



Project 1 Specification

Faculty	Information Technology	Module Name	Python Programming
Module Code	ITPYA0	Project Number	1
Total Marks	150	Copy Editor	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

A video store has asked you to develop a system that will enable them to keep record of all their customers as well as their videos. This system would have a server and client side applications. The server application should connect to a MySQL database called video_store. The client machines request certain data from the server, which runs the MySQL database.

Each customer must register before they can hire out videos. This information is stored in a table called customers. The following details 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

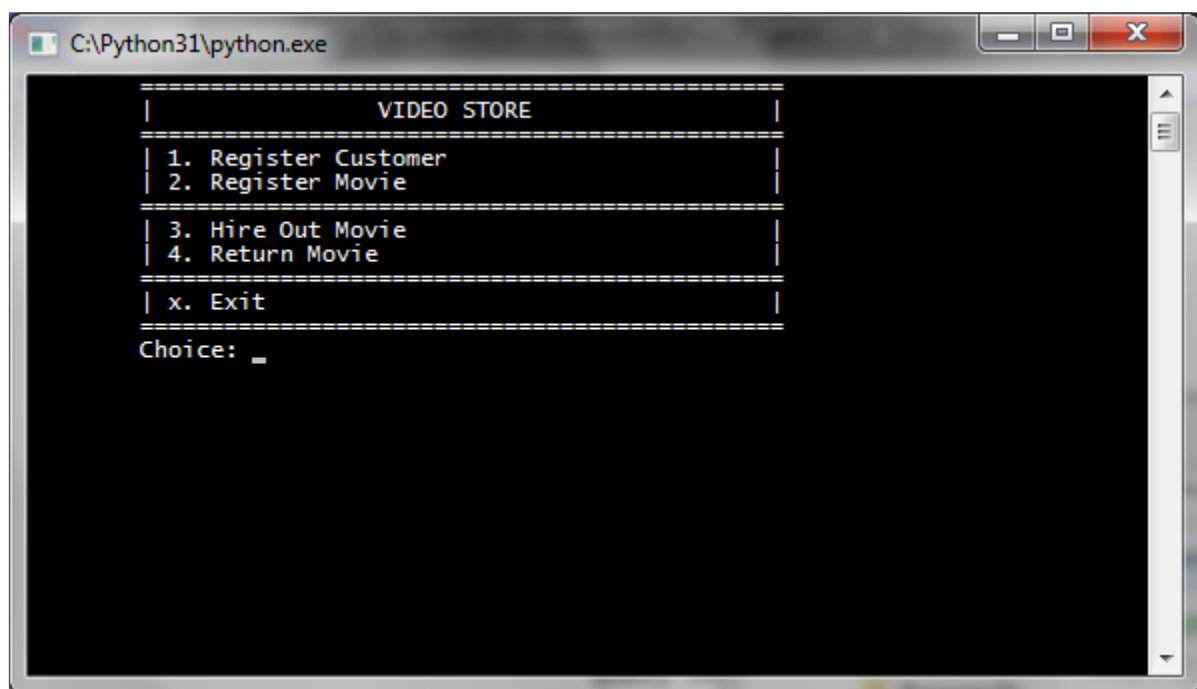
All available videos that customers can hire must be stored in a table called videos. The following detail must be stored in the videos table:

- videoId INT NOT NULL
- videoVer INT NOT NULL
- vname VARCHAR(15) NOT NULL
- type VARCHAR(1) NOT NULL
- dateAdded DATE NOT NULL

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

- custId INT NOT NULL
- videoId INT NOT NULL
- videoVer INT NOT NULL
- dateHired DATE NOT NULL
- dateReturn DATE

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



When the user chooses the **Register Customer** option, then the user must enter the customer's phone number, the server then checks that a person does not already exist in the customers table. If the customer already exists, a message should be printed to indicate this. If the customer is not contained within the database, the user must enter the customer's name, surname, and address. This information must then be sent to the server, and the server must add a new user to the customers table.

When the user chooses the **Register Movie** option, then the user must enter the movie name and type. This information must then be sent to the server, and the server must add a new movie to the videos table. The movie can only be one of two types:

- **Red box** – new movies, this is indicated by a 'R'
- **Black box** – old movies, this is indicated by a 'B'

The videoId is a number that is incremented according to the sequence in which the movies were entered; if movies have the same name, only the version number should increase.

Example:

```
mysql> SELECT * FROM videos;
```

videoId	videoVer	vname	type	dateAdded
1	1	Spider Man	R	2011-07-18
1	2	Spider Man	R	2011-07-18
2	1	Spy	R	2011-07-18
3	1	FBI	B	2011-07-18
1	3	Spider Man	R	2011-07-18
4	1	Grown Ups	B	2011-07-18
1	4	Spider Man	R	2011-07-18
4	2	Grown Ups	B	2011-07-18

8 rows in set (0.00 sec)

When the **Hire Out Movie** option is chosen, the application then prompts for the customer's phone number. The user then enters the videoId of the requested movie and the customer's details are then retrieved from the database. The movie would then be marked as hired out by adding it to the hire table. The dateHired field should be the system's current date.

The **Return Movie** option should request that the user enters the videoId. The hire table should then be updated to indicate that the movie was returned.

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.
- [If applicable, insert any penalties that are present on the marksheet]

Appendix

[Enter text here]