

Projeto

Definição

O PicNotes é um aplicativo desenvolvido para Android. Seu objetivo é auxiliar anotações durante aulas e palestras. Para isso, ele permite que o usuário tire fotos, e então desenhar sobre elas. Esta pode então ser salva e compartilhada.

User Story

Como um estudante

Para que eu possa registrar o conteúdo das aulas

Eu gostaria de fazer anotações em cima de fotos tiradas da lousa

Esta é a funcionalidade mais básica do programa. Ele utiliza a funcionalidade da câmera padrão do Android, salva a imagem em um diretório do próprio aplicativo, e o carrega para um *Canvas* mostrado na tela.

Atualmente, as anotações feitas são apenas da cor azul, por ainda não ter sido implementado um menu para trocar de cor.

Para compartilhar a imagem, o usuário pode utilizar o menu. Este é acessado ao passar o dedo da esquerda para a direita sobre o canto esquerdo do aplicativo. Neste menu, há a opção de Compartilhar, e ao clicar nela, aparecerá um menu com todos os aplicativos instalados que podem receber a imagem.

Implementação

O código do projeto pode ser acessado em
<https://github.com/GianAwesome/PicNotes>

O projeto foi feito utilizando o Android Studio, tendo foco na versão 5.0 do Android (Lollipop).

Exemplo

the resulting dialog is shown in figure 1.

Optionally, you can set some standard extras for the intent: `EXTRA_EMAIL`, `EXTRA_SUBJECT`. If an application is not designed to use them, it simply ignores them.

Note: Some e-mail applications, such as Gmail, expect a `String[]` for extra email addresses. You should add these to your intent.

Send Binary Content

Binary data is shared using the `ACTION_SEND` action combined with setting the `EXTRA_STREAM`. This is commonly used to share an image but can be used to share any binary data.

```
Intent shareIntent = new Intent();
shareIntent.setAction(Intent.ACTION_SEND);
shareIntent.putExtra(Intent.EXTRA_STREAM, uriToImage);
shareIntent.setType("image/jpeg");
startActivity(Intent.createChooser(shareIntent, getResources().getText(R.string.send_image)));
```

Note the following:

- You can use a MIME type of `"*/*"`, but this will only match activities that are able to handle any type of data.
- The receiving application needs permission to access the data the `Uri` points to.
 - Store the data in your own `ContentProvider`, making sure that other apps have access. A common mechanism for providing access is to use `per-URI` permissions which are temporary.
 - Use the system `MediaStore`. The `MediaStore` is primarily aimed at video, audio, and images. At API level 11 it can also store non-media types (see `MediaStore.Files` for more information). After which a `content://` style `Uri` suitable for sharing is passed to the provider.
- Use the system `MediaStore`. The `MediaStore` is primarily aimed at video, audio, and images. At API level 11 it can also store non-media types (see `MediaStore.Files` for more information). After which a `content://` style `Uri` suitable for sharing is passed to the provider.

system `MediaStore` the content is accessible to any app on the device.

Send Multiple Pieces of Content

To share multiple pieces of content, use the `ACTION_SEND_MULTIPLE` action together with the `EXTRA_STREAMS` extra. The extra should be a `Uri[]` array of `Uri` objects. For example, if you share 3 JPEG images, the MIME type should be `"image/*"` to match an activity that handles any type of image. You should also store your preferences for site-specific language and display options.

UML

