My system is a basic shop and inventory management system.

I used ScriptableObjects to handle the Items, as this makes easier for designers to come up and create new items to the game.

It started very simple, using just primitives, I used a square for the player and a circle for the shop. Using programming principles, I tried to make sure that classes were only capable of handling what each class should.

At first it was simply opening the shop and seeing the items available. I had never built an UI like this before, so at first it was a challenge to make the information on the screen was not static, such as the amount of items on the store.

I created a class for handling the basics of an item line in the shop, an ItemTemplate, that made things easier on how to approach the rest of the shop iteration.

When implementing how to equip items, at first I made into a bool, checking if the item was equipped or not. As this could make the game harder to scale over time, this was updated by using a new class EquippedItem class, and using an enum for item types, to make sure you could have different items equipped, such as headgear and chest.

By using an event system, I could easily handle how the buttons of each item worked on the UIManager. As the game only contains one canvas, I turned my UIManager into a Singleton, for global access and single point of control.

For someone who had never built an UI like this, I believe I was able to deliver a functioning and clean system, that is easily scalable. By using ScriptableObjects and designing a system using the ItemTemplate, and the enum for items, this project could easily be expanded. Due to the short time of the interview, it was setup to complete the most basics tasks it had to, but the thinking and development of the system, had in mind a greater and bigger system, that could handle its size and operations easily.