

---

# 카카오톡 계정 관리 시스템

C팀 : 박정호, 김진호, 신재하, 김수진

---

## 01. 유스케이스 다이어그램

---

## 02. 클래스 다이어그램

---

## 03. 시퀀스 다이어그램

---

## 04. Q & A

---

## 시스템 상황 분석

---

### 카카오톡 계정관리 시스템

- 카카오톡 계정관리 시스템은 회원가입, 로그인, 로그아웃, 아이디찾기, 비밀번호 재설정, 회원정보 수정, 연결 서비스 삭제, 계정탈퇴 기능이 있다.

### 카카오톡 DB 시스템

- 입력한 사용자의 아이디와 비밀번호로 회원을 확인
- 사용자의 정보를 제공 또는 수정 할 수 있다.

### 회원가입

- 사용자는 회원가입을 할 때 본인인증을 통해 인증을 성공하면 계정을 생성 할 수 있다.
- 1인당 하나의 계정만 만들 수 있으며 회원가입이 되어 있어야만 나머지 기능들을 사용할 수 있다.

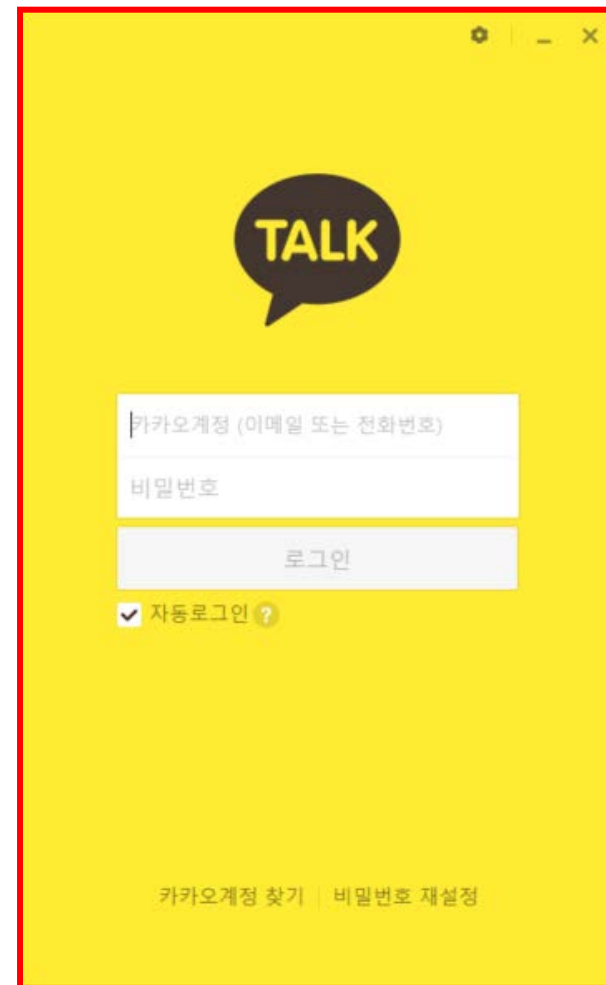
## 시스템 상황 분석

### 로그인

- 사용자는 입력한 아이디와 비밀번호가 DB시스템에 있다면 로그인을 한다.
- 자동 로그인 체크박스에 체크를 한 후 로그인을 하면 이 후 자동 로그인을 할 수 있다.

### 로그아웃

- 사용자는 기기에 로그인되어 있는 계정을 로그아웃 할 수 있다.
- 세션이 만료되면 자동으로 로그아웃 한다.



## 시스템 상황 분석

### 아이디 찾기

- 사용자는 외부 인증 시스템으로 본인인증을 하거나 회원 확인(프로필)을 통해 아이디를 카카오톡 DB 시스템에서 찾을 수 있다.
- 회원확인을 이용한 아이디 찾기는 암호화된 아이디를 보여주고, 외부 인증 시스템을 이용한 아이디 찾기는 암호화되지 않은 아이디를 보여준다.

### 비밀번호 재설정

- 사용자는 아이디를 입력하고, 외부 인증 시스템(이메일로 인증, 휴대폰으로 인증 등..)에서 본인 인증을 통과하면 카카오톡 DB 시스템에서 해당하는 ID에 대한 비밀번호를 재설정 할 수 있다.

카카오계정을 찾을 방법을  
선택해 주세요.

본인인증

휴대폰에 등록된 본인확인 정보로 계정을 찾습니다.

프로필 / 전화번호로 찾기

카카오계정에서 사용한 프로필 이름 및 전화번호를  
입력해주세요.

프로필 / 이메일로 찾기

카카오 서비스에서 이용한 프로필 및 계정의 연락처  
이메일로 계정을 찾습니다.

❗ 비즈니스 계정을 찾으시나요? [사업자등록번호로 찾기](#)

## 시스템 상황 분석

### 계정탈퇴

- 사용자는 카카오톡 계정을 삭제할 수 있다.

<

카카오톡 탈퇴

카카오톡을 탈퇴하면,

- 내 프로필, 친구목록, 대화내용, 구입한 아이템, 선물 받은 아이템(미사용 교환권 등), 그 외 사용자가 설정한 모든 정보가 사라지고 복구가 불가능합니다.
- 참여 중인 모든 대화방에서 나가게 되고, 대화방에서 주고 받은 사진이나 파일 등 모든 정보가 즉시 삭제됩니다. 중요한 정보가 있는지 탈퇴 전에 저장해 주세요.
- 즐겨찾기, 숨김, 차단 여부를 포함한 친구목록 전체가 삭제되고, 카카오톡을 재가입해도 자동으로 복구되지 않습니다.
- 카카오톡을 탈퇴하면서 카카오 계정도 함께 탈퇴되는 경우, 특서랍 서비스는 해지 처리되며 특서랍에 보관된 모든 데이터도 삭제됩니다.

✓

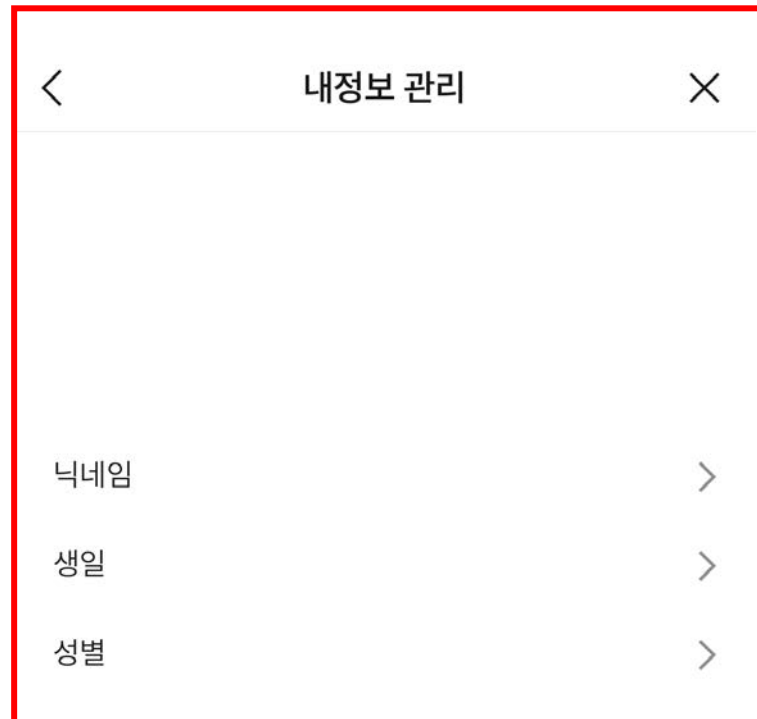
모든 정보를 삭제하는 것에 동의합니다.

카카오톡 탈퇴

## 시스템 상황 분석

### 정보 수정

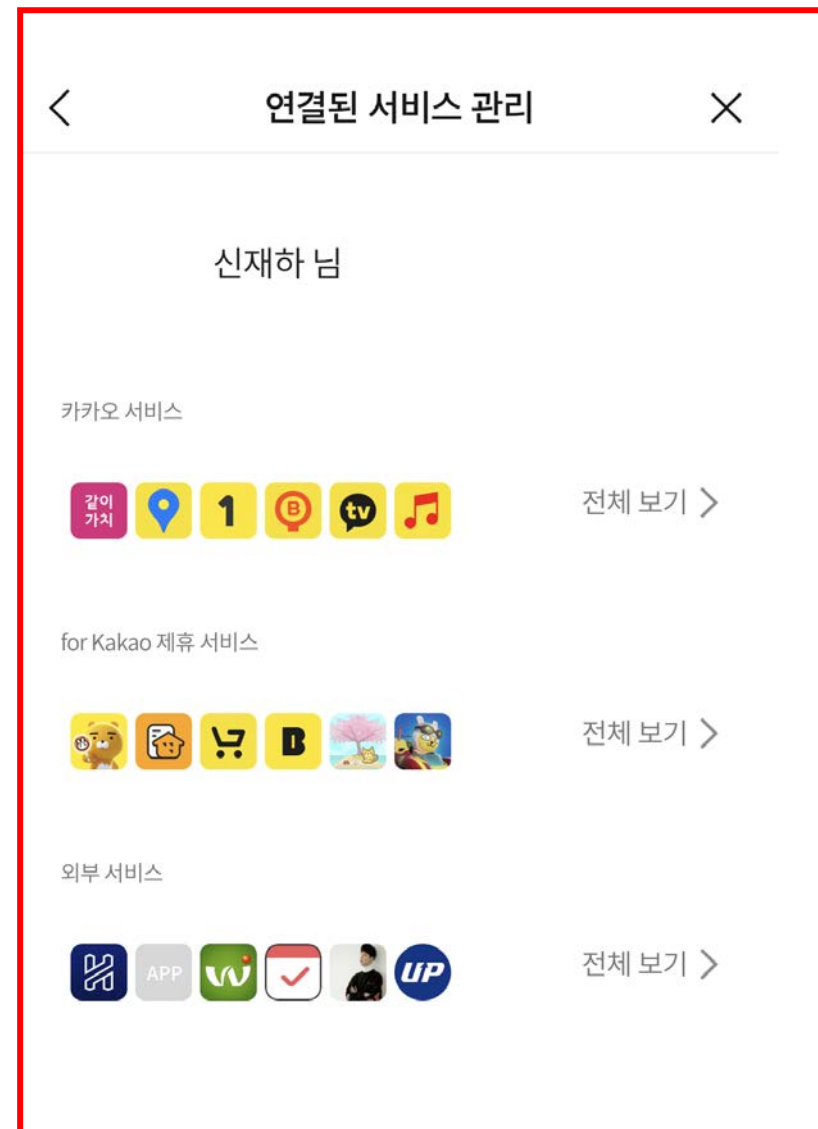
- 사용자는 카카오톡이 로그인되어 있을 때, 2차로 비밀번호를 인증 후, 회원 정보(이름, 전화번호, 성별, 이메일)을 수정 할 수 있다.



## 시스템 상황 분석

### 연결 서비스 관리 (삭제)

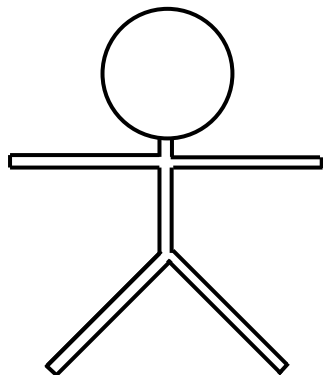
- 사용자는 카카오톡이 로그인되어 있을 때, 2차로 비밀번호를 인증 후, 다른 서비스에 로그인 된 카카오계정 연결을 끊을 수 있다.





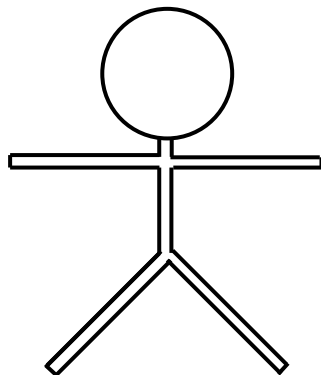
## 액터 식별

---



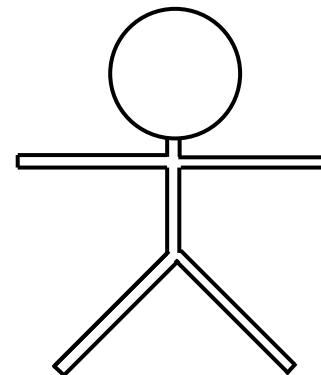
**사용자**

시스템을 사용하는 주체



**카카오톡 DB 시스템**

사용자의 정보를 담고 있는  
DB시스템



**외부 인증 시스템**

본인인증을 하는  
외부 인증 시스템

## 유스케이스 식별

---

U1.회원가입

U2.탈퇴

U3.로그인

U4.로그아웃

U5.아이디 찾기

U6.비밀번호 재설정

U7.정보 수정

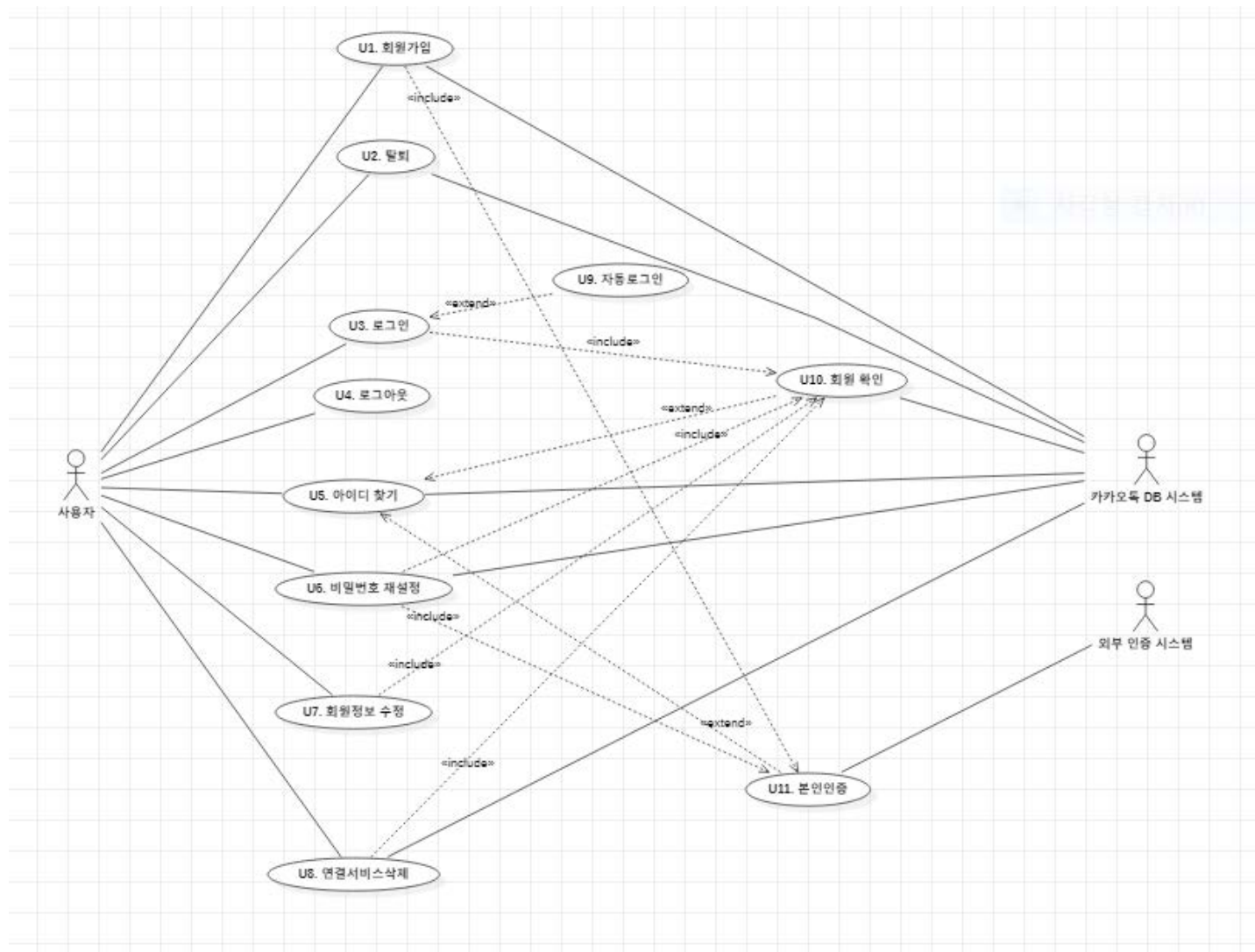
U8.연결 서비스 삭제

U9.자동 로그인

U10.회원 확인

U11.본인 인증

## 유스케이스 다이어그램



## 유스케이스 명세서

유스케이스 명	U1. 회원가입	유스케이스 개요	사용자가 카카오톡 회원으로 가입하기 위한 유스케이스이다.
액터명	사용자, 카카오톡 DB 시스템, 외부 인증 시스템	사전 조건	사용자는 회원으로 가입한 적이 없어야 한다.
정상 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 회원가입을 시도한다.</li> <li>2. 시스템은 서비스약관을 보여준다.</li> <li>3. 사용자는 약관에 동의한다.</li> <li>4. 사용자는 아이디, 비밀번호, 이메일, 닉네임, 전화번호를 입력한다.</li> <li>5. 시스템은 인증번호를 전송한다.</li> <li>6. 사용자는 인증번호를 입력한다.</li> <li>7. 카카오톡 DB 시스템은 기존 가입 회원인지 확인한다.</li> <li>8. 카카오톡 DB 시스템에 회원정보를 저장하고 회원가입에 성공한다.</li> </ol>		
선택 흐름	<ul style="list-style-type: none"> <li>✓ 인증번호가 틀린 경우 인증번호가 올바르지 않다는 메시지를 보여준다.</li> <li>✓ 기존 가입 회원이라면 가입된 계정이 있다는 메시지를 보여준다.</li> </ul>		

# 유스케이스 명세서

유스케이스 명	U2. 탈퇴	유스케이스 개요	사용자가 계정을 탈퇴하기 위한 유스케이스이다.
액터명	사용자, 카카오톡 DB 시스템, 외부 인증 시스템	사전 조건	사용자는 로그인 되어 있어야 한다.
정상 흐름	<div>1. 사용자는 탈퇴를 요청한다.</div> <div>2. 사용자는 개인 정보를 입력한다.</div> <div>3. 외부 인증 시스템은 본인인증을 확인한다.</div> <div>4. 카카오톡 DB시스템은 사용자 정보를 삭제한다.</div>		
선택 흐름	<div>✓ 외부 인증 시스템에서 본인이 아닐 경우 본인 인증에 실패한다.</div>		

## 유스케이스 명세서

유스케이스 명	U3. 로그인	유스케이스 개요	사용자가 로그인을 하기 위한 유스케이스이다.
액터명	사용자, 카카오톡 DB 시스템	사전 조건	사용자는 회원으로 가입되어 있어야 한다.
정상 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 로그인을 시도한다.</li> <li>2. 사용자는 아이디와 비밀번호를 입력한다.</li> <li>3. 사용자는 로그인을 요청한다.</li> <li>4. 카카오톡 DB 시스템은 사용자는 기존 회원인지 확인한다.</li> <li>5. 세션에 로그인정보를 저장한다.</li> <li>6. 사용자는 로그인에 성공한다.</li> </ol>		
선택 흐름	<p>✓ 회원정보가 맞지 않으면 로그인에 실패한다.</p>		

# 유스케이스 명세서

유스케이스 명	U4. 로그아웃	유스케이스 개요	사용자가 로그아웃을 하기 위한 유스케이스이다.
액터명	사용자	사전 조건	사용자는 로그인 상태여야 한다.
정상 흐름	1.    사용자는 로그아웃을 요청한다. 2.    시스템은 세션을 무효화하여 로그인 된 계정을 로그아웃 시킨다.		
선택 흐름	✓ 세션 타임아웃이 발생하면 자동으로 로그아웃 된다.		

## 유스케이스 명세서

유스케이스 명	U5. 아이디 찾기	유스케이스 개요	사용자가 아이디를 분실했을 때 찾기 위한 유스케이스이다.
액터명	사용자, 카카오톡 DB 시스템, 외부 인증 시스템	사전 조건	사용자는 회원으로 가입되어 있어야 한다.
정상 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 아이디 찾기 방법을 선택한다.</li> <li>2. 카카오톡 DB 시스템은 사용자는 기존 회원인지 확인한다.</li> <li>3. 카카오톡 DB 시스템은 사용자에게 아이디를 제공한다.</li> </ol>		
선택 흐름	<ul style="list-style-type: none"> <li>✓ 외부 인증 시스템을 통한 본인 인증을 할 경우 암호화 되지 않은 아이디를 알려준다.</li> <li>✓ 프로필을 이용한 인증을 할 경우 카카오톡 DB 시스템은 암호화 된 아이디를 알려준다.</li> <li>✓ 회원 확인 이후 해당하는 계정이 없으면 "입력하신 정보와 일치하는 계정이 없습니다" 경고창을 출력한다.</li> </ul>		



## 유스케이스 명세서

유스케이스 명	U6. 비밀번호 재설정	유스케이스 개요	사용자가 비밀번호를 재설정하기 위한 유스케이스이다.
액터명	사용자, 카카오톡 DB 시스템, 외부 인증 시스템	사전 조건	고객은 회원으로 가입되어 있어야 한다.
정상 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 비밀번호 재설정을 요청한다.</li> <li>2. 사용자는 아이디를 입력한다.</li> <li>3. 카카오톡 DB 시스템은 기존 가입 회원인지 확인한다.</li> <li>4. 기존 가입 회원이라면 본인인증을 요청한다.</li> <li>5. 외부 인증 시스템은 본인인증을 확인한다.</li> <li>6. 비밀번호를 재설정한다.</li> </ol>		
선택 흐름	<p>✓ 회원 확인 이후 해당하는 계정이 없으면 "입력하신 정보와 일치하는 계정이 없습니다" 경고창을 출력한다.</p>		

## 유스케이스 명세서

유스케이스 명	U7. 정보 수정	유스케이스 개요	사용자가 계정의 정보를 수정하기 위한 유스케이스이다
액터명	사용자, 카카오톡 DB 시스템	사전 조건	사용자는 로그인 상태여야 한다
정상 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 정보 수정을 요청한다.</li> <li>2. 사용자는 사용자 아이디에 해당하는 비밀번호를 입력한다.</li> <li>3. 카카오톡 DB 시스템은 사용자가 입력한 비밀번호가 유효한지 확인한다.</li> <li>4. 사용자가 입력한 비밀번호가 유효하면 사용자는 수정할 정보를 선택한다.</li> <li>5. 사용자는 선택한 정보에 대한 수정할 내용을 입력한다.</li> <li>6. 카카오톡 DB 시스템은 고객의 정보를 수정한다.</li> </ol>		
선택 흐름	<p>✓ 카카오톡 DB 시스템에서 사용자가 입력한 비밀번호가 유효하지 않다면 종료한다.</p>		

## 유스케이스 명세서

유스케이스 명	U8. 연결 서비스 삭제	유스케이스 개요	사용자가 계정에 연결된 서비스를 삭제하는 유스케이스이다.
액터명	사용자, 카카오톡 DB 시스템	사전 조건	사용자는 로그인 상태여야 한다
정상 흐름	<ol style="list-style-type: none"> <li>1. 사용자는 연결 서비스 삭제를 요청한다.</li> <li>2. 사용자는 사용자 아이디에 해당하는 비밀번호를 입력한다.</li> <li>3. 카카오톡 DB 시스템은 사용자가 입력한 비밀번호가 유효한지 확인한다.</li> <li>4. 사용자가 입력한 비밀번호가 유효하면 DB 접근 시스템은 사용자 계정에 연결된 서비스들을 출력한다.</li> <li>5. 사용자는 연결된 서비스들 중 삭제할 서비스를 선택한다.</li> <li>6. 카카오톡 DB 시스템은 사용자가 선택한 서비스를 삭제한다.</li> </ol>		
선택 흐름	<ul style="list-style-type: none"> <li>✓ 카카오톡 DB 시스템에서 사용자가 입력한 비밀번호가 유효하지 않다면 종료한다.</li> <li>✓ 카카오톡 DB 시스템에서 사용자 계정에 연결된 서비스가 없다면 연결된 서비스가 없다는 문구를 출력한다.</li> </ul>		

# 유스케이스 명세서

유스케이스 명	U9. 자동로그인	유스케이스 개요	사용자가 로그인에 성공하면 이후 자동으로 로그인이 되는 유스케이스이다.
액터명	사용자	사전 조건	고객은 회원으로 가입되어 있어야 한다.
정상 흐름	1. 사용자는 자동로그인 체크박스에 체크한다. 2. 사용자는 로그인에 성공한다. 3. 이후 카카오톡 실행 시 사용자는 자동으로 로그인이 된다.		
선택 흐름	✓ 자동로그인에 체크되어 있지 않으면 일반적인 로그인을 한다.		

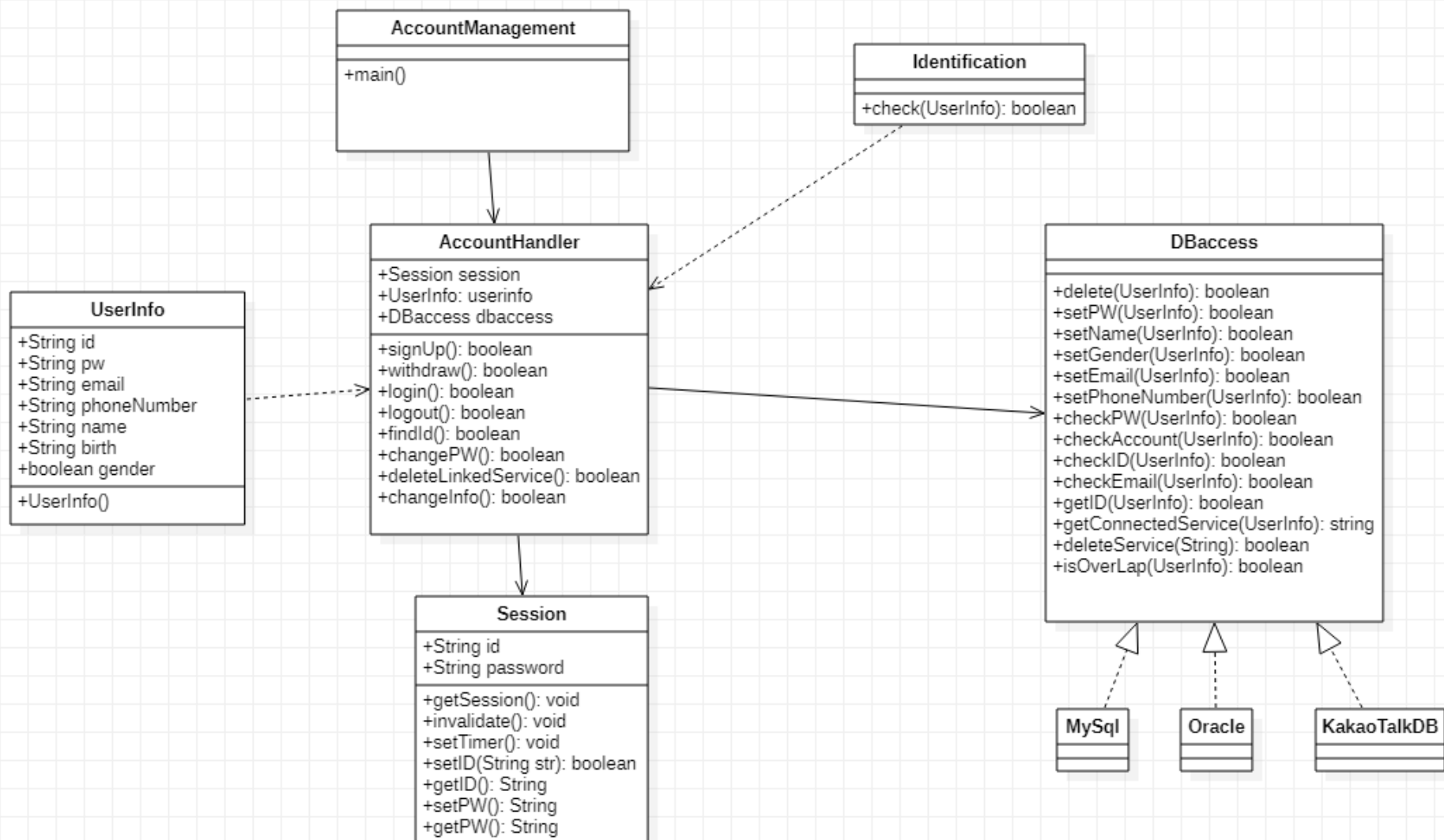
# 유스케이스 명세서

유스케이스 명	U10. 회원확인	유스케이스 개요	사용자가 입력한 계정 정보가 카카오톡 DB 시스템에 존재하는지 판단하는 유스케이스이다.
액터명	사용자, 카카오톡 DB 시스템	사전 조건	사용자는 회원으로 가입되어 있어야 한다.
정상 흐름	1. 사용자는 입력한 정보를 받아온다. 2. 사용자는 입력한 정보가 카카오톡 DB 시스템에 존재하는지 확인한다. 3. 카카오톡 DB 시스템은 계정이 존재하는 것을 알려준다.		
선택 흐름	✓ 계정이 존재하지 않으면 없다고 알려준다.		

# 유스케이스 명세서

유스케이스 명	U11. 본인인증	유스케이스 개요	사용자가 입력한 정보가 본인인지 외부 인증 시스템에서 판단하는 유스케이스이다.
액터명	사용자, 외부 인증 시스템	사전 조건	사용자는 개인 정보가 입력되어 있어야 한다.
정상 흐름	1. 사용자는 입력한 개인정보(이름, 주민등록번호, 전화번호)를 받아온다. 2. 사용자는 입력한 정보가 본인인지 외부 인증 시스템을 통해 확인한다. 3. 입력한 정보가 본인이면 본인 인증에 성공한다.		
선택 흐름	✓ 입력한 정보가 본인이 아닐 경우 본인 인증에 실패한다.		

## 클래스 다이어그램



# C1. AccountManagement

AccountManagement
+main()

클래스 명	AccountManagement	
클래스 설명	필드변수	
	메소드	+main() : 메인 * 메인 메소드 내 AccountHandler 생성자 및 switch-case를 위한 필드 변수 선언



## C2. AccountHandler

AccountHandler
+Session session +DBAccess dbaccess
+signUp(): boolean +withdraw(): boolean +login(): boolean +logout(): boolean +findID(): boolean +changePW(): boolean +changeInfo(): boolean +deleteLinkedService(): boolean

클래스 명	AccountHandler	
클래스 설명	필드변수	+Session session : 세션 생성자 +DBAccess dbaccess : DB접근 생성자
	메소드	+signUp() : 회원 가입 +withdraw() : 탈퇴 +login() : 로그인 +logout() : 로그아웃 +findID() : 아이디 찾기 +changePW() : 비밀번호 변경 +changeInfo() : 정보 변경 +deleteLinkedService() : 연결 서비스 삭제  * 모든 메소드 내 UserInfo 생성자 선언

### C3. UserInfo

UserInfo
+String id +String pw +String email +String phoneNumber +String name +String birth +boolean gender
+UserInfo()

클래스 명	UserInfo	
클래스 설명	필드변수	+String id : 아이디 +String pw : 비밀번호 +String email : 이메일 +String phoneNumber : 전화번호 +String name : 이름 +String birth : 생년월일 +boolean gender : 성별
	생성자	+UserInfo() : 생성자 * <b>UserInfo()</b> 는 모든 유저 정보를 null로 초기화

# C4. Session

Session
-String ID -String PW -int timer
+getSession(): Session +invalidate(): void +setTimer(): void +setID(String str): boolean +getID(): String +setPW(String str): boolean +getPW(): String

클래스 명	Session	
클래스 설명	필드 변수	-String ID : 아이디 -String PW: 비밀번호 -int timer: 타이머 시간(10분)
	메소드	+getSession(): 세션과 연결 +invalidate(): 세션과 연결 해제 +setTimer(): 세션 타이머를 실행 +setID(String str): 세션에 아이디를 저장 +getID(): 세션에서 아이디를 가져옴 +setPW(String str): 세션에 비밀번호를 저장 +getPW(): 세션에서 비밀번호를 가져옴

## C5. DBAccess

DBAccess
-Connection conn
+delete(UserInfo): boolean +setPW(UserInfo): boolean +setName(UserInfo) : boolean +setGender(UserInfo) : boolean +setEmail(UserInfo) : boolean +setPhoneNumber(UserInfo) : boolean +checkPW(UserInfo): boolean +checkAccount(UserInfo):boolean +checkID(UserInfo): boolean +checkEmail(UserInfo) : boolean +getID(UserInfo): String +getLinkedService(UserInfo): String +deleteService(String): boolean +isOverLap(UserInfo): boolean

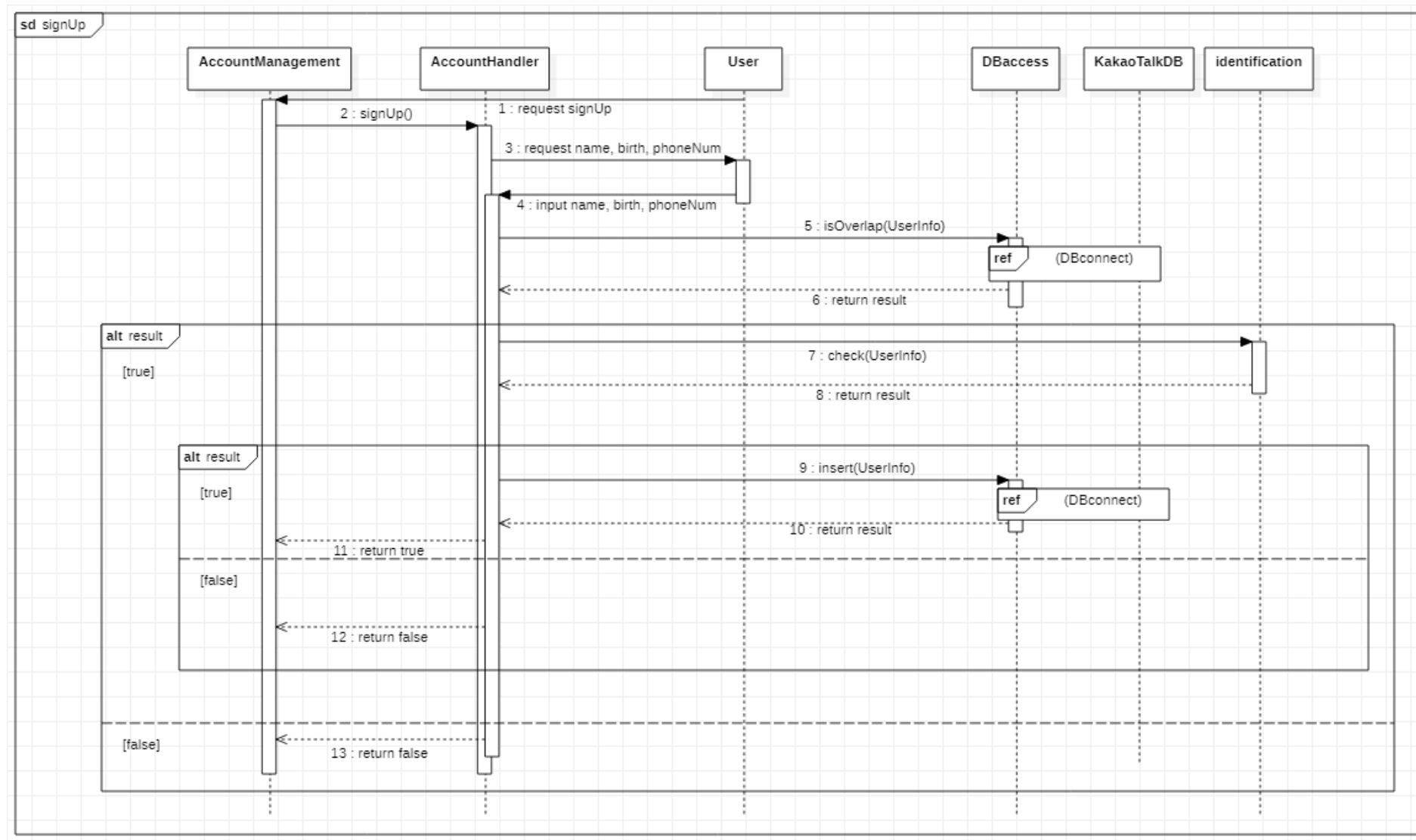
클래스 명	DBAccess	
클래스 설명	필드변수	-Connection conn: 데이터베이스 접근을 위한 변수
	메소드	+delete(UserInfo) : 유저정보를 DB에서 삭제 +setPW(UserInfo) : 사용자의 비밀번호를 변경 +setName(UserInfo) : 사용자의 이름 변경 +setGender(UserInfo) : 사용자의 성별 변경 +setEmail(UserInfo) : 사용자의 이메일 변경 +setPhoneNumber(UserInfo) : 사용자의 전화번호 변경 +checkPW(UserInfo) : 유저 정보와 비밀번호가 일치하는지 확인 +checkAccount(UserInfo) : 아이디와 비밀번호가 일치하는지 확인 +checkID(UserInfo): 아이디가 존재하는지 확인 +checkEmail(UserInfo) : 이름, 이메일이 일치하는지 확인 +getID(UserInfo) : 유저정보에 해당하는 계정의 ID를 반환 +getLinkedService(UserInfo): 유저정보에 연결된 서비스들을 반환 +deleteService(String): 사용자가 선택한 서비스를 DB에서 삭제 +isOverLap(UserInfo): 사용자가 입력한 계정 정보가 중복되는지 확인  * <b>UserInfo 객체를 매개변수로 받을 때 null값은 사용하지 않음</b>

# C6. Identification

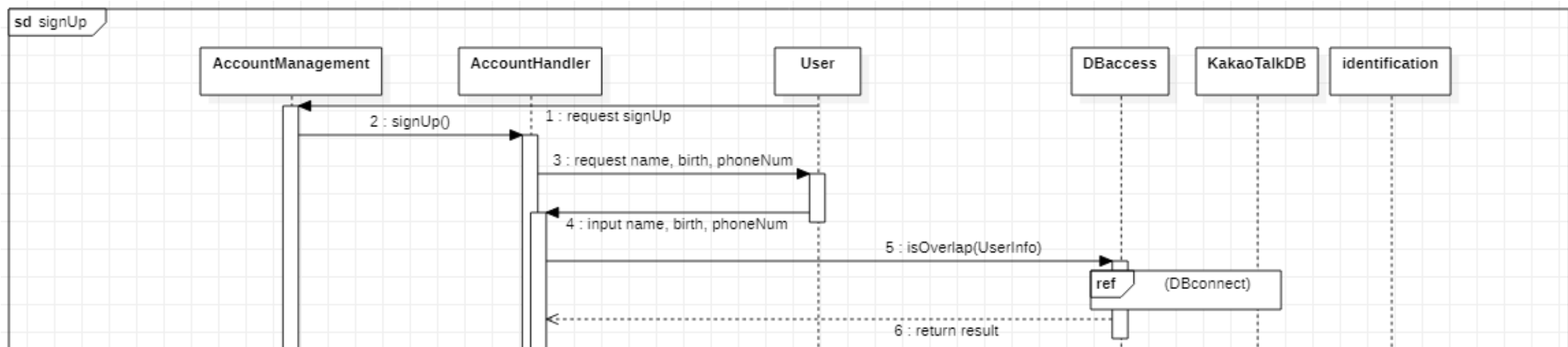
Identification
+check(UserInfo): boolean

클래스 명	Identification	
클래스 설명	필드변수	
	메소드	+check(UserInfo) : 사용자의 정보를 받아 외부 인증 시스템을 이용해 본인이 맞는지 확인. * <b>UserInfo 객체를 매개변수로 받을 때 null값은 사용하지 않음</b>

# S1. SignUp

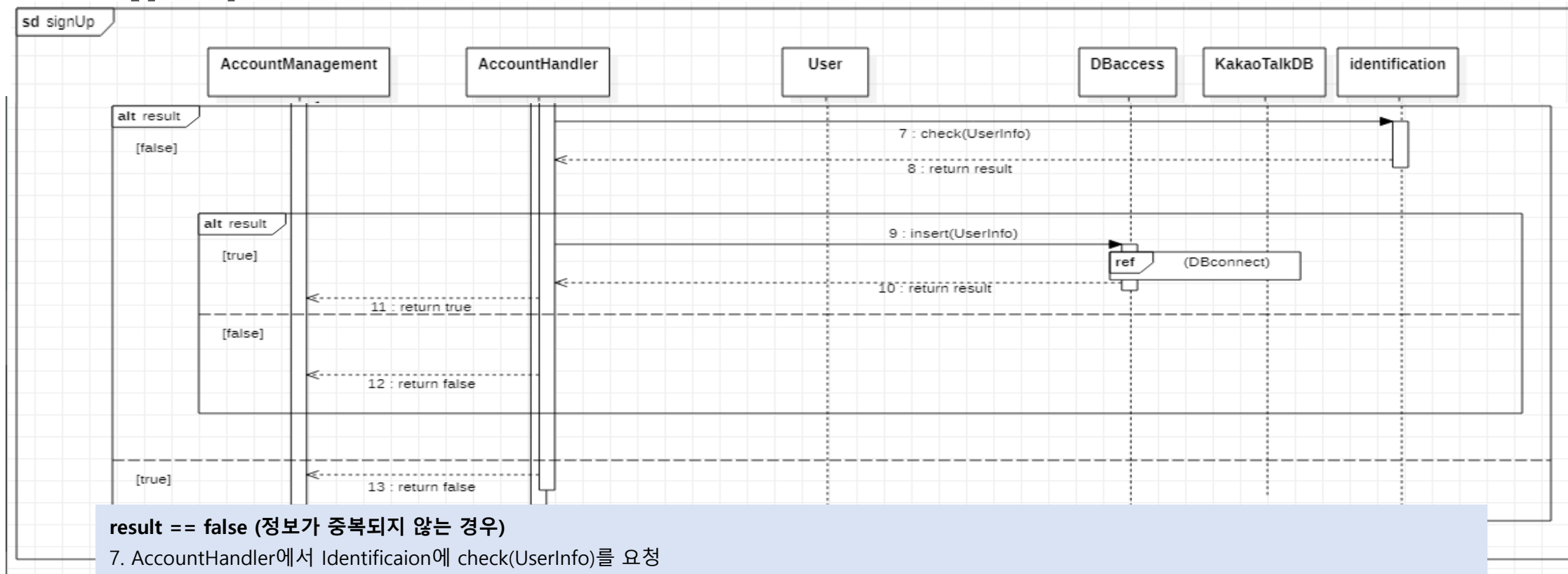


## S1. SignUp



1. 사용자가 회원가입을 요청
2. AccountManagement가 AccountHandler에 signUp()을 요청
3. AccountHandler가 사용자에게 회원가입을 위한 사용자정보를 요청
4. 사용자가 사용자정보를 입력(UserInfo구조체에 저장)
5. AccountHandler가 DBaccess에 isOverlap(UserInfo)를 요청
6. 결과 반환

# S1. SignUp



**result == false (정보가 중복되지 않는 경우)**

7. AccountHandler에서 Identification에 check(UserInfo)를 요청

8. 결과를 반환

**result == true ( 가입 가능한 정보인 경우)**

9. AccountHandler에서 DBaccess에 Insert(UserInfo)를 요청

10. 결과를 반환

11. 회원가입 기능 true 리턴

**result == false ( 가입 불가능한 정보인 경우)**

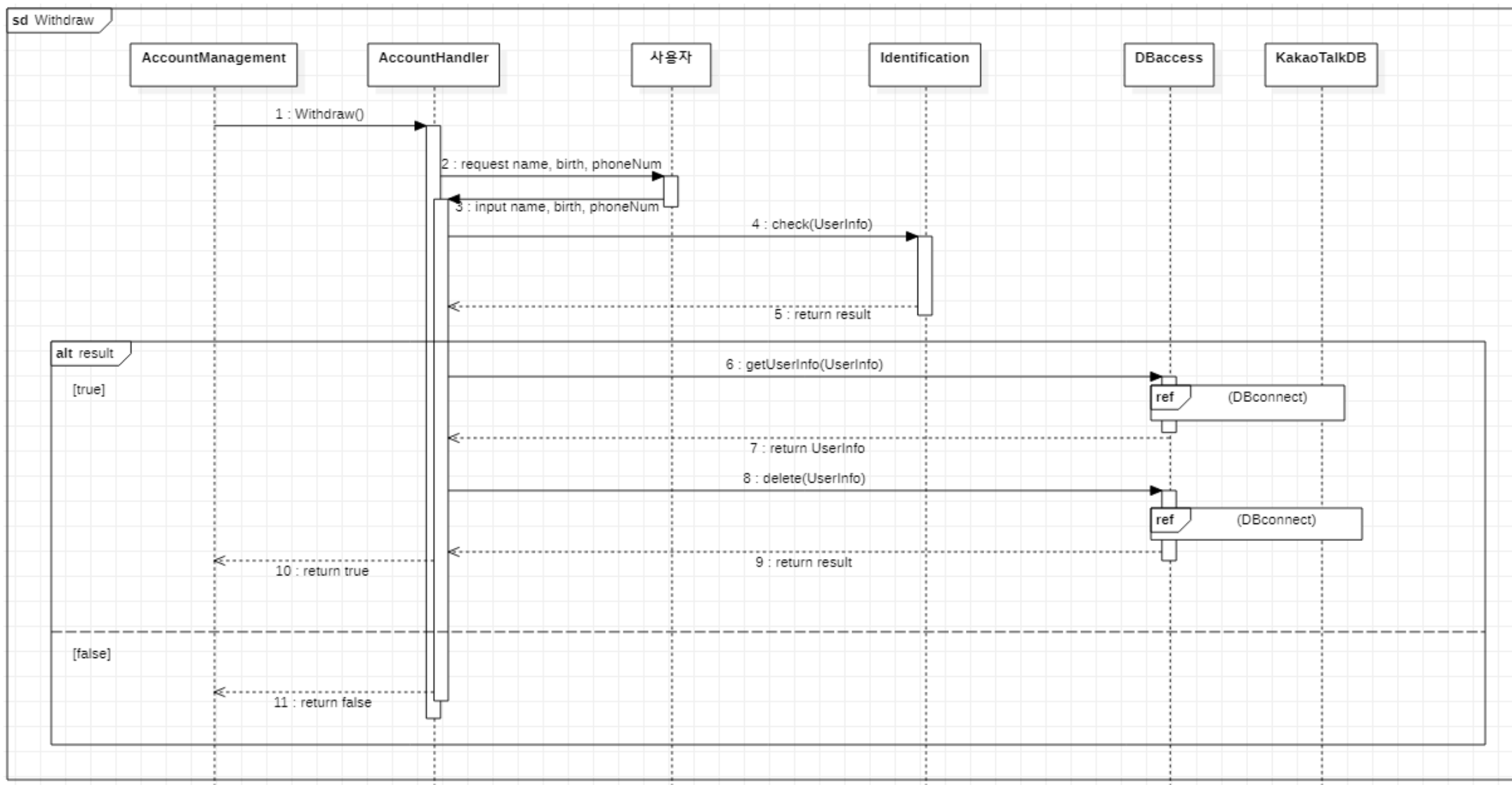
12. 회원가입 기능 false 리턴

**result == true (정보가 중복되는 경우)**

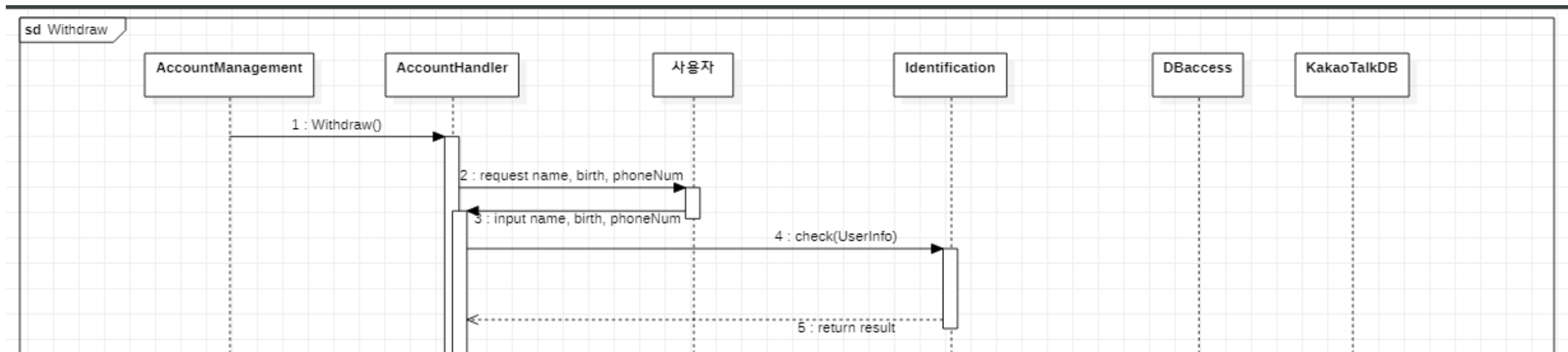
13. 회원가입 기능 false 리턴



## S2. Withdraw

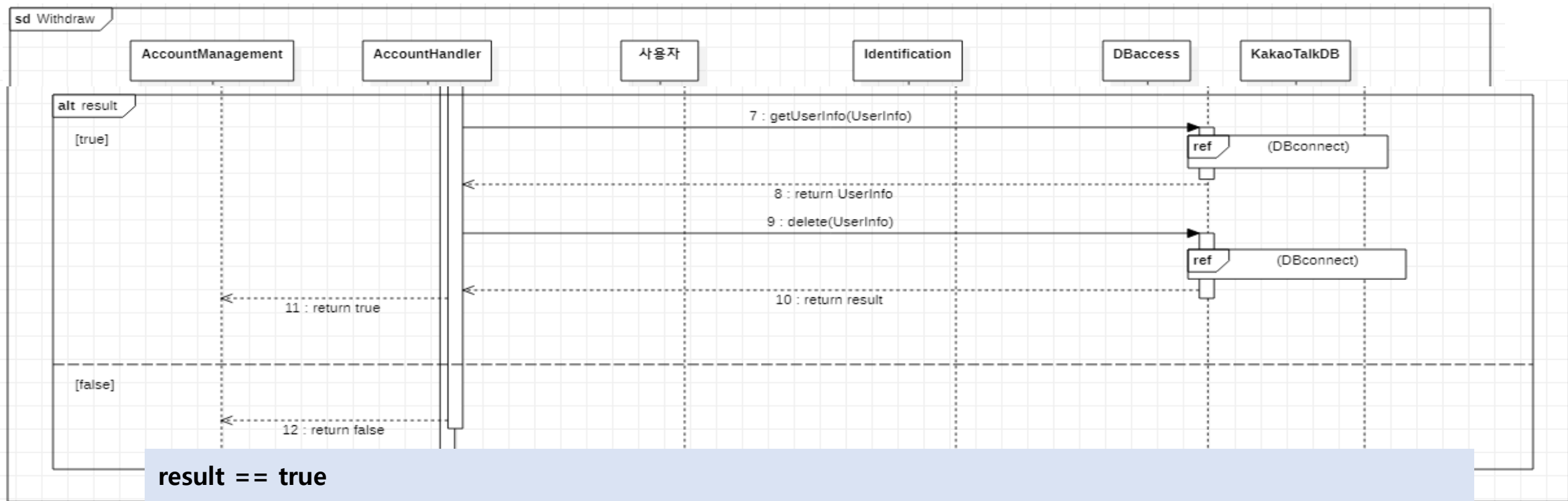


## S2. Withdraw



1. 사용자가 탈퇴를 요청
2. AccountManagement에서 AccountHandler로 Withdraw()을 요청
3. 사용자에게 본인인증을 위한 정보(이름, 주민번호, 전화번호)를 요청
4. 사용자가 정보를 입력(UserInfo 구조체에 저장)
5. AccountHandler에서 Identification에 check(UserInfo)를 요청
6. 확인 결과 반환

## S2. Withdraw



**result == true**

7. AccountHandler에서 DBaccess에 getUserInfo(UserInfo)를 요청

8. 본인인증 정보를 토대로 사용자의 모든 UserInfo 반환

9. AccountHandler에서 DBaccess에 getUserInfo(UserInfo)를 요청

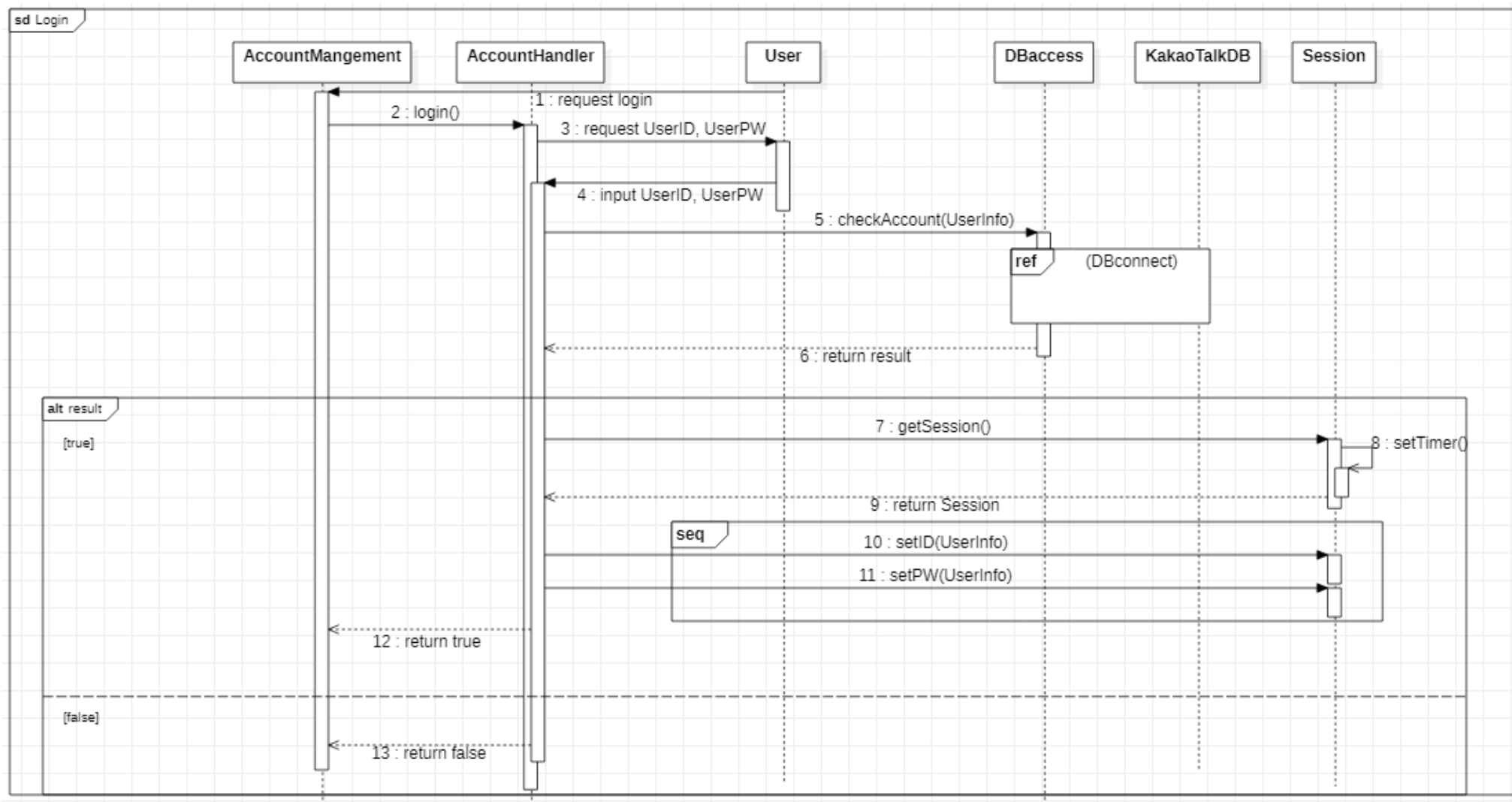
10. 결과를 반환

11. 탈퇴기능 true 리턴

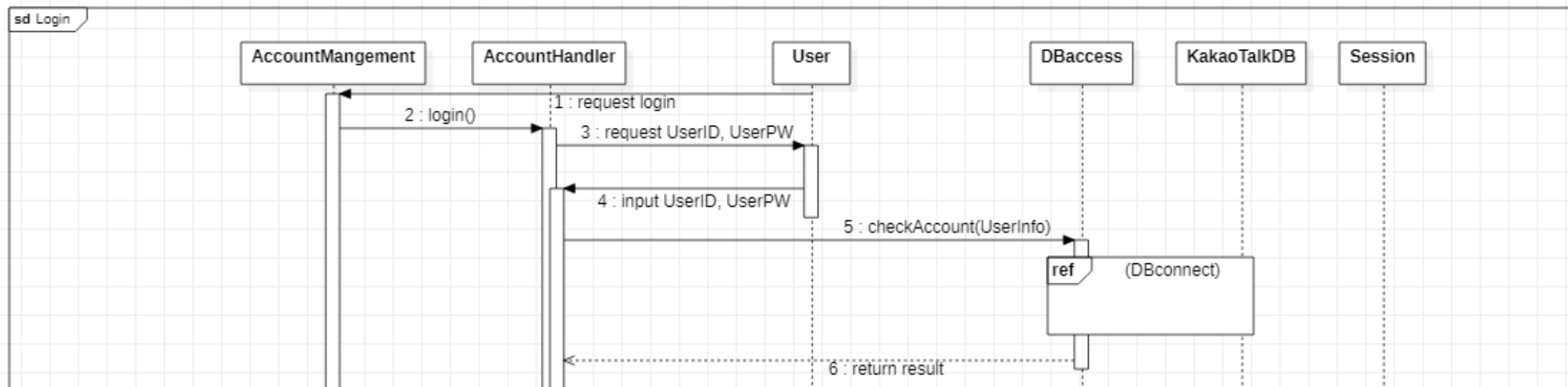
**result == false**

12. 탈퇴기능 false 리턴

## S3. Login

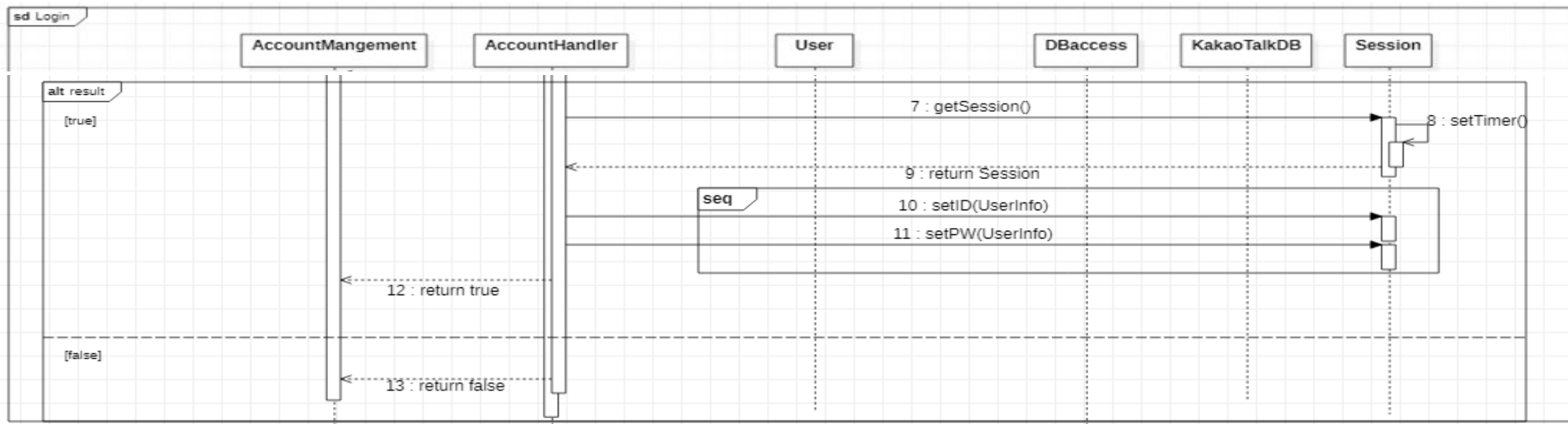


## S3. Login



1. 사용자가 로그인을 요청
2. AccountManagement에서 AccountHandler로 login()을 요청
3. 사용자에게 UserID와 UserPW를 요청
4. 사용자가 UserID와 UserPW를 입력(UserInfo 구조체에 저장)
5. AccountHandler에서 DBaccess에 checkAccount(UserInfo)를 요청
6. 확인 결과 반환

## S3. Login



**if (result == true)**

7. AccountHandler에서 Session으로 getSession을 요청

8. Session에 setTimer()를 요청

9. Session을 반환

10. AccountHandler에서 Session에 setID(UserInfo)를 요청

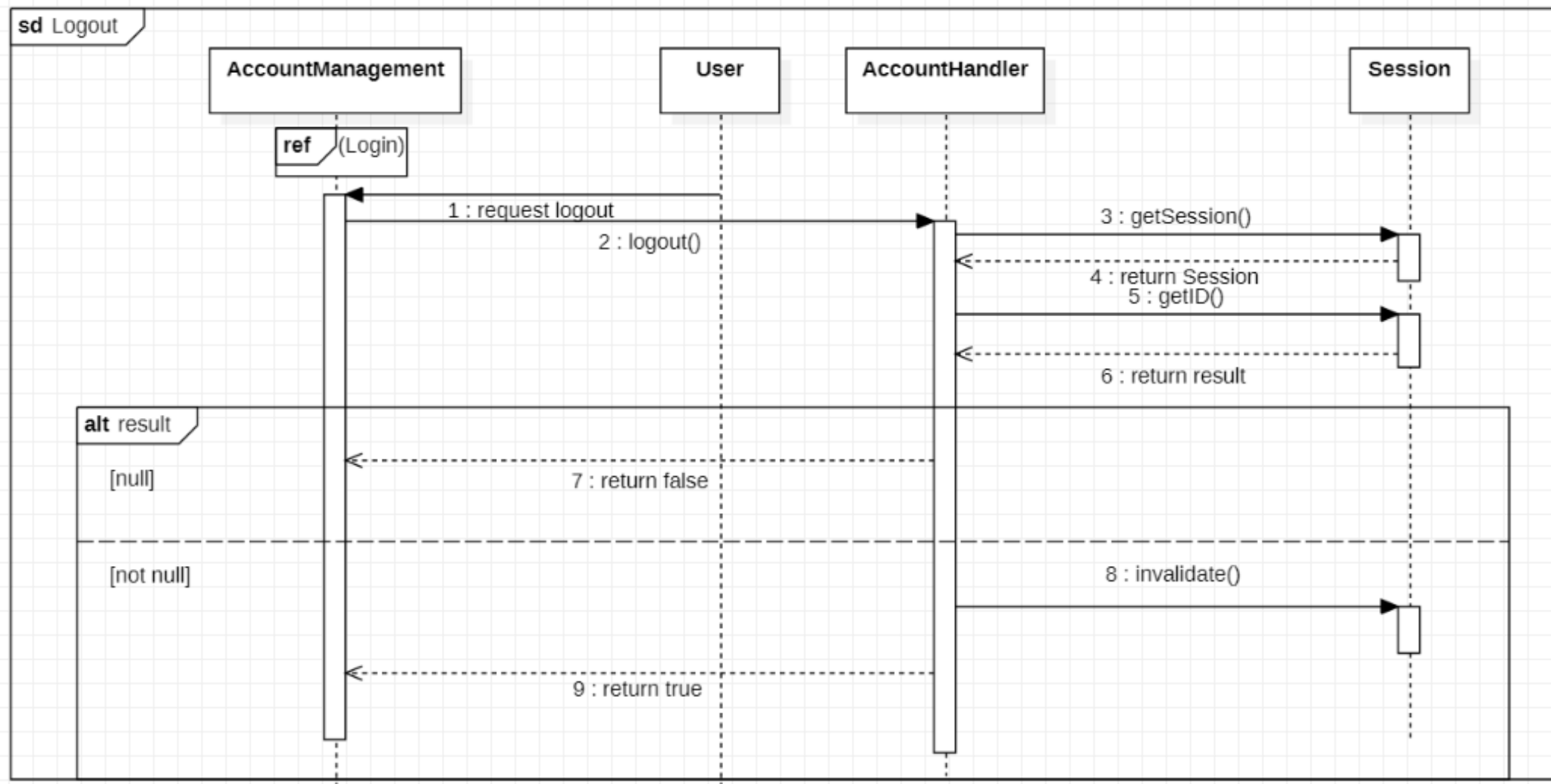
11. AccountHandler에서 Session에 setPW(UserInfo)를 요청

12. true 반환(로그인성공)

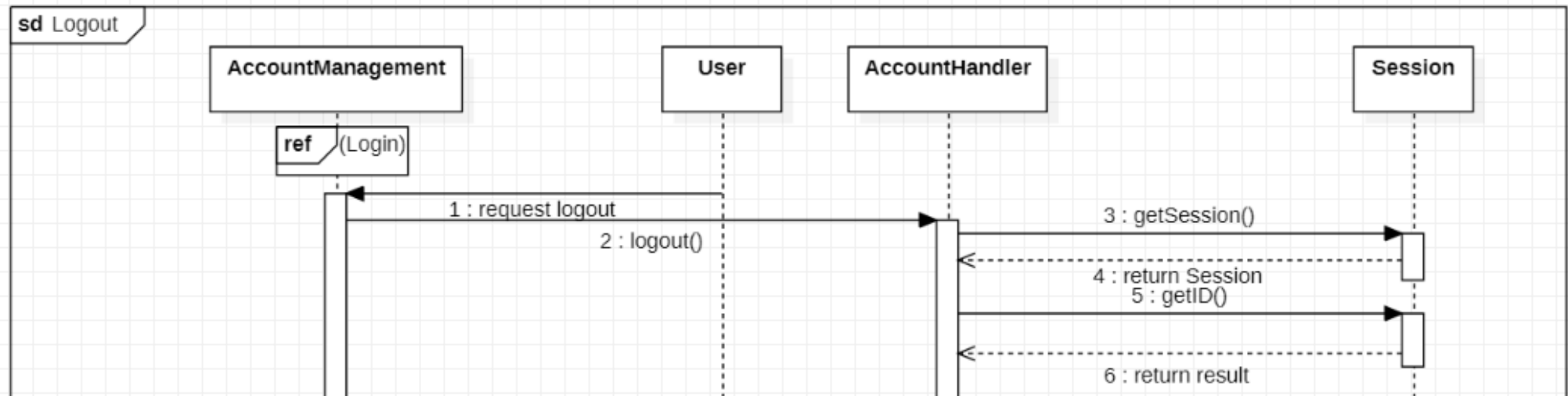
**if (result == false)**

13. false 반환(로그인실패)

## S4. Logout



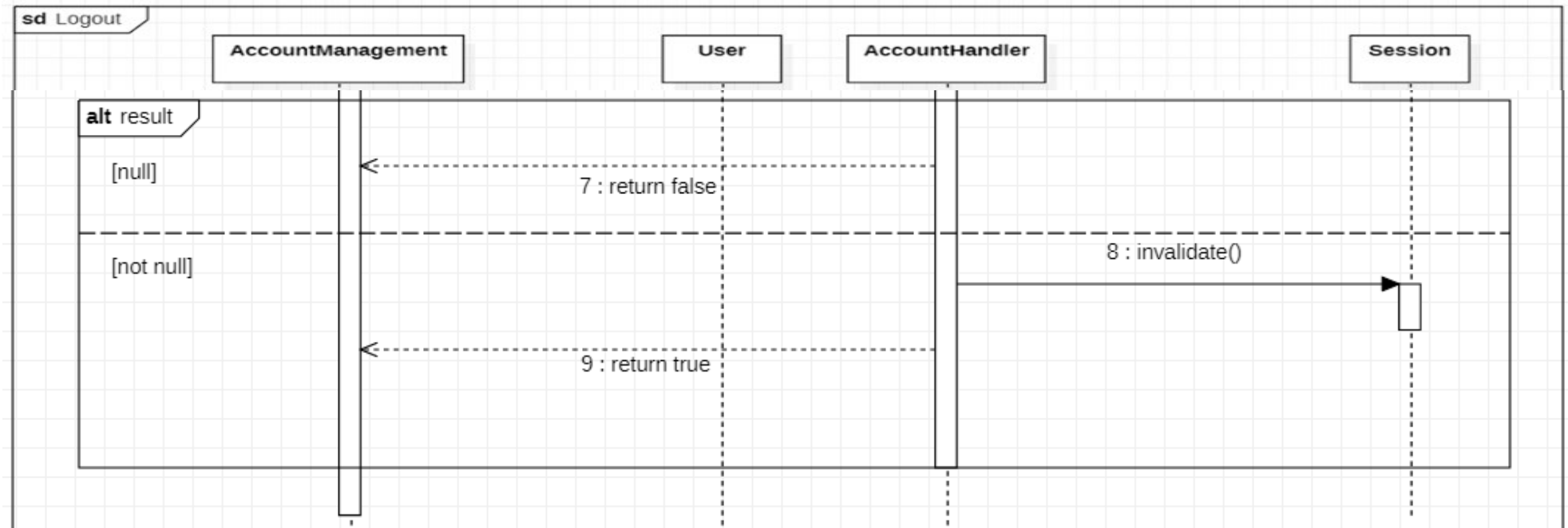
## S4. Logout



1. 사용자가 로그아웃을 요청
2. AccountManagement가 AccountHandler에 logout()을 요청
3. AccountHandler가 Session에 getSession()을 요청
4. Session 반환
5. AccountHandler가 Session에 getID()를 요청
6. 결과 반환



## S4. Logout



### If (result == null)

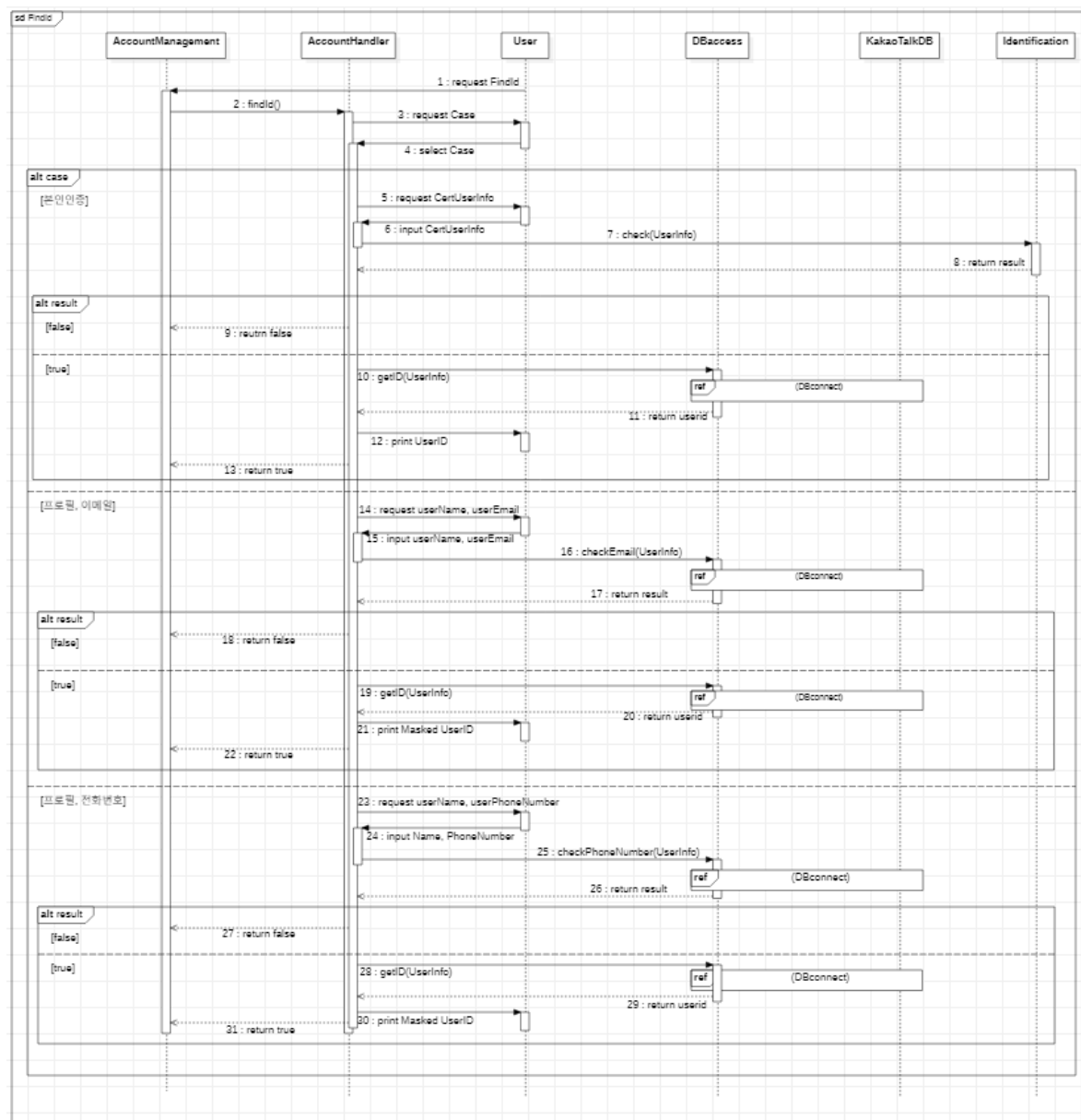
7. false 반환(세션 타임아웃으로 자동로그아웃 된 상태)

### If (result == not null)

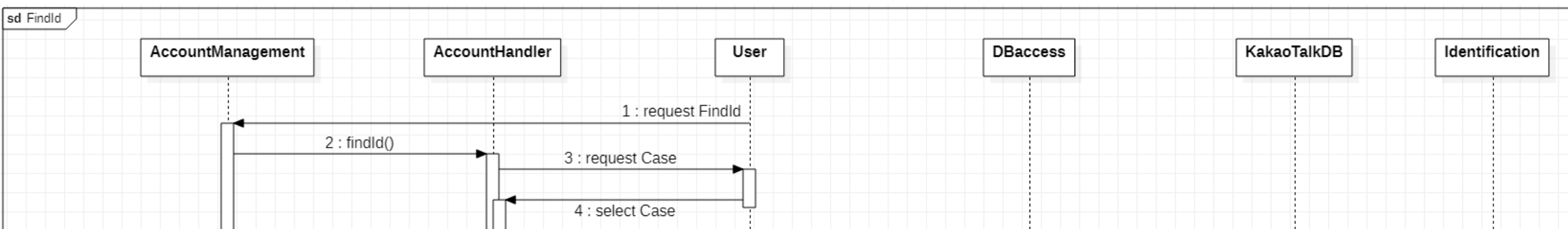
8. AccountHandler가 Session에 invalidate()를 요청

9. true 반환(로그아웃 성공)

## S5. FindID

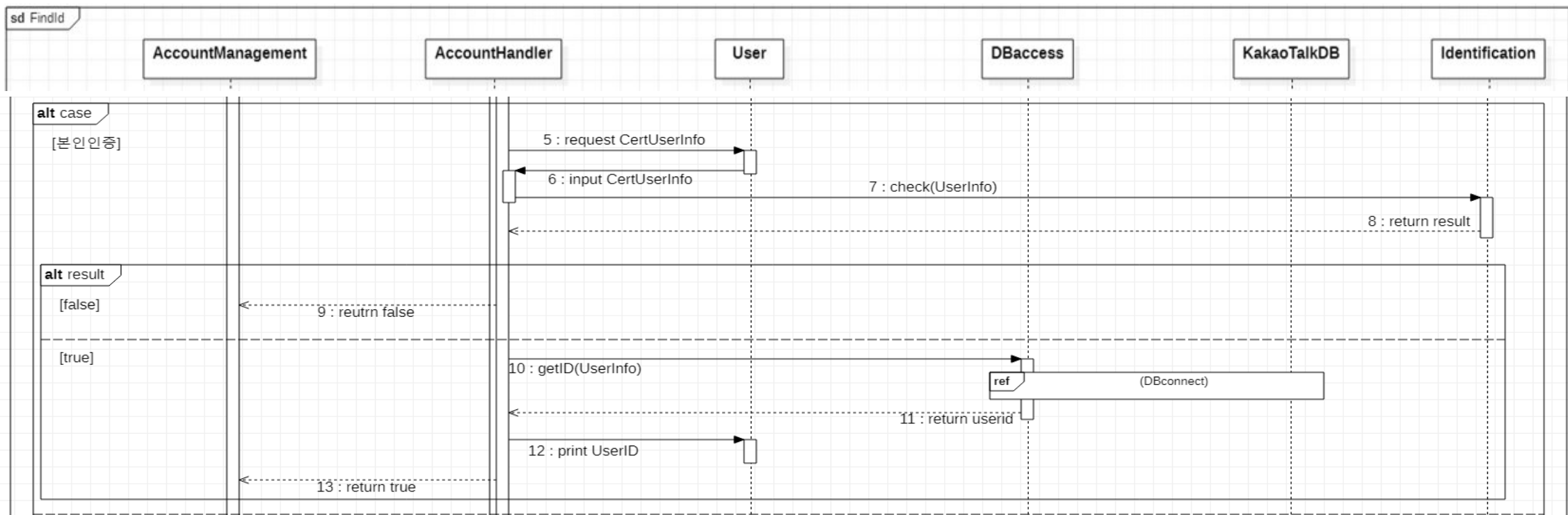


## S5. FindID



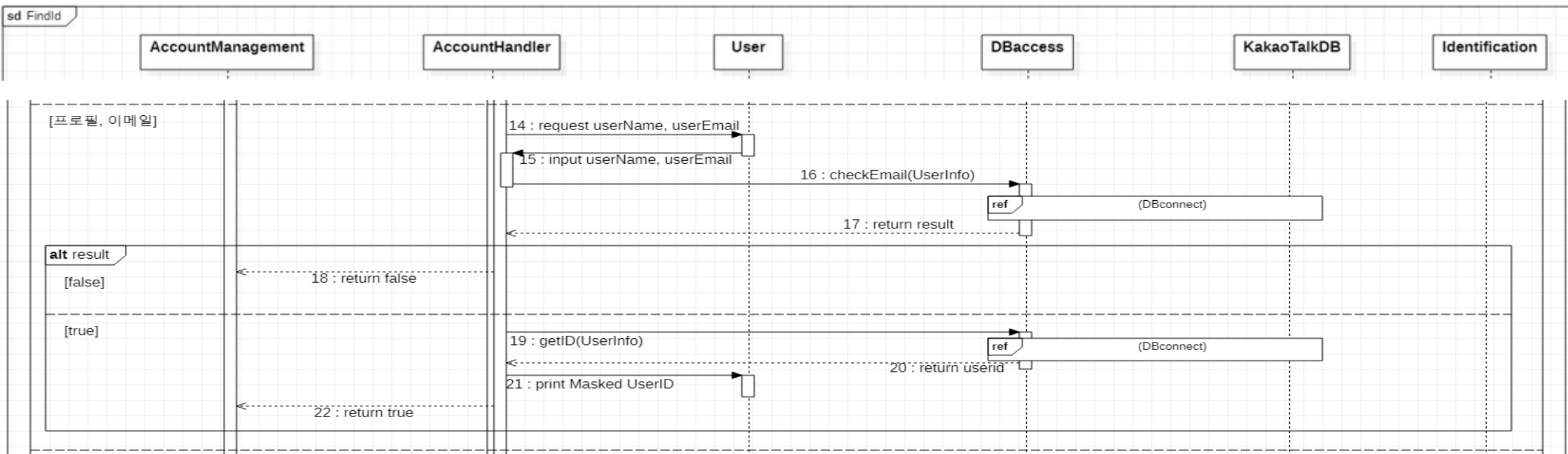
1. 사용자는 FindId 기능 요청
2. AccountManagement에서 findid() 호출
3. 사용자에게 인증방법을 선택을 요청
4. 사용자는 인증방법을 선택

## S5. FindID



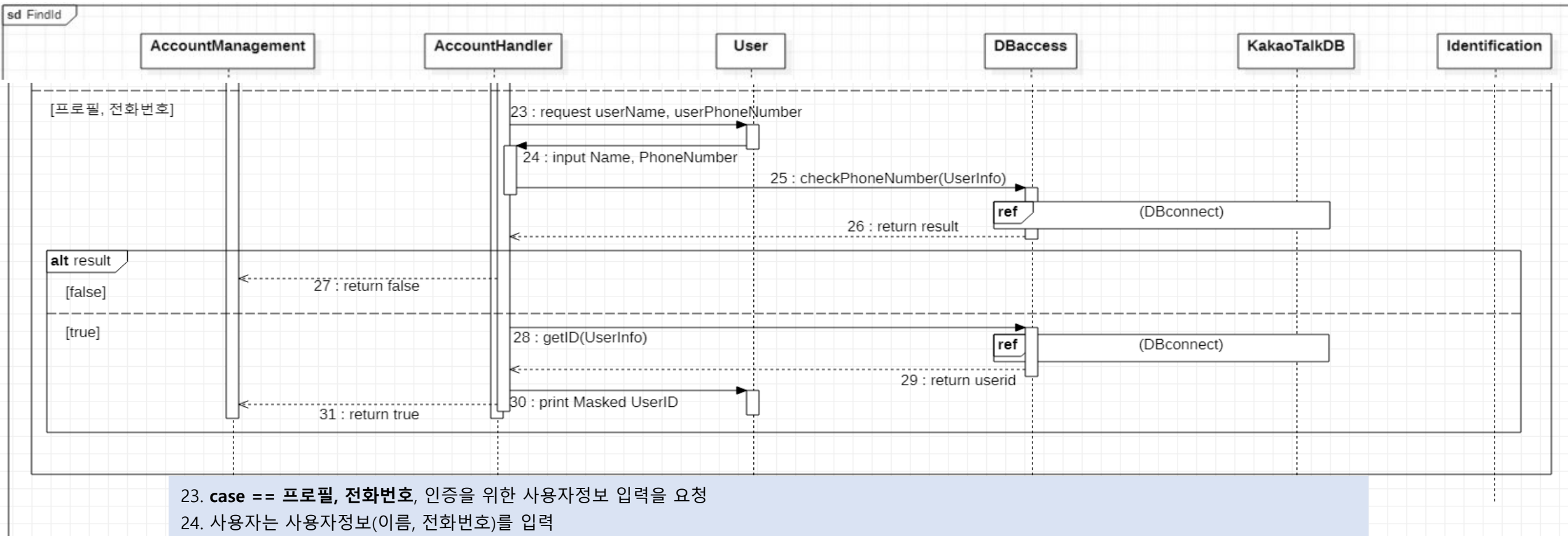
5. **case == 본인인증**, 본인인증을 위한 사용자정보 입력을 요청
6. 사용자는 사용자정보(이름, 전화번호, 주민번호)를 입력
7. 인증을 위해 check(UserInfo)를 통해 외부인증시스템에 정보를 전달
8. 인증결과 반환
9. **인증결과 == false**, 인증결과가 false일 때, 아이디찾기 기능 false 반환
10. **인증결과 == true**, getID(UserInfo)를 통해 ID가져옴
11. 사용자의 ID 반환
12. 사용자에게 ID 출력
13. 아이디찾기 기능 true 반환

## S5. FindID



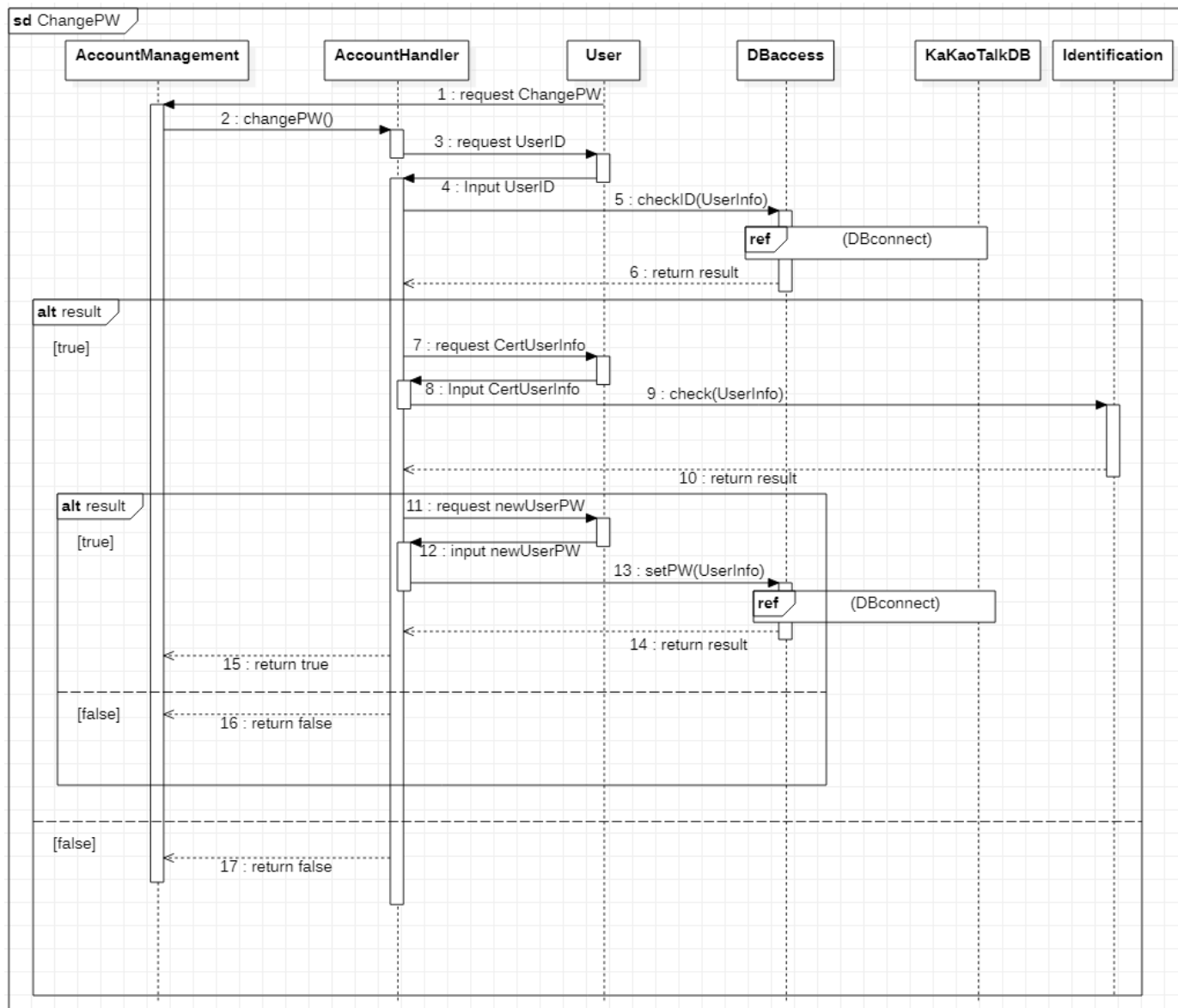
14. **case == 프로필, 이메일**, 인증을 위한 사용자정보 입력을 요청
15. 사용자는 사용자정보(이름, 이메일)를 입력
16. 인증을 위해서 checkEmail(UserInfo)를 통해 DBaccess에 정보 전달
17. 인증결과 반환
18. **인증결과 == false**, 아이디찾기 기능 false반환
19. **인증결과 == true**, getID(Userinfo)를 통해 ID가져옴
20. 사용자의 ID반환
21. 사용자에게 마스킹된 아이디 출력
22. 아이디찾기 기능 true반환

## S5. FindID

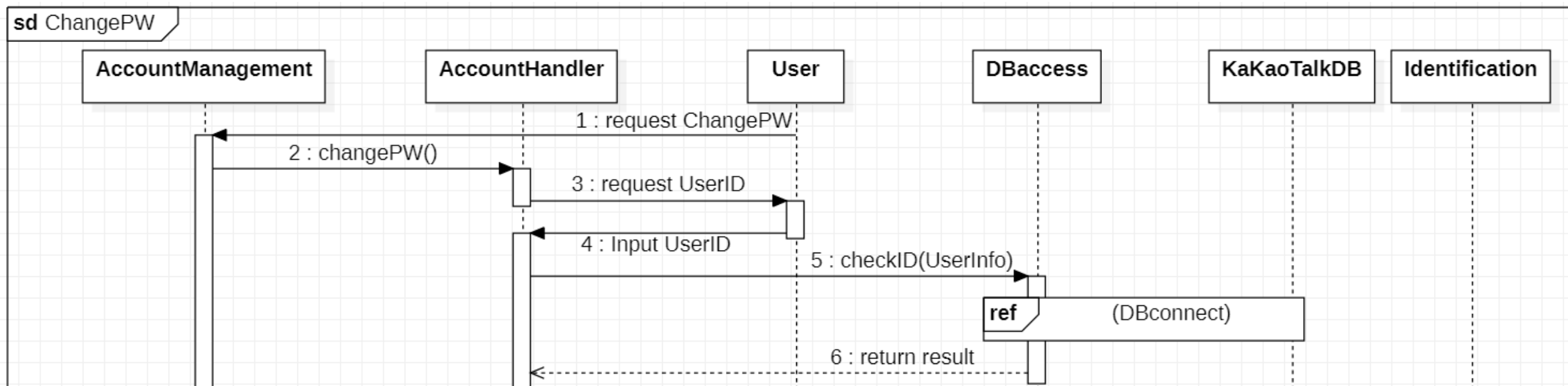


23. **case == 프로필, 전화번호**, 인증을 위한 사용자정보 입력을 요청
24. 사용자는 사용자정보(이름, 전화번호)를 입력
25. 인증을 위해서 checkPhoneNumber(UserInfo) DBaccess에 정보 전달
26. 인증결과 반환
27. **인증결과 == false**, 아이디찾기 기능 false반환
28. **인증결과 == true**, getID(UserInfo)를 통해 ID가져옴
29. 사용자의 ID반환
30. 사용자에게 마스킹된 아이디 출력
31. 아이디찾기 기능 true반환

## S6. ChangePW



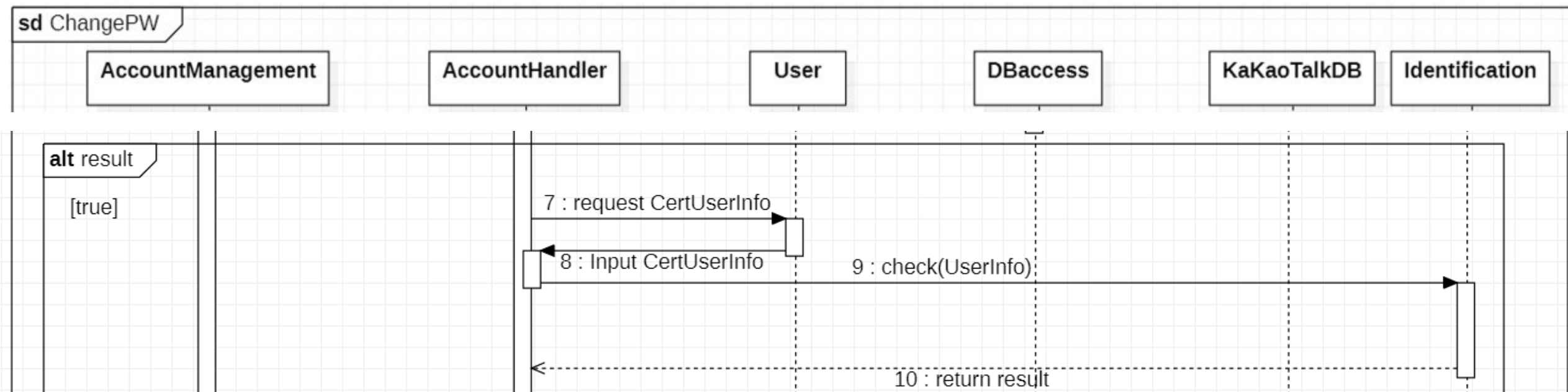
## S6. ChangePW



1. 사용자는 ChangePW 기능 요청
2. AccountManagement에서 findid() 호출
3. 사용자에게 비밀번호를 변경할 계정의 ID입력 요청
4. 사용자는 ID 입력
5. DBaccess에게 checkID(UserInfo)를 통해 정보전달
6. 인증결과 반환

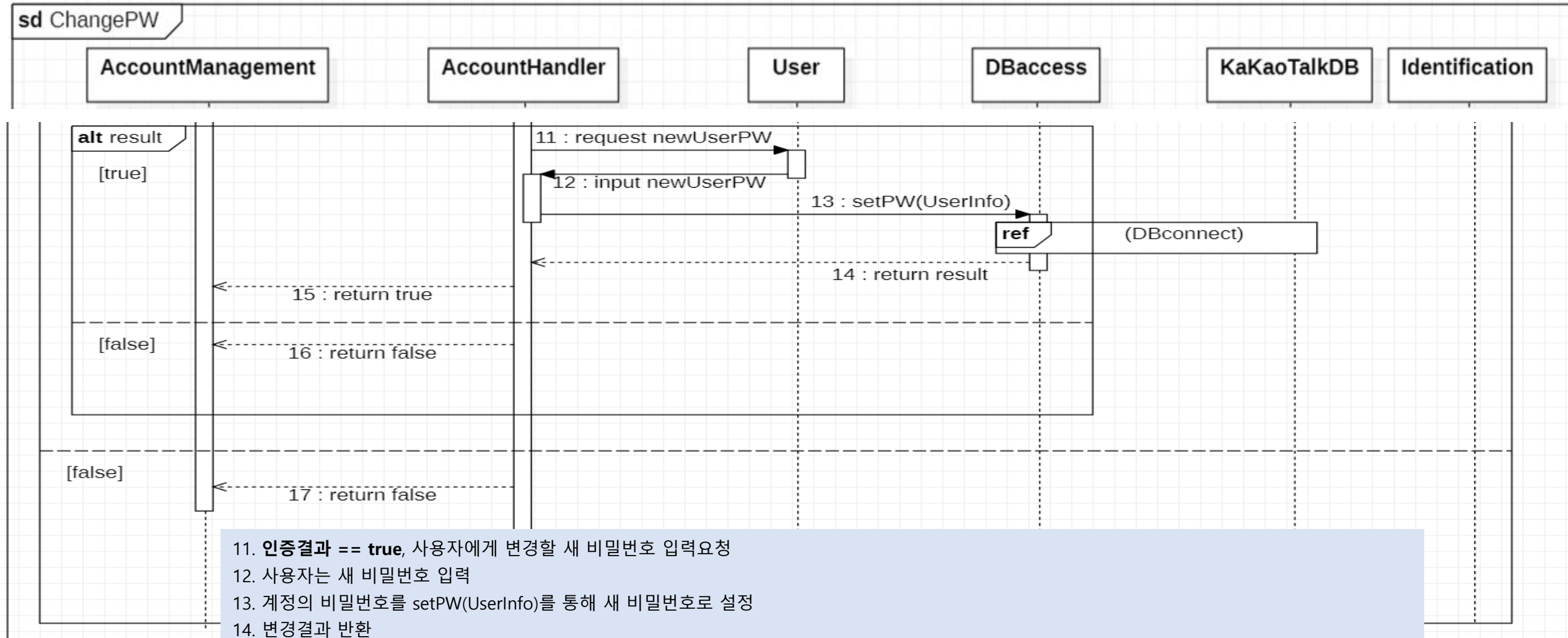


## S6. ChangePW



7. 인증결과 == true, 사용자에게 본인인증을 위한 사용자 정보입력 요청
8. 사용자는 사용자정보(이름, 전화번호, 주민번호)를 입력
9. 인증을 위해서 check(UserInfo)를 통해 외부인증시스템에 정보를 전달
10. 인증결과 반환

## S6. ChangePW



11. 인증결과 == true, 사용자에게 변경할 새 비밀번호 입력요청

12. 사용자는 새 비밀번호 입력

13. 계정의 비밀번호를 setPW(UserInfo)를 통해 새 비밀번호로 설정

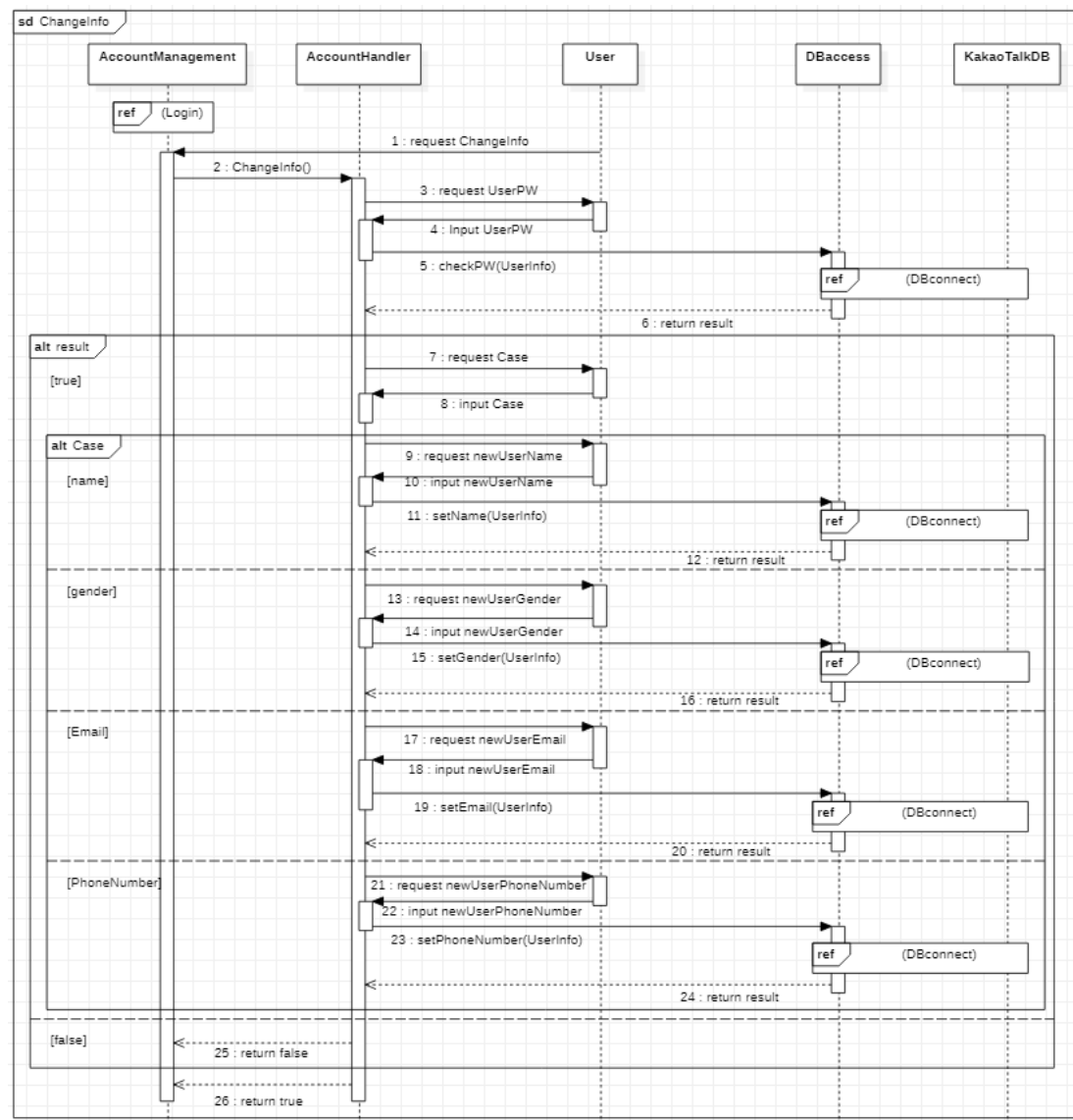
14. 변경결과 반환

15. 변경결과 == true(성공), 비밀번호변경 기능 true 반환

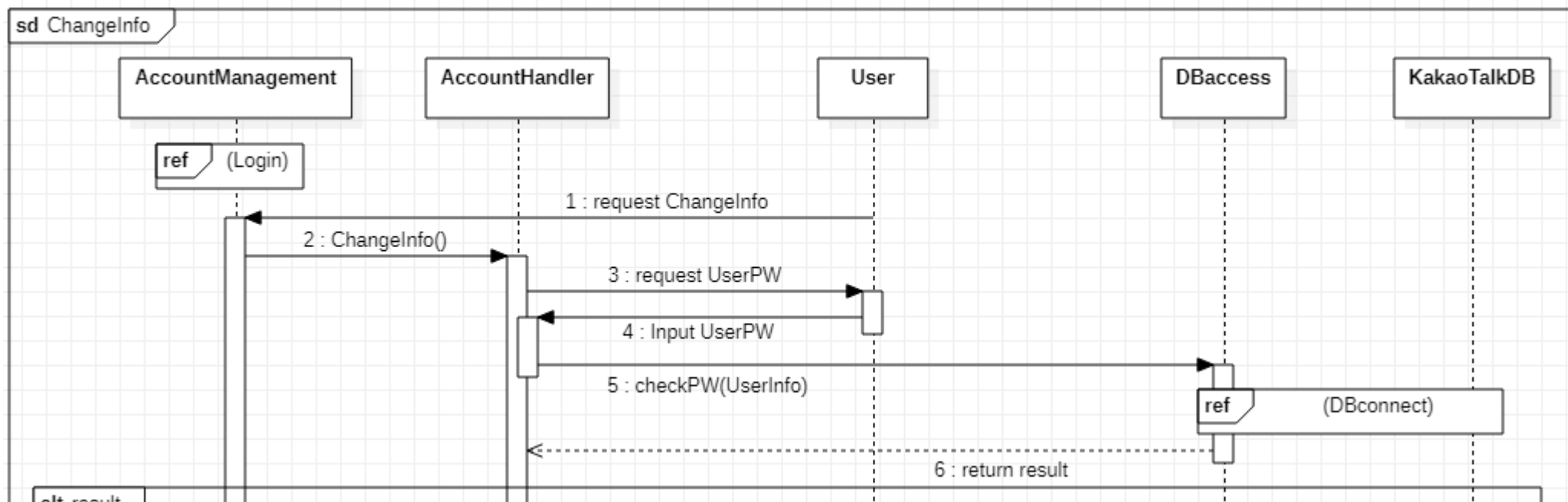
16. 변경결과 == false(실패), 비밀번호변경 기능 false 반환

17. 인증결과(6. 인증결과 리턴) == false, 비밀번호변경 기능 false 반환

## S7. ChangeInfo

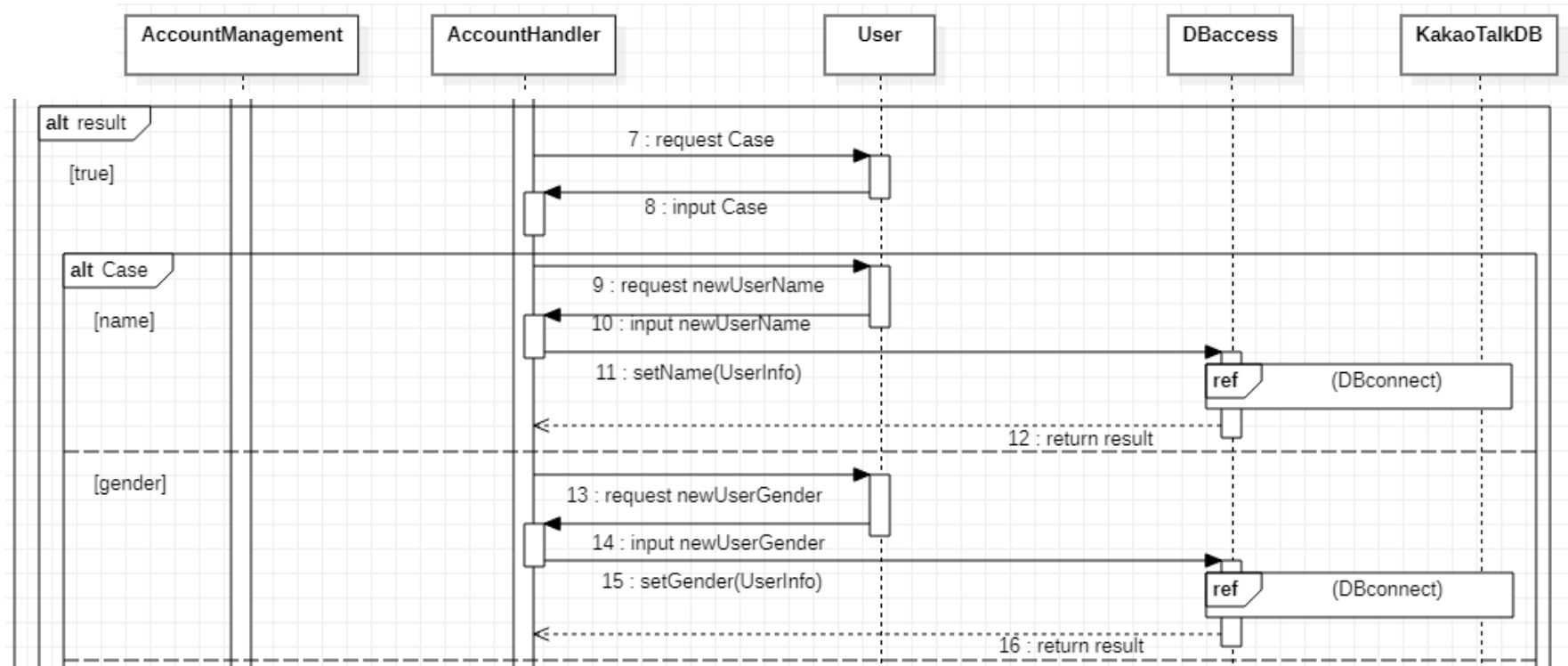


## S7. ChangelInfo



1. 사용자가 정보 변경을 요청
2. AccountManagement가 AccountHandler에게 ChangelInfo()를 요청
3. AccountHandler가 사용자에게 사용자 인증을 위한 사용자 정보(사용자 아이디에 맞는 사용자 비밀번호)를 요청
4. 사용자가 사용자 정보를 입력 (UserInfo에 저장)
5. AccountHandler가 Dbaccess에게 사용자 인증을 위해 입력한 UserInfo를 매개변수로 받아 전달
6. 결과 반환

## S7. ChangeInfo



**if (result == true) // 사용자 인증 성공 시**

7. AccountHandler는 사용자에게 어떤 정보를 변경할 것인지에 대한 선택 요청

8. 사용자는 AccountHandler에게 변경할 정보를 선택

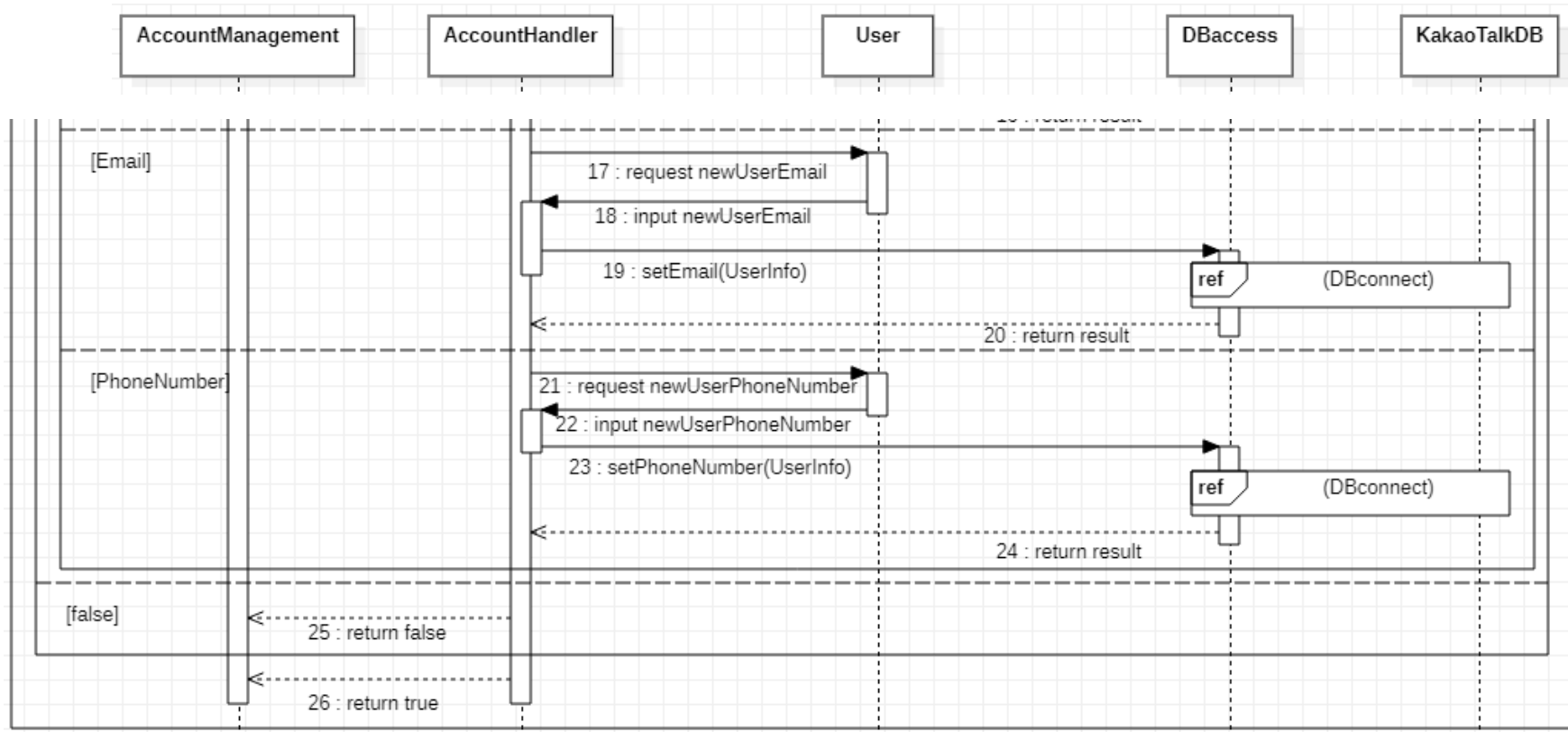
**switch(case)**

AccountHandler는 사용자에게 선택한 정보의 새로운 값 요청

사용자는 AccountHandler에게 새로운 값 입력 (UserInfo에 저장)

AccountHandler는 DBaccess에게 사용자 정보 재설정을 위해 입력한 UserInfo를 매개변수로 받아 전달  
결과 반환

## S7. ChangeInfo

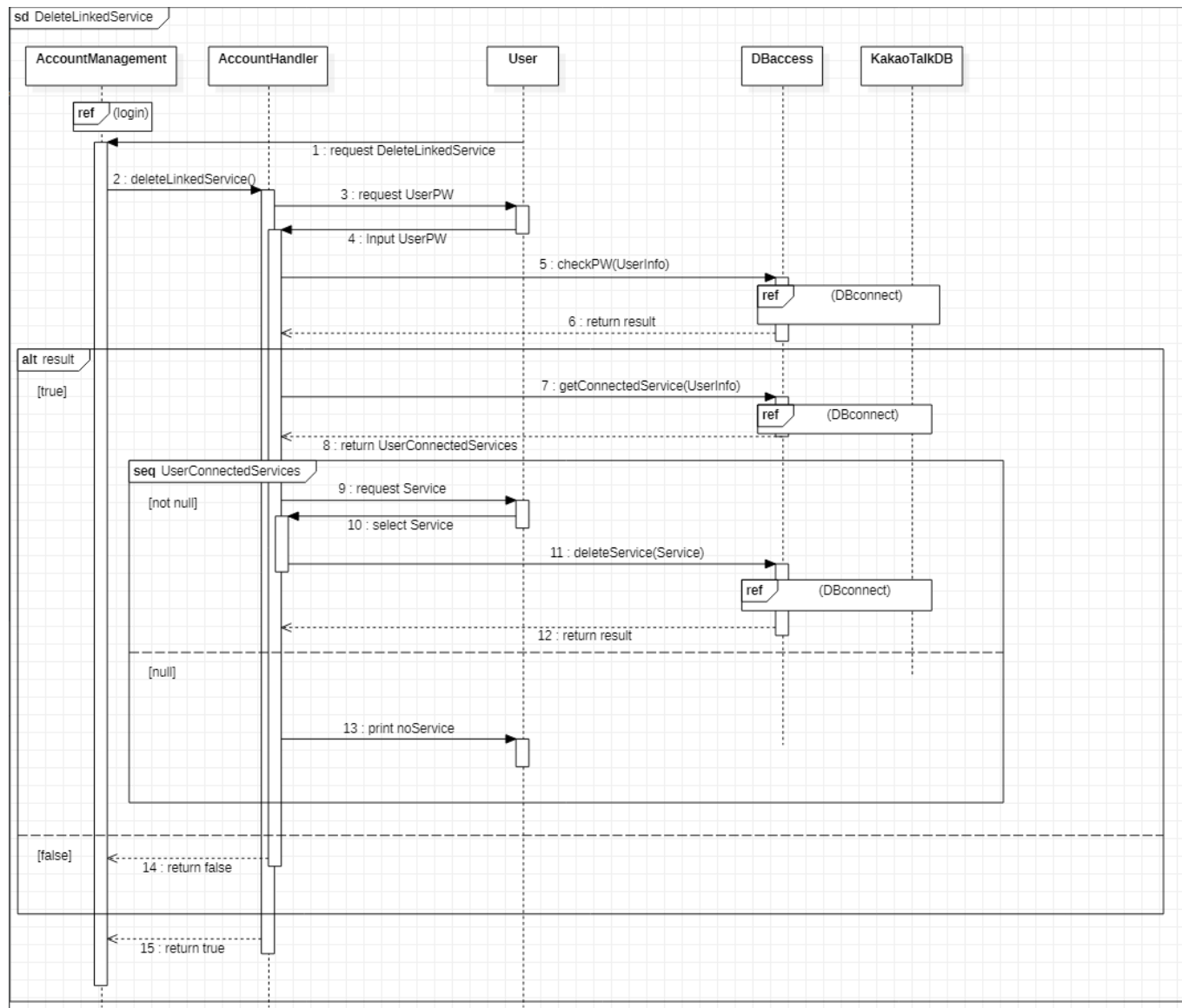


if (result == false) // 사용자 인증 실패 시

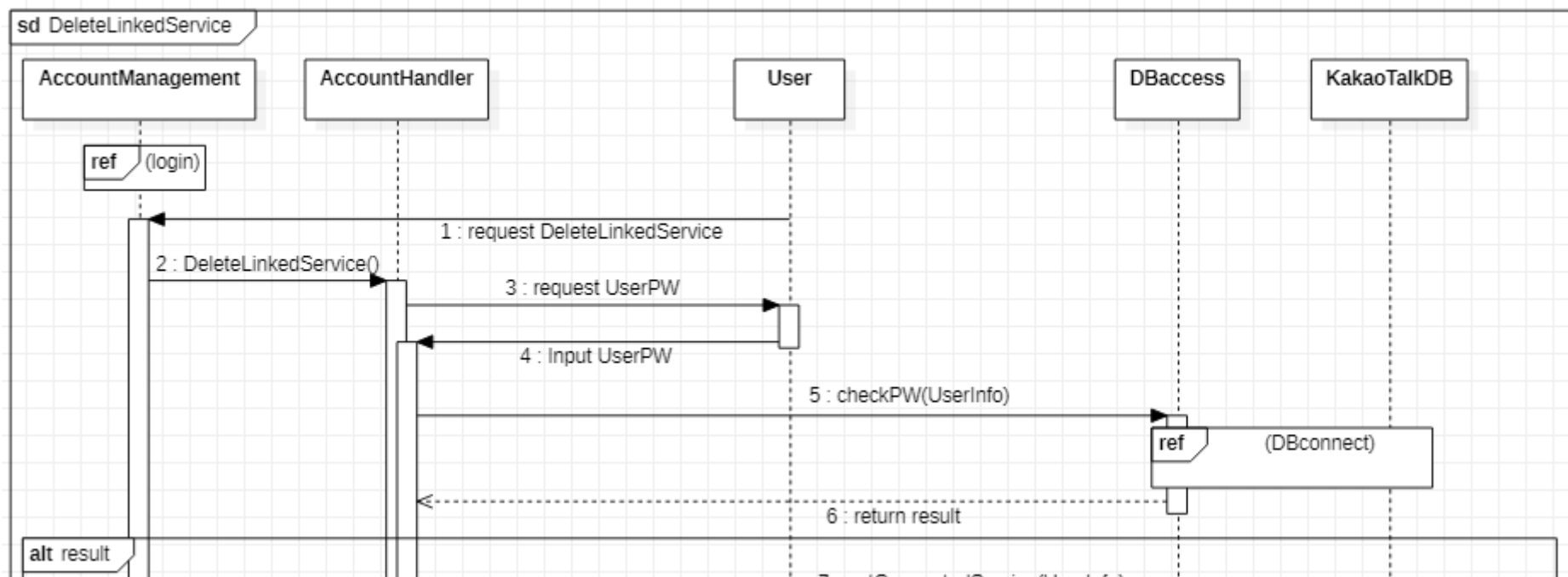
25. false 반환

26. true 반환

## S8. DeleteLinkedService



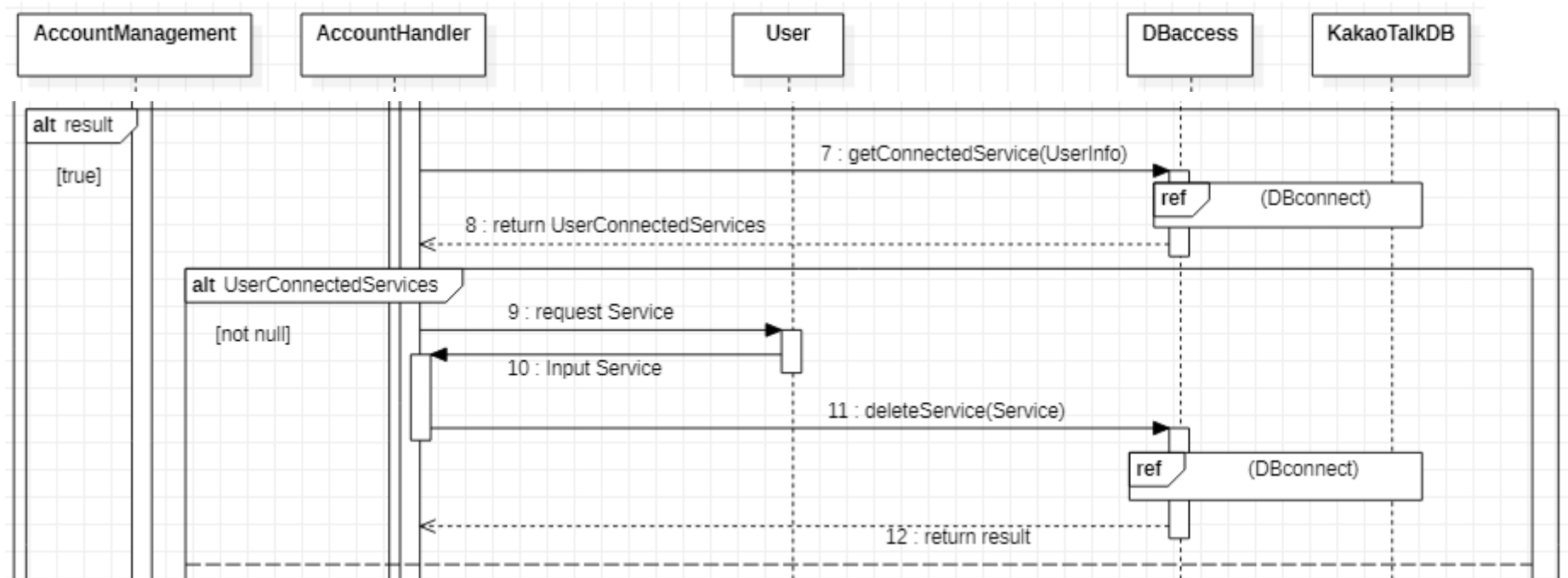
## S8. DeleteLinkedService



1. 사용자가 연결 서비스 삭제를 요청
2. AccountManagement가 AccountHandler에게 DeleteLinkedService()를 요청
3. AccountHandler가 사용자에게 사용자 인증을 위한 사용자 정보(사용자 아이디에 맞는 사용자 비밀번호)를 요청
4. 사용자가 사용자 정보를 입력 (UserInfo에 저장)
5. AccountHandler가 DBaccess에게 사용자 인증을 위해 입력한 UserInfo를 매개변수로 받아 전달
6. 결과 반환



## S8. DeleteLinkedService



**if (result == true) // 사용자 인증 성공시**

7. AccountHandler는 DBaccess에게 사용자 정보에 연결되어 있는 서비스를 얻기 위해 UserInfo를 매개변수로 받아 전달

8. 결과 반환

**if (UserConnectedServices != null) // 사용자 정보에 연결된 서비스가 있을 시**

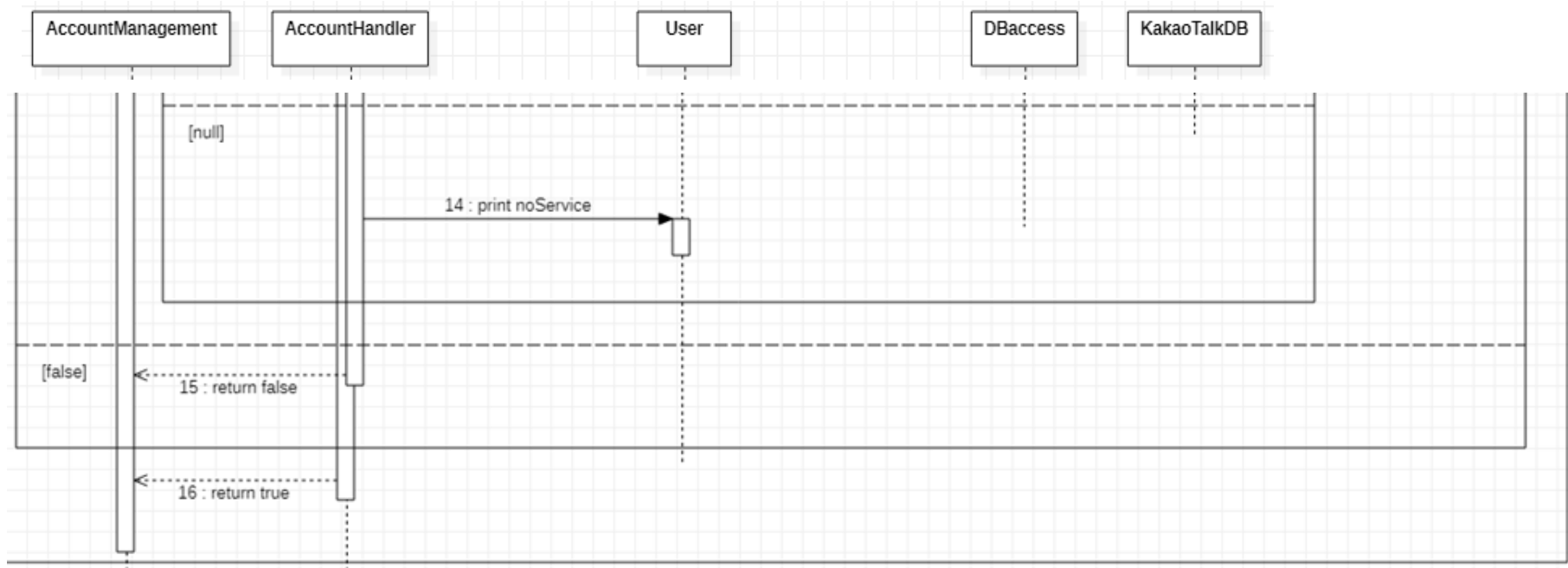
9. AccountHandler는 사용자에게 삭제할 서비스를 요청

10. 사용자는 AccountHandler에게 삭제할 서비스(Service)를 입력

11. AccountHandler는 DBaccess에게 입력한 서비스를 삭제하기 위해 입력한 Service를 매개변수로 받아 전달

12. 결과 반환

## S8. DeleteLinkedService



**else if (UserConnectedServices == null) // 사용자 정보에 연결된 서비스가 없을 시**

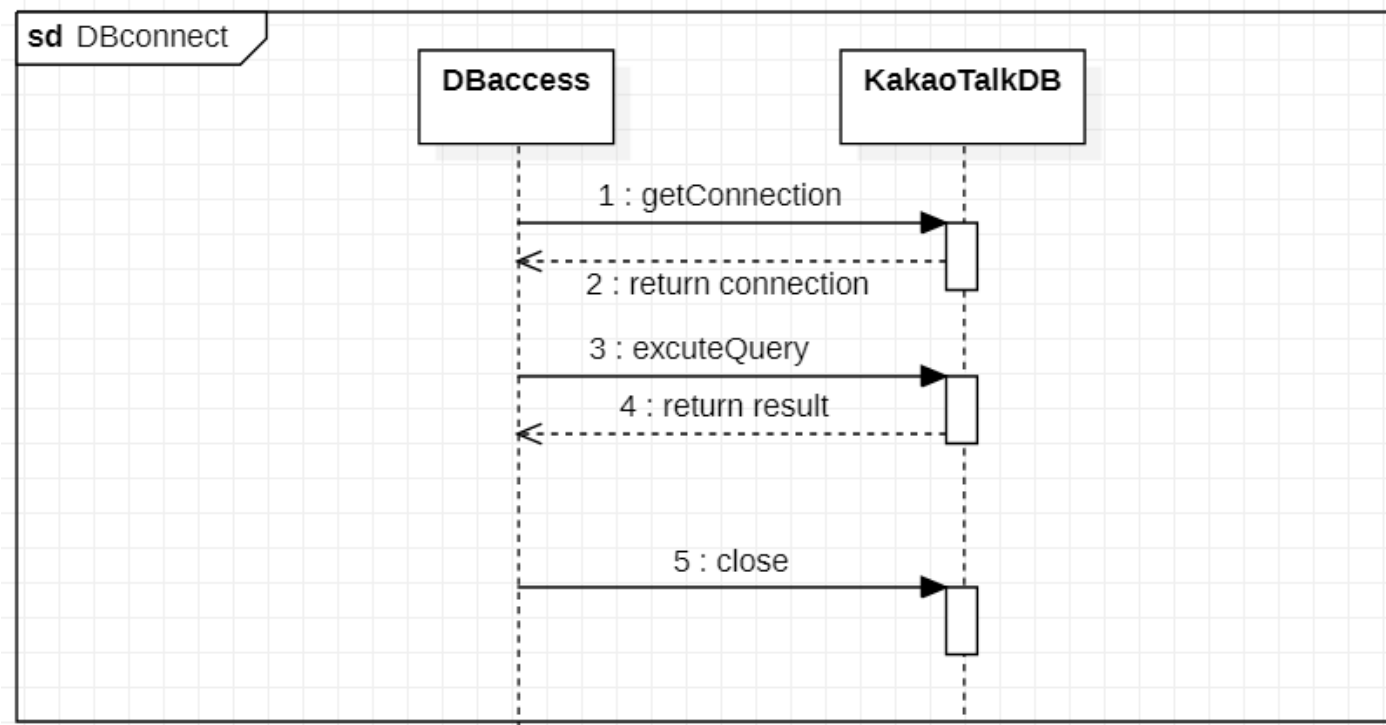
13. AccountHandler는 사용자에게 연결된 서비스가 없음을 출력

**else if (result == false) // 사용자 인증 실패 시**

14. false 반환

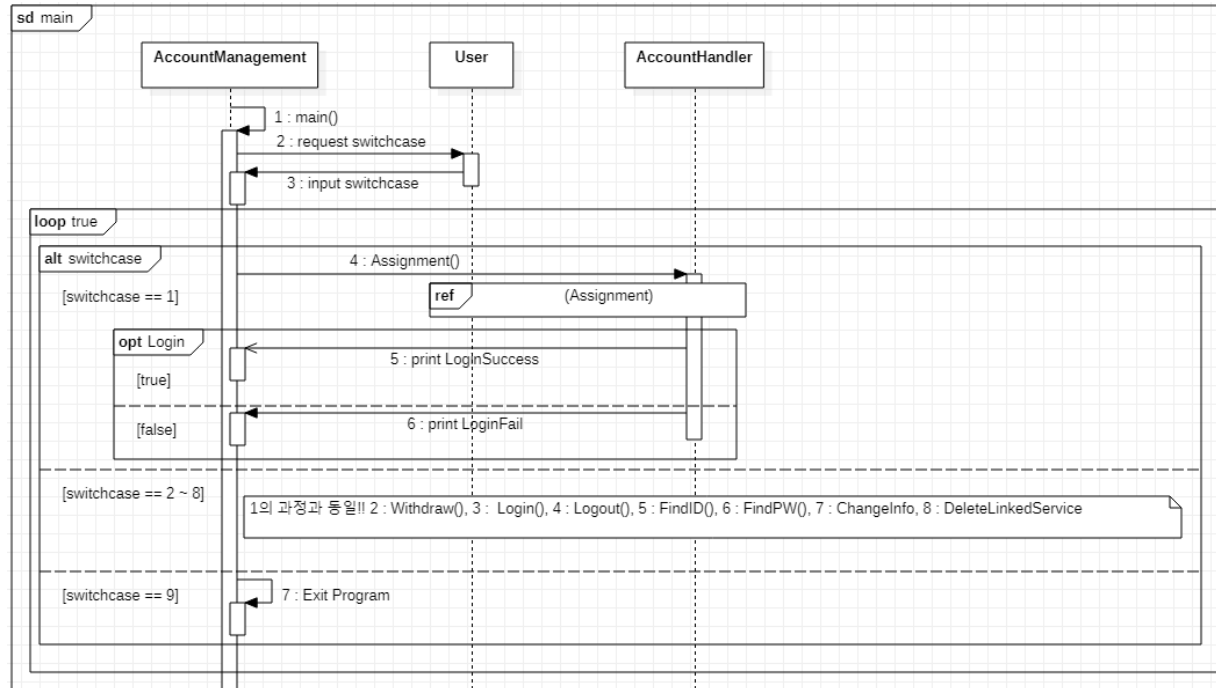
15. true 반환

## S9. DBconnect



1. DBaccess에서 KakaoTalkDB에 getConnection을 요청
2. 연결상태를 반환
3. DBaccess에서 KakaoTalkDB에 excuteQuery를 요청
4. 결과 반환
5. DBaccess에서 KakaoTalkDB에 close를 요청

## S10. Main



1. AccountManagement는 main()을 호출

2. AccountManagement는 User에게 case를 요청

3. User는 AccountManagement에게 case를 입력

\* 이후 User가 입력한 switchcase (1 ~ 8)에 따라 AccountHandler 객체에서 입력한 값에 따른 메소드 실행

\* User가 입력한 switchcase 값이 9라면 프로그램 종료

---

Q & A

---