

# deep;user

## Cigarette - 버튼 / 실시간 계산 / 프로그레스바 / 한도초과

좌측 상단의 Today 수를 2회 연속 클릭하면 수가 증가하고, 3회 연속 클릭하면 수가 감소합니다. 이 증감량에 맞춰 우측의 개비 수와 주기적으로 변경되는 '이렇게 살다간...' 하단의 정보가 수정됩니다. 만약 정해진 일일 최대치의 수량을 넘어서면 횡수 퍼센트가 초기화 됩니다.

하단 프로그레스바의 경우 한 분기의 목표 흡연량을 준수한 기간을 나타냅니다. 페르소나는 이를 일 흡연 7개비 / 30일로 설정해두었다고 가정하였고, 현재 19일 준수한 상태입니다.



b\_InfoPanel.pde

```
int dayOfWell = 19;
```

'b\_InfoPanel.pde'에서의 위 코드 부분을 찾아 수를 변경할 수 있습니다.(1~10)

## Issue and Today Task

좌측의 이슈 위젯은 농장 별 확인되지 않은 이슈의 개수를 농장 우측에 표시합니다.

우측의 당일 일정 위젯은 현재 시간을 기반으로 우선순위를 설정하여 최대 3개의 일정을 띄웁니다. 아직 진행되지 않은 가장 최근 일정을 우선으로, 만약 진행되지 않은 일정이 3개 미만일 경우, 진행된 일정 중 남는 개수만큼 가장 최근 일정부터 투명도를 적용하여 표시합니다. 해당 일정은 현재 날짜, 시간을 반영합니다.



e\_CalendarPanel.pde

```
// 일정 데이터 초기화
String[][] initializeScheduleData() {
}
```

위 데이터의 변경을 원하는 일자의 데이터를 변경하여 테스트해 볼 수 있습니다.

+기말 발표 현장에서 수정한 사항은 다음과 같습니다.

{"19:00 일상 - @나영 LP바", "11:30 미팅 - 운영 보고서 검토", "08:30 점검 - 개포CT 환  
풍기 교체", "15:30 납품 - emart 천안점"}, // 19일

// 위 19일자 일정에서

1. "11:30 미팅 - 운영 보고서 검토"를 지움 - 점검이 뜨는지 확인(연하게) - 위젯 크기와  
가독성을 고려해 최대 3개의 일정을 우선순위로 띄우게 하였음. 우선 1순위로 앞으로 다  
가오는 일정을 시간 순으로 띄움. 남은 일정이 3개가 안될 때부터 지난 일정에 투명도를  
추가해서 지난 일정이라는 것을 한 눈에 알아볼 수 있게 띄워줌.
2. 위 미팅을 다시 넣고 시간을 "07:30"으로 변경 - 지난 일정이 여러 개면 가장 최근의 것
3. "19:00 일상 - @나영 LP바", "11:30 미팅 - 운영 보고서 검토" 2개를 지움 - 두 개면  
두 개 띄움

## 모니터링 - 점포별 / 농장 상태 / 회전 / 프로그레스바 / 카테고 리버튼

먼저 최상단에 해당 농장의 위치를 진한 글씨로 표시하고 다음에 나올 농장을 오른쪽에 연하  
게, 이전의 농장을 왼쪽에 연하게 표시합니다. (스와이프 됩니다)

위쪽의 농장 멀티 세팅을 위한 위젯은 각 농장별로 저장되며, 해당 버튼을 클릭 시 표준 → 최  
적 → 절약 순으로 사이클을 돌릴수 있습니다.

아래쪽의 농장 코스트 그래프 위젯은 한달 기준으로 얼마나 소비하였는지 3가지 다른 자원  
과 기타 소비량을 마지막에 붙여서 총 4가지 소비를 표현합니다. 그래프 하단에는 목표량 대  
비 현재 소비량(색상O)과 현재 날짜 기준으로 얼마나 썼어야 했는지를 표현(흰색)을 바 형식  
으로 표현하였습니다. 만약, 현재 소비량이 최대치를 넘길 경우, 붉은색을 띄는 바가 오른쪽  
에서부터 역으로 진행됩니다.



d\_MonitoringAndCostPanel.pde

```
int costThreshold = 2700; // -> 1500 // 붉은 색 확인
```

## Calendar - 날짜 / 색 순서대로 / 하단 위젯 위치 / 5개 이상 일 경우

각 카테고리별로 특정 색상을 지정하여 해당 카테고리로 저장한 일정들은 해당 색상으로 달력에 표시됩니다. 해당 위젯은 각 날짜를 클릭해 세부 정보를 보거나 다시 클릭하여 전체 일정을 간단히 봅니다. 이때 일정의 개수가 4개를 넘어서 5개 이상이 된다면 해당 일정 갯수에 따라 위젯 자체의 크기를 키우도록 구현하였습니다. 이에 따라 하단 납품일정 위젯의 위치도 유동적으로 조정되게 하였습니다.



e\_CalendarPanel.pde

```
// 일정 데이터 초기화
String[][] initializeScheduleData() {
}
```

해당 부분의 데이터 부분을 수정하여 다양한 결과를 확인해 보실 수 있습니다.

## Delivery

오늘을 기준으로 다음달의 오늘 즉, 12월 19일 기준으로 1월 19일까지의 납품 일정을 불러옵니다. 해당 위젯은 일정 수에 따라 유동적으로 높낮이가 조절되며, 제일 아래 위치한 박스에 하단 곡면을 주어 디자인을 통일합니다.



f\_Deliverymanagementtable.pde

```
String[][] data = {
}
```

해당 부분의 데이터 부분을 수정하여 다양한 결과를 확인해 보실 수 있습니다.