Implementation Plan

1. **SearchEngine**

- **Attributes:**
  - `crawler`: An instance of `WebCrawler` responsible for fetching web pages.
  - `parser`: An instance of `HTMLParser` used to extract plain text from web pages.
  - `indexer`: An instance of `ContentIndexer` responsible for indexing content for fast search.
- **Methods:**
  - `SearchEngine()`: Constructor to initialize the search engine with necessary components.
  - `buildIndex()`: Builds an index from the crawled and parsed content.
  - `search()`: Searches the indexed content for a query.

---

2. WebCrawler

- **Methods:**
  - `crawl()`: Starts the crawling process to fetch web pages.
  - `fetchPage()`: A private method to fetch an individual page from the internet.

---

3. HTMLParser

- **Methods:**
  - `extractText()`: Extracts plain text content from HTML pages.

---

4. ContentIndexer

- **Attributes:**
  - `index`: A map (Dictionary) that stores a mapping of search terms (Strings) to lists of pages (List of Strings) that contain those terms.
- **Methods:**
  - `addToIndex()`: Adds a page's content to the index.
  - `search()`: Searches the index for specific terms.
  - `getIndex()`: Retrieves the current index.

---

5. Logger

- **Methods:**

    - `log()`: A static method used to log important events or errors in the system. (Denoted by `$` to indicate its static nature)

---

6. Main

- **Methods:**
    - `main()`: The main entry point of the application. (Static method, denoted by `$`)

Relationships Between Classes:

- **Composition Relationships:**
    - `SearchEngine` has a **composition** relationship with `WebCrawler`, `HTMLParser`, and `ContentIndexer`. This means `SearchEngine` owns and manages these objects (filled diamond).

- **Dependency Relationships:**
    - `SearchEngine` and `Main` both have a **dependency** relationship with `Logger` (represented by a dotted arrow). This indicates that `SearchEngine` and `Main` rely on `Logger` for logging events.
    - `Main` also has a **dependency** relationship with `SearchEngine`, as it calls the methods of `SearchEngine` to start the search process.

Role of Each Class:

- **SearchEngine**: The heart of the system, managing the search process by coordinating crawling, parsing, and indexing.
- **WebCrawler**: Fetches web pages from the internet for processing.
- **HTMLParser**: Extracts plain text from web pages to enable indexing.
- **ContentIndexer**: Stores the parsed content in an index, allowing for efficient searching.
- **Logger**: Captures important system events and errors for troubleshooting and monitoring.
- **Main**: The entry point of the application, responsible for starting the process and managing user interaction.

Summary of the projects plan:

This diagram and the accompanying descriptions provide a clear overview of the project structure. It not only breaks down the attributes and methods of each class but also outlines the relationships between them. I will be using Agile methodology extreme programming (XP) variant thus, it is important to understand that any kinds of the project's implementation might change overtime as i began familiarize my self with the nature of the project.