



사운드클라우드 미니 프로젝트

🕒 생성일	@2021년 8월 24일 오후 3:17
☰ 태그	프로젝트

사운드 클라우드 v1.0

소프트웨어 아키텍처

- 서버 - 클라이언트

핵심 기능

- 로그인 확인
→ (파일에 Id/pw 적어놓고 일치하면 로그인 성공만 출력할 예정)
- 선택가능한 메뉴 출력/선택
→ (음악목록, 음악 재생, 감상평 작성란)
- 서버쪽에 음원파일을 담아두고, 해당 파일 리스트를 뿌려주는 기능
- 클라이언트 측에서 음원 재생 기능
→ 몇초간 재생할지, 일시정지, 정지 ? 기타 등등 더 논의 필요
- 클라이언트가 음악 다운로드
- 사용자 로그 남기기(타임 스탬프)

역할 분담

- 네트워크 공통 부분
 1. 초기화 (IP 및 PORT 정보 불러오도록 해야함)

2. 다중 스레드환경은 힘들것으로 보여서 단일 스레드로 구현하는게 좋을듯
3. 입출력 스트림 설정
4. 로그인 기능 (login();)
 - a. 클라이언트 입장시 로그인 확인 여부 출력
 - b. 서버에 등록되었는 파일에 담긴 id와 pw가 클라이언트가 입력한 id/pw와 일치 하는지 확인
 - c. 일치하면 로그인 성공 출력하여 클라이언트로 출력

- 서버 부분

1. 로그인 기능 (login();)
 - a. 클라이언트 입장시 로그인 확인 여부 출력
 - b. 서버에 등록되었는 파일에 담긴 id와 pw가 클라이언트가 입력한 id/pw와 일치 하는지 확인
 - c. 일치하면 로그인 성공 출력하여 클라이언트로 출력
2. 선택가능한 메뉴 출력
3. 클라이언트가 선택한 메뉴의 기능 구현하기 (서버/클라이언트 어느쪽에서 해야되는 지?)
 - a. 음원 리스트 조회
 - b. 음원 재생
 - 재생
 - 정지
 - 일시정지
 - 또는, 처음부터 몇초만 재생하도록 설정
 - c. 음원 삭제
 - d. 리뷰 남기기
 - e. 다운로드 수 조회
4. 클라이언트가 음악을 선택하면 해당 음악을 클라이언트에 전송해야함
5. 타임 스탬프 구현하여 이벤트(서버 오픈, 사용자 접속 등) 발생시 기록 남겨주는 로 그 기능

- 클라이언트 부분

1. 로그인 기능

- a. 서버 입장시 로그인 정보 입력

2. 서버와 동일하게 선택 가능한 메뉴를 구현하거나, 서버에서 구현한 메뉴를 받도록 구현

3. 서버에서 출력된 음악 리스트중에 음악 선택이 가능하도록 해야함

4. 다운 받은 데이터를 어디로 받을지 경로 설정 해줘야함

5. 서버와 마찬가지로 로그 기능

6. 음악 재생하는 기능(다운받은것을 재생하는 방식으로 ?)

7. 음악 멈추는 기능

진행 방향

- 기존에 작성했던 채팅 프로그램을 바탕으로, 해당 소스파일에서 기능 구현 해보기

기타 정해야하는 항목 (추후 논의 후 정할 예정)

- 구현할 기능의 메소드 이름정하기
- 프로젝트 및 폴더 이름 정하기
- 어떤 자료구조를 사용할지?
- 각 메소드별 주요 사용될 함수, 자료구조 정하기

각 기능별 구현 내용

▼ 서버 - 클라이언트 부분

서버측 - ServerSocket / 클라이언트 - Socket

1. 서버소켓에는 포트번호를 매개변수로 넣어줌

- a. 서버에는 서버소켓, 클라이언트 소켓 두개를 만들어 주어야함

2. 클라이언트 소켓부분에는 서버 IP와 포트번호를 매개변수로 넣어줌

3. 서버와 클라이언트를 서버측에서 클라이언트소켓.accept를 해줘서 연결해줌

4. 각 스트림을 소켓에 연결해줘야함

- a. 데이터 타입이 다양한 경우 DataIn/OutputStream 사용하기
- b. 버퍼드리더/라이터 쓰면 성능 향상을 기대할 수 있음
- c. 원래 read로 데이터를 다 읽으면 -1을 반환하지만, 어떤 이유에선지 우리쪽 메소드는 다른 숫자를 반환함(in/out풋 스트림의 경우 255를 반환했음)

▼ 로그인 부분

▼ 디렉토리 내 파일 리스트 출력 부분

File 클래스의 new File() 생성자의 매개변수로 파일 경로를 기입하여 해당

▼ 음악 재생 기능

▼ 음악 다운로드 기능

▼ 댓글 기능

구현하면서 느낀점

▼ 느낀점

- 서버와 클라이언트의 통신을 위해 매번 같은 위치나 순서로 데이터 송/수신을 해줘야 했는데, 이부분이 너무 힘들었던 것으로 기억함.
- 뭔가 한부분이라도 어긋나면 교착상태에 빠져 프로그램이 진행 되지 않았고, 해당 오류를 잡기위해 많은 시간을 소모하였음.