

## Explanation of Data Analysis Layer (UML Model diagram)

At the heart of our system lies the `PatientDataGenerator` class, which serves as the fundamental building block for generating patient data. Whether it's for analogue simulation or real-world applications, this class ensures that the system can cater to various requirements. The `generate` method, which takes an integer as input, signifies the ability to dynamically adjust the quantity of data being generated, thereby offering unparalleled flexibility in data creation.

To further enhance the system's adaptability and maintainability, we've introduced the `OutputStrategy` interface and its implementations, `TcpOutputStrategy` and `TcpInputStrategy`. This implementation of the Strategy Pattern allows us to seamlessly switch between different output methods, depending on the current needs or preferences. Both strategies make use of `ServerSocket` and `Socket` interfaces, emphasizing the system's robust network capabilities for secure and reliable data transmission.

The `AlertGenerator` class, equipped with its `generate` method, underscores the system's proficiency in generating alerts based on predefined criteria or thresholds. This functionality is crucial in any monitoring or safety-critical system, enabling quick and effective responses to potential issues or irregularities.

Lastly, the `SimpleWebSocketServer` class, which inherits from both `WebSocketOutputStrategy` and `WebSocketServer`, exemplifies the integration of WebSocket technology. This technology facilitates real-time data transmission, enabling efficient, bidirectional communication between the system and its users. This design choice not only enhances the user experience but also ensures that modern, interactive applications can leverage the system's capabilities to their fullest.

In summary, the design choices of our system reflect its flexibility, scalability, and adaptability. It can easily adapt to various output strategies and seamlessly integrate with modern communication protocols, enabling efficient data transfer and superior performance.