MVC is an architectural pattern utilized to help separate an application into three separate parts: A model, a view, and a controller. Each component has its own role to play. It is very popular in web development due to its ability to be scaled up into very large projects.

The model part of this architectural design refers to the logic that happens using the data between the view and the controller. For creating an account for example, the model would take a username and password, store it in a database, and take the user to the next page. This would be the logic that happens according the user input data.

The view is the user interface component of the architectural design. This has to do with all of the visuals that the user sees. For example, the customer side of the view would show consumer functions. For a bank, this would be things like make a deposit, pay a card, view statements, etc.

The controller is where all the magic happens. It works closely with the model and view to process the incoming requests from each. It manipulates the data input from the user in the view, and changes the view accordingly in order to present the desired function to the user according to the model. When the database is updated, the model is utilized to ensure that the data is going to the correct place.

The MVC architecture paves the way for a faster development process. Multiple parts of the MVC could be worked on at the same time by multiple programmers, speeding up the development process. It also provides multiple views, so code does not have to be replicated for pc to mobile device. Also, the modification of different parts of this model do not change the other. For example, the model could be modified without necessarily having a change in the view aspect of the MVC.

Some disadvantages to MVC are the complexity aspect. As the project keeps growing, the complexity grows with it and it gets harder to access. Frequent updates can also cause the view to take longer to make. The view can also fall behind all the updates because of this. The last drawback is that the MVC architecture has some strict rules over the methods. The controller keeps a keen eye on the events based off the view and tries to complete the appropriate actions to each event. These events don’t have strict access, which turns out to be a big drawback of the MVC