**How Singleton design pattern improves the design of GetSetGo:**

In the application we have written a logger class. It should register all the actions performed by recording the date and time details and the action details which are performed in a single file.

For example, every time a new attendee is registered in the event a log had to be created with the details of the user including the time and date of registration. In a similar way, whenever there is an update about to event description or details of an event, these changes had to be logged in a single file.

If we use more than one logger, the logfile is total chaos. By creating more than one instance all the loggers’ conflict within each other. If there is more than one client application trying to use the same utility logger, it might lead to creating of multiple instances of this class and it can cause issues during concurrent access to the same logger file.

We can make use of the logger class as a singleton class and provide a global point of reference so that all the users can make use of the logger class and there would be not more than one user who will access logfile at the same time.