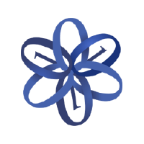
Vocational school for computer programming and innovation



School project

Theme:

Kindergarten project

|  |  |
| --- | --- |
| **Prepared by:**  Georgi Koradov  Aleksander Patrashkov | **Consultant:**  Victor Stoev |
| Burgas 2021 | |

**CONTENTS**

[1. Topic 2](#_Toc77310582)

[2. Authors 2](#_Toc77310583)

[2.1. Back-End Developer 2](#_Toc77310584)

[2.2. Front-End Developer 2](#_Toc77310585)

[3. Used Technologies 3](#_Toc77310586)

[4. Description of the functions 3](#_Toc77310587)

[4.1. Data layer 3](#_Toc77310588)

[4.2. Presentation Layer 4](#_Toc77310589)

[5. Block scheme 5](#_Toc77310590)

## Topic

Our project is about kindergarten management applications. It allows candidacies to be sorted and searched for by parents according to certain criteria. The parent needs to make a registration in order to submit a candidacy. According to point sets for each of the districts of the city, points can be given, which form a score for applying to a kindergarten.

## Authors

## Back-End Developer

Our team Back-End Developer is Georgi Koradov

## Front-End Developer

Our team Front-End Developer is Aleksandar Patrashkov

## Used Technologies

* We used Microsoft PowerPoint to create a presentation, with which we could show you, our work.
* The documentation was made using Microsoft Word.
* Visual Studio is an integrated development environment that our developers used to create the Bletchley-like game.
* In Git Hub we organized our meetings, monitored the progress of our project, and submitted our changes.
* We used Discord as means of communication.
* C++ is the programming language we used for the development of our project.
* With Slide Go we were able to make our documentation and presentation more pleasing to the eye.

## Description of the functions

## Data layer

|  |  |  |
| --- | --- | --- |
| Name: | Parameters: | Description: |
| grantAccessForAdmin | string username, string password, int count, ADMIN\* admins | Checks if you enter valid username and password |
| checkForWrongInput | int variable | Makes you enter a new option until you have entered a correct one |
| findUserIndexByUsername | string username, int count, USER\* users | Finds index of the username of the given name |
| checkForAge | USER\* users, int& counter | Checks whether the child meets the requirements |
| checkForAgeDisplay | USER\* users, int& counter | Displays a message if you dont meet the requirements |
| createUser | USER\* users,int& counter | Creates user only if you have the admin rights |
| showUserInfo | USER users | Shows user's info |
| getUserIndexById | USER\* users, int& counter, int id | Gets the index of the user by the given id |
| getUser | USER\* users, int& counter, int id |  |
| updateUser | USER\* users, int counter, int id, USER newUser | Overwrites the students data |
| deleteUserById | USER\* users, int &counter, int id | Deletes user's info by the given id |
| searchByParentLastName | USER\* users, int& counter, string lastName | Searches user by last name of the parent |
| searchByChildName | USER\* users, int& counter, string childName | Searches user by the child name |

## Presentation Layer

|  |  |  |
| --- | --- | --- |
| Name: | Parameters: | Description: |
| showAllUsersInfo | USER\* users, int& counter | Shows all user's info |
| editUserMenu | USER\* users, int& counter | Menu for the edit user option |
| deleteUserMenu | USER\* students, int& counter | Menu for the delete user option |
| searchUserByCriteriaMenu | USER\* users, int& counter | Menu for the search by criteria functions |
| adminMenu | int count, ADMIN\* admins, int userId, USER\* users, int& counter | Shows the admin rights |
| login | int count, ADMIN\* admins, USER\* users, int& counter | Login function for admins only |
| registrationAsParent | USER\* users, int& counter | Registration for the parent |
| registrationAsAdmin | ADMIN\* admins, int& adminCounter | Registration for admin |
| mainMenu | USER\* users, int& counter, ADMIN\* admins, int& adminCounter | Shows main menu |

## Block scheme

Diagram

Description automatically generated