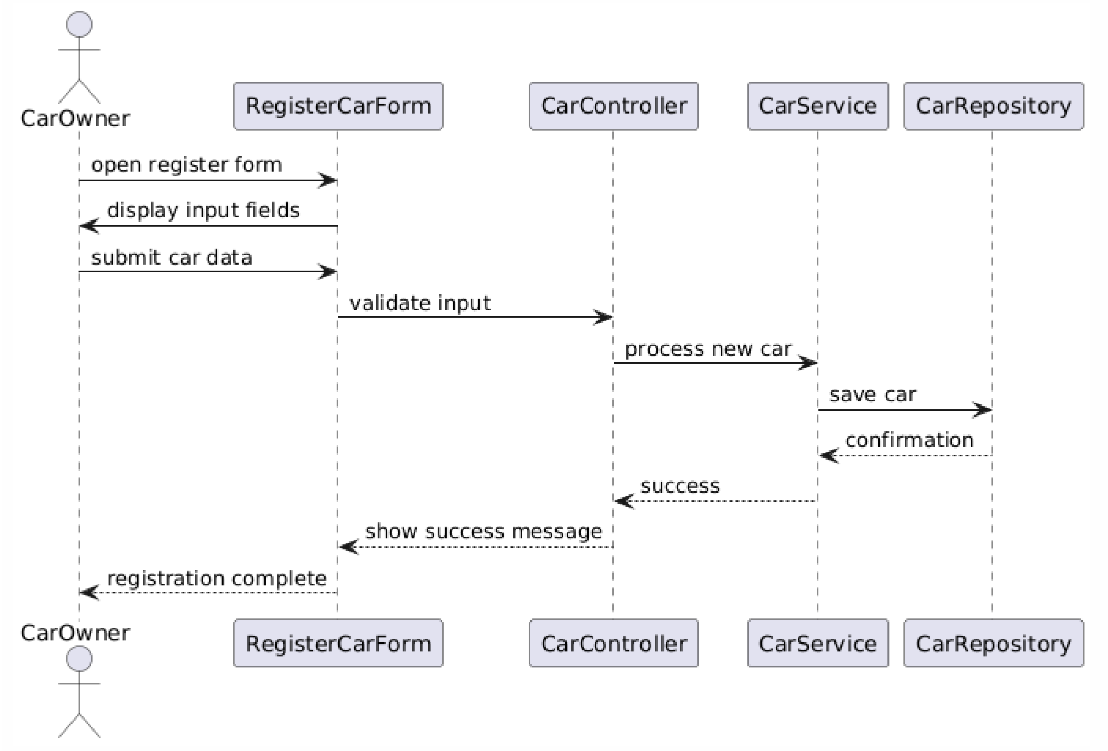
# Register Car

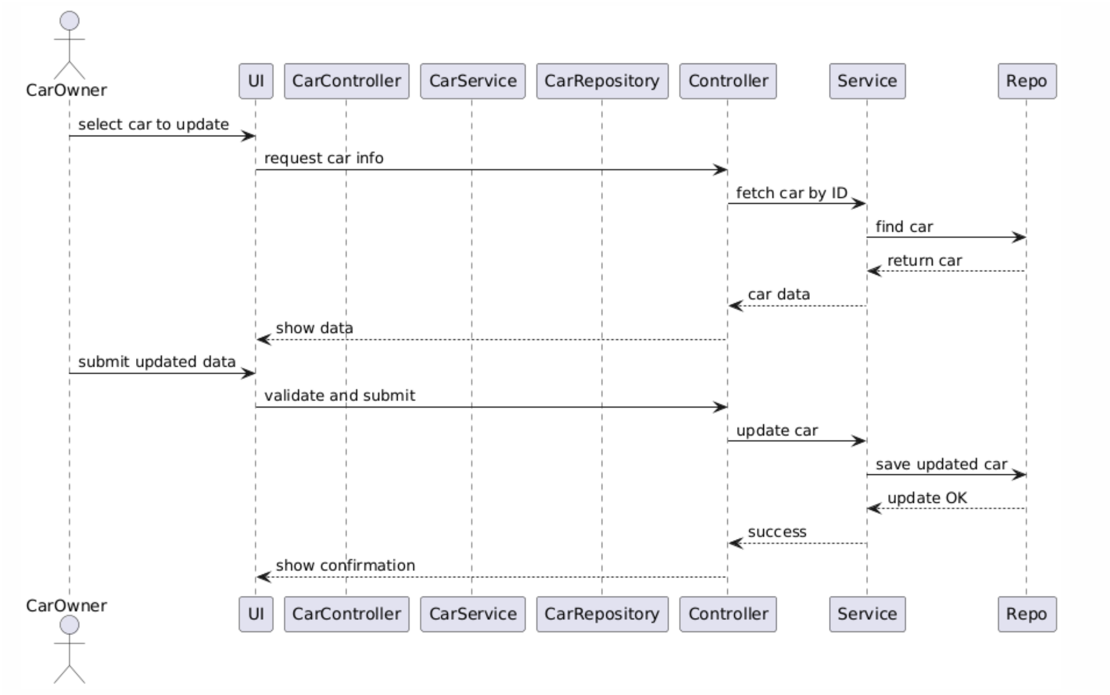
```
@startuml
actor CarOwner
participant "RegisterCarForm" as UI
participant "CarController" as Controller
participant "CarService" as Service
participant "CarRepository" as Repo

CarOwner -> UI : open register form
UI -> CarOwner : display input fields
CarOwner -> UI : submit car data
UI -> Controller : validate input
Controller -> Service : process new car
Service -> Repo : save car
Repo --> Service : confirmation
Service --> Controller : success
Controller --> UI : show success message
UI --> CarOwner : registration complete

@enduml
```



# Update Car Info

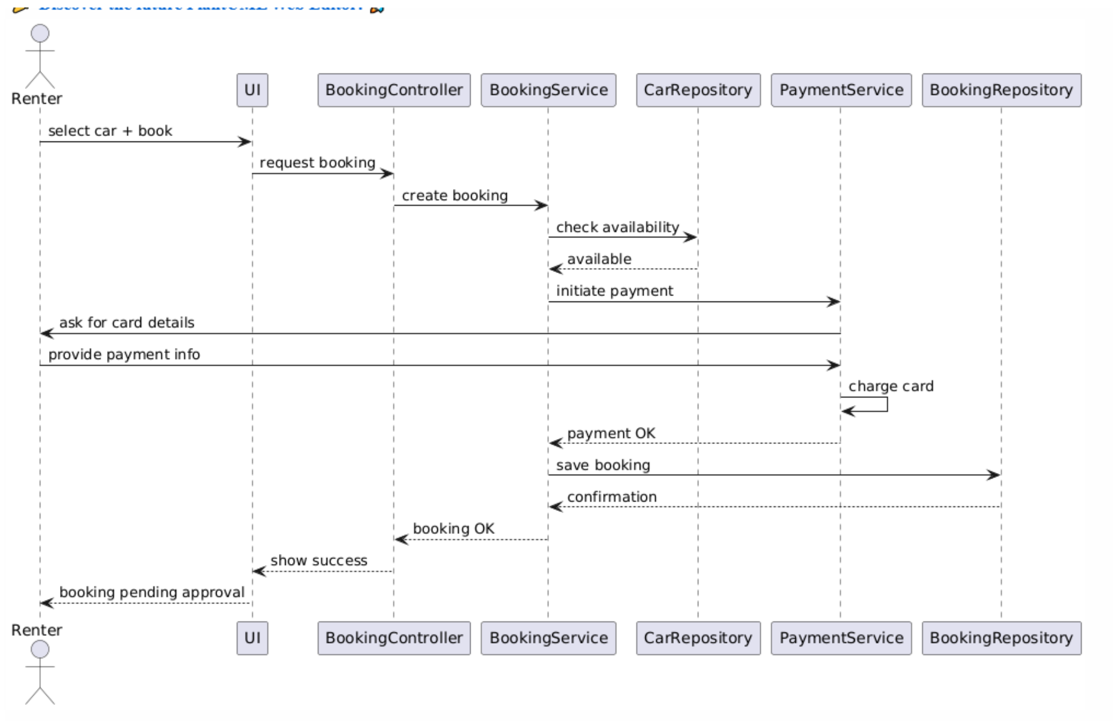
```
@startuml
actor CarOwner
participant UI
participant CarController
participant CarService
participant CarRepository

CarOwner -> UI : select car to update
UI -> Controller : request car info
Controller -> Service : fetch car by ID
Service -> Repo : find car
Repo --> Service : return car
Service --> Controller : car data
Controller --> UI : show data
```

```
CarOwner -> UI : submit updated data

UI -> Controller : validate and submit

Controller -> Service : update car

Service -> Repo : save updated car

Repo --> Service : update OK

Service --> Controller : success

Controller --> UI : show confirmation


@enduml
```



Book Car + Pay for Car

```
@startuml

actor Renter

participant UI

participant BookingController

participant BookingService
participant CarRepository
participant PaymentService
participant BookingRepository

Renter -> UI : select car + book
UI -> BookingController : request booking
BookingController -> BookingService : create booking
BookingService -> CarRepository : check availability
CarRepository --> BookingService : available

BookingService -> PaymentService : initiate payment
PaymentService -> Renter : ask for card details
Renter -> PaymentService : provide payment info
PaymentService -> PaymentService : charge card
PaymentService --> BookingService : payment OK

BookingService -> BookingRepository : save booking
BookingRepository --> BookingService : confirmation
BookingService --> BookingController : booking OK
BookingController --> UI : show success
UI --> Renter : booking pending approval

@enduml
```



# Approve Booking

@startuml

actor CarOwner

participant UI

participant BookingController

participant BookingService

participant BookingRepository

participant NotificationService

CarOwner -> UI : view pending bookings

UI -> Controller : get requests

Controller -> Service : fetch pending bookings

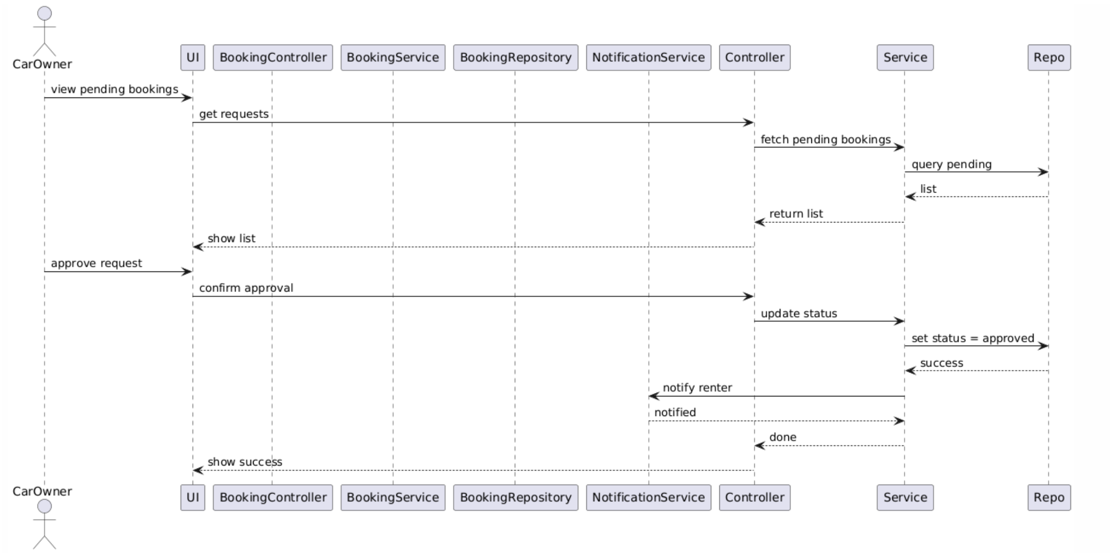
Service -> Repo : query pending

Repo --> Service : list

Service --> Controller : return list

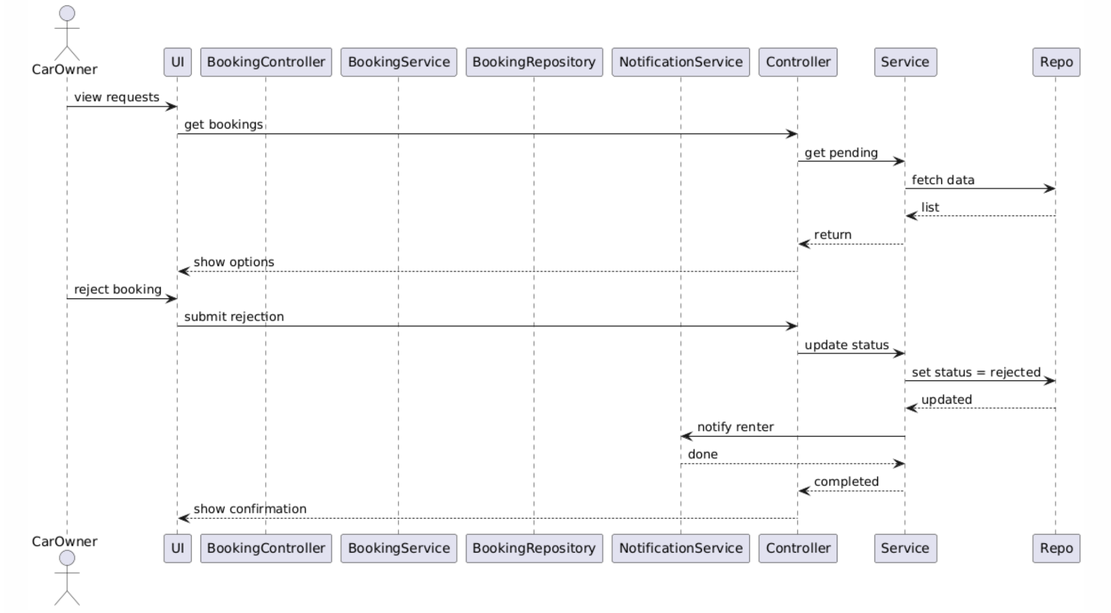
Controller --> UI : show list

```
CarOwner -> UI : approve request

UI -> Controller : confirm approval

Controller -> Service : update status

Service -> Repo : set status = approved

Repo --> Service : success

Service -> NotificationService : notify renter

NotificationService --> Service : notified

Service --> Controller : done

Controller --> UI : show success


@enduml
```



Reject Booking

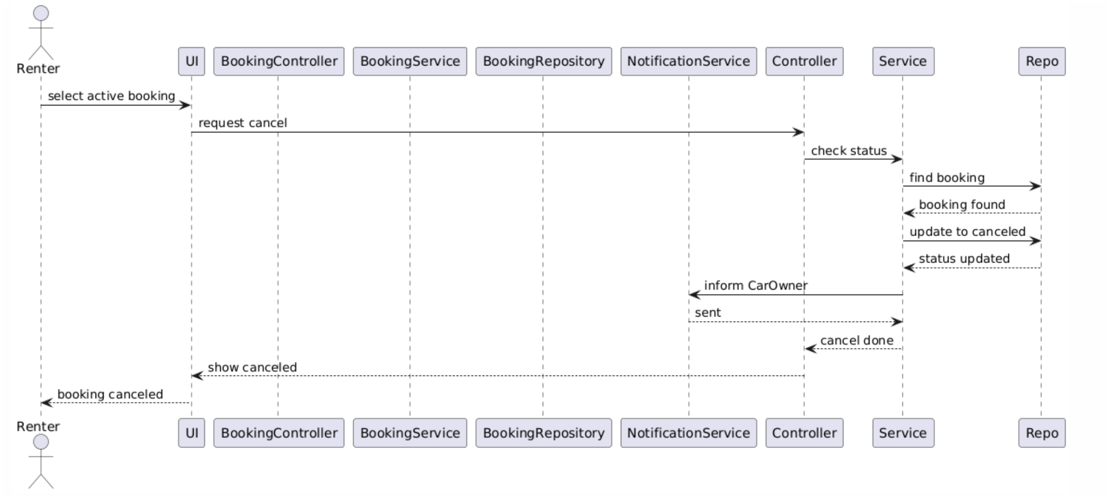
```
@startuml

actor CarOwner

participant UI

participant BookingController

participant BookingService

participant BookingRepository
participant NotificationService

CarOwner -> UI : view requests

UI -> Controller : get bookings

Controller -> Service : get pending

Service -> Repo : fetch data

Repo --> Service : list

Service --> Controller : return

Controller --> UI : show options


CarOwner -> UI : reject booking

UI -> Controller : submit rejection

Controller -> Service : update status

Service -> Repo : set status = rejected

Repo --> Service : updated

Service -> NotificationService : notify renter

NotificationService --> Service : done

Service --> Controller : completed

Controller --> UI : show confirmation

@enduml
```



Cancel Booking

```
@startuml

actor Renter

participant UI

participant BookingController

participant BookingService

participant BookingRepository

participant NotificationService



Renter -> UI : select active booking

UI -> Controller : request cancel

Controller -> Service : check status

Service -> Repo : find booking

Repo --> Service : booking found



Service -> Repo : update to canceled

Repo --> Service : status updated

Service -> NotificationService : inform CarOwner
```

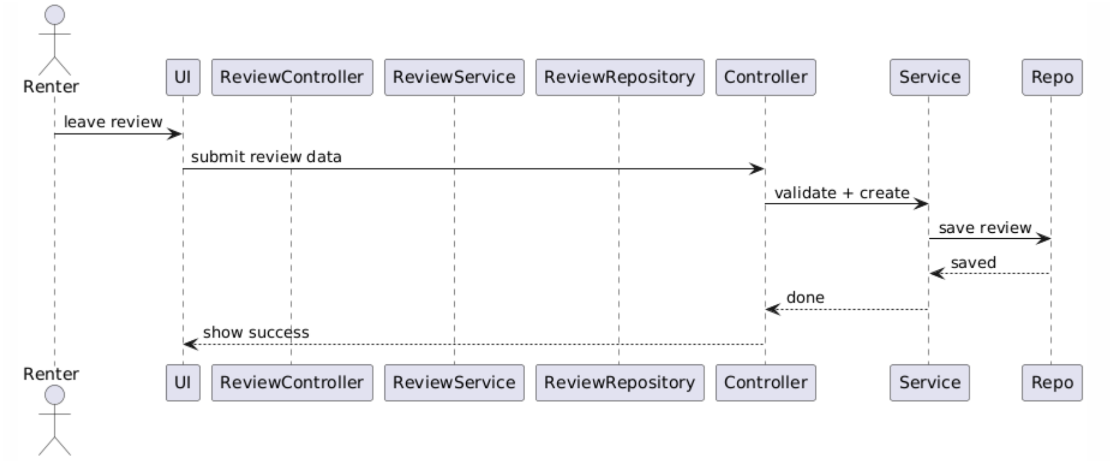
```
NotificationService --> Service : sent
Service --> Controller : cancel done
Controller --> UI : show canceled
UI --> Renter : booking canceled

@enduml
```



Leave Review

```
@startuml
actor Renter
participant UI
participant ReviewController
participant ReviewService
participant ReviewRepository

Renter -> UI : leave review
UI -> Controller : submit review data
Controller -> Service : validate + create
Service -> Repo : save review
Repo --> Service : saved
```

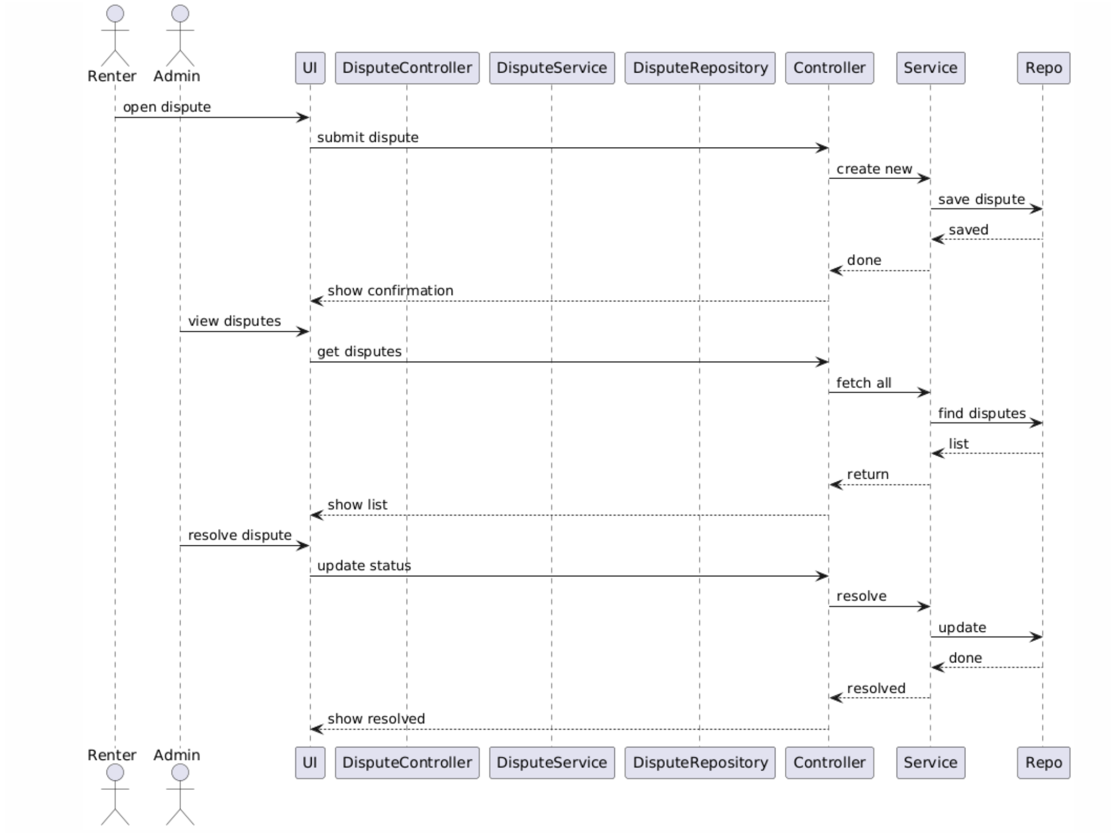
```
Service --> Controller : done

Controller --> UI : show success


@enduml
```



Resolve Dispute

```
@startuml

actor Renter

actor Admin

participant UI

participant DisputeController

participant DisputeService

participant DisputeRepository


Renter -> UI : open dispute

UI -> Controller : submit dispute

Controller -> Service : create new

Service -> Repo : save dispute

Repo --> Service : saved

Service --> Controller : done

Controller --> UI : show confirmation
```

```
Admin -> UI : view disputes
UI -> Controller : get disputes
Controller -> Service : fetch all
Service -> Repo : find disputes
Repo --> Service : list
Service --> Controller : return
Controller --> UI : show list

Admin -> UI : resolve dispute
UI -> Controller : update status
Controller -> Service : resolve
Service -> Repo : update
Repo --> Service : done
Service --> Controller : resolved
Controller --> UI : show resolved

@enduml
```

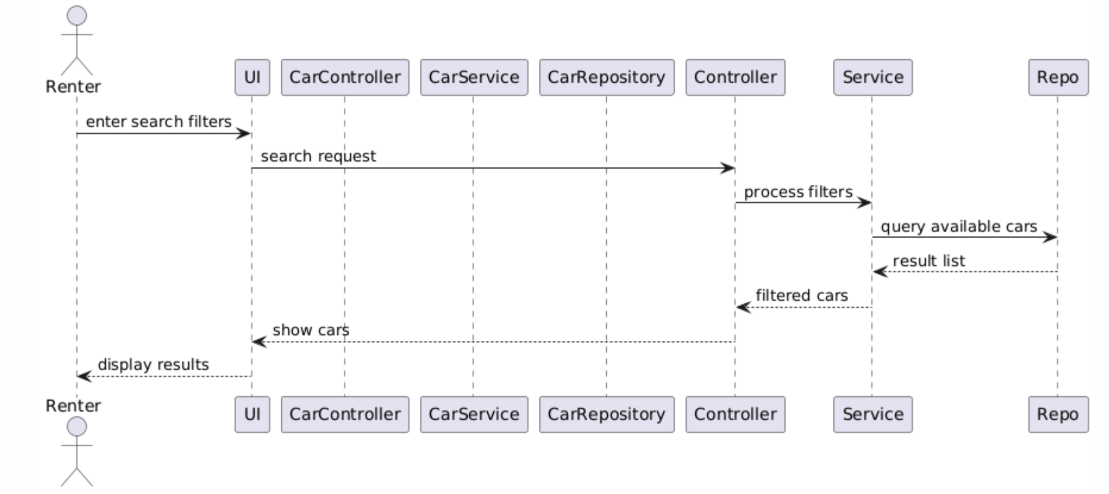


Search Car

```
@startuml
actor Renter
participant UI
participant CarController
participant CarService
participant CarRepository

Renter -> UI : enter search filters
UI -> Controller : search request
Controller -> Service : process filters
Service -> Repo : query available cars
Repo --> Service : result list
Service --> Controller : filtered cars
Controller --> UI : show cars
```

UI --> Renter : display results

@enduml



# Manage Users & Cars

@startuml

actor Admin

participant UI

participant AdminController

participant UserService

participant CarService

participant UserRepository

participant CarRepository

Admin -> UI : open management panel

UI -> Controller : get all users + cars

Controller -> UserService : get users

UserService -> UserRepository : find all

UserRepository --> UserService : user list

Controller -> CarService : get cars

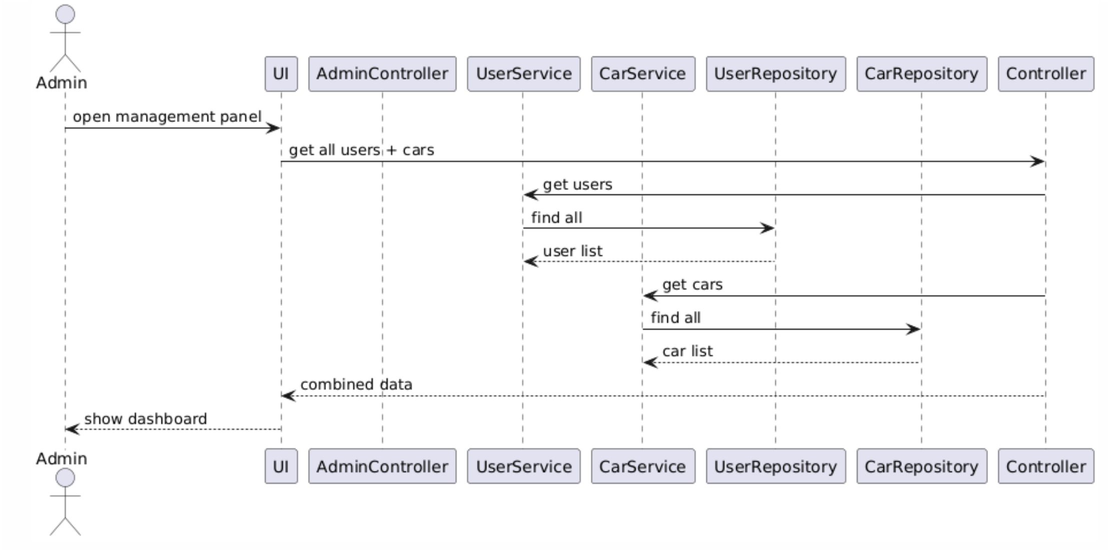
CarService -> CarRepository : find all

CarRepository --> CarService : car list

Controller --> UI : combined data

UI --> Admin : show dashboard

@enduml



# View Reports

@startuml

actor Admin

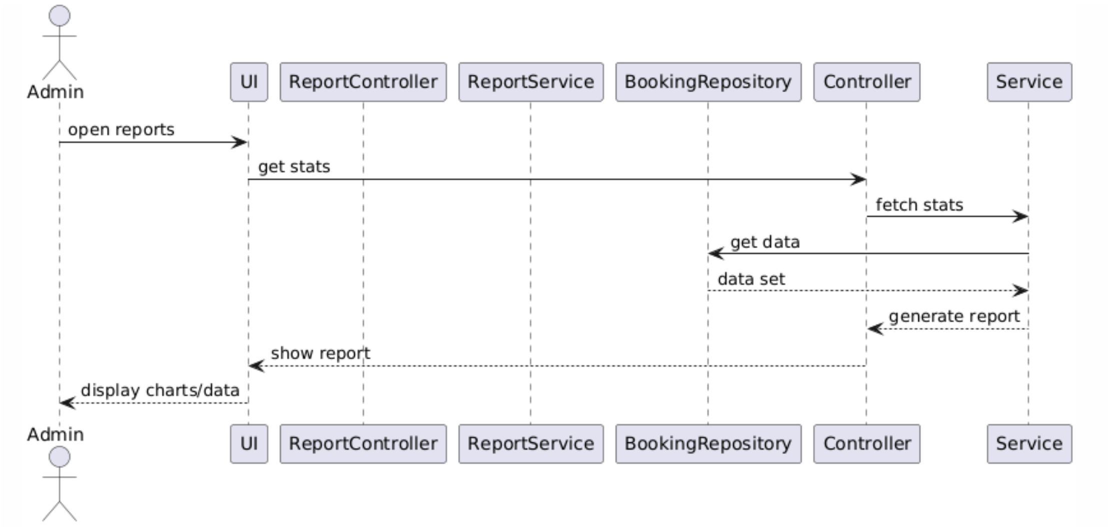
participant UI

participant ReportController

participant ReportService

participant BookingRepository

```
Admin -> UI : open reports

UI -> Controller : get stats

Controller -> Service : fetch stats

Service -> BookingRepository : get data

BookingRepository --> Service : data set

Service --> Controller : generate report

Controller --> UI : show report

UI --> Admin : display charts/data


@enduml
```


Admin
UI
ReportController
ReportService
BookingRepository
Controller
Service
open reports
get stats
fetch stats
get data
data set
generate report
show report
display charts/data
Admin
UI
ReportController
ReportService
BookingRepository
Controller
Service