

ITMS-448/548

Cyber Security Technologies

# “Media Flayer”

## Risk Management Plan

Nov, 2024

### Overview

This is the Risk Management Plan for the Media Flayer OSINT application project. Various potential risks and impacts will be discussed alongside strategies used to mitigate the risks and ensure the best possible usability.

### Goals

1. **Reliable API Integration:** Ensuring the use of well documented apis alongside a strong community, as well as implementing a fallback option in order to handle any data source disruption.
2. **Prioritize Data Security and Privacy:** We must ensure all data handling is in compliance with the latest privacy laws as well ensuring we are practicing secure coding and encryption in order to protect users.
3. **Create a User-friendly GUI Interface:** In order to ensure the best user experience there must be allocated time for design and development alongside user feedback to refine any discrepancies.
4. **Stability throughout the Application:** Throughout deployment all bugs must be reported to ensure future smooth collaborations across team members, version control will collectively be smoother to resolve any conflicts.

## Risks/Impacts

Risk	Likelihood	Impact	Description
API Integration Delays	Medium	High	Api may change, lack support, or may be incorrectly documented which may result in hindered development. Functionalities from apis are crucial which would make any delays require immediate reworking.
Scope Creep	Medium	High	Scope Creep requires stretching resources within the team to implement additional features or functions and ultimately complete the deliverable.
Data Security/Privacy	Low	High	Attacks such as a SQL injection leave applications vulnerable to users input data. This can result from insecure apis and lack of encryptions.
GUI Development Technical Challenges	Medium	Medium	Constructing an application to ensure a usable GUI may pose challenges across compatibility and accessibility. GUI frameworks, User feedback, and optimized designs lead towards the best user experience.
Insufficient Testing	Medium	High	Skipping integration or system level tests may result in undetected bugs which may lead to crashes, broken functions and vulnerabilities upon launch. Proper constant testing may pose a challenge with insufficient automation.
Version Control	Medium	Medium	Merging integrations upon Github may cause a clash upon development, without proper citing and organization teams may lose changes or overwrite something causing future issues.

## Strategies

1. API Delay Mitigation
  - Selecting Reliable APIs
  - Early Testing of APIs for apparent delays or disruptions
  - Using Mock tools to simulate API results throughout development
2. Security and Privacy Concerns
  - Secure Coding practices such as sanitization or input validation is crucial for regulations
  - The use of Encryption such as SSL/TLS or AES is essential to protect data during in transit and at rest
  - Role based access control may be used to limit access into sensitive data
  - The use of third party services or tools may also be beneficial to projects with handling data.
3. GUI Development
  - Conducting user testing will ensure all accessibility and usability requirements are met and other expectations may be reworked inside of the framework
  - Cross platform testing is another form of ensuring functionality amongst different devices
  - Ensuring the framework is used effectively during development is key to ensure all users are able to access, easily use and browse securely.
4. Testing/Rework Control
  - Frequent commits with descriptions will allow for team members to understand changes at much smaller sizes every push/pull.
  - Frequent Code reviews will allow to ensure consistency, lower the risk of errors and effectively control any conflicts
5. Risk Management Best Practices
  - Communication is a crucial form in which expectations and risks are addressed to ensure future success
  - Regular assessments along the project lifecycle is key to identify and rework any risk
  - Ultimately ensuring all team members are resolving conflicts while merging efficiently is crucial in development