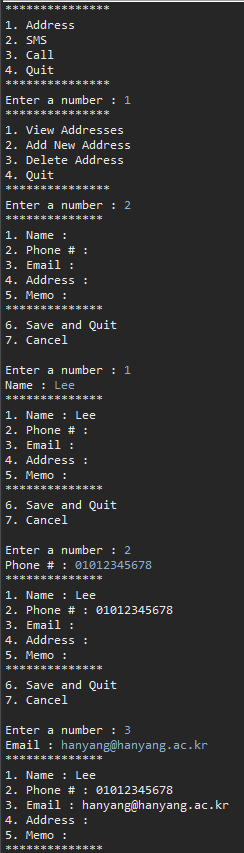
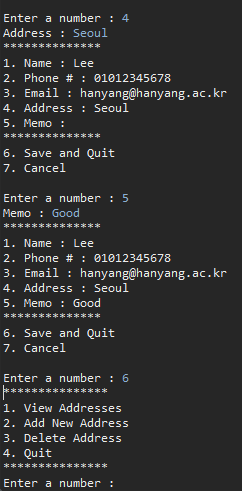
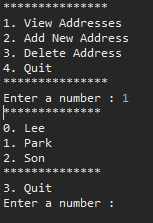
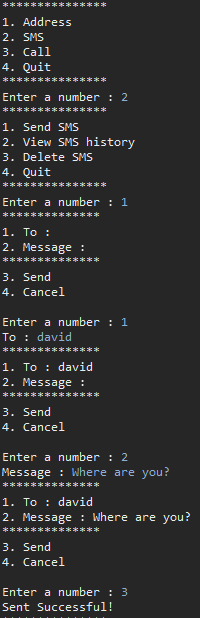
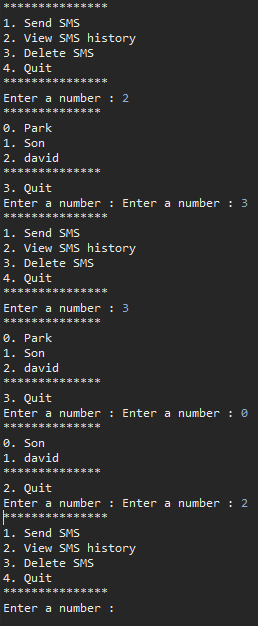
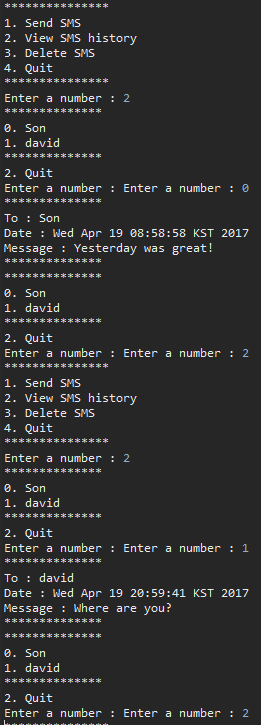
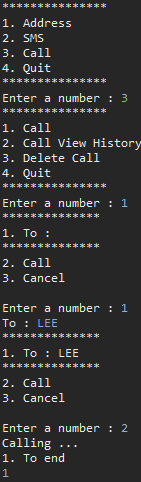
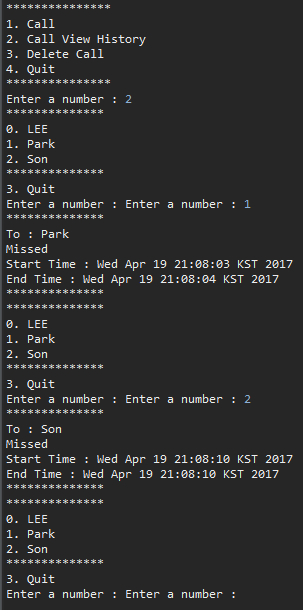
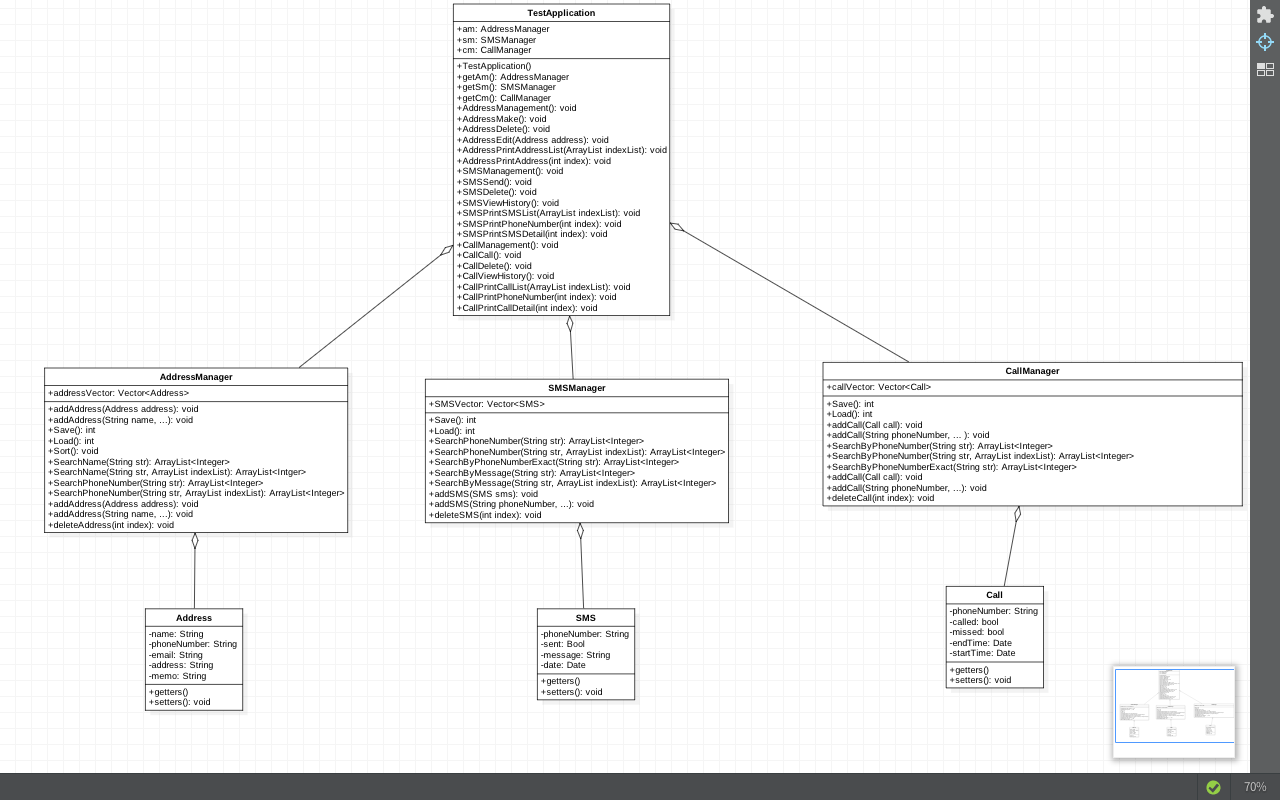
이호준 2016024984

이상진 2016024939

1. 메인메뉴에서 1번을 선택하므로써 Address에 들어갈 수 있다. 하위 메뉴인 Address에는 주소록 보기, 새로운 주소록 추가, 주소록 삭제, 상위 메뉴로 돌아가기가 있으며, 왼쪽 그림처럼 2번을 선택하여 새로운 주소록을 만들 수 있으며, 저장하고 메뉴로 나간 뒤에 1번을 선택하여 저장되어있는 주소록을 전체적으로 볼수 있다. 또한, 3번을 선택하여 원래 있던 주소록 중 삭제할 주소록을 선택하여 삭제할 수도 있다.

왼쪽 그림은 주소록에 20개의 주소록을 입력 받아 저장하였다.

1. 메인 메뉴에서 2번을 선택하면 위 그림과 같이 SMS 메뉴로 들어가지게 된다. SMS를 보내거나 이미 보낸 SMS를 열람하거나 아니면 SMS를 삭제할 수 있다. 첫번째 그림과 두번째 그림은 새로운 SMS를 작성하는 과정이며, 마지막 그림은 이미 보낸 SMS를 열람하는 과정이다. View SMS에서 SMS를 언제 보냈는지는 시스템 시간으로 계산하여 저장된다.
2. 왼쪽 그림은 메인 메뉴에서 3번을 선택하여 Call메뉴에 들어간 뒤 Call하기를 선택하여 수신인을 입력하고 Call하는 과정을 기록한 화면이다. 사용자는 수신인을 직접 입력하며, 입력한 다음 Call을 누르면 수신이 간다. 전화를 다 한다음에는 To end를 눌러 전화를 끊을 수 있다. 이렇게 전화를 하면 Call View history에 자동으로 저장되어 열람할 수 있다.
3. 오른쪽 그림은 위 과정을 통하여 Call을 한 뒤 자동으로 저장된 기록을 열람하는 과정이다. 사용자는 열람하고 싶은 기록을 선택하여 열람할 수 있다. SMS와 마찬가지고 시스템 시간도 저장되어 수신 날짜를 알 수 있다. 또한, 메뉴에서 3번을 선택하여 삭제하고 싶은 기록을 삭제 할 수도 있다.
4. 본 프로젝트 진행에 사용된 UML Diagram이다. Test application을 중심으로, Address Manager, SMS Manager, Call Manager을 구성하였으며, 사용된 함수나 클래스는 위 다이어그램에 나타나 있는것과 일치한다.

이번 프로젝트는 C++로 짜여진 첫번째 프로젝트를 Java 기반으로 다시 구현하는 것이었습니다. 프로그램 자체의 설계는 거의 비슷하게 하였지만, 기본적으로 다른 언어로 프로그램을 짜는 것이기 때문에 큰 틀의 차이보다 사소한 차이가 더 많이 보였습니다. 첫번째 프로젝트 대부분을 “재사용”하였지만, 말 그대로 재사용이 아닌, 큰 틀과 맥락은 똑같이 하면서 디테일한 부분은 이번 프로젝트에 맞도록 설계하였습니다.

이번 프로젝트를 진행하면서 첫번째 프로젝트와는 다르게 Text File 대신 XML과 JSON 기반으로 파일을 설계하였습니다. 매번 사용하던 Text File과 양식이 달라 처음에는 많이 생소하였지만, 이런 포맷 자체가 속성과 값을 쌍으로 제공하는 표준 포맷이라서 사용하면 사용할수록 더 편리하였습니다. 또한, 독립형 데이터 포맷으로서 Java에 더욱 쉽게 사용할 수 있었습니다.