**화장품 사용 주기 스케줄링 앱 기획서**

**0. 앱 개발 배경**

개발자(주혜진)는 기존에 아모레퍼시픽의 ‘아모레몰 앱’ 내 **스킨노트** 기능을 활용해 왔습니다. 해당 기능은 사용자의 얼굴을 촬영하여 모공, 주름, 홍조, 수분 등 피부 상태를 분석하고, 사용자가 선택한 화장품 사용 여부를 체크하며, 피부 컨디션을 기록하도록 유도합니다.

그러나 이 기능은 사용 단계가 많고 번거로우며, 무엇보다 **화장품 사용 주기를 관리하는 기능의 부재**라는 한계가 있습니다. 또한, 해당 기능이 쇼핑몰 앱 내 부가 기능으로만 제공된다는 점에서 사용성과 독립성이 부족하다고 느꼈습니다.

따라서 본 앱은 **화장품 사용 주기 관리**를 핵심으로 삼고, 사용자 편의성을 고려하여 **필수 입력을 최소화한 독립 앱**으로 개발하고자 기획되었습니다.

**1. 앱 주제**

**화장품 사용 주기 스케줄링 앱**  
사용자가 등록한 화장품의 사용 주기에 따라 오늘 써야 할 화장품을 보여주고, 사용 여부를 체크하며, 지정한 시간에 푸시 알림을 받을 수 있는 앱.

**2. 핵심 타겟 (User 정의)**

* **기초화장품을 여러 종류 사용해서 주기 관리가 필요한 사용자**
* **피부가 예민하지만 새로운 화장품 사용을 즐기고, 그에 따라 적응 주기를 세심하게 관리하고 싶은 사용자**
* **계획적인 스킨케어 루틴을 통해 피부 상태를 관리하고 싶은 사용자**

**3. 핵심 기능 정의 (MVP)**

(※ 세부 내용은 아래 "요구사항 정의서"에 포함되어 있음)

**[전체 기능 상세 정리]**

1. **화장품 등록 기능**
   * 브랜드, 제품명, 주요 성분, 기능, 카테고리를 입력하여 등록할 수 있음
   * 주요 성분 / 기능 / 카테고리는 기본 목록을 제공. (기타 포함)
   * 카테고리는 토너, 세럼 등 화장품 분류에 해당
2. **전체 화장품 목록 보기**
   * 등록된 모든 화장품 리스트 확인 가능
   * 각 화장품의 사용 횟수 및 간단 평가 요약 제공
   * 잘쓴템 / 무난템 / 비추템으로 자동 분류하여 시각적으로 구분
3. **화장품 사용 체크 기능**
   * 오늘 사용해야 할 화장품 목록을 우선 제시
   * 목록 그대로 사용했다면 원클릭으로 전체 사용 완료 처리 가능
   * 다른 화장품을 사용했다면 전체 목록 중에서 개별 선택 가능
4. **대시보드 기능**
   * 오늘 사용할 화장품 목록 표시
   * 감성 요소로 “피부 관리를 시작한 지 n일이 되었습니다” 문구 제공
   * 사용자에게 지속적인 앱 사용을 유도하는 목적
5. **푸시 알림 기능**
   * 사용자가 지정한 시간에 오늘 사용할 화장품 목록을 간단한 알림 형태로 제공
   * 예: 오후 8시, 세안 전 알림을 통해 사용 유도
6. **화장품 사용 평가 기능**
   * 필수 아님 (선택적 기능)
   * 간단 평가: 좋아요 / 무난해요 / 싫어요 중 택 1 (화장품별)
   * 상세 평가: 100자 이내 텍스트 메모, 오늘 전체 사용 후 감상 기록
7. **화장품 추천 기능 (회고 기반)**
   * “1년 전 오늘” 잘쓴템 리마인드 기능 제공
   * 계절별 회고 기능으로 “작년 봄에 잘 썼던 템”을 가을에 다시 보여주는 형태도 고려
8. **광고 기능 (선택 사항)**
   * 대시보드 하단에 신제품 광고 배너 삽입 가능
   * 타깃 사용자의 특성(신제품에 민감함)을 고려한 전략적 배치
9. **화장품 이미지 등록 (부가기능)**
   * 제품 이미지 등록 가능 (중요도 낮음, 최후 구현 예정)

**[1순위 – 반드시 구현할 기능]**

* 화장품 등록: 이름, 사용 주기 설정
* 오늘 사용할 화장품 목록 보기 (대시보드 형태)
* 사용 완료 체크 기능
* 사용자 지정 시간 푸시 알림 (Flutter Local Notifications 사용 예정)

**[2순위 – 여유 있을 경우 추가 구현]**

* 사용 후 평가 (좋아요 / 무난해요 / 싫어요 선택 + 100자 이내 메모) ※ 선택 입력
* 1년 전 오늘의 ‘잘쓴템’ 리마인드 알림
* 좋아요 기준 상위 화장품 모아보기 (잘쓴템/비추템/무난템)
* 화장품별 사용 횟수 및 평가 요약 보기

※ 광고 기능, 이미지 등록, 화장품 카테고리 분류 등은 배포 후 업데이트 시 반영 예정

**3-1. 기능 우선순위 전략**

본 프로젝트는 1순위 기능을 중심으로 개발하며,  
2순위 기능은 개발 기간에 따라 단계적으로 추가 예정.

**4. 경쟁 앱 간단 분석 (시장 조사)**- 대부분 스킨케어/다이어리 앱은 ‘기록’에만 초점을 둠  
- 주기에 따라 **자동알림**을 해주는 건 적음  
- 이 앱은 **주기 알림**을 중심 기능으로 삼아 차별화함

**5. 앱의 차별성**

* 단순 기록이 아닌, \*\*‘주기 기반 알림’\*\*을 핵심으로 한 실질적 사용 유도
* 장기 사용 시, **‘1년 전 오늘’ 리마인드**를 통한 감성 요소 제공
* \*\*‘좋아요 누적 기반 제품 추천’\*\*을 통한 개인화 기능 내장

**6. 예상 기술 스택**

* 개발 환경: Flutter (Dart), Android 대상
* 알림 기능: flutter\_local\_notifications 패키지
* 데이터 저장: **로컬 DB (SQLite + shared\_preferences)** 기반. 회원가입 없이 단말에 직접 저장하며, 서버 연동은 구현하지 않음.
* 디자인 도구: Figma 또는 손그림 와이어프레임

**7. 로그인 및 데이터 저장 방식**

본 앱은 **회원가입 없이, 사용자 단말 내에 데이터를 저장**하는 구조로 개발됩니다.

* 별도의 로그인 없이 실행 즉시 사용 가능하며, 사용자 경험의 간소화를 지향합니다.
* 화장품 등록 정보, 주기 설정, 평가 등 모든 데이터는 SQLite 및 shared\_preferences를 통해 로컬에 저장됩니다.
* 앱 삭제 시 데이터는 초기화되며, 추후 클라우드 연동 기능은 제공되지 않습니다.

이러한 구조는 서버 관리 없이 단기간 내 개발 완료를 목표로 하는 본 프로젝트의 목적에 부합하며, 개인정보 수집 이슈 없이 앱을 운영할 수 있는 장점이 있습니다.

* 개발 환경: Flutter (Dart), Android 대상
* 알림 기능: flutter\_local\_notifications 패키지
* 데이터 저장: 로컬 DB
* 디자인 도구: Figma 또는 손그림 와이어프레임

**8. 사용자 흐름 User Flow Diagram**앱 실행 → 메인 화면(오늘 써야 할 화장품 목록) → 화장품 추가 → 주기 설정 → 완료 체크

**9. Use Case Diagram (10주차)**텍스트, 도표, 그림, 스케치이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**10. 요구사항 정의서**

**1. 개요**

* 본 문서는 "화장품 사용 주기 스케줄링 앱"의 기능 및 시스템 동작에 대한 요구사항을 정의한다.
* 사용자는 다양한 화장품을 등록하고, 주기에 따라 사용을 관리하며, 푸시 알림과 사용 기록 기능을 통해 피부 루틴을 체계화할 수 있다.

**2. 기능 요구사항**



 **2.1 화장품 등록**

* 사용자는 브랜드명, 제품명, 주요 성분, 기능, 카테고리를 등록할 수 있다.
* 주요 성분, 기능, 카테고리는 기본 목록에서 선택하거나 사용자 정의 입력이 가능하다.

**2.2 전체 화장품 목록 보기**

* 사용자 등록 화장품 리스트를 전체적으로 확인 가능
* 각 항목에 대해 사용 횟수, 평가 결과, 분류(잘쓴템/무난템/비추템)를 표시

**2.3 화장품 사용 체크**

* 오늘 사용 예정인 화장품 목록을 제공
* 해당 목록 그대로 사용할 경우 원클릭 등록 가능
* 대체 제품 사용 시 전체 목록에서 수동 선택 가능

**2.4 대시보드**

* 오늘 사용할 화장품 목록을 표시
* “피부 관리를 시작한 지 N일” 문구 제공
* 푸시 알림 접근, 평가 접근 등의 진입 지점 역할 수행

**2.5 푸시 알림**

* 사용자가 설정한 시간에 당일 사용해야 할 화장품 목록을 알림으로 제공

**2.6 사용 후 평가 (선택)**

* 화장품별로 좋아요 / 무난해요 / 싫어요 중 하나 선택
* 상세 메모(100자 이내)를 통해 전체 사용 후 경험 기록 가능

**2.7 회고 기반 추천**

* “1년 전 오늘의 잘쓴템”, “작년 봄의 추천템” 등의 데이터를 기반으로 추천 제공

**2.8 광고 기능 (선택)**

* 대시보드에 신제품 광고 배너 삽입 기능 제공 (타겟 유저 성향 기반)

**2.9 화장품 이미지 등록 (후순위)**

* 사용자가 제품 이미지를 등록 가능

**3. 비기능 요구사항**

* 앱은 Flutter 기반으로 Android 환경에서 동작
* 사용자 데이터는 로컬 DB 또는 Firebase에 저장
* 푸시 알림은 flutter\_local\_notifications 패키지를 통해 처리
* 성능 저하 없이 30개 이상의 화장품을 등록하고 주기별로 처리 가능해야 함

**4. 우선순위 및 구현 전략**

| **기능** | **우선순위** | **구현 여부** |
| --- | --- | --- |
| 화장품 등록, 목록 보기, 사용 체크, 알림 | 상 | 초기 MVP에서 반드시 구현 |
| 평가, 회고 기반 추천 | 중 | 시간 여유 있을 시 구현 |
| 광고, 이미지 등록 | 하 | 추후 업데이트 또는 옵션 기능 |

* 개발 환경: Flutter (Dart), Android 대상
* 알림 기능: flutter\_local\_notifications 패키지
* 데이터 저장: 로컬 DB 또는 Firebase (선택)
* 디자인 도구: Figma 또는 손그림 와이어프레임

**11. 사이트맵**텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**12. 데이터베이스 구조**: 설정값은 SharedPreferences에, 사용자 데이터나 기록은 Drift에 저장하는 구조  
텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다. **1) AppSettings (앱 전역 설정) – SharedPreferences에 저장**: 회원가입 없이 앱을 사용하는 사용자가 설정하는 공통값 저장(푸시 알림, 시작일 등)

* skinCareStartDate (TEXT)
  + - 사용자가 지정한 ‘피부관리 시작일’ (ISO 8601 형식)
* pushTime (TEXT)
  + - 푸시 알림 시각 ("09:00" 형식)
* pushEnabled (BOOLEAN)
  + - 푸시 알림 사용 여부 (0=OFF, 1=ON)

**2) Cosmetic (화장품)**

: 사용자가 등록한 화장품 기본 정보 저장

* cosmeticId (INTEGER, PRIMARY KEY AUTOINCREMENT)
* brand (TEXT, NOT NULL)
* name (TEXT, NOT NULL)
* category (TEXT, NOT NULL)
* cycleId (INTEGER, NOT NULL)
  + FK → Cycle(cycleId)
* lastUsedDate (DATE)
  + 최종 사용 날짜 (ISO 8601). 오늘 사용 체크 시 업데이트됨.
* UsedCount (INTEGER, NOT NULL DEFAULT 0)
  + 누적 사용 횟수 (체크 시마다 증가)
* createdAt (TEXT)
* updatedAt (TEXT)

**3) Cycle (사용주기)**: 화장품별 사용 주기(반복, 커스텀) 세부 설정 저장

* cycleId (INTEGER, PRIMARY KEY AUTOINCREMENT)
* type (TEXT NOT NULL)
  + enum {"REPEAT", "CUSTOM", "RCMD"}
* repeatInterval (INTEGER)
  + "n일마다" 반복 주기.
  + type='REPEAT'일 때 필수
* customDays (TEXT)
  + type='CUSTOM'일 때 필수
  + JSON 문자열 형태로 커스텀 주기용 사용 요일 및 시간대 매핑 저장 (예: { "MON": [1,0], "TUE": [0,1], ... })
  + 요일별 [낮, 밤] 형식으로 표현: [1,0]=낮에 사용, [0,1]=밤에 사용, [0,0]=사용 안 함
* timeOfDay (TEXT NOT NULL)
  + enum {"DAY", "NIGHT", "ALL"}
  + DAY=아침, NIGHT=저녁, ALL=아침&저녁
  + type='REPEAT'일 때 필수
* createdAt (TEXT)
* updatedAt (TEXT)

설명: type 필드에 따라 주기 계산 로직이 달라집니다.

1. REPEAT: repeatInterval과 timeOfDay를 사용해 UsagePlan을 생성합니다.
2. CUSTOM: customDays를 사용해, 요일별 계획을 계산합니다.
3. RCMD : 기존에 설정된 함수 로직대로 계산.

**4) UsagePlan (화장품 사용 계획)**  
: 사용자가 실제로 계획한 날짜와 시간대(아침/저녁)에 대한 기록 저장

* planId (INTEGER, PRIMARY KEY AUTOINCREMENT)
* cosmeticId (INTEGER, NOT NULL)
  + FK → Cosmetic(cosmeticId)
* planDate (TEXT, NOT NULL)
  + 사용 예정 날짜 (ISO 8601, 날짜만)
* planTimeOfDay (TEXT, NOT NULL)
  + enum {"DAY", "NIGHT", "ALL"}
* createdAt (TEXT)
* updatedAt (TEXT)

**5) UsageRecord (화장품 사용 체크)**

: UsagePlan에 속한 항목의 실제 사용 여부(체크)를 기록

* recordId (INTEGER, PRIMARY KEY AUTOINCREMENT)
* planId (INTEGER, NOT NULL)
  + FK → UsagePlan(planId)
* usedTimeOfDay (TEXT, NOT NULL)
  + enum {"DAY", "NIGHT", "ALL"}, 실제 사용한 시간대
* useDate (TEXT, NOT NULL)
  + 실제 사용 날짜 (ISO 8601)
* createdAt (TEXT)
* updatedAt (TEXT)

**6) UsageEvaluation (사용 평가)**: UsageRecord에 대한 평가(좋아요/무난해요/비추)와 메모를 저장

* evaluationId (INTEGER, PRIMARY KEY AUTOINCREMENT)
* recordId (INTEGER, NOT NULL, unique)
  + FK → UsageRecord(recordId)
* rating (TEXT)
  + enum {"LIKE", "OKAY", "DISLIKE"}
* createdAt (TEXT)
* updatedAt (TEXT)

**13. 앱 테마**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**14. 와이어프레임**

**1) 대시보드 화면 (홈)**

텍스트, 스크린샷, 도표, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**2) 오늘 사용할 화장품 추가 화면**텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**3) 오늘 사용 체크 화면**

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**4) 평가 입력 화면 (선택 입력)**

* ~~전체에 대한 상세 평가(메모): 100자 이내 텍스트 입력 (업데이트 시 반영)~~

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**5) 전체 목록 화면**

텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**6) 등록된 화장품 수정/삭제 화면** 텍스트, 스크린샷, 번호, 디스플레이이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**7) 화장품 등록 화면**

텍스트, 스크린샷, 번호, 도표이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**8) 설정 화면**

* ~~데이터 초기화 or 내보내기 (업데이트 시 반영 항목)~~

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**~~9) 추천/리마인드 화면 (업데이트 시 반영 항목)~~**

* **~~1년 전 오늘 썼던 화장품 리스트~~**
* **~~계절 회고 추천: "작년 봄에 잘 썼던 템" 리스트~~**
* **~~리마인드 보기 전용 (수정 불가, 참고용)~~**

**15. Google Play 출시를 목표로 한 개발 일정의 단계**

**(1) 아이디어 구체화 및 계획 수립**

* **앱의 목적과 주요 기능 정의**  
  **: 화장품 사용 주기 + 후기 기록 앱**
* **타겟 사용자 및 시장 조사**: 일반 사용자 (자기 관리형 뷰티앱)
* **프로젝트 목표 설정 및 초안 작성**

**(2) 기획 및 설계**  
: 간단한 회원가입 또는 **회원가입 없이 바로 사용** UX 설계

* **기능 목록 확정**
* **앱 정보 구조(IA) 및 사용자 흐름(User case diagram) 작성**
* **와이어프레임 제작**
* **플레이스토어 페이지 정보 구상**
  + 앱 이름 : MY SKINCARE MATE
  + 설명 : 이 앱은 사용자가 자신이 사용하는 화장품 정보를 등록하고, 설정한 사용 주기에 따른 오늘 사용할 화장품 목록을 보여주고, 사용자가 사용 여부를 체크하고, 지정한 시간에 푸시 알림을 발송하는 기능을 제공합니다.  
    사용자는 이러한 기능들을 통해 피부를 개선할 수 있습니다.

**(3) 개발 환경 설정**

* 개발 플랫폼 결정  
  : Flutter + Android Studio 환경 구축 (에뮬레이터, 실기기 테스트)
* 필요한 툴 및 라이브러리 세팅
* 로컬 데이터 저장소 선정  
  : 회원가입 없이 사용하고 단말에 모든 데이터를 저장할 것이므로, Firebase 같은 원격 백엔드를 쓰지 않고 로컬 스토리지만으로 구현.  
  : SharedPreferences와 Drift 함께 사용.  
   - SharedPreferences에는 설정을 키-값으로 저장.  
  (데이터를 빠르고 간단하게 저장하고 가져올 때 유용하므로)  
   - Drift에는 화장품 정보, 사용 주기 및 평가 데이터 저장  
  1) Drift를 주요 저장소로 선정한 이유  
  - 테이블 스키마를 Dart 코드로 정의하고 컴파일 시 타입 검증이 되므로, SQL 작성 중 컬럼 오류를 미연에 방지할 수 있습니다.  
  - 버전별 마이그레이션을 자동 생성/관리해 주어 향후 스키마 변경 시 수작업 부담이 크게 줄어듭니다.  
  - ORM 형태의 CRUD API를 제공해, 복잡한 조인·필터·정렬 로직을 쉽게 작성할 수 있어 개발 생산성이 높아집니다.  
  - 외래 키 제약과 인덱스 설정을 그대로 사용할 수 있어, 로컬에서도 MariaDB처럼 관계형 구조를 유지하며 성능 최적화가 가능합니다.  
  - 비동기 쿼리 지원으로 UI가 멈추지 않고, 쿼리 결과를 타입 안전 객체로 받아 유지보수와 리팩터링이 용이합니다.
* 버전 관리 시스템 설정 (Git)

**(4) 기능 개발**

* 화장품 등록 기능
* 등록한 화장품 목록 보기 기능
* 등록된 화장품 편집
* 화장품 사용 체크
* 대시보드
* 푸시 알림
* 사용 평가

**(5) 테스트**

* 실제 안드로이드 기기에서 테스트
* 사용자 피드백 수집
* 동작/디자인 개선 사항 정리

**(6) 디버깅 및 수정**

* 발견된 오류 및 버그 수정
* 성능 최적화
* 디자인 및 기능 피드백 반영

**(7) 배포 준비 (기말 발표는 여기까지 준비)**

* **앱 아이콘, 스플래시 화면 추가**
* **개인정보 처리방침 작성** (플레이스토어 요구사항)
* **앱 번들(.aab) 빌드**
* **플레이콘솔에 앱 등록**
* **앱 설명, 스크린샷, 정책 문서 업로드**
* **내부 테스트 트랙 업로드 및 테스트**

**(8) 출시 (여기서부터는 방학 중 계획이므로 간트 차트에는 생략)**

* 앱 심사 요청
* 승인 후 출시
* 출시 알림 및 주변에 공유

**(9) 유지보수 및 개선**

* 초기 사용자 피드백 수집
* 버그 수정 및 업데이트
* 장기 유지 여부 판단





**페이지 관리**

