FTEC 5660 HW1 Report

Zhuoyu Li, 1155246051

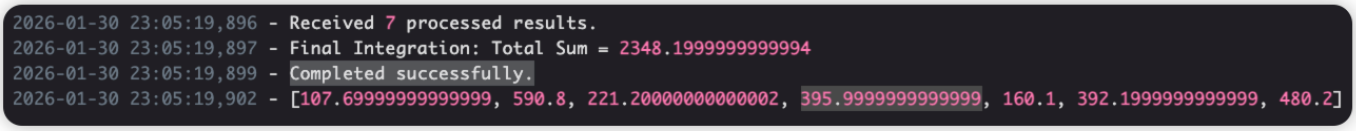
The implementation is designed as a modular pipeline using Langgraph with the following key components:

1. Planning Stage
   The agent first interprets the user query and determines whether it falls within the supported capability scope (e.g., total amount paid, or amount before discount). If the query is unsupported, the agent safely interrupts execution.

2. Analysis Stage
   For valid queries, the agent dynamically generates a Python function `compute(receipt)` that encodes the required calculation logic. This function is generated once per query and reused across all receipts, ensuring consistency and minimizing redundant reasoning.

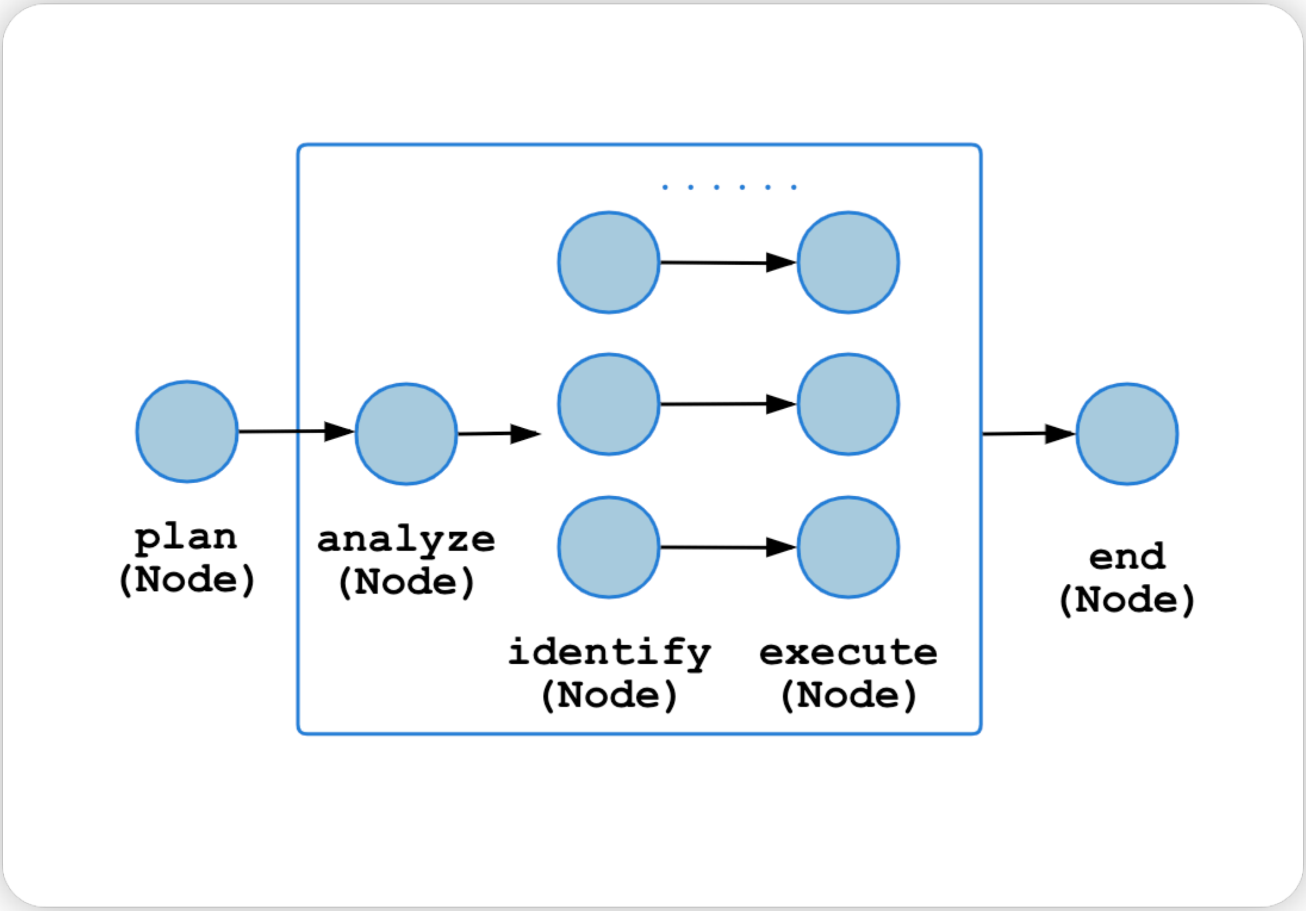
3. Execution Stage (Per Receipt)
   Each receipt image is processed independently: LLM extracts structured bookkeeping data (items, quantities, line subtotals, discounts, subtotal, rounding). Then, the generated `compute` function is executed in a restricted environment to compute a numeric result for that receipt. All receipt executions are run in parallel to improve efficiency

Finally, the agent aggregates the per-receipt results to produce the final answer like:



```
2026-01-30 23:05:19,896 - Received 7 processed results.
2026-01-30 23:05:19,897 - Final Integration: Total Sum = 2348.1999999999994
2026-01-30 23:05:19,899 - Completed successfully.
2026-01-30 23:05:19,902 - [107.69999999999999, 590.8, 221.20000000000002, 395.9999999999999, 160.1, 392.1999999999999, 480.2]
```

The node and workflow design:


plan
(Node)
analyze
(Node)
identify
(Node)
execute
(Node)
end
(Node)