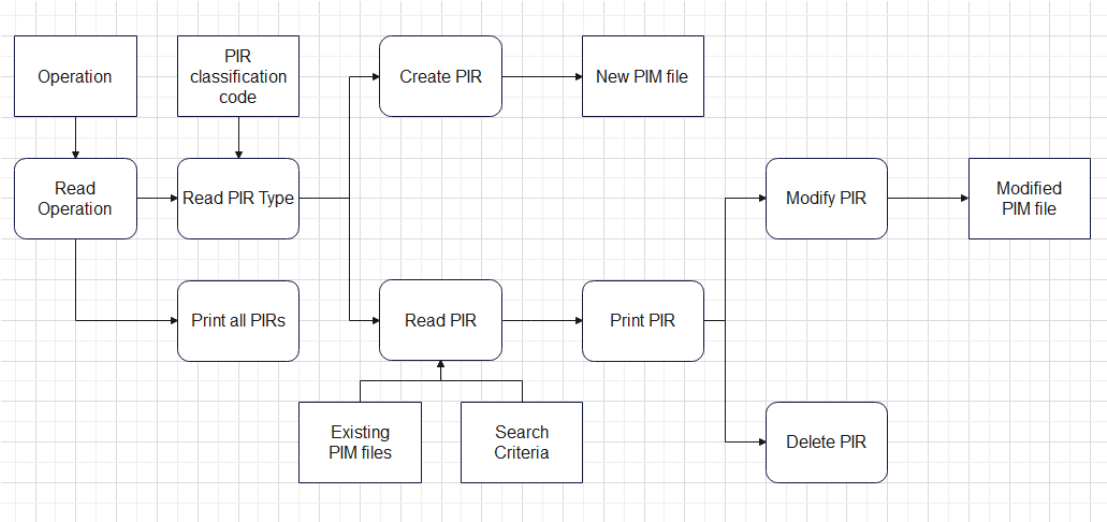
System architecture

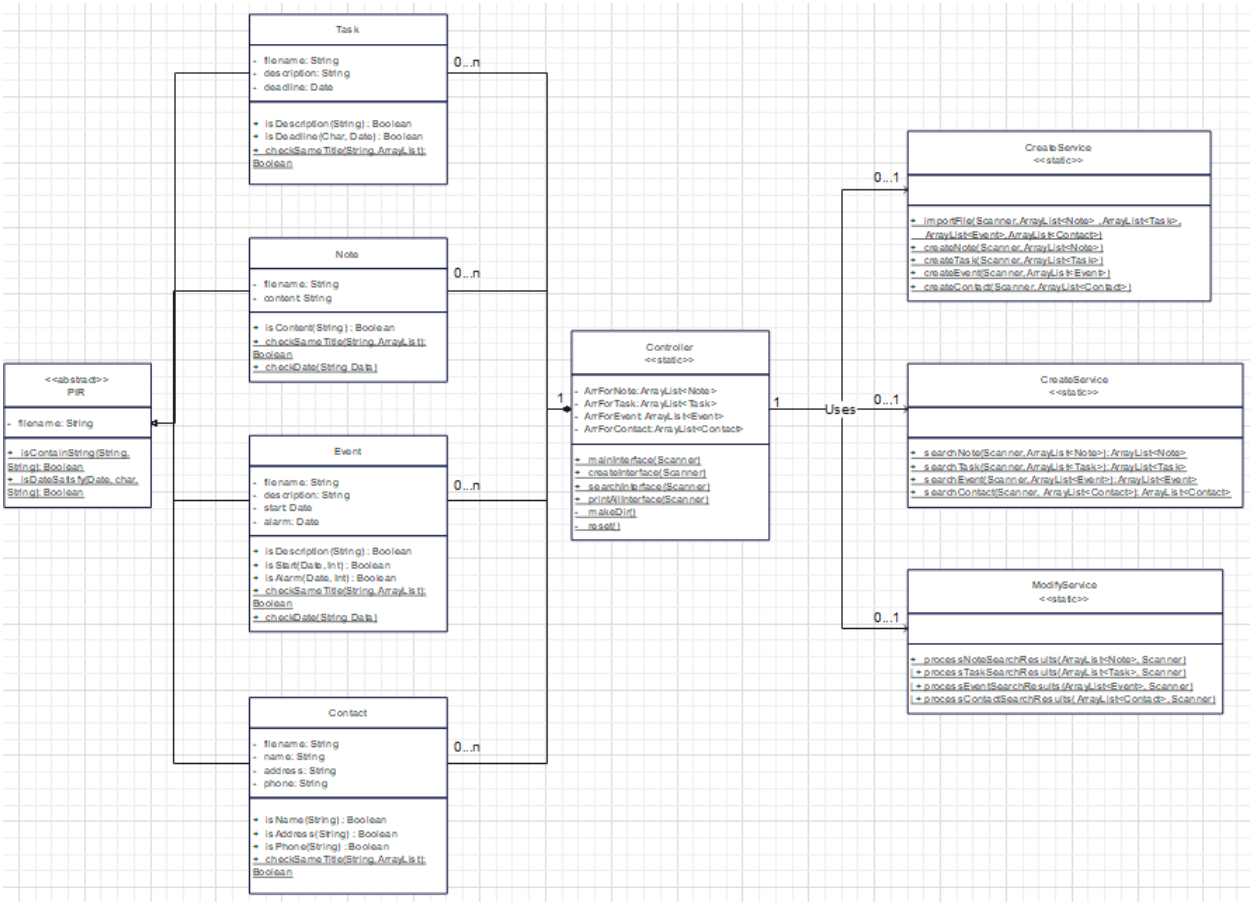


The PIM system adopts the pipe-and-filter architecture, ideal for transaction-based data processing common in PIM software.

The choice is motivated by its developer-friendly nature, offering simplicity and clarity in design, facilitating effective development and maintenance.

Additionally, the architecture aligns with the PIM system's characteristic of limited user interactions, providing a structured approach to manage such interactions efficiently.

Structure of and relationship among major code components



The system is composed of nine classes, excluding the Main Class used for initialization. These classes are grouped into three main clusters: PIR classes, Service classes, and Controller. PIR classes are dedicated to representing records within the PIM system, playing a fundamental role in managing information.

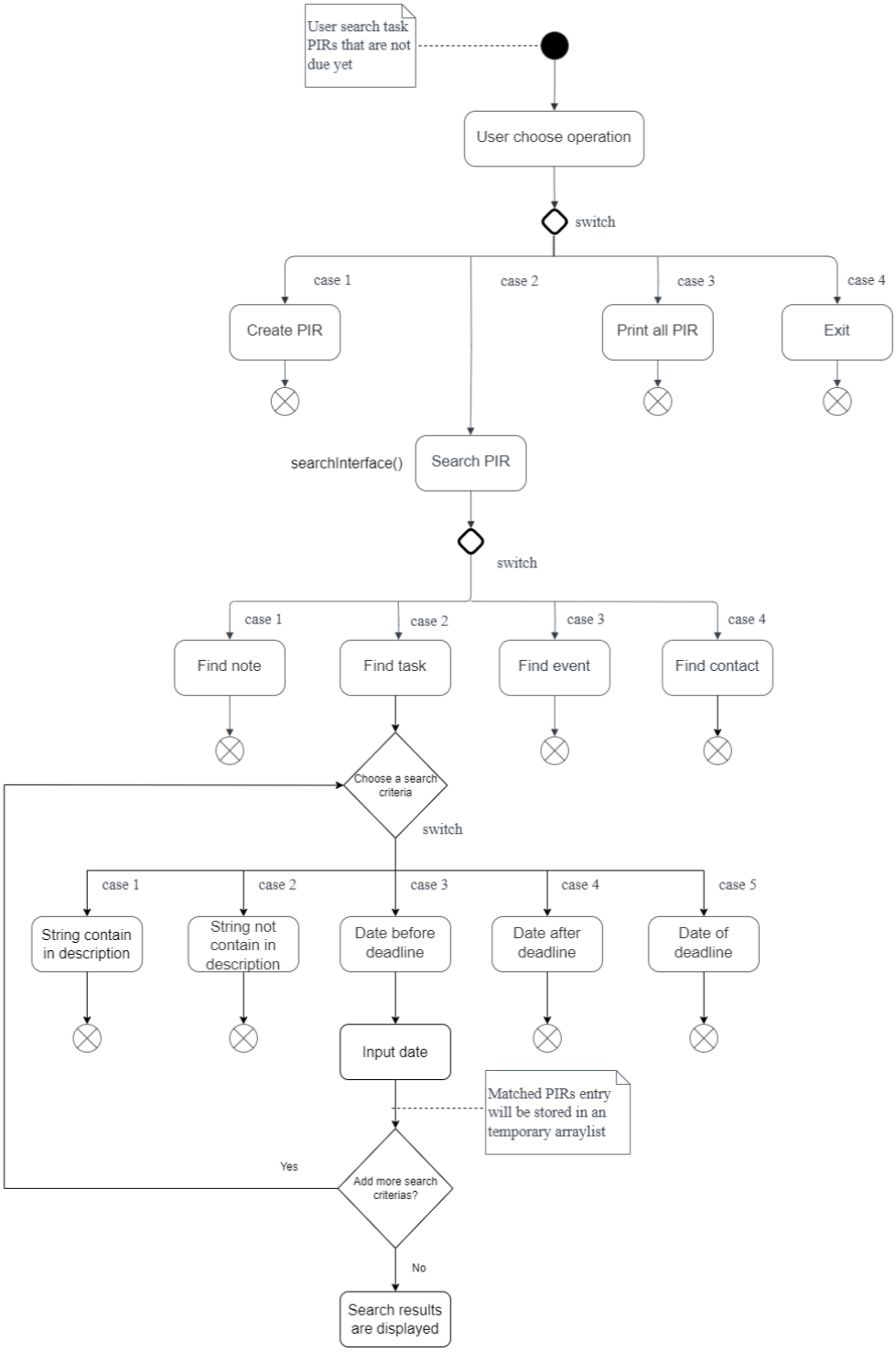
The Controller class acts as the procedural controller of the system, guiding users in their interactions with the PIM functionality. This central role positions the Controller as a key orchestrator of the overall user experience and system operations.

Service classes, forming another cluster, provide specific operations to the PIRs. Notably, these classes are static, eliminating the need for instances and emphasizing their focused and utility-driven nature.

Of significance is the Delete operation, specifically mentioned for its integration with the modifyService. This integration is highlighted, likely due to the simplicity and efficiency achieved by combining these operations. The explicit attention to the Delete operation

underscores its importance in the overall functionality of the system. In summary, the system architecture is organized around distinct class clusters, each serving a specific purpose in managing and interacting with the PIM system.

Example use


User search task PIRs that are not due yet
User choose operation
switch
case 1
case 2
case 3
case 4
Create PIR
Print all PIR
Exit
searchInterface()
Search PIR
switch
case 1
case 2
case 3
case 4
Find note
Find task
Find event
Find contact
Choose a search criteria
switch
case 1
case 2
case 3
case 4
case 5
String contain in description
String not contain in description
Date before deadline
Date after deadline
Date of deadline
Input date
Matched PIRs entry will be stored in an temporary arraylist
Yes
Add more search criterias?
No
Search results are displayed

Assume the user wants to find all task PIRs that are not due. After the user initializes the system, they will choose option 2, search PIRs. Then, the user will be prompted to choose the type of PIRs they want to find. Since the user is finding task PIRs, they will choose option 2, “find task.” As the user searches for task PIRs that have not passed the deadline, they will choose option 3, “date before deadline.“ The user will then input the date. The search results will be displayed after the matched PIR entries are stored in a temporary array list, and the user has no additional search criteria.