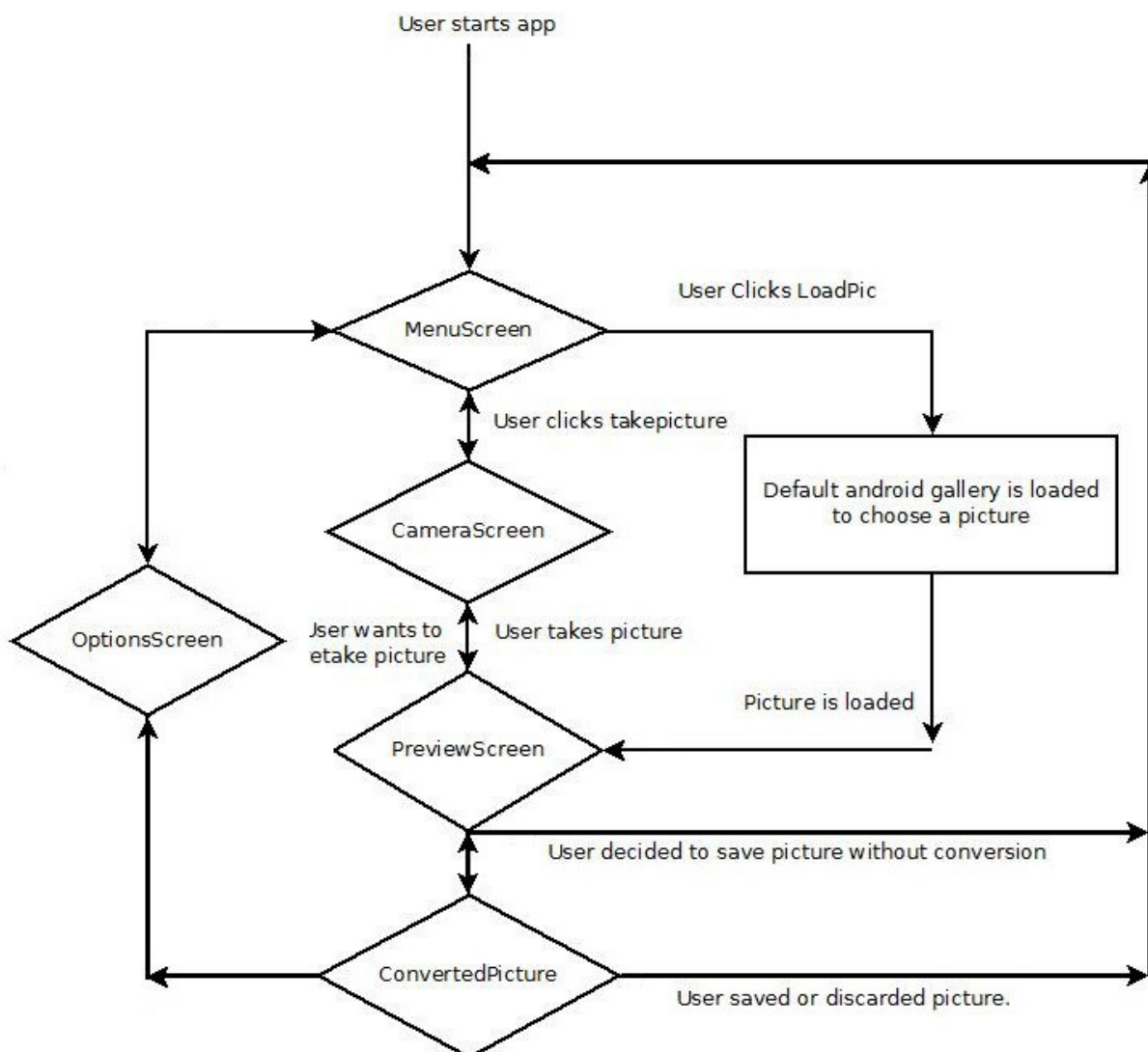# Developer manual Asciicam

Asciicam is an application developed mainly for android phones running version 2.2 or above. In order to start developing from this project, please do the following:

- Clone workspace from https://github.com/Ossianm/Grupp29Asciicam .

- Install development software Eclipse, Android SDK, Android AVD (For testing), Java SE6 and all android addons for eclipse.

- Create a new testproject for Grupp29Asciicam named AsciicamTest and copy all inside test folder to the project.
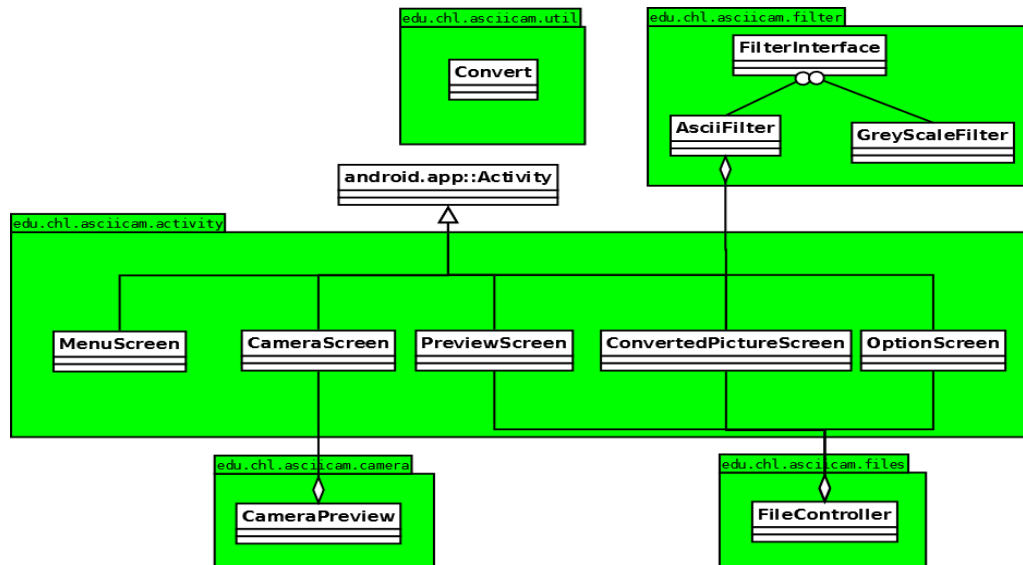
Application flow

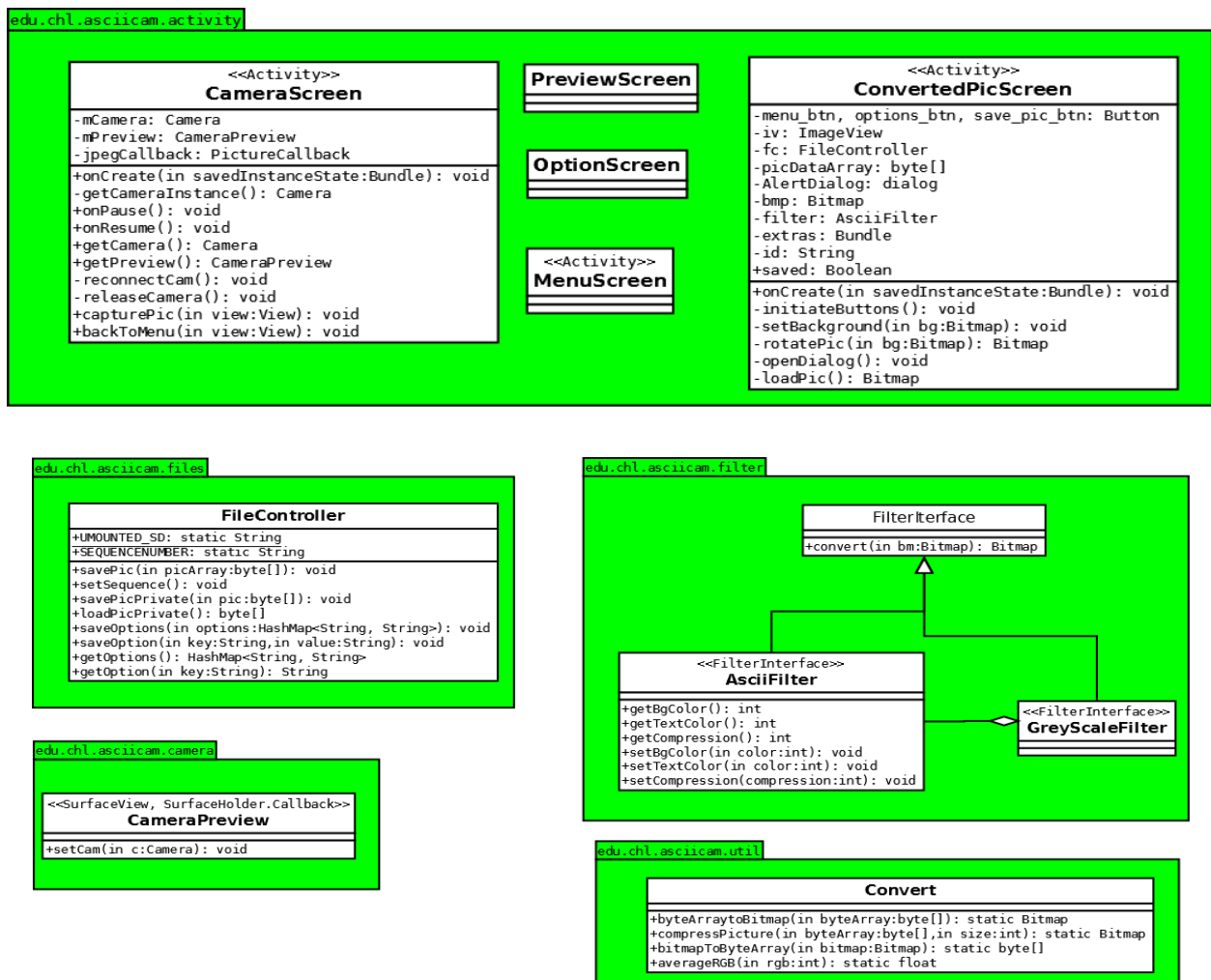Below is a flowchart diagram which describes how the application is intended to work for a user:

All diamond cells are intended to be Activities using android API or custom made APIs to perform tasks.

Class Model

UML for project:

Packages in the project:

edu.chl.asciicam.activity

```
<<Activity>>
CameraScreen
-mCamera: Camera
-mPreview: CameraPreview
-jpegCallback: PictureCallback
+onCreate(in savedInstanceState:Bundle): void
-getCameraInstance(): Camera
+onPause(): void
+onResume(): void
+getCamera(): Camera
+getPreview(): CameraPreview
-reconnectCam(): void
-releaseCamera(): void
+capturePic(in view:View): void
+backToMenu(in view:View): void
```

```
PreviewScreen
```

```
OptionScreen
```

```
<<Activity>>
MenuScreen
```

```
<<Activity>>
ConvertedPicScreen
-menu_btn, options_btn, save_pic_btn: Button
-iv: ImageView
-fc: FileController
-picDataArray: byte[]
-AlertDialog: dialog
-bmp: Bitmap
-filter: AsciiFilter
-extras: Bundle
-id: String
-saved: Boolean
+onCreate(in savedInstanceState:Bundle): void
-initiateButtons(): void
-setBackground(in bg:Bitmap): void
-rotatePic(in bg:Bitmap): Bitmap
-openDialog(): void
-loadPic(): Bitmap
```

edu.chl.asciicam.files

```
FileController
+UMOUNTED_SD: static String
+SEQUENCENUMBER: static String
+savePic(in picArray:byte[]): void
+setSequence(): void
+savePicPrivate(in pic:byte[]): void
+loadPicPrivate(): byte[]
+saveOptions(in options:HashMap<String, String>): void
+saveOption(in key:String,in value:String): void
+getOptions(): HashMap<String, String>
+getOption(in key:String): String
```

edu.chl.asciicam.camera

```
<<SurfaceView, SurfaceHolder.Callback>>
CameraPreview
+setCam(in c:Camera): void
```

edu.chl.asciicam.filter

```
FilterInterface
+convert(in bm:Bitmap): Bitmap
```

```
<<FilterInterface>>
AsciiFilter
+getBgColor(): int
+getTextColor(): int
+getCompression(): int
+setBgColor(in color:int): void
+setTextColor(in color:int): void
+setCompression(compression:int): void
```

```
<<FilterInterface>>
GreyScaleFilter
```

edu.chl.asciicam.util

```
Convert
+byteArraytoBitmap(in byteArray:byte[]): static Bitmap
+compressPicture(in byteArray:byte[],in size:int): static Bitmap
+bitmapToByteArray(in bitmap:Bitmap): static byte[]
+averageRGB(in rgb:int): static float
```

As we can see in the chart, the intention is to create 3 APIs, one for camera, one for filehandling and one for the filter. The rest of the coding should be handled by public APIs in the activities.

Testing

It is really important to create a new android test project for testing.

In Eclipse choose New --> Other --> Android --> Android Test Project and name it AsciicamTest.

The source code for testing is included in github reopsitory but it will not compile and run without being imported to a test project. It is also important to note that a AVD (Android Virtual Device) or a phone connected by USB is required to run the test project.