

2. Analysis



| | |
|-------------|---------------------|
| Student No. | 22311977 |
| Name | 송서현 |
| E-mail | angels0315@daum.net |

[Revision history]

| Revision date | Version # | Description | Author |
|---------------|-----------|-------------|--------|
| 08/05/2025 | 1.0 | First Draft | 송서현 |
| 15/06/2025 | 2.0 | 즐거찾기 추가 | |
| | | | |
| | | | |
| | | | |
| | | | |

= Contents =

| | |
|-----------------------------------|----|
| 1. Introduction | 4 |
| 2. Use case analysis | 5 |
| 3. Domain analysis | 16 |
| 4. User Interface prototype | 18 |
| 5. Glossary | 19 |
| 6. References | 19 |

1. Introduction

1) Summary

여행 계획을 짤 때 갔던 곳이나 가게 이름이 생각나지 않아 불편함을 겪은 적이 있을 것이다. 기존 지도 앱이나 사진 앨범만으로는 여행 동선 전체를 기억하기 어려운 점이 있다.

이런 불편함을 해결하기 위해 사용자가 직접 장소를 마커를 표시하고 그 표시한대로 여행 코스를 저장할 수 있는 웹사이트를 고안하였다. ‘Tripiary’는 사용자가 직접 자신만의 개성적인 여행 코스를 짜서 언제든지 다시 확인 할 수 있고 마커에 메모를 작성하여 여행 당시의 감상도 적을 수 있다.

이를 통해 사용자는 자신만의 여행 코스를 저장하고 여행 계획을 세울 수 있으며, 다른 사람들의 코스도 구경할 수 있다.

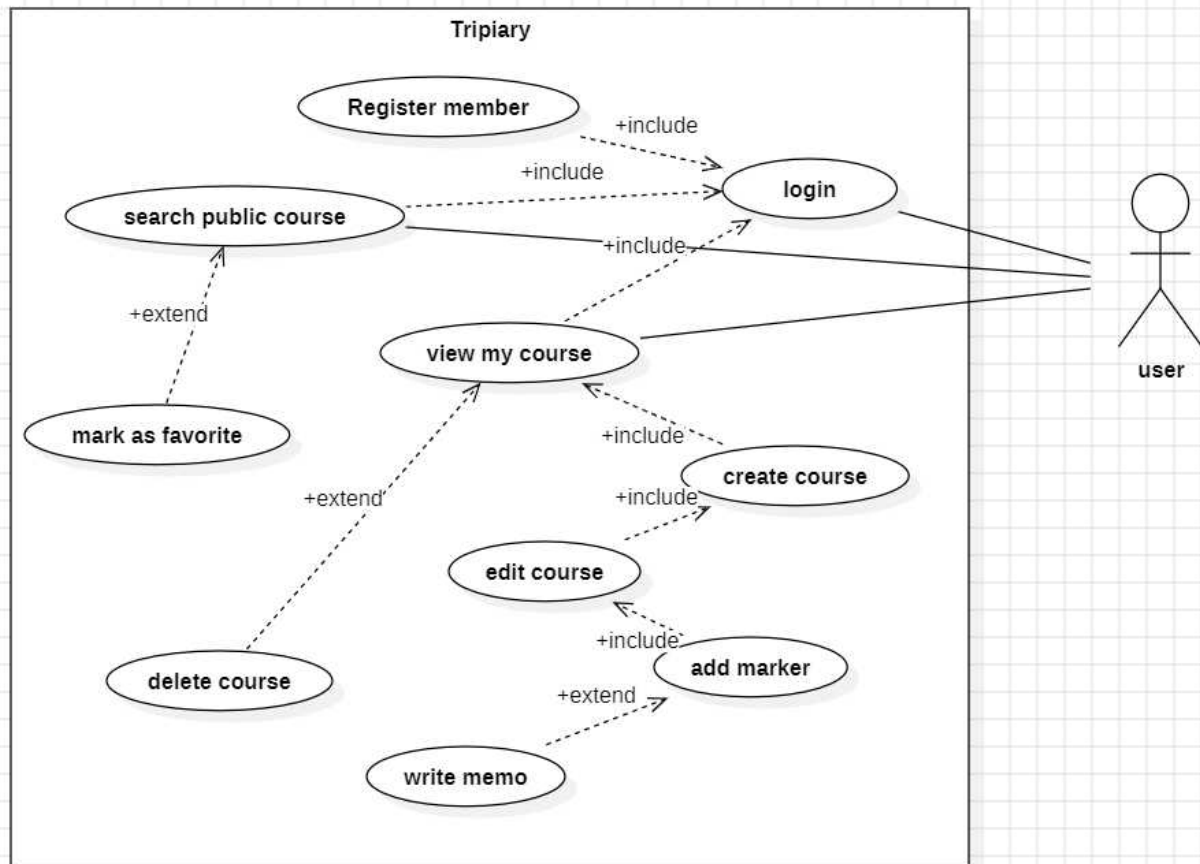
2) Describe the features of project

‘Tripiary’는 사용자의 위치 기반 정보와 시각적인 인터페이스를 활용하여, 여행의 순간을 기록하고 공유할 수 있도록 설계된 여행 코스 기록 플랫폼이다. 사용자는 지도 위에서 자신만의 여행 코스를 생성하고, 각 장소에 대해 검색을 통해 마커를 추가한 뒤 메모를 작성함으로써 여행의 기억을 개별 장소에 담을 수 있다. 이러한 마커들은 코스 단위로 저장되며, 사용자는 작성한 코스를 수정하거나 삭제할 수 있다. 코스를 ‘공개’로 설정하면 다른 사용자와 공유할 수 있으며, 모든 공개 코스는 카드 형태로 표시되어 열람이 가능하다.

특히 공개 코스에는 하트 아이콘을 통해 즐겨찾기 기능을 제공하고 있으며, 사용자 개인의 즐겨찾기 목록은 Firebase를 통해 저장되어 빠르게 접근할 수 있다. 화면 우측 하단에 고정된 하트 버튼을 통해 즐겨찾기한 코스만 필터링하여 열람하는 기능도 제공함으로써, 사용자 맞춤형 탐색이 가능하다. 로그인 및 회원가입 기능은 Firebase 인증 기반으로 작동하여, 로그인 여부에 따라 기능 접근성이 달라지도록 구성되어 있다. Tripiary는 지도 기반의 실용적인 기능성을 통해 여행을 의미 있게 기록할 수 있도록 한다.

2. Use case analysis

1) Use case diagram



2) Use case description

| Use Case #1 : Login | |
|-------------------------|--|
| GENERAL CHARACTERISTICS | |
| Summary | 사용자가 아이디, 비밀번호를 입력하여 로그인 성공/실패 여부를 확인하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |

| | |
|------------------------|---|
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사용자가 회원가입이 된 상태여야 한다. |
| Trigger | 사용자가 Tripiary에 로그인 하려 할 때 |
| Success Post Condition | 사용자는 로그인에 성공한다. 이후 회원가입을 제외한 모든 기능을 이용할 수 있다. |
| Failed Post Condition | 사용자는 로그인에 실패한다. 이후 회원가입을 제외한 모든 기능을 이용할 수 없다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 Tripiary에 로그인할 때 실행된다. |
| 1 | 사용자는 로그인 페이지에서 아이디와 비밀번호 입력 후 로그인 버튼을 누른다. |
| 2 | Tripiary에 등록된 회원인지 데이터베이스에서 비교를 통해 확인한다. |
| 3 | 비교 후 등록된 회원이라면 로그인에 성공하고 메인 화면을 띄운다. |
| 4 | 이 Use case는 로그인이 성공하면 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 아이디 또는 비밀번호가 잘못되어 로그인에 실패하는 경우 a-1. 아이디 또는 비밀번호가 틀렸다는 메시지를 띄운다. |
| 2 | b. 등록된 회원이 아닌 경우 b-1. 등록된 회원이 아님을 메시지로 띄워준다. |
| 3 | c. 아이디나 비밀번호가 입력되지 않은 경우 c-1. 아이디 또는 비밀번호를 입력해달라는 메시지를 띄워준다. |
| RELATED INFORMATION | |
| Performance | <=5초 |
| Frequency | Variable |
| Concurrency | 제한 없음 |
| Due Date | |

| Use Case #2 : Register Manager | |
|--------------------------------|--|
| GENERAL CHARACTERISTICS | |
| Summary | 사용자가 시스템을 사용하기 위해 회원 등록할 때 사용하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사이트에 접속해 있고 통신이 가능해야 한다. |
| Trigger | 로그인 화면에서 회원가입 버튼을 눌러 회원가입을 하려 할 때 |
| Success Post Condition | 사용자는 회원가입에 성공한다. 이후 로그인 기능을 이용할 수 있다. |
| Failed Post Condition | 사용자는 회원가입에 실패한다. 이후 모든 기능을 이용할 수 없다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 'Tripiary'에 회원가입을 할 때 실행된다. |
| 1 | 사용자는 로그인 화면에서 회원가입 버튼을 누른다. |
| 2 | 시스템은 회원가입 화면을 띄운다. |
| 3 | 사용자는 정보를 입력하고 회원가입 버튼을 누른다. |
| 4 | 시스템은 회원가입의 성공을 판단한다. 성공했다면 로그인 화면을 띄운다. |
| 5 | 이 Use case는 회원가입이 성공하면 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 중복된 아이디가 존재하는 경우 a-1. 중복된 아이디가 존재한다는 메시지를 띄운다. |
| 2 | b. 비밀번호가 6자리 미만인 경우 b-1. 비밀번호는 6자리 이상으로 입력하라는 메시지를 띄운다. |
| 3 | c. 비어있는 입력정보가 있는 경우 c-1. 비어있는 정보를 입력하라는 메시지를 띄운다. |
| RELATED INFORMATION | |
| Performance | <=5초 |
| Frequency | 사용자당 1번 |
| Concurrency | 제한 없음 |
| Due Date | |

| Use Case #3 : Search public course | |
|------------------------------------|--|
| GENERAL CHARACTERISTICS | |
| Summary | 다른 사용자가 공개한 여행 코스를 검색하고 열람하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사용자가 로그인한 상태여야 한다. |
| Trigger | ‘공개 코스 검색’ 메뉴 선택했을 때 |
| Success Post Condition | 공개 코스 목록이 화면에 출력된다. |
| Failed Post Condition | 네트워크 오류 또는 결과 없음이 화면에 표시된다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 메인화면에서 ‘공개 코스 검색’ 버튼을 누를 때 시작된다. |
| 1 | 사용자가 메인화면에서 ‘공개 코스 검색’ 버튼을 누른다. |
| 2 | 검색 필터나 키워드를 입력한다. |
| 3 | 시스템이 공개 코스 목록을 불러온다. |
| 4 | 결과를 화면에 출력한다. |
| 5 | 이 Use case는 시스템이 검색된 코스 출력 후 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 네트워크 오류가 발생한 경우 a-1. 네트워크 오류 발생했다는 오류 메시지를 띄운다. |
| 2 | b. 검색 결과가 없는 경우 b-1. 검색 결과가 없을 경우 ‘일치하는 코스가 없습니다’ 메시지를 띄운다. |
| RELATED INFORMATION | |
| Performance | <=5초 |
| Frequency | Variable |
| Concurrency | 제한 없음 |
| Due Date | |

| Use Case #4 : Mark as favorite | |
|--------------------------------|-----------------------------|
| GENERAL CHARACTERISTICS | |
| Summary | 다른 사람의 코스를 즐겨찾기 목록에 추가하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |

| | |
|------------------------|--|
| Preconditions | 사용자가 다른 사용자의 공개 코스를 보고 있는 상태여야 한다. |
| Trigger | ‘즐거찾기’ 버튼을 클릭했을 때 |
| Success Post Condition | 코스가 즐거찾기 목록에 저장된다. |
| Failed Post Condition | 코스가 즐거찾기 목록에 저장되지 않는다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 공개 코스 조회 중 공개 코스 카드의 하트 아이콘을 클릭할 때 실행된다. |
| 1 | 사용자가 코스 조회 중 공개 코스 카드의 하트 아이콘을 클릭하여 즐거 찾기 추가/제거를 설정한다. |
| 2 | 서버에 즐거찾기 요청을 전송한다. |
| 3 | 성공 메시지 표시 및 UI 반영한다. |
| 5 | 화면 오른쪽 하단의 하트 버튼을 눌러 즐거찾기한 코스만 필터링하여 볼 수 있다. |
| 6 | 이 Use case는 즐거찾기 목록에 추가가 성공하면 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 로그인하지 않은 경우 경고창을 통해 알림을 표시한다. |
| RELATED INFORMATION | |
| Performance | <=3초 |
| Frequency | Variable |
| Concurrency | 제한 없음 |
| Due Date | |

| Use Case #5 : View my course | |
|------------------------------|--|
| GENERAL CHARACTERISTICS | |
| Summary | 사용자가 자신이 저장한 코스 목록을 열람하라는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사용자가 로그인되어 있어야 한다. |
| Trigger | 사용자가 메인화면에서 '내 코스 보기' 메뉴를 클릭했을 때 |
| Success Post Condition | 사용자 코스 목록이 표시된다. |
| Failed Post Condition | 코스 목록을 불러오지 못한다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 메인 화면에서 '내 코스 보기' 버튼을 눌렀을 때 실행된다. |
| 1 | 사용자가 메인 화면에서 '내 코스 보기' 버튼을 누른다. |
| 2 | 시스템이 해당 사용자의 ID에 연결된 코스를 조회한다. |
| 3 | 조회된 코스 목록이 출력된다. |
| 4 | 이 Use case는 사용자의 코스 목록 출력에 성공하면 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 서버가 지연되었을 경우 a-1. '로딩 중입니다.' 메시지를 화면에 띄운다. |
| 2 | b. 저장된 코스가 없을 경우 b-1. '저장된 코스가 없습니다.' 메시지를 화면에 띄운다. |
| RELATED INFORMATION | |
| Performance | <=5초 |
| Frequency | Variable |
| Concurrency | 제한 없음 |
| Due Date | |

| Use Case #6 : Create course | |
|-----------------------------|--|
| GENERAL CHARACTERISTICS | |
| Summary | 사용자가 새로운 여행 코스를 생성하고 제목, 날짜 등을 입력해 저장하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사용자가 로그인되어 있어야 한다. |
| Trigger | 사용자가 메인화면에서 '코스 생성' 메뉴를 클릭했을 때 |
| Success Post Condition | 새 코스가 생성되고, 코스 목록에 저장된다. |
| Failed Post Condition | 필수 입력값이 없거나 네트워크 오류 발생하여 코스가 생성되지 않는다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 메인 화면에서 '새 코스 만들기' 버튼을 눌렀을 때 실행된다. |
| 1 | 사용자가 메인 화면에서 '새 코스 만들기' 버튼을 누른다. |
| 2 | 코스 제목을 입력한다. |
| 3 | 사용자가 '저장' 버튼을 누른다. |
| 4 | 시스템이 입력값의 유효성을 검사한다. |
| 5 | 서버에 코스 데이터가 저장 후 코스가 생성되었다는 메시지를 출력한다. |
| 6 | 이 Use case는 사용자의 새로운 코스 생성에 성공하면 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 필수 입력값이 비어 있을 경우 a-1. 필수 입력값이 비어 있을 경우 '필수 항목을 입력하세요' 메시지를 화면에 출력한다. |
| 2 | b. 코스 저장 중 네트워크 오류가 발생한 경우 b-1. '저장 실패' 메시지를 표시하고 재시도 옵션을 제공한다. |
| RELATED INFORMATION | |
| Performance | <=3초 |
| Frequency | Variable |
| Concurrency | 제한 없음 |
| Due Date | |

| Use Case #7 : Edit course | |
|---------------------------|---------------------------------|
| GENERAL CHARACTERISTICS | |
| Summary | 사용자가 생성하여 저장된 코스를 수정하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사용자가 로그인되어 있고, 생성된 코스가 있어야 한다. |
| Trigger | 사용자가 코스 편집 화면에서 '수정' 버튼을 클릭했을 때 |
| Success Post Condition | 사용자가 수정한 내용이 저장된다. |
| Failed Post Condition | 수정한 내용이 저장되지 않는다. |

| MAIN SUCCESS SCENARIO | |
|-----------------------|--|
| Step | Action |
| S | 사용자가 편집 화면에서 '코스 수정' 버튼을 클릭했을 때 실행된다. |
| 1 | 사용자가 수정하고 싶은 코스를 선택 후 '코스 수정' 버튼을 클릭한다. |
| 2 | 사용자가 장소를 검색한다. |
| 3 | 검색된 장소들이 지도상에 마커로 표시된다. |
| 4 | 검색된 장소들 중 원하는 장소에 코스 추가 버튼을 선택한다. |
| 5 | 장소가 추가된 버튼은 추가완료로 표시되고 지도 상에 별 마커로 표시된다. |
| 6 | 변경 완료 후 '저장' 클릭 시 시스템이 서버에 수정 데이터를 전송한다. |
| 7 | 수정 완료 후 코스 목록 화면으로 이동한다. |
| 8 | 이 Use case는 사용자가 코스 수정 성공하면 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 필수 입력이 누락 되었을 경우 a-1. '필수 입력 사항이 누락 되었습니다.'라는 메시지를 화면에 출력한다. |
| 2 | b. 저장에 실패 할 경우 b-1. 저장이 실패되었다는 경고 메시지를 표시하고 재시도 옵션을 제공한다. |
| RELATED INFORMATION | |
| Performance | <=3초 |
| Frequency | Variable |
| Concurrency | 제한 없음 |
| Due Date | |

| Use Case #8 : Write memo | |
|--------------------------|--|
| GENERAL CHARACTERISTICS | |
| Summary | 사용자가 마커에 대한 메모를 작성하고 저장하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사용자가 코스를 생성하고 코스에 마커가 추가된 상태여야 한다. |
| Trigger | 사용자가 코스 편집 화면에서 마커 클릭 후 '메모 작성' 버튼을 누른다. |
| Success Post Condition | 메모가 해당 마커에 저장된다. |
| Failed Post Condition | 메모가 저장되지 않아 입력 내용이 유실 될 수 있다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 코스 편집 화면에서 마커 선택 후 '메모 작성' 버튼을 누를 경 |

| | |
|----------------------------|--|
| | 우 실행된다. |
| 1 | 사용자가 코스 편집 화면에서 마커 선택 후 '메모 작성' 버튼을 누른다. |
| 2 | 메모 작성란에 내용을 입력한다. |
| 3 | 사용자가 저장 버튼을 누르면 메모 내용이 서버에 전송된다. |
| 4 | '저장' 버튼을 누르면, 시스템이 정보를 검증한다. |
| 5 | 저장 성공 시 확인 메시지가 화면에 표시된다. |
| 6 | 이 Use case는 마커에 메모 작성 후 저장이 성공하면 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 사용자가 입력 없이 저장 시도하였을 경우 a-1. '내용을 입력해주세요' 메시지 화면에 출력한다. |
| 2 | b. 서버 응답 지연 또는 저장 실패하였을 경우 b-1. 저장이 실패되었다는 경고 메시지를 표시하고 재시도 옵션을 제공한다. |
| RELATED INFORMATION | |
| Performance | <=2초 |
| Frequency | variable |
| Concurrency | 제한 없음 |
| Due Date | |

| | |
|------------------------------------|---|
| Use Case #9 : Delete course | |
| GENERAL CHARACTERISTICS | |
| Summary | 사용자가 저장한 코스를 삭제하는 기능 |
| Scope | Tripiary |
| Level | User level |
| Author | |
| Last Update | |
| Status | Analysis |
| Primary Actor | User |
| Preconditions | 사용자가 로그인하고 코스를 생성한 상태여야 함. |
| Trigger | 사용자가 저장한 코스를 삭제하는 기능 |
| Success Post Condition | 코스가 완전히 삭제되고 목록에서 제거된다. |
| Failed Post Condition | 코스가 삭제되지 않는다. |
| MAIN SUCCESS SCENARIO | |
| Step | Action |
| S | 사용자가 나의 코스 목록에서 '삭제' 버튼 클릭할 경우 실행된다. |
| 1 | 사용자가 삭제하고 싶은 코스 옆에 있는 '삭제' 버튼을 누른다. |
| 2 | 시스템이 삭제 여부를 재확인한다. |
| 3 | 확인 후 삭제 요청 서버에 전송한다. |
| 4 | 삭제 성공 후 성공하였다는 메시지를 화면에 표시하고 코스 목록을 업데이트한다. |
| 5 | 이 Use case는 사용자가 선택한 코스 삭제 성공 시 끝난다. |
| EXTENSION SCENARIOS | |
| Step | Branching Action |
| 1 | a. 사용자가 취소 선택하였을 경우 |

| | |
|---------------------|---|
| | a-1. 삭제 옵션이 중단된다. |
| 2 | b. 네트워크 오류가 발생하여 삭제하는 것을 실패하였을 경우 b-1. 삭제 실패하였다는 에러 메시지를 화면에 출력한다. |
| RELATED INFORMATION | |
| Performance | <=2초 |
| Frequency | variable |
| Concurrency | 제한 없음 |
| Due Date | |

3. Domain analysis

1) Main_Tripiary

Tripiary 시스템의 중심 클래스이다. 전체 앱 흐름을 관리하며, 로그인 상태, 화면 전환, 주요 기능 호출 등을 조율하는 중심 클래스 역할을 한다.

2) Registration

사용자가 Tripiary에 가입하기 위한 기능을 담당한다. 사용자는 이메일, 비밀번호, 닉네임 등을 입력하여 계정을 생성하고, 이후 로그인 기능을 사용할 수 있게 된다.

3) Login

사용자 인증을 수행하는 클래스이다. 사용자가 입력한 이메일과 비밀번호가 기존 계정 정보와 일치하는지를 검증하며, 인증이 완료되면 사용자가 로그인되고 메인 화면으로 이동시킨다.

4) UserManagement

사용자 정보의 조회, 수정, 탈퇴를 관리하는 클래스이다. 회원가입된 사용자들의 기본 정보 리스트를 관리하며, 향후 관리자 페이지가 추가된다면 확장 가능한 구조이다.

5) CourseCreation

새 코스 생성 및 편집 기능을 담당한다. 코스 추가, 장소 검색, 마커 배치, 메모 저장 등을 담당한다.

6) MarkerManagement

지도 위에 사용자가 마커를 추가하거나 수정, 삭제하는 기능을 수행한다. 장소 이름, 위도/경도, 메모 등의 정보를 포함하며 코스의 주요 구성 요소로 작동한다.

7) MemoWriter

각 마커에 부착되는 메모 정보를 다루는 클래스이다. 사용자가 장소에 대한 느낌이나 정보를 짧은 글로 작성할 수 있게 한다.

8) CourseViewer

사용자가 저장한 여행 코스를 열람하고 수정, 삭제하는 기능을 제공하는 클래스이다. 코스를 선택하면 상세 정보를 표시하며, 해당 코스에 대한 편집 기능으로 연결된다.

9) CourseSearch

다른 사용자가 공유한 공개 코스를 검색하고 열람하는 기능을 담당하는 클래스이다. 키워드, 지역, 테마 등으로 검색하며 결과 리스트를 반환한다.

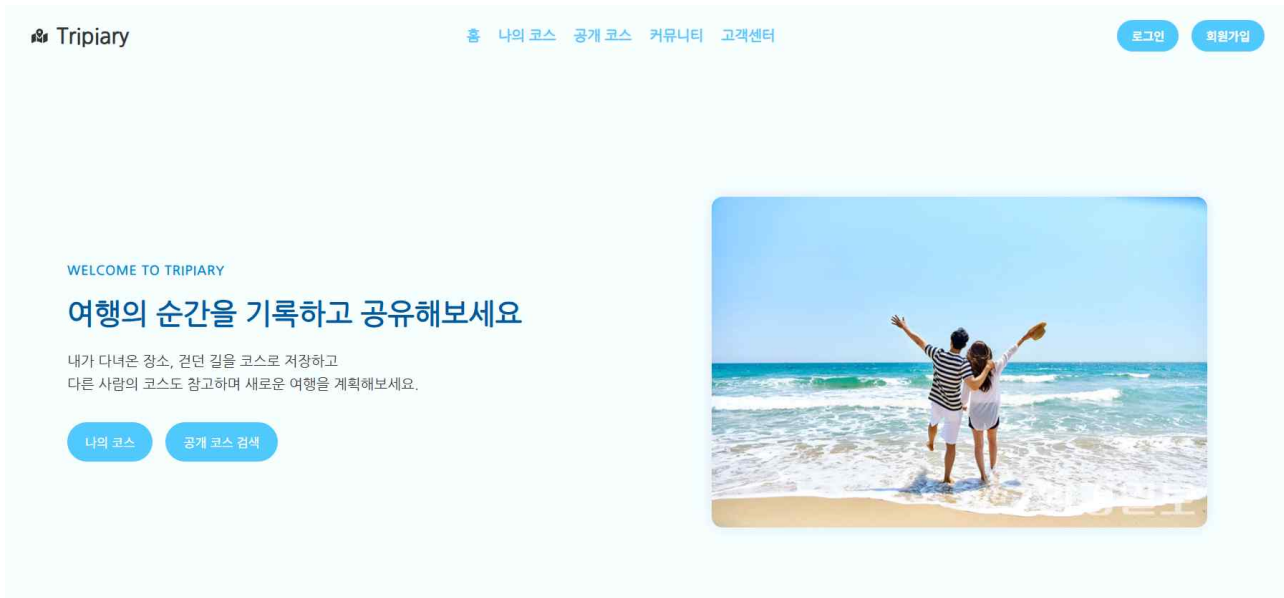
10) FavoriteManager

사용자가 공개 코스를 즐겨찾기에 등록하거나 해제하는 기능을 담당하는 클래스이다. 공개 코스 카드에 있는 하트 아이콘을 통해 해당 코스를 즐겨찾기 상태로 전환하며, 이 정보는 Firebase Firestore의 사용자 별 하위 컬렉션에 저장된다. 또한, 공개 코스 페이지 우측 하단의 고정 하트 버튼을 통해 사용자는 등록한 즐겨찾기 코스만 필터링하여 볼 수 있다.

4. User Interface prototype

1) 메인화면

사용자가 웹사이트에 접속하면 메인 화면이 출력된다. 사용자는 로그인 후 기능들을

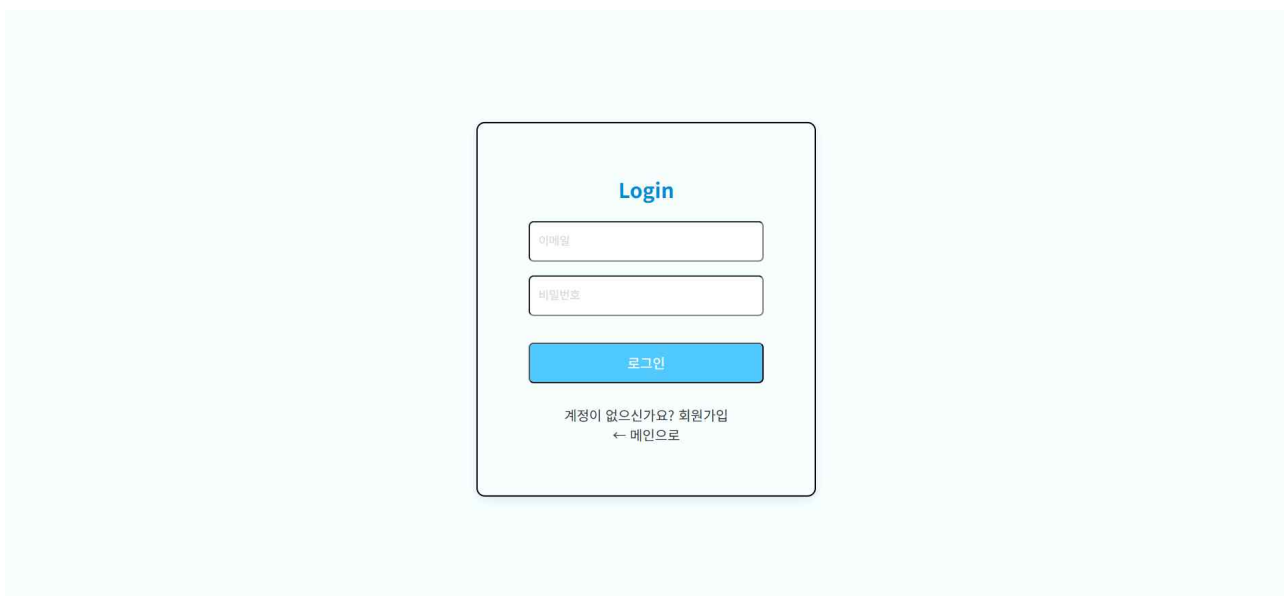


이용할 수 있다.

사용자는 메인 화면에서 나의 코스, 공개 코스, 커뮤니티 사용이 가능하다.

2) 로그인

사용자는 로그인 버튼을 선택하였을 경우 로그인 화면으로 넘어간다. 사용자가 회원가 입한 이메일과 비밀번호를 입력하면 서버에서 확인 후 로그인된다.



3) 회원가입

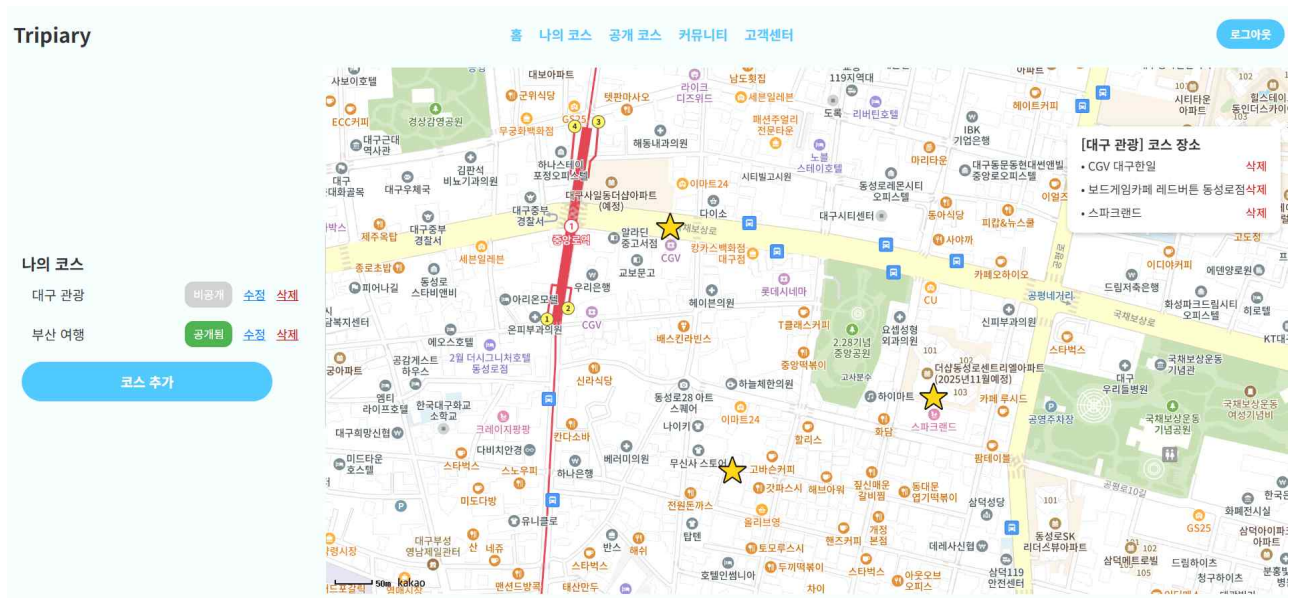
사용자가 회원가입 버튼을 선택하였을 경우 회원가입 페이지로 넘어간다. 사용자가 이메일을 작성하고 중복확인 버튼을 누른 후 인증이 된 경우 회원가입이 완료된다.

회원가입

이미 계정이 있으신가요? 로그인
← 메인으로

4) 나의 코스 페이지

사용자가 '나의 코스' 버튼을 선택한 경우 이 페이지로 넘어온다. 사용자는 코스 추가 버튼을 눌러 코스 이름을 적고 코스를 추가할 수 있다. 또한 코스 이름 옆에 삭제 버튼을 눌러 삭제도 가능하다.



공개코스 페이지 - 페이지에 접근했을 시 기본 페이지

Tripiary



공개 코스 페이지 - 목록 옆 수정 버튼을 눌렀을 경우

이 화면은 목록 옆 수정 버튼을 눌렀을 경우이다. 상단에 검색할 수 있는 검색창이 나오고 우측에는 검색결과 목록이 보여진다. 목록의 강 장소를 클릭했을 시 해당 장소를 지도에서 보여주고 옆에 추가 버튼을 선택할시 지도 위 마커가 별로 표시되고 해당 장소가 코스에 추가된다.

Tripiary

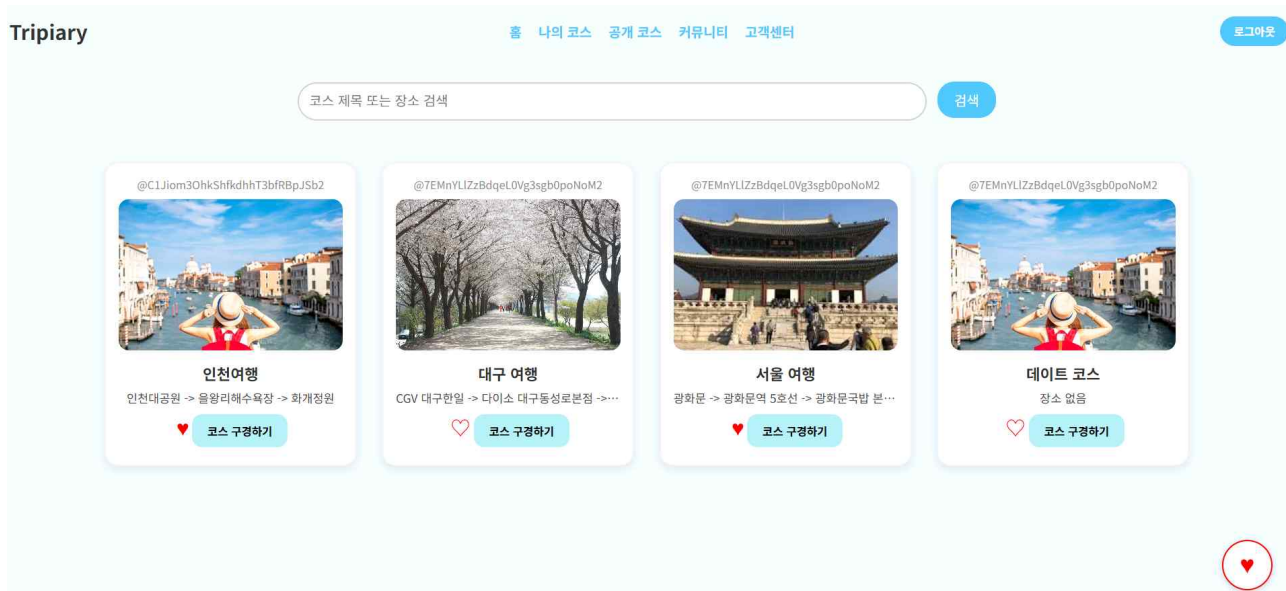


공개 코스 페이지 - 저장된 코스를 눌렀을 경우

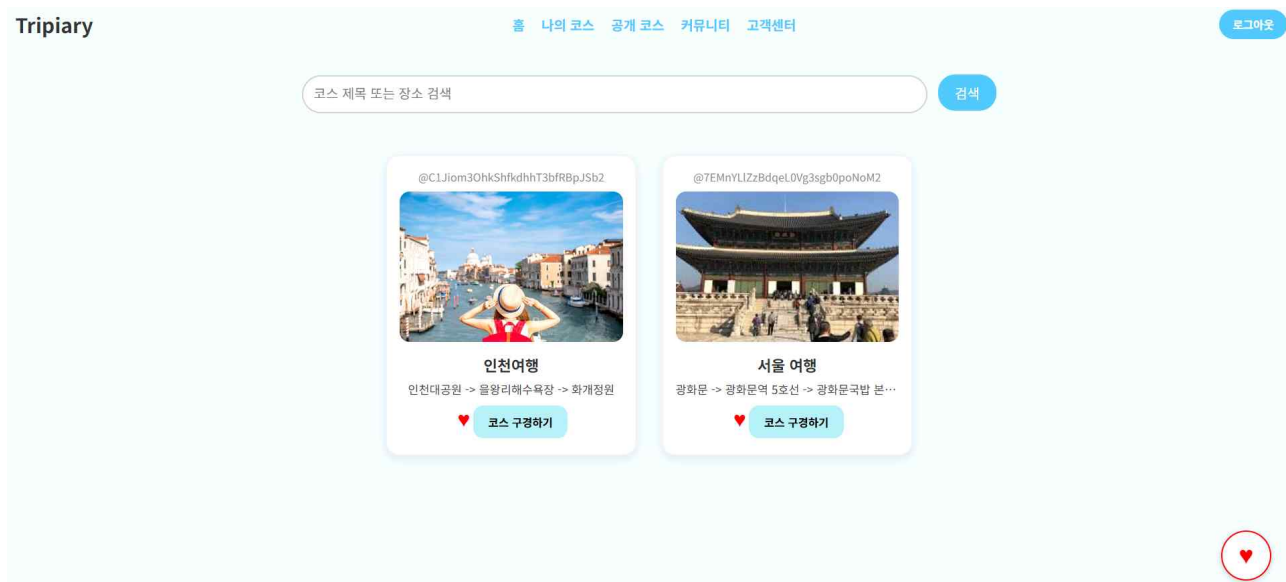
이 화면은 나의 코스 목록중 하나를 선택했을 때 나오는 화면이다. 지도 위에는 내가 추가한 장소들이 별 마커로 표시되고 우측에는 나의 코스에 어떤 장소들이 저장되어 있는지 보여준다. 목록에서 삭제 버튼을 눌러 장소 삭제도 가능하다.

5) 공개 코스 검색

사용자가 공개 코스 버튼을 눌렀을 경우 이동하는 화면이다. 상단에는 장소나 사용자 이름을 검색하여 찾을 수 있고 밑에는 추천 코스가 뜬다. 하트를 누르면 즐겨찾기 항목에 저장된다.



공개 코스 검색 - 공개 코스 페이지를 들어갔을 때의 화면



공개 코스 페이지 - 하단 하트 버튼을 눌렀을 경우 화면

5. Glossary

| terms | description |
|-----------|--|
| user | 이 웹사이트를 사용하는 사용자 |
| course | 사용자가 마커와 경로를 기반으로 구성한 여행 경로 |
| marker | 지도 위에 사용자가 직접 추가한 장소 표시 |
| memo | 마커에 작성하는 짧은 메모 |
| UI | 사용자가 보는 화면 |
| API | 다른 시스템의 기능을 내 프로그램에서 사용하게 해주는 인터페이스 |
| authentic | 진짜인 확인하는 것 |

6. References

<https://brunch.co.kr/@oksambari/91>

<https://koreascience.kr/article/CFKO202223650463394.view?orgId=anpor&hide=breadcrumb,journalinfo>

<https://gnaseel.tistory.com/22>