

CSP-586 UML FINAL PROJECT TEAM#5

Requirements/Design document

Huang, Silin

A20383068

Xu, Chen

A20377739

Xu, Yifan

A20385903

1. Brief project overview statement.

The main purpose of this project is to build a dashboard to load and analyze the education data of world university ranking. This project will use freeware dataframe.js and load dataset by CSV files. Meanwhile, this project is using some concepts of object-oriented techniques to build a dashboard for data analytics and data visualization. The dashboard will allow users to select from five different datasets:

Dataset:

1. cwurRanking.csv (Center for World University Rankings)
2. sjtuRanking.csv (Shanghai Jiao Tong University Ranking)
3. timesRanking.csv (Times Higher Education)
4. OECDaverage.csv
5. schoolInCountry.csv

All these dataset extract from

<https://www.kaggle.com/mylesoneill/world-university-rankings/data>

<http://cwur.org/>

<http://www.shanghai ranking.com/ARWU2016.html>

<https://www.timeshighereducation.com/world-university-rankings>

Moreover, user can filter dataset to see specific data. Display each dataset into different kind of charts. Also, user can filter the rows and columns of choice while plotting the data in charts.

2. Requirements/Features List

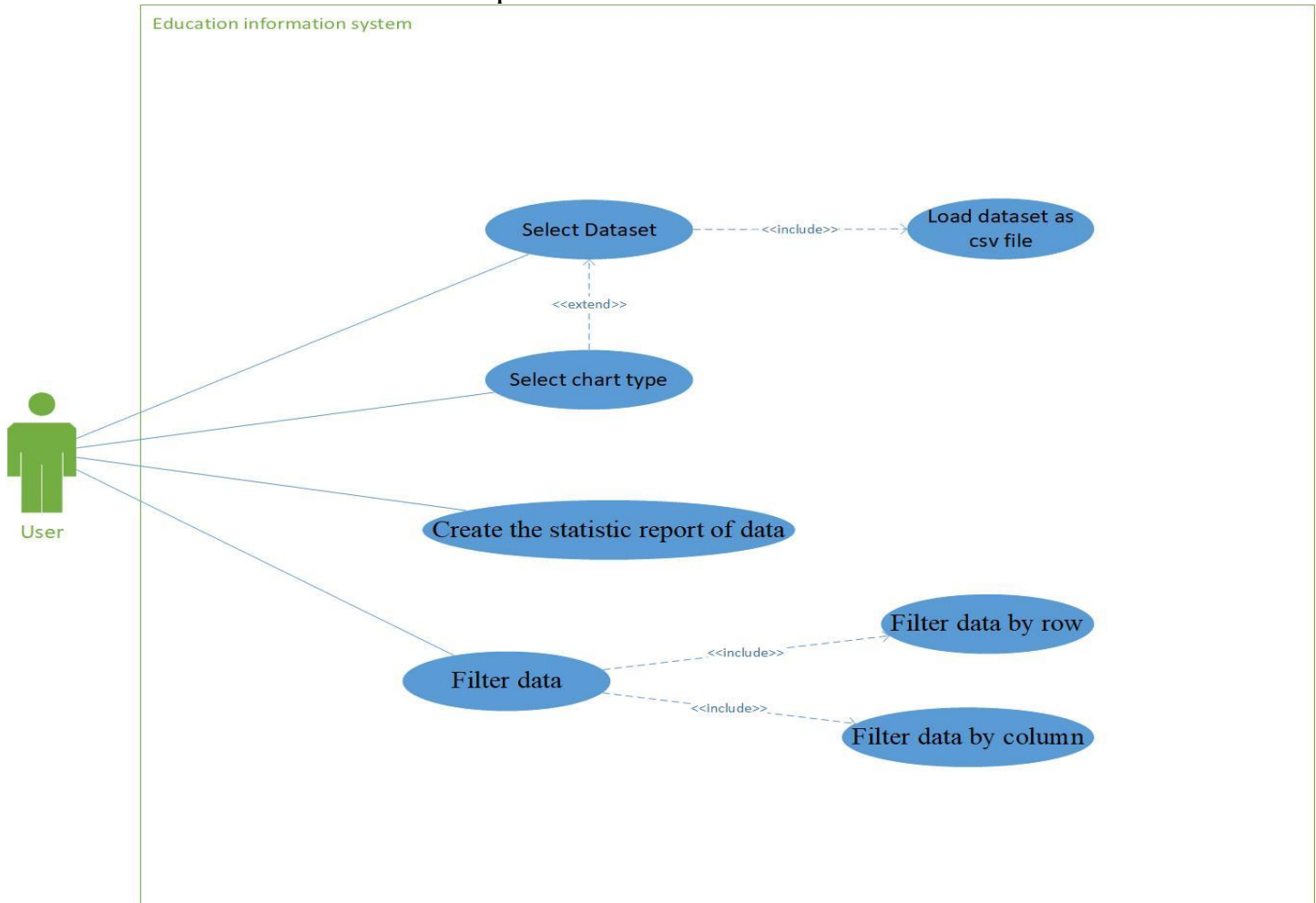
- (1) The user can select education datasets.
- (2) Once the user selects any datasets, the data will be loaded and displayed in the dashboard.
- (3) The user can operate data in min value, max value and average value.
- (4) The user can calculate the count value, the standard value and deviation value.
- (5) The user can sort data in ascending and descending order
- (6) The user can apply data in many chart types: Line, Bar, Pie, Stacked and Pivot.
- (7) The user can modify and filter data by rows and columns

3. Use Cases and Use Case Diagram

- (1) Actors: User
- (2) Use cases:
 - Select dataset
 - Load dataset as csv file

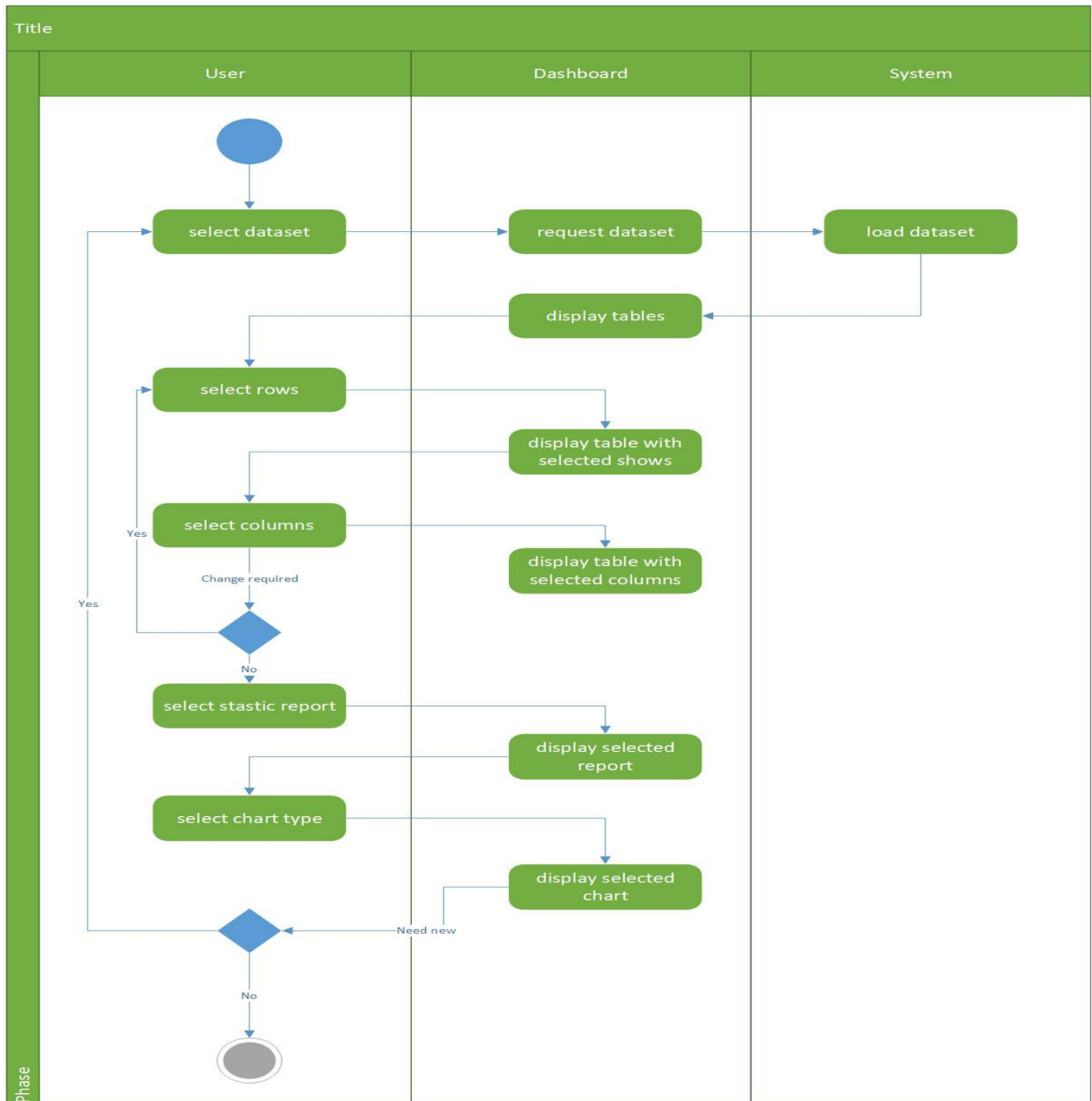
- Select chart type
- Filter data
 - Filter data by row
 - Filter data by column

-- Create the statistic report of data



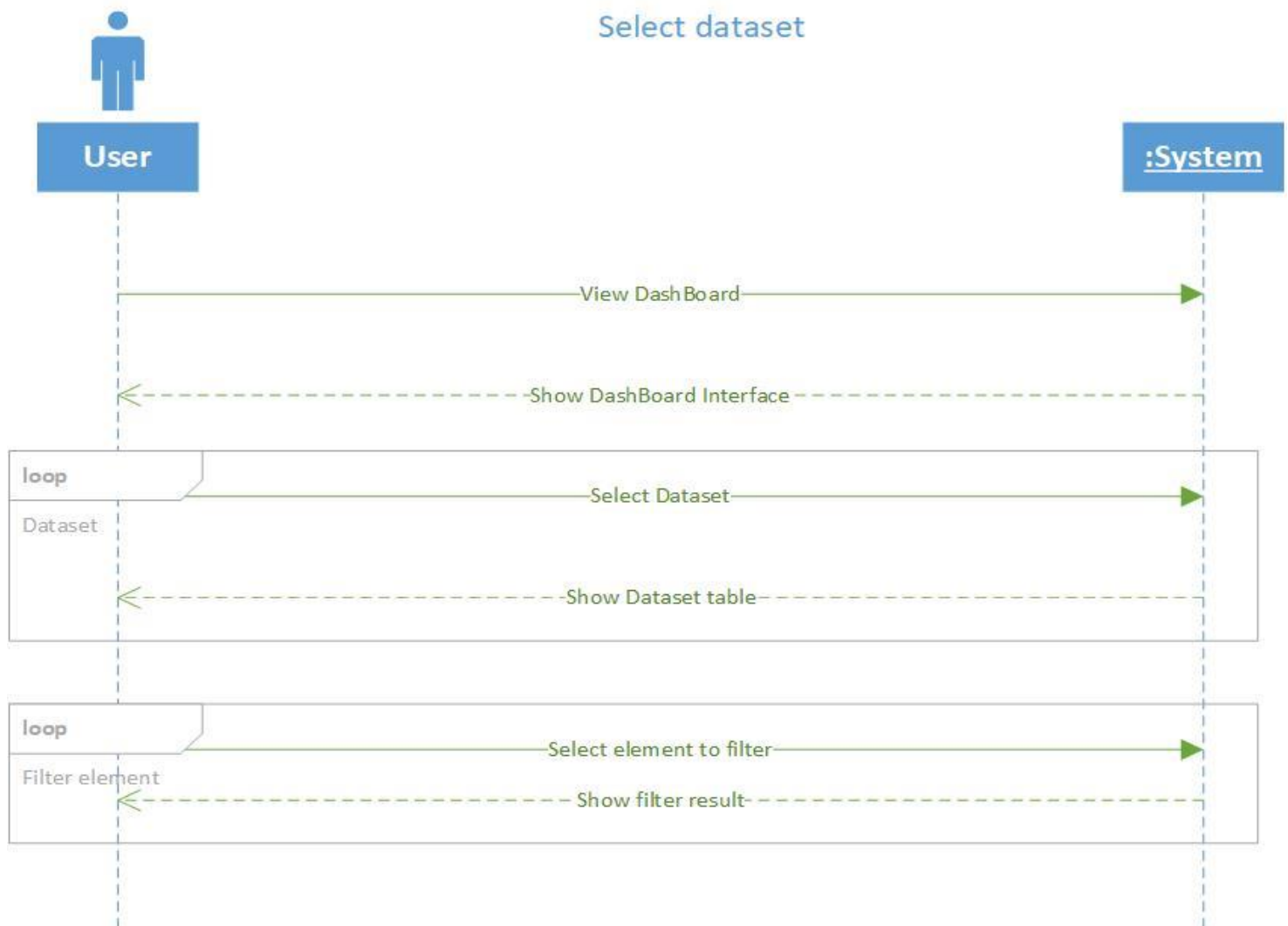
There are some sub-functions in use cases 'select dataset', 'select chart type' and 'create the statistic report of data'. Such as select specific dataset in 'select dataset', select bar chart, pie chart and pivot chart in 'select chart type', and show min, mean, and max in 'create the statistic report of data'. But these sub-functions can be treated as values, so they do not need to be shown in use case diagram.

4. Activity Diagrams



5. Sequence Diagrams

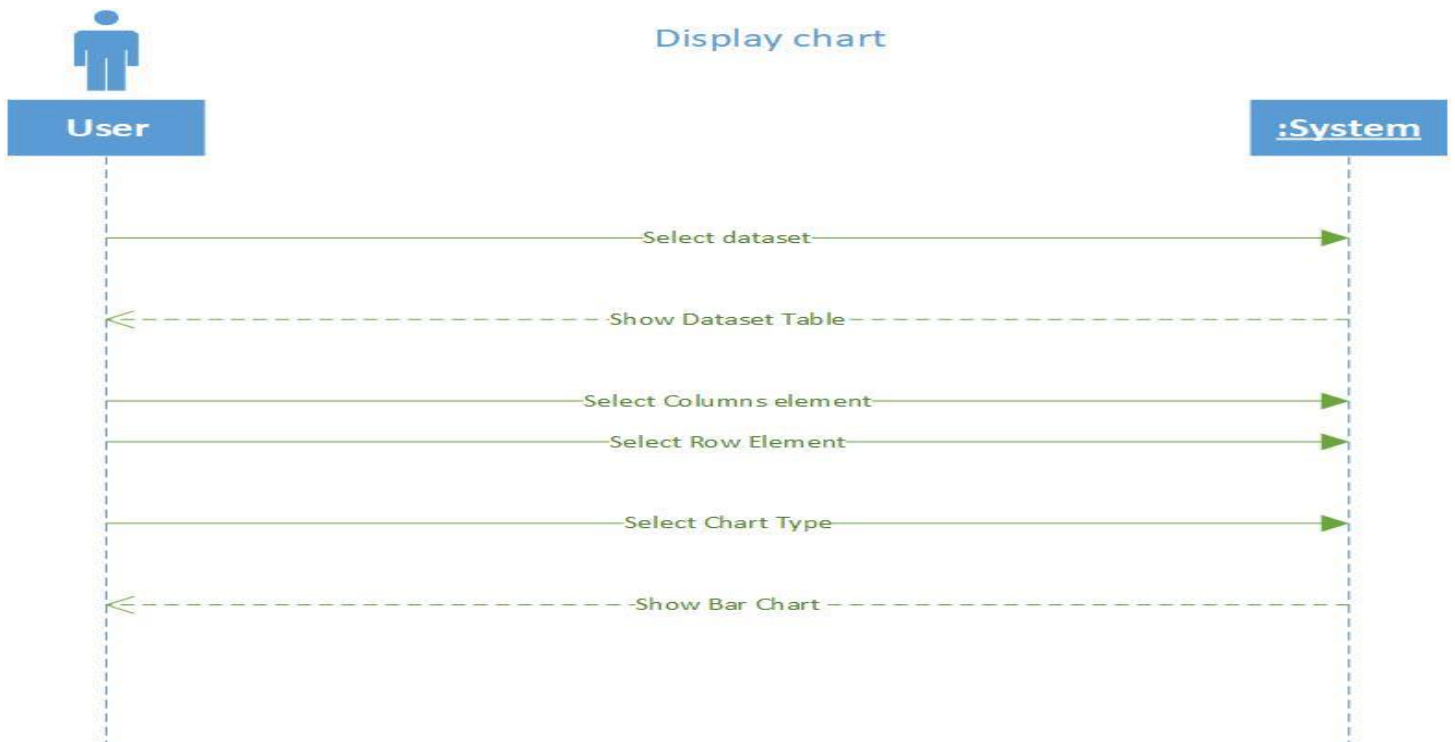
(1) select dataset



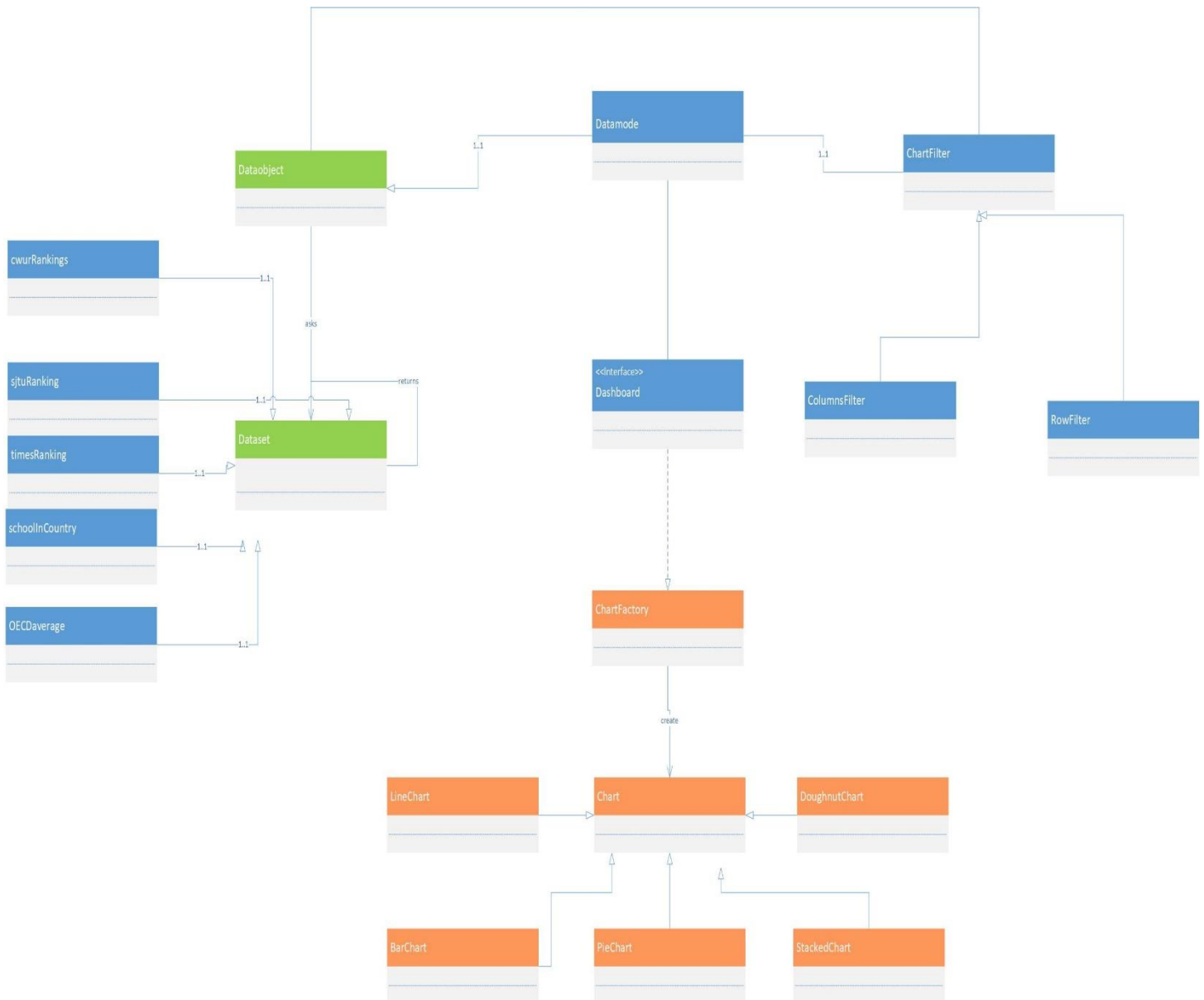
(2) Show data in descending order



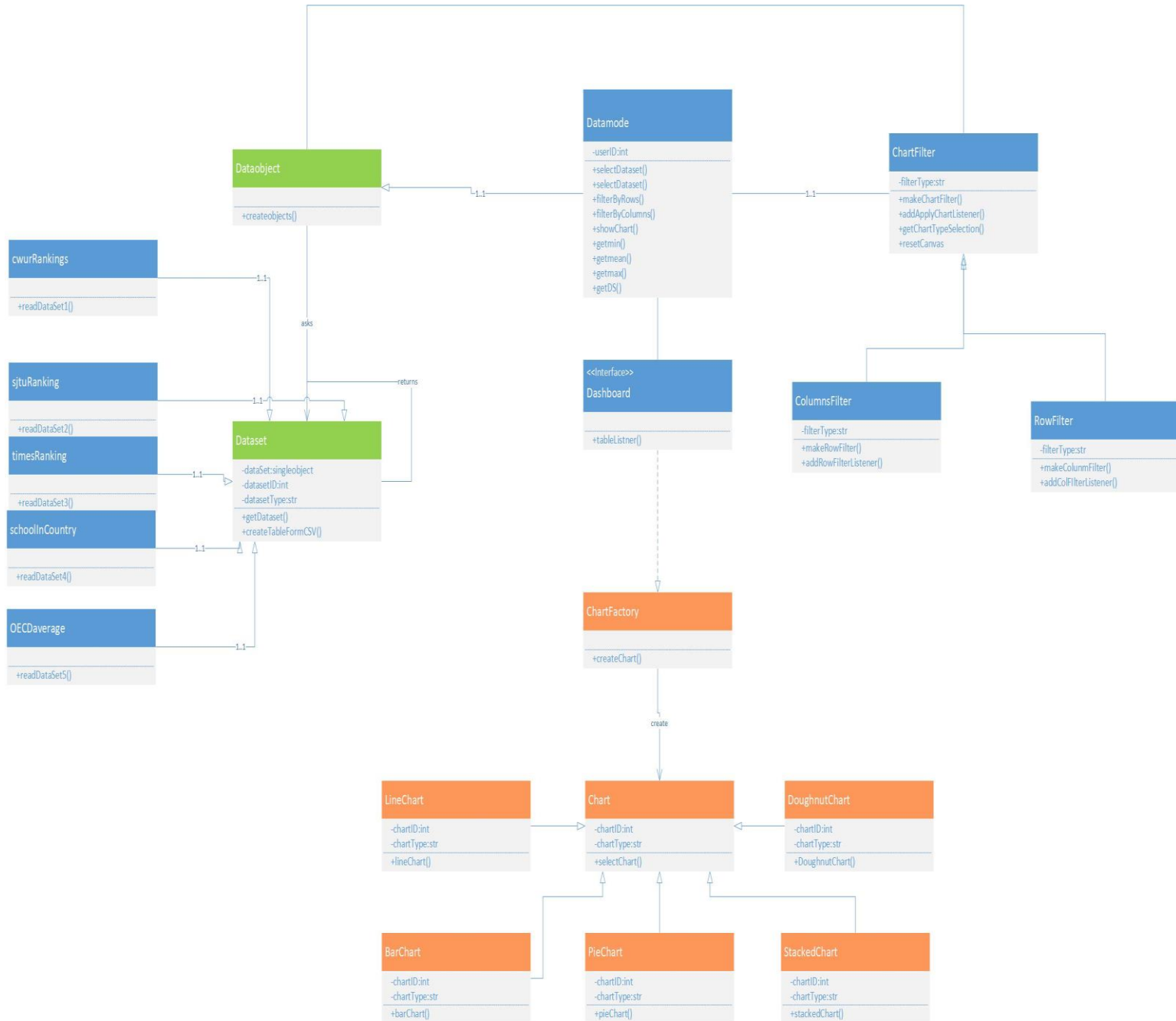
(3) Display chart



6. Domain Model Class Diagram

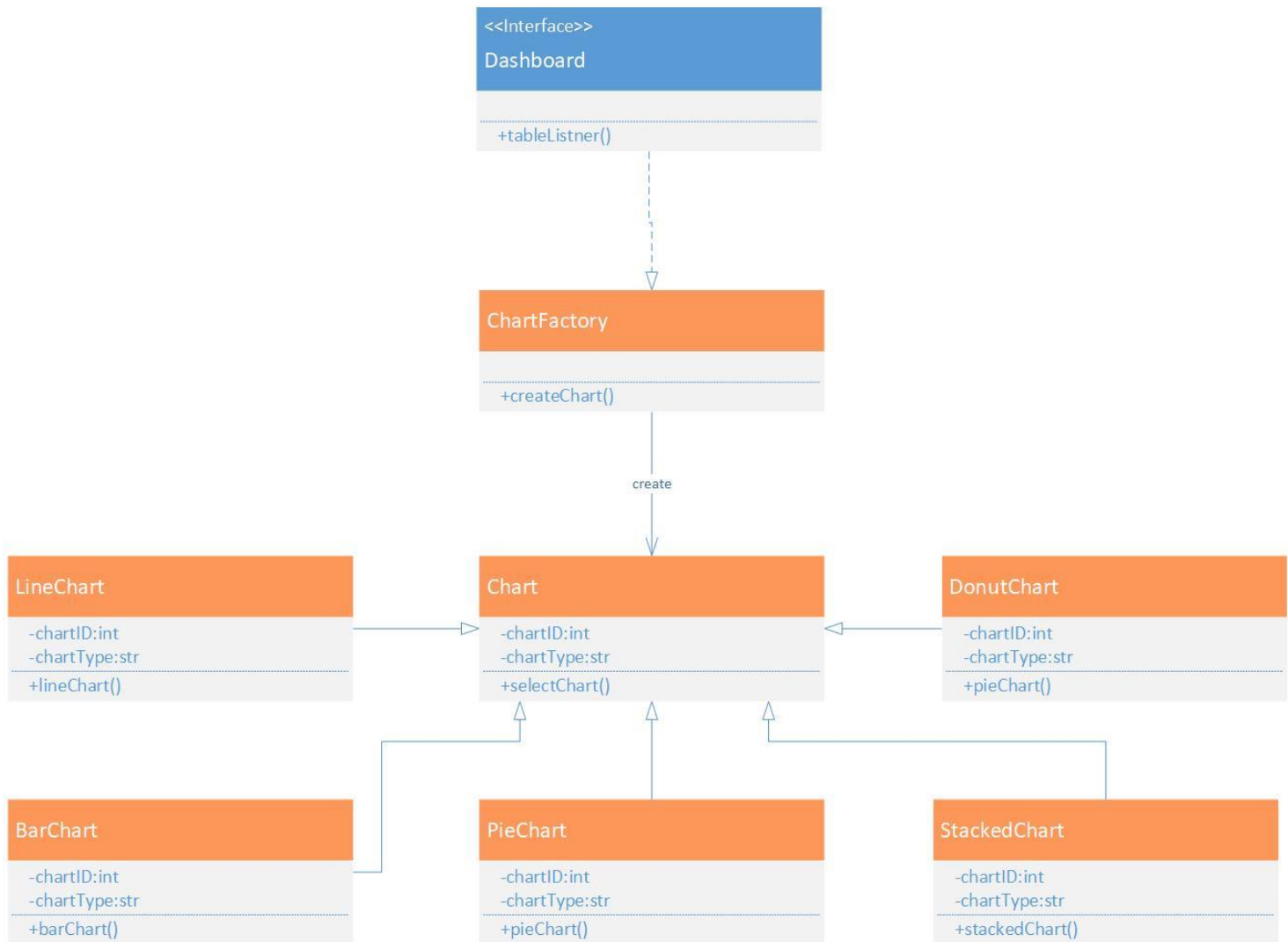


7. Design Model Class Diagram



8. Documentation and class diagrams for Design Patterns used

(1). Factory method design pattern

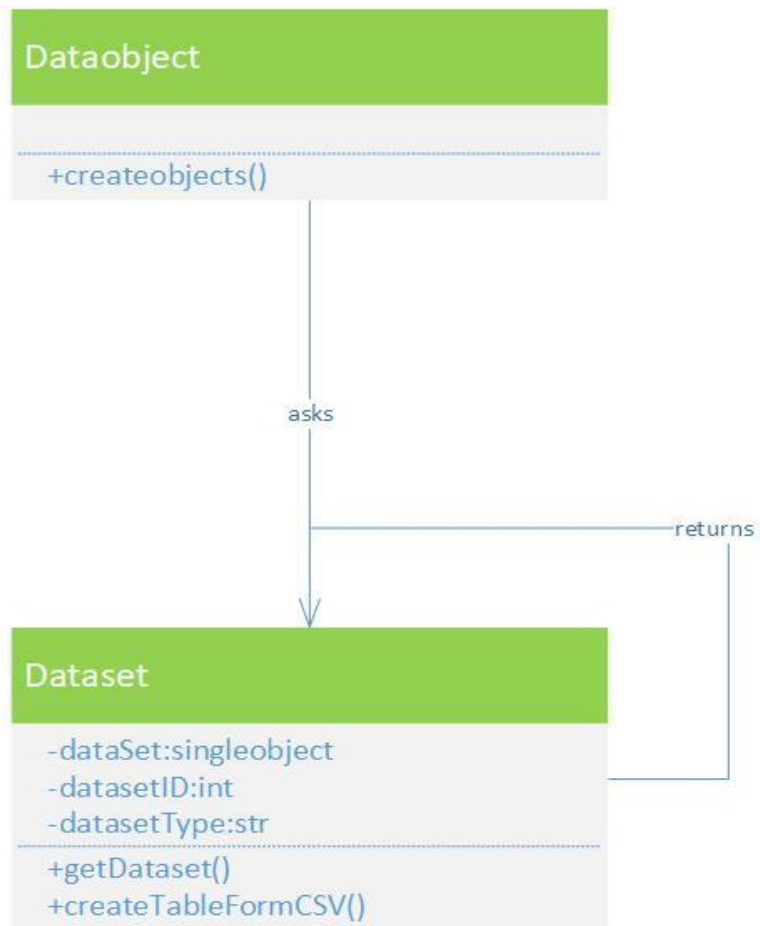


This “Factory Method Design Pattern” is used for creating an chart object and the subclasses decide which class to instantiate.

We select 6 types chart opinions, since we pass the type value to the chart class and based on the type of chart, we instantiate the correct concrete class and assign it to the chart instance variable.

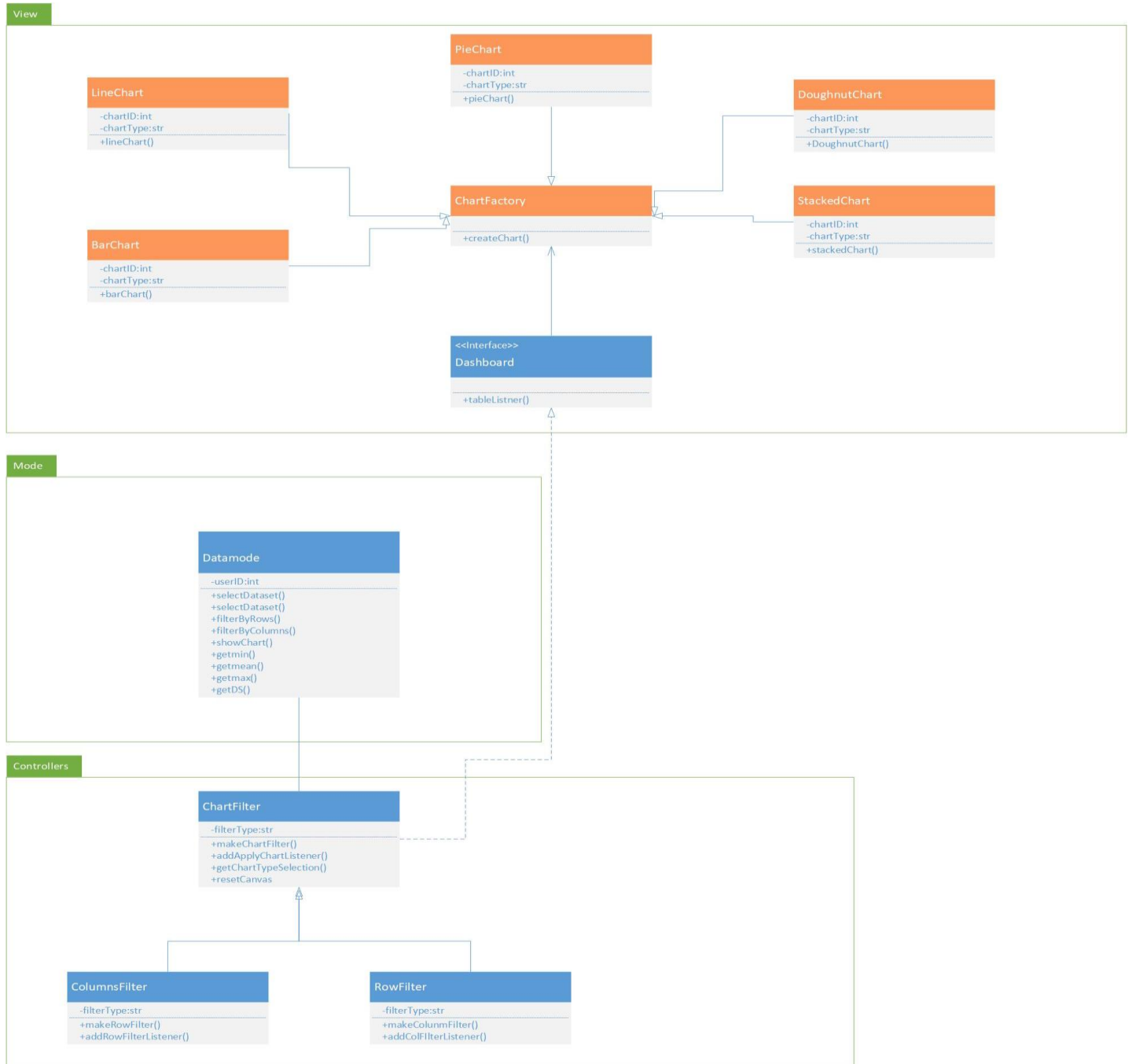
All the methods are provided by the createChart() method which is super class acting as an factory method.

(2). Singleton design pattern



The Dataset class creates only one instance of a dataset for further operations such as data load, filter and chart showing. This provides a global point to access to it. The system will know which dataset has been loaded in dashboard

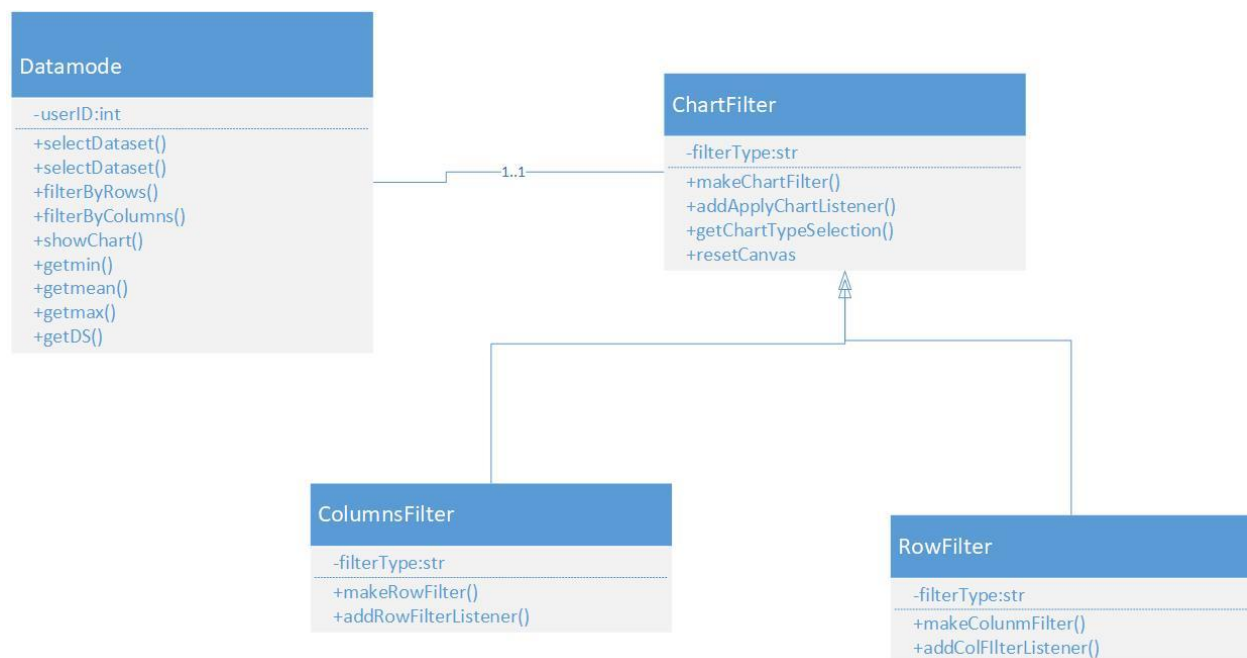
(3). MVC architectural design pattern



MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

- **Model** – Datamode represents an object carrying data. It can also have logic to update controller if its data changes.
- **View** – The class in view represents the visualization of the data that model contains.
- **Controller** - Chartfilter acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

(4). Iterator design pattern



the iterator pattern is a design pattern in which an iterator is used to traverse a container and access the container's elements. Because we need to select rows and columns or deselect it. We could use ChartFilter to traverse rows or columns in different ways. Meanwhile, we can traverse data many times in row and columns.