

인수인계

2018년 2월 2일, 연구개발부 이하영

목차

인수인계

목차

프로젝트 관리

1. 소스 코드

1.1. 계정 정보

2. 데이터

개발 업무

1. WebTex3D

1.1. 계정 정보(공통)

1.2. Tex3DW

1.3. 시뮬레이션 모듈

1.4. 설치 완료 후, 시뮬레이션이 되지 않을 때 대응

2. WebTex3D 리뉴얼

2.1. 리뉴얼 내용

2.2. 추가개발내용

3. WebTex3D Mobile

3.1. 서버 정보

3.2. 담당 업무

3.3. 프로젝트 설명

3.3.1. 리뉴얼 적용

3.3.2. 서버 선택 기능 관련

3.3.3. 레이어매칭 적용

3.3.3.1. 개발 방식

3.3.3.2. 주의 사항

3.4. 위험 요소

4. Fukushin Dress

4.1. 참여 인원

4.2. 담당 업무

4.3. 프로젝트 설명

5. Sunregent

5.1. 서버 정보

5.2. 계정 정보

5.3. 참여 인원

5.4. 담당 업무

5.5. 프로젝트 설명

5.5.1. 카테고리 구조

5.5.2. Product 검색

5.5.3. Fabric 검색

5.5.4. 레이어 구조

6. Matsuzaki

6.1. 참여 인원

6.2. 담당 업무

6.3. 프로젝트 설명

6.3.1. 카테고리 연결

6.3.2. Fabric 검색

7. Elite

7.1. 서버 정보

7.2. 참여 인원

7.3. 담당 업무

7.4. 프로젝트 설명

7.4.1. 데이터베이스 구조

7.4.2. 용어 설명

7.4.3. 데이터 업로드

 네이밍 규칙

 업로드 순서

7.5. 시뮬레이션

 초기 화면

 Style 선택

 Fabric 선택

 시뮬레이션 원리

8. Smart

8.1. 서버 정보

8.2. 참여 인원

8.3. 담당 업무

8.4. 프로젝트 설명

8.4.1. 데이터베이스 구조

8.4.2. 용어 설명

8.4.3. 데이터 업로드

 네이밍 규칙 & 업로드 데이터

 최종 결과물

 업로드 순서

8.4.3. 주문 관리

 8.4.3.1. add_option

 8.4.3.2 estimate

 8.4.3.3 contract

 8.4.3.4 pamphlet

8.4.4. 관리자

 8.4.4.1. 디자인정보 > 옵션 관리

 8.4.4.2. 영업현황 > 매출 현황

9. WebTex3D Shirts

기타 업무

1. 홈페이지 관리

1.1. 관리자 계정

1.2. 입사자/퇴사자 계정 관리

1.3. 문의 사항 답변

1.4. 자격증 업무

 1.4.1. 자격 검정 시험 등록

- 1.4.2. 자격증 응시자 등록
- 1.4.3. 배포판 프로그램 문의 관련
- 1.5. 학생용 특가 프로그램(ASP)
 - 1.5.1. 프로그램 실행 관련 문의
 - 1.5.2. 이메일 수신 문제
 - 1.5.3. 환불 요청 대응
 - 1.5.4. 유플러스 전자결제
- 1.6. 홈페이지 프론트엔드 관리
 - 1.6.1. 연혁 수정
 - 1.6.2. 팝업 수정
- 2. Texclub 서버 관리
 - 2.1. 서버 계정 정보
 - 2.2. 서버 접속
 - 2.2.1. 원도우
 - 2.2.2. Mac
 - 2.3. 메일 서버 관리
 - 2.3.1. 메일 계정 관련
 - 2.3.2. 메일 서버 장애 대응
- 3. 대량메일 발송
 - 3.1 계정 정보
 - 3.2 주소록
 - 3.2.1. 주소록 그룹관리
 - 3.2.2. 주소록에 등록할 파일 생성 방법
 - 3.3 TexPro 100배 활용하기
- 4. 가비아
 - 4.1. 계정 정보
 - 4.2. 도메인 관리

프로젝트 관리

1. 소스 코드

BitBucket에서 프로젝트별로 관리가 되고 있습니다. 비공개 모드는 최대 5명까지 무료로 사용할 수 있으며, 초과가 되면 결제가 필요합니다.

1.1. 계정 정보

ID : webmaster@texclub.com PW : yw_rhksflwk/0

2. 데이터

- 웹팀 소유의 외장하드에 프로젝트별로 db_dump 파일과 data 파일이 저장되어 있습니다.

개발 업무

1. WebTex3D

webtex3d는 영우의 기본 솔루션으로, 고객의 요구사항에 맞게 커스터마이징 하여 판매가 이루어집니다.

1.1. 계정 정보(공통)

- ID : admin
- PW : duddn3d

1.2. Tex3DW

WebTex3D에서 사용될 데이터를 업로드 하는 응용 프로그램입니다. 이 프로그램에서 작업하는 데이터의 사이즈는 `webtex3d/media/data/webtex3d/sim.ini` 파일의 Size에 입력한 값과 동일해야 합니다.

- 기본적으로 stylefor.us를 포함한 WebTex3D 사이즈는 **858x897 pixels** 입니다.
- mobile.stylefor.us의 사이즈는 **1080x1780 pixels** 입니다.
- 업로드 시 위에 명시한 사이즈에 맞게 작업이 되지 않았다면, 웹에서 정상적으로 보이지 않거나 오류가 발생할 수 있습니다.

1.3. 시뮬레이션 모듈

product와 fabric의 파라미터를 받아 시뮬레이션 결과 이미지로 돌려주는 모듈입니다. 시뮬레이션 모듈은 웹 인증으로 관리가 됩니다. 서버에 프로젝트가 설치된 후, 첫 시뮬레이션을 실행하면 웹 인증창이 뜨는 데(또는 `webtex3d/media/data/webtex3d/Tex_Sim.exe` 파일을 실행), 알맞은 인증 정보를 입력해 주고, 이메일 인증까지 하면 웹 인증이 완료되며, 시뮬레이션 모듈을 사용할 수 있습니다.

웹 인증에 관련된 정보는 texclub 홈페이지의 고객관리시스템 > 웹 인증에서 확인 할 수 있습니다.

1.4. 설치 완료 후, 시뮬레이션이 되지 않을 때 대응

- 프로젝트 설치 경로와 setting 파일 내의 경로가 동일한지 확인합니다. WebTex3D는 많은 부분이 하드 코딩으로 되어 있어서 아래의 모든 파일을 확인해주어야 합니다.
 - urls.py
 - settings.py
 - wsgi.py
 - sim.ini
- `media/data/simulation/(simulation.id/10000)` 폴더가 존재하는지 확인합니다.
- 작업관리자를 열어 tex3d_sim 프로세스를 종료한 후, 다시 시뮬레이션을 실행하세요.

2. WebTex3D 리뉴얼

<http://www.stylefor.us>

155서버의 `C:\var_w3d\www`에 설치되어 있습니다.

기존의 WebTex3D를 새롭게 개편한 버전입니다. 추가되면 좋을 기능과 불편한 사항 등 회의를 통해서 다른 팀의 의견을 수렴하였고, 그 결과로 나온 기획을 토대로 진행이 되었습니다.

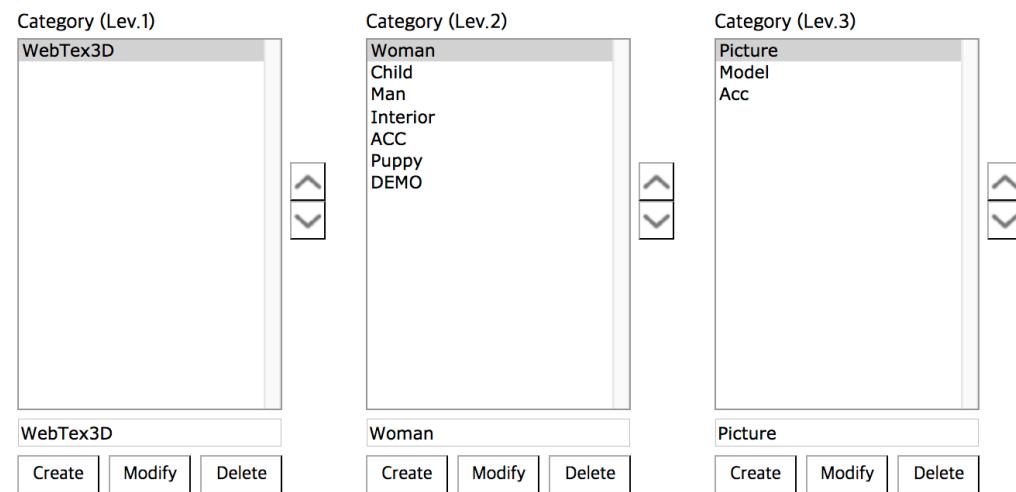
webtex3d 프로젝트를 포크하여 front-end만 변경하는 방식으로 진행하였습니다. 리뉴얼 전의 webtex3d는 git의 webtex3d에서 관리되고 있으며, 리뉴얼 버전은 webtex3d-new에서 관리가 되고 있습니다.

리뉴얼 전의 webtex3d는 <http://www.demo.stylefor.us>에서 확인할 수 있습니다.

2.1. 리뉴얼 내용

- 기존 기능과 달라진 사항
 - 전체적인 UI
 - product와 fabric의 카테고리의 레벨 구조

기존에는 1, 2, 3 레벨을 모두 사용하였지만, 리뉴얼 기획 단계에서 UI 구성이 어렵고, 2단계가 적합하다고 결정이 되었습니다. 따라서, 리뉴얼 버전에서는 1레벨은 부모 카테고리의 의미로만 두며, 뷰어 페이지에서 보여지는 카테고리는 1레벨 카테고리의 첫 번째 카테고리에 속한 2, 3 레벨의 카테고리입니다.



- product와 fabric의 선택 방식 변경

모바일, 태블릿 환경에서는 드래그 이벤트가 불편하므로, 클릭 이벤트로 변경되었습니다.

- 추가된 사항
 - 시뮬레이션 초기화
 - 시뮬레이션 결과 이미지의 구글, facebook, twitter, 카카오톡 공유 기능
 - 최근 사용한 원단의 캐시 기능
 - 원단 확대/축소 기능
 - 반응형 웹

2.2. 추가개발내용

- 레이어매칭 기능
 - 박진 사원이 개발한 내용이며, `/viewerMatching` 으로 접근할 수 있습니다.
 - 기존의 시뮬레이션은 jpg 이미지로 리턴이 되는데요. png 이미지로 리턴이 되는 모듈을 추가적으로 구성하여 시뮬레이션이 실행되면 데이터베이스에 png 이미지를 함께 저장합니다.
 - 기본 뷰어 페이지에서는 시뮬레이션 결과 이미지로 jpg를 사용하며, 레이어 매칭 페이지에서는 png를 사용합니다.

3. WebTex3D Mobile

<http://mobile.stylefor.us>

국책과제인 DPing의 서버로 사용되는 WebTex3D입니다. DPing 개발자인 신승민 사원과 박진 사원이 진행하던 프로젝트이며, 박진 사원의 퇴사로 인하여 인수인계를 받아 처리한 사항들이 있습니다. 추가적인 내용은 박진 사원의 인수인계 문서를 참고해주세요.

3.1. 서버 정보

가비아의 gCloud 서버를 사용합니다.

- 서버 IP : 182.162.142.159
- ID : Administrator
- PW : 9yKd7u9#^Hqq

3.2. 담당 업무

1. 리뉴얼 적용
2. 서버 선택 기능 관련
3. 레이어매칭 적용

3.3. 프로젝트 설명

3.3.1. 리뉴얼 적용

DPing의 앞/뒤 전환 기능으로 인해 카테고리가 4단계로 확장되었기 때문에, 기존 UI를 쓸 수 없는 상황이었습니다. webtex3d 리뉴얼 작업 진행시 변경된 사항들을 확장된 카테고리에 맞게 적용했으며, DPing에 구현되어 있는 앞/뒤 전환 기능을 구현했습니다.

3.3.2. 서버 선택 기능 관련

박진 사원이 서버 등록 및 조회 기능을 개발하였으며, 다음의 추가적인 개발을 진행하였습니다. 관리자 페이지의 authentication 탭에서 서버를 등록하면 랜덤으로 인증코드를 생성하며, 인증 코드를 매개변수로 받아 서버의 정보를 리턴합니다.

3.3.3. 레이어매칭 적용

박진 사원이 개발 완료한 레이어매칭 기능을 변경된 카테고리에 맞게 적용하고, 모델 선택 기능을 추가 개발하였습니다. mobile.stylefor.us/viewerMatching 페이지에서 확인이 가능합니다. 모델, 하위, 상의 이미지의 z-index를 조정하여 png 이미지를 겹치는 형태로 개발이 진행되었습니다. 레이어매칭 페이지에서는 앞/뒤 전환 기능을 구현하지 않았으며, 추후에 개발을 진행할 수도 있습니다. DPing에서도 레이어매칭을 개발 진행중이므로, 이 부분에 대하여 신승민 사원과의 협업이 요구됩니다.

3.3.3.1. 개발 방식

레이어매칭 페이지과 뷰어페이지의 차이점은 다음과 같습니다.

- 모듈/시뮬레이션 결과 이미지의 포맷
 - 뷰어 페이지는 기존의 **jpg 모듈**을 사용, **jpg 이미지를** 돌려받습니다.

- 레이어매칭 페이지는 png 모듈을 사용하여, 결과 이미지로 png 이미지를 돌려받습니다.
- 선택 Product의 갯수
 - 뷰어 페이지는 1개 입니다.
 - 레이어매칭 페이지에서는 상의 1개, 하의 1개로 총 2개 입니다.

The screenshot shows the Dping (디핑) application interface. On the left, there is a sidebar with various product categories: Sample, Color, Print, Weave, Bed&Sofa, Wallpaper, Floor, Traditional suit, and test. Under the Color category, 'color1' is selected. Below the sidebar is an 'Upload' button with a downward arrow icon. The main area has three tabs at the top: Top_Product, Bot_Product, and Fabric. The Top_Product tab is active, showing a 'Recent' section with four color swatches labeled 04 (pink), 05 (orange), 06 (teal), and 14 (brown). Below this are four rows of four color swatches each, labeled 01 through 17. To the right of the color swatches is a circular color palette with a minus sign on the left and a plus sign on the right, labeled '14'. A small number '14' is also visible near the bottom of the palette. Further to the right is a 3D mannequin wearing a brown short-sleeved button-down shirt and pink shorts. At the bottom of the interface are copyright information ('COPYRIGHT © 2017 YOUNGWOO C&I ALL RIGHTS RESERVED.') and share/share icons.

```
> == $0


```

레이어 매칭의 핵심 개념은 Elite, Smart처럼 png 이미지를 css 속성인 z-index 값을 사용하여 겹쳐주는 것 입니다. model 이미지의 z-index값은 -1로 가장 작으며, Bot_Product에서 선택한 Product의 시뮬레이션 이미지는 100, Top_Product에서 선택한 Product의 시뮬레이션 이미지는 200 입니다. 그러므로 모델 이미지 > 하의 > 상의 순으로 이미지가 겹쳐지게 되는 것 입니다.

3.3.3.2. 주의 사항

레이어매칭 전용으로 분리된 카테고리나 관리자 페이지를 구성하는 형태로 개발이 되어 있지 않고, 기존 구조의 카테고리를 그대로 사용합니다. 즉, Top_Product와 Bot_Product에는 기존 구조의 동일한 전체 카테고리를 두 번 받아오며, Top_Product 탭을 이미 눌렀음에도 하위의 top/bottom 카테고리를 다시 선택해야 합니다. 또한, 레이어매칭 카테고리의 이름 앞에는 "#"을 붙여서 페이지 접근시 이름으로 구분하여 보이게/안보이게 하는 형태로 하드 코딩이 되어 있습니다.

그렇기 때문에 지금의 구조를 벗어나는 형식의 개발이 요구된다면 시간이 많이 소요될 것이며 코드 관리가 어려워질 것이므로 시간을 들여서라도 정확한 기획 하에 리팩토링을 하는 것이 좋을 것 같습니다.

3.4. 위험 요소

1. 앞/뒤 product의 불안한 연결 구조

앞, 뒤의 product가 데이터베이스 상으로 연결되어 있지 않으며, 4레벨인 Front, Back 카테고리에 올려진 Product의 갯수와 순서로 인지하고 있습니다. (Front 카테고리의 다섯 번째 Product의 뒷면은 Back 카테고리의 다섯 번째 Product이다.) 그렇기 때문에 다음과 같은 문제점이 있습니다.

- Front, Back 카테고리에 올려진 Product의 갯수나 순서가 다르면 Web에서는 물론, DPing에서도 올바르게 보여지지 않습니다.
- 앞면만 있는 Product라도 Front, Back 카테고리에 두 번 업로드 해야 합니다.

이와 같은 문제의 해결과, 계속 되는 기획의 변경, 확장성을 위해서는 카테고리의 재설계가 꼭 필요하지만, DPing 앱이 상용화 되어 있어 현실적으로 변경이 어렵습니다.

2. 앞/뒤 product의 FabricGroup 갯수가 동일해야 합니다.

앞/뒤 product의 FabricGroup의 갯수가 같다면 문제가 없지만, 반대의 경우는 문제가 있습니다. 시뮬레이션 함수를 실행하면 인자로 `product의 id + FabricGroup에 선택된 fabric id list` 를 넘깁니다. 예를 들어, 앞면의 FabricGroup은 셔츠 + 넥타이 + 바지 3개입니다. 뒷면은 넥타이가 보이지 않으므로, 뒷면의 FabricGroup은 셔츠 + 바지 2개입니다. 앞면의 시뮬레이션 함수 실행시, 빨강(셔츠) + 노랑(넥타이) + 파랑(바지)의 원단 리스트를 인자로 보냅니다. 뒷면 전환 기능을 실행하면, FabricGroup은 2개인데, 빨강(셔츠) + 노랑(바지) + 파랑(?) 3개를 인자로 보냅니다. 이렇게 다른 원단으로 맵핑이 되는 결과가 발생합니다. 이 문제를 방지하기 위해서는 디자인 작업 시에 FabricGroup의 갯수와 순서가 동일해야 합니다.

3. DPing의 레이어 갯수는 1개입니다.

기존의 WebTex3D는 여러 개의 레이어와 FabricGroup으로 구성되어 있지만, mobile.stylefor.us의 데이터는 레이어 한 개 + 여러개의 FabricGroup으로 구성되어있습니다. DPing의 개발 초기부터 레이어는 1개로 설계, 개발이 되었습니다.

4. Fukushin Dress

<http://sys.sho-fukushindress.jp/>

WebTex3D를 고객의 요구에 맞춰 커스터마이징 한 프로젝트입니다.

4.1. 참여 인원

이하영, 김찬율

4.2. 담당 업무

UI 변경과 원단 정보의 변경입니다.

4.3. 프로젝트 설명

커스터마이징 된 항목은 다음과 같습니다.

- UI 변경
- 시뮬레이션 완료된 제품의 위시리스트 기능
 - Sunregent의 카트 기능과 동일합니다.
- 위시리스트에 담긴 제품 주문창으로 전달
- 구글 번역 기능

5. Sunregent

<http://curtain.sunregent.co.jp/>

The screenshot shows a user interface for selecting curtains. At the top, there is a navigation bar with links for '既製' (Ready-made), 'オーダー' (Order), 'マイセレクション' (My Selection), '1. Product 카테고리' (1. Product Category), 'adminさん こんにちは!' (Hello admin), and 'logout'. Below the navigation is a large image of a living room with a brown sofa and patterned curtains. To the right of the sofa is a search bar labeled 'スタイル 検索' (Style Search) and a panel labeled '4. Product 검색 패널' (4. Product Search Panel) containing nine thumbnail images of different curtain styles. Below this is a section labeled '6. 시뮬레이션 이미지' (6. Simulation Image) showing a preview of the curtains in the room. Further down is a section labeled '7. 현재 product의 layer 리스트와 각 layer에 mapping된 fabric의 이름' (7. Current product's layer list and names mapped to each layer's fabric). This section includes a table with two rows:

Layer	Fabric Name
드레ープ	未選択
레ース	未選択

At the bottom right are buttons for '2. Product 리스트' (2. Product List), '3. Layer 리스트' (3. Layer List), '4. Product 검색 패널' (4. Product Search Panel), and '5. Fabric 검색 패널' (5. Fabric Search Panel). There are also buttons for 'ドレープ' (Drape) and 'レース' (Sheer). At the very bottom are buttons for 'SNS', 'お気に入りに追加' (Add to Favorites), and 'お気に入りを見る' (View Favorites).

5.1. 서버 정보

- 서버IP : 153.120.127.154
- ID : Administrator
- PW : MFLLo8VYA

5.2. 계정 정보

로그인 페이지는 `/login_site` 입니다.

- ID : admin
- PW : keyucadmin

5.3. 참여 인원

김찬율, 이하영, 박진, 강해석

5.4. 담당 업무

- 카테고리 구조의 변경
- Product, Fabric 검색

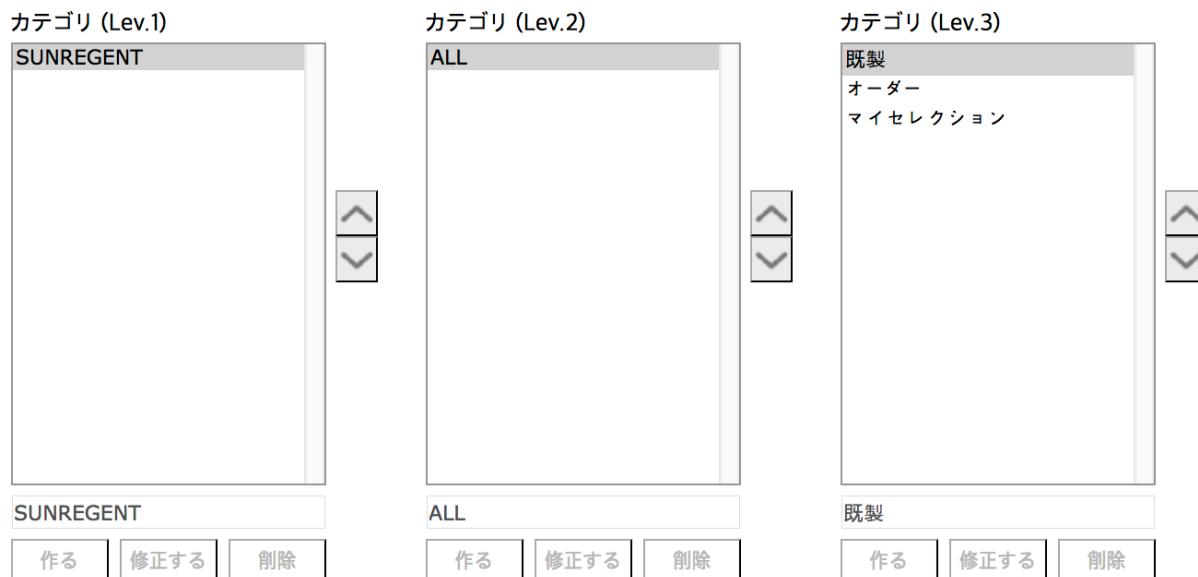
5.5. 프로젝트 설명

Sunregent 프로젝트의 주요 커스터마이징 내용은 다음과 같습니다.

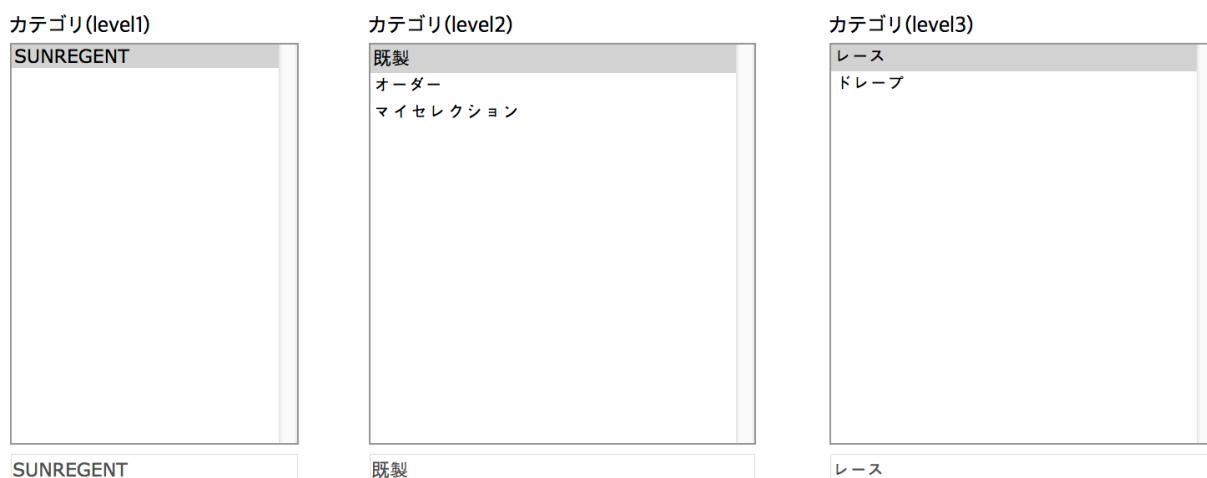
- 반응형 웹(PC, 태블릿, 모바일)
- 카테고리 구조의 변경
- Product 검색
- Fabric 검색
- 원단의 업로드 방식(csv 파일 등록)
- 시뮬레이션 결과의 위시리스트 기능
- 위시리스트 PDF로 저장 및 인쇄
- SNS 공유

5.5.1. 카테고리 구조

Product 카테고리



Fabric 카테고리



- Product와 Fabric의 메인 카테고리인 既製(기성), オーダー(오더), マイセレクション(나의선택)는 GlobalCategory 테이블에 의해서 서로 연결이 되어 있습니다.
- 카테고리 분류는 위와 같이 고정이 되어있으므로, 관리자 페이지에서의 카테고리 추가, 수정, 삭제, 순서 변경 기능을 제한하였습니다.
- Fabric 3 레벨 카테고리는 각각의 메인 카테고리가 가질 수 있는 레이어 이름들을 카테고리로 등록해둔 것 입니다. 메인 카테고리를 선택하고 product를 선택한 후 mapping 할 레이어를 선택하면 메인 카테고리 > 레이어 이름과 동일한 카테고리에 올려진 Fabric들 중에서 검색을 하게 됩니다.
- 다음은 메인 카테고리와 그에 속한 레이어, 즉 Fabric 2 레벨 카테고리와 그에 속한 3 레벨 카테고리입니다.
 - 既製
 - レース
 - ドレープ
 - オーダー
 - レース
 - ドレープ
 - マイセレクション
 - レース
 - ドレープ
 - ソファー
 - ラグマット
 - ベッドカバー
 - クッション
 - 床

5.5.2. Product 검색

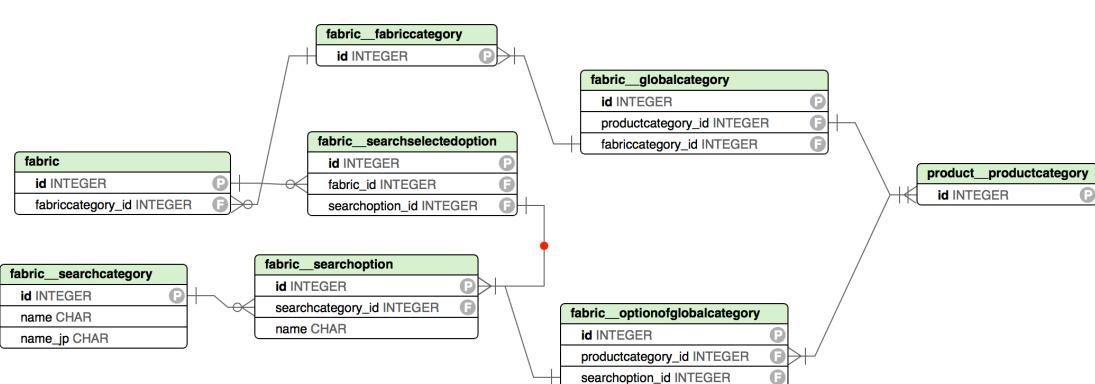
- 메인 카테고리마다 검색 항목이 다릅니다.
- Product의 검색 항목에 대한 옵션은 관리자 페이지에서 지정합니다.
- 검색 조건은 검색 항목끼리 AND, 검색 항목 내의 옵션들은 OR 입니다.
- Product 검색 항목과 관련된 테이블입니다.
 - 部屋タイプ : product_searchroom
 - スタイル : product_searchstyle
 - 窓の種類 : product_searchwindow
- 기존 webtex3d-keyuca-curtain 프로젝트에 개발이 되어 있던 검색 기능을 그대로 사용하였습니다. 항목이 변경될 때 마다 데이터베이스 마이그레이션을 적용해야하며, viewer 페이지의 html, 관리자 페이지의 html을 모두 수정해야 하는 불편함이 있습니다. 다음 이미지는 product_searchstyle 테이블입니다.

	id [PK] serial	product_id integer	option0 boolean	option1 boolean	option2 boolean	option3 boolean	option4 boolean	option5 boolean	option6 boolean
1	16	19	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
2	17	18	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE

5.5.3. Fabric 검색

- 메인 카테고리 + 레이어(Fabric 레벨 3 카테고리)마다 검색 항목, 옵션이 다릅니다.

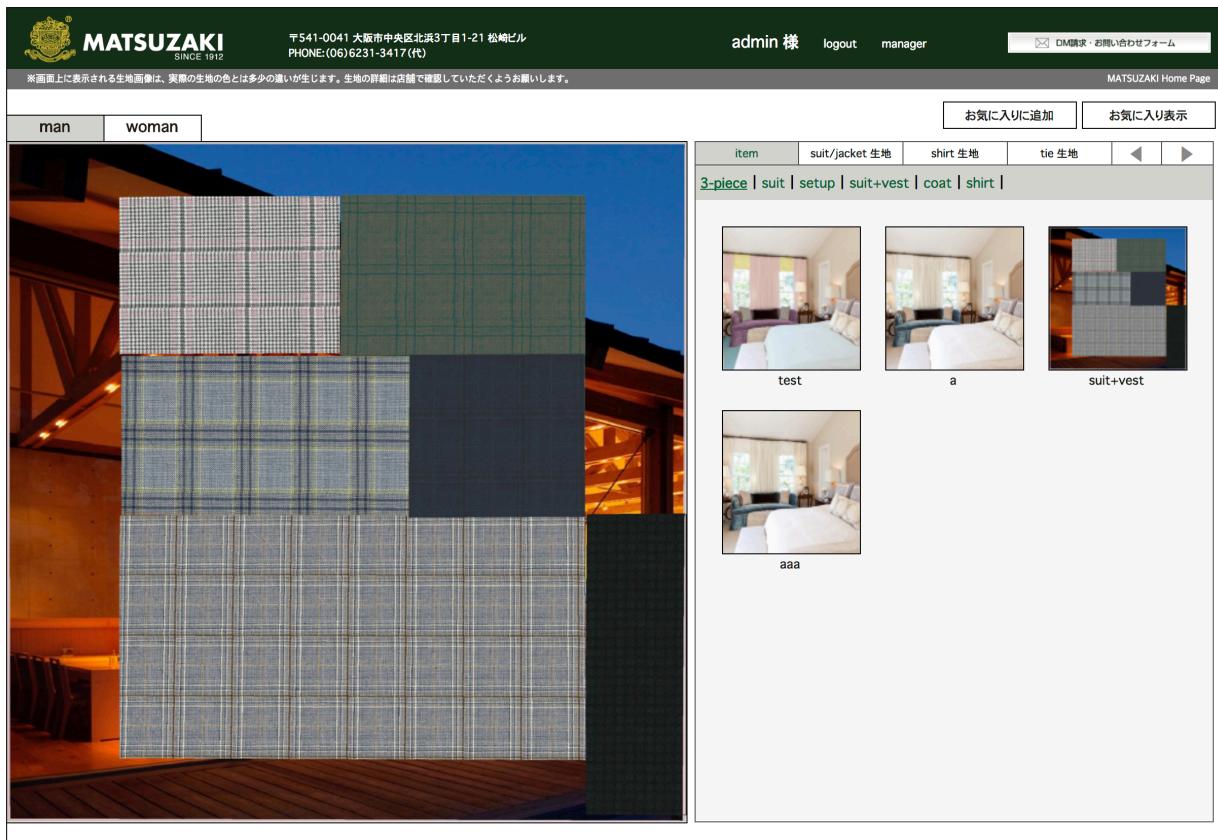
- 검색 항목에 대한 옵션은 csv 파일을 업로드 하여 지정합니다.
- 검색 조건은 Product와 동일하게 항목끼리는 AND, 항목 내의 옵션끼리는 OR 조건입니다.
- Fabric 검색 항목과 관련된 테이블입니다. product와 마찬가지로, 기존에 있던 검색 기능을 그대로 사용하였지만 계속 되는 요청 사항으로 인하여 검색 항목이 점점 복잡해져, 대응하기 쉬운 형태로 데이터베이스 구조를 변경하였습니다.
 - fabric_searchcategory : 검색 항목들을 관리하는 테이블입니다.
 - fabric_searchoption : 검색 항목별 옵션 리스트를 관리하는 테이블입니다.
 - fabric_searchselectedoption : fabric이 가지는 검색 옵션들이 저장되는 테이블입니다.
 - fabric_optionofglobalcategory : 메인 카테고리별 보여지는 검색 항목의 옵션들을 관리하는 테이블입니다.
- Fabric 검색과 관련된 데이터베이스 구조입니다.



5.5.4. 레이어 구조

- Product가 가지는 레이어의 이름은 fabric 3 레벨 카테고리와 동일해야 하며, 하나의 레이어가 가지는 fabric group은 반드시 1개입니다.

6. Matsuzaki



6.1. 참여 인원

이하영, 박진, 강해석

6.2. 담당 업무

- Product 카테고리와 Fabric 카테고리의 연결 작업
- Fabric 검색

6.3. 프로젝트 설명

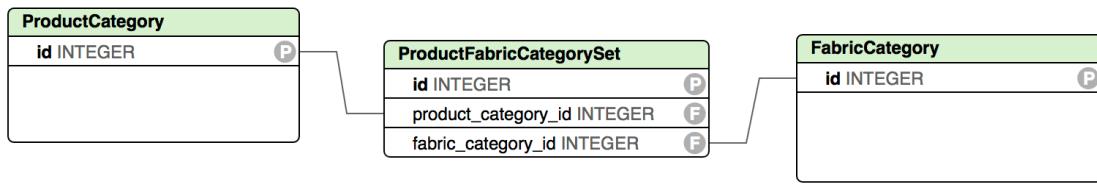
주요 커스터마이징 내용은 다음과 같습니다.

- 카테고리 연결
- Fabric 검색
- 시뮬레이션 결과의 위시리스트 기능
- UI 변경

6.3.1. 카테고리 연결

Product 카테고리마다 보여지는 Fabric 카테고리의 항목들이 다릅니다. 특정 Product 카테고리와 연결되는 Fabric 카테고리들은 fabric_productfabriccategoryset 테이블에 등록되어 있습니다.

변경된 데이터베이스 구조와 fabric_productfabriccategoryset을 구성하는 데이터입니다.



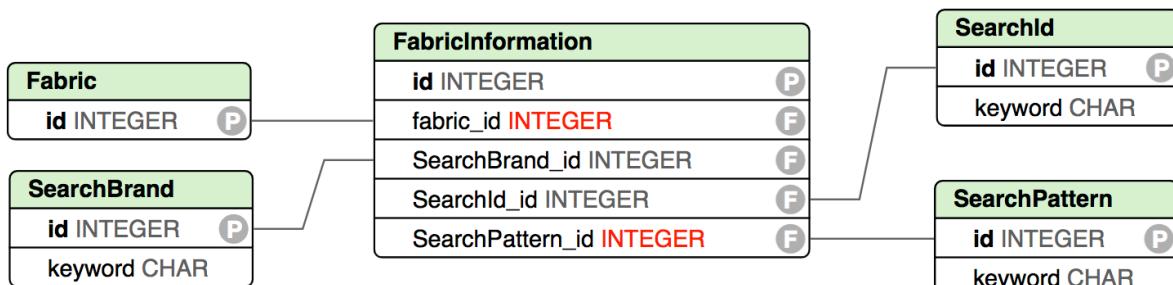
	id [PK] serial	product_category_id integer	fabric_category_id integer	
1	1	5	132	
2	2	5	138	
3	3	5	140	
4	4	5	141	
5	5	5	142	
6	6	6	132	
7	7	6	138	
8	8	6	140	
9	9	6	141	
10	10	6	142	
11	11	7	132	
12	12	7	136	
13	13	7	138	
14	14	7	140	
15	15	7	141	
16	16	7	142	
17	17	8	132	
18	18	8	134	

Fabric 카테고리는 Product의 레이어 이름과 같습니다.

1. Product를 선택하고,
2. Fabric 카테고리를 선택하면
3. 선택한 Fabric 카테고리와 이름이 같은 Product의 레이어(FabricGroup)이 자동으로 mapping됩니다.

6.3.2. Fabric 검색

원단마다 검색 항목별 keyword는 1개씩 가질 수 있습니다. 그러므로, 각 원단 검색 항목은 Fabric과 1:1 관계인 FabricInformation에서 참조합니다. 원단 검색의 데이터베이스 구조입니다.



7. Elite

[WebTex3d-Shirts](#)는 특정 부분의 Style 변경이 가능하며, 원단 mapping이 가능한 프로그램입니다. 하지만, 속도가 매우 느리다는 단점이 있습니다. Elite는 WebTex3d-Shirts의 단점을 보완하여 기획된 프로그램입니다.

Shirts와의 차이점은 다음과 같습니다.

- 시뮬레이션 처리 방식
 - Shirts : 시뮬레이션 모듈을 사용합니다.
 - Elite : 모듈을 사용하지 않고, 미리 모든 경우의 수를 렌더링 한 이미지들을 업로드하여, 웹에서 필요한 이미지를 불러와서 조합하는 형태로 시뮬레이션을 처리합니다.
- 속도
 - Shirts : Style에 따른 레이어 파일들이 많기 때문에 시뮬레이션 처리시 선택한 Style들의 레이어 파일을 열고, 원단을 mapping하는 과정을 거쳐야 합니다. 그러므로 속도가 매우 느립니다.
(최소 8초이며, 원단 크기가 크거나 dpi가 높을 수록 더욱 느립니다.)
 - Elite : 미리 렌더링된 이미지들을 불러와서 보여주는 뷰어 역할만 하기 때문에 속도가 빠릅니다.
- 원단
 - Shirts : Fabric 카테고리에 올려진 모든 원단을 제한 없이 사용할 수 있습니다.
 - Elite : 미리 지정한 원단으로 렌더링이 된 이미지를 사용하므로, 원단의 제한이 있습니다.

7.1. 서버 정보

- 서버IP : 125.131.176.25
- ID : Administrator
- PW : qwer1234!

7.2. 참여 인원

이하영, 박진, 강해석

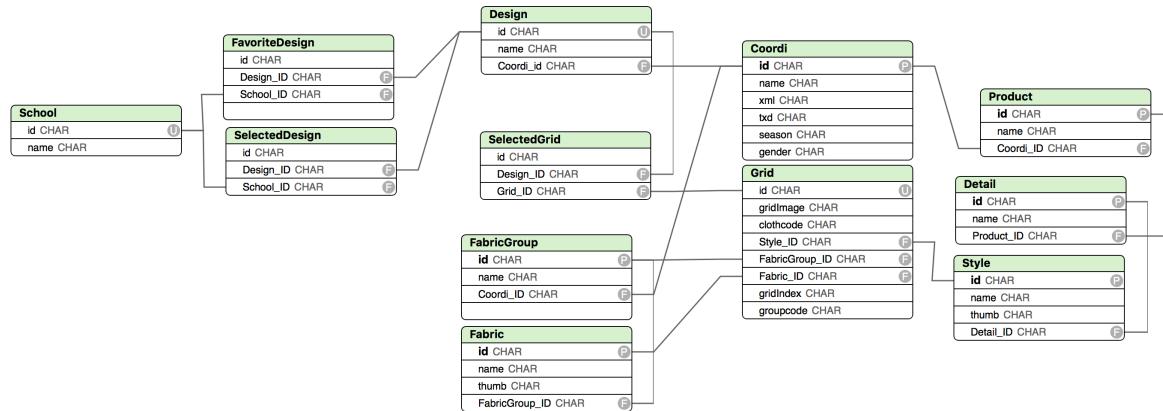
7.3. 담당 업무

1. 시뮬레이션 데이터베이스 설계
2. Tex3DW에서 웹으로 업로드
3. 시뮬레이션 페이지 개발

7.4. 프로젝트 설명

7.4.1. 데이터베이스 구조

시뮬레이션의 데이터베이스 구조입니다.



7.4.2. 용어 설명

- **Coordi** : 업로드 되는 하나의 세트입니다.
- **Product** : 하나의 Coordi를 구성하는 옷입니다.
하나의 Coordi에 속하는 Product는 기본으로 상의, 하의가 있습니다.
- **Detail** : 하나의 Product를 구성하는 특정 부분입니다.
예를 들어, 상의(Product)에 속하는 Detail은 앞판, 뒷판, 소매 등이 있습니다.
- **Style** : Detail에 속하는 스타일 목록입니다.
예를 들어, 상의(Product)의 카라(Detail)에 속하는 Style에는 차이나 카라, 라운드넥 등이 있습니다.
- **FabricGroup** : 하나의 원단으로 mapping되는 범위입니다.
- **Fabric** : FabricGroup에 속하는 원단 목록입니다.
- **Grid** : 시뮬레이션의 단위가 되는 이미지 조각입니다.
하나의 그리드는 Style, FabricGroup, Fabric의 정보를 가지고 있습니다.

7.4.3. 데이터 업로드

Tex3DWE 프로그램에서 디자인 작업이 완료된 후 업로드를 실행하면 웹으로 전송될 파일들이 `C:\Youngwoo\` 경로에 생성이 되며, 모든 파일들이 생성되면 `/createproduct` url을 호출, 파일을 전송합니다. 전송된 xml 파일과 이미지 파일들의 네이밍 규칙에 따라서 데이터들을 구분한 후, 데이터베이스에 insert 합니다. 다음은 전송되는 파일들의 키 값과 설명입니다.

- **tex3dXml** : 미리 규칙을 정해둔, 데이터베이스에 저장이 될 정보를 담은 xml 파일
- **StyleThumb** : 시뮬레이션에서 버튼으로 사용될 **스타일**의 썸네일 이미지 리스트
- **FabricThumb** : 시뮬레이션에서 버튼으로 사용될 **원단**의 썸네일 이미지 리스트
- **GridImages** : 시뮬레이션에 사용될 png 조각 이미지 리스트

네이밍 규칙

- Coordi의 이름

○ `<Coordi name="S16B01116THFPPP.txd" id="admin" />`

- 첫 번째 문자는 시즌을 의미합니다.
 - W : 동복
 - S : 하복
 - A : 생활복
- 2-3번째 문자는 연도를 의미합니다.
- 4번째 문자는 성별을 의미합니다.
 - B : 남
 - G : 여
- StyleThumb의 파일 이름
 - 3자리 숫자로 구성이 되어 있습니다.
 - 1번째 숫자는 Product, 2번째는 Detail, 3번째는 Style의 index를 의미합니다.
- FabricThumb의 파일 이름
 - 4자리 숫자로 구성 되어 있습니다.
 - 1-2번째 숫자는 FabricGroup의 index를 의미하며,
3-4번째 숫자는 FabricGroup에 속한 Fabric의 index를 의미합니다.
- GridImages의 파일 이름
 - 7자리 숫자로 구성되어 있습니다.
 - 5-7번째 숫자는 StyleThumb의 네이밍 규칙과 동일하며,
1-4번째 숫자는 FabricThumb의 네이밍 규칙과 동일합니다.

업로드 순서

1. 먼저, xml 파일을 읽고, 데이터베이스에 저장합니다.

```
<ProductInformation>
  <Coordi name="S16B01116THFPPP.txd" id="admin"/>
  <ProductList>
    <Product name="하의" product_index="0" cloth_code="TEST01">
      <Detail name="바지" product_index="0" detail_index="0">
        <Style name="BPS_03" style_index="0"/>
      </Detail>
      <Detail name="바지 주머니" product_index="1" detail_index="1">
        <Style name="BPP-02" style_index="0"/>
      </Detail>
    </Product>
    <Product name="상의" product_index="1" cloth_code="TEST01">
      <Detail name="상의_앞판" product_index="0" detail_index="0">
        <Style name="BTSF-06" style_index="0"/>
        <Style name="BTSF-03" style_index="1"/>
      </Detail>
      <Detail name="상의_뒷판" product_index="1" detail_index="1">
        <Style name="BTSB-01" style_index="0"/>
      </Detail>
      <Detail name="상의_카라" product_index="2" detail_index="2">
        <Style name="BTSC-07" style_index="0"/>
      </Detail>
    </Product>
  </ProductList>
</ProductInformation>
```

```
<Style name="BTSC-04" style_index="1"/>
</Detail>
<Detail name="상의_소매" product_index="3" detail_index="3">
    <Style name="BTSS-01" style_index="0"/>
    <Style name="BTSS-03" style_index="1"/>
</Detail>
<Detail name="상의_주머니" product_index="4" detail_index="4">
    <Style name="BTUP-01" style_index="0"/>
    <Style name="BTUP-10" style_index="1"/>
    <Style name="BTUP-14" style_index="2"/>
</Detail>
</Product>
<Product name="악세사리" product_index="2" cloth_code="TEST01">
    <Detail name="넥타이" product_index="0" detail_index="0"/>
</Product>
</ProductList>
<FabricGroupList>
    <FabricGroup group_code="00" name="(자켓 기본)" />
    <FabricGroup group_code="01" name="(자켓 배색)" />
    <FabricGroup group_code="02" name="(자켓 단추)" />
    <FabricGroup group_code="03" name="(자켓 와펜)" />
    <FabricGroup group_code="04" name="(조끼 앞판)" />
    <FabricGroup group_code="05" name="(조끼 뒷판)" />
    <FabricGroup group_code="06" name="(조끼 배색)" />
    <FabricGroup group_code="07" name="(조끼 단추)" />
    <FabricGroup group_code="08" name="(조끼 와펜)" />
    <FabricGroup group_code="09" name="(악세서리)" />
    <FabricGroup group_code="10" name="(하의 기본)">
        <Fabric fabric_index="00" name="STNA04.jpg"/>
        <Fabric fabric_index="01" name="RTCA02.jpg"/>
        <Fabric fabric_index="02" name="RTCA03.jpg"/>
        <Fabric fabric_index="03" name="RTKA04.jpg"/>
        <Fabric fabric_index="04" name="TTRA13.jpg"/>
        <Fabric fabric_index="05" name="RTMA01.JPG"/>
    </FabricGroup>
    <FabricGroup group_code="11" name="(하의 배색)" />
    <FabricGroup group_code="12" name="(상의 기본)">
        <Fabric fabric_index="00" name="RTBA15.jpg"/>
        <Fabric fabric_index="01" name="RTBA07101.jpg"/>
        <Fabric fabric_index="02" name="RTBA10.jpg"/>
        <Fabric fabric_index="03" name="RTBA11.jpg"/>
        <Fabric fabric_index="04" name="RTBA13.jpg"/>
    </FabricGroup>
    <FabricGroup group_code="13" name="(상의 배색)">
        <Fabric fabric_index="00" name="RTBA09.jpg"/>
        <Fabric fabric_index="01" name="CTMA06.JPG"/>
        <Fabric fabric_index="02" name="MTCA01.JPG"/>
        <Fabric fabric_index="03" name="RTBA12.jpg"/>
        <Fabric fabric_index="04" name="RTBA15.jpg"/>
    </FabricGroup>
</FabricGroupList>
```

```

</FabricGroup>
<FabricGroup group_code="14" name="(상의 배색2)">
    <Fabric fabric_index="00" name="STNA04.jpg"/>
    <Fabric fabric_index="01" name="MTRA01.JPG"/>
    <Fabric fabric_index="02" name="RTCA03.jpg"/>
    <Fabric fabric_index="03" name="RTKA04.jpg"/>
</FabricGroup>
</FabricGroupList>
</ProductInformation>

```

- Product, Detail, Style의 속성값인 index는 StyleThumb 네이밍 규칙에 사용되며, FabricGroup과 Fabric의 속성값인 index가 FabricThumb 네이밍 규칙에 사용됩니다.
 - FabricGroup의 순서와 이름은 고정으로 15개가 있으며, Tex3DWE 프로그램에서 하드코딩 되어 있습니다.
 - 주의 해야할 점은 Product, Detail, Style의 index는 시뮬레이션 페이지에서 css의 속성인 z-index 값으로 활용이 됩니다. 그러므로 밑에 깔려야 할 Product의 index가 낮아야 합니다. 즉, 하위의 index 가 가장 낮아야 하며, 자켓의 index가 가장 높아야 합니다.
2. StyleThumb 파일을 저장합니다. 1번에서 미리 xml을 읽고, 데이터베이스에 Simulation에 필요한 데이터를 저장해두었습니다. Style 파일의 이름의 index 정보와 일치하는 Style을 조회한 후, 파일을 저장해줍니다.

```

for style in enumerate(request.FILES.getlist('StyleThumb')):
    styleFile = style[1].name

    product_index = styleFile[0:1] # style 파일의 1번째 문자
    detail_index = styleFile[1:2] # style 파일의 2번째 문자
    style_index = styleFile[2:3] # style 파일의 3번째 문자

    product_object = Product.objects.get(coordi=coordi,
index=product_index)
    detail_object = Detail.objects.get(product=product_object,
index=detail_index)
    style_object = Style.objects.get(detail=detail_object,
index=style_index)
    style_object.thumb.save(thumb, ContentFile(style[1].read()))

```

3. FabricThumb 파일을 저장합니다. Fabric 파일의 이름과 index 정보가 일치하는 Fabric을 조회한 후, 파일을 저장해줍니다.

```

for fabric in enumerate(request.FILES.getlist('FabricThumb')):
    fabricFile = fabric[1].name

    group_code = fabricFile[0:2] # fabric 파일의 1~2번째 문자
    fabric_code = fabricFile[2:4] # fabric 파일의 3~4번째 문자

    fabricgroup_object = FabricGroup.objects.filter(coordi=coordi,
code=group_code)

```

```

# fabricgroup이 Tex3dWE에서 하드코딩 되어있기 때문에,
# fabricgroup안에 fabric이 비어있는 경우의 예외 처리입니다.
if fabricgroup_object.count() == 0:
    pass
else:
    fabricgroup_object = fabricgroup_object[0]
    fabric_object = Fabric.objects.get(
        fabricGroup=fabricgroup_object,
        code=fabric_code
    )
    fabric_object.thumb.save(thumb, ContentFile(fabric[1].read()))

    fabricgroup_object.save()
    fabric_object.save()

```

4. GridImages 파일을 저장합니다. Grid 파일 이름에서 추출해낸 style, fabricgroup, fabric의 index와 일치하는 데이터를 조회하여, 참조할 수 있도록 저장합니다. 이렇게 해서 하나의 Grid는 Product-Detail-Style의 정보와 FabricGroup-Fabric의 정보를 가지게 됩니다.

```

for grid in enumerate(request.FILES.getlist('GridImages')):
    file = grid[1].name

    styleIndex = file[6:7]
    fabricCode = file[2:4]

    fabricgroup = FabricGroup.objects.get(coordi=coordi,
    code=file[0:2])
    fabric = Fabric.objects.get(fabricGroup=fabricgroup,
    code=file[2:4])
    product = Product.objects.get(coordi=coordi, index=file[4:5])
    detail = Detail.objects.get(product=product, index=file[5:6])
    style = Style.objects.get(detail=detail, index=file[6:7])

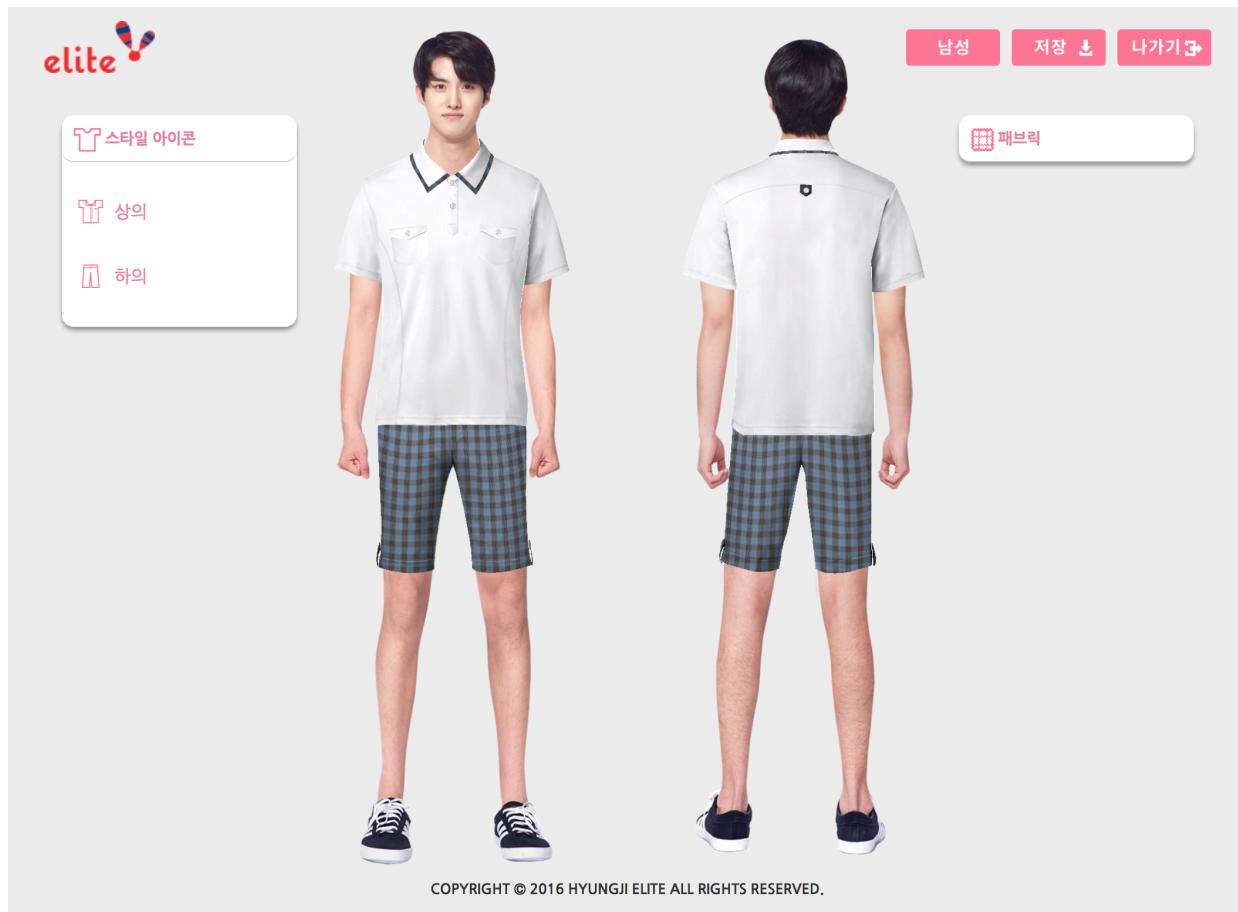
    grid_object = Grid(
        fabric=fabric,
        fabricGroup=fabricgroup,
        style=style,
        groupCode=file[4:7]
    )
    grid_object.pngPath.save(filename, ContentFile(f.getvalue()))
    grid_object.save()

    # 시뮬레이션 페이지 진입시 보여줄 디자인을 지정합니다.
    # 기본으로 지정되는 디자인은 모든 Product-Detail의 첫 번째 Style과,
    # 모든 FabricGroup의 첫 번째 Fabric 입니다.
    if styleIndex == "0" and fabricCode == "00":
        selectedgrid = SelectedGrid(design=design, grid=grid_object)
        selectedgrid.save()

```

7.5. 시뮬레이션

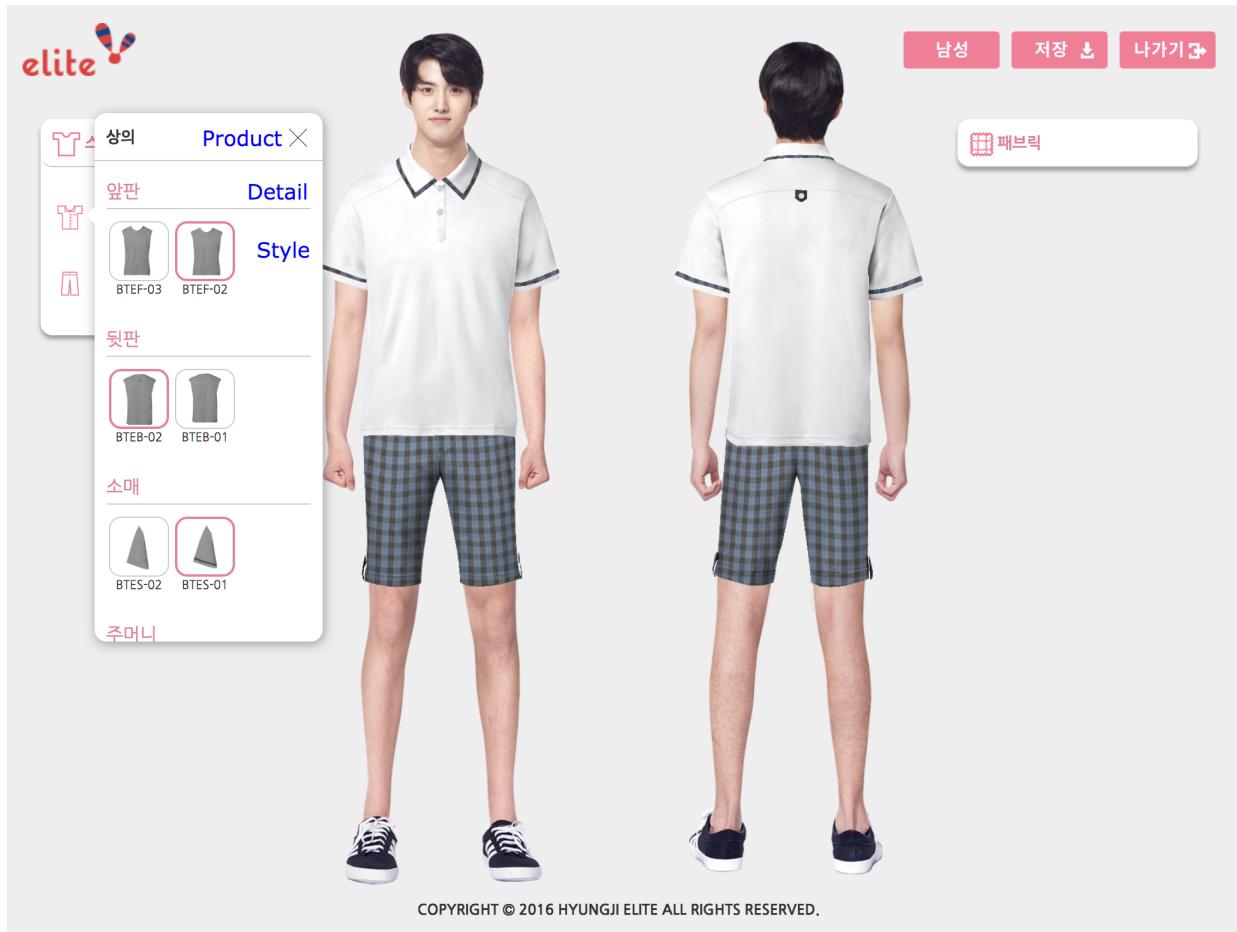
초기 화면



화면 구성은 다음과 같습니다.

- 왼쪽 : Style 선택 메뉴
- 중앙 : 시뮬레이션 이미지 영역
- 오른쪽 : Fabric 선택 메뉴

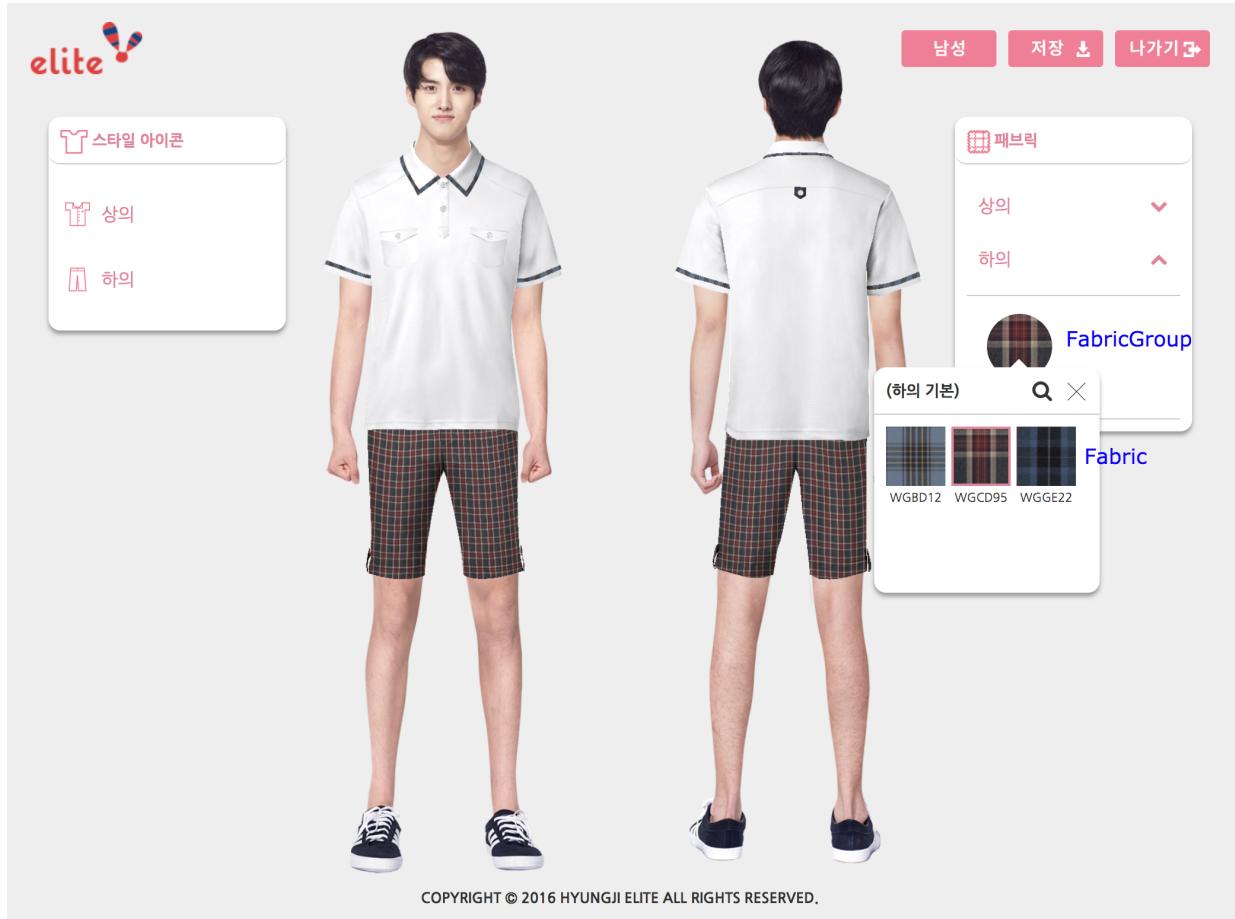
Style 선택



상의 Product를 클릭하여 열린 Style 변경 메뉴입니다. 상의에는 앞판, 뒷판, 소매, 주머니 등의 Detail이 있습니다. 앞판에는 BTEF-03, BTEF-02 2가지의 Style이 있습니다.

상의의 앞판과 소매의 Style을 변경했습니다. 초기화면의 시뮬레이션 이미지와 비교해보면, 디자인이 변경된 것을 확인할 수 있습니다.

Fabric 선택



하의 기본 FabricGroup을 클릭하여 열린 Fabric 변경 메뉴입니다. 하의 기본에는 WGBD12, WGCD95, WGGE22 3가지의 Fabric이 있습니다.

WGCD95 Fabric으로 변경했습니다. 바지의 원단이 변경된 것을 확인할 수 있습니다.

시뮬레이션 원리

시뮬레이션 영역의 이미지는 하나의 이미지로 보이지만, 실제로는 선택된 Style, Fabric과 일치하는 Grid의 png 조각들이 모여서 하나의 코디를 이루는 것입니다.

아래의 html을 보면, `.img-grids` 이미지들이 차곡차곡 쌓여있는 것을 알 수 있습니다. 이미지가 쌓이는 순서는 업로드 할 때 저장된 Product, Detail의 index입니다. 예를 들면, Product의 index가 1, Detail의 index가 3인 그리드 이미지의 z-index는 130이 됩니다. 모델 이미지인 ``는 z-index:0으로 설정해두었으므로, 가장 아래에 깔리게 됩니다.

```
▼<div class="layout-avatar">
  
  ▼<div class="avatar-product-950">
     == $0
    
  </div>
  ▼<div class="avatar-product-951">
    
    
    
    
    
    
    
  </div>
</div>
```

또, product의 종류별로 <div> 태그가 grid 이미지들을 감싸고 있는데요. 동복의 경우 자켓, 조끼의 on-off 기능이 있기 때문에, off를 하면 css의 display 속성을 이용하여 div 내의 이미지들을 숨겨주고, on을 하면 다시 보여주는 형태로 개발을 하였습니다.

8. Smart

Elite 프로젝트의 확장된 버전입니다.

Elite는 Coordinater 단위로 업로드하며, Coordinater라는 범위 내에서 디자인을 하는 것이 목적입니다. Smart는 이미 디자인이 완료된 Product 단위로 업로드하며, 업로드 된 Product를 조합, 판매하는 것이 목적입니다.

8.1. 서버 정보

이준 사원의 인수인계 문서를 참고해주세요.

V:\current\Document\11. 인수인계\20. 개발 담당자 풀더\이준

8.2. 참여 인원

이하영, 박진, 강해석, 이예지

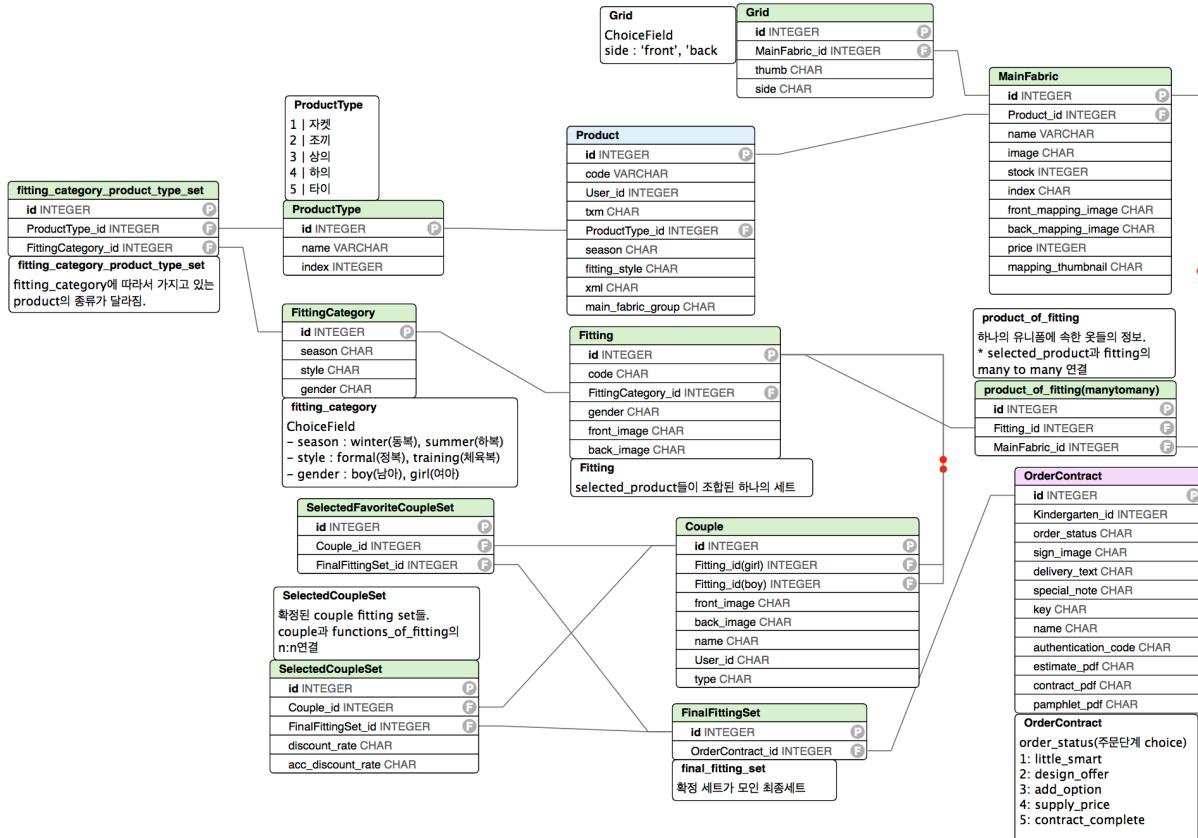
8.3. 담당 업무

1. 데이터베이스 설계
2. TexMD에서 웹으로 업로드
3. 주문 시스템 개발
4. 옵션 관리
5. 사이즈 측정 통계
6. 판매 현황 통계

8.4. 프로젝트 설명

8.4.1. 데이터베이스 구조

시뮬레이션 데이터베이스 구조입니다. 전체 erd는 인수인계 폴더의 smart_erd.png를 참고해주세요.



8.4.2. 용어 설명

- Product : 업로드 되는 단위입니다. Elite와 동일하게 자켓, 조끼, 상의 등이 있습니다.
- MainFabric : 버튼으로 사용될 메인 원단, 그 원단으로 조합이 완료된 Product 이미지가 저장됩니다.
- Fitting : 코디가 완료된 한 세트의 옷을 의미합니다. Product들이 조합되어 하나의 Fitting이 생성됩니다.
- Couple : 남녀의 Fitting이 조합되어 하나의 Couple이 생성됩니다.
- SelectedCoupleSet : 주문 확정한 Couple입니다.
- SelectedFavoriteCoupleSet : 관심으로 등록한 Couple입니다.
- FinalFittingSet : 주문을 확정한 최종 세트입니다.

1개의 주문당 동복정복, 동복체육복, 하복정복, 하복체육복 최대 4개 입니다.

8.4.3. 데이터 업로드

simulation app에 대한 설명입니다.

네이밍 규칙 & 업로드 데이터

Elite 프로젝트와 동일합니다.

최종 결과물

- Smart 프로젝트는 Style이 바뀔 필요가 없으며, 상품의 색상이 확정되어 있습니다.
- 최종 결과물인 Grid 이미지들이 조합된 Product 이미지는 웹에서 합성으로 생성합니다.
- 업로드 처리가 끝난 Product들은 admin 페이지에서 확인이 가능합니다.

The screenshot shows a web-based product management interface for 'little smart'. It displays three product entries (W04, W05, W06) with their details, images, and size charts.

No.	이름	계절	복종	품목	업로더	등록일	txm파일	공개
246	W04	동복	체육복	셔츠	admin	2017년 8월 21일 2:50 오후	txm다운로드	<input checked="" type="checkbox"/>
381	W05	동복	체육복	셔츠	admin	2017년 8월 22일 6:24 오후	txm다운로드	<input checked="" type="checkbox"/>
368	W06	동복	체육복	셔츠	admin	2017년 8월 22일 4:09 오후	txm다운로드	<input checked="" type="checkbox"/>

Product W04:

90	100	110	120
28개	27개	69개	75개
130	140	150	총합
25개	24개	9개	257개

Product W05:

90	100	110	120
0개	0개	0개	0개
130	140	150	총합
0개	0개	0개	0개

Product W06:

90	100	110	120
20개	41개	39개	40개
130	140	150	총합
40개	21개	0개	201개

업로드 순서

- TexMD에서 업로드를 실행하면, Simulation app의 upload/views.py에 있는 UploadProduct() 함수가 호출됩니다.
- 먼저, xml 파일을 읽고, 데이터베이스에 저장합니다.

```
<ProductInformation>
  <Coordi name="N19.txm" id="admin" season="summer" style="training"
mainfabric="01"/>
  <ProductList>
    <Product name="shirts" product_index="0" cloth_code="TEST01">
      <Detail name="Front" product_index="0" detail_index="0">
        <Style name="N19" style_index="0"/>
      </Detail>
      <Detail name="Back" product_index="1" detail_index="1">
        <Style name="N19" style_index="0"/>
      </Detail>
```

```

        </Product>
    </ProductList>
    <FabricGroupList>
        <FabricGroup group_code="00" name="0">
            <Fabric fabric_index="00" name="HTT111.jpg"/>
            <Fabric fabric_index="01" name="HTT111.jpg"/>
        </FabricGroup>
        <FabricGroup group_code="01" name="0">
            <Fabric fabric_index="00" name="HPQ454.jpg"/>
            <Fabric fabric_index="01" name="HPQ328.jpg"/>
        </FabricGroup>
        <FabricGroup group_code="02" name="0">
            <Fabric fabric_index="00" name="HPQ104.jpg"/>
            <Fabric fabric_index="01" name="HPQ104.jpg"/>
        </FabricGroup>
        <FabricGroup group_code="03" name="0">
            <Fabric fabric_index="00" name="HPQ001.jpg"/>
            <Fabric fabric_index="01" name="HPQ001.jpg"/>
        </FabricGroup>
        <FabricGroup group_code="04" name="0">
            <Fabric fabric_index="00" name="HPQ454.jpg"/>
            <Fabric fabric_index="01" name="HPQ328.jpg"/>
        </FabricGroup>
        <FabricGroup group_code="05" name="0">
            <Fabric fabric_index="00" name="HPQ104.jpg"/>
            <Fabric fabric_index="01" name="HPQ104.jpg"/>
        </FabricGroup>
        <FabricGroup group_code="06" name="0"/>
        <FabricGroup group_code="07" name="0"/>
        <FabricGroup group_code="08" name="0"/>
        <FabricGroup group_code="09" name="0"/>
        <FabricGroup group_code="10" name="0"/>
        <FabricGroup group_code="11" name="0"/>
        <FabricGroup group_code="12" name="0"/>
        <FabricGroup group_code="13" name="0"/>
    </FabricGroupList>
</ProductInformation>

```

- <`Coordi` /> 의 `mainfabric` 속성은 메인으로 사용할 FabricGroup의 code입니다.
- Product 단위로 업로드가 되므로, <`Product`> 1개가 업로드 되며, 앞/뒤 전환을 위한 Front, Back <`Detail`> 이 있습니다. TexMD에 Detail은 Front/Back 2개로 고정 되어 있습니다.
- <`Coordi` /> 에 지정한 `mainfabric` 에 해당되는 <`FabricGroup`> 의 정보는 MainFabric 테이블에 저장이 되며, 나머지 FabricGroup의 정보는 버립니다.

```

main_fabric_group_index = int(product.main_fabric_group)

xml_fabric_group = root.find("FabricGroupList")
[main_fabric_group_index]
for xml_fabric in xml_fabric_group.iter("Fabric"):
    main_fabric = MainFabric(product=product)
    fabric_code = xml_fabric.attrib["fabric_index"]
    main_fabric.index = convertFabricCodeFormat(fabric_code)
    main_fabric.name = xml_fabric.attrib["name"]
    main_fabric.save()

```

3. FabricThumb 이미지를 저장합니다. main fabric group에 해당되는 Fabric만 저장합니다.

```

for fabric in enumerate(fabricThumbs):
    fabricFile = fabric[1].name
    group_code = fabricFile[0:2]
    fabric_code = fabricFile[2:4]

    if group_code == product.main_fabric_group:
        m_fabric = MainFabric.objects.get(product=product,
                                         index=convertFabricCodeFormat(fabric_code))
        m_fabric.image.save('main_fabric/' + fabricFile,
                           ContentFile(fabric[1].read()))
    else:
        pass

```

4. Grid 이미지를 저장합니다.

```

for grid in enumerate(gridImages):
    grid_name = grid[1].name
    fabric_code = grid_name[2:4]

    # 저장된 MainFabric에서 grid 파일 이름의 3-4번째 코드를(string 값)
    # 숫자로 변환한 index와 일치하는 fabric을 검색하고, 참조합니다.
    main_fabric = MainFabric.objects.get(product=product,
                                         index=convertFabricCodeFormat(fabric_code))

    grid_object = Grid(fabric=main_fabric)
    grid_object.save()

    # Grid 파일 이름의 detail_index로 front/back을 판별하여 side에 저장해줍니다.
    detail_index = grid_name[5:6]
    if detail_index == '0':
        grid_object.side = 'front'
    else:
        grid_object.side = 'back'

```

5. 이제, 저장된 그리드 이미지를 합성하여 최종 결과물로 만듭니다.

```
def createProductThumbnail(product):
    main_fabrics = MainFabric.objects.filter(product=product)
    IMAGE_EXTENSION = '.png'

    # 업로드 된 원본 이미지의 사이즈 입니다
    original_width =
    Grid.objects.filter(fabric__product=product).first().image.width
    original_height =
    Grid.objects.filter(fabric__product=product).first().image.height

    # resize 할 width 값 지정
    fitting_resize_width = 500           # 최종 결과물의 width
    product_thumb_resize_width = 300    # 버튼으로 사용될 이미지의 width

    # height가 비율에 맞게 resize 될 수 있도록, 사이즈를 구하는 식입니다.
    fitting_thumb_size = (fitting_resize_width, (fitting_resize_width * original_height / original_width))
    product_thumb_size = (product_thumb_resize_width,
    (product_thumb_resize_width * original_height / original_width))
    resize_ratio = fitting_thumb_size[0] / original_width

    # 합성을 위해, product의 원단들을 순회합니다.
    for main_fabric in main_fabrics:
        # side가 front이면서 fabric이 일치하는 grid를 조회한 후, 합성합니다.
        front_grids = Grid.objects.filter(fabric=main_fabric,
        side='front').order_by('id')
        front_image = compositeGrid(front_grids, fitting_thumb_size)
        image_filename = 'mapping/front/' + str(main_fabric.id) +
        IMAGE_EXTENSION
        main_fabric.front_mapping_image.save(image_filename,
        ContentFile(front_image.read()))
        front_image.close()

        # 버튼으로 쓰일 썸네일을 만듭니다.
        crop_thumb = cropProductThumb(product,
        main_fabric.front_mapping_image, resize_ratio, product_thumb_size)
        image_filename = 'mapping/thumb/' + str(main_fabric.id) +
        IMAGE_EXTENSION
        main_fabric.mapping_thumb.save(image_filename,
        ContentFile(crop_thumb.read()))
        crop_thumb.close()

        # back 이미지도 동일하게 처리합니다.
```

Grid 이미지의 합성을 처리하는 함수입니다.

```

# front 또는 back의 그리드들과, resize 될 사이즈를 파라미터로 받습니다.
def compositeGrid(grids, thumb_size):
    idx = 0
    composite_img = ''

    # grid들을 순회하며, 차곡차곡 합성해줍니다.
    for grid in grids:
        if idx == 0:
            composite_img = Image.open(grid.image)
        else:
            img = Image.open(grid.image)
            # png 이미지를 합성하는 Pillow 라이브러리의 함수입니다.
            # with문은 파일을 열어 처리한 후, 바로 닫아주는 구문입니다.
            # 파일을 닫아주는 것은 매우 중요합니다(!)
            with composite_img as c:
                composite_img = Image.alpha_composite(c, img)
            grid.image.close()
            img.close()
        idx = idx + 1

    # thumb_size로 resize
    composite_img.thumbnail(thumb_size, Image.ANTIALIAS)

    temp_handle = StringIO()
    composite_img.save(temp_handle, 'PNG')
    temp_handle.seek(0)

    thumbImage = ContentFile(temp_handle.read())

    temp_handle.close()
    composite_img.close()

    return thumbImage

```

Product의 종류별로 위치를 지정하여 자른 썸네일을 생성하는 함수입니다. Product의 이미지는 Fitting으로 조합되기 위한 형태이기 때문에 버튼으로 사용될 부분만 잘라서 저장해주는 처리가 필요합니다.

```

def cropProductThumb(product, img, ratio, thumb_size):
    crop_img = Image.open(img)
    product_type = product.product_type.name

    crop_size = 0
    # product의 종류의 따라 잘려질 네 점의 위치를 지정합니다.
    # 원본 Grid 이미지의 사이즈 기준입니다.
    if product_type == 'skirt' :
        crop_size = (int(round(100 * ratio)),
                     int(round(1000 * ratio)),
                     int(round(900 * ratio)),

```

```

        int(round(1660 * ratio)))
elif product_type == 'pants' :
    crop_size = (int(round(100 * ratio)),
                  int(round(1000 * ratio)),
                  int(round(900 * ratio)),
                  int(round(2170 * ratio)))
elif product_type == 'acc' :
    crop_size = (int(round(300 * ratio)),
                  int(round(560 * ratio)),
                  int(round(720 * ratio)),
                  int(round(1050 * ratio)))
else:
    crop_size = (int(round(100 * ratio)),
                  int(round(450 * ratio)),
                  int(round(900 * ratio)),
                  int(round(1440 * ratio)))

# 지정한 위치로 이미지를 crop 합니다
crop_img = crop_img.crop(crop_size)

# 썸네일로 사용할 크기로 비율에 맞게 resize 합니다
crop_img.thumbnail(thumb_size, Image.ANTIALIAS)

temp_handle = StringIO()
crop_img.save(temp_handle, 'PNG')
temp_handle.seek(0)

thumbImage = ContentFile(temp_handle.read())

temp_handle.close()
crop_img.close()

return thumbImage

```

8.4.3. 주문 관리

order_contract app에 대한 설명입니다.

8.4.3.1. add_option

악세사리

외펜

OptionMainCategory



확정된 커플 세트

가방

OptionSubCategory



Option



앞/뒤 전환 버튼



- 시뮬레이션 페이지에서 확정 커플, 관심 커플을 지정한 이후 다음 단계의 페이지입니다.
(주문 생성 ~ 시뮬레이션에 대한 내용은 강해석씨의 인수인계 자료를 참고해주세요.)
- 관리자 - 옵션관리에서 등록한 옵션 카테고리와 옵션이 보여집니다.
- 옵션을 선택하면 모든 커플 세트에 옵션 이미지가 시뮬레이션 되며,
하단의 앞/뒤 전환 버튼으로 앞/뒤 시뮬레이션이 가능합니다.

1. 디자인 제안 2. 추가옵션 3. 공급가격 4. 계약서 작성

admin 님 로그인
유경유치원 의 리틀스마트

로그아웃

기능(Function) 선택

우리 아이 반딧불
미아방지 TAG
엘보우 페치
보들보들 지피랩
우당탕 패드
신축성
ATB-UV+
항균
스스로 단추
클린밴드

저장

확정 커플

관심 커플

◀ ▶

- 확정 커플을 클릭하면 열리는 확정 커플 변경 modal창입니다.
- 확정 커플을 등록해둔 관심 커플로 변경할 수 있습니다.
- 커플의 기능을 선택할 수 있습니다.

8.4.3.2 estimate

1. 디자인 제안 2. 추가옵션 3. 공급가격 4. 계약서 작성

admin 님 로그인
유경유치원 의 리틀스마트

로그아웃

품번	구분	TAG가격	공급가격	이미지
J01-003	자켓(남아)	95,400원	66,800원	
K01-003	조끼(남아)	40,300원	28,300원	
Y03-001	셔츠(남아)	28,200원	19,800원	
P01-001	바지(남아)	51,500원	36,100원	
A03-004	악세서리(남아)	7,000원	7,000원	
남아 합계		222,400원	158,000원	
J01-004	자켓(여아)	95,400원	66,800원	
K01-003	조끼(여아)	40,300원	28,300원	
Y01-003	셔츠(여아)	27,100원	19,000원	
S03-001	치마(여아)	48,200원	33,800원	
A01-004	악세서리(여아)	7,000원	7,000원	
여아 합계		218,000원	154,900원	
합계		440,400원	312,900원	

- option 선택 다음 단계의 페이지입니다.
- 이미지를 클릭하면 add_option 페이지에 있던 특정 커플 변경 modal창이 뜹니다.
- 공급가격을 누르면 할인율을 입력할 수 있습니다. estimate 페이지로 첫 진입이라면 비밀번호 입력창이 뜹니다. 할인율을 입력하면 테이블의 공급가격이 할인된 가격으로 변경됩니다. 최대 할인율은 정복은 45%, 체육복은 50%입니다.

```
def getMaximumDiscountRate(style):
    max_rate = 0

    if style == 'formal':
        max_rate = 45
    elif style == 'training':
        max_rate = 50

    return max_rate
```

옵션의 최대 할인율은 관리자 페이지에서 변경 가능합니다.

8.4.3.3 contract



S'FD

유정유치원 귀하

(서명)

아래와 같이 공급 계약합니다.

합계 금액 514,000원 (VAT 포함)

공급자	사업자등록증	116-81-33765			
	상호	(주)영우씨엔아이	대표자	최영석	(인)
	사업장소재지	서울시 구로구 디지털로31길 20 에이스데크노타워 5차 901호			
	업체	IT		업종	IT
	담당자	이준		연락처	02-851-3000

품목	시즌	상품코드	세부품목	수량	개별가격		합계		
					세제외	세포함	세제외	세포함	
정복	동복	J01-003	자켓(남아)	1 개	60,728원	66,800원	60,728원	66,800원	
		K01-003	조끼(남아)	1 개	25,728원	28,300원	25,728원	28,300원	
		Y03-001	셔츠(남아)	1 개	18,000원	19,800원	18,000원	19,800원	
		P01-001	바지(남아)	1 개	32,819원	36,100원	32,819원	36,100원	
	하복	A03-004	악세서리(남아)	1 개	6,364원	7,000원	6,364원	7,000원	
		J01-004	자켓(여아)	1 개	60,728원	66,800원	60,728원	66,800원	
		K01-003	조끼(여아)	1 개	25,728원	28,300원	25,728원	28,300원	
		Y01-003	셔츠(여아)	1 개	17,273원	19,000원	17,273원	19,000원	
기타	기타	S03-001	치마(여아)	1 개	30,728원	33,800원	30,728원	33,800원	
		A01-004	악세서리(여아)	1 개	6,364원	7,000원	6,364원	7,000원	
		H04-004	셔츠(남아)	1 개	35,728원	39,300원	35,728원	39,300원	
		Z03-001	바지(남아)	1 개	39,273원	43,200원	39,273원	43,200원	
총계	총계	H05-004	셔츠(여아)	1 개	35,728원	39,300원	35,728원	39,300원	
		R09-001	치마(여아)	1 개	33,000원	36,300원	33,000원	36,300원	
		소계		14개	428,189원	471,000원	428,189원	471,000원	
기타	기타	TWA03	와펜	1 개	0원	0원	0원	0원	
		B03	가방	1 개	39,091원	43,000원	39,091원	43,000원	
총계		소계		2개	39,091원	43,000원	39,091원	43,000원	
				16개	467,280원	514,000원	467,280원	514,000원	

- 견적 페이지 다음 단계인 계약 페이지입니다.
- 상단의 서명 버튼을 누르면 canvas 모달창이 뜨며, canvas에 그려진 서명 이미지를 저장합니다.
- 세부품목별 수량을 입력하면 전체 테이블의 가격이 자동으로 변동 됩니다.

8.4.3.4 pamphlet



- 마지막 단계인 팜플렛 페이지입니다.
- 계약이 완료된 세트의 팜플레이 보이며, 계약 완료 내용을 이메일로 발송할 수 있습니다.
- 이메일로 발송되는 내용은 다음과 같습니다.
 - 사이즈 추천 시스템에 진입하기 위한 인증코드
 - 견적 페이지를 이미지로 변환한 pdf 파일
 - 계약 페이지를 이미지로 변환한 pdf 파일
 - 팜플렛 페이지를 이미지로 변환한 pdf 파일

```
@csrf_exempt
def passSendMail(request):
    order_id = request.POST.get('order_id')
    email_to = request.POST.getlist('email[]')

    order = OrderContract.objects.get(id=order_id)
    email_to.append(order.kindergarten.user_profile.user.email)
```

```

# 사이즈 추천 및 사이즈 현황 페이지로 이동하기 위해 필요한 인증코드 6자리 생성
if order.authentication_code:
    authentication_code = order.authentication_code
else:
    authentication_code = randint(100000, 999999)
    order.authentication_code = authentication_code
    order.save()

# 이메일 주소
subject = '[스마트학생복이 만든 리틀스마트]' + str(order.kindergarten) +
' 공급 계약 체결의 件'

# 이메일 내용
message = '\n안녕하세요. ' + str(order.kindergarten) + '님\n' \
    '스마트학생복이 만든 리틀스마트 입니다. \n' \
    '아이들이 입는 옷인 만큼 안정성을 최우선 하는 \n' \
    '아이를 키우는 원복 "리틀스마트"를 선택 해주셔서 감사합니다. \n \n' \
    '사이즈 추천 시스템 (학부모님용) : http://'+ str(request.get_host()) \
    +'measurement/size_recommend/intro/' + str(order_id) + '\n\n' \
    '인증코드 : '+str(authentication_code) +' (사이즈추천시스템, 사이즈 현황  

인증코드입니다. 등록 필수 사항) \n\n' \
    '사이즈 제출 현황 (유치원용) : http://'+ str(request.get_host()) \
    +'measurement/stats/submit/guest/' + str(order_id) + '\n

kwargs = dict(
    to=email_to,
    from_email=settings.EMAIL_HOST_USER + '@s-fd.com',
    subject=subject,
    body=message,
    bcc=[ 'little-smart@s-fd.com' ]
)

# 첨부 파일로 보낼 pdf 파일 경로
message = EmailMultiAlternativesWithEncoding(**kwargs)
message.attach_file(order.estimate_pdf.path)
message.attach_file(order.contract_pdf.path)
message.attach_file(order.pamphlet_pdf.path)
message.send()

variables = { 'success':True}
return HttpResponse(json.dumps(variables),
content_type="application/json")

```

8.4.4. 관리자

8.4.4.1. 디자인정보 > 옵션 관리

[simulation / option / views.py]

■ 메인 카테고리

악세사리

와펜

카테고리명

추가

삭제

■ 서브 카테고리

가방 [최대할인률 : 60%]

카테고리명

최대할인률

추가

삭제



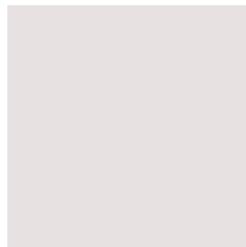
■ 이름

■ 가격

옵션 항목 추가

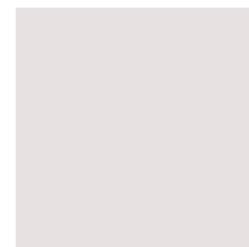
클릭하여 썸네일 이미지를
업로드 하세요.

앞



Upload

뒤



Upload

COPYRIGHT @ 2017 SMART FNC ALL RIGHT RESERVED

- add_option 페이지를 구성하는 옵션의 메인 카테고리, 서브 카테고리, 옵션 이미지를 관리하는 페이지입니다.
- 서브 카테고리의 최대 할인율은 estimate 페이지에서 할인율 지정시 사용이 됩니다.
- 옵션을 등록할 때 필요한 이미지는 3가지입니다. 앞면 또는 뒷면 중 1가지와 썸네일 이미지는 필수입니다.
 - 썸네일로 사용할 이미지
 - 앞면 시뮬레이션 이미지
 - 뒷면 시뮬레이션 이미지
- 옵션의 앞/뒤 이미지는 Fitting 이미지 기준으로 업로드 하며, 시뮬레이션에 사용되는 이미지는 Couple 기준으로, 이미지를 겹쳐서 Couple 형태로 만들어주는 처리를 합니다.

```
def AddOption(request):
    sub_category_id = request.POST.get('sub_category')
    name = request.POST.get('optionName')
    price = request.POST.get('optionPrice')
```

```

thumb = request.FILES['file-image-thumb']
front = request.FILES['file-image-front'] if 'file-image-front' in
request.FILES else ''
back = request.FILES['file-image-back'] if 'file-image-back' in
request.FILES else ''

sub_category = OptionSubCategory.objects.get(id=sub_category_id)

newOption = Option.objects.create(
    sub_category=sub_category,
    name=name,
    price=price,
    visible=True
)
newOption.thumb = thumb

# front 이미지 파일이 있다면 image_composite() 함수 호출, 매개변수로 전달
if not front == '':
    front_img = image_composite(front)
    newOption.front_image.save('front.png',
ContentFile(front_img.read()))
    front_img.close()

if not back == '':
    # front와 동일

# ... 생략

def image_composite(option_img):
    img = Image.open(option_img)
    origin_width, origin_height = img.size

    # fitting 이미지의 width는 200px 입니다.
    resize_width = 200

    # 비율에 맞게 resize 될 사이즈를 구하고, resize 합니다.
    new_size = (resize_width, resize_width * origin_height / origin_width)
    img.thumbnail(new_size, Image.ANTIALIAS)

    # couple 이미지의 width는 351px, height는 위에서 resize된 이미지의 height 입니다.
    composite_width = 351
    composite_height = new_size[1]

    # option 이미지를 couple 사이즈에 맞게 2개로 합성해줍니다.
    # 겹쳐지는 이미지는 첫 번째 이미지의 (155, 0) 위치에서 시작합니다.
    couple_image = Image.new('RGBA', (composite_width, composite_height))
    couple_image.paste(img, (0, 0))
    couple_image.paste(img, (155, 0))

```

```

temp_handle = StringIO()
couple_image.save(temp_handle, 'PNG')
temp_handle.seek(0)

composite_img = ContentFile(temp_handle.read())
temp_handle.close()
couple_image.close()
img.close()

return composite_img

```

합성된 Option 이미지입니다.

(원본은 배경이 투명인 png 이미지이지만, 이해를 위해 배경색을 변경하였습니다.)



- 옵션의 카테고리 및 옵션을 삭제하면 ForeignKey로 연결되어 있는 사용중인 옵션들의 데이터 이상이 발생합니다.
그러므로 데이터베이스 상에서 완전히 삭제되는 것이 아닌, visible값을 False로 줘서 보여지지 않도록 처리했습니다.

8.4.4.2. 영업현황 > 매출 현황

[\[
order_contract/custom_admin/business_status/sales_status.py \]](http://211.238.177.158:8888/order_contract/custom_admin/sales_status/)

little smart	디자인정보		유저관리		영업현황		로그아웃	
	계약현황	인기순위	매출현황					

기간 : ~

1) 유통별 낙찰현황

유통	완료										계약중	
	동복정복		동복체육복		하복정복		하복체육복		소계		소계	
	원수	주문수량	원수	주문수량	원수	주문수량	원수	주문수량	원수	주문수량	원수	
(주)영우씨엔아이	1	2	0	0	1	2	0	0	2	4	0	
리틀스마트	1	8	1	4	1	4	1	2	4	18	0	
합계	2	10	1	4	2	6	1	2	6	22	0	

2) 상권별 낙찰현황

유동	완료										계약중	
	동복정복		동복체육복		하복정복		하복체육복		소계		소계	
	원수	주문수량	원수	주문수량	원수	주문수량	원수	주문수량	원수	주문수량	원수	
서울	0	0	0	0	0	0	0	0	0	0	0	0
경기	0	0	0	0	0	0	0	0	0	0	0	0
인천	0	0	0	0	0	0	0	0	0	0	0	0
강원	0	0	0	0	0	0	0	0	0	0	0	0
전라	0	0	0	0	0	0	0	0	0	0	0	0
경남	0	0	0	0	0	0	0	0	0	0	0	0
경북	0	0	0	0	0	0	0	0	0	0	0	0
서부	0	0	0	0	0	0	0	0	0	0	0	0
부산	0	0	0	0	0	0	0	0	0	0	0	0
합계	0	0	0	0	0	0	0	0	0	0	0	0

3) 제품별 판매현황

품목	품목명	가격ZONE										[단위: 원/VAT포함가] [기준:공급가격]	
		~19,999	20,000~25,000	25,001~30,000	30,001~35,000	35,001~40,000	40,001~45,000	45,001~50,000	50,001~60,000	60,001~70,000	70,001~		
동복정복	자켓	2	0	0	0	0	0	0	0	0	2	6	
	조끼	2	0	8	0	0	0	0	0	0	0	0	
	셔츠	4	2	0	0	0	0	0	0	0	0	0	
	바지	0	0	0	0	3	0	0	0	0	0	0	
	치마	0	0	0	2	1	0	0	0	0	0	0	
	원피스	0	0	0	0	0	0	0	0	0	0	0	

- 유통별, 상권별, 제품별로 order_status의 상태에 따라 통계를 낸 페이지입니다.
 - 완료 : order_status = 5
 - 계약중 : order_status < 5
- 기간을 설정하여 조회가 가능합니다.
- 유통별 낙찰현황
 - Company 테이블에 저장되어 있는 데이터를 순회하며, 복종/시즌별 계약 완료와 계약중인 주문을 count 합니다.
- 상권별 낙찰 현황

- **# 통계 낼 지역들을 변수에 담습니다.**

```
AREA_LIST = [ "서울", "경기", "인천", "강원", "전라", "경남", "경북", "서부",
    "부산" ]
```
- Kindergarten 테이블의 area에는 상권이 저장되어 있습니다.
- AREA_LIST를 순회하며, area가 일치하는 주문을 조회, count 합니다.

```
for area in AREA_LIST:
    # 계약중인 주문 리스트를 조회, count 합니다.
    under_contract = Kindergarten.objects.filter(
        smart_kindergarten_area=area,
        ordercontract_order_status__lt=5,
        ordercontract_create_date__year__gte=from_year,
        ordercontract_create_date__month__gte=from_month,
        ordercontract_create_date__year__lte=to_year,
        ordercontract_create_date__month__lte=to_month
    ).distinct().count()

    # 계약 완료된 주문 리스트를 조회합니다.
    area_quantity_list = ProductQuantity.objects.filter(
        order_contract_kindergarten_smart_kindergarten_area=area,
        order_contract_order_status=5,
        order_contract_contract_date__year__gte=from_year,
        order_contract_contract_date__month__gte=from_month,
        order_contract_contract_date__year__lte=to_year,
        order_contract_contract_date__month__lte=to_month
    )
```

● 제품별 판매현황

- Product의 Type별 특정 가격대의 주문 수량을 count 합니다.

```
def salesByProduct(from_year, from_month, to_year, to_month):
    # 시즌/복종별 필요한 데이터들을 미리 dictionary에 담아둡니다.
    PRODUCT_TYPE = {
        'winterformal': {
            'title': '동복정복',
            'items': ['jacket', 'vest', 'shirts', 'pants', 'skirt', 'one-piece',
                      'overall', 'dress', 'acc']
        },
        'wintertraining': {
            'title': '동복체육복',
            'items': ['shirts', 'pants']
        },
        'summerformal': {
            'title': '하복정복',
            'items': ['shirts', 'pants', 'skirt', 'acc']
        }
    },
```

```

'summertraining': {
    'title': '하복체육복',
    'items': ['shirts', 'pants']
}
}

product_set = {}

# 주문 완료된 Product의 최종 가격, 수량 정보가 저장되어 있는 데이터인
# ProductQuantity를 조회합니다.
products = ProductQuantity.objects.filter(
    order_contract__order_status=5,
    order_contract__contract_date__year__gte=from_year,
    order_contract__contract_date__month__gte=from_month,
    order_contract__contract_date__year__lte=to_year,
    order_contract__contract_date__month__lte=to_month
)

product = []
# 위에 정의해두었던, 시즌/복종 dict를 순회합니다.
for type in TYPE_LIST:
    keyword = str(type['season'] + type['style'])
    product_set['items'] = []

    temp = {
        'name': PRODUCT_TYPE[keyword]['title'],
        'child': []
    }

    # i번째 시즌/복종의 items에 저장된 list를 순회합니다.
    for key in PRODUCT_TYPE[keyword]['items']:
        # loop를 순회하며, value에 따라 동적으로 쿼리를 생성합니다.
        q = {
            '{0}__{1}'.format('main_fabric_product', 'season'): type['season'],
            '{0}__{1}'.format('main_fabric_product', 'style'): type['style'],
            '{0}__{1}'.format('main_fabric_product_product_type', 'name'): key,
        }
        # 위에서 조회한 ProductQuantity 중에서
        # i번째 시즌/복종, j번째 product_type인 productQuantity를 조회합니다.
        # ex) i=0, j=0일 때, 동복/정복 - jacket인 product
        product_of_type = products.filter(**q)

        # 최종 공급가의 가격을 기준으로, 가격 범위로 판매 수량을 count 합니다.
        list = product_of_type.aggregate(
            price1=Coalesce(Sum(
                Case(When(
                    supply_price__gte=0,      # supply_price가 0보다 크거나 같고

```

```

        supply_price__lt=20000, # 20000보다 작은
        then='quantity')         # quantity의 수량을 count
    ),
    output_field=IntegerField()
), 0), # 판매 수량이 없으면, default는 0으로 지정합니다.
price2=Coalesce(Sum(
    Case(When(
        supply_price__gte=20000,
        supply_price__lte=25000,
        then='quantity')
    ),
    output_field=IntegerField()
), 0),
price3=Coalesce(Sum(
    Case(When(
        supply_price__gte=25001,
        supply_price__lt=30000,
        then='quantity')
    ),
    output_field=IntegerField()
), 0),
# ... 생략
)
list[ 'name' ] =
ProductType.objects.get(name=key).get_name_display()
temp[ 'child' ].append(list)

product.append(temp)
product_set[ 'items' ] = product
return product_set

```

9. WebTex3D Shirts

<http://shirts.stylefor.us>

프로젝트의 위치는 155 서버의 `c:\var_shirt\www` 입니다.

기타 업무

1. 홈페이지 관리

<http://www.texclub.com>

1.1. 관리자 계정

ID : admin

PW : rhksflwk-851

1.2. 입사자/퇴사자 계정 관리

회원 권한에 따라서 인트라넷의 접근이 제한됩니다. 관리자 계정으로 로그인 후, 회원 권한을 설정할 수 있습니다. 신규 입사자의 권한은 5 또는 10으로 설정해주어야 하며, 퇴사자 발생 시 권한을 다시 2로 내려주어야 합니다. 퇴사자가 많으므로 비밀번호를 변경하는 것을 권장합니다.

1.3. 문의 사항 답변

정기적으로 Q&A 게시판을 확인하여 문의사항에 대한 답변을 해주어야 합니다. 개발팀에서 답변을 할 수 있는 내용이 아니라면, 디자인팀이나 정비팀에게 전달을 해주시면 됩니다.

Q&A 게시판 목록입니다.

- 고객센터 > Q&A
- 자격증 > Q&A
- 학생용 특가판매 > Q&A

1.4. 자격증 업무

한국직업연구진흥원(www.kivd.or.kr/)에서 컴퓨터 패션 디자인 운용마스터의 자격 검정 시험을 주관하고 있습니다. 자격 검정 시험의 응시자는 2개월간 배포판 프로그램을 사용할 수 있습니다.

1.4.1. 자격 검정 시험 등록

1. 홈페이지 인트라넷의 고객 관리 시스템에 접속합니다.
2. 오른쪽 상단의 **자격증** 메뉴를 선택합니다.
3. **자격증시험관리** 메뉴를 선택합니다.
4. 시험회차와 일정을 입력한 후 시험을 등록합니다.

자격검정일정은 한국직업연구진흥원 홈페이지에 접속하여 확인할 수 있습니다.

1.4.2. 자격증 응시자 등록

1. 응시자 명단은 진흥원에서 전달 받아야 합니다.
매 회차마다 진흥원에 연락을 해서 요청해야 합니다. (02-3676-0413)
2. 진흥원에서 받은 명단을 csv파일 형태로 저장합니다. 순서는 (이름, 이메일 주소)입니다.
3. 자격증 메뉴의 **자격증응시관리** 메뉴를 선택합니다.
4. 응시자를 등록할 시험 회차와 필기/실기 시험을 선택합니다.
5. 오른쪽 하단의 새자격증응시자등록 메뉴를 선택하고, csv 파일을 등록합니다.

1.4.3. 배포판 프로그램 문의 관련

자격증 응시자가 사용할 수 있는 프로그램을 배포판 프로그램이라고 합니다. 워터마크 출력, 저장 포맷 제한 등 여러 가지 제한 사항이 있는데, 홈페이지의 **자격증 > 배포판** 메뉴에서 확인할 수 있습니다. 시험을 접수 했는데 인증이 되지 않는다고 문의가 오는 경우가 있습니다.

먼저, 등록한 접수자 명단에 고객의 이름이 있는지 확인합니다.

인증 내역에 **자격증 신청 이메일과 홈페이지 이메일이 일치하지 않음**이라는 문구가 있을 경우의 해결 방법입니다.

1. texclub 홈페이지의 회원 정보 페이지에 접속합니다.
2. 고객의 이메일 주소를 자격증 접수 이메일과 동일하게 변경합니다.
(관리자만 회원 정보 변경을 할 수 있으므로, 직접 변경 해주어야 합니다.)
3. 다시 응시관리 메뉴로 접근하여 **인증되어 배포판 프로그램을 사용할 수 있음**으로 문구가 변경되었는지 확인합니다.
4. 정상적으로 인증이 되는지 확인합니다.

접수자 명단을 받지 못했거나, 인증되었다고 보이지만 어떤 문제로 인증이 되지 않는 경우가 있습니다. 그럴 경우에 쿠폰 발급을 해주어야 합니다.

1. 자격증 메뉴의 **쿠폰발급관리** 메뉴를 선택합니다.
2. 쿠폰을 발급 한 후, 마지막 페이지로 이동합니다.
3. 발급된 쿠폰의 쿠폰키를 고객의 메일로 전송해준 후, 쿠폰번호로 인증을 하도록 안내합니다.

1.5. 학생용 특가 프로그램(ASP)

학생용 특가 프로그램은 기간제로 texPro 시리즈를 구매할 수 있는 프로모션입니다. textile, knit, weave 3가지 프로그램을 정식 버전과 동일하게 사용할 수 있습니다. **인트라넷 > 고객관리시스템 > ASP** 메뉴에서 학생용 프로그램의 결제 내역을 확인할 수 있습니다.

1.5.1. 프로그램 실행 관련 문의

ASP 프로그램은 관리 페이지에서 시리얼키를 확인할 수 없기 때문에, 원격 지원을 통해 재설치를 해드리는 방식으로 대응을 하여야 합니다. 고객이 설치 메뉴얼대로 설치를 하지 않았던 문제가 대부분이었습니다. 설치 메뉴얼은 **학생용 특가판매 > 사용메뉴얼**에 나와있습니다.

원격 접속은 윈도우 PC에서는 애니서포트(822.co.kr)를 이용하며, mac의 경우 팀뷰어 프로그램을 이용합니다. (애니서포트는 사내 동시 이용이 불가하므로, 정비팀이나 디자인팀이 원격 지원을 하고 있다면 팀 뷰어 프로그램을 이용해주세요.)

1.5.2. 이메일 수신 문제

간혹, daum이나 hotmail 등 특정 메일은 발송이 되지 않는다고 합니다. 회원 정보 수정에서 다른 이메일 주소로 변경해드린 후, 다시 인증을 할 수 있도록 안내합니다.

1.5.3. 환불 요청 대응

원칙적으로는 환불이 되지 않지만, 그러나 프로그램이 설치되지 않는 문제로 인해 사용하지 않았다는 것이 확인 되면 김도영 과장님에게 전달해드리면 됩니다.

1.5.4. 유플러스 전자결제

pgweb.uplus.co.kr 학생용 프로그램의 결제는 유플러스 전자결제 시스템을 통해 이루어집니다. 상점 관리자 기능을 사용할 일은 거의 없지만, 가끔 학생용 프로그램 취소, 환불 처리를 개발팀에서 처리해주어야 하는 경우가 있습니다. **결제 취소 가이드**(계정 정보는 변경 되었으므로 무시해주세요)를 참고하여 대응합니다.

상점관리자 계정 정보입니다.

상점ID : texclub 사용자 ID : admin 비밀번호 : texclub18@

1.6. 홈페이지 프론트엔드 관리

texclub의 서버 환경은 php + centOS로 구성되어 있습니다. 연혁이나 팝업 수정을 주로 하기 때문에 php를 수정하는 일은 거의 없으며 html+css+javascript로 간단한 수정 작업이 요구됩니다.

WinSCP 또는 cyberduck 프로그램을 실행합니다. 홈페이지를 구성하는 파일들은 `/home/texclub_new`에 위치하고 있으며, 특정 페이지를 수정하기 위해서는 이 경로를 기준으로 주소창과 동일한 경로를 따라가면 됩니다.

1.6.1. 연혁 수정

The screenshot shows the homepage of the Texclub website. On the left is a vertical navigation menu with links like '회사소개', '제품소개', '디자인샘플', etc. The main content area features a large purple background image of a fabric roll. On the right, there are links for 'TEXPRO 드라이버 자료', 'TEXPRO 제품구입', and 'INTRANET'. A sidebar on the right contains links to various services: 'texpro 교육용특가판매', '디자인샘플', '교육동영상', 'POP KOREA 디자인서비스', 'NCS 국가자격증리퍼포먼스', 'FAQ', 'Q&A', '구인구직', and 'Member's Download'.

회사연혁

HOME > 회사소개 > 연혁

2017

- 12 2017 제7회 항조우 국제 의류, 악세서리 가공(OEM/ODM) 전시회 참가 [IMAGE](#)
- 11 2016 심천 국제 의류 OEM/ODM 전시회 참가 [IMAGE](#)
- 11 2016년 부산국제신발섬유페션전시회 참가 및 2016년 중소기업 융복합기술개발사업 1, 2차년도 성과발표 세미나 개최 [IMAGE](#)
- 09 모바일용 섬유 패션 디자인 앱, DPing 2차 리뉴얼 출시
- 06 모바일용 섬유 패션 디자인 앱, DPing 출시
- 2016년 중소기업 융복합기술개발사업 1차년도 성과발표 세미나 개최 [IMAGE](#)
- Centric PLM 세미나 개최 [IMAGE](#)
- 05 제8회 부산국제기계대전 전시회 참가 및 2016년 중소기업 융복합기술개발사업 1차년도 성과발표 세미나 개최 [IMAGE](#)

연혁에 들어가는 내용은 류성훈 이사님이 전달해주시며, 한글, 영문, 중문 페이지 총 3개의 페이지를 수정해야 합니다.

위에서 설명한 대로, 이 페이지를 수정하기 위해서는 `/home/texclub_new/html/company/history.php` 파일을 수정하면 됩니다. texclub 홈페이지는 인코딩이 엉망진창입니다. 그렇기 때문에 텍스트만 수정해야 하는 간단한 작업도 조금 돌아가야 하는 불편함이 있습니다.

texclub 홈페이지의 파일을 수정할 때는 notepad++를 사용해주세요. sublime text 등, 인코딩이 지원되지 않는 에디터로 파일을 수정하면 인코딩이 전부 깨져버리는 문제가 발생할 수 있습니다.

일단, 파일을 수정하는 방식은 다음과 같습니다.

history.php 파일을 열고, 이전에 작성된 것과 같은 형태로 텍스트를 추가하면 됩니다. 혹시 모를 상황에 대비하여, 파일을 수정하기 전에는 백업을 해주세요.

한글과 영문 페이지는 이와 같이 하면 문제가 없지만, 중문 페이지의 경우는 조금 복잡합니다.

```
<table width="637" border="0" cellspacing="0" cellpadding="0">
<tr>
<td height="20">&ampnbsp</td>
<td valign="top"></td>
<td>&ampnbsp</td>
<td valign="top" style="padding-top:3px"><p></p></td>
<td>&ampnbsp</td>
<td>
<p>召#24320;2016年中企&#19994;夕合技&#26415;&#24320;&#21457;事&#19994;1次年成果&#21457;表&#35762;座
<a style='cursor:hand' onclick="MM_openBrWindow('http://www.texclub.com/image/popup/201706/end.php?image_num=01','','width=550,height=430')"><br />
</p>
</td>
</tr>
```

이렇게 알아볼 수 없는 형태의 텍스트가 나타납니다. 이 곳에 똑같은 외계어를 입력하는 형태로 수정이 되어야 하는 데요.

1. 먼저 LibreOffice를 열고 html 파일 형식으로 저장합니다.
2. 전달받은 중국어 텍스트를 붙여넣고 저장 합니다.
3. 저장한 html 파일을 notepad++로 열어줍니다.
4. 다음과 같이 인코딩이 깨진 형태로 보입니다. 이제 p태그 내부의 쓸데없는 태그들을 지우고, 텍스트를 추려냅니다.

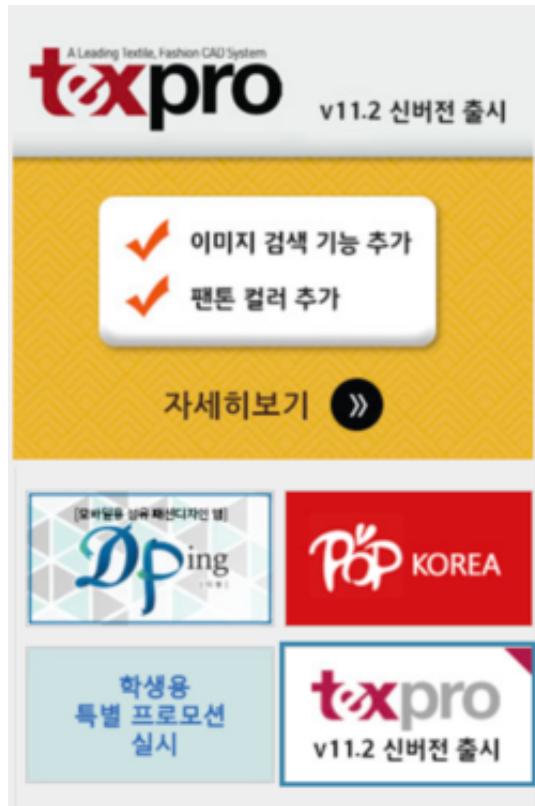
```
5. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr"/>

<title></title>
<meta name="generator" content="LibreOffice 5.1.3.2 (Windows)"/>
<meta name="created" content="2018-01-12T14:55:53.532000000"/>
<meta name="changed" content="2018-01-12T14:56:35.876000000"/>
<style type="text/css">
@page { margin: 2cm }
</style>
</head>
<body lang="ko-KR" dir="ltr">
<p style="font-variant: normal; letter-spacing: normal">
```

```
<font color="#565656"><font size="2" style="font-size: 10pt"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">&#21442;加</span></span></font></font><font face="바탕, serif"><font size="2" style="font-size: 10pt"><span lang="en-US"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">2017</span></span></font></span></font><font size="2" style="font-size: 10pt"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">年&ldquo;釜山&#22269;&#38469;鞋&#19994;及&#32442;&#32455;服&#35013;展&#35272;&#20250;&rdquo;以及召&#24320;</span></span></font></font><font face="바탕, serif"><font size="2" style="font-size: 10pt"><span lang="en-US"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">2016</span></span></font></span></font><font size="2" style="font-size: 10pt"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">年中小企&#19994;&#22805;合技&#26415;&#24320;&#21457;事&#19994;</span></span></font></font><font face="바탕, serif"><font size="2" style="font-size: 10pt"><span lang="en-US"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">1</span></span></font></span></font><font size="2" style="font-size: 10pt"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">次年和</span></span></font></font><font face="바탕, serif"><font size="2" style="font-size: 10pt"><span lang="en-US"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">2</span></span></font></span></font></font><font size="2" style="font-size: 10pt"><font face="Tahoma"><span style="font-style: normal"><span style="font-weight: normal">次年成果&#21457;表&#35762;座</span></span></font></font></font></p>
</body>
</html>
```

6. 추려낸 텍스트를 history.php에 붙여넣으면 완료입니다.

1.6.2. 팝업 수정



팝업창은 `/home/texclub_new/index.php` 파일의 내용을 수정합니다. 108 Line `<!-- 팝업 시작 -->` 부터 팝업창에 관한 코드가 구성되어 있습니다.

\$URL1~4는 popup 레이어를 클릭하면 이동할 url 주소 정보를 담고 있는 변수입니다.

```
<!------- 팝업 시작 ----->
<?
$URL_1 = "http://www.texclub.com/bbs/board.php?
bo_table=bbs_notice&wr_id=208";
$URL_2 = "http://www.texclub.com/bbs/board.php?
bo_table=bbs_notice&wr_id=185";
$URL_3 = "/html/asp/intro.php";
$URL_4 = "http://www.texclub.com/bbs/board.php?
bo_table=bbs_notice&wr_id=206";
?>
```

팝업창과 연결될 게시물을 등록한 뒤, 경로를 복사하여 \$URL 변수에 담습니다.

팝업창에 보여줄 이미지를 `/home/texclub_new/popup/` 경로에 넣어주고, 메인 이미지, 썸네일 이미지 경로를 변경해주면 됩니다.

2. Texclub 서버 관리

texclub은 가비아에서 운영되고 있습니다.

2.1. 서버 계정 정보

ID : root@texclub.com PW: yw_rhksflwk/0

2.2. 서버 접속

2.2.1. 윈도우

putty.exe 또는 winSCP 프로그램을 통해서 접속합니다.

2.2.2. Mac

terminal 또는 cyberduck 앱을 통해서 접속합니다.

2.3. 메일 서버 관리

2.3.1. 메일 계정 관련

신입사원이 들어오면 메일 계정을 생성해주어야 합니다. terminal 또는 putty.exe로 접속한 뒤, 다음과 같이 입력합니다. 비밀번호를 입력해야 할 경우의 명령어도 다음과 같습니다.

```
// 메일 계정 생성
$ adduser 생성할아이디
$ passwd 생성된아이디
$ 비밀번호입력

// 비밀번호 변경
$ passwd 아이디
$ 비밀번호입력
```

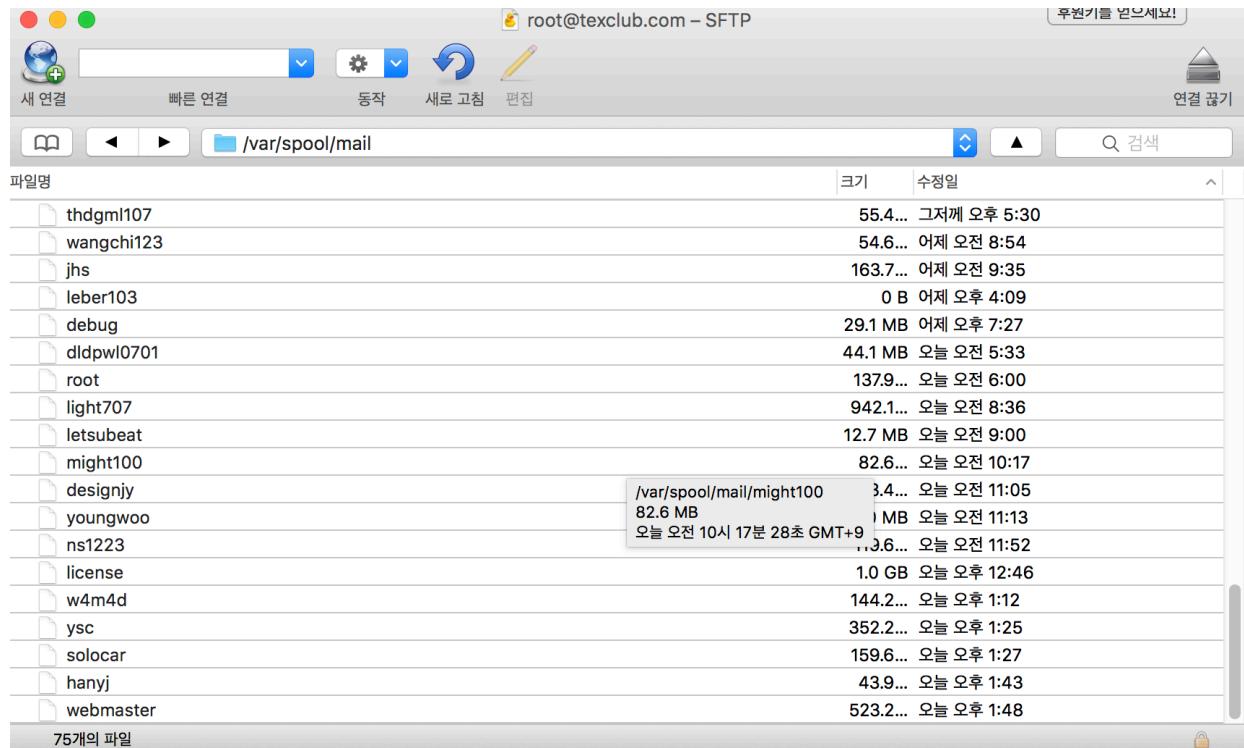
계정이 생성되면 윈도우는 라이브메일, mac은 메일 앱에 계정을 등록해주어야 합니다.

- 이메일 주소 : 생성한ID@texclub.com
- 계정 유형 : POP
- 받는 메일 서버 : texclub.com
- 보내는 메일 서버 : texclub.com

2.3.2. 메일 서버 장애 대응

가비아로 서버를 옮긴 이후 부터는 현재까지 문제가 없었지만, 이전에 테크넷이라는 곳에서 서버를 운영했을 당시에는 메일이 수신되지 않거나 수신된 메일이 반복해서 수신되는 등의 문제가 있었습니다. 대부분은 파일 시스템의 용량 부족으로 인한 문제였는데요. 이럴 경우, 메일 서버의 로그 파일을 지워주면 해결이 됩니다.

/var/spool/mail/ 위치에는 메일의 로그 파일들이 위치하고 있습니다. 용량이 큰 파일들을 선정해서 파일을 삭제 해주어도 되며(메일이 들어오면 자동으로 생성됩니다), cat 명령어를 실행하여 파일을 비워주는 것도 좋습니다.



3. 대량메일 발송

Texclub 유저에게 안내 메일이나 연하장, 100배 활용하기 등을 보낼 때의 메뉴얼입니다.

3.1 계정 정보

<http://youngwoocni.sendmail.cafe24.com/>

ID : youngwoocni PW : texpro2012

3.2 주소록

3.2.1. 주소록 그룹관리

- 대량메일 홈페이지의 주소록 메뉴를 선택합니다.

The screenshot shows the 'Group Management' section of the Cafe24 Mail API. On the left sidebar, under '주소록' (Address Book), there is a '그룹관리' (Group Management) section. The main area displays a table of groups with columns: 그룹명 (Group Name), 메일수 (Number of Mails), 마지막 수정일 (Last Modified Date), and several buttons for managing each group (조회, 그룹명수정, 삭제, 개인추가, 메일보내기). There are 15 entries listed, such as '1-10회 지도사 명단' (1-10th Director List) with 99 members and '170920texpro_user' with 10,130 members. At the bottom of the table, there are navigation links for page 1, 2, 3, and a '▶' button.

- 상단의 그룹추가 버튼을 클릭합니다.
- 그룹선택 select box에서 메일을 발송할 그룹을 선택 합니다.

The screenshot shows the 'Group Selection' page. On the left sidebar, under '주소록', there is a '그룹관리' section. The main area has two tabs: '파일업로드' (File Upload) and '직접입력' (Direct Input). Under '파일선택' (File Selection), there is a dropdown menu '---그룹을 선택하세요---' and two radio buttons: '동일메일주소 건너뛰기' (Skip same email address) and '동일메일주소 추가하기' (Add same email address). Below this is a '새그룹명' (New Group Name) input field. Under '파일 찾기' (File Search), there is a '파일 선택' (Select File) button and a note '* 1회당 최대 10만건까지 추가할 수 있습니다.' (Up to 100,000 items can be added per submission). At the bottom right are '저장' (Save) and '목록' (List) buttons.

- 발송할 대상이 없거나, 간신을 해야 할 경우 새그룹명을 입력하고, 주소록 파일을 등록합니다.

3.2.2. 주소록에 등록할 파일 생성 방법

새 주소록에 추가할 파일을 생성하는 방법입니다.

- Texclub 홈페이지에 관리자 계정으로 로그인하고, 관리자 화면의 회원관리로 이동합니다.

관리자화면
Administrator

HOME LOGOUT

회원관리

- ▶ 회원관리
- ▶ 포인트관리
- ▶ 회원메일발송
- ▶ 접속자현황
- ▶ 투표관리

처음 (총회원수 : 14,731, 차단 : 14, 탈퇴 : 12)

회원아이디	이름	별명	권한	포인트	최종접속	수신	공개	인증	차단	그룹	+	
alsk335	김예진	보라들이이이	2 ▲	0	18-01-31	✓			□			
texprostudy	사용자	리더	2 ▲	0	18-01-29	✓			□			
hej7989	홍은진	아윤맘	2 ▲	0	18-01-28	✓		✓	□			
jang314	장혜진	키퀴	2 ▲	0	18-01-25				□			
skkjhh8357	김지연	김지연123	2 ▲	0	18-01-20				□			
skarnd3121	남궁나연	쌀쌀	2 ▲	0	18-01-26	✓			□			
rkaede11	이미연	lime	2 ▲	0	18-01-23	✓			□			
cyj6213	차영진	차영진	5 ▲	0	18-02-01	✓	✓	✓	□			
simsh78	심서희	xxxxeeerrr	2 ▲	0	18-01-15		✓		□			
2eedahye	이다혜	다혜가다혜	2 ▲	0	18-01-13				□			
zasmeen	이수진	zasmeen	2 ▲	0	18-01-22	✓			□			
euijoo1009	조의주	equix	2 ▲	0	18-01-06	✓	✓		□			
moneylook	송효진	trutre	2 ▲	0	17-12-29				□			
qaz9610	조영아	cloverslove	2 ▲	0	18-01-07	✓			□			
ssm921224	석상민	석상민	2 ▲	0	17-12-21	✓		✓	□			

선택수정 선택삭제

주소록 다운로드 수신동의자만 선택

1 2 3 4 5 6 7 8 9 10 다음 맨끝

2. 하단의 주소록 다운로드를 선택하면, csv 파일이 다운로드 되거나 다음 화면으로 이동합니다.

다운로드 되지 않았다면 오른쪽 마우스를 클릭하여 다른이름으로 저장 -> csv 형태로 저장합니다.

이광수, mrbaen@naver.com
 최영석, ysc@texclub.com
 김수영, yousy@nate.com
 황경원, dongkyung68@hanmail.net
 정재훈, double97@kdiasol.co.kr
 박은정, i840204@nate.com
 이승용, leber103@naver.com
 조향미, youmhuh1004@hanmail.net
 윤정주, blueam2@lycos.co.kr
 김상환, kkh0812@yahoo.co.kr
 최원규, wkchoi23@hotmail.com
 서호영, shyoung0614@nate.com
 김지훈, kjh0761@naver.com
 문경민, shining@texclub.com
 홍일표, hip001_guy@yahoo.co.kr
 강지연, 31101@hanmail.com
 송희원, ztoky@yahoo.co.kr
 조남억, lgf_007@hotmail.com
 신순영, shin1459@hanmail.net
 김은아, kea1974@freechal.com
 윤용권, yyk6803@yahoo.co.kr
 흥영철, narangooo@hanmail.net
 김영욱, theporsche@hanmail.net
 유현자, glala0525@hanmail.net
 김영화, qjqjrnlts@hanmail.net
 이정원, killer790513@hanmail.net
 김민정, kmj1341@hanmail.net
 이지현, ez2@yonsei.ac.kr
 엄기승, asadalto@hanmail.net
 장경혜, dbora@freechal.com
 김 춘 동, ufo2358@big3size.net
 이익근, lyccsun@naver.com
 유재역, indiao512@hanmail.net

3.3 TexPro 100배 활용하기

한달에 한 번, texclub의 user에게 **TexPro 100배 활용하기** 메일을 발송해야 합니다.

- Texclub 홈페이지 > 고객센터 > TexPro 100배 활용하기 메뉴로 이동하여, 발송할 이미지를 첨부하고, 글을 등록합니다.

이미지 파일은 디자인팀에서 작업하며, 아래의 경로에 있습니다. 발송 완료한 이미지는 **전송완료** 풀더로 이동해둡니다. 더이상 이미지가 없다면, 디자인팀에 요청을 하시기 바랍니다.

- \211.238.177.222\디자인팀\01.기타서식자료\02.100배활용하기

- 등록한 글로 이동하여 이미지 주소를 복사합니다.



- 대량메일 페이지로 이동하여, 메일보내기 메뉴를 선택합니다.
- 보내는 사람, 받는 사람, 제목 등을 입력하고, 기존메일 불러오기를 선택하여 최근에 보낸 TexPro 100배 활용하기 메일을 불러옵니다.

5. 메일 제목과 내용의 주제, 첨부 이미지를 변경해줘야 합니다. html 소스로 변환한 후, 이미지의 위치를 찾아서 볼사항 이미지 주소로 변경해줍니다.

6. 마지막으로, 수신동의문구의 날짜를 오늘 날짜로 변경하고 메일을 발송합니다.

4. 가비아

Texclub 서버, DPing 서버, stylefor.us, mobile.stylefor.us의 도메인 등을 가비아에서 관리하고 있습니다. 박진 사원이 관리하고 있던 내용이며, 이예지 사원이 인수인계를 받았습니다. 이 외의 내용은 박진 사원과 이예지 사원의 인수인계 문서를 참고하시기 바랍니다.

4.1. 계정 정보

- 도메인, Texclub 서버의 관리 계정
ID : youngwoocni
PW : texpro17@
- gCloud 서버(DPing 서버)의 관리 계정
ID : youngwoo9
PW : rhksflw-851

4.2. 도메인 관리

- My가비아 > 도메인 > stylefor.us의 서비스정보 관리로 이동합니다.
- 오른쪽의 DNS 레코드 설정 관리를 클릭합니다.

My가비아

1:1문의 | 1544-4370

The screenshot shows the Gabia service management interface for the domain stylefor.us. At the top, there are tabs for '서비스 관리' (Service Management), '서비스 연장' (Service Renewal), '결제 관리' (Payment Management), and 'My 정보 관리' (My Information Management). The main area displays service information for 'stylefor.us' (Domain). It includes the service name, provider (Gabia), creation date (2018-10-23), expiration date (263 days later), and a promotional banner for Gabia services. Below this, the '서비스 정보' (Service Information) section provides detailed service settings, including basic information, contact details, and domain registration information. On the right side, there are sections for managing additional services like DNS records, domain connections, and port forwarding.

- stylefor.us의 서브 도메인을 볼 수 있습니다. 서버의 IP가 변경, 서브 도메인 추가/삭제 작업을 할 수 있습니다.

DNS 레코드 설정

호스트명	IP	비고
stylefor.us	211.238.177.155	
www.stylefor.us	211.238.177.155	
shirts.stylefor.us	211.238.177.155	
webtex3d.stylefor.us	211.238.177.155	
mobile.stylefor.us	182.162.142.159	
fabric.stylefor.us	211.238.177.155	
keyuca.stylefor.us	211.238.177.155	
dping.stylefor.us	211.238.177.155	
map.stylefor.us	211.238.177.155	
demo.stylefor.us	211.238.177.155	

▶ A 레코드(서브 도메인 설정)

호스트명	값
@.stylefor.us	v=Spf1 ip4:211.238.177.155 ~all