Reflection and Traceability Report on Uno Flip Remix

Team 24
Zain-Alabedeen Garada
Kevin Ishak
Mingyang Xu
Jianhao Wei

1 Changes in Response to Feedback

This section outlines the changes implemented based on feedback received from TAs, our supervisor, peer reviews, and usability testing. The modifications are organized into milestones such as ”TA Feedback,” ”Peer Review,” and ”Final Documentation Updates” for easier tracking. This feedback can be found using the feedback item’s associated issue on our github.

1.1 SRS and Hazard Analysis

Changes to SRS and Hazard Analysis along with the feedback, response, and associated issues can be found in the tables below:

Table 1: Changes for SRS Documentation

Feedback Source	**Feedback Item**	**Response**	**Issue #**
TA	Undesired event handling not stated (e.g., ML predictions).	Added FR14 in Section 10 to define system behavior for undesirable ML outputs.	#84
TA	Requirements likely/unlikely to change not linked to assumptions or constraints.	Section 15 now links requirements to change likelihoods, and traces them to assumptions/constraints.	#84
TA	Missing context diagram.	Added a use case diagram and updated Section 8.1 to provide context.	#84
TA	Reflections missing.	Added a dedicated Appendix for SRS reflection.	#84

TA	Redundant acronym definitions.	Consolidated acronym definitions into Section 4.1; removed repeats.	#84
TA	Tables lack captions and formatting overflows.	Long tables are now properly formatted using `longtable` with captions and correct column widths.	#84
TA	Missing figures where applicable (e.g., business model, product boundary).	Section 7.1 now contains a class diagram, and Section 8.1 includes a system boundary diagram.	#84
TA	Legal requirements not hyperlinked.	Section 17 hyperlinks now point to referenced laws like PIPEDA and AODA.	#84
TA	Figures not included elsewhere for evaluation.	Added all referenced figures into proper LaTeX figure environments with captions and numbers.	#84
TA	3D rendering not mentioned in requirements.	Added FR11 in Section 10 mandating 3D rendering for all gameplay.	#84
TA	Ambiguous requirements: "modern design" unclear.	Clarified "modern design" in AR1 by adding UI responsiveness, contrast standards, and layout principles.	#84
TA	AR/SR sections mention a website not documented elsewhere.	Removed mentions of a website in SR1 and clarified its context.	#84
TA	Repeated section labels (e.g., AR in 14.3 and 10.1).	Renamed overlapping section labels to maintain uniqueness across document.	#84
TA	No traceability diagram or matrix included.	Added a traceability matrix in Section 26, linking FRs to PUCs, assumptions, and risks.	#84
TA	Verification methods unclear; fit criteria missing.	All FRs and NFRs now include a fit criterion column in Section 10 and 14 respectively.	#84
TA	Measurability lacks rationale.	Each requirement now includes a "Rationale" column in Section 10/14.	#84
TA	Casual vs. enthusiast gamers not addressed in requirements.	Personas refined and corresponding FRs/NFRs adapted to differentiate user needs.	#84

TA	Incorrect client listed (should be Dr. Paul Rapoport).	Replaced “McMaster University” with Dr. Paul Rapoport as primary client.	#84
TA	Dev team listed as stakeholder without justification.	Removed dev team from stakeholder list in Section 6.	#84
TA	No prioritization or phasing of requirements.	Prioritization added in Section 21; requirements tagged as Essential, Expected, or Optional.	#84
TA	No diagrams to clarify ambiguous designs.	Added new diagrams in Section 8.2 showing card interactions and turn transitions.	#84
TA	No peer review issues listed.	Section 23 now includes peer review issues with hyperlinks to GitHub.	#84
TA	Members missing from informal presentation.	Documented presentation attendance and rationale in Section 25.	#84
TA	Standards like ESRB not considered.	ESRB and ISO standards added to Section 17 and hyperlinked.	#84
TA	System/subsystem boundaries unclear.	Section 8.2 updated with diagrams and tables clarifying subsystem responsibilities.	#84
TA	Stakeholder list includes dubious roles.	Re-evaluated stakeholders and removed those with unclear impact.	#84
TA	Medical risks of 3D graphics not addressed.	Added NFR addressing visual strain/epileptic triggers in Section 14.	#84
TA	Regulatory references not hyperlinked.	Hyperlinks added throughout Section 17 for all referenced regulations.	#84
TA	Game rules not formalized as requirements.	Added FR15–FR17 to specify core Uno Flip game rule implementations.	#84
Peer Review	Document Introduction.	Introduction updated to describe UNO Flip Remix’s gameplay theme, project context, and intended user audience.	#1
Peer Review	Personas and User Characteristics.	Section 2.3 added to outline key player personas, including Competitive Player, Casual Player, and Developer.	#2

Peer Review	Traceability Matrix Missing.	Traceability matrix included at the end of Section 4, linking each requirement ID to its corresponding test case.	#3
Peer Review	Likely and Unlikely Section.	Section 5 added to explain expected feature changes (e.g., UI tweaks) and unlikely changes (e.g., rule set overhaul).	#4
Peer Review	Revision Table Format.	Revision History Log added on page 1, formatted with date, contributors, and brief change descriptions.	#5
Peer Review	Verifiable Requirements.	Functional requirements revised to include measurable criteria (e.g., “display last three player moves” instead of “track moves”).	#6
Peer Review	Ambiguous Requirements.	Reworded vague statements in Section 3.3 to clarify inputs, expected outputs, and system response.	#7

Table 1: Traceability Table for SRS Documentation — Issue #84

Table 2: Changes for Hazard Analysis

Feedback Source	**Feedback Item**	**Response**	**Issue #**
TA	List of Tables missing.	Added a formatted List of Tables before Section 1.	#88
TA	No proper attribution to Nancy Leveson’s work.	Added footnote in Section 1 referencing Leveson’s work with full citation.	#88
TA	Erroneous ‘i‘ in Table 1/Contents.	Removed stray characters from ToC entries and fixed page numbering.	#88
TA	Excessive whitespace on pg. 9.	Adjusted LaTeX spacing to remove unnecessary vertical gaps.	#88

TA	Roadmap placed on project timeline instead of implementation roadmap.	Moved roadmap to dedicated "Roadmap" section and aligned it with mitigation timeline.	#88
TA	Less than 5 issues created for hazards.	Added additional failure modes including latency sync, AI misbehavior, and invalid card handling.	#88
Peer Review	Improper GitHub Usage.	GitHub repo link and contributor references were clarified via the revision log and contributor section in the appendix.	#8
Peer Review	Improper use of Revision History.	Revision History Log added at the start of the document including contributor names, dates, and change summaries.	#9
Peer Review	Mistakes in Formatting and Attention to Detail.	Fixed alignment of list items, removed layout artifacts, added missing figure/table captions, and corrected inconsistencies across section headers.	#10
Peer Review	FMEA Table Missing Components.	Table 2 expanded to include multiple new failure modes including server crashes, client-side bugs, and network latency, each with mitigation.	#11
Peer Review	Requirements not Cascaded to the SRS.	Section 6 requirements (AR, IR, PR) are now explicitly tied to concepts introduced in the SRS, such as admin controls, authentication, and encryption.	#12
Peer Review	Roadmap for the Future.	Section 7 provides a timeline and post-mitigation review phase. Includes integration of security and hazard controls into development and testing cycles.	#13
Peer Review	Unrealistic Assumptions.	Section 4 revised with realistic assumptions: e.g., acknowledging illegal inputs, network instability, and potential desync, and identifying server-side mitigation strategies.	#14

Table 2: Traceability Table for Hazard Analysis — Issue #88

1.2 Design and Design Documentation

Changes to Design Documentation (MG and MIS) along with the feedback, response, and associated issues can be found in the tables below:

Table 3: Changes for MG and MIS

Feedback Source	Feedback Item	Response	Issue #
TA	No clear testing timeline, especially for pre-Rev 0 phase.	Added a timeline in the design document outlining test expectations and milestones prior to Rev 0.	#90
TA	Turn Management Module should be an abstract object.	Refactored Turn Management Module as an abstract object in MG and MIS, indicating logic-heavy design.	#90
TA	Broken references in MG and MIS.	Repaired all broken internal references and links throughout the documents.	#90
TA	Traceability matrix lacks hyperlinking and module identifiers.	Enhanced traceability matrix with hyperlinks and clearly labeled module identifiers.	#90
TA	Inconsistent uses hierarchy and module decomposition.	Unified module listings across hierarchy, decomposition, and uses diagram; added missing modules.	#90
TA	Game Logic and Score Tracking module listed in MG but missing from MIS.	Added detailed MIS entries for Game Logic and Score Tracking modules.	#90
TA	Verification Module in MIS but missing from MG.	Added Verification Module to MG hierarchy and traceability matrix.	#90
TA	Exported constants lack type information.	Added types (e.g., int, String) to all exported constants in MIS modules.	#90
TA	Exported access programs lack output descriptions.	Provided full descriptions and output formats for all access routines.	#90
TA	UI module could be simplified via a state diagram.	Added UI state diagram to MG Section 10 to clarify flow and transitions.	#90

TA	Game logic unclear; likely due to missing Game Logic module.	Reintroduced and elaborated the Game Logic module in both MG and MIS.	#90
TA	Uno Flip rule logic not specified.	Detailed Flip-side logic in Card Effect and Turn Management modules in MIS.	#90
TA	No GitHub Actions included.	Added GitHub Actions pipeline explanation in design documentation and CI/CD workflow.	#90
Peer Review	Hyperlinks between MIS sections are missing.	Cross-references were added throughout to link module names with their respective MIS subsections for smoother navigation.	#42
Peer Review	Inconsistent syntax definitions across modules.	Syntax section was updated across all modules to consistently use ‘Exported Constants‘ and ‘Access Programs‘ format, reducing structural discrepancies.	#51
Peer Review	Missing semantics for some access routines.	All access routines now include transition or output details in their ‘Access Routine Semantics‘ section, ensuring completeness.	#52
Peer Review	Vague testing coverage in timeline.	Testing plans for MIS modules were clarified to indicate test types and coverage expectations in the corresponding unit test mapping (see VnVReport Table 5).	#53
Peer Review	Lack of cohesion in module responsibilities.	Module responsibilities were refined: Save/Load now focuses solely on persistence logic, and Animation separates display logic from game logic.	#54
Peer Review	MG7 Multiplayer networking module in Software decision.	Section 7.5.1 clearly defines the Multiplayer Networking Module's secrets and services, including protocol use, matchmaking logic, and synchronization.	#44
Peer Review	MG7 UI Module in Software decision.	Section 7.5.2 expanded to explain layout, HUD components, interaction model, and accessibility assumptions. Services and responsibilities clarified.	#45

Peer Review	Missing modules in Use Hierarchy diagram (MG-9).	Use hierarchy in Section 9 was updated to include previously missing leaf modules and reflect accurate dependencies for CE, UI, and VO modules.	#46
Peer Review	Legend for shorthands in Use Hierarchy Diagram.	A legend was added beneath Figure 1 to decode abbreviations like CE, VO, UI, TM, and SL, improving diagram clarity.	#47
Peer Review	MG12 Timeline Feedback.	Timeline in Section 12 includes clear task phases, member assignments, and versioning alignment for MG Rev 0 and Rev 1. Responsibilities are explicitly tracked.	#48
Peer Review	Consistency issues regarding module details between the Module Guide and MIS.	Module services, naming, and responsibilities were aligned with the MIS; table headers and decomposition structure were revised for clarity and consistency.	#49

Table 3: Traceability Table for MG and MIS Design Implementation — Issue #90

1.3 VnV Plan and Report

Changes to VnV Plan and Report along with the feedback, response, and associated issues can be found in the table below:

Table 4: Changes for VnV Plan

Feedback Source	**Feedback Item**	**Response**	**Issue #**
TA	Standard VnVPlan.tex/pdf template not used.	Replaced previous document structure with course-standard LaTeX template, including properly formatted title page, section headers, and table styles.	#89
TA	Appendix used incorrectly to present test plan.	Moved test plan content from Appendix into main body under appropriate test sections (4.x). Appendix now includes only supporting materials like survey screenshots and raw data.	#89

TA	Improperly formatted List of Tables; no hyperlinks.	Reformatted List of Tables to enable clickable links to all table entries.	#89
TA	Wasted space in Section 4 due to bullet/number indentation.	Modified lists and environments to remove excessive indentation and used compact itemization for clearer and space-efficient layout.	#89
TA	Inconsistent text style between 4.1 and 4.2.	Standardized text formatting in Section 4 using consistent fonts, spacing, and test case structure across all sub-sections.	#89
TA	Hazard Analysis document not referenced.	Added cross-references to Hazard Analysis in Sections 3.7 and 4, linking risk-related tests with identified hazards.	#89
TA	Verification tools in Section 3.6 not specific.	Updated Section 3.6 to specify tools such as NUnit, Unity Test Framework, Unity Profiler, and GitHub Actions for CI/CD.	#89
TA	Appendix mentions Java and JavaScript despite Unity C# project.	Removed irrelevant mentions of Java and JavaScript tools. Revised all references to align with Unity and C ecosystem.	#89
TA	Ambiguous test input descriptions (e.g., ”System manages player turns”).	Revised all test cases to clearly separate ’Inputs’ and ’System State’ and used specific actions and expected behaviors.	#89
TA	Test descriptions are high-level and vague.	Expanded each test case with step-by-step procedures, clear derivation, control type, input/output, and validation expectations.	#89
TA	Suggestion to convert test tables to matrices.	Converted test cases into matrix-style tables with traceability to requirements for clarity and structured presentation.	#89
TA	Tests labeled as static are actually dynamic.	Reviewed and reclassified test types; static tests (e.g., code reviews, static analysis) clearly distinguished from dynamic ones.	#89
TA	Quality of written issues is low.	Revised issue tracker content and documented issues with concise summaries, links to code, and consistent formatting.	#89

TA	Only two extras allowed; justification missing.	Listed only two extras (Usability Testing and GenderMag Analysis) with detailed justification in Section 2.3.	#89
TA	Static testing wrongly interpreted in some cases.	Clarified interpretation of static vs. dynamic testing; added examples of both in test descriptions and Section 3.5.	#89
TA	NFR/FR test descriptions ambiguous.	Separated functional and non-functional tests in Sections 4.1 and 4.2 with clearly defined scope and expected outcomes.	#89
TA	Appendix content on Java/JS tools is misplaced.	Removed misplaced tooling content from Appendix and added raw survey data screenshots instead.	#89
TA	Reflection appears to have only two participants.	Updated Section 12 to include contributions from all team members with summary of their reflections.	#89
TA	Shared doc platform not ideal for contribution tracking.	Explained GitHub issue tracker and commit history use for tracking team contributions; included sample evidence.	#89
TA	Grammatical issues throughout.	Performed grammar review using both peer feedback and automated tools; edited entire document for clarity.	#89
Peer Review	List of Tables is not Complete and Automatic.	Implemented a List of Tables with hyperlinks to improve navigation.	#15
Peer Review	Multiple Style Errors.	Fixed font size inconsistencies, unified header styling, normalized test case table format, and removed spacing artifacts.	#16
Peer Review	Improper use of GitHub.	Section 2.4 and Appendix now include commit hashes, GitHub repository links, and descriptions of test workflows in GitHub Actions.	#17
Peer Review	Inputs to Functional Req. Tests not Allways Specific.	Rewrote test case inputs in Section 4.1 to distinguish between user actions, system state, and environmental conditions.	#18

Peer Review	Missing some Tests for NFR.	Added NFR test cases in Section 4.2 to cover accessibility, maintainability, availability, and appearance.	#19
Peer Review	Unclear NFR Test Criteria.	Reworded all NFR test descriptions in Section 4.2 to include expected outcomes, success thresholds, and measurement conditions.	#20
Peer Review	Reflection Skills to Acquire is Vague.	Expanded Section 6 with a paragraph for each team member outlining their learning outcomes, challenges, and skills to build.	#21

Table 4: Traceability Table for VnV Plan — Issue #89

Table 5: Changes for VnV Report

Feedback Source	**Feedback Item**	**Response**	**Issue #**
TA	Missing hyperlink for `report.pdf` on pg. 12.	Added clickable hyperlink to `report.pdf` in Section 6 introduction using LaTeX `\href` command.	#91
TA	Inline code should use monospaced font (not bold).	Replaced all inline code formatting with `\texttt{}` or `lstlisting` where appropriate for clarity and readability.	#91
TA	More analysis needed for usability data (35% observed lag).	Expanded Section 4.1.1 to discuss observed lag, contributing factors, and future UI performance improvements.	#91
TA	Survey results used for NFRs need more detail.	Clarified methodology for concurrent user simulation in Section 4.2.5; linked outcomes to capacity requirements.	#91
TA	Survey data does not fully support NFR satisfaction.	Added follow-up analysis in Section 4.2.5 and reflected limitations of survey scope; outlined next steps for addressing gaps.	#91

Peer Review	Changes made in response to VnV.	Added Section 7 to describe coverage limitations and newly added test cases for missing features including chat and multiplayer.	#71
Peer Review	Document hyperlinks for clarity.	Embedded internal hyperlinks in traceability and unit test sections (Sections 9–10) to improve document navigation.	#72
Peer Review	Types of Code Coverage Techniques Used Unexplained.	Section 11 now explains the use of Unity's built-in test coverage tools and outlines both Edit Mode and Play Mode strategies.	#73
Peer Review	Requirements not having associated tests.	Traceability Tables 1–4 in Section 9 were expanded to include missing requirement-to-test mappings such as CFR2 and PR2.	#74
Peer Review	Unclear Justification for 80% Code Coverage Target.	Section 11 clarifies the 80% coverage target and breaks down achieved percentages by module (see Figures 9–12).	#75
Peer Review	Justification for the Need for a New CI Action Lacking.	Section 8 discusses limitations of current CI tooling and outlines need for Unity-compatible CI enhancements for future tests.	#76
Peer Review	Inconsistencies between VnV Plan Tests and VnV Report tests.	Reflection appendix details differences between planned and executed tests, with justifications for any changes.	#77
Peer Review	Test Cases should show which requirement they are validating more clearly.	All test cases in Sections 3–4 were updated with explicit references to their originating requirement IDs.	#78

Table 5: Traceability Table for VnV Report — Issue #91

2 Challenge Level and Extras

2.1 Challenge Level

The challenge level for the project is **Basic** as agreed upon by the course instructor. This classification correctly represents the project's scope and complexity.

2.2 Extras

This project incorporated three extras: usability testing, GenderMag personas, and a user guide. Usability testing involved 17 participants and focused on evaluating UI clarity, game flow, accessibility, and engagement; results led to actionable improvements such as adding a tutorial, refining the UI, and implementing in-game help. The team also created and applied two GenderMag personas (Abi and Tim) to assess inclusivity in user interactions, which resulted in design changes like improved exit visibility and implementing non-gendered terminology. Finally, a detailed user guide was developed, providing step-by-step instructions, visual aids, and troubleshooting tips to support onboarding across various setups.

3 Design Iteration (LO11 (PrototypeIterate))

The final design and implementation of UNO Flip Remix took shape through an iterative process grounded in user feedback, validation, and continuous refinement. Early documentation established the foundation for gameplay mechanics and multiplayer logic, but initial drafts revealed weaknesses in cohesion and testability. Peer reviews and TA feedback identified vague requirements and unclear module responsibilities, prompting significant restructuring across all documents to support clarity and traceability. Usability testing played a key role in refining the user experience. Participants struggled with interface clarity and onboarding, which led to the addition of a tutorial on how to play and revised feedback prompts. Feedback from these sessions directly shaped improvements to navigation and input clarity, ensuring that the system responded intuitively to player actions. These evolving design decisions were grounded in user needs and continually evaluated through validation planning and traceability analysis, ensuring the final product remained accessible and aligned with client expectations.

4 Design Decisions (LO12)

The project faced key limitations in tooling support, platform compatibility, and networking stability, all of which directly influenced core design choices. For example, Unity's multiplayer capabilities posed challenges with synchronization and desync events, which were documented in the VnV Plan and led the team to adopt a simplified local multiplayer test mode early in development. Similarly, CI tooling lacked full compatibility with Unity's play mode testing, limiting automation and prompting a shift toward manual test tracking and Unity's built-in tools. These limitations helped narrow the project's scope and encouraged a more focused, maintainable implementation of essential features like card flipping, turn management, and UI clarity.

Several assumptions outlined in the SRS and Hazard Analysis documents shaped the early system design. The team initially assumed that users would be familiar with UNO mechanics and playing on desktop systems with reliable network connections. This assumption informed early UI designs that lacked tooltips and in-game instructions. However, usability testing later contradicted this by revealing confusion around flip mechanics, leading to the inclusion of an interactive tutorial and accessible help features. Similarly, the assumption of network stability was challenged during multiplayer trials, prompting the integration of improved state synchronization to handle miscommunication issues.

Time constraints, team experience, and course requirements shaped the overall project scope and technical stack. The development plan explicitly prioritized a rule-based AI over reinforcement learning to reduce complexity within the available timeline. Additionally, course-imposed deliverables like the MIS and MG encouraged rigorous modular decomposition, which resulted in clear boundaries between modules such as Turn Management, Card Effects, and Networking. These constraints ensured the final product remained aligned with both technical feasibility and academic expectations.

5 Economic Considerations (LO23)

UNO Flip Remix targets a growing market of casual and competitive digital card game players who seek engaging multiplayer experiences beyond existing options like Uno Online. The game's innovative flip mechanic and online synchronization could give it niche appeal within the gaming space. Marketing would involve targeted outreach through platforms like Steam, Discord, and mobile app stores, along with community-based promotion via Reddit posts and streamers who play similar games such as Hearthstone. Based on the current implementation and features, the estimated production cost for a commercial-ready version, factoring in hosting and support—would be around $8,000$–10,000. At a projected sale price of $.99, about 2,000–2,500 units would need to be sold to break even. Alternatively, if we use an open-source model, user attraction could center on GitHub visibility, subreddit engagement, and plugin-mod support. With over 100 million UNO players globally and strong mobile adoption rates, the potential user base for a digital remix with innovative gameplay mechanics should exceed a million users, assuming competitive gameplay, visibility and sustained updates.

6 Reflection on Project Management (LO24)

6.1 How Does Your Project Management Compare to Your Development Plan

Our project management mostly aligned with the original Development Plan, especially following the proof-of-concept (POC) demo. While we initially underused GitHub's issue tracker and committed large code blocks, we adjusted our practices in the second semester by using GitHub Issues to organize and monitor tasks more effectively. We consistently followed our meeting and communication plan, holding weekly syncs on Microsoft Teams and increasing frequency as deadlines approached. The team's workflow model, which included a Git-flow structure and peer-reviewed pull requests, was generally followed throughout the project.

6.2 What Went Well?

We successfully completed our core gameplay mechanics and learned to use new technologies, including Unity and C# . The team adapted well to rotating roles, functioning effectively without a formal project leader. Work was delegated clearly, and communication improved steadily. Once we adopted GitHub Issues more seriously, our collaboration and accountability became more structured, making it easier to track progress and manage deliverables.

6.3 What Went Wrong?

Early in the project, we struggled with proper version control practices. We did not correctly use GitHub to its full abilities as we rarely committed incrementally and didn't use the issue tracker during the first semester. This made it difficult to track changes or distribute tasks clearly. On the technical side, the team faced a steep learning curve with Unity and C# which delayed our implementation timeline and required additional time for debugging and research during the early stages.

6.4 What Would you Do Differently Next Time?

In future projects, we would adopt GitHub Issue tracking from the very beginning, as it proved to be a powerful organizational and planning tool once fully integrated. Additionally, we would make smaller, more frequent commits to improve traceability and simplify debugging. These practices would ensure better documentation of our development process and make it easier to manage both code and collaboration from day one.

7 Reflection on Capstone

7.1 Which Courses Were Relevant

SFWRENG 3DB3 - Databases.

ENGINEER 3PX3 - Integrated Engineering Design Project 3.

SFWRENG 3A04 - Software Design III.

SFWRENG 4HC3 - Human-Computer Interfaces.

SFWRENG 3RA3 - Software Requirements and Security Considerations.

7.2 Knowledge/Skills Outside of Courses

- TCP Server Implementation.
- Deep understanding of UNO Flip rules.
- C and Unity.