**Project Name**: PatientSky

**Document name**: PatientSey app

**Date**: 03-06-2024

**Version**: 0.1

**Author**: Prashantha

1. Introduction

This document serves as a guide for the design and development of PatientSky app. It outlines the project's objectives, scope, requirements and design details.

1. Project Overview

Overview of the project is to list available times for a meeting for several participants.

1. Requirements

AppointmentData, Timeslot and doctors list with corresponding information.

1. Design Details

* Create a .Net core webAPI project (EG\_PatientKeyAPI)
* Create folder to store named PatientSkyData under the project and store the appointment and timeslot input data in the json format.
* Create a new Model folder and create below class file with required fields as below
  + Class Appointment.cs
    - Id (string)
    - Patient\_id(string)
    - CalenderId(string)
    - Start (Datetime)
    - End (Datetime0
    - PatientCommet (Object)
    - Note (Object)
    - TimeSlotTypeId(string)
    - TypeId (Object)
    - State(int)
    - OutOfOfficeLocation(string)
    - OutOfOffice (Boolean)
    - Completed (Boolean)
    - IsScheduled (Boolean)
  + Class Doctor.cs
    - CalenderId (String)
    - FirstName (String)
    - LastName(string)
  + Class TimeSlot.cs
    - Id(string)
    - CalenderId(string)
    - TypeId(string)
    - Start(Datetime)
    - End(Datetime)
    - PublicBookable(boolean)
    - outOfOffice (Boolean)
* Create below DTO classes with fields mentioned below under Model folder to manage input and output results.
  + AppointmentCheckDTO
    - CalenderID(string)
    - Duration(int)
    - Start (Datetime)
    - End (Datetime)
  + AvailableTimeModelDTO
    - RequestedTimeWithDoctor(int)
    - AvailableFromTime (Datetime)
    - AvailableToTime (Datetime)
    - DoctorUnavailableFrom (Datetime
  + AvailableAppointmentsDTO
    - Doctor (Doctor class type)
    - AvailableTimeModelDTO
    - List<TimeSlot>
  + AvailableTimeSlot
    - TimeSlotId(string)
    - TypeId(string)
    - Start (Datetime)
    - End (Datetime)
  + TakenAppointmentsDATO
    - Doctor (of type Doctor class)
    - Appointment (of type appointment class)
* Create Data folder under project and add a new class AppDbcontext.cs and do the following.
  + Create list object of Appointment, Timeslot and Doctors type and inside the constructer assign value from PatientSkyData folder’s json files into the corresponding list object.
  + For the doctors list manually assign the values by add new list item of type Doctor
* Create a folder names Services under the project and add a new service class with AppointmentService.cs and do the following.
  + Create new object for AppDbContext and implement below methods.
    - GetTokenAppointments (): Retrieve all taken appointments within the date range and calenderId.
    - GetTimeSlotInTimePeirod (): This method is to get taken appointmentsfrom the defined start, end, and list calenderIds and duration.
    - GetAvaliableAppointments (); Method to get available appointmentsfrom defined start, end date, list of taken appointments, timeslots and duration.
    - CreateAvailableAppointments (): Method to create available appointments.
* Create new AppointmentController under controller folder and do the following.
  + Create a readonly object of AppointmentService type.
  + Create a constructor which takes AppointmentService as a parameter and assign the value to the readonly object created above.
  + Write a Post method AvailableDoctors which takes appointmentCheckDTO as input and returns availableAppointments. Keep the return type as IActionResult.