**Gentlix Bank – Documentation**

Visual interface functions:

**int main(int argc, char \*argv[])** Create the GTK Application, activate the main window for it and return the run status.

* *int argc* - represents the number of command-line arguments passed to the program and is required for GTK Library
* *char \*argv[]* - representing the command-line arguments. Each element of the array is a string containing one of the command-line arguments.
* Return application\_status after activate.

**void activate\_main\_menu(GtkApplication \*app, gpointer data)** Create main window and it's widgets. Here are the first options when you open the app.

* *GtkApplication \*app* - provides a reference to the GTK application object, which you can use to interact with the GTK application and perform various operations through it. Here is used only to create the main window for the linked app for int main.
* *gpointer data* - is a generic pointer that can be used to pass additional data to the callback function. Here is passed a NULL one.

**void show\_register\_interface(GtkWidget \*widget, gpointer data)** Create the register interface with input fields.

* *GtkWidget \*widget* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the window from which this menu was opened to hide it

**void show\_login\_interface(GtkWidget \*widget, gpointer data)** Create the login interface with input fields.

* *GtkWidget \*widget* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the window from which this menu was opened to hide it

**void show\_account\_interface()** Create new window with all account options after the user is loggeed.

**void show\_edit\_account\_interface(GtkApplication \*app, gpointer data)** Create the edit account interface with same fields as register one but blank ones are ignored.

* *GtkWidget \*widget* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the window from which this menu was opened to hide it

**void show\_all\_transactions\_interface(GtkWidget \*widget, gpointer data)** Create the all transactions interface and a grid table with all transactions for logged user.

* *GtkWidget \*widget* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the window from which this menu was opened to hide it

**void show\_new\_transaction\_interface(GtkWidget \*widget, gpointer data)** Create the new transaction interface with fields for informations and a menu which transaction types.

* *GtkWidget \*widget* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the window from which this menu was opened to hide it

Service functions:

**void create\_an\_account(GtkApplication \*app, gpointer data)** Manage the inputs from register interface.

* *GtkApplication \*app* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the location inside the memory for inputs

**void login\_to\_an\_account(GtkApplication \*app, gpointer data)** Manage the inputs from login interface.

* *GtkApplication \*app* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the location inside the memory for inputs

**void make\_a\_payment(GtkWidget \*widget, gpointer data)** Manage the inputs from new trancation interface and work with them inside the memory to remove money from user account.

* *GtkApplication \*app* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the location inside the memory for inputs

**void make\_a\_transaction(GtkWidget \*widget, gpointer data)** Manage the inputs from new trancation interface and work with them inside the memory to transfer money from user account to another one.

* *GtkApplication \*app* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the location inside the memory for inputs

**void withdraw\_from\_balance(GtkWidget \*widget, gpointer data)** Manage the inputs from new trancation interface and work with them inside the memory to remove money from user account.

* *GtkApplication \*app* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the location inside the memory for inputs

**void add\_to\_balance(GtkWidget \*widget, gpointer data)** Manage the inputs from new trancation interface and work with them inside the memory to add money in user account.

* *GtkApplication \*app* - isn't used inside the function but is required from previous gtk function call
* *gpointer data* - provides the location inside the memory for inputs

Check functions:

* int check\_date\_for\_transaction(const gchar \*day, const gchar \*month, const gchar \*year)
* short check\_available\_date(const gchar \*day, const gchar \*month, const gchar \*year)
* short check\_available\_date\_with\_bypass(const gchar \*day, const gchar \*month, const gchar \*year)
* int available\_account\_type(const gchar \*checked\_account\_type)
* int password\_differ(const gchar \*password1, const gchar \*password2)
* int string\_have\_only\_digits(const gchar \*checked\_string)
* int string\_have\_only\_digits\_extended(const gchar \*checked\_string)
* int string\_have\_only\_letters(const gchar \*checked\_string)
* int account\_tag\_is\_used(const gchar \*checked\_tag)

Domain:

**struct account** - Store user’s account

* int transaction\_index
* float account\_balace
* char tag[20]
* char first\_name[16]
* char second\_name[16]
* char password[32]
* char type[16]
* char phone\_number[11]
* struct date\_format birthday
* struct affiliates\_format affiliates[128]
* struct transaction\_format transaction[512]

**struct date\_format** – Store a calendar data

* short day
* short month
* short year

**struct transaction\_format** – Store user’s transaction

* float amount
* char type[16]
* char category[16]
* char description[100]
* struct date\_format date

**struct affiliates\_format** – Store user’s affiliates

* char tag[40]
* char first\_name[20]
* char second\_name[20]
* char activity\_domain[16]
* char phone[11]