시제품 개발 계획서
(Prototype Development Plan)

Ver. 1.1

2016. 09. 29.

한국외국어대학교

정보통신공학과

Intermind(6팀)

문서 정보

구 분	소 속	성 명	날 짜	서 명
작성자	한국외국어대학교	김 형 욱	2016. 09. 28	
	한국외국어대학교	김 진 우	2016. 09. 28	
검토자	한국외국어대학교	정 재 원	2016. 09. 28	
	한국외국어대학교	이 경 민	2016. 09. 28	
	한국외국어대학교	권 영 은	2016. 09. 28	
	한국외국어대학교	김 진 우	2016. 09. 28	
사용자				
승인자	한국외국어대학교	홍 진 표	2016. 09. 29	

개정 이력

버전	작성자	개정일자	개정 내역	승인자
1.0	김형욱, 김진우	2016. 09. 27.	초안 작성	
	검토자	정재원, 이경민, 권영은		
1.1	김형욱, 김진우	2016. 09. 28.	개정내역	
	검토자	정재원, 이경민, 권영은		
1.2				
	검토자			
2.0				
	검토자			

목 차

표 목 차

그 림 목 차

1. 일반

1.1 개요(소개)

O2O Editor(Off-line to On-line Editor)는 종이에 쓴 hand-writing 및 그림을 인식하여 저장 후, 편집가능 한 PC File 형태로 바꿔줌으로써 Off-line(hand-writing 및 On-line(PC Editor) 두 곳 모두에서 편집이 가능한 제품모델을 지향하고 있다.

시제품 개발 계획서란 제품 개발하기 위한 계획을 만드는 것 이므로 계획을 잘 짜려면 제품 구현까지 '우리가 해야 할 일'과 '우리에게 주어진 조건'을 명확히 해야 한다.

'해야 할 일'을 명확히 하기 위해 Top-down 방식으로 고안 하려는 제품(O2O Editor)을 분석해 보았다.

다시 말해, 제품 구현을 위한 전체적인 과정에서부터, 그 과정에 필요한 기술을 분석하는 방식을 택하였다. 따라서 본 문서의 서술 과정 또한 이와 같을 것이다. 따라서 글의 전개는 다음과 같다.

1. 본 제품(O2O Editor)의 전체적인 시스템(O2O-Editor Server + Client) 구조에 관한 서술.

2. 본 제품(O2O Editor)의 Server 구조에 관한 서술.

3. 본 제품(O2O Editor) Server의 Service Procedure에 관한 서술.

4. 위의 명세를 구현하기 위한 대표 기술.

5. 대표 기술에 따른 세부 기술.

와 같은 사항을 분석하며 '해야 할 일'을 명확히 하였고, 제품 구현에 따른 제약 조건 (시간, 개발 비용, 인력)들을 고려하여 시제품 개발 계획서를 구성하였다.

1.2 목적

Off-line(Root)에서 글과 그림을 인식할 뿐 아니라 태그서비스를 이용해 글씨 font 작업과 같은 Editing을 종이(Root)에서 할 수 있는 제품구현을 장기적인 목표로 두고 있다.

구현 과정 속에서 특히, Character Recognition과 같은 부분은 MS, Google, Apple과 같은 Leading Company도 완벽히 구현하지 못하고 있는 실정이다.

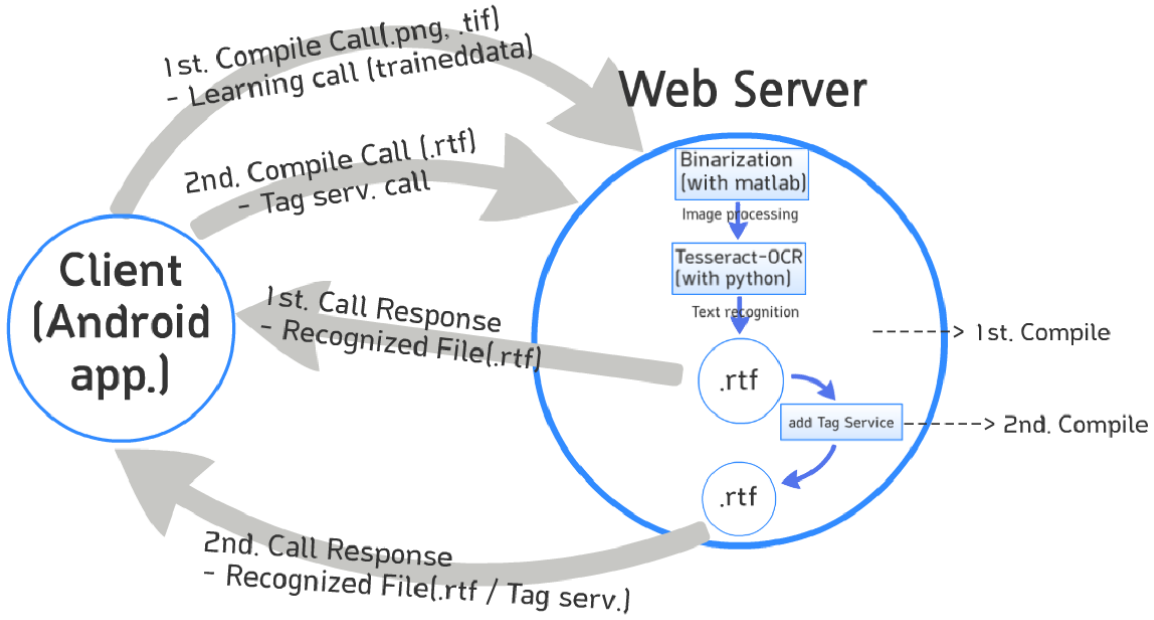
이에, 우리 O2O-Editor 또한 이러한 한계와 어려움을 많은 Research와 자문을 통해 인식하고 인정하는 바이다. 하지만 O2O-Editor는 '지적 호기심'과 '꿈에 대한 도전' 으로 이 분야에 대한 연구를 하게 되었고, 단기적으로는 아래서 언급할 Restriction (e.g. User should write your character in grid which we provide) 을 User에게 줌으로써 사용자의 'Hand-writing pc text화'에 대한 욕구를 조금이나마 먼저 충족시킴을 목표로 하고 있다.

단순히 사용자의 글씨를 인식해서 문서화 하는 것으로 끝나는 것이 아니라, 종이에 적힌 태그서비스를 이용해 인식한 이미지의 내용을 수정 가능한 면에서 에디터의 기능을 제공한다. 또한 결과를 rtf 파일로 제공하여 사용자가 따로 결과를 수정하여 off-line editor의 기능을 부각시킬 것이다.

문서 제목
Code Runner : Solution for Recognition and Execution of Handwritten Code

[Table 1] 관련 문서

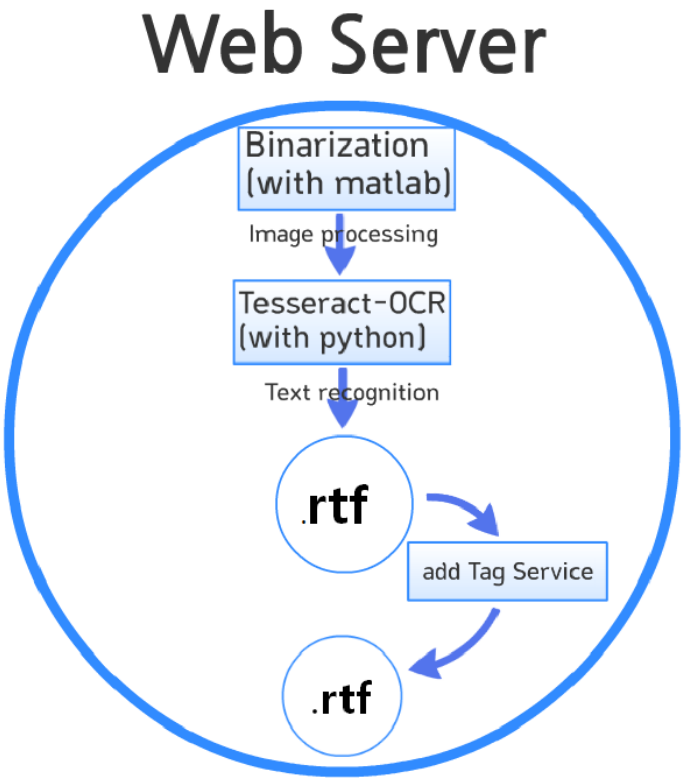
1.3 O2O Editor Whole System Structure



[Figure 1] **시스템 구성도**

Tesseract를 포함한 시스템 주요 과정이 사용자의 핸드폰에서는 수행되기 무리라고 판단되어 주요 과정을 서버에서 진행하는 client - server구조를 선택하게 되었다. 사용자는 서버에 단순히 문자인식을 할 이미지를 보내주고 서버가 주요과정인 문자인식, tag서비스 제공, rtf파일 변환 등의 과정을 수행한다.

1.4 O2O Editor Server Structure



[Figure 2] 서버 구성도

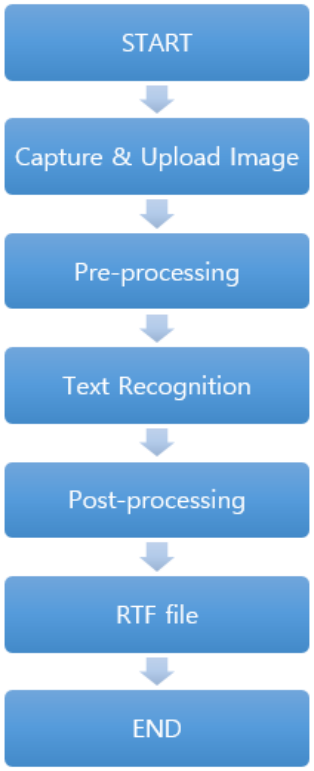
서버의 경우, web 서버를 이용할 예정이며 시스템의 주요 과정을 수행한다. Preprocessing 과정을 통해, 사용자로부터 전송받은 이미지를 문자인식에 유리한 Binarization 이미지로 만들어 준다.

새로 만들어진 이미지는 Text recognition 과정을 통해 사용자의 문자를 인식 후 컴퓨터 텍스트 형태로 변환해 준다. 기존의 OCR엔진은 정형화된 문자만 인식하기 때문에 사용자의 손글씨를 인식하기 위해, 사용자의 글자 데이터를 기존의 OCR엔진의 데이터에 추가한 데이터를 사용한다.

위에 과정 후에도 문자인식률은 완벽하지가 않다. 따라서 조금이라도 문자인식률을 높이기 위해, 자주 사용하는 문자에 대해서 Heuristic 알고리즘을 사용한다.

이러한 문자인식 과정을 거친 인식된 문자들은 태그서비스를 통해 편집된 rtf파일로 사용자에게 제공되거나, 이러한 과정 없이 rtf파일로 사용자에게 제공될 수 있다

1.5 O2O Editor Server Service Procedure



[Figure 3] 서비스 과정

1.6 Software

[Figure 4] 사용 소프트웨어

[사진출처 1] http://www.brainsciencetools.com/?q=content/matlab-course-life-sciences-summer-2016

[사진출처 2] http://developers-club.com/posts/262483/

[사진출처 3] https://a2ua.com/android.html

2. 주요 기술 (인식률 제고)

2.1 Preprocessing

사용자로부터 얻은 이미지를 Binarization(이원화) 과정을 통해 Text recognition에 적합한 이미지로 변환한다.

사용자가 핸드폰 카메라로 사진을 찍는 경우, 다음과 같은 문제가 발생한다.

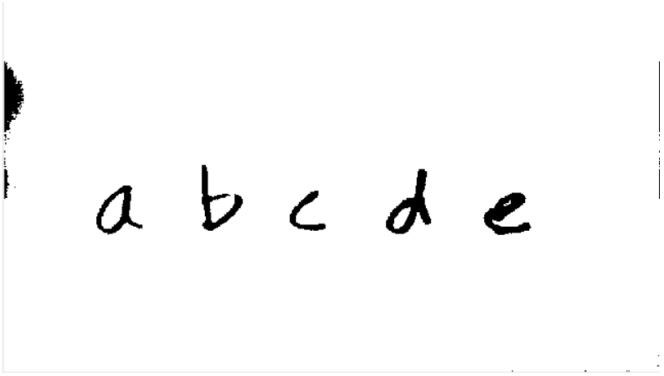
1) 사진마다 조명이 다르기 때문에 사진마다 다른 명암이 생긴다. 이로 인해, 문자 인식률이 떨어지게 된다.
2) 사진에 찍힌 카메라와 종이에 티끌이 노이즈의 원인이 된다.
3) 문자의 가장자리 부분은 낮은 해상도로 인해 흐려진다.

이를 해결하기 위해 다음과 같은 과정을 거치게 된다.

2.1.1 Binarization (2원화)

사진의 특정 부분을 사용하기 위해서는 사진을 기하학적으로 취급해야 한다. 이를 위해서는 2원화(0,1로 표현)된 사진을 사용해야 한다. 색채의 사진으로부터 2원화된 이미지를 얻는 과정을 Binarization (2원화)라고 한다.

이때, 0과 1로 나누는 기준, 에러를 최소화 시키는 임계 값을 threshold라 하는데, threshold는 Otsu 메소드를 이용해 얻을 수 있다. Otsu 메소드를 사용했을 경우, 이미지의 픽셀들을 2개의 클래스로 나누는 데, 두 클래스간에 분산을 최소화 하고, 각각의 클래스 내의 분산을 최대화하는 threshold를 찾는다.



[Figure 5] **Binarization 과정을 거친 이미지**

[출처] [네이버 지식백과] 2 원화 [Binarization, 二元化] (비파괴 검사 용어사전, 2008. 1. 20., 도서출판 노드미디어)

http://terms.naver.com/entry.nhn?docId=443625&cid=42377&categoryId=42377

[출처][Source Factory] [Image Processing] 영상처리 기법 –Otsu Threshold (Otsu 이진화, Otsu binarization)

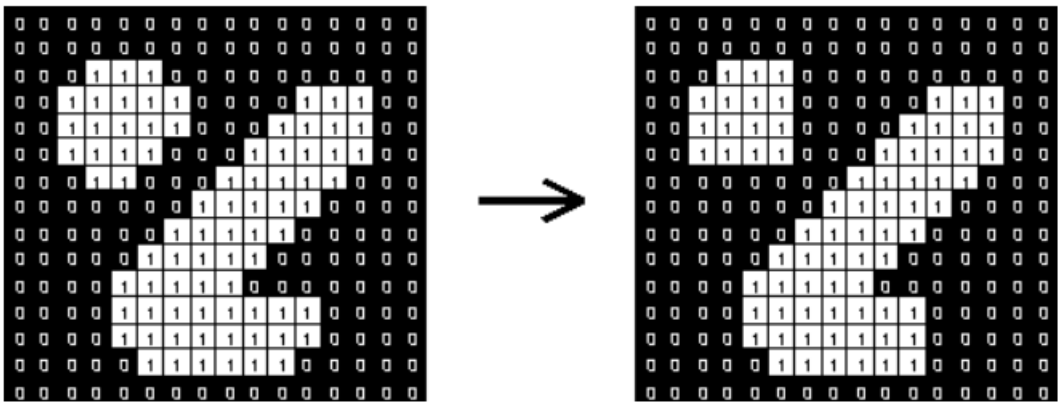
http://mauver.kr/wp/archives/339

2.1.2 Small region removal

노이즈의 원인이 되는 카메라와 종이의 티끌을 제거해 주는 과정이다. 이미지의 특정 pixel을 기준으로 작은 부분을 제거해주는 방식을 사용한다.

2.1.3 Morphological Opening (형태학적 열기)

인식할 문자의 가장자리 부분은 낮은 해상도로 인해 흐려지고, 2원화 과정에서 잘못 분류될 수 있다. 그래서 Morphological Opening을 이용해 경계면은 부드럽게 하여 문자가 더 현실적으로 보이게 만들어 준다.



[Figure 6] **Morphological Opening 과정을 거친 이미지**

[출처][다크 프로그래머]영상 이진화(binarization,threshold) http://darkpgmr.tistory.com/115

[사진 출처] http://homepages.inf.ed.ac.uk/rbf/HIPR2/figs/openbin.gif

2.2 Text Recognition (with Tesseract-OCR)

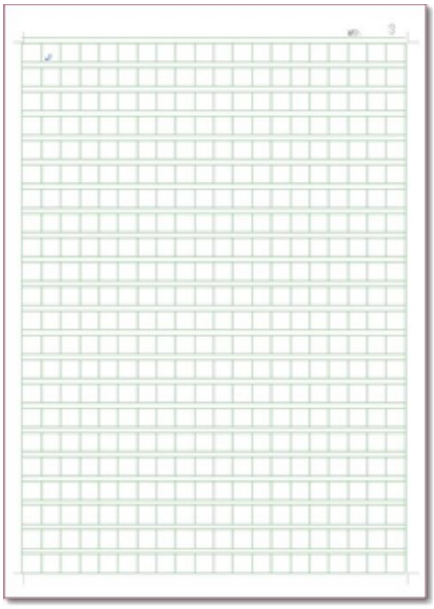
Binarization 과정을 거친 이미지를 Tesseract-OCR 엔진을 이용해 사용자의 글자를 인식한다. 기존에 OCR 엔진 데이터에 사용자의 글자 데이터를 추가하여 문자 인식률을 더 높여준다. 또한 OCR엔진에 더 명확한 사용자의 글자 데이터를 주기 위해 격자 무늬 종이 위에 문자를 입력하는 방식을 사용한다. 이를 통해 OCR엔진은 더 명확하게 문자를 한 글자 단위로 인식하게 된다.

2.2.1 Tesseract-OCR

문자 인식(Text Recognition)은 Tesseract-OCR 엔진을 이용해 광학 문자인식(OCR) 과정을 진행한다. Tesseract-OCR 엔진은 정형화된 글자를 인식하기 위해 디자인 되었다. 또한 여러 대기업 (Google, Apple, MS, ABBYY 등)에서도 손 글씨를 위한 제품은 아직 개발이 되지 않았다.

교내 창업동아리 경진대회에 심사위원으로 온 김영재 바풀(Bapul) CTO로부터 자문을 구해본 결과, 사용자의 traineddata를 기존에 OCR엔진에 추가하면 사용자의 글씨체에 대한 인식률을 높일 수 있다는 점을 다시 한번 확인했다. 하지만 이마저도 정확하지 않는 OCR 엔진의 Layout analysis와 내부적인 문자분할로 인해 정확한 결과를 얻지 못한다.

우리는 이러한 한계를 해결하고자 OCR 엔진이 한 글자 단위로 인식하기 쉬운 격자무늬 종이(원고지 형식) 위에 글자를 입력하는 방식을 사용하고자 한다.



[Figure 7] 격자무늬 종이 (원고지 형식)

[자문] 김영재 바풀(bapul) CTO http://stackoverflow.com/users/361100/youngjae

[사진 출처] http://ming0211.tistory.com/88

2.3 Post-processing

Text Recognition 과정만으로는 완벽한 문자인식을 할 수 없다. 따라서 문자인식률을 더 높이기 위해, post processing 과정이 필요하다. Post-processing은 자주 사용하는 문자에 관해서 Heuristic 알고리즘을 사용한다. 이번 프로젝트에서는 문서 편집 기능을 하는 태그서비스 부분에 Heuristic 알고리즘을 이용해 태그서비스에 대한 문자 인식률을 높인다.

2.3.1 Heuristic 알고리즘

Tesseract-OCR 만으로는 정확한 문자인식이 불가능하다. 따라서 자주 사용하는 문자, 여기서는 문서의 편집기능을 하는 태그서비스에 관해서 라도 문자인식률을 높여주기 위해 Heuristic 알고리즘을 사용한다.

Heuristic 알고리즘은 경험에 기반하여 문제를 해결하거나 학습하거나 발견해 내는 방법을 말한다. 컴퓨터과학 또는 공학에서는 한정된 시간 내에 수행하기 위해 최적의 해 대신 현실적으로 만족할 수 있는 해를 구하는 방법이다. 컴퓨터과학 또는 공학에서 Heuristic 알고리즘(또는 단순히 Heuristic)을 이용하여 실제로 많은 시나리오가 존재할 수 있는 문제의 용인할 수 있는 해(Solution)을 찾아내고 있다. 하지만 그 해가 최적의 해임을 밝히는 형식을 갖춘 증명은 하지 않는다. 증명을 통해서 얻어진 최적의 해는 아니지만 최적의 해에 근사한 해일 것이다. Heuristic 알고리즘은 전형적으로 최적의 해를 찾는 알려진 방법이 없을 때 사

용한다.

이를 이용해 태그서비스에 관한 <text>를 따로 해(Solution)로 모아 놓고, 이와 비슷한 <text>를 기존에 저장해 놓은 해로 대체해 준다.

[출처] [네이버 블로그: editing00] Heuristic과 MetaHeuristic

http://blog.naver.com/editing00/40106277856

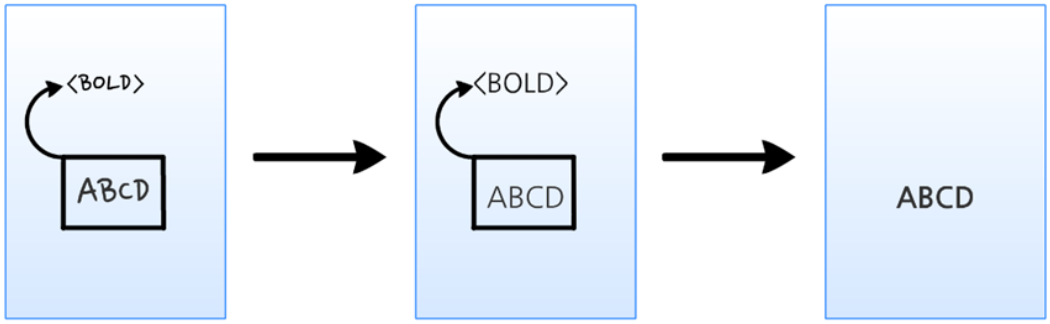
[사진 출처] http://homepages.inf.ed.ac.uk/rbf/HIPR2/figs/openbin.gif

2.4 구현을 위한 세부 기술

2.4.1 Tag service

위에 문자인식과정과 post processing 과정을 통해 생성된 문자를 이용해 태그서비스를 제공한다. 태그서비스는 xml에서 고안한 기술로, 특정 구역에 대해 태그('<>') 부여하여 태그 내부의 attribute에 따른 편집을 하게 하는 기술이다.

이는 O2O Editor만의 editing 기술로, 이로 인해 종이와 펜으로도 문서를 편집할 수 있다.



[Figure 8] Tag Service 동작 과정

2.4.2 rtf 파일 변환

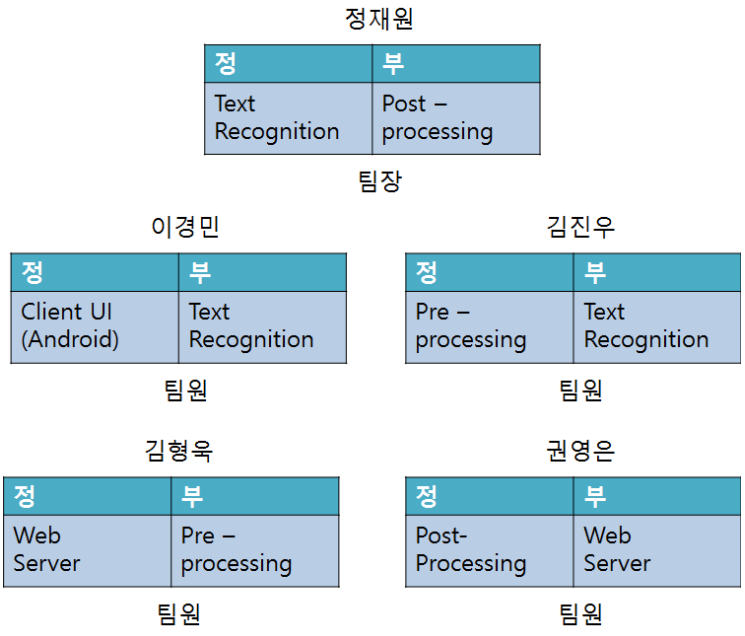
pyRTF 라이브러리를 이용해 사용자가 기존의 문자 인식된 데이터와 태그서비스의 결과를 rtf파일로 생성한다.

3. 세부 추진 계획 및 일정

위에서 구현에 필요한 기술들을 알아보았다. '해야 할 일'을 정리 하자면 Intermind 팀은 위에서 언급한 기술을 연구 해야 할 것이고, 이에 기반해 O2O Editor 구현을 진행 해야 할 것이다. 이에 따라 주어진 조건을 고려하여 세부 추진 계획 및 일정을 작성 하였다.

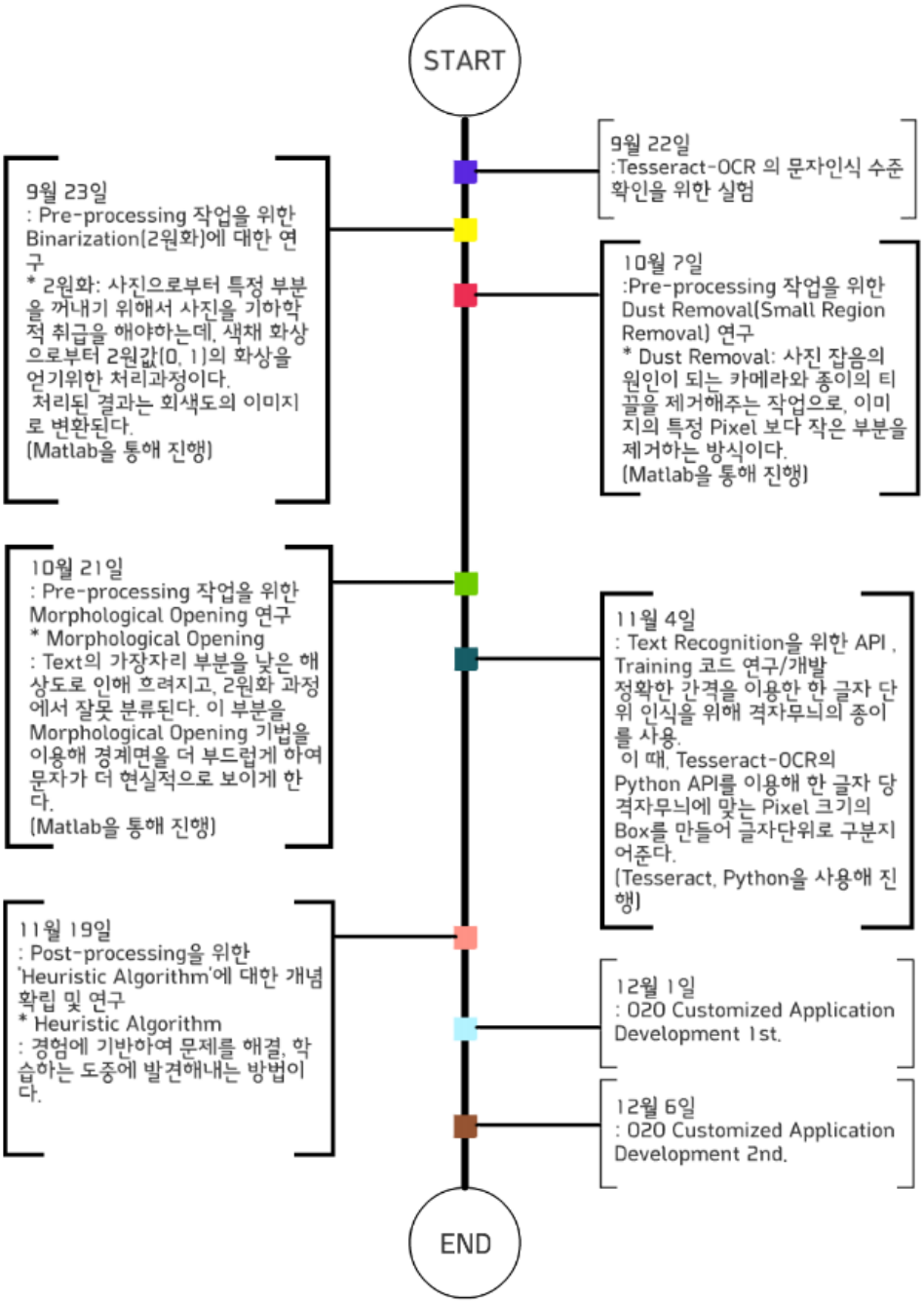
주어진 조건을 고려하면 다음과 같다. 먼저, 우리에게 주어진 인력은 5명, 그리고 주어진 시간은 68일 정도가 된다. 마지막으로 개발 비용을 고려해 볼 것이다.

먼저 주어진 인력에 따른 업무 분담이다.



[Figure 9] 팀원 업무 분담표

또 시간에 따른 세부 일정이다



[Figure 10] 프로젝트 세부 일정

투자 개발 비용에 대해 고려해 보겠다.

본 O2O-Editor는 '소프트웨어 개발' 로 많은 비용을 요하지 않는다. 후에 Web-server에 대한 비용이 고려될 수 있는데, 이 부분은 월 22만원 정도가 예상된다. 이에 따라 초기에는 창업 지원금을 지원 받은 200만원을 사용할 예정이고, 후에는 Web-server 운용 시 광고 배너를 통해 수익을 창출할 것 이다. 또한 인건비에 관한 부분이 고려될 수 있다. 이 부분에 대해선 팀원 모두 '개발에 따른 사용자의 편리 도모' 라는 가치관을 공유하고, 돈에 관해선 욕심이 없기 때문에 개발 후 이익창출 전 까지는 인건비 또한 비용을 요하지 않을 것이다.