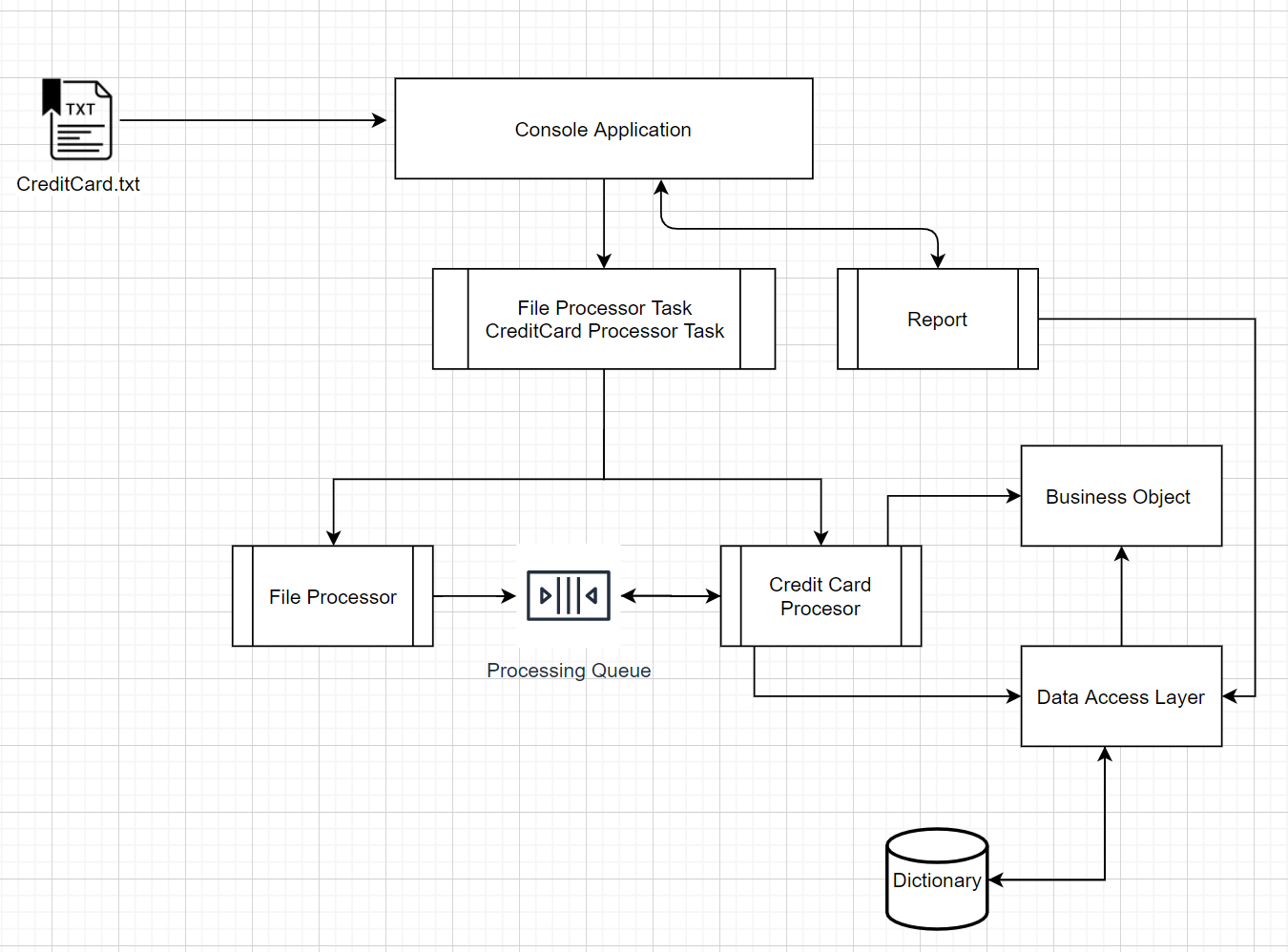
**Overview of design:**

I have taken publisher and subscriber model in my design. Reason for choosing this approach is to make sure file processing is getting processed faster. I have used .Net Task Parallel library which does multithreading and gracefully handles threads.

Since the requirement is to output result in ascending order and it also mentions there won’t be any duplicate name in Add Operations, in order to accomplish these task I decided to store data in SortedDictionary data structure since it performs insert operation faster and faster data retrieval.

The below diagram articulates the overview on my design.



**Why you picked the programming language you used**

Visual Studio is primary editor which I have been using to code and it also provides extensive feature for developers to develop application in much faster way instead of writing lot of codes.

The reason for choosing C# as the programming language is,

* I have been having experience in developing application using Microsoft Technology stack and primarily C# as my development language.
* Any .Net application can be executed in any platform where .Net Frame is been installed since all .Net supported is compiled to Intermediate language which then compiles to machine native code.
* Automatic memory management for managed code and extension for clearing memory using IDispose interface for un-managed code
* In built library for Parallel processing using Task Parallel Library (TPL)
* Synchronization using lock commands for critical execution.
* Unit Test Framework support for C# language

**How to install any required dependencies (runtimes, frameworks, etc)**

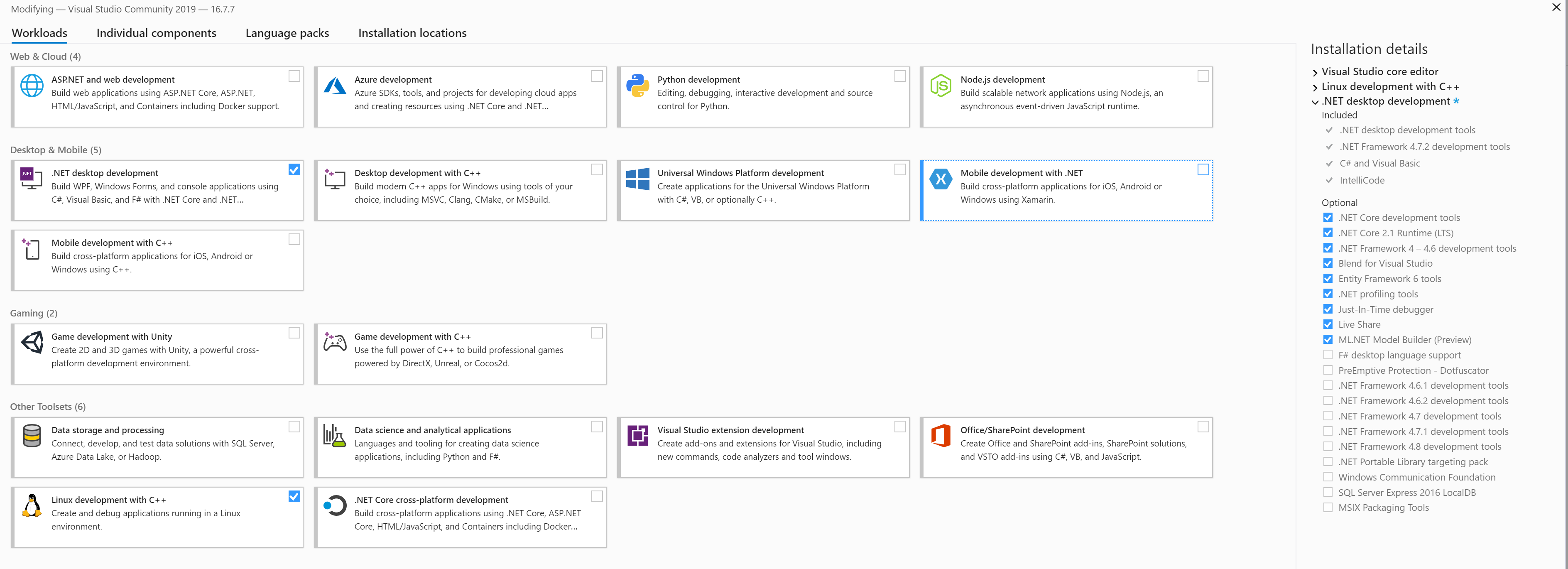
1. Navigate to this [link](https://visualstudio.microsoft.com/downloads/?utm_medium=microsoft&utm_source=docs.microsoft.com&utm_campaign=inline+link&utm_content=download+vs2019) or

Url : <https://visualstudio.microsoft.com/downloads/?utm_medium=microsoft&utm_source=docs.microsoft.com&utm_campaign=inline+link&utm_content=download+vs2019>

1. Download [Visual Studio 2019 Community edition](https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16) or

Url : <https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16>

1. Install the executable by double clicking on the file. Installable will be in the file name of “**vs\_community\_\_1841164483.1600738911.exe**”
2. Visual Studio Community 2019 – 16.7.7 window will be opened.
3. Select .**NET desktop development** under Desktop & Mobile (5) tab as shown below



1. Installable location can be updated but can to go with default location which will be displayed under the bottom of the same window
2. Click on Install button.

**Note : This might take couple of mins and restarts.**

**How to build, package or compile your code if applicable**

* **Build** :

1. Unzip CreditCardProcessor.zip file. For now lets assume its Unzipped to E:\CreditCardProcessor
2. Open **Developer Command Prompt for VS 2019**
3. Navigate to E:\CreditCardProcessor
4. Run the below command

**MSBuild CreditCardProcessing.sln -t:Rebuild -p:Configuration=Release**

1. Step 4 will result in CreditCardProcessor.exe executable under the below location

E:\CreditCardProcessing\bin\Release\

**How to run your code and tests**

* **How to run code and test the application:**

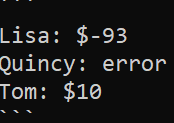
**Note : As part of Build section🡪 Step 5, executable will be located in**

**E:\CreditCardProcessing\bin\Release\**

1. Open **Developer Command Prompt for VS 2019**
2. Navigate to **E:\CreditCardProcessing\bin\Release\** location
3. Run the below command and lets assume data file is in **E:\Data\input.txt** location

**CreditCardProcessor.exe “E:\Data\input.txt”**

Execution of the above command will produce the below output



* **How to run test code:**

**Note : As part of Build section 🡪Step 5, test binaries will be located under the below**

**location**

**E:\Study\CreditCardProcessing\DataAccessLayerTests\bin\Release\netcoreapp3.1**

1. Open **Developer Command Prompt for VS 2019**

Navigate to E:\Study\CreditCardProcessing\DataAccessLayerTests\bin\Release\netcoreapp3.1

1. Run the below command

**vstest.console.exe CreditCardProcessor.DataAccessLayerTests.dll**

Execution of the above command will produce the below output

