Bikes sales Program

David Atkins

Specification

This project is a bike sales program which will allow the user to select different kinds of bikes such as mountain and road bikes. It will allow the users to browse all these different bikes and allow them to see a range of prices for these different bikes. They will be able to filter the bikes which are shown via entering in the max budgets they want to spend. Once they have viewed the bikes available, they will then be able to enter their email and have the bikes they have ordered sent to them.

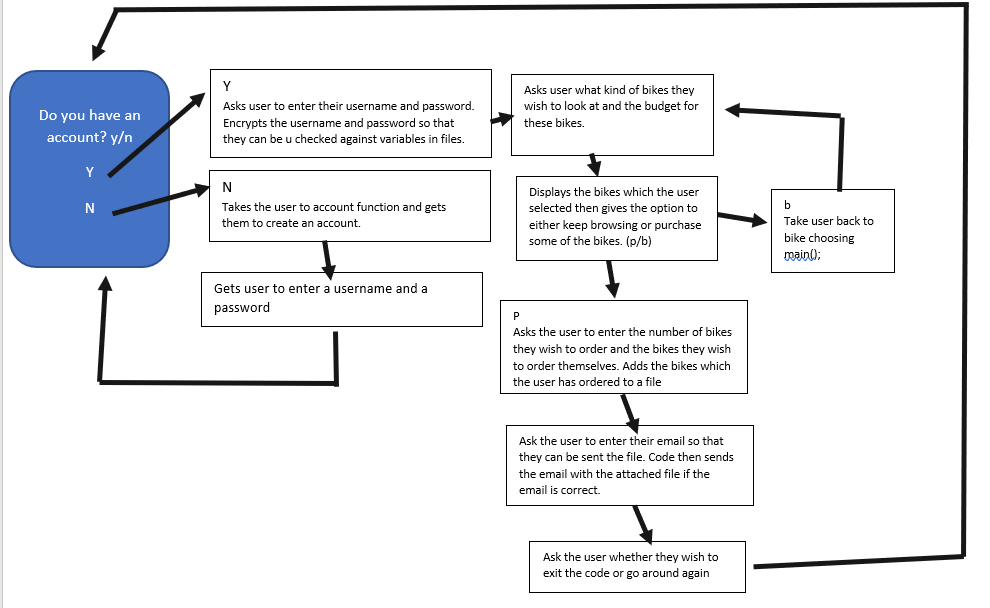
The program will run via visual studios and uses outside files in order to store important information which the program needs in order to function. This includes the bikes which will be on display and encrypted information such as the user’s username and password.

The program will be linear having the user go through the programming, selecting the options they want the code to do along the way. This will include selecting all the filters they want for the bikes and viewing and selecting information like which bikes they wish to order. The programming will later ask the user to input their email address, when they get to the email function, to allow them to get the email with the attached file containing the bikes they ordered.

Key features:

* Username and password encryption login.
* Selecting files depending on the bike option
* Email which sends the bikes they have ordered
* Filtering down bike options
* The ability to keep browsing or order bikes

Design and implementation



Design:

The code will be split up into many sections to allow for the code to easily pass from one section to another. The code will start on main but then be taken to the account function if the user is not yet signed in. On the account function the user will be asked if they have an account or not. If they do then they will simply be able to sign in and carry on with the code. Otherwise they will be asked to create a username and password. Once the account function is done then the code will move on back to main. The username and password will be stored on a text file.

Once back at main the user will select the kind of bikes they wish to look at and the budget for the prices they’re going to look at. This will determine the information which will be shown to them later on in the code. There will be two functions for the bikes, this includes a function for road bikes and a function for mountain bikes. Each of these functions will get the budget information from the main function in the code, this information will mainly include the budget which will be needed in order to get the information on what bikes to display. Once the bikes selected have been displayed the user will get the option to either keep browsing or purchase some of the bikes. If they select browse they will be sent back to main function where they will re enter the information on bikes they wish to look at. If they select purchase, they will be taken to the purchase function.

On the purchase function first, they will input the number of bikes they wish to purchase. There will not be a limit to this. After the code knows the number of bikes the user wants, they will then enter the bikes they wish to purchase. This will keep looping until the number of bikes the user wished to purchase has been met. The bikes the user inputs will be written onto a text file for later access. The code will then move onto the mail function.

Once at the email function the user will enter their first name, last name and email. This will allow the email to be sent and for customised messages using the user names. The email will be sent straight away and will give a message describing what the email is and an attached document that includes the bikes which the user inputted earlier. If there is a problem with sending the email than the user will be notified and told why the email could not send.

Once the email is sent the last function the user will come to will be an exit function where they will simply get the option to either keep browsing or exit the code. If they wish to keep browsing the code will simply loop back to the beginning. Otherwise the code will end.

Goes to accountNew function

Incorrect input

Yes

No

Enters username

creates username

do they have an account

Creates password

Enters password

Mount

Road or mountain

No

Yes

Yes

No

road

b

P

End

Loop=amount

Goes to purchase function

Displays selected bikes

Goes to mountain bikes function

Goes to road bikes function

Do u wish to exit

Goes to exit function

Sends email

Enters email and First/ second name

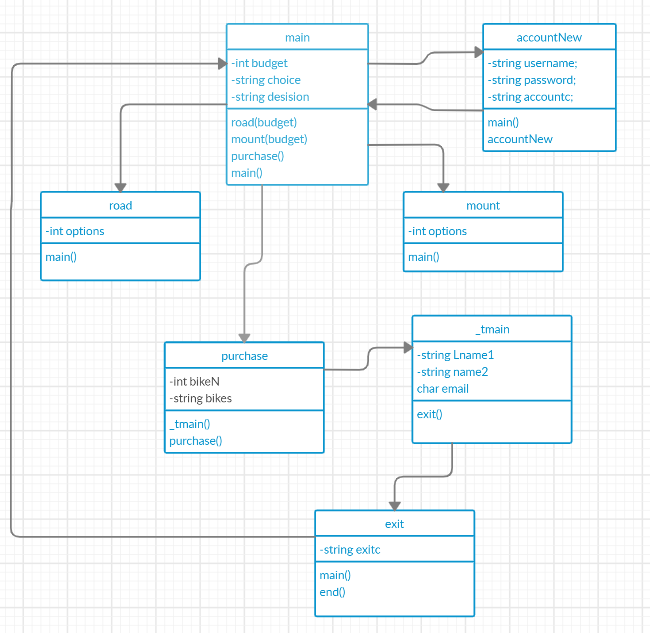
Goes to email function

Enters bikes orders

User enters the amount of bikes

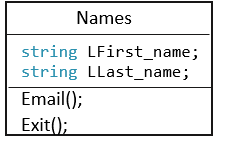
Would you like to purchase or keep browsing?

Displays selected bikes



Structs:

The code uses one struct to store the name of the user on the code so that it can be used in multiple functions to allow for personalised emails and messages.



File structure

The file structure which is used to store the bikes and the information about the bikes is txt file. This stores all of the bikes’ information with one bike and its information per line. This means that when a specific bike is needed the code can get the bike via requesting the text from that line. The bikes are ordered into two files: road and mount. Within the file they are listed via Rbikes for road or Mbikes for mountain bikes. A number follows these names to allow them to be identified easily.

For example - road file:

Rbike1

Rbike2

Rbike3

Rbike4…

For the username and the password, a file is also created for each of them. These again are txt files. These will simply have the username and the password listed on them allowing for the code to easily access them as necessary. The username and the password will be encrypted on the text file they are stored on in order to make sure code understands the text. Any inputs which are checked against the data in the file will also be encrypted.

Test plan:

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Action | Test method | Success criteria |
| 1 | All inputs work | Go through the code and enter all the inputs and make sure that they all give data to the correct variables and work correctly. | All of the inputs worked and they all gave the correct variables the correct data. |
| 2 | Functions work and send the code to the correct function after they are done. | To go through the code and make sure that code goes through the correct order of functions depending on what I’m telling the code to do. | That all the functions work correctly and take the code to the correct next stage. |
| 3 | Loops | Check to make sure all the loops work and loop the section of code they are covering the correct amount of times. | That all the loops work and loops the correct number of times. |
| 4 | Spell and grammar | Go through the output to the user and make sure both the spelling and grammar are correct and professional. | That all the spelling and grammar is correct. |
| 5 | File handling | That when needed the files that are needed for the programming are accessed correctly and extract the information. | The files are accessed and the correct data is taken from them. |
| 6 | Email | Input an email and make sure that the code sends the email. | That the email it has been sent to receives the email. |
| 6.1 | Email test 2. The file is attached to the email | That when the email is sent by the code it also sends the connected file. | The email has the attachment connected to it. |
| 6.2 | Once the email is sent the code moves on to the correct place | Inspect the code so that once the email is sent it moves onto the next phase which should be the exit function. | That the email moves on correctly and there are no issues. |
| 7 | exit | Check the code, when instructed to, exits the code. | That the code exits when it should. |
| 8 | If statements all work | Look through all the if statements in the code to makes sure that they are correct and that they do the correct outcomes depending on the circumstances. | That the if statements work and do the correct response to the criteria given to them. |
| 9 | Discount which should be given if more than one order of bikes happens | Check once the code has been done once and the user wishes to order more bikes that the code gives the 50% discount message with the email. | That the 50% email message is given when the code loops through after making a previous order. |