

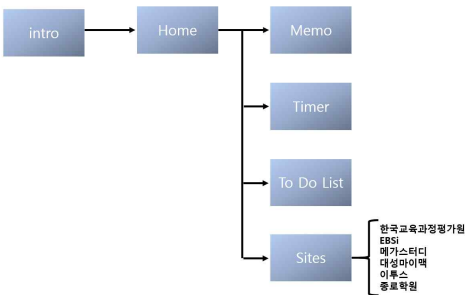
I. 프로그램 개발의 동기 및 목적

현재 공부 일정을 관리해주는 앱은 상당히 보편화되어있다. 그러나, 다양한 기능 제공에 집중하여 공부 일정을 관리하는 기본적인 기능 외에 다른 부가적인 기능들이 포함되어 있는데, 이러한 부가 기능들은 사용 빈도가 낮고 활용도가 떨어지므로 사용자에게 불편함을 미친다. 따라서, 본 개발자는 필요한 기능만을 선별하고, 인터넷 강의 사이트 접속 기능 및 빠른 기능 등 새로운 기능을 추가한 프로그램을 개발하였다. 본 프로그램은 효율적인 학습 일정 관리에 도움이 될 것으로 생각된다.

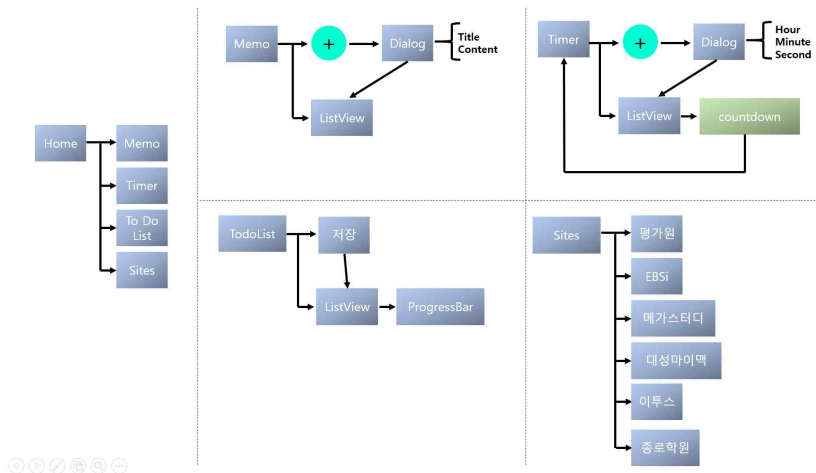
II. 프로그램 소개

본 프로그램은 총 4개의 프래그먼트로 구성되어 있다. Home은 메인화면으로, 명언과 학습 진행률, 빠른 메모 기능을 지원한다. Memo는 메모 화면으로, 리스트뷰에 메모를 추가하여 확인 및 수정, 삭제가 가능하다. 확인은 짧은 터치, 수정 및 삭제는 긴 터치로 할 수 있다. Timer는 타이머이다. 일반 타이머처럼 EditText에 시간을 입력하면 타이머를 실행시킨다. 기존 타이머와의 차별점은 타이머 저장 및 디폴트 시간 제공이다. 사용자가 직접 필요한 타이머를 저장해 언제든 리스트뷰에서 타이머를 불러올 수 있으며, 디폴트 시간을 통해 실제 수능 시간을 편리하게 맞출 수 있다. To Do List는 일정 및 계획 관리 화면으로, 리스트뷰에 계획을 빠르게 저장하고 이행 여부를 체크할 수 있다. Sites는 한국교육과정평가원, ebsi 및 기타 수능 전문 학원 사이트에 빠르게 접속할 수 있는 이미지버튼들을 제공하고 있다. 이러한 프로그램으로써 수능을 대비하는 학생은 자신의 학습 과정을 면밀히 검토하고 효율적인 학습을 진행할 수 있다.

III. 프로그램 구성도

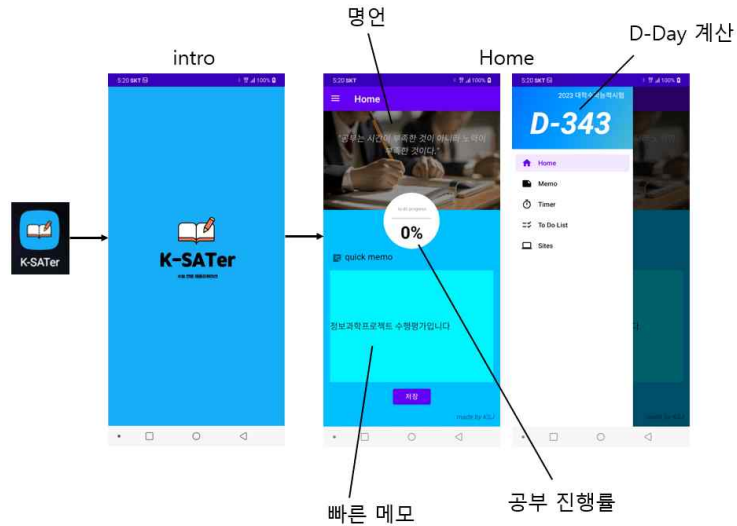


IV. 사용자 메뉴 구성도



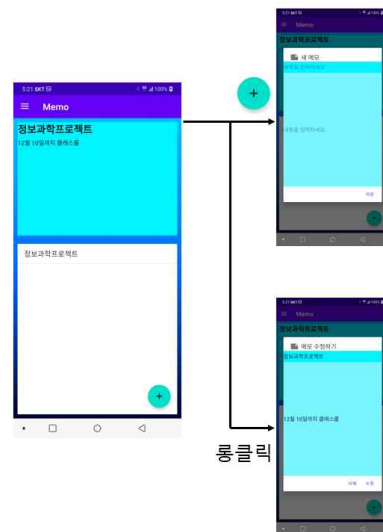
Ⅳ. 주요 실행 결과

가. intro 및 Home



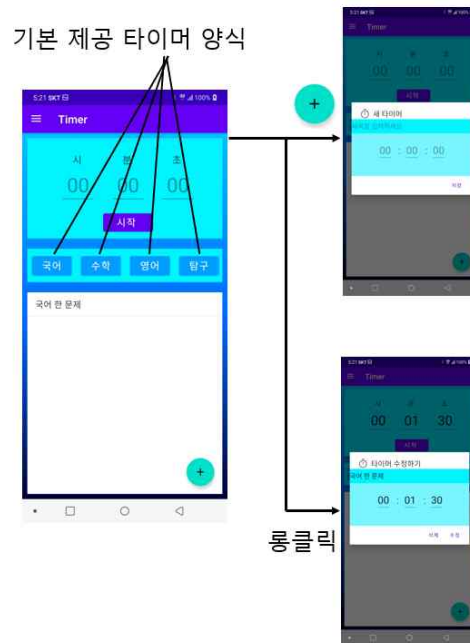
앱을 실행하면 간단한 intro 이후 바로 앱을 사용할 수 있다. Main 화면에는 명언과 빠른 메모, 공부 진행률 표시 기능이 지원된다. 네비게이션뷰를 열면 D-Day 계산과 함께 여러 메뉴 아이템들이 나타난다.

나. Memo



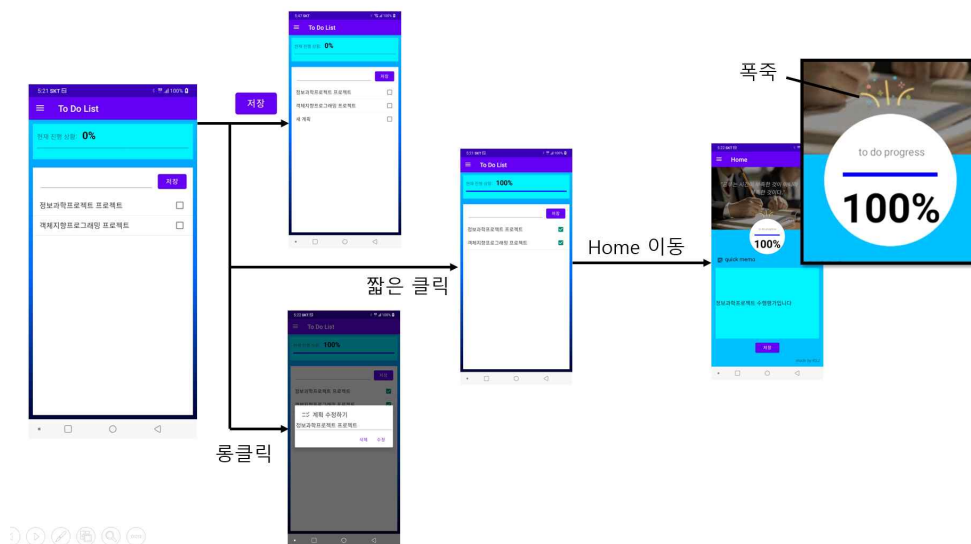
Memo 화면에서는 메모를 작성할 수 있다. floating button을 누르면 메모 추가가 가능하고, listview의 item을 롱클릭하면 메모 수정 및 삭제가 가능하다. 메모 확인은 짧은 클릭으로 가능하다.

다. Timer



Timer 화면에서는 타이머를 실행할 수 있다. Memo와 같이 버튼과 롱클릭으로 타이머를 만들고 고치는 것이 가능하다. 기본 제공 타이머 양식 버튼은 국어, 수학, 영어, 탐구 과목의 실제 수능 시험 시간을 EditText에 집어 넣어주어 편리한 타이머 사용이 가능하게끔 한다. 시작 버튼을 누르면 타이머가 시작된다. 타이머 종료시 화면이 빨간색으로 변한다.

라. To Do List



To Do List 화면에서는 사용자가 계획을 빠르게 입력하고 체크하여 계획의 추가 및 진행 상황을 확인할 수 있다. 계획 입력 후 저장을 누르면 계획이 listview에 추가되고, 롱클릭으로 수정 및 삭제를 할 수 있다. 계획 진행률은 To Do List 화면의 ProgressBar 뿐 아니라 Home 화면의 ProgressBar에도 적용되며, 100%를 달성하면 Home 화면에서 폭죽이 나타난다.

마. Sites



Sites 화면에서는 총 6개의 이미지버튼이 나타나며, 각각을 누르면 빠르게 해당 사이트로 연결되어 필요한 정보를 얻을 수 있다. 실시간 등급컷이나 기타 정보를 얻을 때 여러 사이트를 돌아다니며 비교하게 되는데, 이를 편리하게 할 수 있는 본 프로그램만의 기능이다.

V. 기술적 요소

To Do List에서 아이템을 수정 또는 삭제하면 체크된 항목이 바뀌는 현상이 나타났다. 여러번의 시행 결과 listview의 item checked 여부는 항목이 삭제돼도 index 순대로 그 순서가 유지됨을 알 수 있었다. 따라서, 계획 삭제시 item checked 여부를 따로 수정해주어야 했다.

```
for(int j = i; j < list.getCount() - 1; j++) {  
    list.setItemChecked(j, list.isItemChecked( position: j+1));  
}  
list.setItemChecked( position: list.getCount() - 1, value: false);
```

다음은 그 코드의 일부이다. 삭제된 지점의 인덱스가 i일 때, 기존 인덱스에 해당되는 checked 여부 값을 기존 인덱스보다 1 더 큰 인덱스에 해당되는 checked 여부로 변경하는 작업을 수행한다. 즉, i+1 인덱스부터의 boolean 값들을 앞으로 당겨 저장하는 방식이다. 이렇게 함으로써 중간 지점의 item이 삭제되도 item 별 checked 여부가 달라지지 않도록 하였다.

VI. 참고 문헌 및 웹 사이트

<https://junyoru.tistory.com/80> (대화상자 사이즈 변경 방법)

<https://aries574.tistory.com/73> (타이머 구현)

<https://stackoverflow.com/questions/34973456/how-to-change-text-of-a-textview-in-navigation-drawer-header> (Navigation Drawer Header의 TextView의 text MainActivity에서 변경하는 방법)

<https://jamesdreaming.tistory.com/103> (D-Day 계산)

<https://haruple.tistory.com/150> (intro 제작 방법)

Ⅶ. 기대효과 및 개발 소감

본 프로그램은 간단명료함과 유용함을 동시에 갖추고 있다. 기존의 복잡한 프로그램보다 접근성이 높으며, 따라서 공부에 더 유용한 도움을 줄 수 있을 것으로 예상된다. 특히, 예상 사용자를 수능 수험생으로 제한하여 특화된 기능을 제공함으로 수능 수험생에게 가장 적합한 기능을 제공할 것으로 생각한다. 프로젝트를 진행하며 파일 입출력이 생각보다 까다로운 과정임을 알게 되었고, 특히 try-catch문을 적절히 사용하여 에러를 잡아낼 뿐 아니라 프로그램을 원하는 방향으로 진행시킬 수 있음을 알게 되었다. listview와 파일입출력을 결합하여 내용을 저장하고 삭제하며 수정하는 기능을 구현하는 것이 가장 어려웠다. 각각 수행해도 복잡한 두 기능을 결합시키면서 코드가 길어지고 오류도 많이 발생했으나, 파일을 적절히 생성하여 저장함으로 해결하였다. 코드를 간결하고 가독성있게 만드는 것이 생각보다 어려운 작업임을 느꼈고, 제작한 코드를 보다 간결하게 만들지 못한 점이 아쉬웠다.