**Contributions to the project result – Lukas Heimann**

I had two main responsibilities in the project. The first one was creating a good looking, easy to use, modern user interface. The other one was to write the Camera class that contains all the methods to grab frames from the webcam and then later display them in the GUI.

For the GUI I went with a modern, flat UI design. The main colors of the interface are white and blue, to also match the general design of the THU. I created our Logo with open source icons from %ICON%SOURCES%HERE%. All of the GUI elements are borderless and flat to give the interface a modern look and feel. We didn’t want to have too many different windows for the different parts of our software (main menu, options, attendance mode etc.). Therefore, I used only one form to display the main menu, options menu and training menu.

**Menu Screen:**

The main menu contains 4 buttons.

* ***Start***: starts the attendance mode
* ***Options***: shows the options menu
* ***Train***: shows the training menu
* ***Exit***: closes the window and saves the attendance list as a .csv file.

In the options menu there is a dropdown list containing all cameras that are connected to the computer so the user can choose which camera will be used in the attendance mode. Also, there is a text box where the user can enter a matriculation number. If they then press the “***Delete***” button the student with the matching matriculation number will be deleted form the database.

The training menu is used to enter students in the database. First, the user can enter a students name and matriculation number. When the user presses the “***Capture***” button the software captures ten images of the face in the view of the camera. The images are only captured if the algorithm detects exactly one face in the frame. If they then press the capture button again the software captures ten more images. After enough images have been captured the user can press the “***Save***” button to then save the data in the database.

**Attendance Screen:**

This is the second window and the main part of our software. It contains the camera feed as well as the most important information for the professor.

**The Camera Feed:**

As our idea was to show the names of the students in the camera feed to help professors and lecturers remember the students names, I use a full screen window for the attendance screen. This way I get as much space as possible for the camera feed. The panel that shows the camera takes up ¾ of the screen but the aspect ratio of the camera is preserved (16:9) so the picture doesn’t look stretched.

**The Sided Panel:**

The camera feed taking up ¾ of the screen gives us enough space to put a sidebar on the right-hand side that contains the most important informations for the professor. At the top of the side panel there’s the UniFCR Logo. Then we have an *percentage counter* that shows the percentage of attended students. For this I decided to use the CircularProgressBar plug-in by Soroush Falahati. This creates a modern look and also gives a good visual feedback to the professor who wants to see how many students are currently attending the class. Right beneath that we have a *label* that also shows the total *number of students* who are currently attending the class. The last part of the side panel is the *student list*. This list contains all students that are enrolled in the course (in our case all students in the database). If a student is marked as attended the background color of their table entry changes to give a visual feedback to the professor which specific students are currently attending the lecture. I first wanted to show the full information about the student (first name, last name and matriculation number) in this list. But then the list was too big to fit in the side panel creating unwanted scrollbars. I tried to make the list scrollable by dragging the mouse to hide scrollbar from the UI but in the end this didn’t work out. So, I decided to just show the names of the students which made the list significantly smaller.