프로젝트 레포트

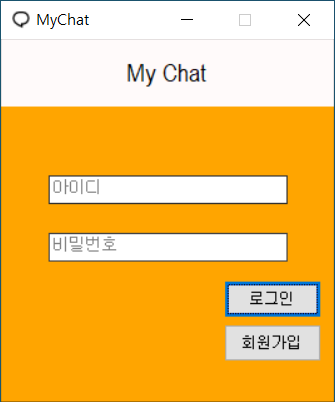
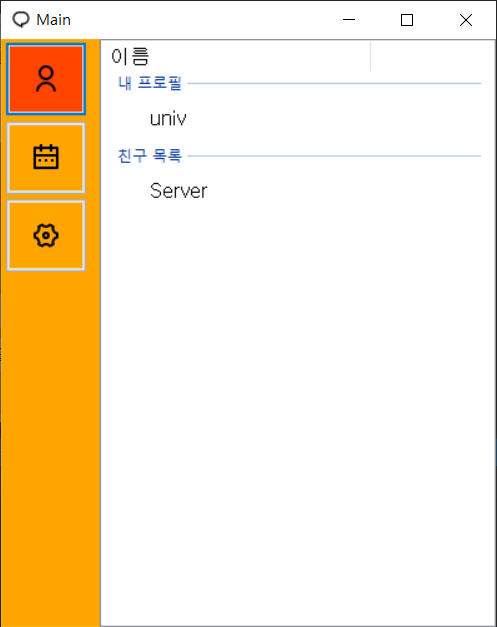
|  |  |
| --- | --- |
| 과목 | 응용소프트웨어실습 |
| 과제 | 프로젝트 개인 레포트 |
| 제출일 | 2020-06-28 |
| 소속` | 컴퓨터소프트웨어학부 |
| 학번 | 2015726012 |
| 이름 | 최세일 |



**목차**

1. **프로젝트 개요**
2. **프로젝트 목표**
3. **구현 방법 및 한계점**
4. **고찰**
5. 프로젝트 개요

아이디를 키로하는 멀티유저 채팅 프로그램

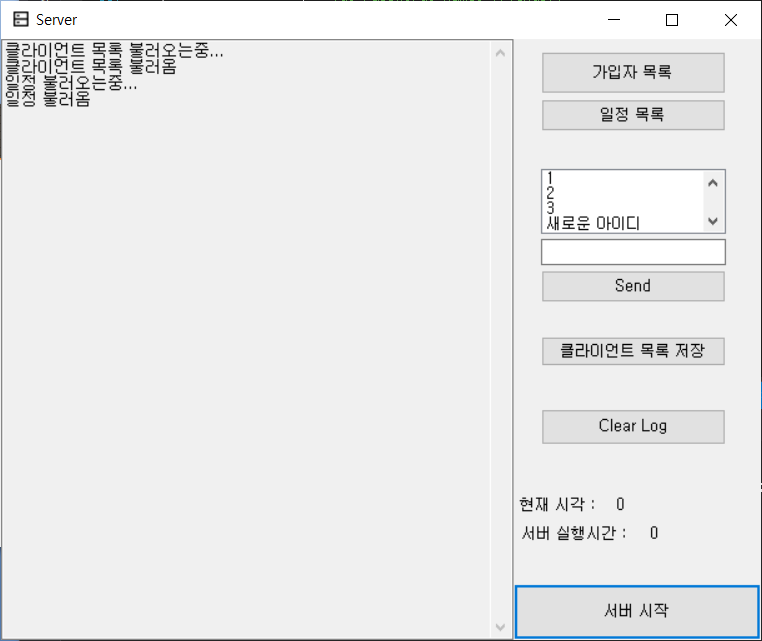
 

* 1. 회원가입을 통한 사용자 등록
  2. 기본적으로 상대의 아이디를 알면 채팅 가능
  3. 일정등록 & 알림 기능
  4. 서버, 클라이언트 기능

|  |  |
| --- | --- |
| **서버** | **클라이언트** |
| * 사용자 등록 & 정보 저장 * 로그인 처리 * 사용자간 통신 제어 * 알림 기능 | * 회원가입 * 아이디를 이용한 채팅 * 일정 등록 & 삭제 |

1. **프로젝트 목표**
   1. 사용자간 채팅뿐 아니라 간단한 캘린더 기능 및 일정 등록기능을 추가한 채팅 플렛폼을 만들고자 하였다.
   2. 데이터 베이스를 연동한 회원등록을 하려 했으나 구현상의 문제로 로컬에 회원정보를 등록하게 됨
   3. 구현시간의 한계로 인해 다수의 기능을 구현하지 못하고 현재의 프로젝트를 만들게 됨
2. **구현 방법 및 한계점**

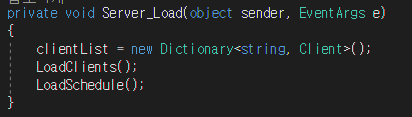
**3-1. 서버**

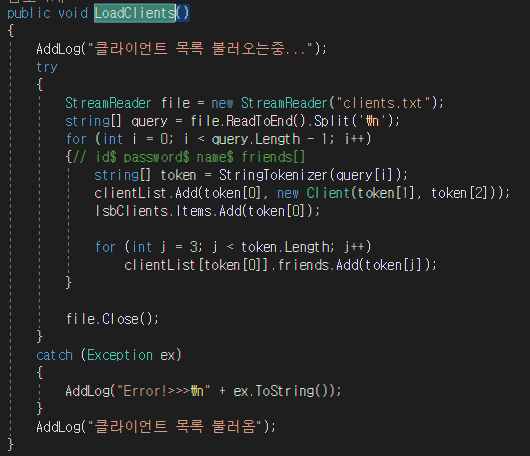


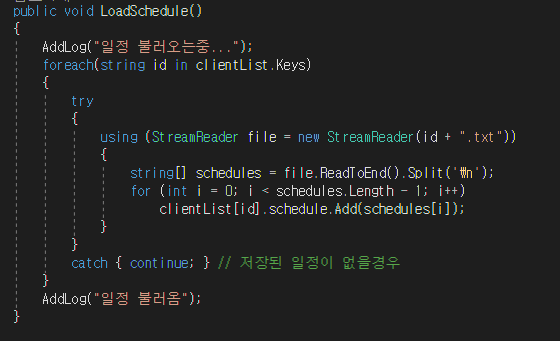
◀ 현재 등록된 사용자

◀ 서버에 현재 저장된 가입자 목록, 가입자별 등록된 목록 보여줌

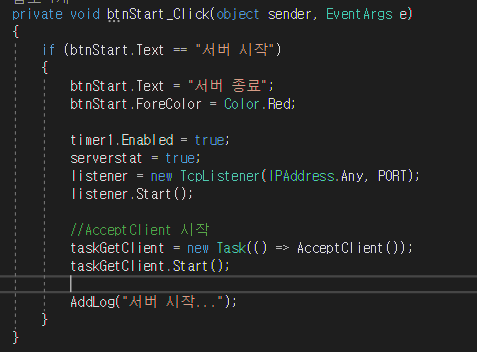
우선 시작시 저장된 사용자 목록을 가져온다.



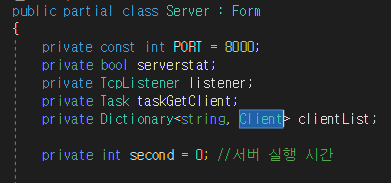
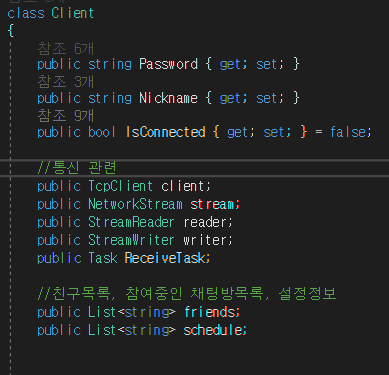




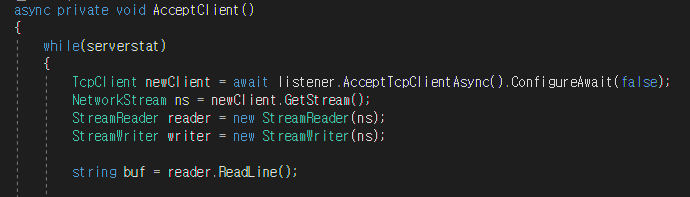
* Streamreader 를 이용하여 로컬에 저장된 파일을 불러온다. 이후에 등록된 사용자들은 회원정보가 저장되고 각자의 일정이 등록된 파일을 만들게 된다.

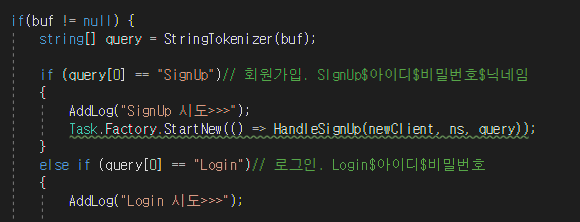


* 서버 시작 버튼을 눌러 클라이언트의 통신을 대기하게 된다.

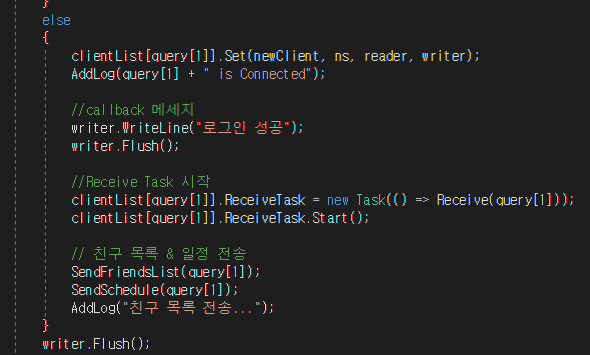
 

* 서버의 정보를 저장하기 위해 생성한 클래스와 서버폼의 변수 부분

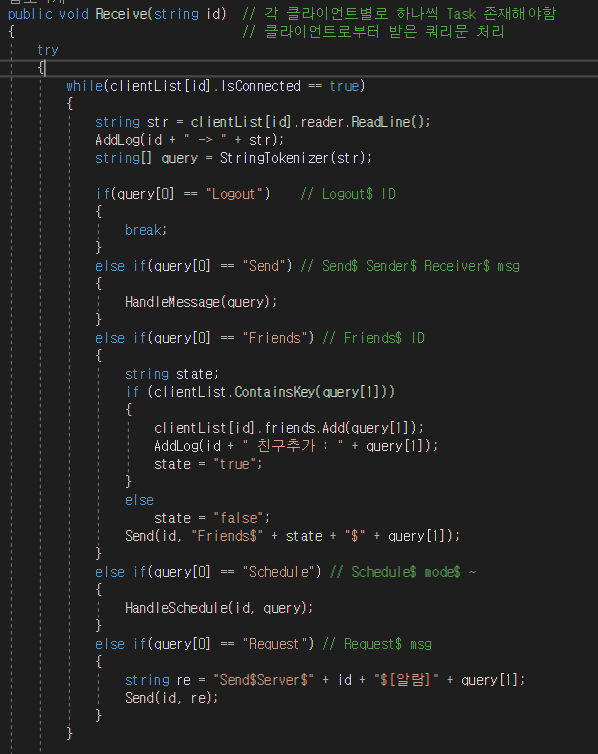




* 클라이언트와의 통신을 대기하게 된다.
* 클라이언트와 통신을 할 때 다양한 기능을 수행하기 위해 미리 약속된 문자열을 보내어 처리하도록 하였다.
* 예를 들면 로그인은 Login& 으로 시작하는 문자열을 보내어 처리하도록 하였고 이후에 Schedule&, 친구목록을 수정하는 Friends&, 채팅을 보낼 때는 Send&로 시작하는 문자열을 클라이언트와 서버가 주고받아 처리하도록 하였다.
* 이 통신방식은 사용자가 채팅할 때 & 문자가 포함된 문자열을 전송할 때 제대로 된 통신을 할 수 없다는 한계점이 있다.



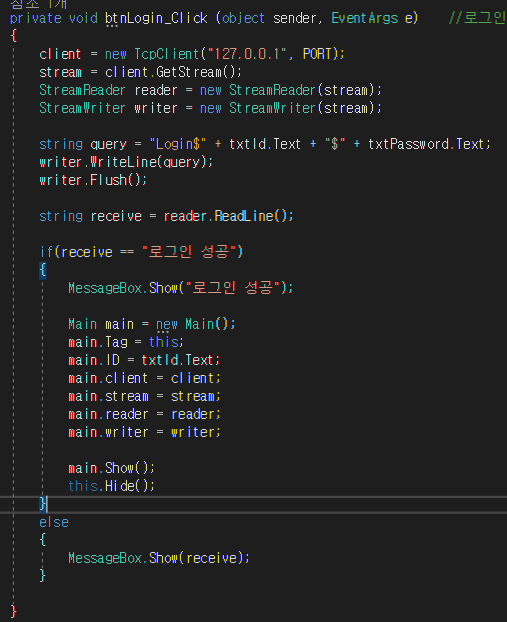
* 로그인에 성공하게 되면 클라이언트에게 저장된 친구목록, 일정목록을 보내주고 새로운 통신을 대기하는 Task를 만들게 된다.

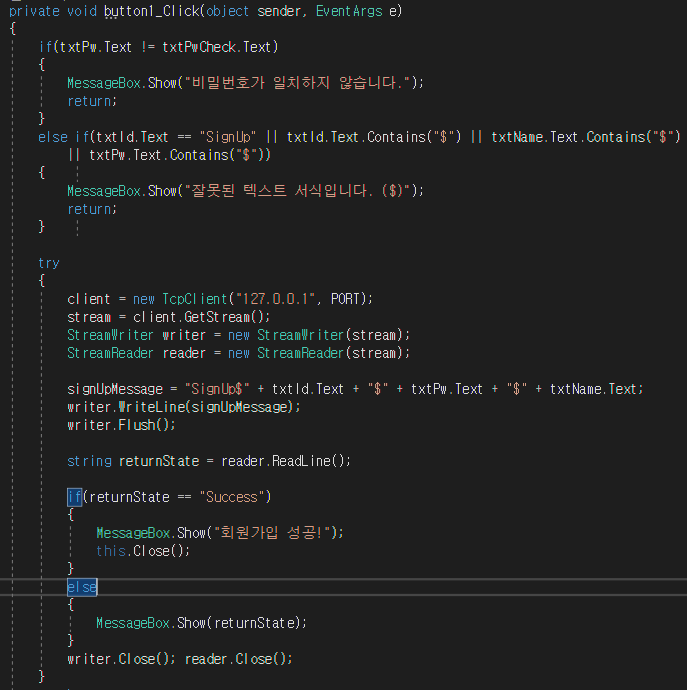


* 클라이언트는 서버에게 Logout, Send, Friends, Schedule, Request 로 시작하는 문자열을 보내기능을 수행하게 된다. 서버는 이를 바탕으로 회원정보를 수정하거나 사용자끼리의 통신을 수행한다.

**3-2. 클라이언트**

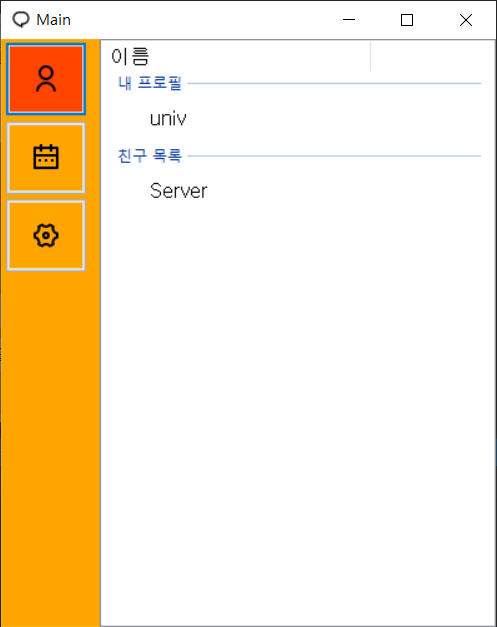
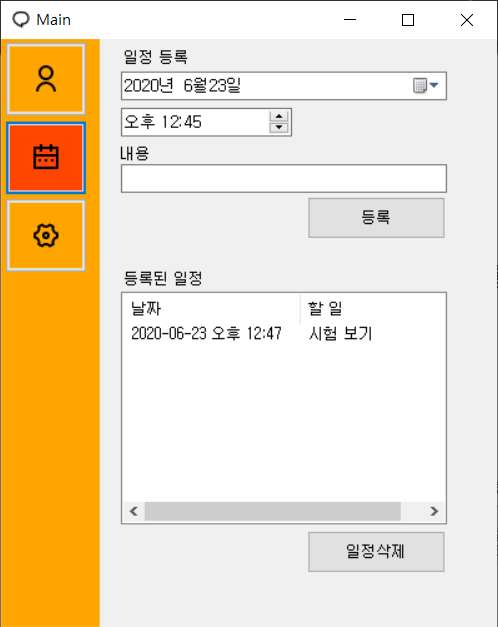
1. **로그인 화면 & 회원가입**

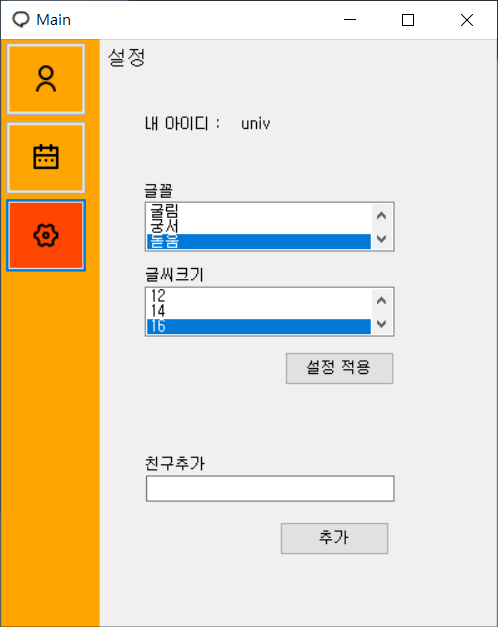
****



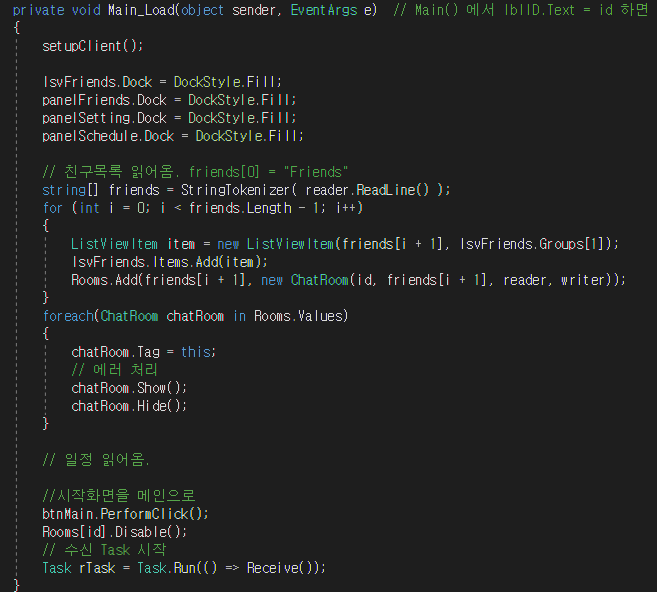
* 왼쪽의 회원가입 화면에서 화면에 정보를 입력한 후 서버의 등록된 회원정보와 대조하여 중복되지 않는 아이디라면 회원가입 성공하도록 하였다. 회원가입 완료시 서버와의 통신은 종료한다.
* 오른쪽의 로그인 화면에서는 입력한 정보를 서버에 보내어 등록된 아이디면 로그인 하도록 하였다.

1. **메인 화면**

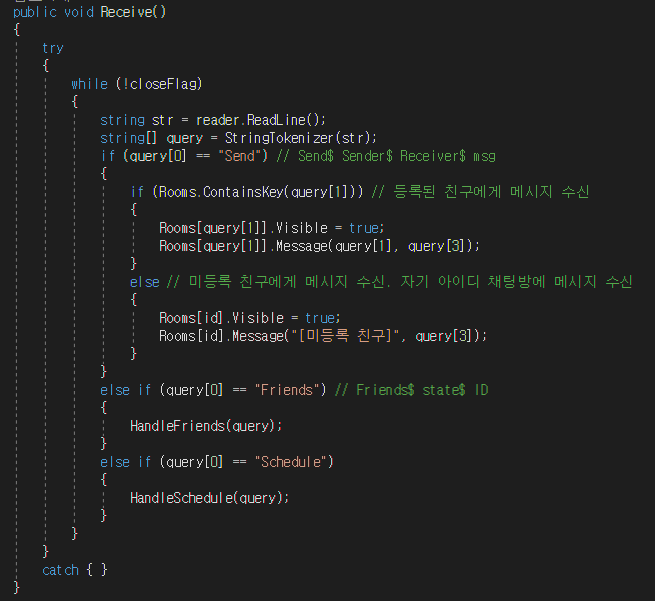
** **

****

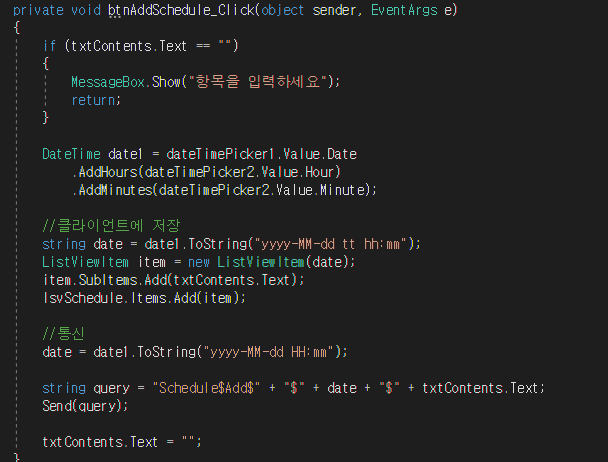
* 로그인시 볼 수 있는 메인 화면. 왼쪽의 버튼을 클릭시 패널이 전환되는 방식이다.

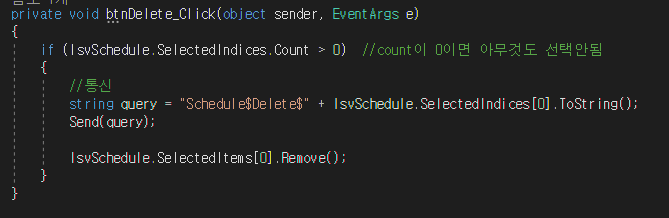


* 메인화면 로드시 실행부분. 우선 불러온 친구목록을 바탕으로 ListViewItem 을 추가하고 ChatRoom폼을 생성하고 로드 하도록 하였다. ChatRoom폼은 우선 Show 한 후에 다시 Hide 하도록 했는데 서버로부터 메시지를 받을 시 새로운 메시지 창을 바로 띄우도록 하기 위해 이런식으로 구현하였다.

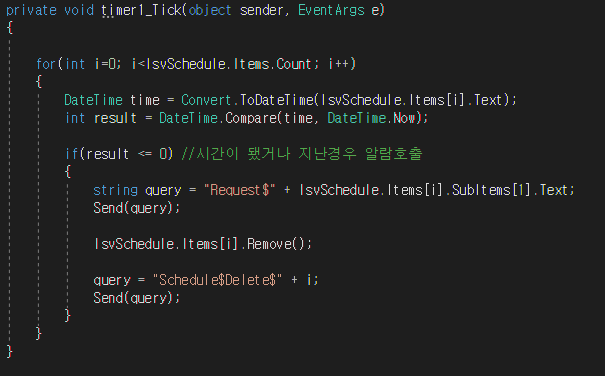


* 서버로부터 받은 메시지 처리. 메인화면 실행 시 루프를 돌면서 실행되고 Send, Friends, Schedule 기능을 서버에게서 받을 때 그에 맞게 처리한다.





* 일정 등록 및 삭제 버튼 구현. 날짜와 시간, 내용을 입력하고 등록을 하게되면 일단 클라이언트에 그 내용을 저장하고 서버로 보내도록 하였다. 서버는 클라이언트의 일정등록 통신을 받고 저장하게 된다.
* 일정을 삭제했을 경우 바로 리스트를 삭제하고 서버에 삭제소식을 알리도록 하였다.



* 클라이언트에서 등록된 일정의 시각이 되었는지 확인하도록 시간을 체크하게 하였다.
* 정해진 시각이 되거나 시간이 지난 후에 프로그램을 실행했을 경우 서버에 Request 로 시작하는 문자열을 보내 등록된 일정 내용을 서버가 보낸 채팅으로 받게 하였다.
* 후에 이 기능을 활용해 챗봇 기능을 구현할 생각이였지만 시간이 모자라 구현하지 못하였다.

**4. 고찰**

* 우선 이번 프로젝트를 진행함에 있어서 시간관리 및 계산을 제대로 수행하지 못하였다. 그러나 일정을 진행하면서 기능을 구현하는데 얼마간의 시간이 걸리게 될지 가늠할 수 있게 되었다.
* 일단 이번 프로젝트의 구현을 함에 있어서 새로운 기능을 적용하지 못하였다. 회원가입을 구현할 때 데이터베이스를 연동하여 회원정보를 저장하려고 처음에 계획하였으나 구현의 미숙함 및 시간이 모자라서 구현하지 못하였다.
* 프로젝트의 방향성 또한 마무리 짓고 나니 상당히 아쉽게 느껴졌다. 처음에 계획할 당시 챗봇과 일정등록 및 수정, 회원가입의 구현 난이도를 생각보다 낮게 느껴서 목표로 잡았으나 생각보다 난이도 있는 부분이고 시간이 오래 걸렸다. 조금 더 괜찮은 방향성, 1인 프로젝트에 알맞은 프로젝트를 고찰해 볼 수 있는 계기가 된 것 같다.