

X세대 온라인 식료품 구매 거부감을 해소하기 위한 새로운 온라인 장보기 경험 설계

1인 프로젝트(2024.04 ~ 09)

작성자 : 심재문

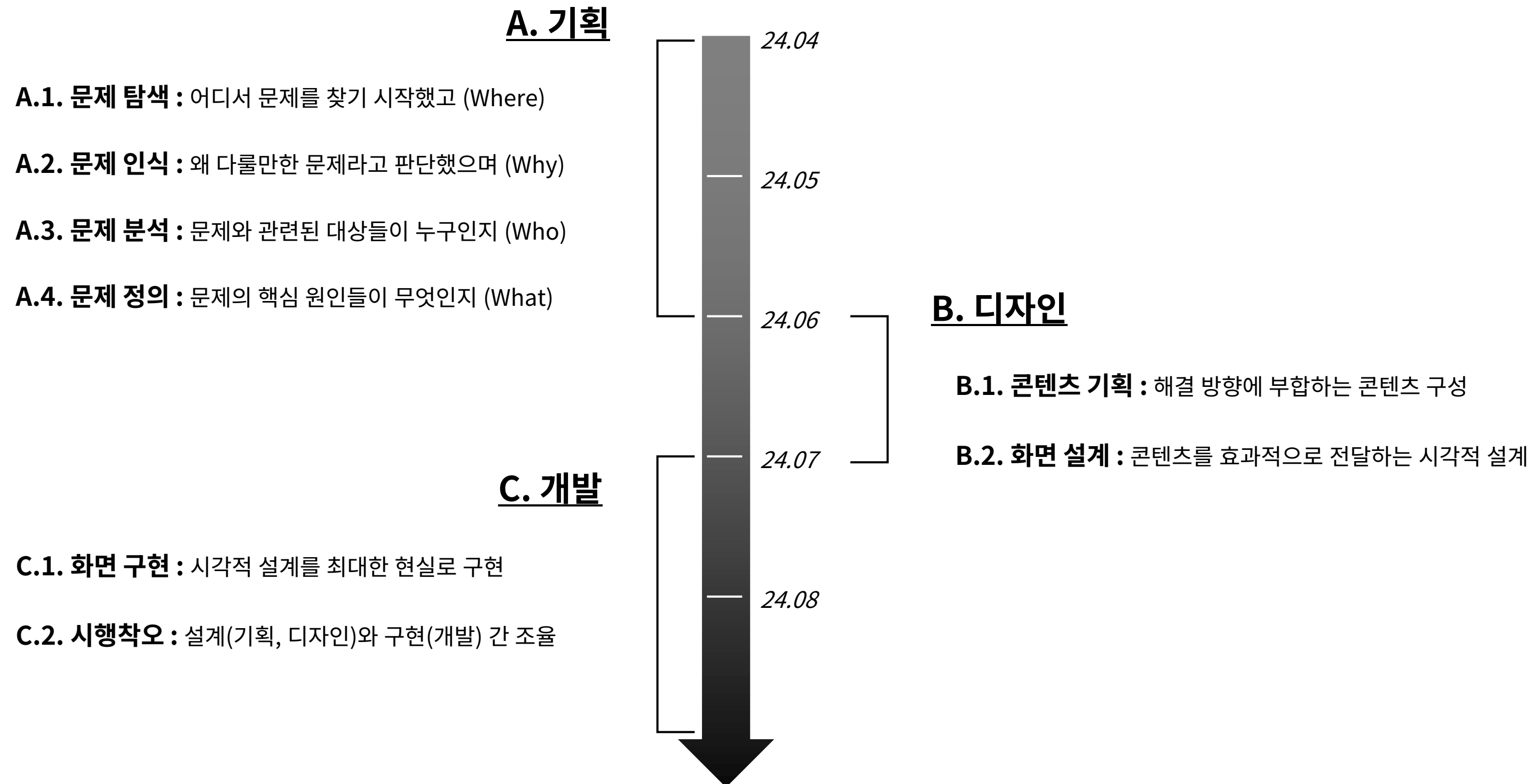
2024.10.26

[Github 보러가기 ↗](#)

개발 경험 [1인 프로젝트, 2024.04 ~ 09]

“ X세대 온라인 식료품 구매 거부감을 해소하기 위한, 새로운 온라인 장보기 경험 설계 ”

[배포된 사이트 보러 가기 ↗](#)



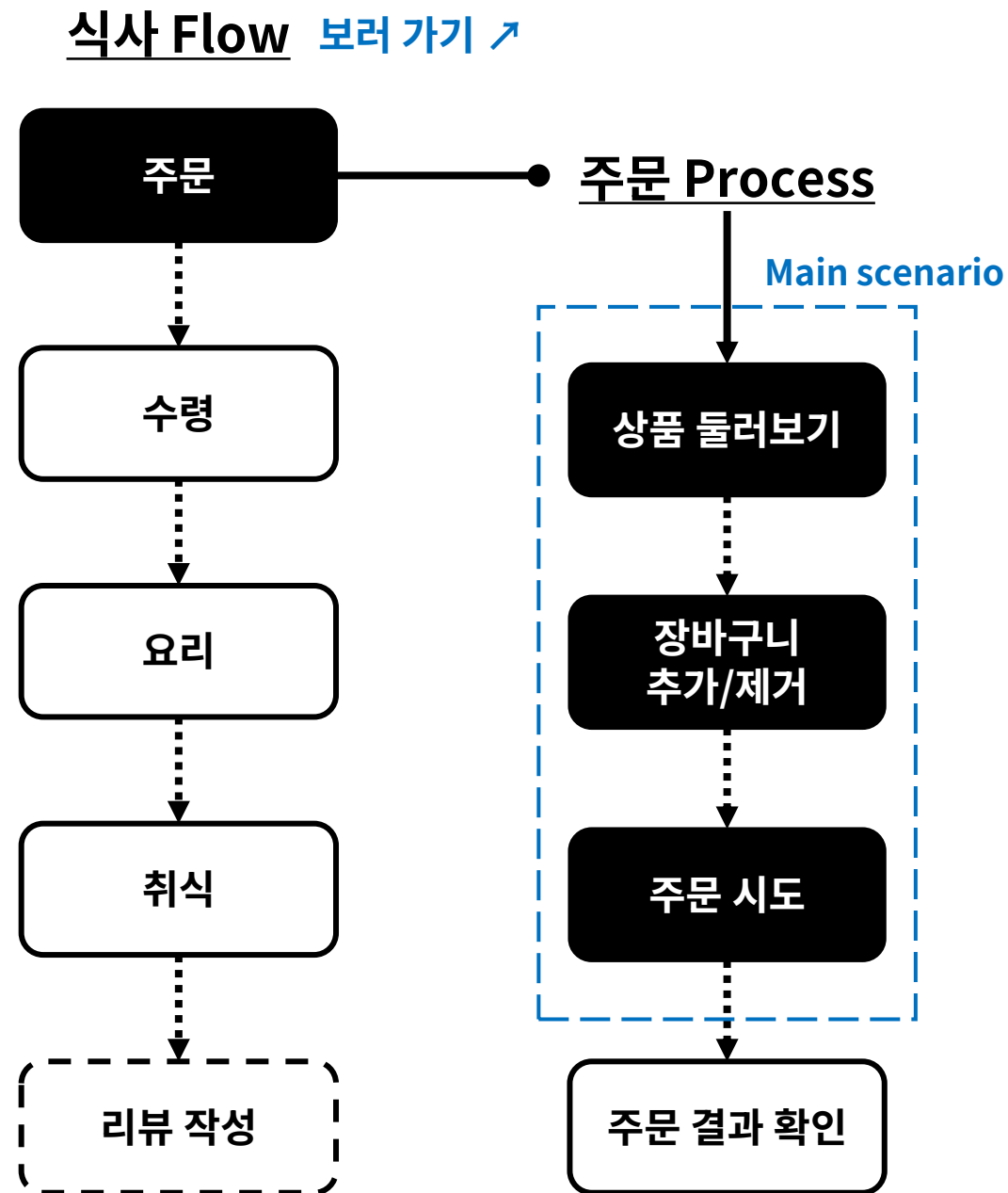
A. 기획

[식사 Flowchart 보러 가기 ↗](#)

[기존 서비스 분석 보러 가기 ↗](#)

A.1. 문제 탐색 : 어디서 문제를 찾기 시작했고 (Where)

탐색 방향 결정(식사 Flow) | 풀이 범위 좁히기(주문 Process)



A.2. 문제 인식 : 왜 다룰만한 문제라고 판단했으며 (Why)

- 단서 : X세대에 온라인 식료품 구매 진입 장벽 존재
- 배경 : 온라인 유통 시장의 꾸준한 성장세

온라인으로 식료품을 구매할 능력은 있지만,

◎ 온라인 주문 역량 보유[대상 : 식품 및 배달 음식]

- ‘혼자서 할 수 있다’ : 50대(75.3%), 60대(36.9%)

실제 구매 행동으로 이어지지 못하는 상황

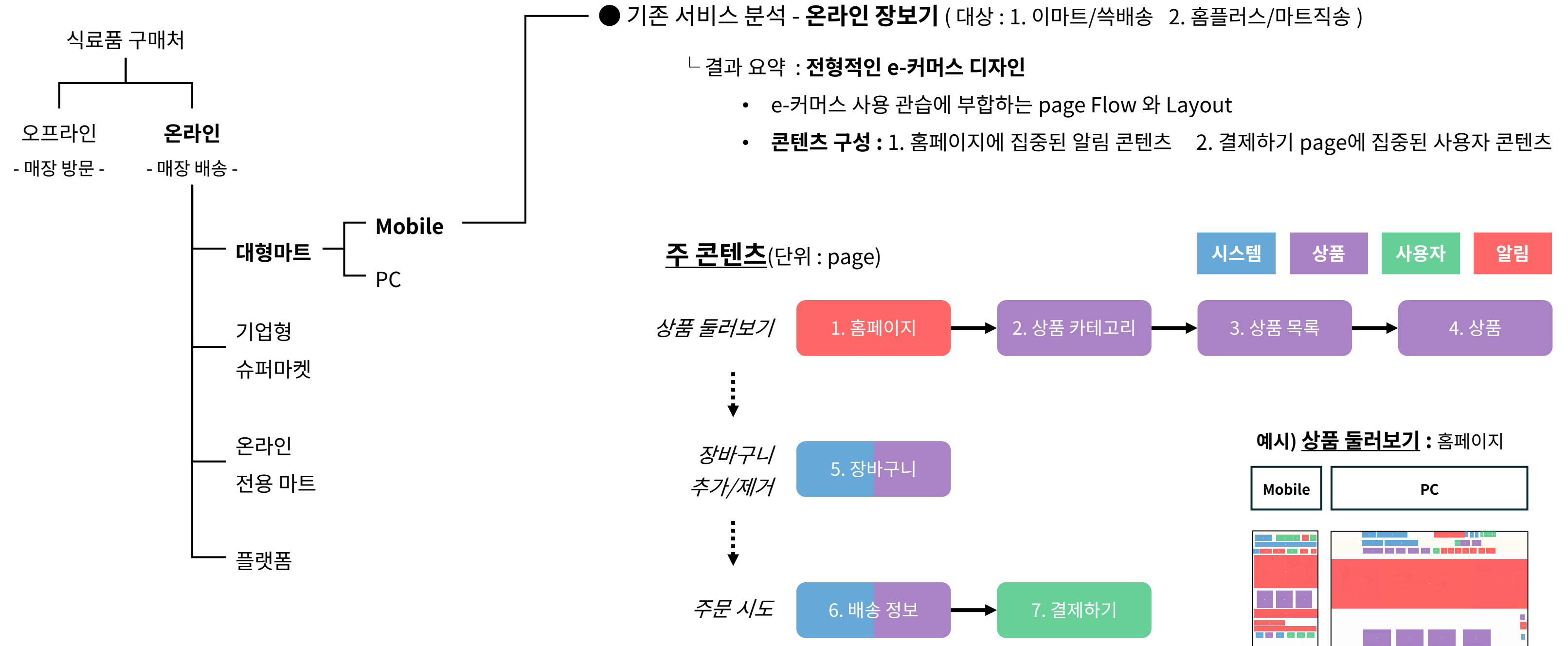
◎ X세대의 낮은 온라인 식료품 구매율

- 온라인 식료품 구매 비중 20% 이하

A.3. 문제 분석(1/2) : 문제와 관련된 대상들이 누구인지 (Who)

[기존 서비스 분석 보러 가기 ↗](#)

Market : 온라인 - 대형마트 - Mobile | **Service** : 온라인 장보기 | **User** : X세대(50대)



A.3. 문제 분석(2/2) : 문제와 관련된 대상들이 누구인지 (Who)

Market : 온라인 - 대형마트 - Mobile | Service : 온라인 장보기 | User : X세대(50대)

X세대를 데이터(1. 인구 통계 2. 신체 능력 3. 인지 능력 4. 행동 패턴)에 근거해 분석한 결과

서비스 접근성

인구 통계

- ✓ 높은 스마트폰 보유율
 - 모바일 서비스 이용 수단 보유

인지 능력

- ✓ 일반 국민 수준의 디지털 리터러시 보유
- ✓ [추측] 식재료/요리 지식 풍부

행동 패턴

- ✓ 스마트폰을 통해 주로 인터넷 이용

기회 요인

인구 통계

- ✓ 인터넷 이용자 중 가장 많은 비중 차지(잠재 고객)

신체 능력

- ✓ 체력 감소에 따른 온라인 장보기 니즈 강화

인지 능력

- ✓ 온라인 식료품 주문 역량 보유
- ✓ [추측] 오프라인 장보기 능숙

행동 패턴

- ✓ 소극적인 온라인 식료품 구매

수익성

인구 통계

- ✓ 넉넉한 가용 자본 보유

행동 패턴

- ✓ 전체 식사의 60% 이상을 직접 조리
 - 식재료 구매 니즈 항상 존재

A.4. 문제 정의 : 문제의 핵심 원인들이 무엇이며 (What)

원인 파악 : X세대가 온라인 식료품 구매 경험을 거부하는 원인 파악 | 해결 방향 도출 : X세대의 온라인 장보기 서비스 진입 장벽 낮추기

● 사용자 조사(미수행)

가설 1. 온라인 장보기 서비스를 구성하는 page 간 서로 다른 행동 패턴을 유발한다.

추정 원인 : page 구성 콘텐츠 유형 & 양

가설 2. 전형적인 e커머스 디자인은 X세대에게 온라인 장보기 거부감을 유발한다.

추정 원인 : 오프라인 장보기 능숙함(= 경험이 만든 관성)

- **피험자** : 50대 남/여 (온라인 장보기 주문을 1년에 5번 이하)
- **과업**(이마트 쓱배송) : 목표 상품들을 장바구니에 담고 주문 완료하기[page 기준]
 - ✓ 홈페이지 > 상품 카테고리 > 상품 목록 > 상품 > 장바구니 > 배송 정보 > 결제하기
- **데이터 수집 방법**
 - ✓ 과업 중 : 화면 녹화, 행동 관찰 카메라, 시선 추적 장치 & 설문지
 - ✓ 과업 종료 직후 : 설문지, 인터뷰
- **객관적 측정 지표**
 - ✓ 과업 완료 시간
 - ✓ 오류율
 - ✓ 시선 추적 데이터
- **주관적 측정 지표**
 - ✓ 설문지(사용성-ASQ, 서비스 이탈 욕구)
 - ✓ 행동 관찰(Think Aloud)
 - ✓ 인터뷰(반구조화) : 추정 원인 중심 질문

◎ **문제 원인 1.** 콘텐츠 과잉 공급(X세대가 한 번에 감당 가능한 콘텐츠 양 초과)

➤ **해결 방향** : page 구성 콘텐츠 최소화

설계 기조 1. 서비스 가치 제안에 필수적인 콘텐츠 중심 구성

설계 기조 2. 콘텐츠들을 복수 page / modal 로 적극 분할

◎ **문제 원인 2.** 장보기 방식 괴리감(익숙함이 유발한 낯설음)

➤ **해결 방향** : 익숙함(오프라인 장보기 경험)을 통해 낯설음(온라인 장보기 거부감) 해소

설계 기조 3. 오프라인 장보기 경험을 연상시키는 UI 디자인

기대 효과

1. 서비스 이용률 증가[AARRR] : 자기효능감 상승 (문제 원인 2.)

2. 서비스 이탈률 감소[AARRR] : 서비스 이탈 욕구 감소 (문제 원인 1.)

B. 디자인

[디자인 Foundation 보러 가기 ↗](#)

[화면 설계 보러 가기 ↗](#)

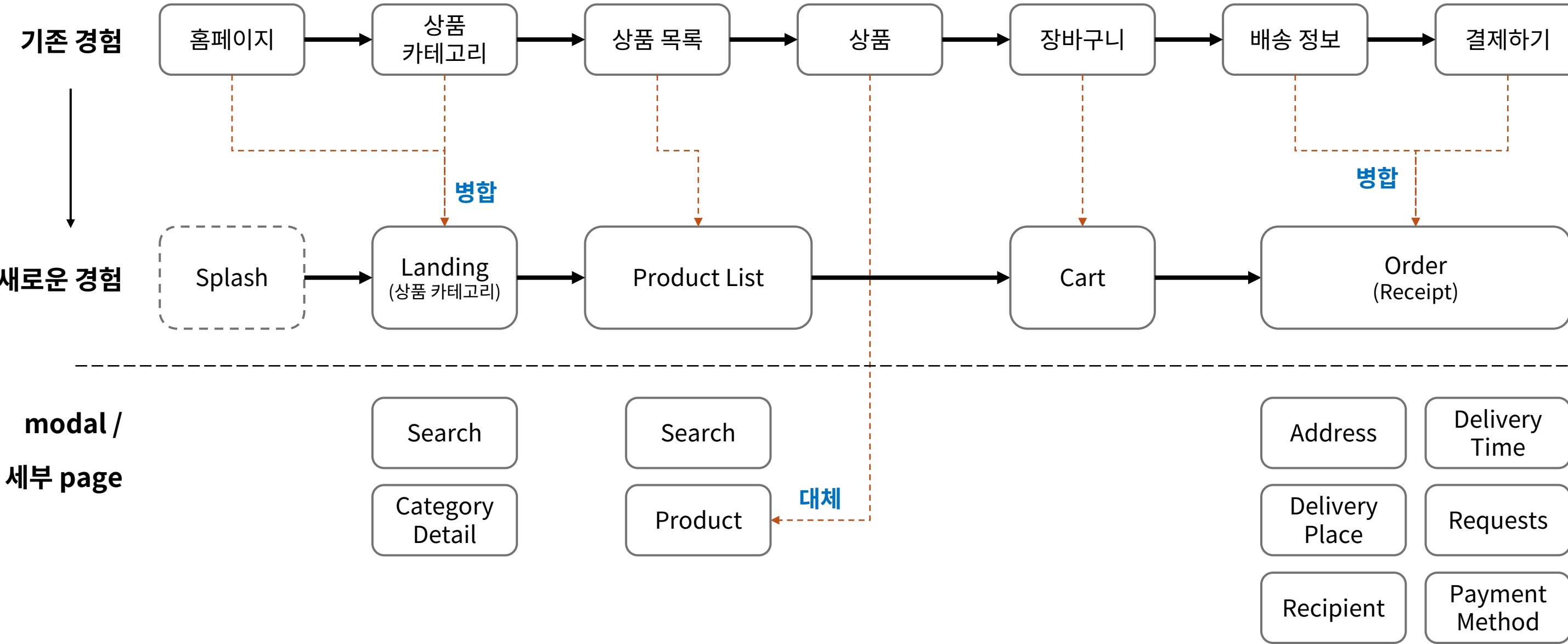
B.1. 콘텐츠 기획(1/2) : 해결 방향에 부합하는 콘텐츠 구성

- 설계 기조 1. 서비스 가치 제안에 필수적인 콘텐츠 중심 구성
- 설계 기조 2. 콘텐츠를 복수 page/modal 로 적극 분할
- 설계 기조 3. 오프라인 장보기 경험을 연상시키는 UI 디자인

page Flow 온라인 장보기 서비스

전체 page 수 감소(7 ⇒ 5) : page 병합 & modal 대체

Order page를 6개 세부 page로 분할



B.1. 콘텐츠 기획(2/2) : 해결 방향에 부합하는 콘텐츠 구성

설계 기조 1. 서비스 가치 제안에 필수적인 콘텐츠 중심 구성 설계 기조 2. 콘텐츠를 복수 page/modal 로 적극 분할 설계 기조 3. 오프라인 장보기 경험을 연상시키는 UI 디자인

콘텐츠 구성 main 태그 구성

모든 page 에서 프로모션 알림 콘텐츠를 배제

page flow가 진행될수록, 콘텐츠 양이 증가하도록 구성

Splash	Landing (상품 카테고리)	Product List	Cart	Order (Receipt)
<div>시스템</div> <div><ul style="list-style-type: none">애니메이션시장 입구로 들어가는 사람</div>	<div>시스템</div> <div><ul style="list-style-type: none">상품 카테고리 더보기</div> <div>사용자</div> <div><ul style="list-style-type: none">사용자 이름</div> <div>상품</div> <div><ul style="list-style-type: none">상품 카테고리 이름배송 점포 이름</div> <div>알림[프로모션 미포함]</div> <div><ul style="list-style-type: none">로그인 필요알림 도착</div>	<div>시스템</div> <div><ul style="list-style-type: none">상품 목록 더보기상품 목록 pagination상품 목록 Filter</div> <div>상품</div> <div><ul style="list-style-type: none">브랜드 이름상품 이미지상품명상품 가격 & 단위 당 가격리뷰 점수 & 리뷰 건수</div> <div>알림[프로모션 미포함]</div> <div><ul style="list-style-type: none">검색된 상품 개수</div>	<div>시스템</div> <div><ul style="list-style-type: none">배송지 변경품질 제거장바구니 담긴 상품 수량 변경장바구니 담긴 상품 제거장바구니 담긴 상품 더보기주문하기</div> <div>사용자</div> <div><ul style="list-style-type: none">배송지 이름 & 주소상품별 장바구니 담긴 개수</div> <div>상품</div> <div><ul style="list-style-type: none">배송 점포 이름상품명상품 가격</div> <div>알림[프로모션 미포함]</div> <div><ul style="list-style-type: none">가장 빠른 배송 가능 시간무료 배송 가능까지 남은 금액장바구니에 담긴 상품 개수 & 총액배송비장바구니 주문 총액</div>	<div>시스템</div> <div><ul style="list-style-type: none">영수증 항목별 변경하기영수증 더보기영수증 세부 내역 보기/숨기기+더보기결제하기</div> <div>사용자</div> <div><ul style="list-style-type: none">배송지 이름 & 주소배송 점포명 & 배송 희망 시간배송 위치 [+ 출입 방법]배송 요청 사항상품 받는 분 이름 & 전화번호결제 방법 [+ 사용할 카드 유형]상품별 주문 수량</div> <div>상품</div> <div><ul style="list-style-type: none">상품명 & 상품 가격</div> <div>알림[프로모션 미포함]</div> <div><ul style="list-style-type: none">부정확한 출입 방법 기재 시, 배송되는 위치최종 결제 금액[총액]배송비 & 주문 상품 개수 & 총액</div>

B.2. 화면 설계(1/3) : 콘텐츠를 효과적으로 전달하는 시각적 설계

■ 설계 기조 1. 서비스 가치 제안에 필수적인 콘텐츠 중심 구성
 ■ 설계 기조 2. 콘텐츠를 복수 page/modal 로 적극 분할
 ■ 설계 기조 3. 오프라인 장보기 경험을 연상시키는 UI 디자인

page Concept 시장을 구경하며 상품을 구매한 경험을 연상시키는 Object 구성



▣ 설계 기조 1. 서비스 가치 제안에 필수적인 콘텐츠 중심 구성 ▣ 설계 기조 2. 콘텐츠를 복수 page/modal 로 적극 분할 ▣ 설계 기조 3. 오프라인 장보기 경험을 연상시키는 UI 디자인

Foundation				
Environment		Base		etc.
<u>Grid</u>	<u>Spacing</u>	<u>Color</u>	<u>Typography</u>	<u>State</u>
<ul style="list-style-type: none">Mobile (360 ~ 639px)	<ul style="list-style-type: none">4px 단위	<ul style="list-style-type: none">전통시장 이미지 구성 색상	<ul style="list-style-type: none">Font Size ≥ 16px	<ul style="list-style-type: none">Enabled
<ul style="list-style-type: none">Tablet (640 ~ 1279px)	<u>Elevation</u> (그림자)	<u>Icon</u>	<u>Shape</u>	<ul style="list-style-type: none">Focus/Selected
<ul style="list-style-type: none">Desktop (1280px ~)	<ul style="list-style-type: none">Material 3 Design Kit	<ul style="list-style-type: none">Action, Alert, Navigation, Social, Domain ...	<ul style="list-style-type: none">Corner Radius : 4, 8, 16px	<ul style="list-style-type: none">Disabled
				<ul style="list-style-type: none">Error/Warning
				<ul style="list-style-type: none">...

Font Size 제한을 통한 콘텐츠 과잉 공급 방지

“ Font Size 를 최소 16px 이상으로 제한함으로써, 각 page 를 구성하는 콘텐츠 양을 간접적으로 조절 ”

B.2. 화면 설계(3/3) : 콘텐츠를 효과적으로 전달하는 시각적 설계

화면 설계 보러 가기 ↗

- ▣ 설계 기조 1. 서비스 가치 제안에 필수적인 콘텐츠 중심 구성
- ▣ 설계 기조 2. 콘텐츠를 복수 page/modal 로 적극 분할
- ▣ 설계 기조 3. 오프라인 장보기 경험을 연상시키는 UI 디자인

화면 설계 각 page 에 대한 1. 시각적 설계 2. 화면설계서 3. Data & State 를 작성



header	
1	<ul style="list-style-type: none">row1-column1 [App 이름]click/touch event : Landing-page로 이동(Self)
2	<ul style="list-style-type: none">row1-column2 [배송 예상 시간]가능한 가장 빠른 배송 시간 표시배송 점포 특정 방법<ul style="list-style-type: none">if 위치 정보 수집 가능<ul style="list-style-type: none">비로그인<ul style="list-style-type: none">단말기 위치 정보 기반 가까운 점포로그인<ul style="list-style-type: none">사용자 주소 기반 배송 권역 점포else 위치 정보 수집 불가능<ul style="list-style-type: none">'로그인이 필요합니다' 문장 표시
3	<ul style="list-style-type: none">row1-column3 [Person & ChatBubbleOutline icon]두 가지 toggle 관계 icon 배치 (교체 방법 : variant)Case 비로그인 : Person icon 표시<ul style="list-style-type: none">click/touch event : Login 페이지(미구현)Case 로그인 : ChatBubbleOutline icon 표시<ul style="list-style-type: none">click/touch event : Notification 페이지(미구현) 이동미확인 알림 존재시, 미확인 알림 개수 숫자 표시<ul style="list-style-type: none">미확인 알림 없는 경우, "On Background" 색상으로 0 숫자 표시최대 알림 표시 개수 = 99, 음수 발생시 error 유발
4	row2 : Search Box Component 참고
-	<ul style="list-style-type: none">TODOrow1-column2<ul style="list-style-type: none">click/touch event : 점포 배송 시간 정보 제공 page 이동해당 page 미구현비로그인-위치 정보 수집 불가시, event listener 미부착

main	
1	<ul style="list-style-type: none">main<ul style="list-style-type: none">click&move/Swipe event : Left - 이전, Right - 다음 상품 분류 목록 표시애니메이션 : 각 item 의 이미지(좌/우 slide) & 분류명(fade) 교체
2	<ul style="list-style-type: none">ProductCategory - item [상품 분류 투명 박스]click/touch event : Landing/CategoryDetail-modal 표시애니메이션 : 0.x 초 Hightlight 후 modal 표시
3	<ul style="list-style-type: none">Navigation - column1,3 [Left/Right LineArrow icon]이전/다음 상품 분류 목록이 없을 경우, 해당 방향 icon 미표시click/touch event : Left - 이전, Right - 다음 상품 분류 목록 표시애니메이션 : 각 item 의 이미지(좌/우 slide) & 분류명(fade) 교체
4	<ul style="list-style-type: none">Navigation - column2 [로그인 필요 문구 & My 일러스트]두 가지 toggle 관계 문구-일러스트 배치 (교체 방법 : variant 조작)Case 비로그인 : 로그인 필요 문구 표시<ul style="list-style-type: none">click/touch event : Login 페이지(미구현) 이동Case 로그인 : My 일러스트 표시<ul style="list-style-type: none">click/touch event : My 페이지(미구현) 이동애니메이션 : My 일러스트가 앞 줄아지는 골목으로 걸어가는 움직임알림 수신시, "알림이 왔습니다" modal 메세지 표시<ul style="list-style-type: none">click/touch event : Notification 페이지(미구현) 이동애니메이션 : 0.x 초 fade 표시 -> 1s 유지 -> 0.x fade 비표시
5	<ul style="list-style-type: none">StoreName<ul style="list-style-type: none">(주의)점포명 길이가 10자 초과[11자 이상] : 앞 9자 표시 + 말줄임표[...]e.g., 가평잡고들시장역사...점
-	<ul style="list-style-type: none">TODOProductCategory - item [상품 분류 투명 박스]상품 카테고리 이름이 영역을 범람할 경우 조치하기(PC) Hover event : 0.x 초 Hightlight 애니메이션

Data

- Static
 - 상품
 - 카테고리
 - 이름
 - 이미지
- Dynamic
 - LogIn/Out
 - 유무
 - user
 - 주소
 - 알림
 - 배송 점포 [특정 방법 : 주소, 기기 위치]
 - 점포명
 - 배송 가능 시간
 - 가장 빠른 배송 가능 시간

Data & State

State

- 상품 검색어 입력값
- page number

C. 개발

[컴포넌트 보러 가기\(Storybook\)](#) ↗

[배포된 사이트 보러 가기](#) ↗

C.1. 화면 구현(1/3) : 시각적 설계를 최대한 현실로 구현

컴포넌트를 뼈대(Presentational) 와 동작(Container) 으로 구분하는 설계 방식 채택

개발 환경 구축

1. Node.js(v20.15.0)-npm 최신화

2. 프레임워크 설치

- ✓ Next.js(TypeScript)
- ✓ Tailwind CSS(디자인 Foundation 주입)

3. Tool 설치

- ✓ ESLint & Prettier
- ✓ Husky & Lint-staged
- ✓ Storybook
- ✓ Vercel Postgres SDK(SQL)
- ✓ SWR

4. DB Seeding(Vercel – Postgres)

5. Design asset 추가

- ✓ Font(Noto Sans KR)
- ✓ 이미지
 - Icon
 - product category
 - product

구현

1. UI(Presentational)

- ✓ 컴포넌트 분류를 위해 Atomic Design 참고
- ✓ **컴포넌트 추출 기준** : 3번 이상 중복된 code

2. Feature(Container)

- ✓ 데이터 가져오기
- ✓ State 관리
- ✓ 애니메이션 구현(e.g., Carousel)

3. Test(미수행)

- ✓ Unit test(Jest)
- ✓ E2E test(Cypress)

Next.js 의 App Router 방식으로 구현

● 폴더 구조

❖ App

- page : 화면
- API : DB Seeding, 데이터 가져오기

❖ Library

- data : 타입 정의, fetch 함수 & mock 데이터
- UI : 구현 지원 함수 & UI 데이터

❖ Component

- UI(Presentational)
 - ✓ atom : 최소 표현 단위
 - ✓ molecule : (atom \geq 1)
 - ✓ organism : (molecule \geq 1) + (atom \geq 0)
- Feature(Container)
- modal : presentation + container

📁 App

- └─ 📁 page
 - | └─ 📁 Splash
 - | └─ 📁 Landing
 - | └─ 📁 ProductList
 - | └─ 📁 Cart
 - | └─ 📁 Order
- └─ 📁 API
 - | └─ 📁 seed
 - | └─ 📁 productCategoryGroups
 - | └─ 📁 cartProducts

📁 Library

- └─ 📁 data
 - | └─ 📄 data.d.ts
 - | └─ 📄 action.ts
 - | └─ 📁 dataset
 - | | └─ 📄 mockProductCategoryData.ts
 - | | └─ 📄 mockProductData.ts
 - | | └─ 📄 productCategoryData.ts
 - └─ 📁 UI
 - | └─ 📄 clipPathFrameGenerator.ts
 - | └─ 📁 dataset
 - | | └─ 📄 clipPathData.ts
 - | | └─ 📄 iconData.ts

📁 Component(UI)

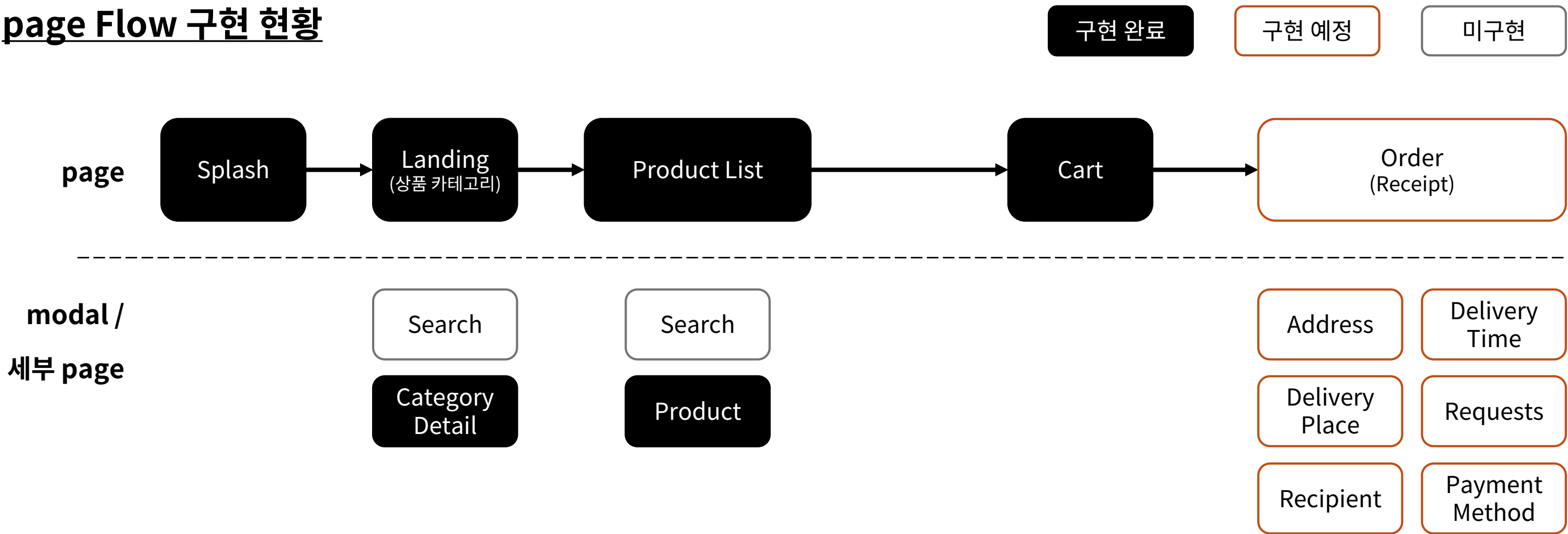
- └─ 📁 atoms
 - | └─ 📁 Icon
 - | └─ 📁 ProductCategoryBox
 - | └─ 📁 TextButton
 - | └─ 📁 ProductCard
 - | └─ 📁 SpeechBubble
 - | └─ 📁 Spinner
 - | └─ 📁 TextWithBorderButton
- └─ 📁 molecules
 - | └─ 📁 IconButton
 - | └─ 📁 IconButtonBack
 - | └─ 📁 IconWithBadgeButton
 - | └─ 📁 IconWithTextButton
 - | └─ 📁 SearchBox
 - | └─ 📁 CategoryDetailCarousel
 - | └─ 📁 ProductCategoryGroup
 - | └─ 📁 ProductListStand
 - | └─ 📁 SeeMoreButton
- └─ 📁 organisms
 - | └─ 📁 Footer
 - | └─ 📁 Header
 - | └─ 📁 ProductCardGroup

📁 Component(...)

- └─ 📁 containers
 - | └─ 📁 CartReceiptContainer
 - | └─ 📁 ProductCategoryGroup...
 - | └─ 📁 ProductListContainer
- └─ 📁 modals
 - | └─ 📁 ProductCardModal

미수행 작업 : 최적화(이미지, 렌더링)

page Flow 구현 현황



C.2. 시행착오(1/3) : 설계(기획, 디자인)와 구현(개발) 간 조율

UI : 개발과 디자인(Figma) 간 구현 방식 차이



디자인

평행사변형 Layer 에 외곽선만 그리면,
손쉽게 속이 보이는 박스를 만들 수 있을 거예요.

예상보다 더 많은 구현 시간 소요



개발

쉽지 않은데요? 예상보다 오래 걸리겠어요.

◎ 시행착오 흐름(상품 카테고리 박스 외곽선 표현)

① (실패) CSS “clip-path” 속성이 적용된 요소에 “border” 속성 적용

- clip-path 가 box model 을 변경하지 못 해, 외곽선 표현 실패

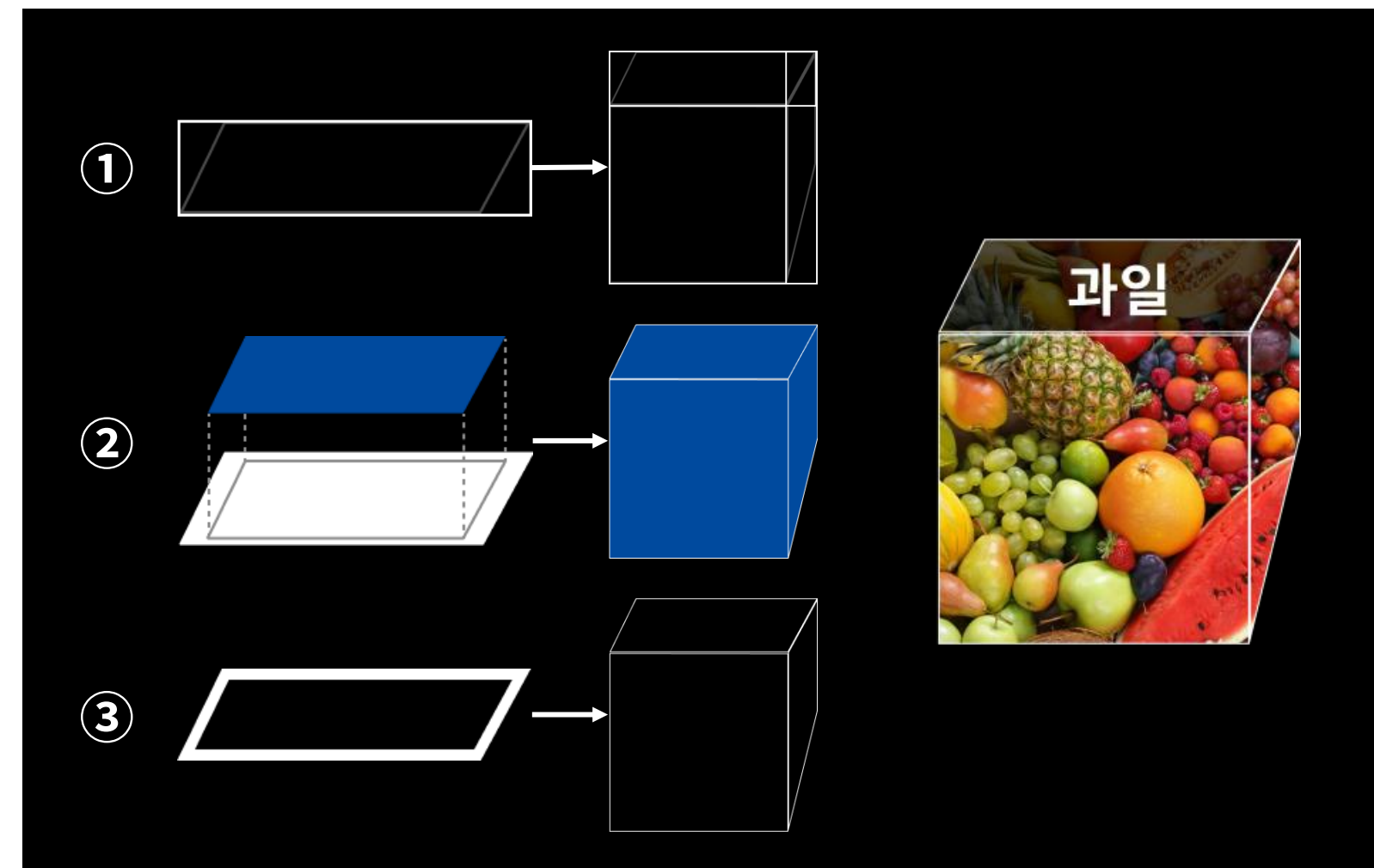
② (실패) 가로-세로 비율이 동일한 서로 다른 크기 layer 겹치기

- 외곽선 표현에는 성공했지만, 박스 안 이미지 표현 불가

③ (성공) CSS “clip-path” 를 액자(Frame) 형태로 구현

- 외곽선 내부/외부 각각에 대한 좌표들을 계산해 사용
 - ✓ 사선 외곽선 좌표를 구하기 위해,

“직선 간 거리 공식”, “두 직선의 교점” 을 사용함



C.2. 시행착오(2/3) : 설계(기획, 디자인)와 구현(개발) 간 조율

Feature(State 관리 & 애니메이션 구현) : 개발 중 디자인 변경

◎ 시행착오 흐름(상품 카드 pagination 표현)

(기존) Numbering pagination

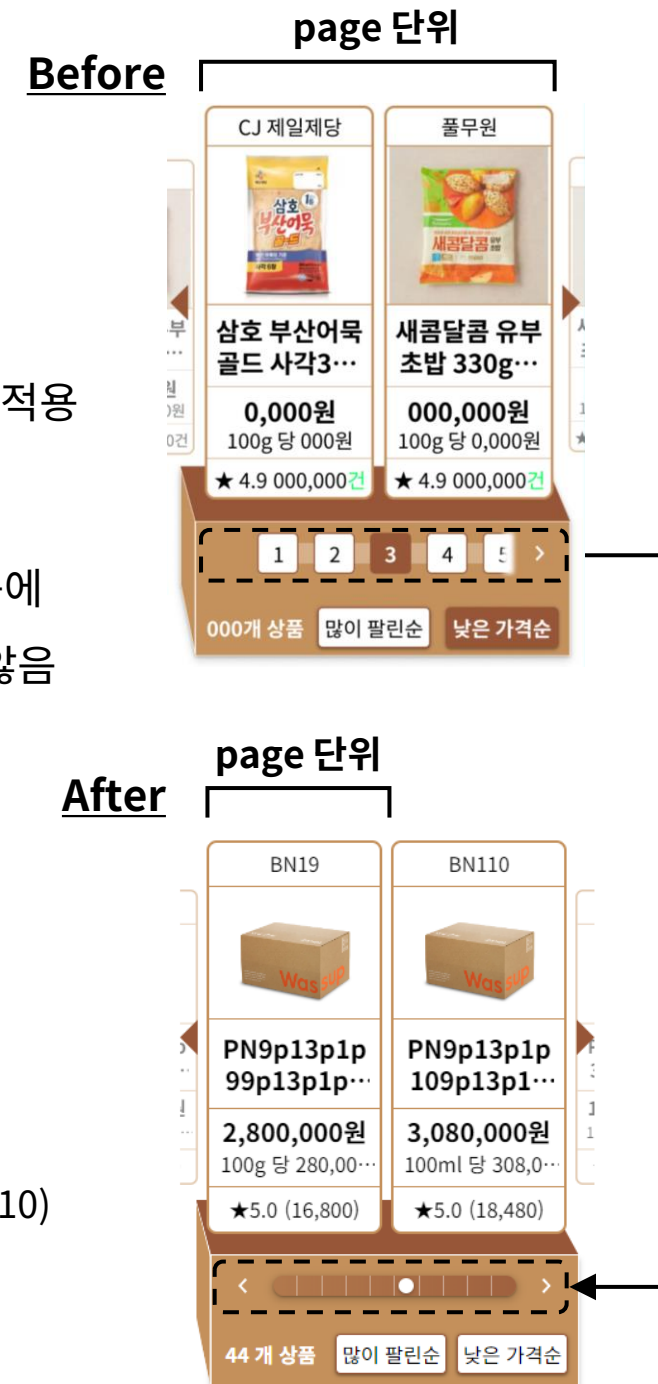
- page 당 2개 상품 카드 할당
- page 변경 시, 2개의 상품 카드가 이동하는 수평 Scroll 애니메이션 적용

❖ 문제

- ✓ 상품 카드가 매대에 진입/이탈할 때, 단일 카드가 아닌 카드 묶음에 CSS scale 애니메이션이 적용되어 변화가 명료하게 인식되지 않음

(변경) Slider pagination

- page 당 1개 상품 카드 할당
 - page 변경 시, CSS scale 애니메이션을 단일 상품 카드에 적용
- ❖ (보류) 문제
- ✓ pagination 표시 영역 분할을 최대 10개로 제한(e.g., 상품 수 > 10)



기획

상품 카드가 매대에 진입/이탈할 때, 카드 묶음이 이동하는 수평 Scroll 애니메이션이 부자연스럽네요.

상품 카드가 묶음이 아닌 하나씩 움직이면 자연스러울 것 같아요.



디자인



기획

상품 카드가 하나씩 움직이도록 변경하면 page number 와 상품 카드가 1:1 대응 관계가 돼서, page 숫자를 표시하는 게 부자연스러울 것 같은데...

그럼 상품 카드가 하나씩 움직이도록 수정하면서, pagination 표시 방식도 Slider 로 바꿔보죠.



개발

page 당 1개 상품 카드가 할당되도록 수정하는 건 문제가 없는데, 상품 카드가 10개 이상이면 Slider 영역을 몇 개로 분할해야 하나요?

C.2. 시행착오(3/3) : 설계(기획, 디자인)와 구현(개발) 간 조율

사용성 개선 : 경험의 연속성 보존



기획

더보기 버튼을 눌러 이동해 다음 상품 카테고리들을 보던 중,
장바구니 page에 잠깐 갔다 돌아왔는데 왜 상품 카테고리
위치가 초기화 된거죠?

pagination 상태 보존 요청 대응

상품 카테고리 pagination 상태를 보존하지 않는
방식으로 구현되어 있어서 그런 건데, 해당 상태를
URL로 관리하면 해당 문제를 해결할 수 있어요.



개발

◎ 시행착오 흐름(상품 카테고리 pagination 상태 관리)

(기존) Local State 로 상태 관리

- 지역 상태를 업데이트해, 더보기 버튼 event 에 대응
- ❖ 문제 : page 이탈 후 복귀 시, 지역 상태 불일치(=상태 초기화)

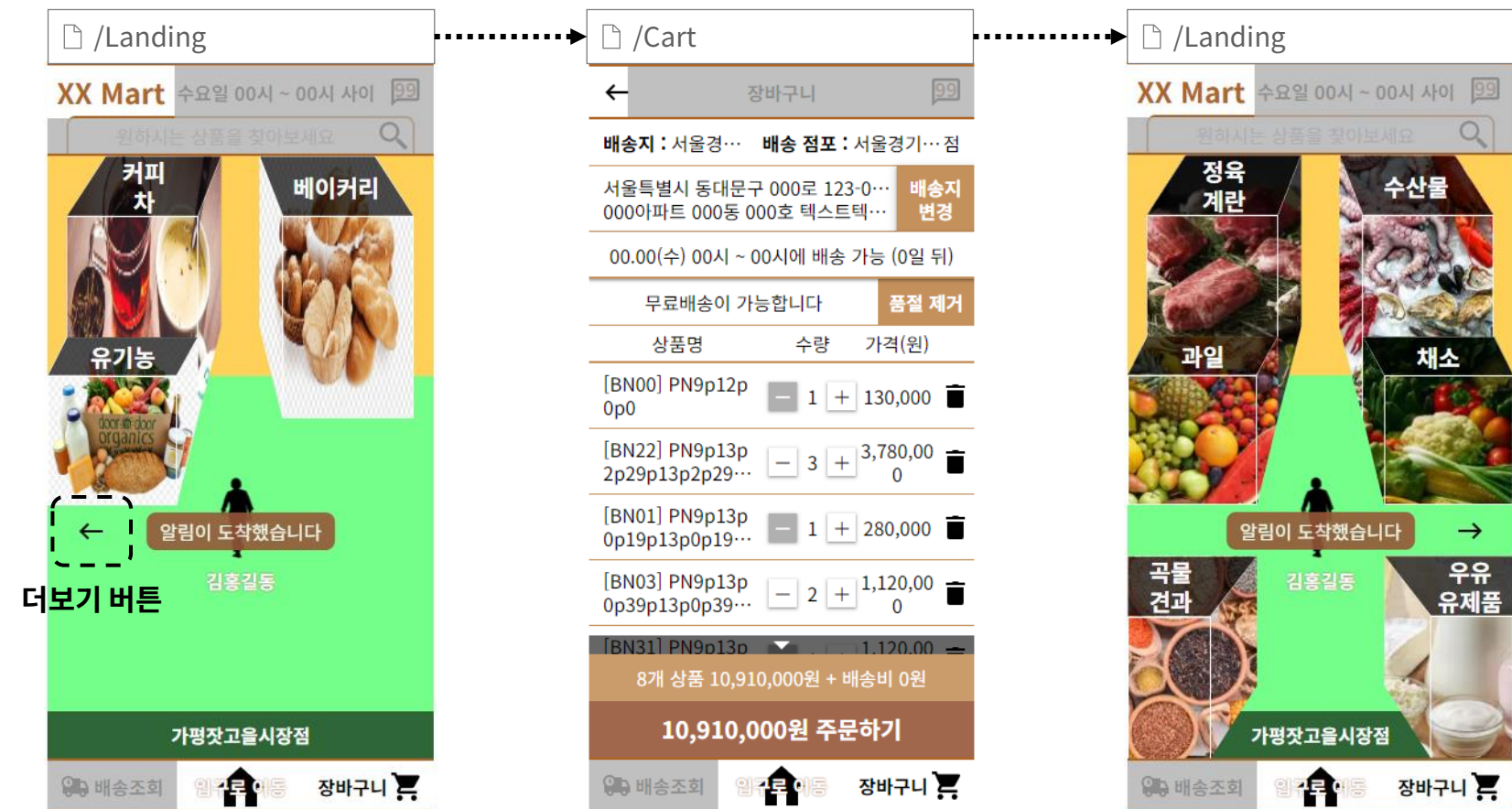
(변경) URL 로 상태 관리

- Query Parameter 를 이용해, pagination 상태 관리
 - ✓ 표기 : “ ?pageNumber={value} ”
- URL에 상태를 보존해, “뒤로가기/앞으로 가기” 시 상태 기억

After



Before



감사합니다