Dry cleaner pickup system:

In this assignment you are going to implement an application that will demonstrate the benefits of maps (Dictionaries in C#). The application should allow users to register the items that someone asks to be cleaned (laundry or dry cleaning) or repaired (shoes). By storing tickets in a Dictionary with the ticket number as key.

Below you see a possible GUI, although you are encouraged to make something more user-friendly.

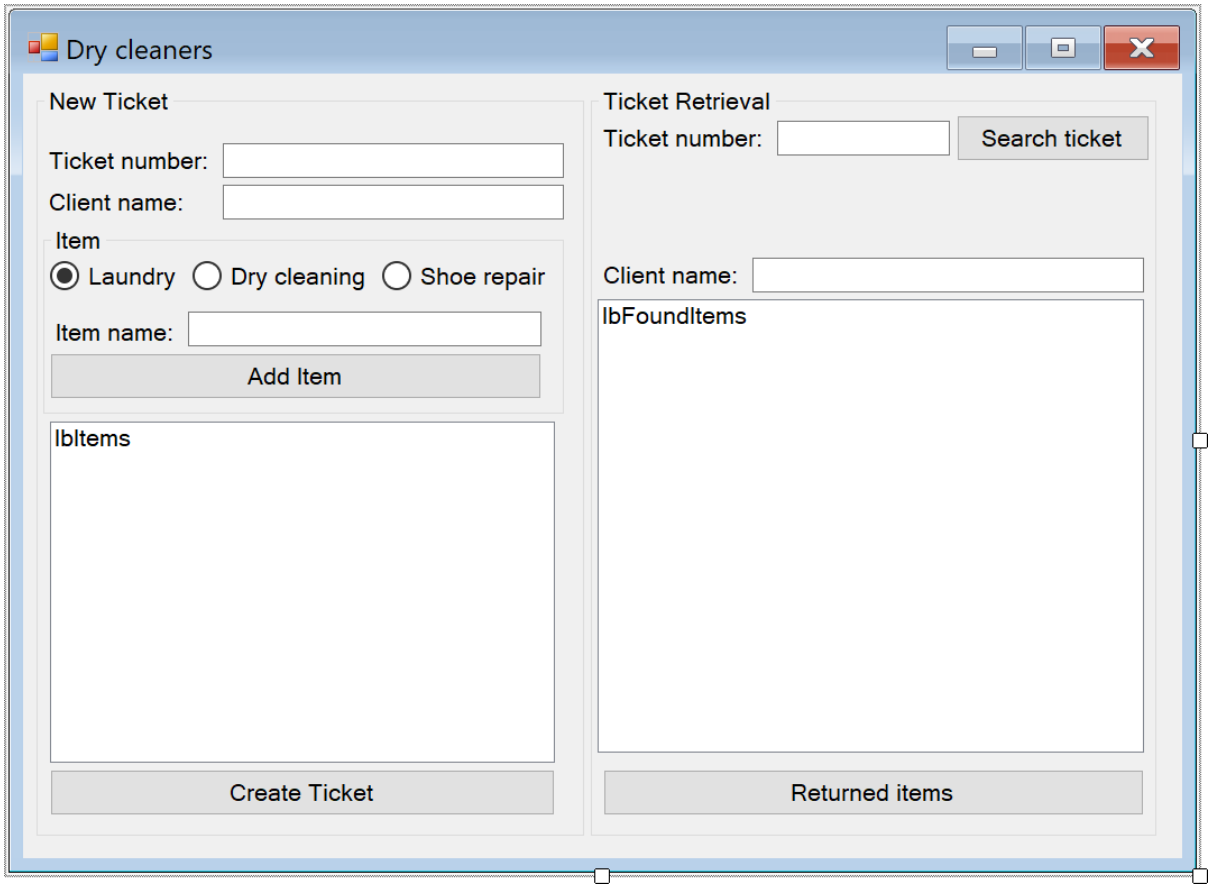

Figure 1: Possible graphical user interface for the dry cleaner form

The form is divided in two groupboxes:

- The left groupbox allows the user (dry cleaner) to create new tickets. This is done when a customer provides the clothes/shoes they want the dry cleaner to service (Laundry, dry cleaning or shoe repair). The user can then enter a ticket number and a client name. Per Item, the user can select a service (the radiobuttons), provide a descriptive item name, and add the item to lbItems (so to the ticket). Once all items are created, the user can press the "Create Ticket" button to complete the ticket. All input fields should be emptied and a new ticket can be created.
- The right groupbox allows the user to find a ticket. When the client comes to pick their items up, the user can ask for the ticket number and enter this at the top and press the button "Search ticket". If the ticket exists, the client name will be displayed in a disabled textbox and the items are displayed in the listbox below. The user can then gather the items and once they returned it to the client, press the button "Returned products".

Below you may find a UML class diagram and requirements that you can use as foundation to implement the code.

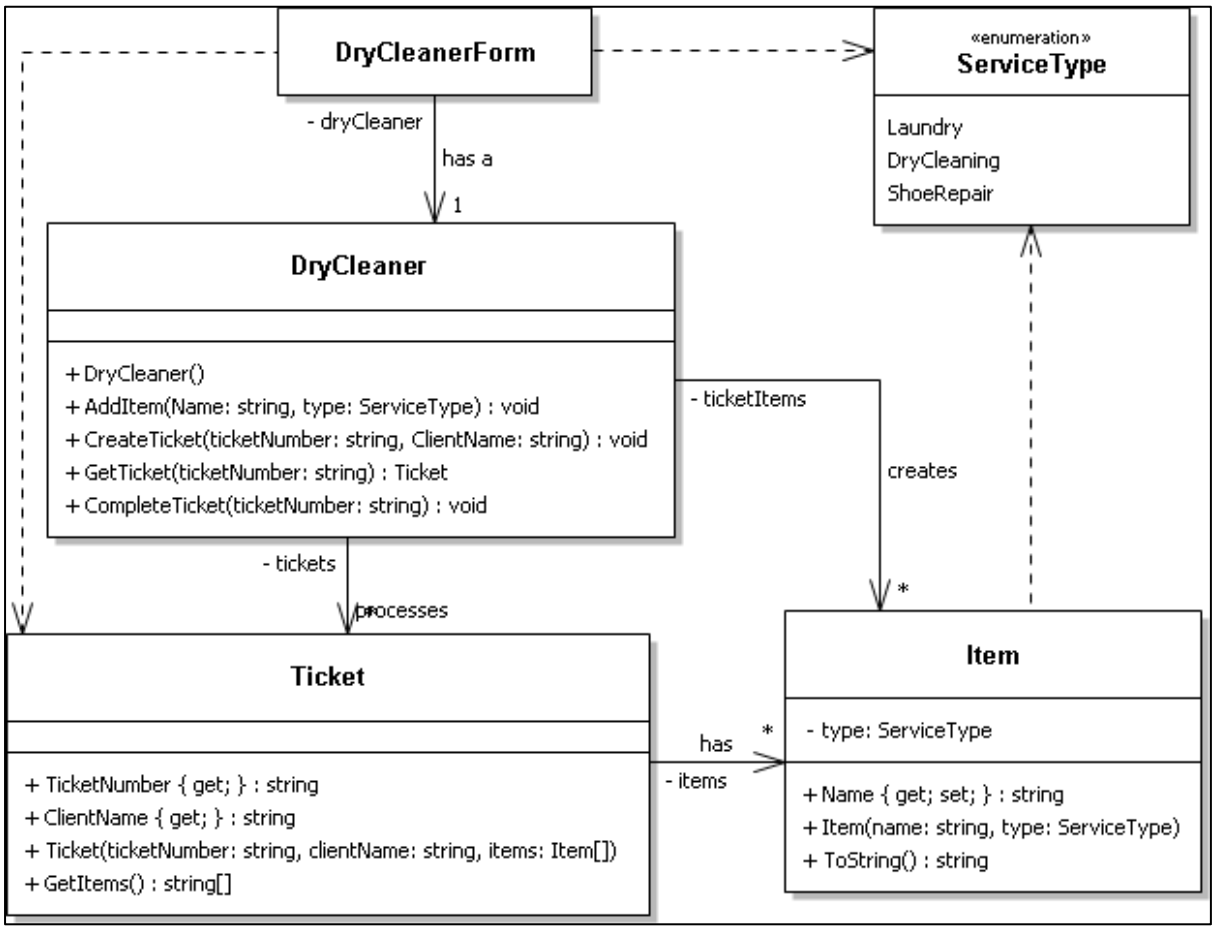

Figure 2: UML Class diagram for the dry cleaner application

When the user adds items, they are temporarily stored in the DryCleaner object (hence the "ticketItems" member). When the ticket is being created, the "ticketItems" is passed along as parameter.

Make sure that "tickets" in the DryCleaner is a map (Dictionary).

Requirements:

FR-01: Add new items to a ticket.
FR-02: Create a ticket.
If the ticket contains a ticket number, client name and at least one item.
FR-03: Search for a ticket by ticket number.
FR-04: Complete a ticket (return items).

Possible extension:
FR-05: Process individual items.
Before a ticket can be returned, the user would first have to tag the items as "serviced" (laundry done, dry cleaned or shoe repaired).