Hazard Analysis
Software Engineering

Team #1, Sanskrit Ciphers
Omar El Aref
Dylan Garner
Muhammad Umar Khan
Aswin Kuganesan
Yousef Shahin

Table 1: Revision History

Date	**Developer(s)**	**Change**
October 10th 2025	Dylan Garner	Added initial hazard analysis
October 10th 2025	Omar El Aref	Added System Boundaries and Components
October 10th 2025	Umar Khan	Added safety and security requirements
...	...	...

Contents

[You are free to modify this template. —SS]

1 Introduction

1.1 Definition of Hazard

Following the definition from hazard analysis literature, a **hazard** is defined as a property or condition in the Sanskrit Manuscript Fragment Reconstruction Platform system together with a condition in the environment that has the potential to cause harm or damage (loss).

1.1.1 System Conditions

System conditions are properties or states of the software that create vulnerability or risk. Examples include:

- **Vulnerable database configuration**: Database exposed to unauthorized access attempts
- **Inaccurate Artificial Intelligence (AI) models**: Script classification model with low accuracy (¡50%)
- **Insufficient input validation**: System accepts malformed or malicious data
- **Poor error handling**: System crashes or enters unstable state on unexpected input
- **Inadequate session management**: Sessions persist beyond intended duration or lack proper isolation
- **Resource allocation issues**: Memory leaks, unbounded queries, or inefficient algorithms
- **Missing accessibility features**: Interface elements without keyboard navigation or screen reader support

1.1.2 Environmental Conditions

Environmental conditions are external circumstances or user behaviors that, when combined with system conditions, can cause harm. Examples include:

- **User attempts unauthorized data access**: User tries to download entire database or access restricted content
- **User trusts inaccurate AI predictions**: Scholar bases research conclusions on low-confidence model outputs
- **Network interruptions during critical operations**: Connection lost while saving research data

- **Concurrent user access to shared resources**: Multiple researchers editing same fragment simultaneously
- **Users with varying technical expertise**: Non-technical users attempting complex system operations
- **Users with accessibility needs**: Users requiring assistive technologies to interact with system

1.2 Project Context and Unique Hazard Considerations

The Sanskrit Manuscript Fragment Reconstruction Platform operates within a distinctive environment that creates unique user experience hazards not typically addressed in conventional software projects:

1.2.1 Diverse User Base with Varying Technical Skills

The system serves users ranging from graduate students learning paleography to expert scholars to archival staff, each with different technical comfort levels and research needs. Poor interface design or insensitive feature implementation could exclude or frustrate significant user groups, leading to abandonment of the tool or ineffective research workflows.

1.2.2 Research Workflow Integration Challenges

Scholars have established research methodologies and workspace preferences developed over years of practice. Software that disrupts familiar workflows, lacks intuitive navigation, or forces users to adapt to rigid system constraints can cause significant frustration and reduce research productivity. Poor performance or unreliable functionality breaks concentration and research flow.

1.2.3 Data Privacy and Confidentiality Concerns

Researchers work with sensitive data including unpublished discoveries, institutional collaborations, and potentially restricted manuscript materials. Unintended data exposure through system vulnerabilities, poor session management, or inadequate access controls could compromise ongoing research, violate institutional agreements, or expose confidential scholarly work to competitors.

2 Scope and Purpose of Hazard Analysis

2.1 Purpose

This hazard analysis identifies and evaluates potential risks associated with the Sanskrit Manuscript Fragment Reconstruction Platform to ensure safe, responsible deployment in academic research environments. The analysis focuses on protecting cultural heritage materials, maintaining scholarly integrity, and preventing negative impacts on Buddhist Studies research. Through systematic verification testing and validation processes, this analysis helps identify potential system failures and ensures that main functionalities operate safely and reliably before deployment.

2.2 Potential Losses

Table 2 summarizes the potential losses that could occur from identified hazards in the system. These losses span data security, user experience, research productivity, and system reliability concerns.

3 System Boundaries and Components

This hazard analysis treats the system as four major component groups that interact to support scholarly reconstruction of manuscript fragments. (For full details, refer to the SRS, Section S.1.)

Major Component Groups

- **Front-end**: Scholar-facing web UI for secure login, batch image uploads, an interactive canvas (arrange/rotate/zoom fragments), match discovery display, and session/annotation saving. Quick list of smaller components:
  - User Authentication Interface
  - Fragment Upload Interface
  - Interactive Canvas Module
  - Match Discovery Module
  - Progress Management Module
- **Backend**: API gateway, authentication/authorization, image preprocessing pipeline (normalization/orientation/format), database access layer, and a match-orchestration service coordinating analysis results. Concise breakdown of sub-components:
  - API Gateway
  - Authentication Service
  - Image Processing Service
  - Database Access Layer
  - Match Orchestration Service
- **Data Storage**: Fragment image store (originals and derivatives with metadata), user accounts/permissions, and project records (arrangements, match history, confidence scores, session snapshots). Brief overview of minor elements:

  - Fragment Image Database
  - User Database
  - Project Database

- **AI/ML**: Services/models for edge/damage matching, handwriting/script classification, OCR text extraction (with confidence), and content similarity/embedding comparisons. Summary of key smaller parts:
  - Edge Pattern Matching Model
  - Handwriting Style Classifier
  - Damage Pattern Recognition Model
  - Text Extraction Model
  - Content Similarity Model

Component Boundaries (Exclusions)

- **UI** presents results and collects inputs; it does not run ML models or make authoritative scholarly decisions.
- **Preprocessing** only standardizes images; it does not judge correctness of reconstructions.
- **Matching** produces scored suggestions; it does not auto-merge/link records without explicit user confirmation.
- **OCR/Transcription** is machine-generated with confidence indicators; final text requires human review before being marked confirmed.
- **Datastores** are system-of-record for this application only; they do not write back to external catalogues/corpora.

See SRS Section S.1 for the detailed component descriptions and interfaces. Note that here I have separated the backend and data storage to make it a little clearer to visualize the components but data storage will be treated as backend for the project.

4 Critical Assumptions

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS]

5 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

6 Safety and Security Requirements

The following safety and security requirements are derived from the identified hazards and their recommended mitigations. Each requirement corresponds to one or more failure modes discussed in the previous section. The intent is to ensure that failures in individual services or user actions do not compromise data integrity, security, or scholarly validity.

- **File Integrity and Upload Validation** *(Priority: High)*
  - The system shall validate all uploaded files for acceptable format (JPG, PNG) and integrity before ingestion.
  - The system shall reject corrupted or incomplete files and provide descriptive feedback to the user.
  - The system shall maintain a checksum for each uploaded image to prevent future corruption or mismatch.

  Rationale: Prevents data corruption and ensures scholarly work is based on intact manuscript images. Critical for maintaining research validity and preventing irreversible data loss.

- **User Interaction Safety** *(Priority: High)*
  - The system shall include undo/redo functionality for all manual alignment or manipulation tools on the interactive canvas.
  - The system shall display alignment guides or grid overlays to minimize misplacement of fragments.
  - The system shall maintain a revision history of all user adjustments to allow recovery from accidental operations.

  Rationale: Protects scholars from losing hours of painstaking manual work due to accidental clicks or misoperations. Essential for user confidence and adoption of the system.

- **Authentication and Access Control** *(Priority: High)*

  - The system shall enforce strong password policies, require secure session tokens, and store credentials in an encrypted manner.
  - The system shall support optional two-factor authentication (2FA) for privileged or administrative accounts.
  - The system shall log all authentication events (login, failed attempt, password reset) and perform periodic security audits.
  - The system shall automatically lock an account after multiple failed authentication attempts within a short time window.

  Rationale: Essential for protecting scholarly research data and preventing unauthorized access to valuable manuscript collections. Required for compliance with institutional security policies.

- **Database Integrity and Recovery** *(Priority: High)*

  - The system shall perform automated daily backups of all databases (fragment metadata, annotations, user projects).
  - The system shall implement transaction rollback and atomic commits to prevent partial data writes.
  - The system shall provide administrators with a verified recovery mechanism for restoring corrupted or lost data.
  - The system shall maintain referential integrity across related tables to prevent inconsistent state.

  Rationale: Critical for preserving years of scholarly research work and manuscript annotations. Prevents catastrophic data loss that could set back research projects by months or years.

- **Service Reliability and Fault Tolerance** *(Priority: Medium)*

  - The system shall detect service timeouts or crashes (e.g., ML inference, orchestration tasks) and automatically retry with exponential backoff.
  - The system shall provide user feedback when a service is unavailable and allow fallback to manual workflows.
  - The system shall implement failover handling for long-running processes to avoid complete job loss.
  - The system shall log all failures and exception traces for later analysis.

  Rationale: Ensures system remains usable even when individual components fail. Prevents user frustration and maintains productivity during system maintenance or component failures.

- **Model Transparency and Scholarly Control** *(Priority: High)*

- The system shall display confidence levels and similarity metrics for all model-generated suggestions (matching, classification, transcription).
- The system shall require explicit human confirmation before accepting any automated fragment match.
- The system shall maintain version control for model outputs, ensuring that updated models do not overwrite prior validated results.

Rationale: Essential for scholarly integrity and peer review. Ensures researchers can evaluate AI suggestions critically and maintain control over their research conclusions. Prevents over-reliance on potentially flawed automated decisions.

- **Network and Connectivity Safety** *(Priority: Medium)*
  - The system shall autosave the current workspace and session data locally every 60 seconds.
  - The system shall sync unsaved data automatically upon reconnection.
  - The system shall warn users of unsent uploads or unsynced work before closing the browser or session.

  Rationale: Protects users from losing work due to network interruptions or browser crashes. Critical for maintaining productivity in academic environments with unreliable internet connectivity.

- **Security of Stored and Transmitted Data** *(Priority: High)*
  - The system shall use TLS (Transport Layer Security) for all client–server communication and encrypt sensitive data in storage.
  - The system shall sanitize and validate all user inputs to prevent injection or cross-site scripting (XSS) attacks.
  - The system shall segregate user data using role-based access to prevent unauthorized access or leakage.

  Rationale: Fundamental security requirement for any system handling academic research data. Protects against data breaches, unauthorized access, and ensures compliance with institutional and legal data protection requirements.

- **Auditability and Traceability** *(Priority: Low)*
  - The system shall maintain an audit log for all critical operations, including uploads, edits, deletions, and administrative actions.
  - The system shall associate every change with a user identity and timestamp.
  - The system shall provide administrators with a secure interface to review audit logs.

Rationale: Important for troubleshooting, compliance, and understanding system usage patterns. However, can be implemented after core functionality is established and is primarily useful for administrative oversight.

7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?
4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

Table 2: Potential Losses from System Hazards

Loss Category	**Specific Losses**
Data Security	• Exposure of confidential research data • Unauthorized access to manuscript database • Unintended data sharing between users • Violation of institutional data agreements • Loss of competitive research advantage
Research Productivity	• Lost research time due to crashes or slow performance • Workflow disruption from poor interface design • Research momentum loss requiring work restart • Inefficient task completion from confusing navigation
Data Integrity	• Data corruption compromising months of work • Inadvertent modification of research data • Loss of unsaved work due to system failures • Incorrect fragment matches from inaccurate AI
User Experience	• System unresponsiveness or freezing • Frequent errors reducing user confidence • Frustration from overly complex interfaces • Inconsistent behavior across browsers/devices
Accessibility & Inclusion	• Exclusion of users with disabilities • Frustration for non-technical users • Reduced adoption by diverse user groups