

Class[DaDaMall]

▼ Service Planning

PRD Product Required Document

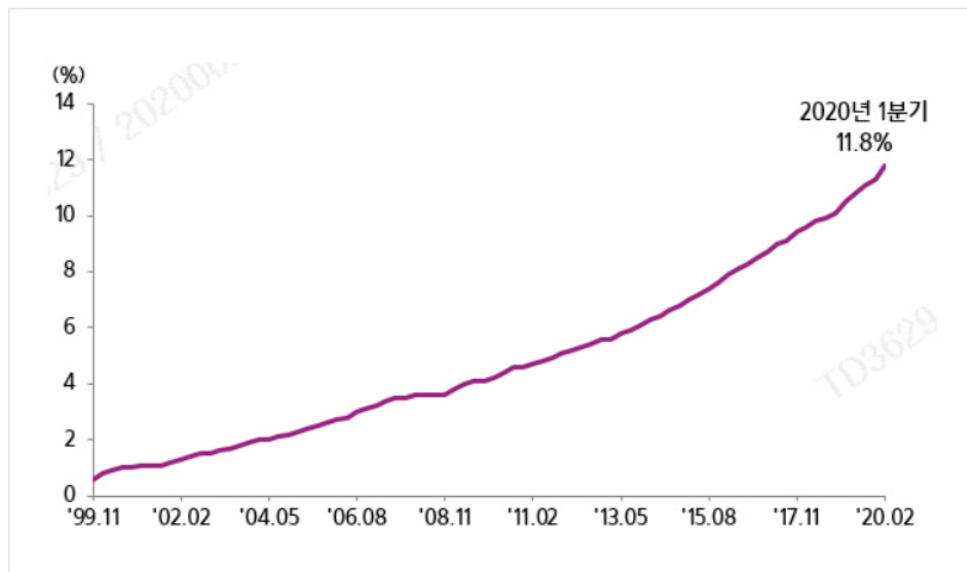
- 우리가 만들고자 하는 것은 무엇인가!
- **왜** 이 제품을 만들어야 하는가!
- 제품 개발을 통해 달성하고자 하는 목표는 무엇인가!
- 목표 달성을 위한 검증 지표는 무엇인가! (정성적이 아닌 정량적)

출처 : 이커머스 시장(E-Commerce)의 성장세는 계속될 전망 (samsungpop.com)

온라인 상거래 시장은 지속해서 성장 전망

이미 우리는 비교적 오래 전부터 이커머스를 통해 다양한 소비를 즐겨왔음에도 불구하고, 전체 소매 판매에서 이커머스를 통한 소매 판매 비중은 아직도 낮은 수준입니다. 전세계 최고의 소비 강국인 미국 역시 이커머스를 통한 소비는 전체 소매 판매액에 11.8%(2020년 1분기 기준)에 불과합니다. 향후 이커머스 시장의 추가적인 성장이 예상되는 이유입니다. 더욱이 최근 코로나19 사태는 이커머스 시장의 성장세를 보다 촉진시키는 매개체가 될 것으로 보입니다. 각국 정부의 이동제한명령으로 온라인을 통한 소매판매가 크게 확대되었던 점에 주목할 필요가 있습니다. 이 과정에서 이미 이커머스를 이용했던 소비자들은 더 많이 이커머스를 이용하게 되었고 과거 온라인 물품 구매 경험이 없던 소비자 또한 이커머스를 경험하게 되었습니다. 특히, 신규 이용자들의 경험은 단순 1회성 경험을 넘어 추가적인 경험으로 확장될 가능성이 높습니다.

[미국 전체 소매판매액 중 이커머스를 통한 판매액 비중 추이]



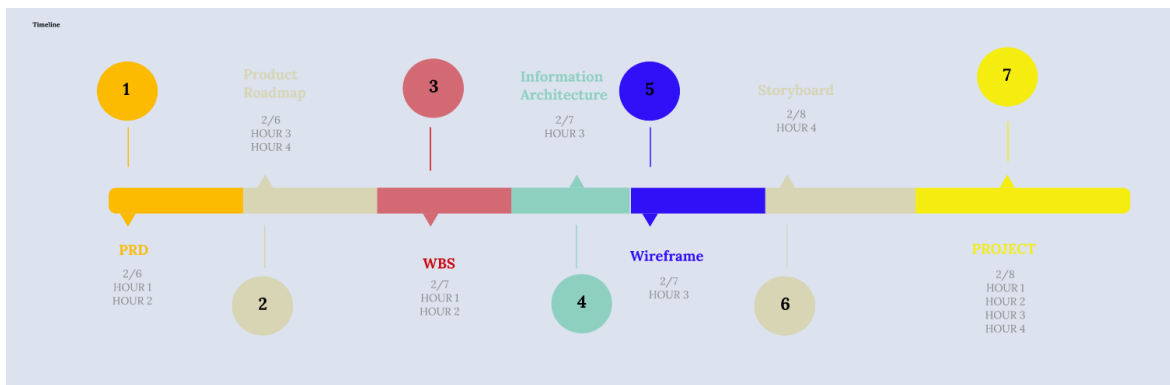
자료: 미국 통계국, Refinitiv, 삼성증권 포트폴리오전략팀

과거의 산업혁명을 거쳐 현대에 이르기까지 소비문화와 경제체제는 많은 변화를 거쳐왔다. 이러한 소비문화의 변화는 2019년 코로나19로 인해 더 가파른 변화를 가졌다. 그로 인해 비대면 시스템과 이커머스, 그리고 쇼핑 물 시장은 점점 더 거대해질 전망이다. 또한 카카오톡, 인스타그램 등의 SNS와 유튜브 등의 플랫폼들도 이러한 이커머스 시스템과 접목을 시키는 상황이기때 이커머스와 쇼핑물 시장의 성장은 필연이 되었다. 이러한 패러다임의 변화로 인해 DaDaMall과 같은 개인 쇼핑물 페이지의 수요는 점점 더 증가할 것이다.

구분	ID	요청사항	필요기능	우선순위	비고
ADMIN	GT-A-001	회원가입 허가	운영자의 허가 후 회원 가입 승인	1	
ADMIN	GT-A-002	회원삭제	운영자에 의한 회원 계정 삭제 기능	2	
ADMIN	GT-A-003	회원관리	운영자가 회원목록을 볼 수 있는 시스템	2	
ADMIN	GT-A-004	상품등록	운영자가 상품을 등록하는 기능	3	
ADMIN	GT-A-005	상품삭제	운영자가 상품을 삭제하는 기능	3	
ADMIN	GT-A-006	상품수정	운영자가 상품을 수정하는 기능	3	
ADMIN	GT-A-007	회원정보 보기	운영자가 회원 상세정보를 확인하	2	

			는 기능		
USER	GT-U-001	회원가입	쇼핑몰 회원가입 기능	1	
USER	GT-U-002	로그아웃	쇼핑몰 로그아웃 기능	2	
USER	GT-U-003	로그인	쇼핑몰 로그인 기능	2	
USER	GT-U-004	본인 정보 수정	자신의 정보를 수정하는 기능	3	
USER	GT-U-005	본인 정보 삭제	자신의 피드를 삭제하는 기능	3	
USER	GT-U-006	본인 정보 보기	자신의 정보를 확인하는 기능	3	
USER	GT-U-007	장바구니	상품을 장바구니에 등록하는 기능	3	
USER	GT-U-008	상품 상세보기	상품 상세정보를 확인하는 기능	3	
USER	GT-U-009	배송정보 입력	상품 배송정보를 등록하는 기능	3	
USER	GT-U-010	상품구매	상품을 구매하는 기능	3	

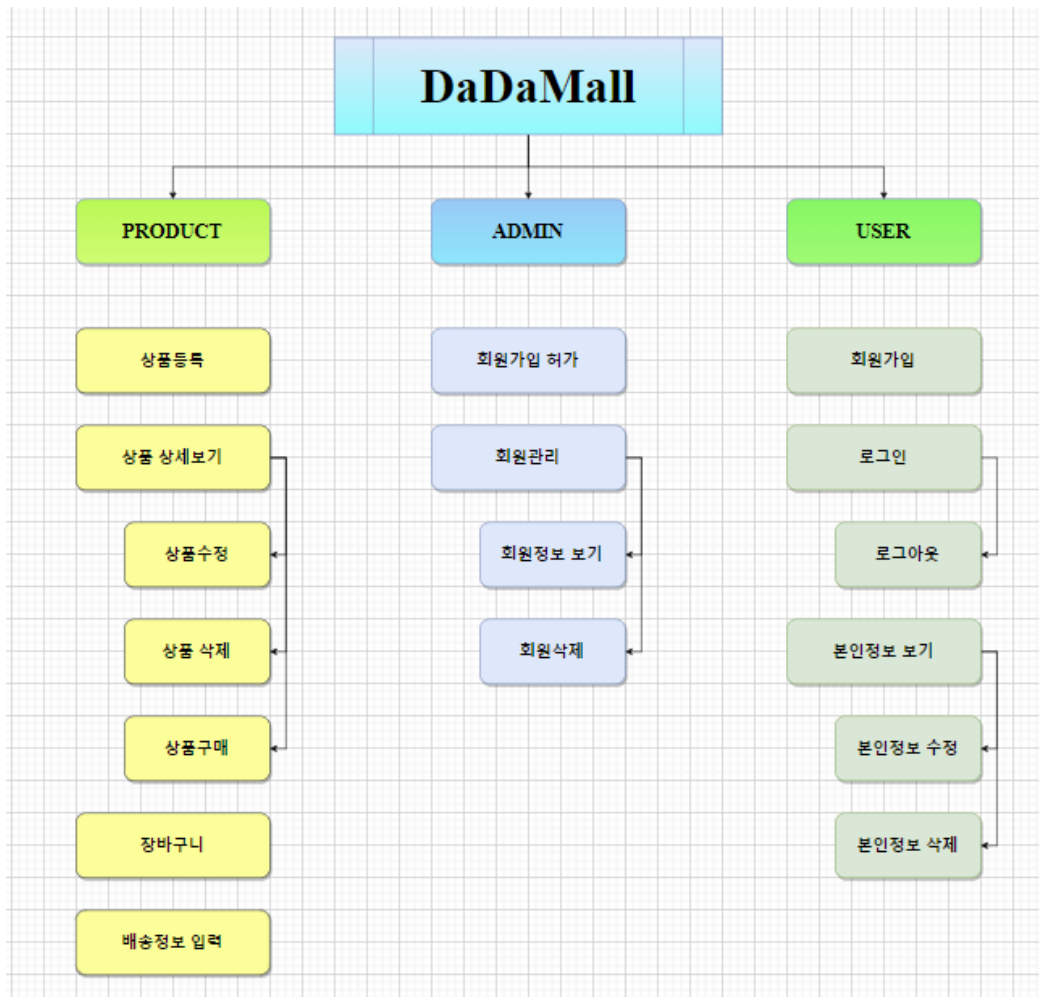
PR Product Roadmap



WBS Work Breakdown Structure

SKIP

IA Information Architecture



WF Wireframe

회원 정보		작성일자		작성일자	페이지
회원 ID	회원명	비밀번호	아이디	비밀번호	5
<div>DaDaMall</div> <div>회원가입</div> <div>아이디(Email)</div> <div>패스워드</div> <div>패스워드(확인) 1</div> <div>이름</div> <div>회원가입 신청 2</div> <div>계속</div>		<div>DaDaMall</div> <div>가입 신청 리스트</div> <div>아이디 이름</div> <div>aaa@naver.com 홍길동 3 승인 거부 4</div> <div>bbb@naver.com 김하나 승인 거부</div> <div>ccc@naver.com 김두리 승인 거부</div>		<div>Description</div> <div>1 패스워드 입력 오류 확인 (JS)</div> <div>2 회원가입 신청 후 담당자의 승인 후 사용가능</div> <div>3 "승인"을 클릭하면 회원가입 승인 후 DB이동</div> <div>4 "거부"를 클릭하면 회원가입 거부 후 DB삭제</div> <div>5</div>	

SB Storyboard

▼ Python

조건문 if

조건문은 특정 조건을 만족할 때 실행되는 코드를 제어한다.
개발하다보면 여러가지 상황을 처리해야 하는 경우가 발생한다.
이때 조건문을 통해 실행되거나 또는 실행되지 않는 코드를 정의할 수 있다.

if 의 기본 문법

JAVA 기본 if문

```
if(조건식) {  
    만족하는 경우;  
}
```

Python 기본 if문

```
if 조건식 :  
    조건을 만족하는 경우 실행코드 # 조건식 뒤에 : 콜론을 잊지 말자. 중괄호는 사용하지 않는다.  
    # 파이썬은 들여쓰기에 매우 민감하다.
```

조건식에는 항상 언제나 최종값이 true 와 false 만 올 수 있다.
다시 한번 강조하지만 만약 들여쓰기를 하지 않으면 오류가 발생된다.

```
x = 11  
if x == 10 :  
    print(x)  
  
a = int(input('정수를 입력하세요 > '))  
  
if a == 5:  
    print(f'입력한 값은 {a}입니다.')
```

조건식을 여러개 취급해야하는 경우는 가능하면 중첩/병렬 if 문을 사용하지 말자.

```
x = 15  
if x >= 10:  
    print('x는 10 이상입니다.')  
    if x <= 10:  
        print('x는 10 이하입니다.')  
        if x == 15:  
            print('x는 15입니다.')  
  
a = int(input('숫자를 입력하세요 > '))  
  
if a > 5:  
    print('입력값이 5보다 큼니다.')  
else :  
    print('입력값이 5보다 작거나 같습니다.')  
  
a = int(input('숫자를 입력하세요 > '))  
  
if a > 5:  
    print('입력값이 5보다 큼니다.')  
elif a == 5:  
    print('입력값이 5입니다.')  
else :  
    print('입력값이 5보다 작습니다.')
```

```

no = int(input('1부터 5 사이의 값을 입력하세요 > '))

if no < 5 and no >=4 : # 양쪽 조건을 동시에 만족
    print('입력값은 4입니다.')
elif no < 4 and no >=3 : # 양쪽 조건을 동시에 만족
    print('입력값은 3입니다.')
elif no < 3 and no >=2 : # 양쪽 조건을 동시에 만족
    print('입력값은 2입니다.')
elif no < 2 and no >=1 : # 양쪽 조건을 동시에 만족
    print('입력값은 1입니다.')
else :
    print('입력값은 5입니다.')

```

월을 입력하세요 > 해당 계절을 출력하세요.
 3,4,5 봄 6,7,8 여름 9,10,11 가을 12,1,2 겨울

```

a = int(input('월을 입력하세요 > '))

if a > 2 and a <= 5:
    print('봄')
elif a > 5 and a <= 8:
    print('여름')
elif a > 8 and a <= 11:
    print('가을')
else:
    print('겨울')

```

`if statement` 조건 구문

정수 하나를 입력 받아서 그 수가 홀수 인지 짝수 인지를 알려주는 프로그램을 만드세요.

```
a = int(input('정수 입력 > '))
```

```
if a%2 == 0:
    print('짝수')
else:
    print('홀수')
```

```
a, b = 100, 200
```

조건부 표현식 `if expression` 조건 표현식

자바의 삼항 연산자와는 달리 `if` 를 사용하여 표현한다.

```
a = int(input('정수 입력 > '))
```

```
answer = '짝수' if a%2 == 0 else '홀수' # True 출력값 if 조건식 else False 출력값
```

```
print(f'입력하신 {a}는 {answer}입니다.')
```

`if` 와 `in` 연산자

성적을 입력 받아서 59점 까지는 F학점

69점 까지는 D학점

79점 까지는 C학점

89점 까지는 B학점

100점 까지는 A학점을

'당신의 점수 {}는 {}학점입니다.'

```
input = int(input('성적 입력 > '))
```

```
if input <= 59 :
    rank = 'F'
elif input <= 69 :
    rank = 'D'
elif input <= 79 :
    rank = 'C'
elif input <= 89 :
    rank = 'B'
else:
    rank = 'A'
```

```
print(f"당신의 점수 {input}는 {rank}학점입니다.")
```


반복문 While, For

While 은 반복횟수가 불명확할 때 주로 사용되고
For 문은 반복횟수가 명확할 때 주로 사용된다.

예를 들어 hi를 10번 반복하여 출력하고자 한다.

```
print('hi')
print('hi')
print('hi')
print('hi')
print('hi')
print('hi')
print('hi')
print('hi')
print('hi')
print('hi')
```

10번의 경우에는 위와 같이 10번 반복할 수 있다. 하지만 1000번이라면 무모하다.

```
for i in range(10):
    print('hi')
```

range 는 인수 3개가 기본이다. range(start, stop, step)

start 생략가능(기본값 : 0에서 시작)

step 생략가능(기본값 : 하나씩 증가)

```
list1 = [10, 30, 20, 50]
```

```
for i in list1 :
    print(i)
```

```
list1 = [10, 30, 20, 50]
```

```
for i in range(4) :
    print(list1[i])
```

리스트와 for 문을 이용하여 1~100까지의 합 구하기

```
numbers = list(range(1,101))
```

```
sum = 0
```

```
for number in numbers:
    sum += number
```

```
print("the sum is", sum)
```

이중 for 문

```
for x in range(2, 10):  
    for y in range(1, 10):  
        print(x, "*", y, "=", x*y)
```

단별로 옆으로 한줄로 찍기

```
for x in range(2, 10):  
    for y in range(1, 10):  
        print(x, "*", y, "=", x*y, end=' \t')  
    print(' ')
```

while

while 조건식:
 수행시킬 구문

```
while True:      # 조건에 True. 무한 루트  
    print('h')
```

```
i = 10  
while i > 5:  
    print('hi')  
    i -= 1 # 파이썬은 자바처럼 i ++ 즉 증감 연산자는 사용하지 않는다. i = i + 1
```

```

import random

temp_no = 0
while temp_no != 3: # 3이 나올때까지 반복하여 난수 발생
    temp_no = random.randint(1, 6) # 끝 숫자 6도 포함된다. 미만이 아닌 이하.

    print(f'발생된 난수는 {temp_no}입니다.')

난수 발생 정리

import random # random 모듈 import 후 사용

print(random.random()) # 0에서 1 미만 실수. float 으로 생성

print(random.randint(1, 10)) # 1에서 10 이하의 정수 하나 생성

print(random.randrange(0, 10, 2)) # 0에서 10 미만 중 2의 배수로 난수 발생

print(random.choice(['봄', '여름', '가을', '겨울'])) # 리스트의 값들 중 하나를 임의로 선택

print(random.sample(range(1, 46), 6)) # range 범위 중 임의의 갯수 추출

list1 = [1, 2, 3, 4, 5]

random.shuffle(list1)

print(list1)

```

```
# break continue
```

```
i = 0
```

```
while i < 100:  
    i += 1  
    if i%2 == 0:  
        print(i)
```

break 강제로 while 문 빠져나가기

continue continue 하단의 내용은 건너뛰고 다시 반복문으로 들어가기

```
i = 0
```

```
while i < 10:  
    i += 1  
    if i % 2 == 0:  
        print(i)
```

```
i = 0
```

```
while i < 10:  
    i += 1  
    if i % 2 == 0: continue  
    print(i)
```

```
i = 0
```

```
while True:  
    i += 1  
    if i == 10: break  
    print(i)
```