**Samuel Windell:** All non permanent conversation should be used with the comment feature to draw immediate attention, and allow for immediate disposal of the text after the matter has concluded. This will keep the document less cluttered. If you don't know how to make a comment, first highlight any text related to the cause of your temporary conversation, and then press the "Add Comment" between "Insert Link" and "Insert Image" or press ctrl + alt + m.

To communicate directly with someone viewing the document at the same time as you, you can use the "Chat" feature. To use the "Chat" feature click the "Person with a message bubble above their head" button in the top right of the document.

**Team customs:**

**Finalized Team Customs: (Official)**

- Extra comments to clarify code and speed up the desk checking process.
- Extra spacing between functions and blocks to assist with readability.
- Two commits per week for consistent progress.
- Tabs (4 spaces) used for all indentations.

**Assigned Roles and Responsibilities: (Official)**

- **Samuel Windell:**
  **Role:** Team Leader
  **Responsibilities:** Assign roles, assign individual goals, assign group goals, pick up the slack wherever needed, submit all group work on time.

- **David Matz:**
  **Role:** Back End Developer
  **Responsibilities:** Develop functionality and stability for the UI, coordinate with front end development members.

- **Donivan Hawkins:**
  **Role:** Front End Developer
  **Responsibilities:** Develop UI layout, coordinate with back end development members.

- **Tracie Lindquist:**
  **Role:** Analyst and Data Manager
  **Responsibilities:** Thoroughly test and debug pull requested code to find logical errors or mishandled data, coordinate with the development team, and accumulate real data results.

- **Tyler Joerendt:**
  **Role:** Back End Developer
  **Responsibilities:** Develop functionality and stability for the UI, coordinate with front end development members.

**Project Summary Reference:**

For this final project, we propose the creation of a Customers database super class, a Vehicle database class, and a RepairMaintenanceOptions database subclass. The customer class holds all existing customers, their vehicle type and identification number, the amount of money that they have spent along with their maintenance and repair history, their phone number, and customer identification number. The Vehicle database class will contain all the vehicles operated on by the business. The RepairMaintenanceOptions database subclass will contain the different maintenance procedures and repair service per vehicle supplied by the business. There will be one window UI that supports searching existing customers with their name or phone number, adding new customers, printing customer information, vehicle options, and maintenance options, as well as exporting information to a text file.

**Priorities:**

**Module 3:**
(
Deadline 1: **Samuel Windell:** Assign all roles, and create their responsibilities. (100%)
Deadline 2: **Samuel Windell:** Finalize the Project proposal, and begin creating individual goals for team members starting in Module 4.

DeadLine 1: **David Matz:** Stay up to date with changes in the “Team_Collaboration” document and comply with needed input. (100%)
DeadLine 2: **David Matz:** Acknowledge the assigned role and responsibilities, Team Customs, and the Project Proposal. Sign off on MO3_Final_Project_Launch. (100%)

DeadLine 1: **Donivan Hawkins:** Stay up to date with changes in the “Team_Collaboration” document and comply with needed input. (100%)
DeadLine 2: **Donivan Hawkins:** Acknowledge the assigned role and responsibilities, Team Customs, and the Project Proposal. Sign off on MO3_Final_Project_Launch. (100%)

DeadLine 1: **Tracie Lindquist:** Stay up to date with changes in the “Team_Collaboration” document and comply with needed input. (100%)

DeadLine 2: **Tracie Lindquist:** Acknowledge the assigned role and responsibilities, Team Customs, and the Project Proposal. Sign off on MO3_Final_Project_Launch. (100%)

DeadLine 1: **Tyler Joerendt**: Stay up to date with changes in the “Team_Collaboration” document and comply with needed input. (100%)
DeadLine 2: **Tyler Joerendt:** Acknowledge the assigned role and responsibilities, Team Customs, and the Project Proposal. Sign off on MO3_Final_Project_Launch. (100%)
)

**Module 3 Conclusion:**

By the end of module 3 the group will have a plan for the final project proposal, will be familiar with the methods and customs of the group as far as coding and collaboration go, and will be prepared to hit the ground running in Module 4.

**Module 4:**
(
Deadline 1: **Samuel Windell:** Help the development team meet their deadlines.
Deadline 2: **Samuel Windell:** Help the development team meet their deadlines. Create deadlines for Module 5.

DeadLine 1: **David Matz:** Create the classes and their attributes. Design an elegant search process for both Customers attributes, Vehicles attributes, and VehicleRepairMaintenance attributes. (100%)
DeadLine 2: **David Matz:** canceled

DeadLine 1: **Donivan Hawkins:** Create a rough representation of the GUI layout. Create all of the essential elements. (100%)
DeadLine 2: **Donivan Hawkins:** canceled

DeadLine 1: **Tracie Lindquist:** Get ahead in your studies to make more time for future deadlines.
DeadLine 2: **Tracie Lindquist:** Get ahead in your studies to make more time for future deadlines.

DeadLine 1: **Tyler Joerendt:** canceled
DeadLine 2: **Tyler Joerendt:** canceled
)

**Module 4 Conclusion:**

By the end of Module 4 the group should have the essentials of the GUI, The classes being used, and a searching feature. This is the skeleton of our project. We will add more features in module 5 as well as troubleshoot any issues there may be.

**Module 5:**

(
Deadline 1: **Samuel windell,** Help the development team reach their deadlines.
Deadline 2: **Samuel windell,** Help the development team reach their deadlines.

DeadLine 1: **David Matz,** Make a matching algorithm that sends the matched data from the search field to the text field to show the users what the current options are based on what has been typed into the search field. (100%)
-modified RepairMaintenanceOptions to have another return function
-modified matchingData() and created updateSearch()
-also created my own branch and borrowed others files and worked in the same folder.
DeadLine 2: **David Matz,** Standby.

DeadLine 1: **Donivan Hawkins,** Standby.
DeadLine 2: **Donivan Hawkins,** Make GUI look more professional with a decorative appeal. Use colors to organize layout. (85% I can't figure out how to make it look more decorative but I did everything else, we just need the person making the search feature to replace my comment in the SearchGUI with the actual data)
DeadLine 1: **Tracie Lindquist,** Fill in database with real data. Test current working program and log results. Take summary and feedback from members on what was difficult and how it was overcome. Log information into the Testing_Log document.
DeadLine 2: **Tracie Lindquist,** wait for returning code to further test. Report findings in the Testing_Log document.

DeadLine 1: **Tyler Joerendt,** Standby.
DeadLine 2: **Tyler Joerendt,** Create a new way to add data to the database through the GUI.
)

**Module 5 Conclusion:**

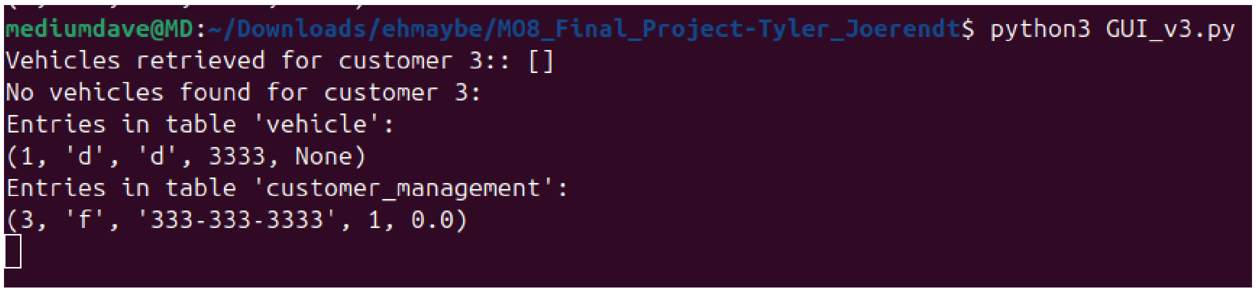
By the end of module 5, the team should have a roughly working project with an acceptable looking GUI layout, search feature and information display presenting the search results, a database holding all of the initial information needed to demonstrate functionality of the project, and a way to add data to the database through the GUI. This is a much more complete version of the project compared to module 4. The next week will consist of testing, reworking code, and organizing the project files.

**Module 6:**

**Announcement:**
Alright guys, this is the week. This week will make or break our project. I need 100% effort from everyone coordinating amongst each other to get this project testable by wednesday night. We can't miss this deadline. We are mostly there. By Wednesday I need… Updated versions of your files pushed to github repository. I need the files properly referencing the files they take data from. I need finished logic that completes the tasks I have discussed with you all. Tracie has a day to test and return results and problems. We continue this pattern to the end. I am sorry things have not worked out as I intended in Module 2, but as the old saying goes, no plan survives contact with the enemy. We have come a long way though, and although I have made slightly exaggerated conclusion predictions, we have not been very far off from intended goals. You should be proud of yourselves for contributing to this project for I believe that everyone has met expectations. I have seen great work from Donivan, Tyler, and David. They have done a great job at following the path I laid for success, and added their own creative input to make the project better than I could have done myself. I am sure Tracie is going to tie this up with a bow for us perfectly as I don't believe she is going to leave any detail unfinished. Our job for the future after this week will be giving her as much time as possible to perform her tasks. We do this by responding to her emails as fast as possible and returning code back to her with the needed fixes as fast as possible. The more time we can give her in these last weeks, the better everything will come together in the end. Good luck you all and thank you so much for your hard work and dedication to this project!

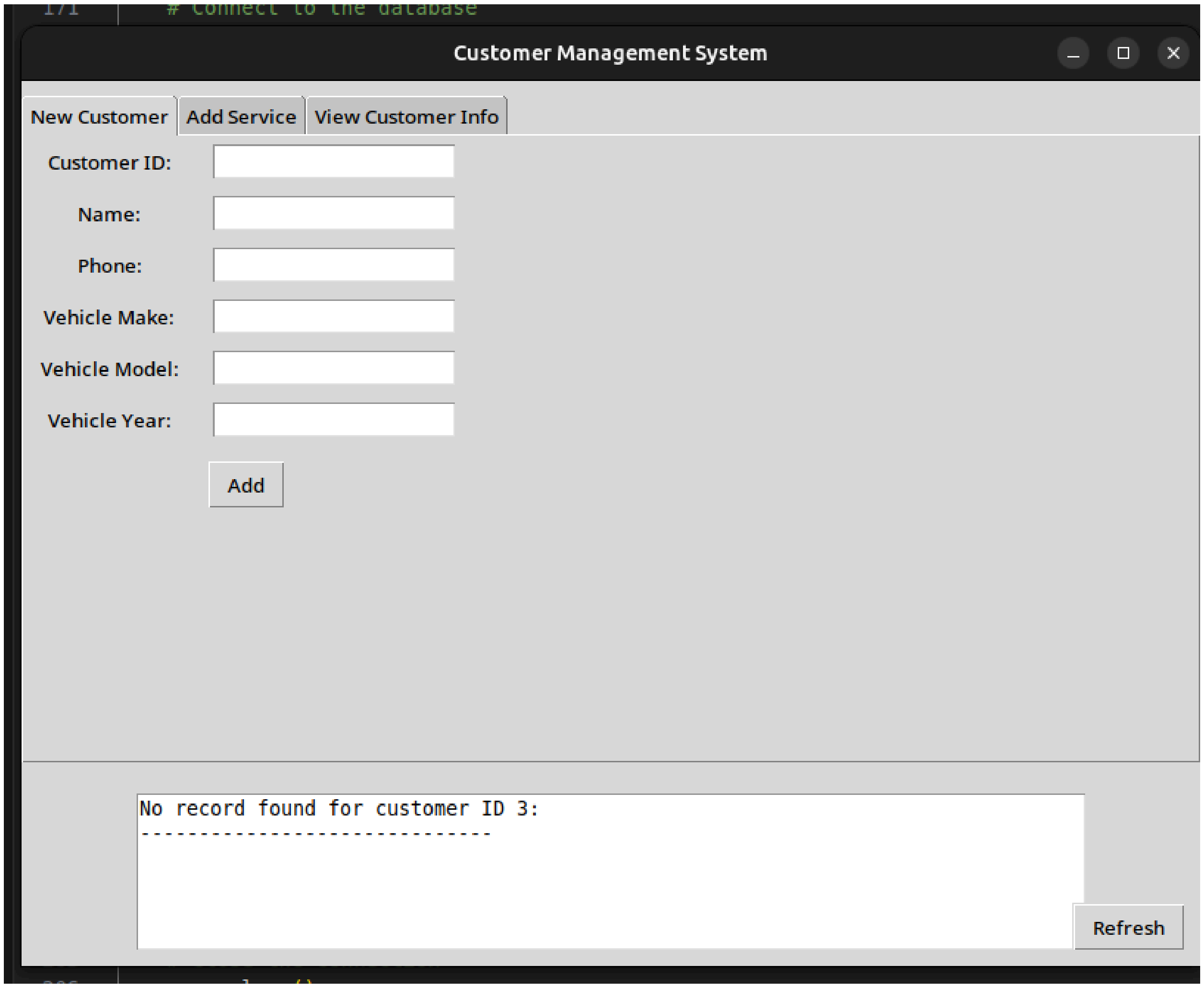
(
Deadline 1: **Samuel Windell:** Help the development team reach their deadlines.
Deadline 2: **Samuel Windell:** Help the development team reach their deadlines.

DeadLine 1: **David Matz:** Work with the team and get the files to be cohesive.
DeadLine 2: **David Matz:** Work with the team and get the files to be cohesive.



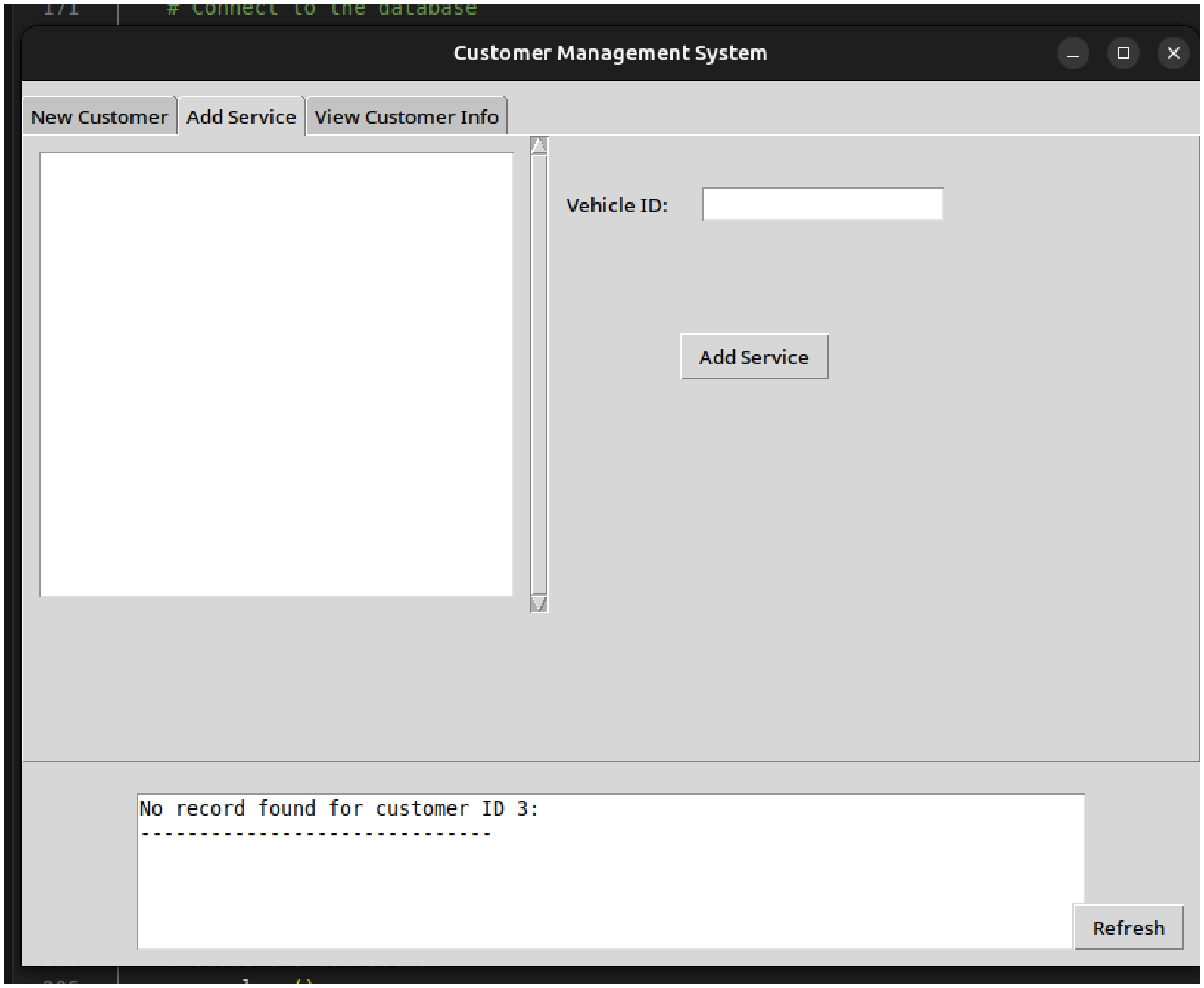
```
mediumdave@MD:~/Downloads/ehmaybe/M08_Final_Project-Tyler_Joerendt$ python3 GUI_v3.py
Vehicles retrieved for customer 3:: []
No vehicles found for customer 3:
Entries in table 'vehicle':
(1, 'd', 'd', 3333, None)
Entries in table 'customer_management':
(3, 'f', '333-333-3333', 1, 0.0)
```

The entry for vehicles is 1 and the vehicle id is 1 in the vehicle table, however the program isn't searching through the table correctly and I am uploading what I have to github and hopefully someone can look it over and figure this out.



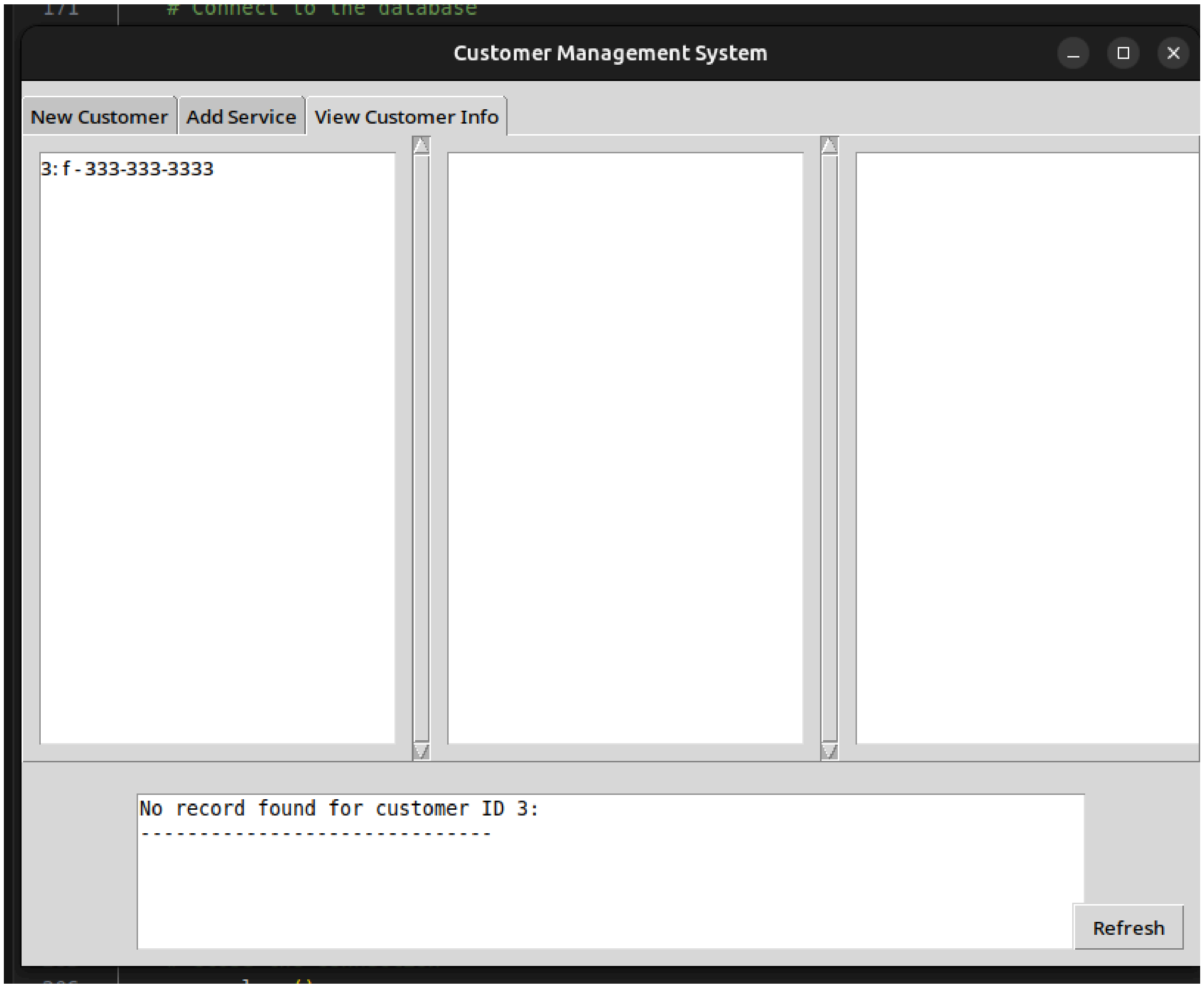
```
No record found for customer ID 3:
------------------------------
```

This adds customer info,



```
No record found for customer ID 3:
------------------------------
```

This gets all the information from repair table and displays them, when you have a vehicle id in place, it should add it to vehicle table entry

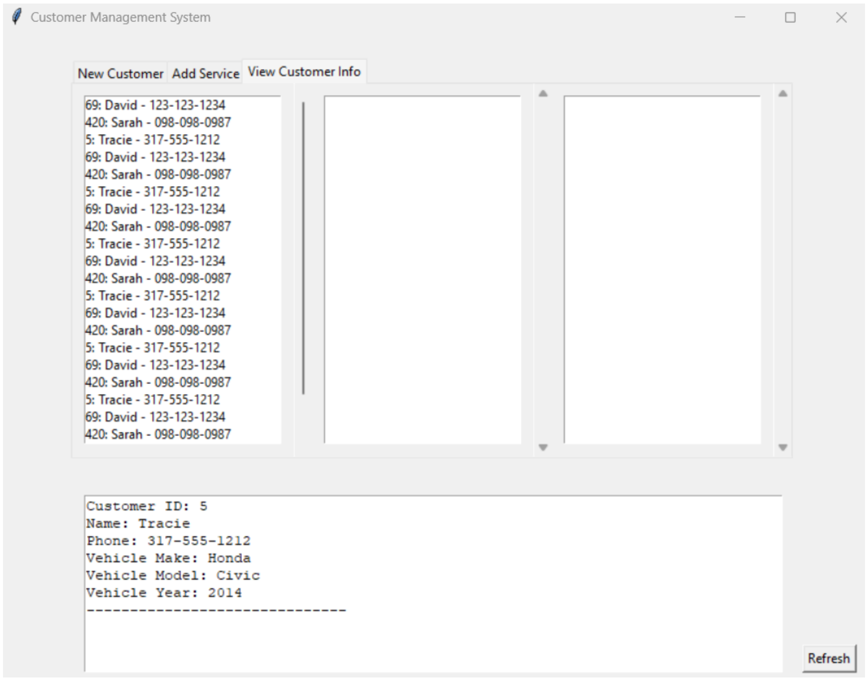


```
3: f - 333-333-3333
```

```
No record found for customer ID 3:
------------------------------
```

This shows list of customers, when selected: displays vehicle associated with customer, when vehicle is selected, it should display the repair option committed on the vehicle. The bottom displays total costs.

Update from initial tests run 7/13/2024:
Customer data is being duplicated in the customer info table. This is what I could see in the View Customer Info tab after I successfully entered new customer information.  There are six copies of every customer record.    -Tracie

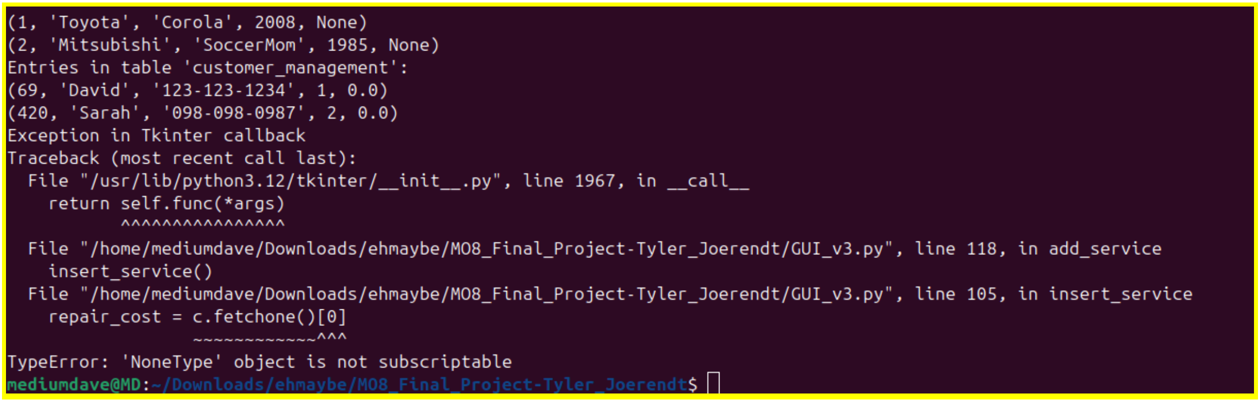


```
69: David - 123-123-1234
420: Sarah - 098-098-0987
5: Tracie - 317-555-1212
69: David - 123-123-1234
420: Sarah - 098-098-0987
5: Tracie - 317-555-1212
69: David - 123-123-1234
420: Sarah - 098-098-0987
5: Tracie - 317-555-1212
69: David - 123-123-1234
420: Sarah - 098-098-0987
5: Tracie - 317-555-1212
69: David - 123-123-1234
420: Sarah - 098-098-0987
5: Tracie - 317-555-1212
69: David - 123-123-1234
420: Sarah - 098-098-0987
5: Tracie - 317-555-1212
69: David - 123-123-1234
420: Sarah - 098-098-0987
```

```
Customer ID: 5
Name: Tracie
Phone: 317-555-1212
Vehicle Make: Honda
Vehicle Model: Civic
Vehicle Year: 2014
------------------------------
```

**Current Problems:**

- **When clicking on a customer, vehicle list cannot be populated as the search isn't working**
- **When adding service to car, Getting the error screenshot below:**
    - **This leads me to believe that I am not properly searching the database correctly, this could also be attributed to the vehicle search problem as well.**
    - **The repair list is updated from the repair table, so I know the repair option exists, but grabbing it doesn't work for some reason. I will pas this problem off to someone else to look at in the mean time.**



```
(1, 'Toyota', 'Corola', 2008, None)
(2, 'Mitsubishi', 'SoccerMom', 1985, None)
Entries in table 'customer_management':
(69, 'David', '123-123-1234', 1, 0.0)
(420, 'Sarah', '098-098-0987', 2, 0.0)
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.12/tkinter/__init__.py", line 1967, in __call__
    return self.func(*args)
           ^^^^^^^^^^^^^^^^
  File "/home/mediumdave/Downloads/ehmaybe/M08_Final_Project-Tyler_Joerendt/GUI_v3.py", line 118, in add_service
    insert_service()
  File "/home/mediumdave/Downloads/ehmaybe/M08_Final_Project-Tyler_Joerendt/GUI_v3.py", line 105, in insert_service
    repair_cost = c.fetchone()[0]
                  ~~~~~~~~~~~~^^^
TypeError: 'NoneType' object is not subscriptable
mediumdave@MD:~/Downloads/ehmaybe/M08_Final_Project-Tyler_Joerendt$
```

**UPDATE:**

- **I changed it so that one vehicle per customer and one maintenance option per vehicle. If more is done, it would be easier to make another database table for**

**service orders and I don't want us to have to recode a bunch of stuff as that would be something to add when the core program works correctly.**

- **I also fixed the refresh button to only update the repair list and customer list on the add service and customer info view tabs. This is because the vehicle/ repair list on the customer view tab are automatically updated when clicked.**
- **The repair table was updated after the commit in database creation, put the commit after so that the changes were made and the table parses correctly when displaying in the GUI (add service tab)**

==

DeadLine 1: **Donivan Hawkins:** Work with the team and get the files to be cohesive.
DeadLine 2: **Donivan Hawkins:** Work with the team and get the files to be cohesive.

DeadLine 1: **Tracie Lindquist:** Work with the team and get the files to be cohesive.
DeadLine 2: **Tracie Lindquist:** Test project, gather testimonials, report to the Testing_Log document. Send issues back to the team. (Get on this as soon as possible for I need some kind of result for the MO6 discussion.)

DeadLine 1: **Tyler Joerendt:** Work with the team and get the files to be cohesive.
DeadLine 2: **Tyler Joerendt:** Work with the team and get the files to be cohesive.
)

**Group tasks**: clean out the repository, separate tasks into individual files (GUI, GUI_to_Database_Translator, Database_to_GUI_Translator*, Database), create cohesion amongst all files, everyone sign off on desk check.

**Module 6 Conclusion:**

By the end of module 6, the project will be fully ready to test. Bugs and problems will surely exist, but we will have the last remaining weeks to make it run correctly.

**Module 7:**

**Announcement:**
I would like everyone to take a moment and leave a comment critiquing the way I have approached the execution of my role. Do you think I have been doing a good job, or have I been minimally meeting your expectations? I would like your thoughts on what you agree with and disagree with on how I have structured our workload and collaboration methods I have provided and encouraged for the group. I have wanted to be more embedded in the team's coding progress, but I find it hard to fit into your assigned work without being more trouble than it's worth. Even when I directly try to influence specific tasks they tend to rub with your individual

interpretations of the route to success. This happened when I had Donivan investing more time into something that didn't end up working for the cohesion the group needed. I feel bad about this since you all are so busy and your time spent is very valuable.

**Tracie Lindquist:**

- Hi, Sam. As far as your leadership goes, you've done a good job driving forward progress for the project. I cannot speak for the rest of the team, but I can say that I've felt a bit of the loop for most of the time. I know how difficult it is to juggle everyone's schedules, but I think if you had tried, we could have found a time that would have worked for everyone to meet once a week. I think that would have helped everyone get on the same page and would have helped us divvy up tasks more evenly. We also would have been able to identify each other's strengths and weaknesses early on. For example, when I said I didn't want to code, that didn't mean I didn't want to/couldn't contribute to anything other than testing. Turns out, I'm a database and architecture whiz and could have helped with designing the structure of the app early on, then we could have handed clear directives to the back and front end coders. And I would have been present to document everything. That would have been a win-win.

  I also think our chosen communication tool hampered more than it helped. Trying to discuss things through text communication ONLY is far more difficult than trying to talk it through over a Teams or Zoom or Discord call. Things can easily get taken out of context and misunderstood and feelings can get bruised when tone is inferred by the receiver that wasn't intended by the sender (sorry; comms undergrad back in the day, can't help myself).

**Donivan Hawkins:**

- I think you have done a wonderful job as the group leader and keeping us all on track. I am glad you were able to put so much effort into making this whole project go as smoothly as possible

**David Matz:**

- I think you did a fantastic job taking charge and making sure everyone is on board. The google docs as a means of communication is fine, but the loads of comments and keeping tack can sometimes be a bit hectic. I usually use discord as you can have many different text/voice channels and having at least two different means of communication can help keep things organized. Like a google docs to help keep track of tasks and comments on jobs, and some communication tool like discord to keep track of progress updates and more technical details. I know physical meetings were not really possible as I work quite a bit of overtime and others schedules do not really line up, but I think we did quite well.

**Name:**

- **Comment**

(

Deadline 1 & 2: **Samuel Windell:** Help the development team meet their deadlines.

DeadLine 1 & 2: **David Matz:** Bridge Tyler and Donivan together creating a three stage development process. Strategize (Take the existing problems delivered by Tracie and come up with solutions for these problems. This stage should consist of discussions and small scale experiments), Implement (Take the concepts from the small scale solutions and apply them to the project itself. Continue discussion ensuring the code remains cohesive.), Push (After a final desk check making sure the code is as it was planned to be, push the changes to the main branch in the repository for tracie to quickly download and test.)

Deadlines 1 & 2: **Donivan Hawkins:** Participate with David and Tyler in making bug fixes.

Deadlines 1 & 2: **Tracie Lindquist:** Return fixed code with new problems and potential solutions as soon as possible.

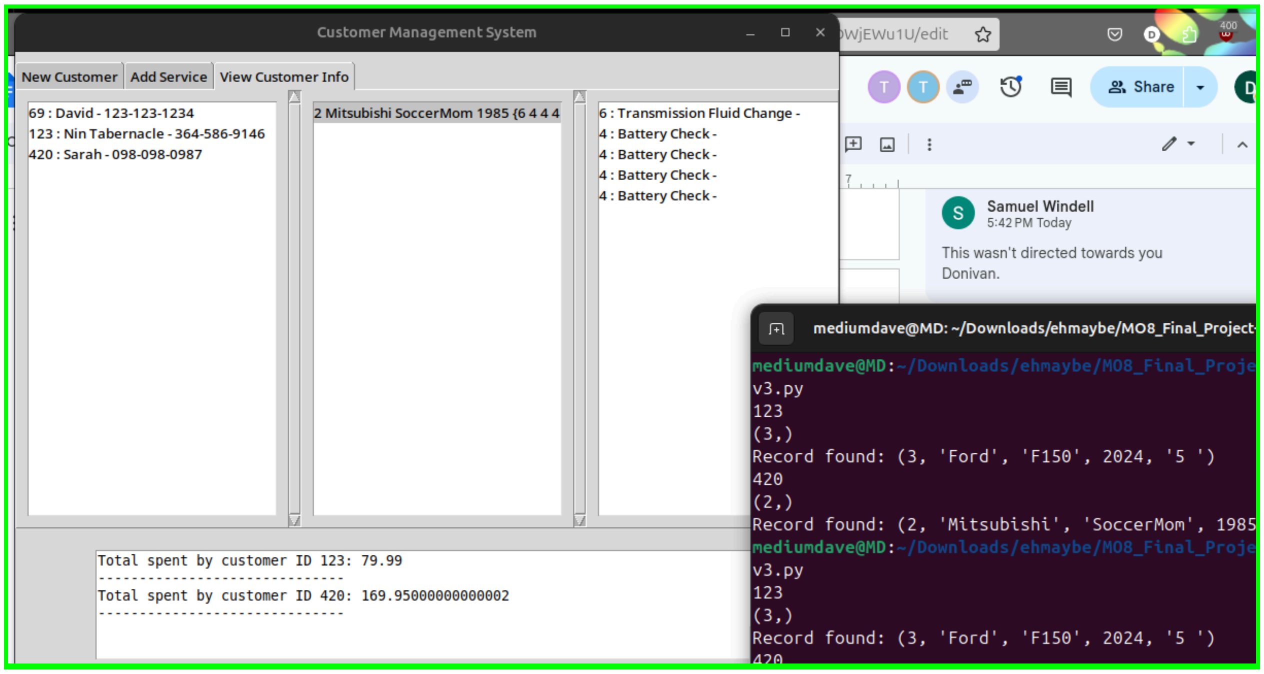
Deadlines 1 & 2: **Tyler Joerendt:** Participate with David and Tyler in making bug fixes.
)

**Problems and Potential Fixes:**

- One issue that could be fixed is in the view customer info. When hitting refresh it doesn't clear the old info in the text box.
    - Having it cleared and then displaying the text should fix that issue so you don't keep seeing the same information in that box every time you refresh.
- Also, I might need some clarification on the vehicle ID because I don't understand how the program has the vehicle ID stored if it's never put in. Or do we still need to create that?     ss
- Does the refresh button need to be on every tab because it only seems to be used on the view customer info tab so this may lead to confusion when it comes to the final product.
    - Suggestion the refresh button on the other tabs could clear the text box fields so that new info may be inputted. The biggest issue for this obviously is that'd have to be coded which I am currently not knowledgeable about however I can look into that.
- Need to update the phone GUI so that whoever inputs the information is aware that the format is XXX-XXX-XXXX.
- On line 126 we have 2 variables that aren't declared according to Python and it looks like this is due to it being outside of `def insert_service() -> None:` However the variables above it are in the same situation except they are recognized and I'm not sure why at first glance.
- Then do we need to display the vehicle table in the view customer info within one of the boxes? Cause the database is storing the information that I input into it however it's not displayed in the GUI at all.

    UPDATE (DAVID Week 7 tuesday):

Got vehicles to print in the list
Repairs working now
Third listbox displaying repairs, also can have multiple repairs now per vehicle



```
69 : David - 123-123-1234
123 : Nin Tabernacle - 364-586-9146
420 : Sarah - 098-098-0987
```

```
2 Mitsubishi SoccerMom 1985 {6 4 4 4
```

```
6 : Transmission Fluid Change -
4 : Battery Check -
4 : Battery Check -
4 : Battery Check -
4 : Battery Check -
```

```
Total spent by customer ID 123: 79.99
------------------------------
Total spent by customer ID 420: 169.95000000000002
------------------------------
```

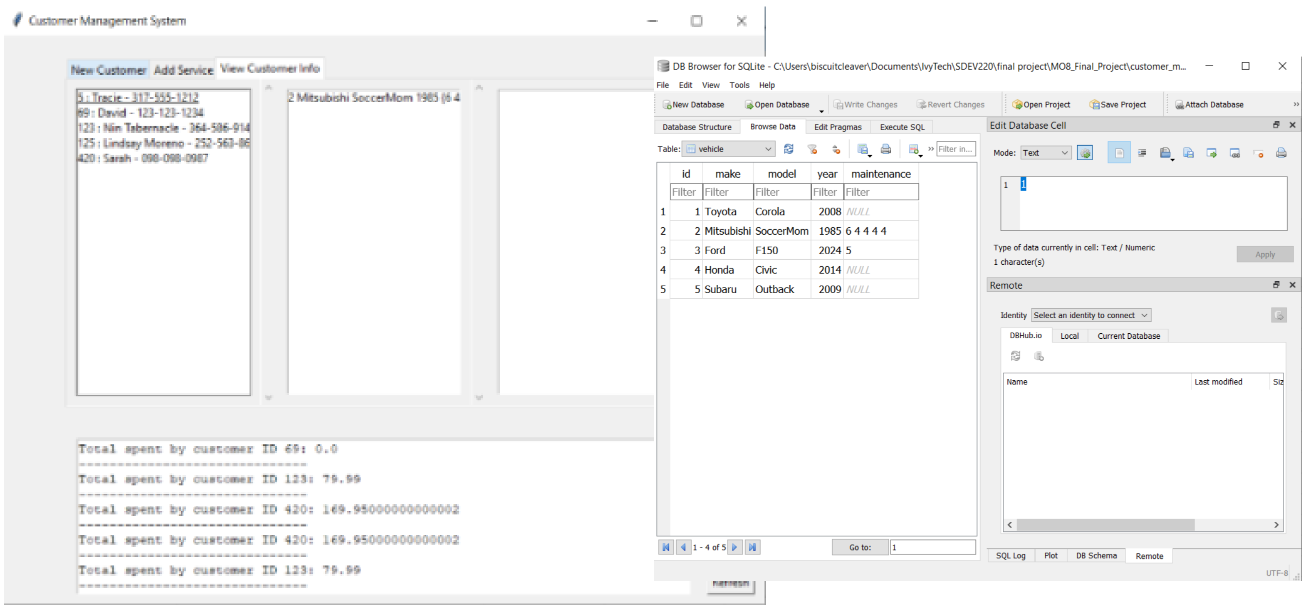
Samuel Windell
5:42 PM Today
This wasn't directed towards you Donivan.

```
mediumdave@MD: ~/Downloads/ehmaybe/MO8_Final_Project
mediumdave@MD:~/Downloads/ehmaybe/MO8_Final_Proje
v3.py
123
(3,)
Record found: (3, 'Ford', 'F150', 2024, '5 ')
420
(2,)
Record found: (2, 'Mitsubishi', 'SoccerMom', 1985
mediumdave@MD:~/Downloads/ehmaybe/MO8_Final_Proje
v3.py
123
(3,)
Record found: (3, 'Ford', 'F150', 2024, '5 ')
420
```

Left refresh button where it is as it updates both add service listbox and customer list.
Refresh button now properly deletes and relists service/customer lists
XXX-XXX-XXXX now displayed next to phone input on new customer
Issue with 126 variables not declared fixed

7/21/2024:
When entering new customer information, a vehicle ID is automatically generated for the vehicle added and the information is captured in the vehicle database, which means the PK/FK relationships in the .db are working correctly. However, new vehicle database information isn't appearing in the second column of the view customer information tab when the refresh button is clicked. Only one vehicle is showing in my View Customer Information tab, no matter how many times I refresh it, even though there are 5 vehicles currently available in my .db.
Note: I'm working in a local copy of the repo.

The vehicle list updates every time you click on a customer -David
Okay, that makes sense. It still makes it confusing to add service information because you have to flip back and forth between screens, but I'll make it work! -Tracie



```
5 : Tracie - 317-555-1212
69 : David - 123-123-1234
123 : Nin Tabernacle - 364-586-914
125 : Lindsay Moreno - 252-563-86
420 : Sarah - 098-098-0987
```

```
2 Mitsubishi SoccerMom 1985 (6 4
```

```
Total spent by customer ID 69: 0.0
--------------------------------
Total spent by customer ID 123: 79.99
--------------------------------
Total spent by customer ID 420: 169.95000000000002
--------------------------------
Total spent by customer ID 420: 169.95000000000002
--------------------------------
Total spent by customer ID 123: 79.99
--------------------------------
```

| | id | make | model | year | maintenance |
|---|---|---|---|---|---|
| 1 | 1 | Toyota | Corola | 2008 | NULL |
| 2 | 2 | Mitsubishi | SoccerMom | 1985 | 6 4 4 4 4 |
| 3 | 3 | Ford | F150 | 2024 | 5 |
| 4 | 4 | Honda | Civic | 2014 | NULL |
| 5 | 5 | Subaru | Outback | 2009 | NULL |

7/21/2024:
Existing service information isn't getting cleared from the bottom of the form on the add service and view customer information pages.
At least on the Add Service tab, should it be blank on entry and only display the vehicle ID and totals for the current session?

This could be an easy fix by deleting the log entrys with the refresh button. -David.

7/21/2024
I'm struggling with the service entry tab.
When a new customer is created, you enter their name and vehicle information.  A new vehicle entry is made in the vehicles table at the same time. However, because the customer ID is generated by the user, and the Auto ID field in the vehicles table assigns the vehicle a sequential number, the vehicle ID is different number than the customer ID. When you go to enter service information, you have no idea what vehicle ID to use for a specific vehicle, because there's no list of available vehicles on that tab and their associated customer names.

The form on the Add Service tab should probably work this way:  start with a select a customer field, once a number is entered, the vehicles associated with that customer should be available in a dropdown menu labeled: select vehicle.  Only after the customer AND the vehicle are identified should the service be selected. The total for that vehicle should tally in the field at the bottom of the screen and should clear when the refresh button is clicked.

I figured you could go to the customer view tab click on a customer and see their vehicle and then you can go back to the service tab and type in the proper number. -David
Yep, figured that out.  I'll make it work.  Way easier than making y'all recode a bunch of stuff.  :)
I should be finished with my testing today, then.  Things are looking good, now that I understand how it's supposed to work!    -Tracie

I retested this.  I entered two services (oil change (1) and tire rotation (2))  for the vehicle in my local version that was assigned ID 5.  It did add the services to the vehicle database and the total costs to the customer database, as expected and it is reflecting the selections in the View Customer Info tab. But on the Add Service tab, I have to scroll all the way to the bottom of the list in the display box to try to find the new entry.  When I do, I see the services listed, but there is no cost information associated with the entries (images on next page).

**Module 7 Conclusion:**
By the end of module 7, the project should be at least 2 bug fixes from completely finished. I can't imagine more than 4 separate issues needing a full deadline's worth of work. Luckily that's all the time we will have.

**Module 8:**
(
Deadline 1 & 2: **Samuel Windell:** Help the development team meet their deadlines.

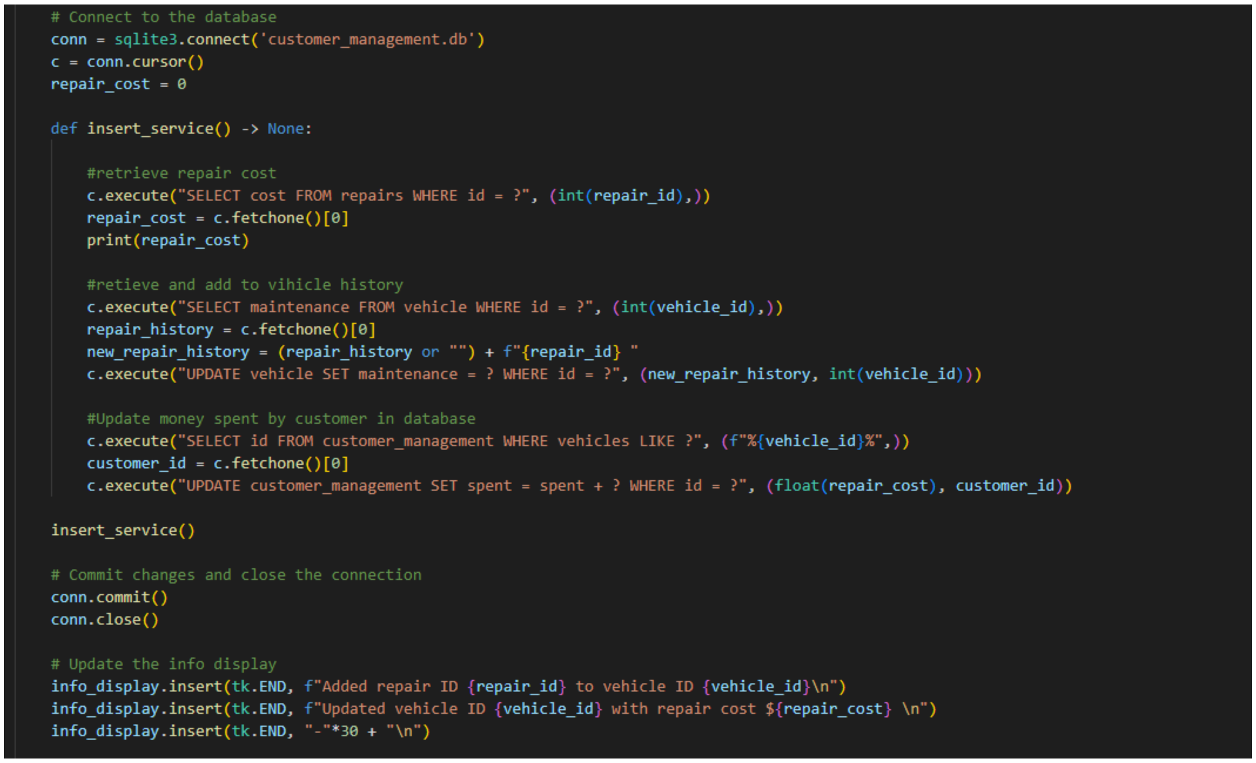
Deadline 1 & 2: **Donivan Hawkins:** Conclude the project with the developers.

DeadLine 1 & 2: **David Matz:** Conclude the project with the developers.

DeadLine 1: **Tracie Lindquist:** Work on finalizing the "User_Manual", "Testimonials", and "SDEV220_10PM08_FinalProjTestForm" documents.
Deadline 2: **Tracie Lindquist:** Test and make final changes to the "SDEV220_10PM08_FinalProjTestForm" document. Finish all documents for submission.

Deadline 1 & 2: **Tyler Joerendt:** Conclude the project with the developers.
)

**Problems and Potential Fixes:**

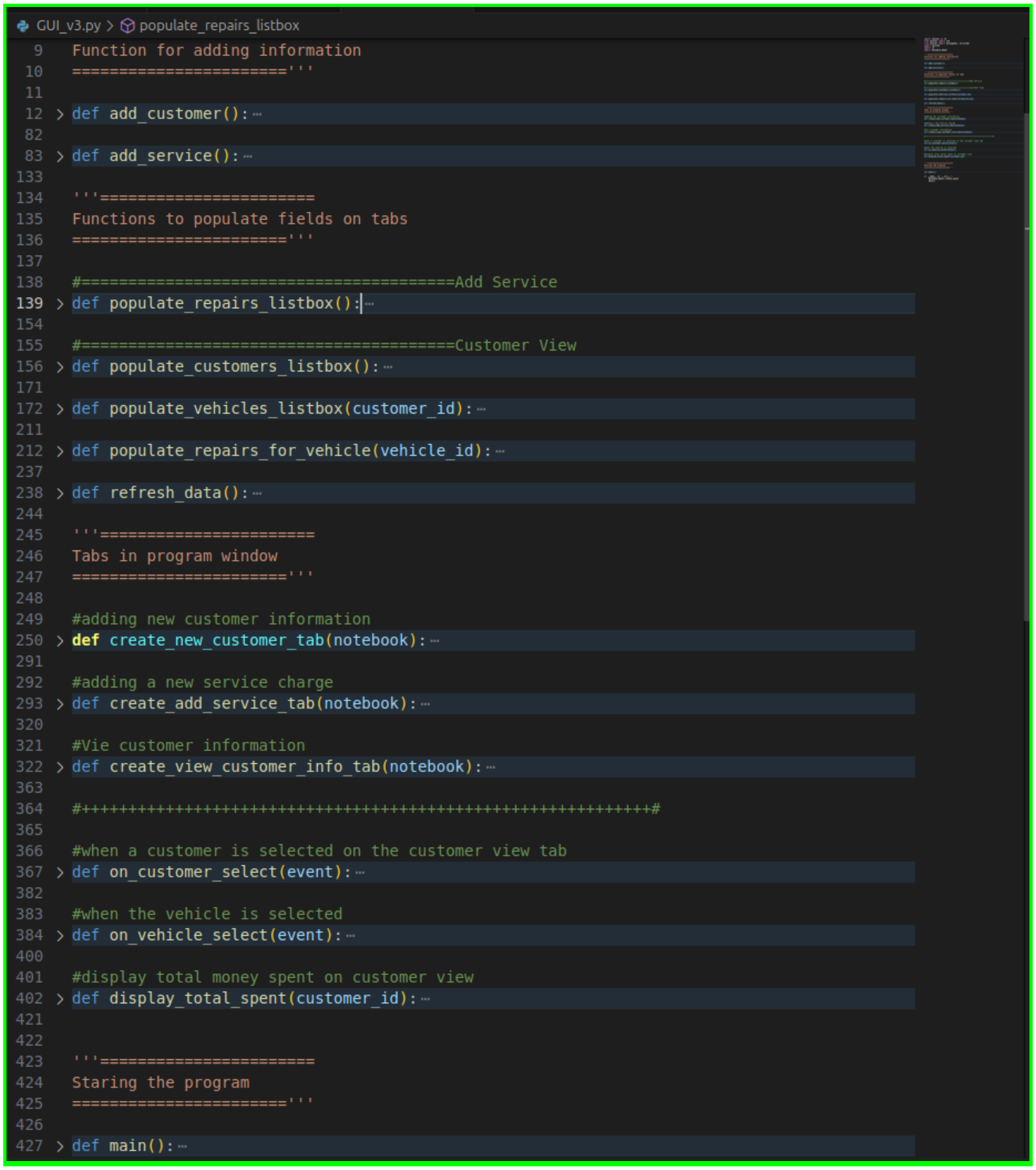


```
# Connect to the database
conn = sqlite3.connect('customer_management.db')
c = conn.cursor()
repair_cost = 0

def insert_service() -> None:

    #retrieve repair cost
    c.execute("SELECT cost FROM repairs WHERE id = ?", (int(repair_id),))
    repair_cost = c.fetchone()[0]
    print(repair_cost)

    #retieve and add to vihicle history
    c.execute("SELECT maintenance FROM vehicle WHERE id = ?", (int(vehicle_id),))
    repair_history = c.fetchone()[0]
    new_repair_history = (repair_history or "") + f"{repair_id} "
    c.execute("UPDATE vehicle SET maintenance = ? WHERE id = ?", (new_repair_history, int(vehicle_id)))

    #Update money spent by customer in database
    c.execute("SELECT id FROM customer_management WHERE vehicles LIKE ?", (f"%{vehicle_id}%",))
    customer_id = c.fetchone()[0]
    c.execute("UPDATE customer_management SET spent = spent + ? WHERE id = ?", (float(repair_cost), customer_id))

insert_service()

# Commit changes and close the connection
conn.commit()
conn.close()

# Update the info display
info_display.insert(tk.END, f"Added repair ID {repair_id} to vehicle ID {vehicle_id}\n")
info_display.insert(tk.END, f"Updated vehicle ID {vehicle_id} with repair cost ${repair_cost} \n")
info_display.insert(tk.END, "-"*30 + "\n")
```

**Ideas for a solution:**
**Repair_cost doesn't seem to be getting manipulated at all to change so it is always set to zero. My idea is maybe we need to get the cost from the database. - Tyler (7/22/2024)**

The repair_cost variable is set to 0 and doesn't change at all because it's not associated with the def. So the variable will always be zero since the two repair_cost variables have nothing to do with each other.

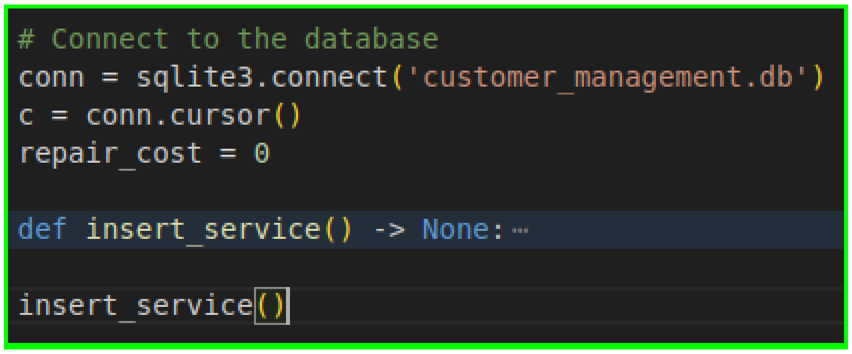
The repair cost variable is only initialized to zero… It is set in the line where the comment says #retrieve repair cost. It finds the cost portion of the repairs id and sets it to repair cost. It then uses that to add to the total spent by the customer. That is how the spent part of the customer goes up
The repair cost isn't supposed to be associated with the function, as the function is being used to break up the code so it can be minimized for readability. The function is called immediately after being created.



GUI_v3.py > populate_repairs_listbox

```python
Function for adding information
======================='''

def add_customer():...

def add_service():...

'''=======================
Functions to populate fields on tabs
======================='''

#========================================Add Service
def populate_repairs_listbox():...

#========================================Customer View
def populate_customers_listbox():...

def populate_vehicles_listbox(customer_id):...

def populate_repairs_for_vehicle(vehicle_id):...

def refresh_data():...

'''=======================
Tabs in program window
======================='''

#adding new customer information
def create_new_customer_tab(notebook):...

#adding a new service charge
def create_add_service_tab(notebook):...

#Vie customer information
def create_view_customer_info_tab(notebook):...

#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

#when a customer is selected on the customer view tab
def on_customer_select(event):...

#when the vehicle is selected
def on_vehicle_select(event):...

#display total money spent on customer view
def display_total_spent(customer_id):...


'''=======================
Staring the program
======================='''

def main():...
```

The code is designed to be easy to read/understand by having things minimizable and that is why nested functions like insert_service are created inside functions and immediately called



```
# Connect to the database
conn = sqlite3.connect('customer_management.db')
c = conn.cursor()
repair_cost = 0

def insert_service() -> None:⋯

insert_service()
```

So, we have no way to display the total cost for the customer?  Or do we only see the total cost for the customer in the bottom text box when we click on a customer name in the View Customer Information tab?   -Tracie

We only can see the total currently we aren't seeing the individual prices of the services

7/22/2023: It also appears that the services list is still creating duplicates when we refresh the GUI.  And the duplicates are getting created in the database. We should only have 40 services in the list and as of 7:40 pm today, I have 600 in my local version of the .db.  This will cause performance issues in the long run.  -Tracie

Also, the text display at the bottom of the form is cumulative as well; to work properly it should only show the information for the selected vehicle because it's getting challenging to tell whether services are successfully getting totalled for customer/vehicle combinations.

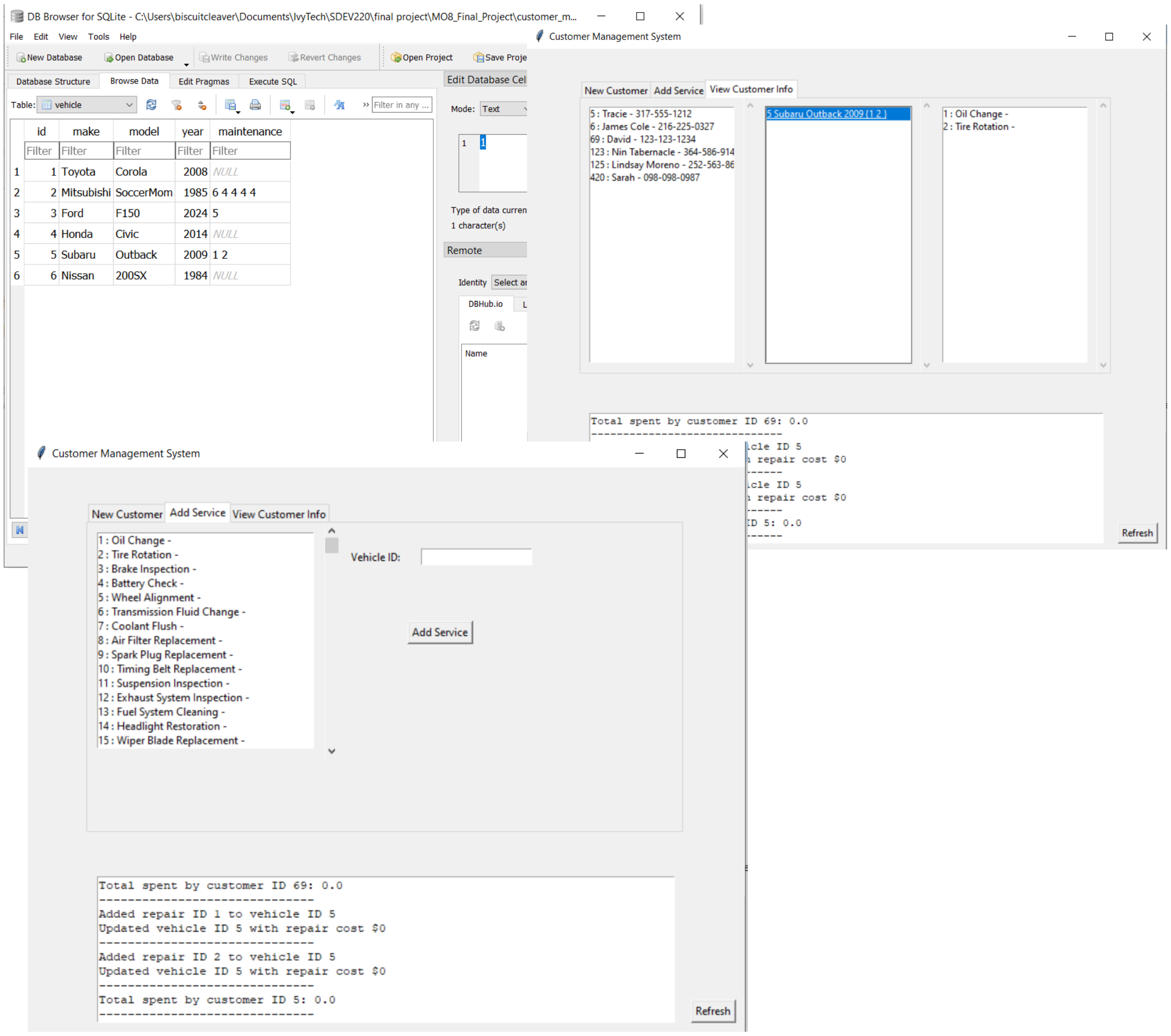


Table: vehicle

| | id | make | model | year | maintenance |
|---|---|---|---|---|---|
| 1 | 1 | Toyota | Corola | 2008 | NULL |
| 2 | 2 | Mitsubishi | SoccerMom | 1985 | 6 4 4 4 4 |
| 3 | 3 | Ford | F150 | 2024 | 5 |
| 4 | 4 | Honda | Civic | 2014 | NULL |
| 5 | 5 | Subaru | Outback | 2009 | 1 2 |
| 6 | 6 | Nissan | 200SX | 1984 | NULL |

Type of data curren
1 character(s)

```
5 : Tracie - 317-555-1212
6 : James Cole - 216-225-0327
69 : David - 123-123-1234
123 : Nin Tabernacle - 364-586-914
125 : Lindsay Moreno - 252-563-86
420 : Sarah - 098-098-0987
```

```
5 Subaru Outback 2009 (1 2 )
```

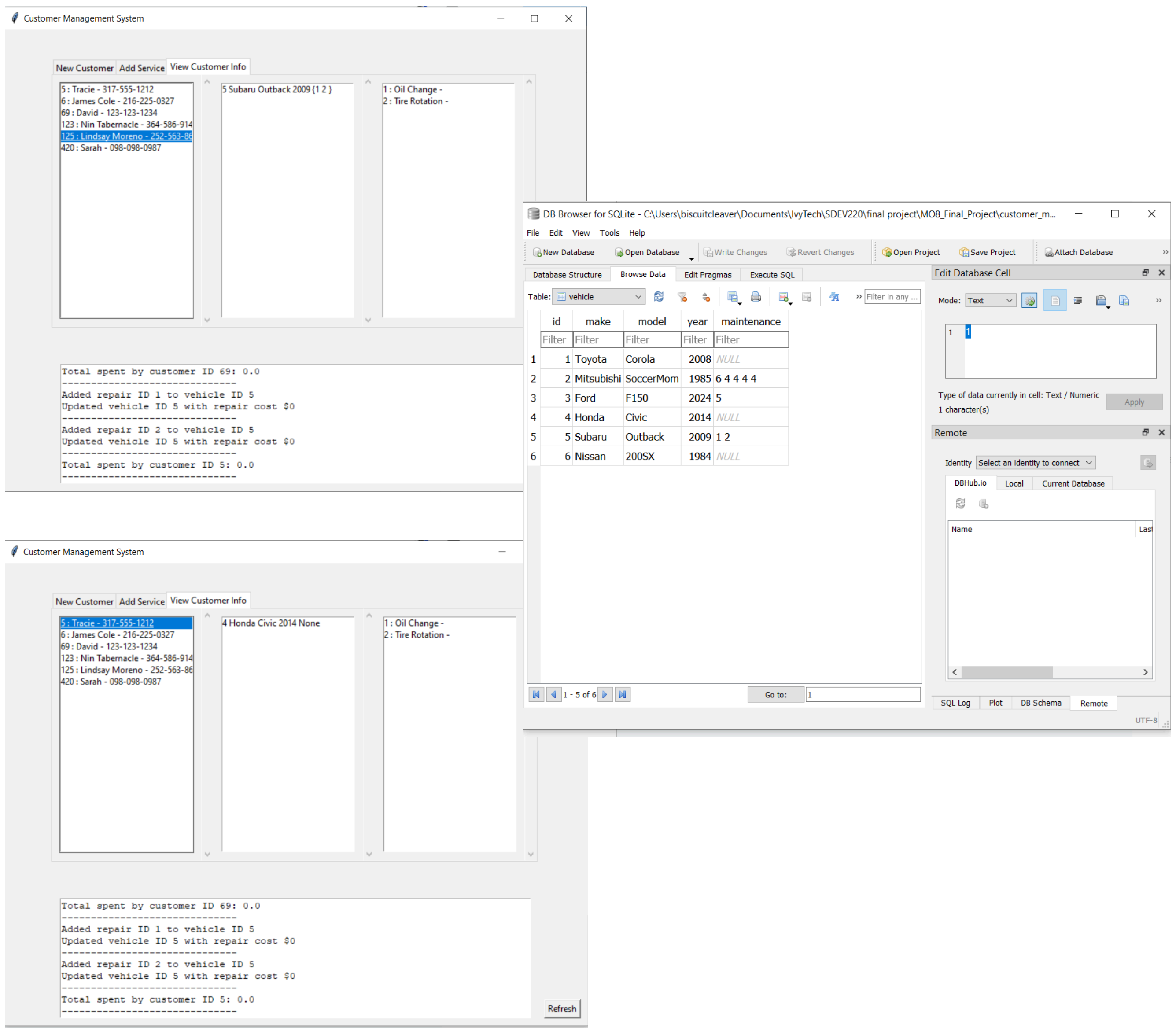
```
1 : Oil Change -
2 : Tire Rotation -
```

```
Total spent by customer ID 69: 0.0
------------------------------
```

```
1 : Oil Change -
2 : Tire Rotation -
3 : Brake Inspection -
4 : Battery Check -
5 : Wheel Alignment -
6 : Transmission Fluid Change -
7 : Coolant Flush -
8 : Air Filter Replacement -
9 : Spark Plug Replacement -
10 : Timing Belt Replacement -
11 : Suspension Inspection -
12 : Exhaust System Inspection -
13 : Fuel System Cleaning -
14 : Headlight Restoration -
15 : Wiper Blade Replacement -
```

```
Total spent by customer ID 69: 0.0
------------------------------
Added repair ID 1 to vehicle ID 5
Updated vehicle ID 5 with repair cost $0
------------------------------
Added repair ID 2 to vehicle ID 5
Updated vehicle ID 5 with repair cost $0
------------------------------
Total spent by customer ID 5: 0.0
------------------------------
```

Service information isn't clearing out of the third display box in the View Customer Information tab when a new customer is selected in the first column. The service details don't disappear until you click on the vehicle associated with the new customer. They should clear as soon as a new customer name is selected in the first box.



```
5 : Tracie - 317-555-1212
6 : James Cole - 216-225-0327
69 : David - 123-123-1234
123 : Nin Tabernacle - 364-586-914
125 : Lindsay Moreno - 252-563-86
420 : Sarah - 098-098-0987
```

```
5 Subaru Outback 2009 {1 2 }
```

```
1 : Oil Change -
2 : Tire Rotation -
```

```
Total spent by customer ID 69: 0.0
------------------------------
Added repair ID 1 to vehicle ID 5
Updated vehicle ID 5 with repair cost $0
------------------------------
Added repair ID 2 to vehicle ID 5
Updated vehicle ID 5 with repair cost $0
------------------------------
Total spent by customer ID 5: 0.0
------------------------------
```

DB Browser for SQLite - C:\Users\biscuitcleaver\Documents\IvyTech\SDEV220\final project\MO8_Final_Project\customer_m...

| | id | make | model | year | maintenance |
|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | Toyota | Corola | 2008 | NULL |
| 2 | 2 | Mitsubishi | SoccerMom | 1985 | 6 4 4 4 4 |
| 3 | 3 | Ford | F150 | 2024 | 5 |
| 4 | 4 | Honda | Civic | 2014 | NULL |
| 5 | 5 | Subaru | Outback | 2009 | 1 2 |
| 6 | 6 | Nissan | 200SX | 1984 | NULL |

1 - 5 of 6

Go to: 1

Type of data currently in cell: Text / Numeric
1 character(s)

```
5 : Tracie - 317-555-1212
6 : James Cole - 216-225-0327
69 : David - 123-123-1234
123 : Nin Tabernacle - 364-586-914
125 : Lindsay Moreno - 252-563-86
420 : Sarah - 098-098-0987
```

```
4 Honda Civic 2014 None
```

```
1 : Oil Change -
2 : Tire Rotation -
```

```
Total spent by customer ID 69: 0.0
------------------------------
Added repair ID 1 to vehicle ID 5
Updated vehicle ID 5 with repair cost $0
------------------------------
Added repair ID 2 to vehicle ID 5
Updated vehicle ID 5 with repair cost $0
------------------------------
Total spent by customer ID 5: 0.0
------------------------------
```

Updating repair_cost in the database is needed to allow for the updated cost to be presented when selecting the vehicle in question. "spent" is used to display total spent, so maybe that is what needs to change.

The on customer_select function should also clear the text field associated with the populate_repairs_for_vehicle function.

Achieving these two things will fix the last two issues we are having with our project.

Tracie noticed that data is being duplicated in the database. Let's look into this and fix it if we have time.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**CURRENT PROBLEMS (7/24/2024):**
NONE
**FIXED:**
Refreshing / service list doesn't reset / adds to bottom
Need to include price with service repair list on customer view
Include price on Add Service Tab
Fix adding services to database every time program opens
When new customer selected in view tab, clear third service text box
Fixed repair cost 0 in text box

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**Module 8 Conclusion:**
Yay we are done! Good job team!