

# Diamond Assisnant App

project id1735

10.06.2023

---

## Problem statement

---

The project solves the problem of storing the information about a diamond assortment, helping its user to view the collection of diamonds with different specs. An app helps user to navigate through the collection in a handy way, performing sorting and filtering diamonds by price. The app is also able to display the information about price and carats of all the diamonds on a scatter plot. There is also an opportunity for user to get the average values of some fields of the table. User can modify every record in the table to fit their needs and save/load different tables from csv files.

## Implementation details

---

The project consists of 9 cpp modules and corresponding headers:

- avgdiamond — shows a window with the information about an average diamond in the table
- budgetchoosedialog — allows user to set the lowest and the highest value for filtering diamonds by their price
- chartview, chartwindow — are used to display the scatter plot of the diamonds' prices and carat values
- diamonddialog — allows user to view and modify the data of one dimond at a time
- diamondproxymodel — performs filtering and sorting of the data

- `diamondstable` — stores the information, loads it from file and saves to file. Also includes the `Diamond` class, which is widely used in the project
- `logowidget` — class inherited from `QWidget`, which is used to display the logo of the project, using `QPainter`
- `mainwindow` — includes all needed signals and slots to perform the actions

The most interesting implementation I would like to share is an approach used to display the scatterplot. To perform this, an additional library was used — **QtCharts, which is mandatory to install before starting the app.**

This helped me to discover the way to attach and use external libraries in my QT projects.

## Results and discussion, conclusion

---

The project is mainly finished, lacking the functionality to add new Diamonds and to remove the existing ones from the table. An additional functionality can be implemented to filter the diamonds not only by their price, but by each of the existing fields. I consider the source code well-organised, readable and structured, however, an inappropriate codestyle was used (I did not figure out how to change the default parameters of code autoformatting in QT Creator).

Implementing the project allowed me to familiarise myself with the development techniques used in industrial programming. It helped me to master the memory management in my C++ projects and made development using QT way easier for me.

All of the remaining functionality was not implemented only because the lack of time and I am able to perform the implementation of every feature during a defence session.