Abstract

# Introduction

## Purpose

Tài liệu này được thực hiện nhằm mục đích giúp các thành viên trong nhóm nắm bắt thông tin chi tiết về dự án Pomodoro. Nội dung bao gồm kế hoạch phát triển frontend (Android) và backend (Spring Boot + MySQL), giúp đảm bảo quá trình triển khai hiệu quả và đồng bộ giữa các phần.

## Scope

Ứng dụng Pomodoro này là một công cụ hỗ trợ quản lý thời gian, tập trung vào việc lập kế hoạch và theo dõi công việc cá nhân. Dự án bao gồm:

* **Frontend:** Ứng dụng Android được phát triển theo kiến trúc MVVM, sử dụng Java/Kotlin, Room Database và Retrofit để giao tiếp với backend.
* **Backend:** Xây dựng bằng Spring Boot với MySQL làm cơ sở dữ liệu, cung cấp API REST để quản lý công việc, lịch trình và bộ đếm Pomodoro.
* **Tính năng chính:** Quản lý phiên Pomodoro, Todo List, Daily Task, lịch trình, Import/Export Plan.
* **Giới hạn:** Ứng dụng chạy hoàn toàn cục bộ, không có tính năng đăng nhập hoặc đồng bộ hóa đám mây.

# Overall Description

## Giới thiệu

Ứng dụng Pomodoro giúp người dùng quản lý thời gian làm việc hiệu quả bằng phương pháp Pomodoro, kết hợp với danh sách công việc (TODO), kế hoạch công việc hàng ngày và lịch trình cá nhân. Ứng dụng chạy hoàn toàn cục bộ trên thiết bị Android, không yêu cầu đăng nhập hay kết nối đám mây.

## Công nghệ sử dụng

* **Frontend:** Ứng dụng Android phát triển bằng Java/Kotlin, theo kiến trúc MVVM, sử dụng Room Database để lưu dữ liệu cục bộ và Retrofit để gọi API từ backend.
* **Backend:** Sử dụng Spring Boot với MySQL để lưu trữ dữ liệu, cung cấp các API RESTful để quản lý công việc, lịch trình, bộ đếm Pomodoro.

## Chức năng chính

* Bộ đếm Pomodoro: Hỗ trợ thiết lập thời gian làm việc và nghỉ ngơi.
* Quản lý công việc (TODO List): Tạo, chỉnh sửa, đánh dấu hoàn thành và xóa công việc.
* Lịch trình công việc: Tạo kế hoạch công việc theo ngày và tuần.
* Nhập/Xuất dữ liệu: Hỗ trợ lưu trữ và khôi phục kế hoạch làm việc.

## Giới hạn của ứng dụng

Ứng dụng chỉ hoạt động cục bộ, không có tính năng đăng nhập hoặc đồng bộ hóa dữ liệu lên server.

# Specific Requirements

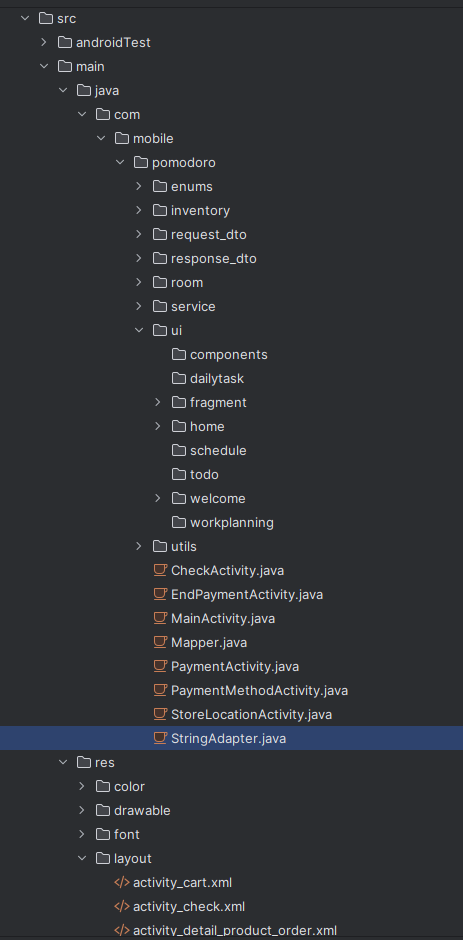
## Front-end

* 1. IDE Setup
     1. Android Studio

Emulator có thể cài máy ảo hoặc sử dụng trực tiếp điện thoại Android, chọn phương án phù hợp với bản thân.

* [Hướng dẫn cài đặt máy ảo Android.](https://youtu.be/GhuiNcOEv1A?si=nzsg-2gjZg90AwPi)
* [Hướng dẫn kết nối Android Studio với điện thoại.](https://youtu.be/apHlphSgC6o?si=wUwbg_ifsdE4pNDs)
  + 1. Config Project (Java)
* Tạo một dự án mới trong Android Studio với **Java** làm ngôn ngữ chính.
* Chọn **Empty Activity** làm template khởi tạo.
* **Name:** OurPomodoro\_App
* **Package name:** com.mobile.pomodoro
* **Minimum SDK:** API 24 (“Nougat”; Android 7.0)
* **Build configuration config:** Kotlin DSL (build.gradle.kts)

Cây thư mục:

src

└── main

├── java

│ └── com

│ └── mobile

│ └── pomodoro

│ ├── enums

│ ├── inventory

│ ├── request\_dto

│ ├── response\_dto

│ ├── room

│ │ ├── DAO

│ │ └── entity

│ ├── service

│ ├── ui

│ │ ├── components

│ │ ├── dailytask

│ │ ├── fragment

│ │ ├── home

│ │ ├── schedule

│ │ ├── todo

│ │ ├── welcome

│ │ └── workplanning

│ └── utils

└── res

└── layout

Git ignore:

[.gitignore](https://drive.google.com/file/d/1Z-s4ed3IBwZOnQkI3e8X8A3XZZAhSlWy/view?usp=sharing)

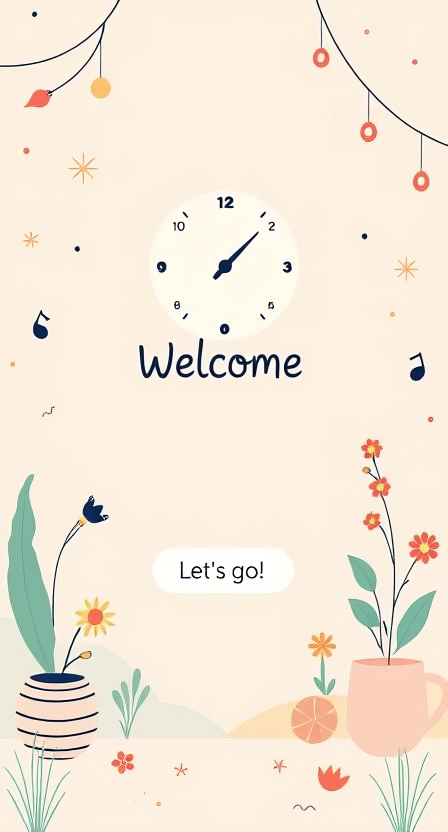
* + 1. Library
* **Room Database**: Quản lý dữ liệu cục bộ.
* **Retrofit**: Giao tiếp với API backend.
* **LiveData & ViewModel**: Hỗ trợ kiến trúc MVVM.
* **Material Components**: Thiết kế giao diện theo Material Design.
* **build.gradle**:

|  |
| --- |
| *dependencies {*  *// Room Database*  *implementation "androidx.room:room-runtime:2.5.2"*  *annotationProcessor "androidx.room:room-compiler:2.5.2"*  *// Retrofit*  *implementation "com.squareup.retrofit2:retrofit:2.9.0"*  *implementation "com.squareup.retrofit2:converter-gson:2.9.0"*  *// ViewModel & LiveData*  *implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:2.6.2"*  *implementation "androidx.lifecycle:lifecycle-livedata-ktx:2.6.2"*  *// Material Components*  *implementation "com.google.android.material:material:1.9.0"*  *}* |

* + 1. Configure AndroidManifest.xml

//

* 1. Welcome Screen
* *Mô tả*

Welcome Screen là màn hình chào mừng đầu tiên khi người dùng mở ứng dụng. Màn hình này sẽ hiển thị hình ảnh bắt mắt để tạo ấn tượng ban đầu và cung cấp một nút "Let's Go", giúp người dùng dễ dàng chuyển đến Home Screen.

* *Thành phần chính:*

Hình ảnh chào mừng: Sử dụng một hình minh họa hoặc logo ứng dụng.

Nút "Let's Go": Khi nhấn vào, ứng dụng sẽ điều hướng đến Home Screen.

* *Luồng hoạt động:*

Khi mở ứng dụng, Welcome Screen sẽ hiển thị.

Người dùng nhấn vào nút "Let's Go".

Ứng dụng sẽ chuyển hướng đến Home Screen và không quay lại Welcome Screen trừ khi cài lại ứng dụng.

*//* *Có thể dùng ảnh có sẵn hình nút let’s go, chỉ cần người dùng tương tác bất kì chỗ nào cũng sẽ // chuyển sang trang home.*

* 1. Home Screen
     1. Mô tả

Home Screen là màn hình chính của ứng dụng Pomodoro, nơi hiển thị **đồng hồ đếm ngược**, **nút điều khiển (Play/Stop)**, và **thanh điều hướng**. Người dùng có thể thay đổi **kế hoạch làm việc (Plan)** và quản lý phiên làm việc của mình.

* + 1. Giao diện chính
* **Đồng hồ đếm ngược** (ở giữa màn hình):
  + Hiển thị thời gian theo định dạng hh:mm:ss.
  + Khi bấm vào đồng hồ, mở ***cửa sổ******thông báo nảy*** để thay đổi thời gian.
  + Đếm ngược khi người dùng nhấn **Play**.
  + Khi về 00:00:00, tự động chuyển sang **công việc tiếp theo** hoặc **trở về trạng thái tĩnh**.
* **Nút điều khiển (Play/Stop)** (bên dưới đồng hồ):
  + Khi nhấn **Play**, bắt đầu đếm ngược.
  + Khi **đang đếm ngược**, nút chuyển thành **Stop**.
  + Khi nhấn **Stop**, mở ***cửa sổ******thông báo nảy*** **xác nhận gián đoạn**, cho phép:
    - **Hoàn thành** công việc.
    - **Bắt đầu lại** kế hoạch làm việc.
* **Tiêu đề phiên làm việc** (phía trên đồng hồ):
  + Hiển thị công việc hiện tại (VD: "Learning English", "Coding").
  + Tiêu đề được đặt khi **tạo một kế hoạch mới**.
  + Có thể thay đổi khi chuyển sang **công việc tiếp theo** (VD: "Short Break").
  + Khi nhấn vào **tiêu đề**, chuyển sang trang **Plan Screen**.
    1. Thanh điều hướng (Bottom Navigation)

Thanh tab nằm phía dưới màn hình chứa **5 trang lần lượt từ trái sang phải**:

* **Home** (hiện tại)
* **Plan**
* **Todo**
* **Daily**
* **Schedule**

Sử dụng **icon** để biểu diễn từng trang.

* + 1. API & Dữ liệu

Khi mở **Home Screen**, ứng dụng sẽ **GET** API để lấy **kế hoạch làm việc gần nhất** từ backend.

Nếu không tìm thấy kế hoạch nào, mặc định:

* **Thời gian:** 00:30:00
* **Tiêu đề:** Work

API lấy kế hoạch gần nhất: GET localhost/api/v1/recent-plan

* + 1. Hướng dẫn
* Đếm ngược:  
  + Sử dụng CountDownTimer.  
  + Định dạng hh:mm:ss bằng SimpleDateFormat.

|  |
| --- |
| (Tham khảo)  private CountDownTimer countDownTimer;  private static final long DEFAULT\_TIME = 1800000; // Mặc định 30 phút  private long timeLeftInMillis = DEFAULT\_TIME;  private void startTimer() {  countDownTimer = new CountDownTimer(timeLeftInMillis, 1000) {  public void onTick(long millisUntilFinished) {  timeLeftInMillis = millisUntilFinished;  updateTimerUI();  }  public void onFinish() {  moveToNextTask();  }  }.start();  }  private void updateTimerUI() {  int hours = (int) (timeLeftInMillis / 3600000);  int minutes = (int) (timeLeftInMillis % 3600000) / 60000;  int seconds = (int) (timeLeftInMillis % 60000) / 1000;  timerTextView.setText(String.format(Locale.getDefault(), "%02d:%02d:%02d", hours, minutes, seconds)); // timerTextView hiển thị timer  } |

* Nút Play/Stop  
  + Khi **Play** -> Bắt đầu CountDownTimer.  
  + Khi **Stop** -> Hiển thị popup xác nhận.

|  |
| --- |
| (Tham khảo)  playButton.setOnClickListener(v -> {  if (countDownTimer == null) {  startTimer();  playButton.setText("Stop");  } else {  showStopConfirmationDialog();  }  });  private void showStopConfirmationDialog() { // Hiển thị thông báo đơn giản  new AlertDialog.Builder(this)  .setTitle("Tạm dừng phiên làm việc?")  .setMessage("Bạn có muốn hoàn thành hoặc bắt đầu lại?")  .setPositiveButton("Hoàn thành", (dialog, which) -> moveToNextTask())  .setNegativeButton("Bắt đầu lại", (dialog, which) -> resetTimer())  .setNeutralButton("Hủy", null)  .show();  } |

* 1. Plan Screen
  2. TODO List Screen
  3. Daily Task Screen
  4. Schedules Screen
  5. Popups
     1. Timer Setter
     2. New TODO
     3. Confirm Delete
     4. Import Plan
     5. Export Plan

## Back-end

* 1. ***Core Setup***
     1. Spring Boot Project
     2. MySQL
     3. Database Connection
     4. JPA Entities
     5. Interfaces and Services
     6. REST Controllers
  2. ***Tasks***
     1. Main Feature
        1. Focus
        2. Short Break
        3. Long Break
        4. View TODO List
        5. View Daily Tasks
        6. View Weekly Planning Schedule
        7. Import/Export Plan
     2. Start Working Plan
     3. Disturb Working Plan
     4. CRUD Plan
     5. Config Plan Time
     6. CRUD TODO Plan
     7. Mark TODOs as Completed
     8. CRUD Daily Tasks
     9. Start Daily Task
     10. Import Plan
     11. Export Plan
     12. Configurable Theme