AI Agent for Documentation and Summation

Risk Assessment

Version 3.0

Team Members:

Abhisar Anand

Ali Mutlek

Brayan Cano

Shar Min

Myles Edwards

Dhyan Patel

Revision History

Date	Version	Description	Name(s)
10/10/25	1.0	First iteration of the risk assessment document	Myles Edwards
11/08/25	2.0	Nothing major needs to be changed but added a section, “8. Limited GPU”, at the bottom	Shar Min
12/10/25	3.0	Minor edits and revisions, minor edits to risk #8	Myles Edwards

Risk Assessment

1. Integration rejection due to underwhelming prototype

Due to the open-ended and research-oriented nature of this project, there is a risk that the developed prototype may not fully align with stakeholder expectations. The project integrates multiple advanced technologies, including speech-to-text, NLP, and secure data storage, which can create ambiguity about the sponsor's desired level of functionality or accuracy.

- **Project Impact:**
  A misaligned or underwhelming prototype could result in rework, wasted effort, or loss of sponsor confidence. It could also require redesigning core components late in the project cycle.

- **Mitigation Strategy:**
  Maintain regular communication with the sponsor and provide iterative demonstrations. Collect structured feedback and document all design and feature decisions to ensure alignment with expectations.

- **Contingency Plan:**
  If the prototype fails to meet expectations, conduct a focused review session with the sponsor to redefine priorities and scope. Adjust deliverables to emphasize core functional features over optional enhancements.

2. Large technology changes

The project relies on APIs and frameworks for transcription, NLP, and secure cloud integration. If these technologies are deprecated, updated, or restricted, the system could lose functionality.

- **Project Impact:**
  Loss or alteration of external dependencies could break existing features and delay project completion due to required refactoring or migration.

- **Mitigation Strategy:**
  Continuously monitor official documentation and release notes for all third-party dependencies. Favor stable, open-source, or locally deployable tools when possible.

- **Contingency Plan:**

Maintain backup alternatives and an abstraction layer between core modules and external APIs. In the event of major changes, migrate to backup technologies while preserving data compatibility.

3. Poor team cohesion

Group projects rely on the effort of each group member doing their best to achieve success for the team. Unfortunately, sometimes team members do not work well with everyone in the team.

- **Project Impact:** Sometimes group members can clash heads on how best to approach a problem/area in a project. This can result in slowed or stalled development times as the group needs the issue resolved before continuing with the project. In extreme cases this can result in the group being unable to complete their product.
- **Mitigation Strategy:** In order to mitigate this issue, frequent meetings are required as well as clear, non-confrontational communication between group members. By meeting often, each member can be on the same page.
- **Contingency Plan:** Should this mitigation strategy fail, then we will seek outside intervention. By reaching out to our project sponsor or faculty mentor, we can resolve the issue impartially and continue development progress.

4. Unplanned sickness/family emergences

Unplanned sickness and family emergencies impede development progress.

- **Project Impact:** When group members have unplanned absences, development suffers as they are unable to achieve the work they were assigned. This can slow down development and in extreme cases, cause the development team to need to do the same amount of work with less people.
- **Mitigation Strategy:** In order to mitigate this risk, we need to have clear communication often. By communicating well, we reduce the impact the absence has on the team as we can plan around it and divide the workload between the remaining members.
- **Contingency Plan:** If this mitigation strategy fails, we can look to communicate with the project sponsor regarding our expectations for the final product. Additionally, we can plan

how we will split their work between the remaining members to achieve the same final product.

5. Scope Creep

The promising or requesting of new features throughout the development lifecycle, resulting in an unattainable end goal.

- **Project Impact:** Scope creep can result in an undeliverable project. If we constantly promise more and more from the optional requirements as we develop items from the requirements, we can run into an issue where we are unable to deliver what we had promised.
- **Mitigation Strategy:** In order to mitigate this risk, we need to ensure that if we ever promise an additional optional requirement, we ensure that we are in a position where we can actually deliver the promise. Additionally, we should never promise another optional requirement until we have completed all the other requirements.
- **Contingency Plan:** Remove unfinished optional features to deliver a complete project and meet project requirements.

6. Data privacy and HIPAA/Security compliance issues

The system processes sensitive clinical data containing Protected Health Information (PHI). Improper storage, handling, or transmission could result in a HIPAA violation.

- **Project Impact:**
  Any data breach or non-compliance incident could disqualify the system from real-world use and expose the team or sponsor to legal liabilities.

- **Mitigation Strategy:**
  Implement end-to-end encryption, secure authentication, and automated PII redaction. Follow the principle of least privilege for all user roles and maintain logs of data access.

- **Contingency Plan:**
  Immediately suspend operations, isolate affected systems, and perform a security audit. Reassess architecture and correct vulnerabilities before reactivation.

7. Model inaccuracy or bias

AI models used for transcription, redaction, and summarization may exhibit inaccuracies or bias —especially with accents, dialects, or specialized medical language.

- **Project Impact:**
  Inaccurate or biased outputs could reduce clinician trust, cause miscommunication, or generate faulty medical notes.

- **Mitigation Strategy:**
  Train and test models on diverse datasets. Conduct human-in-the-loop reviews during testing to evaluate model performance across linguistic and demographic variations.

- **Contingency Plan:**
  Implement post-processing corrections and allow users to edit or verify AI-generated text. Retrain or fine-tune models as new data becomes available.

8. Limited GPU

The AI models (Whisper, Pyannote, OCR) depend on GPU processing, but GPU servers are shared with other university projects.

- **Project Impact:**
  Reduced GPU availability can lead to significant delays in model training, transcription workloads, and large-scale testing. This may push back development milestones and reduce productivity, especially during peak phases of the project.

- **Mitigation Strategy:**
  Schedule GPU usage in advance and optimize workloads by running small-scale or preliminary tests on CPU when possible. Reduce unnecessary GPU use by batching tasks and monitoring usage efficiency.

- **Contingency Plan:**
  If GPUs become unavailable, downscale input files, perform CPU-based testing where feasible, or postpone non-critical testing tasks until resources become available.