**Test Execution Document**

Label Printing and Barcode Generator

19th November,2016

Software Engineering Project

NIIT University

Computer Science Engineering

**Course**

CS301 Software Engineering Project

**Project Group**

Priyanshu Bhatnagar

Rashika Bhargava

Nirali Sampat

Simran Malhotra

**Customer**

Advanced Software

**Project Instructor**

Mr. Amit Kumar

**1. Introduction**

This is the Test Execution Document of the Label Printing and Barcode Generator Software Engineering project at the Computer Science Department of the NIIT University. The Label Printing and Barcode Generator project was given to us by the company Advanced Software based in Surat, Gujarat. Its main goal is to provide an easy to use software for making labels and generating barcodes as well as printing them for various industries like the textile industry, footwear industry, fresh foods industry, examination system, etc on the Universal Windows Platform.This document presents the results of the execution of the different tests presented in the Label Printing and Barcode Generator Test plan document [1]. The document is organized as follows:

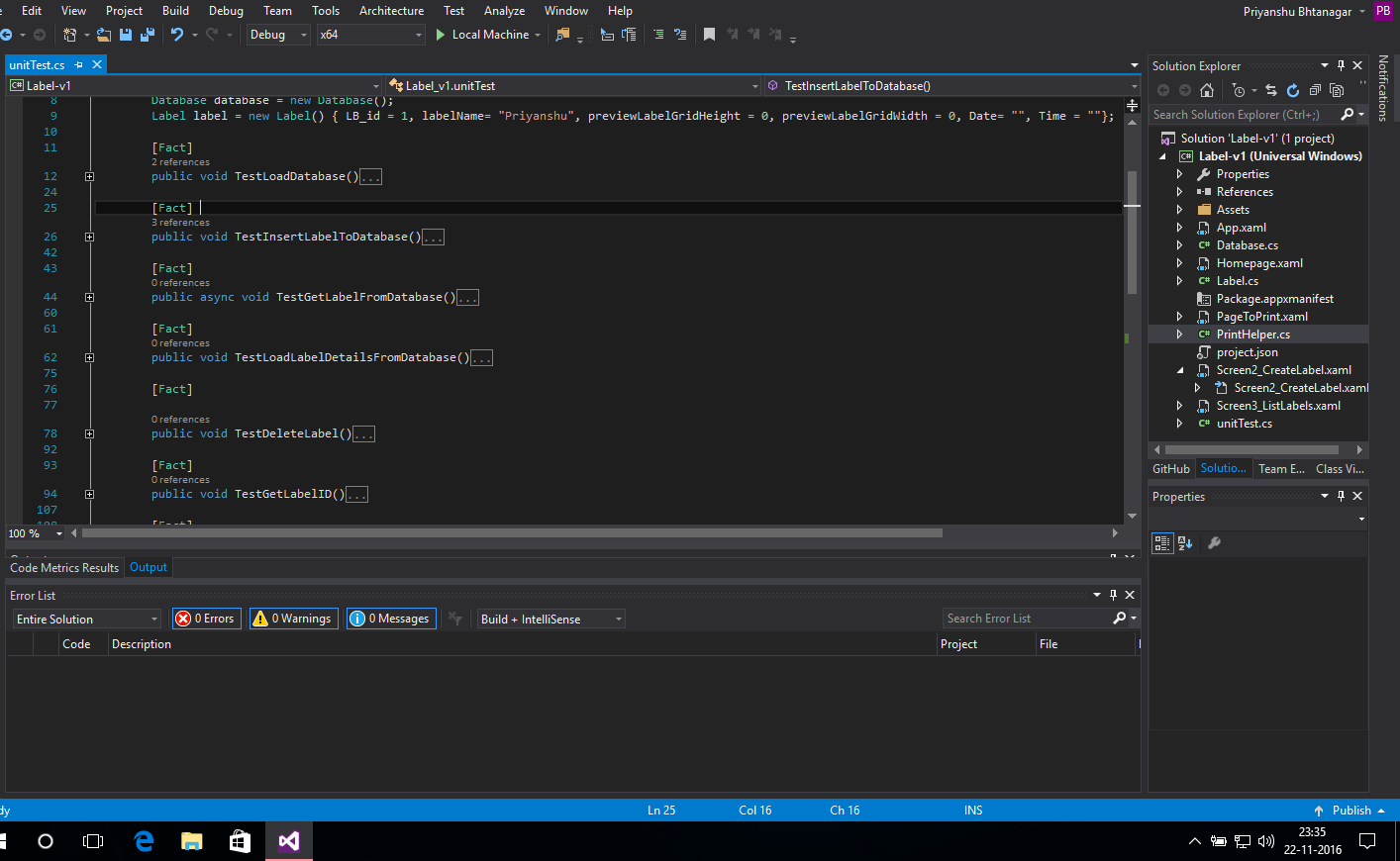
* Section 2 introduces the results of unit testing.
* Section 3 introduces the results of integration testing.
* Section 4 introduces the results of system testing.
* Section 5 introduces the Bugs and Fixes.

The results of the test execution will be presented in subsection as follows: Id and name of the executed test. Description of the test, or collected results. Errors found if any, and maybe a reference to the bug and fixes report. If error was found on certain step of Test Case, the step number is given. All test cases were first executed once and found bugs were fixed. Then all test cases were executed again and no further bugs were found.

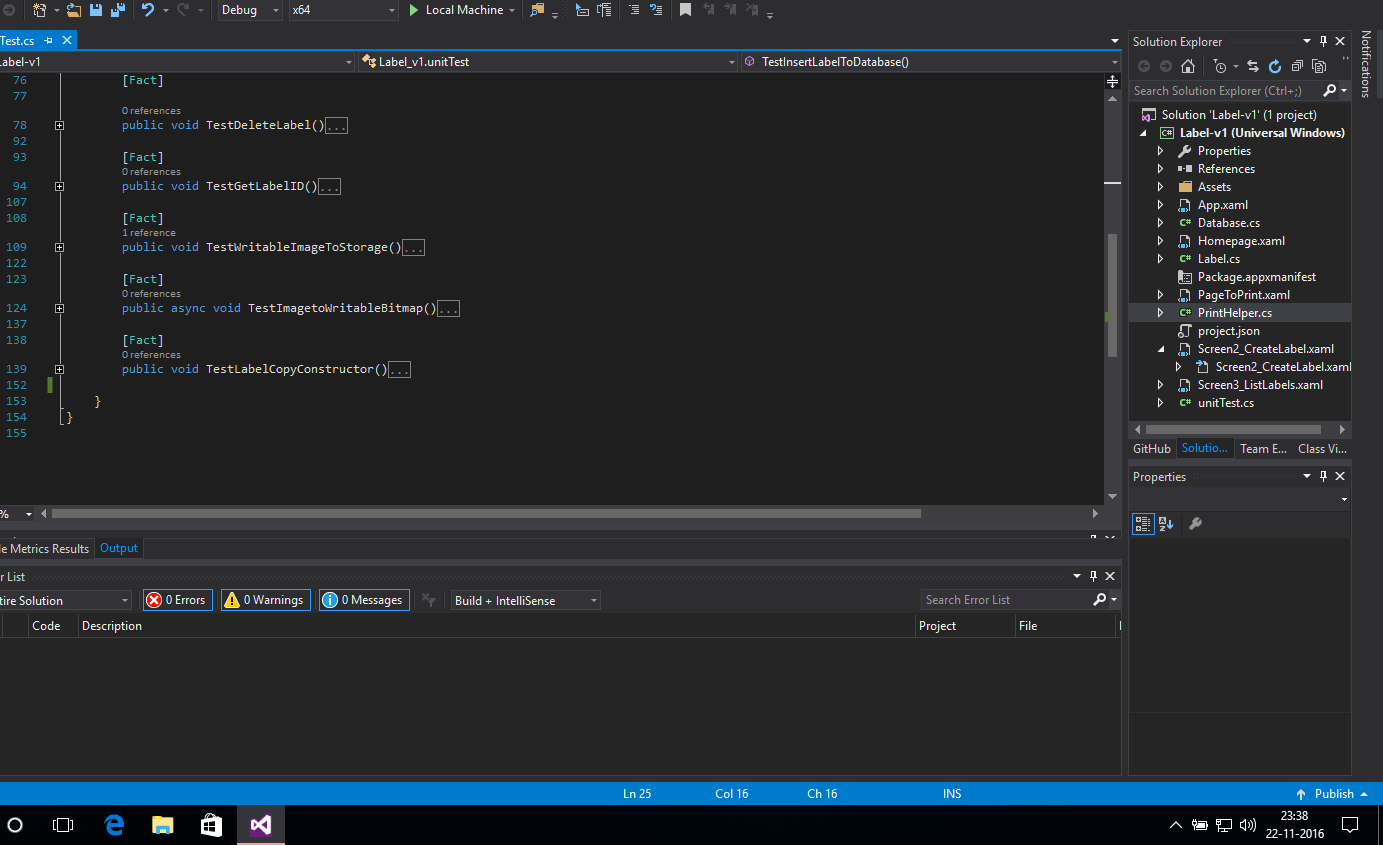
Test were run on following environment: 1.6 GHz Intel Core i5 processor, 8 GB 1600 MHz DDR3 memory, Intel HD Graphics 6000 1536 MB graphics memory, Macintosh HD SSD, Windows 10 VM OS.

**2. Unit Testing**

Unit test were created as xUnit tests and with the help of Visual Studio debugger. Unit test were executed in Visual Studio 2015 Enterprise. Not all classes were tested as it was unnecessary to run test on them. List of Unit tests executed are mentioned below:

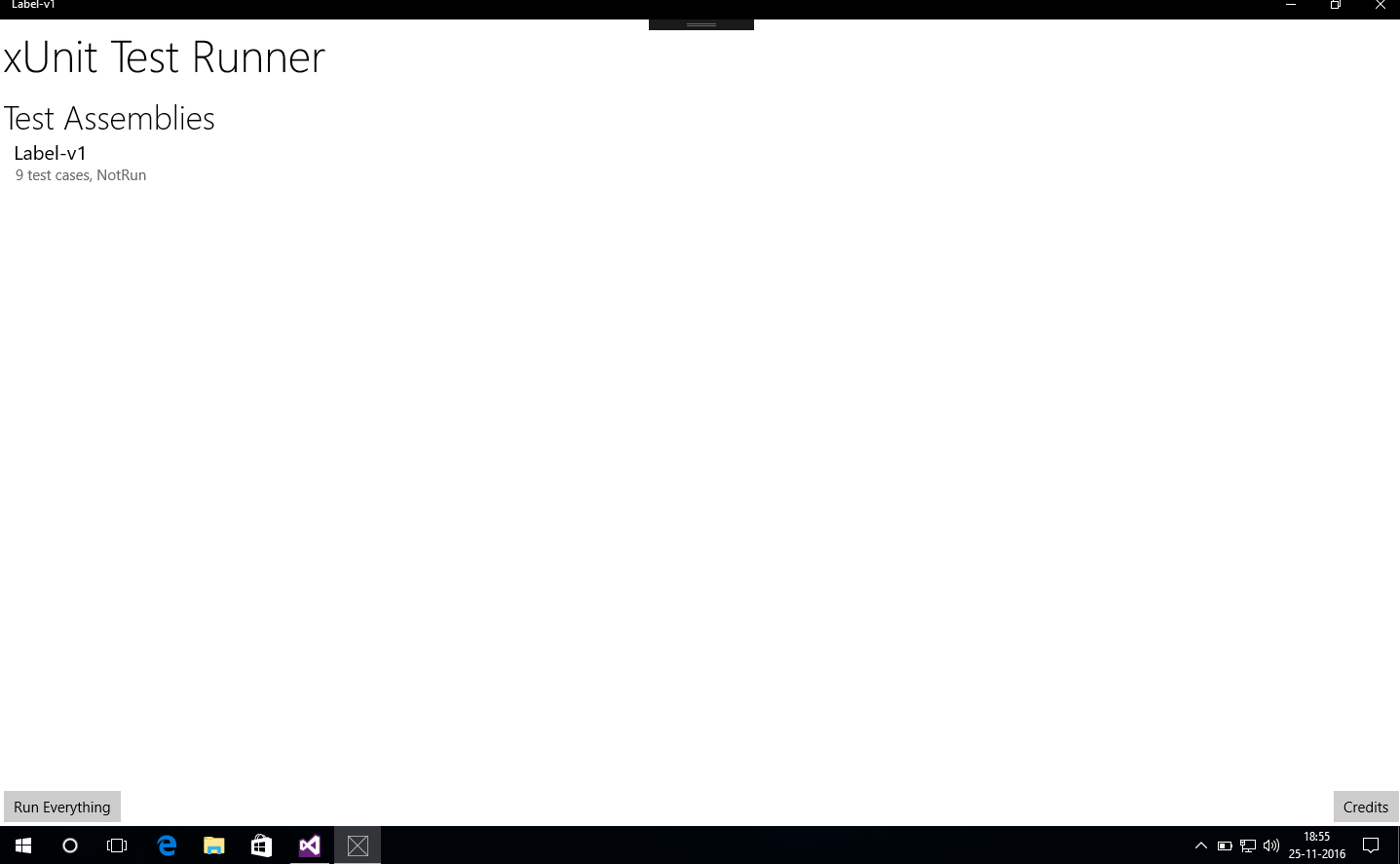


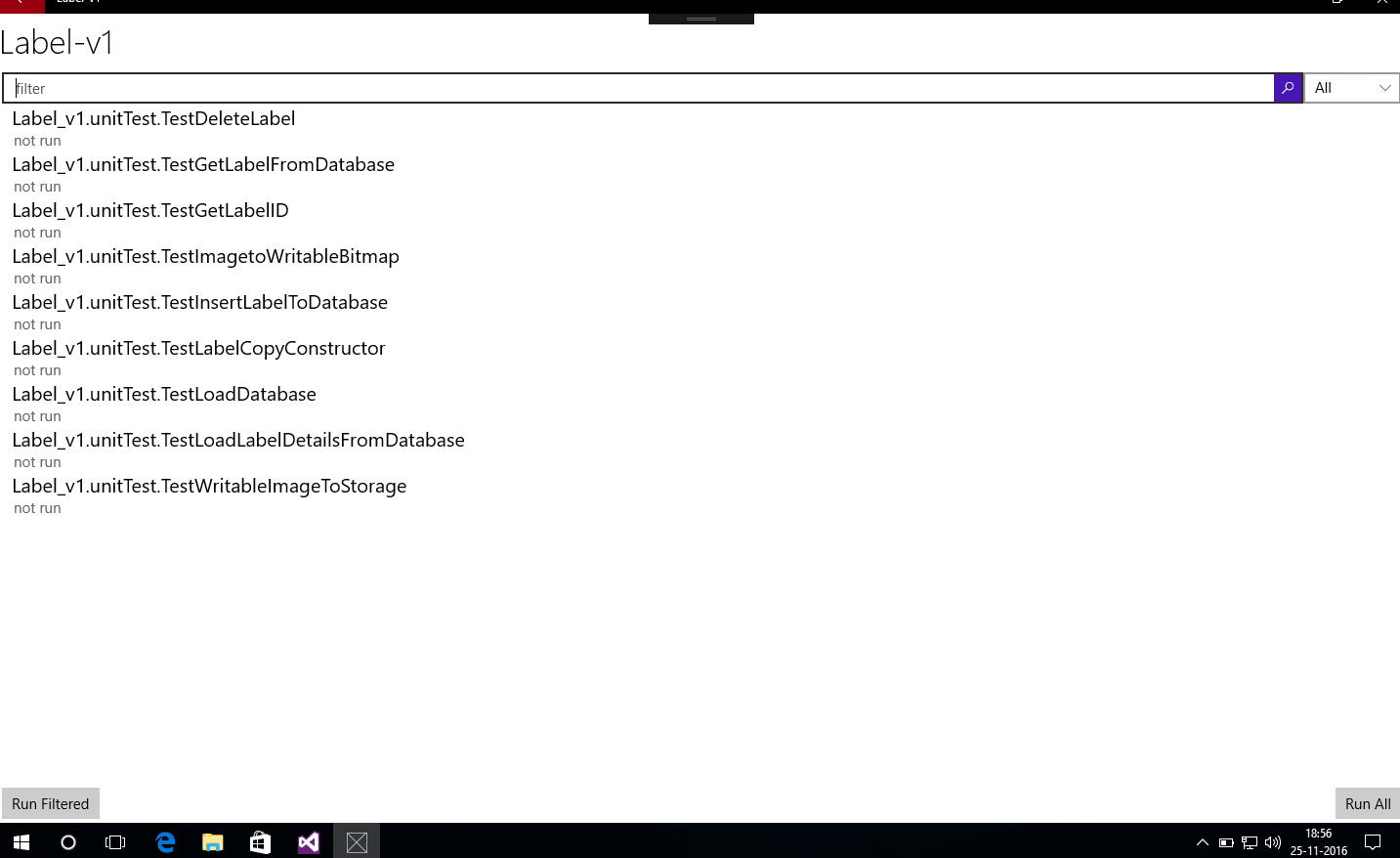
1. Test\_LoadDatabase()
2. Test\_InsertDatabase()
3. Test\_GetLabelFromDatabaase()
4. Test\_LoadLabelDetailsFromDatabase()

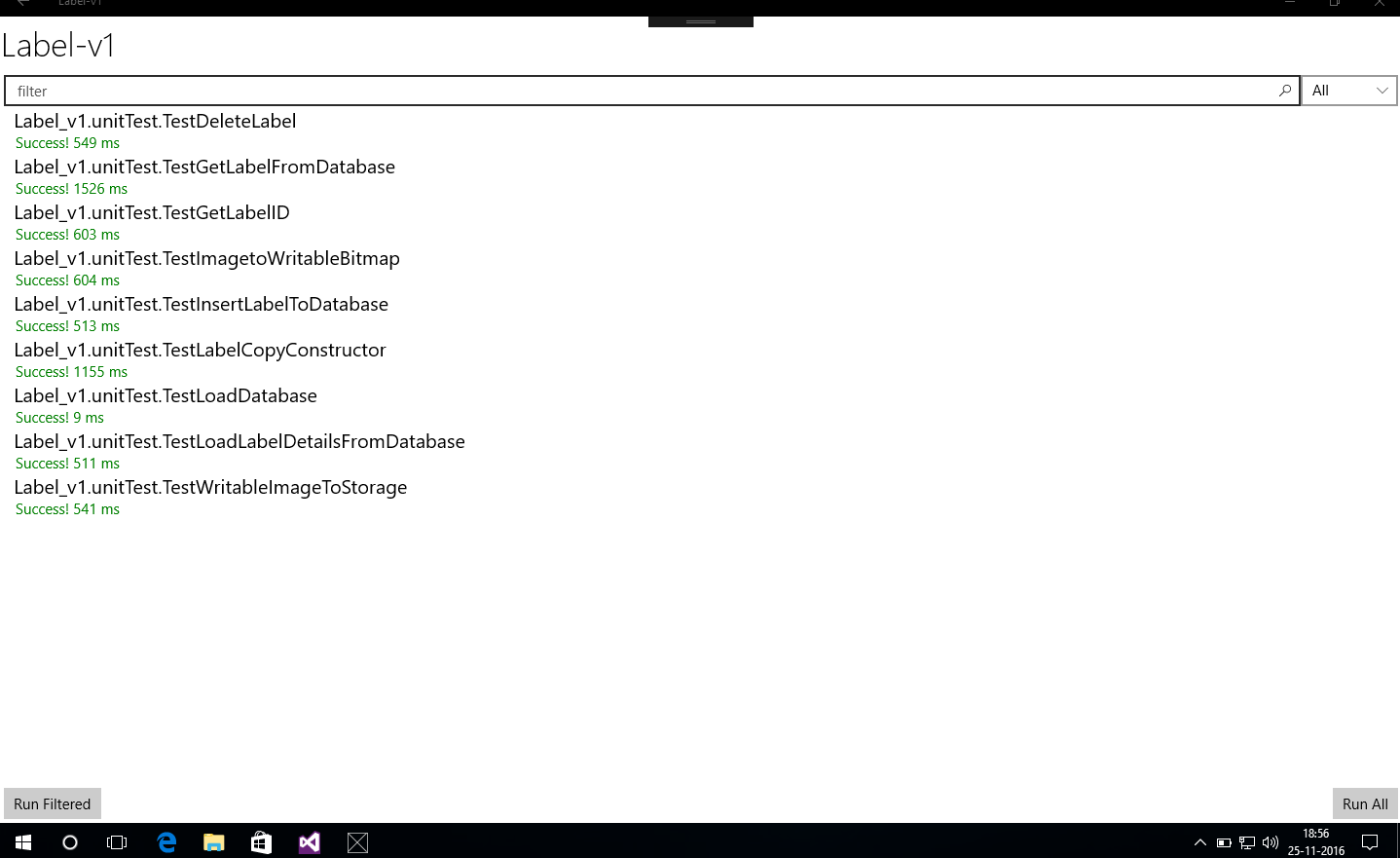


1. Test\_DeleteLabel()
2. Test\_GetLabelID()
3. Test\_WriteableImageToStorage()
4. Test\_ImagetoWriteableBitmap()
5. Test\_LabelCopyConstructor()

These are our modules in our app. We have created test cases and ran them on local machine. Below are the screenshots of the test cases.



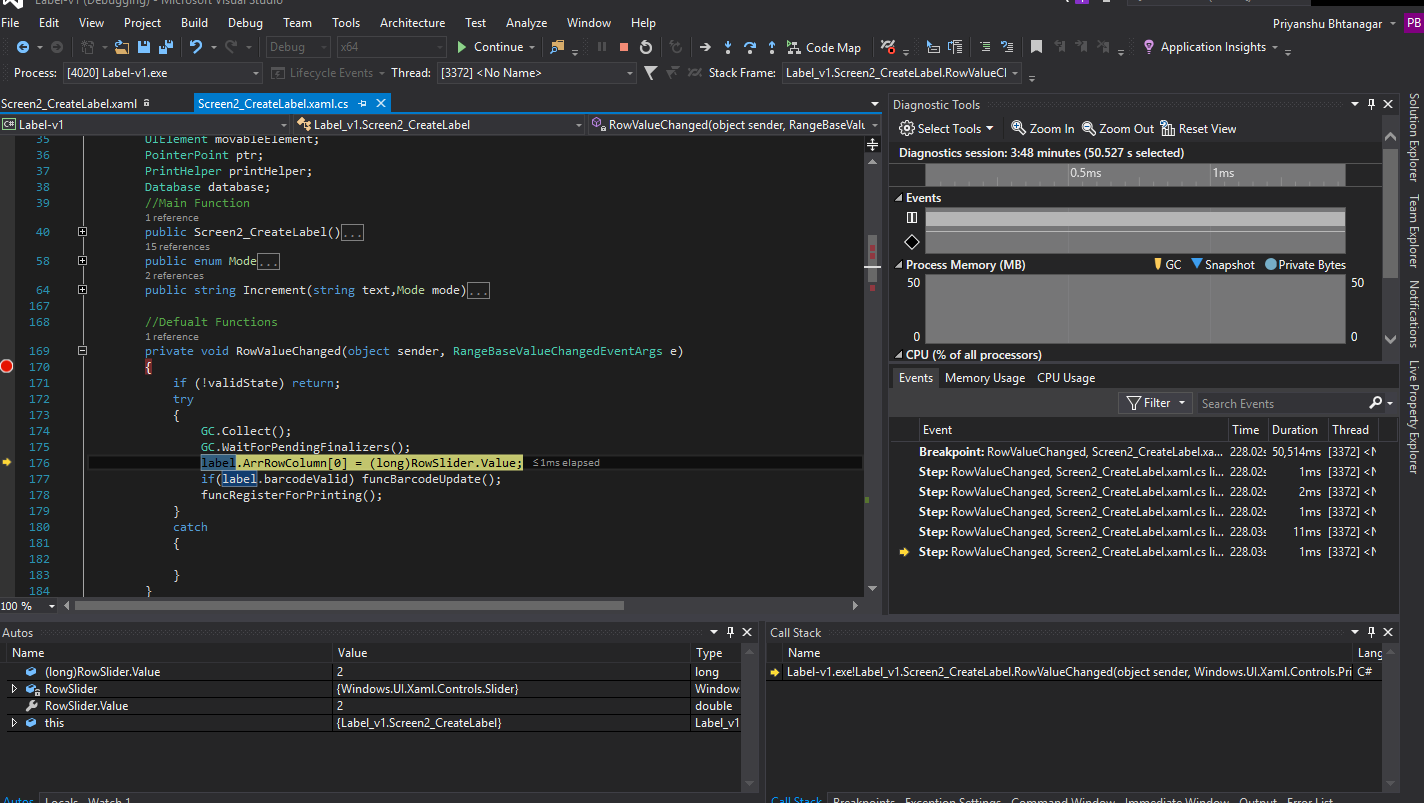




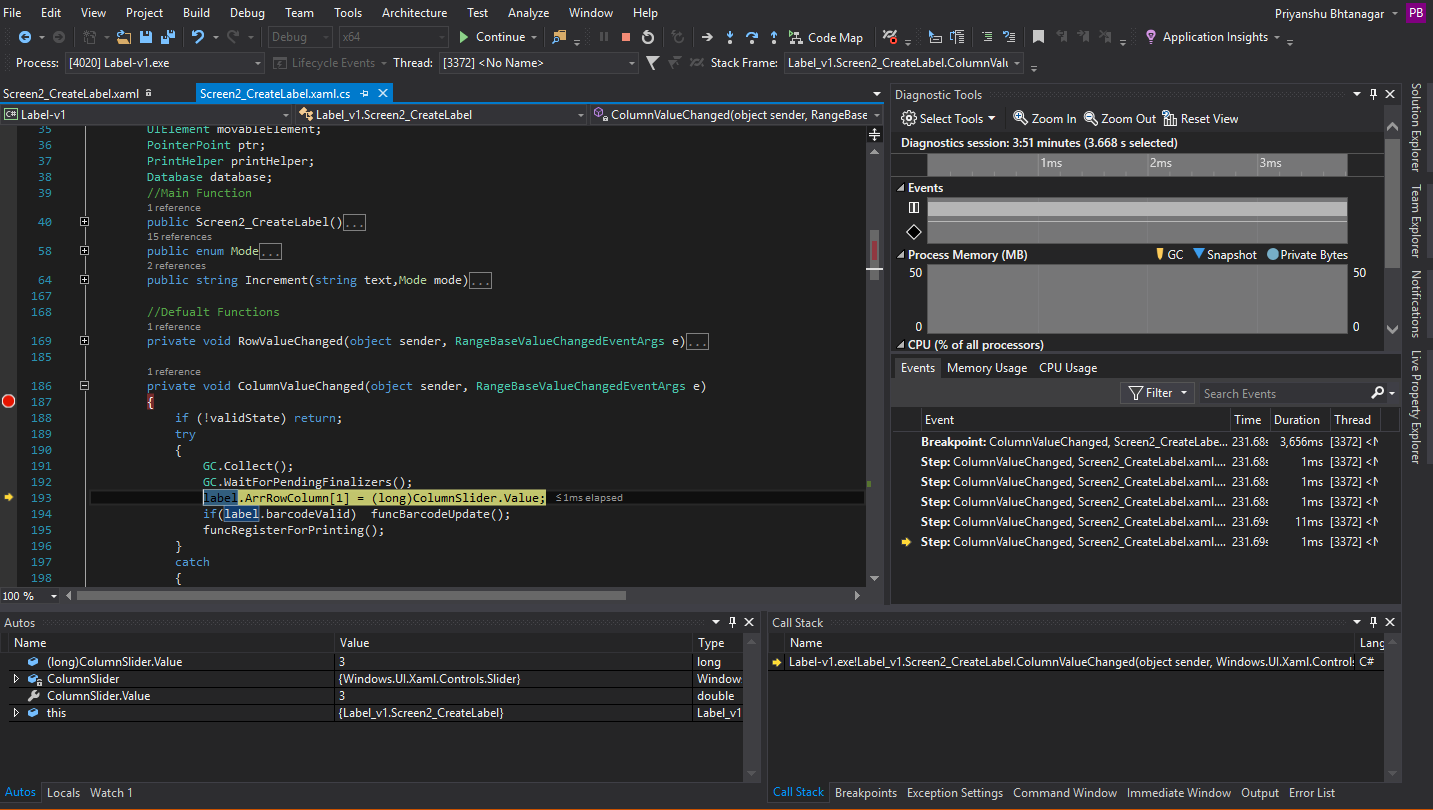
All test were successful.

3. Integration Testing

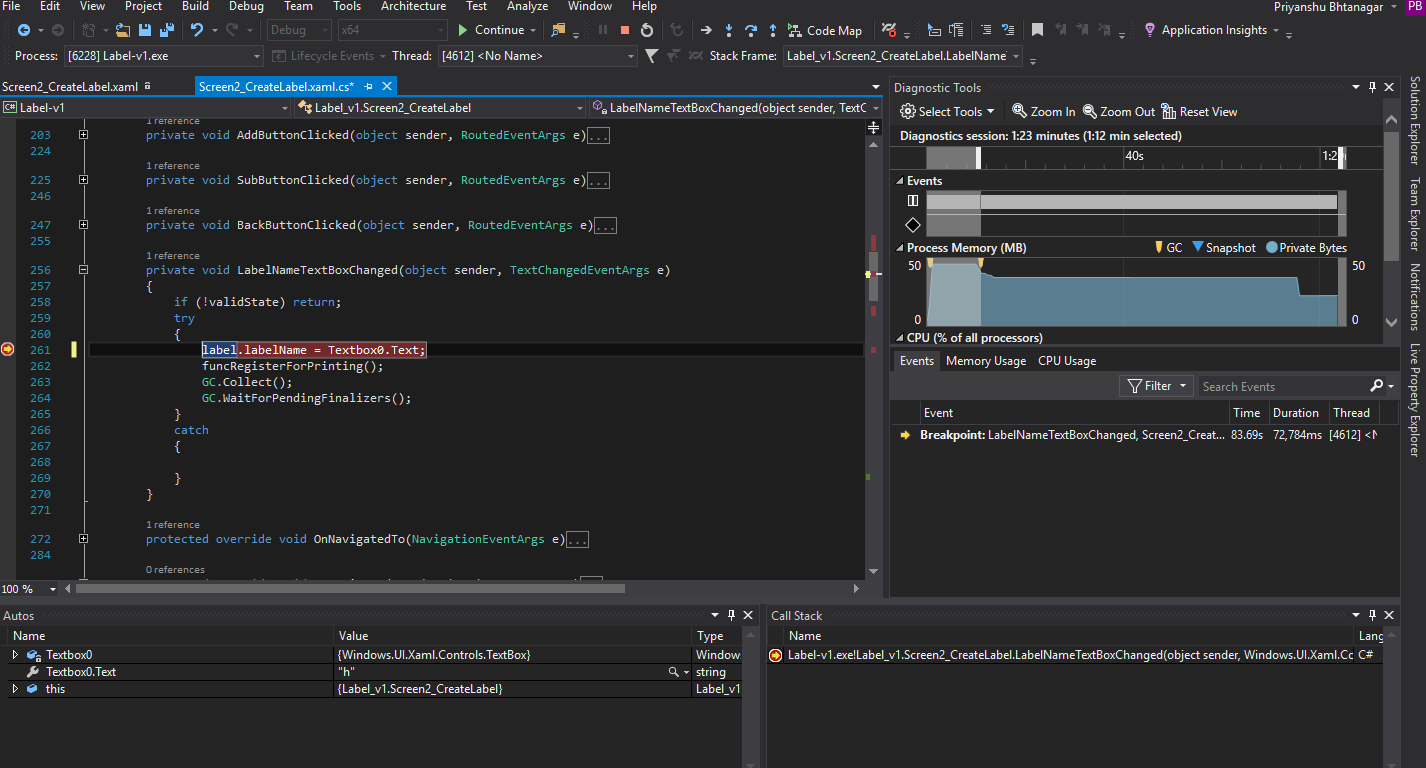
1. First Integration test is to test if number of Row selected from UI Slider module to C# variable.



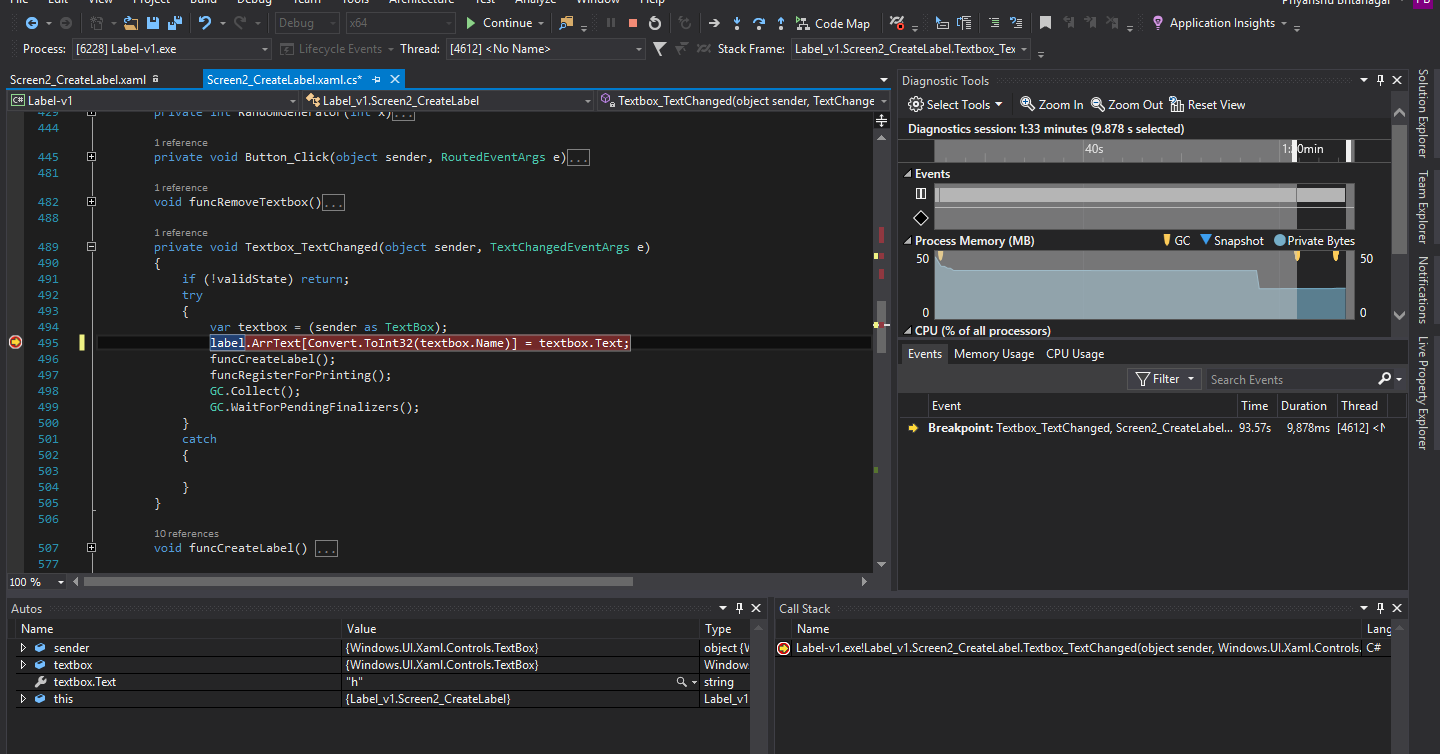
2. Number of Column selected from UI module to Label.ArrRowColumn[1].



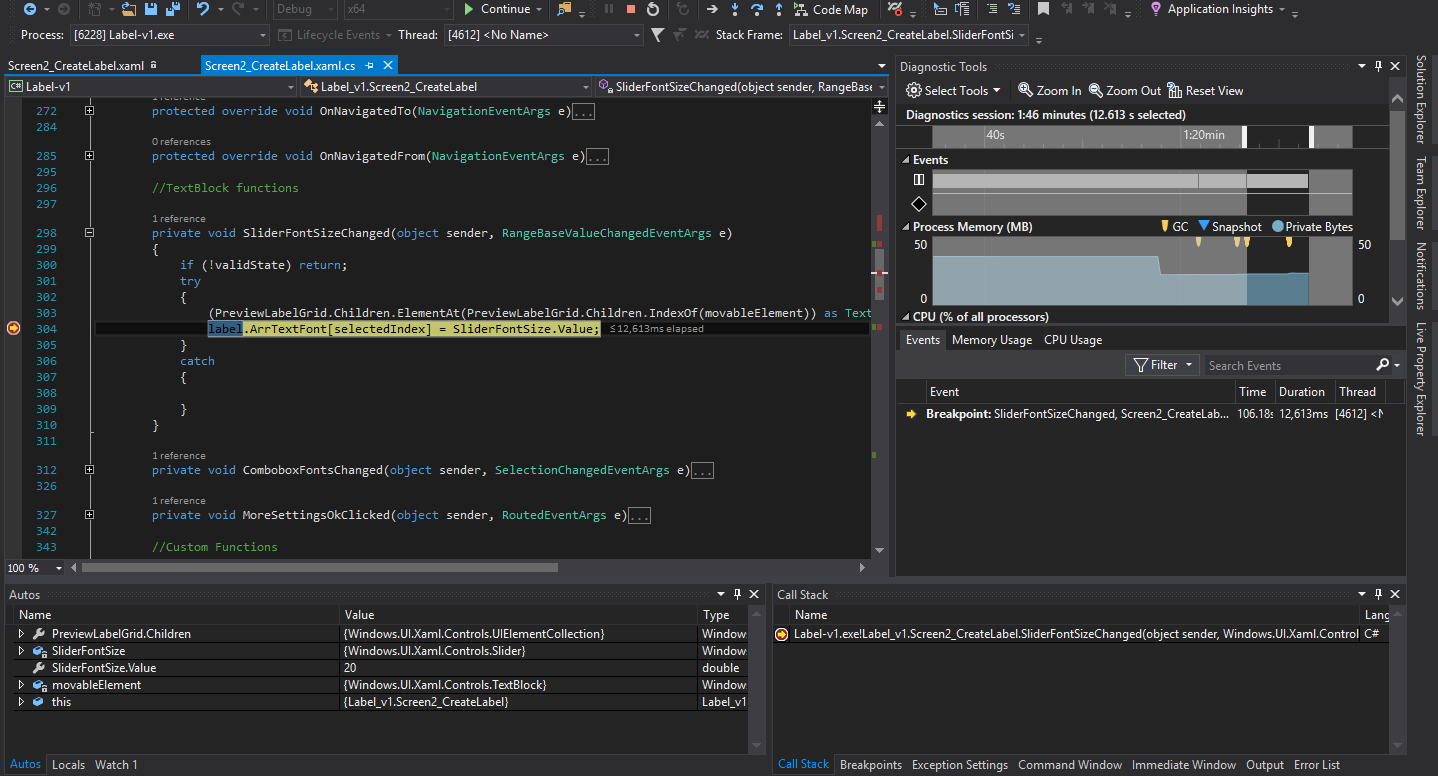
3. We tested if Text of the Label name Textbox is being saving inside the Label object or not. Here UI module and Label class are tested.



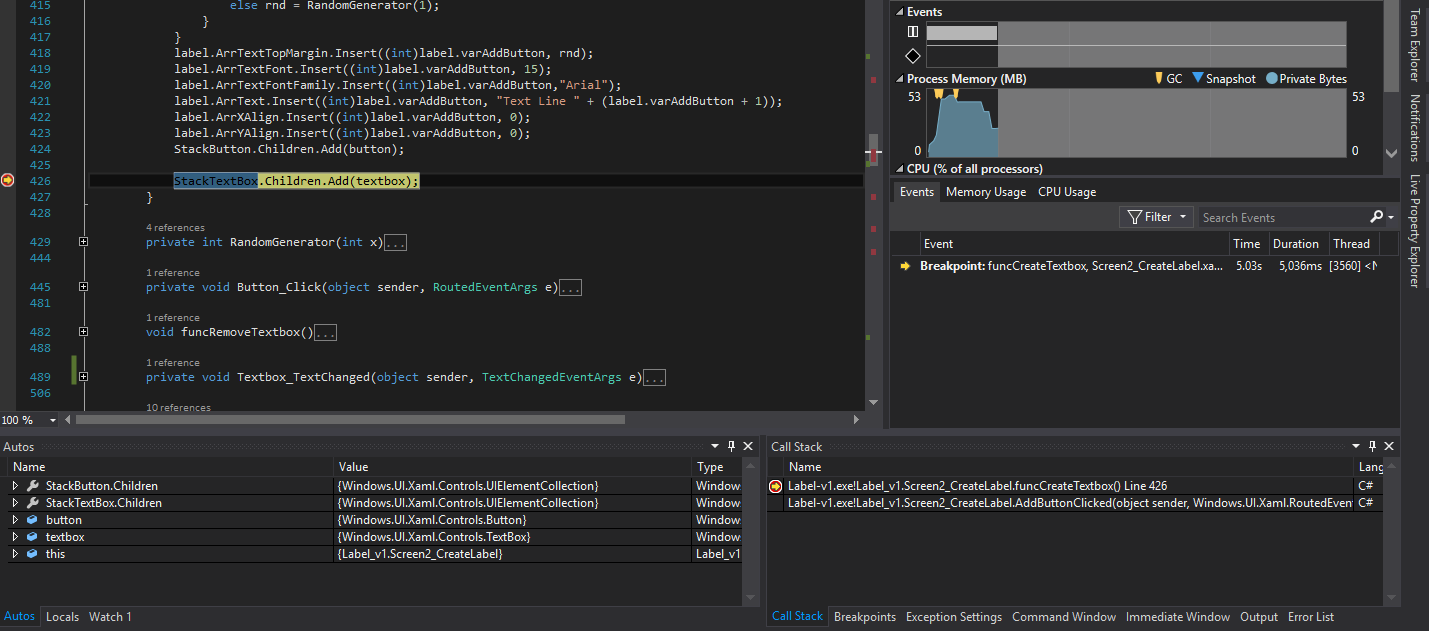
4. We tested if Text of the Textbox is being saving inside the Label object or not. Here UI module and Label class are tested.



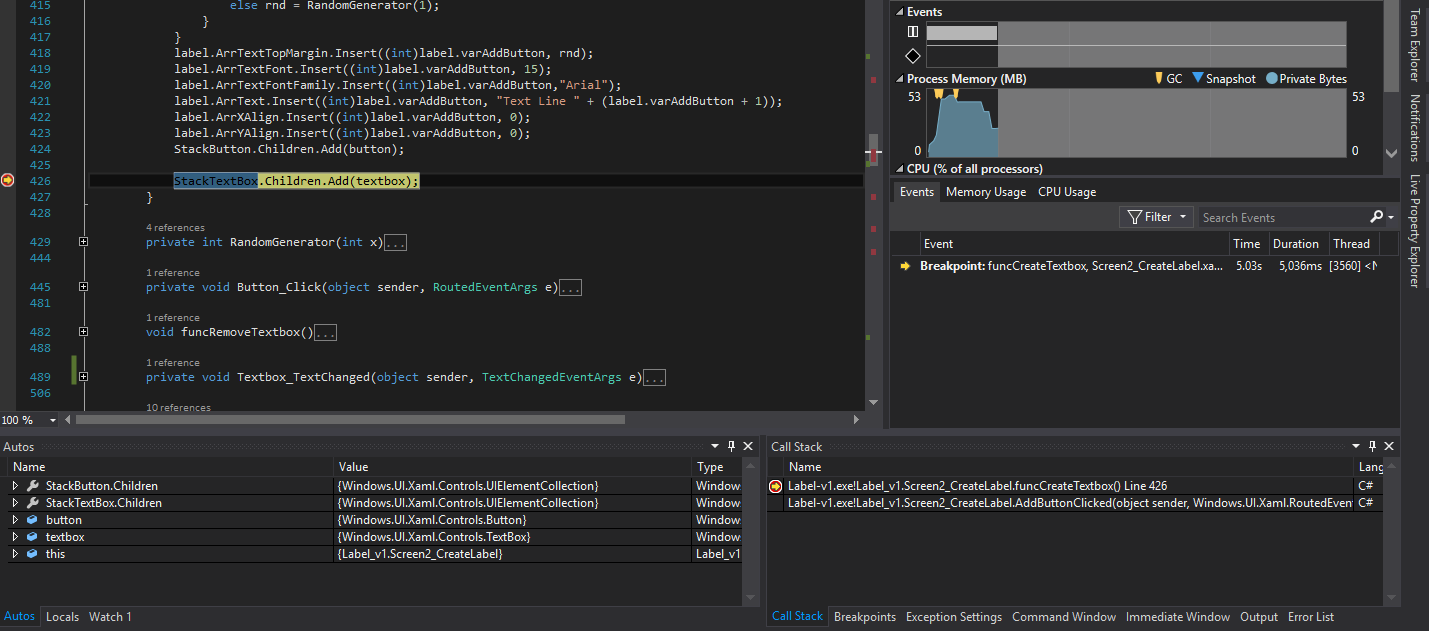
5. We tested if Font size is stored in Label object from Slider module.

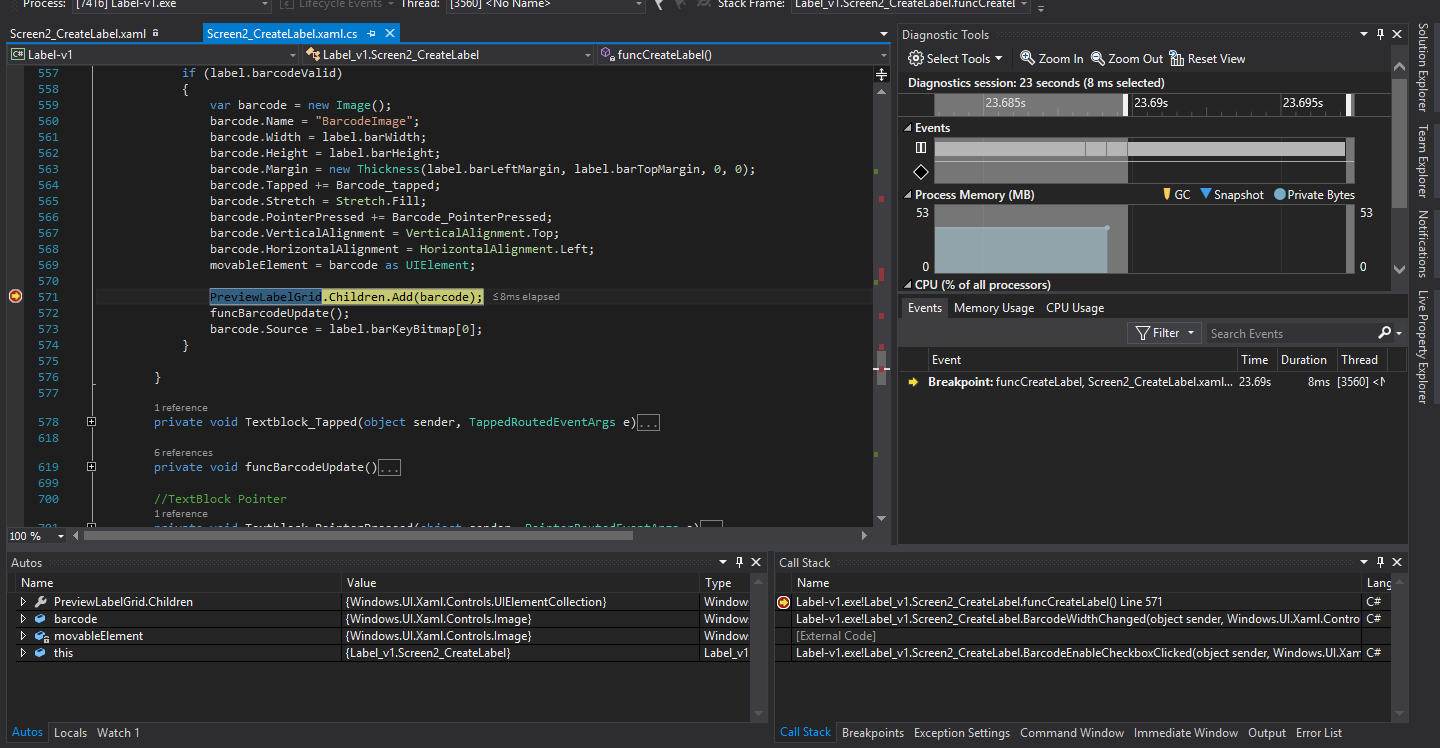


6. We tested if value Combobox module is stored in Label Object

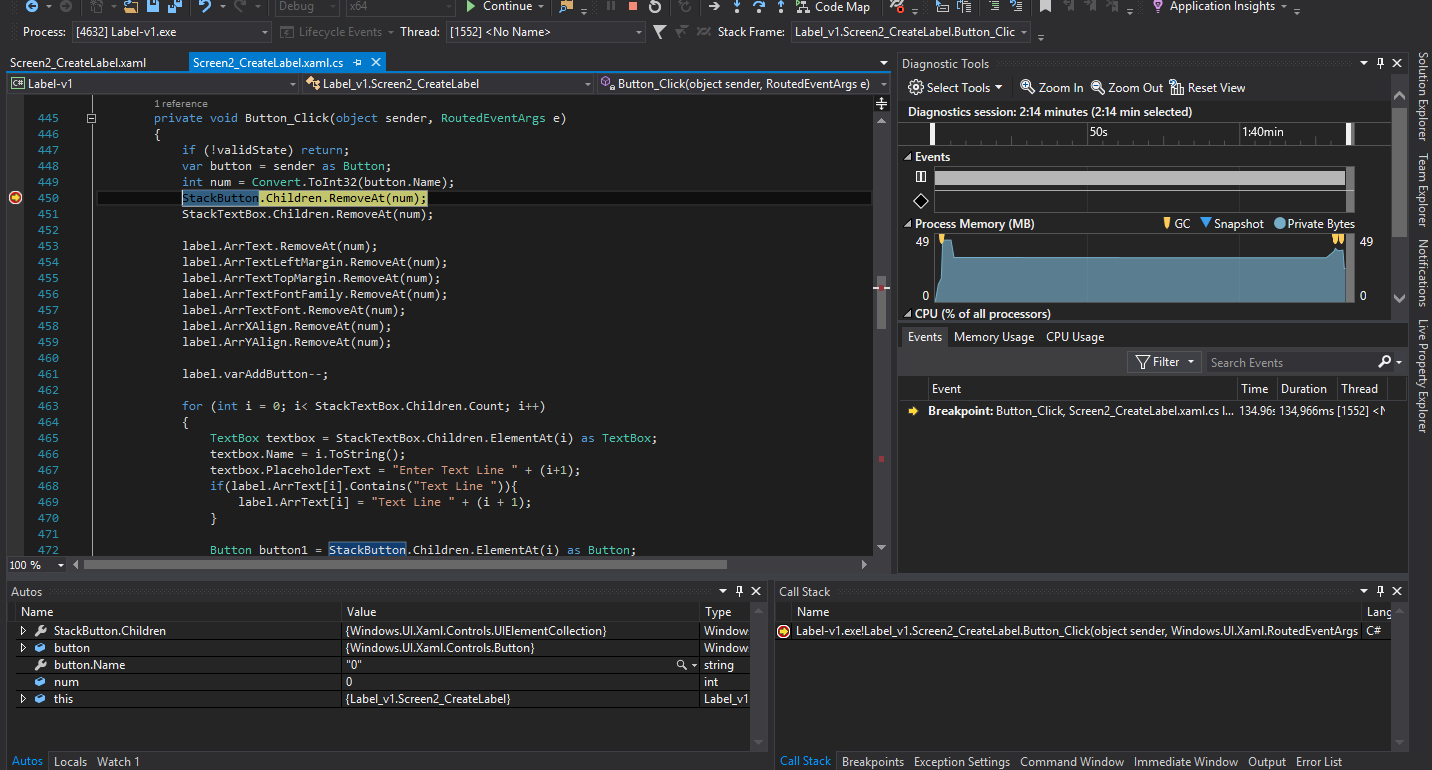


7. We tested if textbox object from TextBox Class is added into StackPanel object from StackPanel class.

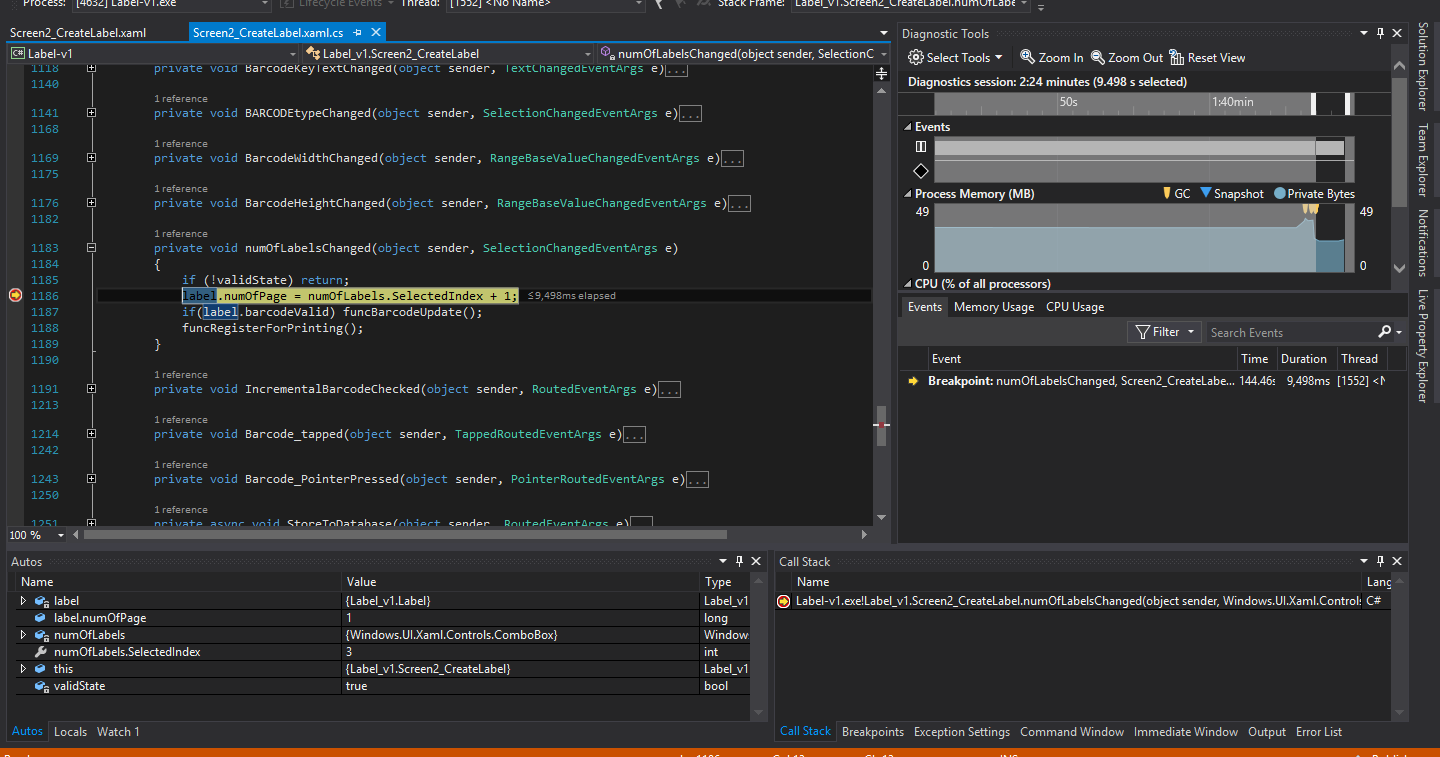


8. We tested if object of Barcode class is added into object of Grid class.

9. We tested if object at integer num is removed from the object of StackPanel class.

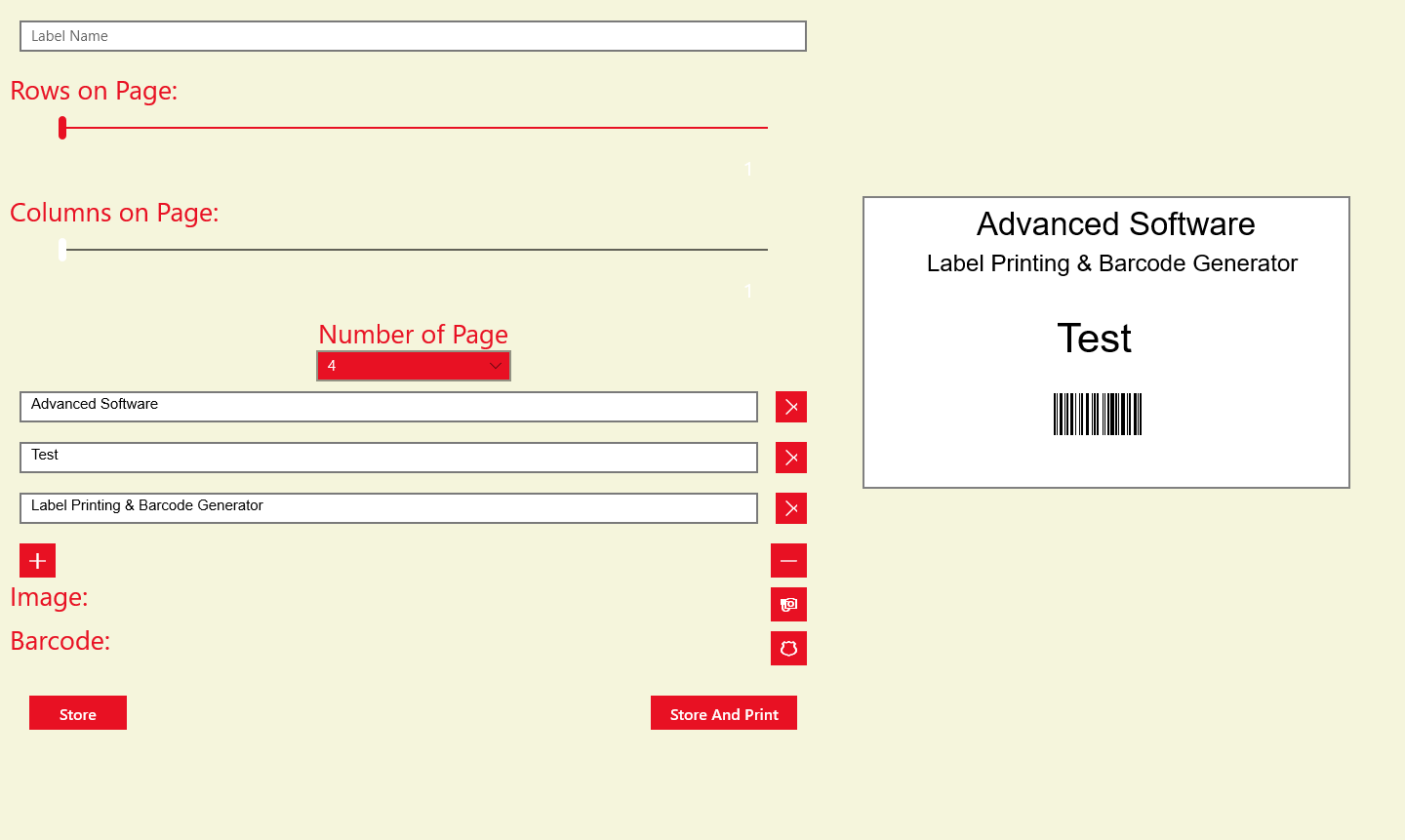


10. We tested if Combobox of Number of Page is returning correct value in the Label object.

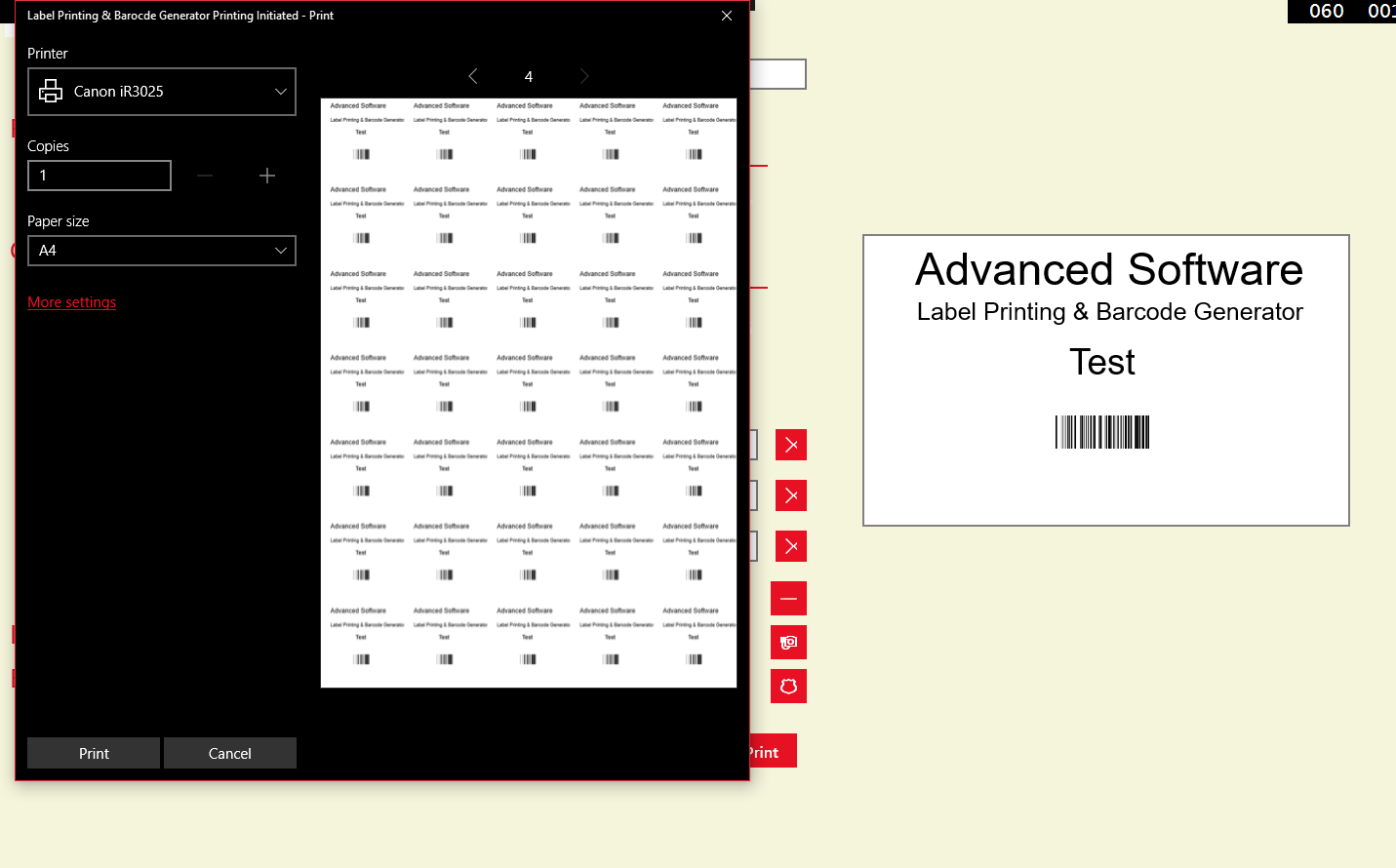


System Testing

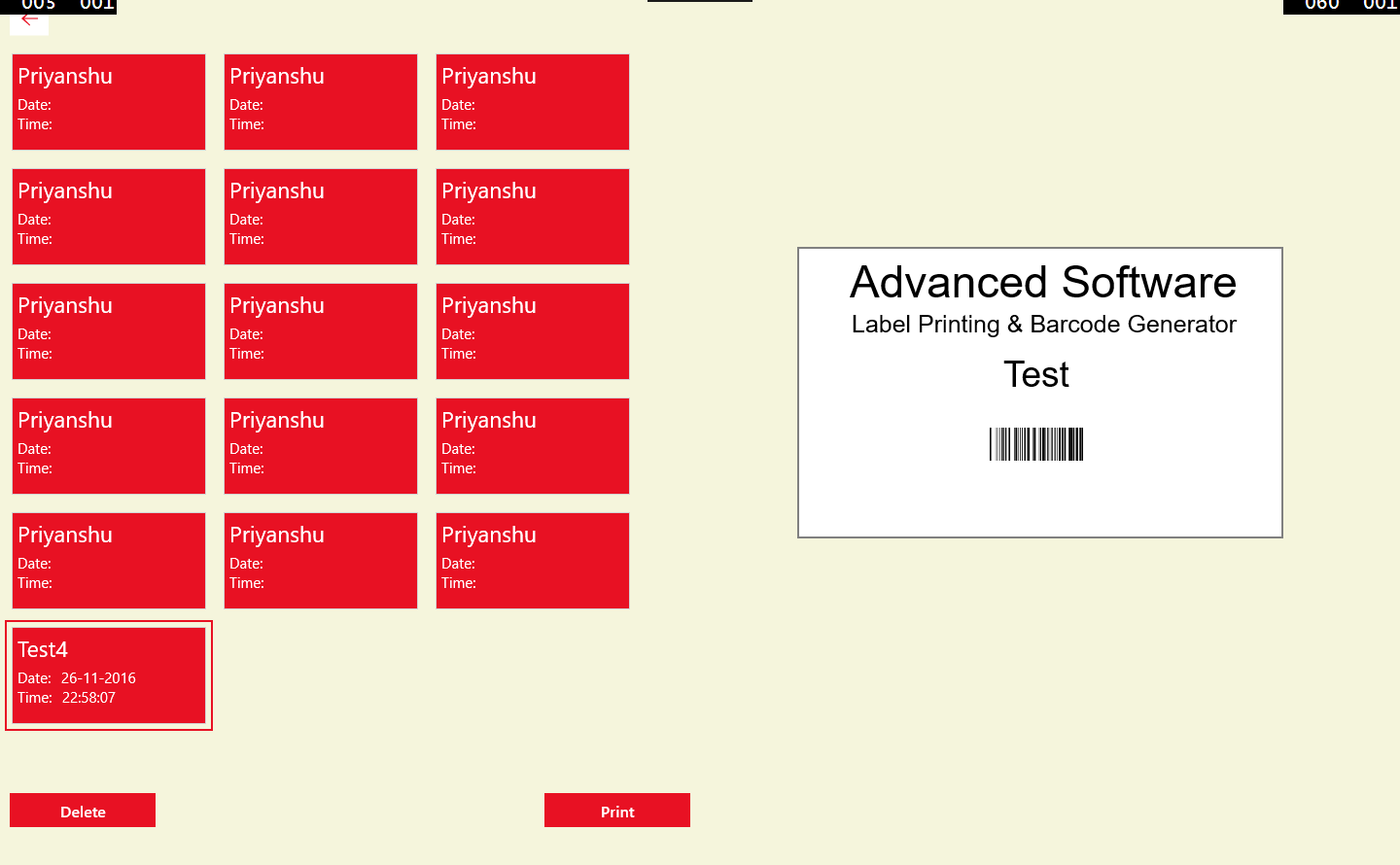
In system testing we decided to design a Label with firm’s name, App’s name, text, a barcode in uniform mode. We selected number of pages in 4. Below you will see the screenshot of the Label.



We then saved the label and moved forward for printing. We selected 7 Rows and 5 Columns. Below you can see the preview of our Label. If you see number of Pages for printing above the preview it is 4 as we selected.



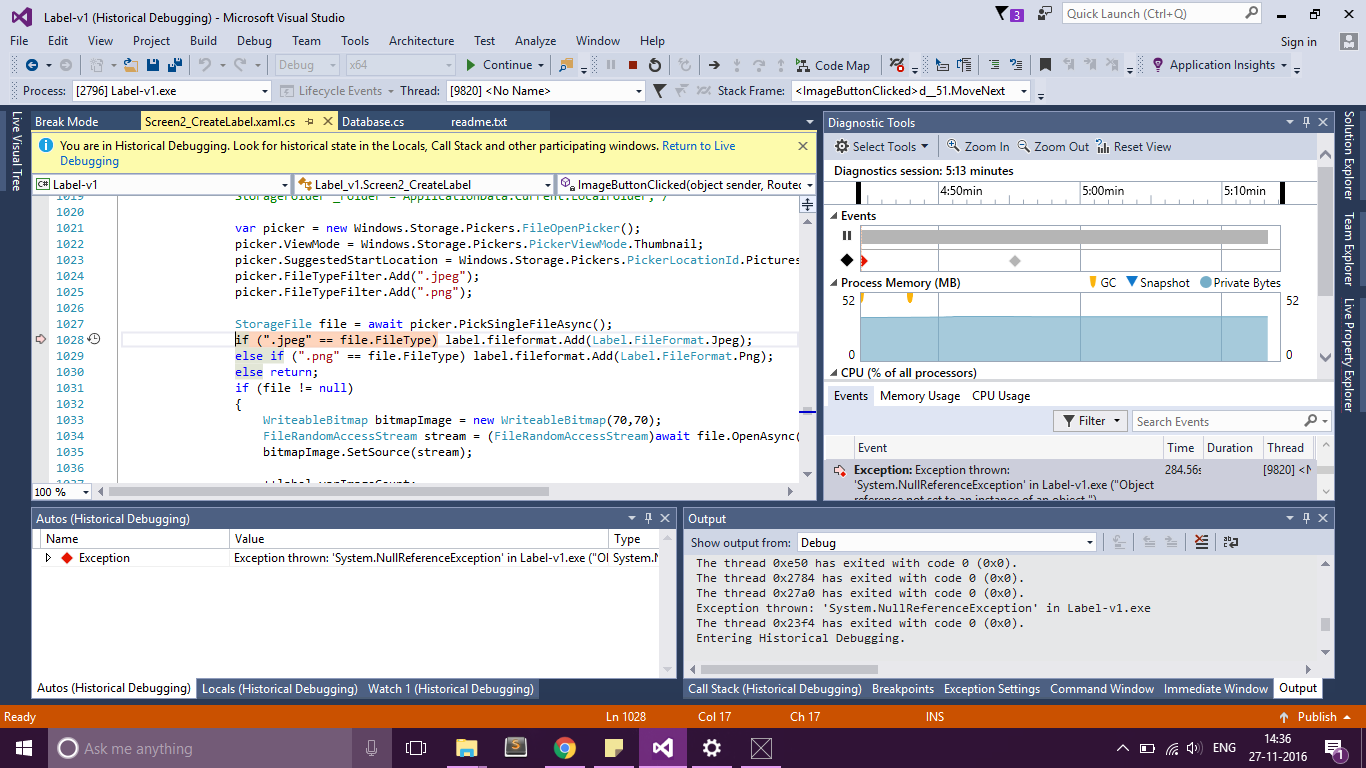
We saved our label with the name of Test4. We checked if everything we had created is saved and properly or not. You can see below that we have selected Test4 at the end and the same preview is available in the preview. We can also now again go ahead with the printing. Creating Labels have become very easy and simple!



Bug And Fixes

The problem occurred in the below screenshot that we can only take “JPEG” type of image format.

We tried to insert an image of type “PNG” which gave this error.



We have few architectural errors which is needed to be solved. We are designing a version 2 of this app which will solve all these sort of problems.