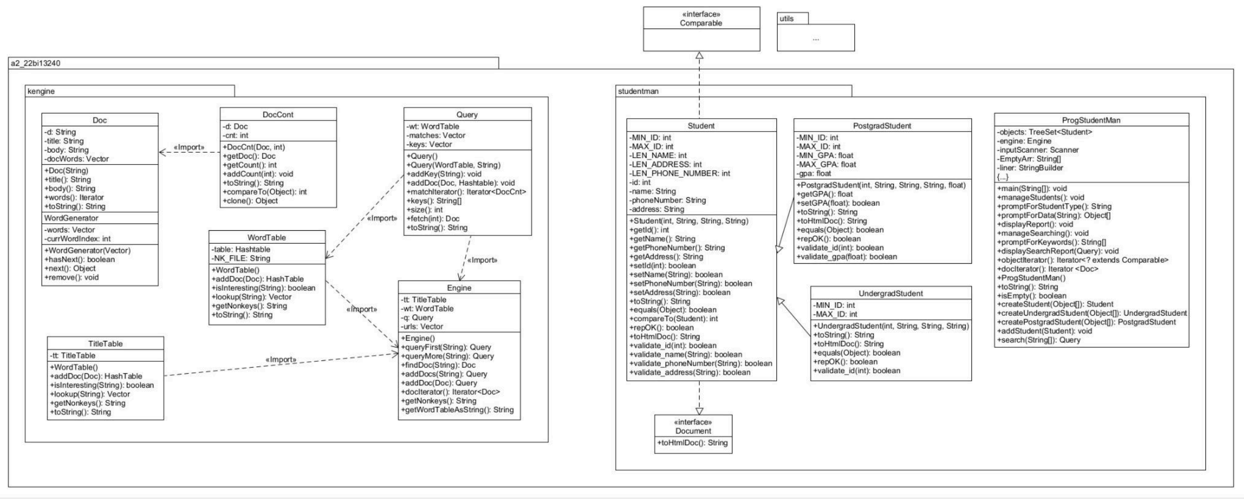
1. **What is the purpose of using the two interfaces Comparable and Document?**
    - **The Comparable interface enables us sorting while Document acts like a template to our classes.**

    Is there another way of designing the same functionality (a) without using Comparable and (b) without using Document? Briefly explain why or why not under each case?
    - Without <Comparable> we can still implement sorting using other algorithms (i.e Bubble Sort, Quick Sort) but it will take a lot of effort and testing to yield the results of Comparable.
    - Without <Document>, the code will still run but will lack some consistency and lookability throughout. An alternative is to code the toHtmlDoc() in each program.
2. **What is the purpose of using the TreeSet class in the software? Is it possible to develop the same software functionality without using this class? Briefly explain why or why not?**
    - It maintains a sorted structure regardless of what gets removed, added or modified.
    - An alternative is to implement my own sorting function, but this will take more effort and hinder the code in terms of efficiency.
3. **The original KEngine library can only search for text documents using keywords. What makes it possible to use this component in your software to search for Student objects using keywords?**
    - Since the <Document> standardizes the format of the code, Kengine can easily search for the keywords with index.
4. **Draw a complete UML design class diagram of the software showing the relevant classes (with attributes and operations) and their dependencies.**



5. **Based on the design class diagram and the previous tasks, identify the implementation strategy that was used to build the software. Briefly discuss this strategy.**

- This strategy promotes code reuse and flexibility. Inheritance allows common functionality to be shared among Studentsubclasses, reducing redundancy. Polymorphism enables treating different student types uniformly, especially when converting them to HTML documents or comparing them. Encapsulation hides internal details of student management and document generation, providing a clean API for interacting with the system.