Clean-code gating and reporting

INTRODUCTION:

The Gating web-service must measure a configured code-metric on a given github repository. The service uses simian and tics tools. When the code-metric confirms to a configured ‘gate’, it will respond with a ‘go’. Otherwise, the response will be ‘no-go’.

Gates need to provide capabilities to check on:

* Thresholds or limits that can be supplied for a given repository.

**Note:** One should install simian and tics tools on their device to run this project.

GUIDE TO UNDERSTAND

-Simian tool is used to detect the duplicates in the source code.

-Tics tool is used to detect the errors and warnings.

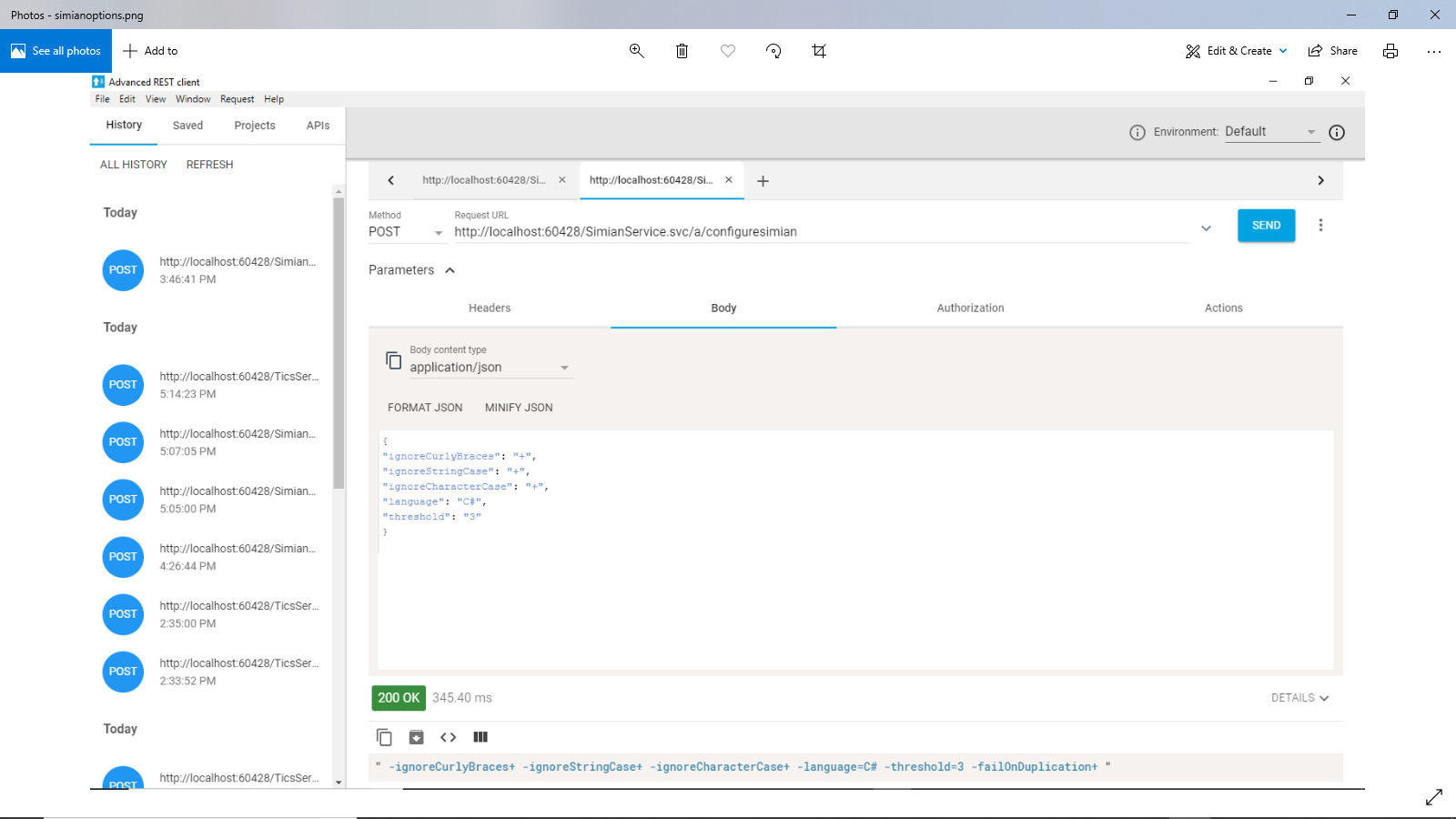
There are two web services in this project.

1. Ticsservice which takes input of GitHub repository Url and Error threshold based on which gating is done.
2. Simian service which takes input of GitHub repository Url and duplicate threshold based on which gating is done.

Following are the steps involved in the complete workflow:-

* The first step to run simian service is to set configuration of the simian tool which can be taken from user
* This can be done at <https://........./configuresimian>
* Which takes input of class simianOptions with properties:
* ignoreCurlyBraces
* ignoreCharacterCase
* threshold
* language

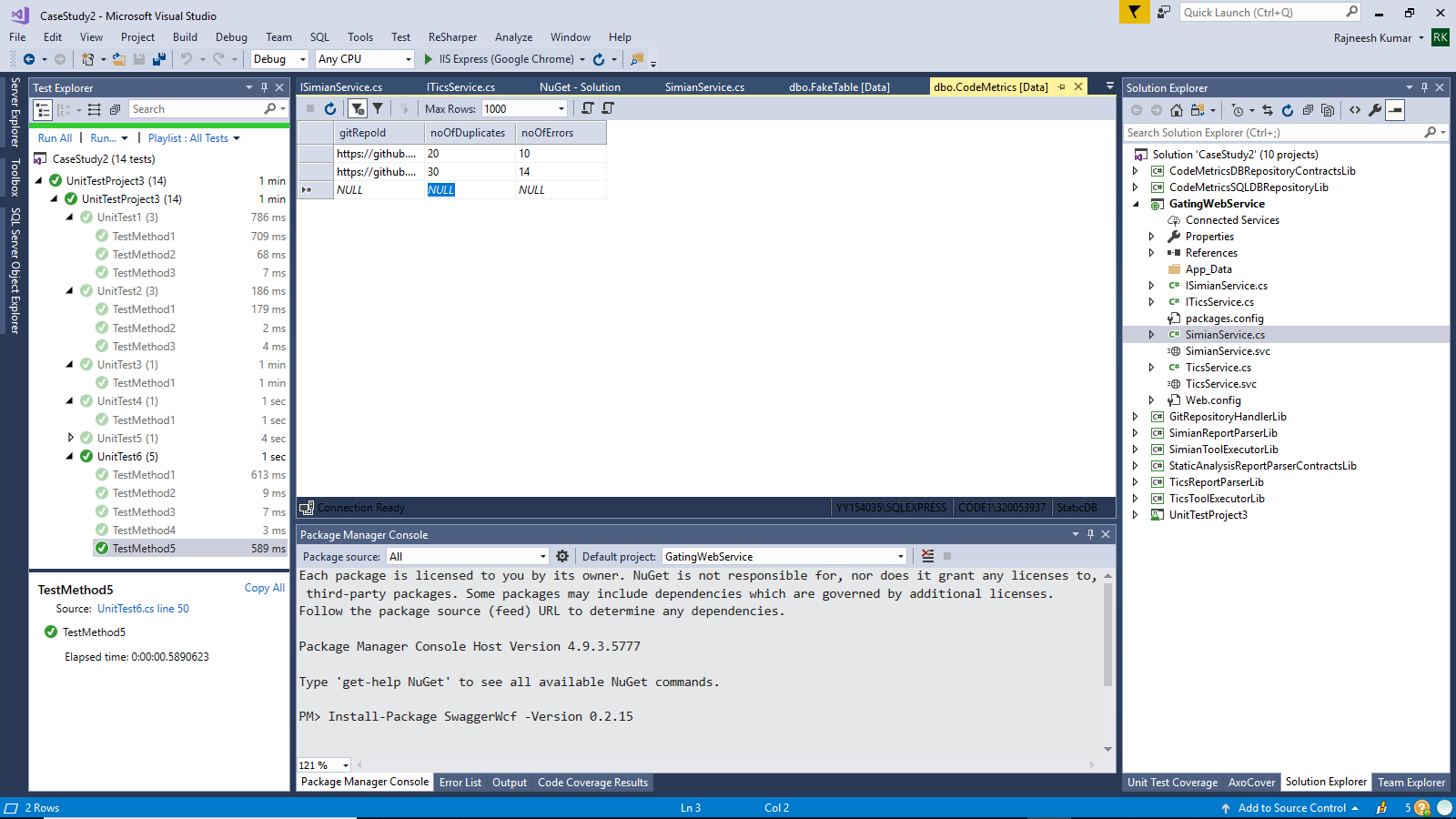
After setting up the configuration one can gate their repository at <https://.........../simianservice>



The details of Errors and Duplicates are maintained in a sql database called “StaticDB” which contains a table called “CodeMetrics”.

The codemetrics has the following schema

Varchar gitRepoId (Primary Key), int noOfDuplicates, int noOfErrors.



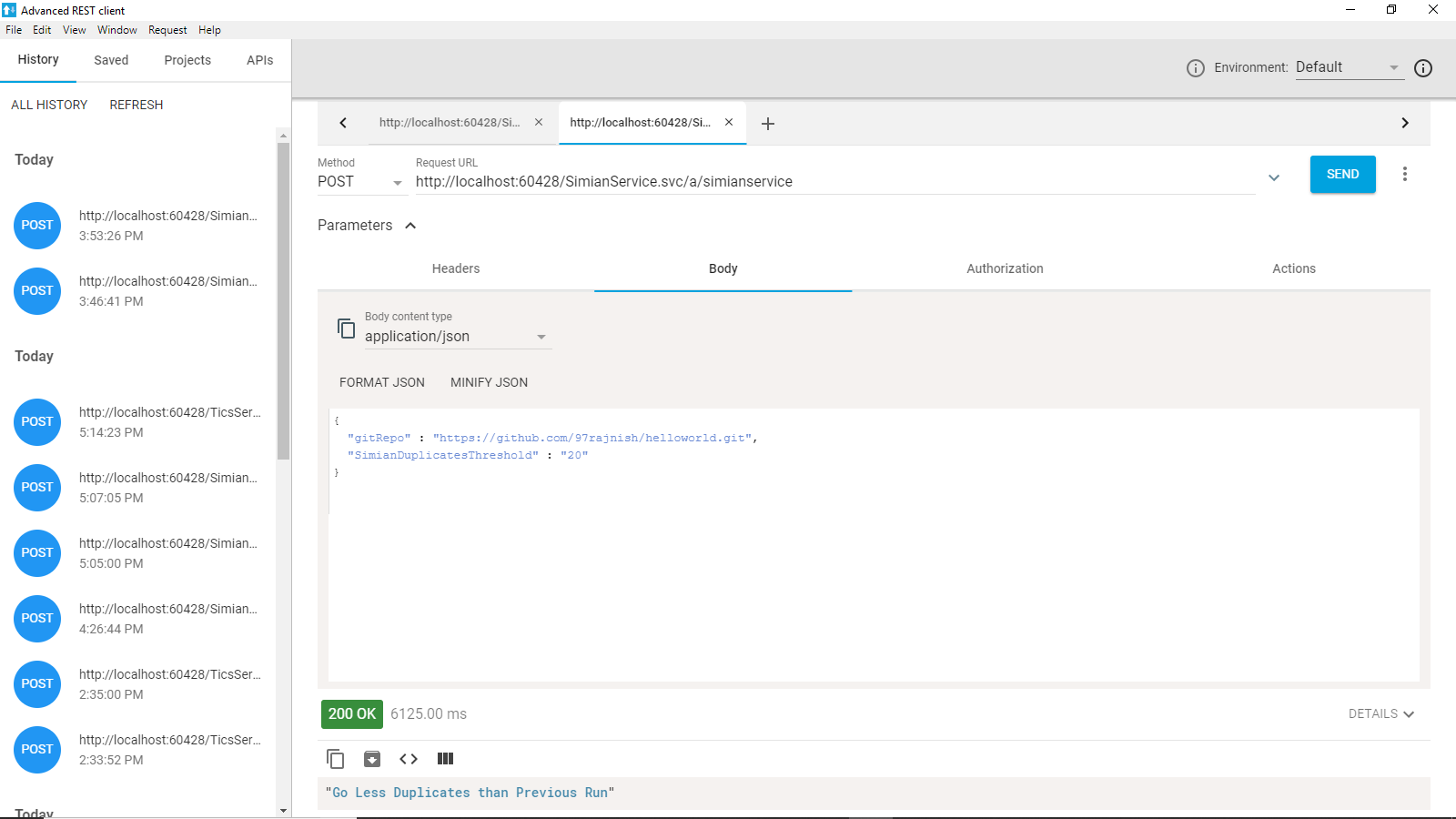
Flow of the Simian Service:

GitRepositoryHandler class clones the input github repository to the folder C:\Temp\CaseStudy1.

SimianToolExecutor class runs simian tool on the cloned repository and generates a report at C:\Users\320053937\Documents\SimianReport.txt.

SimianReportParser class parses the report and gets the total duplicates and also checks it with the previous value stored in the database and if the current value is less than the previous value then updates it to the database with the help of CodeMetricsSqldbRepository class.

Then based on the Exitcode and duplicates threshold provided by the user, the final result Go or No-Go will be displayed along with less or more duplicates from the previous run.



Flow of the Tics Service:

GitRepositoryHandler class clones the input github repository to the folder C:\Temp\CaseStudy1.

TicsToolExecutor class runs Tics tool on the cloned repository and generates a report at C:\Users\320053937\Documents\TicsReport.txt.

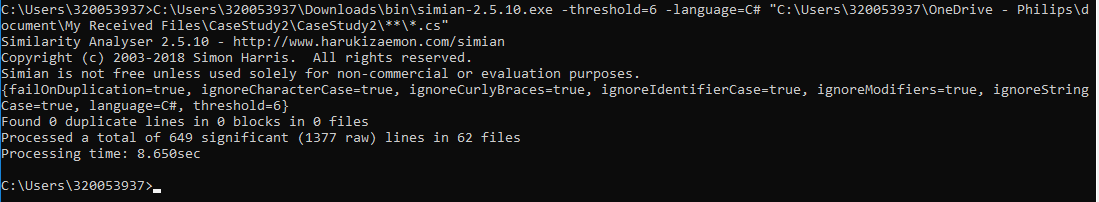
TicsReportParser class parses the report and gets the total errors and also checks it with the previous value stored in the database and if the current value is less than the previous value then updates it to the database with the help of CodeMetricsSqldbRepository class.

Then based on the errors threshold provided by the user, the final result Go or No-Go will be displayed along with less or more errors from the previous run.

# Code Quality Evidence

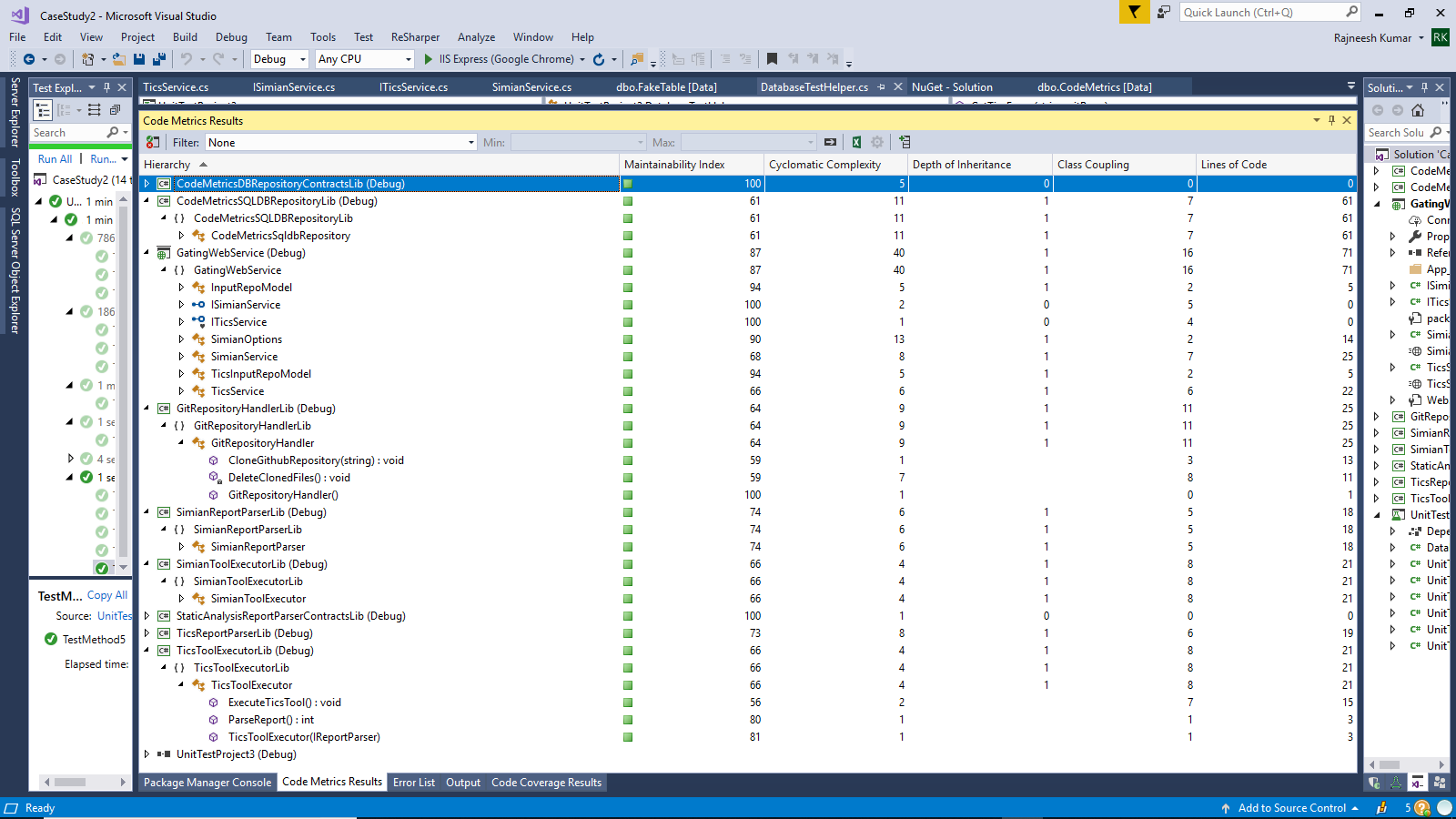
## Duplication

We have used the simian tool to analyze the duplication. We have a 0% duplication for a threshold value of 6.

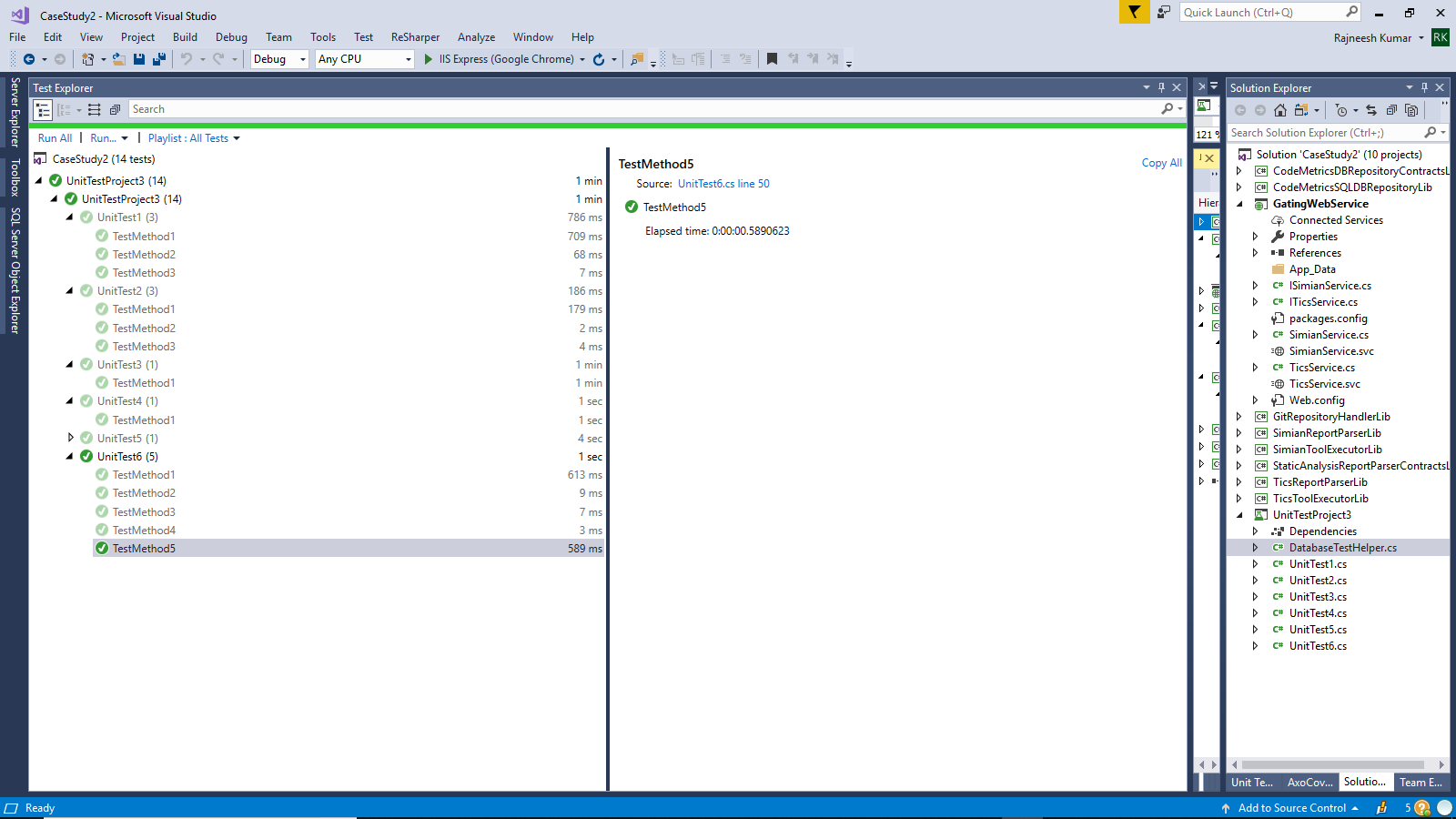


## Cyclomatic Complexity

Delete cloned files method has a complexity of 7 which can be reduced. The maximum cyclomatic complexity that we have set is 3 excluding delete cloned files method.



## Unit Test Report



## Unit Test Coverage Report

64 Percent of the code has been covered through the test cases.

CodemetricSqldbRepository class is the class which has all database related operations. It is not covered as it is not recommended to give the original database to the test environment. So we have created a duplicate database class which implements the interface of CodemetricSqldbRepository.

