HW 5 Instructions

Create a user interface application which allows waiters and managers to login to a restaurant system. Waiters should be able to enter available shifts based on selected day and time. Managers should be able to solve an assignment problem with the additional waiter availability constraint. Managers should also be able to import new reservation text file and perform demand analytics.

Practice 17

Log-In Form Design

- Create a new userform and name it.
- Add a frame with two option buttons: Waiter and Manager. Name both option button controls.
- Add another frame with username label and textbox as well as password label and textbox. Set the password text box password character to “*”. Name both textbox controls.
- Add a checkbox in this second frame to allow users to show/hide password characters. Name this control.
- Add two command buttons to cancel or login. Name both controls.

Log-In Form Event Procedures

- Initialize the userform: set the waiter option button to true, manager option button to false.
- Click the checkbox: if checkbox is true, set password character to “”, otherwise set to “*”.
- Click the cancel button: close the form.
- Click the login button:
    - Check if the username is in the first column of the “LogInInfo” range, otherwise give error message.
    - If username is okay, then record the true password from the “LogInInfo” range and check if the password entered in the textbox is the same as the true password. If not, give error and reset password textbox.
    - If password is okay, then check if waiter option button was selected. If so, make sure that the user role from the “LogInInfo” range is also “Waiter”. Otherwise, tell user they should select “Manager” for given username.
    - If role is okay, then record waiter’s name from “LogInInfo” range and find the column number for this waiter from the first row of the “AvailTable” range. Then clear the contents for this column of the “AvailTable” range, then enter the waiter’s name in the first row of this column.
    - Close the form.
    - If the waiter option button was not selected (manager button selected), then close the form.

Navigation Form Design

- Create a new userform and name it.
- Add two command buttons to view reservations or view workshifts. Name both controls.

Navigation Form Event Procedures

- Click the view reservations button: show the reservations sheet and hide the welcome and workshifts sheets. Close the form.
- Click the view workshifts button: show the workshifts sheet and hide the welcome and reservations sheets. Close the form.
- Go back to the Log-In form event procedure for clicking the login button. If the manager button was selected, show the navigation form (after closing the login form).

Module Code: Log-In, Navigate, Exit Buttons

- Write a sub procedure for the login button on the welcome sheet to show the login form.
- Write a sub procedure for the navigation buttons on the reservations and workshifts sheets to show the navigation form.
- Write a sub procedure for the exit buttons on the reservations and workshifts sheets to show the welcome sheet and hide the current (active) sheet.

Practice 18

Waiter Availability Form Design

- Create a new userform and name it.
- Add a frame with a label and textbox for the waiter's name. Name the textbox control.
- Add another frame with a label and combo box for available day, label and combo box for available shift, and label and listbox for available time. Name the combo box for available day and set the row source to the "Day" range. Name the combo box for available shift and set the row source to the "Shift" range. Name the listbox for the available time.
- Add two command buttons to add availability or be done with the form. Name both command button controls.

Waiter Availability Form Event Procedures

- Initialize the userform: Tell the user that their availability for this weekend has been reset and that they should add available days and times in order to be scheduled.
- Change the available shift combo box: set the row source for the available time list box to the value of the available shift combo box.
- Click the done button: Lookup the column number for the waiter name from the first row in "AvailTable". Calculate total number of shifts as the sum of the values in this column number of the "AvailTable" range. Tell the user they are available for this total number of shifts. Close the form.
- Click the add availability button:
    - For each control on the form, if the name of the control is the name of the available day combo box, available shift combo box, or available time list box, then check if the value of the control is empty. If so, give an error message and exit the sub.
    - If the available day is "Friday" and the available shift is "Brunch" OR if the available day is "Sunday" and the available shift is "Dinner" then tell the user this shift is not available on the selected day.
    - Enter the available day combo box value and available time combo box value into the DGET criteria range "AvailCheck". Add a DGET function in the "AvailWS" range (on the workshifts sheet) to find the workshift number for the given day and time.
    - Record the waiter row as the "AvailWS" DGET value (workshift number). Find the column number for the waiter name from the first row in "AvailTable". Set the "AvailTable" value for this row and column to be 1. (Denotes that waiter is available at this day and time.)
    - Go back to the LogIn form code, in the click log in button event procedure: if the waiter option button was selected, then before closing the form, set the waiter name text box from the waiter availability form to be the waiter name found from the "LogInInfo" range. After closing the form, show the waiter availability form.

Practice 19

Reservations Sheet

- Enter formulas for Total Customers column to calculate the sum of number in party values if the day and time match the current workshift. Use range names for the referenced columns.
- Add data validation to the new reservation cells (this allows manager to enter any additional new reservations): New phone number should be a 10-digit number. Number in party should be between 1 and 8. Day should be from list of “Day” values.
- Enter a DGET formula for the new reservation check to look up the total customers for the new reservation day and time. If there is an error, make the cell blank.
- Insert two option buttons on the sheet for brunch or dinner. Name both controls.
- Insert a command button to add a new reservation and name this control.

Reservations Sheet Code

- Activate the worksheet: set the brunch option button to true and the dinner option button to false.
- Click the brunch option button: If the new day value is “Friday”, then tell the user no brunch on Fridays. Set the brunch option button to false and exit the sub. Otherwise, set the dinner option button to false, select the “NewTime” range and add a new validation rule for the user to select a time from the “Brunch” range. (Record a macro to see code to add list data validation.)
- Click the dinner option button: same as brunch option button, but check that no dinner on “Sunday” and set time list values to the “Dinner” range.
- Click add new reservation button:
    - For each cell in the “NewRez” range, make sure nothing has been left blank. If so, give error message and exit.
    - Check if the new party size value plus the total customers for the current day time (from the DGET function) will be greater than 30. If so, then tell user not enough availability to add this reservation and exit sub.
    - Otherwise, copy the “NewRez” range and paste to once cell below the last cell down from “ImportHere”. Then clear the contents of the “NewRez” range. Update the range names used in your total customers formula by dynamically re-naming the columns of customer reservations (relative to “ImportHere”).

Workbook Code

- Open the workbook: Make sure the welcome sheet is visible. For each worksheet in the workbook, if the name of the worksheet is not “Welcome” then hide the worksheet.

Workshifts Sheet: Assignment Model

- Add the following formulas to the assignment model:
    - Total waiters should be sum of each workshift row.
    - Total needed should be total reservations value (from reservations sheet) divided by 10 (round up).
    - Total workshifts should be sum of each waiter column.
    - Total number of waiters is the sum of the total waiters colulmn.
    - Total max shifts should be sum of each waiter availability column.
- Set up the Solver model with the following components:
    - Minimize total number of waiters by changing the “WaitDV” decision variable cells.
    - Decision variable cells are binary and should be <= “AllWaitAvail” waiter availability range values.
    - Total waiters assigned per shift should be >= total waiters needed per shift.
    - Total workshifts should be <= max shifts per waiter.
    - Use the Simplex method to solve this linear assignment model.
- Add a conditional formatting rule to highlight the row of both assignment and availability tables if the total waiters is less than the total needed.

Module Code: Solve Assignment Model

- Write a new sub procedure to solve the assignment model. (Make sure to add Tools > References > Solver.)
    - Clear the contents of the “WaitDV” decision variable range.
    - Run the solver model and record the solver solution message. Tell the user if no solution could be found.
    - If a solution could be found, then ask the user for the pay rate per shift. Calculate the total pay amount as this pay rate times the total number of waiters from the optimal solution. Tell the user the total number of waiters and the total optimal pay.

Practice 20

Module Code: Import New Data

- If there are already reservation values (if “ImportHere” is not empty) then clear the contents of the customer reservations table. (Do not need to clear if there are not any values in the table.)
- Give the user a message that they should select a text file with new reservation data. Use the first row of the text file as an example to tell the user how the data should be formatted.
- Prompt the user to select a file and record the file name in a string variable.
- Write the code (or record macro) to import the selected user text file. Make sure you do not adjust column width. Make sure you overwrite cells. Check your starting row value, delimiter values, and which columns to skip.
- Update the range names used in your total customers formula by dynamically re-naming the columns of customer reservations (relative to “ImportHere”).

Reservations Sheet: Pivot Chart

- Use the imported reservation data to insert a pivot table on the hidden sheet. Set rows as day then time. Set values to sum of total customers. Add a new item “Brunch” = sum of all brunch times and new item “Dinner” = sum of all dinner times. Filter the pivot table to only show “Brunch” and “Dinner” as the time values.
- Create a pivot chart from this pivot table and place it on the reservation sheet (move chart). Create a pie chart with data labels inside the chart (remove the “Friday Brunch” and “Sunday Dinner” values; remove the legend).
- Record a macro to select the pivot chart and “Refresh” the data. Copy this code to the bottom of the import data sub procedure. Also copy this code to the bottom of the reservations sheet code add reservation procedure.