The sequence diagram for this software has shown in two separate diagrams. The first one depicts an overall view of how the software works. It first start by receiving the argument from the user which is the pathname of a directory or JAR file and check if the arguments is valid. Then, it calls an instance of JavaRetriever class which has the responsibility of iteration through the directory or JAR files recursively and reading all Java files. After that, it enters into a loop which creates ASTParser for each Java files, sets the configuration and create AST as CompilationUnit. Following from last step, it creates a visitor to visit each node of the AST and count the number of declaration and references for each node and finally, print the result for all types on the console. The second diagram show in more detail how the software recursively goes through the directory or JAR files of given pathname. It checks the pathname of interest to find its type, and cording to that, if it’s a directory, a JAR file or a Java file takes the appropriate action. We used abstraction and we did not show all the details in the diagram specifically the steps that it takes to read the content of files to make easier for the client to understand the functionality of the software.