



# **LIFT ON**

BEST ASSISTANT FOR YOUR LIFT LIFE

21912067

김동현

[tmdrl583205@naver.com](mailto:tmdrl583205@naver.com)

- Analysis -

## [ Revision history ]

Revision date	Version #	Description	Author
2023/03/15	0.00	First Document	Kim Dong Hyun
2023/05/20	0.01	Modify Use Case Diagram	Kim Dong Hyun
2023/06/01	0.02	Modify Domain Analysis	Kim Dong Hyun

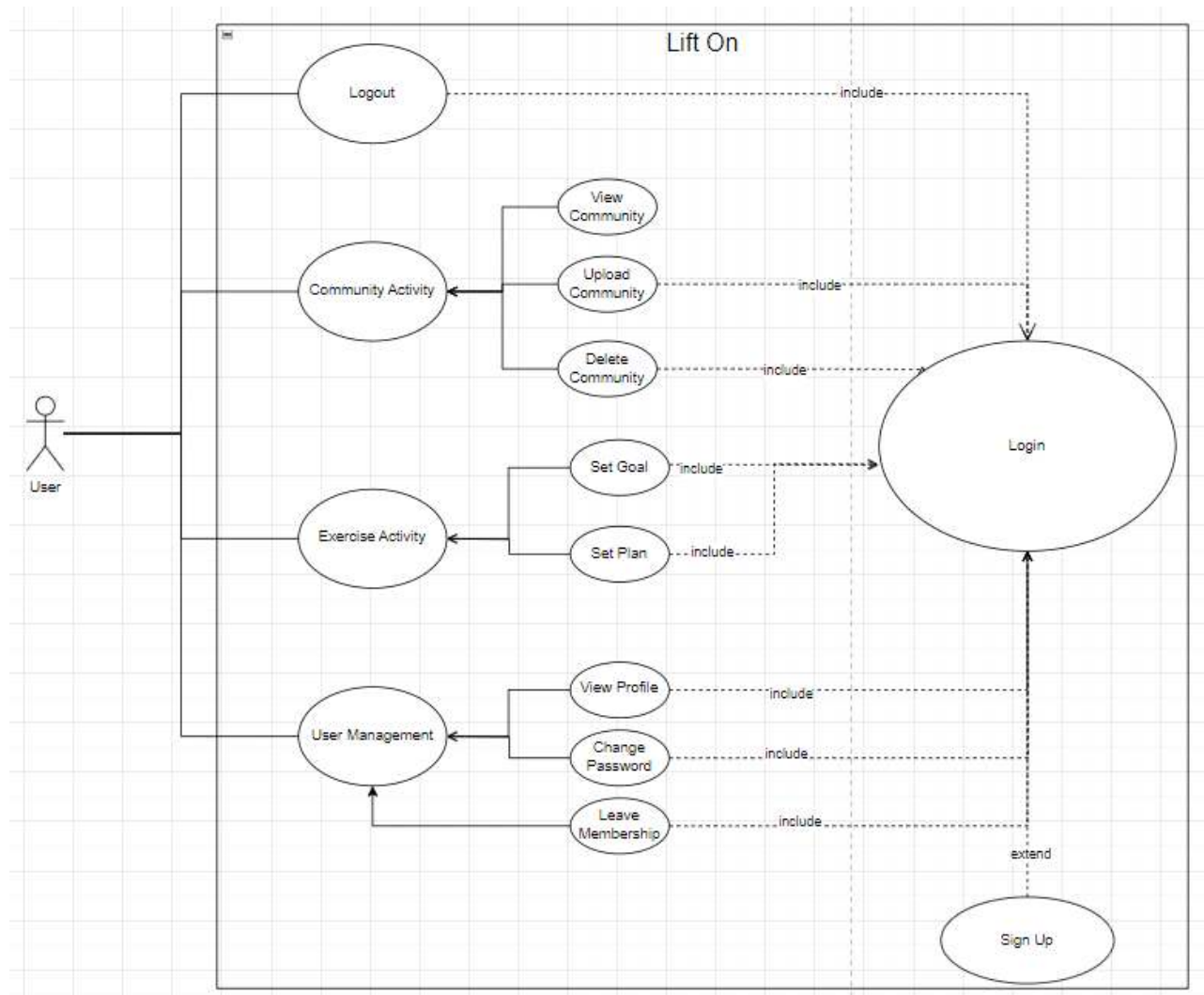
= Contents =

1. Introduction .....	4
2. Use case analysis .....	5
3. Domain analysis .....	18
4. User Interface prototype .....	19
5. Glossary .....	28
6. References .....	28

# 1. Introduction

- 본 프로젝트의 피트니스 애플리케이션 Lift On은 사용자가 자신의 피트니스 목표를 추적하고 개선할 수 있도록 지원하기 위한 애플리케이션이 될 것이다. 본 문서에서는 개념화 문서에서 다뤘던 시스템의 요구사항을 더 자세히 분석해, Use Case에서 시스템이 무엇을 하는지, 유저와 시스템이 어떻게 상호작용하는지를 깊게 알아볼 것이며, Domain Analysis에서 더 자세히 각 시스템, 클래스가 어떤 역할을 하는지, 어떻게 행동하게 되는지 간략히 설명할 것이다. 또한 User Interface prototype에서 대략적인 애플리케이션의 모습을 나타내고 설명할 것이다.
- Lift On의 핵심 기능은 유저의 유형을 일반 유저와 트레이너로 나누어 놓았으며 두 유저 유형의 공통적인 기능은 운동 목표 설정, 실시간 모니터링, 계획 설정, 커뮤니티 활동이며 이를 통해 트레이너는 자신 또는 자신의 상품 등을 홍보할 수 있고 타인에 운동에 대해 조언할 수 있다. 반대로 일반 유저도 커뮤니티에서 활동하며 운동 정보를 얻고, 성장할 수 있다.

## 2. Use case analysis



Use Case #1 : Sign Up	
General Characteristics	
Summary	사용자가 시스템에 원활히 활동하기 위해 회원 등록할 때 사용하는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	System이 원활하게 작동해야 한다.
Trigger	시작화면에서 회원가입을 누른다
Success Condition	Post 회원가입을 진행할 양식을 볼 수 있다.
Failed Condition	Post 회원가입을 진행할 양식을 볼 수 없다.
Main Success Scenario	
Step	Action
1	사용자가 시스템을 실행한다.
2	사용자가 시작 화면에서 회원가입 버튼을 누른다.
3	사용자가 회원가입 양식을 보고 작성한다.
4	사용자가 양식을 제출하면 시스템이 회원가입 성공 여부를 판단하고 로그인 화면을 띄운다.
5	사용자가 성공적으로 회원가입에 성공한 후 로그인할 수 있다.
Extension Scenario	
Step	Branching Action
3	사용자가 양식 작성 중 뒤로 가기 버튼을 누른 경우 : 모든 양식의 정보를 지운다고 알린 후 시작화면으로 넘어가게 한다.
3	사용자가 양식을 올바르게 채우지 않은 경우 : 사용자에게 양식을 다시 채우라는 메시지를 띄운다.
4	이미 중복된 사용자가 있는 경우 : 사용자에게 중복된 사용자가 있다는 메시지를 띄운다.
Related Information	
Performance	≤ 5초
Frequency	사용자 당 1번
Concurrency	None
Due Date	

Use Case #2 : Login	
General Characteristics	
Summary	사용자가 시스템의 기능을 사용하기 위해 자신의 정보를 이용해 인증받는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 회원가입이 되어있어야 한다.
Trigger	시작화면에서 로그인 버튼을 누른다.
Success Post Condition	사용자가 로그인에 성공하고 시스템의 다양한 기능들을 이용할 수 있다.
Failed Post Condition	사용자가 로그인에 실패하여 시스템의 기능들을 이용하지 못한다.
Main Success Scenario	
Step	Action
1	사용자가 시스템을 실행한다.
2	사용자가 시작화면에서 로그인 버튼을 누른다.
3	사용자가 아이디와 패스워드를 입력한다.
4	사용자가 시스템에 로그인 요청을 한다.
5	사용자가 성공적으로 로그인한 후 기능을 사용할 수 있다.
Extension Scenario	
Step	Branching Action
3	사용자가 옳지 않은 아이디 혹은 패스워드를 입력한 경우 : 사용자에게 아이디 또는 패스워드가 옳지 않고 재입력하라고 메시지를 띄운다. (이 경우는 아이디, 패스워드를 입력하지 않은 경우도 포함한다.)
4	시스템에 로그인 요청을 했을 때 에러가 발생하는 경우 : 사용자를 에러 페이지로 유도하고 재로그인 또는 시스템 자체를 재실행하도록 만든다.
Related Information	
Performance	≤ 3초
Frequency	사용자 당 하루 평균 1~3번
Concurrency	None
Due Date	

Use Case #3 : Logout	
General Characteristics	
Summary	사용자가 다른 계정으로 로그인하는 등의 이유로 로그아웃하는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인이 되어있어야 한다.
Trigger	로그아웃 버튼을 누른다.
Success Post Condition	사용자가 로그아웃에 성공하여 다른 목적을 수행할 수 있다.
Failed Post Condition	사용자가 로그아웃에 실패한다.
Main Success Scenario	
Step	Action
1	사용자가 시스템을 실행한다.
2	사용자가 로그인한 상태이다.
3	사용자가 로그아웃 버튼을 누른다.
4	사용자가 시스템에 로그아웃 요청을 한다.
5	사용자가 성공적으로 로그아웃한 후 다른 목적을 수행한다.
Extension Scenario	
Step	Branching Action
4	시스템 문제로 로그아웃에 실패하는 경우 : 사용자를 에러 페이지로 유도하고 재로그인 또는 시스템 자체를 재실행하도록 만든다.
Related Information	
Performance	≤ 3초
Frequency	사용자 당 하루 평균 1~3번
Concurrency	None
Due Date	



Use Case #4 : View Community	
General Characteristics	
Summary	사용자가 커뮤니티에 들어와 사용자들의 게시물을 보는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 시스템을 실행해야 한다.
Trigger	사용자가 시작화면에서 비회원으로 접속한다.
Success Condition	Post 사용자가 커뮤니티 게시글과 댓글을 볼 수 있다.
Failed Condition	Post 사용자가 커뮤니티 게시글과 댓글을 볼 수 없다.
Main Success Scenario	
Step	Action
1	사용자가 시스템을 실행한다.
2	사용자가 시작화면에서 비회원 접속 버튼을 누르거나 로그인 후 커뮤니티에 접속한다.
3	사용자가 게시글과 댓글을 열람한다.
Extension Scenario	
Step	Branching Action
2	시스템 문제로 비회원 접속에 실패하는 경우 : 사용자를 에러 페이지로 유도해 다시 시작화면으로 넘어가게 한다.
3	시스템 문제로 게시글 열람에 실패하는 경우 : 서버 문제로 게시글을 사용자에게 보낼 수 없는 문제가 발생하면 에러 페이지로 유도하거나 사용자에게 에러 메시지를 띄워 재접속하도록 만든다.
Related Information	
Performance	≤ 5초
Frequency	매우 빈번함
Concurrency	None
Due Date	

Use Case #5 : Upload Community	
General Characteristics	
Summary	사용자가 커뮤니티에 들어와 자신이 작성하고 싶은 글을 업로드하는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인되어 있어야 한다.
Trigger	사용자가 게시물 작성 버튼을 클릭한다.
Success Post Condition	서버에 사용자가 등록한 게시물 또는 댓글이 저장된다.
Failed Post Condition	서버에 사용자가 등록한 게시물 또는 댓글이 저장되지 않게 된다.
Main Success Scenario	
Step	Action
1	사용자가 시스템을 실행한다.
2	사용자가 시작화면에서 로그인한다.
3	사용자가 커뮤니티에 접속해 글을 작성한다.
4	서버에 사용자가 작성한 글이 저장된다.
Extension Scenario	
Step	Branching Action
3	<ul style="list-style-type: none"> <li>- 사용자가 자신의 글의 카테고리를 설정하고 싶은 경우 : 사용자가 글을 작성할 때 어떠한 유형의 글을 작성할지 카테고리로 선택할 수 있게 한다.</li> <li>- 사용자가 사진을 업로드하고 싶은 경우 : 파일 선택 버튼을 만들어 사용자가 원활히 이미지를 업로드할 수 있도록 만든다.</li> </ul>
4	<ul style="list-style-type: none"> <li>- 사용자의 유형이 트레이너인 경우 : 사용자의 유형이 트레이너라면 글을 다른 색깔로 표시하거나 특정한 마크를 붙여 일반 사용자가 알 수 있도록 한다.</li> </ul>
Related Information	
Performance	≤ 5초
Frequency	매우 빈번함
Concurrency	None
Due Date	

Use Case #6: Delete Community	
General Characteristics	
Summary	사용자가 자신이 작성한 글을 삭제하는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인되어 있어야 한다.
Trigger	사용자가 게시물의 삭제 버튼을 클릭한다.
Sucess Post	사용자가 글을 삭제할 때 서버에 삭제 요청이 받아들여지고
Condition	데이터 베이스에서 글이 삭제된다.
Failed Post	사용자가 글을 삭제해도 데이터 베이스에 글이 그대로 남게
Condition	된다.
Main Success Scenario	
Step	Action
1	사용자가 시작화면에서 로그인한다.
2	사용자가 커뮤니티에 접속해 자신의 글을 선택하고 삭제한다.
3	서버에 사용자가 삭제한 글이 데이터 베이스에서도 삭제된다.
Extension Scenario	
Step	Branching Action
2	- 사용자가 버튼을 잘못 눌렀을 수도 있으므로 사용자에게 경고 메시지를 띄운다.
3	- 삭제된 글을 다른 사용자가 열람할 경우 : 삭제한 글이 커뮤니티에 표시될 때 제목과 본문을 완전히 지우고 "삭제된 글입니다" 라고 띄우도록 한다. - 글이 삭제되었을 때 댓글의 경우 : 삭제된 글에 달려있는 댓글들도 같이 삭제된다.
Related Information	
Performance	≤ 5초
Frequency	매우 빈번함
Concurrency	None
Due Date	

Use Case #7 : Set Goal	
General Characteristics	
Summary	사용자가 자신의 운동 관련 목표를 세우는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인되어 있어야 한다.
Trigger	사용자가 운동 목표를 세우고 저장한다.
Success Post Condition	사용자가 자신의 목표를 세우고 저장하면 데이터 베이스에 그대로 저장된다.
Failed Post Condition	사용자가 목표를 세우고 저장해도 데이터 베이스에는 아무런 변동이 없다
Main Success Scenario	
Step	Action
1	사용자가 시작화면에서 로그인한다.
2	사용자가 운동 탭을 누르고 자신의 목표를 설정한다.
3	설정한 목표를 저장하고 그 데이터는 데이터 베이스에 저장 된다.
Extension Scenario	
Step	Branching Action
2	- 운동 관련 목표일 것이기에 사용자의 목표 몸무게, 목표 중량 등을 자세히 선택할 수 있도록 양식을 제공한다.
3	- 사용자가 그 전의 목표를 지우기를 원할 경우 : 그 전에 세웠던 목표는 기본적으로 지워지지 않고 사용자의 선택에 의해 지워질 수 있도록 한다. - 사용자가 자신의 목표를 공유하기를 원할 경우 : 저장된 목표를 사용자가 원한다면 커뮤니티에 공유할 수 있도록 만든다.
Related Information	
Performance	≤ 5초
Frequency	n달에 한 번
Concurrency	None
Due Date	

Use Case #8 : Set Plan	
General Characteristics	
Summary	사용자가 자신의 운동 계획을 세우는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인되어 있어야 한다.
Trigger	사용자가 운동 계획을 세우고 저장한다.
Success Post Condition	사용자가 자신의 계획을 세우고 저장하면 데이터 베이스에 그대로 저장된다.
Failed Post Condition	사용자가 계획을 세우고 저장해도 데이터 베이스에는 아무런 변동이 없다
Main Success Scenario	
Step	Action
1	사용자가 시작화면에서 로그인한다.
2	사용자가 계획을 세우고 저장한다.
3	계획은 데이터 베이스에 저장되고 사용자는 목표에 맞는 계획을 세울 수 있다.
Extension Scenario	
Step	Branching Action
2	- 사용자가 편하게 계획을 세우고 싶은 경우 : 사용자가 원 활히 계획을 세울 수 있도록 다양한 운동 종류를 제공한다.
3	- 사용자가 자신이 세운 계획을 공유하고 조언받고 싶은 경 우 : 운동 계획과 완료한 운동을 커뮤니티에 공유할 수 있도 록 한다.
Related Information	
Performance	≤ 5초
Frequency	사용자당 평균 하루 한 번
Concurrency	None
Due Date	

Use Case #9 : View Profile	
General Characteristics	
Summary	사용자가 자신의 개인 정보를 열람할 수 있는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인되어 있어야 한다.
Trigger	사용자가 사용자 관리로 접속한다.
Success Condition	Post 사용자가 자신의 개인 정보를 원활히 열람할 수 있다.
Failed Condition	Post 사용자가 서버로부터 자신의 개인 정보를 열람할 수 없다.
Main Success Scenario	
Step	Action
1	사용자가 시작화면에서 로그인한다.
2	사용자가 서버에 요청해 자신의 개인 정보를 받는다.
3	사용자가 개인 정보를 열람한다.
Extension Scenario	
Step	Branching Action
2	- 시스템 오류로 사용자가 원활히 개인 정보를 받을 수 없는 경우 : 에러 페이지로 유도 후 재접속 또는 재로그인하도록 만든다.
Related Information	
Performance	≤ 5초
Frequency	알 수 없음
Concurrency	None
Due Date	

Use Case #10 : Change Password	
General Characteristics	
Summary	사용자가 자신의 비밀번호를 수정하는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인되어 있어야 한다.
Trigger	사용자가 자신의 비밀번호 수정하기를 원하고 사용자 관리로 접속한다.
Sucess Condition	Post 사용자가 자신의 비밀번호를 수정하면 서버가 이를 저장한다.
Failed Condition	Post 사용자가 자신의 비밀번호를 수정하지 못한다.
Main Success Scenario	
Step	Action
1	사용자가 시작화면에서 로그인한다.
2	사용자가 정보 관리로 접속해 자신의 개인 정보를 서버로부터 받는다.
3	사용자가 양식에 맞게 자신의 비밀번호를 수정한다.
4	수정된 비밀번호가 서버에 저장된다.
Related Information	
Performance	≤ 5초
Frequency	알 수 없음
Concurrency	None
Due Date	

Use Case #12 : Leave Membership	
General Characteristics	
Summary	사용자가 회원 탈퇴하는 기능
Scope	Lift On
Level	User Level
Author	
Last Update	
Status	Analysis
Primary Actor	User
Preconditions	사용자가 로그인되어 있어야 한다.
Trigger	사용자가 회원 탈퇴를 원하고 사용자 관리로 접속한다.
Success Post Condition	사용자가 회원 탈퇴를 하면 서버가 유저의 정보를 모두 지운다.
Failed Post Condition	사용자가 회원 탈퇴에 실패한다.
Main Success Scenario	
Step	Action
1	사용자가 시작화면에서 로그인한다.
2	사용자가 정보 관리로 접속한다.
3	사용자가 확인용 개인 정보를 입력하고 회원 탈퇴한다.
Related Information	
Performance	≤ 5초
Frequency	알 수 없음
Concurrency	None
Due Date	



### 3. Domain analysis

- Post : 게시글에 관한 클래스로, 사용자의 게시글을 요청해 컴포넌트에 전달한다.
- Auth : 사용자의 로그인, 로그아웃, 회원가입 등을 처리하는 클래스이다.
- Comment : 댓글에 관한 클래스로, 사용자의 댓글을 요청해 서버로부터 받아 컴포넌트로 전달한다.
- Goal : 사용자의 운동 목표에 관한 클래스로 서버에 저장된 사용자의 운동 목표를 컴포넌트로 전달한다.
- Plan : 사용자가 짜놓은 계획을 받고 관리하는 클래스이다.
- User : 사용자에 관한 클래스이다. 유저의 타입, 프로필, 각종 정보 등 사용자에 관한 모든 것을 가지는 클래스이다.
- Setting : 사용자가 서버에 유저 관련 정보, 개인 정보에 관한 정보를 요청할 때 이 요청을 처리해주는 클래스이다.
- Exercise : 사용자가 서버에 자신의 운동 관련 정보를 요청할 때 이 요청을 처리하고 데이터를 가져다주는 클래스이다.
- Community : 사용자가 서버에 커뮤니티 관련 정보, 게시물, 댓글과 같은 데이터를 요청하면 요청을 처리하고 가져다주는 클래스이다.

#### 4. User Interface prototype



- 어플리케이션의 시작화면이다. 사용자는 로그인, 회원가입, 비회원 사용을 고를 수 있다.

## Login

- 로그인 화면으로, 사용자가 아이디(이메일)과 비밀번호를 입력해 서버에 로그인할 수 있다.

# Sign Up

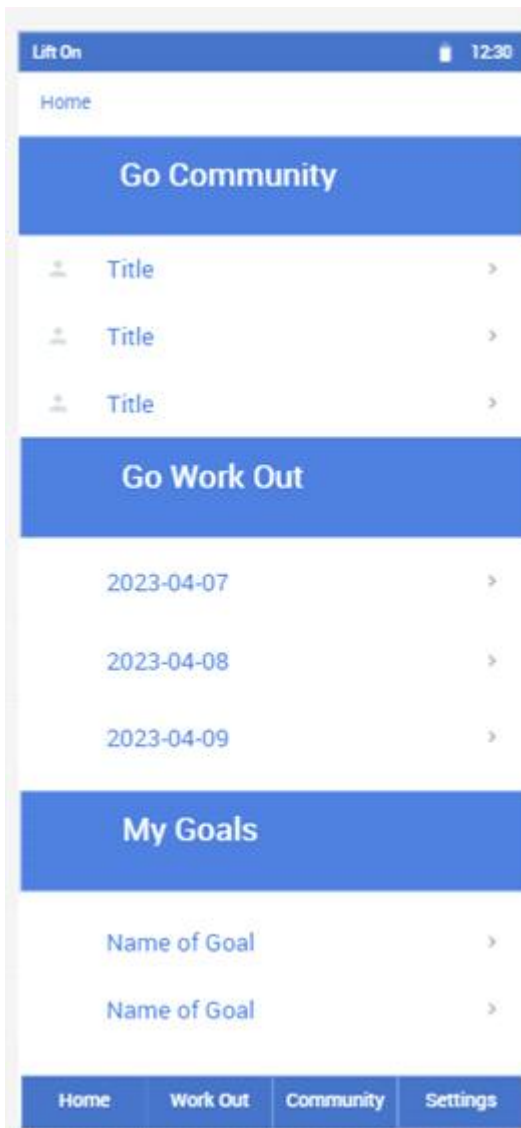
example@example.com

password

confirm

Go

- 회원 가입되지 않은 사용자가 회원 가입을 진행한다. 이메일과 비밀번호를 입력하고 비밀번호 확인을 진행한다. 옳지 않은 이메일 양식, 동일하지 않은 비밀번호를 입력 시 경고창을 상단에 띄운다.



- 홈 화면으로, 커뮤니티의 게시물들, 사용자가 계획한 운동들, 사용자가 세운 목표를 한 눈에 볼 수 있으며 각 컴포넌트를 클릭 시 해당 컴포넌트로 바로 이동할 수 있다.



- 사용자가 운동했던, 운동을 진행할 캘린더를 살펴볼 수 있으며 어느 날에 운동을 했는지, 운동을 하지 않았는지를 살펴볼 수 있다. 날짜를 누를 시 운동을 이미 진행했으며 과거의 날짜라면 운동 기록을, 운동을 진행하지 않았거나 오늘 또는 미래의 날짜라면 운동 계획을 띄운다.
- 또한 운동 목표 컴포넌트를 가지고 있으며, 컴포넌트 버튼 클릭 시 목표 탭으로 넘어간다.



- 사용자가 진행할 운동 계획이 있다. 운동 시작 전에는 오른쪽 하단의 버튼이 Start로 되어있으며, 운동 시작 시 상단의 타이머가 작동되며, 원하는 운동을 클릭해 진행할 수 있다. 또한 왼쪽 하단의 버튼을 통해 원하는 운동을 추가할 수 있다. 운동을 끝내고 싶다면 Start에서 Done으로 바뀐 오른쪽 하단의 버튼을 클릭해 끝낸다. 또한 운동 기록은 서버에 저장되어 사용자가 원할 때 서버에 요청해 열람할 수 있다.

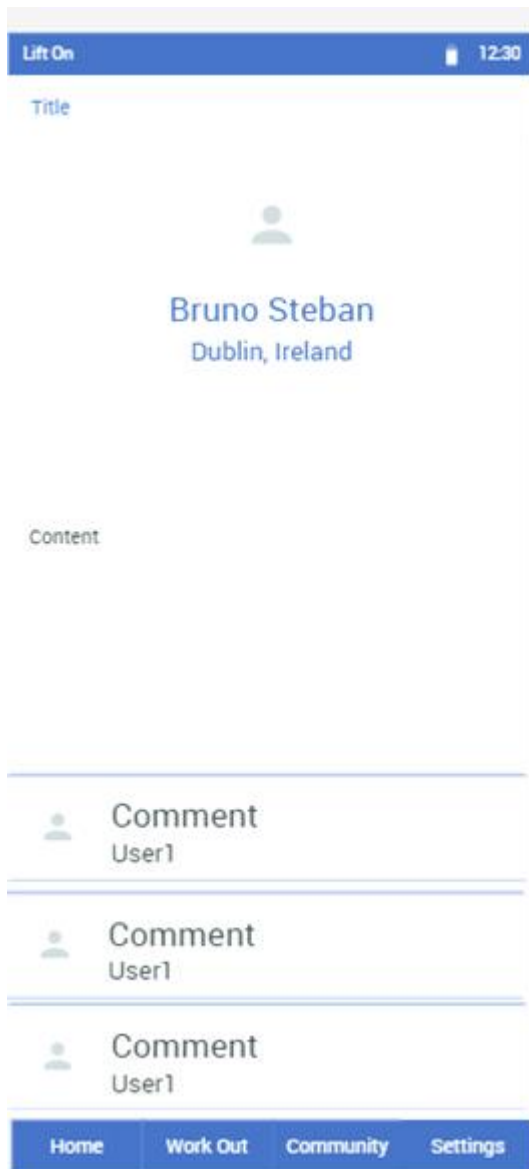
The screenshot shows a mobile application interface for setting fitness goals. At the top, a blue header bar contains the text "Lift On" and a battery status icon with the time "12:30". Below the header, the section is titled "Goal". It contains four input fields, each with a label to its left: "Weight", "BMI", "Fat Mass", and "Muscle Mass". Below these fields is a section titled "Track Your Body" which contains four placeholder images for before and after photos. At the bottom of this section is a blue button labeled "Save". The very bottom of the screen features a navigation bar with four tabs: "Home", "Work Out", "Community", and "Settings".

- 사용자가 운동 목표를 세울 수 있다. 여러 가지 목표를 세울 수 있는데, 목표 몸무게, 목표 BMI, 목표 체지방량, 목표 골격근량을 정할 수 있다. 또한 아래 사진들을 통해 자신의 몸을 before & after 방식으로 체크할 수 있다. 하단의 Save 버튼을 누르면 서버로 데이터들이 보내지고 저장된다. 사용자는 서버의 데이터를 언제나 열람할 수 있다.

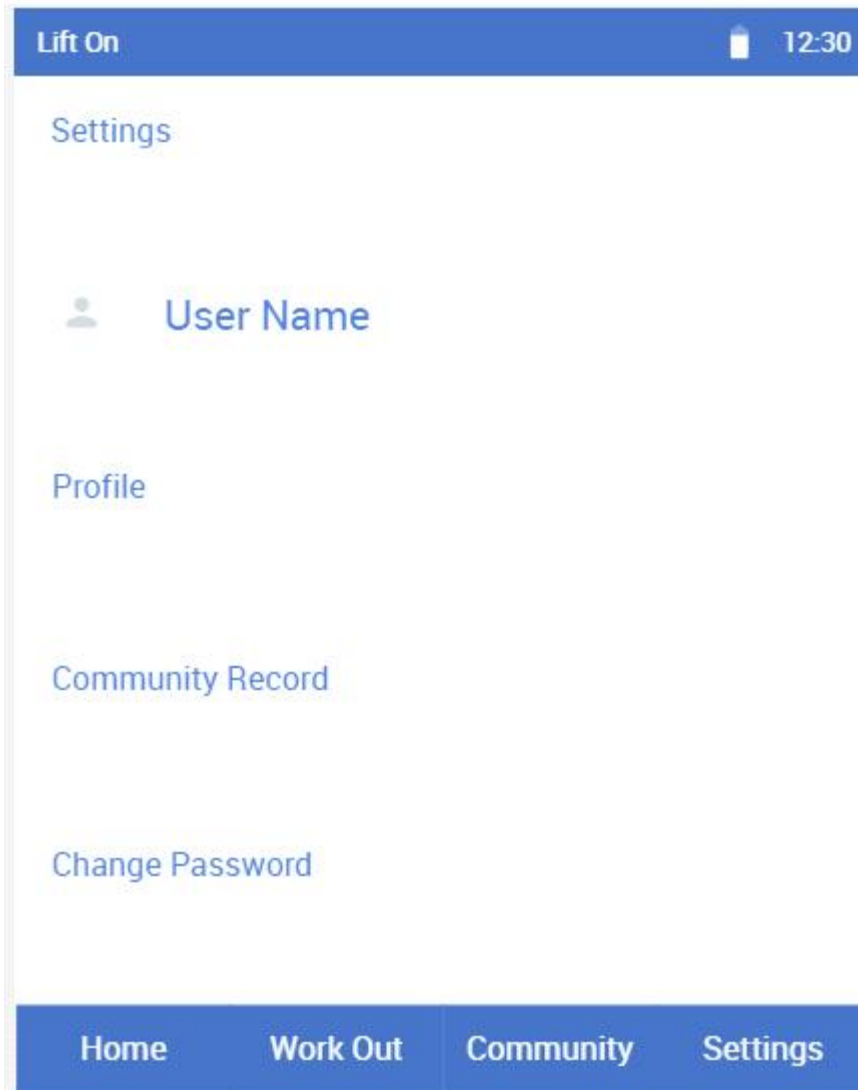


- 커뮤니티에 입장해 사용자들의 글을 열람할 수 있다. 이 화면에서는 사용자의 로그인 필요하지 않다. 각 컴포넌트를 클릭하면 사용자의 글로 이동해 글의 내용, 댓글 등을 자세히 살펴볼 수 있다.





- 앞서 말했듯 사용자의 글을 자세히 볼 수 있다. 글의 주인이 내용을 수정, 삭제를 진행하고 다른 사용자가 댓글을 달 수도 있다. 이 화면에서 트레이너가 댓글로 조언을 남기거나, 글 자체를 쓸 수 있다. 트레이너인지 일반 사용자인지는 글 내용 상단의 프로필을 통해 확인할 수 있다.



- 설정 화면으로, 사용자가 자신의 프로필을 수정하거나, 비밀번호를 바꾸는 등의 개인 정보 수정, 커뮤니티에서 활동한 기록을 열람하고 관리하는 등의 다양한 작업을 수행할 수 있다.

## 5. Glossary

- **유저** : 실질적인 앱의 사용자로, 운동을 즐기고 배우기 위한 일반 사용자와 운동을 가르치고 조언하는 트레이너로 나누어져 있다.
- **계정 상태** : 현재 유저가 로그인했는지, 로그아웃했는지 등의 상태
- **커뮤니티** : 유저들이 커뮤니티 공간에서 다양한 운동 정보를 주고받으며 서로 성장을 도모할 수 있도록 한다
- **서버** : 사용자의 요청을 처리하고 사용자가 원하는 모든 정보를 처리한다.

## 6. References

- Use Case Diagram : <https://narup.tistory.com/70>
- User Interface : <https://www.fluidui.com>