Parking Reservations

In this exercise you will build a simple system to handle customers booking all-day parking at a single carpark.

Using the language of your choice, write code to support the requirements and tests to exercise them. Preferred languages are Python, TypeScript, Kotlin, Swift or Go.

Important: You must include instructions on how to run your tests in a section of your README file.

Requirements:

- A carpark consists of 4 bays numbered 1-4 inclusive.
- A customer can book all-day parking at a carpark with a given date if there is a free bay available.
- A customer can only make one booking a day
- Bookings must be made at least 24h in advance of the booking date
- A carpark can be queried for all valid bookings on a given date
- A booking includes details of the time it was made and the customer who made it
- A customer record includes a name and license plate string for a single car

An example test:

1. Alice, Bob, Charlie and Dave book a parking bay for 1 June 2022.
2. Ed tries to book a parking bay on the same day but is unable to because there are no free days.
3. Ed books a parking bay for 2 June 2022.
4. Querying bookings for 1 June 2022 returns 4 bookings with the allocated bay and each customer's details.

We're looking for:

- Modeling of the domain
- Consideration of edge cases
- Tests that document the behaviour of the code
- Code that is simple, concise and idiomatic
- A README file describing any assumptions made, design decisions and trade-offs in the implementation.

Do not include:

- Any code not required to implement the requirements
- A command line tool. The only code that will be run or inspected are the tests.
- User interface code
- HTTP server or client code
- Persistence
- Concurrency handling