Vocational high school in computer programming

and innovation

**MusalaSoft Project**

Team Amigos

|  |  |
| --- | --- |
| **Team Members** | **Role** |
| *Viktor Velizarov* | *Back-end developer* |
| *Mario Slavov* | *Front-end developer* |
| *Kaloqn Dimov* | *Scrum Trainer* |
| *Viktor Kavaldjiev* | *QA Engineer* |

**1.Table of contents:**

1. **Contents**
2. **Topic**

**2.1*. Short Description***

**2.2*. Purpose***

**2.3*. Goals***

1. **Progression of development**

**3.1. *Main steps***

**3.2*. Difficulties***

1. **Description of used functions**

**2.Topic:**

**2.1. Short Description:**

***Our program is a combination of menus that allow the user to add students and teachers in the program and organize them in different teams and schools. The user can add as many students, teachers, teams and schools as he wants and has to enter the information for every signle one of them manually. After organizing, he can check different reports about them.***

**2.2. Purpose:**

***The purpose of the program is to make organizing tams of students and schools way easier. It is extremely easy to use and anybody who needs to do such organizations will have an easy time doing it. Also the user can furthermore add more reports suiting his needs, in addition to to the ones already added.***

**2.3Goals**

***The goal of our team is to make an easy to use and very reliable program ,that will help anybody who needs to organize teams of students or schools. Also the user will be able to use different reports to furthermore help with the organization process.***

**3. Progress of development**

**3.1. Main steps in the implementation of the project**

* **Defining the main idea**
* **Create a Github repository and organize the tasks**
* **Write the back-end(logical) part of the code**
* **Add front-end(design) to the code so the user has feedback**
* **test the program and find bugs and report them to the dev team**
* **Make the documentation and the presentation of the project.**
* **3.2. Difficulties we went through**

***One of the biggest difficulties we went through was linking all the menus together. After all we managed to do that by organizing the code into different files and cleaning the program. Another difficulty we faced was making sure all the reports work as intended and that there are no fatal errors in the code that break it apart. We managed to fix most of the fatal errors and made all the reports work as planned thanks to our QA engineer who gave us perfect feedback on our mistakes. Thanks to him now we have a working program that can help anyone organize teams and schools easier.***

**Technologies used*: Visual Stuido, GitHub***

**4. Description of used functions:**

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Short Description** | **Function arguments** |
| **bool checkInteger** | **Checks if the function argument is and integer** | **string str** |
| **int getIndexById** | **Gets every student’s index , searching it by its ID** | **STUDENTS\* students**  **int count**  **int ID** |
| **void createStudent** | **create a student by using the original copies of the student’s count and the maxId.** | **STUDENTS\* students**  **int& studentCount**  **int& maxStudentID, STUDENTS newStudent** |
| **void updateStudent** | **swap the chosen student we want to update with a new one , called newStudent.** | **STUDENTS\* students**  **int studentCount**  **int studentID**  **STUDENTS newStudent** |
| **void deleteStudent** | **Moves students in the array one step forward and deletes one of the students.** | **STUDENTS\* students**  **int& studentCount**  **int studentID** |
| **STUDENTS getStudent** | **Returns the students by their indexes** | **STUDENTS\* students**  **int& studentCount**  **int studentID** |
| **bool checkAvailableStudents** | **Checks if there are any students with a given role** | **STUDENTS\* students**  **int& studentCount**  **string searchedRole** |
| **int getTeacherIndexByID** | **Gets every teacher’s index , searching it by its ID** | **TEACHERS\* teachers**  **int& teacherCount**  **int teacherID** |
| **void createTeacher** | **create a teacher by using the original copies of the teacher’s count and the maxId.** | **TEACHERS\* teachers**  **int& teacherCount int& maxTeacherID**  **TEACHERS newTeacher** |
| **void updateTeacher** | **swap the chosen teacher we want to update with a new one , called newTeacher.** | **TEACHERS\* teachers**  **int teacherCount**  **int teacherID**  **TEACHERS newTeacher** |
| **void deleteTeacher** | **Moves teachers in the array one step forward and deletes one of the teachers.** | **TEACHERS\* teachers**  **int& teacherCount**  **int teacherID** |
| **TEACHERS getTeacher** | **Returns the teachers by their indexes** | **TEACHERS\* teachers**  **int& teacherCount**  **int teacherID** |
| **int getTeamIndexByID** | **Gets every team’s index , searching it by its ID** | **TEAMS\* teams**  **int& teamsCount**  **int teamID** |
| **void createTeam** | **create a team by using the original copies of the team’s count and the maxId.** | **TEAMS\* teams**  **int& teamsCount**  **int& maxTeamID TEAMS newTeam** |
| **void updateTeam** | **swap the chosen team we want to update with a new one , called newTeam.** | **TEAMS\* teams, int teamsCount, int teamID, TEAMS newTeam** |
| **void deleteTeam** | **Moves teams in the array one step forward and deletes one of the teams.** | **TEAMS\* teams**  **int& teamsCount**  **int teamID**  **STUDENTS\* students**  **int& studentCount TEACHERS\* teachers**  **int& teacherCount** |
| **TEAMS getTeam** | **Returns the teams by their indexes** | **TEAMS\* teams**  **int& teamsCount**  **int teamID** |
| **void createTeamMenu** | **Used to call the createTeam function and add components to assemble a team.** | **TEAMS\* teams**  **int& teamCount**  **int& maxTeamID**  **TEACHERS\* teachers**  **int& teacherCount STUDENTS\* students**  **int& studentCount** |
| **int getSchoolIndexByID** | **Gets every school’s index , searching it by its ID** | **SCHOOLS\* schools**  **int& schoolCount**  **int schoolID** |
| **void createSchool** | **create a school by using the original copies of theschool’s count and the maxId.** | **SCHOOLS\* schools**  **int& schoolCount**  **int& maxSchoolID**  **SCHOOLS newSchool** |
| **void updateSchool** | **swap the chosen school we want to update with a new one , called newSchool.** | **SCHOOLS\* schools, int schoolCount, int schoolID, SCHOOLS newSchool** |
| **void deleteSchool** | **Moves schools in the array one step forward and deletes one of the schools.** | **SCHOOLS\* schools, int& schoolCount, int schoolID, STUDENTS\* students, int studentCount, TEACHERS\* teachers, int teacherCount, TEAMS\* teams, int teamCount** |
| **SCHOOLS getSchool** | **Returns the schools by their indexes** | **SCHOOLS\* schools, int& schoolCount, int schoolID** |