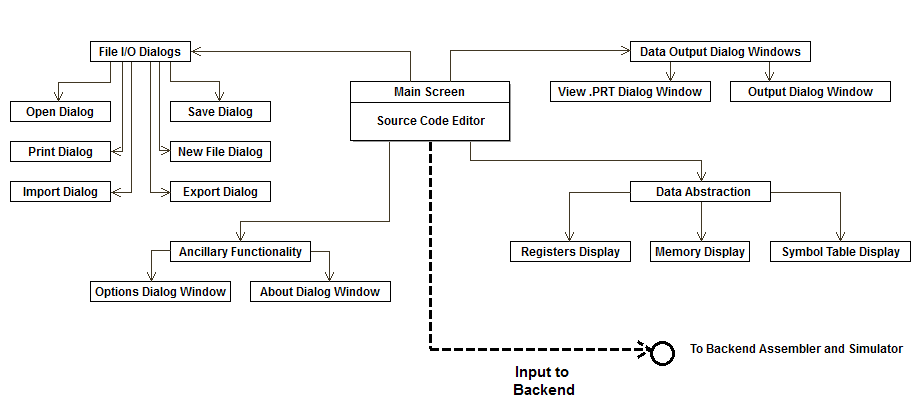
**Architectural Design**

The ASSIST/UNA software has been divided into two portions: Frontend and Backend. The Frontend comprises the ASSIST/UNA graphical user interface and all associated dialogs and components. The Backend comprises the Assembler and Simulator portions and associated functionality. The exact interfacing between the Frontend and Backend will be determined in the Detailed Design.

**Frontend Organization**

The root of the Frontend design is the Main Screen, which contains the Source Code Editor. All of the required functionality is achieved by additional classes. All file input and output will be handled by the native Windows input and output dialog windows (e.g., Save). The visual components that relay information from the Backend have been organized accordingly. Data Abstraction consists of the components that relay the registers, memory, and symbol table content from the Backend Simulator. The Data Output Dialog Windows consists of the View .PRT Dialog Window and the Output Dialog Window, which display the .PRT file and simulated executed program output from the Backend Assembler and Simulator. The Ancillary Functionality consists of the Options Dialog Window and the About Dialog Window. The Options Dialog Window will allow the user to modify the Backend Assembler options and additional ASSIST/UNA options. Data inputs from the Frontend will be transferred through the Main Screen to the Backend.

Figure 1: The initial Frontend architectural organization.



**Backend Data Flow**

The Backend has been divided primarily into the Assembler, Simulator, and the Error Detection modules. The Assembler will translate the input source code into object code and partially generate the .PRT file. Assembly errors will be reported to the Error Detection module during the second pass of assembly. The Simulator will simulate execution on the IBM/360 mainframe, and it will partially generate the .PRT file. Runtime errors will be reported to the Error Detection module. Errors, program output, and the .PRT file will be relayed to the Frontend.

Figure 2: The initial Backend architectural data flow.

