

Cardio Fitness Center app for managers;

Author: Student#1608

Group: 224-2;

Submission date: 10.06.2023

1. Problem statement:

Making an app which keeps track of purchases of all treadmills of the company. It helps fitness club administration to manage and analyze the sale of products using certain patterns.

Implementation details:

I employed a range of tools including QPainter, Dialog, QMessageBox, QFileDialog, QFile, and other similar built-in extensions and data types.

The project implementation involves various tasks, such as connecting signals and slots, handling user input, reading and writing data from/to a file, displaying data in a table view, editing and deleting data, saving data, and presenting statistics based on the data. I have also included optional features, such as replacing the pie chart with tables for displaying statistics. Here are some details about the approaches used:

File Handling: The application supports loading data from a CSV file and saving data to a CSV file. For this purpose, I utilized the QFile and QTextStream classes to read and write data in the CSV format.

User Interface: The user interface was designed using Qt Designer, which provides a visual way to create and organize widgets.

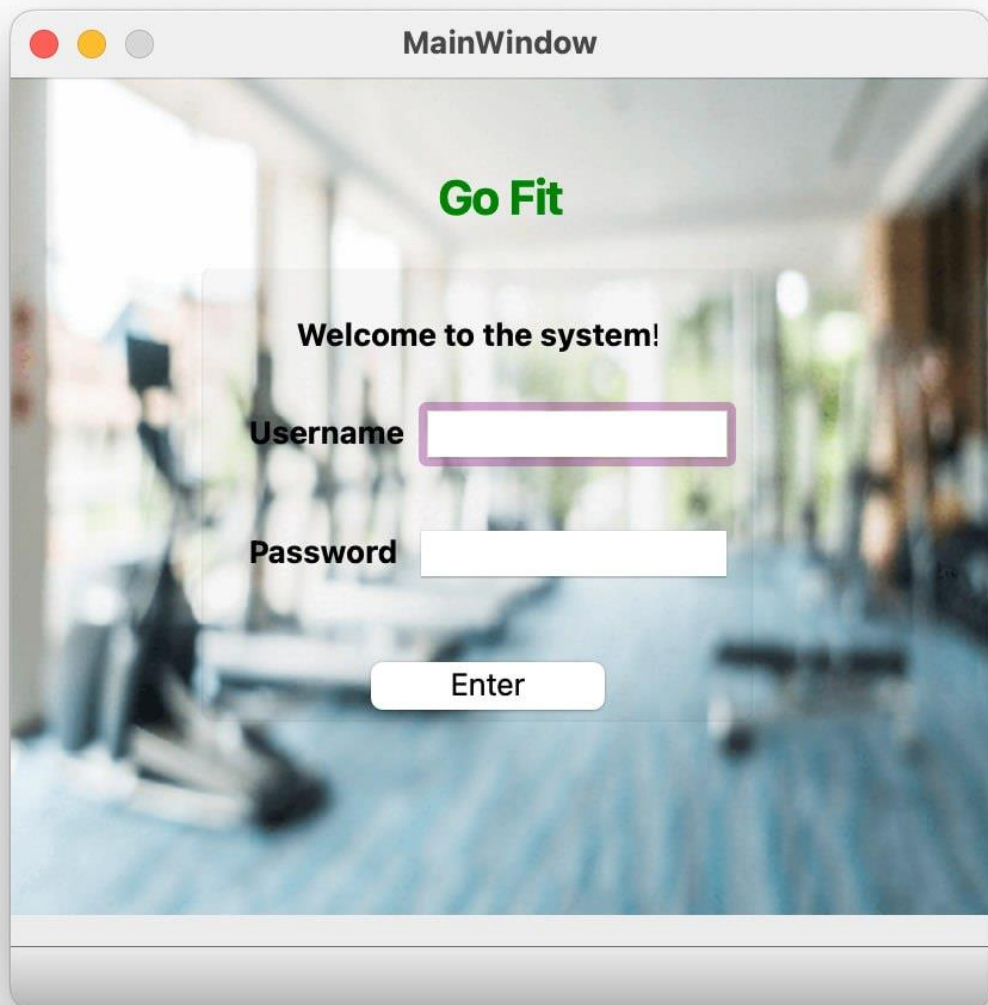
The "Add" class represents a dialog window used for adding new rows of data. It consists of a form layout with labels and line edit fields for each attribute of a treadmill purchase. To create this class, I gathered information [online](#) and adapted it to correctly work with my data.

Sorting and Filtering: To enable sorting and filtering capabilities in the table view, I used the `QSortFilterProxyModel` class. I found relevant [information](#) on how to use it and successfully implemented it in my code.

I also have functions such as editing existing rows, deleting rows, and displaying statistics in a separate window called "Statistics." These features are implemented using various slots and signals in the authorization class. However, I encountered some challenges with this particular part. First time the program used to crash when I tried to add a new purchase. To resolve this issue, I introduced a new function that displays a dialog box using `QInputDialog::getItem()`. This function prompts the user to select the necessary options for adding a new purchase. I'm highly satisfied with this function as it not only improves the overall appearance but also ensures that the format of the added data in the table maintains the same "style" as the original data.

Statistical Analysis: The application includes a separate window called "Statistics," which is an instance of the Statistics class. I created a new class specifically for this purpose. By clicking on the "Statistics" button, the Statistics window displays a table view and provides a method to populate the table with data for statistical analysis. I also spend a lot of time on this part since I needed to create it so it worked correctly even after adding a new data.

Here are some pictures of the app:



| Product | Age | Gender | Relationship | Fitness | Income |
|---------|-----|--------|--------------|---------|--------|
| TM798 | 22 | Male | Single | 3 | 48658 |
| TM798 | 22 | Male | Single | 5 | 54781 |
| TM798 | 22 | Male | Single | 5 | 48556 |
| TM798 | 23 | Male | Single | 5 | 58516 |
| TM798 | 23 | Female | Single | 4 | 53536 |
| TM798 | 23 | Male | Single | 5 | 48556 |
| TM798 | 24 | Male | Single | 5 | 61006 |
| TM798 | 24 | Male | Partnered | 5 | 57271 |
| TM798 | 24 | Female | Single | 5 | 52291 |
| TM798 | 24 | Male | Single | 5 | 49801 |
| TM798 | 25 | Male | Partnered | 5 | 49801 |
| TM798 | 25 | Male | Partnered | 4 | 62251 |
| TM798 | 25 | Female | Partnered | 5 | 61006 |
| TM798 | 25 | Male | Partnered | 3 | 64741 |
| TM798 | 25 | Male | Partnered | 4 | 70966 |
| TM798 | 25 | Male | Partnered | 5 | 75946 |

3. Results and discussion.

I implemented almost all mandatory features and also I added all additional features I was asked to add. Of course, I had some problems with this project and there are some features that can be changed so the final work could look better. For example, I couldn't manage to make a function which is literally rewrites loaded f in resources CSV file with added information by clicking on the button "Save" in menu bar, instead I decided to add a button "Save as" which works perfectly and lets user save updated file on PC. I also really wanted to create pie charts but unfortunately I couldn't simply include necessary libraries to do so. The last thing that I would also try to create code which is drawing the logo, but instead I made my own one and I'm totally satisfied with the way it looks like. The rest of the work is absolutely fine and I'm happy with its final look.

4. Conclusion

I think a big work was done to make this app working correctly. I faced a lot of problems while doing it especially when the program crashed all the time and I needed to go through the whole code to find where I should add something or instead , remove. I think some features can be added such as , for example, adding ID/Name of the customer in csv file so the search bar can be added.