1st Approach:
Advantages: Separation of the components makes it easier to debug and modify the code if problems arise.

Disadvantages: There is a close coupling of views and controllers to a model. There can be an excessive amount of updates.

2nd Approach:
Advantages: This approach helps to make the interfaces more loosely coupled. Using interfaces also helps to reduce the complexity of the design. It also allows the code to be agnostic of the type of view. If more types of views are added later, the base code doesn't have to change.

Disadvantages: Using the MVC approach when applied to components that are getting constantly updated can cause the data display to slow down. This also makes the interface highly dependent on the model component.

3rd Approach:
Advantages: The use of the observer pattern allows for there to be loose coupling between the components it interacts with. It also allows large numbers of views to be updated at once, without it having to matter what types of views they are.

Disadvantages: The problem with using the observer pattern in this case is that the components are not guaranteed to be in any given order, this can cause issues such as the ticket printing before it is displayed on the display.