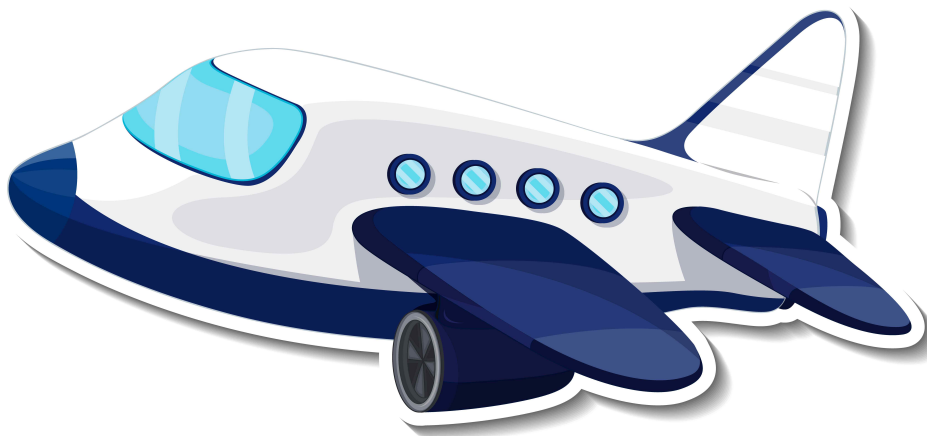


3. Design

all-travel



Student No	22012237
Name	윤우탁
E-mail	ywt0100@naver.com

[Revision history]

Revision date	Version #	Description	Author
06/07/2024	1.00	first document	
06/14/2024	1.01	개발 도구 변경	

= Contents =

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

1. Introduction

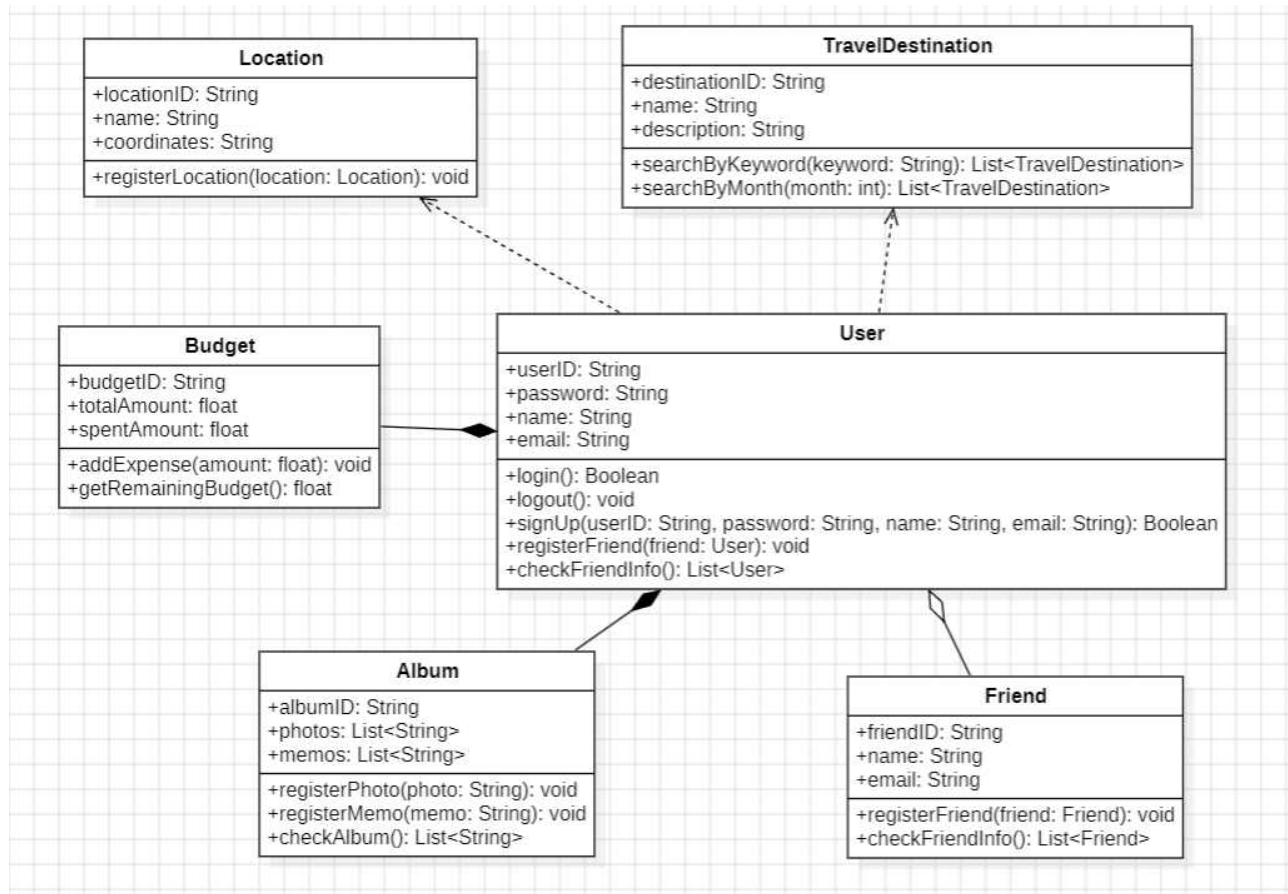
- Summarize the contents of this document.
- Describe the important points of your design.
- 12pt, 160%.

이 앱을 통해 사용자는 키워드별, 월별로 여행지를 추천받을 수 있고, 추천받은 여행지의 주요 명소와 간단한 소개를 확인할 수 있다. 원하는 장소를 등록하면 등록된 모든 장소를 방문하는 최적의 경로를 추천받을 수 있다. 또한 예산 관리를 위해 전체 예산을 등록하고 개인/단체 지출을 등록하면 잔액과 사용 내용을 확인할 수 있다. 간단한 메모와 사진 등록을 통해 여행지의 추억을 남길 수 있으며 다양한 기능을 친구 등록을 통해 친구들과 공유할 수 있다.

해당 문서는 Analysis에 이은 Design 문서로 class diagram, sequence diagram, state machine diagram과 같은 diagram을 통해서 시스템을 설명하고, implementation requirements를 통해 시스템을 실행하기 위한 운영 환경을 설명한다.

2. Class diagram

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



1) User

Attributes
+userID: String : 사용자의 아이디
+password: String : 사용자의 비밀번호
+name: String : 사용자의 이름
+email: String : 사용자의 이메일
Methods
+login(): Boolean : 사용자가 로그인 시도
+logout(): void : 사용자가 로그아웃
+signUp(userID: String, password: String, name: String, email: String): Boolean : 사용자가 회원가입
+registerFriend(friend: User): void : 사용자가 친구 등록
+checkFriendInfo(): List<User> : 사용자가 친구 정보 확인

2) TravelDestination

Attributes
+destinationID: String : 여행지 ID
+name: String : 여행지 이름
+description: String : 여행지 설명
Methods
+searchByKeyword(keyword: String): List<TravelDestination> : 키워드로 여행지 검색
+searchByMonth(month: int): List<TravelDestination> : 월로 여행지 검색

3) Location

Attributes
+locationID: String : 장소 ID
+name: String : 장소 이름
+coordinates: String : 장소 좌표
Methods
+registerLocation(location: Location): void : 장소 등록

4) Budget

Attributes
+budgetID: String : 예산 ID
+totalAmount: float : 총 예산 금액
+spentAmount: float : 사용한 금액
Methods
+addExpense(amount: float): void : 지출 추가
+getRemainingBudget(): float : 남은 예산 확인

5) Album

Attributes
+albumID: String : 앨범 ID
+photos: List<String> : 사진 목록
+memos: List<String> : 메모 목록
Methods
+registerPhoto(photo: String): void : 사진 등록
+registerMemo(memo: String): void : 메모 등록
+checkAlbum(): List<String> : 앨범 확인

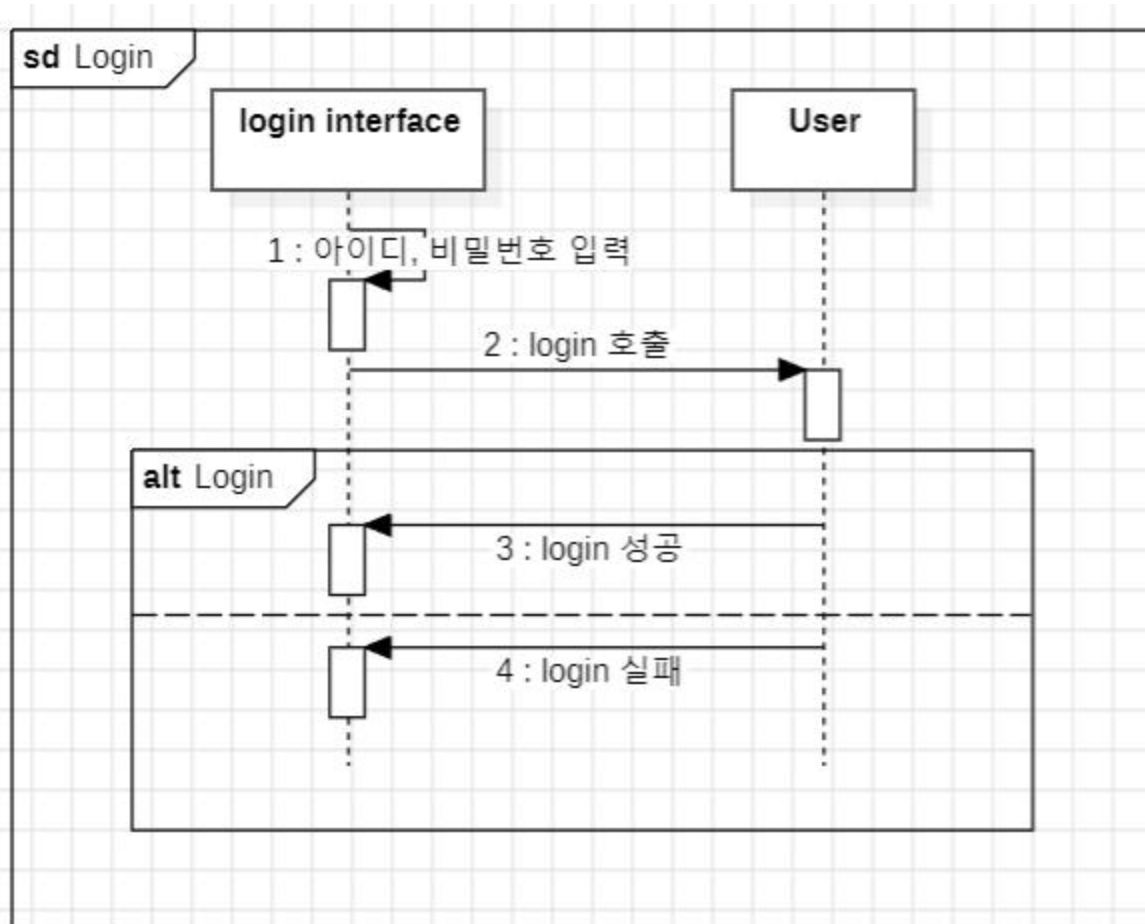
6) Friend

Attributes
+friendID: String : 친구 ID
+name: String : 친구 이름
+email: String : 친구 이메일
Methods
+registerFriend(friend: Friend): void : 친구 등록
+checkFriendInfo(): List<Friend> : 친구 정보 확인

3. Sequence diagram

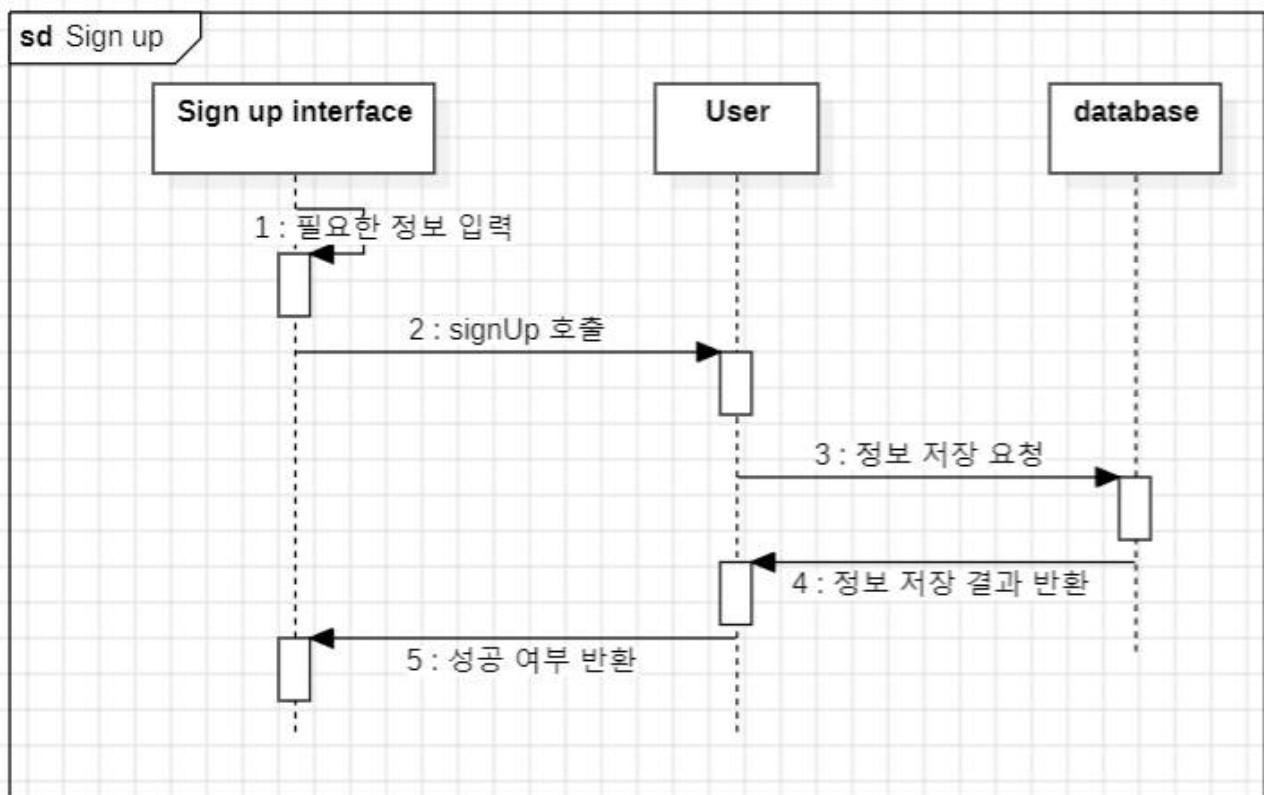
- Draw sequence diagrams for the whole functions of your system (this is related to the use cases you made in the Conceptualization phase).
- Explain each sequence diagram.
- 12pt, 160%.

1) Login



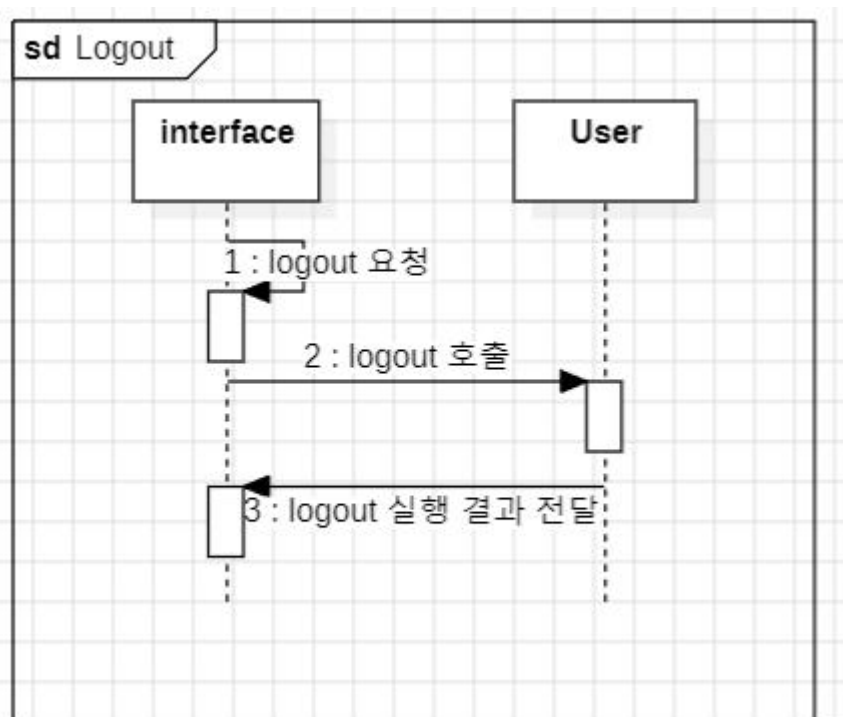
사용자가 로그인 인터페이스에서 아이디, 비밀번호를 입력 합니다. 입력된 아이디와 비밀번호를 가지고 User 클래스의 login() 메소드가 호출 됩니다. login 메소드 내부에서 사용자의 인증 정보가 확인 되면 로그인 성공 여부를 나타내는 boolean 값을 반환 합니다.

2) Sign up



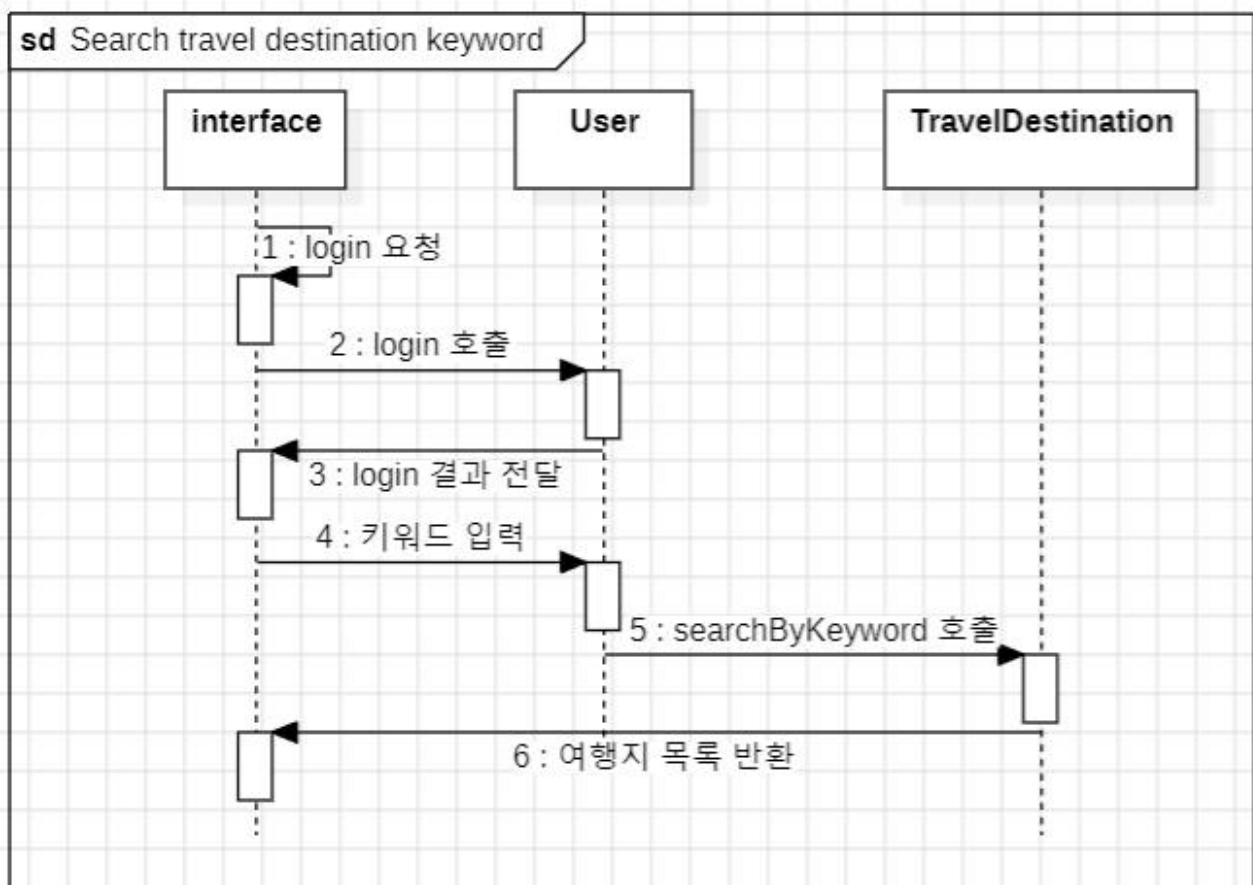
사용자가 회원가입 인터페이스에서 회원가입에 필요한 정보를 입력 합니다. 입력된 정보를 가지고 User 클래스의 signUp() 메소드가 호출 됩니다. User 클래스는 사용자가 입력한 정보를 데이터베이스에 저장요청 합니다. 데이터베이스는 사용자가 입력한 정보 저장 결과를 User 클래스에 반환 합니다. User 클래스는 회원가입 인터페이스에 성공 여부를 반환 합니다.

3) Logout



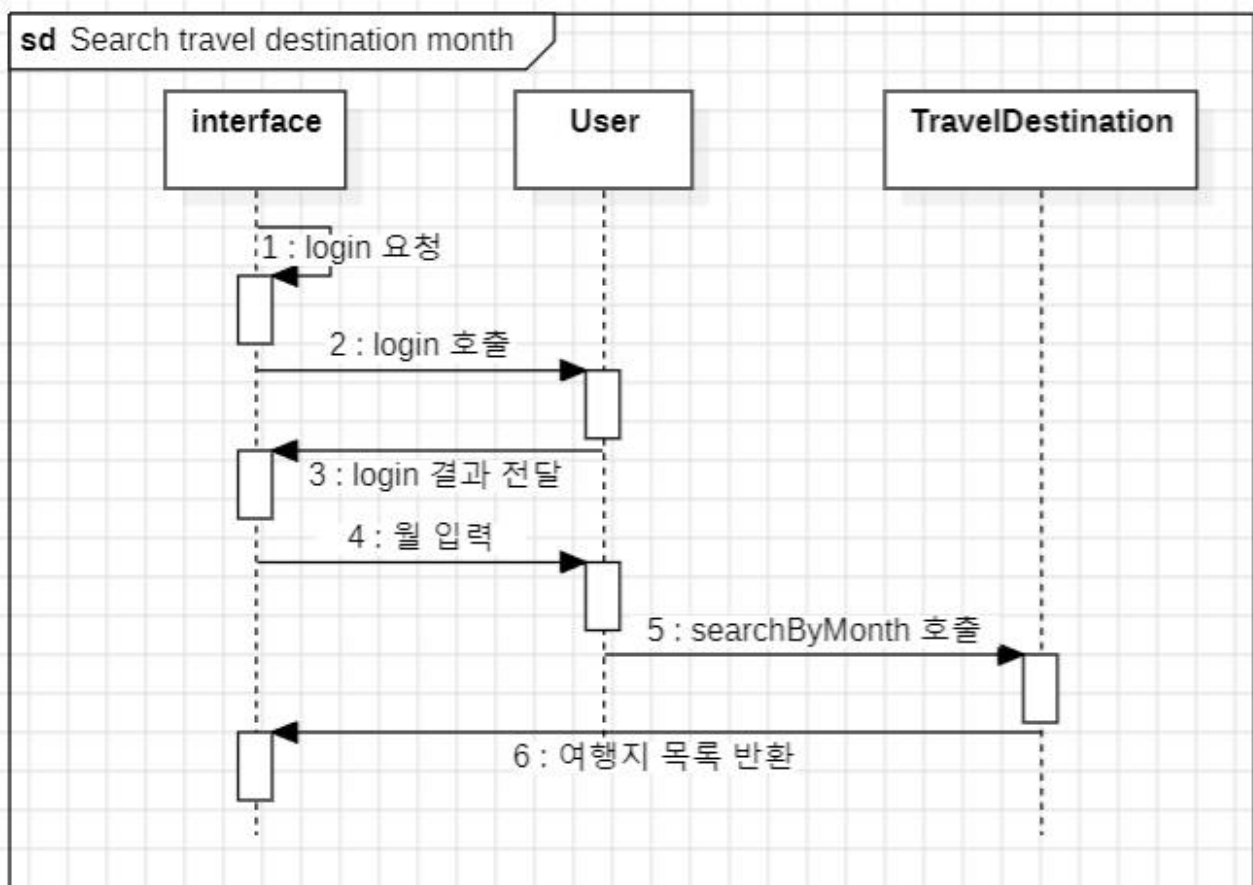
사용자가 인터페이스에서 로그아웃을 요청하면 User 클래스의 logout() 메소드가 호출 됩니다. logout() 메소드가 실행된 결과를 인터페이스에 알립니다.

4) Search travel destination keyword



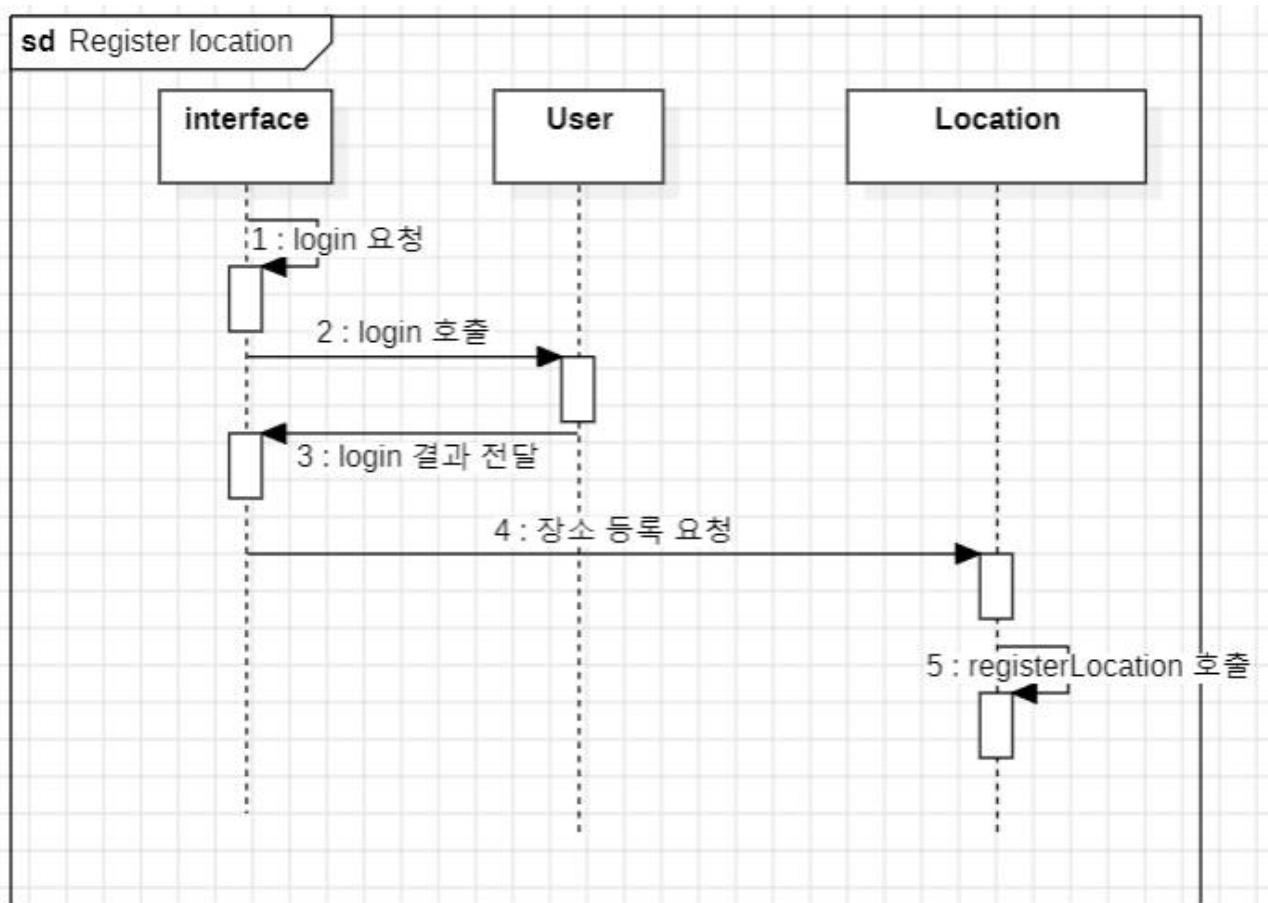
사용자가 interface에서 login을 요청합니다. User 클래스의 login() 메소드가 호출 됩니다. login() 메소드의 결과가 interface에 전달 됩니다. 로그인에 성공한 사용자는 원하는 키워드로 여행지를 검색할 수 있습니다. interface에서 키워드를 입력하면 User 클래스에서 searchByKeyword() 메소드가 호출됩니다. TravelDestination 클래스는 해당 키워드에 맞는 여행지 목록을 검색하여 반환 합니다.

5) Search travel destination month



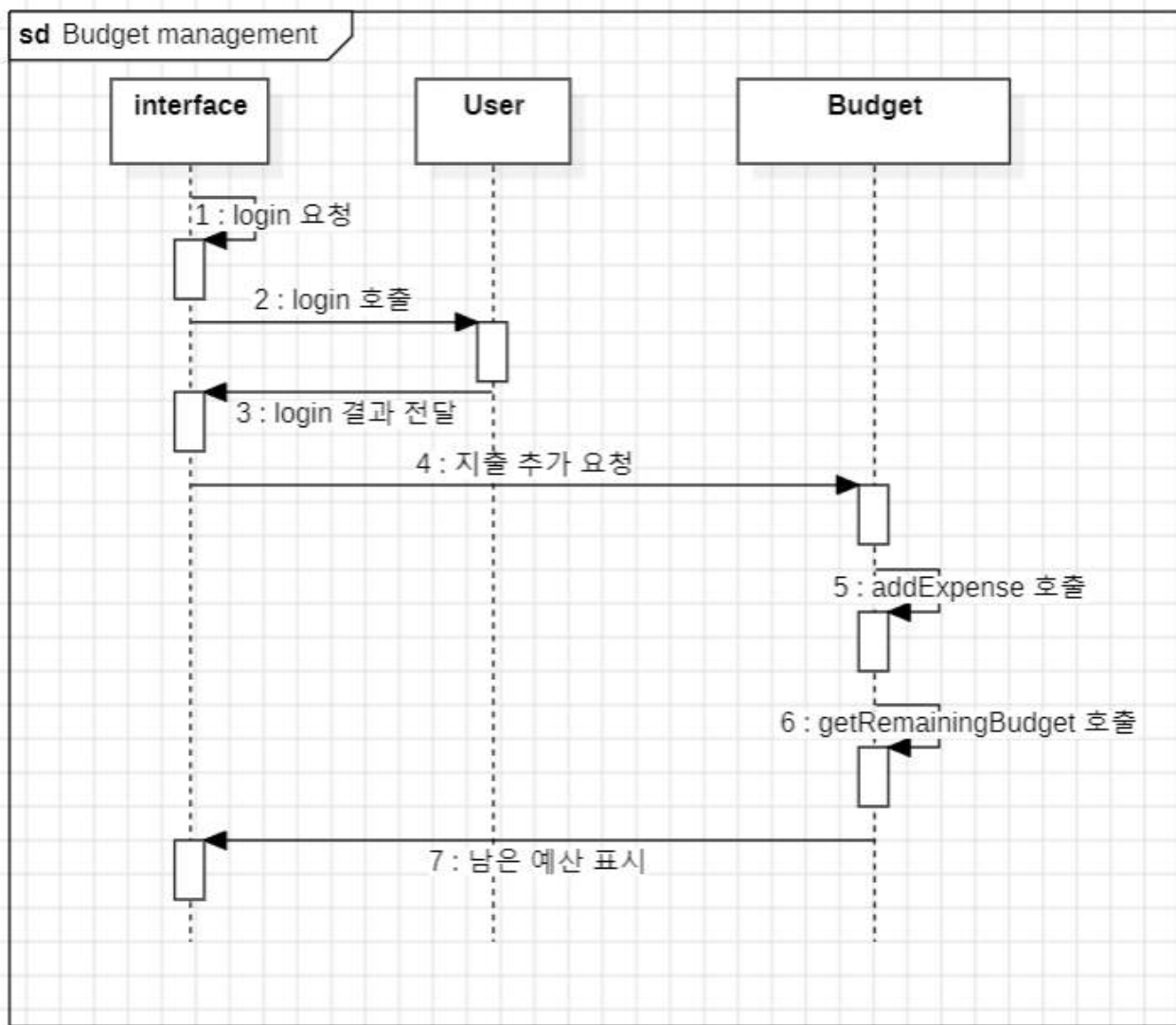
사용자가 interface에서 login을 요청합니다. User 클래스의 login() 메소드가 호출 됩니다. login() 메소드의 결과가 interface에 전달 됩니다. 로그인에 성공한 사용자는 원하는 월로 여행지를 검색할 수 있습니다. interface에서 월을 입력하면 User 클래스에서 searchByMonth() 메소드가 호출됩니다. TravelDestination 클래스는 해당 월에 맞는 여행지 목록을 검색하여 반환 합니다.

6) Register location



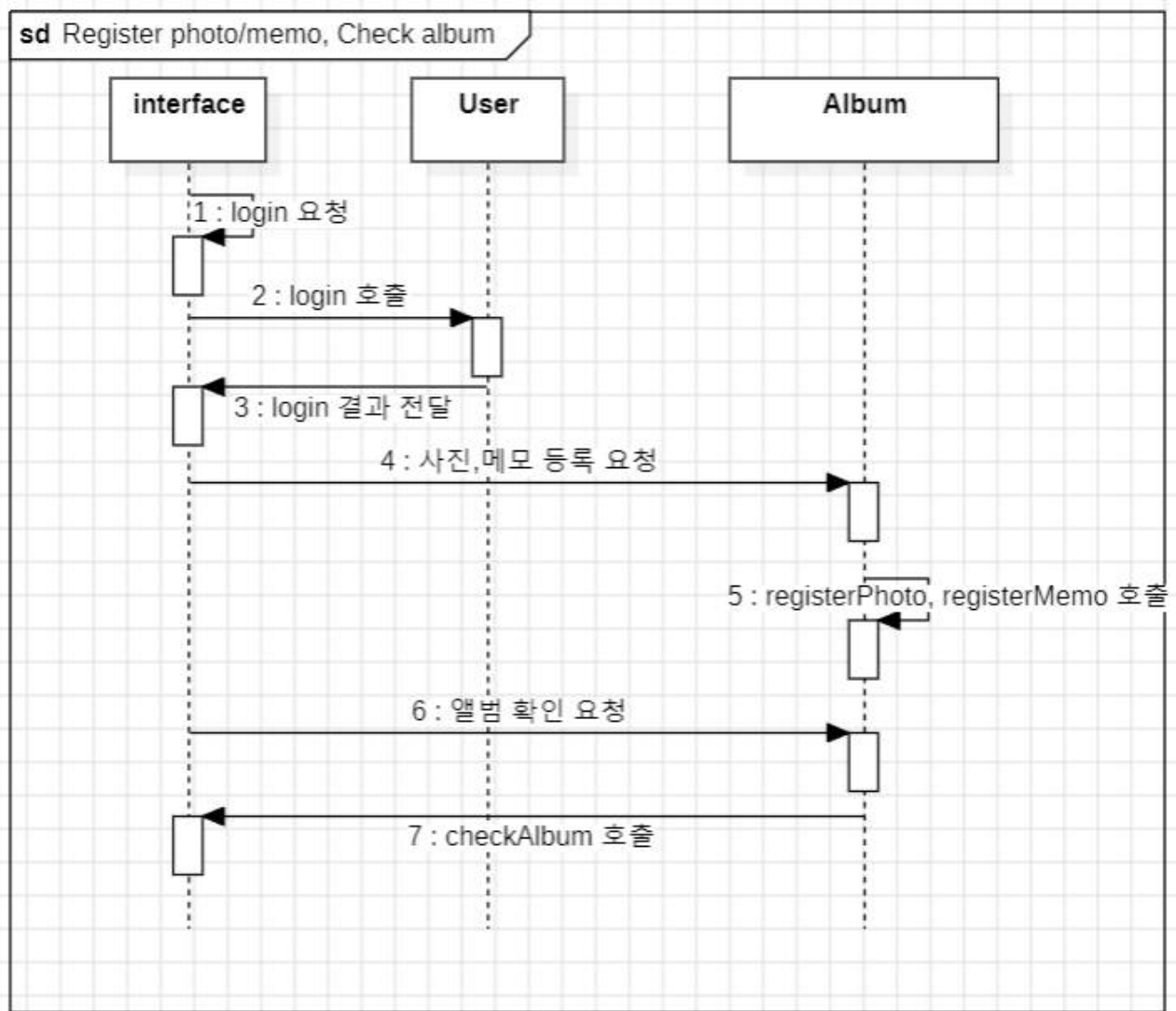
사용자가 interface에 login을 요청합니다. User 클래스의 login() 메소드가 호출 됩니다. login() 메소드의 결과가 interface에 전달 됩니다. 로그인에 성공한 사용자는 경로 추천을 위한 장소를 등록할 수 있습니다. 사용자는 interface에서 장소 등록 요청을 합니다. Location 클래스는 registerLocation() 메소드를 호출합니다. Location 클래스는 장소가 유효한지 판단 후 등록합니다.

7) Budget management



사용자가 interface에 login을 요청합니다. User 클래스의 login() 메소드가 호출 됩니다. login() 메소드의 결과가 interface에 전달 됩니다. 로그인에 성공한 사용자는 예산 관리 기능을 사용할 수 있습니다. 사용자가 interface에서 지출추가(전체 예산 포함)를 요청 합니다. Budget 클래스에서 addExpense() 메소드가 호출됩니다. getRemainingBudget() 메소드가 호출 됩니다. 계산된 남은 예산이 interface에 표시 됩니다.

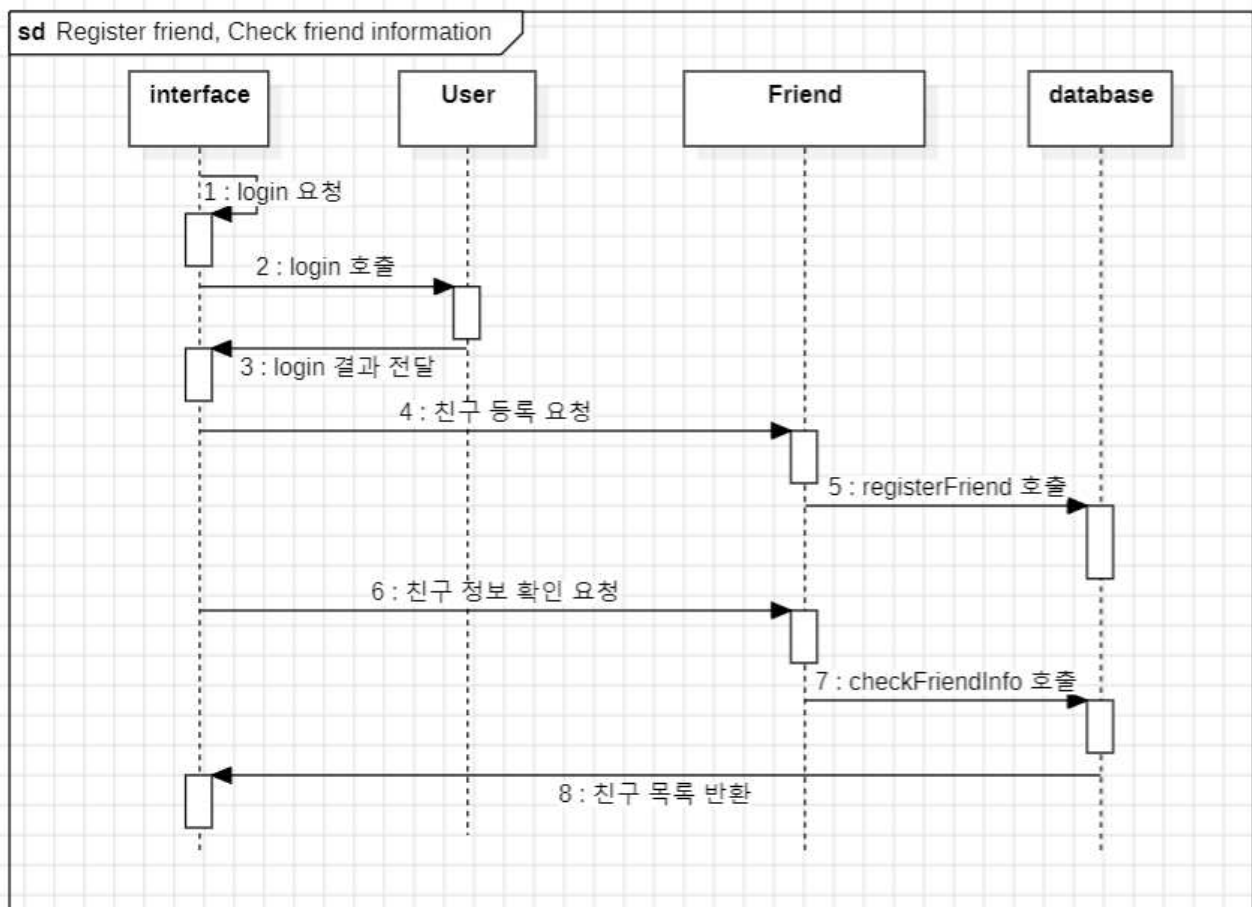
8) Register photo/memo, Check album



사용자가 interface에 login을 요청합니다. User 클래스의 login() 메소드가 호출 됩니다. login() 메소드의 결과가 interface에 전달 됩니다. 로그인에 성공한 사용자는 사진 /메모 등록 기능, 앨범 확인 기능을 사용할 수 있습니다. 사용자는 interface에서 사진, 메모 등록 요청을 합니다. Album 클래스가 registerPhoto() 메소드와 registerMemo() 메소드를 호출해 사진과 메모를 등록합니다.

사용자가 앨범 확인 요청을 합니다. Album 클래스는 checkAlbum() 메소드를 호출해 앨범을 interface에 표시 합니다.

9) Register friend, Check friend information

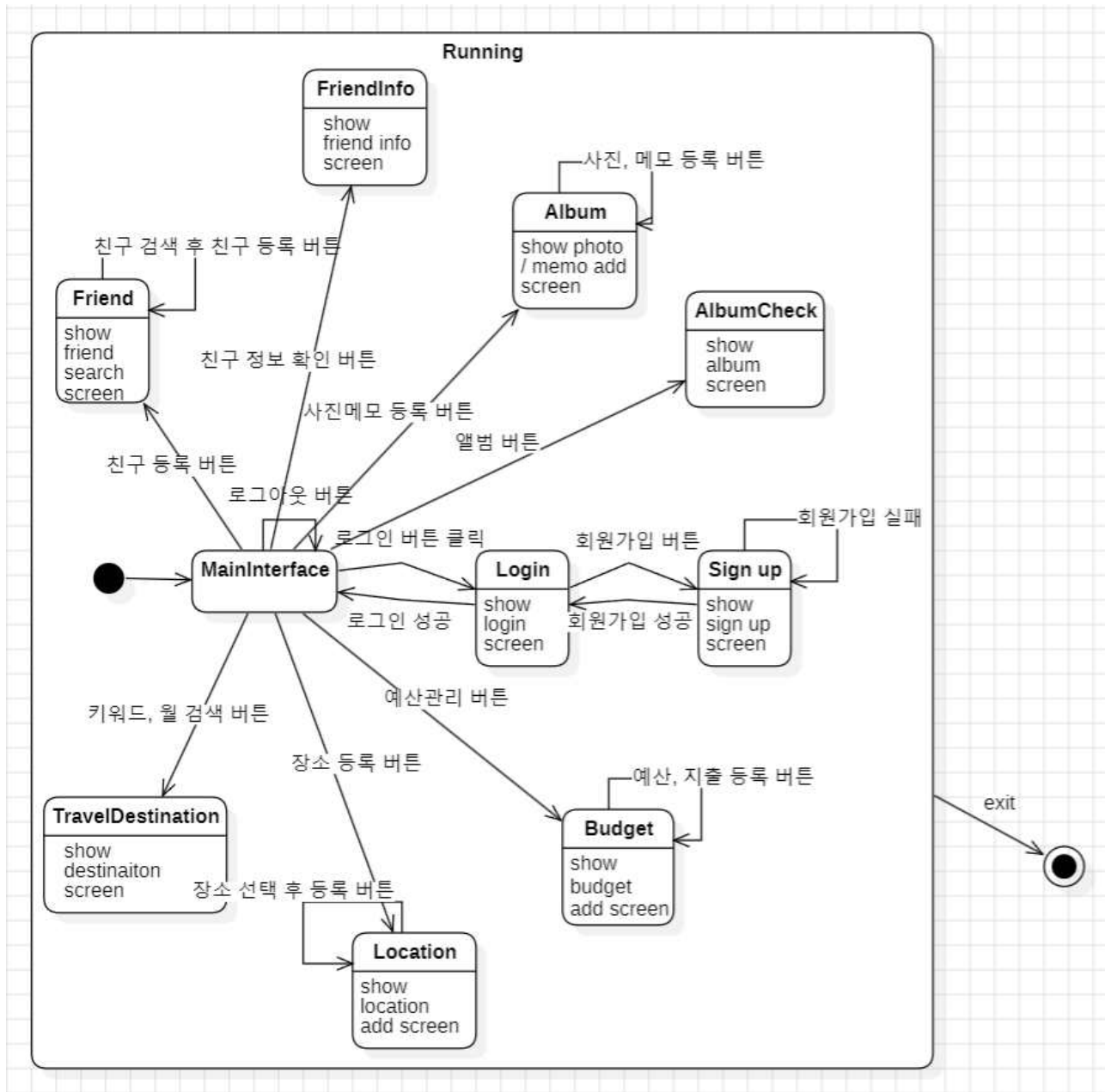


사용자가 interface에 login을 요청합니다. User 클래스의 login() 메소드가 호출 됩니다. login() 메소드의 결과가 interface에 전달 됩니다. 로그인에 성공한 사용자는 친구 등록 기능과 친구 정보 확인 기능을 사용할 수 있습니다. 사용자가 interface에서 친구 등록 요청을 합니다. Friend 클래스는 registerFriend() 메소드를 호출해 database에 친구 정보를 등록 합니다.

사용자가 interface에서 친구 정보 확인 요청을 합니다. Friend 클래스는 checkFriendInfo() 메소드를 호출해 database에서 interface로 친구 목록을 반환 하도록 합니다.

4. State machine diagram

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



시스템 실행 시 메인 화면에서 시작하며 로그인 버튼을 누를 경우 로그인 화면으로 이동합니다. 회원가입 버튼을 누를 시 회원가입 화면으로 이동합니다. 회원 가입이 성공하면 로그인 화면으로 이동하고, 실패할 경우 회원가입 화면에 남아 있게 됩니다. 로그인 성공시 메인화면으로 이동하고, 로그인 버튼이 로그아웃으로 바뀝니다. 로그아웃 버튼을 누를 경우 로그아웃이 되며 로그아웃 버튼이 로그인 버튼으로 바뀝니다. 메인

화면에서 키워드 또는 월을 입력하고 검색 버튼을 누르면 키워드, 월에 맞는 여행지 목록 화면으로 이동하게 됩니다. 메인화면에서 장소 등록 버튼을 누르면 장소 등록 화면으로 이동하며 원하는 장소를 등록할 수 있습니다. 메인화면에서 예산 관리 버튼 클릭할 경우 예산 관리 화면으로 이동하게 되며 총 예산과 지출을 입력하면 남은 예산을 확인할 수 있습니다. 메인화면에서 친구 등록 버튼을 누를 경우 친구 등록 화면으로 이동해 친구를 등록할 수 있으며 친구 정보는 메인화면의 친구 정보 확인 버튼을 통해 확인할 수 있습니다. 메인화면에서 사진/메모 등록 버튼을 누르면 사진, 메모를 등록해 앨범을 만들 수 있으며 만들어진 앨범은 메인화면의 앨범 버튼을 통해 확인할 수 있습니다.

5. Implementation requirements

- | |
|---|
| <ul style="list-style-type: none">- Describe operating environments to implement your system.- 12pt, 160%. |
|---|

CPU : Intel i5 이상

RAM : 최소 4GB

저장공간 : 최소 100MB의 여유공간

운영체제 : windows 10 이상

개발도구 : Python

6. Glossary

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

term	description
method	클래스에 구현된 함수
class diagram	class 라는 객체 지향 설계 단위를 이용하여 시스템의 구조를 표현한 diagram
sequence diagram	객체간의 동적 상호작용을 시간적 개념을 중심으로 모델링 한 diagram
state machine diagram	객체 lifetime 동안의 변환될 수 있는 모든 상태를 정리해둔 diagram

7. References

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

[OSSDesign] 06_Structural Modeling I

[OSSDesign] 07_Structural Modeling II

[OSSDesign] 08_Behavior modeling I

[OSSDesign] 09_Behavior Modeling II