



C조 설계 구현

- E조 -

20173059 박찬흠
20193062 박찬양
20213139 최주찬
20143146 이진희

목 차

1. 개발 환경

2. 구현전 분석

3. 유스케이스 개요

4. 시퀀스 다이어그램 구현 설명

5. 구현 어려웠던 점

개발 환경

언어

JAVA

사용 IDE

VScode , Eclipse, IntelliJ IDEA

시스템 상황 분석

목적

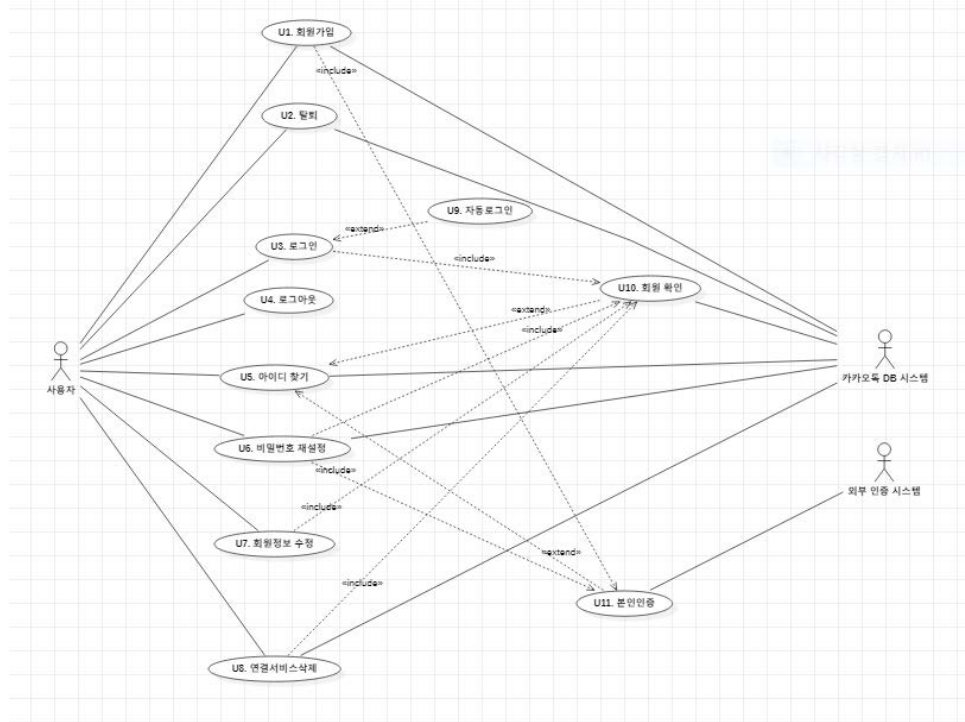
계정관리, DB, 회원가입 등을 할 줄 알아야한다.

제공

유스케이스, 클래스, 시퀀스 다이어그램 제공받음.

구현 전 분석

1. 유스케이스 수신



구현 전 분석

U1.회원가입

U2.탈퇴

U3.로그인

U4.로그아웃

U5.아이디 찾기

U6.비밀번호 재설정

U7.정보 수정

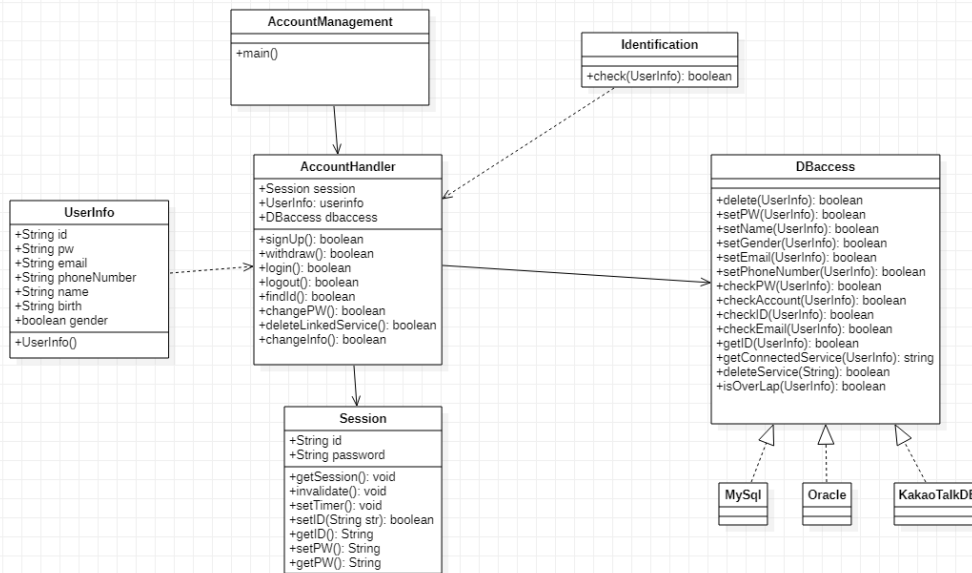
U8.연결 서비스 삭제

U9.자동 로그인

U10.회원 확인

U11.본인 인증

구현 전 분석



구현필요 클래스

AccountManagement

AccountHandler

Session

Identification

UserInfo

DBAccess

MySql

Oracle

KakaoTalkDB

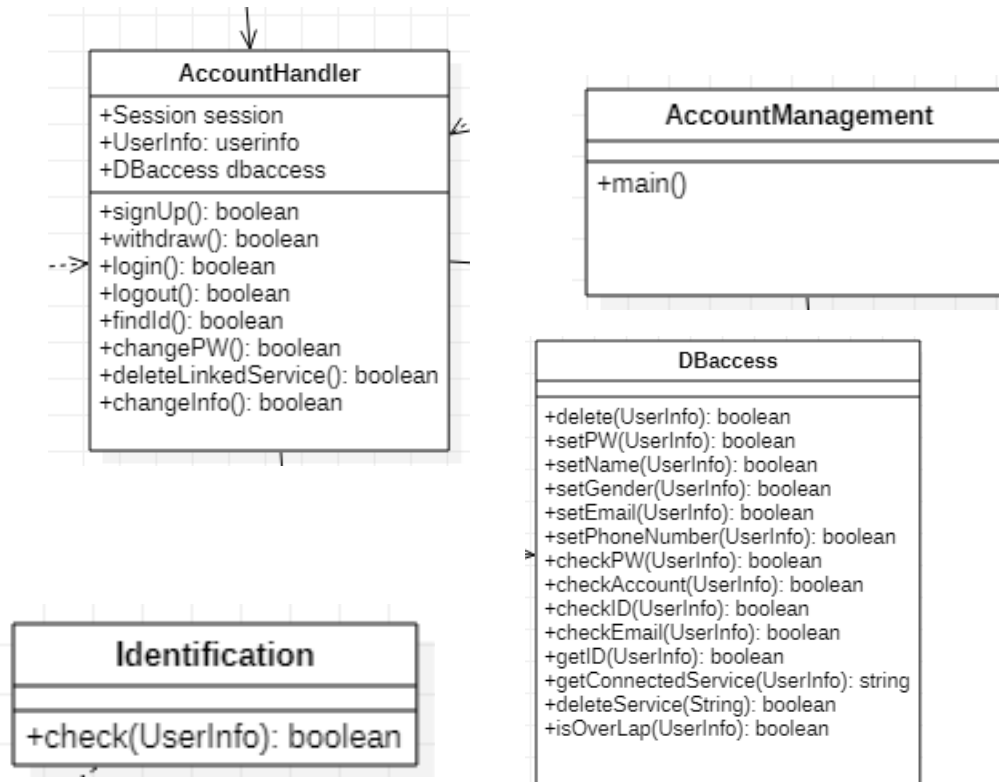
S1. Sign up 유스케이스

유스케이스

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">1. 사용자는 회원가입을 시도한다.2. 시스템은 서비스약관을 보여준다.3. 사용자는 약관에 동의한다.4. 사용자는 아이디, 비밀번호, 이메일, 닉네임, 전화번호를 입력한다.5. 시스템은 인증번호를 전송한다.6. 사용자는 인증번호를 입력한다.7. 카카오톡 DB 시스템은 기존 가입 회원인지 확인한다.8. 카카오톡 DB 시스템에 회원정보를 저장하고 회원가입에 성공한다.		
선택 흐름	<ul style="list-style-type: none">✓ 인증번호가 틀린 경우 인증번호가 올바르지 않다는 메시지를 보여준다.✓ 기존 가입 회원이라면 가입된 계정이 있다는 메시지를 보여준다.		

S1. Sign up 필요 클래스

필요 클래스



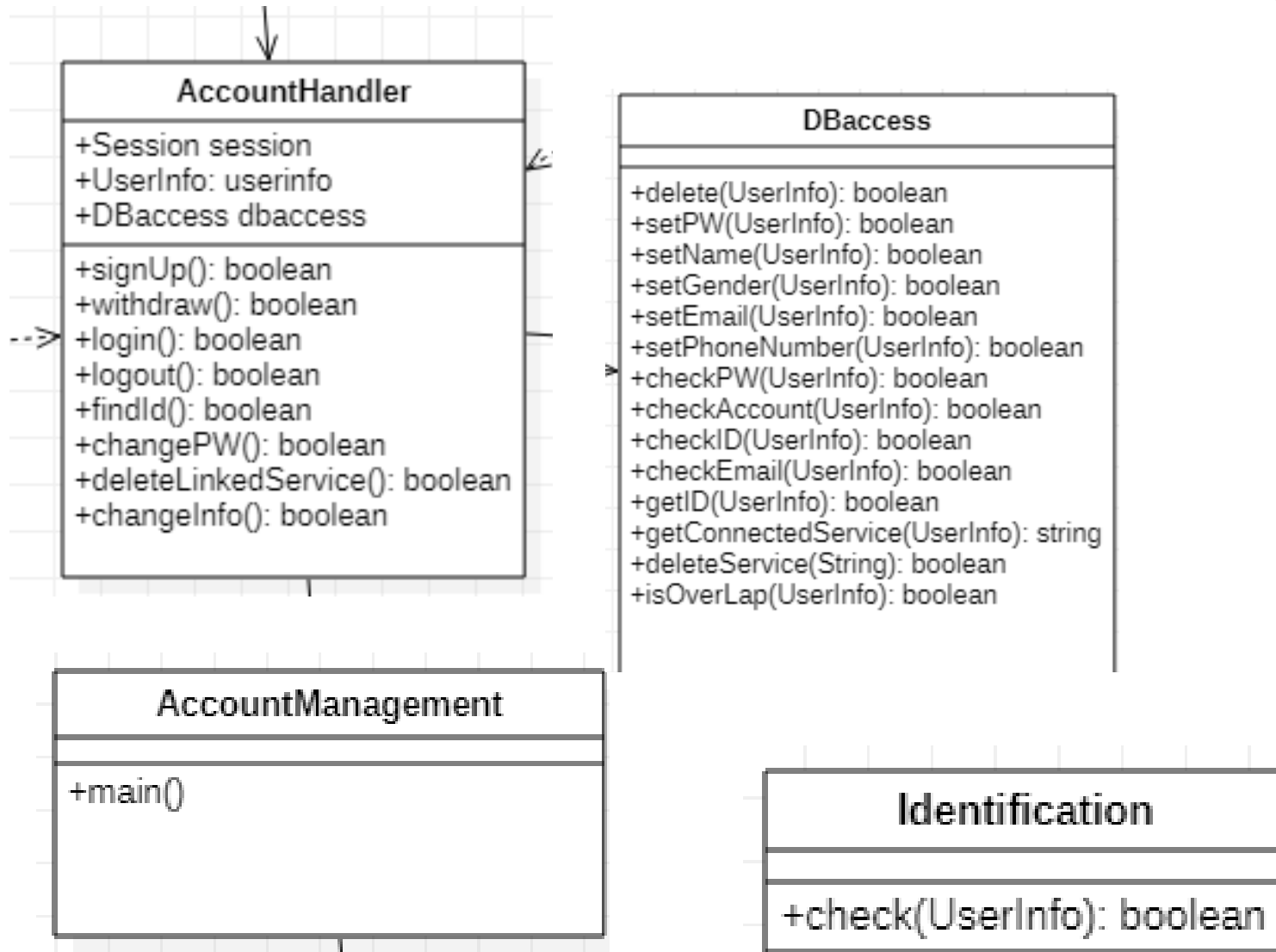
S2.Withdraw 유스케이스

유스케이스

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	<ol style="list-style-type: none">1. 사용자는 탈퇴를 요청한다.2. 사용자는 개인 정보를 입력한다.3. 외부 인증 시스템은 본인인증을 확인한다.4. 카카오톡 DB시스템은 사용자 정보를 삭제한다.		
선택 흐름	✓ 외부 인증 시스템에서 본인이 아닐 경우 본인 인증에 실패한다.		

S2.Withdraw 필요클래스

필요 클래스



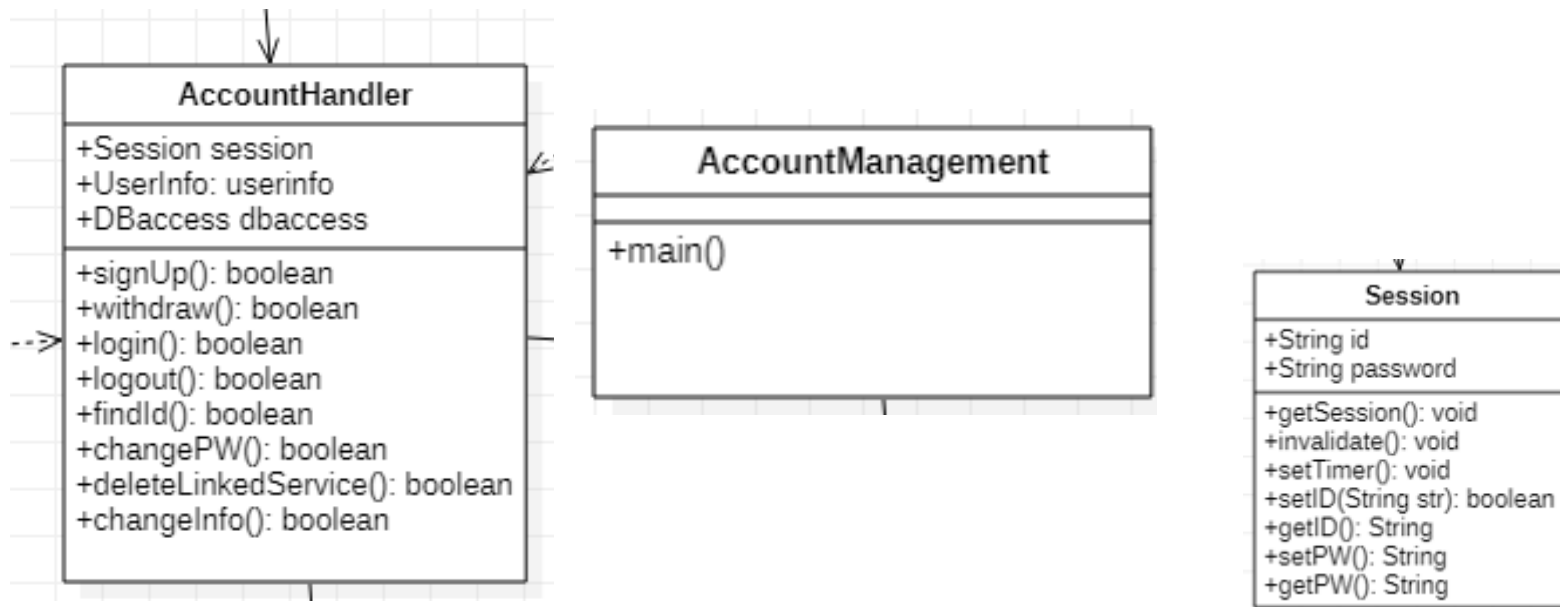
S4. Logout 유스케이스

유스케이스

유스케이스 명	U4. 로그아웃	유스케이스 개요	사용자가 로그아웃을 하기 위한 유스케이스이다.
액터명	사용자	사전 조건	사용자는 로그인 상태여야 한다.
정상 흐름	<ol style="list-style-type: none">1. 사용자는 로그아웃을 요청한다.2. 시스템은 세션을 무효화하여 로그인 된 계정을 로그아웃 시킨다.		
선택 흐름	✓ 세션 타임아웃이 발생하면 자동으로 로그아웃 된다.		

S4. Logout 필요 클래스

필요 클래스



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

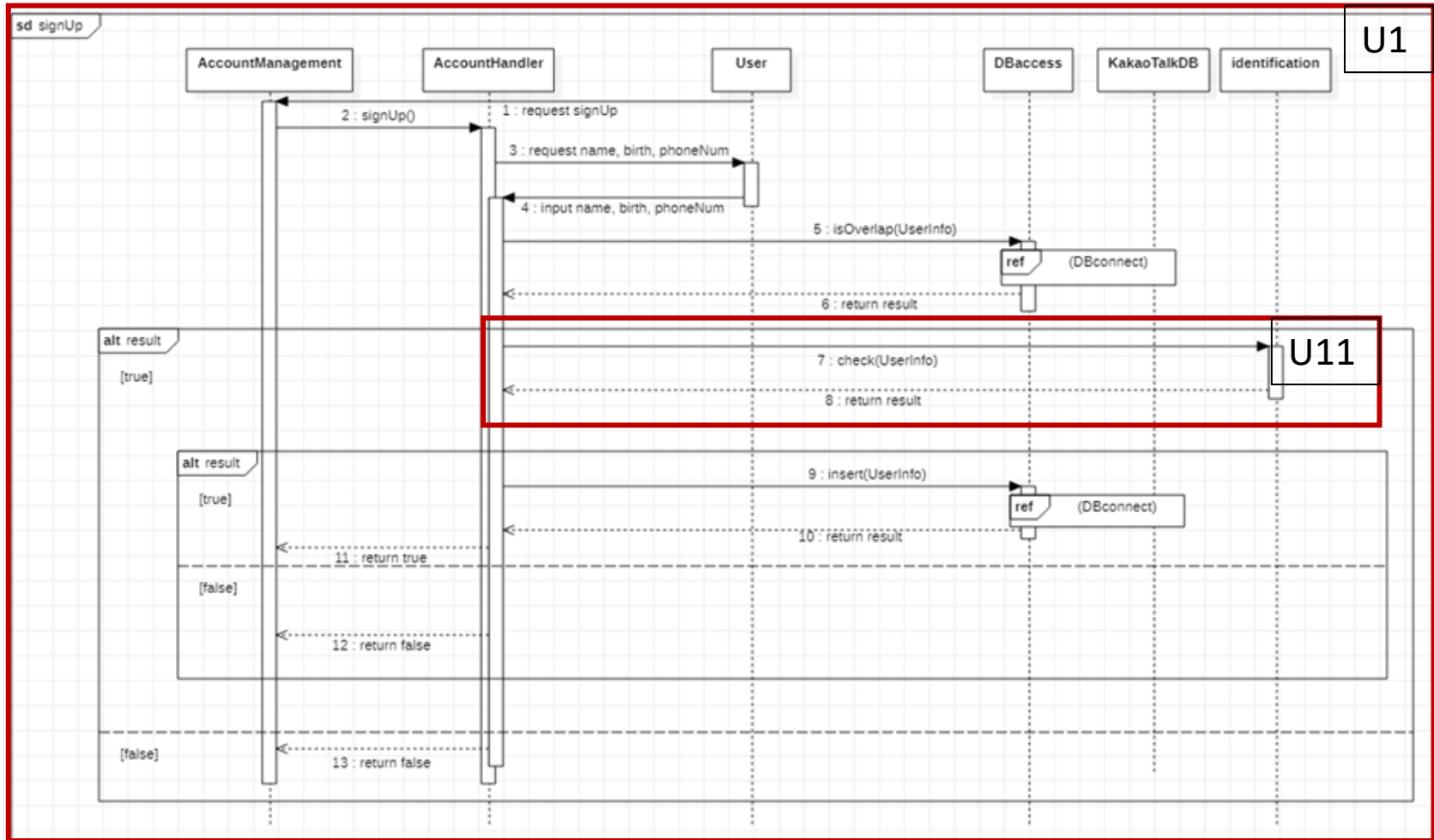
U11.본인 인증

시퀀스 다이어그램 분석

S1. SignUp 분석

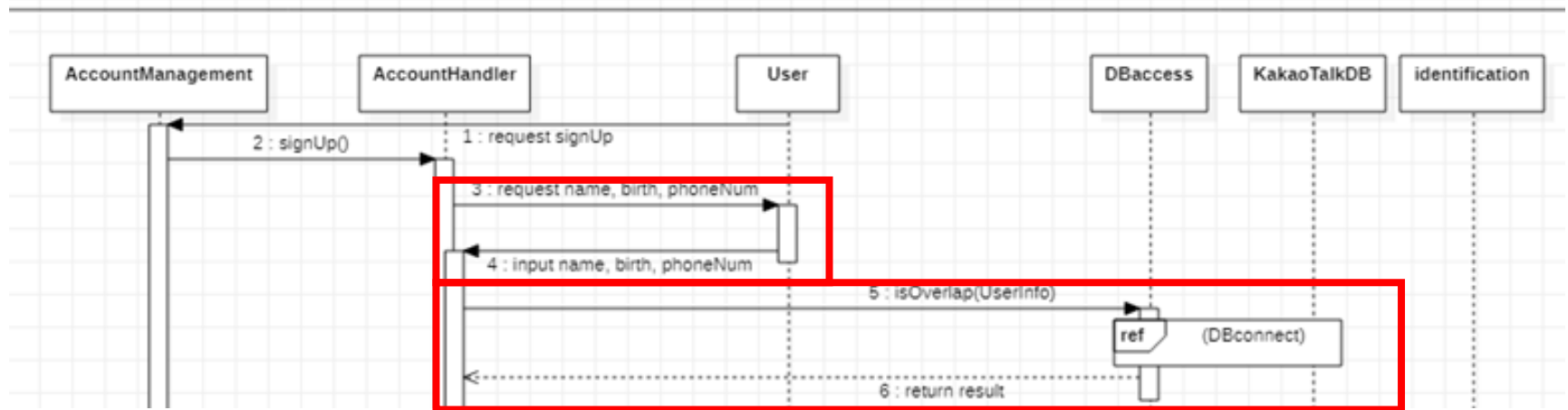
U1.회원가입

U11.본인 인증



시퀀스 다이어그램 분석

S1. Sign up 분석 – AccountHandler.signUp()



```
public boolean signUp() {
    System.out.println("requested signUp by user\nenter name, birth and phoneNum");
    Scanner sc = new Scanner(System.in);
    boolean result = true;
    userInfo.name = sc.next();
    userInfo.birth = sc.next();
    userInfo.phoneNumber = sc.next();
    sc.close();

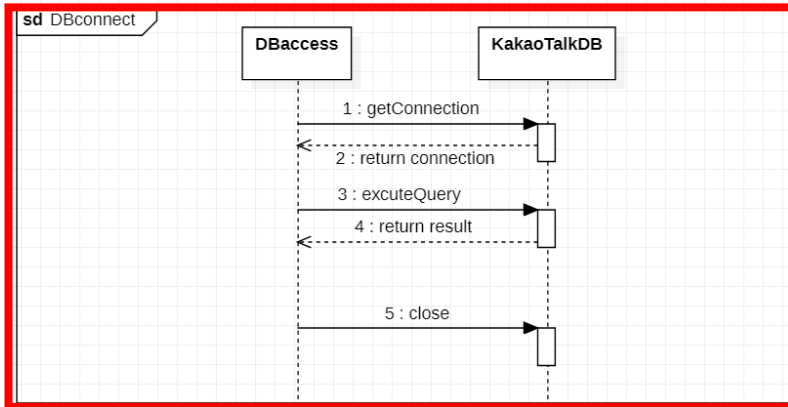
    result = dbaccess.isOverLap(userInfo);
}
```

3,4

5,6

시퀀스 다이어그램 분석

S1. Sign up 분석 – Dbaccess.isOverLap()



1,2,3,4,5

```
public boolean isOverLap(UserInfo userInfo) {
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    // 예외처리 연결이갑자기끊겼다면지 등으로 이상하게 받아온경우

    // 사용자가 입력한 계정 정보가 중복되는지 확인, 보낸 값과 받아온 값이 같다면, 즉 중복된다면 false
    if (userInfo.id.equals(result.id) && userInfo.pw.equals(result.pw)) {
        return false;
    }
    return true;
}
```

시퀀스 다이어그램 분석

S1. Sign up 분석 – AccountHandler.signUp()



```
if (result == true) {  
    // 사용자의 정보를 받아 외부시스템에서 확인  
    boolean result1 = new identification().check(userInfo);  
  
    // result1 == true 가입 가능할 정보일 경우  
    if (result1 == true) {  
        // dbaccess에 추가  
        boolean result2 = dbaccess.insert(userInfo);  
        // 회원가입 기능하면 true 리턴  
        System.out.println("able to sign up ");  
        return result2;  
    } else {  
        // 회원가입 기능 불가능하면 false 리턴  
        System.out.println("Unable to sign up ");  
        return result1;  
    }  
}
```

7,8

9,10,11,12

시퀀스 다이어그램 분석

S1. Sign up 분석 -identification.check()

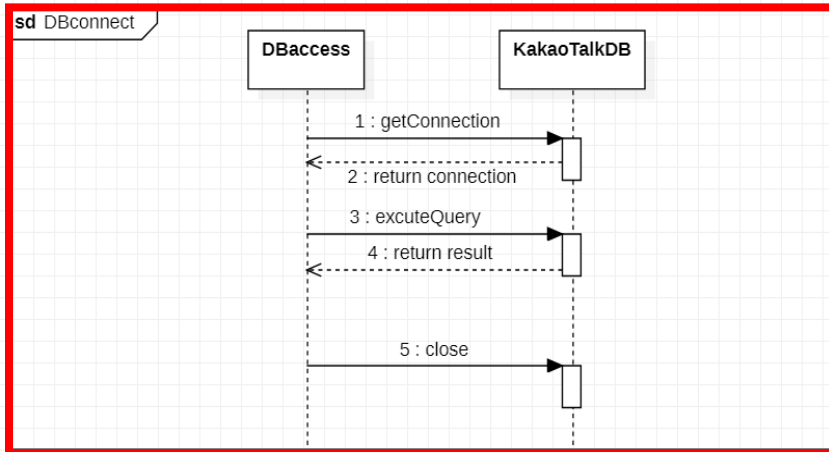


```
public class identification {
    public boolean check(UserInfo userInfo)
    {
        boolean result = true;
        System.out.println("외부 시스템에서 체크");
        return result;
    }
}
```

7,8

시퀀스 다이어그램 분석

S1. Sign up 분석 – DBaccess.insert()

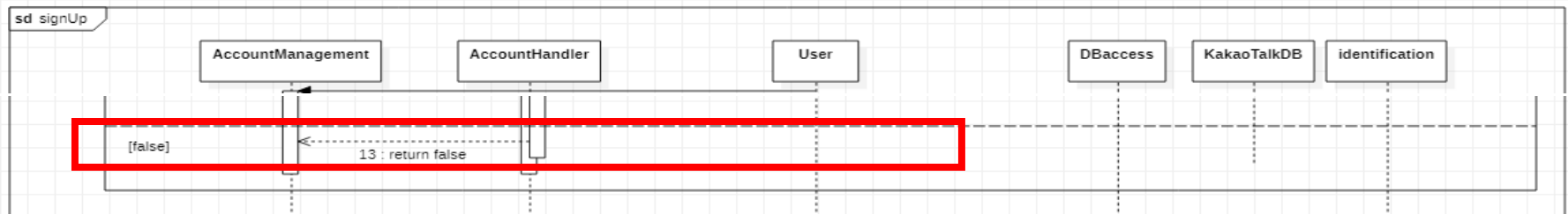


1,2,3,4,5

```
// 시퀀스 다이어그램에는 있지만 클래스에 없어서 추가
public boolean insert(UserInfo userInfo) {
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo); // DB에 사용자 정보를 입력한다.
    kakaoTalkDB.close();
    // 예외처리 연결이 갑자기 끊겼다면지 등으로 이상하게 받아온 경우
    // 성공적일 경우
    if (result != null) {
        return true;
    }
    // 실패할 경우
    return false;
}
```

시퀀스 다이어그램 분석

S1. Sign up 분석 – AccountHandler.signUp()



```
} else {  
    // result == false  
    // 정보가 중복되는 경우  
    // 회원가입 기능 false 리턴  
    System.out.println("Unable to sign up ");  
    return result;  
}
```

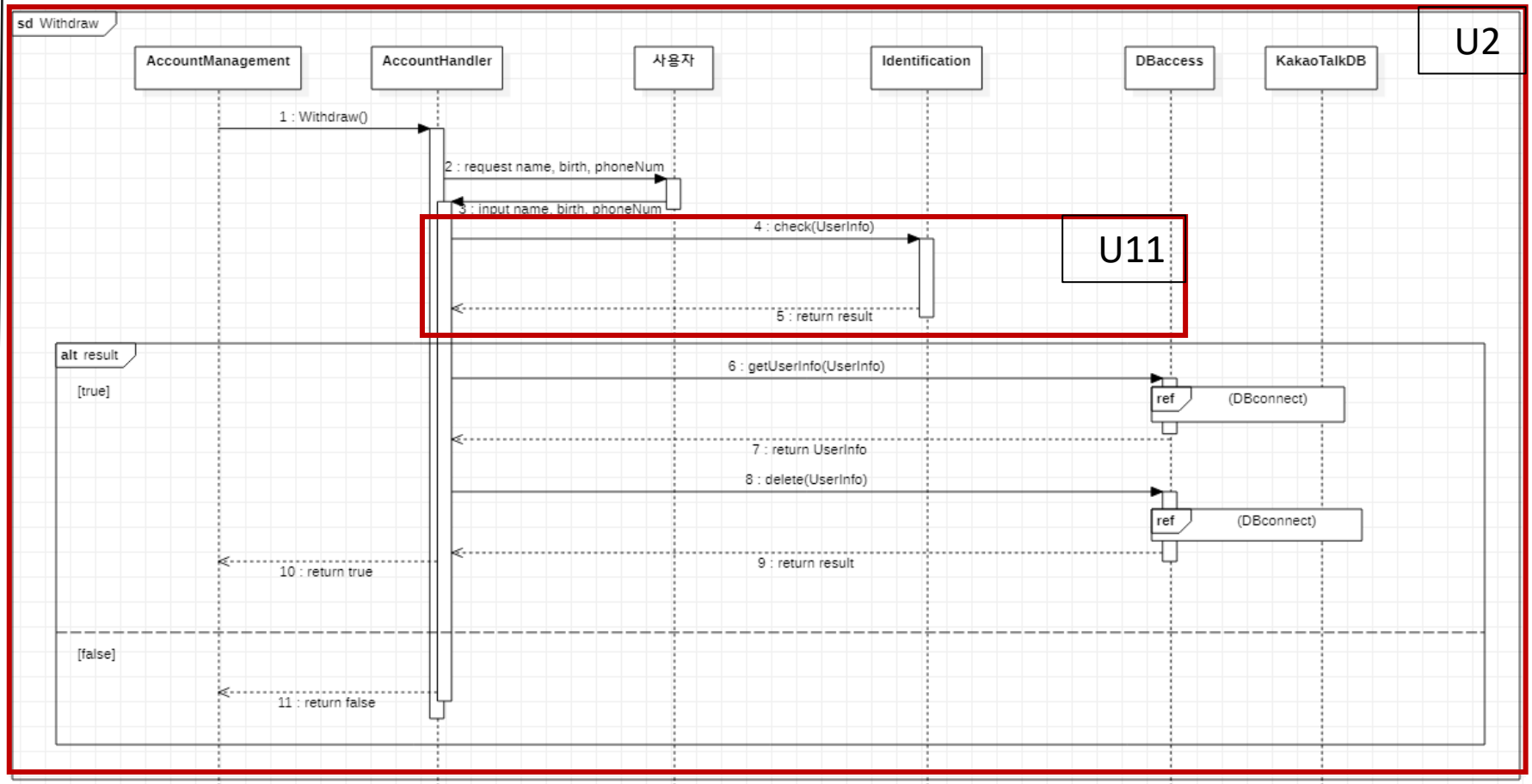
13

시퀀스 다이어그램 분석

S2. Withdraw 분석

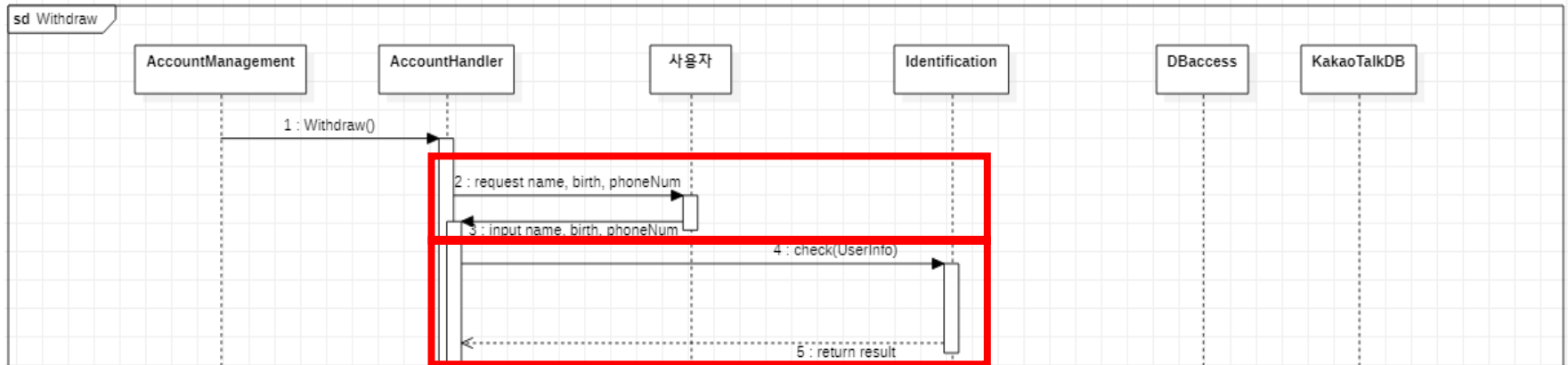
U2.
탈퇴

U11.
본인 인증



시퀀스 다이어그램 분석

S2. Withdraw 분석 – AccountHandler.withdraw()



```
public boolean withdraw() {
    System.out.println("requested withdraw by user\nenter name, birth and phoneNumber");
    Scanner sc = new Scanner(System.in);
    boolean result = true;
    userInfo.name = sc.next();
    userInfo.birth = sc.next();
    userInfo.phoneNumber = sc.next();
    sc.close();

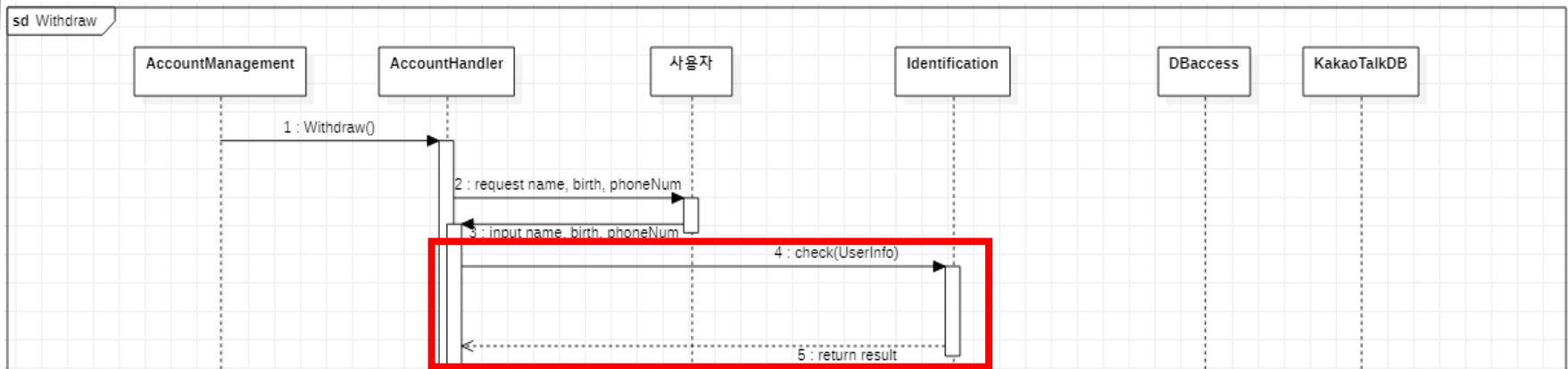
    // 사용자의 정보를 받아 외부시스템에서 확인
    result = new identification().check(userInfo);
}
```

4,5

2,3

시퀀스 다이어그램 분석

S2. Withdraw 분석 – identifcation.check()



```
public class identification {  
    public boolean check(UserInfo userInfo)  
    {  
        boolean result = true;  
        System.out.println("외부 시스템에서 체크");  
        return result;  
    }  
}
```

→ 4,5

시퀀스 다이어그램 분석

S2. Withdraw 분석 - AccountHandler.withdraw()



```
if (result == true) {  
    // 본인인증 정보를 토대로 사용자의 모든 UserInfo 반환  
    UserInfo userinfo = dbaccess.getUserInfo(userInfo);  
    // 적절하게 값이 삭제가 되었을 시 리턴 값으로 true;  
    boolean result1 = dbaccess.delete(userinfo);  
    System.out.println("withdraw successful! ");  
    return result1;  
}  
// 본인이 아닐 경우  
else {  
    System.out.println("withdraw unsuccessful! ");  
    return false;  
}
```

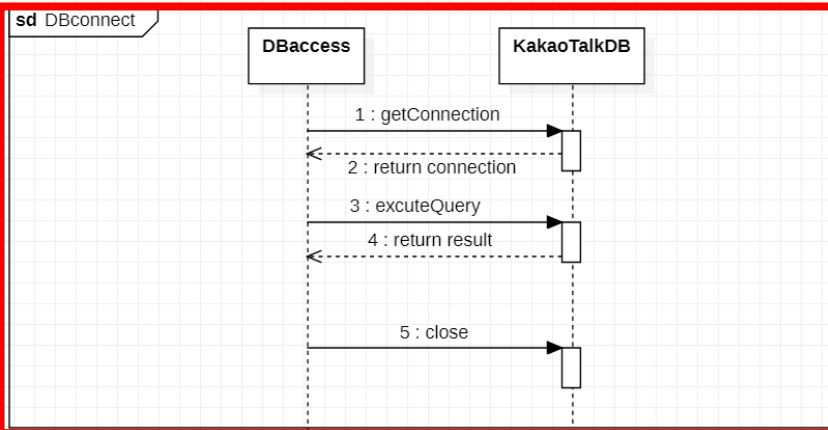
6,7

8,9,10

11

시퀀스 다이어그램 분석

S2. Withdraw 분석 – DBaccess.getUserInfo(), DBaccess.delete()



```
// 시퀀스 다이어그램에는 있지만 클래스에 없어서 추가
public UserInfo getUserInfo(UserInfo userInfo) {
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();

    // 예외처리 연결이 갑자기 끊겼다면지 등으로 이상하게 받아온경우
    if (result != null) {
        return result;
    }
    return null;
}
```

```
public boolean delete(UserInfo userInfo) {
    // 사용자 정보 삭제
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();

    //예외처리
    //삭제가 실패할 경우
    if (result == null) {
        return false;
    }
    //성공할 경우
    return true;
}
```

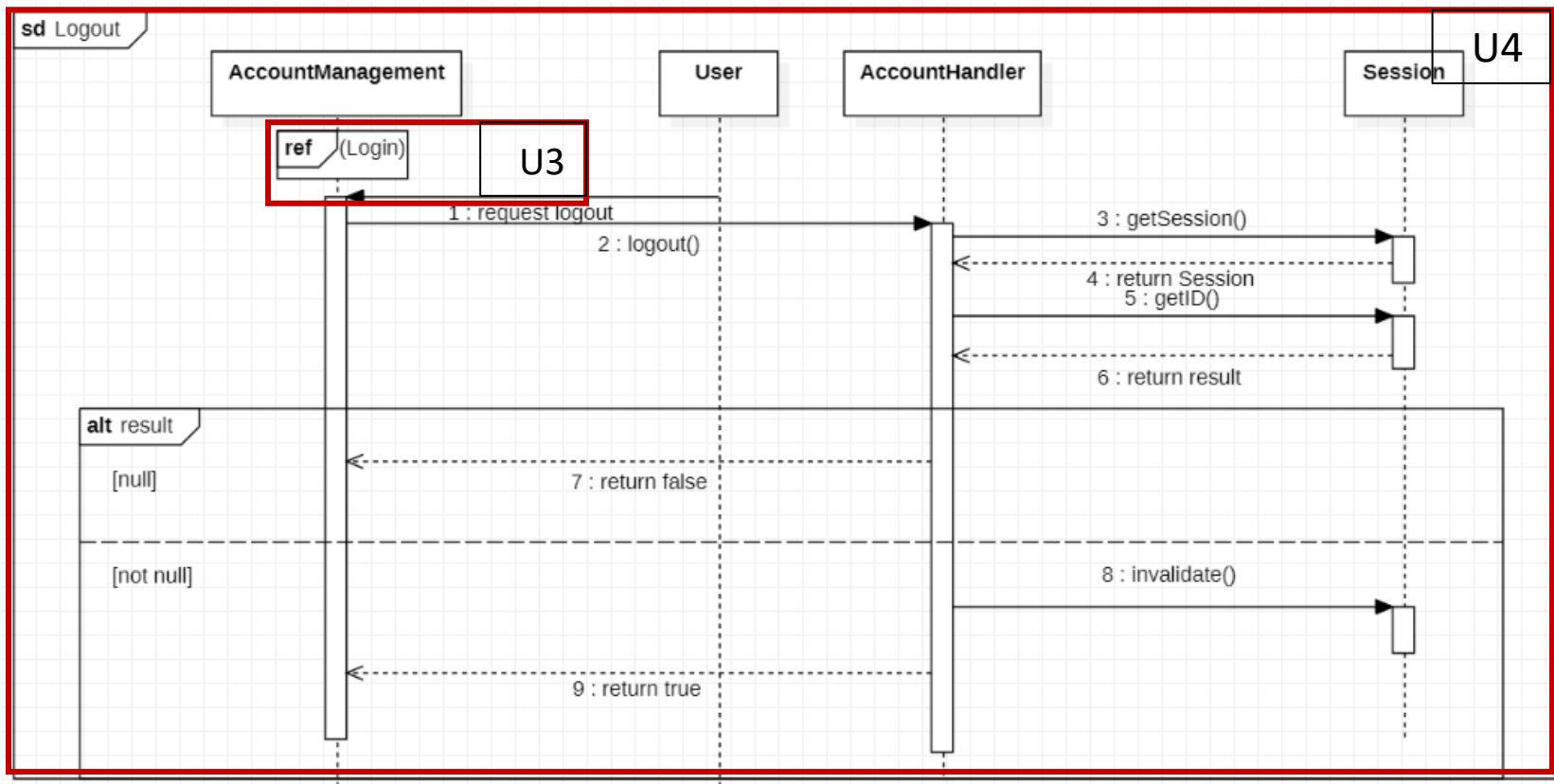
1,2,3,4,5

시퀀스 다이어그램 분석

S4. Logout 분석

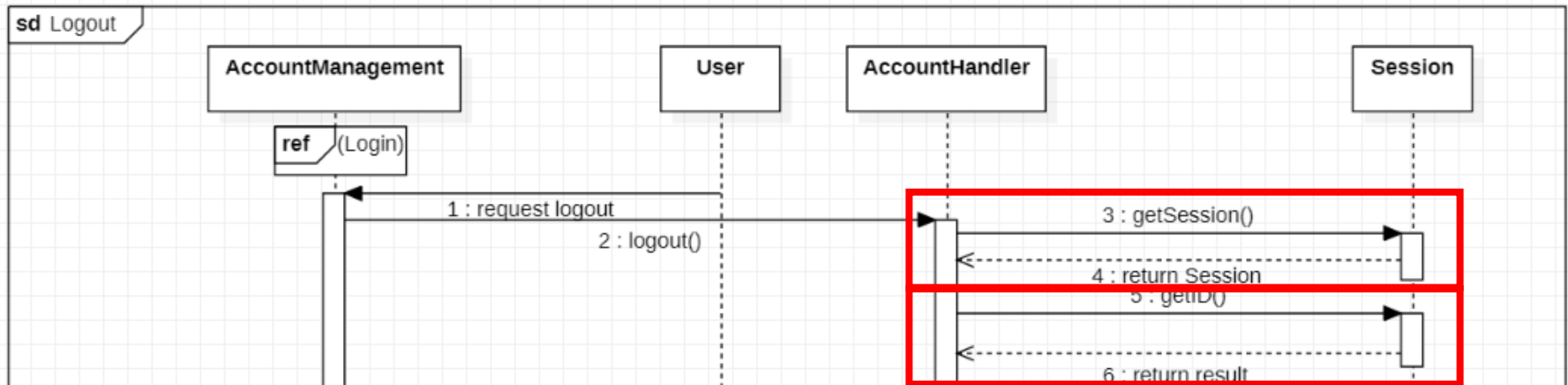
U3.로그인

U4. 로그아웃



시퀀스 다이어그램 분석

S4. Logout 분석 – AccountHandler.logout()



```
public boolean logout() {  
    System.out.println("requested logout by user");  
    // 세션연결  
    session.getSession();
```

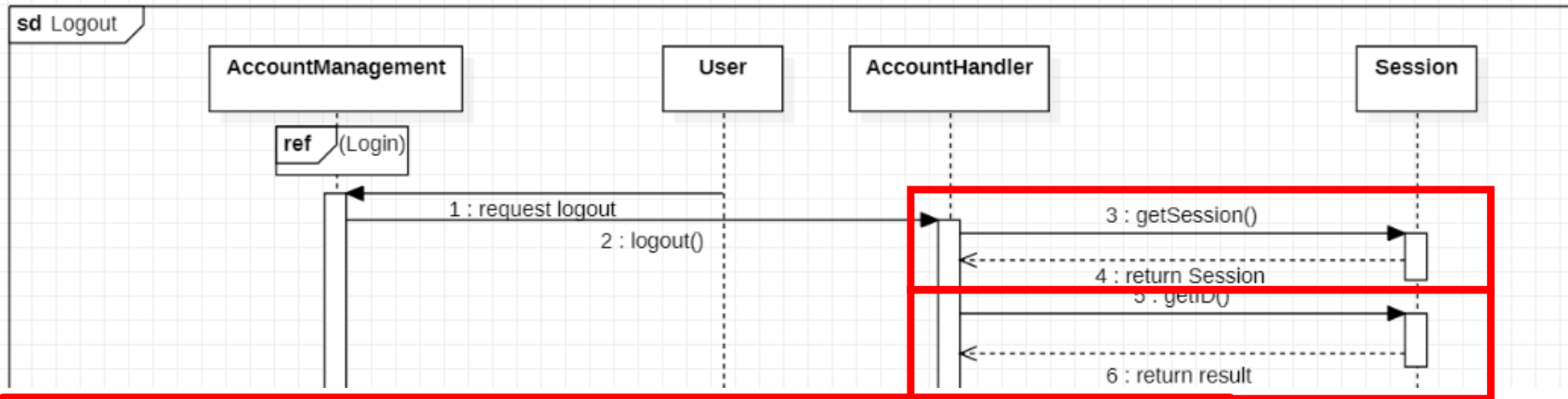
3,4

```
    // result 값으로 아이디 값을 받는다.  
    String result = session.getID();
```

5,6

시퀀스 다이어그램 분석

S4. Logout 분석 – Session.getSession(), Session.getID()



```
public Session getSession() //getSession 반환형 void로 수정하라 하심
// 메소드 이름이 get임, 세션을 연결은 한다고하는데 어디에 연결되는지에 대해서는 설계 x 따라서 Session으로 반환함
//하지만 새로운 세션을 연다고 가정했을 때 그 정보가 유지되지 않으므로 getSession의 자료형을 Session으로 바꾸어 구현하도록함
{
    Session newSession = this; //this = 새로운 세션이라 가정
    System.out.println("세션 연결됨");
    timer = 10; //생성자 없어서 여기서 초기화 클래스다이어그램에 10으로 하라고 명시됨
    setTimer();

    return newSession;
}
```

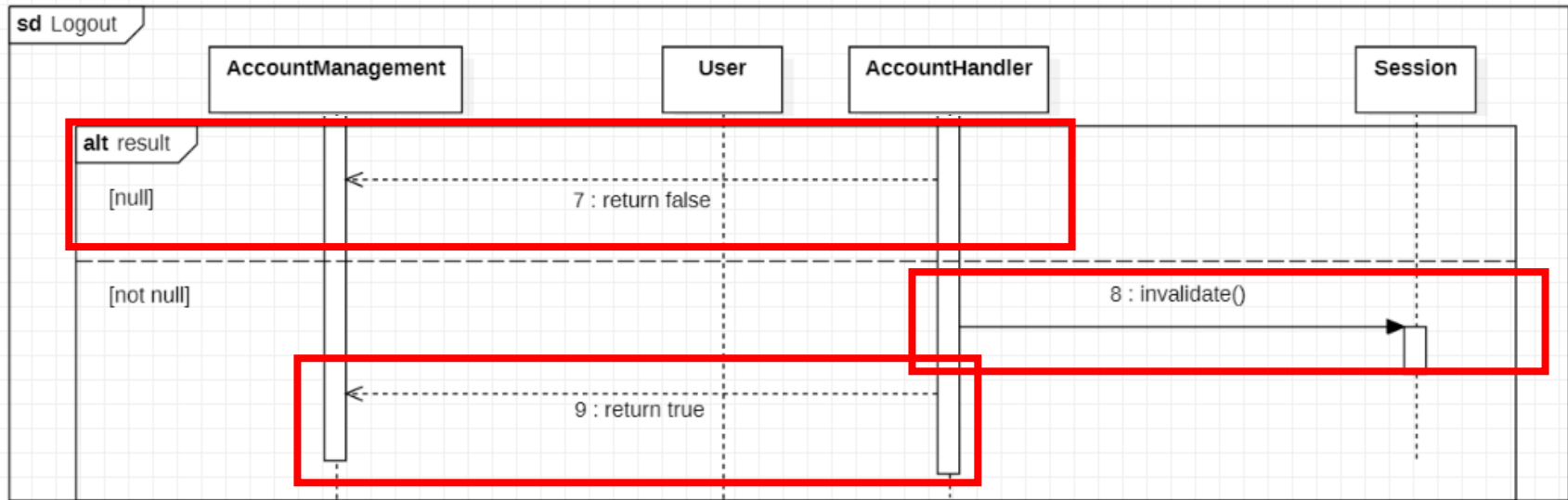
3,4

```
public String getID() //세션에서 아이디를 가져옴
{
    return id;
}
```

5,6

시퀀스 다이어그램 분석

S4. Logout 분석 – AccountHandler.logout()



```
// 아이디가 없는 경우
if (result == null) {
    /*
     * 세션 타임아웃으로 자동로그아웃 된 상태임으로 false 반환
     */
    System.out.println("Automatically log out ");
    return false;
} else {
    // 세션과 연결 해제
    session.invalidate();
    // 그 후, 로그아웃이 성공을 하였다고 true 반환
    System.out.println("Logout Successful! ");
    return true;
}
```

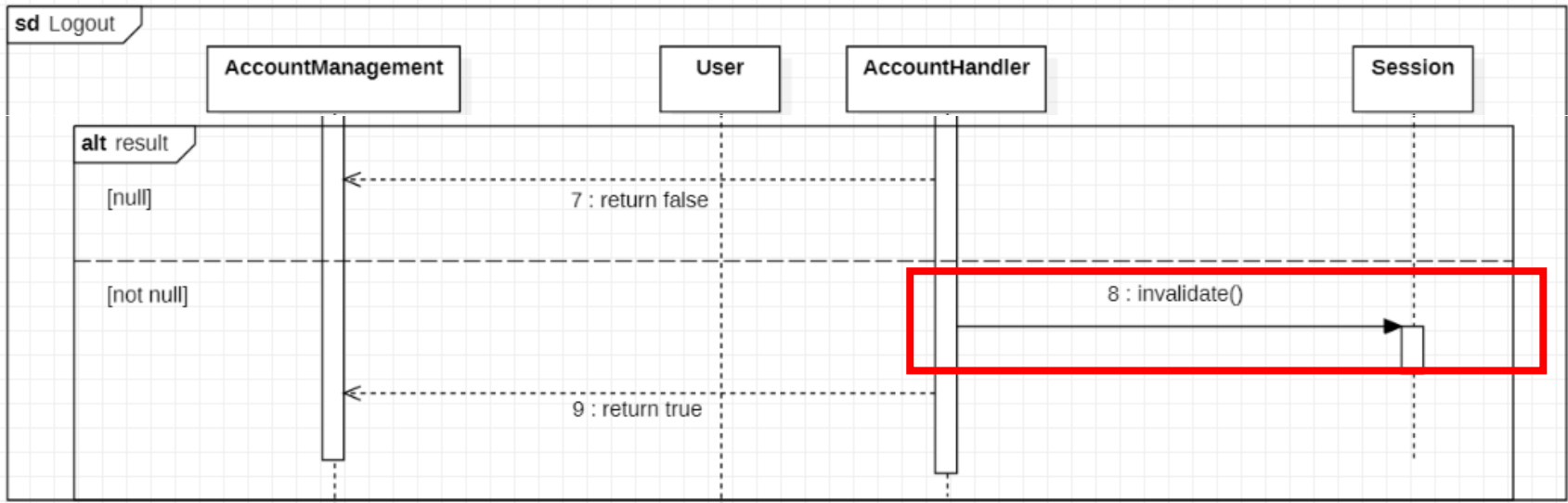
7

8

9

시퀀스 다이어그램 분석

S4. Logout 분석 – Session.invalidate()



```
public void invalidate() //세션과 연결 해제
{
    System.out.println("세션 연결 해제됨");
}
```

8

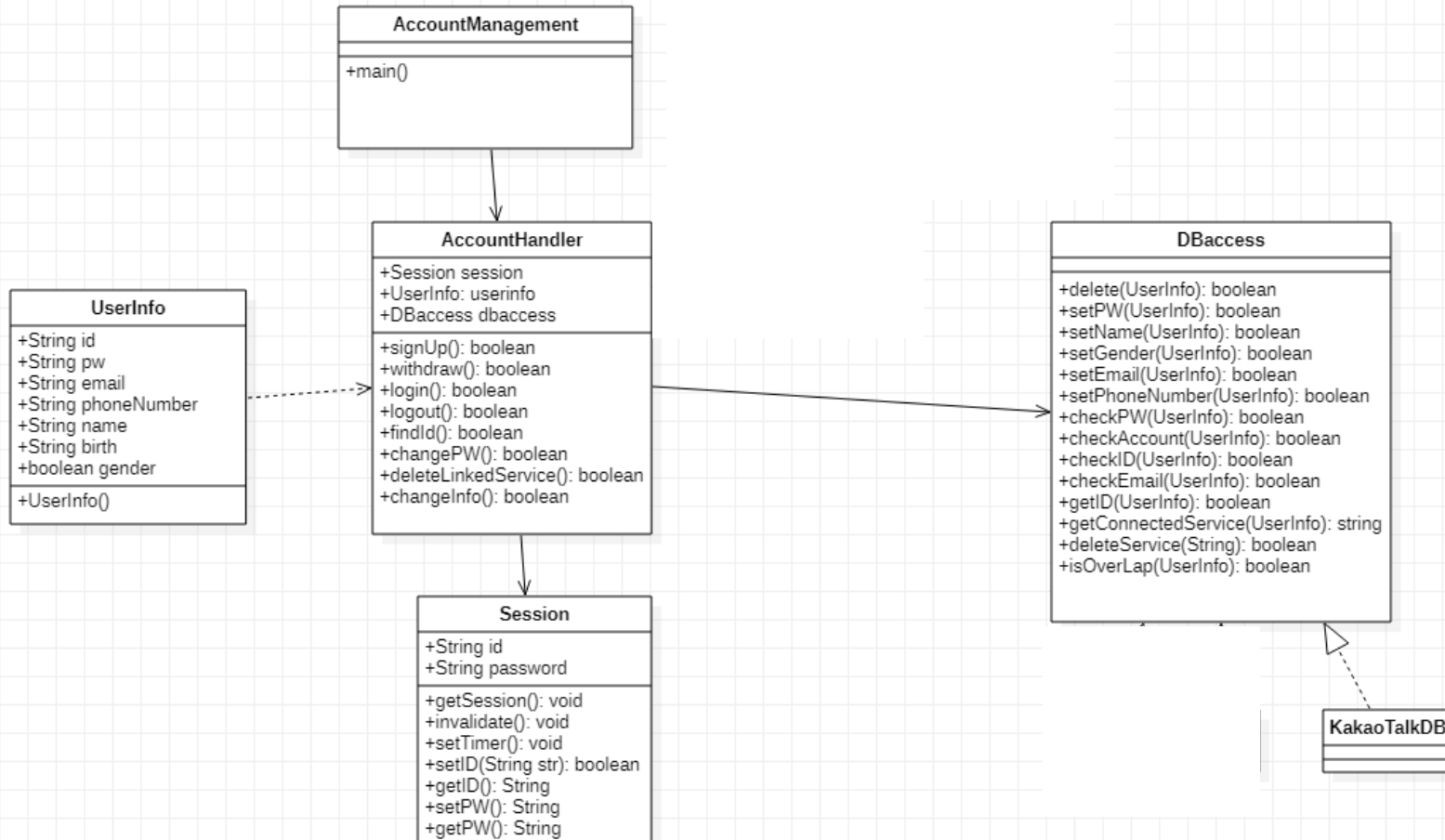
S3.Login 유스케이스

유스케이스

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">1. 사용자는 로그인을 시도한다.2. 사용자는 아이디와 비밀번호를 입력한다.3. 사용자는 로그인을 요청한다.4. 카카오톡 DB 시스템은 사용자는 기존 회원인지 확인한다.5. 세션에 로그인정보를 저장한다.6. 사용자는 로그인에 성공한다.		
선택 흐름	✓ 회원정보가 맞지 않으면 로그인에 실패한다.		

S3.Login 필요 클래스

필요 클래스



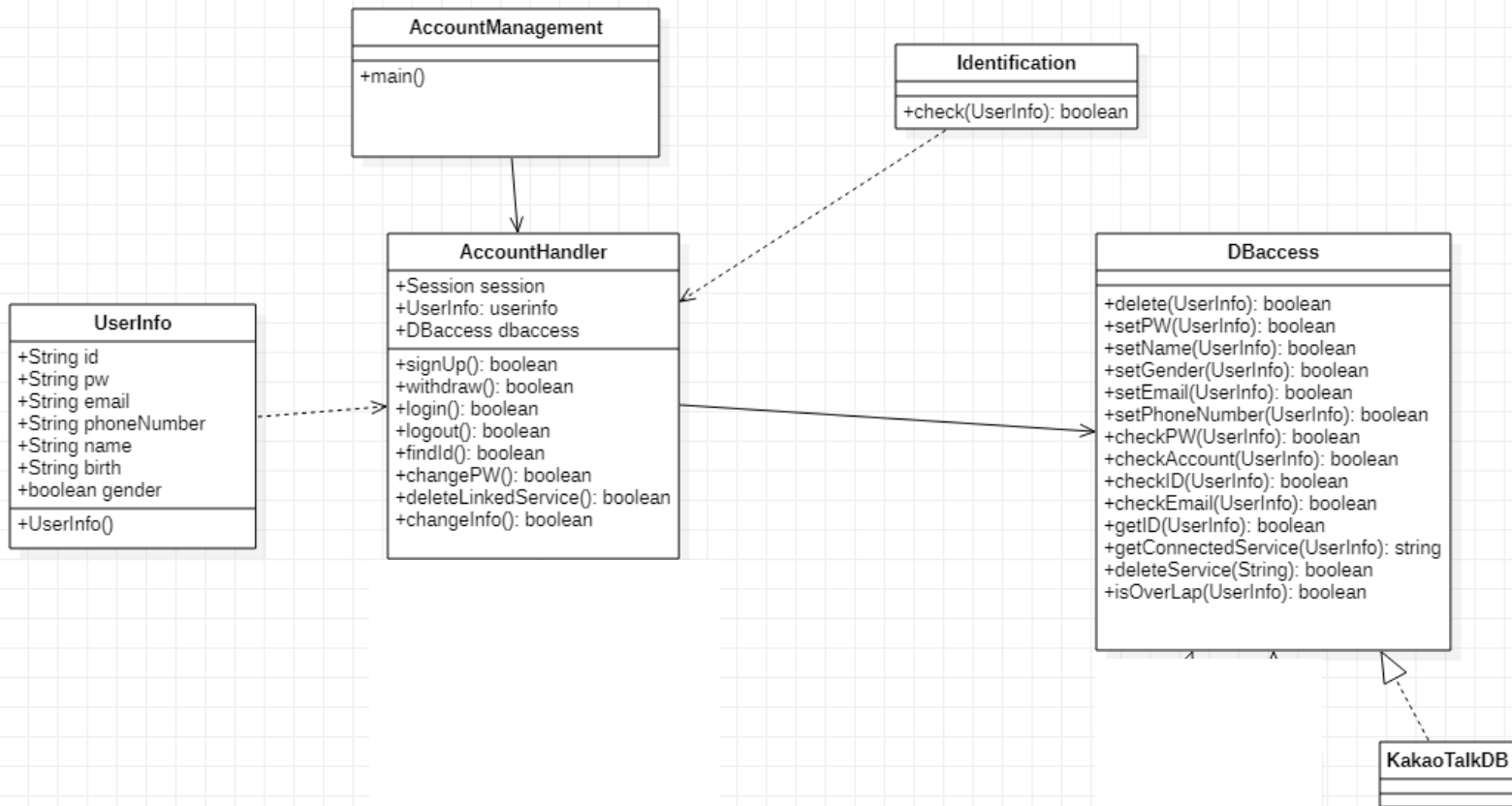
S5.FindID 유스케이스

유스케이스

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

S5.FindID 필요 클래스

필요 클래스



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

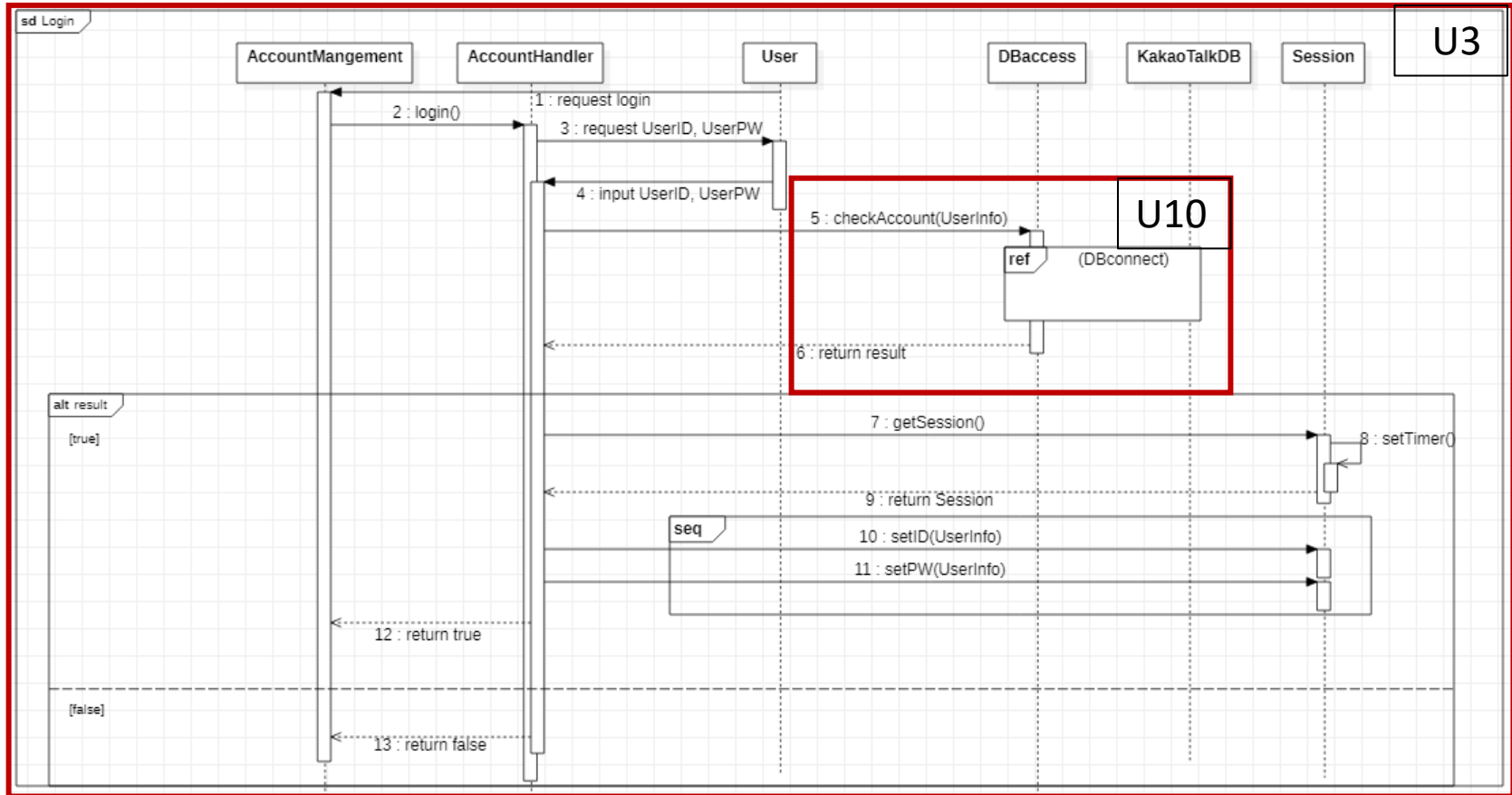
U11.본인 인증

시퀀스 다이어그램 분석

S3. Login 분석

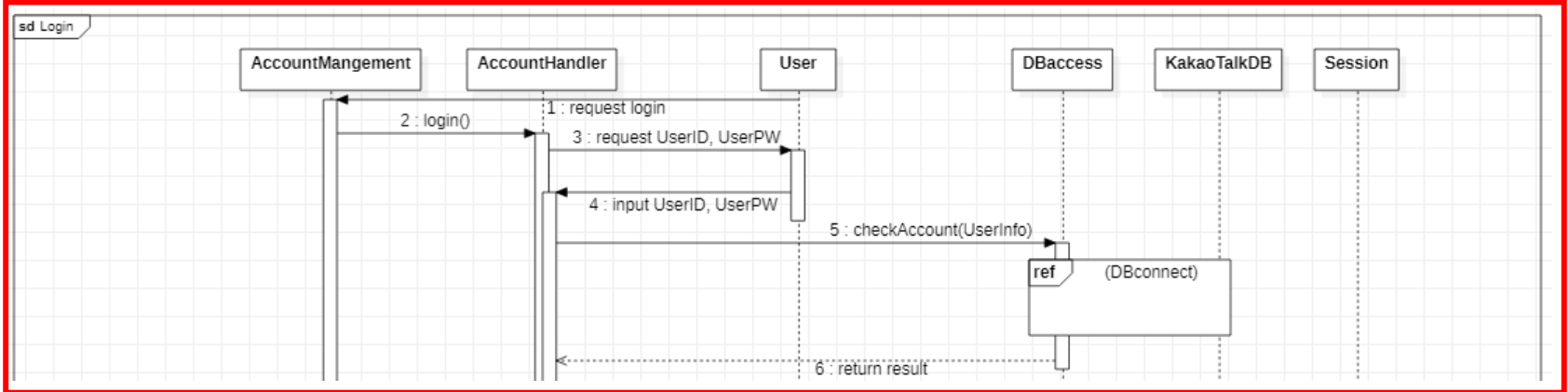
U3.로그인

U10.회원
확인



시퀀스 다이어그램 분석

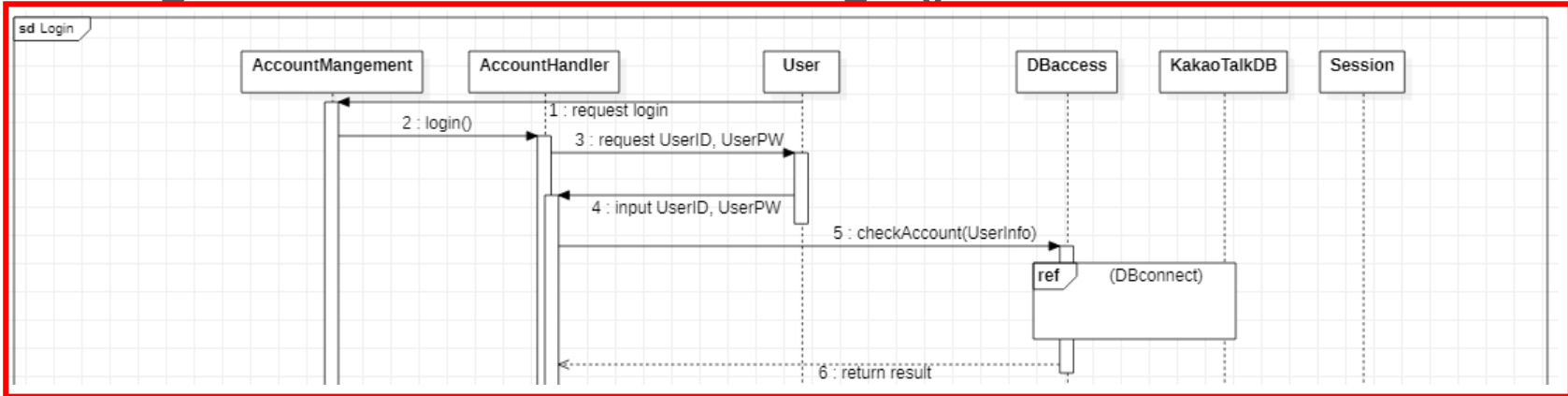
S3. Login 분석



- 1: UI적인 부분이므로 메소드 존재 x
- 2: UI에서 사용자가 클릭했다고 가정 login() 실행

시퀀스 다이어그램 분석

S3. Login 분석 – AccountHandler.login()

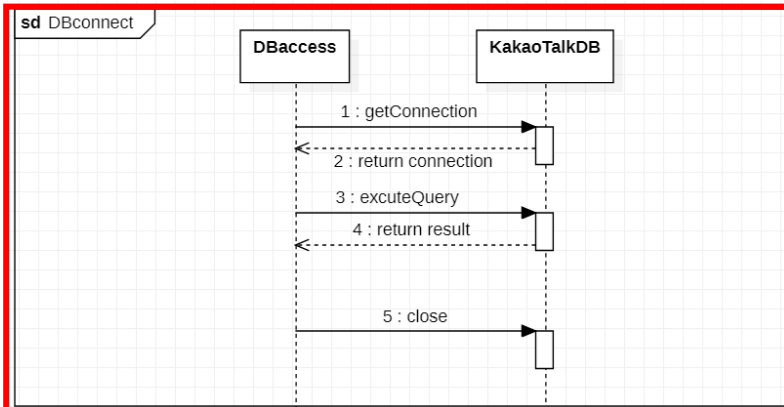


```
4 public class AccountHandler {
5     Session session = new Session();
6     UserInfo userInfo = new UserInfo();
7     DBAccess dbaccess = new DBAccess();
8     public boolean login()
9     {
10         System.out.println("requested login by user\nenter id and password");
11         Scanner sc = new Scanner(System.in);
12         boolean result = true;
13         userInfo.id = sc.next();
14         userInfo.pw = sc.next();
15         result = dbaccess.checkAccount(userInfo);
16         if(result)
17         {
18             session = session.getSession(); //void 형이라고 하심
19             session.setID(userInfo);
20             session.setPW(userInfo);
21         }
22         sc.close();
23         return result; //always return true there is no case that return value can be false
24     }
}
```

3,4,5

시퀀스 다이어그램 분석

S3. Login 분석 – DBAccess.checkAccount()

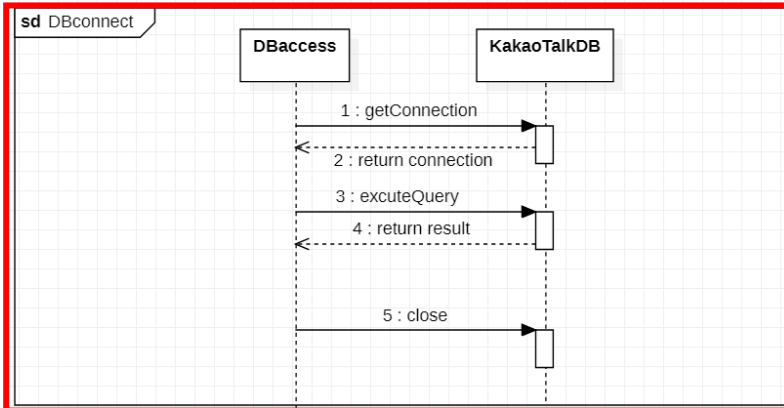


```
26 public boolean checkAccount(UserInfo userInfo){
27     KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
28     boolean connection = kakaoTalkDB.getConnection(); //connection 변수 사용 x
29     UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
30     kakaoTalkDB.close();
31     //예외처리 + id, pw 동일한지 체크
32     if(result != null && userInfo.id.equals(result.id) && userInfo.pw.equals(result.pw))
33     {
34         return true;
35     }
36     return false;
37 }
```

1,2,3,4,5

시퀀스 다이어그램 분석

S3. Login 분석 – KakaoTalkDB 메소드

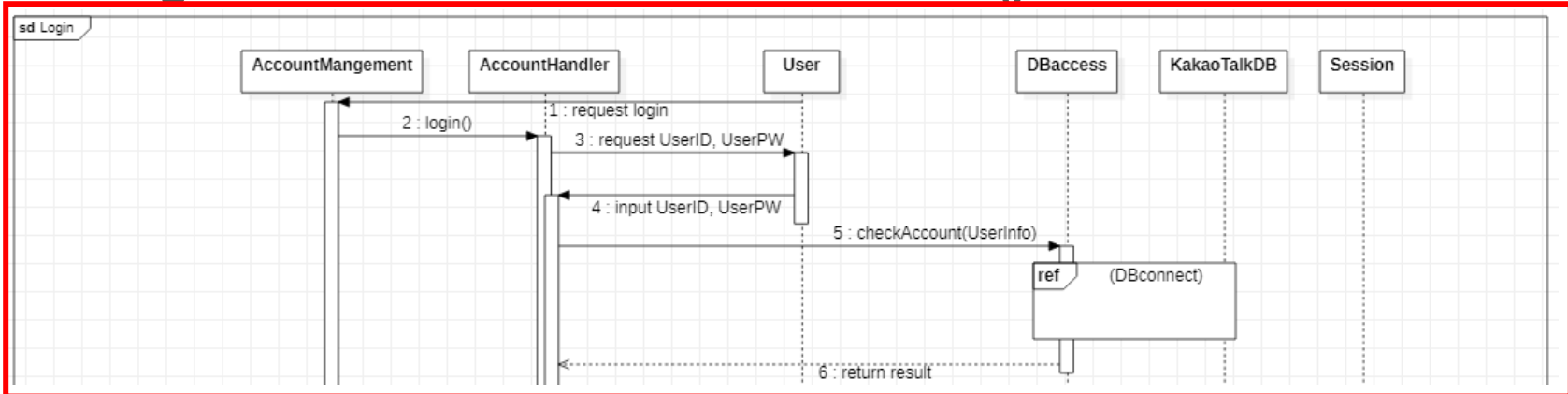


```
1 public class KakaoTalkDB extends DBaccess
2 {
3
4 public boolean getConnection()
5 {
6     boolean result = true;
7     System.out.println("connect complete from kakaotalk database");
8     return result;
9 }
10
11 public UserInfo excuteQuery(UserInfo UserInfo) //메소드,파라미터 명시 x 클래스 다이어그램에도 명시 x
12 {
13     UserInfo result = null; //db에서 가져왔다 가정
14     System.out.println("exeture query successs from kakaotalk db with using UserInfo");
15     return result;
16 }
17
18 public void close()
19 {
20     System.out.println("close success");
21 }
22 }
23
24 }
```

메소드 명시 x
임의로 메서드 구현

시퀀스 다이어그램 분석

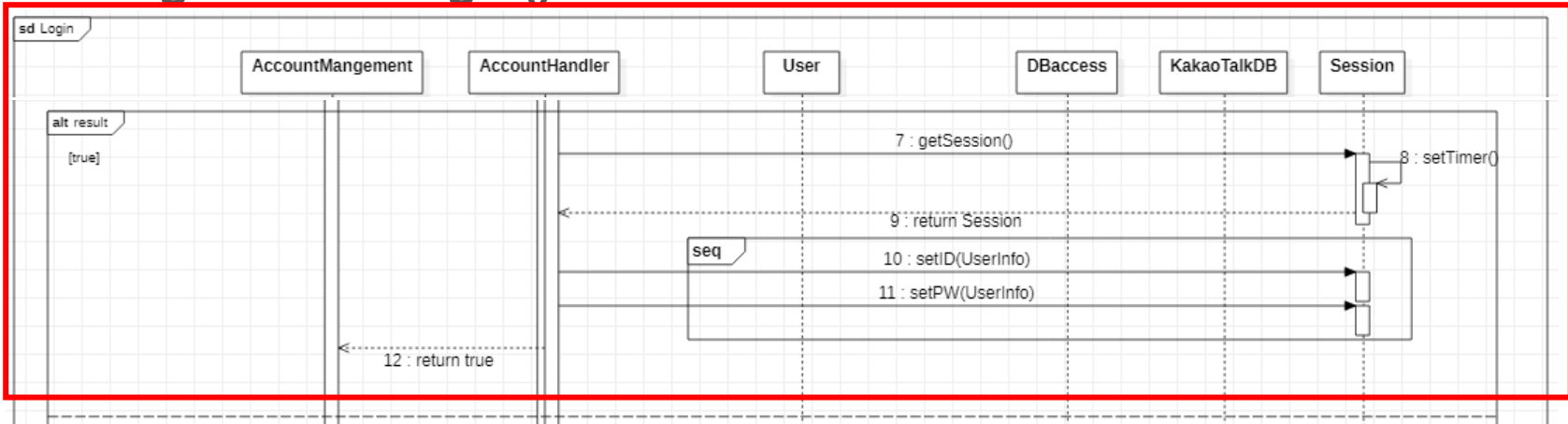
S3. Login 분석 – DBAccess.checkAccount()



```
26 public boolean checkAccount(UserInfo userInfo){
27     KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
28     boolean connection = kakaoTalkDB.getConnection(); //connection 변수 사용 x
29     UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
30     kakaoTalkDB.close();
31     //예외처리 + id, pw 동일한지 체크
32     if(result != null && userInfo.id.equals(result.id) && userInfo.pw.equals(result.pw))
33     {
34         return true;
35     }
36     return false;
37 }
```

시퀀스 다이어그램 분석

S3. Login 분석 – login()

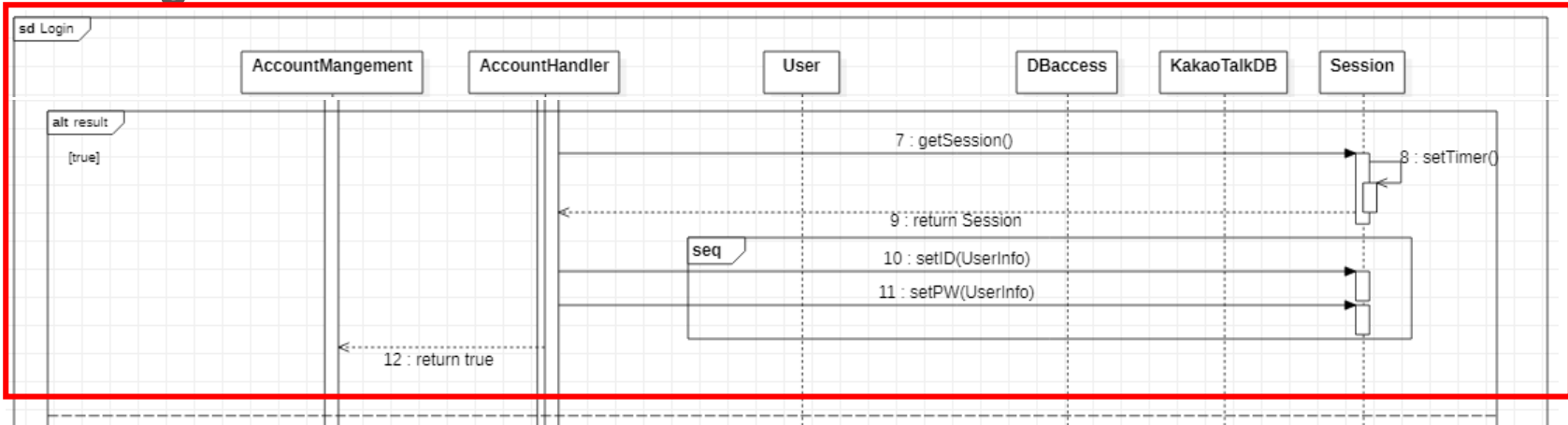


```
4 public class AccountHandler {
5     Session session = new Session();
6     UserInfo userInfo = new UserInfo();
7     DBAccess dbaccess = new DBAccess();
8     public boolean login()
9     {
10         System.out.println("requested login by user\nenter id and password");
11         Scanner sc = new Scanner(System.in);
12         boolean result = true;
13         userInfo.id = sc.next();
14         userInfo.pw = sc.next();
15         result = dbaccess.checkAccount(userInfo);
16         if(result)
17         {
18             session = session.getSession(); //void 형이라고 하심
19             session.setID(userInfo);
20             session.setPW(userInfo);
21         }
22         sc.close();
23         return result; //always return true there is no case that return value can be false
24     }
25 }
```

7,10,11

시퀀스 다이어그램 분석

S3. Login 분석 – Session 메소드



```
public Session getSession()
{
    Session newSession = this; //this = 새로운 세션이라 가정
    System.out.println("세션 연결됨");
    timer = 10; //생성자 없어서 여기서 초기화
    //timer 값은 int 10이라고 명시됨
    setTimer();
    return newSession;
}
```

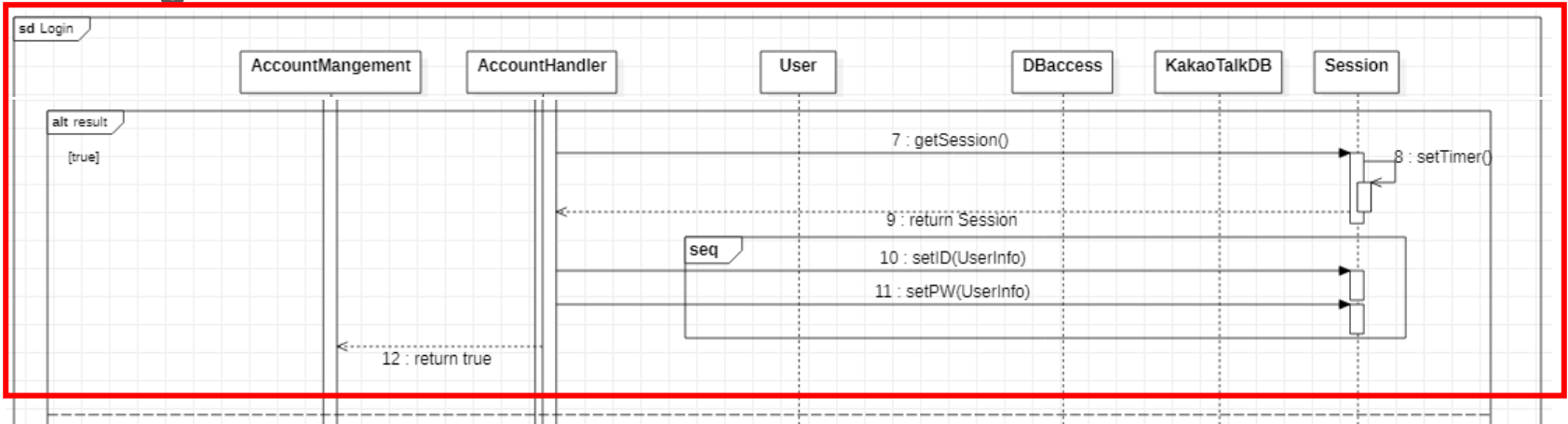
7,8,9

```
public void setTimer()
{
    new Thread()
    {
        Calendar cal;
        int currentMin;
        public void run()
        {
            cal = Calendar.getInstance();
            currentMin = cal.get(Calendar.MINUTE); //현재 분 저장
            System.out.println("TimerSetedMin : "+currentMin);
            System.out.println("max timer set : "+ timer);
            //set은 된 상태이지만 그 이후 동작 없음
        }
    }.start();
}
```

8

시퀀스 다이어그램 분석

S3. Login 분석 – Session 메소드



```
//시퀀스 다이어그램과 클래스 다이어그램 명시된 파라미터 다음 -> 시퀀스를 따르기로함
public void setID(UserInfo userInfo) //시퀀스다이어그램에 반환형없어서 boolean -> void로 바꿈
{
    this.id = userInfo.id;
}
```

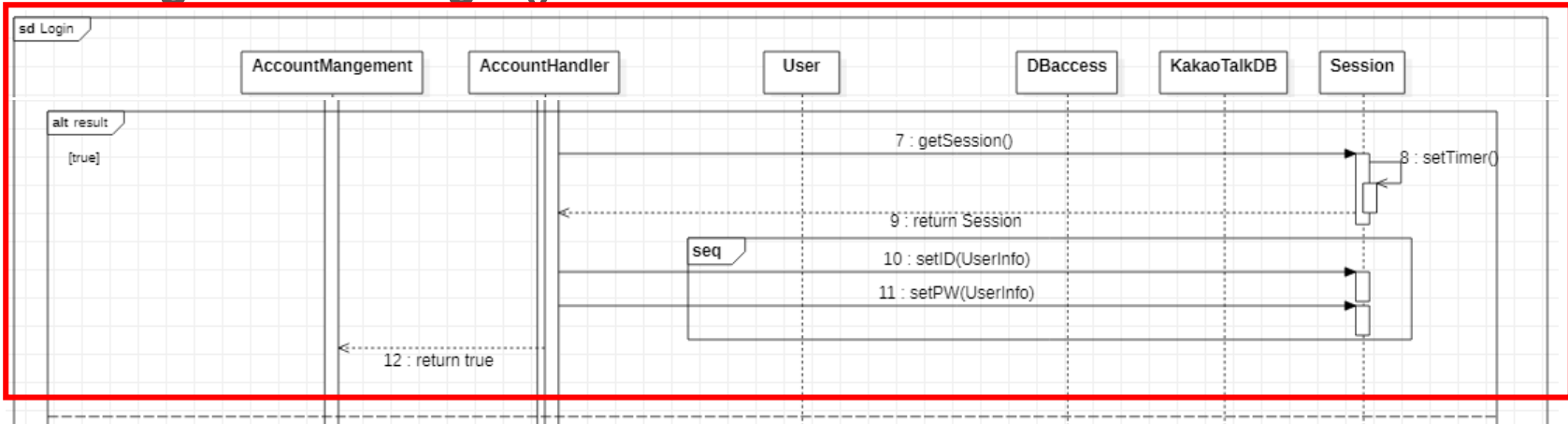
10

```
public void setPW(UserInfo userInfo) //시퀀스다이어그램에 반환형없어서 boolean -> void로 바꿈
{
    this.password = userInfo.id;
}
```

11

시퀀스 다이어그램 분석

S3. Login 분석 – login()



```
4 public class AccountHandler {
5     Session session = new Session();
6     UserInfo userInfo = new UserInfo();
7     DBaccess dbaccess = new DBaccess();
8     public boolean login()
9     {
10         System.out.println("requested login by user\nenter id and password");
11         Scanner sc = new Scanner(System.in);
12         boolean result = true;
13         userInfo.id = sc.next();
14         userInfo.pw = sc.next();
15         result = dbaccess.checkAccount(userInfo);
16         if(result)
17         {
18             session = session.getSession(); //void 형이라고 하심
19             session.setID(userInfo);
20             session.setPW(userInfo);
21         }
22         sc.close();
23         return result; //always return true there is no case that return value can be false
24     }
25 }
```

12

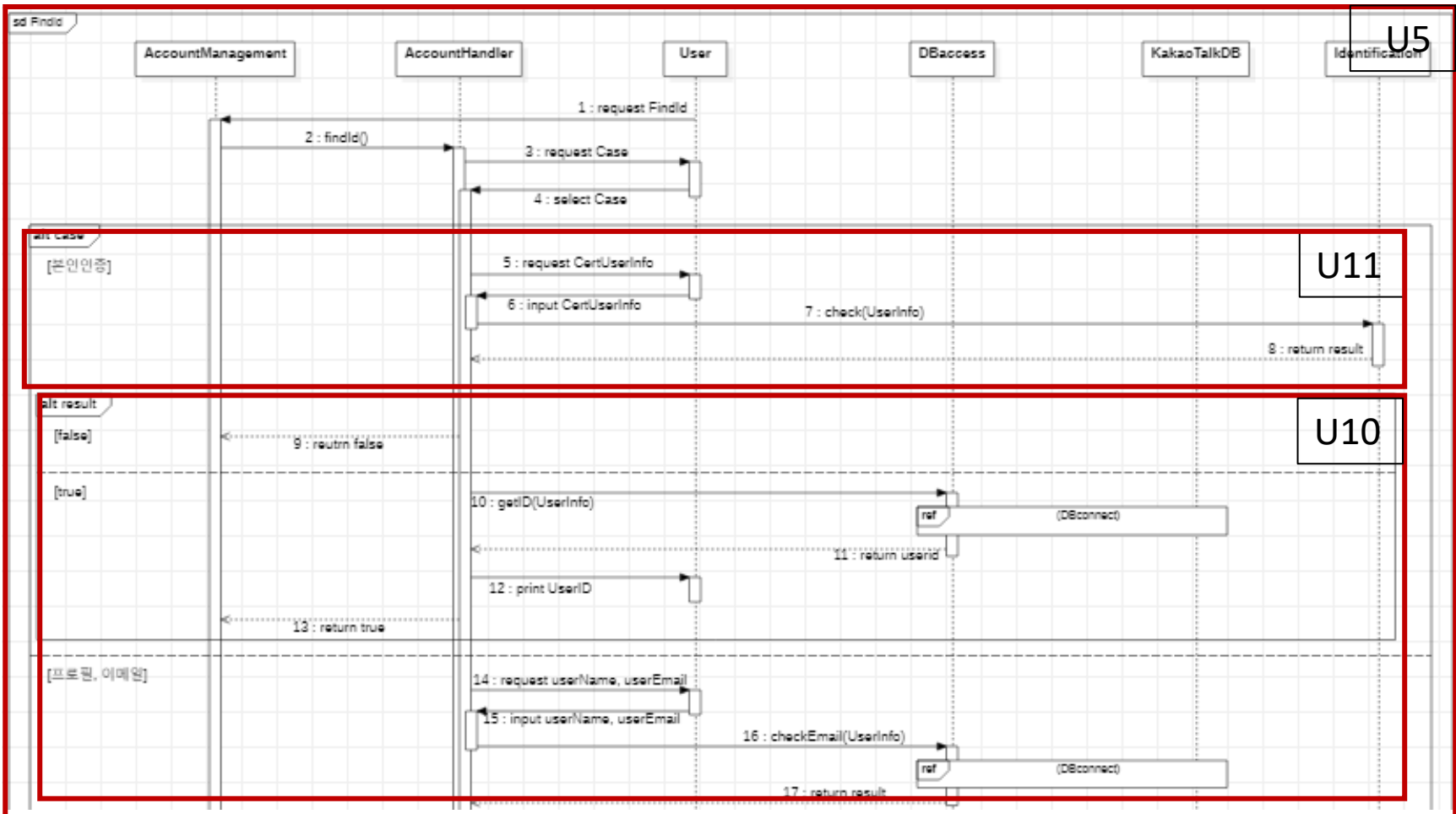
시퀀스 다이어그램 분석

S5. FindID 분석

U5.아이디
찾기

U11.
본인 인증

U10.
회원 확인



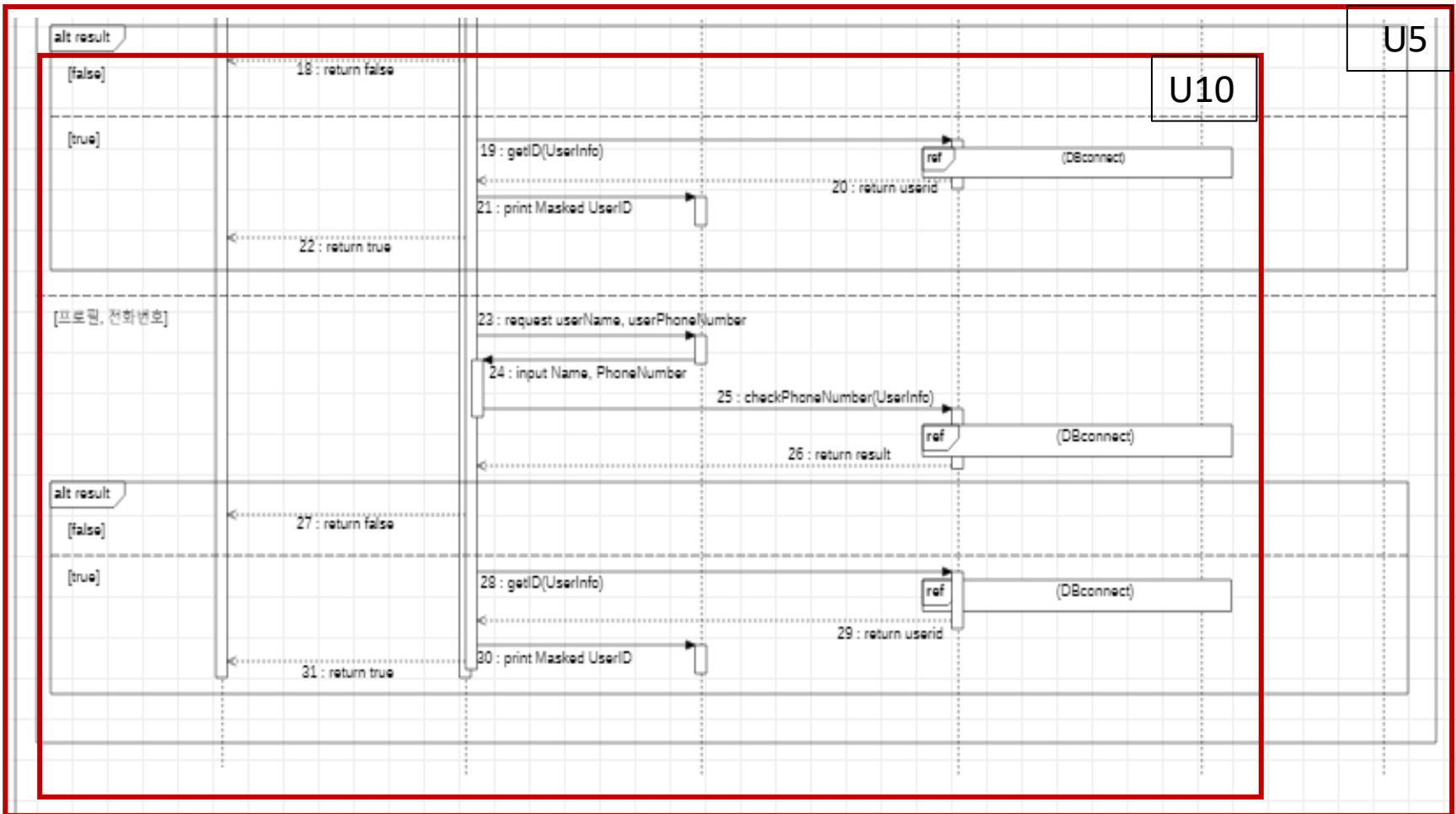
시퀀스 다이어그램 분석

S5. FindID 분석

U5.아이디
찾기

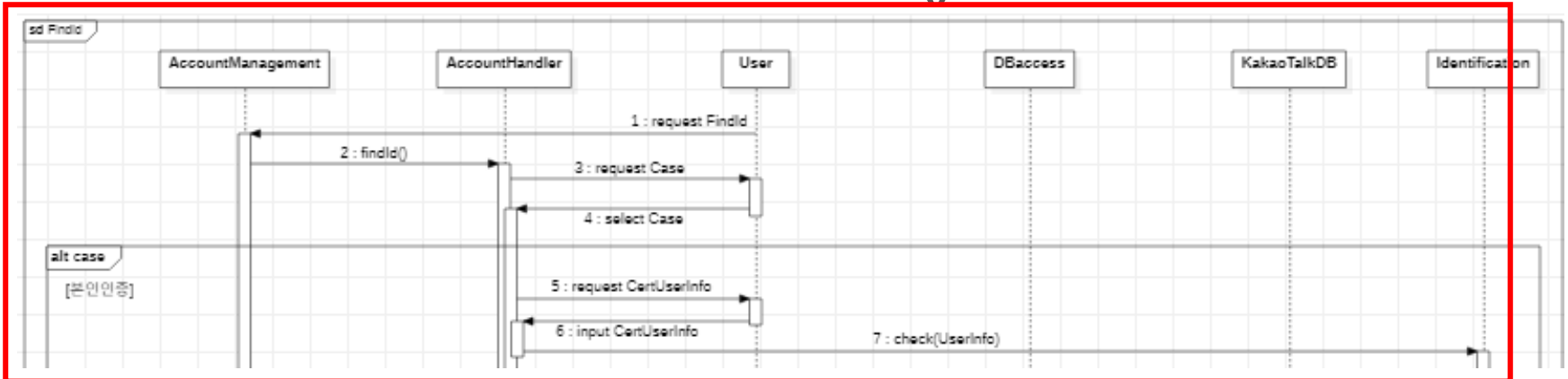
U11.
본인 인증

U10.
회원 확인



시퀀스 다이어그램 분석

S5. FindID 분석 – AccountHandler.findID()

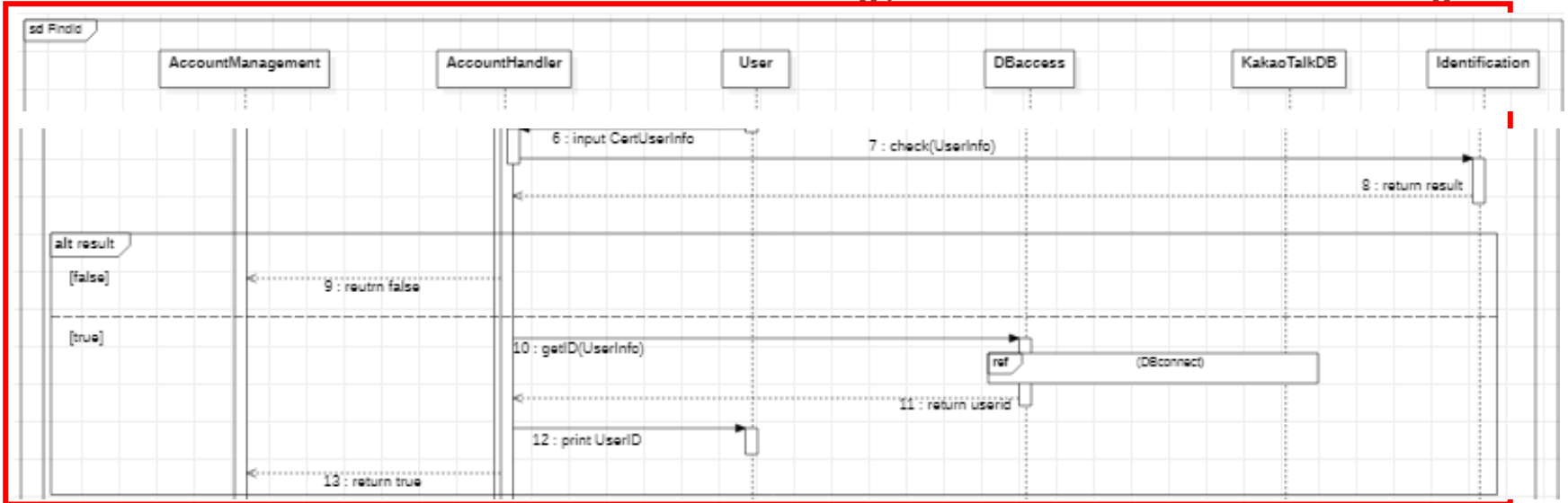


```
41 public boolean findID()
42 {
43
44     String userID;
45     UserInfo inputUserInfo = new UserInfo();
46     Scanner sc = new Scanner(System.in);
47     int userCase = 0;
48     boolean result = false;
49
50     System.out.println("requested login by user");
51     System.out.println("enter user case");
52     userCase = sc.nextInt();
53     System.out.println("enter user name");
54     inputUserInfo.name = sc.next(); //모든 경우에 이름 입력받을
55     //case == 0이면 본인인증
56     if(userCase == 0)
57     {
58         //set이 아닌 get이라는 점에서 필드값을 쓰는게 아니라 판단
59         System.out.println("enter user phone number and birth");
60         inputUserInfo.phoneNumber = sc.next();
61         inputUserInfo.birth = sc.next();
62         result = new Identification().check(inputUserInfo);
```

3,4,5,6,7

시퀀스 다이어그램 분석

S5. FindID 분석 – identification.check(), AccountHandler.findID()



```
public boolean check(UserInfo userInfo)
{
    boolean result = true;
    System.out.println("외부 시스템에서 체크");
    return result;
}
```

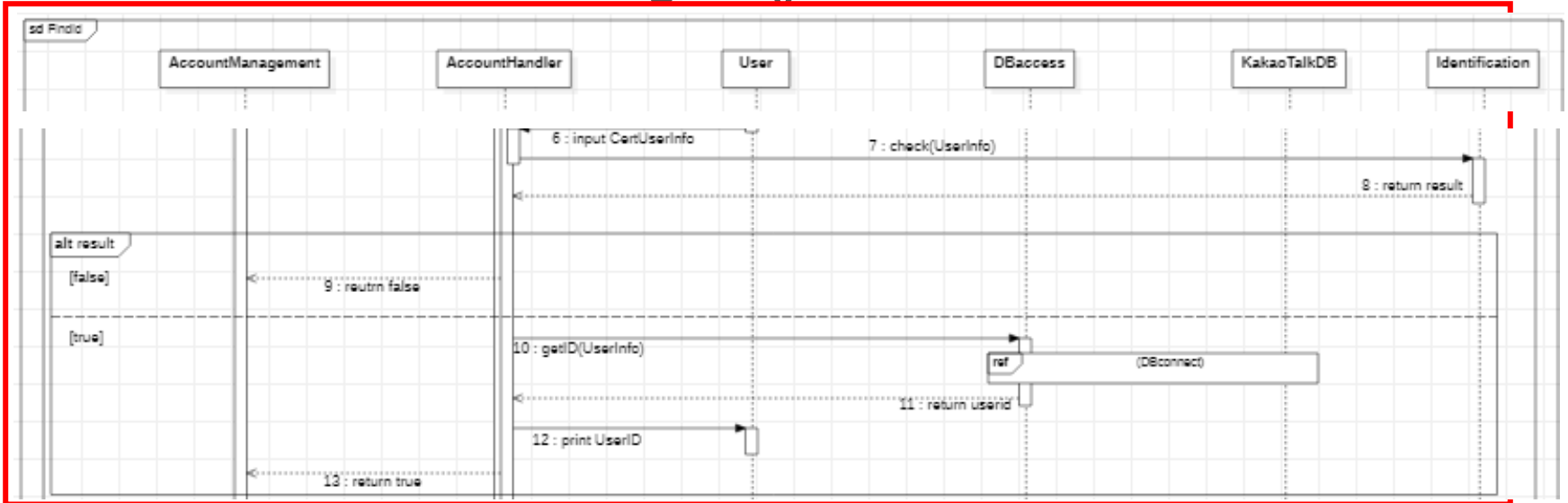
7,8

```
result = new identification().check(inputUserInfo);
if(result)
{
    userID = dbaccess.getID(inputUserInfo);
    System.out.println("userID : "+userID); //null이 반환되면 null출력
}
}
```

10,11,12

시퀀스 다이어그램 분석

S5. FindID 분석 – DBAccess.getID()



```
public String getID(UserInfo userInfo)
```

```
{
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); //변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 연결이갑자기끊겼다면지 등으로 이상하게 받아온경우
    if(result == null)
    {
        return null;
    }
    return result.id;
}
```

Ref : DBconnect

11

시퀀스 다이어그램 분석

S5. FindID 분석 – AccountHandler.findID()



```
else if(userCase == 1)
{
    System.out.println("enter user email");
    inputUserInfo.email = sc.next();
    result = dbaccess.checkEmail(inputUserInfo);
    if(result)
    {
        userID = dbaccess.getID(inputUserInfo);
        System.out.println("userID : "+userID); //null이 반환되면 null출력
    }
}
```

15,16,17

시퀀스 다이어그램 분석

S5. FindID 분석 – DBaccess.checkEmail()



```
public boolean checkEmail(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); //변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리없지만 예외처리추가
    if (result == null || !(result.email.equals(userInfo.email)))
    {
        return false;
    }
    return true;
}
```

Ref : DBconnect

17

시퀀스 다이어그램 분석

S5. FindID 분석 – AccountHandler.findID()

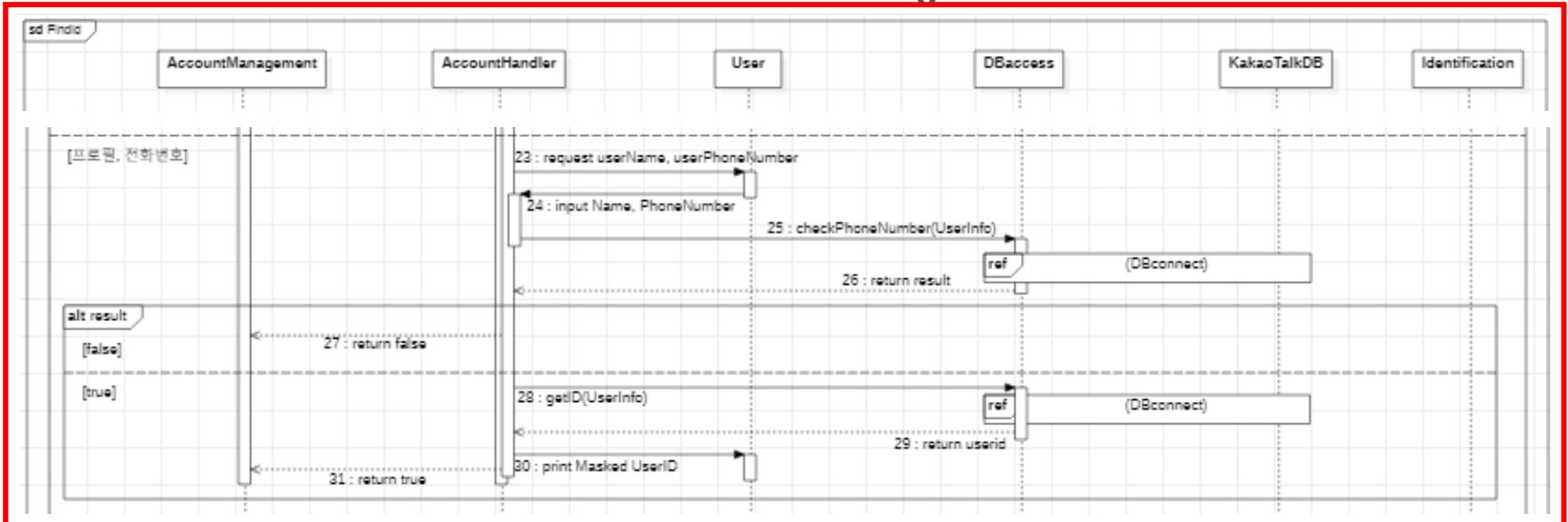


```
else if(userCase == 1)
{
    System.out.println("enter user email");
    inputUserInfo.email = sc.next();
    result = dbaccess.checkEmail(inputUserInfo);
    if(result)
    {
        userID = dbaccess.getID(inputUserInfo);
        System.out.println("userID : "+userID); //null이 반환되면 null출력
    }
}
```

19,20,21

시퀀스 다이어그램 분석

S5. FindID 분석 – AccountHandler.findID()



```
else
{
    System.out.println("enter user phoneNumber");
    inputUserInfo.phoneNumber = sc.next();
    result = dbaccess.checkPhoneNumber(inputUserInfo);
    if(result)
    {
        userID = dbaccess.getID(inputUserInfo);
        System.out.println("userID : "+userID); //null이 반환되면 null출력
    }
}

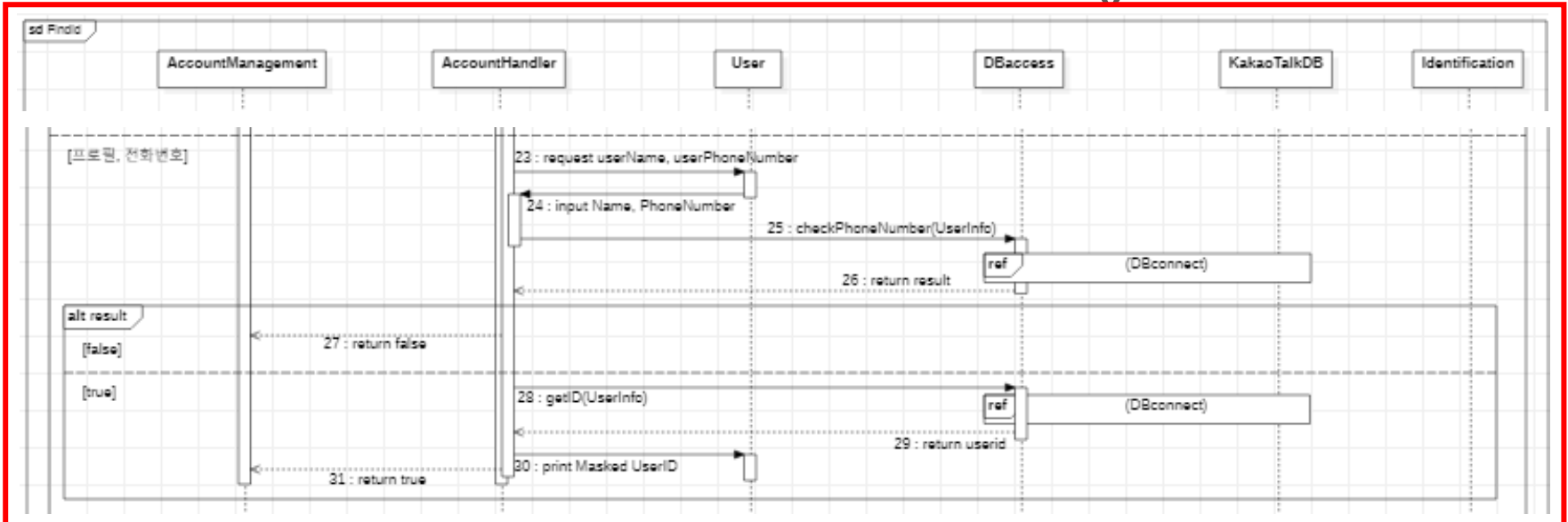
sc.close();
return result;
```

23,24

25,26

시퀀스 다이어그램 분석

S5. FindID 분석 – Dbaccess.checkPhoneNumber()



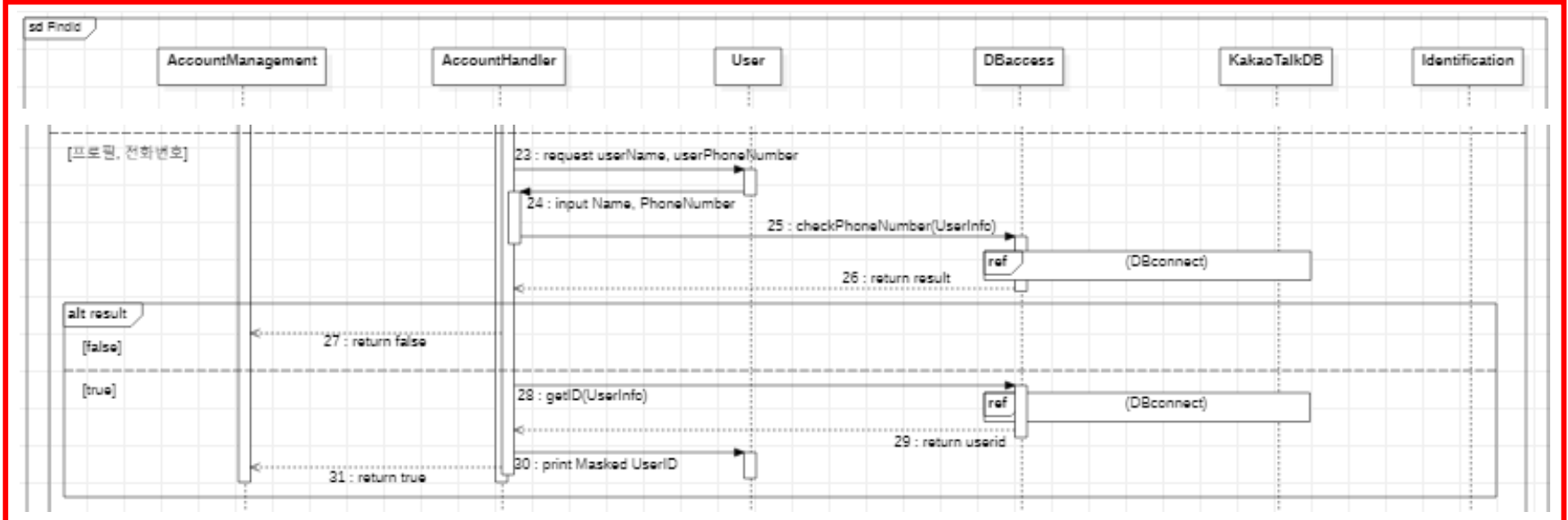
```
public boolean checkPhoneNumber(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); //변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리없지만 예외처리추가
    if (result == null || !(result.email.equals(userInfo.phoneNumber)))
    {
        return false;
    }
    return true;
}
```

Ref : DBconnect

26

시퀀스 다이어그램 분석

S5. FindID 분석 – AccountHandler.findID()



```
else
{
    System.out.println("enter user phoneNumber");
    inputUserInfo.phoneNumber = sc.next();
    result = dbaccess.checkPhoneNumber(inputUserInfo);
    if(result)
    {
        userID = dbaccess.getID(inputUserInfo);
        System.out.println("userID : "+userID); //null이 반환되면 null출력
    }
}

sc.close();
return result;
```

28,29,30

31을 비롯한
AccountHandler에서
AccountManagment로 반환

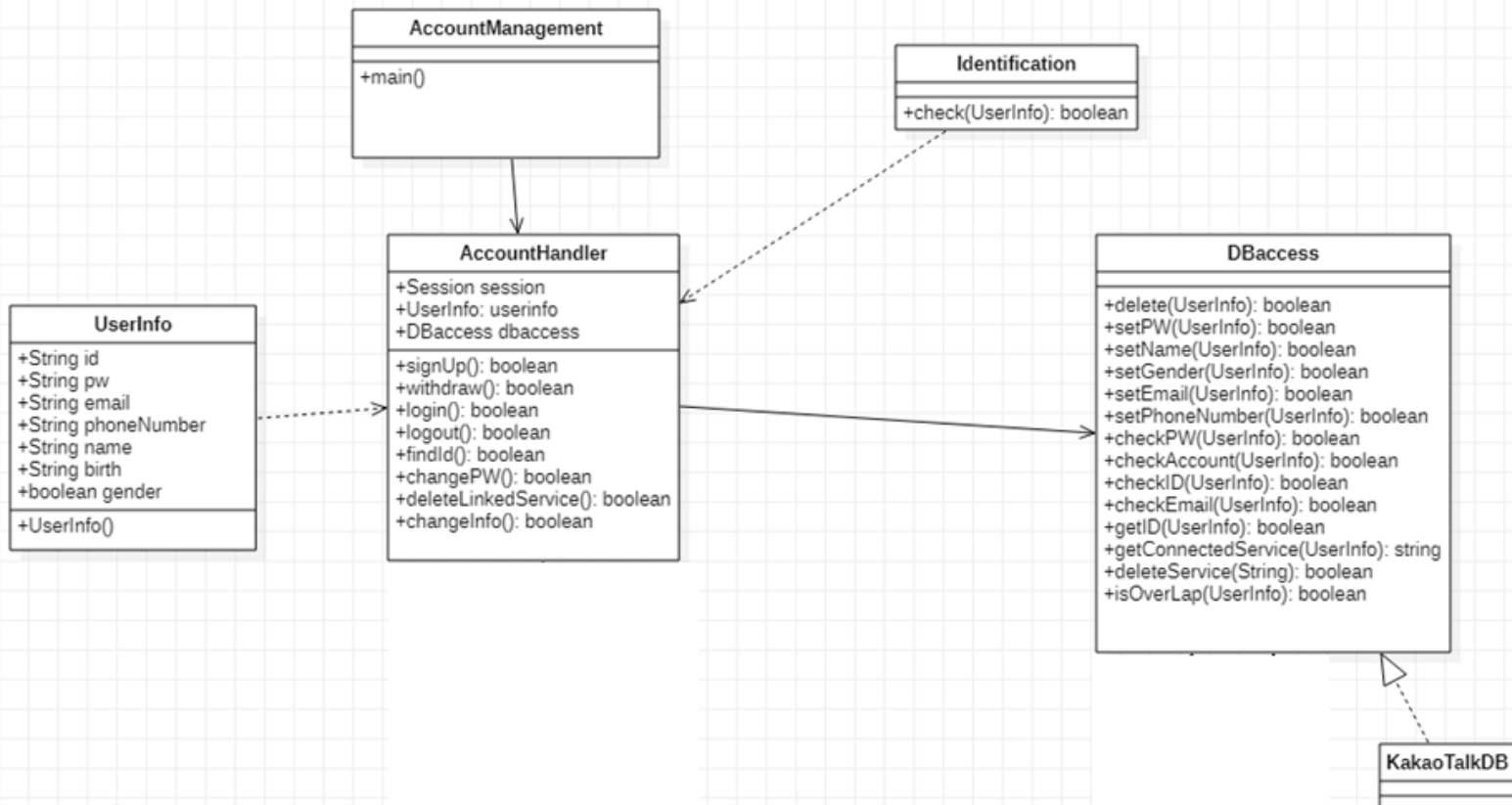
S6.ChangePW 유스케이스

유스케이스

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"> 1. 사용자는 비밀번호 재설정을 요청한다. 2. 사용자는 아이디를 입력한다. 3. 카카오톡 DB 시스템은 기존 가입 회원인지 확인한다. 4. 기존 가입 회원이라면 본인인증을 요청한다. 5. 외부 인증 시스템은 본인인증을 확인한다. 6. 비밀번호를 재설정한다. 		
선택 흐름	<p>✓ 회원 확인 이후 해당하는 계정이 없으면 "입력하신 정보와 일치하는 계정이 없습니다" 경고창을 출력한다.</p>		

S6.ChangePW 필요 클래스

필요 클래스



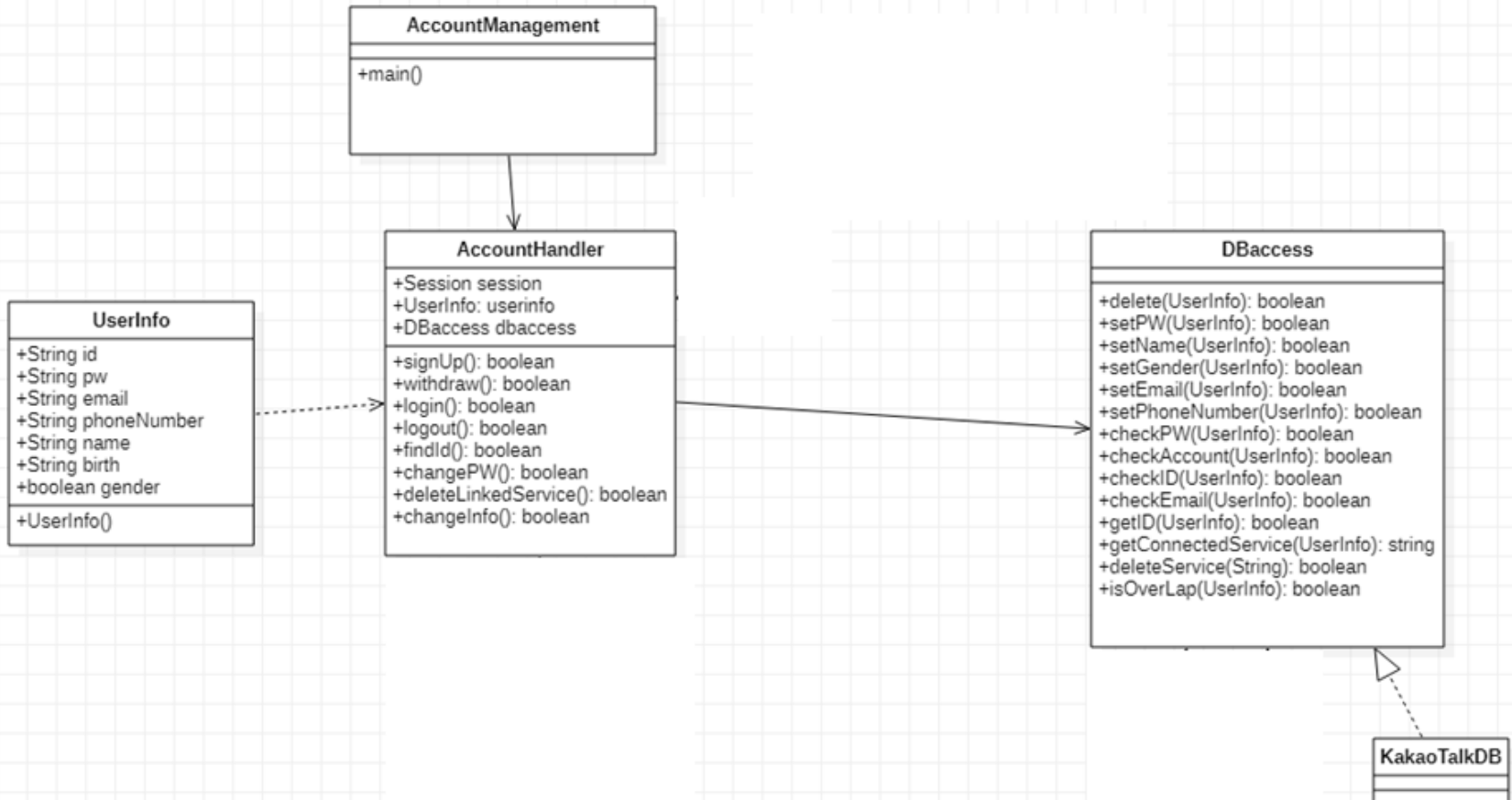
S7.ChangeInfo 유스케이스

유스케이스

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

S7.ChangeInfo 필요 클래스

필요 클래스



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

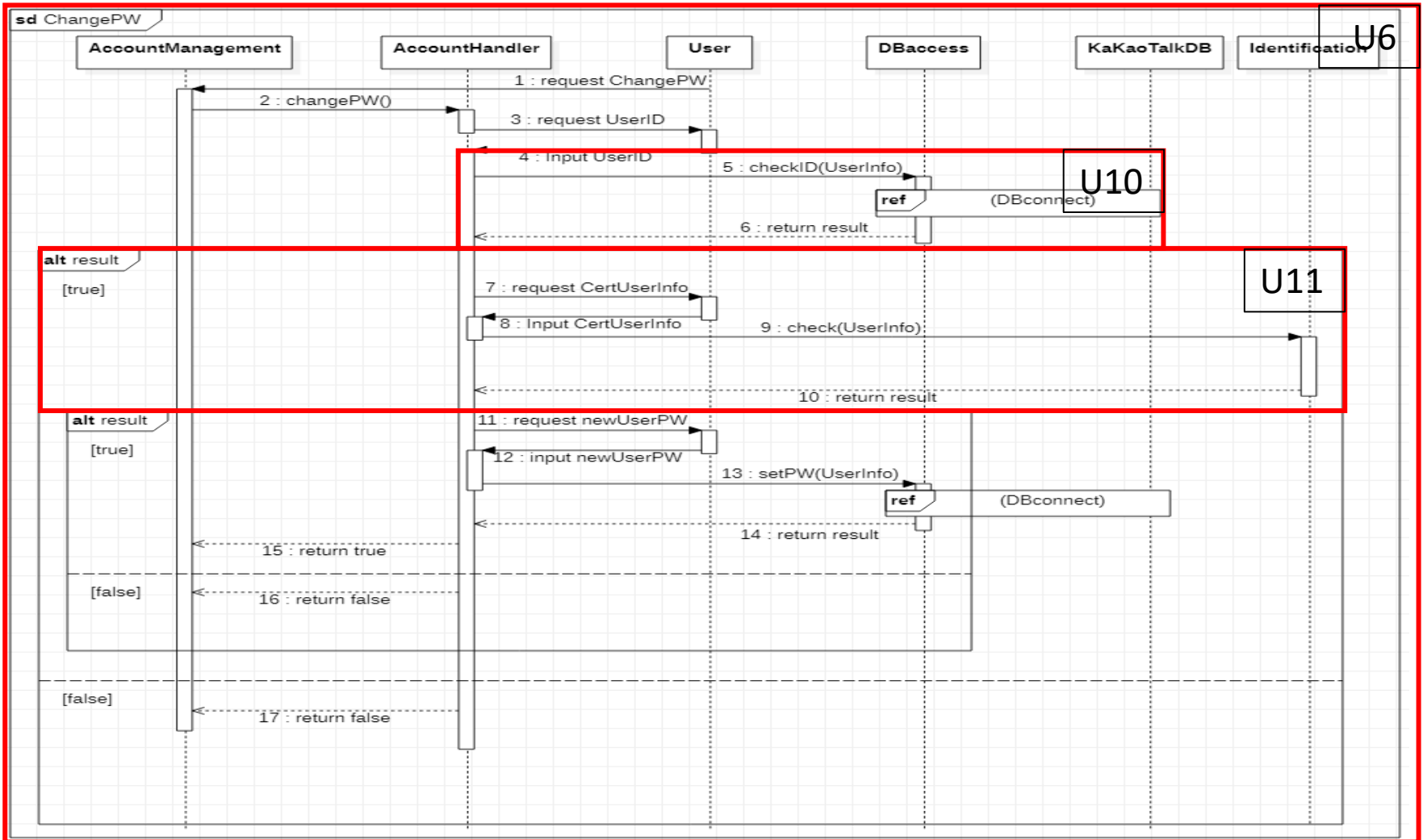
시퀀스 다이어그램 분석

S6. ChangePW 분석

U6.
비밀번호
변경

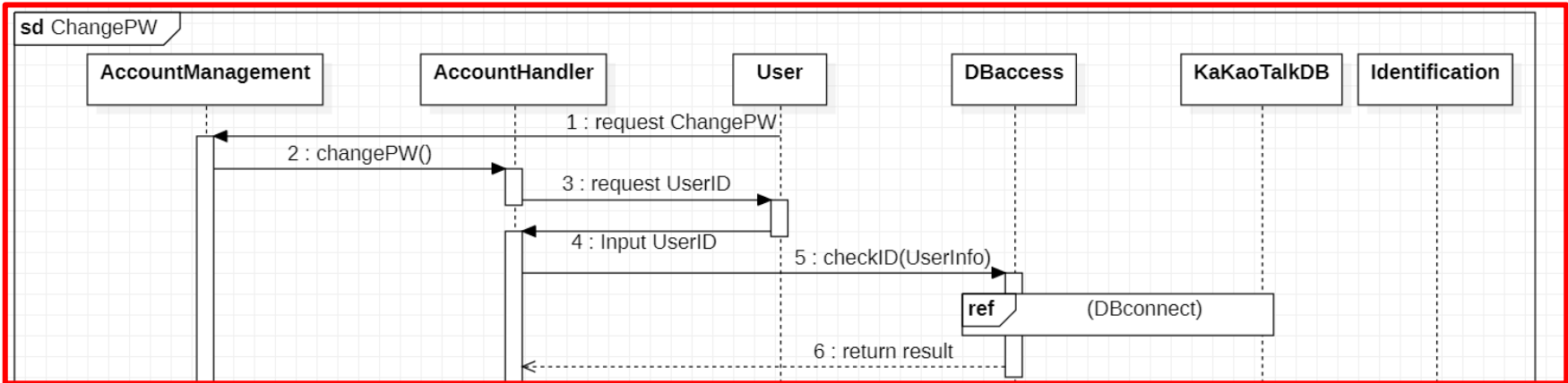
U10.
회원 확인

U11.
본인 인증



시퀀스 다이어그램 분석

S6. ChangePW 분석 – AccountHandler.changePW(), DBaccess.checkID()



```
public boolean changePW() {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("ID : ");  
    userInfo.id = sc.next();  
    if (dbaccess.checkID(userInfo))
```

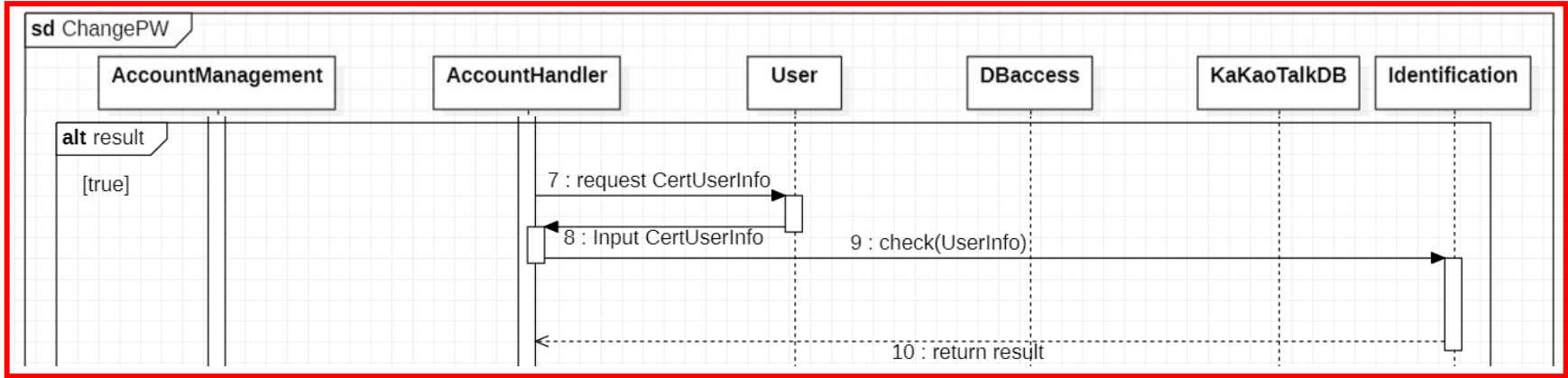
3,4,5,6

```
public boolean checkID(UserInfo userInfo){  
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();  
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀  
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);  
    kakaoTalkDB.close();  
    // 예외처리 없지만 예외처리 추가  
    if (result != null && userInfo.id.equals(result.id)) {  
        return true;  
    }  
    else return false;  
}
```

5,6

시퀀스 다이어그램 분석

S6. ChangePW 분석 – AccountHandler.changePW(), identification.check()



```
if (dbaccess.checkID(userInfo))
```

alt result // true

```
{
```

```
System.out.println("Name : ");
```

```
userInfo.name = sc.next();
```

```
System.out.println("Phone Number : ");
```

```
userInfo.phoneNumber = sc.next();
```

```
System.out.println("Identification Number : ");
```

```
userInfo.birth = sc.next();
```

```
if (new identification().check(userInfo))
```

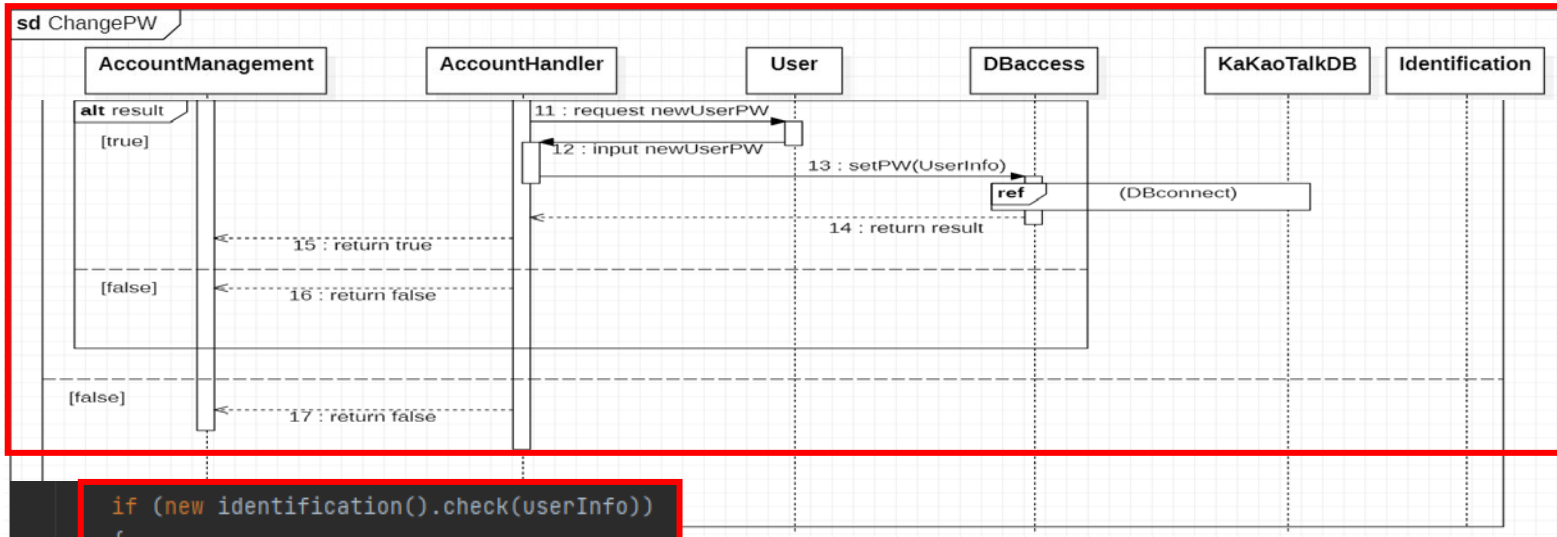
7,8

9,10

```
public boolean check(UserInfo userInfo)
{
    boolean result = true;
    System.out.println("외부 시스템에서 체크");
    return result;
}
```

시퀀스 다이어그램 분석

S6. ChangePW 분석 – AccountHandler.changePW()



```
if (new identification().check(userInfo))
{
    System.out.println("New PW : ");
    userInfo.pw = sc.next();
    sc.close();
    return dbaccess.setPW(userInfo);
} else
{
    sc.close();
    return false;
} else
{
    sc.close();
    return false;
}
```

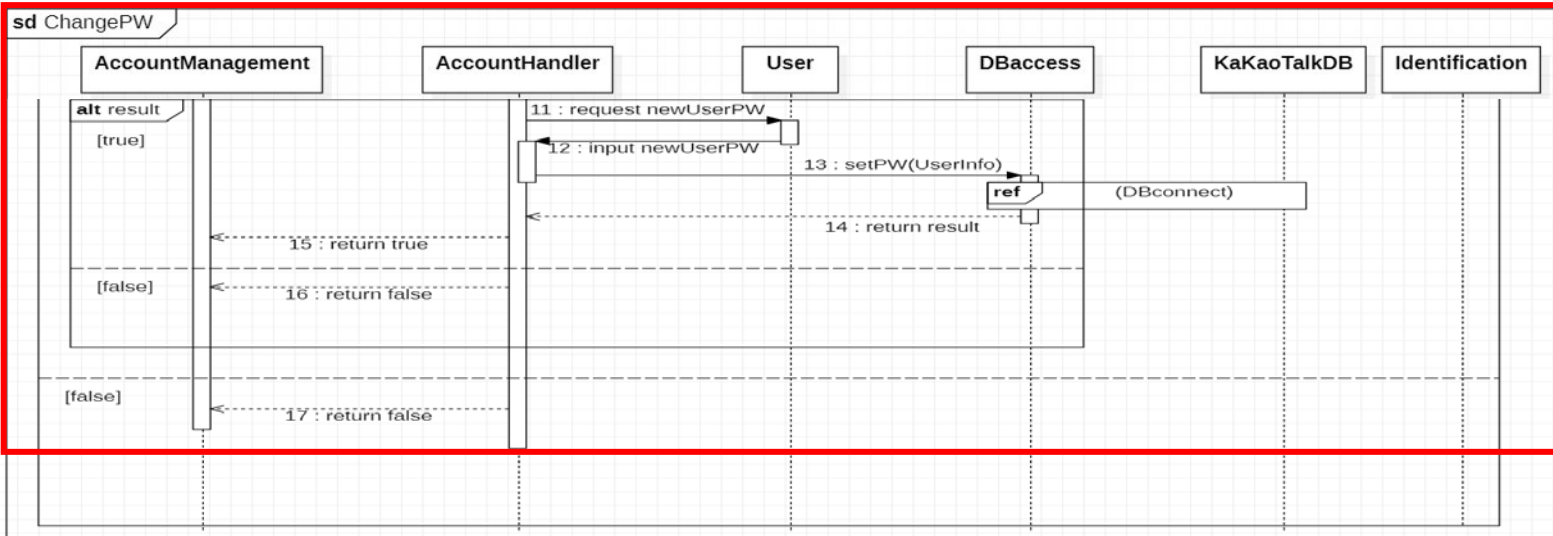
11,12,13,14,15

16 // 본인인증 실패

17 // 회원인증 실패

시퀀스 다이어그램 분석

S6. ChangePW 분석 – DBaccess.setPW()



```
public boolean setPW(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection();
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 없지만 추가
    if(result == null) return false;
    else return true;
}
```

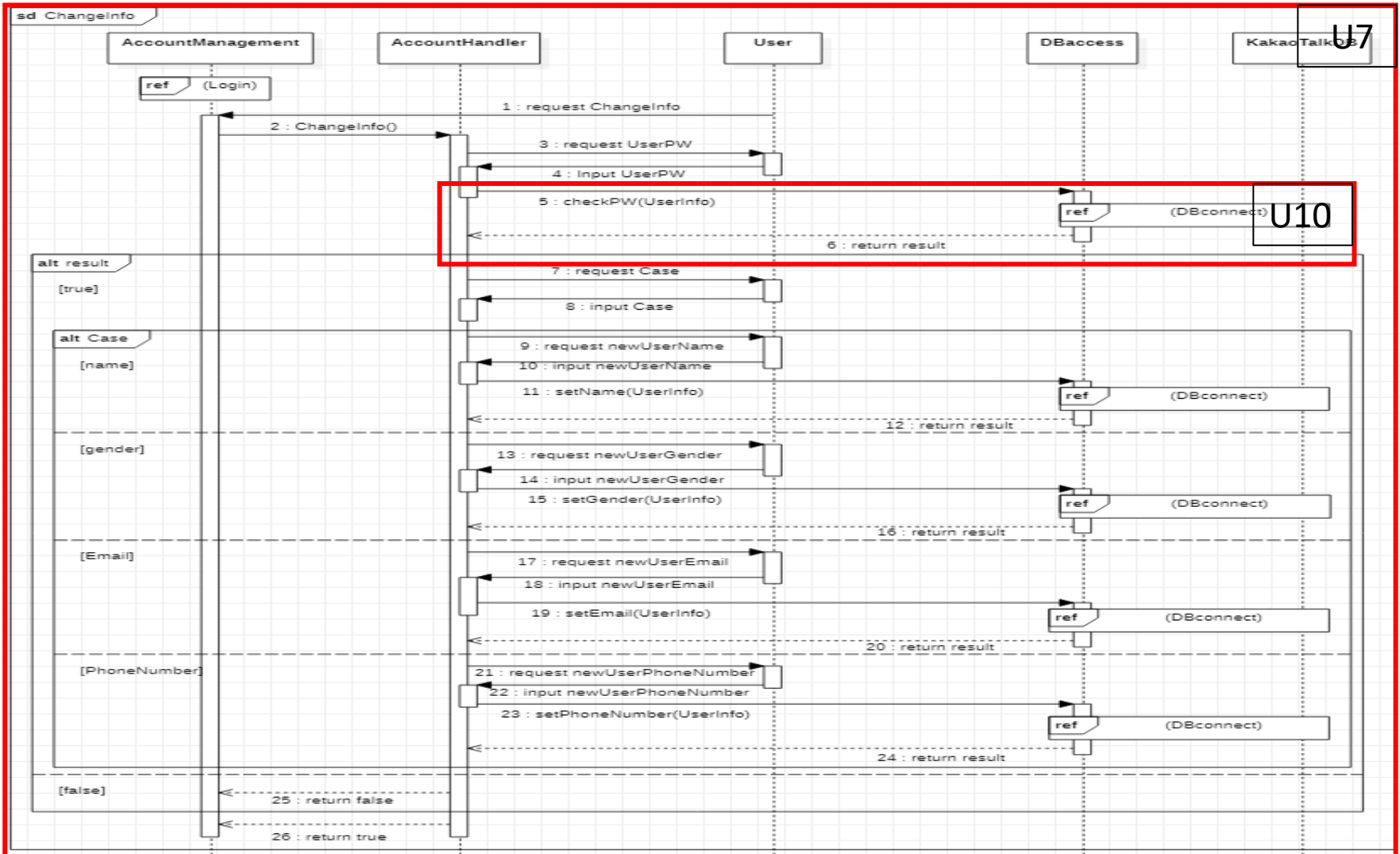
13,14

시퀀스 다이어그램 분석

S7. ChangeInfo 분석

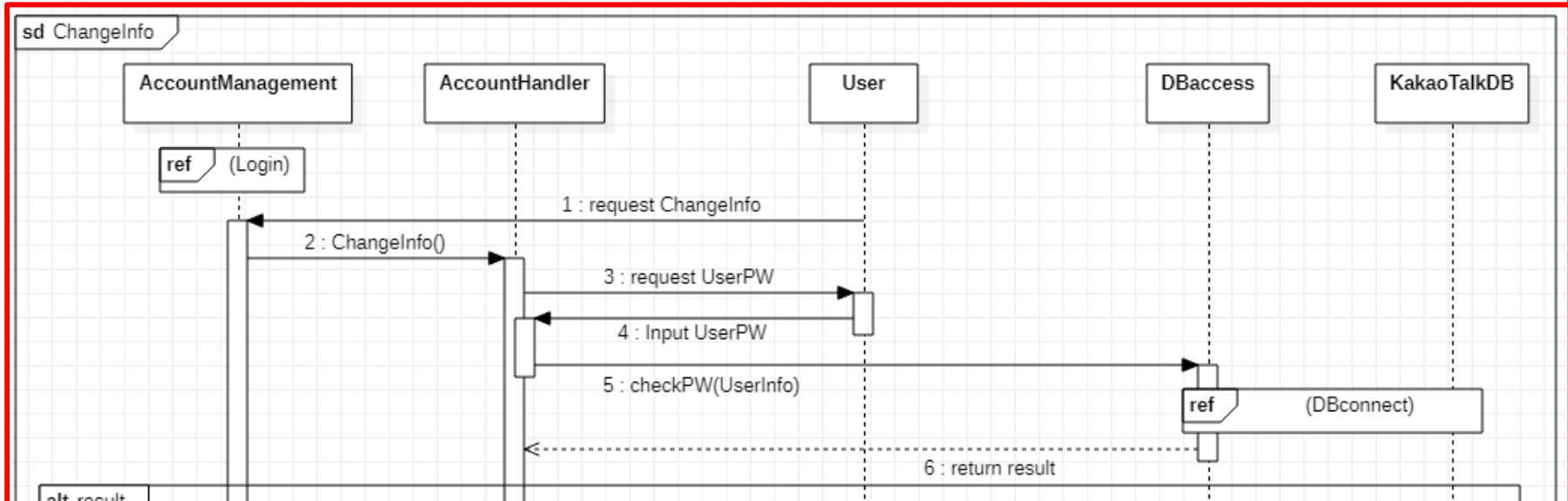
U7.
정보 수정

U10.
회원 확인



시퀀스 다이어그램 분석

S7. ChangeInfo 분석 - AccountHandler.changePW(), DBAccess.checkPW()



```
public boolean changeInfo() {
    Scanner sc = new Scanner(System.in);
    int command = 0;
    System.out.println("PW : ");
    userInfo.pw = sc.next();
    if (dbaccess.checkPW(userInfo))
```

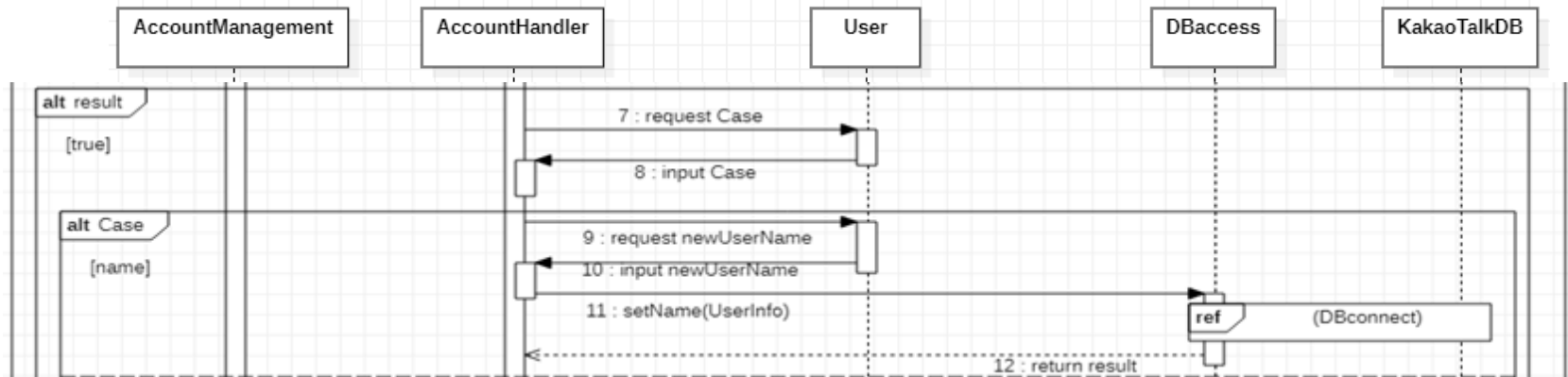
3,4,5,6

```
public boolean checkPW(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 없지만 추가
    if (result != null && userInfo.pw.equals(result.pw)) {
        return true;
    }
    else return false;
}
```

5,6

시퀀스 다이어그램 분석

S7. ChangeInfo 분석 - AccountHandler.changePW(),DBaccess.setName()



```
if (dbaccess.checkPW(userInfo))
```

```
{
    while(true)
    {
        → 7,8
```

```
        System.out.println("1. Change Name");
        System.out.println("2. Change Gender");
        System.out.println("3. Change Email");
        System.out.println("4. Change PhoneNumber");
        System.out.println("What to do : ");
        command = sc.nextInt();
```

```
        switch (command)
```

```
        {
            case 1: // 이름 변경 → 9,10
```

```
                System.out.println("new Name : ");
                userInfo.name = sc.next();
                sc.close();
```

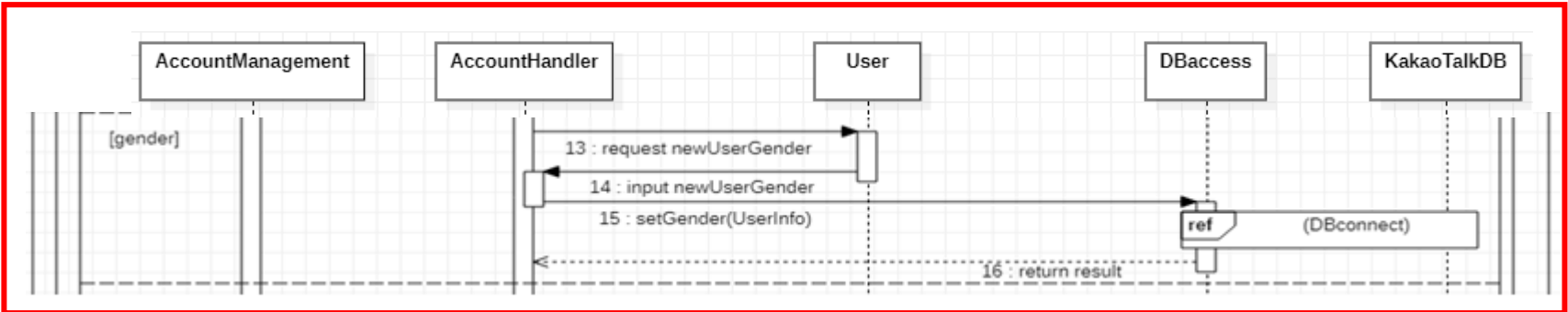
```
                return dbaccess.setName(userInfo); → 11,26
```

```
public boolean setPW(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection();
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 없지만 추가
    if(result == null) return false;
    else return true;
}
```

11,12

시퀀스 다이어그램 분석

S7. ChangeInfo 분석 - AccountHandler.changePW(),DBAccess.checkPW()



```

case 2:    // 성별 변경
System.out.println("new Gender(남/여) : ");
String strGender = sc.next();
while (true) {
    if (strGender.equals("남")) {
        userInfo.gender = true;
        return dbaccess.setGender(userInfo);
    } else if (strGender.equals("여")) {
        userInfo.gender = false;
        return dbaccess.setGender(userInfo);
    } else {
        System.out.println("Error !");
        System.out.println("Please select one of them ( 남 / 여 )");
    }
}
}
    
```

```

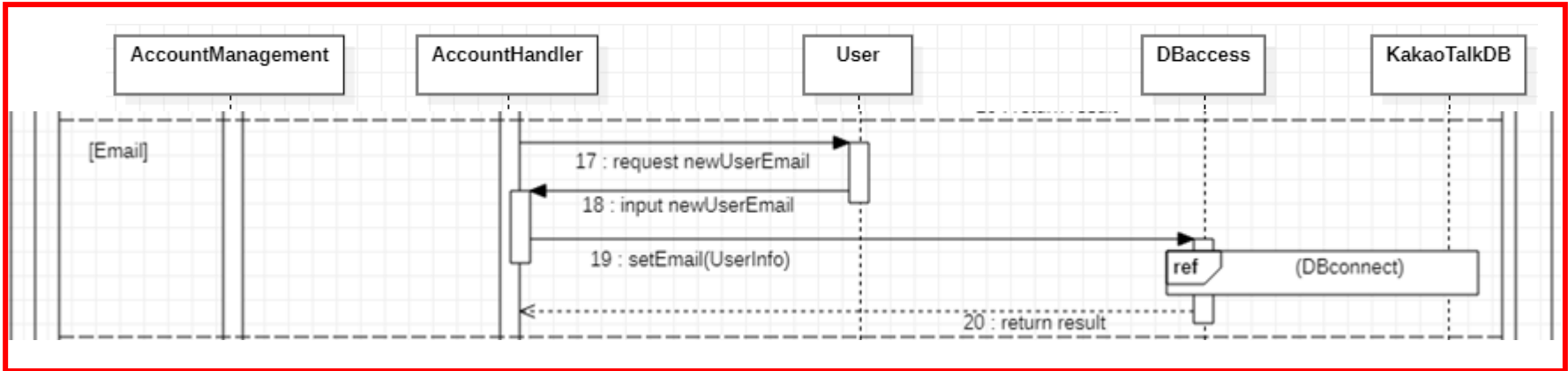
public boolean setGender(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection();
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 없지만 추가
    if(result == null) return false;
    else return true;
}
    
```

15,16

13,14,15,26

시퀀스 다이어그램 분석

S7. ChangeInfo 분석 - AccountHandler.changePW(), DBAccess.checkPW()



```
case 3: // Email 변경
    System.out.println("new Email : ");
    userInfo.email = sc.next();
    sc.close();
    return dbaccess.setEmail(userInfo);
```

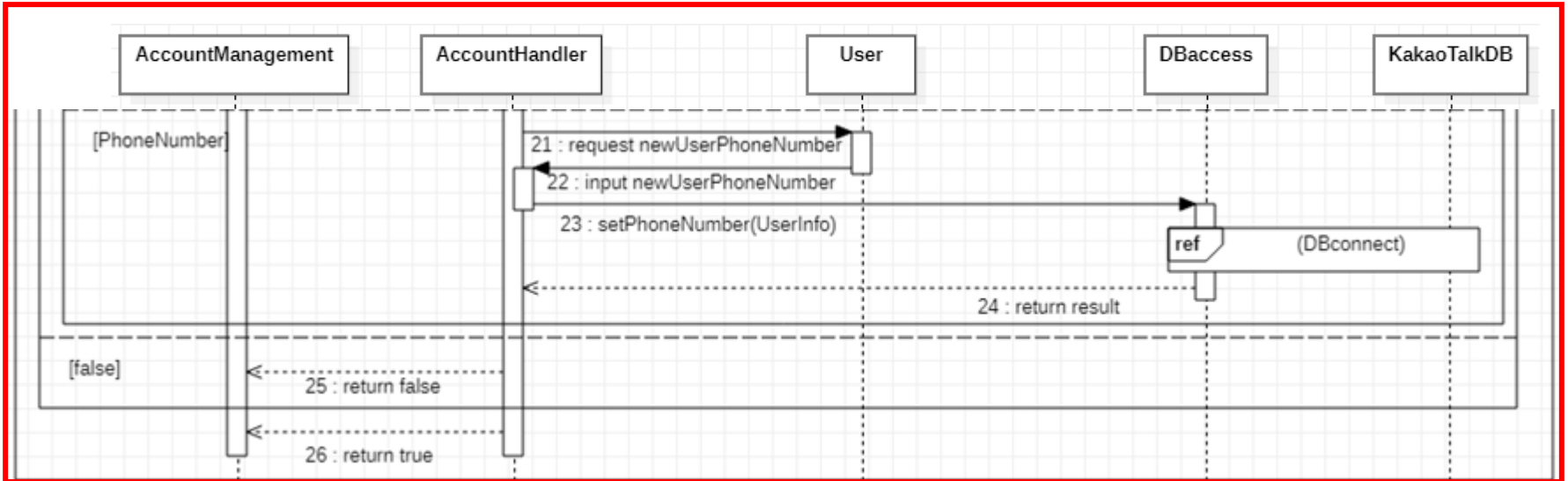
17,18,19,26

```
public boolean setEmail(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection();
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 없지만 추가
    if(result == null) return false;
    else return true;
}
```

19,20

시퀀스 다이어그램 분석

S7. ChangeInfo 분석 - AccountHandler.changePW(),DBAccess.checkPW()



```
case 4: // 전화번호 변경
    System.out.println("new PhoneNumber : ");
    userInfo.phoneNumber = sc.next();
    sc.close();
    return dbaccess.setPhoneNumber(userInfo);
default :
    System.out.println("error!");
}
```

21,22,23,26

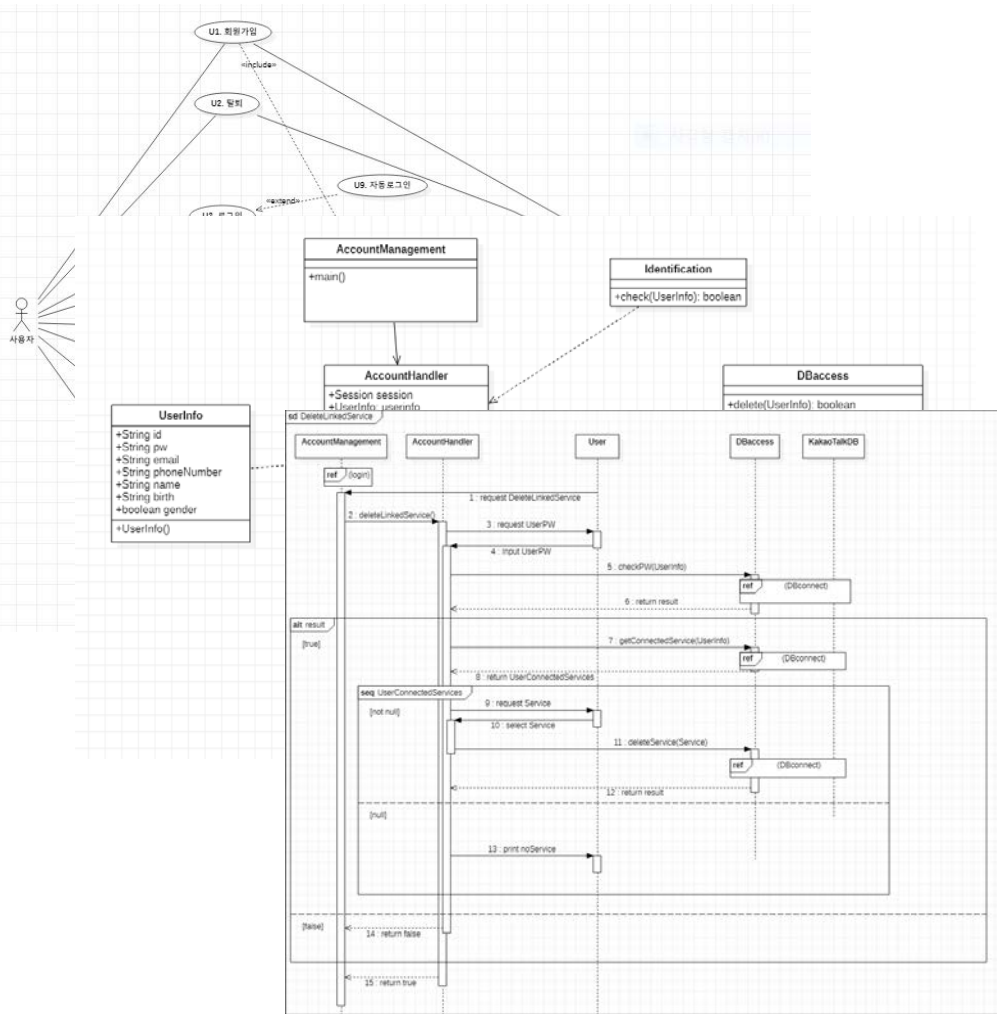
```
}else
{
    sc.close();
    return false;
}
```

25

```
public boolean setPhoneNumber(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection();
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 없지만 추가
    if(result == null) return false;
    else return true;
}
```

23,24

구현 전 분석



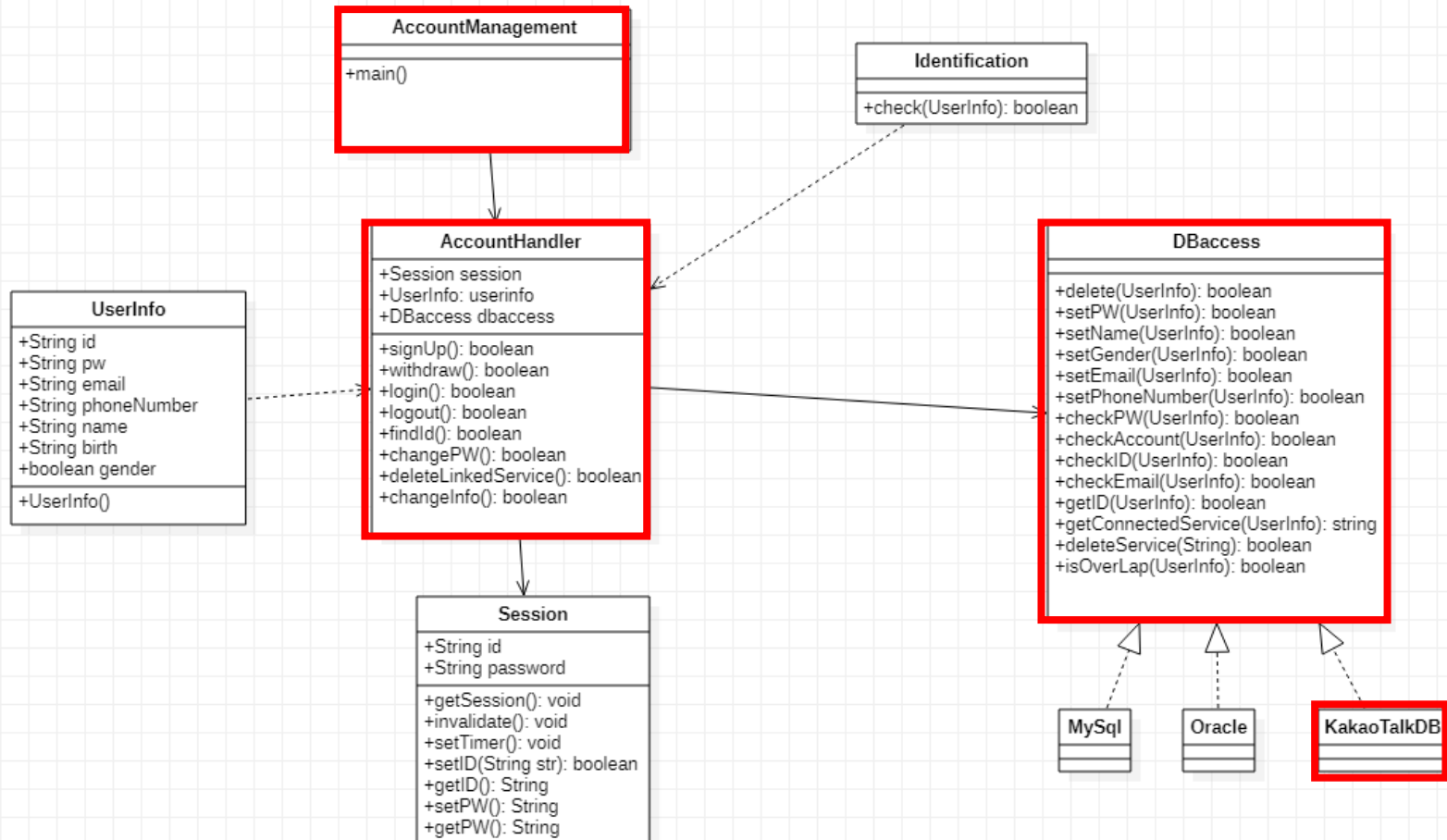
1. 설계도 수신
2. 구현 전 필요 작업 확인
3. 작업 시작

S8. DeleteLinkService

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">1. 사용자는 연결 서비스 삭제를 요청한다.2. 사용자는 사용자 아이디에 해당하는 비밀번호를 입력한다.3. 카카오톡 DB 시스템은 사용자가 입력한 비밀번호가 유효한지 확인한다.4. 사용자가 입력한 비밀번호가 유효하면 DB 접근 시스템은 사용자 계정에 연결된 서비스들을 출력한다.5. 사용자는 연결된 서비스들 중 삭제할 서비스를 선택한다.6. 카카오톡 DB 시스템은 사용자가 선택한 서비스를 삭제한다.		
선택 흐름	<ul style="list-style-type: none">✓ 카카오톡 DB 시스템에서 사용자가 입력한 비밀번호가 유효하지 않다면 종료한다.✓ 카카오톡 DB 시스템에서 사용자 계정에 연결된 서비스가 없다면 연결된 서비스가 없다는 문구를 출력한다.		

S8. DeleteLinkService 필요 클래스

필요 클래스



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

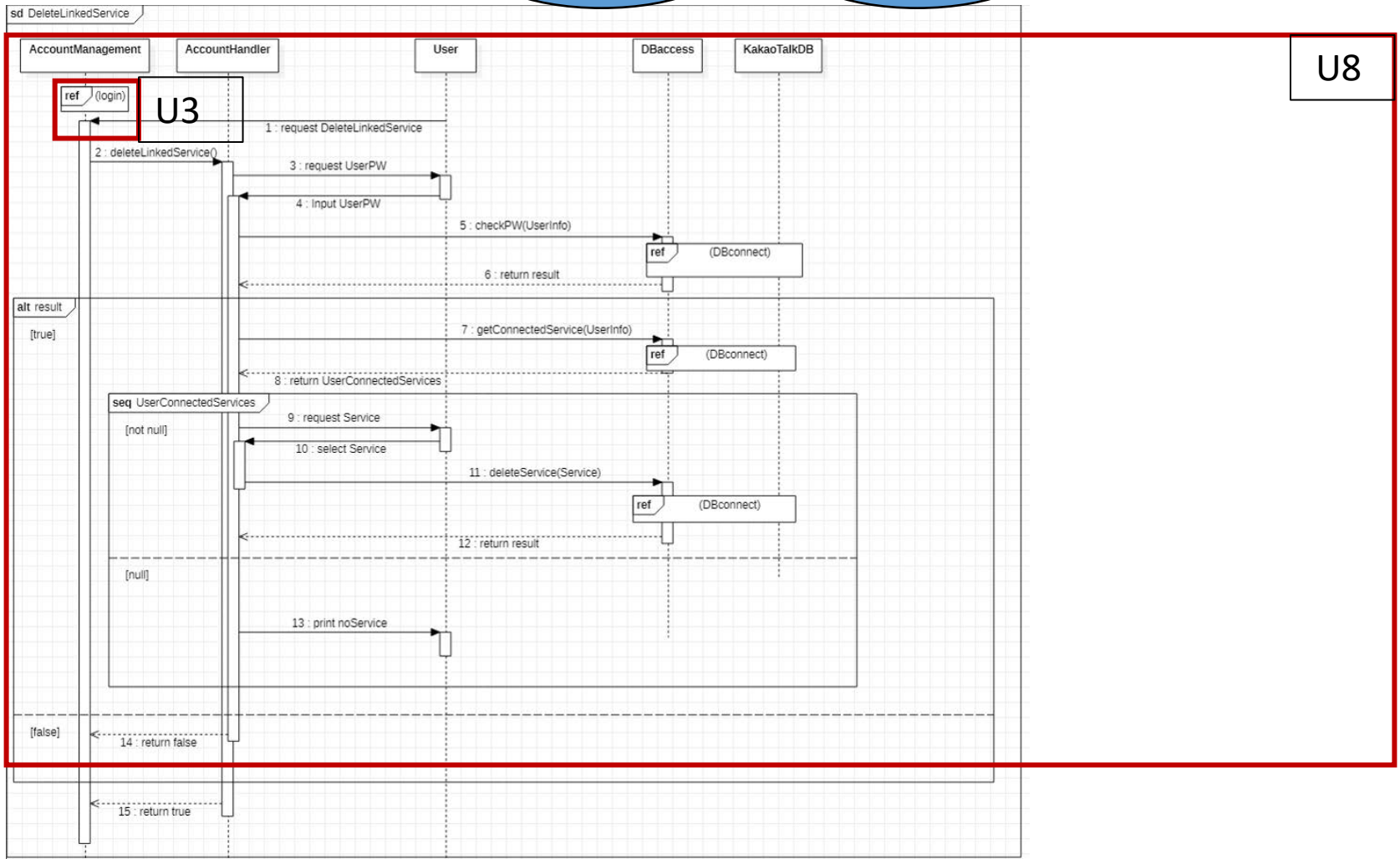
U11.본인 인증

시퀀스 다이어그램 분석

S8. 서비스 삭제

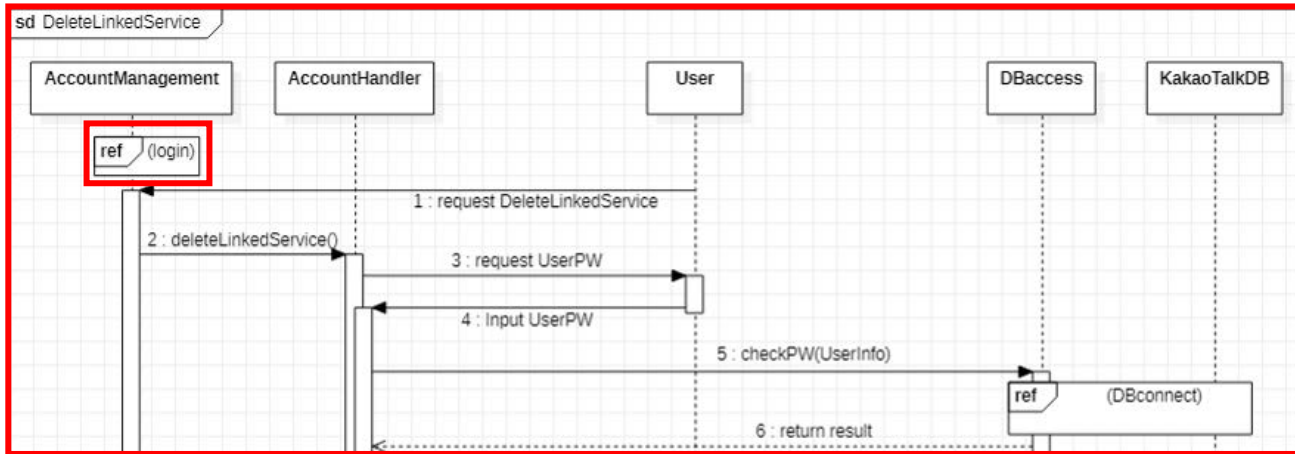
U3.로그인

U8. 연결 서비스
삭제



시퀀스 다이어그램 분석

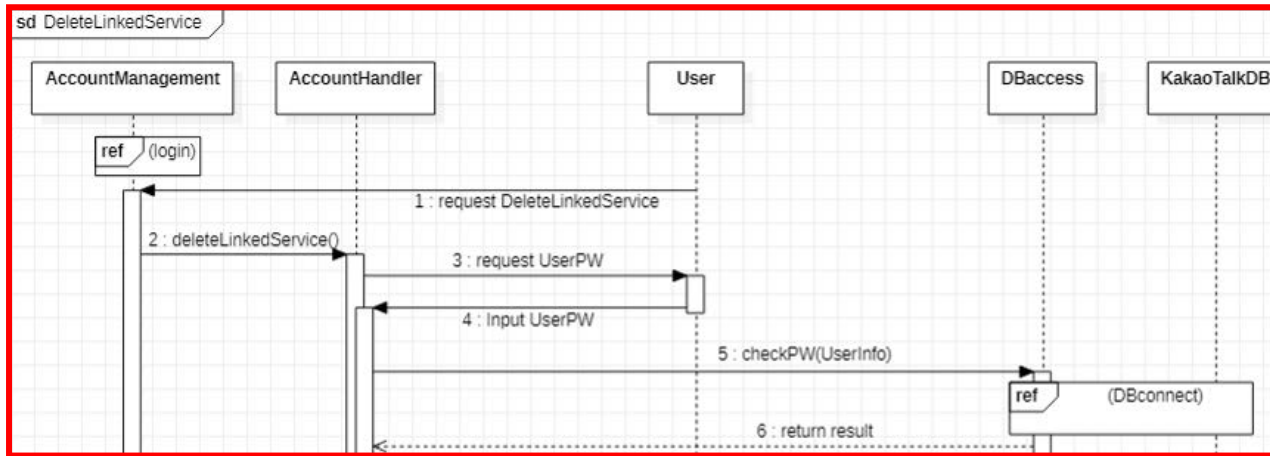
S8. 서비스 삭제 분석



- 1: UI적인 부분이므로 메소드 존재 x
- 2: UI에서 사용자가 클릭했다고 가정 login() 실행
3. ref로 Login 이라고 표기하였으나 시퀀스 다이어그램으로 무엇을 하는 것인지 알 수 없음.
 - 3-1. 유스케이스 다이어그램에는 명시가 되어있지만 AccountManagement에서 구현할 경우 AccountHandler에 전달해야하는 파라미터값이 클래스다이어그램과 맞지 않음.

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – AccountHandler.deleteLinkedService()



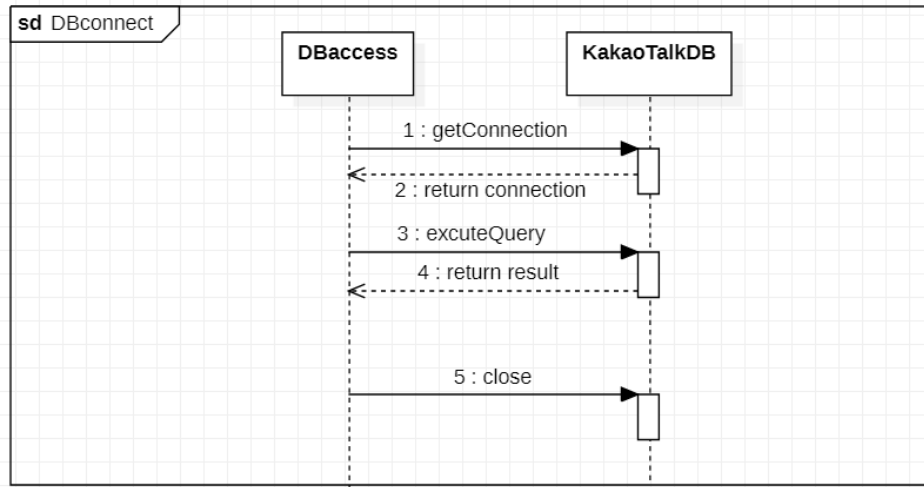
```
public boolean deleteLinkedService() {
    boolean login = handler.login(); //로그인 성공시 진행
    if (!login) { //실패시 중단
        return false;
    }
    //없는부분 끝
    Scanner sc = new Scanner(System.in); //비밀번호 요청
    userInfo.pw = sc.next();
    sc.close();
    boolean result = dbaccess.checkPW(userInfo); //비밀번호 대조
}
```

REF

2, 3, 4, 5

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – Dbaccess.checkPW()



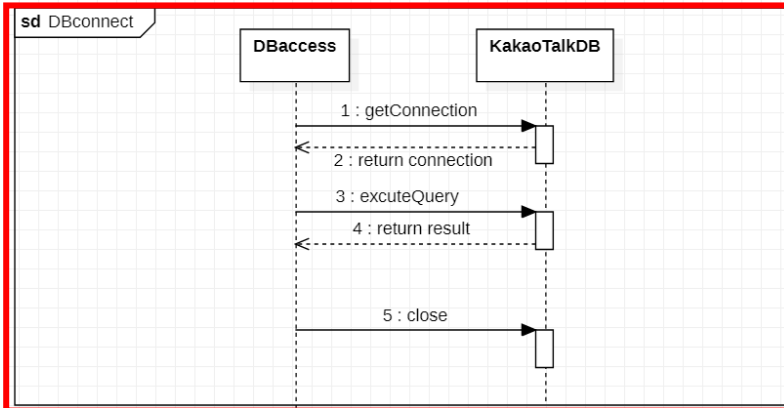
```
public boolean checkPW(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();

    if (result != null && userInfo.pw.equals(result.pw)) {
        return true;
    }
    else return false;
}
```

1,2,3,4,5

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – Dbaccess.checkPW()

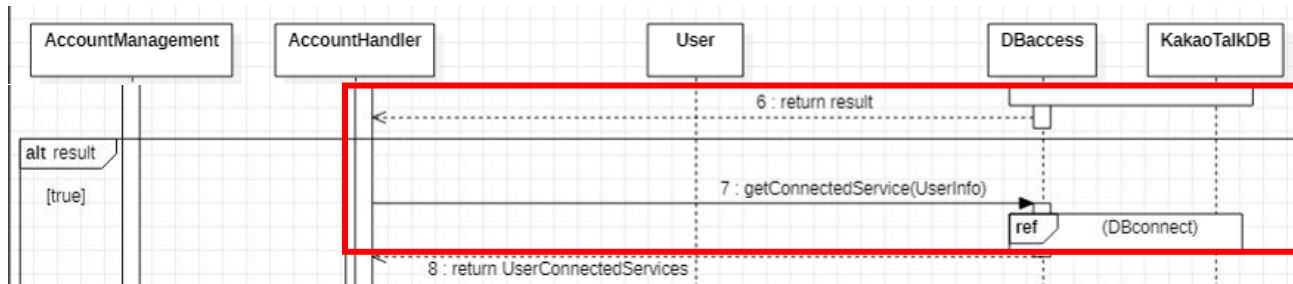


```
1 public class KakaoTalkDB extends DBaccess
2 {
3
4 public boolean getConnection()
5 {
6     boolean result = true;
7     System.out.println("connect complete from kakaotalk database");
8     return result;
9 }
10
11 public UserInfo excuteQuery(UserInfo UserInfo) //메소드,파라미터 명시 x 클래스 다이어그램에도 명시 x
12 {
13     UserInfo result = null; //db에서 가져왔다 가정
14     System.out.println("exeture query successs from kakaotalk db with using UserInfo");
15     return result;
16 }
17
18 public void close()
19 {
20     System.out.println("close success");
21 }
22 }
23
24 }
```

메소드 명시 x
임의로 메서드 구현

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – AccountHandler.deleteLinkedService()



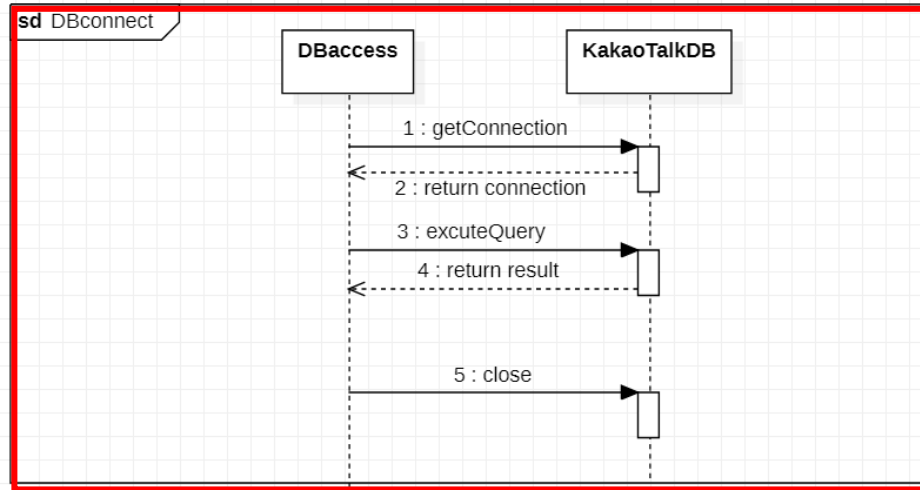
```
boolean result = dbaccess.checkPW(userInfo);    //비밀번호 대조

if (result){
    //서비스는 복수가 될 수 있기때문에 String 에서 List<String>으로 변경함
    List<String> service = dbaccess.getConnectedService(userInfo); //서비스 가져오기
    if (service != null){
        sc = new Scanner(System.in);           //삭제할 서비스 작성
        boolean delresult = dbaccess.deleteService(sc.next()); //서비스 삭제 요청
        sc.close();
    }else if (service == null){
        System.out.println("no Service");      //서비스 없을경우 서비스없음 표시
    }
}else if (!result){
    return false; //연결이 실패적이면 (비밀번호 대조 실패) false 리턴
}
return true;    //문제없다면 true 리턴
```

6, 7

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – Dbaccess.getConnectionService()

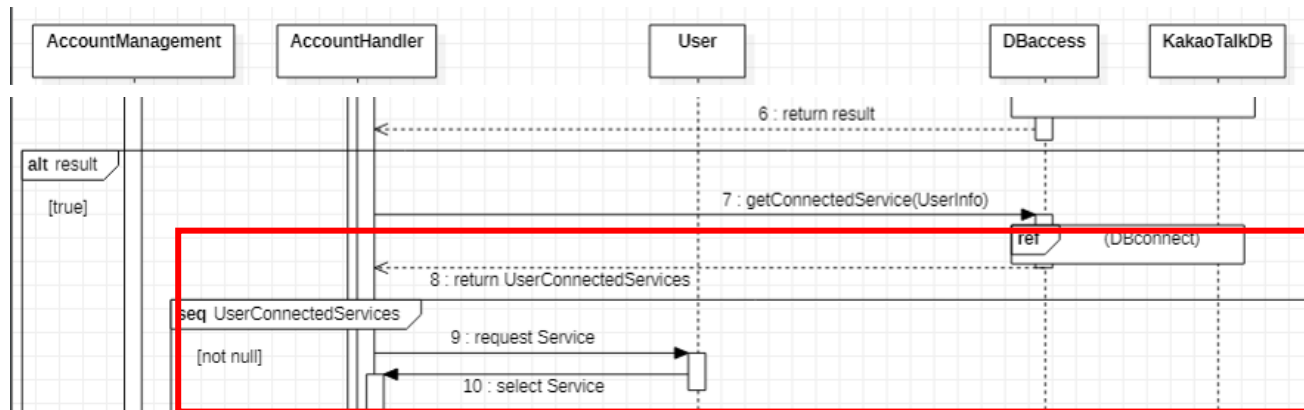


```
public List<String> getConnectionService(UserInfo userInfo) {  
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();  
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀  
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo); // DB에 사용자 정보를 입력한다.  
    kakaoTalkDB.close();  
  
    if (result == null || result.service == null){  
        return null;  
    }else{  
        return result.service;  
    }  
}
```

→ 1,2,3,4,5

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – AccountHandler.deleteLinkedService()



```
boolean result = dbaccess.checkPW(userInfo); //비밀번호 대조

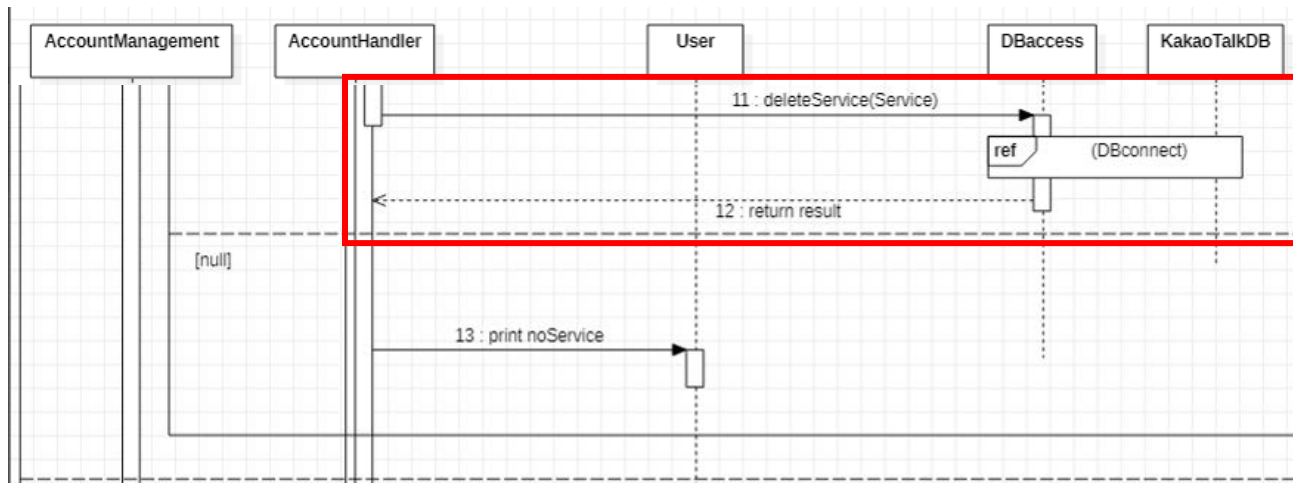
if (result){
    //서비스는 복수가 될 수 있기때문에 String 에서 List<String>으로 변경함
    String service = dbaccess.getConnectedService(userInfo); //서비스 가져오기
    if (service != null){
        sc = new Scanner(System.in); //삭제할 서비스 작성
        boolean delresult = dbaccess.deleteService(sc.next()); //서비스 삭제 요청
        sc.close();
    }else if (service == null){
        System.out.println("no Service"); //서비스 없을경우 서비스없음 표시
    }
}else if (!result){
    return false; //연결이 실패적이면 (비밀번호 대조 실패) false 리턴
}

return true; //문제없다면 true 리턴
```

8, 9, 10

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – AccountHandler.deleteLinkedService()

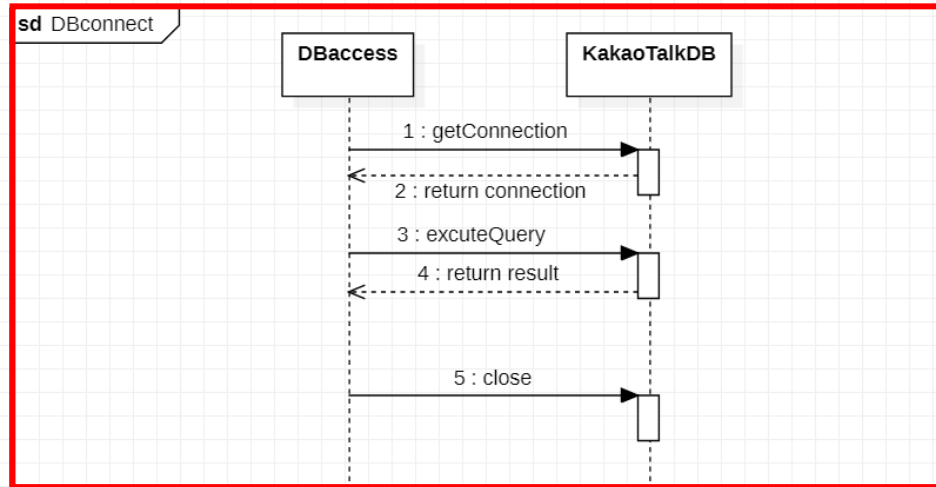


```
if (result){
    //서비스는 복수가 될 수 있기때문에 String 에서 List<String>으로 변경함
    String service = dbaccess.getConnectedService(userInfo);    //서비스 가져 오기
    if (service != null){
        sc = new Scanner(System.in);                //삭제할 서비스 작성
        boolean delresult = dbaccess.deleteService(sc.next()); //서비스 삭제 요청
        sc.close();
    }else if (service == null){
        System.out.println("no Service");           //서비스 없을경우 서비스없음 표시
    }
}else if (!result){
    return false; //연결이 실패적이면 (비밀번호 대조 실패) false 리턴
}
return true; //문제없다면 true 리턴
```

11

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – Dbaccess.deleteService()



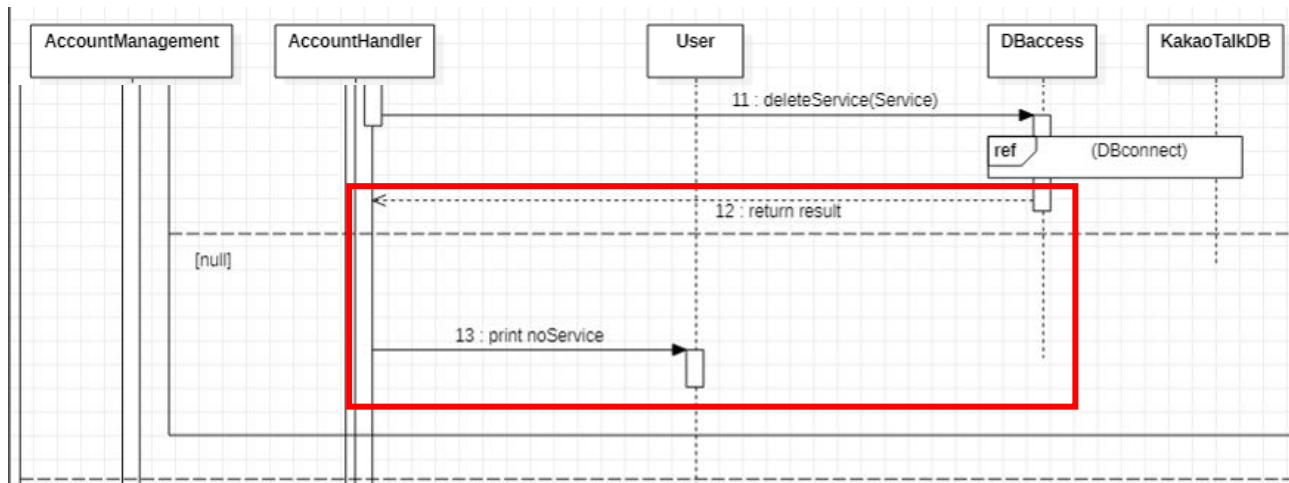
```
public boolean deleteService(String service) {
    delres = userInfo.service.remove(service); //로컬 DB에서 서비스 제거
    if (!delres){ //실패시 리턴
        return false;
    }
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo); // DB에 사용자 정보를 입력한다.
    kakaoTalkDB.close();

    if (result != null){
        return true;
    }else if (result == null){
        return false;
    }
}
```

→ 1,2,3,4,5

시퀀스 다이어그램 분석

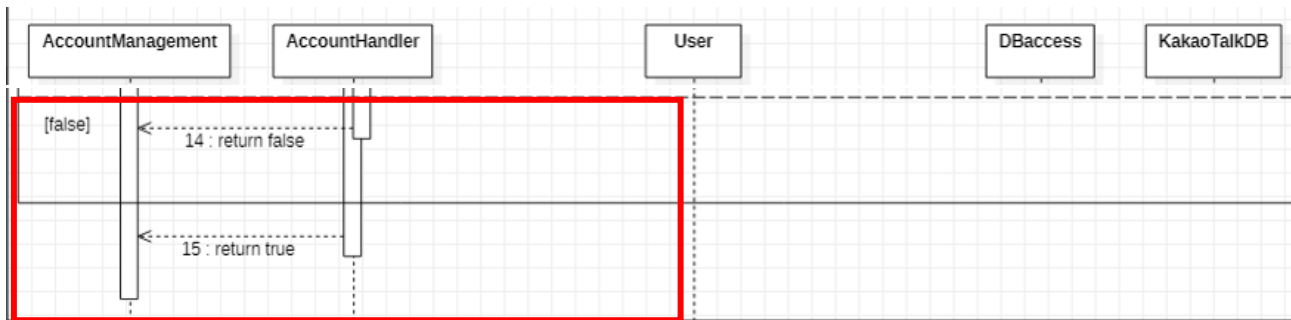
S8. 서비스 삭제 분석 – AccountHandler.deleteLinkedService()



```
if (result){
    //서비스는 복수가 될 수 있기때문에 String 에서 List<String>으로 변경함
    String service = dbaccess.getConnectedService(userInfo);    //서비스 가져 오기
    if (service != null){
        sc = new Scanner(System.in);                //삭제할 서비스 작성
        boolean delresult = dbaccess.deleteService(sc.next()); //서비스 삭제 요청 → 12
        sc.close();
    }else if (service == null){
        System.out.println("no Service");           //서비스 없을경우 서비스없음 표시 → 13
    }
}else if (!result){
    return false;    //연결이 실패적이면 (비밀번호 대조 실패) false 리턴
}
return true;    //문제없다면 true 리턴
```


시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – AccountHandler.deleteLinkedService()



```
boolean result = dbaccess.checkPW(userInfo);    //비밀번호 대조

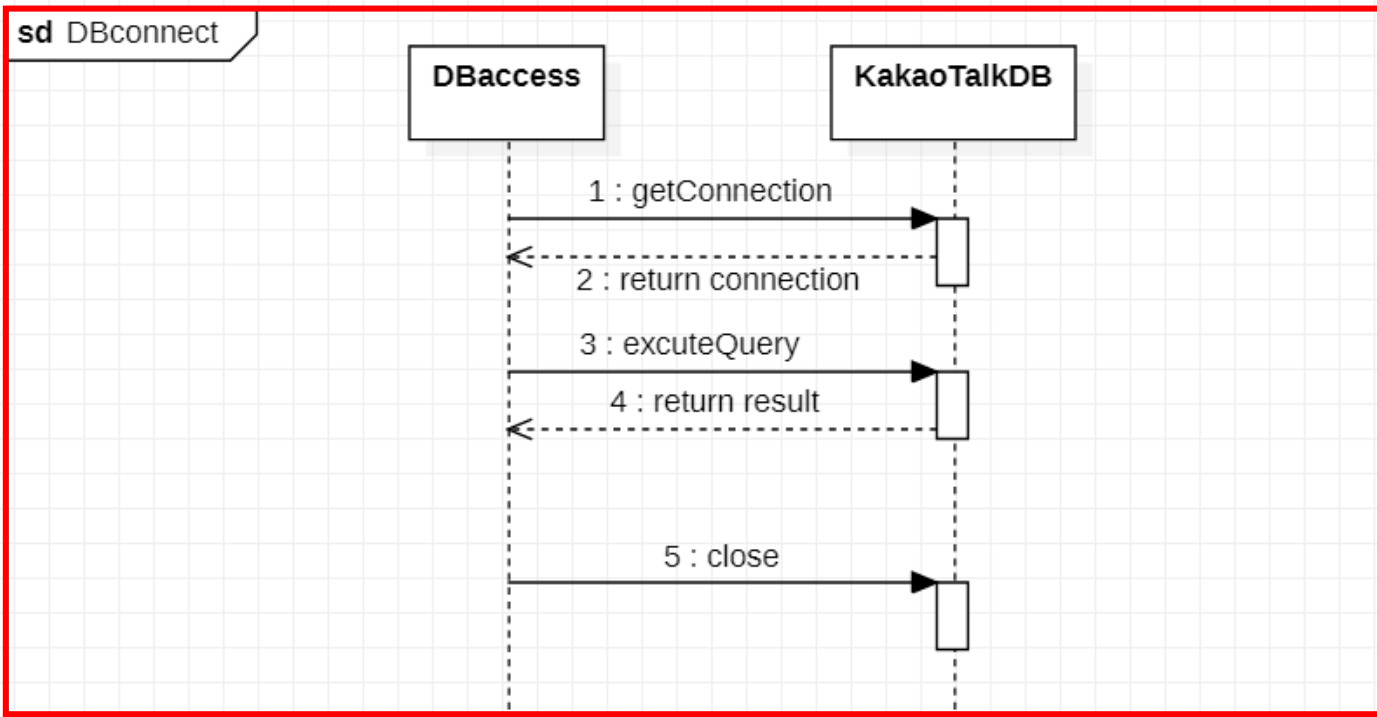
if (result){
    //서비스는 복수가 될 수 있기때문에 String 에서 List<String>으로 변경함
    String service = dbaccess.getConnectionService(userInfo);    //서비스 가져 오기
    if (service != null){
        sc = new Scanner(System.in);                //삭제할 서비스 작성
        boolean delresult = dbaccess.deleteService(sc.next()); //서비스 삭제 요청
        sc.close();
    }else if (service == null){
        System.out.println("no Service");            //서비스 없을경우 서비스없음 표시
    }
}else if (!result){
    return false; //연결이 실패적이면 (비밀번호 대조 실패) false 리턴
}
return true;    //문제없다면 true 리턴
```

14

15

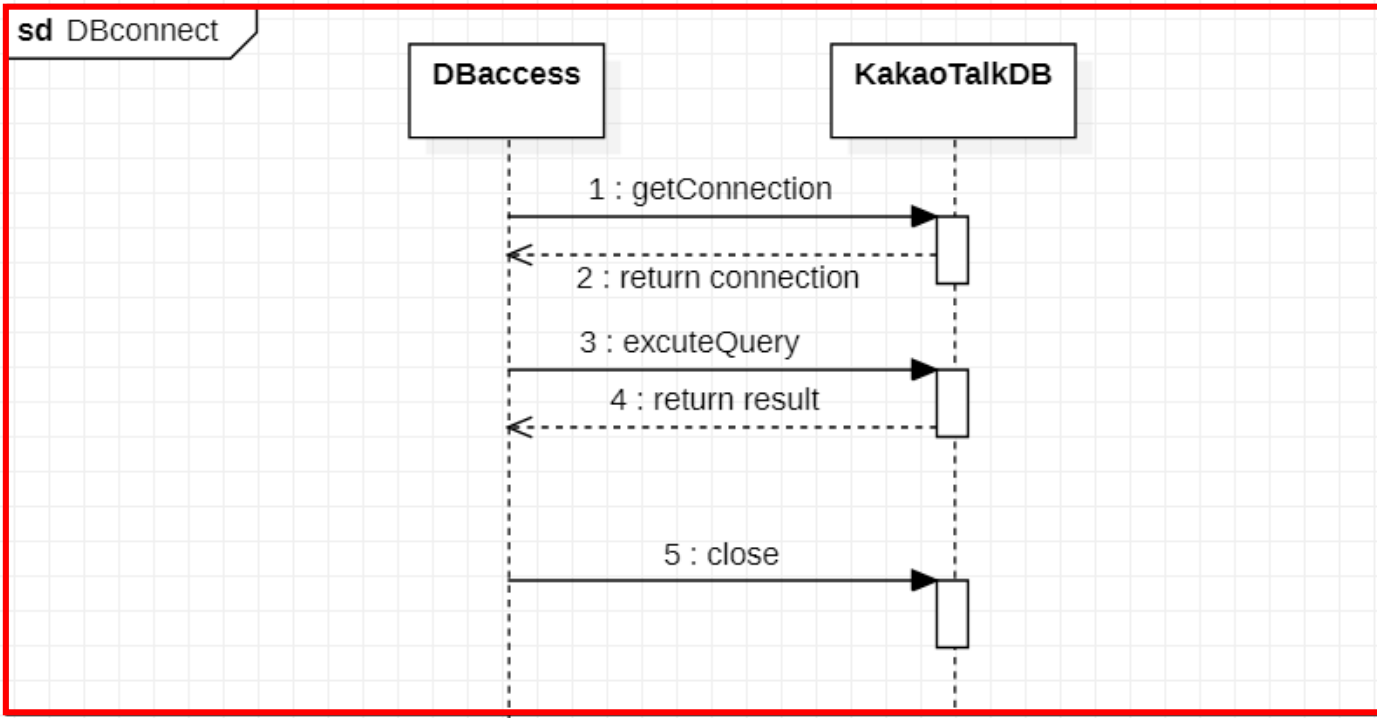
시퀀스 다이어그램 분석

S9. DBConnect



시퀀스 다이어그램 분석

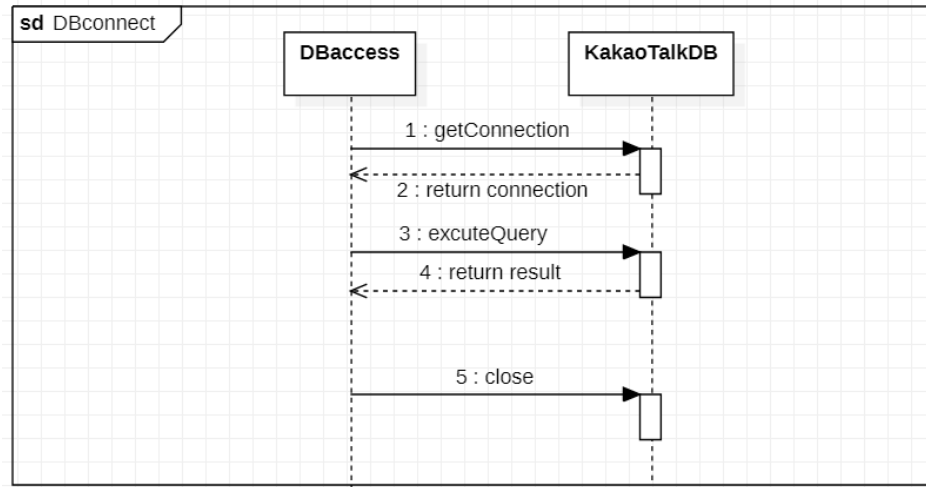
S9. DBConnect



1. 클래스 다이어그램과 일치하지 않음
2. 메소드가 명시되지 않음 (행위만 작성됨)

시퀀스 다이어그램 분석

S9. DBConnect – Dbaccess.checkPW()



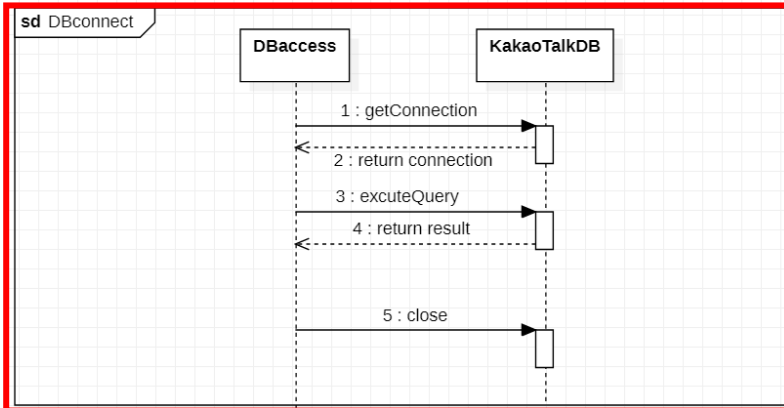
```
public boolean checkPW(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); // 변수 안씀
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();

    if (result != null && userInfo.pw.equals(result.pw)) {
        return true;
    }
    else return false;
}
```

1,2,3,4,5

시퀀스 다이어그램 분석

S8. 서비스 삭제 분석 – KakaoTalkDB

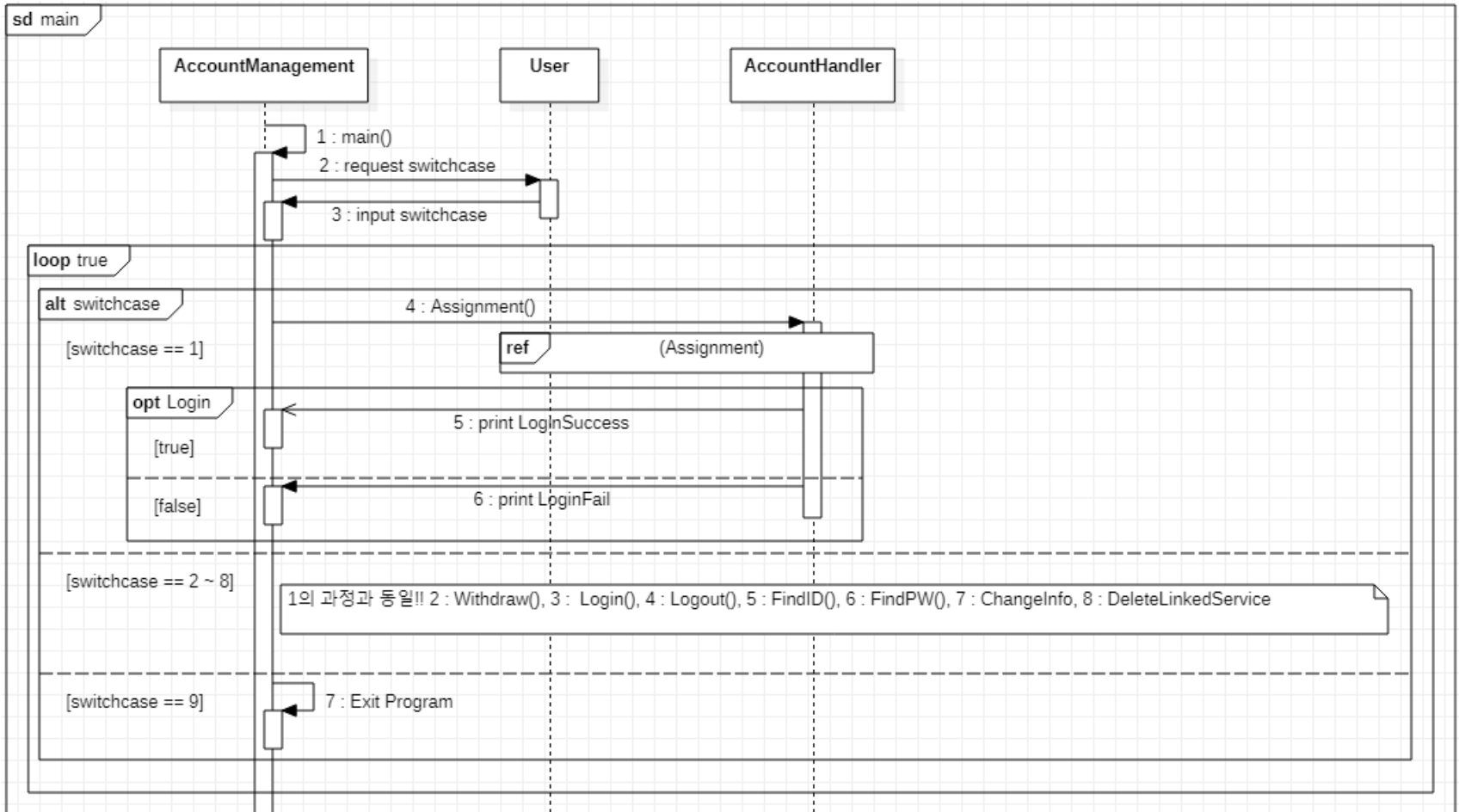


```
1 public class KakaoTalkDB extends DBaccess
2 {
3
4 public boolean getConnection()
5 {
6     boolean result = true;
7     System.out.println("connect complete from kakaotalk database");
8     return result;
9 }
10
11 public UserInfo excuteQuery(UserInfo UserInfo) //메소드,파라미터 명시 x 클래스 다이어그램에도 명시 x
12 {
13     UserInfo result = null; //db에서 가져왔다 가정
14     System.out.println("exeture query successs from kakaotalk db with using UserInfo");
15     return result;
16 }
17
18 public void close()
19 {
20     System.out.println("close success");
21 }
22 }
23
24 }
```

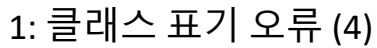
메소드 명시 x
임의로 메서드 구현

시퀀스 다이어그램 분석

S10. Main

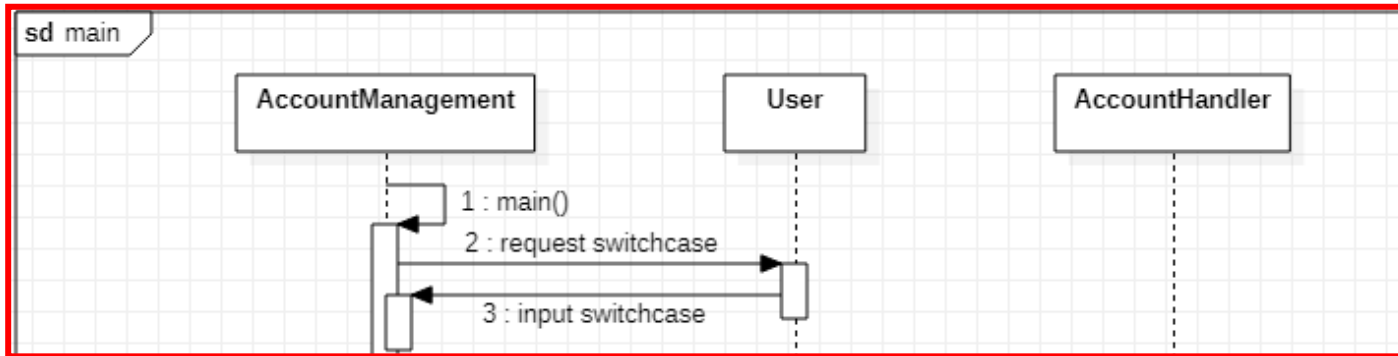


S10. Main 분석



시퀀스 다이어그램 분석

S10. Main 분석 – AccountHandler.deleteLinkedService()

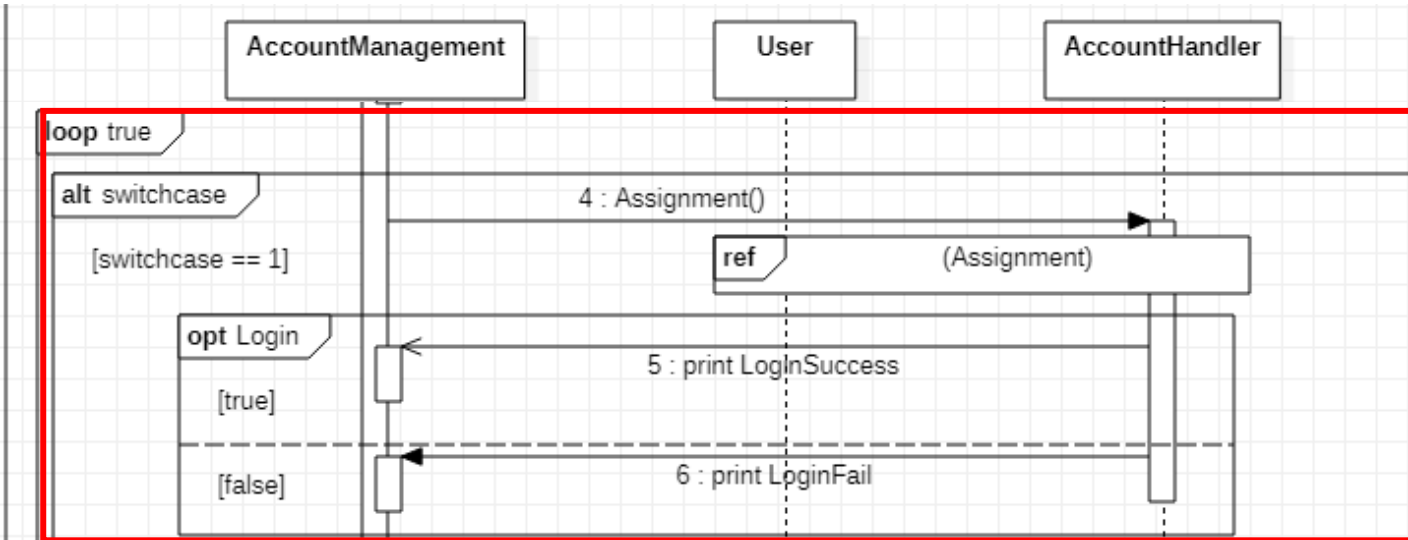


```
public static void main(String[] args) {
    AccountHandler handler = new AccountHandler();
    //(S2, S3)
    Scanner sc = new Scanner(System.in);
    int option = sc.nextInt();
    sc.close();
}
```

1, 2, 3

시퀀스 다이어그램 분석

S10. Main 분석 – AccountHandler.deleteLinkedService()

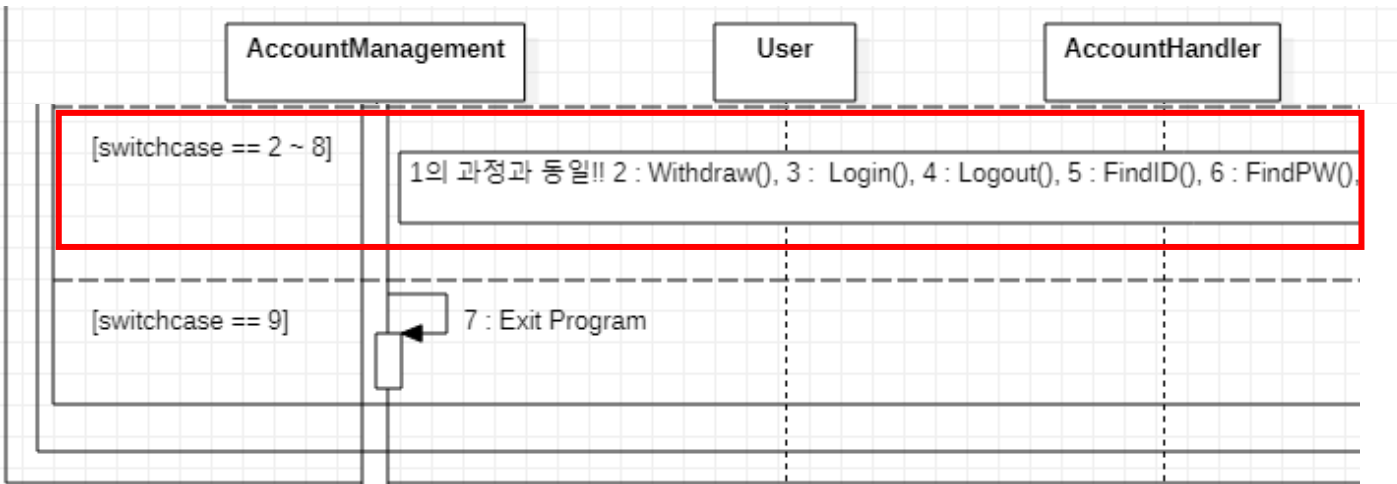


```
boolean result = false;
while (true){
    switch (option){
        //(S4)
        case 1 :
            result = handler.signUp();
            if (result){    //(S5)
                System.out.println("SignUp Success");
            }else{        //(S6)
                System.out.println("SignUp Fail");
            }
            break;
    }
}
```

4, 5, 6

시퀀스 다이어그램 분석

S10. Main 분석 – AccountHandler.deleteLinkedService()

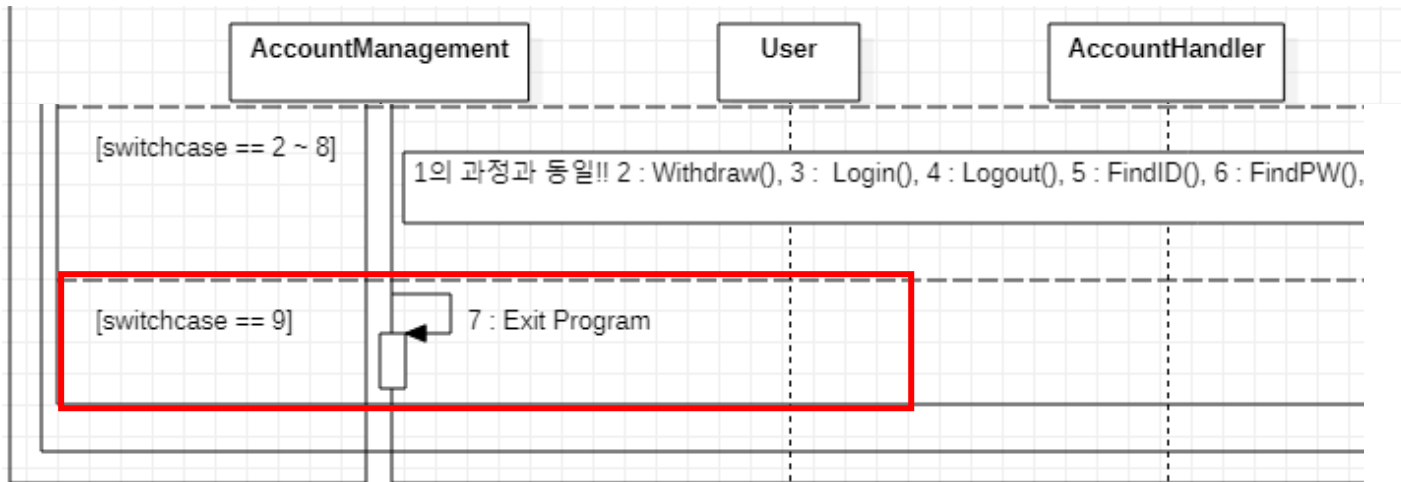


```
case 2:
    result = handler.withdraw();
    if (result){
        System.out.println("withDraw Success");
    }else{
        System.out.println("withDraw Fail");
    }
    break;
```

→ 노트 예시

시퀀스 다이어그램 분석

S10. Main 분석 – AccountHandler.deleteLinkedService()



```
case 8:
    result = handler.deleteLinkedService(); //서비스 삭제 요청
    if (result){ //성공시 삭제되었다고 알림, 실패시 실패라고 알림
        System.out.println("deleteLinkedService Success");
    }else{
        System.out.println("deleteLinkedService Fail");
    }
    break;
}

// (S7)
if (option == 9){
    break;
}
```

7

구현 어려웠던 점

1. 메소드 명명 추상적

2. 구현의 의미가 없는 메소드

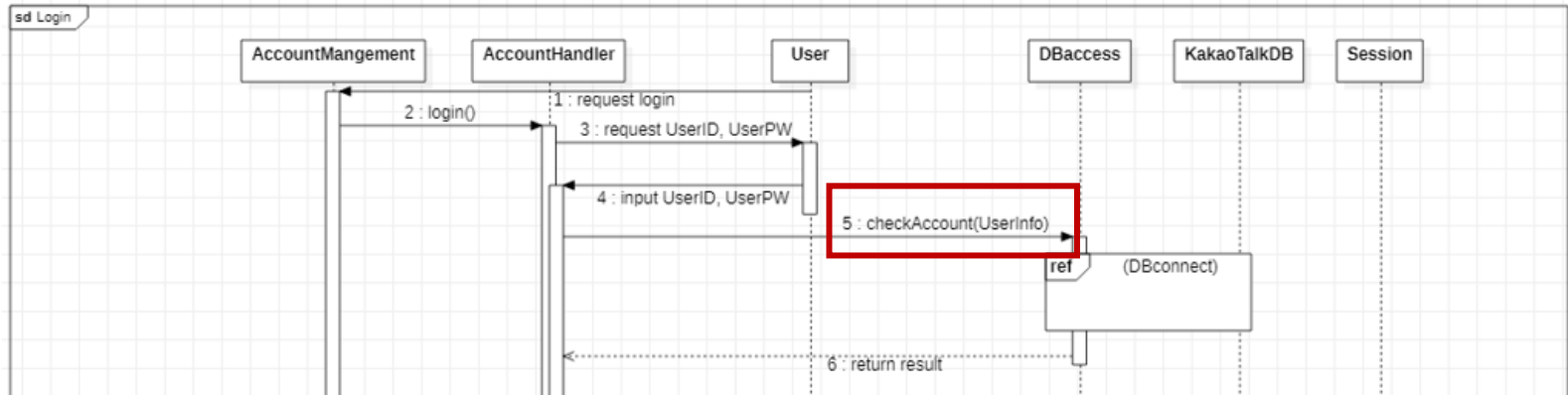
3. 문서상 누락

4. 재활용하지 않는 메소드

5. 유스케이스, 시퀀스 다이어그램
1:1 대응 X

구현 어려웠던 점

메소드 명명 추상적

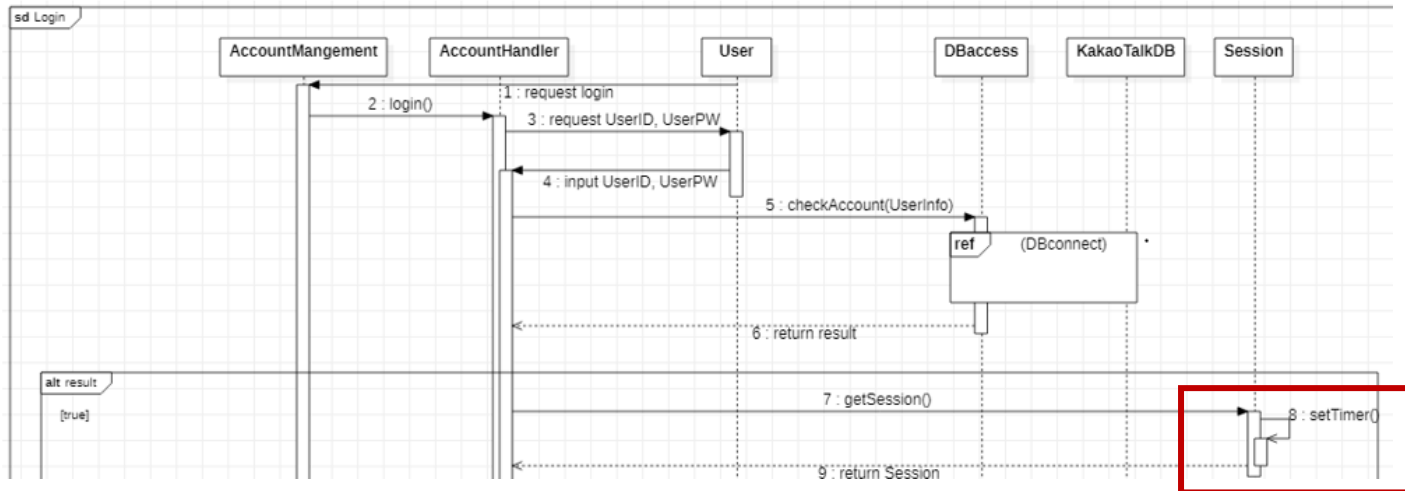


5: 클래스 다이어그램에는 아이디와 패스워드로 계정을 체크한다 명시
유스케이스, 시퀀스에서는 알기 힘들

checkAccount메소드 안에 checkID(),checkPW()의 메소드를 더 만들어
동작자체를 확실히 명시하면 좋을 것 같음

구현 어려웠던 점

구현의 의미가 없는 메소드



요구되는 기능 중 자동 로그인 해제 등의 기능이 있는데
이를 위한 setTimer 메소드는 명시가 되어있으나
Timer를 체크하거나 조건에 따라
자동 로그인 해제를 한다던가 하는 메소드가 없음
메소드 누락일 가능성도 존재하나
결론적으로 setTimer은 타이머를 set 하긴 하지만
set한 타이머를 사용하지 않아 어떻게 set할지 알기 어려움

구현 어려웠던 점

문서상 누락

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">1. 사용자는 정보 수정을 요청한다.2. 사용자는 사용자 아이디에 해당하는 비밀번호를 입력한다.3. 카카오톡 DB 시스템은 사용자가 입력한 비밀번호가 유효한지 확인한다.4. 사용자가 입력한 비밀번호가 유효하면 사용자는 수정할 정보를 선택한다5. 사용자는 선택한 정보에 대한 수정할 내용을 입력한다.6. 카카오톡 DB 시스템은 고객의 정보를 수정한다.		
선택 흐름	✓ 카카오톡 DB 시스템에서 사용자가 입력한 비밀번호가 유효하지 않다면 종료한다.		

alt Case	
[name]	
[gender]	
[Email]	
[PhoneNumber]	

유스케이스의 표현이 모호함 (시퀀스 다이어그램)

구현 어려웠던 점

문서상 누락

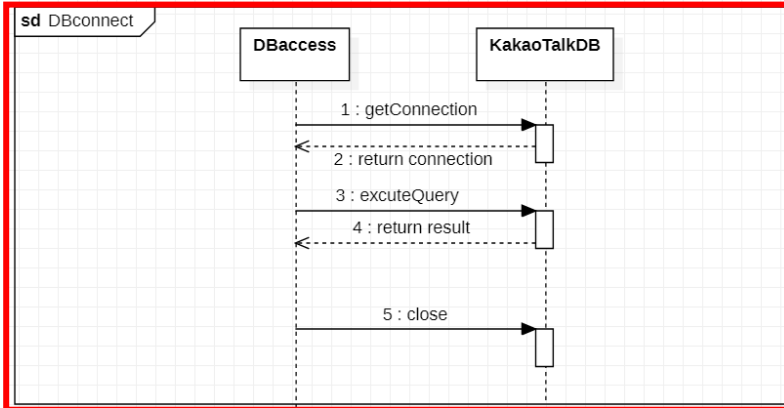
유스케이스 명세서

유스케이스 명	U9. 자동로그인	유스케이스 개요	사용자가 로그인에 성공하면 이후 자동으로 로그인이 되는 <u>유스케이스</u> 이다.
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.		

유스케이스에 있는 '자동로그인'의 내용이 이후 클래스 다이어그램, 시퀀스 다이어그램에 나오지 않음.

구현 어려웠던 점

재활용 하지 않는 메소드



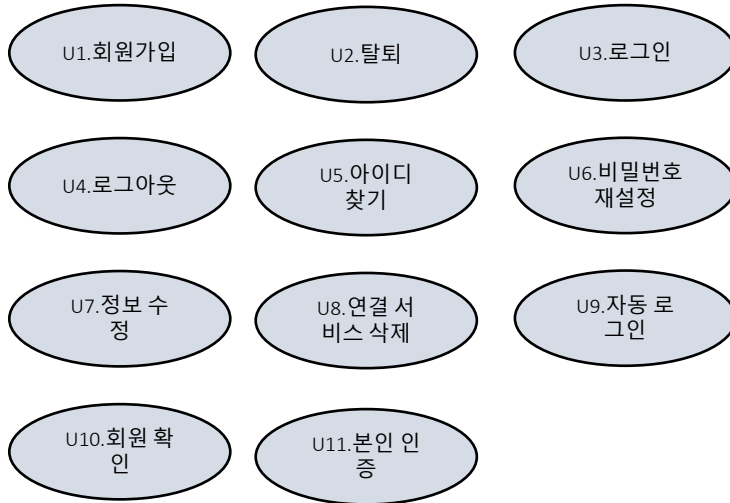
```
public boolean checkAccount(UserInfo userInfo){
    KakaoTalkDB kakaoTalkDB = new KakaoTalkDB();
    boolean connection = kakaoTalkDB.getConnection(); //connection 변수 사용 x
    UserInfo result = kakaoTalkDB.excuteQuery(userInfo);
    kakaoTalkDB.close();
    //예외처리 + id, pw 동일한지 체크
}
```

Dbconnect에 해당되는 코드

Dbconnect reference는 위 메소드 외 많은 곳에서 사용하지만
메소드로 감싸지 않아 구현상 메소드가 재활용되지 않음
메소드로 감쌌다면 구현시에 Dbconnect()만 사용하면 되지만
그렇지 않아 getSession부터 시작해서 중복된 메소드를 계속 호출함

구현 어려웠던 점

유스케이스, 시퀀스 다이어그램 1:1 대응 X



실제 시퀀스 개수는 10개 즉 완벽하게 일치되지 않고
유스케이스 = 시퀀스가 아니어서
중복되지만 재활용을 하지 않는 부분이 많음