

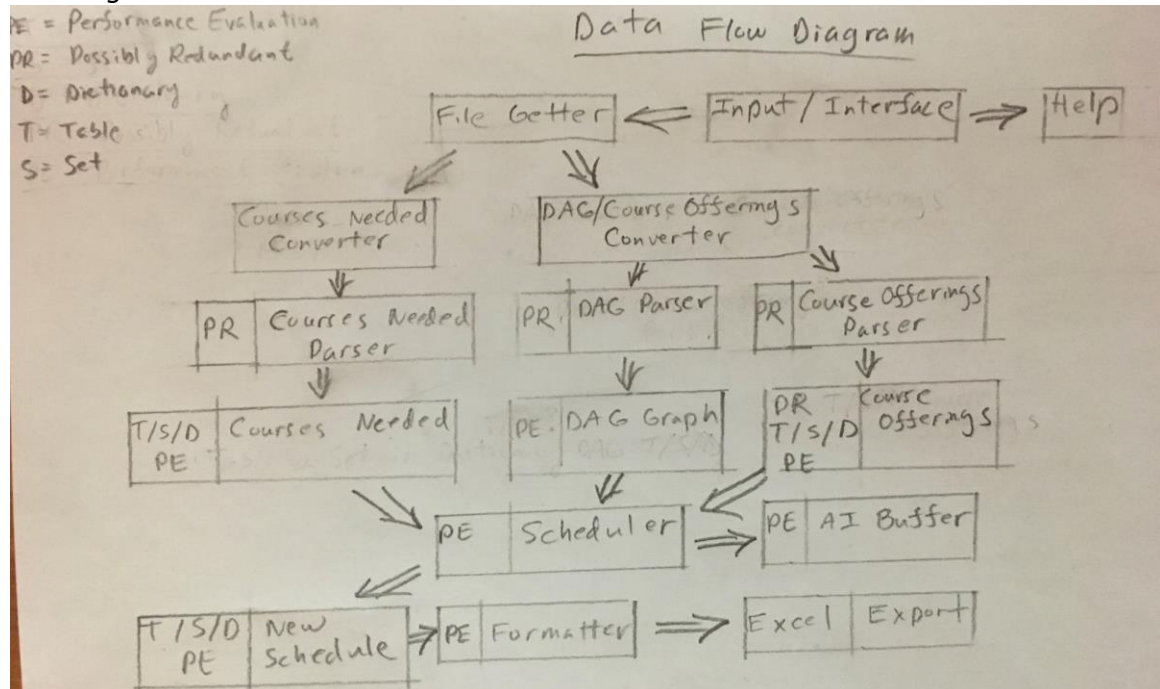
```

sequenceDiagram
    participant User
    participant File-Getter
    participant File-Parser/Converter
    participant Scheduler
    participant Formatter

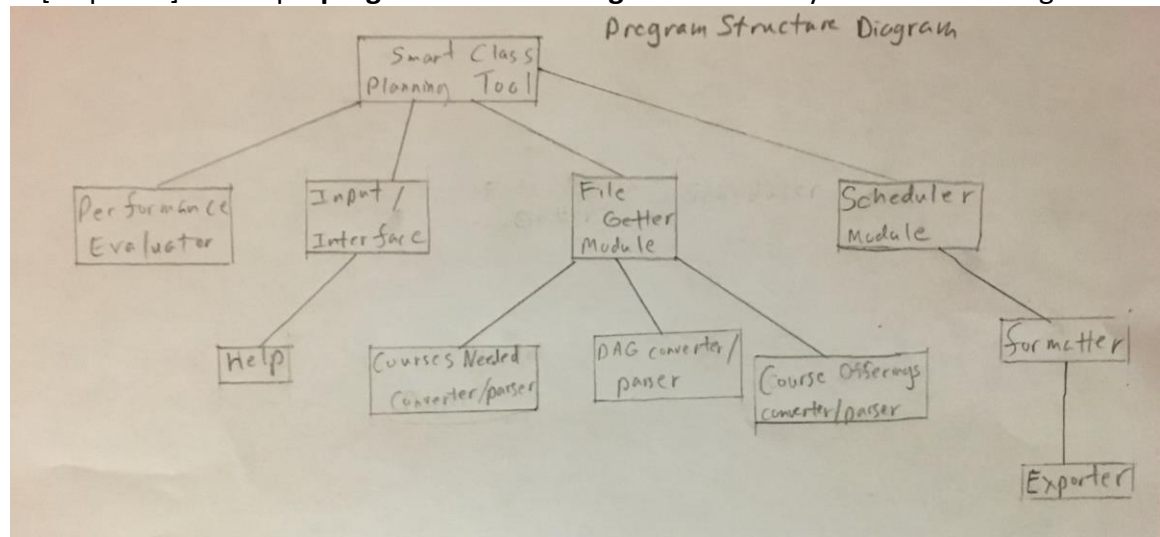
    User->>File-Getter: downloads courses_needed.pd5
    File-Getter->>File-Parser/Converter: input 1 (file1)
    File-Parser/Converter->>Scheduler: needed_courses_dictionary/set
    File-Getter->>File-Parser/Converter: input 2 (file2)
    File-Parser/Converter->>Scheduler: DAG-parser(file2)
    File-Parser/Converter->>Scheduler: DAG graph
    File-Getter->>File-Parser/Converter: input 3 (file3)
    File-Parser/Converter->>Scheduler: offering-parser (file3)
    File-Parser/Converter->>Scheduler: offering-selection/set
    Scheduler->>Scheduler: AI buffer
    Scheduler->>Scheduler: scheduler(needed_courses, DAG, offering)
    Scheduler->>Scheduler: new-schedule
    Scheduler->>Formatter: format(new-schedule)
    Scheduler->>Formatter: export(file4)
    Scheduler->>Scheduler: file4 = path+ugrad.xls+x
  
```

The diagram illustrates the sequence of operations in a course recommendation system. It starts with a **User** who triggers the process by downloading `courses_needed.pd5`. This leads to the **File-Getter** stage, which provides three input files (`file1`, `file2`, `file3`) to the **File-Parser/Converter**. `file1` is converted into a `needed_courses_dictionary/set`, `file2` into a `DAG-parser(file2)` and a `DAG graph`, and `file3` into an `offering-parser (file3)` and an `offering-selection/set`. These processed inputs are then sent to the **Scheduler**. The scheduler uses an `AI buffer` and performs a `scheduler(needed_courses, DAG, offering)` to generate a `new-schedule`. Finally, the **Formatter** stage takes the new schedule and performs `format(new-schedule)` and `export(file4)`, resulting in `file4 = path+ugrad.xls+x`. The diagram also includes performance evaluation steps at the bottom: `PE.performance_eval()` and `PE.performance_eval()`.

3. [30 points] Develop a **data flow diagram** for our group project - A Smart Class Planning Tool.



4. [30 points] Develop a **program structure diagram** based on your data flow diagram.



Submission:

- Submit your solution as a PDF file named as "hw_modeling.pdf" through Cougar view