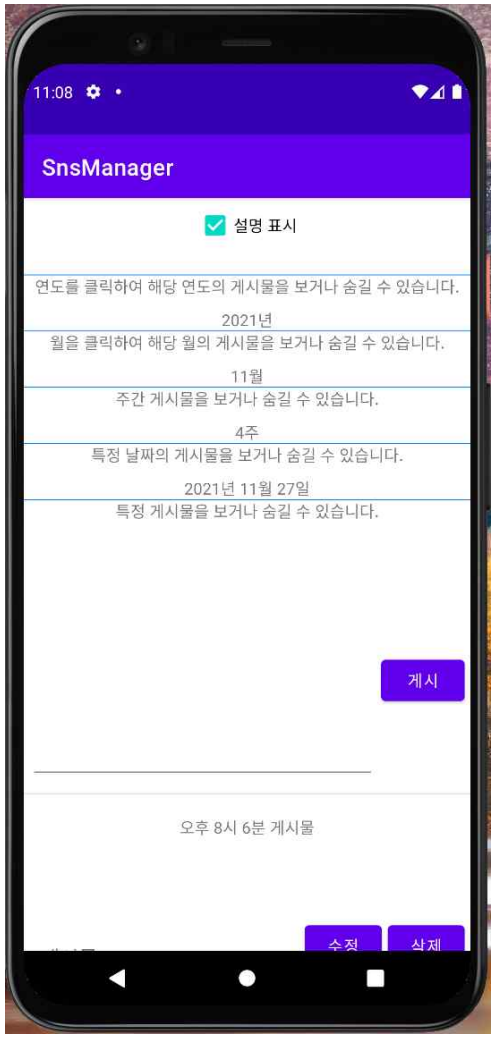
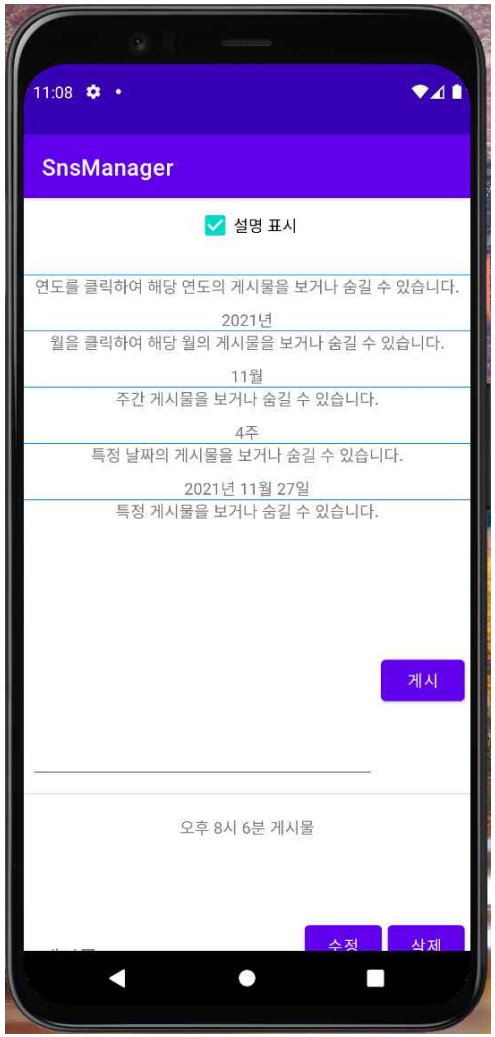


안드로이드 커스텀 역순 중첩 리스트 뷰

김진희

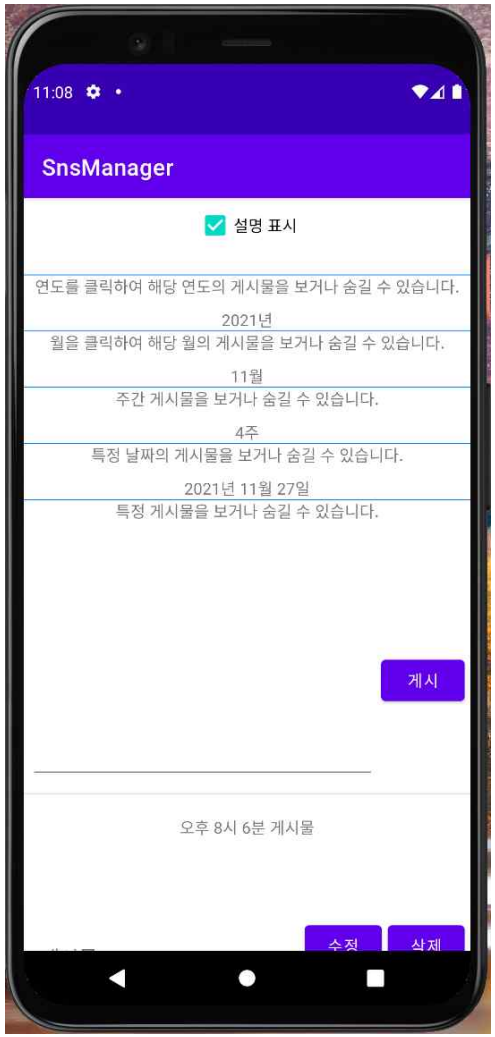


주제 선정 배경

- 안드로이드 버전 SNS 관리자 프로젝트의 UI로 기획한 형태

- 널리 사용되는 SNS는 타임라인을 제공하여 최신 순으로 작성 게시물을 볼 수 있다.

- 따라서 게시물이 쌓이면 작성한지 오래된 게시물은 찾기 어려워지므로, 년, 월, 주, 일 순으로 분류해서 게시물을 관리할 수 있는 앱을 만들고 싶었다.



왜 역순 중첩 리스트 뷰 인가?

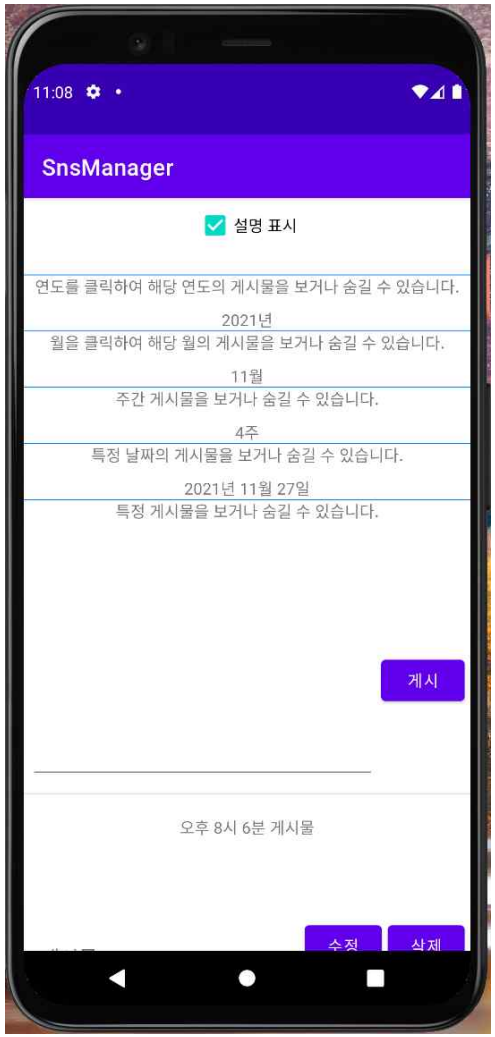
1. 역순인 이유: 앞서 언급했듯이 SNS난 최신 게시물을 최상단에 보여준다.

2. 따라서 왼쪽 그림과 같이 연, 월, 주, 일 별 최신 최신 게시물을 최상단에 보여주려면 각 기간별 내용물을 역순으로 표시해야 한다.

3. 기간 별 내용 보여주는 방법으로 생각해 낸 것

   1) 프래그먼트 안에 프래그먼트를 넣는 방식 (중첩 프래그먼트)

   2) 리스트 뷰 안에 리스트 뷰를 넣는 방식 (중첩 리스트뷰)



프래그먼트 vs 리스트 뷰

맨 처음에는 프래그먼트를 중첩하는 방식으로 시작

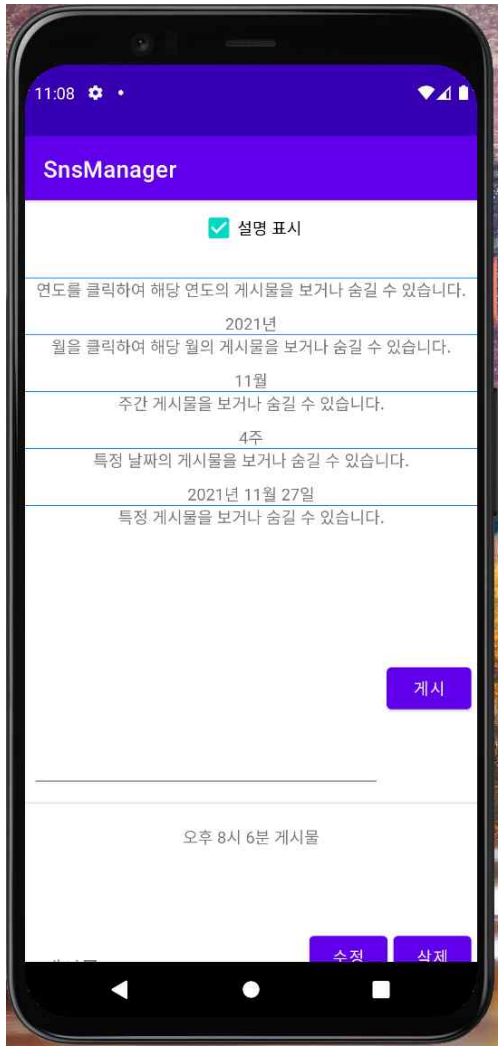
```
FragmentManager childFragmentManager = getChildFragmentManager();
childFragmentManager.beginTransaction()
        .add(R.id.inner_fragment, fragment)
        //.addToBackStack(null)
        .commitNow();
```

시도 후 발견한 문제점

- 차일드 프래그먼트를 만드는 것은 내부적으로는 레이아웃에 **add(View view)** 메서드를 호출하는 방식으로 구현

1) 역순으로 뷰를 배치하려면 만들어진 뷰를 제거(remove)하고 다시 배치하는 식으로 구현해야 함

2) 목록 형태로 다수의 item을 보여주는 데에 최적화된 뷰는 레이아웃이 아니라 리스트 뷰(**ListView**)이다.



중첩 리스트 뷰 구현

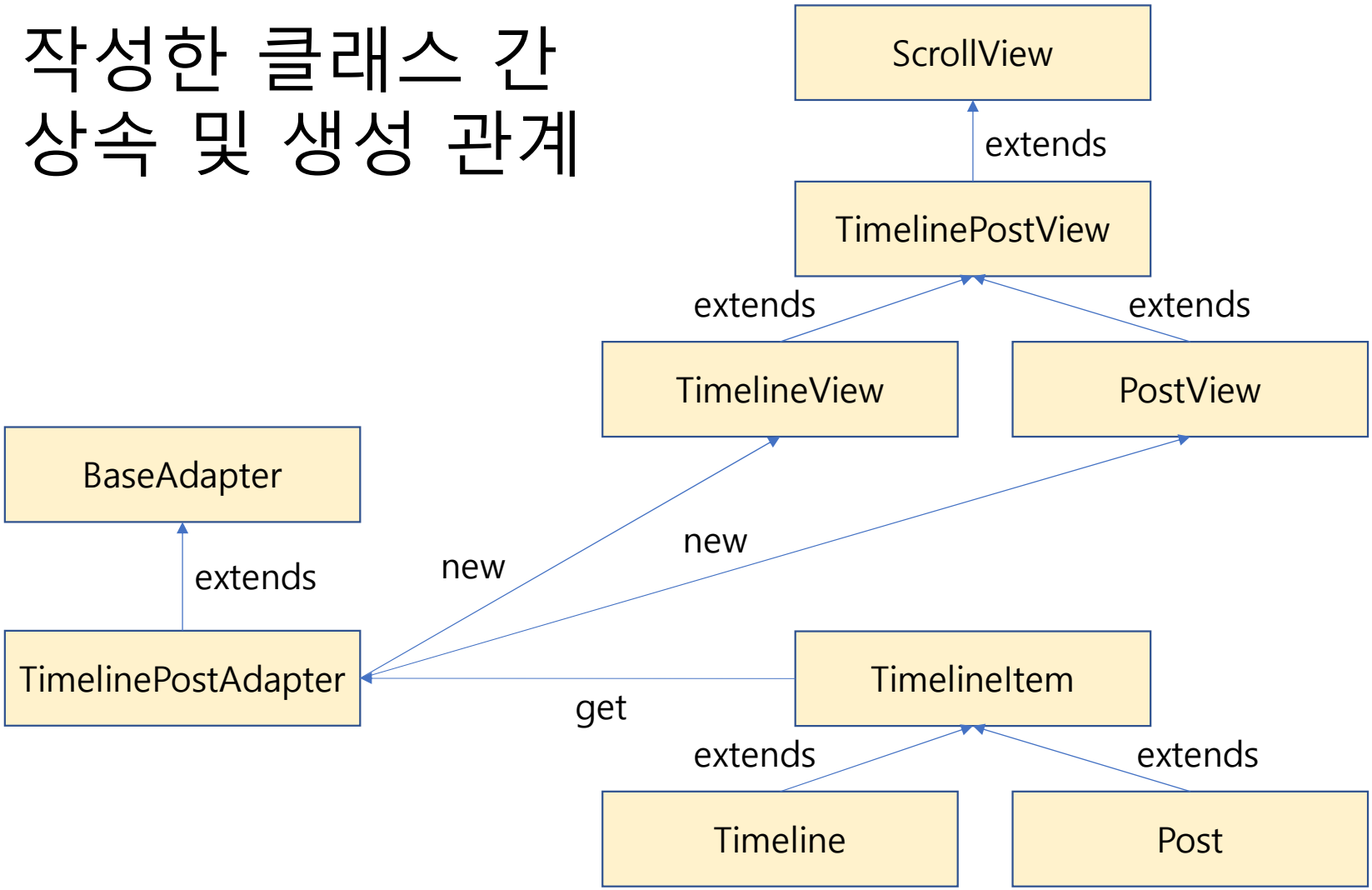
1. 리스트 뷰 (ListView)를 포함한 커스텀 뷰와 말단 리스트 뷰의 원소를 표현할 커스텀 뷰를 만든다.
   (timeline.xml, post.xml 파일 보기)
   - 각 뷰의 이름을 각각 TimelineView, PostView 라고 짓는다.

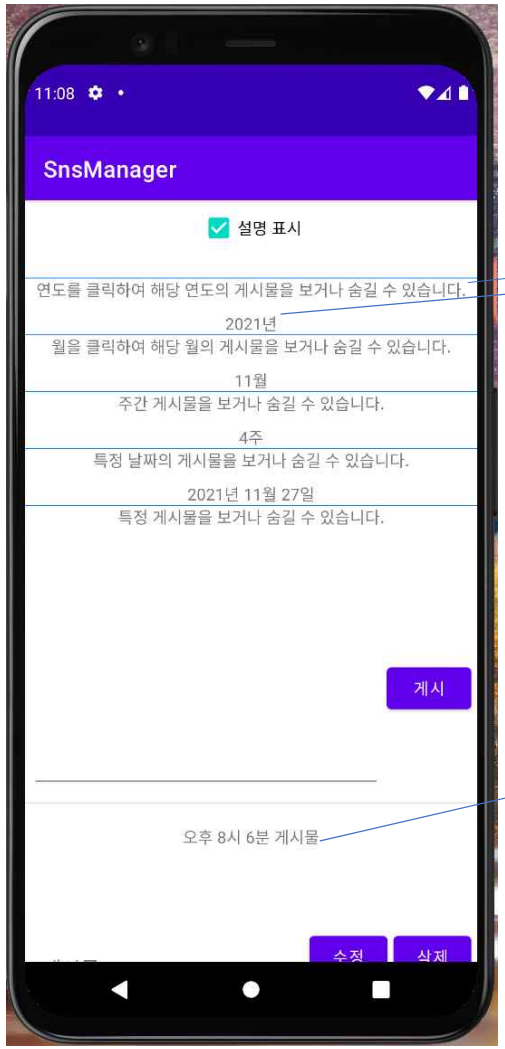
2. 리스트 뷰는 TimelineView와 PostView를 모두 포함할 수 있도록 공통 부모 클래스를 만들고
   (TimelinePostView), 각 뷰에서 표현할 데이터에 저장하는 클래스를 만든다.
   - TimelinePostView는 스크롤 뷰를 상속하여 만든다. (스크롤 뷰를 써야 높이 조절이 자유로움)

   뷰 클래스: TimelineView, PostView, TimelinePostView
   데이터 클래스: Timeline, Post, TimelineItem

3. BaseAdapter를 상속한 커스텀 어댑터를 만든다.
   어댑터 클래스: TimelinePostAdapter

작성한 클래스 간 상속 및 생성 관계



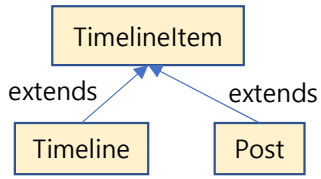


데이터 구조

```
public class Timeline extends TimelineItem {
    private final String information, time;
    private final Vector<TimelineItem> items;
```

Vector: ArrayList와 거의 비슷한 것

items는 TimelineItem에 대한 벡터로 Timeline 혹은 Post를 원소로 가질 수 있다.




```
public class Post extends TimelineItem {
    private String timeTitle;
    private final JSONObject jsonObject;
    private boolean writeMode;
```

게시물 내용
게시물 상태

게시물 생성과 삭제 구현

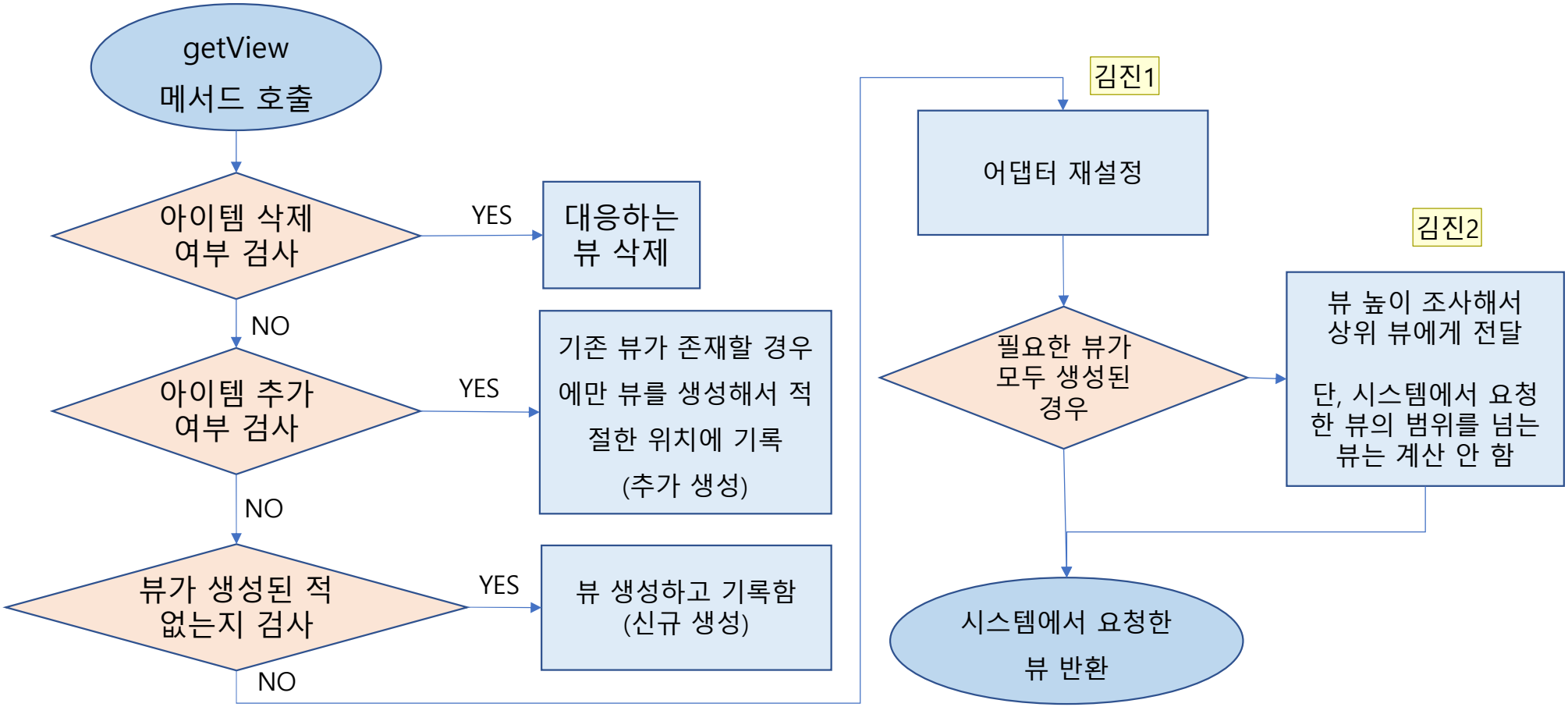
```
public class Timeline extends TimelineItem {
    private final String information, time;
    private final Vector<TimelineItem> items;
                    ⋮
    public void removeChild(TimelineItem timelineItem) {
        items.remove(timelineItem);
        if (items.size() <= 0) {
            super.requestRemoveThis();
        }
    }

    public void addTimeline(String information, String time) {
        Timeline childTimeline = new Timeline( parentTimeline: this, information, time);
        // 맨 위에 맨 마지막 것을 추가함
        items.add( index: 0, childTimeline);
    }

    public void addPost(String timeTitle, JSONObject jsonObject) {
        Post childPost = new Post( parentTimeline: this, timeTitle, jsonObject);
        // 맨 위에 맨 마지막 것을 추가함
        items.add( index: 0, childPost);
        //items.add(childPost);
    }
```

게시물 생성 및 삭제는 단지, Timeline에서 items 벡터에 데이터를 추가하거나 삭제만 하면, 어댑터가 감시해서 뷰를 생성하거나 삭제하여 사용자에게 보여준다.

어댑터의 구현 (가장 어려운 부분)



슬라이드 9

김진1 왜 그런지는 모르지만 이것을 해야지 높이가 제대로 계산됨

김 진희, 2021-11-28

김진2 뷰 인덱스 계산 안 하면 오류나고 프로그램 멈춤

김 진희, 2021-11-28

시연

이번 프로젝트의 의의

레이아웃과 리스트 뷰에 대해 더 잘 이해할 수 있었다.

Q & A

감사합니다