

**TeamsOnBudget**

*School Management System*

## 

**Table of Contents**

[TeamsOnBudget 1](#_Toc66310420)

[Learning Management System 1](#_Toc66310421)

[Project participants 3](#_Toc66310423)

[Our Idea TeamsOnBudget 5](#_Toc66310424)

[Project System 5](#_Toc66310425)

[Project Diagram 6](#_Toc66310426)

# **Project participants**



**Name**: Atanas Burmov

*Email*: [AABurmov18@codingburgas.bg](mailto:AABurmov18@codingburgas.bg)



**Name**: Stoyan kolev

*Email*: [SNKolev18@codingburgas.bg](mailto:SNKolev18@codingburgas.bg)



**Name**: Alexander Dinev

*Email*: [AGDinev18@codingburgas.bg](mailto:AGDinev18@codingburgas.bg)

Consultant



**Name**: Petur Georgiev

*Email*: [PGergiev@codingburgas.bg](mailto:PGergiev@codingburgas.bg)

# **Our Idea TeamsOnBudget**

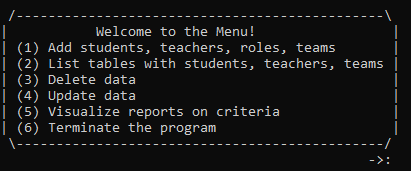


## The main idea of the project is to help educators manage their school work with a focus on interaction and collaborative construction of content, and it is in continual evolution. During the COVID-19 outbreak we reached out that many students can’t continue their education, because of the problems with live sessions. With the opportunity to create we invent a solution written on C++, which is completely free and with minimal system requirements.

## *This is our first version of the project and it’s still under development.*

# **Project System**

## *Main Menu*

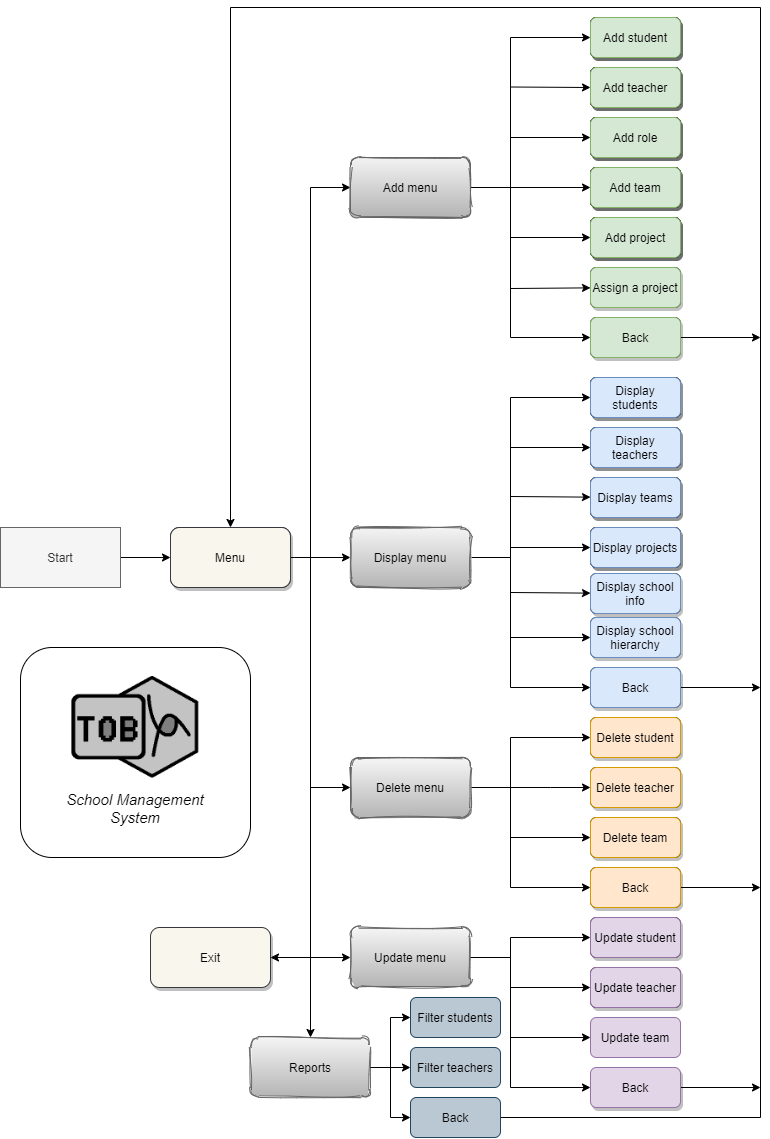


*The user has to choose one of the options and decide what to do.*

### He can add students, teachers and teams. Students can be modified with an important role and then to be added into teams, after that a particular teacher can be added to consult this team. They can be modified in the future, if needed. He can view all students, teachers and created teams. There is an opportunity to delete and update the data too.

# **Project Diagram**

## *It contains future upgrades!*



# **Table with functions**

|  |  |  |  |
| --- | --- | --- | --- |
| Name of function and its type | Arguments | Short description | Returned value |
| void writeRolesInTxt | vector<string>& whiteListedRoles | This function is used to write white listed roles in a .txt file | Returns nothing |
| void writeStudentsInTxt | vector<STUDENT>& students | This function is used to write new students in a .txt file | Returns nothing |
| void writeTeachersInTxt | vector<TEACHER>& teachers | This function is used to write new teachers in a .txt file | Returns nothing |
| void writeTeamsInTxt | vector<TEAM>& teams | This function is used to write new teams in a .txt file | Returns nothing |
| void writeProjectsInTxt | vector<TEAM\_PROJECT>& project, string fileName | This function is used to write new projects in a .txt file | Returns nothing |
| void writeSchoolInTxt | string name, string city, string address | This function is used to write school information in a .txt file | Returns nothing |
| vector<string> readRolesFromTxt() | No arguments | This function is used to read all roles from roles.txt file | Returns vector parsed and read white listed roles from the roles.txt file |
| vector<STUDENT> readStudentsFromTxt() | No arguments | This function is used to read all students from students.txt file | Returns vector parsed and read  students from students.txt |
| vector<TEACHER> readTeachersFromTxt() | No arguments | This function is used to read all teachers from teachers.txt file | Returns vector parsed and read teachers from teachers.txt |
| vector<TEAM> readTeamsFromTxt() | No arguments | This function is used to read all teams from teams.txt file | Returns vector parsed and read teams from teams.txt |
| vector<TEAM\_PROJECT> readProjectsFromTxt | string fileName | This function is used to read all projects from projects.txt file | Returns vector parsed and read projects from projects.txt |
| STUDENT parsedStudentInfo | string info | Parses a information about a student that was previously read from students.txt | Returns a parsed student information |
| TEACHER parsedTeacherInfo | string info | Parses a information about a teacher that was previously read from teachers.txt | Returns a parsed teacher information |
| TEAM parsedTeamInfo | string info | Parses a information about a team that was previously read from teams.txt | Returns a parsed team information |
| TEAM\_PROJECT addProject | string info | Parses a information about a project that was previously read from projects.txt | Returns a parsed project information |
| void updateStudentData | vector<STUDENT>& students, vector<TEAM>& teams, vector<TEACHER>& teachers | Updates a student data by prompting the user to input new information about him | Returns nothing |
| void updateTeacherData | vector<TEACHER>& teachers, vector<TEAM>& teams | Updates a teacher data by prompting the user to input new information about him | Returns nothing |
| void updateTeamsData | vector<TEAM>& teams, vector<TEACHER>& teachers, vector<STUDENT>& students | Updates a team data by prompting the user to input new information about him | Returns nothing |
| void deleteStudentData | vector<STUDENT>& students, vector<TEAM>& teams | Deletes all existing information about a student by prompting the user to enter the his email | Returns nothing |
| void deleteTeacherData | vector<TEACHER>& teachers, vector<TEAM>& teams | Deletes all existing information about a teacher by prompting the user to enter the his email | Returns nothing |
| void deleteTeamsData | vector<TEAM>& teams, vector<TEACHER>& teachers | Deletes all existing information about a team by prompting the user to enter the name of the team | Returns nothing |