Himalayan expeditions from 1905 to 2020 year

1. *Subject area*

You are being asked to analyze information about Himalayan expeditions stored in the respective .CSV file**,** through using C++ app with a Qt GUI based on Model-View internal design, implemented through Master-Detail user interface pattern.

2. *Data model*

The database is structured and defined by its ten attributes, “peak_id” (ID of a peak), “peak name” (Name of a peak), “name” (Alpinist Name), “yr_season” (Year and season), “date”, “time”, “citizenship”, “gender”, “age”, “is_O2_used” (did the climber use oxygen while climbing or not).

3. *Main features*

There are a few things you must do in order to satisfy the establishment.

- Your application’s window and it’s UI elements must be dynamically adjustable according to the screen size.
- Your application must be structured properly. That is, it must display the main menu frame at the launch, where the user could have two options to select from: “Start” and “Help”.

A. Help

Clicking on that option brings the user to the frame defined by the following requirements.

- You are required to implement a subitem labeled “About”, that would open a dialog window with anonymous information about the author in the form “Student No.###” at activation.
- Dialog window must also contain a logo made by using the given Qt tools and overriding the paintEvent() method.

B. Start

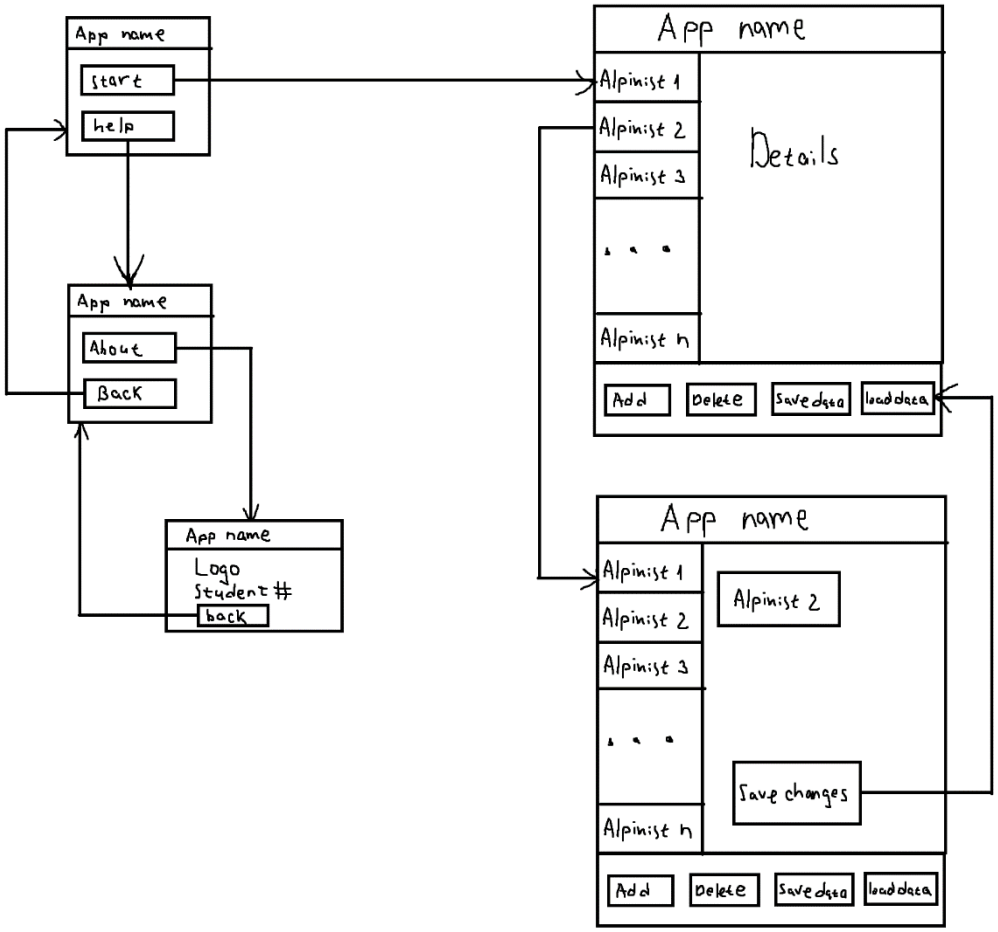
Clicking on that option brings the user to the frame defined by the following requirements.

- The window should have a list view on the left side for the Master aspect, and a Detail view on the right side, which activates when an entry from the Master list is selected.
- The user should be able to add and delete entries of alpinists.
- The user should be able to edit any given field of alpinists.
- Your application must be capable of loading the database from a .CSV file and save changes to a .CSV file.
- There should be UI elements implemented which would represent the number of alpinists for each value of attribute “gender”. These elements should be updated with each change to the database.
- A total counter of alpinists in the database should also be implemented. It also must be continuously updated.

4. Additional features

- You could implement the ability for a user to sort the Master list by a certain attribute of the table.
- You could implement filtering of the entries by setting up an interval for the numeric fields, e.g. for AGE it could be [20; 40], or a specific value for the non-numeric fields.

5. GUI Sketch



6. Additional info

No additional information and resources beyond the topics of our course are required to develop this application. GLHF.