

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 logged.

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 transaction 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 transaction 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 transaction 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 transaction 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_balance
- 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]