

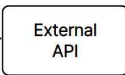








산학프로젝트(교과) 결과보고서(요약)

| | | | | |
|---|---|-----|--------------------|-----|
| 프로젝트명 | 프랜차이즈 매장을 위한 실시간 재고 관리 및 자동 발주 추천 시스템 | | | |
| 팀명 | Re:fill | | 팀장 | 전유진 |
| 참여기업 | | | 담당자 | |
| 수행기간 | 2025. 3. 10 ~ 2025. 6. 16 | | 소요예산 | |
| 소요예산 세부내역 | 세목 | | 금 액 | |
| | | | | |
| | | | | |
| | | | | |
| 참여인원 | 구분 | 인원수 | 성명(모두 기재) | |
| | 교수 | | | |
| | 석박사과정 | | | |
| | 학부생 | 4 | 전유진, 곽유나, 김서영, 양진영 | |
| | 기업체 | | | |
| | 계 | 4 | | |
| 추진배경 | 팀원들의 현장 경험을 바탕으로 수기 기반 재고 관리의 비효율과 발주 누락 문제를 해결하고자 개발된 프로젝트. ○ 팀원 모두가 실제 현장에서 재고 관리의 비효율을 체감함 - 이를 바탕으로 반복적인 재고 누락과 발주 실수 문제를 해결하기 위한 앱 개발이 기획됨 ○ 프랜차이즈 매장 상당수가 여전히 수기 또는 엑셀 기반으로 재고를 관리 중임 - 실시간 재고 파악이 어렵고, 수작업 과정에서 오류가 발생할 가능성이 높아 전반적인 운영 효율성이 저하됨 ○ 발주 마감 시간, 날씨나 공휴일 등의 외부 변수에 대한 대응 체계가 미흡함 - 제때 발주하지 못해 품질이 발생하고, 매출 손실 및 고객 불편으로 이어지는 문제가 반복됨 | | | |
| 목표 및 내용 | 프랜차이즈 매장의 비효율적인 재고 및 발주 관리를 개선하고, 실시간 관리 및 예측 기능을 갖춘 협업 중심 모바일 앱을 개발하고자 함. ○ 프랜차이즈 매장의 재고 및 발주 관리 과정을 디지털화하고, 운영 효율성을 향상하는 것이 주요 목표임 - 실시간 재고 확인, 최소 수량 기반 자동 발주, 외부 요인(날씨·공휴일 등)을 반영한 예측 시스템을 통해 수기 기반 관리의 한계를 극복하고자 함 ○ 매장 구성원 간의 협업과 본사-지점 간 정보 통합 관리 체계를 구축함 - 역할별 권한 설정, 매장 단위 실시간 채팅, 발주 알림 기능을 통해 내부 커뮤니케이션을 개선하고 운영의 일관성을 확보함 | | | |
| <div><div><div><p>Authentication</p></div><div><p>Cloud Firestore</p></div><div><p>External API</p></div></div><div><p>Re:fill</p></div><div><div><p>Functions</p></div><div><p>직원</p></div></div></div> <div>※ Re:fill 시스템 아키텍처 개요도</div> <div><div><div><p>점주(owner)</p></div><div><p>매니저(manager)</p></div><div><p>직원(staff)</p></div></div><div><div><ul style="list-style-type: none">• 재고 입력 및 수정• 팀원 초대 및 권한 설정 (매니저/직원 권한 부여)• 발주 생성 및 최종 승인</div><div><ul style="list-style-type: none">• 재고 입력 및 수정• 재고 부족 항목 확인 및 발주</div><div><ul style="list-style-type: none">• 재고 입력 및 수정</div></div><div><p>실시간 채팅, 알림 수신</p></div></div> <div>※ 사용자 역할 기반 기능 흐름도</div> | | | | |
| 기대효과 | 재고 실수 방지, 역할 기반 권한 관리, 실시간 커뮤니케이션을 통해 매장 운영의 효율성과 안정성을 높임. ○ 사람이 놓치기 쉬운 발주 실수를 최소화하고, 품질로 인한 판매 손실을 줄일 수 있음 - 최소 수량 기반의 재고 부족 경고 및 자동 발주 추천 기능을 통해 수작업의 한계를 보완함 ○ 역할 기반 권한 관리를 통해 업무 혼선 없이 안정적인 매장 운영이 가능함 | | | |

- 점주, 매니저, 직원 간 권한을 명확히 구분하고, 데이터 접근을 제어하여 정보 오남용을 방지함
- 매장 내 실시간 커뮤니케이션 체계를 구축하여 지시사항 전달 누락을 방지함
- 매장 단위 실시간 채팅과 읽음 확인 기능을 도입하여 외부 메신저 없이도 원활한 내부 소통이 가능함

1. 프로젝트 개요

| 프로젝트명 | 프랜차이즈 매장을 위한 실시간 재고 관리 및 자동 발주 추천 시스템 |
|-------|--|
| 주제영역 | <input type="checkbox"/> 생활 <input checked="" type="checkbox"/> 업무 <input type="checkbox"/> 공공/교통 <input type="checkbox"/> 금융/핀테크 <input type="checkbox"/> 의료 <input type="checkbox"/> 교육 <input type="checkbox"/> 유통/쇼핑 <input type="checkbox"/> 엔터테인먼트 <input type="checkbox"/> 기타() |
| 기술분야 | <input type="checkbox"/> IoT <input checked="" type="checkbox"/> 모바일 <input type="checkbox"/> 데스크톱 SW <input type="checkbox"/> 인공지능 <input type="checkbox"/> 보안 <input type="checkbox"/> 가상현실 <input type="checkbox"/> 빅데이터 <input checked="" type="checkbox"/> 자동제어기술 <input type="checkbox"/> 블록체인 <input type="checkbox"/> 영상처리 <input type="checkbox"/> 기타() |
| 성과목표 | <input checked="" type="checkbox"/> 논문게재 및 포스터발표 <input checked="" type="checkbox"/> 앱등록 <input type="checkbox"/> 프로그램등록 <input type="checkbox"/> 특허 <input type="checkbox"/> 기술이전 <input checked="" type="checkbox"/> 실용화 <input type="checkbox"/> 공모전(공모전명) <input type="checkbox"/> 기타() |

2. 프로젝트 추진배경

Re:fill은 재고 누락, 발주 실패, 수요 예측 오류 등 프랜차이즈 매장의 운영 비효율 문제를 해결하기 위해 개발된 프로젝트이다.

프랜차이즈 매장은 본사의 통합 지침 하에 운영되지만, 재고 및 발주 업무는 각 지점의 책임자나 직원에게 전가되는 구조가 일반적이다. 특히 중소 규모 매장에서는 재고 시스템이 제대로 갖춰져 있지 않아 여전히 수기나 엑셀 기반으로 재고를 관리하는 경우가 많다. 이로 인해 실시간 재고 파악이 어렵고 실수로 인한 재고 누락, 초과 발주, 품질 등의 문제가 빈번하게 발생한다.

또한, 프랜차이즈 구조상 발주 시간이 제한되어 있는 경우가 많아, 담당자가 부재중이면 제때 발주하지 못해 매출 손실이나 고객 응대 차질로 이어질 수 있다. 더욱이 날씨, 요일, 공휴일 등 외부 변수에 따른 수요 변화에 대응할 수 있는 체계적인 시스템이 부족하여 대부분 감과 경험에 의존한 발주가 이뤄지고 있는 실정이다.

이러한 문제는 팀원 모두가 프랜차이즈 카페 아르바이트 경험을 통해 현장에서 직접 목격하고 체감한 부분이며, 이로 인해 Re:fill 프로젝트는 실시간 재고 파악, 자동 발주 추천, 외부 요인 기반 수요 예측 기능을 통합한 실용적 모바일 플랫폼 개발을 목표로 시작되었다.

3. 프로젝트(주제) 목표 및 내용


Re:fill의 궁극적인 목표는 프랜차이즈 매장에서의 재고 관리와 발주 과정을 디지털화하고, 보다 효율적이고 정확한 운영 환경을 구축하는 것이다. 이를 위해 실시간 재고 확인, 자동 발주 추천, 외부 요인을 반영한 수요 예측 기능을 통합한 모바일 앱을 개발하였다.

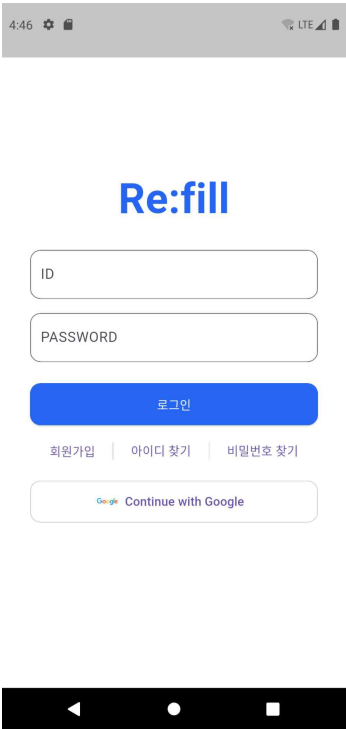
기존의 수기 또는 엑셀 기반 재고 관리 방식은 실시간 반영이 어렵고, 오류 발생 가능성이 높아 실제 매장 운영에 많은 비효율을 초래하고 있다. Re:fill은 이러한 문제를 해결하고자, 재고 수량이 최소 기준을 밑도는 경우 자동으로 발주 목록에 포함하는 시스템과 날씨 및 공휴일, 요일 등을 고려한 수요 예측 기능을 도입하였다. 이를 통해 과잉 발주나 품질 같은 문제를 사전에 방지할 수 있다.

또한, 본사와 각 매장이 동일한 기준으로 재고 및 발주 데이터를 관리할 수 있도록 정보의 통합 관리 구조를 제공하며 점주와 매니저, 직원 간의 역할을 구분하여 효율적인 팀 운영이 가능하도록 했다. 구성원 간 채팅 및 알림 기능도 탑재하여 발주나 재고 관련 사항을 실시간으로 공유할 수 있다.

Re:fill은 Flutter 프레임워크 기반으로 개발되어 Android와 iOS 양쪽에서 사용 가능하며, Firebase를 통한 데이터 연동, 사용자 인증, 클라우드 저장 기능을 지원함으로써 크로스 플랫폼 환경에서도 안정적으로 동작한다.

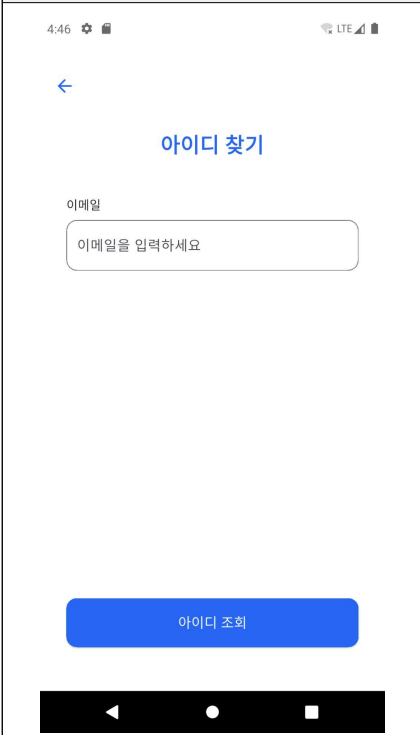
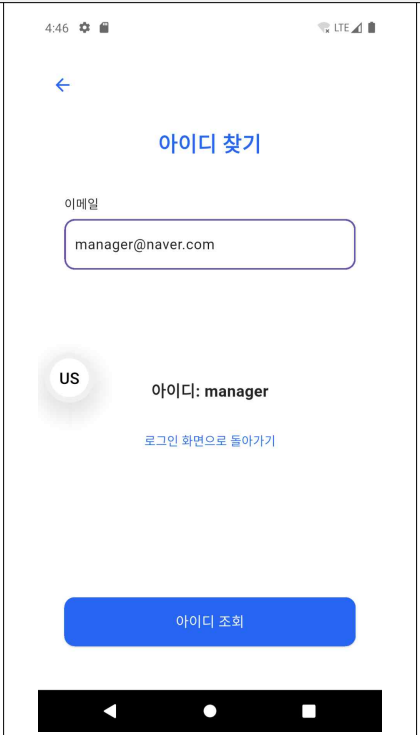
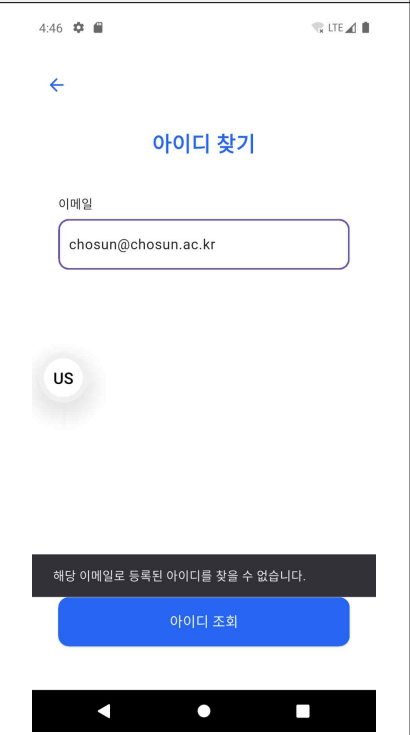
4. 프로젝트 세부내용

| Re:fill 앱의 스플래시 화면 | |
|---|--|
|  <p>※ 스플래시 화면</p> | UI 구성 <ul style="list-style-type: none"> • 배경: Re:fill의 메인 컬러인 선명한 파란색(#2563EB)을 사용함 • 중앙: 정렬된 흰색 볼드체 "Re:fill" 로고를 배치함 • 불필요한 요소 없이 단일 시각 요소로 구성해 간결하고 집중도 높은 화면을 구현함 |
| | UX 흐름 <ul style="list-style-type: none"> • 앱 실행 시 가장 먼저 나타나는 초기 화면임 • 사용자가 앱이 정상적으로 실행되고 있음을 인지하도록 짧은 시간 노출함 • 일정 시간이 지나면 로그인 화면으로 자동 전환되어 다음 단계로 이어짐 |
| | 기능 설명 <ul style="list-style-type: none"> • Firebase 초기화 및 사용자 로그인 상태를 확인함 • 앱 사용 전 필요한 데이터 로딩 작업을 수행함 • 초기화 후 로그인 화면으로 자동 전환되어 인증 절차를 시작함 |
| | 디자인 의도 <ul style="list-style-type: none"> • 브랜드 인지도를 강조하기 위한 미니멀한 화면 구성임 • 복잡한 정보 없이 로고에 시선을 집중시킬 수 있도록 설계함 • 신뢰감과 전문성, 심플함을 동시에 전달하는 디자인 의도를 반영함 |

| Re:fill 앱의 로그인 화면 | | |
|--|---------------|--|
|  <p>※ 로그인 화면</p> | UI 구성 | <ul style="list-style-type: none"> 상단: Re:fill 로고가 Re:fill의 메인 컬러로 강조되어 있음 입력 필드: ID, PASSWORD 입력창이 상하로 배치되어 있으며, 라운드 테두리로 부드러운 인상을 제공함 로그인 버튼: 파란색 배경으로 시각적 강조 효과를 주며, 가장 눈에 띄는 위치에 배치함 하단 링크: 회원가입, 아이디 찾기, 비밀번호 찾기 기능을 동일한 텍스트 스타일로 정렬함 Google 로그인 버튼: 흰 배경과 Google 로고를 조합하여 보조 로그인 옵션으로 제공함 |
| | UX 흐름 | <ul style="list-style-type: none"> 사용자는 ID와 비밀번호 입력 또는 Google 계정 연동 중 하나를 선택하여 로그인이 가능함 하단의 세 가지 텍스트 버튼을 통해 사용자는 간편하게 회원가입 및 계정 복구 절차를 진행할 수 있음 로그인 정보를 입력한 후 로그인 버튼을 누르면 로그인 성공 여부에 따라 다음 화면(홈 또는 매장 선택 화면 등)으로 이동함 |
| | 기능 설명 | <ul style="list-style-type: none"> Firebase Authentication 기반으로 로그인 기능을 구현함 ID/비밀번호 로그인 외에도 Google OAuth 연동을 통한 소셜 로그인 기능을 지원함 로그인 성공 시, 사용자 역할(점주/직원)에 따라 화면이 자동 분기되도록 처리됨 아이디 찾기 및 비밀번호 찾기 기능은 Firestore 기반 사용자 정보 검색 기능과 연동되어 작동함 |
| | 디자인 의도 | <ul style="list-style-type: none"> 전체적으로 화이트 배경에 파란색 포인트 컬러를 사용하여 깔끔하고 전문적인 인상을 전달함 사용자의 시선 흐름을 고려해 정보가 단계적으로 배치됨 |

Re:fill 앱의 회원가입 화면

| Re:fill 앱의 회원가입 화면 | | | |
|--|---|--|--|
| <div><div><div><div>4:46</div><div><div></div><div></div><div></div></div><div>LTE</div></div><div><div>←</div><div>회원가입</div><div><div><div>ID</div><div>중복확인</div></div><div><div>이름</div></div><div><div>이메일</div></div><div><div>비밀번호</div></div><div><div>비밀번호 확인</div></div><div>회원가입</div></div></div><div><div>4:46</div><div><div></div><div></div><div></div></div><div>LTE</div></div><div><div>←</div><div>회원가입</div><div><div><div>ID</div><div>중복확인</div></div><div><div>사용 가능한 ID입니다.</div></div><div><div>이름</div></div><div><div>이메일</div><div>hosun@chosun.ac.kr</div></div><div><div>비밀번호</div><div>.....</div></div><div><div>비밀번호 확인</div><div>.....</div></div><div>회원가입</div></div></div><div><div>4:46</div><div><div></div><div></div><div></div></div><div>LTE</div></div><div><div>←</div><div>회원가입</div><div><div><div>ID</div><div>중복확인</div></div><div><div>이미 사용 중인 ID입니다.</div></div><div><div>이름</div></div><div><div>이메일</div><div>chosun</div></div><div><div>비밀번호</div><div>.....</div></div><div><div>비밀번호 확인</div><div>.....</div></div><div><div>올바른 이메일 형식이 아닙니다.</div></div><div><div>특수문자를 포함해야 합니다.</div></div><div>회원가입</div></div></div></div><div><div>※ 회원가입 화면</div><div>※ 회원가입 성공 화면</div><div>※ 회원가입 실패 화면</div></div></div> | | | |
| UI 구성 | <ul style="list-style-type: none">상단: 파란색 텍스트의 "회원가입" 타이틀과 좌측에 뒤로 가기 버튼이 위치함입력 필드: ID 입력창 우측에는 중복 확인 버튼이 배치되어 있어 실시간 확인이 가능하며 그 아래로 이름, 이메일, 비밀번호, 비밀번호 확인 항목이 순차적으로 정렬됨하단: Re:fill의 메인 컬러로 회원가입 버튼이 배치되어 주요 액션을 강조함 | | |
| UX 흐름 | <ul style="list-style-type: none">사용자가 ID를 입력하고 중복 확인 버튼을 누르면 서버에서 해당 ID의 사용 가능 여부를 실시간으로 확인하여 피드백을 제공함모든 항목이 유효하게 입력된 경우, 회원가입 버튼 클릭 시 계정이 정상적으로 생성됨입력값에 오류가 있는 경우 해당 항목 아래에 빨간색 오류 메시지가 표시되며, 입력창 테두리 또한 붉은색으로 변경되어 시각적으로 오류를 즉시 인지할 수 있도록 설계함 | | |
| 기능 설명 | <ul style="list-style-type: none">Firestore와 Firebase Authentication을 연동하여 사용자 계정 정보를 등록함ID 중복 확인 기능은 Firestore의 users 컬렉션을 조회하여 실시간으로 사용 가능 여부를 반환함이메일 형식, 비밀번호 강도(예: 특수문자 포함 여부 등)는 클라이언트 측에서 유효성 검사를 수행함모든 입력 항목이 유효한 경우, Firebase Auth에 사용자 계정을 생성하고 동시에 Firestore에 사용자 메타데이터를 저장함 | | |
| 디자인 의도 | <ul style="list-style-type: none">가입 절차를 한눈에 파악할 수 있도록, 심플하고 직관적인 입력 UI로 구성함실시간 피드백(예: "사용 가능한 ID입니다", "특수문자를 포함해야 합니다")을 제공하여 사용자 혼란을 최소화함입력 오류가 발생한 항목은 테두리와 메시지를 빨간색으로 표시하여, 즉각적인 수정 유도를 돕도록 설계함 | | |

| Re:fill 앱의 아이디 찾기 화면 | | | |
|----------------------|---|--|---|
| |  |  |  |
| | ※ 아이디 찾기 화면 | ※ 아이디 찾기 성공 화면 | ※ 아이디 찾기 실패 화면 |
| UI 구성 | <ul style="list-style-type: none">상단: “아이디 찾기”라는 파란색 텍스트와 좌측 뒤로 가기 버튼이 위치함입력 필드: 이메일 주소를 입력할 수 있는 단일 입력창으로 구성됨하단: Re:fill의 메인 컬러로 아이디 조회 버튼이 배치되어 있음조회 결과는 이메일 입력창 아래에 아이디 또는 안내 문구로 표시됨실패 시에는 화면 하단에 검정 배경의 스낵바 메시지 형태로 오류 알림이 출력됨 | | |
| UX 흐름 | <ul style="list-style-type: none">사용자가 가입 시 등록한 이메일을 입력한 뒤 아이디 조회 버튼을 누르면 해당 이메일과 연동된 아이디를 확인할 수 있음일치하는 계정이 있는 경우, 아이디가 굵은 텍스트로 화면에 바로 표시되며 로그인 화면으로 돌아가기 버튼이 함께 제공됨등록되지 않은 이메일일 경우, 별도의 페이지 전환 없이 화면 하단에 오류 메시지를 표시하여 즉각적인 피드백을 제공함 | | |
| 기능 설명 | <ul style="list-style-type: none">Firestore에 저장된 사용자 정보를 이메일 기준으로 조회함입력한 이메일과 일치하는 문서가 존재할 경우, 해당 사용자의 ID가 화면에 출력됨일치하는 정보가 없을 경우, “해당 이메일로 등록된 아이디를 찾을 수 없습니다.”라는 문구가 스낵바 메시지 형태로 표시됨사용자 편의를 위해 조회 결과에 따라 로그인 화면으로 바로 이동할 수 있는 선택지를 함께 제공함 | | |
| 디자인 의도 | <ul style="list-style-type: none">필요한 입력 항목만 제공하여 간결하고 집중도 높은 인터페이스를 구현함성공과 실패 상황에 따라 시각적으로 명확히 구분되도록 설계함전체적으로 Re:fill의 메인 컬러를 일관되게 사용하여 Re:fill의 브랜드 정체성을 유지함 | | |

Re:fill 앱의 비밀번호 찾기 화면

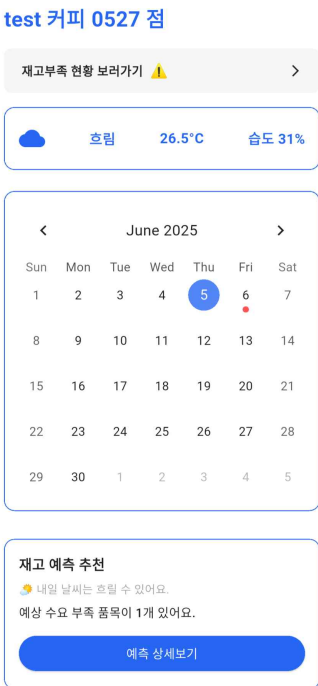
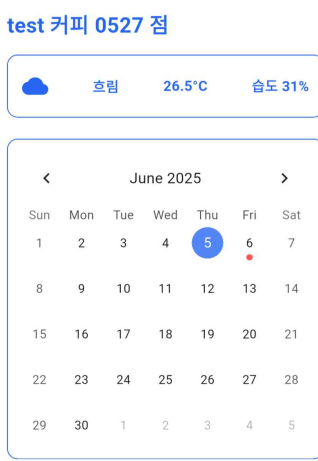
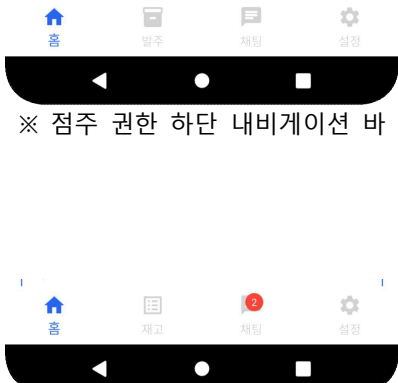
| | | |
|----------------------|--|------------------------|
| | | |
| <p>※ 비밀번호 찾기 화면</p> | <p>※ 비밀번호 찾기 성공 화면</p> | <p>※ 비밀번호 찾기 실패 화면</p> |
| <p>UI 구성</p> | <ul style="list-style-type: none"> 상단: "비밀번호 찾기"라는 파란색 텍스트와 좌측 뒤로 가기 버튼이 위치함 입력 필드: 이메일 주소를 입력할 수 있는 단일 입력창으로 구성됨 하단: Re:fill의 메인 컬러로 비밀번호 재설정 메일 보내기 버튼이 배치되어 있고 로그인 화면으로 돌아가기 링크 텍스트가 표시됨 성공 및 실패 결과는 스낵바 메시지 형태로 사용자에게 전달됨 | |
| <p>UX 흐름</p> | <ul style="list-style-type: none"> 사용자가 가입 시 등록한 이메일을 입력하면 Firebase를 통해 해당 이메일로 비밀번호 재설정 메일이 발송됨 유효한 이메일 입력 시 "비밀번호 재설정 메일을 보냈습니다"라는 안내 메시지가 출력됨 등록되지 않은 이메일일 경우, "해당 이메일로 가입된 계정이 없습니다"라는 오류 메시지가 하단에 표시됨 화면을 유지한 채 피드백이 제공되므로 사용자는 즉시 수정 후 재시도할 수 있음 | |
| <p>기능 설명</p> | <ul style="list-style-type: none"> Firebase Authentication의 sendPasswordResetEmail() 기능을 활용하여 비밀번호 재설정 메일을 발송함 입력된 이메일의 유효성을 검사하고 가입된 계정 여부를 확인한 뒤, 조건이 충족되면 전송이 이루어짐 실패 시에는 서버로부터 전달받은 에러 메시지를 사용자에게 안내함 전송된 이메일에는 비밀번호 재설정 링크가 포함되어 있으며 사용자가 직접 비밀번호를 변경할 수 있음 | |
| <p>디자인 의도</p> | <ul style="list-style-type: none"> 단일 입력창 구성으로 사용자 시선을 분산시키지 않고 집중도를 높임 스낵바 메시지를 활용해 간결하면서도 자연스러운 상태 피드백을 제공함 별도의 화면 전환이 없는 흐름으로 이어져 직관적인 사용자 경험을 유도함 | |

Re:fill 앱의 초기 진입 화면 (권한 설정 화면)

| | |
|---|--|
| <div><div><div>10:53</div><div>새로운 매장 생성</div></div><div><div>←</div><div>새로운 매장 생성</div></div><div><div>앞 단어 (예: 서울대)</div><div>커피 뒤 단어 (예: 정문점)</div></div><div><div>주소</div></div><div><div>생성</div></div></div> <div>※ 첫 로그인 시 매장 선택 화면 (권한 설정 화면)</div> | <div><div><div>10:53</div><div>초대 코드 입력</div></div><div><div>←</div><div>초대 코드 입력</div></div><div><div>초대 코드를 입력하세요.</div></div><div><div>매장 가입하기</div></div></div> <div>※ 초대 코드 입력 화면</div> |
| <div><div>UI 구성</div></div> <div><ul style="list-style-type: none">• 회원가입 후 첫 로그인 시, 계정에 등록된 매장이 없는 사용자에게 표시되는 화면임• 상단: Re:fill 로고와 함께 메인 슬로건 “자동 발주, 똑똑한 재고 관리의 시작”이 표시되며, 그 아래로 두 개의 선택 카드가 배치됨• + 새로운 매장 생성 버튼: 사람과 문서 아이콘 일러스트와 함께 구성되어, 사용자가 팀을 만든다는 흐름의 시각적 유도를 제공함• + 초대 코드 입력 버튼: 두 사람이 링크 아이콘을 들고 있는 일러스트와 함께 구성되어, 기존 팀에 멤버로 참여할 수 있다는 흐름을 안내함</div> | <div><div>UX 흐름</div></div> <div><ul style="list-style-type: none">• 이 화면은 앱 최초 로그인 시 한 번만 나타나며, 사용자의 역할을 설정하는 진입점 역할을 수행함• 사용자가 + 새로운 매장 생성 버튼을 선택하면 store 정보를 입력하는 절차로 이어지며, 해당 사용자에게는 점주(owner) 권한이 부여됨• 사용자가 + 초대 코드 입력 버튼을 선택하면 기존 매장에 합류하게 되며, 자동으로 직원(staff) 권한이 부여됨• 역할 설정이 완료되면, 이후 홈 화면으로 이동함</div> |
| <div><div>기능 설명</div></div> <div><ul style="list-style-type: none">• 매장 생성 시 Firestore의 stores 컬렉션에 새로운 문서가 생성되며, 생성자의 UID는 role: owner 로 등록됨• 초대 코드는 Firestore에 미리 발급된 매장 코드와 매칭되어야 하며, 유효한 경우 사용자가 해당 매장의 staff로 추가됨• 이후 로그인 시, 사용자 계정에 연결된 storeId를 기반으로 홈 화면 UI가 자동으로 분기됨• 사용자 권한은 Firestore의 users 컬렉션에 role 필드로 저장되며, 이후 화면 전환 흐름 및 기능 제어에 활용됨</div> | |


| | |
|-------------------|--|
| 디자인 의도 | <ul style="list-style-type: none"> • 처음 앱에 접속한 사용자가 자신의 역할을 직관적이고 자연스럽게 선택할 수 있도록 구성함 • 일러스트가 포함된 카드형 버튼을 사용하여 단순 텍스트 버튼보다 시각적 설득력과 집중도를 높임 • 점주(owner)와 직원(staff)의 진입 흐름을 명확히 분리하여, 이후 권한 기반 기능 제한이 정확하게 적용되도록 설계함 |
|-------------------|--|

Re:fill 앱의 권한별 홈 화면 및 하단 내비게이션 바 구성

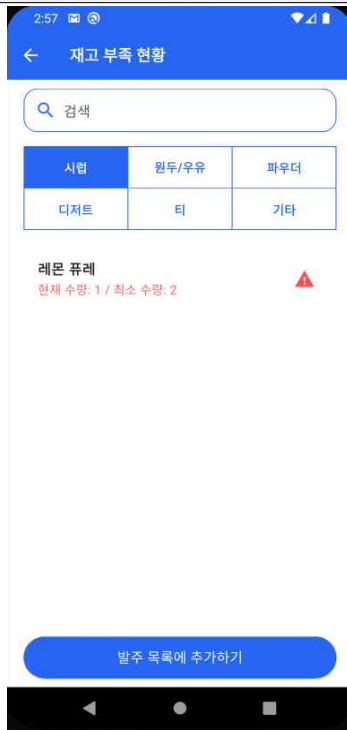
| | | |
|---|---|--|
|  <p>test 커피 0527 점</p> <p>재고부족 현황 보러가기</p> <p>흐림 26.5°C 습도 31%</p> <p>June 2025</p> <p>재고 예측 추천</p> <p>내일 날씨는 흐릴 수 있어요. 예상 수요 부족 품목이 1개 있어요.</p> <p>예측 상세보기</p> <p>※ 점주 및 매니저 권한 홈 화면</p> |  <p>test 커피 0527 점</p> <p>흐림 26.5°C 습도 31%</p> <p>June 2025</p> <p>※ 직원 권한 홈 화면</p> |  <p>※ 점주 권한 하단 내비게이션 바</p> <p>※ 매니저 및 직원 권한 하단 내비게이션 바</p> |
|---|---|--|

| | |
|------------------|---|
| UI 구성 | <ul style="list-style-type: none"> • 상단: 현재 매장 이름과 실시간 날씨 정보(날씨 아이콘, 기온, 습도)가 함께 표시되어 외부 환경을 고려한 재고 예측에 도움을 줌 • 중단: 날짜 확인이 가능한 달력 위젯을 배치하여 일자 기반 업무 흐름 관리가 가능함 • 점주(owner)와 매니저(manager) 권한을 가진 사용자의 경우, 홈 화면 최상단에 재고 부족 현황 보러가기 버튼과 하단에 재고 예측 추천 기능이 배치됨 • 하단 내비게이션 바 구성은 사용자 권한에 따라 다르게 적용됨: 점주 - 홈/발주/채팅/설정, 매니저·직원 - 홈/재고/채팅/설정 • 매니저 및 직원은 발주 기능에 직접 접근하지 못하며, 이를 통해 권한에 따른 기능 접근 제한을 명확히 구현함 |
| UX 흐름 | <ul style="list-style-type: none"> • 로그인 이후, 사용자의 권한(role)에 따라 자동으로 맞춤형 홈 화면 UI와 하단 내비게이션 바가 설정됨 • 점주(owner)는 발주 및 매장 관리 기능 중심의 홈 화면을 제공받으며, 매니저(manager)와 직원(staff)은 실시간 데이터 확인과 내부 커뮤니케이션 중심의 UI로 구성됨 • 역할별로 최적화된 기능만 노출되며, 하단 내비게이션 바를 통해 각 권한에 맞는 주요 화면으로 빠르게 이동할 수 있음 |

| | |
|--------|---|
| 기능 설명 | <ul style="list-style-type: none"> • Firebase Firestore의 users 컬렉션 내 role 필드를 기준으로 UI 구성 요소를 조건별로 분기함 • 사용자 권한에 따라 Firestore 데이터의 읽기/쓰기 권한을 차등 설정함 • 발주 기능은 stocks/{storeid}/items, orderTemplates 등의 경로를 기반으로 Firestore 데이터와 자동 연동됨 • 하단 내비게이션 바는 Flutter의 BottomNavigationBar 위젯으로 구성되며, 선택된 탭은 Re:fill의 메인 컬러로 강조함 |
| 디자인 의도 | <ul style="list-style-type: none"> • 역할 기반 UI 차별화를 최소한의 시각적 변경만으로 자연스럽게 전달하여 사용자 간 위화감을 최소화함 • 하단 내비게이션 바는 간결한 아이콘 중심 구성으로, 사용자가 직관적이고 빠르게 주요 기능에 접근이 가능함 • 권한별 기능 차이는 UI에서 불필요하게 드러내지 않도록 설계하여, 전체적인 브랜드 일관성과 사용자 신뢰감을 유지함 |

| Re:fill 앱의 홈 화면 주요 버튼 설명 ①: 주간 날씨 예보 화면 | | | |
|---|--------|---|--|
|  <p>※ 주간 날씨 예보 화면</p> | UI 구성 | <ul style="list-style-type: none"> • 상단: Re:fill의 메인 컬러의 "날씨 예보" 타이틀과 뒤로 가기 아이콘을 배치해 명확한 화면 맥락을 전달함 • 날짜/요일/날씨 상태/최고·최저 기온을 한 줄에 정돈된 형태로 수평 정렬하여 정보 탐색이 용이함 • 날씨 상태는 아이콘과 텍스트의 조합으로 시각적 직관성과 함께 언어적 설명을 병행해 사용자의 이해도 향상에 도움을 줌 | |
| | UX 흐름 | <ul style="list-style-type: none"> • 홈 화면 상단의 날씨 위젯 클릭 시 진입함 • 사용자는 5일간 날씨 변화를 직관적으로 확인이 가능함 • 뒤로 가기 클릭 시 홈 화면으로 자연스럽게 복귀함 | |
| | 기능 설명 | <ul style="list-style-type: none"> • 외부 날씨 API(OpenWeatherMap)로부터 주간 날씨 예보를 가져와 Provider 또는 Firestore로 캐싱함 • 각 날짜 정보는 예측 발주 함수 및 /recommendations/{storeid} 문서에서 발주량 계산에 활용됨 • 기상 정보는 수요 예측에 영향을 주는 핵심 요소로 연결됨 | |
| | 디자인 의도 | <ul style="list-style-type: none"> • 아이콘 중심의 구성으로 모바일에서도 빠르게 정보 인지가 가능함 • 반복적 레이아웃을 통해 예측 흐름을 방해하지 않는 서브 정보 구조임 • 날씨 정보 자체보다는 예측 기반 발주 UX를 위한 보조 기능으로 설계함 | |

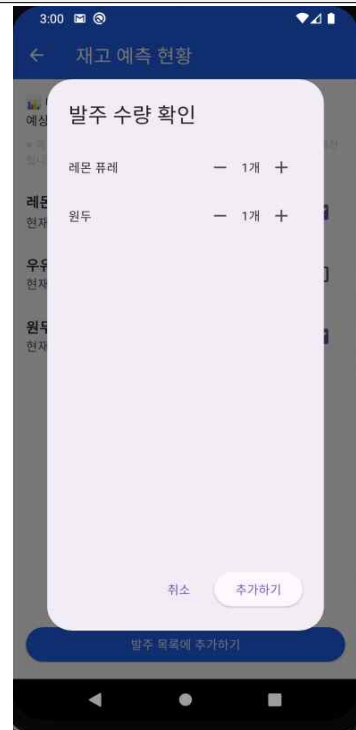
Re:fill 앱의 홈 화면 주요 버튼 설명 ②: 재고 부족/예측 현황 화면



※ 재고 부족 현황 화면



※ 재고 예측 현황 화면



※ 재고 예측 현황 발주 화면

UI 구성

- 상단: “재고 부족 현황” 또는 “재고 예측 현황” 제목이 표시되어 현재 상태를 구분할 수 있음
- 재고 부족 현황 화면: 각 품목의 현재 수량과 함께 경고 아이콘이 나타나며, 하단에는 발주 목록에 추가하기 버튼이 배치되어 부족한 항목을 바로 발주로 연동 가능함
- 재고 예측 현황 화면: 날씨나 공휴일 등 외부 요인 기반으로 추천된 항목들이 리스트 형태로 출력되며, 각 항목 옆에 선택용 체크박스가 포함됨
- 사용자가 항목을 선택하면 버튼이 활성화되고, 발주 목록에 추가하기 버튼을 클릭하면 발주 화면으로 연동되어 각 품목의 발주 수량을 조절할 수 있음
- 수량 조절 후 추가하기 버튼을 누르면 해당 정보가 발주 화면으로 연동됨

UX 흐름

- 홈 화면에서 재고 부족 현황 보러가기 버튼 또는 예측 상세보기 버튼을 클릭하면 해당 화면으로 이동함
- 사용자에게 부족하거나 추천된 품목 리스트가 보이고, 필요한 항목만 체크하여 선택할 수 있음
- 선택한 항목은 수량 조절이 가능하며, 추가 버튼을 누르면 발주 화면에 자동으로 반영됨

기능 설명

- 실시간 재고 정보는 `stocks/{storeId}/items`에서 불러오고, 예측 데이터는 `recommendations/{storeId}` 경로의 Firestore 문서에서 수신됨
- 발주 항목은 `orderTemplates` 컬렉션과 연동되어 자동 발주 구성이 가능함
- 재고 부족 상태는 `quantity < minQuantity` 조건으로 판단하며, UI에서도 경고 아이콘으로 강조 표시함
- 예측 수요 계산은 Firebase Cloud Functions의 `generateStockRecommendations` 함수를 통해 계산되어 저장됨
- 사용자가 선택한 품목은 수량 조절 후 상태가 유지된 채 발주 화면으로 전달되어 실제 발주로 이어질 수 있음

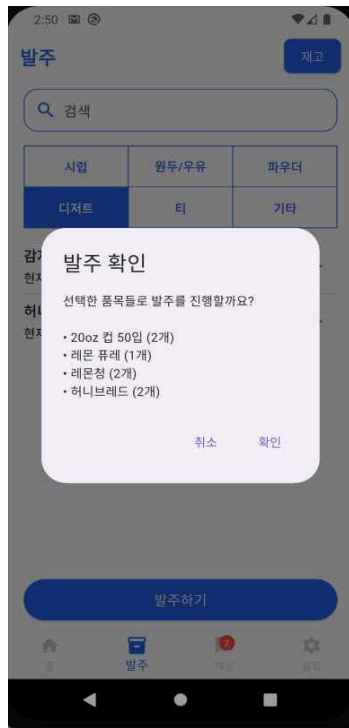
디자인 의도

- 체크박스과 수량 조절로 직관적인 발주가 가능함
- 자동 추천과 사용자의 판단이 결합된 하이브리드 UI임

Re:fill 앱의 발주 화면



※ 발주 화면



※ 발주 확인 팝업

UI 구성

- 발주 화면은 “시럽”, “원두/우유” 등 카테고리 탭으로 구성되어 사용자가 원하는 품목 군을 빠르게 탐색할 수 있도록 함
- 각 카테고리 내에서는 품목명, 현재 수량, 최소 수량, 발주 수량을 한눈에 비교할 수 있게 리스트 형태로 시각화함
- 우측 수량 조절 버튼은 사용자가 직관적으로 발주 수량을 설정할 수 있도록 도와주며, 하단의 발주하기 버튼을 통해 전체 발주 흐름을 이어감
- 발주하기 버튼 클릭 시, 선택된 품목들의 발주 수량을 요약 정리한 팝업이 표시되어 사용자 확인 후 최종 발주 진행이 가능함

UX 흐름

- 홈 화면 또는 예측/부족 추천 화면에서 발주 화면으로 자연스럽게 이동되는 흐름으로 구성함
- 사용자에게는 품목별 추천 수량이 미리 입력되어 있으며, 직접 수량을 조절할 수 있어 유연한 발주가 가능함
- 수량 조절 후 발주하기 버튼을 누르면 발주 확인 팝업이 뜨고, 사용자는 최종적으로 발주 내용을 검토한 뒤 실행 여부를 결정함
- 발주 확정 시, 데이터는 Firebase에 저장되며 화면은 초기화되어 다음 발주 준비 상태로 전환됨
- 확인 팝업 단계를 통해 실수로 인해 잘못된 발주를 예방하고, 전체 흐름의 안정성을 강화함

기능 설명

- 실시간 재고 정보는 `stocks/{storeId}/items` 경로에서 불러오며, 각 품목의 수량 정보가 발주 화면에 즉시 반영됨
- 발주 수량은 `orderTemplates` 구조를 기준으로 관리되며, 사용자가 입력한 값은 항목 별로 개별 저장됨
- 발주가 완료되면 `orders/{storeId}` 경로에 새로운 문서가 자동 생성되어 실제 발주 내역이 기록됨
- 사용자가 발주 화면을 벗어났다가 돌아오더라도 입력했던 수량이 유지되도록 상태 저장 또는 캐싱 처리를 적용함
- 품목별 수량 조절은 Flutter의 상태 관리 방식인 `setState` 또는 `Provider` 기반으로 구현되어 직관적인 UI 반응성을 제공함

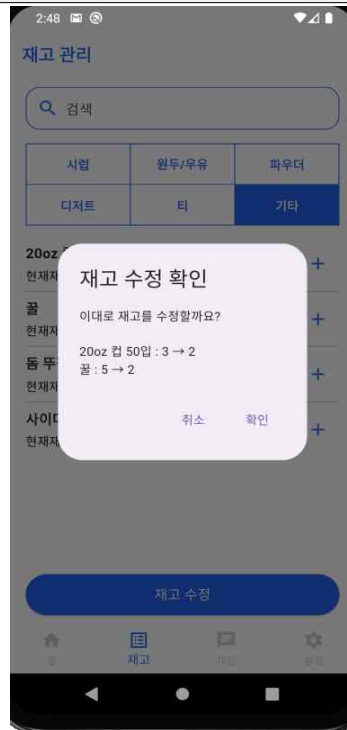
디자인 의도

- 테이블 기반 UI를 통해 품목별 수량 비교와 발주 입력 과정을 직관적으로 구현함
- 카테고리 탭으로 품목을 분류하여 가독성과 탐색 효율을 동시에 확보함
- 발주 전 팝업창을 통해 사용자가 입력 내용을 한 번 더 확인할 수 있도록 설계, 실행 실수 예방함
- 전체 화면 구성은 실제 발주 경험을 고려한 구조 설계임
- 상단 검색 필드를 통해 원하는 품목을 빠르게 탐색할 수 있도록 지원, 정보 접근성을 높임

Re:fill 앱의 재고 화면



※ 점주 권한의 재고 관리 화면



※ 직원 권한의 재고 관리 화면
및 팝업



※ 직원 권한의 재고 관리 성공
화면

UI 구성

- 점주의 경우, 발주 화면에서 우측 상단 재고 버튼을 클릭하면 바로 진입할 수 있음
- 점주의 화면에서는 최소 수량까지 함께 표시되어 발주 기준 판단이 가능하며, 직원 화면은 단순 수량 조절 중심으로 간소화함
- 수량 수정 후에 재고 수정 버튼을 누르면 "재고 수정 확인" 팝업이 등장하며 변경 내용을 한번 더 검토할 수 있도록 구성하여 실수를 방지함
- 최종 저장 시 확인 메시지를 통해 사용자에게 결과를 명확히 피드백하며, 전체적으로 역할에 따라 정보를 차별화된 방식으로 제공해 관리 효율을 높임

UX 흐름


- 점주는 발주 화면에서 재고 수정이 필요한 경우 우측 상단 버튼을 통해 즉시 재고 관리 화면으로 진입할 수 있도록 구성함
- 직원은 하단 내비게이션 바의 재고 탭을 통해 상시 접근할 수 있으며, 현장 재고 변화에 따라 실시간 수량 수정이 가능함
- 사용자 권한(role)에 따라 진입 방식과 화면 정보에 차이를 두어, 점주는 전반적인 흐름과 기준을 설정하고, 직원은 현장 상황을 반영하는 보조 역할에 집중하도록 설계함
- 이로써 운영 책임자와 현장 담당자의 역할이 자연스럽게 분리되며, 동시에 하나의 시스템 안에서 효율적인 협업이 가능하도록 지원함

기능 설명

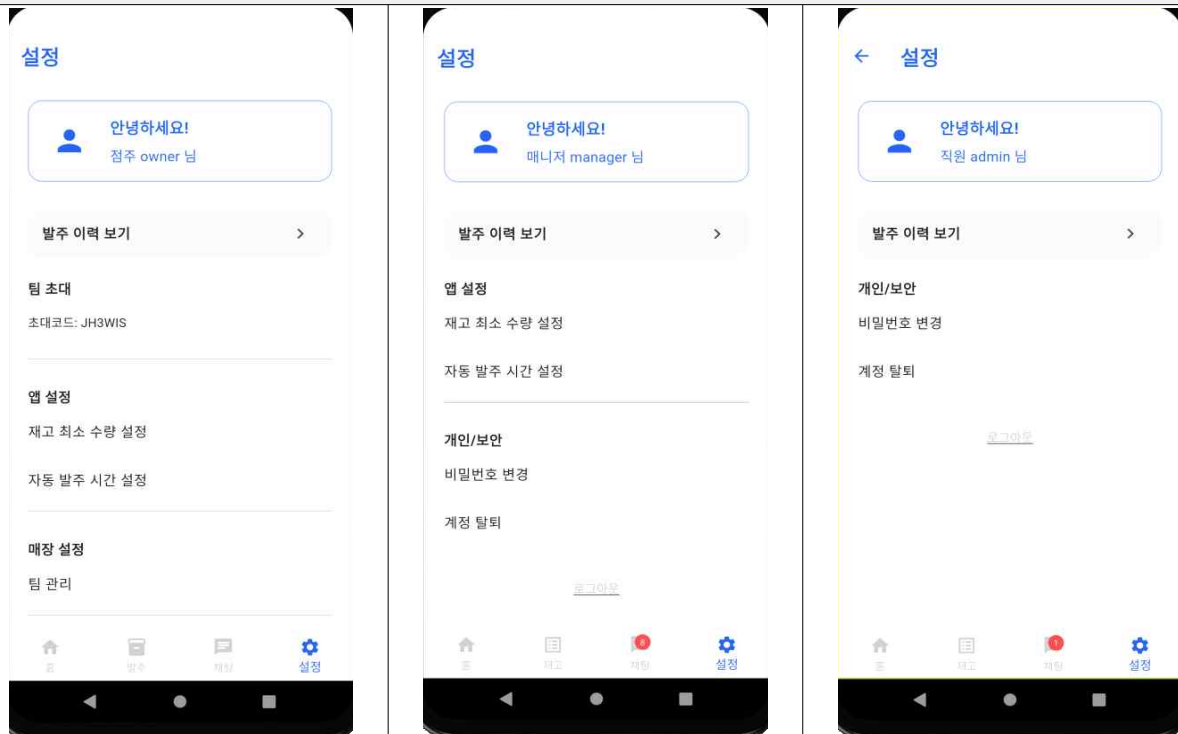
- 실시간 재고 관리는 Firestore의 stocks/{storeId}/items 경로에서 이루어짐
- 여러 항목을 한 번에 저장할 경우 Firestore의 write batch 기능을 활용하여 트랜잭션성이 있는 저장 처리가 가능함
- 저장이 완료되면 UI에 실시간으로 반영되고, 성공 여부는 스낵바 메시지 형태로 즉각 피드백함

디자인 의도

- 역할에 따라 기능을 분리하여 혼선을 줄이고 각자의 책임을 명확히 함
- 재고 수정 전, 재고 수정 확인 팝업을 통해 사용자의 실수를 방지하고 안정성을 높임
- 상단 검색 필드를 통해 원하는 품목을 빠르게 탐색할 수 있도록 지원, 정보 접근성을 높임

| Re:fill 앱의 매장 전용 채팅 화면 | | |
|---|---|--|
|  <p>※ 매장 전용 채팅 화면</p> <p>1 채팅</p> <p>※ 빨간 배지로 알림 표시</p> | UI 구성 <ul style="list-style-type: none"> • 각 메시지에 발신자 역할이 아이콘으로 함께 표시되어 사용자 간 구분이 쉬움 • 말풍선 색상 및 정렬(자신: 파란색/오른쪽, 타인: 회색/왼쪽)을 통해 직관적인 대화 흐름을 유도함 • 메시지 옆에 실시간 읽음 수 확인 기능을 제공하여 커뮤니케이션의 투명성 강화함 • 채팅 아이콘에 빨간 배지로 미확인 메시지를 알려 사용자의 반응을 유도함 | |
| | UX 흐름 <ul style="list-style-type: none"> • 사용자는 매장 내 구성원들과 실시간으로 메시지를 주고받을 수 있음 • 새로운 메시지가 도착하면 하단 채팅 탭에 빨간색 배지가 표시되어 빠른 접근을 유도함 • 입력한 메시지는 Firebase Realtime Database 또는 Firestore에 즉시 저장되고, 실시간으로 화면에 반영됨 | |
| | 기능 설명 <ul style="list-style-type: none"> • Firebase 기반의 실시간 채팅 구조로, 메시지 전송 시 작성자 ID, 역할, 내용, 타임스탬프, 읽음 상태가 함께 저장됨 • 각 메시지에 대해 사용자별 읽음 여부를 기록하고, 이를 바탕으로 읽음 수 또는 모두 읽음 상태를 표시함 • 채팅 탭의 알림 배지는 Firestore의 message 컬렉션 변경 사항을 감지하여 실시간으로 UI에 반영됨 • 역할(role) 필드를 기반으로 점주/매니저/직원 아이콘이 자동 출력되며, 가독성과 메시지 이해도를 높임 | |
| | 디자인 의도 <ul style="list-style-type: none"> • 말풍선 색상과 정렬(자신: 파란색/오른쪽, 타인: 회색/왼쪽)을 통해 메시지 흐름을 직관적으로 파악 가능함 • 역할에 따라 각각의 아이콘으로 시각화되어 커뮤니케이션의 명확성을 높임 • 하단 내비게이션 바로 빠른 화면 전환이 가능하며, 새 메시지는 빨간색 배지로 알림을 제공해 커뮤니케이션 누락을 방지함 | |

Re:fill 앱의 권한별 설정 화면




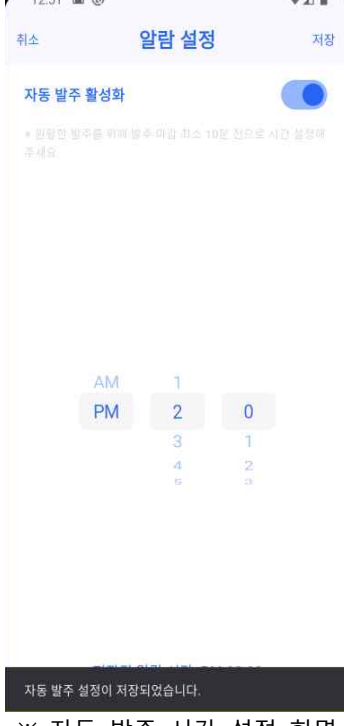
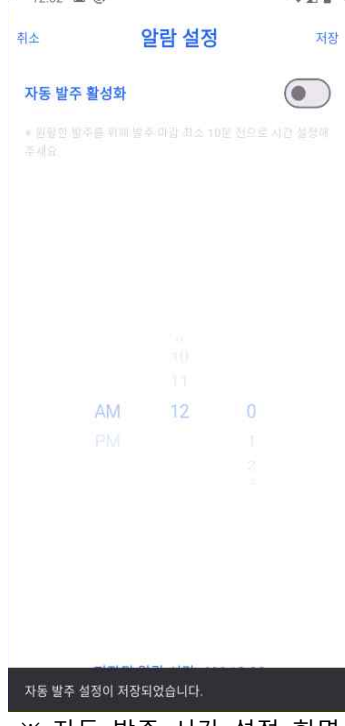
※ 점주 권한 설정 화면

※ 매니저 권한 설정 화면


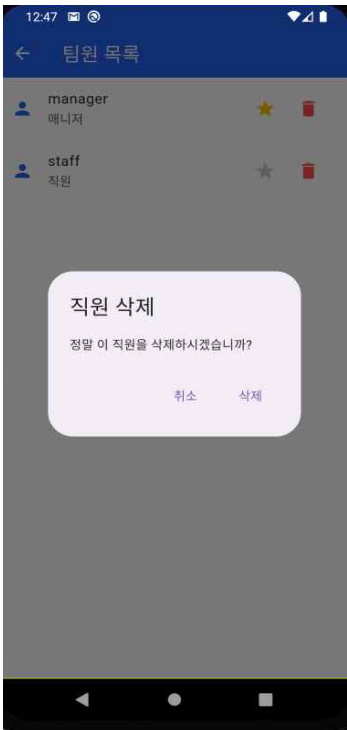
※ 직원 권한 설정 화면

| | |
|---------------|--|
| UI 구성 | <ul style="list-style-type: none"> 상단: 사용자의 이름과 역할을 함께 표시해 개인화된 인사 및 권한 인지를 돕는 구조임 점주는 팀 관리 및 알림 설정 등 고급 기능을 사용할 수 있고, 매니저는 일부 권한만 제한적으로 사용이 가능함 직원은 기본적인 정보 확인과 보안 설정만 가능하도록 기능이 간소화되어 역할별 사용 경험 이 자연스럽게 구분되도록 설계됨 |
| UX 흐름 | <ul style="list-style-type: none"> 로그인 시, 사용자의 역할(role)은 Firestore의 users 컬렉션에서 자동으로 불러와 설정 화면 렌더링에 반영됨 점주/매니저/직원 각각에게 기능 접근 범위를 다르게 설정하여 실제 권한 기반의 사용 흐름을 구현함 화면 레이아웃은 통일된 구조를 유지하되, 기능 버튼의 노출 여부만으로 역할 차이를 표현해 사용자 경험의 일관성과 명확성을 동시에 확보함 |
| 기능 설명 | <ul style="list-style-type: none"> 각 사용자의 역할은 Firestore users 컬렉션 내 role 필드(owner, manager, staff)로 명확히 구분됨 설정 화면 진입 시, 해당 역할을 기반으로 기능이 조건부 렌더링되며 불필요한 기능 노출을 제한함 초대 코드 생성 및 팀 관리 권한은 오직 점주에게만 허용됨 이러한 구조는 권한 오용을 방지하고, 실수로 인한 설정 변경이나 데이터 손상 가능성을 최소화하기 위한 보안 설계임 |
| 디자인 의도 | <ul style="list-style-type: none"> 전체적인 시각적 일관성을 유지하면서도 역할에 따라 기능 노출 여부만 조정해 복잡도를 줄임 사용자는 화면 구성 자체는 동일하게 경험하지만, 자신에게 필요한 기능만 표시되어 혼란 없이 권한 범위를 인지 가능함 하나의 UI 구조 안에서 버튼의 노출/비노출만 다르게 구성함으로써 단순하고 효율적인 권한 기반 UX를 구현함 |

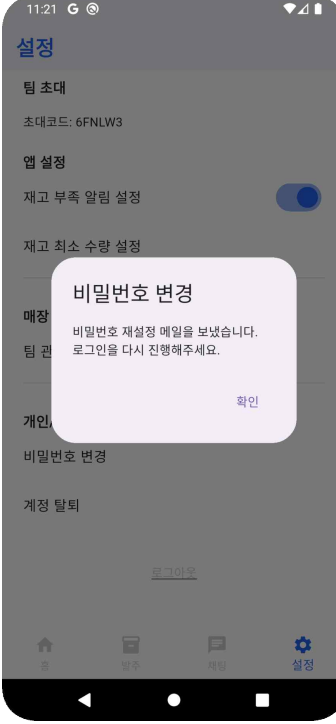
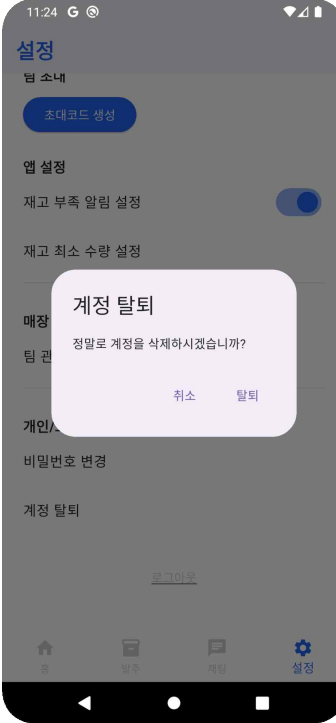
Re:fill 앱의 설정 화면 ①: 앱 설정 - 점주(owner), 매니저(manager) 권한 전용

| | | | |
|---|---|--|--|
|  <p>※ 재고 최소 수량 설정 화면</p> |  <p>※ 자동 발주 시간 설정 화면 (활성화)</p> |  <p>※ 자동 발주 시간 설정 화면 (비활성화)</p> | |
| UI 구성 | <ul style="list-style-type: none">최소 수량 설정은 품목별 입력 필드를 활용해 매장 상황에 맞는 기준값을 유연하게 설정할 수 있도록 함자동 발주 기능은 별도의 토글 버튼과 시간 선택 위젯을 통해 시각적으로 명확하게 표현되며, AM/PM 구분과 시간/분 단위 선택이 직관적으로 이루어지도록 설계됨하단: 설정된 정보는 실시간으로 표시됨관리자 전용 UI로서 불필요한 조작을 방지하고 전체 설정 흐름을 직관적이고 안정감 있게 설계함 | | |
| UX 흐름 | <ul style="list-style-type: none">최소 수량은 입력과 동시에 Firestore에 저장되며, 재고 부족 여부 판단 및 발주 예측 기능에 실시간으로 반영됨사용자는 각 품목별 기준을 자유롭게 설정할 수 있어 매장 운영 특성에 맞는 조건 설정이 가능함설정된 발주 시간은 Firestore 및 SharedPreferences에 모두 저장되며, 설정 완료시 해당 시각이 하단에 표시되어 사용자가 변경 여부를 쉽게 인지할 수 있음설정 값은 앱 재시작 시에도 유지되며, 발주 예정 시간 이전에 알림 전송 및 자동 실행이 가능하도록 흐름을 설계함 | | |
| 기능 설명 | <ul style="list-style-type: none">최소 수량 설정은 Firestore의 <code>/stocks/{storeId}/items</code> 경로에 각 품목의 <code>minQuantity</code>에 값을 저장하며, 재고 예측 및 자동 발주 실행 시 이 값을 기준으로 부족 여부 판단에 활용됨자동 발주 기능은 Firebase 기반 background service와 연동되어 사용자가 설정한 시간에 <code>autoOrderExecution()</code>이 실행됨<code>/recommendations/{storeId}/items</code>의 <code>recommendedExtra > 0</code>인 항목을 발주 대상으로 삼고 <code>/orders</code>에 저장 후 동시에 재고 수량을 증가 처리함 | | |
| 디자인 의도 | <ul style="list-style-type: none">자동 발주라는 핵심 기능의 신뢰성과 직관성을 강조하기 위해, 설정 인터페이스를 단순하고 명확하게 구성함사용자 실수를 줄이기 위해 시간 선택은 iOS 스타일로 구현했으며, 토글 버튼을 통해 기능 ON/OFF 여부를 쉽게 제어할 수 있도록 함관리자의 설정 권한이 강조되도록 일반 직원은 접근 불가능하게 설계함 | | |

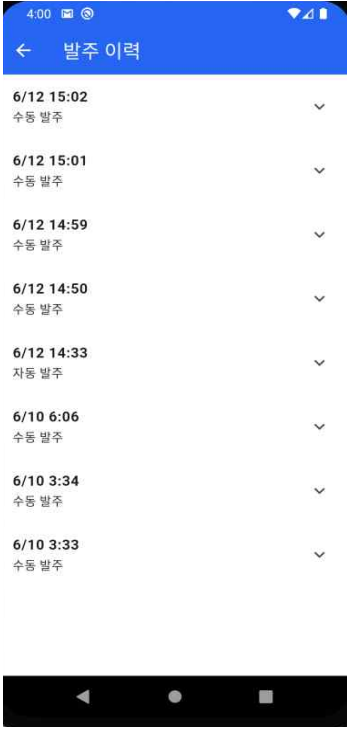

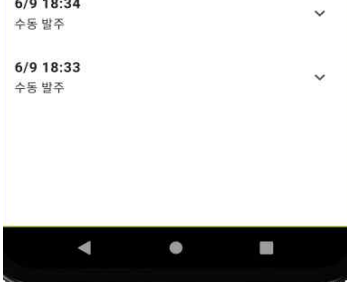
Re:fill 앱의 설정 화면 ②: 매장 설정 - 점주(owner) 권한 전용

| | | |
|--|----------------------|--|
|  <p>※ 팀원 목록 화면</p> | <p>UI 구성</p> | <ul style="list-style-type: none"> • 팀원 목록 화면은 점주(owner) 권한에서만 접근 가능하며, 모든 팀원의 이름과 직책(직원/매니저)이 리스트 형태로 표시됨 • 각 팀원 옆에는 별 아이콘을 활용해 매니저 직책 여부를 직관적으로 구분할 수 있으며 점주는 해당 아이콘을 눌러 역할을 실시간으로 변경할 수 있음 • 목록은 Firebase 기반으로 자동 동기화되며, 리스트 스타일은 간결하고 관리 중심의 구성으로 설계됨 |
| <p>※ 팀원 목록 화면</p> | <p>UX 흐름</p> | <ul style="list-style-type: none"> • 점주는 버튼 클릭 한 번으로 팀원의 역할을 매니저에서 직원으로, 직원에서 매니저로 전환할 수 있으며 변경 즉시 Firestore에 반영되어 다른 사용자에게도 실시간 적용됨 • 사용자는 별도의 저장 버튼 없이 즉시 적용되는 반응형 UX를 경험할 수 있으며 직관적인 아이콘 UI를 통해 팀원 권한 구조를 빠르게 파악하고 조정할 수 있음 |
|  <p>※ 직원 삭제(휴지통) 팝업</p> | <p>기능 설명</p> | <ul style="list-style-type: none"> • 팀원 데이터는 Firestore의 stores/{storeId}/members 경로에 저장되어 있으며 role 필드 값을 staff 또는 manager로 업데이트함 • 역할 변경은 onTap 이벤트를 통해 처리되며 실시간으로 Firestore에 반영됨 • 데이터 변경 후 전체 팀원 목록 UI에 자동 업데이트 적용되어 별도 새로고침 없이 즉시 반영됨 |
| <p>※ 직원 삭제(휴지통) 팝업</p> | <p>디자인 의도</p> | <ul style="list-style-type: none"> • 점주의 권한을 시각적으로 명확히 전달하고, 역할 변경을 최소한의 조작으로 수행할 수 있도록 경량화된 관리자 UI를 설계함 • 별 아이콘을 통한 직책 구분은 상징성과 실용성을 동시에 갖추며 앱 내 커뮤니케이션과 권한 구조를 명확히 인식하도록 돕는 UI 요소임 • 설정 화면의 하위 기능으로서 통일된 스타일을 유지하며 실시간 협업 및 운영 관리에 중점을 둔 구조로 구성함 |

Re:fill 앱의 설정 화면 ③: 개인/보안

| | | |
|--|----------------------|---|
|  <p>※ 비밀번호 변경 설정 화면</p> | <p>UI 구성</p> | <ul style="list-style-type: none"> 계정 관련 기능(비밀번호 변경, 계정 탈퇴, 로그아웃)은 모든 사용자에게 동일하게 제공되며, 접근 경로를 단순화하여 빠른 조작이 가능함 민감한 작업(비밀번호 변경, 계정 탈퇴)은 팝업창을 통해 사용자의 실수를 방지하고 재확인을 유도함 |
| | <p>UX 흐름</p> | <ul style="list-style-type: none"> 사용자는 설정 화면을 통해 자신의 계정 상태를 직접 관리할 수 있으며, 비밀번호 변경은 Firebase의 이메일 인증 기반으로 처리됨 계정 탈퇴는 파괴적 행동을 인식시키기 위해 팝업을 띄우고, 최종 클릭 후 Firebase Auth 계정과 Firestore 사용자 문서가 함께 삭제됨 처리 후에는 자동으로 로그인 화면으로 리다이렉션되어 사용자 흐름이 단절되지 않도록 설계됨 전반적으로 민감 기능에 대한 재확인을 통해 안정성과 신뢰도를 확보함 |
|  <p>※ 계정 탈퇴 설정 화면</p> | <p>기능 설명</p> | <ul style="list-style-type: none"> 비밀번호 변경 기능은 <code>sendPasswordResetEmail()</code> 또는 <code>reauthenticateWithCredential()</code> 메서드를 통해 사용자 인증 후 처리되며, 이메일 기반 재설정 흐름을 제공함 계정 삭제 기능은 Firebase Authentication에서 계정 제거와 동시에 Firestore users 문서도 삭제되어 데이터 일관성을 유지함 파괴적 행동에 대해서는 2차 확인 팝업을 제공하여 실수로 인한 데이터 손실을 방지하고 사용자 신뢰도 향상을 도모함 |
| | <p>디자인 의도</p> | <ul style="list-style-type: none"> 사용자 자율 관리를 기반으로 계정 관련 기능을 설정 화면 내 명확하게 구분함 보안 관련 항목은 파괴적 행동에 대한 경고 절차를 통해 실수 방지와 사용자 데이터 보호를 강화하여 신뢰성을 높임 |

Re:fill 앱의 설정 화면 ④: 발주 이력

| | | |
|---|---------------|---|
|  <p>※ 발주 이력 화면</p> | UI 구성 | <ul style="list-style-type: none"> • 설정 내 발주 이력 항목을 통해 사용자는 자신의 발주 내역을 날짜별로 확인할 수 있음 • 각 발주 날짜 항목은 접이식 UI로 구성되어 있으며 클릭 시 상세 품목 및 수량 정보가 펼쳐지는 구조임 • 최근 발주부터 순차적으로 정렬되어 있으며 전체 발주 흐름을 시각적으로 확인할 수 있도록 디자인됨 |
|  <p>※ 발주 이력 화면</p> | UX 흐름 | <ul style="list-style-type: none"> • 사용자는 복잡한 설정 없이 발주 이력 메뉴를 클릭함으로써 즉시 발주 내역을 열람할 수 있으며 항목 클릭만으로 상세 내용이 토글되어 UX 흐름이 단순하고 직관적임 • 실시간으로 Firestore와 연동되며 과거 발주 내역도 자동 저장되어 시간 경과 후에도 기록 확인이 가능함 |
|  <p>※ 발주 이력 리스트 확장</p> | 기능 설명 | <ul style="list-style-type: none"> • 발주 이력 데이터는 Firestore의 /orders 컬렉션에 저장되어 있으며 storeId와 날짜 기준으로 필터링됨 • 각 문서에는 발주 일자, 품목 리스트, 수량 등의 정보가 포함되어 있으며 설정 화면에서는 이를 간결하게 리스트로 시각화함 • 리스트 항목 클릭 시 ExpansionTile 위젯을 통해 상세 정보가 출력되며 데이터는 최신순으로 정렬됨 |
| | 디자인 의도 | <ul style="list-style-type: none"> • 반복적 발주 업무에 있어 기록 확인 및 운영 추적의 중요성을 고려하여 사용자가 이력을 쉽게 열람할 수 있도록 단순하고 가독성 높은 UI로 설계함 • 한 화면에서 다수의 발주 이력을 확인할 수 있도록 스크롤 중심의 레이아웃으로 구성하였으며 정보량은 최소화하되 핵심 데이터를 빠르게 확인할 수 있도록 함 |

5. 프로젝트 결과물에 대한 기술

| 구분 | 기능정의 | 세부기능 설명 |
|---------------|--------------------|---|
| 재고 예측 및 발주 추천 | 재고 부족 예측 및 추천 발주 | Firebase Cloud Functions에서 내일의 날씨/공휴일을 기반으로 수요를 예측 후 부족 품목 리스트를 /recommendations/{storeid} 경로에 저장함. 예측 수요량이 최소 수량보다 높으면 발주 추천 품목으로 표시됨. |
| 발주 기능 연동 | 발주 목록과 수량 유지 | 예측/부족 화면에서 선택된 품목을 OrderScreen으로 전달, 수량 조정 가능함. 뒤로가기를 눌러도 수량이 유지되어 발주 편의성을 높임. |
| UI 권한 자동 분기 | 역할 기반 UI 렌더링 | Firestore의 사용자 role 값에 따라 앱 내 자동 분기됨. (예: 점주는 팀 관리 메뉴 노출, 매니저/직원은 미노출.) |
| 초대 코드 기반 팀 관리 | 팀원 초대 및 가입 시스템 | 점주가 생성한 6자리 난수 형태의 초대 코드를 직원이 입력하면 해당 매장에 자동 가입됨. 가입 전 기존 매장 소속 여부를 확인해 중복 방지 로직 적용함. |
| 실시간 채팅 | 매장별 그룹 채팅 | Firebase 기반 채팅 기능임. 각 매장 단위로 채팅방이 생성되며, 역할별 아이콘 및 이름이 표시됨. 읽음/미확인 메시지 수 표시가 포함됨. |
| 자동 발주 기능 | 예측 수요 기반 자동 발주 실행 | Firebase Cloud Functions와 flutter_background_service를 연동하여, 사용자가 설정한 시간에 맞춰 자동으로 예측 수요 기반 발주를 실행함. Firestore의 /recommendations/{storeid}/items에서 recommendedExtra > 0인 품목만 필터링하여 발주목록을 생성하고, /orders에 저장함과 동시에 /stocks/{storeid}/items의 수량을 자동으로 증가시킴. |
| 인증 및 보안 | Firebase 기반 인증 시스템 | Firebase Authentication으로 로그인 기능을 구현함. 이메일/비밀번호 및 Google 인증을 지원함. 민감 기능에는 재인증 처리를 적용함. |

6. 프로젝트 진행내용

가. 참여인원 및 담당 역할

| 연번 | 소속학과 | 성명 | 수행역할 분담내용 |
|----|------------------------|-----|--|
| 1 | AI소프트웨어학부 (컴퓨터공학전공) | 전유진 | Firebase 이메일/비밀번호 기반의 인증 연동, 아이디(이메일) 찾기 및 비밀번호 재설정 기능 구현, 발주 카테고리 필터링 기능, 재고 예측 화면 및 수량 추천 팝업 구현, 날씨 위젯 및 홈 UI용 DB 설계, 발주 로직 및 품목 DB(지점/프랜차이즈별) 구축, 장바구니 방식의 발주 페이지 및 재고 부족 연동 기능 구현, 실시간 채팅 기능(안 읽은 알림, 읽음 처리 등 포함), 사용자 권한에 따른 화면 분기, 앱 알림 기능, 초대 코드 기반 매장 가입 시스템 및 DB 설계, 회원가입 및 로그인 일부 기능 구현, 자동 발주 시간 DB 구현, 자동 발주 활성화 기능 포함 및 경고문구 삽입, 자동 발주 활성화 스위치 기능 구현, 채팅 공지사항 구현, 전반적인 오류 수정 및 화면 UI 개선, 회의록 작성, 1차 및 최종 발표, 논문 초안 작성, 전반적인 문서 작업 마무리 |
| 2 | AI소프트웨어학부 (컴퓨터공학전공) | 곽유나 | 앱 아이콘 및 스플래시 화면 제작, 홈 화면 UI(날씨, 재고 부족 알림, 캘린더 등) 디자인 및 UI 구현 담당, 회의록 작성, 주제 발표, 포스터 제작 |
| 3 | AI소프트웨어학부 (컴퓨터공학전공) | 김서영 | 전체 UI 구조 설계, 로그인/회원가입/발주/설정 등 주요 화면의 UI 구현 전반 담당, 앱 컬러감 전체 통일, 3차 발표 및 발표 자료(ppt) 제작, 회의록 작성, 포스터 제작 |
| 4 | AI소프트웨어학부 (컴퓨터공학전공) | 양진영 | Firebase 초기 연동 및 Google 로그인 구현, 회원가입/계정 찾기 기능 완성, 설정화면 전체 기능(로그아웃/비밀번호 변경/탈퇴/팀 관리/발주 이력 보기) 구현, 디렉토링 리팩토링 및 디자인 정리, 발주/재고 화면 세부 |

