Vocational school for computer programming and innovation

|  |
| --- |
| @codingburgas |

**String Project**

|  |  |  |
| --- | --- | --- |
| **Prepared by:** | | **Consultant:** |
| Stoyan Nikolaev Kolev | | Petar Georgiev |
| Alexander Galinov Dinev | |  |
| Simeon Radostinov Nikolov | |  |
|  |
|  |
|  |

Table of contents

[1 Short description of the programme 3](#_Toc42898159)

[2 AUTHORS 3](#_Toc42898160)

[3 Consultant 4](#_Toc42898161)

[4 SUMMARY 4](#_Toc42898162)

[4.1 Goals 4](#_Toc42898163)

[4.2 Main stages in the implementation of the project 4](#_Toc42898164)

[4.3 Project complexity level 4](#_Toc42898165)

[5 Block scheme 4](#_Toc42898166)

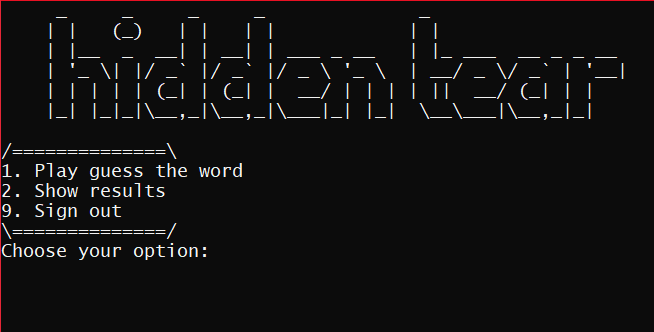
[6 Table with functions 5](#_Toc42898167)

[Table 1 3](#_Toc41862521)

[Table 2 3](#_Toc41862522)

# Short description of the programme

Our project is a programme written on Visual Studio 2019. It is called “Guess the word”. As you can think that is game where you have to guess a word with a given definition.



# AUTHORS

Our team consist of three members. The contact information and the roles of each one of us are described in Table 1:

Table 1

|  |  |
| --- | --- |
| **Members** | **Role** |
| Stoyan Nikolaev Kolev – IX A | Manager |
| Alexander Galinov Dinev – IX V | Developer |
| Simeon Radostinov Nikolov – IX G | Designer |

# 

# Consultant

Table 2

|  |  |
| --- | --- |
| Name: | E-mail: |
| Petar Georgiev |  |

# **SUMMARY**

## Goals

Our goal is to teach people on the different words.

## Main stages in the implementation of the project

The first step was to form up as a team. Then we had to decide what kind of programme we are going to do. In the end we had to write the code and make the documentation & presentation.

## Project complexity level

We encountered some problems and bugs while making the programme, like writing and reading information to/from files. Also we had some bugs with the random generation.

# Block scheme

Figure 1

# Table with functions

Table 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of the function and its type** | **Arguments** | **Description** | **Returned value** |
| void readQuestions | GAME& quiz, int& countQuestions | It keeps the words and the hints in a structure | It returns nothing |
| void readAccounts | Int& countUsers, USER\* users | Reads the data from “accounts.txt” and stores them in a structure | It returns nothing |
| int randomInt | int min, int max | Generates random number | Returns a random rotation value from the interval [0, the number of questions] |
| void welcome | No arguments | It prints out welcome text | It returns nothing |
| int randomIndexWord | int lastIndexOfQuestions | Generates random index | Variable randomIndex |
| void deleteQuestion | GAME& quiz, int index, int countQuestions | Deletes a question with a its index by moving it on the nex position | It returns nothing |
| void guessTheWord | USER\* users, int loggedUser | Runs “Guess the word”. Generates random word from “words.txt” and the user has to guess it | It returns nothing |
| void showScore | USER\* users, int loggedUser | Shows how many points the user has won | It returns nothing |
| bool grantAccess | string username, string password, int count, USER\* users | Depending on the user’s input, it return true or false | true or false |
| int findUserByUsername | string username, int count, USER\* users | Checks if there are any users and if it returns the index of the user | user’s index or -1 |
| bool checkPassword | string password | Checks if user’s password meets all the requirments | true or false |
| void registration | USER\* users, int& countUsers | Registers a user and writes his credentials in “account.txt” | It returns nothing |
| bool userMenu | int count, USER\* users, int loggedUser | When the user is logged it shows him menu with options | true or false |
| void showWordsAndDefinitons | No arguments | Prints out all the words with their definitions which are in the “words.txt” | It returns nothing |
| void login | int count, USER\* users | Prompts the user to enter his credentials and then he is able to play the game | It returns nothing |
| bool adminMenu | USER\* users | Gives the admin a menu where he can add and show words. Also list all registered users | true or false |
| void login | int count, USER\* users | Shows a menu depending on the credentials | It returns nothing |
| bool menu | int& count, USER\* users | The main that gives you the options to create an account, login or just exit the program | true or false |
| void lineDesignUp | No arguments | Design for the upper part of the menu | It returns nothing |
| void lineDesignDown | No arguments | Design for the upper part of the menu | It returns nothing |
| bool checkUsername | string username | Checks if username contains more than 5 characters | true or false |
| void showAllUsers | No arguments | List all registered users usernames | It returns nothing |