

# Case: Hospital Appointment Scheduling

## Objective

As part of your interview process, we ask you to complete a coding exercise to evaluate your **coding skills** in terms of **correctness, performance, maintainability, extensibility, and security**. This exercise will also serve as a basis for discussion during your interview.

## Background

The hospital is planning to expand its operations by **adding more departments** and introducing **advanced booking rules**. However, the current implementation has significant technical debt, making it **difficult to maintain** and extend. Specifically, adding new departments or department-specific rules requires changes to the core service, which **increases complexity**, reduces **maintainability**, and slows down development. A more flexible and modular approach is required to ensure the system can easily adapt to future growth and evolving requirements without extensive rework.

## Task

Your goal is to enhance the existing codebase to address the following requirements:

- Redesign the current implementation to ensure that new departments and their specific validations (e.g., **RequiresReferral**, **RequiresInsuranceApproval**) can be added **without modifying** the core **AppointmentService** logic.
- Refactor the department-specific rules into a **modular** and **reusable** structure.
- Ensure appropriate **error messages** are displayed to the user in case of failures.
- Ensure the system **gracefully handles** unsupported departments or missing validation configurations.

## Notes

- You should spend no more than a few hours on this task. Focus on demonstrating your understanding of clean code and extensible design.
- Use appropriate tools and frameworks to streamline the development process and focus on the core requirements.
- This is a conceptual exercise. Do not spend time on unit testing or implementing placeholder methods (e.g., **IsAssignedToGP**, **IsValidReferral**). These methods are provided as examples and do not need to be fully implemented.
- Follow proper coding practices to showcase your understanding of maintainable and modular code structure.
- Ensure that your solution can be built and run without modification by the evaluators. Provide clear instructions if any setup is required.

## Submission

Add the code to a **Git repository** and send the link to **Zlatimir.Zahariev@getinge.com** and **Jesper.Smith@getinge.com** at least **24 hours before your interview**.