



🖼 커버 추가 💬 댓글 추가

Sequence Diagram

1. REQ-001: 메인 페이지 네비게이션

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

User->>Frontend: 메인 페이지 접속 요청 (/)

Frontend->>Backend: GET /navigation

Backend-->>Frontend: 네비게이션 데이터 반환 (로그인, 대여 페이지 링크 포함)

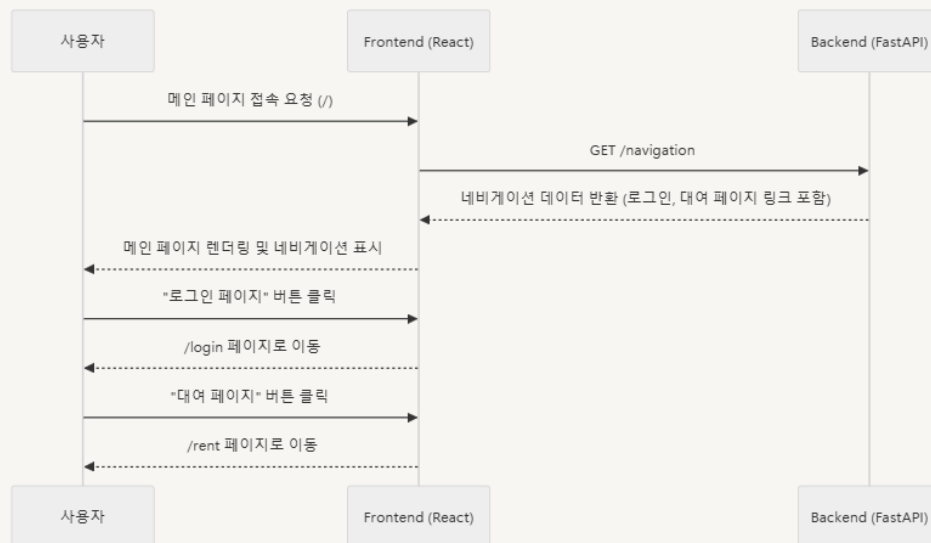
Frontend-->>User: 메인 페이지 렌더링 및 네비게이션 표시

User->>Frontend: "로그인 페이지" 버튼 클릭

Frontend-->>User: /login 페이지로 이동

User->>Frontend: "대여 페이지" 버튼 클릭

Frontend-->>User: /rent 페이지로 이동



설명:

1. 메인 페이지 접속:

- 사용자가 메인 페이지에 접속하면 프론트엔드는 백엔드에서 네비게이션 데이터를 요청하고, 받은 데이터를 기반으로 네비게이션 바와 메인 페이지를 렌더링합니다.

2. 네비게이션 버튼 동작:

- 사용자가 "로그인 페이지" 버튼을 클릭하면 `/login` 페이지로 이동합니다.
- 사용자가 "대여 페이지" 버튼을 클릭하면 `/rent` 페이지로 이동합니다.

2. REQ-002: 사용자 로그인

sequenceDiagram

participant User as 사용자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (User Data)

User->>Frontend: 로그인 정보 입력 (ID, PW) 후 "로그인" 버튼 클릭
Frontend->>Backend: POST /user/login (ID, PW)
Backend->>DB: 사용자 ID와 비밀번호 조회
DB-->>Backend: 사용자 인증 결과 반환 (성공 또는 실패)

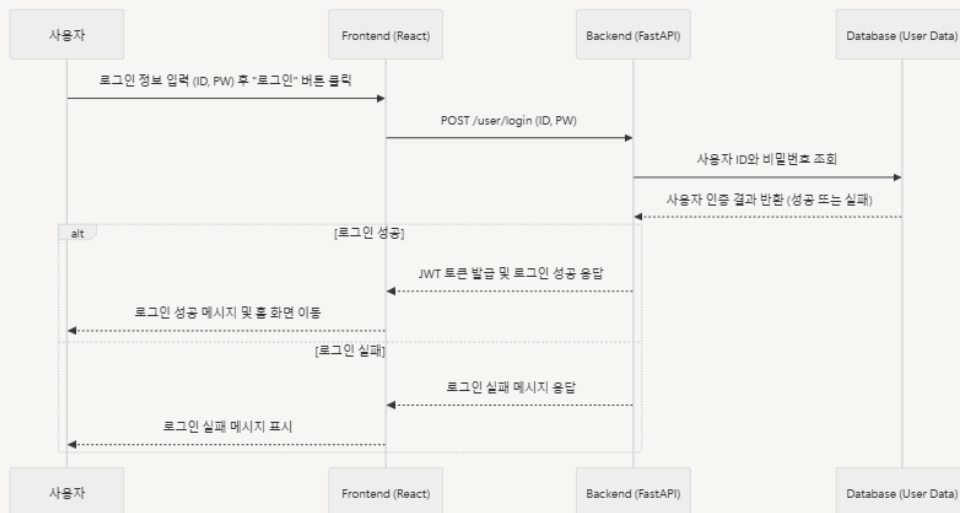
alt 로그인 성공

Backend-->>Frontend: JWT 토큰 발급 및 로그인 성공 응답
Frontend-->>User: 로그인 성공 메시지 및 홈 화면 이동

else 로그인 실패

Backend-->>Frontend: 로그인 실패 메시지 응답
Frontend-->>User: 로그인 실패 메시지 표시

end



설명:

1. 로그인 요청:

- 사용자가 로그인 정보를 입력하고 "로그인" 버튼을 클릭하면, 프론트엔드는 백엔드에 ID와 비밀번호를 전달합니다.

2. 사용자 인증:

- 백엔드는 데이터베이스에서 ID와 비밀번호를 조회하여 사용자 인증을 수행합니다.

3. 응답 처리:

- 로그인 성공 시 JWT 토큰이 발급되며, 프론트엔드로 성공 메시지가 반환됩니다. 사용자는 홈 화면으로 이동합니다.
- 로그인 실패 시 실패 메시지가 반환되며, 사용자는 실패 알림을 확인합니다.

3. REQ-003: 사용자 회원가입

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

participant DB as Database (User Data)

User->>Frontend: 회원가입 정보 입력 후 "회원가입" 버튼 클릭

Frontend->>Backend: POST /user/register (ID, PW, 기타 정보)

Backend->>DB: 사용자 정보 중복 확인 (ID, 이메일 등)

DB-->>Backend: 중복 여부 결과 반환

alt 중복 없음

Backend->>DB: 사용자 정보 저장 (비밀번호 bcrypt 암호화)

DB-->>Backend: 저장 완료 확인

Backend-->>Frontend: 회원가입 성공 응답

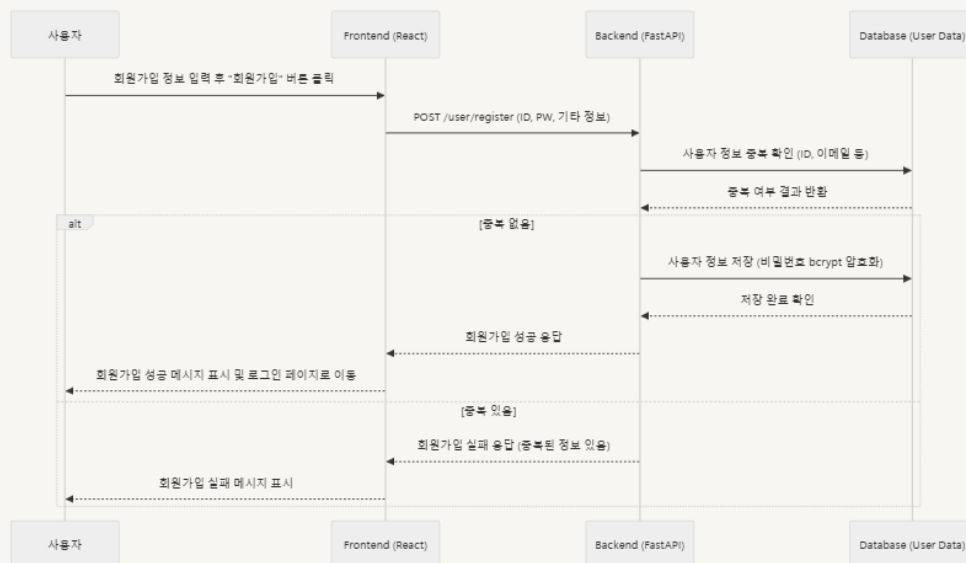
Frontend-->>User: 회원가입 성공 메시지 표시 및 로그인 페이지로 이동

else 중복 있음

Backend-->>Frontend: 회원가입 실패 응답 (중복된 정보 있음)

Frontend-->>User: 회원가입 실패 메시지 표시

end



설명:

1. 로그인 요청:

- 사용자가 로그인 정보를 입력하고 "로그인" 버튼을 클릭하면, 프론트엔드는 백엔드에 ID와 비밀번호를 전달합니다.

2. 사용자 인증:

- 백엔드는 데이터베이스에서 ID와 비밀번호를 조회하여 사용자 인증을 수행합니다.

3. 응답 처리:

- 로그인 성공 시 JWT 토큰이 발급되며, 프론트엔드로 성공 메시지가 반환됩니다. 사용자는 홈 화면으로 이동합니다.
- 로그인 실패 시 실패 메시지가 반환되며, 사용자는 실패 알림을 확인합니다.

4. REQ-004: 대여 페이지 접속

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

participant DB as Database (Module Data)

User->>Frontend: 대여 페이지 접속 요청 (/rent)

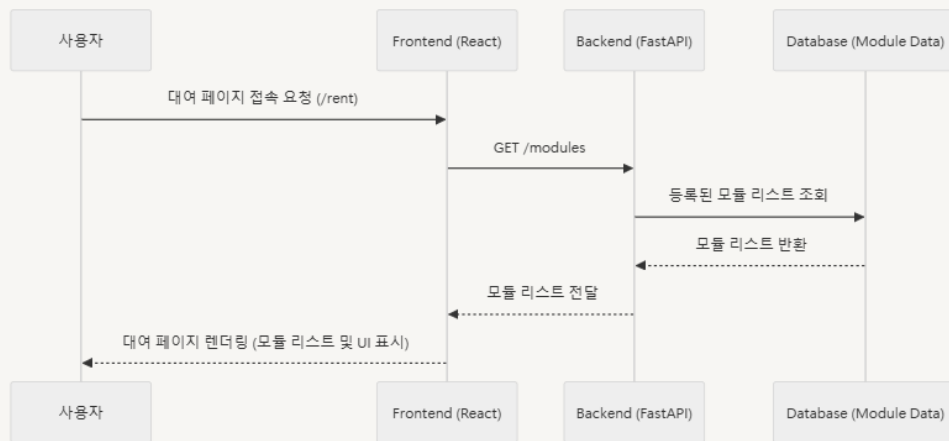
Frontend->>Backend: GET /modules

Backend->>DB: 등록된 모듈 리스트 조회

DB-->>Backend: 모듈 리스트 반환

Backend-->>Frontend: 모듈 리스트 전달

Frontend-->>User: 대여 페이지 렌더링 (모듈 리스트 및 UI 표시)



설명:

1. 대여 페이지 요청:

- 사용자가 대여 페이지에 접속하면, 프론트엔드는 백엔드에 모듈 데이터를 요청합니다.

2. 모듈 데이터 조회:

- 백엔드는 데이터베이스에서 등록된 모듈 리스트를 조회하고, 이를 프론트엔드로 전달합니다.

3. 대여 페이지 렌더링:

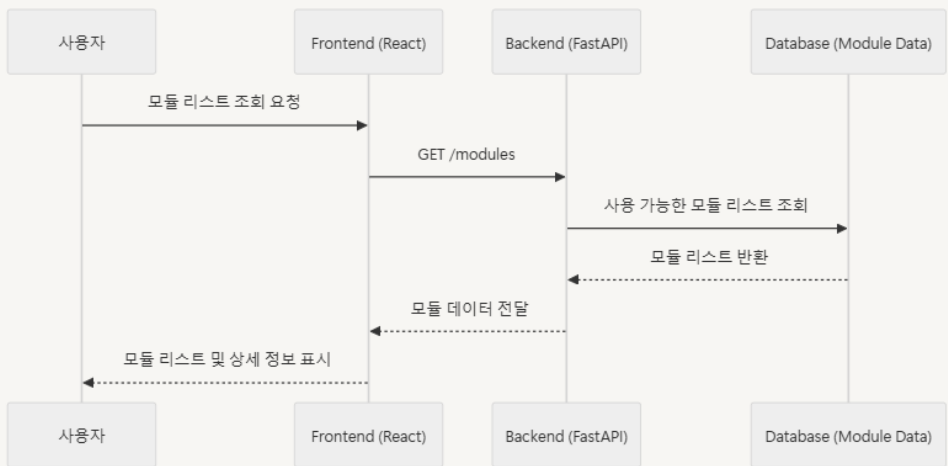
- 프론트엔드는 반환된 모듈 리스트를 기반으로 대여 페이지를 렌더링하여 사용자에게 표시합니다.

5. REQ-005: 모듈 목록 조회

sequenceDiagram

participant User as 사용자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Module Data)

User->>Frontend: 모듈 리스트 조회 요청
Frontend->>Backend: GET /modules
Backend->>DB: 사용 가능한 모듈 리스트 조회
DB-->>Backend: 모듈 리스트 반환
Backend-->>Frontend: 모듈 데이터 전달
Frontend-->>User: 모듈 리스트 및 상세 정보 표시



설명:

1. 모듈 리스트 요청:

- 사용자가 모듈 목록 조회 요청을 하면, 프론트엔드는 백엔드에 모듈 데이터를 요청합니다.

2. 데이터베이스 조회:

- 백엔드는 데이터베이스에서 사용 가능한 모듈 리스트와 상세 정보를 조회하여 반환합니다.

3. 모듈 리스트 표시:

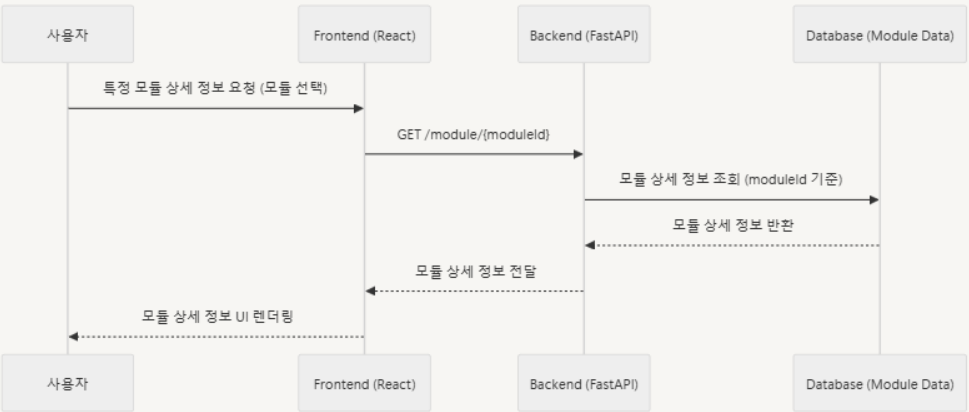
- 프론트엔드는 반환된 모듈 데이터를 기반으로 사용자에게 모듈 리스트와 상세 정보를 표시합니다.

6. REQ-006: 모듈 정보 상세 조회

sequenceDiagram

participant User as 사용자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Module Data)

User->>Frontend: 특정 모듈 상세 정보 요청 (모듈 선택)
Frontend->>Backend: GET /module/{moduleId}
Backend->>DB: 모듈 상세 정보 조회 (moduleId 기준)
DB-->>Backend: 모듈 상세 정보 반환
Backend-->>Frontend: 모듈 상세 정보 전달
Frontend-->>User: 모듈 상세 정보 UI 렌더링



설명:

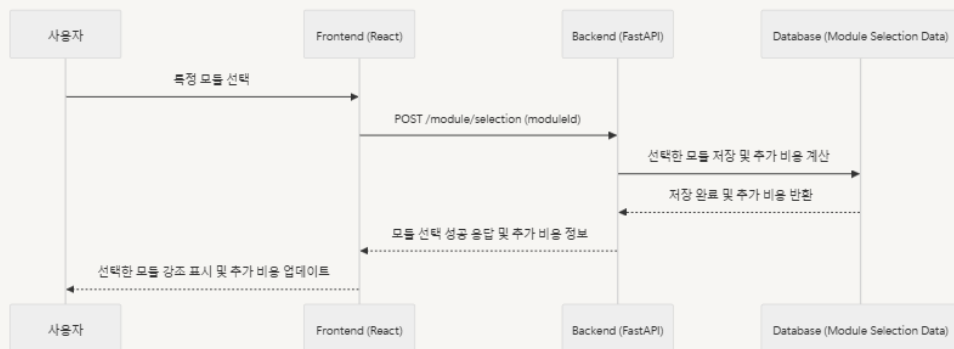
1. 모듈 상세 정보 요청:
 - 사용자가 특정 모듈을 선택하면, 프론트엔드는 해당 모듈의 상세 정보를 요청합니다.
2. 모듈 데이터 조회:
 - 백엔드는 데이터베이스에서 요청받은 모듈 ID를 기준으로 상세 정보를 조회합니다.
3. 모듈 정보 렌더링:
 - 프론트엔드는 반환된 상세 정보를 UI에 표시하여 사용자에게 보여줍니다.

7. REQ-007: 모듈 선택

sequenceDiagram

participant User as 사용자
 participant Frontend as Frontend (React)
 participant Backend as Backend (FastAPI)
 participant DB as Database (Module Selection Data)

User->>Frontend: 특정 모듈 선택
 Frontend->>Backend: POST /module/selection (moduleId)
 Backend->>DB: 선택한 모듈 저장 및 추가 비용 계산
 DB-->>Backend: 저장 완료 및 추가 비용 반환
 Backend-->>Frontend: 모듈 선택 성공 응답 및 추가 비용 정보
 Frontend-->>User: 선택한 모듈 강조 표시 및 추가 비용 업데이트



설명:

1. 모듈 선택 요청:

- 사용자가 특정 모듈을 선택하면, 프론트엔드는 선택한 모듈의 정보를 백엔드로 전달합니다.

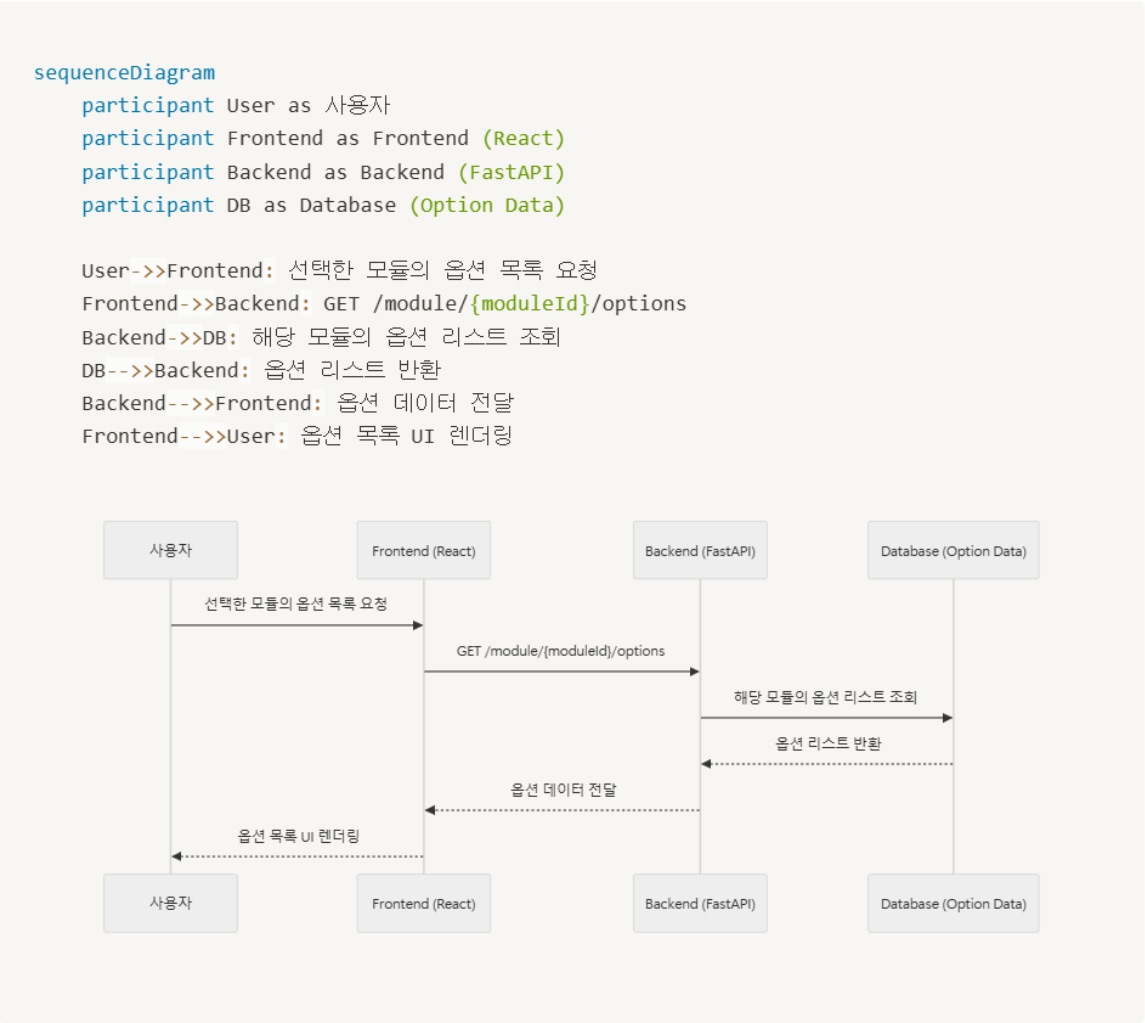
2. 선택 저장 및 비용 계산:

- 백엔드는 데이터베이스에 선택한 모듈 정보를 저장하고, 추가 비용을 계산합니다.

3. 응답 및 UI 업데이트:

- 반환된 데이터를 기반으로 프론트엔드는 UI를 업데이트하여 선택한 모듈을 강조하고 추가 비용을 사용자에게 표시합니다.

8. REQ-008: 옵션 목록 조회



설명:

1. 옵션 목록 요청:

- 사용자가 특정 모듈을 선택하면, 프론트엔드는 해당 모듈의 옵션 목록을 백엔드에 요청합니다.

2. 옵션 데이터 조회:

- 백엔드는 데이터베이스에서 선택한 모듈에 맞는 옵션 리스트를 조회합니다.

3. 옵션 목록 표시:

- 프론트엔드는 반환된 옵션 데이터를 기반으로 UI를 렌더링하여 사용자에게 옵션 목록을 표시합니다.

9. REQ-009: 옵션 상세 조회

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

participant DB as Database (Option Data)

User->>Frontend: 특정 옵션 상세 정보 요청 (옵션 선택)

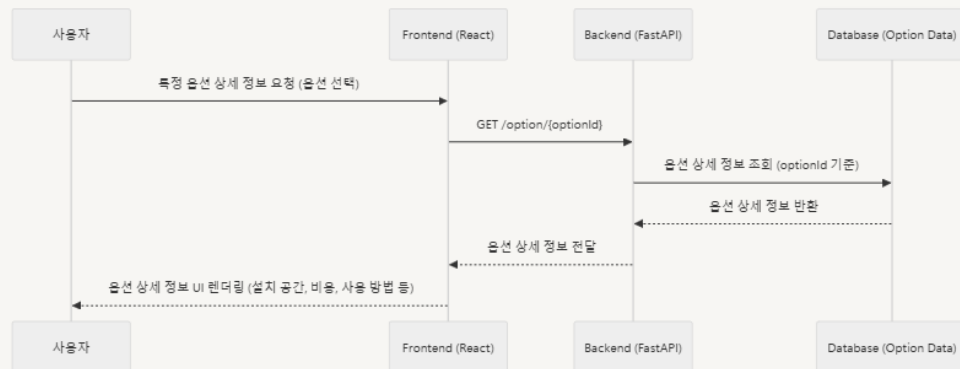
Frontend->>Backend: GET /option/{optionId}

Backend->>DB: 옵션 상세 정보 조회 (optionId 기준)

DB-->>Backend: 옵션 상세 정보 반환

Backend-->>Frontend: 옵션 상세 정보 전달

Frontend-->>User: 옵션 상세 정보 UI 렌더링 (설치 공간, 비용, 사용 방법 등)



설명:

1. 옵션 상세 정보 요청:

- 사용자가 특정 옵션을 선택하면, 프론트엔드는 해당 옵션의 상세 정보를 백엔드에 요청합니다.

2. 옵션 데이터 조회:

- 백엔드는 데이터베이스에서 요청받은 옵션 ID를 기준으로 상세 정보를 조회합니다.

3. 옵션 정보 렌더링:

- 프론트엔드는 조회된 옵션 정보를 UI로 렌더링하여 사용자에게 상세 정보를 제공합니다.

10. REQ-010: 옵션 선택

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

participant DB as Database (Option Selection Data)

User->>Frontend: 특정 옵션 선택

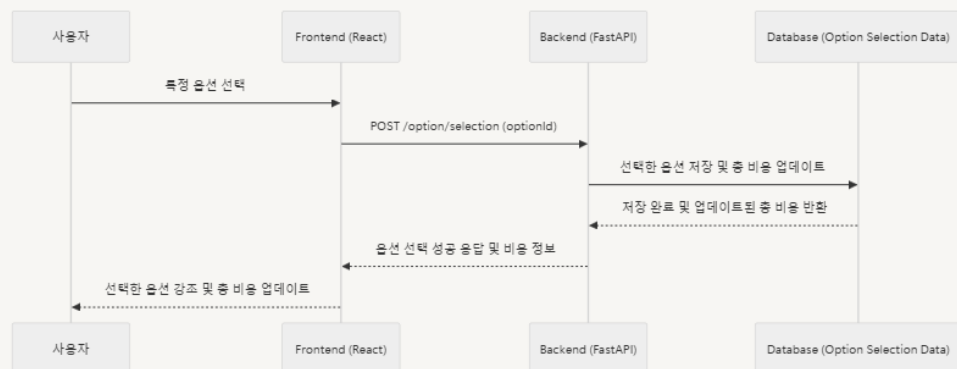
Frontend->>Backend: POST /option/selection (optionId)

Backend->>DB: 선택한 옵션 저장 및 총 비용 업데이트

DB-->>Backend: 저장 완료 및 업데이트된 총 비용 반환

Backend-->>Frontend: 옵션 선택 성공 응답 및 비용 정보

Frontend-->>User: 선택한 옵션 강조 및 총 비용 업데이트



설명:

1. 옵션 선택 요청:

- 사용자가 특정 옵션을 선택하면, 프론트엔드는 선택한 옵션의 정보를 백엔드에 전달합니다.

2. 선택 저장 및 비용 계산:

- 백엔드는 데이터베이스에 선택한 옵션을 저장하고, 총 비용을 업데이트합니다.

3. 응답 및 UI 업데이트:

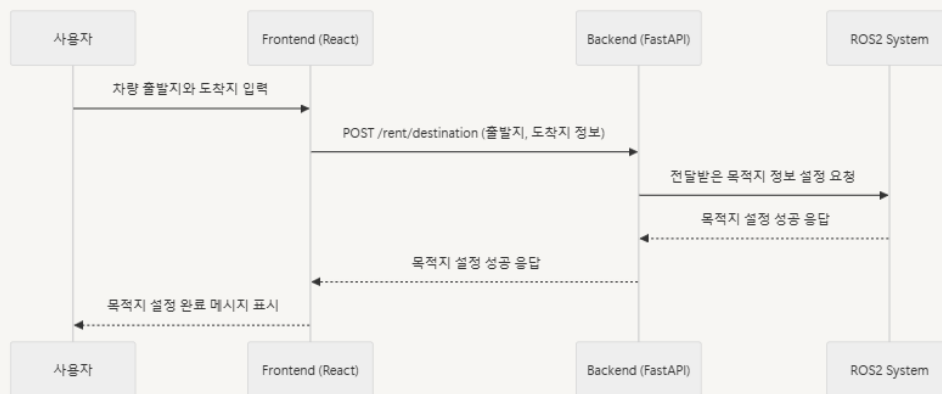
- 프론트엔드는 반환된 데이터를 기반으로 UI를 업데이트하여 선택한 옵션을 강조하고 업데이트된 총 비용을 사용자에게 표시합니다.

11. REQ-011: 목적지 설정

sequenceDiagram

participant User as 사용자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant ROS2 as ROS2 System

User->>Frontend: 차량 출발지와 도착지 입력
Frontend->>Backend: POST /rent/destination (출발지, 도착지 정보)
Backend->>ROS2: 전달받은 목적지 정보 설정 요청
ROS2-->>Backend: 목적지 설정 성공 응답
Backend-->>Frontend: 목적지 설정 성공 응답
Frontend-->>User: 목적지 설정 완료 메시지 표시



설명:

1. 목적지 입력 요청:

- 사용자가 차량의 출발지와 도착지를 입력하면, 프론트엔드는 해당 정보를 백엔드로 전달합니다.

2. 목적지 설정 요청:

- 백엔드는 ROS2 시스템에 목적지 설정 요청을 전달합니다.

3. 응답 및 UI 업데이트:

- ROS2 시스템과 백엔드의 응답을 받은 후, 프론트엔드는 사용자에게 목적지 설정 완료 메시지를 표시합니다.

12. REQ-012: 차량 도착 정보 표시

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

participant ROS2 as ROS2 System

User->>Frontend: 차량 이동 정보 요청

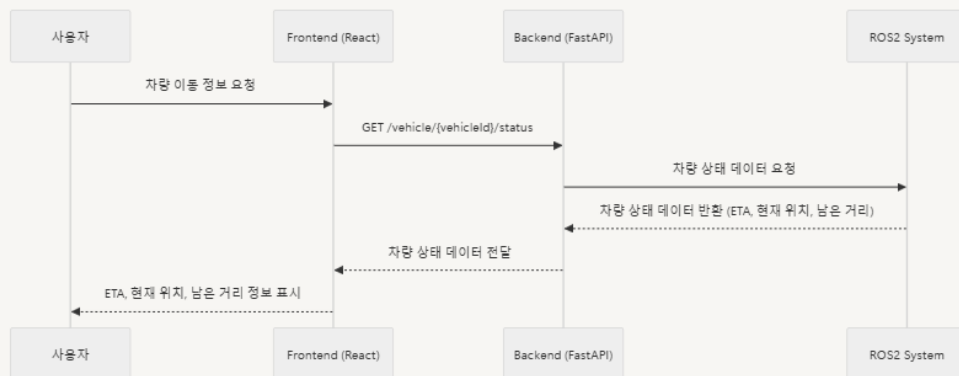
Frontend->>Backend: GET /vehicle/{vehicleId}/status

Backend->>ROS2: 차량 상태 데이터 요청

ROS2-->>Backend: 차량 상태 데이터 반환 (ETA, 현재 위치, 남은 거리)

Backend-->>Frontend: 차량 상태 데이터 전달

Frontend-->>User: ETA, 현재 위치, 남은 거리 정보 표시



설명:

1. 차량 상태 요청:

- 사용자가 차량 이동 정보를 요청하면, 프론트엔드는 백엔드에 차량 상태 데이터를 요청합니다.

2. 차량 상태 데이터 수집:

- 백엔드는 ROS2 시스템에서 차량의 ETA, 현재 위치, 남은 거리 데이터를 조회합니다.

3. 정보 표시:

- 프론트엔드는 반환된 데이터를 사용자에게 표시하여 차량 이동 정보를 실시간으로 제공합니다.

13. REQ-013: 대여 요청

sequenceDiagram

participant User as 사용자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Rental Data)
participant ROS2 as ROS2 System

User->>Frontend: 대여 신청 정보 입력 후 요청

Frontend->>Backend: POST /rent/request (모듈, 옵션, 대여 기간, 목적지 정보)

Backend->>DB: 대여 신청 정보 저장

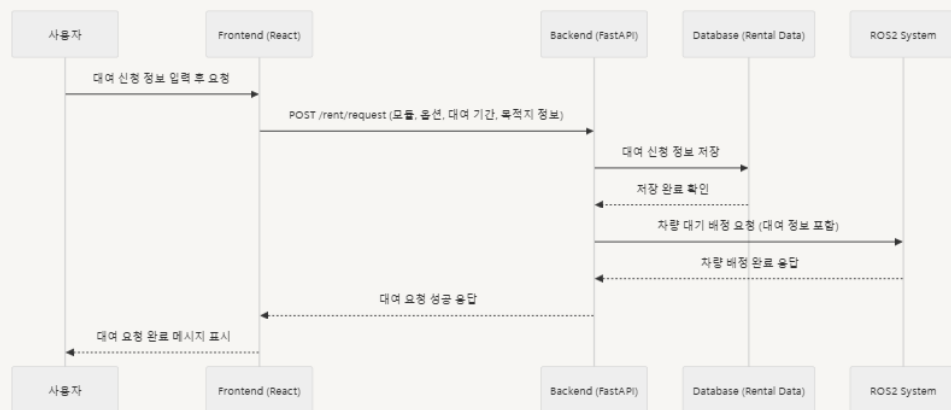
DB-->>Backend: 저장 완료 확인

Backend->>ROS2: 차량 대기 배정 요청 (대여 정보 포함)

ROS2-->>Backend: 차량 배정 완료 응답

Backend-->>Frontend: 대여 요청 성공 응답

Frontend-->>User: 대여 요청 완료 메시지 표시



설명:

1. 대여 요청 입력:

- 사용자가 모듈, 옵션, 대여 기간, 목적지 정보를 입력하고 대여 신청을 요청합니다.

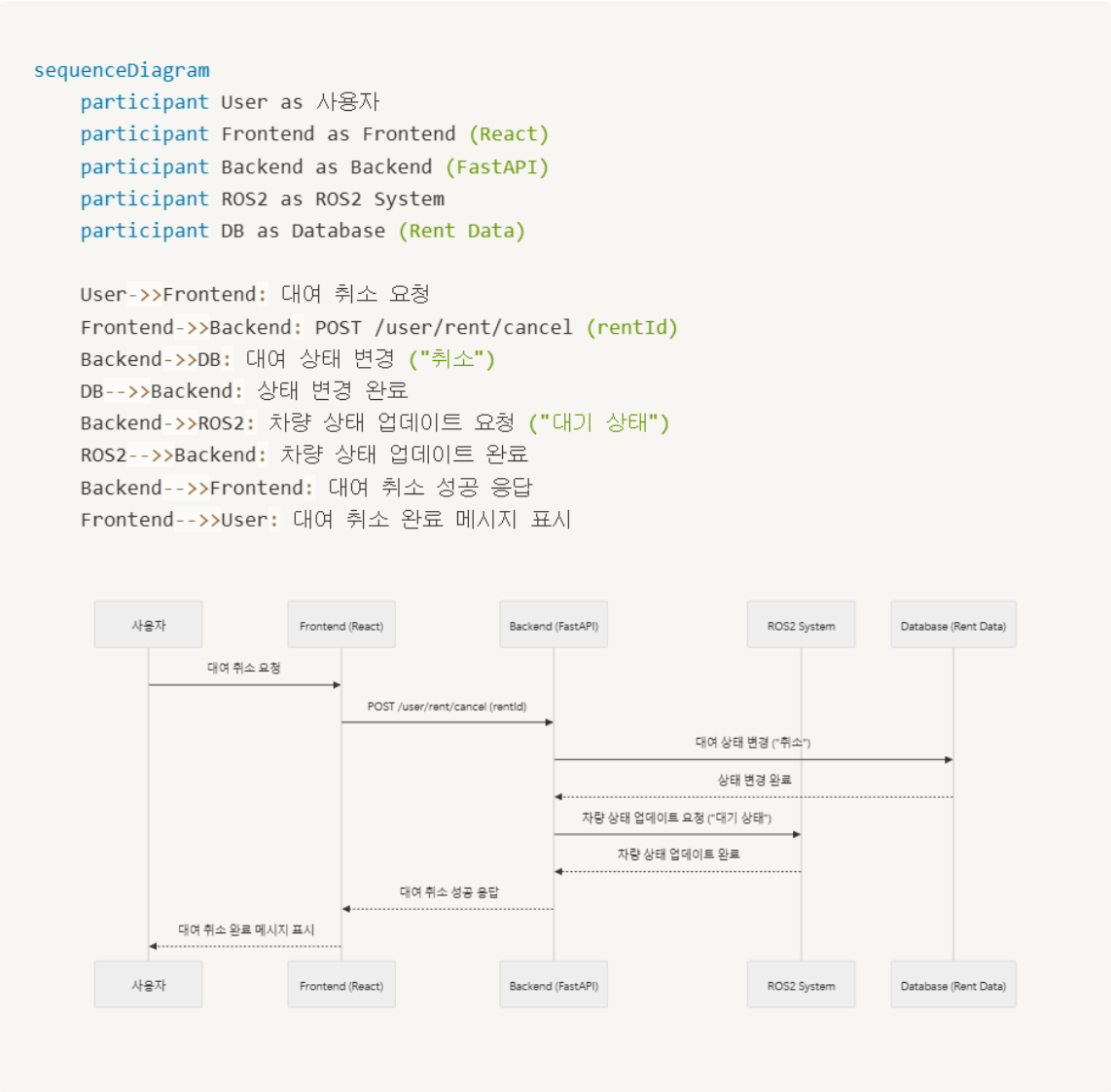
2. 대여 정보 저장:

- 백엔드는 데이터베이스에 대여 정보를 저장한 뒤, ROS2 시스템에 차량 배정을 요청합니다.

3. 대여 완료 응답:

- ROS2로부터 차량 배정 완료 응답을 받은 후, 프론트엔드는 사용자에게 대여 요청 완료 메시지를 표시합니다.

14. REQ-014: 차량 대여 취소



설명:

1. 대여 취소 요청:
 - 사용자가 대여 취소를 요청하면, 프론트엔드는 해당 대여 ID를 백엔드로 전달합니다.
2. 상태 변경 및 차량 업데이트:
 - 백엔드는 데이터베이스에서 대여 상태를 "취소"로 변경하고, ROS2 시스템에 차량 상태를 "대기 상태"로 업데이트 요청합니다.
3. 응답 처리:
 - 모든 작업이 완료되면 프론트엔드는 사용자에게 대여 취소 완료 메시지를 표시합니다.

15. REQ-015: 차량 운행 중 정보 표시

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

participant ROS2 as ROS2 System

User->>Frontend: 차량 운행 중 정보 요청

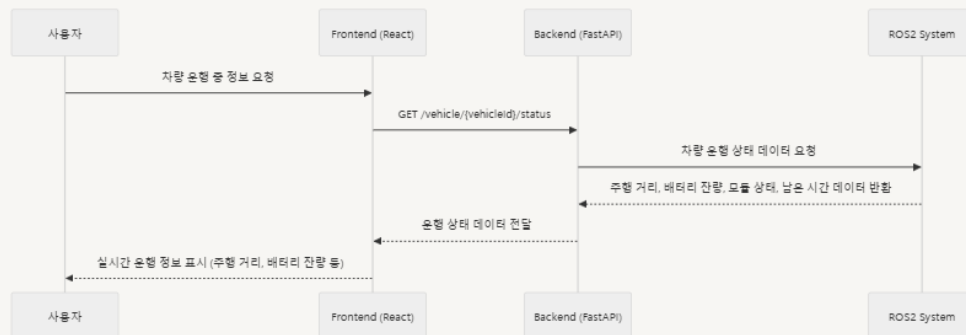
Frontend->>Backend: GET /vehicle/{vehicleId}/status

Backend->>ROS2: 차량 운행 상태 데이터 요청

ROS2-->>Backend: 주행 거리, 배터리 잔량, 모듈 상태, 남은 시간 데이터 반환

Backend-->>Frontend: 운행 상태 데이터 전달

Frontend-->>User: 실시간 운행 정보 표시 (주행 거리, 배터리 잔량 등)



설명:

1. 운행 정보 요청:

- 사용자가 차량 운행 중 정보를 요청하면, 프론트엔드는 백엔드에 차량 상태 데이터를 요청합니다.

2. 운행 상태 데이터 수집:

- 백엔드는 ROS2 시스템에서 주행 거리, 배터리 잔량, 모듈 상태, 남은 시간 등의 데이터를 조회합니다.

3. 운행 정보 표시:

- 프론트엔드는 반환된 데이터를 기반으로 실시간 운행 정보를 사용자에게 표시합니다.

16. REQ-016: 차량 운행 중 목적지 변경

sequenceDiagram

participant User as 사용자

participant Frontend as Frontend (React)

participant Backend as Backend (FastAPI)

participant ROS2 as ROS2 System

User->>Frontend: 새로운 목적지 입력 요청

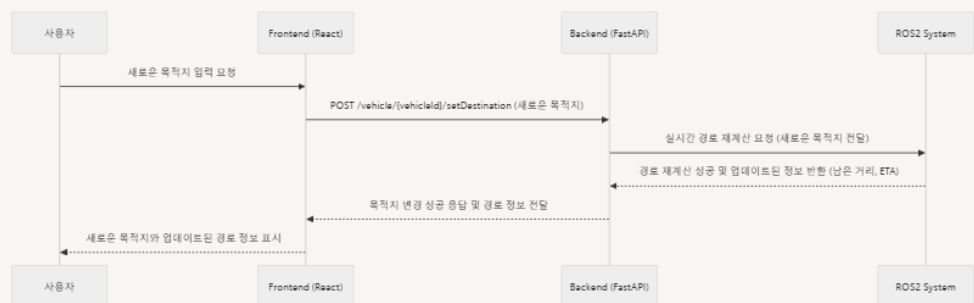
Frontend->>Backend: POST /vehicle/{vehicleId}/setDestination (새로운 목적지)

Backend->>ROS2: 실시간 경로 재계산 요청 (새로운 목적지 전달)

ROS2-->>Backend: 경로 재계산 성공 및 업데이트된 정보 반환 (남은 거리, ETA)

Backend-->>Frontend: 목적지 변경 성공 응답 및 경로 정보 전달

Frontend-->>User: 새로운 목적지와 업데이트된 경로 정보 표시



설명:

1. 새로운 목적지 요청:

- 사용자가 차량 운행 중 새로운 목적지를 입력하면, 프론트엔드는 백엔드에 해당 정보를 전달합니다.

2. 실시간 경로 재계산:

- 백엔드는 ROS2 시스템에 실시간 경로 재계산 요청을 보내고, 새로운 목적지에 따른 업데이트된 경로 정보를 수신합니다.

3. 응답 처리 및 정보 표시:

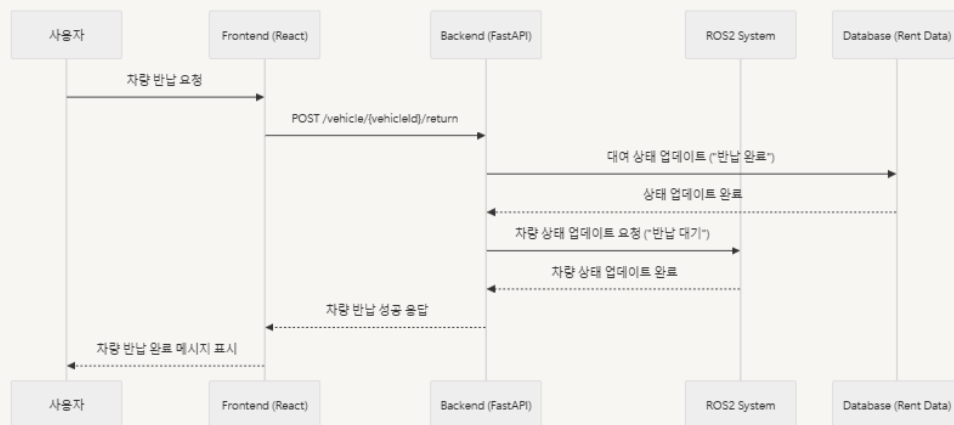
- 프론트엔드는 반환된 정보를 사용자에게 표시하여 새로운 목적지와 업데이트된 운행 경로를 제공합니다.

17. REQ-017: 차량 반납

sequenceDiagram

participant User as 사용자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant ROS2 as ROS2 System
participant DB as Database (Rent Data)

User->>Frontend: 차량 반납 요청
Frontend->>Backend: POST /vehicle/{vehicleId}/return
Backend->>DB: 대여 상태 업데이트 ("반납 완료")
DB-->>Backend: 상태 업데이트 완료
Backend->>ROS2: 차량 상태 업데이트 요청 ("반납 대기")
ROS2-->>Backend: 차량 상태 업데이트 완료
Backend-->>Frontend: 차량 반납 성공 응답
Frontend-->>User: 차량 반납 완료 메시지 표시



설명:

1. 차량 반납 요청:

- 사용자가 차량 반납을 요청하면, 프론트엔드는 백엔드에 요청을 전달합니다.

2. 대여 상태 업데이트:

- 백엔드는 데이터베이스에서 해당 대여 기록의 상태를 "반납 완료"로 업데이트합니다.

3. 차량 상태 변경:

- 백엔드는 ROS2 시스템에 차량 상태를 "반납 대기"로 업데이트하도록 요청합니다.

4. 응답 처리 및 완료 메시지:

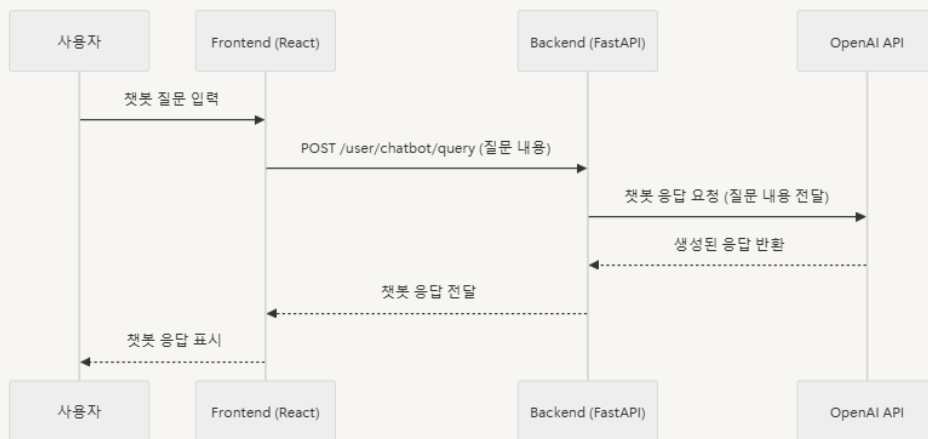
- 모든 작업이 완료되면 프론트엔드는 사용자에게 차량 반납 완료 메시지를 표시합니다.

18. REQ-018: 챗봇 기능

sequenceDiagram

```
participant User as 사용자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant OpenAI as OpenAI API
```

```
User->>Frontend: 챗봇 질문 입력
Frontend->>Backend: POST /user/chatbot/query (질문 내용)
Backend->>OpenAI: 챗봇 응답 요청 (질문 내용 전달)
OpenAI-->>Backend: 생성된 응답 반환
Backend-->>Frontend: 챗봇 응답 전달
Frontend-->>User: 챗봇 응답 표시
```



설명:

1. 질문 입력 요청:

- 사용자가 챗봇 창에 질문을 입력하면, 프론트엔드는 질문 내용을 백엔드로 전달합니다.

2. 챗봇 응답 생성:

- 백엔드는 OpenAI API를 호출하여 질문에 대한 적절한 응답을 생성합니다.

3. 응답 처리 및 표시:

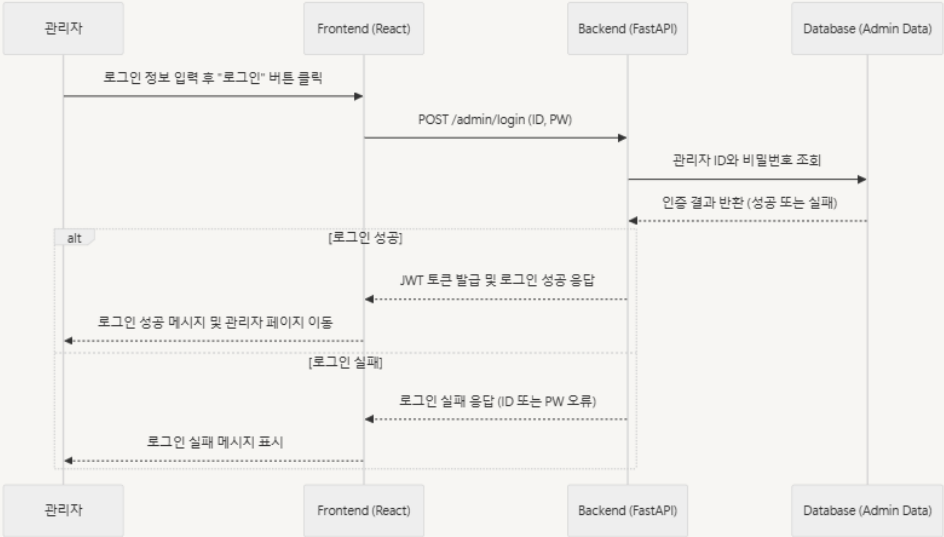
- 백엔드에서 반환된 응답을 프론트엔드가 사용자에게 표시합니다.

19. REQ-019: 관리자 로그인

```
sequenceDiagram
    participant Admin as 관리자
    participant Frontend as Frontend (React)
    participant Backend as Backend (FastAPI)
    participant DB as Database (Admin Data)

    Admin->>Frontend: 로그인 정보 입력 후 "로그인" 버튼 클릭
    Frontend->>Backend: POST /admin/login (ID, PW)
    Backend->>DB: 관리자 ID와 비밀번호 조회
    DB-->>Backend: 인증 결과 반환 (성공 또는 실패)

    alt 로그인 성공
        Backend-->>Frontend: JWT 토큰 발급 및 로그인 성공 응답
        Frontend-->>Admin: 로그인 성공 메시지 및 관리자 페이지 이동
    else 로그인 실패
        Backend-->>Frontend: 로그인 실패 응답 (ID 또는 PW 오류)
        Frontend-->>Admin: 로그인 실패 메시지 표시
    end
end
```



설명:

1. 로그인 요청:

- 관리자가 로그인 정보를 입력하고 "로그인" 버튼을 클릭하면, 프론트엔드는 ID와 비밀번호를 백엔드로 전달합니다.

2. 관리자 인증:

- 백엔드는 데이터베이스에서 관리자 ID와 비밀번호를 조회하여 인증 결과를 반환합니다.

3. 응답 처리:

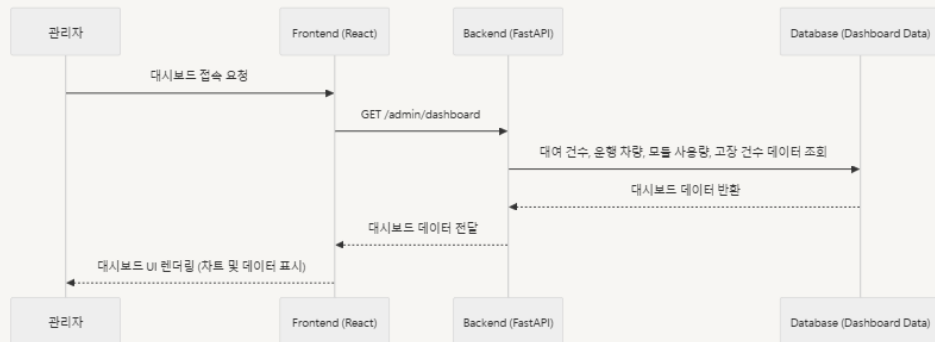
- 인증 성공 시 JWT 토큰이 발급되며 관리자 페이지로 이동합니다.
- 실패 시 로그인 실패 메시지가 표시됩니다.

20. REQ-020: 관리자 대시보드

sequenceDiagram

participant Admin as 관리자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Dashboard Data)

Admin->>Frontend: 대시보드 접속 요청
Frontend->>Backend: GET /admin/dashboard
Backend->>DB: 대여 건수, 운행 차량, 모듈 사용량, 고장 건수 데이터 조회
DB-->>Backend: 대시보드 데이터 반환
Backend-->>Frontend: 대시보드 데이터 전달
Frontend-->>Admin: 대시보드 UI 렌더링 (차트 및 데이터 표시)



설명:

1. 대시보드 요청:

- 관리자가 대시보드 페이지에 접속하면, 프론트엔드는 백엔드에 대시보드 데이터를 요청합니다.

2. 데이터 조회:

- 백엔드는 데이터베이스에서 대여 건수, 운행 차량, 모듈 사용량, 고장 건수 등의 데이터를 조회합니다.

3. UI 렌더링:

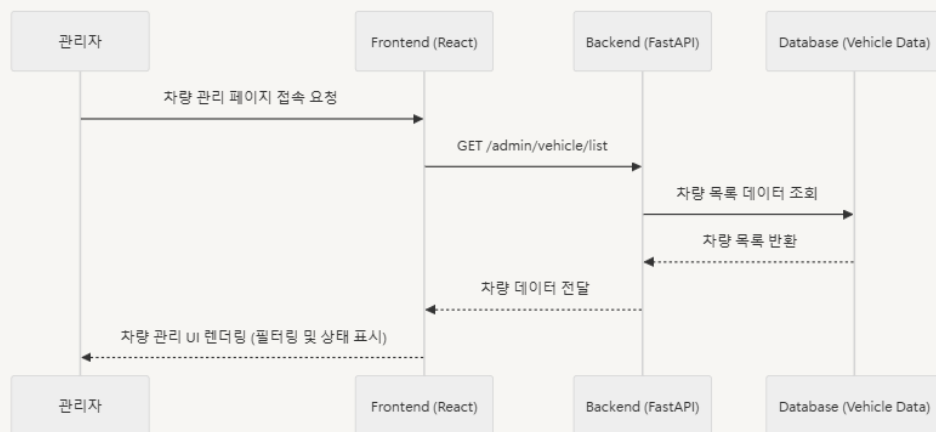
- 프론트엔드는 반환된 데이터를 기반으로 차트와 데이터가 포함된 대시보드를 렌더링하여 관리자에게 표시합니다.

21. REQ-021: 차량 관리 페이지

sequenceDiagram

participant Admin as 관리자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Vehicle Data)

Admin->>Frontend: 차량 관리 페이지 접속 요청
Frontend->>Backend: GET /admin/vehicle/list
Backend->>DB: 차량 목록 데이터 조회
DB-->>Backend: 차량 목록 반환
Backend-->>Frontend: 차량 데이터 전달
Frontend-->>Admin: 차량 관리 UI 렌더링 (필터링 및 상태 표시)



설명:

1. 차량 관리 페이지 요청:

- 관리자가 차량 관리 페이지에 접속하면, 프론트엔드는 백엔드에 차량 데이터 요청을 보냅니다.

2. 차량 데이터 조회:

- 백엔드는 데이터베이스에서 차량 목록 데이터를 조회하여 반환합니다.

3. UI 렌더링:

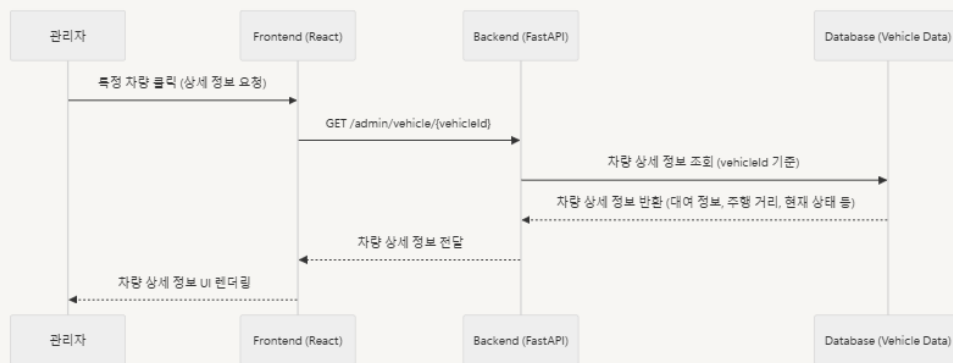
- 프론트엔드는 차량 데이터를 기반으로 필터링 및 상태 표시가 포함된 차량 관리 페이지를 렌더링하여 관리자가 볼 수 있도록 합니다.

22. REQ-022: 차량 정보 상세 조회

sequenceDiagram

participant Admin as 관리자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Vehicle Data)

Admin->>Frontend: 특정 차량 클릭 (상세 정보 요청)
Frontend->>Backend: GET /admin/vehicle/{vehicleId}
Backend->>DB: 차량 상세 정보 조회 (vehicleId 기준)
DB-->>Backend: 차량 상세 정보 반환 (대여 정보, 주행 거리, 현재 상태 등)
Backend-->>Frontend: 차량 상세 정보 전달
Frontend-->>Admin: 차량 상세 정보 UI 렌더링



설명:

1. 상세 정보 요청:

- 관리자가 특정 차량을 클릭하면, 프론트엔드는 해당 차량의 상세 정보를 백엔드에 요청합니다.

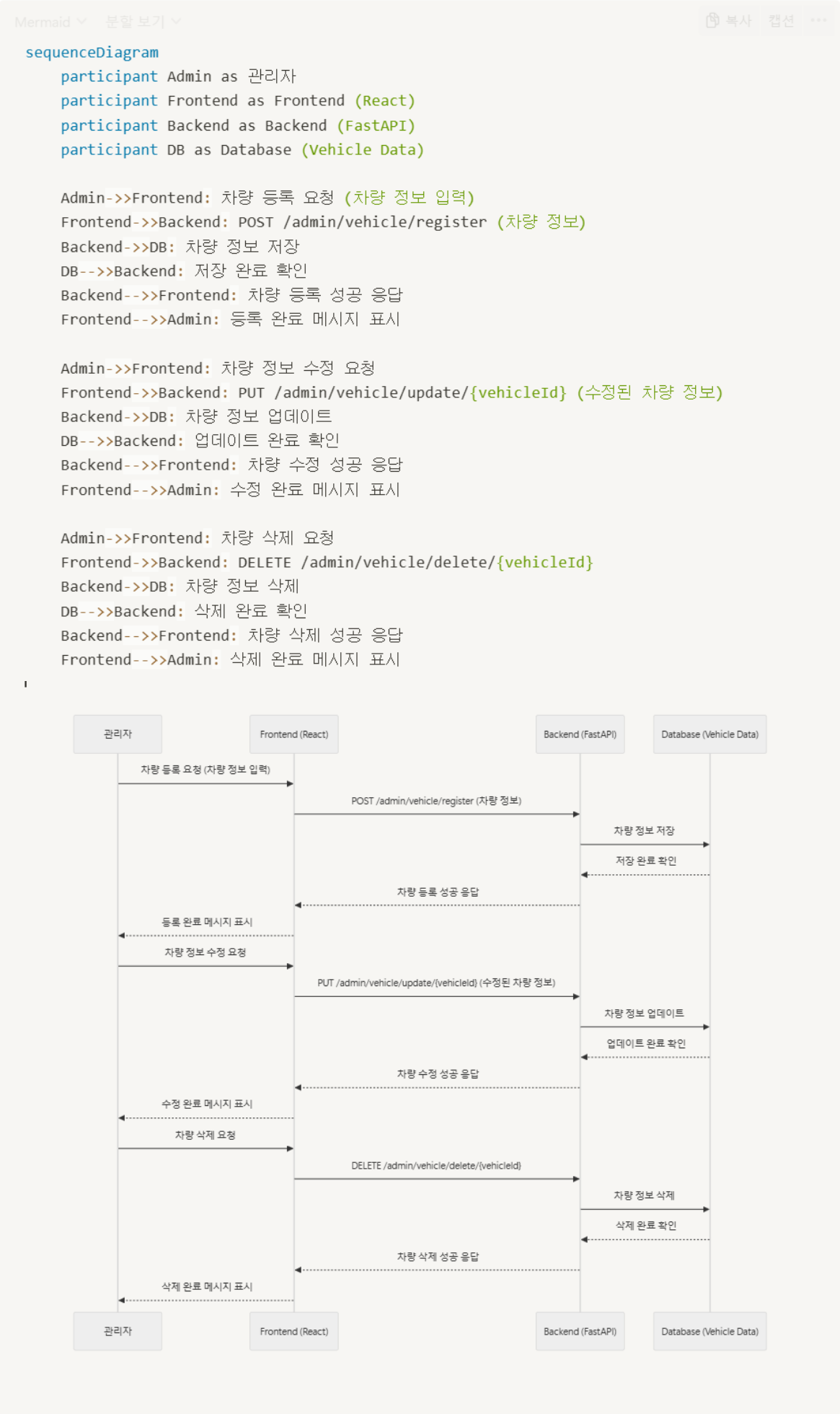
2. 차량 데이터 조회:

- 백엔드는 데이터베이스에서 요청받은 차량 ID를 기준으로 대여 정보, 주행 거리, 현재 상태 등의 상세 정보를 조회합니다.

3. UI 렌더링:

- 프론트엔드는 조회된 데이터를 기반으로 차량의 상세 정보를 관리자에게 표시합니다.

23. REQ-023: 차량 등록/수정/삭제



설명:

1. 차량 등록:

- 관리자가 차량 정보를 입력하고 등록 요청을 하면, 차량 정보가 데이터베이스에 저장됩니다.

2. 차량 수정:

- 관리자가 수정된 정보를 입력하여 요청하면, 데이터베이스에서 해당 차량의 정보가 업데이트됩니다.

3. 차량 삭제:

- 관리자가 차량 삭제 요청을 하면, 데이터베이스에서 해당 차량 정보가 삭제됩니다.

4. 응답 처리:

- 각 작업 완료 후, 프론트엔드는 관리자에게 성공 메시지를 표시합니다.

24. REQ-024: 모듈/옵션 관리 페이지

sequenceDiagram

participant Admin as 관리자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Module/Option Data)

Admin->>Frontend: 모듈/옵션 관리 페이지 접속 요청

Frontend->>Backend: GET /admin/module-set/list

Backend->>DB: 모듈 세트 데이터 조회

DB-->>Backend: 모듈 세트 데이터 반환

Backend-->>Frontend: 모듈 세트 데이터 전달

Frontend-->>Admin: 모듈 세트 UI 렌더링

Admin->>Frontend: 특정 모듈 수정/삭제 요청

Frontend->>Backend: PUT/DELETE /admin/module/update/{moduleId} 또는 /admin/module/delete/{moduleId}

Backend->>DB: 모듈 정보 수정 또는 삭제

DB-->>Backend: 수정/삭제 완료 확인

Backend-->>Frontend: 성공 응답 전달

Frontend-->>Admin: 수정/삭제 완료 메시지 표시

Admin->>Frontend: 새로운 옵션 등록 요청

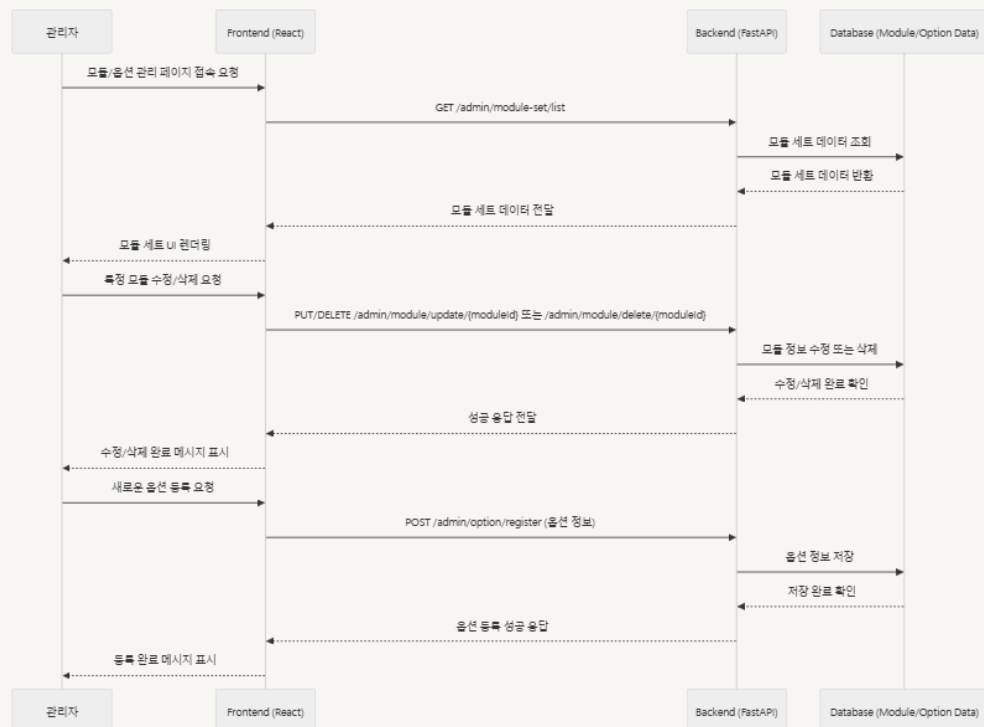
Frontend->>Backend: POST /admin/option/register (옵션 정보)

Backend->>DB: 옵션 정보 저장

DB-->>Backend: 저장 완료 확인

Backend-->>Frontend: 옵션 등록 성공 응답

Frontend-->>Admin: 등록 완료 메시지 표시



설명:

1. 모듈/옵션 조회:

- 관리자가 모듈/옵션 관리 페이지에 접속하면, 프론트엔드는 백엔드에서 모듈 세트 데이터를 조회하여 표시합니다.

2. 모듈 수정/삭제:

- 관리자가 특정 모듈을 수정 또는 삭제하면, 요청에 따라 데이터베이스에서 변경이 이루어집니다.

3. 옵션 등록:

- 관리자가 새로운 옵션 정보를 입력하고 등록 요청을 하면, 옵션 데이터가 데이터베이스에 저장됩니다.

4. 응답 처리:

- 각 작업 완료 후, 프론트엔드는 관리자에게 성공 메시지를 표시합니다.

25. REQ-025: 유지보수/정비 관리

sequenceDiagram

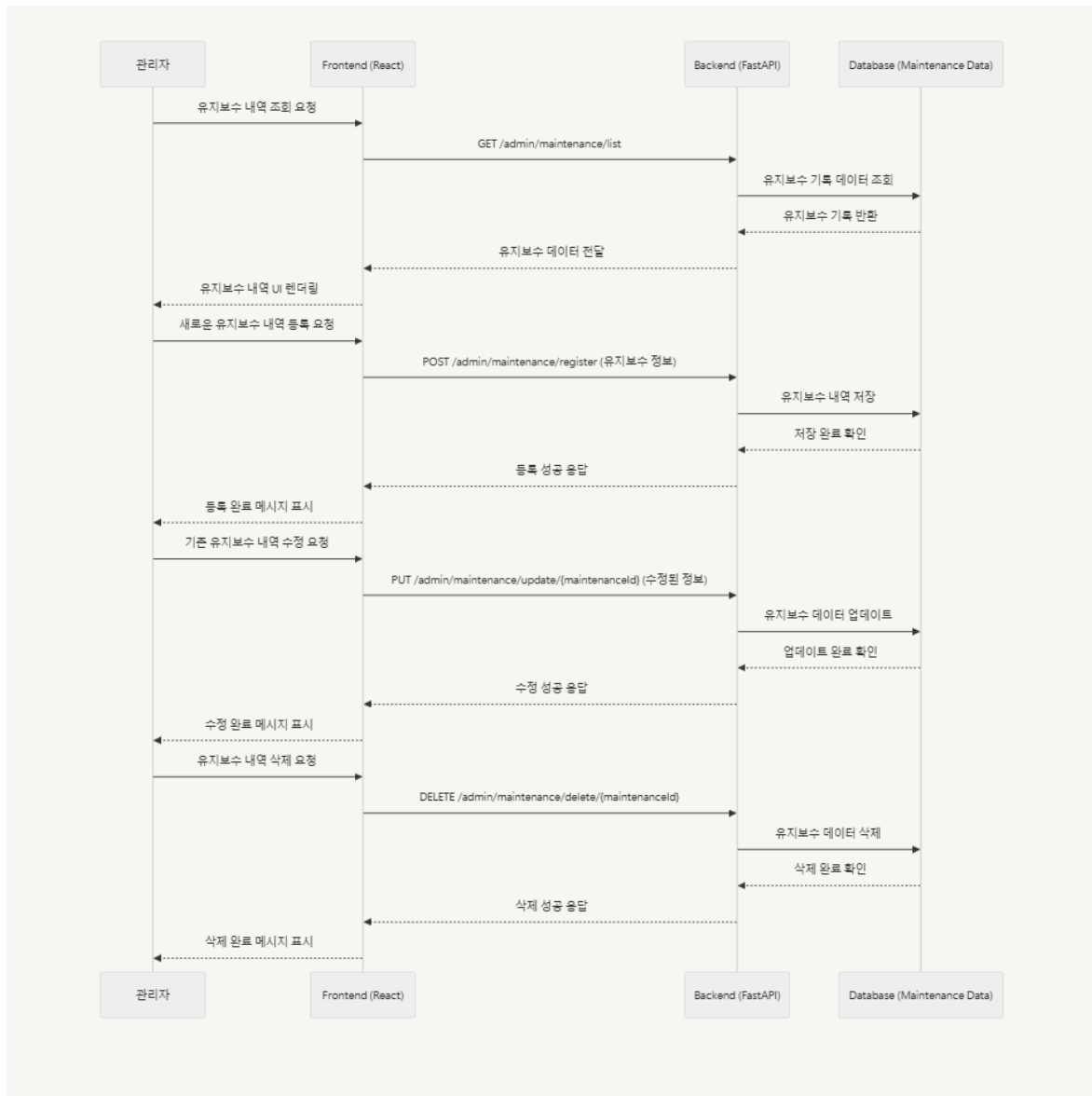
participant Admin as 관리자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Maintenance Data)

Admin->>Frontend: 유지보수 내역 조회 요청
Frontend->>Backend: GET /admin/maintenance/list
Backend->>DB: 유지보수 기록 데이터 조회
DB-->>Backend: 유지보수 기록 반환
Backend-->>Frontend: 유지보수 데이터 전달
Frontend-->>Admin: 유지보수 내역 UI 렌더링

Admin->>Frontend: 새로운 유지보수 내역 등록 요청
Frontend->>Backend: POST /admin/maintenance/register (유지보수 정보)
Backend->>DB: 유지보수 내역 저장
DB-->>Backend: 저장 완료 확인
Backend-->>Frontend: 등록 성공 응답
Frontend-->>Admin: 등록 완료 메시지 표시

Admin->>Frontend: 기존 유지보수 내역 수정 요청
Frontend->>Backend: PUT /admin/maintenance/update/{maintenanceId} (수정된 정보)
Backend->>DB: 유지보수 데이터 업데이트
DB-->>Backend: 업데이트 완료 확인
Backend-->>Frontend: 수정 성공 응답
Frontend-->>Admin: 수정 완료 메시지 표시

Admin->>Frontend: 유지보수 내역 삭제 요청
Frontend->>Backend: DELETE /admin/maintenance/delete/{maintenanceId}
Backend->>DB: 유지보수 데이터 삭제
DB-->>Backend: 삭제 완료 확인
Backend-->>Frontend: 삭제 성공 응답
Frontend-->>Admin: 삭제 완료 메시지 표시



설명:

1. 유지보수 내역 조회:

- 관리자가 유지보수 내역을 조회하면, 프론트엔드는 백엔드에서 데이터를 요청하고 반환된 데이터를 UI에 표시합니다.

2. 유지보수 내역 등록:

- 관리자가 새로운 유지보수 정보를 등록하면, 해당 데이터가 데이터베이스에 저장됩니다.

3. 유지보수 내역 수정:

- 관리자가 기존 내역을 수정 요청하면, 데이터베이스에서 해당 데이터를 업데이트합니다.

4. 유지보수 내역 삭제:

- 관리자가 특정 내역을 삭제 요청하면, 데이터베이스에서 해당 데이터가 삭제됩니다.

5. 응답 처리:

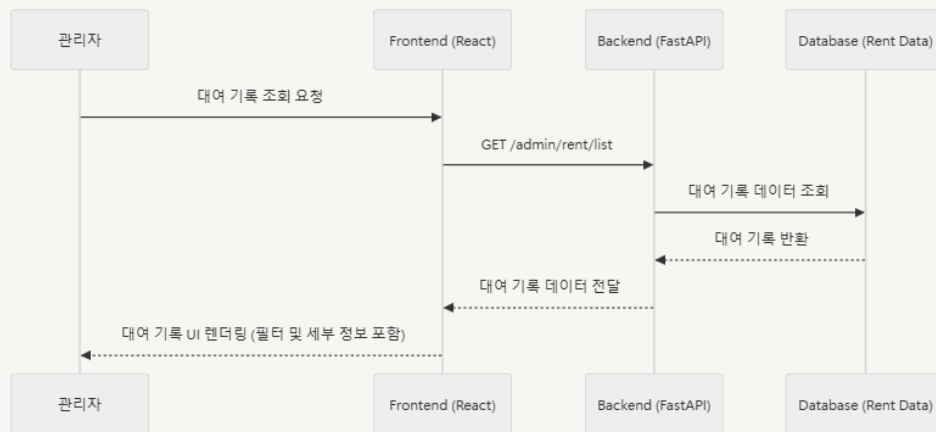
- 각 작업 완료 후, 프론트엔드는 관리자에게 완료 메시지를 표시합니다.

26. REQ-026: 대여 기록 조회

sequenceDiagram

```
participant Admin as 관리자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Rent Data)
```

```
Admin->>Frontend: 대여 기록 조회 요청
Frontend->>Backend: GET /admin/rent/list
Backend->>DB: 대여 기록 데이터 조회
DB-->>Backend: 대여 기록 반환
Backend-->>Frontend: 대여 기록 데이터 전달
Frontend-->>Admin: 대여 기록 UI 렌더링 (필터 및 세부 정보 포함)
```



설명:

1. 대여 기록 요청:

- 관리자가 대여 기록을 조회하면, 프론트엔드는 백엔드에 대여 기록 데이터를 요청합니다.

2. 데이터 조회:

- 백엔드는 데이터베이스에서 모든 대여 기록(대여 ID, 사용자, 차량, 모듈, 대여 기간 등)을 조회하여 반환합니다.

3. UI 렌더링:

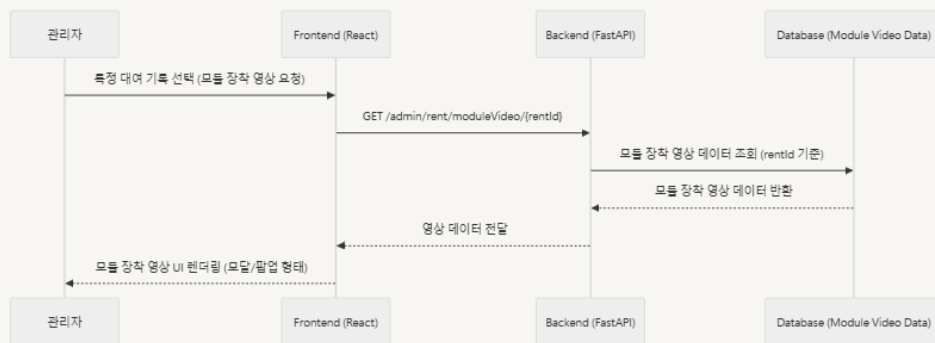
- 프론트엔드는 조회된 대여 기록 데이터를 기반으로 필터링 및 세부 정보가 포함된 UI를 관리자에게 표시합니다.

27. REQ-027: 모듈 장착 영상 확인

sequenceDiagram

participant Admin as 관리자
participant Frontend as Frontend (React)
participant Backend as Backend (FastAPI)
participant DB as Database (Module Video Data)

Admin->>Frontend: 특정 대여 기록 선택 (모듈 장착 영상 요청)
Frontend->>Backend: GET /admin/rent/moduleVideo/{rentId}
Backend->>DB: 모듈 장착 영상 데이터 조회 (rentId 기준)
DB-->>Backend: 모듈 장착 영상 데이터 반환
Backend-->>Frontend: 영상 데이터 전달
Frontend-->>Admin: 모듈 장착 영상 UI 렌더링 (모달/팝업 형태)



설명:

1. 모듈 장착 영상 요청:

- 관리자가 특정 대여 기록을 선택하면, 프론트엔드는 해당 대여 기록의 모듈 장착 영상을 요청합니다.

2. 데이터 조회:

- 백엔드는 데이터베이스에서 해당 대여 기록(rentId)의 모듈 장착 영상을 조회합니다.

3. UI 렌더링:

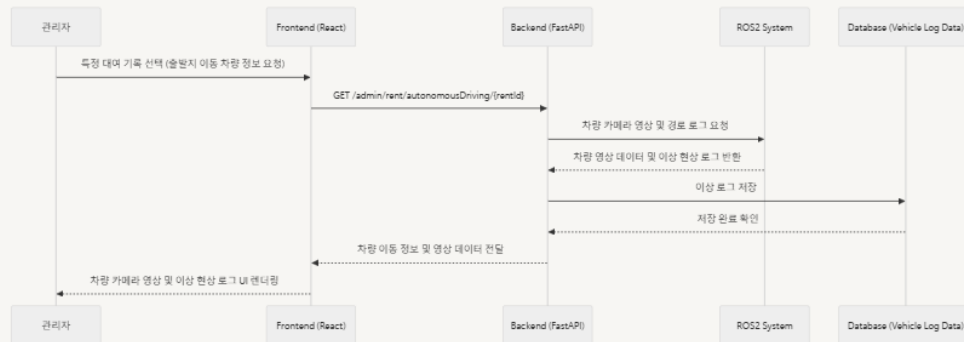
- 프론트엔드는 반환된 영상 데이터를 기반으로 모달 또는 팝업 형태로 영상을 관리자에게 표시합니다.

28. REQ-028: 출발지 이동 차량 확인 기능

sequenceDiagram

participant Admin as 관리자
 participant Frontend as Frontend (React)
 participant Backend as Backend (FastAPI)
 participant ROS2 as ROS2 System
 participant DB as Database (Vehicle Log Data)

Admin->>Frontend: 특정 대여 기록 선택 (출발지 이동 차량 정보 요청)
 Frontend->>Backend: GET /admin/rent/autonomousDriving/{rentId}
 Backend->>ROS2: 차량 카메라 영상 및 경로 로그 요청
 ROS2-->>Backend: 차량 영상 데이터 및 이상 현상 로그 반환
 Backend->>DB: 이상 로그 저장
 DB-->>Backend: 저장 완료 확인
 Backend-->>Frontend: 차량 이동 정보 및 영상 데이터 전달
 Frontend-->>Admin: 차량 카메라 영상 및 이상 현상 로그 UI 렌더링



설명:

1. 출발지 이동 정보 요청:

- 관리자가 특정 대여 기록을 선택하면, 프론트엔드는 출발지로 이동 중인 차량의 정보를 백엔드에 요청합니다.

2. 카메라 영상 및 로그 조회:

- 백엔드는 ROS2 시스템에서 차량의 카메라 영상 데이터와 경로 이탈 등의 이상 현상 로그를 요청합니다.

3. 이상 로그 저장:

- ROS2로부터 받은 이상 현상 로그를 데이터베이스에 저장합니다.

4. UI 렌더링:

- 반환된 차량 이동 정보와 영상을 프론트엔드는 UI로 렌더링하여 관리자가 확인할 수 있도록 표시합니다.