Logo

Description automatically generated

# 

8.11.2021

Table of content

[1. Information about our team 2](#_Toc95210971)

[2. Recap 2](#_Toc95210972)

[2.1 Main goal 2](#_Toc95210973)

[2.2 Tasks 2](#_Toc95210974)

[2.3 Realisation (apps and language that we used) 2](#_Toc95210975)

[3. Structure 3](#_Toc95210976)

[3.1 Structure of the files 3](#_Toc95210977)

[3.2 Structure of the functions 4](#_Toc95210978)

[3.3 Block scheme 6](#_Toc95210979)

[4. Summary 7](#_Toc95210980)

# Information about our team

|  |  |  |
| --- | --- | --- |
| Name | Class | Roles |
| Petya Petkova | 10 V | **Scrum Trainer** |
| Vasil Mladenov | 10 V | **Back-end Developer** |
| Daniel Georgiev | 10 B | **Front-end Developer** |
| Ivelin Vasilev | 10 A | **QA Engineer** |

# Recap

## Main goal

Our task is to create an application for registration and management of historical events, using linked lists.

## 2.2 Tasks

1. We collected information about the task.
2. We assigned roles.
3. We made a logo.
4. We made a README file and added “about” on GitHub.
5. We made a Testing plan.
6. We wrote the Front-End.
7. We wrote the Back-End.
8. We made a design for the presentation and documentation.
9. We made presentation, documentation, and QA documentation.

## 2.3 Realisation (apps and language that we used)

1. Apps, that we used:

* **Teams, Discord** and **GitHub** for communication and team synchronization.
* **Visual Studio** for writing the code.
* **PowerPoint** for the making of the presentation.
* **Word** for the making of the documentation.
* **Excel** for the making of the QA documentation

1. Programming language(s) that we used:

* **C++**

# Structure

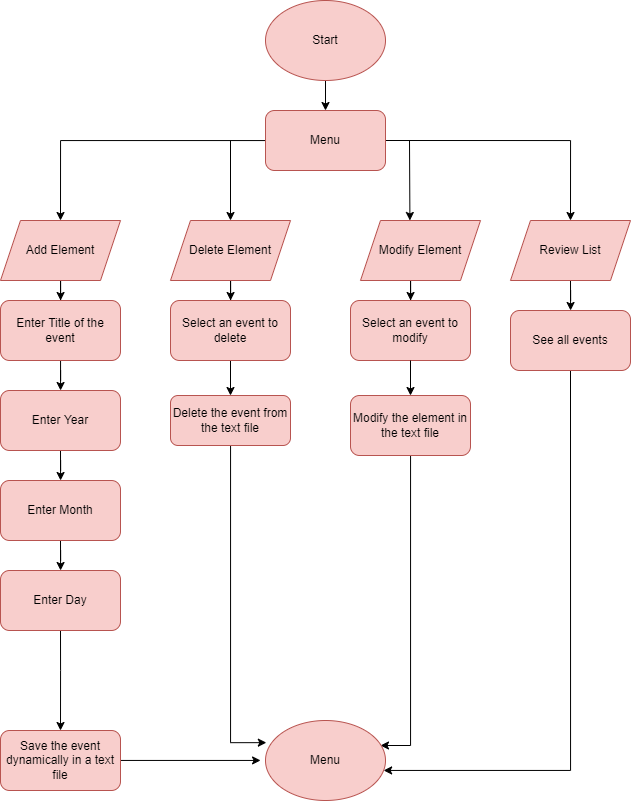
## 3.1 Structure of the files

|  |  |  |  |
| --- | --- | --- | --- |
| № | | Name of the file | What is this file for? |
| 1. | HistoryProject.cpp | Connects all the files together |
| 2. | Menu.cpp | Contains the code for the front end and the menu |
| 3. | LinkedList.cpp | Contains the code that adds, deletes, modifies and reviews the events |
| 4. | Menu.h | Header file for Menu.cpp |
| 5. | LinkedList.h | Header file for LinkedList.cpp |

## 3.2 Structure of the functions

|  |  |  |
| --- | --- | --- |
| № | Name of the file | What is this function for? |
| 1. | void menu() | Calls the menu |
| 2. | void load() | Prints a message when an event is added |
| 3. | void dynamicMenu(Node\* Head) | Add dynamic menu where you can view and add elements |
| 4. | void removeMenu(Node\* Head) | Add dynamic menu where you can view and delete elements |
| 5. | void linkedList::prependNode  (Node\*& Head, string userTitle, int userDay, int userMonth, int userYear) | Add new element at the beginning of the list |
| 6. | void linkedList::appendNode  (Node\*& Head, string userTitle, int userDay, int userMonth, int userYear) | Add new element at the end of the list |
| 7. | void linkedList::addNode  (Node\*& Head) | Enter event and choose whether to put it in the front or the back of the list |
| 8. | void linkedList::removeGivenNode  (Node\*& Head, string keyTitle, int keyDay, int keyMonth, int keyYear) | Remove events from the list |
| 9. | void linkedList::removeNode  (Node\*& Head) | Select event and delete it afterwards |
| 10. | void linkedList::modifyNode  (Node\*& Head) | Select element to modify |
| 11. | void linkedList::modifyGivenNode  (Node\*& Head, string keyTitle, int keyDay, int keyMonth, int keyYear) | Modify element |
| 12. | void linkedList::printNodes  (Node\* Head) | Display events |
| 13. | int checkDay  (int userMonth, int userYear) | Checks if the day you entered is valid and makes you enter it again if it is not |
| 14. | int checkYear() | Checks if the year you entered is valid and makes you enter it again if it is not |
| 15. | int checkMonth() | Checks if the month you entered is valid and makes you enter it again if it is not |
| 16. | void fileFunctions::writeToFile  (Node\* Head) | Add events to a text file |
| 17. | long long int decToBin(int dec) | Convert numbers from decimal to binary |
| 18. | long long int binToGraysCode(int dec) | Convert numbers from binary to Gray code |
| 19. | int grayCodeConversion  (long long temp) | Use gray code as binary and convert it to decimal |
| 20. | void linkedList::addBetween  (Node\*& Head, string userTitle, int userDay, int userMonth, int userYear) | Add events between two others |
| 21. | Bool checkBigger  (Node\* firstNode, Node\* secondNode) | Checks which event is more recent |

## 3.3 Block scheme



# Summary

In this school project we learned how to work in team better. We learned new things about linked lists and understood how the gray code works and how we can use it in a program.