Designing an Intuitive Cybersecurity Teaching Interface for Non-technical Educators

by

Xu Zheng Tan

School of Information Technology and Engineering,
University of Queensland.

Submitted for the degree of
Masters of Information Technology
in the division of Information Technology.

June 2023

ii

Xu Zheng Tan
xuzhengtan@uqconnect.edu.au

28th November 2023
Prof Michael Brünig,
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Simmons,

In accordance with the requirements of the degree of Bachelor of Engineering (Pass/Honours) in the division of Computer Systems Engineering / Electrical and Electronic Engineering, I present the following thesis entitled

"Designing an Intuitive Cybersecurity Teaching Interface for Non-technical Educators".

This work was performed under the supervision of Dr Marie Boden and Mr Joshua Scarsbrook.

I declare that the work submitted in this thesis is my own, except as acknowledged in the text and footnotes, and has not been previously submitted for a degree at the University of Queensland or any other institution.

Yours sincerely,

Xu Zheng Tan.

iv

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my supervisors, Dr Marie Boden and Joshua Scarsbrook, for their invaluable guidance and support throughout the course. Their expertise, encouragement, and constructive criticism were essential in shaping my research and improving the quality of my work. It was a privilege to work with them and I am grateful to have had the opportunity to learn from them over the course of two semesters.

Secondly, I would like to extend my heartfelt thanks to all of the participants who generously gave their time and energy to participate in the user testing for my research project. Your valuable insights, feedback, and contributions have been instrumental in shaping my research.

Finally, I would like to thank my friends and family for their encouragement throughout my academic journey, especially my parents who supported and encouraged me to further my post-graduate studies. Their contributions have been immeasurable, and I am truly blessed to have you in my life.

Abstract

Contemporary society's increasing reliance on digital technologies has created a growing demand for cybersecurity experts, resulting in a growing emphasis on teaching cybersecurity skills and concepts in higher education such as high-schools and universities. However, creating effective cyberlearning environments is often a complex and challenging task, particularly for educators without extensive programming and system administration knowledge. Additionally, there is a lack of education resources that cater to educators who lack extensive experience and knowledge in programming and system administration.

To address this gap, this thesis presents the design and implementation of a user-friendly cyber-teaching platform that enables non-technical educators to enhance the teaching of cybersecurity concepts through hands-on practical experience in a simulated environment. The platform aims to empower educators to create and manage cyber exercises and lesson plans effectively, with a focus on ease of use as a key success factor.

The primary goals of the project are to design an intuitive, user-friendly interface prototype of a cyber-teaching platform that does not require any prior knowledge to operate, and to identify key features needed for teachers to set up cybersecurity environments. The platform design is iteratively improved through continuous user testing and prototyping, incorporating feedback to ensure ongoing development and refinement.

The research evaluates the platform's impact on cybersecurity education, examining its potential to improve learning outcomes and enhance student engagement. The results of the research will contribute to the ongoing development and refinement of cyber-teaching platforms, with the potential to make significant contributions to cybersecurity education. By providing practical and accessible resources, this platform aims to equip educators with the tools necessary to prepare students for the increasingly complex world of cybersecurity.

Contents

1 Introduction

This thesis report describes a project of a design and implementation of a front-end system prototype with a user-friendly interface for a cyber-teaching environment. The intended use is for school teachers to set up cyber ranges easily, without prior programming and system administration knowledge.

1.1 Background

The growing reliance of contemporary society on digital technologies has significantly increased the demand for skilled workers in cybersecurity [1]. According to a report from McAfee, the cost of cybercrime damages in 2020 exceeded one trillion dollars [2], highlighting the importance of teaching cybersecurity in higher education and professional learning [2]. However, current teaching materials often fail to address the constantly evolving nature of web attacks. Passive learning, such as theory-based approaches is insufficient for students to gain a thorough understanding of core cybersecurity principles [3]. Practical, hands-on experience is essential for mastering the concepts and principles of cybersecurity, and technology such as cyber ranges and test beds offers a more practical solution for providing students with valuable learning experiences in a simulated environment [4].

However, setting up these learning environments can be overwhelming for non-technical educators such as high-school teachers [5]. Without prior programming experience and system administration capabilities, they may not be able to implement and set up these systems successfully. This poses a challenge for many colleges and high schools looking to incorporate cybersecurity concepts into their classrooms [3]. According to research findings [6], many educators teaching cybersecurity faced the trouble of resources that assumed prior knowledge, making it difficult to deliver effective cybersecurity education to students. This has resulted in a growing need for professional development courses to upskill educators, but this can be a significant obstacle due to the time constraints and costs involved [7, 6].

The successful implementation of effective cybersecurity education requires an intuitive education platform that provides teachers with the necessary tools and resources to create and manage cyber-learning environments. To address these challenges, this thesis presents the design of a user-friendly cyber-teaching platform that aims to allow non-technical educators to create and manage cybersecurity exercises and lesson plans easily. The project identifies the key features needed for setting up proper cyber teaching environments, creating an optimal solution for non-technical educators to present and set up lessons. Through multiple iterative user testing and prototyping processes, the platform's ease of use and effectiveness in improving learning outcomes and student engagement are evaluated. The results of this research will contribute to the ongoing development and refinement of cyber-teaching platforms, with the objective of making a significant contribution to cybersecurity education. The design of this platform hopes to address the critical need for accessible and practical resources for cybersecurity education, ultimately helping to close the gap in cybersecurity expertise and empower the next generation of cybersecurity professionals.

1.2 Motivation

Individual research has shown that there is a lack of literature on the design of user interfaces of cyber learning platforms that cater to non-experts. While there have been extensive studies on user-friendly interfaces, there is a lack of research on the topic of cybersecurity teaching environments. As a result, there is a need to address the gaps in the literature to gain a better understanding of user interfaces for cybersecurity teaching platforms. This research can contribute to the development of effective cyberlearning platforms, making cybersecurity education accessible and relevant to a broader range of learners.

The lack of research on this topic motivates the need to move this project forward. As such, it has the potential to lead to undiscovered possibilities and innovation in the field of cybersecurity education, addressing the growing demand of experts in cybersecurity. Furthermore, this research can provide valuable insights for the ongoing development and refinement of cyber-teaching platforms, enabling educators to create effective learning environments for cybersecurity education.

1.3 Research Question

To achieve the main objective of this project, which is to improve the usability and efficiency of the cyber-teaching platform for non-technical educators, several research questions need to be addressed. The primary research question is:

"How can we design a user-friendly interface that allows educators without programming and system administration backgrounds to create complex cybersecurity learning environments for educational activities?"

To answer this question, the project will also address the following research questions:

1. What are the key design features that should be included in a user-friendly cyber-teaching platform for non-technical educators?

2. What is the optimal approach to user testing and prototyping in the development of a user-friendly cyber-teaching platform for non-technical educators?

3. How can a user-friendly cyber-teaching platform be adapted to meet the needs of different types of learners and diverse student populations?

1.4 Scope

This thesis project is a collaboration between the Author and supervisors Dr. Marie Boden and Mr. Joshua Scarsbrook. It forms part of a larger project called Tiny Range, initiated by Supervisor Joshua Scarsbrook, which aims to develop a cyber-teaching platform. The back-end system is being developed by supervisor Joshua while the focus area of this thesis is to design the user-friendly front-end parts of the platform based on the existing back-end system. The project will concentrate on four main areas:

1. Designing an intuitive user-friendly interface prototype of a cyber-teaching platform that does not require any prior knowledge to operate.

2. Identifying key features needed for a teacher to set up cybersecurity environments.

3. Conducting continuous user testing and prototyping to iteratively improve the platform design.

4. Evaluating the impact of the platform on cybersecurity education, examining its potential to improve learning outcomes and enhance student engagement.

However, the project's scope does not cover the design of the learning environments from the student perspective. Additionally, the project will not focus on developing advanced technical skills and knowledge in cybersecurity or creating new cybersecurity principles. Instead, it will leverage existing principles and best practices to create effective learning experiences. The project will also not create new, unique individual cyber exercises. Finally, the project's design will be constrained by available time and resources and will focus on ensuring the platform's usability and effectiveness for non-technical educators.

1.5 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides a comprehensive review of the literature on cybersecurity technologies, and cybersecurity in education, including case studies on similar existing software, and the challenges faced by non-technical educators. Chapter 3 describes the approach direction for the design and implementation of the cyber-teaching platform, including its key features and functionality. Moreover, the platform is evaluated through iterative user testing and prototyping processes, with a focus on its ease of use and identifying key features that are necessary for improving an educator's experience in setting up learning environments. Chapter 4 then concludes the results and discusses the limitations of the project, offering recommendations for future research and development in this field.

2 Literature Review

This section of the thesis provides a comprehensive review of the literature on cybersecurity technologies, cybersecurity in education, case studies on similar existing software, and the challenges faced by non-technical educators. With the increasing importance of cybersecurity in today's digital world, it has become an important task to provide effective cybersecurity education and training to students, professionals, and individuals of all ages. This review aims to explore the current state of research on cybersecurity in education and identify gaps in the literature. By analyzing the existing research in this field, this review aims to provide a foundation that paves the direction for a design and implementation of an effective cyber-teaching platform that meets the needs of non-technical educators.

2.1 Technologies in cybersecurity

To gain a comprehensive understanding of how cybersecurity education learning environments are established, it is essential to examine the key tools that form the foundation of this process and how it is used in education. Two important technologies in cybersecurity are Cyber Ranges and Virtual Machines.

2.1.1 Cyber Ranges

Cyber Ranges are an essential tool used to teach cybersecurity principles and core concepts. It is a essentially a simulated environment designed to mimic a real world network, allowing users to practice various cybersecurity skills and techniques in a safe and controlled environment [8]. These ranges offers students a sandbox environment where they can safely explore and experiment with different types of applications and techniques of exploitation [9], they also allow users to use the same type of tools that they might use as a cybersecurity professional [3]. As a result, many schools and universities maintain continous cyber ranges for teaching activities and cyber competitions [10].

According to the National Institute of Standards and Technology (NIST) [11], the use cases for Cyber Ranges in education can be generalized as:

1. Individuals or organisations looking to train and educate operators.

2. Educators in schools (K-12, colleges) looking to implement basic up to advanced courses on cybersecurity education.

3. Individuals seeking training to move into cybersecurity related jobs and po-

sitions.

Cyber Ranges are extremely important tools for teaching cybersecurity skills. However, they can be quite complex to set up and operate, especially for non-technical educators [12]. These ranges typically require a range of technical skills, such as networking and system administration, to set up and maintain, which can be an instrumental challenge for educators without a technical background. Additionally, cyber ranges can be expensive to implement and maintain, requiring specialized hardware and software.

For these reasons, cyber ranges may not be a practical solution for all educators, particularly those who lack technical expertise or those who work in resource-constrained environments. In order to address this gap in the field of cybersecurity education, there is a need for more accessible and user-friendly cybersecurity teaching platform that can be easily set up and operated by non-technical educators. Such a platform could help to expand the possibilities of cybersecurity education and make it more widely accessible to a broader range of learners.

2.1.2 Virtual Machines

A cyber range is a simulated enviroment that provides realistic training scenarios to develop and improve cybersecurity skills. On the other hand, a virtual Machine (VM) is an isolated computing environment created within a physical computer. It allows multiple operating systems to run on the same computer simultaneously, and hence each operating system is isolated from the others.

VMs have become a popular tool in cybersecurity education due to their ability to create isolated, safe environments for students to practice and learn about various cybersecurity concepts [8]. By running multiple virtual machines on a single physical machine, instructors can create a variety of different network configurations and scenarios for students to work with [12]. The use of VMs has also been shown to enhance and improve student learning outcomes in cybersecurity education. In a study conducted by Bottazi et al [8], students who were taught using VMs demonstrated a statistically significant improvement in their knowledge of network security concepts compared to those who were taught using traditional methods. Similarly, Dhillon et al [13] found that the use of VMS in classrooms improved student engagement and participation in cybersecurity labs.

Overall, Virtual Machines (VMs) have emerged as a powerful tool for cybersecurity education and can be used for various purposes, such as creating safe and controlled testing environments for malware analysis, vulnerability testing, and penetration testing. However, similar to Cyber Ranges, the issue remains the same. Both of these technologies require a certain level of expertise to configure and maintain. According to research conducted by Beauchamp [6], current resources used for education purposes (such as Cyber Ranges and VMs) assume prior technical knowledge in order to properly manage and set up. Thus, educators may struggle with issues such as network configuration, storage management, and ensuring the security of the environments themselves. This gap in the field of cybersecurity education presents an opportunity for this thesis, where the design of a cybersecurity education platform targeted towards non-technical educators can be useful in bridging this gap.

2.2 Cybersecurity in Education

Cybersecurity is still a relatively new discipline in the field of education. Due to this, there is a debate surrounding the best approach to teaching this subject. Some believe that cybersecurity courses should prioritize teaching students about the principles and concepts of cybersecurity to build secure systems, whereas others argue that courses should focus on adversarial thinking in order to defend against potential attacks [13, 14]. Additionally, Schnieder [9] criticizes how current teaching materials in cybersecurity education ignore the constantly evolving nature of web attacks. This highlights the importance of hands-on learning in cybersecurity education as opposed to relying solely on abstract and theoretical lesson materials. In fact, research has shown that mastery of cybersecurity concepts and principles is most effectively achieved through the practical implementation of knowledge. Thus, Cyber Ranges and Virtual Machines have emerged as valuable tools for providing students with a hands-on learning environment [14].

2.3 Building blocks of a cybersecurity class

According to Jan et al. [4], a cybersecurity class can take on various forms, but the most common ones are cyber exercises such as serious games, solving complex assignments, or just following step-by-step instructions. These activities may or may not be supervised by an instructor who can be either a human or a machine. Jan et al [4] suggest that a hands-on cybersecurity class typically consists of three main elements: the sandbox, learning analytics, and class delivery methods.

Sandbox

In cybersecurity, a sandbox is an isolated and safe place for students to apply their knowledge and practice their skills. A sandbox is generally set up using a single virtual machine, or Cyber Ranges. Once the VMs are deployed and running, students can interact with them [15]. In the context of cybersecurity education, sandboxes can be accessed either locally or remotely, either as a single virtual machine (VM) or as a network of VMs. Remote access to the sandbox can be achieved through protocols such as Remote Desktop Protocol (RDP) for Windows systems and Secure Shell (SSH) for Linux systems. Console access is also available, provided by the virtualization platform, enabling students to manage the input and output of the VMs.

Learning Analytics

Learning Analytics technology, including educational data mining (EDM) [16] and learning analytics (LA) [17], plays an important role in understanding and enhancing teaching. These technologies enables educators to collect and analyze data in classes to provide feedback and insights to the students for them to reflect on their understanding of core concepts in class and enable them to monitor their progress [18]. Additionally, these analytics can also be used by the educator to assess student learning, identify areas that require improvement, and evaluate the effectiveness of classroom resources [4].

Classroom Delivery methods

Classroom delivery methods combine project-based learning, labs integrated with lectures, and the use of rich text instructions or games for cybersecurity exercises. These methods involve computer-assisted instructions with the utilization of a sandbox that may incorporate automated tutoring systems which enhances students learning experiences and promotes students' self-directed learning [19].

2.3.1 Types of hands-on cybersecurity exercises

Cybersecurity exercises in the form of simulation environments provided by Cyber Ranges and VMs are an effective tool for training students in a realistic and hands-on way to prepare them for real-life cyber threats. These technologies offer the flexibility for students to modify the parameters of the session, and compile and extract reports based on each student's performance, enhancing the overall learning experience. Cybersecurity exercises conducted within this ranges can take on different forms. Some of the most common training activities can be described as follows [20]:

1. Role-based Training

This practical training method allows students to assume different unique roles that takes place in a cyber threat. For example, the trainee can assume the role of the cyber defender, hacker, operator, or training instructor to emulate a scenario of a real-life cyber-attack situation [21].

2. Simulation Training

Simulation exercises are another common type of cyber exercise used in cybersecurity education. These exercises aim to simulate real-world situations and scenarios that students may encounter in their future cybersecurity careers. These exercises can range from simple scenarios, such as phishing attacks or malware infections, to complex scenarios involving network penetration testing and incident response.

3. Mock Attack Training

Mock attack training is a type of cybersecurity exercise that simulates an actual cyber-attack on a network or system. This exercise is designed to expose students to various cybersecurity threats and vulnerabilities and train them on how to respond and defend against them.

4. Competitions

Cyber-security competitions are aimed to develop the student's problem solving skills, teamwork, and cybersecurity knowledge by providing exercises based on real-life scenarios in a competitive environment. One such example of a competition is the "Capture-The-Flag" wide-area security hacking competition involving multiple teams [22].

The review of existing cyber exercises used in cybersecurity education can inform the design of a cyber-teaching platform that caters to non-technical educators. By identifying the key features required to support modern cyber exercises, the usability and effectiveness of the platform can be improved, resulting in authentic and engaging learning experiences.The examples of cyber exercises used in this section will also be utilized for the prototype design of the cyber-education platform.

2.4 Teaching cybersecurity from the perspective of educators

Insightful research done by Beauchamp et al [6], paints a general picture of an educator's perspective in teaching cybersecurity in classes. The paper explores how the use of cyber ranges in the context of education motivated educators in the cybersecurity space. Additionally, the research focuses on understanding the educator's motivation and identifying a few potential variations depending on the educator's teaching experience and instructional level.

This study is conducted using a mixed-methods approach of both quantitative and qualitative methodologies [23] and conducted using the VaCR (Virginia Cyber Range) and participants with a large range of experiences from novice to experience cybersecurity educators. This study resulted in four main discoveries [6].

1. Educators' Motivation: Most of the educators using the cyber range taught at the high school level and many did not have any formal cybersecurity background.

2. Importance of Cybersecurity Education: The study showed that educators recognized the importance of integrating cyber ranges into their cybersecurity classrooms. They also believe that using the cyber range improved their knowledge in the space and made them better overall educators.

3. Value of the Cyber Range: Educators highly valued the content that is provided by the cyber range for their classes. Furthermore, by introducing practical training through the use of cyber ranges, it contributed to much higher student interest and engagement.

4. Costs and challenges: Educators felt the need for additional preparation and professional development due to frustration from the user-friendliness of the cyber range resources for less-experienced users, suggesting the need for improvements. They were also emotionally stressed in classes because they were unprepared for the relevant cybersecurity content if they did not have prior experience or spent the effort to educate themselves.

The authors determined that even though the impact of cyber ranges on student engagement yielded positive results, current cyber range resources are not intuitive for educators without proper background in cybersecurity. Results from close-ended questions regarding educators' expectancy of success using cyber ranges for cybersecurity teaching also discovered new findings. Figure 1 depicts the responses for items within the expected range of success.

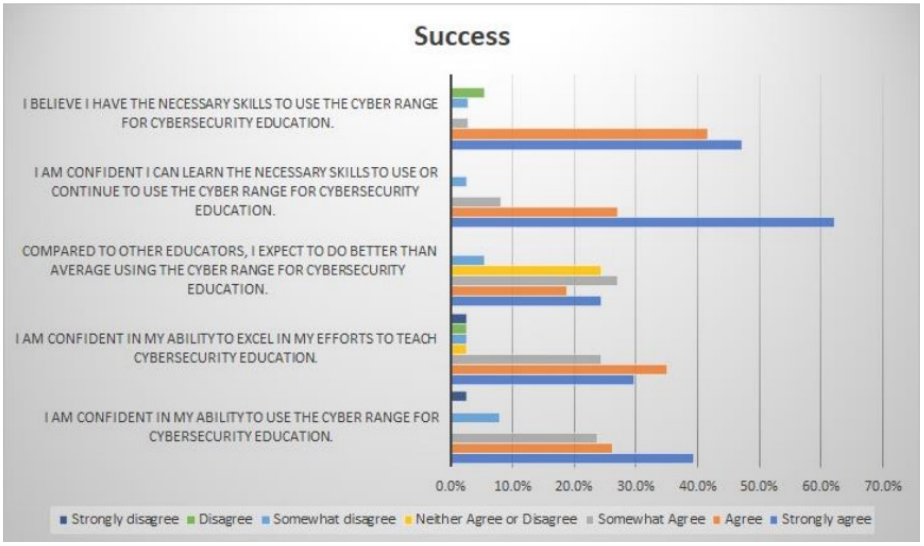

Figure 1: Bar Chart of Expected Success Close-Ended Questions [11]

The results [6] found that the educators who agreed to the questions shared a prior experience and background in cybersecurity. These educators felt that the cyber range used in the testing was intuitive and easy to use. On contrast, educators who disagreed shared a lack of experience in cybersecurity. Many of these educators shared that they found the cyber range challenging to use and found instructional materials to support educators to be lacking. These educators without prior knowledge or experience in cybersecurity also expressed the need for additional professional development and user-friendly resources designed specifically for beginners. These requirements are necessary to enhance their confidence in effectively teaching cybersecurity education using cyber ranges.

Cyber Range Integration into Education

Most educators agreed that the integration of the VaCR(cyber range) in cybersecurity education were important as it facilitated hands-on practical application of the course content and enhances student learning and understanding of core concepts in a safe environment [6]. The use of the cyber range also strengthened the educator's cybersecurity pedagogical content knowledge (PCK) and their ability to become better cybersecurity educators. These findings align with previous research that emphasizes the positive impact of technology on pedagogical practices when educators feel more confident when utilizing it effectively [24].

Student Interest and Engagement

Educators also highly valued the use of cyber range as a resource to improve their cybersecurity-related courses. By providing engaging and accessible hands-on activities through gamification and capture-the-flag activities, the study found that students showed a higher student interest [6]. Prior research has shown that sustained student interest plays a significant role in influencing attention, goals and learning outcomes [25]. Sustained student interest leads to a deeper appreciation and overall better outcomes, thus efforts should continue to integrate the use of cyber ranges in cybersecurity education programs to promote student interests and engagement.

Educator Time and Effort

Many educators also acknowledged the importance of continuous learning and a professional mindset when learning the cyber range is neccessary in order to ensure preparedness and competence in the classroom [6]. Often, educators teach multiple subjects that require substantial effort and time for lesson preparation. Prior studies have shown that learning about effective teaching practices and preparation time have influenced educator's adoption of these practices [26] , they also influenced how likely the educators were to persist at teaching [27]. Therefore, the study states the importance of academic administrators allocating additional time and resources for professional development and content preparation to integrate these cyber ranges effectively [6].

Overall, these findings provide insights into the experiences of educators using a cyber range for cybersecurity education and provides informed recommendations for improving the usability for cyber ranges in an educational setting. These insights gained from educators who use cyber ranges to teach are highly valuable and essential in making informed decisions about the integration and ongoing improvement of a cyber-teaching platform prototype. The results from this study also reinforce the need of having more accessible resources in cyber teaching platforms, supporting the development of a user-friendly cyber-teaching platform for non-experienced educators as presented in this paper.

2.5 Visual Programming

The next section of the study presents case studies that focuses on gaining a deeper understanding of interface design and finding the optimal approach to creating a user-friendly platform that reduces the barrier of entry into cybersecurity. Case studies are a valuable research method that allows researchers to utilize real-world examples to explore more specific phenomena in their context. In this case, this case study's focus is on examining the design and implementation of interfaces that support beginner-friendly user usage. Furthermore, additional focus will be allocated to identifying interfaces that cater to teachers, identifying the necessary features and functionalities needed to create and manage classes and challenges and preferences faced by educators while using education platforms. This case study provides an opportunity to explore the nuances of interface design and ultimately discover some effective strategies for creating a user-friendly cyber-teaching platform that empowers educators to effectively set up cyber ranges and teach cybersecurity classes.

Early case studies were focused on identifying a strategy to lower the barrier of entry to the cyber exercise teaching platform for non-technical users. These findings led to the concept of visual programming that explores the potential of utilizing block programming to enable the creation and management of cyber exercises. In this context, visual programming is a type of programming that utilizes graphical elements such as shapes, symbols, and flowcharts to produce code logic instead of traditional text-based coding [28].

2.5.1 Block Scripting

To find an optimal solution for the functionality of creating cyber exercises, a case study was conducted on a pre-existing block programming method - Blockly.

A study conducted by Benjamin et al, introduced a novel system that was designed to introduce children into programming by leveraging visual approaches [29]. This system used a scratch script which was implemented using the Scratch Programming language [30], which utilized code blocks from the Scratch Community blocks system.

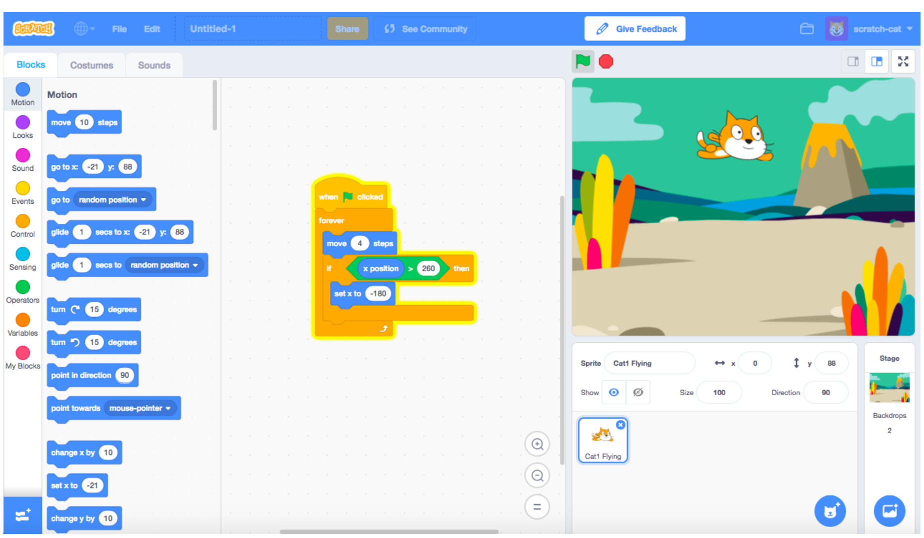

Figure 2: The Interface of Scratch

This scratch script allowed the children who were beginners to programming to perform iterative programming by dragging and stacking blocks into the canvas to create code logic. After pressing the "submit" key, the script would iterate through the blocks and execute the actions. According to the author, the Scratch language follows and object-centered programming method, which differs from the typical object-oriented approach commonly used for programming languages. Each visual block of the scratch script is a programming primitive that contains distinct behavior that can control the actions of graphical objects displayed on the screen [30].

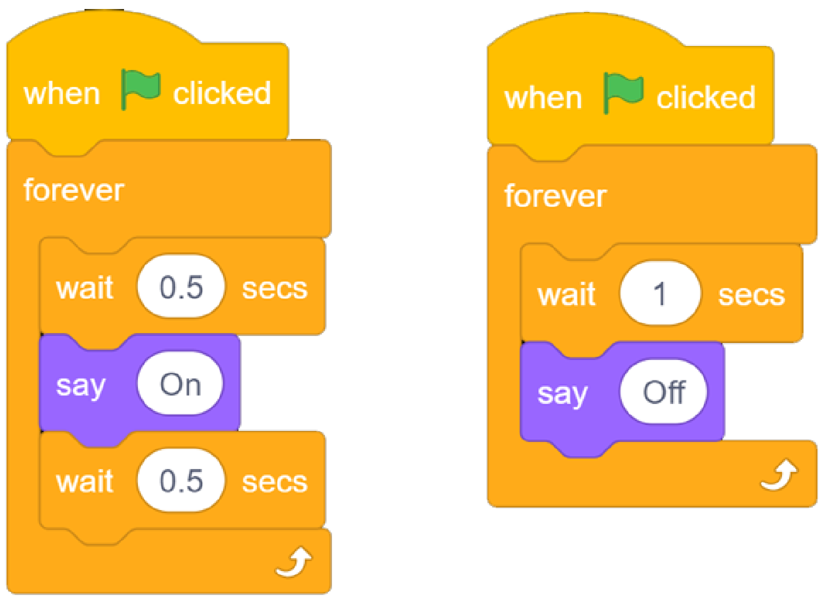

Figure 3: A blockly script, consisting of stacked visual blocks.

The research findings demonstrated the effectiveness of using visual programming when introducing children to programming. By presenting programming concepts in a visual and intuitive manner, the study found that young learners were much more motivated and engaged to programming, regardless of their young age. Overall, the findings indicated that visual programming using block scripting proved to be an effective strategy to introduce non-technical users to programming, it helps break down the barriers of entry for individuals with limited programming experience. These findings proves to be helpful for designing an interface and strategy for teachers to create cybersecurity exercises for classes.

By incorporating the block method into the creation of cybersecurity exercises, the prototype can enable educators to design cyber activities without the need for extensive programming and system administration skills. This intuitive drag-and-drop functionality from the block scripting method can allow educators to construct code logic by stacking blocks, facilitating the creation of complex cybersecurity exercises in a more accessible way.

2.5.2 Flow Scripting

Another type of visual programming focusing on the ease of use for non-technical users is flowchart-based programming. Flowchart-based programming is essentially a visual programming paradigm that represents the structure, code, and flow of a program using flowcharts. Instead of writing lines of code traditionally [31] , users can create flowcharts consisting of various shapes and arrows to visually depict the sequence of actions, and control the flow within the program.

The process of connecting a flow chart follows a visual and intuitive approach that resembles connecting elements in children's activities such as connecting the dots, making it easy for beginners to follow [32]. Furthermore, the step-by-step approach of connecting flows is similar to solving puzzles or completing a sequence, making it a more visually stimulating, and enjoyable process for users, including non-technical individuals.

One such example of flowchart programming is Node-RED. Node-RED is a popular visual programming tool that provides an easy way to create applications through the connection of predefined functional blocks, known as nodes, to define the flow of data and actions. It is designed in a way to make it user-friendly and accessible, allowing individuals with little programming background to participate in the development process [33].

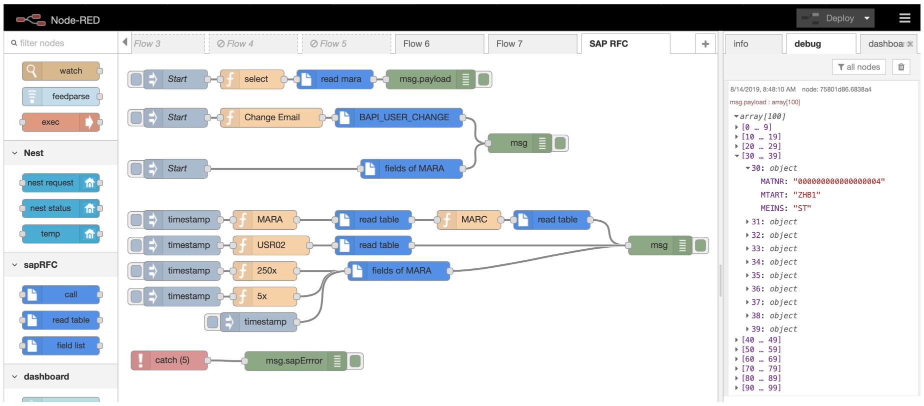

Figure 4: User Interface of the Node-RED platform

Similar to blockly and the block-scripting approach, Node-RED simplifies the creation of code logic using a drag-and-drop approach. In the Node-RED platform, users can select nodes from a library, such as input, output nodes and processing nodes, and connect them to build a flow-chart structure. These connections between each node represents the flow of data and events, helping users to define and visualize the flow of logic and fucntionality of their programs [33]. By utilizing the system, non-technical users can quickly prototype and deploy applications leveraging the visual interface. Thus, since then Node-RED has gained great popularity and adoption in domains such as IoT, home automation, and data integration due to its simplicity and ease of use [34].

Overall, flowchart scripting presents a similar solution to block scripting, as both focuses on the ease of use of constructing a logical flow. Users using block scripting drag and drop blocks together to build the program's functionality and define the program's flow. Similarly, flowchart scripting uses the same visual approach using connecting nodes to visualize logic structure. Both provides a visual and tactile programming experience while also eliminating the potential of syntax errors and code correctness while encouraging logical thinking and problem-solving skills. As it seems, both approaches look to be good candidates for prototyping, specifically for the feature of allowing educators to create cyber exercises, with the goal of increasing the accessibility and ease of use while removing the technicality and experience needed.

2.6 Designing for E-teaching platforms

In the realm of educational technology, it is important for the educator's platform to be effective in creating engaging and efficient learning experiences. These interfaces play a crucial role in helping educators to teach classrooms and support interactions with students. In many ways, the success of the e-learning platform depends on the quality of the user interface [35]. In addition to User Interface Design principles, the next part of the literature review studies the principles and concepts of learning and teaching in the design of educational interfaces.

2.6.1 UI Principles for education platforms

E-teaching is defined as teaching through electronic devices such as computers, and the Internet. According to Faghih et al [35], all platforms that focus on teaching and learning using these technological devices should adhere to the following characteristics:

1. All educational content should align with the intended learning objectives.

2. The curriculum should be tailored to fit the characteristics and tools of e-learning.

3. Factors that influence education should be taken into account.

4. Both individual and collaborative learning, as well as group teaching, should utilize suitable methods.

5. Multimedia elements such as voice, images, and text play an important role in presenting lessons.

2.6.2 Features required for effective and intuitive education platforms

To shed light on the key features that are required for e-learning platforms, Lee et al [36] develop a conceptual framework for an e-learning platform design to support researchers and educators in developing an effective platform for teaching in higher education. The Author states that each learning platform's required features and characteristics are dependent on the main usage and intended purpose of the application within the field of education [36]. Hence, research on instructional design begins their focus on understanding the needs and outcomes of e-learning platforms in education, particularly emphasizing the creation of advanced features to facilitate more effective learning experiences. An example is the implementation of features such as media exchange (audio, images, and videos), facilitated file sharing, and instant discussion. It is found that by implementing these functionalities, student participation in online discussion is greatly increased, further enhancing the platform's use in education [37].

Lee et al [36] also emphasizes the benefits of rich media influences as a learning and teaching tool as the increase in media usage tends to lead to positive experiences that result in an increase in student assignment-completion productivity. As such, it is also important to avoid monotonous interface backgrounds and overly formal design arrangements when designing an educational platform.

Al Ajlan [38] classifies that in order for an e-learning platform to be effective and highly functional, there is a need to incorporate three main features - learning tools, support tools, and technical tools. On the other hand, Garrote [39] asserts that the most appropriate features for e-learning platforms should be instead distribution tools, communication tools, Interactive tools, and course Administration tools. However, the author argues that a successful platform involves more than a simple combination of teaching content and online functionality. Instead, it should require incorporating a diverse range of features that address four main aspects: sharing, searching, networking and organizing [36]. These features aim to fulfill the needs of demand, flexibility, prompt delivery, and an efficient education environment. The four features can be described by the figure below:

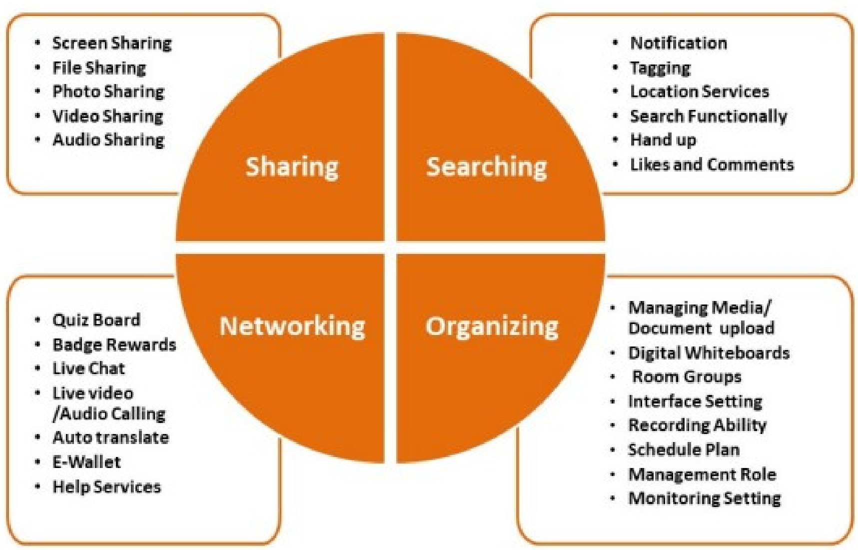

Figure 5: The four tool categories in education platforms [new 7]

The four tool categories can be summarised as follows:

1. Sharing tools:

A sharing tool that enables both instructors and students to upload various types of media files and text documents. According to the author, sharing tools should encompass features that enable instant acquisition of media, links, and files during live teaching sessions. Furthermore, the author suggested an enhancement for the sharing functionality which involves the introduction of a cloud-based folder where students and teachers can easily access all accumulated media and files with just a simple click [36].

2. Searching tools:

Easy access to information exploration tools for instructors and students during their learning activities. This segment has the potential to expand based on the required use and feedback of students and educators. But, some recommended features by the author such as notifications, tagging, location, search functionality, hand raising, likes, and comments could be incorporated to improve the experience of the education platform [36]

3. Networking tools:

Networking tools promote interaction, provide rewards, reactions, and immediate feedback, fostering student engagement and collaboration to enhance the learning experience. These can be tools such as quiz boards that enable real-time multiple-choice knowledge evaluation games or formative assessments such as voting and student opinion polls. Furthermore, badge rewards in the form of stickers or emoticons play an important role in motivating students and fostering positive attitudes [36].

4. Organizing tools:

Course organizing tools facilitate monitoring and recording of the educational process. These tools also allow instructors to evaluate and manage courses, as well as customize screen layouts based on student preferences and instructional needs. Pireva et al [40] emphasize the importance of student control over the screen interface to promote interaction. Thus, features such as digital whiteboards can be used to improve student responsiveness and facilitate presentations in various classes. Additional tools such as schedule planning can also help aid both teachers and students to schedule classes, announcements, reminders, and events, helping them stay organized with assignment submissions and agenda reminders.

3 Methodology

This next section of the thesis presents an iterative design and prototyping process which was employed throughout the development of this project, along with the findings from each user testing session. This section aims to provide a detailed account of each step taken throughout the project, leading up to the design and implementation of the final prototype.

3.1 Iterative design and prototyping

The iterative design and prototyping approach have been widely recognized and used in the industry of user-centered and intuitive user interfaces and is an essential process in all development of user interfaces [41]. This approach of prototyping involves creating multiple iterations of a design, gathering feedback through user testing, and applying modifications based on the findings. This process is repeated until the desired user experience is achieved. This strategy frequently evaluates prototypes that increase in technological complexity. Furthermore, conducting these evaluations in environments in which the platform will ultimately be used, will help to effectively address all kinds of contextual factors [42].

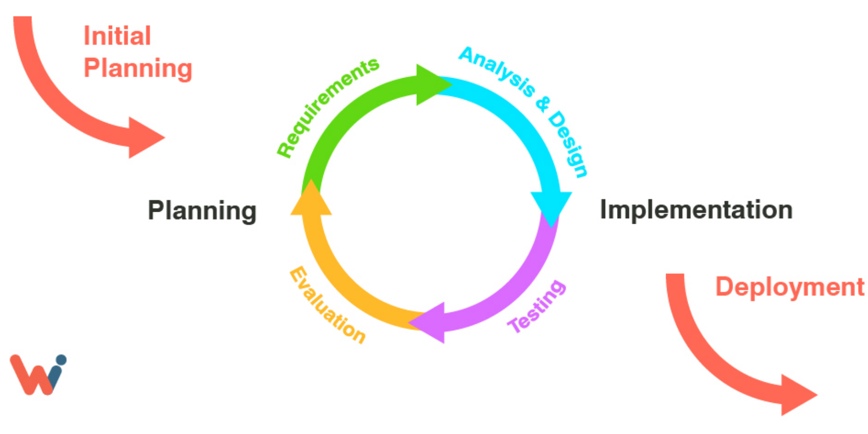

Figure 6: Iterative Design Model [42]

Over the course of the project, the iterative design process follows the iterative process that was presented by Nielsen, J [43] these key steps were as follows:

1. Understanding User Needs

This first step begins by outlining the understanding of user needs through comprehensive user research. However, individual research suggests that there is a lack of literature on the design of user-interfaces of cyber-platforms for non-technical educators. Thus, further research was focused on understanding the needs of edcuators who teach cybersecurity (section 2.2, section 2.3), and the features that were needed on the platform. Additionally, further research was conducted to discover ways to incorporate features that cater to non-experts without prior programming (section 2.5)

2. Generating Design Ideas

Based on the insights gained from individual user research, a few early iterations were generated through brainstorming using early sketches and low-fidelity prototypes. These earlier prototypes were less focused on the aesthetics of the design but more on solving the problem which were allowing non-technical educators to set up cyber exercises and plan lessons.

3. Creating Prototypes

These design ideas gained in the previous stage were then turned into interactive representations of the user interface. In this process, low-fidelity prototypes were converted and improved into high-fidelity prototypes which are ready for user testing. Additional navigational and interactive functionalities were also added.

4. Testing and Evaluation

Once the prototypes were created, they were tested with actual users, in this context, the users were high school teachers who were teachers in the IT sector. This usability testing included tasks, observations, and questionnaires to access user experience.

5. Analyzing Feedback

The feedback collected from users was critically analyzed to identify areas of improvement and areas where the design aligns with user needs and expectations. The early feedback gained resulted in major gains in usability as the true "usability catastrophe" were identified and improved upon, this is typical in early iterations according to Nielsen, J. [43]. This feedback that was attained from this process was used to guide the decision-making process for the next iterations.

5. Iterate and Refine

Based on the feedback and analysis collected, changes were made to the design, addressing all of the usability issues, and design direction, enhancing the overall user experience. This step leads to the creation of a new iteration of prototypes.

6. Repeating the process

This iterative design continues with each new iteration, incorporating feedback, making improvements, and conducting user testing again. Ideally, each iteration would be better than the previous version, and fewer issues in the prototype would be found through each user-testing session, as this is the general nature of iterative design [43].

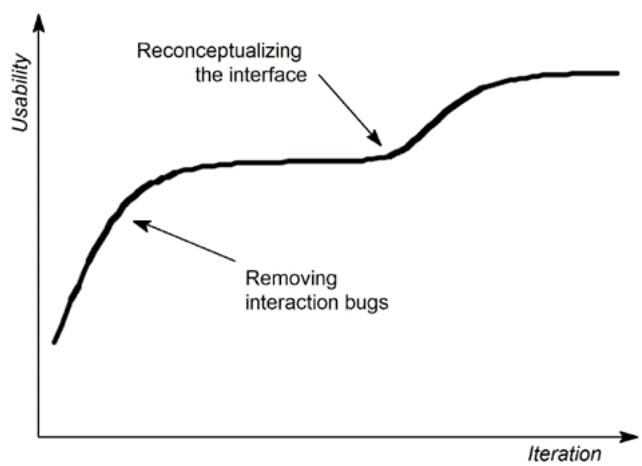

Figure 7: Measured usability of prototype will go up for each iteration.[43]

3.2 Early State of Tiny Range

The early stage of the platform, Tiny Range was developed by supervisor Mr Johsua Scarsbrook. This early version of the platform provided a student's version of the platform that showcased various cyber exercises set up on cyber ranges, varying in difficulty and requiring different skill sets.

To facilitate an educational event - Girls Do Cyber, aimed at introducing high school girls to the field of cybersecurity, Tiny Range was utilized to conduct a series of cyber exercises. These exercises varied from beginner to advanced levels, allowing students to apply their IT knowledge and skill to attempt each challenge effectively.

Figure 8: The user interface of the student's version of Tiny Range

Figure 9: An example of a challenge in Tiny Range

Essentially, two versions of Tiny Range will ultimately be developed and implemented, the student platform and the teacher's platform. Using this early version of the student's version of Tiny Range as a starting point, the goal of this project was to design and implement the teacher's version of the platform, a user interface for non-technical educators to set up, manage and customize these cyber exercises.

3.3 Goals

1. Successfully design an intuitive user-friendly interface prototype of a cyber-teaching platform that does not require any prior knowledge to operate.

2. Identify and define the key design features that contribute to a user-friendly cyber-teaching platform for teachers.

3. Develop an optimal approach for user testing and prototyping in the development process of the platform.

4. Iteratively improve the design of the platform and reduce usability issues through user-testing findings.

The success of these goals will be determined by response and feedback after conducting user-tests with cyber-teaching which does not require any prior knowledge to operate.

3.4 Generating Design Ideas

In the process of iterative design, the step of generating design ideas plays an important role in addressing the challenge of assisting non-technical users in setting up and managing cyber exercises. The main approach to this section was to discover ways of making a difficult task requiring prior technical experience and expertise into an easy and manageable task for non-technical users. A few key design ideas that emerged during this exploration were the utilization of visual programming, specifically, block programming and flow scripting. This section delves into the details of how visual programming was considered and implemented as a potential solution, it also explores the low-fidelity prototypes that were generated as a result of implementing this key concept.

3.4.1 Visual Programming

As described in (section 2.5), Visual Programming is a programming paradigm that enables the creation of code logic using graphical elements, such as shapes, symbols, and flow charts instead of the traditional text-based coding method [44]. This approach provides a graphical representation of the program's structure and allows users to visually connect and arrange these elements to define the program's behavior [45]. This approach simplifies the programming and system administration process by eliminating the need to write complex lines of code, reducing the technical requirements from users [46].

The idea of utilizing this concept of programming came after the case studies were done (section 2.5). Two main types of visual programming were identified, the block programming method (Blockly, Scratch) [47, 48], and the flow-Scripting method [33]. Thus, based on these methods of visual programming, a few low-fidelity prototypes were created for the feature of allowing educators to create cyber exercises using visual elements.

3.4.2 Low-Fidelity prototypes

The focus of these early prototypes was to create conceptual prototypes for the cyber exercise creation method. The main functionality of this feature is to simulate the setup of a cyber exercise environment. This feature allows users to set individual parameters and values of a virtual machine (VM), as well as establish a connection to the Secure Shell (SSH) for remote access. The VM will then be routed to the host router, and can then be accessed.

The GUI layout was inspired by Visual Van Gogh [49], a visual programming tool designed for creating graphical user interfaces (GUIs) without the need for traditional text-based programming. This early prototype GUI layout includes four main features, the toolbox, the canvas, the output screen, and the properties window.

1. Toolbox - The toolbox contains a set of code blocks that represents different actions and commands that can be used to create and customize cyber exercises. These code blocks serve as building blocks for constructing the code logic.

2. Canvas - The canvas is where the user drags and drops to visually arrange the code blocks to form the desired sequence of actions. It also provides a visual representation of the code logic that is being created.

3. Output Screen - The output screen displays the code logic after the code blocks are joined together. It allows the user to review and verify the logic before finalizing and constructing the cyber exercise.

4. Properties - The properties window provides a way to customize and modify the fields and parameters of the code blocks. It also offers additional flexibility and customization options for creating individual-tailored cyber exercises.

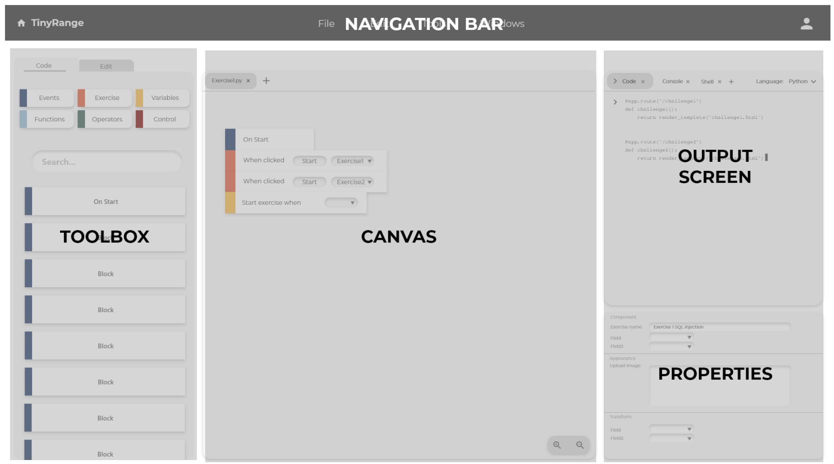

Figure 10: GUI layout including the toolbox, canvas, output screen, and properties window

Block-Programming early prototype

Using case studies of Blockly and Scratch (section 2.5), an early conceptual prototype was designed using AdobeXD. This block prototype feature uses the previous GUI layout as a base and builds upon it. On the left side of the screen, there is a toolbox containing color-coded code blocks. Each code block serves a different function and can be dragged and connected together in the center section of the GUI, the canvas. By connecting the code blocks together on the canvas, users can set up and edit the individual parameters and values of the VM. These details can then be displayed on the right panel of the screen, the output screen allows users to review their code logic in real-time to assess its functionality. Additionally, the properties window, situated at the bottom right panel, provides additional customization options for the cyber exercise where educators can input the exercise name, upload lesson images, and make other relevant modifciations to personalize the teaching experience.

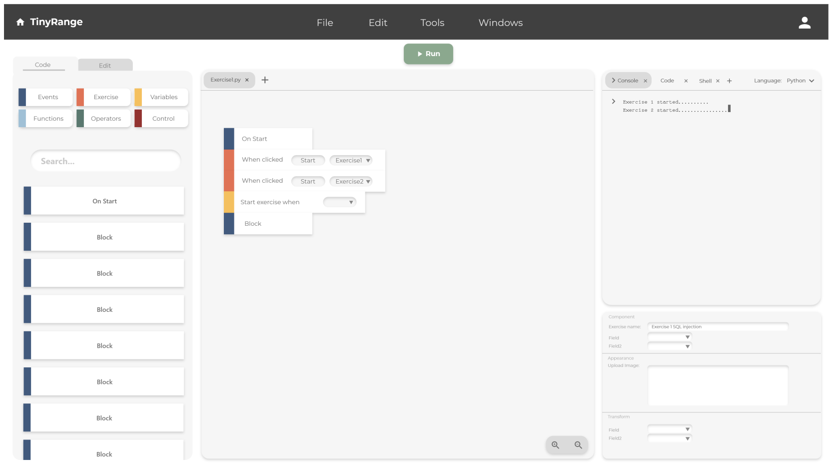

Figure 11: Early design of the Block-Programming approach to setting up cyber exercises

Flow Scripting early prototype

The flow scripting prototype utilizes a GUI that is similar to the block prototype but with a different programming approach. In this prototype, the GUI incorporates flow scripting similar to Node-RED [34] (section 2.5.2). This type of programming approach enables users to create structured code flows by dragging and dropping predefined code blocks, known as nodes onto the canvas and connecting them. These connections between each node represents the flow of data and events, helping users to define and visualize the flow of the cyber exercise.

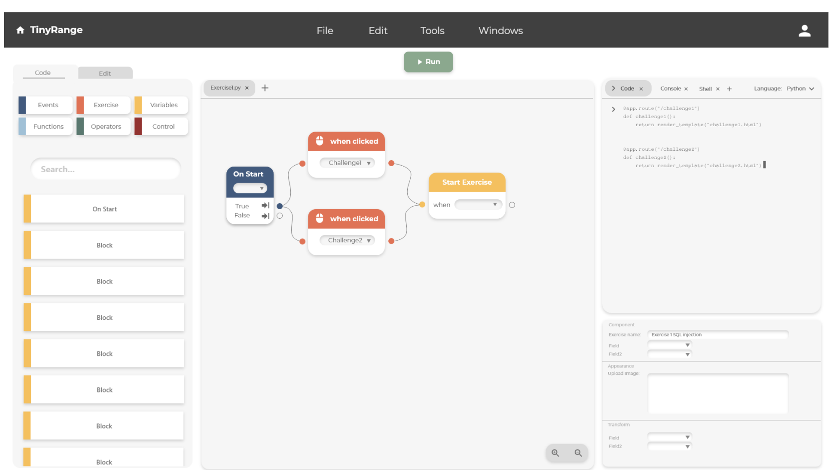

Figure 12: Early design of the flow scripting approach to setting up cyber exercises

A dashboard prototype was also designed to visualize the navigability of the education platform. The early dashboard design focuses on the function of setting up and managing cyber exercises.

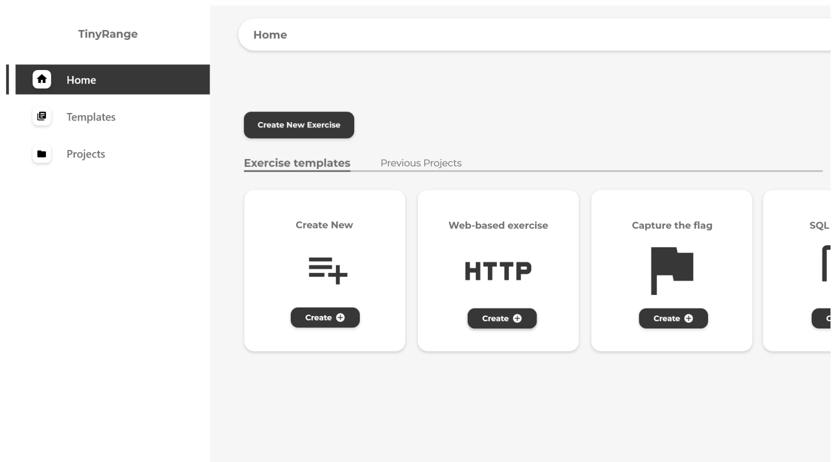

Figure 13: The early GUI design of the dashboard of the Tiny Range platform.

Essentially, this section provides an initial early approach to providing a solution to enable non-technical users in setting up cyber ranges. This early prototype leads to the development of a higher fidelity prototype that effectively introduces additional key features and functionality that are required in educational platforms.

3.5 First Iterations

The next section, will discuss the initial iterations of prototyping that were undertaken to evolve the low-fidelity prototype in the previous section into a high-fidelity prototype which added functionality and navigability. The goal of this phase of prototyping was to create a more refined prototype that could be used for user testing and evaluation.

3.5.1 Cyber exercise creation

During this process, a decision was made to use block programming as the primary cyber exercise creation method in the high-fidelity prototype rather than the flow-scripting method. This was due to several factors including the easier implementation (due to existing frameworks, e.g Blockly [47]), a wider range of usage, and overall suitability for the intended purpose.

Block-programming approach

The high-fidelity version of the cyber exercise creation method improved the overall GUI layout and design, these improvements were done based on personal observations and supervisor feedback. The main modifications involved restructuring the code blocks to create larger blocks that group together elements with similar logic. Furthermore, the code blocks were redesigned to feature jigsaw-like shapes, enabling seamless connections between them. These enhancements and improvements were implemented to align with the blocky [47] program framework, which was intended to use for the implementation of the platform. Additionally, additional GUI design improvements were made such as the enlargement of canvas size, and navigation bar for better usability.

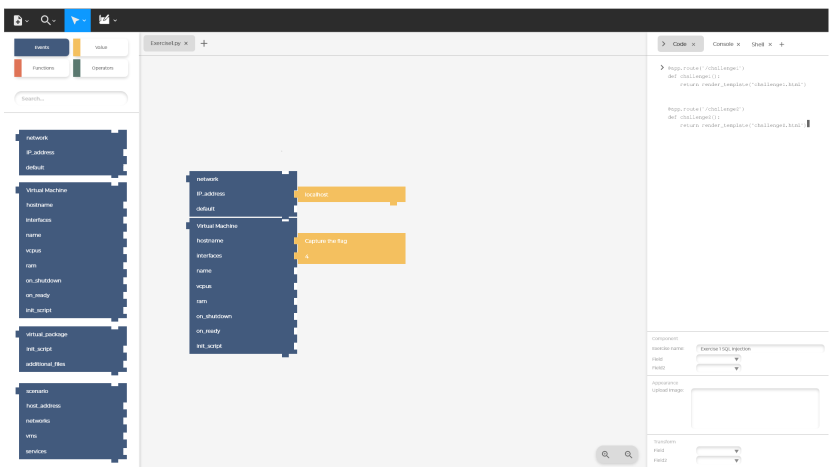

Figure 14: First Iteration of the cyber exercise creation feature using the block approach.

Limitations: However, a few limitations were observed to this approach. While the interface was designed to be intuitive, non-technical users may still struggle with the complex process of setting up virtual machines required for the cyber exercise. This was largely due to confusion and uncertainty about how to go to the next step without clear instructions as technical terminologies and parameters to fill up were still unfamiliar to new users.

Wizard Interface approach

To solve the main limitation of the lack of direction from the block-type approach, a new type of interface - the wizard interface was explored. A wizard interface is a user interface design method that guides users through a step-by-step process to complete a task or achieve a specific goal [50]. It often provides the users with a series of sequential and well-defined next steps, often accompanied with explanatory instructions and interactive elements, to ensure a structured user experience and to provide sufficient guidance [51].

In the context of assisting educators in setting up cyber-learning environments, the wizard interface can provide a step-by-step approach followed by explanatory and instructional elements allowing non-technical educators to follow along and create effective cyber exercises to their requirements.

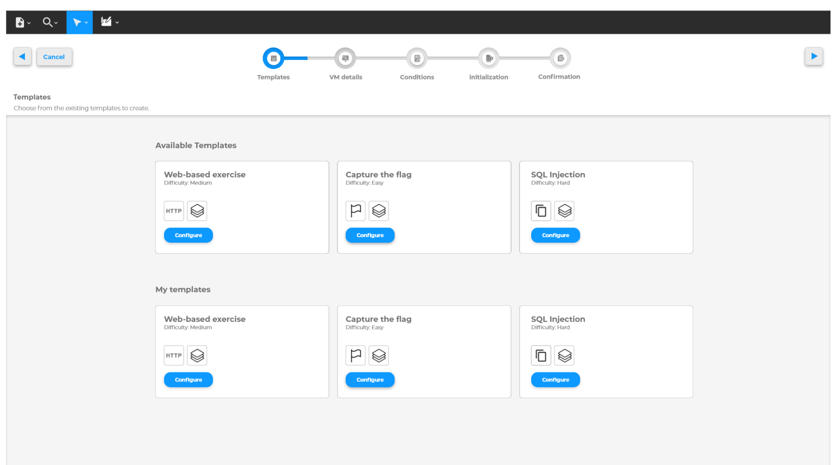

Figure 15: Template selection page for cyber exercises

Using the concept of guiding users through tasks, the wizard prototype begins by allowing the educator to select a template for the type of exercise they wish to set up. For example, if they select the web-based exercise template, the interface would load up a predefined template for this specific exercise with parameters assigned to set up this environment.

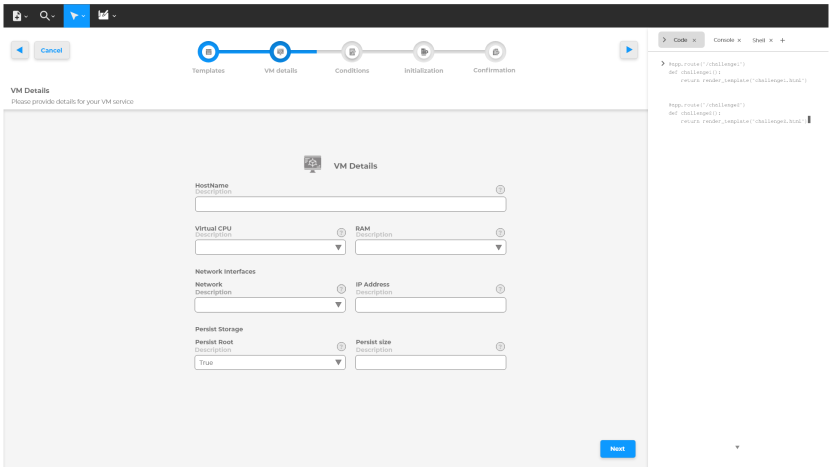

Figure 16: Users fill in parameters and fields to their requirements.

Once the template is selected, the user will be brought through a series of step-by-step tasks to complete. These tasks include necessary fields and parameters for the user to select or fill up, the user is able to fill in values to set up the cyber exercise to suit their own classroom requirements. Furthermore, to help users understand the type of value to input for each field, help widgets that stores a set of instructions that popup when clicked are placed on every field.

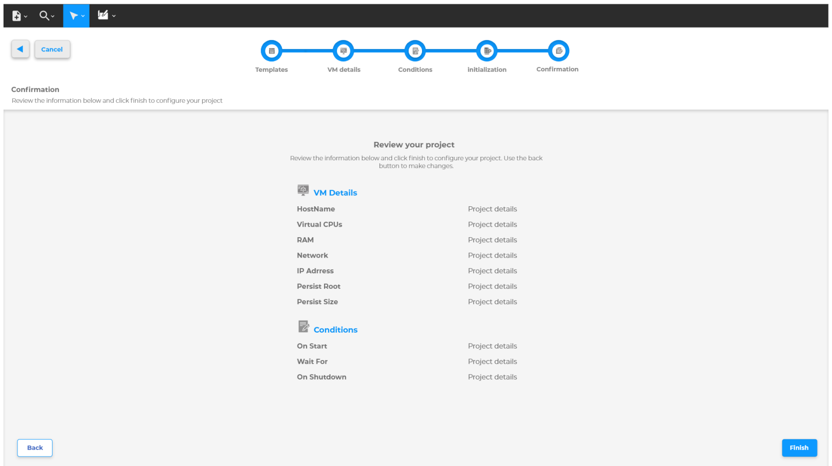

Figure 17: Review page to review all parameters before building cyber exercise.

Finally, after successfully filling up all of the fields and parameters, the user would be able to perform a final review before building the cyber exercise.

Limitations Even though this approach to setting up cyber exercises seemed to be more guided compared to the block-type approach, there seemed to be a limitation of the a lack of flexibility and customization to be considered [52]. This limitation is mainly due to the predefined structure and sequence of the wizard interface whereas a drag-and-drop approach of the block-type approach offers a higher level of flexibility. This is because users can create their own workflows, define their own logic, and arrange blocks in various combinations to achieve desired outcomes.

Hybrid Approach: Combining Block Type and Wizard Interface

In an attempt to leverage the benefits of both block-type programming and wizard interfaces while reducing their limitations, a hybrid approach was designed. This approach attempts to combine the flexibility and customization of block-type programming with the structured guidance of the wizard interface. This approach employs a drag-and-drop functionality where users can construct code logic by assembling blocks, similar to the block approach. Additionally, to provide a more guided and structured user experience, a wizard progress bar is implemented at the top of the interface which serves as a visual indicator, showing users their current step and overall progress within the setup process. Each step corresponds to a specific task or objective that users will need to complete to advance in the wizard interface.
Thus, with three of the different cyber exercise creation prototypes implemented, the decision on which approach to implement is ultimately dependent on the findings from user testing sessions.

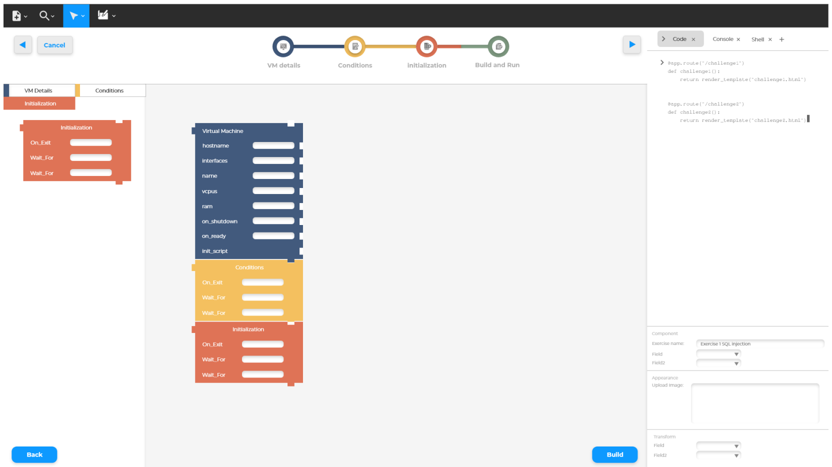

Figure 18: Hybrid approach prototype combining both approaches.

3.5.2 Lesson Planner Feature: An organization tool

The literature review conducted (section 2.6) highlighted the need for effective organization tools within education platforms which allows instructors to evaluate and manage courses. This key feature led to the implementation of a lesson-planner feature prototype, which provides educators with the ability to manage and organize individual lesson plans.

The left section of the interface houses a toolbox that contains various cyber exercises which were created using the cyber exercise creation feature. These cyber exercises are categorized and color-coded based on their difficulty levels or specific lesson objectives. Instructors can then browse through the toolbox and select the desired cyber exercises which they wish to include in their lesson plans, based on their lesson objectives. By dragging and dropping these exercises into the lesson folders in the central canvas screen, instructors can easily arrange the sequence and structure of the lesson. On the right side of the interface, a properties window provides additional customization options for each lesson plan, such as the lesson duration, difficulty level, lesson name, adding lesson images, or adding specific instructions. This allows for additional customization capabilities to cater to the more diverse learning styles and needs from students.

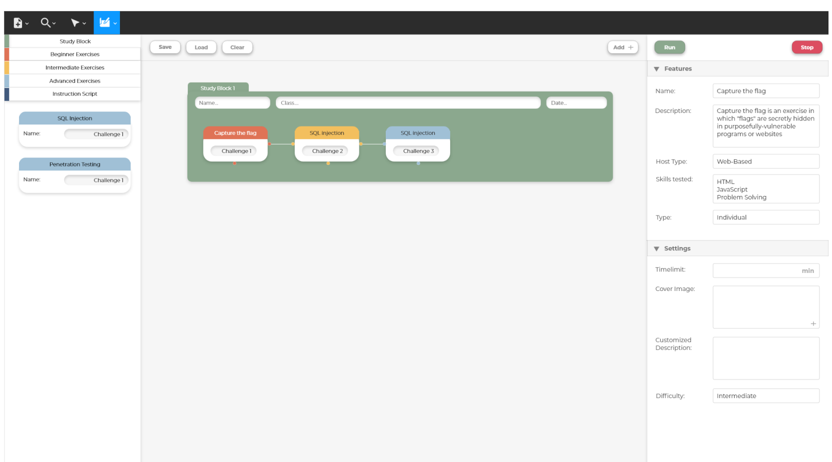

Figure 19: First iteration of lesson planner feature

Building upon the initial design of the lesson planner feature, a second iteration of the lesson planner was designed to incorporate an additional level of management and organization using a calendar and timetable feature. In this updated version, the lesson planner interface still retains its original layout, however, a new section is introduced which displays a timetable view. This feature provides instructors with a visual representation of their course schedule, allowing them to view their entire teaching timeline. This added feature also allows instructors additional management functionalities, where they can make adjustments to the course schedule by dragging and rescheduling lessons to different dates and timeslots.

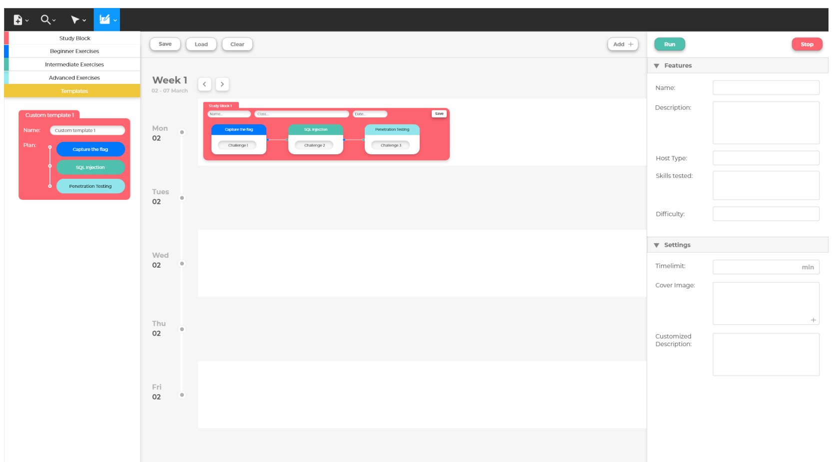

Figure 20: Second iteration of lesson planner feature with added calendar integration

3.6 User Testing

3.6.1 Testing Methodology

In order to evaluate the user experience and usability of each feature of the platform prototype, a few user testing sessions were conducted. These testing sessions employed a mixed-methods approach which was informed by similar case studies and research conducted on the usability of user interface systems [53, 54, 55], combining quantitative and qualitative methods to gather comprehensive data on the system's effectiveness.

The primary objective of the user testing session was to gather feedback from actual high school teachers who teach cybersecurity or IT subjects. Thus, two high school teachers participated in the testing process, one specialized in cybersecurity classes, while the other teaches IT subjects. The user testing findings provide significant insight as they had firsthand experience in teaching these subjects.

Qualitative data was gathered through a think-aloud protocol where participants are encouraged to think aloud and express their opinions on the design, layout, and features of the platform. Furthermore, a questionnaire was administered after the testing session. The questions asked were used to gain feedback on the usability of the system and some insights from their industry knowledge. Some of the questions asked include:

1. Which cyber exercise creation prototype was the easiest and hardest to use?
2. Was there anything about the interface that you were unclear about?
3. Which interface was most visually appealing, would you rather have a different color scheme?
4. After a walkthrough, would you feel confident in using the interfaces to create and modify cyber ranges for students?
5. Did you feel lost at all in any prototype, if yes which?
6. Are there any confusing terminology or terminologies that you would change?
7. What did you think about the lesson planner functionality, is there a better way to develop study plans?
8. Was the lesson plan interface easy to use?
9. Would you feel confident in using the interface to create and modify lessons?

A quantitative Likert scale was also created. The Likert scale consisted of a series of statements that the respondents are asked to evaluate based on a multipoint scale ranging from strongly agree to strongly disagree. Some of the questions that were asked included:

1. The interfaces were easy to navigate and intuitive?
2. The interfaces were visually appealing.
3. The interface is well organized.
4. The colors used in the interface were pleasing.
5. The fonts used in the interface were pleasing.
6. It is easy to navigate through the interface.

7. The interface is easy to remember how to use.
8. The terminologies used are easy to understand.

3.6.2 Testing Process

The testing process was conducted by the author and supervisors, with the author facilitating the testing and the supervisor observing and taking down detailed notes of the participant's responses and comments. The testing process began with a walkthrough of the prototype. The participant was guided through the various screens and features of the platform, while the author acted as a facilitator to explain the purpose and functionality of each feature and element, this step helped the participant to familiarize themselves with the platform's layout, navigation, and features.

Following the walkthrough, the participant was encouraged to engage in a think-aloud protocol, which involved the participant verbalizing their thoughts and opinions on each feature of the platform. This think-aloud protocol was particularly valuable in capturing the participant's ideas and criticisms. At the end of the walkthrough, a discussion took place about additional ways to improve the platform. Moreover, qualitative and quantitative questions were asked to gain additional feedback. Throughout the testing process, the participant's rich insights and perspectives were recorded and noted down to collect comprehensive data collection about the platform.

3.7 Test findings and evaluation

3.7.1 Cyber Exercise Creation Feature

Participants acknowledged that all three cyber exercise creation prototypes namely the block approach, wizard interface, and the hybrid prototype all had their usefulness in their respective contexts. The block-based approach was said to be more favorable for early educators such as primary school teachers as they often rely on visual and graphical teaching methods. It also resembles scratch with they might be familiar with. However, the participants expressed that the code output section, located in the right side of the GUI, would not be helpful for teachers. They mentioned that teachers typically will not require or contain the ability to read and interpret the code provided. Therefore, this section was deemed not useful and relevant to their needs. Furthermore, participants also noted that this approach had its limitations in not providing clear guidance of the next steps they should follow.

While the block-based approach might be more suitable for early educators, on the other hand, the second prototype the wizard approach was said to be more suitable for educators in higher education settings such as high school teachers. This was because high school teachers preferred a more structured approach such as form fillings. Participants appreciated the clear structure and the ability to input information systematically for the wizard interface. Finally, the hybrid prototype, combining both the block-based and structured approaches, received mixed feedback and was perceived to be more confusing compared to the earlier prototypes. They expressed a preference for the more simplistic approach of the previous prototypes.

3.7.2 Feedback of lesson planner feature

The user testing session also included an evaluation of the lesson planner feature, which resulted in a significant discovery. Participants expressed that they had a strong preference for a pre-defined complete experience that could be easily customized rather than having to create each cyber exercise from scratch, as in the previous cyber exercise creation features. This preference was supported by previous research which suggests that most teachers were driven by time constraints and limited resources issues when preparing their lessons [56]. Hence, having access to pre-made materials that can be customized and adapted to their needs would save a lot of time and was highly valued.

The first participant provided a highly valuable insight to how the lesson planner feature should be designed to create a more complete and pre-defined experience. According to their recommendation, the lesson planner feature should be divided into three sections:

1. Lesson Objectives - This section would consist of a bank of lesson aims and objectives that teachers can select for a particular lesson. These objectives will serve as the base and foundation for designing particular lessons and be used to align with desired learning outcomes.

2. Lesson Activities - This section would include a bank of pre-defined and complete cyber exercises that are relevant to the lessons based on the lesson objectives that the teachers selected. These exercises would be categorized and organized based on their alignment with the curriculum and the lesson objectives.

3. Resources - The resources section would provide supporting materials such as website links, worksheet links, helpful videos or any other relevant resources that can help enhance and inform the teacher about the particular lesson objective. These resources will essentially provide supplemental materials for teachers.

Additionally, participants also highlighted the importance of having the platform be highly customizable so that teachers can tailor the lesson plan to the preferences and requirements of individual students, ensuring a more personalized and effective learning experience.

3.7.3 Adding a Student Performance Page

Participants also emphasized the need for an additional student performance feature in the cyber teaching platform. This feature ideally would enable teachers to monitor student performance in their classes, it would be a highly valuable feature as it would provide teachers with real-time analytics on student activities and enable them to track student progress efficiently.

The participants recommended that the student performance feature should offer these key functionalities. Firstly, the feature should allow teachers to keep track of student progress, including tasks completed, and assignments submitted. Additionally, they also suggested incorporating a marking system that indicates whether the student has passed or failed the particular cyber exercise. Furthermore, a comparative element should be included which pitches a student's progress and performance against their peer, as this would encourage them to be competitive using this gamified element. This could be done through a visual representation of a graph or a ranking system that provides students with insights on how they are performing in the classroom.

3.7.4 UI improvements

Overall, the participants expressed satisfaction with the design of the GUI of the cyber teaching platform, stating that it was well-suited for teachers. However, they also suggested a few improvements to enhance the user experience. One of the key recommendations was to consider font size within the interface. Teachers often spend long periods of time working with the platform, and larger fonts can greatly help to improve readability and reduce eye strain. Furthermore, participants also suggested incorporating more colors into the interface. Specifically, the color could be used to color code as a way to differentiate elements and improve the organization and categorization of resources in the platform.

By incorporating these user-driven findings from these user testing sessions, the next prototype can improve on its usability and effectiveness as a teaching tool for teachers. Overall, the user testing sessions provided great and valuable insights that can guide the further development and refinement of this platform, ensuring the alignment of the preferences of educators in this field

.

3.8 Final Iteration

The final prototype iteration of the high-fidelity prototype was developed based on the valuable insights gathered from the user-testing session. Several significant improvements and additions were made with the goal of enhancing the overall user-experience and addressing the limitations identified through testing. It should also be mentioned that many iterations have been developed in between user testing and prototype improvements, and this section describes the final iteration.

3.8.1 Lesson Planner Redesigned

An important insight emerged during the user testing session which showed that teachers preferred focusing on the overall structure and object of their lessons rather than customizing individual exercise details. This finding aligns with research conducted by Beckett et al [56], which emphasizes the importance of focusing on clear lesson objectives and instructional strategies for creating effective lesson plans. Hence, it was decided that an overarching lesson planning feature would be more beneficial than allowing educators to create individual cyber exercises.

The lesson planner prototype was redesigned to simplify the process of selecting cyber exercises that are relevant and aligning with specific lesson objectives that the teachers want to teach. By streamlining the exercise selection, teachers can save time and effort and ensure that the overall lesson plan follows the desired learning outcomes. The lesson planner prototype consists of three main sections, as described by a high school teacher participant in the user testing session: 1. Lesson Objectives, 2. Lesson Activities, 3. Lesson Resources.

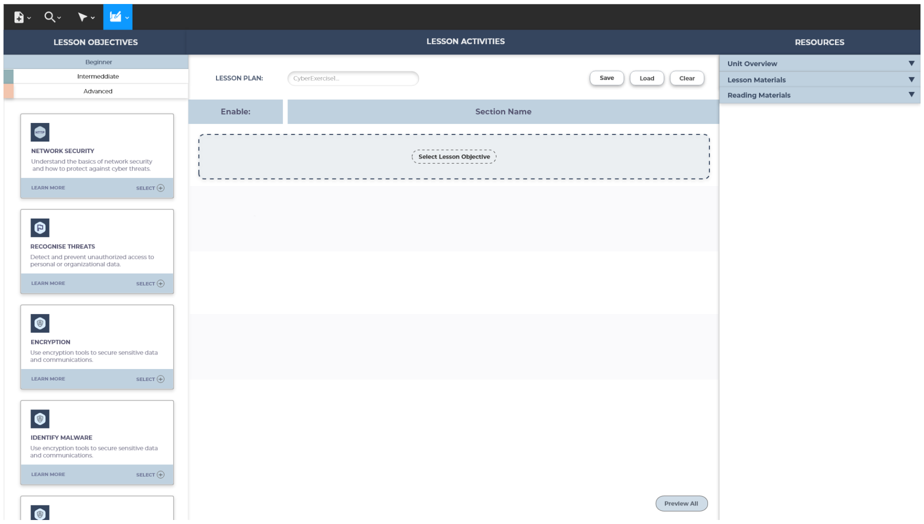

Figure 21: Lesson Planner feature before the teacher selects a learning objective.

1. Lesson Objectives: On the left side of the page, this section offers a bank of predefined aims and objectives that teachers can choose for their lessons. These objectives serve as the base for constructing specific lessons and cyber exercises that align with the objective chosen.

2. Lesson Activities: This section is in the middle of the page. It provides a collection of pre-defined and complete cyber exercises that are relevant to the selected lesson objectives. The cyber exercises are categorized and color-coded based on their difficulty and alignment with the lesson objectives. These exercises can be added, removed, or moved up and down based on the desired sequence in the lesson.

3. Resources: The resources section on the right side of the page provides supplementary materials such as website links, worksheets, and helpful videos to support the teacher's understanding of the specific lesson objective.

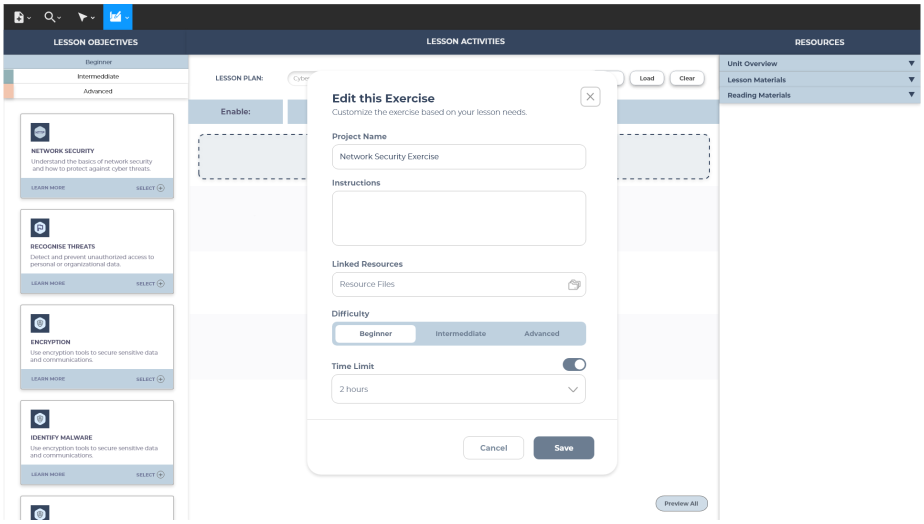

Figure 22: Users fill in parameters and fields to their requirements.

When the teacher accesses the lesson planner page, they are prompted to select a lesson objective. And once the lesson objective is selected, the teacher can then customize the lesson plan to suit their requirements. Lesson names, added instructions, linked files, difficulty, and time limits are some of the parameters that can be edited before a lesson plan is constructed.

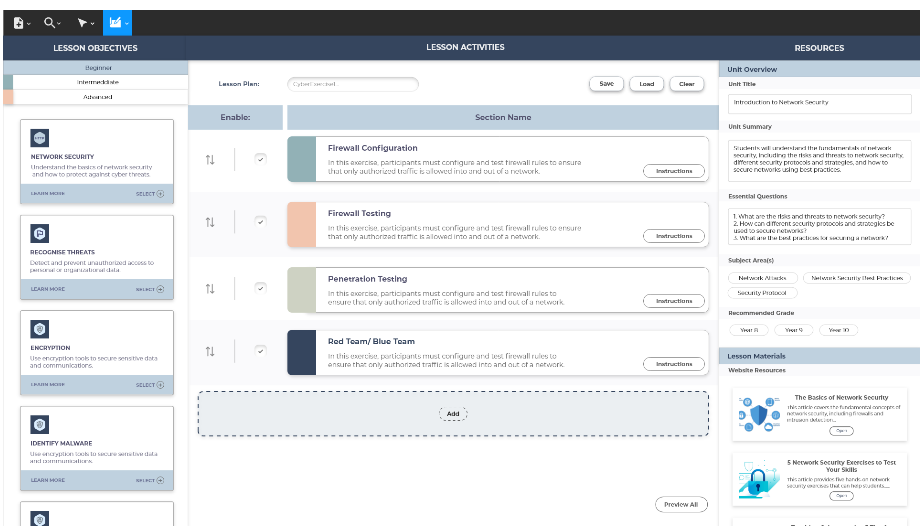

Figure 23: Lesson Planner after selecting and customizing a lesson objective.

After the lesson objective is chosen and configurations are edited, the lesson planner will then provide teachers with a compilation of all the relevant cyber exercises and lesson resources that are related to the objective chosen. For example, if the lesson objective such as "Network Security" is chosen, the platform will then provide cyber exercises such as "Firewall Testing", "Penetration Testing", and "Firewall Configuration" which are relevant exercises to educate students on the principles and concepts of network security. Additionally, the resources section provides the teachers with supporting materials including:

1. Unit Overview - contains the unit title, a unit summary that summarises the overarching unit so that teachers can easily understand, subject areas that the lesson covers, recommended student grade for the difficulty level (Year 9, Year 10...etc), and essential questions that students should aim to answer after each exercise.

2. Lesson Resources - Contains website resources that are relevant to the topic and reading materials. All of these resources support teachers to quickly gain a grasp of the overarching topics to better prepare for the lesson.

Once the cyber exercises are properly set up and arranged, the teacher can then preview the lesson from the student's perspective in their platform. On the left side of the student's user interface, the selected lesson objectives are displayed, providing the students with an overview of the purpose and goals of the lesson. The center of the student's user interface showcases all of the cyber exercises that were set up by the teacher previously, allowing students to access and interact with them. On the right side of the student's user interface are their profile details with information such as exercises completed, and badges collected.

These extra gamification elements adds an extra layer of engagement and motivation for students to actively participate in the learning process. The design of this student's interface prototype also incorporates a lot of cyber-inspired aesthetics, to create a more immersive and visually appealing, gamified experience for students. overall this preview feature aids teachers to make any necessary adjustments to ensure that the lesson plan is effective and engaging for their students.

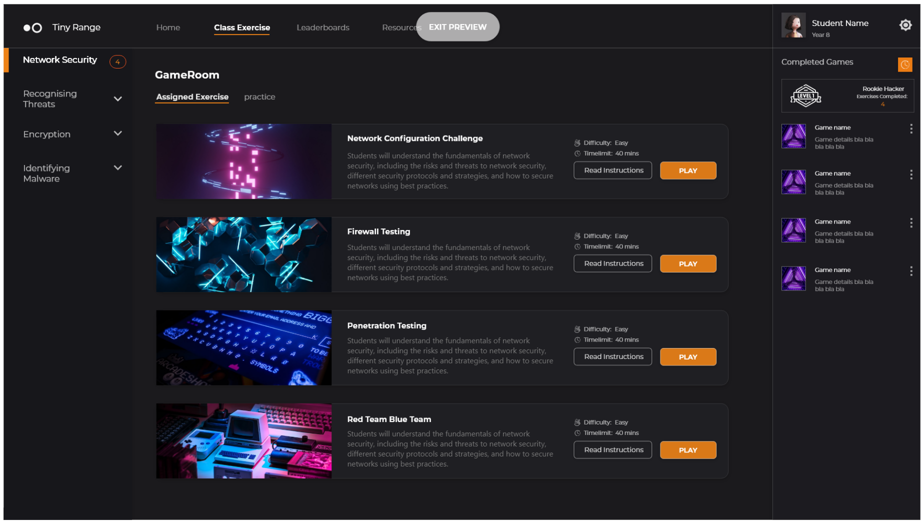

Figure 24: Preview of the lesson from the student's platform.

3.8.2 Dashboard implementation

In addition to the lesson plan feature, a navigable dashboard was implemented to provide a centralized hub for instructors to access all key features and functionalities of the e-learning platform. This interface allows for navigation and efficient management of various components.

The left section of the dashboard features the instructor's profile and provides quick access to all of the important features such as the lesson planner and student performance sections. Next, the lesson plan section in the center of the dashboard serves as a repository for all the complete lessons created by the teacher. It displays the status of each lesson indicating whether it is completed or in progress, helping instructors keep track of their progress in creating lessons.

To further assist instructors in managing their classes, a timetable schedule feature that allows teachers to organize their classes and allocate specific dates and times for each lesson is provided, aiding teachers with efficient lesson planning. Moreover, on the right side of the dashboard is an overarching resources section that provides access to all materials related to the lesson objectives. Finally in the bottom right of the dashboard is a student performance section which provides a summarised version of student performance. It is also linked to the more extensive student performance feature.

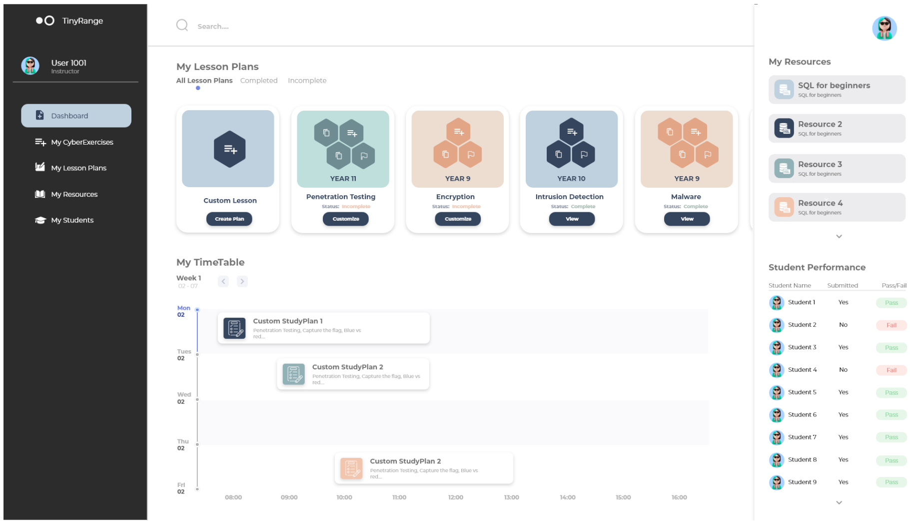

Figure 25: Users fill in parameters and fields to their requirements.

3.8.3 Addition of student Performance page

Based on the valuable feedback provided by participants during the user testing phase, a new student performance page was introduced. This section serves as a tool for teachers to track student performance effectively throughout their lessons.

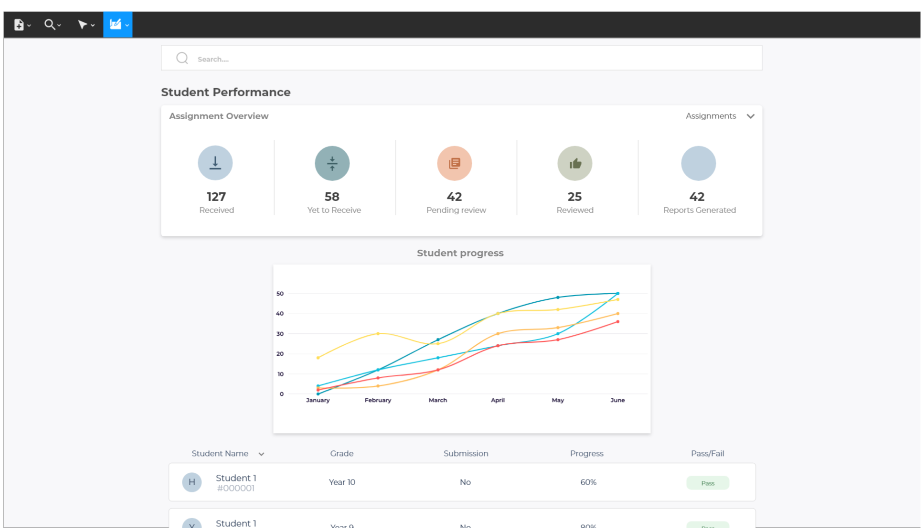

Figure 26: Student’s performance section

In this prototype, the assignment overview section allows teachers to easily view and manage student submissions, they can also quickly identify which exercises have been received from students and track the status of exercise reviews to ensure that they stay informed of student progress in lessons. Additionally, the prototype incorporates a database of student information. This database includes student names, grades, submission status, and student progression in lessons. Furthermore, a key feature of the student performance prototype is the student progress graph. This graph visually represents the progress of each student in comparison to the rest of the cohort. This may help teachers identify students who may need additional support and also foster a healthy sense of competition among students to motivate them.

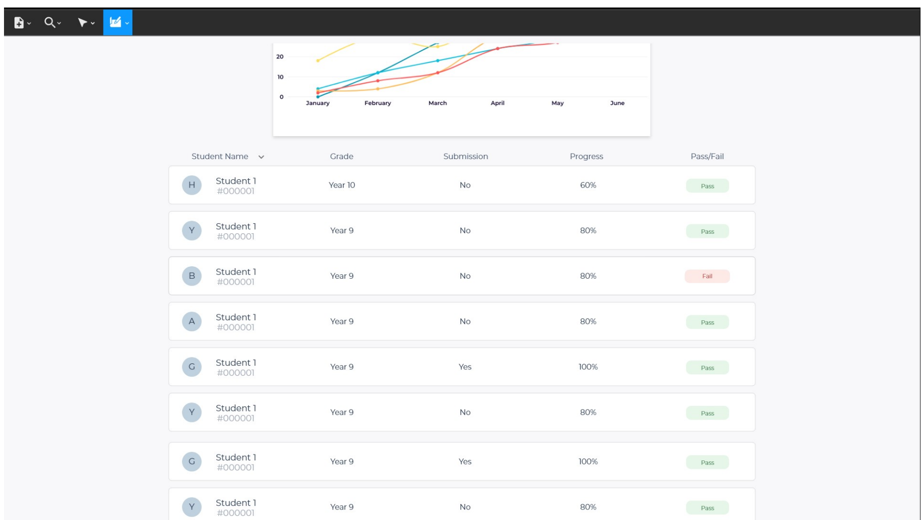

Figure 27: Student's Database

4 Conclusion

4.1 Review of Performance and Acheivements

In reviewing the performance and achievement against the initially stated plan and goals, I believe that this project has made great progress toward designing and building a user-friendly education platform for non-technical educators. The research questions to be answered were initially stated as follows:

1. What are the key design features that should be included in a user-friendly cyber-teaching platform for non-technical educators?

2. What is the optimal approach to user testing and prototyping in the development of a user-friendly cyber-teaching platform for non-technical educators?

3. How can a user-friendly cyber-teaching platform be adapted to meet the needs of different types of learners and diverse student populations?

The initial research questions were centered around addressing the lack of literature on user interfaces for cybersecurity teaching platforms and the need to make cybersecurity education accessible to non-experts, and this project has provided satisfactory solutions to all of the research questions.

Firstly, the key design features needed for the education platform were identified through extensive literature review and user-testing, and interviews with industry professionals - high school teachers in their respective IT fields. These design features include:

- A comprehensive lesson planner with customizable complete, predefined resources and provided cyber exercises.
- A preview of the student's platform after lessons are set up by the teacher.
- A student performance feature that helps teachers track all of their student's progress.
- A dashboard with clear navigation and organization of resources that provides easy access to key features.
- A cyber exercise creation feature which allows teachers to set up and manage individual cyber exercises to their classroom requirements.

By identifying and incorporating these key design features, the platform becomes more accessible and user-friendly for non-technical educators to set up learning environments, achieving the initial goal of the project. Furthermore, an optimal approach to user testing and prototyping was identified through the project using a mixed-methods approach combining quantitative and qualitative methods using informed case studies [53, 54, 55] done similar preexisting user-tests for system user interfaces. Including think-aloud protocols, questionnaires and interviews to gather important insights.

Finally, to address the challenge of adapting the platform to meet the needs of different types of students and learners, the platform was designed with flexibility and customization options in mind. The ability of teachers to customize resources and lessons allowed educators to tailor the lessons to the specific requirements of different student populations. Features and customizable options such as varying difficulty levels of cyber exercises, adaptable lesson planning tools, and multiple learning pathways all support different learning styles to provide and inclusive cyber-teaching experience for students.

4.2 Limitations

This project has encountered several limitations that may have influenced the project scope and outcomes. Some of the limitations encountered include:

- **Time Constraints:** This project was conducted within a limited timeframe, which restricted the depth of explorations in some aspects, certain features or functionalities could be explored more to enhance user experience. Furthermore, timing constraints did not allow for the full implementation of improvements based from user feedback.

- **Limited User Test Sample:** The lack of variation and number of participants limited the user test findings. The participants are all from a highschool teaching background, hence most of the feedback gained would be more suitable for a higher education teachers platform, disregarding some of the earlier educations such as primary schools. Thus, the limited sample may not fully represent the diverse range of potential users or educational contexts.

- **Scope of Expertise:** The project primarily focused on the design and prototyping of the user interface for the platform. As such, it did not delve extensively into the creation of more advanced cybersecurity content or address some of the more technical aspects of cyber technologies such as cyber ranges and virtual machines.

- **Scalability:** This project focused on a specific front-end system and as such did not further explore the scalablity of the platform to accommodate a larger number of users or handle complex educational scenarios such as large student databases. Hence, further development and testing would be required to improve and assess the platform's scalability in real-life educational settings.

- **Lack of research for the student's perspective:** Another limitation of this project is the limited focus on the student's perspective and platform as the scope was mainly focused on the educator's platform. Hence, this limitation may result in potential gaps in addressing student engagement, learning outcomes, and overall satisfaction in lessons.

4.3 Future Work and Improvements

The limitations and outcomes of the presented project offer several opportunities for potential improvements that could enhance the cyber-teaching platform given a larger time frame and more resources.

- **More In-depth user testing and research:** In the future, conducting more user research, including surveys, and user testing would provide more valuable insights into educator's specific needs and preferences. Furthermore, user-testing could be done with teachers who teach at varying levels of education, from primary school teachers up to university teachers to gain a more comprehensive overview of their preferences.
- **Research on Student's Experience:** The project could expand its focus onto the student's perspective to identify features that could increase student engagement, interactivity and personalized learning.
- **Integration with Learning Management Systems (LMS):** The project could also explore the integration of the platform with existing Learning Management Systems to provide a more seamless access and integration of course materials, assignments, grading and communication tools.
- **Addition of Collaboration and Social Learning Features:** Further features that enable collaboration and social learning such as discussion forums, and peer-to-peer feedback could add an extra level of collaboration between the teacher and the student to promote knowledge sharing and active engagement.
- **Enhanced Analytics:** Designing more advanced analytics features and reporting capabilities could provide teachers with more actionable insights into student performance, and progress and provide an understanding of areas of improvement. This would enable instructors to make data-driven decisions to tailor their teaching strategies and provide timely feedback and support to their students.

4.4 Summary of Outcomes

In conclusion, this thesis project started out with an aim to design and implement a functional prototype of a user-friendly front-end system for non-technical educators to set up cyber-learning environments. And, through an iterative design process that cycled through the processes of testing, evaluation, analysis, and prototype iterations, ended up with a final iteration of the platform focused on a user-centered approach.

The design process began with comprehensive user research to understand teachers' needs to identify key features to implement in the platform. Continuing with generating design ideas, through brainstorming with early sketches and low-fidelity prototypes, these early ideas are converted into high-fidelity prototypes that were ready for user testing. Once the prototypes were created, they were tested with actual high school teachers to gain insight into targetted users, and these findings were then evaluated and used to iterate and refine the prototype. This process was then repeated until the final iteration was obtained.

The project addressed all of the research questions and goals that were stated from the start. The key design features of the cyber-teaching platform were identified and implemented. These features include a comprehensive lesson planner, a student performance tracking feature, a navigable dashboard, a cyber exercise creation feature, and a student platform preview feature. User testing and feedback played a crucial role in refining and improving the platform's design. Participants provided valuable insights and recommendations which were considered and implemented to suit teachers' needs. During the user testing session, the main insight was how teachers preferred complete, pre-defined experiences that could be edited to suit their requirements, instead of having to create resources from scratch. Thus, the key features of the platform were designed with customizability in mind. These flexible features provide educators with the tools to create engaging and more customized learning experiences for diverse student population.

Despite the project's completion, there are still several limitations that suggest that the prototype can still be improved much further, as the time frame and available resources restricted the scope and depth of the project. Furthermore, the lack of diverse user testing participants of educators from various educational levels would restrict the test findings as it would be more suitable for high-school teachers who were the main subject group tested. Additionally, the lack of research on the student's perspective is another notable limitation that would be beneficial for further exploration in this area.

Looking ahead, there seem to also be many opportunities for further improvements and in-depth research. Some of the possible opportunities would be to enhance the student experience through extensive research on student engagement and learning experiences. Furthermore, features such as integration with

Learning Management Systems, collaboration features, advanced analytics, and scalability are some of the areas that could be explored in a larger time frame.

In conclusion, this theses project addressed the gap of the lack of resources catering to non-expert educators in setting up learning environments, providing them with a user-friendly platform to create engaging and effective learning environments. Finally, by addressing the limitations and pursuing further improvements, the platform, Tiny Range has the potential to become a valuable tool in enhancing cybersecurity education and meeting the evolving needs of educators and students in this field.

References

[1] C. Ed, "Cybersecurity curricular guideline." http://dx.doi.org/10.1145/3368308.3415364, 2017.

[2] Z. M. Smith, E. Lostri, and J. A. Lewis, "The hidden costs of cybercrime. mcafee." https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-ofcybercrime.pdf, 2020.

[3] S. S. Oh, N. Stickney, D. Hawthorne, and D. Matthews, "Teaching web-attacks on a raspberry pi cyber range." http://dx.doi.org/10.1145/3368308.3415364, 2020. Accessed: 29/5/2023.

[4] J. Vykopal, P. Celeda, P. Seda, V. Svabensky, and D. Tovarnak, "Scalable learning environments for teaching cybersecurity hands-on," in *IEEE Frontiers in Education Conference (FIE)*, 2021.

[5] A. Security, "Options for setting up a cyber range." https://www.ariessecurity.com/options-for-setting-up-a-cyber-range-build-vs-buy, 2021.

[6] C. L. Beauchamp, "Exploring cyber ranges in cybersecurity education." https://vtechworks.lib.vt.edu/handle/10919/109524, 2022.

[7] R. Guiler, "Building a cyber training range on a budget." https://www.youtube.com/watch?v=NQaVqN1HFPs, 2018.

[8] J. Bottazzi, P. Falcarin, and F. Martinelli, "A comprehensive study on virtualization in cybersecurity education," *IEEE Transactions on Education*, vol. 62, no. 1, pp. 13–20, 2019.

[9] F. B. Schneider, "Cybersecurity education in universities," *IEEE Security Privacy*, vol. 11, no. 4, pp. 3–4, 2013.

[10] C. Franklin, "7 university-connected cyber ranges to know now." https://www.darkreading.com/cloud/7-university-connected-cyberranges-to-know-now/d/d-id/1331224, 2017.

[11] V. Giuliano and V. Formicola, "Icsrange: A simulation-based cyber range platform for industrial control systems." https://doi.org/10.48550/arXiv.1909.01910, 2019.

[12] B. Bhattacharya, J. A. Clark, A. Ghosal, and R. K. Shyamasundar, "Cyber-ranges: An overview and research perspectives," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (Udupi), pp. 2143–2147, 2017.

[13] G. Dhillon and C. K. Lim, "Utilizing virtual machines in cyber security education: An empirical study," *Computers & Education*, vol. 107, pp. 54–67, 2017.

[14] S. T. Hamman, K. M. Hopkinson, W. Patterson, and C. University, "Teaching adversarial thinking for cybersecurity." `https://f.hubspotusercontent40.net/hubfs/5627441/PDFs/Adversarial`, 2021.

[15] P. Point, "What is a sandbox." `https://www.proofpoint.com/au/threat-reference/sandbox`, 2018.

[16] C. Lang, G. Siemens, A. Wise, and D. Gašević, eds., *Handbook of Learning Analytics.* 1st ed., 2017.

[17] C. Romero, S. Ventura, M. Pechenizkiy, and R. S. Baker, eds., *Handbook of Educational Data Mining.* 2010.

[18] C. Hundhausen, D. Olivares, and A. Carter, "Ide-based learning analytics for computing education: A process model, critical review, and research agenda," 2017.

[19] K. Sanders, J. Boustedt, A. Eckerdal, R. McCartney, and C. Zander, "Folk pedagogy: Nobody doesn't like active learning," 2017.

[20] E. Ukwandu, M. Farah, H. Hindy, D. Brosset, D. Kavallieros, R. Atkinson, C. Tachtatzis, M. Bures, I. Andonovic, and X. Bellekens, "A review of cyber-ranges and test-beds: Current and future trends," *Sensors*, vol. 20, p. 7148, 2020.

[21] P. Toth and P. Klein, "A role-based model for federal information technology/cyber security training," tech. rep., 2013.

[22] S. M. Glumich and B. A. Kropa, "Defex: Hands-on cyber defense exercise for undergraduate students," 2011.

[23] J. Creswell, *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research.* 2015.

[24] S. Cheng, L. Lu, K. Xie, and V. W. Vongkulluksn, "Understanding teacher technology integration from expectancy-value perspectives," *Teaching and Teacher Education*, vol. 91, 2020.

[25] S. Hidi and K. A. Renninger, "The four-phase model of interest development," *Educational Psychologist*, vol. 41, no. 2, pp. 111–127, 2006.

[26] C. Finelli, K. Richardson, and S. Daly, "Factors that influence faculty motivation of effective teaching practices in engineering," in *2013 ASEE Annual Conference & Exposition Proceedings*, (Atlanta, GA), pp. 23.590.1–23.590.11, 2013.

[27] H. M. G. Watt, P. W. Richardson, and K. Smith, *Global perspectives on teacher motivation.* Cambridge University Press, 2017.

[28] S. S. Oh, N. Hawthorne, D. Matthews, and Suzanne, "Teaching web-attacks on a raspberry pi cyber range," 2020.

[29] S. Dasgupta and B. M. Hill, "Scratch community blocks: Supporting children as data scientists," 2017.

[30] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Programming for all," 2009.

[31] R. Fling, *Designing Effective Speech Interfaces.* O'Reilly Media, 2009.

[32] J. T. McDonnell and J. J. McDonnell, "An experimental comparison of diagramming techniques for uml," *IEEE Software*, vol. 14, no. 2, pp. 59–67, 1997.

[33] N. Hall, "Node-red: An open-source tool for the internet of things," *IT Professional*, vol. 18, no. 4, pp. 27–33, 2016.

[34] O. Drögehorn *et al.*, "Visual scripting for smart home iot applications with node-red," in *Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pp. 643–650, 2020.

[35] B. Faghih, M. Azadehfar, and S. Katebi, "User interface design for e-learning software," 2014.

[36] *Integrating Design Features for E-Learning Platforms*, vol. 82, 2022.

[37] S. A. Salloum, A. Q. Alhamad, M. Al-Emran, A. A. Monem, and K. Shaalan, "Exploring students' acceptance of e-learning through the development of a comprehensive technology acceptance model," *IEEE Access*, vol. 7, pp. 128445–128462, 2019.

[38] A. S. Al-Ajlan, *A comparative study between e-learning features.* London, UK: IntechOpen, 2012.

[39] R. Garrote Jurado, T. Pettersson, A. Regueiro Gomez, and M. Scheja, "Classification of the features in learning management systems," in *Proceedings of the XVII Scientific Convention on Engineering and Architecture*, (Havana City, Cuba), pp. 24–28, November 2014.

[40] K. Pireva, F. Nuqi, and E. Hajrizi, "The impact of lmss in learners'," in *Proceedings of the Quality in Blended Learning Conference*, (Wiener Neustadt, Austria), February 2014.

[41] J. Rubin and D. Chisnell, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests.* Wiley, 2008.

[42] T. Tullis and W. Albert, *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics: Second Edition.* 2008.

[43] J. Nielsen, "Iterative user-interface design," *Computer*, vol. 26, no. 11, pp. 32–41, 1993.

[44] C. Kelleher and R. Pausch, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers," *ACM Computing Surveys (CSUR)*, vol. 37, no. 2, pp. 83–137, 2005.

[45] M. Resnick and B. Silverman, "Some reflections on designing construction kits for kids," in *Constructionism in practice: Designing, thinking, and learning in a digital world*, pp. 143–164, Routledge, 2005.

[46] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The scratch programming language and environment," *ACM Transactions on Computing Education (TOCE)*, vol. 10, no. 4, pp. 1–15, 2010.

[47] "Blockly." `https://developers.google.com/blockly`. Accessed: 29/5/2023.

[48] "Scratch." `https://scratch.mit.edu`. Accessed: 29/5/2023.

[49] Z. Yan and K. Zhang, "A visual programming tool for user interface and web page generation," in *Proceedings Technology of Object-Oriented Languages. TOOLS 27 (Cat. No.98EX224)*, (Beijing, China), pp. 181–185, 1998.

[50] J. Nielsen, "Enhancing the explanatory power of usability heuristics," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 152–158, 1994.

[51] Y. H. Wong and H. Peng, "Wizard-based interface design and authoring tool for creating learning interfaces," *Interactive Learning Environments*, vol. 25, no. 8, pp. 1007–1023, 2017.

[52] E. Berndt and A. Braun, "The role of wizards for easing the use of software applications: An empirical study," *International Journal of Human-Computer Studies*, vol. 67, no. 11, pp. 993–1003, 2009.

[53] C. H. Lee, W. J. Yang, and C. Y. Huang, "Usability testing of learning management systems: A case study," *Journal Name*, 2020.

[54] S. H. Liu and Y. T. Huang, "Usability evaluation of learning management systems: A case study," *Journal Name*, 2008.

[55] R. B. German, K. L. Maloney, and K. M. Coleman, "User-centered design and evaluation of a cybersecurity training program for employees," *Journal Name*, 2004.

[56] D. Beckett, P. Hager, and H. Cahill, *Learning to Teach in the Primary School*. Routledge, 2017.

5 Appendix A

This appendix consists of all of the user-testing questions that are used. These questions include both qualitative (open-ended questions) and quantitative questions (Likert Scale).

Quantitative Analysis: Likert Scale

Please indicate your level of agreement with the following statements. where **1 = Strongly Disagree** and **5 = Strongly Agree**:

(1) The interfaces were easy to navigate and intuitive?

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(2) The interfaces were visually appealing

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(3) The interface is well organized

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(4) The colors used in the interface were pleasing

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(5) The fonts used in the interface were pleasing

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(6) It is easy to navigate through the interface

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(7) The interface is easy to remember how to use

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(8) The terminologies used are easy to understand

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Figure 28: Quantitative likert scale form utilized in the user testing.

4.7 Qualitative Questions - Open Ended Questions (First Iteration)

1. Which cyber exercise creation prototype was the easiest and hardest to use?

2. Was there anything about the interface that you were unclear about?

3. Which interface was most visually appealing, would you rather have a different color scheme?

4. After a walkthrough, would you feel confident in using the interfaces to create and modify cyber ranges for students?

5. Did you feel lost at all in any prototype, if yes which?

6. Are there any confusing terminology or terminologies that you would change?

7. What did you think about the lesson planner functionality, is there a better way to develop study plans?

8. Was the lesson plan interface easy to use?

9. Would you feel confident in using the interface to create and modify lessons?

Figure 29: Quantitative likert scale form utilized in the user testing.

Quantitative Analysis: Likert Scale

Please indicate your level of agreement with the following statements. where **1 = Strongly Disagree** and **5 = Strongly Agree**:

(1) The interfaces were easy to navigate and intuitive?

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(2) The interfaces were visually appealing

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(3) The interface is well organized

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(4) The colors used in the interface were pleasing

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(5) The fonts used in the interface were pleasing

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(6) It is easy to navigate through the interface

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(7) The interface is easy to remember how to use

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

(8) The terminologies used are easy to understand

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Figure 30: Quantitative likert scale form utilized in the user testing.

6 Appendix B

This appendix consists of the iteration prototypes, Note that not all of the slides between interactions are included.

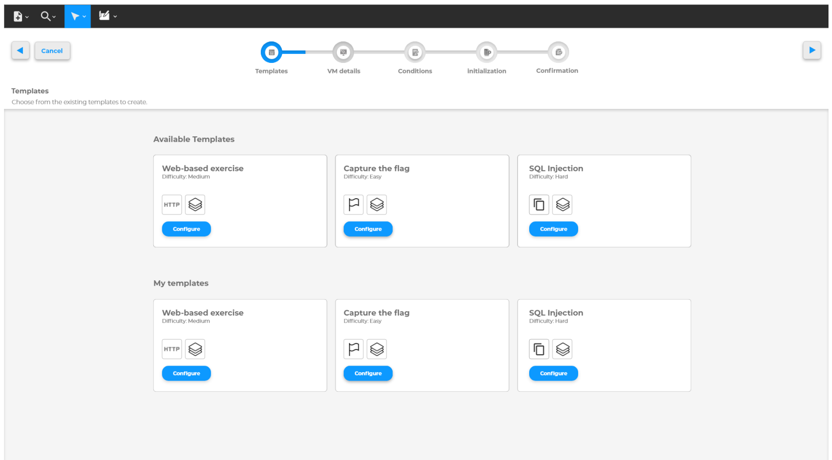

Figure 31: Final Prototype: Wizard Interface - 1

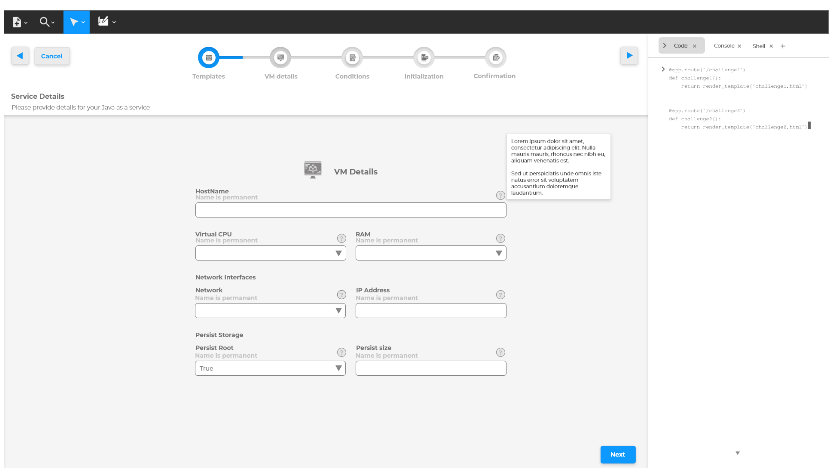

Figure 32: Final Prototype: Wizard Interface - 2

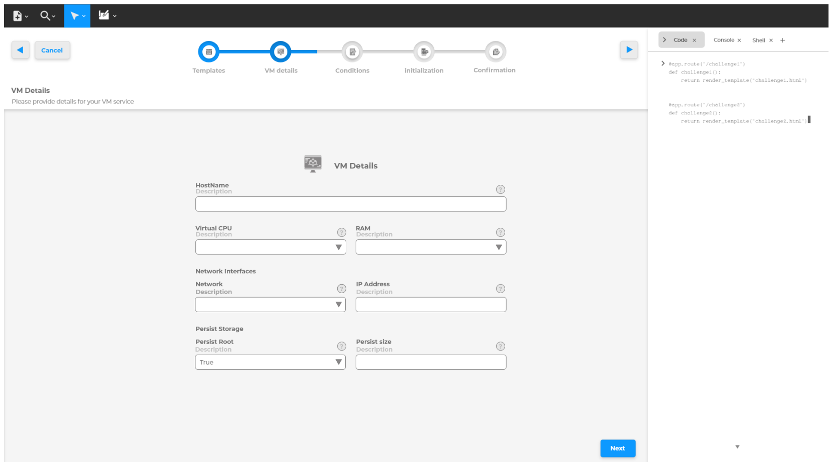

Figure 33: Final Prototype: Wizard Interface - 3

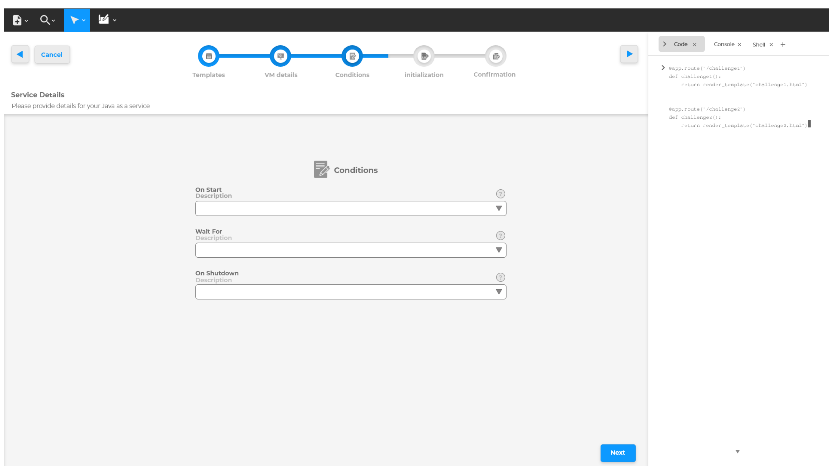

Figure 34: Final Prototype: Wizard Interface - 4

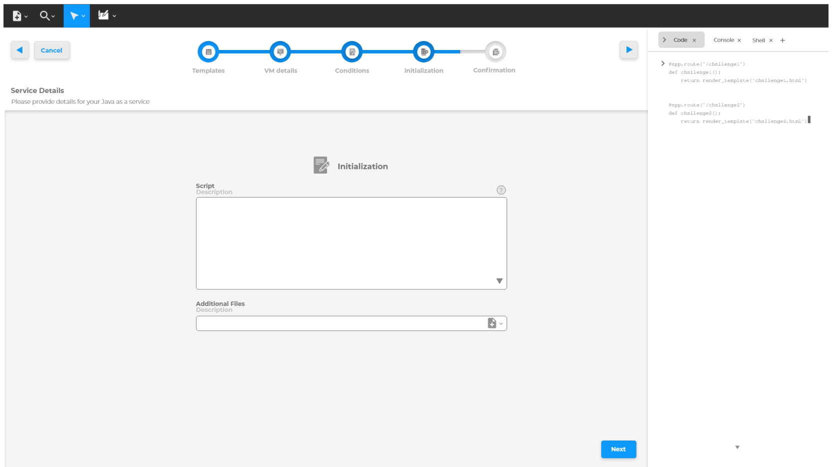

Figure 35: Final Prototype: Wizard Interface - 5

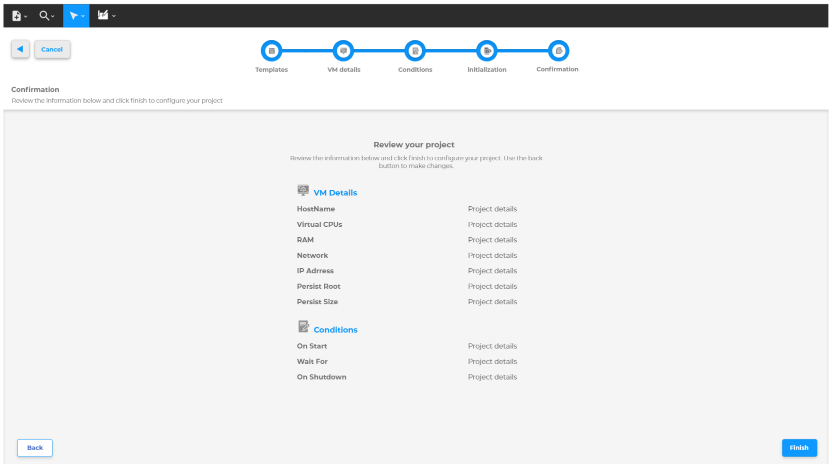

Figure 36: Final Prototype: Wizard Interface - 6

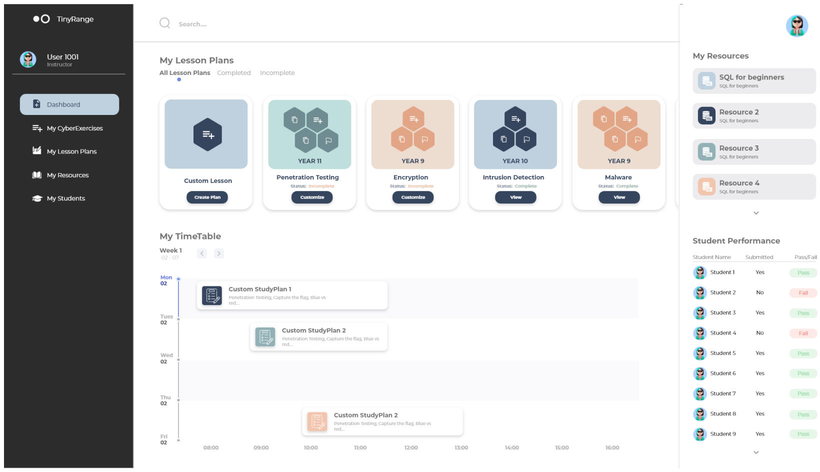

Figure 37: Final Prototype: Dashboard

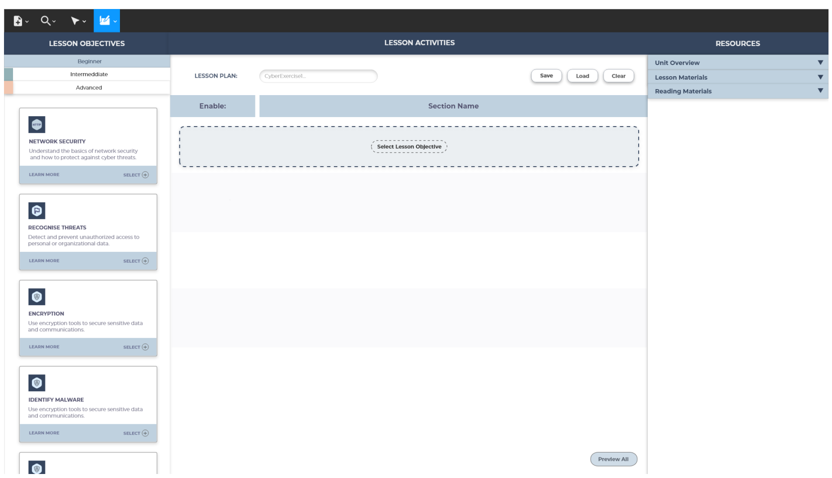

Figure 38: Final Prototype: Lesson Planner - 1

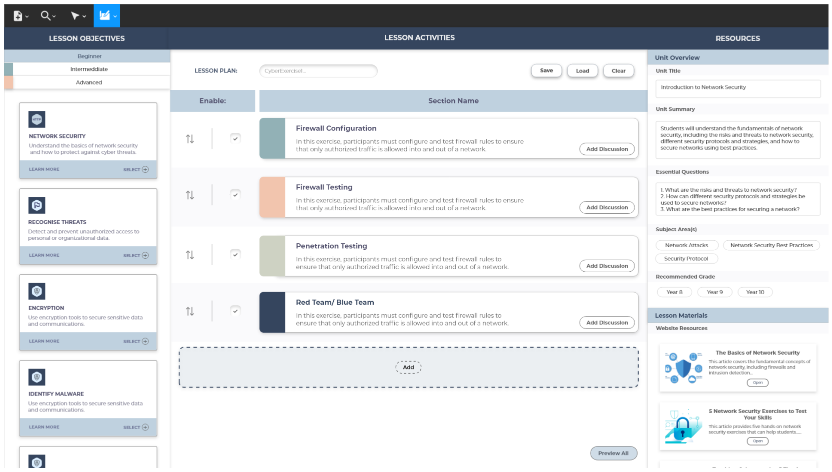

Figure 39: Final Prototype: Lesson Planner - 2

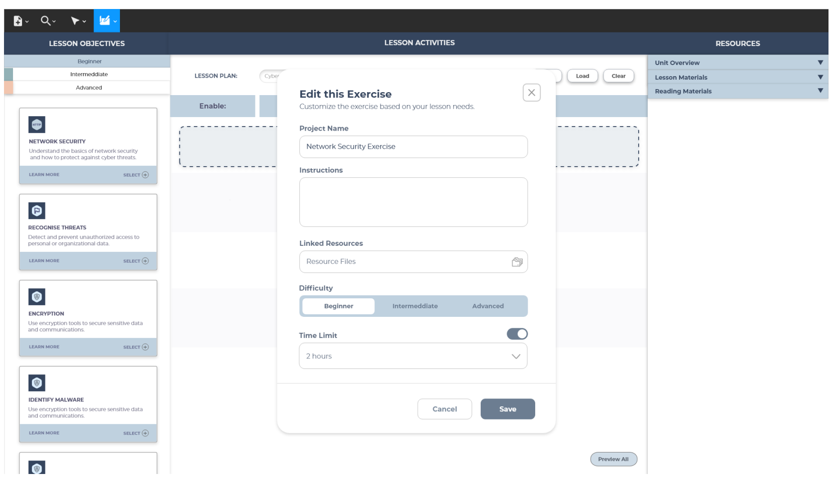

Figure 40: Final Prototype: Lesson Planner - 3

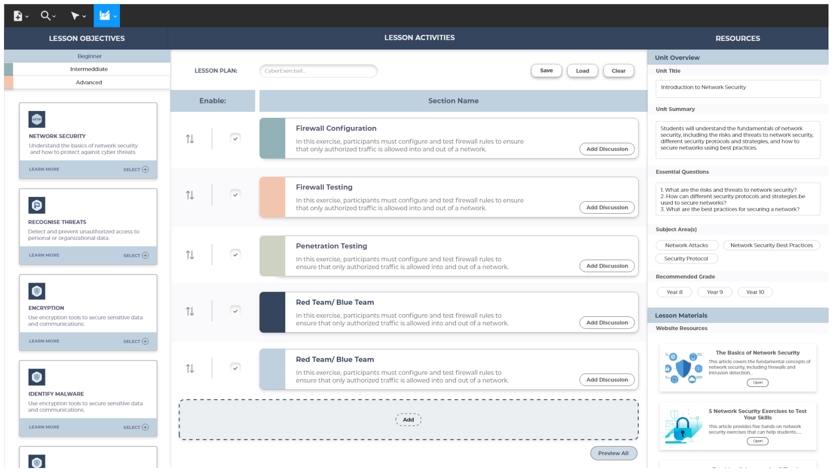

Figure 41: Final Prototype: Lesson Planner - 4

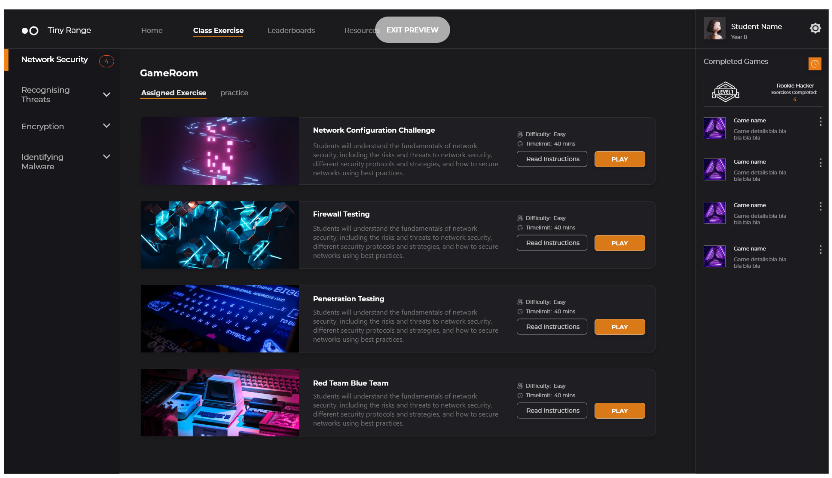

Figure 42: Final Prototype: Student Preview

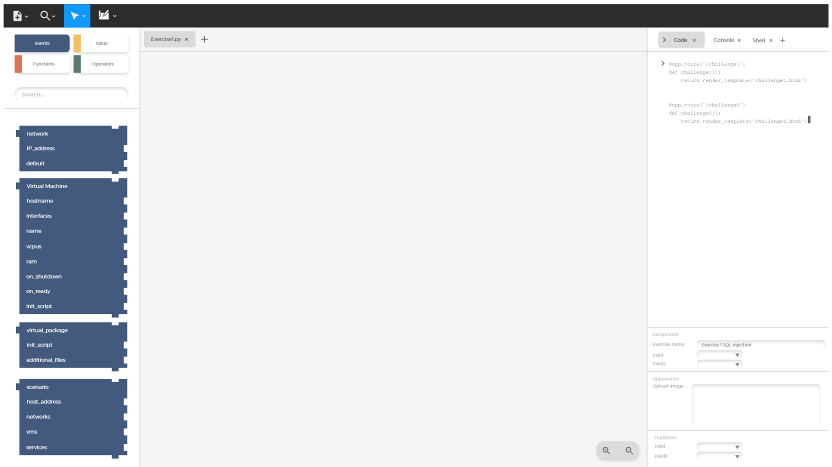

Figure 43: First Iteration: Block Approach - 1

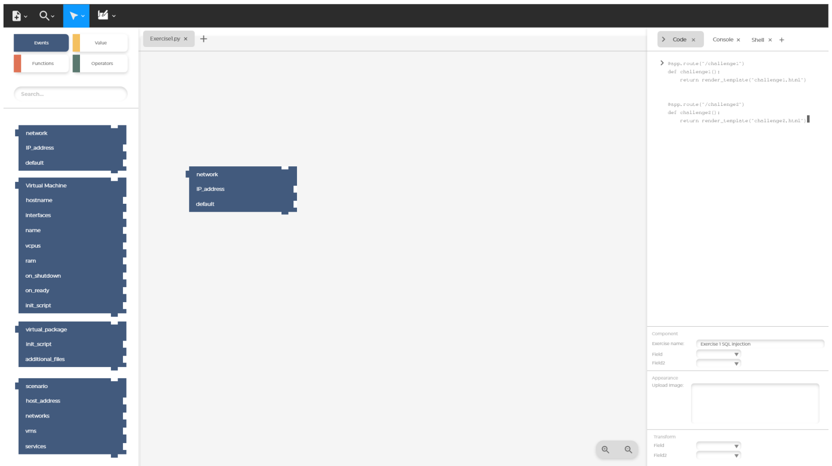

Figure 44: First Iteration: Block Approach - 2

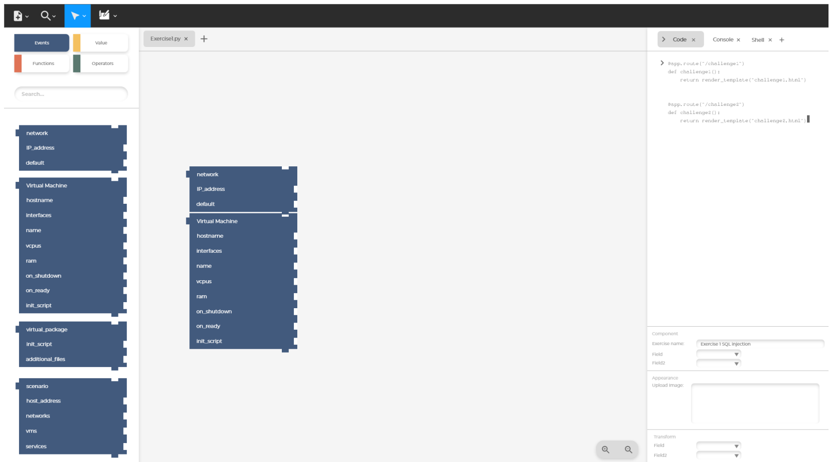

Figure 45: First Iteration: Block Approach - 3

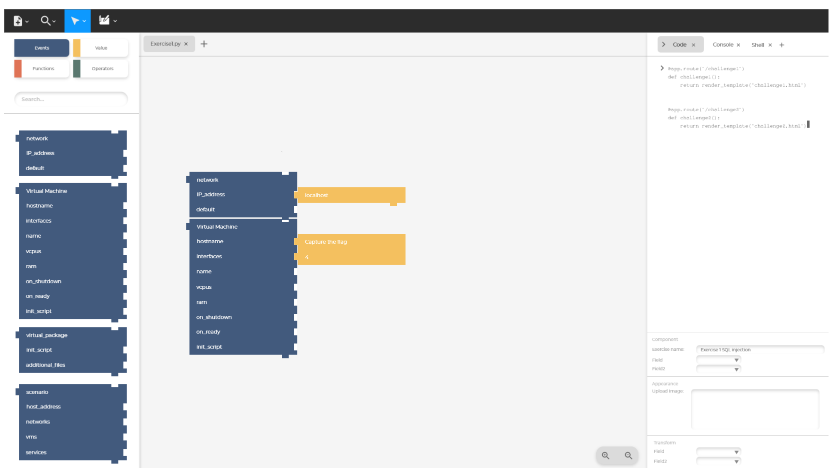

Figure 46: First Iteration: Block Approach - 4

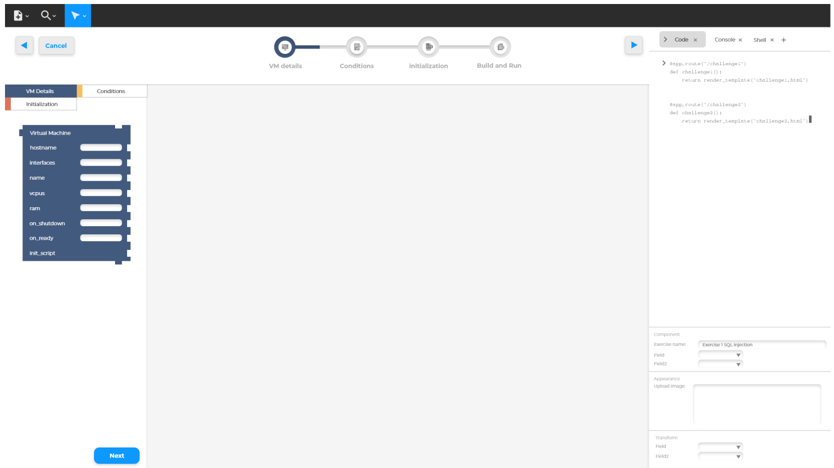

Figure 47: Early Iteration: Hybrid Approach - 1

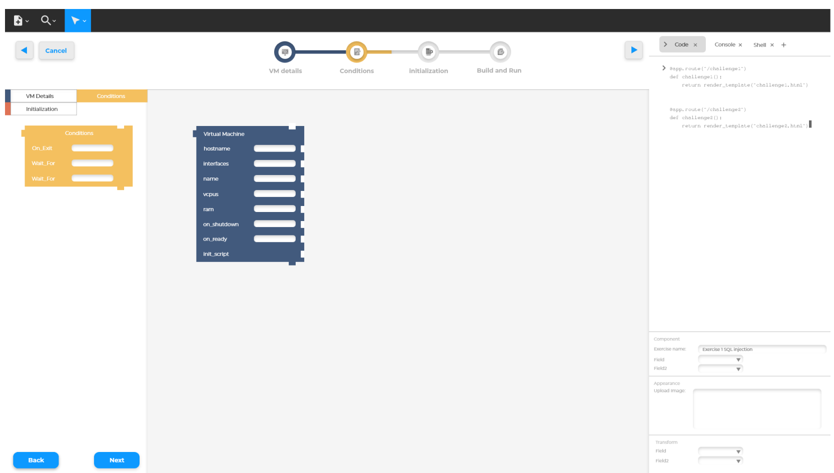

Figure 48: Early Iteration: Hybrid Approach - 2

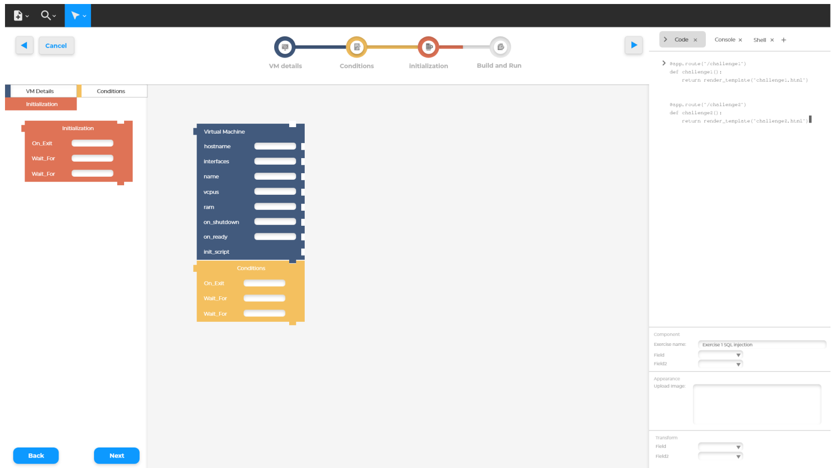

Figure 49: Early Iteration: Hybrid Approach - 3

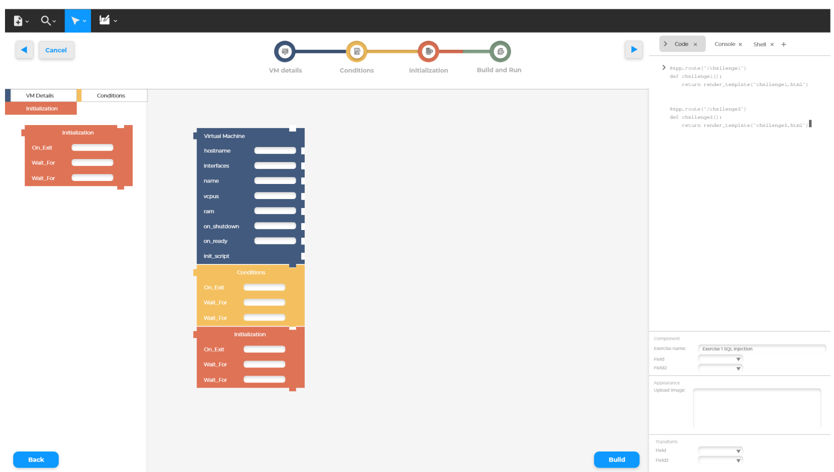

Figure 50: Early Iteration: Hybrid Approach - 4

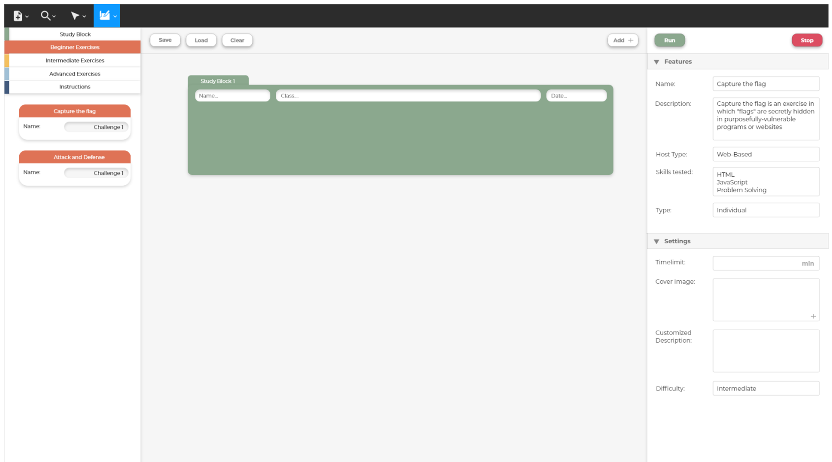

Figure 51: Early Iteration: Lesson Plan - 1

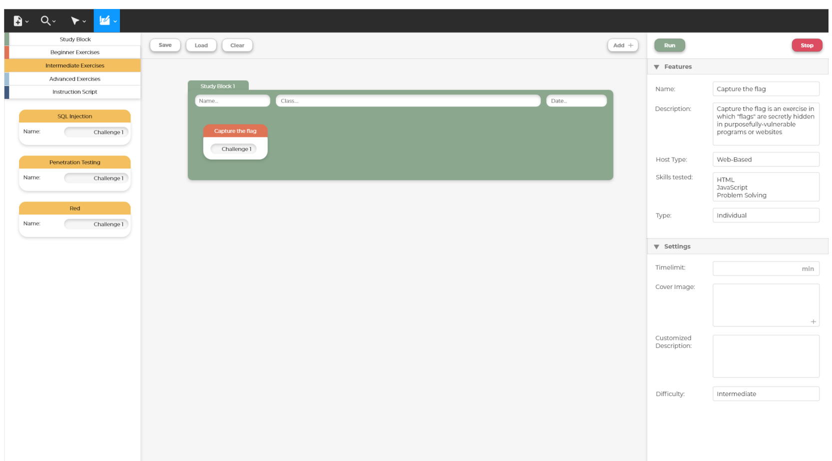

Figure 52: Early Iteration: Lesson Plan - 2

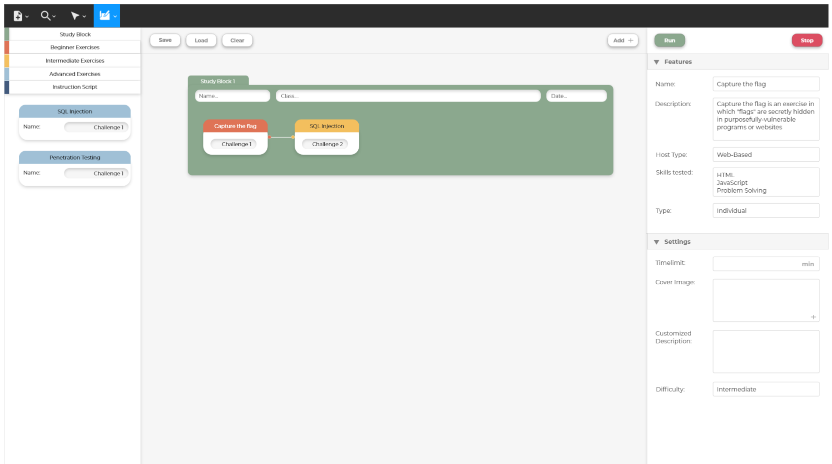

Figure 53: Early Iteration: Lesson Plan - 3

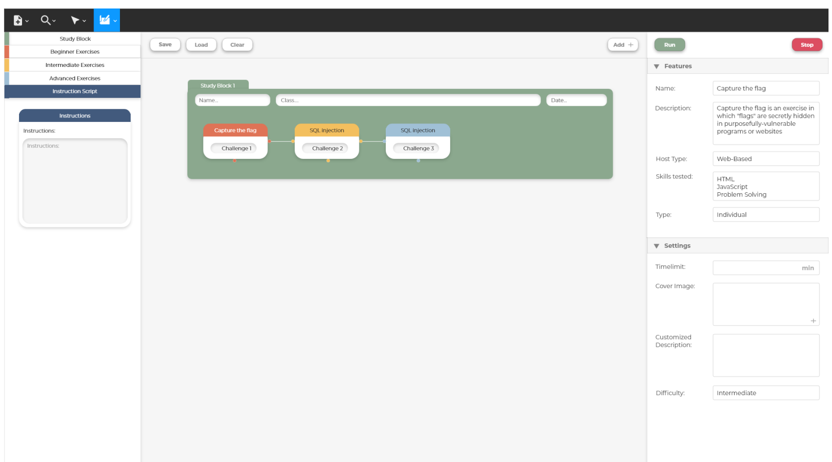

Figure 54: Early Iteration: Lesson Plan - 4

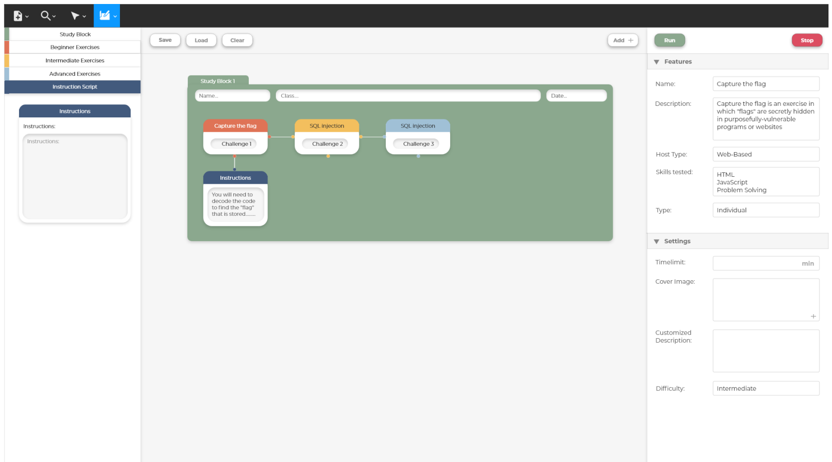

Figure 55: Early Iteration: Lesson Plan - 5