Yoana Stoyanova

Yistoyanova18@codingburgas.bg

Project Management System

A ScaleFocus Project

Contents

[1. Programs used 2](#_Toc77193642)

[2. Summary 3](#_Toc77193643)

[3.1 Goals 3](#_Toc77193644)

[3.2 Stages of realization 3](#_Toc77193645)

[3.3 Level of difficulty main problems during realization 3](#_Toc77193646)

[3.4 Diagram 4](#_Toc77193647)

[3. Description of the functions used 5](#_Toc77193648)

# Programs used

* Visual Studio – to code the project
* Git Hub – for better teamwork
* Word – to make the documentation
* PowerPoint – to make the 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

The project management system manages the operations of an IT company. It brings visibility to an organization of multiple teams, working on multiple projects.

## Stages of realization

***First stage*** (***planning)*** – The first thing was to plan the structure of the whole application, how it would look and what functions would be needed.

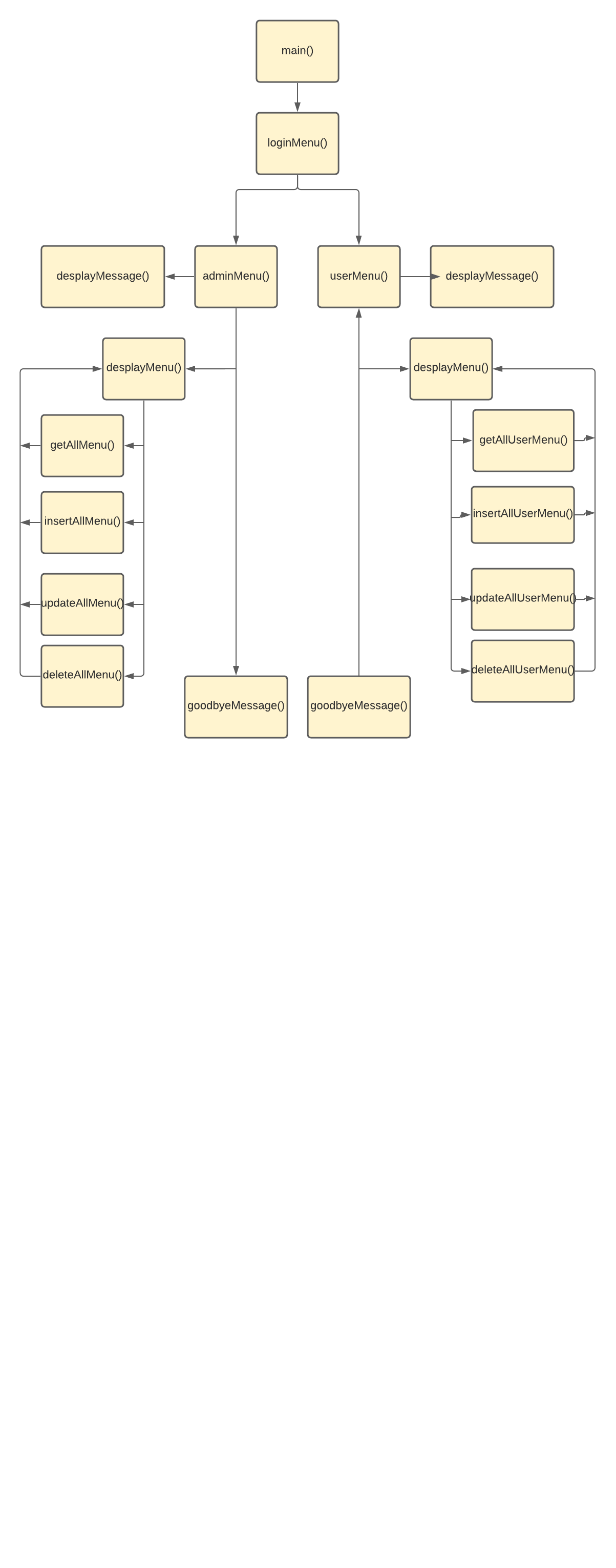
***Second stage*** ***(realization)*** – When everything was planned, the realization of the project began. Trying to stick to my plans, I worked everyday for the past week. When the basic CRUD(create, read, update, delete) functions were done, I moved to the log in and main menus. And finally I passed on to the design of the application when I was done with the menus.

***Third stage (documentation and presentation)*** – Being done with the code, it meant that it was time to do the documentation and finally present in front of the judges.

## Level of difficulty main problems during realization

Most of the realization of the project went smoothly but of course there were complications. For example, one of the main difficulties was the connection between C++ and SQL. But that problem was solved like many others that occurred during the realization of the application.

## Diagram



# Description of the functions used

| Function name | Purpose | Arguments | Returned value |
| --- | --- | --- | --- |
| **getUser()** | Gets all the data from User table in the database | nanodbc::connection conn | vector<USER> |
| **getAllUsers()** | Displays all the data from User table in the database | nanodbc::connection conn | void |
| **getTeam()** | Gets all the data from Team table in the database | nanodbc::connection conn | vector<TEAM> |
| **getAllTeams()** | Displays all the data from Team table in the database | nanodbc::connection conn | void |
| **getProject()** | Gets all the data from Project table in the database | nanodbc::connection conn | vector<PROJECT> |
| **getAllProjects()** | Displays all the data from Project table in the database | nanodbc::connection conn | void |
| **getTask()** | Gets all the data from Task table in the database | nanodbc::connection conn | vector<TASK> |
| **getAllTasks()** | Displays all the data from Task table in the database | nanodbc::connection conn | void |
| **getWorkLog()** | Gets all the data from WorkLog table in the database | nanodbc::connection conn | vector<WORKLOG> |
| **getAllWorkLogs()** | Displays all the data from WorkLog table in the database | nanodbc::connection conn | void |
| **enterInt()** | Lets the user enter value of type int | N/A | int |
| **enterText()** | Lets the user enter value of type string | N/A | string |
| **insertUser()** | Lets the user to create a new user | nanodbc::connection conn | void |
| **insertTeam()** | Lets the user to create a new team | nanodbc::connection conn | void |
| **insertProject()** | Lets the user to create a new project | nanodbc::connection conn | void |
| **insertTask()** | Lets the user to create a new task | nanodbc::connection conn | void |
| **insertWorkLog()** | Lets the user to create a new work log | nanodbc::connection conn | void |
| **updateUser()** | Lets the user to update a user | nanodbc::connection conn, const int& id | void |
| **updateTeam()** | Lets the user to update a team | nanodbc::connection conn, const int& id | void |
| **updateProject()** | Lets the user to update a project | nanodbc::connection conn, const int& id | void |
| **updateTask()** | Lets the user to update a task | nanodbc::connection conn, const int& id | void |
| **updateWorkLog()** | Lets the user to update a work log | nanodbc::connection conn, const int& id | void |
| **USER::deleteUserById()** | Lets the user to delete a user | nanodbc::connection& conn, const int& id | void |
| **TEAM::deleteTeamById()** | Lets the user to delete a team | nanodbc::connection& conn, const int& id | void |
| **PROJECT::deleteProjectById()** | Lets the user to delete a project | nanodbc::connection& conn, const int& id | void |
| **TASK::deleteTaskById()** | Lets the user to delete a task | nanodbc::connection& conn, const int& id | void |
| **WORKLOG::deleteWorkLogById()** | Lets the user to delete a work log | nanodbc::connection& conn, const int& id | void |
| **USER::displayUser()** | Displays the users | N/A | void |
| **TEAM::displayTeam ()** | Displays the teams | N/A | void |
| **PROJECT::displayProject()** | Displays the projects | N/A | void |
| **TASK::displayTask()** | Displays the task | N/A | void |
| **WORKLOG::displayWorkLog()** | Displays the work logs | N/A | void |
| **getProjectById()** | Gets a project by a given id | nanodbc::connection& conn, const int& id | PROJECT |
| **getTaskById()** | Gets a task by a given id | nanodbc::connection& conn, const int& id | TASK |
| **getWorkLogsById()** | Gets a work log by a given id | nanodbc::connection& conn, const int& id | WORKLOG |
| **loginMenu()** | Desplays the log in menu and checks if the user is an admin | nanodbc::connection conn, USER& user | void |
| **login()** | Keeps the information about the logged user | nanodbc::connection conn, string username, string password | USER |
| **displayMenu()** | Displays the main menu | N/A | void |
| **displayMessage()** | Displays Welcome! message | N/A | void |
| **goodbyeMessage()** | Displays the Goodbye! message | N/A | void |
| **displayGetAllMenu()** | Displays the get all menu for the admin menu | N/A | void |
| **displayInsertAllMenu()** | Displays the insert all menu for the admin menu | N/A | void |
| **displayUpdateAllMenu()** | Displays the update all menu for the admin menu | N/A | void |
| **displayDeleteAllMenu()** | Displays the delete all menu for the admin menu | N/A | void |
| **displayGetAllUserMenu()** | Displays the get all menu for the user menu | N/A | void |
| **displayGetAllWorkLogsFormTasksMenu()** | Displays the get for work logs from tasks menu for the user menu | N/A | void |
| **displayInsertAllUserMenu()** | Displays the insert menu for the user menu | N/A | void |
| **displayInsertLogsInTaskMenu()** | Displays the insert all for work logs from tasks menu for the user menu | N/A | void |
| **displayUpdateLogsInTaskMenu()** | Displays the update for work logs from tasks menu for the user menu | N/A | void |
| **displayUpdateAllUserMenu()** | Displays the update menu for the user menu | N/A | void |
| **displayDeleteAllUserMenu()** | Displays the delete menu for the user menu | N/A | void |
| **displayDeleteLogsInTaskMenu()** | Displays the delete for work logs from tasks menu for the user menu | N/A | void |
| **getAllMenu()** | Returns the get all menu | nanodbc::connection conn, USER& user | bool |
| **insertAllMenu()** | Returns the insert menu | nanodbc::connection conn, USER& user | bool |
| **updateAllMenu()** | Returns the update menu | nanodbc::connection conn, USER& user | bool |
| **deleteAllMenu()** | Returns the delete menu | nanodbc::connection conn, USER& user | bool |
| **getAllUserMenu()** | Returns the get all menu for the user menu | nanodbc::connection conn, USER& user | bool |
| **insertAllUserMenu()** | Returns the insert menu for the user menu | nanodbc::connection conn, USER& user | bool |
| **updateAllUserMenu()** | Returns the update menu for the user menu | nanodbc::connection conn, USER& user | bool |
| **deleteAllUserMenu()** | Returns the delete menu for the user menu | nanodbc::connection conn, USER& user | bool |
| **adminMenu()** | Returns the admin menu | nanodbc::connection conn, USER& user | bool |
| **userMenu()** | Return the user menu | nanodbc::connection conn, USER& user | bool |
| **getBackToMenu()** | Returns to the main menu | nanodbc::connection conn | void |