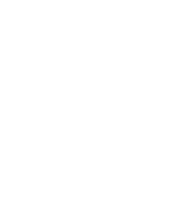
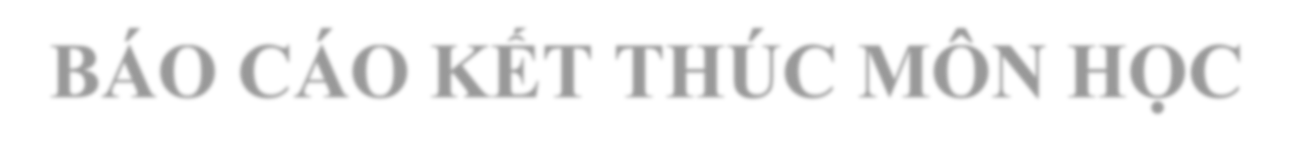
TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC

**KHOA CÔNG NGHỆ THÔNG TIN**

**Lập trình android nâng cao**



**BÁO CÁO KẾT THÚC MÔN HỌC**

**Xây dựng ứng dụng “Tiện ích về thời gian và chuyển đổi”**

|  |
| --- |
| **Giảng viên hướng dẫn: Trương Bá Thái** |
| **Sinh viên thực hiện: Nhóm 3**   1. **Nguyễn Thu Tuyết** 2. **Ngô Thị Ngọc Diễm** 3. **Phạm Minh Hiển** 4. **Nguyễn Anh Tuấn** 5. **Lê Minh Tân**   **Khoa: Công nghệ thông tin Khóa: 2016** |

### Ngày 09 /04/2019

**NHẬT KÝ HOẠT ĐỘNG NHÓM 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **HỌ TÊN** | **CÔNG VIỆC** | **TỰ ĐÁNH**  **GIÁ** | **NHÓM ĐÁNH**  **GIÁ** | **KÝ TÊN** |
| **1** | Nguyễn  Thu  Tuyết | Nghiêm cứu đề tài | 40% | 40% |  |
|  |  | Viết SRS |  |  |
|  |  | Viết SDS |  |  |
|  |  | Viết powerpoint nhóm 3 |  |  |
|  |  |  |  |  |
|  |  | Làm màn hình chính, màn hình đồng hồ, làm màn hình lịch dương, âm  Tổng hợp code nhóm |  |  |
|  |  |  |  |  |
|  |  | Báo cáo cuối kỳ |  |  |
| **2** | Ngô | Nghiêm cứu đề tài | 15% | 15% |  |
|  | Thị | Viết SRS |  |  |
|  | Ngọc  Diễm | Viết SDS |  |  |
|  |  | Màn hình chuông báo |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **3** | Phạm | Nghiêm cứu đề tài | 15% | 15% |  |
|  | Minh  Hiển | Viết SRS |  |  |
|  |  | Viết SDS |  |  |
|  |  | Làm màn hình đếm ngược |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **4** | Nguyễn | Nghiêm cứu đề tài | 15% | 15% |  |
|  | Anh | Viết SRS |  |  |
|  | Tuấn | Viết SDS |  |  |
|  |  |  |  |  |
|  |  | Làm màn hình chuyển đổi đơn vị |  |  |
|  |  |  |  |  |
| **5** | Lê | Nghiêm cứu đề tài | 15% | 15% |  |
|  | Minh  Tân | Viết SRS |  |  |
|  |  | Viết SDS |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  | Làm màn hình bấm giờ |  |  |
|  |  |  |  |  |

Mục lục

[CHƯƠNG 1. MỞ ĐẦU 4](#_bookmark0)

* 1. [Giới thiệu môn học và nhóm thực hiện 4](#_bookmark1)
  2. [Mô tả ứng dụng 4](#_bookmark2)
  3. [Ứng dụng trong cuộc sống 4](#_bookmark3)

[CHƢƠNG 2 PHÂN TÍCH CẤU TRÚC HỆ THỐNG 5](#_bookmark4)

* 1. [Phân tích thiết hệ thống 5](#_bookmark5)
  2. [Thiết kế hệ thống 19](#_bookmark6)

[CHƢƠNG 3. CÀI ĐẶT 34](#_bookmark7)

* 1. [Màn hình chính 34](#_bookmark8)
  2. [Màn hình báo thức 35](#_bookmark9)
  3. [Màn hình bấm giờ 36](#_bookmark10)
  4. [Màn hình đếm ngược 37](#_bookmark11)
  5. [Màn hình chuyển đổi 38](#_bookmark12)

[CHƢƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC 40](#_bookmark13)

* 1. [Kết quả đạt được 40](#_bookmark14)
  2. [Kết luận 40](#_bookmark15)

# CHƯƠNG 1. MỞ ĐẦU

## Giới thiệu môn học và nhóm thực hiện

Môn di động 2 là môn học kế thừa và nâng cao của di động 1. Môn học này giúp sinh viên áp dụng phát triển kỹ năng lập trình, tự tiềm kiếm, cập nhật kiến thức mới giúp áp dụng cho các dự án thực tế. Đồ án lập trình di động 2 là tổng hợp những kiến thức có được từ những môn học trước, tuy nhiên do thời gian thực hiện ngắn nên có thế xuất hiện nhiều hạn chế và lỗi phát sinh ngoài ý muốn.

Kiến thức mới:

Thành viên thực hiện:

* Nguyễn Thu Tuyết (Trưởng nhóm)
* Ngô Thị Ngọc Diễm
* Phạm Minh Hiển
* Nguyễn Anh Tuấn
* Lê Minh Tân

## Mô tả ứng dụng

* + - Tên ứng dụng: App Chuyển Đổi Thời Gian Và Đơn Vị
    - Cần kết nối mạng
    - Android 4.0+
    - Xem giờ, báo thức, bấm giờ, đếm ngược và kèm theo chuyển đổi đơn vị
* Giới thiệu:

Ứng dụng được nhóm thực hiện trong suốt quá trình học của môn lập trình di động 2,

Xây dựng trên nền tảng android 8. (SDK verson 28) và phiên bản SDK thấp nhất là SDK verson 24. Hỗ trợ tốt cho các thiết bị có độ phân giải màn hình từ full HD trở lên và kích thước 4,5 incher.

## Ứng dụng trong cuộc sống

Chỉ với chiếc điện thoại thông minh sau khi đã cài đặt ứng dụng “Tiện ích chuyển đổi thời gian và đơn vị” sẽ được lưu cố định trong máy.

Tên tiện ích đã nói lên tất cả những gì trong app, người sử dụng dễ dàng thao tác trên chiếc dế yêu của mình tiện lợi mang đi: VD: xem giờ, ngày tháng, đặt báo thức kèm với ghi chú công việc để chuông báo là tự động app sẽ làm hết mọi chuyện.

Chuyển đổi thời gian, thời tiết một cách đơn giản, bớt phần nhọc nhằn trong việc tính toán chính xác.

Ứng dụng tuy không lớn nhưng về mặt lợi ích đem lại khá ổn cho người dùng. Giúp

người dùng không trễ công việc. Đồng thời góp phần không hề nhỏ cho việc làm trọng

tài cho các cuộc thi liên quan về thời gian, có chức năng đếm ngược.

# CHƯƠNG 2 PHÂN TÍCH CẤU TRÚC HỆ THỐNG

## Phân tích thiết hệ thống

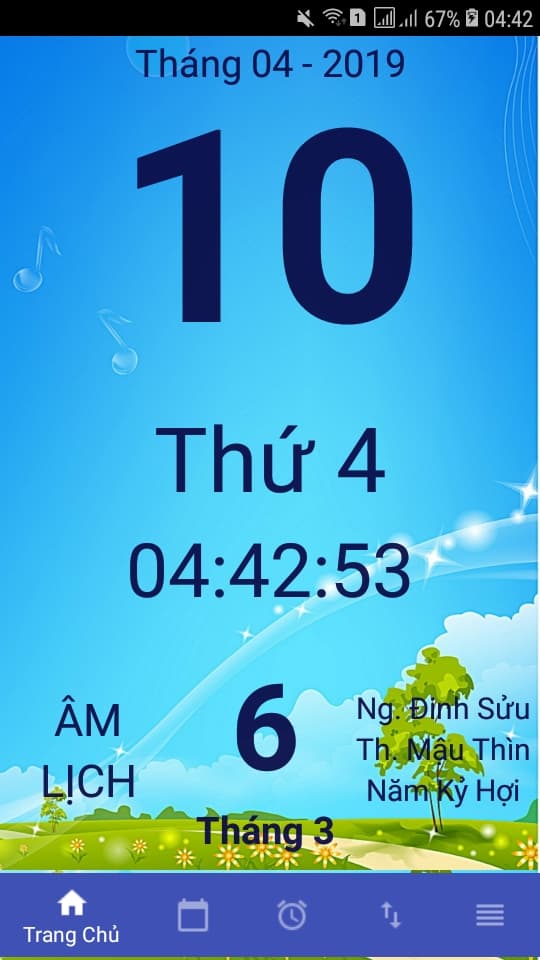
**Tổng quan:** Đây là ứng dụng phần mềm quy mô nhỏ được ích hợp các chức năng dành cho người dùng xem giờ, xem lịch, báo thức và cả đếm ngược chuyển đổi….

### Phạm vi đề tài:

* + - Theo dõi thời gian và chuyển đổi
    - Lập trình trên android studio
    - Hệ điều hành android: 4.0+
    - Ứng dụng lấy API: 24
    - Thiết bị thử ghiệm: CustomPhone, SamSung, HTC 2PS64.
    - Độ phân giải màn hình : 1440 x 2560.

### Chức năng #1: Màn hình giao diện 1 {Màn hình chính} (Nguyễn Thu Tuyết)

* + **Giao diện người dùng**

****

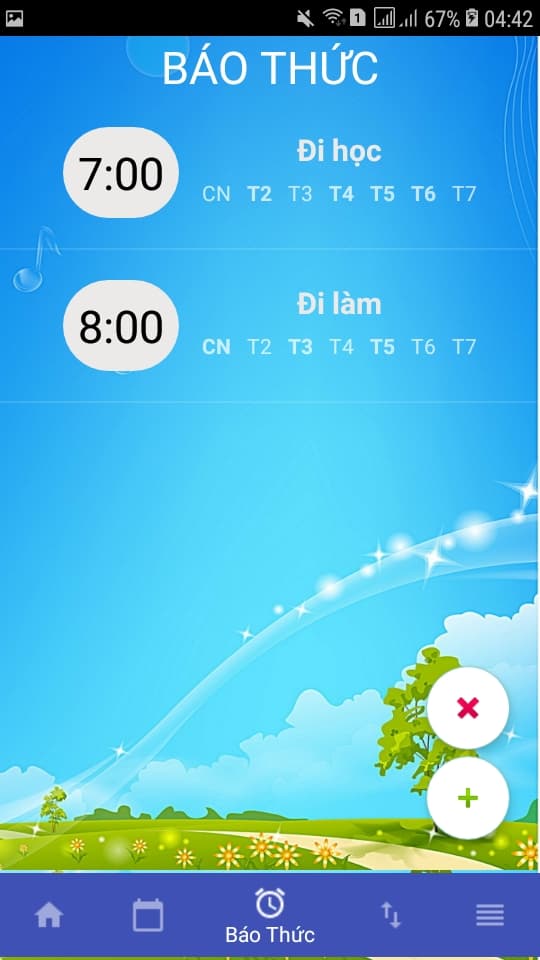
**Hình 1 Màn hình chính**

Màn hình home là mà hình chạy đầu tiên, khi chạy sẽ hiện ra màn hình đồng hồ chính xác. Hiển thị lịch dương và lịch âm và cả can chi con giáp. 🡪 Đồng thời BottomNavigationView cho phép chuyển qua màn hình ứng dụng khác.

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình home – màn hình đồng hồ | Màn hình này gồm các thành phần:   * TextView * FrameLayout * BottomNavigationVew | Thiết kế | Xem được giờ, chạy đúng múi giờ. Xem lịch âm, lịch dương  Hoàn thành:  (%) |
| Button “Home” | * Đây là thành phần trên BottomNavigation * Luôn hiển thị đầu tiên | * Click * Nút luôn hiển thị màu trắng sáng khi mới mở app | Hiển thị màn hình home |
| Button “Lịch Tháng” | * Đây là thành phần trên BottomNavigationView * Chuyển sang giao diện lịch tháng | * Click vào button lịch tháng * Hiển thị màn hình lịch tháng | Hiển thị màn hình lịch tháng |
| Button “Mở Rộng” | * Đây là thành phần trên BottomNavigation   Chuyển sang màn hình mở rộng chứa:   * 2 button: bấm giờ và đếm ngược | * Click * Hiển thị màn hình chứa 2 button bấm giờ và đếm ngược | Hiển thị màn hình mở rộng |
| Button “Bấm Giờ” | * Đây là thành phần trên BottomNavigationView   trong nút mở rộng  Chuyển sang giao diện bấm giờ | * Click vào button mở rộng * Mở ra màn hình chứa button bấm giờ * Hiển thị màn hình với tên đã chọn | Chuyển sang giao diện bấm giờ |
| Button “Đếm Ngược” | * Đây là thành phần trên BottomNavigationView   Chuyển sang giao diện đếm giờ | Click vào button mở rộng  Mở màn hình chứa button đếm ngược   * Hiển thị màn hình với tên đã chọn | Chuyển sang giao diện đếm giờ |
| Button “Báo Thức” | * Đây là thành phần trên BottomNavigationView   Chuyển sang giao diện báo thức | Click vào button báo thức   * Hiển thị màn hình tên đã chọn | Chuyển sang giao diện báo thức |

1. **Chức năng #2: Màn hình giao diện 2(Ngô Thị Ngọc Diễm)**
   * **Giao diện người dùng**

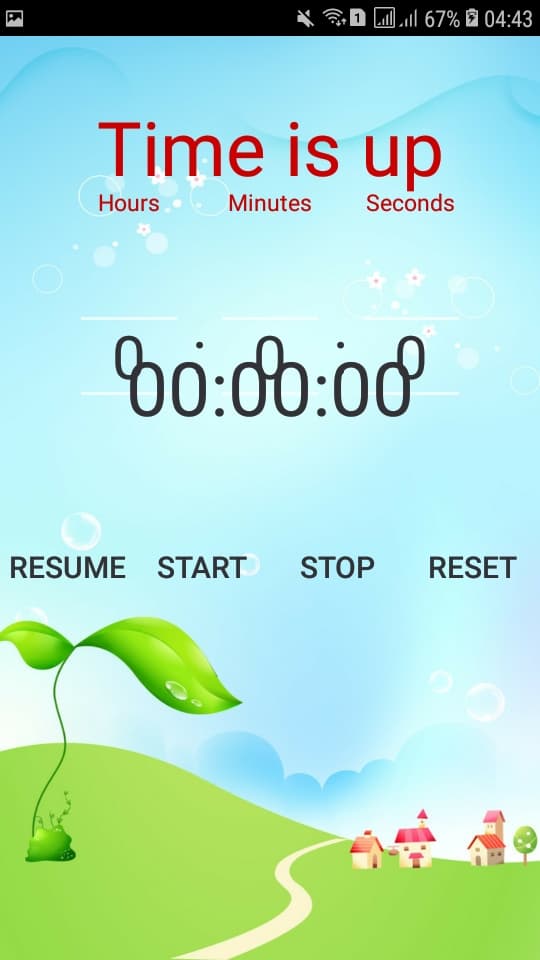
****

**Hình 2 Màn hình báo thức**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình báo thức | Màn hình báo thức có:   * ListView * FloatingActionButton | Thiết kế | Hiển thị màn hình báo thức |
| ListView báo thức | Load ra danh sách lịch đã được tạo ở Activity TaoBaoThuc | Click vào ListViewItem   * Hiện thị màn hình tạo báo thức | Hiển thị màn hình tạo báo thức, có thể chỉnh sửa |
| Button “Xóa” | Khi lick vào nó sẽ Load tới Activity xóa báo thức | Click chọn   * Hiển thị màn hình xóa báo thức | Hiển thị màn hình xóa báo thức, cho phép xóa báo thức vừa tạo |
| Button “Thêm” | Khi click vào nó sẽ Load tới Activity tạo báo thức | Click chọn   * Hiển thị màn hình thêm báo thức | Hiển thị màn hình thêm báo thức, cho phép tạo báo thức mới |

1. **Chức năng #3: Màn hình giao diện 3 (Phạm Minh Hiển)**
   * **Màn hình đếm ngược**

****

**Hình 1 Màn hình đếm ngược**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình đếm ngược | Màn hình đếm ngược gồm có:   * TextView * NumberPicker * Button | Thiết kế | Hiển thị màn hình đếm ngược |
| Khung thời gian | Đếm ngược thời gian bằng những con số NumberPicket | * Click vào bắt đầu * Sẽ tự động thực hiện chức năng đếm ngược | Hiển thị những con số đếm ngược |
| Button “Start” | Nút thực hiện chức năng bắt đầu con số khi người dùng đã cài đặt thời gian đếm ngược | * Click vào bắt đầu * Đồng hồ đếm ngược sẽ thực hiện chức năng đếm ngược | Bắt đầu đếm ngược |
| Button “Stop” | Nút thực hiện việc dừng chương trình | * Click vào nút dừng * Đồng hồ đếm ngược sẽ dừng chương trình đếm ngược | Dừng đếm ngược |
| Button “Resume” | Nút thực hiện việc tiếp tực đồng hồ đếm ngược | * Click vào nút tiếp tục * Đồng hồ sẽ thực hiện tiếp tục chương trình bấm dừng trước đó | Tiếp tục đếm ngược |
| Button “Reset” | Nút thực hiện việc làm mới chương trình | * Click vào nút làm mới * Đồng hồ sẽ thực hiện việc làm mới chương trình | Làm mới chương trình đếm ngược |

1. **Chức năng #4: Màn hình giao diện 4 (Nguyễn Anh Tuấn)**
   * **Màn hình chuyển đổi**

****

**Hình 1 Màn hình chuyển đổi**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình lịch sự | Màn hình chính có:   * 1 textview * 1 listview * 2 Button | Thiết kế | Hoàn thành |

1. **Chức năng #5: Màn hình giao diện 5 (Lê Minh Tân)**
   * **Màn hình bấm giờ**

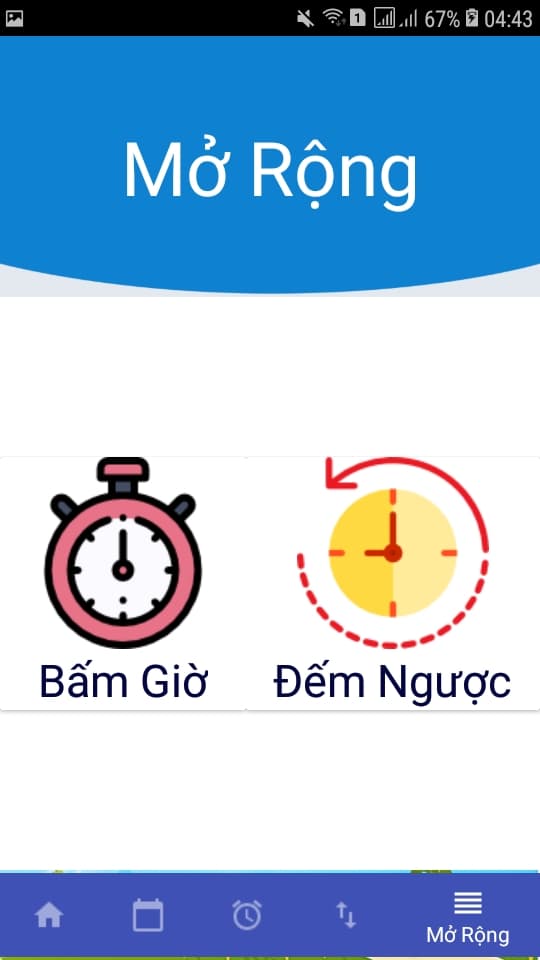
****

**Hình 1 màn hình bấm giờ**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình bấm giờ | Màn hình bấm giờ có:   * TextView * Button * ListView | Thiết kế | Hiển thị màn hình bấm giờ |
| Khung đồng hồ bấm giờ | TextView bấm giờ sẽ chạy chức năng bấm giờ | * Click vào nút Start * Tự động chạy thời gian | Hiển thị giờ được bấm |
| Button “Start” | Thực hiện chức năng chạy chương trình bấm giờ | * Click vào nút Start để chạy chương trình bấm giờ | Chạy chương trình bấm giờ |
| Button “Tách” | Thực hiện việc tách thời gian 🡪 sẽ tách ra thành danh sách | * Click vào nút Tách sẽ tách ra từng thời gian | Tách giờ được bấm |
| Button “Stop” | Thực hiện việc dừng chương trình và xóa bấm giờ củ | * Click vào nút Stop sẽ dừng chương trình | Dừng chương trình bấm giờ |
| Màn hình bấm giờ | Màn hình bấm giờ có:   * TextView * Button * ListView | Thiết kế | Hiển thị màn hình bấm giờ |
|  |  |  |  |

1. **Chức năng #6: Màn hình giao diện 6 (Mở Rộng)**
   * **Màn hình mở rộng**

****

**Hình ảnh mở rộng**

### Yêu cầu chức năng

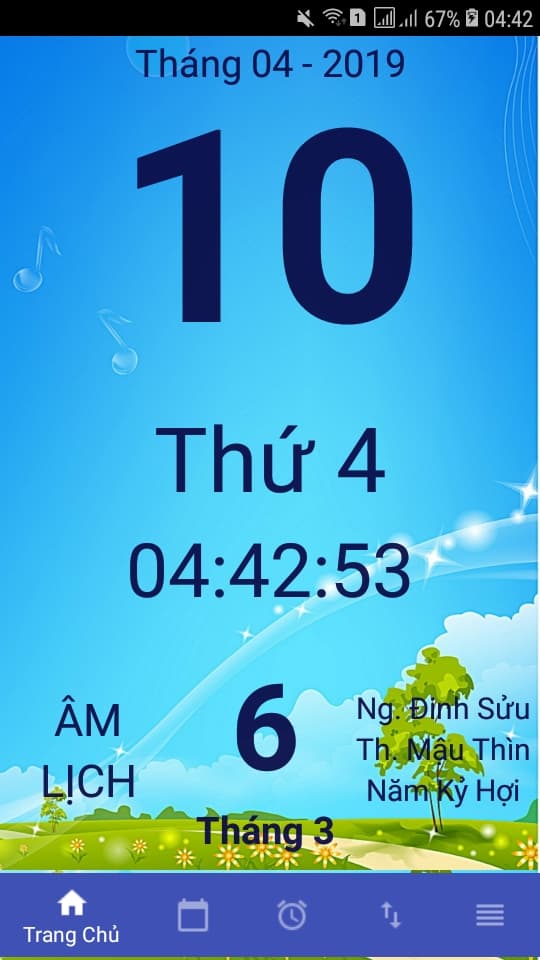
|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình chức năng | Màn hình bấm giờ có:   * TextView * ImageView * CardView | Thiết kế | Hiển thị màn hình chức năng mở rộng |
| CardView Bấm Giờ | Dùng để chuyển qua màn hình bấm giờ | * Click vào * Chuyển qua màn hình | Chuyển sang màn hình bấm giờ |
| CardView Đếm Ngược | Dùng để chuyển qua màn hình đếm ngược | * Click vào * Chuyển qua màn hình đếm ngược | Chuyển sang màn hình đếm ngược |

## Thiết kế hệ thống

**Tổng quan:** Tài liệu dùng để cho nhóm trưởng phân công công việc cho các thành viên giúp phân công công việc 1 cách dễ dàng, bố trí việc làm và phân bổ công việc hợp lý. Mô tả chi tiết công việc cho từng màn hình.

Yêu cầu chức năng:

1. **Màn hình chính**
   * **Giao diện người dùng**

****

**Hình 1 xem giờ**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình home – màn hình đồng hồ | Màn hình này gồm các thành phần:   * TextView * FrameLayout * BottomNavigationVew | Thiết kế | Xem được giờ, chạy đúng múi giờ. Xem lịch âm, lịch dương  Hoàn thành:  (%) |
| Button “Home” | * Đây là thành phần trên BottomNavigation * Luôn hiển thị đầu tiên | * Click * Nút luôn hiển thị màu trắng sáng khi mới mở app | Hiển thị màn hình home |
| Button “Lịch Tháng” | * Đây là thành phần trên BottomNavigationView * Chuyển sang giao diện lịch tháng | * Click vào button lịch tháng * Hiển thị màn hình lịch tháng | Hiển thị màn hình lịch tháng |
| Button “Mở Rộng” | * Đây là thành phần trên BottomNavigation   Chuyển sang màn hình mở rộng chứa:   * 2 button: bấm giờ và đếm ngược | * Click * Hiển thị màn hình chứa 2 button bấm giờ và đếm ngược | Hiển thị màn hình mở rộng |
| Button “Bấm Giờ” | * Đây là thành phần trên BottomNavigationView   trong nút mở rộng  Chuyển sang giao diện bấm giờ | * Click vào button mở rộng * Mở ra màn hình chứa button bấm giờ * Hiển thị màn hình với tên đã chọn | Chuyển sang giao diện bấm giờ |
| Button “Đếm Ngược” | * Đây là thành phần trên BottomNavigationView   Chuyển sang giao diện đếm giờ | Click vào button mở rộng  Mở màn hình chứa button đếm ngược   * Hiển thị màn hình với tên đã chọn | Chuyển sang giao diện đếm giờ |
| Button “Báo Thức” | * Đây là thành phần trên BottomNavigationView   Chuyển sang giao diện báo thức | Click vào button báo thức   * Hiển thị màn hình tên đã chọn | Chuyển sang giao diện báo thức |
| Button “Chuyển Đổi” | * Đây là thành phần trên BottomNavigationView   Chuyển sang giao diện chuyển đổi thời gian và đơn vị | Click vào button chuyển đổi   * Hiển thị màn hình tên đã chọn | Chuyển sang giao diện chuyển đổi thời gian và đơn vị |

**Làm thế nào:** Xem giờ

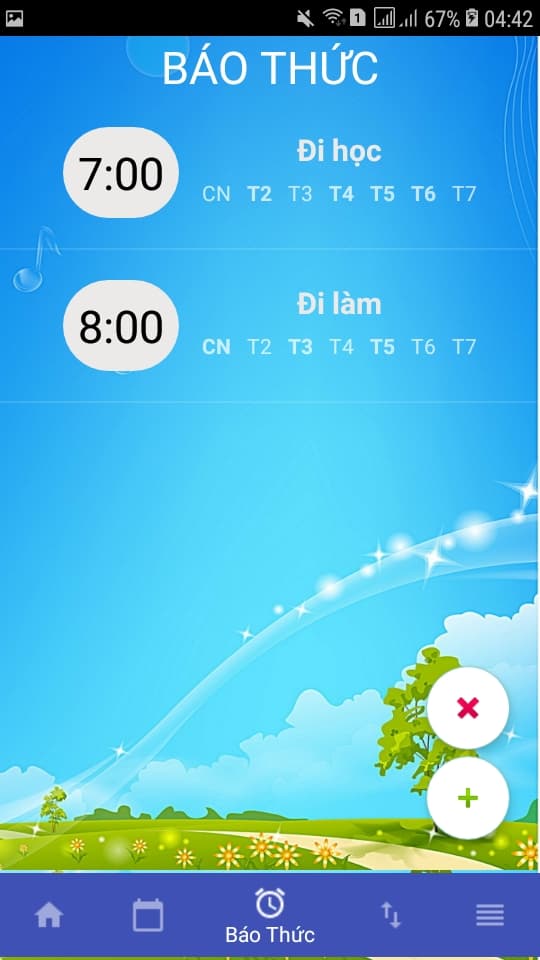
* Chạy ứng dụng
* Chương trình sẽ chạy màn hình xem giờ
* Sau khi người dùng nhấn vào nút sẽ tự động đổi sang màn hình khác

**Làm thế nào:** Màn hình chính

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/hinhnen17"  
 tools:context=".HomeFragment"**>  
  
 *<!--* ***TODO: Update blank fragment layout*** *-->* <**RelativeLayout  
 android:id="@+id/layout\_thu"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"** >  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Tháng 04 - 2019"  
 android:textSize="25dp"  
 android:textColor="@color/color\_main"  
 android:gravity="center"**/>  
  
 <**TextView  
 android:id="@+id/text\_ngay\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="10"  
 android:textStyle="bold"  
 android:textColor="@color/color\_main"  
 android:gravity="center"  
 android:textSize="200dp"**/>  
  
 <**TextView  
 android:id="@+id/text\_thu\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Thứ 2"  
 android:textColor="@color/color\_main"  
 android:layout\_below="@+id/text\_ngay\_main"  
 android:gravity="center"  
 android:textSize="60dp"** />  
  
 <**TextView  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:textSize="50dp"  
 android:textColor="@color/color\_main"  
 android:id="@+id/txtGioPhutGiay"  
 android:layout\_below="@id/text\_thu\_main"** />  
  
 </**RelativeLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/box\_lich"  
 android:layout\_alignParentBottom="true"  
 android:weightSum="3"  
 android:paddingBottom="10dp"  
 android:paddingTop="10dp"** >  
  
 <**RelativeLayout  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:id="@+id/box\_duonglich"  
 android:gravity="center"  
 android:layout\_weight="1"** >  
  
 <**TextView  
 android:id="@+id/text\_lbngayduong"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="ÂM"  
 android:textColor="@color/color\_main"  
 android:gravity="center"  
 android:textSize="30dp"** />  
 <**TextView  
 android:id="@+id/text\_lbthu"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="LỊCH"  
 android:textColor="@color/color\_main"  
 android:gravity="center"  
 android:textSize="30dp"  
 android:layout\_below="@id/text\_lbngayduong"**/>  
 </**RelativeLayout**>  
 *<!--end-->* <**RelativeLayout  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/box\_amlich"  
 android:layout\_weight="1"** >  
  
 <**TextView  
 android:id="@+id/text\_lbngayam"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="6"  
 android:textStyle="bold"  
 android:textColor="@color/color\_main"  
 android:gravity="center"  
 android:textSize="80dp"** />  
 <**TextView  
 android:id="@+id/text\_lbthangam"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Tháng 3"  
 android:textColor="@color/color\_main"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:textSize="@dimen/size\_text\_lbthu"  
 android:layout\_below="@id/text\_lbngayam"**/>  
 </**RelativeLayout**>  
 *<!--end-->* <**RelativeLayout  
 android:id="@+id/box\_congiap"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:layout\_weight="1"** >  
  
 <**TextView  
 android:id="@+id/text\_ngaycongiap"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Ng. Đinh Sửu"  
 android:textColor="@color/color\_main"  
 android:gravity="center"  
 android:textSize="@dimen/size\_ngaythangnamcongiap"**/>  
  
 <**TextView  
 android:id="@+id/text\_thangcongiap"  
 android:layout\_below="@id/text\_ngaycongiap"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Th. Mậu Thìn"  
 android:textColor="@color/color\_main"  
 android:gravity="center"  
 android:textSize="@dimen/size\_ngaythangnamcongiap"**/>  
  
 <**TextView  
 android:id="@+id/text\_namcongiap"  
 android:layout\_below="@id/text\_thangcongiap"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Năm Kỷ Hợi"  
 android:textColor="@color/color\_main"  
 android:gravity="center"  
 android:textSize="@dimen/size\_ngaythangnamcongiap"**/>  
 </**RelativeLayout**>  
  
 </**LinearLayout**>  
  
</**RelativeLayout**>

### Màn hình báo thức

* + **Giao diện người dùng**

****

**Hình 2 Màn hình báo thức**

### Yêu cầu chức năng

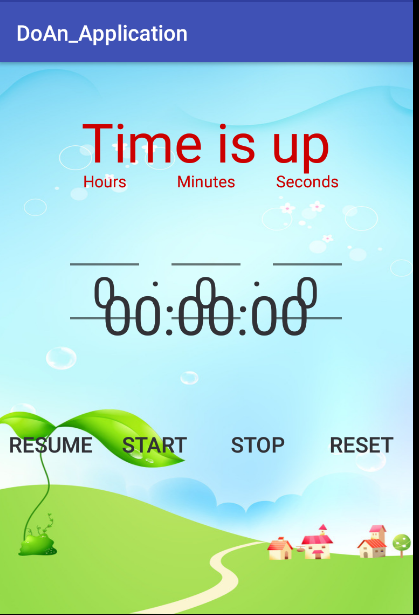
|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình báo thức | Màn hình báo thức có:   * ListView * FloatingActionButton | Thiết kế | Hiển thị màn hình báo thức |
| ListView báo thức | Load ra danh sách lịch đã được tạo ở Activity TaoBaoThuc | Click vào ListViewItem   * Hiện thị màn hình tạo báo thức | Hiển thị màn hình tạo báo thức, có thể chỉnh sửa |
| Button “Xóa” | Khi lick vào nó sẽ Load tới Activity xóa báo thức | Click chọn   * Hiển thị màn hình xóa báo thức | Hiển thị màn hình xóa báo thức, cho phép xóa báo thức vừa tạo |
| Button “Thêm” | Khi click vào nó sẽ Load tới Activity tạo báo thức | Click chọn   * Hiển thị màn hình thêm báo thức | Hiển thị màn hình thêm báo thức, cho phép tạo báo thức mới |

**Làm thế nào:**  màn hình báo thức

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/hinhnen21"  
 tools:context=".BaoThucFragment"**>  
  
 *<!--* ***TODO: Update blank fragment layout*** *-->* <**TextView  
 android:textSize="@dimen/text\_size\_tieude"  
 android:textColor="#fff"  
 android:id="@+id/lbl\_ten"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/text\_lbl\_ten"  
 android:gravity="center"** />  
  
 <**ListView  
 android:id="@+id/listView"  
 android:layout\_height="match\_parent"  
 android:layout\_width="match\_parent"  
 android:layout\_below="@id/lbl\_ten"** />  
  
 *<!--icon xoa bao thuc-->* <**android.support.design.widget.FloatingActionButton  
 android:id="@+id/btnXoaBaoThuc"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_above="@id/btnThemBaoThuc"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginBottom="@dimen/mg\_main\_bottom\_delete"  
 android:layout\_marginRight="@dimen/mg\_main\_right"  
 android:src="@android:drawable/ic\_delete"  
 android:backgroundTint="@color/colorWhite"** />  
  
 *<!--icon tao lich bao thuc moi-->* <**android.support.design.widget.FloatingActionButton  
 android:id="@+id/btnThemBaoThuc"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginRight="@dimen/mg\_main\_right"  
 android:layout\_marginBottom="@dimen/mg\_main\_bottom"  
 android:src="@android:drawable/ic\_input\_add"  
 android:backgroundTint="@color/colorWhite"** />  
  
</**RelativeLayout**>

### Màn hình đếm ngược

* + **Giao diện người dùng**



**Hình 3 Màn hình đếm ngược**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình đếm ngược | Màn hình đếm ngược gồm có:   * TextView * NumberPicker * Button | Thiết kế | Hiển thị màn hình đếm ngược |
| Khung thời gian | Đếm ngược thời gian bằng những con số NumberPicket | * Click vào bắt đầu * Sẽ tự động thực hiện chức năng đếm ngược | Hiển thị những con số đếm ngược |
| Button “Start” | Nút thực hiện chức năng bắt đầu con số khi người dùng đã cài đặt thời gian đếm ngược | * Click vào bắt đầu * Đồng hồ đếm ngược sẽ thực hiện chức năng đếm ngược | Bắt đầu đếm ngược |
| Button “Stop” | Nút thực hiện việc dừng chương trình | * Click vào nút dừng * Đồng hồ đếm ngược sẽ dừng chương trình đếm ngược | Dừng đếm ngược |
| Button “Resume” | Nút thực hiện việc tiếp tực đồng hồ đếm ngược | * Click vào nút tiếp tục * Đồng hồ sẽ thực hiện tiếp tục chương trình bấm dừng trước đó | Tiếp tục đếm ngược |
| Button “Reset” | Nút thực hiện việc làm mới chương trình | * Click vào nút làm mới * Đồng hồ sẽ thực hiện việc làm mới chương trình | Làm mới chương trình đếm ngược |

**Làm thế nào:**  Màn hình đếm ngược

<**RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:animateLayoutChanges="true"  
 android:background="@drawable/hinhnen20"  
 tools:context=".DemNguocActivity"**>  
  
  
 <**LinearLayout  
 android:id="@+id/linearLayout\_timer\_message\_container"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="parent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:gravity="center"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/TimerMessageStyle"  
 android:text="@string/timeIsUP"**/>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:id="@+id/linearLayout\_timer\_input\_number\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="parent"  
 android:layout\_marginTop="100dp"  
 android:gravity="center"  
 android:orientation="horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginRight="10dp"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="2dp"  
 android:layout\_gravity="center"  
 android:theme="@style/NumberPickerTopLabel"  
 android:text="@string/hours"**/>  
 <**NumberPicker  
 android:id="@+id/number\_picker\_hours"  
 android:layout\_width="wrap\_content"  
 android:descendantFocusability="blocksDescendants"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_gravity="center"  
 android:layout\_centerHorizontal="true"  
 android:theme="@style/NumberPickerText"** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_marginTop="8dp"  
 android:layout\_width="20dp"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/NumberPickersSeparator"  
 android:layout\_gravity="center\_vertical"  
 android:text=":"** />  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginRight="10dp"**>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="2dp"  
 android:layout\_gravity="center"  
 android:theme="@style/NumberPickerTopLabel"  
 android:text="@string/minutes"**/>  
 <**NumberPicker  
 android:id="@+id/number\_picker\_minutes"  
 android:layout\_width="wrap\_content"  
 android:descendantFocusability="blocksDescendants"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_gravity="center"  
 android:layout\_centerHorizontal="true"  
 android:theme="@style/NumberPickerText"** />  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_marginTop="8dp"  
 android:layout\_width="20dp"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/NumberPickersSeparator"  
 android:layout\_gravity="center\_vertical"  
 android:text=":"** />  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginBottom="2dp"  
 android:theme="@style/NumberPickerTopLabel"  
 android:text="@string/seconds"**/>  
 <**NumberPicker  
 android:id="@+id/number\_picker\_seconds"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:descendantFocusability="blocksDescendants"  
 android:layout\_gravity="center"  
 android:layout\_centerHorizontal="true"  
 android:theme="@style/NumberPickerText"** />  
 </**LinearLayout**>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:id="@+id/linearLayout\_timer\_container"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="parent"  
 android:layout\_marginTop="200dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:orientation="horizontal"**>  
  
  
 <**TextView  
 android:id="@+id/textView\_timer\_h"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="00:"  
 android:theme="@style/NumberTextViewsStyle"**/>  
  
  
 <**TextView  
 android:id="@+id/textView\_timer\_m"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="00:"  
 android:theme="@style/NumberTextViewsStyle"**/>  
  
 <**TextView  
 android:id="@+id/textView\_timer\_s"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="00"  
 android:theme="@style/NumberTextViewsStyle"**/>  
  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:animateLayoutChanges="true"  
 android:layout\_marginTop="330dp"**>  
  
 <**Button  
 android:id="@+id/button\_resume\_timer"  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/ButtonClockController"  
 android:background="@android:color/transparent"  
 android:text="@string/resume"** />  
  
 <**Button  
 android:id="@+id/button\_start\_timer"  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/ButtonClockController"  
 android:background="@android:color/transparent"  
 android:text="@string/start\_timer"** />  
  
 <**Button  
 android:id="@+id/button\_stop\_timer"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:theme="@style/ButtonClockController"  
 android:background="@android:color/transparent"  
 android:text="@string/stop"** />  
  
 <**Button  
 android:id="@+id/button\_reset\_timer"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:theme="@style/ButtonClockController"  
 android:background="@android:color/transparent"  
 android:text="@string/reset"** />  
  
 </**LinearLayout**>  
</**RelativeLayout**>

### Màn hình bấm giờ

* + **Giao diện người dùng**

****

**Hình 4 màn hình chuyển đổi**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình bấm giờ | Màn hình bấm giờ có:   * TextView * Button * ListView | Thiết kế | Hiển thị màn hình bấm giờ |
| Khung đồng hồ bấm giờ | TextView bấm giờ sẽ chạy chức năng bấm giờ | * Click vào nút Start * Tự động chạy thời gian | Hiển thị giờ được bấm |
| Button “Start” | Thực hiện chức năng chạy chương trình bấm giờ | * Click vào nút Start để chạy chương trình bấm giờ | Chạy chương trình bấm giờ |
| Button “Tách” | Thực hiện việc tách thời gian 🡪 sẽ tách ra thành danh sách | * Click vào nút Tách sẽ tách ra từng thời gian | Tách giờ được bấm |
| Button “Stop” | Thực hiện việc dừng chương trình và xóa bấm giờ củ | * Click vào nút Stop sẽ dừng chương trình | Dừng chương trình bấm giờ |

**Làm thế nào**: Màn hình bấm giờ

<**LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/hinhnen18"  
 tools:context=".BamGioActivity"**>  
  
 <**TextView  
 android:layout\_gravity="center"  
 android:text="00:00:00"  
 android:textSize="60dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="120dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_alignParentTop="true"  
 android:id="@+id/txtTimer"** />  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="4"  
 android:orientation="horizontal"**>  
  
 <**Button  
 android:id="@+id/btnStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:layout\_alignTop="@+id/txtTimer"  
 android:layout\_alignParentLeft="true"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="120dp"  
 android:text="START"** />  
  
 <**Button  
 android:id="@+id/btnPause"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:layout\_alignTop="@+id/txtTimer"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="120dp"  
 android:text="TÁCH"** />  
  
 <**Button  
 android:id="@+id/btnStop"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:layout\_alignTop="@+id/txtTimer"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginTop="120dp"  
 android:layout\_marginRight="20dp"  
 android:text="STOP"** />  
  
   
  
 </**LinearLayout**>  
   
 <**LinearLayout  
 android:id="@+id/simple"  
 android:layout\_weight="4"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"**>  
  
 <**ListView  
 android:id="@+id/lvTime"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
 </**LinearLayout**>  
</**LinearLayout**>

1. **Màn hình chuyển đổi**
   * **Giao diện người dùng**

****

**Hình 5 Màn hình chuyển đổi**

### Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình lịch sự | Màn hình chính có:   * 1 textview * 1 listview * 2 Button | Thiết kế | Hoàn thành |

**Làm thế nào:** Màn hình lịch sử

# CHƯƠNG 3. CÀI ĐẶT

## Màn hình home

**public class** HomeFragment **extends** Fragment {  
  
 **private** TextView **txtGioPhutGiay**;  
  
 **private** Handler **handler**;  
 **private** Runnable **runnable**;  
  
 **public** HomeFragment() {  
 *// Required empty public constructor* }  
  
 @Override  
 **public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
  
 **txtGioPhutGiay** = (TextView) getView().findViewById(R.id.***txtGioPhutGiay***);  
 **handler** = **new** Handler();  
  
 **runnable** = **new** Runnable() {  
 @Override  
 **public void** run() {  
 showDongHo();  
 **handler**.postDelayed(**runnable**, 300);  
 }  
 };  
  
 **handler**.postDelayed(**runnable**, 300);  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment* **return** inflater.inflate(R.layout.***fragment\_home***, container, **false**);  
 }  
  
 **public void** showDongHo() {  
  
 Calendar calendar = Calendar.*getInstance*();  
 SimpleDateFormat mdformat = **new** SimpleDateFormat(**"HH:mm:ss"**);  
 *//Log.e("time: ", mdformat.format(calendar.getTime()));* **txtGioPhutGiay**.setText(mdformat.format(calendar.getTime()));  
 }  
}

## Màn hình xem giờ

1. **public class** LichThangFragment **extends** Fragment {  
     
    **public static final** String ***EXTRA\_DATE*** = **"com.example.application.example.EXTRA\_DATE"**;  
    **public static final** String ***EXTRA\_NUMBER*** = **"com.example.application.example.EXTRA\_NUMBER"**;  
     
    **private static final** String ***TAG*** = **"LichThangFragment"**;  
     
    **private** Calendar **selectedDate**;  
     
    **private** TextView **mDisplayDate**;  
     
    LinearLayout **linearLayout**;  
     
    **private** SupportedDatePickerDialog.OnDateSetListener **mDateSetListener**;  
     
     
     
     
    **public** LichThangFragment() {  
    *// Required empty public constructor* }  
     
    @Override  
    **public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  
    **super**.onActivityCreated(savedInstanceState);  
     
    initial();  
     
   *// Log.e("01/01/2019: ", Utilities.getInstance().convertSolar2Lunar(1, 1, 2019, 7.0) + "");  
   // Log.e("25/02/2019: ", Utilities.getInstance().convertSolar2Lunar(25, 2, 2019, 7.0) + "");  
   // Log.e("10/03/2019: ", Utilities.getInstance().convertSolar2Lunar(10, 3, 2020, 7.0) + "");  
   //  
   // Log.e("15/05/2019: ", Utilities.getInstance().convertSolar2Lunar(15, 5, 2020, 7.0) + "");  
   // Log.e("16/05/2019: ", Utilities.getInstance().convertSolar2Lunar(16, 5, 2020, 7.0) + "");  
   // Log.e("17/05/2019: ", Utilities.getInstance().convertSolar2Lunar(17, 5, 2020, 7.0) + "");  
   // Log.e("18/05/2019: ", Utilities.getInstance().convertSolar2Lunar(18, 5, 2020, 7.0) + "");  
   // Log.e("19/05/2019: ", Utilities.getInstance().convertSolar2Lunar(19, 5, 2020, 7.0) + "");  
   // Log.e("20/05/2019: ", Utilities.getInstance().convertSolar2Lunar(20, 5, 2020, 7.0) + "");  
   // Log.e("21/05/2019: ", Utilities.getInstance().convertSolar2Lunar(21, 5, 2020, 7.0) + "");  
   // Log.e("22/05/2019: ", Utilities.getInstance().convertSolar2Lunar(22, 5, 2020, 7.0) + "");  
   // Log.e("23/05/2019: ", Utilities.getInstance().convertSolar2Lunar(23, 5, 2020, 7.0) + "");  
   // Log.e("1/06/2019: ", Utilities.getInstance().convertSolar2Lunar(1, 6, 2020, 7.0) + "");  
   // Log.e("1/07/2019: ", Utilities.getInstance().convertSolar2Lunar(1, 7, 2020, 7.0) + "");  
   // Log.e("1/08/2019: ", Utilities.getInstance().convertSolar2Lunar(1, 8, 2020, 7.0) + "");  
   // Log.e("1/09/2019: ", Utilities.getInstance().convertSolar2Lunar(1, 9, 2020, 7.0) + "");  
   // Log.e("1/10/2019: ", Utilities.getInstance().convertSolar2Lunar(1, 10, 2020, 7.0) + "");  
   // Log.e("02/12/2019: ", Utilities.getInstance().convertSolar2Lunar(2, 12, 2020, 7.0) + "");  
   // Log.e("01/11/2019: ", Utilities.getInstance().convertSolar2Lunar(1, 11, 2020, 7.0) + "");* ChinaCalendar date = **new** ChinaCalendar(10, 2, 2019, 7);  
    Date result = date.ConVertToLunar();  
    Log.*e*(**"01/01/2019"**, result.getDate() + **"/"** + result.getMonth() + **"/"** + result.getYear());  
     
    date = **new** ChinaCalendar(1, 5, 2019, 7);  
    result = date.ConVertToLunar();  
    Log.*e*(**"01/01/2019"**, result.getDate() + **"/"** + result.getMonth() + **"/"** + result.getYear());  
     
    date = **new** ChinaCalendar(15, 9, 2019, 7);  
    result = date.ConVertToLunar();  
    Log.*e*(**"01/01/2019"**, result.getDate() + **"/"** + result.getMonth() + **"/"** + result.getYear());  
     
    date = **new** ChinaCalendar(2, 12, 1996, 7);  
    result = date.ConVertToLunar();  
    Log.*e*(**"01/01/2019"**, result.getDate() + **"/"** + result.getMonth() + **"/"** + result.getYear());  
     
    date = **new** ChinaCalendar(15, 10, 2020, 7);  
    result = date.ConVertToLunar();  
    Log.*e*(**"01/01/2019"**, result.getDate() + **"/"** + result.getMonth() + **"/"** + result.getYear());  
     
    }  
     
    @Override  
    **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
    Bundle savedInstanceState) {  
    *// Inflate the layout for this fragment* **return** inflater.inflate(R.layout.***fragment\_lich\_thang***, container, **false**);  
    }  
     
    **public void** initial() {  
     
    **linearLayout** = (LinearLayout) getView().findViewById(R.id.***othu2***);  
     
    **linearLayout**.setOnClickListener(**new** View.OnClickListener() {  
    @Override  
    **public void** onClick(View v) {  
     
    openActivity2((LinearLayout) v);  
    }  
    });  
     
    **mDisplayDate** = (TextView) getView().findViewById(R.id.***tvDate***);  
     
    **mDisplayDate**.setOnClickListener(**new** View.OnClickListener() {  
    @Override  
    **public void** onClick(View v) {  
    Calendar calendar = Calendar.*getInstance*();  
    **int** year = calendar.get(Calendar.***YEAR***);  
    **int** month = calendar.get(Calendar.***MONTH***);  
    **int** day = calendar.get(Calendar.***DAY\_OF\_MONTH***);  
     
    **new** SupportedDatePickerDialog(LichThangFragment.**this**.getActivity(), R.style.***SpinnerDatePickerDialogTheme***, **mDateSetListener**, year, month, day).show();  
    }  
    });  
     
    **mDateSetListener** = **new** SupportedDatePickerDialog.OnDateSetListener() {  
    @Override  
    **public void** onDateSet(DatePicker view, **int** year, **int** month, **int** dayOfMonth) {  
    month = month + 1; *// vi thang bat dau tu 0* setDisplayDate(month, year);  
    }  
    };  
     
    **selectedDate** = Calendar.*getInstance*();  
    setDisplayDate(**selectedDate**.get(Calendar.***MONTH***) + 1, **selectedDate**.get(Calendar.***YEAR***));  
    }  
     
    **public void** setDisplayDate(**int** month, **int** year) {  
    String date = **"Tháng"** + **" "** + month + **" "** + **"-"** + **" "** + year;  
    **mDisplayDate**.setText(date);  
    }  
     
    **public void** openActivity2(LinearLayout view){  
     
    TextView tvNgayDuong = (TextView) view.getChildAt(0);  
    TextView tvNgayAm = (TextView) view.getChildAt(1);  
     
    **int** ngayDuong = Integer.*valueOf*(tvNgayDuong.getText().toString());  
    **int** ngayAm = Integer.*valueOf*(tvNgayAm.getText().toString());  
     
    **selectedDate**.set(Calendar.***DAY\_OF\_MONTH***, ngayDuong);  
     
   *// TextView textView1 = (TextView) getView().findViewById(R.id.textView1);  
   // String text = textView1.getText().toString();  
   //  
   // TextView textView2 = (TextView) getView().findViewById(R.id.textView2);  
   // String text2 = textView2.getText().toString();* Intent intent = **new** Intent(**this**.getActivity(), ActivityLichThangAm.**class**);  
    intent.putExtra(***EXTRA\_DATE***, **selectedDate**);  
     
    startActivity(intent);  
    }  
     
    **public void** HienThiLichAm(**int** so){  
     
     
    }  
   }

## Màn hình báo thức

**public class** BaoThucFragment **extends** Fragment {  
  
 **public final static** String ***EXTRA\_BAOTHUCMOI*** = **"BAOTHUCMOI"**;  
 **public final static** String ***EXTRA\_SUABAOTHUC*** = **"SUABAOTHUC"**;  
  
 **public static final int *REQUEST\_CODE\_TAOBAOTHUC*** = 1001;  
 **public static final int *REQUEST\_CODE\_SUABAOTHUC*** = 1002;  
  
 *// button hoan tat va luu bao thuc vao listview* Button **txt\_giohen**;  
  
 *// arraylist* ArrayList<BaoThuc> **arr\_lstBaoThuc**;  
  
 *// adapter* BaoThucAdapter **adapterBaoThuc**;  
  
 *// listview* **private** ListView **mListView**;  
  
 *// them bao thuc* **private** FloatingActionButton **btnThemBaoThuc**;  
  
 *// xoa bao thuc* **private** FloatingActionButton **btnXoaBaoThuc**;  
  
 **public** BaoThucFragment() {  
 *// Required empty public constructor* }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 Log.*e*(**"tag: "**, **"Trong onCreateView"**);  
 *// Inflate the layout for this fragment* **return** inflater.inflate(R.layout.***fragment\_bao\_thuc***, container, **false**);  
 }  
  
 @Override  
 **public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
  
 Log.*e*(**"tag: "**, **"Sau onActivityCreated"**);  
 **mListView** = (ListView) getView().findViewById(R.id.***listView\_bt***);  
  
*// // tạo thông tin báo thức (code cứng)* BaoThuc baothuc1 = **new** BaoThuc(1,**"7:00"**, **"Đi học"**, **null**, **false**, **true**, **false**, **true**, **true**, **true**, **false**);  
 BaoThuc baothuc2 = **new** BaoThuc(2,**"8:00"**, **"Đi làm"**, **null**, **true**, **false**, **true**, **false**, **true**, **false**, **false**);  
  
 **arr\_lstBaoThuc** = **new** ArrayList<BaoThuc>();  
  
 **arr\_lstBaoThuc**.add(baothuc1);  
 **arr\_lstBaoThuc**.add(baothuc2);  
  
 **adapterBaoThuc** = **new** BaoThucAdapter(**this**.getActivity(), R.layout.***baothuc\_item***, **arr\_lstBaoThuc**);  
  
 **mListView**.setAdapter(**adapterBaoThuc**);  
  
 initial();  
 }  
  
 @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 Log.*e*(**"tag: "**, **"Activity Result on fragment"**);  
  
 **if** (requestCode == ***REQUEST\_CODE\_TAOBAOTHUC***)  
 {  
 ((MainActivity)getActivity()).setSelectedItemMenu(R.id.***menu\_bao\_thuc***);  
  
 BaoThuc result = (BaoThuc) data.getSerializableExtra(***EXTRA\_BAOTHUCMOI***);  
  
 **if** (result != **null**)  
 {  
 Log.*e*(**"tag: "**, **"Add new baothuc has been successful"**);  
 result.setId(**arr\_lstBaoThuc**.size() + 1);  
 **arr\_lstBaoThuc**.add(result);  
 **adapterBaoThuc**.notifyDataSetChanged();  
 }  
 }  
 **else if** (requestCode == ***REQUEST\_CODE\_SUABAOTHUC***)  
 {  
 **if** (resultCode == Activity.***RESULT\_OK***)  
 {  
 ((MainActivity)getActivity()).setSelectedItemMenu(R.id.***menu\_bao\_thuc***);  
  
 BaoThuc result = (BaoThuc) data.getSerializableExtra(***EXTRA\_SUABAOTHUC***);  
 Log.*e*(**"Sua bao thuc: "**, result.getId() + **""**);  
  
 **if** (result != **null**)  
 {  
 Log.*e*(**"arr\_lstBaoThuc.size(): "**, **arr\_lstBaoThuc**.size() + **""**);  
  
 **for** (**int** i = 0; i < **arr\_lstBaoThuc**.size(); i++)  
 {  
 **if** (**arr\_lstBaoThuc**.get(i).getId() == result.getId())  
 {  
 Log.*e*(**"get(i): "**, result.getId() + **""**);  
  
 **arr\_lstBaoThuc**.get(i).setsGioHen(result.getsGioHen());  
 **arr\_lstBaoThuc**.get(i).setsMoTa(result.getsMoTa());  
  
 **arr\_lstBaoThuc**.get(i).setChuNhat(result.isChuNhat());  
 **arr\_lstBaoThuc**.get(i).setThuHai(result.isThuHai());  
 **arr\_lstBaoThuc**.get(i).setThuBa(result.isThuBa());  
 **arr\_lstBaoThuc**.get(i).setThuTu(result.isThuTu());  
 **arr\_lstBaoThuc**.get(i).setThuNam(result.isThuNam());  
 **arr\_lstBaoThuc**.get(i).setThuSau(result.isThuSau());  
 **arr\_lstBaoThuc**.get(i).setThuBay(result.isThuBay());  
  
 **break**;  
 }  
 }  
  
 **adapterBaoThuc**.notifyDataSetChanged();  
 }  
 }  
  
 }  
  
 **super**.onActivityResult(requestCode, resultCode, data);  
 }  
  
 **public void** initial() {  
 *// xu ly phan them bao thuc click vao + -> hien thi activity TaoBaoThuc* **btnThemBaoThuc** = (FloatingActionButton) getView().findViewById(R.id.***btnThemBaoThuc***);  
  
 *// khi click vao icon + -> load trag them bao thuc* **btnThemBaoThuc**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(getActivity(), TaoBaoThucActivity.**class**);  
 TaoBaoThucActivity activity = **new** TaoBaoThucActivity();  
  
 startActivityForResult(intent, ***REQUEST\_CODE\_TAOBAOTHUC***);  
 }  
 });  
  
 *// xu ly phan xoa bao thuc -> click vao X -> hien thi activity XoaBaoThuc* **btnXoaBaoThuc** = (FloatingActionButton) getView().findViewById(R.id.***btnXoaBaoThuc***);  
  
 *// khi click vao icon + -> load trag xoa bao thuc* **btnXoaBaoThuc**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
  
 Intent intent = **new** Intent(getActivity(), XoaBaoThucActivity.**class**);  
 XoaBaoThucActivity activity = **new** XoaBaoThucActivity();  
  
 *//thực hiện chuyển list bao thuc sang activity XaoBaoThuc bằng cách passing dữ liệu dùng putExtras* Bundle bundle = **new** Bundle();  
 bundle.putSerializable(**"BaoThuc"**, **arr\_lstBaoThuc**);  
 intent.putExtras(bundle);  
  
 startActivity(intent);  
 }  
 });  
  
 *// xử lý phần click vào listview nó hiện ra activity tạo báo thức* **mListView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Log.*e*(**"tag: "**, **"onItemSelected"**);  
 Intent i = **new** Intent(BaoThucFragment.**this**.getContext(), TaoBaoThucActivity.**class**);  
 i.putExtra(***EXTRA\_SUABAOTHUC***, (BaoThuc)**mListView**.getAdapter().getItem(position));  
 startActivityForResult(i, ***REQUEST\_CODE\_SUABAOTHUC***);  
 }  
 });  
  
 *// xử lý sự kiện click vào button áp dụng khi sửa đổi nội dung báo thức và lưu vào listview màn hình chính* }  
  
 *// set time sau khi click vao button click vào đây để chọn giờ* **public void** setTime(View view){  
 Calendar calendar = Calendar.*getInstance*();  
 **int** hour = calendar.get(Calendar.***HOUR***);  
 **int** minute = calendar.get(Calendar.***MINUTE***);  
 **int** am\_pm = calendar.get(Calendar.***AM\_PM***);  
  
 TimePickerDialog timePickerDialog;  
  
 *// ham xu ly sau khi chon gio* timePickerDialog = **new** TimePickerDialog(BaoThucFragment.**this**.getActivity(), **new** TimePickerDialog.OnTimeSetListener() {  
 @Override  
 **public void** onTimeSet(TimePicker view, **int** hourOfDay, **int** minute) {  
 **txt\_giohen**.setText(hourOfDay+**":"**+minute);  
 }  
 }, hour, minute, **true**);  
 timePickerDialog.show();  
 }  
  
}

## Màn hình đếm ngược

**public class** DemNguocActivity **extends** AppCompatActivity **implements** View.OnClickListener {  
  
 **private static final byte *STATE\_INITIAL*** = 0;  
 **private static final byte *STATE\_START*** = 1;  
 **private static final byte *STATE\_STOP*** = 2;  
 **private static final byte *STATE\_FINISHED*** = 3;  
  
 **private long timerDuration** = 0;  
 **private byte timerState** = ***STATE\_INITIAL***;  
  
 **private static final byte *FRAGMENT\_STATE\_ON\_RESUME*** = 1;  
 **private static final byte *FRAGMENT\_STATE\_ON\_PAUSE*** = 2;  
 **private byte fragmentState**;  
  
 Button **buttonStartTimer**;  
 Button **buttonStopTimer**;  
 Button **buttonResumeTimer**;  
 Button **buttonResetTimer**;  
  
 TextView **textViewTimerSeconds**;  
 TextView **textViewTimerMinutes**;  
 TextView **textViewTimerHours**;  
  
 CountDownTimer **countDownTimer**;  
  
 NumberPicker **numberPickerSeconds**;  
 NumberPicker **numberPickerMinutes**;  
 NumberPicker **numberPickerHours**;  
  
 LinearLayout **linearLayoutTimerContainer**;  
 LinearLayout **linearLayoutTimerInputNumberContainer**;  
 LinearLayout **linearLayoutTimerMessageContainer**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_dem\_nguoc***);  
  
  
  
 }  
  
 @Override  
 **public void** onClick(View view) {  
 **switch** (view.getId())  
 {  
 **case** R.id.***button\_start\_timer***:  
 **if** (**numberPickerSeconds**.getValue() == 0 && **numberPickerMinutes**.getValue() == 0 && **numberPickerHours**.getValue() == 0)  
 {  
 Toast.*makeText*(getBaseContext(), getString(R.string.***wrongInputForTimer***), Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 configStartState();  
 calculateInputTime();  
 countDownTimerCombine();  
 }  
 **break**;  
  
 **case** R.id.***button\_stop\_timer***:  
 **countDownTimer**.cancel();  
 configStopState();  
 **break**;  
  
 **case** R.id.***button\_reset\_timer***:  
 **countDownTimer**.cancel();  
 configInitialState();  
 **break**;  
  
 **case** R.id.***button\_resume\_timer***:  
 countDownTimerCombine();  
 configStartState();  
 **break**;  
 }  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
  
 **fragmentState** = ***FRAGMENT\_STATE\_ON\_PAUSE***;  
 }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
  
 **fragmentState** = ***FRAGMENT\_STATE\_ON\_RESUME***;  
 }  
  
 **private void** calculateInputTime(){  
 **long** seconds = **numberPickerSeconds**.getValue();  
 **long** minutes = **numberPickerMinutes**.getValue();  
 **long** hours = **numberPickerHours**.getValue();  
 **timerDuration** = (seconds + (60 \* minutes) + (60 \* 60 \* hours)) \* 1000;  
 }  
  
 **private void** configure(){  
 findViewByIds();  
 setOnClickListeners();  
 setInputNumbersValues();  
 goToSuitableState();  
 }  
  
 **private void** findViewByIds(){  
 **linearLayoutTimerContainer** = findViewById(R.id.***linearLayout\_timer\_container***);  
 **linearLayoutTimerInputNumberContainer** = findViewById(R.id.***linearLayout\_timer\_input\_number\_container***);  
 **linearLayoutTimerMessageContainer** = findViewById(R.id.***linearLayout\_timer\_message\_container***);  
  
 **numberPickerSeconds** = findViewById(R.id.***number\_picker\_seconds***);  
 **numberPickerMinutes** = findViewById(R.id.***number\_picker\_minutes***);  
 **numberPickerHours** = findViewById(R.id.***number\_picker\_hours***);  
  
 **textViewTimerSeconds** = findViewById(R.id.***textView\_timer\_s***);  
 **textViewTimerMinutes** = findViewById(R.id.***textView\_timer\_m***);  
 **textViewTimerHours** = findViewById(R.id.***textView\_timer\_h***);  
  
 **buttonStartTimer** = findViewById(R.id.***button\_start\_timer***);  
 **buttonStopTimer** = findViewById(R.id.***button\_stop\_timer***);  
 **buttonResumeTimer** = findViewById(R.id.***button\_resume\_timer***);  
 **buttonResetTimer** = findViewById(R.id.***button\_reset\_timer***);  
 }  
  
 **private void** setOnClickListeners(){  
 **buttonStartTimer**.setOnClickListener(**this**);  
 **buttonStopTimer**.setOnClickListener(**this**);  
 **buttonResumeTimer**.setOnClickListener(**this**);  
 **buttonResetTimer**.setOnClickListener(**this**);  
 }  
  
 **private void** setInputNumbersValues(){  
 **final** String[] secondsNumbers = **new** String[61];  
 **for** (**int** i = 0; i < secondsNumbers.**length**; i++)  
 secondsNumbers[i] = String.*format*(**"%02d"**, i);  
  
 String[] hoursNumbers = **new** String[100];  
 **for** (**int** i = 0; i < hoursNumbers.**length**; i++)  
 hoursNumbers[i] = String.*format*(**"%02d"**, i);  
  
 **numberPickerSeconds**.setMaxValue(60);  
 **numberPickerSeconds**.setMinValue(0);  
 **numberPickerSeconds**.setDisplayedValues(secondsNumbers);  
  
 **numberPickerMinutes**.setMaxValue(60);  
 **numberPickerMinutes**.setMinValue(0);  
 **numberPickerMinutes**.setDisplayedValues(secondsNumbers);  
  
 **numberPickerHours**.setMaxValue(99);  
 **numberPickerHours**.setMinValue(0);  
 **numberPickerHours**.setDisplayedValues(hoursNumbers);  
 }  
  
 **private void** goToSuitableState(){  
 **switch** (**timerState**)  
 {  
 **case *STATE\_INITIAL***:  
 configInitialState();  
 **break**;  
 **case *STATE\_START***:  
 configStartState();  
 **break**;  
 **case *STATE\_STOP***:  
 configStopState();  
 **break**;  
 **case *STATE\_FINISHED***:  
 configFinishState();  
 **break**;  
 }  
 }  
  
 **private void** countDownTimerCombine(){  
 **countDownTimer** = **new** CountDownTimer(**timerDuration**, 1000)  
 {  
 @Override  
 **public void** onTick(**long** remainingTime)  
 {  
 **timerDuration** = remainingTime;  
  
 **if** (**fragmentState** == ***FRAGMENT\_STATE\_ON\_RESUME***)  
 calculateRemainingTime(remainingTime);  
 }  
  
 @Override  
 **public void** onFinish()  
 {  
 **timerState** = ***STATE\_FINISHED***;  
 **if** (**fragmentState** == ***FRAGMENT\_STATE\_ON\_RESUME***)  
 configFinishState();  
 }  
 };  
 **countDownTimer**.start();  
 }  
  
 **private void** calculateRemainingTime(**long** remainingTime){  
 **long** seconds = remainingTime / 1000;  
 **long** minutes = seconds / 60;  
 **long** hours = minutes / 60;  
  
 seconds = seconds % 60;  
 minutes = minutes % 60;  
  
 **textViewTimerSeconds**.setText(String.*format*(**"%02d"**, seconds));  
 **textViewTimerMinutes**.setText(String.*format*(**"%02d:"**, minutes));  
 **textViewTimerHours**.setText(String.*format*(**"%02d:"**, hours));  
  
 **if** (hours == 0 && minutes == 0 && seconds < 11)  
 makeTimerRed();  
 **else** makeTimerGray();  
 }  
  
 **private void** configInitialState(){  
 **linearLayoutTimerMessageContainer**.setVisibility(View.***GONE***);  
 **buttonStartTimer**.setVisibility(View.***VISIBLE***);  
 **buttonStopTimer**.setVisibility(View.***GONE***);  
 **buttonResumeTimer**.setVisibility(View.***GONE***);  
 **buttonResetTimer**.setVisibility(View.***GONE***);  
 **linearLayoutTimerInputNumberContainer**.setVisibility(View.***VISIBLE***);  
 **linearLayoutTimerContainer**.setVisibility(View.***GONE***);  
 **timerState** = ***STATE\_INITIAL***;  
 **timerDuration** = 0;  
 **numberPickerSeconds**.setValue(0);  
 **numberPickerMinutes**.setValue(0);  
 **numberPickerHours**.setValue(0);  
 }  
  
 **private void** configStartState(){  
 **if** (**linearLayoutTimerInputNumberContainer**.getVisibility() == View.***VISIBLE***)  
 {  
 **linearLayoutTimerInputNumberContainer**.setVisibility(View.***GONE***);  
 **linearLayoutTimerMessageContainer**.setVisibility(View.***GONE***);  
 **linearLayoutTimerContainer**.setVisibility(View.***VISIBLE***);  
 }  
 **buttonResumeTimer**.setVisibility(View.***GONE***);  
 **buttonStartTimer**.setVisibility(View.***GONE***);  
  
 **buttonStopTimer**.setVisibility(View.***VISIBLE***);  
 **buttonResetTimer**.setVisibility(View.***VISIBLE***);  
 **timerState** = ***STATE\_START***;  
 calculateRemainingTime(**timerDuration**);  
 }  
  
 **private void** configStopState(){  
 **buttonStopTimer**.setVisibility(View.***GONE***);  
 **buttonStartTimer**.setVisibility(View.***GONE***);  
 **linearLayoutTimerInputNumberContainer**.setVisibility(View.***GONE***);  
 **linearLayoutTimerMessageContainer**.setVisibility(View.***GONE***);  
 **buttonResumeTimer**.setVisibility(View.***VISIBLE***);  
 **timerState** = ***STATE\_STOP***;  
 calculateRemainingTime(**timerDuration**);  
 }  
  
 **private void** configFinishState(){  
 **linearLayoutTimerInputNumberContainer**.setVisibility(View.***GONE***);  
 **linearLayoutTimerContainer**.setVisibility(View.***VISIBLE***);  
 **buttonResumeTimer**.setVisibility(View.***GONE***);  
 **buttonStartTimer**.setVisibility(View.***GONE***);  
 **buttonStopTimer**.setVisibility(View.***GONE***);  
 **buttonResetTimer**.setVisibility(View.***VISIBLE***);  
 **timerState** = ***STATE\_FINISHED***;  
 **timerDuration** = 0;  
 **textViewTimerSeconds**.setText(String.*format*(**"%02d"**, 0));  
 **linearLayoutTimerMessageContainer**.setVisibility(View.***VISIBLE***);  
 makeTimerRed();  
 }  
  
 **private void** makeTimerRed(){  
 **textViewTimerSeconds**.setTextColor(getResources().getColor(R.color.***DarkRed***));  
 **textViewTimerMinutes**.setTextColor(getResources().getColor(R.color.***DarkRed***));  
 **textViewTimerHours**.setTextColor(getResources().getColor(R.color.***DarkRed***));  
 }  
  
 **private void** makeTimerGray(){  
 **textViewTimerSeconds**.setTextColor(getResources().getColor(R.color.***DarkGray***));  
 **textViewTimerMinutes**.setTextColor(getResources().getColor(R.color.***DarkGray***));  
 **textViewTimerHours**.setTextColor(getResources().getColor(R.color.***DarkGray***));  
 }  
}

## Màn hình bấm giờ

1. **public class** BamGioActivity **extends** AppCompatActivity {  
     
    Button **btnStop**,**btnStart**, **btnPause** ,**btnDelete**;  
    TextView **txtTimer**;  
    **long lStartTime**, **lPauseTime**, **lSystemTime** = 0L;  
    Handler **handler** = **new** Handler();  
    **boolean isRun**;  
    ListView **listView**;  
    ArrayList<String> **arrayCourse**;  
     
    Runnable **runnable** = **new** Runnable() {  
    @Override  
    **public void** run() {  
    **lSystemTime** = SystemClock.*uptimeMillis*() - **lStartTime**;  
    **long** lUpdateTime = **lPauseTime** + **lSystemTime**;  
    **long** secs = (**long**)(lUpdateTime/1000);  
    **long** mins= secs/60;  
    secs = secs %60;  
    **long** milliseconds = (**long**)(lUpdateTime%1000);  
    **txtTimer**.setText(**""**+mins+**":"** + String.*format*(**"%02d"**,secs) + **":"** + String.*format*(**"%03d"**,milliseconds));  
    **handler**.postDelayed(**this**,0);  
    }  
    };  
     
     
    @Override  
    **protected void** onCreate(Bundle savedInstanceState) {  
    **super**.onCreate(savedInstanceState);  
    setContentView(R.layout.***activity\_bam\_gio***);  
     
    **btnStart** = (Button)findViewById(R.id.***btnStart***);  
    **btnStop** = (Button)findViewById(R.id.***btnStop***);  
    **btnPause** = (Button)findViewById(R.id.***btnPause***);  
    **txtTimer** = (TextView)findViewById(R.id.***txtTimer***);  
    **listView** = (ListView)findViewById(R.id.***lvTime***);  
    **arrayCourse** = **new** ArrayList<String>();  
     
    **final** ArrayAdapter adapter = **new** ArrayAdapter(BamGioActivity.**this**,android.R.layout.***simple\_list\_item\_1***,**arrayCourse**);  
    **listView**.setAdapter(adapter);  
    **btnPause**.setOnClickListener(**new** View.OnClickListener() {  
    @Override  
    **public void** onClick(View v) {  
    String time = **txtTimer**.getText().toString();  
    Toast.*makeText*(BamGioActivity.**this**,time,Toast.***LENGTH\_LONG***).show();  
    **arrayCourse**.add(time);  
    adapter.notifyDataSetChanged();  
    }  
    });  
     
    **btnStop**.setOnClickListener(**new** View.OnClickListener() {  
    @Override  
    **public void** onClick(View v) {  
    **if**(!**isRun**)  
    **return**;  
    **isRun** = **false**;  
    **lPauseTime** = 0;  
    **handler**.removeCallbacks(**runnable**);  
    **txtTimer**.setText(**"00:00:00"**);  
    **arrayCourse**.clear();;  
    adapter.notifyDataSetChanged();  
     
    }  
    });  
     
    }  
     
    @Override  
    **protected void** onStart() {  
    **super**.onStart();  
     
    clickStart();  
    }  
     
    **public void** clickStart(){  
    **btnStart**.setOnClickListener(**new** View.OnClickListener() {  
    @Override  
    **public void** onClick(View v) {  
    **if**(**isRun**)  
    **return**;  
    **isRun** = **true**;  
    **lStartTime** = SystemClock.*uptimeMillis*();  
    **handler**.postDelayed(**runnable**, 0);  
    }  
    });  
    }  
   }

## Màn hình chuyển đổi

1. **public class** ChuyenDoiFragment **extends** Fragment {  
     
    EditText **edtChieuCao**;  
    EditText **edtCanNang**;  
    Button **btnTinh**;  
    Button **btnReset**;  
    TextView **tvChiSo**;  
    TextView **tvDanhGia**;  
    EditText **edtNhietDo**;  
    RadioButton **rdoDoC** ;  
    RadioButton **rdoDoF** ;  
    Button **btnChuyen** ;  
    TextView **tvNhietDo**;  
    Button **btnThoat**;  
     
    **public** ChuyenDoiFragment() {  
    *// Required empty public constructor* }  
     
     
    @Override  
    **public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  
    **super**.onActivityCreated(savedInstanceState);  
     
    **edtChieuCao** = (EditText) getView().findViewById(R.id.***edtChieuCao***);  
    **edtCanNang** = (EditText) getView().findViewById(R.id.***edtCanNang***);  
    **btnTinh** = (Button) getView().findViewById(R.id.***btnTinh***);  
    **btnReset** = (Button) getView().findViewById(R.id.***btnReset***);  
    **tvDanhGia** = (TextView) getView().findViewById(R.id.***txtDanhGia***);  
    **edtNhietDo** = (EditText) getView().findViewById(R.id.***edtNhietDo***);  
    **rdoDoC** = (RadioButton) getView().findViewById(R.id.***rdoBtC***);  
    **rdoDoF** = (RadioButton) getView().findViewById(R.id.***radioButton\_DoF***);  
    **btnChuyen** = (Button) getView().findViewById(R.id.***btnChuyen***);  
    **tvNhietDo** = (TextView) getView().findViewById(R.id.***txtNhietDo***);  
    **btnThoat** = (Button) getView().findViewById(R.id.***btnThoat***);  
    **tvChiSo** = (TextView) getView().findViewById(R.id.***txtChiSo***) ;  
     
    **btnTinh**.setOnClickListener(**new** View.OnClickListener() {  
    @Override  
    **public void** onClick(View v) {  
    **float** chieuCao = Float.*parseFloat*(**edtChieuCao**.getText().toString());  
    **float** canNang = Float.*parseFloat*(**edtCanNang**.getText().toString());;  
     
    **tvChiSo**.setText(String.*valueOf*(canNang / (chieuCao \* chieuCao)));  
     
    **if** (Float.*parseFloat*(**tvChiSo**.getText().toString()) < 18) {  
    **tvDanhGia**.setText(**"Bạn hơi gầy"**);  
    }  
     
    **if** (Float.*parseFloat*(**tvChiSo**.getText().toString()) < 25) {  
    **tvDanhGia**.setText(**"Bạn bình thường"**);  
    }  
     
    **if** (Float.*parseFloat*(**tvChiSo**.getText().toString()) >25 || (Float.*parseFloat*(**tvChiSo**.getText().toString()) < 30)) {  
    **tvDanhGia**.setText(**"Bạn béo phì cấp độ 1"**);  
    }  
     
    **if** (Float.*parseFloat*(**tvChiSo**.getText().toString()) > 30 || (Float.*parseFloat*(**tvChiSo**.getText().toString()) < 35)) {  
    **tvDanhGia**.setText(**"Bạn béo phì cấp độ 2"**);  
    }  
     
    **if** (Float.*parseFloat*(**tvChiSo**.getText().toString()) > 35) {  
    **tvDanhGia**.setText(**"Bạn béo phì cấp độ 3"**);  
    }  
     
    }  
    });  
     
    **btnChuyen**.setOnClickListener(**new** View.OnClickListener() {  
    @Override  
    **public void** onClick(View view) {  
     
    **float** nhietDo = Float.*parseFloat*(**edtNhietDo**.getText().toString());  
     
     
    **if** (**rdoDoC**.isChecked() == **true**) {  
     
    **double** tinh = (nhietDo - 32) / 1.8;  
    **tvNhietDo**.setText(tinh + **""**);  
     
    }  
     
    **if** (**rdoDoF**.isChecked() == **true**) {  
     
    **double** tinh = (nhietDo \* 1.8) + 32;  
    **tvNhietDo**.setText(tinh + **""**);  
     
    }  
     
     
     
    }  
    });  
     
     
     
    }  
     
    @Override  
    **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
    Bundle savedInstanceState) {  
    *// Inflate the layout for this fragment* **return** inflater.inflate(R.layout.***fragment\_chuyen\_doi***, container, **false**);  
    }  
     
   }

# CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

## Kết quả đạt được

* + - Hệ thống phải xem được đúng giờ quốc tế, xem được lịch đúng ngày kèm theo cả lịch âm
    - Hệ thống phải báo thức được và ghi chú được
    - Hệ thống phải thực hiện được đếm ngược thời gian
    - Hệ thống phải thực hiện được bấm giờ và ngắc được khoảng thời gian
    - Hệ thống phải đổi được đơn vị thời tiết, tính được chỉ số BIM

## Kết luận

### Những việc đã làm được:

* Lấy được giờ và xem được giờ đúng
* Báo thức được, hẹn giờ báo thức
* Bấm được giờ
* Đếm được giờ

### Những việc chưa làm được:

* Còn nhiều bug trong quá trình làm đồ án, cố gắng tra và sửa lỗi nhưng vẫn không xử lý được.
* Còn một số màn hình sử dụng code cứng.
* Phần màn hình lịch dương vẫn chưa thực hiện được chọn tháng và hiện ra màn hình lịch theo đúng tháng
* Đồ án vẫn có nhiều bất cập vì thời gian còn hạn chế nên chưa thể hoàn thành được đầy đủ các chức năng.