

캡스톤디자인 설계서

IT응용공학과 송재훈

V1.2

목차

[1. 개정 이력 2](#_Toc117727529)

[2. 개요 2](#_Toc117727530)

[3. 색상 팔레트 2](#_Toc117727531)

[4. iOS App 3](#_Toc117727532)

[5. 애플워치 App 13](#_Toc117727533)

[6. 용어사전 13](#_Toc117727534)

# 개정 이력

|  |  |  |
| --- | --- | --- |
| 버전 | 날짜 | 사유 |
| 1.0 | 2022.09.15 | 최초 작성 |
| 1.1 | 2022.09.27 | iOS App View 작성,  색상 팔레트 추가 |
| 1.2 | 2022.10.26 | iOS App View 최신화 |
|  |  |  |

# 개요

1. 본 문서의 글꼴은 맑은 고딕(제목, 본문)을 사용하고, 글꼴의 크기는 혼동되지 않도록 하여 문서의 가독성을 높일 수 있도록 한다.
2. 개발 내용에 대해 상세히 기록하고, 변동이 있을 시 수정하여 최산화한다.
3. 용어 사전을 통해 혼동되거나, 쉽게 접하지 못하는 단어를 설명한다.
4. 뜻이 같은 단어들은 최대한 한 단어로 기입하도록 한다.(예 : Face, 페이스, 바탕화면 -> Face)

# 색상 팔레트

|  |  |  |  |
| --- | --- | --- | --- |
| 1, 0, 0 | 209, 47, 47 | 54, 25, 28 | 220, 72, 128 |
| 27, 29, 31 | 120, 221, 67 | 37, 55, 29 | 224, 254, 83 |
| 255, 255, 255 | 89, 194, 220 | 30, 51, 58 | 117, 251, 177 |
| 247, 206, 70 | 159, 165, 166 |  |  |

# iOS App

## ContentView

|  |  |
| --- | --- |
| 화면 명 | ContentView |
| 화면 이미지 | |
| <MainView>    <SidemenuView>    <Coupon 호출>    <Coupon 입력 및 등록> | |
| <Coupon 입력 결과>    <포인트 지급>    <포인트 최대 지급>    <레벨 업>    <액티비티 링 진척도 표현> | |
| 화면 설명 | |
| <ContentView>  앱을 처음 실행했을 때 노출되는 화면이다. <MainView>가 기본 값으로 호출되며 <ContentView>안에 Tapbar 기능을 통해 <AchivementView>, <StorageView>를 구분하고 버튼을 누를 시 버튼에 해당하는 View로 이동한다.  <MainView>  현재 레벨, 액티비티 링 진척도, 사이드 메뉴 버튼(햄버거 버튼), 현재 선택된 펫, 현재 보유중인 포인트, 레벨 업을 위한 포인트 게이지와 남은 포인트, 포인트 지급 버튼, 애플워치 공유 버튼(미구현), 탭 뷰가 존재한다.  포인트 지급 버튼으로 현재 보유중인 포인트를 차감해서 포인트 지급을 할 수 있고, 만약 레벨 업에 필요한 포인트를 모두 지급하면 레벨 업을 위한 버튼으로 변경된다. 또한 다시 한 번 포인트를 소모해서 레벨 업을 할 수 있고, 레벨 업을 통해 현재 소유하고 있지 않은 펫 중 랜덤으로 한 마리를 해금한다.  <SidemenuView>  사이드 메뉴 버튼을 통해 <SidemenuView>를 불러온다. 사이드 메뉴에는 사용자 ID(미구현), 쿠폰, 도움말(미구현), 개발자 연락(미구현)의 기능을 할 수 있게 하고, 사용자 ID는 사용자를 식별하기 위한 값, 쿠폰은 이벤트 등으로 지급되는 쿠폰을 입력할 수 있게 한다. 현재는 디버깅을 위한 코드를 입력하는 용도로 데이터의 상호작용을 위해 사용한다. 사이드 메뉴를 벗어나기 위해선 사이드 메뉴 버튼을 다시 한 번 누르거나 화면을 슬라이드하여 벗어날 수 있다.  디버깅 코드 목록 : [givexp10, givexp100, givexp1000, clearxp, kcal100, kcal1000, exer10, exer100, stand1, stand10, clearact, clearstorage]  도움말은 App의 첫 시작 시 사용자를 위한 도움말을 제공하고 이후에는 나타나지 않으며 도움말 버튼을 클릭 시 나타나도록 구현, 개발자 연락은 개발자 이메일 등 문제, 버그 발생 시 연락하기 위한 용도이다. | |

## StorageView

|  |  |
| --- | --- |
| 화면명 | StorageView |
| 화면이미지 | |
| <StorageView>    <펫 해금>    <펫 선택 변경>    <펫 선택 변경 후 MainView>    <모든 펫 해금> | |
| 화면 설명 | |
| <StorageView>  현재 소유 중인 펫과 미해금된 펫에 대한 정보를 나타내는 View이다. 현재 소유 중인 펫의 수와 존재하는 총 펫의 수를 나타낸다. 해금되지 않은 펫은 미해금 여부를 식별할 수 있도록 블러 처리와 함께 상호작용을 막고, 해금된 펫은 클릭을 통해 <MainView>에 노출될 펫을 선택할 수 있다. 또한 현재 선택한 펫과 그렇지 않은 펫을 구별하기 위해 펫 이름의 색을 변경한다. 선택 : 흰색, 미선택 : 회색  현 View에서는 <StorageView> 내 스크롤 기능을 보여주기 위한 쓰레기 프레임이 추가 되었다. | |

## AchivementView

|  |  |
| --- | --- |
| 화면명 | AchivementView |
| 화면이미지 | |
| <AchivementView>    <업적 달성 및 수령>    <보상 수령> | |
| 화면 설명 | |
| <AchivementView>  업적 목록을 현재 업적 달성 수와 총 업적 달성 수, 달성하기 위한 조건, 달성도를 나타내고 업적의 진행도에 따라 보상 수령, 진행 중, 수령 완료를 나타내 준다.  업적을 달성하면 보상을 수령할 수 있고, 보상 수령을 통해 포인트를 지급한다. | |

## 펫 이미지

|  |  |
| --- | --- |
| 벡터그래픽이(가) 표시된 사진  자동 생성된 설명1 | 벡터그래픽이(가) 표시된 사진  자동 생성된 설명2 |
| 벡터그래픽이(가) 표시된 사진  자동 생성된 설명3 | 벡터그래픽이(가) 표시된 사진  자동 생성된 설명4 |
| 벡터그래픽이(가) 표시된 사진  자동 생성된 설명5 |  |

# 애플워치 App

## 개요

# 용어사전

|  |  |  |
| --- | --- | --- |
| 1. | 애플워치 페이스, Face | 애플워치를 켰을 때 나오는 바탕화면으로 휴대폰, 컴퓨터의 바탕화면과 같음  예 : 시계이(가) 표시된 사진  자동 생성된 설명  애플 기본제공 Face 리스트  <https://support.apple.com/ko-kr/guide/watch/apde9218b440/watchos> |
| 2. | 컴플리케이션, Complication | 애플워치에서 스마트폰의 위젯과 같은 개념으로 1. 예 이미지의 좌우 상단의 작은 아이콘이다.  아이콘을 통해 정보를 제공할 수 있고, 클릭시 해당 앱, 기능으로 바로가기 이동을 할 수 있다. |
| 3. | AoD, Always on Display | 기기가 잠금화면 일 때 화면이 완전히 꺼지지 않고 필요한 정보를 제공하는 상태이며 애플워치에서는 5세대부터 이 기능을 제공한다.  예 : 아이폰 13에 AOD(Always On Display)가 드뎌! |
| 4. | 액티비티 링 | 애플에서 기본으로 제공하는 기능으로 R : 움직이기 링, G : 운동하기 링, B : 일어서기 링으로 3가지가 있으며 당일 할당량에 따라 링이 점점 채워진다. 사용자가 수치를 직접 설정할 수 있다.  움직이기 링 : 활동 소모 칼로리를 기록  운동하기 링 : 운동에 준하는 활동을 몇 분이나 했는지 표시  일어서기 링 : 그날의 일어서기 목표를 얼마만큼 도달했는지 확인하며, 온종일 주기적으로 일어나도록 유도  예 :  애플 액티비티 링 소개  <https://www.apple.com/kr/watch/close-your-rings/> |
| 5. | View | SwiftUI에서 App의 UI의 일부를 표현하며, 이러한 View들을 설정하는데 사용하는 modifiers 메서드를 제공한다.  쉽게 말해 화면을 표현하는 단위이다,  “A type that represents part of your app’s user interface and provides modifiers that you use to configure views.” - <https://developer.apple.com/documentation/swiftui/view> |