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

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.

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

Creating the manual UML requires significant effort, including time spent reading and understanding the code. The LLM-generated UML required much less effort.