**Boost Physio Clinic – Appointment Management System**

**Final Report**

**Submitted By**Mohit Rathor

Student ID: 23058531

**Module**

7COM1025-0105-2024 - Programming for Software Engineers

**Program**

MSC Advanced Computer Science

Table of Contents

[1. Introduction 3](#_Toc196155725)

[2. System Functions 3](#_Toc196155726)

[2.1. Patient Management 3](#_Toc196155727)

[2.2. Physiotherapist and Slot Management 3](#_Toc196155728)

[2.3. Appointment Booking 4](#_Toc196155729)

[2.4. Cancellation and Attendance 4](#_Toc196155730)

[2.5. Rescheduling Appointments 4](#_Toc196155731)

[2.6. Generate Report 4](#_Toc196155732)

[2.7. Error Handling 4](#_Toc196155733)

[3. Design and Implementation 5](#_Toc196155734)

[3.1 Object-Oriented Principles 5](#_Toc196155735)

[3.2 UML Class Diagram 5](#_Toc196155736)

[3.3 Class Designs 6](#_Toc196155737)

[3.3 Prevention of Logical Errors 7](#_Toc196155738)

[3.5 Data Structure Use 7](#_Toc196155739)

[3.6 Design Patterns and Principles Used 7](#_Toc196155740)

[4. Testing and Validation 8](#_Toc196155741)

[5. Version Control and Project Snapshots 9](#_Toc196155742)

# 1. Introduction

# A Java-based program called the Boost Physio Clinic (BPC) System was created to make running a physiotherapy clinic easier. It facilitates the effective management of treatment plans, appointment scheduling, and patient records. Errors are minimized, duplicate bookings are prevented, and available treatment times are clearly visible thanks to the system. BPC is a command-line tool that allows users to create reports, add patients, and make or cancel appointments. It handles time and data structures using the built-in Java libraries. The system makes use of the concepts of object-oriented programming. Enums are used to handle appointment statuses for control and clarity. All things considered, the BPC System is a well-organized and effective way to run a physical therapy clinic. It can be enhanced with features like a graphical user interface and is dependable and simple to use.

# 2. System Functions

## Numerous features are available in the BPC System to help the clinic run smoothly. These include scheduling appointments, dealing with cancellations and attendance, managing patients and physiotherapists, preventing errors, and producing reports. Together, all of the features make sure that users may utilize the system in an understandable, effective, and error-free manner.

## 2.1. Patient Management

## By enabling users to add new patients with automatically created unique IDs, the system manages patient care. The registration process is short and easy because only the patient's name, address, and phone number are needed. Any record can be deleted by users if necessary.

## 2.2. Physiotherapist and Slot Management

## The system has specified physiotherapists, each of whom is linked to a distinct specialty. Each physiotherapist has a weekly treatment schedule with automatically generated sessions that don't overlap. In order to maintain a clear and manageable schedule, the system makes sure that these slots are only added when there are no conflicts with other sessions that are already scheduled.

## 2.3. Appointment Booking

## You can schedule an appointment by choosing the type of treatment or the physiotherapist's name. Following selection, the system shows every slot that meets the requirements. It makes sure that a patient doesn't have many appointments at the same time and guarantees that each slot is reserved just once. This keeps reservations from conflicting and guarantees equitable scheduling for every patient.

## 2.4. Cancellation and Attendance

## If an appointment has not yet been attended, it may be canceled. Likewise, appointments can only be recorded as attended if they haven't been canceled. The matching time slot becomes available for future reservations if an appointment is either canceled or marked as attended. The appointment ID is not used for subsequent appointments, though, and is kept reserved. This keeps accurate records and avoids misunderstandings.

## 2.5. Rescheduling Appointments

## Additionally, only appointments with BOOKED status can be rescheduled using the system. The integrity of previous appointment data is maintained because appointments that are attended or cancelled cannot be rescheduled. If an appointment is still pending, users can choose a different time slot, which increases flexibility while preserving logical consistency.

## 2.6. Generate Report

## Reports that show all of each physiotherapist's appointments—booked, canceled, and attended—are generated by the system. The report not only displays the appointment data but also rates physiotherapists from highest to lowest in terms of the number of appointments they have attended. This aids in assessing workload distribution and performance.

## 2.7. Error Handling

# There are error-handling methods in the system. It makes sure that a patient doesn't have overlapping appointments, prevents duplicate patient IDs, and guarantees that appointments cannot be modified once they have been canceled or attended. The system guides users with unambiguous error messages if they enter incorrect information while navigating menus. By combining these characteristics, a complete and useful management solution for physiotherapy clinics is produced, increasing user satisfaction, efficiency, and dependability.

# 3. Design and Implementation

## 3.1 Object-Oriented Principles

## Object-oriented design concepts are closely followed by the BPC System. All class attributes are kept private and made public using getter and setter methods in order to preserve encapsulation. In order to reduce repetition, inheritance is employed effectively, with the Person class acting as a base class for both the patient and the physiotherapist. In treatment scheduling, polymorphism is used to manage various treatment kinds under common structures such as AppointmentSchedule and TreatmentSlot, enabling flexible yet uniform handling.

## 3.2 UML Class Diagram

The links between the BPC System's fundamental classes are shown graphically in the UML class diagram that follows. The inheritance structure, relationships, and important multiplicities between items like patients, physiotherapists, treatment slots, and appointments are depicted. The following UML class diagram illustrates the connections between the basic classes of the BPC System graphically. It shows the linkages, inheritance structure, and significant multiplicities between things like patients, physiotherapists, treatment times, and appointments.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

Figure 3.2.1 UML Class Diagram

## 3.3 Class Designs

**Person (Abstract)**: This is a base class containing shared attributes like id, name, phoneNumber, and address as you see in It provides a unified structure for all types of people in the system.

**Patient**: Inherits from the Person class. This class represents individual patients and maintains a list of appointment IDs for tracking their bookings. Patients are added through the menu, and only name, address, and phone number are required. The ID is generated automatically by the system. This is achieved through iterating over the existing list of patients and collecting all used IDs in a HashSet using the following code as shown in Figure 3.3.1.

A screen shot of a computer code

AI-generated content may be incorrect.

Figure 3.3.1 HashSet Method

This ensures that no duplicate IDs are assigned. Patients are added through a simple interface where only the name, address, and phone number are needed.

**Physiotherapist**: includes further features like expertiseArea and a collection of TreatmentSlot objects that indicate their timetable, and it also derives from Person. Every physiotherapist has their own specializations and non-conflicting slots preloaded into the system.

**TreatmentSlot**: symbolizes a therapy session. Details such as the type of treatment, the time frame, the physiotherapist, and the status of the reservation are included for each slot. It offers ways to schedule or cancel a session and guarantees that no slot falls on top of another.

**AppointmentStatus** (Enum): Defines the possible statuses for an appointment. This makes appointment tracking clearer and avoids relying on string values.

**AppointmentSchedule**: This class holds details about a booked appointment, including patient, physiotherapist, time, treatment, and status. It uses an Enum AppointmentStatus to track whether an appointment is BOOKED, CANCELLED, or ATTENDED.

**AppointmentRequest**: A helper class is used temporarily during the booking process to hold a patient's treatment needs and find a matching slot. This improves modularity and keeps the booking process organized.

**MainMenuControl**: Acts as the core controller of the system. It maintains all lists such as patients, physiotherapists, treatments, and appointments. It contains the main logic for adding and removing patients, booking, canceling, attending appointments, and generating reports.

**BPC\_MainMenu**: This class uses a command-line menu to provide the user interface. For actions like adding patients, scheduling appointments, and creating reports, it presents alternatives and communicates with the user. Its real activities are handled by MainMenuControl.

## 3.3 Prevention of Logical Errors

To ensure data consistency, the system prevents double bookings by checking whether a patient already has an appointment at a given time. It also avoids reusing the same appointment ID after cancellation or attendance, maintaining a permanent record. Additionally, once a slot is booked, it cannot be rebooked unless it is cancelled or marked attended. The system avoids overlapping time slots during the creation of treatment schedules and restricts attending or cancelling appointments more than once.

## 3.5 Data Structure Use

## To manage groups of patients, physiotherapists, appointments, and slots, the system makes considerable use of list collections. Adding, deleting, and searching are made simple by this dynamic data structure. Java's LocalDateTime and related classes are used to handle dates and times, enabling accurate formatting, comparison, and conflict checking for appointments and slots. The system remains legible and efficient because to these common data types and structures.

## 3.6 Design Patterns and Principles Used

# The BPC System uses a number of design patterns and concepts to encourage clarity and maintainability: Creational Pattern: Object creation is centralized and modular, but no explicit factory class is utilized. For instance, elements of the Factory pattern are reflected in the controlled instantiation of patients and appointments based on user input. Structural Pattern: The components of the system are arranged using composition and inheritance. Physiotherapists effectively use structural design by managing lists of TreatmentSlot objects, while patients and physiotherapists extend Person. Behavioral Pattern: MainMenuControl, which manages all activities started from the interface, gives the system command-like behavior. AppointmentStatus also handles appointment state changes (such as from BOOKED to CANCELLED) by acting as a state pattern.

# 4. Testing and Validation

Using JUnit 5 in IntelliJ IDEA, unit testing was carried out to verify the BPC system's essential features. Essential functionality like patient management, appointment scheduling, cancelation, attendance, and report production were examined in a special test class called MainMenuControlTest.   
The test situations listed below were discussed:

**Add and Remove Patient**: Ensures a patient can be added and removed correctly by verifying the patient list before and after the operation.

**Book Appointment**: Simulates booking an appointment using a prepared request and verifies the system output to confirm the success. Here my test case is failed because there is no slot now.

**Attend Appointment**: Marks a booked appointment as attended and checks if the status updates correctly.

**Cancel Appointment**: Cancels a booked appointment and ensures the slot is freed and status is updated.

**Generate Report**: Simulates attended and booked appointments across different physiotherapists and confirms that the output generated matches the expected counts.

# These tests offer a starting point for verifying that the system's main workflows operate as intended. Testing guarantees that system limitations—like not rescheduling unscheduled appointments or arranging overlapping times—are respected. Coverage concentrated on the business logic within MainMenuControl, and JUnit tests were developed within the distinct package JTest.

# 5. Version Control and Project Snapshots

Git, a distributed version control system, was used to monitor the BPC System's development. Git made it easier to keep track of all updates, manage source code changes, and safely test out new features. The project was stored on a GitHub repository. Every significant modification, such adding a feature or resolving a bug, was recorded with an explanation. My GitHub URL to view the project is: https://github.com/Mohit-Rathor96/Mohit\_BPC\_System.git

In the following snapshots File structure of the project in IntelliJ IDEA and GitHub Repository are displayed.

A computer screen shot of a program

AI-generated content may be incorrect.

Figure 5.1: File structure of the project in IntelliJ IDEA

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5.2: GitHub Repository