# **Process**

The process I took to complete this task was as follows:

I began this program by opening NetBeans and creating a new package and in it creating my first page named “MainApp” in this java class I created a menu that prints to screen a list of options the user can select from, to allow my program know what the user inputs I imported a Scanner which will listen to the keyboard for keystrokes. I put the menu code into a “do loop” because I want it to keep being printed to screen as long as the user is selecting options, I used a “do loop” because no matter what my menu will be printed and then after java will check for a condition whether to continue or not. The only option that will make the program stop is 5 as this is the exit option for the user and the break for the program the code is “while (opt != 5);” this code basically lets the loop keep going as long as “opt” the variable that contains the user input does not contain 5. To get each “opt” input to trigger specific methods I will write later I use a “switch” statement.

I then needed to connect to the Database but first I need to create an “Event” class so that when I read a row from my Database I can store it in my Event object. I made sure the attributes for my Event class matched the one in my Database exactly. This class is basically a blueprint of what makes up an event it has all the attributes of an event and the constructor so I can make more events from this blueprint this class also contains the get and sets I will need later on; other than that it’s a basic class to code.

The next step in connecting to my Database I had to create a class to connect to my Database I called this “DBConnection” in here I use a singleton pattern and the definition of a singleton patter is: *“In*[*software engineering*](http://en.wikipedia.org/wiki/Software_engineering)*, the singleton pattern is a*[*design pattern*](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))*that restricts the*[*instantiation*](http://en.wikipedia.org/wiki/Instantiation_(computer_science))*of a class to one*[*object*](http://en.wikipedia.org/wiki/Object-oriented_programming)*. This is useful when exactly one object is needed to coordinate actions across the system. The concept is sometimes generalized to systems that operate more efficiently when only one object exists, or that restrict the instantiation to a certain number of objects.”* basically what this means and why I need it is because I need one and only one connection to my database if I have more this can cause me some problems, in the singleton pattern I have a method called “getInstance” and what this does is creates a connection object from the “java.sql.Connection” package for me and that “connection” object will represent my connection to the database, if anything may go wrong I have an SQl exception that will load the drivers from the the “MySQL” server and again if that fails there is another precaution the ClassNotFoundException is thrown.