School District

User Manual



Team Members: Dumitrescu Tivig, Vlad Cristian, Natan Lellouche, Kyle Husbands

420-331-VA

Teacher: Afzal, Syed Naseem

Monday, May 2nd, 2022

**Table of Contents**

Introduction ………………………………………………………………………….……… 3

Project Summary ………………………………………………………………………...… 3

Project Function ………………………………………………………………….………… 3

Primary Focus ……………………………………………………………...…….………… 3

Exception Handling ……………………………………………………………...………… 4

Project Capability ………………………………………………………………………...… 5

Basic UI ………………………………………………………………….…...………...…… 6

Main Form ………………………………………………………………….……………….. 7

Department Form ………………………………………………………………………...… 7

Student Form ……………………………………………………………………………….. 10

Teacher Form ………………………………………………………………………….….…12

Staff Form ……………………………………….………………………………………...… 14

Globalization/Localization ……….…………….………………………………….…….…. 16

1. **Introduction**

This User Manual contains all the information for the user to take full advantage of all its utilities. This manual will go over the description of the system’s capabilities, its different ways of using it, and its various features. There will be step-by-step instructions for each of the features that are presented.

**1.1 Project Summary**

As explained in the proposal that was submitted, this software was made as a department management system for any school district where the admin of the program could create departments and add students, teachers, and staff members to any given department. Our program main utility is focused more on the academic boards that want to create a database management system for their school and manage the employees and students. Our program is user-friendly, has a lot of flexibility, and is reliable.

Reason For Project: A local school district needs a new database management system to manage the different departments. Our goal with this program is to give them a new management system that will help them become more organized and to keep track of their schools.

**1.2 Primary Functions**

This program’s primary functions are to provide schools with the help they need to keep their databases organized and easy to access. The way the program helps the user is by giving a unique ID to each person that is registered. With this ID the user will always be able to perform tasks and accomplish what they wish. For the school’s departments, they will each be assigned an ID which will be used the same way as the ID that is given to the people. This all helps organize the school’s databases.

**1.3** **General** **Exception Handling**

* If user tries to access any menu before loading a department file.

Graphical user interface, application

Description automatically generated

* If user tries to insert an ID that already exists a message will pop up to stop it
* If user tries to open an incorrect text file a message will pop up telling them they need to open the correct file.
* If a user tries to insert more than one Dean into a department a message box will show up telling them that there is already a Dean
* If a user tries and delete a Dean that does not exist a message box will show up telling the user that a Dean does not exist there
* If the User attempts to open any of the view student/teacher/staff or department menus without loading a file into the database, there will be an error message.
* If the User attempts to add a student/teacher/staff or department with the incorrect ID format, there will be an error message.
* If the User attempts to view the Department view details screen without loading each file, there will be an error message.

**Pictures:**

Graphical user interface, application

Description automatically generated

**1.4 Program Capabilities**

In this section, the program’s capabilities will be discussed in more detail and how they work. In the Department group, there are the teachers, staff and Students. Each one of them has a unique ID which we use to find and change things according to the user. A student is recognized based on the first digit of it’s ID(‘1’). A teacher is recognized based on the first digit of it’s ID(‘2)’. A staff member is recognized based on the first digit of it’s ID(‘3’). A department is recognized based on the first digit of it’s ID(‘1=4’). Our program will let the user search for a department, student, teacher, or staff by the ID when it is created. The user will also be able to implement a text file. This gives the user the option to read, write and overwrite files that will be put in the database. The user will also be able to add new information once a file has been loaded or created. This lets the user update or change information regarding any of the text files.

The program user will be able to delete any staff, student, or department by using the specific ID but there are checkers in place to make sure the user can only select IDs that already exist. After the loading of the teacher, the user can assign a dean who is responsible for the department. You can also remove the dean with their ID.

Of course, the teachers and staff are not working for free so the user can use the program to calculate the salary for each employee. The teacher’s salary will be calculated by the number of hours worked and he/she will have a bigger salary based on their degree. On the other hand, staff members will be paid according to their hours worked and their job. The staff can also be awarded a bonus if they worked enough hours. For both methods the calculation is done automatically as soon as the user puts in the required information so the user does not have to worry about calculation errors.

**Pictures:**

**1.5 Basic UI**

This application is programmed using the C# language and comes with a very simple user interface and has a light design. The UI has multiple different forms, but each form is uniform in background color and font to not confuse the user while navigating. The forms background is steel blue with each buttons being either an old lace or bone color. The fonts for all the writing is also either the default visual studio font or a specific cursive font style.

**2.1 Main Form**

This is the form that opens the moment that the user runs the program. The goal of this form is to Greet the User and to offer him options on what he would want to do next.

**Pictures:**

**Graphical user interface, application

Description automatically generated**

**2.1 Department Form**

This is the main Department form where the user can choose what he wants to do in regard to anything with the Departments. This page can also lead to either the view Department form or the add Department form.

* **ViewDeparment Form:**

This form allows you to view a table with all of the information from the department file. This form is also where the user is able delete, search, and select any row that the user wants using buttons found next to the table. Furthermore, any changes made to the table in this form will also affect the text file that it is linked to.

* **AddDepartment Form:**

This form allows the user to add a new type of Department into the Department.txt file. The user is able to do this by entering all the necessary information needed and then clicking the enter button. Like the viewDepartment Form all the information that was added will also be inserted into the corresponding text file.

* **View Department details Form:**

This form allows the user to see all the current students/teacher/staff members that are currently apart of that Department. You can do this by using the search function in the View Department Form, if the user inputs a correct department ID, it will then cross reference all the matching department IDs with the IDs of the students/teacher/staff members, then will highlight all the records that match.

**Graphical user interface

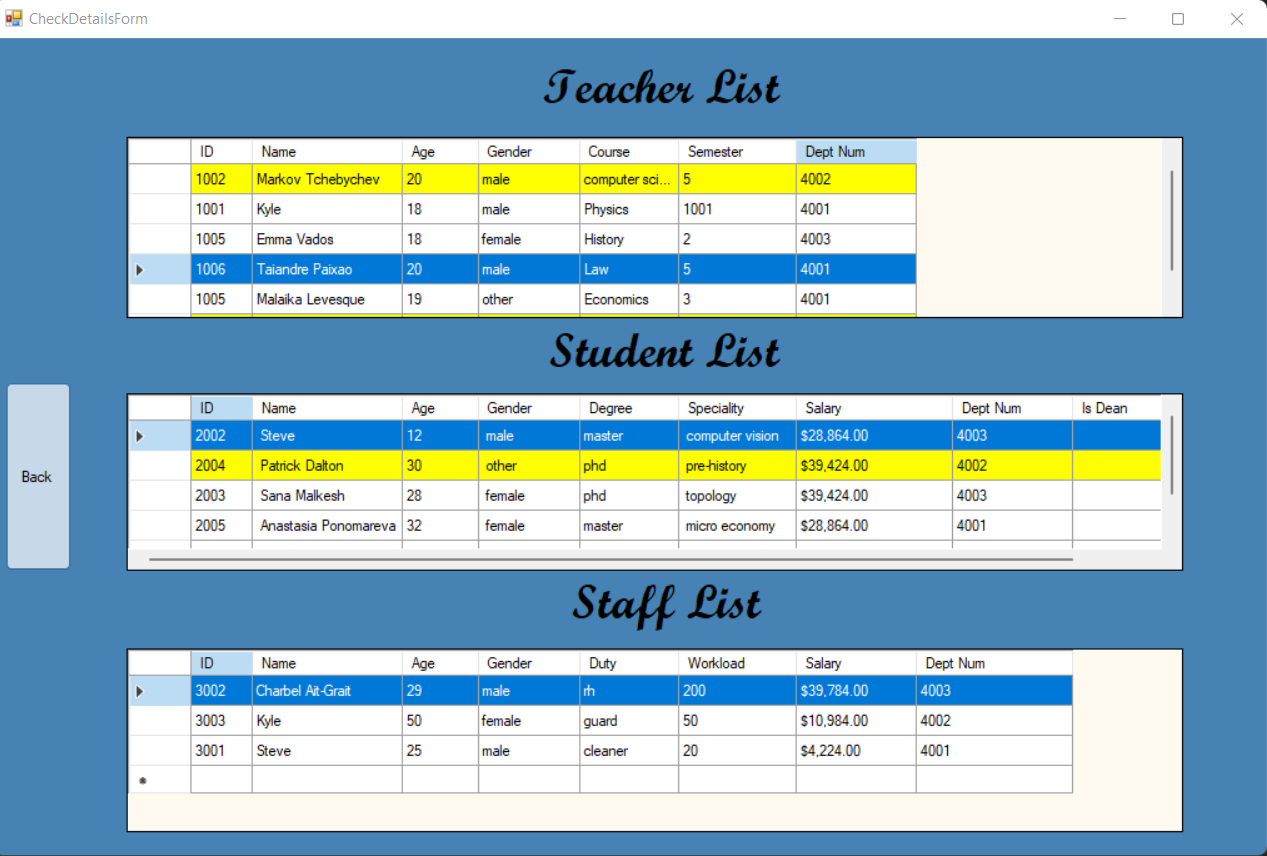
Description automatically generatedPictures:**

**Graphical user interface, application

Description automatically generated**

**A picture containing graphical user interface

Description automatically generated**

****

**2.2 Student Form**

This is the main Student form where the user can choose what he wants to do in regard to anything with Students. This page can also lead to either the view Student form or the add Student form.

* **ViewStudent Form:**

This form allows you to view a table with all of the information from the Student file. This form is also where the user is able delete, search, and select any row that the user wants using buttons found next to the table. Furthermore, any changes made to the table in this form will also affect the text file that it is linked to.

* **AddStudent Form:**

This form allows the user to add a new type of Student into the Student.txt file. The user is able to do this by entering all the necessary information needed and then clicking the enter button. Like the viewStudentForm all the information that was added will also be inserted into the corresponding text file.

**Pictures:**

**Graphical user interface, application, website

Description automatically generated**

**Table

Description automatically generated with medium confidence**

Graphical user interface, application

Description automatically generated

**2.3 Teacher Form**

This is the main Teacher form where the user can choose what he wants to do in regard to anything with Teacher. This page can also lead to either the view Teacher form, the add Teacher form, or the set Dean form.

* **ViewTeacher Form:**

This form allows you to view a table with all of the information from the Teacher file. This form is also where the user is able delete, search, and select any row that the user wants using buttons found next to the table. Furthermore, any changes made to the table in this form will also affect the text file that it is linked to.

* **AddTeacher Form:**

This form allows the user to add a new type of Teacher into the Teacher.txt file. The user is able to do this by entering all the necessary information needed and then clicking the enter button. Like the viewTeacherForm all the information that was added will also be inserted into the corresponding text file.

* **SetDean Form:**

This form allows the user to make one teacher per Department a Dean. The user is able to do this by searching up the teachers ID into the search bar, clicking the search button and then clicking enter.

**Graphical user interface, application

Description automatically generatedPictures:**

**Graphical user interface, application

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**2.4 Staff Form**

This is the main Department form where the user can choose what he wants to do in regard to anything with the Departments. This page can also lead to either the view Department form or the add Department form.

* **ViewStaff Form:**

This form allows you to view a table with all of the information from the Staff file. This form is also where the user is able delete, search, and select any row that the user wants using buttons found next to the table. There is also a specific method incorporated into the table were when the user updates a staff members workload its Salary also changes. This is done automatically, and the user does not have to manually input it. Any changes made to the table in this form will also affect the text file that it is linked to.

* **AddStaff Form:**

This form allows the user to add a new type of Staff into the Staff.txt file. The user is able to do this by entering all the necessary information needed and then clicking the enter button. Like the viewStaffForm all the information that was added will also be inserted into the corresponding text file.

**Graphical user interface

Description automatically generatedGraphical user interface, application, Word

Description automatically generatedPictures:**

**Word

Description automatically generated with medium confidence**

**2.5 Globalization/Localization**

This feature was added as per the instructions of the teacher and is meant to offer an option for users who do not speak English to still use the program to the fullest of its abilities. The way that our group went about to implement the globalization was to create a completely different class in which a method was created using both the System. Configuration but also System.XML.

Text

Description automatically generated

Graphical user interface, application

Description automatically generatedText

Description automatically generatedOnce the class was implemented the next step was to add the code into the form that had the language picker. This made it like that once a language was picker the Form would change automatically. There was also a checker put into each language like that the other forms would stay to the language that the user picked.

The final step was to change the label text for each form depending on the language and use the checker in the first form to automatically change the forms language based on what the user chose.

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, diagram

Description automatically generated with medium confidence