Project title

Autor: Antonio Jose Donis Hung

Teacher: Ingeniera Diana Teresa Gómez Forero

Universidad Pontificia Bolivariana Seccional Bucaramanga

2020

Purpose

System scope

User characteristics

Restrictions

Requirements

|  |  |
| --- | --- |
| FR01 | Registry submission dialog |
| Description | The user can add new registries. To the database.   |  |  | | --- | --- | | Column name | Description | | Id | This is invisible to the user and will only help index registries inside the database. | | Date | This field is user controllable and by default it is selected the current day. | | Category | Is user controllable and by default it selects the last category used. | | Type | This field represents if the amount is a money income or a money outcome. | | Amount | This is a numeric field that the user enters. | | Currency | This is the currency which were the amount of money is represented, by default is used the system currency. | | Description | This is a description for the registry, it can be empty. |   Check the inputs are valid.  Add the inputs to a database table called “registries”. |
| Actor | User. |
| Wanted results | The registry will be added to the database table “registries”. |
| User inputs | * Date. * Category. * Type. * Amount. * Currency. * Description. |
| Process | * Capture the inputs inside a view dialog. * Id is selected by the same database. * Date is by default selected by the system (today) or can be customized by the user and is represented in a string Year/Month/Day. * Category will be going to be selected by the user inside other view dialog. * Type is selected from a selection field were the options are “Income” and “Outcome”. * Amount is a numeric value which can be controlled by the user and it can’t be a negative value. * Currency will be going to be selected by the user inside other view dialog. * Description is user controllable and can by default is empty. (Optional Field). * Check the that the inputs are valid. * Date is with the format Year/Month/Day and it does not contain an SQL injection. * Category is inside the table “categories” and it does not contain an SQL injection. * Type is “Income” or “Outcome” and it does not contain an SQL injection * Amount is positive. * Currency is in the currency’s files and it does not contain an SQL injection. * Description does not contain an SQL injection. * If input is not valid show a Dialog that shows a “Invalid input” and after closing it, maintains the view. * If input is valid, add the validated input to the database table “registries” and close the dialog. |

|  |  |
| --- | --- |
| FR02 | Top menu bar |
| Description | The user has a menu bar with the next options.   |  |  | | --- | --- | | Option category | Description | | File | Is the file control option and adds this sub options.   |  |  | | --- | --- | | Sub option name | Description | | New | Creates a new file. | | Open | Opens a existing file. | | Save | Save current file. | | Save as | Save current file in new path. | | Exit | Terminates the program. | |   The program executes the sub option of each available options category. |
| Actor | User |
| Wanted results | The execution of the associated sub option to each option category.   * File * When “New” is selected a dialog is opened to the user and let he select a file path and name. * When “Open” is selected a dialog is opened to the user and let he select a file path and name. * When “Save” is selected, the current file is saved to its original path. * When “Save as” is selected, a dialog is opened to the user where he selects a file path and name to save the file. * When “Exit” is selected the program ends. |
| User inputs | 1. Option category button. 2. Sub option button. 3. Possible spawned dialog. |
| Process | 1. Select the option category. 2. Select the sub option. 3. Show the corresponding dialog for each option, if that option has one, otherwise execute the associated process. |

|  |  |
| --- | --- |
| FR03 | File Options |
| Description | After a working file is path these buttons will be able to change the main view.   |  |  | | --- | --- | | Button Name | Button Description | | Registry | Here registries are showed, the sum of then is showed, and are two buttons, one to delete a selected registry and other to add a new registry | | Search | Here is possible to spawn a dialog with the results of a search in a results dialog, by default user can search for categories, but also there is an “Advance search” feature that spawns a dialog that add more search features. |   Show the wanted view to the user’s main view. |
| Actor | User |
| Wanted results | Show the View when the button is pressed. |
| User inputs | Button press. |
| Process | 1. User press button. 2. Main view is changed to the selected one. |

|  |  |
| --- | --- |
| FR04 | No opened file main view |
| Description | If user don’t open/create a file, by default the main view will always contain this view that have two buttons which are.   |  |  | | --- | --- | | Button Name | Button Description | | New file | Opens a dialog that let user select a file path and name to then work with. | | Open file | Open a dialog that let user select a file path and name to then work with. |   Show this view is no file was specified. |
| Actor | User |
| Wanted results | Show this view, and if the user decides to press one of the buttons inside it, run the corresponding process. |
| User inputs | * “New file” button pressed. * “Open file” button pressed. |
| Process | 1. User press button “New file” button or “Open file” button. 2. Show the corresponding dialog associated with the button pressed. |

|  |  |
| --- | --- |
| FR05 | Registry main view |
| Description | Is presented this view to the user when a file was opened. This view has these features:   |  |  | | --- | --- | | Feature name | Description | | Sum Field | Here is the sum of the entire database translating every “Outcome” to negative values and every currency to the currency selected by the system. | | Currency translation | This is a button that by default show in its text the name of the currency that the system is using, when is pressed a dialog that let the user select other currency is opened and then reloads the view. | | Delete Button | When registries where selected in the table view, this button delete all then from the database and then reloads the view. | | Add button | This button opens a dialog that let user add a new registry to the database and then reload the view. | | Table view | This view shows the last one hundred registries (this to crash prevention) and only shows this column of each database row Date, Category, Amount, Currency; it is possible to select any row of the table view, this will open a new dialog that show all the content of that registry, also every row has a check mark that let multiple selection easier. |   Present this view to the user if he opens a file and when “Registry” button is pressed. |
| Actor | User |
| Wanted results | Show the entire view. |
| User inputs | * Currency button pressed. * “Delete” button pressed. * “Add” button pressed. * Selected table view registry. * Marked check parameter of a table view registry (row). |
| Process | * Currency button.  1. Open a dialog that select let the user select a currency. 2. Reload the view  * “Delete” button.  1. Delete from the database all selected registries. 2. Reload the view.  * “Add” button.  1. Spawn a dialog in that let user create a new registry and insert it in the database. 2. Reload view.  * Selected table view registry (row).  1. Opens a dialog that show the complete information of the registry  * Marked check parameter of a table view registry (row).  1. Add to a list the information of the registry (row). |

|  |  |
| --- | --- |
| FR06 | Search main view |
| Description | Is presented this view to the user when a file was opened. This view has these features:   |  |  | | --- | --- | | Feature name | Description | | Category Field | This is a button that when is pressed a dialog that let the user select a available category is spawned. | | Search | This button approaches the category selected and then make a Search inside the database; it spawns the results in a new dialog. | | Advance search | This button spawns a new dialog that let user add more search filters. |   Present this view to the user if he opens a file and when “Registry” button is pressed. |
| Actor | User |
| Wanted results | Show the entire view. |
| User inputs | * Category button pressed. * “Search” button pressed. * “Advance search” button pressed. |
| Process | * Category button pressed.  1. Opens a dialog that let user select an available category. 2. Change the “Category” title of the button to the name of the category.  * “Search” button pressed.  1. Run the search. 2. Spawn a dialog with the results.  * “Advance search” button pressed.  1. Spawn the advance search dialog. |

|  |  |
| --- | --- |
| FR07 | View registry dialog |
| Description | The user can see the content of a registry   |  |  | | --- | --- | | Column name | Description | | Date | The user sees what the Date column contains. | | Category | The user sees what the Category column contains. | | Type | The user sees what the Type column contains. | | Amount | The user sees what the Amount column contains. | | Currency | The user sees what the Currency column contains. | | Description | The user sees what the Description column contains. |   Also, it has a button that let the user spawn a new dialog that let him edit the registry (row).   |  |  | | --- | --- | | Button name | Description | | Edit | Let user edit the registry |     The user can see what each registry column has  The user can spawn a new dialog that let edit it |
| Actor | User. |
| Wanted results | Show the columns Date, Category, Type, Amount, Currency and Description in this view |
| User inputs | * Edit button |
| Process | * Show the columns Date, Category, Type, Amount, Currency and Description in this view |

|  |  |
| --- | --- |
| FR08 | Registry edition dialog |
| Description | The user can edit and existing registry.   |  |  | | --- | --- | | Column name | Description | | Id | This is invisible to the user and will only help index registries inside the database. | | Date | This field is user controllable and by default it is selected the current day. | | Category | Is user controllable and by default it selects the last category used. | | Type | This field represents if the amount is a money income or a money outcome. | | Amount | This is a numeric field that the user enters. | | Currency | This is the currency which were the amount of money is represented, by default is used the system currency. | | Description | This is a description for the registry, it can be empty. |  |  |  | | --- | --- | | Button name | Description | | Done | Terminates the edition | | Cancel | Cancel the edition |   Check the inputs are valid.  Add the inputs to a database table called “registries”. |
| Actor | User. |
| Wanted results | The registry will be added to the database table “registries”. |
| User inputs | * Date. * Category. * Type. * Amount. * Currency. * Description. |
| Process | * Capture the inputs inside a view dialog. * Id is selected by the same database. * Date is by default selected by the system (today) or can be customized by the user and is represented in a string Year/Month/Day. * Category will be going to be selected by the user inside other view dialog. * Type is selected from a selection field were the options are “Income” and “Outcome”. * Amount is a numeric value which can be controlled by the user and it can’t be a negative value. * Currency will be going to be selected by the user inside other view dialog. * Description is user controllable and can by default is empty. (Optional Field). * Check the that the inputs are valid. * Date is with the format Year/Month/Day and it does not contain an SQL injection. * Category is inside the table “categories” and it does not contain an SQL injection. * Type is “Income” or “Outcome” and it does not contain an SQL injection * Amount is positive. * Currency is in the currency’s files and it does not contain an SQL injection. * Description does not contain an SQL injection. * If input is not valid show a Dialog that shows a “Invalid input” and after closing it, maintains the dialog. * If input is valid, add the validated input to the database table “registries”, erase the original registry (row) and close the dialog. |

|  |  |
| --- | --- |
| FR09 | Select currency Dialog |
| Description | This dialog let user select a currency from the currencies reference file.  It has inputs:   |  |  | | --- | --- | | Input name | Description | | Currency name | This is a text box were user can put a query to search. | | Search | This button uses the query specified before and update the table view with the results found. |   Output fields:   * There is only one output field and it is the table, and its column are  |  |  | | --- | --- | | Column name | Description | | Currency | Show the currency name. | | Rate | Show it’s rate vs the USD. | |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * Query * Search button |
| Process | * If no query was specified.  1. Show first one hundred entries.  * If query was specified a “Search” button pressed.  1. Show entries related to the query. |

|  |  |
| --- | --- |
| FR10 | Select category dialog |
| Description | This dialog let user select, add or delete a category from the categories table  its inputs are:   |  |  | | --- | --- | | Input name | Description | | Delete | This button deletes each select category and all created entries with that category from the database. | | Add | Spawn a dialog that let user create a new category. | | Query | This query let user search for categories. | | Search | This button executes the search using the given query. | | Check box | Add category to a list that then will let the user delete those rows. |   its outputs are:   |  |  | | --- | --- | | Output name | Description | | Table | This table has the last one hundred categories added to the database.  Its column are:   |  |  | | --- | --- | | Column name | Description | | Category | This is the name of the category | | |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * “Delete” button * “Add” button * Query * “Search” button * Check box from table |
| Process | * “Delete” button  1. Delete all the selected rows from the table 2. Reload the Category dialog  * “Add” button  1. Spawn a dialog that let user create a user 2. Reload the category dialog  * “Search” button  1. Validate the query to check if there is not SQL injection 2. Search for categories related to the query 3. Reload table with results  * Check box  1. Add the selected category to a list to later work with it using the delete button. |

|  |  |
| --- | --- |
| FR11 | Search results dialog |
| Description | This dialog let user interact with the results that were found.  its inputs are:   |  |  | | --- | --- | | Input name | Description | | Delete | This button deletes each selected row and reload the table. | | Translation Currency | This button let user select a currency to work with, translating each row to the one selected (by default is the previously selected by the system) | | Check box | This is a field in the table that let user select a registry to then delete it with the delete button |   its outputs are:   |  |  | | --- | --- | | Output name | Description | | Sum | This field show the sum of all entries in the field and translate is amount to the system currency | | Table | This table has the only one hundred results.  Its column are:   |  |  | | --- | --- | | Column name | Description | | Date | This is the date of the entry | | Category | This is the category of the entry | | Amount | This is the amount of the entry | | Currency | This is the currency of the entry | |   Show the results of the search. |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * “Delete” button * Translation Currency * Check box |
| Process | * “Delete” button  1. Delete all the selected rows from the table 2. Reload the Results dialog  * Translation currency button  1. Spawn a dialog that let user create a user 2. Reload the category dialog  * Check box  1. Add the selected category to a list to later work with it using the delete button. |

|  |  |
| --- | --- |
| FR12 | Advance search dialog |
| Description | In this dialog is where the advance search happens. |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * Category * Currency * Amount From * Amount to * Type * From date * To date * Search button |
| Process | * Using all that parameters, the program will make a search and return in the Search results dialog the rows in the database related to the parameters specified. |

|  |  |
| --- | --- |
| FR13 | Invalid input Dialog |
| Description | In this dialog alert the user that is input in a specific parameter is invalid (contain SQL injection) |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * Ok Button |
| Process | * Show a dialog which the parameter that could have an SQL injection * The Ok button close the dialog |

|  |  |
| --- | --- |
| FR14 | No results Dialog |
| Description | This dialog is spawned when no results where found by the search function |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * Ok Button |
| Process | * Show a dialog saying that no results could be founded. * The Ok button close the dialog |

|  |  |
| --- | --- |
| FR14 | New category dialog |
| Description | Dialog that let user input the new category name |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * Category name (text box) * “Cancel” button * “Done” button |
| Process | * “Cancel” button  1. Cancel the category creation 2. Close dialog  * “Done” button  1. Check the Category name for SQL injection 2. Add category name in the database 3. Close dialog |

|  |  |
| --- | --- |
| FR14 | Delete confirmation Dialog |
| Description | This dialog is spawned when the user wants to delete anything |
| Actor | User |
| Wanted results | Show the entire dialog. |
| User inputs | * “No” button * “Yes” button |
| Process | * “No” button  1. Cancel the deletion  * “Yes” button  1. Confirm the deletion |