TRƯỜNG ĐẠI HỌC NÔNG LÂM THÀNH PHỐ HỒ CHÍ MINH

Khoa Công Nghệ Thông Tin



**ỨNG DỤNG QUẢN LÝ THỜI GIAN POMODORO**

BÁO CÁO TIẾN ĐỘ

**Giảng viên hướng dẫn: Nguyễn Tất Toàn**

Tác giả: Nguyễn Thái Toàn

TPHCM, Ngày 20 tháng 04 năm 202

Contents

[II. Sinh viên thực hiện 3](#_Toc200030648)

[III. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 4](#_Toc200030649)

[1. Bối cảnh và lý do chọn đề tài 4](#_Toc200030650)

[2. Mục tiêu của đề tài 4](#_Toc200030651)

[3. Công nghệ sử dụng 4](#_Toc200030652)

[IV. Cơ sở lý thuyết 5](#_Toc200030653)

[1. Tổng quan về phương pháp Pomodoro 5](#_Toc200030654)

[2. Kiến thức về Android Java 5](#_Toc200030655)

[3. Kiến thức về Spring Boot 5](#_Toc200030656)

[4. Mô hình Client–Server 6](#_Toc200030657)

[5. Cơ sở dữ liệu 6](#_Toc200030658)

[6. Kiến thức về API RESTful 6](#_Toc200030659)

[V. Phân tích và thiết kế hệ thống 7](#_Toc200030660)

[1. Sơ đồ kiến trúc tổng thể (Architecture Diagram) 7](#_Toc200030661)

[2. Cơ sở dữ liệu và mô hình ERD 8](#_Toc200030662)

[VI. Thiết kế giao diện người dùng 9](#_Toc200030663)

[1. Màn hình đăng nhập, đăng ký 9](#_Toc200030664)

[2. Màn hình chào mừng 10](#_Toc200030665)

[3. Màn hình chính 11](#_Toc200030666)

[4. Màn hình trang Todo 12](#_Toc200030667)

[5. Màn hình trang Plan 13](#_Toc200030668)

[6. Màn hình trang Daily Task 14](#_Toc200030669)

[7. Màn hình trang Settings 15](#_Toc200030670)

[VII. Đánh giá – Nhận xét 16](#_Toc200030671)

[1. Ưu điểm 16](#_Toc200030672)

[2. Hạn chế 16](#_Toc200030673)

[3. Hướng phát triển tương lai 16](#_Toc200030674)

[VIII. Kết luận 18](#_Toc200030675)

# Sinh viên thực hiện

1. 21130569 - Nguyễn Thái Toàn
2. 22130016 - Trần Tiến Anh
3. 22130021 - Nguyễn Thái Bảo
4. 22130067 - Hồ Minh Hải
5. 22130199 - Lê Yến Nhung

|  |  |
| --- | --- |
| Sinh viên thực hiện | Công việc được phân công |
| 21130569 - Nguyễn Thái Toàn | * Phân chia công việc. * Viết báo cáo. * Quản lý tiến độ, đảm bảo chất lượng đồ án. * Xây dựng cấu trúc cơ bản Backend. * Xây dựng chức năng đăng nhập/đăng ký. * Xây dựng chức năng lấy plan gần nhất. * Xây dựng chức năng Frontend gửi request về Backend bằng retrofit. * Thiết kế database. |
| 22130016 - Trần Tiến Anh | * Xây dựng các màn hình đăng ký, đăng nhập. * Xây dựng màn hình trang chủ. * Xây dựng màn hình trang Setting. * Xây dựng màn hình trang Daily Task. * Xây dựng chức năng đếm ngược. |
| 22130021 - Nguyễn Thái Bảo | * Xây dựng chức năng xử lý thông tin gửi đến từ Frontend. * Xây dựng chức năng lấy Daily Task theo id. * Xây dựng các chức năng tạo, xóa, cập nhật Daily Task. |
| 22130067 - Hồ Minh Hải | * Xây dựng các chức năng tạo, xóa, cập nhật Todo. * Xây dựng chức năng lưu Plan. * Xây dựng chức năng lấy toàn bộ Daily Task của user. * Xây dựng chức năng thêm Daily Task. * Xây dựng chức năng xử lý thông tin gửi đến tư |
| 22130199 - Lê Yến Nhung | * Xây dựng màn hình trang todo. * Xây dựng màn hình trang plan. * Xây dựng màn hình trang chào mừng. * Xây dựng chức năng đếm ngược. * Xây dựng chức năng chuyển tiếp công việc. * Xây dựng chức năng tạo nghỉ ngắn, dài. |

# GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## Bối cảnh và lý do chọn đề tài

Trong thời đại số hiện nay, con người thường xuyên phải đối mặt với tình trạng mất tập trung, trì hoãn và thiếu hiệu quả trong công việc hoặc học tập. Một trong những giải pháp hiệu quả đã được khoa học chứng minh là phương pháp Pomodoro – kỹ thuật quản lý thời gian thông qua việc chia nhỏ thời gian làm việc thành các khoảng cố định (thường là 25 phút), xen kẽ với các khoảng nghỉ ngắn.

Việc ứng dụng kỹ thuật Pomodoro thông qua một hệ thống phần mềm không chỉ giúp người dùng dễ dàng áp dụng phương pháp này mà còn hỗ trợ theo dõi hiệu suất làm việc, xây dựng thói quen tập trung, và cải thiện khả năng quản lý thời gian cá nhân.

Chính vì vậy, nhóm đã quyết định xây dựng đề tài **"Ứng dụng quản lý thời gian học tập/làm việc theo phương pháp Pomodoro trên nền tảng Android"**, với tên gọi **OUR\_Pomodoro**.

## Mục tiêu của đề tài

* Xây dựng một ứng dụng di động Android cho phép người dùng:
  + Tạo và quản lý kế hoạch làm việc theo phương pháp Pomodoro.
  + Theo dõi thời gian làm việc, thời gian nghỉ ngơi và lịch sử hoạt động.
  + Quản lý danh sách công việc (To-do List) và lịch trình hàng ngày.
  + Nhận các thông báo nhắc nhở đúng thời điểm.
* Phát triển hệ thống backend sử dụng Spring Boot để:
  + Xử lý lưu trữ và đồng bộ dữ liệu giữa nhiều thiết bị.
  + Hỗ trợ quản lý người dùng và đăng nhập bảo mật.
  + Phân tích dữ liệu để hỗ trợ thống kê và đánh giá hiệu suất.

## Công nghệ sử dụng

* **Frontend**: Android Java
  + Xây dựng giao diện người dùng thân thiện, dễ sử dụng.
  + Tương tác với backend thông qua Retrofit API.
* **Backend**: Spring Boot (Java)
  + Thiết kế API RESTful.
  + Kết nối cơ sở dữ liệu (MySQL/PostgreSQL) để lưu trữ thông tin người dùng, kế hoạch và lịch trình.

# Cơ sở lý thuyết

## Tổng quan về phương pháp Pomodoro

Phương pháp Pomodoro là một kỹ thuật quản lý thời gian được phát triển bởi Francesco Cirillo vào cuối những năm 1980. Ý tưởng cốt lõi là chia thời gian làm việc thành các khoảng thời gian tập trung cao độ (gọi là "Pomodoro") – thường là 25 phút – xen kẽ với các khoảng nghỉ ngắn (5 phút), và sau 4 chu kỳ sẽ có một khoảng nghỉ dài hơn (15–30 phút).

Lợi ích của Pomodoro:

* Tăng khả năng tập trung và năng suất làm việc.
* Giảm tình trạng kiệt sức và trì hoãn.
* Tạo thói quen làm việc khoa học.

Quy trình Pomodoro:

1. Xác định công việc cần làm.
2. Đặt hẹn giờ 25 phút (1 Pomodoro).
3. Làm việc không bị gián đoạn cho đến khi hết giờ.
4. Nghỉ ngắn 5 phút.
5. Sau mỗi 4 Pomodoro, nghỉ dài từ 15–30 phút.

## Kiến thức về Android Java

Android là hệ điều hành mã nguồn mở dựa trên nhân Linux, được Google phát triển và sử dụng phổ biến trong các thiết bị di động. Ứng dụng Android có thể được xây dựng bằng Java hoặc Kotlin.

Kiến thức Android liên quan đề tài:

* Activity/Fragment: Thành phần chính của giao diện người dùng.
* ViewModel: Quản lý dữ liệu và trạng thái giao diện.
* Room Database: Thư viện quản lý cơ sở dữ liệu SQLite cục bộ.
* Notification: Hệ thống nhắc nhở người dùng thực hiện phiên làm việc Pomodoro.
* Retrofit: Thư viện giao tiếp API RESTful giữa frontend và backend.

## Kiến thức về Spring Boot

**Spring Boot** là một framework giúp đơn giản hóa việc phát triển ứng dụng backend bằng Java. Nó cung cấp cấu trúc rõ ràng, dễ cấu hình, hỗ trợ RESTful API, kết nối cơ sở dữ liệu, và bảo mật.

**Các thành phần chính dùng trong đề tài:**

* **@RestController**: Định nghĩa các API endpoint.
* **@Service**: Xử lý logic nghiệp vụ.
* **@Repository + JPA/Hibernate**: Truy xuất cơ sở dữ liệu.
* **Spring Security (nếu dùng)**: Bảo mật API và xác thực người dùng.

## Mô hình Client–Server

Ứng dụng xây dựng theo mô hình **Client – Server**:

* **Client**: Ứng dụng Android giao diện người dùng, xử lý tương tác.
* **Server**: Ứng dụng Spring Boot xử lý dữ liệu, quản lý logic và API.
* Hai phần giao tiếp thông qua REST API.

## Cơ sở dữ liệu

Đề tài sử dụng **MySQL** làm hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Dữ liệu người dùng, kế hoạch Pomodoro, To-do list, lịch trình,… được lưu trữ dưới dạng bảng với các mối quan hệ rõ ràng (1–n, n–n).

## Kiến thức về API RESTful

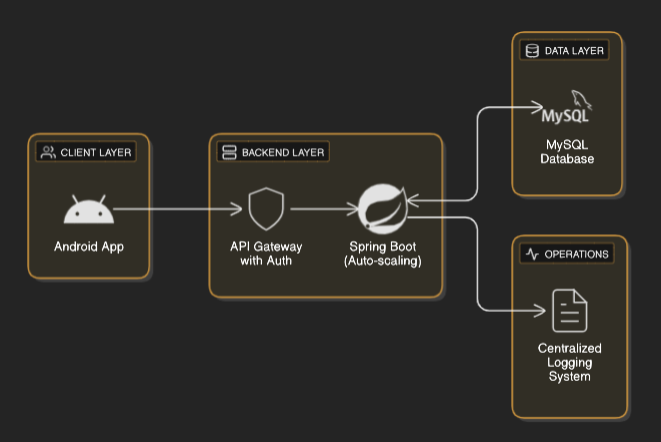
RESTful API là kiến trúc phổ biến trong giao tiếp client–server. Các phương thức HTTP chính:

* GET: Lấy dữ liệu.
* POST: Tạo dữ liệu mới.
* PUT/PATCH: Cập nhật dữ liệu.
* DELETE: Xóa dữ liệu.

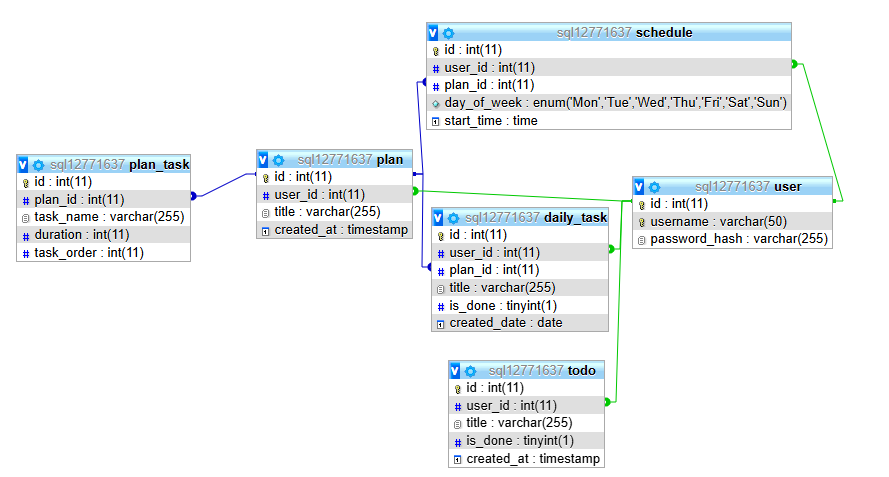
REST API giúp frontend (Android) và backend (Spring Boot) giao tiếp với nhau một cách hiệu quả, dễ dàng mở rộng và bảo trì.

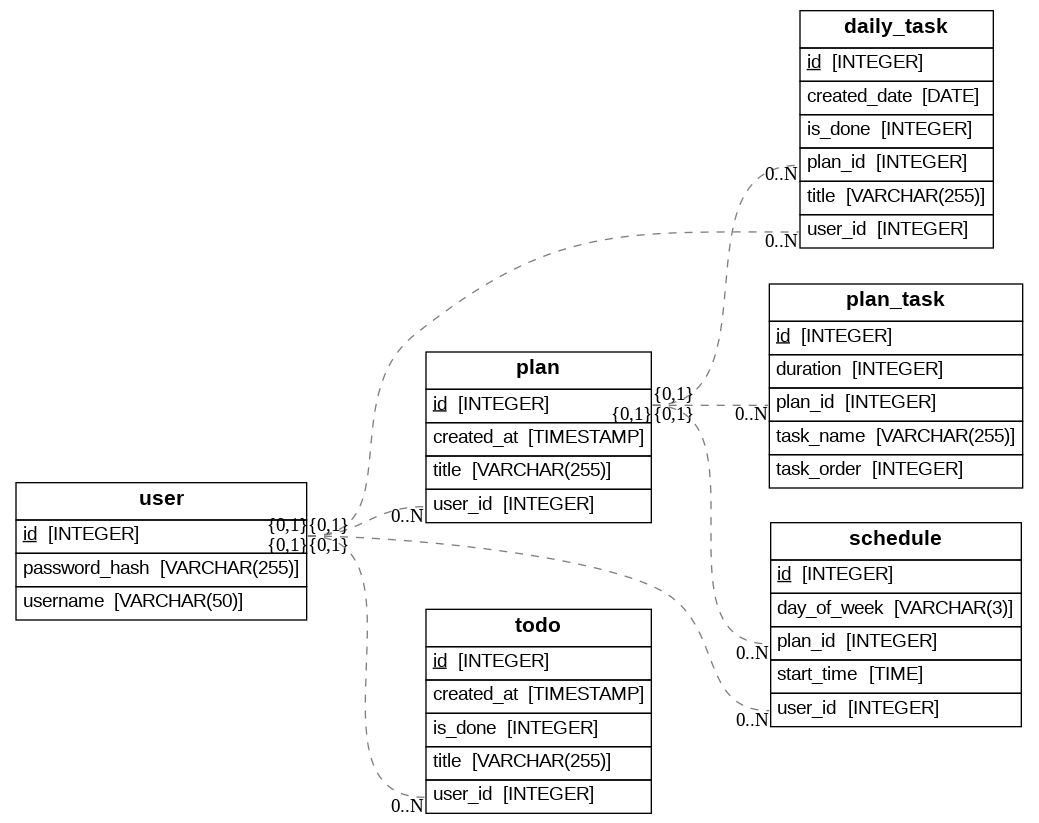
# Phân tích và thiết kế hệ thống

## Sơ đồ kiến trúc tổng thể (Architecture Diagram)



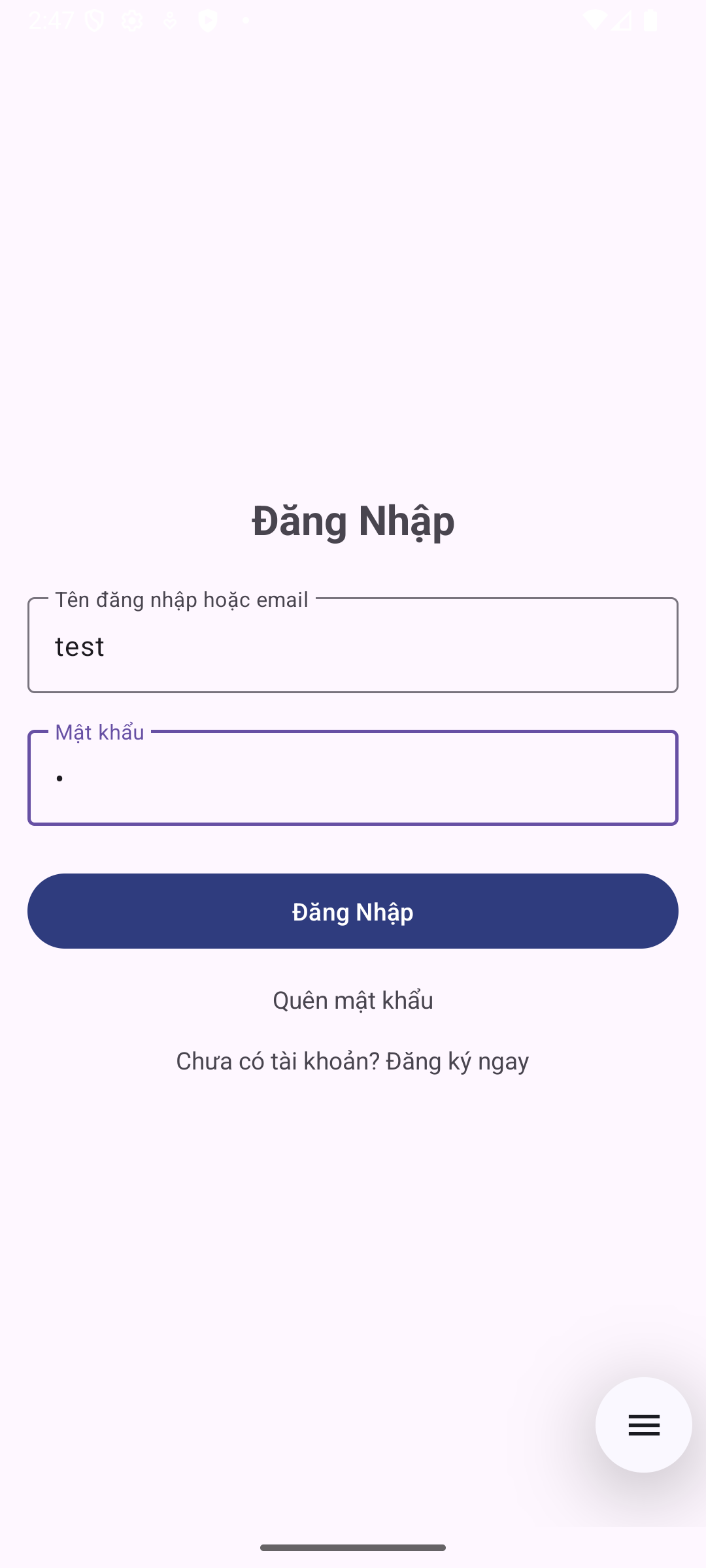
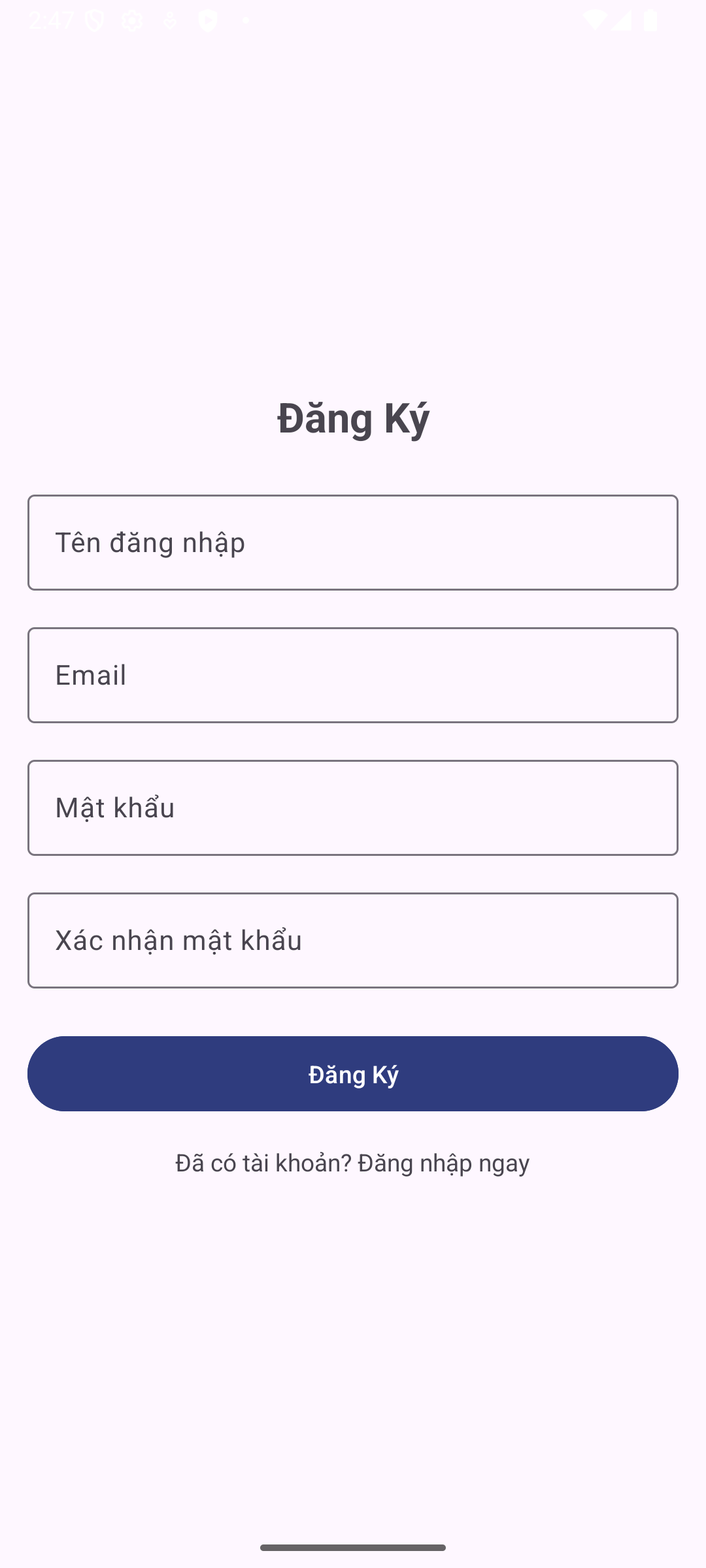
## Cơ sở dữ liệu và mô hình ERD

* Cơ sở dữ liệu:
* Mô hình ERD:



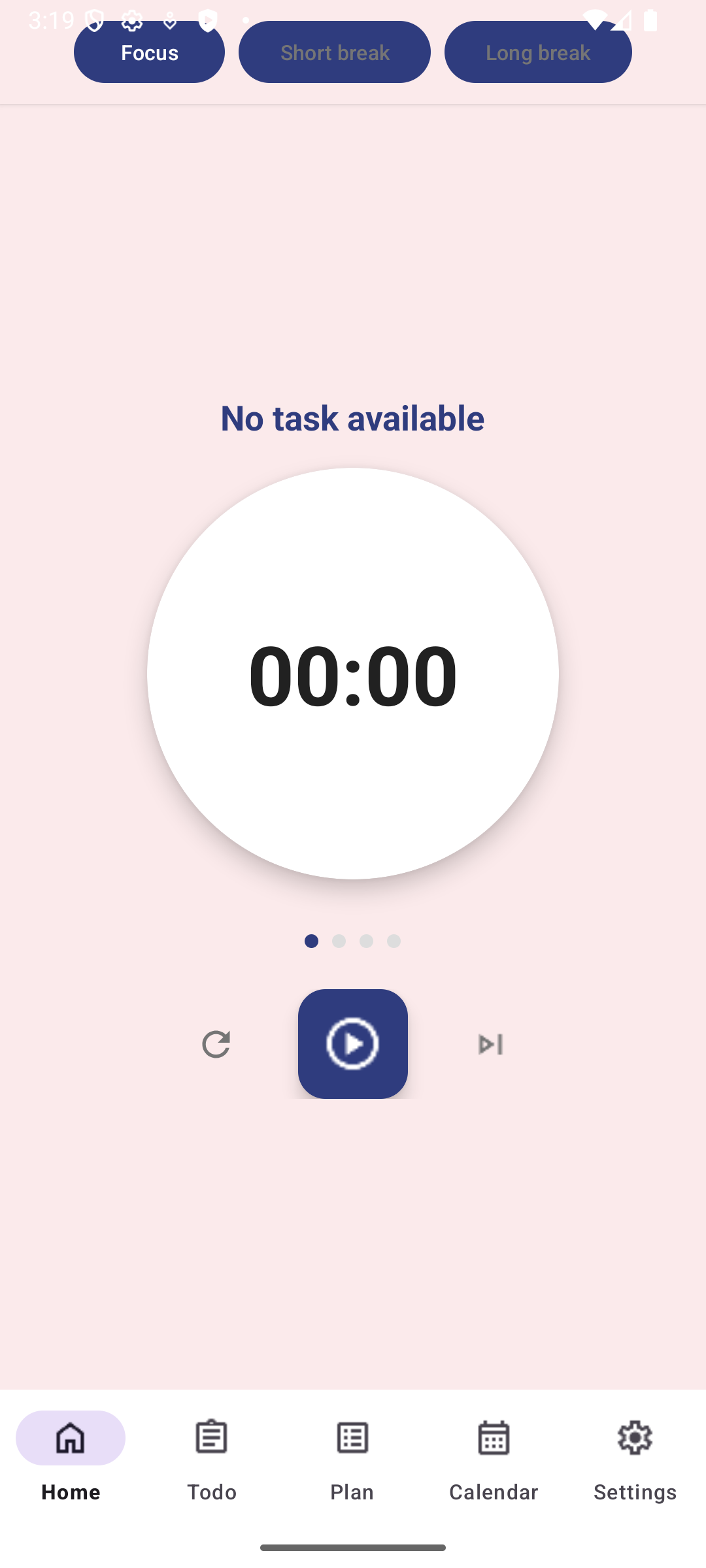
# Thiết kế giao diện người dùng

## Màn hình đăng nhập, đăng ký

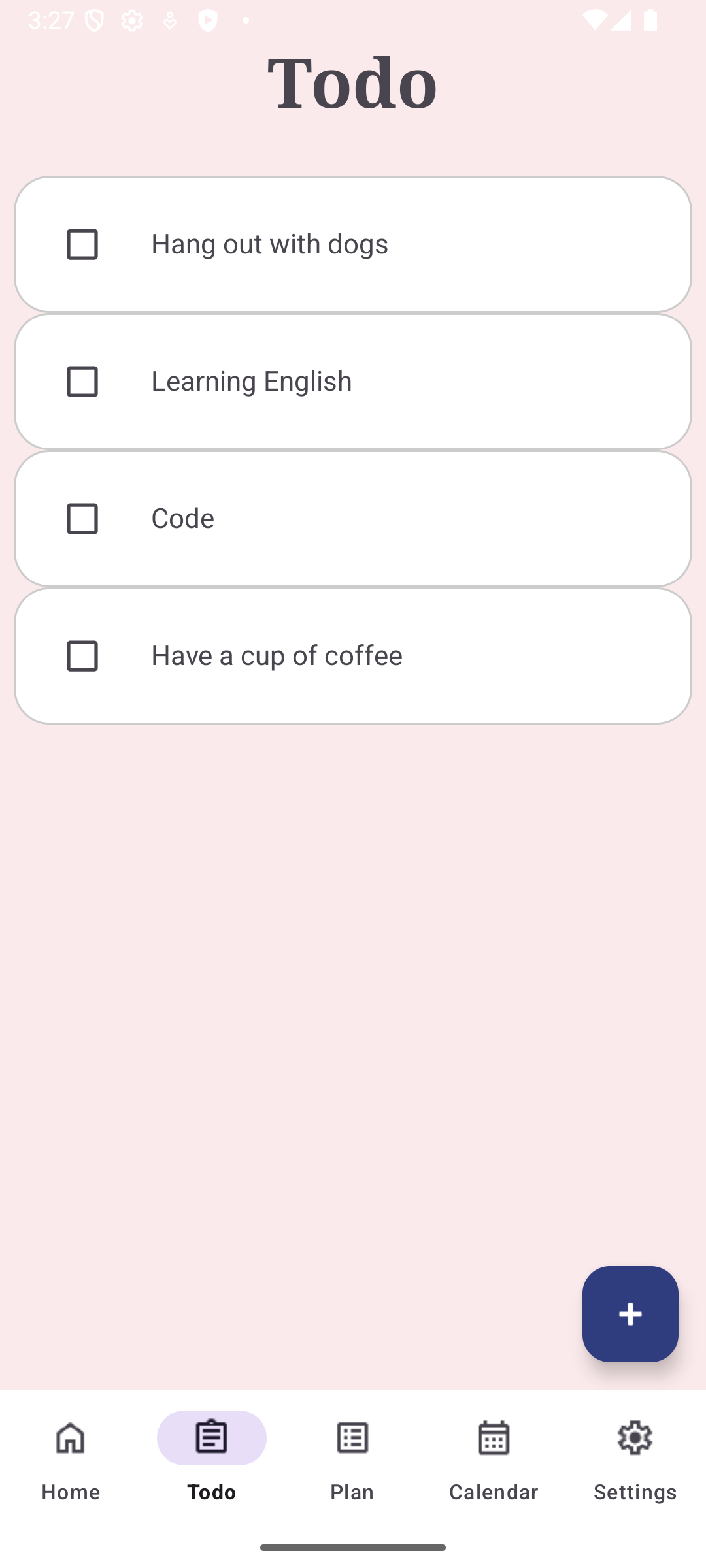
 

## Màn hình chào mừng

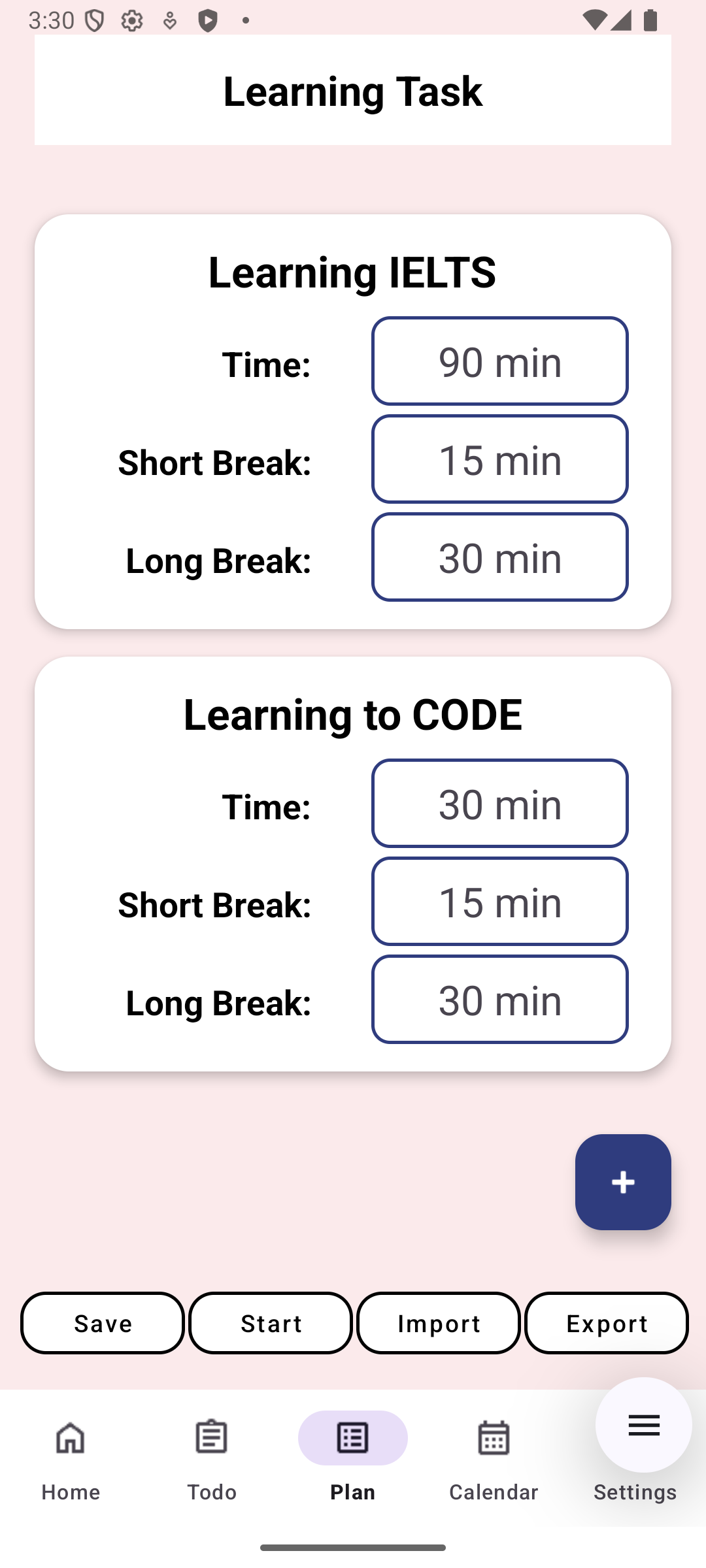
## Màn hình chính



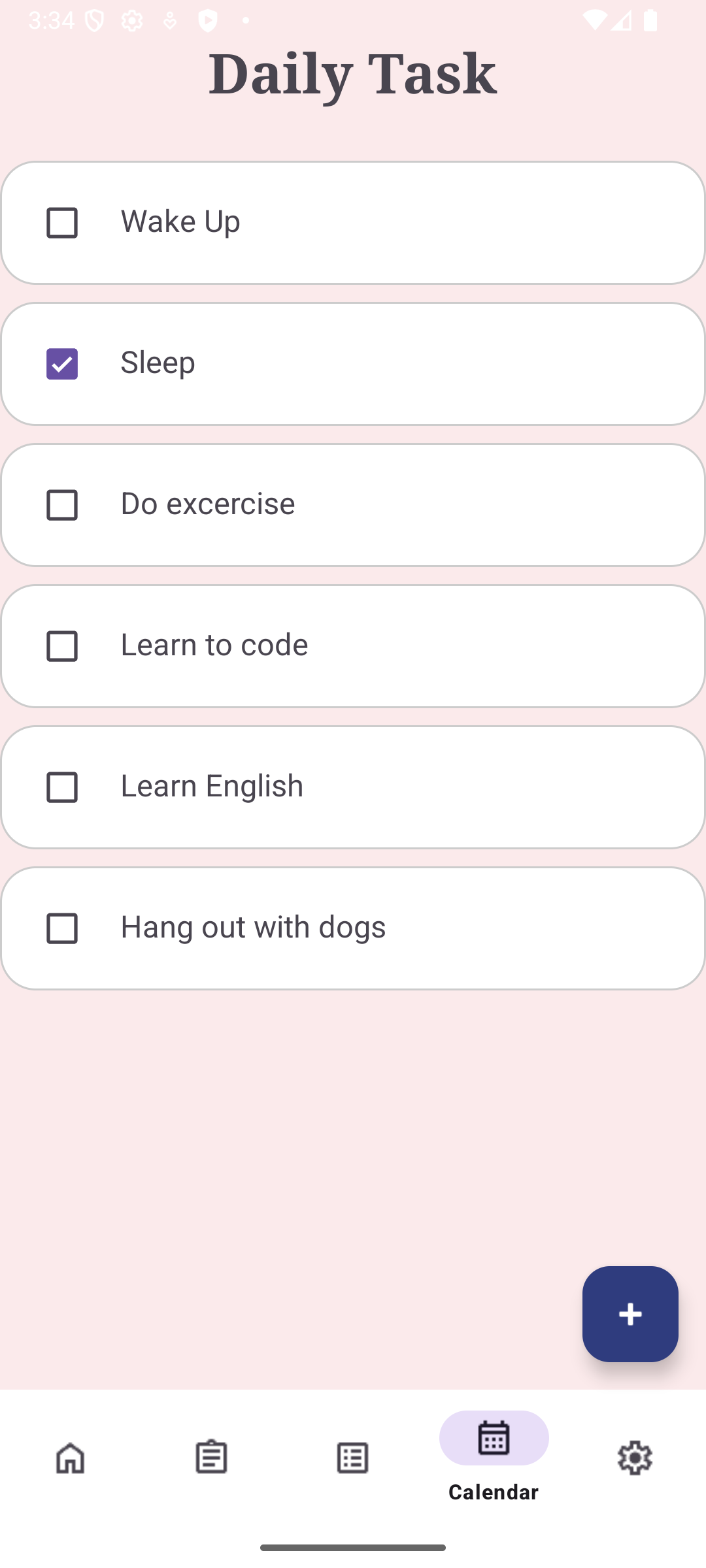
## Màn hình trang Todo



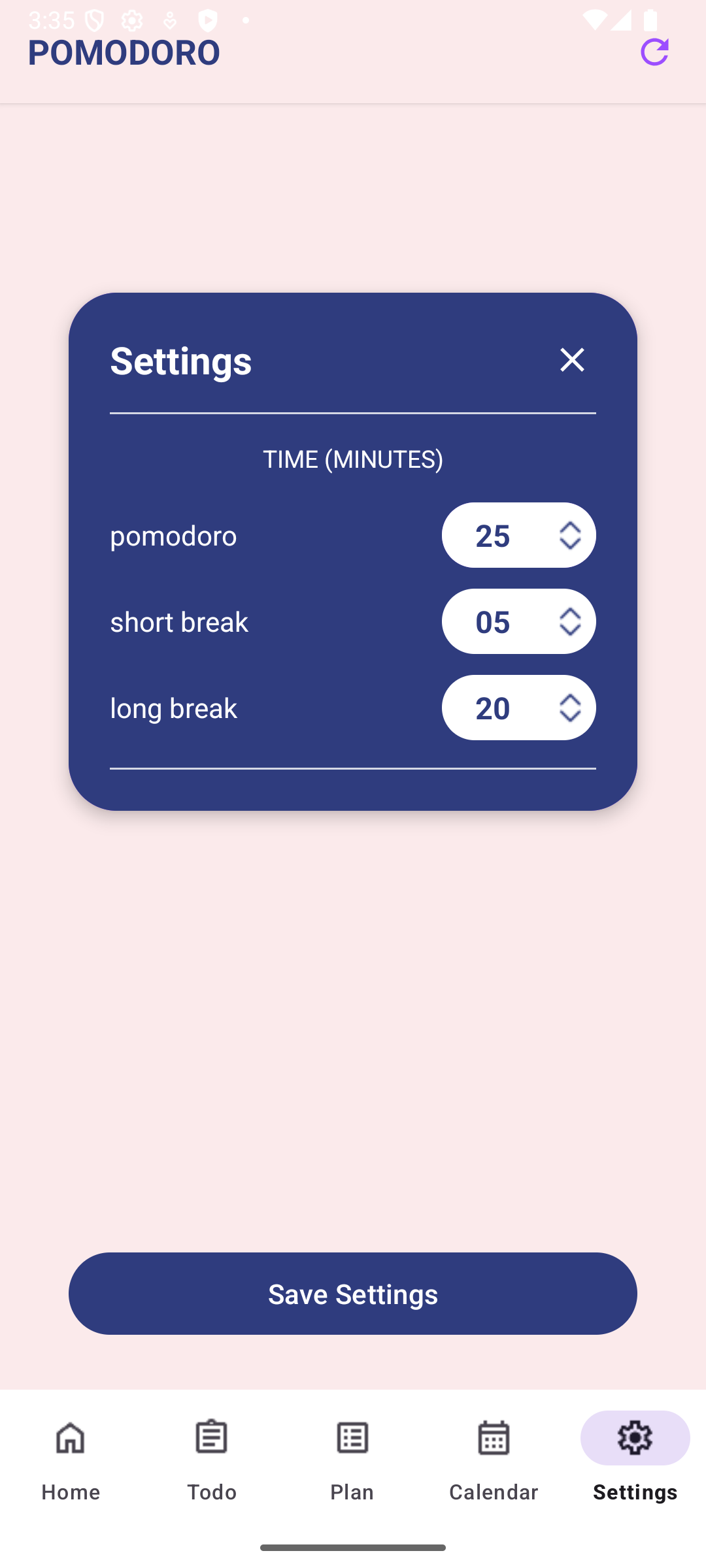
## Màn hình trang Plan



## Màn hình trang Daily Task



## Màn hình trang Settings



# Đánh giá – Nhận xét

## Ưu điểm

 **Ứng dụng thiết thực**: Đề tài giải quyết vấn đề phổ biến về quản lý thời gian và tập trung trong học tập/làm việc – rất phù hợp với học sinh, sinh viên, người đi làm.

 **Áp dụng phương pháp khoa học**: Việc ứng dụng phương pháp Pomodoro giúp người dùng có quy trình làm việc rõ ràng, khoa học và hiệu quả hơn.

 **Tách biệt frontend và backend**: Sử dụng mô hình Client–Server với Android Java và Spring Boot giúp hệ thống dễ mở rộng và bảo trì.

 **Giao diện thân thiện, dễ sử dụng**: Ứng dụng xây dựng giao diện trực quan, tập trung vào trải nghiệm người dùng với các chức năng cơ bản như: Pomodoro, lịch trình, to-do list,...

 **Tích hợp cơ sở dữ liệu**: Dữ liệu người dùng và các hoạt động được lưu trữ an toàn, có thể khôi phục và phân tích.

## Hạn chế

 **Chưa tích hợp đồng bộ đám mây (cloud)**: Dữ liệu người dùng hiện chưa được lưu trữ trên nền tảng đám mây, gây khó khăn khi sử dụng nhiều thiết bị hoặc khi gỡ cài đặt.

 **Chưa hỗ trợ đa người dùng mạnh mẽ**: Nếu không có tính năng đăng nhập/đăng ký hoặc phân quyền người dùng rõ ràng, việc cá nhân hóa trải nghiệm còn hạn chế.

 **Thiếu báo cáo thống kê**: Chưa có biểu đồ hay tổng hợp thời gian làm việc, số phiên Pomodoro hoàn thành theo ngày/tuần để người dùng tự đánh giá hiệu suất.

 **Không có chức năng chia sẻ hay nhắc lịch thông minh**: Ứng dụng còn thiếu tính năng chia sẻ kế hoạch hoặc đồng bộ với lịch Google, tạo nhắc việc thông minh.

## Hướng phát triển tương lai

 **Tích hợp Firebase hoặc AWS để đồng bộ dữ liệu người dùng**.

 **Bổ sung trang thống kê dữ liệu Pomodoro theo ngày, tuần, tháng** (sử dụng biểu đồ).

 **Tạo hệ thống tài khoản cá nhân** giúp người dùng đăng nhập, lưu tiến độ và dùng trên nhiều thiết bị.

 **Thêm tính năng nhắc nhở thông minh**, kết hợp AI để gợi ý lịch làm việc hiệu quả.

 **Gamification**: Thêm phần thưởng hoặc cấp bậc khi người dùng hoàn thành mục tiêu để tăng động lực.

# Kết luận

Qua quá trình nghiên cứu và phát triển, nhóm đã hoàn thành đề tài "Ứng dụng quản lý thời gian Pomodoro" với chức năng chính bao gồm: tạo kế hoạch làm việc theo chu kỳ Pomodoro, quản lý công việc theo ngày (Daily Task), to-do list, và hiển thị lịch sử hoạt động. Hệ thống được xây dựng theo mô hình Client–Server, trong đó ứng dụng Android (Java) đóng vai trò giao diện người dùng và backend sử dụng Spring Boot đảm nhận xử lý dữ liệu, xác thực và lưu trữ trên MySQL.

Bên cạnh việc áp dụng lý thuyết về kỹ thuật Pomodoro, đề tài còn giúp nhóm củng cố và nâng cao kiến thức lập trình di động, lập trình backend, thiết kế cơ sở dữ liệu cũng như khả năng làm việc nhóm. Mặc dù vẫn còn một số hạn chế như chưa hỗ trợ đồng bộ đám mây, chưa có thống kê trực quan và tính năng chia sẻ thông minh, nhưng đề tài đã đạt được mục tiêu đề ra ban đầu và có tiềm năng phát triển mạnh mẽ trong tương lai.

Trong giai đoạn tiếp theo, nhóm định hướng sẽ tích hợp các tính năng nâng cao như: đồng bộ đa thiết bị bằng Firebase, thống kê hiệu suất bằng biểu đồ, và nhắc nhở thông minh bằng AI nhằm hoàn thiện ứng dụng và mang lại trải nghiệm tốt hơn cho người dùng.