Our architecture comprises 4 main components: Graphical User Interface (GUI), Logic, Parser, Storage. All tasks/events entered by the user are saved as Event objects which may be passed between components (except GUI).

**GUI:** Users interact only with the GUI mainly in the form of text input, though there are certain buttons that provide alternatives to text input. The GUI captures text input and conveys it to Logic, and obtains all required output from Logic as well.

**Logic:** This acts as a facade shielding the more complex implementations from the GUI. It comprises the Logic class which interacts with Parser and Storage to implement the actual user command, and the Display class which stores all output required to be displayed by the GUI.

**Parser:** Parser translates raw user input into commands intelligible by the program by converting inputs into Event objects. This is done by passing inputs through classes ParserProcessor and InputStringSplit. Generally, Parser is able to track an Event and a command (which are part of its attributes). Logic gets this information through Parser's API, then uses it to call the APIs from Storage to execute the desired command.

**Storage:** In Storage, the class EventStorage keeps track of a vector of Events that have been entered by the user but not deleted. All actual adding, deleting and editing of Events is implemented by manipulating this vector. The information in the vector is written to a text file after every command is executed, and read from the text file whenever the program is started. The EventArchive class stores similar information but with additional information for command type. This is to faciliate undo operations. Finally, the Search class implements various search functions required by EventStorage to find certain Events.