Phone Booker App
Design Spec

Overview

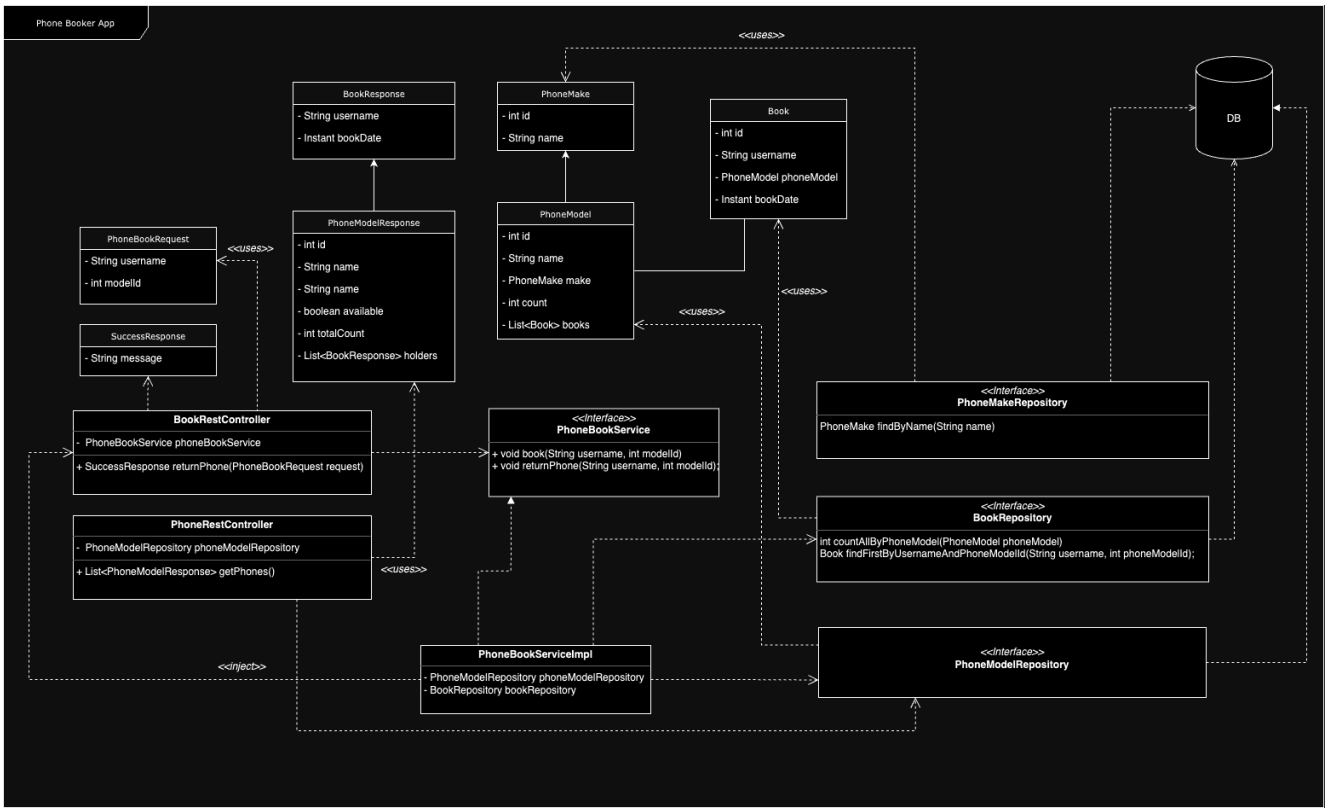
Phone booker app lets you make phone bookings and returns used for testing purposes among the developers. There are already 10 predefined mobile phones stored in the database.

Components

System uses N-tier layer to manage booking:

- **Repository**: to manage bookings and retrieve mobile phones
- **Service**: to make booking and return using BookingRepository and PhoneModelRepository
- **Controller**: REST API entry points used by clients

The application is based on Spring Framework, so all the components are based on framework components (CrudRepository, Dependency Injection)



UML diagram of the components

Repository

Repository is used to retrieve and store from/to the database.
There are 3 repositories:

- **BookRepository**: used to manage bookings
- **PhoneMakeRepository**: used to manage phone makes
- **PhoneModelRepository**: used to manage phone models

These are all interfaces and all of them extend from the CrudRepository interface provided by Spring to manage data using the JPA standard. Underlying implementation is purely provided by Spring JPA. The related models are:

- **Book**
- **PhoneMake**
- **PhoneModel**

Service

There is only **PhoneBookService** which is implemented by **PhoneBookServiceImpl**. It provides 2 methods:

- **book()** - makes booking using username and phone model id. The method logic is **synchronized** to be able to make 1 booking at a time.
- **returnPhone()** - this is an idempotent method that returns the phone using username and phone model id.

Controller

There 2 controllers used by clients to make REST API:

- **BookRestController** - used to make booking and returns.
- **PhoneRestController** - It provides only a single method for listing phones. Each phone contains details as well as availability status and current holders if there are any.

REST API spec

Endpoint	Description	Example
GET /api/phones	List the phone models	curl -XGET http://localhost:8080/api/phones
POST /api/book	Book the phone by username and modelID	curl -XPOST http://localhost:8080/api/book -H "Content-Type: application/json" -d '{"username": "test", "modelId": 1}'
POST /api/book/return	Return the phone by username and modelID	curl -XPOST http://localhost:8080/api/book/return -H "Content-Type: application/json" -d '{"username": "test", "modelId": 1}'