

문제정의서(연구계획서)

과제명	자연어 처리 기반 서비스 프로그램 개발
-----	-----------------------

조	02 조
지도교수	정상근 교수님 (서명)
조원	201402330 김연훈 201402387 이동원 201604136 김노은

1. 연구의 필요성

현재 모바일 어플리케이션 시장에는 다양한 종류의 기록 보조 어플리케이션이 존재한다. 그러나 이들 중 최신 AI 기술을 접목해 개인 비서 개념을 유지하는 트렌디한 일정 및 기록 관리 어플은 다수 존재하지만, 여전히 기록의 번거로움을 개선해 주지는 못하고 있으며 1차원적인 기록 및 저장 기능에만 초점을 맞추고 있다는 한계점을 파악하여 본 어플리케이션을 구현하고자 한다.

이를 위해 텍스트 마이닝 및 **Dictionaty** 기반 등의 감성분석 자연어 처리 기술을 활용해 개선된 일기 어플리케이션을 개발할 것이다.

본 어플리케이션을 개발함으로써 얻을 수 있는 기대효과는 다음과 같다. 구글에서 제공하는 구글 포토 서비스는 개인이 저장한 사진이라는 단일 데이터로부터 AI가 자동으로 편집한 동영상 포맷의 추억 비디오로는 가공된 데이터를 제공한다. 이처럼 자연어처리 기반 어플리케이션을 활용해 단순한 텍스트 데이터를 개인이 입력함으로써 개인 심리 일대기나 통계 정보와 같은 머신러닝 기술의 산물을 얻어 새로운 기록 형태의 유행을 주도하고자 한다.

2. 연구의 목표 및 내용

자연어처리(NLP, **Natural Language Processing**)란 인간의 언어 현상을 컴퓨터와 같은 기계를 이용해서 모사할 수 있도록 연구하고 이를 구현하는 인공지능의 주요 분야 중 하나이다. 인공지능 분야의 기술적인 측면뿐만 아니라 연구 대상이 언어이기 때문에 언어 현상의 내적 기재를 탐구하는 언어 인지 과학과 연관이 깊다. 구현을 위해 수학적 통계적 도구를 많이 활용하며 특히 머신러닝 도구를 많이 사용하는 대표적인 분야이다. 자연어 처리는 음성 인식, 내용 요약, 번역, 감성 분석, 텍스트 분류, 질의 응답, 챗봇, 클러스터링, 정보검색 등 다양한 응용이 이루어지고 있다.

해당 어플리케이션은 자연어처리 기술 중 감성 분석(**sentiment analysis**) 기술을 활용해 구현되며, 감성 분석은 크게 다음의 세 단계로 분류할 수 있다. 먼저 다양한 소셜 미디어 매체로부터 분석에 활용할 정보를 수집하는 데이터 수집 단계가 첫 번째이며, 두 번째는 앞서 수집한 데이터로부터 발안자의 핵심 의견을 걸러내는 주관성 탐지 과정이다. 마지막 단계에서는 그동안 추출한 정보로부터 사용자의 감성 데이터를 분석해 분류하는 과정이다. 개발 과정 중 데이터 수집을 위해 소셜 미디어 플랫폼인 트위터 내의 문장을 수집할 계획이며, 데이터 수집

에 사용할 패키지는 **Rtweet**이다. **konlpy**와 **googlelanguageR** 등을 활용해 감성 분석 **API**를 개발하며 안드로이드 스튜디오와 **firebase**를 통해 **UI** 및 사용자 **DB**를 구축한다.

감성 분석 테크닉의 정밀도는 인간의 언어를 컴퓨터가 얼마나 적절하게 이해했는가 여부에 따라 결정된다. 그렇기 때문에 기존에 수행되어 왔던 단순한 '지식의 축적'을 통한 기술과는 차이점을 가지 갖는다. 또한, 본 어플리케이션의 개발 대상인 한국어는 언어 자체가 영어와 같은 다양한 언어와 다르게 복잡한 문법 체계를 가지고 있기 때문에 그 구조와 의미를 분석하는 일에 더욱 깊은 의의를 갖는다. 주어와 동사 및 목적어 등의 기본 문법 어순이 복잡하게 얽혀 있거나 빈번한 요소의 생략이 일어나는 등 문장을 모호하게 하는 변화 체계가 다양하다. 이러한 모호성을 고려하여 원활한 어플리케이션 개발을 수행하기 위해 '형태소 분석', '구문 분석', 그리고 '의미 분석' 순의 개괄적인 절차를 이해한 뒤 한국어 자연어처리 어플리케이션 개발을 수행할 것이다.

3. 연구의 추진전략 및 방법

개발에 앞서 자연어 처리 기술에 관한 논문과 자료들을 수집하고 우리가 하고자 하는 방향에 맞는 방법들을 공부하고 활용한다. 또한 다양한 패키지들을 숙지하여 개발에 있어서 더 수월하게 한다. 고려해봄직한 패키지들은 다음과 같다.

tidyverse

- 데이터 전처리와 시각화에 특화된 패키지 이다.

Stringr

- 문자열 전처리에 특화된 패키지. 정규표현식과 같이 사용하면 효과를 극대화 시킬 수 있다.

googleLanguageR

- 구글 클라우드 플랫폼에서 제공하는 자연어 처리 **API** 인터페이스 패키지 이다.

KoNLP

- 국내 대표적인 자연어 분석 패키지 이다. 한국어를 분석할 수 있는 총 **27**개의 함수가 들어있다.

RcppMeCab & RmecabKo

- **KoNLP**와 마찬가지로 형태소 분석 엔진이다. 다만 활용 시 별도 터미널을 활용하여 설치해야 하는 불편함이 있다.

이러한 패키지들과 오픈소스 기반 텍스트 마이닝 미니 예제코드를 통해 결과물을 내보며 모델을 현한다. 자연어

처리 분야 연구가 수행됨에 있어서 주로 영어 등 국외 언어에 초점을 맞추어 수행된 소스가 많기에 본 프로젝트의 연구 대상인 한국어 자연어처리 기반 소프트웨어 제작을 위하여 한국어 감성 사전 등 공개되어 있는 오픈소스를 적절하게 활용하여 정확도를 높이고자 한다.

클라이언트와 서버간의 통신은 소켓통신을 이용한다. 대부분의 채팅은 **TCP/IP** 방식을 채택한다. 소켓은 응용프로그램에서 **TCP/IP**를 이용하는 창구 역할을한다. 소켓통신에 대한 공부를 마치고 소켓프로그래밍을 통해 다음 사항들에 유의하며 서버를 구축한다. 클라이언트-서버 통신 모델이며 항상 서버 프로그램이 먼저 수행되고 있어야한다. 서버는 **socket()**을 호출하여 통신에 사용할 소켓을 하나 개설하고 이때 리턴된 소켓번호와 자신의 소켓 주소를 **bind()**를 호출하여 서로 연관시켜 둔다. 다음에 서버는 **listen()**을 호출하여 클라이언트로부터의 연결요청을 기다리는 수동 대기모드로 들어간다. 클라이언트로부터의 연결요청이 왔을 때 이를 처리하기 위하여 **accept()**를 호출한다. 서버는 **accept()** 시스템 콜에서 기다리고 있다가 클라이언트가 **connect()**를 호출하여 접속요구를 해오면 이를 처리한다. 연결이 성공적으로 이루어지면 서버는 클라이언트와 데이터를 송수신할 수 있게 된다. 클라이언트는 **socket()**을 호출하여 소켓을 만든 후 **bind()**를 부를 필요 없이, 서버에게 연결요청을 보내기 위하여 **connect()**를 호출한다. 이렇게 서버와 연결이 이루어지면 서버와 데이터를 송수신할 수 있다. 이와 같은 사항들을 단계마다 테스트를 통해 점검하고 원활한 서버-클라이언트 통신이 이루어 질 수 있도록 한다.

어플리케이션 배포 이후 관리 전략은 다음과 같다. 다양한 요구사항을 수행하는 프로젝트이기 때문에 예상하지 못한 요소로부터 오류가 발생할 위험성이 존재한다. 그렇기 때문에 사용자로부터 모델의 성능 측면 외에도 안드로이드 어플리케이션의 기본적인 **UI** 및 구성 등의 디자인 측면에 대하여 최대한 많은 피드백을 받음으로써 주기적인 유지보수를 진행할 것이다. 또한 개인의 기록이나 언어 사용 습관에 따라 특정 어휘의 사용 빈도가 상이하기 때문에 분석 오류를 높이는 고난도 어휘나 신조어 등장에 대비하기 위해 사용자가 개별 태그를 제작해 단어 인식의 정확도를 높이고자 한다.

앞서 기술한 연구내용 및 전략의 효과적인 수행을 위한 협업 전략은 다음과 같다. 버전 관리 프로그램 깃허브를 활용함으로써 효율적인 팀 단위의 코드 작성을 수행할 것이며, **Google docs**나 **Google form**같은 문서들을 적극 활용하여 동시적 문서작업과 설문조사 및 피드백을 수행하도록 한다. 회의를 통해 세부적인 개발 단계를 정하고, 그 단계가 완료될 때 마다 테스트를 진행한다. 잦은 회의를 통해 원활한 의사소통과 다양한 피드백으로 뒤늦은 수정 및 보완을 최대한 줄이며 진행한다.

4. 연구 팀의 구성 및 과제 추진 일정

팀장 : 이동원 팀원 : 김연훈, 김노은

3월

프로젝트 주제 선정과 팀 단위 개발 및 회의 환경 구축

프로젝트 코드 관리 : Github 저장소를 활용한 코드 백업 및 공동 개발

프로젝트 문서 관리 : Google Docs를 활용한 문서 공동 작업

4월

프로젝트 주제 관련 세부사항 논의와 관련 기술 공부 및 숙지

대략적인 프로토타입을 통해 피드백을 받으며 수정 및 보완을 통해 실질적인 설계에 돌입

5월

메모 작성 및 편집 기능을 수행할 수 있는 클라이언트 프로그램(안드로이드 어플리케이션)과

메모를 분석하여 결과를 반환하는 기능을 수행하는 서버 프로그램을 동시 개발

클라이언트 또는 서버의 일정 단위 개발이 완료될 때마다 상호 통신이 잘 이루어지는지 테스트를 진행

6월

5월 작업에 이어 개발을 마무리 짓고 사용자들에게 피드백을 받아 최종적으로 수정을 거친 뒤 마무리한다.

－ 참고문헌(Reference)

소셜 분석, googleLanguageR 소개(Junhewk Kim) :

<https://junhewk.github.io/text/2017/10/05/googlelanguageR-and-novel-analysis/>

소켓 시스템 :

http://jkkang.net/unix/netprg/chap2/net2_1.html

빅데이터분석(김진석):

http://datamining.dongguk.ac.kr/lectures/bigdata/_book/index.html

요구사항명세서

(Software Requirements Specification)

과제명	자연어 처리 기술을 이용한 서비스 프로그램 개발
-----	----------------------------

조	02 조
지도교수	정상근 교수님 (서명)
조원	201402330 김연훈 201402387 이동원 201604136 김노은

Table of Contents

1. Introduction	1
1.1. Purpose	1
1.2. Scope	1
1.3. Definitions, acronyms, and abbreviations	1
1.4. References	1
2. External Interface Requirements	2
2.1. 사용자 인터페이스 (User Interface)	2
2.2. 소프트웨어 인터페이스 (Software Interface)	2
2.3. 통신 인터페이스 (Communication Interface)	2
3. System Features	3
3.1. 사용자 인증 (User Authentication)	3
3.1.1. 설명 및 우선순위 (Description and Priority)	3
3.1.2. 기능 요구사항 (Functional Requirements)	3
3.2. 기록 관리 (Data Management)	3
3.2.1. 설명 및 우선순위 (Description and Priority)	3
3.2.2. 기능 요구사항 (Functional Requirements)	3
3.3. 통계 및 일지 관리 (Statistical and Schedule Management)	3
3.3.1. 설명 및 우선순위 (Description and Priority)	3
3.3.2. 기능 요구사항 (Functional Requirements)	3
4. Other Nonfunctional Requirements	4
4.1. 성능 요구 (Performance Requirements)	4
4.2. 보안 요구 (Security Requirements)	4
4.3. 소프트웨어 품질 속성 (Software Quality Attributes)	5
5. Other Requirements	6
5.1. H/W 제약 조건	6
5.2. 자원, 인력에 대한 제약 조건	6

1. Introduction

1. Purpose

본 프로젝트의 주제는 자연어 처리 기술을 사용하여 기존의 기록 어플에서는 찾아볼 수 없는 새로운 서비스를 제공하는 어플을 만드는 것이다. 기존의 기록 어플과 달리 이 프로젝트 어플은 자연어 처리 기술을 통한 통계 집계 및 시각화 기능과 깔끔한 정렬과 커스터마이징 기능을 통해 개인의 개성 표출 욕구를 충족시켜 주고 텍스트 데이터의 부족한 직관성을 보완 하여 기존 이용자들에게 신선한 경험과 유의미한 통계 집계 서비스를 제공한다.

이를 통해 그동안 기록 어플을 사용해 보았지만 큰 효용성을 못 느꼈거나 단순히 기록만 하고 앞으로 나아갈 지표로 삼지 못하였던 사람들, 그리고 어차피 간단한 메모 들만 그때 그때 눈에 보이는 곳에만 적어 뒀다가 정작 필요할 때 찾지 못했던 사람들, 자신만의 다이어리를 만들고 싶었던 사람들에게 매력적으로 다가갈 수 있을 것이다.

2. Scope

본 어플은 자연어 처리 기술을 이용하여 사용자의 텍스트로부터 데이터를 추출하여 통계를 내고 다양한 포맷의 시각화 도구를 통해 간편하게 볼 수 있도록 표시해 주는 것에 주 목적을 둔다.

기본적으로 다른 기록 어플처럼 기록과 저장 기능을 제공하며 깔끔한 UI를 통해 직관적으로 정렬해 준다.

자연어 처리 기술은 사용자가 기록하는 텍스트 속에서 정보를 추출하고 파악하는데 사용되며, 그 중에서도 일정과 통계, 감정선을 잡는데 주력한다. 일정을 등록하면 해당 내용이 자동적으로 그날 하루 통계에 채워지고 다른 기록들과 합쳐져 매일매일 통계를 낸다. 이러한 통계가 쌓이면 주, 월 단위로 확대가 되고 이를 통해 사용자는 자신의 생활을 시각적으로 볼 수 있게 된다. 이러한 행위적인 통계 이외에 기록에서 사용자의 감정 정보를 추출하여 별도의 감정 그래프를 만들어 보여준다.

또한 사용자가 원하는 대로 테마를 정하고 꾸밀 수 있는 커스터마이징 기능을 제공하여 나만의 일기를 만들고 있다는 느낌을 받을 수 있게끔 한다. 이러한 기능들을 통해 사용자에게 글로 쓰는 일기에서는 파악 하기 힘든 일상을 한눈에 볼 수 있고 깔끔한 기록과 본인만의 감성을 표현 할 수 있는 일기 어플을 제공한다.

자연어 처리 기술이 계속해서 발전하고 여러 분야에서 사용되고 있는 만큼, 보다 더 세세한 추출과 문장 생성을 통해 개인의 맞춤 전기 제작 서비스나 추출 경로를 텍스트에서 음성으로 변환하여 생활 녹음을 통한 일지 작성 등과 같은 확장성 또한 고려해 볼 수 있다.

3. **Definitions, acronyms, and abbreviations**

- 자연어 처리(NLP)

컴퓨터를 이용해 사람의 자연어를 분석하고 처리하는 기술이다.

- 커스터마이징

기본적으로 고객의 요구에 따라 제품을 만들어주는 맞춤제작 서비스를 뜻하며 소비자가 원하는 형태로 재구성, 재설계하는 것을 말한다.

- 파이 차트

범주별 구성비율을 단위 원에 부채꼴로 표현한 그래프이다.

- 워드 클라우드

메타 데이터에서 얻어진 태그들을 분석하여 중요도나 인기도 등을 고려하여 시각적으로 늘어놓아 표시하는 것이다. 시각적인 중요도를 강조하기 위해 중요도(혹은 인기도,빈도수)에 따라 글자의 색상이나 굵기등 형태가 변한다.

- GUI(graphical user interface)

사용자가 편리하게 사용할 수 있도록 입출력 등의 기능을 알기 쉬운 아이콘 따위의 그래픽으로 나타낸 것이다.

- 데이터의 무결성

데이터의 정확성과 일관성을 유지하고 보증하는 것이다.

- 백업

사용자의 실수나 컴퓨터의 오류, 바이러스 등으로 원본이 손상되거나 잃어버릴 경우를 대비해 원본을 미리 복사해 두는 것이다.

- Constraint Layout

복잡한 레이아웃을 단순한 계층구조를 이용해 표현할 수 있는 View Group이다. Relative Layout보다 유연하고 다양한 기능을 제공하고 유지보수와 성능 면에서 뛰어나다.

- Relative Positioning

상대 위치 지정으로 번역되며 레이아웃에 뷰를 배치할 때 뷰 위젯들의 상대적 위치 관계에 따라 뷰가 표시될 영역을 지정하는 방법이다. 레이아웃 중첩으로 인한 성능 이슈 고민을 줄여주고 화면 구성의 복잡성을 낮춰주기 때문에 화면 구성에 소요되는 시간과 비용을 아낄 수 있다.

- Margin

View의 여백과 관계되는 개념으로 뷰 경계와 다른 뷰 사이의 거리를 margin이라 한다.

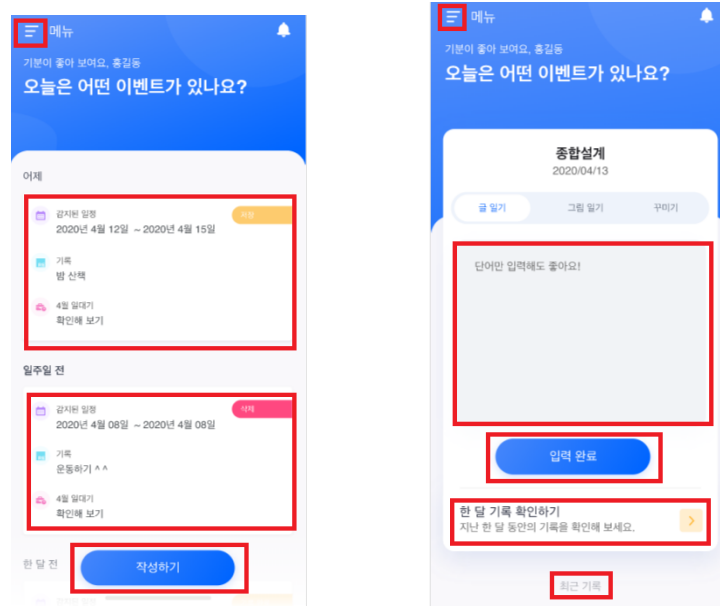
4. References

- Single Word Analysis of Early 19th Century Poetry Using tidytext(Laurence Sonnenberg):
<http://blog.eighty20.co.za/package%20exploration/2017/06/12/sentiment-blog-post/>
- App Store review mining with R + Google Cloud Machine Learning(Fredrik Cederlöf):
<https://towardsdatascience.com/spotify-app-review-mining-with-r-google-cloud-machine-learning-feb6f9c3b75f>
- xwMOOC 자연어처리 - 텍스트:
<https://statklee.github.io/text/nlp-naver-news-mecab.html#>
- “구매후기 한 줄에 고객의 이런 속마음이” 마케팅 난제, 속 시원히 풀어주는 분석(신경식):
https://dbr.donga.com/article/view/1202/article_no/8891
- 소셜 분석, googleLanguageR 소개(Junhewk Kim):
<https://junhewk.github.io/text/2017/10/05/googlelanguageR-and-novel-analysis/>
- 빅데이터분석(김진석):
<http://datamining.dongguk.ac.kr/lectures/bigdata/ book/index.html>

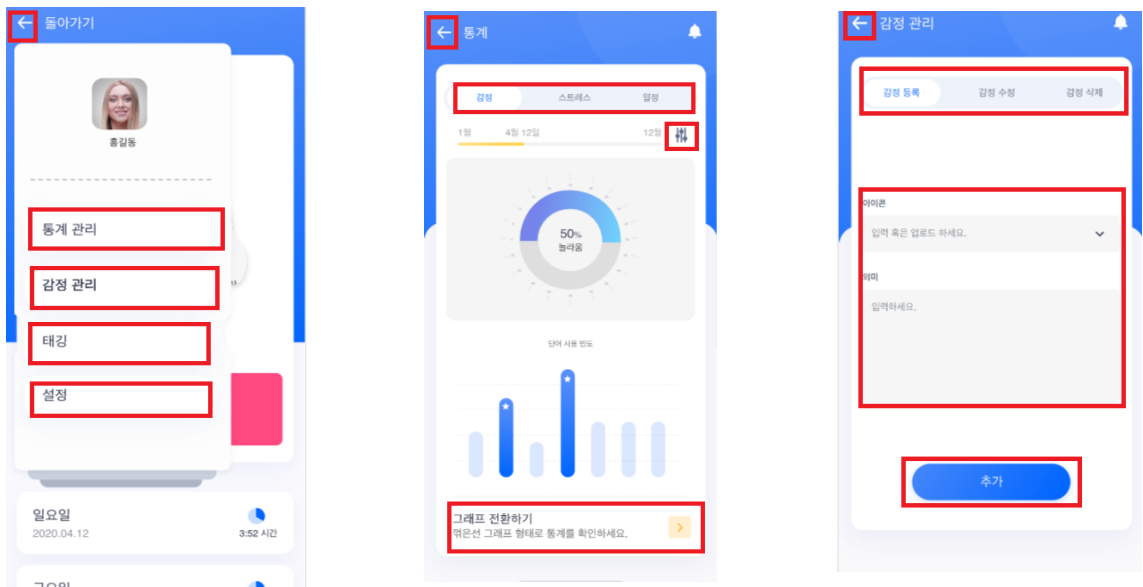
2. External Interface Requirements

5. 사용자 인터페이스 (User Interface)

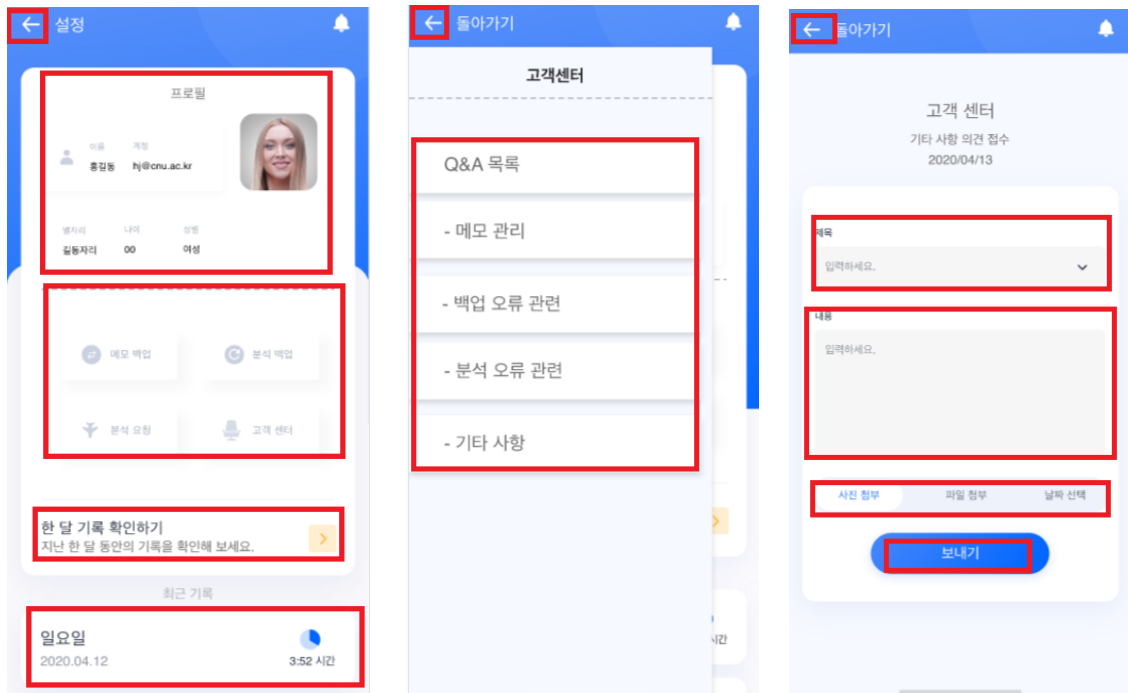
팝업이나 어플 아이콘을 통해 어플리케이션을 호출하며, 다음과 같은 화면을 확인할 수 있다.



기본 화면에서는 그동안 기록했던 기록들을 시간에 따라 정렬해 준다. 기록들은 터치를 통해 확인 및 수정이 가능하다. 사용자는 작성하기 버튼을 통해 작성화면에서 자신의 기록을 저장 할 수 있다. 좌측 상단에 메뉴 버튼을 통해 다양한 통계 서비스에 접근할 수 있다.



좌측 하단 메뉴 버튼을 통해 통계 관리, 감정 관리, 태깅 등의 서비스를 제공한다. 사용자는 각 버튼 터치를 통해 해당 서비스 UI를 확인 할 수 있다. 통계와 감정 관리 서비스 및 기간 등을 설정할 수 있으며, 통계 분석 결과의 시각화 형태 변환 가능하다. 좌측 상단 버튼을 통해 이전 화면으로 돌아갈 수 있으며 안드로이드 핸드폰에서 제공하는 하단 네비게이션 바를 통한 뒤로 가기 역시 가능하다.



설정 버튼을 통해 유저의 프로필 및 각종 설정 메뉴에 접근할 수 있으며 이곳에선 태그 관리 및 서비스 센터 문의 및 각종 메모 관리 메모 분석 관리 등의 역할을 수행 할 수 있다.

6. 소프트웨어 인터페이스 (Software Interface)

자연어처리를 통한 감성 분석(Sentiment Analysis) 기술 구현을 위해 오피니언 마이닝(Opinion Mining)'을 수행하고자 다음과 같은 인터페이스를 구축하도록 한다. 전체적인 연구 절차는 다음과 같은 흐름으로 진행된다. 먼저 소셜 미디어(트위터) 크롤링 이후 데이터 전처리, 감성 분석(Sentiment Analysis) 수행 API 개발, 어플리케이션 개발, 배포 순으로 이루어진다.

해당 소프트웨어가 포함하는 라이브러리 리스트와 각각의 사용 목적은 다음과 같다. rtweet 패키지는 트위터 데이터 송수신을 담당하며, search_tweets(), get_timeline(), get_followers(), get_favorites() 컴포넌트를 통해 인증을 수행한다. tidyverse 패키지를 통해 데이터 전처리 및 시각화를 진행하며 Google Cloud Platform 에서 제공하는 NLP API인 googleLanguageR 패키지를 활용하여 감성 분석을 수행한다. KoNLP와 RcppMeCab & RmecabKo & SentimentAnalysis 패키지를 통해 한국어를 분석 및 형태서 분석, 감정분석을 수행한다.

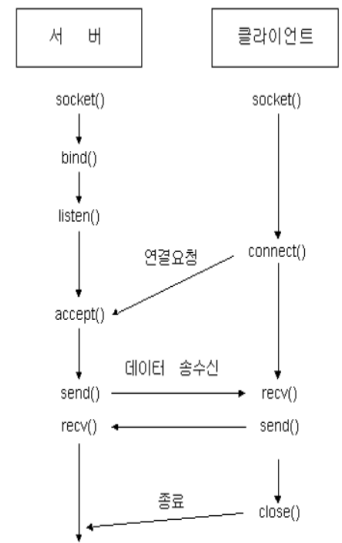
해당 소프트웨어 내 res_sentiment, senti_words_kr 등의 컴포넌트는 크롤링을 통해 수집한 데이터와 분석 라이브러리를 활용하여 감성 분석(Sentiment Analysis)을 수행한다.

7. 통신 인터페이스 (Communication Interface)

소켓통신을 통해 정보를 주고받는다. 서버 측에서 ServerSocket을 만들어 `acpt()`메소드로 클라이언트와 연결을 기다린다. 이처럼 서버에서도 서버 소켓을 이용해서 포트를 열어놓고 클라이언트와 연결을 기다리고, `ServerSocket.accept()`메소드를 통해 클라이언트와 연결이 되는 순간 서버에서 클라이언트 소켓을 생성해서 client 측의 클라이언트 소켓과 통신을 한다.

서버는 많은 클라이언트(사용자)가 접속할 수 있도록 만들어 졌기 때문에 동시에 여러 작업을 하는 경우 작업을 하나씩 하는게 아니라 하나의 클라이언트가 접속하면 Thread를 추가로 생성해서 각 클라이언트들이 수행하는 작업을 각각의 Thread안에서 처리하게 한다.

안드로이드 에서는 UI를 메인스레드에서만 컨트롤 할 수 있기 때문에 Thread를 생성,사용하면서 UI를 컨트롤 하기위해서 Handler을 사용해준다. 기본적인 소켓의 흐름은 우측의 이미지와 같다.



3.System Features

1. 사용자 인증 (User Authentication)

1.설명 및 우선순위 (Description and Priority)

이용자에 따라 입력 데이터를 분리하여 관리하고 발생 가능한 보안 및 성능 이슈를 효과적으로 관리하기 위해 ID, PW 등의 개인정보를 활용한 사용자 인증을 실시한다. 본 기능의 중요도는 상, 중, 하 중 ‘상 단계’에 속한다.

2.기능 요구사항 (Functional Requirements)

요구사항 분류		기능
요구사항 번호		SEF-001
요구사항 명칭		사용자 인증 (User Authentication)
요구사항 상세설명	정의	각 이용자 데이터를 별도 관리할 수 있도록 계정 인증 시스템 정의
	세부 내용	<div>○ 사용자 인증 시 요구하는 정보는 다음과 같다.</div> <div>- 아이디, 비밀번호, 나이, 성별, 생년월일, 복구 이메일 계정</div> <div>○ 계정 규격을 다음과 같이 정의한다.</div> <div>- 비밀번호는 영어 대,소문자 및 특수문자를 필수 1개씩 포함한 10자리 이상으로 구성</div> <div>- 아이디 중복 여부 및 마지막 문자에 특수문자를 포함하지 않는 구성 여부 검사</div> <div>- 아이디 및 등록된 이메일 계정 변경은 불가능하다.</div> <div>○ 별도의 사용자 요청이 없을 시 계정 데이터 보관 기간은 1년 이내로 제한한다.</div>
산출정보		데이터 저장 및 보안 성능 개선
관련 요구사항		

3.System Features

2. 기록 관리 (Data Management)

1.설명 및 우선순위 (Description and Priority)

본 어플리케이션의 기본 개념트를 수행하기 위해 사용자로부터 데이터를 입력 받거나 이를 수정 및 삭제 하며 다음과 같은 요구사항을 준수해야 한다. 본 기능의 중요도는 상, 중, 하 중 ‘중 단계’에 속한다.

2.기능 요구사항 (Functional Requirements)

요구사항 분류	기능	
요구사항 번호	SEF-002	
요구사항 명칭	기록 관리 (Data Management)	
요구사항 상세설명	정의	기본 데이터 조작 시스템 정의
	세부 내용	<div>○ 인증을 거친 사용자로부터 기록 데이터를 입력 받고 일자별로 관리한다.</div> <div>○ 추후 분석 정확도를 고려하여 개인이 태그를 정의할 수 있다.</div> <div>○ 데이터 수정이나 삭제 시 기존 분석 결과에 업데이트가 이루어져야 한다.</div> <div>○ 사용자는 개인의 선호도에 따라 입력 박스 테마를 커스텀할 수 있다.</div> <div>○ 데이터 입력 과정에서 날짜는 변동할 수 없도록 함으로써, 당일 입력/수정/삭제 시도에 의미를 부여한다.</div>
산출정보		데이터 조작 체계 수립
관련 요구사항		

3.System Features

3. 통계 및 일정 관리 (Statistical and Schedule Management)

1.설명 및 우선순위 (Description and Priority)

이용자 데이터에서 추출한 감정 분석 결과에 따라 산출된 통계 및 일정 데이터를 관리 및 제시하도록 하며, 정보 전달의 직관성을 높이기 위해 사용자 선호도에 따라 다양한 시각화 방법을 커스텀할 수 있어야 한다. 본 기능의 중요도는 상, 중, 하 중 ‘상 단계’에 속한다.

2.기능 요구사항 (Functional Requirements)

요구사항 분류		기능
요구사항 번호		SEF-003
요구사항 명칭		통계 및 일정 관리 (Statistical and Schedule Management)
요구사항 상세설명	정의	데이터 분석 시스템 정의
	세부 내용	<div>○ 통계 분석 후 사용자에게 결과 레포트를 제공하며, 포함 정보는 다음과 같다.</div> <div>- 감정의 분포, 기록 연대기, 사용 시간 분포</div> <div>○ 일정 추출 후 사용자에게 한 차례 점검을 진행하여 추출의 정확도를 높이도록 한다.</div> <div>○ 통계 분석 후 사용자가 설정할 수 있는 시각화 테마 종류는 다음과 같다.</div> <div>- 파인차트/ 원 그래프 Pie Chart, 워드 클라우드 차트, 방사형 차트</div>
산출정보		데이터 분석 체계 수립 및 성능 개선
관련 요구사항		

4.Other Nonfunctional Requirements

1. 성능 요구 (Performance Requirements)

요구사항 분류	성능	
요구사항 번호	PER-001	
요구사항 명칭	분석 목표	
요구사항 상세설명	정의	입력 데이터를 활용한 분석 목표 정의
	세부 내용	<ul style="list-style-type: none">○ 사용자가 입력한 정보에서 발생할 수 있는 모든 오류에 대한 메시지를 정보 입력 후 3초 이내에 제시하여야 함<ul style="list-style-type: none">* 단, 대용량 파일 또는 대량통계 조회 시는 예외로 함○ 오류 메시지는 사용자가 인지하여 즉시 조치할 수 있도록 작성되어야 함<ul style="list-style-type: none">- 통계 기능 등 5초 이상 소요되는 작업은 작업 진행사항 디스플레이(Status Bar 또는 팝업)를 통해 사용자에게 내용을 알려야함
산출정보		
관련 요구사항		

요구사항 분류	성능	
요구사항 번호	PER-002	
요구사항 명칭	오류 응답 시간	
요구사항 상세설명	정의	오류 응답 시간 목표 정의
	세부 내용	<ul style="list-style-type: none">○ 사용자 로그인 시 입력한 정보에서 발생할 수 있는 모든 오류에 대한 메시지를 정보 입력 후 3초 이내에 제시하여야 함○ 사용자 데이터 입력 시 작성/수정/삭제 실패 등 발생할 수 있는 모든 오류에 대한 메시지를 정보 입력 후 3초 이내에 제시하여야 함○ 오류 메시지는 사용자가 인지하여 즉시 조치할 수 있도록 작성되어야 함○ 즉각 처리가 불가능한 오류일 경우 고객관리 페이지로 이동하여 즉시 상담 조치할 수 있도록 해야 함
산출정보		
관련 요구사항		

4.Other Nonfunctional Requirements

2. 보안 요구 (Security Requirements)

요구사항 분류	보안 요구사항	
요구사항 번호	SER-001	
요구사항 명칭	기술적 보안	
요구사항 상세설명	정의	기술적 보안 요건
	세부 내용	<div>○ 공통사항</div> <div>- 사용자 계정 보호 및 입력 데이터 암호화 및 보안대책 강구</div> <div>- 비인가자의 접근 및 정보 시스템의 불법적인 접근을 차단하기 위해 일정 관리자에게 제한적으로 접근권한을 부여</div> <div>- 정보시스템 관리자가 업무별, 데이터별 중요도에 따라 나머지 구성원에게 접근 권한을 차등 부여할 수 있도록 하여 운영</div> <div>- 입력 데이터의 무결성과 가용성을 유지하기 위해 백업 계획을 수립·이행하며, 사고 발생시 적시에 복구할 수 있도록 관리체계 마련</div> <div>- 각급기관 도입을 위한 상용 정보보호시스템 보안성 검토 지침(국정원) 준수</div> <div>- 시스템의 안정적인 운영을 위하여 보안취약점 발견 시 분석 및 조치를 수행하여야 함</div>
산출정보	보안관리계획서, 점검내역	
관련 요구사항		

4.Other Nonfunctional Requirements

3. 소프트웨어 품질 속성 (Software Quality Attributes)

요구사항 분류		품질
요구사항 번호		QUR-001
요구사항 명칭		신뢰성(reliability)
요구사항 상세설명	정의	신뢰성 개념 정의
	세부 내용	<ul style="list-style-type: none">○ 시스템은 통상적인 업무시간 동안 가용성을 보장하여야 하며, 시스템 조건이 무엇이든지 간에 모든 채널에 동일한 자료 및 결과를 생성하고 인도해야 함○ 시스템은 정상상태에서 매일 24시간 동안 무중단으로 운영되어야함○ 복구할 수 없는 자료의 손실 혹은 유출로 이어질 수 있는 관리 오류를 방지하고, 오류가 발생하는 즉시 사용자에게 관련 메시지를 공지할 것○ 사용자의 입력 오류나 시스템의 오류 발생 시 오류메시지를 3초 이내에 사용자에게 제시하여야 함○ 에러복구, 장애대책 확보 등 신뢰성 있는 서비스환경을 제공하여야 함
산출정보		과제 계획서
관련 요구사항		

요구사항 분류		품질
요구사항 번호		QUR-002
요구사항 명칭		정확성(correctness)
요구사항 상세설명	정의	정확성 개념 정의
	세부 내용	<ul style="list-style-type: none">○ 시스템은 주기적 학습을 통해 일정 수준 이상의 정확도를 보장하여야 하며, 중도 발생하는 데이터의 수정 및 삭제 결과를 즉각 반영하고 결과 업데이트를 제공해야 함○ 사용자로부터 정확도 의심 의견 수렴 시 하루 이내에 시스템 점검을 진행하여야 함
산출정보		과제 계획서
관련 요구사항		

5.Other Requirements

1.H/W 제약 조건

Android 운영체제를 기준으로 모바일 어플리케이션 개발을 진행하며 필요에 따라 추후에 IOS 버전 배포를 기획할 것이다. 개발 시, ConstraintLayout을 활용해 Relative Positioning 혹은 Margins 등의 제약 조건을 준수해야 한다.

2.자원,인력에 대한 제약 조건

소프트웨어의 개발 기한은 2020년 6월 이내로 제한하며, 이용자 의견 수렴을 위해 해당 기간 동안 ‘시스템 개발’과 ‘설문 후 분석 절차’가 주기적으로 반복되어야 한다.

유스케이스 명세서

(Usecase Specification Document)

과제명	자연어 처리 기반 서비스 프로그램 개발
-----	-----------------------

조	NaLang 조
지도교수	정상근 교수님 (서명)
조원	201402330 김연훈 201402387 이동원 201604136 김노은

Table of Contents

1. Introduction	1
1.1. Objective	1
2. Usecase Diagram	2
2.1. 설정 Diagram	2
3. Usecase Specification	3
3.1. 메모 작성	3
3.2. 메모 수정	3
3.3. 메모 삭제	3
3.4. 감정 편집	3
3.5. 태그 편집	3
3.6. 회원 가입	3
3.7. 메모 백업	3
3.8. 분석 백업	3
3.9. 분석 요청	3
3.10. 고객센터 문의	3

1. Introduction

1.1. Objective

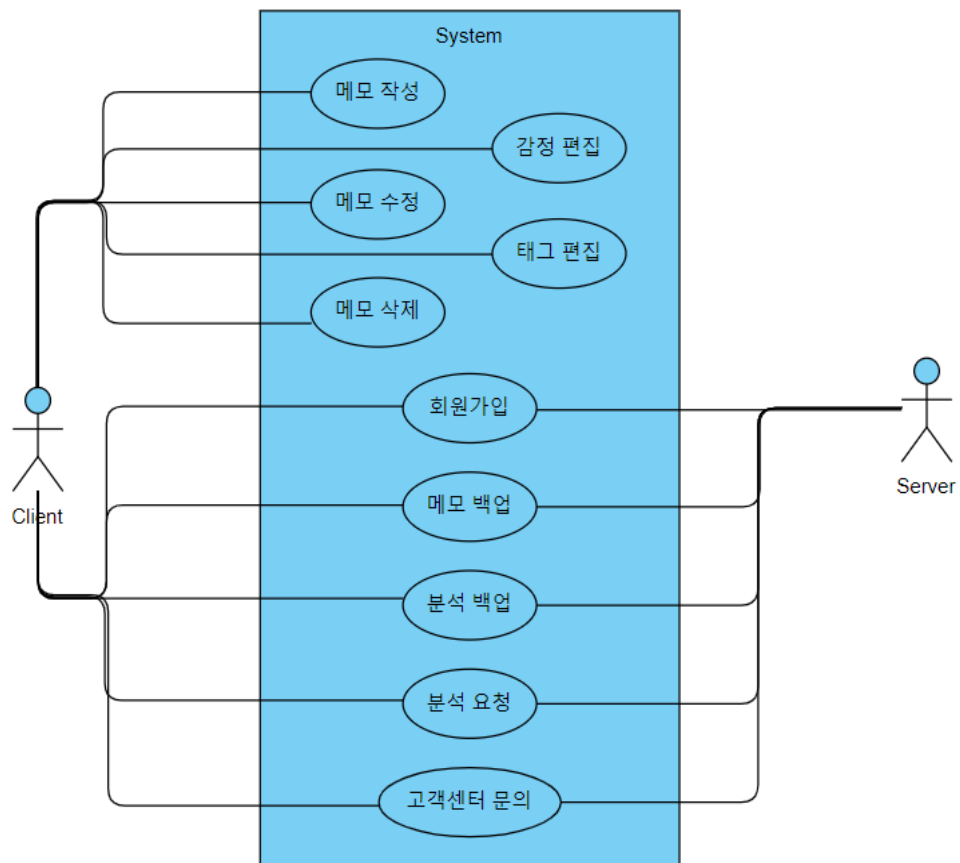
이 문서는 메모 프로그램의 기능을 명시하고 있다. 메모 작성 및 수정 삭제와 같은 기본적인 작성 기능과 회원가입, 메모 분석, 요청 메모 및 분석 내용 백업 등 서버와의 통신 기능에 대한 유스케이스 다이어그램을 작성하였고 각 유스케이스에 대한 명세를 포함하였다.

2. Usecase Diagram

2.1. 설정 Diagram

채팅 프로그램에서 설정 기능을 수행하는 서브시스템에 대한 유스케이스 다이어그램은 다음과 같다.

그림 1. 설정 서브시스템에 대한 유스케이스 다이어그램



3. Usecase Specification

3.1. 회원 가입

Usecase 이름	회원 가입
ID	USE_01
간략 설명	본 어플리케이션을 사용하기 위한 회원 가입 절차에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	- 사용자는 어플리케이션을 설치한다.
Main Flow	1) 사용자는 성명, 생년월일, 전화번호를 입력 후 서버에 인증 번호를 요청한다. 2) 서버는 사용자에게 SMS 로 인증 번호를 전송한다. 3) 사용자는 전송된 인증 번호를 회원 등록 화면에 입력하고 인증을 요청한다. 4) 서버는 사용자를 확인하고, 사용자를 등록한다. 5) 서버는 사용자 등록을 완료한 뒤, 사용자에게 이름을 요청한다. 6) 사용자는 자신의 이름을 입력하고, 완료 버튼을 누른다. 7) 서버는 사용자의 이름을 저장한 뒤, 등록에 성공하였음을 알린다.
Post-Conditions	- 사용자는 회원으로 등록된다.
Alternative Flow	3-1) 인증번호가 잘못되어 인증에 실패하였다. - 사용자는 서버로부터 다시 인증 번호를 요청하여, 다시 인증 과정을 수행한다. 6-1) 사용자가 아무런 입력을 하지 않고 완료 버튼을 눌렀다.

	<ul style="list-style-type: none"> - 사용자에게 이름이 입력되지 않았음을 알리고, 다시 입력을 요청한다. <p>1~7) 사용자가 회원 등록 과정 중 프로그램을 종료하였다.</p> <ul style="list-style-type: none"> - 유스케이스를 다시 시작한다.
--	--

3.2. 메모 작성

Usecase 이름	메모 작성
ID	USE_02
간략 설명	메모를 작성하는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	<ul style="list-style-type: none"> - 사용자는 회원가입이 되어있는 상태이다.
Main Flow	<ol style="list-style-type: none"> 1) 사용자는 메모를 입력한다 2) 저장과 동시에 해당 내용은 서버로 전송된다. 3) 사용자에게 정상적으로 저장 되었음을 알린다.
Post-Conditions	<ul style="list-style-type: none"> - 사용자가 입력한 텍스트는 정상적으로 서버로 전송되어야 한다.
Alternative Flow	-

3.3. 메모 수정

Usecase 이름	메모 수정
ID	USE_03
간략 설명	이미 작성이 완료된 하나의 메모를 수정하는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	<ul style="list-style-type: none">- 사용자는 회원가입이 되어있는 상태이다.- 저장한 메모가 있어야 한다.
Main Flow	<ol style="list-style-type: none">1) 사용자는 수정 버튼을 통해 수정 화면에 진입한다.2) 메모를 수정한다.3) 저장 버튼을 통해 수정된 메모를 저장하고 서버로 전송한다. 이때 메모를 처음 작성했던 시간 정보는 바꾸지 않는다.
Post-Conditions	<ul style="list-style-type: none">- 수정된 메모는 서버로 전송되고, 기존의 메모를 대체해야 한다.
Alternative Flow	<p>3-1)저장 버튼을 누르지 않고 수정 화면을 이탈하려 할 경우</p> <ul style="list-style-type: none">- '바뀐 내용이 존재합니다. 저장하시겠습니까?' 팝업을 띄워 사용자에게 다시 확인을 받는다.

3.4. 메모 삭제

Usecase 이름	메모 삭제
ID	USE_04
간략 설명	작성이 완료된 메모들 중 하나를 선택하여 삭제하는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	<ul style="list-style-type: none">- 사용자는 회원가입이 되어있는 상태이다.- 작성되어 저장한 메모가 있어야 한다
Main Flow	<ol style="list-style-type: none">1) 사용자는 삭제하고 싶은 메모를 골라 삭제버튼을 누른다.2) 시스템은 '삭제하시겠습니까?' 팝업을 통해 다시 한번 확인을 받는다3) 사용자는 '예' 버튼을 누른다4) 메모는 삭제되는 동시에 해당 변화를 서버에 전송한다.
Post-Conditions	<ul style="list-style-type: none">- 삭제된 메모는 정상적으로 반영이 되어야한다.
Alternative Flow	<p>3-1) 사용자가 '아니오' 버튼을 누르는 경우</p> <ul style="list-style-type: none">- 삭제를 진행하지 않는다.

3.5. 태그 편집

Usecase 이름	태그 편집
ID	USE_05
간략 설명	서버의 메모 분석을 돕기 위해 사용자에게만 의미 있는 단어에 태그를 붙이는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	- 사용자는 회원가입이 되어있는 상태이다.
Main Flow	1) 사용자는 메뉴를 통해 태그 편집에 진입한다. 2) 사용자가 원하는 태그와 그 태그로 분류될 단어를 입력한다. 3) 저장 버튼을 누르면, 해당 내용을 서버로 전송하는 동시에 전송이 마무리 될 때 까지 '전송중...' 이라는 팝업을 띄운다. 4) 전송이 마무리되면 완료되었다는 메시지와 함께 팝업창을 변환한다. 5) 사용자는 처리되었음을 확인하고 확인 버튼을 누른다.
Post-Conditions	- 서버에 있는 모델은 전송 받은 사용자가 원했던 단어를 지정한 태그에 반영한다.
Alternative Flow	2-1) 텍스트가 아닌 이모티콘을 입력할 경우 - 시스템은 텍스트만 입력해 달라는 경고창을 띄운다.

3.6. 메모 백업

Usecase 이름	메모 백업
ID	USE_06
간략 설명	사용자 데이터 손실을 막기 위해 서버에 백업해 두는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	- 사용자는 회원가입이 되어있는 상태이다.
Main Flow	<ol style="list-style-type: none"> 1) 메모 백업은 수동과 자동 백업으로 분류된다. 2) 사용자는 어플리케이션 내에서 기록 데이터를 작성한다. 3) 기록 후 작성 버튼을 눌러 데이터를 저장한다. 4) 작성과 동시에 어플리케이션은 기록 데이터를 서버에 송신한다. 5) 서버는 데이터의 일정 중복 여부를 검사하고, 데이터를 저장한다. 6) 어플리케이션은 사용자에게 기록 완료 팝업을 제시하며, 자동 백업을 수행한다 7) 사용자는 메모 카테고리에서 백업 버튼을 누른다. 8) 어플리케이션 내 버튼 클릭 시점까지 보유하고 있던 데이터를 서버에 송신한다. 9) 서버는 갱신된 범위의 데이터를 새로 덮어쓰워 저장한다. 10) 어플리케이션은 사용자에게 기록 완료 팝업을 제시하며, 수동 백업을 수행한다.
Post-Conditions	- 사용자 데이터가 서버에 저장된다.
Alternative Flow	<p>4-1) 서버 동기화에 오류가 발생해 백업이 수행되지 못한다.</p> <p>사용자에게 재클릭을 요청한다.</p>

3.7. 분석 백업

Usecase 이름	분석 백업
ID	USE_07
간략 설명	분석된 결과를 서버에 백업하는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	<ul style="list-style-type: none"> - 사용자는 회원가입이 되어있는 상태이다. - 사용자가 기록 데이터 통계 분석을 실시한다.
Main Flow	<ol style="list-style-type: none"> 1) 분석 결과 데이터 백업은 수동과 자동 백업으로 분류된다. 2) 어플리케이션은 보유한 기록 데이터를 바탕으로 추출한 감정 분석 결과에 대한 통계 분석을 수행한다. 3) 사용자는 분석 데이터 백업 버튼을 눌러 데이터를 서버에 저장한다. 4) 버튼 클릭과 동시에 어플리케이션은 기록 데이터를 서버에 송신한다. 5) 서버는 요청된 데이터를 새로 덮어쓰워 저장한다. 6) 상기 52 에서 54 번 과정을 거쳐 어플리케이션은 사용자에게 백업 완료 팝업을 제시하며 수동 백업을 수행한다. 7) 어플리케이션은 통계 분석 후 10 분이 경과하면 데이터를 서버에 송신함으로써 자동 백업을 수행한다. 8) 어플리케이션은 새로운 메모 작성 혹은 수정, 삭제 발생 시 데이터를 서버에 송신함으로써 자동 백업을 수행한다. 9) 서버는 요청된 해당 데이터를 새로 덮어쓰워 저장한다.
Post-Conditions	<ul style="list-style-type: none"> - 통계 분석 데이터가 서버에 저장된다.
Alternative Flow	8-1) 작성/수정/삭제 시 통신 오류로 백업이 수행되지 않는다.

	- 사용자에게 백업 권장 알림 팝업을 제시한다.
--	----------------------------

3.8. 분석 요청

Usecase 이름	분석 요청
ID	USE_08
간략 설명	작성한 메모들을 서버에서 분석하도록 요청하는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	<ul style="list-style-type: none"> - 사용자는 회원가입이 되어있는 상태이다. - 사용자가 기록 데이터를 작성/수정/삭제한다.
Main Flow	<ol style="list-style-type: none"> 1) 분석 요청은 수동과 자동 요청으로 분류된다. 2) 사용자는 어플리케이션을 통해 기록 데이터 작성한다. 3) 사용자는 데이터 분석 요청 버튼을 눌러 분석을 수행한다. 4) 버튼 클릭과 동시에 어플리케이션은 해당 시점까지의 기록 데이터를 서버로 송신한다. 5) 서버는 보유하고 있는 데이터를 활용해 감성 분석을 실시한다. 6) 처리 결과를 바탕으로 추출된 감성 정보를 활용하여 통계 결과를 제시한다. 7) 사용자는 선호도에 따라 통계 분석 결과 시각화 방법을 변경한다. 8) 어플리케이션을 선택된 시각화 방식을 반영한다. 9) 사용자는 어플리케이션을 통해 기록 데이터 수정/삭제를 수행한다. 10) 어플리케이션은 해당 시점까지의 기록 데이터를 서버로 송신한다.

	11) 상기 5 번에서 8 번 과정을 반복한다.
Post-Conditions	- 기록 데이터 통계 분석이 수행된다.
Alternative Flow	6-1) 통계 수치에 오류가 발생한다. 사용자는 서버 및 고객센터에 정정 요청을 보낸다.

3.9. 고객센터 문의

Usecase 이름	고객센터 문의
ID	USE_9
간략 설명	사용자가 프로그램 사용에 관한 문의를 진행하는 과정에 대해 명세한다.
Actor	Client(Initiator), Server
Pre-Conditions	<ul style="list-style-type: none"> - 사용자는 회원가입이 되어있는 상태이다. - 사용자가 어플리케이션 활용 중 오류 및 문의사항을 접한다.
Main Flow	<ol style="list-style-type: none"> 1) 사용자는 어플리케이션 내 고객관리 메뉴를 통해 문의사항을 작성한다. 2) 사용자는 작성 완료 버튼을 누른다. 3) 버튼 클릭과 동시에 어플리케이션은 서버에 송신을 요청한다. 4) 서버는 송신자 계정 정보와 문의 내용을 저장한다. 5) 처리와 미처리로 분류된 데이터는 관리자 열람 여부에 따라 재분류된다. 6) 응답 정보는 계정 내 이메일 주소를 통해 전송된다. 7) 서버는 문의사항 요청 데이터를 저장한 뒤, 요청이 처리되었음을 알린다.
Post-Conditions	- 문의사항이 관리자에게 전달된다.

Alternative Flow	<p>3-1) 사용자가 아무런 입력을 하지 않고 전송 버튼을 눌렀다.</p> <p>사용자에게 문의 내용이 입력되지 않았음을 알리고, 다시 입력을 요청한다.</p> <p>3-2) 사용자가 회원 등록 과정 중 프로그램을 종료하였다. 유스케이스를 다시 시작한다.</p>
------------------	---

유튜브 링크 : <https://www.youtube.com/watch?v=07HBnE06fzo&feature=youtu.be>

깃허브 링크 : https://github.com/yeonhunkim/Reminder/tree/master/design_sprint/week4