This project leverages the proprietary TsukiSuite and Tsuki2D libraries, both of which are my personal creations. It's important to highlight that Tsuki2D is not available for public use. I've included it in this context solely to demonstrate my competencies and technical prowess as a software engineer. With respect to the exclusive nature of this work, I kindly request that you do not distribute it. Your understanding and cooperation in maintaining the confidentiality of Tsuki2D are greatly appreciated.

I focused on creating a robust and engaging inventory system that would not only serve as a foundational gameplay mechanic but also enhance the overall user experience through interactive elements and seamless integration with other game systems.

The architecture of the inventory system is built upon several key components, including Inventory, InventoryHandler, and various specific handlers like BuyInventoryHandler, SellInventoryHandler, and EquipInventoryHandler. Each component plays a crucial role in managing the game's items, transactions, and player interactions.

The Inventory class is central to the system, designed to hold items and manage their addition and removal. It supports dynamic inventory sizes and includes methods for item stacking, ensuring a fluid and intuitive inventory management experience. Random item generation at the start of the game introduces an element of surprise and variety, enriching the gameplay from the very beginning.

The abstract class InventoryHandler and its concrete implementations (BuyInventoryHandler, SellInventoryHandler, and EquipInventoryHandler) encapsulate the actions a player can perform with items, such as purchasing from a shopkeeper, selling items back, or equipping items. This design allows for easy expansion and modification of inventory interactions, showcasing the system's flexibility and scalability.

The Interactable and InteractionAttachment classes facilitate player interaction with the game world, enabling players to engage with entities and items within the environment. This interaction mechanism is crucial for creating an immersive and dynamic game world where players can discover, interact, and manipulate objects and characters.

On a personal note, assessing my performance in developing this system, I believe I have demonstrated a strong understanding of object-oriented design principles and a keen eye for user experience. By employing inheritance and abstraction, I created a system that is both extensible and maintainable. The use of serialized fields for configuration through the Unity Editor enhances the system's flexibility, allowing for easy adjustments and testing.

However, there is always room for improvement. In retrospect, I could have optimized the system's performance further by refining the search algorithms used in inventory management and exploring more efficient data structures for storing and accessing items.