Reflective Report

Design and Creative Technologies

Torrens University, Australia

Student: Luis Guilherme de Barros Andrade Faria - A00187785
Subject Code: MFA 502
Subject Name: Mathematical Foundations of Artificial Intelligence
Assessment No.: 2A
Title of Assessment: Reflective Report, Set 1, Problem 1
Lecturer: Dr. James Vakilian
Date: Oct 2025

Table of Contents

1. Introduction and Overview

The idea of developing *EigenAI* emerged after a discussion with my lecturer, Dr. James Vakilian, where it was emphasized that the project should not only perform the requested calculations but also provide an interactive environment allowing any user to input their own matrices and obtain results dynamically.

From that conversation, I envisioned creating a smart learning assistant that combines theory and computation: a web application where students can experiment with mathematical operations while visualizing the process through a clear, guided interface.

My approach was to build *EigenAI*, a modular web system where users could input matrices, trigger determinant or eigenvalue computations, and observe step-by-step explanations of the results.

I chose *Streamlit* for the frontend because of its simplicity and interactive capabilities, allowing rapid UI development integrated with pure Python logic on the backend.

The project follows a layered architecture:

- **Presentation layer**: Streamlit interface (user input, progress animations, and results visualization).
- **Business logic layer**: Custom Python functions implementing the determinant calculation.
- **Persistence layer**: Streamlit's in-memory session state (lightweight storage for user inputs and temporary data).

This structure reflects the Software Engineering Principles I learned in the previous term: separation of concerns, modularity, and maintainability.

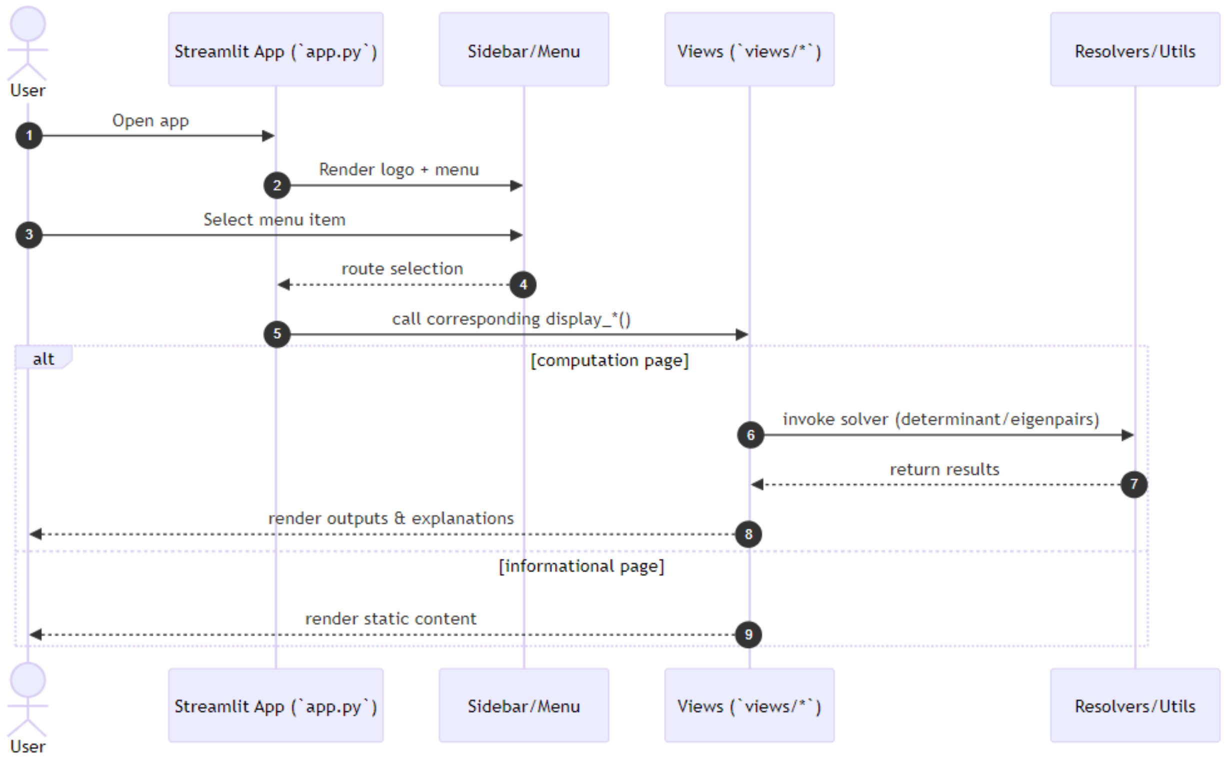

Figure 1: Sequence diagram illustrating the interaction between `app.py`, sidebar/menu, views, and resolvers/utils modules.

2. Mathematical Approach

The determinant algorithm relies on the Laplace (cofactor) expansion and recursive decomposition. At its core, the determinant expresses how a matrix transforms space: whether it scales, compresses, or collapses it.

The recursive algorithm expands along the first row of the matrix and repeatedly removes one row and one column to compute the minor matrices, alternating signs (+/−) in each recursive call.

The computation stops when it reaches the base cases:

- For 1×1 → det = A_{00}
- For 2×2 → det = (ad − bc)

The final determinant is the cumulative sum of the scaled minors.

This approach reflects both the mathematical definition and a divide-and-conquer programming pattern, mirroring how AI systems recursively solve sub-problems in optimization or neural-network computations.

3. Programming Methods

The determinant logic is implemented in `determinant.py` with the following key functions:

- `shape(A)` and `is_square(A)` to validate inputs.
- `minor_matrix(A, drop_row, drop_col)` to construct sub-matrices.
- `determinant(A)` to compute recursively.

These functions were integrated with the Streamlit UI (`set1Problem1.py`) where users enter values row by row. Upon submission, the system runs the determinant logic, updates a progress bar (`st.progress()`), and displays results with step explanations.

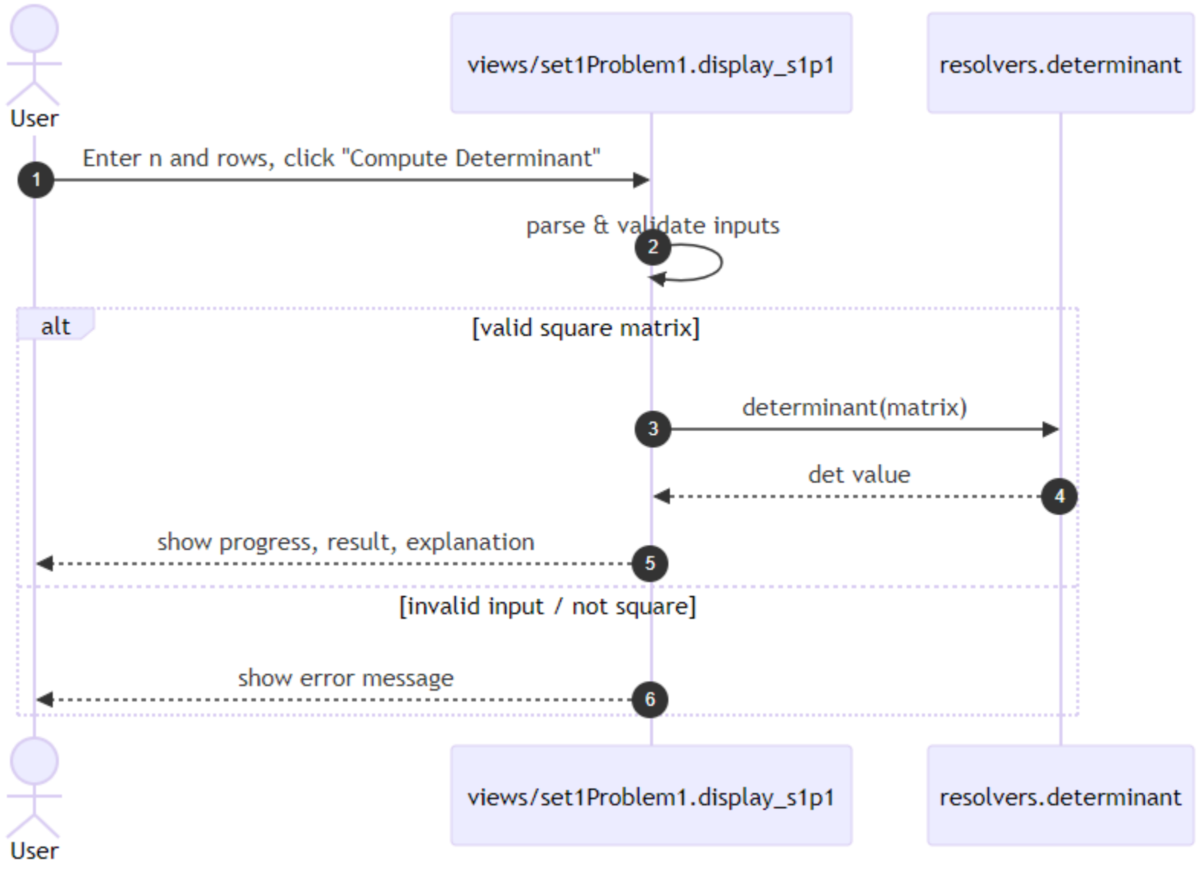
User
views/set1Problem1.display_s1p1
resolvers.determinant
1
Enter n and rows, click "Compute Determinant"
2
parse & validate inputs
alt
[valid square matrix]
3
determinant(matrix)
4
det value
5
show progress, result, explanation
[invalid input / not square]
6
show error message
User
views/set1Problem1.display_s1p1
resolvers.determinant

Figure 2: Sequence diagram showing the flow between `views/set1Problem1.py` and `resolvers/determinant.py` modules.

Error handling was added for:

- Non-square matrices,
- Empty input fields,
- Invalid numeric values.

This makes the system resilient and user-friendly, essential characteristics for an educational tool.

4. What Went Right

- Clear separation between UI and logic, allowing independent testing.
- Educational user experience: users can learn through interaction rather than passive observation.
- Streamlit Cloud provided a lightweight and accessible deployment environment for showcasing both Set 1 and Set 2 problems.

5. What Went Wrong

- Initial difficulty parsing comma-separated inputs into matrix lists.
- Minor bugs appeared with matrices containing negative or floating-point numbers.
- Recursive stack usage grows quickly for matrices larger than 6×6, limiting performance.

6. Uncertainties

I am still exploring performance optimizations for larger matrices and future versions might introduce memorization, transition the logic to NumPy (for matrix slicing efficiency), or event integrate a robust framework like Fast API or Flask, although for the current assessment, only pure Python was allowed.

7. Personal Insight

Initially, the determinant felt like a purely mechanical process. However, once I implemented the recursive expansion, its elegance and structure became clear — each minor matrix represents a smaller window into the overall geometry of transformation.

Writing the algorithm manually forced me to think like the computer while reasoning like a mathematician, bridging intuition and formal logic.

This was my first time designing a recursive mathematical function for an educational platform, and the sense of seeing it come alive visually through Streamlit was deeply satisfying.

8. Conclusion

This project solidified my understanding of both recursion and matrix algebra as essential building blocks in Artificial Intelligence. It reinforced the link between theoretical math and real-world software design, where algorithms are not just abstract proofs but functional components of intelligent systems.

In future iterations, it is planned to optimize the determinant solver and extend EigenAI with visual graph explanations for matrix transformations.

Statement of Acknowledgment

I acknowledge that I have used the following AI tool(s) in the creation of this report:

- OpenAI ChatGPT (GPT-5): Used to assist with outlining, refining structure, improving clarity of academic language, and supporting APA 7th referencing conventions.

I confirm that the use of the AI tool has been in accordance with the Torrens University Australia Academic Integrity Policy and TUA, Think and MDS's Position Paper on the Use of AI. I confirm that the final output is authored by me and represents my own critical thinking, analysis, and synthesis of sources. I take full responsibility for the final content of this report.

9. References

Dash, R. B., & Dalai, D. K. (2008). *Fundamentals of Linear Algebra*. ProQuest Ebook Central.

Torrens University Australia (2025). *MFA501 Module Notes – Linear Transformations and Matrix Operations.*

EigenAI Project (2025). *GitHub Repository.* Retrieved from https://github.com/lfariabr/masters-swe-ai/tree/master/2025-T2/T2-MFA/projects/eigenai

Streamlit, Inc. (n.d.). *Streamlit documentation.* Retrieved from https://docs.streamlit.io/