

# Risk Management Plan

SSE 657

Erin Cargin     Dylan DeVries  
Jarod Miller     Joel Seepersaud

## 1. Risk Identification

Throughout the software development phase, key risks have been identified and documented. These risks have been assessed in terms of likelihood and program impact. Rather than categorize these risks with monetary numbers, we have decided to use the amount of hours that the issues would take to resolve. Using an hour amount gives us a better picture as to the amount of schedule impact that these risks would cause and allows for better planning to mitigate them. Our risks have been identified in the following table:

	Risk Item	Summary
1	Inaccurate Time Estimates	During the planning phase of each sprint, our team estimates how long each task will take. As most of the team members are new to this software, these estimates may not be accurate, which could cause schedule delays.
2	Database Implementation	Creating and integrating with an unknown database will be difficult. Our team does not have any prior database background, so it is possible that this ambitious task may not be achieved within our timeframe.
3	Database Tier Overrun	We are currently operating in the free database tier. Should we get near our tier limits, we will need to migrate to another database solution.
4	Experimental GUI Software	Our project utilizes the python library Dear PyGUI to display our user interface. This library is still being actively developed, so some features are still under construction. It also does not have the best documentation and resources to learn. It is possible that some planned user interface features are not possible with this library.

## 2. Risk Analysis and Mitigation

### 2.1 Inaccurate Time Estimates

	Option	Outcome	Loss Probability	Loss Magnitude (hours)	Risk Exposure (hours)	Combined Risk Exposure
1	Plan sprint to best of ability	Accurate time estimates	0.50	0	0	15
		Inaccurate time estimates	0.50	30	15	

Due to our team not having extensive experience with our software tools, time estimating tasks was difficult. To mitigate the risk of inaccurate planning, the team planned features that could be cut out if schedule became an issue.

### 2.2 Database Implementation

	Option	Outcome	Loss Probability	Loss Magnitude (hours)	Risk Exposure (hours)	Combined Risk Exposure
1	Create DB on Server	Create DB in timeframe	0.60	0	0	8
		Significant development delay	0.40	20	8	
2	Use Local DB	Create DB in timeframe	1.00	0	0	0

Although creating the local database would be less risky, this has a significant application viability impact. We have decided to create the DB on the server, with some mitigation methods. To mitigate this risk, we have planned more time to create the database and have chosen a DB with extensive documentation and tutorials, so the implementation is easier.

## 2.3 Database Tier Overrun

	Option	Outcome	Loss Probability	Loss Magnitude (hours)	Risk Exposure (hours)	Combined Risk Exposure
1	Utilize Free tier in database pricing	No Tier Overrun	0.90	0	0	3
		Tier Overrun	0.10	30	3	

When adjusting for loss probability, utilizing the free DB tier makes this risk minimal. However, to mitigate the risk of having to migrate to another DB solution, our team will be regularly checking our utilization. If the database queries accumulate to within 80% of our maximum allowed queries, we will eliminate all DB calls from the code in our testing and only call them when the functionality is needed.

## 2.4 Experimental GUI Software

	Option	Outcome	Loss Probability	Loss Magnitude (hours)	Risk Exposure (hours)	Combined Risk Exposure
1	Dear PyGUI Library	All required user interface capability available	0.80	0	0	3
		Missing capability	0.20	15	3	
2	TKinter Library	All required user interface capability available	0.95	0	0	0.75
		Missing capability	0.05	15	0.75	

When adjusting for loss probability, the two libraries have a similar risk exposure. However, Dear PyGui offers a clean user interface and unique look, which TKinter does not. Although it is more risky to use this library than an established one like TKinter, the benefits outweigh the risks. To mitigate this risk, a team member was tasked with conducting a compatibility analysis on the library to give us insight into the issue early. Our team also budgeted additional time for implementing the user interface.