Background

You are a lead software engineer at **healthcare solution**, a company specializing in healthcare management systems. The team is responsible for developing and maintaining a comprehensive application used by medical professionals to manage patient records, appointments, and billing. Recently, a new requirement emerged: to support multiple types of medical services, each with distinct billing rules and appointment scheduling logic.

Scenario

The current implementation of the billing and scheduling system has become unwieldy. The ServiceManager class, which handles both billing and appointment scheduling, has grown significantly as new types of medical services have been added. As a result, the class now contains a mix of responsibilities, making it challenging to understand, maintain, and extend.

Challenges

1. **Mixed Responsibilities:** The ServiceManager class handles both billing and appointment scheduling, which are distinct responsibilities. This makes the class large and complex.
2. **Difficulty in Extending:** Adding support for new service types or altering billing and scheduling rules requires extensive modifications within the ServiceManager class.
3. **Maintenance Issues:** The class has become difficult to maintain due to its size and the complexity of its methods.

4. **Testing Difficulties:** The combined logic for billing and scheduling makes it challenging to write focused unit tests for either aspect.

Goal

To refactor the existing code to improve maintainability and extendibility by separating concerns and simplifying the logic. This will involve reorganizing the code to handle billing and scheduling responsibilities in a more modular way.