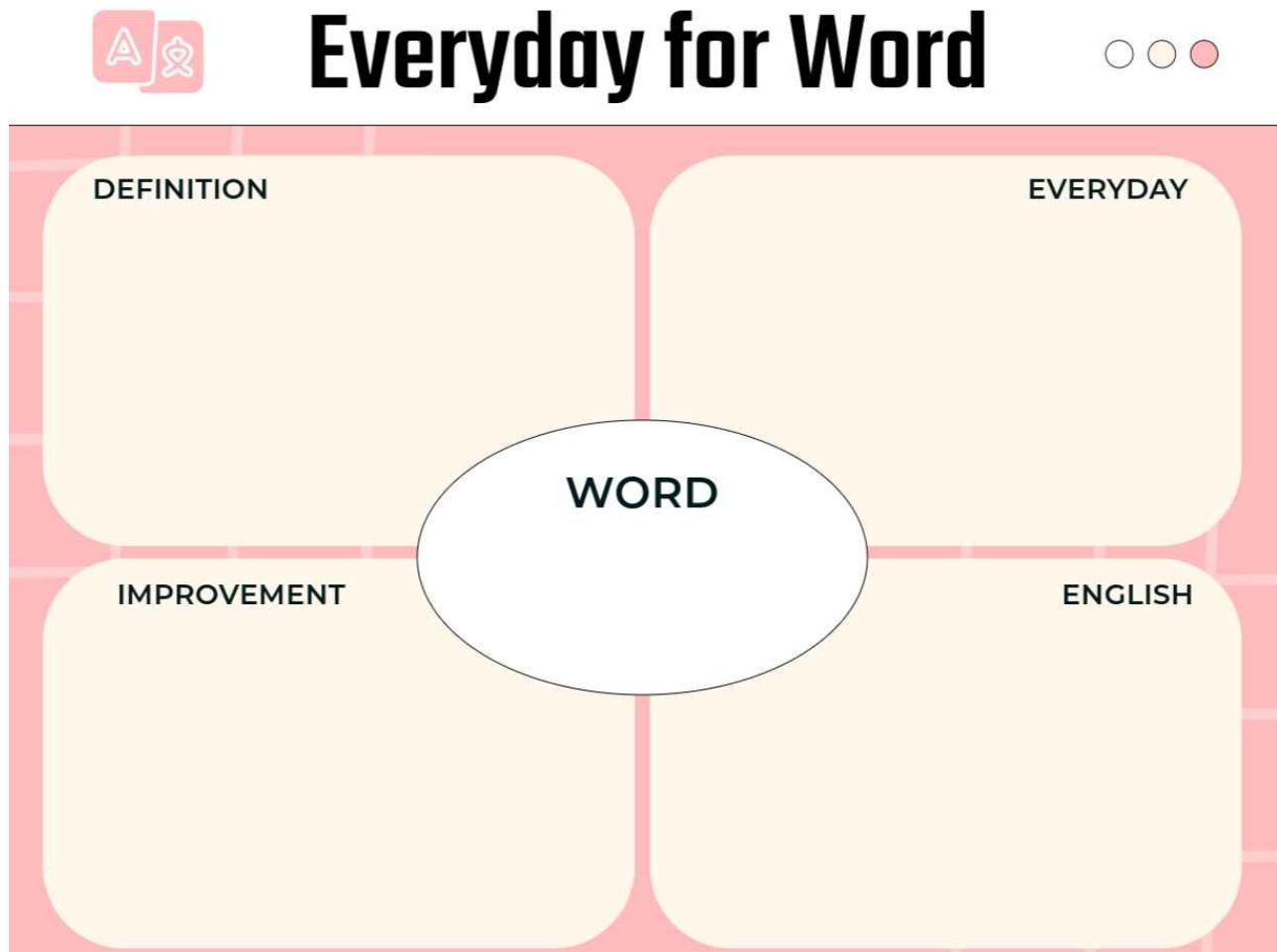


3. Design



(22212040, 김가현, hyeonnni@naver.com)

[Revision history]

Revision date	Version #	Description	Author
2024/05/07	1.1	로고 변경, 로그아웃 기능 삭제	김가현
2024/05/21	1.2	앱 개발에서 웹 개발로 변경	김가현
2024/06/04	1.3	분석 단계에서 DailyWordTest 수정	김가현

= Contents =

1. Introduction	
2. Class diagram	
3. Sequence diagram	
4. State machine diagram	
5. Implementation requirements	
6. Glossary	
7. References	

1. Introduction

- Summarize the contents of this document.
- Describe the important points of your design.
- 12pt, 160%.

1) Summary

취업이나 다양한 분야에서 도전을 하고자 할 때, 토익 점수는 종종 필수적인 요구 사항으로 제시됩니다. 토익 점수는 국제적인 비즈니스 환경에서의 소통 능력을 평가하는 중요한 지표로 인정받고 있습니다. 이에 따라 많은 사람들이 토익 시험을 준비하고 이를 통해 자신의 영어 능력을 증명하고자 합니다.

이러한 요구를 충족하기 위해 "Everyday for Word"라는 시스템을 개발하게 되었습니다.

2) Features of "Everyday for Word"

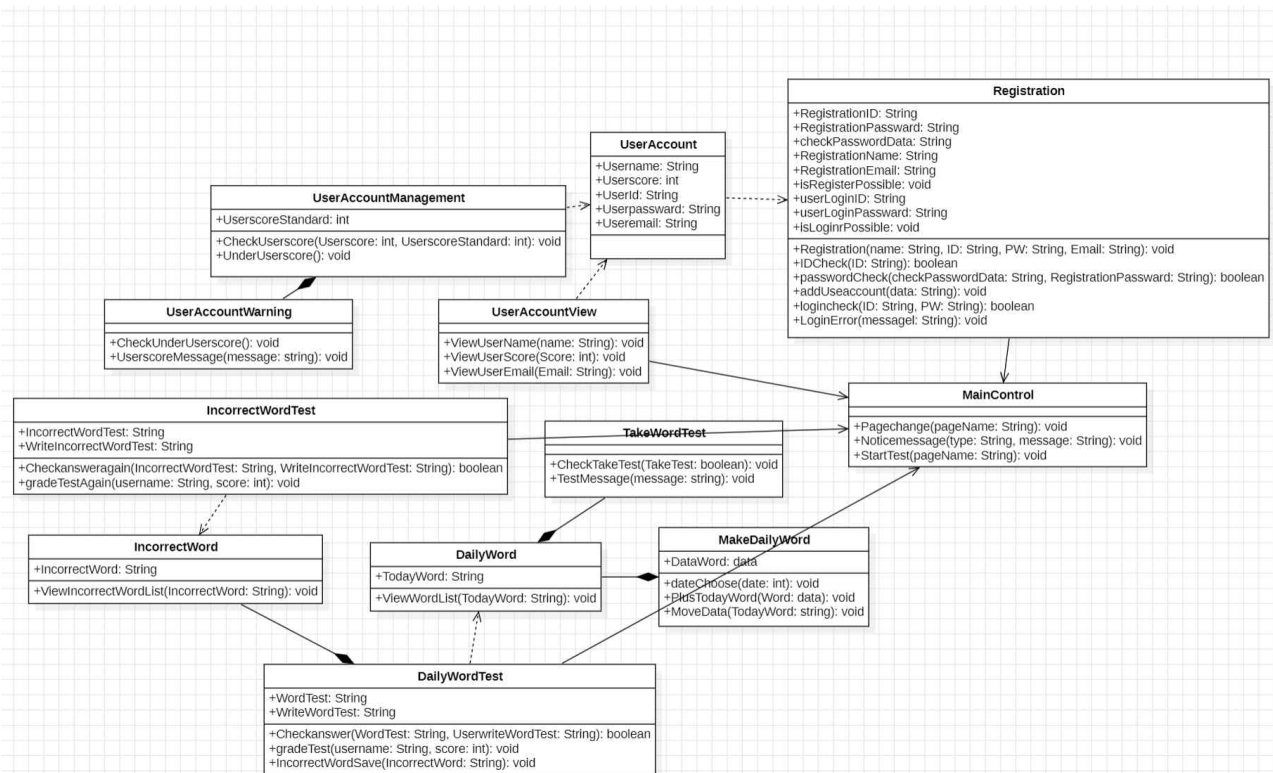
"Everyday for Word"는 매일 영어 단어를 외우는 데에 중점을 두며, 이를 위해 출석점수 관리 시스템을 도입하여 사용자들이 매일 정해진 양의 단어를 응시하도록 유도합니다. 또한, "Everyday for Word"는 단어 시험 기능을 통해 사용자들이 올바르게 단어를 외웠는지를 확인할 수 있습니다. 이를 통해 사용자들은 자신의 학습 진행 상황을 파악하고, 부족한 부분을 보완할 수 있습니다. 더불어, 복습 단어 확인 및 복습 단어 테스트 기능을 제공하여 사용자들이 학습한 단어를 확실히 기억하고 있는지를 확인할 수 있습니다..

"Everyday for Word"의 이러한 특징들은 사용자들의 꾸준한 학습 습관을 형성하고, 영어 어휘력을 지속적으로 향상시키는 데에 도움이 됩니다. 효과적으로 영어 단어를 습득하며 복습 하기에, 영어 능력을 더욱 키울 수 있습니다.

Design 단계에선 class diagram, sequence diagram, state machine diagram 을 그리고 각 diagram에 대한 설명 하여 Everyday for Word가 디자인 되는 지를 상세 하게 잘 이해할 수 있도록 한다.

2. Class diagram

- Draw a class diagram.
- Describe each class in detail (attributes, methods, others) (table type).
- 12pt, 160%.



1) MainControl

Attributes
Methods
+Pagechange (pageName: String) : void : 화면 전환 (회원가입에서 로그인으로 화면 전환 등등)
+Noticemessage (type: String, message: String) : void : type에 맞는 알림창을 띄워주고 message로 알림 메시지 지정
+StartTest(pageName: String) : void : 현재 페이지를 단어 시험을 응시하는 페이지로 이동

2) UserAccount

Attributes
+ Username : String : 사용자 이름
+ Userscore : int : 사용자 출석점수

+ UserId : String : 사용자 ID
+ Userpassword : String : 사용자 비밀번호
+ Useremail : String : 사용자 이메일
Methods

3) Registration

Attributes
+RegistrationID : String : 사용자가 회원가입시 입력한 아이디
+RegistrationPassword : String : 사용자가 회원가입시 입력한 비밀번호
+checkPasswordData: String : 사용자가 회원가입시 입력한 비밀번호 확인
+RegistrationName : String : 사용자가 회원가입시 입력한 이름
+RegistrationEmail : String : 사용자가 회원가입시 입력한 이메일
+isRegisterPossible: void : 회원가입 성공여부 결과를 가지는 값
+userLoginID : String : 사용자가 로그인 시 입력한 아이디
+userLoginPassword : String : 사용자가 로그인 시 입력한 비밀번호
+isLoginrPossible: void : 로그인 성공여부 결과를 가지는 값
Methods
+Registration(name:String, ID:String, PW:String, Email:String) : void :회원 가입 생성자
+IDCheck(ID:String) :boolean : 회원등록 화면에서 입력한 아이디(학번)가 이미 등록된 것은 아닌지 확인.
+passwordCheck(checkPasswordData : String, RegistrationPassword : String) : boolean :회원등록 창에서 비밀번호, 비밀번호 확인란에 입력한 비밀번호가 같은지 확인하고 틀리다면 메시지를 띄워줌
+addUseaccount (data : String):void : user account 에 회원정보 추가
+logincheck(ID:String, PW:String): boolean : ID와 비밀번호가 저장 되어있는지 확인
+LoginError(messagel:String) :void : 사용자가 로그인 할 때 아이디와 비번을 잘못입력한 우 메시지를 띄어준다.

4) UserAccountView

Attributes
Methods
+ ViewUserName (name : String) : void : 사용자의 계정정보 중 이름 확인 + ViewUserScore (Score : int) : void : 사용자의 계정정보 중 출석점수 확인 + ViewUserEmail (Email:String) : void : 사용자의 계정정보 중 이메일 확인

5) DailyWord

Attributes
+ TodayWord : String[] : 사용자가 외워야 할 단어 리스트
Methods
+ ViewWordList (TodayWord : String) : void : 사용자가 외워야 할 단어 리스트를 화면에 나타냄

6) DailyWordTest

Attributes
+ WordTest : String[] : 사용자가 단어들을시험칠 때 화면에 나타나는 시험 내용 저장 + WriteWordTest: String[] : 사용자가 테스트 응시할 때 입력한 내용 저장
Methods
+ Checkanswer (WordTest : String[], UserwriteWordTest: String[]) : boolean : 사용자의 단어 테스트 채점 +gradeTest(username: String, score : int) : void : 사용자의 이름과 점수가 나타남 + IncorrectWordSave (IncorrectWord : String[]) : void : 오답인 단어들을 data에 저장해둠

7) IncorrectWord

Attributes
+ IncorrectWord : String : 오답인 단어 리스트
Methods
+ ViewIncorrectWordList (IncorrectWord : String) : void : 사용자가 맞추지 못한 단어 리스트를 화면에 나타냄

8) IncorrectWordTest

Attributes	
+ IncorrectWordTest : String[]	: 사용자가 틀린 단어들을 다시 시험칠 때 화면에 나타나는 시험 내용 저장
+ WriteIncorrectWordTest: String[]	: 사용자가 틀린 단어들로만 다시 테스트 응시할 때 입력한 내용 저장
Methods	
+ Checkansweragain (IncorrectWordTest : String[], WriteIncorrectWordTest: String[])	: boolean : 사용자의 오답 단어 테스트 채점
+ gradeTestAgain (username: String, score : int)	: void : 사용자의 이름과 점수가 나타남

9) UserAccountManagement

Attributes	
+ UserscoreStandard : int	: 사용자의 출석 점수 기준
Methods	
+ CheckUserscore (Userscore : int , UserscoreStandard : int)	: void : UserscoreStandard에 따른 기준으로 Userscore (사용자 출석점수)를 확인하여 기준 맞는지 아닌지 확인
+ UnderUserscore : void	: 사용자의 출석점수가 기준 맞으면 그 내용을 저장한다

10) MakeDailyWord

Attributes	
+ DataWord : data	: 사용자가 저장해둔 날짜 별 단어 리스트 데이터이고 이 단어들이 TodayWord에 각각 저장된다
Methods	
+ dateChoose (date : int)	: void : 사용자가 몇 번째 단어를 입력해야 하는지 선택할 수 있음
+ PlusTodayWord (Word : data)	: void : 사용자가 각각의 날짜에 입력한 단어 리스트를 데이터에 저장한다
+ MoveData (TodayWord : string[])	: void : 사용자가 오늘 외워야 하는 단어 (TodayWord)로 DataWord에서 각각의 날짜에 맞게 단어 리스트를 이동시켜줌

11) TakeWordTest

Attributes	
Methods	

<p>+ CheckTakeTest (TakeTest : boolean) : void : 사용자가 단어테스트 데이터 저장 여부 확인으로 응시 여부 판단</p> <p>+ TestMessage (message : string) : void : 사용자에게 단어 테스트 응시하라는 message를 전송한다</p>

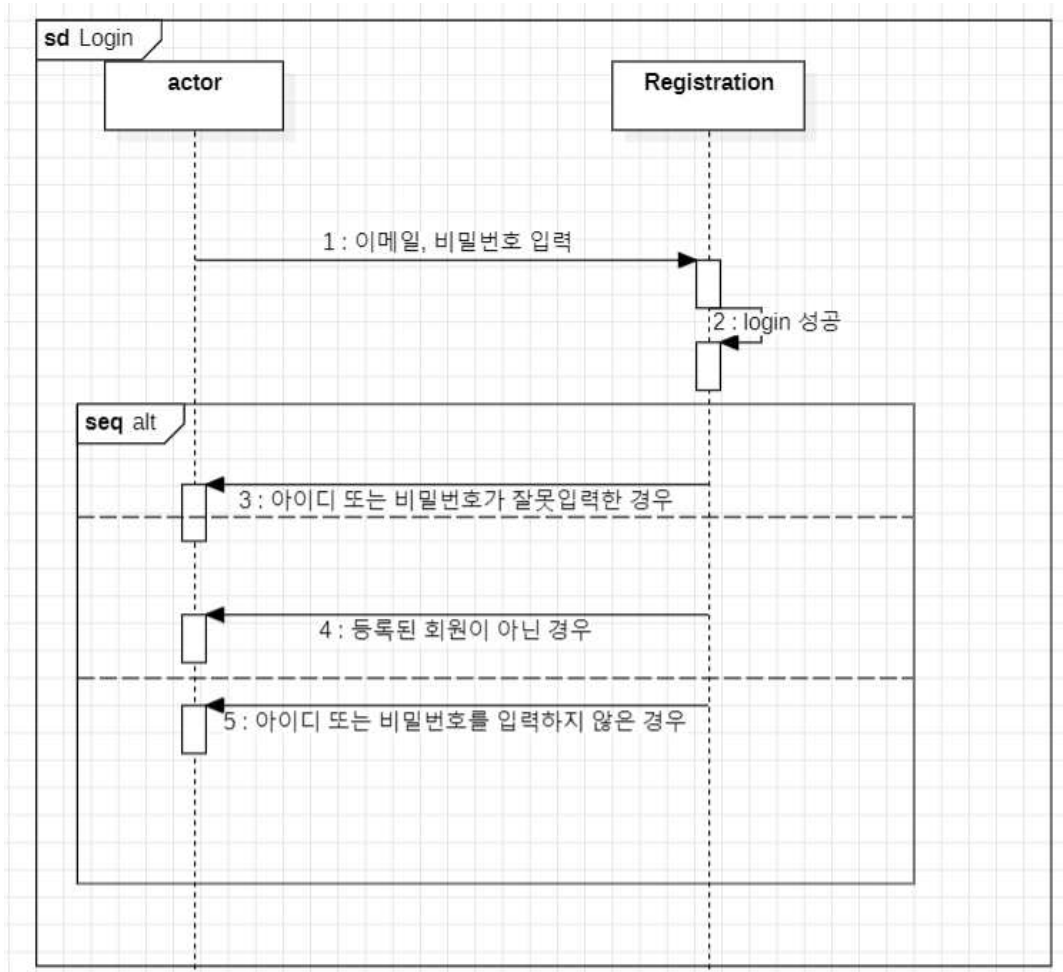
12) UserAccountWarning

Attributes
Methods
+ CheckUnderUserscore : void : 사용자의 출석 점수가 기준 밑이라는 내용이 저장되어있는지 여부를 판단하여 메시지 전송 여부를 선택한다
+ UserscoreMessage (message : string) : void : 사용자에게 출석 점수 경고 message를 전송한다

3. Sequence diagram

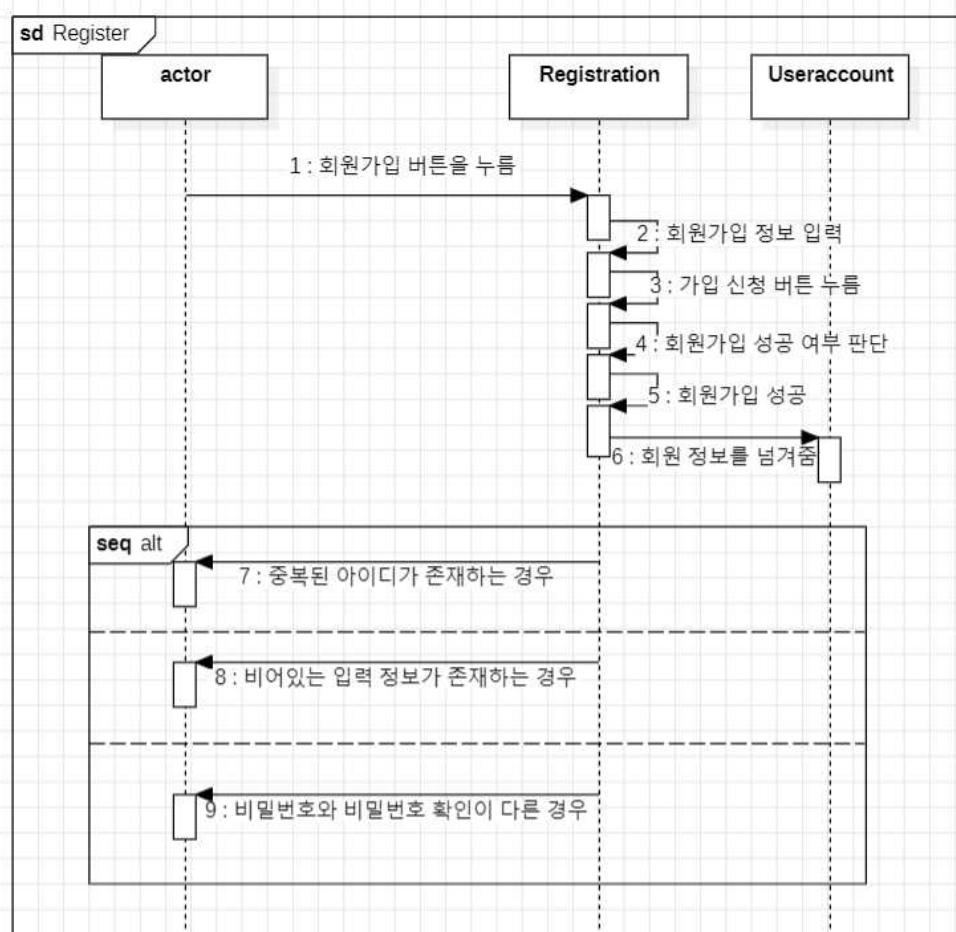
- Draw sequence diagrams for the whole functions of your system (this is related to the use cases you made in the Conceptualization phase).
- Explain each sequence diagram.
- 12pt, 160%.

1) Login



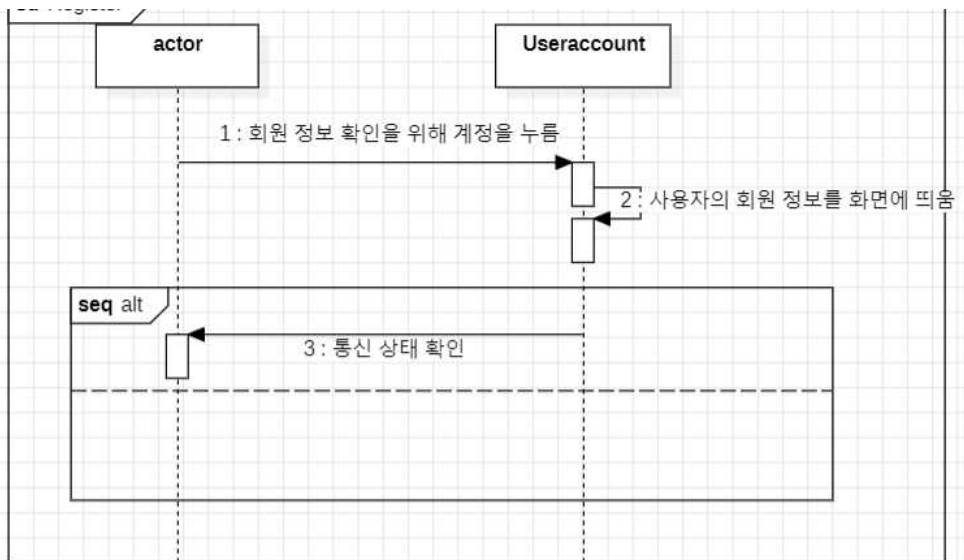
Login Use case 에서의 Sequence Diagram이다. User이 login을 시도하게 되면 Registration 클래스에서 회원이 가입 되어있는지 본 후 아이디와 비밀번호를 확인한 후 로그인에 성공하게 된다. 아이디 또는 비밀번호가 잘못 입력한 경우, 등록된 회원이 아닌 경우, 아이디 또는 비밀번호를 입력하지 않은 경우 login을 하지 못한다.

2) Register



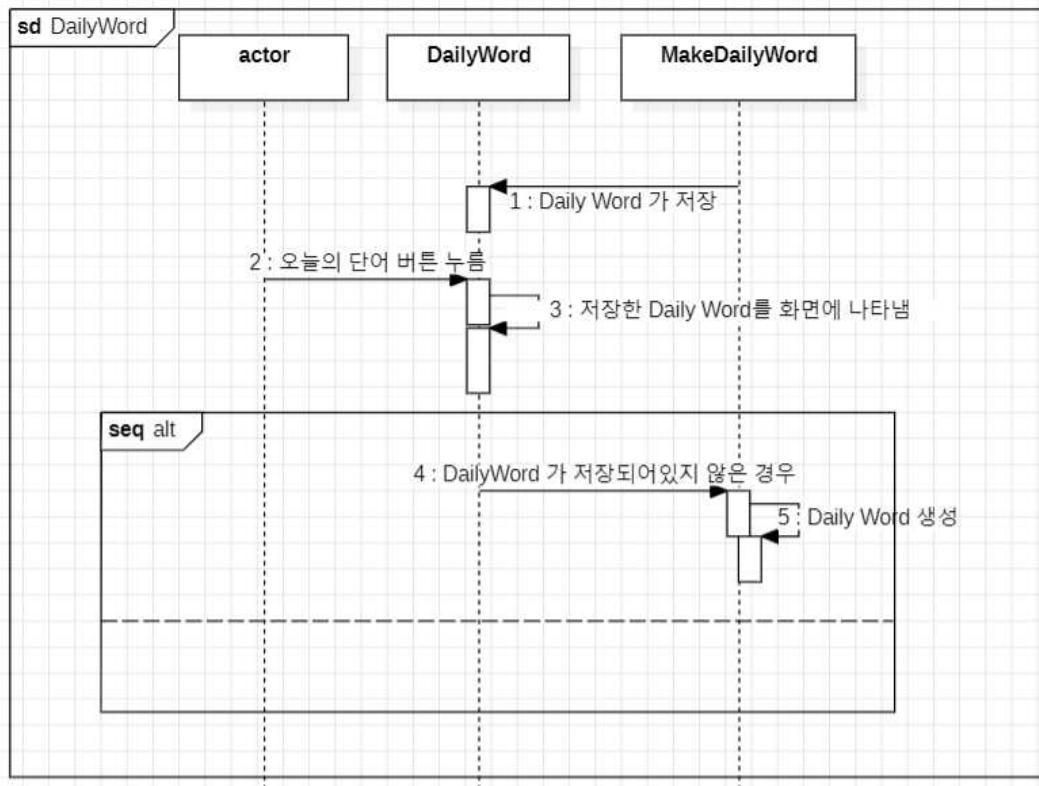
Register case 에서의 Sequence Diagram이다. User이 회원가입 버튼을 눌러 회원가입을 시도하게 되면 Registration 클래스에서 회원가입에 필요한 정보를 입력하는 화면을 띄운다. 그리고 User이 정보를 입력하고 가입 신청 버튼을 누르게 되면 Registration에서 회원가입 성공 여부를 판단한다. 이때 중복된 아이디가 존재하거나 입력정보중 비어있는 곳이 있는 경우, 비밀번호와 비밀번호 확인이 다른 경우 회원가입에 실패하게 된다. 회원가입 성공 여부를 판단해 성공하게 되면 회원이 입력한 회원 정보를 User account에 전달한다.

3) View profile



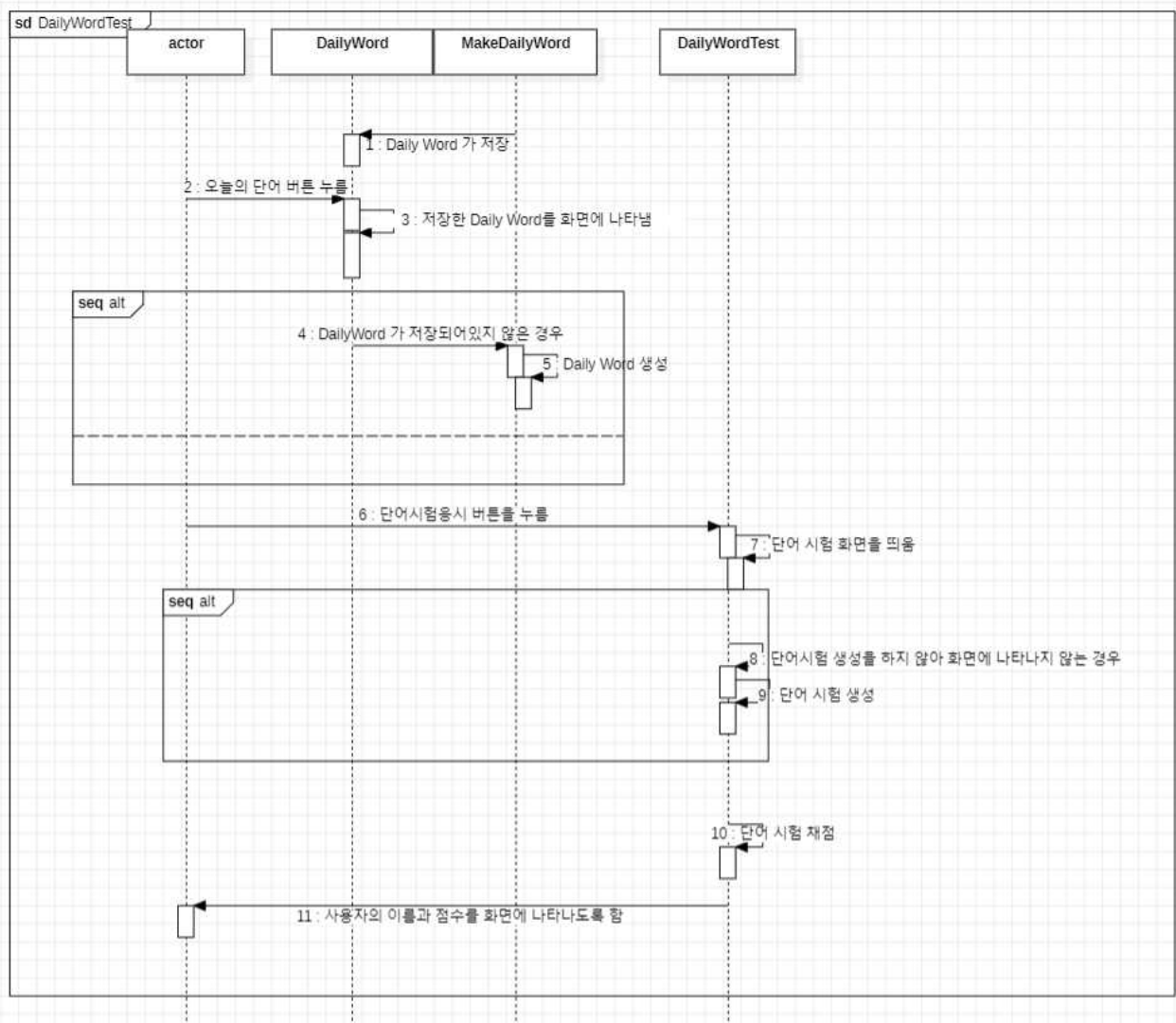
View profile Use case 에서의 Sequence Diagram이다. User이 회원 정보를 확인하기 위해 메인화면에서 자신의 계정을 누르게 되면 Useraccount 클래스는 사용자의 회원 정보를 화면에 띄우게 된다. 이때 통신이 좋지않으면 사용자의 회원정보를 확인할 수 없게 된다.

4) Daily Word



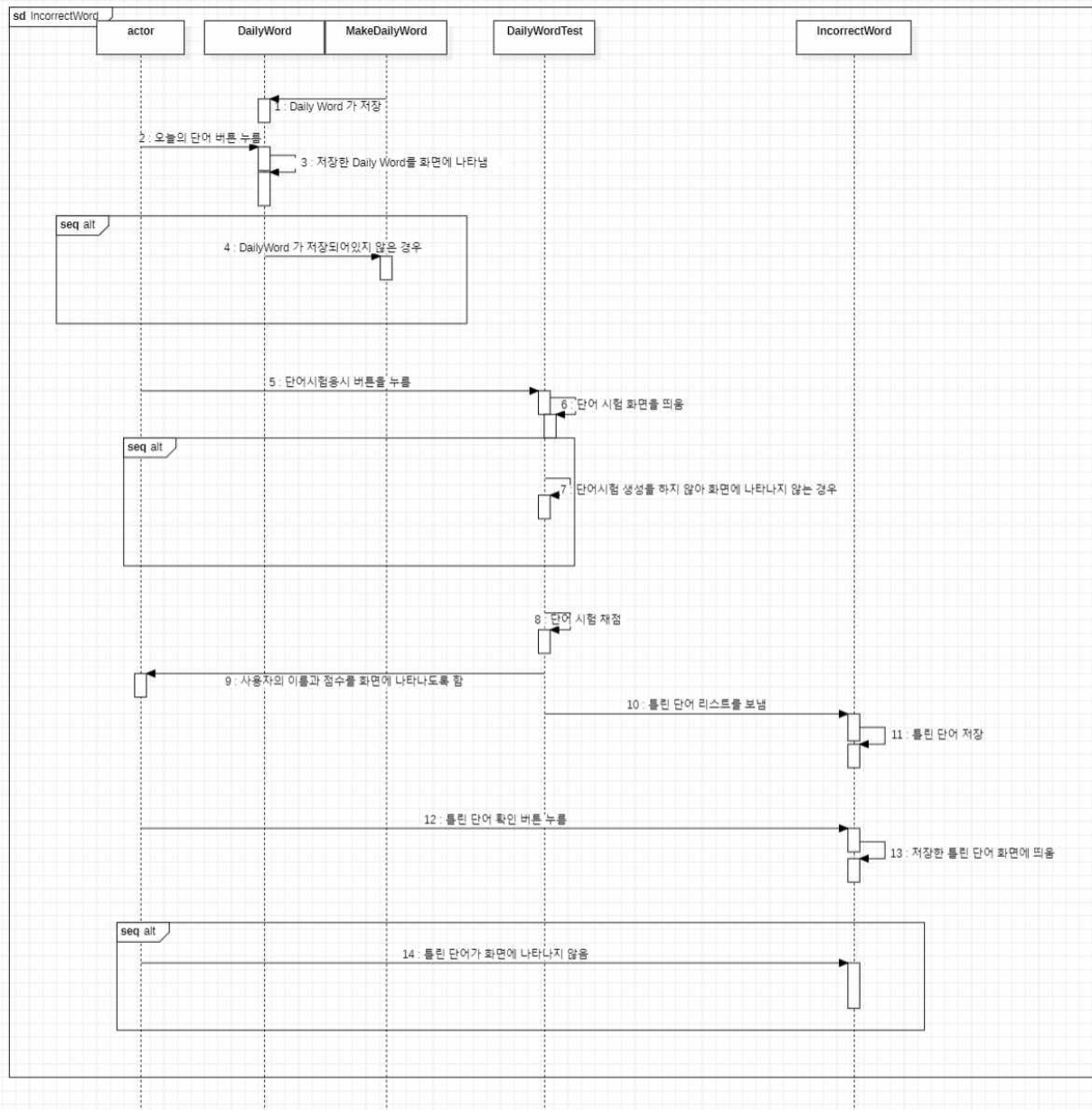
Daily Word Use case 에서의 Sequence Diagram이다. MakeDailyWord 클래스에서 생성한 오늘의 단어를 DailyWord 클래스에 저장하여 오늘의 단어를 외울 수 있도록 한다. 그리고 User이 오늘의 단어 버튼을 누르게 되면 DailyWord는 저장한 단어들을 화면에 띄워서 User이 단어를 외울 수 있도록 한다. 만약 사용자가 버튼을 눌러도 단어들이 저장되어 있지 않은 경우 MakeDailyWord 클래스에서 단어를 생성하도록 요청한다.

5) Daily Word Test



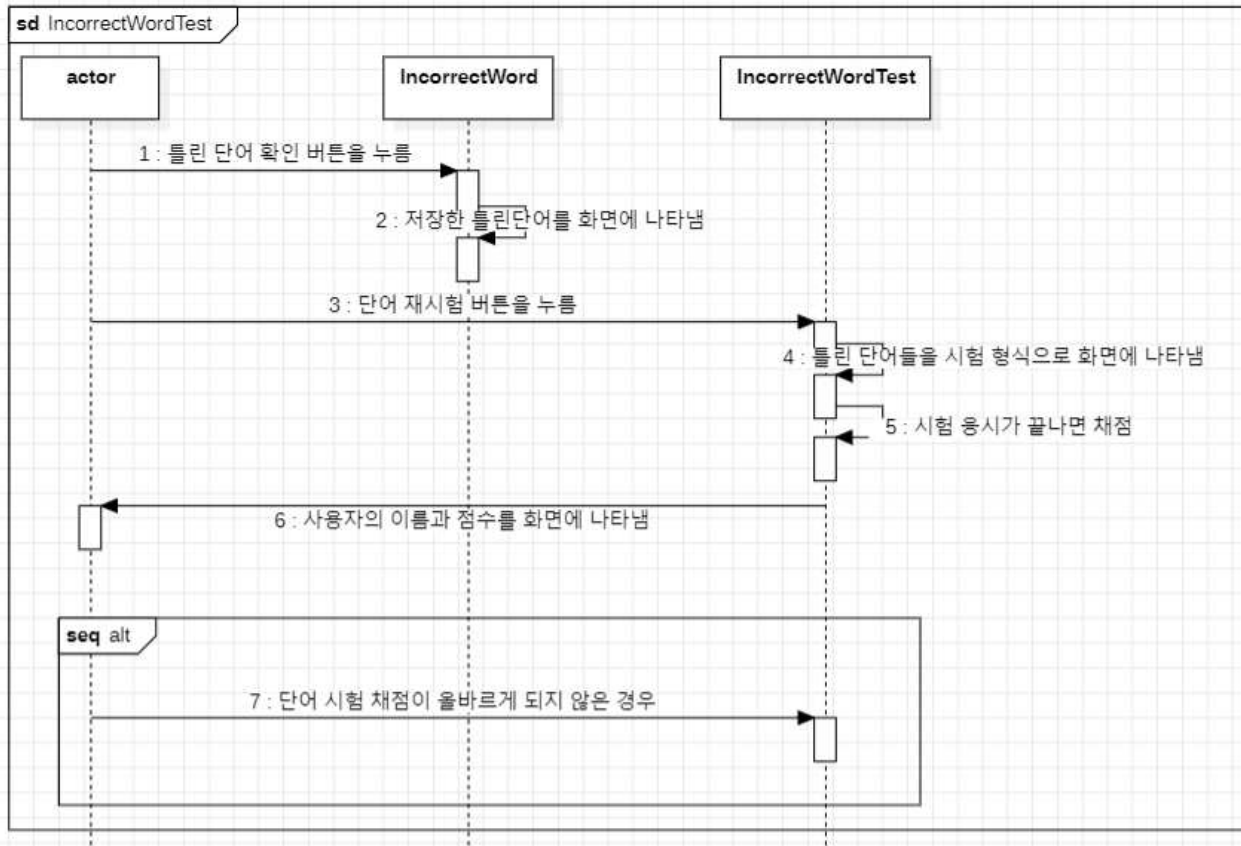
Daily Word Test Use case 에서의 Sequence Diagram이다. User이 단어시험응시 버튼을 누르게 되면 DailyWordTest 클래스에선 단어 시험을 생성하여 화면에 띄우게 된다. 이때 단어시험 생성을 하지않아 화면에 나타나지 않는 경우 DailyWordTest 는 단어 시험을 생성하여 화면에 나타날 수 있도록 한다. 단어 시험을 응시하면 채점하여 사용자의 이름과 점수를 화면에 나타나도록 한다.

6) Incorrect Word



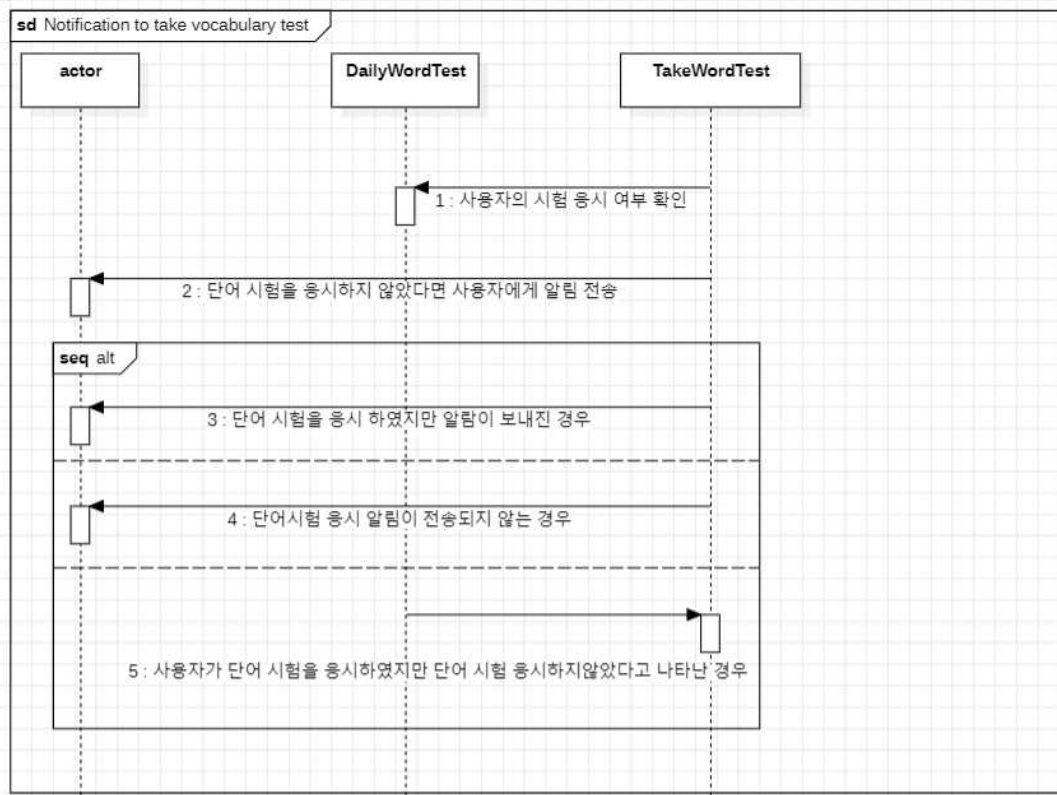
Incorrect Word Use case 에서의 Sequence Diagram이다. 단어 시험 응시를 하게 되면 틀린 단어들을 IncorrectWord 클래스에 저장하게 된다. 그리고 User이 틀린 단어 확인 버튼을 누르게 되면 화면에 틀린 단어를 나타나게 한다.

7) Incorrect Word Test



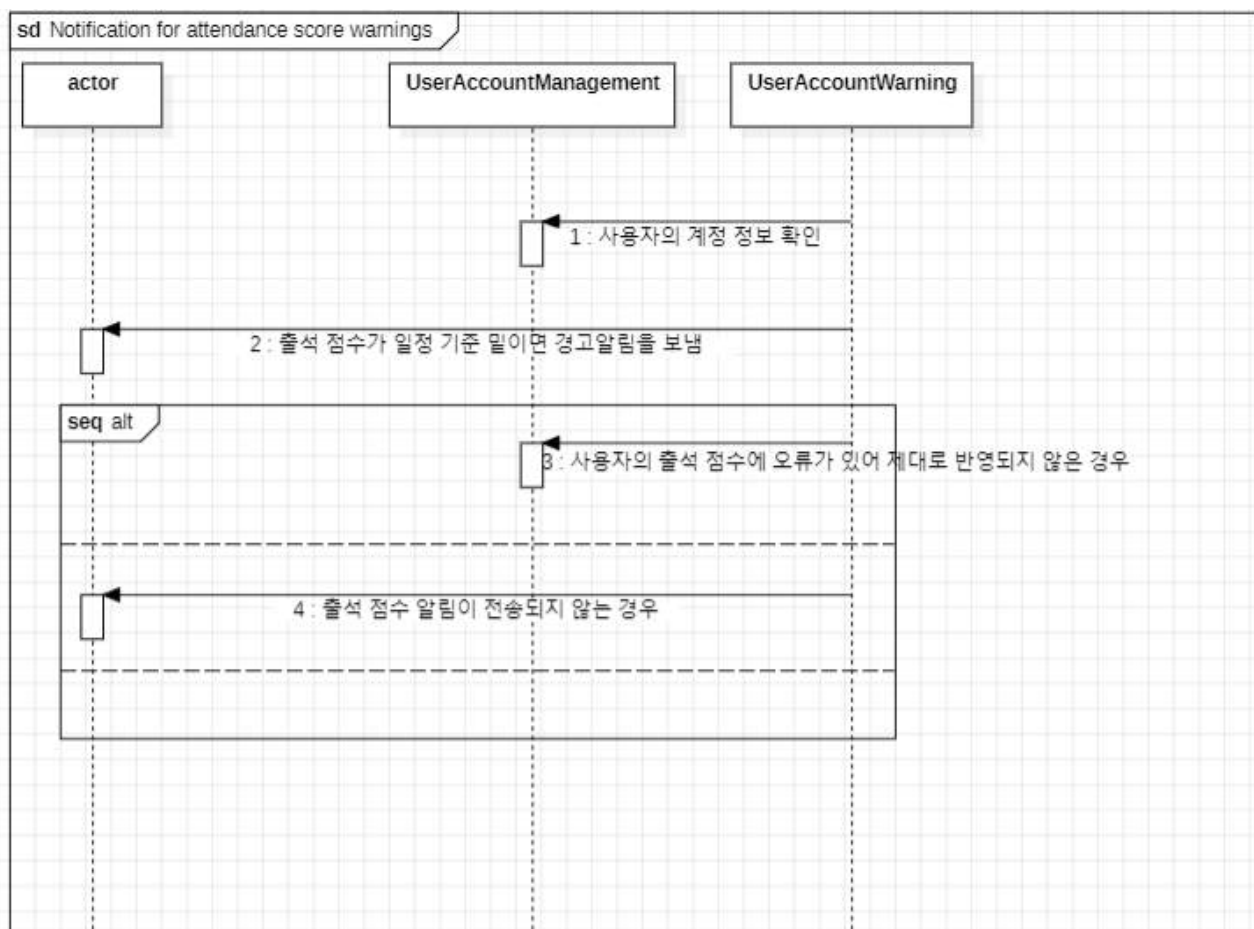
Incorrect Word Test Use case 에서의 Sequence Diagram이다. 틀린 단어 암기가 끝나게 되면 User은 단어 재시험 버튼을 누르게 되면 IncorrectWordTest 클래스에서 틀린 단어들을 시험 형식으로 화면에 나타나도록 하고 User의 시험 응시가 끝나면 사용자의 이름과 점수를 화면에 나타나도록 한다.

8) Notification to take vocabulary test



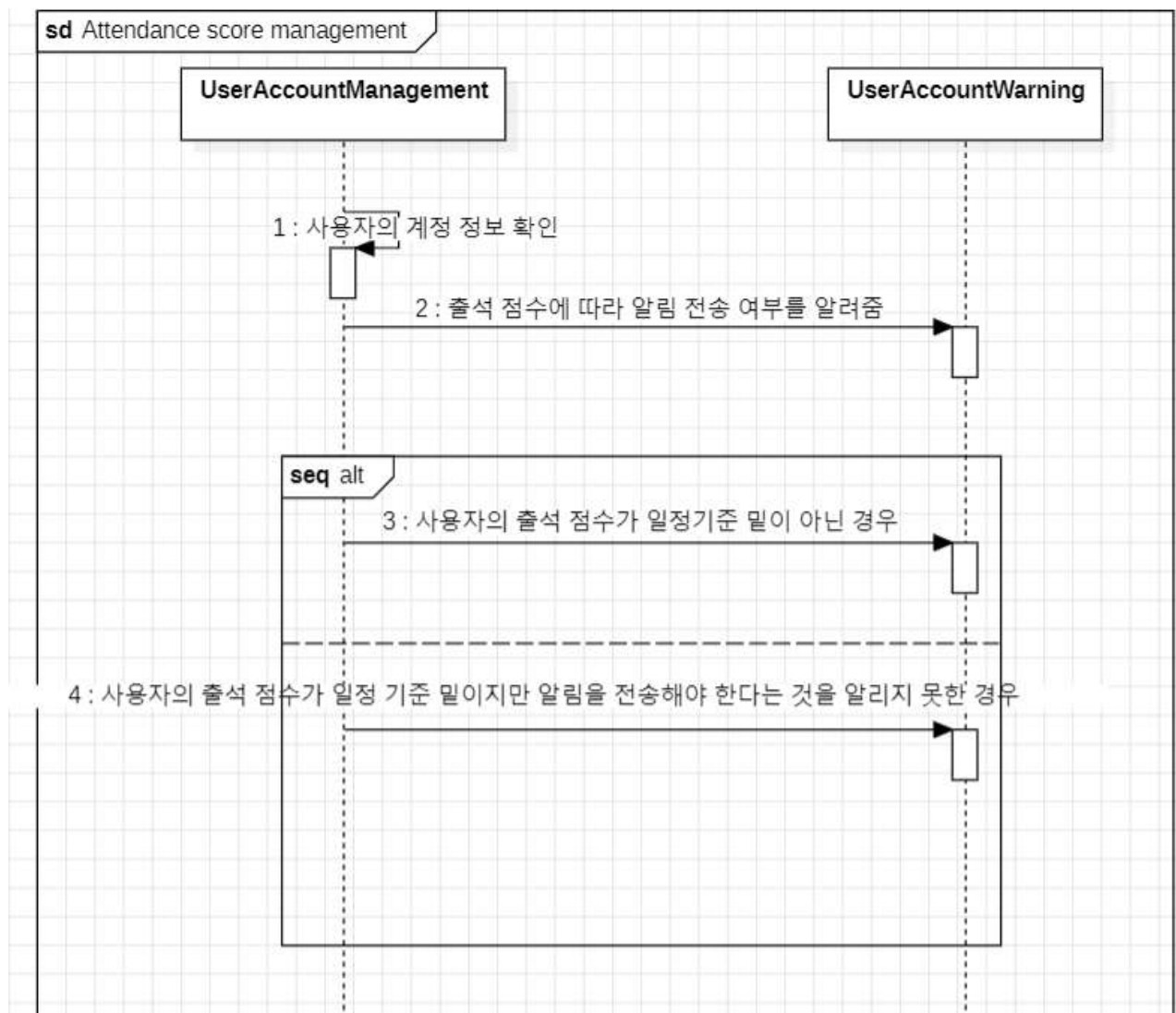
Notification to take vocabulary test Test Use case 에서의 Sequence Diagram이다. Actor의 단어 시험 응시 여부를 확인하고 사용자에게 단어 시험을 응시하라는 알림을 전송한다.

9) Notification for attendance score warnings



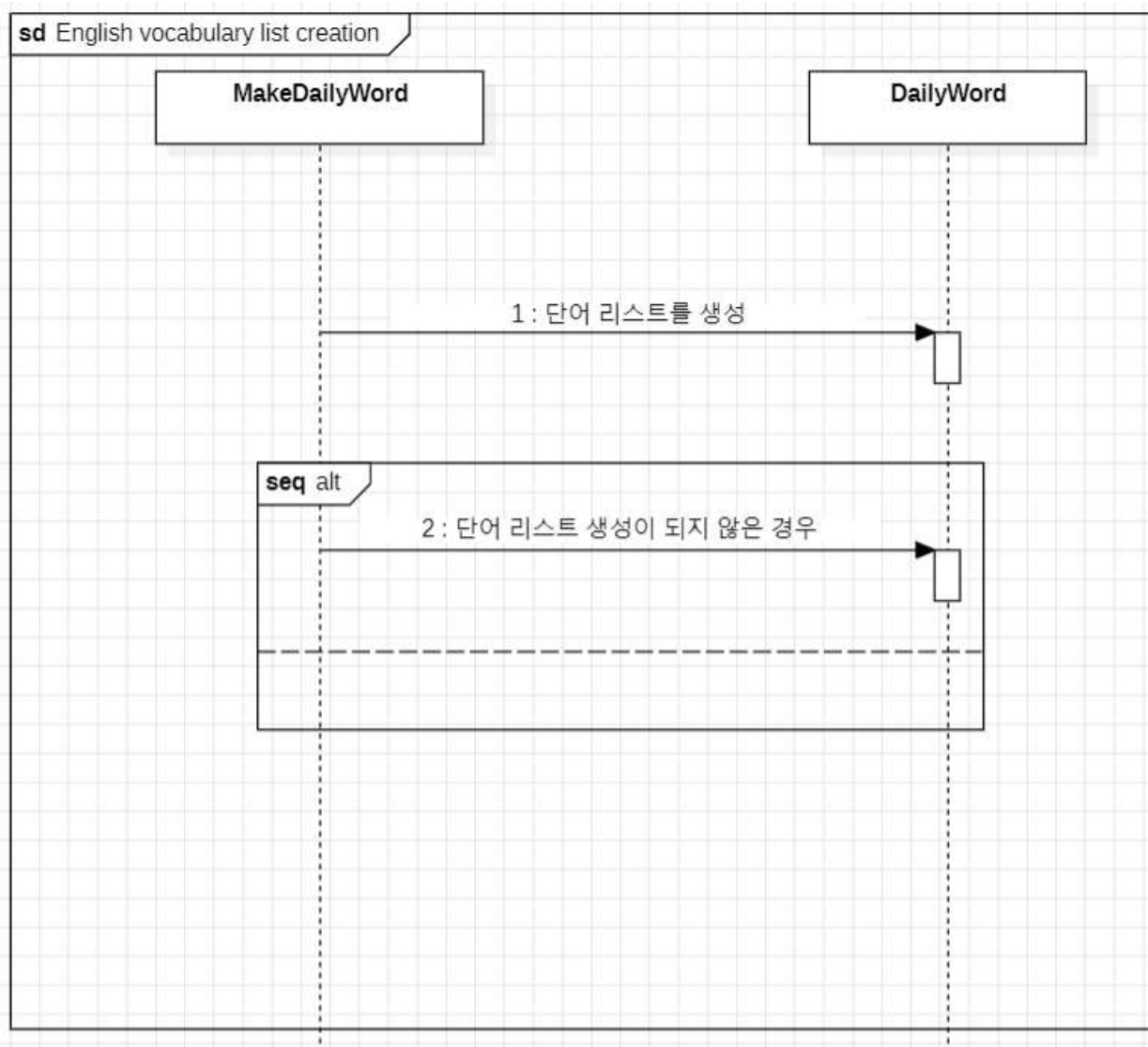
Notification for attendance score warnings Use case 에서의 Sequence Diagram이다. Actor의 계정 정보를 확인하여 사용자의 출석점수에 따라 사용자에게 출석점 수 경고 알림을 전송한다. 사용자의 출석 점수에 오류가 있거나 출석 점수 알림이 전송되지 않은 경우 다시 클래스를 확인해볼 필요가 있다.

10) Attendance score management



Attendance score management Use case 에서의 Sequence Diagram이다. Actor의 계정 정보를 확인하여 사용자의 출석점수에 따라 사용자에게 출석점수 경고 알림을 전송해야한다는 것을 UserAccountWarning 클래스에 알린다. 사용자의 출석 점수가 일정 기준 밑이 아닌 경우나 사용자의 출석 점수가 일정 기준 밑이지만 알림을 전송해야 한다는 것을 알리지 못한 경우 다시 클래스를 확인해볼 필요가 있다.

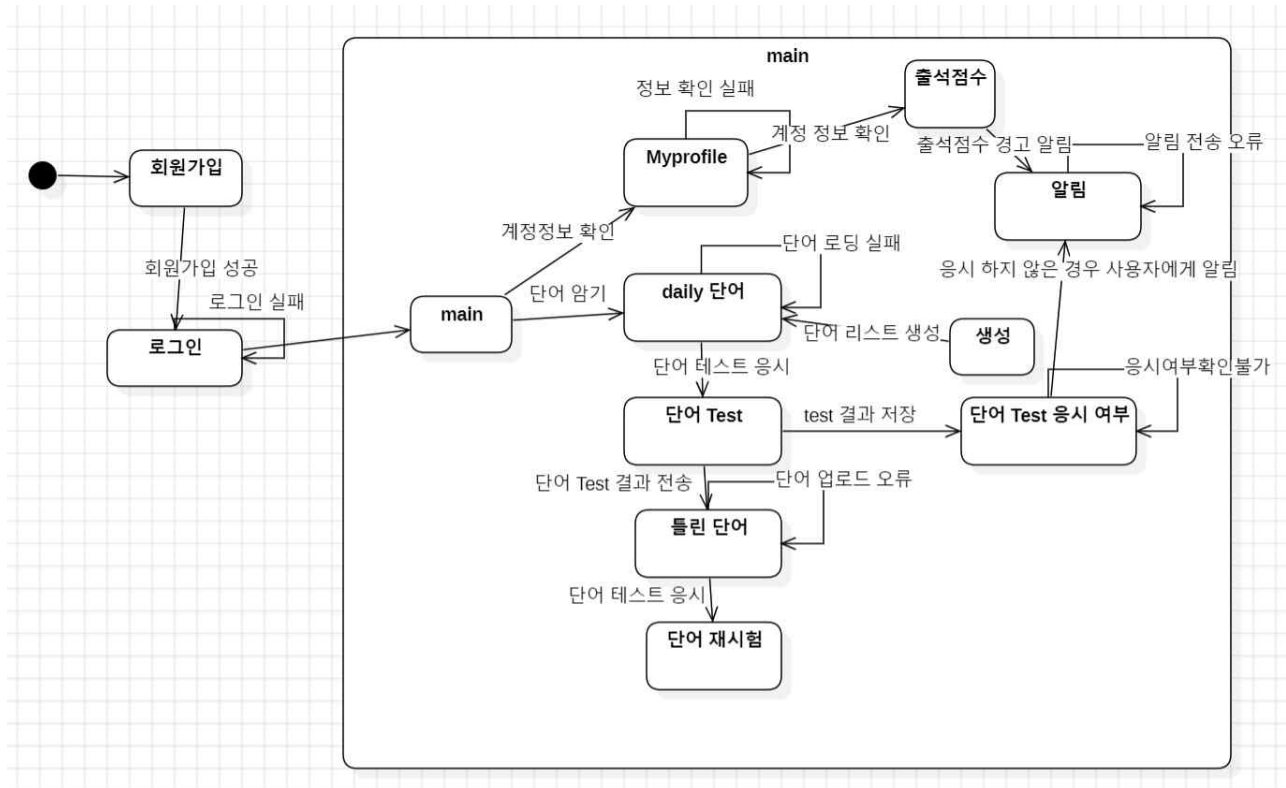
11) English vocabulary list creation



English vocabulary list creation Use case 에서의 Sequence Diagram이다. MakeDailyWord 클래스에서 단어 리스트를 생성하여 DailyWord에 저장할 수 있도록 한다.

4. State machine diagram

- Draw state machine diagrams for the client and the server system.
- Explain each state machine diagram.
- 12pt, 160%.



회원가입을 하여 로그인에 성공하면 main화면에서 단어 리스트를 보고 단어를 암기하고 단어 test를 진행한다. 이때 단어 test를 응시하였다는 정보를 저장하고 저장이 되지않은 경우는 사용자에게 알림이 가게 된다. 또한 단어 test를 응시하지 않으면 출석점수가 감점 되는데 이러한 경우 일정기준 밑으로 출석점수가 내려가게 되면 또한 알림이 전송된다. 그리고 단어 Test 결과에 따라 틀린 단어들만 다시 외운 후 단어 재시험을 응시할 수 있도록 한다.

5. Implementation requirements

- Describe operating environments to implement your system.
- 12pt, 160%.

CPU는 Intel PENTIUM IV 이상이어야 하고 RAM은 1G 이상이어야 한다.

또한 통신이 가능한 상태이고 네트워크에 연결되어있어야 하며 코드를 실행시킬 수 있는 환경이어야 한다.

6. Glossary

- Specifically describe all of the terms used in this documents.
- 12pt, 160%.

Terms	Descirption
생성	사용자가 외워야 할 오늘의 단어 List를 생성하는 것을 의미한다.
출석점수	사용자가 회원가입을 하면 사용자에게 100점이라는 출석점수가 부과된다. 그리고 사용자가 매일 단어 시험을 응시하지 않으면 100점에서 10점씩 감점 된다.
일정기준	사용자의 출석점수가 일정 기준 밑으로 내려가게 되면 경고 알림을 받게 되는데 이때 점수 기준은 50점, 30점을 기준으로 한다.

7. References

- Describe all of your references (book, paper, technical report etc).
- 12pt, 160%.