InstaLike - Documentation

Project Description:

- Capture Image from camera and display images captured, filtering images (Blur, Grayscale), filtering all images in directory, add image Info (Title, Caption, CaptureDate, Favorite)
- This Project Include c++, qml (Qt), Cmake, Compiler MSVC

Project API:

ImageProcessor Class

Header	**#include <imageprocessor.h>**
Inherits	**QThread**

Public Functions

QString	**processImage()**
QStringList	**FileNames()**
void	**setProgress**(float newProgress)
void	**setProcessImage**(*const* QString& path)
void	**setFilterType**(QString filterType)
void	**setFilterValue**(float NewValue)
void	**qImageSave**(QString path, QString id)
void	**setFolderPath**(QString path)
void	**applyToall**(QString saveDirectory)

const QString &ImageProcessor::**processImage**() :

Return Image in form of QUrl to display.

const QStringList &ImageProcessor::**fileNames**() *const* :

Return List of Images in the directory (~ ListDir)

```
void ImageProcessor::setProcessImage( const QString& path)
```

convert QString image Path to QImage and Process image to Blur, Grayscale or Original

void ImageProcessor::**setFilterType**(QString filterType)

set filter type. Types : "Blur", "Gray", "Original"

void ImageProcessor::**setFilterValue**(float NewValue)

set filter value. Used for bluring.

void ImageProcessor::**qImageSave**(QString m_savePath, QString id)

save Image with savePath and the id. Image will be saved under name of id.

void ImageProcessor::**setFolderPath**(QString path)

set folder path to show Images.

```
void ImageProcessor::setProgress(float newProgress)
```

Used in Multithreading, to set new progress

void ImageProcessor::**applyToall**(QString saveDirectory)

Filter all images in the folder.

Jsonates Class

Header	**#include <jsonates.h>**
Inherits	**QObject**

Public Functions

QJsonObject	**readJson**(QString filePath)
QString	**imgTitle** ()
QString	**imgCaption**()
QString	**captureDate**()
QString	**imgPath** ()
bool	**favorite**()
int	**getId**()
void	**setId**()
void	**setImgTitle** (*const* QString &newImgTitle)
void	**setImgCaption** (*const* QString &newImgCaption)
void	**setCaptureDate** (*const* QString &newCaptureDate)
void	**setFavorite** (bool newFavorite)
void	**checkImgInfo**()
void	**setJson**(int my_id);

QJsonObject JSONates::**readJson**(QString **filePath**)

Read Json file and return json object

void JSONates::**setId()**

Generate a random 64 bit number

void JSONates::**setJson**(int **my_id**)

Create Json file if it doesn't exist. Or read json file if exist and append next informations to the file.

void JSONates::**setSavefile**(QString **path**)

set Json save file location.

void JSONates::**setImgTitle**(*const* QString &**newImgTitle**)

set Image title for Informations.

void JSONates::**setImgCaption**(*const* QString &**newImgCaption**)

set Image Caption for Inforamtions.

void JSONates::**setCaptureDate**(*const* QString &**newCaptureDate**)

set Capture Date for Informations.

void JSONates::**setFavorite**(bool **newFavorite**)

set if Image is Favorite or not.

const QString &JSONates::**imgPath**() *const*

Return Image Path.

void JSONates::**setImgPath**(*const* QString &**newImgPath**)

set image Path to be saved in.

void JSONates::**checkImgInfo**()

Read json file and create json Array

QString &JSONates::**imgTitle**()

Return Image Title.

QString &JSONates::**imgCaption**()

Return Image Caption.

QString &JSONates::**captureDate**()

Return capture date.

bool JSONates::**favorite**() *const*

Return if the Image is Favorite or not.