universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

CyberSecurityBoard: A modeling tool for creating cyber security themed storyboards

verfasst von | submitted by

Bak. Mehrudin Šabani

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. Dr. Gerald Quirchmayr

Abstract

Educating and training the workforce in cyber security is both a necessity for companies and organizations and, at the same time, an increasingly demanding challenge regarding the effectiveness and efficiency of content delivery. One of the major trends in this context is the exploration of more practical and engaging training methods. This work seeks to explore the potential of utilizing storyboards and storyboarding techniques as a means of teaching cybersecurity topics.

Storyboards are a delivery method that offer a more engaging way to communicate the message to the learners with the use of real-life simulation-based scenarios. Storyboards provide an enhanced learning experience by structuring the content in a shorter, more memorable, and overall, more enjoyable manner.

The process of creating a storyboard usually requires multiple software tools and resources like graphics and images. A major goal of this work is developing a software solution that makes the process of creating storyboards simpler and more agile by providing everything necessary in a single tool. The developed software solution for this work is therefore called CyberSecurityBoard.

CyberSecurityBoard is a web-based application that has a built-in tool for creating cyber security-themed storyboards. Users of this application can create their own storyboards or explore storyboards created by other users. CyberSecurityBoard has built-in graphics and images of characters, objects, symbols, and everything else that is necessary to create a storyboard.

Even though storyboarding techniques show great promise as an alternative learning method, the content they provide is static. Many studies suggest that integrating gamified and interactive components into lecture materials greatly increases learners' engagement and knowledge retention rates. For that reason, a part of this thesis is also exploring the potential of integrating interactive components in storyboards and extending CyberSecurityBoard with interactive features.

Zusammenfassung

Die Schulung und Ausbildung von Mitarbeitern und Mitarbeiterinnen im Bereich Cybersicherheit ist für Unternehmen und Organisationen eine Notwendigkeit und zugleich eine zunehmend anspruchsvolle Herausforderung hinsichtlich der Effektivität und Effizienz der Bereitstellung von Inhalten. Einer der wichtigsten Trends in diesem Zusammenhang ist die Erforschung praxisnäherer und ansprechenderer Schulungsmethoden. Diese Arbeit soll das Potenzial der Verwendung von Storyboards und Storyboarding-Techniken als Mittel zur Vermittlung von Cybersicherheitsthemen beleuchten.

Storyboards sind eine Bereitstellungsmethode, die eine ansprechendere Möglichkeit bietet, den Lernenden die Inhalte mithilfe realitätsnaher simulationsbasierter Szenarien zu vermitteln. Storyboards bieten ein verbessertes Lernerlebnis, indem sie den Inhalt kürzer, einprägsamer und insgesamt unterhaltsamer strukturieren.

Der Prozess der Erstellung eines Storyboards erfordert normalerweise mehrere Softwaretools und Ressourcen wie Grafiken und Bilder. Ein Hauptziel dieser Arbeit ist die Entwicklung einer Softwarelösung, die den Prozess der Erstellung von Storyboards einfacher und flexibler macht, indem sie alles Notwendige in einem einzigen Tool bereitstellt. Die für diese Arbeit entwickelte Softwarelösung heißt daher CyberSecurityBoard. CyberSecurityBoard ist eine webbasierte Anwendung mit einem integrierten Tool zur Erstellung von Storyboards zum Thema Cybersicherheit. Benutzer dieser Anwendung können ihre eigenen Storyboards erstellen oder Storyboards anderer Benutzer verwenden. CyberSecurityBoard verfügt über integrierte Grafiken und Bilder von Charakteren, Objekten, Symbolen und allem anderen, was zum Erstellen eines Storyboards erforderlich ist.

Auch wenn Storyboarding-Techniken als alternative Lernmethode sehr vielversprechend sind, sind die von ihnen bereitgestellten Inhalte statisch. Viele Studien legen nahe, dass die Integration spielerischer und interaktiver Komponenten in Vorlesungsmaterialien das Engagement und die Wissensspeicherrate der Lernenden erheblich erhöht. Aus diesem Grund untersucht ein Teil dieser Arbeit auch das Potenzial der Integration interaktiver Komponenten in Storyboards und der Erweiterung von CyberSecurityBoard mit interaktiven Funktionen.

Acknowledgements

First and foremost, I would like to thank Prof. Gerald Quirchmayr for giving me the opportunity to work on this topic. His guidance and feedback while writing the thesis were of great value. It gave me the opportunity to learn so much during this long journey. I am especially grateful for the incredible amount of support, patience, and understanding shown by Prof. Gerald during some challenging times while I was managing both full-time work and finishing the thesis.

Most importantly, I want to express my thanks and gratitude to both of my parents, Arif and Nurisa, who always believed in me and supported me through all of my academic journeys.

I would also like to thank Project COLTRANE and UNISA for granting me permission to use the characters designed by them in my project. Their work also helped me expand my knowledge and understanding of cybersecurity education through the use of cyber ranges and collaborative learning.

Last but not least, I would like to thank all of my close friends and family members, not explicitly named here, who offered me guidance and moral support throughout the course of my studies.

Contents:

CHAPTER 1

Introduction

1.1 Problem statement

Cybersecurity is an ever-evolving topic that requires constant knowledge improvement to adapt to the constant changes. Lack of an adequate work force and an increase in cyber threats push the question of providing quality cybersecurity-related education even further [1, 2, 9]. Even though a very broad toolset and cybersecurity-related content have been available for decades, the gap in the skillset and lack of workforce suggest that further improvements need to be made [2, 19, 31, 49].

1.2 Aim of the work

The emphasis is put on interactive, engaging, and practical training approaches since cybersecurity is a field that requires very practical knowledge. Utilizing storyboarding techniques to teach cybersecurity topics has a lot of potential but is not a very well researched topic. Therefore, the following research questions and development goals have been set:

Research questions and goals:

1. What are the common positive characteristics of interactive and engaging training methods that contribute to an enhanced learning experience?
2. Do interactive storyboards (that will be developed using CyberSecurityBoard) have the potential to be an effective training method?
3. Do interactive storyboards have traits of positive characteristics of the evaluated methods?

4. What are the best story-telling practices for creating an engaging, captivating, and interactive storyboard with CyberSecurityBoard?
5. Applicability and Effectiveness of CyberSecurityBoard:
    - Evaluate how well CyberSecurityBoard can be applied in educational settings.
    - Measure the effectiveness of the tool in enhancing learning outcomes by conducting a user experience survey.

Development Goals:

1. Develop a learning platform with an integrated storyboarding tool (in the form of a web application) for developing modular scenarios composed of storylines, characters, interactions, etc.
2. Integrate features and graphics (such as characters, objects, speech bubbles, text, etc.) and everything else that is necessary to create a storyboard without the need of using external tools.
3. Created storyboards should be accessible by other registered users.
4. Minimalistic UI/UX approach in order to be understandable by a wider audience.
5. Using our own developed software, create educational cyber security storyboards.

1.3 Methodological approach

1. Evaluating the current state of cyber security education. Teaching methods, curriculum models, and supporting technologies.
2. Evaluating the training approaches that are of an interactive and engaging nature, the conducted studies, their successful implementation, and ultimately their impact on learners.
3. Comparison of interactive storyboarding with other methods by constructing a comparison framework.
4. Evaluation of existing tools for creating educational storyboards.
5. Researching best UI/UX practices to make CyberSecurityBoard an intuitive, easy-to-use tool.

CHAPTER 2

Literature, State of the Art

This chapter offers a detailed overview of the current cybersecurity situation, primarily in Europe but also globally. Literature on the most common cyber threats, education approaches and models to teaching cybersecurity, as well as supporting technologies, will be reviewed. The goal is to find potential gaps in the state-of-the-art cybersecurity teaching methods or uncover areas for potential improvement.

2.1 The current cybersecurity situation in Europe

Even though the field of cybersecurity is expanding, the skills and knowledge of the workforce in the field have not increased adequately [1]. The supply of qualified workers is not meeting the demand in Europe as well as worldwide [2]. Lack of a cybersecurity-specialized workforce logically leads to a higher number of cyber incidents [9]. Besides the lower supply of qualified workers in the cybersecurity field, general cybersecurity knowledge needs to be improved among the workforce in other fields as well, in order to reduce the number of cyber incidents, which can cause dramatically significant losses in worst-case scenarios [9].

It is extremely important to be cyber protected since cyber attacks can affect anyone, ranging from a single individual to large institutions and businesses. ENISAs (European Union Agency for Cybersecurity) Landscape report [3] tracks down the biggest cyber threats that have happened within a year, and the latest published statistic tracks data from July 2021 to July 2022 shows that there are eight prime threats. A threat is categorized as prime due to its popularity, impact, and prominence during the reporting period. These are the eight prime threats:

1. Ransomware: A type of attack where the victim's assets are stolen / taken control of and the attacker demands ransom in exchange for the assets.

2. Malware: performing an unauthorized process that will compromise a system's confidentiality, integrity, or availability.
3. Social engineering: exploiting a human error. It uses various forms of manipulation to trick the victim into making mistakes and thus granting access to sensitive information, systems, etc.
4. Threats against data: A collection of threats with the main goal of gaining unauthorized access to data. This threat is the basis of many other threats, such as the ones mentioned above.
5. Threats against availability (Denial of Service, or DoS): A successful DoS attack is when the users are not able to access a system or service. This attack is most commonly performed by either purposely exhausting the service resources or overloading the network infrastructure.
6. Threats against availability (Internet threats): restricting access to the internet, or at least a part of it.
7. Information manipulation: spreading fake or biased information across news sources, social media, etc. A strong example is the war between Russia and Ukraine, which has shown new ways to use this threat by trying to enforce biased views on people's perceptions of the war.
8. Supply Chain Attacks: A type of cyber attack that seeks to infiltrate the system through an outside partner that already has authorized access.

Tracking down the relevant threats is useful because it helps us better understand the current status of cybersecurity across the EU and develop new strategies on how to combat those threats.

European countries are aware of the challenges that exist within an ever-changing field such as cyber security. In an effort to increase skill and raise awareness, a lot of European countries have taken initiatives in the following forms: raising awareness among the general population, introducing cyber security programs in primary and secondary education, promoting cyber security in higher education, and organizing cybersecurity exercises and challenges [2, 61].

Even though some formidable initiatives have been taken, the gaps in cyber security still persist. ENISA has the following recommendations [2, 3] to address the cybersecurity shortage:

- Increase enrollment in cybersecurity programs
- Encourage a unified approach across government, industry, and universities.
- Understand job market needs and trends
- Collaborations between European member states

2.2 Teaching cybersecurity through engaging and interactive methods

Cybersecurity can be effectively taught using many different methods, just like most other topics. In this sub-chapter, we investigate learning methods that lean towards a more interactive and engaging approach compared to traditional methods. We then try applying those methods for teaching cyber security-based topics.

The motivation for exploring alternative learning methods is the fact that the current methods don't provide satisfactory results. One of the most consistent complaints from employers against cyber security education programs is that they place high emphasis on theory and book learning. According to the employers [4], this prevents the students from gaining practical skills that they need for tackling challenges in the real world. Of course, one of the best methods for gaining practical skills is to face real-world challenges in an actual work environment. A good example is taking an internship in parallel while studying. However, discussion of that topic is out of the scope of this work. Instead, the focus is on replacing or reducing classic book learning methods with more engaging activities and problem-oriented methods. The goal of these methods is to establish a connection between theory and practice by providing the students with a more hands-on and interactive approach.

The proposed learning methods in this work are activity-based learning, problem-based learning, game-based learning, as well as scenario-based learning. All of these methods can be appropriate for all age groups and skill levels if applied correctly. Another reason why these learning methods were chosen is that contemporary learning approaches such as the ones we mentioned have attracted considerable attraction recently in the education sphere [5].

2.2.1 Activity-based learning (ABL)

Activity-based learning is a learning method that allows students to experience and interact with the lecture content. This educational approach emphasizes active participation of the students in the learning process. In contrast with the traditional lecture-based teaching, it incorporates various activities that promote hands-on learning, critical thinking, and problem-solving skills. In order to incorporate activity-based learning, the teachers must adjust the classroom environment to the students requirements in order to make the learning process more enjoyable. Because activity-based learning aims to connect students with real-world practical problems, it increases learners attention and engagement, resulting in higher knowledge retention [6, 7, 62]. The positives of activity-based learning are:

Student Initiative and Engagement: ABL often motivates the students to actively participate and take charge of their learning journey. The students often explore a topic even beyond the lecture materials, which leads to the discovery of new ideas and even surpassing set learning goals [62, 63].

Enhanced learning outcomes: Studies have shown that ABL improves cognitive, affective, and behavioral outcomes in students. It helps in better retention and understanding of the material compared to traditional teaching methods [64, 65].

Diverse Learning Activities: ABL incorporates a variety of activities such as games, worksheets, project-based learning, and interactive strategies like flipped learning and group discussions. These activities cater to different learning styles and make learning more engaging and effective [65].

Technology Integration: The integration of technology, including E-Learning and M-Learning, enhances the effectiveness of ABL by providing interactive and flexible learning environments. This approach is particularly beneficial in higher education and professional training settings [66].

2.2.2 Problem-based learning

In problem-based learning, students are confronted with authentic real-world problems that they usually have to solve in teams or by themselves [10]. This method aims to improve problem-solving skills, self-directed learning, and collaboration between students. Instructors

are acting as facilitators instead of being the primary source of information [67, 68]. Study sessions are spent by either:

- Groups or individual students reporting their progress and issues that they encountered during their individual learning, discussing plans of future work [69].
- Short lectures dealing with and clarifying common issues or difficulties, suggesting additional learning issues
- Engaging in complex problems that do not have a single correct answer [69].
- Collaboration with other classmates by solving a problem with group effort.

Problems may vary in complexity and range. There can be single-topic, single-discipline problems that can be solved in a matter of days to more complex multidisciplinary problems that may take months to solve.

2.2.3 Gamification and game-based learning

This learning method is an educational approach that uses serious games as a learning method into the learning process in order to enhance students engagement, motivation, and ultimately knowledge [70, 71]. Another related concept to game-based learning is gamification. Although both methods rely on game elements, gamification is a more lightweight approach. Instead of spending resources to develop a serious game, it adds gamified elements to already available learning materials, thus making it more interactive and engaging [72].

2.2.4 Scenario-based learning

Scenario-based learning is an educational approach that uses realistic scenarios to engage students in active learning [73]. It is also a method that encourages learners to improve upon their real-life skills during the learning process. It helps close the gap between theory and practice and is highly conducive to learners becoming more skilled, especially when they are unsure about certain topics. Sheridan and Kelly [5] claim that scenarios should have a connection with the real world; this way, the students can establish a connection with real-world application methods that the learners will eventually encounter in the future in their professional careers. One of the challenges of scenario-based learning is that it should not only offer realistic learning experiences but also be interesting enough to keep the

learners attention. Therefore, good storywriting is also a necessity to successfully implement this method [74].

An interesting observation is that the reviewed learning methods are not completely separate entities. They have distinguishable defined boundaries but can easily intersect with one another. Problem-based learning may use activity-based learning methods; game-based learning has an activity-based nature that often uses problem-based approaches to solving tasks and so on.

2.3 Delivery methods for teaching cybersecurity

This chapter part explores delivery methods in terms of how the learning material is structured (curriculum models), explores ways it can be taught, and offers an overview of supporting technologies that are being applied in the field of cybersecurity.

2.3.1 Curriculum models

As briefly mentioned in the beginning of this chapter, cyber threats are growing in both the private and public sectors. Continuous curriculum development is critical because the cyber threats keep evolving. Constantly updating the skills and knowledge of the workforce has become a necessity.

A curriculum can be defined as a design that follows a carefully constructed path of learning material that is based on what the teacher or institution aims to accomplish over the course of the class. There are five categories [11] the curriculums can be classified as:

- Subject: A subject-centered curriculum is the most common one. Students are enrolled in many classes that cover different subjects; these types of curriculums are most commonly used in secondary and higher education.
- Integrated: In integrated curriculums, the focus is on building skills rather than obtaining theoretical knowledge. This framework integrates the desired skill into multiple subject areas; this way, the students can improve their skills while practicing in multiple contexts.
- Spiral: This type of curriculum is all about repetition. Students improve their skills by repeating identical tasks or exercises with variable details.

- Inquiry: In this type of curriculum, students focus on specific, usually open-ended problems to which they need to apply their skills and critical thinking in order to find a solution to a problem.
- Experiential: This curriculum gives the students the opportunity to learn through experience. Such curriculum is mainly used after a student has mastered skills associated with the subject and is in need of practical knowledge to further improve.

The above-listed curriculums are a general classification. However, in the real world, institutions and companies often practice combining multiple curriculum models to create a suitable teaching framework that suits their personal needs most efficiently. In a study [12], cybersecurity workers in Norwegian critical infrastructure companies were interviewed about their company's training offerings and the content of those training programs. According to the study, these fields in cyber security were the most common focus of current training offerings:

- Information handling (information disclosure, information sharing, and reporting).
- Human factor aspects (communication, trust management, teamwork skills, decision-making).
- Procedures and preparedness plans for cyber incidents.
- Security Management System (risk assessment & management, mitigation strategies, control strategies, documentation).
- Cyberthreats and relevant potential attacks and system vulnerabilities.
- Surveillance.
- Managerial skills training.
- Crisis contingency and management.
- Incident management and response.
- Intrusion detection.

Investigating how organizations structure their training programs to meet their requirements offers valuable insights for future studies to build upon. Exploring different curriculum models and the relevant cyber security fields helps us better understand what topics to prioritize and how to structure content delivery in the later stages of this work.

2.3.2 Frameworks

Framework in the context of this work is combining multiple curriculum models to construct a custom learning path that satisfies the specific demands of an organization or institution. Frameworks allow instructors to align learning goals with classroom activities as well as integrate assessment into learning. The learners are then able to use the framework as a self-service tool to interact with the content provided.

It is also worth mentioning that there are existing frameworks for constructing cyber security-based curricula. One good example is Cyber2yr2020 [8], which is a framework mainly used for constructing curriculums in two-year-long community colleges in the USA. According to their curricular framework, cyber security is divided into eight knowledge areas/domains:

1. Data security
2. Software Security
3. Component Security
4. Connection Security
5. System Security
6. Human Security
7. Organizational Security
8. Societal Security

Another example is the CyTrONE framework [15]. It uses input from a training coordinator to generate the training content for a particular training session and uploads it to an e-learning system. It is also able to create a cyber range (explained in the following sub-chapters) training environment corresponding to the training content. An updateable training database allows for automatic content generation. The training content can be easily updated by the instructor. This framework is web-based and is also optimized for mobile users.

The NICE (National Initiative for Cybersecurity Education) framework is a workforce framework that establishes a taxonomy and common lexicon that describes cybersecurity work and workers irrelevant to where or for whom the work is performed. This framework has a very different use compared to the previous example; it allows the user to learn about

potential roles in cybersecurity and the respective skills needed for those roles. It is constructed of the following components:

- 7 Categories: high-level grouping
- 33 specialty areas: distinct areas in cyber security
- 52 work roles: the specific details, which include the knowledge, skills, abilities, and tasks

Other interesting curriculums include:

NIST Cybersecurity Framework, a policy framework for how private sector organizations can improve their cybersecurity management [88].

CISA Cybersecuroty Training Framework, which is a cybersecurity agency that offers the latest cybersecurity standards, best practices, and guidelines to organizations [89].

EDUCAUSE Cybersecurity Framework is specifically oriented towards higher education institutions. It provides digital learning strategies to institutions that help boost student success [90].

2.3.3 Ways of teaching cybersecurity

Teachers, instructors, or organizations have a broad toolset available, which (if the resources and circumstances allow it) they can take advantage of to effectively teach cyber security topics. As of today, the most popular ways of teaching cybersecurity are through:

Traditional lectures: Students are passive spectators while the lesson is being taught by the teacher either orally or as a combination of visual presentations with detailed explanations. This way is used to teach the theoretical part of a subject. This is included in the subject-based curriculum mentioned in the previous chapter.

Exercises and practicals: teaching through providing practical examples and giving students the opportunity to solve practical exercises. This way of teaching is part of the integrated curriculum.

Cyber exercises: organized exercises that can be either local or international. This form of teaching is usually a hosted event where participants can gain new relevant knowledge through provided exercises. For example, ENISA has been organizing EU-wide exercises for the past 15 years, almost every year; it is called Cyber Europe [75].

Online delivery: As implied from the name, online delivery can be used as an alternative way to the first two points (traditional lectures and exercises); instead of attending a lecture in person, the students do it online. Besides the live online lectures, online pre-recorded courses are extremely popular and can be accessed whenever the student wants.

2.3.4 Shortcomings and potentials of the teaching methods

Traditional lectures are often heavily theory-based, and as we learned from the second chapter, cyber security is a field that needs to be taught interactively with real-world examples and exercises. Traditional lectures have their own place and offer immense benefits, such as social interactions. Students can usually receive instant help or feedback on their task.

Exercises and practicals are a more recommended way to teach cybersecurity, according to employers [4], because the students get practical experience that they can directly apply in a working environment. When constructing exercise-based teaching methods, the teacher needs to be aware of the relevant topics. Since cybersecurity is a rapidly changing topic, the content of the exercises needs to be regularly updated.

Cyber exercises are a great way to increase practical knowledge as well as find social circles with similar interests. Cyber exercises usually provide the most relevant and up-to-date topics to the participants.

Popularity of online lectures is only increasing, especially since the COVID-19 pandemic. One big benefit for many people is that they are able to enjoy the lectures from the comfort of their homes. The downside of online delivery is that students don’t always get answers on demand, and there are other psychological factors that cause students to lose focus on the lecture.

2.4 Supporting technologies

As already stated, online delivery is incredibly popular when it comes to teaching computer science-based courses. When it comes to cyber security, there is an array of supporting technologies available in the form of an online service or an application. Supporting technologies are meant to be used as a supplementary tool to ones studying and are usually extremely useful if implemented and used properly.

2.4.1 CMS and LMS

Content management systems are web-based systems that allow the user to add and manage media content without the worry of how the backend system works. Content management systems (CMS) and learning management systems (LMS) have been around for a longer period and are well-standardized technology. LMS can be considered a subgroup of CMS, they are used by educational institutions and organizations. Some of the most widely used LMS's of today are: Moodle, Edmodo, Blackboard, Skillsoft, etc. [76, 77, 78, 79]. All of them offer basic LMS features as well as unique features that are suitable for different user groups. They can be either a free or paid service. The core offering of all these learning systems is the following: uploading of video, presentation, or text-based lectures; grading of assignments; forums for discussions; uploading of assignments and exercises.

2.4.2 Content repositories

A content repository is a broad term; in this context, it is a database of digital content displayed in an understandable and easy-to-navigate way to the end-user. The user is able to browse digital content and view the details of it. Today's most popular content repositories are repositories containing video courses; they are either free, premium, or freemium. Examples of such repositories include the very well-known: Udemy, Coursera, Youtube, Pluralsight, etc.

2.4.3 Cyber ranges

Cyber ranges are simulation-based interactive technology in an isolated environment that provides hands-on cybersecurity practice from a beginner-to-expert level. They provide a secure and legal environment for cybersecurity education, practice, and warfare training. It is an innovative method that allows trainees the opportunity to recognize and respond to real-world challenges in a controlled environment [18, 19]. This way, the real infrastructure is never at risk as a consequence of cybersecurity training. The proposed classification [18] for cyber ranges is:

1. Scenarios: A scenario accurately represents the operational environment and training requirements and drives the execution of the training to achieve a certain objective.
2. Monitoring: Monitoring includes the methods, tools, and layers at which real-time monitoring of cyber security exercises and tests is being performed.

3. Teaming: Teaming either includes an individual and/or a group of individuals that participate in a cyber security exercise or test (developing, designing, etc.).

2.4.4 Virtual Reality

VR still counts as a relatively new technology. Despite that, many efforts are being invested into exploring the capabilities of this technology in the cybersecurity field. The feature to simulate real-world infrastructure and environments is extremely compatible with a field like cybersecurity [81]. Since this is still a new concept, more research on how to optimize the use of VR for educational purposes is needed.

2.5 Establishing the direction of the thesis

The direction in which the master thesis is headed is inspired by the scenario-based learning method previously mentioned. This work seeks to investigate the potential of utilizing storytelling scenarios as a method for teaching cybersecurity. For creating the scenarios, storyboarding is the proposed solution as a very agile and efficient method.

A storyboard is a visual representation of a story, structured of chronologically stacked images or sketches. Storyboards are mostly associated with video production because they are very useful for breaking down scenes into individual panels. However, the thesis will focus on using digital storyboards as a tool for education. For example, telling a story by simulating a real-life problem (and providing a solution to that problem) can significantly increase the user's interest and motivation to learn about a certain topic.
Storyboarding implements a combination of many engaging and activity-based methods mentioned in this chapter.

The inspiration for utilizing storyboards as a means of teaching cybersecurity initially came from SAP Scenes [13]. This is a tool that has been developed in collaboration with SAP and Apphaus. It provides the users with a very rich array of graphics covering many different categories. Graphics for devices, buildings, backgrounds, characters, and many more are available. The graphics have a hand-drawn, cartoony-like style. As an end result, the created storyboards have a more casual, easy-going tone. Nonetheless, this method has proven to be a very effective, cheap, agile, and efficient way of conveying the message.

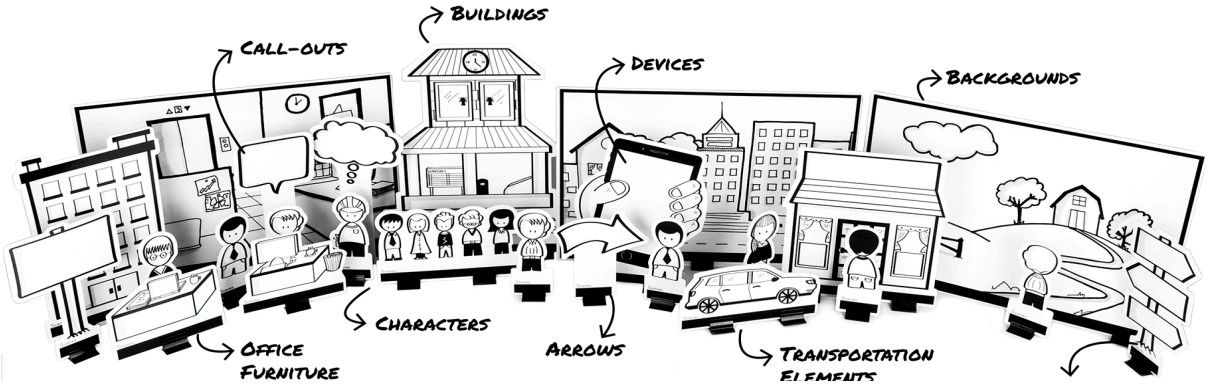

Figure 1.1: SAP Scenes by AppHaus

The work of this thesis goes a step further. A digital tool will be developed for creating and sharing storyboards with other users. The tool should be usable by both learners and storyboard creators (teachers). The emphasis is put on cybersecurity, so the integrated graphics and features will lean towards cybersecurity-themed topics.

A takeaway from this chapter is that current cybersecurity teaching, although very broad, is still not fulfilling the needs of learners and employers. Cyber ranges, storyboarding, and scenario-based approaches show very promising results. Traditional content should be complemented with animated self-service modules that contain practical cases. The storyboarding approach developed in the thesis can serve as a basis for case-oriented teaching, enabling scenario-based learning, etc. It can also be applied in preparing exercises for cyber ranges and for supporting desktop cyber exercises.

CHAPTER 3

Learning from the most common cybersecurity training approaches

The first part of this chapter evaluates the most common methods and technologies for teaching cybersecurity that were briefly discussed in the second chapter. A deeper analysis as well as research results of the effectiveness of those teaching approaches will be investigated in an effort to figure out which are the common characteristics responsible for enhancing the learning process.

The second part of this chapter tries to find the appropriate placement and utilization of the interactive storyboarding teaching method developed in this work.

And finally, the third part contains survey results of user-reported experience of a demo version of the CybersecurityBoard.

3.1 Comparison of cybersecurity training approaches

In order to figure out what characteristics improve the learning experience, we investigate the most popular approaches to teaching cybersecurity. By analyzing conducted studies, research results, and successful use cases, the goal is to find the trends that contribute to high engagement and retention rates for learners or participants.

3.1.1 Cyber Ranges

Cyber ranges are the most high-end technology for cybersecurity training that will be discussed in this work and therefore the most costly solution to implement. In exchange, they offer the realest simulations of the real world while completely protecting real infrastructure from getting accidentally damaged. The goal of this training method is to provide the learners with practical knowledge [18, 19, 20]. Cyber ranges have the potential to contain very advanced lecture content, which means that even seasoned experts can use cyber ranges to further improve their practical knowledge.

What's really interesting is that there exist cyber ranges that have randomized scene generation technology. Such is the case with the CyExec and SecGEN [22, 23]. This means that learners avoid receiving the same content every time they use the cyber range. The learning objective remains the same, but the scenarios presented to the learners may be different after each restart. The randomness factor is much more in line with real-life unpredictability, leading to a further enhanced learning experience and a higher engagement from the learners side.

Cyber ranges, although mainly used for cyber security training, have other fields where they are applied. They allow testing and integration of many different cybersecurity technologies since they already have capable infrastructure to accomplish these tasks [24, 25].

There are existing examples of successful integrations of cyber ranges within organizations, educational institutions, government agencies, etc. Good examples of these cases are the National Cyber Range (NCR) and AIT Cyber Range [21, 25].

The NCR is used by the United States Department of Defense, developed by DAPRA [80], with the objective to train the workforce on cybersecurity threats and defenses. It supports a wide range of activities, such as cyber warfare exercises, technology testing, and training programs. This particular Cyber Range uses very advanced simulation and virtualization technologies in order to create scalable and realistic cyber environments. After performing cybersecurity training with the NCR, participants abilities to identify and mitigate cyber attacks increased significantly.

AIT Cyber Range, designed by the Austrian Institute of Technology, is another cyber range whose objective is to provide a flexible environment for cybersecurity training, exercises, and research. It contains exercises such as ransomware attacks, phishing simulations, insider threat detection, etc. Participants are able to gain practical experience in handling real-world cyber incidents.

Drawbacks and Challenges of Cyber Ranges

The biggest drawback of Cyber Ranges is their cost. The hardware required for the virtualization technology, development costs, maintenance, and preparing learning materials (like scene design) are all factors that contribute to a very high cost. Because of this, accessibility of cyber ranges is relatively low. Nakata et al. propose the use of virtualization containers [22]. In their work, by utilizing virtualization containers, they managed to reduce the memory consumption by 50% and storage use to 1/60. Another proposal by Stickney et

al. is to take advantage of cheap but powerful enough hardware, such as the Raspberry Pis [27]. A complete prototype built with Raspberry Pis costs around 250 US dollars and consumes less power than a traditional solution, which makes it a much more affordable and therefore more accessible option for educational purposes. Another contribution that comes more as a side effect is the random scene generation feature discussed a bit earlier. This reduces the cost by avoiding having to design and develop more individual scenarios.

3.1.2 Cyber Games

Cyber games are serious games in the context of cyber security. They are educational tools designed to teach cybersecurity principles through interactive and engaging gameplay [34]. Cyber games simulate real-world cyberattack scenarios, thus providing practical learning experience.

A study by S. Ros et al. [28] analyzed the self-perception of students success and learning retention using a cyber game in an online cybersecurity course. The game was designed based on a framework utilizing cognitive constructivism learning theory [33]. The game uses metaphors with a storytelling approach to present cybersecurity content to students. The study included a detailed curriculum design of the proposed game, analysis of students success (self-perception), evaluation of the game's effectiveness, and discussions on the obtained results. The subjects of the study were 248 Computer Science students, and the provided game was not compulsory. Out of 248 students, 119 decided to play the game. They were labeled the “gamer group,” or GG; the other group was labeled the “non-gamer group,” or NGG. It is important to notice that this selection may be prone to bias and may be hiding the fact that more engaged students were more likely interested in playing the game to begin with.

The results are portrayed in the following table:

Groups	The final grade (PE + F2F) (out of 10)	Only PE (out of 3)	Only F2F (out of 7)
GG	7,56 = 2,68 + 4,88	1,87	-
NGG	6,64 = 2,33 + 4,31	1,49	3,71

Figure 3.1: Final grades of the GG and NGG students. PE is a practical midterm exercise, while F2F is the final face-to-face exam. The GG group performed better in both. The grading system is Spanish.

The study showed positive results, as the GG students performed better in both the midterm exercise as well as in the final exam. The difference is not dramatic, but an increase in the final grade of around 14% is significant nonetheless.

Just like cyber-ranges, the applicability of cyber games is very versatile. Cybergames can cover just about any cybersecurity-related topic and can be used by any user group, whether trained professionals or complete beginners. It simply depends on the game and the level of complexity it features.

Drawbacks and Challenges of Cyber Games

Developing well-structured cybergames is costly. It takes time, money, and a team of developers as well as domain experts [35]. Tailoring a custom serious game for specific needs of an organization can potentially be very expensive. The cost depends on a lot of factors: such as game mechanics, graphics and animation, game engine licensing, etc. Using custom tailored serious games to train the employee is not a common practice in organizations. This space is mostly occupied by cyber ranges.

Available offerings: Cyber Games

There are many free, freemium, and affordable cyber games accessible for everyone. Some of the many available options include: Centigrade [32], CyberGamesUk [36], and CyberGame [37]. All of these are available to the public and offer quality immersive, simulation, and scenario-based cyber games, as well as additional motivators such as competitions and rewards. The games cover a wide range of different cyber security topics,

such as phishing awareness, secure coding practices, incident response training, password security, etc.

3.1.3 Gamification

There is a distinguishable difference between gamification and cyber games in the context of cyber security training. While cyber games are full-fledged educational games, built from the ground up, gamification is simply enhancing the already available materials with game elements. Gamification makes learning more interesting and engaging through the integration of interactive elements such as points, badges, quizzes, flags, leaderboards, challenges, etc. [30][38].

Proper use of game elements significantly increases engagement and retention rates, students show higher motivation and sustained interest when learning through gamified methods [28, 30]. Integrating game elements into learning content is an effective method for increasing critical thinking skills. One good example is using Capture the Flag (CTF) competitions. L. Tobarra et. al. [29] integrated CTF gamification elements in a CyberSecurity University course. The participants had to solve a wide array of different tasks in various cybersecurity topics: cryptography, digital forensics, reverse engineering, and web exploitation. The demographics of the participants were categorized as follows: occupation, gender, age group, familiarity with cybersecurity. Around a third of participants were very unfamiliar with cybersecurity; however, more than half of them had a computer science-related job. Overall, the results indicated that incorporating gamified elements, such as CTF competitions, can be an effective approach to enhance learning outcomes and critical thinking skills in cybersecurity education within the Computer Engineering domain. Using a custom measurement framework the researchers set in the paper, the final results of the participant group were (M: 6.96, SD: 2.10), and the results of the non-participant group were (M: 5.95, SD: 2.21). M is for mean (average) score size, while SD refers to standard deviation. The participant group shows an improvement of 17% compared to the non-participant group.

Game-based learning increases the self-perception of the success of students and raises their motivation to be more engaging with the material. Integrated game elements like achievement tracking, reward points, progress tracking, etc. give the subjective sense of greater achievement, which leads to an increased motivation [30].

Personalized learning corresponding to the students' knowledge level is another problem that can be tackled quite effectively by using game-based systems. With the rise of AI in recent years, tailoring a personalized learning path has been made a lot easier. There are already existing AI-assisted Co-pilot systems that generate personalized exercises based on trainees' skills, guiding them towards areas requiring improvement and enhancing the learning experience [31].

Drawbacks and challenges of gamification

Since gamification is basically an enhancement of existing content, there aren't many drawbacks to implementing it. However, this does not make it a flawless solution. In some cases, the learners can miss the objective of the lecture. For example, M. Malone et al. constructed a framework for measuring students skills within an active learning setting. Although learning through gamification showed positive promises, there were cases where a semi-structured gamified exercise led learners to discover solutions that did not meet the learning objectives. The issue is that learners can sometimes get distracted and start exploring topics that are not part of the main objective. This can be remedied by providing the students with a clear structure and well-defined objectives, which they can use as a reminder to stay on track.

Another consequence of using gamification is a lack of randomization. Complex real-world tasks can be replicated only with limitations since gamified elements are not dynamic and are always pre-defined. This leaves a predefined amount of mistakes the learners can make, which sets boundaries to learning. Learners may not be encouraged to critically think about the task and may falsely overestimate their knowledge of a certain topic. If the gamified tasks are too rigid, it is challenging for instructors to pinpoint the weaknesses of the learners and assess their skill level.

Existing learning platforms with gamified learning content.

Kahoot! [40]: A game-based learning platform that enables the gamification of learning content by adding quizzes, puzzles, interactive games, real-time challenges, enriched multimedia, etc. It is probably the largest platform on this list, with more than a billion users, according to Kahoot itself.

Cybrary [41]: strictly focused on cybersecurity topics. It provides cybersecurity courses and online cybersecurity training tailored for individuals as well as larger

organizations. Cybrary includes gamified content such as structure learning paths, skill assessments , hands-on-lab exercises, CTF challenges, etc.

CyberStart [42]: Just like Cybrary, it is also a gamified cybersecurity-oriented learning platform. It offers engaging simulation-based game elements (an interesting example: hunting down cyber criminals), hands-on challenges, and it is based on real-world scenarios.

TryHackMe [43]: Learning cybersecurity through interactive lab exercises provides structured learning paths and self-paced learning to meet the individual demands of each learner. It also uses a very hands-on and practical approach, like most of the platforms mentioned here.

3.1.4 Virtual Reality (VR)

Virtual reality is a newer entrance to the already rich toolset for learning and teaching cybersecurity topics, and it is gaining a lot of attention in recent years as a simulation-based teaching method with some similar elements to a cyber range. The similarities between VR and Cyber Ranges lie in the fact that both of these learning approaches offer safe environments with no risks of damaging real infrastructure and are based on simulations of the real world [81].

The quality of VR is practically limited to the quality of the developed software designed to simulate the environment and partially the VR hardware itself. Since the learners immerse themselves in a simulated VR world, the cybersecurity topics that can be covered with the use of VR are theoretically endless. The unique advantage of VR is that it offers simulations with the highest immersion out of all the discussed approaches in this work, and it is constantly improving and evolving [82].

There are studies that back up the effectiveness of applying VR in the cybersecurity field. VR training can significantly improve knowledge and performance in identifying cyber threats, according to the studies of Datel et al. [83]. In this case study, seventeen college ROTC Navy midshipmen used VR for a period of 8 weeks, with cybersecurity training taking part two times per week. The provided VR application was an immersive application of a U.S. Navy destroyer. Three different systems were hacked during the simulation trainings: Voyage Management System, Radar Detection System, and Automatic Detection System. After each session, the participants held discussions with their peers and played the role of instructor. After the 8-week-long training, the participants saw a significant improvement in their knowledge of identifying cyber threats.

Another use case of integrating VR in cybersecurity is the CiSE-ProS VR [81], whose objective is enabling computer scientists, developers, and engineers to develop cybertraining programs with a focus on cyber security. It uses the well-known, consumer-available HTC Vive for its hardware. The main user experience consists of four activities: 1) tutorial, 2) entering/exiting the data center, 3) inspecting the data center, and 4) replacing hardware. Which means that this particular VR teaches hardware-oriented cyber security. Activities such as inspecting racks, replacing RAM sticks, and checking the "Data Center" can be simulated. This is a very strong advantage of VR; while other methods can provide hands-on activities for software-related tasks, they are limited to theoretical interpretation when it comes to teaching about hardware.

Integrating VR with other technologies, such as AI and blockchain, can further enhance the learning experience of the participants. Such integration has the potential to further improve the security and simulation of the environments.

Challenges and Limitations of Virtual Reality.

Current VR training systems lack automated performance assessment tools, which means that the learners can't get constructive feedback in order to identify potential gaps in their knowledge [84].

Although intuitively we can conclude that VR training improves practical knowledge, there is still not enough empirical evidence supporting this fact. The effectiveness of transferring skills obtained from VR training to real-world tasks is still not a very well researched topic. This suggests that there may still be available room for improvement in VR and cybersecurity integration.

So far, the use of VR is very promising; it is not the cheapest solution, and some organizations and institutions may still be hesitant to use it. But if we follow the trend of the more immersive and more realistic a simulated environment is, the higher the engagement of the learners, and consequently, the higher their knowledge retention [85]. However, since it is still a relatively new technology, it needs to overcome some challenges. But the potential of this method is undeniably very high, with a very promising future.

3.1.5 Hands-on activity methods

Hands-on activity is a very practical-oriented approach with the purpose of increasing the understanding and utilization of practical skills in real-world applications. These activities

range from virtual labs and simulations to competitions and mobile device-based exercises, all designed to enhance learning and engagement.

The hands-on learning method gives the learners the opportunity to practice their skills in a safe environment like a virtual laboratory. To clear things up, hands-on activity is a broader term. As an example, cyber ranges can be classified as a hands-on activity method. The hands-on approach aims to improve problem-solving skills with simulation-based exercises like penetration testing or social engineering. The learners may put themselves in the position of the attacker, thus receiving valuable insights on how to counter the threats.

Cloud-based virtual labs like V-Lab offer scalable, flexible, and reconfigurable environments for network security education, accessible remotely [87]. V-Lab, as well as many other cloud-based labs such as Blue Team Labs and Over the Wire [91, 92], support many experiment-based, hands-on exercises that have shown that they are capable of significantly improving the learners practical knowledge. They offer simulation-based safe environments, often called sandbox environments, collaboration tools, incidence response resources, analysis tools, etc. Often these cloud-based labs have a free-tier offering, which makes this a very accessible method for learning.

Mobile device-based labs provide an affordable, sustainable, and accessible solution for hands-on learning, allowing students to practice anytime and anywhere. Low-cost equipment and open-source software can be used to create effective hands-on educational activities, making cyber security education more accessible.

Hands-on training systems tailored for beginners help students learn cyber security concepts without the need for advanced knowledge or instructor presence. Cyber-physical systems (CPS) security education can be enhanced through hands-on labs that simulate real-world scenarios, such as vehicle control system vulnerabilities.

A study conducted by Konak et al. emphasizes the importance of hands-on exercises and highlights their effectiveness [86]. The participants were K-12 level students, with the aim of preparing younger generations to satisfy the increasing demands of digital security.

The findings are portrayed in the following table:

		Difference (Post / Pre)		95% Confidence Interval of the Difference		
Learning Outcome Area	**Percent Increase in Means**	**Mean**	**Std. Dev.**	**Lower Bound**	**Upper Bound**	**t**
System Administration	**43%**	**38**	**31**	**29**	**48**	**8.02**
Networking	**200%**	**54**	**28**	**45**	**63**	**12.15**
Cyber Threat Identification	**26%**	**22**	**28**	**13**	**31**	**5.04**
Cryptography	**141%**	**45**	**31**	**35**	**54**	**9.23**
Test Overal	**94%**	**38**	**19**	**32**	**44**	**12.91**

Figure 3.2: Comparison of the pre-program and post-program test results of the study conducted by Konak et al.

The students who took part in the hands-on exercises showed a substantial improvement over the group that didn't. The topics covered were system administration, networking, cyber threat identification, and cryptography. Overall, an increase in obtained knowledge has been accomplished in all of the subjects (results visible in the table, Figure 3.2).

Drawbacks and challenges of activity-based learning

Overall, hands-on activities in cyber security education are a very accessible and valuable method for developing practical skills and understanding real-world applications. The drawbacks of this method are not that specific and not something that was not mentioned before. A limiting factor is the quality of the design of the exercises. Poorly designed hands-on activities may not align with real-world applications, resulting in knowledge gaps. Another downside may be that students may rely too much on relying on predefined

structures and following a set of step-by-step guides. This reliance results in passive learning and avoiding critical thinking about a problem, which can result in further knowledge gaps.

3.2 Comparison of CyberSecurityBoard with other approaches

Storyboarding techniques are rather underexplored, especially when it comes to cybersecurity education. There are many research papers related to storyboarding techniques in education. However, there are barely any study cases where storyboarding was implemented for cybersecurity education in contrast to the other discussed approaches.

Because of this, the motivation to conduct a small survey arose, as gathering insights from participants would help clarify if the storyboarding method is heading in a good direction. A small user study was conducted where 25 participants were asked to navigate and experience a storyboard created with CyberSecurityBoard. The final results gave a positive impression. The participants were mostly satisfied and agreed that the storyboard content provided in CyberSecurityBoard is a feasible way of teaching cybersecurity topics. The results below are only a part of the survey that has been conducted for this work:

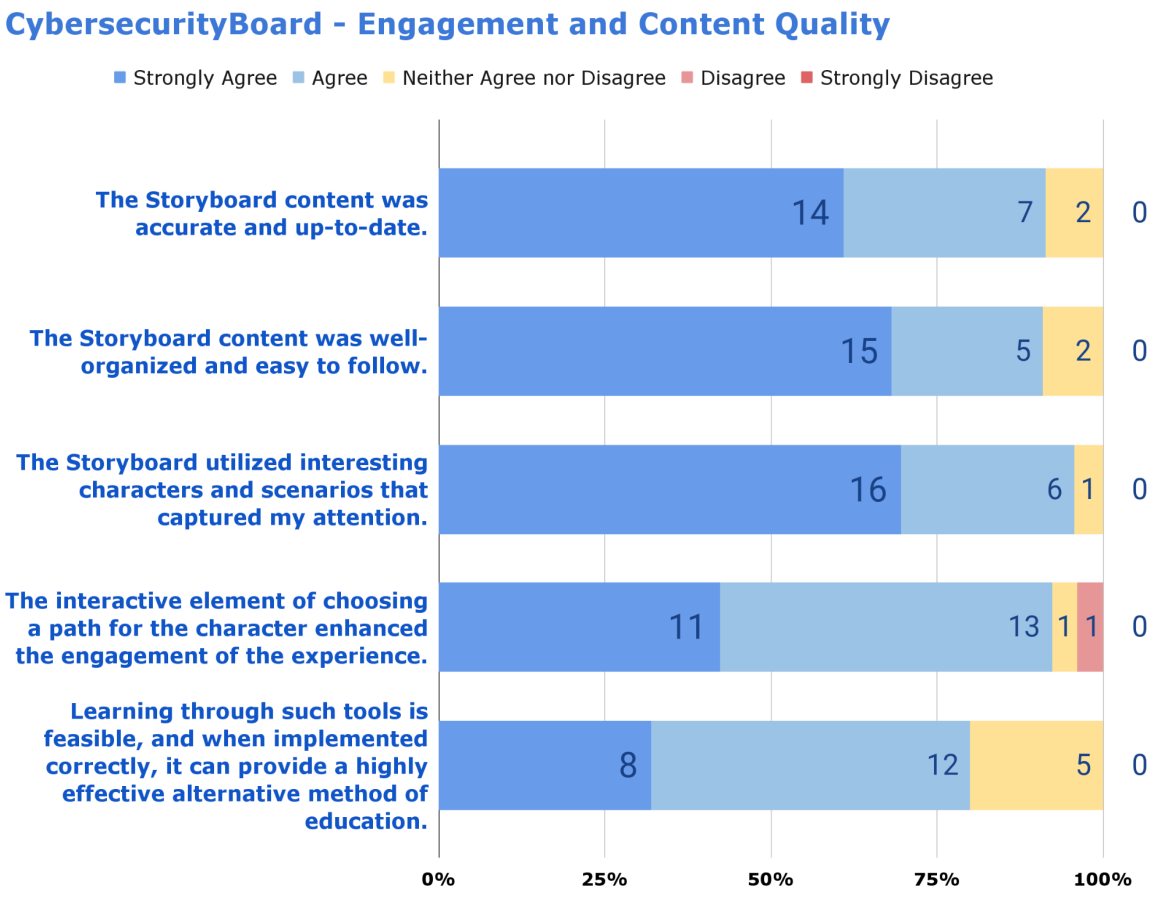

Figure 3.3: Survey results for engagement and content quality of a storyboard created with CyberSecurityBoard, answered by 25 participants.

The survey takers felt that learning through an interactive storyboard is possible and that it has good potential. The overall feedback is positive, with very few negative or neutral remarks. The participants also felt that the provided content has still some room for improvement.

3.2.1 Comparison Framework

From the evaluated training methods and technologies, there is a noticeable pattern of the strategies and characteristics that are being used. The most emphasized were engaging content, interactivity, real-world simulation-based scenarios, and hands-on exercises. Currently there isn't available research that directly compares the discussed methods, and it makes little sense to do so since the methods are not exactly directly comparable. Choosing one as the ultimate teaching method is impossible in this regard. Each of them serves their unique purpose and field of application. A more sensible comparison would be to pick meaningful key characteristics and investigate how well they stack up against each other. For this purpose, a framework has been constructed to hypothetically make the comparison and give us an idea where each of the methods stands.

This custom-constructed comparison framework helps us figure out if storyboards (created with CyberSecurityBoard) offer a meaningful addition to what is already available, and if so, where does their field of application lie? By evaluating the education approaches mentioned in this chapter, together with traditional methods as the base for comparison and CyberseCurityBoard storyboards. The following characteristics have been taken into consideration as the most meaningful ones for the comparison:

1. **Engagement**: How much is the provided content interactive? How much is the learner actively involved? Are there interactive tools, exercises, and similar factors involved in the learning process?
2. **Effectiveness**: This category focuses on the knowledge retention rates of the learners. How much practical knowledge does the learner obtain after finishing training with a respective method?
3. **Applicability**: This category takes into account the range of cybersecurity topics that the training method covers and which target audience it can reach. Up to what level of detail can a topic be effectively explained, the field of cybersecurity where it can be optimally applied, etc.

4. **Cost**: This involves the financial investment required to implement the training method. It includes direct costs such as the hardware, software, and instructor costs, as well as indirect costs like time and resource allocation. Cost plays a huge role in how much a cybersecurity training method is accessible.

It is extremely difficult to represent the comparison points between each training method in exact numbers since a wide range of factors may contribute to each point. Instead, a more generalized approach has been adopted to rate engagement, effectiveness, applicability, and cost. The ratings are categorized into levels: low, medium, and high, but also include intermediate ranges, such as low to medium and low to high, to provide a more detailed assessment. This generalized rating system allows for a flexible and holistic comparison, accommodating the diverse nature of cybersecurity training methods and the varied needs of different learners and organizations. The ratings are a rough estimation based on what was stated about these approaches in the first part of this chapter (3.1). Based on this, here are the compiled results for each learning approach and the argumentation for its rating:

Traditional:

- **Engagement**: Low to medium, book learning and listening to lectures are usually non-interactive. Exercises and discussions improve the engagement.
- **Effectiveness**: Low to medium - reported lower retention rates in comparison to other methods
- **Applicability**: High - The oldest method naturally has a high applicability. Any topic can be learned in theory.
- **Cost**: Medium - traditional methods are generally less expensive to implement, but purchasing books and recruiting trainers increases costs.

Cyber Ranges:

- **Engagement**: High - offers the best real-world simulations and infrastructure that can be indistinguishable from the real one.
- **Effectiveness**: High - very realistic simulations provide very high retention rates.
- **Applicability**: Low to medium - ususally implemented for more complex, more advanced topics in mind.

- **Cost**: High - expensive hardware, implementation, and maintenance costs make this solution very expensive. Usually accessible for larger institutions that have more resources.

Cyber Games:

- **Engagement**: High - captivating game elements can hold the learners attention for very long periods.
- **Effectiveness**: Medium to High - real-world simulations and interactive content offer high retention rates. The realism is on a lower level compared to cyber ranges.
- **Applicability**: Medium - a cyber game is usually limited to a number of topics.
- **Cost**: Low to High - a cyber game can be completely free and available to anyone, or it can be custom developed for an organization's needs, which depending on the requirements, can be very costly.

Gamification / Gamified Content:

- **Engagement**: Medium to High - very engaging, but not as immersive as some other methods. Also does not simulate the real world.
- **Effectiveness**: Medium - definitely a step above traditional book learning but still less engaging than a full-fledged cyber game or a cyber range.
- **Applicability**: High - pretty much any content can be enhanced with game elements.
- **Cost**: Low - takes little time and not much effort to implement gamified elements in learning content. There are many tools, free of charge, that can gamify study content.

Virtual Reality:

- **Engagement**: High - very realistic, very immersive experience.
- **Effectiveness**: Medium to High - there is not much research backing up the retention rates of VR, but orienting by the trend higher engagement = higher retention rates, VR should achieve quite high in this regard.
- **Applicability**: Low to Medium - development of VR games is very complex; currently, there aren’t many VR games available that teach cybersecurity.
- **Cost**: Medium to High - high hardware costs, developing games for VR takes a lot of time and has high costs.

Hands-on Exercise:

- **Engagement**: Medium to High - practice-oriented and interactive, but not as immersive as VR or cyber ranges.
- **Effectiveness**: Medium to High - research shows very high reported
- **Applicability**: High, as hands-on exercises can cover many topics and skill levels.
- **Cost**: Medium, relatively affordable, requires resources for designing and setting up the exercises

CybersecurityBoard:

- **Engagement**: Medium - offers interactive, real-world simulations acted out by drawn characters. Lacks gamified components and interactive exercises.
- **Effectiveness**: Medium to High - according to the conducted survey, most of the participants reported positive results.
- **Applicability**: Medium to High - very versatile in covering almost any topic. Text, code snippets, images, and symbols can cover a very wide range of cybersecurity topics.
- **Cost**: Low - little time needed to create a storyboard, everything is provided in a single place. No need to purchase additional tools or materials.

	Engagement	**Effectiveness**	**Applicability**	**Cost**
Traditional	Low-medium	Low-medium	High	Medium
Cyber Ranges	High	High	Medium	High
Cyber Games	High	Medium-High	Medium	Low-High
Gamified Content	Medium-High	Medium	High	Low
Virtual Reality	High	Medium-High	Low-medium	Medium-High
Hands-on Exercise	Medium-High	Medium-High	High	Medium
CybersecurityBoard	Medium	Medium-High	Medium-High	Low

Figure 3.4: Results of the comparison framework; The ratings are an estimation based on the research stated in Chapter 3.1.

From the estimated results, it is visible that there is no best approach. Each of them can be effectively utilized in different scenarios. Overall, CybersecurityBoard has the potential to do quite well in comparison to other available training methods. It especially excels in how quick and easy it is to create learning materials from scratch. Cybersecurity covers many different levels of learners; however, the most recommended utilization of the cybersecurity board is to provide learning materials to beginners and intermediate-level learners. The strategy of delivering content of storyboards is to provide smaller pieces of information per storyboard window. For this reason, complex topics should be avoided, since designing a storyboard for large, complex, and advanced topics will result in a very large storyboard.

3.3 State-of-the-Art tools similar to CyberSecurityBoard

Although one may argue that there are plenty of tools available that can create educational storyboards, that is only partially true. Creating a static storyboard can be accomplished with dozens of free or premium tools already available, but the process of doing so may not be as straightforward. With the currently available tools, the user is likely to go through the process of:

- Purchasing or creating custom graphics for specific needs,
- Searching for custom graphics through third-party providers,
- Uploading the graphics to the platform

And even after all these steps have been taken care of, the storyboard will be a static storyboard with no interactiveness. And the value of interactive learning is way too valuable and beneficial to be left ignored.

The following table is a collection of the most popular platforms/tools that can be used to create a storyboard:

Name	Field	User Interaction	Builtin IT/Security graphics	Builtin Character graphics
Canva	General	No	Yes	Yes
Milanote	General	No	No	No
VisMe	General	No	No	No
StoryBoardThat	General	No	No	No
Boords	General	No	No	No
Creately	General	No	Yes	No
CyberSecurityBoard	CyberSecurity	Yes	Yes	Yes

Figure 3.5: Comparison of CyberSecurityBoard to other similar tools

The tools and platforms shown in the table are a mixture of general content creation tools and specialized tools for creating storyboards. Although many of them check a lot of criteria, they still lack many beneficial features in contrast to CyberSecurityBoard. Creating a storyboard that is instantly available to other users, cybersecurity-themed graphics, and interactive storytelling are features unique to CyberSecurityBoard.

CHAPTER 4

Modeling of cybersecurity storyboards

This chapter goes in depth, showcasing the features of CybersecurityBoard. Since the implemented tool is meant to be used by both instructors and learners, two different views have been developed to meet the needs of both user groups. Throughout the chapter, the perspective developed for the instructors will be referred to as “Creator Mode" and the perspective developed for the learner will be referred to as “Viewer Mode”. The second part of this chapter presents the results of a user experience survey, and the final part offers recommendations to optimally create effective storyboards.

4.1 Navigating through CyberSecurityBoard

CyberSecurityBoard is a tool for creating and exploring interactive and engaging storyboards. It is developed as a Web-based application, optimized for PC. In order to optimally use every feature, it is necessary to have a keyboard and mouse. Even though the tool can be opened and partially used on mobile devices, the experience will be limited.

Since this tool is a prototype, it mostly contains the core features needed to meet the demands of this work. Cybersecurity offers the following features: user registration, built-in graphics (covering characters, cybersecurity-themed symbols, and other types of graphics that were deemed necessary), creating storyboards with a simple drag-and-drop user interface, viewing the created storyboards by other registered users, and an and an interactive component where the user gets to make a decision for the character within the story. The upcoming chapters explain all of these features in a guide-like manner.

4.1.1 Basic Navigation

When visiting Cybersecurtyboard [44], the user is greeted with the home page, as shown in the figure 4.1:

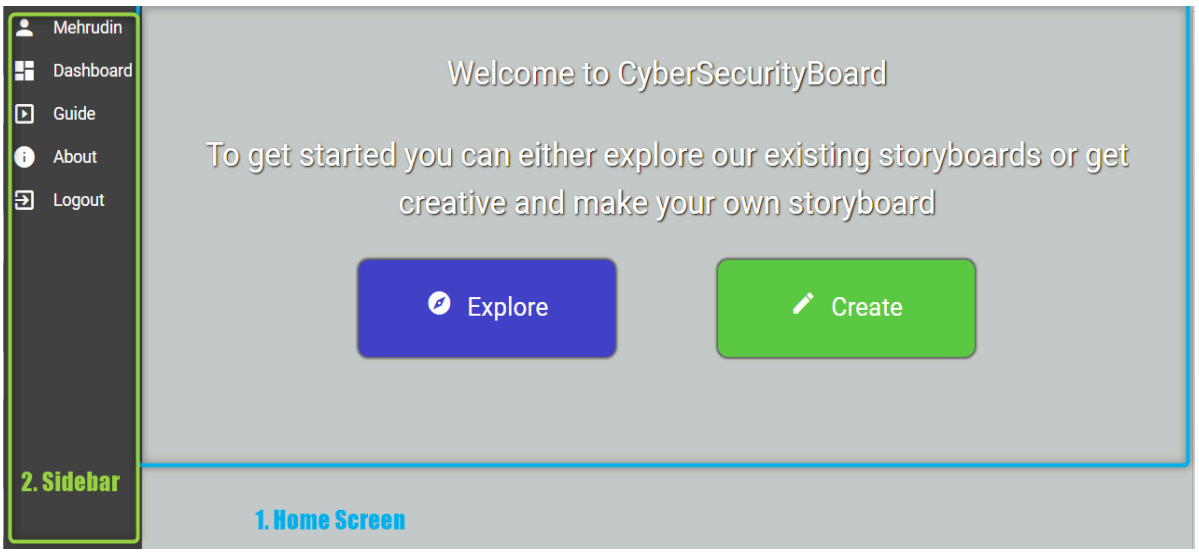

Figure 4.1: The home screen of CyberSecurityBoard consists of two areas: 1. The home screen and 2. The navigation sidebar

The home screen design takes on a minimalist approach, offering the user just two options: “Explore” and “Create”. The design decision behind this simple screen is to provide the user with only as little as necessary information. The two big buttons are meant to capture the user's attention to the core functionalities of the tool; the smaller details and features are slowly introduced along the way. The intention is to make the tool as accessible as possible to a wider target audience and to minimize confusion [45].

The other component that is constantly visible on any page is the navigation sidebar on the right. The navigation sidebar navigates to the following pages:

- Username: navigates to the home page (Figure 4.1).
- Dashboard: The dashboard holds a repository of storyboards created by the user that is currently logged in.
- Guide: This page shows in-depth, step-by-step documentation on how to use CyberSecurityBoard and its features.
- About is the standard about page; it contains a brief summary of what the project is about.
- Logout: standard logout feature to end user session.

In case the user is not logged in, only two tabs are visible on the sidebar: “Register” and "Login", which are standard options for creating an account and logging in with existing credentials.

4.1.2 Creating Storyboards: Instructor Perspective

A registered user has both “Creator” and “Viewer” access rights, which means any user has the right to create their own storyboards or explore storyboards created by other users.

Managing already-created storyboards is handled in the Dashboard. The dashboard shows only the storyboards created by the currently logged-in user with granted CRUD rights.

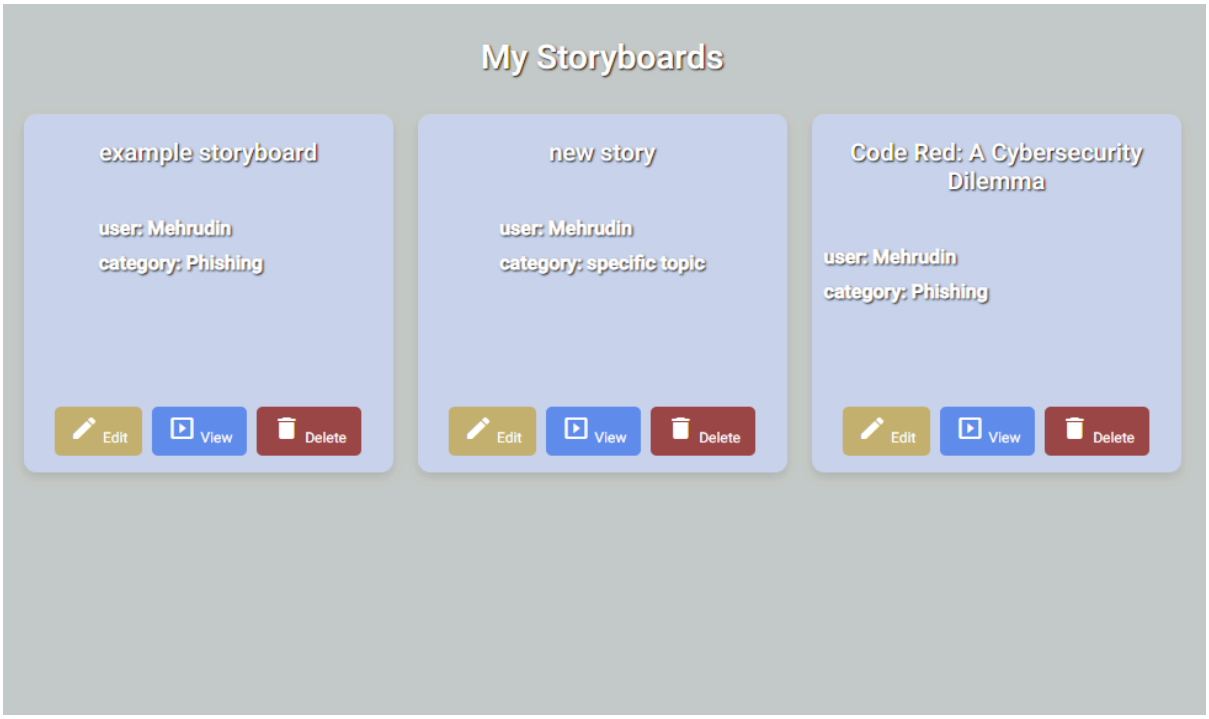

Figure 4.2: Dashboard page for managing storyboards

To create a storyboard, the user has to open the “Create” page (Figure 4.1). After starting the storyboard creation process, the following window pops up:

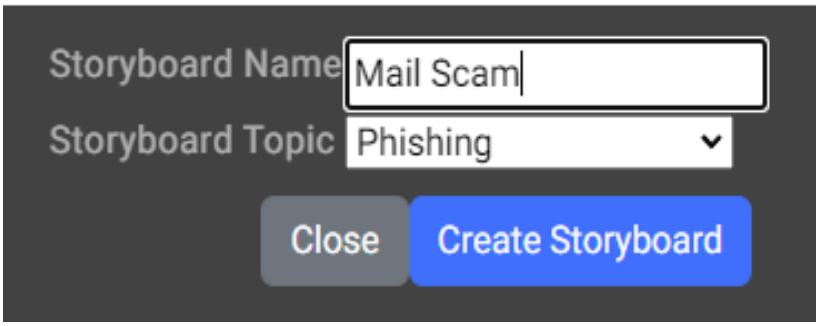

Figure 4.3: Popup window before creating a storyboard

In the input fields, the user has to choose a name and a topic for the storyboard. The preselected topics include the most popular cybersecurity threats [Chapter 2] [46] [47]: Malware, Phishing, DoS/DDos Attacks, Insider Threats, Social Engineering, Zero Day Exploit, IoT Vulnerabilities, Supply Chain Attacks. If the pre-selected topics do not contain the topic the user wishes to use, manual input is also allowed. After pressing “Create Storyboard," the process of creating a storyboard is already started. An entry in the database with the storyboard name, topic, and the user who created it will be written. The user will be automatically redirected to the “Creator Mode” view, which looks like this:

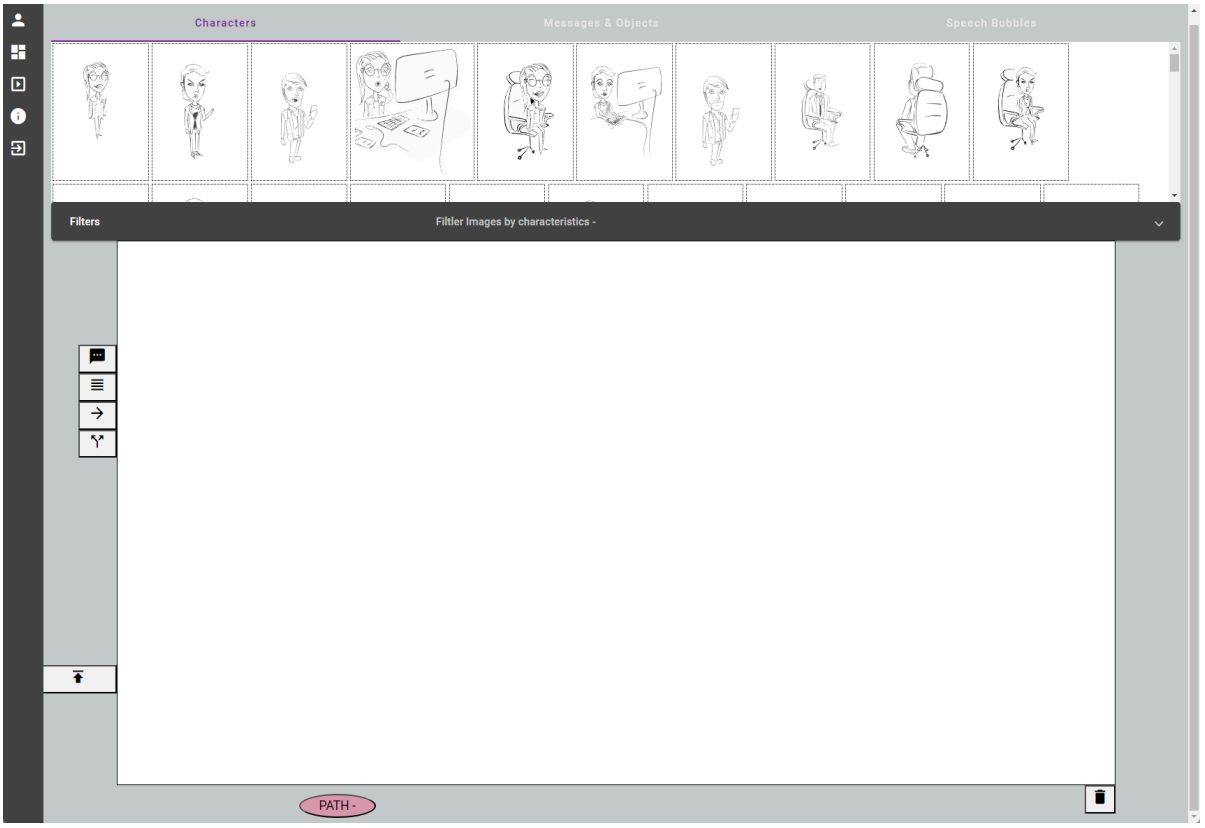

Figure 4.4: CyberSecurityBoard, Creator Mode

The “Creator Mode” view is a more complex view since it introduces a lot more elements on the screen. Again, the design decision for this view aims for a minimalistic representation in the UI; in order to reduce complexity, some components are displayed only on demand (such as the filter bar, graphics in the top bar are categorized into three tabs, two of which are hidden at a time, etc.).

“Creator mode” is essentially composed of four segments:

1. Graphics topbar

This part contains all graphics that can be dragged and dropped inside the storyboard.

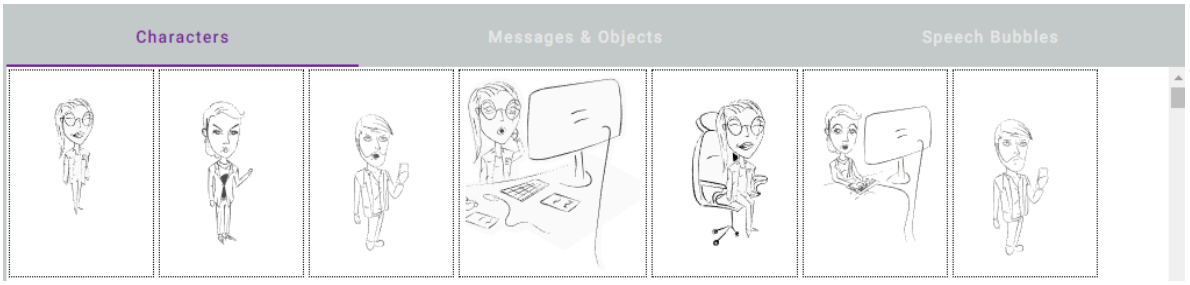

Figure 4.5: Graphics Topbar

The graphics topbar has three navigation categories:

- Characters: This tab contains all human characters. There is a wide range of characters, with different personality types, professions, genders, etc. To convey a more impactful message [58], the provided characters come with facial expressions that show their emotions. The characters are drawn in five different versions, with different facial expressions portraying the following emotions: angry, happy, surprised, scared, and neutral. All of the characters are provided by the Coltrane project [48].
- Messages and Objects: all symbols and objects related to cybersecurity scenarios, such as PC screens, hacking and vulnerability symbols, message types, emails, etc., are contained in this section.
- Speech Bubbles: As the name implies, it contains all graphics related to speech bubbles, which are used to showcase a character is either thinking or talking or to simply explain a scene as the story narrator.

The tab that is opened by default is the “Characters” tab; the other tabs are opened and loaded on demand, thus reducing initial load times and complexity of the UI.

2. Storyboard Window

The storyboard window is the space where characters and objects can be dragged from the topbar and inside the storyboard window. The user may change the size and position of an image that is within the storyboard window.

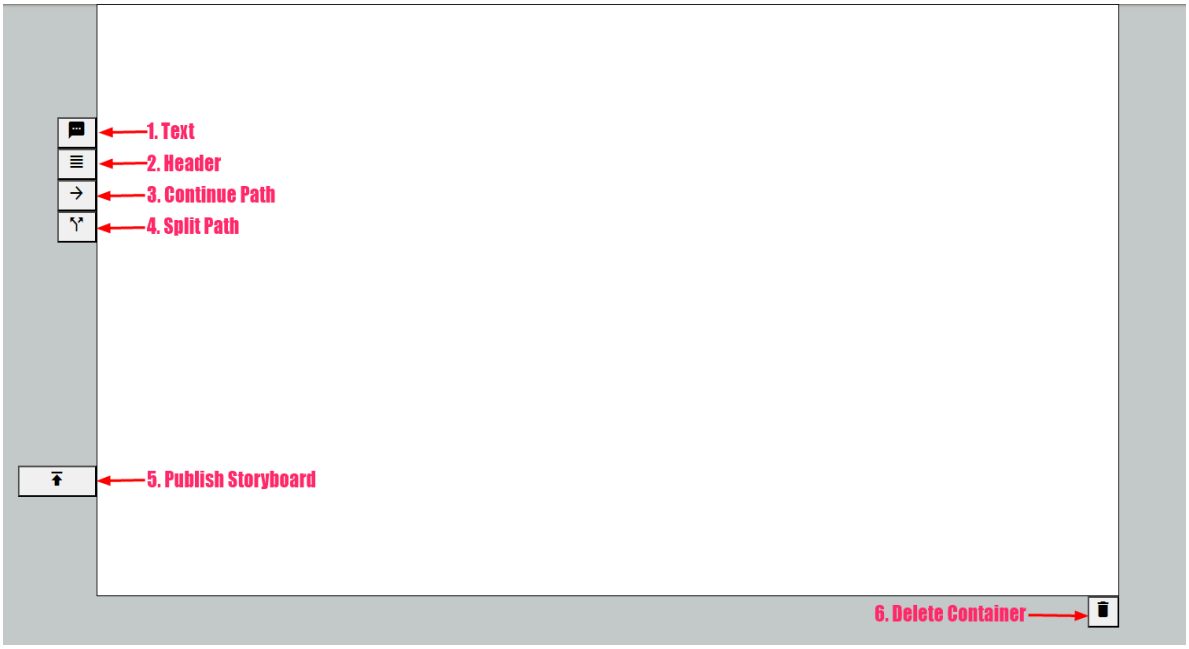

Figure 4.6: Storyboard Window

The six buttons shown in the figure above have the following functionalities:

1. Text: inserts a regular text object within the storyboard window.
2. Header: inserts a header text object within the storyboard window. It simply has a different font size and style suited for creating titles for a storyboard.
3. Continue path: Continues the storypath within the single story line, or simply put, it just creates a new single storyboard window.
4. Split path: divides the story into N number of different outcomes; N number of windows for each outcome are created in this case. A more detailed explanation of the story paths will be covered in this and the following chapter.
5. Publish Storyboard: saves all the changes and displays the storyboard in "Viewer Mode".
6. Delete container: Deletes the current active container and all the objects within it.

3. Filter dropdown bar

Since CyberSecurityBoard contains many images, finding the right one may be time-consuming. In order to solve this problem, the filter bar has been implemented. It retrieves only the images that are of the user's specific interest. This view is hidden and is expanded on demand. When expanded, it looks like this:

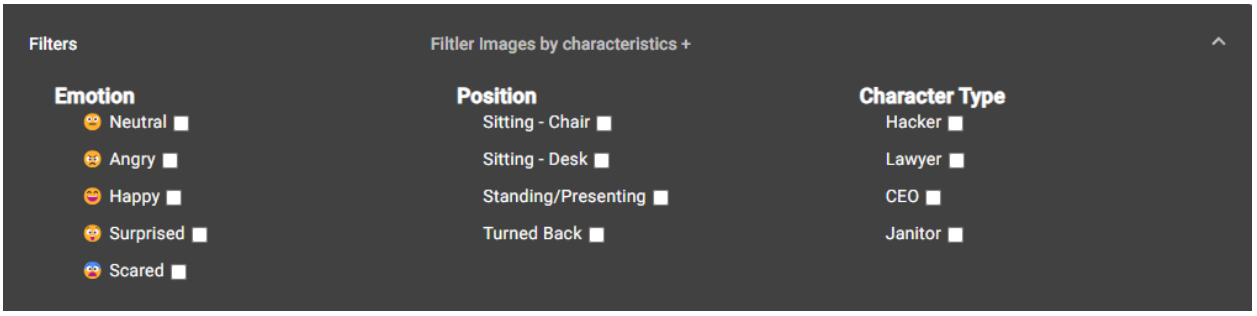

Figure 4.7: Filter bar

The Filter bar dynamically changes content based on the active bar in the Graphics Topbar. In the case of Figure 4.7, it shows the filter for when the "Characters" tab is active and therefore displaying filters related only to characters. These filters include:

- **Emotions**: Almost every character has versions representing five different emotions.
- **Position**: indicates the character's physical stance, such as sitting or standing.
- **Character Role/Type**: filters for specific character types/roles like hacker, CEO, lawyer, etc.

If the active bar in the Graphics Topbar (Figure 4.5) is set to “Messages & Objects,” various filters related to messages and objects will be displayed. These include symbols for different types of messages and categories of objects, such as smartphones, PC screens, etc.

4. Story path tree

Story path is a feature unique to CyberSecurityBoard; it can hold an N number of alternate story time lines, each one telling a different story. As an example, let’s say a character needs to make a decision to trust an unknown source link, whether to open it or not. Split path with N = 2 will create two storylines, one where the character opens the link and the other one where the character does not trust the source and does not open the link. This gives the learner an opportunity to simulate the consequences of a right and wrong decision, thus increasing engagement and effectiveness of learning [49]. In theory, the number of paths could be unlimited; however, for the sake of a stable user experience, the number of story paths has been limited to 5. Here is an example of how a storypath may look like:

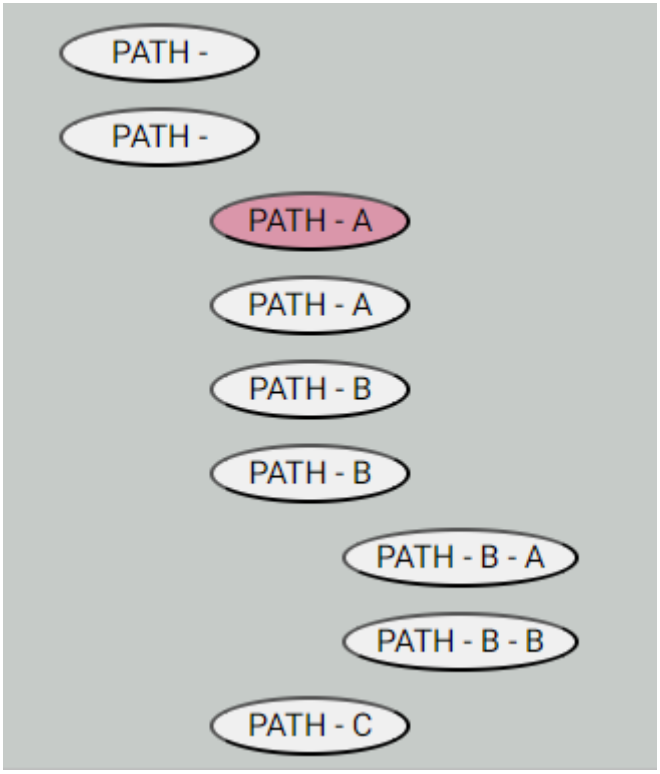

Figure 4.8: Story path navigation in "Creator Mode"

The storypath is located around the Storyboard window and is visually represented by a collection of buttons arranged in a pattern resembling a structured tree diagram. Each time a path has been created using the "Continue path" option, only a single button gets created in the same row as the previous one. However, if a path using the "Split path" option gets created, at least 2 new paths get created; the new paths are considered children or sub-storylines relative to the path they were created at. The children's paths are always displayed below the parent and spaced to the right. To better explain how it works, consider the visual representation in Figure 4.8, from which the following conclusion can be drawn:
The initial storyline, "PATH," progresses in a single narrative until the second slide, where it branches into three sub-storylines: "PATH-A," "PATH-B," and "PATH-C." "PATH-A" consists of two slides, concluding its narrative on the second slide. "PATH-B" continues until its second slide, where it further splits into two additional paths. In a more realistic example, the tree would be much larger with more nested storylines.

A story path is created by simply pressing the "continue path" or "split path" button (described in Figure 4.5). The continue path sets the N to 1, and no input is required; it simply continues the story along a single storyline. On the other hand, "split path" not only

creates multiple storylines, it also creates the following objects on the slide where it was utilized:

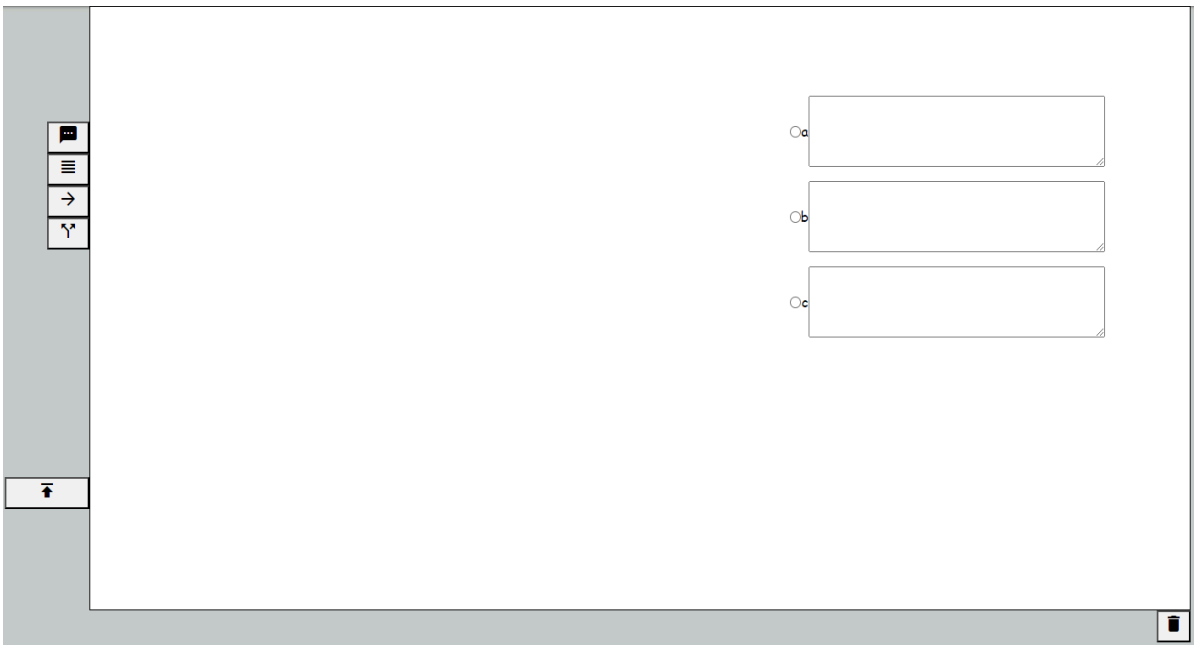

Figure 4.9: Slide where storypath with N = 3 was created

From the image above, we see that three input boxes along with three buttons labeled a, b, and c were created. The input boxes allow the instructor to describe the path choice, and the letter is used to identify the path. Clicking on the button will navigate the user to the respective storyline path. The algorithm of the "split path" feature will be explained in Chapter 5 in more detail.

4.1.3 Navigating a Storyboard: Learners Perspective

The "Viewer Mode," or the view designed for the learners, is essentially a version of the "Creator Mode" with removed features related to creating a storyboard and a different user interface. To open a storyboard in this mode, the user needs to open the "Explore" page from the main menu. This page shows the repository of all the storyboards created by all registered users. It looks like this:

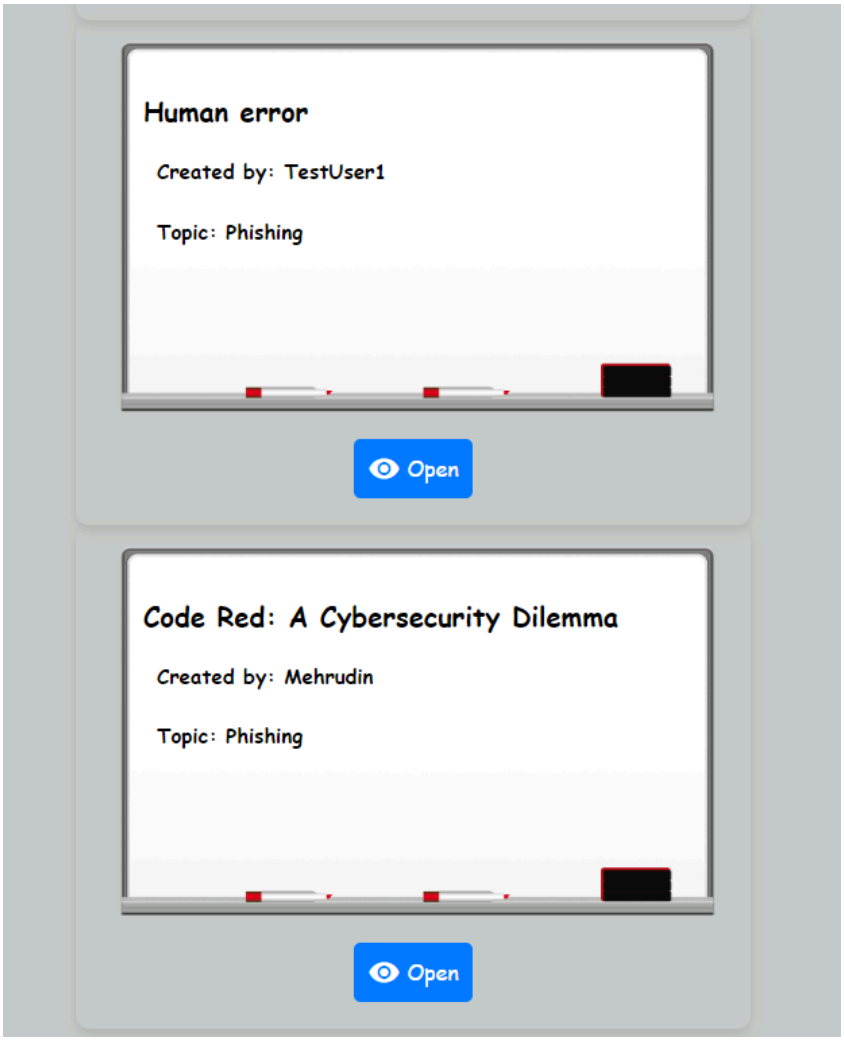

Figure 4.10: The explore page of CyberSecurityBoard

A storyboard is identified by its title, the creator's username, and the related topic that the storyboard is about. The whiteboard image used for displaying the storyboards is a free graphic provided by Pixabay [50].

Opening the storyboard navigates the user to the following view:

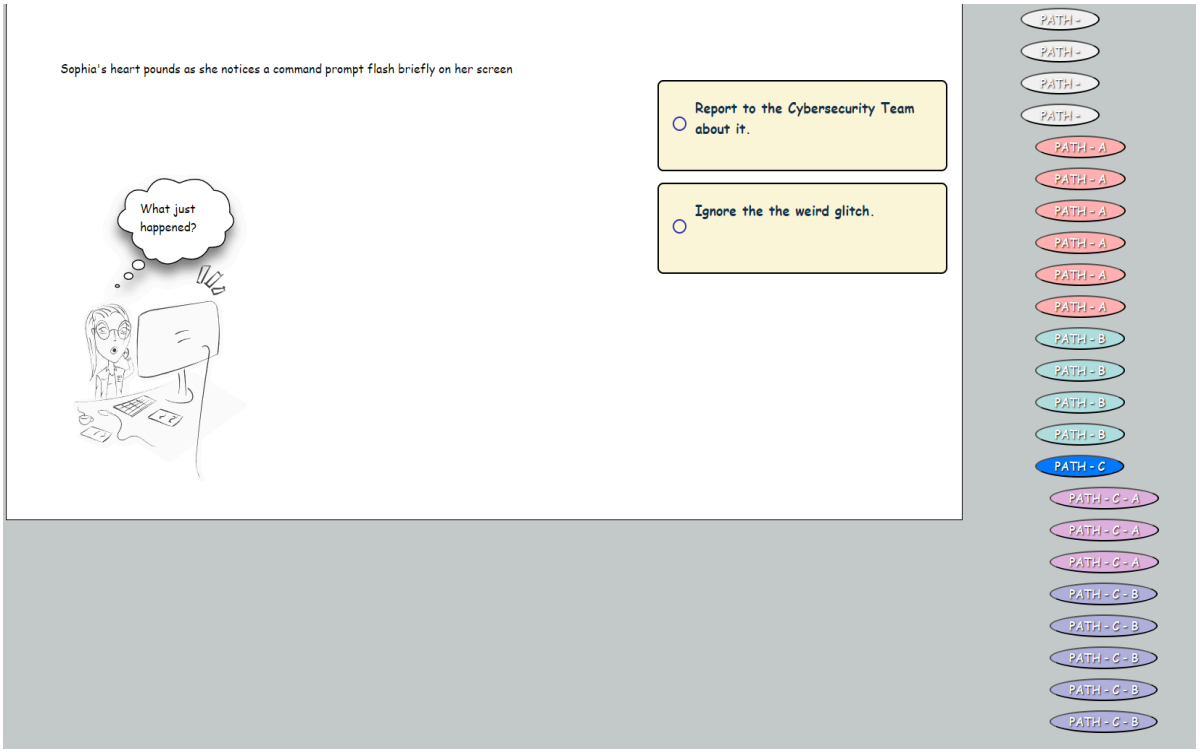

Figure 4.11: Storyboard in "viewer mode"

There are many notable similarities between "creator mode" and "viewer mode." What's different is that the topbar for dropping graphics is removed along with the filter, the user is not allowed to manipulate the images inside the storyboard, the design of the path tree is colored for more intuitive navigation, and the generated decision buttons have a different style that corresponds more to the storyboarding theme.

4.1.4 User experience and Usability surveys

A survey on the experience of using CyberSecurityBoard has been conducted. This survey was taken by 25 participants as the second part of the survey mentioned in Chapter 3. The participants were asked about their background and experience using similar tools. Around two-thirds had either an IT background or some experience using the tool; the remaining participants did not have any experience. Before taking the survey, the participants were given short instructions on how to use the tool; alternatively, the guide page from CyberSecurityBoard has been provided as well, with the intention to provide the less tech-oriented users with very clear, more detailed instructions on how to use the tool. The responses were overall positive from both groups; this is a positive indication that the minimalistic approach to designing the tool was a successful strategy. Overall, most users were able to create a storyboard, while only a few struggled with using the split path feature.

Most participants agreed that they perceive CyberSecurityBoard as a tool that can be used to transmit knowledge. A few participants felt that improvements could be made regarding additional features. The result of the survey is portrayed below.

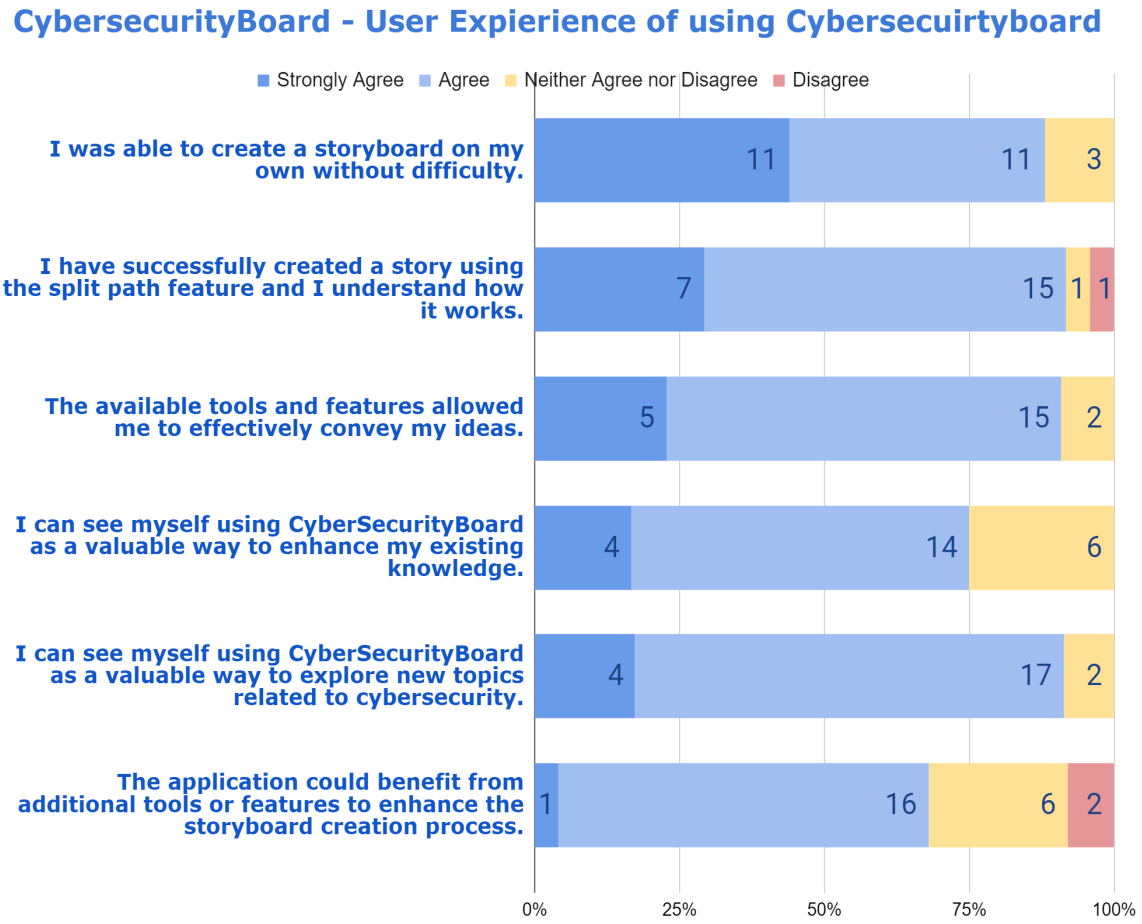

Figure 4.12: User experience survey results

Survey participants were asked to provide feedback on their overall subjective experience of using CyberSecurityBoard, focusing on its general functionality and user interface (UI). The main purpose of this survey was to check if the current version has an acceptable level of usability and to pinpoint the weak points for future improvements.

The results of the survey can be observed in the below figure:

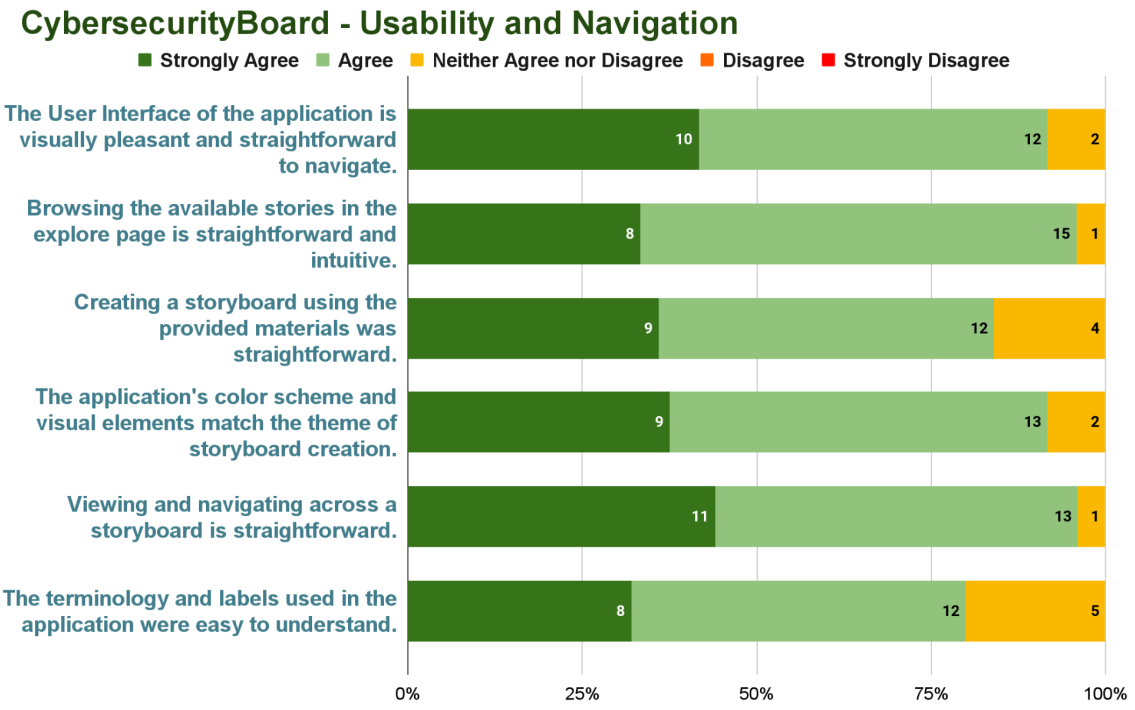

Figure 4.13: Usability and Navigation survey results

The feedback was generally positive, and only minor bug fixes and UI improvements have been made since the survey publication.

4.2 Modeling a Storyboard: Techniques for Creating Engaging Study Content

The first part of this chapter explains how to technically use the software, but that's only half the story. Understanding the techniques of good storytelling is just as important. So the following question needs to be answered as well: How to optimally utilize storyboarding techniques to create the most engaging learning materials with the best possible knowledge transmission. It is well established that storytelling techniques boost the students cognitive abilities [51]. Integrating interactive components, such as making a decision in the place of the character in the story, CyberSecurityBoard, further amplifies the immersiveness of storytelling. To create an optimal storyboard with CyberSecurityBoard, practicing good storytelling patterns is highly recommended. The proposed structure for creating a storyboard is categorized in the following points:

- State the learning objects as clearly as possible.

This will guide your storyboard's content and structure. Clear objectives assist in creating an absorbed and coherent narration that touches on the intended learning results directly.

- Understand the target audience

Know who your students are and what interests them to make the story interesting. Take into account their past experiences, culture, and preferences. The content of the storyboards needs to be adapted on an individual level of the class. The content becomes relevant to the participants, thus meeting their educational requirements well.

- Integrate characters that are interesting, familiar, and relatable.

Introduce characters who share similar problems or go through similar experiences as the participating students. Give them different personalities, backgrounds, and goals so that they become relatable. When characters can be related by students into a story through having distinct personality traits, backgrounds, goals, etc., then the learning experience becomes more immersive and personal.

- Create a compelling narrative.

Write a script that captures the learners attention, describes the scenery, simulates realistic conversations, messages, notifications, etc.

Writing scripts for the stories was heavily inspired by the "Ed-W" modeling methodology for designing educational stories, developed by Maria et. al. [52]. In this model, a five-step guide is proposed, each step chronologically describing how the story should unfold for effective story writing. The model has been slightly adjusted for this case by taking into account the interactive component where the learner has the option to take a decision and observe the consequences of the made decision. Most of the steps remain the same; steps 2 and 3 have been slightly altered to be more suitable for the interactive storyboards. These are the steps to the proposed story writing:

Step 1: Introduction to the Problem, facing a challenging situation.

In the initial phase of the story, the hero encounters a challenging situation where they need to act by applying their special skills or knowledge required in the field.

Step 2: Learners directly impact the story with their decisions

There are parts in the story where the learner is asked for input (making a decision for the hero character). The hero performs the action based on the input, and the learners get to observe the outcome of it. In the case of choosing the wrong option / decision, the situation for the hero gets worse, demonstrating the impact of incorrect knowledge. Thanks to this

interactive component, the learner receives active engagement and immediate feedback based on the decisions made.

Step 3: Recursive Deterioration Based on Wrong Choices.

The deeper the story unfolds, the stakes of making a wrong choice are higher. After initially making a wrong choice, the hero may get the opportunity to redeem himself, but too many consequent wrong choices may lead to fatal outcomes. The idea is to apply recursive learning patterns so the learners can observe and learn from their mistakes.

Step 4: External Factors Provide Temporary Relief (optional).

This step is optional because it does not test the user's knowledge but may contribute to the immersiveness of the story. In this time point of the story, the hero gets temporary relief; an external factor eases the situation (a stroke of luck, a colleague helping out, etc.), but the underlying problem is still not solved. The learner may have a false sense of security, but if a wrong decision was made, the negative consequences will be later revealed.

Step 5: Reflection and Resolution

Whether the choices were right or wrong, the hero reflects on the decisions that were made. The hero is either praised for making the right choices or criticized for making the wrong ones. Either way, an explanation in form of a final overview is provided to solidify the learners knowledge. This step should encourage the learners to revisit the timeline where the wrong decisions were made and correct their mistakes.

Of course, this is just a proposed pattern inspired by “Ed-W” [52] modeling and altered to be more suitable for recursive learning. for effective storytelling. It does not have to be the mold for every story, but it is a reliable and proven storytelling method.

Another recommendation is to carefully consider how many times to use the “Split Path” feature. Splitting a story into multiple timelines increases the number of windows exponentially. Each child storyline can become a parent to other child storylines, creating an unmanageable number of storyboard windows if not controlled.

4.2.1 Modeling a Storyboard: Practical Example

In the scope of this work, another goal is to create a storyboard with the implemented tool. A storyboard called "Code Red: A Cybersecurity Dilemma" has been created in that regard, and it is publicly available to visit [53]. The created storyboard follows the proposed pattern for storytelling. Here is a concrete example of how the story was planned before it was created with CyberSecurityBoard:

Define the learning objectives:

Objective 1: Understand the importance of identifying and avoiding phishing attempts.
Objective 2: Understand the importance of Cybersecurity department in verifying suspicious activities
Objective 3: Learn the consequences of poor cybersecurity practices.
Objective 4: Improve and develop critical thinking and decision-making skills necessary for cyber threat situations

Understand the target audience:

The audience this is aimed at are IT students, IT professionals, and generally all individuals interested in improving their practical cybersecurity knowledge. The assumption is that the audience has basic IT knowledge and understanding of IT concepts, so they can understand the content in the story. The audience members are open to alternate learning methods, particularly scenario-based learning, and enjoy interactive and engaging content.

Introduction of the main and side character

Sophia Turner (the main character of the story) is a software engineer at NextGen Inovations. She is part of an innovative and high-stakes AI project. There are three other side characters, each playing a significant role around Sophia's journey.

Sarah is Sophia's friendly coworker. She is also part of the project and may be used to provide the relief factor (step 4 in storytelling, described previously) by offering help to Sophia in a difficult situation.

Adrian is the project manager. His character will be used to explain how Sophia's decisions affected the project or the company, depending on the severity of the situation. He either praises right decisions or criticizes incorrect decisions and offers countermeasures.

Another side character is Gary, the lead of the cybersecurity team. His role is to explain the outcomes of the decisions made by Sophia and how the team has been affected by those decisions.

Create a compelling narrative:

Introduce Sophia, her role at the company, the project she is working on and other small but interesting details to make her relatable. The first phase of the story starts off slowly, introducing the remaining side characters, their roles, the atmosphere in the office, etc. The setting should simulate a typical real-life office day. Slowly introduce the first trigger during Sophia's early work hours. The first trigger comes in the form of a phishing attempt via email, Sophia is facing her first challenge, and her decision-making skills are now tested. Three decision options are presented:

1) The correct decision, which leads to the shortest possible path and a happy outcome

2) The incorrect decision, which prolongs the story but gives Sophia the oportunity to correct her mistake, the length of the path is unknown at the point as it depends on the upcoming triggers.

3) The fatal incorrect decision, this decision also leads to the shortest path, but the outcome of the story is negative.

Upcoming triggers follow a similar pattern, but the problem complexity and the seriousness of the situation increase with each following trigger. Ultimately leading the main character to either succeed or fail at the end.

The learners are encouraged to explore all possible paths, since each story line is different and contains unexplored pieces of information. This approach broadens the learners perspective on the many possibilities of “what if” scenarios. By iterating through all the story paths, the learner is given the opportunity to reflect, correct their misconceptions and deepen their knowledge.

CHAPTER 5

Prototype implementation

This chapter covers the software development aspects of the work. The technology stack and other software tools used to develop CyberSecurityBoard and argumentation why these particular technologies were used. The other part of the chapter discusses the software architecture decisions in more detail.

5.1 Technology Stack

The choice to make CyberSecurityBoard a web-based application was easy. A web-based application offers several advantages: accessibility, cross-platform compatibility, and easy integration with other services. Most modern tools that offer similar functionalities [Chapter 3] are also web-based. Since they are not that computationally demanding, there is no need to implement the tool in the most performant language/framework, and the cross-platform compatibility of web-based applications offers a great advantage.

When it comes to web development, there are numerous technology stacks and options for frontend and backend frameworks. For the frontend, most common frameworks are JavaScript frameworks like React, Angular, and Vue.js; Python frameworks such as Django and Flask; For the backend there are Java frameworks like Spring and Struts; and PHP frameworks, among others. Making a suitable choice can be quite challenging, but ultimately, the technologies listed below were chosen.

Angular 15 (for the frontend) along with **Firebase services** (handling the databases and the backend) were picked as the core technology stack for this project. Angular is a full-fledged framework, which enforces stricter clean code practices and offers a very rich toolset of external official as well as custom libraries. [54, 55]. This offers a great advantage for agile development while following good programming practices without having to put a lot of conscious effort. Firebase offers a lot of low-code backend services,

which the developer can take advantage of to shorten the development process. Since both technologies are created by Google, they integrate very well with each other. So the main reasoning for this choice was balancing between agile development and enforcing good programming practices.

Some key features and advantages of using Angular:

- **Support for Single Page Applications (SPAs)**. SPAs eliminate unnecessary page reloads, and only reload the components the changes were made to. Since the project is massively multimedia oriented, this feature is a must.
- **Angular is a full-fledged framework**. This simply means that a complete solution for building web applications is provided out of the box. Relying on external libraries (which often offer buggy solutions) is not strongly emphasized. Other built in tools that come with this framework are a powerful router, form handling, HTTP client, comprehensive testing utilities, etc. This integrated approach streamlines the development process and ensures consistency across the application.
- **Modular Architecture**. Breaking down the project into smaller modules enhances its scalability, maintainability, and reusability. Angular, compared to most frontend frameworks and libraries, takes a stricter stance on coding practices and file organization. While some may view this lack of freedom as a drawback, in this particular project it proved to be a valuable asset. Utilizing TypeScript's static typing as well as an opinionated structured approach with TypeScript's static typing allowed for a well-organized codebase and a clear separation of concerns. The project was structured into distinct modules, each addressing a specific domain, resulting in a more manageable and coherent development process. Furthermore, Angular's strict conventions and recommended practices make it harder to implement poor coding practices and disorganized file structures. This approach not only improves code quality but also makes it easier for Angular newcomers to be more productive and less concerned with figuring out the best practices.
- **Strong community and ecosystem.** Angular being the second most popular framework, only second to React [60], as well as a product created and maintained by Google means that developers can enjoy very strong community-driven assistance. Again, this is especially beneficial for newcomers to the framework.

Key features and benefits of using Firebase:

User Authentication. User access rights, login, registration and other common functionalities are all included in this service. Configuration requires very little code and integration with an angular application is very straightforward. The service offers various registration methods, but for simplicity's sake, registrations via emails were used.

Cloud Hosting. For making the software tool available in the cloud the Cloud Hosting service was used. This service also works great with Angular since it is optimized for Single Page Applications. In order to deploy the project to the cloud, it firstly needs to be built into a single file and then it can be uploaded. After the upload process a URL is generated to access the project in the cloud. By default, every Firebase project has free subdomains on the web.app and firebaseapp.com domains. Firebase automatically provisions SSL certificates for all domains so that the content is served securely.

File Storage. A file-storage system was necessary to store the images for the characters, symbols, and speech bubbles (shown in Chapter 4). Cloud storage for Firebase is another service that was used to solve this problem. All of the images used in the project are stored on the cloud server and later retrieved by reference.

Realtime Database. Cloud Firestore is a NoSQL cloud-based database. The data is stored as JSON and is synchronized in realtime. This ensures that users across different clients always receive the latest information. It has even offline capabilities in case the application loses connection.

The combination of Angular and Firebase services made the development process agile by utilizing the free services. The overhead of implementing common server setups and functions was avoided.

Before starting the development process, conceptualization and initial design was created with Figma [56]. Figma is a practical web-based tool used for creating user interfaces. All of the tools and services for this project, including Figma, were free of charge (free tier versions).

5.2 Integration of Firebase backend services with Angular Framework

Angular's dependency injection system is used to integrate Firebase into an Angular application. But before the injected service is recognized by the framework, it needs to be installed, configured, and initialized.

The command “ng add @angular/fire” installs all the necessary firebase packages to the project. In order for the angular application to connect and identify with the installed firebase services, entering Firebase project credentials in the “environment.ts” file is needed. The credentials (parameters like API key and project ID, etc.) are automatically generated and provided by Firebase when setting up the project. The configuration looks like this:

```
export const environment = {
  firebase: {
    projectId: 'cybersecurityboard-a5dc4',
    appId: '1:867938367080:web:1a8d144f0315c05385addd',
    storageBucket: 'cybersecurityboard-a5dc4.appspot.com',
    apiKey: 'apikey123',
    authDomain: 'cybersecurityboard-a5dc4.firebaseapp.com',
    messagingSenderId: '867938367080',
    measurementId: 'G-X9GX1TZKLE',
  },
};
```

This identifies the backend services and allows Angular to establish a connection with them.

Now that a connection has been established, it is possible to initialize Firebase services. The services are initialized in the app.module.ts file of the Angular project. This step ensures that the necessary services (in this case: authentication, database, storage, etc.) are available for dependency injection. Finally, the individual Angular components may now use the necessary services via dependency injection. For improved modularity and reducing repetitiveness, Angular recommends the use of service classes. For this project, service classes are created to encapsulate different Firebase functionalities. These service classes are then injected into Angular components by demand. The communication between

the cloud services of Firebase and the Angular client is performed with HTTP requests. The figure below explains the communication concept in more detail:

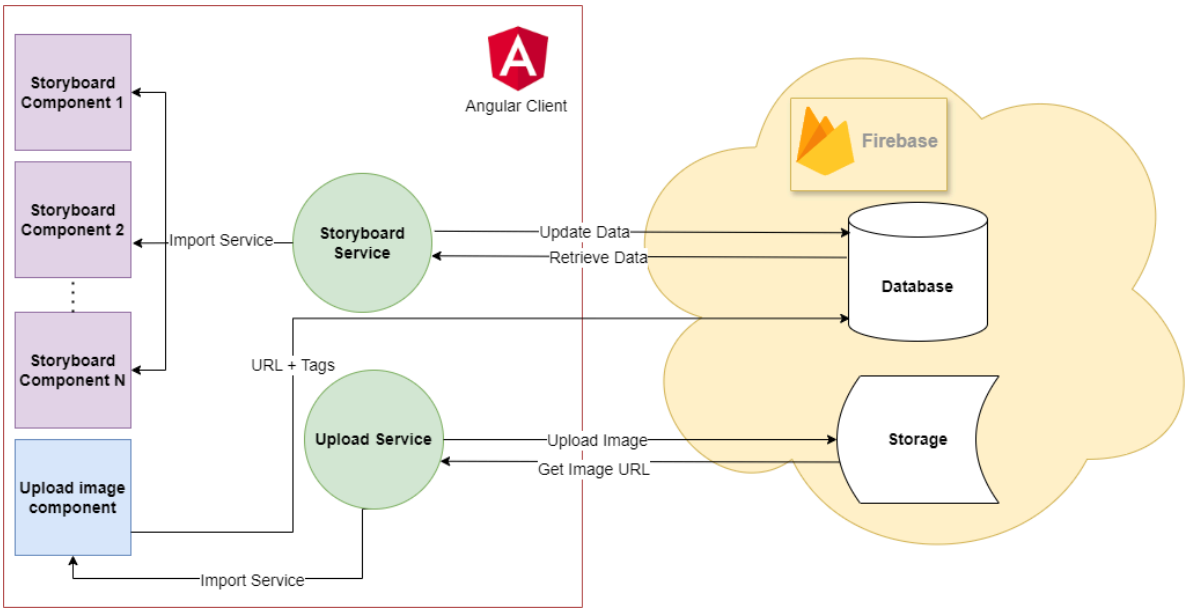

Figure 5.1: Communication between the Angular client and Firebase Services (Firebase Storage and Cloud Firestore)

A high-level representation of one of the CyberSecurityBoard's core functions is shown in Figure 5.1: uploading images to the cloud storage and retrieving them by reference.

The upload service is imported (injected) in the Upload Image component. Using a front-end form that is part of the Upload image component. Using the form, the user can upload an image from local storage and attach several tags to the image, which are later used to identify and categorize the image. After uploading the image, the upload service stores the uploaded image into Firebase cloud storage. The cloud storage provides URL's for each image that it stores. Using the same upload service, the URL of the uploaded image gets retrieved as a return parameter. The upload image component receives the URL from the service class and additionally attaches image tags that were set through user input (upload form). The tags along with the image URL are then inserted into the cloud database. As an end result, a successful upload stores an image in the storage and references that image URL with additional tags in the cloud database.

The storyboard service is the largest service class as it contains all service methods necessary for manipulating the image objects within a storyboard. Since the cloud database is already filled with image urls (which reference to the storage) and tags thanks to the upload service, there is no need for the storyboard service to access the storage. The storyboard service handles operations such as inserting an image inside the storyboard window, resizing an image or changing its position, or inserting text. All of these operations trigger either an update and/or a retrieve from the database. The next sub-chapter thoroughly explains the structure of the cloud database.

5.3 Database structure

JSON plays a crucial role in how data is structured in Firebase's database cloud solution, Cloud Firestore [55]. It is a scalable NoSQL cloud database, this implies that data is stored in documents, which are essentially JSON-like objects. In the case of Cloud Firestore, a document is part of a collection that holds an array of documents, each identified by a unique ID. Each document consists of key-value pairs; the keys are field names, and the values are, as the name implies, the values that the fields contain. The values can be either strings, numbers, booleans, arrays, or nested objects.

In the case of the CyberSecurityBoard project, the database consists of two collections: imageData and storyboards. The “imageData” collection holds documents that are created using the “Upload service” class, it stores the URL links of images along with optional tags describing the image. The JSON-like structure looks like this:

```
{
  "categoryTag": "",
  "emotionTag": "happy",
  "imageLink": [
"https://firebasestorage.googleapis.com/v0/b/cybersecurityboard-a5dc4.appspot.com/o/1708266490499-5.svg?alt=media&token=7d1d3ec0-ffc7-48ed-9d67-9cc195d8a359"
  ],
  "imageName": "Janet_standing",
  "imageTag": "character",
  "positionTag": "standing"
}
```

The code snippet above is a JSON interpretation of how the data is organized in the

document. Besides the URL of the image, the following properties are attached to the image:

- categoryTag: an optional parameter that describes which category an image belongs to. An example of categoryTag can be the role of a character, like a hacker.
- emotionTag: used to associate an emotion to the character, useful for filtering characters by emotion
- imageName: name that describes the image, useful for manual debugging and searching the database
- imageTag: can be either “character”, “speech_bubble” or “symbol”. It is used to group the images into the three main categories for the topbar navigation (Chapter 4, Figure 4.5).
- positionTag: describes the physical position of a character, used for filtering and categorizing images.

The other collection, named "storyboards" has a more complex structure. This collection contains documents responsible for the representation of a storyboard in the UI. It contains data about storyboard windows, the objects within the storyboard windows (images, texts), as well as property data of the objects (position in the window, size, etc.). It also contains general storyboard data, such as the name of the storyboard, the username of the creator, id, etc.

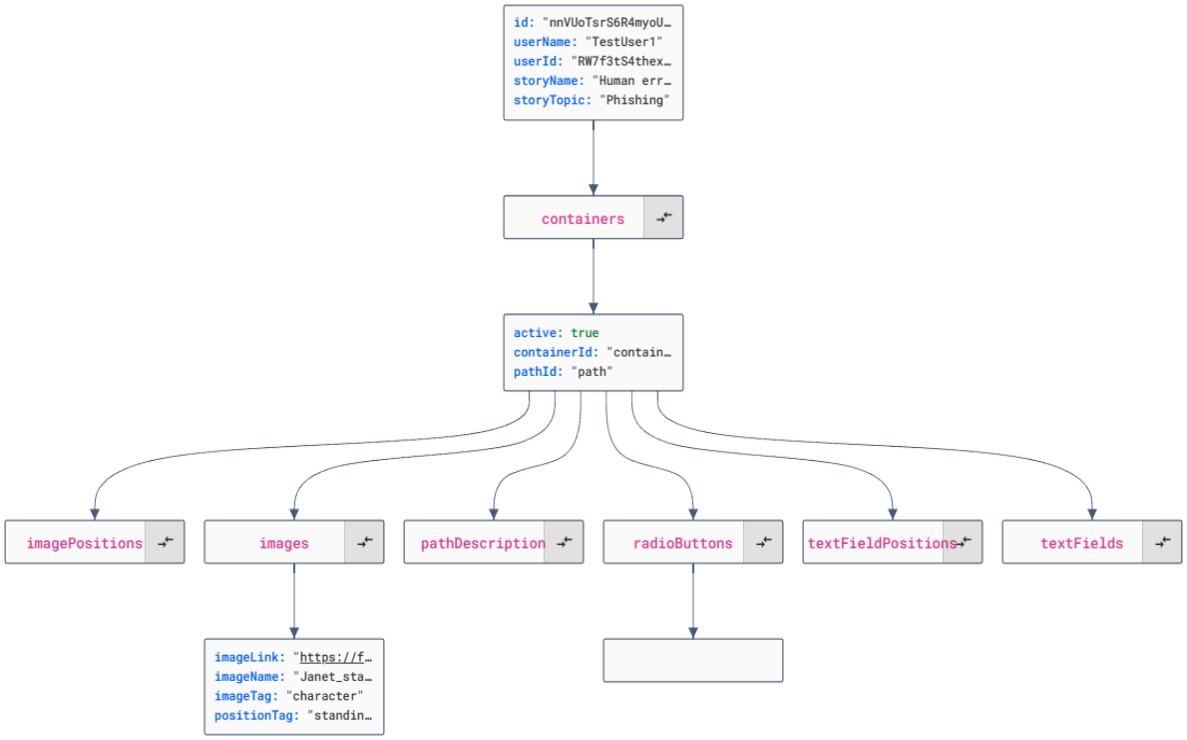

Figure 5.2: Visual representation of "storyboards" document within the NoSQL cloud database

The first five fields are simple strings that describe the storyboard. The fields in question are: id, userName, userId, storyName, storyTopic. These are used to identify and generally describe a storyboard. The "containers" field, however, is a nested array that holds the following non-array fields:

- Active: a boolean that checks if the current slide is open or not
- containerId: a string that represents the id of the storyboard window
- pathId: a string, dynamically generated, used for tracking the navigation path.

As well as the following array fields:

- imagePositions: x and y positions describing the physical position of an image inside the storyboard window
- Images: array of image URL's located in storyboard window
- textFieldPositions: similar to imagePositions, but used for texts
- textFields: similar to images, instead of holding URL strings, it contains the dialogue texts or headers of a storyboard.
- radioButtons: dynamically generated as a consequence of splitting the path into multiple timelines. It is an array of boolean values used to navigate to the corresponding path.

5.4 Controlling the backend services with service classes

The encapsulated methods located in Angular service classes, particularly the service class named “StoryboardService” are responsible for all CRUD operations in the Firebase cloud database.

Let's examine the method for saving storyboards inside the database:

```
  async saveStoryboards(storyboards: Storyboard[]) {
    const storyboardsForFirestore = storyboards.map((storyboard) => {
      const containersForFirestore = storyboard.containers.map((container) => {
        const imagesArray = Object.entries(container.images).map(([key, value])
=> ({ key, value }));
        const imagePositionsArray =
Object.entries(container.imagePositions).map(([key, value]) => ({ key, value
}));
        const textFieldsArray = Object.entries(container.textFields).map(([key,
value]) => ({ key, value }));
        const textFieldPositionsArray =
Object.entries(container.textFieldPositions).map(([key, value]) => ({ key,
value }));
        // const radioButtonArray =
Object.entries(container.radioButtons).map(([key,value]) => ({key, value}));
        const pathDescriptionArray =
Object.entries(container.pathDescription).map(([key, value]) => ({ key, value
}));

        return {
          ...container,
          images: imagesArray,
          imagePositions: imagePositionsArray,
          textFields: textFieldsArray,
          textFieldPositions: textFieldPositionsArray,
          // radioButton: radioButtonArray
          pathDescription: pathDescriptionArray

        };
      });
      return {
        ...storyboard,
        containers: containersForFirestore
      };
    });

    const storyboardsRef = collection(this.firestore, 'storyboards');
    const newStoryboards = [];
    for (const storyboard of storyboardsForFirestore) {
```

```
      if (storyboard.id) {
        const docRef = doc(storyboardsRef, storyboard.id);
        await updateDoc(docRef, storyboard);
        newStoryboards.push(storyboard);
      } else {
        const docRef = await addDoc(storyboardsRef, storyboard);
        newStoryboards.push({ ...storyboard, id: docRef.id });
      }
    }
    return newStoryboards;
  }
```

The saveStoryboards manages storyboard data within the Firestore database. It accepts an array of storyboard objects, each containing various properties. The method performs the following key actions:

Transformation: Making sure that each storyboard is transformed in a compatible format before passing it to the firebase service.

Update or Create: For each storyboard, the method checks if it has an existing ID:

- If it does, it updates the corresponding Firestore document.
- If it doesn't, it creates a new document in Firestore.

Return Values: After processing, the method returns an array of storyboard objects that now include their respective Firestore document IDs, making sure that users have access to the latest data structure.

This approach allows for efficient data management and ensures that the storyboard information is accurately reflected in the Firestore database.

Retrieving a single storyboard is possible with the getStoryBoardId method:

```
  async getStoryboard(storyboardId: string): Promise<Storyboard | null> {
    const storyboardRef = doc(this.firestore, 'storyboards', storyboardId);
    const storyboardDoc = await getDoc(storyboardRef);

    if (storyboardDoc.exists()) {
      const storyboardData = storyboardDoc.data();
      const containersData = storyboardData['containers'];

      const containers: Container[] = containersData.map((containerData:
Container) => {
        const imagesArray = Object.entries(containerData.images).map(([key,
value]) => ({ key, value }));
        const imagePositionsArray =
```

```
Object.entries(containerData.imagePositions).map(([key, value]) => ({ key,
value }));

        // Get text values from database
        const textFieldsArray =
Object.entries(containerData.textFields).map(([key, value]) => ({ key, value
}));
        const textFieldPositionsArray =
Object.entries(containerData.textFieldPositions).map(([key, value]) => ({ key,
value }));

        const pathDescriptionArray =
Object.entries(containerData.pathDescription).map(([key, value]) => ({ key,
value }));

        const images = imagesArray.reduce((acc: { [key: string]: string }, obj:
any) => {
          acc[obj.key] = obj.value.value;
          return acc;
        }, {});

        const imagePositions = imagePositionsArray.reduce(
          (acc: { [key: string]: { x: number; y: number, width: number, height:
number } }, obj: any) => {
    acc[obj.key] = obj.value.value;
            return acc;
          },
          {}
        );

        const textFields = textFieldsArray.reduce((acc: { [key: string]: string
}, obj: any) => {
          acc[obj.key] = obj.value.value;
          return acc;
        }, {});

        const textFieldPositions = textFieldPositionsArray.reduce(
          (acc: { [key: string]: { x: number; y: number, width: number, height:
number } }, obj: any) => {
            acc[obj.key] = obj.value.value;
            return acc;
          },
          {}
        );

        const pathDescription = pathDescriptionArray.reduce((acc: { [key:
string]: string }, obj: any) => {
          acc[obj.key] = obj.value.value;
          return acc;
```

```
        }, {});
        const radioButtons = containerData.radioButtons || {}; // Check if
radioButtons property exists, otherwise fallback to an empty object

        return {
          id: containerData.id,
          active: containerData.active,
          pathId: containerData.pathId,
          images,
          imagePositions,
          textFields,
          textFieldPositions,
          radioButtons,
          pathDescription

        };
      });

      return {
        storyName: storyboardData['storyName'],
        containers: containers,
      };
    } else {
      return null;
    }
  }
```

The getStoryboard method as the name implies, is used for retrieving storyboard data from the Firestore database. The method accepts the storyboardId as an argument, and retrieves the storyboard with the particular unique Id.

- A reference in Firestore for the storyboard document is created
- The next step is to check whether the document exists, and fetch it
- If the document exists, it extracts the storyboard data and processes the associated containers, including images, text fields, and their positions.
- It then constructs an object with relevant storyboard data (storyboard name, and container data attached to the storyboard)
- It returns null, in case the document is not found

Deleting a container or a complete storyboard with deleteStoryboard and deleteContainer methods:

```
  deleteStoryboard(storyboardId: string): Promise<void> {
    return new Promise((resolve, reject) => {
      const storyboardRef = doc(this.firestore, 'storyboards', storyboardId);
      deleteDoc(storyboardRef)
        .then(() => {
          console.log('Storyboard deleted');
          const currentStoryboards = this.storyboards.getValue();
          const updatedStoryboards = currentStoryboards.filter(storyboard =>
storyboard.id !== storyboardId);
          this.storyboards.next(updatedStoryboards);
          resolve();
        })
        .catch((error) => {
          console.error('Error deleting storyboard', error);
          reject(error);
        });
    });
  }

  deleteContainer(storyboardId: string, containerId: string, pathId: string):
Promise<void> {
    return new Promise(async (resolve, reject) => {
      const storyboardRef = doc(this.firestore, 'storyboards', storyboardId);
      try {
        // Retrieve the current storyboard document
        const docSnap = await getDoc(storyboardRef);
        if (docSnap.exists()) {
          const storyboardData = docSnap.data();
          const updatedContainers =
storyboardData['containers'].filter((container: any) =>
            container.id !== containerId || container.pathId !== pathId);

          // Update the storyboard document with the filtered containers array
          await updateDoc(storyboardRef, { containers: updatedContainers });
          console.log('Container deleted');
          resolve();
        } else {
          console.log("No such storyboard!");
          reject("No such storyboard!");
        }
      } catch (error) {
        console.error('Error deleting container', error);
        reject(error);
      }
    });
  }
```

The deleteStoryboard and deleteContainer methods are asynchronous functions designed to manage the deletion of storyboards and their components in the database.
Breakdown of deleteStoryboard:

- Initiating a new promise for handling asynchronous operations.
- Storyboard Reference: similarly to the previous methods, a reference is created in Firestore
- Delete Document: the deleteDoc removes the data from Firestore and leaves a confirmation message
- If deletion was successful, it logs a confirmation message. The list of storyboards is then refreshed, and the Id of the deleted storyboard is removed from the list.
- Updates the subject with the new list of storyboards.
- Resolves the promise.
- In case an error occurs during deletion, it logs the error and rejects the promise.

Breakdown of deleteContainer:

- Initiating a new promise for handling asynchronous operations.
- Storyboard Reference: creates a reference in Firestore
- Retrieve Document: Uses getDoc to fetch the storyboard document data.
- If the document exists, it extracts the current containers.
- Filters the containers array to remove the specified container using both containerId and pathId.
- Update Document: Updates the storyboard document with the filtered containers array using updateDoc.
- In case of successful handling it logs a confirmation message and resolves the promise.
- In case of an error it logs the error and rejects the promise.

In summary, these methods facilitate the deletion of storyboards and their containers from Firestore while managing the application's state accordingly.

5.4.1 Utilizing the backend services in the components

The methods discussed in the previous chapter were the core backend logic, implemented in the service class. Now we discuss the methods that call the service class within a component class.

The addContainer method:

```
 async addContainer() {
    this.containers.forEach((container) => (container.active = false));
    const newContainer: Container = {
      id: `container${this.containers.length}`,
      active: true,
      images: {},
      imagePositions: {},
      textFields: {},
      textFieldPositions: {},
      pathId: 'path',
      pathDescription: {}
    };
    this.containers.push(newContainer);
    const storyboard = await
this.storyBoardService.getStoryboard(this.storyId!);
    storyboard!.id = this.storyId!;
    storyboard!.containers = this.containers;
    const storyboardId = this.route.snapshot.paramMap.get('id');
    this.handleStoryboardOperations(storyboardId!, this.containers);
  }
```

This method creates a new container in the database. It injects the storyboard service to use the getStoryboard method, which retrieves the current storyboard we are working on. Specifying the id of the storyboard is crucial, otherwise instead of only updating a storyboard with a new container, a whole new storyboard will be created. The new container is set to active on creation, which means the user gets navigated to the newly created container.

The splitStoryPath method is very often mentioned in this work. It is responsible for enabling the feature of multiple storylines, which was discussed in the fourth chapter, explaining how it looks in the frontend.

```
  async splitStoryPath(n: number) {
    const activeContainer = this.containers.find(container =>
container.active);
    if (!activeContainer) return;
    const activeIndex = this.containers.indexOf(activeContainer);
    const pathIds = Array.from({ length: n }, (_, i) => activeContainer.pathId
```

```
+ String.fromCharCode(97 + i));
    const newContainers: Container[] = pathIds.map(pathId => ({
      id: `container${this.containers.length}`,
      active: false,
      images: {},
      imagePositions: {},
      textFields: {}, // Initialize textFields as an empty object for each new
container
      textFieldPositions: {}, // Initialize textFieldPositions as an empty
object for each new container
      radioButtons: {}, // Initialize radioButtons as an empty object for each
new container
      pathDescription: Array(n).fill({ text: '' }),
      pathId,
    }));

    // Generate n text fields and radio buttons for the active container
    for (let i = 0; i < n; i++) {
      const newPathId = String.fromCharCode(97 + i);

      // Ensure that activeContainer.pathDescription[i] is defined
      activeContainer.pathDescription[i] = activeContainer.pathDescription[i]
|| { text: '' };

      // Create a new object for radioButtons with the new key-value pair for
the active container
      activeContainer.radioButtons = {
        ...activeContainer.radioButtons,
        [newPathId]: false
      };
    }
    // Insert new containers at the correct index
    this.containers.splice(activeIndex + 1, 0, ...newContainers);
    const storyboard = await
this.storyBoardService.getStoryboard(this.storyId!);
    storyboard!.id = this.storyId!;
    storyboard!.containers = this.containers;

    const storyboardId = this.route.snapshot.paramMap.get('id');
    this.handleStoryboardOperations(storyboardId!, this.containers);
  }
```

The code snippet above does the following:

1. Find Active Container: First step is to locate the currently active container, if there is no active container set, the function exits early.

2. Once the active container has been located, path ids are generated for the child containers (for the purpose of unique identification). The path is an extension of the parent container, with the addition of a unique alphabetic letter at the very end.
3. Now that new unique ids for the child containers exist, a creation of a container can be triggered and the ids are passed to it. A container is created with its properties set to initial values.
4. Radio buttons and textfields positions are added to the active container (which is the parent) which are used for handling user input for a decision and describing the decision.
5. The new containers are inserted into the container array.
6. The storyboard is updated with the newly added child containers and elements in the parent container.
7. The method called at the end: handleStoryboardOperations is used to reflect the changes in the UI.

The split-path algorithm

The split path algorithm was developed with efficient unique id handling in mind. Since a storyboard container already contains a unique id, by just adding a unique alphabetic letter to that path a combination of up to 26 new unique ids can be obtained. This approach also offers an easy way of tracking a parent-to-child container relationship. Since there are 26 letters in the alphabet, and the split path has been artificially limited to only 5 children per parent container, only the first five letters of the alphabet are utilized for generating a new id. So a child can be very easily identified because it contains the id of the parent and only an additional letter at the very end of the id.

The algorithm is explained in the graph below:

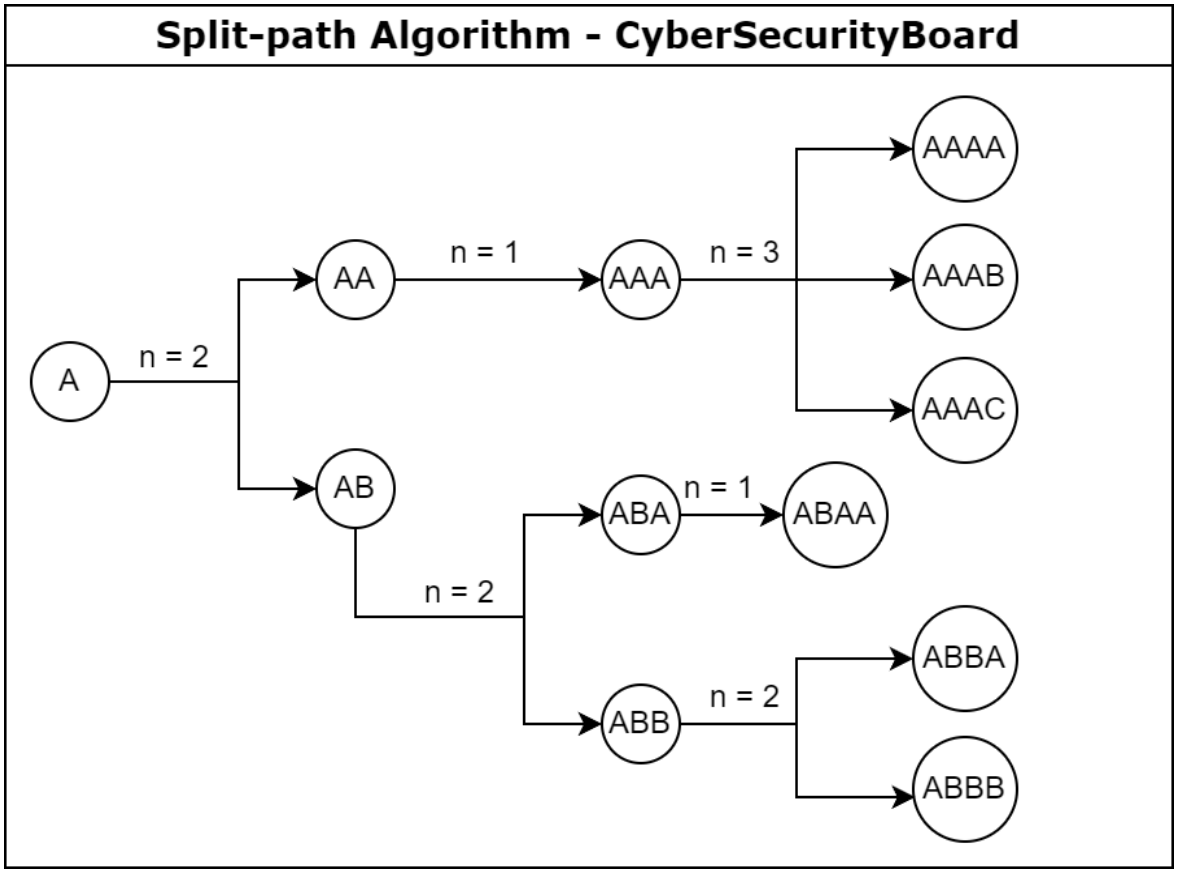

Figure 5.3: Visualisation of split-path algorithm

The image above portrays the algorithm as a Tree graph with ever-expanding nodes. The n is the user input; in the case of CybersecurityBoard it is limited to 5, in order to avoid massive computing tasks or accidental high user input. The n is the number of child nodes belonging to a parent node. The names of the nodes are assigned alphabetically and in the following order:

The initial node gets assigned the letter A, the second generation has two letters, the third three, etc. Meaning that the more letters a node contains, the longer the path to the initial parent. The first node in any new path receives the letter A, and since the limit to splitting paths is set to 5, that means that the following letters are being used: A,B,C,D,E.
A being the first node and E being the last node of a “generation” of containers.
The algorithm comes with following advantages:

1. **Distinct ID for every node**

Since every node gets assigned a different letter depending on its position and generation, that means it automatically receives a unique identifier as well. Since there is a generated unique identifier for every path; no additional ID's are necessary for the nodes.

2. **Easy navigation**

By looking at a node, we can instantly tell who are the parents of a child node, as well as the child of the child nodes, etc. Navigating backwards is also very intuitive, by simply subtracting the letters. An Example: Path named: ABBA. We can conclude that this path comes from the following ancestors: ABB, AB, and finally A.

The method onDrop enables the functionality of dragging an image from the Top Graphics bar and dropping it into the container window. Once an image has been dropped into the container, an entry inside the database is made with the URL link of the dropped images as well as all the relevant data an image needs to be displayed in the window (position on the x and y axes, width, and height).

```
  async onDrop(event: DragEvent) {
    event.preventDefault();
    const url = this.draggedImage;
    if (!url) return;
    const activeContainer = this.containers.find((container) =>
container.active);
    if (!activeContainer) return;
    const containerElement = document.getElementById(activeContainer.id);
    if (!containerElement) return;

    const x = event.offsetX;
    const y = event.offsetY;

    const newIndex = Object.keys(activeContainer.images).length.toString();
    activeContainer.images[newIndex] = url;
    // Set the dimensions to the stored values
    activeContainer.imagePositions[newIndex] = {
      x, y,
      width: 100,
      height: 100,
    };

    if (
      Object.keys(activeContainer.images).some((key) =>
activeContainer.images[key] === undefined) ||
      Object.keys(activeContainer.imagePositions).some((key) =>
activeContainer.imagePositions[key] === undefined)
    ) {
      console.error('Error: Undefined values found in container data');
```

```
    } else {
      // Get the storyboard id
      const storyboardId = this.route.snapshot.paramMap.get('id');

      this.handleDropDownOperations(storyboardId!, this.containers, newIndex);
    }
    this.cd.detectChanges();
    this.updateElementPositions();

  }
```

And finally, the saveAndPublishStory method saves all the local changes to the database. This method is triggered on many occasions, after dropping an image, updating the position or size, and publishing the story. The reason for saving so often is to preserve as much work as possible in case the application crashes or the connection breaks.

```
  saveAndPublishStory() {

    const storyboardId = this.route.snapshot.paramMap.get('id');
    this.handleStoryboardOperations(storyboardId!, this.containers);

    const modifiedUrl = this.router.url.replace('/edit/', '/view/');
    this.router.navigateByUrl(modifiedUrl);
  }

  async handleDropDownOperations(storyboardId: string, containers: Container[],
droppedElementIndex: string) {
    if (storyboardId) {
      const storyboard = await
this.storyBoardService.getStoryboard(storyboardId);
      storyboard!.id = storyboardId;
      storyboard!.containers = containers;
      await this.storyBoardService.saveStoryboards([storyboard!]);
      this.updateSingleElementPosition(droppedElementIndex);
    }
  }

  async handleStoryboardOperations(storyboardId: string, containers:
Container[]) {
    if (storyboardId) {
      const storyboard = await
this.storyBoardService.getStoryboard(storyboardId);
      storyboard!.id = storyboardId;
      storyboard!.containers = containers;
      await this.storyBoardService.saveStoryboards([storyboard!]);
      this.updateElementPositions();
    }
  }
```

5.4.2 Project structure and how to run it

The project has a typical Angular architecture, with components consisting of four files: css, html, typescript, test files (spec.ts extension). To run the project locally, Angular version 15 or above needs to be installed on the machine, as well as the necessary npm libraries need to be downloaded and installed via the “npm install” command.

The project can be downloaded from the open github repository: https://github.com/MehrudinSabani/CyberSecurityBoard

Running the project on the local machine is performed with the command “ng serve” from the project directory. The unit tests have been developed with the default tools provided by the framework: Jasmine and Karma. Small or insignificant components do not have implemented unit tests. The unit tests can be performed with the command “ng test”. For testing components, mock services that resemble Firebase services have been developed to mimic their functionality in order to avoid accessing the real database while testing.

CHAPTER 6

Conclusions, lessons learned

Exploring the state-of-the-art tools, methods, and overall approaches to teaching cybersecurity have shown that scenario-based interactive storyboards have immense potential. Integrating storyboards into cybersecurity education is an area where little work has been done, but it is worth further researching in this direction because of the promising results it has shown. With the developed software alongside this work, the process of creating interactive storyboards has been made agile and straightforward.

6.1 Key Findings

1. **Cybersecurity is an ever-changing topic.**

People involved in this field need to improve their knowledge, as new threats, emerging technologies, and evolving attack strategies require them. Teaching methods need to keep up with these demands and constantly look for improvements and efficiency in how a topic can be taught.

2. **Interactivity, simulation, and gamification are the key characteristics of future training methods for cybersecurity training.**

Throughout this work, constant emphasis on the importance of engaging content was put. Characteristics such as interactivity, simulation of the real world, gamification, and practical hands-on exercises are what increase engagement and implicitly improve the learners experience. These findings hint at which direction cybersecurity training methods should move in.

3. **Applicability, low cost, and flexibility are the major strength points of CyberSecurityBoard.**

As a minimalistic web-based application, CyberSecurityBoard is easily accessible to a wide range of users. Regarding content, topics of most cybersecurity branches can be taught using this tool.

Compared to other high-engagement, simulation-based training methods like VR or cyber ranges, CyberSecurityBoard is a much more cost-effective solution that still manages to deliver high-quality educational experiences. Of course, the quality of immersiveness and engagement is not on par with these methods, but the setup time and extremely low costs to create a storyboard may be worth it in some use cases, especially in cases where lighter, less complex topics can be explained just as effectively.

6.2 Lessons learned

1. **Importance of Integrating Interactive Elements:**

Every teaching method that was used for the comparison framework had an interactive component as part of the delivery strategy. Inclusion of the “Split Path” feature where the learners can observe the consequences of their decisions and recursively navigate to the right path was a crucial part of the delivery strategy for CyberSecurityBoard. This little feature significantly enhanced the learning experience.

Firebase services facilitated efficient development and easy maintenance.

2. **Need for Comprehensive Content:**

Having a tool that enables you to create a storyboard does not solve the problem of quality content delivery. Without proper structuring and designing the learning material, even the highest-end technologies will not deliver the desired results. In the case of storyboarding, developing good storywriting strategies and structuring the lecture is crucial. The content has to be relevant, immersive, and up to date.

3. **User Interface and Experience:**

Designing the front-end of such tools plays a crucial role in end user satisfaction and optimal use. The development approach of this tool is to utilize minimalistic design and display advanced features only when necessary. It also separates the usage into two user groups “Learners” and “Creators” for which separate UI/UX has been developed.

4. **Scalability and Maintenance:**

The project has been developed with scalability in mind, so that further development and implementation of additional features can be built on top of the current solution without the need to rewrite the core logic.

6.3 Future work, improvements, and suggestions

1. **Create more storyboards covering a broader range of topics and knowledge levels:**

As of now, there are only a few storyboards created with the recommended story writing practices. The content in the storyboards is mostly beginner friendly and covers only a few topics. It is certainly interesting to further explore the limits of CyberSecurityBoard by creating more complex stories aimed towards a more experienced audience and covering a broader range of topics.

2. **Gamify the tool:**

Integrating game elements like scoreboards, quizzes, and flags will make the learning experience even more interesting.

3. **Tracking tool for self-assessment**:

Implement a progress tracking tool that the learners can use for self-assessment. This will allow the learners to easily keep track of their progress and better pinpoint their strengths and weaknesses.

4. **Integration with AI:**

In recent years, AI has been one of the most talked about topics in the IT world. Continuous efforts are being made to integrate AI into every field. Investigating the potential of AI integration with CyberSecurityBoard is also an interesting direction. Many processes can be automated or made easier, such as story writing, image generation for the characters, etc.

6.4 Concluding Remarks

This thesis contributes to the field of cybersecurity education by introducing and exploring the idea of using storyboards as a method for teaching cybersecurity topics. The use of storyboards as a teaching method shows promising potential, and the research presented in this work shows that increasing learners' engagement leads to increased knowledge retention.

Following that trend, as the practical part of this work, a custom software tool that can create engaging storyboards has been developed. In order to make the storyboards more engaging, interactive components, simulations based on real-life scenarios, and good practices for effective storytelling have been researched and implemented in a practical example. The conducted survey where the participants got the chance to experience an interactive storyboard confirmed that this teaching method shows promising potential.

Overall, this work offers a foundation for future research and development in using storyboards for cybersecurity education and suggests a promising direction for innovative teaching methods.

Bibliography

[1] ENISA-https://www.enisa.europa.eu/news/enisa-news/higher-education-in-europe-understanding-the-cybersecurity-skills-gap-in-the-eu/. Accessed 11.25.2022

[2] Nurse, J. R., Adamos, K., Grammatopoulos, A., & Di Franco, F. Addressing the EU cybersecurity skills shortage and gap through higher education. European Union Agency for Cybersecurity (ENISA) Report (2021).

[3] Ifigeneia, Lella, T. Marianthi, and M. Apostolos. "Enisa threat landscape 2021." ENISA, Oct (2022).

[4] Crumpler, William, and James A. Lewis. *The cybersecurity workforce gap*. Washington, DC, USA: Center for Strategic and International Studies (CSIS), 2019.

[5] Hursen, Cigdem, and Funda Gezer Fasli. "Investigating the Efficiency of Scenario Based Learning and Reflective Learning Approaches in Teacher Education." *European Journal of Contemporary Education* 6.2: 264-279. (2017)

[6] Zamir, W. Comparative analysis of Activity Based Learning System Vs Lecture Method on Students' performance (Doctoral dissertation, Thesis for: B. Ed. Secondary, Virtual University of Pakistan) (2020).

[7] BUMC-https://www.bumc.bu.edu/jmedday/archives/2016-workshops/activity-based-learning-disruptive-innovation-in-education-across-three-schools/#:~:text=Activity%2Dbased%20learning%20is%20a,interact%20with%20ideas%20and%20information/. Accessed 30.11.2022

[8] Cyber2yr2020-https://www.nationalcyberwatch.org/2020/12/acm-cyber2yr2020-curriculum-guidelines/. Accessed 30.11.2022

[9] CSIS-https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents/. Accessed: 30.11.2022

[10] Duch, Barbara J., Susan E. Groh, and Deborah E. Allen. The power of problem-based learning: a practical" how to" for teaching undergraduate courses in any discipline. Stylus Publishing, LLC. (2001).

[11] Study-https://study.com/learn/lesson/curriculum-models-types-components.html/. Accessed: 30.11.2022

[12] N. Chowdhury, N. Espen, R. Kine and V. Gkioulos. International Journal of Safety and Security Engineering, (2021).

[13] AppHaus. https://apphaus.sap.com/resource/scenes/. Accessed 12.12.2022

[14] AustCyber-https://www.austcyber.com/resources/dashboards/NICE-workforce-framework/. Accessed 12.01.2022

[15] Beuran, Razvan, et al. "Integrated framework for hands-on cybersecurity training: CyTrONE." Computers & Security 78: 43-59 (2018).

[16] Nist Framework - https://www.nist.gov/cyberframework/. Accessed 10.01.2023

[17] Beveridge, Robert. "Effectiveness of increasing realism into cybersecurity training." International Journal of Cyber Research and Education (IJCRE) 2.1 (2020): 40-54.

[18] Yamin, M.M., Katt, B., & Gkioulos, V. Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. Comput. Secur., 88, (2020).

[19] Podnar, T., Dobson, G., Updyke, D., & Reed, W. Foundation of Cyber Ranges, (2021)

[20] Linardos, V. Development of a cyber range platform (Doctoral dissertation, University of Piraeus (Greece)), (2021)

[21] Leitner, M., Frank, M., Hotwagner, W., Langner, G., Maurhart, O., Pahi, T., Reuter, L., Skopik, F., Smith, P., & Warum, M. AIT Cyber Range: Flexible Cyber Security Environment for Exercises, Training and Research. Proceedings of the 2020 European Interdisciplinary Cybersecurity Conference, (2020).

[22] Nakata, Ryotaro, and Akira Otsuka. "Cyexec*: A high-performance container-based cyber range with scenario randomization." IEEE Access 9 : 109095-109114, (2021).

[23] Nakata, R., & Otsuka, A. CyExec*: Automatic Generation of Randomized Cyber Range Scenarios, 226-236, (2021).

[24] Lates, Ionut. "Cyber Ranges Implementation Methodology." Proceedings of the International Conference on Business Excellence. Vol. 16. No. 1. 2022.

[25] Ferguson, Bernard, Anne Tall, and Denise Olsen. "National cyber range overview." 2014 IEEE Military communications conference. IEEE, 2014.

[26]

[27] Oh, S. K., Stickney, N., Hawthorne, D., & Matthews, S. J.. Teaching web-attacks on a raspberry pi cyber range. In Proceedings of the 21st Annual Conference on Information Technology Education (pp. 324-329), (2020).

Correct sources:

[28] Ros, S., Gonzalez, S., Robles-Gómez, A., Tobarra, L., Caminero, A., & Cano, J. Analyzing Students' Self-Perception of Success and Learning Effectiveness Using Gamification in an Online Cybersecurity Course. IEEE Access, 8, 97718-97728, (2020).

[29] Tobarra, L., Trapero, A., Pastor, R., Robles-Gómez, A., Hernández, R., Duque, A., & Cano, J. Game-based Learning Approach to Cybersecurity. 2020 IEEE Global Engineering Education Conference (EDUCON), 1125-1132, (2022).

[30] Matovu, R., Nwokeji, J., Holmes, T., & Rahman, T. Teaching and Learning Cybersecurity Awareness with Gamification in Smaller Universities and Colleges. 2022 IEEE Frontiers in Education Conference (FIE), 1-9, (2022).

[31] Diakoumakos, I. Enhancing Cyber Security Education and Training through Gamification. Proceedings of the 2nd International Conference of the ACM Greek SIGCHI Chapter, (2023).

[32] Centigrade-https://cybersecurity.centigrade.de/. Accessed 22.06.2024

[33] Katsantonis, N. M., Kotini, I., Fouliras, P., & Mavridis, I. . Conceptual framework for developing cyber security serious games. (pp. 872-881). IEEE, (2019).

[34] Sathish, Kumar, Kannaiah., S., S., Aravinth. Designing Cybersecurity AI Based Awareness Games for Citizens: Best Practices and Future Directions. 407-412, (2023).

[35] Riedel, Johann CKH, Yanan Feng, and Aida Azadegan. "Serious games adoption in organizations–An exploratory analysis." Scaling up Learning for Sustained Impact: 8th European Conference, on Technology Enhanced Learning, EC-TEL 2013, Paphos, Cyprus, September 17-21, 2013. Proceedings 8. Springer Berlin Heidelberg, 2013.

[36] CyberGamesUk - https://cybergamesuk.com/. Accessed 24.06.2024

[37] CyberGame - https://cybergame.sk-cert.sk/. Accessed 24.06.2024

[38] Enhancing Cyber-attacks Awareness via Mobile Gamification Techniques. International Journal of Advanced Research in Technology and Innovation. (2022).

[39] Malone, M., Wang, Y., James, K., Anderegg, M., Werner, J., & Monrose, F. To Gamify or Not?: On Leaderboard Effects, Student Engagement and Learning Outcomes in a Cybersecurity Intervention. Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, (2021).

[40] Kahoot - https://kahoot.com/. Accessed 22.06.2024

[41] Cybrary - https://www.cybrary.it/. Accessed 22.06.2024

[42] CyberStart - https://cyberstart.com/. Accessed 22.06.2024

[43] TryHackMe - https://tryhackme.com/. Accessed 22.06.2024

[44] CybersecurityBoard - https://cybersecurityboard-a5dc4.web.app/. Accessed 01.07.2024

[45] Sani, S., & Shokooh, Y. Minimalism in designing user interface of commercial websites based on Gestalt visual perception laws (Case study of three top brands in technology scope). 2016 Second International Conference on Web Research (ICWR), 115-124, (2016).

[46] Humayun, M., Niazi, M., Jhanjhi, N., Alshayeb, M., & Mahmood, S. Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study. Arabian Journal for Science and Engineering, 45, 3171–3189. https://doi.org/10.1007/s13369-019-04319-2, (2020).

[47] Tsvetanova, A., & Stefanova, M. Key Cybersecurity Threats. Mathematics, Computer Science and Education. https://doi.org/10.54664/vflf8577, (2022).

[48] https://coltrane.ait.ac.at/. Accessed 06.07.2024

[49] Chernikova, O., Heitzmann, N., Stadler, M., Holzberger, D., Seidel, T., & Fischer, F. . Simulation-Based Learning in Higher Education: A Meta-Analysis. Review of Educational Research, 90, 499 - 541. https://doi.org/10.3102/0034654320933544, (2020).

[50] https://pixabay.com/de/illustrations/whiteboard-wei%C3%9Fe-tafel-3205371/. Accessed 20.05.2024

[51] H., Tabakova. Storytelling as a means of activating students' cognitive activity in the modern educational space. Naukovì zapiski Berdâns′kogo deržavnogo pedagogìčnogo unìversitetu, (2023)

[52] Maria, Kordaki. On the Design of Educational Digital Stories: The Ed-W Model. Procedia - Social and Behavioral Sciences, (2014)

[53] https://cybersecurityboard-a5dc4.web.app/storyboard/edit/vKDfmeleNxogebXzmO0g. Accessed 20.07.2024

[54] Angular - https://angular.dev/. Accessed 20.07.2024

[55] Firebase - https://firebase.google.com/. Accessed 20.07.2024

[56] Figma - https://www.figma.com/de-de/. Accessed 20.07.2024

[57] CyberSecurityBoard - https://github.com/MehrudinSabani/CyberSecurityBoard

[58] Shi, Y., Cao, N., Ma, X., Chen, S., & Liu, P. EmoG: Supporting the Sketching of Emotional Expressions for Storyboarding. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3313831.3376520, (2020)

[59] Гумарова, A., J.,, R., A, V., & G., K. MODERN FRAMEWORKS FOR WEB-APPLICATION DEVELOPMENT. Ġylym žźne bìlìm. https://doi.org/10.52578/2305-9397-2023-3-2-42-50, (2023).

[60] https://www.wearedevelopers.com/magazine/best-frontend-frameworks-for-developers/. Accessed 25.07.2024

[61] Pencheva, D., Hallett, J., & Rashid, A. Bringing Cyber to School: Integrating Cybersecurity Into Secondary School Education. IEEE Security & Privacy, 18, 68-74. https://doi.org/10.1109/MSEC.2020.2969409, (2020).

[62] Kanchana, D., Patchainayagi, D., & Rajkumar, D. (2019). EMPOWERING STUDENTS TO BECOME EFFECTIVE LEARNERS THROUGH ACTIVITY BASED LEARNING. Humanities & Social Sciences Reviews. https://doi.org/10.18510/hssr.2019.757.

[63] Lathigara, A., Tanna, P., & Bhatt, N. Activity Based Programming Learning. Journal of Engineering Education Transformations. https://doi.org/10.16920/JEET/2021/V34I0/157202, (2021).

[64] Kurian, B. The Study of Activity Based Learning (ABL) and their Challenges in Implementation for Higher Education Institutions. Asian Journal of Science and Applied Technology. https://doi.org/10.51983/ajsat-2022.11.2.3216, (2022).

[65] Patil, U., Budihal, S., Siddamal, S., & Mudenagudi, U. Activity Based Teaching Learning: An Experience. Journal of Engineering Education Transformations, 29. https://doi.org/10.16920/jeet/2016/v0i0/85433, (2016)

[66] Margaryan, A., Collis, B., & Cooke, A. Activity-based blended learning. Human Resource Development International, 7, 265 - 274. https://doi.org/10.1080/13678860410001676574, (2004).

[67] Hmelo‐Silver, C. Problem-Based Learning: What and How Do Students Learn?. Educational Psychology Review, 16, 235-266. https://doi.org/10.1023/B:EDPR.0000034022.16470.F3, (2004).

[68] Beringer, J. Application of Problem Based Learning through Research Investigation. Journal of Geography in Higher Education, 31, 445 - 457. https://doi.org/10.1080/03098260701514033, (2007).

[69] Gallagher, S., Stepien, W., & Rosenthal, H. The Effects of Problem-Based Learning On Problem Solving. Gifted Child Quarterly, 36, 195 - 200. https://doi.org/10.1177/001698629203600405, (1992).

[70] Subhash, S., & Cudney, E. Gamified learning in higher education: A systematic review of the literature. Comput. Hum. Behav., 87, 192-206. https://doi.org/10.1016/j.chb.2018.05.028, (2018).

[71] Liu, Z., Shaikh, Z., & Gazizova, F. Using the Concept of Game-Based Learning in Education. Int. J. Emerg. Technol. Learn., 15, 53-64. https://doi.org/10.3991/ijet.v15i14.14675, (2020).

[72] Delgado-Algarra, E. Gamification and Game-Based Learning. Research Anthology on Developments in Gamification and Game-Based Learning. https://doi.org/10.4018/978-1-6684-3710-0.ch043, (2022)

[73] Abdullah, H. (2018). Using Scenario-Based Learning for Pre-Service EFL Teacher Preparation. , 42, 83-123. https://doi.org/10.21608/jfees.2018.79284.

[74] Chan, Y., Rainey, J., & Park, C. (2019). Essentials of Scenario Building. Comprehensive Healthcare Simulation: Anesthesiology. https://doi.org/10.1007/978-3-030-26849-7_3.

[75] CyberEurope-https://www.enisa.europa.eu/topics/training-and-exercises/cyber-exercises/cyber-europe-programme/. Accessed 25.07.2024

[76] Moodle - https://moodle.com/de/. Accessed 25.07.2024

[77] Blackboard - https://help.blackboard.com/. Accessed 25.07.2024

[78] Edmodo - https://www.edmodo.com/lander/. Accessed 26.07.2024

[79] Skillsoft - https://www.skillsoft.com/de/. Accessed 26.07.2024

[80] DAPRA - https://www.darpa.mil/. Accessed 27.07.2024

[81] Seo, J. H., Bruner, M., Payne, A., Gober, N., McMullen, D., & Chakravorty, D. K. Using virtual reality to enforce principles of cybersecurity. The Journal of Computational Science Education, 10(1), (2019).

[82] Armas, C., Tori, R., & Netto, A. Use of virtual reality simulators for training programs in the areas of security and defense: a systematic review. Multimedia Tools

and Applications, 79, 3495 - 3515. https://doi.org/10.1007/s11042-019-08141-8 (2019).

[83] Dattel, A., Goodwin, T., Brodeen, H., Friedenzohn, D., Ochoa, O., Wang, H., Gao, P., Haris, S., & Parkar, I. Using Virtual Reality for Training to Identify Cyber Threats in the Bridge of a Ship. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 66, 1957 - 1961. https://doi.org/10.1177/1071181322661325, (2022).

[84] Using Extended Reality to Support Cyber Security. Advances in Electronic Government, Digital Divide, and Regional Development. https://doi.org/10.4018/978-1-7998-1562-4.ch008, (2020).

[85] Tight, M. Student retention and engagement in higher education. Journal of Further and Higher Education, 44, 689 - 704. https://doi.org/10.1080/0309877X.2019.1576860, (2020).

[86] Konak, Abdullah. "Experiential learning builds cybersecurity self-efficacy in K-12 students." Journal of Cybersecurity Education, Research and Practice 2018.1 (2018).

[87] Xu, Le, Dijiang Huang, and Wei-Tek Tsai. "Cloud-based virtual laboratory for network security education." IEEE Transactions on Education 57.3: 145-150, (2013).

[88] Krumay, Barbara, Edward WN Bernroider, and Roman Walser. "Evaluation of cybersecurity management controls and metrics of critical infrastructures: A literature review considering the NIST cybersecurity framework." Secure IT Systems: 23rd Nordic Conference, NordSec 2018, Oslo, Norway, November 28-30, 2018, Proceedings 23. Springer International Publishing, (2018).

[89] Möller, Dietmar PF. "NIST cybersecurity framework and MITRE cybersecurity criteria." Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices. Cham: Springer Nature Switzerland,. 231-271, (2023).

[90] Educasue - https://er.educause.edu/. Accessed, 29.07.2024

[91] Blue Team Labs - https://blueteamlabs.online/. Accessed 02.08.2024

[92] Over the Wire - https://overthewire.org/wargames/. Accessed 02.08.2024