UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS Honours
Final Paper 2021

Title: **COCE- A Concept Map InfoVis Creation Tool**

Author: **Laaiqah Hassim**

Project Abbreviation: **MASH**

Supervisor(s): **Professor Hussein Suleman**

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	15
System Development and Implementation	*0*	*20*	15
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

COCE

A Concept Map InfoVis Creation Tool

Laaiqah Hassim
Computer Science Department
University of Cape Town
Cape Town, South Africa
hsslaa002@myuct.ac.za

ABSTRACT

The progression of digitizing cultural heritage acts as an efficient mechanism for ensuring the long-term preservation of information and knowledge on cultural heritage while making it more accessible to students, researchers, and the community. With even more data stored in digital archives, it is imperative to investigate new cultural heritage information visualization methods, to allow researchers, or anyone of interest, to create relationships or illustrations with information, for further analysis. This paper proposes a desktop web application as a tool for creating user-generated, digital concept maps to visualize information and the relationships between them. User evaluations suggested that there is a great potential for a concept map creation tool, given the prototype is further refined and more advanced features are introduced.

KEYWORDS

Information visualization; Cultural Heritage Preservation; Concept Map Creation; Web-based tool

1 INTRODUCTION

South Africa is rich in culture, with a population that is multi-racial, multi-cultural, and multi-lingual. Culture is continuously changing, and it is critical to document and protect its historical link to the past. There are effective mechanisms for preserving cultural heritage, such as digital archives and digital libraries, where interested visitors can access it at any time and location. However, with the constant addition of information and digitally stored data, new methods for visualizing data are required.

Academics produce scholarly articles and other artifacts, such as timelines, diagrams, and other complex representations of knowledge, to link information for further analysis. With the evolution of digital libraries, it would be ideal for researchers to produce and interact or view and preserve created artifacts in a digital form. As a result, methods for exploring data visualization by creating virtual exhibitions for viewing and displaying digital content emerged [22].

There are tools that exist to create exhibits of this data, for example, content management systems (CMS's) or diagramming software. Omeka is a popular CMS used to visualize cultural heritage information in the form of user-created websites. However, once created the site cannot be exported into a format that allows it to be preserved back into an archive. Diagramming software such as draw.io [3] or LucidChart [13] allows created mappings of information to be exported but in a static representation. It would be favorable for researchers to produce and store interactive mappings of data.

The limitations of existing approaches to visualize cultural heritage knowledge are discussed further in section 2.

1.1 Aims and Objectives

This study aims to answer the following research question:

What is the experience of users when using an integrated concept mapping and complex object creation tool to create and edit complex digital objects?

This study presents the implementation and evaluation of an experimental concept mapping system that allows users to create, edit and visualize digital content and the relationships between them. The system is a web application with two main components: Complex Object Creator and Editor (COCE) and Complex Object Mappings Renderer (COMR). The entire system is called MASH; however, this paper focuses on the COCE component.

As its name suggests, COCE handles the creation and editing of complex objects and COMR handles the rendering of the concept map. In the context of this study, a complex object consists of a map with nodes storing digital content and the relationships (or edges) between them.

1.2 Report Structure

The purpose of this report is to discuss the COCE. It begins by explaining the background to motivate this study (section 2). Subsequently, the report describes the implementation of COCE (section 3), followed by the methods used to evaluate the prototype (section 4) and its results and findings (section 5), and ending with conclusions, which includes future work (section 6).

2 BACKGROUND

2.1 Cultural Heritage Archives

Cultural heritage digital archive's principal function is to preserve digital cultural objects. Digital objects are the data stored in archives, controlled by rules that govern the interaction of the digital objects and their repositories. This set of rules is called a Repository Access Protocol [1].

Metadata describes digital objects. It structures the data referring to digital objects, permitting discovery and usage of the content in digital collections and repositories. An example of a popular metadata standard is Dublin Core [10]. It is a descriptive metadata standard and is one of the most popular metadata formats because of its simplicity [7]. Digital cultural objects stored in these archives are heterogeneous, so its digital object types include text, image, video, and audio [22]. Dublin Core standardization is prevalent in cultural heritage archives because it can accommodate heterogeneous cultural heritage digital object types [9]. Often, a combination of metadata standards is used in an archive to attain more digital objects types.

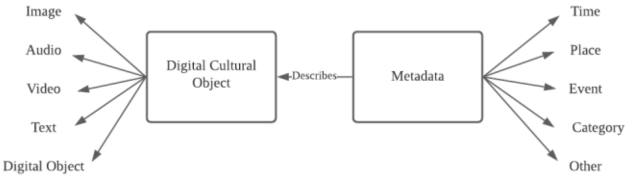


Figure 1. Relationship between digital cultural object and its metadata.

Since digital archive's prime function is to preserve data, other specialized systems such as content management systems or diagramming software are necessary to create exhibits or associations to visualize digital object data stored in archives.

2.2 Content Management Systems

Content management systems (CMS) handle and visualize the content of digital objects. It provides a method of creating websites and displaying digital data without the need to write HTML web pages from scratch– essentially allowing any non-technical user to “create” websites [17]. Three popular CMS used to visualize cultural heritage information are Omeka, Islandora, and Drupal. The ultimate difference between these CMS is the way they manage their digital objects.

Drupal is an open-source CMS. Islandora is a tool that integrates Drupal (for user interface), Fedora (for managing digital objects), and Solr (for indexing) [2,8,12,15]. Omeka has free or paid plans and gives users an option to either install it or use a hosted version; however, Drupal does not offer complete hosting, so users are required to download the software and develop the site locally [2,15].

Islandora and Omeka can store any digital object type with any metadata standard, whereas Drupal offers various, but limited, metadata standards and digital object types [2,8].

A mentionable difference between Drupal and Omeka is usability. Although both are highly interoperable and customizable, making them ideal for integrating into other systems, Drupal has a steep learning curve making it difficult for those who have never built a site on Drupal before [2].

Omeka’s user-friendly platforms allow people with a range of expertise to use it easily [15]. It has a vast range of plugins to extend its functionality. In terms of cultural heritage exhibitions, plugins include maps and timelines [15].

These CMS are a great way to allow users to publish content into sites to create or visualize collections of data. However, the sites created from CMS cannot be exported into a self-contained archival object that can be preserved back into a digital archive.

2.3 Diagramming Software

Another tool to visualize cultural heritage digital content is diagramming software. Relationships between content can be depicted using diagrams created with software such as Draw.io [3] or LucidChart [13].

Outputs produced from these include mind maps, tree diagrams, and network diagrams. Unlike CMS its outputs can be exported into different formats such as pdf or png formats, and preserved back into archives. The limitation of these tools is that the outputs they produce are static representations of data. Another limitation is the amount of data stored in these diagrams. Large amounts of information will cluster the visual and render it unreadable.

A way to combat this is by creating two kinds of pages, one with a map and one with details of a node (one for each node on a map). Nodes on maps will contain embedded web links, and once clicked, will navigate to another page with detailed information. It can be cumbersome; users may feel reluctant to do this. Our application eliminates this hassle by handling the interactivity and rendering of maps and only requires users to input the data they would like to visualize.

2.4 Approaches to Visualizing Cultural Heritage Data

There are many cultural heritage visualization tools, each using different kinds of user interfaces and methods. A vast majority of cultural heritage visualization approaches rely on metadata, with many of them also including a visual representation of the digital content itself [22]. There are two broad categories of data collection for rendering visualizations in cultural heritage: temporal and non-temporal visualizations.

The notion of time is often studied with respect to cultural heritage, making temporal visualization a crucial approach when visualizing cultural heritage [22]. It involves collecting and displaying information relative to the time in the form of timelines and a time axis.

Non-temporal visualization has the most diverse visualization methods, consisting of hierarchical diagrams, maps, 3D visualizations, and networks [22]. A common type is the use of geospatial online databases to create dimensional data visualizations. It allows storing and displaying data related to positions on the earth’s surface [18]. This method is suitable for visualizing locations relative to cultural heritage; however, it is not easy to view relationship links between objects that are not located in the same area.

Both temporal and non-temporal can accommodate digital object content or its metadata.

3 PROTOTYPE IMPLEMENTATION

This section describes the design and implementation of COCE.

3.1 Approach and Design considerations

The COCE utilized an agile methodology: a lightweight iterative development cycle [11]. Due to its flexibility during iterative cycles and its ability to adapt previous features to account for changes and modifications, it acts as the most suitable software development methodology. Paired with agile development was a feature-driven development (FDD) approach, where one feature is developed at a time [11]. Considering the time constraint of the project, FDD allowed the rapid development of a working prototype in a short period, where every 2 to 10 days a working feature is produced [14]. This is favorable for the short life cycle of the project.

3.1.1 Target User

When building a system, it is critical to consider its intended user and future use, or it might not meet a user's need or level of experience [22]. The COCE is not limited to accommodate users from a specific academic background. It was designed to allow users with any technical experience to use and effortlessly master. Its design is simple, and tasks should be easy to execute without instructions on how to perform them.

3.2 Architecture

Figure 1 depicts the model view controller (MCV) pattern that influences the basic architecture for COCE. The model stores and handles the data in the system by using a flat-file structured database called MongoDB. The view is responsible for the user interface of the system. Bootstrap, HTML, CSS, JS, and Jinja renders the view, which displays information from the model to the user. The controller serves as a midpoint between the model and view, allowing the view to access data from the model and the model to store information from the view for later retrieval. Flask, a micro web framework, handles the controller.

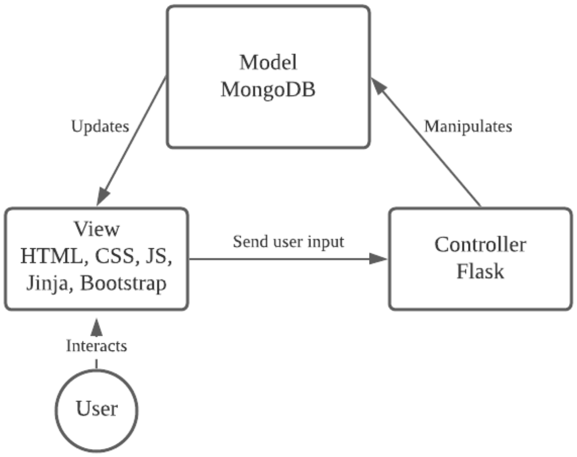


Figure 2. Architecture of COCE

3.3 System Features

The system comprises two main parts: Authentication and COCE. Authentication permits only users who created maps to access it, and COCE allows users to create and edit maps. These functionalities are explained further in Table 1.

Table 1. System Features

Authentication	A login system enables confidentiality by allowing users to register for an account before using the system so that the maps they create are kept private.
Dashboard	Display a list of user-created maps and a user-defined description for that map. From the dashboard, a user can click to edit an existing map or create a new one.
New Map	Allows a user to create a new concept map. A user is required to specify a map name and give it a description. A user creates nodes with related information and link nodes to form relationships.
Edit Map	Allow a user to edit a created map. One can edit a map name or description, a node title, description or details, or the links (edges) between two nodes.

3.4 COCE System Implementation

3.4.1 Visual Granularity

Visual granularity refers to the level of detail presented to the user when visualizing the information. This can range from viewing single artifacts to collections of cultural heritage information [22]. For the purpose of this study, we plan to investigate a cultural heritage visualization in the form of networks, more specifically, for creating concept maps. The COCE will accept and render information entered by the user pertaining to nodes and relationships. A user will be presented with an overview of a collection of related data linked with edges, and once they click a node, they should view the digital content represented by that node. The COCE will accept three types of node information, namely: Title, Description and Details. Figure 3 illustrates the map a user can create with nodes and relationships. Figure 4 displays the details of a node once clicked in the COCE.

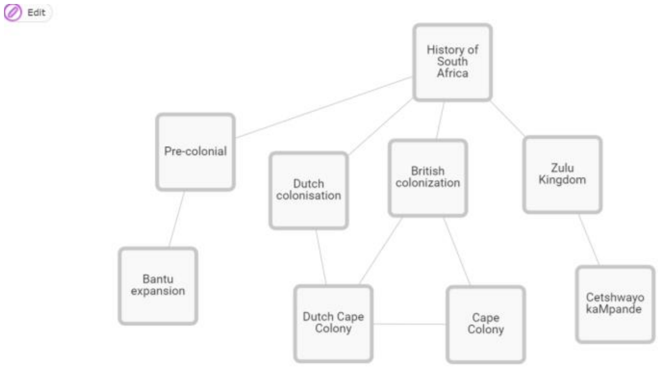
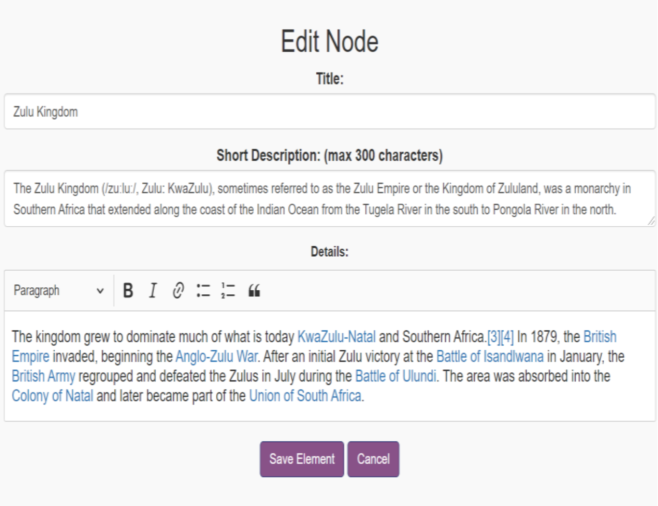


Figure 3. An editable concept map, displaying node titles and relationships to other nodes, from the COCE.



Edit Node

Title:

Zulu Kingdom

Short Description: (max 300 characters)

The Zulu Kingdom (/zu:lu:/, Zulu: KwaZulu), sometimes referred to as the Zulu Empire or the Kingdom of Zululand, was a monarchy in Southern Africa that extended along the coast of the Indian Ocean from the Tugela River in the south to Pongola River in the north.

Details:

The kingdom grew to dominate much of what is today KwaZulu-Natal and Southern Africa.[3][4] In 1879, the British Empire invaded, beginning the Anglo-Zulu War. After an initial Zulu victory at the Battle of Isandlwana in January, the British Army regrouped and defeated the Zulus in July during the Battle of Ulundi. The area was absorbed into the Colony of Natal and later became part of the Union of South Africa.

Figure 4. A bootstrap modal that pops up when a user clicks a node. It allows a user to view and edit details of a node, from the COCE.

3.4.2 Flask

Flask is a python microweb framework [4]. Frameworks help developers make easier, scalable, efficient, and maintainable web applications by providing extensions and reusable code for repeated operations. Flask framework has few dependencies on external libraries and is flexible, making it favorable to use because extensions and libraries can be added as a system needs [4]. Flask extensions used in COCE include Flask-Login and Flask-PyMongo. Flask also integrates with Werkzeug, a comprehensive Web Server Gateway Interface (WSGI) web application library [23].

3.4.3 Vis.js

Vis.js is a JavaScript, dynamic browser-based visualization library [21]. It is responsible for map creation and manipulation. COCE specifically uses the Network component of vis.js, to input and render information of nodes and edges, and the DataSet component, to store data in a JSON format. Together these results in a network with relationships [20].

Each node or edge, and its related information, are represented by JSON objects. JSON objects are key-value pairs that can store unstructured data. Global or local data can be stored in JSON objects. Global refers to all the nodes or edges in a map; local referring to a specific node or edge within a map. Node JSON objects store the details input by a user, such as a title, description, and details; additional information includes its unique node id and its x and y positions relative to the map canvas. Edge JSON objects store the node IDs of the two nodes it links with.

3.4.4 Security

The system uses Flask-Login, a Flask extension that facilitates user authentication [4]. It permits flask to manage users, by monitoring users' log-in status: it keeps track of a particular user that logged in and whether they remain logged in once the web application closes. This prevents unwanted access to the application and secures user-created data to remain confidential to that user. Only registered users can use the concept map tool.

Flask-Login handles the users and their access to the system; however, it needs a method to secure passwords saved from the user and stored in the database. A simple cryptographic hash algorithm, called SHA256, was used for this. SHA256 takes a string and produces an almost unique 256-bit signature that is saved in the database instead of the plaintext password [16]. When a user logs in, the password they entered is hashed using the SHA256 algorithm, and the unreadable 256-bit string associated with a username is fetched from the database and compared to the user input. If the strings are equal, a user gets logged in otherwise a message is displayed and a user has to try to log in again.

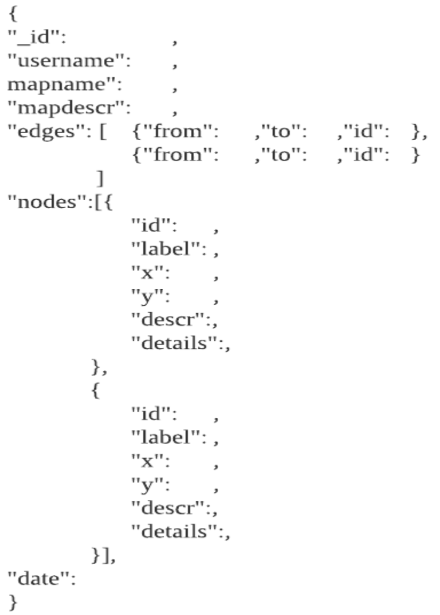
3.4.5 Flask-Pymongo

MongoDB was the database chosen to represent the model of the system to handle data. It is a scheme-less NoSQL database that stores data in unstructured key-value pair JSON documents.

MongoDB is a document-oriented database model, where each value stored in the database is called a document. Each document can differ, hence its notion of scheme-less. Each map can contain multiple nodes and edges specified by the user. MongoDB supports the unpredictable structure of data that comes from user-generated maps. The systems database consists of two collections of documents; user collection and map collection. Each user document stores the username, first name and the unreadable 256-bit string created from the plaintext password. Table 2 describes the data stored in a single map document and Figure 3 depicts its structure.

Table 2. Data stored in Map Document

_id	Unique ID given to identify this map.
username	Username of the user who created the map.
mapname	Title of the concept map.
mapdescr	Short (300 word) text to describe this map.
edges	A list of edges/links between nodes. This stores a list of edge json objects. Each edge object stores the node id of the two nodes it connects to and the unique ID given to the edge.
nodes	A list of nodes. This stores a list of edge json objects. Each node object stores the unique ID given to the node, a label that is displayed to the user in the concept map, x and y co-ordinates of the nodes position on the map canvas, a 300-word short description of the node and the detailed information of the node.
date	The date a map was created. It a map was edited this is replaced by the last date it was modified.



```
{
"_id":              ,
"username":      ,
mapname":       ,
"mapdescr":     ,
"edges": [    {"from":     ,"to":    ,"id":   },
              {"from":     ,"to":    ,"id":   }
         ]
"nodes":[{
              "id":     ,
              "label": ,
              "x":      ,
              "y":      ,
              "descr":,
              "details":,
         },
         {
              "id":     ,
              "label": ,
              "x":      ,
              "y":      ,
              "descr":,
              "details":,
         }],
"date":
}
```

Figure 5. General structure of a map document stored in MongoDB.

PyMongo is the MongoDB driver for python; it is a library that enables the interaction between MongoDB and Python. Flask-Pymongo is a wrapper around PyMongo that is used to access the database because it provides convenience helpers.

The MongoDB database was deployed on the cloud using Atlas. MongoDB Atlas is a cloud database created by the same developers of MongoDB. It manages the database on a cloud server specified by its user; we used AWS cloud service provider for database deployment.

We chose to deploy our database on the cloud, instead of creating one locally because our web application is hosted on Heroku, and Heroku is not suitable for persistent storage [6]. Heroku has an ephemeral filesystem, which means that any changes in the filesystem after a certain period are erased and reset to their initial state [6]. If the database was created locally, all data stored in the database would be erased after Heroku's reset time. Hence our reason for database deployment on Atlas.

4 EVALUATIONS

4.1 Standard User Evaluations

The standardized USE (Usefulness, Satisfaction, and Ease of Use) questionnaire was used to assess participant's experience when using the COCE [20]. This questionnaire evaluates the user's subjective usability of the app [5]. It measures usability in four categories: usefulness, ease of use, ease of learning, and satisfaction [5]. Each category comprised a list of statements that a user needed to rate on a scale of one (strongly disagree) to 5 (strongly agree). The USE questionnaire also asks the user to give three positive and three negative aspects of the system.

In addition to the USE questionnaire, participants were asked to state the feature they felt was the most and least useful and to give their overall feedback on their experience when using the COCE.

4.2 Participant Recruitment

A total of 25 participants were recruited to participate in the user evaluations. Participants consisted of UCT staff, UCT students, and non UCT affiliated participants, all with varying academic backgrounds. Participants were recruited by a combination of snowball sampling and by circulating invites and having interested participants contact us. There were no specific requirements for participants to be selected, as we were evaluating the tool with users from different specializations, refer to appendix A; however, users needed to have a laptop or desktop and a stable internet connection to perform the evaluations. Evaluations were conducted asynchronously in the form of an online questionnaire.

Potential participants were sent invitations that explained the purpose of the project and the evaluation process. If they were interested, they had to contact me, stating their interest to participate in the study. After obtaining informed consent, they were sent an email containing a link to a questionnaire and a reminder that they may opt-out of the study at any time if they wish. The questionnaire included a list of tasks that users had to complete using COCE, followed by a set of questions that users were required to answer once after completing the tasks. Section 4.1 described the questions.

4.3 Ethical Clearance

This study aims to evaluate an experimental concept mapping system, with participants to gain insight on the feasibility of this tool and its future uses. To use participants, ethics clearance needed to be obtained. An application to the Science Faculty Research Ethics Committee for research involving human subjects was completed, submitted, and approved. Subsequently, further steps were required to use UCT staff and students as participants. Application forms to have access to UCT staff and UCT students were sent to HR and DSA, respectively. Permission to use UCT staff and UCT students were achieved. Once approved participants could be recruited by informed consent.

5 RESULTS AND DISCUSSION

5.1 User Evaluation Results

This subsection summarizes the results obtained from the user evaluations. Table 3 summarizes the quantitative results from the USE questionnaire. It gives the average ratings of each item within one of the four categories of subjective usability: Usefulness, Ease of Learning, Ease of Use, and Satisfaction. Ratings are on a scale of 1 (Strongly disagree) to 5 (Strongly agree). Tables 4 and 5 summarize the qualitative results of the USE questionnaire: three negative and three positive aspects of the COCE. Tables 6 and 7 lists the most useful and least useful features, according to participants. This subsection rounds off with Table 8, which gives an analysis of the overall feedback given by each participant by classifying phrases of their feedback with specified codes.

Detailed results on how much participants rated each item in the USE questionnaire are illustrated in the pie charts in appendix B. The complete user feedback on three negative and positive aspects are grouped into themes in Appendix C. Feedback on the most and least useful features are grouped into themes in Appendix D.

Table 3. The average COCE USE Questionnaire ratings from 25 participants.

#	Category	Average
USEFULNESS		
1	It helps me be more effective.	3.60
2	It helps me be more productive.	3.60
3	It is useful.	4.20
4	It gives me more control over the activities in my life.	3.04
5	It makes the things I want to accomplish easier to get done.	3.43
6	It saves me time when I use it.	3.32
7	It meets my needs.	3.43
8	It does everything I would expect it to do.	4.13
Total Average:		3.59
	EASE OF USE	
1	It is easy to use	4.43
2	It is simple to use	4.57
3	It is user friendly.	4.30
4	It requires the fewest steps possible to accomplish what I want to do with it	4.13
5	It is flexible.	4.04
6	Using it is effortless.	3.96
7	I can use it without written instructions.	3.87
8	I don't notice any inconsistencies as I use it.	4.00
9	Both occasional and regular users would like it.	4.15
10	I can recover from mistakes quickly and easily.	4.41
11	I can use it successfully every time.	4.45
Total Average:		4.21
	EASE OF LEARNING	
1	I learned to use it quickly.	4.68
2	I easily remember how to use it.	4.60
3	It is easy to learn to use it.	4.68
4	I quickly became skillful with it.	4.25
Total Average:		4.55
	SATISFACTION	
1	I am satisfied with it.	3.84
2	I would recommend it to a friend.	3.56
3	It is fun to use.	3.71
4	It works the way I want it to work	3.68
5	It is wonderful.	3.56
6	I feel I need to have it.	2.68
7	It is pleasant to use.	3.92
Total Average:		3.56

Table 4. Summary of Main Negative Aspects
No customization of nodes (e.g., shapes) and edges (e.g., line types)
Too basic design – no appeal.
Unclear how to perform some tasks.
Compulsory node description and details – tedious.
Compatibility issues: some browsers create issues.
Not unique, not understanding full extent of utility.
Usability issues: zooming in and out, linking edges.

Table 5. Summary of Main Positive Aspects
Simple GUI is effective and easy to understand.
Easy to create, edit and remove maps, nodes and edges.
Can be used to visualize different kinds of information - fluid.
Can store large amount of information.
User friendly
Resulting maps are informative – serves its purpose.

Table 6. Summary of most useful features
Repositioning of nodes and edges once placed.
Versatility of information to store in nodes.
Storing large information in nodes.
Creating personal account to save work.
Dragging edge from node to node to connect.

Table 7. Summary of least useful features
Lack of additional features.
Compulsory description and details of nodes.
Description and details of nodes was repetitive.
One type of node and edge.
Clicking add edge and add node all the time.

Table 8. Overall Feedback on COCE

#	Feedback	Codes
1	Overall, very simple and easy to use, which made for a good experience.	
2	I think it looks promising but I would hope there is more to come.	Good experience
3	Great to mind map. Easy and efficient to use.	
4	It was something new and refreshing. It just needs some minor adjustments to make it more attractive. Overall, it is a tool that can be used within various fields of study.	Negative experience
5	Nice idea. Should be encouraged.	Suggestion
6	Very good experience	
7	It was a very unpleasant experience mainly due to poor design but the concept has a lot of potential	Versatile
8	This is a very easy to use and helpful tool that - with some improvements regarding	

	additional options - I could imagine making use of.	Good Concept
9	It needs further improvements but going in right direction.	
10	Not a very good experience.	
11	It is a pleasant website. I see how it can be useful but it is no use for me	Good Usability
12	Pleasant.	
13	It was quite easy to understand how to use but the only thing I was confused on was the edges.	More features
14	It was pleasant playing around with this feature. Not a lot of complaints, it easily does what it was made to do.	Unclear Concept
15	In general, the program did its job and served its purpose in creating the desired outputs. Not the biggest fan of the look and mundane tasks, but as a whole, good program, but needs minor refinements	
16	Satisfactory	
17	The feature is simple and easy to use. However, the website does not explain the potential on creating one of these maps.	
18	I think the tool was overall well designed and sublimely easy to use. Even without instructions it would be simple for a new user to pick up and use.	
19	I enjoyed using the website, very user friendly and straight forward.	
20	GREAT CONCEPT	
21	It was very to the point and very easy to use. While I would have wanted some extra oomph to the feature, it did its job and that's all that matters in this case.	
22	This feature was actually very helpful and easy to use	
23	It is a positive experience, if I ever need to make a similar chart, I will use this.	
24	I think the websites and the features are very unique and handy, I haven't used anything like this before. I also think it will be useful in many different scenarios from studying to making a family tree.	
25	It was convenient to use and time saving	

5.2 Usability Evaluations

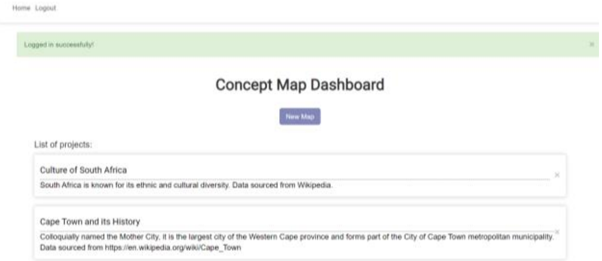
The USE questionnaire evaluated the usability of COCE in four categories, namely Usefulness (U), Ease of Use (EU), Ease of Learning (EL) and Satisfaction (S).

EU and EL are both similar: EU relates to the ease of using a system once a user has learnt it, whereas EL relates to the ease of using a system when a user uses it for the first time.

EL scored the highest ratings, with an average of 4.55 across all its items, with EU following close behind with an average of 4.21. Two items in EL: 'I learned to use it quickly' and 'It is easy to learn to use it.' has a rating of 4.68, which is larger than any rating across all other categories. These ratings reflected in the user feedback, in Appendix C and D, where users commended the simplicity of features and their ability to grasp how to accomplish tasks quickly. When asked to list the most positive aspects of COCE, most users praised its easiness to complete tasks, from navigation and creating nodes to editing and linking edges.

In a study to investigate the relationship between usability and interface aesthetic, it was found that visual appeal plays a role in users' subjective feedback on the usability of an application [19]. Users perceived notion of usability is that if a system is aesthetic, it is more usable [19]. For this reason, a basic minimalistic design was used for COCE to help limit bias on colour schemes, fonts, etc., when determining the usability and to make the main focus on functionality instead. Figure 6 and 7 displays the simple user interfaces of the concept map dashboard and concept map creation screens. According to participants, this simple design acted as both a positive and negative aspect of COCE. We received clear feedback on how its simplicity made it easy to perform tasks. However, many users commented on the lack of appeal when asked to note the most negative aspects of COCE. Despite gaining good feedback on the usability of the application, users showed a lack of satisfaction when using the COCE. This can explain why the S ratings had the lowest average, of 3.56, from the four categories.

The design of COCE was simple enough to perform required tasks easily however, it also lessoned the engagement and enthusiasm when using the application, resulting in a low satisfaction rating.



Home Logout

Logged in successfully!

Concept Map Dashboard

List of projects:

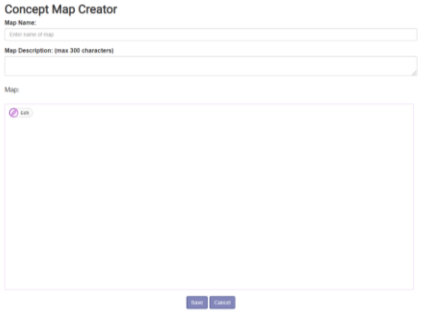
Culture of South Africa

South Africa is known for its ethnic and cultural diversity. Data sourced from Wikipedia.

Cape Town and its History

Colloquially named the Mother City, it is the largest city of the Western Cape province and forms part of the City of Cape Town metropolitan municipality. Data sourced from https://en.wikipedia.org/wiki/Cape_Town

Figure 6. Concept map dashboard which shown once a user logs in successfully. It displays a list of user-created projects, a button to create a new map and a logout option on the top bar of the web page.



Concept Map Creator

Map Name:

Enter name of map

Map Description: (max 300 characters)

Map:

Figure 7. The concept map creator screen, with a blank name, blank description and empty map.

5.3 Proposed COCE Changes

Overall users were content with the prototype. Although we built a prototype to test the functionality and feasibility of a concept mapping creation tool, users wanted more functionality. This simple prototype can be taken a step further and developed into a well-rounded concept mapping website creation tool. Table 9 summarizes the participants suggested improvements to COCE.

Table 9. Suggested improvements according to participants.

More captivating design and layout of GUI.
Allow customization of nodes and edges.
Refine edges that connect nodes – more precision required.
Allow various types of information to store in nodes (e.g., video).
More help functionality.
Shortcuts to replace mundane tasks.
Increase compatibility with other browsers
Remove compulsory restriction on node description and details.
“Archive dumping” feature.

5.3.1 Design Improvements

One aspect, critiqued by most participants, is the user interface of COCE. They found that it was monotonous and wanted a more appealing interface. This can easily be fixed by changing the colour schemes and adding backgrounds. More interactivity can also increase user satisfaction such as animations and transitions when navigating between screens.

5.3.2 Usability Changes

In addition to aesthetic design, users would like more supporting features. COCE should incorporate more help prompts, icons and tooltips to give users hints on functionality and unclear buttons. A user commented that they “Don't really understand what it should be or the full extent of the utility”, and proposed that “Few suggestions/examples on the website about the functionality and how to build concept maps” might help. Simple to advanced help functionality includes a standard About page on the web application outlining COCEs general purpose or a Help tab that can explain how to perform different tasks in both written and video tutorial formats.

5.3.3 Concept Map Modifications

Participants were happy with the creation of maps, however many suggested customizations of nodes and edges. This includes changing a nodes shape, size and colour. Users commented on how node size and colour can help differentiate significant nodes over other nodes of minor importance. Edge customization can include having different type of lines, such as dashed or dotted, and different line widths. A user mentioned how different lines can help signify the strength of relationships between nodes, for example a dashed line can illustrate a weak relationship between nodes. Another user mentioned how they would like to have arrowed lines to show direction to emphasize relationship direction between nodes.

Users expressed interest in removing mundane tasks and creating shortcuts to perform common tasks such as adding a node or edge. Currently, when a user wants to add a node or edge, they click the respective button and place a single node or edge on the map canvas. A user suggested the functionality to click the add node/edge button once, and then insert the desired number of nodes or edges, without repeatedly having to click to add a new one.

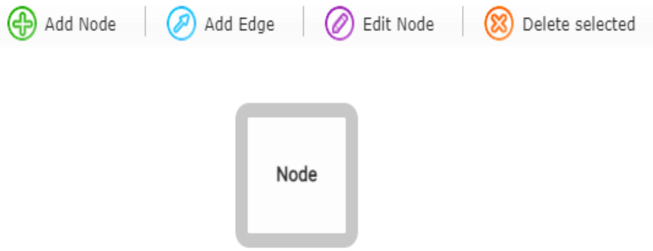


Figure 8. The basic design of a node on a map. Map editing options are displayed above the node: Add node, add edge, edit node and delete selected.

5.3.4 Advanced Features

Despite some basic features such as design changes and customizing concept maps, users suggested some advanced features that can be beneficial for a cultural heritage concept map creation system.

One is the linking of concept maps. An option to import a map created on the system to link with another map can allow bigger collections of data to be visualized.

Another feature is ‘Archive dumping’. This is an option to allow a user to dump data that they may potentially use to create nodes in maps.

COCE was commended for storing large amounts of information, however, there was a suggestion to include different types of information. Our prototype focused on mapping textual information. The ckEditor that is used to accept node details can be expanded to include other visual formats such as images, videos and audio.

5.4 Limitations

This study was conducted in the midst of the Covid-19 global pandemic, hence face-to-face communication was prohibited with anyone involved in the study. Communication between team members and supervisors was restricted to emails and video conferencing tools, such as Jitsi.

Ethical clearance to use UCT students in research was obtained very late in the study. If clearance was obtained earlier, more students could have been recruited to participate in the study and expand on the evaluation results for a more diverse feedback result.

Another limitation is that the user evaluations were conducted online. If users did not understand a question, or input mistakes there is no way to know. Online evaluations restrict user engagement, and feedback could be limited compared to if evaluations were face-to-face.

6 CONCLUSIONS

The research question for this study was to test the user experience of users when using an integrated concept mapping and complex object creation tool to create and edit complex digital objects. This web application contributes to the cultural heritage informatics and information visualization field. This paper outlines the implementation and evaluation of the COCE, a complex object creator and editor. 25 participants evaluated the COCE and most of the qualitative results suggested that the concept is promising, and that users are interested in this tool and its future potential.

This general concept map tool can be taken a step further and made into a specialized concept mapping tools that can be used by different specializations dealing with large amounts of data and who are interested in using a concept mapping tool to categorize and condense the information they see.

Perhaps it would have been beneficial to work directly with academics when creating the prototype and develop features they find helpful.

6.1 Future Work

Future work for this study can be taking this prototype and developing a fully functioning concept creation tool using the feedback received from this study. With the direction of professionals from various fields, this simple concept mapping tool can be modified, with advanced features, to create a specialized concept mapping tool that can be used in different academic niches. This involves incorporating HCI principles in interface design and developing advanced features such as linking of maps and archive dumping. Furthermore, the system can incorporate visualizing metadata of digital cultural objects instead of just its content.

ACKNOWLEDGMENTS

The author gratefully acknowledges the input and time of all the participants who participated in the user evaluation process and provided useful feedback. The author would also like to extend their deepest gratitude to their project supervisor Professor Hussein Suleman, second reader, Mr Aslam Safla, and project partner, Yashkir Ramsamy, for their guidance and feedback that made the completion of this project possible.

REFERENCES

[1] Arms, W., Blanchi, C. and Overly, E. 1997. An Architecture for Information in Digital Libraries. Mirror.dlib.org. http://mirror.dlib.org/dlib/february97/cnri/02arms2.html #digital-object

[2] Avgousti, A., Papaioannou, G. and Gouveia, F. 2019. Content Dissemination from Small-scale Museum and Archival Collections: Community Reusable Semantic Metadata Content Models for Digital Humanities. code{4}lib journal. https://journal.code4lib.org/articles/14054.

[3] Diagrams.net. 2021. Flowchart Maker & Online Diagram Software. App.diagrams.net. https://www.diagrams.net/

[4] Flask. 2021. Welcome to Flask — Flask Documentation (2.0.x). https://flask.palletsprojects.com/en/2.0.x/

[5] Gao, M., Kortum, P. and Oswald, F., 2018. Psychometric Evaluation of the USE (Usefulness, Satisfaction, and Ease of use) Questionnaire for Reliability and Validity. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 62(1), pp.1414-1418. DOI: https://doi.org/10.1177/1541931218621322

[6] Heroku. 2021. Cloud Application Platform | Heroku. https://www.heroku.com/

[7] Iris Xie, Krystyna K. Matusiak. 2016. Discover Digital Libraries.

[8] Jordan, M. and McLellan, E. 2016. PREMIS in OpenSource Software: Islandora and Archivematica. Digital Preservation Metadata for Practitioners. DOI: https://doiorg.ezproxy.uct.ac.za/10.1007/978-3-319-43763-7_16

[9] Kakali, C. et al., 2021. Integrating Dublin Core metadata for cultural heritage collections using ontologies.

[10] Kalfatovic, M., Kapsalis, E., Spiess, K., Van Camp, A. and Edson, M. 2008. Smithsonian Team Flickr: a library, archives, and museums collaboration in web 2.0 space. Archival Science 8, 4, 267-277

[11] Kumar, G. and Bhatia, P., 2012. Impact of Agile Methodology on Software Development Process. International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(4), pp.46-50.

[12] Lagoze, C. and Van de Sompel, H. 2003. The making of the Open Archives Initiative Protocol for Metadata Harvesting. Library Hi Tech 21, 2, 118-128. DOI: https://doi.org/10.1108/07378830310479776

[13] LucidChart. 2021. Intelligent Diagramming | Lucidchart. Lucidchart.

[14] Lynn, R., 2021. What is FDD in Agile? | Planview. Planview.

[15] Puckett, J. and Leslie, S. 2016. Omeka. Journal of the Medical Library Association. http://dx.doi.org/10.3163/1536-5050.104.4.030

[16] Python Pool. 2021. Python SHA256: Implementation and Explanation.

[17] Seadle, M., 2006. Content management systems. Library Hi Tech, 24(1), pp.5-6.

[18] Society, N., 2021. GIS (Geographic Information System). National Geographic Society.

[19] Tractinsky, N., Katz, A. and Ikar, D., 2000. What is beautiful is usable. Interacting with Computers, 13(2), pp.127-145.

[20] USE Questionnaire. 2021. USE Questionnaire: Usefulness, Satisfaction, and Ease of use. Garyperlman.com.

[21] Vis. 2021. vis.js - Network documentation. Visjs.github.io.

[22] Windhager, F. et al., 2019. Visualization of Cultural Heritage Collection Data: State of the Art and Future Challenges. IEEE Transactions on Visualization and Computer Graphics, 25(6), pp.2311-2330.

[23] Werkzeug. 2021. Werkzeug Documentation (2.0.x). Werkzeug.palletsprojects.com.

Supplementary Information

A. 25 Participants Specializations

Philosophy and Political Science	Marine Biology
Historical studies	Civil Engineering
Accounting	Electrical and computer engineering
Linguistics	Computer Science
History	Heritage Inventories
Chemical Engineering	Computer Science & Computer Engineering
Scholar	Chemistry
Urban Studies	Applied Biology and Ecology & Evolution
Computer Science	Biochemistry and Genetics
Electrical and Electronic Engineering	Biological Anthropology
Medicine	Computer Science and Business Computing
English Literature, Philosophy, and Psychology	Human Physiology & Anatomy, Biochemistry
Information Systems	

B. Number of participants grouped by ratings for each USE questionnaire item.

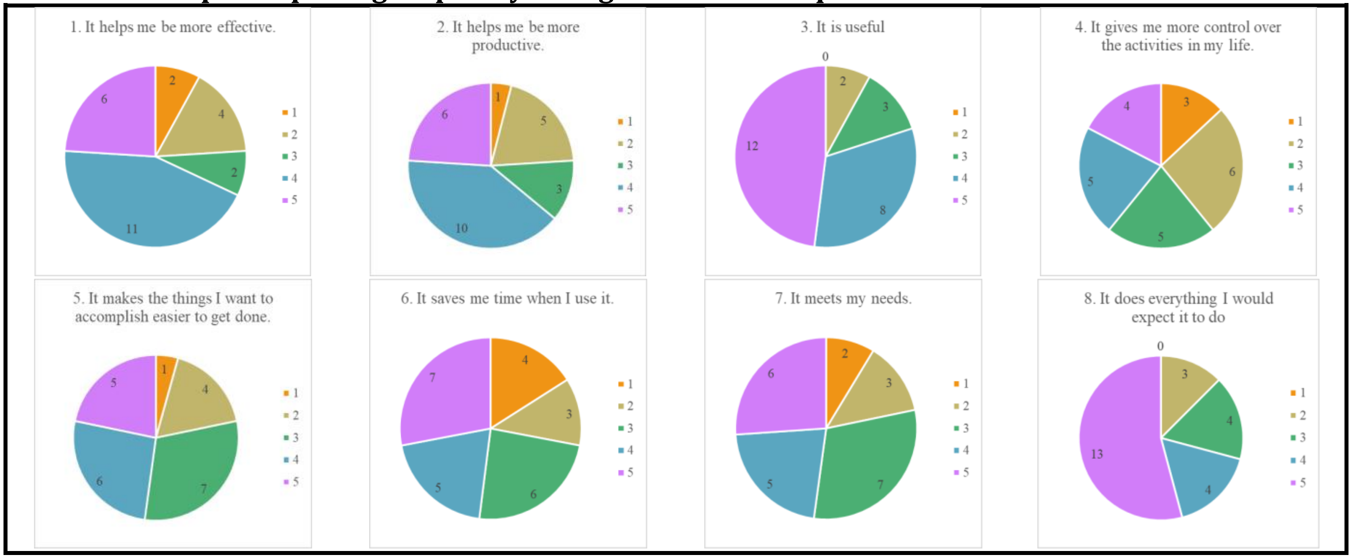


Figure 9. Pie charts illustrating the number of participants grouped by their rating for each item on the Usefulness category of usability. They were asked to rate different items on a scale of 1 (strongly disagree) to 5 (strongly agree). The items measured in this category (from left to right, top to bottom) includes: 'It helps me be more effective', 'It helps me be more productive', 'It is useful', 'It gives me more control over the activities in my life', 'It makes the things I want to accomplish easier to get done', 'It saves me time when I use it', 'It meets my needs', 'It does everything I would expect it to do

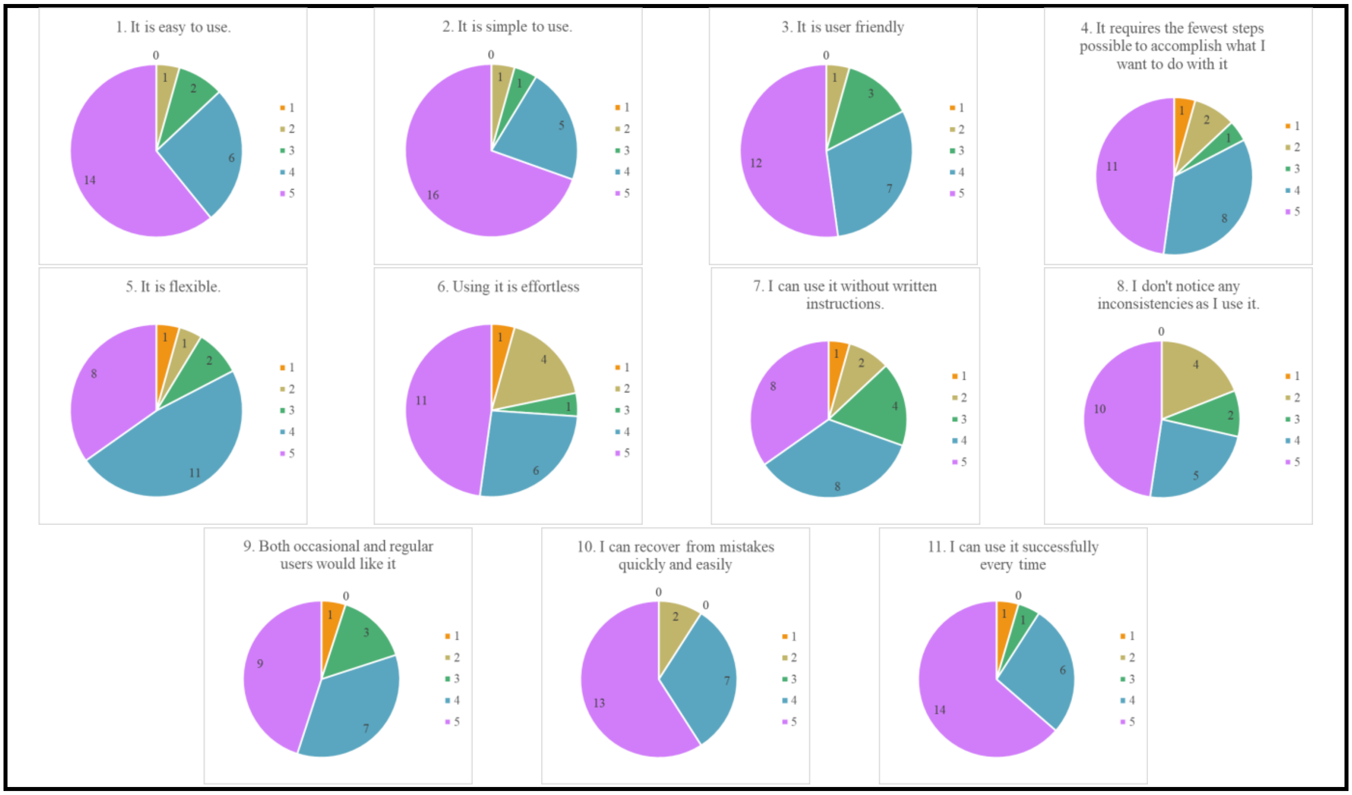


Figure 10. Pie charts illustrating the number of participants grouped by their rating for each item on the Ease-of-Use category of usability. They were asked to rate different items on a scale of 1 (strongly disagree) to 5 (strongly agree). The items measured in this category (from left to right, top to bottom) includes: 'It is easy to use', 'It is simple to use', 'It is user friendly', 'It requires the fewest steps possible to accomplish what I want to do with it', 'It is flexible', 'Using it is effortless', 'I can use it without written instructions', 'I don't notice any inconsistencies as I use it', 'Both occasionally and regular uses would like it', 'I can recover from mistakes quickly and easily', 'I can use it successfully every time'.

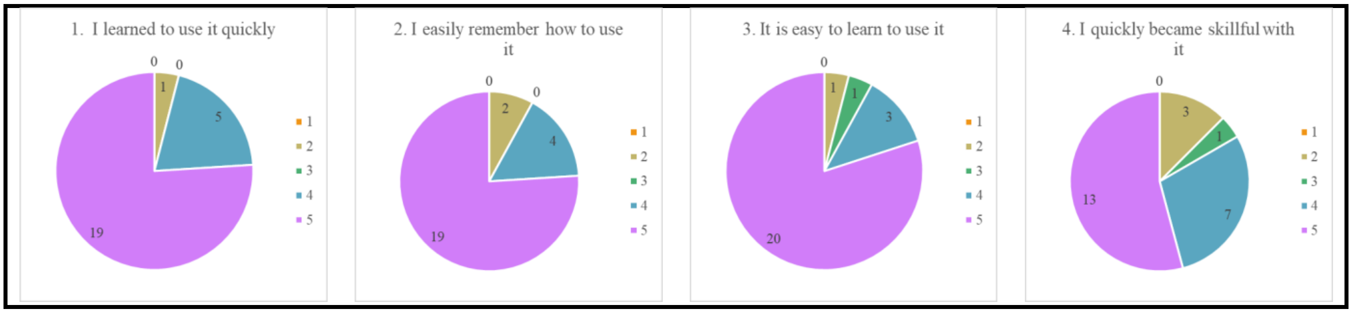


Figure 11. Pie charts illustrating the number of participants grouped by their rating for each item on the Ease of Learning category of usability. They were asked to rate different items on a scale of 1 (strongly disagree) to 5 (strongly agree). The items measured in this category (from left to right, top to bottom) includes: 'I learned to use it quickly', 'I easily remember how to use it', 'It is easy to learn to use it', 'I quickly became skillful with it'.

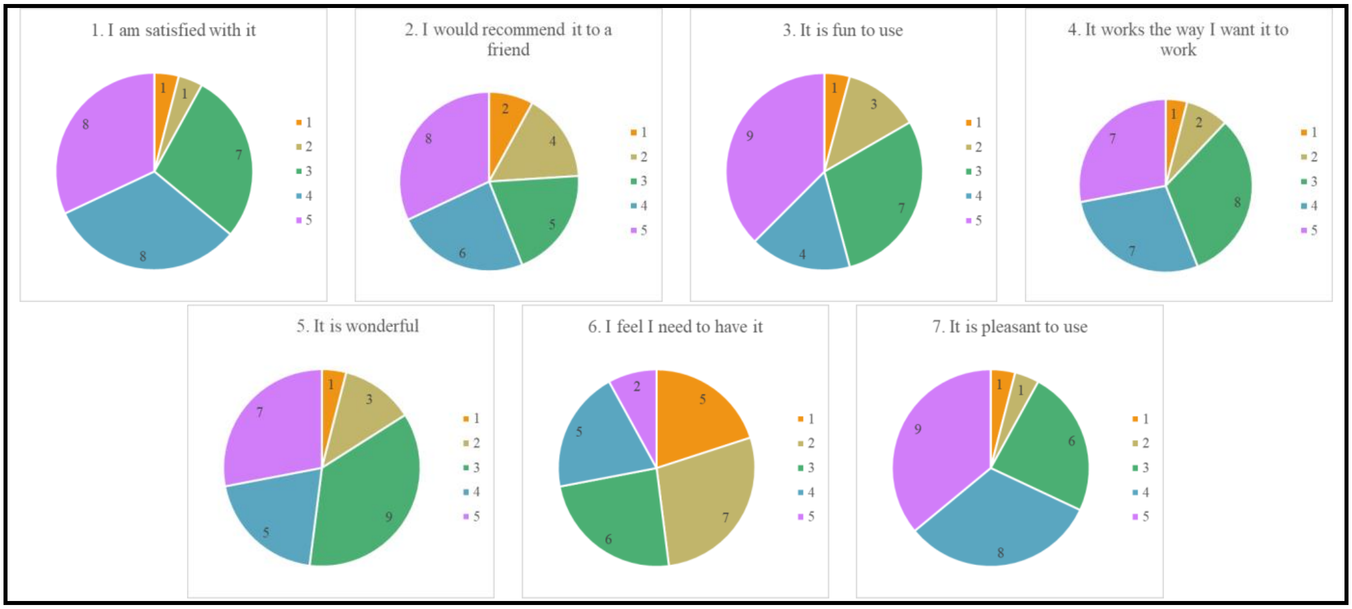


Figure 12. Pie charts illustrating the number of participants grouped by their rating for each item on the Satisfaction category of usability. They were asked to rate different items on a scale of 1 (strongly disagree) to 5 (strongly agree). The items measured in this category (from left to right, top to bottom) includes: 'I am satisfied with it', 'I would recommend it to a friend', 'It is fun to use', 'It works the way I want it to work', 'It is wonderful', 'I feel I need to have it', 'It is pleasant to use'.

C. Feedback of Participants: Negative and Positive Aspects of COCE from 25 participants.

Table. Negative and Positive aspects about the COCE.

Theme	**Feedback**
	NEGATIVE ASPECTS
Node Customization	"Can't customize shape/color of nodes"; "be able to change the shapes and forms of the nodes and the design of the edges", "No recommended layouts or formats"; "I would like there to be different colours available for nodes and there being stickers or something to distinguish which aspects are more important than others"
Web App Appeal	"Lack of Colour"; "Dull"; "Poor Design/Layout"; "It looks very basic, it needs more colour possibly bigger buttons";
Edges Improvements	"Edges could be slightly bigger"; "Edges do not have arrowheads"; "doesn't have different types of links"; "Missing variable node connections. At times having the ability to visually differentiate between different types of relationships through the edge line will be beneficial"
Inconsistencies	"Nodes and edges do not show up effectively."; "The 'edge' or line does not connect properly if I want to join to elements from the center"; "People are in the habit of pressing enter when finish typing items in, once enter is pressed it shows a new map when renaming the map."; "Sometimes difficult to connect nodes with edges"
Unclear features	"I couldn't find the delete button at first (it wasn't obvious)"; "The instructions about the edge (connecting the nodes) were a bit confusing for the first time"; "I did not really understand how the edges worked."; "I found adding a node difficult at first because I didn't realize you click to add."
Usability Improvements	"Adding paragraphs and descriptions to everything is tedious"; "There are no shortcuts with adding new elements"; "I feel like there's no need to keep pressing add link if you want to add a link, rather just click it once, then add all the links, then unclick it. The same applies to the nodes"; "The compulsory criteria on the details input. At times I would like to input the title and description, and then come back to the details to populate"
Zoom issue	"When I Scroll down the map window it automatically zooms in or out"; "I don't like that you can't see what the nodes display, unless you really zoom in and see the text within the squares."; "Zooms in and out too much, making it difficult to find the nodes if you've zoomed in too much"
Concept issues	"Works well and all but not unique - other programs do the same thing", "Only help students that use mind mapping to learn", "There are better alternatives", "Don't really understand what it should be or the full extent of the utility"; "Seems limited in the number of things you can do on it."

Compatibility	“It doesn't work on touch browser, it required google chrome”; “Issue adding node”
POSITIVE ASPECTS	
Design	“Simple layout”; “Basic Design”, “Simple user interface”; “The visualization is clear and very legible”; “Nice, clean appearance”; “user interface is easy to understand”; “The GUI is simple and effective.”
Easiness	“It’s very easy to use and master”; “Easy to follow and navigate”; “Edge feature easily connects relevant nodes”; “Can edit and remove items easily”; “Pretty easy to use and maneuver”; “Easy signup”; “Can easily organize ideas/concepts”; “easy to change things”; “easy to rearrange nodes”; “Edges connect automatically”; “I like how the input prompts are easy to use”
Usability	“Very user friendly”; “It is easy to use”; “It has a basic input and output characteristic”; “Simple enough”; “Easy to understand”; “It is intuitive”; “Fast and smooth”; “Clear”; “It is very easy to use and make sense of what is required and how to achieve what is needed to be achieved”; “little to no loading times”; “Very responsive”; “Very efficient”; “Allows me to drag nodes”; “The system is easy to use and you get the hang very quickly.”
Creating Maps	“It makes creating maps a lot faster and convenient compared to other platforms”; “Does what it is set out to do”; “Having the different categories, e.g., details or description”; “The maps that result from the system are neat and informative.”; “faster than word for creating mind maps”;
Versatile	“Useful for a variety of circumstances”; “More or less fluid”; “It could be useful in project development”; “Nice flexibility in adding details to nodes”
Useful	“Can be helpful when trying to organize yourself”; “Can get a copy of your diagram for external use”; “Efficient”; “Saves the mess of doing it on paper.”; “It can hold quite a lot of information.”; “Great way to visualize something”;

D. Feedback of Participants: Most and Least Useful Features of COCE from 25 participants.

Table. Most Useful and Least Useful Features According to Participants	
Theme	**Feedback**
Most Useful features	
Usability	“Probably how easy it is to use.”; “Very basic design, makes application very efficient. Editing is extremely easy. Easy to map notes and tasks that need to be done”; “I found that the display of the map being made was incredible useful when I tried to remember the information”;
Versatility	“The ability to add any element which means the maps or diagrams can be of any topic. There are no topic limitations.”; “Filling in the form for the information that would be inserted.”
Adding edges	“Easily connecting nodes with edges”; “Being able to connect edges to the nodes”; “The easy to drag node edges. It helped to quickly and easily link the relationships between the nodes”; “I think the most useful feature is being able to add and link nodes in an easy manner. In addition to this, the viewing of the map once creative is very simple yet effective.”
Re-positioning nodes	“The option of re-arranging the nodes”; “Being able to place nodes anywhere and then simply linking them, while also being able to move the nodes wherever.”; “Dragging nodes to other positions once placed”; “The drag function; it is nice to be able to physically move the nodes around as opposed to any other method.”
Details of nodes	“The most useful feature was the possibility of adding descriptions, which allows additional info to be added to a node. It would be useful to also have a description option for edges so that relationships can also be further specified.”; “The fact that it can hold a lot of data pertaining to an idea or concept of the node.”; “The most useful feature was the fact that each node had subheadings so that it is easier to organize the idea you would like to digitalize.”; “Creating nodes with details and descriptions”; “adding nodes, it’s very it’s simple and quick compared to word, which requires, adding a shape, then editing the shape to add text.”
Creating Account	“Having an account to be able to save my work”
Least Useful features	
Lack of features	“The lack of additional features (but maybe those are coming later)”
Nothing	“There wasn't a feature that I felt didn't enhance ease of use and the application overall.”; “I think all features were pretty essential”; “Not entirely sure that I can highlight one; due to the simple design, it seems like every feature had a purpose”; “Nothing” “In my experience, I did not find any of the features to be useless or least useful. I think all the features work together to form a cohesive and complementary environment.”
Description for Nodes	“Adding a 'description' to each element.”, “Description and details both serve the same function when creating a node”, “The description and the details can be similar at times, so it is time consuming to be forced to write details

	before you can proceed"; "The compulsory field criteria on the details field. At times you would want to rapidly create nodes where the title and description are sufficient, and then go back to complete the details"
Too much detail	"The feature that could hold extensive information of the idea or concept of the node"; "The description and details served the same purpose"
Undifferentiated Edges	"Edges, but I wouldn't say it was completely useless. I just didn't like the fact that it did not show the flow of the nodes. It does not show the order in which I wanted the reader to view my nodes."
Adding nodes/edges	"Having to click add link or add node every time one is to be added. It's a mundane task";
Formatting	"Formatting the text i.e., font color, font type"

E. Feedback of Participants: Suggested Improvements for COCE from 25 participants.

Suggested Improvements	
Theme	Feedback
Design	"Bigger buttons to select"; "reduce width of Textboxes little bit"; "Design a more interesting and immersive interface for the user, develop more tools to use to personalize the concept map experience more and make the connection between this mapping tool and cultural heritage objective clearer."; "I think so GUI components can be made more obvious like the logout button and pop-up instructions when adding or deleting nodes. I also think the incorporation of more colour into the GUI would increase customer satisfaction"
Customization of nodes and edges	"Adding a feature that allows for customization of the color/shape of the nodes might enhance the use of the application"; "Perhaps different colours for the various elements to show distinction"; "Be able to change sizes/shapes of nodes"; "adding the possibility of giving details on edges as well as nodes would be helpful. Also, the ability to resize the nodes to show their importance and edit the width of the edges so as to illustrate the strength of the relationship would be helpful."; "Make the "hitboxes" for the nodes larger"; "Add some colour and arrow heads to the edges. Also create more shape options and possibly animation features"; "different types of links (edges)"; "Add more colour"; "Different node edge options (for example: a dashed line to represent weak relationships)"; "Allow us to choose the shape of the nodes"; "add different colour options for nodes and ways to distinguish from important ones"; "having an arrow headed linker and more formatting options to the text"
Refined edges	"When adding links, the cursor doesn't attach very well to the other link, it needs very good precision, needs improvement"
More node data types	"More flexibility in creation of map. I want to be able to put more visual data"
Temporary Data Input	"Create an option for dumping an "archive": data that the user may or may not use in the final presentation, but can store, retrieve from and go back to."
Remove compulsory data input on node	"Removal of the compulsory criteria on the details input box"
Compatibility	Perhaps an app would be much better than a website since I have noticed inconsistencies across browsers
Shortcuts	"Remove mundane tasks (constantly having to click add link whenever a new link is to be added)";
Help prompts	"Few suggestions/examples on the website about the functionality and how to build concept maps"; "explanation of what a 'node' and 'element' is"; "Maybe provide more examples for a description vs detail"