CS5200 Homework 2

Jinlei Kuang

1 Class Diagram

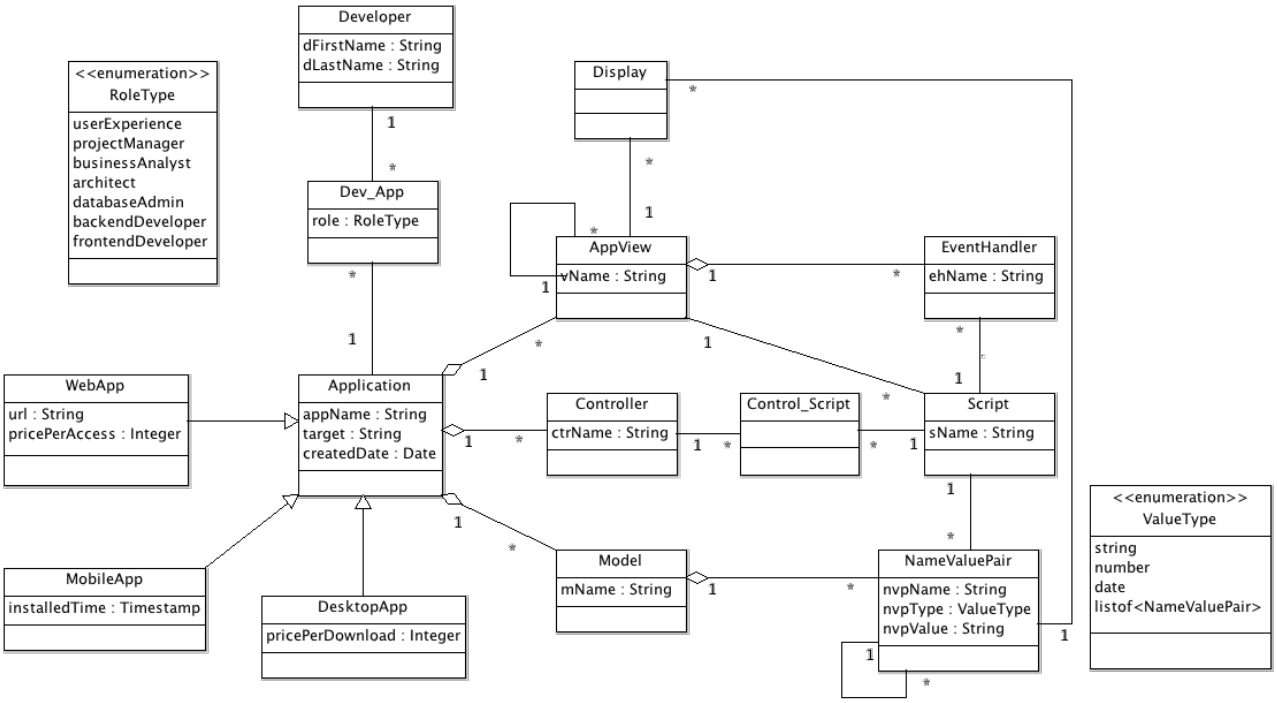

Figure 1

2 Analysis

2.1 Generalization/Specialization

According to the requirement spec, Web applications, Mobile applications and Desktop applications are three different types of Applications. So each of them *is a* application, which makes Application a base class, and WebApp, MobileApp and DesktopApp are all subclasses of Application.

2.2 Enumeration

As we can see from the requirement spec, the roles of the developers and the value types of the name value pairs are both limited and fixed. There's no other possibilities of their values in this system. So we can use two enumerations to represent these two types.

2.3 Reify Many-Many Relationships

There're two Many-Many relationships in total:

- Controller-Script

- AppView-Data(Name Value Pair)

So we need to reify these relations by adding an extra class between the two sides of each one.

2.4 Reify Multi-Way Relationships

According to the requirement spec, the relationships between Developer, Role and Application are multi-way. So we reify it by conecting both Developer and Application to Role, to make two many- one relationships.

2.5 Aggregation/Composition

There're 5 relationships that might be Aggregation or Composition.

- Application to AppView
- Application to Controller
- Application to Model
- AppView to Event Handler
- Model to Name Value Pair

The first 3 relations are the same, and they don't necessarily have lifecycle dependencies. So these three are Aggregation; a event handler is a part of a view, so their relationship is also Aggregation; if we delete a model, its name value pairs can still be used in other model, so Aggregation, too.
By the way, considering the restoration after deleting. Therefore, all the five relationships should be Aggregation.

2.6 Inadequate/Redundant relationships

- By checking the Class Diagram shown in Figure 1, we can see that all the relationships are clear and there're no multi-way relationships. So there're no inadequate relationships.
- In the cycle of AppView, EventHandler, and Script, the event handler in the view invoke scripts, but script may define a total different view from the one that invoke it. Under such circumstance, we cannot treat any of these three relationships as a redundant relationship.