BOOKATURF

KASHSHAF LABIB

ID: 210042151

Contents

- *Overview*
- *Scope of the Project*
- *Key Functionalities*
- *UML Class Diagram*
- *SOLID Application on BookATurf*
- *Avoided Code Smells*

Overview

- To cater to the increasing demand of the of the turf grounds across the country, this project provides a comprehensive solution for managing turf bookings, ensuring a smooth and efficient experience for customers. However, for the sake of simplicity, the project is designed in the console to demonstrate the basic functionalities of a Turf booking app which can help designing an actual web app in future.
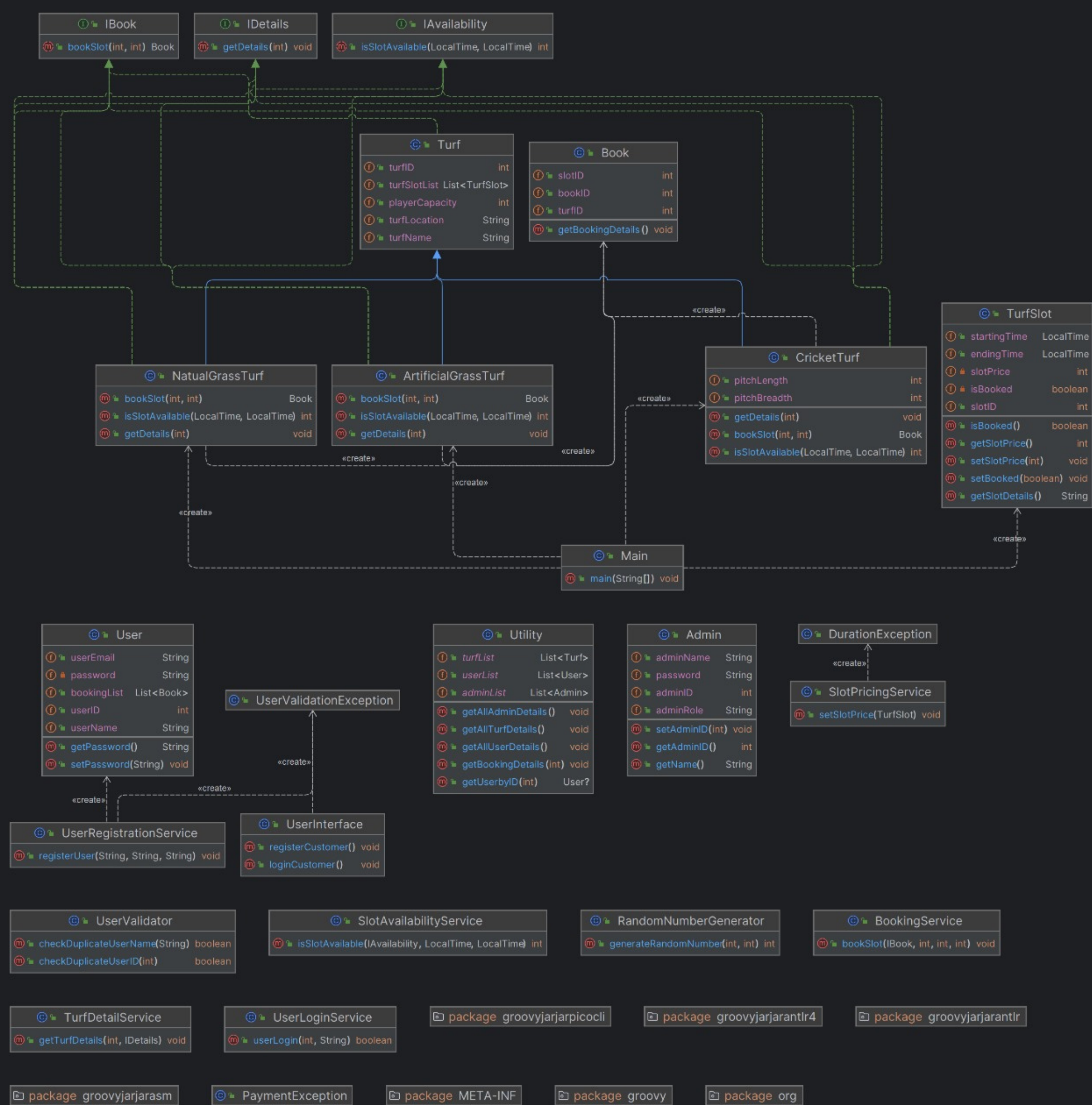
Scope of BookATurf

- The scope of the "BookATurf Management System" revolves around facilitating the booking and management of different types of turfs, such as Artificial Grass Turfs, Natural Grass Turfs, and Cricket Turfs. The system allows users to register, log in, explore available turfs, book slots, and view booking details.

Key Functionalities

- User Registration and Login

- Turf Exploration

- Slot Booking

- Slot Availability Checking

- Exception Handling

Interfaces

IBook
- bookSlot(int, int) Book

IDetails
- getDetails(int) void

IAvailability
- isSlotAvailable(LocalTime, LocalTime) int

Classes

Turf
- turfID : int
- turfSlotList : List<TurfSlot>
- playerCapacity : int
- turfLocation : String
- turfName : String

Book
- slotID : int
- bookID : int
- turfID : int
- getBookingDetails() void

TurfSlot
- startingTime : LocalTime
- endingTime : LocalTime
- slotPrice : int
- isBooked : boolean
- slotID : int
- isBooked() boolean
- getSlotPrice() int
- setSlotPrice(int) void
- setBooked(boolean) void
- getSlotDetails() String

NatualGrassTurf
- bookSlot(int, int) Book
- isSlotAvailable(LocalTime, LocalTime) int
- getDetails(int) void

ArtificialGrassTurf
- bookSlot(int, int) Book
- isSlotAvailable(LocalTime, LocalTime) int
- getDetails(int) void

CricketTurf
- pitchLength : int
- pitchBreadth : int
- getDetails(int) void
- bookSlot(int, int) Book
- isSlotAvailable(LocalTime, LocalTime) int

Main
- main(String[]) void

User
- userEmail : String
- password : String
- bookingList : List<Book>
- userID : int
- userName : String
- getPassword() String
- setPassword(String) void

UserValidationException

Utility
- turfList : List<Turf>
- userList : List<User>
- adminList : List<Admin>
- getAllAdminDetails() void
- getAllTurfDetails() void
- getAllUserDetails() void
- getBookingDetails(int) void
- getUserbyID(int) User?

Admin
- adminName : String
- password : String
- adminID : int
- adminRole : String
- setAdminID(int) void
- getAdminID() int
- getName() String

DurationException

SlotPricingService
- setSlotPrice(TurfSlot) void

UserRegistrationService
- registerUser(String, String, String) void

UserInterface
- registerCustomer() void
- loginCustomer() void

UserValidator
- checkDuplicateUserName(String) boolean
- checkDuplicateUserID(int) boolean

SlotAvailabilityService
- isSlotAvailable(IAvailability, LocalTime, LocalTime) int

RandomNumberGenerator
- generateRandomNumber(int, int) int

BookingService
- bookSlot(IBook, int, int, int) void

TurfDetailService
- getTurfDetails(int, IDetails) void

UserLoginService
- userLogin(int, String) boolean

package groovyjarjarpicocli

package groovyjarjarantlr4

package groovyjarjarantlr

UserValidationException

package groovyjarjarasm

PaymentException

package META-INF

package groovy

package org

SRP in BookATurf

- SlotAvailabilityService: This class solely handles the availability of turf slots for the users.
- BookingService: This class has method that corresponds to only booking of a turfslot for a user.
- SlotPricingService: This class has method that sets the price of a turfslot according to the duration of a booking.
- SlotDetailService: This class is responsible to show the details to the users who want to book turfslots.
- UserRegistrationService: Dedicated class for the user registration service which ensures the uniqueness of the username and the userID.
- UserLoginService: This class solely handles the login of a user by checking the existence of the user in the system.

SRP in BookATurf

- SlotAvailabilityService: This class solely handles the availability of turf slots for the users.

- UserValidator: BookATurf ensures unique username and userID for the users who register to the system and this class handles the requirements for the task.

- RandomNumberGenerator: This class provides random numbers to generate userID for the customers who register to the system.

OCP in BookATurf

- In the project, the Turf class is open for extension (by creating subclasses like ArtificialGrassTurf, NaturalGrassTurf, and CricketTurf), and closed for modification. We can create any kind of turf of a different type and have the code running bug free.

LSP in BookATurf

- This principle states that subtypes must be substitutable by its parent type and from the UML class diagram, it is clear that the subtypes like ArtificialGrassTurf, NaturalGrassTurf and CricketTurf do not violate the rule.

ISP in BookATurf

- 'IAvailability' interface: This interface imposes a contract to check the availability of a turf slot in a certain time.

- 'IDetails' interface: This interface imposes a contract to provide the details of turfs and their slots to the users.

- 'IBook' interface: This interface provides methods of booking to be implemented by a class.

DIP in BookATurf

- The 'ArtificialGrassTurf', 'NaturalGrassTurf' and 'CricketTurf' classes depend on abstractions (interfaces) like 'IAvailability', 'IBook', 'IDetails' and do not depend on concrete implementation.

Avoided Code Smells

- Inappropriate Name

- Duplicate Code

- Conditional Complexity

- Large Class

- Feature Envy

- Lazy Class

- Refused Bequest

- Shotgun Surgery

THANK YOU

For Your Kind Patience