**4 - 시스템 구성도 ( 아키텍처 ) 대본**

저희가 만드는 손 세정제 관리 도구의 시스템 구성도에 대해 설명드리겠습니다. 우선, 디바이스에서 Load Cell 을 통해서 손 세정제의 현재 무게를 측정해 ESP-01 모듈을 이용해 서버로 이 데이터를 전송합니다. 이 데이터는 서버에서 관리자가 보기 편한 형태로 가공되어 있습니다.

손 세정제를 불특정 사용자가 사용할 경우 무게의 변동이 있을 것이고, 그 무게를 측정해 지속적으로 서버로 전송합니다. GCP 서버는 전달받은 데이터를 DB에 저장된 손세정제 잔여 상황 정보와 비교하고, Web Page를 통해 잔여 상황과 그 정보에 따라서 손 세정제 잔량에 대해 부족 이벤트가 발생한 경우 해당 위치의 손 세정제 잔량이 부족함을 Web View 에서 알림으로 표기합니다.

**5 – 리소스 구조**

리소스 구조를 나타내기 위해 클래스 다이어그램을 사용했습니다. APP, HS\_status, Warning 클래스가 손 소독제 관리를 위해 사용되는 세 가지 클래스입니다.

APP 클래스와 HS\_status 클래스는 1대1 양방향 연관관계, HS\_status 클래스와 Warning의 클래스는 Aggregation ( 집합 연관 )관계입니다. HS\_status 클래스에는 warning 클래스를 로드하는 메소드와 데이터 요청 메소드, Show() Method와 손 소독제의 현재 잔량과 꽉 찬 상태일 때의 정보를 가진 Attribute 들을 포함하고 있고, APP 클래스에서는 HS\_status 클래스를 로드하는 메소드를 포함합니다. APP 에서 HS\_status를 로드하고, HS\_status에서는 Warning을 로드해 warn\_level 정보와 받아온 현재 잔량 데이터가 warn\_level이면 warn() 메소드를 실행해 경고 알림을 실행합니다. 아키텍처와 비교해 설명드리자면 show() 메소드는 잔여 상황 표기를, warn() 메소드는 잔여 상황 정보에 따른 이벤트 표기에, HS\_Cur\_amount 속성은 손세정제 잔여 상황 정보에 해당한다고 볼 수 있습니다.

Method들은 Public으로, Attribute들은 Private으로 선언해서 특정 손소독제에 대해 특정 정보가 1대1로 연결되는 관계를 갖게 됩니다.

**6 – 대표 서비스 프로시져**

대표 서비스 프로시져를 나타내기 위해 시퀀스 다이어그램을 사용했습니다. 인스턴스는 App, HS\_status, Warning의 3개의 클래스와 DB 서비스에 해당합니다.

서비스 프로시져를 단계별로 설명드리겠습니다.

우선, App 인스턴스에서 load\_HS\_status() 메소드로 call Message를 보내고, Opt에서 True/False에 관계 없이 DB의 작업 후 서비스 단계의 마지막에 HS\_status 인스턴스에서 show() 메소드를 return 받습니다.

App 인스턴스에서 load\_HS\_status() 메소드를 이용해 서비스 단계를 시작하고, HS\_status 인스턴스는 손 소독제의 잔량과 총량을 파라미터로 request\_data를 실행해 DB 서비스에 전달하고, send\_data() 메소드를 이용해 데이터를 return 받습니다. Return 받은 데이터를 이용해 서비스 단계를 진행하되, If에 해당한다고 볼 수 있는 Options 으로 잔량이 위험 수치에 이르러 부족하다고 판단될 경우 load\_warning() 를 실행해 Warning 인스턴스에 call Message를 보내고, Warning 인스턴스는 warn() 를 App에 return Message로 보냅니다. 앞서 말씀드린 show()의 실행은 Option과 관련 없이 실행되는 메소드로, 만약 If에서 True 반응이 나오는 잔여량이 부족한 경우에는 현재 잔량과 warn() 알림이 App 인스턴스로 return 되고, False 반응이 나와 잔여량이 충분한 경우에는 잔량만을 볼 수 있습니다.