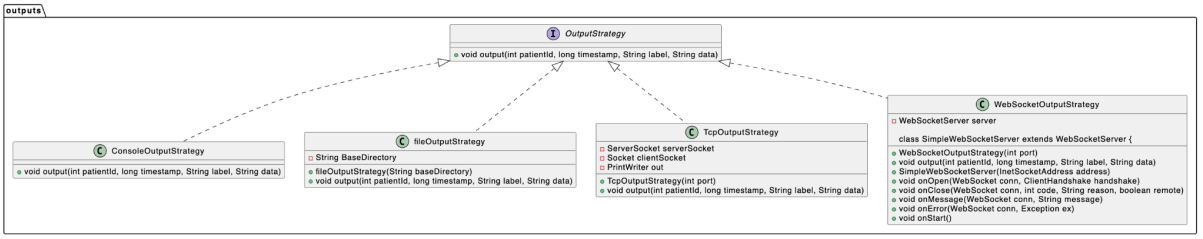
Output System UML



The com.cardio_generator.outputs package is responsible for defining various strategies to output patient data generated by the health data simulator. Each output strategy implements the OutputStrategy interface, ensuring a consistent method for outputting data regardless of the specific mechanism used. The package includes classes for outputting data to the console, files, TCP sockets, and WebSocket connections.

The OutputStrategy interface defines the contract for outputting patient data. Implementing classes must provide a method to output patient data, including patient ID, timestamp, label, and data. The void output(int patientId, long timestamp, String label, String data) method outputs the data for a given patient.

The ConsoleOutputStrategy class implements the OutputStrategy interface to output patient data to the console. It uses the System.out.printf method to format the output. The void output(int patientId, long timestamp, String label, String data) method outputs data to the console in a formatted manner.

The fileOutputStrategy class implements the OutputStrategy interface to output patient data to files in a specified directory. It uses a ConcurrentHashMap to manage file paths for different labels and ensures that data is appended to the correct file. The fileOutputStrategy(String baseDirectory) constructor initializes the base directory for output files. The void output(int patientId, long timestamp, String label, String data) method outputs data to files, creating directories and files as needed, and appending the data in a formatted manner.

The TcpOutputStrategy class implements the OutputStrategy interface to send patient data over TCP sockets. It initializes a TCP server socket on a specified port and accepts client connections to send the data. The TcpOutputStrategy(int port) constructor initializes the TCP server socket on the given port and handles client connections. The void output(int patientId, long timestamp, String label, String data) method sends data over the TCP connection to connected clients, formatting it appropriately.

The WebSocketOutputStrategy class implements the OutputStrategy interface to send patient data over WebSocket connections. It initializes a WebSocket server on a specified port and broadcasts data to all connected clients. The WebSocketOutputStrategy(int port) constructor initializes the WebSocket server on the given port. The void output(int patientId, long timestamp, String label, String data) method broadcasts data to all connected WebSocket clients, formatting it appropriately.

The SimpleWebSocketServer is a private nested class within WebSocketOutputStrategy that extends WebSocketServer to handle WebSocket connections. It manages the lifecycle of connections, including opening, closing, receiving messages, and handling errors. The SimpleWebSocketServer(InetSocketAddress address) constructor initializes the WebSocket server with a given address. The void onOpen(WebSocket conn, ClientHandshake handshake) method handles new connections. The void onClose(WebSocket conn, int code, String reason, boolean remote) method handles closed connections. The void onMessage(WebSocket conn, String message) method handles incoming messages (not used in this context). The void onError(WebSocket conn, Exception ex) method handles errors during WebSocket communication. The void onStart() method indicates that the server has started successfully.