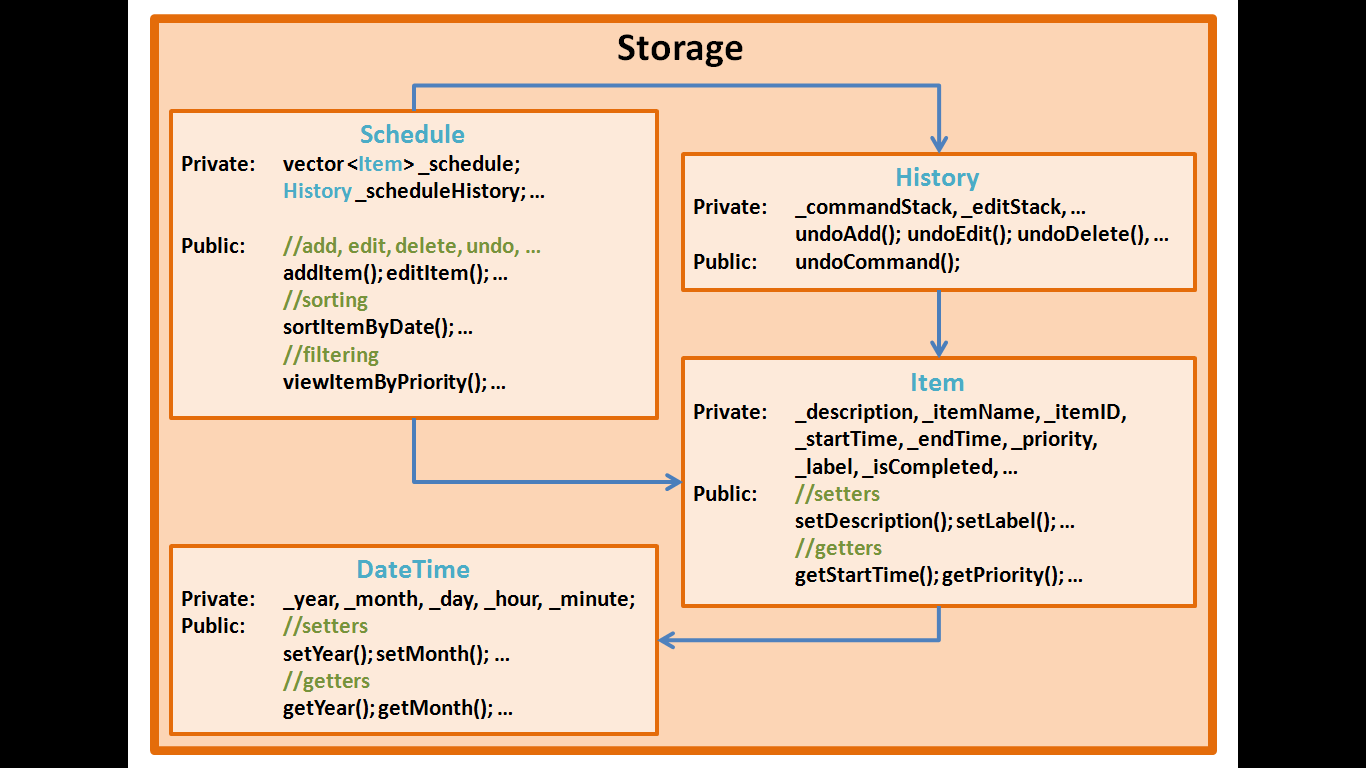
## Storage



Storage is currently implemented in the most basic and straightforward method – declaring and calling objects of a certain class. Each of the sub-components – schedule, history, item, and dateTime, has the basic setters (used to assign values during *add* and *edit* functions) and getters (used extensively in *display* and *edit* functions). Schedule is slightly more functional with certain types of sorting implemented.

However, moving forward, it is important to implement storage as via pointers to objects, and in order to implement OO fully, logic will only pass values/pointers/objects to Schedule, and will not directly access the Item or dateTime objects via the getters (pass by reference). Local copies will be passed via the getter methods.

## Tests

For the Storage component, test-driven development has been employed in most parts (exception to Schedule). Current tests are fairly comprehensive, although more efficient methods should be written in future to better compare between class objects (overload of Boolean == and !=, for example). This would allow better and less time-consuming usage of *Assert::AreEqual (... , ...)*.