Household Equipment Repair Program

SENG 3110 - Algorithms & Data Structure

By: Chris Coulthard, Luka Aitken, Toma Aitken

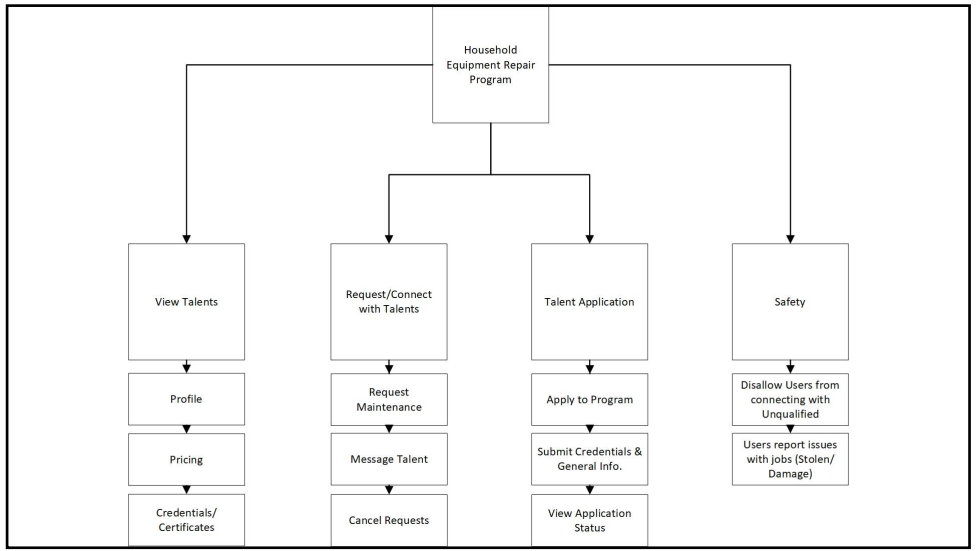
Index

Problem Definition

- There is a large selection of home repair solutions that homeowners can choose from, but nearly infinite selection isn’t always a good thing.
- Trying to find the perfect talent for repairs, many options do not provide the right qualities for the problem.
- It takes time to search through a list of talent repairs to find the right talent.
    - That doesn’t overcharge.
    - Provides quality work.
    - Can be trusted.
- With a Household Equipment Repair Program, all the time previously being used searching is now used for repairs, allowing homeowners to experience a house in top quality again.

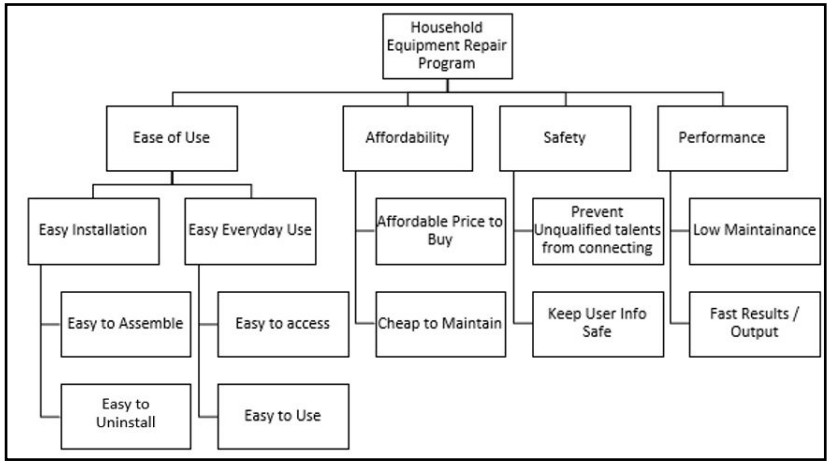
Functions

- The system should let the user add their request and connect with the right talent.
- The system should display the price for the talent, how much it would cost to hire talent, and talent rates.
- The system should display important information about the talent and skills the talent can perform by displaying their credentials and experiences.
- The system should stop users from connecting with unqualified talent to prevent unfair prices/rates and low-quality repairs.



Objectives

- Cost-effective System.
- Easy to Use.
- Provides an easy way to communicate with qualified talents.
- Prevent Homeowners from requesting help from unqualified talent.



Constraints

- The Household Equipment Repair program should:
  - Use a certain amount of proper algorithms and data structures in C/C++.
  - Have GUI for better user interactions and ease of use.
  - Be reliable and give correct information about the talent’s details and pay rate.
  - Be aesthetically pleasing and well organized for the users so it is easy to understand.
  - Be cheap to maintain while providing the best services for the user.
  - Be sustainable enough for all users using the program.
  - Improve communication between users and talents.
  - Keep users' information safe by regulating unqualified talents.

Requirements

Functional:

- User Opens the program.
- User chooses household repair assist type.
- User chooses talent to help assist in repairs based on location, cost, and qualification.

Non-Functional:

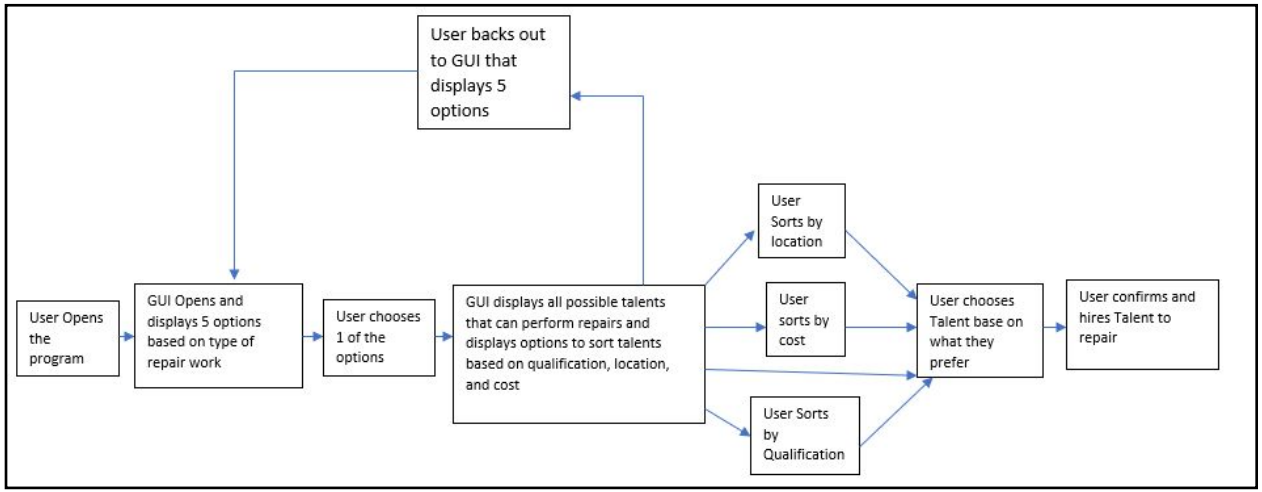
- Should keep user information safe when using the program.
- Give results of qualified talent efficiently and fast.
- Should list the most unqualified at the bottom.

Scope & Complexity

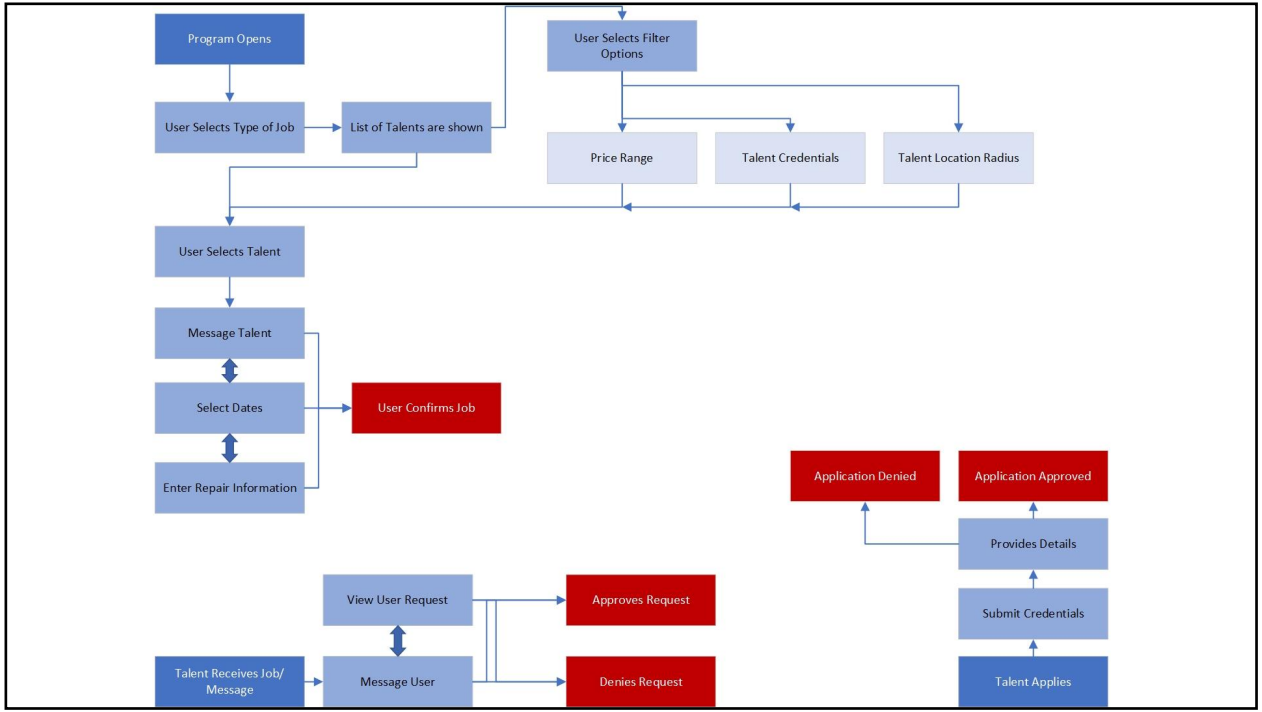
- The program will be on Windows desktop using C++.
- The program will only be accessible in Kamloops, B.C.
- The program will have different job options to fix repairs.
- The qualifications, location and cost will be assumed by us.

Windows®

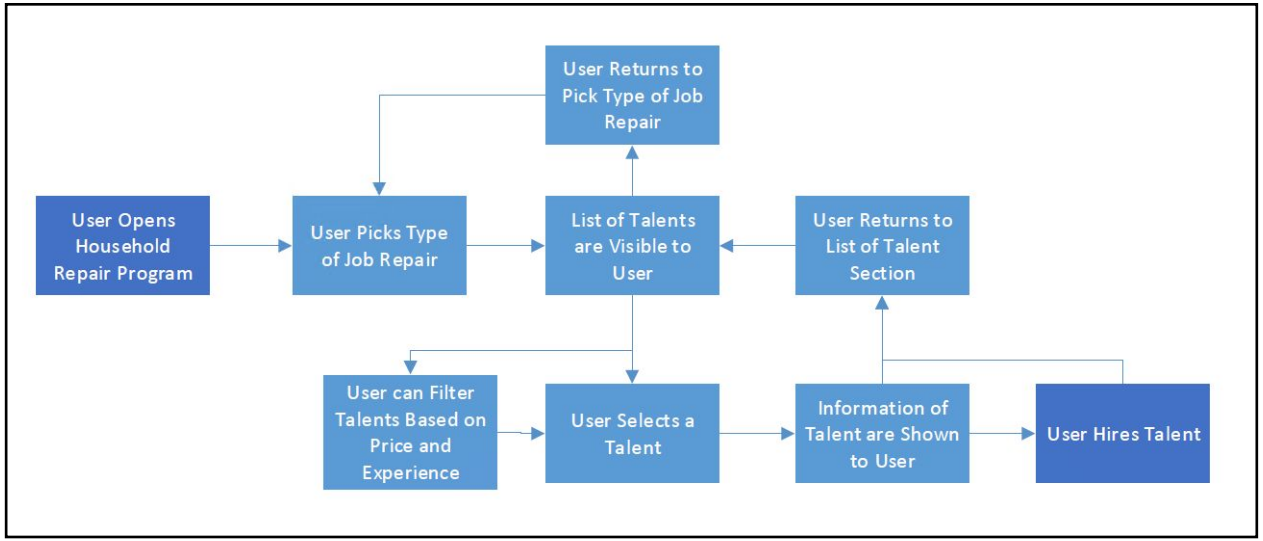
Design Idea #1



Design Idea #2

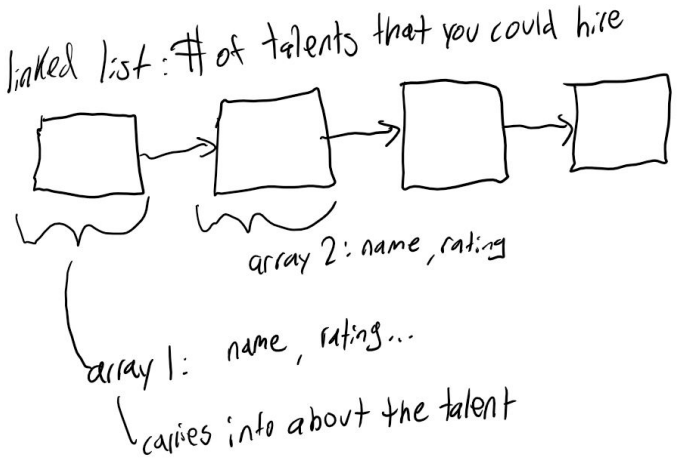


Design Idea #3



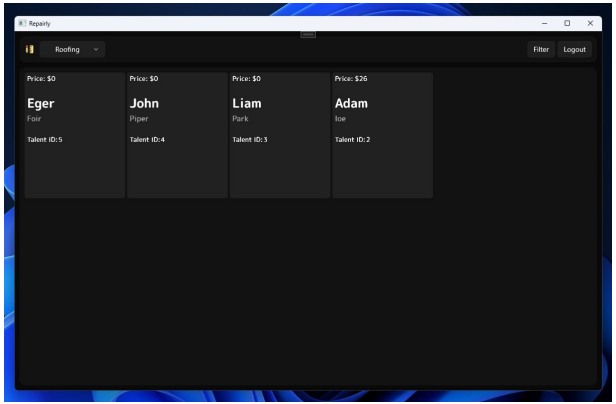
Alternative Solution

- Used Linked Lists, Arrays and Selection Sort.
  - In each part of the Linked List had an array representing the talent with their information
- Reason why we didn't use this solution was that:
  - when trying to sort, the program would need to convert strings into numeric values in order to sort.
  - Selection sort was consider more unstable and higher complexity.



Final Solution

- Implementation of aspects from all 3 design ideas.
- Used Linked lists, arrays, classes, maps and Insertion sort to do the functions of the program.
  - The map was used to define the talent's available jobs and what each job cost..
  - One class to create the talents with their information and another class to store the talents into the linked list.
  - Insertion sort to sort each talent by an attribute.
- Used Win32, C++/WinRT and WinUI 3 to create the GUI of our Household Equipment Repair Program which included:
  - Sign up and Login screen.
  - View to see all talents.
  - Buttons for choosing Job types, Filtering by attributes and Logging out.



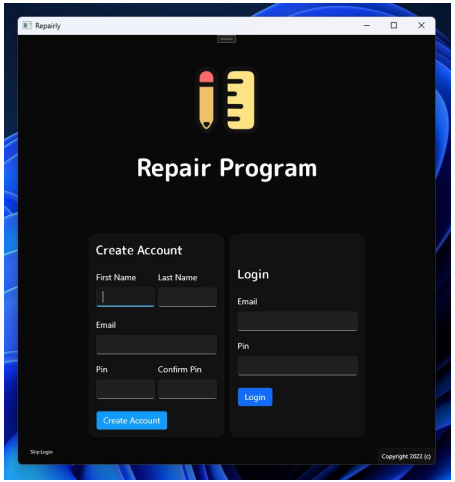

```
Program Started
Enter a job type: (First Letter Upper-Case, rest lower-cased)
Plumber, Roofer, Carpenter, Painter, Flooring, Electrician: Flooring
Job: Flooring Name: Eger Foir Price: 27
Job: Flooring Name: John Piper Price: 87


Job: Flooring Name: Gus Johnson Price: 12
After Sort --------------------
Job: Flooring Name: Gus Johnson Price: 12
Job: Flooring Name: Eger Foir Price: 27
Job: Flooring Name: John Piper Price: 87
```

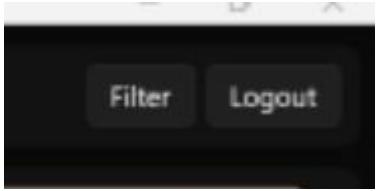
Decision Matrix

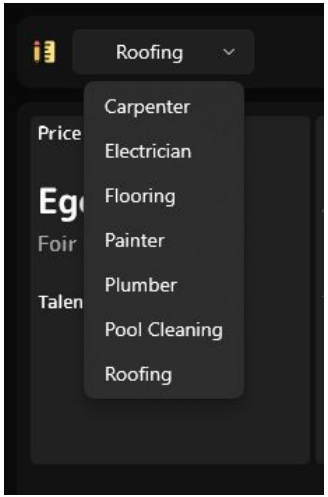
		Solutions			
		Original		Main	
Criteria	Weight	Score	Partial Score	Score	Partial Score
Aesthetics	0.25	6/10	0.150	7/10	0.175
Complexity	0.30	7/10	0.210	6/10	0.180
Simplicity	0.20	5/10	0.100	8/10	0.160
Performance	0.25	5/10	0.125	9/10	0.225
Sum	1.00		0.585		0.740

Features

- Allows users to either sign up or login into the program to find talents.
  - Once the user signs up, they can login in.
  - After logging in the user can
    - View the talents by clicking on the job type button and choosing a job type.
    - Sort the talents by clicking on the Filter Button.
    - Logout of the program







Considerations

Environmental:

- Used proper data types to make the program much faster.
- Used a stable sorting algorithm that can sort without failing.

Societal:

- Made our GUI:
  - Easy to understand.
  - Easy to use.
  - Aesthetically pleasing.

Considerations Cont.

Safety:

- Implemented a Sign-in screen to only allow users who have an account with the program.
- Can organize talents by different attributes to allow users to better view the difference between each talent.

Economic:

- Cost-Effective Program.
- Not behind any paywall or subscription.

Limitations

- Weren’t able to get the GUI and C++ Program to work properly.
  - Not a lot of information on the GUI maker and used a different language.
- Wanted to implement:
  - A search bar to be able to search talents rather than sorting or choosing Job type.
  - A text file that holds all the information of the talents, which could be fetch into the program.

Lifelong Learning

We learned many valuable skills when building the Household Equipment Repair Program

- Learn how to develop and code a GUI.
- How to use the different algorithms and data structures together.
- How to make a very user-friendly and aesthetic GUI.
- Learned how to use Win32, C++/WinRT, and WinUI 3 in Visual Studios.
    - Win32 is used as the main set of Windows APIs for developing 32-bit applications.
    - C++/WinRT was used to project C++17 language for Windows runtime APIs.
    - WinUI 3 was used as the native UI platform components that are bundled with the Windows App SDK.

Conclusion

- Went through many considerable design ideas for our program.
  - How to program will function.
  - Where we will develop our GUI.
  - What kind of algorithms and data structure will be used.
- Biggest challenges for homes are managing the home itself and the things that could go wrong when looking for talents such as:
  - Overpriced Talents.
  - Talents that are unqualified to do the repair.
- With the use of the program, it allows users to find an accessible and easy way in finding the right talent for their household repairs.

Future Work

- Finish and have a fully functional GUI.
- Be able to have a database to store all talents.
- Allow talents to sign up and enter their details.
- Have a search bar to allow users to search talent names.

Thank You For Your Attention!!

Any Questions?

References

- https://us.123rf.com/450wm/lightfieldstudios/lightfieldstudios1810/lightfieldstudios181026883/lightfieldstudios181026883.jpg?ver=6
- https://bs-uploads.toptal.io/blackfish-uploads/components/seo/content/og_image_file/og_image/987138/0823-DashboardDesign-Dan-Social-e319a5a8a7ceb75b9e5010740700d409.png
- https://www.windsor.edu/wp-content/uploads/2017/06/requirements.jpg
- https://pentagram-production.imgix.net/ea053844-c063-4130-9425-4a193f82e1e3/ps_windows_01.jpg?crop=edges&fit=crop&h=630&rect=67%2C364%2C1665%2C1040&w=1200
- https://www.fujifilm.com/fbhk/-/media/fbhk/7,-d-,-insights/1027683138_1920x1080.jpg
- https://static.igem.org/mediawiki/2018/8/89/T--Waterloo--HP-SC.png
- https://mobile-cuisine.com/wp-content/uploads/2013/10/savings-ahead-e1453560429196.jpg
- https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTcY3zPEz9sNd3DqcD8iGk8JKsEFoiwDnGazw&usqp=CAU
- https://static.vecteezy.com/system/resources/thumbnails/002/272/250/small/browser-search-bar-template-simple-minimal-design-with-magnifying-glass-search-icon-free-free-vector.jpg
- https://envirocleanglobal.com/wp-content/uploads/2017/02/healingworld.jpg
- https://static.thenounproject.com/png/56875-200.png
- https://upload.wikimedia.org/wikipedia/commons/thumb/5/59/Visual_Studio_Icon_2019.svg/1200px-Visual_Studio_Icon_2019.svg.png
- https://imageio.forbes.com/specials-images/imageserve/601972f44da43f2fe82d7958/WHAT-S-NEXT-/960x0.jpg?format=jpg&width=960
- https://www.springboard.com/blog/wp-content/uploads/2020/07/what-are-data-structures-and-algorithms.png
- https://www.icegif.com/wp-content/uploads/clapping-icegif-10.gif