Emojo – Instagram photo emotion analyzer

Emojo is a WPF/Android app that allows users to analyze emotions on their recent Instagram photos in general or on each photo separately.

Github: https://github.com/peramor/Emojo

Team:

Roman Maltsev – Android UI/Logic integration

Ivan Kamakin – Backend logic and API communication

Mikhail Zhilyakov – Database setup and maintenance, custom API

Schmygalyova Evgenia – WPF UI/Logic Integration

Code overview

Logic classes – Emojo.Lib

Models:

Models.InstagramPhoto – an intermediate model for initial media data from Instagram.

Models.User – a model for user data from Instagram.

Models.Photo – a final model for media data, complete with properties for emotion ratings.

ViewModels.OAuthBuildModel – intermediate model for Instagram API access token building (Android specific)

API access:

Instagram.InstagramGetter and **Instagram.IInstagramGetter** – a class responsible for Instagram API access (getting user and media data) and its respective interface.

EmotionAPI.EmotionsAPIGetter and

EmotionAPI.IEmotionsAPIGetter – a class responsible for Microsoft Emotion API access (emotion recognition) and its respective interface.

Code overview

Logic classes – Emojo.Lib

General logic:

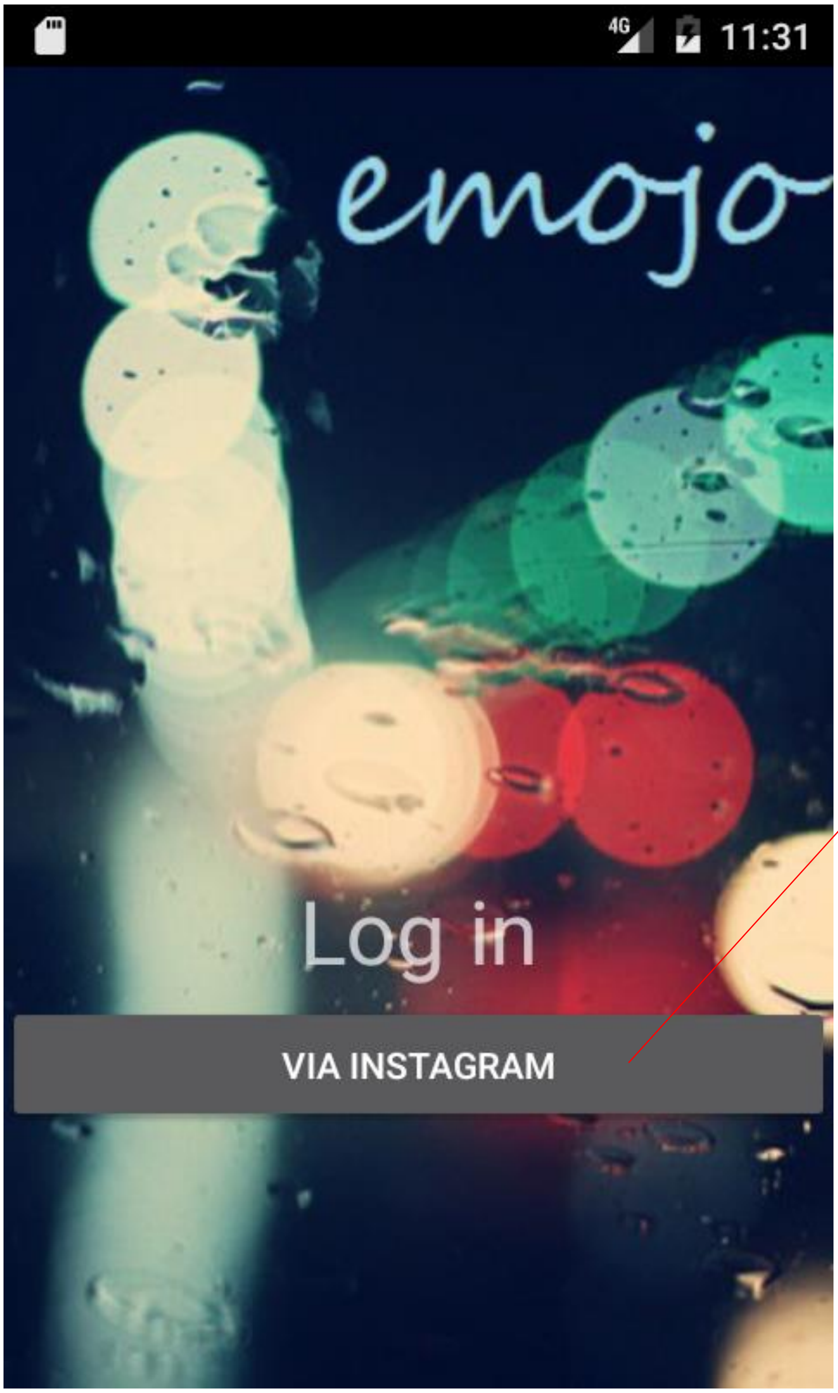
- **DB** – a class for custom database API access (sending queries to DB) and response processing.
- **Repository** – a class for final data storage. Loads data using API classes and DB class and works with it locally.
- **AndroidRepository** – Android-specific Repository wrapper utilizing OAuthBuildModel.
- **InterfaceFactory** – a factory class for interface building and potential fast implementation switching.

Code overview

UI classes – Emojo.Droid

LoginActivity:

Log in screen with a respective button.



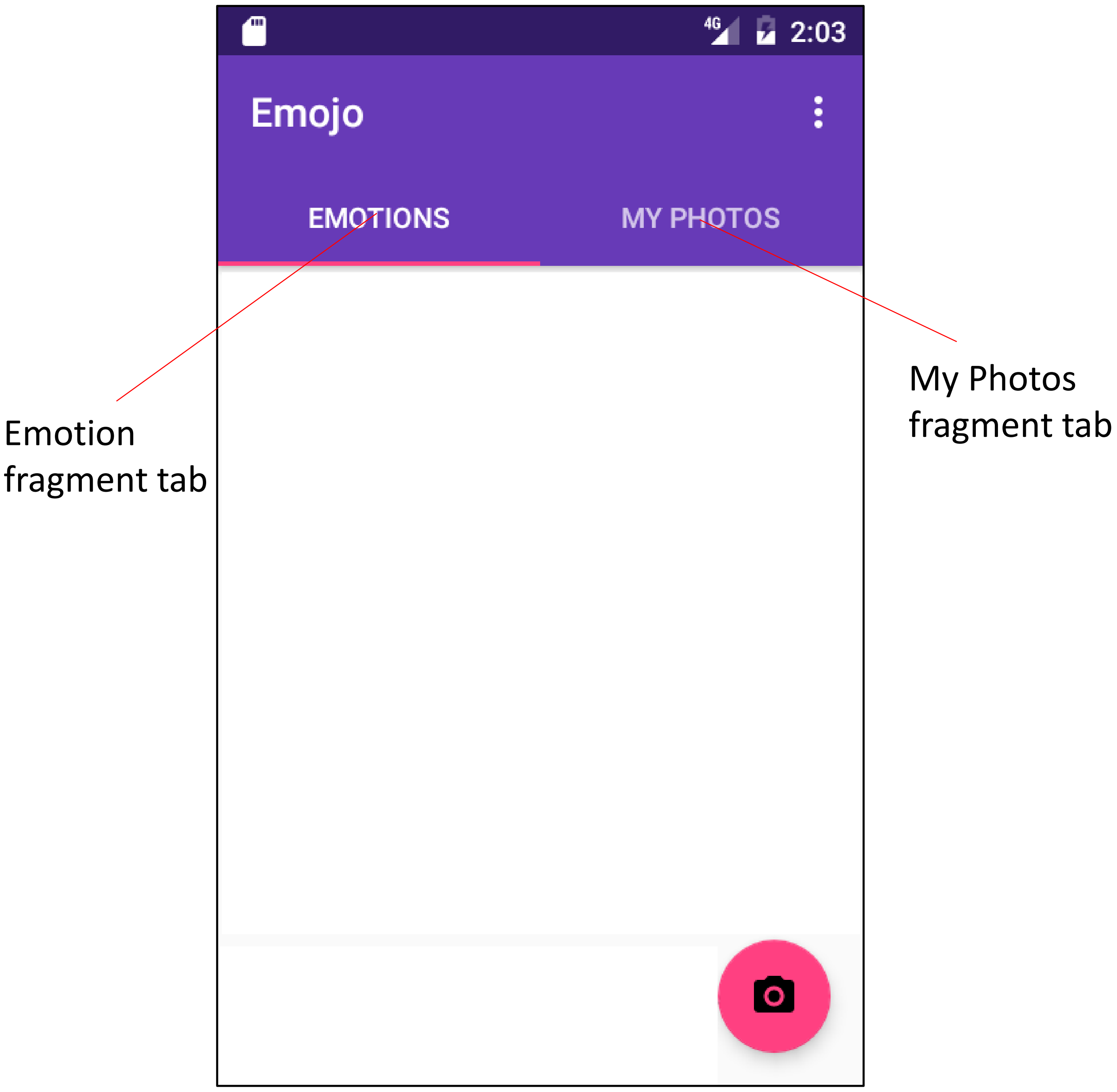
Login button

Code overview

UI classes – Emojo.Droid

MainActivity:

Main screen with a container of fragments which are switched using the tab bar.

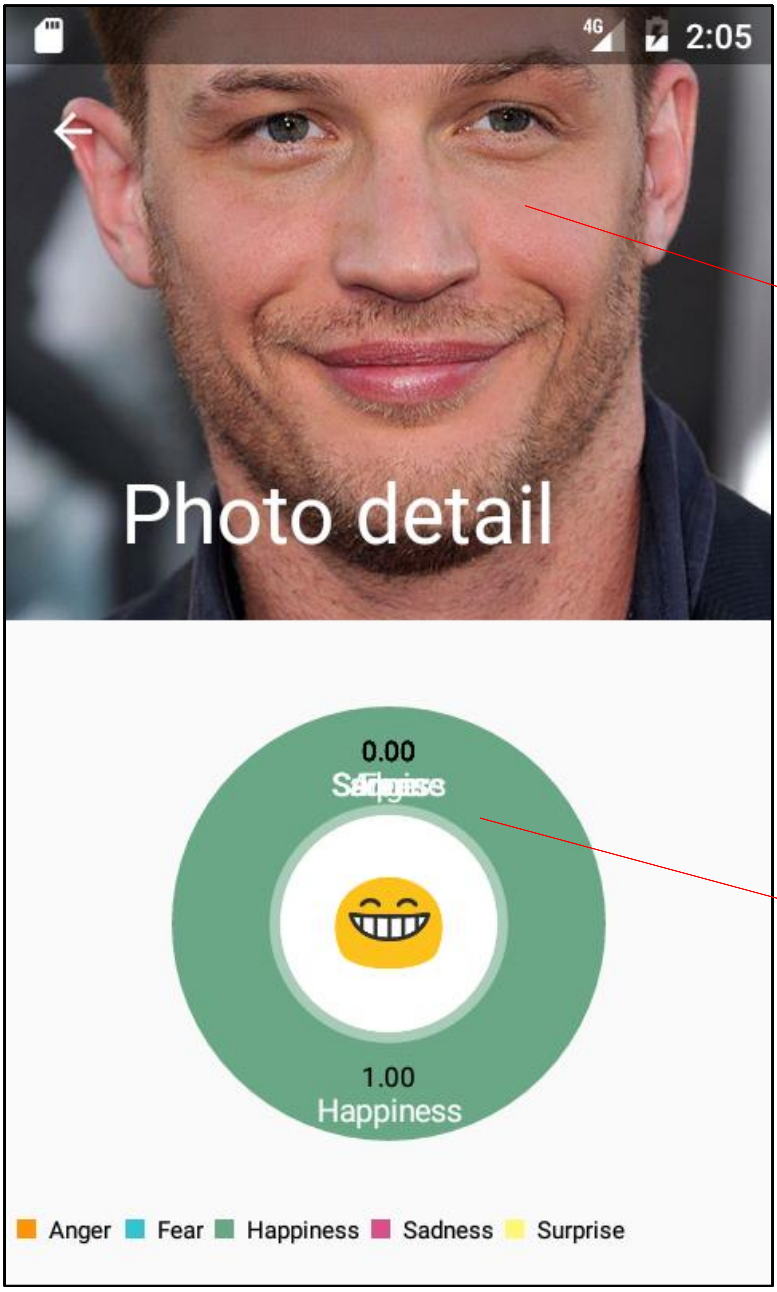


Code overview

UI classes – Emojo.Droid

AccountDetailActivity:

A screen which shows emotion statistics for a chosen photo, accessed from the “My Photos” fragment.



Chosen photo

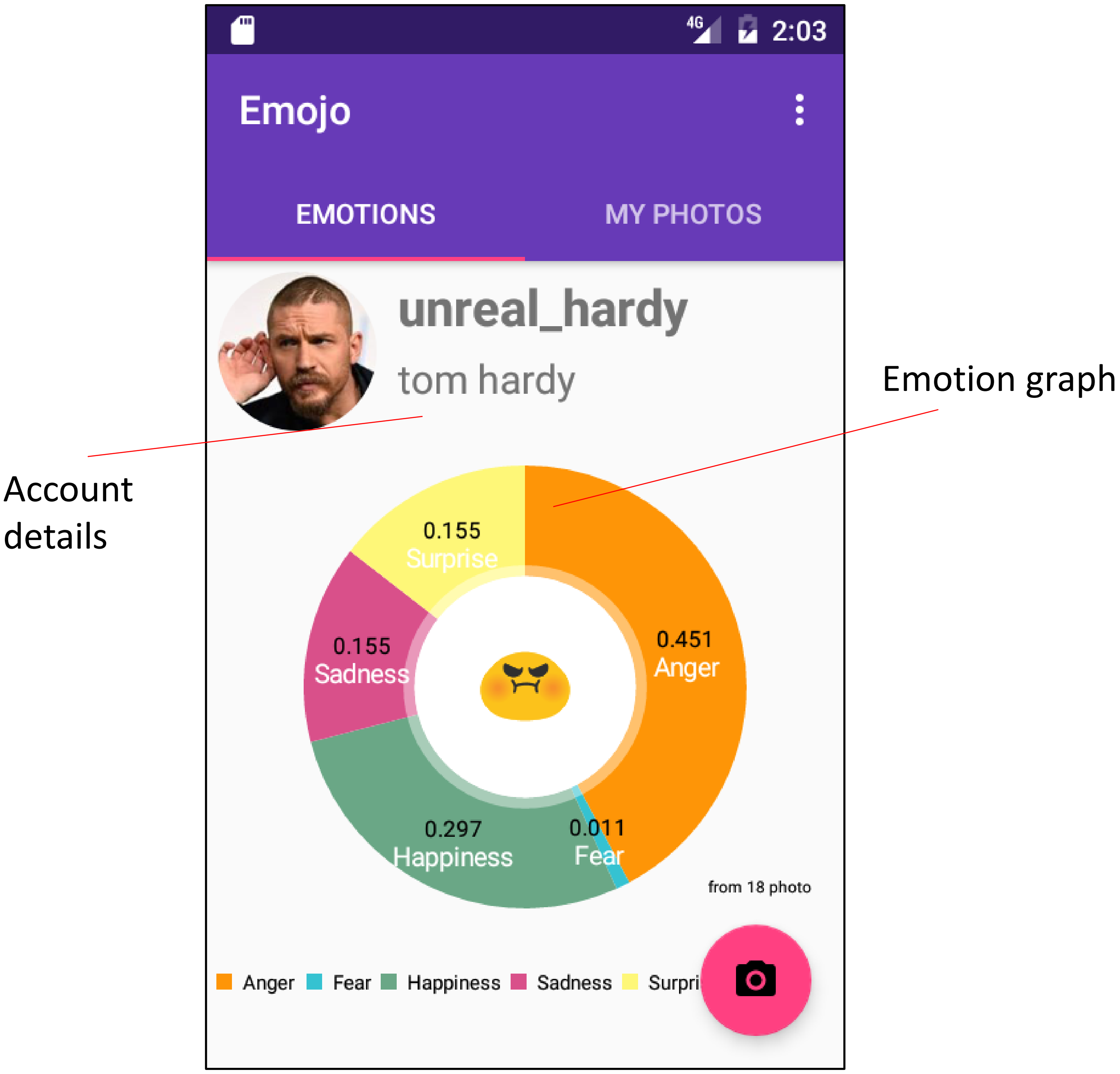
Emotion graph

Code overview

UI classes – Emojo.Droid

FragmentEmotions:

A tab bar fragment which shows emotion statistics over an entire profile.



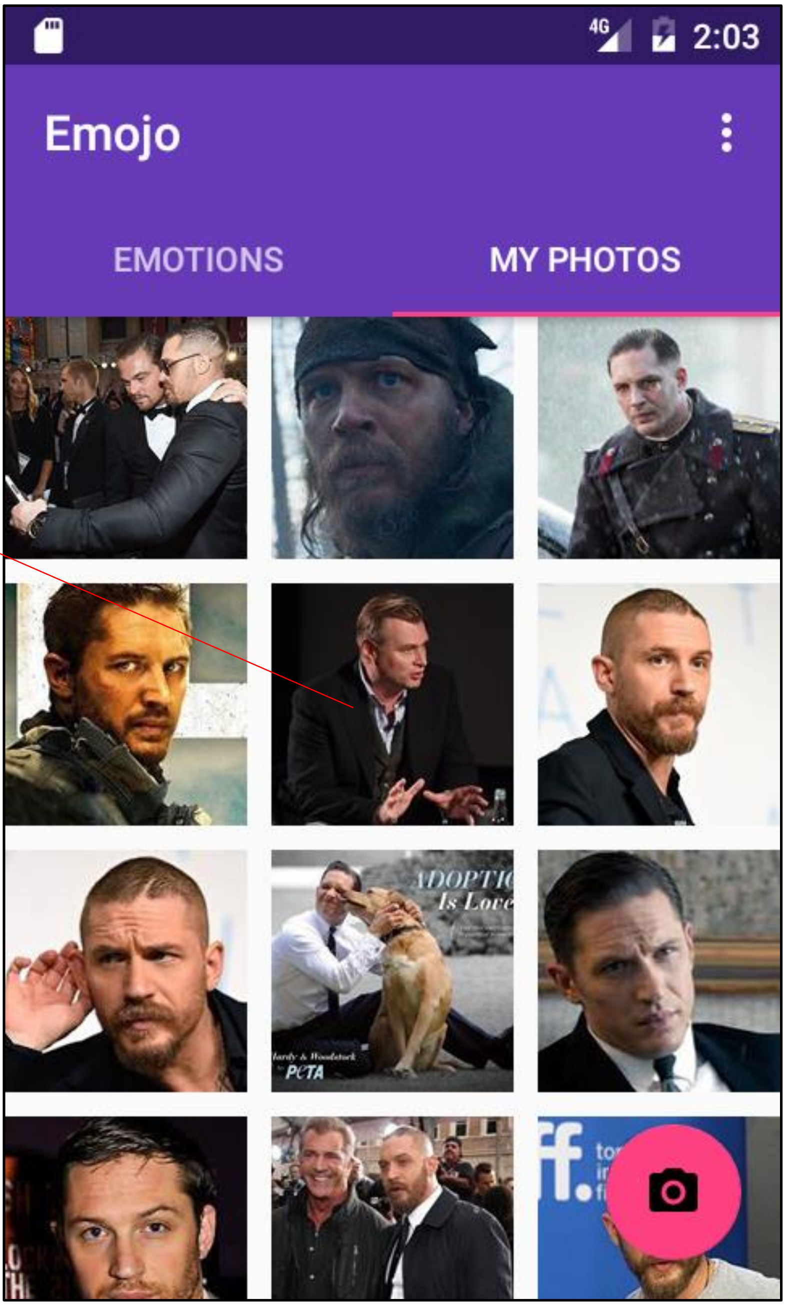
Code overview

UI classes – Emojo.Droid

FragmentMyPhoto:

A fragment with a container of user photos, where a specific photo can be chosen.

User photos



Code overview

UI classes – Emojo.Droid

BitmapHelpers – a helper Android-specific class for image handling.

ChartFactory – a factory for different types of charts.

IChartBuilder – an interface for chart classes.

LineChartBuilder – implements **IChartBuilder** for line graphs.

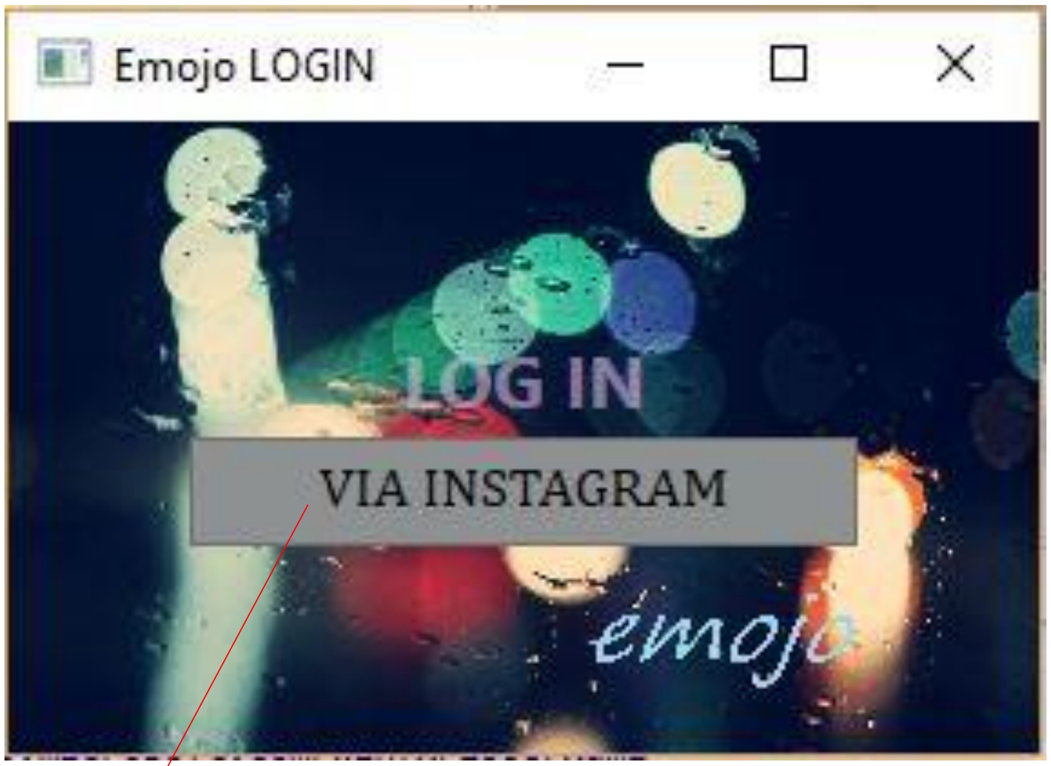
PieChartBuilder – implements **IChartBuilder** for pie graphs.

Code overview

UI classes – Emojo.WPF

MainWindow:

Starting window with a user login button. Calls other windows and acts as a main app entry point.



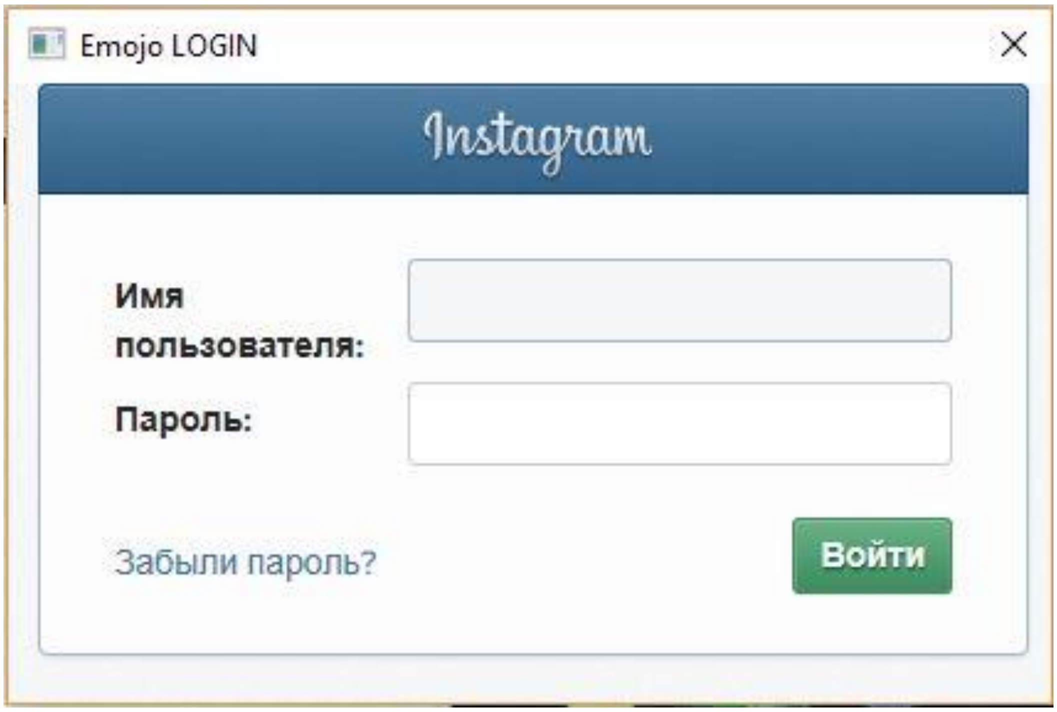
Login button

Code overview

UI classes – Emojo.WPF

LoginBrowserWindow:

A simple window with WebBrowser control. Direct an unauthorized user to Instagram login page, otherwise just proceeds directly to the Info window.



Code overview

UI classes – Emojo.WPF

FinalInfo:

The main window of an application, shows user photos, allows to pick a photo from a list. Has 3 donut graphs: emotion distribution over an entire photo set, emotion distribution on a chosen photo, and number of pictures with one person or many people. Has a logout button. Takes some time to load, during this time shows “Loading…” text.

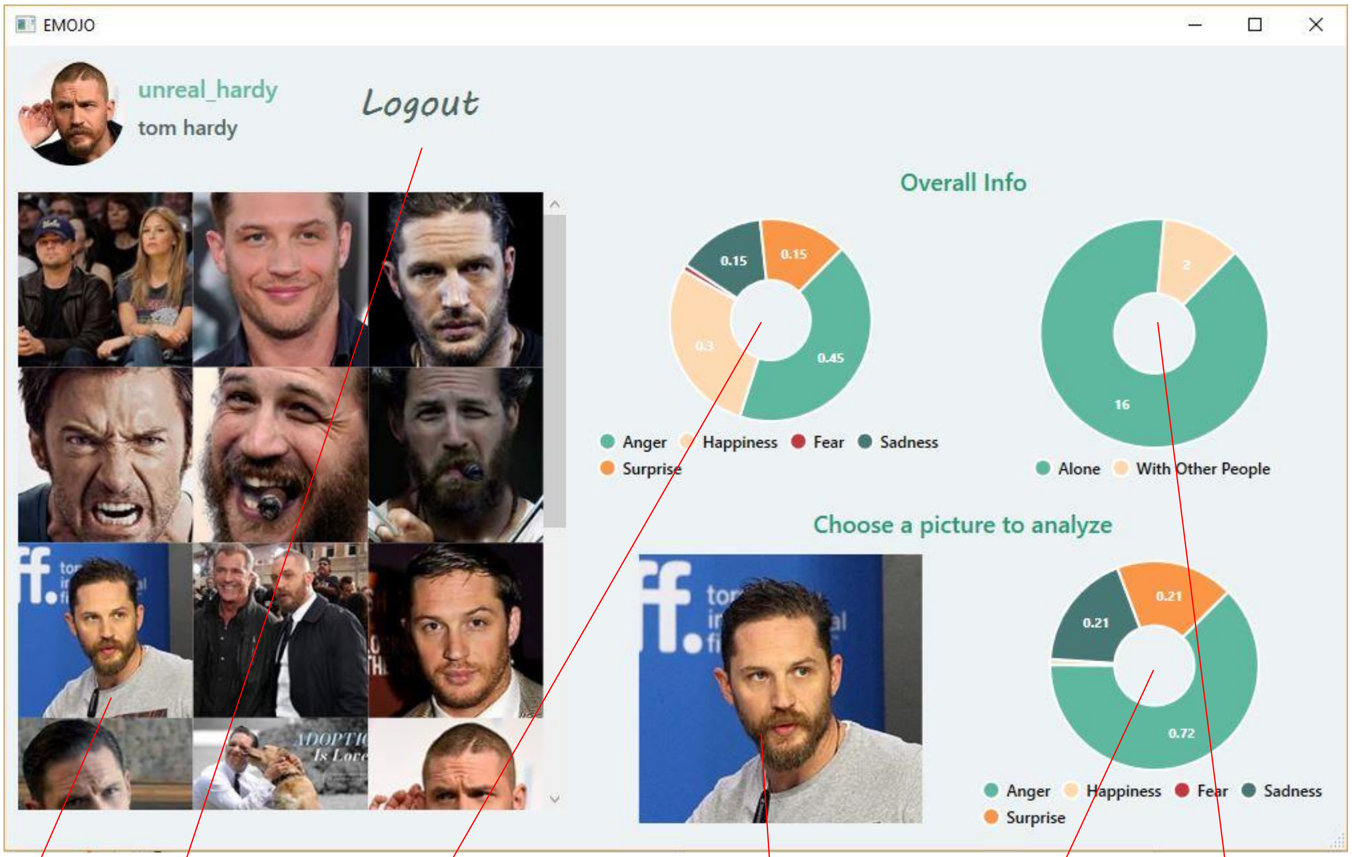


Photo list

Logout button

Emotion chart on a whole profile

Chosen picture

Chosen picture emotion chart

Number of people counts

Code overview

UI classes – Emojo.WPF

ImageDownloader – a helper WPF-specific class for asynchronous image downloading from a URL.

Additional comments

Due to Instagram app policy our app cannot be accessed by general users as of now. Instagram only allows access to sandbox users added manually.

Therefore, a special account has been set up for testing:

Username: unreal_hardy

Password: asdfasdf11

It is strongly recommended to test the app using these credentials, otherwise it will not function correctly.

If required, during the presentation the app will be used with credentials of one of the creators to show that it works for any account.