

## Project 2 Specification

<b>Faculty</b>	Information Technology	<b>Module Name</b>	Python Programming
<b>Module Code</b>	ITPYA0	<b>Project Number</b>	2
<b>Total Marks</b>	150	<b>Copy Editor</b>	Ms Nicole Stern

### Instructions to Student

1. All work, including draft notes, must be submitted with the completed project.
2. The program must be operational with as few faults as possible.
3. 20% will be deducted for this project if it is returned for resubmission due to plagiarism.
4. 10% will be deducted for this project if it is returned for resubmission

### Resource Requirements

- Python Programming Learning material on myLMS may be referenced.

### Delivery Requirements (evidence to be presented by students)

The project submission must include:

- Program Design
- Project source code
- User documentation
- Databases (if any)
- Resources (if any)
- System requirements (if any)
- Rough work (if any)

## **Plagiarism and Referencing**

Consult the section at the end of this document, which outlines how negative marking will be applied as well as the way in which it will affect the assignment mark.

---

## Section A

### Question 1

150 Marks

---

#### Scenario

You work for an industrial store, and management has requested you to write an application that customers can use at a till. The idea is that the customer can just hand his/her shopping list to a teller. The teller would then match all the items to a database on a server. After all the items have been added, an invoice is printed, and the customer can collect their items.

The system would have a server and client side applications. The server application should connect to a MySQL database called network\_store. The client machines request certain data from the server, which runs the MySQL database. Each customer must register before they can buy any items. This information is stored in a table called customers. The following must be stored in the customers table:

- custId          INT                          PRIMARY KEY                          AUTO INCREMENT
- fname          VARCHAR(40)          NOT NULL
- sname          VARCHAR(40)          NOT NULL
- address          VARCHAR(40)          NOT NULL
- phone          VARCHAR(10)          NOT NULL                          UNIQUE

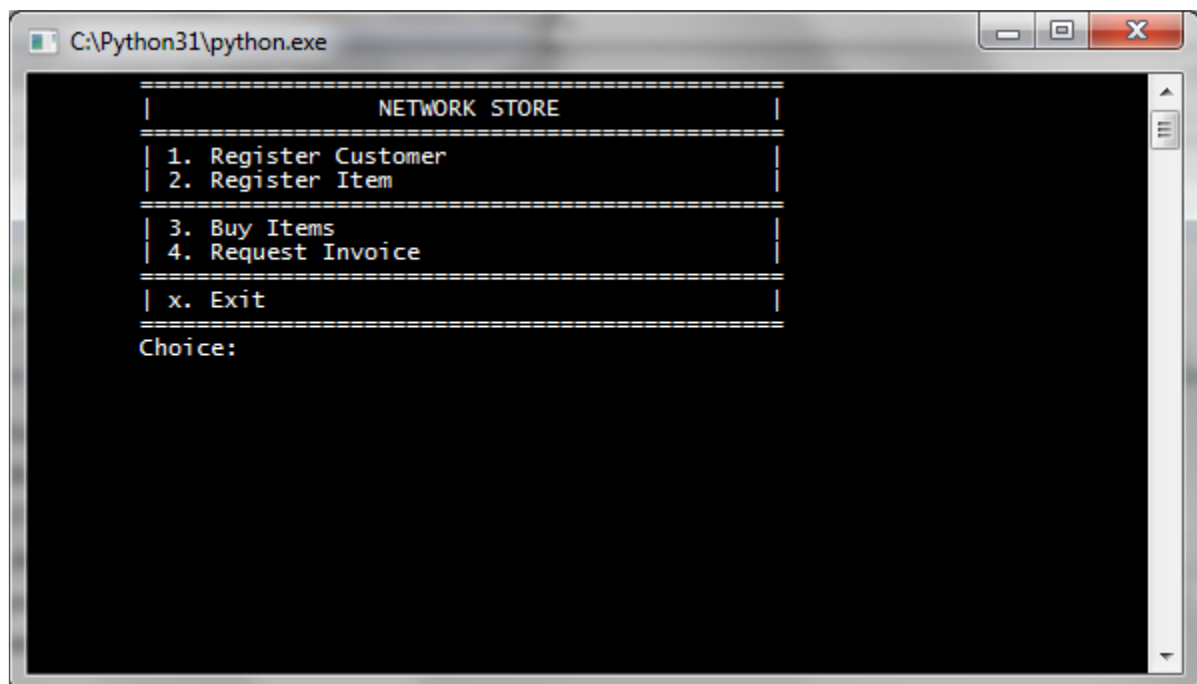
All items that customers can buy must be stored in a table called items. The following must be stored in the items table:

- itemId          INT                          PRIMARY KEY                          AUTO INCREMENT
- iname          VARCHAR(50)          NOT NULL
- descrip          VARCHAR(100)          NOT NULL
- price          FLOAT                          NOT NULL
- count          INT                          NOT NULL

A third table must be added to keep record of all transactions. The following should be stored to the invoice table:

- invoiceId      INT                      PRIMARY KEY                      AUTO INCREMENT
- custId          INT                      NOT NULL
- dateBought    DATE                      NOT NULL
- totalPrice     VARCHAR(10)           NOT NULL

The client should have a menu similar to the following example:



When the user chooses the **Register Customer** option, then the user must enter his/her name, surname, address and phone number. This information must then be sent to the server, and the server must add a new customer to the customers table.

When the user chooses the **Register Item** option, the application then prompts the user to enter the item's name, description, price and count (the amount of an item being added). This information must then be sent to the server, and the server must add a new item to the items table.

The itemId is a number that is incremented according to the sequence in which the items were entered

When the **Buy Items** option is selected, the teller requests the customer's name and the name and/or description of an item. The client application then sends a request to the server to find all

items that match the criteria that are in stock. The appropriate item is then chosen by the customer. The teller should continue adding items to the invoice until xxx is entered. An invoice is then written to a text file. The text file's name should be the invoiceId. The invoice should look similar to the following example:

```
=====INVOICE=====
Customer Name: Jan
Customer Surname: Visser
Invoice number: 15
=====
ITEMS
=====
14" Aluminium cutting blade
100 tooth non-ferrous metals
+++++
R356.06 X 1
-----
R356.06
=====
Prince Manufacturing Hydraulic PTO Pump
HC-PTO-2A NEW
+++++
R2732.07 X 3
-----
R8196.21
=====
Total: R8552.27
=====
```

When the **Request Invoice** option is selected, the teller is prompted to enter an invoice number. The application then sends a request to the server, which replies with a report that has been in a specific text file for each invoice.

---

End of Question 1

---

## Section B

---

### Plagiarism and Referencing

Eduvos places high importance on honesty in academic work submitted by students, and adopts a policy of zero tolerance on cheating and plagiarism. In academic writing, any source material e.g. journal articles, books, magazines, newspapers, reference material (dictionaries), online resources (websites, electronic journals or online newspaper articles), must be properly acknowledged. Failure to acknowledge such material is considered plagiarism; this is deemed an attempt to mislead and deceive the reader, and is unacceptable.

Eduvos adopts a zero tolerance policy on plagiarism, therefore, any submitted assessment that has been plagiarised will be subject to severe penalties. Students who are found guilty of plagiarism may be subject to disciplinary procedures and outcomes may include suspension from Eduvos or even expulsion. Therefore, students are strongly encouraged to familiarise themselves with referencing techniques for academic work. Students can access the Guide to Referencing on *myLMS*.

---

## Negative Marking

---

- At the discretion of the marker, if a student has committed plagiarism, an immediate 0% will be awarded for the project and 10% will be deducted from their next submission.