Building a Platform for the Escape Game Community

COM3610 Dissertation Project

Simon Fish | Supervisor: Andrew Stratton

This report is submitted in partial fulfilment of the requirement for the degree of Computer Science with a Year in Industry by Simon Fish.

11th May 2020

Signed Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Simon Fish

Abstract

Escape rooms are physical, interactive experiences in which a group of participants must solve puzzles to escape a locked room, solve a mystery, or otherwise meet some goal in a particular timespan. The primary aim of this project is to build a tool for escape room maintainers. The original brief expressed intent to craft a networked solution for use within the escape room itself. Subsequently, this focus was changed by the literature survey, which concluded that implementation of technology inside escape games requires careful consideration. The end product is a web application functioning as a social network between escape room maintainers and enthusiasts, which meets requirements established by surveying and discussing with maintainers. This dissertation paper serves to document, and justify decisions made along, the process of development of this product.

COVID-19 Impact Statement

The COVID-19 outbreak meant that escape rooms were completely closed to business. This made face-to-face contact with involved escape room maintainers impossible. Maintainers were not in a position to discuss student projects due to the situation, which is universally jeopardising business. I had already engaged with maintainers in person and online. It became impossible to follow this up meaningfully. It was impractical to change the project's focus at the point of the pandemic, so a decision was made to continue supporting the project for after the lockdown order had been lifted.

Acknowledgements

Thanks to Andrew Stratton for supervising and providing a guiding hand during the project. Thanks also to the Lockup Escape Room in Sheffield for providing helpful insights and allowing me to trial an escape room with a group. Thank you to those working in Development at UKCloud for giving me the skills needed to build Blacklight—special mentions to Ben, Chris, and Oliver. Finally, my deepest thanks to my family, my friends at University of Sheffield Tea Society, and my mentor Dalbinder. All have strongly supported me this year.

Contents

Chapter 1

Introduction

Please note the following definitions:

Maintainers refer to those who own and run escape rooms. They are responsible for such things as building and maintaining the escape room, and running the experience for groups. In many cases, they are also responsible for elements of the escape room experience outside of the room itself, such as maintaining its image on social media.

Enthusiasts refer to attendees of escape rooms—particularly those who take a firm interest in them, regardless of their degree of experience.

Escape rooms are physical, interactive experiences in which a group of participants must solve puzzles to escape a locked room, solve a mystery, or otherwise meet some goal in a particular timespan. They are a phenomenon that has existed since around 2007 (Nicholson 2015) that forms a growing industry.

The concept of an escape room is applied in a variety of forms. Most commonly, escape rooms are permanent fixtures on which maintainers might run several different escape games. Other variations include portable escape rooms run in shopping centres (Gündüz 2018) and similarly challenging and captivating experiences that assist education (López 2019; Rouse 2017; Peleg et al. 2019; Beguin et al. 2019). Of late, the importance of virtual reality escape rooms and in-home escape rooms is on the rise.

The escape room industry ranges from enthusiasts-turned-maintainers, who run singular escape rooms for small groups, to major companies such as SCRAP, which deliver escape games to dozens, if not hundreds, of people. It is particularly notable that during the COVID-19 pandemic taking hold at the time of writing, maintainers are moving towards products that enthusiasts can use in their own homes.

Though many escape rooms live up to their name, the aim in an overwhelming 54% of cases is not to escape the room—goals range from investigating a crime or mystery to engaging with the supernatural or solving a murder (Nicholson 2015). The goal can be particularly unique in educational applications of escape rooms, in which the subject matter and goal are often inspired by a learning objective.

The goal in this project was to understand the needs of the commercial escape room industry, such that a product could be developed to remedy them. Such a product would aim to increase efficiency or expand the industry with new capabilities. With a strong understanding of this in mind, and feedback from the community itself, a product could then be designed and built to target some subset of these needs. These needs may be related to various issues, such as making sure a timer is visible to the participating group and maintainer, or to processes that currently take more time than necessary, such as posting photos of teams to social media (Lockup Escape Rooms 2019).

Research Questions

Two research questions were identified, which the literature review stage of the project is focused towards answering:

RQ1: What has been reported about concepts used in escape rooms applied in different environments?
RQ2: Based on this, what can be established as the requirements for escape games, independent of their environment?

Relationship between Project and Degree Programme

The project did not directly tie in with any of my modules this year. I acknowledged during research that my solution could potentially build on skills learned from modules such as *COM3505 Internet of Things*, should my solution have incorporated microcontroller hardware. In such an instance, I would likely have elected to use the ESP32 microcontroller out of familiarity.

My approach has, however, been influenced by modules from previous years. Particularly, my grounding in software development comes from modules such as *COM1001 Introduction to Software Development* and *COM3420 Software Hut*. The concept of agile processes was introduced in the former, though my experience with *COM390 Year in Industry* presented this in a practical manner that more directly inspired my approach. The limited time available in *COM3420 Software Hut* meant that many Rails best practices were not transferred through that module—instead, many of these came through to me during my year in industry.

My approach was not reminiscent of the Scrum process used in my team last year. Instead, I elected to move towards use of Kanban, which would allow me to be more flexible with my priorities in light of time constraints and changes. This created a challenge for me. My work during *COM390 Year In Industry* gave me a strong set of principles regarding software development—I wished to include industry-standard processes in my work, including continuous integration, vulnerability testing, and measurement of test coverage. Time constraints meant I needed to make compromises here—I could not ensure that automated tests covered a majority of the code, let alone track test coverage. In essence, my experience during my year in industry widened the scope of my capabilities and priorities alike, but I could not apply them to the fullest.

Chapter 2

Literature Survey

This chapter uses relevant literature to discuss the nature of escape rooms' use of technology. The survey identifies factors such as reliability and inspirations such as asynchronous gameplay. These go on to form guidance for strong implementations of technology in escape rooms. The definition of escape rooms was covered in the introduction; this chapter explores them in further depth.

Process

The following keywords were identified for use when searching for material to reference and discuss in the literature survey.

- escape room / puzzle hunt
- maintainer / owner / host
- implementation
- software
- virtual reality
- education / classroom

This led to the following searches.

- escape room owner
- escape room host
- escape room maintainer
- escape room software
- escape room education
- escape room classroom
- virtual reality escape room
- portable escape room

The search was conducted using the Google Scholar search engine. Approximately 40 results from each search were checked. On this basis, approximately 320 papers were explored, not accounting for overlap.

Some additional searches were made based on discoveries such as escapED (Clarke et al. 2016) and Breakout EDU ("Breakout EDU" 2020) to see where they had been applied. The white paper by Nicholson (2015) proved to be a key point of reference across much of the relevant literature.

Survey

In order to explore the priorities of escape rooms, it can be useful to first study their history. Nicholson (2015) documents escape rooms as the culmination of a variety of media. He identifies puzzle hunts as team-based problem-solving challenges. Treasure hunts and generic team-based problem-solving challenges, such as those used in corporate team-building exercises, appear most similar to escape rooms. Live-action roleplaying (LARP) is another such inspiration (Nicholson

2015). Both escape rooms and LARP can provide entertainment through immersion in a story as the *"hero"*, a factor that players enjoy (DuPlessie 2013). Nicholson presents the precursors to escape rooms in further depth in his paper—Figure 2.1 summarises these.

Figure 2.1: Nicholson (2019) presents the precursors to, and inspirations for, the escape room phenomenon in this diagram. Adapted with permission from Nicholson, Scott. 2015. "Peeking Behind the Locked Door: A Survey of Escape Room Facilities." *White paper available online at http://scottnicholson.com/pubs/erfacwhite.pdf.*

Escape rooms have grown popular in many locations across the world as a recreational activity (Nicholson 2015; Stasiak 2016), even serving as a tourist attraction (Dilek and Kulakoglu Dilek 2018). Nicholson reports that the *Real Escape Game* by SCRAP was the earliest well-documented activity branded as such. SCRAP has gone on to develop escape rooms at a much larger scale than the typical escape room. Its *Real Escape Game Event* serves *"hundreds or thousands of players in a large space"*—a far cry from the average team size of 4.58 people (Nicholson 2015). SCRAP remains one of the longest-serving purveyors of escape game experiences. Technology plays an active part in rooms they have offered, such as *Pacific Rim: Shatterdome Defenders* (SCRAP 2018).

Escape room maintainers apply technology in varying ways, both inside and outside the experience. On the outside of the escape room experience, maintainers implement leaderboards, share team photos to social media, and interact with the team via a screen or handheld transceiver in the room. Inside the room, entire puzzles are driven by technology in rooms that integrate it comfortably, and it can drive visual effects such as lasers and smoke. These implementations can be inventive and support rooms in novel ways; *The Gallery*[1] (visited in July 2019), for example, uses a hidden camera to take a photo of attendees, applies filters, and displays the photo among works of art. However, there is a general aversion to the use of technology in escape rooms, for a variety of reasons.

In discussion with Lockup Escape Rooms (2019), it was suggested that the time investment, reliability, and necessary expertise are some of the greatest contributing factors against the use of technology in escape rooms. Reliability particularly guides this; a break in the flow of the escape room experience can shatter participants' immersion and lead to negative reviews (Lockup Escape Rooms 2019). Application of technology in escape rooms thus brings with it some uncertainty.

Escape rooms seek to develop immersion. Technology can bolster or break this, depending on the implementation. Immersion can be defined as captivating the participant and making them feel invested in a situation or story. It can be achieved through the use of theatrics such as special effects, acting, props, and room design. In his example, DuPlessie (2013) creates immersion

[1] https://escapist.nl/en/

through sound and steam to cover for his room's rotating mechanic—distracting the participants from this gives a sense of reality to the situation they are in, immersing them in the environment and story fully. DuPlessie is critial of digital, rather than physical, interaction in escape rooms, citing the importance of immersion, and recommends movement away from what he calls the *"glowing rectangles"* as our medium of choice.

Physical interaction can be considered a foundation point of many escape rooms. Inclusion of props, costumes, and mechanics can greatly improve immersion. 70% of escape rooms employ a search for physical objects as part of the experience (Nicholson 2015), which supports this. However, escape rooms in the classroom tend to forgo this. Rooms of this nature may prioritise having multiple participating groups in the room at one time. They may also use less props in order to cut down on the time it takes to reset the room.

Prioritising multiple attendees over immersion, a study by López-Pernas et al. (2019) took place in a computer lab with groups at computers. This helped to *"decrease the time investment for the course staff"* who were running the exercise. This study also reveals a (context-specific) benefit to using smaller groups. The context of programming allowed the pairs of participants to reap the benefits of pair programming (Williams and Upchurch 2001; McDowell et al. 2002). In cases such as these, forgoing immersion for additional educational benefit seems worthwhile and can make a marked impact. The experience likely contributed to the *"statistically significant ($p < 0.001$)"* 23% increase in passes.

A study by López (2019) organised its puzzles in a manner whereby puzzles could be completed in any order, allowing multiple groups to attend the room at once. One example puzzle given in the study was an exercise likely done on paper. This would allow all groups in the class to attempt the puzzle without it needing to be reset, but the absence of physical interaction or props would likely reduce participants' immersion in the scenario.

Nicholson (2018) suggests that an *"ongoing narrative"* between activities can remedy this—this would *"empower them to be more engaged than if the games are over in an hour"*. Dietrich (2018) trialled a similar escape game in a classroom environment, to which 62% of participants with prior escape game experience *"felt a similar sensation in the game presented here"*. This suggests that strong and imaginative implementations of escape rooms in the classroom can preserve the level of immersion traditional escape rooms often display with adequate results.

Technology can enhance these implementations of escape rooms, preserving some level of immersion by providing something interactive and physical. Ross and Bell (2019) applied a decoder box developed for use in educational escape rooms (Ross 2019). Ross (2019) reports that "the cost of a single complete system is $30.00 AUD" (around £16 at the time of writing) and that it is reprogrammable to handle several solutions. This arguably makes it an inexpensive and reusable solution for classroom escape rooms. Participants in studies that have used the decoder box prefer *"the user interface of something physical rather than a computer based abstraction"* (Ross 2019) or *"paper and pen or an app on their phone"* (Ross and Bell 2019).

Technology, particularly monitors, may be avoided in the escape room due to the issue of visibility. Discussion with Lockup Escape Rooms (2019) highlighted that visibility of the puzzle to the entire team should be a priority. Nicholson (2015) warns of the danger of removing just one player from the *"mental space"* of the team—this is particularly an issue if a puzzle requires one person to work at a computer alone. Even if a screen-based puzzle calls for multiple people, the available space and the visibility of the screen dictates how many members of the group can interact with it. Unless monitors are suitably placed and large enough to be viewed by a full party, the whole team may not be able to interface with a puzzle that applies technology.

A greater number of monitors, or larger monitors that allow for multiple users to interact with the puzzle (e.g. a multi-touch screen), could both counteract this. Having said that, these come with both a literal and figurative price—the cost of the resources themselves, and the space within the escape room in which they can be implemented. As such, these factors make them difficult to implement with repeat effectiveness across different escape rooms.

Rouse (2017) applied the idea of using a computer in the classroom using a game loaded from a memory stick. This application seems understandable; assuming a younger audience, students would be likely to have some basic level of expertise in, and enthusiasm for, handling computers.

However, in practice, this application of technology brings to mind the image of a small group of people crowding around a screen, embodying a negative example in following this law of visibility. Nicholson (2018) supports this idea, suggesting that games using a screen do not emphasise the *"face-to-face contact"* players share in live-action games.

Other forms of technology can still enable this *"face-to-face contact"* that escape rooms prioritise. The Nintendo Switch video game *1-2-Switch*[2] often makes a point of pulling away from the screen. The majority of its minigames rely on what is done physically with the Joy-Con controller. Players are often instructed to face their opponent directly, and may rely on audio cues and/or precise vibration in the Joy-Con during gameplay. It is an effective example of how to apply technology in a way that does not lock singular players into staring at a screen.

Poor implementation does not necessarily mean technology should not be excluded from escape room environments outright. Instead, technology has the potential to inspire change in escape rooms by targeting the time it takes to reset a room back into a playable state. Escape rooms and technology are inherently linked— digital escape-the-room games such as *Myst* precede and inspire physical escape rooms (Nicholson 2015). In these games, partipicants solve similar puzzles within the limits of the player-character's capabilities, applying and sometimes combining an inventory of objects to create new tools for use in their escape.

These forms of escape room can be reset instantly by simply resetting the game. Whether this is done by restarting an attempt or restarting the program itself, it is often trivial compared to how long it takes to reset escape rooms. In discussion with Lockup Escape Rooms (2019), he established that resetting physical escape rooms can often take as long as 15 minutes, and that it is something escape room maintainers generally seek to optimise; the shorter a reset takes, the more time is available to welcome customers.

Quick resets are a priority in many different situations. A Turkish company, Tuzak, develops portable escape games that are run in shopping centres (Gündüz 2018). Their choice to move to shopping centres was likely made in an effort to capture new customers, making it all the more important that the room could be reset quickly. Several studies cover the use of escape rooms as a means for education (López 2019; Rouse 2017; Peleg et al. 2019; Beguin et al. 2019), which would also prioritise this to fit within timetabled school hours. The aforementioned application by López (2019) allowed multiple groups to tackle the escape rooms at the same time by removing the need to reset puzzles entirely—students needed to combine the solutions to puzzles on different worksheets to complete the room.

The field of virtual reality is capable of creating escape rooms that bridge the boundary between the digital (e.g. *Myst*) and the physical. The maintainer of Lockup Escape Rooms (2019) theorises that virtual reality (VR) escape rooms such as *EXIT VR*[3] may be the next stage for the industry, which allow immersive rooms to be created while effectively eliminating the issue of resetting the room as above. This would bring the escape room cycle full circle, reincarnating the modern wave of physical escape rooms in the digital form that inspired them.

Pendit et al. (2017) exercised this in the creation of their virtual escape room *The Last Breakout*. This application revealed some potential caveats that may be visible in the creation of VR escape rooms. Unreal Engine 4 assisted in making the project feasible, but limited knowledge of how to use it resulted in difficulties. These included motion sickness from excessive movement, user experience as regards knowing when they might have picked up an object, and display of reading materials (Pendit et al. 2017).

These could be remedied with more effective VR development experience, but user interface issues remain something that cannot be tackled without the correct principles. In a study by Smith and Ericson (2009), it was found that the context of virtual reality clouded the line between a video game and a simulation of reality—one child reportedly asked *"How do I kneel?"* when inside the simulation. This misunderstanding could be attributed to how crouching in video games works—it is most often toggled by, or activated by holding, a button input. While Smith and Ericson (2009) acknowledge the *"built-in capability for higher levels of actual human–computer interaction than traditional video games"*, this boundary should not be ignored as the escape room industry explores VR.

[2] https://www.nintendo.co.uk/Games/Nintendo-Switch/1-2-Switch-1173186.html
[3] https://exit-vr.de/en/

While the escape room industry investigates the *"glowing rectangles"* DuPlessie (2013) warns against, the video games industry that lent it inspiration sometimes takes small strides to recede from them. One strategy that both industries are applying is asynchronous gameplay. In asynchronous gameplay, two parties have different experiences that interlink to achieve the same goal. Each industry can learn from the other's examples in applying this concept.

The video game *Keep Talking and Nobody Explodes*[4] is a popular example, in which one player must defuse a bomb while the others guide them using the bomb's manual[5]. Conceptually, the game shares fundamentals with escape rooms—players work together to solve puzzles through logic and communication. The communication aspect is of most importance to this point: the game's asynchronous gameplay depends on communication for success.

Escape rooms also exemplify asynchronous gameplay. The escape room *Meltdown* (Lockup Escape Rooms 2019) begins with most of the party in individual chambers, with the designated 'leader' communicating from outside to coordinate the group. An escape room study by Clarke et al. (2016) demonstrated a similar mechanic, using laptops with Skype to facilitate communication between two groups in separate rooms. One of the puzzles in *The Gallery* sees groups pulling ropes to lower a metal ball through a hole in a cabinet. This requires two parties—one to pull the ropes, and another to monitor the ball as it is lowered—as the cabinet cannot be seen from where the ropes are.

Escape rooms already apply asynchronous gameplay, but *Keep Talking and Nobody Explodes* demonstrates that with technology, it is possible in almost any location through the use of portable gaming platforms like the Oculus Quest or Nintendo Switch. This is all the more important during the COVID-19 pandemic, in which companies are turning to other measures to continue the escape room experience. SCRAP, for example, is even resorting to livestreamed escape room experiences[6]. This area seems to be ripe with potential in such difficult times.

In summary, technology brings a variety of benefits, from quick resets when used as part of a puzzle to novel ideas that may not otherwise be possible. Many ideas and inspirations have been discussed here. The strength of escape rooms as an educational tool has also been demonstrated here, which is worth consideration when building for the escape room industry. However, there is a debate as to whether the use of technology in escape rooms is always a viable option. Escape room maintainers value reliability, with negative reviews being the consequence for ill implementation (Lockup Escape Rooms 2019). One of the greatest points of contention is immersion. Care must be taken to ensure that if the implementation separates one player from the group, it is applied in an engaging manner. Excessive reliance upon screens is something that should be avoided in the name of immersion.

It is clear that a balance should be maintained between these factors, and as such, it is difficult to approach a one-size-fits-all solution for use inside an escape room. However, it is clear that technology can advance escape rooms with careful implementation. In particular, it can be used outside of the experience itself in matters such as booking rooms, keeping a leaderboard, or advertising through social media or otherwise. Software developed by Buzzshot (2018), Xola, Inc. (2018) and Resova Ltd (2020) (eponymous in each case) capitalise on the potential for this—Buzzshot is a general package covering all three listed areas, whereas the others focus primarily towards bookings.

This project goes on to define advertising over social media as a feature of interest to maintainers. It was identified during research that the escape room community may not have a unifying platform. The project sought to meet both of these needs by creating a platform shared by escape room maintainers and enthusiasts.

[4]https://keeptalkinggame.com/
[5]https://bombmanual.com
[6]https://www.facebook.com/events/359013048391029/

Chapter 3

Requirements and Analysis

The initial specification for the project was a set of networked microcontrollers/systems-on-chips that would unite to track the escape room experience. This was abandoned as the idea of building an escape room, even in the form of a prototype, was out of scope. To seal this decision, my literature survey concluded that digital uptake in escape rooms may be middling due to the inherent risk and time involved.

To find some insight into the challenges escape room maintainers face, I met with the maintainer of the Lockup Escape Rooms in Sheffield on the 20th November. They had around two years' experience in doing so at that time. We discussed some areas that could be targeted and some ideas that I had previously prepared. This would go on to inspire a survey I would distribute to the wider community of maintainers (Appendix II).

The survey used the ideas The Lockup Escape Rooms and I had discussed. The Facebook group created by Nicholson (2015) would serve as the audience for the survey. It is for escape room enthusiasts worldwide, and includes a smaller number of maintainers. I chose to survey this group as it appeared to be a central hub for what seemed, to an outsider, to be a sparse online community. The survey needed to be agreed to by the group moderators before being sent out. Hearing the opinions of an escape room maintainer helped me to anticipate the potential results and begin planning.

These results were found from the seven responses received:

- Four maintainers already share photos online, but the remaining three are all interested by the prospect of it
- Memberships, in-character communications, advertisement among other escape rooms, and participant metrics are all of interest to at least five members of the group
- Memberships are agreed upon by the five interested parties as something that should not be done without the involvement of technology
- Ease of use *always* influences each maintainer's decision when investing in technical solutions to problems
- Reliability *always* influences all bar one of the group's decisions in the above

Notable comments included that one maintainer sought to *"strictly limit/eliminate the use of"* technology in their escape rooms. The following reasoning was given for these decisions:

- *"Losing revenue"* in the event of failure without failover. Another maintainer agreed that *"dependability and available work around (sic)"* were agreed to be of importance. This aligned with my discussion with Lockup Escape Rooms.
- *"Each room only lasts 1-1½ years"*, reducing the maintainer's budget for props for each room. *"[The maintainers] do not invest as much into props unless they can be reused in another game"*, meaning investment in technology as a showpiece is generally avoided in that particular offering.

I was able to elicit a direction from these results—the idea of a social network shared by escape room maintainers and enthusiasts allowed me to potentially target several of these factors at once.

Primarily, this social network would allow escape rooms to advertise among others of their ilk and share photos. I concluded my analysis of the results by setting a goal statement.

Goal Statement

To design and build a system that:

- allows escape room maintainers to share photos with their community
- allows escape room maintainers to advertise among other escape rooms
- allows escape room enthusiasts to discover new escape rooms to tackle
- allows escape room enthusiasts to track which escape rooms they have cleared

---

With these as an initial guide, I began to set requirements using the MSCW system (see Appendix I), aiming to complete all defined as **Must**- and **Should**-Have by the end of the project. User stories were written with parts of the goal statement ('epics') in mind and estimated according to their complexity using a modified Fibonacci scale.

In agile software development, complexity estimates are preferred to time estimates as the former are easier to settle upon (Karlesky and Voord 2008). Stories with estimates any greater than 13 would need to be broken down into smaller stories. This was a principle employed by my team and I on my year in industry. Radigan (2018) suggests a differing scale and limit, but in either case, it is a priority to keep stories small and manageable. Velocity is of the essence, and smaller tickets assist with that.

Another factor that was kept in mind when creating stories was keeping them open to interpretation. Stories, as often as possible, would need to describe what the user would wish to achieve, as opposed to what the developer working on them should aim to do. Framing stories from this perspective allowed their implementation to remain open to change and interpretation. By not enforcing how a ticket should be implemented until it is worked on, it could be ensured that the right tool for the job would be chosen.

As mentioned in the previous chapter, I have had more hands-on experience with Scrum than with Kanban. Kanban is employed when more flexibility is desired. It prioritises throughput and encourages a "culture of 'done' " by enforcing work-in-progress limits (Rehkopf 2018). With this in mind, it would be difficult to set more than two development milestones. Treating each MSCW category as an epic and taking the amount of time necessary for work on this report and my other modules into account, I aimed to work on the project for a month, devoting two to three weeks to each milestone.

I figured that each milestone would take a similar amount of time—while Rails' built-in generators would likely ease the burden of scaffolding the initial functionality, setting that groundwork in the right way would take slow and careful design decisions. These would dictate the flow of the rest of the project. In the end, the following deadlines were decided:

- **Development commences** March 12th
- **Must milestone completed** March 26th *(actual completion date: March 24th)*
- **Should milestone completed** April 16th *(actual completion date: April 13th)*

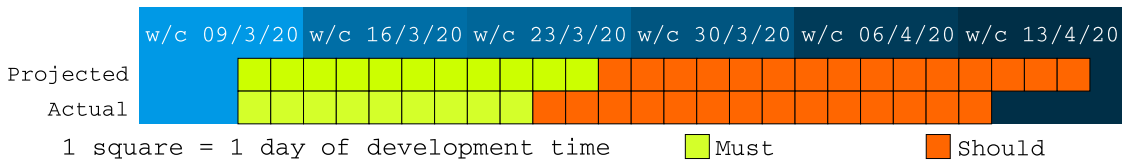

Figure 3.1: A Gantt chart representing my set goals versus the final completion dates.

After setting my focus, I decided my solution would take the form of a web application. Ruby on Rails was chosen as the development platform. The QOC analysis below was the source of this decision, in which I weighed up other potential candidates such as Iron (Rust), ASP.NET MVC 5 (C#), and Next.js (JS).

Table 3.1: QOC chart (MacLean et al. 1991) for choice of stack, with priority values and totals revealed.

	Priority	4	5	4	2	4	3	3	5
Total	**Tool** **Criteria**	Familiarity	Documentation	Stability	Linux support	Docker resources	Developer tools (generators)	Community	Development cycle
145	Rails (ERB)	5	5	5	5	5	5	5	4
108	Next.js	3	4	3	5	5	1	2	5
109	ASP.NET MVC 5	2	5	5	1	5	5	3	2
66	Iron	1	2	1	5	3	1	1	4
130	Rails (React)	4	4	5	5	5	3	5	4

I chose to prioritise documentation and speed of development cycle over all else. With sufficient documentation and fast compile times, I would be able to work quickly and effectively. Stability and familiarity also factored into my decision strongly—I would need to be sure that the language interface would not radically change during the course of development, and that I would feel comfortable with the language chosen. Docker resources were also prioritised so that less work would be necessary to deploy the application in a container in a production environment.

Originally, I weighed ERB and React against one another as view engines. This estimation was taken on the assumption that I would use React even where interactivity and state management were not required. In hindsight, this approach would have slowed development. Instead, my final decision to apply React only where functional improved the quality of the application without hampering development time.

However, doing so came at the cost of accessibility. With wider prior knowledge, perhaps this may not have been the case—the web design strategy of progressive enhancement (Champeon and Finck 2003), in combination with use of the WAI-ARIA standard ("Accessible Rich Internet Applications (Wai-Aria) 1.1" 2017), could have been used to mitigate this from the outset. I was not aware of progressive enhancement as a priority during development. (Herlihy 2013) investigated that 1.1% of `gov.uk` users were not getting JS enhancements at that time—though this may have changed in either direction over time, it is still a significant figure and signifies the importance of the need for `noscript` fallbacks.

Chapter 4

Design and Architectural Decisions

This chapter discusses choices I made with regards to design methodology and choice of approach.

Atomic Design

In summary, Atomic Design[1] serves as *"not a linear process, but rather a mental model to help us think of our user interfaces as both a cohesive whole and a collection of parts* at the same time*"* (Frost 2016). Atomic Design guides its users to classify components by their scale and function, promoting reuse of smaller components to form greater wholes.

This is best expressed with an example. A component classed as an atom would be a `div` with its inner text mapped to its state and passed in through props—it does not control or communicate with any other components. A molecule may contain this element and a `button` that, when clicked, changes the state of the controllable `div` to something predefined. While Atomic Design in full uses a five-level hierarchy—atoms, molecules, organisms, templates, and pages—I did not employ organisms or templates in my use of the methodology.

The rules of Atomic Design make it easier to enforce the Single Responsibility Principle (Martin and Martin 2006) and segment code in a way that makes it more maintainable. This was the strongest factor in my decision to employ it. However, while it served useful in that regard, it was at times difficult to ascertain what should be an atom and what should be a molecule. In the final codebase, some components are considered atoms where they should be molecules.

Atoms were designed with the intent of taking in one or several Rails objects as props, and representing those. Molecules would comprise atoms, and in some specific cases use API calls to retrieve the objects to display. Both of these would be used for rendering as part of a page by the Rails templating engine.

I was introduced to the concept by an internal development team during my year in industry. Personal projects I have taken on in React thus far were all in motion before I was introduced to this, so I saw Blacklight as an excellent opportunity to apply it.

Choosing an Authentication Service

It was essential to choose the best authentication service for the project's use case. Allocation of this dependency would strongly affect development time, for better or worse. Personal familiarity lies with Keycloak in this instance—on my year in industry, I gained a thorough knowledge of it. Other options explored included Dex, ORY Hydra, and Auth0. Auth0 differs in that it is authentication as a service—it hosts its own authentication instance for one to use and manage through their platform. The rest are self-hosted. Self-hosting was a trade-off that was almost

[1] https://atomicdesign.bradfrost.com/chapter-2/

immediately acknowledged—control and convenience were in the balance. Though I would usually elect to favour the first option, convenience and stability drove the decision.

Table 4.1: Comparison of potential options. Regarding Devise compatibility, `:database_authenticatable` is an argument to `ActiveRecord::Model#devise`. The rest are Ruby gems (packages) that extend the OmniAuth protocol to provide support for those authentication methods.

Option	SaaS?	Documentation	Provides frontend?	Devise compatibility
ORY Hydra	No	Good	No	`omniauth_openid_connect`
Keycloak	No	Familiar	Yes	`omniauth_openid_connect`
Dex	No	Very good	No	`omniauth_openid_connect`
Auth0	Yes	Very good	Yes	`omniauth-auth0`
Devise (database)	No	Very good	Yes	`:database_authenticatable`

Keycloak dictates the database structure and must be talked to through its API. This leaves control out of the developer's hands, which is a problem from a software architecture perspective. If the developer is responsible for self-hosting the solution, they should be able to diagnose problems with the database. They should also be able to optimise its performance not just through additional resource, but through manual changes such as indexing. Conversely, Devise's user model is dictated by what the user generates, and the corresponding database table can be changed in line with other Rails models, making it preferable.

Hydra presents itself as being a thinner and faster alternative which does not serve its own frontend. This is positive, but demands that the developer create their own frontend—combined with unfamiliarity towards Hydra, this would use more time than necessary. Were time not an issue, Hydra may had far stronger consideration.

In the instance that Devise vulnerabilities were revealed and patched in Ruby, Blacklight would need to be upgraded in line with this. Using OmniAuth with Devise serves as a thinner layer over Devise and allows developers to easily migrate authentication providers, should they wish to. Still, if breaking changes are introduced in Devise, an OmniAuth provider would potentially delay upgrading Devise, as it would need to be changed to work with this new version.

Auth0 was chosen as the project's authentication provider, albeit as a compromise in the name of stability and convenience. One benefit is that it is externally hosted with a served login page, saving further frontend development and creating one less self-hosting dependency. Deployment also became much easier as a result of this choice—Blacklight could be deployed through Heroku buildpacks without needing to run a separate authentication service on another dyno.

In this instance, the choice to use Auth0 as an authentication provider was valid for several reasons. Creating a dependency of large size would mean that in the event of a situation that called for further development, an extortionate amount of time would need to be expended. Such situations could include, for example, a security flaw or a significant update. Centralising this on two major dependencies—Auth0 and the `omniauth-auth0` gem, which are both managed by Auth0 Inc.—puts it in far safer hands.

Content Input and Presentation

It was a priority to allow escape game maintainers a strong degree of control over their profile through intuitive means. They would be able to make their escape game more visible on the site through searches by entering details such as its relative difficulty and location. In particular, a rich text description was identified as an essential feature that would help maintainers to make listings that accurately represent their escape games.

This would include both text formatting and external links. Inspiration was taken from Kickstarter[2], a crowdfunding site which allows rich-text descriptions. Campaign runners use these to good effect with inline images acting as headings. Inline images were decided to be out of scope during development, but rich text was still essential to allow maintainers to bring enhanced structure, character and personality to their descriptions.

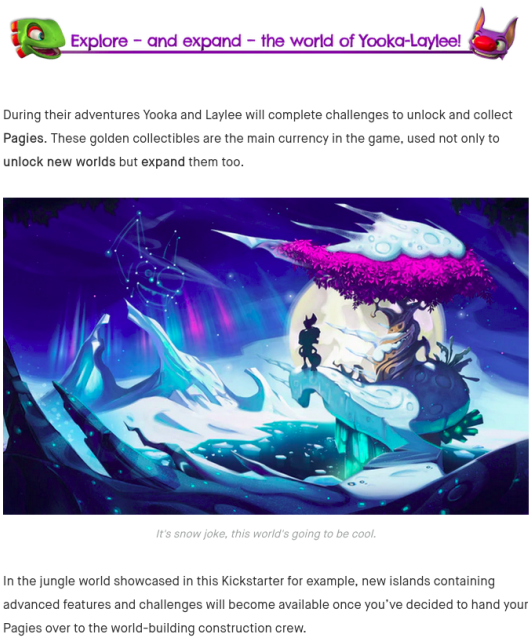

Figure 4.1: A rich-text description from Kickstarter, showcasing use of bold text and inline images with captions. Playtonic Games, viewed on April 17, 2020 "Yooka-Laylee - A 3D Platformer Rare-vival!" (Screenshot by author)

Rails delivers rich text inputs and storage through its ActionText extension, which was included in version 5. Restrictions on access to ActionText features were not identified—such restrictions would be vital in ensuring a level playing field on the site for all maintainers. ActionText was thus foregone in favour of Markdown, as Markdown engines for Rails such as `Redcarpet`[3] control access to features such as HTML backwards compatibility and the inclusion of inline images.

In selection of Markdown as the engine to drive rich text for Blacklight, it was strongly acknowledged that users may find learning Markdown "prohibitively difficult" (Ovadia 2014). However, Markdown is employed on blogging sites such as Tumblr, Reddit and WordPress (Ovadia 2014), with which I expected escape room maintainers may have some familiarity.

A minimal, yet intuitive, approach was built (Figure 4.2). This used a live preview, which showed users how their input would be rendered on `escape_game#show` for their escape room. A link to CommonMark's guide to Markdown[4] is shown beside the input field, which gives a quick overview of Markdown syntax.

[2]https://kickstarter.com
[3]https://github.com/vmg/redcarpet
[4]https://commonmark.org/help/

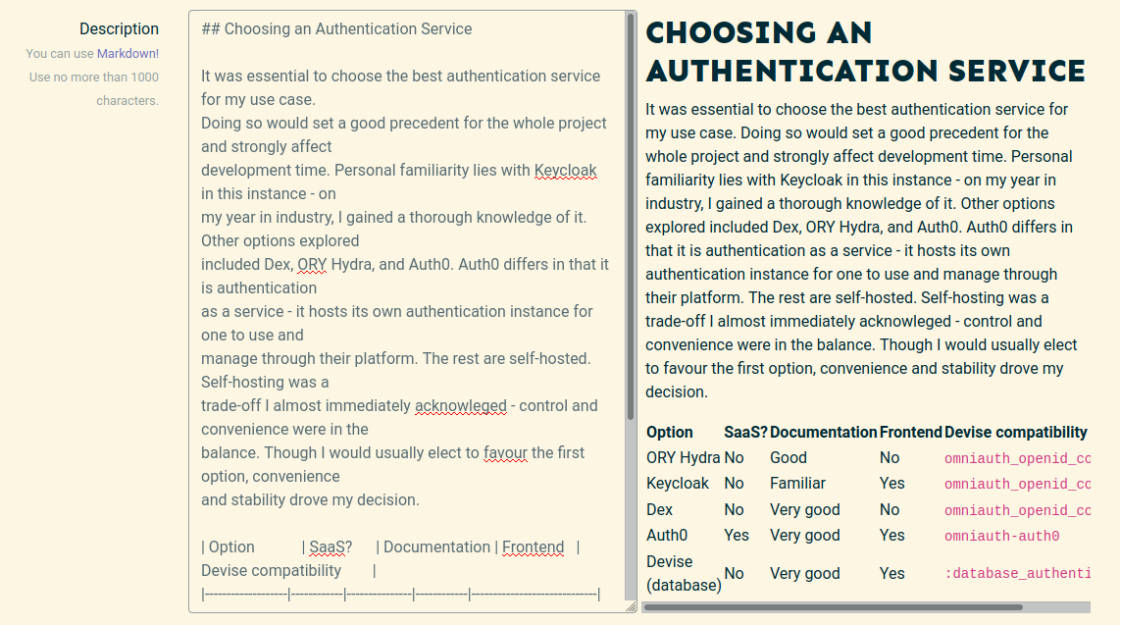

Figure 4.2: The Markdown preview field as at commit `fd1d6fd`.

Chapter 5

Implementation and Testing

My commitment to emulating industry best practices carried through to implementation. I felt it vital to use and work with such things as continuous integration servers, containerisation, and linting tools to maintain code quality and readability. Of course, none of these decisions were taken lightly.

While there is still some discourse over whether containerisation in development environments is entirely appropriate, I chose to use it to maintain some degree of parity between running locally and running in CI. Another reason why I made this choice was that popular production platforms such as Amazon Web Services are backed by containers—running the app using containers sets a good precedent to running it on a platform like Kubernetes with the right expertise.

Continuous integration tests code at each commit. I was able to configure CircleCI to build the Docker image used locally, and lint and test the code—this combined well with my use of the GitHub Flow[1], in which I would aim to get both tests and linting passing before merging a development branch into the `master` branch. Towards the end of my development cycle, when I had deployed to Heroku, I would also run through a set of checkout testing steps I devised as a manual end-to-end regression test.

Automated Testing

I aimed to use test-driven development to the best of my ability. Primarily, this meant securing controller routes and making sure that novel routes outside the standard Rails scaffold functioned as expected. This included:

- additional routes created beyond a standard Rails scaffold, such as those that make the current user a maintainer/enthusiast or those that remove image associations from a `Clear` or `EscapeGame` object
- secured routes—tests were created that challenged whether some attacking user could change a user ID in a form to steal ownership of a record
- some validations, such as those relating to limits on the number of image uploads for `EscapeGame`s and `Clear`s.
- some queries, particularly those relating to user privacy and escape room hiding.

SimpleCov was the standard on my year in industry with regards to coverage testing. I would have employed this and aimed for at least 80% test coverage across controllers, models and helpers, had time allowed for it.

The RSpec test suite ran at each commit on CircleCI's platform, halting in the instance of a failed Docker build, linting failure, or failed test. Though the environment in which I ran tests locally reflected its configuration, continuous integration served to ensure that the `master` branch continued to pass and was stable.

[1] https://guides.github.com/introduction/flow/

There were instances in which merging was delayed. CircleCI limits usage and fails all builds that fall outside a given quota. In these instances, I would either wait until my credits with CircleCI had reset and rerun the build, or trust that running the tests locally qualified instead and merge. In the event of a CI failure in an industry scenario, this would be discussed in the development department—generally, CI serves as a strict gatekeep to `master`, so in the majority of cases, development departments would resolve not to merge to `master` until their CI server functioned again. Due to time constraints, I was not quite so strict around it.

As I expected, this created a difficulty. With `master` having been changed without being monitored by regular CI runs, new issues had reached the branch—particularly, absence of the Rails master key meant that encrypted credentials couldn't be accessed during the application's initialisation cycle. I was able to fix this and proceed, but in an industry environment with several developers merging to `master`, this could have caused a significant slowdown in development due to potentially overlapping changes.

Below, an example from `spec/controllers/clears_controller_spec.rb` is featured. This is one of the automated tests in the application's RSpec suite, which is run by CircleCI at each commit. These tests ensure that none other than the creator of a clear can edit it—the helper method `random_user`, when given an `owner`, selects or creates a user other than the `owner`. This user becomes what the tests call the `attacking_user`—they are signed in and try to act against the `Clear` record. Both instances should raise a `404` error. `404` is generally returned across Blacklight when the user does not have access to the record in question, and is given back in the instance of an `ActiveRecord::RecordNotFound` error.

```
RSpec.describe ClearsController, type: :controller do
  before(:each) do
    @clear = create(:clear)
  end

  it 'does not allow users to edit others\' clears' do
    clear = @clear
    attacking_user = random_user(owner: @clear.user)
    sign_in attacking_user
    expect do
      put :update, params: {
        id: clear.to_param,
        clear: attributes_for(:clear).merge(user: attacking_user)
      }
    end.to raise_error(ActiveRecord::RecordNotFound)
  end

  it 'does not allow users to delete others\' clears' do
    clear = @clear
    attacking_user = random_user(owner: @clear.user)
    sign_in attacking_user
    expect do
      delete :destroy, params: {
        id: clear.to_param
      }
    end.to raise_error(ActiveRecord::RecordNotFound)
  end
end
```

Focused Frontend Testing

I omitted frontend testing that strayed too far beyond scaffolded tests due to time constraints. However, image attachment forms required testing to make sure that they behaved as users intended. One such case is detailed here.

Rails 6.0.0 introduced a change whereby newly-uploaded attachments replaced existing attachments,

as opposed to being appended to the existing group of attachments. Pull request #36716[2] against the `rails` codebase describes the difference between `rails`' behaviour before and after 6.0.0 in this situation. Blacklight was generated as a `rails` 6.0.2 application, and so needed a workaround for the new behavior. This was to include the signed IDs as hidden fields.

The test below checks that if an image is attached to a record, a hidden field appears for it in the form when it is rendered.

```
it 'doesn\'t overwrite files on update' do
  # Attach an image so one already exists.
  image_to_attach = File.open(
    Rails.root.join(
      'spec',
      'fixtures',
      'files',
      'escape_game',
      'SSBU-Big_Blue.png'
    )
  )
  @clear.images.attach(
    io: image_to_attach,
    filename: 'original_image.png',
    content_type: 'image/png'
  )
  render
  # Make sure there's a hidden field for the image that already exists on the
  # clear
  assert_select 'form[action=?][method=?]', clear_path(@clear), 'post' do
    assert_select 'input[multiple=multiple][type=hidden]' \
                  '[name=clear\[images\]\[\]]'
  end
end
```

However, my use of React components for major features like the Explore page means that swathes of the app are untested.

`react-rails` documents component testing against rendered Rails views[3]. Stateless components such as `Avatar`s would be better tested this way—all that would need to be asserted is that they are rendered with the correct props. To state that more generically, the assertion is that the correct arguments are passed to them at page render.

Some components are dependent on asynchronous functionality—the Explore view makes a request to the app's `/explore.json` endpoint on first render, and again afterwards. In these situations, directly employing Facebook's Jest[4] would be preferable.

Automated Regression and Checkout Testing Against Production

Checkout testing usually comprises a series of steps taken from an end user perspective against a production instance. It ensures that end users can continue to use the application as intended by checking that all features of the experience work as intended.

During my year in industry, checkout testing after deployment was not automated. Developers sought to automate the process so that deployment might not consume as much time. Libraries such as Puppeteer[5] can be useful for such a purpose—previously, I have used Puppeteer to create PDF forms of webpages repeatably[6].

[2] https://github.com/rails/rails/pull/36716
[3] https://github.com/reactjs/react-rails/blob/d5da11129459cd75fd003c75319b1f7440c37322/README.md#test-component
[4] https://jestjs.io
[5] https://github.com/puppeteer/puppeteer/
[6] https://github.com/boardfish/CV/blob/ed664d0e87e4d0ec1d6afab8214a5753e033669e/topdf.js

I have documented checkout testing steps in Blacklight's README, with clear indication that if the project were to continue, checkout testing would be automated. If possible, I would also follow up deployment with automated checkout testing at every instance. The checkout testing steps are replicated below.

Checkout Testing Steps

Check that the...

- ...homepage renders correctly, particularly with regards to React components like the navbar (visual check)
- ...CSS is applied to the elements (visual check)
- ...Bootstrap JS from asset pipeline works (try to open the Filters dropdown on Explore)
- ...Auth0 login flow works—i.e. it is possible to log in and hold a session
- ...Auth0 logout flow works—i.e. it is possible to log out and your session is destroyed
- ...controllers work—i.e. Explore returns escape games if they exist. Create one if it does not exist and make sure that the list of escape games you own on your `user#show` page displays it.
- ...JS views work—i.e. it is possible to remove images from a listing and have the corresponding table row disappear

Security Concerns

I aimed to make as complete a software product as possible, given the time. Part and parcel of this was ensuring security. In Rails, this tends to mean enforcing restraints on which users can edit models, and which fields a user is able to submit. There are two parts to this:

1. ensuring that users not associated with the object cannot edit it
2. ensuring that the object's associated user cannot be changed

In Blacklight, this is done as follows:

1. retrieving the object by way of a query that asks for it among only those the current user owns; returning a 404 status code otherwise
2. setting the object's associated user to the current user

On its own, the second part of my solution would be insecure. The more standard way of doing this on update would be to strip out the incoming user ID from the hidden field in the form. However, there is good reasoning behind it. This methodology is used on create as well as on update, which means that users cannot create records in other users' names.

Chapter 6

Results

The product of my development process is a web application written in Ruby on Rails, with React.js used to generate and drive some areas of the frontend. This is titled Blacklight—its namesake is the kind of UV light which is employed in puzzles in escape rooms all too regularly. Blacklight serves to emulate its namesake, being:

- . . . present in the majority of escape rooms
- . . . used by maintainers as an escape room tool
- . . . used by enthusiasts to discover something new

At the time of writing, Blacklight can be accessed at https://blacklight-dev.herokuapp.com.

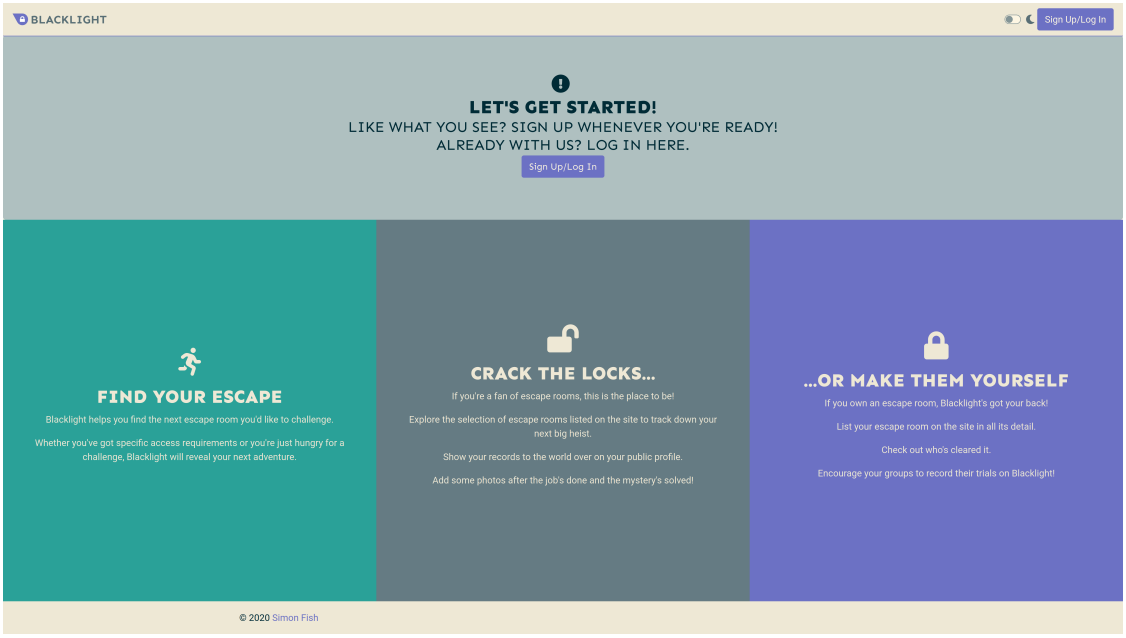

Figure 6.1: Blacklight's homepage as at commit `87a35f25`.

Functionality Overview

Blacklight allows users to log in through the Auth0 authentication service only, though this allows for plenty of expansion in and of itself. Users are prompted to define themselves as an escape game maintainer, enthusiast, or both. This only affects the user's experience, not their level of access.

Escape game maintainers have options revealed to them to create and manage escape games (Figure 6.2). These escape games have an array of associated data that can be added, including the expected time to complete them, their location (by way of Google Maps integration), images,

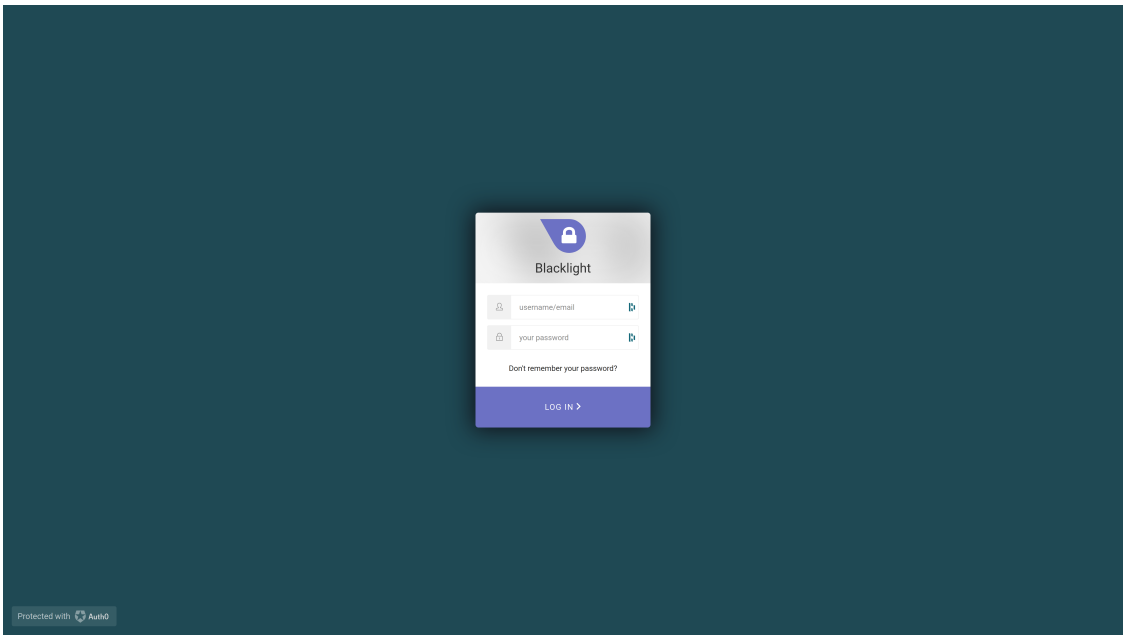

Figure 6.2: Logging in to Blacklight with Auth0. When this screenshot was taken, it was not possible to sign up as I did not want to expose the development instance to unwanted visitors.

and a Markdown-compatible extended description. Listings can also be hidden from view and access via a checkbox on the editing form.

Enthusiasts can browse these escape games and mark them as cleared by selecting the lock icon on any listing (Figure 6.4). This is consistent across the site. Doing so adds the escape game to the 'My Cleared Games' section of the site for that user. There, users can upload photos to associate with each escape room, which also appear to the left of the list in a singular photo gallery.

Users have profiles (Figure 6.5) on which they can write a bio, set their location, and link to their own website. Here, those interested can see all escape games by a maintainer, and all escape games cleared by an enthusiast. Which of these is shown depends on whether the user being viewed has self-assigned as a maintainer, enthusiast, or both.

Practical Use

Were Blacklight to be released, I would encourage its use in the following ways:

- Using the Explore view when searching for new escape games to tackle alone or with friends
- Linking to escape games during discussion and recommendations on other platforms
- After completion, marking an escape game as cleared and uploading photos directly to the site to commemorate the experience
- Showing records of escape games cleared, and the associated photos, to others in person or online

Discussion

To give a summary answer as regards the project's success, the goals of the project have been met. A tool has been developed that allows escape room maintainers to advertise their rooms and upload photos, and allows enthusiasts to do the same in relation to escape rooms that they have cleared. Measures have been taken to ensure that it is secure, functionally consistent, well-designed and feature-complete. Despite this, I cannot shake the notion that with more time available, Blacklight would feel whole, and potentially viable for public rollout. Of course, in the current climate, Blacklight may serve little purpose unless it were repurposed—in any other world, my personal feelings towards it would be justified.

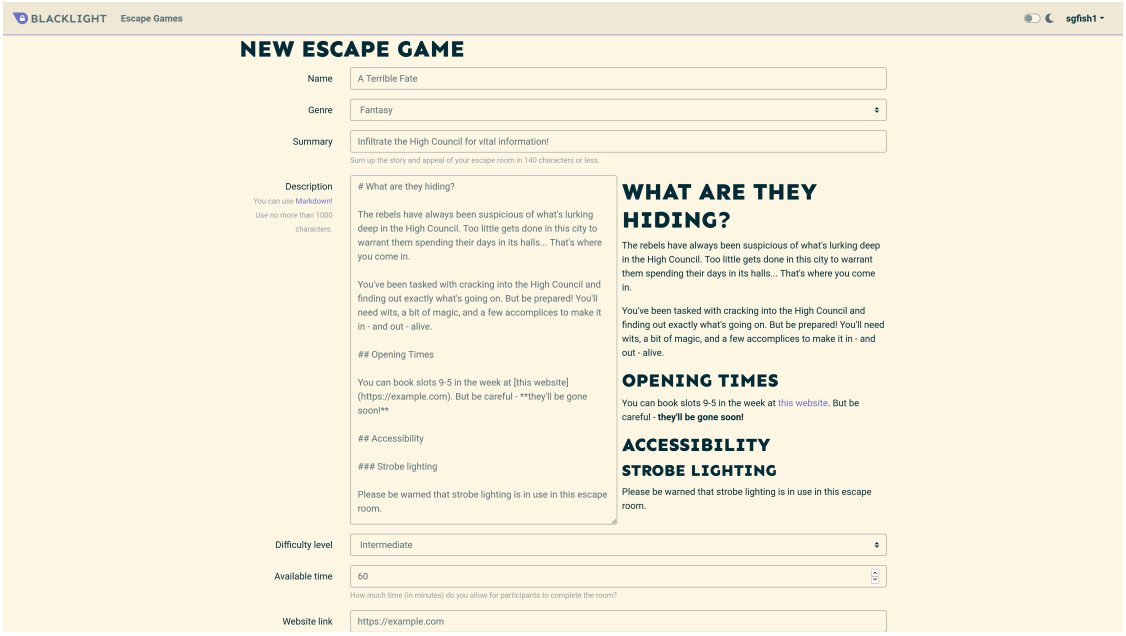

Figure 6.3: Blacklight's escape game creation form as at commit `87a35f25`.

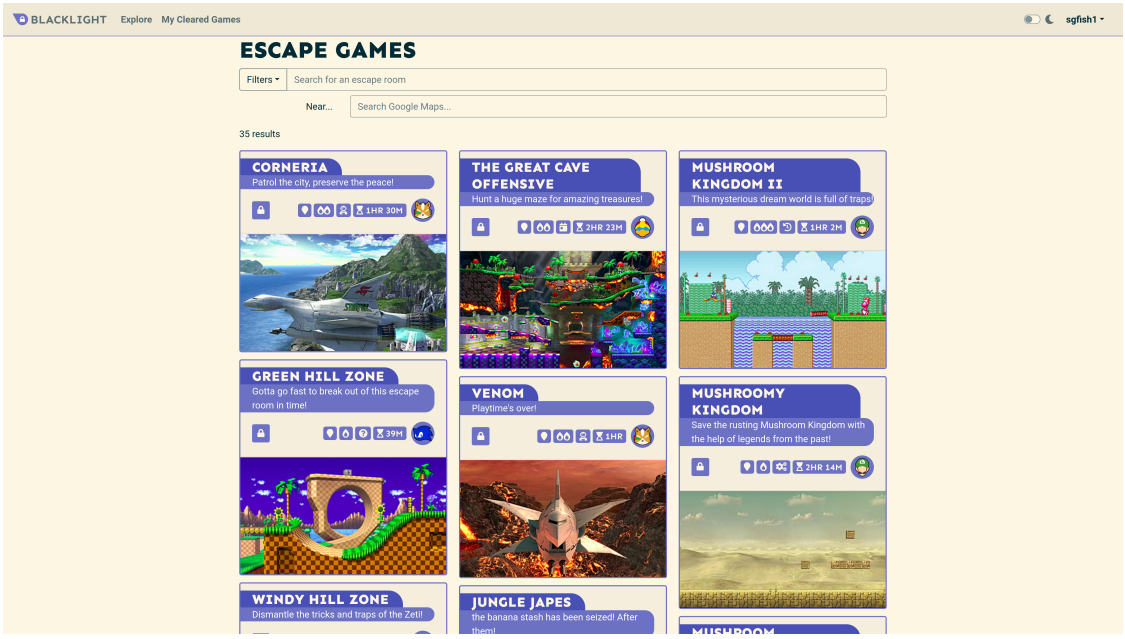

Figure 6.4: Blacklight's Explore view as at commit `87a35f25`. Seed data used in this screenshot uses screenshots from *Super Smash Bros. Ultimate*.

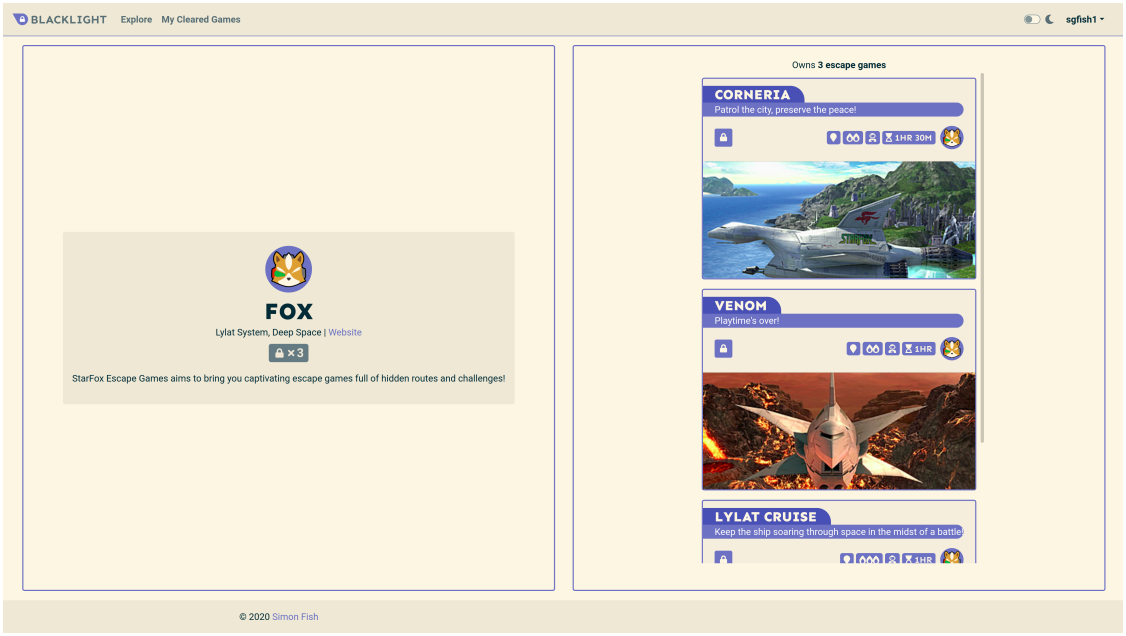

Figure 6.5: Blacklight's profile view as at commit `87a35f25` from the perspective of a user who is a maintainer. Seed data used in this screenshot uses screenshots from *Super Smash Bros. Ultimate*.

My implementation of continuous integration and containerisation surpasses previous attempts I have made in personal projects. I feel as though I have refined skills related to testing, and have particularly made strides in building more interactive frontends with React. Room for improvement still remains—late in development of Blacklight, I encountered limits around my use of CircleCI, and I might have been able to capitalise on the premium feature of Docker layer caching. I was alerted to a potentially more suitable option in GitHub Actions, which would have alleviated both woes—with 3,000 free minutes of running available under my current GitHub plan, I would have been able to run somewhere over 400 builds at the average rate per month, which I expect would have been more than enough. For comparison, development completed with 234 builds in total.

I recognise that my focus towards industry as opposed to academia complicated the process and used much of my available time. Some might also say that my focus on the practical, as opposed to the theoretical, limited the potential for my project to be particularly novel in its approach. One survey respondent even suggested that the intention was to introduce technology to *"trivial features"*. Still, in spite of having had to limit the scope of the project further and forgo many ideas that might have even given Blacklight commercial viability, I am very satisfied with the quality of the product that I was able to make.

Further Work

The existing framework should serve to make the following features straightforward to implement. Security, basic stability, and social features take priority. Blacklight would be in a deployable state with features listed under **Immediate Priorities**—these would provide a stable experience with strong social links between users. Particularly, the increase in test coverage would make Blacklight a more stable platform on which to develop.

Initial Scope

Immediate Priorities

- **Pagination of outputs**
- **>80% test coverage**
- **Improvements to factors relating to authentication**, such as:

  - introducing two-factor authentication
  - permitting the use of social network accounts such as Facebook and Twitter for login.
- **The ability for users to add others as friends**
  Doing so would allow users to track other users' activity on their timeline
- **Leaderboards for each escape game**
- **A timeline for logged-in users**
  This would show the aforementioned, alongside introducing new escape games that had opened in the vicinity of the user
- **The ability to reorder photos on escape games and clear records**

Additional Features

- **Time recording and milestones for escape games**
  Design and functionality would take strong inspiration from LiveSplit[1], and escape game maintainers would have a method of completing these for enthusiasts who attend their rooms
- **Metrics for maintainers**
  Feedback on how many times their escape games and profile had been viewed recently
- **Toggling of privacy on a user's profile per-field**
  e.g. visible to friends only or private

Further Ideas

Many other ideas were discussed, but were not included in the original scope. These included the following:

Immediate Priorities

- **Measures to include "in-home" escape games**
  These features would be immediately vital due to the COVID-19 outbreak. They would aim to ensure that Blacklight would remain a viable product for use by the industry. Blacklight has caveats that make it inappropriate for the industry at present, such as some reliance on escape rooms having a physical location. These features would encompass:
  - an additional filter to select only "in-home" escape-games or standard escape rooms
  - further fields available to the `EscapeGame` model, such as a definitive `price`
    This was omitted as a field in its own right in Blacklight, as 39% of escape rooms charge per team and 55% charge per person, with others pricing either via a *"base cost... plus an additional fee per player"*, or *"banded models"* in which groups of different sizes are charged differently. (Nicholson 2015). Such a difference calls for at least two unique approaches to both collecting the data from, and representing the data to, the user.
- **An extensive tagging system**
- **Increased depth of filtering in the Explore view**
  Included with a view to allowing those with accessibility issues to quickly find rooms suitable for them. This would be particularly effective in tandem with the above suggestion. It is of note that maintainers can outline this kind of information in the description of their escape rooms. However, this cannot be searched or filtered.
- **Refinement of the API to be more standard**
  This would include OpenAPI documentation[2], whose correctness could then be verified against the app using a tool such as Apivore[3]. OpenAPI documentation opens opportunities for a vast array of integrations with Blacklight's API using tools such as OpenAPI Generator[4], saving developers the need to stub or create their own language-specific clients for Blacklight.
- **The ability to bookmark escape games as an enthusiast**
- **A review system**
  This would be specific to Blacklight (i.e. not backed by existing Google Maps data), with some way of weighing more experienced users' feedback against newer users.

[1] https://github.com/LiveSplit/LiveSplit
[2] https://swagger.io/specification/
[3] https://github.com/westfieldlabs/apivore
[4] https://github.com/OpenAPITools/openapi-generator

Additional Features

- **Additional measures to improve accessibility**
  Blacklight's dark mode and use of the Solarized[5] theme, which uses CIELAB ("Colorimetry — Part 4: CIE 1976 L*a*b* colour space" 2019) to maintain readable color contrasts, encourages readability and visibility across the site. However, additional measures could have been taken to improve usability of the site with assistive technologies in line with the WAI-ARIA specification ("Accessible Rich Internet Applications (Wai-Aria) 1.1" 2017), particularly with regards to React-heavy views such as the Explore view.
- **A searchable Google Maps view**
- **Integration with social media**
  More specifically, this would be to allow Blacklight to consume more data from Google Maps and display other social media embeds for escape room pages. Doing so would allow Blacklight to support escape rooms' existing social media presence, and integrate location data into the experience in a more natural way.
- **Maintainer groups**
  This has a use case particularly in the instance of large franchises where multiple people may be responsible for the franchise's online image. The feature would allow multiple users to have ownership of an escape game.

[5] https://github.com/altercation/solarized

Chapter 7

Conclusion

The needs of escape room maintainers, both internally to the escape room experience and externally of the community, were investigated. This concluded in a decision to build a social network, which both allowed maintainers to advertise among other escape rooms and share photos. The product of this was Blacklight, a responsive web application that meets needs expressed by maintainers and suggested by enthusiasts.

Research concluded that the escape room industry is wary in its use of technology. The experience delivered by escape rooms regularly relies on the physical, as opposed to the digital, in order to build excitement and immersion. While technology can be, and has been, employed in engaging and novel ways, maintainers are still cautious in their uses of it. Stability and reliability have been cited as some of the highest-priority reasons for this.

For those reasons, it is difficult to develop technology for use inside the escape room itself. Instead, a solution was considered that focuses more on improving the day-to-day operation of escape rooms as businesses. Surveying the escape room maintainer community resulted in a decision to build a social network between maintainers and enthusiasts, allowing maintainers to advertise to enthusiasts on a mutually beneficial platform.

The design of this social network took several major design considerations into account. Auth0 was chosen as the application's authentication service to reduce the impact of an additional dependency. The implementation of Markdown was a difficult choice due to uncertainty over whether it is intuitive for users. Atomic Design (Frost 2016) helped to guide the structure of the frontend code.

Over roughly 4½ weeks of development time, Blacklight was engineered through thorough application of industry practices such as use of Kanban and automated testing. The result is a web application using Ruby on Rails and React that makes strides towards being secure and providing a strong user experience. It enables escape room maintainers to list their escape rooms, and enables enthusiasts to mark which rooms they have cleared and commemorate their experiences with photos.

This was able to meet the basic requirements I defined at the start of the project. With more time available, additional features could be added to make it a fully compelling product in its own right. At present, the industry would struggle to use it to its full capacity due to its reliance on permanent escape rooms at physical locations. There is further room for growth beyond the minimum viable product in terms of features, but more basic priorities should be addressed before then. These include reaching adequate test coverage, pagination of outputs, enabling two-factor authentication, and addressing missing accessibility features.

Chapter 8

References

N.B.: Dates given for webpages link to the earliest recorded scrape of the webpage by the Wayback Machine[1], or the date of access if this is unavailable.

"Accessible Rich Internet Applications (Wai-Aria) 1.1." 2017. Recommendation. W3C.

Beguin, Erwan, Solal Besnard, Adrien Cros, Barbara Joannes, Ombeline Leclerc-Istria, Alexa Noel, Nicolas Roels, et al. 2019. "Computer-Security-Oriented Escape Room." *IEEE Security & Privacy* 17 (4): 78–83.

"Breakout EDU." 2020. https://www.breakoutedu.com/ (Accessed: 11th May, 2020).

Buzzshot. 2018. "Buzzshot Escape Room Software." https://buzzshot.co/ (Accessed: 8th May, 2020).

Champeon, Steven, and Nick Finck. 2003. "Inclusive Web Design for the Future with Progressive Enhancement." http://hesketh.com/publications/inclusive_web_design_for_the_future/ (Accessed: 17th April, 2020).

Clarke, Samantha, Sylvester Arnab, Helen Keegan, Luca Morini, and Oliver Wood. 2016. "EscapED: Adapting Live-Action, Interactive Games to Support Higher Education Teaching and Learning Practices." In *International Conference on Games and Learning Alliance*, 144–53. Springer.

"Colorimetry — Part 4: CIE 1976 L*a*b* colour space." 2019. Standard. International Organization for Standardization/CIE International Commission on Illumination.

Dietrich, Nicolas. 2018. "Escape Classroom: The Leblanc Process—an Educational 'Escape Game'." *Journal of Chemical Education* 95 (6): 996–99.

Dilek, Sebahattin Emre, and Nur Kulakoglu Dilek. 2018. "Real-Life Escape Rooms as a New Recreational Attraction: The Case of Turkey." *Anatolia* 29 (4): 495–506.

DuPlessie, Matthew. 2013. "Go Analogue." YouTube. July 11, 2013. https://youtu.be/tTcl5I0Wbzk?t=668.

Frost, Brad. 2016. *Atomic Design*.

Gündüz, Şafak. 2018. "Preventing Blue Ocean from Turning into Red Ocean: A Case Study of a Room Escape Game." *Journal of Human Sciences* 15 (1): 1–7.

Herlihy, Peter. 2013. "How Many People Are Missing Out on Javascript Enhancement?" https://gds.blog.gov.uk/2013/10/21/how-many-people-are-missing-out-on-javascript-enhancement/ (Accessed: 17th April, 2020).

Karlesky, Michael, and Mark Voord. 2008. "Agile Project Management," October.

Lockup Escape Rooms, Maintainer of. 2019. Personal communication.

[1] https://archive.org/web/web.php

López, Ángela Gómez. 2019. "The Use of Escape Rooms to Teach and Learn English at University." *Research, Technology and Best Practices in Education*, 94–102.

López-Pernas, Sonsoles, Aldo Gordillo, Enrique Barra, and Juan Quemada. 2019. "Examining the Use of an Educational Escape Room for Teaching Programming in a Higher Education Setting." *IEEE Access* 7: 31723–37.

MacLean, Allan, Richard M Young, Victoria ME Bellotti, and Thomas P Moran. 1991. "Questions, Options, and Criteria: Elements of Design Space Analysis." *Human–Computer Interaction* 6 (3-4): 201–50.

Martin, Robert C, and Micah Martin. 2006. *Agile Principles, Patterns, and Practices in C# (Robert C. Martin)*. Prentice Hall PTR.

McDowell, Charlie, Linda Werner, Heather Bullock, and Julian Fernald. 2002. "The Effects of Pair-Programming on Performance in an Introductory Programming Course." In *Proceedings of the 33rd Sigcse Technical Symposium on Computer Science Education*, 38–42.

Nicholson, Scott. 2015. "Peeking Behind the Locked Door: A Survey of Escape Room Facilities."

———. 2018. "Creating Engaging Escape Rooms for the Classroom." *Childhood Education* 94 (1): 44–49.

Ovadia, Steven. 2014. "Markdown for Librarians and Academics." *Behavioral & Social Sciences Librarian* 33 (2): 120–24. https://doi.org/10.1080/01639269.2014.904696.

Peleg, Ran, Malka Yayon, Dvora Katchevich, Mor Moria-Shipony, and Ron Blonder. 2019. "A Lab-Based Chemical Escape Room: Educational, Mobile, and Fun!" *Journal of Chemical Education* 96 (5): 955–60.

Pendit, Ulka Chandini, Muhaimin Bin Mahzan, Mohamad Danial Fadzly Bin Mohd Basir, Mazlan Bin Mahadzir, and Siti Noraishah binti Musa. 2017. "Virtual Reality Escape Room: The Last Breakout." In *2017 2nd International Conference on Information Technology (Incit)*, 1–4. IEEE.

Radigan, Dan. 2018. "What Are Story Points and How Do You Estimate Them?" https://www.atlassian.com/agile/project-management/estimation (Accessed: 11th May, 2020).

Rehkopf, Max. 2018. "Kanban Vs Scrum." https://www.atlassian.com/agile/kanban/kanban-vs-scrum (Accessed: 17th April, 2020).

Resova Ltd. 2020. "Resova." https://resova.com/built-for/industries/escape-rooms (Accessed: 8th May, 2020).

Ross, R. 2019. "Design of an Open-Source Decoder for Educational Escape Rooms." *IEEE Access* 7: 145777–83.

Ross, R., and C. Bell. 2019. "Turning the Classroom into an Escape Room with Decoder Hardware to Increase Student Engagement." In *2019 IEEE Conference on Games (CoG)*, 1–4.

Rouse, Wendy. 2017. "Lessons Learned While Escaping from a Zombie: Designing a Breakout Edu Game." *The History Teacher* 50 (4).

SCRAP. 2018. "Pacific Rim: Shatterdome Defenders | Real Escape Game Created by Scrap - Voted America's No. 1 Escape Room." https://realescapegame.com/pacificrim/ (Accessed: 8th May, 2020).

Smith, Shana, and Emily Ericson. 2009. "Using Immersive Game-Based Virtual Reality to Teach Fire-Safety Skills to Children." *Virtual Reality* 13 (2): 87–99.

Stasiak, Andrzej. 2016. "Escape Rooms: A New Offer in the Recreation Sector in Poland." *Turyzm* 26 (1): 31–47.

Williams, Laurie, and Richard L Upchurch. 2001. "In Support of Student Pair-Programming." *ACM SIGCSE Bulletin* 33 (1): 327–31.

Xola, Inc. 2018. "Escape Room Booking Software | Xola." https://www.xola.com/booking-software/escape-rooms (Accessed: 8th May, 2020).

Chapter 9

Appendices

Appendix I: Initial Requirements

MSCW	Relates to...	As a...	I want to...	So that...	Estimate
M	sharing photos	maintainer	upload photos to the site	Potential customers have an idea of what the room is like and who else has been	5
M	sharing photos	maintainer	remove photos from the site	In the event of a DMCA/GDPR request, i can answer appropriately	2
M	advertising	maintainer	create a public listing	My room is advertised among others	8
M	advertising	maintainer	remove a public listing	My room is no longer shown, if that's what i wish for	5
M	advertising	maintainer	edit a public listing	My room's details are up to date	5
M	advertising	maintainer	create multiple public listings against the same profile	All of my available escape games can be represented individually	2
S	advertising	maintainer	show links to the rest of my public listings from each one	Customers can browse between my available rooms to find one that's suitable for them	2
S	advertising	maintainer	define the difficulty of my room	Potential customers know what they're up against	2
S	advertising	viewer	see the difficulties of rooms	I know what i'm up against	1
M	advertising	viewer	search for escape rooms	It's easier for me to find a room suitable for me	13
S	advertising	viewer	filter a room search by difficulty	The room i choose to take on has the appropriate level of challenge	3
S	advertising	viewer	filter a room search by location	The room i choose to take on is in the appropriate location for me	5
S	advertising	viewer	filter a room search by distance from me	The room i choose to take on is in the appropriate location for me	5
S	sharing photos	viewer	upload photos to the site against my profile	I have a public record of which escape rooms i have attended	5
M	sharing photos	viewer	upload photos to the site against an escape room	The photos are visible on my public profile and friends/family can view them	5
M	login	user	sign up with email and password	I have an account	8
S	login	user	have my password checked for strength	It is verified secure	5
W	login	user	have a public profile	My records are visible to others	5
W	login	user	toggle privacy on my profile	It is not visible to others if i don't want it to be	2
W	login	user	toggle privacy on my profile per-field	I have control over what others can see about me	13
C	login	user	sign in with other social networks (facebook, twitter, tumblr)	It is easier for me to share to those networks	8
M	social	viewer	record that i have finished a room	It appears on my profile	8

MSCW	Relates to...	As a...	I want to...	So that...	Estimate
W	social	maintainer	configure milestones for my room	Multi-stage rooms can be represented appropriately	13
W	social	user	view a timeline	Others' clears are visible to me	5
S	login	user	reset my password via email	My password is changed	5
C	login	user	use two-factor authentication	The login flow has an additional security layer	13
S	advertising	maintainer	view metrics for my profile		21
M	advertising	maintainer	choose which of my photos is the default shown	In search results, the default is shown first	13
S	advertising	maintainer	change the order in which my photos appear	My profile displays the photos in this order	13
S	advertising	maintainer	use a markup language to write my profile	Titles, bolded sections, etc. can be shown with the correct formatting	5

Appendix II: Escape Room Owner Survey

Eligibility

Please check if you are a member of the Escape Room Enthusiasts group on Facebook: https://www.facebook.com/groups/escaperoomenthusiasts
Checkbox answer

Do you run/own an escape room?
Option select answer

- *Yes*
- *No*

Your Escape Room *Please note that all questions are optional.*

Please state the location of your escape room(s).
Text entry answer

For how many years have you been working in the escape room industry?
Option select answer

- *Less than 1 year*
- *1 year*
- *2 years*
- *3 years*
- *4 years*
- *5 years or more*

Which of the following features do you already offer as part of the escape room experience?
Matrix answer

Rows:

- *Leaderboard*
- *Sharing photos with the community*
- *Membership*
- *In-character communication with the party before booking*
- *Communication with the group during the session (e.g. timer,hints)*
- *Metrics (e.g.: how long does it take for people to complete puzzles?)*
- *Matching solo participants/small parties into groups*
- *Advertisement among other escape rooms*

Columns:

- *Not offered*
- *Interested in offering*
- *Offered*

Please provide any additional details with regards to the previous question here.
Text entry answer

To which of the following features would/do you apply technology in your offering?
Matrix answer

Rows:

- *Leaderboard*
- *Sharing photos with the community*
- *Membership*
- *In-character communication with the party before booking*
- *Communication with the group during the session (e.g. timer, hints)*
- *Metrics (e.g.: how long does it take for people to complete puzzles?)*
- *Matching solo participants/small parties into groups*
- *Advertisement among other escape rooms*

Columns:

- *I would not apply technology to this*

- *I would apply technology to this*
- *I currently apply technology to this*

Please provide any additional details with regards to the previous question here.
Text entry answer

Which factors influence your decision when investing in technical solutions to these problems?
Matrix answer

Rows:

- *Cost*
- *Time taken to implement*
- *Reliability*
- *Ease of use*

Columns:

- *Never*
- *Rarely*
- *Sometimes*
- *Often*
- *Always*
- *I don't know*

Please name any other factors of importance not mentioned.
(In particular, I am interested to know whether they are a priority or significant benefit for your business.)
Text entry answer