

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐHQG TP.HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**  
**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



**TÀI LIỆU HƯỚNG DẪN CÀI ĐẶT CÔNG CỤ VÀ BIÊN DỊCH MÃ NGUỒN**

**ỨNG DỤNG LUYỆN THI IELTS CHO HỆ ĐIỀU HÀNH ANDROID**

**GVHD:**

TS. Ngô Huy Biên

**Sinh viên thực hiện:**

19127626 Lê Nguyễn Tú Văn

19127631 Dương Tiến Vinh

19127373 Hồ Văn Duy

19127294 Nguyễn Trần Thiện  
Toàn

19127607 Trần Nguyên Trung

19127597 Trần Khả Trí

Tp. Hồ Chí Minh, tháng 7 năm 2023

## **Mục lục**

<b>Hướng dẫn cài đặt công cụ và biên dịch mã nguồn Backend</b>	<b>3</b>
<b>Tổng quan</b>	<b>3</b>
<b>Cài đặt công cụ</b>	<b>3</b>
<b>Cài đặt asdf:</b>	<b>3</b>
<b>Cài đặt công cụ hỗ trợ</b>	<b>3</b>
<b>Cài đặt Go và các công cụ liên quan</b>	<b>4</b>
<b>Tải mã nguồn</b>	<b>5</b>
<b>Sử dụng mã nguồn</b>	<b>5</b>
<b>Khởi động hệ thống thủ công</b>	<b>5</b>
<b>Dừng hệ thống thủ công</b>	<b>6</b>
<b>Khởi động hệ thống tự động</b>	<b>6</b>
<b>Dừng hệ thống tự động</b>	<b>6</b>
<b>Các lệnh hỗ trợ khác</b>	<b>7</b>
<b>Biên dịch mã nguồn</b>	<b>7</b>
<b>Hướng dẫn cài đặt công cụ và biên dịch mã nguồn Frontend</b>	<b>8</b>
<b>Tổng quan</b>	<b>8</b>
<b>Cài Android Studio</b>	<b>8</b>
<b>Tạo máy ảo android</b>	<b>9</b>
<b>Cài Flutter SDK</b>	<b>13</b>
<b>Tải mã nguồn</b>	<b>16</b>
<b>Sử dụng mã nguồn</b>	<b>17</b>

# Hướng dẫn cài đặt công cụ và biên dịch mã nguồn Backend

## Tổng quan

Hệ thống Backend của nhóm được lập trình bằng ngôn ngữ Go. Để biên dịch mã nguồn thành các file thực thi, người lập trình viên cần cài đặt Go và các công cụ liên quan.

Yêu cầu hệ thống:

- Hệ điều hành: Ubuntu 22.10.
- Terminal: zsh 5.9 (x86\_64-ubuntu-linux-gnu).

## Cài đặt công cụ

Cài đặt [asdf](#):

Version: v0.10.2.

```
git clone https://github.com/asdf-vm/asdf.git ~/.asdf --branch v0.10.2
```

```
echo ". ~/.asdf/asdf.sh" >> ~/.zshrc  
echo ". ~/.asdf/completions/asdf.bash" >> ~/.zshrc
```

## Cài đặt công cụ hỗ trợ

- Cài đặt các công cụ hỗ trợ:

```
sudo apt-get install -y git build-essential
```

- Cài đặt Docker và Docker Compose:

```
#!/usr/bin/env bash  
  
# Uninstall old versions  
sudo apt-get remove docker docker-engine docker.io containerd runc  
  
# Set up the repository  
  
# Update the apt package index and install packages to allow apt to use a  
# repository over HTTPS  
sudo apt-get update  
  
sudo apt-get install \  
    ca-certificates \  
    curl \  
    gnupg \  
    lsb-release  
  
# Add Docker's official GPG key
```

```
sudo mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Use the following command to set up the repository
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install Docker Engine

# Update the apt package index
sudo apt-get update

# Install Docker Engine, containerd, and Docker Compose.
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

# Post installation steps

# Add your user to the docker group
sudo usermod -aG docker $USER
```

## Cài đặt Go và các công cụ liên quan

- Cài đặt [Go](#):

Version: 1.20.4.

```
asdf plugin add golang https://github.com/kennyp/asdf-golang.git

asdf install golang 1.20.4

asdf global golang 1.20.4
```

- Cài đặt [go-migrate](#):

Version: 4.16.0.

```
asdf plugin add gomigrate https://github.com/joschi/asdf-gomigrate

asdf install gomigrate 4.16.0

asdf global gomigrate 4.16.0
```

- Cài đặt [Buf CLI](#):

Version: 1.19.0.

```
asdf plugin add buf https://github.com/truepay/asdf-buf
```

```
asdf install buf 1.19.0
```

```
asdf global buf 1.19.0
```

## Tải mã nguồn

Tải mã nguồn từ GitHub:

```
git clone https://github.com/the-company-tcus/backend.git
```

Người dùng sẽ đăng nhập bằng tài khoản GitHub với username và với password là Personal Access Token (PAT) của người dùng.

Note: Lưu ý Personal Access Token của người dùng cần được cấp quyền repo. Ngoài ra, người dùng cần phải được nhóm cấp quyền truy cập vào GitHub Organization của nhóm.

## Sử dụng mã nguồn

Note: Các lệnh make phải được chạy ở root directory của mã nguồn (nơi có file Makefile).

## Khởi động hệ thống thủ công

Để chạy hệ thống, người lập trình viên sẽ chạy lệnh sau:

### 1. Chạy CSDL:

Chạy PostgreSQL database container và chạy go-migrate để tạo các bảng trong database:

```
make up
```

### 2. Chạy các service:

#### ● api-gateway service:

```
go run cmd/server/main.go api-gateway
```

#### ● auth service:

```
go run cmd/server/main.go auth
```

#### ● user service:

```
go run cmd/server/main.go user
```

#### ● eureka service:

```
go run cmd/server/main.go eureka
```

- notification service:

```
go run cmd/server/main.go notification
```

## Dừng hệ thống thủ công

1. Xóa PostgreSQL database container:

```
make down
```

2. Sau đó, người lập trình viên có thể dừng các service bằng cách nhấn Ctrl + C.

## Khởi động hệ thống tự động

Note: Cách này được khuyến khích cho các lập trình viên Frontend (Mobile).

Để chạy hệ thống, người lập trình viên sẽ chạy lệnh sau:

```
make compose-up
```

Hoặc:

Khởi động hệ thống "sạch" (Rebuild lại Docker images):

```
make compose-fresh
```

Note: Lệnh này nên được các lập trình viên sử dụng khi có thay đổi về source code.

## Dừng hệ thống tự động

Để dừng hệ thống, người lập trình viên sẽ chạy lệnh sau:

```
make compose-down
```

Hoặc:

Dừng hệ thống, nhưng không xóa Keycloak container:

```
make compose-quickdown
```

Note: Việc khởi động lại Keycloak container sẽ mất nhiều thời gian hơn so với việc khởi động lại các service khác. Nên việc sử dụng lệnh này sẽ giúp cho việc phát triển nhanh hơn.

### Các lệnh hỗ trợ khác

- Generate protobuf với Buf CLI:

```
make gen-proto-go
```

### Biên dịch mã nguồn

Để biên dịch mã nguồn, người lập trình viên sẽ chạy lệnh sau:

```
go build -o main ./cmd/server/main.go
```

File main sẽ được tạo ra trong thư mục hiện tại. Người lập trình viên có thể chạy file main để khởi động service.

Ví dụ:

```
./main [service]
```

# Hướng dẫn cài đặt công cụ và biên dịch mã nguồn Frontend

## Tổng quan

Hệ thống Frontend của nhóm được lập trình bằng ngôn ngữ Dart, cụ thể sử dụng Framework là Flutter. Với thêm công cụ Android Studio để tạo thiết bị giả lập để chạy mã nguồn.

Yêu cầu hệ thống:

- Hệ điều hành: Ubuntu 22.10.
- Terminal: zsh 5.9 (x86\_64-ubuntu-linux-gnu).

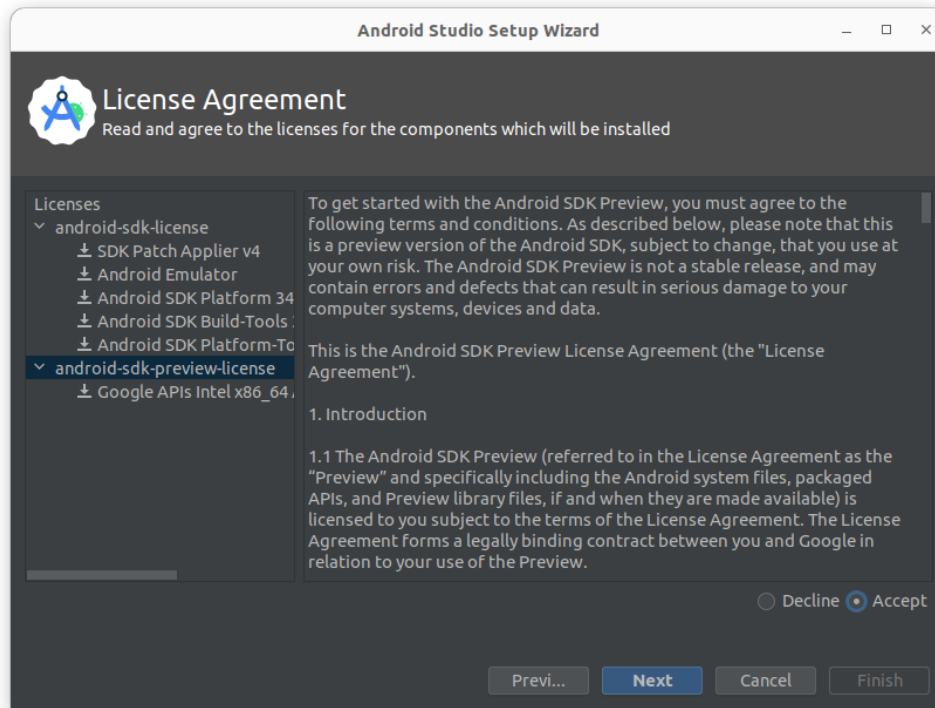
## Cài Android Studio

Tải tập tên cài đặt Android Studio [ở đây](#)

Sau khi tải xong, giải nén file .zip. Rồi bật terminal rồi di chuyển đến (đường dẫn đến thư mục giải nén khi này)/android-studio/bin/. Sau đó gõ lệnh

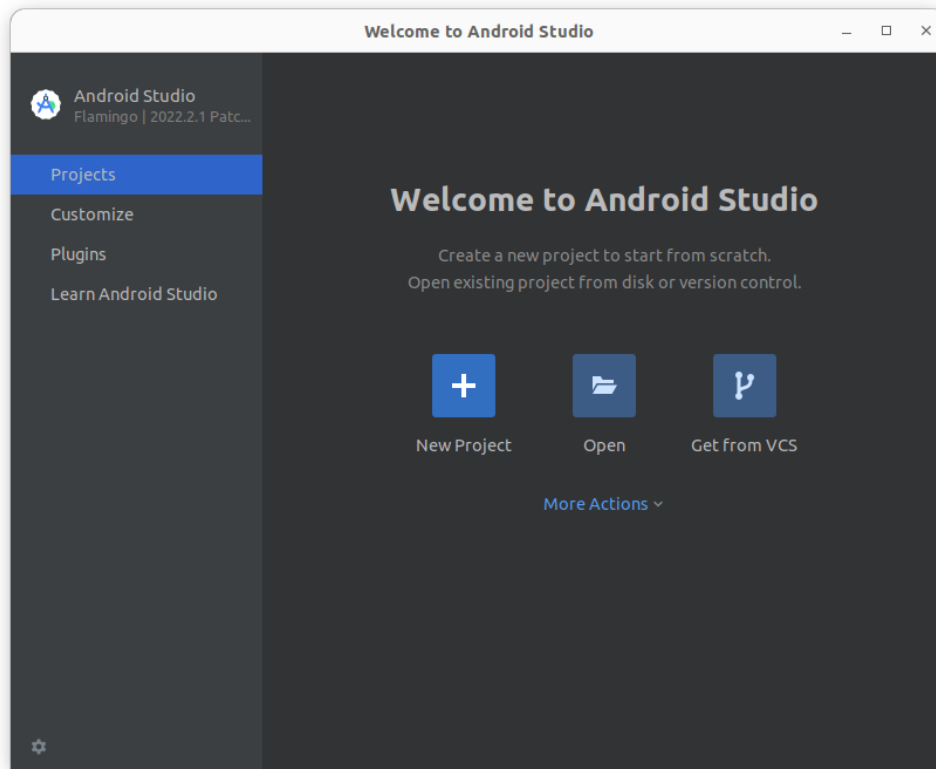
```
./studio.sh
```

Khi đó Android studio sẽ khởi động, chúng ta sẽ thiết lập cá nhân và cài đặt tất cả những gì ứng dụng yêu cầu.



Sau khi chọn ‘Accept’ và ‘Next’ rồi đợi cài đặt

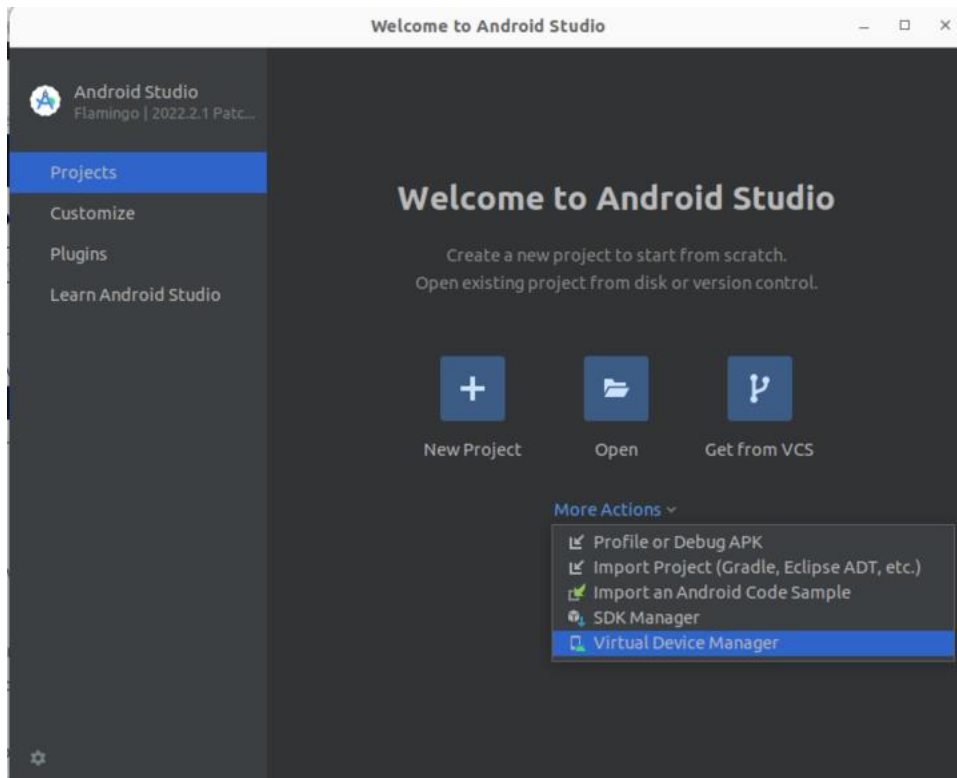




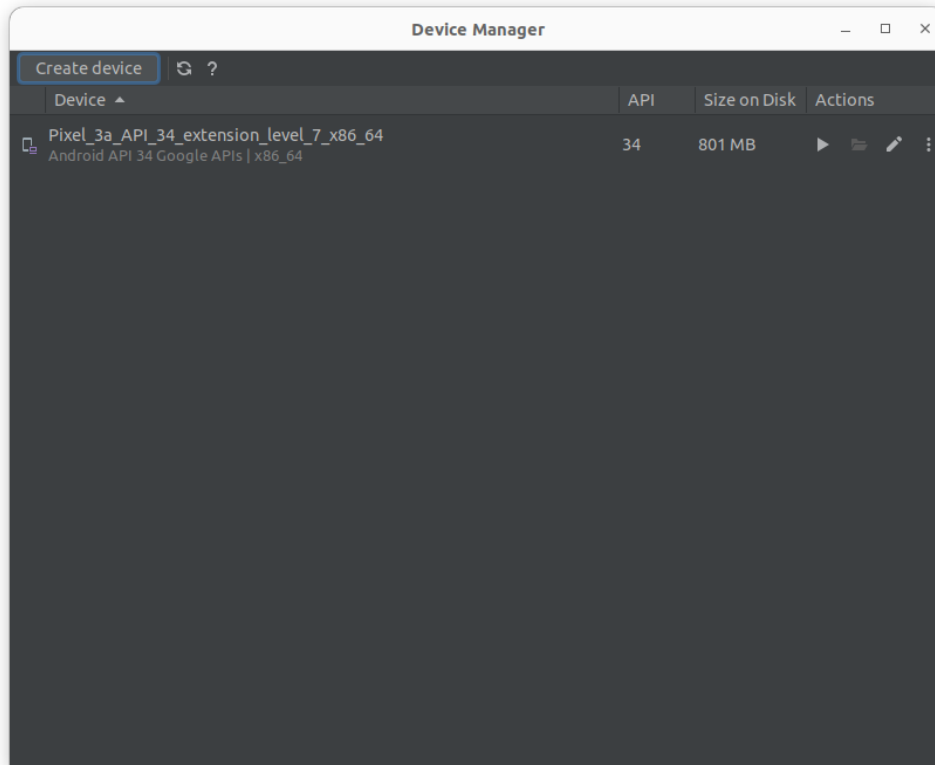
Như vậy là xong

### **Tạo máy ảo android**

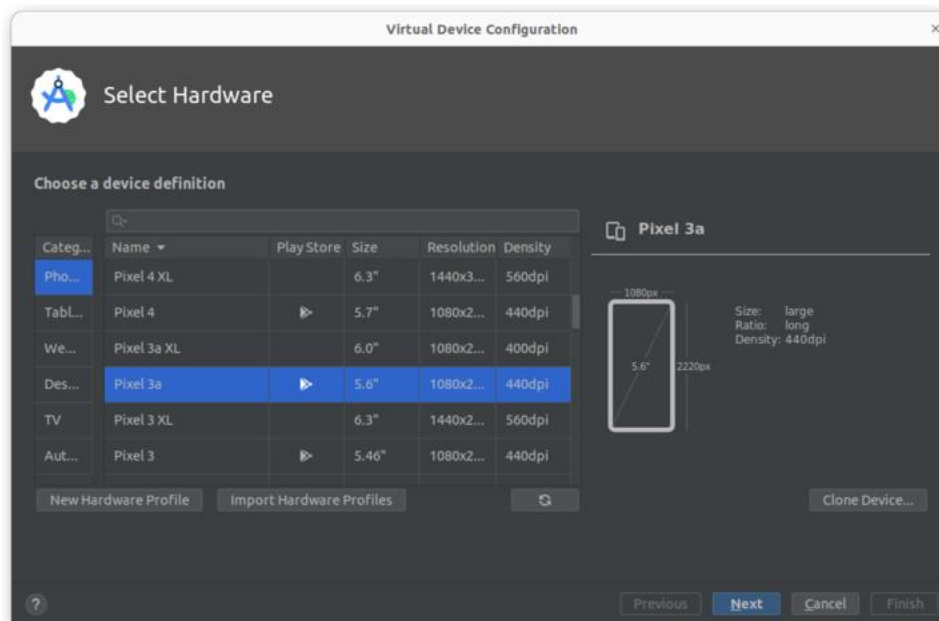
Tiếp theo chúng ta sẽ tạo máy ảo. Bấm More Actions -> Virtual Device Manager



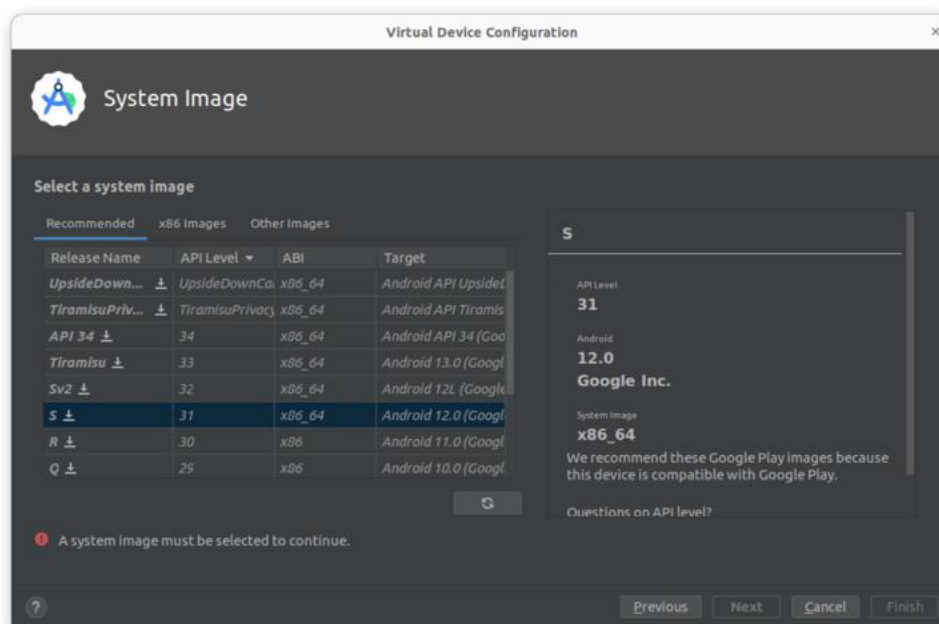
Tiếp tục chúng ta bấm Create device để tạo thiết bị



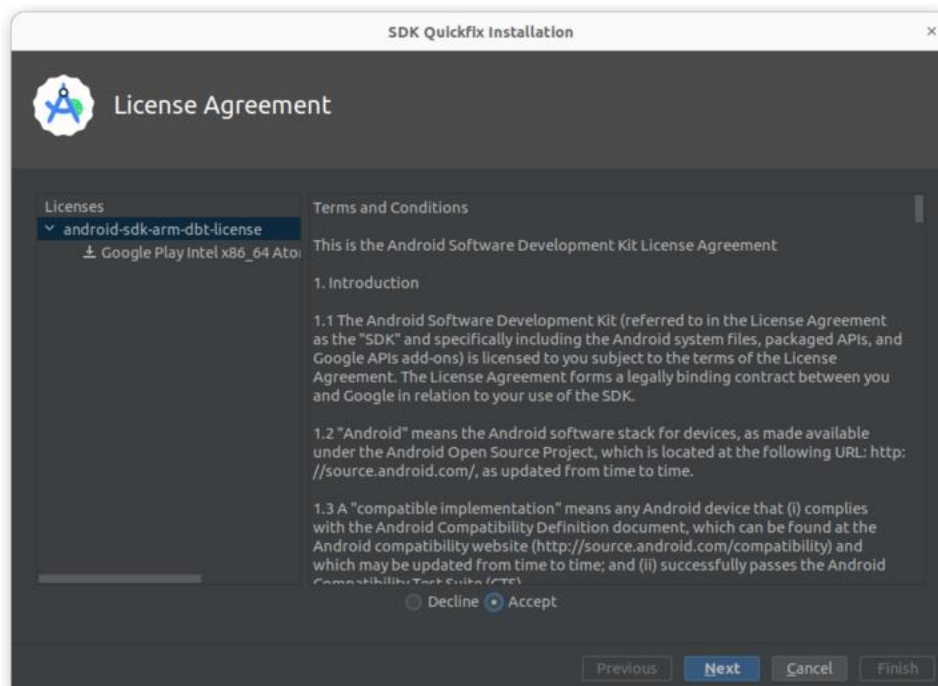
Chọn thiết bị Pixel 3a



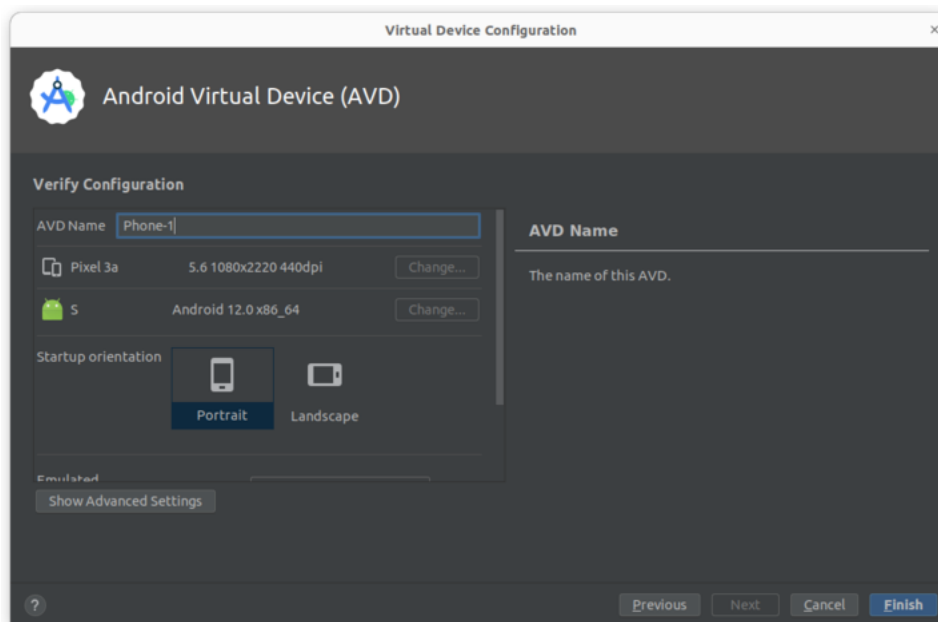
Chọn system image là S. Hiện tại chúng ta không có bấm vào dấu mũi tên xuống để tải



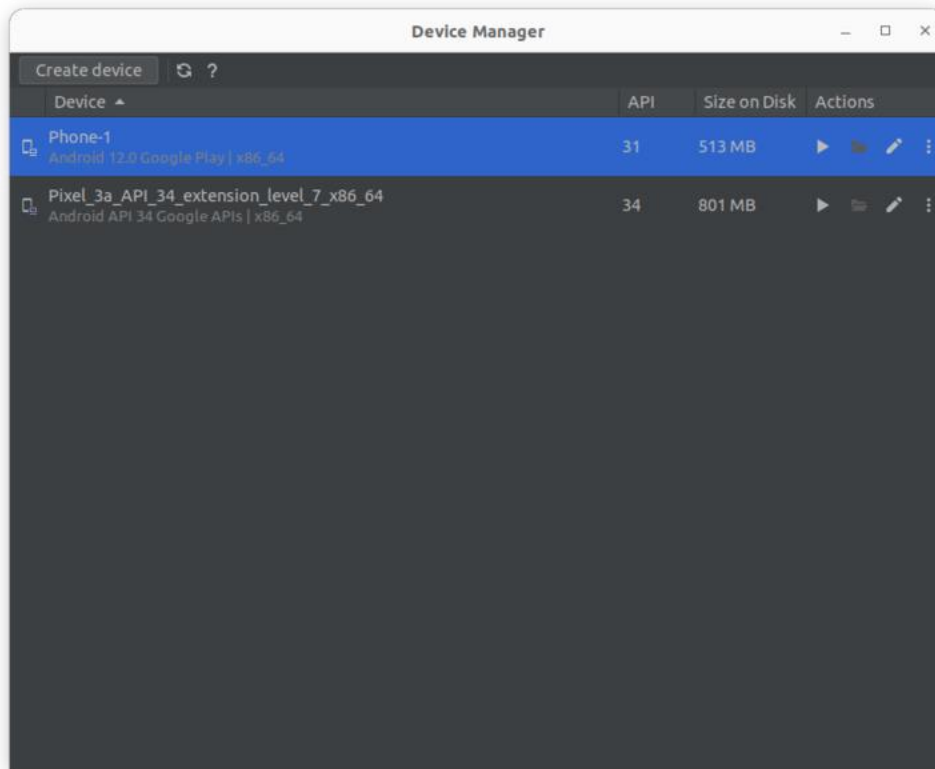
Bấm 'Accept' và 'Next' để cài đặt



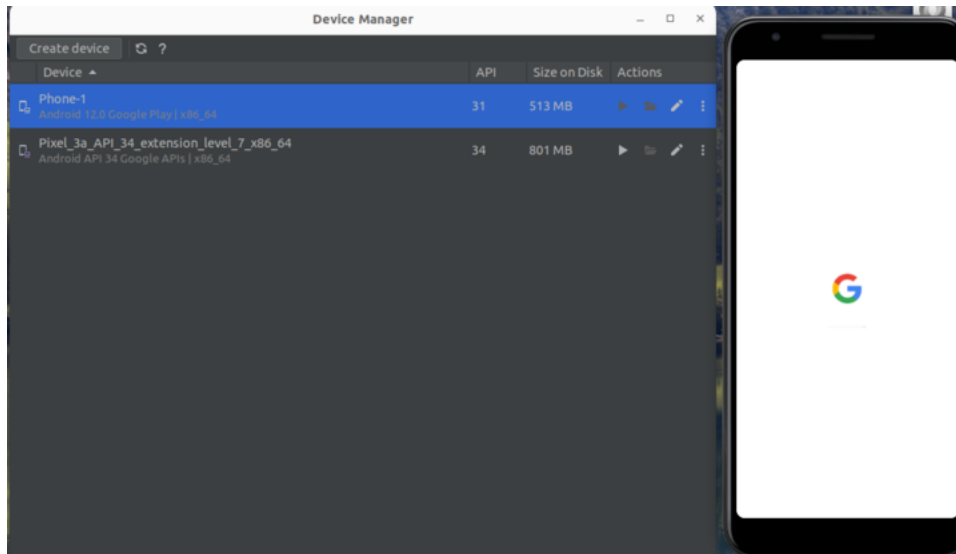
Sau đó, đặt tên thiết bị. Ở đây là 'Phone 1' và bấm 'Finish'



Như vậy là đã tạo thành công. Bấm nút 'Play' (mũi tên trỏ qua phải dưới chữ Actions) để khởi chạy máy ảo.



Như vậy là đã tạo thành công thiết bị ảo



## Cài Flutter SDK

Vào [đây](#), để tải Flutter SDK version 3.7.7

## Stable channel (Linux)

Select from the following scrollable list:

<a href="#">3.7.11</a>	x64	f72efea	4/13/2023	2.19.6	<a href="#">3.7.11 file</a>
<a href="#">3.7.10</a>	x64	4b12645	4/6/2023	2.19.6	<a href="#">3.7.10 file</a>
<a href="#">3.7.9</a>	x64	62bd795	3/31/2023	2.19.6	<a href="#">3.7.9 file</a>
<a href="#">3.7.8</a>	x64	90c64ed	3/23/2023	2.19.5	<a href="#">3.7.8 file</a>
<a href="#">3.7.7</a>	x64	2ad6cd7	3/9/2023	2.19.4	<a href="#">3.7.7 file</a>
<a href="#">3.7.6</a>	x64	12cb4eb	3/2/2023	2.19.3	<a href="#">3.7.6 file</a>
<a href="#">3.7.5</a>	x64	c07f788	2/23/2023	2.19.2	<a href="#">3.7.5 file</a>

Sau đó giải nén. Mở file `.zshrc` hoặc `.bashrc` để cập nhật môi trường

```
export PATH=$PATH:/home/vinh/Downloads/flutter_linux_3.7.7-stable/flutter/bin
```

Trong đó `/home/vinh/Downloads` là đường dẫn đến thư mục giải nén hồi nãy

Tiếp theo chạy dòng lệnh dưới đây để cài các công cụ hỗ trợ cho Flutter SDK

```
sudo apt-get install clang cmake ninja-build pkg-config libgtk-3-dev liblzma-dev
```

Sau khi cài xong chạy lệnh dưới đây để kiểm tra bổ sung những thứ cần thiết cho Flutter

```
flutter doctor
```

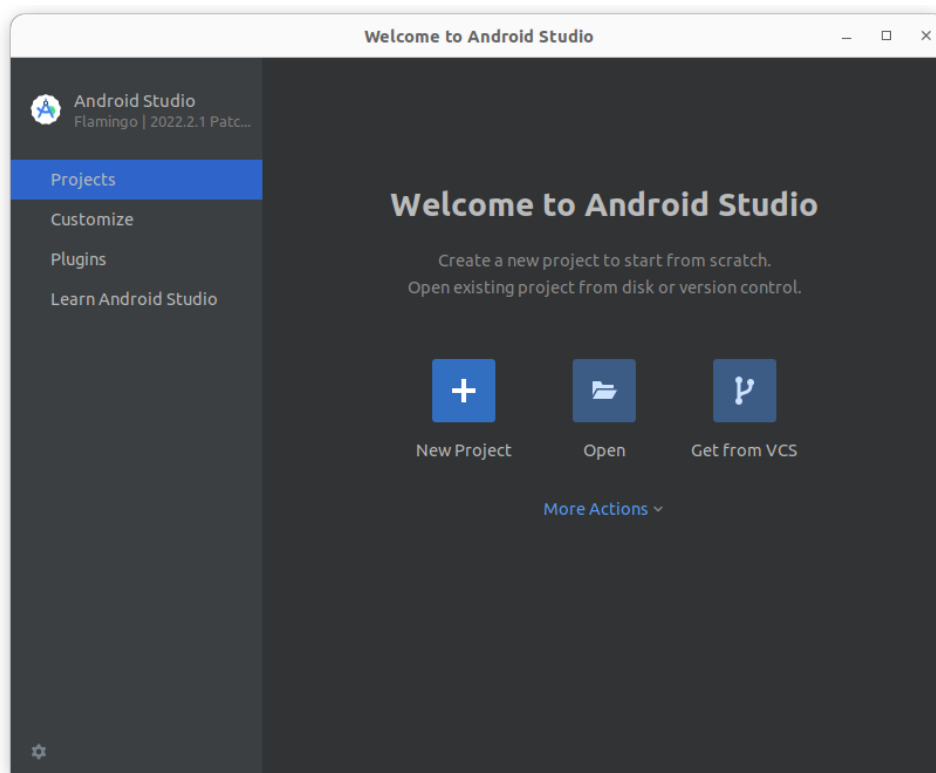
```
1: ~ x *
Google Privacy Policy (https://policies.google.com/privacy).

You have received two consent messages because the flutter tool is migrating to
a new analytics system. Disabling analytics collection will disable both the
legacy and new analytics collection systems. You can disable analytics reporting
by running 'flutter --disable-telemetry'

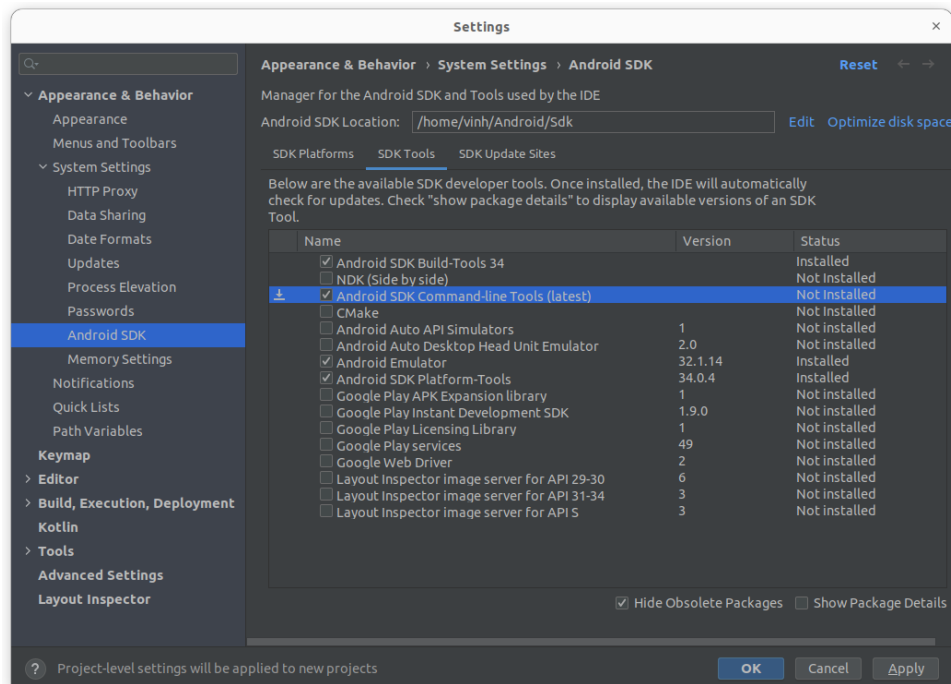
Flutter assets will be downloaded from https://storage.googleapis.com. Make sure
you trust this source!
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.10.5, on Ubuntu 22.10 5.19.0-46-generic, locale
en_US.UTF-8)
[!] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
    × cmdline-tools component is missing
      Run 'path/to/sdkmanager --install "cmdline-tools;latest"'
      See https://developer.android.com/studio/command-line for more details.
    × Android license status unknown.
      Run 'flutter doctor --android-licenses' to accept the SDK licenses.
      See https://flutter.dev/docs/get-started/install/linux#android-setup for
more details.
[✓] Chrome - develop for the web
[✓] Linux toolchain - develop for Linux desktop
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.80.0)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.
```

Hiện chúng ta thiếu Android toolchain. Chúng ta quay lại Android Studio. Chọn More Actions - > SDK manager



Sau đó chọn như hình bên dưới . Cài đặt Android SDK Command-line Tools (lastest)



Sau khi cài đặt xong thì chúng ta chạy lại lệnh flutter doctor để kiểm tra lại

```
1: ~ x +
flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.10.5, on Ubuntu 22.10 5.19.0-46-generic, locale en_US.UTF-8)
[!] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[✓] Chrome - develop for the web
[✓] Linux toolchain - develop for Linux desktop
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.80.0)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.
```

Như vậy là chúng ta đã hoàn thành việc cài đặt Flutter SDK

## Tải mã nguồn

Tải mã nguồn từ GitHub:

git clone <https://github.com/the-company-tcus/frontend.git>



Người dùng sẽ đăng nhập bằng tài khoản GitHub với username và với password là Personal Access Token (PAT) của người dùng.

Note: Lưu ý Personal Access Token của người dùng cần được cấp quyền repo. Ngoài ra, người dùng cần phải được nhóm cấp quyền truy cập vào GitHub Organization của nhóm.

### Sử dụng mã nguồn

1. Sau khi tải mã nguồn xong chúng ta vào thư mục của mã nguồn (frontend).
2. Chạy lệnh dưới đây để cập nhật thư viện

Flutter pub get

3. Bấm vào [đây](#) để tải các thư mục cần thiết để chạy ứng dụng
4. File upload-keystore.jks và google-services.json bỏ vào thư mục android/app
5. File local.properties bỏ vào thư mục android
6. Bật thiết bị android (Thiết bị chúng ta đã bật khi này)
7. Chạy lệnh dưới đây để chạy mã nguồn

Flutter run --release

8. Terminal sẽ hỏi chọn thiết bị nào. Ta chọn thiết bị cần để chạy mã nguồn