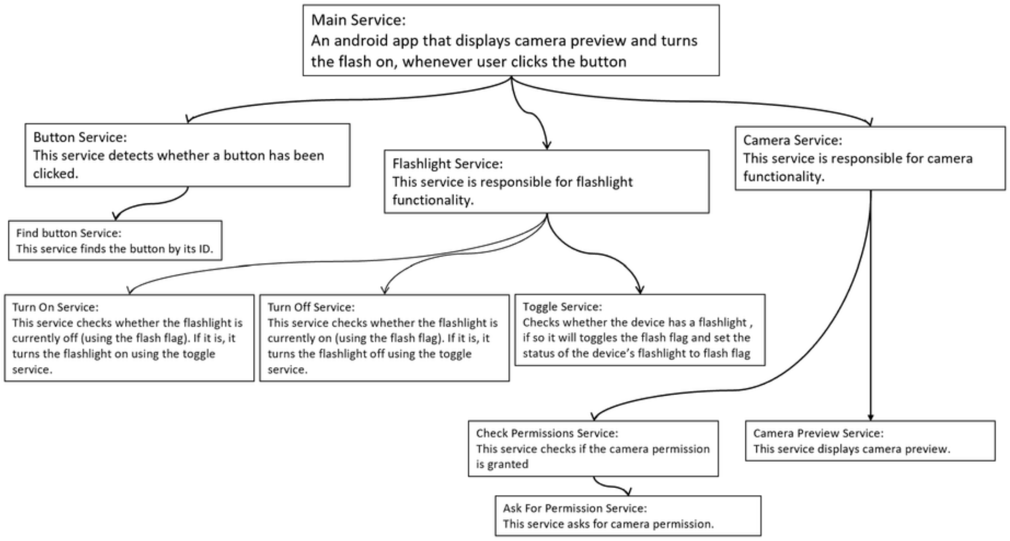
Software Architecture

I tried to implement this project, considering SOLID and clean code principles, and using microservices.

Microservices:


Main Service: An android app that displays camera preview and turns the flash on, whenever user clicks the button
Button Service: This service detects whether a button has been clicked.
Flashlight Service: This service is responsible for flashlight functionality.
Camera Service: This service is responsible for camera functionality.
Find button Service: This service finds the button by its ID.
Turn On Service: This service checks whether the flashlight is currently off (using the flash flag). If it is, it turns the flashlight on using the toggle service.
Turn Off Service: This service checks whether the flashlight is currently on (using the flash flag). If it is, it turns the flashlight off using the toggle service.
Toggle Service: Checks whether the device has a flashlight , if so it will toggles the flash flag and set the status of the device's flashlight to flash flag
Check Permissions Service: This service checks if the camera permission is granted
Camera Preview Service: This service displays camera preview.
Ask For Permission Service: This service asks for camera permission.

SOLID:

The principles associated with this project are Single Responsibility and Open/Closed:
Single Responsibility:

This app consists of two classes and many functions, each of which has only one responsibility.

1. **HomeScreen** Class: This class is only responsible for functionalities within the Home Screen.
   a. `initializeViews`: This function is responsible for initializing the views within the Home Screen
   b. `getCamera`: This function is responsible for getting the back camera's ID.
   c. `setupListeners`: This function is responsible for executing a certain activity when a certain button is clicked.
   d. `turnOn`: This function is responsible for turning the flashlight on when the flashlight is currently off, using the toggle method.
   e. `turnOff`: This function is responsible for turning the flashlight off when the flashlight is currently on, using the toggle method.
   f. `toggle`: This function is responsible for setting the flashlight mode to the opposite of the current mode.
   g. `startCameraPreviewActivity`: This function is responsible for starting the camera screen activity.
2. **CameraScreen Class:** This class is only responsible for functionalities within the Camera Screen.
   a. `initializeViews`: This function is responsible for initializing the views within the Camera Screen.
   b. `getPermission`: This function is responsible for requesting camera permissions when the permissions are not yet granted.
   c. `hasPermission`: This function is responsible for checking if the required camera permissions are granted.
   d. `startCamera`: This function is responsible for starting the camera, and then executing a callback with the camera object.
   e. `setupFlashlightButton`: This function is responsible for executing toggle activity when the flashlight button is clicked.
   f. `toggle`: This function is responsible for setting the flashlight mode to the opposite of the current mode.

Open Closed:

The classes and functions are closed for modification but open for extension.

Clean Code:

- Names:
  - They are meaningful and intention revealing.
  - They make meaningful distinctions.
  - They are pronounceable.
  - Nouns are used for class names.
  - Verbs are used for function names.
- Functions:
  - They have only one responsibility.
  - They have the smallest number of arguments possible.
- Formatting:
  - Concepts are vertically separated.
  - Related lines of code are vertically dense.
  - Dependent functions are close.
  - Indentations are used.
- Commenting:
  - However, the code speaks for itself; relevant comments are added.
  - Comments are not misleading.