**FlipFit - SDE2**

Design a backend system for a new enterprise application that Flipkart is launching, FlipFit.  
Flipkart is partnering up with gyms across Bangalore to enter into the fitness space. For the Beta launch the requirements are as follows:

* There are only 2 centers for now - **Koramangala** and **Bellandur**. We plan to expand to multiple others if we get traction.
* Each center has **n slots of an hour each**. For eg the Bellandur center has only 6 slots - 3 in the morning of **an hour each** from **6am to 9am** and similarly 3 in the evening from **6pm to 9pm.** All the centers are open **7 days a week.**
* Each slot at a center can have **n possible workout variations** - **Weights**, **Cardio, Yoga, Swimming etc.** There could be newer workouts added at center/slot level in the future. Adding new workout types should be easy and lead to minimal changes in application.
* For a given center, the number of seats available per workout slot is fixed. This number applies to each slot of each workout type at that center.
* For simplicity’s sake you can assume that the workout info will be entered by the Admin only once.
* User can perform the following operations:
* Register onto the platform
* View the workouts availability/unavailability for a particular day
* Book a workout if seats are available in particular time slot at particular center
* View their plan, based on day / workout type / center as input. It should be easy to change the criteria for viewing.

***Bonus****:* return the nearest available time slot (can be before or after chosen slot) for a given workout/center/user combination, keeping in mind the user’s other booked slots. This behaviour should trigger if the user’s chosen slot is not available.

**Guidelines**:

* Time: 90mins
* Write modular and clean code.
* **A driver program/main class/test case is needed to test out the code by the evaluator with multiple test cases**. But do not spend too much time in the input parsing. Keep it as simple as possible.
* **Evaluation criteria: Demoable & functionally correct code, Code readability, Proper Entity modelling, Modularity & Extensibility, Separation of concerns, Abstractions. Use design patterns wherever applicable**
* You are not allowed to use any external databases like MySQL. Use only in memory data structures.
* No need to create any UX or command prompt.
* Please focus on the Bonus Feature only after ensuring the required features are complete and demoable.
* No need to exactly follow the output format shown in examples. Any way of output is fine as long as all the information is conveyed.

**Example:**

Center: Koramangala

Num of slots per day: 2

Num of seats per workout: 5

Slots:

* Hour 7  
  Workout types: Cardio, Weights
* Hour 10  
  Workout Types: Weights

Center Bellandur

Num of slots per day: 1

Num of seats per workout: 1

Slots:

* Hour 23:

Workout Types: Swimming

→ Register User1

→ Register User2

→ Register User3

→ Register User4

→ User1 view available workout slots for Day3

Day3, Hour7, Koramangala, Cardio, 5 seats

Day3, Hour7, Koramangala, Weights, 5 seats

Day3, Hour10, Koramangala, Weights, 5 seats

Day3, Hour23, Bellandur, Swimming, 1 seat

→ User1 books “Day3, Hour23, Bellandur, Swimming”

Successfully booked

→ User1 views their workout plan for Day3

Day3, Hour23, Bellandur, Swimming

→ User2 books “Day3, Hour23, Bellandur, Swimming”

Failed - Slot full

→ User2 views available workout slots for Day3

Day3, Hour7, Koramangala, Cardio, 5 seats

Day3, Hour7, Koramangala, Weights, 5 seats

Day3, Hour10, Koramangala, Weights, 5 seats

→ User2 tries to book 1 slot for the following

Day1, Hour7, Koramangala, Weights

Day2, Hour23, Bellandur, Swimming

Day3, Hour7, Koramangala, Weights

Day4, Hour23, Bellandur, Swimming

Day7, Hour7, Koramangala, Weights

(All bookings will succeed)

→ User2 books “Day1, Hour7, Koramangala, Cardio”

Failed - User2 is already booked in Day1 Hour7

→ User2 views their workout plan for Kormangala Center

Day1, Hour7, Koramangala, Weights

Day3, Hour7, Koramangala, Weights

Day7, Hour7 Koramangala, Weights

*(Not all variations of required functionalities are mentioned in this test case. Please go through the requirement description carefully.)*

**(Bonus)**

→ User3 books “Day7, Hour7, Koramangala, Weights” **(for the example, assume this slot is already full)**

Nearest open slot: “Day7, Hour10, Koramangala, Weights”, Successfully booked