번역채팅 졸업작품 보고서

정보통신공학전공 임동주

졸업작품 지도교수 [ ]

제1장 작품의 개요

인터넷이 발전함에 다른 나라와의 벽이 점점 허물어지고 있다. 하지만 언어라는 벽이 가로막고 있어서 소통에 어려움이 있었다. 요즘 기술이 좋아져서 웹에서 바로 번역을 해준다. 하지만 그건 일반적인 문서의 이야기다. 요즘에는 번역채팅 앱도 나오며 이미지 내 텍스트도 번역을 해주는 사례도 보이고 있다. 하지만 나는 기계에 종속 되어있는 앱이 아닌 어떤 기기에서도 인터넷만 된다면 동작이 가능한 웹을 기반으로 번역 채팅 앱을 만들어 보는 것이 좋겠다고 생각했다. 그리하여 이번 졸업과제 작품으로 웹을 기반으로 한 문자 번역 채팅. 더 나아가서 이미지 내 글자도 번역해 주는 “번역채팅”을 만들게 되었다.

제2장 작품의 원리 및 구성

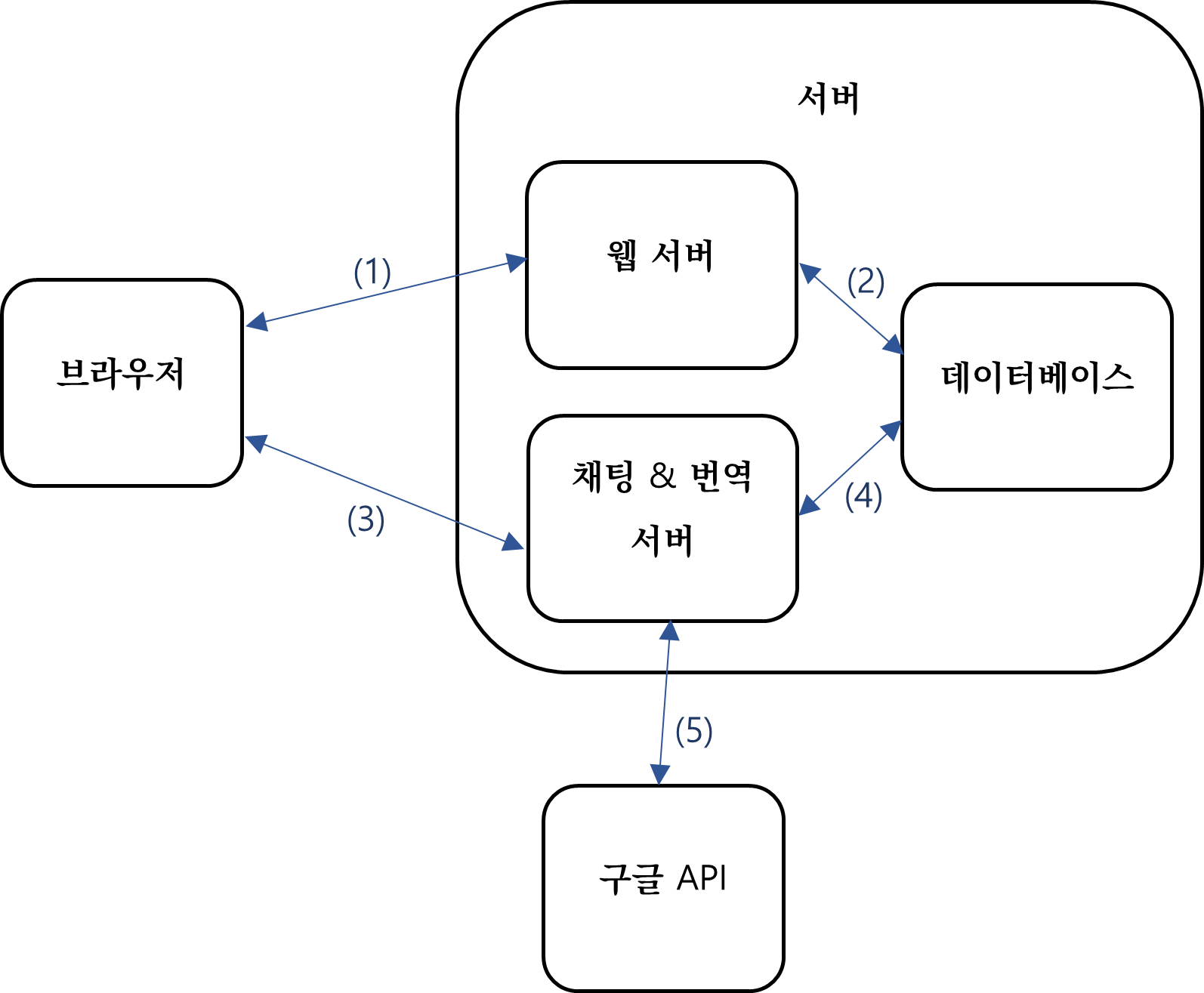
서버의 구성과 단어 정의는 다음과 같다.

서버: 젯슨 나노로 이루어져 있으며 웹서버와 채팅 및 번역서버가 탑재 되어있다.

웹서버: 아파치로 만들어져 있으며 채팅 웹의 기본 html문서를 만들어 보내주는 기능을 한다. 직접적인 채팅기능은 하지 않으며 채팅기능을 하는 자바 스크립트가 포함된 html파일을 만들어준다. 아파치는 자동으로 서비스에 등록되어 자동 실행됨.

채팅 및 번역서버: 파이썬으로 만들어져 있으며 브라우저의 자바 스크립트 웹 소켓과 직접 통신을 하며 채팅, 번역 기능을 직접 수행한다. 수동으로 서비스에 등록하여 자동 실행됨.

데이터베이스: 마리아 데이터베이스로 구축했으며 사용자 정보, 채팅방 정보, 채팅방 연결 정보, 채팅 데이터, 언어 코드 데이터 등 데이터를 저장한다.



<그림 1> 전체 흐름도

**(1). 브라우저 요청 & 웹 서버 응답**

- 회원가입: 아이디, 비밀번호1,2 입력. 아이디는 100자를 초과할 수 없으며 중복확인을 진행 해야함. 중복확인 완료 후 입력된 아이디를 변경하면 중복확인을 다시 진행 해야함.

- 로그인: 아이디, 비밀번호 입력. 아이디, 비밀번호 둘 중 하나라도 맞지 않다면 없는 정보라는 안내문 출력됨.

- 로그아웃: 회원 로그아웃. 관련 세션 모두 해제

- 채팅방 만들기: 채팅방 이름 입력, 비공개 여부 체크 가능. 체크하면 본인만 볼 수 있음.

- 프로필: 아이디, 설정언어 확인가능, 언어 변경창으로 이동 가능, 계정 삭제 가능. 계정 삭제의 경우 한 번 더 물어보는 절차를 거침.

- 번역 지원언어: 설정 가능한 언어 목록을 볼 수 있음. 회원은 설정된 언어는 초록 글씨로 보이며 다른 언어를 누르면 변경 가능. 비회원은 상호작용 없음.

- 채팅방 종류: 비회원은 보이지 않으며 누르면 선택창이 나와서 “채팅방”, “개인방”, “내가만든” 3개중 하나를 선택 가능. 변경을 누르면 채팅방의 경우 모든 “공개방”이 목록으로 나옴. “개인방”의 경우 본인이 채팅방 생성시 “비공개 여부” 체크한 방만 나옴. “내가만든”의 경우 본인이 생성한 모든 공개방, 개인방 목록이 보임. 모든 목록은 5개씩 끊어서 보여주며 5개를 초과하면 아래에 페이지를 만들어 줌

**(2). 웹서버와 데이터베이스 데이터 이동**

텍스트, 스크린샷, 도표, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

<그림 2> 데이터베이스 다이어그램

- 회원가입: 사용자가 입력한 정보를 데이터베이스에 저장함. 비밀번호는 암호화 진행 후 데이터베이스에 저장.스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

<그림 3> 비밀번호 암호화

* 로그인: 사용자가 입력한 정보를 토대로 일치하는 정보가 있는지 확인. 비밀번호 암호화 진행 후 비교하여 일치 여부를 보고 판단. 맞는 정보면 세션에 정보 저장

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

<그림 4> 비밀번호 비교

- 채팅방 만들기: 사용자가 입력한 정보로 데이터베이스에 채팅방 생성. 각 채팅방은 구별이 가능해야 해서 숫자와 알파벳 대문자로 이루어진 난수 문자열을 길이는 16으로 만들어서 고유 문자열로 부여한다. 만약 이미 있는 할당된 문자열이라면 다시 만든다. 하지만 확률 상 상당히 희박하여 새로 만드는 경우는 없다고 보면 된다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

<그림 5> 채팅방 데이터베이스 구조

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<그림 6> 랜덤 문자열 생성하는 함수

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

<그림 7> 고유 문자열 할당하는 코드

- 프로필: 세션 정보를 토대로 사용자의 아이디, 설정언어 가져옴

- 번역 지원언어: 해당 데이터가 있는 테이블의 모든 데이터를 가져옴

.

- 채팅방 종류: 3가지의 채팅방 옵션 중 해당 조건에 맞는 데이터를 설정된 페이지만큼만 가져온다. 해당 코드는 너무 길어서 부록 전체코드 참고.

**(3). 채팅 데이터 송수신**

- 최초 연결: "본인 아이디, 채팅방 아이디” 형식으로 웹 소켓을 통하여 한번 보냄.

* 연결 완료 시: 웹 소켓의 프로토콜(tcp)은 연결지향. 송수신 대기.

- 문자열 및 이미지 송신: 해당 데이터열에 데이터열 길이를 헤더로 넣고 1024바이트씩 나눠서 보냄. 그렇지 않으면 큰 사진의 경우 웹 소켓으로 넘어가지 않음.

텍스트, 스크린샷, 보라색, 바이올렛색이(가) 표시된 사진

자동 생성된 설명

<그림 8> 웹 소켓 서버로 데이터 송신

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

<그림 9> json 형식

- 문자열 및 이미지 수신: 서버에서 번역 데이터가 json형식으로 넘어옴. 필요한 데이터 뽑아서 사용. 자료형이 문자열의 경우 보낸 사람의 아이디와 시간을 제목으로 하고 문자열을 아래에 작성하여 <div>태그에 넣어 html문서에 추가한다. 자료형이 이미지인 경우 보낸 사람의 아이디와 시간을 제목으로 하고 바로 아래에 사진을, 사진 아래에 번역한 문자열을 결합하여 문자열과 같이 문서에 추가한다. 보낸 사용자가 나라면 배경색을 초록색으로 하고 그 외 데이터는 파란색으로 한다.

텍스트, 스크린샷, 멀티미디어, 폰트이(가) 표시된 사진

자동 생성된 설명

<그림 10> 문자열 수신

텍스트, 스크린샷, 멀티미디어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

<그림 11> 이미지 수신

**(4)(5). 채팅기능**

- 최초 연결: 클라이언트로부터 사용자 아이디, 채팅방 아이디를 얻음. 사용자 아이디와 연결 소켓의 포트를 이용하여 또 다른 고유 아이디를 만듦. 이유는 같은 사용자가 다른 디바이스 및 새창으로 연결했을 경우 각자 다른 연결이 되기 때문에 각 사용자에 대한 식별이 아닌 연결에 대한 식별이 필요하기 때문. 초기화가 이뤄졌다면 접속한 채팅방의 기존 데이터를 사용자에게 맞는 언어로 번역하여 보내주고 누락된 메시지에 대해서는 번역을 진행하여 보내준다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<그림 12> 최초 연결시 사용자 정보 및 객체 저장

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

<그림 13> 사용자 연결 식별 아이디 생성자

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

<그림 14> 파이썬 클래스

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<그림 15> 채팅방 클래스 세부구조

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<그림 16> 기존 메시지 번역 및 송신

- 데이터 송수신: 사용자가 데이터를 송신하면 문자열인 경우 데이터베이스에 저장 후 해당 사용자와 같은 채팅방에 있는 모든 연결에 각 언어 설정에 맞는 언어로 번역 후 송신. 이미지의 경우 이미지에서 문자열을 추출한 후 그 위치에 네모 상자를 그리고 사진은 디렉토리에 저장. 문자열과 이미지의 위치를 데이터베이스에 저장하고 다음은 문자열과 같이 같은 채팅방 사용자들에게 번역하여 송신. 여기서 이뤄지는 번역은 받거나 추출한 문자열을 경로(5)에서 구글(API)을 통해 이뤄진다. 스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

<그림 17> 채팅방 데이터 전체 번역 송신 병렬처리

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

<그림 18> 채팅방 데이터 각 번역 및 송신처리

제3장 결론 및 작품제작 후기

이번 프로젝트를 진행하며 다시 한번 크게 느낀 바로는 내가 원하는 기능이 이미 존재하는 라이브러리가 상당히 많으며 라이브러리가 없다면 즉, 누군가가 만들어 둔 것을 가져다 사용하지 못했다면 나는 이렇게 큰 프로젝트를 진행할 수 없었을 것이라는 점이다. 역시 기술은 다양한 사람들의 집단 지성으로 만들어져 가는 것이다. 원하는 기능이 조금만 복잡하면 그리고 만들어 둔 것을 공유하지 못한다면 각자 스스로 똑같은 기능을 만들어내며 재창조만 반복하며 발전하지 못했을 것이다. 솔직히 이번 졸업작품을 혼자 진행하며 버거운 점도 많았다. 졸업작품 제작을 진행할 때 실행해야 했던 프로그램만 하더라도 상당히 다양했다. 번역 기능 및 채팅 기능을 만드는 것도 쉽지 않았는데 웹페이지까지 만드니 더욱 신경 쓸 것이 많았다. 그렇게 느꼈기 때문에 라이브러리가 더욱더 소중하게 느껴졌다. 그리고 이것저것 만들고 싶은 기능이 많이 있었는데 아쉽게도 여기에 구현하지 못한 부분이 많이 있다. 욕심은 많은데 모두 구현하기에는 부족한 부분이 있어서 아쉬움도 많이 남는다. 처음부터 계획을 확실히 잡고 천천히 쌓아가야 했는데 급한 마음에 난잡하게 진행하다 보니 특히 웹의 코드가 유지, 보수 측면에서 너무나도 아쉬운 결과가 나왔다. 최소 한달정도는 계획을 세우며 확실히 기반을 만드는 기간으로 정하는 것이 좋았을 것 같다.

부록

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

<그림 19> 채팅방 클래스 구조1

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

<그림 20> 채팅방 클래스 구조2

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<그림 21> 사용자 클래스 구조

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

<그림 22> 메시지 번역

참고문헌

[1] 인프런, <https://www.inflearn.com/questions/39965/call-to-undefined-function-mysqli-connect-%EC%97%90%EB%9F%AC>

[2] 파이썬 도큐먼트, <https://docs.python.org/ko/3.9/howto/logging.html>

[3] MakingRobot, <https://makingrobot.tistory.com/74>

[4] jehyunlee, <https://jehyunlee.github.io/2023/02/20/Python-DS-128-transqual/>

[5] pypi googletrans, <https://pypi.org/project/googletrans/3.1.0a0/>

[6] modulesGoogletrans 3.0.0 documentation, <https://py-googletrans.readthedocs.io/en/latest/>

[7] mokeya.tistory, <https://mokeya.tistory.com/146>

[8] tesseract-ocr github, <https://github.com/tesseract-ocr/tessdata/>

[9] PHP Manual, <https://www.php.net/manual/en/book.mysql.php>

[10] wildmental.log, <https://velog.io/@wildmental/%EA%B7%B8%EB%A6%AC%EA%B8%B0-%EA%B7%80%EC%B0%AE%EC%9D%80-ERD-DBdiagram.io%EB%A1%9C-5%EB%B6%84%EB%A7%8C%EC%97%90-%EC%99%84%EC%84%B1%ED%95%98%EA%B8%B0>

[11] DB diagram, <https://dbdiagram.io/home>

[12] Air on the C String, <https://velog.io/@d2h10s/pythton-%EC%9E%90%EB%8F%99%EC%9C%BC%EB%A1%9C-UML-Class-Diagram-%EB%A7%8C%EB%93%A4%EA%B8%B0>

[13] Graphviz, <https://graphviz.org/>