A screenshot of a login screen

Description automatically generated with medium confidenceDesign All Login Screen Mock up

Button – btnClose

Allows the user to exit the program

Button – btnEnter

Allows the user log in

Textbox – txtPassword

Allows the user to input their password

Textbox – txtUsername

Allows the user to input their username

All text **arial black** by default

All buttons **dark pastel blue**

Background is **grey**

All median text 12pt

All large text 24pt

Data Dictionary   
Login:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Scope (Local/Global) | Format | Purpose |
| strUsername | String | Local | XXXXXXXXXX | Stores what the user inputs for their username |
| strPassword | String | Local | XXXXXXXXXX | Stores what the user inputs for their password |
| strCorrectUsername | String | Global | XXXXXXXXXX | Reads the correct username from the UserLogins.txt textfile to validate against the inputted username |
| strCorrectPassword | String | Global | XXXXXXXXXX | Reads the correct password from the UserLogins.txt textfile to validate against the inputted password |

Object Description  
Login:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Method/Event | Properties | Purpose |
| txtUsername | Textbox | Method | Arial, 12pt  ReadOnly: False, “” | Allows the user to input their username |
| txtPassword | Textbox | Method | Arial, 12pt ReadOnly: False, “” | Allows the user to input their password |
| btnClose | Button | Event: Click | Arial, 12pt, “Close” | Allows the user to exit the program |
| btnEnter | Button | Event: Click | Arial, 12pt, “Enter” | Allows the user to enter their login details |

Pseudocode  
Login:

A picture containing text, screenshot, software, font

Description automatically generated

Design 1 Mock up

Combo box - cbo[colname]

Each combo box represents a separate column where data is inputted. Can be modified at any time.

Button – btnAddData  
Adds inputted data to specified column

Textbox – txtAddData  
Allows user to input data that will be appended to one of the columns

Combo box – cboAddDataCol  
Allows user to select which column to input data to

Menu strip item – “File”  
Contains all software file functionality

Menu strip item – “Edit”  
Contains all software editing functionality

Menu strip item – “View”  
Contains all editorial functionality

A screenshot of a computer

Description automatically generated with low confidence

All text **arial black** by default

All buttons **dark pastel blue**

All combo boxes **pastel blue**

List box **dark grey**

Background is **grey**

All small text 10pt

All median text 12pt

All large text (header) 18pt

Menustrip is **white**

Button – btnSearch   
Searches all items within selected column and shows all other items in the cells adjacent

Textbox – txtSearchInput  
Reads the user input search for cboSelectTextColour

Combo box – cboSearchSelect   
Allows the user to select a column that will be searched

List box – lstStats   
Displays all previous shown searches

Radio button – optSelectCellColour   
Allows user to select the colour of the boxes containing data they enter   
  
Radio button – optSelectTextColour   
Allows user to select the colour of their text

Button – btnAddColumn   
Adds the inputted string from txtAddColumn as a new column in dgdmain

Textbox – txtAddColumn  
Allows the user to input a new column

Button – btnUpdateColumn  
Updates the column name

Combo box – cboSelectColumn  
Allows the user to select a column that will be renamed

Data Dictionary   
Design 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Scope (Local/Global) | Format | Purpose |
| blnLoggedIn | Boolean | Global | True/False | Stores whether the user has successfully logged in |
| arrCol[columnname] | Array | Local | [XXX,XXX] | Stores data for each individual column within a class |
| arrClass[classname] | Multidimensional array | Local | [[XXX],[XXX]] | Stores each individual class within the database |
| arrDatabase | Multidimensional array | Global | [[XXX],[XXX]] | Stores all the classes within one array |
| strColumnCount | Integer | Local | XX | Stores the number of columns within the class database |
| strNewColumnName | String | Local | XXXXXXXXXX | Stores the new name for the designated column |
| arrTextColour | Array | Local | [XXX,XXX] | Stores all the different colour options for the text |
| strSelectedTextColour | String | Local | XXXXXXXXXX | Reads the selected colour and applies it |
| arrCellColour | Array | Local | [XXX,XXX] | Stores all the different colour options for cells |
| strSelectedTextColour | String | Local | XXXXXXXXXX | Reads the selected colour and applies it |
| arrStats | Array | Local | [XXX,XXX] | Reads stats for chosen column |
| arrThemes | Multidimensional array | Global | [[XXX],[XXX]] | Stores all the different theme options available |
| arrChosenTheme | Array | Global | [XXX,XXX] | Stores the current theme data |
| strSearchResults | String | Local | XXXXXXXXXX | Stores the results from the search |

Object Description  
Design 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Method/Event | Properties | Purpose |
| cbo[colname] | Combo box | Method | Arial, 12pt  Readonly: False | The combo boxes where data inputted is saved |
| btnAddColumn | Button | Event: Click | Arial, 12pt, “Add” | Adds the inputted string from txtAddColumn as a new column in dgdmain |
| txtAddColumn | Textbox | Method | Arial, 12pt  Readonly: False, “” | Allows the user to input a new column |
| btnUpdateColumn | Button | Event: Click | Arial, 10pt, “Save” | Updates the selected cboSelectColumn column |
| cboSelectColumn | Combo box | Method | Arial, 12pt  Readonly: True, “” | Allows the user to select a column to be updated |
| optSelectTextcolour | Radio button | Method | Arial, 12pt, “” | Allows the user to select a text colour |
| optSelectCellcolour | Radio button | Method | Arial, 12pt, “” | Allows the user to select a cell colour |
| cboSearchSelect | Combo box | Method | Arial, 12pt, “” | Allows the user to select a column that will be searched |
| txtSearchInput | Text box | Method  Readonly: False | Arial, 12pt, “” | Reads the user input search for cboSelectTextColour |
| btnSearch | Button | Event: Click | Arial, 10pt, “Search” | Searches all items within selected column and shows all other items in the cells adjacent |
| mnuMain | Menu Strip | Method | Arial, 12pt, “” | Stores and displays all the menu functions |
| mnuFileSave | Menu Strip Button | Event: Click | Arial, 12pt, “Save” | Saves the current class to an XML file |
| mnuFileSaveAll | Menu Strip Button | Event: Click | Arial, 12pt, “Save All” | Saves all classes to an XML file |
| mnuFileNew | Menu Strip Button | Event: Click | Arial, 12pt, “New” | Creates a new class with corresponding data grid |
| mnuFileOpen | Menu Strip Button | Event: Click | Arial, 12pt, “Open” | Opens (imports) a previously saved file |
| mnuEditSort | Menu Strip Button | Event: Click | Arial, 12pt, “Sort” | Sorts chosen column numerically or alphabetically |
| mnuViewTheme | Menu Strip Button | Event: Click | Arial, 12pt, “Theme Settings” | Allows user to change the software theme |
| mnuEditStatistics | Menu Strip Button | Event: Click | Arial, 12pt, “Statistics” | Displays statistics of a chosen column on lstStats |
| lstStats | List box | Method  Readonly: True | Arial, 10pt, “” | Displays all the past stats that has been calculated |
| btnAddData | Button | Event: Click | Arial, 10pt, “Add” | Adds inputted data to specified column |
| txtAddData | Textbox | Method | Arial, 10pt, “” | Allows user to input data that will be appended to one of the columns |
| cboAddDataCol | Combo box | Method | Arial, 10pt, “Add to”  Readonly: True | Allows user to select which column to input data to |

Design 2 Mock up

Button – btnSearch   
Searches all items within selected column and shows all other items in the cells adjacent

Textbox – txtSearchInput  
Reads the user input search for cboSelectTextColour

Combo box – cboSearchSelect   
Allows the user to select a column that will be searched

Button – btnAddColumn   
Adds the inputted string from txtAddColumn as a new column in dgdmain

Textbox – txtAddColumn  
Allows the user to input a new column

Combo box – cboAddColunm   
Allows the user to select a column to be updated

Button – btnUpdateColumn  
Updates the column name

Textbox – txtUpdateColumn  
Allows the user to input the new nae of their selected column

Combo box – cboSelectColumn  
Allows the user to select a column that will be renamed

List box – lstStats   
Displays all previous shown searches

Menu strip item – “File”  
Contains all software file functionality

Menu strip item – “Edit”  
Contains all software editing functionality

Menu strip item – “View”  
Contains all editorial functionality

A screenshot of a computer screen

Description automatically generated with low confidence

Radio button – optSelectCellColour   
Allows user to select the colour of the cell they are currently editing   
  
Radio button – optSelectTextColour   
Allows user to select the colour of their text

All text **arial black** by default

All buttons **dark pastel blue**

All combo boxes **pastel blue**

List box **dark grey**

Background is **grey**

All small text 10pt

All median text 12pt

All large text (header) 18pt

Datagrid cells **white** by default with **black text** 10pt

Menustrip is **white**

Datagridview – dgdMain  
Displays the class of which the user is currently editing as well as all the data contained

Button – Exit  
Closes the program

Data Dictionary   
Design 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Scope (Local/Global) | Format | Purpose |
| blnLoggedIn | Boolean | Global | True/False | Stores whether the user has successfully logged in |
| arrClass[classname] | Multidimensional array | Global | [[XXX],[XXX]] | Stores each individual class within the database |
| arrDatabase | Multidimensional array | Global | [[XXX],[XXX]] | Stores all the classes within one array |
| strColumnCount | Integer | Local | XX | Stores the number of columns within the class database |
| strNewColumnName | String | Local | XXXXXXXXXX | Stores the new name for the designated column |
| arrTextColour | Array | Local | [XXX,XXX] | Stores all the different colour options for the text |
| strSelectedTextColour | String | Local | XXXXXXXXXX | Reads the selected colour and applies it |
| arrCellColour | Array | Local | [XXX,XXX] | Stores all the different colour options for cells |
| strSelectedCellColour | String | Local | XXXXXXXXXX | Reads the selected colour and applies it |
| arrStats | Array | Local | [XXX,XXX] | Reads stats for chosen column |
| arrThemes | Multidimensional array | Global | [[XXX],[XXX]] | Stores all the different theme options available |
| arrChosenTheme | Array | Global | [XXX,XXX] | Stores the current theme data |
| strSearchResults | String | Local | XXXXXXXXXX | Stores the results from the search |

Object Description  
Design 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Method/Event | Properties | Purpose |
| dgdMain | Datagridview | Method | Arial, 10pt  Readonly: False  Allowusertocreaterows: True | Displays and allows user to edit the current class |
| btnExit | Button | Event: Click | Arial, 12pt, “Exit” | Exits the program |
| btnAddColumn | Button | Event: Click | Arial, 12pt, “Add” | Adds the inputted string from txtAddColumn as a new column in dgdmain |
| txtAddColumn | Textbox | Method | Arial, 12pt  Readonly: False, “” | Allows the user to input a new column |
| btnUpdateColumn | Button | Event: Click | Arial, 10pt, “Save” | Updates the selected cboSelectColumn column |
| cboSelectColumn | Combo box | Method | Arial, 12pt  Readonly: True, “” | Allows the user to select a column to be updated |
| optSelectTextcolour | Radio button | Method | Arial, 12pt, “” | Allows the user to select a text colour |
| optSelectCellcolour | Radio button | Method | Arial, 12pt, “” | Allows the user to select a cell colour |
| cboSearchSelect | Combo box | Method | Arial, 12pt, “” | Allows the user to select a column that will be searched |
| txtSearchInput | Text box | Method  Readonly: False | Arial, 12pt, “” | Reads the user input search for cboSelectTextColour |
| btnSearch | Button | Event: Click | Arial, 10pt, “Search” | Searches all items within selected column and shows all other items in the cells adjacent |
| mnuMain | Menu Strip | Method | Arial, 12pt, “” | Stores and displays all the menu functions |
| mnuFileSave | Menu Strip Button | Event: Click | Arial, 12pt, “Save” | Saves the current class to an XML file |
| mnuFileSaveAll | Menu Strip Button | Event: Click | Arial, 12pt, “Save All” | Saves all classes to an XML file |
| mnuFileNew | Menu Strip Button | Event: Click | Arial, 12pt, “New” | Creates a new class with corresponding data grid |
| mnuFileOpen | Menu Strip Button | Event: Click | Arial, 12pt, “Open” | Opens (imports) a previously saved file |
| mnuEditSort | Menu Strip Button | Event: Click | Arial, 12pt, “Sort” | Sorts chosen column numerically or alphabetically |
| mnuViewTheme | Menu Strip Button | Event: Click | Arial, 12pt, “Theme Settings” | Allows user to change the software theme |
| mnuEditStatistics | Menu Strip Button | Event: Click | Arial, 12pt, “Statistics” | Displays statistics of a chosen column on lstStats |
| lstStats | List box | Method  Readonly: True | Arial, 10pt, “” | Displays all the past stats that has been calculated |

Design Final Mock up

Menu strip item – “File”  
Contains all software file functionality

Menu strip item – “Edit”  
Contains all software editing functionality

Menu strip item – “View”  
Contains all editorial functionality

A close-up of a check

Description automatically generated with low confidence

All text **arial black** by default

All buttons **dark pastel blue**

All combo boxes **pastel blue**

List box **dark grey**

Background is **grey**

All small text 10pt

All median text 12pt

All large text (header) 18pt

Datagrid cells **white** by default with **black text** 10pt

Menustrip is **white**

Datagridview – dgdMain  
Displays the class of which the user is currently editing as well as all the data contained

List box – lstStats   
Displays all previous shown searches

Button – btnAddColumn   
Adds the inputted string from txtAddColumn as a new column in dgdmain

Textbox – txtAddColumn  
Allows the user to input a new column

Button – Exit  
Closes the program

Button – btnSearch   
Searches all items within selected column and shows all other items in the cells adjacent

Textbox – txtSearchInput  
Reads the user input search for cboSelectTextColour

Combo box – cboSearchSelect   
Allows the user to select a column that will be searched

Combo box – cboSelectCellColour   
Allows user to select the colour of the cell they are currently editing   
  
Combo box – cboSelectTextColour   
Allows user to select the colour of their text

Button – btnUpdateColumn  
Updates the column name

Textbox – txtUpdateColumn  
Allows the user to input the new nae of their selected column

Combo box – cboSelectColumn  
Allows the user to select a column that will be renamed

Data Dictionary   
Final:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Scope (Local/Global) | Format | Purpose |
| blnLoggedIn | Boolean | Global | True/False | Stores whether the user has successfully logged in |
| arrClass[classname] | Multidimensional array | Global | [[XXX],[XXX]] | Stores each individual class within the database |
| arrDatabase | Multidimensional array | Global | [[XXX],[XXX]] | Stores all the classes within one array |
| strColumnCount | Integer | Local | XX | Stores the number of columns within the class database |
| strNewColumnName | String | Local | XXXXXXXXXX | Stores the new name for the designated column |
| arrTextColour | Array | Local | [XXX,XXX] | Stores all the different colour options for the text |
| strSelectedTextColour | String | Local | XXXXXXXXXX | Reads the selected colour and applies it |
| arrCellColour | Array | Local | [XXX,XXX] | Stores all the different colour options for cells |
| strSelectedCellColour | String | Local | XXXXXXXXXX | Reads the selected colour and applies it |
| arrStats | Array | Local | [XXX,XXX] | Reads stats for chosen column |
| arrThemes | Multidimensional array | Global | [[XXX],[XXX]] | Stores all the different theme options available |
| arrChosenTheme | Array | Global | [XXX,XXX] | Stores the current theme data |
| strSearchResults | String | Local | XXXXXXXXXX | Stores the results from the search |

Object Description  
Final:

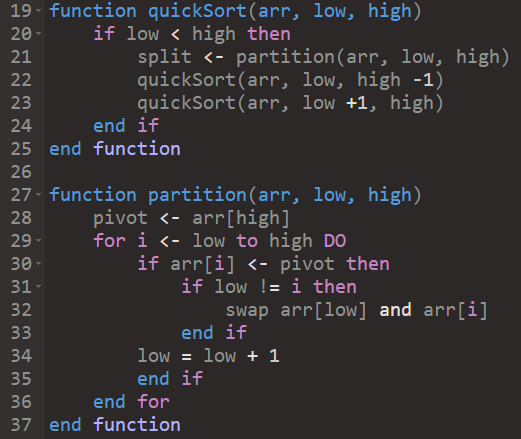
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Method/Event | Properties | Purpose |
| dgdMain | Datagridview | Method | Arial, 10pt  Readonly: False  Allowusertocreaterows: True | Displays and allows user to edit the current class |
| btnExit | Button | Event: Click | Arial, 12pt, “Exit” | Exits the program |
| btnAddColumn | Button | Event: Click | Arial, 12pt, “Add” | Adds the inputted string from txtAddColumn as a new column in dgdmain |
| txtAddColumn | Textbox | Method | Arial, 12pt  Readonly: False, “” | Allows the user to input a new column |
| btnUpdateColumn | Button | Event: Click | Arial, 10pt, “Save” | Updates the selected cboSelectColumn column |
| cboSelectColumn | Combo box | Method | Arial, 12pt  Readonly: True, “” | Allows the user to select a column to be updated |
| cboSelectTextcolour | Combo box | Method Readonly: True | Arial, 12pt, “” | Allows the user to select a text colour |
| cboSelectCellcolour | Combo box | Method | Arial, 12pt, “” | Allows the user to select a cell colour |
| cboSearchSelect | Combo box | Method | Arial, 12pt, “” | Allows the user to select a column that will be searched |
| txtSearchInput | Text box | Method  Readonly: False | Arial, 12pt, “” | Reads the user input search for cboSelectTextColour |
| btnSearch | Button | Event: Click | Arial, 10pt, “Search” | Searches all items within selected column and shows all other items in the cells adjacent |
| mnuMain | Menu Strip | Method | Arial, 12pt, “” | Stores and displays all the menu functions |
| mnuFileSave | Menu Strip Button | Event: Click | Arial, 12pt, “Save” | Saves the current class to an XML file |
| mnuFileSaveAll | Menu Strip Button | Event: Click | Arial, 12pt, “Save All” | Saves all classes to an XML file |
| mnuFileNew | Menu Strip Button | Event: Click | Arial, 12pt, “New” | Creates a new class with corresponding data grid |
| mnuFileOpen | Menu Strip Button | Event: Click | Arial, 12pt, “Open” | Opens (imports) a previously saved file |
| mnuEditSort | Menu Strip Button | Event: Click | Arial, 12pt, “Sort” | Sorts chosen column numerically or alphabetically |
| mnuViewTheme | Menu Strip Button | Event: Click | Arial, 12pt, “Theme Settings” | Allows user to change the software theme |
| mnuEditStatistics | Menu Strip Button | Event: Click | Arial, 12pt, “Statistics” | Displays statistics of a chosen column on lstStats |
| lstStats | List box | Method  Readonly: True | Arial, 10pt, “” | Displays all the past stats that has been calculated |

File Dictionary

|  |  |  |
| --- | --- | --- |
| Name | Type | Purpose |
| UserLogins.txt | Plain text file | Stores the users login data to check against when the user attempts to log in |
| AllClasses.xml | XML file | Stores all the individual classes within the database in one larger file |
| [classname]Class.xml | XML file | Stores all the individual classes as separate smaller files |
| ThemeSettings.csv | CSV file | To be created once the user adds any customization to their software them. Stores theme data such as colour hex codes. |

Pseudocode  
Final:

Sort:

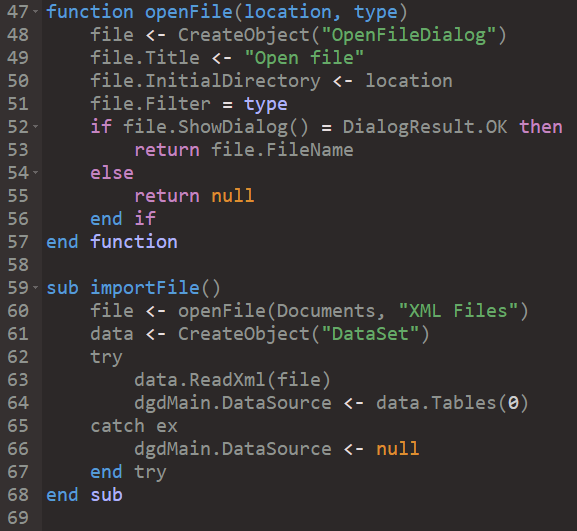


Search:

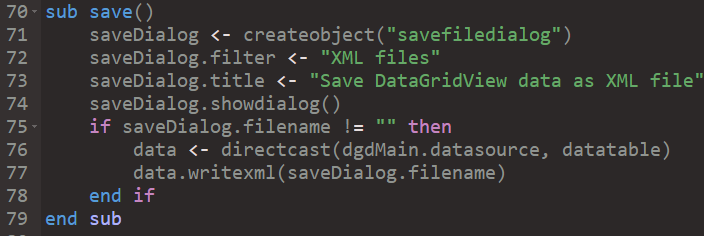
A picture containing text, screenshot, font

Description automatically generated

Open and Import:



Save:



Justification

Design 1:

* Meets most functional requirements in the SRS at a decent standard
* Meets most non-functional requirements in the SRS however at a low standard
* Complies with most restraints in the SRS
* Meets 70% of areas covered by scope

Pros:

* Clear search method
* Easy to select colours with radio button
* Adding and editing columns is easy

Cons:

* Method to add and view data is too clunky and very hard to understand
* Would consume more CPU% due to extra processes running
* Limited colours available due to amount of space extra radio buttons would take up
* Generally hard to read and use

Design 2:

* Meets all functional requirements in the SRS at a decent standard
* Meets most non-functional requirements in the SRS at a decent standard
* Complies with all restraints in the SRS
* Meets 90% of areas covered by scope
* Generally higher in efficiency

Final:

* Meets all functional requirements in the SRS at a high standard
* Meets most non-functional requirements in the SRS at a good standard
* Complies with all restraints in the SRS
* Meets all areas covered by scope
* Decided to follow design 2 closely as it scored much higher than design 1
* Main flaws of design 2 were readability which were improved in the final
* Kept datagrid from design 2 as it would be much easier to use than many combo boxes acting as columns
* Swapped out radio buttons for colours with combo boxes; provides more colour options and less screen clutter
* Overall very similar to design 2 just slightly higher level of functionality and usability

Evaluation criteria

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Requirement | Strategies | Effectiveness & Efficiency |
| EC01 | (FR01) The user can login to access the rest of the system | Attempt login with admin pass   Observe Client logging in using their user info | Effectiveness:   * Reads inputted passwords correctly? * Easy to read?   Efficiency:   * How long does user input take to process? * Is there timely feedback to show that the user’s details are correct/incorrect? |
| EC02 | (FR02) The user can quickly create a ‘class’ with unique parameters | The user is to create a class with many different parameters  Observe client and make note of any difficulties  Measure the time It takes to create a new class | Effectiveness:   * Easy to create a new class with unique parameters?   Efficiency:   * How long does it take to create a new class? |
| EC03 | (FR03) The user can quickly input student data into previously created or imported classes and modify it | Observe client inputting and modifying data and make note of any difficulties   Measure the time it takes to input and change data  Interview client about their experience adding and modifying data | Effectiveness:   * Easy to input and modify data?   Efficiency:   * Is the method of data input quick and intuitive? * Is this solution faster than the original? |
| EC04 | (FR04) Data inputted into class parameters can be sorted and searched | Observe client sorting and searching data and make note of any difficulties  Measure the time it takes to sort and search data  Interview client about their experience sorting and searching data | Effectiveness:   * Is data sorted correctly? * Search results accurate and relevant?   Efficiency:   * Sorting process quick and efficient? * Search function fast and responsive? |
| EC05 | (FR05) Data can be saved or imported | The user is to test any relevant file functionality  Inspect the contents within saved files | Effectiveness:   * Does the system save data accurately and reliably? * Does the importing feature read previously exported data correctly?   Efficiency:   * How long does it take to save and import data? |
| EC06 | (FR06) Easy ability to switch in-between classes | User is to switch between many preimported classes and investigate the accuracy of each class switched | Effectiveness:   * Is it easy to navigate between different classes? * Does the system retain the progress when switching?   Efficiency:   * Is the class switching process quick and lag free? |
| EC07 | (NFR01) Easy to use and understand with clean and simple UI | Interview client about their thoughts and experience with the UI | Effectiveness:   * Is the UI intuitive and easy to navigate? * Does it provide clear instructions and visual cues?   Efficiency:   * Is the UI smooth during interactions? |
| EC08 | (NFR02) Calculations and stats shown by the software are correct | The user is to test a range of calculations and stat analysis | Effectiveness:   * Are the calculations and stat analysis 100% accurate and error-free? * Do the stats and calculations provide useful analysis?   Efficiency:   * Are the calculations performed within an acceptable time? * Does it impact system performance heavily when performing larger calculations? |
| EC09 | (NFR03) The software can run-on low-end devices without using more than 30% CPU | Test the software on a lower end device and measure the time taken to perform the functionalities  Interview client and inquire about their satisfaction of the software on their device | Effectiveness:   * Does the software run smoothly on low-end devices? * Is the functionality fully accessible without significant performance issues?   Efficiency:   * Does the software stay under the CPU usage limit of 30% on lower end devices? |