B.

#### 4.2.1 Processing data

The purpose of this part is to prepare and process the data that will be used in visualization. There are three process in this part including Text preprocessing, Finding correlation and Create JSON files.

##### Text Preprocessing

Before finding the correlations between clauses and annexes in ISO/IEC 27001:2013 and ISO/IEC 27002:2013, it is necessary to prepare data first. As shown in Listing 4.1 and 4.2, the input from ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents had been converted into text files; then, with the SpaCy library [21], each line in the text files were separated into tokens using *nlp()* function. For the first token of the line, it will be checked whether the line is the headline of the contents; otherwise it will be determined as a detail of each clause or annex to be processed for finding the correlation later. Each part of the document will be collected into variables to be used later. The variables are dictionary-type variables including:

1. *mapNameCls* that maps between clause No. and clause name
2. *mapClause* that maps between clause No. and its sub clause list
3. *mapNameSub* that maps between sub clause No. and subclause name
4. *mapDetail* that maps between subclause No. and its detail
5. *mapNameA* that maps between annex No. and its name
6. *mapNameSubA* that maps between sub-annex No. and its name
7. *mapDetailA* that maps between sub-annex No. and its detail
8. *mapNameSubSub* that maps between sub of sub-annex No. and its detail

Figure 4.1 shows the example of the mapping results of *mapClause* and *mapDetail* variable which are the variables for mapping a clause number with its sub clause list and mapping a sub clause number with its detail, respectively.

Listing 4.1: Code for text preprocessing of ISO/IEC 27001:2013

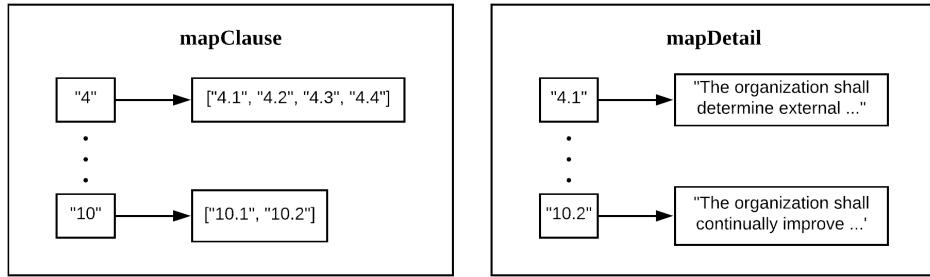
```
import os
import spacy
for filename in os.listdir(directory):
    if(filename.endswith(".txt")):
        f = open(os.path.join(directory, filename))
        for x in f:
            doc = nlp(x)
            # check number pattern of clause
            if(doc[0].shape_ == 'dd' or doc[0].shape_ == 'd'):
                mapClause[clsno] = subclslist
                subclslist = []
                mapNameCls[doc[0].text] = doc[2:len(doc)-1].text # map Clause no with Clause name
                clsno = doc[0].text # map Clause no with Clause name
                if(doc[1].text == ')'):
                    detail += "\n" + doc[0:len(doc)].text # append detail of SubClause
                    if(not detail): # map detail of the last SubClause if detail is not empty
                        mapDetail[subcls] = detail
                        mapDetail[subSub] = detail
                        continue # continue to newline (detail)
                    # check number pattern of subclause
                    if(doc[0].shape_ == 'd.d' or doc[0].shape_ == 'd.dd' or doc[0].shape_ == 'dd.d' or doc[0].shape_ == 'dd.dd'):
                        mapDetail[subcls] = detail # map detail of the last SubClause
                        subcls = doc[0].text # get New SubClause No
                        subclslist.append(subcls)

                mapNameSub[subcls] = doc[2:len(doc)-1].text # map SubClause no with SubClause name
                detail = doc[2:len(doc)-1].text # init new detail of SubClause
                continue
                else:
                    detail += "\n" + doc[0:len(doc)].text # append detail of SubClause
                    f.close()
                else:
                    continue
            # housekeeping for the empty key
            mapDetail[subcls] = detail # map detail of the last SubClause
            dict.pop(mapDetail, None) # delete value of empty key (debug)
            mapClause[clsno] = subclslist
            dict.pop(mapClause, None)
            empty_keys = [k for k,v in mapClause.items() if not v]
            for k in empty_keys:
                del mapClause[k]
```

Listing 4.2: Code for text preprocessing of ISO/IEC 27002:2013

```
annex = open("27002.txt")
for x in annex:
    doc = nlp(x)

    if(doc[0].shape_ == 'dd' or doc[0].shape_ == 'd'):
        aNo = "A" + doc[0].text
        mapNameA[aNo] = doc[2:len(doc)-1].text # map Annex no with Annex name
        if(doc[1].text == ')'):
            detail += "\n" + doc[0:len(doc)].text # append detail of SubAnnex
        if(not detail2):
            mapDetailTemp[subSubNo] = detail2 # map detail of the last SubAnnex if detail is not empty
            mapDetailA[subANo] = mapDetailTemp
            mapDetailTemp = {}
            continue
        if(doc[0].shape_ == 'd.d' or doc[0].shape_ == 'd.dd' or doc[0].shape_ == 'dd.d' or doc[0].shape_ == 'dd.dd'):
            mapDetailTemp[subSubNo] = detail2
            detail2 = ""
            subSubNo = ""
            mapDetailA[subANo] = mapDetailTemp # map detail of the last SubAnnex
            mapDetailTemp = {}
            subANo = "A" + doc[0].text # get New SubAnnex No
            mapNameSubA[subANo] = doc[2:len(doc)-1].text # map SubAnnex no with SubAnnex name
            detail = doc[2:len(doc)-1].text # init new detail of SubAnnex
            continue
        if(doc[0].shape_ == 'd.d.d' or doc[0].shape_ == 'd.dd.d' or doc[0].shape_ == 'd.d.dd' or doc[0].shape_ == 'd.dd.dd' or doc
           [0].shape_ == 'dd.d.d' or doc[0].shape_ == 'dd.d.dd' or doc[0].shape_ == 'dd.dd.d' or doc[0].shape_ == 'dd.dd.
           dd'):
            if(doc[1].text == ')'):
                detail2 += "\n" + doc[0:len(doc)].text
            continue
        mapDetailTemp[subSubNo] = detail2
        subSubNo = "A" + doc[0].text # get New SubAnnex No
        subSub = doc[2:len(doc)-1].text
        mapNameSubSub[subSubNo] = subSub # map SubAnnex no with SubAnnex name
        detail2 = doc[2:len(doc)-1].text # init new detail of SubAnnex
        else:
            detail2 += "\n" + doc[0:len(doc)].text
        mapDetailA[subANo] = mapDetailTemp
        dict.pop(mapDetailA, "")
    annex.close()
```

Figure 4.1: The example for *mapClause* and *mapDetail* variables

### Finding correlations for each corpus

To compute the similarity scores between clauses and annexes, the ‘corpusAll’ variable will collect both sub clause details from ISO/IEC 27001:2013 and sub annex details from ISO/IEC 27002:2013 as a corpus list; consequently, with *TfidfVectorizer* function from scikit-learn [22], it will generate the tf-idf sparse matrix of the corpus list with its setting. In this project, the corpus will be tokenized<sup>1</sup> and lemmatized using a function from the spacy library, and stop words in English will be eliminated from the term list as shown in Listing 4.3.

Listing 4.3: Creating sparse matrix from *TfidfVectorizer* library

```

def keep_token(t):
    return (t.is_alpha and
            not (t.is_space or t.is_punct or
            t.is_stop or t.like_num or (len(t.text)==1)))
def lemmatize_doc(doc):
    doc = nlp(doc)
    return [t.lemma_.lower() for t in doc if keep_token(t)]
# extract word by using tokenization and lemmatization from spacy
# eliminate stop words in English
# corpusAll: all subclauses and sub annexes
vectorizer = TfidfVectorizer(analyzer=lemmatize_doc, stop_words='english')
tfword = vectorizer.fit_transform(corpusAll)

```

Then, the program shown in Listing 4.4, will compute cosine similarity results of the sparse matrix with *coseine\_similarity()* function from the *sklearn.metrics.pairwise* library and keep those results mapped with its clause or annex number by checking whether the mapped corpuses are a clause or annex. Figure 4.2 shows certain results from the IDF table of terms-document matrix while each row represents documents in which documents from ISO/IEC 27001:2013 are shown in dark blue color and documents from

ISO/IEC 27002:2013 are shown in light blue color and each column represents terms within documents.

**Listing 4.4:** Computing similarity scores from every corpus

```
for i in range(len(corpusAll)):
    cosim = cosine_similarity(tfword[i:i+1], tfword)
    count = 0
    tmp = {}
    simTmp = []
    aout = inCorpus[i].startswith('A.')

    for a in cosim:
        for sim in a:
            ain = inCorpus[count].startswith('A.')
            if aout:
                if(not ain):
                    tmp[inCorpus[count]] = sim
                    simTmp.append(sim)
                else:
                    if ain:
                        tmp[inCorpus[count]] = sim
                        simTmp.append(sim)
            count = count + 1
            if aout:
                annexSim[inCorpus[i]] = tmp
                allAnnexSim[inCorpus[i]] = simTmp
                else:
                    clsSim[inCorpus[i]] = tmp
                    allClsSim[inCorpus[i]] = simTmp
```

In the code shown in Listing 4.5, after computing the similarity results between clause and annex, the results will be filtered by the average of its similarity score to extract its related clause or annex for creating a JSON file in the later process. Since its similarity scores are mostly close to the others, its average score will be added with its standard deviation for filtering.

**Listing 4.5:** Filtering similarity scores with its average scores

```
for cls in clsSim.items():
    resTemp = {}
    clsRes = []
    for ann, sim in alist.items():
        if(sim > (np.average(allClsSim[cls])+(np.std(allClsSim[cls])))):
            resTemp[annex] = sim
            clsRes.append(annex)
            resultCls[cls] = resTemp
    mapClsRes[cls] = clsRes
```

Figure 4.2: The idf table result

### Creating JSON files

After finding the similarities between ISO/IEC 27001:2013 and ISO/IEC 27002:2013, the result will be used to generate a JSON file for creating a visualization. With the results collected in dictionary variables, it is able to use *dumps()* function to create a JSON file using its keys and values inside the variable as shown in Listing 4.6. The example for using a dictionary variable to keep a key with its value for a JSON file is shown in Figure 4.3.

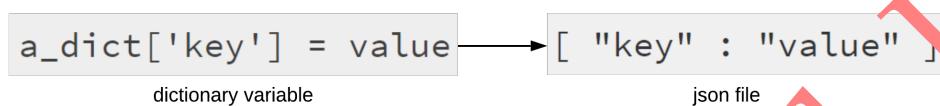


Figure 4.3: The example of dictionary variable to JSON file

Listing 4.6: Code for creating JSON file

```

j = open("data.json", "w")
j_ls = []
tmp_cls = {}
tmp_sub = {}
ann_ls = []
con_ls = []
for clsno, sub in mapClause.items(): # mapClause is a dictionary that maps between clause no and the list of that clause
    tmp_cls = {}
    tmp_cls["name"] = mapNameCls[clsno]
    tmp_cls["clause"] = clsno
    con_ls = []
    for subcls in sub:
        tmp_sub = {}
        if(subcls=='0'):
            continue
        tmp_sub["sename"] = mapNameSub[subcls]
        tmp_sub["clause"] = subcls
        ann_ls = []
        for ann in results[subcls]:
            ann_ls.append(ann)
        tmp_sub["annex"] = ann_ls
        con_ls.append(tmp_sub)
        tmp_cls["content"] = con_ls
    j_ls.append(tmp_cls)
jfile = json.dumps(j_ls, indent=4)
j.write(jfile)
j.close()
  
```

As a result, the JSON file will be generated following the template as in Figure 4.4.

```
[
  {
    "name": "clause name",
    "clause": "clause no",
    "content": [
      {
        "scname": "subclause1 name",
        "sclause": "subclause1 no",
        "annex": ["A.4.1.1", "A.4.1.2", "A.7.1.2"...]
      },
      {...}, ... ,
    ],
    ...
  ]
]
```

Figure 4.4: The example of the result JSON file

#### 4.2.2 SAGE2 Application

##### Sunburst Visualization

The implementation of Sunburst visualization on the SAGE2 application starts with initializing the SAGE2 application setting. The code shown in Listing 4.7 shows the process of creating the element that will be shown in the application and its attribute. The first element acts as a container for another element, while the second element acts as the container for the text input element. The other two input elements are the submit button and the clear button as shown in Figure 4.5. The final element is the svg element which will be used to append the visualization.

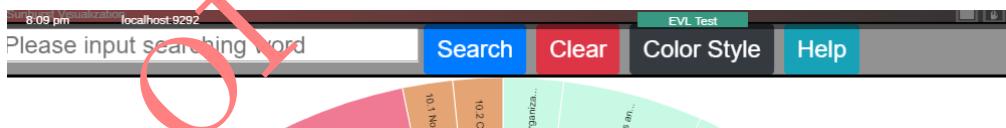


Figure 4.5: The search bar and other buttons in the Sunburst visualization SAGE2 application

Listing 4.7: Create element for the Sunburst Visaulization application

```
/* Form element */
const form = d3.select(this.element)
.append("div")
.style("background-color", "#929292")
.style("border-bottom", "3px solid black")
.style("font", "16px Arial");

/* Create container use for auto-complete function and append it to the form */
const autocomplete = form.append("div")
```

```

.attr("id", _this.id + "autocomplete")
.style("position", "relative")
.style('display', 'inline-block');

/* Create text field for query and append it to the form */
const query = autocomplete.append("input")
.attr("type", "text")
.attr("id", _this.id + "query")
.attr("placeholder", "Please input searching word")
.attr("size", "30")
.style("font-size", "24px");

/* Create submit button and append it to the form */
const submitButton = form.append("input")
.attr("type", "submit")
.attr("value", "Search")
.on("click", function(){
  _this.dict = _this.createDictionary1(_this.iso27001);
  _this.createSunburst(_this.dict, _this);
})
.style("font-size", "24px");

/* Create clear pop-up button and refresh the screen and append it to the form */
const clearButton = form.append("input")
.attr("type", "submit")
.attr("value", "Clear")
.on("click", function(){
  document.getElementById(_this.id + 'query').value = "";
  _this.createSunburst(_this.createDictionary1(_this.iso27001), _this);
  _this.sendDataToChildrenApps("close");
})
.style("font-size", "24px");

const svg = d3.select(this.element)
.append("svg")
.attr("id", _this.id + "partitionSVG")
.style("width", "2000")
.style("height", "2000");

```

The code in Listing 4.8 is the first part of the application which reads the information of data structure of ISO/IEC 27001:2013 and ISO/IEC 27002: 2013 that have already been processed from the previous step. The information will be kept as JSON objects and will be used to create a visualization. Also, the information from this function will be used as a wordpool for the auto-complete function.

After receiving the JSON files, the createSunburst function shows the process of creating the visualization using D3.js library [20]. It starts with receiving the ISO27001 information from the previous process as a parameter and uses it to create a visualization node as shown in Listing 4.9. The size of the visualization is also set in this part.

Listing 4.9: Transform data into a node for the visualization

Listing 4.8: Create dictionary from ISO/IEC 27001 :2013 and 27002 :2013

```
/* Read JSON file of ISO/IEC 27001:2013 */
function read27001(){
readFile(_this.resrcPath + 'Data/ISO27001.json', async function(error, data2) {
...
}, "JSON");
}

/* Read JSON file of ISO/IEC 27002:2013 */
function read27002(){
readFile(_this.resrcPath + 'Data/ISO27002.json', async function(error, data1){
...
}, "JSON");
}
```

```
// Set property for the visualization
const format = d3.format(",d");
const width = 2000;
const height = 2000;
const radius = width / 6;
const arc = d3.arc()
.startAngle(d => d.x0)
.endAngle(d => d.x1)
.padAngle(d => Math.min((d.x1 - d.x0) / 2, 0.005))
.padRadius(radius * 1.5)
.innerRadius(d => d.y0 * radius)
.outerRadius(d => Math.max(d.y0 * radius, d.y1 * radius - 0));
const partition = data => {
const root = d3.hierarchy(data)
.sum(d => d.size)
return d3.partition()
.size([2 * Math.PI, root.height + 1])(root);
};
//create a root node from the ISO information
const root = partition(data);
```

The next part is the JavaScript code in Listing 4.10 which shows the process of creating svg tags or nodes from the information gained from the root node. Then each node will be assigned its data from the ISO information appended to its parent according to the ISO information. This function also contains the setting for each node such as node color, transparency, and highlight stroke as shown in Figure 4.6. Finally, the root node will append to the SAGE2 application as shown in Listing 4.10.

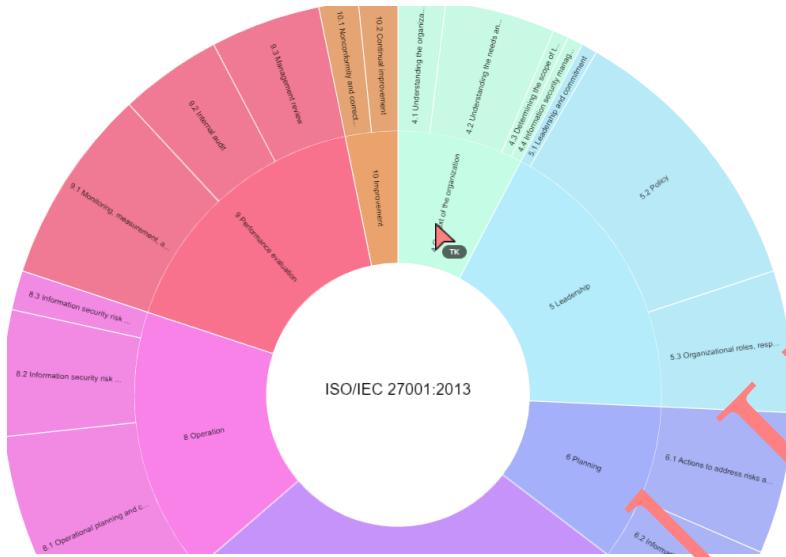


Figure 4.6: The nodes in the Sunburst visualization SAGI2 application

#### Listing 4.10: Create node for visualization

```
// Append the visualization to the container that have been prepared.
const svg = d3.select('#' + rt.id + 'partitionSVG')
  .attr("viewBox", [0, 0, width*2, width*2])
  .style("font", "20px sans-serif");

const g = svg.append("g")
  .attr("transform", `translate(${width / 2},${width / 2})`)

// Create each node and assign information to it.
const path = g.append("g")
  .selectAll("path")
  .data(root.descendants().slice(1))
  .join("path")
  .attr('fill', d => d.depth > 0 ? h(`#${hval(d.data.color)}`, 100%, 50%) : 'white')
  .attr("fill-opacity", d => arcVisible(d.current) ? (d.children ? 0.6 : 0.4) : 0)
  .attr("d", d => arc(d.current))
  .attr('stroke', d => arcVisible(d.current) && (d.data.highlight || d.data.childHighlight) ? "#F00" : null)
  .attr('stroke-width', d => arcVisible(d.current) && (d.data.highlight || d.data.childHighlight) ? 3 : 0)
  .on("click", pc, Up);

path.filter(d => d.children)
  .style("cursor", "pointer")
  .on("click", rt.currentState == 0 ? clicked1 : clicked2);

path.append("title")
  .text(d => `${d.data.name}`);

const label = g.append("g")
  .attr("pointer-events", "none")
  .attr("text-anchor", "middle")
  .style("user-select", "none")
  .selectAll("text")
  .data(root.descendants().slice(1))
```

```

.join("text")
.attr("dy", "0.35em")
.attr("fill-opacity", d => +labelVisible(d.current))
.attr("transform", d => labelTransform(d.current))
.style("font", d => d.children == null ? "25px sans-serif" : "20px sans-serif")
.text(d => shorterName(d.data.name));

```

Every node within the visualization except leaf nodes have two functions when clicked as shown in Listing 4.11. The first function is when the clicked node is the parent application. Then, the system will create a child application with the information of the subtree with the clicked node as a root as shown in Figure 4.7.

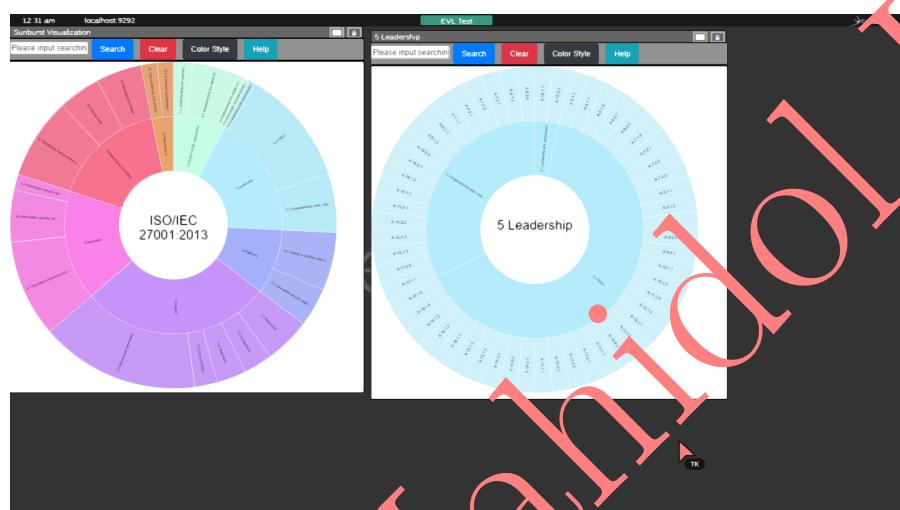


Figure 4.7: New application that shows the information of Clause 5: Leadership

Listing 4.11: Click function in each node for main application

```

// Open new application with the clicked node information
function clicked1(p) {
var a = p.data.name.split(" ");
var b = a[0].split(".");
var d = [];
d.push(rt.iso27001[b[0]-4]);
if(rt.checkClause[b[0]] == 0){
rt.checkClause[b[0]] = 1;
rt.launchAppWithValues("ISO_mapping",{
name:p.data.name, //Name of clicked clause
state: 1,
query: document.getElementById(rt.id + 'query').value, // current query in parent app
iso2: rt.iso27002
});
d = d;
x: rt.positionX, // Position X of the current application
y: rt.positionY // Position Y of the current application
}, rt.positionX + rt.sage2_width, rt.positionY + rt.sage2_height);
}
}

```

The second function is the function shown in Listing 4.12 that makes all the nodes in the visualization, including the ones that are not visible to make a transition. The clicked node will act as a root node and its child node will move to the inner circle layer while its grandchild node will move to the outer circle layer. The other node that is not related to the root node will become invisible. The example process of this function can be seen in Figure 4.8.

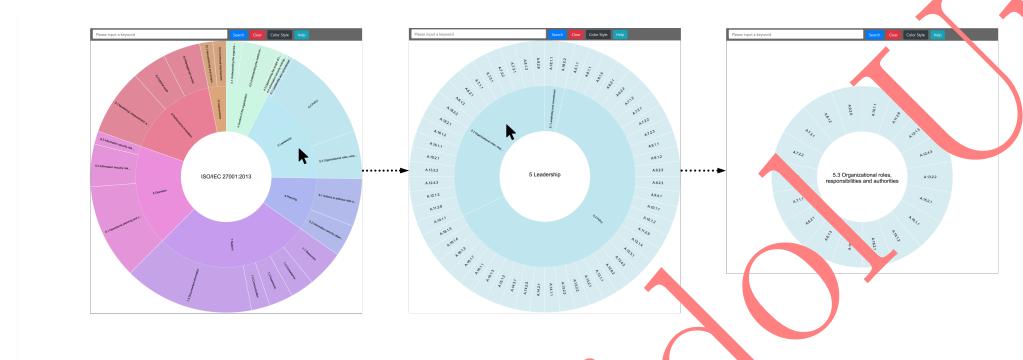


Figure 4.8: The transition process in the Sunburst visualization when user clicked on a node.

Listing 4.12: Click function in each node for child application

```
// The click function that transition the node when clicked
function clicked2(p) {
  parent.datum(p.parent || root);

  // Set up parent node's name
  if(p.data.name.length > 39){
    var tmp = p.data.name.substring(0, 2) + "..."
    parentText.text(tmp);
    parent.select("title")
    .text(`${p.data.name}`);
  }
  else {
    parentText.text(p.data.name);
    parent.select("title")
    .text(`${p.data.name}`);
  }

  root.each(d => d.target = {
    x0: Math.max(0, Math.min(1, (d.x0 - p.x0) / (p.x1 - p.x0))) * 2 * Math.PI,
    x1: Math.max(0, Math.min(1, (d.x1 - p.x0) / (p.x1 - p.x0))) * 2 * Math.PI,
    y0: Math.max(0, d.y0 - p.depth),
    y1: Math.max(0, d.y1 - p.depth)
  });
}
```

```
y1: Math.max(0, d.y1 — p.depth)
});

const t = g.transition().duration(750);

// Transition the data on all arcs, even the ones that aren't visible,
// so that if this transition is interrupted, entering arcs will start
// the next transition from the desired position.

path.transition(t)
  .tween("data", d => {
    const i = d3.interpolate(d.current, d.target);
    return t => d.current = i(t);
  })
  .filter(function (d) {
    return +this.getAttribute("fill—opacity") || arcVisible(d.target);
  })
  .attr("fill—opacity", d => arcVisible(d.target) ? (d.children ? 0.6 : 0.4) : 0)
  .attr('stroke', d => arcVisible(d.target) && (d.data.highlight || d.data.childHighlight) ? "#F00" : null)
  .attr('stroke—width', d => arcVisible(d.target) && (d.data.highlight || d.data.childHighlight) ? 3 : 0)
  .attrTween("d", d => () => arc(d.current));

label.filter(function (d) {
  return +this.getAttribute("fill—opacity") || labelVisible(d.target);
}).transition(t)
  .attr("fill—opacity", d => +labelVisible(d.target))
  .attrTween("transform", d => () => labelTransform(d.current));
}
```

The leaf nodes represent the annexes in ISO/IEC 27002: 2013 that have the content related to the information of the node's parent. Users can interact with the leaf nodes, and the application will open a pop-up window which contains the information about the annex information that is stored in the leaf node. The JavaScript code in Listing 4.13 shows the process of creating a pop-up window. It uses the parameter from the current leaf node and sends it to open a new child application for the pop-up window. The example of the pop-up application can be seen in Figure 4.9.

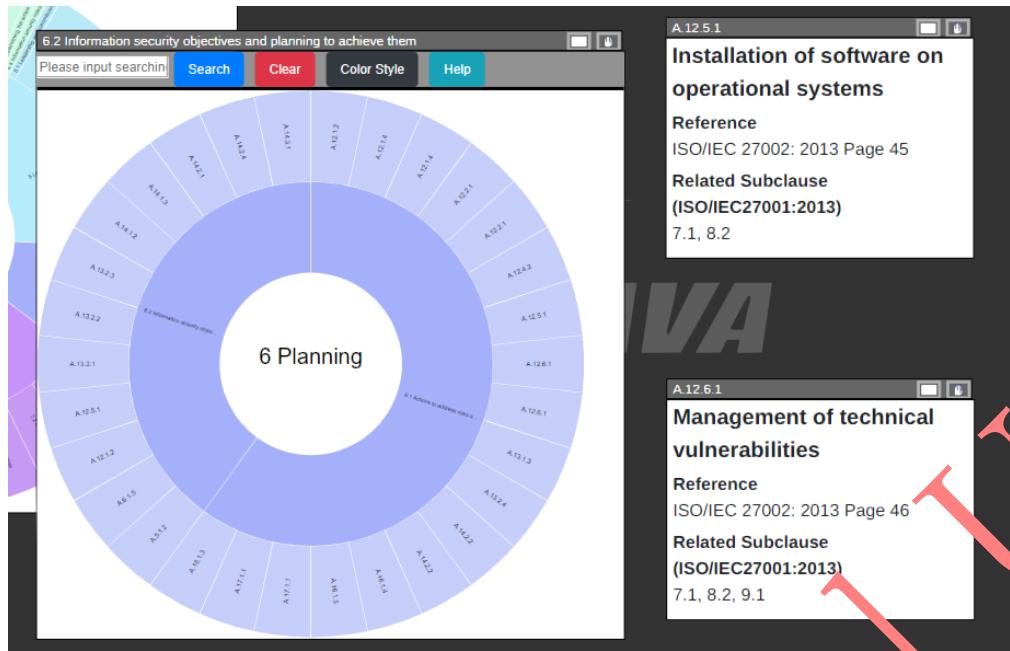


Figure 4.9: The pop-up application that shows the information of Annex 12.5.1 and 12.6.1 and their references.

Listing 4.13: Popup function for each node in the visualization (SAGE2 version)

```
// Pop-up function
function popup(d){
if(arcVisible(d.current)){
// If the popup is existed, do not create again
if(rt.checkPopup[d.data.name] == 0) {
rt.checkPopup[d.data.name] = 1;
// Start new App with these parameter
rt.launchAppWithValues("ISO_mapping",{
state: 2,
info: rt.iso27002,
name: d.data.name,
x: rt.positionX,
y: rt.positionY
}, 0, 0);
}
}
}
```

After creating the child application window, the process takes the parameters from the parent application and uses it to initialize the application as shown in Listing 4.14.

Listing 4.14: Initialize pop-up application in SAGE2

```
_this.currentState = 2;  
// Set up the new app and its content  
  
var posAdjust = {};  
posAdjust.appPositionAndSize = {};  
posAdjust.appPositionAndSize.elemId = _this.id;  
posAdjust.appPositionAndSize.elemLeft = _this.sage2_x + data.customLaunchParams.x;  
posAdjust.appPositionAndSize.elemTop = _this.sage2_y + data.customLaunchParams.y;  
posAdjust.appPositionAndSize.elemHeight = 160;  
posAdjust.appPositionAndSize.elemWidth = 300;  
  
wsio.emit("updateApplicationPositionAndSize", posAdjust);  
  
_this.updateTitle(data.customLaunchParams.name);  
_this.annexName = data.customLaunchParams.name;  
  
var annex = data.customLaunchParams.name.split(".");  
var info = data.customLaunchParams.info;  
var fname = info.children[parseInt(annex[1])-5]  
.children[parseInt(annex[2])-1].children[parseInt(annex[3])-1].name;  
var page = info.children[parseInt(annex[1])-5]  
.children[parseInt(annex[2])-1].children[parseInt(annex[3])-1].page;  
  
var cDetail = document.createElement('p');  
cDetail.innerHTML = fname;  
  
var cRef = document.createElement('p');  
cRef.innerHTML = 'page ' + page;  
  
_this.element.appendChild(cDetail);  
_this.element.appendChild(cRef);
```

### Zoomable Packing Visualization

In the Zooable visualization, there are 2 major functions which are a function for creating an object and a function for visualizing from the object as shown in Listing 4.15.

Listing 4.15: Functions in Zoomable Visualization

```
/* Create a function to create an overall object */  
function createDir(data, arrX, txt) {  
/* Create objects for annex */  
var totalAnneX = []  
for(var a = 0; a < arrX.length; a++) {  
...  
}  
/* Create objects for clause */  
var clauseList = []  
for(var i = 0; i < data.length; i++) {  
...  
}
```

In the first function, *creating an object*, it receives three parameters which are data from the JSON file of ISO/IEC27001:2013, data from the JSON file of ISO/IEC27002:2013, and a query from the search bar as shown in Listing 4.16. In this function, there are two important parts.

Listing 4.16: Overall in createDict function

```
/* Create a function to create an overall object */
function createDict(data, anx, txt) {
  /* Create objects for annex */
  var totalAnnex = []
  for(var a = 0; a < anx.length; a++) {
    ...
  }
  /* Create objects for clause */
  var nodeList = []
  for(var i = 0; i < data.length; i++) {
    ...
  }
}
```

The first part is creating an object from ISO/IEC27002:2013 that is received from a parameter. The purpose of the code shown in Listing 4.17 is to insert the object to be an element in each object of ISO/IEC27001:2013, to be able to highlight annexes from their name, and to add annexes' names into an auto-complete word list in order to do *auto-complete function*. It starts with looping all elements in the ISO/IEC27002:2013 object and loops all inner elements until finding the necessary data and creates an object to keep these data. The object that is created consists of name, number, and reference of the annex. In the end, the object will be appended into a list to make it easy to be used in the future.

Listing 4.17: Create Annex Object

```
/* Create objects for annex */
var totalAnnex = []
for(var a = 0; a < anx.length; a++) {
  var sTopic = anx[a].subtopic
  for(var b = 0; b < sTopic.length; b++) {
    var ssTopic = sTopic[b].subtopic
    for(var c = 0; c < ssTopic.length; c++) {
      /* Push obj of annexs to the array */
      totalAnnex.push({
        name: ssTopic[c].name,
        annex: 'A.' + ssTopic[c].annex,
        ref: ssTopic[c].page
      })
    }
  }
}
```

The second part is creating an object from ISO/IEC27001:2013 that is received from a parameter. The purpose of the code shown in Listing 4.18 is to create an object to return back to create the visualization. It starts with looping all elements in ISO/IEC27001:2013 and gathers all necessary data the same as in the previous part. However, since this object has to be used for creating the visualization by D3.js, three keys are required for D3.js which are *name* (name of the node), *children* (list of descendants of the node) and *value* (to specify the size of the node, it is set to be 1's in the code). Besides those three keys, there are four other necessary keys which are *color*, *highlight*, *annex*, and *annexObj*. The *color* is used to set the color of each node by the color of the clause node. The *highlight* is used to emphasize the nodes that have the query contained in its name. It is done by receiving the query from a parameter and creates the *setHighlight* function shown in Listing 4.19 that will return true if there is the query in the word. The *annex* is used to tell whether this node is an annex node in order to do some functions in the visualization function. The *annexObj* is used to insert the annex data within clause data to provide information for creating a pop-up window.

Listing 4.18: Create clause object

```
var nodeList = []
for(var i = 0; i < data.length; i++) {
  var name = data[i].name
  var number = data[i].clause
  var tname = number + '' + name
  var child = []
  var inside = data[i].content
  /* Append the name into the autocomplete list */
  if(!wordList.includes(tname))
    wordList.push(tname)

  for(var j = 0; j < inside.length; j++) {
    var sname = inside[j].sname
    var snumber = inside[j].sclause
    var stname = snumber + '' + sname
    var schild = []
    var inside2 = data[i].content[j].annex
    /* Append the name into the autocomplete list */
    if(!wordList.includes(stname))
      wordList.push(stname)
    for(var k = 0; k < inside2.length; k++) {
      var aname = inside2[k]
      /* Append the name into the autocomplete list */
      if(!wordList.includes(ename))
        wordList.push(ename)
      var aObject
      for(var p = 0; p < totalAnnex.length; p++) {
```

```
if(totalAnnex[p].annex == aname) {  
    aObject = totalAnnex[p]  
    if(!wordList.includes(totalAnnex[p].name))  
        wordList.push(totalAnnex[p].name)  
    break  
}  
}  
  
/* Can search from annex's name or annex number */  
highlight = setHighlight(aObject.name) || setHighlight(cname)  
schild.push({  
    name: cname,  
    children: [],  
    value: 1,  
    color: data[i].clause,  
    highlight: highlight,  
    annex: true,  
    annexObj: aObject  
})  
}  
}  
highlight = setHighlight(stname)  
child.push({  
    name: stname,  
    children: schild,  
    value: 1,  
    color: data[i].clause,  
    highlight: highlight,  
    annex: false,  
    annexObj: null  
})  
}  
}  
highlight = setHighlight(tname)  
nodeList.push({  
    name: tname,  
    children: child,  
    value: 1,  
    color: data[i].clause,  
    highlight: highlight,  
    annex: false,  
    annexObj: null  
})  
}  
}  
var node = {  
    name: 'ISO20001',  
    children: nodeList,  
    value: 1,  
    color: 1,  
    highlight: false,  
    annex: false,  
    annexObj: null  
}  
  
/* Return node to create a visualization */  
return node
```

Listing 4.19: The function for set highlight value in the clause object

```
/* For set highlight data in each node */
```

```
function setHighlight(t) {  
if(t.toLowerCase().includes(txt) && txt != "")  
return true  
return false  
}
```

In the second function, *visualizing function*, it receives two parameters which are data from the JSON file of ISO/IEC 27001:2013 and a node object that is created from the previous function. As shown in Listing 4.20, there are five important parts in this function.

Listing 4.20: Overall visualization function

```
function visualize(data, node) {  
  
...  
const pack = data => d3.pack()  
  
...  
const svg = d3.select(main).append('svg')  
  
...  
const vnode = svg.append('g')  
.selectAll('circle')  
  
...  
const label = svg.append('g')  
  
...  
function zoomTo(v) {  
...  
}  
function zoom(d) {  
...  
}
```

The first part is shown in Listing 4.21. It works by converting the object from the previous function into a pack by using the *d3.pack* function. The first thing is converting the object into a hierarchy format by using the *d3.hierarchy function*, and setting the value of a parent node to be equal to the sum of all of their children nodes. Therefore, the size of the nodes will depend on the amount of their children.

Listing 4.21: Converting to d3.pack

```
const pack = data => d3.pack()  
.size([width, height])  
.padding(3)(d3.hierarchy(data))
```

```
.sum(d => d.value)
.sort((a, b) => b.value - a.value))
```

The second part shown in Listing 4.22 is creating the main container. In this part, ‘svg’ tag is used to create the visualization because it is easy to create any shape in HTML.

Listing 4.22: Creating the main container

```
/* Append all of the visualization to the container in <body> */
const svg = d3.select(main).append('svg')
  .attr('viewBox', `-$ {width/2} -$ {height/2} ${width} ${height}`)
  .style('display', 'block')
  .style('margin', '0 - 14px')
  .style('background', bgColor)
  .style('cursor', 'pointer')
  .on('click', function() {
    /* If click at the same point or click annex -> go back to all figure */
    return zoom(root)
  })
```

The third part is creating nodes from *d3.pack* together with their label as shown in Figure 4.10. In Listing 4.23, a ‘circle’ tag is used to represent each node. The color of each node depends on its clause and depth. Moreover, the stroke will be set according to the *highlight* data. If *highlight* is true, the stroke will be changed to yellow and become thicker as shown in Figure 4.11. On the other hand, if *highlight* is false, the stroke will happen only when there is a cursor hovering over it. In addition, this part has an onclick listener. If the node is not an annex, the visualization will zoom to that node, but if the node is an annex, a popup will be displayed as shown in Figure 4.12. Popups are created from an ISO/IEC 27002:2013 object that is embedded in the object of ISO/IEC27001:2013 from the *createDict function*.

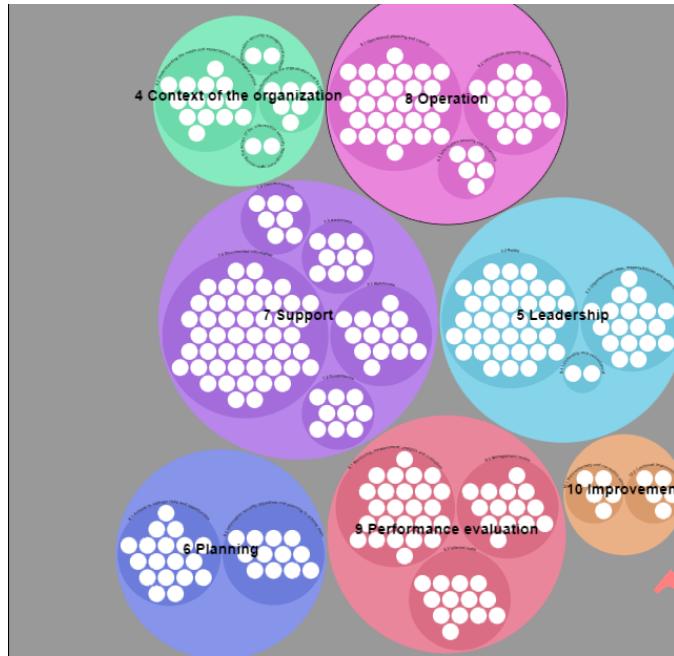


Figure 4.10: The nodes in the Zoomable Circle Packing visualization SAGE2 application.

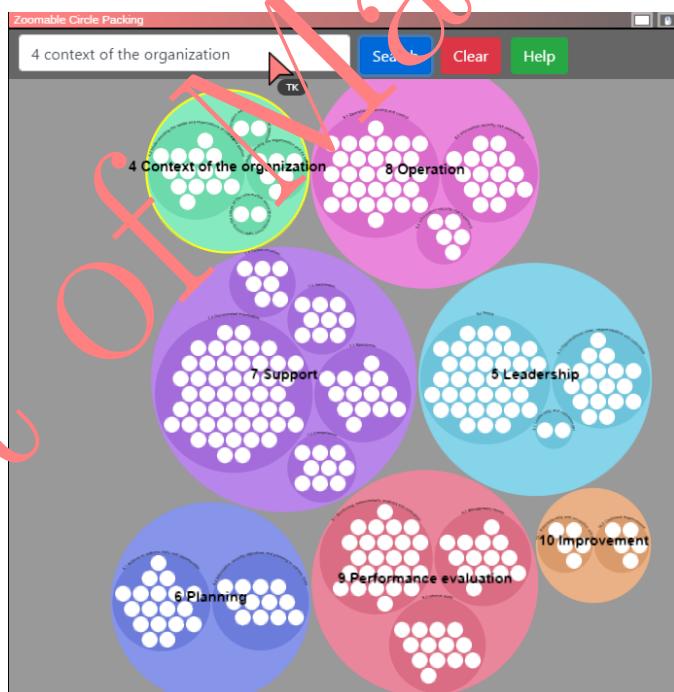


Figure 4.11: Clause 4, Context of the organization, highlighted with yellow stroke as it is matched with search query.

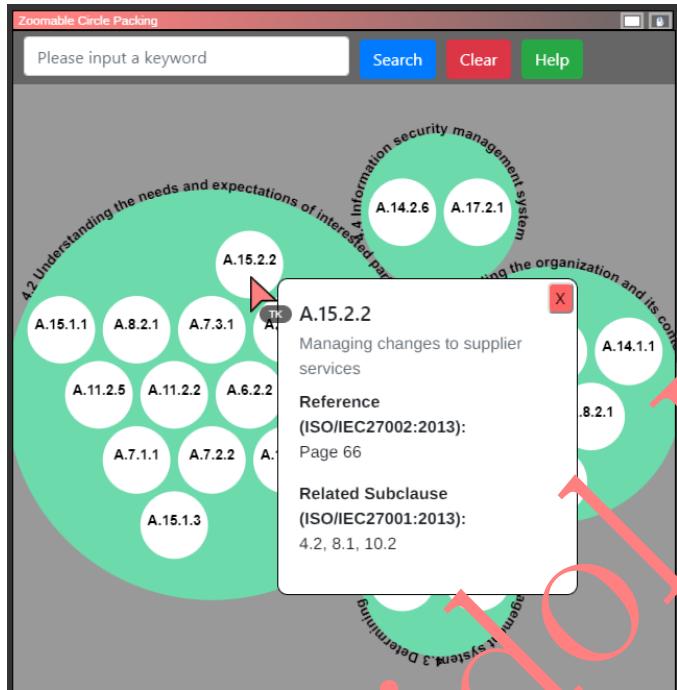


Figure 4.12: The pop-up window that contains information about Annex 15.2.2 in ISO:IEC 27002:2013 and its references.

Listing 4.23: Creating nodes (circles)

```

/* Create a circle (node) from each descendant */
const vnode = svg.append('g')
  .selectAll('circle')
  .data(root.descendants().slice(1))
  .join('circle')
  .attr('fill', d => d.children ? `#${hVal(d.data.color)}, ${sVal(d.depth)}%, ${lVal(d.depth)}%` : 'white')
  .attr('stroke', d => d.data.highlight ? highColor : null)
  .attr('stroke-width', d => !d.data.highlight ? 1 : 3)
  .on('mouseover', function() {
    d3.select(this).attr('stroke', '#000')
  })
  .on('mouseout', function(d) {
    if(!d.data.highlight)
      d3.select(this).attr('stroke', null)
    else {
      d3.select(this).attr('stroke', highColor)
      d3.select(this).attr('stroke-width', 3)
    }
  })
  .on('click', function(d) {
    if(d.data.annex == true) {
      /* Set the size of the popup */
      const wBox = 200
      const hBox = 250
      /* Set the position of the popup */
      const rWidth = window.innerWidth/2 - wBox/2
      const rHeight = window.innerHeight/2 - hBox/2
      if(document.getElementById(d.data.name) == null) {
        var pTop = d3.event.clientY
        var pLeft = d3.event.clientX
        /* Create the popup title */
        var title = document.createElement('div')
        title.id = d.data.name
        title.innerHTML = d.data.name
        title.style.position = 'absolute'
        title.style.left = pLeft + 'px'
        title.style.top = pTop + 'px'
        title.style.backgroundColor = 'white'
        title.style.border = '1px solid black'
        title.style.padding = '5px'
        title.style.fontSize = '14px'
        title.style.fontWeight = 'bold'
        title.style.cursor = 'pointer'
        title.addEventListener('click', function() {
          d3.select(this).remove()
        })
        document.body.appendChild(title)
        /* Create the popup content */
        var content = document.createElement('div')
        content.id = d.data.name + '_content'
        content.style.position = 'absolute'
        content.style.left = pLeft + 'px'
        content.style.top = pTop + 'px'
        content.style.backgroundColor = 'white'
        content.style.border = '1px solid black'
        content.style.padding = '10px'
        content.style.fontSize = '12px'
        content.style.lineHeight = '1.5em'
        content.innerHTML = 'This is a sample content for the selected node.'
        document.body.appendChild(content)
      }
    }
  })
}

```

```
var cTopic = document.createElement('h3')
cTopic.innerHTML = d.data.name
cTopic.setAttribute('style', 'margin: 5px')
/* Create the popup close button */
var cButton = document.createElement('button')
cButton.setAttribute('class', 'cButton')
cButton.innerHTML = 'X'
cButton.id = 'close_' + d.data.name
/* Create the detail in the popup */
var cDetail = document.createElement('p')
cDetail.innerHTML = '<b>Name: </b>' + d.data.annexObj.name
cDetail.setAttribute('style', 'margin: 5px')
var cRef = document.createElement('p')
cRef.innerHTML = '<b>Page: </b>' + d.data.annexObj.ref
cRef.setAttribute('style', 'margin: 5px')
var cContainer = document.createElement('span')
cContainer.setAttribute('class', 'drag')
cContainer.setAttribute('style', 'top:' + pTop + 'px;left:' + pLeft + 'px;')
cContainer.id = d.data.name
cContainer.appendChild(cTopic)
cContainer.appendChild(cButton)
cContainer.appendChild(cDetail)
cContainer.appendChild(cRef)
var validate = body.appendChild(cContainer)
$('.drag').draggable()
$('.drag').css('width', '${wBox}px')
$('.drag').css('height', '${hBox}px')
$('.drag').css('position', 'fixed')
$('.drag').css('border', '5px solid black')
$('.drag').css('background-color', 'white')
$('.cButton').css('background-color', 'rgb(255,100,100)')
$('.cButton').css('position', 'absolute')
$('.cButton').css('top', '0px')
$('.cButton').css('right', '0px')
$('.cButton').click(function() {
  var span = document.getElementById(this.id.slice(6))
  body.removeChild(span)
})
}
/* Do not zoom to the root if press annex */
d3.event.stopPropagation()
} else {
  focus !== d && (zoom(d), d3.event.stopPropagation());
}
})
```

The code shown in Listing 4.24 is to create the *zoom function* in order to do the onclick listener of the previous part. This part will receive a node and focus on it. The focus node will be zoomed in on and its children's label will appear instead of its label that will be faded.

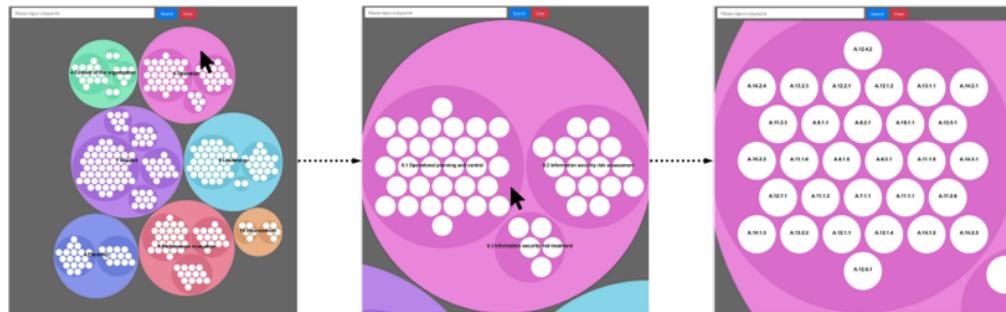


Figure 4.13: The transition in the Zoomable Circle Packing visualization when the user clicks on each node

Listing 4.24: A function to zoom in on a node

```
function zoom(d) {
  const focus0 = focus
  focus = d
  const transition = svg.transition()
  .duration(d3.event.altKey ? 7500 : 750)
  .tween('zoom', d => {
    const i = d3.interpolateZoom(view, [focus.x, focus.y, focus.r*2])
    return t => zoomTo(i(t))
  })
  label.filter(function(d) {
    return d.parent === focus || this.style.display === 'inline'
  })
  .transition(transition)
  .style('fill-opacity', d => d.parent === focus ? 1 : 0)
  .on('start', function(d) {
    if(d.parent === focus) this.style.display = 'inline'
  })
  .on('end', function(d) {
    if(d.parent !== focus) this.style.display = 'none'
  })
}

function zoomTo(v) {
  const k = width / v[2]
  view = v
  label.attr('transform', d => `translate(${(d.x - v[0]) * k}, ${(d.y - v[1]) * k})`)
  vnode.attr('transform', d => `translate(${(d.x - v[0]) * k}, ${(d.y - v[1]) * k})`)
  vnode.attr('r', d => d.r * k)
}
```

### 4.2.3 Web Application

The Web application version is the conversion of the SAGE2 application. Most of the functions remain the same because both versions are based on JavaScript language. The difference between both versions is the method that is used to read JSON and *pop-*

*up function.* In SAGE2, JSON is read by the SAGE2 built-in function, while in the web application, JQuery is used for reading JSON files as shown in Listing 4.25. For *pop-up function* as shown in Listing 4.26, in the web application, the frame is created in the same screen instead of creating a new window in the SAGE2 application as shown in Figure 4.14.

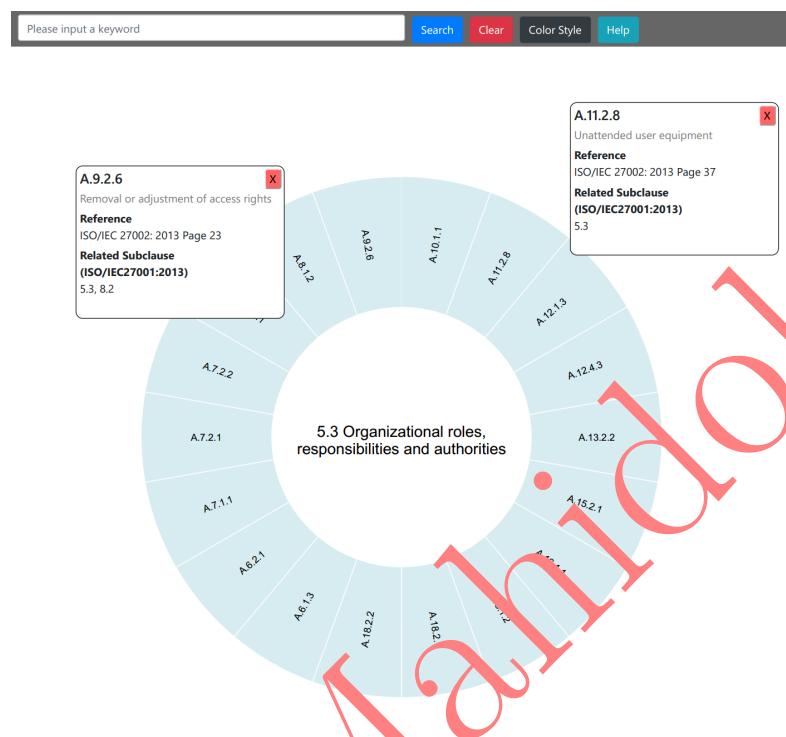


Figure 4.14: The pop-up window that contains information of Annex 9.2.6 and 11.2.8 in the Sunburst visualization on the web application.

Listing 4.25: Read JSON file of the ISO/IEC 27001:2013 and ISO/IEC 27002:2013 document (Web application version)

```
$getJSON('./ISO27002.json', data => {
  iso27002 = createDictionary27002(data);
  $getJSON('./ISO27001.json', data => {
    iso27001 = createDictionary27001(data);
    ....
  });
});
```

Listing 4.26: Popup function for each node in the visualization (Web application version)

```
var pTop = d3.event.clientY
var pLeft = d3.event.clientX
var annex = d.data.name.split(".");
var aname = iso27002.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1]
  .children[parseInt(annex[3])-1].name;
var page = iso27002.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1]
  .children[parseInt(annex[3])-1].page;
/* Create the popup title */
var cTopic = document.createElement('h5')
cTopic.innerHTML = d.data.name
cTopic.setAttribute('style', 'margin: 5px')

/* Create the popup close button */
var cButton = document.createElement('button')
cButton.setAttribute('class', 'cButton')
cButton.innerHTML = 'X'
cButton.id = 'close_' + d.data.name

/* Create the detail in the popup */
var cDetail = document.createElement('p')
cDetail.innerHTML = aname
cDetail.setAttribute('style', 'margin: 5px; color: gray')

var cRef = document.createElement('p')
cRef.innerHTML = "<b>Reference</b><br>ISO/IEC 27002: 2013 Page " + page;
cRef.setAttribute('style', 'margin: 5px')

var cContainer = document.createElement('span')
cContainer.setAttribute('class', 'drag')
cContainer.setAttribute('style', 'top:${pTop}px;left:${pLeft}px;')
cContainer.id = d.data.name
cContainer.appendChild(cTopic)
cContainer.appendChild(cButton)
cContainer.appendChild(cDetail)
cContainer.appendChild(cRef)

var validate = document.body.appendChild(d(cContainer))
$('.drag').draggable()
$('.drag').css('width', `${wBox}px`)
$('.drag').css('height', `${hBox}px`)
$('.drag').css('position', 'fixed')
$('.drag').css('border', '1px solid black')
$('.drag').css('border-radius', '10px')
$('.drag').css('background-color', 'white')

$('.cButton').css('background-color', 'rgb(255,100,100)')
$('.cButton').css('position', 'absolute')
$('.cButton').css('top', '3px')
$('.cButton').css('right', '3px')
$('.cButton').css('border-radius', '5px')
$('.cButton').click(function() {
  var span = document.getElementById(this.id.slice(6))
  document.body.removeChild(span)
})
```

## CHAPTER 5

### TESTING AND EVALUATION

#### 5.1 Unit Tests

In this project, there are 10 participants including 3 information security experts, 2 IT staff, and 5 ICT students. The applications used during tests are two visualizations – Sunburst and Zoomable Circle Packing visualizations, in website version since online testing is required due to the quarantine period in the COVID-19 situation. Files provided for tests include consent forms, informational slides, ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents, and for both ISO/IEC 27001:2013 and ISO/IEC 27002, testers were required to delete it right after finishing the test due to the licensing issue.

#### 5.2 Usability Test Procedure

As shown in Figure 5.1, before starting tests, testers were asked to complete a consent form attached in their email and a pre-test questionnaire on Google forms which asked for age, gender, and familiarity levels for each term that was frequently used in the project. Then, they got brief information about the project specifically the project's processes, objectives, background information, and test procedure. Next, they were informed about a manual guide for each visualization and tried practicing both visualizations. Consequently, testers would be asked to practice the visualizations following the provided questions in 'Question: Part A'. In part A, there are 2 sections of question which are sunburst visualization and zoomable circle packing visualization and 5 questions for each section. During this part, testers could call for hints whenever they needed. They, however, were unable to call for help during a test on 'Question: Part B' which contained 3 sections. Those are manual documents, sunburst visualization, and zoomable circle packing visualization where each section contains 5 questions. Lastly, testers completed the post-test form on Google Forms which asked for their satisfaction levels and opinions about the efficiency of the visualization applications to finish their

test.

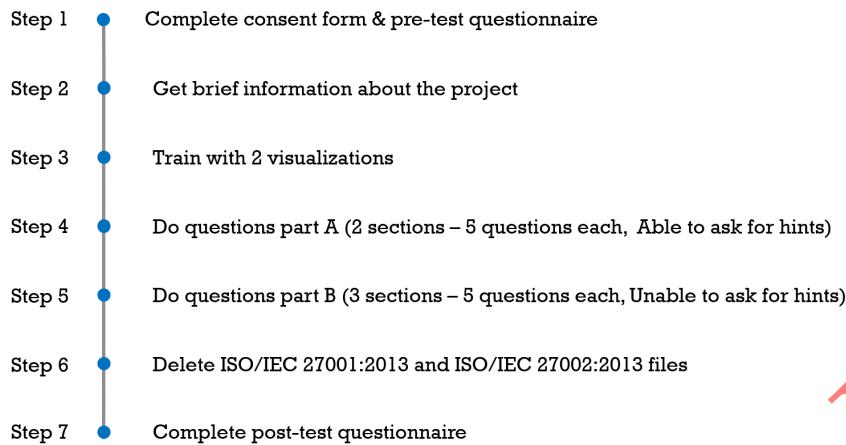


Figure 5.1: The procedure in usability testing.

For the training section, the following question in ‘Question : Part A’ was given to the participants for practicing on Sunburst and Zoomable Circle Packing visualization, respectively, and for the first three questions, participants were asked to find the work covered area for ISMS implementation and the rest of the questions were asked to find the specific terms in the standard documents.

#### **Training Question: [Sunburst Visualization]**

1. How many annexes related to sub clause 8.3?
2. How many annexes related to sub clause 5.1?
3. Please find at least 3 annexes related to sub clause 10.2
4. Please find sub clauses related to term ‘Documented Information’
5. Please find annexes related to the term ‘Policy’, their references in ISO/IEC 27002:2013 document, and their related subclauses.

#### **Training Question: [Zoomable Circle Packing Visualization]**

1. How many annexes related to sub clause 4.3?
2. How many annexes related to sub clause 9.2?

3. Please find at least 3 annexes related to sub clause 4.4
4. Please find sub clauses related to term ‘Corrective action’
5. Please find annex related to term ‘Use of assets’, their references in ISO/IEC 27002:2013 document, and their related subclauses

Then, in the testing section, participants received the following questions for finding answers through a document manually, using Sunburst visualization and using Zoomable Circle Packing Visualization, respectively. The first two questions were asked to find the working range of ISMS implementation namely asking about the amount of code of practices for a work area and the rest were asked to find about the given specific terms in a document.

#### **Testing Question: [Manual Document]**

1. How many sub annexes (A.x.x.x) are included in Annex 9?
2. How many sub annexes (A.x.x.x) are included in Annex 5?
3. Please find sub clauses related to term ‘Information security risk assessment’
4. Please find sub clauses related to term ‘planning and control’
5. Please find annexes related to the term ‘Information security roles’, and their references in ISO/IEC 27002:2013 document.

#### **Testing Question: [Sunburst Visualization]**

1. How many sub annexes (A.x.x.x) are included in Annex 7?
2. How many sub annexes (A.x.x.x) are included in Annex 11?
3. Please find sub clauses related to term ‘Information Security Risk Treatment’
4. Please find sub clauses related to term ‘organizational role’
5. Please find annexes related to the term ‘Segregation of duties’, their references in ISO/IEC 27002:2013 document, and their related subclauses.

**Testing Question: [Zoomable Circle Packing Visualization]**

1. How many sub annexes (A.x.x.x) included in Annex 15?
2. How many sub annexes (A.x.x.x) included in Annex 13?
3. Please find sub clauses related to term ‘Leadership and commitment’
4. Please find sub clauses related to term ‘management review’
5. Please find annexes related to the term ‘Terms and conditions of employment’, their references in ISO/IEC 27002:2013 document, and their related subclauses.

### 5.3 Evaluation

Figure 5.2 and Figure 5.3 represent the average time of each question during the training and testing sections of usability tests, respectively. For the questions in the test section, the first and second questions asked users to find the amount of sub clauses or sub annexes as a scope for work while the other questions asked for finding the clause or annex related to the given term to test the efficiency of the search feature of both visualizations. As a result, Figure 5.3 shows that the average time decreases greatly when users use the visualizations to find an answer compared with searching for the given terms manually in a document file. Also, according to figures, it can be determined that the Sunburst visualization is more complex than the Zoomable Circle Packing visualization since users need to spend more time using Sunburst to find an answer to each question than using Zoomable Circle Packing.

Figure 5.4, Figure 5.5 and Figure 5.6 represent the average time on each question during the test section separated by 3 testing types including Manual Document, Sunburst Visualization and Zoomable Circle Packing Visualization, respectively.

Figure 5.7 represents the correctness of each tester type in each test section namely Manual Document, Sunburst Visualization, and Zoomable Circle Packing Visualization, respectively. Figure 5.7 shows that although the experts can get an answer precisely in every test section, the other testers are getting more accurate answers when they used the visualizations. Accordingly, it implies that with both visualizations, users can scope

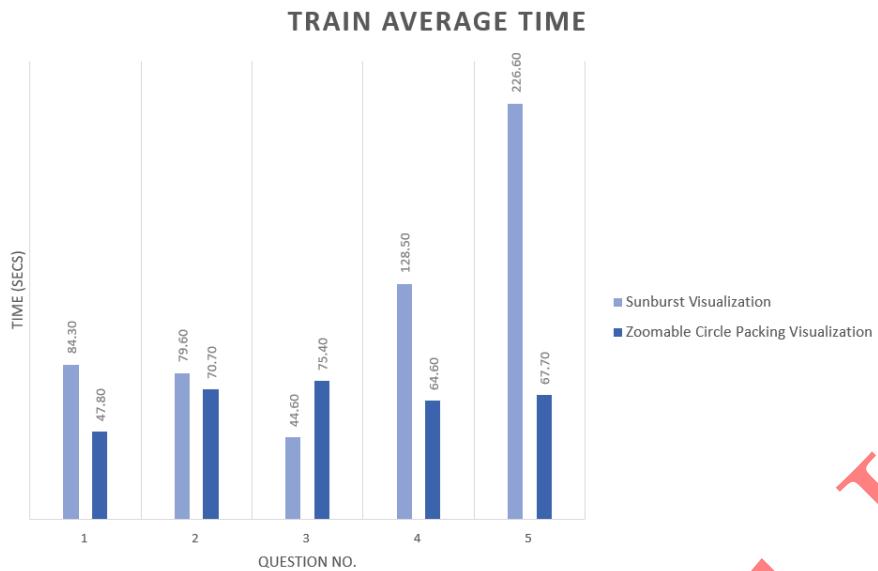


Figure 5.2: Average time - train question

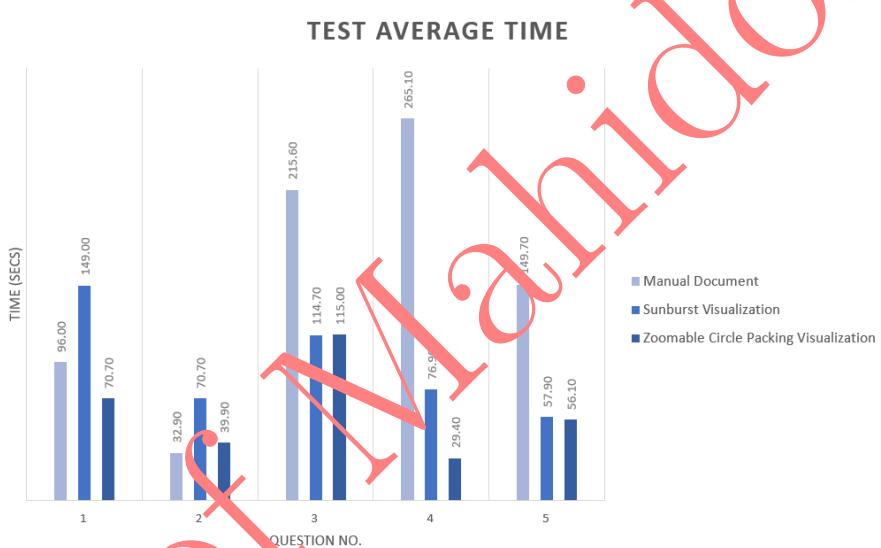


Figure 5.3: Average time - test question

the area of work faster and more accurately than implementing the information security management system (ISMS) by using the documents manually.

After finishing the test sections, participants were asked to complete the post-test form that asked their opinion and satisfaction about the project. As a result, Figure 5.8 shows the effectiveness levels of each visualization and SAGE2 application according to users' opinions. Users thought both visualizations are helpful for an ISMS implementation; however, due to the COVID-19 situation, the SAGE2 application was unable to

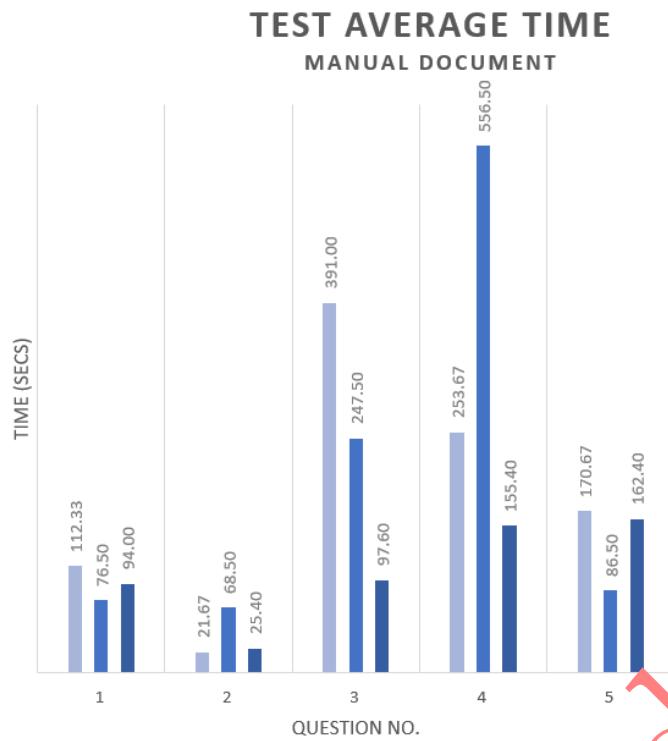


Figure 5.4: Average time - test question section 1: Manual document



Figure 5.5: Average time - test question section 2: Sunburst visualization

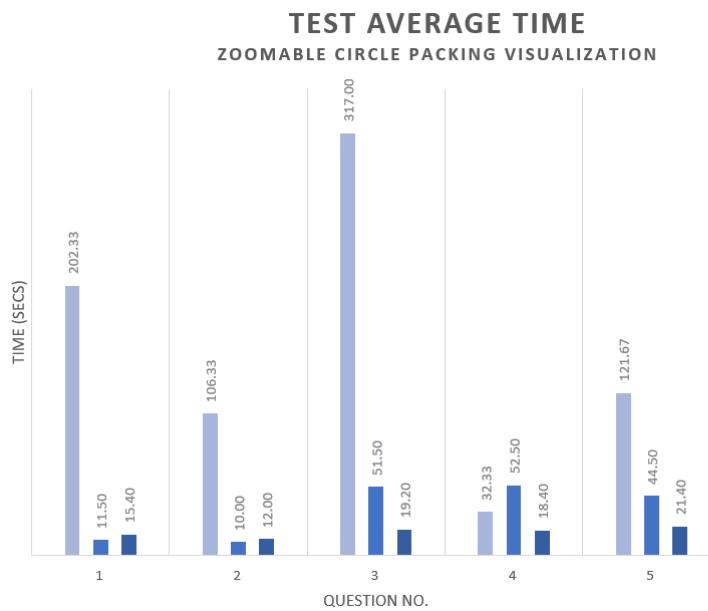


Figure 5.6: Average time - test question section 3: Zoomable circle packing visualization

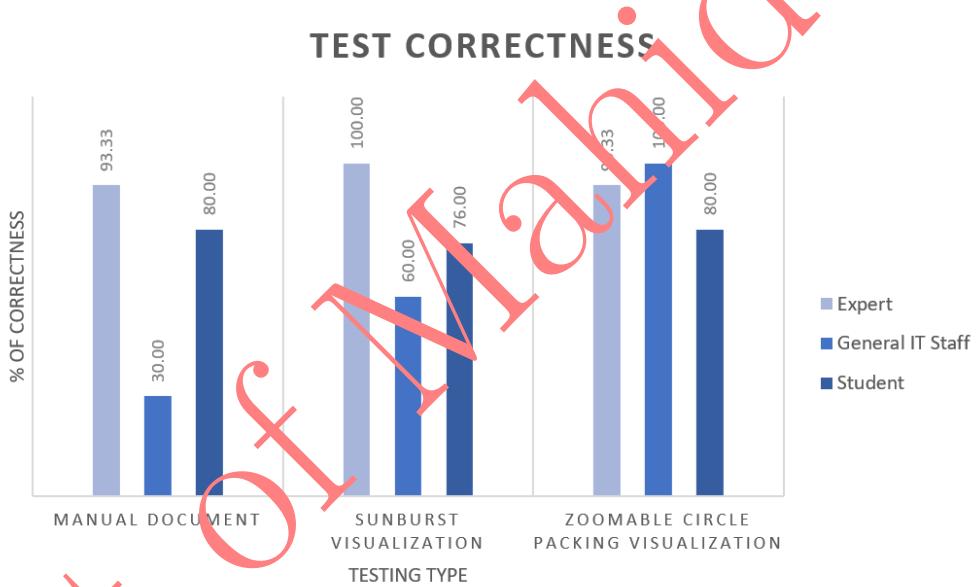


Figure 5.7: The correctness from the test separated by tester type

launch for the usability test since it requires the specific environment to test efficiently, so users only knew how it launched but did not use it for the tests.

Moreover, with Figure 5.9, it shows that users are quite satisfied using both visualizations. To be specific, users slightly preferred the Sunburst visualization over the Zoomable Circle Packing visualization according to their satisfaction score, and they are

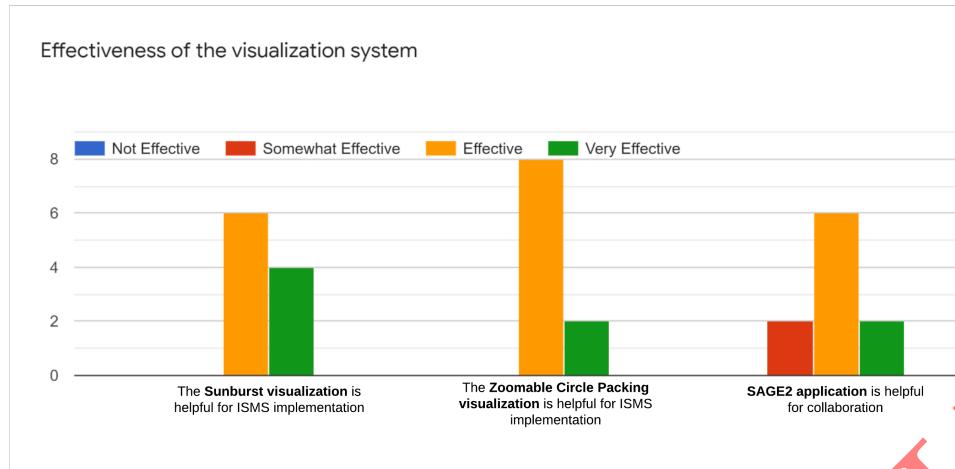


Figure 5.8: Effectiveness of the visualization from tester's opinion

more impressed with the Search feature than the Pop-Up feature. Meanwhile, without using the SAGE2 application on the test, the user satisfaction with the application is varied.

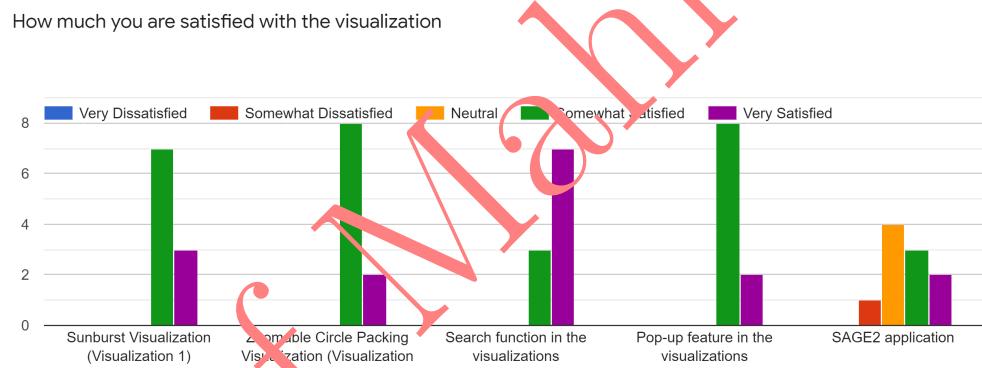


Figure 5.9: The user satisfaction for the visualization

## CHAPTER 6

### IMPROVEMENT

From the feedback in the usability test, we have improved our application as follows:

#### 6.1 Sunburst Visualization

The first improvement for the Sunburst visualization is the color of the visualization. According to the usability test, some test subjects suffered from color blindness which makes it hard to see the difference between each clause in the Sunburst visualization. The test version can be seen in Figure 6.1. In order to solve this problem, the Greyscale color is adopted so the color blind person could see the difference for each clause in the visualization. The example of the greyscale version can be seen in Figure 6.2. The method to use the greyscale version is changing the color saturation to be 0 percent which means color with shades of grey and assigning each clause for lightness as shown in Listing 6.1.

Listing 6.1: Change the salutation and lightness color to be greyscale

```
if(document.getElementById('color').getAttribute('class') == 'btn btn-warning'){
    sVal = 0;
    lVal = d3.scaleLinear()
        .domain([0, 7])
        .range([0, 90]);
}
```

Moreover, the user manual function also has been added into the application as well. The method to open the user manual is clicking the help button. As shown in Figure 6.3, the help button position is next to the color style button. When the help button is clicked, the application will open the user manual as a new application on SAGE2 as shown in Figure 6.4, while the user manual will be a pop-up window in the web application version. There are a total of three pages of the user manual for the Sunburst visualization.

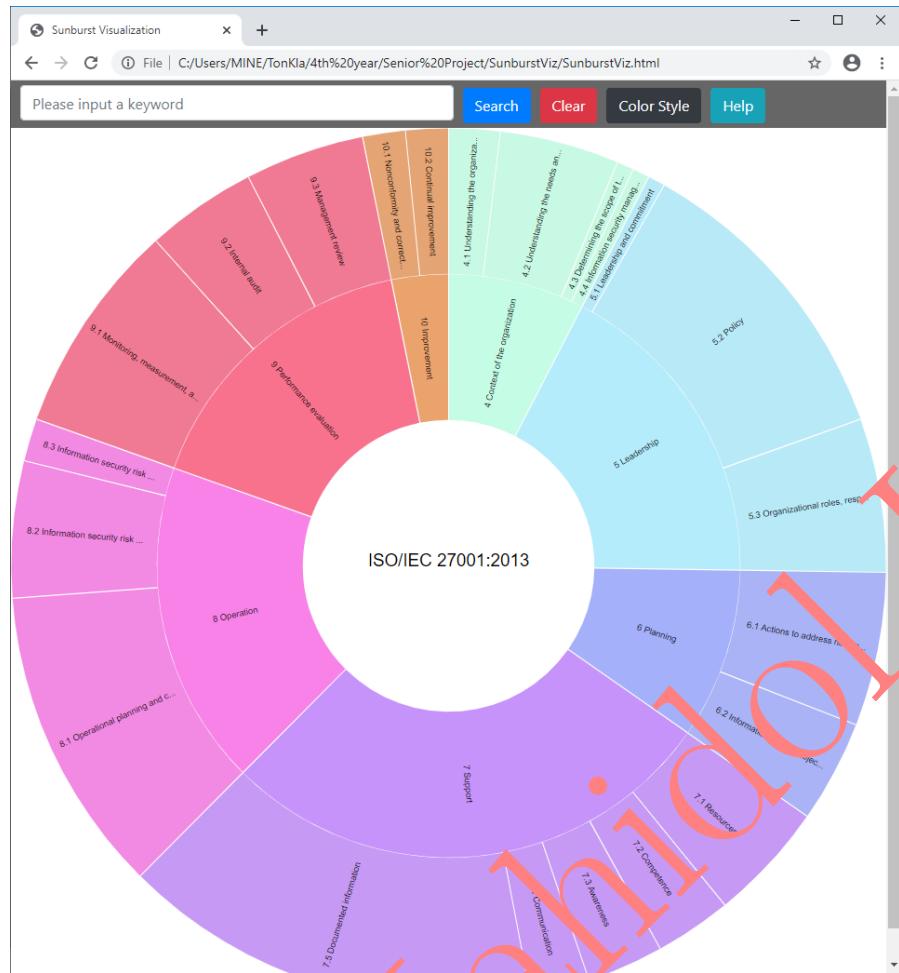


Figure 6.1: The color version of the Sunburst visualization

Lastly, the pop-up window that contains the reference to ISO/IEC 27002:2013 has been improved as well. First, the design of the pop-up has been changed and also the information of related other subclauses has been added to the pop-up. The example of this change can be seen in Figure 6.5.

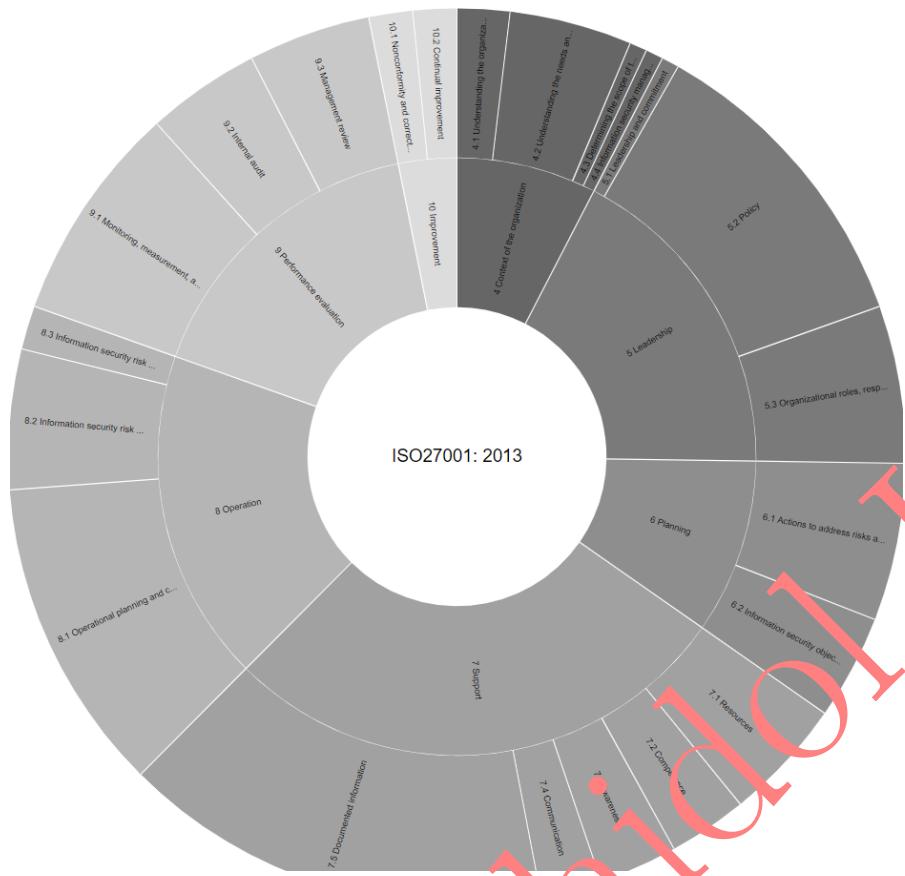


Figure 6.2: The greyscale version of the Sunburst visualization



Figure 6.3: The help button next to the blind color button in the Sunburst visualization

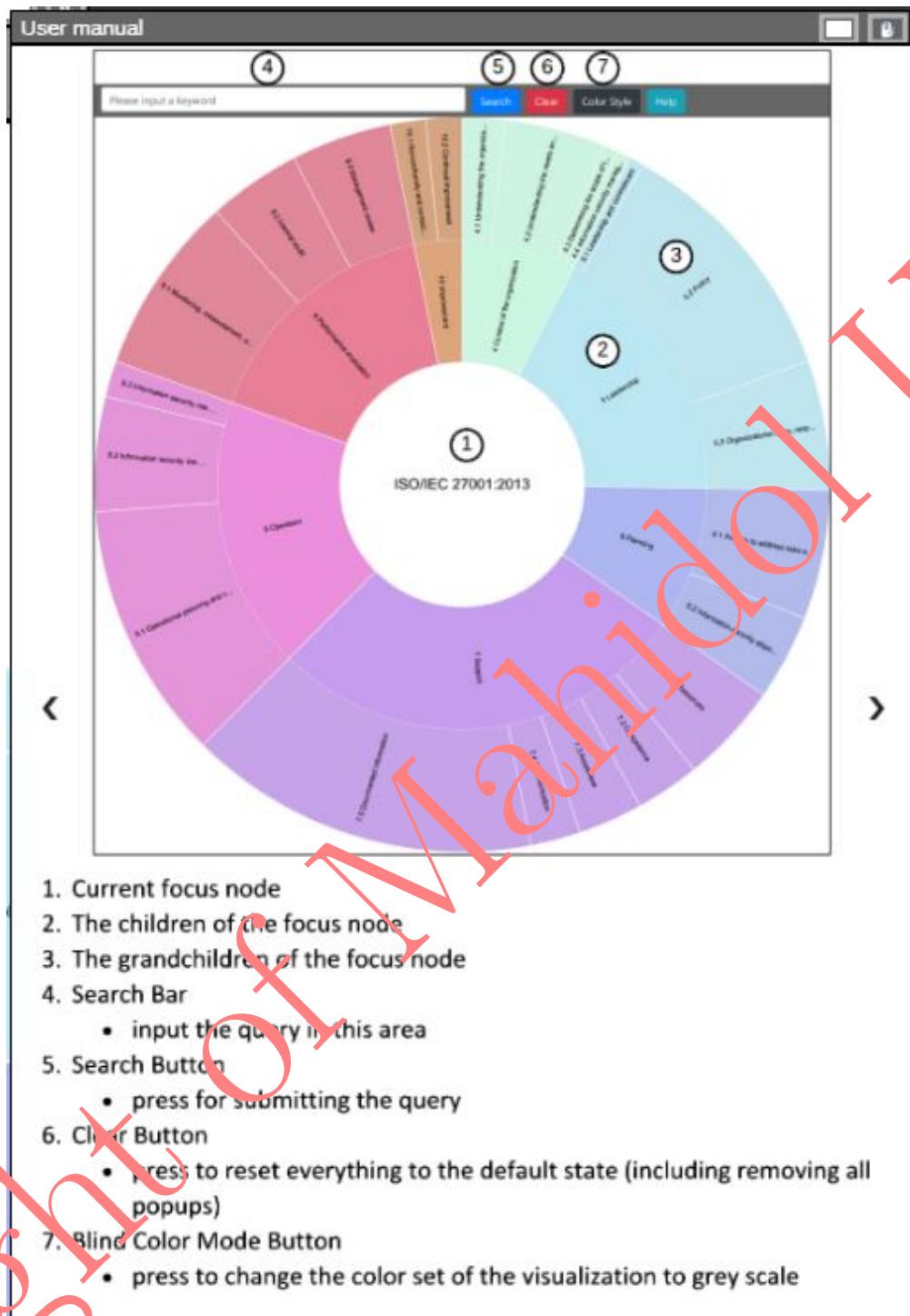


Figure 6.4: The user manual of the Sunburst visualization on the SAGE2 application

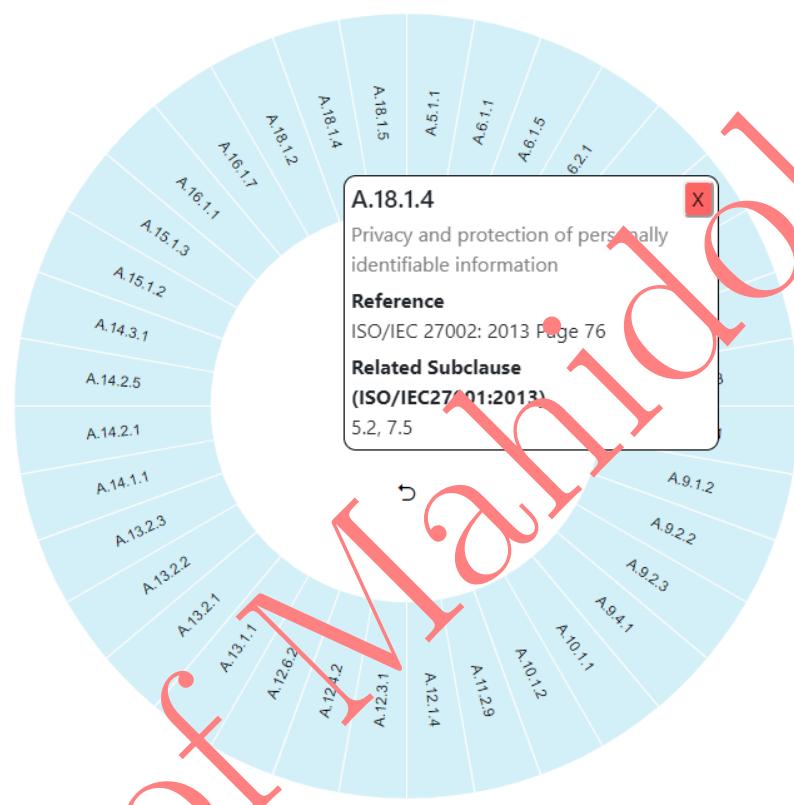


Figure 6.5: The pop-up window with new design and its related subclauses

## 6.2 Zoomable Circle Packing Visualization

The first improvement of Zoomable Circle Packing Visualization is the text display. In the previous version of the visualization, the text of nodes will appear when its parent node is focused upon. However, after doing usability testing, there are some recommendations from test subjects about showing the text in a deeper layer since it is easier to go to the specific part without going to each layer. There are many styles that were implemented in order to do this. The first version of the implementation is displaying the next two layers as in Figure 6.6 and Figure 6.7.

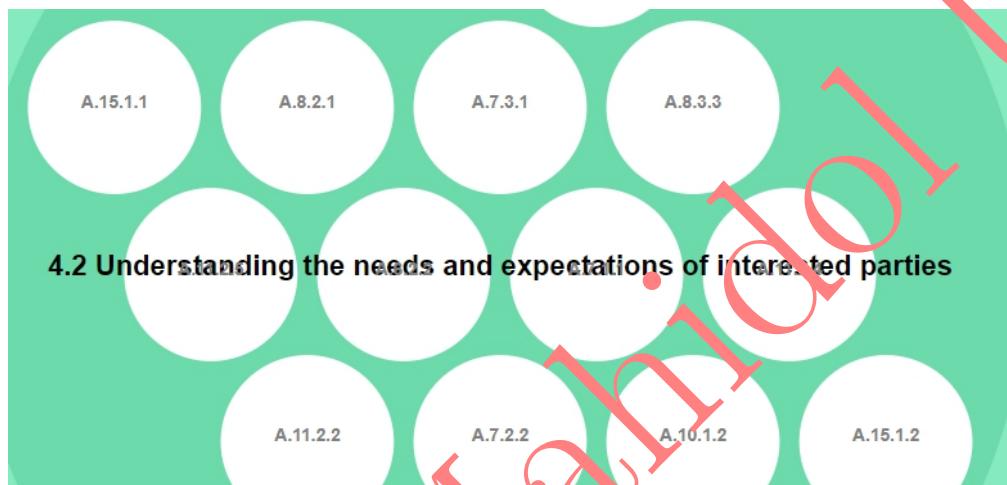


Figure 6.6: The text of the next two layers will appear after zooming to a node

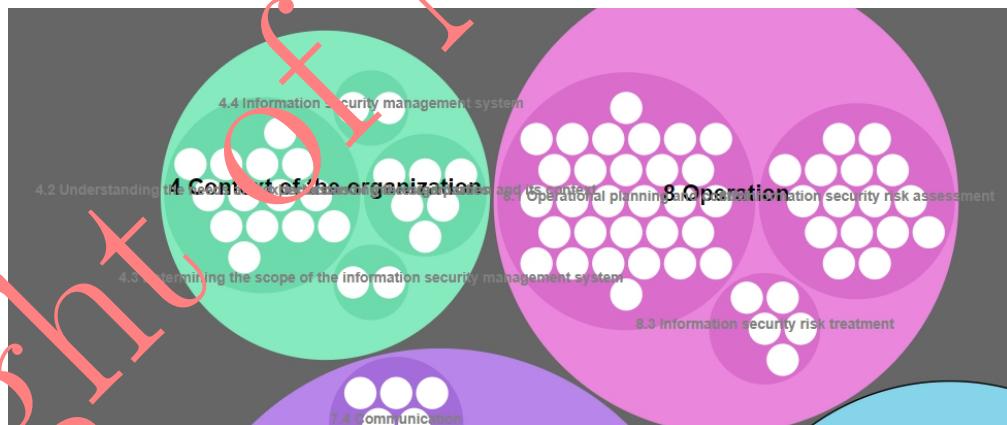


Figure 6.7: The text of the upper layer is overlapped with the text of the lower layer

According to Figure 6.7, there are some problems with this implementation. First, the size of the circle of the next layer is too small compared with the text length, so

it will lead to a lack of legibility. Second, the text of the upper layer is overlapped with the lower layer.

The first problem is solved by cutting the text into a length that does not exceed the circle and use ‘...’ for the leftover text as in Figure 6.8. Another problem is solved by shifting the text of the upper layer into the top, so it will not be overlapped by the text of the lower layer as in Figure 6.9.

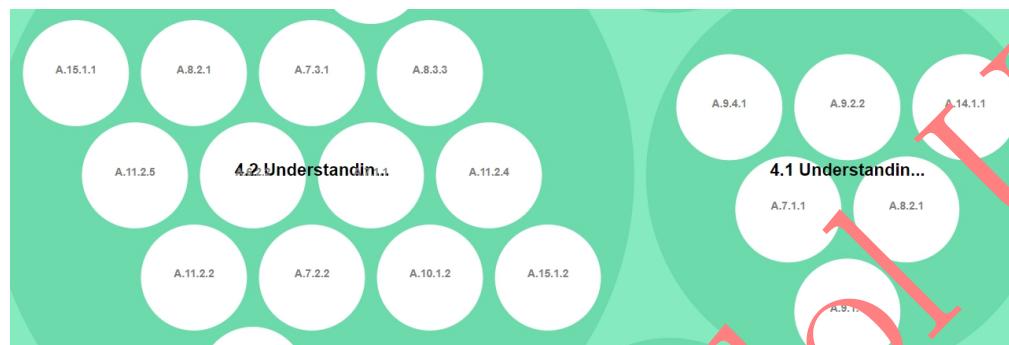


Figure 6.8: The text is cut in order to fit within the circle size

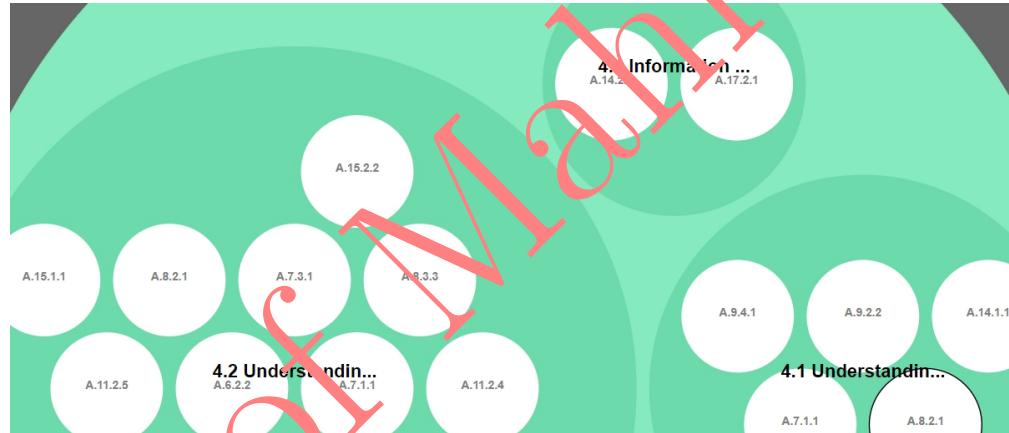


Figure 6.9: Shifting the text of the upper layer to the top of the text of the lower layer

However, after receiving advice from the advisors, the text cannot be cut because it will be contrary to the purpose of the visualization that is to make it easier for users to understand the standard. For example, according to Figure 6.9, subclause 4.1 and subclause 4.2 have the same name that may be confusing; therefore, another style will be used instead.

Since the text does not have to be cut, the text is separated into many lines instead as in Figure 6.10, and it is shifted to avoid overlapping.

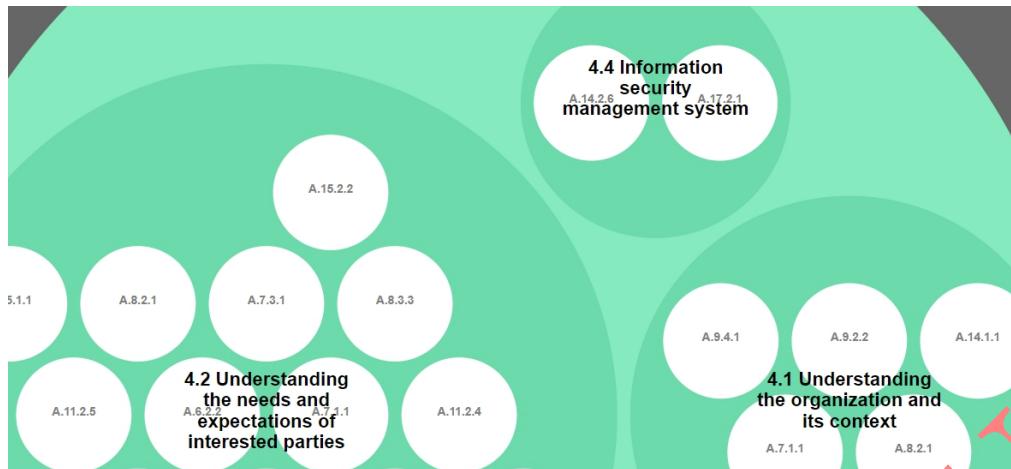


Figure 6.10: The text is separated into several lines

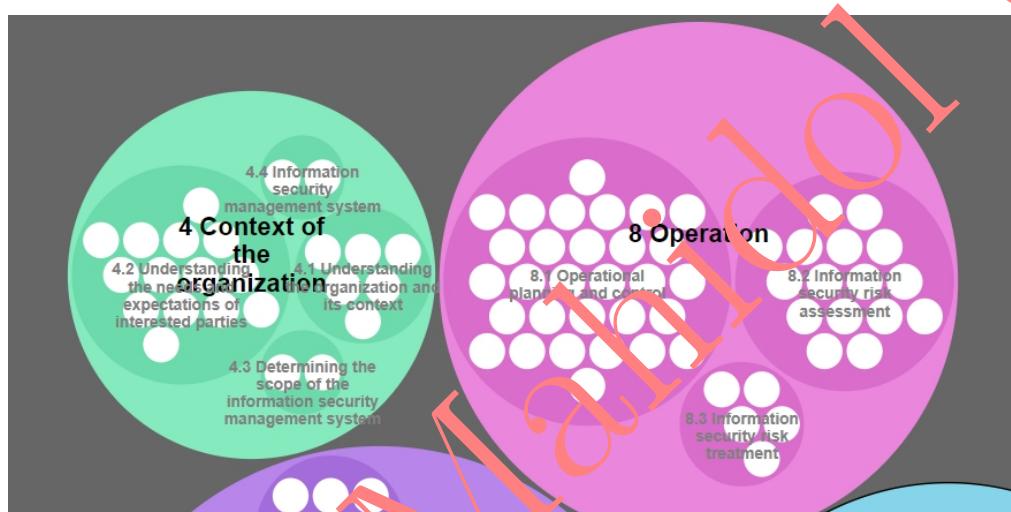


Figure 6.11: The visualization of text separation that does not zoom to any layer

On the other hand, this style brings the previous problems back which are that the size of circles are too small as compared with the text, and overlapping of the text that you can see in Figure 6.11. After facing this problem, it can be concluded that the text cannot be in the center of the circle because of limited space. The next style that is used to handle these problems is moving the text to the top of the circle as in Figure 6.12 and Figure 6.13.

After using this style, the advisors gave some advice about a lot of space and the text that can be seen on only one level. Consequently, it is fixed by showing the text of the next two layers, but it is not readable as in Figure 6.14, and was consequently fixed in Figure 6.15.

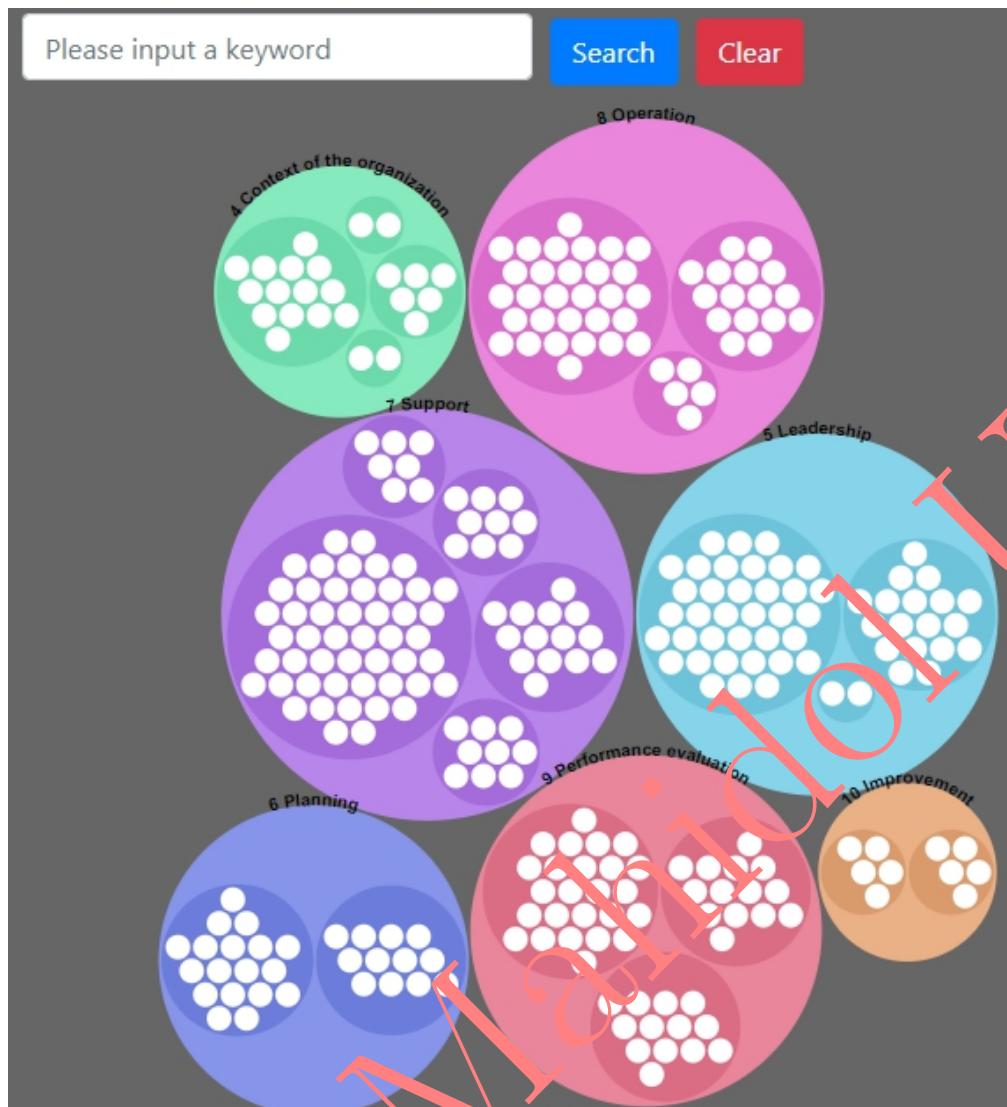


Figure 6.12: The text is moved to the top of the circles



Figure 6.13: The text is moved to the top of the circles after zooming one level

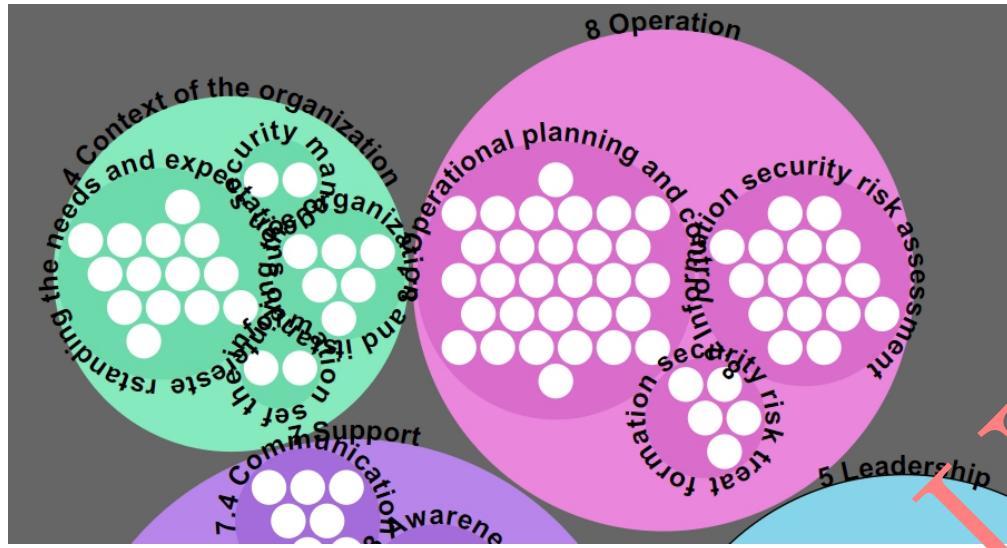


Figure 6.14: Displaying the text of the next two layers

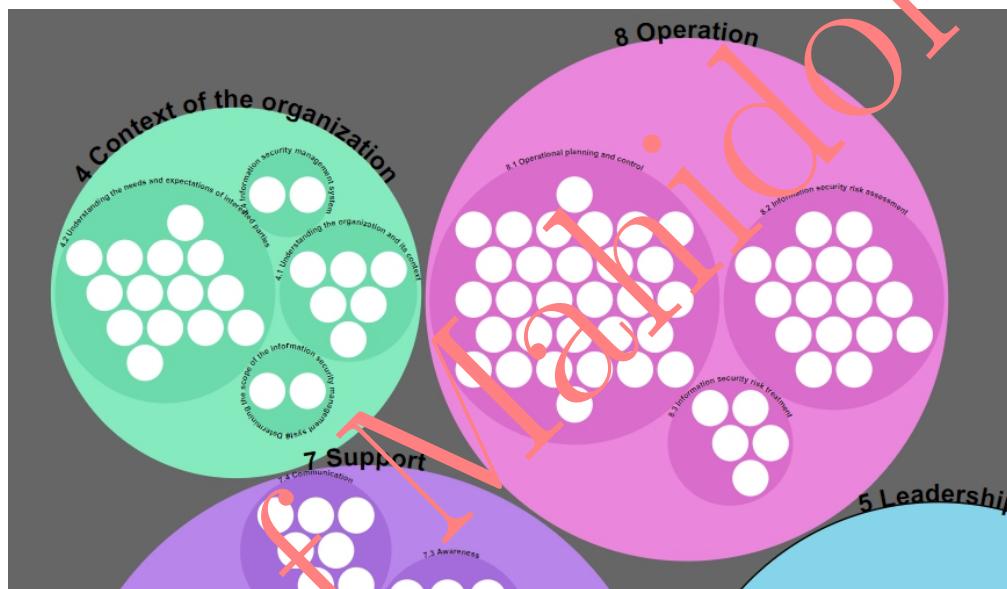


Figure 6.15: The size of the text of the below level is decreased to make it be more readable

However, there were three items of feedback given after using this style. First, there is a lot of space within the circles. Second, the background is rather dark which leads to unclear text of the clause. Third, the text of the subclause is too small. As a result, it is solved by changing the background color to be lighter, moving the text of the clause to the center of the circle and increasing the text size of the subclause when zooming to that level. The final version of the visualization is demonstrated in Figure 6.16 and Figure 6.17.

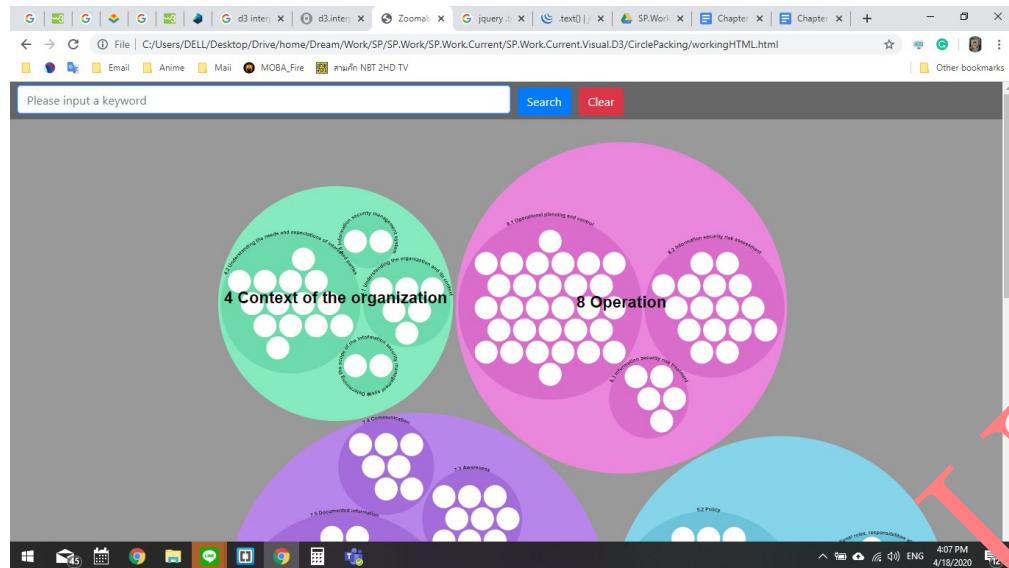


Figure 6.16: The text of the clause is moved to the center of the circle

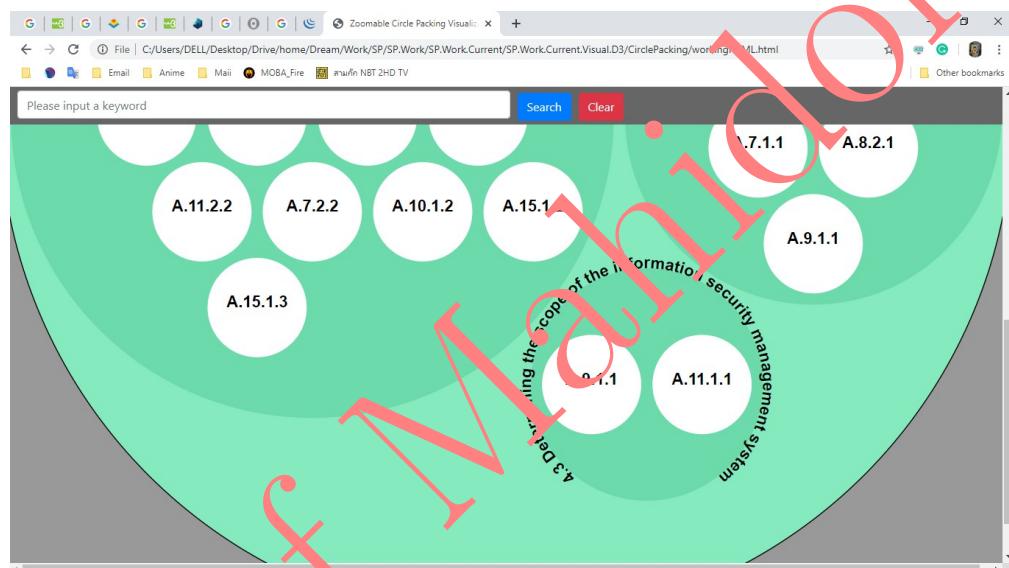


Figure 6.17: The text size of the subclause is increased when zooming

Moreover, after doing usability testing, the subjects suggested improvement of two things. First, in the popup, it should add the related subclause in order to make it easier to find the relationship as in Figure 6.18. Another is that it should have a manual for helping to guide users using the program as in Figure 6.19.

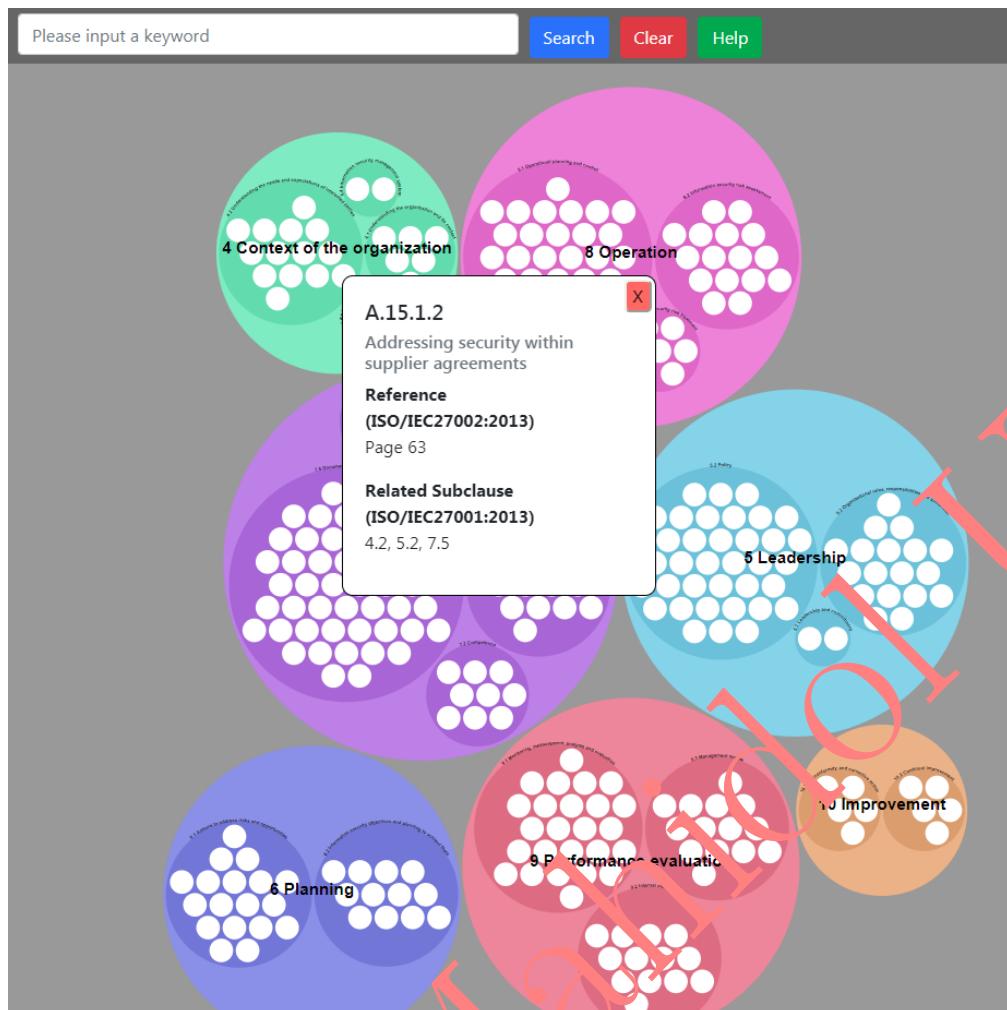


Figure 6.18: Add the related subclauses into popups

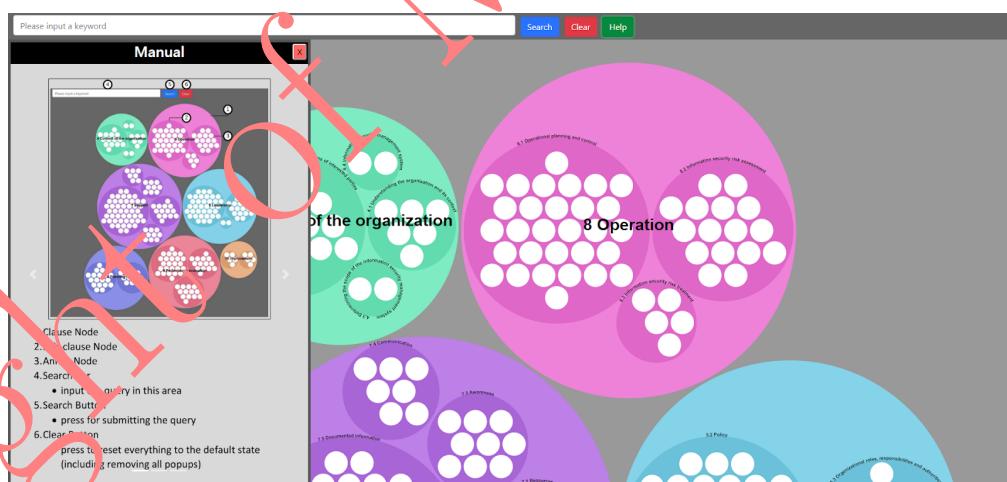


Figure 6.19: Add help button and manual to help users using the program

### 6.3 Natural Language Processing

Figure 6.20 shows the example of *idf scores* from the term-document matrix generated from the *Tfidfvectorizer function* from scikit-learn where the first column represents each corpus including subclauses in a blue color and annexes in a light blue color and the others represent each term with its idf score in each row. With the example, it can be determined that there are certain correlations between subclauses and annexes; the results, however, are still incorrect due to the range of a term which might be a keyword for a corpus. To illustrate this, ‘interested parties’ will be tokenized and lemmatized into the words ‘interest’ and ‘party’ separately, while it is needed to be together as a keyword. Currently, it is unable to tokenize and lemmatize a word by setting *ngram\_range* since a callable function is used in the ‘analyzer’ setting in the *TfidfVectorizer function* for using tokenization and lemmatization functions from Spacy.

	Terms																														
id	ability	able	able	able	able	able	able	able	able	able	able	able	able	able	able	able															
10.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4.1	0.2032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2376	0	0	0	0	0	0	0	0	0	0	0	0	0	
5.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0232	0	0	0	0	0	0	0	0	0	0	0		
6.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0326	0	0	0	0	0	0	0	0	0	0	0		
7.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0666	0	0	0	0	0	0	0	0	0	0	0		
8.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0485	0	0	0	0	0	0	0	0	0	0	0		
8.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0231	0	0	0	0	0	0	0	0	0	0	0		
8.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0643	0	0	0	0	0	0	0	0	0	0		
8.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0495	0	0	0	0	0	0	0	0	0	0	0		
8.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1259	0	0	0	0	0	0	0	0	0	0		
8.10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0531	0	0	0	0	0	0	0	0	0	0	
8.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0204	0	0	0	0	0	0	0	0	0	0	0	
8.12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0854	0	0	0	0	0	0	0	0	0	0	0	
8.13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0535	0	0	0	0	0	0	0	0	0	0	0	
8.14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0319	0	0	0	0	0	0	0	0	0	0	0	
8.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0236	0	0	0	0	0	0	0	0	0	0	0	
8.16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0387	0	0	0	0	0	0	0	0	0	0	0	
8.17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0351	0	0	0	0	0	0	0	0	0	0	0	
8.18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.051	0	0	0	0	0	0	0	0	0	0	0	
8.19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0436	0	0	0	0	0	0	0	0	0	0	0	
8.20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0395	0	0	0	0	0	0	0	0	0	0	0	
8.21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0331	0	0	0	0	0	0	0	0	0	0	0	
8.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0449	0.0868	0	0	0	0	0	0	0	0	0	0	
8.23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.163	0	0	0	0	0	0	0	0	0	0	0	
8.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0655	0	0	0	0	0	0	0	0	0	0	0	
8.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0905	0	0	0	0	0	0	0	0	0	0	0	
8.26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0527	0	0	0	0	0	0	0	0	0	0	0	
8.27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0322	0	0	0	0	0	0	0	0	0	0	0	
8.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0744	0	0	0	0	0	0	0	0	0	0	0	
8.29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0417	0	0	0	0	0	0	0	0	0	0	0	
8.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0476	0	0	0	0	0	0	0	0	0	0	0	
8.31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0447	0	0	0	0	0	0	0	0	0	0	0	
8.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0626	0	0	0	0	0	0	0	0	0	0	0	
8.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.061	0	0	0	0	0	0	0	0	0	0	0	
8.34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0359	0	0	0	0	0	0	0	0	0	0	0	
8.35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0643	0	0	0	0	0	0	0	0	0	0	0	
8.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0641	0	0	0	0	0	0	0	0	0	0	0	
8.37	0	0	0	0																											

Therefore, in the future, the problem should be solved by modifying the *TfidfVectorizer* or *tokenization function* to be able to set *ngram\_range* for tokenizing word phrases in a corpus, so it will be able to keep keywords of each corpus.

## CHAPTER 7

### CONCLUSIONS

The developed application provides the interactive visualizations that allow users to represent the information for implementing ISMS using ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents. The application is designed to be suitable for multiple display walls and to enhance collaborative working. The application is composed of two sections. The first section is Natural Language Processing (NLP) in which the purpose is to find the relationship between ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents. The second section is the Visualization application which can be separated into two parts: SAGE2 application and web-based application. The purpose of the visualization is to represent the information of a guideline to implement ISMS and to represent the relationship between two ISO security standards from the NLP process to users. There are two versions of visualization including the Sunburst visualization and the Zoomable Circle Packing visualization.

The problem and limitations are mostly found in the implementation process on the SAGE2 application and the NLP process. The first problem is about the method to implement the SAGE2 application. The version of SAGE2 that we used to develop the application is version 4.0.12 which is not the most recent version. If users use different versions, there might be some problems. For example, some functions are available in the older version, but are not available in the newer version. The next problem is the result of the NLP process in which the similarity score is not stable and results inaccurately. This is due to the fact that the content in some clauses from ISO/IEC 27001:2013 is very little as compared to the annexes from ISO/IEC 27002:2013. As a result, the JSON files from the NLP process are still not applicable, while the JSON files that are used to generate the visualizations are still the manually mapped ones.

In conclusion, according to the usability testing, users with the visualization spend less time finding the content from ISO/IEC security standard documents than

finding it manually. In the near future, the visualization can be improved in many ways. First, the visualization could be developed to be compatible with other information security standards as well. In the NLP part, the similarity score could be further improved to give more accurate results with different methods of finding text similarity.

Copyright of Mahidol University

## APPENDIX A

### USER MANUAL FOR USING VISUALIZATION

#### A.1 How to run the application in SAGE2

1. Download SAGE2 from <http://sage2.sagecommons.org/downloads/> and install it on a server computer.
2. Go to the SAGE2 folder and Launch the batch file (Launcher.bat in Figure A.1).

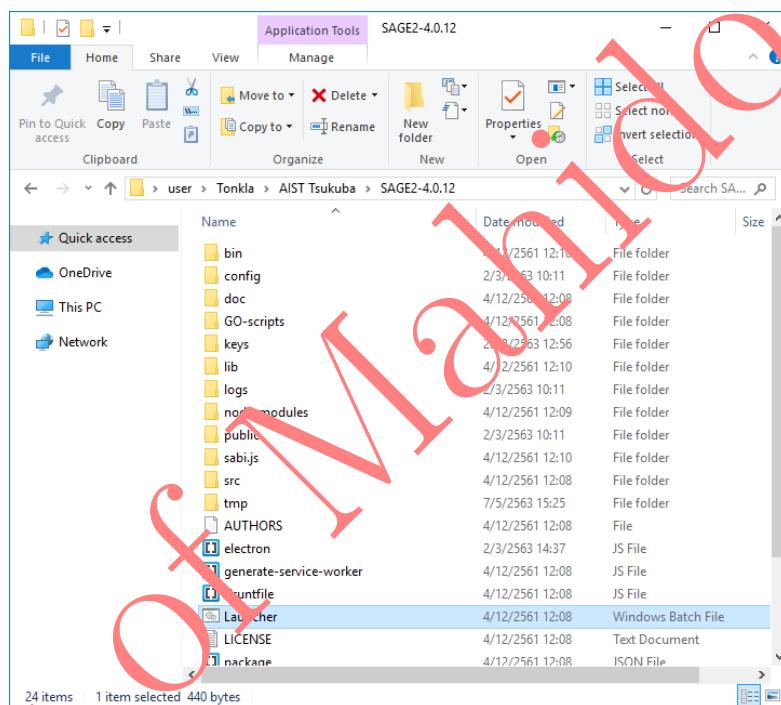


Figure A.1: Launcher.bat in the SAGE2 folder

3. In the SAGE2 control window as shown in Figure A.2 , press ‘Start’ to launch the program.
4. Open the <Server IP address>:9292 to open the SAGE2 console and press ‘Log in as guest’. The result is shown in Figure A.3.

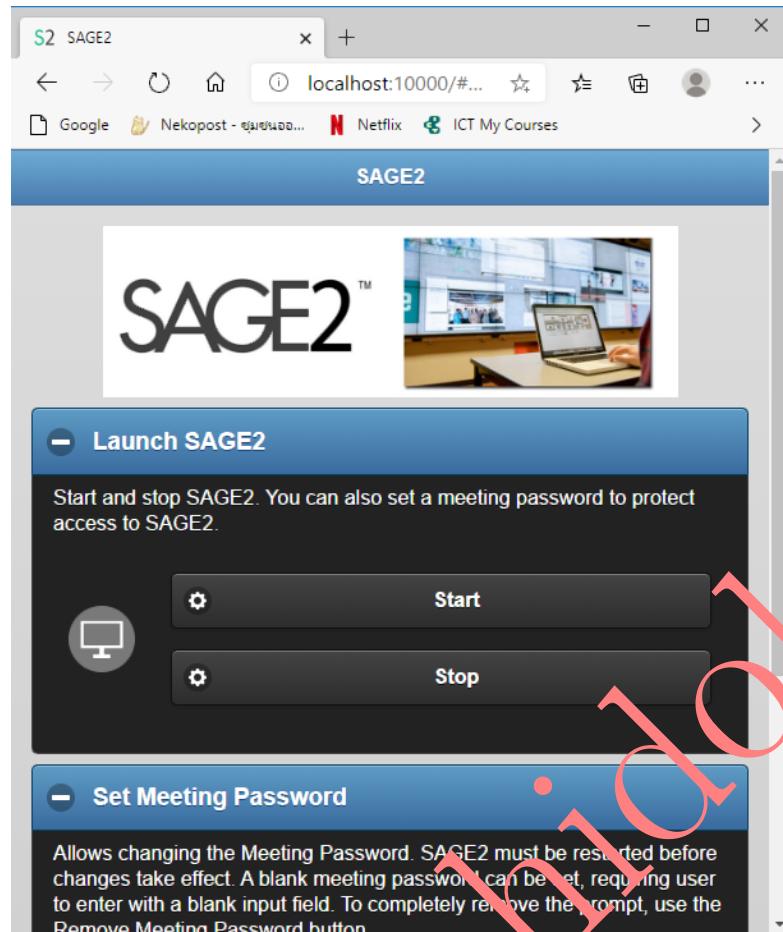


Figure A.2: The SAGE2 control window



Figure A.3: The SAGE2 console

5. Drop the zip file of the application through the SAGE2 console to install the application as shown in Figure A.4.
6. If the application has already been installed, a user can press 'App Launcher' and

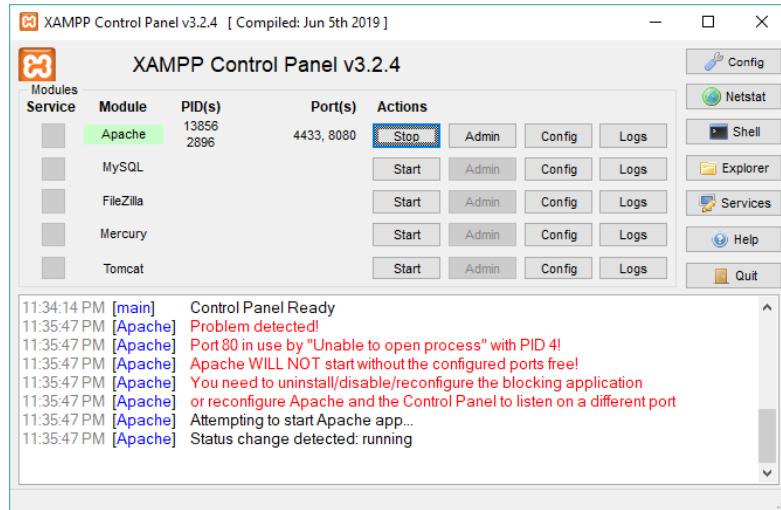


Figure A.4: When the user drops a Zip file in the SAGE2 console

choose the application to launch as shown in Figure A.5.

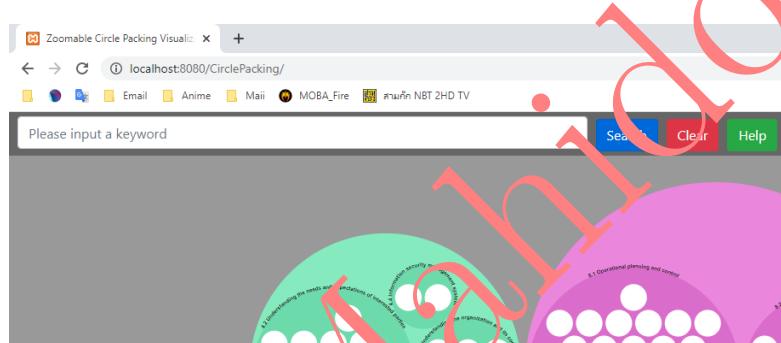


Figure A.5: List of applications that have already been installed on SAGE2

7. The application will be launched and the 'SAGE2 Pointer' can be used to interact with the application as shown in Figure A.6.

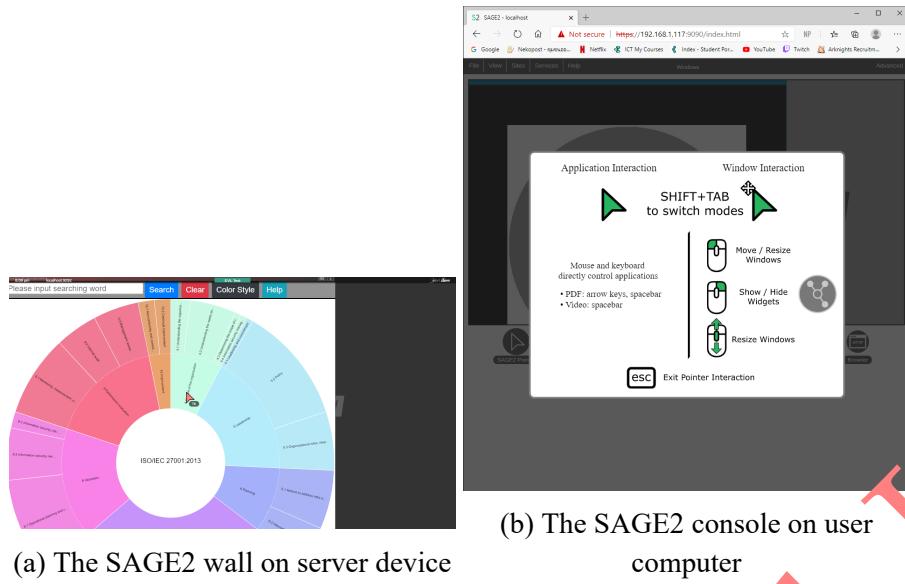


Figure A.6: The example of structure from the create JSON process

## A.2 How to run application in web application

1. Download XAMPP from <https://www.apachefriends.org/download.html> and install.
2. Go to the XAMPP folder and open the 'htdocs' folder as shown in Figure A.7.

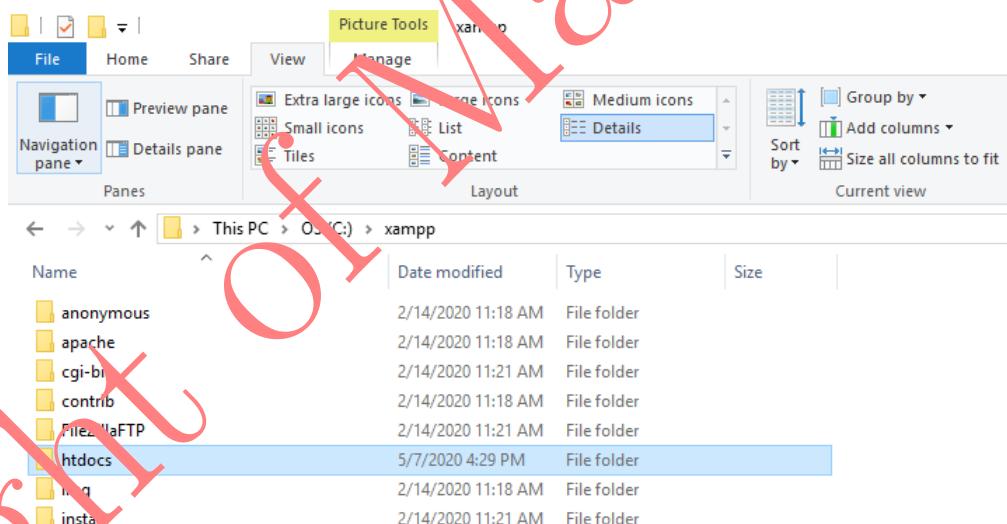


Figure A.7: The XAMPP folder

3. Move/Copy the project folder to this folder and rename it from 'workingHTML.html' to 'index.html' as shown in Figure A.8.

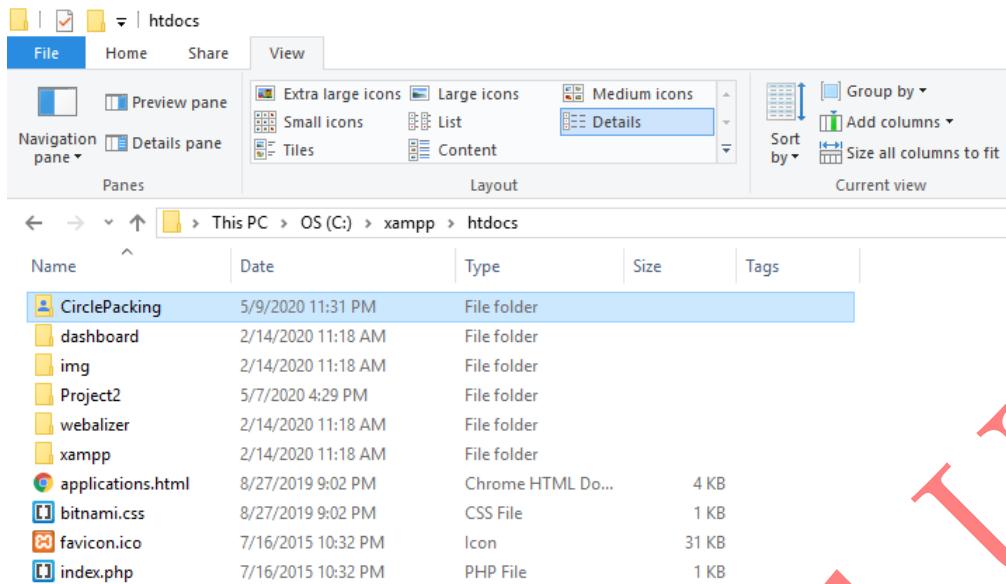


Figure A.8: Move the application folder to htdocs in the XAMPP folder

4. Go back and launch the XAMPP application (xampp-control.exe) as shown in Figure A.9.

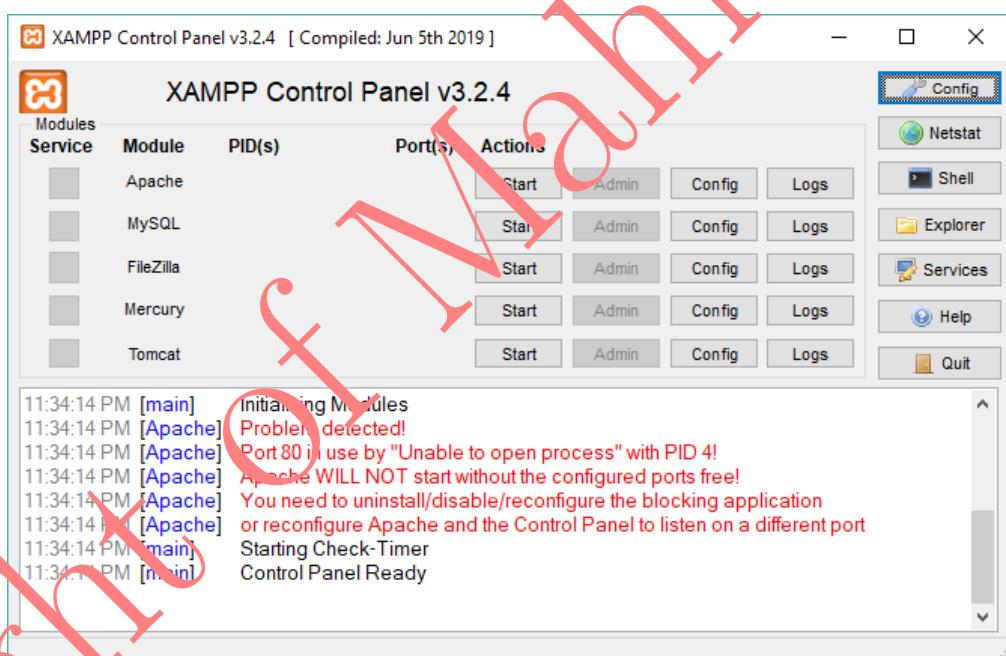


Figure A.9: Launch the XAMPP application

5. Start the ‘Apache’ module. Then, use the Apache port to access the project (port 8080 in Figure A.10).

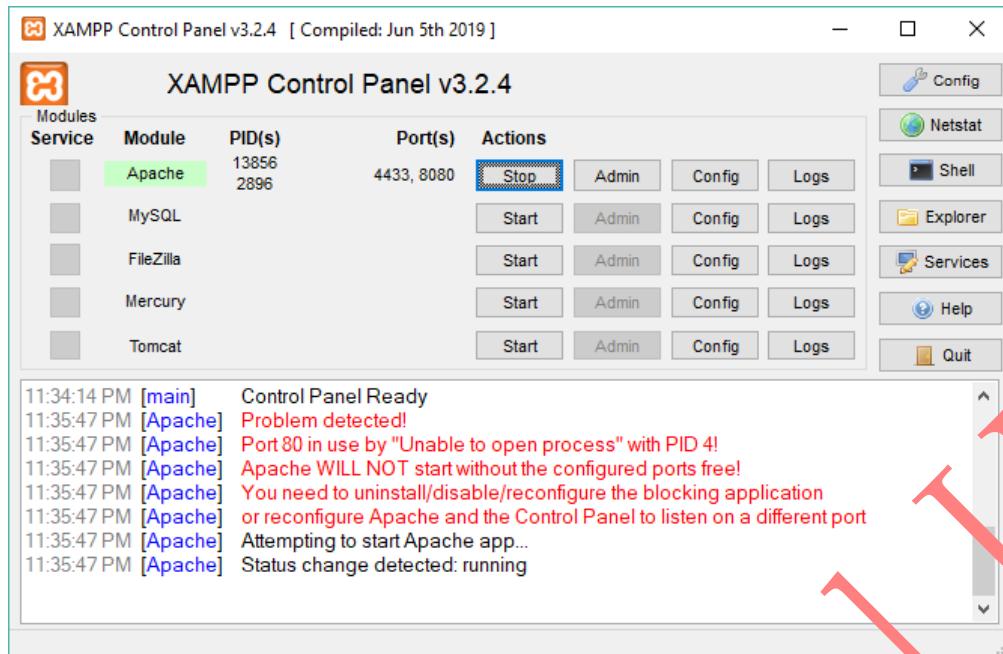


Figure A.10: Start the Apache module

6. Go to `localhost:[port]/[folder name]` to run the project (`localhost:8080/CirclePacking` in Figure A.11).

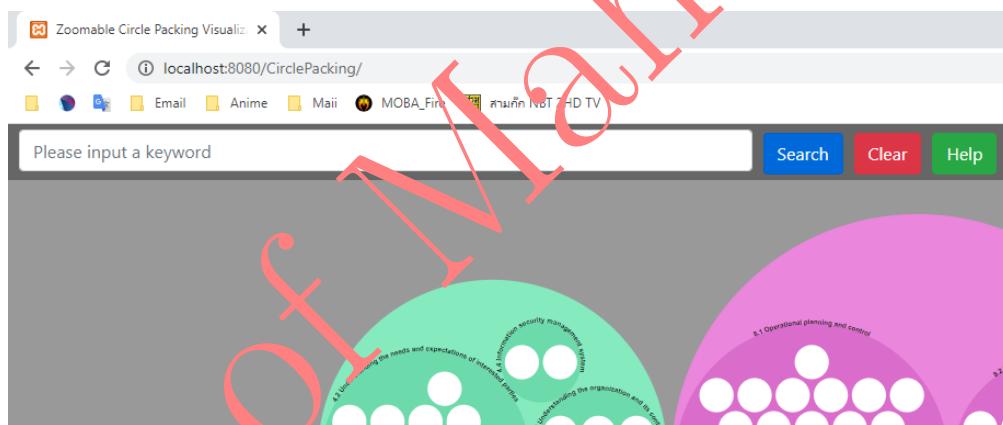


Figure A.11: The URL for the Zoomable circle packing visualization

### A.3 SAGE2 application

SAGE2 application is an application that allows users to access the same screen through its server, so with its environments and features, it will be helpful for those who use it for collaboration. On the user browser, there is a page that shows the overall screen of the SAGE2 application as the screen of Number 1 in Figure A.12. Users can

use the SAGE2 Pointer at Number 2 to access the SAGE2 application screen, and they can launch the program on the screen using the App Launcher button at Number 3.



Figure A.12: The console of the SAGE2 application

With the SAGE2 Pointer, users can access the SAGE2 screen and play with applications launched inside the SAGE2 application as shown in Figure A.13 and Figure A.14.



Figure A.13: SAGE2 screen

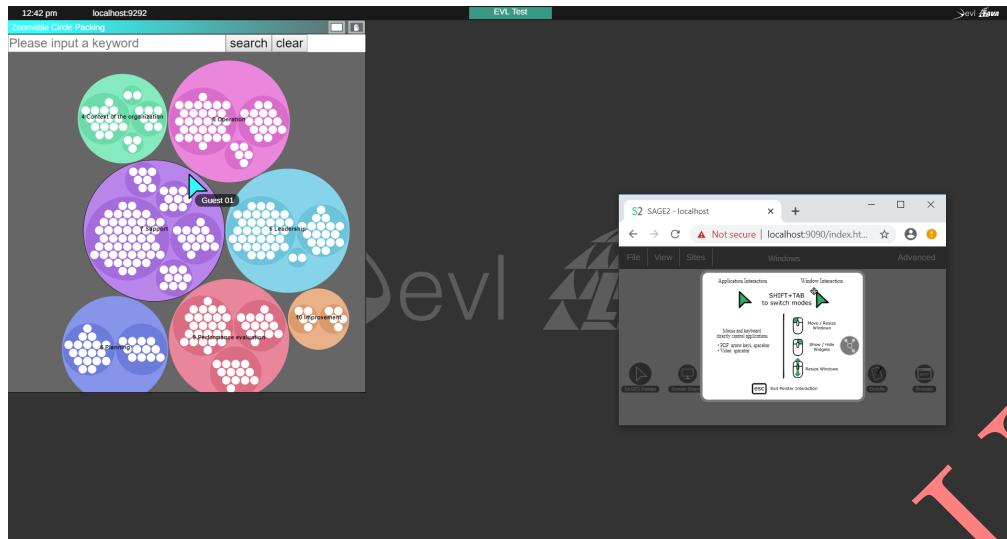


Figure A.14: Interact with the SAGE2 application using the SAGE2 pointer

#### A.4 Sunburst Visualization

The default perspective of the Sunburst visualization consists of 6 parts shown in Figure A.15 including:

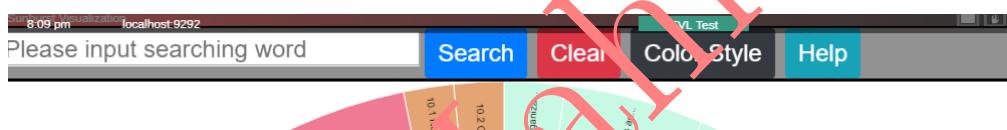


Figure A.15: The main page of the Sunburst visualization

1. Main topic at Number 1 represents the header of the focusing area, so it will show the name of the document as the header of the default page.
2. Section at Number 2 represents each main section of the document.
3. Sub section at Number 3 represents the sub sections inside each section.
4. Search bar at Number 4 represents a textbox to receive a query for searching. The search box has an autocomplete function which will help users to find related terms easily.
5. Search button at Number 5 is used to process a query after clicking the button.
6. Clear button at Number 6 is used to return the visualization to its default.

7. Blind color button at Number 7 is used to swap the color style between HSL color and grey scale.

To change the focus area, users can click on any section, and the clicked section will take the place of its previous visualization as shown in Figure A.16. Then, if users need to go back to its previous page, users can click on the middle white area to go back.

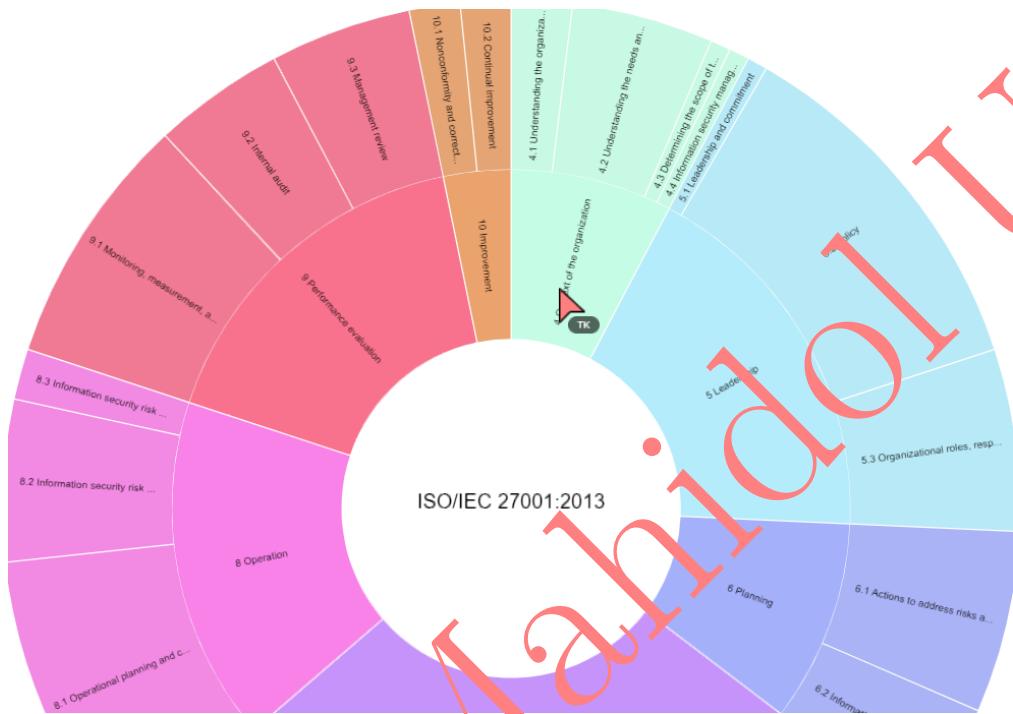


Figure A.16: The difference between each focus area of the Sunburst visualization

The search feature allows users to put a query into the search bar, and after clicking on the search button, it will highlight sections related to the searched term as shown in Figure A.17.

With the pop-up feature, users can click on an annex; then, the pop-up of the clicked annex will show up with the annex information including the annex name and its reference page in the ISO/IEC 27002:2013 document as shown in Figure A.18.

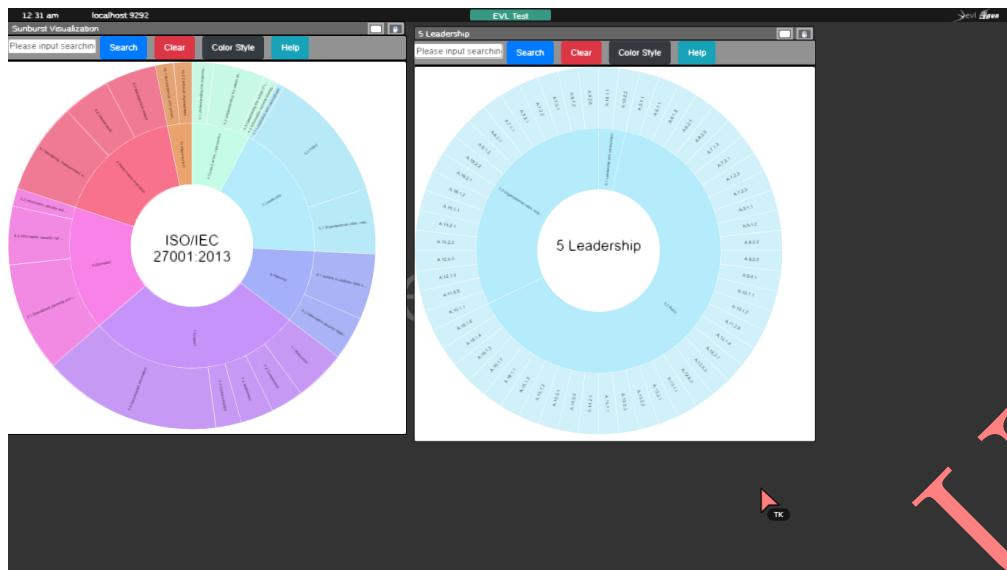


Figure A.17: The Sunburst visualization with the result from the search function

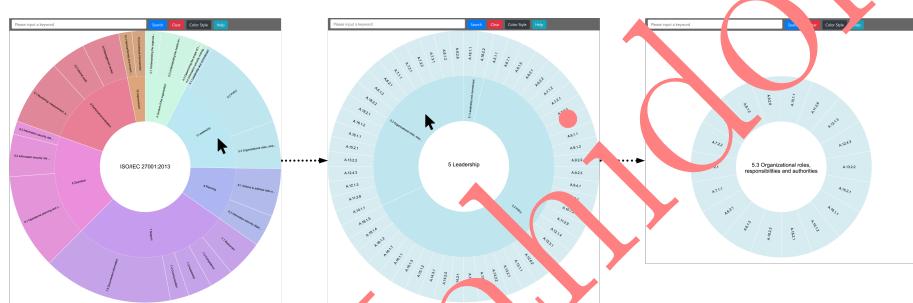


Figure A.18: Pop-up window that contains the information from the ISO/IEC 27001:2013 document

## A.5 Zoomable Circle Packing Visualization

The default perspective of the Zoomable circle packing visualization consists of 6 parts as shown in Figure A.19 including:

1. A clause node at Number 1 represents a clause in ISO/IEC27001:2013 standard.
2. A subclause node at Number 2 represents a subclause in ISO/IEC27001:2013 standard.
3. Annex node at Number 3 represents an annex in ISO/IEC27002:2013 standard that maps to the subclause where it is located.

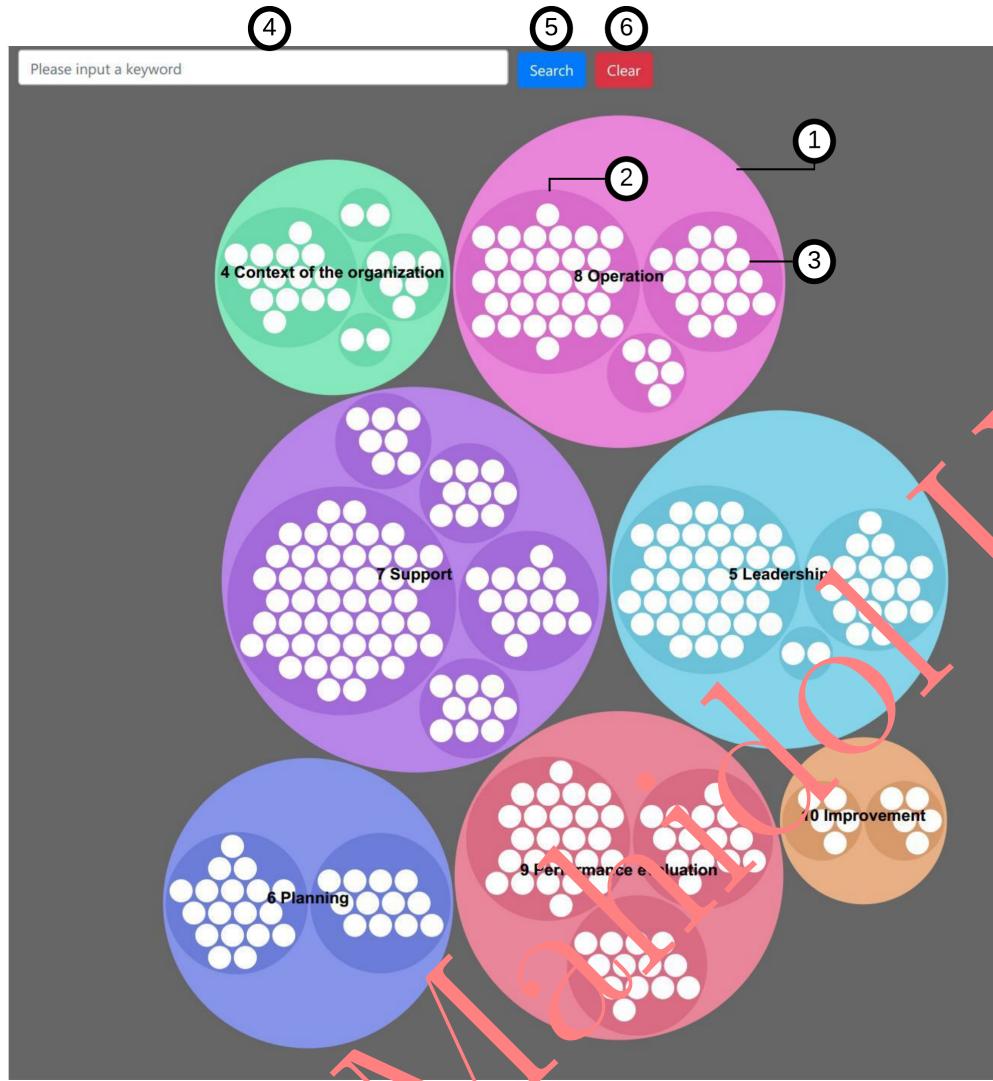


Figure A.19: The main page of the Zoomable Circle Packing visualization

4. The search bar at Number 4 represents a textbox to receive a query for searching. The search box has an autocomplete function which will help users to find related terms easily.

5. The Search button at Number 5 is used to process a query after clicking the button.

6. The Clear button at Number 6 is used to return the visualization to its default state.

In order to change the focus area, users can click on the node that they want to see, and the clicked node will be zoomed in as shown in Figure A.20. Then, if users want to focus on another node, users can click at the focussing node and it will zoom out to the default state.

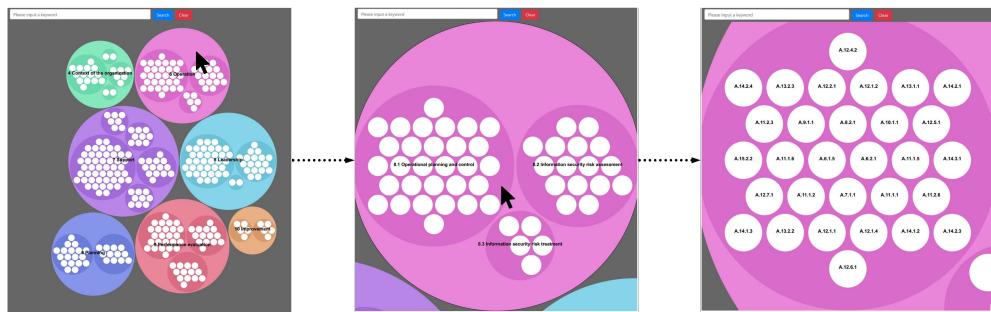


Figure A.20: The difference between each focus area of the Zoomable Circle Packing visualization

In this visualization, there is a search function that uses the search bar and the search button. In order to use this feature, users need to input keywords into the search bar and press the search button. Then, the node related to the keyword will be highlighted with yellow as shown in Figure A.21.

Moreover, there is another feature that is a pop-up feature. In this feature, users can press on the annex node for which they want to see the information, and the annex popup will appear the same as shown in Figure A.22.

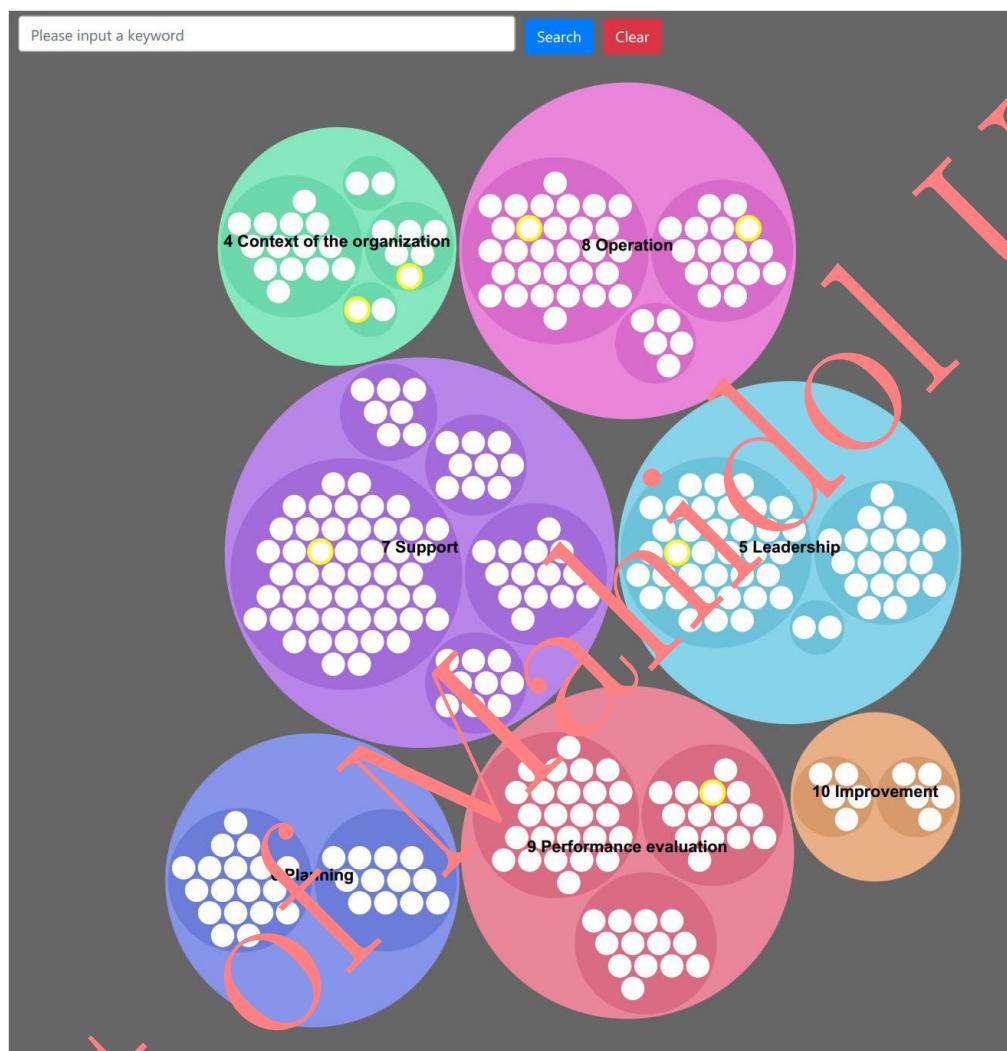


Figure A.21: The Zoomable Circle Packing visualization with the result from the search function

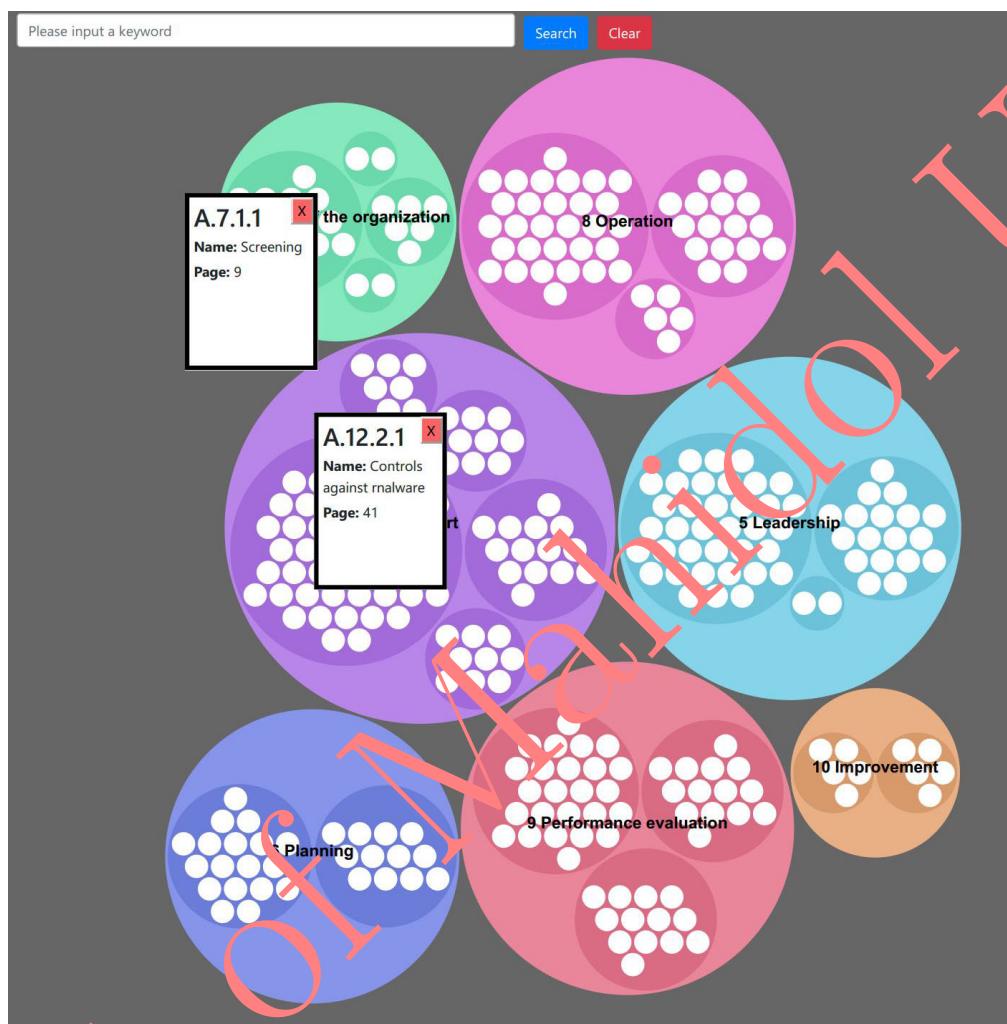


Figure A.22: Pop-up window that contains the information from the ISO/IEC 27002:2013 document

## APPENDIX B

### FULL VERSION OF IMPLEMENTATION CODE

#### B.1 Natural Language Processing

Listing B.1: Natural Language Processing Implementation

```
import os
import spacy
from spacy.tokens import Span

from sklearn.feature_extraction.text import TfidfVectorizer
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

from spacy.vocab import Vocab
import json

vocab = vocab = Vocab()
nlp = spacy.load("en_core_web_md")
def keep_token(t):
    return (t.is_alpha and
            not (t.is_space or t.is_punct or
                 t.is_stop or t.like_num))

def lemmatize_doc(doc):
    return [t.lemma_ for t in doc if keep_token(t)]

# directory to files
directory = "27001/"

# ISO var declare
detail = ""
clsno = ""
subclsno = ""
subSubclsno = ""
subclslist = []
mapNameCls = {} # clause, name
mapClause = {} # clsno, list(subclsno)
mapNameSub = {} # sclause, sname
mapDetail = {} # sclause, detail
mapSubSub = {} # ssclause, detail

for filename in os.listdir(directory):
    if(filename.endswith(".txt")):
        f = open(os.path.join(directory, filename))
        for x in f:
```

```
doc = nlp(x)
# check number pattern of clause
if(doc[0].shape_ == 'dd' or doc[0].shape_ == 'd'):
    mapClause[clsno] = subclslist
    subclslist = []
    mapNameCls[doc[0].text] = doc[2:len(doc)-1].text # map Clause no with Clause name
    clsno = doc[0].text # map Clause no with Clause name
    if(doc[1].text == ')'):
        detail += "\n" + doc[0:len(doc)].text # append detail of SubClause
        if(not detail): # map detail of the last SubClause if detail is not empty
            mapDetail[subcls] = detail #
            mapDetail[subSub] = detail
            continue # continue to newline (detail)
    # check number pattern of subclause
    if(doc[0].shape_ == 'd.d' or doc[0].shape_ == 'd.dd' or doc[0].shape_ == 'dd.d' or doc[0].shape_ == 'dd.dd'):
        mapDetail[subcls] = detail # map detail of the last SubClause
        subcls = doc[0].text # get New SubClause No
        subclslist.append(subcls)

        mapNameSub[subcls] = doc[2:len(doc)-1].text # map SubClause no with SubClause name
        detail = doc[2:len(doc)-1].text # init new detail of SubClause
        continue
    else:
        detail += "\n" + doc[0:len(doc)].text # append detail of SubClause
f.close()
else:
    continue

# housekeeping
mapDetail[subcls] = detail # map detail of the last SubClause
dict.pop(mapDetail, "") # delete value of empty key (debug)

mapClause[clsno] = subclslist
dict.pop(mapClause, "")

empty_keys = [k for k,v in mapClause.items() if not v]
for k in empty_keys:
    del mapClause[k]

# Annex var declare
detail = ""
detail2 = ""
aNo = ""
subANo = ""
subSubNo = ""
subSubSub = ""

mapNameA = {} # aNo, AName
mapNameSubA = {} # subANo, subAName
mapDetailA = {} # subANo, detail
mapDetailTemp = {}
mapNameSubSub = {} # subSubANo, detail

annex = open("27002.txt")
for x in annex:
    doc = nlp(x)

    if(doc[0].shape_ == 'dd' or doc[0].shape_ == 'd'):
```

```

aNo = "A." + doc[0].text
mapNameA[aNo] = doc[2:len(doc)-1].text # map Annex no with Annex name
if(doc[1].text == ')':
    detail += "\n" + doc[0:len(doc)].text # append detail of SubAnnex
if(not detail2):
    mapDetailTemp[subSubNo] = detail2 # map detail of the last SubAnnex if detail is not empty
    mapDetailA[subANo] = mapDetailTemp
    mapDetailTemp = {}
    continue
if(doc[0].shape_ == 'd.d' or doc[0].shape_ == 'd.dd' or doc[0].shape_ == 'dd.d' or doc[0].shape_ == 'dd.dd'):
    mapDetailTemp[subSubNo] = detail2
    detail2 = ""
    subSubNo = ""
    mapDetailA[subANo] = mapDetailTemp # map detail of the last SubAnnex
    mapDetailTemp = {}
    subANo = "A." + doc[0].text # get New SubAnnex No
    mapNameSubA[subANo] = doc[2:len(doc)-1].text # map SubAnnex no with SubAnnex name
    detail = doc[2:len(doc)-1].text # init new detail of SubAnnex
    continue
if(doc[0].shape_ == 'd.d.d' or doc[0].shape_ == 'd.dd.d' or doc[0].shape_ == 'dd.d.d' or doc[0].shape_ == 'dd.dd.d'):
    ↪ shape_ == 'dd.d.d' or doc[0].shape_ == 'dd.d.dd' or doc[0].shape_ == 'dd.dd.d' or doc[0].shape_ == 'dd.dd.dd':
    if(doc[1].text == ')'):
        detail2 += "\n" + doc[0:len(doc)].text
        continue
    mapDetailTemp[subSubNo] = detail2
    subSubNo = "A." + doc[0].text # get New SubAnnex No
    subSub = doc[2:len(doc)-1].text
    mapNameSubSub[subSubNo] = subSub # map SubAnnex no with SubAnnex name
    detail2 = doc[2:len(doc)-1].text # init new detail of SubAnnex
else:
    detail2 += "\n" + doc[0:len(doc)].text
mapDetailA[subANo] = mapDetailTemp
dict.pop(mapDetailA, "")
annex.close()

# var sim declare
avgSim = 0.0
count = 0
corpusAll = []
inCorpus = {}

for a,b in mapDetailA.items():
    corpusAll.append(b)
    inCorpus[count] = a
    count = count + 1

A'':= len(corpusAll)

for a,b in mapDetailA.items():
    for c,d in b.items():
        if(not c.startswith('A.')):
            continue
        corpusAll.append(d)
        inCorpus[count] = c
        count = count + 1

```

```
docs = [lemmatize_doc(nlp(doc)) for doc in corpusAll]
vectorizer = TfidfVectorizer(analyzer='word', ngram_range=(1,2), stop_words='english', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b')
tfword = vectorizer.fit_transform(corpusAll)

cosim = []
clsSim = {}
annexSim = {}
tmp = {}
aout = False
ain = False
allClsSim = {}
allAnnexSim = {}
simTmp = []

for i in range(len(corpusAll)):
    cosim = cosine_similarity(tfword[i:i+1], tfword)
    count = 0
    tmp = {}
    simTmp = []
    aout = inCorpus[i].startswith('A.')

    for a in cosim:
        for sim in a:
            ain = inCorpus[count].startswith('A.')
            if aout:
                if(not ain):
                    tmp[inCorpus[count]] = sim
                    simTmp.append(sim)
                else:
                    if ain:
                        tmp[inCorpus[count]] = sim
                        simTmp.append(sim)
            #print("Sim to " + inCorpus[count] + ":" + str(sim))
            count = count + 1
    #print("-----")
    if aout:
        annexSim[inCorpus[i]] = tmp
        allAnnexSim[inCorpus[i]] = simTmp
    else:
        clsSim[inCorpus[i]] = tmp
        allClsSim[inCorpus[i]] = simTmp

f = open("out/outputCls.txt", "w")
a, b, Sim = 0, 2
avgCl = {}
avgAnnex = {}

for a,b in clsSim.items():
    sim = 0.0
    s = "----- Subclause " + a + "
    f.write(s)
    avgSim = 0.0
    count = 0
    for c,d in b.items():
        avgSim += d
        count += 1
    avgCl[a] = avgSim / count
    f.write(str(avgSim / count))

for a,b in annexSim.items():
    sim = 0.0
    s = "----- Subclause " + a + "
    f.write(s)
    avgSim = 0.0
    count = 0
    for c,d in b.items():
        avgSim += d
        count += 1
    avgAnnex[a] = avgSim / count
    f.write(str(avgSim / count))
```

```
count += 1
s = "Annex\t" + c + "\tsimilarity: " + str(d) + "\n"
f.write(s)

if (count > 0):
    avgCls[a] = avgSim/count
f.close()

f = open("out/outputAnnex.txt", "w")
for a,b in annexSim.items():
    sim = 0.0
    s = "----- Annex " + a + "-----\n"
    f.write(s)
    avgSim = 0.0
    count = 0
    for c,d in b.items():
        avgSim += d
        count += 1
        s = "Subclause\t" + c + "\tsimilarity: " + str(d) + "\n"
        f.write(s)

    if (count > 0):
        avgAnnex[a] = avgSim/count
f.close()

resultCls = {}
resultAnnex = {}
resTemp = {}
clsRes = []
annexRes = []
mapClsRes = {}
mapAnnexRes = {}

relateClsFile = open("out/relateClsISO.txt", "w")

for cls,alist in clsSim.items():
    s = "----- Subclause " + cls + "-----\n"
    relateClsFile.write(s)
    resTemp = {}
    clsRes = []
    for annex,sim in alist.items():
        if(sim > (avgCls[cls] * 1.7)):
            resTemp[annex] = sim
            clsRes.append(annex)
            s = "Annex\t" + annex + "\tsimilarity: " + str(sim) + "\n"
            relateClsFile.write(s)
    resultCls[cls] = resTemp
    mapClsRes[cls] = clsRes

relateClsFile.close()

relateAnnexFile = open("out/relateAnnexISO.txt", "w")

for annex,clslist in annexSim.items():
```

```

s = "----- Annex " + annex + "
      ↗ ----- \n"
relateAnnexFile.write(s)
annexRes = []
resTemp = {}
for cls,sim in clslist.items():
    if(sim > (avgAnnex[annex] * 1.7)):
        resTemp[cls] = sim
        annexRes.append(cls)
        s = "Subclause\t" + cls + "\tsimilarity: " + str(sim) + "\n"
        relateAnnexFile.write(s)
resultAnnex[annex] = resTemp
mapAnnexRes[annex] = annexRes

relateAnnexFile.close()

j = open("out/data.json", "w")
j_ls = []
tmp_cls = {}
tmp_sub = {}
ann_ls = []
con_ls = []
for clsno, sub in mapClause.items():
    tmp_cls = {}
    tmp_cls["name"] = mapNameCls[clsno]
    tmp_cls["clause"] = clsno
    con_ls = []
    for subcls in sub:
        tmp_sub = {}
        if(subcls=='0'):
            continue
        tmp_sub["sename"] = mapNameSub[subcls]
        tmp_sub["sclause"] = subcls
        ann_ls = []
        for ann in resultCls[subcls]:
            ann_ls.append(ann)
        tmp_sub["annex"] = ann_ls
        con_ls.append(tmp_sub)
    tmp_cls["content"] = con_ls
    j_ls.append(tmp_cls)

jfile = js.dumps(j_ls, indent=4)
j.write(jfile)
j.close()

```

## B.2 SAGE2 Application

Listing B.2: SAGE2 Application for Sunburst Visualization

```

// SAGE2 is available for use under the SAGE2 Software License
//
// University of Illinois at Chicago's Electronic Visualization Laboratory (EVL)
// and University of Hawai'i at Manoa's Laboratory for Advanced Visualization and
// Applications (LAVA)
//

```

```
// See full text, terms and conditions in the LICENSE.txt included file
//
// Copyright (c) 2014

"use strict";

var SunburstViz = SAGE2_App.extend({
    /**
     * The Initialization function for SAGE2 application
     * Parameter: data
     */
    init: function(data) {
        this.SAGE2Init("div", data);

        this.element.style.backgroundColor = '#FFFFFF';

        var _this = this;
        this.id = data.id;

        this.passSAGE2PointerAsMouseEvents = true;

        this.modifiers = [];
        this.wordList = [];
        this.iso27001 = [];
        this.iso27002 = [];
        this.currentFocus = -1;
        this.checkPopup = [];
        this.checkHelp = 0;
        this.checkClause = [];
        this.annexName = "";
        this.clauseNum = 0;
        this.currentState = 0;

        // For keeping the current position
        this.positionX = 0;
        this.positionY = 0;

        // Store the position X, Y of the child applications
        this.childPos_x = new Array();
        this.childPos_y = new Array();

        if(typeof data.customLaunchParams != 'undefined'){

            if(data.customLaunchParams.state == 2){
                // The initialization for pop-up application
                _this.currentState = 2;
                // Set up the new app and its content
                var posAdjust = {};
                posAdjust.appPositionAndSize = {};
                posAdjust.appPositionAndSize.elemId = _this.id;
                posAdjust.appPositionAndSize.elemLeft = _this.sage2_x + data.customLaunchParams.x;
                posAdjust.appPositionAndSize.elemTop = _this.sage2_y + data.customLaunchParams.y;
                posAdjust.appPositionAndSize.elemHeight = 200;
                posAdjust.appPositionAndSize.elemWidth = 280;

                wsio.emit("updateApplicationPositionAndSize", posAdjust);
            }
        }
    }
});
```

```
_this.updateTitle(data.customLaunchParams.name);
_this.annexName = data.customLaunchParams.name;

var annex = data.customLaunchParams.name.split(".");
var info = data.customLaunchParams.info;
var aname = info.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1].children[parseInt(annex[3])  
    ↪ -1].name;
var page = info.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1].children[parseInt(annex[3])  
    ↪ -1].page;
var subClause = info.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1].children[parseInt(annex[3])  
    ↪ -1].control;

/* Create the detail in the popup */
var cDetail = document.createElement('p');
cDetail.innerHTML = "<b>" + aname + "<b>";
cDetail.setAttribute('style', 'font-size: 20px; margin: 0px 5px 5px 5px;')

var cRef = document.createElement('p');
cRef.innerHTML = "<b>Reference</b><br>ISO/IEC 27002: 2013 Page " + page;
cRef.setAttribute('style', 'margin: 5px');

var cSubClause = document.createElement('p')
cSubClause.setAttribute('class', 'card-text')
cSubClause.innerHTML = "<b>Related Subclause<br>(ISO/IEC 27001:2013)</b><br>";
cSubClause.setAttribute('style', 'margin: 5px')

for(var i = 0; i < subClause.length; i++) {
    cSubClause.innerHTML += subClause[i];
    if(i != subClause.length - 1)
        cSubClause.innerHTML += ',';
}

_this.element.appendChild(cDetail);
_this.element.appendChild(cRef);
_this.element.appendChild(cSubClause);

else if(data.customLaunchParams.state == 3){
    // The initialization for User manual application
    _this.currentState = 3;

    // Set up the new app and its content
    var posAdjust = {};
    posAdjust.appPositionAndSize = {};
    posAdjust.appPositionAndSize.elemId = _this.id;
    posAdjust.appPositionAndSize.elemLeft = data.customLaunchParams.x + 100;
    posAdjust.appPositionAndSize.elemTop = data.customLaunchParams.y;
    posAdjust.appPositionAndSize.elemHeight = 1000;
    posAdjust.appPositionAndSize.elemWidth = 500;

    wsio.emit("updateApplicationPositionAndSize", posAdjust);

    _this.updateTitle("User manual");

    var slideContainer = document.createElement('div');
    slideContainer.setAttribute('style', 'margin:auto; position:relative;')

    /* User manual content for picture 1 */
```

Copyright © Mahidol University

```
var slide1 = document.createElement('div');
slide1.setAttribute('class', 'mySlides');
var slide1Pic = document.createElement('img');
slide1Pic.setAttribute('src', '_this.resrcPath + 'manual/manual Sunburst—1.png');
slide1Pic.setAttribute('style', 'width:100%');
slide1.appendChild(slide1Pic);

/* User manual content for picture 2 */
var slide2 = document.createElement('div');
slide2.setAttribute('class', 'mySlides');
var slide2Pic = document.createElement('img');
slide2Pic.setAttribute('src', '_this.resrcPath + 'manual/manual Sunburst—2.png');
slide2Pic.setAttribute('style', 'width:100%');
slide2.appendChild(slide2Pic);

/* User manual content for picture 3 */
var slide3 = document.createElement('div');
slide3.setAttribute('class', 'mySlides');
var slide3Pic = document.createElement('img');
slide3Pic.setAttribute('src', '_this.resrcPath + 'manual/manual Sunburst—3.png');
slide3Pic.setAttribute('style', 'width:100%');
slide3.appendChild(slide3Pic);

/* Button for previous page in User manual */
var prev = document.createElement('a');
prev.setAttribute('class', 'prev');
prev.onclick = function(){plusSlides(-1)};
prev.innerHTML = "\u2190";

/* Button for next page in User manual */
var next = document.createElement('a');
next.setAttribute('class', 'next');
next.onclick = function(){plusSlides(1)};
next.innerHTML = "\u2192";

slideContainer.appendChild(slide1);
slideContainer.appendChild(slide2);
slideContainer.appendChild(slide3);
slideContainer.appendChild(prev);
slideContainer.appendChild(next);
_this.element.appendChild(slideContainer);

var slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
    showSlides(slideIndex += n);
}

function currentSlide(n) {
    showSlides(slideIndex = n);
}

function showSlides(n) {
    var i;
    var slides = document.getElementsByClassName("mySlides");
    var dots = document.getElementsByClassName("dot");
    for (i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
    }
    dots[slideIndex].style.background = "#f0f0f0";
    slides[n].style.display = "block";
    dots[slideIndex].style.background = "#f0f0f0";
}
```

```
if (n > slides.length) {slideIndex = 1}
if (n < 1) {slideIndex = slides.length}
for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
}
slides[slideIndex - 1].style.display = "block";
}
}
else{
    // The initialization for a Clause visualization
    _this.currentState = 1;

    // Set up the new app and its content
    var posAdjust = {};
    posAdjust.appPositionAndSize = {};
    posAdjust.appPositionAndSize.elemId = _this.id;
    posAdjust.appPositionAndSize.elemLeft = data.customLaunchParams.x;
    posAdjust.appPositionAndSize.elemTop = data.customLaunchParams.y;
    posAdjust.appPositionAndSize.elemHeight = 1000;
    posAdjust.appPositionAndSize.elemWidth = 1000;

    wsio.emit("updateApplicationPositionAndSize", posAdjust);

    _this.updateTitle(data.customLaunchParams.name);

    createButtons();

    _this.iso27002 = data.customLaunchParams.iso2;
    _this.iso27001 = data.customLaunchParams.iso1;
    _this.createSunburst(_this.createDictionary(_this.iso27001), _this);
}

else{
    // Set up for the main app
    createButtons();
    read27002();

    function read27001(){
        readFile(_this.resrcPath + 'Data/ISO27001.json', async function(error, data2) {
            _this.iso27001 = data2;
            _this.dict = await _this.createDictionary1(_this.iso27001);
            _this.createSunburst(_this.dict, _this);
        }, "JSON");
    }

    function read27002(){
        readFile(_this.resrcPath + 'Data/ISO27002.json',
        async function(error, data1){
            _this.iso27002 = await _this.createDictionary2(data1);
            read27001();
        }, "JSON");
    }
}

// Create search bar and other buttons.
function createButtons(){
    // Container for the search bar
```

```
const form = d3.select(_this.element)
    .append("div")
    .style("background-color", "#929292")
    .style("border-bottom", "3px solid black")
    .style("font", "16px Arial");

// Setup element for auto-complete function
const autocomplete = form.append("div")
    .attr("id", _this.id + "autocomplete")
    .style("position", "relative")
    .style('display', 'inline-block');

// Search query input
const query = autocomplete.append("input")
    .attr("type", "text")
    .attr("id", _this.id + "query")
    .attr("placeholder", "Please input searching word")
    .attr("size", "30")
    .style("font-size", "24px");

// Submit button element
const submitButton = form.append("input")
    .attr("type", "submit")
    .attr("value", "Search")
    .attr("class", "btn btn-primary")
    .on("click", function(){
        // Call the function to re-generate visualization again with search query
        _this.dict = _this.createDictionary1(_this.iso27001);
        _this.createSunburst(_this.dict, _this);
    })
    .style("font-size", "24px")
    .style("margin", "0px 5px");

// Clear button element
const clearButton = form.append("input")
    .attr("type", "submit")
    .attr("value", "Clear")
    .attr("class", "btn btn-danger")
    .on("click", function(){
        // Call the function to re-generate visualization without any highlight and popup
        document.getElementById(_this.id + 'query').value = "";
        _this.createSunburst(_this.createDictionary1(_this.iso27001), _this);
        _this.sendDataToChildrenApps("close");
    })
    .style("font-size", "24px")
    .style("margin", "0px 5px");

// Blind color mode button element
const color = form.append('input')
    .attr("type", "button")
    .attr("id", _this.id + "color")
    .attr("value", "Color Style")
    .attr("class", "btn btn-dark")
    .on("click", function(){
        // Change the button style and the visualization color
        if(this.getAttribute('class') == 'btn btn-dark'){
            this.setAttribute('class', 'btn btn-warning');
```

```
        _this.dict = _this.createDictionary1(_this.iso27001);
        _this.createSunburst(_this.dict, _this);
    }
    else{
        this.setAttribute('class', 'btn btn—dark');
        _this.dict = _this.createDictionary1(_this.iso27001);
        _this.createSunburst(_this.dict, _this);
    }
})
.style("font—size", "24px")
.style("margin", "0px 5px");

// Help button element
const help = form.append('input')
    .attr("type", "button")
    .attr("id", _this.id + "help")
    .attr("value", "Help")
    .attr("class", "btn btn—info")
    .on("click", function(){
        // Open new application that contain the user manual for this visualization
        if(_this.checkHelp == 0){
            _this.checkHelp = 1;
            _this.launchAppWithValues("Sunburst Visualization",
                state: 3,
                x: _this.positionX,
                y: _this.positionY
            ), 0, 0);
        }
})
.style("font—size", "24px")
.style("margin", "0px 5px");

//Container for the visualization
const svg = d3.select(_this.element)
    .append('svg')
    .attr("id", _this.id + "partitionSVG")
    .style("width", "2000")
    .style("height", "2000");
}

/**
 * The Function to create Data Dictionary from ISO/IEC 27001:2013
 * Parameter: data (JSON data from ISO/IEC27001:2013 document)
 */
createDictionary1: function(data){

    var txt = document.getElementById(this.id + 'query').value.toLowerCase();
    var checked = true;
    var highlight;

    if(txt.length == 0){ checked = false; }

    var nodeList = [];
    for(var i = 0; i < data.length; i++) {
        var name = data[i].name;
        var number = data[i].clause;
```

```
var child = [];
this.checkClause[number] = 0;
this.clauseNum = number;

if(!this.wordList.includes(name)) this.wordList.push(name);
for(var j = 0; j < data[i].content.length; j++) {
    var sname = data[i].content[j].sname;
    var snumber = data[i].content[j].sclause;
    var schild = [];
    var cHighlight = false;

    if(!this.wordList.includes(sname)) this.wordList.push(sname);
    for(var k = 0; k < data[i].content[j].annex.length; k++) {
        var annex = data[i].content[j].annex[k].split(".");
        var aname = this.iso27002.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1].children[
            ↗ parseInt(annex[3])-1].name;
        var a = data[i].content[j].annex[k];
        highlight = setHighlight(aname);
        var size = 1;

        if(!this.wordList.includes(aname)) {
            this.wordList.push(aname);
            this.checkPopup[data[i].content[j].annex[k]] = 0;
        }
        schild.push({
            name: a,
            size: size,
            color: data[i].clause,
            highlight: highlight
        });
        if((highlight)) cHighlight = true;
    }
    highlight = setHighlight(sname);
    child.push({
        name: snumber + " " + sname,
        children: schild,
        color: data[i].clause,
        highlight: highlight,
        childHighlight: cHighlight
    })
}
if(this.currentState == 1){
    var data = {
        name: number + " " + name,
        children: child,
        color: data[i].clause
    }
    return data;
}
highlight = setHighlight(name);
nodeList.push({
    name: number + " " + name,
    children: child,
    color: data[i].clause,
    highlight: highlight
})
}
```

```
var data = {  
    name: 'ISO/IEC 27001:2013',  
    children: nodeList,  
    color: 1  
}  
  
function setHighlight(text){  
    if(checked){  
        if(text.toLowerCase().includes(txt)) {  
            return true;  
        }  
    }  
    return false;  
}  
return data;  
},  
  
/**  
 * The Function to create Data Dictionary from ISO/IEC 27002:2013  
 * Parameter: data (JSON data from ISO/IEC27002:2013 document)  
 */  
createDictionary2: function (data){  
    var nodeList = [];  
    for(var i = 0; i < data.length; i++) {  
        var name = data[i].name;  
        var number = data[i].annex;  
        var child = [];  
  
        for(var j = 0; j < data[i].subtopic.length; j++) {  
            var sname = data[i].subtopic[j].name;  
            var snumber = data[i].subtopic[j].annex;  
            var schild = [];  
  
            for(var k = 0; k < data[i].subtopic[j].subtopic.length; k++) {  
                var ssname = data[i].subtopic[j].subtopic[k].name;  
                var ssnumber = data[i].subtopic[j].subtopic[k].annex;  
                var page = data[i].subtopic[j].subtopic[k].page;  
                var control = data[i].subtopic[j].subtopic[k].control;  
  
                schild.push({  
                    name: ssname,  
                    annex: ssnumber,  
                    control: control,  
                    page: page  
                });  
            }  
            child.push({  
                name: snumber + " " + sname,  
                children: schild  
            });  
        }  
        nodeList.push({  
            name: number + " " + name,  
            children: child  
        });  
    }  
    var data = {
```

```

        name: 'ISO/IEC 27002:2013',
        children: nodeList
    }
    return data;
};

/***
 * The Function to create the sunburst Visualization
 * Parameter: data (Data dictionary from createDictionary1 function), rt(This SAGE2 application)
 */
createSunburst: function(data, rt){
    d3.select("#" + rt.id + 'partitionSVG').selectAll("*").remove();
    const format = d3.format(",d");
    const width = 2000; // Visualization width
    const height = 2000; // Visualization height
    const radius = width / 6; // Visualization radius
    const arc = d3.arc()
        .startAngle(d => d.x0)
        .endAngle(d => d.x1)
        .padAngle(d => Math.min((d.x1 - d.x0) / 2, 0.005))
        .padRadius(radius * 1.5)
        .innerRadius(d => d.y0 * radius)
        .outerRadius(d => Math.max(d.y0 * radius, d.y1 * radius - 1));

    const partition = data => {
        const root = d3.hierarchy(data)
            .sum(d => d.size)
        return d3.partition()
            .size([2 * Math.PI, root.height + 1])(root);
    };
    const root = partition(data);
    var hVal = d3.scaleLinear().domain([0, 7]).range([0, 270])
    var sVal = d3.scaleLinear().domain([0, 7]).range([100, 50])
    var lVal = d3.scaleLinear().domain([0, 7]).range([80, 40])

    if(document.getElementById(rt.id + 'color').getAttribute('class') == 'btn btn-warning'){
        sVal = d3.scaleLinear().domain([0, 5]).range([0, 0]);
        lVal = d3.scaleLinear().domain([0, 7]).range([0, 90]);
    }
    root.each(d => d.current = d);

    const svg = d3.select("#" + rt.id + 'partitionSVG')
        .attr("viewBox", [0, 0, width*2, width*2])
        .style("font", "20px sans-serif");

    const g = svg.append("g")
        .attr("transform", `translate(${width / 2},${width / 2})`);

    // Element for all the nodes
    const path = g.append("g")
        .selectAll("path")
        .data(root.descendants().slice(1))
        .join("path")
        .attr('fill', d => d.depth > 0 ? `hsl(${hVal(d.data.color)}, ${sVal(d.depth)}%, ${lVal(d.data.color-4)}%)` : 'white')
        .attr("fill-opacity", d => arcVisible(d.current) ? (d.children ? 0.6 : 0.4) : 0)
        .attr("d", d => arc(d.current))
        .attr('stroke', d => arcVisible(d.current) && (d.data.highlight || d.data.childHighlight) ? "#F00" : null)
}

```

```
.attr('stroke-width', d => arcVisible(d.current) && (d.data.highlight || d.data.childHighlight) ? 3 : 0)
.on("click", popUp);

path.filter(d => d.children)
.style("cursor", "pointer")
.on("click", rt.currentState == 0 ? clicked1 : clicked2);

path.append("title")
.text(d => `${d.data.name}`);

// Element for every label on the node
const label = g.append("g")
.attr("pointer-events", "none")
.attr("text-anchor", "middle")
.style("user-select", "none")
.selectAll("text")
.data(root.descendants().slice(1))
.join("text")
.attr("dy", "0.35em")
.attr("fill-opacity", d => +labelVisible(d.current))
.attr("transform", d => labelTransform(d.current))
.style("font", d => d.children == null ? "25px sans-serif" : "20px sans-serif")
.text(d => shorterName(d.data.name));

// Element of Center circle (parent node)
const parent = g.append("g")
.datum(root)
.on("click", clicked2);

parent.append("title")
.text(d => d.parent == null ? d.data.name : `${d.parent.data.name}`)

const parentCircle = parent.append("circle")
.attr("r", radius)
.attr("fill", "none")
.attr("pointer-events", "all");

const parentText = parent.append("text")
.attr("text-anchor", "middle")
.attr("dy", "0.3em")
.style("font", "40px sans-serif")
.text(root.data.name)
.call(wrap, 650);

/*
 * The Function to create a new application with the information of only 1 clause
 * Parameter: p (the clicked node);
 */
function clicked1(p) {
    var a = p.data.name.split(" ");
    var b = a[0].split(".");
    var d = [];
    d.push(rt.iso27001[b[0]-4]);
    if(rt.checkClause[b[0]] == 0){
        rt.checkClause[b[0]] = 1;
        rt.launchAppWithValues("Sunburst Visualization",{
            name: p.data.name, // Clicked node name
            ...
```

```

state: 1, // State 1 means clause application
query: document.getElementById(rt.id + 'query').value, // Current search query
iso2: rt.iso27002, // Data dict from ISO 27002
node: d, // The node that are children of current node
x: rt.positionX, // Current position x
y: rt.positionY // Current position y
}, rt.positionX + rt.sage2_width, rt.positionY + rt.sage2_height);
}

/***
 * The Function to transition the current visualization
 * Parameter: p (the clicked node);
 */
function clicked2(p) {
  parent.datum(p.parent || root);
  parentText.text(p.data.name);
  wrap(parentText, 650)
  root.each(d => d.target = {
    x0: Math.max(0, Math.min(1, (d.x0 - p.x0)) * 2 * Math.PI,
    x1: Math.max(0, Math.min(1, (d.x1 - p.x0)) * 2 * Math.PI,
    y0: Math.max(0, d.y0 - p.depth),
    y1: Math.max(0, d.y1 - p.depth)
  });
  const t = g.transition().duration(750);

  // Transition the data on all arcs, even the ones that aren't visible
  // so that if this transition is interrupted, entering arcs will start
  // the next transition from the desired position.
  path.transition(t)
    .tween("data", d => {
      const i = d3.interpolate(d.current, d.target);
      return t => d.current = i(t);
    })
    .filter(function (d) {
      return +this.getAttribute("fill-opacity") || arcVisible(d.target);
    })
    .attr("fill-opacity", d => arcVisible(d.target) ? (d.children ? 0.6 : 0.4) : 0)
    .attr("stroke", d => arcVisible(d.target) && (d.data.highlight || d.data.childHighlight) ? "#F00" : null)
    .attr("stroke-width", d => arcVisible(d.target) && (d.data.highlight || d.data.childHighlight) ? 3 : 0)
    .attrTween("d", d => () => arc(d.current));

  label.filter(function (d) {
    return +this.getAttribute("fill-opacity") || labelVisible(d.target);
  }).transition(t)
    .attr("fill-opacity", d => +labelVisible(d.target))
    .attrTween("transform", d => () => labelTransform(d.current))
}

/***
 * The Function to wrap the text that exceed 1 line
 * Parameter: text (The text from each node), len (The limit length for one line);
 */
function wrap(text, len) {
  text.each(function() {
    var txt = d3.select(this)
    var words = txt.text().split(/\s+/).reverse()

```

```
var word
var line = []
var lineNumber = 0
var lineHeight = 1.1
var tspan = txt.text(null).append('tspan')
    .attr('x', 0)
    .attr('y', 0)
    .attr('dy', 0 + 'em')

while(word = words.pop()) {
    line.push(word)
    tspan.text(line.join(" "))
    if(tspan.node().getComputedTextLength() > len) {
        line.pop()
        tspan.text(line.join(" "))
        line = [word]
        lineNumber++
        var tmp = lineNumber * lineHeight
        console.log('Multiplication:')
        console.log(tmp)
        tspan = txt.append('tspan')
            .attr('x', 0)
            .attr('y', 0)
            .attr('dy', `${tmp}em`)
            .text(word)
    }
}
}

/***
 * The Function to create a new popup application, using clicked node information
 * Parameter: d (the clicked node);
 */
function popUp(d) {
    if(arcVisible(d)) {
        // If the popup existed, do not create again
        if(rt.checkPop[d.data.name] == 0) {
            rt.checkPop[d.data.name] = 1;
            // Start new App with these parameter
            rt.launchAppWithValues("Sunburst Visualization",{
                state: 2, // State 2 means annex pop-up window
                info: rt.iso27002, // Data dict from ISO 27002
                name: d.data.name, // Name of the clicked annex
                x: rt.mousePositionX, // Current pointer position X
                y: rt.mousePositionY // Current pointer position Y
            }, 0, 0);
        }
    }
}

/***
 * The Function to check if the current arc is visible
 * Parameter: d (node);
 */
function arcVisible(d) {
    return d.y1 <= 3 && d.y0 >= 1 && d.x1 > d.x0;
```

```
}

/**
 * The Function to check if the current label is visible
 * Parameter: d (node);
 */
function labelVisible(d) {
    return d.y1 <= 3 && d.y0 >= 1 && (d.y1 - d.y0) * (d.x1 - d.x0) > 0.03;
}

/**
 * The Function to shorter name if its exceed the node size
 * Parameter: d (node);
 */
function shorterName(d){
    if(d.length > 30){
        return d.substring(0, 30) + "...";
    }
    return d;
}

/**
 * The Function to transform the label of the node
 * Parameter: d (node);
 */
function labelTransform(d) {
    const x = (d.x0 + d.x1) / 2 * 180 / Math.PI;
    const y = (d.y0 + d.y1) / 2 * radius;
    return `rotate(${x - 90}) translate(${y},0) rotate(${x < 180 ? 0 : 180})`;
},
removeClause: function(data){
    this.checkClause[data] = 0;
},
removePopup: function(data){
    this.checkPopup[data] = 0;
},
load: function(date) {
    this.refresh(date);
},
draw: function(data){
},
/*
 * Method called by SAGE2, and calls the application 'quit' method
 *
 * @method terminate
 */
terminate: function() {
    // Close all children before closing the app
    if(this.currentState == 2) this.sendDataToParentApp("removePopup", this.annexName);
    else if(this.currentState == 1) this.sendDataToParentApp("removeClause", this.clauseNum);
}
```

```
this.sendDataToChildrenApps("close");
    if (typeof this.quit === 'function') {
        this.quit();
    }
    if (isMaster && this.hasFileBuffer === true) {
        wsio.emit('closeFileBuffer', {id: this.div.id});
    }
    // hide remote pointers if any
    SAGE2RemoteSitePointer.appQuitHidePointers(this);
    // remove values placed on server
    this.serverDataRemoveAllValuesGivenToServer();
},

/**
 * Close the application itself
 *
 * @method close
 */
close: function() {
if(this.currentState == 2) this.sendDataToParentApp("removePopup", this.annexName);
else if(this.currentState == 1) this.sendDataToParentApp("removeClause", this.clauseNum)
// Close all children before closing the app
this.sendDataToChildrenApps("close");
    // send the message to server
    wsio.emit('deleteApplication', {appId: this.id});
},
event: function(eventType, position, user_id, data, date) {
    if (eventType === "pointerPress" && (data.button === "left")) {
        // Left click
    } else if (eventType === "pointerMove") {
        // move
        this.positionX = position.x;
        this.positionY = position.y;
    } else if (eventType === "pointerRelease" && (data.button === "left")) {
        // Left click release
    } else if (eventType === "pointerScroll") {
        // Scroll events: reverse the amount to get correct direction
    } else if (eventType === "widgetEvent") {
        // widget events
    } else if (eventType === 'keyboard' || (data.code == 8 && data.state == 'down')) {
        var a, b, i, j;
        var count = 0;
        var query = document.getElementById(this.id + 'query');
        var val = query.value;

        if(eventType === 'keyboard')
            val += data.character;
        else
            val = val.substr(0, val.length - 1);
        closeAllLists();
    }
    if(!val) return false;
}
```

```
a = document.createElement('div');
a.setAttribute('id', this.id + 'query_autocomplete-list');
a.setAttribute('class', 'autocomplete-items');
a.setAttribute('style', 'position:absolute; border:1px solid #d4d4d4; left:0; right:0; border-bottom:none; border-top:none; z-index:99; top:100%; background-color:#fff;');
query.parentNode.appendChild(a);
for(i = 0; i < this.wordList.length; i++) {
    j = 0;
    while(j < this.wordList[i].length) {
        if(j == 0 || !this.wordList[i].substr(j - 1, 1).localeCompare(" ")) {
            if(this.wordList[i].substr(j, val.length).toUpperCase() == val.toUpperCase()) {
                b = document.createElement('div');
                b.innerHTML = this.wordList[i].substr(0, j);
                b.innerHTML += '<strong>' + this.wordList[i].substr(j, val.length) + '</strong>';
                b.innerHTML += this.wordList[i].substr(j + val.length);
                b.innerHTML += '<input type="hidden" value="" + this.wordList[i] + "">';
                b.setAttribute("style", "padding: 10px; cursor: pointer; background-color: #fff; border-bottom: 1px solid #d4d4d4");
                b.addEventListener('click', function(e) {
                    query.value = this.getElementsByTagName('input')[0].value;
                    closeAllLists();
                });
                a.appendChild(b);
                count++;
                break;
            }
        }
        j++;
    }
    if(count == 10) break; // Limit the autocomplete to 10 elements
}
} else if(eventType === "spec.Key") {
    if(data.code == 38 && data.state == 'down') {
        var x = document.getElementById(this.id + 'query_autocomplete-list');
        if(x) x = x.getElementsByTagName('div');
        this.currentFocus--;
        addActive(x, this);
    }
    // Down Arrow
    if(data.code == 40 && data.state == 'down') {
        var x = document.getElementById(this.id + 'query_autocomplete-list');
        if(x) x = x.getElementsByTagName('div');
        this.currentFocus++;
        addActive(x, this);
    }
    // Enter
    if(data.code == 13 && data.state == 'down') {
        var x = document.getElementById(this.id + 'query_autocomplete-list');
        if(x) x = x.getElementsByTagName('div');
        if(this.currentFocus > -1) {
            if(x) x[this.currentFocus].click();
        }
    }
}

function addActive(x, y) {
    if(!x) return false;
```

```

removeActive(x);
if(y.currentFocus >= x.length) y.currentFocus = 0;
if(y.currentFocus < 0) y.currentFocus = x.length - 1;
x[y.currentFocus].setAttribute('style', 'padding: 10px; cursor: pointer; background-color: #ccc; border-bottom: 1
    ↪ px solid #d4d4d4');
}

function removeActive(x) {
for(var i = 0; i < x.length; i++) {
    x[i].setAttribute('style', 'padding: 10px; cursor: pointer; background-color: #fff; border-bottom: 1px solid
        ↪ d4d4d4');
}
}

function closeAllLists(ele) {
var x = document.getElementsByClassName('autocomplete-items');
for(var i = 0; i < x.length; i++) {
    if(ele != x[i] && ele != query)
        x[i].parentNode.removeChild(x[i]);
}
}

document.addEventListener('click', function(e) {
    closeAllLists(e.target);
});
}
);
});
```

Listing B.3: SAGE2 Application fo. (Zoomable Circle Packing Visualization

```

"use strict";

/**
 * Create a child app: http://sage2.sagecommons.org/wp-content/api/classes/SAGE2_App.html (launchAppWithValues)
 * Source Code: http://sage2.sagecommons.org/wp-content/api/files/public_src_SAGE2_App.js.html#l1176
 */

var working = SAGE2_App.extend({
    init: function(data) {
        /* Set <div> as the container of all elements */
        this.SAGE2Init("div", data);
        this.element.style.backgroundColor = "white";
        this.passSAGE2PointerAsMouseEvents = true;
        this.currentFocus = -1; // Cannot be in the event function because the value will be zero all the time
        this.wordList = [];
        this.iso;
        this.iso2;

        /* For creating a child application */
        this.sentFocus;
        this.isChildApp = false;

        /* For save mouse pointer */
        this.pointerX;
        this.pointerY;
```

```
var _this = this;
var open = false;

/* Receive data from parent app */
if(data.customLaunchParams != undefined) {
    this.sentFocus = data.customLaunchParams.sentNode;
    this.isChildApp = true;
}

/* Read file 2 before file 1 because we runAllProgram in reading file 1 */
readFile(_this.resrcPath + "ISO27002.json", function(error, inf) {
    _this.iso2 = inf;
    readFile(_this.resrcPath + "ISO27001.json", function(error, inf) {
        _this.iso = inf;
        runAllProgram(");
        }, "JSON");
    }, "JSON");

/* Search Container */
const searchContainer = d3.select(_this.element).append('div')
    .style('position', 'relative')
    .style('top', '0px')
    .style('background-color', '#666')
    .style('padding', '5px')

/* Search Bar (ID = 'query') */
const searchBar = searchContainer.append('input')
    .attr('id', this.div.id + '_query')
    .attr('type', 'text')
    .attr('placeholder', 'Please input a keyword')
    .attr('class', 'form-control')
    .style('width', '50%')
    .style('display', 'inline-block')
    .style('margin', '0px 5px')

/* Search Button (ID = '_submit') */
const searchButton = searchContainer.append('input')
    .attr('id', this.div.id + '_search')
    .attr('type', 'button')
    .attr('value', 'Search')
    .attr('class', 'btn btn-primary')
    .style('margin', '0px 5px')

/* Clear Button (id = '_clear) */
const clearButton = searchContainer.append('input')
    .attr('id', this.div.id + '_clear')
    .attr('type', 'button')
    .attr('value', 'Clear')
    .attr('class', 'btn btn-danger')
    .style('margin', '0px 5px')

/* Help Button (id = '_help) */
const helpButton = searchContainer.append('input')
    .attr('id', this.div.id + '_help')
    .attr('type', 'button')
    .attr('value', 'Help')
    .attr('class', 'btn btn-success')
```

```
.style('margin', '0px 5px')

/* Create manual popup */
var manualContainer = document.createElement('div')
manualContainer.setAttribute('style', 'width:50%; position:fixed; border:5px solid #555;')
manualContainer.id = this.div.id + '_manualContainer'
manualContainer.style.backgroundColor = 'black'
manualContainer.style.visibility = 'hidden'

var manualTitleBar = document.createElement('h3')
manualTitleBar.style.backgroundColor = 'black'
manualTitleBar.style.color = 'white'
manualTitleBar.style.textAlign = 'center'
manualTitleBar.innerHTML = 'Manual'

var manualCloseBtn = document.createElement('button')
manualCloseBtn.setAttribute('style', 'position:absolute; top:3px; right:3px; background-color:rgb(255,100,100); border-radius:3px;')
manualCloseBtn.innerHTML = 'X'
manualCloseBtn.id = this.div.id + '_manualClose'

var manual = document.createElement('div')
manual.setAttribute('class', 'carousel slide')
manual.setAttribute('data-ride', 'carousel')
manual.id = this.div.id + '_manual'

var manualOL = document.createElement('ol')
manualOL.setAttribute('class', 'carousel-indicators')

var manualLI0 = document.createElement('li')
manualLI0.setAttribute('data-target', '#'+_this.div.id+'._manual')
manualLI0.setAttribute('data-slide-to', '0')
manualLI0.setAttribute('class', 'active')

var manualLI1 = document.createElement('li')
manualLI1.setAttribute('data-target', '#'+_this.div.id+'._manual')
manualLI1.setAttribute('data-slide-to', '1')

var manualLI2 = document.createElement('li')
manualLI2.setAttribute('data-target', '#'+_this.div.id+'._manual')
manualLI2.setAttribute('data-slide-to', '2')

manualOL.appendChild(manualLI0)
manualOL.appendChild(manualLI1)
manualOL.appendChild(manualLI2)

var manualDIV = document.createElement('div')
manualDIV.setAttribute('class', 'carousel-inner')

var manualIN0 = document.createElement('div')
manualIN0.setAttribute('class', 'carousel-item active')

var manualIN1 = document.createElement('div')
manualIN1.setAttribute('class', 'carousel-item')

var manualIN2 = document.createElement('div')
manualIN2.setAttribute('class', 'carousel-item')
```

```
var manualIMG0 = document.createElement('img')
manualIMG0.setAttribute('src', _this.resrcPath + 'manv21.png')
manualIMG0.setAttribute('class', 'd-block w-100')

var manualIMG1 = document.createElement('img')
manualIMG1.setAttribute('src', _this.resrcPath + 'manv22.png')
manualIMG1.setAttribute('class', 'd-block w-100')

var manualIMG2 = document.createElement('img')
manualIMG2.setAttribute('src', _this.resrcPath + 'manv23.png')
manualIMG2.setAttribute('class', 'd-block w-100')

manualIN0.appendChild(manualIMG0)
manualIN1.appendChild(manualIMG1)
manualIN2.appendChild(manualIMG2)

manualDIV.appendChild(manualIN0)
manualDIV.appendChild(manualIN1)
manualDIV.appendChild(manualIN2)

var manualPREV = document.createElement('a')
manualPREV.setAttribute('class', 'carousel-control-prev')
manualPREV.setAttribute('href', '#' + _this.div.id + '_manual')
manualPREV.setAttribute('role', 'button')
manualPREV.setAttribute('data-slide', 'prev')

var manualNEXT = document.createElement('a')
manualNEXT.setAttribute('class', 'carousel-control-next')
manualNEXT.setAttribute('href', '#' + _this.div.id + '_manual')
manualNEXT.setAttribute('role', 'button')
manualNEXT.setAttribute('data-slide', 'next')

var manualSPAN0PREV = document.createElement('span')
manualSPAN0PREV.setAttribute('class', 'carousel-control-prev-icon')
manualSPAN0PREV.setAttribute('aria-hidden', 'true')

var manualSPAN1PREV = document.createElement('span')
manualSPAN1PREV.setAttribute('class', 'sr-only')
manualSPAN1PREV.innerHTML = 'Previous'

var manualSPAN0NEXT = document.createElement('span')
manualSPAN0NEXT.setAttribute('class', 'carousel-control-next-icon')
manualSPAN0NEXT.setAttribute('aria-hidden', 'true')

var manualSPAN1NEXT = document.createElement('span')
manualSPAN1NEXT.setAttribute('class', 'sr-only')
manualSPAN1NEXT.innerHTML = 'Previous'

manualPREV.appendChild(manualSPAN0PREV)
manualPREV.appendChild(manualSPAN1PREV)
manualNEXT.appendChild(manualSPAN0NEXT)
manualNEXT.appendChild(manualSPAN1NEXT)

manual.appendChild(manualOL)
manual.appendChild(manualDIV)
manual.appendChild(manualPREV)
```

```
manual.appendChild(manualNEXT)

manualContainer.appendChild(manualTitleBar)
manualContainer.appendChild(manualCloseBtn)
manualContainer.appendChild(manual)

_this.element.appendChild(manualContainer)

/* Set onclick listener to the button */
$('#' + this.div.id + '_clear').click(function() {
    var all = _this.element.getElementsByTagName('span')
    while(all.length > 0)
        _this.element.removeChild(all[0])
    runAllProgram("")
})

$('#' + this.div.id + '_search').click(function() {
    var id = _this.div.id + '_query'
    var txt = document.getElementById(id).value
    runAllProgram(txt);
})

$('#' + this.div.id + '_help').click(function() {
    _this.launchAppWithValues("zoompack_subapp1", {}, 500, 700, 14);
})

$('#' + this.div.id + '_manualClose').click(function() {
    var man = document.getElementById(_this.div.id + '_manualContainer')
    man.style.visibility = 'hidden'
})

$('#' + this.div.id + '_manualContainer').draggable()

function runAllProgram(txt) {
    var query = document.getElementById(_this.div.id + '_query')
    var node = _this.createDict(_this.iso, _this.iso2, txt, _this)
    _this.wordList.sort();
    _this.autocomplete(query, _this.wordList, _this)
    _this.visualize(_this.iso, node, _this)
}

// SAGE2 must have these functions; otherwise, it will have a bug
draw: function(date) {
},
event: function(eventType, position, user_id, data, date) {
    if(eventType == 'pointerRelease') {
        this.pointerX = position.x;
        this.pointerY = position.y;
    }
    // The old code use OR
    if(eventType == 'keyboard' && (data.code == 8 && data.state == 'up')) {
        var a, b, i, j;
        var count = 0;
        var val = document.getElementById(rt.div.id + '_query').value;
    }
}
```

```
if(eventType == 'keyboard')
    val += data.character;
else
    val = val.substr(0, val.length - 1);

closeAllLists();

if(!val) return false;

a = document.createElement('div');
a.setAttribute('id', rt.div.id + '_query_autocomplete_list');
a.setAttribute('class', 'autocomplete-items');
a.setAttribute('style', 'position: absolute; border: 1px solid #d4d4d4; left: 0; right: 0; border-bottom: none; border-top: none; z-index: 99; top: 100%; background-color: #fff;');
query.parentNode.appendChild(a);

for(i = 0; i < this.wordList.length; i++) {
    j = 0;
    while(j < this.wordList[i].length) {
        if(j == 0 || !this.wordList[i].substr(j - 1, 1).localeCompare(" ") ) {
            if(this.wordList[i].substr(j, val.length).toUpperCase() == val.toUpperCase()) {
                b = document.createElement('div');
                b.innerHTML = this.wordList[i].substr(0, j),
                b.innerHTML += '<strong>' + this.wordList[i].substr(j, val.length) + '</strong>';
                b.innerHTML += this.wordList[i].substr(j + val.length);
                b.innerHTML += '<input type="hidden" value="' + this.wordList[i] + '"';
                b.addEventListener('click', function(e) {
                    query.value = this.querySelector('input')[0].value;
                    closeAllLists();
                });
                a.appendChild(b);
                count++;
                break;
            }
            j++;
        }
    }
    if(count == 10) break; // Limit the autocomplete to 10 elements
}

// Up Arrow
if(data.code == 38 && data.state == 'down') {
    var x = document.getElementById(rt.div.id + '_query_autocomplete_list');
    if(x) x = x.getElementsByTagName('div');
    this.currentFocus--;
    addActive(x, this);
}

// Down Arrow
if(data.code == 40 && data.state == 'down') {
    var x = document.getElementById(rt.div.id + '_query_autocomplete_list');
    if(x) x = x.getElementsByTagName('div');
    this.currentFocus++;
    addActive(x, this);
}
```

```
}

// Enter
if(data.code == 13 && data.state == 'down') {
    var x = document.getElementById(rt.div.id + '_query_autocomplete—list');
    if(x) x.getElementsByTagName('div');
    if(this.currentFocus > —1) {
        if(x[x[this.currentFocus].click()]);
        var node = this.createDict(this.iso);
        this.visualize(this.iso, node, this);
    }
}

function addActive(x, y) {
    if(!x) return false;

    removeActive(x);
    if(y.currentFocus >= x.length) y.currentFocus = 0;
    if(y.currentFocus < 0) y.currentFocus = x.length — 1;

    x[y.currentFocus].setAttribute('style', 'background—color:#aaa');
}

function removeActive(x) {
    for(var i = 0; i < x.length; i++) {
        x[i].setAttribute('style', 'background—color:none');
    }
}

function closeAllLists(ele) {
    var x = document.getElementsByClassName('auto—complete—items');
    for(var i = 0; i < x.length; i++) {
        if(ele != x[i] && ele != query)
            x[i].parentNode.removeChild(x[i]);
    }
}

function sth() {
}

document.addEventListener('click', function(e) {
    closeAllLists(e.target);
});
};

/* Create a dictaionary before visualization */
createDict: function(inf, inf2, txt, rt) {

    /* Convert the input to the lowercase before checking with the topic */
    txt = txt.toLowerCase()
    document.getElementById(rt.div.id + '_query').value = txt

    /* Create annex Dict */
    var totalAnnex = []
    for(var a = 0; a < inf2.length; a++) {
        var sTopic = inf2[a].subtopic
```

```
for(var b = 0; b < sTopic.length; b++) {  
    var ssTopic = sTopic[b].subtopic  
    for(var c = 0; c < ssTopic.length; c++) {  
  
        /* Push obj of annexs to the array */  
        totalAnnex.push({  
            name: ssTopic[c].name,  
            annex: 'A.' + ssTopic[c].annex,  
            ref: ssTopic[c].page,  
            sclause: ssTopic[c].control  
        })  
    }  
}  
  
/* Loop all levels to create the dictionary */  
var highlight  
var nodeList = []  
  
for(var i = 0; i < inf.length; i++) {  
    var name = inf[i].name  
    var number = inf[i].clause  
    var tname = number + ' ' + name  
    var child = []  
    var inside = inf[i].content  
  
    if(!rt.wordList.includes(tname))  
        rt.wordList.push(tname);  
  
    for(var j = 0; j < inside.length; j++) {  
        var sname = inside[j].sname  
        var snumber = inside[j].sclause  
        var stname = snumber + ' ' + sname  
        var schild = []  
        var inside2 = inf[i].content[j].annex  
  
        if(!rt.wordList.includes(stname))  
            rt.wordList.push(stname)  
  
        for(var k = 0; k < inside2.length; k++) {  
            var aname = inside2[k]  
  
            if(!rt.wordList.includes(aname))  
                rt.wordList.push(aname)  
  
            var aObject  
            for(var p = 0; p < totalAnnex.length; p++) {  
                if(totalAnnex[p].annex == aname) {  
                    aObject = totalAnnex[p]  
  
                    if(!rt.wordList.includes(totalAnnex[p].name))  
                        rt.wordList.push(totalAnnex[p].name)  
                    break  
                }  
            }  
        }  
    }  
}  
  
highlight = setHighlight(aname) || setHighlight(aObject.name)
```

```
schild.push({
    name: aname,
    children: [],
    value: 1,
    color: inf[i].clause,
    highlight: highlight,
    annex: true,
    annexObj: aObject
})
}

highlight = setHighlight(stname)
child.push({
    name: stname,
    children: schild,
    value: 1,
    color: inf[i].clause,
    highlight: highlight,
    annex: false,
    annexObj: null
})
}

highlight = setHighlight(tname);
nodeList.push({
    name: tname,
    children: child,
    value: 1,
    color: inf[i].clause,
    highlight: highlight,
    annex: false,
    annexObj: null
})
}

var node = {
    name: 'ISO27001',
    children: nodeList,
    value: 1,
    color: 1,
    highlight: false,
    annex: false,
    annexObj: null
}
}

return node

/*
 * Highlight function for highlighting texts */
function setHighlight(text) {
    if(text.toLowerCase().includes(txt) && txt != "")
        return true
    return false
}

sualize: function(inf, node, rt) {
    const color1 = '#9999';
```

```
const hVal = d3.scaleLinear()  
.domain([0, 7])  
.range([0, 270]);  
  
const sVal = d3.scaleLinear()  
.domain([0, 5])  
.range([80, 30]);  
  
const lVal = d3.scaleLinear()  
.domain([0, 5])  
.range([80, 40]);  
  
const width = 1000;  
const height = 1000;  
const highColor = '#ffff1a';  
  
const pack = inf => d3.pack()  
.size([width, height])  
.padding(3)(d3.hierarchy(inf)  
.sum(d => d.value)  
.sort((a, b) => b.value - a.value))  
  
/* Cut the node for child app */  
var root = pack(node);  
if(rt.isChildApp) {  
    for(var i = 0; i < root.descendants().length; i++) {  
        if(root.descendants()[i].data.name == rt.setFocus) {  
            root = root.descendants()[i];  
            break;  
        }  
    }  
}  
  
let focus = root;  
let view;  
  
d3.select(rt.element).select('g').remove()  
  
const svg = d3.select(rt.element).append('svg')  
.attr('viewBox', `-${width/2} ${height/2 - 50} ${width} ${height}`)  
.style('display', 'block')  
.style('margin', '0 -14px')  
.style('background', color1)  
.style('cursor', 'pointer')  
.on('click', function() {  
    return zoom(root);  
})  
  
const vnode = svg.append('g')  
.selectAll('circle')  
.data(root.descendants().slice(1))  
.join('circle')  
.attr('fill', d => d.children ? `hsl(${hVal(d.data.color)}, ${sVal(d.depth)}%, ${lVal(d.depth)}%)` : 'white')  
.attr('stroke', d => d.data.highlight ? highColor : null)  
.attr('stroke-width', d => d.data.highlight ? 3 : 1)  
.on('mouseover', function() {
```

```
d3.select(this).attr('stroke', '#000');
})
.on('mouseout', function(d) {
  if(!d.data.highlight)
    d3.select(this).attr('stroke', null);
  else {
    d3.select(this).attr('stroke', highColor);
    d3.select(this).attr('stroke-width', 3);
  }
})
.on('click', function(d) {
  if(d.data.annex == true) {
    var wBox = 200;
    var hBox = 600;

    var rWidth = window.innerWidth/2 - wBox/2;
    var rHeight = window.innerHeight/2 - hBox/2;

    /* If the popup is existed, do not create again */
    if(document.getElementById(rt.div.id + '_' + d.data.name) == null) {

      /* Create a close button at each popup */
      var cTopic = document.createElement('h5')
      cTopic.innerHTML = d.data.name
      cTopic.setAttribute('class', 'card-title')

      var cSubTopic = document.createElement('p')
      cSubTopic.innerHTML = d.data.annexObj.name
      cSubTopic.setAttribute('class', 'card-subtitle mb-2 text-muted')

      var cButton = document.createElement('button')
      cButton.setAttribute('class', 'card-link cButton')
      cButton.setAttribute('style', 'position: absolute; top:3px; right:3px; background-color:rgb(255,100,100);')
      cButton.innerHTML = 'X'
      cButton.id = 'close_' + rt.div.id + '_' + d.data.name

      var cDetail = document.createElement('p')
      cDetail.setAttribute('class', 'card-text')
      cDetail.innerHTML = '<b>Reference <br>(ISO/IEC27002:2013): </b><br>'
      cDetail.innerHTML += 'Page ' + d.data.annexObj.ref

      var cSubClause = document.createElement('p')
      cSubClause.setAttribute('class', 'card-text')
      cSubClause.innerHTML = '<b>Related Subclause <br>(ISO/IEC27001:2013): </b><br>'

      for(var i = 0; i < d.data.annexObj.sclause.length; i++) {
        cSubClause.innerHTML += d.data.annexObj.sclause[i]
        if(i != d.data.annexObj.sclause.length - 1)
          cSubClause.innerHTML += ','
      }

      var cCardBody = document.createElement('div')
      cCardBody.setAttribute('class', 'card-body')

      var cCard = document.createElement('span')
      cCard.setAttribute('class', 'card')
      cCard.setAttribute('style', 'position:fixed; width:18rem; top:${rt.pointerY}px; left:${rt.pointerX}px')
```

```
cCard.id = rt.div.id + '_' + d.data.name

cCardBody.appendChild(cTopic)
cCardBody.appendChild(cSubTopic)
cCardBody.appendChild(cDetail)
cCardBody.appendChild(cSubClause)
cCardBody.appendChild(cButton)

cCard.appendChild(cCardBody)

rt.element.appendChild(cCard)

/* Append popup window */
$('.card').draggable()
$('.card').css('border', '1px solid black')
$('.card').css('border-radius', '10px')

$('.cButton').css('border-radius', '5px')
$('.cButton').click(function() {
    var span = document.getElementById(this.id.slice(6))
    rt.element.removeChild(span)
})
}

/* Do not zoom out to the root after pressing an annex */
d3.event.stopPropagation()

} else {
    if(!rt.isChildApp) {
        rt.launchAppWithValues("zoompack", {
            sentNode: d.data.name
        }, 100, 100);
        d3.event.stopPropagation()
    } else {
        focus != d && (zoom(d), d3.event.stopPropagation());
    }
}
})

const path = svg.append('g')
.selectAll('path')
.data(root.descendants().slice(1))
.join('path')
.style('stroke-width', 0) // for:DEBUG
.style('stroke', '#ffff00') // for:DEBUG
.style('fill', 'none')
.attr('d', d => {
    return `M 0,${d.r} A ${d.r},${d.r} 0 1,1 0.01,${d.r}`
})
.attr('id', d => `path_${rt.div.id}_${d.data.name}`)

const label = svg.append('g')
.style('font', 'bold 1.3vw sans-serif')
.attr('pointer-events', 'none')
.attr('text-anchor', 'middle')
.selectAll('text')
.data(root.descendants())
.join('text')
```

```
.style('fill—opacity', d => {
    if(d.parent === root)
        return 1
    if(d.parent !== null && d.parent.parent === root)
        return 1
    return 0
})
.style('display', d => {
    if(d.parent === root)
        return 'inline'
    if(d.parent !== null && d.parent.parent === root)
        return 'inline'
    return 'none'
})
.attr('id', d => 'text_' + rt.div.id + '_' + d.data.name)
.attr('class', d => {
    if(d.depth == 2)
        return 'labelDoTop'
    else
        return 'labelNone'
})
.attr('data', d => d.data.name)
.text(d => d.data.annex || d.depth == 1 ? d.data.name : "")
.attr('font—size', d => {
    if(d.parent !== null && d.parent.parent === root) {
        if(d.depth == 2) return '0.4vw'
        else return '1.5vw'
    }
    return '1.5vw'
})
}

zoomTo([root.x, root.y, root.x - 21]);

const selectLabel = svg.selectAll('.labelDoTop').call(toTop)

function toTop(text) {
    text.each(function() {
        var txt = d3.select(this)
        var message = txt.attr('data')

        txt.append('textPath')
            .attr('xlink:href', '#path' + txt.attr('id').slice(4))
            .attr('startOffset', '50%')
            .attr('text—anchor', 'middle')
            .text(message)
    })
}

if(rt.isChildApp) {
    for(var i = 0; i < root.descendants().length; i++) {
        if(root.descendants()[i].data.name == rt.sentFocus) {
            focus != root.descendants()[i] && zoom(root.descendants()[i]);
            break;
        }
    }
}
```

```
function zoomTo(v) {
    const k = width / v[2];
    view = v;

    path.attr('transform', d => `translate(${(d.x - v[0]) * k}, ${ (d.y - v[1]) * k})`)
    path.attr('d', d => {
        return `M 0,${d.r*k} A ${d.r*k},${d.r*k} 0 1,1 0.01,${d.r*k}`
    })

    label.attr('transform', d => {
        if(d.data.annex || d.depth == 1)
            return `translate(${(d.x - v[0]) * k}, ${ (d.y - v[1]) * k})`
        else
            return 'translate(0, 0)'
    })

    vnode.attr('transform', d => `translate(${(d.x - v[0]) * k}, ${ (d.y - v[1]) * k})`,
    vnode.attr('r', d => d.r * k);
}

function zoom(d) {
    const focus0 = focus;
    focus = d;
    console.log("===== BEGIN =====")

    const transition = svg.transition()
    .duration(750)
    .tween('zoom', d => {
        const i = d3.interpolateZoom(view, [focus.x, focus.y, focus.r * 2]);
        return t => zoomTo(i(t));
    });

    label.filter(function(d) {
        if(d.parent !== null && d.parent.parent === focus || d.parent === focus)
            return true
        if(this.style.display == 'inline')
            return true
        else
            return false;
    })
    .transition(transition)
    .style('fill-opacity', d => {
        if(d.parent !== null && d.parent.parent === focus || d.parent === focus)
            return 1
        else
            return 0
    })
    .on('start', function(d) {
        if(d.depth == 3) {
            console.log('----- Start -----')
            console.log(focus)
            console.log(d)
        }
        if((d.parent !== null && d.parent.parent === focus) || d.parent === focus)
            this.style.display = 'inline'
        if(focus.depth != 1 && d.depth == 2)
            d3.select(this).attr('font-size', '0.4vw')
```

```
        })
        .on('end', function(d) {
            if(d.depth == 3) {
                console.log('----- End -----')
                console.log(focus)
                console.log(d)
            }
            if((d.parent !== null && d.parent.parent !== focus) && d.parent !== focus)
                this.style.display = 'none'
            if(focus.depth == 1 && d.depth == 2)
                d3.select(this).attr('font-size', '1.5vw')
        })
    },
    autocomplete: function(obj, list, rt) {
        var currentFocus

        /* Execute a function when someone writes in the text field */
        obj.addEventListener("input", function(e) {
            var a, b, i, j
            var val = this.value

            /* Close any already open lists of autocompleted values */
            closeAllLists()

            if(!val) return false
            currentFocus = -1

            /* Create a 'div' element that will contain the items (values) */
            a = document.createElement("div")
            a.setAttribute("id", rt.id + "autocomplete-list")
            a.setAttribute("class", "autocomplete-items")

            /* Append the 'div' element as a child of the autocomplete container */
            this.parentNode.appendChild(a)

            /* For each item in the array */
            for (i = 0, i < list.length; i++) {

                /* Check if the item starts with the same letters as the text field value */
                j = 0
                while(j < list[i].length) {
                    if(j == 0 || !list[i].substr(j - 1, 1).localeCompare(" "))
                        if (list[i].substr(j, val.length).toUpperCase() == val.toUpperCase()) {

                            /* Create a 'div' element for each matching element */
                            b = document.createElement("div")

                            /* Make the matching letters bold */
                            b.innerHTML = list[i].substr(0, j)
                            b.innerHTML += "<strong>" + list[i].substr(j, val.length) + "</strong>"
                            b.innerHTML += list[i].substr(j + val.length)

                            /* Insert a input field that will hold the current array item's value */
                            b.innerHTML += "<input type='hidden' value='" + list[i] + "'>"
                        }
                    j++
                }
            }
        })
    }
},
```

Copyright © Mahidol University

```
/* Execute a function when someone clicks on the item value (DIV element) */
b.addEventListener("click", function(e) {

    /* Insert the value for the autocomplete text field */
    obj.value = this.getElementsByTagName("input")[0].value

    /* Close the list of autocompleted values */
    closeAllLists()
})

a.appendChild(b)
break
}
}
j++
}
}
}

/* Execute a function presses a key on the keyboard */
obj.addEventListener("keydown", function(e) {
    var x = document.getElementById(rt.div.id + "autocomplete-list")
    if (x) x = x.getElementsByTagName("div")
    if (e.keyCode == 40) {
        currentFocus++
        addActive(x)
    } else if (e.keyCode == 38) {
        currentFocus--
        addActive(x)
    } else if (e.keyCode == 13) {
        e.preventDefault()
        if (currentFocus > -1) {
            if(x) x[currentFocus].click()
        }
    }
})

function addActive(x) {
    /* A function to classify an item as "active" */
    if(!x) return false

    /* Start by removing the "active" class on all items */
    removeActive(x)

    if (currentFocus >= x.length) currentFocus = 0
    if (currentFocus < 0) currentFocus = (x.length - 1)

    /* Add class "autocomplete-active" */
    x[currentFocus].setAttribute('style', 'padding: 10px; cursor: pointer; background-color: #ccc; border-bottom: 1px
        ↪ solid #d4d4d4')
}

function removeActive(x) {
    /* A function to remove the "active" class from all autocomplete items */
    for (var i = 0; i < x.length; i++) {
        x[i].setAttribute('style', 'padding: 10px; cursor: pointer; background-color: #fff; border-bottom: 1px solid #
            ↪ d4d4d4')
    }
}
```

```
}

function closeAllLists(elem) {
    /* Close all autocomplete lists in the document, except the one passed as an argument */
    var x = document.getElementsByClassName("autocomplete-items")
    for (var i = 0; i < x.length; i++) {
        if (elem != x[i] && elem != obj) {
            x[i].parentNode.removeChild(x[i])
        }
    }
}

/* Execute a function when someone clicks in the document */
document.addEventListener("click", function (e) {
    closeAllLists(e.target)
})
};

});
```

### B.3 Web Application

Listing B.4: Web Application for Sunburst Visualization

```
/* The container to contain all search bar */
var searchContainer = document.createElement('div');
searchContainer.setAttribute('style', 'position:sticky; top:0px; background-color:#666; padding: 5px');

/* Search bar for input a query */
var searchBar = document.createElement('input');
searchBar.setAttribute('type', 'text');
searchBar.setAttribute('placeholder', 'Please input a keyword');
searchBar.setAttribute('class', 'form-control');
searchBar.setAttribute('style', 'width:50%; display:inline-block; margin:0px 5px;');
searchBar.id = 'query';

/* Search button */
var search = document.createElement('input');
search.setAttribute('type', 'button');
search.setAttribute('value', 'Search');
search.setAttribute('class', 'btn btn-primary');
search.setAttribute('style', 'margin:0px 5px;');
search.id = 'search';

/* Clear button */
var clear = document.createElement('input');
clear.setAttribute('type', 'button');
clear.setAttribute('value', 'Clear');
clear.setAttribute('class', 'btn btn-danger');
clear.setAttribute('style', 'margin:0px 5px;');
clear.id = 'clear';

/* Blind color button */
var color = document.createElement('input');
color.setAttribute('type', 'button');
color.setAttribute('value', 'Color Style');
```

```
color.setAttribute('class', 'btn btn—dark');
color.setAttribute('style', 'margin:0px 5px;');
color.id = 'color';

/* Help button for open user manual */
var help = document.createElement('input');
help.setAttribute('type', 'button');
help.setAttribute('value', 'Help');
help.setAttribute('class', 'btn btn—info');
help.setAttribute('style', 'margin:0px 5px;');
help.id = 'help';

/* Append all elements to body */
searchContainer.appendChild(searchBar);
searchContainer.appendChild(search);
searchContainer.appendChild(clear);
searchContainer.appendChild(color);
searchContainer.appendChild(help);

document.body.appendChild(searchContainer);

/* Setup for user manual pop—up */
var modalContainer = document.createElement('div');
modalContainer.setAttribute('style', 'display: none; position: fixed; z-index: 1; padding-top: 100px; left: 0; top: 0; width: 100%; height: 100%; overflow: auto; background-color: rob(0,0,0); background-color: rgba(0,0,0,0.4);');

var modalContent = document.createElement('div');
modalContent.setAttribute('style', 'max-width: 1000px; background-color: #fefefe; margin: auto; padding: 20px; border: 1px solid #888; width: 80%;');

/* Close button for user manual */
var closeModal = document.createElement('div');
closeModal.setAttribute('class', 'close');
closeModal.id = "closeModal";
closeModal.innerHTML = "&times;";

var slideContainer = document.createElement('div');
slideContainer.setAttribute('style', 'margin:auto; position:relative');

/* User manual content for picture 1 */
var slide1 = document.createElement('div');
slide1.setAttribute('class', 'mySlides');
var slide1Num = document.createElement('div');
slide1Num.setAttribute('class', 'numbertext');
slide1Num.innerHTML = '1/3';
var slide1Pic = document.createElement('img');
slide1Pic.setAttribute('src', 'manual/manual Sunburst—1.png');
slide1Pic.setAttribute('style', 'width:100%');
slide1.appendChild(slide1Num);
slide1.appendChild(slide1Pic);

/* User manual content for picture 2 */
var slide2 = document.createElement('div');
slide2.setAttribute('class', 'mySlides');
var slide2Num = document.createElement('div');
slide2Num.setAttribute('class', 'numbertext');
slide2Num.innerHTML = '2/3';
```

```
var slide2Pic = document.createElement('img');
slide2Pic.setAttribute('src', 'manual/manual Sunburst—2.png');
slide2Pic.setAttribute('style', 'width:100%');
slide2.appendChild(slide2Num);
slide2.appendChild(slide2Pic);

/* User manual content for picture 3 */
var slide3 = document.createElement('div');
slide3.setAttribute('class', 'mySlides');
var slide3Num = document.createElement('div');
slide3Num.setAttribute('class', 'numbertext');
slide3Num.innerHTML = '3/3';
var slide3Pic = document.createElement('img');
slide3Pic.setAttribute('src', 'manual/manual Sunburst—3.png');
slide3Pic.setAttribute('style', 'width:100%');
slide3.appendChild(slide3Num);
slide3.appendChild(slide3Pic);

/* Button for previous page in User manual */
var prev = document.createElement('a');
prev.setAttribute('class', 'prev');
prev.onclick = function(){plusSlides(-1)};
prev.innerHTML = "\u2190";

/* Button for next page in User manual */
var next = document.createElement('a');
next.setAttribute('class', 'next');
next.onclick = function(){plusSlides(1)};
next.innerHTML = "\u2192";

slideContainer.appendChild(slide1);
slideContainer.appendChild(slide2);
slideContainer.appendChild(slide3);
slideContainer.appendChild(prev);
slideContainer.appendChild(next);

modalContent.appendChild(closeModal);
modalContent.appendChild(slideContainer);
modalContainer.appendChild(modalContent);
document.body.appendChild(modalContainer);

d3.select(document.body)
.append('svg')
.attr('id', 'partitionSVG');

main();

/* Set listener of each button */
$('#search').click(function() {
    main();
})

$('#clear').click(function() {
    clearAll();
})
```

```
$('#color').click(function(){
    if(this.getAttribute('class') == 'btn btn—dark'){
        this.setAttribute('class', 'btn btn—warning');
        main();
    }
    else{
        this.setAttribute('class', 'btn btn—dark');
        main();
    }
})

$('#help').click(function(){
    modalContainer.style.display = "block";
})

$('#closeModal').click(function(){
    modalContainer.style.display = "none";
})

window.onclick = function(event) {
    if(event.target == modalContainer)
        modalContainer.style.display = "none";
}

var wordList = [];
var iso27001, iso27002;
var slideIndex = 1;
showSlides(slideIndex);

// Read JSON file and use them to create visualization
function main(){
    $.getJSON('./data/ISO27002.json', data => {
        iso27002 = createDictionary27002(data);
        $.getJSON('./data/ISO27001.json', data => {
            iso27001 = createDictionary27001(data);
            wordList.sort();
            autocomplete(document.getElementById("query"), wordList);
            createSunBurst(iso27001);
        });
    });
}

// Popup function
function clearAll() {
    document.getElementById('query').value = "";
    var del = document.getElementsByTagName('span')
    while(del.length > 0)
        document.body.removeChild(del[0])
}

main();
}

// Close the pop-up window that have the same id as input parameter
function closePopup(id) {
    var tmp = document.getElementById(id)
    document.body.removeChild(tmp)
}
```

```
// Create data dictionary from ISO/IEC 27002:2013
function createDictionary27002(data) {
    var nodeList = [];
    for(i = 0; i < data.length; i++) {
        var name = data[i].name;
        var number = data[i].annex;
        var child = [];

        for(j = 0; j < data[i].subtopic.length; j++) {
            var sname = data[i].subtopic[j].name;
            var snumber = data[i].subtopic[j].annex;
            var schild = [];

            for(k = 0; k < data[i].subtopic[j].subtopic.length; k++) {
                var ssname = data[i].subtopic[j].subtopic[k].name;
                var ssnumber = data[i].subtopic[j].subtopic[k].annex;
                var page = data[i].subtopic[j].subtopic[k].page;
                var control = data[i].subtopic[j].subtopic[k].control;

                schild.push({
                    name: ssname,
                    annex: ssnumber,
                    control: control,
                    page: page
                });
            }
            child.push({
                name: snumber + " " + sname,
                children: schild
            });
        }
        nodeList.push({
            name: number + " " + name,
            children: child
        });
    }
    var data = {
        name: 'ISO27002: 2013',
        children: nodeList
    };
    return data;
}

// Create data dictionary from ISO/IEC 27001:2013
function createDictionary27001(data) {
    var txt = document.getElementById('query').value.toLowerCase();
    var checked = true;
    var nodeList = [];

    // In case there is no query
    if(txt.length == 0){ checked = false; }
    for(i = 0; i < data.length; i++) {
        var name = data[i].name;
```

```
var number = data[i].clause;
var child = [];
if(!wordList.includes(name)) wordList.push(name);
for(j = 0; j < data[i].content.length; j++) {
    var sname = data[i].content[j].sname;
    var snumber = data[i].content[j].sclause;
    var schild = [];
    var cHighlight = false;

    if(!wordList.includes(sname)) wordList.push(sname);
    for(k = 0; k < data[i].content[j].annex.length; k++) {
        var annex = data[i].content[j].annex[k].split(".");
        var aname = iso27002.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1].children[parseInt(
            ↗ annex[3])-1].name;
        var highlight = false;
        var a = data[i].content[j].annex[k];
        if(setHighlight(aname) || setHighlight(data[i].content[j].annex[k])) highlight = true;
        var size = 1;

        if(!wordList.includes(aname)) {
            wordList.push(aname);
            wordList.push(data[i].content[j].annex[k]);
        }
        schild.push({
            name: a,
            size: size,
            color: data[i].clause,
            highlight: highlight
        });
        if((highlight)) cHighlight = true;
    }
    highlight = setHighlight(sname);
    child.push({
        name: snumber + " " + sname,
        children: schild,
        color: data[i].clause,
        highlight: highlight,
        childHighlight: cHighlight
    })
}
highlight = setHighlight(name);
nodeList.push({
    name: number + " " + name,
    children: child,
    color: data[i].clause,
    highlight: highlight
})
}
var data = {
    name: 'ISO/IEC 27001:2013',
    color: 1,
    children: nodeList
}
return data;

function setHighlight(text){
    if(checked){
```

```
        if(text.toLowerCase().includes(txt)) { return true; }
    }
    return false;
}

// Create Sunburst visualization
function createSunBurst(data){
    d3.select('#partitionSVG').selectAll("*").remove(); // Clear the old visualization
    const format = d3.format(",d");
    const width = 2000; // Visualization width
    const height = 2000; // Visualization height
    const radius = width / 6; // Visualization radius
    const arc = d3.arc()
        .startAngle(d => d.x0)
        .endAngle(d => d.x1)
        .padAngle(d => Math.min((d.x1 - d.x0) / 2, 0.005))
        .padRadius(radius * 1.5)
        .innerRadius(d => d.y0 * radius)
        .outerRadius(d => Math.max(d.y0 * radius, d.y1 * radius - 1));

    const partition = data => {
        const root = d3.hierarchy(data)
            .sum(d => d.size)
        return d3.partition()
            .size([2 * Math.PI, root.height + 1])(root);
    };

    const root = partition(data);

    // Set up color HSL value
    var hVal = d3.scaleLinear().domain([0, 7]).range([0, 170])
    var sVal = d3.scaleLinear().domain([0, 5]).range([100, 50])
    var lVal = d3.scaleLinear().domain([0, 7]).range([-80, 40])

    if(document.getElementById('color').getAttribute('class') == 'btn btn-warning'){
        sVal = d3.scaleLinear().domain([0, 5]).range([0, 0]);
        lVal = d3.scaleLinear().domain([0, 7]).range([0, 90]);
    }

    root.each(d => d.current = d);

    const svg = d3.select('#partitionSVG')
        .attr("viewBox", [0, 0, width, width])
        .style("font", "20px sans-serif");

    const g = svg.append("g")
        .attr("transform", `translate(${width / 2},${width / 2})`);

    // Create the node for all element
    const path = g.append("g")
        .selectAll("path")
        .data(root.descendants().slice(1))
        .join("path")
        .attr("fill", d => d.depth > 0 ? `hsl(${hVal(d.data.color)}, ${sVal(d.depth)}%, ${lVal(d.data.color - 4)}%)` : 'white')
        .attr("fill-opacity", d => arcVisible(d.current) ? (d.children ? 0.6 : 0.4) : 0)
        .attr("d", d => arc(d.current))
}
```

```
.attr('stroke', d => arcVisible(d.current) && (d.data.highlight || d.data.childHighlight) ? "#F00" : null)
.attr('stroke-width', d => arcVisible(d.current) && (d.data.highlight || d.data.childHighlight) ? 3 : 0)
.on("click", function (d){
    // Create pop-up function
    var wBox = 300 // popup width
    var hBox = 220 // popup height
    if(arcVisible(d.current)){
        if(document.getElementById(d.data.name) == null) {
            var pTop = d3.event.clientY
            var pLeft = d3.event.clientX
            var annex = d.data.name.split(".");
            var aname = iso27002.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1]
                .children[parseInt(annex[3])-1].name;
            var subClause = iso27002.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1]
                .children[parseInt(annex[3])-1].control;
            var page = iso27002.children[parseInt(annex[1])-5].children[parseInt(annex[2])-1]
                .children[parseInt(annex[3])-1].page;

            /* Create the popup title */
            var cTopic = document.createElement('h5')
            cTopic.innerHTML = d.data.name
            cTopic.setAttribute('style', 'margin: 5px')

            /* Create the popup close button */
            var cButton = document.createElement('button')
            cButton.setAttribute('class', 'cButton')
            cButton.innerHTML = 'X'
            cButton.id = 'close_' + d.data.name

            /* Create the detail in the popup */
            var cDetail = document.createElement('p')
            cDetail.innerHTML = aname
            cDetail.setAttribute('style', 'margin: 5px; color: gray')

            var cRef = document.createElement('p')
            cRef.innerHTML = "<b>Reference</b><br>ISO/IEC 27002: 2013 Page " + page;
            cRef.setAttribute('style', 'margin: 5px')

            var cSubClause = document.createElement('p')
            cSubClause.setAttribute('class', 'card-text')
            cSubClause.innerHTML = "<b>Related Subclause</b>(ISO/IEC27001:2013)<br>"
            cSubClause.setAttribute('style', 'margin: 5px')

            for(var i = 0; i < subClause.length; i++) {
                cSubClause.innerHTML += subClause[i]
                if(i != subClause.length - 1)
                    cSubClause.innerHTML += ','

            }

            var cContainer = document.createElement('span')
            cContainer.setAttribute('class', 'drag')
            cContainer.setAttribute('style', 'top:' + pTop + 'px;left:' + pLeft + 'px;')
            cContainer.id = d.data.name
            cContainer.appendChild(cTopic)
            cContainer.appendChild(cButton)
            cContainer.appendChild(cDetail)
            cContainer.appendChild(cRef)
        }
    }
})
```

```
cContainer.appendChild(cSubClause)

var validate = document.body.appendChild(cContainer)
$('.drag').draggable()
$('.drag').css('width', `${wBox}px`)
$('.drag').css('height', `${hBox}px`)
$('.drag').css('position', 'fixed')
$('.drag').css('border', '1px solid black')
$('.drag').css('border-radius', '10px')
$('.drag').css('background-color', 'white')

$('.cButton').css('background-color', 'rgb(255,100,100)')
$('.cButton').css('position', 'absolute')
$('.cButton').css('top', '3px')
$('.cButton').css('right', '3px')
$('.cButton').css('border-radius', '5px')
$('.cButton').click(function() {
    var span = document.getElementById(this.id.slice(6))
    document.body.removeChild(span)
})
} else alert(d.data.name + ' is already selected')

});

path.filter(d => d.children)
.style("cursor", "pointer")
.on("click", clicked);

path.append("title")
.text(d => `${d.data.name}`);

const label = g.append("g")
.attr("pointer-events", "none")
.attr("text-anchor", "middle")
.style("user-select", "none")
.selectAll("text")
.data(root.descendants().slice(1))
.join("text")
.attr("dy", "0.3em")
.attr("fill-opacity", d => +labelVisible(d.current))
.attr("transform", d => labelTransform(d.current))
.style("font", d => d.children == null ? "25px sans-serif" : "20px sans-serif")
.text(d => shorterName(d.data.name))

// Parent node element
const parent = g.append("g")
.datum(root)
.on("click", clicked);

const parentTitle = parent.append("title")
.text(d => d.parent == null ? "ISO/IEC 27001:2013" : `${d.parent.data.name}`)

const parentCircle = parent.append("circle")
.attr("r", radius)
.attr("fill", "none")
.attr("pointer-events", "all");
```

```
const parentText = parent.append("text")
    .attr("text-anchor", "middle")
    .style("font", "40px sans-serif")
    .text("ISO/IEC 27001:2013");

const backSign = parent.append("text")
    .attr("text-anchor", "middle")
    .attr("dy", "4em")
    .attr("font-size", "40px")
    .text(parentText.text == "ISO/IEC 27001:2013" ? "\u2B8C" : "");

/* Wrap Case 2: Wrap function */
function wrap(text, len) {
    text.each(function() {
        var txt = d3.select(this)
        var words = txt.text().split(/\s+/).reverse()
        var word
        var line = []
        var lineNumber = 0
        var lineHeight = 1.1

        tspan = txt.text(null).append('tspan')
            .attr('x', 0)
            .attr('y', 0)
            .attr('dy', 0 + 'em')

        while(word = words.pop()) {
            line.push(word)
            tspan.text(line.join(" "))
            if(tspan.node().getComputedTextLength() > len) {
                line.pop()
                tspan.text(line.join(" "))
                line = [word]
                lineNumber++
                var tmp = lineNumber * lineHeight
                console.log('Multiplication:')
                console.log(tmp)
                span = txt.append('tspan')
                    .attr('x', 0)
                    .attr('y', 0)
                    .attr('dy', `${tmp}em`)
                    .text(word)
            }
        }
    })
}

// Transition all nodes when user clicks on the node
function clicked(p) {
    parent.datum(p.parent || root);
    parentText.text(p.data.name);
    parentTitle.text(p.data.name);
    if(p != root) backSign.text("\u2B8C");
    wrap(parentText, 650)

    root.each(d => d.target = {
```

```
x0: Math.max(0, Math.min(1, (d.x0 — p.x0) / (p.x1 — p.x0))) * 2 * Math.PI,  
x1: Math.max(0, Math.min(1, (d.x1 — p.x0) / (p.x1 — p.x0))) * 2 * Math.PI,  
y0: Math.max(0, d.y0 — p.depth),  
y1: Math.max(0, d.y1 — p.depth)  
});  
  
const t = g.transition().duration(750);  
  
// Transition the data on all arcs, even the ones that aren't visible,  
// so that if this transition is interrupted, entering arcs will start  
// the next transition from the desired position.  
path.transition(t)  
  .tween("data", d => {  
    const i = d3.interpolate(d.current, d.target);  
    return t => d.current = i(t);  
  })  
  .filter(function (d) {  
    return +this.getAttribute("fill—opacity") || arcVisible(d.target);  
  })  
  .attr("fill—opacity", d => arcVisible(d.target) ? (d.children ? 0.5 : 0.4) : 0)  
  .attr("stroke", d => arcVisible(d.target) && (d.data.highlight || d.data.childHighlight) ? "#F00" : null)  
  .attr("stroke—width", d => arcVisible(d.target) && (d.data.highlight || d.data.childHighlight) ? 3 : 0)  
  .attrTween("d", d => () => arc(d.current));  
  
label.filter(function (d) {  
  return +this.getAttribute("fill—opacity") || labelVisible(d.target);  
}).transition(t)  
  .attr("fill—opacity", d => +labelVisible(d.target))  
  .attrTween("transform", d => () => labelTransform(d.current))  
};  
  
function arcVisible(d) {  
  return d.y1 <= 3 & & d.y0 >= 1 && d.x1 > d.x0;  
}  
  
function labelVisible(d) {  
  return d.y1 <= 3 && d.y0 >= 1 && (d.y1 — d.y0) * (d.x1 — d.x0) > 0.03;  
}  
  
function shorterName(d){  
  if(d.length>30){  
    return d.substring(0, 30) + "..."  
  }  
  return d;  
}  
  
function labelTransform(d) {  
  const x = (d.x0 + d.x1) / 2 * 180 / Math.PI;  
  const y = (d.y0 + d.y1) / 2 * radius;  
  return `rotate(${x — 90}) translate(${y},0) rotate(${x < 180 ? 0 : 180})`;  
}  
  
function autocomplete(inp, arr) {  
  /*the autocomplete function takes two arguments,  
  */
```

```
the text field element and an array of possible autocompleted values:*/
var currentFocus;
/*execute a function when someone writes in the text field:*/
inp.addEventListener("input", function(e) {
    var a, b, i, j, val = this.value;
    /*close any already open lists of autocompleted values*/
    closeAllLists();
    if (!val) { return false; }
    currentFocus = —1;
    /*create a DIV element that will contain the items (values):*/
    a = document.createElement("DIV");
    a.setAttribute("id", this.id + "autocomplete—list");
    a.setAttribute("class", "autocomplete—items");
    /*append the DIV element as a child of the autocomplete container:*/
    this.parentNode.appendChild(a);
    /*for each item in the array...*/
    for (i = 0; i < arr.length; i++) {
        /*check if the item starts with the same letters as the text field value:*/
        j = 0;
        while(j < arr[i].length){
            if(j == 0 || !arr[i].substr(j—1, 1).localeCompare(" ")) {
                if (arr[i].substr(j, val.length).toUpperCase() == val.toUpperCase()) {
                    /*create a DIV element for each matching element:*/
                    b = document.createElement("DIV");
                    /*make the matching letters bold:*/
                    b.innerHTML = arr[i].substr(0, j);
                    b.innerHTML += "<strong>" + arr[i].substr(j, val.length) + "</strong>";
                    b.innerHTML += arr[i].substr(j + val.length);
                    /*insert a input field that will hold the current array item's value:*/
                    b.innerHTML += "<input type='hidden' value='" + arr[i] + "'>";
                    /*execute a function when someone clicks on the item value (DIV element)*/
                    b.addEventListener("click", function(e) {
                        /*insert the value for the autocomplete text field:*/
                        inp.value = this.getElementsByTagName("input")[0].value;
                        /*close the list of autocompleted values,
                        (or any other open lists of autocompleted values:*/
                        closeAllLists();
                    });
                    a.appendChild(b);
                    break;
                }
            }
            j++;
        }
    });
    /*execute a function presses a key on the keyboard:*/
    inp.addEventListener("keydown", function(e) {
        var x = document.getElementById(this.id + "autocomplete—list");
        if (x) x = x.getElementsByTagName("div");
        if (e.keyCode == 40) {
            /*If the arrow DOWN key is pressed,
            increase the currentFocus variable:*/
            currentFocus++;
            /*and make the current item more visible:*/
            addActive(x);
        }
    });
});
```

```
    } else if (e.keyCode == 38) { //up
        /*If the arrow UP key is pressed,
        decrease the currentFocus variable.*/
        currentFocus--;
        /*and and make the current item more visible*/
        addActive(x);
    } else if (e.keyCode == 13) {
        /*If the ENTER key is pressed, prevent the form from being submitted.*/
        e.preventDefault();
        if (currentFocus > -1) {
            /*and simulate a click on the "active" item*/
            if (x[currentFocus].click());
        }
    });
}

function addActive(x) {
    /*a function to classify an item as "active":*/
    if (!x) return false;
    /*start by removing the "active" class on all items:*/
    removeActive(x);
    if (currentFocus >= x.length) currentFocus = 0;
    if (currentFocus < 0) currentFocus = (x.length - 1);
    /*add class "autocomplete-active":*/
    x[currentFocus].setAttribute('style', 'padding: 10px; cursor: pointer; background-color: #ccc; border-bottom: 1px solid #d4d4d4');
}

function removeActive(x) {
    /*a function to remove the "active" class from all autocomplete items:*/
    for (var i = 0; i < x.length; i++) {
        x[i].setAttribute('style', 'padding: 10px; cursor: pointer; background-color: #fff; border-bottom: 1px solid #d4d4d4');
    }
}

function closeAllLists(elmnt) {
    /*close all autocomplete lists in the document,
    except the one passed as an argument.*/
    var x = document.getElementsByClassName("autocomplete-items");
    for (var i = 0; i < x.length; i++) {
        if (elmnt != x[i] && elmnt != x[0]) {
            x[i].parentNode.removeChild(x[i]);
        }
    }
}

/*execute a function when someone clicks in the document:*/
document.addEventListener("click", function (e) {
    closeAllLists(e.target);
});

function plusSlides(n) {
    showSlides(slideIndex += n);
}

function currentSlide(n) {
    showSlides(slideIndex = n);
}

function showSlides(n) {
```

```
var i;
var slides = document.getElementsByClassName("mySlides");
var dots = document.getElementsByClassName("dot");
if (n > slides.length) {slideIndex = 1}
if (n < 1) {slideIndex = slides.length}
for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
}
slides[slideIndex - 1].style.display = "block";
}
```

Listing B.5: Web Application for Zoomable Circle Packing Visualization

```
/* Keep body in a variable */
const body = document.body
body.setAttribute('style', 'margin:0px')

/* The container to contain all search bar */
var searchContainer = document.createElement('div')
searchContainer.setAttribute('style', 'position:sticky; top:0px; background-color:#666; padding: 5px')

/* Search bar for input a query */
var searchBar = document.createElement('input')
searchBar.setAttribute('type', 'text')
searchBar.setAttribute('placeholder', 'Please input a keyword')
searchBar.setAttribute('class', 'form-control')
searchBar.setAttribute('style', 'width:50%; display:inline-block; margin:0px 5px;')
searchBar.id = 'query'

/* Search button */
var search = document.createElement('input')
search.setAttribute('type', 'button')
search.setAttribute('value', 'Search')
search.setAttribute('class', 'btn btn-primary')
search.setAttribute('style', 'margin:0px 5px;')
search.id = 'search'

/* Clear button */
var clear = document.createElement('input')
clear.setAttribute('type', 'button')
clear.setAttribute('value', 'Clear')
clear.setAttribute('class', 'btn btn-danger')
clear.setAttribute('style', 'margin:0px 5px;')
clear.id = 'clear'

/* Create a help button */
var help = document.createElement('input')
help.setAttribute('type', 'button')
help.setAttribute('value', 'Help')
help.setAttribute('class', 'btn btn-success')
help.setAttribute('style', 'margin:0px 5px;')
help.id = 'help'

/* Create manual popup */
var manualContainer = document.createElement('div')
manualContainer.setAttribute('style', 'width:30%; position:fixed; border:5px solid #555;')
```

```
manualContainer.id = 'manualContainer'
manualContainer.style.backgroundColor = 'black'
manualContainer.style.visibility = 'hidden'

var manualTitleBar = document.createElement('h3')
manualTitleBar.style.backgroundColor = 'black'
manualTitleBar.style.color = 'white'
manualTitleBar.style.textAlign = 'center'
manualTitleBar.innerHTML = 'Manual'

var manualCloseBtn = document.createElement('button')
manualCloseBtn.setAttribute('style', 'position:absolute; top:3px; right:3px; background-color:rgb(255,100,100); border-radius: 50%; width: 10px; height: 10px;')
manualCloseBtn.innerHTML = 'X'
manualCloseBtn.id = 'manualClose'

var manual = document.createElement('div')
manual.setAttribute('class', 'carousel slide')
manual.setAttribute('data-ride', 'carousel')
manual.id = 'manual'

var manualOL = document.createElement('ol')
manualOL.setAttribute('class', 'carousel-indicators')

var manualLI0 = document.createElement('li')
manualLI0.setAttribute('data-target', '#manual')
manualLI0.setAttribute('data-slide-to', '0')
manualLI0.setAttribute('class', 'active')

var manualLI1 = document.createElement('li')
manualLI1.setAttribute('data-target', '#manual')
manualLI1.setAttribute('data-slide-to', '1')

var manualLI2 = document.createElement('li')
manualLI2.setAttribute('data-target', '#manual')
manualLI2.setAttribute('data-slide-to', '2')

manualOL.appendChild(manualLI0)
manualOL.appendChild(manualLI1)
manualOL.appendChild(manualLI2)

var manualDIV = document.createElement('div')
manualDIV.setAttribute('class', 'carousel-inner')

var manualIN0 = document.createElement('div')
manualIN0.setAttribute('class', 'carousel-item active')

var manualIN1 = document.createElement('div')
manualIN1.setAttribute('class', 'carousel-item')

var manualIN2 = document.createElement('div')
manualIN2.setAttribute('class', 'carousel-item')

var manualIMG0 = document.createElement('img')
manualIMG0.setAttribute('src', './img/manv21.png')
manualIMG0.setAttribute('class', 'd-block w-100')
```

```
var manualIMG1 = document.createElement('img')
manualIMG1.setAttribute('src', './img/manv22.png')
manualIMG1.setAttribute('class', 'd-block w-100')

var manualIMG2 = document.createElement('img')
manualIMG2.setAttribute('src', './img/manv23.png')
manualIMG2.setAttribute('class', 'd-block w-100')

manualIN0.appendChild(manualIMG0)
manualIN1.appendChild(manualIMG1)
manualIN2.appendChild(manualIMG2)

manualDIV.appendChild(manualIN0)
manualDIV.appendChild(manualIN1)
manualDIV.appendChild(manualIN2)

var manualPREV = document.createElement('a')
manualPREV.setAttribute('class', 'carousel-control-prev')
manualPREV.setAttribute('href', '#manual')
manualPREV.setAttribute('role', 'button')
manualPREV.setAttribute('data-slide', 'prev')

var manualNEXT = document.createElement('a')
manualNEXT.setAttribute('class', 'carousel-control-next')
manualNEXT.setAttribute('href', '#manual')
manualNEXT.setAttribute('role', 'button')
manualNEXT.setAttribute('data-slide', 'next')

var manualSPAN0PREV = document.createElement('span')
manualSPAN0PREV.setAttribute('class', 'carousel-control-prev-icon')
manualSPAN0PREV.setAttribute('aria-hidden', 'true')

var manualSPAN1PREV = document.createElement('span')
manualSPAN1PREV.setAttribute('class', 'sr-only')
manualSPAN1PREV.innerHTML = 'Previous'

var manualSPAN0NEXT = document.createElement('span')
manualSPAN0NEXT.setAttribute('class', 'carousel-control-next-icon')
manualSPAN0NEXT.setAttribute('aria-hidden', 'true')

var manualSPAN1NEXT = document.createElement('span')
manualSPAN1NEXT.setAttribute('class', 'sr-only')
manualSPAN1NEXT.innerHTML = 'Previous'

manualPREV.appendChild(manualSPAN0PREV)
manualPREV.appendChild(manualSPAN1PREV)
manualNEXT.appendChild(manualSPAN0NEXT)
manualNEXT.appendChild(manualSPAN1NEXT)

manual.appendChild(manualOL)
manual.appendChild(manualDIV)
manual.appendChild(manualPREV)
manual.appendChild(manualNEXT)

manualContainer.appendChild(manualTitleBar)
manualContainer.appendChild(manualCloseBtn)
manualContainer.appendChild(manual)
```

```
/* Create the main div to contain the visualization */
var main = document.createElement('div')

/* Append all elements to body */
searchContainer.appendChild(searchBar)
searchContainer.appendChild(search)
searchContainer.appendChild(clear)
searchContainer.appendChild(help)

body.appendChild(searchContainer)
body.appendChild(manualContainer)
body.appendChild(main)

/* Add event listener to the body */
body.addEventListener('keydown', event => {
    /* ENTER button do like press search button */
    if(event.keyCode == 13) {
        var txt = document.getElementById('query').value
        mainFn(txt)
    }
    /* ESC button do like press clear button */
    else if(event.keyCode == 27) {
        var del = document.getElementsByTagName('span')
        while(del.length > 0)
            body.removeChild(del[0])
        mainFn('')
    }
})

/* Set listener of each button */
$('#search').click(function() {
    var txt = document.getElementById('query').value
    mainFn(txt)
})

$('#clear').click(function() {
    var del = document.getElementsByTagName('span')
    for(var i = del.length - 1; i >= 0; i--) {
        if(del[i].getAttribute('typePopup') == 'popupAnnex') {
            body.removeChild(del[i])
        }
    }
    var man = document.getElementById('manualContainer')
    man.style.visibility = 'hidden'
    mainFn('')
})

$('#help').click(function() {
    var man = document.getElementById('manualContainer')
    man.style.visibility = 'visible'
})

$('#manualClose').click(function() {
    var man = document.getElementById('manualContainer')
    man.style.visibility = 'hidden'
})
```

```
$( '#manualContainer' ).draggable()

/* Run main at the first time */
mainFn()

/* For saveing data from the files */
var iso27001, iso27002
var wordList = []

/* The function for */
function mainFn( txt ) {
    /* Read both files */
    $.getJSON( './data/ISO27002.json', data2 => {
        $.getJSON( './data/ISO27001.json', data1 => {
            var query = document.getElementById('query')
            var node = createDict( data1, data2, txt )
            wordList.sort()
            autocomplete( query, wordList )
            visualize( data1, node )
        })
    })
}

/* Create a function to create an overall object */
function createDict( data, anx, txt ) {
    /* Get the value from the search bar */
    txt = txt.toLowerCase()
    document.getElementById('query').value = txt

    /* Create objects for annex */
    var totalAnnex = []
    for( var a = 0; a < anx.length; a++ ) {
        var sTopic = anx[a].subtopic
        for( var b = 0; b < sTopic.length; b++ ) {
            var ssTopic = sTopic[b].subtopic
            for( var c = 0; c < ssTopic.length; c++ ) {
                /* Push obj of annexes to the array */
                totalAnnex.push({
                    name: ssTopic[c].name,
                    annex: 'A.' + ssTopic[c].annex,
                    ref: ssTopic[c].page,
                    clause: ssTopic[c].control
                })
            }
        }
    }

    /* Create a variable for keeping highlight value */
    var highlight

    var nodeList = []
    for( var i = 0; i < data.length; i++ ) {
        var name = data[i].name
        var number = data[i].clause
        var tname = number + ' ' + name
    }
}
```

```
var child = []
var inside = data[i].content

/* Append the name into the autocomplete list */
if(!wordList.includes(tname))
    wordList.push(tname)

for(var j = 0; j < inside.length; j++) {
    var sname = inside[j].scname
    var snumber = inside[j].sclause
    var stname = snumber + ' ' + sname
    var schild = []
    var inside2 = data[i].content[j].annex

    /* Append the name into the autocomplete list */
    if(!wordList.includes(stname))
        wordList.push(stname)

    for(var k = 0; k < inside2.length; k++) {
        var aname = inside2[k]

        /* Append the name into the autocomplete list */
        if(!wordList.includes(aname))
            wordList.push(aname)

        var aObject
        for(var p = 0; p < totalAnnex.length; p++) {
            if(totalAnnex[p].annex == aname) {
                aObject = totalAnnex[p]

                /* Append the annex description if there is the annex in the system */
                if(!wordList.includes(totalAnnex[p].name))
                    wordList.push(totalAnnex[p].name)
                break
            }
        }

        /* Can search from annex's name or annex's number */
        highlight = setHighlight(aObject.name) || setHighlight(aname)
        child.push({
            name: aname,
            children: [],
            value: 1,
            color: data[i].clause,
            highlight: highlight,
            annex: true,
            annexObj: aObject
        })
    }
}

highlight = setHighlight(stname)
child.push({
    name: stname,
    children: schild,
    value: 1,
    color: data[i].clause,
    highlight: highlight,
```

Copyright

```
        annex: false,
        annexObj: null
    })
}

highlight = setHighlight(tname)
nodeList.push({
    name: tname,
    children: child,
    value: 1,
    color: data[i].clause,
    highlight: highlight,
    annex: false,
    annexObj: null
})
}

var node = {
    name: 'ISO27001',
    children: nodeList,
    value: 1,
    color: 1,
    highlight: false,
    annex: false,
    annexObj: null
}

/* Return node to create a visualization */
return node

/* For set highlight data in each node */
function setHighlight(t) {
    if(t.toLowerCase().includes(txt) && tx != '')
        return true
    return false
}

/* Create a function to visualize the object */
function visualize(data, node) {
    /* Set a background color of the visualization */
    const bgColor = '#999'

    /* Set colors of each node */
    const hVal = d3.scaleLinear()
        .domain([0, 7])
        .range([0, 270])

    const sVal = d3.scaleLinear()
        .domain([0, 5])
        .range([80, 30])

    const lVal = d3.scaleLinear()
        .domain([0, 5])
        .range([80, 40])

    /* Set default size of the visualization */
}
```

```
const width = 1000
const height = 1000
const highColor = '#fffffa'

const pack = data => d3.pack()
  .size([width, height])
  .padding(3)(d3.hierarchy(data)
    .sum(d => d.value)
    .sort((a, b) => {
      return b.value - a.value
    }))
  var root = pack(node)

let focus = root
let view

/* Delete the last visualization */
d3.select('svg').remove()

/* Append all of the visualization to the container in <body> */
const svg = d3.select(main).append('svg')
  .attr('viewBox', `-${width/2} ${height/2} ${width} ${height}`)
  .style('display', 'block')
  .style('margin', '0 -14px')
  .style('background', bgColor)
  .style('cursor', 'pointer')
  .on('click', function() {
    /* If click at the same point or click area --> go back to all figure */
    return zoom(root)
  })
  /* Create a circle (node) from each descendant */
  const vnode = svg.append('g')
    .selectAll('circle')
    .data(root.descendants().slice(1))
    .join('circle')
    .attr('fill', d => d.children ? `hsl(${lVal(d.data.color)}, ${sVal(d.depth)}%, ${lVal(d.depth)}%)` : 'white')
    .attr('stroke', d => d.data.highlight ? highColor : null)
    .attr('stroke-width', d => d.data.highlight ? 3 : 1)
    .on('mouseover', function() {
      d3.select(this).attr('stroke', '#000')
    })
    .on('mouseout', function(d) {
      if(d.data.highlight)
        d3.select(this).attr('stroke', null)
      else {
        d3.select(this).attr('stroke', highColor)
        d3.select(this).attr('stroke-width', 3)
      }
    })
    .on('click', function(d) {
      if(d.data.annex == true) {
        /* Set the size of the popup */
        const wBox = 200
        const hBox = 250
      }
    })
  })
```

```
/* Set the position of the popup */
const rWidth = window.innerWidth/2 - wBox/2
const rHeight = window.innerHeight/2 - hBox/2

if(document.getElementById(d.data.name) == null) {
    var pTop = d3.event.clientY
    var pLeft = d3.event.clientX

    var nTopic = document.createElement('h5')
    nTopic.setAttribute('class', 'card—title')
    nTopic.innerHTML = d.data.name

    var nSubTopic = document.createElement('h6')
    nSubTopic.setAttribute('class', 'card—subtitle mb—2 text—muted')
    nSubTopic.innerHTML = d.data.annexObj.name

    var nDetail = document.createElement('p')
    nDetail.setAttribute('class', 'card—text')
    nDetail.innerHTML = '<b>Reference<br>(ISO/IEC27002:2013)</b><br>' + 'nDetail.innerHTML += 'Page ' + d.data.annexObj.ref

    var nSubClause = document.createElement('p')
    nSubClause.setAttribute('class', 'card—text')
    nSubClause.innerHTML = '<b>Related Subclause<br>(ISO/IEC_27001:2013)</b><br>'

    for(var i = 0; i < d.data.annexObj.sclause.length; i++) {
        nSubClause.innerHTML += d.data.annexObj.sclause[i]
        if(i != d.data.annexObj.sclause.length - 1)
            nSubClause.innerHTML += ','
    }

    var nClose = document.createElement('button')
    nClose.setAttribute('class', 'card—link nClose')
    nClose.setAttribute('style', 'position: absolute; top:3px; right:3px; background-color:rgb(255,100,100);')
    nClose.innerHTML = 'X'
    nClose.id = 'close_' + d.data.name

    var nCardBody = document.createElement('div')
    nCardBody.setAttribute('class', 'card—body cButton')

    var nCard = document.createElement('span')
    nCard.setAttribute('class', 'card')
    nCard.setAttribute('style', 'position:fixed; width:18rem; top:${pTop}px; left:${pLeft}px;')
    nCard.setAttribute('typePopUp', 'popupAnnex')
    nCard.id = d.data.name

    nCardBody.appendChild(nTopic)
    nCardBody.appendChild(nSubTopic)
    nCardBody.appendChild(nDetail)
    nCardBody.appendChild(nSubClause)
    nCardBody.appendChild(nClose)

    nCard.appendChild(nCardBody)

    body.appendChild(nCard)

    $('.card').draggable()
```

```
$('.card').css('border', '1px solid black')
$('.card').css('border-radius', '10px')

$('.nClose').css('border-radius', '5px')
$('.nClose').click(function() {
    var span = document.getElementById(this.id.slice(6))
    body.removeChild(span)
})

/*
 * Do not zoom to the root if press annex *
d3.event.stopPropagation()
} else {
    focus !== d && (zoom(d), d3.event.stopPropagation());
}
})

/* Create a path to place the text labels */
const path = svg.append('g')
.selectAll('path')
.data(root.descendants().slice(1))
.join('path')
.style('stroke-width', 0)
.style('stroke', '#ffff00')
.style('fill', 'none')
.attr('d', d => {
    return 'M ' + (0) + ',' + (d.r) + ' A ' + (d.r) + ',' + (d.r) + ' 0 1,' + (0.0) + ',' + (d.r)
})
.attr('id', d => 'P_' + d.data.name)

/* Text label of each node */
const label = svg.append('g')
.style('font', 'bold 18px sans-serif')
.attr('pointer-events', 'none')
.attr('text-anchor', 'middle')
.selectAll('text')
.data(root.descendants())
.join('text')

.style('fill-opacity', d => {
    if(d.parent === root)
        return 1
    if(d.parent !== null && d.parent.parent === root)
        return 1
    return 0
})
.style('display', d => {
    if(d.parent === root)
        return 'inline'
    if(d.parent !== null && d.parent.parent === root)
        return 'inline'
    return 'none'
})
.attr('id', d => '#T_' + d.data.name)
.attr('class', d => {
    if(d.depth == 2)
        return 'labelDoTop'
```

```
        else
            return 'labelNone'
    })
    .attr('data', d => d.data.name)
    .text(d => d.data.annex || d.depth == 1 ? d.data.name : '')
    .attr('font-size', d => {
        if(d.parent !== null && d.parent.parent === root)
            return '4.5px'
        return '16px'
    })
}

zoomTo([root.x, root.y, root.r*2])

/* Proceed a function on each label to place it along the path */
const selectLabel = svg.selectAll('.labelDoTop')
.call(toTop)

function toTop(text) {
    text.each(function() {
        var txt = d3.select(this)
        var message = txt.attr('data')

        txt.append('textPath')
        .attr('xlink:href', '#P' + txt.attr('id').slice(2))
        .attr('startOffset', '50%')
        .style('text-anchor', 'middle')
        .text(message)
    })
}

/* Translate path, circle, label while zooming */
function zoomTo(v) {
    const k = width / v[2]
    view = v

    label.attr('transform', d => {
        if(d.data.annex || d.depth == 1)
            return `translate(${(d.x - v[0]) * k}, ${(d.y - v[1]) * k})`
        else
            return 'translate(0, 0)'
    })
    path.attr('transform', d => `translate(${(d.x - v[0]) * k}, ${(d.y - v[1]) * k})`)
    path.attr('cx', d => {
        return 'M ' + (0)*k + ',' + (d.r)*k + ' A ' + (d.r)*k + ',' + (d.r)*k + ' 0 1,1 ' + (0.01)*k + ',' + (d.r)*k
    })
    vnode.attr('transform', d => `translate(${(d.x - v[0]) * k}, ${(d.y - v[1]) * k})`)
    vnode.attr('r', d => d.r * k)
}

/* Zooming function */
function zoom(d) {
    const focus0 = focus
    focus = d

    const transition = svg.transition()
    .duration(d3.event.altKey ? 7500 : 750)
    .tween('zoom', d => {
```

```
const i = d3.interpolateZoom(view, [focus.x, focus.y, focus.r*2])
return t => {
    zoomTo(i(t))
}
})

label.filter(function(d) {
    if(d.parent === null && d.parent.parent === focus || d.parent === focus)
        return true
    if(this.style.display === 'inline')
        return true
    return false
})
.transition(transition)
.style('fill-opacity', d => {
    if(d.parent === null && d.parent.parent === focus || d.parent === focus)
        return 1
    return 0
})
.on('start', function(d) {
    if(d.parent === null && d.parent.parent === focus || d.parent === focus)
        this.style.display = 'inline'
    if(focus.depth != 1 && d.depth == 2)
        d3.select(this).attr('font-size', '4.5px')
})
.on('end', function(d) {
    if(d.parent === null && d.parent.parent != focus && !parent)
        this.style.display = 'none'
    if(focus.depth == 1 && d.depth == 2)
        d3.select(this).attr('font-size', '8px')
})
}

/*
Autocomplete function in the search bar */
function autocomplete(obj, lists) {
    var currentFocus

    /* Execute a function when someone writes in the text field */
    obj.addEventListener("input", function(e) {
        var a, b, i, j
        var val = this.value

        /* Close any already open lists of autocompleted values */
        closeAllLists()

        if(!val) return false
        currentFocus = -1

        /* Create a 'div' element that will contain the items (values) */
        a = document.createElement("div")
        a.setAttribute("id", "autocomplete-list")
        a.setAttribute("class", "autocomplete-items")

        /* Append the 'div' element as a child of the autocomplete container */
        this.parentNode.appendChild(a)
```

```
/* For each item in the array */
for (i = 0; i < list.length; i++) {

    /* Check if the item starts with the same letters as the text field value */
    j = 0
    while(j < list[i].length) {
        if(j == 0 || !list[i].substr(j - 1, 1).localeCompare(" ")) {
            if (list[i].substr(j, val.length).toUpperCase() == val.toUpperCase()) {

                /* Create a 'div' element for each matching element */
                b = document.createElement("div")

                /* Make the matching letters bold */
                b.innerHTML = list[i].substr(0, j)
                b.innerHTML += "<strong>" + list[i].substr(j, val.length) + "</strong>"
                b.innerHTML += list[i].substr(j + val.length)

                /* Insert a input field that will hold the current array item's value */
                b.innerHTML += "<input type='hidden' value='" + list[i] + "'>"

                /* Execute a function when someone clicks on the item value (Div element) */
                b.addEventListener("click", function(e) {

                    /* Insert the value for the autocomplete text field */
                    obj.value = this.getElementsByTagName("input")[0].value

                    /* Close the list of autocompleted values */
                    closeAllLists()
                })
                a.appendChild(b)
                break
            }
        }
        j++
    }
}

/* Execute a function press : a key on the keyboard */
obj.addEventListener("keydown", function(e) {
    var x = document.getElementById("autocomplete-list")
    (x) x = x.getElementsByTagName("div")
    if (e.keyCode == 40) {
        currentFocus++
        addActive(x)
    } else if (e.keyCode == 38) {
        currentFocus--
        addActive(x)
    } else if (e.keyCode == 13) {
        e.preventDefault()
        if (currentFocus > -1) {
            if(x) x[currentFocus].click()
        }
    }
})

function addActive(x) {
```

```
/* A function to classify an item as "active" */
if(!x) return false

/* Start by removing the "active" class on all items */
removeActive(x)

if (currentFocus >= x.length) currentFocus = 0
if (currentFocus < 0) currentFocus = (x.length — 1)

/* Add class "autocomplete—active" */
x[currentFocus].setAttribute('style', 'padding: 10px; cursor: pointer; background—color: #ccc; border—bottom: 1px
    ↪ solid #d4d4d4')
}

function removeActive(x) {
    /* A function to remove the "active" class from all autocomplete items */
    for (var i = 0; i < x.length; i++) {
        x[i].setAttribute('style', 'padding: 10px; cursor: pointer; background—color: #fff; border—bottom: 1px solid #d4d4d4')
    }
}

function closeAllLists(elem) {
    /* Close all autocomplete lists in the document, except the one passed as an argument */
    var x = document.getElementsByClassName("autocomplete—items")
    for (var i = 0; i < x.length; i++) {
        if (elem != x[i] && elem != obj) {
            x[i].parentNode.removeChild(x[i])
        }
    }
}

/* Execute a function when someone clicks in the document */
document.addEventListener("click", function (e) {
    closeAllLists(e.target)
})
```

## REFERENCES

- [1] Marr B., “How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read”; [cited 11 Nov. 2019].
- [2] Security R., “Request the 2019 Q1 Data Breach QuickView Report”; [cited 11 Nov. 2019].
- [3] Rouse M., “What is information security management system (ISMS)? - Definition from WhatIs.com”; [cited 11 Nov. 2019].
- [4] “NIST SP 800-53 Rev. 4 - NIST Special Publication 800-53 Revision 4”, Security and Privacy Controls for Federal Information Systems and Organizations; 2015. 4, [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-53r4>.
- [5] “COBIT 2019”, ISACA; . , [Online]. Available: <http://www.isaca.org/COBIT/Pages/default.aspx>.
- [6] “ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems— Requirements”, ISO/IEC; 2013. Edition 2, [Online]. Available: <https://www.iso.org/standard/54534.html>.
- [7] “ISO/IEC 27002:2013 Information technology — Security techniques — Code of practice for information security controls”, ISO/IEC; 2013. Edition 2, [Online]. Available: <https://www.iso.org/standard/54533.html>.
- [8] Marco Angelini GS. Simone Lenti. “CRUMBS: A cyber security framework browser”, IEEE; 2017, [Online]. Available: [10.1109/VIZSEC.2017.8062194](https://doi.org/10.1109/VIZSEC.2017.8062194).
- [9] Robert Dale AHPLMWCGACSDMM. David D. Palmer, Classical Approaches, Nitin Indurkhyia FJD., . editor. 978-1-4200-8593-8. Taylor and Francis Group; 2010.

- [10] Thomas K Landauer DL Peter W Foltz;.
- [11] Rubayyi Alghamdi KA., “A Survey of Topic Modeling in Text Mining”, International Journal of Advanced Computer Science and Applications. Jan 2015;(1), [Online]. Available: <http://dx.doi.org/10.14569/IJACSA.2015.060121>.
- [12] Christine K. Mulunda LM. Peter W. Wagacha. “Review of Trends in Topic Modeling Techniques, Tools, Inference Algorithms and Applications”, IEEE; 2018, [Online]. Available: <https://doi.org/10.1109/ISCMCI.2018.8703231>.
- [13] David M. Blei MIJ. Andrew Y. Ng, “Latent Dirichlet Allocation”, Journal of Machine Learning Research. Mar 2003;(30), [Online]. Available: <http://dl.acm.org/citation.cfm?id=944919.944937>.
- [14] David M. Blei JDL.. “Correlated Topic Models”, Cambridge MA, USA: MIT Press; 2005, [Online]. Available: <http://dl.acm.org/citation.cfm?id=2976248.2976267>.
- [15] Ilya Malyanov SL. Brian J. d'Auriol, “Visualization experience and related process modeling”, Journal of Visual Languages and Computing. March 2013;(4), [Online]. Available: <http://dx.doi.org/10.1016/j.jvlc.2013.03.001>.
- [16] Wenwen Dou XWZMWR. Li Yu, “HierarchicalTopics: Visually Exploring Large Text Collections Using Topic Hierarchies”, IEEE Transactions on Visualization and Computer Graphics. Dec 2013;(12), [Online]. Available: [10.1109/TVCG.2013.162](https://doi.org/10.1109/TVCG.2013.162).
- [17] “Nureva™ Wall WM408i For expansive visual collaboration”, Nureva Inc.;, [Online]. Available: <https://www.nureva.com/hubfs/product-resource/visual-collaboration/nureva-visual-collaboration-solutions-wm408i-wall-product-sheet.pdf?hsCtaTracking=51125ea3-962d-4e41-a2ca-d2652889a273%7C3981dc2c-d047-46c6-97da-0790d41fd2eb>.
- [18] PRYSM, “Interactive Displays | Prysm LPD 6K | Visual Collaboration”; [cited 13 Nov. 2019].

- [19] Thomas Marrinan ANKBVMLRLLAJL. Jillian Aurisano. “SAGE2: A new approach for data intensive collaboration using Scalable Resolution Shared Displays”;
- [20] Bostock M., “D3.js - Data-Driven Documents”; [cited 15 May. 2020].
- [21] “spaCy · Industrial-strength Natural Language Processing in Python”; [cited 15 May. 2020].
- [22] “scikit-learn: machine learning in Python — scikit-learn 0.23.0 documentation”; [cited 15 May. 2020].

## BIOGRAPHIES

<b>NAME</b>	Miss Boonrak Kaewnopparat
<b>DATE OF BIRTH</b>	26 February 1998
<b>PLACE OF BIRTH</b>	Bangkok, Thailand
<b>INSTITUTIONS ATTENDED</b>	Satriwithaya School, 2015: High School Diploma Mahidol University, 2020: Bachelor of Science (ICT)
<b>NAME</b>	Mr. Nasorn Niampradit
<b>DATE OF BIRTH</b>	19 January 1998
<b>PLACE OF BIRTH</b>	Bangkok, Thailand
<b>INSTITUTIONS ATTENDED</b>	Suankularb Wittayalai, 2015: High School Diploma Mahidol University, 2020: Bachelor of Science (ICT)
<b>NAME</b>	Mr. Chayapon Puakalong
<b>DATE OF BIRTH</b>	11 July 1998
<b>PLACE OF BIRTH</b>	Lopburi, Thailand
<b>INSTITUTIONS ATTENDED</b>	Suankularb Wittayalai, 2015: High School Diploma Mahidol University, 2020: Bachelor of Science (ICT)