

Final Project Write-Up

Executive Summary

I own a large collection of movies; 336 to be exact. Friends/ward members from my apartment complex visit my apartment all the time asking to borrow one of them (my apartment is nicknamed “CraigBox”). I have no problem lending my movies out (I make sure everyone knows they are liable for any damage incurred), but one of the biggest problems I face is that I don’t keep any type of record of which person has which movie, so I often forget to follow up with people about returning them.

My project fixes this problem. I have created a video library check-out (and check-in) system which I can use to add movies to my Access database, keep track of which people have borrowed my movies (and update their status accordingly), and send notifications (via SMS text message or E-Mail) to people who have borrowed them.

Implementation

CraigBox consists of 4 main actions: adding a movie to the Access database, checking out a movie, checking in a movie, and sending notifications.

Adding a Movie to the Access Database

When the user clicks the “Add Movie To Database” button, a user form displays asking for information regarding the movie title, UPC code, and the type of movie (i.e., Blu Ray, DVD—the default, 3D Combo). When the user presses “OK,” the program checks to ensure there is a value for both the movie title and UPC code. The program also uses a function to ensure the UPC code entered is a valid UPC code and that the movie does not already exist in the database. Once the information has passed the test, a new record is inserted into the database, and after waiting a few seconds, the program initiates a message box letting the user know the record was inserted successfully.

Checking Out a Movie

When the user clicks “Use CraigBox,” a user form is displayed which asks for the UPC code of the desired movie. When the user clicks the “Check Out” button, the program checks to ensure there is a value in the box and uses a function to ensure the value is a valid UPC code. The program also ensures the UPC code exists by matching it to a UPC code in the database. Then, the UPC Code is compared to the “Checked Out” sheet to ensure the movie has not already been checked out.

Once these tests have been passed, another user form appears asking for the user’s information, like Name, E-Mail address, phone number, and wireless carrier (T-Mobile is the default). Since there are no land lines in the complex, it is assumed that each person owns a cell phone. Each field is checked to ensure the user has entered a value. The program checks to ensure the E-Mail is valid by looking for “@” in the middle of the string and a “.com” at the end of the string (I

decided not to allow “.net”—they are used less often—or “.edu” E-Mails—they expire soon after someone graduates). The phone number is also checked to ensure only numeric numbers have been entered and only 10 digits exist (dashes are stripped out).

The program then accesses the record associated with the UPC code and changes its status to “OUT” and inserts the information provided by the user. The updated record is then printed to the spreadsheet.

Checking In a Movie

When the user clicks “Use CraigBox,” a user form appears asking for the UPC code of the movie. The program again checks to ensure a value has been entered and uses a function to ensure that the UPC code is valid. Then, the UPC is compared to the list of movies that have been checked out to ensure that it has indeed been checked out (and notifies the user if it has not). Then, the program accesses the database and changes the status to “IN” and inserts NULL values for the user’s personal information. The information is then updated/(un)written to the spreadsheet.

Sending E-Mail and Text Notifications

Each respective button sends notifications via Microsoft Outlook to every user who has borrowed a movie. The program looks at each individual record contained in the spreadsheet and sends a message to the E-Mail (for texts, the phone number and carrier) supplied with the title and type of movie contained in the body of the message.

Learning & Difficulties

Most of the problems I encountered were due to syntax errors, but were easily solved by searching for the usual syntax online. However, probably the biggest error I faced was a logic error. While creating the “For” loop to generate messages to be sent to people who had checked out a movie, my variable, “i”, kept skipping a number (1 became 3 rather than 2 after the “Next” statement). I figured out the problem by setting a break point just before the end of the “Next” statement and cycled through the “For” loop one statement at a time. I realized I had been mixing up “For” loops with “Do Until” statements, as I had inserted the statement “i = i + 1”. “For” loops do not need this statement as they step through each integer defined by the user (i.e., Step 1 → the default, Step 2, etc.).

I also had some difficulties sending the text messages. Originally, I was going to use GMail to send the texts, but decided against it due to time constraints (I already had the E-Mail set up through Outlook and decided the Gmail approach was not realistic/true to my project as I would not send E-Mails and texts via different E-Mail accounts). I used the same code to send E-Mails, but had to make some changes.

Probably the biggest lesson I learned while completing this project was the importance of break points in how where my programs deviate from what I expect them to do. I also learned a lot just by searching the internet for ways that would help to solve the problems I was encountering. For example, I had a hard time figuring out why, when trying to add information about a new movie to the database, a message box popped up saying “The movie was not added.” My problem was that I was not giving the database enough time to add the record before checking to

see if it was successfully added (something I had never encountered before). While researching a similar problem online, I was able to solve a problem I didn't know I had.

Assistance

The only assistance I received was from Professor Allen during his office hours. Also, the “isValidUPC” function is basically the same function as the “isValidISBN” number I created in class (with a few tweaks here and there).