# EMS Technical Requirements and Required Functionality

In this section, I will endeavor to describe the requirements and desired behaviour of the EMS Solution that you and your team will be developing. This description will be done in point form below (so if you need any clarification – **ask!**) For a complete description of what makes a valid field and what makes a valid employee – see **Appendix B** at the end of this otherwise already long description.

* The solution will include another class library (call it Supporting) consisting of the FileIO class and Logging class
* The *Supporting* class library will contain :

## The FileIO class

* + - This class contains the methods necessary to open (for reading or writing) and close Employee DBase files.
    - This class will also contain methods to read a database record and to write a database record. As you can see in the DBase example file (below), there is one database record for each Employee object.
    - You may also wish to create methods to
      * Not only read but also parse the records within a file – parsing involves the field value extraction from the pipe-delimited CSV record. You must architect your EMS design such that the FileIO class and methods are used to read and write the employee object information to and from the database file – but the FileIO class itself does not create, modify or *know* about the Employee class itself
        + be aware that it is possible that records contained within the file may contain data about an employee which is not valid or complete and therefore an the resultant object would either not be created or would be left incomplete and invalid
        + in any case, once a record is read and parsed from the incoming file, the Logging class is used to record this event – this logging event would indicate if the record just parsed generated a valid employee-type or not
      * create an Employee file record (in the correct format) and write out the record to the DBase file
      * for a sample EMS DBase file and its required format see [this sample DBase file](http://www.conestogac.on.ca/~set/courses/year2/sq-i/assignments/Project-Sample-EMS-DBase-File.txt)
    - please note that the root location of all DBase files is a subdirectory called “DBase” found in the same directory / location as the executable
      * *while developing and debugging your application within Visual Studio, this means that the “DBase” directory needs to be in the same directory as the source code*
    - the following types of events are to be logged from this class:
      * opening and reading/parsing of a file
        + include a summary event when this is done indicating the total number of records read, total number valid and total number invalid
      * opening and writing of a file
        + include a summary event when this is done indicating the total number of records written, total number valid and total number invalid
      * any fileIO related errors

## The Logging class

* + - This class is called upon in order to audit (or track) the usage , status and outcomes of many of the other class libraries methods
      * Having a Logging class will prove to be your lifeline / saving grace and in the work place, thinking of auditing user actions sets you apart from other developers – especially if you log events intelligently and accurately
      * The use of a logging class is quite common in more real-world applications and is quite an invaluable tool in performing support of an application (during integration and system testing as well as customer support) – it logs significant events and results. The purpose and goal of any entry made in the logging file is to help recreate (and trace) the sequence of events that occurred and lead to a certain point in processing … a customer reported error perhaps …
    - This class will contain methods able to open (for appending) the log file for that day, write out a logging event in the correct format and close the log file
      * Things for you to consider in this class … when should the log file be opened? When should the log file be closed? Before you answer remember that the logging class will be called not only in a standalone testing program, but will be called upon during your Unit Testing. These methods will be called by many other methods in the other classes.
      * Depending on how you handle your logging file you may or may not be able to look at its contents while you are running the application.
      * The logging filename will be in the format : **ems.YYYY-MM-DD.log** (where YYYY-MM-DD is the day that the logging file was created)
    - Each logging event will be contain the following fields :
      * Event Timestamp in the format YYYY-MM-DD hh:mm:ss, eg. “2010-09-01 07:02:00”
      * [ClassName.MethodName] logging the event – eg. “[FulltimeEmployee.Validate]”
      * Event details
      * So an example might be :

2010-09-01 07:02:00 [FulltimeEmployee.Validate] Employee - Clarke,Sean (333 333 333) - INVALID