DSW03A1

Semester Test 1
Due: 29 March 2023 [Due @: 18:00]
Marks: 75

UNIVERSITY OF JOHANNESBURG

Instructions

Complete this test and submit a **cleaned** and **compressed** Visual Studio Solution folder with your test response before the submission deadline via Blackboard. Please further note:

- This is semester test 1 for DSW03A1, which counts **10%** towards your final Mark
- Zip the completed solution from Visual Studio as follows: **Student Number Surname.zip.**
- **IMPORTANT**: Please name your Visual Studio Solution like so
  - [STUDENT_NO]_[SURNAME]_SemTest1
  - (e.g. 2018993320_Mhlanga_SemTest1
- Upload the completed solution using Blackboard under the" Test Submission" link.
- Please save your work regularly to prevent any data loss.
- **PLAGIARISM**: Please refer to your Learning Guide as well as the latest University of Johannesburg's plagiarism policy document entitled: "POLICY: PLAGIARISM"
- **COPYING**: This is an individual assignment; if any copying is detected, all parties involved will score a **0%** for the assignment and **WILL** face disciplinary consequences.
- **NO LATE SUBMISSION**: If you do not submit within the given time frame, zero will be awarded.

Background:

Recall from LU1 that Big Data is often large volume of data – both structured and unstructured; and that it may be analysed computationally to reveal patterns, trends, and associations; keep this mind as you engage through this assessment.

You are required to develop a C# windows-based application that consumes a dataset called “**MOCK_DATA.csv**”. The application is to read the dataset and display the data based on the user’s selection.

The application must have two class:

- **Data** class: contains attributes of each record in the dataset.
- **DataManager**: responsible for reading the data, determining the domain type and counting the number of records for each domain.

Now follow the UML class diagram and the description below and complete the **Data** class. Save file as **Data**

Data
- number: int - name: char - surname: string -email: string - gender: string - ip: string
+ Number: int + Name: string + Email: string + Gender: string + IP: string

The class contains the above class variables (data members). The class also includes properties for each data member.

Now follow the UML class diagram and the description below and complete the **DataManager** class. Save file as **DataManager**

DataManager
- data[]: Data
+ readFromFile(): Data[] + determineDomain(string): Data[] + getCounter(string): int + getRecord(int): Data

Attribute/Method/Class Name	Description
data[]	**Data members.** **data** is an array that must store objects of type Data.

readFromFile(): Data[]	The method reads data from the csv file. The methods must split each field in the record and set the object's attribute or data member. The method must return the array of type Data containing objects which have been set with the information read from the file.
determineDomain(string): Data[]	The method accepts the domain name via parameters e.g. ".com" The method must determine and store all records ending with the accepted domain in another array of type Data. The method must return the array of type Data.
getCounter(string): int	The method accepts the domain name via parameters e.g. ".com". The method must count and return the number of records for that given domain name.
getRecord(int): Data	The method accepts an index via parameters. The method must return an object from data[] array at the given index.

Form Load

When the form load, the data from the csv file must be read and each object must be set with data from each record.

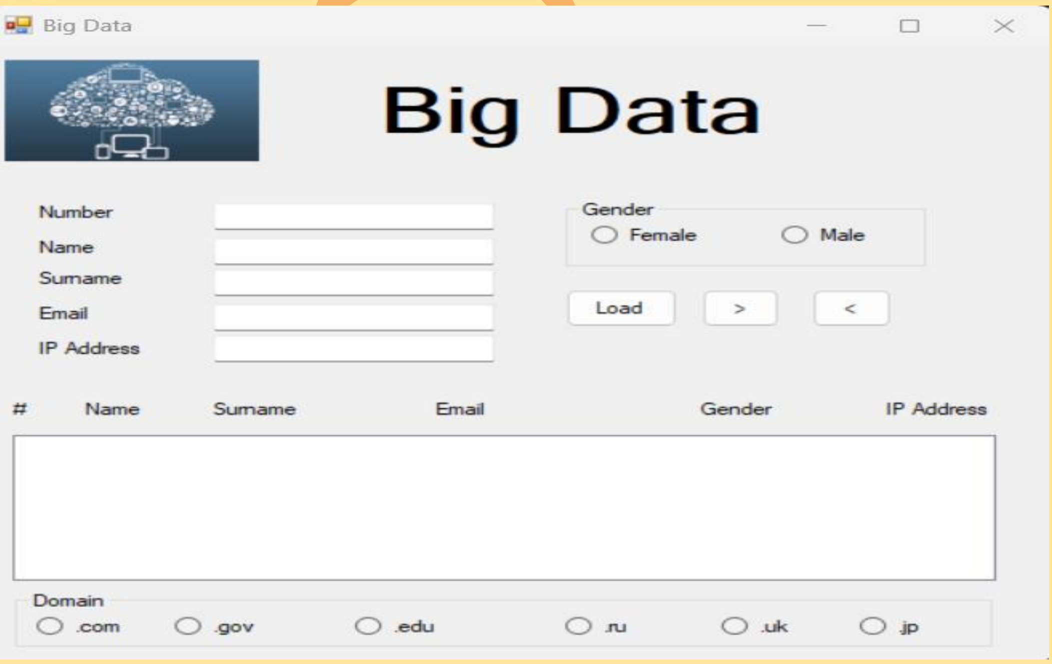
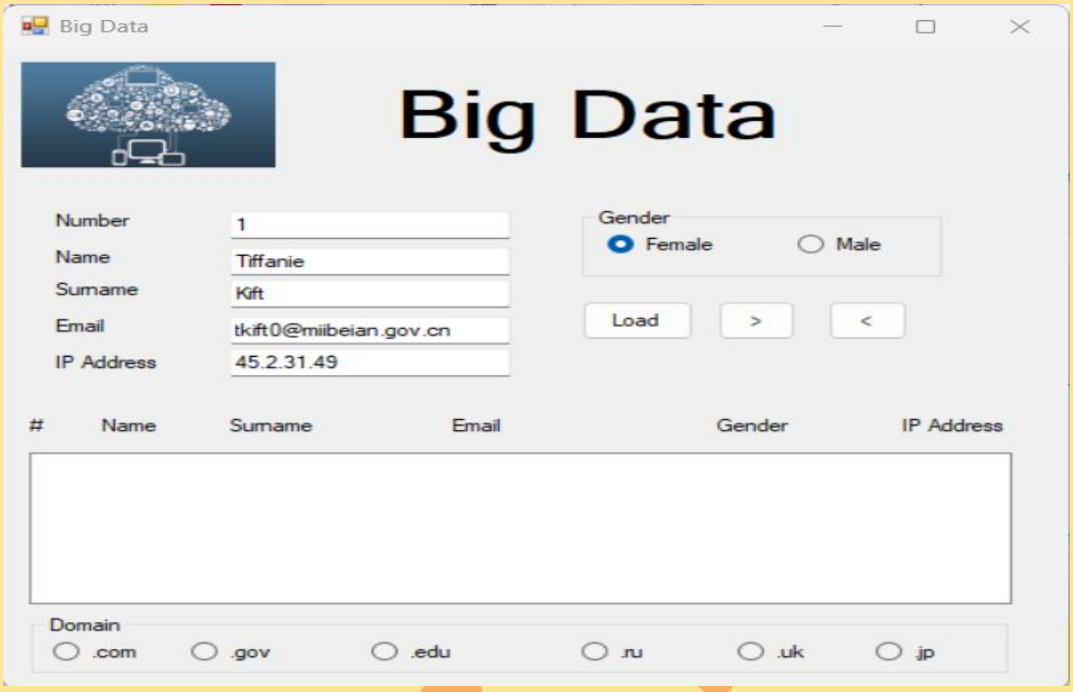


Figure 1

Load Button

The load button must display the first record in the text fields as shown in **Figure 2.**



Number: 1

Name: Tiffanie

Surname: Kift

Email: tkift0@miibeian.gov.cn

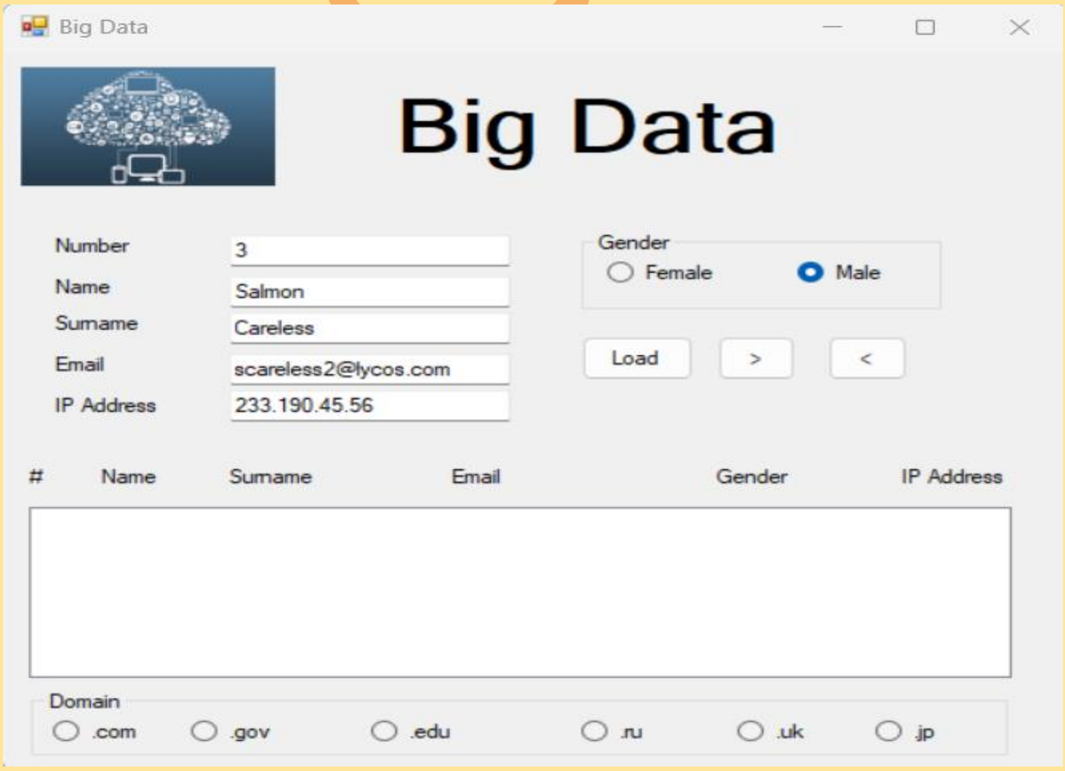
IP Address: 45.2.31.49

Gender: ● Female ○ Male

Figure 2

> Button

The > button must display the next record in the text fields as shown in **Figure 3.**



Number: 3

Name: Salmon

Surname: Careless

Email: scareless2@lycos.com

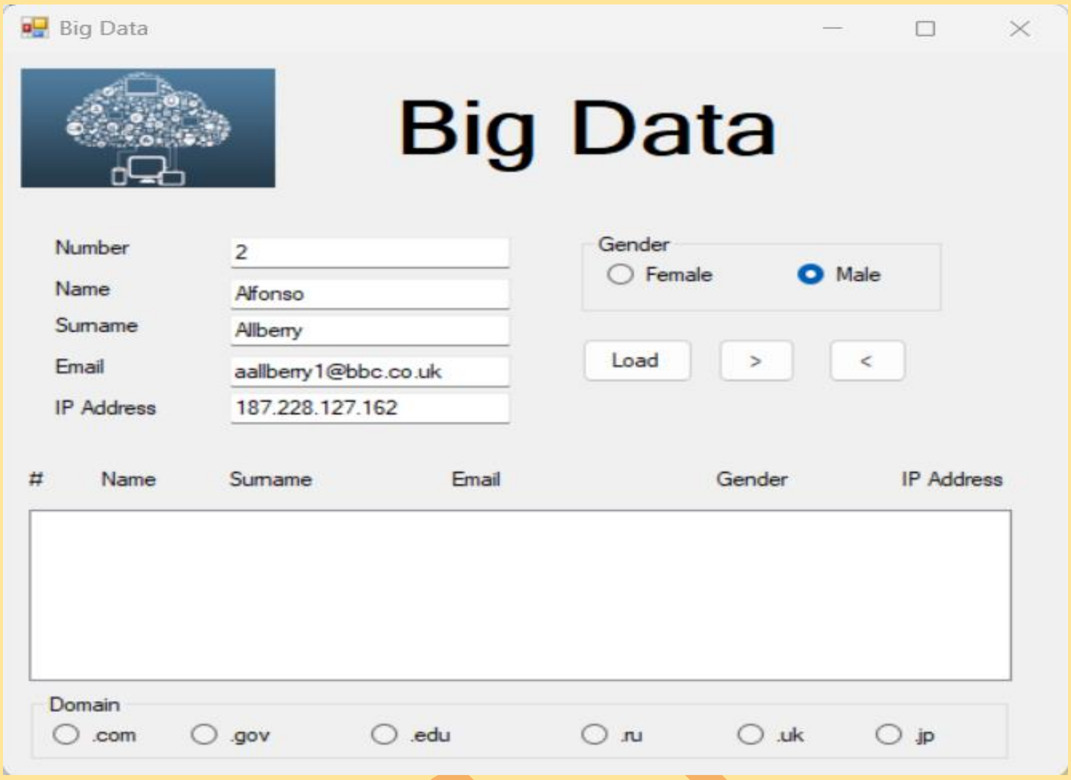
IP Address: 233.190.45.56

Gender: ○ Female ● Male

Figure 3

< Button

The < button must display the previous record in the text fields as shown in **Figure 4.**



Number	2
Name	Alfonso
Surname	Allberry
Email	aallberry1@bbc.co.uk
IP Address	187.228.127.162

Gender: ○ Female ● Male

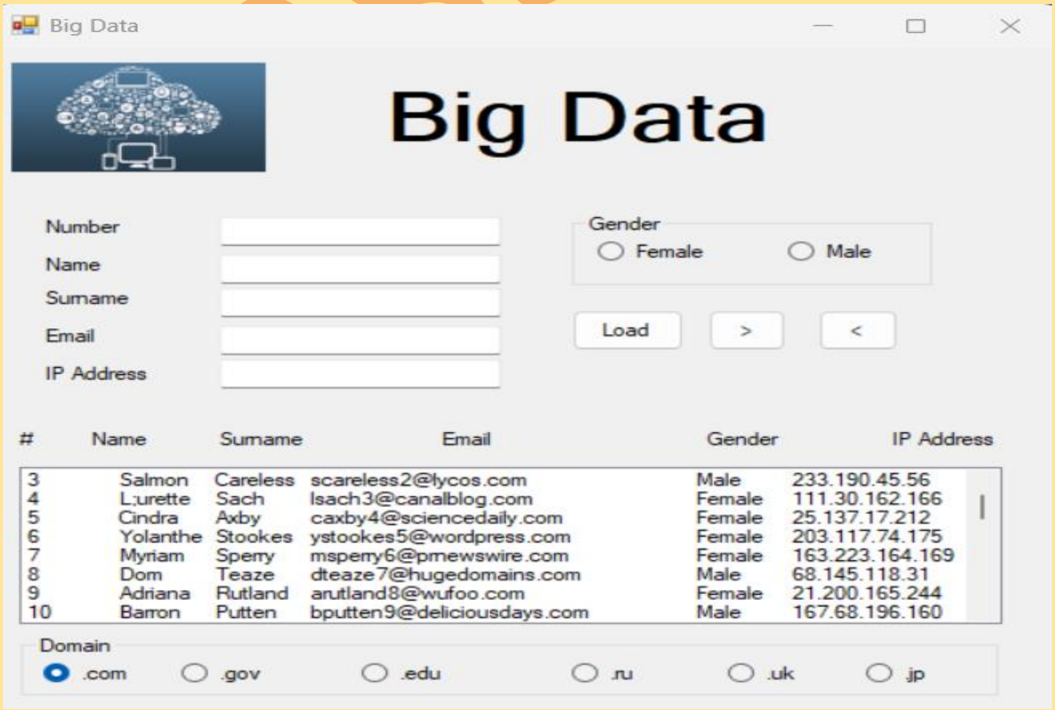
#	Name	Surname	Email	Gender	IP Address

Domain: ○ .com ○ .gov ○ .edu ○ .ru ○ .uk ○ .jp

Figure 4

Radio Buttons

Each radio button displays all records having that domain the, for example the ".com" domain shown in **Figure 5.**



Number	
Name	
Surname	
Email	
IP Address	

Gender: ○ Female ○ Male

#	Name	Surname	Email	Gender	IP Address
3	Salmon	Careless	scareless2@lycos.com	Male	233.190.45.56
4	L;urette	Sach	lsach3@canalblog.com	Female	111.30.162.166
5	Cindra	Axby	caxby4@sciencedaily.com	Female	25.137.17.212
6	Yolanthe	Stookes	ystookes5@wordpress.com	Female	203.117.74.175
7	Myriam	Sperry	msperry6@prnewswire.com	Female	163.223.164.169
8	Dom	Teaze	dteaze7@hugedomains.com	Male	68.145.118.31
9	Adriana	Rutland	arutland8@wufoo.com	Female	21.200.165.244
10	Barron	Putten	bputten9@deliciousdays.com	Male	167.68.196.160

Domain: ● .com ○ .gov ○ .edu ○ .ru ○ .uk ○ .jp

Figure 5

Submission Guidelines

Please adhere to the instructions above and the following guidelines for your submission:

- The test **MUST** be submitted via Blackboard; no emailed or printed assignments will be accepted.
- You **MUST** *clean* your Visual Studio Solution before compressing and submitting in order to reduce the overall size of your files.
- Ensure that your **full name** and **student number** appear in your code as comments at the beginning of the program and the project folder.
- Ensure that your code works as **marks are allocated only for working code**
- As good coding practice, marks will be awarded for the use of good programming principles, which include the naming of variables accordingly, providing meaningful comments, etc.
- Marks will be allocated in the following manner:

Criteria	**Weight (%)**
Ability to create Data Class	5
Ability to create an array of type **Data** in the DataManager class	2
Ability to create **readFromFile** method	17
Ability to create **determineDomain** method	9
Ability to create **getCounter** method	4
Ability to create **getRecord** method	2
Ability to create an object of **DataManager** in the GUI to be used throughout	2
Ability to use **DataManager** object in the GUI to display data from the csv file	9
Ability for the load, > next, and previous < buttons in the GUI to work accordingly	9
Ability for the domain name radio button to display the appropriate records	16
TOTAL	75

Table 1: Semester Test 1 Mark Allocation