

AQUA Project

Mentored by TechnoLogica

Contents

[1. Authors 2](#_Toc75812930)

[2. Programs we used 2](#_Toc75812931)

[3. Summary 3](#_Toc75812932)

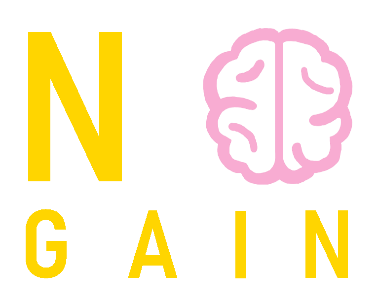
[3.1 Goals 3](#_Toc75812933)

[3.2 Stages of realization 3](#_Toc75812934)

[3.3 Level of difficulty main problems during realization 4](#_Toc75812935)

[3.4 Diagram 4](#_Toc75812936)

[4. Description of the functions used 5](#_Toc75812937)



# Authors

|  |  |
| --- | --- |
| Yoana Ivanova Stoyanova - 10v | [yistoyanova18@codingburgas.bg](mailto:yistoyanova18@codingburgas.bg) |
| Ivan Dyankov Staev - 9v | [IDStaev19@codingburgas.bg](mailto:IDStaev19@codingburgas.bg) |
| Marieta Petkova Stoycheva - 9g | [MPS](mailto:MPS)toycheva19@codingburgas.bg |
| Magdalena Iliyanova Omayska - 10v | [miomayska18@codingburgas.bg](mailto:miomayska18@codingburgas.bg) |

# Programs we used

* Visual Studio Code – to make our README.md
* Visual Studio – to code our project
* Discord – to communicate
* Git Hub – for better teamwork
* Photoshop – for our logo
* Word – to make our documentation
* PowerPoint – to make our presentation
* Nanodbc – a special library to have access to the database.

Requirements for this project to run: SSMS, VS, ODBC 17 DRIVER

Since nanodbc is an external library that’s crucial to this project’s success, here we’ll go over an explanation on how it’s installed and connected.

1. Go to your search bar and open x64\_x84 Cross Tools Command Prompt for VS 2019
2. From the command prompt, access your repos folder. The path to said folder should be something like C:\Users\<your\_username>\source\repos
3. Once you’ve managed to get there, copy and paste these commands line by line:

git clone https://github.com/nanodbc/nanodbc.git

cd nanodbc

mkdir build

cd build

cmake ..

cmake --build .

ctest -V --output-on-failure

1. Open Visual Studio 2019 and load up the project
2. Right click on the project’s name and choose properties
3. Go to the category C\C++ on the left menu, go to Additional Include Directories, select the drop-down menu, select edit, click the yellow icon, click the three dots, go to the place the nanodbc repo is stored (most probably C:\Users\<your\_username>\source\repos) and select nanodbc\nanodbc. Once done, the whole field should read something along the lines of C:\Users\<your\_username>\source\repos\nanodbc\nanodbc
4. Go to the category Linker, go to General, click Additional Library Directories, select the drop-down menu, click the three dots, go to the place the repo is stored, select nanodbc\build\Debug. Once done, the whole thing should read C:\Users\<your\_username>\source\repos\nanodbc\build\Debug
5. Go to the category Linker, go to Input, click Additional Dependencies, choose the drop-down menu, choose edit, write “nanodbc.lib”
6. Select OK

# Summary

## Goals

Our goals are to make a program which is used to store data about glacial lakes. We also want it to be easy for everyone to use.

## Stages of realization

***First stage*** (***planning)*** – The first thing was to create a group in Discord, to distribute roles to each person in our team and to decide which days and what time we can talk to discuss our problems.

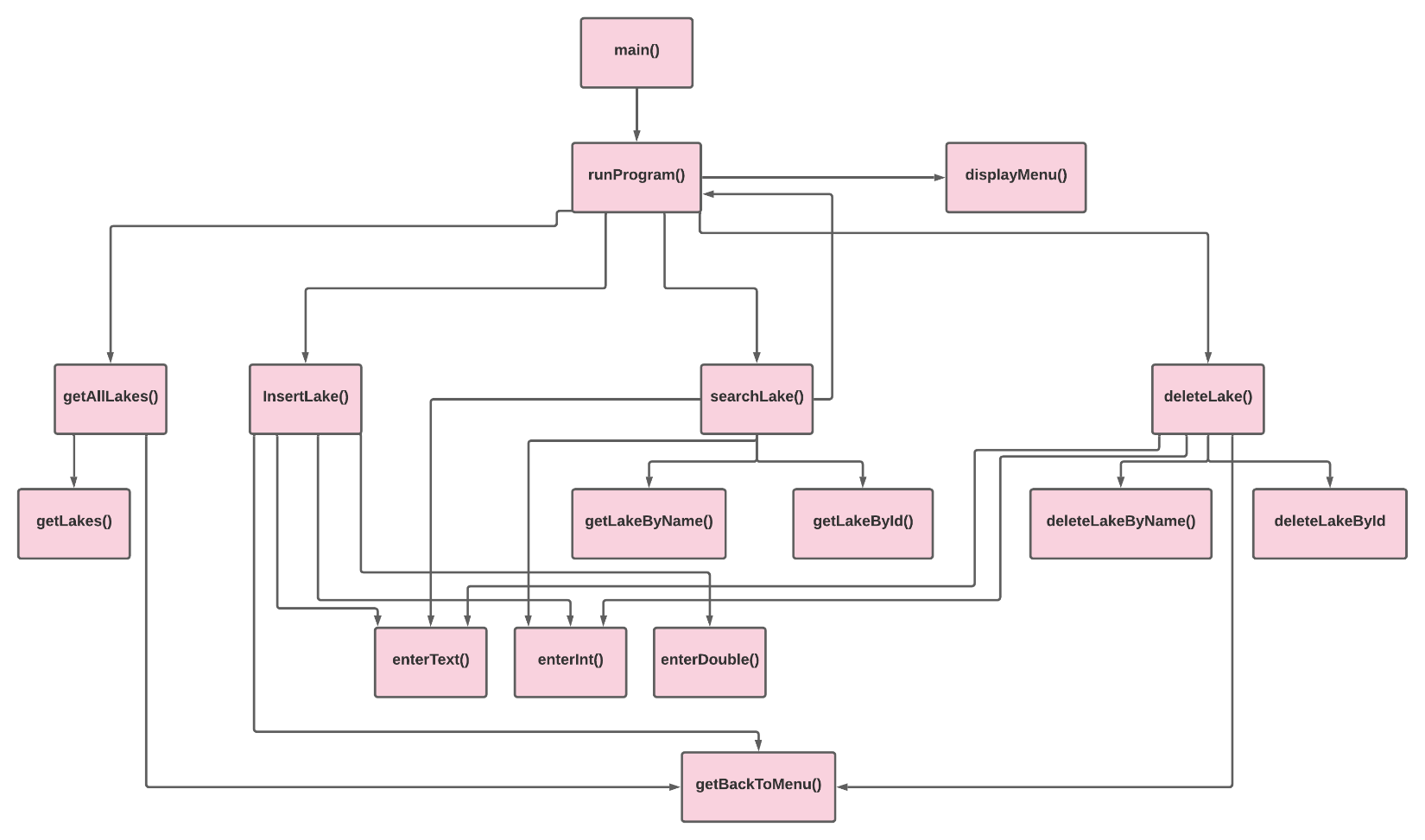
***Second stage*** ***(realization)*** - When we had discussed every problem and every role, we started to work on design and code and almost every day we had to talk about what's going on and who's got what so far.

***Third stage (documentation and presentation)*** - When we have finished with work on code, we should write documentation and make presentation to present our project easier.

## Level of difficulty main problems during realization

We had bunch of problems such as our organization. It was pretty hard because we didn’t really know each other but with good communication we got through that. Also, the time was a problem and that’s why we were struggling at the beginning.

## Diagram



# Description of the functions used

| Function name | Purpose | Arguments | Returned value |
| --- | --- | --- | --- |
| **runProgram()** | Returns the main menu of the program | nanodbc::connection conn | bool |
| **displayMenu()** | Displays the interface of the main menu | N/A | void |
| **deleteLake()** | Prints the delete menu | nanodbc::connection conn | void |
| **deleteLakeById()** | Deletes a like with a given Id | nanodbc::connection conn, const int& id | bool |
| **deleteLakeByName()** | Deletes a like with a given Name | nanodbc::connection conn, const string& lakeName | bool |
| **search()** | Prints the search menu | N/A | void |
| **getLakeByName()** | Searches a lake with a given name | nanodbc::connection conn, const string& lakeName | LAKE |
| **getLakeById()** | Searches a lake with a given Id | nanodbc::connection conn, const string& lakeName | LAKE |
| **insertLake()** | Inserts a new lake | nanodbc::connection conn | void |
| **enterDouble()** | Lets the user enter value of type double | N/A | double |
| **enterInt()** | Lets the user enter value of type int | N/A | int |
| **enterText()** | Lets the user enter value of type string | N/A | string |
| **getAllLakes()** | Prints all the data from the database | nanodbc::connection conn | void |
| **getLakes()** | Gets all the data from the database | nanodbc::connection conn | vector<LAKE> |
| **getBackToMenu()** | Returns to the main menu | nanodbc::connection conn | void |