캡스톤디자인I 중간보고서

<table>
<tr><td rowspan="2">제 목</td><td colspan="2">국문</td><td colspan="3">팀 프로젝트 분석 소프트웨어(팀 메이트)</td></tr>
<tr><td colspan="2">영문</td><td colspan="3">Team Project Analysis Software(TeaMate)</td></tr>
<tr><td rowspan="3">진 행
상 황</td><td>중요마일스톤</td><td colspan="4">● 팀 메이트 전체적 기능 정리 및 구현 가능성 판단
● 데이터 정제 기능 - 소스코드 작성
● 데이터 분석 기능 - 클래스 설계서 및 소스코드 작성
● 인터페이스 - UI 프로토타입 설계
● 테스트 - 시나리오 작성</td></tr>
<tr><td>진행상황</td><td colspan="4">● 전체 시스템 구성도 작성 완료
● 테스트를 위한 진로체험 프로그램 계획 수립 완료
● 기능 구현을 위한 소스코드 작성 중
● 클래스 설계서 작성 완료
● 각 역할을 수행하기 위한 학습 수행 진행 중</td></tr>
<tr><td>산출물</td><td colspan="4">요구사항 정의서(별첨 1), 중간보고서(별첨 2)</td></tr>
<tr><td rowspan="4">팀
구성원</td><td>학년</td><td>학 번</td><td>이 름</td><td colspan="2">연락처(전화번호/이메일)</td></tr>
<tr><td></td><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td></td><td colspan="2"></td></tr>
<tr><td colspan="6"></td></tr>
</table>

[별첨1]

프로젝트명 : Team Project Analysis Software(TeaMate)

소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더): 윤선희
이재윤
한세영
대표 연락처: 010-6648-9609
e-mail: ysh5260@naver.com

목차

1. 개요
2. 시스템 장비 구성요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항

1. 시스템 개요

1) 목표

팀 프로젝트를 수행하면 평가를 하더라도 평가에 불만을 가지는 사람이 있어서 이러한 불만을 줄이고, 채팅 기록을 통해 팀 프로젝트의 기여도를 그래프로 시각화해 형평성 있는 평가에 도움을 주는 소프트웨어를 개발하고자 함

- 시퀀스 다이어그램(sequence digram)

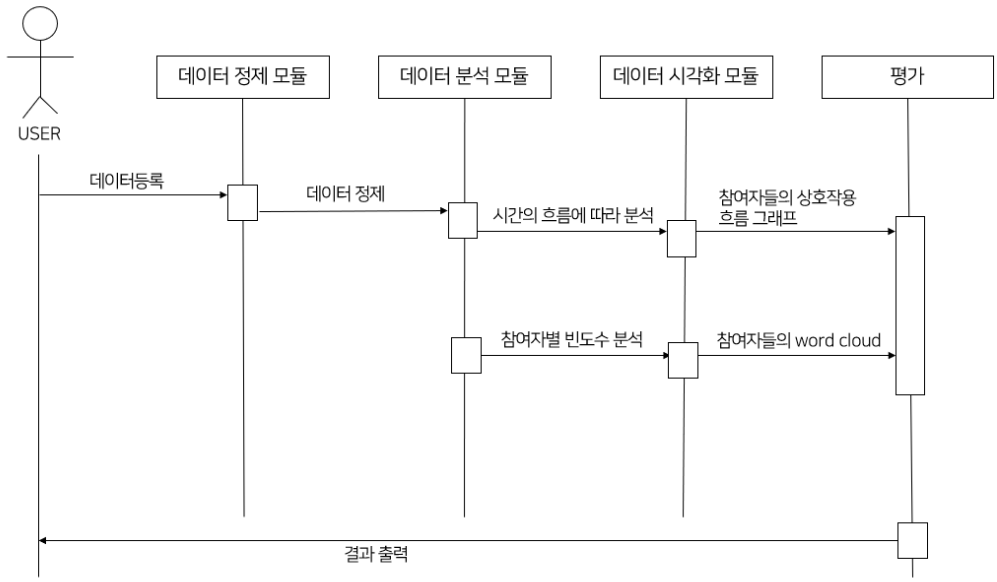

그림 1 시퀀스 다이어그램

- 시나리오

1. 팀 프로젝트를 수행한 후 채팅 데이터를 평가자에게 제출
2. 제출된 데이터를 정제
   2-1. 의미 없는 단어, 의성어 삭제
3. 정제된 데이터를 분석한다.
   3-1. 참여자 간 상호작용을 시간의 흐름에 따라 분석
   3-2. 참여자들의 채팅 참여도 측정
4. 측정된 값을 가지고 그래프로 시각화
5. 평가자는 출력된 그래프를 참고하여 평가 가능

2) 소프트웨어 전제 조건

1. 채팅 데이터는 텍스트(txt) 파일로 입력받음
2. 정제 기능에서 키워드를 설정한 후 단어를 삭제한다는 것은 욕설, 의성어와 같은 단어로 함
3. 정제된 데이터에서 참여자 간 상호작용한 대화가 많으면 프로젝트 참여율이 높은 것으로 가정
4. 채팅 데이터를 통해 주제에 관련된 단어와 대화의 빈도가 높은 순으로 기여도가 높다고 가정
5. 정제된 데이터로 참여자 간, 전체채팅 word cloud를 제작하여 공통된 부분이 많은 단어들을 주제 단어로 선정
6. 주제 단어를 많이 말한 참여자는 기여도가 높은 것으로 판단

3) 시스템 구성도

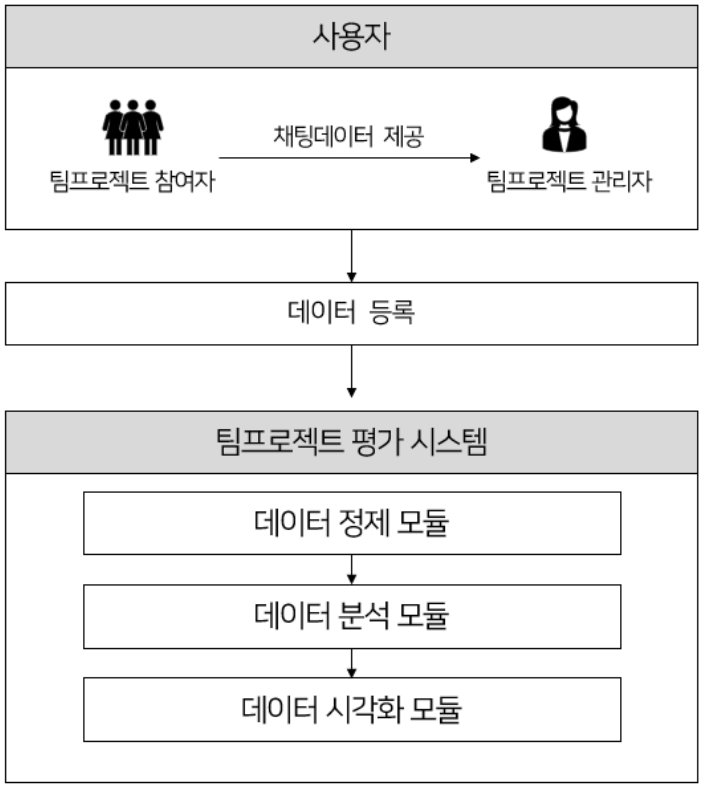

그림 2 TEAMate 구성도

● 팀 프로젝트 평가 시스템
- 사용자가 참여자의 데이터를 평가 시스템에 등록한다.
- 등록하고 나면 팀 프로젝트의 평가 시스템에서 데이터를 정제, 분석한 후 시각화하여 정량적 평가를 위한 수치를 나타내어 준다.

● 데이터 정제 시스템
- 수집된 채팅 데이터를 정제하는 시스템
- 데이터 정제는 의성어 삭제, 의미 없는 대화를 삭제한다.

● 데이터 분석 시스템
- 정제를 거친 채팅 데이터를 가지고 팀원별 상호작용, 참여도를 측정한다.
- 측정한 값을 개인별, 팀원별로 word cloud와 막대그래프로 측정값을 시각화한다.

● 평가 시스템
- 위의 과정을 거친 후 출력된 결과를 가지고 팀 프로젝트 참여자를 평가한다.

2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001		
요구사항 명칭		소프트웨어 사용 PC	응락수준	필수
요구사항 분류		시스템 장비 구성		
요구사항 상세 설명	정의	사용자 PC		
	세부 내용	- 장비 품목 : 사용자 컴퓨터 - 장비 수량 : 1 - 장비 기능 : 소프트웨어 사용 - 장비 성능 및 특징 • OS : Windows 10 pro • CPU : Intel i5-9400F (6 Core, 2.90 GHz) • RAM : 16GB • GPU : NVIDIA GeForce GTX1660		

3. 기능 요구사항

1) 데이터 정제

요구사항 고유번호		SFR-001		
요구사항 명칭		데이터 정제 기능	응락수준	필수
요구사항 분류		기능		
요구사항 상세 설명	정의	- 수집된 데이터를 정리 및 정제		
	세부 내용	- 채팅 데이터 입력 - 사용자 파일 별로 형태소 분리 - 의미 없는 대화 삭제 - 키워드 설정 후 채팅 내 단어 삭제		

2) 데이터 분석

요구사항 고유번호		SFR-002		
요구사항 명칭		데이터 분석 기능	응락수준	필수
요구사항 분류		기능		
요구사항 상세 설명	정의	- 수집 및 정제된 데이터를 분석		
	세부 내용	- 개인별 대화 빈도수 측정 - 참여자 간 대화 상호작용 측정 - 주제에 대한 단어의 연관성 분석		

3) 데이터 시각화

요구사항 고유번호		SFR-003		
요구사항 명칭		데이터 시각화 기능	응락수준	필수
요구사항 분류		기능		
요구사항 상세 설명	정의	- 분석된 데이터를 그래프, word cloud로 출력		
	세부 내용	- 참여자별 word cloud를 출력 - 참여자별 word cloud를 비교 - 주제에 관한 대화 개인별 기여도를 막대그래프로 출력 - 출력된 결과물을 가지고 참여도를 평가		

4. 성능 요구사항

요구사항 고유번호		PER-001		
요구사항 명칭		메모리 linear		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	- 채팅 참여자가 많아질수록 메모리 linear가 높아진다.		
	세부 내용	- 메모리 누수를 줄이기 위해서 채팅 참여자가 많을수록 메모리도 linear 하게 증가시킨다.		

요구사항 고유번호		PER-002		
요구사항 명칭		CPU 점유율		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	- 다른 프로그램의 실행을 방해하지 않기 위해 CPU 점유율을 줄인다.		
	세부 내용	- 멀티코어가 아닌 싱글코어를 이용해서 CPU 점유율이 90%를 넘어가지 않게 한다.		

5. 인터페이스 요구사항

요구사항 고유번호		SIR-001		
요구사항 명칭		데이터 입력 및 정제 파일		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세설명	정의	데이터 채팅 입력 파일과 정제 기능 동작 시		
	세부 내용	- 채팅 데이터 입력, 정제, 분석은 텍스트(txt) 파일을 기준으로 한다. - 채팅 데이터는 사용자 UI에 채팅 파일 위치를 작성되도록 한다.		

요구사항 고유번호		SIR-002		
요구사항 명칭		데이터 출력		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세설명	정의	데이터에 대한 Word cloud와 그래프 출력		
	세부 내용	- 전체 대화와 참여자별 Word cloud, 개인별 기여도 막대그래프, 참여자별 상호작용을 추정한 그래프는 사진(png) 파일로 저장된다. - 해당 출력 파일은 입력 텍스트(txt) 파일과 같은 위치에 저장된다.		

6. 데이터 요구사항

요구사항 고유번호	DAR-001		
요구사항 명칭	초기자료 구축		
요구사항 분류	데이터	응락수준	필수
요구사항 상세 설명	- TEAMate 시스템의 테스트를 수행하기 위해 초기 데이터를 구축하여야 한다. - 진로 체험 행사를 통해 초기 데이터를 얻는다 - 학생들을 개별적으로 식별할 수 있도록 이름이 겹치지 않도록 학번을 사용하도록 한다. - 카카오톡 PC 버전의 대화 내용 내보내기 기능을 통해 만들어진 데이터를 사용한다.		

요구사항 고유번호		DAR-002		
요구사항 명칭		데이터 수집 요구사항		
요구사항 분류		데이터 요구사항	응락수준	필수
요구사항 상세설명	정의	데이터 수집 요구사항		
	세부 내용	- 과거 조사 방법의 장단점을 분석하여 현장 조사 방법 선정 - 수집자료 : 수집 대상 명, 자료 형태, 수량, 생성연도		

요구사항 고유번호		DAR-003		
요구사항 명칭		초기 데이터 구축		
요구사항 분류		데이터 요구사항	응락수준	필수
요구사항 상세설명	정의	데이터 수집		
	세부 내용	- 초기 데이터는 진로 체험 행사를 통해 수집한다.		

요구사항 고유번호		DAR-004		
요구사항 명칭		데이터 무결성		
요구사항 분류		데이터 요구사항	응락수준	필수
요구사항 상세설명	정의	데이터 무결성 적합성		
	세부 내용	• 데이터 무결성 요구사항 - 정합성 규칙 작성/ 데이터 분석, 정제, 관리 개선 방안 도출 - 허용되지 않는 사용자나 자료의 변경을 제어 • 데이터 정확성 및 무결성 정의 - 데이터 이관에 필요한 안정적인 하드디스크 공격 확보 - 데이터의 연계 시, 데이터의 정합성 및 무결성을 체크하고 로그 유지		

7. 테스트 요구사항

요구사항 고유번호		TER-001		
요구사항 명칭		단위테스트		
요구사항 분류		데이터 요구사항	응락수준	필수
요구사항 상세설명	정의	단위테스트 요구사항		
	세부 내용	• 단위테스트 수행 절차 - 단위시험의 범위, 수행 절차, 조직, 일정, 시험환경 및 평가 기준을 구체적으로 수립(시나리오, 처리 절차, 수행 데이터, 예상 결과 등을 사전 정의) - 단위시험 시에 다음 내용을 점검 *결함 유형 분석 *결함발견 추세 분석(시험일시, 발견결함 수)		

요구사항 고유번호		TER-002		
요구사항 명칭		통합테스트		
요구사항 분류		데이터 요구사항	응락수준	필수
요구사항 상세설명	정의	통합테스트 요구사항		
	세부 내용	• 통합테스트 일반사항 - 시나리오에 따라 단위테스트가 완료된 프로그램들을 대상으로 검증 - 결함을 파악하고 원인을 추적하여 결함을 제거 - 통합테스트 시나리오에 대한 설명과 시연을 동해진행 - 실제 운영환경과 같게 구성하여 실시하고 연계를 포함		

요구사항 고유번호		TER-003		
요구사항 명칭		통합테스트		
요구사항 분류		데이터 요구사항	응락수준	필수
요구사항 상세설명	정의	통합테스트 요구사항		
	세부 내용	• 통합테스트 진행을 위한 요구사항 - 발생 가능한 상황에 대해서 시나리오를 작성하여 테스트 데이터를 입력하여 테스트해야 하며, 각종 유형별로 테스트 계획서 완성 - 테스트 결과 및 조치내역의 이력 관리 - 통합테스트 진행 시 실제 사용자 또는 이에 따르는 수준의 테스트를 진행 • 통합테스트 관리 - 결함을 파악하고 추적하여 결함을 제거		

8. 보안 요구사항

요구사항 고유번호	SER-001		
요구사항 명칭	개인 정보 삭제		
요구사항 분류	보안	응락수준	필수
요구사항 상세 설명	- 채팅 내 사진, 파일 제거 - 소프트웨어 실행 중 생성되는 정제된 데이터 파일은 실행 완료 시 제거 - 인터페이스 결과 화면에서 채팅 데이터에 접근 불가		

요구사항 고유번호	SER-002		
요구사항 명칭	채팅 로그 조작 방지		
요구사항 분류	보안	응락수준	필수
요구사항 상세 설명	- 주제에 관련된 내용을 말한 사람들을 오름차순 그래프로 출력 - 그래프 출력 시 주제와 관련된 단어를 아래에 출력		

9. 품질 요구사항

요구사항 고유번호		QUR-001		
요구사항 명칭		프로그램 신뢰성	응락수준	필수
요구사항 분류		품질		
요구사항 상세 설명	정의	- 프로그램 관리 및 운영		
	세부 내용	- 데이터가 누락되는 경우가 발생하지 않도록 신뢰성 검사를 통해 방지한다.		

요구사항 고유번호		QUR-002		
요구사항 명칭		프로그램 안전성	응락수준	필수
요구사항 분류		품질		
요구사항 상세 설명	정의	- 프로그램 관리		
	세부 내용	- 테스트 기간 발생한 결함과 오작동을 검출하여 기록하고 보완한다. - 장시간 통합테스트를 통해 오버플로 및 오류 발생을 테스트한다.		

요구사항 고유번호		QUR-003		
요구사항 명칭		프로그램 효율성	응락수준	필수
요구사항 분류		품질		
요구사항 상세 설명	정의	품질 관리(기술 관점)		
	세부 내용	- 애자일 개발을 사용하지만 하나의 기능이 끝날 때마다 문서화한다.		

요구사항 고유번호		QUR-004		
요구사항 명칭		산출물 관리	응락수준	필수
요구사항 분류		품질		
요구사항 상세 설명	정의	품질 관리(프로젝트 관리 관점)		
	세부 내용	- 산출물 및 각종 안내서(시스템 운영자, 사용자 안내 등)의 관리방안을 제시해야 한다. - 개발 및 커스터마이징 보고서를 제출해야 한다. - 사업자는 사업 추진 과정에서 추진되는 생산물, 산출물에 대해 종류, 내용, 작성 및 제출 시기 등을 제시해야 한다.		

10. 제약 사항

요구사항 고유번호		COR-001		
요구사항 명칭		시스템 개발	응락수준	필수
요구사항 분류		제약사항		
요구사항 상세 설명	세부 내용	- 프로그래밍 언어 : Python - 사용 라이브러리 : Konlpy, WordCloud, Matplotlib - 개발 도구 : Colaboratory		

요구사항 고유번호		COR-002		
요구사항 명칭		업무 모듈화 및 자원 활용 방안	응락수준	필수
요구사항 분류		제약사항		
요구사항 상세 설명	세부 내용	- 유연성, 확장성을 확보할 수 있도록 모듈화 개발전략을 반영함 - 현재 활용 가능한 소프트웨어를 최대한 활용하며 수행함		

11. 프로젝트 관리 요구사항

요구사항 고유번호		PMR-001		
요구사항 명칭		프로젝트 수행 조직	응락수준	필수
요구사항 분류		프로젝트 관리		
요구사항 상세 설명	세부 내용	- 윤선희 : Matplotlib를 이용한 데이터 시각화(그래프 추출) - 이재윤 : Python(Konlpy)를 이용한 데이터 수집 및 정제 코드 설계 - 한세영 : Python(Wordcloud)를 이용한 데이터 분석(주제 파악) 코드 설계		

요구사항 고유번호		PMR-002		
요구사항 명칭		프로젝트 일정계획	응락수준	필수
요구사항 분류		프로젝트 관리		
요구사항 상세 설명	세부 내용	- 데이터 수집 : 7월 이전 - 코드 작성 : 8월 이전 - 테스트 및 디버깅 기간 : 9월 - 논문 작성 : 10월		

[별첨2]

중간보고서

Ⅰ. 요구사항 정의서에 명세 된 기능에 대하여 현재까지 분석, 설계, 구현(소스코드 작성) 및 테스팅한 내용을 기술하시오.

1. 프로젝트 개요

1.1. 프로젝트 목적

팀 과제는 대학이나 중·고등학교에서도 많이 이루어지고 있다. '팀 과제 수행 경험에 관한 연구' 논문에 따르면 팀워크의 질을 높이기 위해서는 소통과 상호협력이 가장 중요하다고 말한다. 현재의 팀 과제 수행 평가는 결과를 가지고 평가하는 방식이다. 평가 방식에 결과뿐만이 아닌 팀 과제 수행과정도 평가한다면 결과만을 가지고 평가한 것보다 형평성 있는 평가를 할 수 있을 것이다.
따라서 이 프로젝트는 팀 과제 수행에서 사용된 채팅 데이터를 가지고 참여자 간의 의사소통 상호작용과 채팅의 빈도수를 측정하는 것으로 팀 과제 수행과정에서 참여자가 얼마나 기여했는지를 정량적으로 평가할 수 있도록 한다.

1.2. 프로젝트 전제 조건

- 데이터 수집
  - 데이터 수집을 위해 진로 프로그램 연계를 해야 함.
- 기능 구현 학습
  - 데이터 정제 및 분석 기능을 구현하기 위해 python을 학습
- 관리 및 개발
  - 개발 산출물을 관리하여 보고서 작성
  - 소프트웨어 유지가 원활하게 이루어지게 함
  - 지속적인 소프트웨어 오류 측정 후 보수
- 원활한 의사소통
  - 개발 방향 설정 및 목표 설정을 위해 지속적인 소통과 회의가 필요

1.3. 시스템 구성도

- 사용자가 팀 프로젝트 참여자로부터 채팅 데이터를 받아 평가 시스템에 입력한다.
- 팀 프로젝트 평가 시스템
  - 사용자가 참여자의 데이터를 평가 시스템에 등록한다.
  - 등록하고 나면 팀 프로젝트의 평가 시스템에서 데이터를 정제, 분석한 후 시각화하여 정량적 평가를 위한 수치를 나타내어 준다.
- 데이터 정제 시스템
  - 수집된 채팅 데이터를 정제하는 시스템
  - 데이터 정제는 의성어 삭제, 의미 없는 대화를 삭제한다.
- 데이터 분석 시스템
  - 정제를 거친 채팅 데이터를 가지고 팀원별 상호작용, 참여도를 측정한다.
  - 측정한 값을 참여자 별로 word cloud와 막대그래프로 측정값을 시각화한다.
- 평가 시스템
  - 위의 과정을 거친 후 출력된 결과를 가지고 팀 프로젝트 참여자를 평가한다.

1.4. 기능별 상세 요구사항

- 데이터 정제 : 입력된 데이터를 정리 및 정제
  - 사용자 채팅 기록을 텍스트(txt) 파일로 입력받는다.
  - 삭제해야 할 키워드(의미 없는 대화, 욕설 의성어 등)를 삭제한다.
  - 데이터의 형태소를 Konlpy로 나눠 전체 대화와 참여자별 대화를 텍스트(txt) 파일로 저장한다.
- 데이터 분석 : 정제된 데이터를 분석
  - 정제된 전체 대화와 참여자별 대화를 Word cloud로 출력해 비교한다.
  - 비교하여 공통된 높은 빈도수의 단어를 주제 단어로 선정한다.
  - 참여자별 시간에 따른 대화 상호작용 추정하여 저장한다.
- 데이터 시각화 : 분석된 데이터를 Word cloud, 그래프로 출력
  - 전체 대화와 참여자별 Word cloud를 출력한다.
  - 주제 단어에 관한 대화 개인별 기여도를 막대그래프로 출력한다.
  - 참여자별 상호작용을 추정한 그래프를 원과 화살표 그래프로 출력한다.

Ⅱ. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

2. 프로젝트 수행

2.1 프로젝트 추진전략

2.1.1 개발전략

기존의 프로젝트들은 과정을 평가하지 않고 수행결과 만으로 평가하였다. 수요자, 프로젝트 참여자의 중심에서 해당 소프트웨어를 제작한다. 프로젝트의 과정을 평가하기 위해 채팅 내용의 기여도를 측정하는 소프트웨어를 제작하여 좀 더 공정한 평가를 할 수 있도록 기획하였다. 이 소프트웨어의 적용 대상은 중·고등학교 학생들과 대학교 교양 수업 학생들을 대상으로 적용한다.

2.1.2. 실증전략

소프트웨어 테스트 진행 시 실제 적용을 확인하기 위해서 중·고등학생들에게 직접 적용하기로 하였다. 중·고등학생들에게 진로 수업 시간에 VR 수업을 진행한다. VR content 관리 플랫폼을 이용하여 학생들이 VR 방 탈출을 팀별로 직접 참여하게 한다. 학생들의 자리를 분리해서 Kakao 채팅으로만 대화를 시켜 운영하고, 해당 채팅 데이터를 받아서 소프트웨어를 테스트하고 설문조사를 받아서 해당 프로그램과 비교함으로써 소프트웨어의 정확도를 측정하여 오차를 줄이고, 정확도를 높인다.

2.2. 프로젝트 수행방법

- 대면 회의
  - 주 1회 회의
  - 주간 개발상황 점검 및 팀원 간의 피드백 실시
  - 새로운 주간 개발 목표 수립
- 비대면 회의
  - 디스코드 이용(녹음/녹화/채팅)
  - 디스코드를 이용해 자료 공유 및 의견 공유
  - 새로운 주간 개발 목표 수립

2.3. 프로토타입 및 소스코드

2.3.1. word cloud

그림 3 word cloud 출력화면

채팅 데이터 전체의 Word cloud 실행화면, 캡스톤 진행 채팅 내용을 적용한 화면이다.
해당 채팅 데이터의 형태소별로 나눠, Word cloud를 출력하였다.
데이터 정제 과정에서 더 많은 의미 없는 단어 삭제 설정이 진행 중이고, 참여자별 word cloud를 비교과정을 진행 중이다.

<소스코드>

```
#구글 드라이브 사용 구너한 설정
from google.colab import drive
drive.mount('/gdrive')
from google.colab import drive
drive.mount('/content/drive')

#파일 저장 위치 설정
import os
colab_path = "/content/drive/MyDrive/데이터분석"


chat = open(os.path.join(colab_path, "카카오톡 데이터 분석 파일.txt"), 'r',encoding="utf-8")
#추출을 위한 채팅 원본 파일 읽기 모드

allchat = open(os.path.join(colab_path, "allchat.txt"), 'w', encoding="utf-8")
shchat = open(os.path.join(colab_path, "shchat.txt"), 'w', encoding="utf-8")
jychat = open(os.path.join(colab_path, "jychat.txt"), 'w', encoding="utf-8")
sychat = open(os.path.join(colab_path, "sychat.txt"), 'w', encoding="utf-8")
#각 사용자별로 새 텍스트 파일을 생성

sh = "윤선희 :"
jy = "재윤 :"
sy = "한 세영 :"
# 해당 문자열이 들어있으면 추출해서 따로 저장

nonchating = ['사진','샵검색','이모티콘','동영상']
# 일반적인 채팅 내용이 아닌 것들은 예외처리

while True:
```

```
    line = chat.readline()
    #print(line)
    if not line:
        break

    isOk = True
    for item in nonchating:
        if item in line:
            isOk = False
            break

    if isOk:
        if sh in line:
            line = line.split(" : ")[1]
            allchat.write(line)
            shchat.write(line)
        elif jy in line:
            line = line.split(" : ")[1]
            allchat.write(line)
            jychat.write(line)
        elif sy in line:
            line = line.split(" : ")[1]
            allchat.write(line)
            sychat.write(line)

chat.close()
allchat.close()
shchat.close()
jychat.close()
sychat.close()

import operator

allchat = open(os.path.join(colab_path, "allchat.txt"), 'r',encoding="utf-8")
allchatfreq = open(os.path.join(colab_path, "allchatsfreq.txt"), 'w', encoding="utf-8")
shchat = open(os.path.join(colab_path, "shchat.txt"), 'r',encoding="utf-8")
shchatfreq = open(os.path.join(colab_path, "shchatsfreq.txt"), 'w', encoding="utf-8")
jychat = open(os.path.join(colab_path, "jychat.txt"), 'r',encoding="utf-8")
jychatfreq = open(os.path.join(colab_path, "jychatsfreq.txt"), 'w', encoding="utf-8")
sychat = open(os.path.join(colab_path, "sychat.txt"), 'r',encoding="utf-8")
sychatfreq = open(os.path.join(colab_path, "sychatsfreq.txt"), 'w', encoding="utf-8")

#단어와 빈도수를 저장할 freq파일을 생성
def get_freq(chat_file, out_file):
        #chat_file은 대화 내용이 담긴 텍스트 파일, out_file은 빈도를 저장할 file
    freq = {}
    # 단어와 빈도를 저장할 딕셔너리 생성

    while True:
        line = chat_file.readline()
        if not line:
            break
                #읽어오는 부분은 동일하다
        lines = line.split(" ")
```

```
                    #공백으로 단어를 구분한다
        for i in lines:
          #구분한 단어들을,
            i = i.replace('\n', '')
            #개행문자를 제거한 후
            count = freq.get(i, 0)
            #기존에 저장된 빈도 딕셔너리에서 가져온다
            freq[i] = count + 1

    frequency_list = sorted(freq.items(), key=operator.itemgetter(1), reverse=True)

    '''
    # sorted(dict.items(), key=operator.itemgetter(0))
    # key=operator.itemgetter(0)는 정렬하고자 하는 키 값을 0번째 인덱스로 한다..

    '''
    for words in frequency_list:
        out_file.write(words[0] + " : " + str(words[1]) + "\n")
        #정렬된 딕셔너리 {단어} : {빈도} \n 로 저장한다.

    chat_file.close()
    out_file.close()

get_freq(allchat, allchatfreq)
get_freq(shchat, shchatfreq)
get_freq(jychat, jychatfreq)
get_freq(sychat, sychatfreq)

import matplotlib.font_manager as fm
import matplotlib.pyplot as plt

path = '/usr/share/fonts/truetype/nanum/NanumGothicEco.ttf'
font_name = fm.FontProperties(fname = path, size=10).get_name()
print(font_name)
plt.rc('font', family = font_name)
fm._rebuild()

!pip install nltk
!pip install --upgrade pip
!pip install JPype-0.6.3-cp37-cp37m-win_amd64.whl
!pip install konlpy
!pip install wordcloud
!pip install --upgrade gensim

from konlpy.tag import Kkma
kkma = Kkma()
import nltk

from konlpy.corpus import kobill

doc_ko = open(os.path.join(colab_path, "allchat.txt"), 'r',encoding="utf-8").read()
doc_ko

from konlpy.tag import Okt
```

```
t=Okt()
tokens_ko = t.nouns(doc_ko)
tokens_ko

ko = nltk.Text(tokens_ko, name = '채팅내용')
print(len(ko.tokens))
print(len(ko.tokens))
ko.vocab()

!pip install matplotlib
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname = "/usr/share/fonts/truetype/nanum/NanumGothicEco.ttf").get_name()
rc('font', family = font_name)

from wordcloud import WordCloud, STOPWORDS

data = ko.vocab().most_common(150)
path = '/usr/share/fonts/truetype/nanum/NanumGothicEco.ttf'
wc = WordCloud(font_path = path, relative_scaling = 0.2, background_color = 'white').generate_from_frequencies(dict(data))

plt.figure(figsize=(12,8))
plt.imshow(wc)
plt.axis('off')
plt.show
```

2.3.2. 개인별 막대그래프

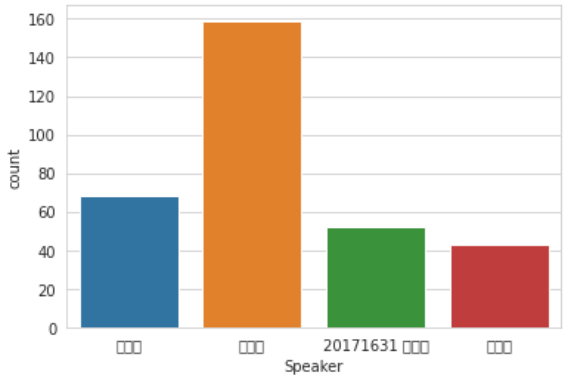

그림 4 참여자별 대화 빈도수 막대그래프

참여자의 대화 내용의 양을 분석하여 출력한 막대그래프다.
해당 채팅 데이터의 의미 없는 대화를 제거하여 참여자별 대화 양을 막대그래프를 출력하였다.
데이터 정제 과정에서 더 많은 의미 없는 단어 삭제 설정이 진행 중이고, 해당 단어 선택을 하여 단어별 대화 양을 출력을 진행 중이다.

<소스코드>

```
import re
import pandas as pd
import datetime as dt
```

```
import io

def read_kko_msg(filename):
    with open(filename, 'r', encoding='utf-8') as f:
        msg_list = f.readlines()
    return msg_list

def apply_kko_regex(msg_list):
    kko_pattern = re.compile("\[([\S\s]+)\] \[(오전|오후) ([0-9:\s]+)\] ([^\n]+)")
    kko_date_pattern = re.compile("--------------- ([0-9]+년 [0-9]+월 [0-9]+일) ")

    emoji_pattern = re.compile("["u"\U0001F600-\U0001F64F"  # emoticons
                               u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                               u"\U0001F680-\U0001F6FF"  # transport & map symbols
                               u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                               "]+", flags=re.UNICODE)

    kko_parse_result = list()
    cur_date = ""

    for msg in msg_list:
        # 날짜 부분인 경우
        if len(kko_date_pattern.findall(msg)) > 0:
            cur_date = dt.datetime.strptime(kko_date_pattern.findall(msg)[0], "%Y년 %m월 %d일")
            cur_date = cur_date.strftime("%Y-%m-%d")
        else:
            kko_pattern_result = kko_pattern.findall(msg)
            if len(kko_pattern_result) > 0:
                tokens = list(kko_pattern_result[0])
                # 이모지 데이터 삭제
                tokens[-1] = re.sub(emoji_pattern, "", tokens[-1])
                tokens.insert(0, cur_date)
                kko_parse_result.append(tokens)


    kko_parse_result = pd.DataFrame(kko_parse_result, columns=["Date", "Speaker", "timetype", "time", "contents"])
    kko_parse_result.to_csv("kko_regex.csv", index=False)

    return kko_parse_result

if __name__ == '__main__':
    #file_ex = open(os.path.join(colab_path, "카카오톡 데이터 분석 파일.txt"), 'r')
    msg_list = read_kko_msg("/gdrive/My Drive/캡스톤 디자인/데이터 분석 연습/카카오톡 데이터 분석 파일.txt")
    apply_kko_regex(msg_list)

df = pd.read_csv("kko_regex.csv")
df.head()

# 날짜로 변환
df.Date = pd.to_datetime(df.Date)

# 파생변수 추가
## 년월일,요일
df["year"] = df['Date'].dt.strftime('%Y')
```

```
df["month"] = df['Date'].dt.strftime('%m')
df["day"] = df['Date'].dt.strftime('%d')
df["weekday"] = df['Date'].dt.weekday

## 24시간제 표기
df["24time"] = df["timetype"] + " " + df["time"]
df["24time"] = df["24time"].map(lambda x : x.replace("오전","AM"))
df["24time"] = df["24time"].map(lambda x : x.replace("오후","PM"))

temp = []
transform_time = []
for i in range(len(df)) :
    time = df["24time"][i]
    temp.append(dt.datetime.strptime(time,"%p %I:%M"))
    transform_time.append(temp[i].time())

df["24time"] = transform_time

## 글자 수
title_len = []

for i in range(len(df)):
    ttl = len(str(df['contents'][i]))
    title_len.append(ttl)

df['length'] = title_len
df.head()

# 데이터 확인
df.head()

import matplotlib.font_manager as fm
import matplotlib.pyplot as plt

path = '/usr/share/fonts/truetype/nanum/NanumGothicEco.ttf'
font_name = fm.FontProperties(fname = path, size=10).get_name()

print(font_name)
plt.rc('font', family = font_name)
fm._rebuild()

# 그래프 모듈 설치
import seaborn as sns

# 그래프 삽입 모듈 및 그래프 내 한글 폰트 삽입 코드
import matplotlib
import matplotlib.pyplot as plt



sns.set_style('whitegrid')
sns.countplot(x= "Speaker",data=df)
plt.show()

# 데이터 전처리
```

```
def preprocessing(text):
    # 개행문자 제거
    text = re.sub('\\\\n', ' ', str(text))
    # 특수문자,자음 제거
    text = re.sub('[.,;:\)*?!~`^\-_+<>@\#$%&=#}※ㄱㄴㄷㄹㅁㅂㅅㅇㅈㅊㅋㅌㅍㅎㅠㅜ]', '', text)
    # 중복 생성 공백값
    text = re.sub(' +', ' ', text)
    return text

# 불용어 제거
def remove_stopwords(text):
  # 띄어쓰기 기준으로 단어삭제# 불용어 제거



    tokens = text.split(' ')
    stops = ['ㅎㅎㅎ','ㅎㅎ','ㅋㅋ','ㅋ','ㅎㅎ','ㅎ','ㅋㅋㅋ','ㅋㅋ','ㅠㅠㅠㅠ','ㅠㅠ',
             "그냥","거기","지금","이제","우리","일단","한번","나도","하는","그게","약간","그거","해서","재미","뭔가","이모티콘",
             "존나", "누가", "하기", "하는데", "거의", "할게", "이번", "이건", "사실", "정도", "갑자기", "혹시", "보고","하노",]
    meaningful_words = [w for w in tokens if not w in stops]
    return ' '.join(meaningful_words)

df['new'] = df['contents'].apply(preprocessing)
df['nnew']= df['new'].apply(remove_stopwords)

df.head()

# 한글 형태소분석기 모듈 설치
!pip install konlpy
from konlpy.tag import Okt

# okt패키지 사용
okt = Okt()
nouns = []
morphs = []
pos = []

for i in range(len(df)):
    no = okt.nouns(str(df['nnew'][i]))
    mo = okt.morphs(str(df['nnew'][i]))
    po = okt.pos(str(df['nnew'][i]))
    nouns.append(no)
    morphs.append(mo)
    pos.append(po)

# 파생변수 추가
df['nouns'] = nouns
df['morphs'] = morphs
df['pos'] = pos

## 동사 추출
verbs1 = df['pos']
dict_verb = []

for i in range(len(verbs1)):
```

```
    vb = dict(verbs1[i])
    dict_verb.append(vb)

df['dic_verb'] = dict_verb

## 동사 추출
verb_2 = []

for i in range(len(dict_verb)):
    verb_3 = [str(key) for (key, value) in dict_verb[i].items() if value == 'Verb']
    verb_2.append(verb_3)

df['verb'] =verb_2

## 문장 생성
df['sentence'] = df['nouns'] + df['verb']
df.head()

tokenized_doc = df['sentence']
detokenized_doc = []

for i in range(len(df)):
    t = ' '.join(tokenized_doc[i])
    detokenized_doc.append(t)
df['headline_text'] = detokenized_doc

df.head()

ndf = pd.DataFrame(df["nnew"])
ndf.columns=["clean data"]
ndf.reset_index(drop=True)

ndf.to_csv("ndf.csv", index=False)
ndf.head()
```

2.4. 테스트 계획

- 데이터 정제 테스트
  ① 형태소 정제 정확도 테스트
    - word cloud 출력을 통해 의미 없는 대화의 분포가 들어있는지 확인한다.
- 데이터 분석 테스트
  ① word cloud 분석을 통해 주제 키워드의 상관관계 조사
    - 참여자 별로 word cloud를 출력 후 분석해 주제와 관련된 단어와 관계가 있는지 판단한다.
  ② 반복적인 테스트를 통해 시간의 흐름에 따른 대화와 도출된 그래프 상관관계 조사
    - 시간의 흐름에 따라 누가 더 많이 채팅에 기여했는지를 알아보고 참여자별 막대그래프를 통해 오차가 없는지 조사한다.
  ③ 테스트를 통해 분석 오차율을 줄이는 것이 목표

2.5. 문제점 및 해결방법

- 문제점 : 프로젝트 일정계획
  - 중·고등학교 수업이 4월로 예정되어 여기서 채팅데이터를 수집할 예정이었는데, 코로나 19 상황이 악화되면서 수업취소와 일정 지연으로 인해 채팅 데이터 수집이 지연되었다.
- 해결 방법: 대학 교양 수업에서 사용했던 채팅 데이터를 수집 후 적용하였다.

2.6. 프로젝트 조직도

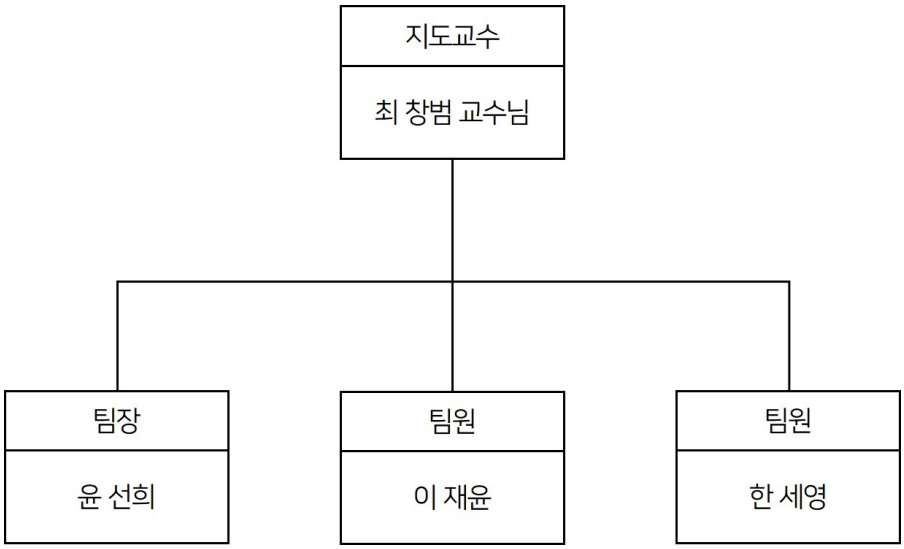

그림 5 팀 조직도

2.6.1 역할분담

구분	이름	세부역할
팀장	윤 선희	- 프로젝트 일정관리 - 데이터 시각화 코드 설계 (matplotlib 사용)
팀원	이 재윤	- 각종 문서 수집 및 작성 - 데이터 수집 코드 설계 - 데이터 정제 코드 설계 (python 사용)
팀원	한 세영	- 프로그램 주요 기능 설계 - 데이터 분석 코드 설계 (python 사용)

2.6.2 수행계획

- 1학기

주요 수행 내용	추진 일정(월 별)																	
	3월		4월		5월		6월		7월		8월		9월		10월		11월	
학습자료조사																		
소프트웨어 설계																		
데이터 수집																		
소프트웨어 개발 (코딩)																		
테스트 및 디버깅																		
수정 및 기능 개선																		
논문 작성																		

캡스톤 디자인 I 중간보고서 채점표

평가도구	평 가 항 목	평 가 점 수				
		1	2	3	4	5
중간 보고서 및 실행 결과	1. 요구사항 정의서(기능, 성능, 인터페이스 등)가 구체적으로 작성되었는가?					
	2. 요구분석, 설계 산출물(모델, 프로토타입 등)의 내용이 충실한가?					
	3. 설계 및 구현 문제를 위해 적용한 이론, 문제해결 방법이 제시되었으며 그 적용이 적합한가?					
	4. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)가 버그 없이 실행되었는가?					
	5. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)의 성능 요구사항은 충족되었는가?					
도구활용	6. 설계 및 구현을 위해 도구가 적절히 활용되었는가?					
	7. 도구의 활용수준(능숙도)은 프로젝트 수행에 적합한가?					
팀원의 업무 및 역할	8. 팀원의 업무분담에 따른 역할 및 협력이 충실히 이루어졌는가? (평가자에 의한 질의)					
	9. 프로젝트 중간 진척상황에 대해 팀원이 충분히 인지하고 있는가?(평가자에 의한 질의)					
합계						

***검토 의견(최종완료 때까지 보완해야할 점에 대해 작성 요망)**

심사위원(소속): **(이름)** **(인)**