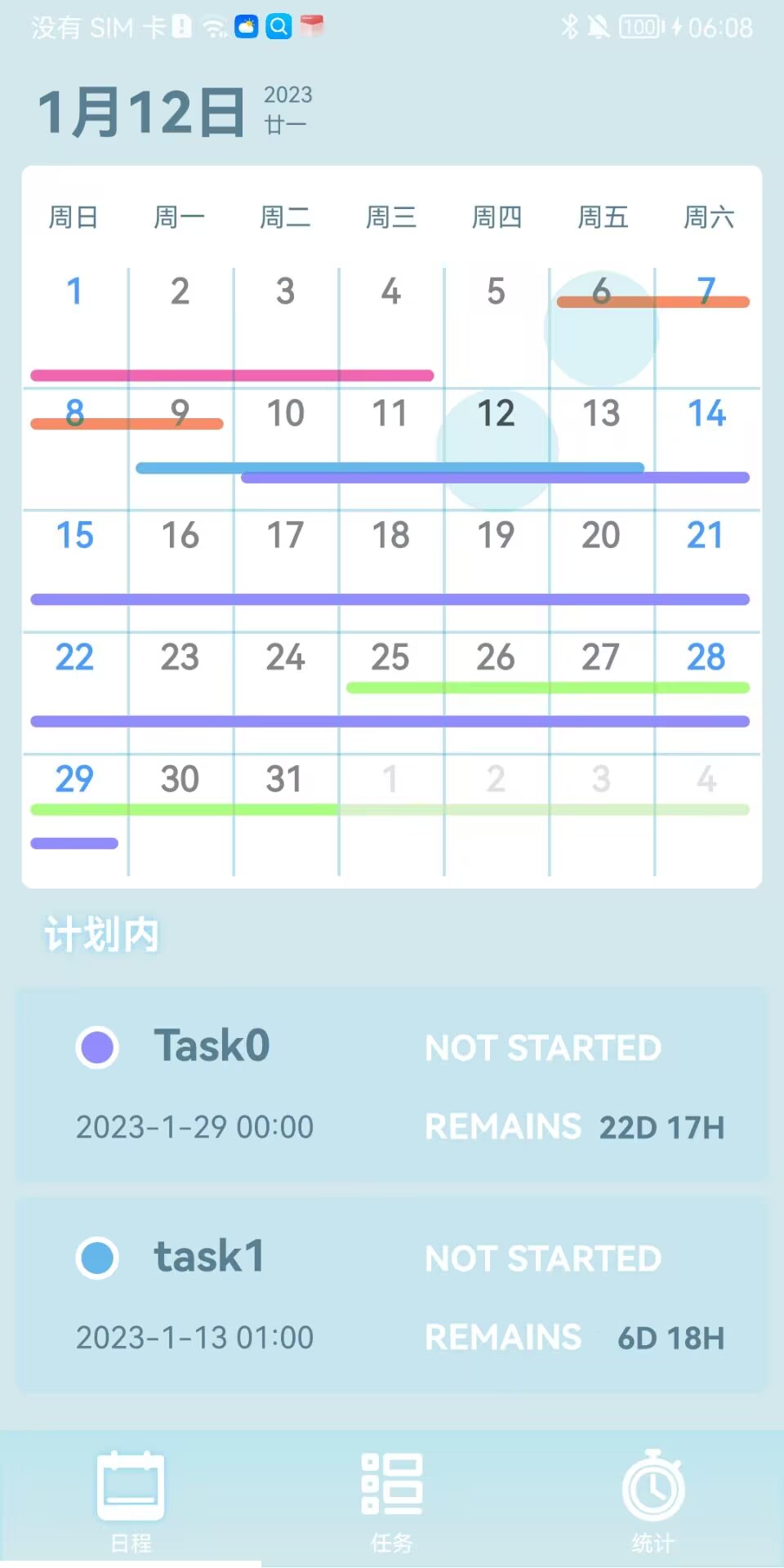
APP实验报告\_DeadLiner\_20302010055\_梁海媛

**一、主要架构；**

采用了单Activity多Fragment的结构，在MainActivity中使用ViewPager+TabView的结构进行导航，可以左右滑动到达不同页面，或点击Tab上的图标到达目标页面；分别制作了Carlendar界面、Cata界面、Task界面和Static界面，外加一个简单的widget小组件，接下来将一一进行讲解



**二、calendar界面**

Calendar界面包含一个日历视图以及一个展示用Recyclerview，先从日历视图介绍

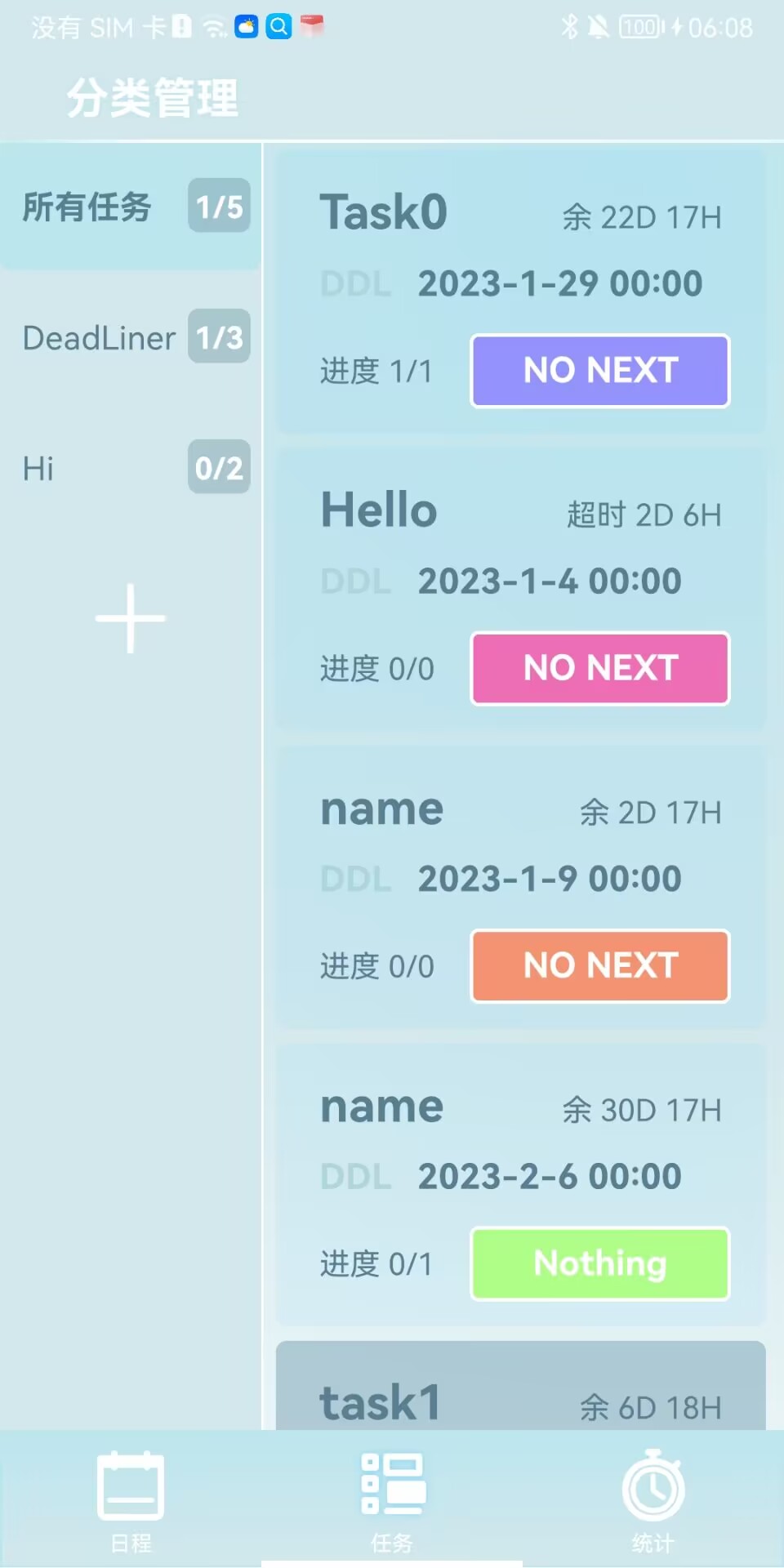
1.日历视图

日历视图使用了CalendarView类，它封装了canvas的绘图细节，以及一些基础的日历工具；这里主要的工作量是弄懂整个类怎么工作，简单来说，需要在界面初始化时传入需要的数据，然后自定义视图UI，我这里最终的一个展示如右图，还算比较完美地还原了开题报告时的预期，可以给不同的task指定不同颜色，并根据开始时间和DDL在日历视图上进行显示，其颜色条跨度即代表了task的持续时间；这里颜色的排布是根据task颜色的HSV值做的，所以如果排列得当的话其实可以得到一条彩虹（

日历可以上下滑动以及选择日期，选择日期的关联功能在下面介绍；

2.计划内RecyclerView

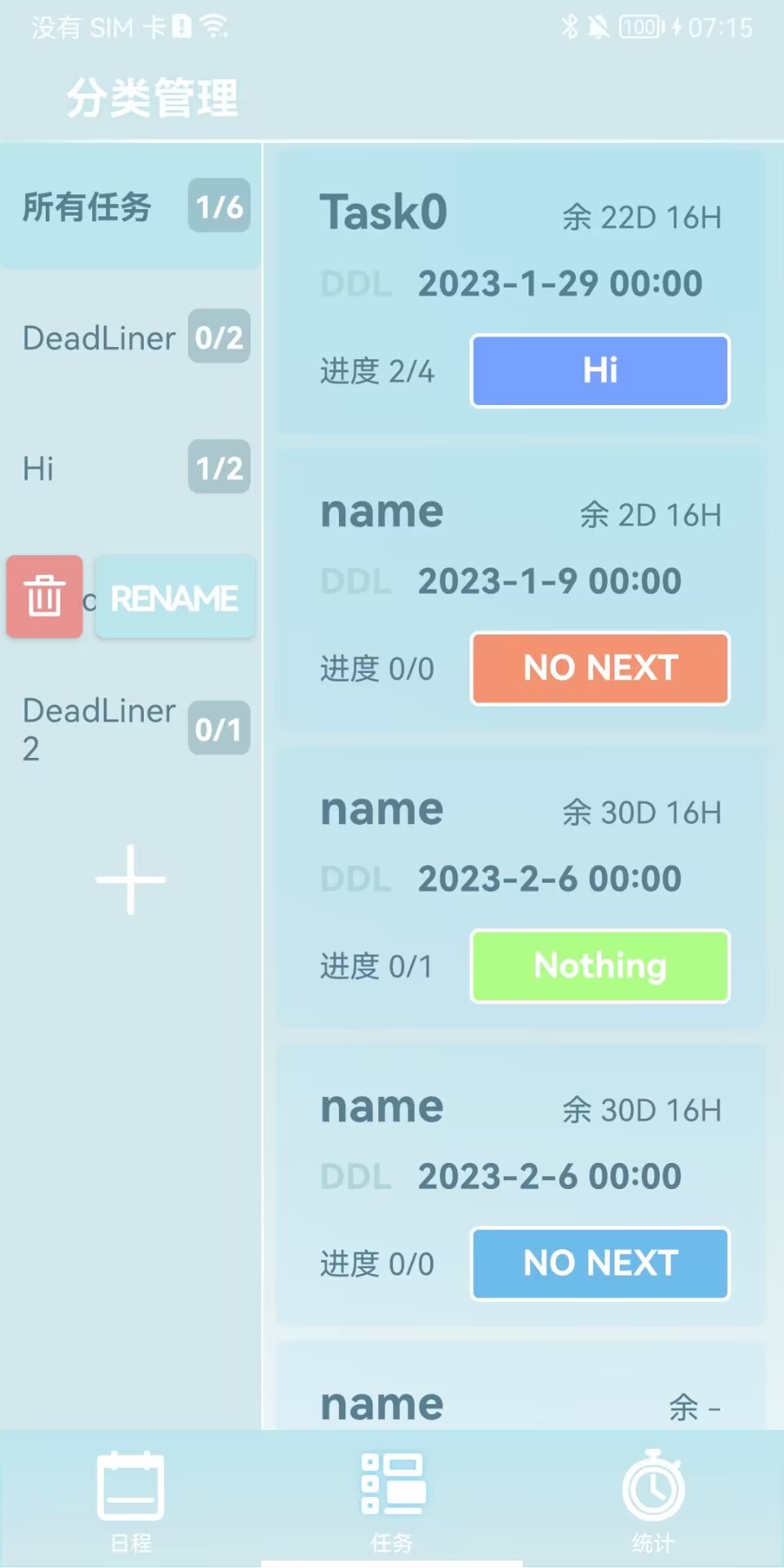
这个列表根据日历选择的日期更新那一天计划内的tasks（计划内意味着该日期落在开始时间与DDL之间）这边与日期先后相关的判断使用了Calendar类作为Util，比如剩余天数和是否已经开始该task（这两项的判断以真实的时间为基准）；另外左边的小圆点是自定义view实现的



**三、cata界面**

**cata**界面由左右两个RecyclerView构成，其中左边展示已有的分类，右边根据左边的分类选择展示其中的tasks；

1.Cata中展示了分类名称以及其中task的完成进度，点击分类可以切换右边的画面，点击加号能够新建一个分类；

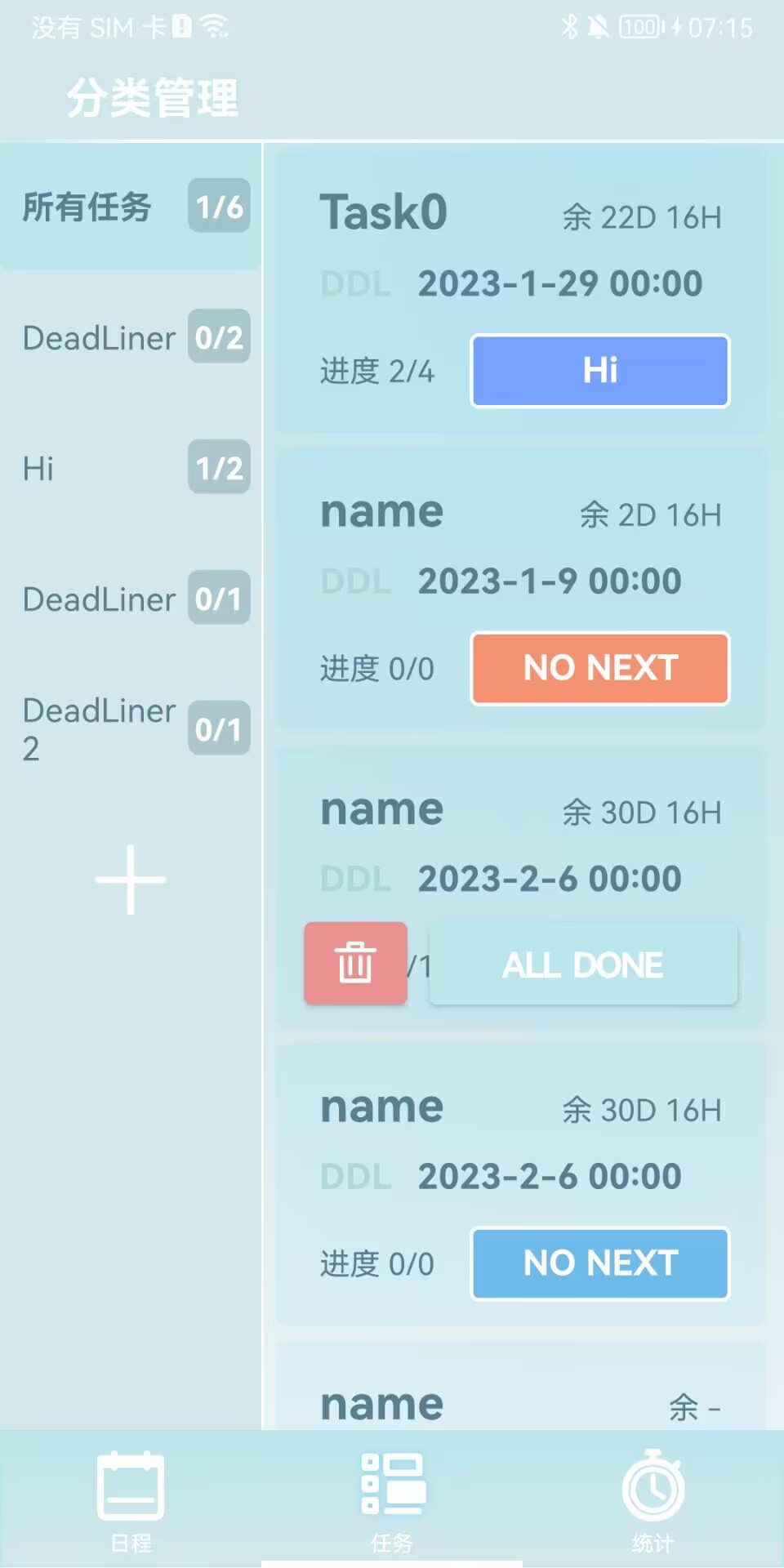
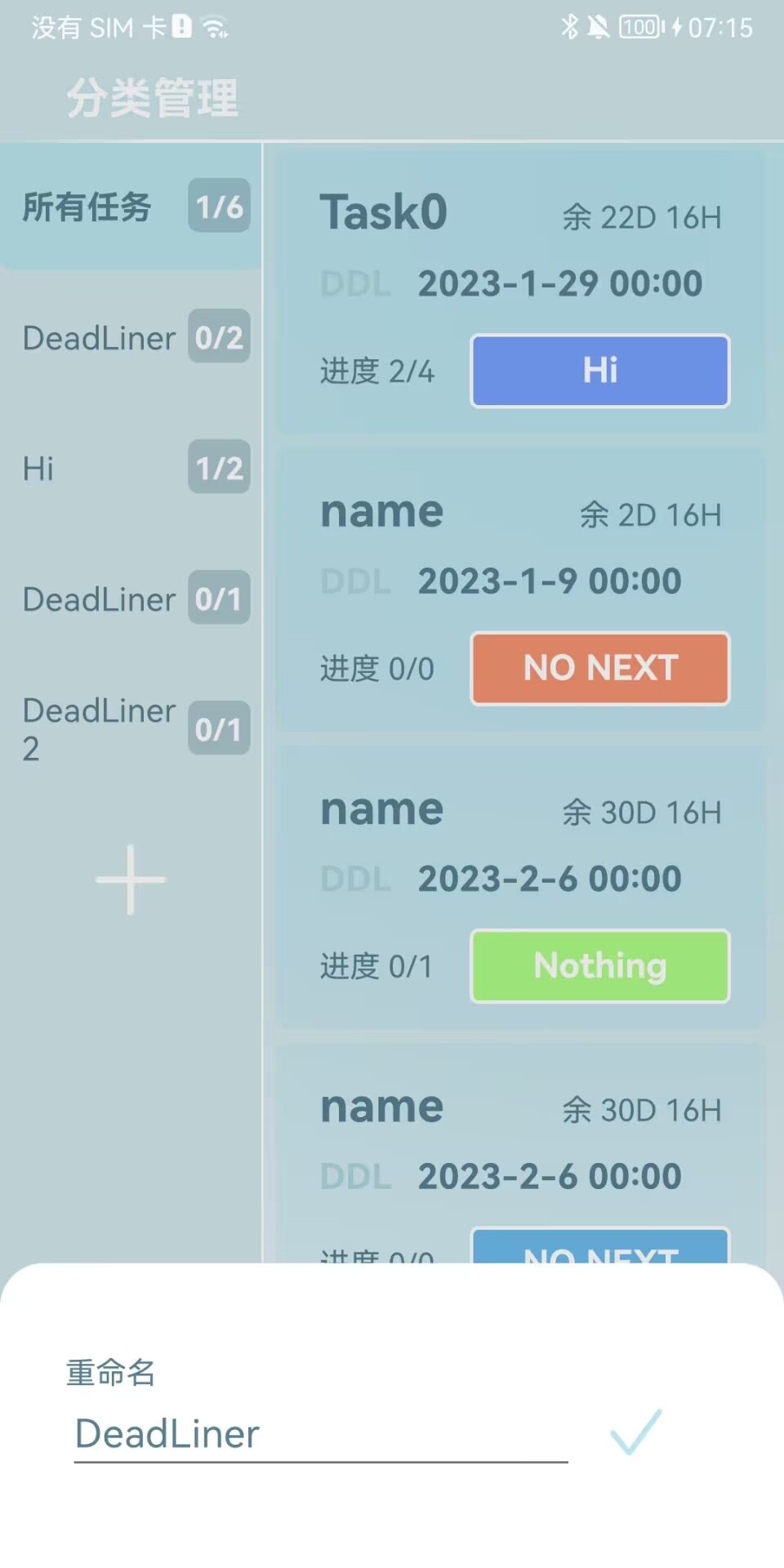


长按可以选择是否删除或重命名该分类，选择删除时会进行弹窗提示，选择重命名时会进行底部弹窗修改；

2.task\_review中展示了task的基本信息如名称、DDL、剩余时间、进度和下一个待办事项（没有则显示NO NEXT），其中待办事项的颜色是根据task储存的颜色值确定的，可以进入详情界面进行修改



长按可以选择删除该task或将task设置为全部完成（全部完成与其中细分是否完成不冲突），或者将已经全部完成的task取消全部完成，或选择删除，同样会进行弹窗提示；



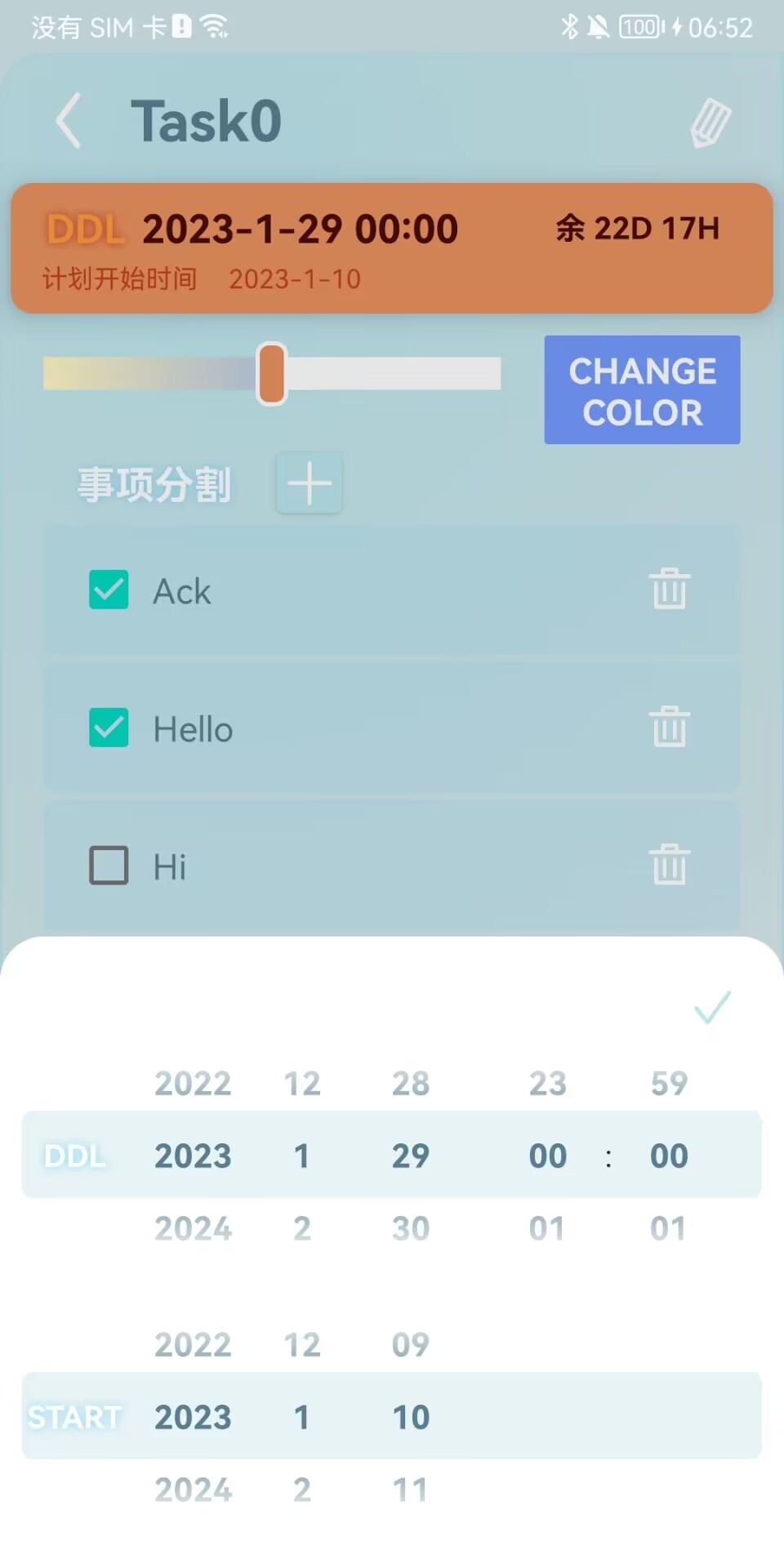
点击task可以进入Task\_info的详细内容界面，接下来将介绍；



**四、task\_info界面**

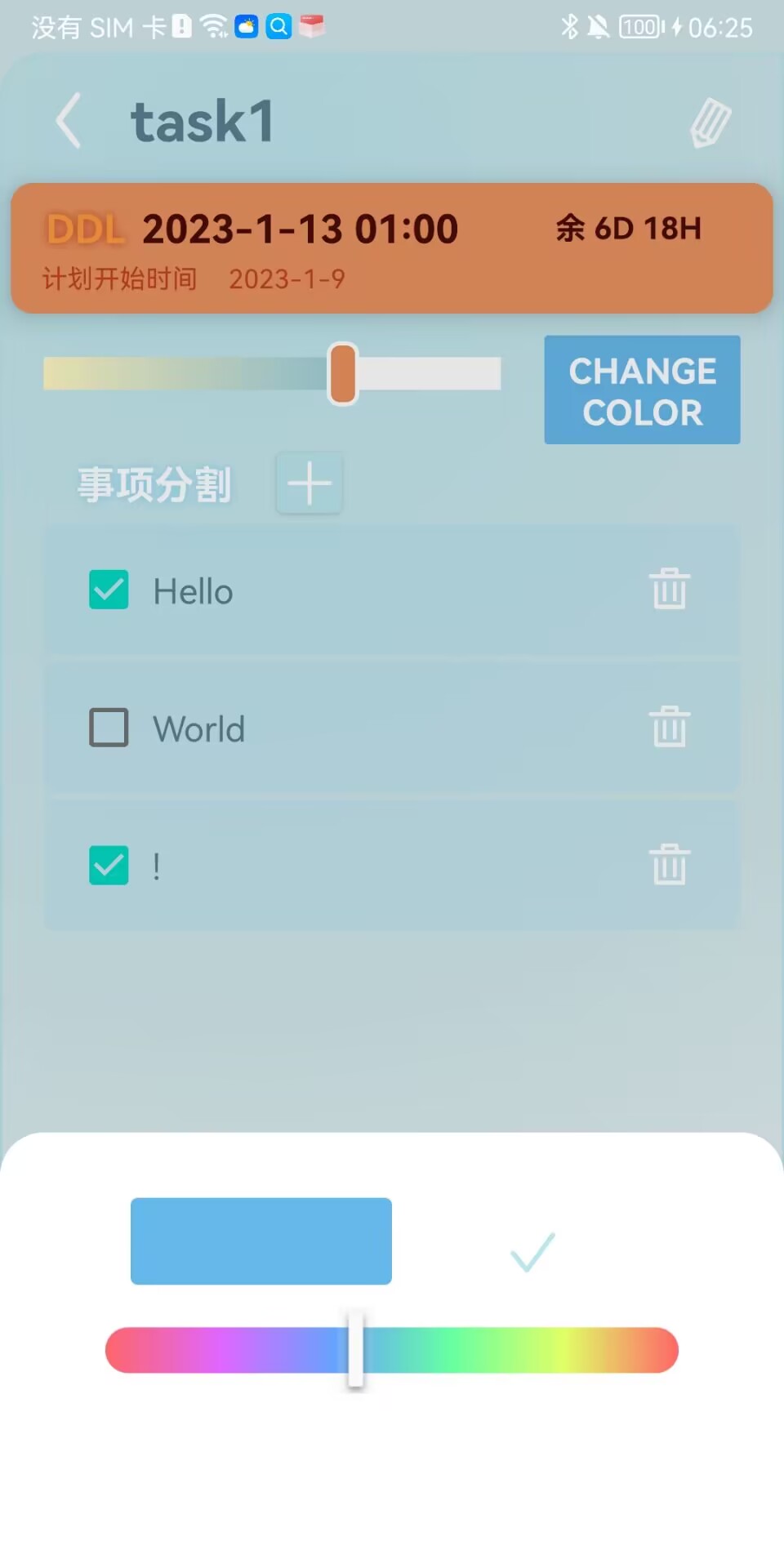
该界面包含了对于task的主要操作，点击右上角的铅笔可以重命名task，长按橙色框会出现弹窗修改ddl和start\_time，日期的选择由DatePicker和TimePicker重写UI实现；点击CHANGE COLOR会出现弹窗修改task颜色，task颜色由自定义view类实现，绘制渐变的view并绑定滑块位置与选定颜色（展示了一下界面）；

下半部分的列表展示了属于该task的事项分割，可以删除、修改名称以及点击checkbox完成，值得一说的是这里实现了拖动排序的功能（右图展示拖动时的view），可以拖动确定事项之间的顺序，该顺序会被存在数据库内以满足cata界面展示task的下一个代办事项的功能（即未被check的事项）具体由ItemTouchHelper实现，重写了一些交换位置后的逻辑代码，以完成顺序调整后的存储；

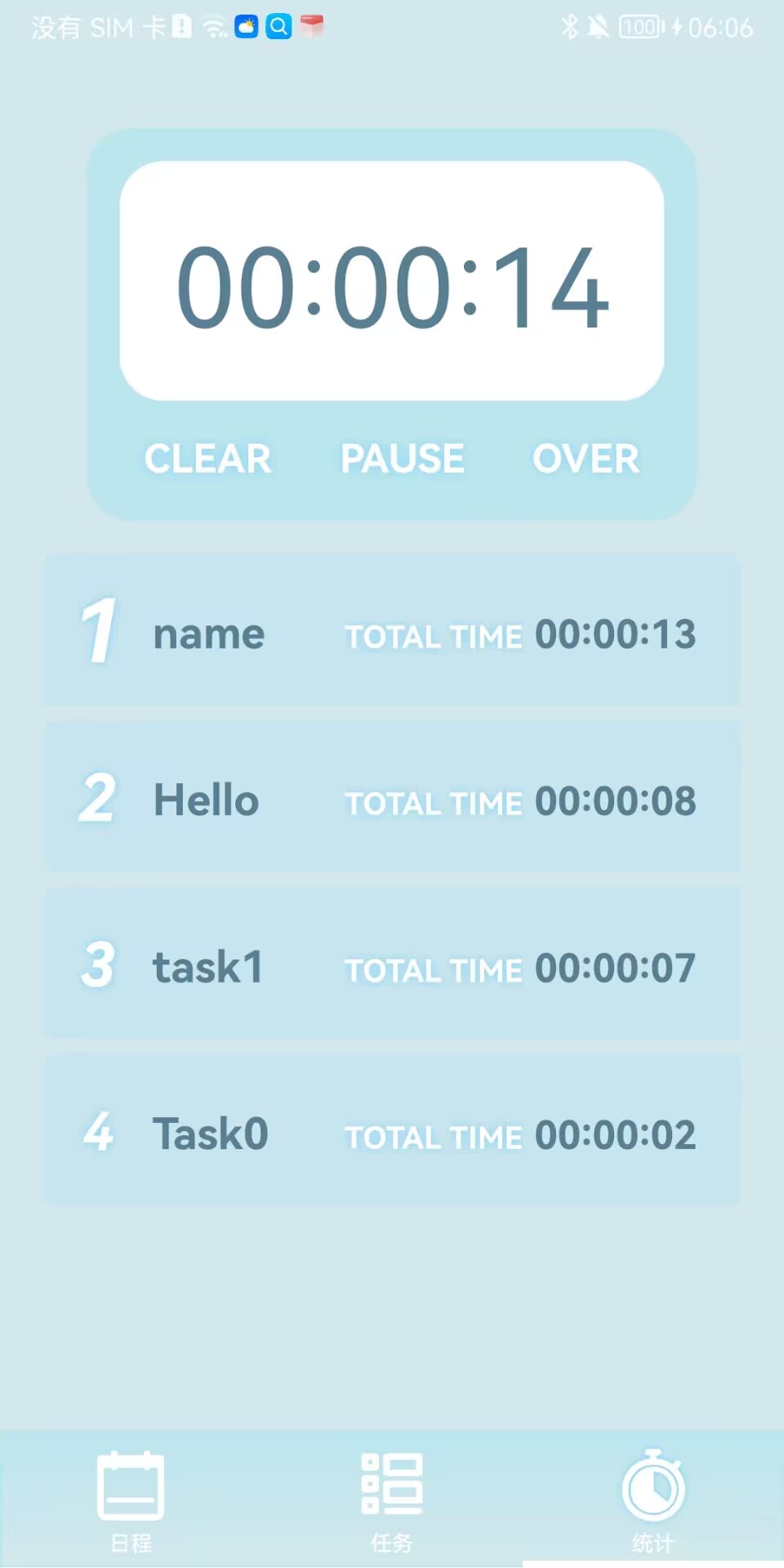


task颜色会决定左侧完成情况进度条的配色，事项分割的完成情况会决定进图条的进度情况，进图条由LinearLayout包裹后设置layout\_weight实现；

**五、static界面**



该界面实现了一个较为简单的计时器，以及历史以来花费时间的展示列表，计时器可以开始、暂停、清空和结束，计时计算由子线程完成，主线程更新UI；点击结束后会出现弹窗令其选择适配的task，确定之后，数据库将每段时间分别入库，新的总时间花费排名会更新在下方的列表中显示时间排名、task名称和总用时，总用时由数据库的聚合函数得到，这一页的功能相对简单；



**六、widget组件**



由于时间原因，Widget也只做了很简单的尝试，即显示最紧迫的未完成DDL，当task界面进行DDL时间修改，或新建task时，MainActivity会进行广播更新组件

**七、其他**

除了业务基本实现以外，这次还着重注意了配色、排版和UI设计，例如这次APP中的所有icon都是另外画的，并且对APP内部件的阴影、发光和半透明效果进行了额外处理，当然DeadLiner并不完美，在一些代码健壮性或排版或功能实现上还有待提高，不过这些会在不久的将来得到修缮，这学期的代码编写使我受益良多。