EXPERIMENT NO. 07

Implementation of data flow design pattern

Aim: Application & Analysis of data flow design patterns in the case study

Description : The aim of performing this experiment is to implement a set of particular design patterns in your project and show how your project adapts to that particular design pattern and show the changes that have been achieved by applying that particular design pattern to your project.

Design patterns are well-proved solution for solving the specific problem/task.

Data Flow Style

- Has the goal of modifiability
- Characterized by viewing the system as a series of transformations on successive pieces of input data
- Data enters the system and then flows through the components one at a time until they are assigned to output or a data store
- Batch sequential style
  - The processing steps are independent components
  - Each step runs to completion before the next step begins

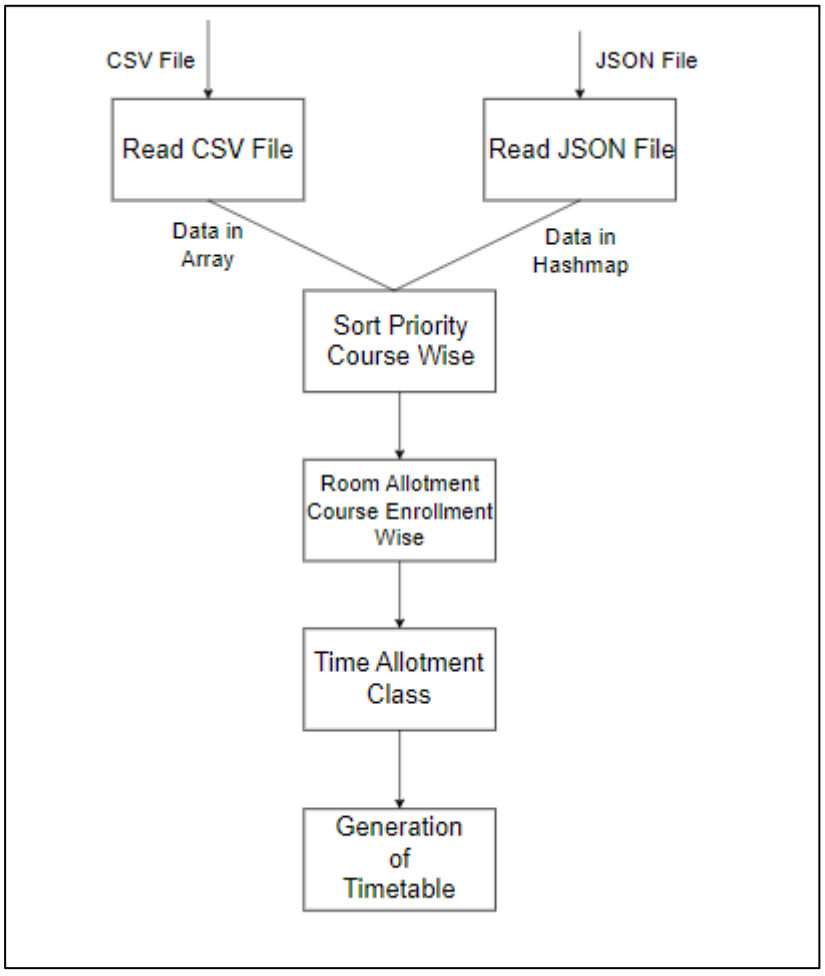
Pipe-and-filter style

  - Emphasizes the incremental transformation of data by successive components
  - The filters incrementally transform the data (entering and exiting via streams)
  - The filters use little contextual information and retain no state between instantiations
  - The pipes are stateless and simply exist to move data between filters
- Advantages
  - Has a simplistic design in which the components interact with the environment
  - Consists of no more and no less than the construction of its parts
  - Simplifies reuse and maintenance
  - Is easily made into a parallel or distributed execution in order to enhance system performance
- Disadvantages

- Implicitly encourages a batch mentality so interactive applications are difficult to create in this style
- Ordering of filters can be difficult to maintain so the filters cannot cooperatively interact to solve a problem
- Exhibits poor performance
    - Filters typically force the least common denominator of data representation (usually ASCII stream)
    - Filter may need unlimited buffers if they cannot start producing output until they receive all of the input

Each filter operates as a separate process or procedure call, thus incurring overhead in set-up and take-down time

Architecture Diagram:



Conclusion:

The architecture diagram represents the patterns implemented in the Course Scheduling System.