Risk Consequence Table Write-Up

Risk	Description
Complicated user interface (UI) that leads to usability issues	Since our target users consist of families in North America, there is no guarantee that all our users will be tech-savvy. Especially since one of our primary user classes consists of busy parents, who are generally older adults, they may not be as familiar with navigating modern interfaces. Thus, there is a risk that if the UI is too complicated for our users, they become frustrated and seek alternatives to our app. Furthermore, usability issues could lead to negative reviews that affect the application's overall image to help busy families.
Performance issues due to slow networks	Depending on the strength of our users' internet connection and the network connections on our system's backend, there is a risk that performance for actions will be slow on our system. This could be frustrating for users, since if performance is slow, then they may spend more time waiting to accomplish actions on the application, such as booking out a resource they need immediately. Additionally, as our users are busy families, slower performance may lead to unsynced schedules as our system communicates and receives updates from adjacent systems, such as calendar services.
Security issues that lead to data breaches	Our system has information that pertains to families' day-to-day lives and account information for digital subscriptions that families have. Thus, it is imperative that we maintain security for this data. However, with the possibility of ransomware attacks and common software security threats, this cannot be 100% guaranteed. If we experience a data breach, then this would harm the integrity of our system and we could lose many existing or future users who could benefit from the system.
Adjacent systems that are experiencing downtime	As our system has a heavy reliance on adjacent systems, such as calendar services and social media services, the actions performed on our systems depend on communications with these adjacent systems. If these adjacent systems experience downtime, due to scheduled maintenance or an unexpected issue, then this may impact our system's ability to help the user accomplish the various workflows that they need to accomplish on our system.

Risk	Likelihood Reasoning
Complicated user interface (UI) that leads to usability issues	We gave this a likelihood of 0.5 because usability is a key component of our system. Since we are targeting users that have a wide range of capabilities with technology, it is hard to guarantee that the design of our system is intuitive enough for our older adult users (in contrast, this may not be as big of an issue with our young adult users). As usability in the UI is also within our control, we have a greater responsibility to deliver a system that is user friendly. Additionally, even with user testing, it is hard to predict whether the UI works for our user class until we collect sufficient user feedback and iterate on the designs.
Performance issues due to slow networks	We gave this a likelihood of 0.4 because our users' network speeds are out of our control. Additionally, the network speeds on our backends are also slightly out of our control, even with the assumption that we use the best network on the market. Thus, while this is likely to occur, we put it at a 0.4 because usually it is a temporary disruption to the workflow and should be fixed relatively quickly.
Security issues that lead to data breaches	We gave this a likelihood of 0.6 because software security is a big issue, and breaches may happen even if they were not done so by malicious intent from our internal system. However, we are currently aware of this issue, and thus are keeping security in mind as we build out the system. While this can occur, we believe that with the proper measures in place, we can mitigate part of the threat from an early stage.
Adjacent systems that are experiencing downtime	We gave this a likelihood of 0.5 because our system relies quite heavily on communication with adjacent systems for the user to perform their workflows. This is also outside of our control, so even if an adjacent system is down for scheduled maintenance, we need to respond to the notice on our end, but cannot prevent it from happening. However, as we are relying on reputable systems (e.g. social media and calendar services) that do not usually experience downtime, we do not see this threat as occurring frequently.

Requirement	Highest Impact Risk	Highest Impact Risk Reasoning
Plan a family activity	Adjacent systems that are experiencing downtime	We estimate that the "adjacent systems being down as the highest impact risk since planning a family activity relies on being able to find a time in the schedule that suits all family members. Thus, receiving the information regarding all family members' availability for a given time range is important to facilitate the planning of this activity. This requires that the calendar services are functioning properly.
Coordinate the usage of shared resources	Performance issues due to slow networks	We estimate that the “performance issues due to slow networks” as the highest impact risk since, when a family member wants to book a resource, it is likely they need the resource right away. Thus, if the network is slow, they may be unable to complete their booking for their immediate use. Also, the slow network would be frustrating for any busy family member.
Connect with family members to keep up-to-date	Adjacent systems that are experiencing downtime	We estimate that the adjacent systems being down as the highest impact risk since there is a heavy reliance on receiving information about posts on social media to help facilitate connections between family members. Thus, if the social media services are down, then this hinders the ability for family members to get updates about each others' lives and they may not easily complete the workflow.
Monitor shared digital subscriptions	Security issues that lead to data breaches	We estimate that the security issues that may lead to data breaches as the highest impact risk since this use case allows families to keep track of sensitive credential information related to subscription accounts. Thus, if the integrity of the data becomes compromised, this puts families' credential information at risk.
Collaborate on shared media playlists	Adjacent systems that are experiencing downtime	We estimate that the adjacent systems being down as the highest impact risk since this use case relies heavily on media services to enable families to collaborate on creating shared media playlists. Thus, if the media services are down, it would be difficult for family members to create these playlists at all.

Risk Countermeasures Table Write-Up

Countermeasure	Description
C1. Testing (A/B testing, integration testing, stress testing etc.)	Before the system proceeds to the production stage, it needs to be thoroughly tested by several testing strategies such as A/B testing, integration testing, and stress testing. Through user feedback collected from A/B testing, we hope to find a UI that performs better, and thereby improve the user experience of the system. Additionally, traffic control problems that cause low robustness and connection issues to the adjacent systems are expected to be discovered and eliminated at an early stage. Security vulnerabilities can be found by security testing, and appropriate mitigation should be performed after testing.
C2. Lazy loading and caching frequently accessed family member information	The frontend of the system determines what is necessary for the initial experience and lazy-load content only if user interaction happens, which reduces load time and conserves bandwidth. The frequently accessed data of the family members can also be cached, so that the data from the previous visit are available to the user -- even if the network connection is unstable or the adjacent system is down.
C3. Option of following manual steps to complete the tasks	For the tasks that rely on adjacent systems, our system provides the option of following manual steps for an alternative workflow, which re-uses other existing features, such as the chat. For example, if the calendar services are down or the network is not good enough to support Application Programming Interface (API) calls for auto-generating a common availability of the family, users can choose to manually select a time slot for a family activity, and confirm the time through the chat. We also hope this helps users who are not very tech-savvy to have alternative ways to access our services as well.
C4. Data encryption	During the early stages of development, the developers decide and utilize an appropriate encryption strategy for data storage of the system. The encryption translates data into ciphertext, which makes the data less enticing to cyber-criminals because the encrypted data cannot be read without the private key. It protects stored sensitive information and enhances the security of network communications from hijacking and exploitation by unauthorized parties.

Countermeasure	Highest Mitigated Risk	Impact Estimate Reasoning
C1. Testing (A/B testing, integration testing, stress testing etc.)	Complicated UI that leads to usability issues (0.7)	A/B testing of countermeasure 1 provides a technique to show users multiple variants of a design to discover which one has a better performance. Commitment to time-costly UI changes can be avoided if the test results prove that the design is ineffective. The highest mitigated risk is creating a complicated UI that is hard for the users to understand. Since this countermeasure is very effective in terms of this risk, we have given a high impact estimation of 0.7 to it. By adapting this countermeasure, we are able to create a system that has high user engagement.
C2. Lazy loading and caching frequently accessed family member information	Performance issues due to slow networks (0.6)	Countermeasure 2 focuses on fast loading time and low usage of the bandwidth of the system. We have assigned “performance issues due to slow networks” to be its highest mitigated risk because we do not have control over the network connection of the user, and the common way to improve the system’s performance under such conditions is to optimize its time efficiency. Since we cache the frequently accessed family member information for the user, and perform lazy-loading, users need only to download some of the data in the system for the first time visit, and the contents are readily available regardless of the strength of the network connection.
C3. Option of following manual steps to complete the tasks	Adjacent systems that are experiencing downtime (0.8)	We have estimated “adjacent systems that are experiencing downtime” as the highest mitigated risk of countermeasure 3 because the countermeasure allows users to have an alternative way to access the services provided by the system in a manual way without the help of the third-party APIs. It received the highest impact estimation score because the manual options can decouple the dependencies on the adjacent systems when they are unavailable, and maintain the usability of the services provided to a certain extent.
C4. Data encryption	Security issues that lead to data breaches (0.7)	Data encryption has no impact on UI design, network connection, or adjacent system availability. The sole mitigated risk of countermeasure 4 is the security issues that lead to data breaches, so this risk naturally falls under the highest mitigated risk. Countermeasure 4 focuses on preventing cyber attacks by making the data unreadable. The value of the data is lost if the confidential data cannot be decrypted, which leads to less attempts made to exploit the security vulnerabilities.

Mitigated Risk Write-up

Optimal Countermeasures

C1. Testing (A/B, integration, stress testing, etc.) (Overall single effect = 1.5925)
C3. Option of following manual steps to complete tasks (Overall single effect = 1.3959)

We chose C1 and C3 because they strongly address our two highest risk criticalities: complicated UI that leads to usability issues (risk criticality of 1.085) and adjacent systems that are experiencing downtime (risk criticality of 1.16).
First, C3 is the most effective countermeasure that mitigates our highest risk criticality of 1.16 for adjacent systems that are experiencing downtime. However, there is one risk that C3 does not address and it only slightly mitigates the remaining two risks. To supplement this decision, we also chose C1 as it moderately addresses all remaining risks, which in combination with C3, most effectively balances and reduces the final risk criticalities as shown below.

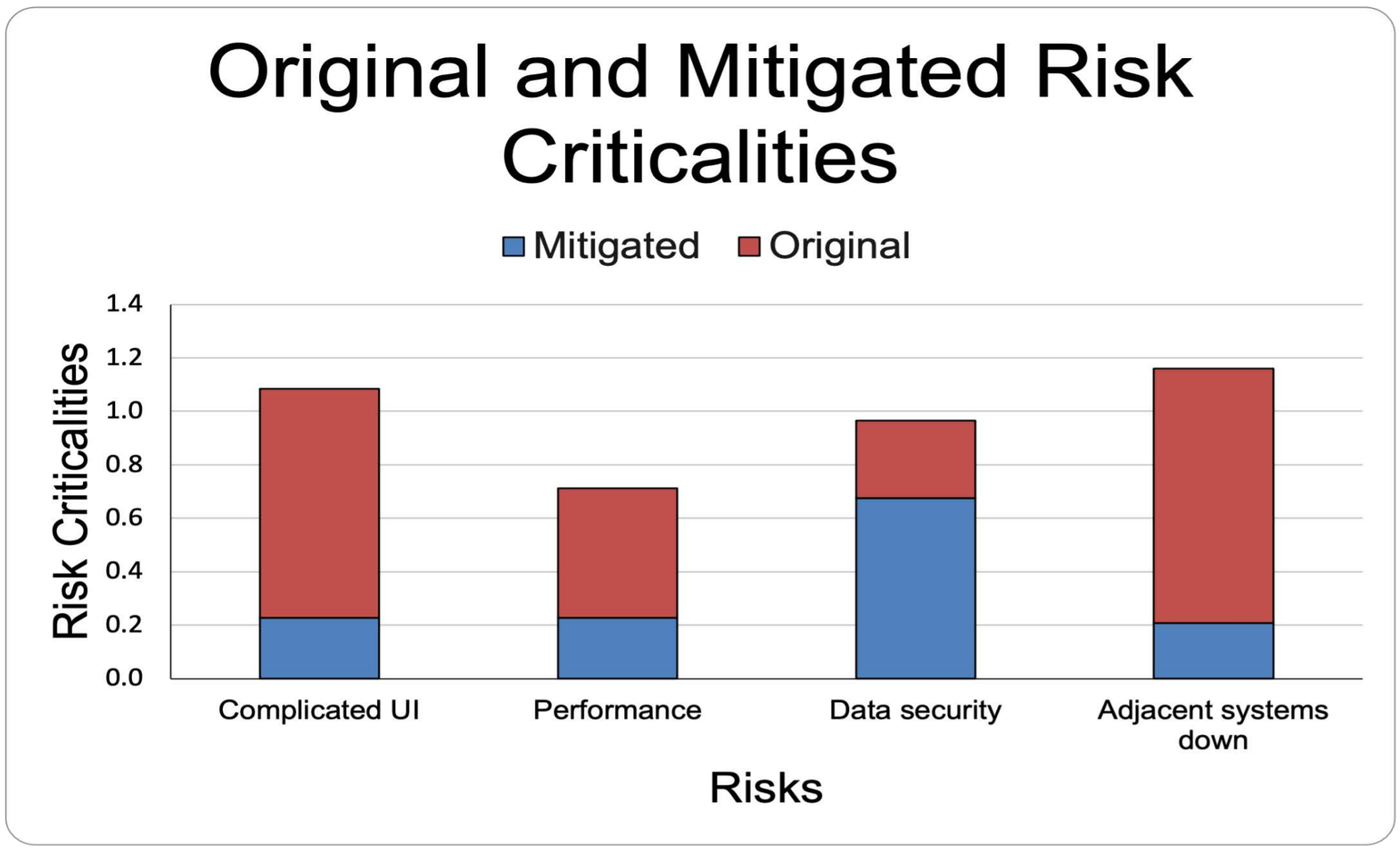
Mitigated Risk Criticality Calculations

Risk	Mitigated Risk Criticality
Complicated user interface (UI) that leads to usability issues	(1.085) * (1 - 0.7) * (1 - 0.3) = 0.2276
Performance issues due to slow networks	(0.712) * (1 - 0.6) * (1 - 0.2) = 0.2278
Security issues that lead to data breaches	(0.966) * (1 - 0.3) * (1 - 0) = 0.6762
Adjacent systems that are experiencing downtime	(1.16) * (1 - 0.1) * (1 - 0.8) = 0.2088

For each risk, we calculated the mitigated risk criticality by finding the remaining proportion of risk that is not mitigated by each optimal countermeasure and multiplying that value with the original risk criticality.
For example, for the first risk in the table (complicated user interface (UI) that leads to usability issues), we did the following:

1. The risk has an original risk criticality value of 1.085.
2. The countermeasure C1 reduces that risk by 0.7 and C3 reduces it by 0.3.
3. To calculate the mitigated risk, we found the proportion of unmitigated risk by calculating (1 - 0.3) and (1 - 0.7), and multiplied these values with 1.085 to find the proportion of total remaining unmitigated risk.

Mitigated Risk Graphs



In this bar chart, we can compare the original risk criticality values for each risk in red and the mitigated risk criticality values in blue. We can see that all risk criticalities significantly decrease after applying the two optimal countermeasures and are overall balanced. The exception is data security which still has a higher remaining risk criticality compared to the other risks. However, its mitigated risk criticality value is significantly less than the original, and it was originally not one of the highest risks to focus on, so we conclude that its remaining risk is adequately lowered.