

For the design pattern, I used the Adapter pattern and implemented it with JAVAfx. This pattern is self-explanatory as its purpose is acting like an adapter to connect two incompatible interfaces that otherwise can't be connected directly. It acts like an adapter for example, a headphone jack to lightning adapter for iPhone, this allows user to still listen to songs through a wired headphone even though many smart phone companies are opting to remove the 3.5mm headphone jack/port.

My approach to use this pattern to my app was to create an app that was based in United States and so I used TESLA for my app. Since Tesla is headquartered in US, they follow Imperial system (mph, \$) for listing their cars specs. As we all know TESLA is now a big company, they would need to expand their network to get more clients and so I made the app to accommodate clients from United Kingdom. So, whenever a client uses my app they'll be required to select an option for which country they are logging in from. As soon as the client hits their country button the UI takes them to a new page/scene which consists of information about the TESLA's four famous cars (models S,3,X,Y) and their specs. The specs on this page are in metric system (£, kmh) if the user selects "UK". Thus, the use of Adapter pattern was implemented in my work/app.

Advantages: I think the main advantage I found was I didn't had to write new code separately for the clients from UK and no modifications to the original class/code was done. This is a huge advantage as we can add/remove new clients and be more user friendly for customers from other countries. In addition, we could reuse our original code and this brings a lot flexibility to both end users and developers. Not to mention, we could use polymorphism to move between

multiple implementations of adapters providing much more flexibility than writing new classes/modifying code.

Disadvantages: I haven't found any disadvantage so far when doing the homework. However, during my research I found that in some circumstances, developers might have to write multiple adaptations (adapter chain) to reach the type which is required.