Fall 2018

Mazetec: A Scenario-Based Learning Platform

Daniel Bietz

MAZETEC: A SCENARIO-BASED LEARNING PLATFORM

by

DANIEL E. BIETZ

(Under the Direction of Andrew Allen)

ABSTRACT

This work presents Mazetec, a scenario-based learning platform for delivering non-linear scenarios format asynchronously. It enables subject matter experts to create interactive, state-dependent case studies or courses with branching logic for online learning and knowledge testing. Mazetec is a complex web application designed to deliver decision-based or case-based educational scenarios and simulations in a time-limited, non-linear format. There are many e-learning systems in the open source and commercial markets, but while these systems may have similar functions, we have found none that are both domain independent and able to deliver state-dependent content asynchronous and non-linearly. Mazetec can serve as a standalone training system or serve as a supplementary activity provider to an existing course in an organization's existing learning management system (LMS).

INDEX WORDS: Scenario-based learning system, Serious game, xAPI, Learning management system, Simulations, Case-based reasoning, Exploratory learning environment

MAZETEC: A SCENARIO-BASED LEARNING PLATFORM

by

DANIEL E. BIETZ

B.S., University of Idaho, 2011

A Thesis Submitted to the Graduate Faculty of Georgia Southern University

in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

MAZETEC: A SCENARIO-BASED LEARNING PLATFORM

by

DANIEL E. BIETZ

Major Professor:	Andrew Allen
Committee:	Lixin Li
	Vladan Jovanovic

Electronic Version Approved:
December 2018

ACKNOWLEDGMENTS

I would like to thank Alauna Safarpour for her ongoing support and encouragement throughout this process. I would like to thank Dr. Paul Quinnett of the QPR Institute for his support, encouragement, and advice, which was instrumental to the direction of my research. I owe my deepest thanks go to Dr. Andrew Allen, who read and commented on numerous manuscripts. Dr. Allen's guidance was essential to my successful completion of this master's degree.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

INTRODUCTION

The economic, social, and health benefits of achieving higher education and practicing lifelong learning are widely researched, known, and accepted. It is also known what a disadvantage it is to have limited access to educational resources and training opportunities. This is not only a significant problem in rural areas, but also for most working adult learners who want to continue their education but do not have time to attend evening classes or figure out the daily objectives of an online course [1], [2]. Online learning is viewed as one cost effective approach to this problem.

Distance education enrollment has grown over the last 14 years, and roughly 30% of all college and university students have enrolled in an online course [3]. Concurrently, learning management systems (LMS) have grown and evolved to meet the demands and learning styles of students and content authors. They reach more people with fewer or limited resources. The federal government, as well as nearly all university systems in the United States and Fortune 500 companies, have an online learning component. It is convenient for adult learners who have full time jobs and challenging schedules, if the learning tasks are clear. The primary types of online learning environments are synchronous and asynchronous, or a combination of both.

Many of the synchronous online learning systems attempt to recreate the classroom experience with a live instructor lecturing over a video feed at a set date and time. In synchronous systems, students interact in real-time with their instructor and fellow students. The asynchronous learning systems enable on-demand learning, where the learner consumes the content – e.g., text, video, or slides – at a self-determined pace, location, and time.

MOOCs (massive open online courses) recreate the classroom experience in an asynchronous manner. Coursera, a MOOC provider, boasts having as many as 120,000 students enroll in a single course [4]. Yet the average MOOC completion rate stands at 13%; that's an

87% dropout rate [4]. This speaks volumes about student engagement in the “one size fits all” MOOC format. A four-year, E.U.-commissioned case study on MOOCs at France Université Numérique (FUN) from October 2013 - July 2015 found that out of 1,800,000 enrolled students across 140 courses offered through 50 different educational institutions, the rate of completion of the courses was about 10% [4]. Simply recreating the classroom experience online is no longer a novel approach and carries with it the same limitations that online learning was supposedly going to solve, such as adapting the learning to individual needs.

Asynchronous online learning is not actively engaging because the learner's experience is passive and motivated by self-interest, such as simply completing the course. Most online learning and distance education courses today are asynchronous and present information in what the instructor determines to be the most logical order, often from beginning to end. The learner's opportunities to interact with the content are limited to quizzes, assigned homework, or references to external content. Student grades are the ultimate determinant of how well the content of an online course was organized, applicable to the students, and clearly communicated. In these asynchronous passive learning environments, the action of learning is separated from the action of doing. According to the educational theory of constructivism discussed in the literature review the best way to learn is to practice in the moment so the learner receives immediate feedback on what he or she is learning, understanding, and comprehending. A passive course may demotivate learners if they don't immediately relate to the importance of the content or how it applies to their area of interest. Learner disinterest may initiate a pattern of disengagement, poor performance, and failure to complete the training.

It is also challenging to teach case-based scenarios in high-pressure domains in online learning environments. Case-based scenarios in high pressure domains includes areas such as suicide prevention. For example, the job of a suicide crisis line volunteer may include the following scenario:

> *The phone rings, you answer, the caller is crying. They called to decide whether life is worth living or to die by suicide. You have 60 seconds to talk them down and if you say the wrong thing, the caller may die by suicide.*

There are many jobs that require making difficult decisions under pressure; however, there are few where someone's life is at stake. Suicide prevention training is required for health care practitioners in 29 states [5], but while the certificate satisfies the legal requirement, it is the training that enables the health care workers to save lives.

This method of instruction is called *Constructivism* [6], and it is based on the belief that efficient learning occurs when learners are actively challenged to simultaneously understand and apply what they just learned to a simulated challenge, followed by immediate performance feedback as opposed to passively viewing or reading information. This is related to research [7] on state-dependent memory that has found that if the task is performed under pressure, training should be conducted under pressure. However, there are no generic distance education learning management systems today that uses both a dynamic constructivist approach that are able to simulate pressure and deliver the course in an unsupervised asynchronous manner.

This thesis presents introduces Mazetec, an adaptive scenario-based e-learning platform. Mazetec implements a state-dependent, constructivist, pedagogical approach to online learning. It enables subject matter experts (SMEs) to create interactive, state-dependent, case studies or courses with branching logic for online learning and knowledge testing. Mazetec is a complex web application designed to deliver decision-based or case-based educational scenarios and simulations in a time-unlimited or time-limited, non-linear, online format for high-pressure domains, such as crisis lines, emergency medicine, police, crisis management, or cyber security. A learner must navigate to the correct end of a learning maze within the set time limitation or not. The non-linear flexibility to create many different concurrent paths and outcomes in a course allows SMEs to create *teachable moments* by validating a learner's incorrect assumptions and presenting them with the consequences of their actions. In addition, the platform collects a significant number of useful analytics, such as overall completion time, the path of decisions, and time to make each decision, and allows the SME to replay each user's attempt as well as aggregate the analytics for each maze.

It is well-understood that most organizations and institutions that require online learning systems are locked into those solutions. The system presented in this paper was designed to work with these systems to allow SMEs to fill a gap overlooked by commercial vendors in distance education delivery. Mazetec can serve as a standalone training system or as an integrated supplementary activity provider to an existing course in an organization's existing LMS. Mazetec is working toward xAPI compliance to easily integrate with existing LMSs and learning record stores (LRS).

Mazetec is currently being used by the Substance Abuse and Mental Health Services Administration (SAMHSA), an operating division of the Department of Health and Human Services (HHS) of the federal government, in conducting a study to better understand how the use of active learning strategies (the Mazetec software)[1] teach the QPR suicide mitigation intervention affects trainee knowledge and skills retention over time. Specifically, the SAMHSA study seeks to determine if the Mazetec "interventions following training participation increase

---

[1] The Mazetec software is referred to as "the booster" in the study design documentation because it is utilized as a refresher training in the controlled experimental design.

the effectiveness of gatekeeper [suicidal behavior identification and mitigation] trainings, particularly in terms of promoting identification of at-risk youth" [8]. Although the results of this study are not yet available, preliminary data analysis indicates evidence of user engagement and demonstrates the utility of the various analytics collected by the Mazetec system.

The main research questions of this thesis are:

1. How can we engineer a learning system to build, manage, and deliver unsupervised branching case-based scenarios to simulate a high-pressure situation that a learner must resolve such as suicidal person in crisis and the learner talking them down?
2. Will educators and learners find value in a scenario-based learning system?
3. Does the use of these interactive scenario interventions during or following training participation increase effectiveness of the trainings?

The rest of the thesis is organized as follows, Chapter 1, relevant background information is introduced, which will allow a more detailed understanding of the problem domain and the literature review. Chapter 2 presents a domain analysis of online learning management systems for the relevant problem domains. Chapter 3 explores these problems further with scenario-based learning system feature analysis. Chapter 4 breaks down the feature analysis and proposes a technical design for a learning system architecture and how it works. Chapter 5 discusses the technical implementation of a prototype system that was engineered. Chapter 6 presents the study results and discusses the results and the implications. Chapter 7 concludes the results, discusses Mazetec's contributions, and discusses future research.

CHAPTER 1 - BACKGROUND

This section will provide the motivation for the topic and review relevant literature related to the objectives of the thesis.

Motivation

The inspiration for the Mazetec system stems from a 15-year relationship with online learning and a leap of faith from a longtime client and now business partner. From the first online course I built 15 years ago to the most recent ones I have experienced, I have found no constraint has been more limiting than linearity. The real world seldom follows a straight line; it is a trial by fire, which can be easily organized into situations, actions, and outcomes. I have found that most courses I have built and experienced try to follow this pattern, especially continuing education courses and on-the-job training. However, they are constrained to a linear navigation model that only allows users to move forward or backwards. They also leave the course author with little choice but to organize the content linearly for all students to click through. While this works well for some domains, it is not ideal for all domains, nor is it ideal for all learners or learning styles.

The first online course I created taught high-school students about the importance of credit and the economic consequences of mismanaging one's credit. In an effort to make the content more engaging and memorable, the course was written as a dialog between two characters, and all of the scenes were illustrated as cartoons with a quiz at the end. It was organized in the same linear fashion as hundreds of online courses with the back and next buttons to navigate through the content. Students blew through them, clicking the next button as fast as the page would load and guessing their way through the quiz. Or, they would Google the questions looking for answers, or ask someone next to them for the answers. Then, they printed the certificate and were on their way.

My experience in the federal government has been no different. Annually, employees are required to complete mandatory cyber security, ethics, and diversity training. Most employees click through as fast as the page will load to reach the quiz and either Google or guess the answers to meet the 80% competency threshold. They then print the certificate and repeat the next year.

In 2009, I started consulting for the QPR Institute, Inc. QPR is a suicide mitigation training company with in-person and online training on how to recognize the early signs of suicidal behavior and how to react and respond to get the person the help they need. In the mid-1990s, QPR developed a custom domain-specific, 1-hour online training program called QPR GateKeeper Training to teach anyone the basics of suicidal behavior and what to do if they encounter a suicidal person. QPR wanted a flexible online system that the founder, Dr. Paul Quinnett, Ph.D., could use to create their advanced courses for professionals that ranged between 2 and 50 hours. The requirements were the same as many other courses one might encounter, such as video, pictures, typographical content formatting, quizzes, resource links, surveys, and downloadable documents. I used Joomla, CMS (Content Management System), and the commercial LMS component J!LMS (Joomla Learning Management System), along with a myriad of support components to meet the business needs.

QPR's advanced courses were a significant improvement in managing the course content and the new functionality, which, at the time, allowed them to stay ahead of their competition. The improvement to the administration functionality allowed QPR to created one master course and then duplicate and tailor it to each specific profession. For example, for the QPR system there is a 5% difference between the nurse course and the occupational therapist course is about 5%. However, their courses are different because their consumers demand it. Despite the new authoring features, the learner's experience between the original QPR GateKeeper training program and the professional courses are much the same. While the advanced courses offer more media and formatting, the path through the content from beginning to end is linear. The learner completes each step in order with intermittent quizzes to validate the learning before allowing him or her to progress to the next step.

It was therefore not surprising that when Washington State passed the Matt Adler Suicide Assessment, Treatment and Management Act (HB2366), QPR started seeing customers click through a 6-hour, state-mandated training course in 15 minutes. QPR added more images and video, more frequent knowledge check quizzes limited to three questions, competency exams, minimum time on each page, and limits on quiz attempts with a waiting period between attempts to deter learners from skipping ahead and guessing the quiz. Yet, there were still a handful of people who just wanted to click through as fast as possible. While all of these restrictions were intended to ensure the learners were engaging with the content and learning, instead they just had the effect of frustrating and confusing people.

Dr. Quinnett and I started a discussion on ways to improve engagement, knowledge, and skill retention in an online learning environment from the learner's perspective. Dr. Quinnett described the job of a suicide crisis line volunteer.

> *The phone rings, you answer, the caller is crying. They called to decide whether life is worth living or to die by suicide. You have 60 seconds to talk them down, and if you say the wrong thing the caller may die by suicide.*

There are many jobs that require making difficult decisions under pressure; however, there are few where someone's life is at stake. Suicide prevention training is required for health care practitioners in 29 states [5], but while the certificate satisfies the legal requirement, it is the training that enables the health care workers to save lives.

In negative course survey feedback, the highest occurring complaints were from health care workers complaining the course took too much time to complete. The typical use case for the health care learner involves a doctor, a nurse, or a counselor who is fully employed in a health care setting and who is required to complete this training. The training is paid for by their employer; however, many are required to complete it during non-working hours. The learner is highly educated, and many think they already know the material or will not ever need it. However, they take it because it is required. As one student explained in an open-ended post-survey question following the QPR training: "This is a required module by our state licensing agency and a complete waste of time for ortho surgeons and most docs. Granted, it is well done and actually a good module, just 100% not applicable to my personal practice, and just something I HAVE to do for my CME." The organizations purchasing training vary widely, yet most who purchase it are responding to a legal requirement. While we cannot control the learners' busy schedule or the environment, we can control the content and the training delivery mechanism.

If learners are taking the training online asynchronously at their own pace with the TV on in the background or while talking on the phone, and they just click through the content and play the video but walks away, and Googles or guesses through the quiz, how could anyone expect them to retain anything? Most people treat required training as a chore to get through as quickly as possible, but when you are the legislator, the company leader, or the manager, the training is a benefit to everyone and is often provided at great financial expense. As researchers we ask, what can we do to make learners want to learn the material? More specifically, what can we do to engage the learner in such a manner that they treat the training with the importance expected of them and in such a way that improves knowledge retention and job preparedness, and it works at a distance, asynchronously with self-guided content?

In this thesis I present preliminary evidence that indicates students required to complete two time-limited scenario-based training "booster" sessions with the Mazetec software, actually go back and complete the maze several more times than they are required to. This, I contend, indicates that the Mazetec content delivery format engages learners, encouraging them to voluntarily re-take the training maze more often than required for their professional programs. This demonstrates a clear point of departure from the typical mandatory training program behavior of clicking through as quickly as possible and only completing the training after multiple email reminders (even threats of repercussions) force the trainee to do so.

Literature Review

Learning Technologies

Online learning, learning management systems, and educational technologies are no doubt an industry of their own. However, it was as recently as 2013 that the Institute of Electrical and Electronics Engineers (IEEE) approved the Learning Technology Standards Committee (LTSC), of which this author is a member of the LTSC. The LTSC helps to standardize protocols that promote interoperability among learning systems by establishing technology standards, such as the Experience API (xAPI) protocol. Even more relevant, in the fourth quarter of 2017, the IEEE approved the Industry Connections Industry Consortium on Learning Engineering

(ICICLE), which focuses “on how to design and engineer products that are more efficient, engaging and effective. This [initiative] is the basis for a new discipline that focuses on the engineering aspects of learning systems” [9]. The creation of the new “learning engineering” discipline and support for these IEEE groups is clear evidence that there is a strong demand for new learning system standards, methods, and technologies. This work seeks to contribute to this new growing body of knowledge.

Connolly, Stansfield, & McLellan [10] state an online learning system needs to be problem-focused so that the user feels the problem and environment are authentic, realistic, and sufficiently complex, and need to include conflict, interaction, cooperation, and competition.

Dagger, O’Connor, Lawless, Walsh, and Wade [11] explored the history of e-learning platforms and pointed out future challenges the next-generation platforms may face. They identified a shift from generic solutions to topic-specific applications that will need to support a wider range of interoperability architectures and the ability to integrate with existing and newly adopted services such as federated authentication and data interoperability.

The European Commission established a study to map and analyze prospective technologies for learning (MATEL) in Europe. The 14-month-long MATEL study included over 200 stakeholders (e.g. policy and decision makers, teachers, trainers, technology providers, technology developers, students, parents, and researchers) to understand how learning technology is changing and to discover new innovative learning technologies. In regards to addressing experiential workplace training, the study turned to serious games as a tool to emulate workflows and personalize learning [12]. Serious games are computer games designed and developed with the intent to educate the player on a particular topic in an engaging interactive environment that allows the player to assume the roles of decision-makers in simulated situations that recreate reality and allows them to practice their skills [13].

While there are many ways to teach, the foundation on which the Mazetec application was developed addresses the first research question. If learners think they already know the content and method and how to talk down a suicidal person, then how can we simulate a suicidal person to create the feeling of pressure in an unsupervised learning system and require the learner to talk them down?

Constructivism approaches to online education

Constructivism is an educational philosophy based on scientific evidence, that can summarized as “learn by doing” and is one of the most cited pedagogical approaches for unsupervised learning systems [6]. Learners construct their own knowledge, understanding, and meaning of a subject by exploring a domain, having experiences within that domain, and reflecting on those experience to reconcile new information with similar past experiences, either to challenge or reinforce their beliefs [6]. The constructivist view encourages active learning by using real-world scenarios, case-based scenarios, problem-based scenarios, or experimentation in a safe learner-centered environment to construct one’s knowledge and to reflect on how understanding is changing.

Huang [6] argues that a new pedagogy is needed, identifies the barriers of using a constructivist approach in online learning, and proposes an implementation approach that addresses the barriers. The instructor cannot simply recreate the classroom experience online; the learner’s physical isolation demands that the learner moves from passive receiver to active controller of his or her own learning. Huang proposes the learning environment needs to be interactive and correspond to predetermined learner-centric, real world problems or conflictoriented scenarios that can be easily navigated by the learner and evaluated by the instructor. For example, assume the role of a nurse. A patient walks in with numbness in his left arm and chest pains, what do you do? This approach shifts the distance learner from passive learning to active learning and puts her in control of a more memorable and interactive experience.

Diniz dos Santos, Strada, and Bottino [14] have cited constructivism as the instructional pedagogy in distance education applications that embed learning in realistic contexts, such as games and simulations. They found in their extensive review of sustainability learning via digital serious games that constructivism was used to foster "content transfer between game actions and real world concepts". Literature [15] on crisis management training systems also cite using the constructivist approach to education as opposed to teacher-centric or the instructivist approach. In medical education, there is a growing body of evidence that case-based testing, dubbed *test-enhanced learning*, is increasingly being used to train clinical reasoning. A recent review found statistical significance in short-term learning and long-term learning 6 months after the treatment [16].

State-Dependent Stress and Pressures

State-dependent theory argues that if you work under pressure, learning needs to happen under pressure. Extensive evidence from controlled experiments shows that information learned in a specific mental state or physical environment is recalled more effectively when the subject is returned to that same mental state or environment (see [17] for evidence on physical context; see [7] for evidence on mental state). One of the most famous studies on state-dependent learning conducted by Eich [7] asked subjects to study words while smoking either marijuana or a tobacco cigarette and then varied the substance smoked prior to being tested on the material. The researchers found that the participants who smoked the same substance before the exam and while studying performed better on the exam than those who smoked different substances studying versus just prior to the exam. This suggests that information recall is better when it is performed in the same mental state the subject was in when the information was first encoded.

Other literature shows evidence that this carries over to a state of stress or pressure. The effects of time pressure can induce a stressful state in a simulated environment. Studies examining decision-making under time-pressure in a simulated environment found significant evidence that limiting the available time in a computer-based simulated training had a measurable effect of a participants' decisions [18]–[20]. For example [21], when the Israeli Air Force (IAF) examined the effects of time pressure on top military commanders' decisions in over 74 combat simulations, they found that time pressure has more of a negative impact on less-experienced commanders than experienced commanders.

The findings in [18]–[21] suggest under time pressure people use the availability heuristic [22] because the participants' decisions are biased by the cognitive procedures that are readily available from prior real world or training experience. As the number of mistakes under time pressure increases it suggests the learner didn't encode the training information in the correct state or lacks first-hand experience.

Lin and Su [19] studied time pressure in computer training software by constraining the time the participant had to respond to an emergency chemical spill simulation. They found that the time pressure significantly decreased decision accuracy compared to unlimited time. They also found time-constrained, decision-based training to be statistically significant and required fewer overall training sessions because the learner encoded the information in the real-world, decision-consequence sequence and stressful mental state. This suggests that information is more readily available when the training resembles the sequence and state of the real world. The findings make this approach to training very appealing since it is low cost, low barrier, and highly effective. Research by Bacon et al. [23] on developing realistic crisis management training systems also supports these findings.

To incorporate this evidence into Mazetec, I employ the use of a timer to simulate the pressure a practitioner would experience while on the job. For instance, the suicide mitigation training simulates the pressure on the learner to respond to text messages from a suicidal patient in a chat room quickly to avoid the patient logging off because of non-response by the crisis line worker. Next chapter 2 presents the findings from a domain analysis exploring the state of online learning and different approaches researchers have taken to achieve a branch scenario system.

CHAPTER 2 – DOMAIN ANALYSIS

This section reviews the feature domain of educational serious games and simulations to achieve branching learning scenarios and discusses the literature and development efforts in computer science education technology. There are many learning management systems; but within the domain of serious games and simulations there are a few commercial platforms that do not require technical expertise. However, there are many open source, academic, and domain-specific applications.

Serious Games

Serious games are computer games designed and developed with the intent to educate the player on a particular topic in an engaging interactive environment that allows the player to assume the roles of decision-makers in simulated situations that recreate reality and allows them to practice their skills [13].

Serious Game Feature Analysis

The Society for Research in Higher Education conducted a systematic feature-oriented domain analysis of educational serious games for higher education [24]. The analysis methodically breaks down the learning process in conjunction with game features and how those features connect with the learning process using Bloom's taxonomy of learning outcomes [25]. Bloom defined the *cognitive domain* (Remembering, Understanding, Applying, Analysing, Evaluating, Creating) as "a student's intellectual level that is what a student knows and how they organise ideas, opinions and thoughts. The cognitive domain connects with in-game activities that advances learning and knowledge and are integrated throughout in-game learning experiences" [24].

As explained in [24] a game should have clear goals and competitive elements and be designed within a set of structured rules, choices, and feedback to enable instructors and learners to monitor goal achievement. Goals should be achieved by the learner's decisions in the game, and the instructor should be able to see the series of decisions made within the game's session. Games rules are structured in two ways for learners, through emergence and progression. The emergence design gives the user a small number of rules he/she must use to achieve the object of the game. The progression design requires the player to complete a set of predefined actions to achieve the game objective. The predefined actions follow a sequence of the events in a story; however, it may also have rules that can change with player actions. Game feedback comes in the form of: (1) progress bars (2) in-game hints, (3) scoring (4) achievements, (5) experience points, (6) virtual currencies (7) prompts, (8) assessment tools, and (9) dashboards.

Table 1 categorizes game features as follows: activities, outcomes, feedback, assessment, rules, goals, challenges, motivation, collaboration, competition, and feedback and maps these features with the learning attribute and game mechanics.

Table 1. *Serious Game Features.*

Feature	Learning Attribute	Game Mechanic
Goals and Choices	Learning objective and interaction	Game journal, missions, objective cards, storytelling, nested dialogues, puzzles, NPCs / avatars
Tasks, activities and challenges	Information transmission	NPC-based task description, progress bars; multiple choices to select, major tasks, branch tasks, puzzles, research points, study, requirements
Rules		scoring, moving, timers levels, progress bars, 'game instructions including victory

		conditions
Collaboration and competition	Collaborative	Role-playing, community collaboration, epic meaning, bonuses, contest, scoring, timers, coins, inventories, leader boards, communal discovery
Feedback	Discussion and argumentation Information transmission	Game hints, NPCs, game levels, gaining/loosing lives, progress bars, dashboards; lives/virtual currencies to be used for buying game items from an online inventory; progress trees

Serious Game Architectures

Software architectures organize and structure a system's components into a high-level blueprint, much like that of a house. A reference architecture is a template that can generalize a set of implementation guidelines for a whole system or set of system features within an existing domain. Reference architectures may vary in the level of detail they provide; however, they are a good starting point for the best practices within a domain, which helps to facilitate the design decisions in the early phases of a project and promote reusability and interoperability, reducing development costs through component reuse.

System components are broken down to their functional requirements, which defines what the system is supposed to achieve. Functional requirements are defined in terms of use-case scenarios. Scenarios paint a picture of the system need, and each individual scenario describes one part of the need, often mapping to a component or set of components. Thus, the inverse of a scenario describes one viewpoint of the problem or gap. Scenarios not only serve as the preferred input for most architecture development methods, but are also used to judge how well the system is achieving the goal it originally intended to solve. This evaluation process is carried out by tracing the requirements across the system architecture to the intended end-users' experiences. Once the requirements are defined, the architect can proceed to propose the architecture design; however, in practice this logical approach is difficult to achieve.

Service-Oriented Reference Architecture for Serious Games (SORASG)

Carvalho [26] argues that a reference architecture for a reusable architecture would greatly minimize the cost of serious educational game construction, as well as reduce the time needed to create such games. She proposes service-oriented reference architecture for serious games (SORASG) as a means for software reuse to reduce costs. A service-oriented architecture (SOA) defines a software system as components, services, and consumers in terms of logically compartmentalized business outputs to achieve flexibility in the development and easier maintenance. SOA abstracts and modularized functionality as service so contracts between endpoints can be standardized, placing formal obligations between the consumer and the provider component of the service. This level of abstraction increases reusability and interoperability but is highly complex to implement.

The SORASG design categorizes the features into six functional domains: feedback, assessment, personalization and adaptability, game connectors, user profiling, and data logging. Since it is only a proposed reference architecture it does not address student-instructor interaction or information storage and retrieval.

Serious Game Business Scenarios

Carvalho [26] also identified a robust set of domain-specific business goals and system scenarios in her literature analysis and validated the sets with questionnaires and interviews with groups and individuals. Based on her findings, each scenario was assessed by its business value and impact on the architecture and then prioritized on the importance to the system as high, medium, or low. The business values were evaluated by “Reduce costs while maintaining quality” or “Reuse of technological solutions”. After she implemented and tested her approach, she discovered that network delays interrupted game play because the player interface was waiting for instructions from the game engine, due to degraded performance in the service layer.

RAGE Serious Game Architecture

Concurrently, another architecture was being developed for serious games by the RAGE project [27], funded by the Horizon-Program of the European Commission with the goal of interoperable components to support assets portability across systems, many programming languages, and independent of any one game engine. The RAGE project proposes a component-based architectural design so game assets, such as a statistical component, can be plugged into any other game that meets the RAGE design standard. This is accomplished by reducing the component code to JavaScript Object Notation (JSON) and by having the respective game studio rebuild the code base in the selected language, which can then be integrated with the game. While this is a significant step forward, the architecture was planned for compiled games designed in desktop studios, such as Blender and Unity3D. When the authors tried to implement the interoperable RAGE architecture with web-based languages JavaScript and TypeScript, the compiler was not able to recreate the methods present in the class nor the values that needed to be computed during the compiling process. To work around this issue the authors recommend avoiding methods; however, this is an impossibility in systems with lots of functionality because methods are the blocks of code that contain the game’s functionality.

Challenges with Serious Games

Many of the meta-analysis papers and research reviews analyze, study, and discuss serious games; however, regardless of their findings few, if any, actually make it to the market.

None of the articles describe the monumental undertaking of developing a 2D, let alone a 3D, educational game or simulation. I have experienced the process of developing a game on which we worked for 3 years and spent millions of dollars that never made it to market. The process of planning, designing, and developing a game from concept to the first functional prototype is enormously time-consuming, complex, and expensive. SMEs driving a project can easily find themselves lost in the thousands of decisions that need to be made, from the style, texture, and color of the avatars hair to the lighting orientation and particle density of a cloud. It is easy to lose sight of intended learning outcomes.

While Carvalho [26] argues serious games are not updated due to few reusable parts, this author believes it is due to the significant amount of effort in the first place and a realization that slides, lectures, and quizzes are far less effort and easier to evaluate.

In a systematic literature review of games for computing education, Petri and Gresse von Wangenheim [28] observed there was a spike in educational game publications between 2009 and 2012 and a large decline between 2013 through 2017, which they attribute to the complexities and challenges in game development. Of the 112 articles they reviewed describing 117 studies on 106 different serious educational games, only 19 studies were conducted with scientific rigor however, it was noted there were still challenges with collecting pre and post data in the learning environment and obtaining a sample size sufficiently large enough to produce statistically significant results. I speculate that the remaining 98 studies in their analysis [28] tested games that were built as black-boxes; in other words, researchers have no way of assessing what the player is doing inside the game (in-game events such as players' actions, scores, achievements, level progression, and so on).

As Ludwig [16] found, text-based serious games can deliver similar educational gains as their more expensive, 3D-based counterparts. For that reason, this work explores text-based serious games. Throughout the literature I discovered a number of simple branching text-based serious games – e.g. Alzheimer's awareness [29], teaching insulin management to medical doctors [13], environmental sustainability [14], Zoe Quinn's Depression Quest [30], crisis management [23], and connect with Haji Kamal [31], and then I discovered the Open Labyrinth project, a scenario-based learning platform.

Open Labyrinth is the product of the WAVES project [32] is an E.U.- commissioned, 3-year project to develop guidelines, tips, and tools for scenario-based learning (SBL). The WAVES project produces reports and a collection of Virtual Scenarios (VS) for different disciplines to demonstrate the potential and capabilities of the scenario in different learning activities. The WAVES project assets SBL is a more engaging and immersive experience for learners that connect their sense of identity and emotional involvement. The objective of the WAVES architecture is to develop an authoring platform to create and deliver VSs for different disciplines for large and small groups and for self-directed learners. VSs can be linear or semi-linear, or branched to be interactive and allow embedded multimedia, such as video and assessments. While the project doesn't recommend any particular technological approach, it does carry out a thorough needs analysis to support decision-making and problem-based learning and defines SBL application use-cases and authentication patterns for both academic and workplace learning. As originally identified, the goal of WAVE is to extend online courses by adding engaging and authentic interactive scenarios to linear content.

CHAPTER 3 – SCENARIO-BASED LEARNING SYSTEM FEATURE ANALAYSIS

This chapter analyzes the user requirements that will define the scope of the technical solution. The WAVES project commissioned by the E.U. is working toward producing a toolkit to define the domain for scenario-based learning (SBL) systems. As a part of this toolkit, the WAVES project conducted a 44-item online survey [32], which affirmed the strong need for a scenario-based learning platform that is accessible by subject matter experts across all domains. The survey gathered requirements for a scenario-based learning platform in April 2016. The responses totaled 161 participants from 21 different countries. The survey targeted learners, educators, and technologists who were already associated with the project.

The reports identified these key takeaways: usability, mobile-friendly for educators and learners, and integration with existing LMSs. Integration was identified as the single most important feature for an SBL platform's adoption.

The analysis reduced the feedback from respondents into 78 user stories to describe the needs and wants of a particular user from the perspective of three user roles – a learner, an educator, and a technologist. However, the user stories don't always clearly translate to system features. For example, the requirement to "allow learners to experience the consequences of their decisions in a safe environment" [6] doesn't clearly map to a system feature; however, we can deconstruct the requirement to a proposed set of system features. Allowing users to experience the consequences of their decisions requires the system to support a course with branching logic, thus, the feature is *branching logic*. To satisfy *safe environment* the scenario must synthetically simulate, and therefore, must need a viewer. However, in order for the learner to experience the simulation, he/she must interact or play with it. Thus, the system feature is a *scenario player*.

User stories are generally technology-independent; however, to develop them into system requirements, it's helpful to choose the technical approach and delivery. I chose to approach the solution with the software-as-a-service (SaaS) paradigm as a web application. After analyzing the reported user stories [32], I removed duplicate user stories and reduced them further into requirements, categorized them by user role, and using the deconstruction method above, assigned a set of system features found in tables 2, 3, and 4.. They describe the most important requirements for each user role and a system feature set to accomplish the requirement was assigned.

Table 2. *Maps the "Learner" User-role Requirements to a Scenario Based-learning System Feature Set*

User Role	Requirement	Feature Set
Learner	Allow learners to experience the consequences of their decisions in a safe environment	- Scenario Player - Branching logic
Learner	Virtual scenario should be able to mimic real life situations	- Mixed Content - Game features
Learner	A feedback mechanism during and after the simulated scenarios	- Authoring Platform
Learner	Scenarios should be placed in context including theories and concepts,	- Content
Learner	As a learner, I would like to work with other real people in the SBL system, so that I can be involved in simulation of real life discussions	- Multiplayer
Learner	Virtual scenarios should be based on realistic cases	- Contextual Guidance
Learner	The scenario player is easy to use and the scenario is easy to navigate	- Design - Navigation

Table 3. *Maps the "Educator" User-role Requirements to a Scenario-based Learning System Feature Set*

User Role	Requirement	Feature Set
Educator	Easy and intuitive to create scenarios; support and guidance; develop decisions, options and apply consequences for them, so that my scenarios can be relevant to real life.	- Authoring tool - Contextual help - Branching - Design
Educator	Scenario design guidelines; assess quality of a scenario; system feedback on good "wrong paths"	- Contextual help - System Analysis
Educator	A method to monitor a learner's progress and view history of all the decisions the learner made in the scenario	- Learning Analytics
Educator	Ability to create multiple scenarios	- Authoring tool - Dashboard - Storage
Educator	Ability to integrate with other existing systems such as an institutional LMS	- Integration - Interfaces - Plugins
Educator	Content should be assessable on desktop, tablet, and mobile devices	- Design - Responsive
Educator	Ability provide instant feedback, so the learner immediately understands the consequences of a decision	-Authoring tool -Branching -Feedback
Educator	The system needs to be simple to use and well-documented	-User Guide -Contextual Help
Educator	Ability to preview my scenario on the fly while creating it	-Authoring tool -Preview Function
Educator	Easy to use and easy to navigate	-Design -Navigation
Educator	Virtual scenarios should be based on realistic cases	-Contextual Guidance

We can then further analyze the system features by logically grouping the features into a parent-child relationship. This is a recursive process where features are organized, and functionality gaps are revealed, thus, starting the process over and further refining the features.

Table 4 groups SBL parent and child features and gaps.

Table 4. *Scenario-based learning system feature gap analysis*

Parent Feature	Child Feature	Gaps
Scenario Player	-Navigate branching -Responsive -Simple design -Scenario focused	-Learner access patterns -Scenario player access controls -Player use-cases
Authoring tool	-Create branching scenarios -Feedback -Contextual Help -Preview Function -Responsive -Dashboard -User guide	-Scenario composition -Author access patterns -Scenario access patterns -Account management -Scenario access controls
Learning Analytics	-Individual history -Aggregate history -Path analysis	-Data models
External Connector	- Integration - Interfaces - Plugins	-Communication protocol -Authentication and data standards

Defining a Scenario

As the scenario feature gaps emerged in the table above, they are explored further in this section and the following sections. Continuing the previous table, we identified that for the scenario to exist it must first be created by the educator, and the educator user stories indicate he or she would like a simple tool to author the scenario. However, in order to create the scenario authoring tool, the components of a scenario must be defined. Based on the user stories and previously identified features, a scenario can be defined as a collection of the contextual information, decision points, feedback, consequences, and a start and endpoint connected in such a way that a learner can experience the consequences of his or her decisions. A possible depiction of what it might look like is illustrated as a graph structure in Figure 1.

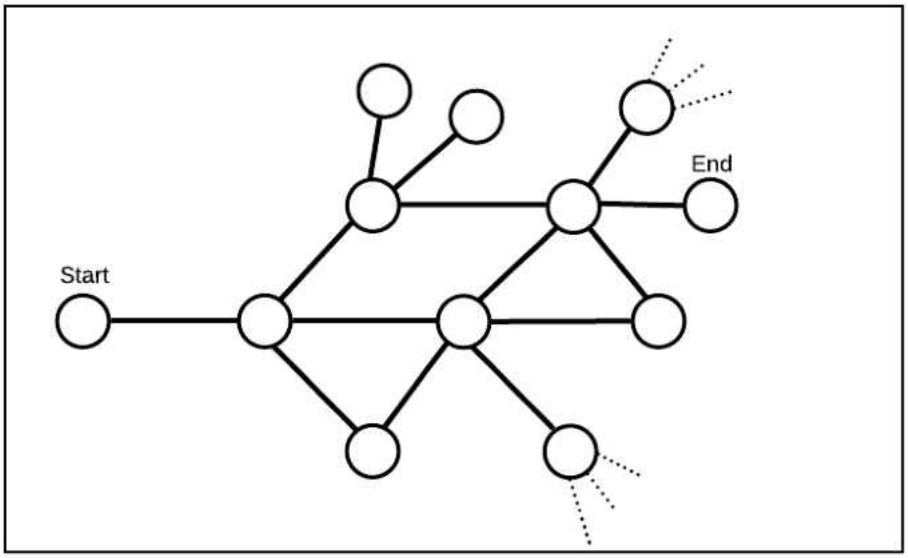

Figure 1. A concept illustration of a scenario map

Viewing the scenario as a graph helps to simplify it into a collection of interconnected nodes, where each node represents a decision, contextual information, feedback, consequence start, or endpoint. The node types emerge as more detailed interaction that can be described using the illustration. The learner navigates from node to node by making decisions (or traversing specific edges) and experiencing consequences, until he or she finds the endpoint. In order for the learner to make decisions, he or she needs to be given some contextual information and a set of options from which to choose. Consequences and feedback serve the same basic function within the scenario by providing the user with positive, negative, or neutral contextual information. However, the term feedback is not specific enough for a node type because every choice provides the learner with feedback. If a learner makes a decision that leads to another scene, the feedback is the next scene and the associated options. There may also be cases when the feedback is not a consequential decision, nor is it a path forward; in context help would more accurately be a *hint*. For the purpose of simplicity, we will call a consequence a *warning*. The feedback nodes are a warning type node and a hint type node.

There are two methods of ending a scenario: a successful finish or a failure to finish. Thus, we have the finish type node and the fail type node. To finish, the learner must reach a finish node. However, a failure may result from an incorrect decision determined by the educator. In suicide prevention, poor decisions can be fatal; therefore, the session must end immediately, so that the learner can also experience the consequences of the fatal decision. However, the learner should also have the opportunity to immediately restart the scenario to try again.

As we reviewed in the literature, feeling stressed or pressured can significantly impact one's decisions, and we discovered in the theory of state dependence that people who train under simulated pressure perform better in the real world under pressure than those who did not train under stress. Evidence presented in the literature proved that pressure can be simulated by limiting and manipulating the available time the learner has to complete the activity. Therefore, it can be postulated that limiting the time a learner has to navigate a scenario and evoking feelings of pressure and stress during training will increase the decision error rate. In addition, with the possibility of a learner choosing a fatal decision in a time-limited scenario, the learner may also fail if he does not reach a successful finish node before the time runs out.

Limiting the time also gives us the ability to distinguish between *warning* type node behaviors and *hint* node behaviors. Learners are directed to the warning node when they make a poor decision, as in a real life poor decision often wastes time and adds pressure. So, in a time-limited scenario the warning node should also have the ability to subtract a variable amount of time set by the author for a more realistic and authentic learning scenario. Hints, on the other hand, are for learners who know they need help. Thus, an author may want to use it as a point of reflection, so it would have the optional ability to pause the timer in a time-limited scenario.

Color coding the nodes also helps to further distinguish the type. Table 5 lists the Mazetec node types, a description of its behavior, and color coding.

Table 5. *List of Mazetec node types a user can experience in a scenario*

Path Nodes	Description	-Behavior	Color
Start Node	Scenario starting point, provides baseline contextual information, and starts the session.	-Start session -Next	White
Scene Node	This is the decision point this node provides contextual information with options for the learner to choose from. Options connect to other nodes.	-Many choices	Blue
Warning Node	This node provides consequences to the learner by "warning" the user with the provided message and sends them back to the previous node.	-Subtract time -Back	Yellow
Hint Node	This node provides a hint to the learner. The user is provided a feedback message and sends them back to the previous node.	-Pause time -Back	Turquoi se
Fail Node	Landing on this node ends the session and displays a failure message determined by the author. The learner is given the option to restart the scenario.	-End session -Restart	Red
Timeout Node	In a time-limited scenario, when the time runs out the session ends, and the learner is redirected to the timeout node. It displays a timeout message determined by the author. The learner is given the option to restart the scenario.	-End session -Restart	Red
Finish Node	Landing on this node ends the session and displays a successful finish message and "passes" the learner.	-End session	Green

In order to build a scenario, the nodes need to link together; in graph terminology these are called "edge*s*." The edges are the node's options. For example, an author constructs a scene node that lists a patient's symptoms and asks the learner how to proceed given the following set of choices: decision A 'take a sample,' decision B 'ask a follow up question.' Decisions A and B are the edges and link to their respective nodes.

Scenario Authoring Portal

Reviewing the educator requirements, the author needs a simple point and click interface that is easy to navigate and mobile-friendly. In this system environment, the author needs the ability to create the scenario with in-context help, a feature to preview the scenario during the creation process. The system needs to support an author's ability to create multiple scenarios and a method to monitor the progress of learners and view the history of a learner's decisions in the scenario. Authors also need the ability to integrate the scenario-based learning system with other existing systems, such as their institutions learning management system. As identified previously, user system accounts are needed to save multiple scenarios. However, first there must be an *author* user role and a method to authenticate users in the role before allowing access to an *authoring portal*. Each author will have her own user account and scenarios will be associated with her account. An authenticated account begets additional requirements for user login, user registration, profile management, access controls, permissions, and security considerations. However, these will be addressed in the next chapter discussing the technical implementation. The author portal gives us a landing page to organize the author's functionality. The author portal functionality can be organized into five primary functional views a dashboard, authoring tool, analytics, profile manager, and the scenario player.

Scenario Authoring Tool

The authoring tool will need to handle all aspects of the scenario creation process and yet be intuitive and easy to use. Through previous analysis, the composition of a scenario was defined as a collection of nodes connected by edges. Additionally, the scenario will need a baseline configuration to define its variables and behaviors. For example, each scenario will have a title, introductory description or instruction, determine access methods, and, if time-limited, it should have a time-limit and a timeout modal. Table 6 lists the considered scenario configurations. The baseline configuration should be known to the system before the nodes are loaded into the player because it may impact the available options or behaviors during play.

Table 6. Scenario Configuration Parameters

Configuration Parameter	Description
Title	Title of the scenario
Description	The description is stored as an object
Randomize option order	[IF ENABLED APPLIES TO THE WHOLE SCENARIO] randomizes the option set for each scene presented to users
Timer	enabled or disabled
Time limit	[IF TIMER ENABLED] total in seconds
Time penalty	[IF TIMER ENABLED] time in seconds
Time bonus	[IF TIMER ENABLED] time in seconds
Timeout title	[IF TIMER ENABLED] title of the modal
Timeout message	[IF TIMER ENABLED] message in the modal

Once a scenario is configured, the author is ready to start building his specific scenario. It is known from prior analysis that all scenarios will have a start node. Thus, the system will generate the starter node using the scene type for the author to build upon. Once the first node in a scenario is populated with content and choices, the challenge is connecting the parent node's edges (choices) to the corresponding child nodes. In the process of connecting the edges the author will need to ability to construct new node types configure it and connect it. During this scenario design process, the author will need to preview her creation along the way.

System Accounts

It is expected that most authors creating the scenarios will be affiliated with an organization such as a university or educational institution, business, or government agency. My experience in government has taught me that there are multiple decoupled roles within most large organizations. Once a potential author discovers the software and likes what he sees, he will want to test it on some sort of trial basis. If it matches her needs, she will want her IT department to check to see if it's compatible with the existing system. If it is, the organization will have some kind of purchase approval process. Then it's ready to be purchased by someone in the billing department and then sent on to the IT department to integrate the software. During this time, the SME may or may not have already started creating and testing scenarios.

The processes organizations go through to adopt new software take time and are messy. From my experience in government, it can take three months to years years to purchase software. To that end, the organization should have an account with multiple user accounts to give the flexibility needed for a large organization. However, there may also be instances where there is only one user that fills all of the roles.

We cannot predict who will serve in what role or an organization's process. So, regardless of the order of operations, we can determine a few things. When a new user signs up for the first time, it should create an organizational account. It should be incredibly easy for users in the system to invite another person to access the account without the need to share their account credentials. Presumably, once an SME tries it, he will invite his IT folks, then invite the purchaser. The SME may then want to invite a student to create the scenarios for her and then possibly another SME to share and review her work. Thus, the baseline account should be an organization type with which multiple user accounts can be associated. We can make the assumption that to join an organization, a user must be invited by a member of that organization, and all user-created data (e.g. scenario, usage data) should be owned by the organization. If a user account is deleted the scenarios created by that user will still exist within the organizational account and be accessed by the other members.

At this point in time there is only a need for a single authenticated role, which is the role of the scenario author; however, the need for more system roles will be explored in the future. Table 7 illustrates the types of system accounts.

Table 7. *Account Type Descriptions*

Account Type	Description
Organization Entity	-Created when a new system user creates an account -Represents an organization with multiple users. -Constrained by a limited number of users determined by a plan -Users join by being invited by organization members -Must have at least one member.
User Account	-Created by a registration process -An authenticated account representing an individual. -If a user is creating an account for the first time, the organization account and their user account will be created simultaneously. -Can invite non-registered users to join via email, which are sent through a registration process and associated with the organization.
Author Role	-Default user account role -Default Organization admin level permissions -All of the following accounts are Author accounts -Must be associated with one and only one organization

Permissions

Account access begets the need for permissions management. User permissions should default to the same level of access for all the users within an organizational account. However, we can still give users control by allowing them to manage the level of access different users have in each major component of the system. Scenarios should be contained to an organization and accessible to all of the organization's members, but the author should have the ability to limit the visibility of a scenario, such as a draft from the rest of the organization. Authors should also be able to sort and filter by scenario author.

If the author shares her scenario, the recipient should be able to play it without any system credentials, and the author should be able to view her usage statistics. Usage statistics in a learning system are referred to as *learning analytics*, which will be addresses later. An author should be able to share a scenario by means of a link or integrate it into another learning environment. Table 8 describes the account features and permissions.

Table 8. *Account Features and Permissions*

Account Features	Requirement Discovery
Invitation function	-Invite users to create an account and be associated with the organization account from to which they were invited. -Should be constrained to the number of users defined by the organizational account.
Account Management	Invite users to the organization function Control access permissions
Author Access Permissions	Any organizational user can control Read, Write, Edit permissions for the following sections: - Maze - Payment and Billing - Connector - Analytics
Maze Access Permissions	- Maze - Hide from organization

	- Sharable protected link - Protect access link with password

Scenario Player

The scenario player is the Learners interface with the scenario. It displays all of the content and presents the defined options to the learner. The scenario player should be simple, distraction free, and easy to use. It needs to execute and behave as the author intended in the scenario. It should be able stand on its own or be able to be integrated into another external system.

Accessing the Player

If an author shares a scenario, learners should be able to access it without the need for credentials. Learners will be accessing it from another LMS, a MOOC, or by a link over email. System interoperability was the highest predictor of SBL system adoption of a scenario-based learning system, so there should be an endpoint to authenticate and track the incoming user.

Learning Analytics

The goals of the learning analytics are to act as a data collection instrument for all users and to make the data accessible to the admins, authors, and externally-connected systems. The collected player data can be mapped to goals and also provide a framework for processing, analyzing and interpreting data streams generated from online learning experiences with respect to the learning goals. In addition, the analytics should also generate standardized learning records so the SBL system is interoperable with other systems.

The Experience API (Application Programming Interface) or xAPI for short, is a new e-learning standard developed by the Advanced Distributed Learning Initiative (ADL) and adopted in October, 2017. The xAPI specification takes the perspective that e-learning today is an ecosystem of platforms with many silos of data and it disassembles e-learning and makes it available. xAPI successfully frees data so that it may be exchanged between many systems, by redefining user activity data in e-learning systems to the simplest of terms Actor:Verb:Object “I did this” or “Sue completed programming course” activity statement.

We can take advantage of this standardization effort in learning analytics. xAPI defined data and communication models to standardize how track learning activities are tracked, thus making them interoperable and portable between learning systems. The xAPI specifications are designed to standardize tracking activities and interaction inside educational resources and statements represent the learner’s sequence of interactions. Given the growing adoption and government support we can settle on the xAPI standard with confidence.

This chapter analyzed and discussed the system requirements derived from the user stories. The primary components that are needed in the system are the scenario authoring tool, scenario player, account management system, analytics, and an integrator. Figure 2 depicts a feature map for the SBL system.

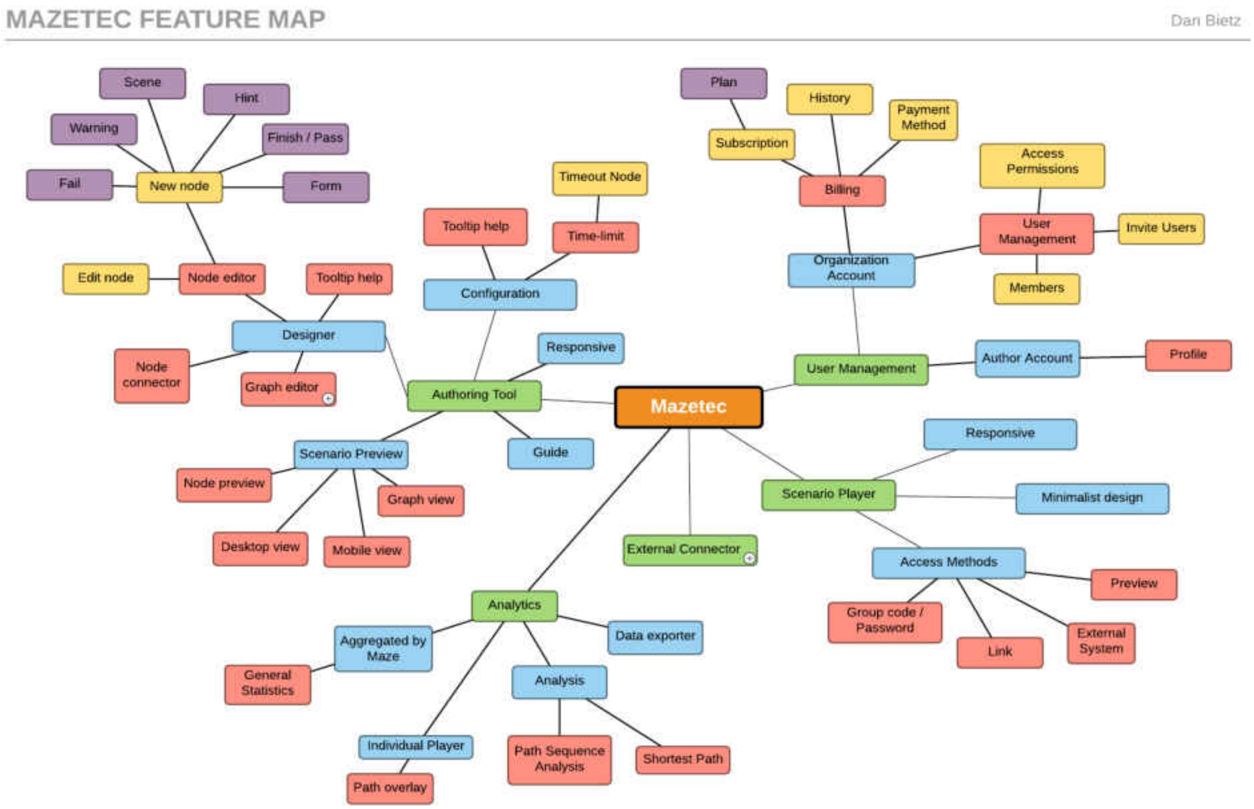

Figure 2. Displays a feature map for the scenario-based learning system.

CHAPTER 4 - SBL TECHNICAL DESIGN

Overview

The previous chapter examined the scenario-based learning system (SBL) user needs and described them in terms of system requirements. This chapter builds upon the requirements analysis, proposes a technical and architectural approach, identifies system standards, and discusses design decisions that will take place during the development phase. It will summarize the key system components and identify the system technologies. Throughout the rest of the paper we will interchangeably use the terms *scenario-based learning system* and *virtual scenarios* with *Mazetec* and *mazes*.

Objectives

The development objectives are to create a minimum viable product using a simple architecture to create a prototype of the SBL software for use in a study. During the development process we will use libraries, frameworks, and tools that already exist. However, this will be a new standalone system. Technical decisions will be made by balancing functionality, complexity, and schedule. As discussed, the WAVES project validated the market demand for SBL systems existence, but the way in which we meet those needs can significantly impact usage. The goals of the system are to allow educators to create branching virtual scenarios we'll call "mazes" in a simple and easy to use point and click environment and to be able to integrate with an institution's existing learning management system (LMS).

User Definitions and Use Cases

As we recall from the requirements analysis, educators and learners in existing institutions are the primary target audience; from the system's perspective we will call the user roles authors and learners.

Authors

Authors create mazes, manage and share access links, and review overall and individual maze analytics. Authors are typically subject matter experts (SMEs) in a particular domain. Authors should be able to embed a maze in another system's online course, so the author should have the ability to configure the Mazetec system to integrate with her institution's system. The author should be an authenticated role, meaning she will have a username and password or some other credentialed access to the system, such as OpenID.

Learners

The learners are the end-users that play the maze. Learners may access a maze by clicking a link, being automatically redirected from their system to maze player, or within their institution's LMS or online course via an embedded maze.

When a learner plays a maze, his interactions in the scenario generate usage data (i.e. his chosen path through the scenario), which should be saved in the Mazetec system. This usage data can be anonymous, identified, or a mix of both, depending on the scenario's configuration. Identified usage data comes from learners who either may have been transferred to the scenario system from their institutions system or may be prompted to enter their name, email, or other identifying information to access the maze. The learners who are referred from another system should arrive with a payload that includes user identifiers from their system. This may be a username or id, an email, or something else, along with the referring organization's homepage. Including the homepage along with the provided user identifier will help to avoid database conflicts as the application grows.

Use Cases

Figure 3 is a use case diagram of each user's role within the system.

Use Case 1: The author creates the maze and the learner plays it.

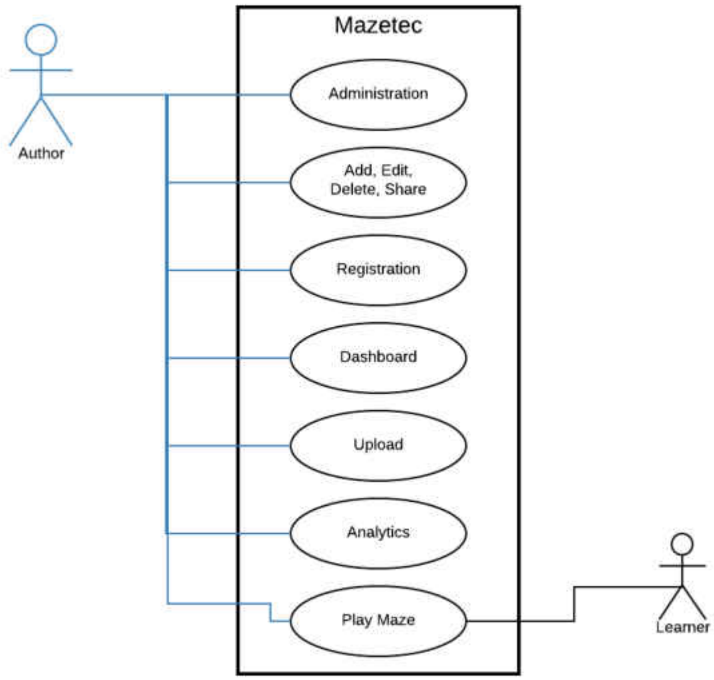

Figure 3. User case 1 diagram of an author and a learner.

User Case 2: The learner accesses the maze through his institution's LMS. This may be achieved through an embedded maze player or a redirect.

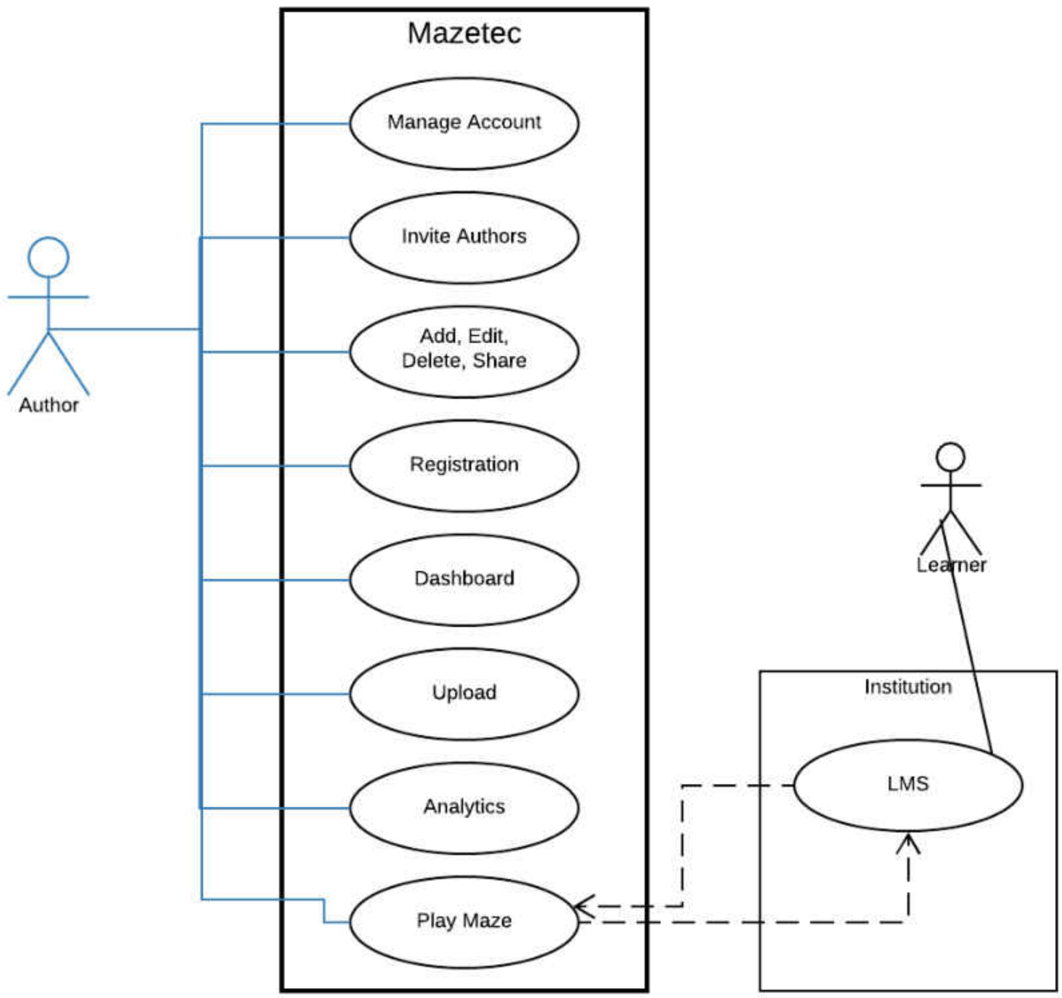

Figure 4. Use case 2.

Use Case 3: The Learner accesses the player directly however their usage data is sent to the institutions Learning Repository Store.

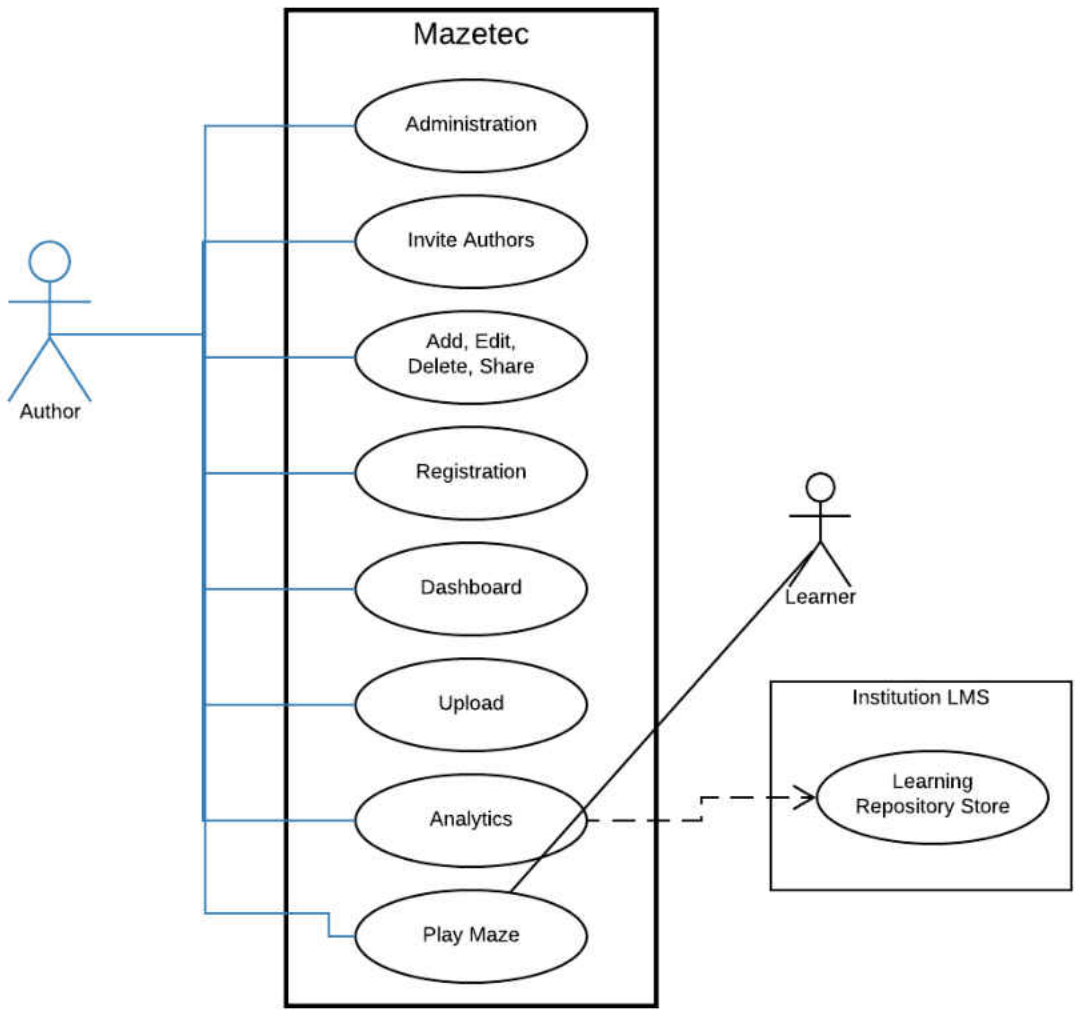

Figure 5. Use case 3.

Use Case 4: Author and learner accessing the system using their institution's credentials.

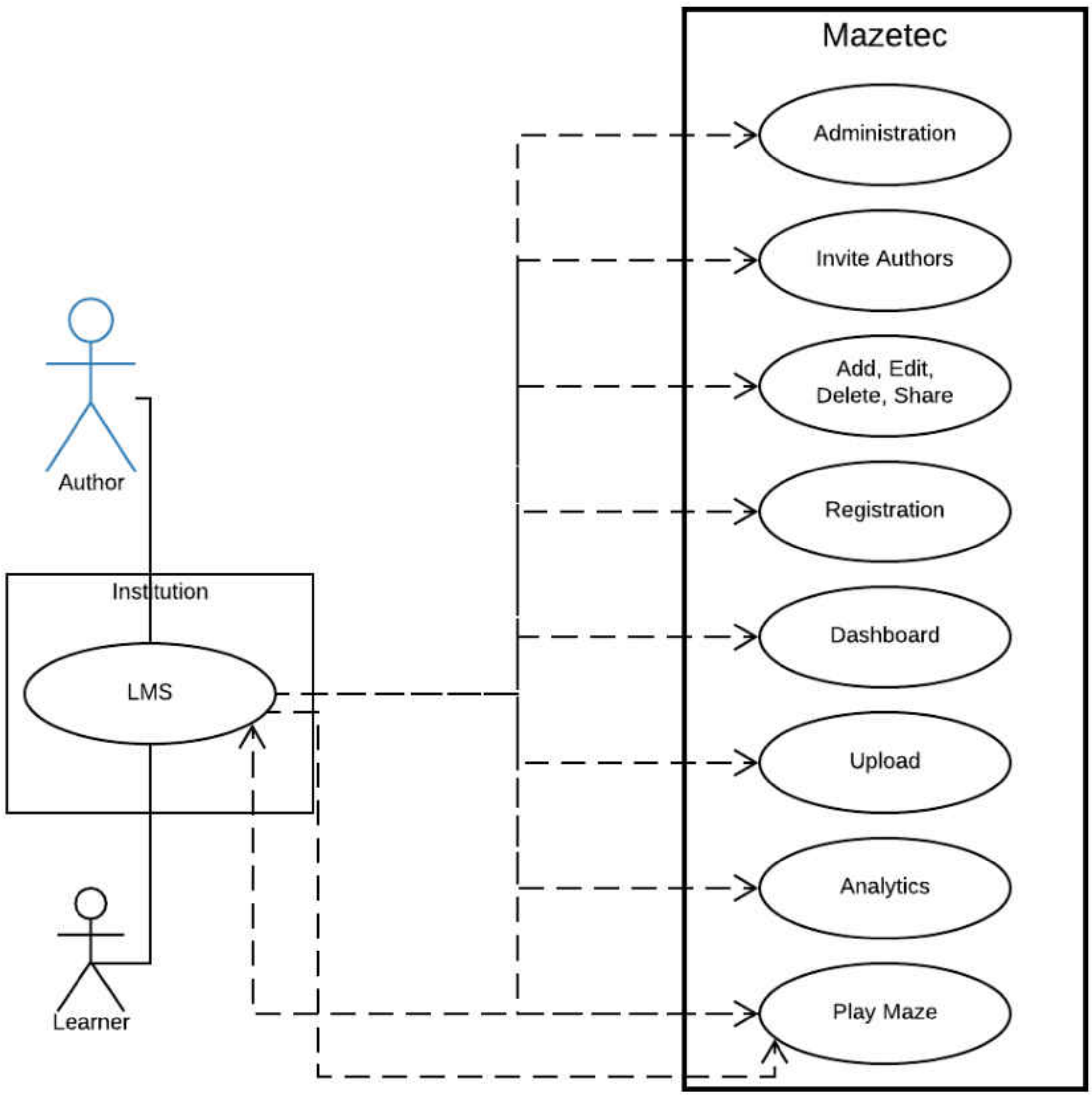

Figure 6. Use case 4.

Functional System Components

This section reviews the system functionality, presents a logical architecture which organizes the system functionalities, and explores the system data architecture, system security, and system interoperability and integration.

User Management

The scenario authors will need to use an authenticated user account. All user accounts will default to the author role, which will determine the default permissions. To support a multiple user account structure, the system will need the following functionality authentication: user registration, profile management, and permissions.

During the requirements analysis phase, we found most large organizations will require multiple users to test, integrate, and procure a single account. So, an author account must be associated with an organization. Knowing this, there are two cases in which a user can create an account. The first case is shown in Figure 3. A user navigates to the website, selects “create new account,” and is directed to a registration process where his user account is created, which triggers the system to create an organization entity associated with that author. In the second case, an author in an organization sends an account invitation by email containing a link with a single use, time-bound JavaScript Web Token (JWT) token that contains the inviting organization’s identifier. When the end-user clicks the link, she should be redirected to the account registration page, and the system uses the token data to create the relationship between the new author and the organization. Just as employees come and go, the organization lives on with the existing employees. Therefore, the organization should not be deleted when its first user is deleted or removed.

From the system perspective the purpose of the organization entity is to serve as a hub for an account. Account activities, such as billing integration, need to be at the hub, so the SMEs can focus on creating scenarios and analyzing the results. Figure 7 shows the organization use case.

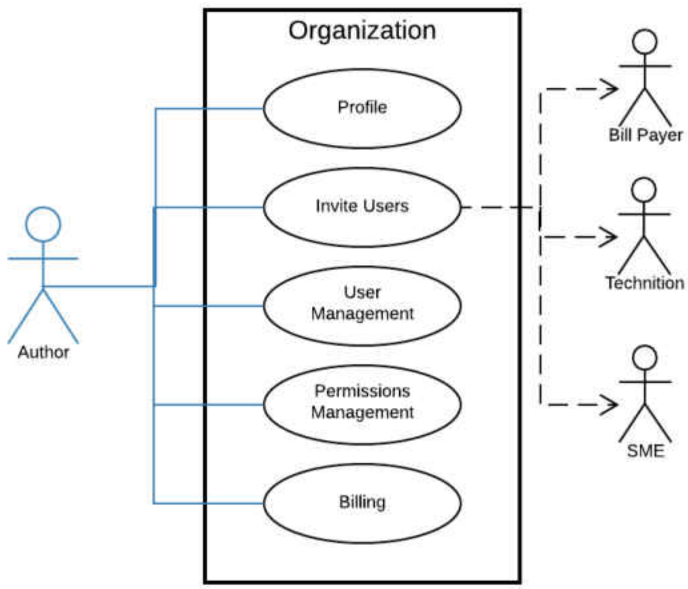

Figure 7. Organizational use case.

Figure 8 is a depiction of the user management features.

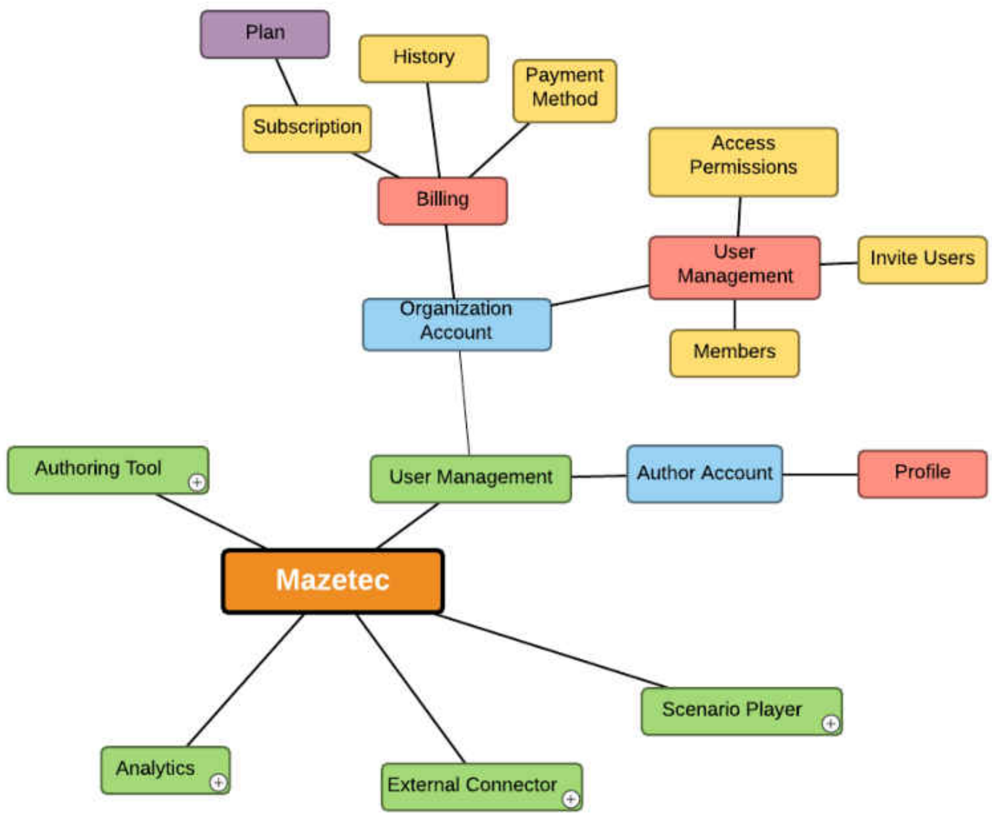

Figure 8. Feature map of the user management

Author Data Model

An author has one and only one organization, but an organization has one to many authors. An author on behalf of the organization may invite another author to join the organization account. Authors have permissions that determine their access within the account. For this project all authors will have the same level of access until a need arises. The organization has a subscription which is an instance of a plan. A subscription has a start and an end date, the price paid, and a plan id. A plan defines the number of author accounts the organization can have, the features the organization's authors have access to in the system, a price, a name, a description, and the billing frequency. The organization will have zero or many payment methods. They may also have no transaction history, if they do have a transaction history then they must have at least one to many transactions. Figure 9 is a high-level account entity relationship diagram illustrating these relationships.

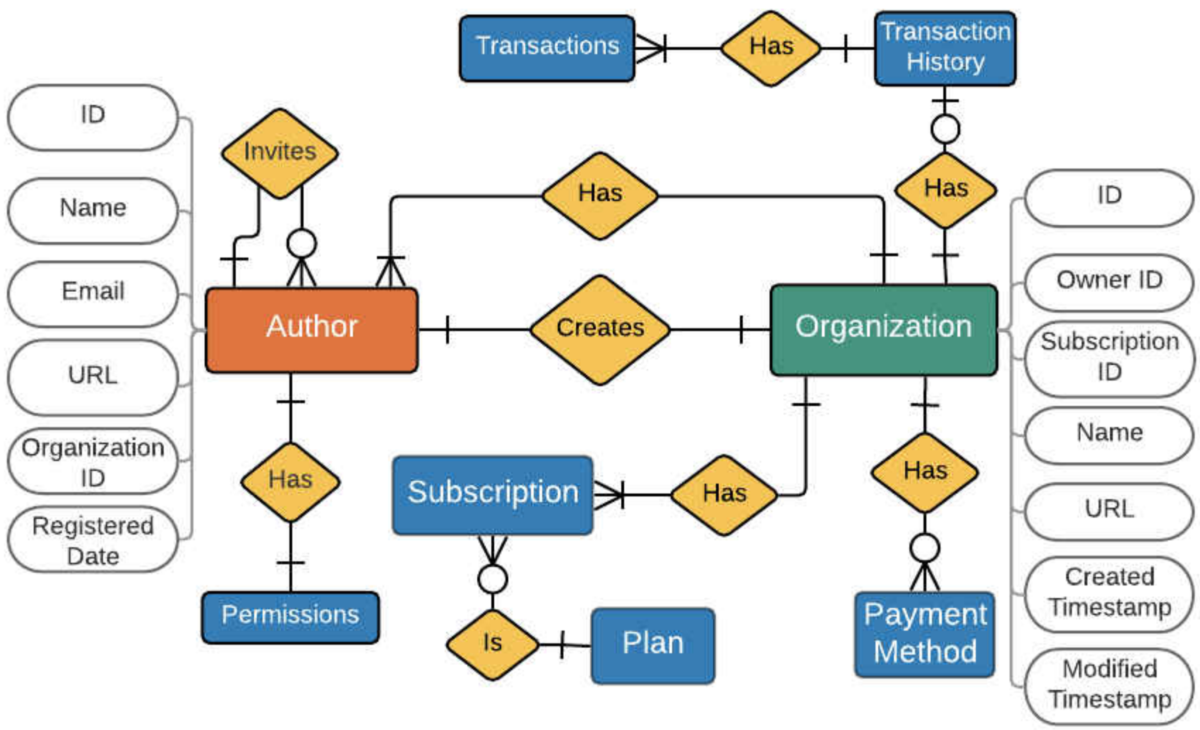

Figure 9. Author role entity-relationship (E-R) diagram

Learner Data Model

The learner's relationship with the system is much more direct than that of the author's role. The system may or may not know who a learner is, but the learner will still need an authorization token weather he is accessing a maze anonymously or through a credentialed system. If the learner is accessing a maze from another system, we may not know what data they are arriving with, but it is critical to know in order to be able to maintain system interoperability.

The Experience API (xAPI) protocol defines the possible identifiers that can anticipate the payload we will receive from other xAPI compliance systems. The xAPI protocol uses the term actor because it is more widely applicable. When a user from another system, as depicted in

Use Case 2 (Figure 4), per xAPI protocol, the incoming learner may be identified in one of three ways shown in Table 10.

Table 9. *xAPI Learner Identifiers and Payload.*

Identifier	**Payload**
Email	"actor": { "mbox": "mailto:db@mazetec.io", "mbox_sha1sum": "98sd098f8s0dfsd08f8df", "name": "Daniel Bietz" }
OpenID	"actor": { "openid": "http://dbietz.openid.mazetec.io", "name": "Daniel Bietz", "objectType": "Agent" }

External System e.g. university account	"actor": {
	"account": {
	"name": "db06146",
	"homePage": "http://georgiasouthern.edu"
	},
	"name": "Daniel Bietz",
	"objectType": "Agent" }

As you can see, there is still some variability with how the learner will be identified. The learner identified should be stored because the identifier data object will need to be joined with the learner's maze activity data later. Every action a learner performs will be stored as a statement and each statement will need to identify the learner. It is in our best interest to keep the data the same, so we can return it to the referring system as it was sent to Mazetec. If we use a relational database with the 3rd normal form, we would end up with many tables and many join operations but use the minimal amount of space. If we choose a noSQL database we could store each JSON document in a single collection, but we would be trading space to reduce complexity. However, joining the learner to their activity statements would be more computationally expensive than in a relational database. Examining the user needs, there isn't a need to decompose this further. We can achieve data atomicity by using PostgreSQL. PostgreSQL allows us to take a hybrid approach; it is both a relational database and a JSON document database. In figure 10, the learner table has an ID, a timestamp, and an object field. In the object field we can store the JSON documents.

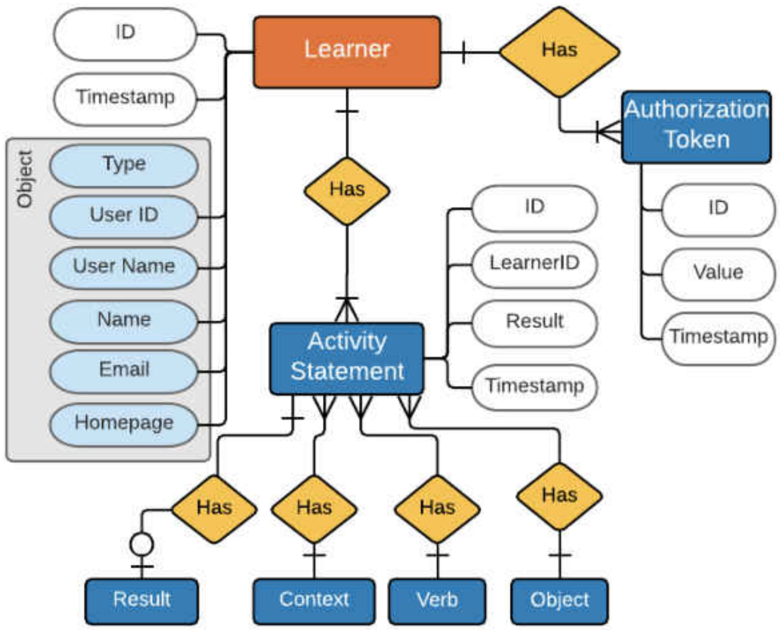

Figure 10. Learner role entity relationship diagram

User Authentication

User authentication will need to be accomplished OAuth 2.0 and OpenID using a Javascript Web Token (JWT). The authentication component will be used frequently throughout the system so this will be implemented as a service. Authors and external systems will need authenticate with the system before accessing data.

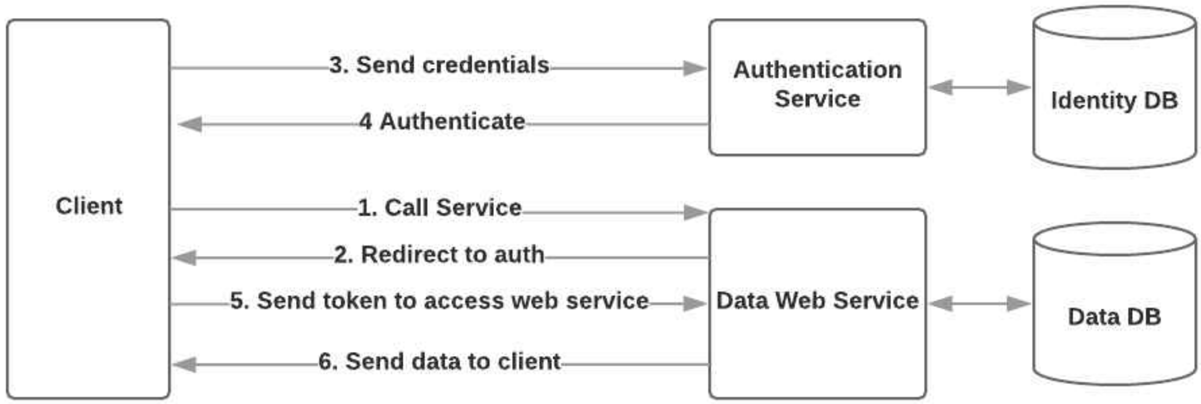

Figure 11. Authorization service

Authorization

Once a user or external system has authenticated, the entities permissions should be checked before granting them access; this is called “authorization.” This is accomplished by evaluating the roles/privileges associated with the current user and determining if the user is authorized to access the resource. Information on what roles/privileges are associated with a specific user is stored in the database.

Mazetec Authoring Tool

This section reviews the Mazetec scenario authoring tool. Once authenticated and logged into the system, the author will land on a dashboard panel showing a snapshot of his account activity such as number of mazes played. The site map (Figure 12) depicts how the authoring tool should be organized given all of the requirements so far.

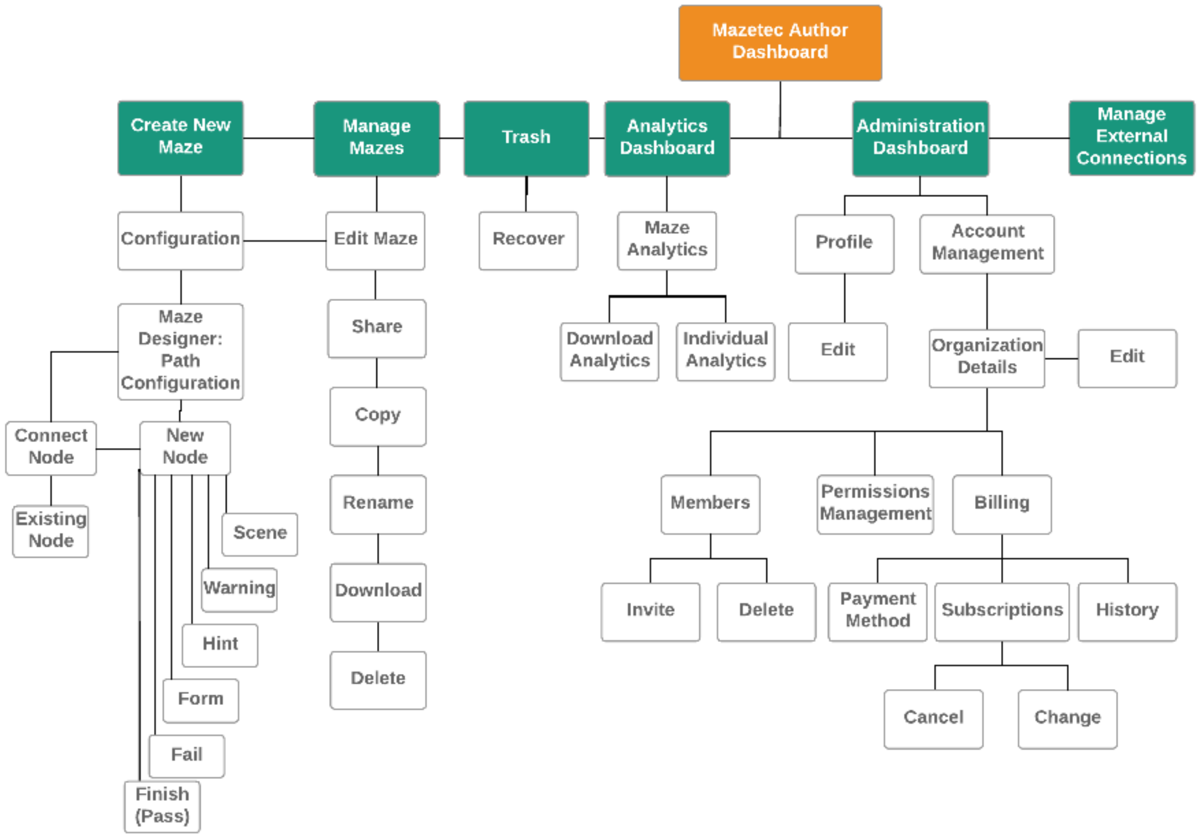

Figure 12. Site map.

Maze Designer - Scenario Authoring Component

The maze designer is the focus on the Mazetec system where the scenario authoring will take place. The component will need to create, configure, build, test, and preview scenarios. The feature map in Figure 13 helps to determine the components of the organizational structure.

Figure 13. A feature map of the scenario authoring tool.

The Authoring tool has three primary features. The scenario designer, configuration, node editor, and the preview function. Considering the most common access patterns, the author will start with the maze configuration, followed by the scenario design where they add, connect, and edit nodes, and then preview their maze with iterative back and forth between the designer and the previewer. We can logically group the designer, configuration and node editor into a large tightly coupled component, and the preview function will be an instance of the maze player component.

Maze Configuration

The configuration page should contain all of the global parameters outlined in Table 11 and behaviors. The scenario title and description are constants, but the time-limitation feature should be disabled by default and hide all of its configuring properties unless it is enabled. The configuration will also determine the available options in the node editor. For example, if the timer is enabled, the Hint type node should display an option to pause the timer, but if the timer is disabled this option should be hidden from the node editor. Given the limited number of configurations at this time, a single page should suffice.

Table 10. *Scenario Configuration Properties*

Maze Designer Configuration Fields	**Behavior**
Title	Displayed to user on start page
Description	Displayed to user on start page
Timer (Y/N)	Displays the timer if enabled
Time Limit (IF TIMER ENABLED)	Displays a popup modal and ends the scenario session when the timer reaches 0.
Penalty Time (IF TIMER ENABLED)	Subtracts time from Play timer, if a warning node is selected
Bonus Time (IF TIMER ENABLED)	Adds time if a scene is selected
Timeout Title (IF TIMER ENABLED)	Displayed title of timeout modal
Timeout Message (IF TIMER ENABLED)	Displayed message of timeout modal

Maze Designer

Once a scenario is configured, the author is ready to design. All scenarios must begin with a starter node of Scene type for the author to build upon. The node editor will allow the author to edit the node properties, and it should adapt properties of the node in question. After the first node is populated, the author will need to create another node and connect the nodes together.

Edge Connector subcomponent

The simplest solution is to connect the edge and create a new node at the same time. This can be visualized by expanding the starter node to reveal the options (edges). We preface each option with a "+" button to indicate to connect or add to the option. When the author clicks the "+" she should be presented with a pop-up modal to either create a new node or connect it to an existing node, so the nodes in a scenario can be reused. This will be the edge connector subcomponent of the maze designer.

Node Editor subcomponent

If the author selects "Create a new node," then logically the author should first determine the type of node (scene, hint, warning, form, fail, or pass) he wants to create. Knowing the type of node before launching the node editor allows us to hide unnecessary features and improves system usability. When the author selects the node type, the node editor subcomponent should adapt to determine the options and configurations that fit the selected type.

If the author chooses to connect the option to an existing node, then she should be presented with a list of existing nodes in the scenario that she can connect within the edge connector. Clicking a node should link the edge of the parent node to the selected child node, and the edge should be labeled with the option text. This approach gives the author the ability to create the node, add options and link those options with other nodes.

The maze designer authoring diagram (Figure 14) illustrates the workflow for this process.

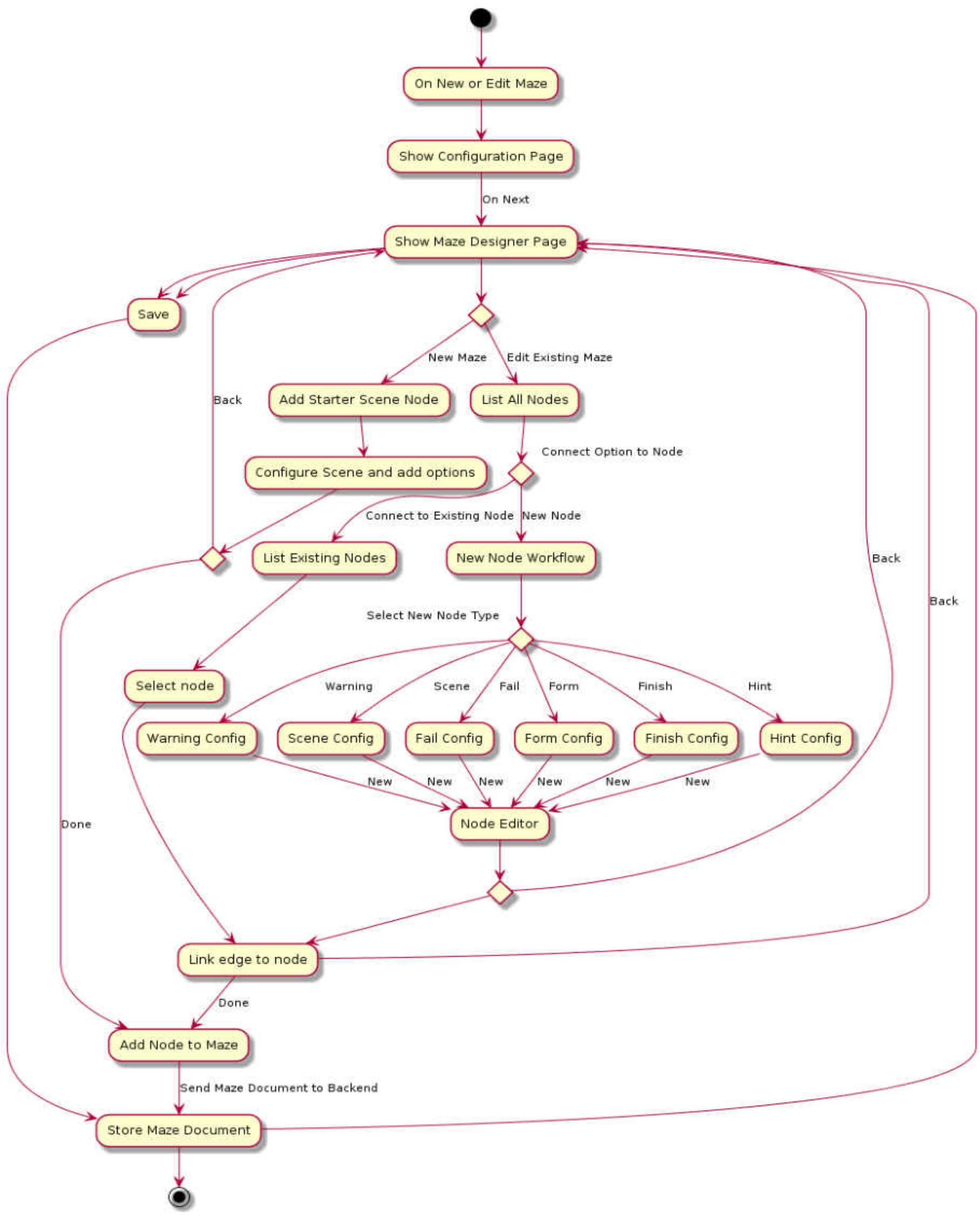

Figure 14. Maze authoring workflow

Maze Graph

As the nodes and edges are populated and connected to form the scenario, they are also creating a graph data structure of nodes (vertices) that are connected by directional edges. A node's options are its edges, the option text is the edge label. Every scenario has one and only one start node which is the graph's root node. The graph is labeled so that each vertex (node) is given a unique label set by the author. Vertices are not required to have a unique parent and can have a node that has more than one inbound edge. The order of a graph is determined by the number of nodes (V). The size of a graph is the number of directed edges (A). G=(V, A). The height a graph is the length of the largest downward path from the root node to the leaf.

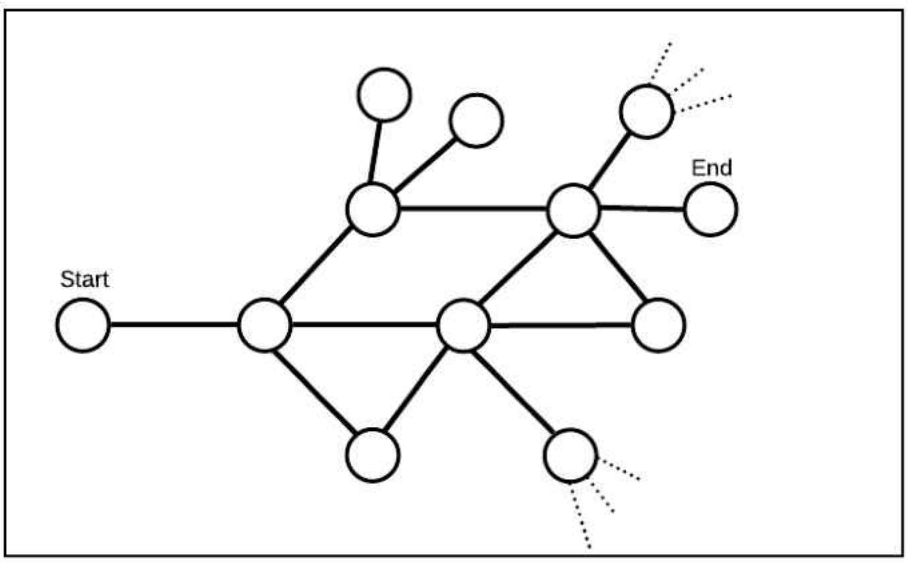

Figure 15. Depiction of a data graph.

Maze Nodes

The nodes represent contextual information, feedback, consequence start, or exit point that are experienced by the learner, which are called "path nodes." In the requirements analysis, we identified eight path node types. There are common properties and behaviors that all nodes will share such as a name and ID, and the type allows us to consider using polymorphism and the concept of inheritance in two different ways. In method A, there is a supertype and subtypes. For example, a supertype may be a node, and the subtype may be a scene. In this method there is no change to the original entity types, and it is mapped with the "is-a" one-to-one relationship. For example, a scene is-a node.

In method B, we consider removing the inheritance relationship in two ways. In B-1 the properties and attributes of the supertype can be moved to the subtype, and the supertype and inheritance relationship are removed, and a *type* property is added to the new standalone subtypes. For example, this would mean eliminating the node supertype and each node would have a class of its own.

In B-2, the properties and attributes of the subtype entity node can be rolled up in the supertype, which also eliminates the inheritance relationship and the subtypes. In this case, again, we also add the *type* property, but there is only the supertype with all possible attributes. For example, a node supertype *is-a* scene node and a warning node, a hint node, a fail node, and a Finish node.

Given the project is at such an early stage in development it may be advantageous to keep the design simplistic and duplicate some node properties and behaviors until we are certain the design is robust. However, it is certain that a node is a node and has an ID and a name. Each node below is a subtype will inherit from a supertype node, which has an ID, name, and type.

Start Node

The start node is the graph's root node. It inherits all of the inherits all of the properties of the scene node. The start node is automatically created when a new Maze is created.

Scene Node

The scene nodes drive the scenario forward. They provide contextual information and the choices that connect to other nodes. The scene node is broken down into two components: the scene itself and the choices it provides. The description field contains the contextual information displayed to the user. If the timer is enabled, they will have bonus and penalty values.

Each scene will also have a set of one or many options for the user to choose from. Each option will have the option text that is presented to the user, the ID linking the next node, and if the timer is enabled the author will be able to assign a bonus value.

Figure 16 depicts a simple data model for the scene node.

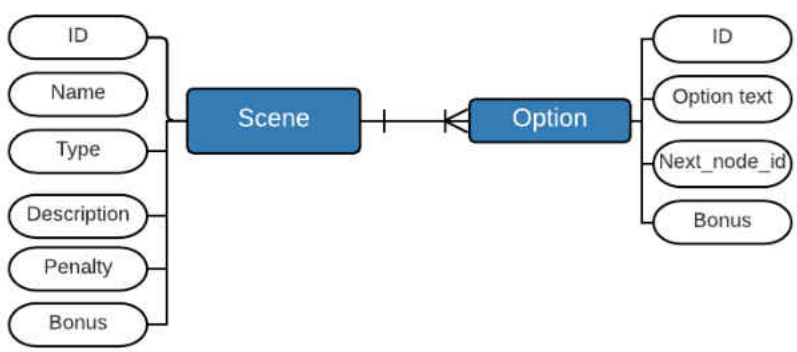

Figure 16. Scene node data model.

Warning Node

The warning node shown in figure 17 is of type "warning," has a title that is displayed, includes a description that displayed to the learner, and if timer is enabled it has an optional penalty time. The warning node is intended to be linked to wrong or poor choices. The warning node only has a back button that returns the learner to the previous node.

Figure 17. Warning node data model.

Fail Node

Fail nodes shown in figure 18 are for unacceptable or fatal choices during play; reaching this node immediately ends the scenario. The use case is in the medical professions, where some decisions may be fatal. There is no back option, only the option to restart the scenario.

Figure 18. Fail node data model.

Finish Node

Reaching the finish node (shown in figure 19) indicates a successful completion of the maze scenario. It ends the learner's session.

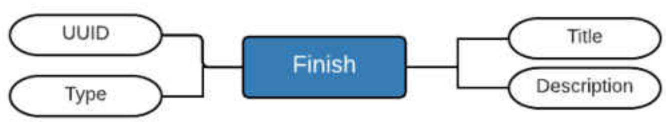

Figure 19. Finish node data model.

Hint Node

The hint node shown in figure 20 provides information to the player and includes an additional option to pause the timer, for example an author can configure this node to pause the timer while the player is viewing this node in a time-limited maze. The duration is captured in the user's play analytics. The timer is resumed when the player returns to the maze. This node includes a back button that returns the user to the previous node.

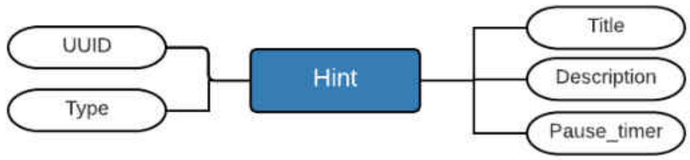

Figure 20. Hint node data model.

Maze Preview

The preview function is an instance of the maze player component. However, the implementation needs should allow the author to traverse the maze, viewing it as a student would on both a desktop and on a mobile device as highlighted in the requirements section.

Version Control

As authors update and change the configuration of the maze over time the system should automatically store the history of changes. This can be accomplished by saving the version difference or incremental history by reusing the “save” protocol in the maze designer component.

Maze Player Component

The maze player component is the interface by which the learner traverses a maze. The maze player needs to execute all of the behaviors defined in the maze so this component will naturally be tightly coupled with the maze designer component. Mazes are an author's intellectual property so regardless of a maze’s access permissions, the maze player should first authorize any user trying to access a maze. This will help provide protections against bots and anyone trying to access the maze without the link. When a link is shared it should contain a JWT token that authorizes the session. The activity diagram (Figure 21) shows how the maze player authorizes play.

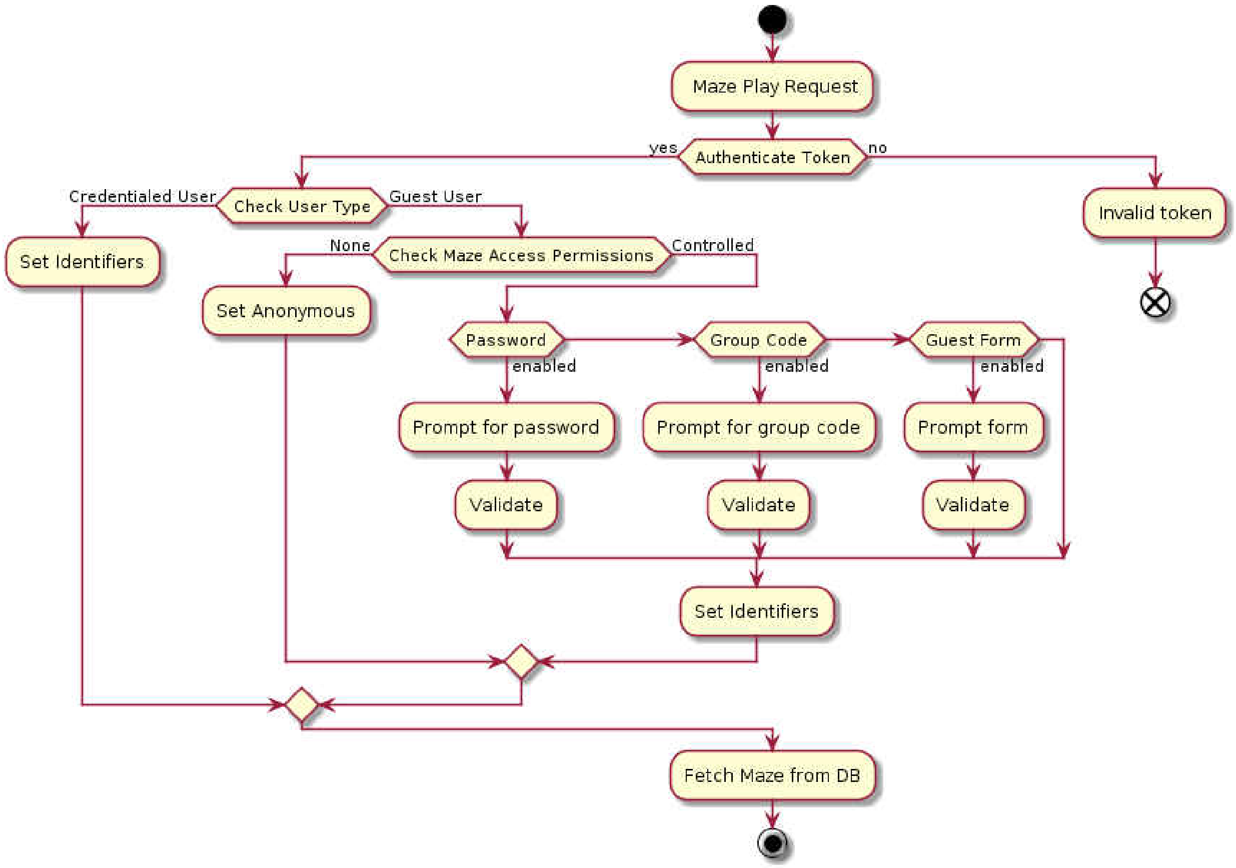

Figure 21. Maze player access activity diagram.

Once a session is authorized the maze must be able to traverse every aspect of the global scenario parameters, each node, and its unique configurations. Figure 22 depicts how the maze player traverses a maze.

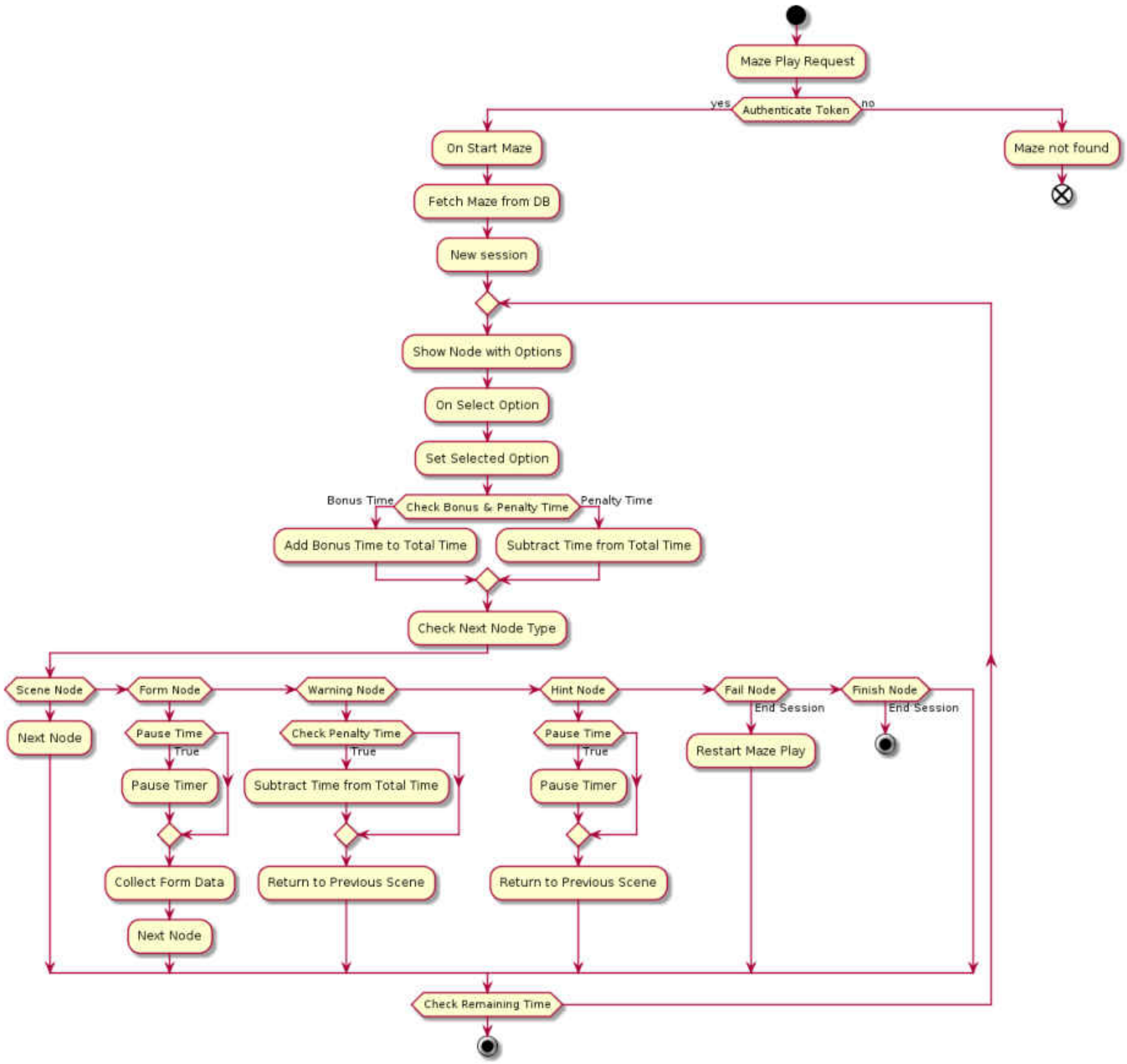

Figure 22. Maze player traversing a maze activity diagram.

Analytics

The goal of the learning analytics is to act as a data collection instrument for all users and make the data accessible to the admins, authors, and externally connected systems so the collected player data can be mapped to goals. The data can also provide a framework for processing, analyzing, and interpreting data streams generated from online learning experiences with respect to the learning goals. The usage Analytics should follow the xAPI format so less processing is needed to make the system interoperable. The xAPI data structure also happens to align with the data value 2.0 methodology.

Data vault alignment

- The data structure is source system agnostic
- Data loaded into other systems (e.g. LMS, LRS) is exactly as it is in the source system
- Business keys (Actor, Verb, Object) and their semantic meaning are defined by the xAPI protocol's data dictionaries
- It's recommended that data is staged in a non-production system such as a Learning Repository Store (LRS)
- Once activity statements are generated the data is immutable and resistant to change

Goals of the activity statements

- The statements are subject-oriented: Actor, learner, author
- Integrated: follows standard definitions and data structures of the xAPI protocol
- Time-variant: based on the time and date of the action
- Non-volatile: Activity statements are immutable
- Summarized: Activity statements can be summarized the system or exported to a BI tool

In table 11, the elements of an xAPI activity statement are described.

Table 11. *Elements of the Activity Statement*

Statement Element	**Definition**	**Value**
Actor - Required	An identity of the person who did the thing.	The user identifier (id or email), identifying the player that performed the event
Verb - Required	The action being done by the actor.	A selection, representing the player's choice type of interaction;
Object - Required	The thing the actor is acting on. This is normally an activity, but can also be a person, group or even another statement.	The option the user selected or the name of the node
Result - Optional	The outcome of the experience e.g. success, completion, score etc.	Pass or fail Activity time duration
Context - Required	The context of the experience, e.g. the larger learning activity this formed a part of, any other related activities, the instructor or team, the platform and language used in the experience.	The URL of the players location within the maze, maze name, platform name
Timestamp - Required	When the experience happened.	time and date

These statements will define and describe the actions the learners perform during the training. The will enable authors to plan, monitor, regulate, and evaluate their learning. Additionally, the xAPI statements will give the authors the ability to trace a learner's actions in sequential order and then can be summarized for the author. From these statements we should be able to produce the following summarized analytics listed in the table 12.

Table 12. *Showing the Analytic View and Data Captured*

Analytic	Details
Usage statistics for each Maze	Average completion time Average time per question Attempt count ● Fails ● Timeouts ● Completions ● Abandonments Excel Export
Individual Trainee Attempt Analytic Records	Date and time Visitor name or username if available Result Total scenes selected Total nodes viewed Average time per question Duration Duration answering questions For each node (question, warning, fail, or finish) Duration of time to answer each question If time restriction is enabled: Time remaining on the clock. Sequential Path Taken Bonuses Time (Seconds added to the timer) Penalties (Seconds subtracted from the timer)

The activity diagram (Figure 23) illustrates how the learner analytics are captured during a session.

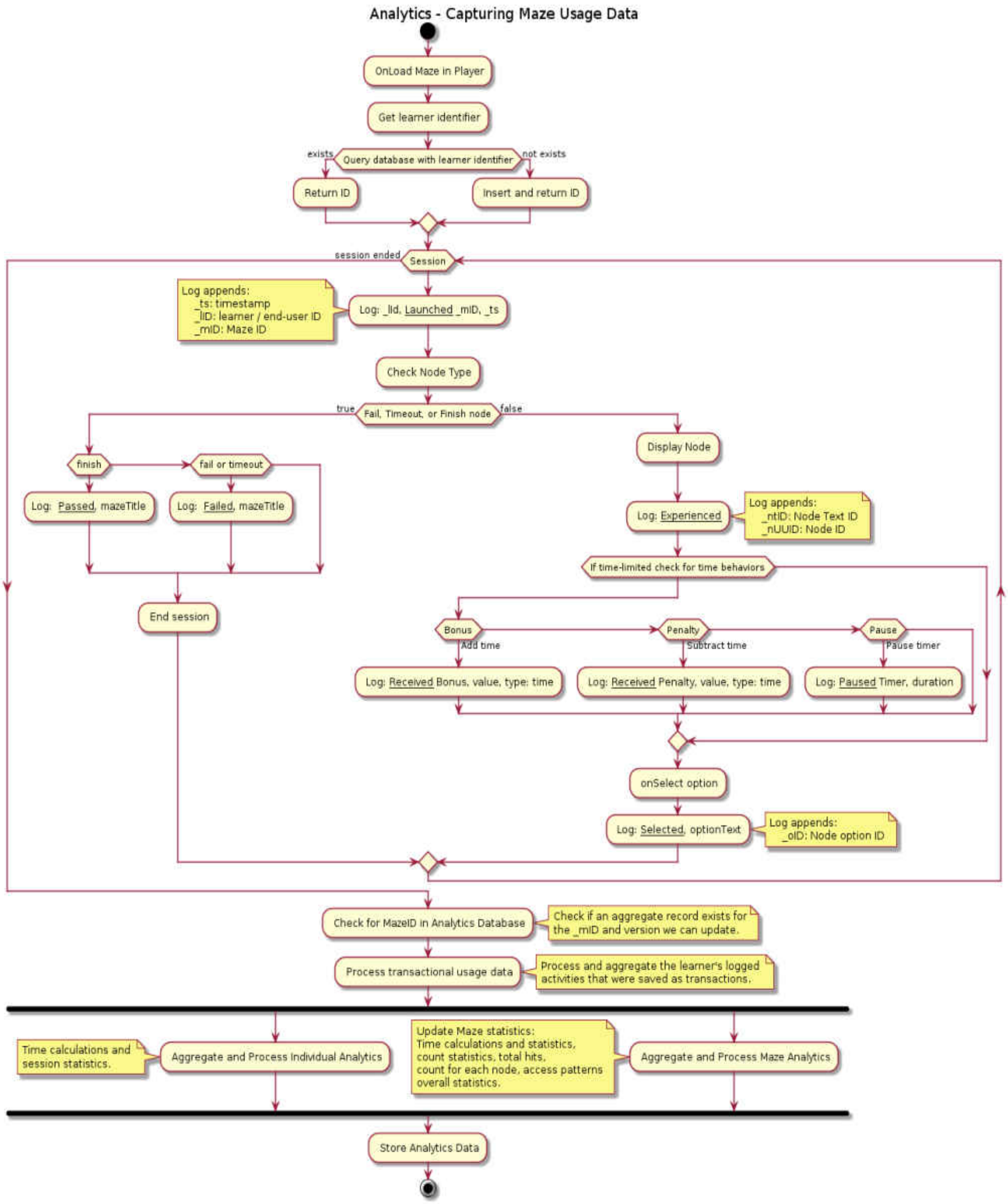

Figure 23. Showing the analytic view and data captured.

Logical Architecture

The logical architecture of the Mazetec platform consists of the author portal supporting external and internal users, functional modules supporting business functionality, common functional components and data repositories.

The Mazetec application will be built using a monolithic architecture with frontend, backend (REST + server page rendering), and database. Monolithic architecture will have multiple decoupled components. While one may argue these components maybe large and contain more than one domain or tightly couple with in single components, they are faster to develop for a prototype system.

The monolithic design can host its components on separate tiered servers or on a single server. It can also scale on demand with an API gateway and load balancer between multiple server instances for horizon load balancing and vertical scaling.

We can divide the system into to contain the system features at a high-level. There are three layers to the Mazetec application on a single tier.

- Frontend Web Application Layer – The web application layer contains the client-facing applications with which users interact.
- Backend (REST + server page rendering) – It is composed of several functional components.
- DB Data Layer – The data layer contains the databases.

Data Repositories

This section describes the data goals and design decisions. The most important aspect of the software is its ability to use time to create a feeling of pressure and track user actions overtime time with precision. In a time-limited maze, simulating a high-pressure rapid response scenario must not be interrupted by loading time during play. In order to achieve prevent loading delays during play, we must load the entire Maze before a user begins. Prefetching was considered to address this, the system would need to load all of the linked nodes in the background each time the user made a section in the scenario to prevent any load time, however, this would add complexity to the implementation and would also consume more computational cycles than loading the whole scenario at once into the client side memory. This is also one of the few special cases where querying denormalized data is faster than a query retrieving normalized data with many joins and it will be less taxing on the system.

Maze Data Model

The maze data model is a dependent structure, node options would not exist without a node, a questionset would not exist without a maze. If a Maze is deleted, the configuration, questionset, nodes and options should also be deleted. When a Maze is loaded into the player, all of it should be loaded to prevent loading time during play. This denormalized structure is a commitment to the precomputed joins.

The maze data access patterns are predictable, all of the maze data will be loaded before play or configuring to prevent any loading during play. Denormalizing the data model is a commitment to the predictable access pattern, and since the player will process much or all of it, precomputing the joins I think will reduce the system complexity and implementation. The data duplication and reduced update performance were found to be acceptable tradeoff to retrieve all of the information and its nodes in a single look-up query.

Question Set Maze Graph

The traversable scenario itself is stored in the embedded QuestionSet document. The data model looks like a tree, but it's actually a graph because it allows for recursion and parent-child relationships that are bi-directional. This is to account for the Hint and Warning nodes. Within the Maze Graph several relationship patterns are used. A maze has one QuestionSet. A questionset has many nodes. The model in figure 24 depicts the entity relationships in a a maze scenario.

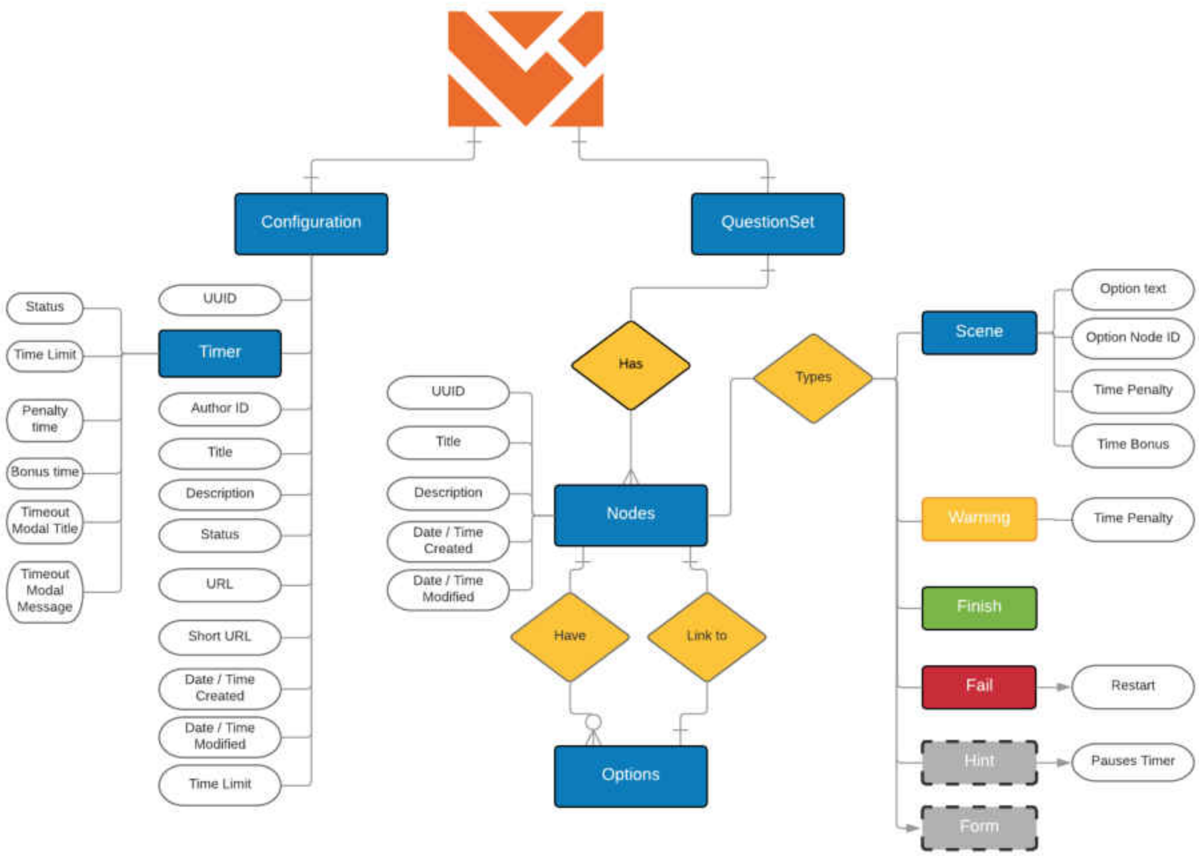

Figure 24. Maze ER data model in the maze graph "QuestionSet" collection.

Cardinality Constraints

Minimum or maximum number of nodes that can participate in an edge type. Since we cannot predict how the authors will design their domain specific knowledge graph we do not want to limit the number of nodes that can be involved in an edge type. For example, up and to this point in time there is no reason for us to constrain the possible connection a node can have to no fewer than 2 or no more than 5 nodes.

While there are many candidate database management systems in the market, MongoDB was selected for its ability to handle nested documents, data aggregation, and its support community. Research [33] comparing the to four NoSQL database management systems (Cassandra, MongoDB, OrientDB, and Redis) found the performance was uniform.

MongoDB uses a document-oriented database. These documents typically use a structure similar to JSON (JavaScript Object Notation). JSON is a human and machine-readable way to model data that is closely aligns with the class object structures in the application. MongoDB uses Binary JavaScript Object Notation (BSON). BSON is binary-coded serialization of JSON which enables faster data traversal, provides unique data types, and document encoding and decoding. When discussing MongoDB JSON and BSON are used interchangeably.

MongoDB doesn’t enforce consistency which is a benefit to the development of the application, however, it does allow for schema validation. This is particularly important because each maze will contain N nodes and M option with K relations forming a network graph. This adaptability is beneficial for modeling denormalized and polymorphic data during development and in production.

Maze Data

The recommended strategy by MongoDB is to embed parent-child relationships when both the parent and child will be loaded all at once at the same time to avoid the cost of an unnecessary JOIN, however, that has a trade-off of its own such as increased storage space from storing duplicate key values. The practice of embedding relationships is recommended only when the amount of data transferred in a read operation requires reading the whole data model at a time. Storing the maze data in an embedded document is a trade-off between space and complexity.

As discussed, the number of maze configuration attributes is static. The Maze Data Model uses an embedded document to describe a one-to-one relationship between the configuration and the Maze. This pattern is a recommended by MongoDB. The Configuration sub-document stores all for the predictable static Maze properties. The flexibility of the embedded document allows the application feature set to grow without significant changes to the configuration schema. The next chapter discussed the technical implementation of the Mazetec platform.

CHAPTER 5 – TECHNICAL IMPLEMENTATION

Overview

This chapter discusses technical implementation of the prototype Mazetec Scenario-Based Learning system. Hypothesis 1 asks if a scenario-based learning (SBL) system can be engineered. The prototype was built to focus on the core SBL authoring features and additional features to develop a minimum viable product to test the remaining hypothesis. The Mazetec SBL prototype system presented here consists of the scenario authoring tool, the scenario player, usage analytics and provides a multi account structure.

Implementation

The Mazetec SBL application was developed with a variety of frameworks. The client side of the application is based on AngularJS frameworks, JavaScript, PHP, JavaScript Object Notation (JSON), HTML5, and CSS3. The backend was developed with PHP and the Laravel framework. The application database uses MongoDB for the maze and usage analytics database because of its simple integration and use of JSON and the Laravel framework uses Redis database for caching. Figure 25 depicts the prototype system’s architecture.

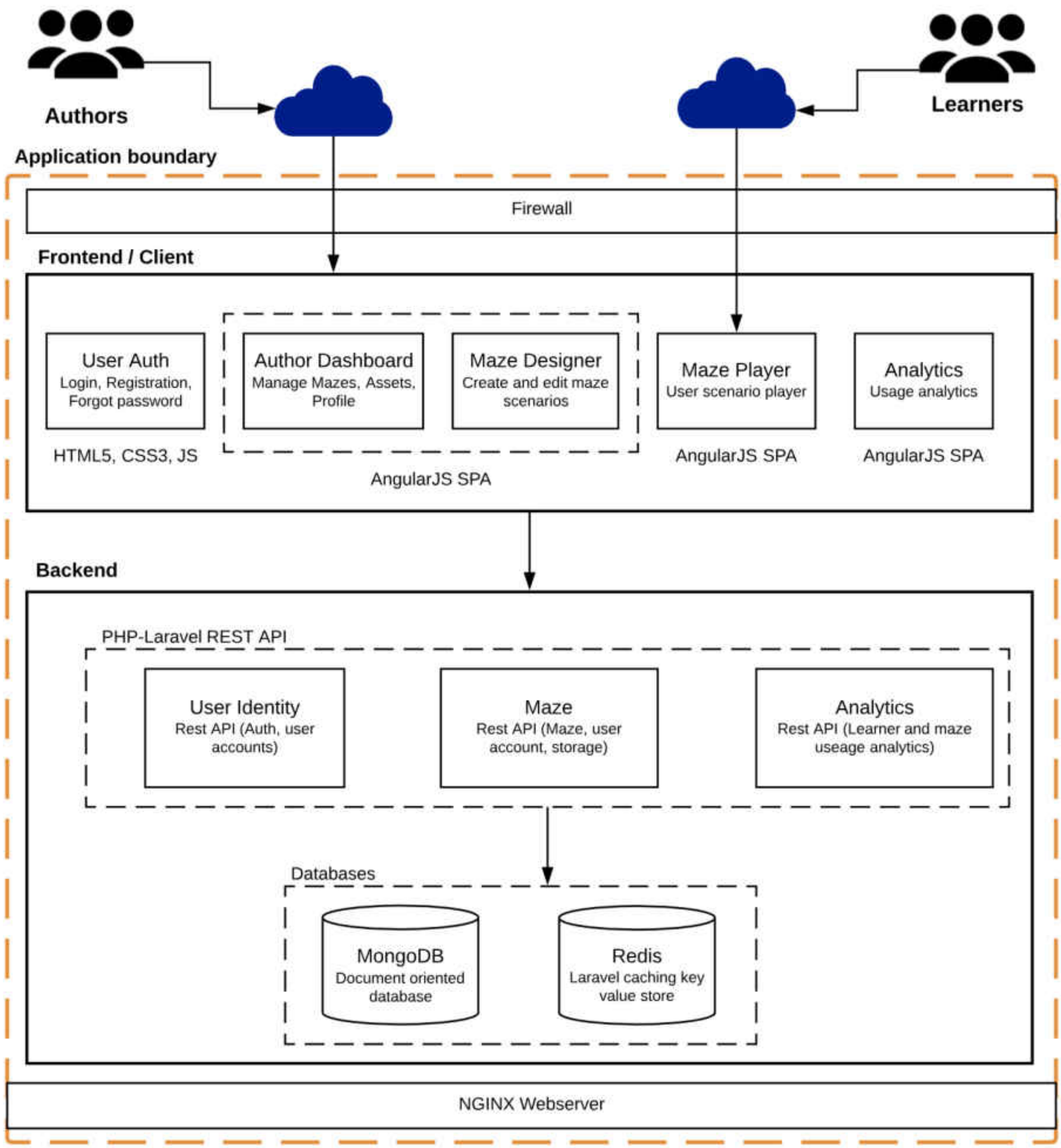

Figure 25. Mazetec architecture.

The objective of the prototype architecture is to allow the system users (authors and learners) access the application over internet. When the user navigates to the application website their request is first checked by a basic firewall limiting access to ports 80 and 443, then passed along to the nginx webserver. The user's browser only has access to the frontend client side of the application. When a user selects an option on the frontend, the request is passed from the frontend (AngularJS) to the backend (Laravel) where it is processed using AJAX (asynchronous JavaScript). The databases are only accessed through the backend using prepared query statements, this security measure helps to mitigate common data breaches. When Laravel queries MongoDB, MongoDB will validate the query and checks whether the ID exists in database if yes it will return the data. The backend may then perform additional processing and then returns the AJAX request back to the client side of the application. The client-side application then updated the view that is displayed to the user.

System Components

As discussed previously the system uses a monolithic architecture. The frontend client facing components of the Mazetec system were organized into five components User Authentication, Author Dashboard, Maze Editor, Maze Player, and Analytics. The application backend was built in PHP with the Laravel framework. The backend contains three main components User Identity, Maze, and Analytics components. The prototype has two databases MongoDB and Redis. Redis is required by the Laravel framework for general application caching. MongoDB is used as the system's primary database and consists of nine data collections.

Frontend Layer

All of the frontend components except the user authentication component are tightly coupled components built as AngularJS Single Page Application (SPA). SPA allows us to load all of the initial HTML upfront opposed to full page reloads, this reduces the applications data usage and speeds up the performance of the application because only JSON data is be requested from the server rather than all of the HTML, CSS, and JavaScript. Another advantage of using the SPA is that the HTML rendering process is performed on the client side by the user's browser thereby reducing additional computational processing load on the server.

User Authentication Component

The frontend User Authentication component contains the user registration, forgot password, and system login. This base authentication functionality is provided by the Laravel framework. Figure 26 shows the implementation of the user login view of the user authentication component.

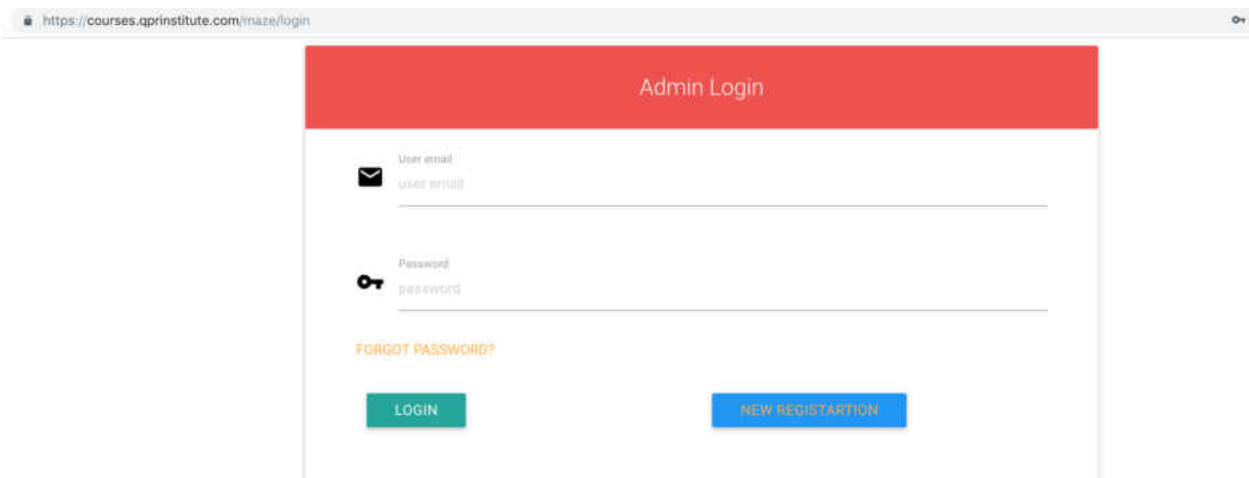

Figure 26. User login view.

Figure 27 shows the implementation of the user registration view of the user authentication component.

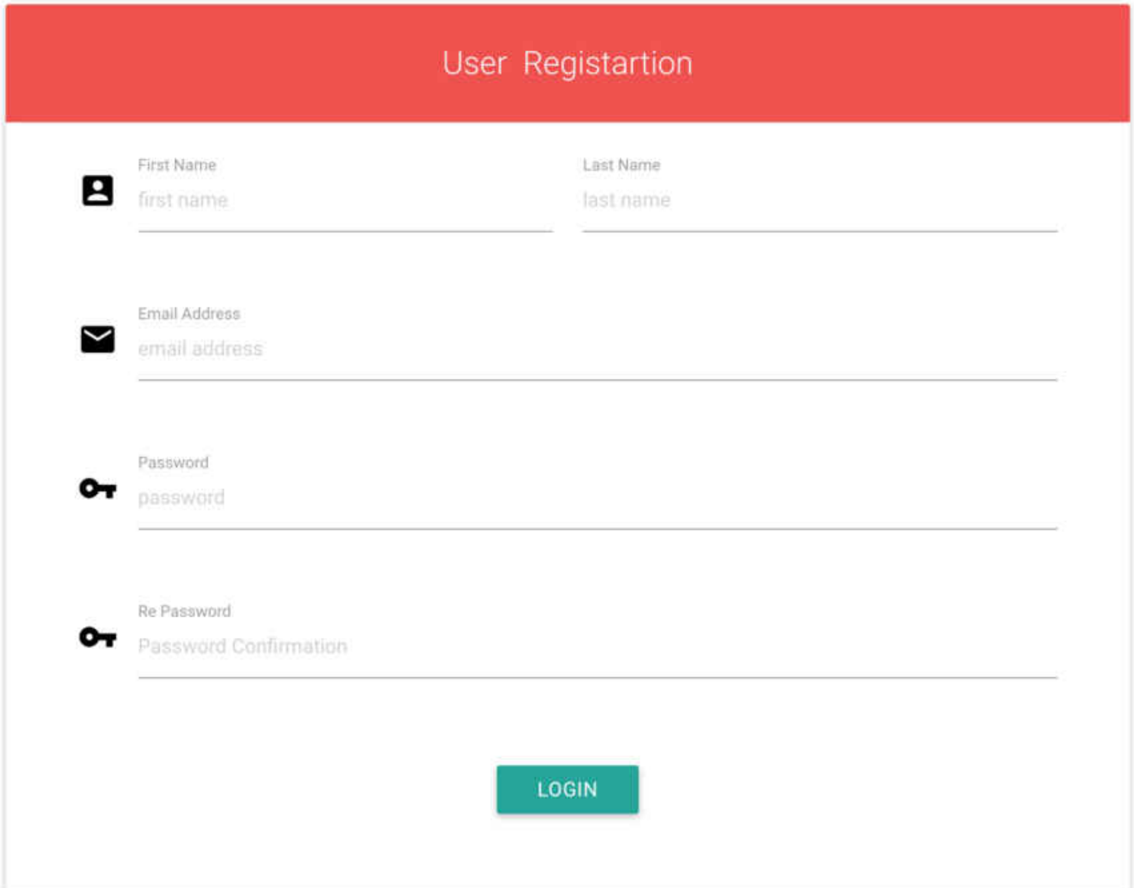

Figure 27. User registration view.

Author Dashboard and Maze Designer Component

The Dashboard component is the Author Portal that contains the user profile management and maze manager. The user profile manager allows an authenticated user to change the password, email, and manage their profile. The maze manager provides a create new maze button and lists all of the scenarios and displays buttons for each list item to play, edit, rename, share, delete, and download the scenario. The new maze and the edit maze functionality both launch the maze designer component. The Maze designer component contains the maze configuration page, maze designer, previewer, node editor, and node connector. There is also a node component and contains all of the subtypes which are globally tagged and shared across the platform SPAs. Figure 28 shows the implementation of the author dashboard.

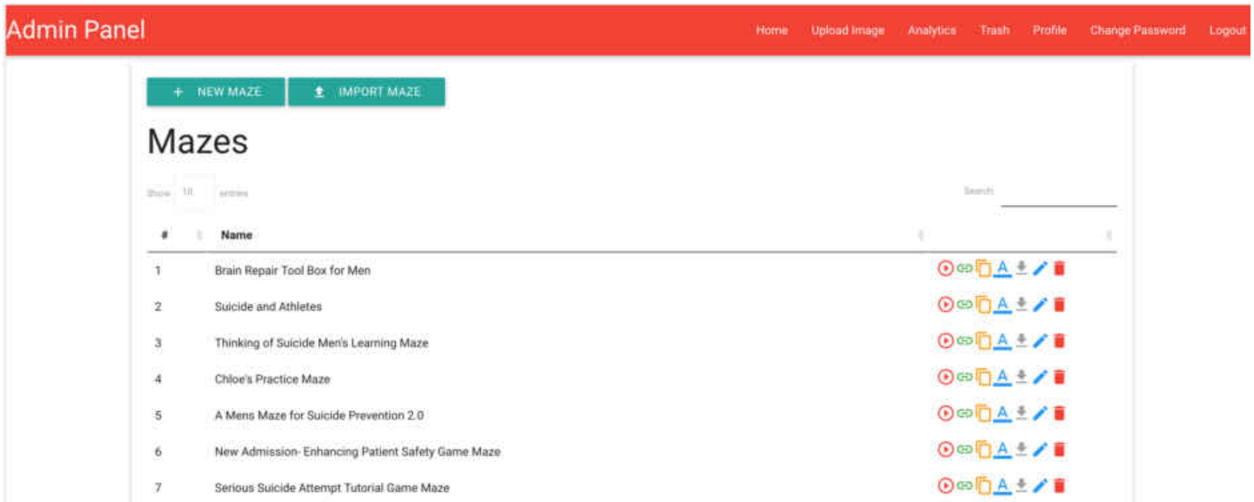

Figure 28. Author dashboard Mazetec prototype system.

Figure 29 shows the implementation of the maze configuration page of the maze designer component.

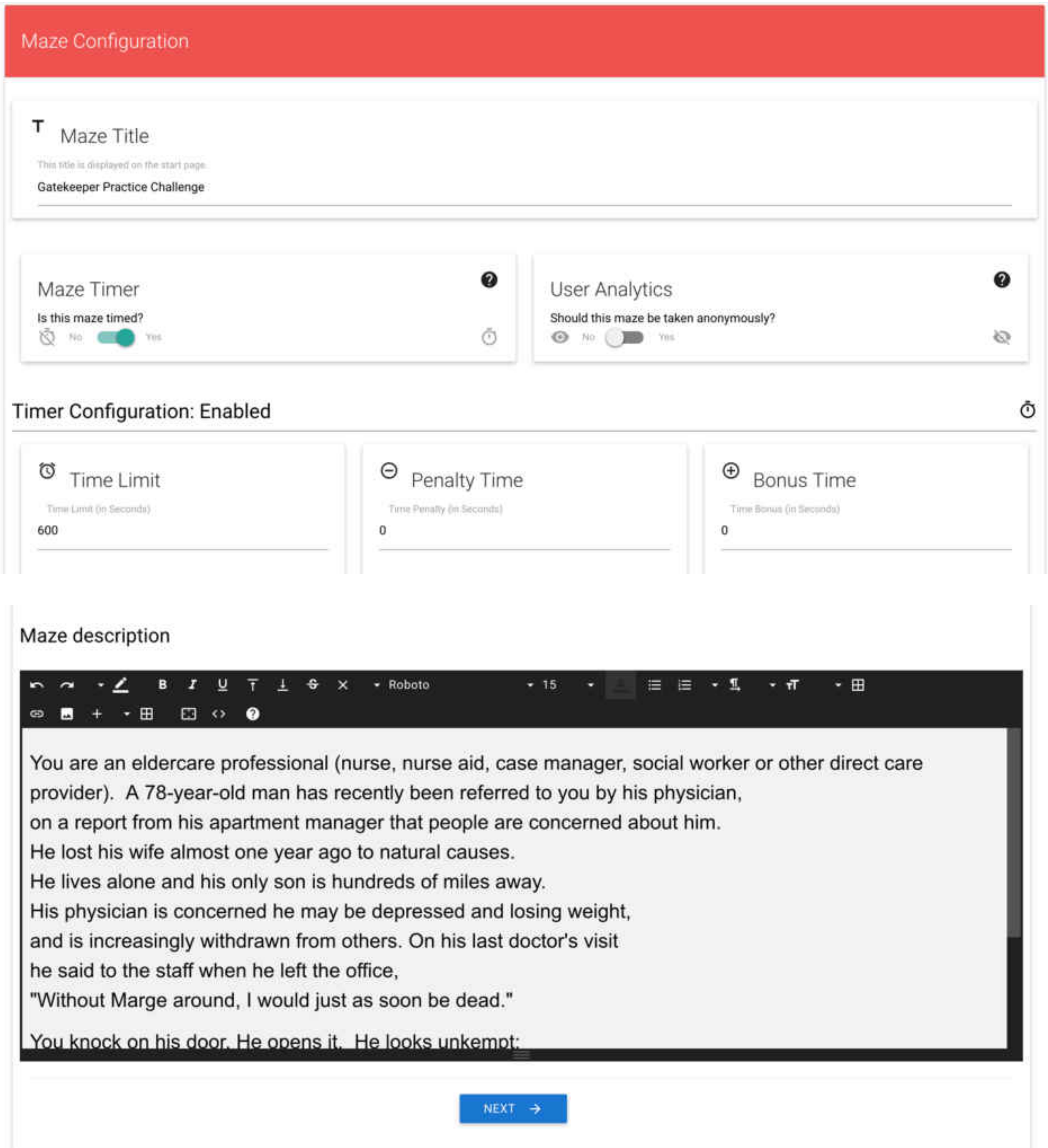

Figure 29. Maze configuration page.

Figure 30 shows the initial maze designer view with one start scene node. The user can click the pencil icon on the right to edit the start scene node as shown in figure 31.

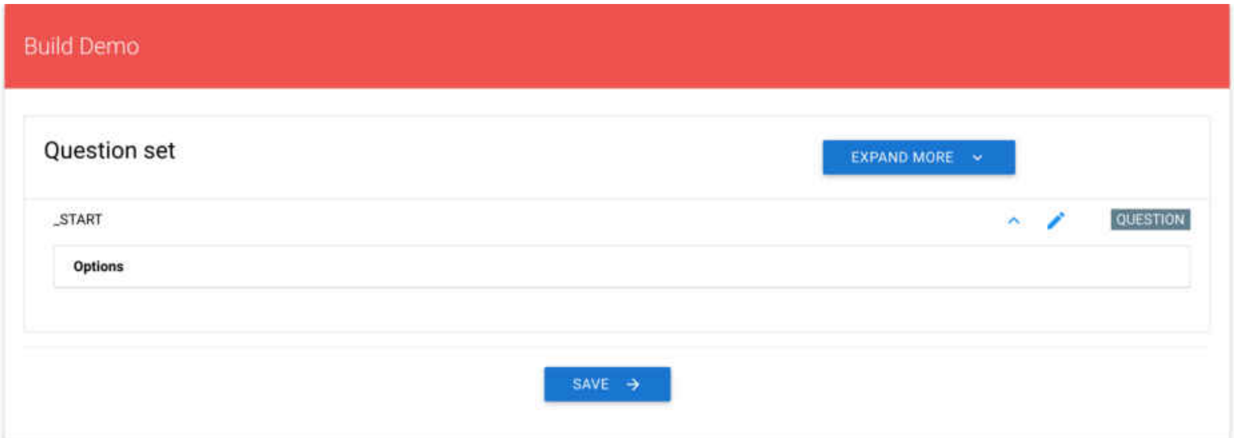

Figure 30. Maze Designer with starter scene node.

Figure 31 shows the node editor editing the start node in the maze designer component. Scene text and four options were added to this node.

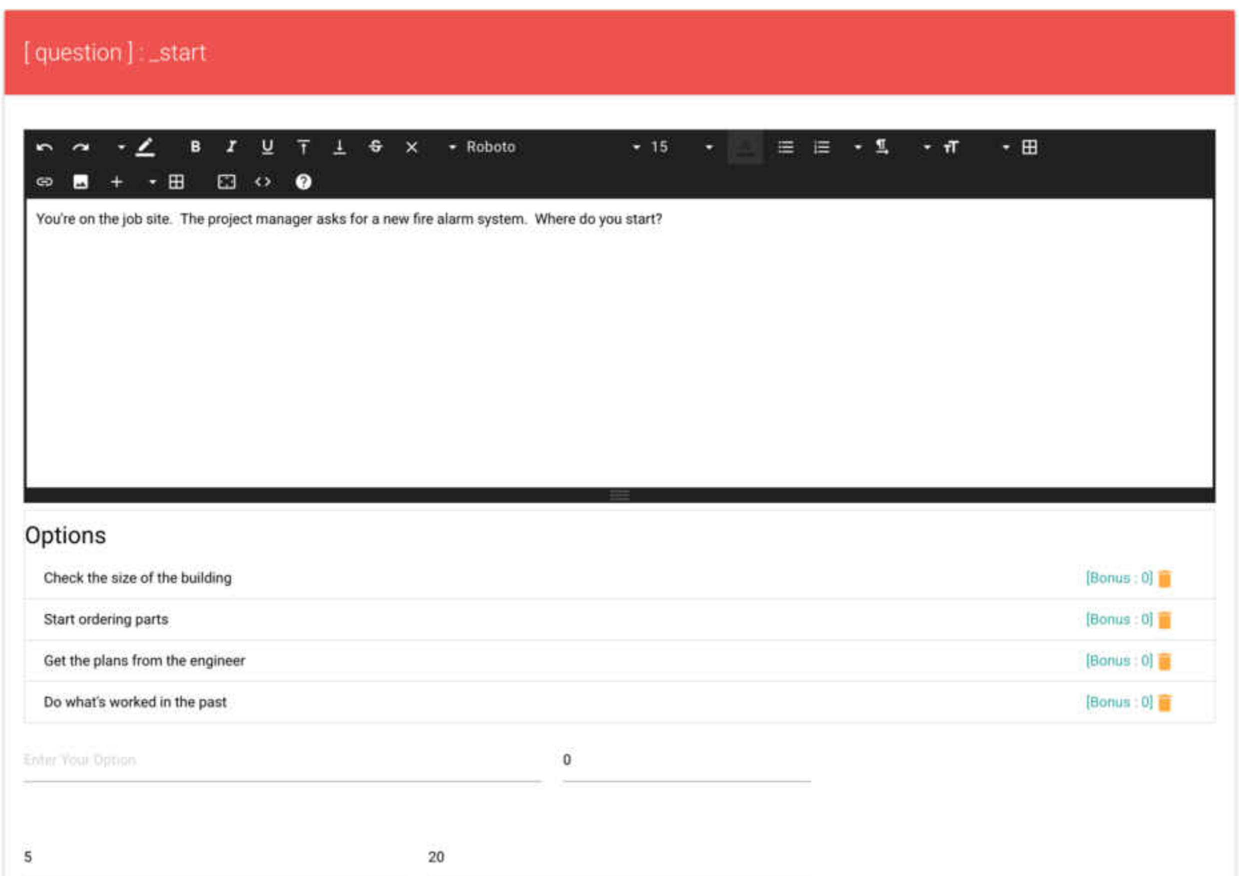

Figure 31. Node Editor – Scene.

Figure 32 shows the view of the maze designer after the start node is populated and expanded. As discussed previously, a user links the edges of nodes together by clicking the "+" and either creating a new node or connecting to an existing node with the node connector shown in figure 33.

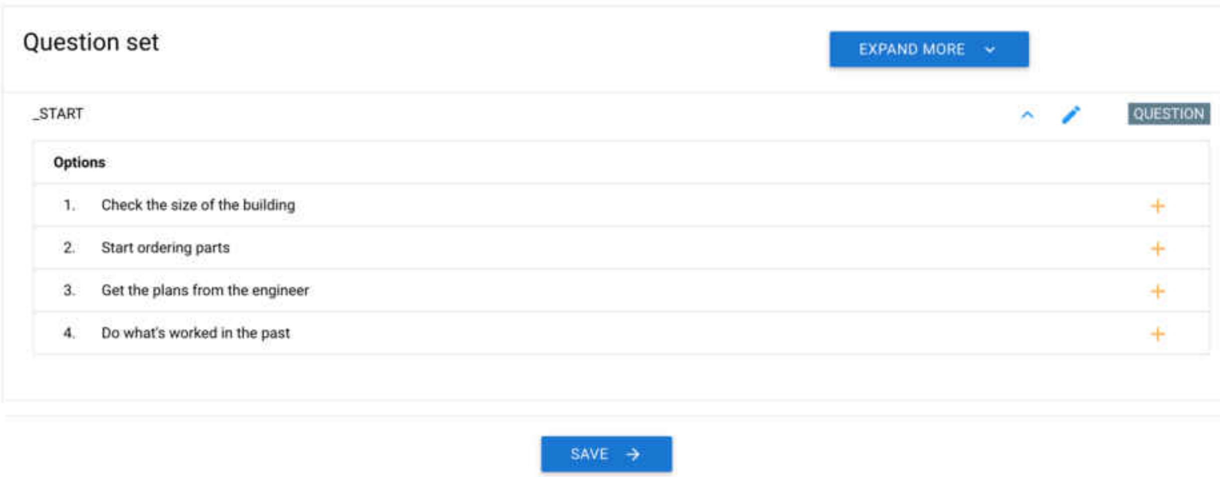

Figure 32. Maze Designer with the start node configured with options.

Figure 33 shows the first step of the node connector when a user clicks the "+" the node connector component is called.

Figure 33. Node connector wizard step 1.

Figure 34 shows step 2 of the node connector wizard displaying a dropdown selector of the existing nodes or the option to create a new node.

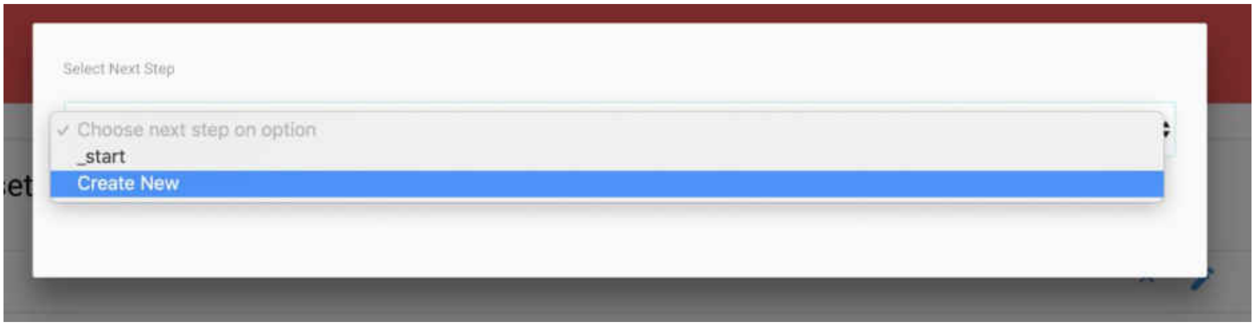

Figure 34. Node connector wizard step 2

Figure 35 shows the node connector component after the option to create a new node was selected. The user is then prompted to select the type of node they want to create and connect to the option they selected. The node type "question" has been changed to "scene" and "dead" has been changed to "fail".

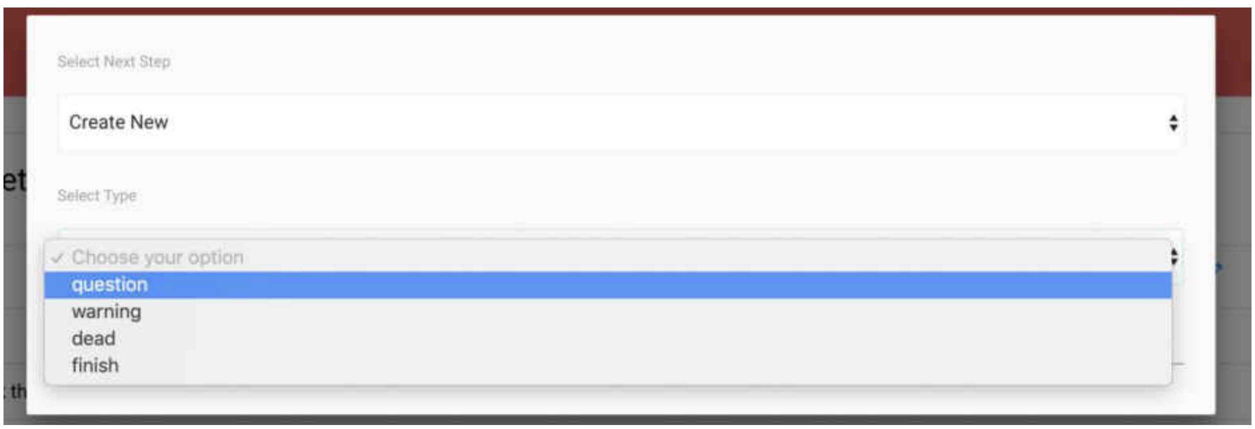

Figure 35. Node connector wizard, create new node, choose node type.

Figure 36 shows the maze designer view with seven nodes of different types.

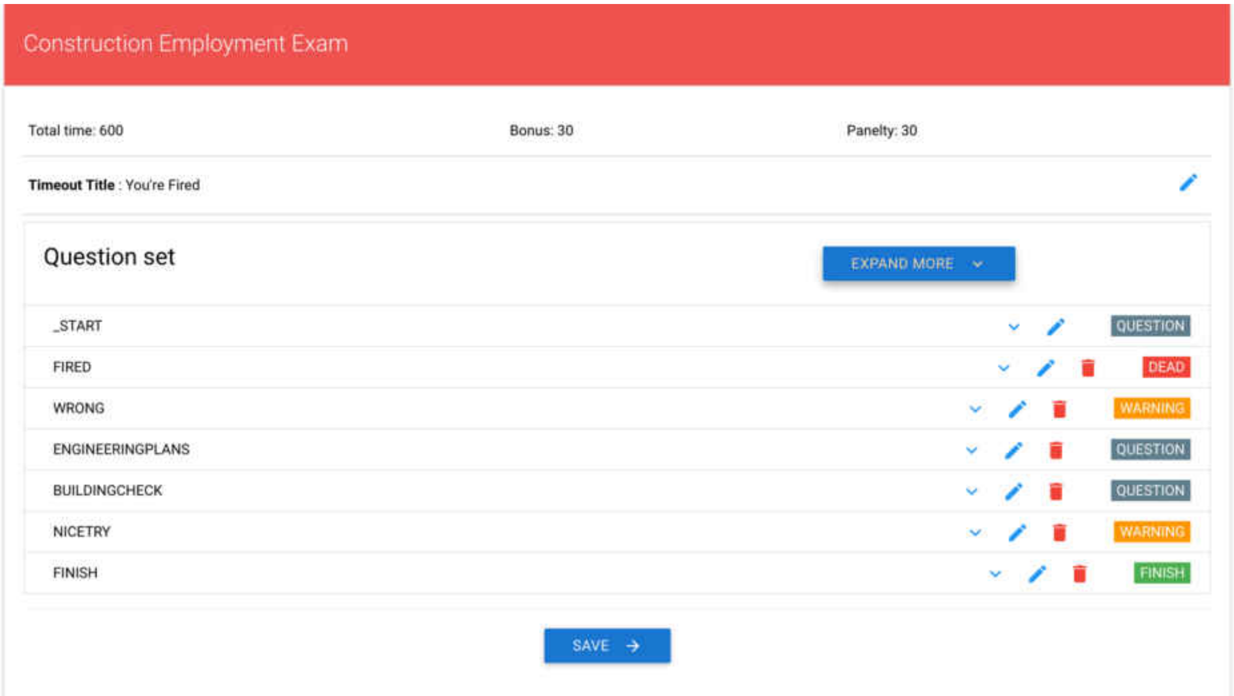

Figure 36. A view many nodes in a maze.

Figure 37 shows the expanded view of each node which details how the nodes are connected together. To the right of the option text where the "+" was previously is the tag for the node the option is connected to. The color of the tag indicates the type of node.

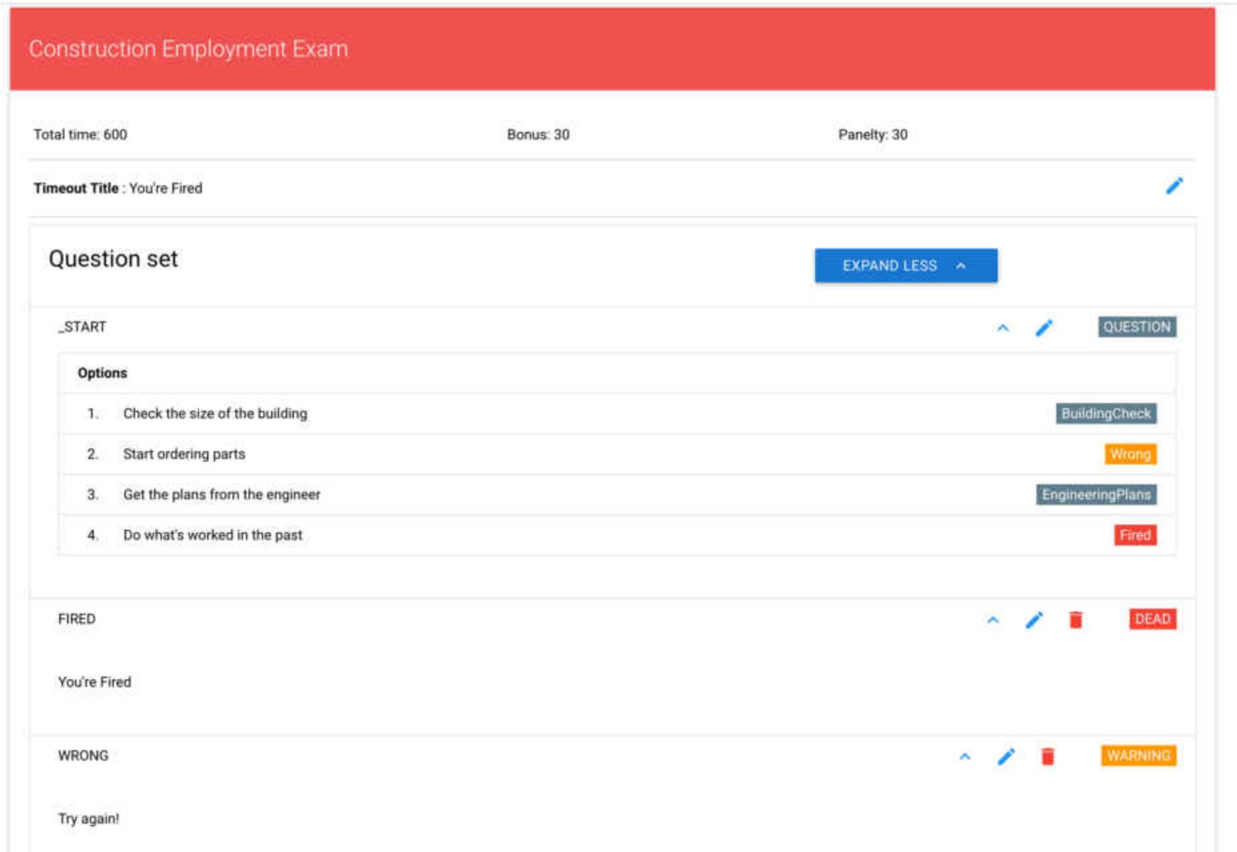

Figure 37. A completed maze, showing node options visually reference connected nodes.

Maze Player Component

The Maze Player component is a standalone SPA that allows the user and authors to navigate the scenarios. The player component leverages the node component and also provides the preview function within the Maze Designer. As users traverse a maze the player also generates the usages data which is processed and stored which is also consumed by the analytics.

Figure 38 shows the maze player component with a construction exam. The title and contextual information are defined in the configuration. At this point in time the entire maze is loaded into the browser's memory. The "start" button traverses to the first node.

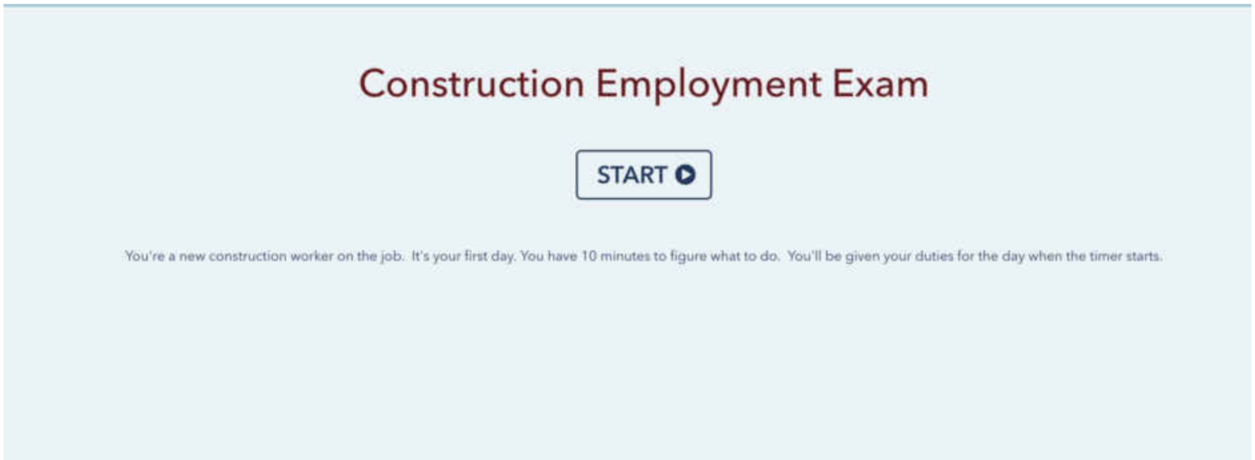

Figure 38. Maze player start page.

Figure 39 shows the start node in the maze, this is a scene node. At the top there is a body field that sets the scene and the options are displayed below.

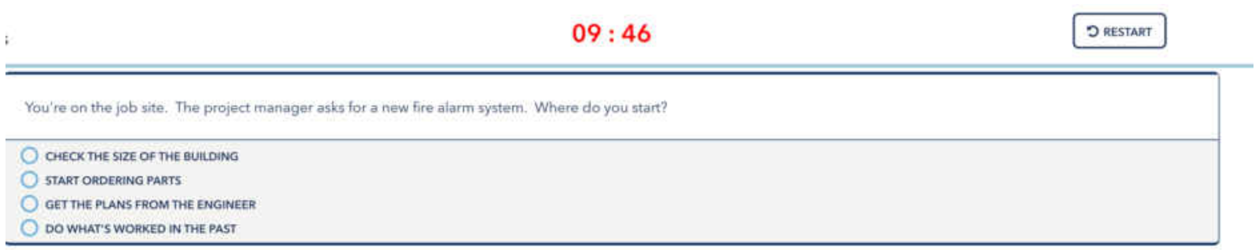

Figure 39. Scene node displayed in the maze player.

Figure 40 shows a warning node with a message and a back button. This warning node subtracted 30 seconds from the timer shown in red above the node.

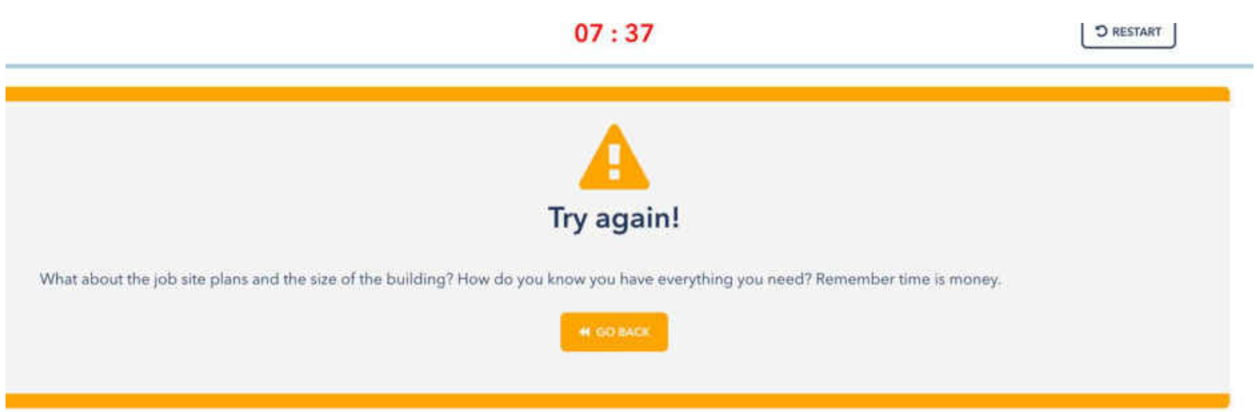

Figure 40. Warning node displayed in the maze player.

Analytics Component

The analytics component is a standalone SPA designed to pre-process the usage analytics after a session to update the aggregate maze analytics and also display the usage analytics by maze and individual within a maze. It reuses the node component. The analytics pre-processes the average finish time, number of attempts by completion type, and average number of scenes and it displays each individual attempt. For each individual attempt it shows the user's history of decisions within a single session.

Figure 41 displays the list of maze's by maze title, last modified date, and the version number. The turquoise button on the right links the aggregated analytics page for the selected maze.

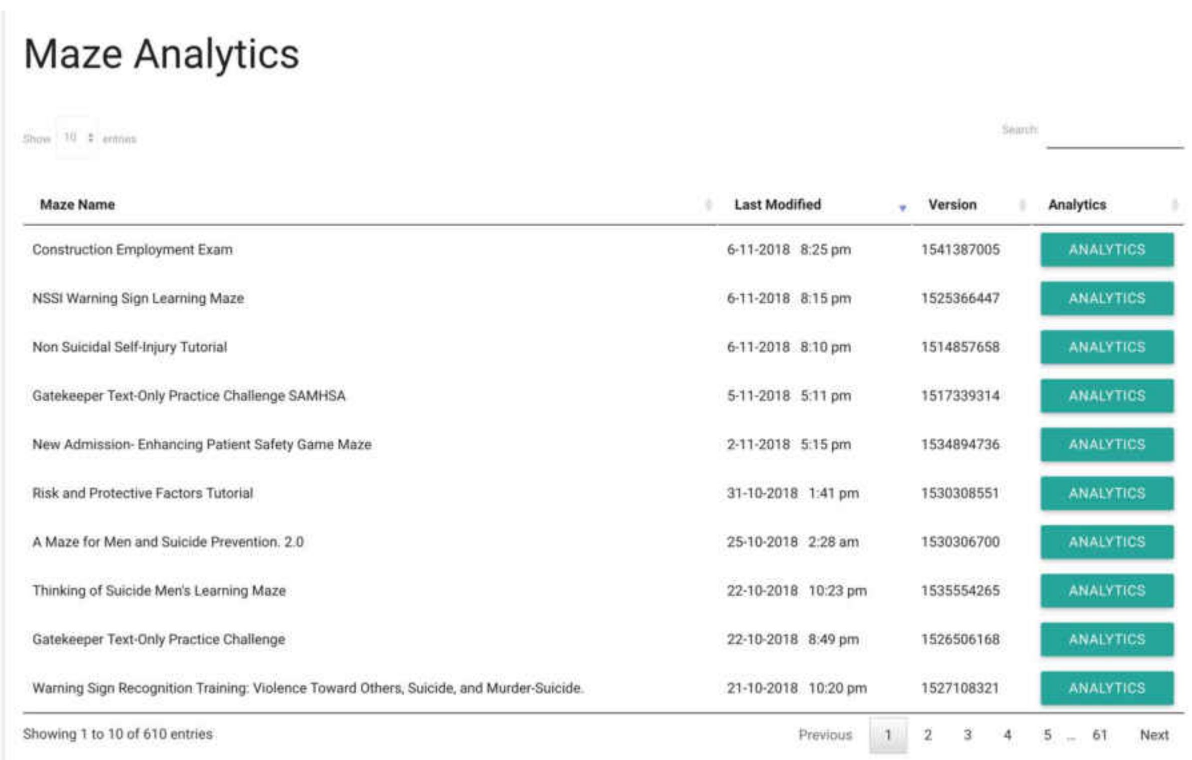

Figure 41. Showing the maze analytics list view.

Figure 42 shows the aggregated analytics view of a single maze and a summary of usage statistics. At the bottom of the page individual sessions are listed with a few detailed fields.

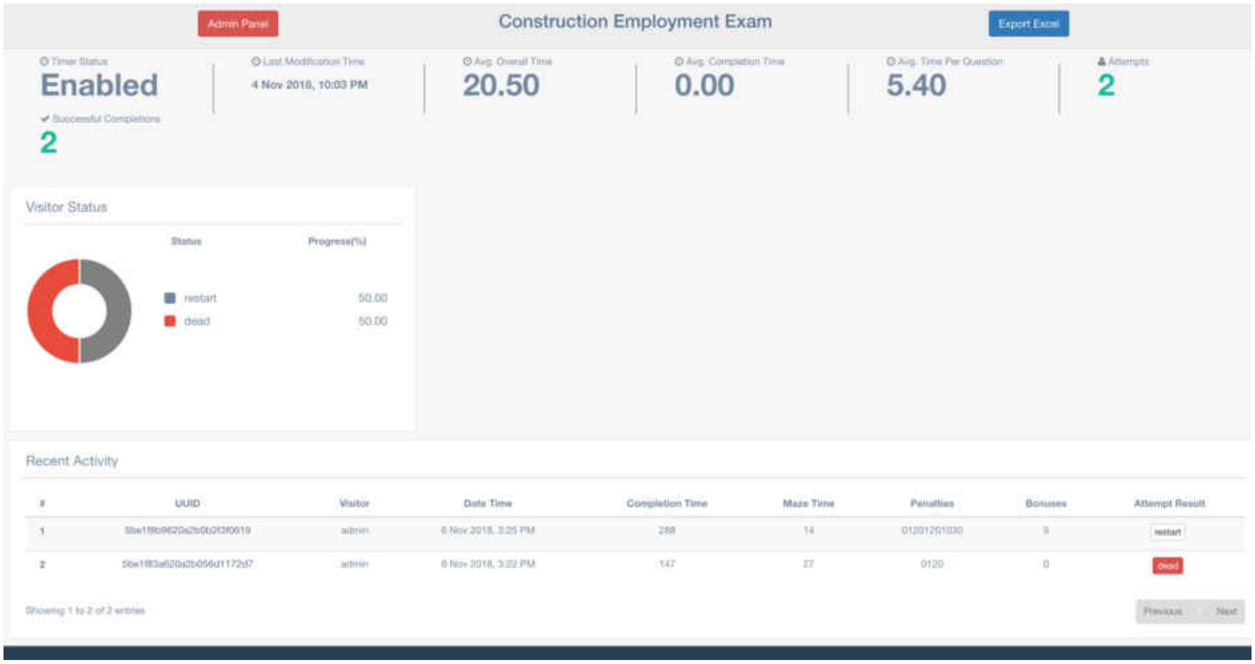

Figure 42. Showing the aggregated maze analytics view.

Figure 43 shows the individual session statistics and the sequence of user choices. Each node in the users can be expanded to show the option the user selected.

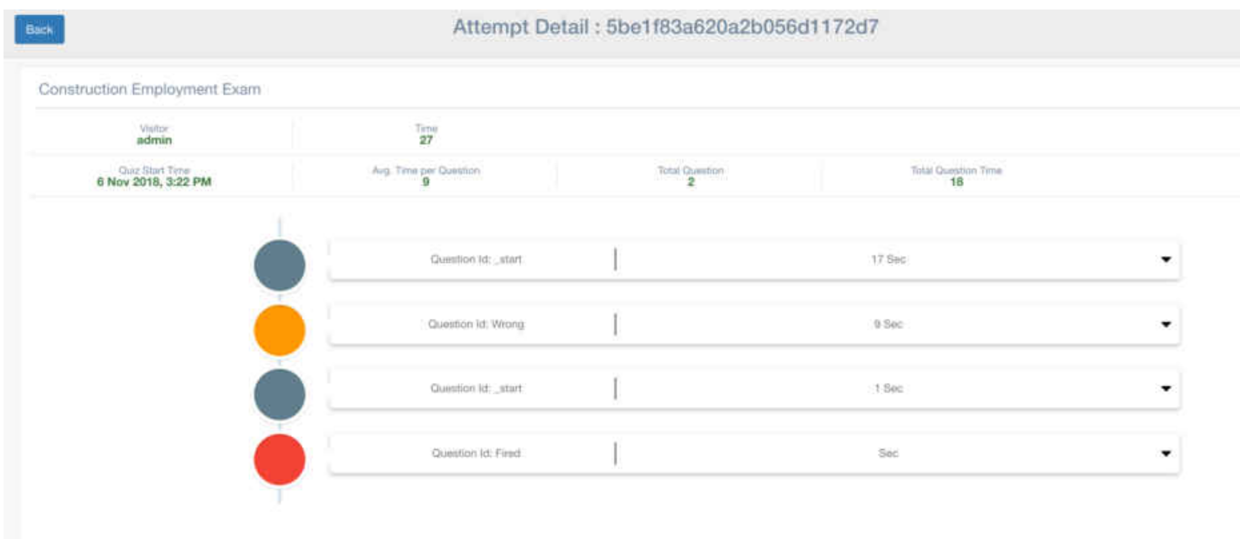

Figure 43. Showing an individual's path taken and their aggregated analytics

Figure 44 shows the nodes in the order the user user's selected them and show which option the user chose.

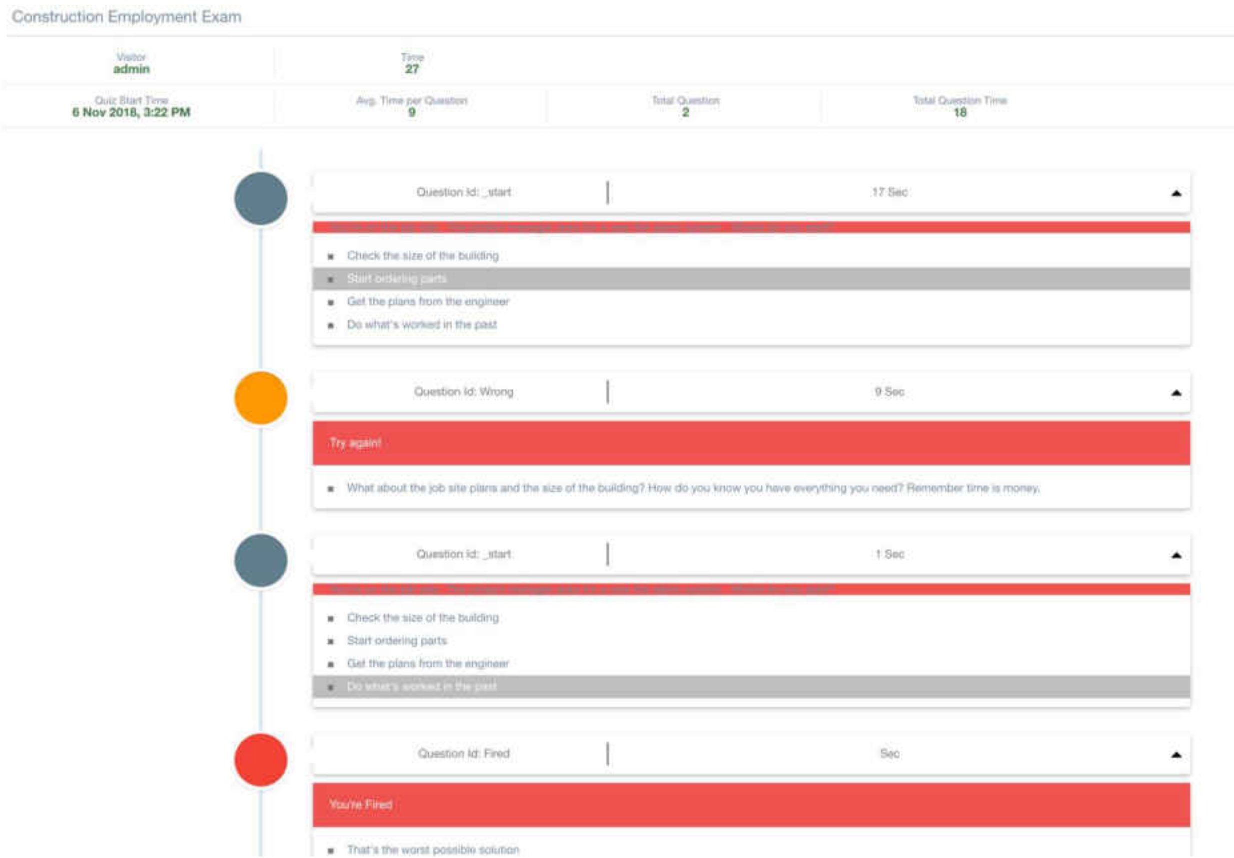

Figure 44. Showing the individual analytics view with the user's selection for each node.

Backend

The backend contains three main components User Identity, Maze, and Analytics components. Each of these components have their own REST API (Application Programming Interface) built in PHP using the Laravel framework. The REST API allows the client-side SPA component to send the user data to the backend component through a dedicated port within the system. This not only improves the security posture of the system its more efficient programming practices and makes it easier to scale the application from layers on a single server to tiers across multiple servers.

User Identity Component

The user identity component authenticates a user's credentials and authorizes the user to assume to the 'Author' role or the learner role depending on the component the user is requesting access to. The user account, registration, access tokens are also handled here. All of the client-side SPA components use the User Identity component validate that the user's token is valid and they are authorized to access the component. The tokens are handled by the JavaScript Web Token (JWT) library. The token is a based64 encoded JSON payload that is encrypted using a hash generated by Laravel and secret key.

Maze Component

The Maze backend component handles all of the maze scenario related activities. During development it was the first component developed. It contains all of the server side functions for the dashboard, maze designer, node editor, node connector, maze configuration, save, update, delete, download, data model construction, and link generation. It also contains all of the node types and behaviors for each node and performs all of the time manipulation processing. It performs all of the maze CRUD (Create, Read, Update and Delete) with the application's database. The maze component constructs the maze JSON data objects are that are stored in the database.

Analytics component

The analytics backend component is used by the Maze player during play to capture the activities performed by the user. As the user completes a maze the analytics component reconstructs the user's path through the maze timestamping every activity and the context with which it was performed.

Databases

The prototype has two databases MongoDB and Redis. Redis is required by the Laravel framework for general application caching. MongoDB is used as the system's primary database and consists of multiple data collections described in Table 13.

Table 13. *Lists application database collection names and description in MongoDB*

Collection Name	Description
Question Set "Maze Graph"	Stores each Maze's configuration and question graph.
Analytics	The Analytics MongoDB collection contains each user's maze session details.
Maze Aggregate	This collection is used to store a Maze play averages and aggregated numbers such as counts.
Token	API generated tokens
Visitor	This collection is used to store visitor's unique ID and session ID generated by Laravel application.
Users	User details

CHAPTER 6 - RESULTS

Overview

This section reviews the hypotheses, describes the research methodology and discusses the results.

Hypotheses

H1: A web-based SaaS application will improve the management and authoring of interactive branching case-based scenarios and enable non-technical subject matter experts to build and manage scenario creation.

H2: Mazetec will reduce the expertise, time and cost barriers imposed by existing scenario-based learning systems while providing greater functionality. It can be seamlessly integrated with an organization's existing LMS.

H3: Unsupervised online distance learners will indicate that they enjoy and benefit from practicing their knowledge in a non-linear case-based scenario when embedded as an activity in an linear course.

H4: Unsupervised online case-based scenarios will improve knowledge and skill retention.

Research Design

The hypotheses were tested using a variety of methodologies including a mix of qualitative survey evidence, a domain-oriented feature analysis of existing scenario-based learning systems, and quantitative analytic data from users completing mazes.

In order to test H1 I first had to build the Mazetec system and evaluate whether non-technical subject matter experts were able to create branching scenarios independently using the software. I find that the system is effective at enabling non-technical SME's to create branching content independently and I present the feedback from beta testers as evidence.

To test H2, which states that Mazetec will reduce the time, expertise, and cost barriers associated with existing scenario-based learning platforms while increasing functionality, I presented a domain-oriented feature analysis of existing scenario-based learning systems. I found that Mazetec compares favorably to existing SBL systems because it requires little to no technical expertise from authors or users, provides useful user analytics to authors, and includes additional functionality such as the timer and a variety of nodes in addition to other advantages.

To test H3, which states that users will enjoy and benefit from practicing their knowledge in a branching scenario, I leveraged the results of surveys that collected open-ended user feedback from May 31, 2012 to October 18, 2018 that had several mazes embedded in linear online QPR suicide mitigation training courses. I found that of the users who cited a specific feature of the course that they enjoyed, the majority cited the maze as the most enjoyable and beneficial feature of the training.

I performed a preliminary evaluation of the Mazetec concept with a manual implementation in which we investigated the users' perception of the usability and usefulness of the method. The goal was to obtain early user feedback to address issues, particularly on usability, before proceeding with more extensive user testing. The first study evaluated Mazetec as a supplemental activity in an unsupervised online course and subsequently on its own.

To test H4 we joined a multi-year study in February 2018 conducted by the Substance Abuse and Mental Health Services Administration (SAMHSA), an operating division of the Department of Health and Human Services (HHS) of the federal government. SAMHSA is currently conducting a study to better understand how the use of active learning strategies (the

Mazetec software[2] and in-person role play interactions) affects trainee knowledge and skills retention over time in regards to the QPR suicide mitigation intervention. Specifically, the SAMHSA study seeks to determine if the Mazetec scenario-based "interventions following training participation increase the effectiveness of gatekeeper [suicidal behavior identification and mitigation] trainings, particularly in terms of promoting identification of [suicidal behaviors among] at-risk youth" (SAMHSA Study Documentation pg 2). While the study has not concluded at this time, we analyzed the data produced by the Mazetec system thus far and present preliminary results and statistical findings and discuss their implications.

Hypothesis 1: Mazetec can be built and Non-Technical SMEs can use it

Previous chapters have discussed in-depth the design and implementation details of the Mazetec software. This indicates that a scenario-based learning platform can indeed be built consistent with Hypothesis 1. However, my first hypothesis also states that a scenario-based learning platform can be created such that non-technical subject matter experts can use it to author, deploy, and analyze mazes independently. As I explained earlier, this was a primary requirement when building the system, and as I will discuss in the next section, it also serves to differentiate Mazetec from existing scenario-based learning platforms.

The subject matter expert (SME) for whom the software was originally targeted to, Dr. Paul Quinnett, holds a Ph.D. in psychology and is 78-years-old. It was crucial that the software be intuitive and user-friendly. The introduction identified Dr. Quinnett's primary challenge with suicide mitigation training online, which is that many of the learners think they already know the content and the method to talk down a suicidal person so they click through the training as

---

[2] The Mazetec software is referred to as "the booster" in the study design documentation because it is utilized as a refresher training in the controlled experimental design.

quickly as possible. Using the Mazetec system, Dr. Quinnett created simple and complex, time-limited suicide mitigation scenarios. The scenarios are divided into multiple branching scenes, each with one or many decisions from which the learner to chooses. Each choice leads the learner on a different path, allowing him to experience the consequences of his decision. The learners goal is simply to navigate to the end of the scenario thereby saving the person in crisis.

As discussed earlier, common technical implementation barriers were identified and subsequent solutions emerged throughout the design and development process such as the authoring tool its subcomponents and the respective data models to support these features.

Hypothesis 2: Mazetec compares favorably Existing SBL Systems

In this section, I discuss the results of a domain-oriented feature analysis of existing scenario-based learning systems. Consistent with H2, I find that the Mazetec platform has more SBL features compared to existing systems by requiring little to no technical expertise from authors or users, which provides useful user analytics to maze authors and includes additional functionality, such as the timer and a variety of nodes.

Table 14 compares domain features, costs, and limitations of four existing text-based branching scenario author platforms to the Mazetec platform. The four existing platforms are Open Labyrinth, Smart Builder, Twine, and Articulate Storyline. All of the platforms discussed here are branching capable, meaning they have the functionality to create maze scenarios. One of the most important elements of any scenario-based learning system is the level of expertise required to use the system. If the system requires authors to possess even rudimentary technical ability, a large portion of subject matter experts will not be able to use the system. I therefore have to rule out Open Labyrinth because, while it is free, the open source software is no longer supported by its developers, requires hosting, patching, operations, and maintenance and thus, has a steep technical learning curve. At face value authors using Smart Builder, Twine, Articulate Storyline, and Mazetec do not need to possess formal technical and systems knowledge in order to use them.

Table 14. Scenario-Based Learning Systems Feature Comparison

	Mazetec	**Open Labyrinth**	**Smart Builder**[3]	**Twine**[4]	**Articulate Storyline**[5]
Cost	TBD	Free	$1,399/annually or $139/month	Free	$1,299 annually per user
Branching capable	Yes ✓	Yes ✓	Yes ✓	Yes ✓	Yes ✓
Level of Expertise	Low	High	Medium	Low	Medium
Code Free Authoring	Yes ✓	No	Yes ✓	Yes ✓	Yes ✓
Scenario Deployment	Link	Link	HTML Export	HTML Export	HTML Export
Scenario Hosting	Cloud	On Premise	Additional fee or self-hosted	No	Additional fee or self-hosted
Authoring Platform	Browser - based	Browser - based	Windows or MacOS application	Windows or MacOS application	Windows or MacOS application
Code Free Authoring	Yes ✓	No	Yes ✓	Yes ✓	Yes ✓ +Optional
Mobile Compatible	Yes ✓	No	No	No	Yes ✓

[3] https://www.smartbuilder.com/elearning-examples/
[4] 1. C. Klimas, Twine, [online] Available: http://twinery.org.
[5] https://community.articulate.com/articles/6-sweet-branching-scenario-examples

Scenario Player					
Mobile Compatible Authoring Environment	Yes ✓	No	No	No	No
Interoperable	Yes ✓	Yes ✓	No	No	Yes*
Learning Analytics	Yes ✓	Yes ✓	Limited with additional fee	No	Yes*
Time Manipulation Capability	Yes ✓	No	Limited	No	Limited
Supported	N/A	Abandoned	Yes ✓	Community	Yes ✓

I also included a comparison based on the level of expertise required for authors. Although four systems (including Mazetec) do not require coding ability to author mazes, the authoring tools vary greatly in the technical expertise required for initial setup, configuration, scenario-authoring, and scenario-deployment. For instance, Smart Builder requires its users to be able to install the software on a computer, its scenario authoring demands above basic technical knowledge including an understanding of HTML DOM events (e.g. on-hover and on-click), and it requires the author to create and place each option or button such as “Begin” and “Next.” Authors must also have a rudimentary grasp of how layering and page design work in addition to basic programming logic including loops, arrays, and if-then statements. Since the authoring tool is desktop-based software, the author is limited to deploy their scenario to their account, or they must host it online themselves or through their organization. In addition, while it claims to be xAPI compatible for an additional monthly fee, its usage statistics are limited to scenario start date, end date, and quiz question responses. It does not include any node-to-node navigation statistics. For these reasons, Smart Builder is described as requiring a *medium* level of expertise. Articulate Storyline is also coded as requiring a medium level of expertise for similar reasons as Smart Builder. However, Articulate Storyline is capable of capturing comprehensive analytics; it requires a high level of technical knowledge and development to fully realize what Mazetec offers out-of-the-box.

Only Mazetec and Twine require a *low* level of expertise to use the software. A low level of expertise indicates an individual with no formal training or grasp of programming logic is able to author mazes. Unfortunately, although Twine does not require a great deal of expertise to author scenarios, the lack of hosting means that a higher level of expertise is required to actually deploy the scenario and have users complete it and even higher to integrate it with another system. Additionally, it does not capture any scenario usage statistics - a significant drawback. In contrast, scenarios or mazes authored using the Mazetec platform do not require technical expertise in hosting since mazes are built and shared online. The authoring tool is entirely web-based meaning an author only needs access to a browser to create, edit, and preview mazes.

Mazetec is integrated with the QPR Institute's existing LMS. When a learner in the LMS reaches a maze in the linear course, the scenario maze player opens in a lightbox overlaying the course. In the background, the maze player calls the Mazetec backend and authenticates, retrieves, and loads the maze into the player. Once the maze is loaded, the player component captures each event and passes the usage data to the analytics component. As each attempt is stored in the Mazetec system the user's results are returned to the LMS. If the learner finishes the Maze, she is able to move forward, otherwise she must try again. The Mazetec analytics groups the individual sessions by maze, and each session is identifiable by its LMS username and system ID. To date over 944 LMS learners have completed the embedded mazes within the QPR Institute's courses.

Of the five total systems compared, only Mazetec combines simple browser-based point-and-click environment, scenario hosting, system interoperability, time manipulation features, and comprehensive analytics designed for time-based branching scenarios.

Hypothesis 3: User Enjoyment and Early Concept Evaluation

To test H3, which states that users will enjoy and benefit from practicing their knowledge in a branching scenario, I leveraged the results of a post-course survey that collected open-ended user feedback from May 2012 to October 18, 2018. During this period, learners were required to complete maze scenarios embedded in linear online QPR suicide mitigation training courses which ranged between two and forty hours.

Mazes were first implemented manually in HTML as a proof of concept and more recently via the Mazetec system implemented through a custom developed API. From a user's perspective, there was no difference between both forms of implementation. For this reason, I pool the results over the time period. I find that of the users who cited a specific feature of the course that they enjoyed, a plurality cited the maze as the most enjoyable and beneficial feature of the training.

Results

Consistent with hypothesis 3, user feedback indicates that the scenario-based training maze was more engaging than information presented in alternative formats such as videos, statistics, lectures, or assessments.

Between May 2012 to October 18, 2018, a total of 5,208 users completed one of QPR's courses that contained an embedded maze scenario. Of these over five-thousand users, 1,923 users completed an optional post-training course survey in which they answered a host of questions regarding their experience and level of enjoyment with the course. This completion rate of approximately 37 percent is not an uncommon response rate for voluntary surveys. Typically, survey researchers must offer a financial incentive or force respondents to complete surveys (e.g. required for course credit, etc) to increase response rate. A voluntary response rate of 37% for a survey offering no incentives is therefore not out of the ordinary.

Of the 1,923 total users who completed the survey, not all of them answered every question (respondents had the option to skip particular questions that they did not wish to answer). Allowing respondents to skip questions is also a common practice in surveys. There were several open-ended questions, and respondents were more likely to skip these questions than the multiple-choice questions. Open-ended questions require more effort than multiple choice questions because of the additional thinking time to formulate one's own answers as well as the added effort of typing those answers out in a comprehensible manner. That being said, open-ended survey questions are very useful because they allow a researcher to determine what respondents are thinking about without forcing them to conform to specific closed answer options. For the purposes of this survey, it was important to ask respondents what they liked most about the training and obtain their honest, unfiltered feedback in a way that was not prejudiced by my own preconceptions. If I had asked a closed-ended question such as, "What feature did you enjoy most in the course?" and gave options such as, "videos," "statistics," "quizzes," or "mazes," I may have influenced respondents to select one of these options instead of reporting a different feature that they preferred but that I simply had not thought of.

To gauge the level of interest and enjoyment with Mazetec relative to other course components, I asked the following open-ended question: "What was most beneficial and unique about this training?” About 64% of the total respondents, or 1,225 people, provided some answer to this question, which I then coded based on references to various course components. The majority (61%) of these answers were not-categorizable as referencing a specific course component. Not-categorizable responses included general comments such as, "good," "informative training," "not sure," "the content," etc. Responses that were categorizable referenced a specific aspect of the training, such as videos, assessments, lectures, tools, statistics, or the maze. Responses that were coded as “maze” used various terms to refer to the maze component as the course did not specifically call the section a maze. The most common references that were coded as “maze” include the "maze," "interactive scenarios," and "real-life trainings." The coding for this question is displayed in Table 15, and the breakdown of the coded answers are displayed as a pie chart in Figure 45.

Table 15. *Coding of Open-Ended Responses to Most Beneficial/ Unique Course Feature*

Terms used	Coding
Interactive Scenarios	Maze
Case Studies	Maze
Role Plays	Maze
Practice Sessions	Maze
Games	Maze
Practice Simulation	Maze
Practice scenarios	Maze

Video	Videos
Videos	Videos
Assessments	Assessments
Statistics	Statistics
Facts and figures	Statistics
Facts	Statistics
Numbers	Statistics
Lectures	Lectures
Tools	Tools

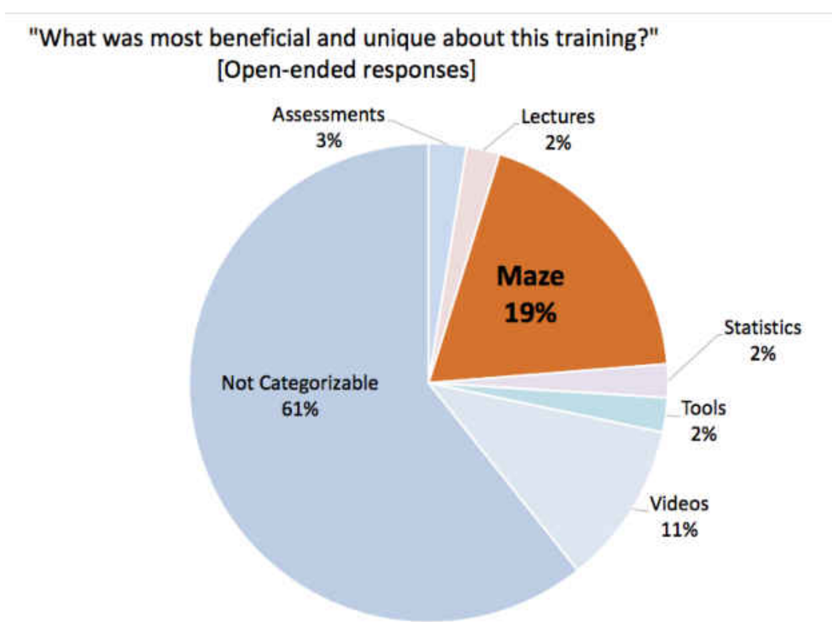

Figure 45. QPR post survey results: Breakdown of user-reported most beneficial/ unique course feature.

As I already mentioned, most responses were too general to categorize as referencing a specific feature. However, 39% of respondents did reference a specific aspect of the course that they found most beneficial and unique. Figure 46 displays the breakdown of categorizable responses by the specific feature or content they referenced.

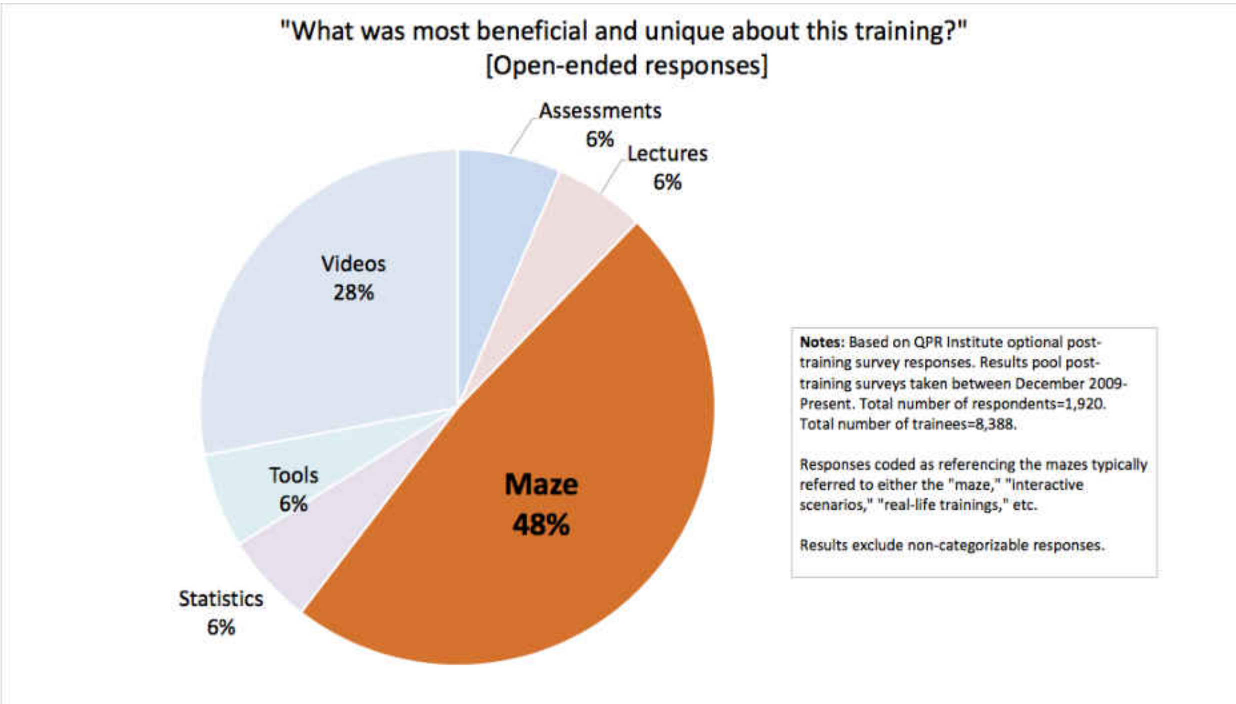

Figure 46. Breakdown of user-reported most beneficial/unique course feature.

Among those people listing a specific aspect they found most beneficial and unique about the training, nearly half of respondents referenced the maze. The next most common category referenced was the videos (28%), followed by an even proportion of individuals who referenced either the assessments (6%), lectures (6%), tools (6%), or statistics (6%). The number of respondents who stated that they found the maze to be the most beneficial and unique aspect of the training was nearly double those who said they found the videos to be the most beneficial and unique. This large proportion of respondents who enjoyed the maze component of the course enough to take the time and effort to fill out the survey and reference it in an open-ended question indicates that respondents enjoy and find the maze beneficial, consistent with my prediction in Hypothesis 3.

Before moving on to the discussion of the final hypothesis. It is important to underscore these open-ended responses given the pool of potential and actual respondents. Recall that 245 people who completed the survey specifically took the time to mention the maze as being the most unique and beneficial aspect of the training. Yet an analysis of the word count of the various types of content within each of the courses indicates that the maze comprises a very small percentage of the overall course content, only between 1.6% and 3.3%, depending on the specific course, see Table 16 and Table 17. Considering the fact that each individual user spends only a small fraction of their time in the course on the maze, the fact that 20% of survey respondents took the time and energy to specifically mention it as the most unique and beneficial feature of the course.

Table 15. *Comparing Time Spent in the Linear Course vs the Mazetec Branching Scenario Embedding in the Course*

Activity	**Word Count**	**Average Completion Time (Minutes)**
Mazes		
NSSI Warning Sign Learning Maze	497	3.22
Non Suicidal Self-Injury Tutorial	753	8.30
Total	1,250	11.52
Courses with embedded Mazes		
QPRT Management Course	77,412	600

QPR for Nurses	64,114	420
QPR Triage Level II	54,433	600
QPR for School Health Professionals	38,496	210

Table 16. *Percentage of Maze content compared to the relative course content*

Course Name	**% of course content that is the Maze**	**% of course time users spend in the maze**	**Number of Post Survey Respondents**
QPRT	1.61%	1.92%	84
Nursing	1.95%	2.74%	48
QPR Triage Level II	2.30%	1.92%	46
QPR for School Health Professionals	3.25%	5.48%	16
*Remaining courses less than 4%			35

Hypothesis 4: Mazes will improve knowledge and skill retention (SAMHSA Research Study)

The final hypothesis (*H4*) is that the use of scenario-based mazes created with Mazetec improves users' knowledge and skill retention. The complete results of the formal study for H4 are not yet available; however, the preliminary data for the SAMHSA research study provides useful data on the system and the capabilities of the analytics and identifies an intriguing phenomenon that I had not predicted. Specifically, I find evidence of voluntary additional maze-play by some users (completed maze attempts beyond what is required for the training).

Overview

The SAMHSA study is designed to evaluate the efficacy of various components of suicide mitigation training, including the traditional QPR online gatekeeper course, in-person training, and the online maze created with the Mazetec platform. The full study design and details are in [8], but for the purposes of this overview, I will focus only on the components that pertain to testing the efficacy of the maze training created with the Mazetec platform. The study required a randomly selected subset of participants to complete the maze component of the QPR training twice with a 3-month lag period in between. A different group was randomly assigned to complete only the traditional online QPR gatekeeper course (without the maze). At the conclusion of the study, these groups will complete an assessment of their knowledge, and the groups will be compared. If the Mazetec training is successful at increasing skill retention and knowledge, participants in the group randomly assigned to take the maze should demonstrate higher scores on the post-assessment evaluation than the groups that completed only the basic QPR training.

Unfortunately, due to delays outside this researcher's control, the full results of the SAMHSA research study are not yet available, and therefore, the veracity of H4 cannot yet be determined. In lieu of evaluating the complete study results, I examine the preliminary results of the study in this section.

Results

One of the most interesting aspects of the preliminary SAMHSA data is the frequency that some participants complete the maze. In the study, participants are required to complete the maze twice with a three-month lag period in between. However, not only do many participants in the study complete the maze more often than is required (e.g. more than twice), they do so in a very short period of time. For instance, they take the training twice in the same day, twice over a span of two weeks, or twice over a span of two months (in addition to taking the training after the requisite three-month lag).

While it is unclear what is driving this behavior, several factors lead me to believe that this is indicative of greater engagement and voluntary behavior on the part of participants. The chief alternative explanation – that the participants have forgotten that they have already taken the training – is implausible for those participants who take the training multiple times in the same day or the same month and then go on to take it a third or a fourth time when asked. In addition, I see some evidence that participants are trying to cover up the fact that they are taking the maze more often than required. Specifically, participants exhibiting this behavior spell their names differently each time (e.g. failing to list a middle name in one entry, listing a middle initial name in another entry, then listing their full middle name in another entry), and using a mix of personal and work email accounts. Taken as a whole, this indicates preliminary evidence of engagement in the form of voluntary maze completion among a subset of 22 identifiable respondents. To be clear, this means that 22 people completed the maze more often and/or more frequently than they were required to. There is strong face validity that the maze prompts greater learning engagement in the form of voluntary maze-play among a subset of participants. I will present more details on this phenomenon later in this section, but first, I will provide a description of the data, which will illuminate the utility of the various analytics that the Mazetec system generates.

General description of the Preliminary SAMHSA data.

The preliminary SAMHSA data contains information about 155 total attempts at completing the maze for 92 unique individuals. An individual is considered unique if either his self-provided name matches perfectly, if this self-provided email addresses matches perfectly, or if the differences between the self-provided name is minor and he could not possibly match to any other individual (e.g. Sam Smith and Sam S. Smith with samssmith@email.com provided for both name entries would be considered a match).

For each attempt, Mazetec generates the data fields shown in Table 18.

Table 17. *Learner's Maze Analytic Variables Produced by Mazetec*

Variable Title	**Variable Name**	**Description**
_id	ID	A unique identifier for each attempt.
avgQuestionTime	Average Question Time	Total time spent on the maze divided by the total questions
created_at	Created At	Logs the start date and time of the attempt
endType	End Type	Categorical variable describing the outcome of the attempt. Options include "finish", "restart", "timeout"
metauser_email	User Email	Self-reported email address of user
metauser_name	User Name	Self-reported name of user
overallTime	Overall Time	Start timestamp - End Time Stamp

totalQuestion	Total Questions	Number of scene nodes a user experiences
quiz_id	Quiz ID	Uniquely Identifies Maze

Using these fields, I generated additional variables for analysis as show in Table 18.

Table 18. Study Variables

Variable Title	**Variable Name**	**Description**
uniqueIndividual	Unique Individual	A unique numeric indicator that groups the attempts by individual. For instance if John Doe is the third unique individual in the dataset and he takes the maze four times, he will have four entries in the dataset and will have a *uniqueIndividual* value=3 for all four rows.
Engagement	Engagement	Coded as 1 if the the unique Individual has more than two attempts at the maze with End Type= "finish" OR the created at date and times indicate attempts that are earlier than what is required (e.g. completing the maze twice in the span of a week or a month).
nodes	Nodes	Represent the contextual information, decisions, and feedback. Each node is a

		page of contained information. For example a Question node presents multiple options to the user to choose from
errorrate	Error Rate	Calculated as: *nodes-totalQuestion* An error rate of 3 indicates the user made 3 mistakes in the maze. Mistakes include selecting the wrong option or requesting a hint.
improved	Improved Error Rate	Coded as 1 if the error rate is lower in a later (e.g. second, third, fourth, fifth) attempt compared to the first attempt. Only compares attempts with End Type= "finish".
improvedOT	Improved Overall Time	Coded as 1 if the Overall Time is lower in a later (e.g. second, third, fourth, fifth) attempt compared to the first attempt. Only compares attempts with End Type= "finish".

Summary Statistics

Of the 155 total attempts, the vast majority (90.32%) were successful attempts (End Type= "finish"). 7.1% of attempts were restarts, and 2.58% were timeouts. Eleven unique individuals are responsible for the 15 total unsuccessful attempts (either restart or timeout). The overwhelming proportion of successful completions and the small number of individuals responsible for the few unsuccessful attempts that did occur indicate that, in general, participants were not confused by the maze task, nor did many feel the need to restart the maze.

Table 19. Summary Statistics for Preliminary SAMHSA Data

	N	Mean	St. Dev.	Min	Max
Average Question Time (seconds)	155	16.03	7.66	0	42.42
Overall Time (seconds)	155	239.25	129.77	3	600
Total Questions	155	13.25	4.01	0	23
Total Nodes	155	17.28	6.13	1	36
Error Rate	155	4.04	2.47	1	13

Table 2: Summary Statistics for Preliminary SAHMSA Study Data by End Type

	N	Mean	St. Dev.	Min	Max
Finish					
Overall Time (seconds)	140	242.26	109.65	58	585
Total Nodes	140	18.34	4.81	12	36
Error Rate	140	4.17	2.40	1	13
Restart					
Overall Time (seconds)	11	69.73	91.82	3	330
Total Nodes	11	4	3.29	1	12
Error Rate	11	1.64	1.03	1	4
Timeout					
Overall Time (seconds)	4	600	0	600	600
Total Nodes	4	16.75	11.09	7	31
Error Rate	4	6	3.83	3	11

The first part of Table 19 presents summary statistics for the Average Question Time, Overall Time, Total Questions, Total Nodes, and Error Rate variables. The second table in Table 19 displays the Overall Time, Total Nodes, and Error Rate by the attempt End Type (Finish, Restart, and Timeout). The mean Average Question Time in seconds for all respondents was 16 seconds. The distribution of Average Question Time has a mild positive skew, meaning that the mean Average Question Time is slightly inflated due to outliers with high Average Question Times. The median value is 15 seconds. Three attempts had a minimum value of 0 seconds (all of these attempts were immediate restarts), and the maximum value indicates some individuals spent close to seven-tenths of a minute on each of the maze questions. Examining the data more closely, nine out of the ten attempts with the longest Average Question Time were successful *finish* attempts. This indicates that the large range of Average Question Time Values is not driven by unsuccessful attempts (such as timeout-s) inflating the results, but rather some individuals appeared to spend much longer on the maze than others did.

The Overall Time variable only partially conforms to the information gleaned from the distribution of the Average Question Time. Overall Time indicates the total number of seconds spent on the maze. The mean Overall Time of about 242 seconds corresponds to a maze completion time of around 4 minutes. Like the distribution of Average Question Time, the distribution of Overall Time is skewed to the right, indicating a mean that is inflated by a few large outliers. The standard deviation of about 130 seconds indicates that about 68% of the total attempts have an overall completion time of between 2 minutes and 6 minutes. Moreover, it indicates that nearly a third of attempts took less than 2 minutes or more than 6 minutes. Examining the distribution of Overall Completion Time more closely indicates that only one of the attempts took less than 1 minute to complete (all of the other extremely short attempts were restarts). Indeed, ten out the eleven total restart attempts took 99 seconds (a little over 1.5 minutes) or less in Overall Time. By definition, the four timeout attempts are the maximum values of Overall Time with values of 600 seconds, or 10 minutes. As indicated in Table 2, the range of successful maze completions (finishes) is between about 1 minute and 9 minutes 45 seconds. The range of Overall Time for Restarts indicates that, while most restarts occur very early in the maze play, one respondent restarted his maze after 5 and a half minutes of game-play. Examining this attempt more closely reveals that this attempt involved only a single error, 6 nodes, and an Average Question Time of 21 seconds. This means that the user likely got distracted during play and left her screen on a non-time-accruing page such as a hint, prior to coming back to the maze and restarting it.

The minimum possible nodes to complete the maze was 11. As displayed in Table 2 under minimum Total Nodes and minimum Error Rate, no one completed the maze successfully without making at least one error. The single finish attempt with only a single error was completed in 1 minute and 14 seconds. Examining this attempt in more detail shows that the same individual had taken the maze once before this attempt, approximately 7 minutes prior to be exact. In this individual's first attempt, it took him a little over 5 minutes to complete the maze with an error rate of 4. This individual was one of the cases coded as "Engaged": the individual took the maze more frequently than required – about 2 minutes after completing the maze the first time. The individual used a different configuration of his name on the second attempt (listing a middle initial) and used a personal email account in the second attempt (the first attempt was a professional email address). I will return to the discussion of engagement later in this section.

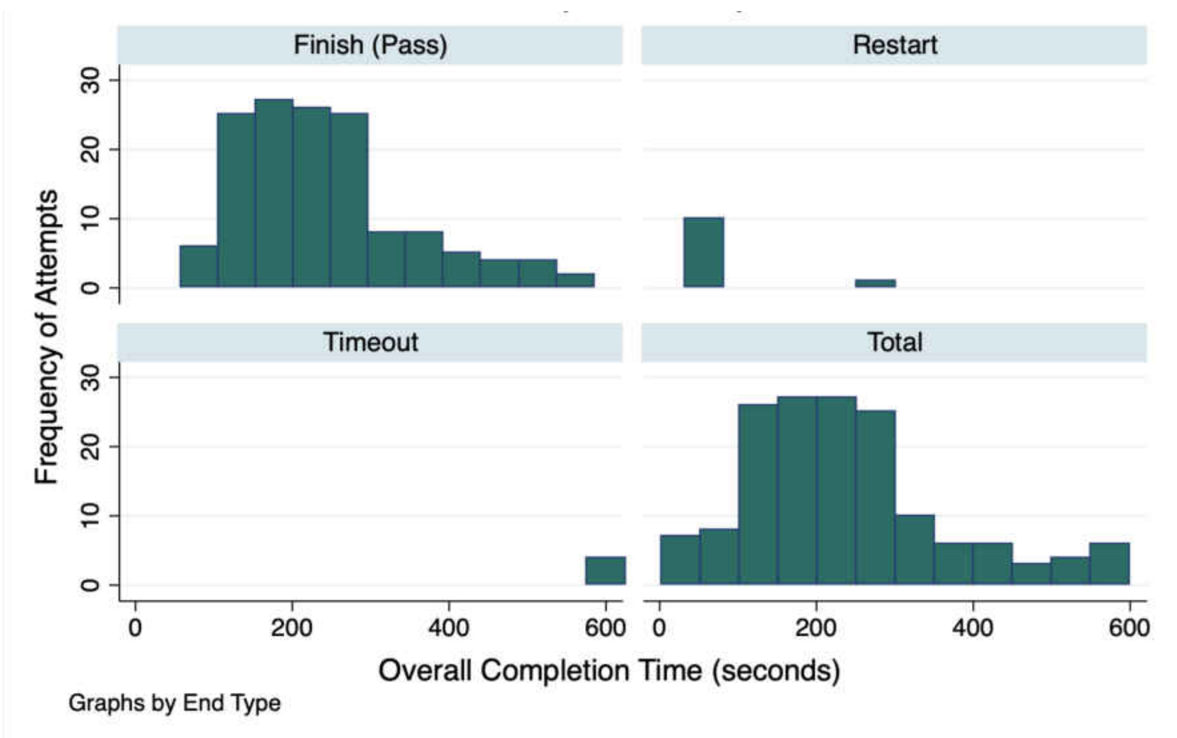

Figure 47. Distribution of overall participant scenario time by end type.

Figure 47 is a set of histograms that display the frequency or number of attempts at completing the maze for a particular completion time and end type. The x-axis is the overall completion time in seconds. For instance, a completion time of 180 seconds corresponds to 3 minutes of game play. The y-axis indicates the frequency or number of attempts that fall within a particular completion time range. Higher bars indicate a greater number of attempts at that completion time. Each of the panels represent the results broken down by "end type" and total attempts, which pools all end types. Recall that the End Types represent one of the current analytics gathered by the system. Participants can either: 1) finish the maze, which indicates a passing score); 2) select to restart the maze (forcing them to go back to the beginning and discard their current attempt); or 3) they may timeout of the maze (meaning that at 10 minutes of game play they had not yet completed the maze).

The top left panel shows the distribution of overall completion time in seconds for those who finished the maze (received a score of pass). You can see from this figure that the distribution of completion times for those who passed the maze is right skewed. Most respondents finished the maze between 3 minutes and 5 minutes, but there are smaller numbers of attempts to complete the maze that took a much longer time (between 6 and 9 minutes) to complete the maze while still finishing. The large range in finish completion times may indicate a large variety in skill, knowledge retention, or play-style among respondents. For instance, someone who spent 300 seconds (5 minutes) completing the maze may have been less secure in her knowledge of the content compared to someone who took only 3 minutes to complete the maze. However, it may also be the case that the person who took 5 minutes to complete the maze actually made fewer mistakes than the person who finished in 3 minutes and made many errors. In the next section, I discuss how overall completion time should be examined with the total number of errors or error rate to classify respondents. However, without the results from the SAMHSA post-test on the knowledge retention, any classification scheme of respondents' expertise is uncertain.

The Relationship between Error Rate and Overall Completion Time

The previous section introduced the question of whether there is a relationship between the error rate and the overall completion time: are these two metrics positively related to one another such that more errors are related to a higher overall completion time? This is an important question to examine because it may be indicative of the best metrics (if any) to use when classifying maze players by expertise. Are those with low error rates and low completion times experts? Without the results of the SAMHSA post-test knowledge assessment, it is not possible to fully answer this question. However, given the data that is available, I can assess a) whether there is a relationship between error rate and overall completion time, b) the strength of that relationship, and c) the direction of that relationship. To evaluate these three dimensions, I first calculated the correlation coefficient or Pearson's *r* between for Overall Completion Time and Error Rate using the following formula:

$$r = \frac{1}{n-1}\sum\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

Where x_i is the observed Overall Completion Time in seconds; $\bar{x}$ is the mean Overall Completion Time in seconds;s_x is the standard deviation of Overall Completion Time; y_i is the observed Error Rate; $\bar{y}$ is the mean Error Rate; and s_y is the standard deviation of the Error Rate.

Performing this calculation yields an *r* of about 0.27. This indicates a weak positive correlation between Overall Completion Time and Error rate. Because the variable Overall Completion Time has a number of extreme outliers (both high and low), I also performed a log transformation of Overall Completion Time and calculated the correlation coefficient using the log transformed variable. This yields a slightly, although not overwhelming, stronger relationship with r=0.30.

Table 20. Overall Completion Time and Error Rate

	Dependent variable:	
	Error Rate	
	(1)	(2)
Overall Time (seconds)	0.005***	
	(0.001)	
Log Overall Time (seconds)		0.999***
		(0.260)
Constant	2.804***	−1.249
	(0.403)	(1.390)
Observations	155	155
R^2	0.073	0.088
Adjusted R^2	0.067	0.082
Residual Std. Error (df = 153)	2.386	2.367
F Statistic (df = 1; 153)	12.135***	14.742***
Note:	*p<0.1; **p<0.05; ***p<0.01	

To confirm that there is a weak positive relationship between these two variables, I performed an Ordinary Least Squares regression in which overall completion time predicts the error rate. I also regressed the log of overall completion time on error rate. The results of these regressions are presented in Table 20. Calculations were performed using Stata 13. The column labeled (1) indicates the results for the following equation:

$$y = b_0 + b_1 * OverallTime + \epsilon$$

The column labeled (2) indicates the results for the following equation:

$$y = b_0 + b_1 * log(OverallTime) + \epsilon$$

The relationship between Overall Time and the Error Rate is positive and statistically significant at $p<.01$. This indicates that a greater Overall Time is associated with a greater number of errors and the relationship observed in these data is highly unlikely to have occurred by chance if a true relationship exists in the population. Substantively, an effect size of 0.005 means that an additional 200 seconds leads to an additional one error, on average. The results for model 2 that uses the log of the Overall Time essentially confirm these findings, although as the comparison between the R-squared values indicates, the log transformation of Overall Time slightly improves model fit (by reducing the impact of outliers on the regression calculation).

Improvement in Error Rate Across Attempts

The next feature of the data I examine is whether there is an improvement in the error rate after the first maze attempt. I classify an improvement in the error rate as any time a subsequent finish attempt has a lower error rate than the first finish attempt. I do not include restart or timeout attempts as the preliminary attempt (when applicable) because many of these attempts involved few questions viewed and likely a great deal of inactive time which makes these attempts non-comparable to the finish attempts.

There are 35 unique individuals who finished a second attempt. Of these 35 potential cases, 10 individuals successfully finished a third attempt. Two individuals completed a fourth attempt, one person successfully finished a fifth attempt, and one person finished a sixth attempt at the maze. Out of the 35 individuals who completed at least a second attempt, 22 of them improved their error rate in their second, third, fourth, fifth, or sixth attempt relative to their first attempt. This represents approximately 63% of individuals who improved their error rate performance across attempts.

Examining the distribution of those who improved their error rate on a subsequent attempt, 15 were those classified as engaged (they either finished the maze more than twice or they finished the maze in a shorter time frame than what was required by the researchers). Of those who followed the requirements of the study – they completed the maze twice with a roughly three-month period in between completions – 7 improved their error rate.

This evidence is not surprising. Many of those classified as “engaged” completed the maze more often than required or in a shorter time frame than what was required, therefore making them more likely to improve their error rate. However, the fact that the error rate improvement is not surprising should not obscure the fact that this improvement in error rate among the “engaged” resulted from voluntary behavior on the part of participants. Participants were not asked or even encouraged to finish the maze more often than twice, or at shorter time intervals. Training that is capable of encouraging voluntary participation such as this, in addition to diminishing the error rate between trials, is promising for the future of the Mazetec software. While I do not currently have the data to prove so quantitatively, the Mazetec software’s ability to engage participants and encourage voluntary participation represents a clear departure from ordinary mandatory training programs.

Overall Completion Time by Engaged and Disengaged

Next, the Overall Completion Time of the engaged compared to those who met the requirements is examined. Please note that for lack of a better term, these participants are referred to as “disengaged,” but in reality, these individuals merely met, rather than exceeded the requirements. As in prior analyses, I focus on comparing similar End Types. Table 21 presents the summary statistics for Overall Completion Time by Engagement among finished attempts.

Table 21. Overall Time by Engaged and Disengaged (Finish Attempts Only)

	N	Mean	St. Dev.	Min	Max
Disengaged					
Overall Time	82	252.70	96.77	106	579
Engaged					
Overall Time	58	227.50	125.02	58	585

As shown in Table 21, the mean overall completion time in seconds is lower among those classified as engaged versus those classified as disengaged. The standard deviation is substantially wider for the engaged, but as a comparison of the minimum and maximum values and the overall distributions displayed in Figure 48 shows, most of this larger standard deviation among the engaged is driven by a greater proportion of respondents with low completion times.

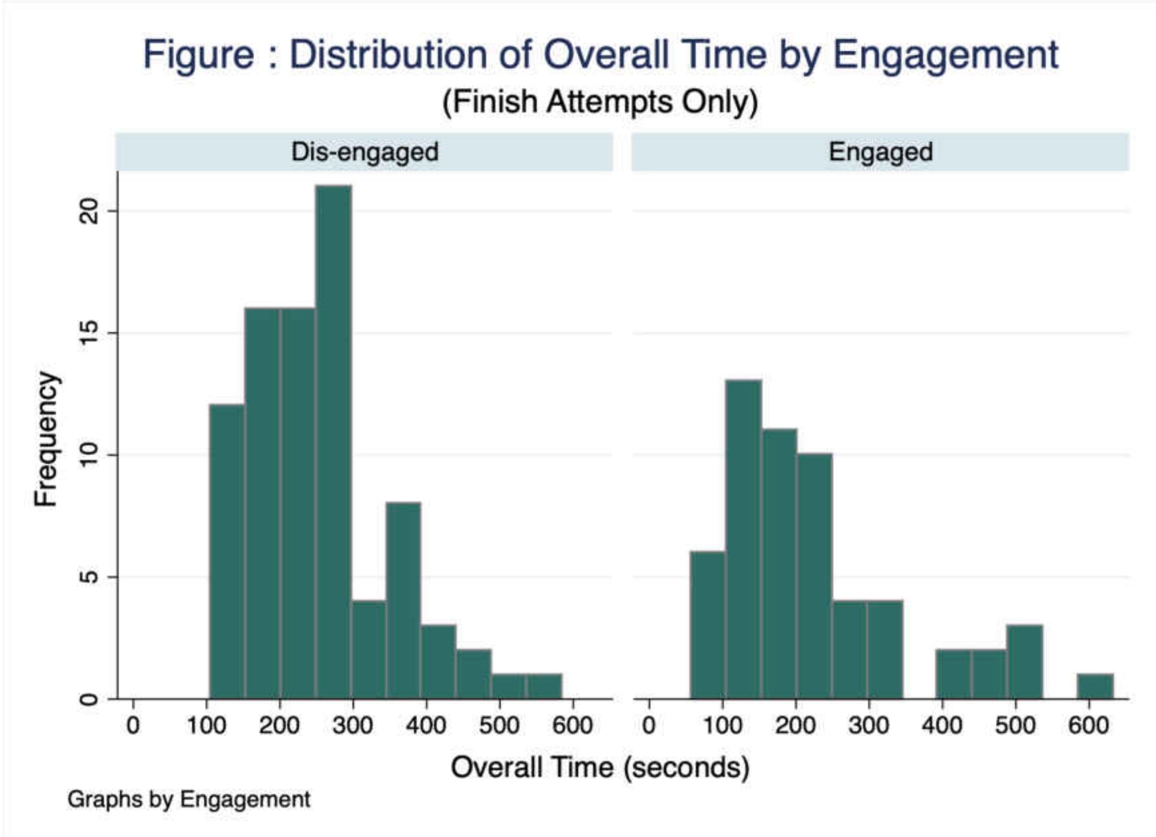

Figure 48. Distribution of overall time by engagement.

Improvement in Overall Time Across Attempts

Earlier, I reported the breakdown of those who improved their error rate in subsequent finished maze attempts by those coded as engaged and those coded as disengaged. I found that more engaged individuals improved their error rate than did disengaged individuals, despite engaged individuals making up a smaller percentage of the overall sample. In this section, I conduct the same analysis for improvement in Overall Completion Time. I coded an individual as improving her completion time if the lowest (best) overall completion time is not her first attempt. Based on this criterion, 24 individuals improved their overall completion time. This means that about 69% of those who completed at least a second attempt improved their completion time relative to their first attempt.[6]

Next, I examined whether engaged individuals were more likely to improve their completion time than disengaged individuals (those who only completed the maze as they were required). Of the 24 individuals who improved their completion time, 17 of them were coded as engaged, meaning they took the maze more often than was required or in a shorter time span that was required. Seven were coded as disengaged, meaning they completed the maze as required by the researchers. This corresponds to the engaged individuals comprising about 71% of the overall completion time improvement cases, compared with just 29% of the improved time cases belonging to the disengaged category.[7]

Table 22. Frequency of Engaged and Disengaged Participants in SAMHSA Study Preliminary Data

	Frequency	Percentage
Dis-engaged	70	76.09%
Engaged	22	23.91%
Total	92	100%

Examining the likelihood that engaged individuals would improve their overall time reveals that 77% of engaged individuals improved their overall completion time on a subsequent

[6] 24/35=68.57% (improved overall time/ completed at least a 2nd attempt).

[7] 17/24= 70.83%; 7/24=29.17%.

attempt presented in Table 23.[8] In contrast, only about 10% of the disengaged improved their overall completion time.[9] As with the error rate improvement, much of what is driving this effect is that the engaged individuals either took the maze more than twice, thereby giving them more opportunities to improve their overall completion time in an attempt after their first- and they took the maze in a shorter time frame than was required- thereby making them more likely to remember the content and improve their completion time as a result. Table 23 presents the breakdown of attempts by Engaged versus Disengaged. Each cell indicates the raw frequency or count of individuals falling into that category. For instance, the top left cell in the table that says 70 indicates that 70 people were both classified as disengaged and finished their first attempt.

Table 23. Frequency of Finished Attempts by Engaged and Disengaged Participants in SAMHSA Study Preliminary Data

	First Attempt	Second Attempt	Third Attempt	Fourth Attempt	Fifth Attempt	Sixth Attempt
Dis-engaged	70	13	0	0	0	0
Engaged	22	22	10	2	1	1
Total	92	35	10	2	1	1

[8] 17/22=77.3% (Total Engaged Improvers/ Total Engaged= Probability that engaged improves their overall completion time).
[9] 7/70=10% (Total Dis-Engaged Improvers/ Total Dis-Engaged= Probability that dis-engaged improve their overall completion time).

As you can see, all those coded as engaged finished a second attempt, but only 13 people, or about 19% of those coded as disengaged, successfully completely a second attempt. Anyone who completed the maze more than twice was coded as engaged, so all the cells for the third to sixth attempts for the disengaged are 0. A total of ten people, or about 11% of the total individuals, and 45% of those coded as engaged, completed a third attempt.[10] Only two people, or about 2% of the total individuals, and about 9% of those coded as engaged, completed a fourth attempt.[11] Only one person, or 1% of the total individuals, and 5% of the engaged, completed a fifth and a sixth attempt.[12]

[10] 10/92=10.87% (total third attempts/ total individuals); 10/22=45.45% (total third attempts/ total engaged).
[11] 2/92=2.17% (total fourth attempts/total individuals); 2/22=9.09% (total fourth attempts/ total engaged).
[12] 1/92=1.09%; 1/22= 4.55%.

Table 24. Frequency of Improved Overall Time by Attempt Number and Engaged vs Disengaged Participants

	First Attempt	Second Attempt	Third Attempt	Fourth Attempt	Fifth Attempt	Sixth Attempt
Dis-engaged						
Total Attempts	70	13	0	0	0	0
Total Improved Overall Time	N/A	**7**	N/A	N/A	N/A	N/A
Engaged						
Total Improved Overall Time	N/A	**9**	**7**	**1**	0	0
Total Attempts	22	22	10	2	1	1
Total Attempts	92	35	10	2	1	1
Total Improved Overall Time	N/A	**16**	**7**	**1**	0	0

As displayed in Table 24, all of those coded as disengaged, who improved their overall completion time, did so on their second attempt. Of the individuals coded as engaged, who improved their overall completion time, 9 did so on their second attempt, 7 did so on their third attempt, and 1 improved their time on their fourth attempt.

Summary Statistics by Attempt Number and Engagement

Finally, whether the engaged and disengaged participants had significantly different error rates or overall completion times across attempts is examined. Table 25 displays a variety of summary statistics for disengaged and engaged participants by attempt number. I wanted to see whether the disengaged participants did noticeably worse on their first attempt (e.g. higher error rates and/or higher overall completion times) compared to the engaged participants.

*Table 25.*Comparing the Mean Error Rate on the First Attempt

	First Attempt	Second Attempt	Third Attempt	Fourth Attempt	Fifth Attempt	Sixth Attempt
Dis-engaged						
Mean Error Rate	4.37	4.39	N/A	N/A	N/A	N/A
Modal Error Rate	5	2	N/A	N/A	N/A	N/A
Min, Max Error Rate	1, 13	2, 11	N/A	N/A	N/A	N/A
Mean Overall Time	259.51	242.69	N/A	N/A	N/A	N/A
Median Overall Time	253.5	174	N/A	N/A	N/A	N/A
Min, Max Overall Time	106, 600	135,579	N/A	N/A	N/A	N/A
Engaged						
Mean Error Rate	4.27	4.04	3.6	2	2	3
Modal Error Rate	4	1 & 2	4	2	2	3
Min, Max Error Rate	1, 9	1,9	2, 5	2,2	2,2	3,3
Mean Overall Time	291.77	217.41	146.4	107.5	145	169
Median Overall Time	269.5	190	141.5	107.5	145	169
Min, Max Overall Time	118, 524	58, 585	67, 228	90, 125	145, 145	169, 169

Comparing the mean error rate on the first attempt for both types of participants reveals that the engaged participants had a slightly lower average error rate than did the disengaged participants. Examining the modal or most common error rates reveals that the modal error rate is 1 point lower among the engaged compared to the disengaged. Examining the frequencies granularly shows that among the engaged, about 64% had an error rate of 4 or lower on their first attempt. In contrast, among the disengaged, only 53% had an error rate of 4 or lower.[13] Higher initial error rates may provide a partial explanation regarding what prompts engaged versus disengaged behavior on the part of participants.

Examining the summary statistics for overall completion time further illuminates what might differentiate engaged versus disengaged participants. The average and median completion times for the engaged participants are higher than for the mean and median completion times for the disengaged participants. This suggests that, on average, participants who go on to display greater interest in the maze displayed signs of greater engagement on their first attempt – namely, they took longer to complete the maze than did disengaged participants, which may indicate greater thoughtfulness or deeper processing of the information.

Evidence of Deception by Engagement: Variation in Email Addresses and Names

It was argued that those individuals who either completed the maze more often than required or more frequently than required did not do so because they forgot they had met the requirements for the study or were confused about them. Instead, these people are demonstrating engagement: they complete the maze more often than is required or in a shorter time frame than required because they find the maze enjoyable or useful, and this leads them to do more work than is necessary.

The chief evidence that it is engagement rather than forgetting stems from indications that users who take the maze more often or in a shorter time frame than required attempt to cover up these attempts by engaging in a pattern of what can be called "deceptive behavior." I define deceptive behavior as listing different email addresses or different configurations of one's name to (presumably) mask multiple attempts from either the researchers or the Mazetec system. Of

[13] 14/22= 64% (error rate less than or equal to 4/ total engaged); 37/70= 53% (error rate less than or equal to 4/ total dis-engaged).

course, to some extent, listing different configurations of one's name or different email addresses is normal behavior. People do not always recall whether they spelled out their full name, only listed a middle initial, or only listed their first and last name. Similarly, people cannot always recall which email address they provided on a webpage. Everyone has had the experience of forgetting your password, your username, and the email used to sign up for a particular website. Luckily, the last option is typically easy to resolve as on average, people have fewer than two email addresses each,[14] so trying both addresses is not as lengthy a process as trying a (potentially) infinite number of passwords and usernames. While some of this behavior may reasonably be expected, if engaged participants are truly trying to mask multiple attempts, then there should be more unique email addresses or different configurations of one's name among engaged participants than among disengaged participants. Indeed, these data show that people listed as engaged are more likely to provide multiple email addresses than are those who are disengaged.

Table 26. Frequencies and Percentages of Individuals Providing Matching Versus Non-matching Email Addresses by Engagement

[14] The estimate for the average number of email addresses per person is based on survey evidence from Zettasphere, a British survey research and marketing firm. Zettasphere. 2018. "The number of email addresses people use [survey data]." Date Accessed: October 23, 2018. Available online at: https://www.zettasphere.com/how-many-email-addresses-people-typically-use/

		Engagement	
		Dis-engaged	*Engaged*
Matching Emails	Frequency Percent	60 85.71%	15 68.18%
Non-Matching Emails	Frequency Percent	10 14.29%	7 31.82%
Total		70 100%	22 100%

Note: Pearson's $\chi^2 = 3.42$. $p = 0.065$

Table 26 displays the raw frequencies and percentages of individuals providing matching versus non-matching email addresses by engagement. Nearly 15% of those classified as dis-engaged provided non-matching email addresses compared to over 31% of engaged who gave different email addresses across attempts. More than double the proportion of engaged participants gave non-matching emails than did disengaged respondents. Typically, statistical tests of significance would be extremely difficult given the very small sample size. Yet, the relationship between providing non-matching email addresses and Engagement is so strong that a Pearson's Chi-Squared test of independence yields a test statistic with $p<0.1$, meaning that the null hypothesis that providing non-matching email addresses is independent of Engagement and can be rejected at the 90% confidence level.[15]

In addition, people listed as engaged are more likely to provide multiple non-matching configurations of their names than are those who are dis-engaged.

Table 27. Frequencies and Percentages of Individuals Providing Matching Versus Non-matching Names by Engagement

[15] Pearson's chi-squared calculated the probability of observing the distribution of frequencies in the data if there is no relationship in the population. The p-value indicates the probability of observing the relationship in the sample data if no true relationship exists in the population. Therefore a p-value of 0.065 means that it is less than 6.5% likely that we would have observed this relationship is there was no relationship between these variables in reality.

		Engagement	
		Dis-engaged	*Engaged*
Matching Names	Frequency	67	18
	Percent	95.71%	81.82%
Non-Matching Names	Frequency	3	4
	Percent	4.29%	18.18%
Total		70	22
		100%	100%

Note: Pearson's $\chi^2 = 4.60$. $p = 0.032$

Table 27 displays the raw frequencies and percentages of individuals providing matching versus non-matching names by Engagement. Just over 4% of those classified as disengaged provided non-matching names, compared to over 18% of engaged who gave different configurations of their names across attempts. More than four times the proportion of engaged participants gave non-matching names than did disengaged participants. As was the case with the non-matching email address, the relationship between providing non-matching names and engagement is so strong that, despite a very small sample size, a Pearson's Chi-Squared test of independence yields a test statistic with p<0.05. This means that the null hypothesis that providing non-matching email addresses is independent of Engagement can be rejected at the 95% confidence level.[16]

[16] Pearson's chi-squared calculated the probability of observing the distribution of frequencies in the data if there is no relationship in the population. The p-value indicates the probability of observing the relationship in the sample data if no true relationship exists in the population. Therefore a p-value of 0.032 means that it is less than 3% likely that we would have observed this relationship is there was no relationship between these variables in reality.

CHAPTER 7 - CONCLUSION

This paper introduced Mazetec, a scenario-based learning software-as-a-service web application. It described the originating idea and initial inspiration and motivation, followed by a literature review of the software's theoretical foundation. A domain analysis and a feature analysis were conducted, and solution concept was developed, the system was then designed and planned. Next, the technical implementation and studies were conducted. The results in this paper have demonstrated the benefits of the Mazetec platform.

In 2017, the IEEE created a new "learning engineering" discipline [9] that focuses on the design and engineering aspects of learning systems that are more efficient, engaging, and effective. As discussed in the results, the Mazetec platform allows subject matter experts to create time-limited branching scenarios quickly in a point-and-click environment to preview and deploy the scenarios with a link. A custom API was developed to prove system interoperability is possible. In order to expand the interoperability of the Mazetec system, I am planning on continuing development to implement OpenID, OAuth 2.0, xAPI (Experience API), and LTI (Learning Tools Interoperability). Further evidence suggested the mazes were a highlight in the online courses taken by learners and the analysis of the SAMHSA study found a subset of learners were engaged by the Mazetec SBL training. The Mazetec system is domain-independent and reduces the time and cost barriers to create, edit and deploy scenarios and analyze the results.

I will continue with the SAMHSA study and I am also planning to commercialize the Mazetec platform, so it can be used by as any researchers, SMEs or otherwise whom are interested or in need of easy to use branching scenario system.

REFERENCES

[1] P. A. Kirschner, J. Sweller, and R. E. Clark, “Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching,” *Educ. Psychol.*, vol. 41, no. 2, pp. 75–86, Jun. 2006.

[2] L. Song, E. S. Singleton, J. R. Hill, and M. H. Koh, “Improving online learning: Student perceptions of useful and challenging characteristics.”

[3] Pew Research Center, “Mobile Technology Fact Sheet,” *Internet, Science & Tech*, 2014. .

[4] Souto- *et al.*, “OpenCases: Case Studies on Openness in Education,” 2016.

[5] “State laws on suicide prevention training for school personnel,” 2016.

[6] H.-M. Huang, “Toward constructivism for adult learners in online learning environments,” *Br. J. Educ. Technol.*, 2002.

[7] J. Eich, H. Weingartner, … R. S.-J. of M. and, and undefined 1975, “State-dependent accessibility of retrieval cues in the retention of a categorized list,” *search.proquest.com*.

[8] Substance Abuse and Mental Health Servicess Administration, “The Garrett Lee Smith Suicide Prevention National Outcomes Evaluation Enhanced Training Component Implementation and Procedures Manual,” 2017.

[9] IEEE ICICLE., “Mission.” [Online]. Available: https://www.ieeeicicle.org/mission/.

[10] T. Connolly, M. Stansfield, and E. McLellan, “Using an online Games-Based learning approach to teach database design Concepts,” *Electron. J. e-Learning*, 2006.

[11] D. Dagger, A. O’Connor, S. Lawless, E. Walsh, and V. P. Wade, “Service-Oriented e-learning platforms: From monolithic systems to flexible services,” *IEEE Internet Comput.*, 2007.

[12] S. Aceto, S. Borotis, J. Devine, and T. Fischer, "Mapping and analysing prospective technologies for learning.," 2014.

[13] L. A. Diehl, E. Lehmann, R. M. Souza, J. B. Alves, R. Z. Esteves, and P. A. Gordan, "A serious game prototype for education of medical doctors and students on insulin management for treatment of diabetes mellitus," in *2011 IEEE 1st International Conference on Serious Games and Applications for Health, SeGAH 2011*, 2011.

[14] A. Diniz dos Santos, F. Strada, and A. Bottino, "Approaching sustainability learning via digital serious games," *IEEE Transactions on Learning Technologies*, 2018.

[15] L. Bacon, L. MacKinnon, and D. Kananda, "Real-time decision support in online training environments," 2015.

[16] S. Ludwig, N. Schuelper, J. Brown, S. Anders, and T. Raupach, "How can we teach medical students to choose wisely? A randomised controlled cross-over study of video- versus text-based case scenarios," *BMC Med.*, 2018.

[17] D. R. Godden and A. D. Baddeley, "Context-dependent memory in two natural environments: on land and underwater , British Journal of Psychology, 66:3 (1975:Aug.) p.325," *Br. J. Psychol.*, 1975.

[18] N. Ahituv, M. Igbaria, and A. V. Sella, "The Effects of Time Pressure and Completeness of Information on Decision Making," *J. Manag. Inf. Syst.*, vol. 15, no. 2, pp. 153–172, Sep. 1998.

[19] D. Y. M. Lin and Y. L. Su, "The effect of time pressure on expert system based training for emergency management," *Behav. Inf. Technol.*, 1998.

[20] L. Guo, "An exploration of dual systems via time pressure manipulation in decision-making problems (Doctoral dissertation)," *eScholarship, EBSCOhost*, 2017.

[21] M. Hahn, R. Lawson, and Y. G. Lee, "The effects of time pressure and information load on decision quality," *Psychol. Mark.*, 1992.

[22] D. Kahneman and A. Tversky, "The simulation heuristic," in *Judgment under uncertainty: Heuristics and biases*, 1982.

[23] L. Bacon, L. MacKinnon, and D. Kananda, "Supporting Real-Time Decision-Making under Stress in an Online Training Environment," *Rev. Iberoam. Tecnol. del Aprendiz.*, 2017.

[24] P. Lameras, S. Arnab, I. Dunwell, C. Stewart, S. Clarke, and P. Petridis, "Essential features of serious games design in higher education: Linking learning attributes to game mechanics," *Br. J. Educ. Technol.*, 2017.

[25] The University of Queensland, "Bloom's taxonomy of educational objectives," *University of Queensland, Teaching and Educational Development Institute*, 1996. .

[26] M. B. Carvalho, "Serious games for learning: A model and a reference architecture for efficient game development," 2016.

[27] W. Van Der Vegt, W. Westera, E. Nyamsuren, A. Georgiev, and I. M. Ortiz, "RAGE Architecture for Reusable Serious Gaming Technology Components," *Int. J. Comput. Games Technol.*, 2016.

[28] G. Petri and C. Gresse von Wangenheim, "How games for computing education are evaluated? A systematic literature review," *Comput. Educ.*, 2017.

[29] B. Cook and P. Twidle, "Increasing awareness of Alzheimer's disease through a mobile game," in *Proceedings - 2016 International Conference on Interactive Technologies and Games: EduRob in Conjunction with iTAG 2016, iTAG 2016*, 2016.

[30] P. Quinn, I. Lindsey, and I. Schankler, "Depression quest," 2013. [Online]. Available:

http://www.depressionquest.com.

[31] C. Moore, "Elearning example: Branching scenario," 2010. [Online]. Available: http://blog.cathy-moore.com/2010/05/elearning-example-branching-scenario/.

[32] D. Schwarz, "Widening access to virtual educational scenarios," 2016.

[33] A. Colaso *et al.*, "Memory Hierarchy Characterization of NoSQL Applications through Full-System Simulation," *IEEE Trans. Parallel Distrib. Syst.*, 2018.