

중년들의 패션친구 “중패친”

2013312262 권영재

2015311777 최민진

2016311589 오정호

2012312989 정정환

2016313475 오현지

2016313354 위응주

목차

1. Preface	
1.1 Objective.....	7
1.2 Readership	7
a. User Requirement readership	7
b. System requirement readership.....	7
1.3 Document Structure.....	8
a. Preface.....	8
b. Introduction.....	8
c. Glossary.....	8
d. User Requirement definition.....	8
e. System Architecture.....	9
f. System Requirement specification.....	9
g. System models.....	9
h. System Evoltuion.....	9
i. Appendices.....	9
j. Index.....	10
2. Introduction.....	12
2.1 Objective.....	12
2.2 Needs.....	12
2.3 중년들의 패션친구.....	15
3. Glossary.....	18
4. User Requirement Definition.....	20
4.1 Objective.....	20

4.2	Functional Requirements.....	20
a.	Sign up.....	20
b.	Log in.....	20
c.	Explore user's preference.....	21
d.	View new clothes.....	21
e.	Configure My page.....	22
f.	Recommend Friends.....	22
g.	Add Friends.....	22
h.	Go to Friends' Page.....	22
i.	Search Friend.....	22
j.	Browse friends' style with user's preferred tag.....	23
4.3	Non-Functional Requirements.....	23
a.	Product Requirements.....	23
b.	Organization Requirements.....	24
c.	External Requirements.....	25
5.	System Architecture.....	26
5.1.	Objective.....	26
5.2	Explore Preference System.....	26
5.3	My Page System.....	28
5.4	Friends Tab System.....	29
5.5	Browse Friends' Style System.....	30
5.6	View New Clothes System.....	31
5.7	New Clothes Crawl System.....	32
5.8	Friends' Page System.....	33

6. System Requirement Specification.....	34
6.1 Objective.....	34
6.2 Functional Requirements.....	34
a. Login With KaKao Function.....	34
b. Sign up Function (회원가입).....	35
c. Sign in Function (로그인).....	36
d. Explore Preference Function.....	37
e. Configure Style Page Function	38
f. Friends' tab Function	39
g. Add Friend Function.....	40
h. Find a Friend Function.....	40
i. New Clothes Crawl Function	41
j. Browse Friends' style Function (관심게시글).....	42
k. Write Comment Function (댓글작성).....	42
l. View New Clothes Function (새로운 옷 탐색).....	43
m. Update Preference Function (취향업데이트).....	44
6.3 Non-Functional Requirements.....	45
a. Product Requirements.....	45
b. Organization Requirements.....	46
c. External Requirements.....	47
6.4 Scenario	47
a. 사용자 사용 시나리오.....	47
a.1 회원가입 시나리오.....	47
a.2 취향분석 시나리오.....	49

a.3 View new Clothes 시나리오.....	51
a.4 자신의 페이지에 포스팅하기 시나리오.....	53
a.5 알립 tab 사용하기 시나리오.....	53
a.6 친구 추가하기.....	54
a.7 친구 게시물에 comment 남기기.....	56
b. 시스템 상 시나리오	57
b.1 취향분석 시나리오.....	57
b.2 크롤링된 이미지 분석 시나리오.....	59
b.3 친구추천 시나리오.....	60
7. System Models.....	61
7.1 Objectives.....	61
7.2 Context Models.....	61
a. Context Model.....	61
b. Process Diagram.....	61
7.3 Interaction Model.....	62
a. Use case Models.....	62
b. Tabular description for each use case.....	63
c. Sequence Diagram.....	71
7.4 Structural Models.....	74
a. User System class Diagram.....	74
7.5 Behavioral Models.....	75
a. Data – Driven Modeling	76.
b. Event – Driven Modeling.....	76
8. System Evolutions.....	79

8.1 Objective.....	79
8.2 User Requirement Evolutions.....	79
a. 구매로 이어질 수 있는 시스템 구축.....	79
b. 개별 스타일 마다 친구 추천.....	80
c. 비슷한 스타일의 모델, 배우의 정보 포함.....	81
d. 소통 기능 강화.....	82
8.3 Hardware Evolutions.....	83
a. FX mirror.....	83
9. Appendices.....	85
9.1 Obejctives.....	85
9.2 Database Architecture.....	85
9.3 Transfer Deep Learning Model.....	86
9.4 Cosine similarity.....	88
10. Index.	90

1. Preface

1.1. Objective

Preface 에서는 본 문서의 예상되는 독자들과 문서의 전반적인 구조, 그리고 각 부분의 역할에 대하여 제시한다. 그리고 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들과 그에 대한 근거들을 서술한다.

1.2. Readership

본 문서는 크게 User requirements 와 System requirements 두 부분으로 구성되어 있다. User requirements 는 사용자의 관점에서 요구사항을 간략히 명시한 것이고, System requirements 는 시스템이 제공하는 기능을 더욱 자세하게 설명한 것이다.

A. User requirements readership

User requirements 의 주요 독자는 해당 시스템의 고객들이기 때문에 시스템이 어떻게 구현되는지 등에 관한 사안에 관심이 없다. 이를 고려하여 전문지식이 없는 독자들을 위해 자연어, 다이어그램 등과 같은 방법으로 시스템이 어떤 기능을 제공하는지를 설명한다. 고객들 외에도 Client managers, system end-users, client engineers, contractor managers, system architects 가 user requirements 의 독자가 될 수 있다.

B. System requirements readership

System requirements 의 주요 독자는 해당 소프트웨어 개발자들이다. 그렇기 때문에 System requirements 에서는 해당 시스템의 특정 기능이 어떻게 구현되어야 하는지에 대해 자세하게 설명되어 있다. Software Developer 외에도 client engineers, system architects, 경우에 따라서는 System end-users 까지 주요 독자가 될 수 있다.

1.3. Document Structure

이 문서는 총 10 개의 부분으로 구성되어 있다. Preface, Introduction, Glossary, User Requirements Definition, System Architecture, System Requirements Specification, System Models, System Evolution, Appendices, Index 로 구성된다. 각 부분의 역할은 다음과 같다.

A. Preface

Preface 에서는 본 문서의 예상 독자가 누구인지 밝히고, 문서의 구조, 각 챕터의 역할 등에 대해 소개한다. 또, Version History 와 그 변경사유, 변경사항에 대해 설명하였다.

B. Introduction

Introduction 에서는 해당 시스템이 어떤 니즈(Needs)를 반영하였는지를 설명한다. 간략한 시스템의 기능을 설명하고 시스템이 다른 시스템들과 어떻게 동작하는지를 설명한다.

C. Glossary

Glossary 에서는 본 문서에서 사용되는 기술 용어가 무엇인지 설명한다. 이 때, 독자의 경험이나 전문지식 등이 어떠할지를 추정해서는 안되고, 가능한 한 모든 용어에 대해 서술해야 한다.

D. User requirements definition

User Requirements Definition 에서는 사용자에게 제공되는 서비스에 대해서 설명한다. 비기능적인 시스템 요구사항도 서술한다. 자연어와 다이어그램과 같은 방법들로 설명 되어있다. 시스템은 여기에 서술된 요구사항들을 반드시 충족할 것이다. User requirements definition 에서는 사용자에게 제공되는 서비스에 대해 설명한다. 이 챕터에 서는 또한 Non-functional system requirements 역시 서술된다. 이 때, 설명은 자연어, 다이어그램 또는 독자들이 이해할 수 있는 다른 개념들이 사용될 수 있다. 제품이나 프로세스가 반드시 따라야 하는 표준이 있다면 반드시 서술되어야 한다.

E. System architecture

System Architecture에서는 목표 시스템 architecture에 대한 high-level에서의 개요를 제시한다. 또, 시스템 모듈 사이에서 기능들의 분포가 어떠 한지를 보여준다.

F. System requirements specification

System Requirements Specification에서는 functional requirements 또는 non-functional requirements에 대해 더욱 자세히 설명한다. 필요하다면, non-functional requirements에 더욱 자세히 설명한다. 다른 시스템에 대한 인터페이스 역시 이 챕터에서 정의된다.

G. System models

System Models에서는 시스템 컴포넌트, 시스템 그리고 시스템 환경 사이의 관계를 보여준다. 가능한 그래픽 모델로는 object models, data-flow models, semantic data models 등이 있다.

H. System evolution

System Evolution에서는 시스템이 세우고 있는 주요한 가정들에 대해 설명한다. 또, 시스템에 일어날 수 있는 예상되는 변화들, 가령, 하드웨어의 발전, 사용자의 니즈 변화 등에 대해 설명한다. 이 챕터는 차후 발생할 시스템의 설계 변경을 피할 수 있도록 돕기 때문에 시스템 설계자들에게 도움이 될 것이다.

I. Appendices

Appendices에서는 개발되는 시스템과 관련된 자세하고 구체적인 정보를 제공한다. 이 프로젝트에서 사용하는 시스템이나 기술 중 특징이라 할 수 있는 것들을 좀 더 자세하게 설명하고, 그런 기술들이 어떻게 적용되는지를 기술한 부분이다.

J. Index

Index 에서는 주요 용어, 다이어그램, 기능에 대한 인덱스 등이 포함된다.

1.4. Version of the Document

A. Version Format

버전 번호는 major.minor[.maintenance]로 구성되며. 문서의 버전은 0.1 부터 시작한다.

B. Version Management Policy

요구사항 명세서를 수정할 때 마다 버전을 업데이트한다. 다만 변경 간의 간격이 1 시간 이내일 때에는 버전 번호를 업데이트하지 않고, 하나의 업데이트로 간주한다. 이미 완성된 파트를 변경할 때에는 Minor number 를 변경하며, 새로운 부분을 추가하거나 문서의 구성이 예전에 비해 괄목할 변화가 있을 경우 Major number 를 변경한다.

C. Version Update History

Version Modified Date Explanation

0.1	2018.10.13	요구사항명세서 문서 10 개의 장에 대한 문서 목차 작성.
1.0	2018.10.14	User Requirement, system Requirements 초안 작성.
2.0	2018.10.15	System Architecture, Scenario 초안 작성.
3.0	2018.10.16	Introduction 작성
3.1	2018.10.17	User Requirement, system Requirements, System Architecture, Scenario 수정사항 반영
4.0	2018.10.24	System Model , Appendices, System Evolution 작성
4.1	2018.10.28	Index 작성 최종완료

Table1. version table

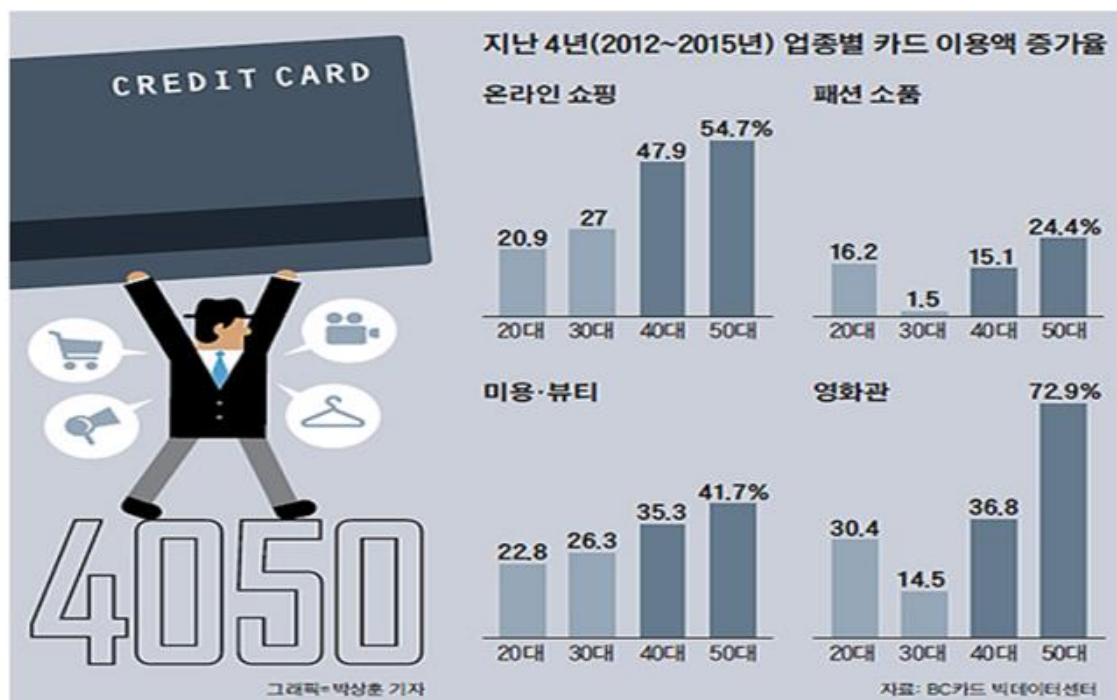
2. Introduction

2.1. Objective

Introduction 에서는 시스템의 필요성과 해당 시스템이 어떠한 Needs 를 반영하고 있는지를 설명한다. 또한 시스템의 기능과 다른 시스템과 어떻게 상호작용하는지 간략하게 설명한다. 또한 시스템이 어떻게 시스템의 목적을 성취하는지 설명한다.

2.2. Needs

최근 들어 중년들의 패션에 대한 관심이 증가하고 있다. 평균 수명이 증가하고, 건강한 노후를 보내는 사람이 많아지면서 중년층은 외모와 패션 나이에서 많은 노년층과 확실한 선 긋기를 하고 싶어하는 욕구가 커지고 있기 때문이다. 과거 한국 사회에서는 40·50 대는 그동안 '직장과 가정에서 가족을 위해 희생하고, 자신을 가꾸는 데는 관심 없는 세대'라는 이미지가 강했다. 하지만 최근 20·30 대 못지않게 피부와 미용 등에 관심이 많고, 영화·뮤지컬 등의 공연 관람을 즐기며, 스마트폰으로 수시로 모바일 쇼핑을 하는 '꽃중년'이 출현하고 있는 것이다.



23 일 BC 카드 빅데이터 센터에 따르면 지난 4 년간(2012~2015 년) 40·50 대는 피부·미용, 온라인 쇼핑, 영화·공연·전시 등의 문화생활, 패션 소품 등 이른바 젊은 소비를 상징하는 여러 업종에서 20·30 대보다 훨씬 높은 카드 사용 액의 증가율을 보였다. 위의 사진과 같이 40·50 대는 패션 소품 구매도 3 년 전보다 20%가량 많이 한 것으로 조사됐는데, 같은 기간 30 대가 1.5% 늘어난 것과 비교하면 증가율이 10 배 이상 높다. 영화나 뮤지컬·연극 등의 공연 관람 시 카드 이용액은 40 대가 지난 3 년간 36.8%, 50 대가 73% 증가해 20 대(30.4%)와 30 대(14.5%)를 압도했다. 이러한 현상에 힘입어 중년을 타겟팅을 한 쇼핑몰이 생겨나고 있다. 예를 들어 중년 남성을 타겟팅 한 주인(<http://www.juinn.co.kr/>)과 중년 여성을 타겟팅 한 마담 4060(<http://www.madam4060.com/>)이 있다.



Madam4060



이들은 중년들의 꾸미고 싶어하는 욕구를 타겟팅을 하여 쇼핑몰을 만들었다.

하지만 이와 같이 중년들을 위한 쇼핑몰이 생기고 중년들의 패션에 대한 관심 증가에도 중년들을 위한 패션 커뮤니티는 없다. 패션에 대한 관심이 증가하게 되면 상세한 리뷰를 보고싶기도 하고 옷을 더 저렴하게 사고 싶어하기 마련이다. 또한, 옷에 대한 정보도 얻고 싶고 자신의 패션을 자랑하거나 패션에 대한 평가를 듣고 싶어한다. 하지만 이러한 욕구를 풀어줄 곳이 중년들에게는 없다. 또한, 패션에 대해 잘 모르는 사람들은 어떠한 옷을 사야 할지 어떠한 색상이 어울릴지 모른다. 검색을 하고 싶어도 정보가 없고

자신과 비슷한 나이대의 사람들이 어떻게 입는지 알 수가 없다. 또한 자신의 패션 취향을 공유하고 그 친구가 입는 옷을 보고 그 옷이 멋있다면 사고 싶어 할 것이다. 이러한 역할을 패션 커뮤니티에서 하고 있다. 하지만 중년들에게는 이러한 대화를 할 공간이 없다.

사람이 관심이 가는 분야가 있다면 조금 더 알고 싶고 다른 사람과 공유를 하고 싶어 하기 마련이다. 이러한 욕구를 20·30 대들은 커뮤니티를 하거나 친구들과 대화를 하면서 풀어나간다. 하지만 중년들은 이러한 욕구를 해소할 곳이 없다. 커뮤니티는 너무 생소하고 친구들과 이야기하기는 어렵기 때문이다. 왜냐하면 가정이 생기고 일에 치여 살다가 보면 친구의 수는 줄어들고 남아있는 친구들이 같은 관심사를 가지지 않을 수 있기 때문이다. 그리고 패션에 관해서가 아니더라도 조금 더 가벼운 대화를 나눌 수 있는 친구를 원하게 된다.

제 목 : 40대 아줌마....늘 혼자인 일상생활....

모히또 | 조회수 : 30,795

하루종일 걸쳐 오는 전화한통 없고...문자 오는것도 없고....

카톡대문글도 매일 똑같고.....그날이 그날이고.....

하루하루 너무 무료한 일상이네요....

일은 하고있는데....있을때만 하는 상황이라...

요즘같은 비수기에는 집에 있는 날이 더 많아요....

아침에 식구를 다 보내고....커피한잔 마시고....청소는 기분따라 하고....

운동도 하고 장보고.....

모두 혼자네요....

이처럼 중년 여성, 중년 남성들 모두 외로움을 느낀다. 그런데 지금까지의 SNS 사이트들의 주 소비층은 10~30 대의 젊은 연령층이다. 중년들만을 targeting 하거나, 그들만의 이야기를 할 수 있는 커뮤니티 공간이 부재한다. Facebook 이나 Instagram 그리고 트위터 등을 중년들이 많이 사용하지 않는 이유는 그들이 그들의 일상과 관련된 글들을 올렸을 때, 주 소비층이 자신들의 자식 세대이므로, 자신들의 일상생활, 힘든 점, 느낀 점 등을 올렸을 때, 자식 세대가 그런 것을 본다고 생각했을 때 느껴지는 사회적 시선, 압박 등의 이유로 SNS 활동을 제대로 하지 못한다. 뿐만 아니라, 젊은 층들에게는

SNS 활동이 어렵지 않고, 어렵더라도, 그런 것들을 이겨낼 수 있는 유인들이 존재하지만, 중년층들에게는 그렇지 않다

이러한 문제점들을 **중년들의 패션친구**를 통해서 해결할 수 있다. 중년들의 패션 친구는 다음과 같은 중년들에게 적합하다.

1. 멋진 패션을 하고 싶은데 돈과 시간이 부족한 사람
2. 패션에 대해 알아가고 싶은 사람
3. 급히 옷을 사야 하는데 옷에 대한 정보가 부족한 사람
4. 비슷한 옷을 더 싸게 사고 싶은 사람
5. 멋진 자신의 패션을 자랑하고 싶은 사람
6. 비슷한 취향의 친구의 패션을 보고 싶은 사람
7. 자신의 패션을 평가받고 나아지고 싶은 사람
8. 패션에 대해 대화를 할 친구가 필요한 사람
9. 외로워서 가벼운 대화를 할 친구가 필요한 사람

이러한 유형에 해당하는 중년들의 needs 를 만족시킬 수 있는 어플리케이션을 개발하는 것이 우리의 목표이다.

2.3. 중년들의 패션친구

중년들의 패션친구는 핸드폰 어플리케이션을 기반으로 하여 총 크게 4 가지 기능을 제공한다.

1. 자신만의 타임라인
2. 추천 친구 & 친구 리스트
3. 친구들의 게시글 보기
4. 자신의 스타일을 정하기



자신만의 타임라인은 자신의 스타일을 정하고 자신만의 게시글을 올려 타임라인을 꾸미는 것이다. 자신이 고른 옷을 이 곳에서 확인을 할 수 있다. 또한, 자신의 게시글에 해시태그를 남겨 쉽게 검색이 가능하고 자신의 게시글에 친구들이 코멘트를 남길 수 있어 소통의 창구가 될 것이다.

추천 친구 & 친구 리스트는 자신의 스타일과 비슷한 친구들을 찾아서 추천을 해주고 친구들을 간편하게 추가할 수 있는 기능을 제공한다. 물론 지인의 ID를 알고 있다면 쉽게 검색을 해서 찾을 수도 있다. 또한, 이미 친구가 된 사람들의 타임라인을 방문해 구경을 할 수 있고 그 사람의 게시글에 코멘트를 남길 수 있다.

친구들의 게시글 보기는 자신이 관심 있어 하는 해시태그를 포함한 친구들의 게시글을 볼 수 있는 기능이다. 자신이 관심이 있는 해시태그 중에서 하나를 선택하여 친구들의 게시글 중 관심이 있는 해시태그가 포함된 게시글만 볼 수 있어서 관심이 없는 것들은 보지 않을 수 있다.

자신의 스타일을 정하기는 여러가지 옷을 보고 자신만의 스타일을 만들고 자신의 스타일을 저장할 수 있는 기능이다. 자신이 관심 있는 스타일이 바뀌었다면 이 곳에서 손쉽게 스타일을 바꿀 수 있다.

중년들의 패션친구가 핸드폰 어플리케이션을 기반으로 하는 데에는 다음과 같은 이유가 있다. KISDI에 따르면 중장년층의 스마트폰 보유율은 2014년 81.6%, 2015년

88.5%, 2016 년 93.6%로 전체 스마트폰 보유율 증가속도보다 빠르게 증가해 옴을 알 수가 있다.

[그림 1] 연령대별 스마트폰 보유율 추이

(단위: %)



또한, 하루 평균 SNS 이용시간은 전체 응답자 기준 전년도에 비해 소폭 감소한 반면, 중장년층의 하루 평균 이용시간은 57 분으로 지속적으로 증가하는 추세임을 알 수가 있다.

[그림 5] 연령대별 SNS 이용자의 하루 평균 이용량 추이

(단위: 분)



위의 상황과 같이 중 장년층들에게 핸드폰이 가장 친숙한 매체이기 때문에 손쉽게 다운로드할 수 있고 편하게 사용할 수 있는 핸드폰 어플리케이션으로 개발을 할 예정이다.

3. Glossary

3.1 Objective

본 문서에서 기재되는 여러가지 기술적인 용어나, 다른 사람들에게는 생소할 수도 있는 용어들의 뜻을 정리한다. 배경지식이 없는 독자가 읽더라도, 문서를 이해하는 데에 어려움이 없도록 하는 것이 그 취지이다. 따라서 독자의 전문지식 수준을 추정하지 않고, 가능한 한 쉽게 용어들을 풀어서 설명한다.

3.2 Term Definition

A. 어플리케이션 사용 상의 용어

단어 s	뜻
타임라인	제목이 보여지고 그 제목을 클릭해야 게시글을 볼 수 있는 일반 게시판과 다르게, 사용자가 올린 글 내용전체가 보여지는 창이다. 그래서 글을 올린 최신 순대로 가장 위에 글이 뜨게 되어있다.
해시태그	SNS에서 사용되는 기호로써 해시기호(#) 뒤에 단어를 붙인 형태를 말한다. 이 해시태그를 통해서 그 단어가 포함된 글들을 검색할 수 있다.
My page	사용자가 "중년들의 패션친구" 어플리케이션을 이용할 때, 자신의 정보 및 자신이 고른 옷들을 볼 수 있는 페이지를 말한다. "친구 페이지"와 상대적인 개념이다.
스타일링	자신의 원하는 옷을 자신의 <u>스타일란</u> 에 추가하는 행위를 말한다.
카카오톡 ID 연계	카카오톡에서 사용하는 ID를 "중년들의 패션친구"에서도 사용할 수 있게끔 카카오톡 서비스를 연결시키는 것을 말한다
탭	비슷한 기능 및 서비스들을 하나로 모아서 볼 수 있게한 창이다.
<u>팝업창</u>	원래 존재하지 않았지만, 어떤 이벤트가 발생했을 때 사용자가 인지할 수 있게끔 새롭게 뜨는 창을 말한다.
닉네임	중년들의 패션친구에서 사용자를 <u>구별짓는</u> 이름이다. 로그인에 사용되는 아이디와는 다른 개념으로써, 게시물이나 comment를 남길 때 표시되는 이름이다.

Table2. 용어설명-어플리케이션상

B. 시스템 상의 용어

단어	뜻
쿼리	정보를 수집하기 위해서 시스템이 다른 시스템에 요청하는 행위
browse	사용자가 그 페이지에 머물면서 돌아보는 행위
api	오픈소스 라이브러리에 접근할 때 사용되는 규칙들
크롤링	인터넷에 존재하는 무수히 많은 정보들 중에서 내가 원하는 정보들만을 추려내어 그 정보들을 가져오는 기술
데이터베이스	서버에서 서비스를 제공할 때, 여러가지 데이터들을 일정한 규칙에 의해 저장하고 있는 저장소
서버	서비스에 응답하는 프로세스와 하드웨어를 통괄하는 용어
시나리오	프로그램이 어떻게 일들을 진행할지, 사용자는 프로그램을 어떻게 실행시킬지를 줄글 형식으로 체계화하여 작성한 글
취향벡터	사용자가 옷들을 고르고 나면, 그 사용자가 어떤 카테고리의 옷들을 좋아하는지를 표시하여 시스템이 이해할 수 있게끔 나타낸 1*p 형태의 행렬이다.
모바일 디바이스	들고 다닐 수 있는, 휴대가 가능한 기기를 말하는 것으로써, 이 기기를 통해서 중년들의 패션친구 서비스를 이용할 수 있다. 대표적으로 스마트폰이 그 예이다.
리소스	컴퓨터 시스템에 관한 여러가지 자원을 총칭하는 말로서, 데이터가 차지하는 공간 중앙처리장치의 시간 등을 지칭한다.

Table3. 용어정리- 시스템상

C. 패션 용어

단어	뜻
데님	옷감의 종류이다. 정통적인 블루 데님은 진한 청색의 경사에 회색이나 표백하지 않은 위사로 짠 천이다. 그 외에도 갈색, 흑색, 백색 외에도 스트라이브나 체크 무늬 등이 있다.
보헤미안	사회 관습에 구애되지 않는 자유분방한 생활을 하는 사람들을 일컫는 말로써, 보헤미안 스타일은 전체적으로 헐렁한 분위기로 옷을 걸쳐 입는 스타일이나, 스페인 풍의 특징을 가진 록을 말한다. 또한 집시 프린트와 같이 이국적인 무늬도 그 특징이다.
Hoodie	옷에 모자가 달린 종류들의 옷을 말한다.

Table4. 용어정리- 패션

4. User Requirement Definition

4. 1 Objective

User Requirements Definition 에서는 사용자에게 어떤 서비스를 제공하는지에 대해서 설명한다. 기능적인 시스템 요구사항뿐만 아니라 비기능적인 시스템 요구사항도 서술한다. 사용자가 이해할 수 있는 자연어와 다이어그램과 같은 개념들로 설명되어 있다. 시스템은 여기에 서술된 요구사항들을 반드시 충족할 것이다.

4. 2. Functional Requirements

[중년들의 패션친구]는 다음과 같은 기능을 제공하여야 한다.

A. Sign up

The image displays three sequential mobile app screens for the '중년들의 패션친구' (Middle-aged People's Fashion Friend) app, illustrating the sign-up process.

- Screen 1 (Left):** The initial sign-up screen. It features the app title '중년들의 패션친구' at the top. Below it are input fields for '아이디' (ID) and '비밀번호' (Password). A '로그인' (Login) button is present. At the bottom, there is a yellow button labeled '카카오계정으로 로그인' (Login with Kakao account) and a link '아이디가 없으신가요? 회원가입' (Don't have an ID? Sign up).
- Screen 2 (Middle):** The second step of the sign-up process. It includes input fields for '아이디' (ID), '비밀번호' (Password), and '비밀번호 확인' (Confirm Password). Below these is a section for '인적 사항' (Personal Information) with fields for '닉네임' (Nickname), '나이' (Age), and '성별' (Gender) with radio buttons for '남' (Male) and '여' (Female). A '회원가입' (Sign up) button is at the bottom.
- Screen 3 (Right):** The final step of the sign-up process. It shows the '인적 사항' (Personal Information) section with fields for '닉네임' (Nickname), '나이' (Age), and '성별' (Gender) with radio buttons for '남' (Male) and '여' (Female). A yellow button labeled '카카오계정으로 로그인' (Login with Kakao account) is at the bottom.

사용자에게 회원 가입을 위해 필요로 되는 정보를 요청하는 기능이다. 사용자는 [중년들의 패션친구]에서 제공하는 서비스를 이용하기 위해서는 회원 가입을 해야 한다. 회원 가입 방법은 크게 두가지로, 일반적인 회원가입과 보다 편리하게 진행할 수 있는 카카오톡으로 회원가입이 있다. 일반적인 회원 가입 시엔 아이디, 비밀번호와 함께 사용자의 성별, 나이를 입력하여야 한다. 카카오톡으로 최초 로그인할 시엔 사용자의 성별, 나이만 입력하며, 이후 로그인 시엔 매번 아이디와 비밀번호를 입력할 번거로움 없이 자동으로 로그인 할 수 있다.

B. Log in

사용자가 회원 가입을 통해 만들어진 아이디를 이용하여 시스템에 로그인하는 기능이다. 사용자는 [중년들의 패션친구]에 가입된 아이디를 이용해, 시스템에 로그인하여 My Page 에 접근할 수 있으며, 시스템에서 제공되는 서비스를 이용할 수 있다.

C. Explore user's preference

최초 로그인 이후 사용자는 등록된 개인 취향이 없을 경우, '취향 분석'을 할 수 있게 해주는 기능이다. 사용자는 카테고리별로 임의로 제공된 옷을 선택하여 선택된 옷들의 속성을 통해 개인의 선호하는 스타일이 무엇인지 알 수 있다.

D. View new clothes

로그인 이후 시행된 '취향분석(Explore user's preference)'의 연장선으로, 사용자는 카테고리별로 임의의 옷들을 선택하여 자신의 입맛대로 스타일링을 할 수 있다. 스타일링을 하게 되면 그 스타일링은 자동으로 분석되어 속성을 tagging 하여 My Page 에 올려진다. 또한 사용자는 스타일링의 속성이 자신의 기존 취향에 반영되어 지속적으로 취향이 갱신됨을 알 수 있다.

E. Configure My page

사용자 본인이 선택한 스타일링과 자신의 선호 속성을 볼 수 있는 기능이다. 취향분석과 새로운 스타일 탐색(View new clothes)에서 선택한 스타일링을 사용자가 스타일을 분류하여 보다 편하게 스타일링을 열람할 수 있다. 또한, 각 스타일링별로 속성 값을 확인할 수 있고 다른 사용자가 남긴 댓글을 통해 피드백을 받을 수 있다. 이로 인해 자신과 맞는 사용자를 찾아 나갈 수 있는 기회가 생긴다.

F. Recommend Friends

사용자의 현재 스타일 취향에 맞게 사용자에게 다른 사용자들을 추천해주는 기능이다. 불필요하게 자신과 맞지 않는 스타일을 가진 사용자들을 탐색할 필요없이 효율적으로 자신의 스타일을 유지하면서 다른 사용자들의 스타일링을 통해 새로운 옷을 탐구하거나 그들과 교류를 할 기회가 마련될 수 있다.

G. Add Friends

사용자가 다른 사용자를 관심 있게 보는 사람으로 추가할 수 있는 기능이다. Friend 로 추가하게 되면 언제든지 friend list 에서 접근하여 열람할 수 있다.

H. Go to Friend's Page

사용자가 다른 사용자의 정보 및 게시물을 볼 수 있는 페이지로 방문하는 기능이다. 다른 사용자의 선호하는 속성뿐만 아니라 그들이 분류한 스타일링을 열람할 수 있다. 또한, 스타일링별로 댓글을 달아 타 사용자와 소통을 할 수 있어 각 옷의 출처가 어디인지, 사이즈는 어떻게 나왔는지 등에 대한 의견을 주고받을 수 있다.

I. Search Friend

사용자가 아이디를 기반으로 다른 사용자를 검색할 수 있는 기능이다. 아이디는 중복이 불가하기 때문에 사용자는 오탈자가 없는 이상 정확히 그 특정인물을 찾을 수 있다.

J. Browse friends' style with user's preferred tag

사용자가 '관심 게시글' 탭에서 사용자의 선호 속성에 해당하는 friend의 게시글을 보여주는 기능이다. 즉, 사용자가 지정한 선호 속성(e.g. 데님, 보헤미안, etc.)과 사용자의 Friend를 기반으로 사용자의 friend가 올린 스타일링에 선호 속성이 포함되어 있을 경우에 그 스타일링들을 모두 보여준다. 사용자는 언제든지 선호 속성을 바꿀 수 있고 스타일 취향에 따라 추천되는 friend 목록이 다르므로 유동적으로 스타일링을 열람할 수 있다.

4.3. Non-Functional Requirements

Non-Functional Requirement로는 Product Requirement, Organization Requirement, External Requirement로 나눌 수 있다. 그리고 각 요소마다 세부적으로 더 분류될 수 있지만, 여기서는 중년들의 패션친구에 필요로 되는 것만 서술하고자 한다.

A. Product Requirements

A.1. Usability requirements

중년들의 패션친구는 중년들이 자신의 스타일을 만들고 그에 대해 친구들과 편히 이야기하며 쉽게 친구가 될 수 있도록 하는 시스템이다. 따라서 사용자가 스타일을 만드는데 불편함을 느끼지 않고 친구를 추가하는 것에 대해 어렵다고 생각하지 않도록 해야 한다. 따라서 스타일을 쉽게 선택할 수 있고 추천 친구를 쉽게 볼 수 있도록 직관적으로 배치해야 한다.

A.2. Performance requirements

모바일 디바이스의 경우 배터리 등의 추가적인 리소스 소모가 사용자에게 부담으로 다가올 수 있다. 따라서 디바이스 내 전력 소모를 최소화할 것이 요구된다. 또한, 옷에 대한 방대한 데이터와 많은 친구를 가질 때의 데이터를 나타내는데 문제가 없어야 한다. 중년들이 스타일을 고를 때 performance 측면에서 문제가 없도록 해야 한다.

A.3. Dependability requirements

모바일 디바이스의 경우 네트워크 상태가 안정적이지 않아 알림 수신이 제대로 이루어지지 않는 경우가 존재할 수 있다. 이러한 상황으로 인해 사용자가 받아야 할 옷에 대한 데이터를 제대로 받지 못할 경우 시스템 자체의 신뢰도가 떨어질 우려가 존재하므로 스타일을 고를 경우 오류 여부를 확인할 수 있도록 요청에 대한 응답 확인 과정이 필수적으로 요구된다.

A.4. Security requirements

중년들의 패션친구 시스템의 경우 사용자의 카카오톡과 연계하여 사용하므로 이에 따른 보안 이슈가 발생할 수 있다. 개인 정보들에 대한 접근의 경우 적합한 권한에 대한 확인을 언제나 요구해야만 하며, 암호화 등의 방법론을 통해 만약의 상황에서 데이터가 유출되었다 할 지라도 실제 개인 정보에 대한 접근을 불허하도록 해야 한다.

B. ORGANIZATIONAL REQUIREMENTS

B.1. Environmental Requirements

우선, 사용자들이 어플리케이션을 구동할 모바일 디바이스가 네트워크 상에 연결되어 있어야 한다. 또한, DB 를 관리하고 개별 모바일 디바이스들과의 통신을 담당할 중앙 서버 및 DB 서버가 상시 구동되어야 한다.

B.2. Operational Requirements

시스템 구성 요소의 경우 항상 사용자의 요청에 대한 응답을 받아들여야 하므로 구동이 멈춰서는 안 된다. 만약 중앙 서버 혹은 DB 서버, 사용자 계정 서버가 구동하지 않는 경우라면 이에 대한 요청을 한시적으로 보류한 뒤 서버 재개 이후 순차적으로 진행될 수 있도록 하여야 한다.

B.3. Development Requirements

새로운 옷에 대한 정보를 쉽게 업데이트 하기 위해 DB 는 등록 및 갱신이 용이하도록 구성하여야 한다. 또한, 어플리케이션은 안드로이드 모바일 기기를

대상으로 하기 때문에, 어플리케이션 부분의 개발 언어는 통상적으로 안드로이드 개발에 널리 사용되는 Java 로 제한된다.

C. EXTERNAL REQUIREMENTS

C.1. Legislative Requirements

카카오톡을 통한 개인 정보 수집의 경우 수집할 개인 정보의 종류 및 취급 방침에 대한 알리를 사전에 제공하여야 한다.

C.2. Safety Requirements

모든 처리가 중앙 서버를 한 번 이상 경유해야 하므로 중앙 서버가 네트워크 과부하에 의해 오작동하지 않도록 서버 용량 및 트래픽에 대한 예측이 빚나가지 않도록 해야 한다. 또한, 카카오톡 연계를 사용하는 경우 암호화와 권한 확인 절차를 반드시 거치도록 하여 개인 정보 유출 피해를 최소화할 수 있도록 한다

5. System Architecture

5.1 Objective

System Architecture에서는 각 시스템의 Architecture에 대한 자세한 개요를 보여 준다.

5.2 Explore Preference System

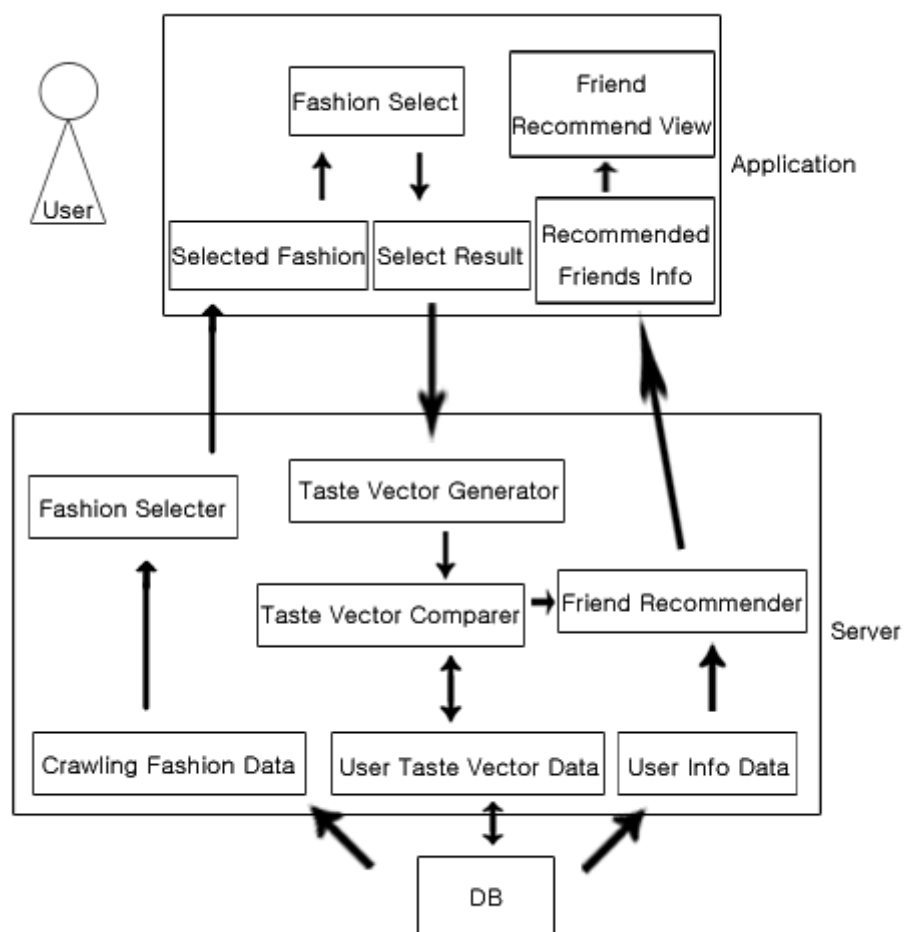


Diagram1. Explore Preference System

취향 분석 시스템(Explore Preference System)은 유저의 의상 선택 결과를 기반으로 취향을 분석하는 시스템이다. 그 과정에서 유저의 취향은 취향 벡터로 변환되며, 이를 데이터베이스에 존재하는 다른 유저들의 취향 벡터와 비교하게 된다. 그 결과로 유사한 취향을 가진 유저를 추려내며, 이를 바탕으로 유저에게 추천 친구를 제시하게 된다.

5.3 My Page System

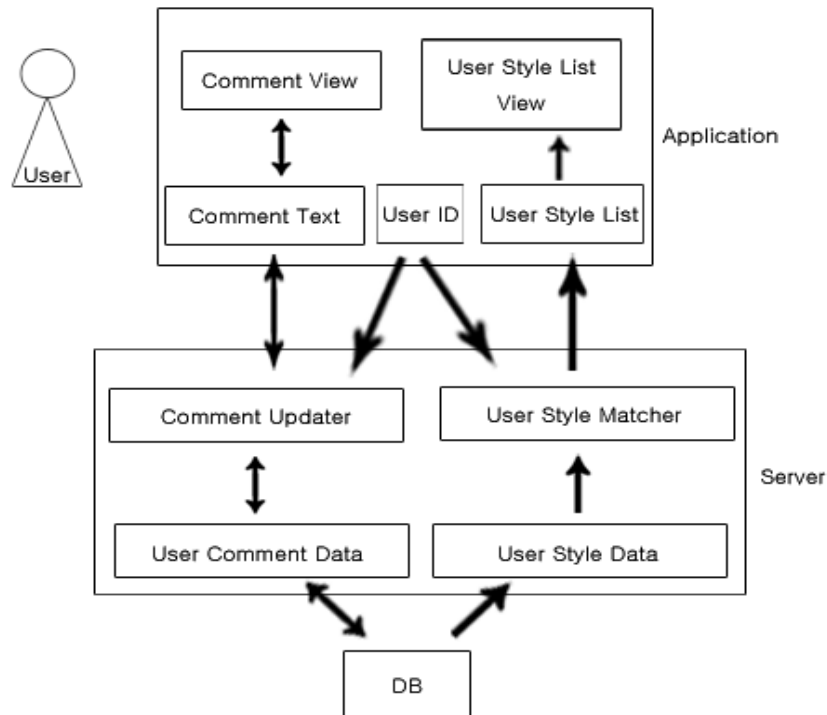


Diagram2 My Page System

마이 페이지 시스템(My Page System)은 유저의 패션에 관한 정보를 표시해주는 페이지로, 크게 두 기능으로 나뉜다. 첫번째 기능으로는 자신이 마음에 들어한 옷에 대해 자신 또는 다른 유저가 comment 를 남길 수 있는 기능이다. 이 comment 들은 데이터베이스에 저장되며, 이는 서버를 경유하여 유저의 화면에 즉각적으로 갱신된다. 또한 유저가 새로운 comment 를 남기면 그 또한 서버를 경유하여 데이터베이스에 저장되어 다른 유저가 볼 수 있게 한다. 두번째 기능으로는 자신이 저장한 스타일을 확인할 수 있다. 스타일은 각 유저별로 따로 데이터베이스에 저장되어 있으며, 서버를 경유하여 자신의 스타일을 화면에 보여주게 된다.

5.4 Friends' Tab System

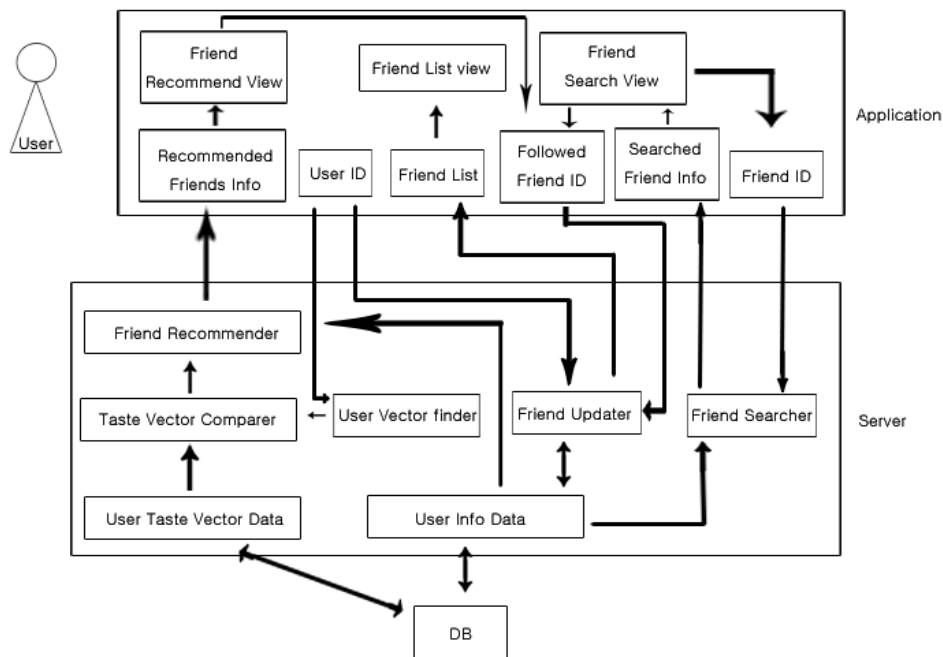


Diagram3 Friend System

친구 탭 시스템(Friends Tab System)은 유저의 친구에 대한 정보를 전체적으로 보여주는 시스템으로, 크게 네가지 기능을 가지고 있다. 첫번째 기능은 추천 친구를 보여주는 기능이다. 추천 친구는 데이터베이스에 저장된 유저의 취향 벡터를 기반으로 해서 이를 다른 유저의 취향 벡터와 비교함으로써 도출해낸다. 이 과정을 통해 도출된 추천 친구의 프로필을 유저의 화면에 출력해준다. 두번째 기능은 친구의 목록을 보여주는 기능이다. 친구의 목록은 유저의 아이디를 기반으로 그 유저의 친구에 해당하는 정보를 데이터베이스에 저장된 유저의 정보에서 추출해낸다. 이 리스트는 추천 친구 또는 친구 검색 기능을 통해 발견한 친구를 새로 추가할 경우에도 즉각적으로 갱신되어 표시되게 된다. 세번째 기능은 친구 검색 기능이다. 이는 친구의 아이디를 입력받아 이와 일치하는 유저가 있는지 데이터베이스에 저장된 유저의 정보와 대조하여 확인하게 된다. 그 결과, 일치하는 유저가 있다면 그 유저의 프로필을 보여주고, 없다면 아무것도 보여주지 않는다. 마지막 기능은 친구 추가 기능이다. 친구 추가는 추천 친구 기능과 친구 검색 기능의 일부에 포함된 기능이기도 하다. 유저가 친구의 프로필에 있는 친구 추가 버튼을 누르게 되면 이 입력을 트리거로 친구로 추가하기로 한 유저의 아이디를 추출해 서버로 보내게 된다. 서버에

서는 이 친구의 아이디를 데이터베이스에 존재하는 친구 리스트에 추가하게 된다.

5.5 Browse Friends' Style System

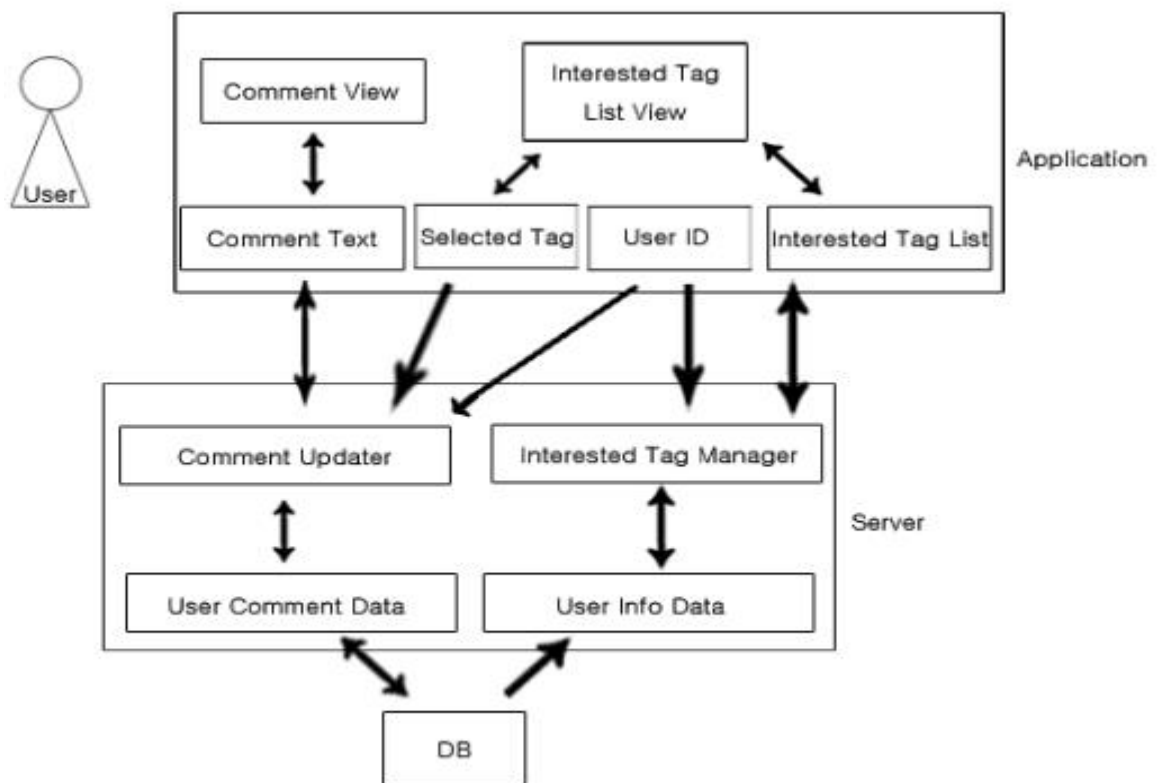


Diagram4 Browse Friends' Style System

관심 게시물 시스템(Browse Friends' Style System)은 자신이 등록한 태그를 가진 친구의 게시물만을 모아서 보여주는 시스템이다. 처음으로 데이터베이스에 저장된 유저의 정보를 불러와 그곳에 저장된 관심 태그 정보를 불러온다. 이를 기반으로 유저가 원하는 태그를 선택할 수 있도록 화면에 표시하게 된다. 유저가 태그를 선택하게 되면 데이터베이스에 저장된 게시물 데이터 중에서 친구의 게시물과 해당 태그를 가진 게시물이라는 조건을 둘 다 만족하는 게시물을 찾게 된다. 이렇게 찾은 게시물만을 모아 유저의 화면에 출력한다.

5.6 View New Clothes System

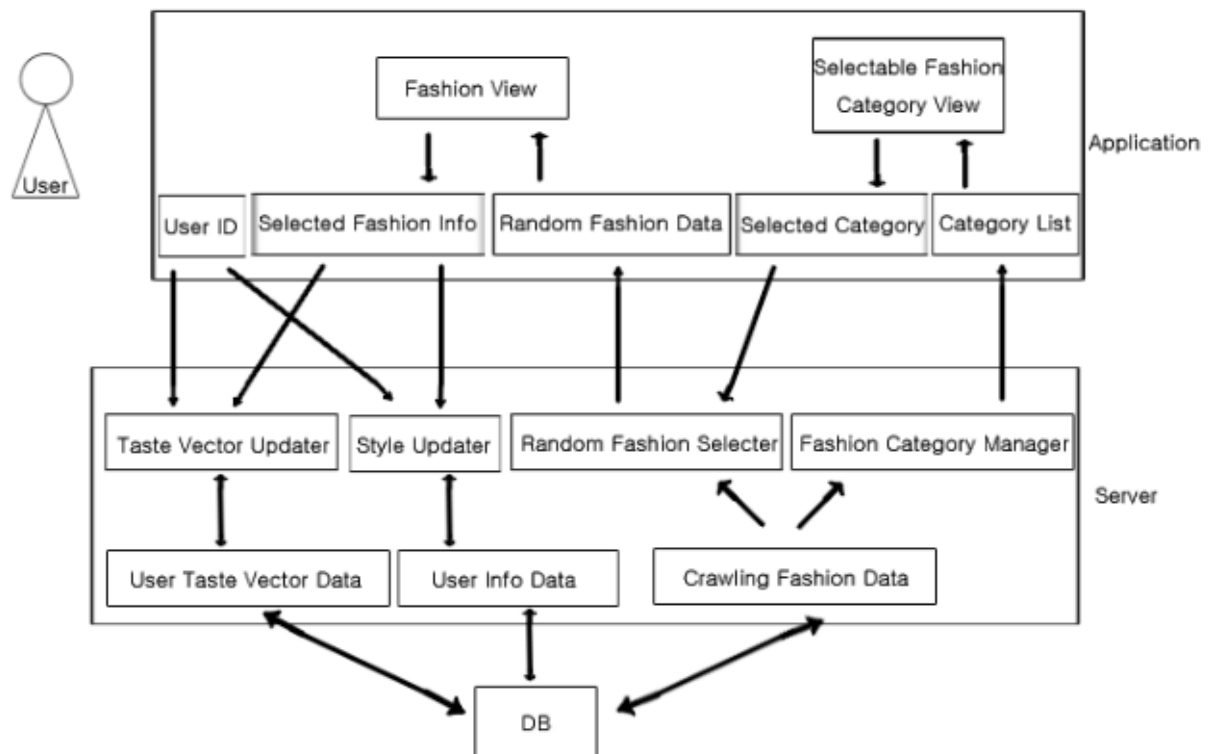


Diagram5 View New Clothes System

새로운 옷 탐색 시스템(View New Clothes System)은 패션 웹사이트에서 크롤링해온 이미지를 유저에게 제시해 마음에 드는 이미지를 자신의 스타일에 추가할 수 있게 해주는 기능이다. 이를 위해 먼저 크롤링해온 이미지를 데이터베이스에서 가져와 각 이미지에 매겨진 태그를 기반으로 각 카테고리를 추출한다. 이렇게 추출한 카테고리를 유저가 선택할 수 있게 화면에 출력하게 된다. 이 기능은 앞서 설명한 관심 게시물 시스템의 태그 선택 기능과 유사하게 작동한다. 이렇게 표시된 카테고리 중 하나를 유저가 선택하게 되면 서버에서 그 입력을 받아들여 크롤링해온 이미지에서 해당하는 카테고리를 태그로 가진 이미지만을 추출해 패션 데이터로서 유저의 화면에 보여주게 된다. 유저는 그렇게 표시된 옷 이미지 중에서 마음에 드는 것을 골라 자신의 스타일에 추가할 수 있다. 만약 유저가 옷을 스타일에 추가하게 된다면 그 이미지에 해당하는 정보들을 서버로 보내게 된다. 서버는 이렇게 받은 이미지를 기반으로 각각 데이터베이스에 존재하는 취향 벡터 정보 데이터와 유저 정보 데이터 중 스타일 정보를 갱신하게 된다.

5.7 New Clothes Crawl System

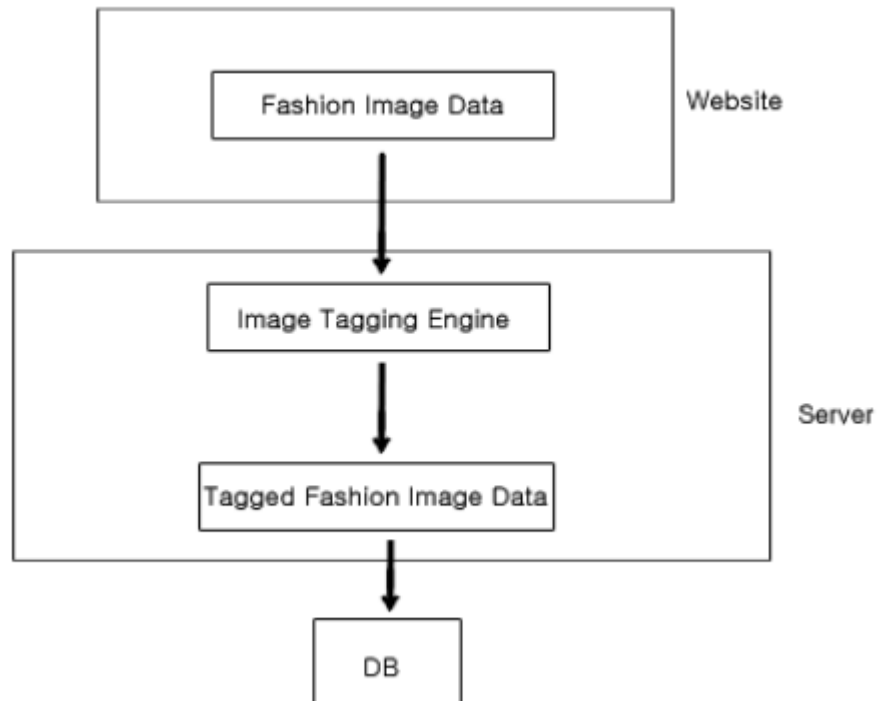


Diagram6. New Clothes Crawl System

새로운 사진 크롤링 시스템(New Clothes Crawl System)은 패션 웹사이트에 존재하는 패션 이미지를 크롤링해서 각 이미지에 해당하는 태그를 추가하는 시스템이다. 이를 위해 딥러닝을 이용한 자동 태그 시스템을 채용하게 된다. 이렇게 태그가 추가된 이미지들은 서버에 크롤링 이미지로서 저장되게 된다.

5.8 Friends' Page System

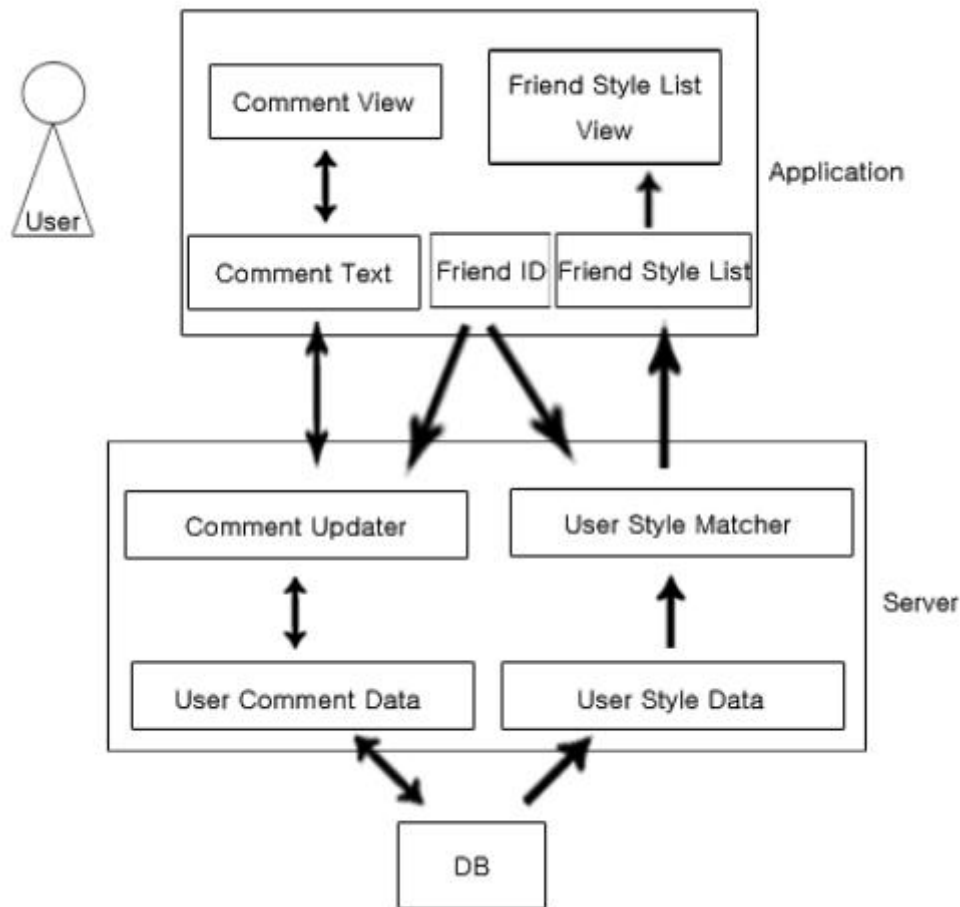


Diagram7 Friend Page System

친구 페이지 시스템(Friend Page System)은 친구의 패션에 관한 정보를 표시해주는 페이지로, 마이 페이지와 동일하게 유사하게 두 기능으로 나뉜다. 첫번째 기능으로는 친구가 마음에 들어한 의상에 comment 를 남길 수 있는 기능이다. 이 comment 들은 데이터베이스에 저장되며, 이는 서버를 경유하여 유저의 화면에 즉각적으로 갱신된다. 또한 유저가 새로운 comment 를 남기면 그 또한 서버를 경유하여 데이터베이스에 저장되어 다른 유저가 볼 수 있게 한다. 두번째 기능으로는 친구가 저장한 스타일을 확인할 수 있다. 스타일은 각 유저별로 따로 데이터베이스에 저장되어 있으며, 서버를 경유하여 친구의 스타일을 화면에 보여주게 된다.

6. System Requirement Specification

6.1 Objective

System Requirement Specification 단계에서는 시스템에 대한 기능적 요구 사항과 비기능적 요구사항에 대해 구체적으로 명시한다. 주된 독자층이 개발자이기 때문에 세부 기능들을 구현하는 데 필요한 구체적인 상황, 사항들을 중심으로 기술한다.

6.2 Functional Requirements

a. Login with Kakao Function

Functional Requirement	
기능	Login with Kakao Function
설명	'중년들의 패션친구' 예상 사용자들의 연령층을 고려하여 편의성을 위해 '카카오톡으로 로그인' 기능을 지원한다. 사용자는 '카카오톡으로 로그인'을 통해 쉽고 빠르게 '중년들의 패션친구'가 제공하는 서비스를 이용할 수 있다.
입력	카카오톡 API 를 이용하여 사용자의 카카오톡 내부 아이디를 얻는다. 최초 로그인 시, 닉네임, 나이, 성별을 추가로 입력 받는다.
출력	로그인이 정상적으로 완료되면, '나의 스타일' 화면으로 이동한다.
처리	사용자의 카카오톡 내부 아이디가 데이터베이스에 등록되어 있지 않다면 최초 로그인으로 간주하며, 시스템은 정보(아이디, 닉네임, 나이, 성별)를 데이터베이스에 저장한다.
조건	사용자는 정보 제공에 동의하여야 한다.

Table5. login with kakao Function

b. Sign up(회원가입) Function

Functional Requirement	
기능	Sign up Function
설명	새로운 사용자 등록을 위해서 필요한 정보를 얻는 기능이다. 카카오톡 아이디가 없는 사용자를 위한 일반적인 '회원가입'을 지원한다 회원 가입시 필요한 정보는 아이디, 비밀번호, 닉네임, 나이, 성별이다.
입력	필요한 정보인 아이디, 비밀번호, 닉네임, 나이, 성별을 각각 입력 받는다.
출력	사용자가 '회원가입'을 완료한 경우, 가입이 완료되었음을 알리는 팝업창을 출력한다. 아이디가 이미 존재하는 경우, 해당 아이디가 기존에 가입되었음을 알리는 팝업창을 출력한다. 닉네임이 이미 존재하는 경우, 해당 닉네임이 기존에 가입되었음을 알리는 팝업창을 출력한다.
처리	사용자가 가입을 완료한 경우, 시스템은 해당 정보(아이디, 비밀번호, 닉네임, 나이, 성별)를 데이터베이스에 저장한다.
조건	

Table6. Sign up Function

c. Sign in(로그인) Function

Functional Requirement	
기능	Sign in Function
설명	<p>사용자가 '회원 가입'을 통해 생성한 아이디를 이용하여 시스템에 로그인하는 기능이다.</p> <p>사용자는 로그인을 통해, '중년들의 패션친구'가 제공하는 서비스를 이용할 수 있다.</p>
입력	사용자로 하여금 '회원 가입'시 생성한 아이디 및 비밀번호를 로그인 형식에 맞추어 입력하게 한다.
출력	<p>로그인 정보가 일치하는 경우, '나의 스타일' 화면으로 이동한다.</p> <p>로그인 정보가 일치하지 않는 경우, 로그인 정보가 틀렸음을 알리는 팝업창을 출력한다.</p>
처리	입력한 아이디와 비밀번호가 모두 해당 데이터베이스와 일치하는지 확인한다.
조건	

Table7. sign in Function

d. Explore Preference (취향 분석) Function

Functional Requirement	
기능	Explore Preference Function
설명	<p>사용자의 취향이 없는 경우, 사용자의 취향을 분석한다. 분석된 취향은 이후 친구 추천, 취향 업데이트 때 사용된다.</p> <p>시스템은 사용자의 취향 분석을 위해 다수의 옷 사진을 사용자에게 보여준다. 사용자는 자신의 취향에 맞는 옷을 선택하여 취향을 간접적으로 전달한다. 시스템은 사용자가 선택한 옷에 달린 속성 정보(태그)를 바탕으로 사용자의 취향을 추론한다.</p>
입력	사용자로 하여금 자신의 취향에 맞는 옷을 하나 이상 선택하도록 한다.
출력	사용자가 아무런 옷도 선택하지 않은 경우, 하나 이상의 옷을 선택하도록 하는 팝업창을 출력한다.
처리	사용자가 선택한 옷들의 정보를 바탕으로 추론한 사용자의 취향을 사용자의 데이터베이스에 추가한다.
조건	사용자의 기존 취향이 존재하지 않는 경우에만, 실행되도록 한다.

Table 8. Explore Preference function

e. Configure Style Page (스타일페이지 출력) Function

Functional Requirement	
기능	Configure My Style Function
설명	사용자의 스타일 및 사용자의 게시 글을 볼 수 있는 화면을 출력해준다. 스타일에서는 '취향 분석' 때 선택한 옷들과 '새로운 옷 탐색' 때 선택한 옷들을 볼 수 있고, 게시 글에서는 글 내용과 다른 친구가 남긴 댓글을 확인할 수 있다.
입력	사용자가 탭의 첫번째 영역인 '내 스타일'을 선택하여 본인의 스타일 페이지에 대한 쿼리를 보낸다. 혹은 사용자가 다른 사용자를 선택함으로써 해당 사용자의 페이지에 대한 쿼리를 보낸다.
출력	사용자 본인 또는 다른 사용자의 스타일 페이지를 출력한다.
처리	사용자 본인 또는 다른 사용자의 아이디를 이용해 스타일 정보와 게시 글들을 데이터베이스에서 가져온다.
조건	

Table 9. Configure My style function

f. Friends tab (친구 탭) Function

Functional Requirement	
기능	Friends tab Function
설명	사용자의 친구 목록 및 추천 친구 목록을 보여주는 화면을 나타내는 기능이다. 사용자의 친구들을 보여주는 동시에 사용자의 취향을 바탕으로, 사용자와 비슷한 취향을 가지는 친구를 추천해준다.
입력	사용자가 탭의 두번째 영역인 '친구'를 선택하여 친구 탭에 대한 쿼리를 보낸다.
출력	사용자의 친구 목록 및 사용자와 비슷한 취향을 가지는 N 명의 추천 친구 목록을 출력한다. 각 항목을 통해 사용자는 친구 추가 또는 해당 사용자의 페이지를 방문할 수 있다.
처리	사용자의 아이디를 바탕으로 친구 목록과 모든 사용자의 취향을 데이터베이스에서 가져온다. 사용자의 취향을 바탕으로 추천 친구를 계산한다.
조건	벌써 친구인 대상에게는 친구 추가를 할 수 없다.

Table 10 Friends ' tab function

g. Add Friend (친구 추가) Function

Functional Requirement	
기능	Add Friend Function
설명	다른 사용자를 사용자의 친구로 추가하는 기능이다. '친구 탭'의 '추천 친구' 또는 '친구 검색'을 통해 친구로 추가될 수 있다.
입력	사용자로부터 특정 사용자에 대해 친구 추가 쿼리를 입력 받는다.
출력	정상적으로 친구 추가가 완료되었다면, 친구로 추가되었다는 팝업창을 출력한다.
처리	사용자 데이터베이스에 새로운 친구의 정보를 추가한다.
조건	벌써 친구인 대상에게는 친구 추가를 할 수 없다.

*Table 11. Add friends function***h. Find a Friend (친구 검색) Function**

Functional Requirement	
기능	Find a Friend Function
설명	다른 사용자의 닉네임을 입력 받아 그 닉네임에 해당하는 사용자를 친구로 추가할 수 있도록 하는 기능이다. 카카오톡 로그인 아이디를 알기 힘들기 때문에 이렇게 했다.
입력	사용자로부터 친구의 닉네임을 입력 받는다.

출력	데이터베이스 상에 존재하는 닉네임이라면 해당 사용자의 정보와 친구 추가 버튼을 출력한다. 단, 이미 친구인 사용자는 친구 추가 버튼이 나타나지 않는다.
처리	
조건	

Table 12. Find a Friend function

i. New Clothes Crawl (새로운 사진 크롤링) Function

Functional Requirement	
기능	New Clothes Crawl Function
설명	웹 상에 존재하는 새로운 옷 사진을 크롤링한 뒤, 기존에 학습된 딥러닝 모델을 활용해 해당 옷에 대한 속성 정보(태그)를 추가하여 저장한다.
입력	
출력	
처리	옷 데이터베이스에 옷의 아이디와 해당 옷의 태그를 저장한다.
조건	이미 데이터베이스 상에 존재하는 옷은 추가하지 않는다.

Table 13. New Clothes Crawl

j. Browse Friends' style (관심 게시글) Function

Functional Requirement	
기능	Browse Friends' style
설명	친구가 올린 게시글 중 사용자가 관심있어 하는 태그를 포함하는 게시글을 출력한다.
입력	사용자가 탭의 세번째 영역인 '관심 게시글'을 선택하여 사용자의 관심 게시글에 대한 쿼리를 보낸다.
출력	사용자의 관심 게시글 화면을 출력한다.
처리	관심 게시글은 친구의 게시글 중 사용자의 관심 태그를 포함하는 것으로 하며, 최신순으로 정렬한다.
조건	친구가 아닌 사용자의 게시 글은 표시되지 않는다.

*Table 14. Browse Friends' style***k. Write Comment (댓글 작성) Function**

Functional Requirement	
기능	Write Comment Function
설명	사용자 또는 친구가 올린 게시 글에 댓글을 추가하는 기능이다. 그 외의 사용자의 게시 글에는 댓글을 달지 못하게 한다.
입력	사용자가 게시 글에 댓글을 추가하는 쿼리를 보낸 경우, 그 내용을 입력 받는다.

출력	게시 글에 새로운 댓글이 추가된 경우, 올바르게 게시되었음을 알리는 팝업창을 출력한다.
처리	데이터베이스에서 해당 게시 글 가장 마지막에 댓글을 추가한다.
조건	게시 글은 반드시 사용자 본인 혹은 친구의 소유여야 한다.

Table 15. Write Comment

I. View New Clothes (새로운 옷 탐색) Function

Functional Requirement	
기능	View New Clothes
설명	사용자에게 새로운 옷들을 카테고리 별로 보여주는 화면을 출력하는 기능이다. 카테고리는 상의, 하의, 신발 등으로 이루어진다.
입력	사용자가 탭의 네번째 영역 '스타일 메이커'를 선택하는 쿼리를 보낸다. '스타일 메이커'화면 내에서 해당 카테고리를 선택하는 쿼리를 보낸다.
출력	'스타일 메이커'에 대한 쿼리를 받으면 첫 카테고리를 출력하며, 특정 카테고리에 대한 쿼리를 받으면 그에 해당하는 화면을 출력한다.
처리	기존에 데이터베이스에 존재하는 옷들을 무작위로 출력한다.
조건	사용자에 성별에 맞는 옷들 만을 출력한다.

Table 16. View New Clothes

m.Update Preference (취향 업데이트) Function

Functional Requirement	
기능	Update Preference Function
설명	사용자가 추가로 선택한 옷을 바탕으로 사용자의 취향 정보를 갱신하는 기능이다.
입력	탭의 네번째 영역 '스타일 메이커'에서 선택했던 옷들의 정보를 받는다.
출력	
처리	새로 얻은 옷들의 태그 정보를 바탕으로 기존의 취향을 새로운 취향으로 갱신한다.
조건	사용자의 기존 취향이 반드시 있어야 한다.

Table 17. Update Preference

6.3 Non Functional Requirements

Non-Functional Requirement 로는 Product Requirement, Organization Requirement, External Requirement 로 나눌 수 있다. 그리고 각 요소마다 세부적으로 더 분류될 수 있지만, 여기서는 중년들의 패션친구에 필요로 되는 것만 서술하고자 한다.

A. Product Requirements

A.1. Usability requirements

중년들의 패션친구는 중년들이 자신의 스타일을 만들고 그에 대해 친구들과 편히 이야기하며 쉽게 친구가 될 수 있도록 하는 시스템이다. 따라서 사용자가 스타일을 만드는데 불편함을 느끼지 않고 친구를 추가하는 것에 대해 어렵다고 생각하지 않도록 해야 한다. 따라서 스타일을 쉽게 선택할 수 있고 추천 친구를 쉽게 볼 수 있도록 직관적으로 배치해야 한다.

A.2. Performance requirements

모바일 디바이스의 경우 배터리 등의 추가적인 리소스 소모가 사용자에게 부담으로 다가올 수 있다. 따라서 디바이스 내 전력 소모를 최소화할 것이 요구된다. 또한, 옷에 대한 방대한 데이터와 많은 친구를 가질 때의 데이터를 나타내어 주는데 문제가 없어야 한다. 중년들이 스타일을 고를 때 performance 측면에서 문제가 없도록 해야 한다.

A.3. Dependability requirements

모바일 디바이스의 경우 네트워크 상태가 안정적이지 않아 알림 수신이 제대로 이루어지지 않는 경우가 존재할 수 있다. 이러한 상황으로 인해 사용자가 받아야 할 옷에 대한 데이터를 제대로 받지 못할 경우 시스템 자체의 신뢰도가 떨어질 우려가 존재하므로 스타일을 고를 경우 오류 여부를 확인할 수 있도록 요청에 대한 응답 확인 과정이 필수적으로 요구된다.

A.4. Security requirements

중년들의 패션친구 시스템의 경우 사용자의 카카오톡과 연계하여 사용하므로 이에 따른 보안 이슈가 발생할 수 있다. 개인 정보들에 대한 접근의 경우 적합한 권한에 대한 확인을 언제나 요구해야만 하며, 암호화 등의 방법론을 통해 만약의 상황에서 데이터가 유출되었다 할 지라도 실제 개인 정보에 대한 접근을 불허하도록 해야 한다.

B. ORGANIZATIONAL REQUIREMENTS

B.1. Environmental Requirements

우선, 사용자들이 어플리케이션을 구동할 모바일 디바이스가 네트워크 상에 연결되어 있어야 한다. 또한, DB 를 관리하고 개별 모바일 디바이스들과의 통신을 담당할 중앙 서버 및 DB 서버가 상시 구동되어야 한다.

B.2. Operational Requirements

시스템 구성 요소의 경우 항상 사용자의 요청에 대한 응답을 받아들여야 하므로 구동이 멈춰서는 안 된다. 만약 중앙 서버 혹은 DB 서버, 사용자 계정 서버가 구동하지 않는 경우라면 이에 대한 요청을 한시적으로 보류한 뒤 서버 재개 이후 순차적으로 진행될 수 있도록 하여야 한다.

B.3. Development Requirements

새로운 옷에 대한 정보를 쉽게 업데이트 하기 위해 DB 는 등록 및 갱신이 용이하도록 구성하여야 한다. 또한, 어플리케이션은 안드로이드 모바일 기기를 대상으로 하기 때문에, 어플리케이션 부분의 개발 언어는 통상적으로 안드로이드 개발에 널리 사용되는 Java 로 제한된다.

C. EXTERNAL REQUIREMENTS

C.1. Legislative Requirements

카카오톡을 통한 개인 정보 수집의 경우 수집할 개인 정보의 종류 및 취급 방침에 대한 알리를 사전에 제공하여야 한다.

C.2. Safety Requirements

모든 처리가 중앙 서버를 한 번 이상 경유해야 하므로 중앙 서버가 네트워크 과부하에 의해 오작동하지 않도록 서버 용량 및 트래픽에 대한 예측이 빗나가지 않도록 해야 한다. 또한, 카카오톡 연계를 사용하는 경우 암호화와 권한 확인 절차를 반드시 거치도록 하여 개인 정보 유출 피해를 최소화할 수 있도록 한다

6.4 Scenario

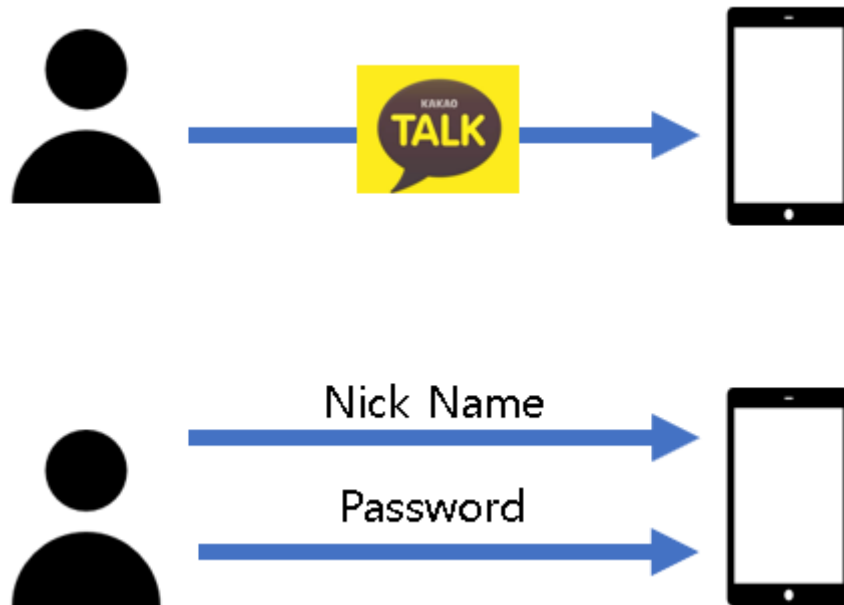
a. 사용자 사용 시나리오

a.1 회원가입 시나리오

Initial Assumption

이 어플을 처음 사용하는 사람이라고 가정한다. 그래서 자신의 아이디가 데이터 베이스에 존재하지 않기 때문에 새롭게 회원가입을 해야한다. 만약에 기존에 가입했던 user라면 , 그래서 데이터 베이스에 아이디가 존재한다면 회원가입 단계를 별도로 거치지 않는다.

Flow of Events



처음에 로그인을 위한 페이지가 뜬다. 아이디와 비밀번호를 입력하는 창도 있고, 카카오톡을 연동하여 로그인 하는 버튼도 있다. 그렇지만, 회원이 아닌 경우에는 회원가입 버튼을 눌러서 회원 가입하는 창으로 이동한다.

그리고 회원가입을 할 때는 2가지 방법이 있다. 카카오톡 아이디로 연동하는 방법이 있고, 그냥 일반적으로 본인의 닉네임과 패스워드를 입력하여 로그인을 하는 방법이 있다. 카카오톡으로 회원 가입을 할 수 있게 한 이유는 “중년들의 패션친구” 라는 어플리케이션의 주된 타겟층이 40~60대 연령대의 사람들이기 때문이다. 40~60 대 분들은 복잡한 인증절차에 익숙치 않고, 또 서툴기 때문에 이러한 서비스들을 잘 이용하지 않는 경우가 많다. 그렇기 때문에 가장 범용적으로 사용되는 그리고, 40~60대 분들도 아이디가 있을만한 카카오톡을 통해서 회원가입이 가능하게 만들었다. 카카오톡 아이디가 있는 것이 확인이 되면 자신의 카카오톡 아이디로 로그인을 할 수 있게 된다. 따라서 별도의 이메일 아이디 입력, 패스워드 입력을 하지 않아도 되고, 본인인증 절차를 거칠 필요가 없다.

그런데 카카오톡 아이디가 여의치 않는 사람들을 위해서 일반적인 회원가입 방법

도 만들었다. 자신이 원하는 닉네임과 패스워드를 입력하면 된다. 그리고 패스워드 확인을 위해 패스워드를 한 번 더 입력받았다.

각각의 방법을 통해서 회원가입을 완료하고 나면 자신의 아이디가 생기게 된다. 그 후에는 기본적인 인적사항인 나이와 성별을 입력한다. 이렇게 기본적인 인적사항을 입력하고 나면 회원가입 절차가 마무리 된다.

생길 수 있는 issue

회원가입을 진행하다가 아이디는 생성이 되었는데, 기본적인 인적사항이 입력되지 않았을 경우에는 다시 인적사항을 입력할 수 있게 하였다. 예를 들어 카카오톡 연동을 통해 카카오톡 아이디를 통해 로그인을 할 수는 있는데 기본적인 인적사항이 빠져있다면 다시 입력하게 하였다.

카카오톡 연동과 마찬가지로 기본적인 인적사항이 제대로 입력되어 있지 않다면, 다시 인적사항을 채워야만 한다.

a.2 취향 분석 시나리오

Initial Assumption

이 단계에서 다루는 취향분석은 취향분석을 처음 실시하는 user들을 대상으로 한다. 이 과정을 거쳤었던 user들은 다시 따로 이 단계를 통해서 취향분석을 실시하지는 않는다. 이 취향 분석을 실시했었던 user들은 차후에 자신의 취향을 반영하기 위해서는 새로 추천되는 상품들을 선택함으로써 변화되는 자신의 취향을 반영할 수 있다. 따라서 이 단계를 이용하는 사람들은 이 취향 분석 단계를 한번도 실행하지 않았던 사람들을 대상으로 한다.

Flow of Events

이 취향 분석단계에서 쓰이는 스타일 메이커 탭에서의 view new clothes 와 마찬가지로 이미지들을 마담 4060이나 꽃중년과 같은 쇼핑몰에서 크롤링 해왔던 이미지들을 사용한다. 따라서 user는 설문조사처럼, 자신이 관심있는 옷을 여러 선택지 중에서 고르라는 지령을 받는다.



위 그림과 같은 UI가 제공되면 user는 자신이 원하는 옷을 선택한다.

처음에 입력된 인적사항 중 성별에 따라서 제공되는 페이지가 다르다. 남성의 경우에는 dress나 블라우스 등의 여성용 의류는 제공되지 않는다. 보통 같은 옷의 범주에서 그 옷의 fabric 이나 패턴, 문양 별로 선택지를 구성한다.

user에게 옷을 제공할 때 shirt 별, pants 별, hoodie 별 이런식으로 제공하기 때문에 같은 범주의 옷 내에서 다양한 스타일의 선택권을 줌으로써 어떤 스타일을 선호하는지 알 수 있다. 그렇게 옷의 종류별로 선택을 5~6페이지에 걸쳐서 선택을 하고 나면 그 user의 취향을 어느정도 파악할 수 있게 된다.

이때 user는 자신이 원하는 만큼 옷을 고를 수 있다. 개수에 구매 받지 않고, 자신이 원하는 옷을 고르고, 그것에 대한 취향을 반영하여 친구를 추천하는 것이

우리 어플리케이션의 목적이다. 그래서 개수에 제한을 두지 않았다.

생길 수 있는 issues

옷을 고르라는 페이지가 5~6페이지이다. 그런데 중간에 user가 그만둘 수 있다. 그럴 경우에는 취향분석 단계를 완료한 것이 아니기 때문에 아예 시작하지 않은 것과 같은 상태가 된다. 그렇기 때문에 다시 로그인을 하게 된다면, 취향분석 단계를 진행해야한다. 예를 들어 user가 3 페이지까지 옷을 골라서 취향분석을 완료하고 어플리케이션을 꺼버렸다면, 다음에 로그인했을 때는 처음부터 다시 옷을 골라야 한다.

a.3 View new clothes

Initial assumption

원하는 옷을 고를 수 있는 user는 정상적으로 회원계정이 생성되어서 인적사항 기입까지 완료된 사람들이다. 또한 취향분석 단계를 거치고 난 사람들이 이용할 수 있는 서비스이다. 취향분석 단계를 거치지 않은 사람들은 취향분석 단계를 먼저 수행하고 나서 이 단계를 거칠 수 있다.

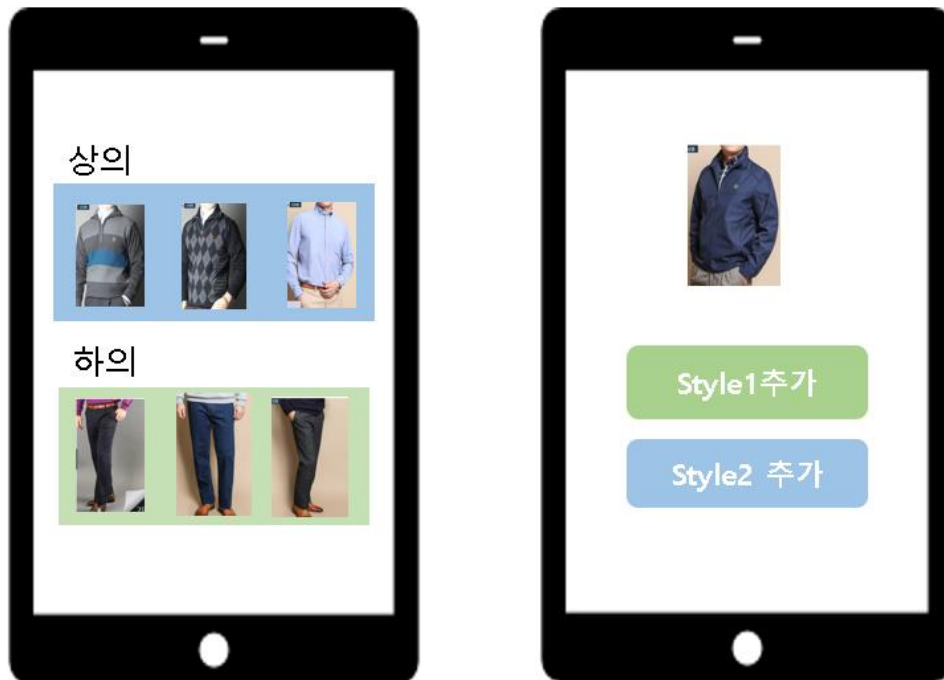
Flow of Events

“스타일 메이커” 라는 탭을 들어간다.

각각의 탭에 기능들이 다 있고, 스타일 메이커 탭에는 새롭게 추천되는 옷들이 있다. 이 때 , 보여지는 옷들은 “마담 4060” 이나 “꽃중년” 과 같은 인터넷 쇼핑 몰 사이트 상에서의 이미지들을 크롤링하여 제공되는 것들이다.

이 때 옷들은 여러가지 종류의 옷들이 제시 되는데 카테고리로 구분하여 카테고리별로 옷들을 제시한다. 카테고리라는 것은 상의, 하의, 외투로 나뉘어 옷이 제공될 예정이며, 남성 user이면 남자 옷들, 여성이면 여성 옷들을 추천하여 옷 이미지들을 띄울 것이다. 여자는 옷들 중에서 마음에 드는 옷을 선택한다. 마음에

드는 옷을 하나 선택한 후, 오른쪽 그림과 같이 어떤 스타일에 추가할 것인지를 정한다.



이렇게 자신이 원하는 스타일에다가 옷을 추가하고 나면, 자기 페이지에서 새롭게 골랐던 옷이 style에 추가되었음을 확인할 수 있다.

이렇게 골랐던 옷들은 지속적으로 user의 취향을 반영하는 데에 사용된다. User에게 가시적으로 드러나는 것은 아니지만, 시스템 내부에서는 그 옷들을 반영하여 그 user의 취향을 나타내는 vector가 달라진다.

생길 수 있는 issues

원하는 옷이 없는 경우에는 반드시 선택하지 않아도 된다. 선택하지 않을 경우에는 취향의 update가 되지 않으므로 추천되는 친구들의 목록에 변화가 없을 것이다. 그렇지만 새로 고침 버튼을 눌러서 친구들의 목록을 다시 뽑아볼 수는 있다.

a.4 자신의 페이지에 포스팅하기

Initial Assumption

자신의 페이지에 포스팅을 할 수 있는 사람들은 정상적인 회원가입 과정을 통해 계정이 생성된 사람들이다. 그리고 회원 가입 시, 처음 제공되는 취향 분석 설문 조사를 완료한 후에 자신의 페이지에 글을 포스팅할 수 있다.

Flow of Events

자신의 글을 포스팅하는 방법은 1가지가 있다.

스타일 메이커 탭에서 원하는 옷을 고른다. 그리고 나면 그 글을 자신의 페이지에 포스팅할 것인지 여부를 묻는 창이 뜬다. 그래서 user가 글을 쓰겠다고 하면 자신이 골랐던 옷이 함께 첨부되어 글을 남길 수 있다. 물론 글을 같이 포스팅해도 되고, 원하지 않는다면 사진만을 올려도 된다.

이 기능을 만들기 전에 중년들이 과연 10~30대의 사람들처럼 활발하게 글을 남기고, 소통할 것인지가 의문이었다. 10~30대 처럼 옷을 착용하고 난 후에 사진을 찍을 것 같지도 않고, 심지어는 사진을 첨부하는 방법도 잘 모를 수 있다고 판단하였다. 그래서 좀 더 편리하게, 자신이 골랐던 옷과 함께 글을 작성할 수 있도록 하였다.

a.5 알림 tab 사용하기

Initial Assumptions

정상적인 회원가입 절차를 통해 성공적으로 계정이 생성된 회원들이 사용한다. 그리고 회원가입 직후, 실시하는 취향분석 설문조사를 완료한 사람들이어야 한다. 왜냐하면 그 결과를 바탕으로 그 user의 취향 vector를 구할 수 있고, 그 vector를 통해서 유사도를 구해, 유사한 다른 사람들을 추천해줄 수 있기 때문이다.

Flow of Events

알림 탭의 가장 큰 목적은 자신이 원하는 태그가 있다면, 그 tag 에 해당하는 옷을 고른 친구들의 포스팅된 글을 보여주는 것이다.

먼저 알림 탭에 들어간다. 자신이 설정한 “관심있는 태그” 가 없다면, 친구 및 친구의 포스팅 된 글이 뜨지 않는다. 그럴 경우에는 태그를 추가해달라는 글이 뜨게 된다.

그렇게 태그를 추가하고 난 후에는 정상적으로 내가 관심있는 태그에 해당하는 친구들의 옷이 포스팅된 글들을 볼 수 있다. 그래서 더 관심이 생긴다면 진짜 그 글을 쓴 사람의 페이지에 들어가서 comment를 남기거나, 다른 게시물들을 살펴볼 수 있다.

알림 탭은 자신이 관심있어하는 태그가 포함된 친구의 게시글들이 페이스북의 타임라인 처럼 올라와 있는 형태이다.

생길 수 있는 issues

이 때 user가 고른 tag가 존재하지 않는다면 tag를 설정해달라는 메시지가 뜬다. 그리고 tag를 추가하는 방법은 알림 탭에서 “#” 기호를 붙이고, 원하는 태그 단어를 입력하면 된다.

a.6 친구 추가하기 (친구 추천목록에서)

Initial Assumptions

회원가입 과정이 성공적으로 진행되어 계정이 있는 사람과 그 후 취향분석까지 완료된 user들에 한해서 이 서비스를 제공한다. 그래야 시스템이 user의 취향을 파악할 수 있고, 그것을 바탕으로 친구를 추천해줄 수 있다.

Flow of Events

여러 개의 탭 중에서 친구관련 탭에 들어간다.



위 화면과 같은 UI가 user에게 제공된다. 그래서 user는 친구를 추가하는 방법이 2가지가 있다.

먼저 자신의 지인 중에 이 어플을 사용하는 사람이 있다면 그 사람의 ID를 직접 입력하여 친구를 추가하는 방법이 있다. 그래서 추천되는 친구 말고도 다른 친구를 추가할 수 있도록 하였다. 이 때, 지인의 아이디를 검색하면 친구 추천 목록처럼 아이디 옆에 + 기호와 같이 뜬다. 그 때 + 버튼을 눌러서 추가하면 된다.

“중년들의 패션친구” 어플리케이션에서는 친구를 추가할 때, 상대방의 의사를 필요로 하지 않는다. 즉 상호 수락을 해야지만 친구가 되는 방식이 아니다. User가 원한다면 그 친구추가 해서 관심있게 지켜보는 친구로서 추가하는 것이라고 말할 수 있다. 상호 수락을 해야만 상대방의 코디들을 볼 수 있다면 “중년들의 패션친구”의 기본철학과 어긋난다고 생각했다. 이 어플리케이션 자체가 자신이 좋

아하는 취향을 오픈하고, 비슷한 취향을 가진 사람들과 자신의 패션을 공유하는 것이다. 그렇기 때문에 보다 많은 사람들의 패션 취향을 쉽게 파악하자는 것이 기본 철학이다. 그렇기 때문에, 상호 수락을 통해야만 상대방의 코드를 볼 수 있다면 상당히 제한적으로 공유가 될 것이라고 생각하였고, 상호 수락까지 걸리는 시간이 오래 걸릴 것이라고 생각하였다.

또한 이 어플리케이션은 다른 사적인 정보를 요구하지도 않고, 오로지 패션에 대한 코드를 공유하는 것이기 때문에 상대방이 자신의 코드를 보고, 본인 또한 상대방의 코드를 보는 것이 당연한 것이기 때문에 굳이 상호수락을 전제하지 않았다.

생길 수 있는 issues

존재하지 않는 ID를 입력하는 경우에는 아무런 검색 결과가 없다고 창이 뜬다. 그래서 존재하는 ID에 한해서만 친구를 추가할 수 있다.

그리고 취향이 분석되지 않아서 user의 취향 vector가 정의되어 있지 않다면, 추천할 수 있는 기준이 없는 것이다. 그래서 추천 친구 목록에 친구가 뜨지 않는다. 따라서 먼저 취향분석을 완료해야 한다.

a.7 친구 게시물에 comment 남기기

Initial Assumption

User가 게시물을 남기려고 하는 사람을 친구 추가 하여 친구인 상태이어야 한다. 그래야 친구의 게시물을 볼 수 있고, 친구의 페이지에 들어가서 글을 남길 수 있다.

Flow of events

친구의 글에 글을 남기는 방법은 2가지가 있다.

먼저 알림 탭에 들어간다. 알림 탭에는 user가 관심있는 태그에 해당되는 친구의

글이 올라온다. 그래서 자신이 원하는 글에 대해서 comment를 남기는 란에 가서 comment를 남기고 확인을 누르면 된다.

또 다른 방법은 친구 탭에 들어간다. 친구 탭에는 이미 나와 친구인 사람들의 목록이 있다. 그래서 그 목록 중에 방문하고 싶은 친구의 페이지에 들어가도 된다. 그렇게 친구의 페이지에 들어가면 친구가 스타일메이커 탭에서 선택했던 옷들이 포스팅 되어 있다. 그래서 user가 관심있는 글에 들어가서 comment를 남기는 란에 comment를 남기고 확인을 누르면 된다.

생길 수 있는 issues

Comment를 쓰기는 했지만, 제대로 확인 버튼을 누르지 않는다면 user가 쓴 comment가 제대로 저장되지 않고, 처리되지 않는다. 그러므로 반드시 확인 버튼을 누를 수 있도록 한다.

B. 시스템 상 시나리오

b.1 취향 분석

Initial Assumptions

User가 자신의 취향을 잘 반영하여 선택할 것임을 가정한다. 그저 장난으로, 아무 생각없이 거의 random에 가깝게 옷을 선택하지는 않을 것이라고 가정한다.

Flow of Events

이 때 user에게 설문조사 형식으로 5~6가지 정도의 질문을 하게된다. 이 때 질문의 유형은 가장 마음에 드는 옷을 고르라는 것으로 대개 동일하다. 그렇지만, 각 질문을 이루는 사진들의 유형이 다르다.

Category 별로 사진을 제시하는 것이 기본 원칙이다. 왜냐하면 여성 분들 중에서

바지가 어떤 문양이든, 치마가 어떤 스타일이든, 단지 치마라는 이유로 바지를 선호할 수도 있기 때문이다. 따라서 기본적인 스타일과 어떤 느낌의 옷을 좋아하는지를 파악하기 위해서는 비슷한 유형의 옷들 중 문양, fabric 이 다른 것들을 위주로 그 선택지를 구성할 것이다.

그리고 나중에는 같은 하의 중에서도 드레스, 치마, 바지, 레깅스 등 어느 것을 선호하는지 상이한 종류의 옷들 중에서 선택할 수 있게 할 것이다.

남성도 마찬가지로 셔츠들 중에서 민무늬, stripe, chino 셔츠 등을 하나의 question으로 묶어서 제시할 것이고, hoodie라면 hoodie 중에서도 zipup 이 좋은지, 캐릭터가 들어가 문양이 좋은지를 비교해볼 것이다. 그리고 나서 상의 중에서 셔츠가 좋은지, hoodie가 좋은지를 비교할 수 있는 선택지를 제시할 것이다.

그렇게 해서 user가 선택지를 고르고 나면, 그 user의 고유의 vector가 나온다. 이 vector는 user가 선택했던 옷에 따라서 달라지는 것으로서 user의 취향을 반영하고 있는 vector이다. User의 vector를 구하는 방법은 다음과 같다.

다양한 옷 라벨들이 쭉 있다면, user가 고른 옷이 해당되는 라벨들을 전부 check 하는 것이다. 그렇게 해당되는 라벨들에 대해서 모두 check를 하고 나면 그 user를 나타내는 vector 가 나오게 되는 것이다.

생길 수 있는 issues

User가 너무 적게 옷을 선택했을 수도 있다. 그런 경우에는 바로 친구 추천이 원활하게 이루어지지 않을 수도 있고, 별로 자신의 취향과 맞지 않는 친구가 추천될 수도 있다. 처음에 심사숙고 하여 옷들을 잘 고르지 못한 경우라면, 추후에 스타일 메이커 탭에서 새로운 옷들을 고를 때에 활발하게 옷을 고르게 된다면 user의 취향 정보에 부족한 정보들이 잘 채워질 것이다.

b.2 크롤링된 이미지 분석

Initial Assumptions

스타일 메이커 탭에서 user가 자신의 취향에 맞는 옷을 고를 때 제공하는 이미지이다. 따라서 이 때도 user가 random이 아니라 자신의 취향에 부합하는 옷을 현명하게 선택해서 고를 것이라는 전제 하에 진행한다.

Flow of Events

"마담 4060", "꽃중년", "마담그레이스" 와 같이 주로 40~60 대를 주 타겟으로 하는 쇼핑몰의 옷 이미지들을 크롤링해서 데이터베이스에 저장한다. 실시간으로 크롤링을 하지는 않고, 주기적으로 기간을 정하여 크롤링을 할 계획이다. 왜냐하면 하루가 다르게 새로운 트렌드의 옷이 나오지는 않다고 판단했기 때문이다.



따라서 데이터베이스에 크롤링된 이미지들을 저장하고, 이 이미지를 스타일 메이커 탭에 게시한다. 이 때 크롤링해온 이미지들은 test data로써 pretrained 된 딥러닝 모델에 의해 다양한 라벨로 분류가 진행된다. 따라서 새로운 test data 에

대해서는 분석하는 시간이 오래 걸리지 않을 것으로 예상된다. 그 라벨별로 태그가 형성되어 같이 게시된다.

생길 수 있는 issues

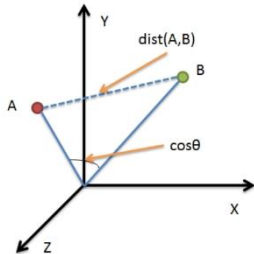
이 과정은 system 내부에서 크롤링을 해오고, 그 이미지들을 분석하는 것이기 때문에 user가 잘못 조작하여 이 프로세스가 잘못되는 경우는 없다.

b.3 친구추천

Initial Assumptions

이 어플리케이션에 가입한 사람이 2명 이상이어야 하고, 그 사람들 모두 취향분석 단계를 완료해서 그 사람들의 취향을 나타내는 vector가 존재한다.

Flow of Events



User가 취향 분석 단계를 통해, 혹은 스타일 메이커 탭에서 원하는 스타일을 선택했을 때의 옷들을 분석하여 어떤 태그를 선호하는지가 vector에 기록이 된다. 이 때 user는 복수의 스타일 (옷장)을 가질 수 있지만, style 1의 취향에 대해서만 친구들이 추천된다.

이 때 친구를 추천하는 방식은 자신의 vector와 다른 친구의 vector 간의 "cosine similarity"를 구한다. 위의 그림과 같이 A vector와 B의 vector의 단순한 거리가 아니라, 얼마나 비슷한 방향인지를 나타내기 위하여 cosine similarity를 구하는 것이다.

이 때 다른 회원들의 style 1에 대해서 user와 cosine similarity를 구하고 그 때 유사도가 85%가 이상인 사람들만 추천한다.

7. System Model

7.1 Objective

System models에서는 시스템 컴포넌트, 시스템 그리고 시스템 환경 사이의 관계를 보여준다. 가능한 그래픽 모델로는 object models, data-flow models, semantic data models 등이 있다.

7.2 Context Model

A. Context model

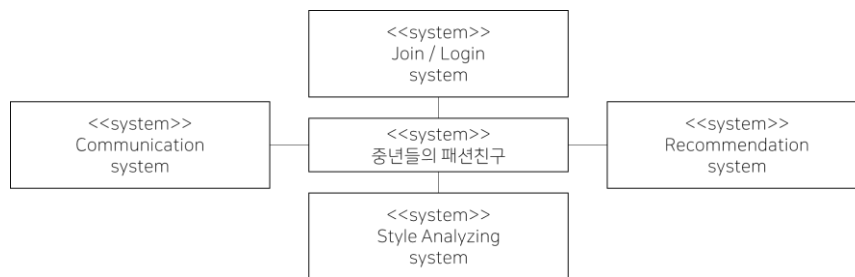


Diagram 8. Context Model

B.Process diagram

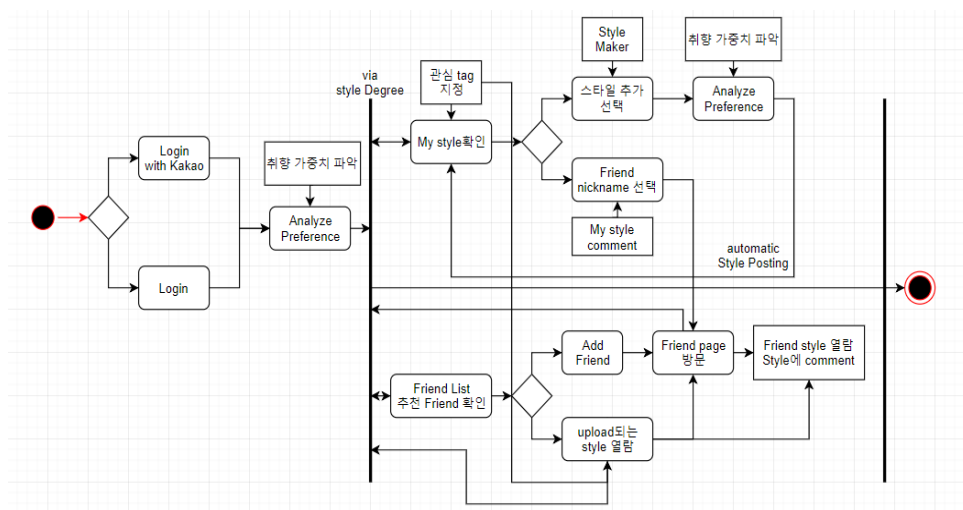


Diagram 9. Process diagram

7.3 Interaction Model

a. Use case model

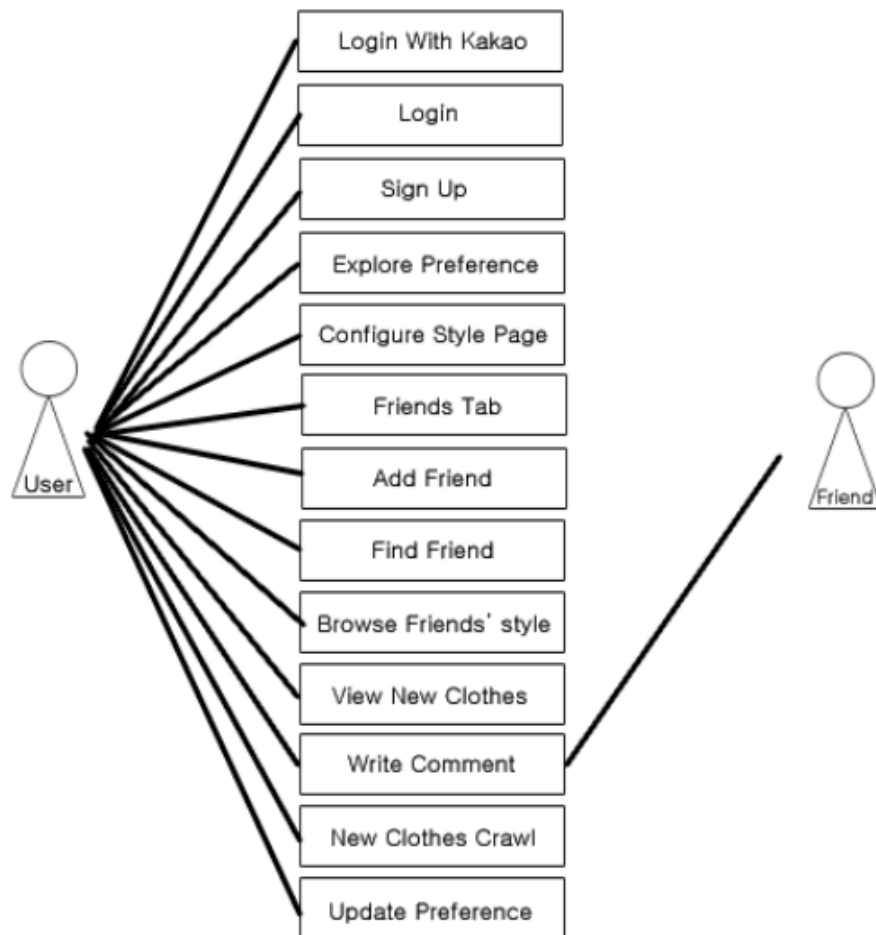


Diagram Use Case Diagram

b. Tabular Description For Each Use Case

b.1 Login With Kakao

Use Case	Login With Kakao
Actor	User
Data	Kakao Account Data(from Kakao Talk Server), User Information Data, User Information DB
Description	카카오톡 API 를 이용하여 사용자의 카카오톡 내부 아이디를 얻는다.
Stimuli	유저가 '카카오톡으로 로그인' 버튼을 눌렀을 경우
Response	<p>사용자의 카카오톡 내부 아이디가 데이터베이스에 등록되어 있지 않다면 최초 로그인으로 간주한다. 최초 로그인 시, 닉네임, 나이, 성별을 추가로 입력 받는다. 그 후, 시스템은 정보(아이디, 닉네임, 나이, 성별)를 데이터베이스에 저장한다. 사용자의 카카오톡 내부 아이디가 데이터베이스에 등록되어 있다면 사용자의 카카오톡 내부 아이디를 이용하여 로그인한다.</p> <p>로그인이 정상적으로 완료되면, '나의 스타일' 화면으로 이동한다.</p>
Comments	

Table 18: Login With Kakao Description

b.2 Login

Use Case	Login
Actor	User
Data	User ID, User Password, User Information DB
Description	사용자로 하여금 '회원 가입'시 생성한 아이디 및 비밀번호를 로그인 형식에 맞추어 입력하게 한다.
Stimuli	유저가 로그인 버튼을 눌렀을 경우
Response	입력한 아이디와 패스워드가 데이터베이스에 등록되어 있지 않다면 로그인은 실패하고, 로그인 정보가 틀렸음을 알리는 팝업창을 출력한다. 입력한 아이디와 패스워드가 데이터베이스에 등록되어 있다면 입력한 아이디와 패스워드를 이용하여 로그인한다. 로그인이 정상적으로 완료되면, '나의 스타일' 화면으로 이동한다.
Comments	

Table19 : Login Description

b.3 Sign Up

Use Case	Sign Up
Actor	User
Data	User ID, User Password, User Information Data, User Information DB
Description	필요한 정보인 아이디, 비밀번호, 닉네임, 나이, 성별을 각각 입력 받는다.
Stimuli	유저가 회원가입 버튼을 눌렀을 경우
Response	사용자가 '회원가입'을 완료한 경우, 가입이 완료되었음을 알리는 팝업창을 출력한다. 사용자가 가입을 완료한 경우, 시스템은 해당 정보(아이디, 비밀번호, 닉네임, 나이, 성별)를 데이터베이스에 저장한다.

	데이터베이스에 입력한 아이디가 이미 존재하는 경우, 해당 아이디가 기존에 가입되었음을 알리는 팝업창을 출력한다. 데이터베이스에 입력한 닉네임이 이미 존재하는 경우, 해당 닉네임이 기존에 가입되었음을 알리는 팝업창을 출력한다.
Comments	

Table 20 Sign up Description**b.4 Explore Preference**

Use Case	Explore Preference
Actor	User
Data	User ID, User Taste Vector Data, User Taste Vector DB, Crawling Fashion Data, Crawling Fashion DB
Description	데이터베이스에서 크롤링 해온 이미지를 가져와 사용자의 인터페이스에 출력하여 사용자로 하여금 그 중에서 자신의 취향에 맞는 옷을 하나 이상 선택하도록 한다.
Stimuli	유저가 로그인 했을 때, 기존의 취향 정보가 존재하지 않을 경우
Response	사용자가 선택한 옷들의 정보를 바탕으로 사용자의 취향을 취향 벡터의 형태로 추출한다. 그 후, 추출한 취향 벡터를 데이터베이스에 저장한다. 사용자가 아무런 옷도 선택하지 않은 경우, 하나 이상의 옷을 선택하도록 하는 팝업창을 출력한다.
Comments	

Table 21. Explore Preference Description

b.5 Configure Style page

Use Case	Configure Style Page
Actor	User
Data	User ID, User Style Data, User Style DB, User Comment Data, User Comment DB
Description	사용자가 본인의 스타일 페이지에 접속하여 자신의 스타일 페이지에 대한 쿼리를 보낸다. 혹은 사용자가 다른 사용자의 스타일 페이지에 접속하여 해당 사용자의 스타일 페이지에 대한 쿼리를 보낸다.
Stimuli	자신 또는 다른 유저의 스타일 페이지에 접속했을 경우
Response	사용자 본인 또는 다른 사용자의 아이디를 이용해 스타일 정보와 게시물들을 데이터베이스에서 가져온다. 이를 기반으로 사용자 본인 또는 다른 사용자의 스타일 페이지를 출력한다.
Comments	

*Table 22. Cofigure style Page Description***b.6 Friend Tab**

Use Case	Friend Tab
Actor	User
Data	User ID, User Friend Data, User Information DB, User Taste Data, User Taste DB
Description	<p>사용자가 친구 페이지에 접속하여 친구 탭에 대한 쿼리를 보낸다.</p> <p>사용자의 아이디를 바탕으로 데이터베이스에서 친구 목록을 가져온다.</p> <p>모든 사용자의 취향 벡터를 데이터베이스에서 가져온다.</p> <p>데이터베이스에서 가져온 사용자의 취향 벡터를 바탕으로 다른 사용자의 취향 벡터와 비교하여 추천 친구를 계산한다.</p>
Stimuli	친구 페이지에 접속했을 경우

Response	사용자의 친구 목록 및 사용자와 비슷한 취향을 가지는 N 명의 추천 친구 목록을 출력한다. 각 항목을 통해 사용자는 친구 추가 또는 해당 사용자의 페이지를 방문할 수 있다.
Comments	

Table 23. Friend tab Description**b.7 Add Friend**

Use Case	Add Friend
Actor	User
Data	User ID, Friend User ID, User Friend Data, User Information DB
Description	사용자로부터 특정 사용자에 대해 친구 추가 쿼리를 입력 받는다.
Stimuli	친구 추가 버튼을 눌렀을 경우
Response	정상적으로 친구 추가가 완료되었다면, 친구로 추가되었다는 팝업창을 출력한다. 그 후, 사용자의 아이디를 기반으로 사용자 정보 데이터베이스에 새로운 친구의 정보를 추가한다.
Comments	

Table 24 : Add Friend Description**b.8 Find Friend**

Use Case	Find Friend
Actor	User
Data	User ID, Friend User Name, User Information Data, User Information DB
Description	사용자로부터 친구의 닉네임을 입력 받는다.
Stimuli	친구 검색 버튼을 눌렀을 경우

Response	닉네임을 유저 정보 데이터베이스에 조회한다. 조회 결과, 데이터베이스 상에 존재하는 닉네임이라면 해당 사용자의 정보와 친구 추가 버튼을 출력한다.
Comments	

Table 25 : Find Friend description

b.9 Browse Friends' style

Use Case	Browse Friends' style
Actor	User
Data	User ID, User Information Data, User Information DB, Comment Data, Comment DB
Description	사용자가 관심 게시글 페이지에 접속하여 사용자의 관심 게시글에 대한 쿼리를 보낸다. 화면에 출력된 관심 태그 리스트 중에서 원하는 태그를 선택하면 그 태그의 종류에 해당하는 친구의 게시글을 출력한다.
Stimuli	관심 게시글 페이지에 접속했을 경우
Response	사용자의 아이디를 통해 데이터베이스에서 관심 태그 정보를 가져와 화면에 출력한다. 사용자가 태그를 선택하면 게시글을 데이터베이스에서 가져와 그 중에서 해당 태그를 가진 게시글만을 추출한다. 이렇게 추출한 게시글을 사용자의 화면에 최신순으로 출력한다.
Comments	사용자가 관심 태그를 바꿀 때마다 그에 맞는 게시글을 출력하는 작업을 반복한다.

Table 26: Browse Friends' style description

b.10 View New clothes

Use Case	View New Clothes
Actor	User
Data	Crawling Fashion Data, Crawling Fashion DB

Description	사용자가 스타일 메이커 페이지에 접속하여 사용자의 스타일 메이커 페이지에 대한 쿼리를 보낸다. 화면에 출력된 카테고리 리스트 중에서 원하는 카테고리를 선택하면 그 카테고리에 해당하는 스타일들을 출력한다.
Stimuli	스타일 메이커 페이지에 접속했을 경우
Response	데이터베이스에서 크롤링한 이미지 데이터를 가져와 데이터에서 카테고리를 추출하여 화면에 출력한다. 사용자가 카테고리를 선택하면 데이터베이스에서 가져온 크롤링한 이미지 데이터 중에서 해당 카테고리를 가진 스타일을 추출한다. 이렇게 추출한 스타일을 사용자의 화면에 랜덤으로 뽑아 출력한다.
Comments	사용자가 카테고리를 바꿀 때마다 그에 맞는 스타일을 출력하는 작업을 반복한다.

Table 27: View New Clothes description

b.11 Write comments

Use Case	Write Comment
Actor	User, Friend User
Data	User ID, Comment Data, Comment DB
Description	사용자가 입력한 Comment 를 읽어온다.
Stimuli	Comment 작성 버튼을 눌렀을 경우
Response	입력한 Comment 를 읽어와 해당하는 Comment 데이터베이스에 추가한다. 댓글이 정상적으로 추가된 경우, 올바르게 게시되었음을 알리는 팝업창을 출력한다. 갱신된 Comment 는 해당 User 의 Page 에도 반영되어 표시된다.
Comments	

Table 28: Write Comment description

b.12 New Clothes Crawl

Use Case	New Clothes Crawl
Actor	User
Data	Fashion Image Data(From Shopping Mall Page), Crawling Fashion Data, Crawling Fashion DB
Description	웹 상에 존재하는 새로운 옷 사진을 크롤링한 뒤, 기존에 학습된 딥러닝 모델을 활용해 해당 옷에 대한 속성 정보(태그)를 추가하여 저장한다.
Stimuli	일정 주기에 따라 자동 갱신
Response	옷 데이터베이스에 옷의 아이디와 해당 옷의 태그를 저장한다. 이미 데이터베이스 상에 존재하는 옷은 추가하지 않는다.
Comments	

*Table 29: New Clothes crawl description***b.13 Update Preference**

Use Case	Update Preference
Actor	User
Data	User ID, User Taste Vector Data, User Taste Vector DB
Description	스타일 메이커 페이지에서 선택한 옷들의 정보를 받는다.
Stimuli	스타일 메이커 페이지에서 옷을 스타일에 추가했을 때
Response	데이터베이스에 취향 벡터 데이터를 가져와 새로 얻은 옷들의 태그 정보를 바탕으로 기존의 사용자의 취향 벡터를 새로운 취향 벡터로 갱신하여 데이터베이스에 저장한다.
Comments	

Table 30: Update Preference description

c. Sequence Diagram

C.1 Sequence Diagram for Explore Preference

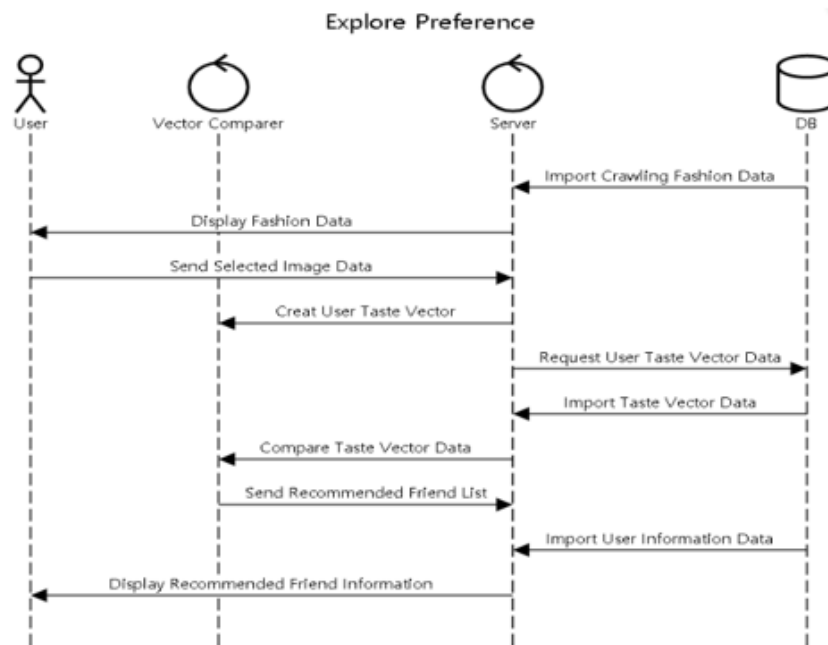


Diagram 11: Sequence diagram for explore preference

C.2 Sequence Diagram for My Page

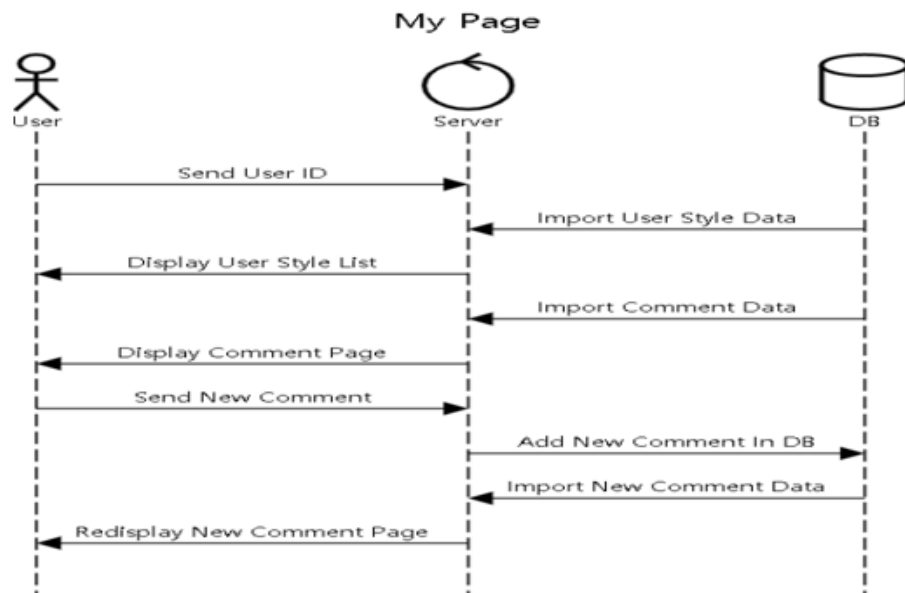


Diagram 12. Sequence diagram for My Page

C.3 Sequence Diagram for Friend's tab

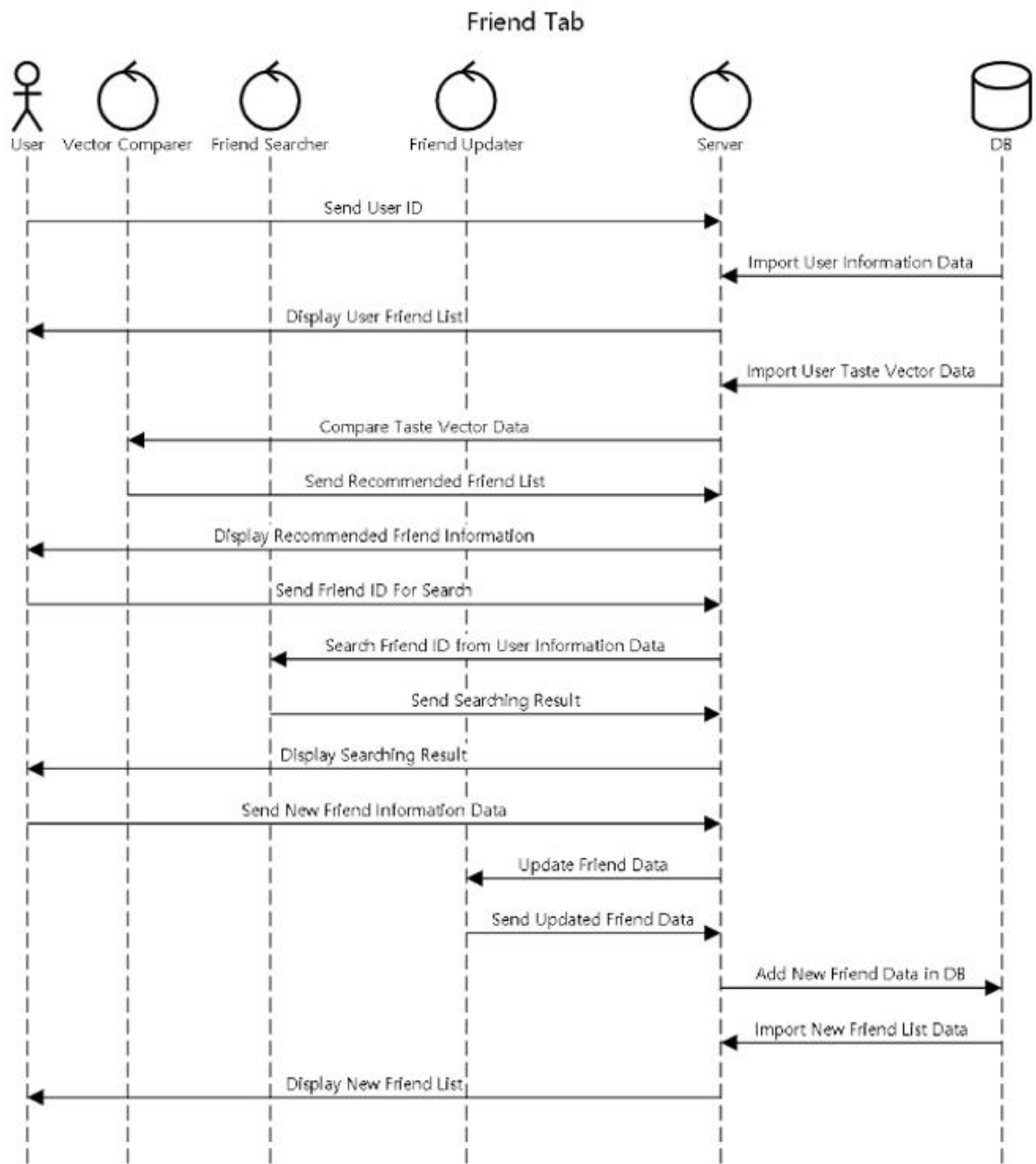


Diagram 13. Sequence Diagram for Friends' tab

C.4 Sequence Diagram for Browse Friends' Style

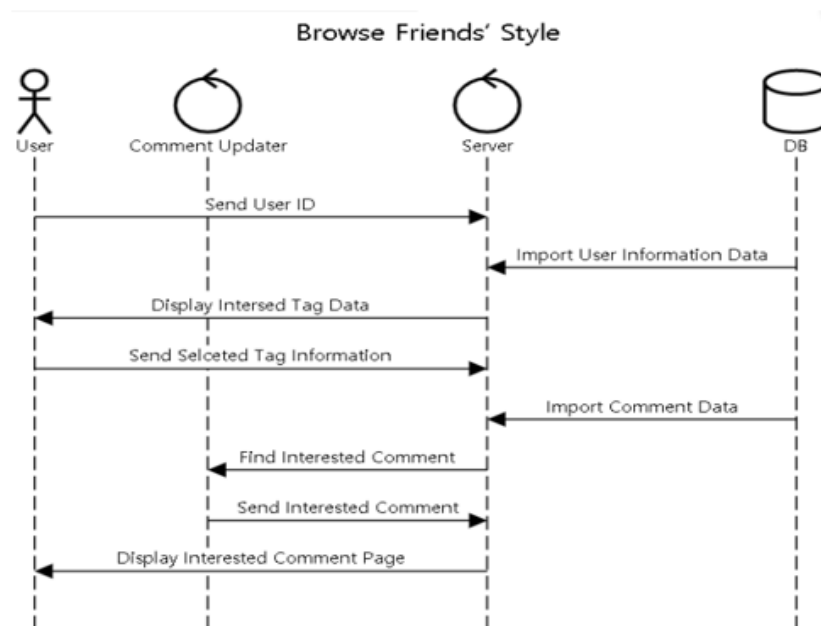


Diagram 14. Sequence Diagram for Browse Friends' style

C.5 Sequence Diagram for View New Clothes

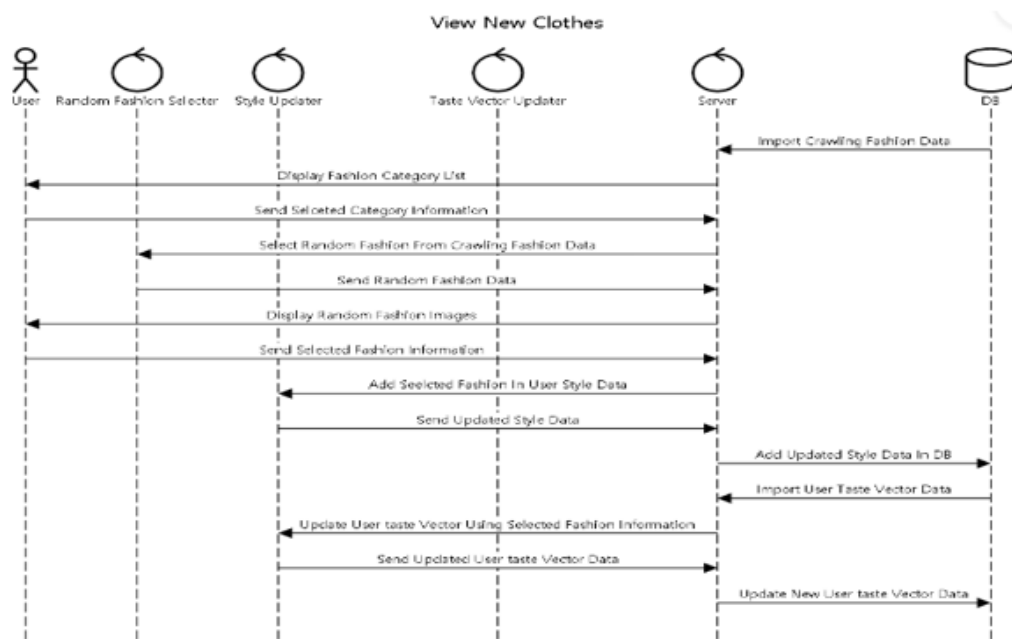


Diagram 15. Sequence Diagram for View New Clothes

7.4 Structural Model

A. User System class diagram

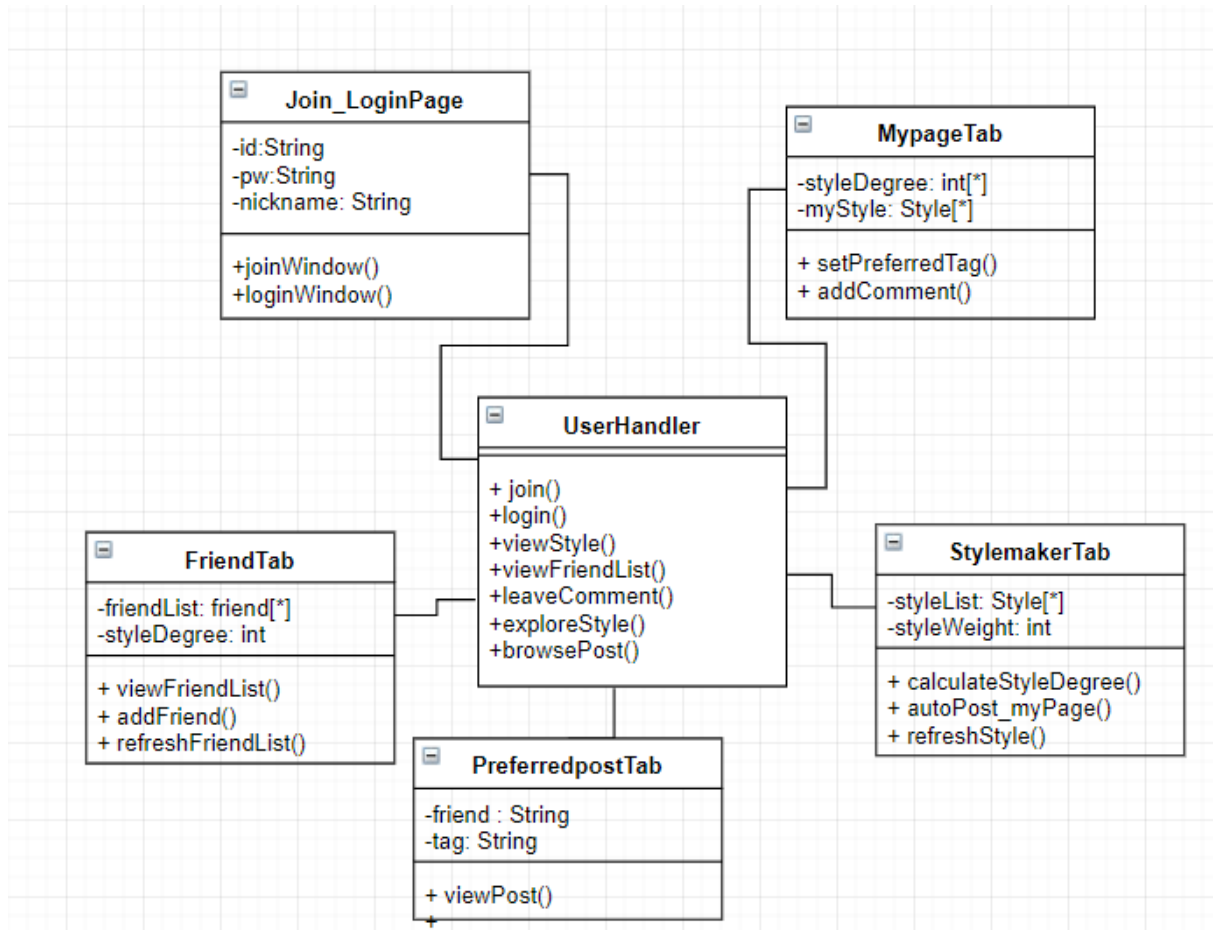


Diagram 16. User system class diagram

7.5 Behavioral Model

이 장에서는 '중년들의 패션친구'의 주요기능인 '추천(Recommendation)'과 '소통 (Communication)'에 중점을 두었다. 여기서 추천이란 취향 분석을 통한 옷 추천 및 친구 추천을 뜻하고, 소통이란 친구들과 친목을 위한 전반적인 행위를 뜻한다.

A. Data Driven Model

A.1 추천

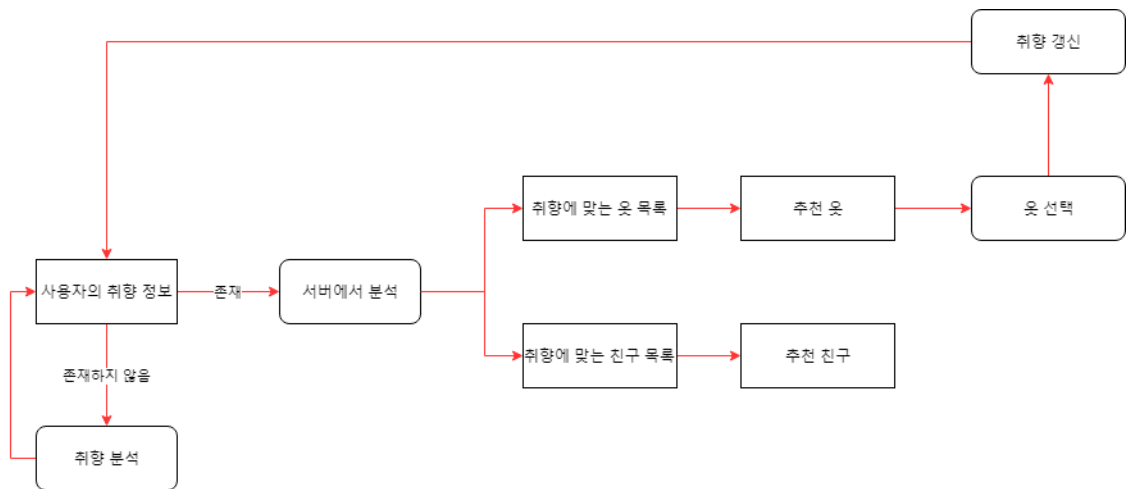


Diagram 17. Data driven – 추천

A.2 소통

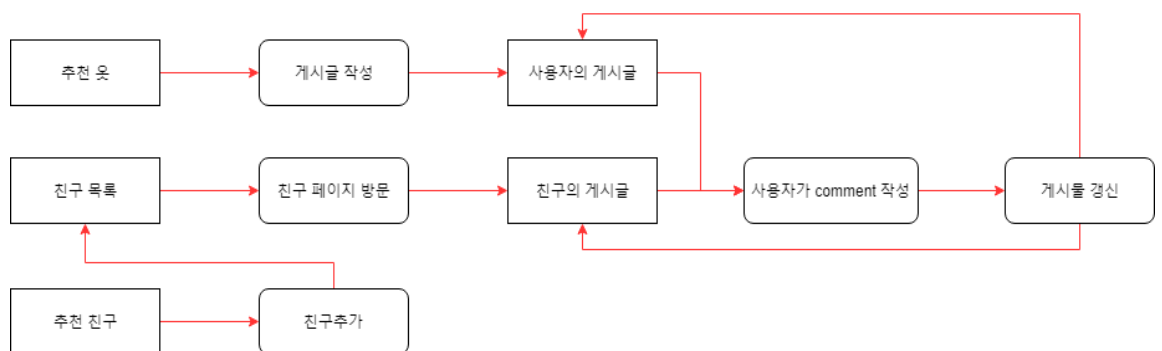


Diagram 18. Data driven – 소통

B. Event Driven Model

B.1 취향분석

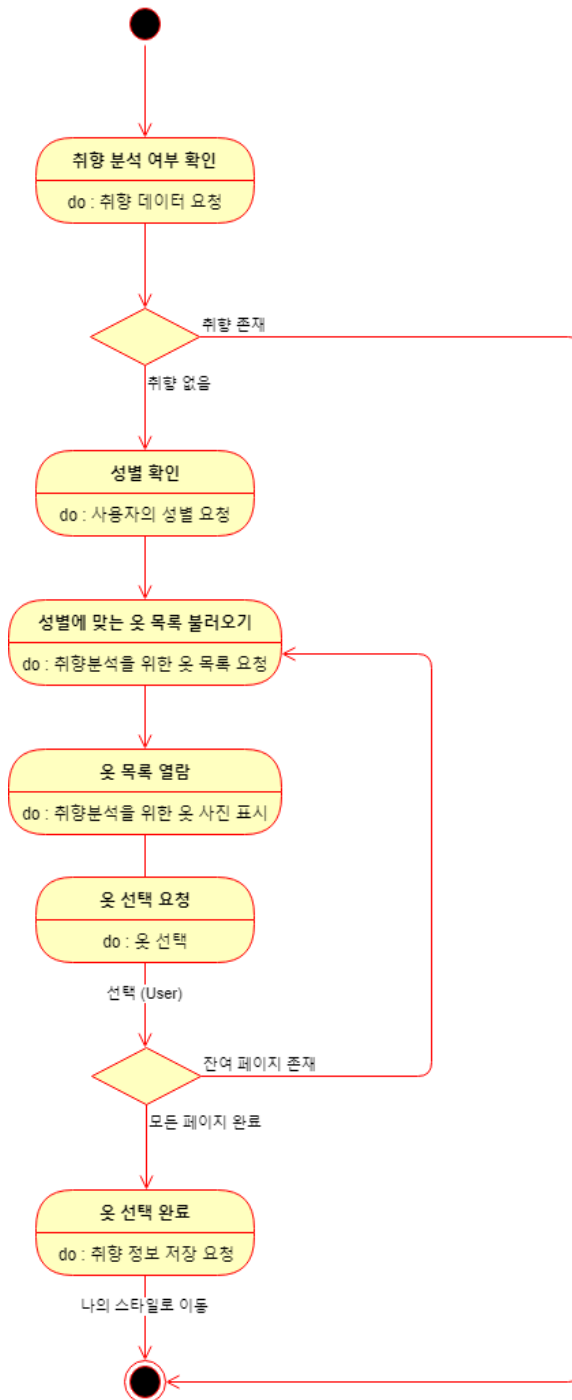


Diagram 19. Event driven – 취향분석

B.2 게시물 포스팅

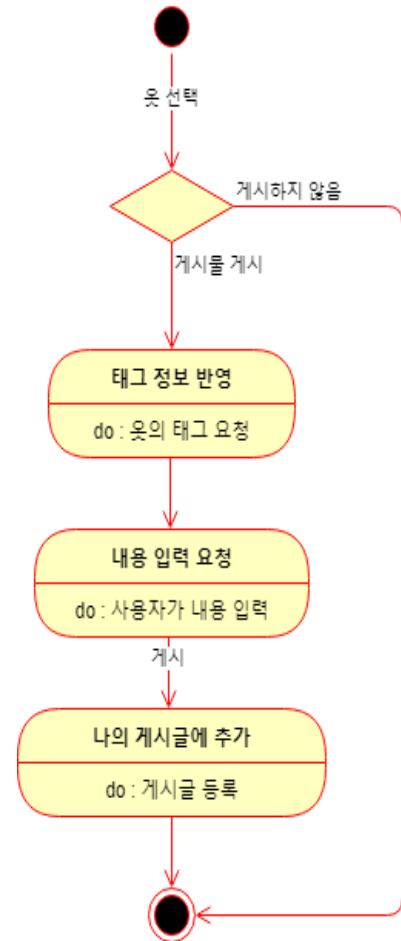


Diagram 20. Event driven-게시물포스팅

B.3 View new clothes

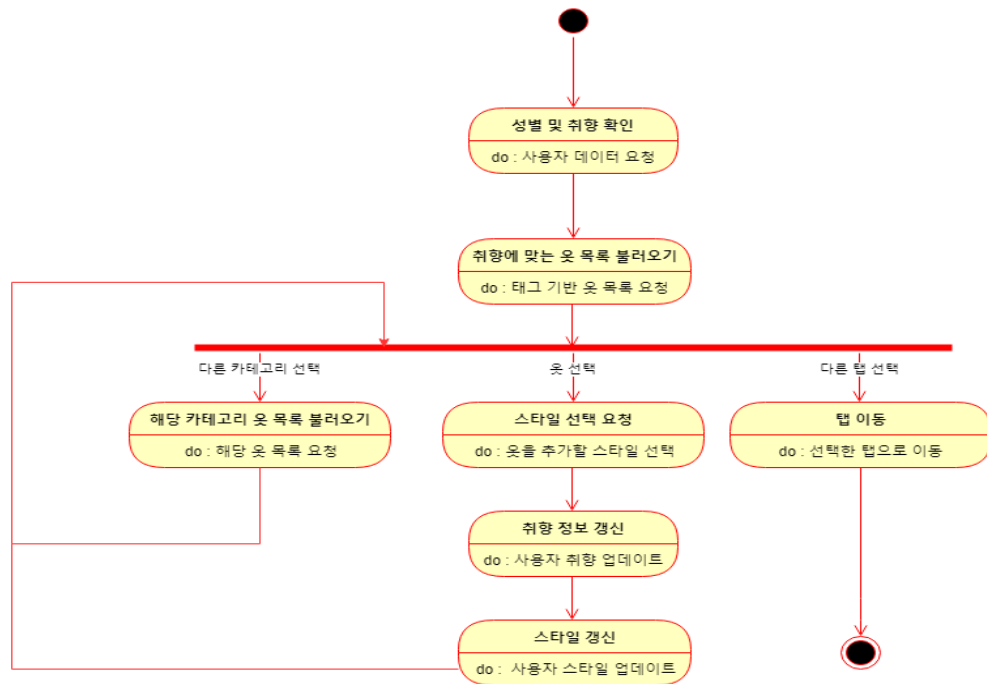


Diagram 21. Event driven – view new clothes

B.4 친구 추가

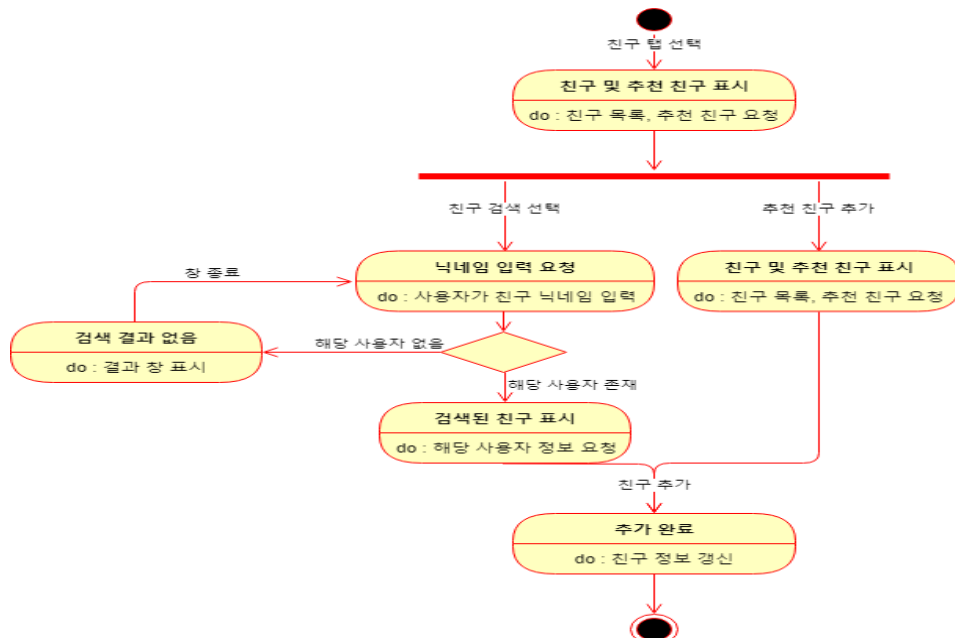


Diagram 22. Event driven – 친구추가

B.5 친구 게시물에 코멘트 남기기

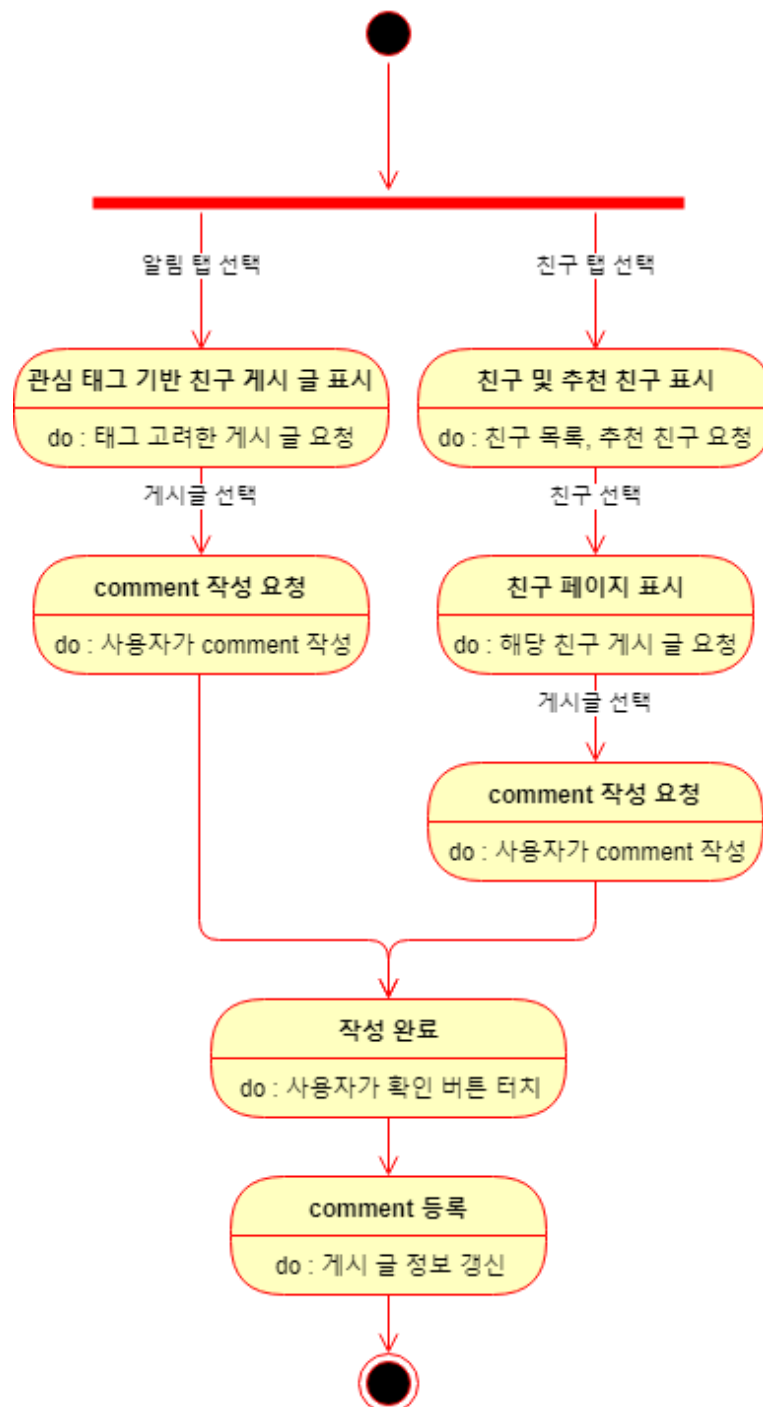


Diagram 23. Event driven – 친구 게시물에 코멘트 남기기

8. System Evolutions

8.1 Objectives

System Evolutions 에서는 시스템이 더 나아갈 수 있는 방향에 대해 설명한다. 그래서 evolution이 진행될 수 있는, 즉 더 나아갈 수 있는 방향으로 크게 카테고리화하고, 각 카테고리마다 발전될 수 있는 방향을 몇 가지로 설명한다.

그 발전 방향들마다 현재 우리 시스템이 갖추고 있지 못한 기능과 그런 이유에 대해 설명하고, 그런 부분이 해결된다면 어떤 식의 모습을 갖추는지, 어떻게 활용할 것인지를 설명하는 부분이다.

8.2 User Requirement Evolutions

a. 구매로 이어질 수 있는 시스템 구축

‘소비자들이 자신의 취향과 부합하는 옷을 다른 사람이 착용했는데, 예뻐보인다면 구매욕구가 생기지 않을까?’ 라는 점을 생각해보았다. 그런 점에서 보면 우리 어플리케이션은 여느 잡지나 패션몰보다 구매욕구를 불러일으키기에는 성공적이다. 왜냐하면 자신과 패션 취향이 비슷한 일반인이 고른 옷이기 때문에, 그 옷은 자연적으로 비슷한 취향 하에서 추천되는 옷이라고 볼 수 있다. 또한 모델이나 배우보다 일반인이 착용했는데도 예뻐보인다면 더 구매욕구가 생길 것이다. 하지만 우리 어플리케이션에서 직접적으로 구매가 일어난다면 전자결제와 같은 시스템이 도입되어야 하는데, 구현 상의 시간과 기술적 어려움으로 그런 시스템을 도입하는 데 까지나아가지는 못하였다. 또한 어떤 옷을 친구가 입었고, 그 옷을 사진을 통해 올렸는데, 그 옷이 어디 제품인지, 어떻게 구했는지를 알기가 어렵기 때문에 그 옷을 구매할 수 있는 사이트로 이어지지 못하는 못하였다.

그렇기 때문에 소비자들의 구매욕구를 실현할 수 있게 해주는 장치가 필요하다. 우리 사이트에서 바로 전자결제를 통해 구매가 이뤄질 수 있게 하려면 판매자가 있어야한다. 그래서 그 판매자가 구매하려는 사람에게 옷을 팔려고 우리 사이트를 통해 내놓을 때에 가능한 시스템이다. 그렇기 때문에 전자결제 시스템을 통해

우리 어플리케이션에서 바로 구매하도록 하는 것은 어려워보인다.

그렇지만, 패션몰에서 판매되고 있는 방대한 양의 옷들의 이미지와 정보들을 다 수집할 수 있다면, 친구가 착용한 옷 사진을 분석하여 어플리케이션의 데이터베이스에서 그와 비슷한 옷을 찾아보고, 사용자가 구매를 원한다면 비슷한 옷들을 리스트업 시킨뒤, 그 옷들은 어느 쇼핑몰 사이트에 있는지를 보여주고, 그 쇼핑몰로 이어질 수 있게 한다면 좋을 것이다.

b. 스타일마다 친구 추천

“중년들의 패션친구”에서 가장 핵심기능은 자신이 고른 옷을 기반으로 하여, 취향이 비슷한 다른 사용자를 추천해주는 것이다. 현재 “중년들의 패션친구”에서 친구를 추천할 때는 사용자가 고른 모든 옷을 기반으로 해서 친구를 추천하게끔 하였다.

그런데 사용자가 항상 한 가지의 스타일만을 고집하는 것이 아니다. 그래서 “중년들의 패션친구”에서도 1개가 아니라 복수의 스타일을 모을 수 있게끔 스타일란을 여러 개 두었다. 그래서 각각의 스타일란은 사용자가 같은 스타일로, 일관적으로 선택한 옷들이 있다. 그렇기 때문에 사용자는 정장이면 정장, 캐주얼이면 캐주얼에 맞게 친구가 추천되어야 하는 것이 합리적이다. 왜냐하면 정장과 캐주얼은 서로 다른 스타일로서 취향벡터를 구할 때 스타일이 섞이기 때문이다. 그래서 그 스타일에 맞게 특화되어 친구가 추천이 안될 수도 있다. 본래 “중년들의 패션 친구”의 취지에 부합하려면 스타일마다 그 스타일에 부합하게끔 친구를 추천해주는 것이 맞다. 왜냐하면 사용자가 서로 아예 다른 스타일을 섞어서 옷을 입지는 않기 때문에, 캐주얼이면 그 캐주얼에 맞는 사용자들을 추천해주면 된다. 즉 캐주얼인데 정장까지 고려해서 친구를 추천해줄 필요는 없다는 것이다.

현재 “중년들의 패션친구”에서는 완전한 시제품이 아니기 때문에 수요가 얼마나 있을지 가늠이 되지 않는다. 스타일 별로 친구를 추천해주기 위해서는 어느 수준 이상의 사용자가 확보되어야 한다. 그렇지만 현재로서는 그만큼의 사용자를 확보

하기가 쉽지 않으므로, 스타일별로그가 아니라, 사용자가 고른 모든 옷들을 기반으로 하여 옷을 고를 수 있게 설정하였다. 따라서 evolution 단계에서는 사용자가 고른 스타일 별로 취향벡터를 구해서 그것을 기반으로 친구를 추천해줄 수 있는 기능이 추가될 것이다.

c. 비슷한 스타일의 모델, 배우의 정보 포함

“중년들의 패션친구”는 사용자와 비슷한 처지의 다른 일반인이 옷을 착용했을 때의 그 친근감 같은 것을 주기 위함도 그 목적이다. 유명 패션 브랜드나 쇼핑몰 사이트는 모델이 옷을 착용한 사진을 게시한다. 이 때 몇몇의 일반인들은 저 옷이 예뻐보이는 이유는 저 모델이 예쁘고 잘생겼거나, 몸이 좋아서 옷핏(fit)이 잘 살기 때문이라고 생각한다. 이 생각에 매우 전적으로 동의하면서, 일반인이 입었을 때도 예뻐보여야 사용자가 입었을 때도 예뻐보일 확률이 높다고 생각하였다. 그래서 “중년들의 패션친구”를 처음 생각했을 때는 패션 모델이나 배우들은 고려하지 않았다.

패션 모델이나 배우들이 일반인에 비해 비현실적인 fit을 가지고 있기 때문에 아무 옷이나 걸쳐도 그 옷이 괜찮아 보이는 것은 사실이다. 그렇지만 패션 모델이나 배우들은 일반인들에 비해 옷을 코디하는 것에 대해서 더 신경을 많이 쓰는 사람들이고, 보여지는 것이 직업이기 때문에 좀 더 센스있는 스타일을 갖추려고 노력한다. 그렇기 때문에 일반인들 중에서도 패션 모델이나 배우들을 참고하고 싶은 사람들이 충분히 있을 것이라고 예상되었다. 비록 그들과 같은 fit은 아니더라도, 그들처럼 센스있게 코디하는 법을 참고하고 싶은 것이다. 이 때도 모든 배우나 패션 모델들을 추천하는 것이 아니라, 그 중에서도 사용자와 스타일이 비슷한 사람들만을 선별적으로 선택하여 추천해준다면 충분히 수요가 있을 것이라고 생각한다.

d. 소통 기능 강화

중년들이 젊은 층들에 비해 SNS 활동을 활발하게 하지 않는다. 페이스북이나 인스타그램과 같은 SNS는 젊은 층들의 것이라 할 수 있다. 그렇지만 중년층들이 소통하고 싶어하지 않아서 SNS 활동을 안하는 것은 아니다. 왜냐하면 카카오톡 사용시간을 비롯하여 스마트폰 사용시간은 중장년층이 매우 높은 것으로 집계된 것으로 보아, 그런 소통활동을 싫어하는 것은 아니기 때문이다. SNS 활동 같은 것은 젊은 층들이나 하는 것인데, 자신들의 연령을 고려한 사회적 시선 때문에 주저할 수도 있고, 또 SNS 활동이 중장년층이 사용하기에는 다소 어려울 수도 있다.

그래서 “중년들의 패션친구”에서는 중년들만의 커뮤니티를 좀더 활성화하기 위한 목적도 있다. 중년들만이 사용하기 때문에 사회적 시선을 굳이 생각하지 않아도 되고, 중장년들층 맞춤으로 어렵지 않게 사용할 수 있도록 구성하는 것이 중년들의 패션친구의 장점이다. 그래서 “중년들의 패션친구”는 그런 점들을 옷을 통하여 소통이 이루어지게끔 하는 구조이다.

그런데 “중년들의 패션친구”에서는 소통할 수 있는 방법이 상대방 (친구)가 옷을 골랐을 때 작성되는 게시글에 대해서 comment를 남기는 정도로만 제공되고 있다. 이 때 comment는 그 사람과 친구인 사람들은 그 사람의 페이지에 들어와서 게시물들을 다 열람할 수 있고, 그 게시물의 댓글까지도 다 확인할 수도 있다. 이렇게 한 이유는 “중년들의 패션친구”라는 어플리케이션에서는 서로가 어떤 옷을 입었는지 정도의 공유만을 생각했기 때문에 개인적인 얘기는 잘 이루어지지 않을 것이라는 판단이 들었기 때문에 그렇게 생각하였다. 꽤 공적인 공간이기 때문에 좀 더 친밀감있고, 사적인 얘기를 할 수 없을 수도 있다. 중년들 간의 소통의 기능이 강화되려면 좀 더 친근감있는 얘기를 할 수 있는 공간이 마련되어야 한다. 단순한 옷 정보에서 더 나아가서 정말 커뮤니티 적인 역할을 수행하기 위해서는 채팅 기능과 같이 좀 더 개인적으로 소통을 진행할 수 있는 공간이 필요하다.

8.3 Hardware Evolutions

a. FX Mirror

FX mirror 라는 것은 증강현실 기술을 기반으로한 3D 가상피팅 시스템을 말한다. 따라서 사용자는 번거롭게 원하는 옷들을 다 입어보지 않아도 되고, 이 기술을 적절히 활용한다면 인터넷 쇼핑몰에서 옷을 구입했을 때의 단점을 극복할 수 있을 것이다.

보통 사람들이 옷을 살 때 인터넷 쇼핑몰이 할인도 더 많이 되고, 옷 종류도 많고, 그러한 옷들을 힘들게 돌아다니지 않고도 한 눈에 잘 살펴볼 수 있음에도 불구하고 백화점이나 매장에 가서 옷을 사는 데에는 다 이유가 있다. 바로 옷을 입어보지 않고 설불리 구입했다가 자신이 생각했던 옷핏(fit)이 다르거나, 옷감, 질감이 다른 경우가 있기 때문이다. 사이즈가 잘 안맞는 경우에는 옷을 입지도 못하기 때문에 사람들은 직접 입어볼 수 있는 매장에서 옷을 구매하는 것을 선호한다.



그런데 이 FX Mirror 기술을 잘 활용하여 “중년들의 패션 친구”에서 활용한다면 사용자가 직접 옷을 입어보지 않고, 자신이 이 옷을 착용했을 때의 느낌을 알 수 있기 때문에 좀 더 관심있게 상대방이 고른 옷들을 살펴볼 것이다.

물론 사용자의 집에 FX mirror 가 있으면 좋겠지만, 그렇지 않은 경우에 핸드폰을

통해서 FX mirror 와 같은 UI를 제공하고, 3D 영상 처리기술을 이용하여 정확하게 사용자의 신체 사이즈를 측정하게 된다면 사용자의 수중에 있지 않은 옷들에 대해서 가상으로 입어보고 구매할지 말지를 정할 수 있게 된다. 그래서 “중년들의 패션친구”에서도 친구가 입는 옷이 어떤 옷이고, 그 옷을 자신이 입게 되면 어떨지가 핵심이기 때문에 이 기술을 도입한다면 좀 더 사용자들에게 완성도 있는 솔루션을 제공할 수 있다.

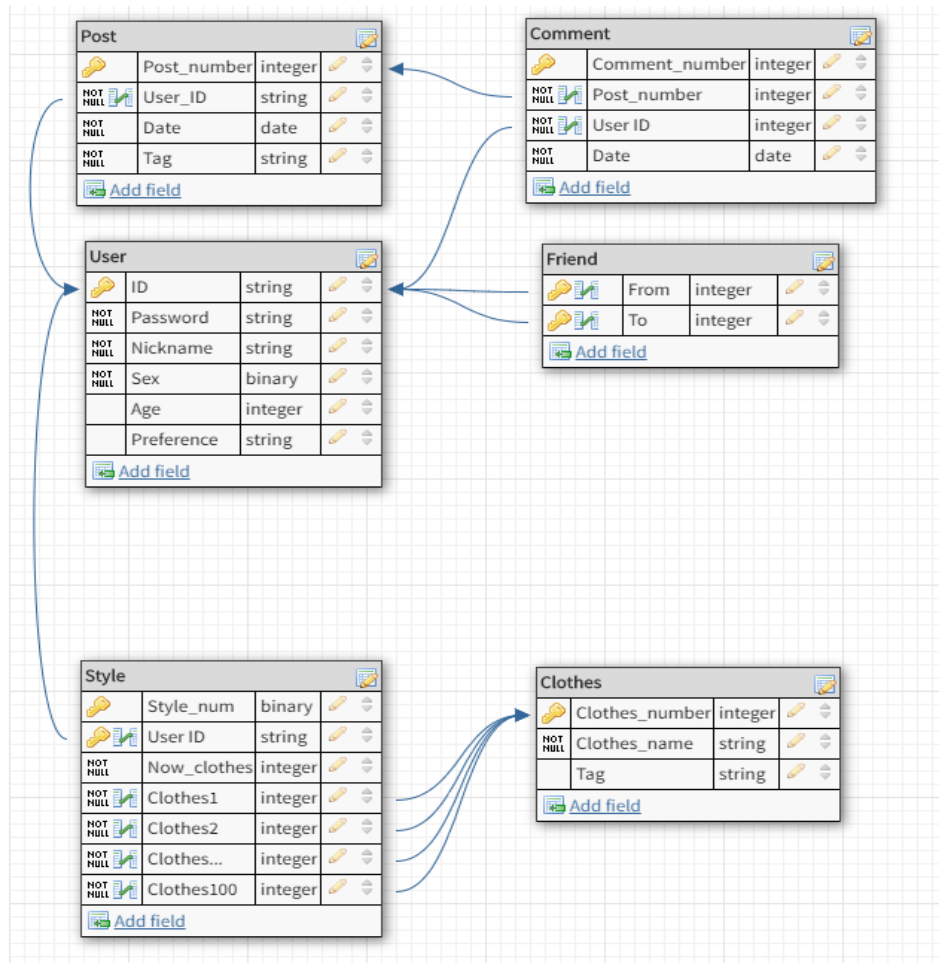
9. Appendices

9.1 Objectives

Appendices 에서는 개발되는 시스템에 사용되는 기술이나, 그 시스템에 대해서 독자의 이해를 돕기 위해서 좀 더 자세하게 기술한 부분이다. 우리 시스템을 가능하게 하기 위해서 필요한 데이터베이스 구성과 “중년들의 패션친구”의 핵심기능인 이미지 인식을 위한 딥러닝 모델과 추천 시스템에 대하여 설명하겠다. 그리고 우리 프로젝트가 사용한 개발 process에 대해서도 간략하게 설명하겠다.

9.2 Database Architecture

‘중년들의 패션친구’를 구현하기 위해 필요한 데이터베이스의 요구사항을 서술한다. 다음은 데이터베이스 내의 각 테이블 구조를 나타낸다. ‘중년들의 패션친구’ 어플리케이션은 서버를 통하여 데이터베이스에 요청을 보내고 데이터를 받게 된다.



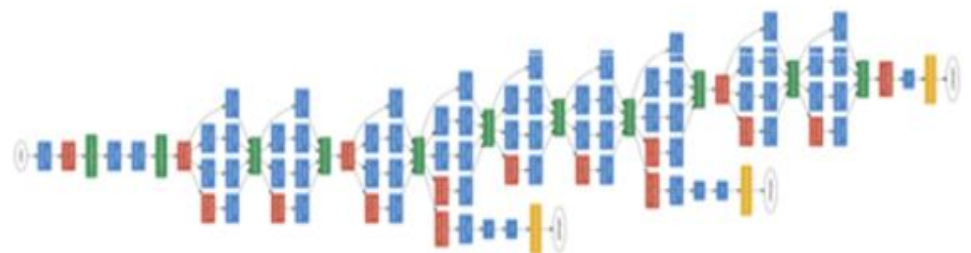
9.3 Transfer model, Deep Learning Model

“중년들의 패션친구”에서는 옷을 제대로 인식할 수 있어야만 그것을 기반으로 유사한 취향 스타일의 친구를 추천해주기 때문에 정확한 이미지 인식기술은 필수적이다. 그렇기 때문에 딥러닝 기술을 이용하여 이미지 분류를 하려고 한다. 그런데 딥러닝 자체도 연산시간이 매우 오래걸리지만, 어느정도의 성능을 보장하기 위해서 layer가 많은 좀 깊은 모델을 사용하게 된다면, 더 연산량이 증가하여 GPU 같은 장비가 있지 않으면 쉽게 접근하기 어렵다.

그래서 “Transfer Learning”을 통하여 최종모델을 학습시킨다. Transfer Learning 이라는 것은 기존의 만들어진 모델을 사용하여 우리의 목적에 맞는 모델을 만드는 것을 말한다. 이 때 이미 모델이 만들어져 있다는 것은 어떤

특정결과를 도출하기 위해 layer마다 그 가중치들이 다 적합이 되어있는 상태를 말한다. 따라서 우리의 데이터에 학습을 시키기 위해서는 최종 Fully Connected Layer만을 다시 재적합 시켜서 우리의 데이터에 맞게끔 결과를 도출할 수 있게끔 조정하는 것이다. 쉽게 말하면 사과를 깎는 기계가 있는데, 우리의 목적은 배를 깎는 것이므로, 배를 받아들일 수 있게끔 사이즈만 조정해주고 나머지는 그대로 사용하여 사과를 깎는 기계로 배를 깎는 경우와 유사하다고 말할 수 있다. 이는 처음부터 배를 깎는 기계를 만드는 것보다 사과를 깎는 기계에서 배를 깎을 수 있게끔만 살짝 조정해주는 것이 훨씬 효율적이라고 할 수 있다.

또한 우리의 데이터 또한 이미지를 인식하여 1000개 남짓한 label에 매칭시키는 것이므로 VGG Net 이 학습되었던 ImageNet과 유사한 환경이라고 할 수 있다. 그래서 transfer learning을 사용하는 것이 훨씬 효율적이다.



VGGNet



VGG Net 구조

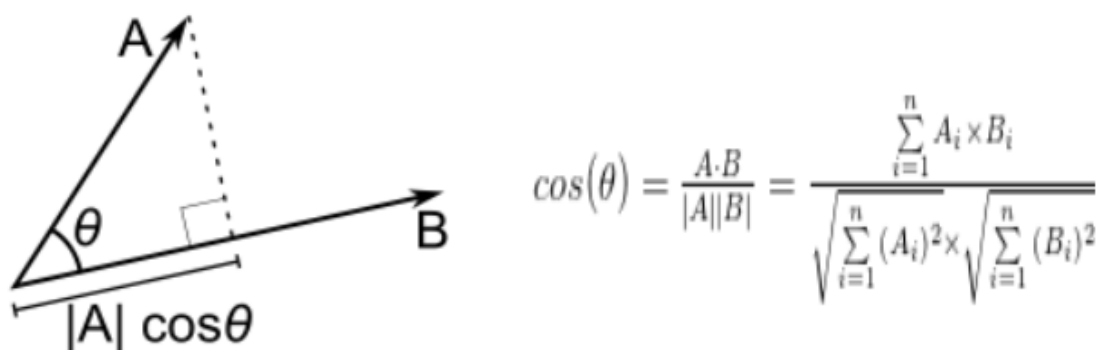
여러 다른 좋은 모델들이 있지만, VGG16 모델이 다른 모델들에 비해 간단하고, 레이어 수가 적기 때문에 가장 많이들 활용된다. 그래서 사용하게 되었다. Googlenet은 1개의 layer 안에 여러 개의 convolution layer가 병렬적으로 존

재하여 1 layer 안에서도 비선형적인 feature map을 더 잘 추출할 수 있다. 그에 반해 VGG16 모델은 기존의 convolution layer를 깊게 쌓은 단순한 모델이다. 단순함에도 불구하고 그 성능이 Google Net 과 별로 차이 나지 않는다.

즉 이미지를 분류하기 위해서 train 된 모델을 사용하는 것이고, 그 때의 가중치를 사용하는 것이다. 그 VGG16 모델에다가 우리의 데이터 셋에 맞게 예측할 수 있게끔 layer를 추가하는 것이다. 그래서 이미 트레인 된 모델에서 우리가 추가한 레이어에 대한 가중치를 적합 시킬 수 있게끔 우리 데이터로 학습을 시킨다.

그러면 처음부터 모델을 구성해서 가중치들을 적합시킬 때보다 시간이 훨씬 줄어들게 된다.

9.5 Cosine Similarity



유사도를 구하는 방법 중의 하나로서, 두 벡터 사이의 각을 코사인(cosine)으로 구해서 그 각을 기준으로 유사도를 판단하는 방법이다. 위의 그림은 2차원 벡터를 기준으로 하여 그 유사도를 구한 것이다. 가시적으로는 3차원까지밖에 표현하지 못하지만, 10, 100차원의 벡터도 유사도를 구할 수 있다.

유사도를 구하는 방법에는 cosine similarity 말고도 Manhattan Distance, 마할라노비스 거리 등의 다양한 방법들이 있다. 그 중 대표적인 거리를 구하는 방법 중의

하나가 Euclidean distance이다. 어떤 데이터들 간의 유사성을 그 데이터 간의 선형적인 거리의 길이로 나타낸 것이다.

단순히 두 벡터 사이의 차이를 Euclidean distance로 나타낼 경우에는 두 벡터의 절대적인 차이에 집중하게 되는 것이고, cosine similarity로 계산한다면 전체적인 비율을 통해 유사도를 결정한다. 예를 들어 a,b라는 두 항목에 대해서 가) 제품은 (1,1) 만큼 요소를 가지고 있고, 나) 제품은 10,10만큼 요소를 가지고 있다. 그에 반해 다)제품은 5,0만큼 요소를 가지고 있다. 이 때 Euclidean 거리로 파악해본다면, 가) 제품과 다)제품이 유사하다고 나올 것이다. 만약에 a,b 두 항목이 전혀 상반되는 항목이라면, 가)는 그것을 반반 가지고 있는 제품이지만, 다) 제품은 b항목이 전혀없는, a항목 친화적인 제품인 것이다.

따라서 상식적으로 보았을 때, 가)제품과 나)제품이 유사하다고 보는 것이 합리적이다. 왜냐하면 두 제품은 a,b 두 항목에 대해 반반 좋아한다고 했다는 성격이 같기 때문이다. 그럼에도 불구하고 Euclidean distance로 계산하게 된다면 그 점을 고려하지 못하고, 절대적인 거리가 가)에서 나)가 더 가깝기 때문에 가)와 나)가 비슷하다는 결론이 도출된다.

따라서 "중년들의 패션친구"에서는 단순히, 사용자가 옷을 고른 전체의 횟수에 의해서 취향 vector가 영향을 받으면 안된다. 전체 횟수 중에서 그 취향의 옷이 차지하는 상대적인 비율을 봐야한다. 그렇기 때문에 Euclidean distance 보단 cosine similarity로 계산하는 것이 더 합리적이다.

10. Index

table 1 version table.....	11
table 2 용어설명 – 어플리케이션상.....	18
table 3 용어정리-시스템상.....	19
table 4. 용어정리- 패션.....	19
table 5. login with kakao Function.....	34
table 6. sign up function.....	35
table7. sign in Function.....	36
Table 8. Explore Preference function.....	37
Table 9. Configure My style function.....	38
Table 10 Friends ' tab function.....	39
Table 11. Add friends function.....	40
Table 12. Find a Friend function.....	40
Table 13. New Clothes Crawl.....	41
Table 14. Browse Friends' style.....	42
Table 15. Write Comment.....	43
Table 16. View New Clothes.....	43
Table 17. Update Preference.....	44
Table 18: Login With Kakao Description.....	63
Table19 : Login Description.....	64
Table 20 Sign up Description.....	65

Table 21. Explore Preference Description.....	65
Table 22. Cofigure style Page Description.....	66
Table 23. Friend tab Description.....	66
Table 24 : Add Friend Description.....	67
Table 25 : Find Friend description.....	67
Table 26: Browse Friends' style description.....	68
Table 27: View New Clothes description.....	69
Table 28: Write Comment description.....	69
Table 29: Update Preference description.....	70
Table 30: Update Preference description.....	70
diagram1. explore Preference system.....	26
diagram2 . My page system.....	28
diagram3. Friend system.....	29
Diagram4 Browse Friends' Style System.....	30
Diagram5 View New Clothes System.....	31
Diagram6. New Clothes Crawl System.....	32
Diagram7 Friend Page System.....	33
Diagram 8. Context Model.....	61
Diagram 9. Process diagram.....	61

Diagram 10. Use case.....	62
Diagram 11: Sequence diagram for explore preference.....	71
Diagram 12. Sequence diagram for My Page.....	71
Diagram 13. Sequence Diagram for Friends' tab.....	72
Diagram 14. Sequence Diagram for Browse Friends' style.....	73
Diagram 15. Sequence Diagram for View New Clothes.....	73
Diagram 16. User system class diagram.....	74
Diagram 17. Data driven – 추천.....	75
Diagram 18. Data driven – 소통.....	75
Diagram 19. Event driven – 취향분석.....	76
Diagram 20. Event driven-게시물포스팅.....	76
Diagram 21. Event driven – view new clothes.....	77
Diagram 22. Event driven – 친구추가.....	77
Diagram 23. Event driven – 친구 게시물에 코멘트 남기기.....	78