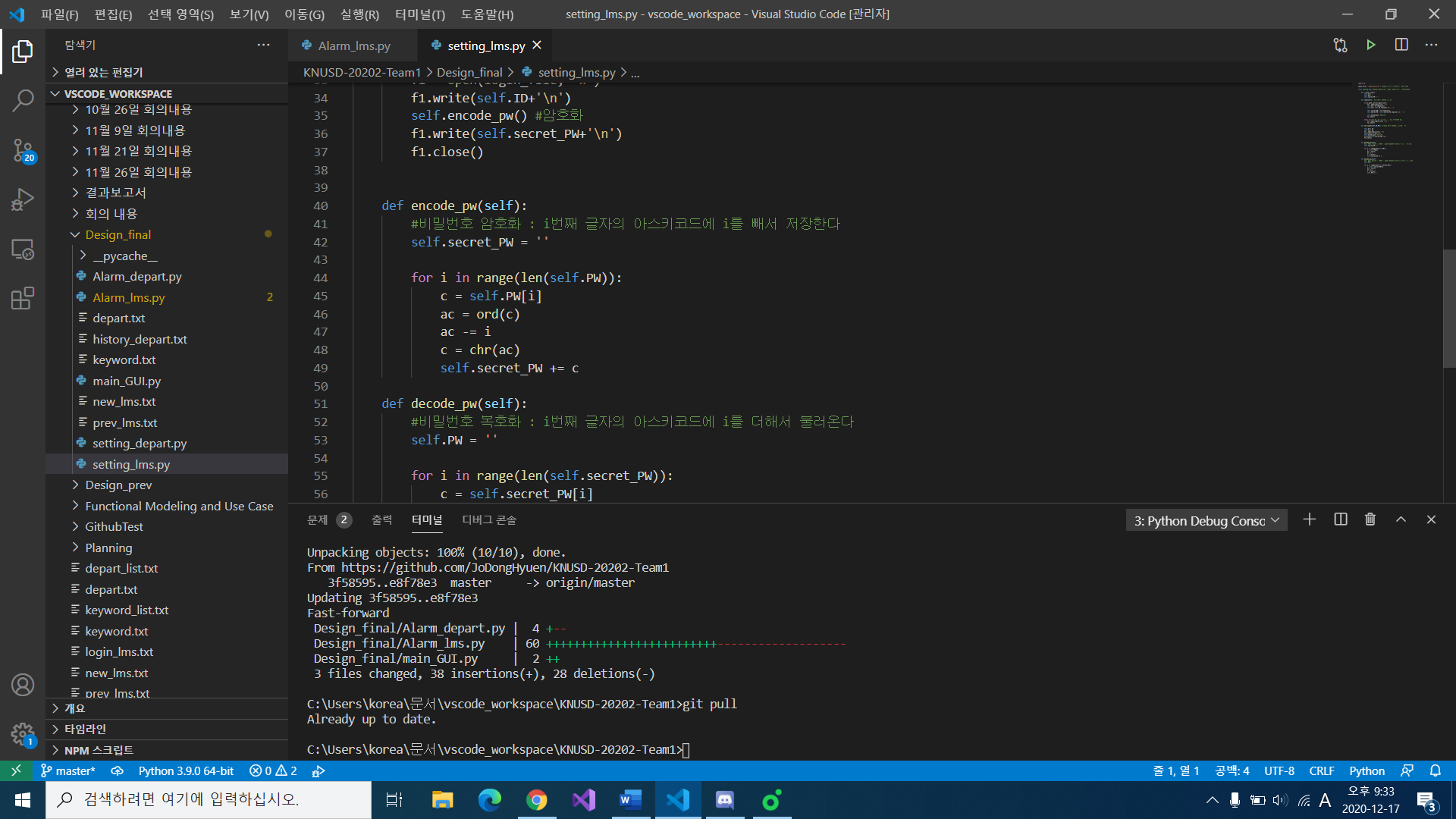
|  |  |  |  |
| --- | --- | --- | --- |
| **소프트웨어설계 프로젝트 결과보고서** | | | |
|  | | | |
| **프로젝트명:** | |  | |
|  | **조** | **팀원:** |  |

1. **프로젝트 개요** (프로젝트 설명, 목적 등)
2. **프로젝트 관리** (프로젝트 일정, 팀원별 역할 포함)
3. **시스템 기능 및 동작** (유스케이스 다이어그램, BPM 액티비티 다이어그램)
4. **시스템 아키텍처** (사용된 HW/SW 툴, 시스템 모델 등)
5. **소프트웨어 상세 설계**

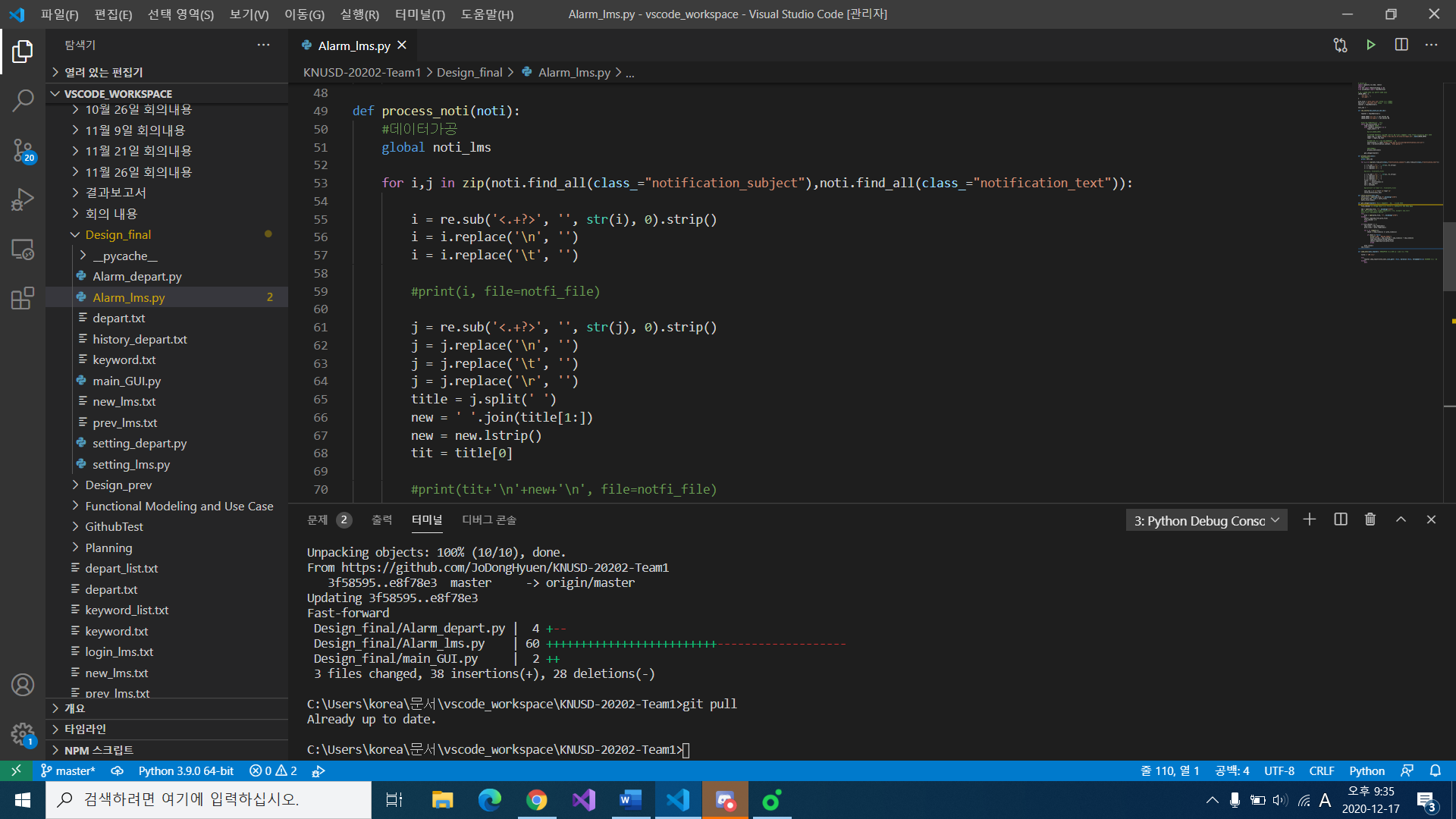
**5.1 정적 모델** (클래스 다이어그램, 컴포넌트 다이어그램 등 구조 표현)

**5.2 GUI 설계**

**5.3 데이터 설계**



login\_lms.txt 파일은 자동로그인을 위한 txt파일로 프로그램을 종료했다가 다시 켤 때 login\_lms.txt 파일에 저장되어 있는 아이디와 비밀번호를 자동으로 연결해주어 자동로그인을 되게끔 하는 파일입니다. 작동원리는 setting\_lms.py 에서 load 메소드로 먼저 login\_lms.txt 파일을 불러옵니다. 이 때 기존 login\_lms.txt 파일이 없을 시 새로 만듭니다. 그리고 로그인을 할 시 login\_lms.txt 파일에 아이디와 비밀번호를 저장합니다. 이 때 비밀번호는 encode\_pw 메소드로 암호화합니다. 암호화 원리는 위의 사진의 코드처럼 비밀번호 각 자리 문자에 그 문자의 인덱스 값만큼 아스키코드에서 빼는 원리입니다. login\_lms.txt 파일이 이미 존재할 시엔 프로그램을 켤 때 load 메소드로 아이디와 암호화된 비밀번호를 불러와서 decode\_pw 메소드로 먼저 비밀번호를 복호화 한 다음 자동으로 로그인을 되게끔 합니다.



new\_noti.txt 파일과 prev\_noti.txt 파일은 통합 알림 프로그램이 계속 실행되면서 알림 창의 내용을 계속 크롤링하여 가져오는데 가져온 알림이 기존의 알림 이랑 비교하면서 새로운 알림 인지 아닌지를 판단할 때 쓰이는 파일입니다. 작동원리는 먼저 Alarm\_lms.py에서 lms\_notify 메소드에서 new\_noti.txt 파일을 생성합니다. 그리고 lms 홈페이지의 알림 창의 내용을 크롤링하고 위의 사진처럼 process\_noti 메소드를 통해 데이터 가공을 해서 사용자의 알림을 10개까지 new\_noti.txt 파일에 저장을 합니다. 그리고 get\_change 메소드를 통해 new\_noti.txt 파일과 prev\_noti.txt 파일을 비교를 시작합니다. 이 때 프로그램이 처음 실행됐을 시 prev\_noti.txt. 파일은 존재하지 않기에 try except를 사용하여 new\_noti.txt 파일의 내용을 prev\_noti.txt shutil.copy를 사용하여 복사합니다. get\_change 메소드로 변경사항이 있을 시 사용자의 윈도우에 알림을 보내고 new\_noti.txt를 prev\_noti.txt로 shutil.copy로 복사합니다. 다음으로 프로그램이 계속 실행되면서 업데이트되는 new\_noti.txt와 prev\_noti.txt를 계속 비교하면서 새로운 알림 인지 아닌지를 판단하는 원리입니다.

1. **소프트웨어 주요 기능** (동적 모델, 시퀀스 다이어그램 등으로 동작 표현)

**6.1 주요기능명**

**6.2 주요기능명**

**...**

1. **테스트 계획 및 결과** (유스케이스별 테스트케이스 및 실제 테스트결과 등)
2. **프로젝트 이슈사항** (프로젝트 수행 시 문제점, 해결방법 및 느낀 점)
3. **실행 결과** (실행 결과 화면 캡처, 시연 동영상 url)
4. **GitHub 업로드 결과** (GitHub url, 업로드 현황, 팀원별 활동 현황 (Insights) 캡처 화면)
5. **참고자료** (프로젝트 수행 시 참고한 책 또는 사이트)