AutoWand

Stephen Albrecht
Sharad Vellore

Individual Contribution Breakdown

Stephen Albrecht and Sharad Vellore proposed the idea of the application. Stephen Albrecht and Sharad Vellore participated equally in the application's implementation, and have both tested the application for errors and bugs. Stephen Albrecht and Sharad Vellore equally participated in designing and performing the usability study. Stephen Albrecht and Sharad Vellore summarized the information collected from the users and integrated it into the application (Stephen Albrecht [20%] and Sharad Vellore [80%]). Stephen Albrecht wrote the final report.

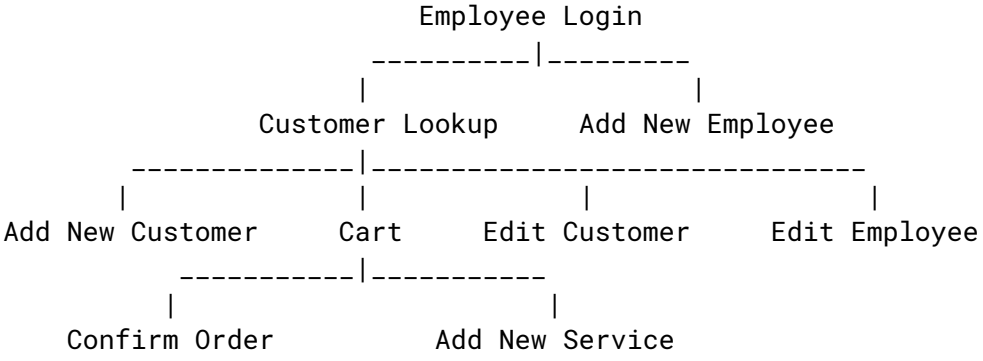
Hardware and Software Requirement

- Microsoft Visual Studio with Visual C# and WPF packages installed

Application Objective

AutoWand is a desktop application designed to facilitate transactions at auto repair shops. Businesses can create accounts for employees, keep a database of customers and offered services, and create invoices for customers.

Application's Flow



Description of Data

The application makes use of three data filles:

1. **Employees**. Contains information on all users of the application.

ID	FirstName	LastName	Username	Password	Permission

2. **Customers**. Contains information on all added customers

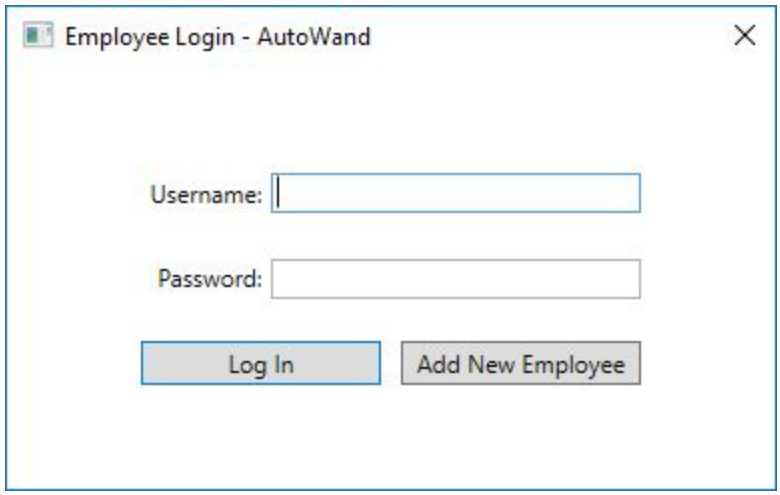
FirstName	LastName	Address	PhoneNumber	EmailAddress

3. **Services.** Contains information on all added services

Name	PartsCost	LaborCost	TotalCost

Main Functionalities/Windows

Employee Login



Employee Login window

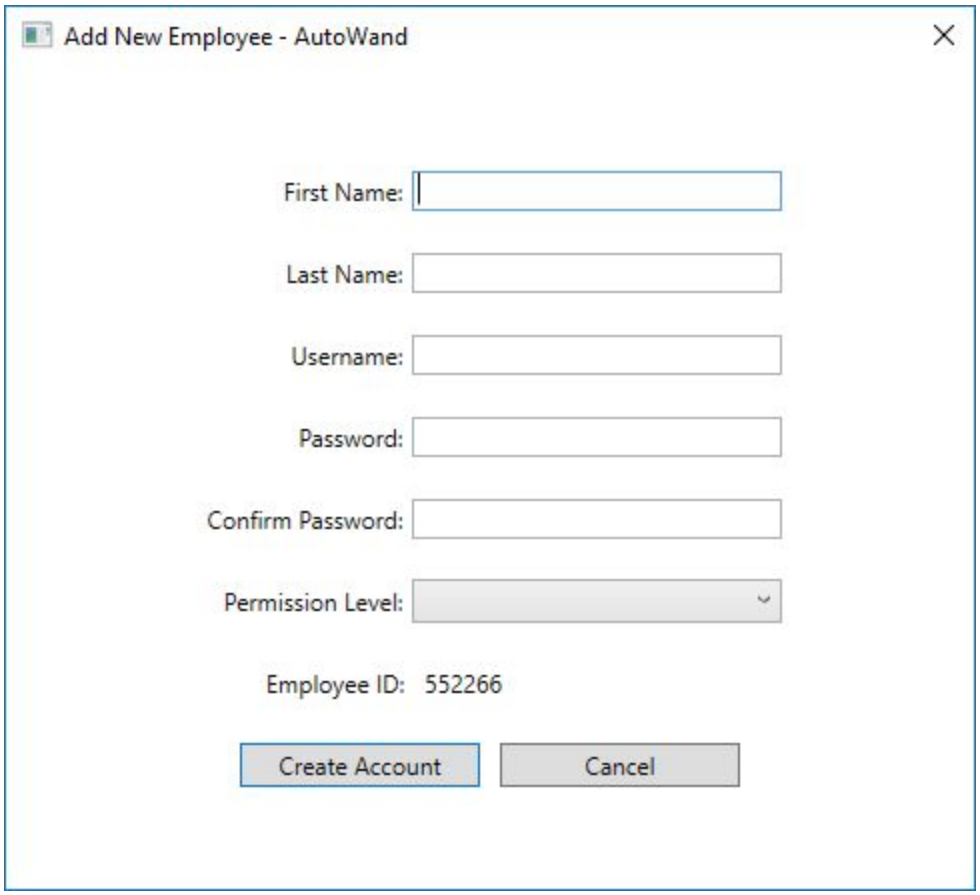
Employees can log in to the application or add a new employee account. Adding a new employee requires entering credentials of a account with Administrator or Manager permissions.

Use the username **admin** with password **1** to create initial accounts.

Add New Employee

Adding a new employee is straightforward. Users must enter the new employee's first and last name, a custom username and password, and choose an appropriate permission level (Administrator, Manager, or Associate). Users with Associate permissions cannot add new accounts, and users with Manager permissions can only create users with

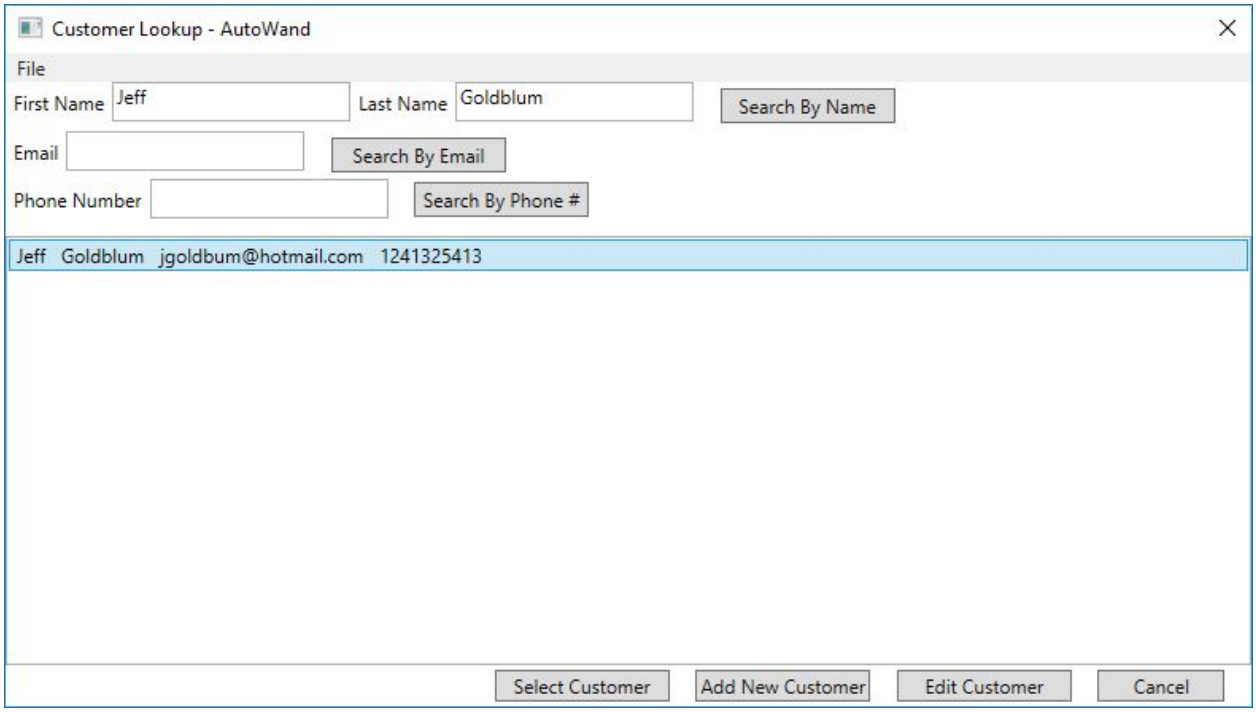
A randomly generated Employee ID is also automatically created for each account.



Add New Employee window

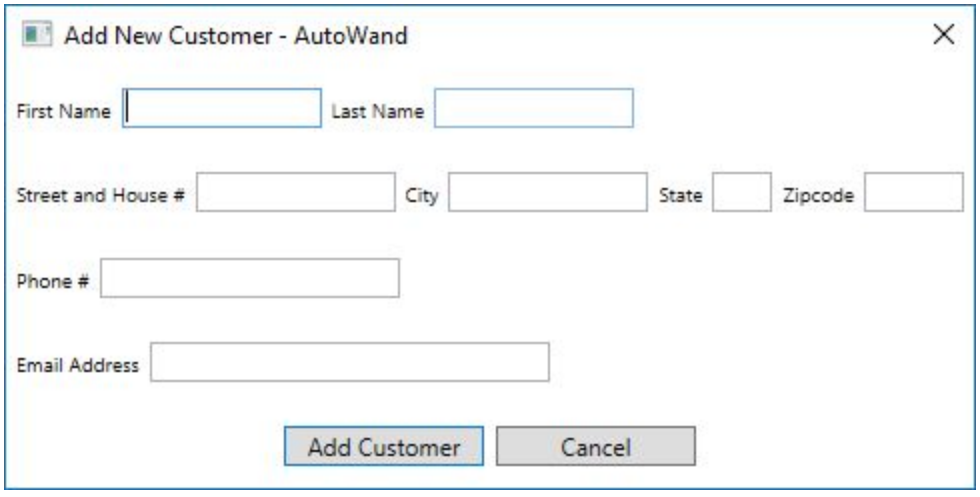
Customer Lookup

After logging in, Users can search for any customer that has an account, or create an account for new customers. Customers have the option of providing either their first and last name, their email, or phone number to help locate them in the system. Once a customer is found, the User should select the correct entry in the list and click the *Select Customer* button to proceed. From this window, users can also edit a selected customer's database entry, or edit their own account information by navigating to File > Edit Account Information.



Customer Lookup window

Add New Customer

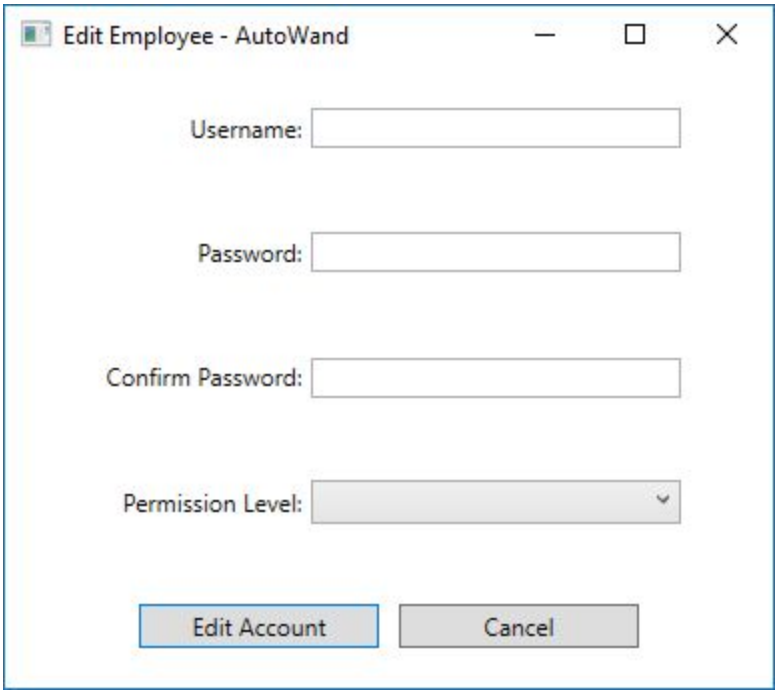


Add New Customer window

Adding a new customer is simple. They must provide their first and last name, home address, phone number, and an email address. Every field is required to create an account.

Edit Account Information

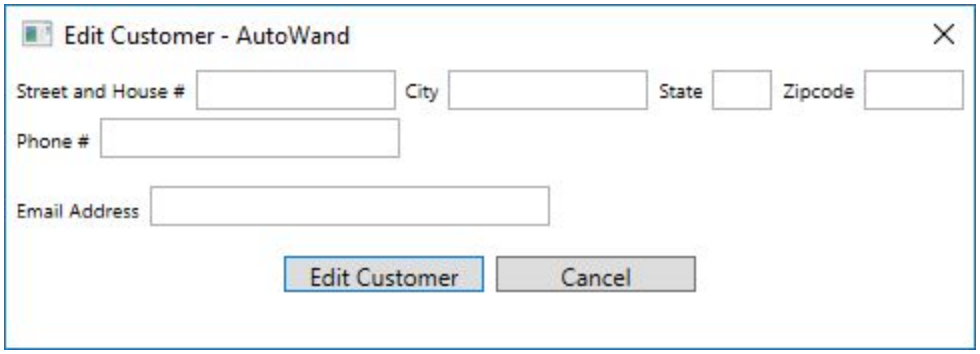
To edit their account information, users must re-enter their account information as if they were creating a new account.



Edit Account Information window

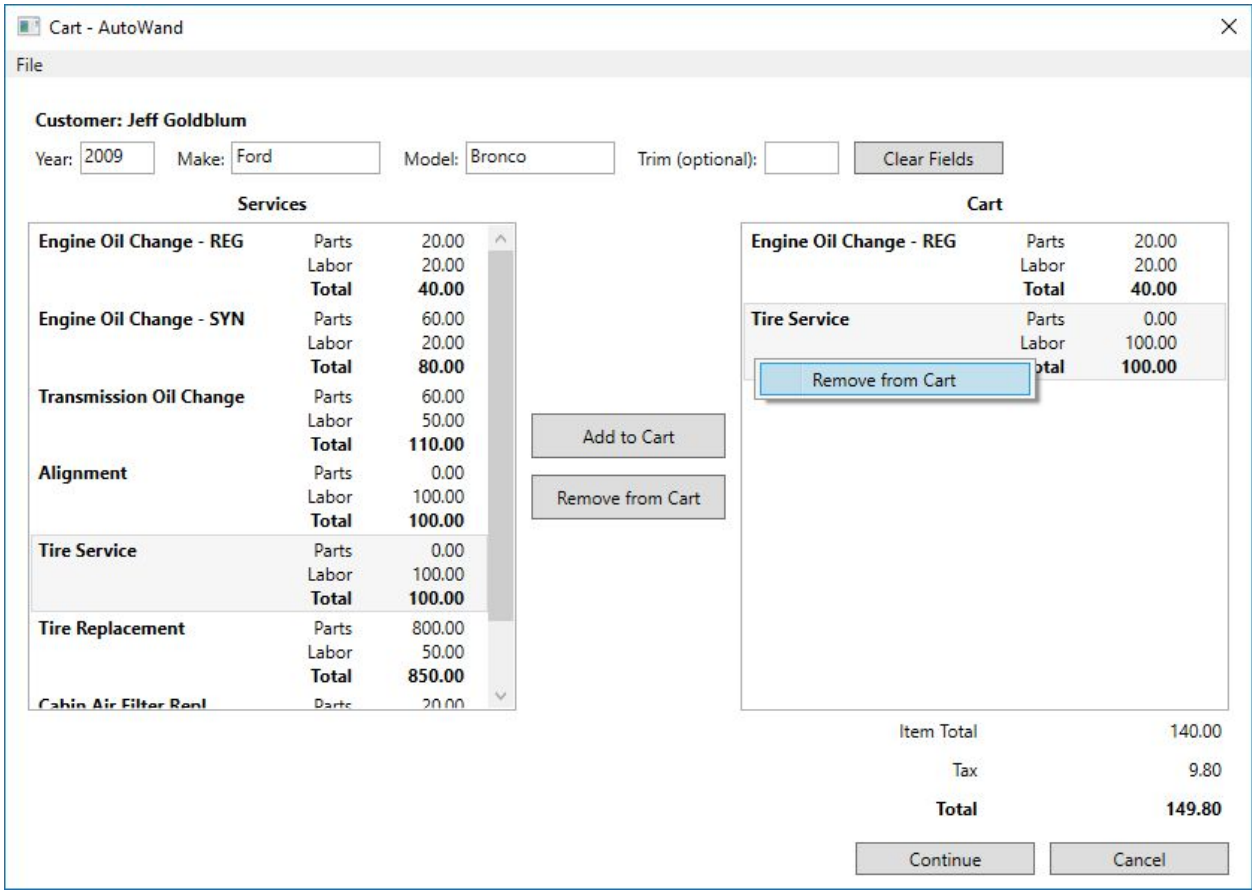
Edit Customer Information

When a customer has a change of address or phone number, their database entry will need to be updated. This can be done by re-entering all of the customer's information.



Edit Customer Information window

Cart



Cart window

In the Cart window, Users must enter the customer's vehicle information to ensure compatible parts are selected.

Offered services are listed in the Services pane on the left of the window. Users can add a service to the customers car either by selecting the service from the list and clicking the *Add to Cart* button in the center of the screen, or by right-clicking a service and clicking the *Add to Cart* button in the popup menu.

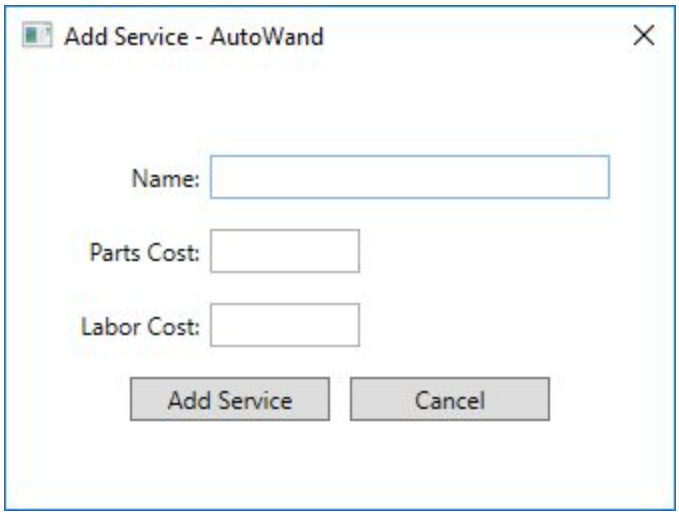
Users can similarly remove services from the customer's cart in the Cart pane by selecting the service and clicking the *Remove from Cart* button in the center of the screen, or by right-clicking the service and clicking the *Remove from Cart* button in the popup menu.

Once the appropriate services are added to the customer's cart and their vehicle information entered, the User can proceed to confirm the order.

Users can also add a new service from the Cart window by navigating to File > Add Service.

Add Service

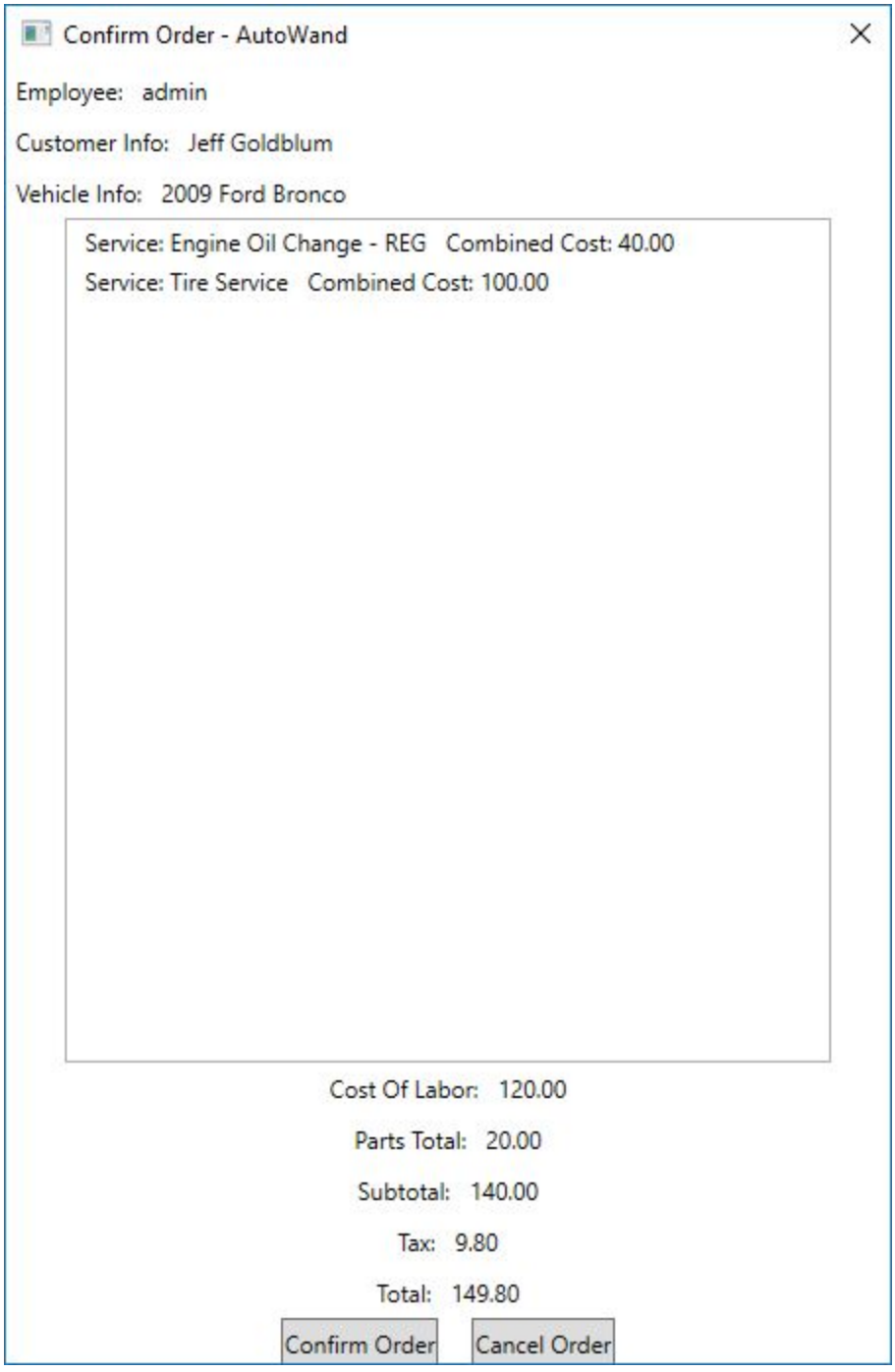
To add a new service, users must enter in a name for the service which is 25 characters or less, as well as dollar amounts for the parts and labor costs. Upon clicking the “Add Service” button, the service will be written to the database and the Cart window will automatically update to reflect the new addition.



Add Service window

Confirm Order

In the Confirm Order window, the order is displayed similarly to an invoice with a cost breakdown. Users can confirm the details of the order with the customer before continuing. In deployment, confirming the order will print an invoice for the customer to sign. In the demo, it terminates the program.



Confirm Order window

Comparison of First and Second Usability Studies

Overview of participants

Group 1 - Average users

Participants	Age	Gender	Occupation
1	56	Female	Mom
2	23	Female	Legal assistant
3	19	Male	AA Student

Group 2 - Computer proficient

Participants	Age	Gender	Occupation
1	31	Male	Engineer
2	30	Male	CSE Student
3	19	Male	ME Student

First Study Results

Group 1 - Average users

Category	Average Score
Novelty	7
Ease of Use	8
Efficiency	7
Attractiveness	5

Group 2 - Computer proficient

Category	Average Score
Novelty	8
Ease of Use	9
Efficiency	8
Attractiveness	7

Second Study Results

Group 1 - Average users

Category	Average Score
Novelty	7
Ease of Use	9
Efficiency	8
Attractiveness	5

Group 2 - Computer proficient

Category	Average Score
Novelty	8
Ease of Use	9
Efficiency	9
Attractiveness	7

Both groups took less time to complete the study than the first one, but this is likely due to the familiarity with our program rather than improvements in the application's flow. However, they praised the implemented changes which made specific parts of the program more intuitive to traverse. Specifically, in our first test most of our users had issues with creating a new employee because it wasn't obvious that they needed to enter credentials before pressing the "Add New Employee" button. In our second test this was not a point of confusion. As a result, our Ease of Use and Efficiency scores went up marginally with group 1, and group 2 increased their average efficiency score by 1. Because the same evaluation method was used between the studies, the added windows expanding the functionality of the application did not factor into the user's remedied scores.

Users still complained about the lack of the application's attractiveness, but we made no efforts to improve this from test 1. Overall, users from both groups agreed that our minor changes were an improvement over the previous iteration of the application.

Future Directions

To be taken seriously, our application should allow the user to modify much of its functionality. For example, shops in different states or locations will have differing taxes. Being able to add or remove various taxes would be a necessary feature. Additionally, different businesses will have different approaches to billing, so being able to modify the breakdown of services (instead of simply a parts cost and a labor cost) is necessary.

Auto repair shops would likely need to know specific part numbers and costs when generating quotes to work on vehicles, so adding a database of parts would be a reasonable goal in the future. These shops would need to be able to order parts in the event that one isn't available, so adding the ability to open links to webpages for ordering a part would also be a good idea.

The appearance of the application could also use improvement. As it stands, all components use their default appearances, giving the application a dull and boring look. Adding more color and experimenting with more visually appealing designs would go far in improving the user experience.

Main Changes Between Midterm and Final Application

New windows

- Added "Edit Account Information" window to let users change their account info
- Added "Edit Customer Information" window to let users change customers' information
- Added "Add Service" window to let users add new services to the database

Implementing User Feedback

- Clicking “Add New Account” button on the Login page without any credentials entered triggers a dialog box letting users know that credentials must be entered to create a new account, rather than just highlighting the required fields.
- The label for the “Trim” textbox on the Cart page was changed to “Trim (optional):” to let users know that the field is not required.

Comparison to Real World Application

There are numerous programs on the market that do in essence service the same requirements that we designed in our application with several key differences. To use as an example for this comparison, a popular automotive search chain, *Jiffy Lube,* utilizes an in house service that they use to manage information about their customers and services, and printing invoices. One of the major differences between our software and *Jiffy Lube’s* is that you cannot use the mouse on their program, meaning that all interaction is done via the keyboard, this can lead to some user frustration as well as be somewhat more time consuming as the user needs to tab through fields to complete forms and so on. Another key difference is that the *Jiffy Lube* software is much more complex and time consuming to navigate, whereas our program flows from login to invoice. Some points of difference where the company’s software is better than ours is in terms of presentation and appearance, there's color and trademarking on their software, making it much more presentable when interacting with a customer. Furthermore, it has much more field time and operation that ours so it is a much more complete product.

Overall Experience

Overall, this was an extensive project which exercised our knowledge in many areas of programming and application design. It forced us to learn advanced topics in WPF and MVVM such as bindings, user controls, view models, etc. However, expanding the functionality of the midterm project felt redundant having completed assignments 4 and 5, which cover several of the same topics. In fact, the assignments had several of the same windows which we have implemented in this project, but some of that could be chalked up to coincidence.

The most challenging part of this project, and the course, was wrapping my head around using VMs and bindings, and why it was useful to use a VM instead of including code in the XAML window’s backend. Trying to do things like validating user input using VMs was much more confusing than when we could easily refer to a component by its name.

I liked how the projects we were assigned seemed more closely related to real-life applications than other classes. I liked that assigning more long-term projects allowed us to expand on applications’ functionality rather than having many small, insignificant projects. I’m much more inclined to show off these larger projects to potential employers.

I learned an entirely new language and several frameworks this semester. I gained a solid appreciation for the power of C# and the ease with which GUI applications can be developed

using the right methods. I feel that the knowledge I gained from this course made me more employable and versatile in my knowledge of CS concepts. As the course went on, nearly every time we moved to a new topic I was impressed with how useful it seemed. In general, I’m very satisfied with my experience.