7COM1025 Coursework

Dr Hui Cheng
School of Physics, Engineering and Computer Science
University of Hertfordshire

02/2025

- Implement a booking system for the Boost Physio Clinic (BPC)
  - Personnel
    - Physiotherapist
      - maintained by the system
      - a unique ID number, full name, address, and telephone number
      - one or more areas of expertise, such as “Physiotherapy”, “Rehabilitation”, “Osteopathy”
      - working timetable
    - Patient
      - maintained by the system
      - a unique ID number, full name, address, and telephone number
  - Treatment
    - Name,
      - such as “Neural mobilisation”, “Acupuncture”, “Massage”, “Mobilisation of the spine and joints”, “Pool rehabilitation”
    - Date/Time, e.g., Thursday 1st May 2025, 10:00-11:00
    - Physiotherapist

- Treatment appointment booking
  - Patients to look up and book treatment appointments
  - Two ways to do it:
    - One is to look up an area of expertise -> see a list of available physiotherapists and their available treatments including the time, and then book the appointment
    - The other is to directly look up a physiotherapist by their name -> find all of his/her available treatments including the time -> book the appointment
  - A patient can change their appointment (i.e., cancelling it and booking a new appointment) or simply cancel it before it takes place.

- For simplicity, design the treatment timetable for all the physiotherapists for a single term of 4 weeks (within a month).
  - For a physiotherapist, the treatment appointments offered by him/her during each week may be different. So the timetable is not repeated every week.
- At the end of the term, the clinic produces a report:
  - listing all treatment appointments for each physiotherapist, i.e., physiotherapist name, treatment name, patient name, time, and appointment status (booked, cancelled, or attended);
  - printing the name list of all the physiotherapists in descending order of the number of appointments being attended.

- In your final program, set up 3-5 physiotherapists, 10-15 patients, and 4-week timetable of the treatments offered by each physiotherapist.
- These pre-registered data can be hardcoded or read from external text files.
- The program should provide functionality to add/remove patients.
- Note:
    - you do not need any kind of external database for this program. The final program should be self-contained. The report can be printed to System.out, or output to a suitable GUI interface
    - you also do not need any kind of security protocol. No login with password

Your tasks

1. Create a class diagram, highlighting the main Java classes in the system and their associations. Include the attributes and key methods.
2. Implement the system, using Junit tests to confirm the functionality. Record all the functionalities of your final program in a video.
3. Write a short report (ideally around 8 pages, but no page limit) describing your design and implementation.

Discuss:

a) Any assumptions made about the system, in addition to the description in the coursework briefing sheet.

b) The overall structure and design of your system.

c) Any design patterns/design principles used.

d) The JUnit tests.

e) Any refactoring used during the development of the system.

In discussions, we want to see your reflections on this small project.

- Deliverables:

A. Develop your code using any version control software. Take snapshots of your project repository and commit messages. Include the snapshots in your report.

B. Submit to StudyNet:

1. Your report with the UML class diagrams and version control snapshots in a single PDF document.

2. A zipped project folder containing the source codes folder, test folder and an executable jar file for running the final program.

3. A screen recording video showing the functionalities of your final program.

No references needed for your report.

Mark breakdown

- We will test your program. Source code will be checked against the report and the video to confirm both that the work has been implemented by yourself, and its design.
    - UML Class diagram. (10 marks)
    - System functions. (40 marks)
    - Use of version control, with suitable commit messages. (5 marks)
    - Appropriate design and implementation. (30 marks)
    - Use of JUnit testing. (10 marks)
    - Report format and writing quality. (5 marks)

- Ideally, your program will provide at least five functionalities:

    1. Add/remove a patient
    2. Book a treatment appointment
    3. Change a booking (including cancel only)
    4. Attend a treatment appointment
    5. Print the report

It is optional to implement the functionalities of registering new physiotherapists / modifying each physiotherapist’s timetable.

Your program should be able to exit whenever there is no other option available.

Book a treatment appointment

- A patient can book any available appointment on the 4-week timetable of any physiotherapist.
- All the appointments are 1 to 1. If an appointment has been booked, another patient cannot book it again.
- A patient cannot book two appointments running at the same time (for testing, in your timetable, you need to set up at least two appointments running at the same time by different physiotherapists)

- The system will prompt the patient to select a way to view the timetable. There are two ways:
  - by specifying the expertise. For example, if a patient has chosen Rehabilitation, then all the available treatment appointments belonging to this expertise will be displayed.
  - by specifying the physiotherapist’s name. For example, if a patient has chosen Helen (assuming Helen is a physiotherapist in BPC), then all the treatment appointments offered by Helen will be displayed.
  - The patient will then book an appointment.
  - No repeated booking (i.e., a patient cannot book the same appointment twice)

Change/Cancel a booking

- Change a booking: cancel and release the appointment + re-booking
- Cancel a booking: cancel and release the appointment
- A booking ID is recommended to be used for each booking in your implementation.

- The system will prompt the patient to specify which booking they want to make a change (maybe through a booking ID)
    - The patient will select a new appointment. The program will do the same checking on the duplicates and time conflicts.
        - If no violation, the change is successful. Note that this is a booking change so the same booking ID will be kept.
        - If there is a violation, the change is unsuccessful. The patient needs to select another new appointment or the program returns to the main interface.
    - Cancelling a booking can be incorporated into this functionality. The booking ID of a cancelled booking will not be reused.
    - Do not forget to release the previously booked appointment if the change/cancelling is successful.

Attend an appointment

- Attend an appointment: selecting this functionality means that the appointment will be attended by the patient.
- Ideally, each booking will have a status. When it is booked (before attending), the status is "booked". When it is attended, the status is changed to "attended". When it is cancelled, the status is “cancelled”.

Print the report

- This report will list the information for each physiotherapist in the term of 4 weeks:
  - all treatment appointments, i.e., physiotherapist name, treatment name, patient name, time, and appointment status (booked, cancelled, or attended).
  - by physiotherapist: one by one
- The report also needs to print the name list of physiotherapists in descending order of the number of appointments being attended.
  - to see which physiotherapist has seen the highest number of patients

- Add/remove a new patient
    - unique ID

Other points

- All the physiotherapists are pre-registered. Some patients are pre-registered.
- For pre-registered physiotherapists and patients, you can hard code the data into your program or you can read the data from an external text file.
- UML diagram is included in your report, not a separate file.
- In the report, provide some snapshots of your commit messages to show that you have used Version Control.
- For JUnit test, since there are only 10 marks assigned to it, you can just pick up about one key method (not getter or setter methods) from each Java class and then write the test cases for them. In total, you should have 5 test cases. To save time, you don't need to write test cases for every method in every Java class.

- When closing the program, you don’t need to save any data you have created while running the program. All the bookings/changes you have made can go.
- No real time in this system: no past, no present, no future. So you don’t need to compare the time.
- An appointment, which has been booked, but not attended/cancelled, will still be shown as booked in the report. There is no missed category.

Video Recording

- Start recording from the time you start your program
- Just demonstrate all the functionalities including all the constraints checking.
  - For example, check that duplicate booking is not allowed, that time conflict booking is not allowed, that ID number must be unique, etc.
- It is not a presentation, so do NOT introduce your design and implementation.