LLM vs Manual Analysis

Classes and Interfaces we identified:

Interfaces:

- IndexManager
- Runnable
- WeblogEntriesPager

Classes:

- LuceneIndexManager
- IndexOperation
- WriteToIndexOperation
- ReadFromIndexOperation
- AddEntryOperation
- ReIndexEntryOperation
- RemoveEntryOperation
- RebuildWebsiteIndexOperation
- RemoveWebsiteIndexOperation
- SearchOperation
- SearchResultMap
- SearchResultList
- FieldConstants
- IndexUtil
- SearchServlet
- OpenSearchServlet
- WeblogSearchRequest
- SearchResultsModel
- SearchResultsPager
- SearchResultsFeedPager

Classes identified by LLM:

- SearchResultList
- LuceneIndexManager
- FieldConstants
- IndexUtil
- IndexOperation
- ReadFromIndexOperation
- WriteToIndexOperation
- SearchOperation
- AddEntryOperation
- ReIndexEntryOperation
- RemoveEntryOperation
- RebuildWebsiteIndexOperation
- RemoveWebsiteIndexOperation

- WeblogEntry
- Weblog
- WeblogCategory
- WeblogEntryComment
- User
- WeblogEntryWrapper

Analysis:

We have included indexing logic, search operations, servlets, models, pagers, utilities, but LLM has removed UI and presentation layers.
LLM grouped classes logically and it focussed on functional responsibilities

Component	Type	Manual UML	LLM UML
IndexManager	Interface	✓	✗
Runnable	Interface	✓	✗
WeblogEntriesPager	Interface	✓	✗
LuceneIndexManager	Class	✓	✓
IndexOperation	Class	✓	✓
WriteToIndexOperation	Class	✓	✓
ReadFromIndexOperation	Class	✓	✓
AddEntryOperation	Class	✓	✓
ReIndexEntryOperation	Class	✓	✓
RemoveEntryOperation	Class	✓	✓

RebuildWebsiteIndexOperation	Class	✓	✓
RemoveWebsiteIndexOperation	Class	✓	✓
SearchOperation	Class	✓	✓
SearchResultMap	Class	✓	✗
SearchResultList	Class	✓	✓
FieldConstants	Class	✓	✓
IndexUtil	Class	✓	✓
SearchServlet	Class	✓	✗
OpenSearchServlet	Class	✓	✗
WeblogSearchRequest	Class	✓	✗
SearchResultsModel	Class	✓	✗
SearchResultsPager	Class	✓	✗
SearchResultsFeedPager	Class	✓	✗
WeblogEntry	Domain Class	✗	✓
Weblog	Domain Class	✗	✓
WeblogCategory	Domain Class	✗	✓

WeblogEntryComment	Domain Class	✗	✓
User	Domain Class	✗	✓
WeblogEntryWrapper	Domain Class	✗	✓

Completeness:

- The manually created UML is very complete, as it captures almost all classes, relationships, and implementation details present in the code.
- The LLM-generated UML is moderately to highly complete, since it focuses only on the most relevant parts of the subsystem and omits secondary or UI-related elements.

Correctness:

- In terms of correctness, manual UML and LLM generated UML are accurate on high level but LLM generated one is abstracted and may miss some low level implementation details.
- On the other hand, the level of detail in manual UML is high, hence that makes it suitable for understanding the exact code structure but it makes the UML harder to read. LLM generated UML is much more readable

Effort:

Manual:

- Creating the manual UML required significant time and effort because it involved carefully reading and understanding the source code to identify relevant classes and interfaces for each subsystem.
- Each class had to be analyzed to document its functionality and interactions, which was time-consuming due to the size and complexity of the codebase.
- Creating UML diagrams using PlantUML also required additional time to correctly model inheritance and relationships, often needing multiple revisions.
- Identifying design strengths, weaknesses, and stating modeling assumptions further increased the overall effort.

LLM:

- In contrast, the LLM-assisted approach required considerably less time and effort.

- The LLM quickly identified core classes and generated a high-level UML representation, reducing the need for extensive code exploration.
- For UML we only had to verify the correctness. But few interfaces and UI related classes were omitted.