Justifications

Patterns used:

MVP architectural pattern is used to reduce the tight coupling between the View and Business Logic. Dependency Injection is used in the technical implementation to develop a loosely coupled code. Unity container is used for registering the dependencies and resolving the dependencies. Constructor injection is used in the technical implementation.

Patterns not used:

Factory pattern could have been used to create the reception staff, Dentist and Patient profile object but all these were anaemic due to the implementation of the Entity Framework and thus it was decided not to add any unnecessary complexity.

No repository pattern followed specifically since the DBcontext in the entity framework served the purpose.

All the business logic for Patients such as Appointment, Medical History and Treatment plan was included in the PatientBL class and this could have been moved to separate BusinessLogic classes such as AppointmentBL, MedicalHistoryBL and TreatmentPlanBL in order to satisfy single responsibility principle. As there was not much functionality except the CRUD operations, it was decided to include everything in PatientBL.