# Class Design Document

*Clinic Appointment & Patient Management System*

**Group B**

Team Members: Halimatu Sadia Mohammed & Hanif Olayiwola

This document describes the object-oriented design of the Clinic Management System, including all classes, their attributes, methods, and relationships.

## 1. Patient Class

**Purpose:** Represents an individual patient in the clinic system.

### Attributes:

- patient_id (str): Unique identifier (e.g., P001, P002)
- name (str): Full name of the patient
- age (int): Patient's age in years
- contact (str): Phone number for communication
- gender (str): Male or Female

### Methods:

- __init__(patient_id, name, age, contact, gender): Constructor to initialize patient
- __str__(): Returns formatted string representation of patient details
-

## 2. Appointment Class

**Purpose:** Represents a scheduled appointment between a patient and doctor.

### Attributes:

- appointment_id (str): Unique identifier (e.g., A001, A002)
- patient_id (str): Links to Patient object
- doctor_id (str): Links to Doctor object
- date (str): Appointment date in YYYY-MM-DD format
- time (str): Start time in HH:MM format (24-hour)
- duration (int): Length of appointment in minutes
- department (str): Type of appointment (Dental, X-Ray, Physio, etc.)
- purpose (str): Reason for the visit
- status (str): 'Booked' or 'Cancelled'

### Methods:

- __init__(appointment_id, patient_id, doctor_id, date, time, duration, department, purpose, status): Constructor
- get_end_time(): Calculates and returns end time based on start time and duration
- __str__(): Returns formatted string representation of appointment details

## 3. Doctor Class

**Purpose:** Represents a doctor in the clinic with availability information.

### Attributes:

- doctor_id (str): Unique identifier (e.g., D001, D002)
- name (str): Doctor's full name
- specialty (str): Medical specialty (e.g., Dentistry, Cardiology)
- available_days (list): Days doctor works (e.g., ['Mon', 'Wed', 'Fri'])
- start_time (str): Work start time in HH:MM format
- end_time (str): Work end time in HH:MM format

### Methods:

- __init__(doctor_id, name, specialty, available_days, start_time, end_time): Constructor
- is_available_on_day(date): Checks if doctor works on given day
- is_within_working_hours(time): Checks if time is within working hours
- __str__(): Returns formatted string representation of doctor details

## 4. ClinicManager Class

**Purpose:** Main system class that coordinates all operations and manages data.

### Attributes:

- patients (list): List of all Patient objects
- doctors (list): List of all Doctor objects
- appointments (list): List of all Appointment objects

### Methods:

***Validation Methods:***

- patient_exists(patient_id): Checks if patient ID exists
- doctor_exists(doctor_id): Checks if doctor ID exists
- get_doctor_by_id(doctor_id): Retrieves doctor object by ID
- check_doctor_availability(doctor_id, date, time): Validates doctor availability
- slot_available(doctor_id, date, time, duration): Checks for scheduling conflicts
- get_patient_name(patient_id): Returns patient name by ID

***Patient Management:***

- add_patient(): Registers a new patient
- search_patient(): Searches patients by ID or name
- show_patients(): Displays all patients in table format

***Doctor Management:***

- show_doctors(): Displays all doctors with schedules
- search_doctor(): Searches doctors by ID, name, or specialty

***Appointment Management:***

- book_appointment(): Books a new appointment with validation
- cancel_appointment(): Cancels an existing appointment
- reschedule_appointment(): Changes appointment date/time
- show_appointments(): Displays all appointments
- search_appointment(): Searches appointments by patient or doctor