GameShop©

***Project documentation***

1. Info about us & our school

1.1 Our school: Professional High school for Computer programming and Innovations

1.2 Our project: GameShop©

1.3 Creators of the project:

Viktor Velizarov(VVVelizarov18@codingburgas.bg)

Georgi Kanev(GBKanev18@codingburgas.bg)

Ivan Parashkevov(ISParashkevov18@codingburgas.bg)

Stefan Ganchev(SPGanchev18@codingburgas.bg)

1.4 Project Manager: Viktor Velizarov([VVVelizarov18@codingburgas.bg](mailto:VVVelizarov18@codingburgas.bg))

2. Documentation content

1. Info about us & our school

2. Documentation content

3. Short description and the goals of our project

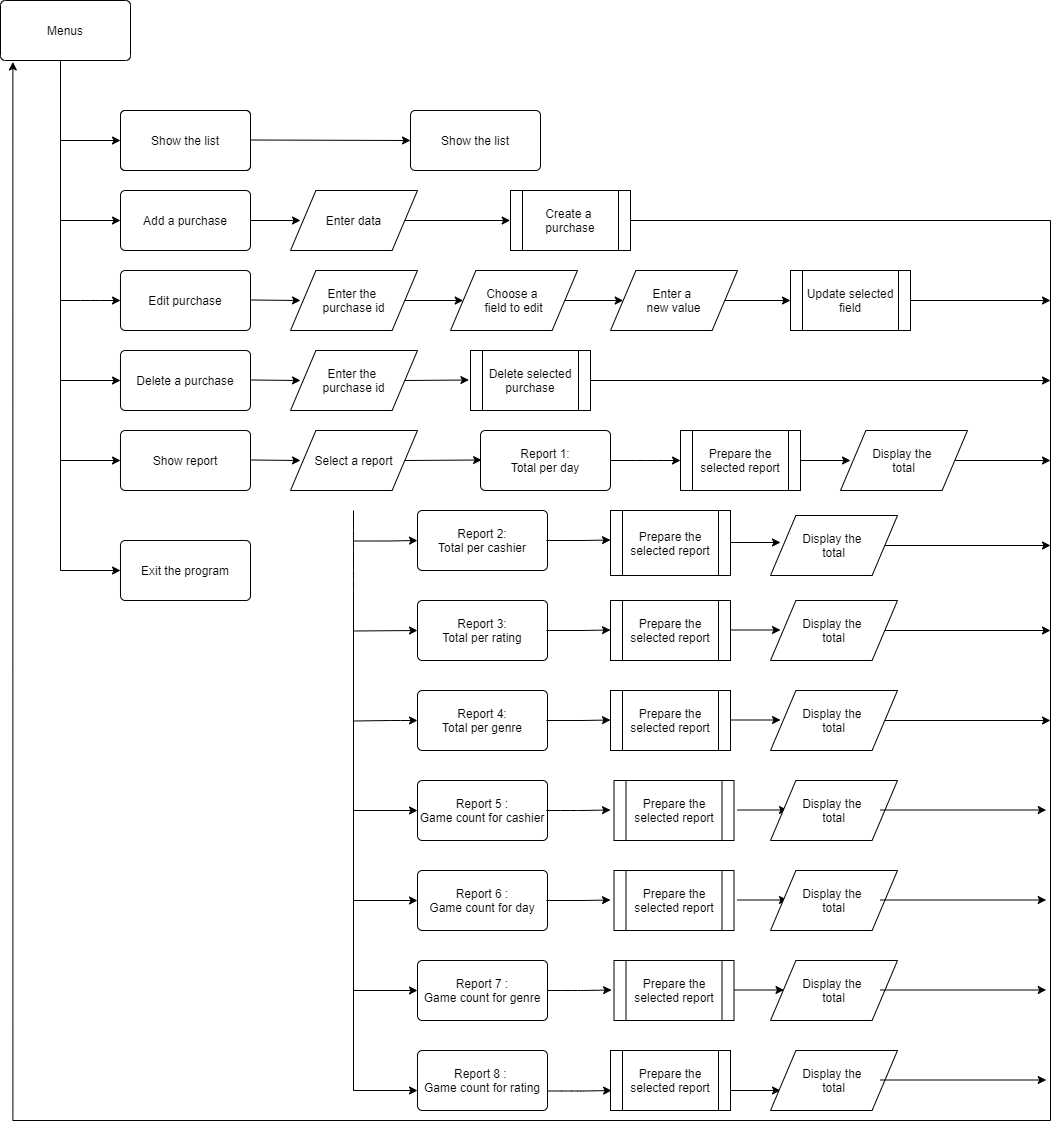
4. Diagram, describing the main menu

5. Description of used functions

3. Short description and the goals of our project

The idea of our project is to create a program which will be used to help cashiers at stores which sell videogames. Cashiers using it can keep track of game purchases and edit every element of them. If they made a mistake while adding a purchase they can also delete it. Another thing cashiers can do is see some reports for example how many video games were sold on a particular day , or how much money has a certain cashier made. There are 8 reports in total for now but we plan on adding more in the future.

4. Diagram, describing the main menu



5. Description of used functions

|  |  |  |  |
| --- | --- | --- | --- |
| Name of the funtion | Short description | Description of arguments | Returned  values |
| getIndexByld | Gets every order’s index , searching it by its ID | GAMES games – array of elements  Int gameCount – number of indexes in the array  Int ID – id values | Returning index as an int value if true , and an error if its false |
| createPurchase | We create a purchase by using the original copies of the game’s count and the maxId. | GAMES games – array of elements  Int gameCount – number of indexes in the array  int maxId – the number of id’s  GAMES newPurchase |  |
| updatePurchase | We swap the chosen purchase we want to update with a new one , called newPurchase. | GAMES games – array of elements  Int gameCount – number of indexes in the array  Int ID – id values  GAMES newPurchase |  |
| deletePurchase | Moves purchases in the array one step forward and deletes one of the purchases. | GAMES games – array of elements  Int gameCount – number of indexes in the array  Int ID – id values |  |
| getPurchase | Returns the purchases by their indexes | GAMES games – array of elements  Int gameCount – number of indexes in the array  Int ID – id values | Returns purchases by their indexes |
| getTotalPerDay | Returns the total for a chosen day by comparing that day to other day inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Char dayofweek – one of the elements in the structure. | Returns a float value of the sum for the total per chosen day |
| getTotalPerCashier | Returns the total for a chosen cashier by comparing that cashier to other cashier inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Char cashier – one of the elements in the structure. | Returns a float value of the sum for the total per chosen cashier |
| getTotalPerRating | Returns the total for a chosen rating by comparing that rating to other rating inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Int rating – one of the elements in the structure. | Returns a float value of the sum for the total per chosen rating |
| getTotalPerGenre | Returns the total for a chosen genre by comparing that genre to other genre inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Char genre – one of the elements in the structure. | Returns a float value of the sum for the total per chosen genre |
| gamesPerCashier | Returns the count of games for a chosen cashier by comparing in to other cashier inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Char cashier – one of the elements in the structure. | Returns an int value of the count of purchases with a chosen cashier |
| gamesPerDay | Returns the count of games for a chosen day by comparing in to other day inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Char day – one of the elements in the structure. | Returns an int value of the count of purchases with a chosen day |
| gamesPerGenre | Returns the count of games for a chosen genre by comparing in to other genre inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Char genre – one of the elements in the structure. | Returns an int value of the count of purchases with a chosen cashier |
| gamesPerRating | Returns the count of games for a chosen rating by comparing in to other rating inputs in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  Int rating – one of the elements in the structure. | Returns an int value of the count of purchases with a chosen cashier |
| createPurchaseMenu | We use this function to fill every element of an index in the array manually | GAMES games – array of elements  Int gameCount – number of indexes in the array  int maxId – the number of id’s | We cout messages to the user so he can see which fields to type in the array |
| showPurchaseMenu | Shows all the elements in the array | GAMES games – array of elements  Int gameCount – number of indexes in the array  int maxId – the number of id’s |  |
| deletePurchaseMenu | Gets an ID from the user and then sends in to the deletePurchase funtion | GAMES games – array of elements  Int gameCount – number of indexes in the array  int maxId – the number of id’s |  |
| editPurchaseMenu | Gets an ID from the user for which purchase they want to change.Then displays a menu of every element in that purchase so they can choose. After they choose it calls updatePurchase to change the chosen purchase | GAMES games – array of elements  Int gameCount – number of indexes in the array |  |
| showReports | Shows a menu of all the reports and lets the user choose one. His choice is then put in a switch that calls other functions to do the job . | GAMES games – array of elements  Int gameCount – number of indexes in the array |  |
| showMenu | Shows the main menu to the user and lets him choose from it. His choice is then put in a Switch that calls other functions to do the job. | GAMES games – array of elements  Int gameCount – number of indexes in the array  int maxId – the number of id’s |  |
| compareStringYes | We use this to compare two strings | String main – one of the strings we want to compare  String yes – second string | Returns value from the comparison – 0 if the strings are the same |
| compareStringNo | We use this to compare another two strings | String main – one of the strings we want to compare  String no – second string | Returns value from the comparison – 0 if the strings are the same |