Play Market Digester

Made By Ivan Sinitsin (214-2)

Supervised By Julio Carrasquel

Submitted at 13.06.2022

Student ID: 177

Specification ID: 156

Problem description:

In this assignment I've developed a wrapper for a dataset provided to me by one of the anonymous students. In this project I've been using various programming patterns and techniques previously learned on this course. Examples from the course lectures and seminars have also been utilized during development.

The idea for the project was to process the dataset given, reformatted into a useful table that can be filtered and searched through in various ways. As well as that, the user's able to pick out specific items and store them in a shopping cart of sorts, which items can be appended and removed.

The dataset in question can be obtained here: https://www.kaggle.com/code/kamganganthony/the-android-app-market-on-google-pla/data.

The specification in question can be obtained here: https://github.com/Miv2nir/dsba-itop2022-hw/blob/master/spec_156.pdf.

Implementation details:

The first major challenge in the implementation was the importing stage. The dataset given to me was not consistent throughout: a lot of fields had excess commas; some had missing values. This led to a rewrite of a handler with manual processing of each line in order to accommodate for the dataset.

Once implemented, the data is then processed to be presented in a form of a table, where a user can use various filters and sorts in order to search for various values present in the dataset. The filtering method involves creating a proxy model with subsequent filter implementation in order to accommodate for the search bar, the Paid/Free toggles and the price range. The search has been tweaked in order to be able look up incomplete queries in the real time. All filters are being in the real time as well.

A user can pick an element in the table, and, if they ever so wish they're able to save the element to the cart. The cart is a separate table, located in another dialog window, accessible by a press of a button, which stores all saved entries from the main table. These entries can also be removed with the use of a removal button on the main table as well. (Couldn't make it on a second window, unfortunately).
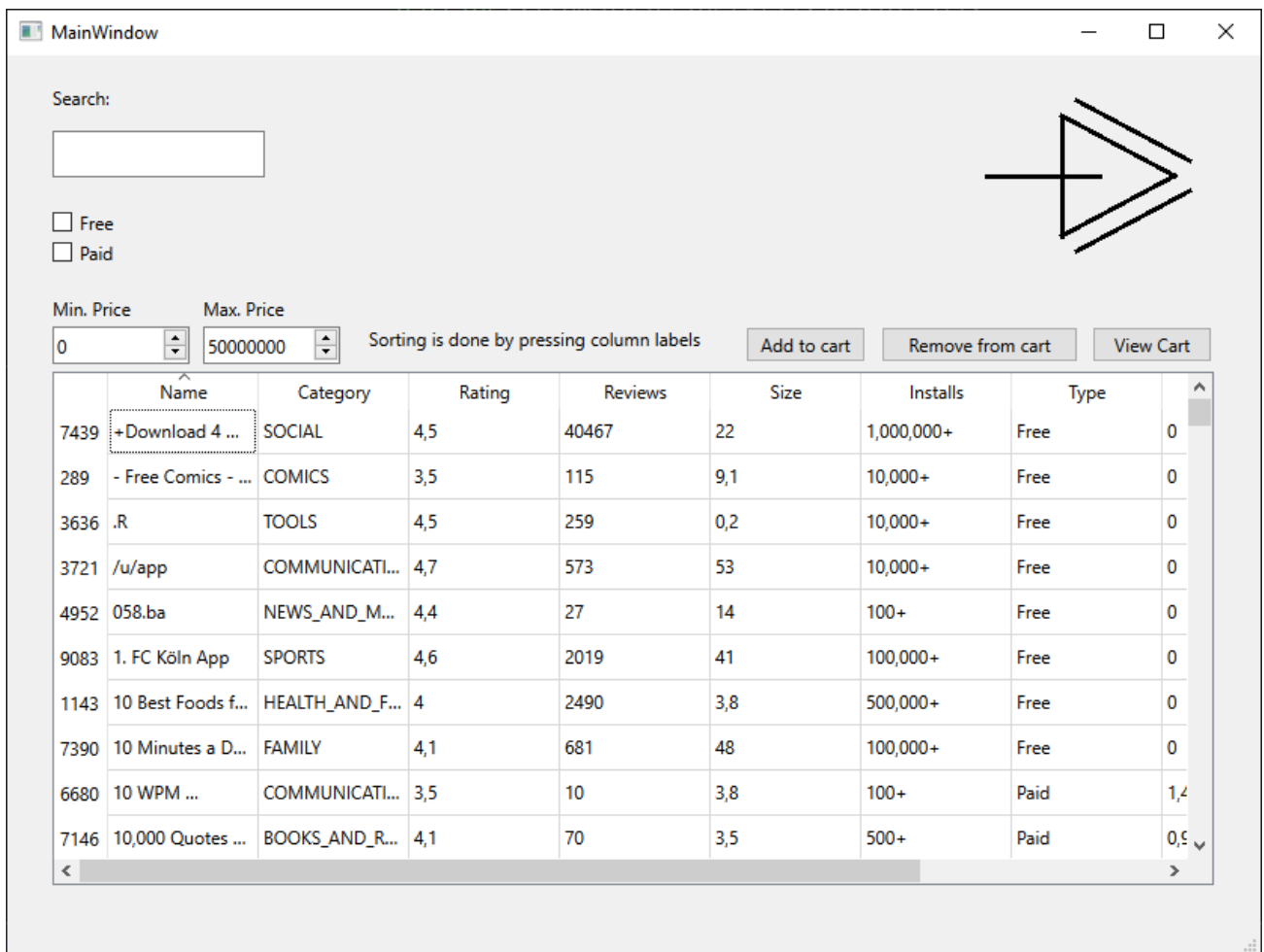
Link to the GitHub repository: https://github.com/Miv2nir/dsba-itop2022-hw.

Results:

1. The project helped to reinforce the knowledge and the reason of studying C++ OOP patterns and using its features.
2. Qt is a really complex library that was not really meant for a small-scale use. There's a lot of unpleasant quarks that it happens to have, which sometimes don't make too much sense from a logical standpoint.
3. When done right, the UI of the app can serve a huge boost in the efficiency and ease of work with a tool. Providing filters as easy and clear ticks or buttons or spin boxes help user comprehend and manipulate provided values in various ways.

Conclusion:

In the end, it's pretty clear that tools such as Qt or C++ are capable of doing a lot of stuff on their own, which can make one life easier for brief moments. The completed project may lack polish or potential features (like a cleaner arrangement of buttons related to the cart, as it would make more sense having

them over at the second window but no way has been figured out to achieve such a thing), full category listings, etc.