CPCS371: Computer Networks I **Fall 2019-2020**
Computer Science Department
Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah KSA.

Project: Student Book Sharing System (25 points)

- The deadline for this project is **Thursday, December 1, 2019**.
- This is a group project; each group should not contain more than 2 members.
- Any form of plagiarism will result in receiving zero in the project.
- WARNING: Late submission will not be accepted. Any project submitted after the cutoff time will receive zero.
- This project has to be submitted both in soft (CD) and hard (report) format
- Your package should include the source files, output screen shots and project report.
- You will be given 10 minutes to demonstrate your project. Questions asked will be related to the working of your project (program).
- This project worth 10% of the overall module marks (100%).

Project Description:

In class, we have discussed both connection oriented and connectionless protocols. In this project you need to develop a TCP-based network application called Student Book Sharing System (SBSS). Any student who want to share his book with other students. He needs to upload book information on the system. Other students can see the list of books on the system and they can reserve it.

For starters, SBSS will be non-graphical. You will only need to develop a command line client and server. The application will only support a limited number of features. The intention is to keep the application programming a little smaller, so that we can focus on the client/server aspects of this application.

All commands should terminate in a sentinel character (#). SBSS users will connect to a server that manages the user commands. Clients should be able to log onto the system, post book information, reserve the book, and log off. The application protocol has the following 7 basic requests that a user can make of the server:

1. **Create an account.** This is the first message that the server expects, so that it can establish an account for the user. When the client creates an account he is logged onto the system as well. A client can't log in without having first created an account and all functions require that the client be logged in first. Message format: 01;username;passwd# Possible server responses: 20;OK# [If the message is well formed and all elements are available, the account will be created and the server will return this message.] 21;Invalid user (or bad password)# [If some element of the message is missing or not properly delineated, the server will return this message.]	2. **Log on.** Once the client created an account, if he has logged off from the system, he can log back on. If he hasn't previously created an account, the logon fails. Message format: 02;username;passwd# Possible server responses: 20;OK# [If the message is well formed and all elements are available, and the account has been created, the user will be logged back on and the server will return this message.] 21;Invalid user (or bad password)# [If some element of the message is missing or not properly delineated,

3. Upload Book Information. If student want to share his book, He will upload the book information in the following format. Message format: 03;username, Bookname;# Possible server responses: 22;Information is uploaded successfully on the system # [System will generate unique book id and store the record in the database. The single record contains following attribute: BookID, name of book, owner, status (Available / Booked)] 23; Invalid format#	**4. View list of Books.** In a normal system, a client might want to know which of the books are available for sale. Message format: 04;LISTBOOKS# Possible server responses: 24;BookID, name, owner, status; Book ID, name, owner, status # The response will contain 0 or more books, depending on how many books are available for sale.
5. Generate Booking Request. Any student who is interested in reading the book, he should generate booking/reservation request in the following format. Message format: 05;username;BookID# Possible server responses: 25;Book has been reserved successfully # [After the reception of booking command, the status of book will change from 'Available' to 'Booked] 23; Invalid format# 25; This book is not available for reservation#	**6. Log off.** As long as the server doesn't crash, a student can log off and then log back on at a later time if he wishes. Message format: 06;username;passwd# Possible server responses: 26;username is logged off successfully# [The user will be logged off] 21;Invalid user (or bad password)# [If some element of the message is missing or not properly delineated, the server will return this message.]
7. Delete Account. A user can intentionally delete his account. The password is included to ensure that users can only delete their own accounts. Message format: 07;username;passwd# Possible server responses: 27;username# [The user account will be deleted] 21;Invalid user (or bad password)# [If some element of the message is missing or not properly delineated, the server will return this message.]	**Default:** If the message received by the server does not have a valid message code in the first two bytes of the message, or if the server is able to detect that the request is improperly formatted, the server should return an "invalid message type" message. Possible server responses: 29;Invalid message#

Program Requirements:

- Enough documentation (either embedded in the code with comments or external) to follow how you parsed the program and to follow what you expect to happen.
- Source code for your programs.
- A Sample output for your client. This can be generated by using the editing features of the command prompt window. If you click on the MSDOS icon in the upper left, you will get a drop down menu that includes an "edit" item. If you select "mark", you can then drag the cursor over everything that you want to keep. When you hit "enter" you will place that information into the clipboard. You can then paste it into a separate word file that you can email.