॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS

INSTITUTE OF TECHNOLOGY

Project Report

ATM Simulator

Submitted By

CS-B (2019-23)

Hitanshu Samantaray : 19070122067

Kenil Patwa : 19070122086

Kavan Batavia : 19070122085

Prakhar Patel : 19070122127

Kashif Majid Dar : 19070122084

Kapish Nikhil Pashine : 19070122080

SYMBIOSIS INSTITUTE OF TECHNOLOGY

(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)

Pune

2019-2020

Abstract

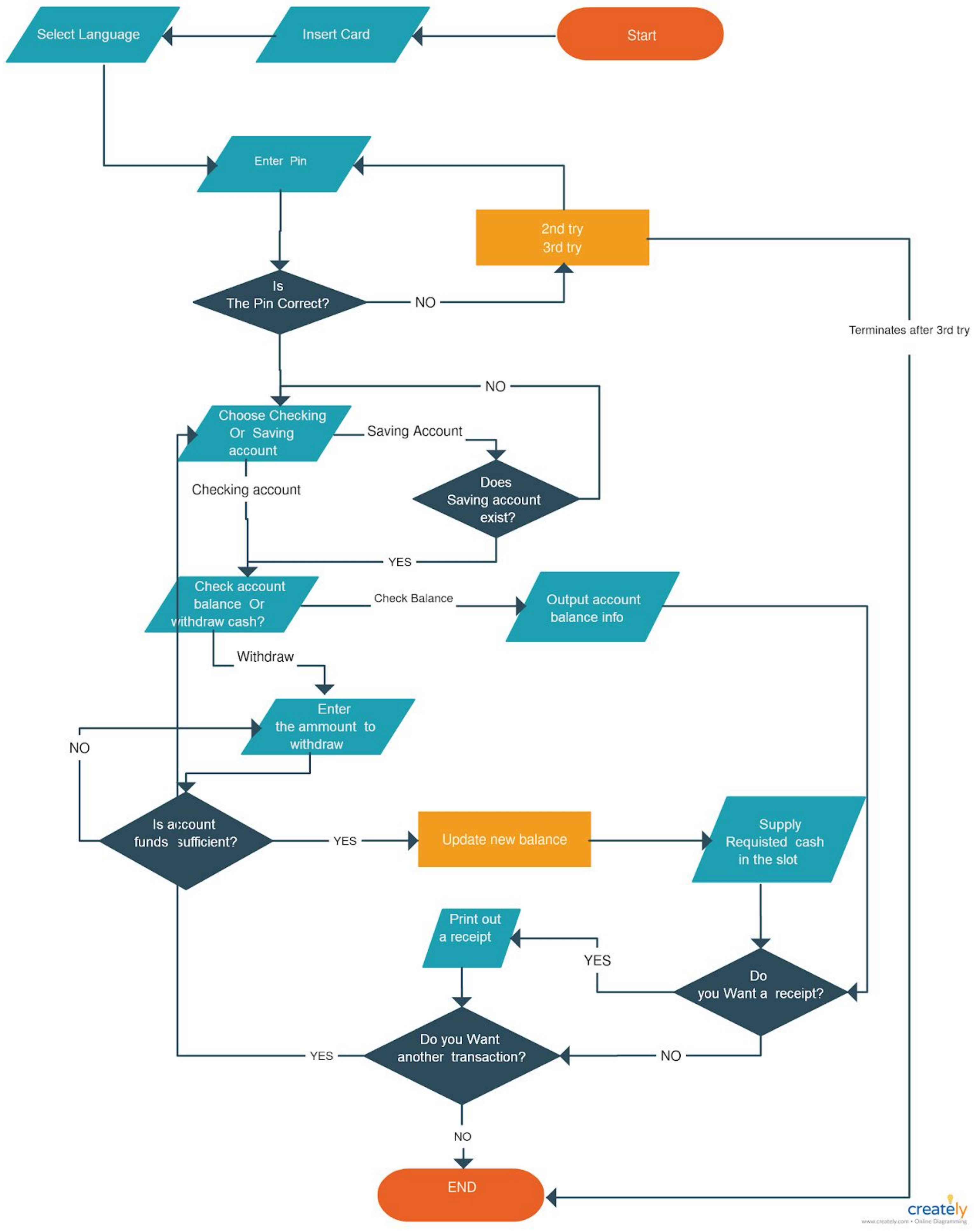
ATM Simulator project is written in C. The project file contains a C script (atm.c). This is a simple GUI system which is very easy to use. Talking about the system, it contains various functions which include Withdraw Cash, Balance Enquiry, Change PIN and Mini Statement. Here, first the user has to insert his/her(cuz feminism) ATM Card. Then the system proceeds toward the next procedure i.e asking the PIN. When a user passes all these sign-in procedures, he/she can use all the banking features . It is really easy to use.

While withdrawing the amount, he/she just has to enter the amount in multiples of 100, then the system calculates the total remaining balance of the respective account and displays it to the user. The user can view all these transactions from the Mini Statement. In this ATM Simulator, the user can also change the PIN. For this, the user has to enter the new PIN and then confirm it in order to change the PIN. This simple GUI-based ATM simulator provides the simple account balance management of the respective account. It contains all the essential features. There is no database connection or either any external text or other files used in this mini project to save user’s data. Everything is set inside the source code whether it's the PIN or the amount.

In order to run the project, you must have installed C and GTK libraries on your PC. This is a simple GUI-Based system, specially written for beginners. ATM Simulator in C project with source code is available on GitHub. For the project demo, have a look at the image slider below.

Algorithm

```
1.  Start
2.  Insert ATM Card
3.  Check Validity of PIN
4.  If PIN matches
        Go to Step 5
    Else
        Display "PIN doesn't match"
5.  Main Menu opens
6.  To Withdraw Cash click Withdraw Cash
7.  Enter Amount to be withdrawn in multiples of 100
8.  Collect Cash
9.  To check current balance click balance enquiry
10. Display "Your balance is __________"
11. To Change PIN click on Change PIN
12. Enter New PIN
13. Enter New PIN again to confirm
14. Submit
15. Display "PIN Changed Successfully"
```



Start
Insert Card
Select Language
Enter Pin
Is The Pin Correct?
NO
2nd try
3rd try
Terminates after 3rd try
NO
Choose Checking Or Saving account
Saving Account
Does Saving account exist?
Checking account
YES
Check account balance Or withdraw cash?
Check Balance
Output account balance info
Withdraw
Enter the ammount to withdraw
NO
Is account funds sufficient?
YES
Update new balance
Supply Requisted cash in the slot
Print out a receipt
YES
Do you Want a receipt?
Do you Want another transaction?
YES
NO
NO
END
creately

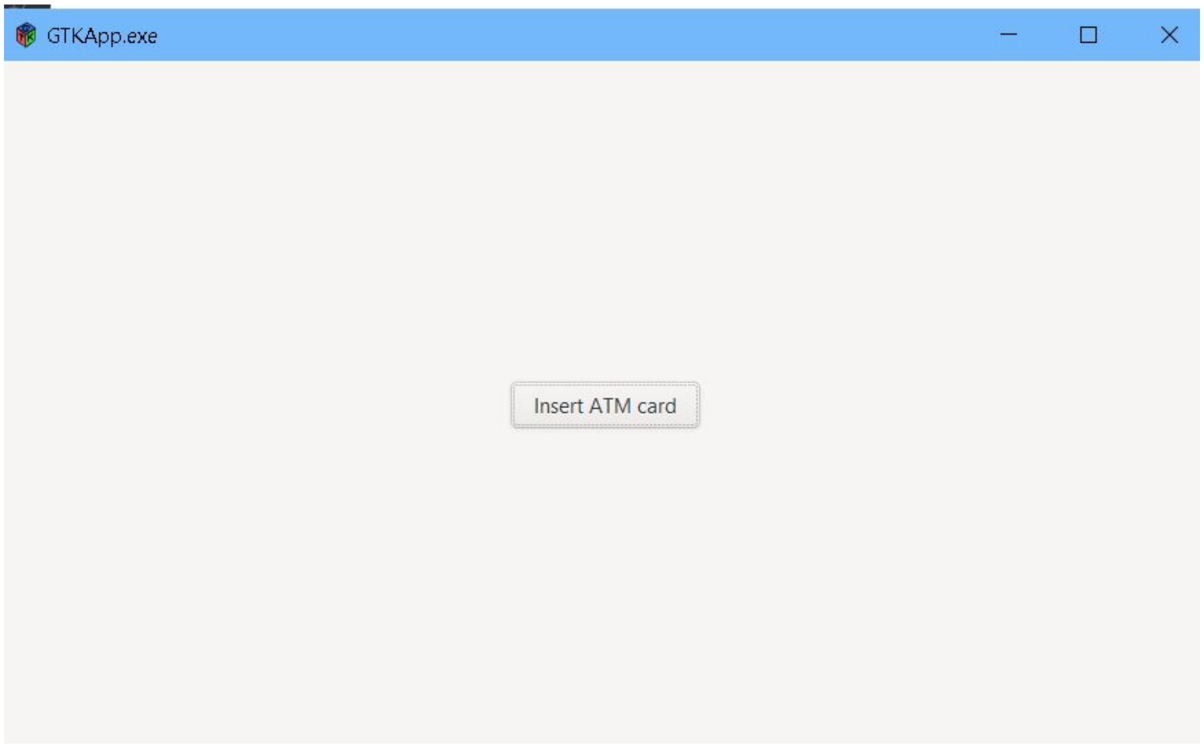
Code

```c
#include <gtk/gtk.h>
#include <stdlib.h>

int pins[100] = {1564, 9856, 3594, 5675, 1568, 3245, 7893, 3131, 6595,
6121};
int bal[100] = {78000, 95000, 65000, 89000, 160520, 256423, 654892, 75362,
548762, 657852};
int index_;
gpointer obj[2];
gpointer gpt;

void show_msg(char msg[]) {
    GtkBuilder *builder;
    GObject *window;
    GObject *lbl;
    GError *error = NULL;

    builder = gtk_builder_new();

    if (gtk_builder_add_from_file(builder, "..\\msg.ui", &error) == 0) {
        g_printerr("Error loading file: %s\n", error->message);
        g_clear_error(&error);
    }
    lbl = gtk_builder_get_object(builder, "lbl");
    gtk_label_set_label(GTK_LABEL(lbl), msg);
}

void ms_menu(GtkWidget *wid, gpointer ptr) {
    show_msg("Collect the receipt");
    gtk_window_close(ptr);
}

void change_pin(GtkWidget *wid, gpointer ptr){
//    GtkEntry *entry1 = g_object_get_data (ptr, "entry1");
//    GtkEntry *entry2 = g_object_get_data (ptr, "entry2");
    int pin1 = atoi(gtk_entry_get_text(GTK_ENTRY (obj[0])));
    int pin2 = atoi(gtk_entry_get_text(GTK_ENTRY (obj[1])));
    printf("%d, %d", pin1, pin2);
    if(pin1 == pin2){
        pins[index_] = pin1;
        show_msg("Pin Changed successfully");
    }
    else{
        show_msg("Pins do not match");
    }
    gtk_window_close(gpt);
}

void cp_menu(GtkWidget *wid, gpointer ptr) {
    GtkBuilder *builder;
    GObject *window ;
    GObject *entry1, *entry2;
    GObject *btn;
    GError *error = NULL;

    builder = gtk_builder_new();

    if (gtk_builder_add_from_file(builder, "..\\CP.ui", &error) == 0) {
        g_printerr("Error loading file: %s\n", error->message);
        g_clear_error(&error);
    }
    window = gtk_builder_get_object(builder, "window");
    entry1 = gtk_builder_get_object(builder, "np1");
    entry2 = gtk_builder_get_object(builder, "np2");
    btn = gtk_builder_get_object(builder, "submit");
    obj[0] = entry1;
    obj[1] = entry2;
//    GObject *context_object = NULL;
//    g_object_set_data (context_object, "entry1", entry1);
//    g_object_set_data (context_object, "entry2", entry2);
    g_signal_connect(btn, "clicked", G_CALLBACK(change_pin), NULL);
    gtk_window_close(ptr);
    gpt = window;
}

void be_menu(GtkWidget *wid, gpointer ptr) {
    GtkBuilder *builder;
    GObject *window;
    GObject *lbl;
    GError *error = NULL;

    builder = gtk_builder_new();

    if (gtk_builder_add_from_file(builder, "..\\BE.ui", &error) == 0) {
        g_printerr("Error loading file: %s\n", error->message);
        g_clear_error(&error);
    }
    lbl = gtk_builder_get_object(builder, "lbl");
    char result[20];
    sprintf(result, "%d", bal[index_]);
    char ch[50] = "Your Balance is:";
    strcat(ch, result);
    gtk_label_set_label(GTK_LABEL(lbl), ch);
    gtk_window_close(ptr);
}

void withdraw_cash(GtkWidget *wid, gpointer ptr) {
    int input = atoi(gtk_entry_get_text(GTK_ENTRY (ptr)));
    if(input % 100 != 0){
        show_msg("Please Enter Amount in multiples of 100");
        return;
    }
    printf("Cash withdrawn: %d from %d", input, index_);
    if(bal[index_] >= input){
        bal[index_] -= input;
        show_msg("Collect your cash");
    }
    else{
        show_msg("insufficient funds!");
    }
    gtk_window_close(gpt);
}

void cw_menu(GtkWidget *wid, gpointer ptr){
    GtkBuilder *builder;
    GObject *window;
    GObject *btn;
    GObject *entry;
    GError *error = NULL;
    builder =gtk_builder_new();

    if (gtk_builder_add_from_file(builder, "..\\CW.ui", &error) == 0) {
        g_printerr("Error loading file: %s\n", error->message);
        g_clear_error(&error);
    }
    window = gtk_builder_get_object(builder, "window");
    entry = gtk_builder_get_object(builder, "entry");
    btn  = gtk_builder_get_object(builder, "submit");
    g_signal_connect(btn, "clicked", G_CALLBACK(withdraw_cash), entry);
    gtk_window_close(ptr);
    gpt = window;
}
```

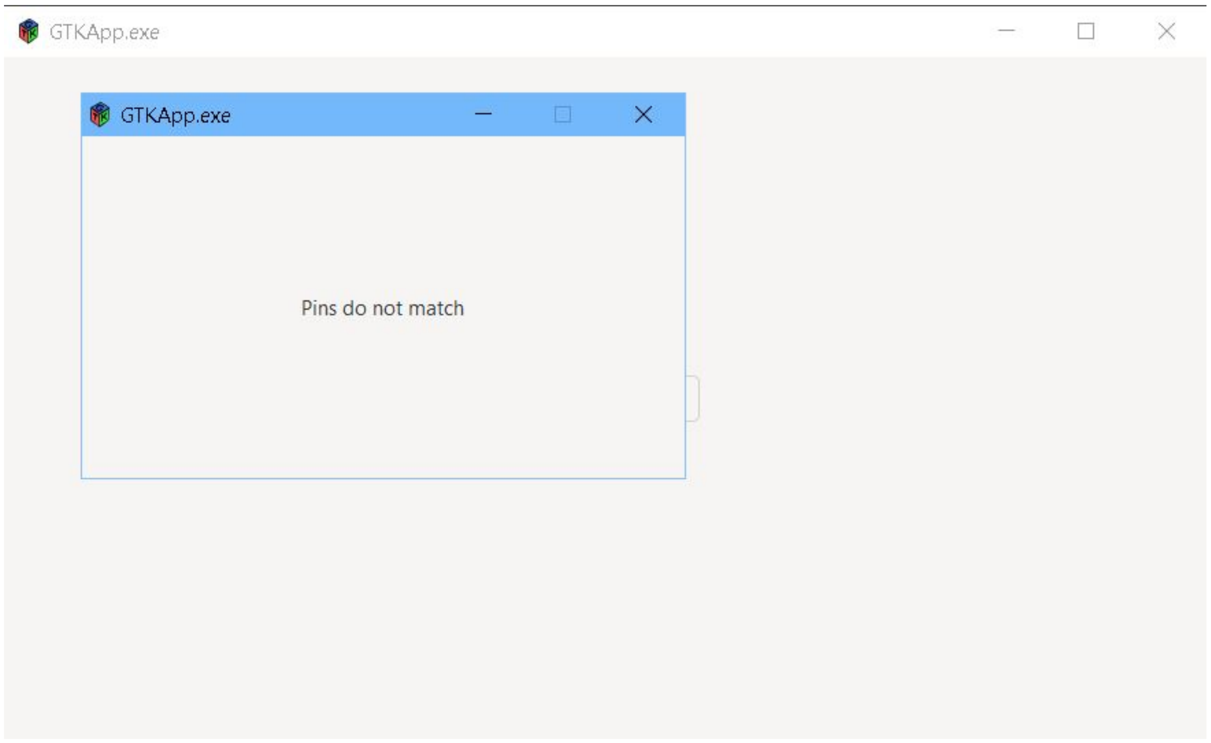
```
void menu() {
    GtkBuilder *builder;
    GObject *window;
    GObject *btn1, *btn2, *btn3, *btn4;
    GError *error = NULL;

    builder = gtk_builder_new();

    if (gtk_builder_add_from_file(builder, "..\\menu.ui", &error) == 0) {
        g_printerr("Error loading file: %s\n", error->message);
        g_clear_error(&error);
    }

    window = gtk_builder_get_object(builder, "window");
    btn1 = gtk_builder_get_object(builder, "CW");
    g_signal_connect(btn1, "clicked", G_CALLBACK(cw_menu), window);
    btn2 = gtk_builder_get_object(builder, "BE");
    g_signal_connect(btn2, "clicked", G_CALLBACK(be_menu), window);
    btn3 = gtk_builder_get_object(builder, "CP");
    g_signal_connect(btn3, "clicked", G_CALLBACK(cp_menu), window);
    btn4 = gtk_builder_get_object(builder, "MS");
    g_signal_connect(btn4, "clicked", G_CALLBACK(ms_menu), window);

}

void check_pin(GtkWidget *wid, gpointer ptr) {
    int input = atoi(gtk_entry_get_text(GTK_ENTRY (ptr)));
    for (int i = 0; i < 100; i++) {
        printf_s("(%d)", pins[i]);
        if (pins[i] == input) {
            index_ = i;
            printf("%d--------",i);
            menu();
            gtk_window_close(GTK_WINDOW(gpt));
            printf("%d", input);
        }
    }

}

static void card_inserted(GtkWidget *widget, gpointer data) {
    GtkBuilder *builder;
    GObject *window;
    GObject *button;
    GObject *entry;
    GError *error = NULL;
    builder = gtk_builder_new();

    if (gtk_builder_add_from_file(builder, "..\\unsaved 1.glade", &error)
== 0) {
        g_printerr("Error loading file: %s\n", error->message);
        g_clear_error(&error);
    }

    window = gtk_builder_get_object(builder, "window");

    entry = gtk_builder_get_object(builder, "pin_entry");

    button = gtk_builder_get_object(builder, "ok");
    g_signal_connect (button, "clicked", G_CALLBACK(check_pin), entry);
    gpt = window;
}

int main(int argc, char *argv[]) {
    GtkBuilder *builder;
    GObject *window;
    GObject *button;
    GObject *entry;
    GError *error = NULL;

    gtk_init (&argc, &argv);

    builder = gtk_builder_new();

    if (gtk_builder_add_from_file(builder, "..\\start_win.glade", &error)
== 0) {
        g_printerr("Error loading file: %s\n", error->message);
        g_clear_error(&error);
        return 1;
    }

    window = gtk_builder_get_object(builder, "window");
    g_signal_connect (window, "destroy", G_CALLBACK(gtk_main_quit), NULL);

    button = gtk_builder_get_object(builder, "insert");
    g_signal_connect (button, "clicked", G_CALLBACK(card_inserted), NULL);

    gtk_main();

    return 0;
}
```

OUTPUT

Insert Card

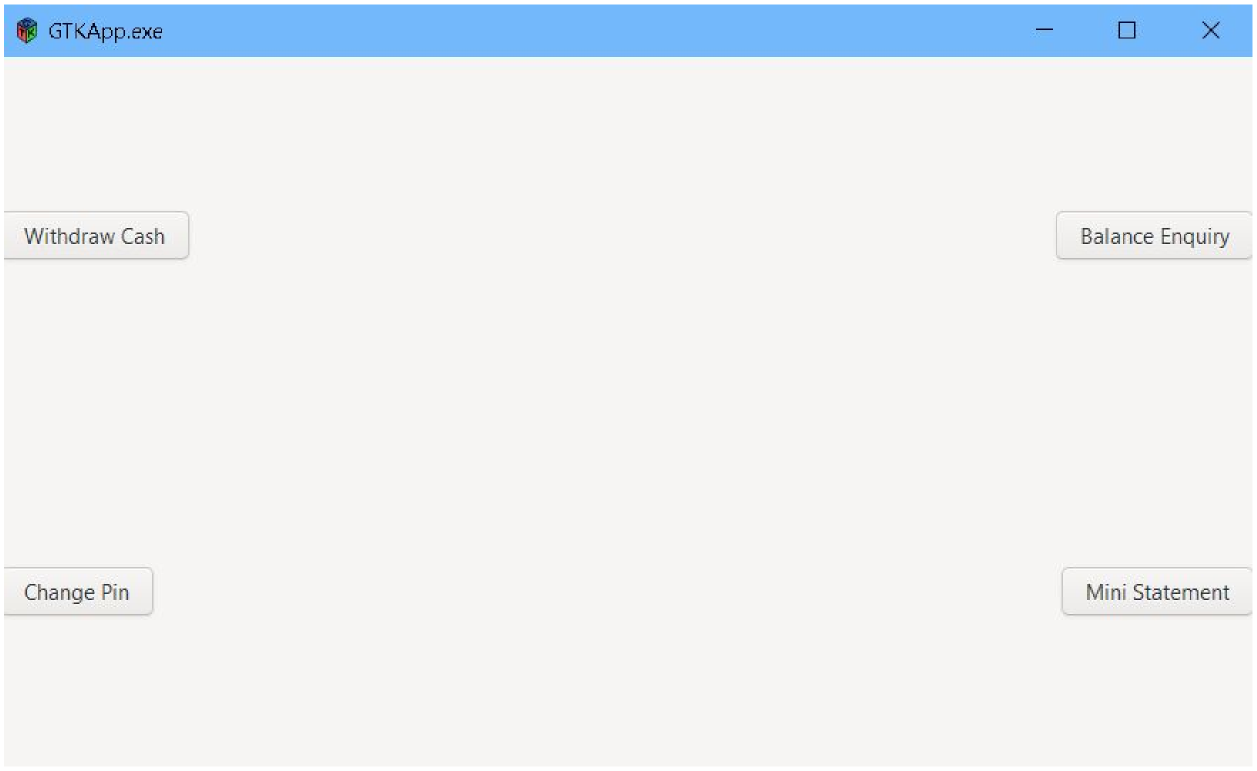


PIN Check

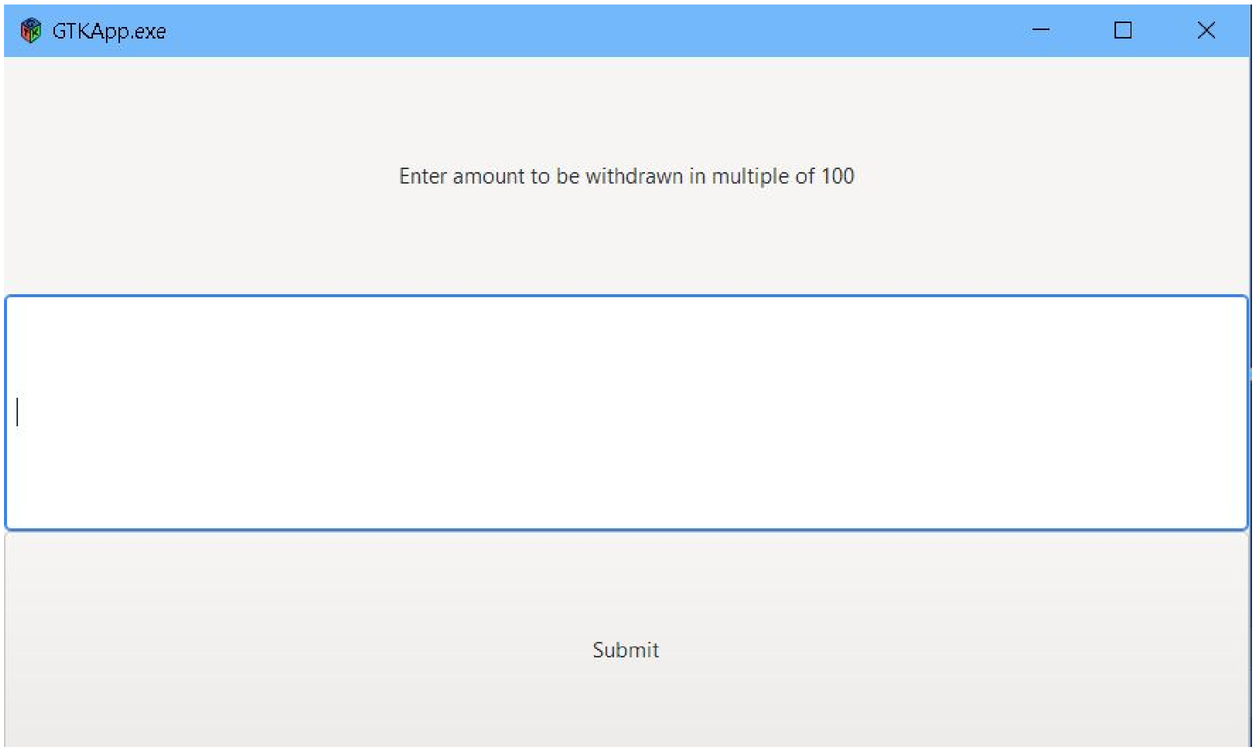


Pins do not match

Main Menu

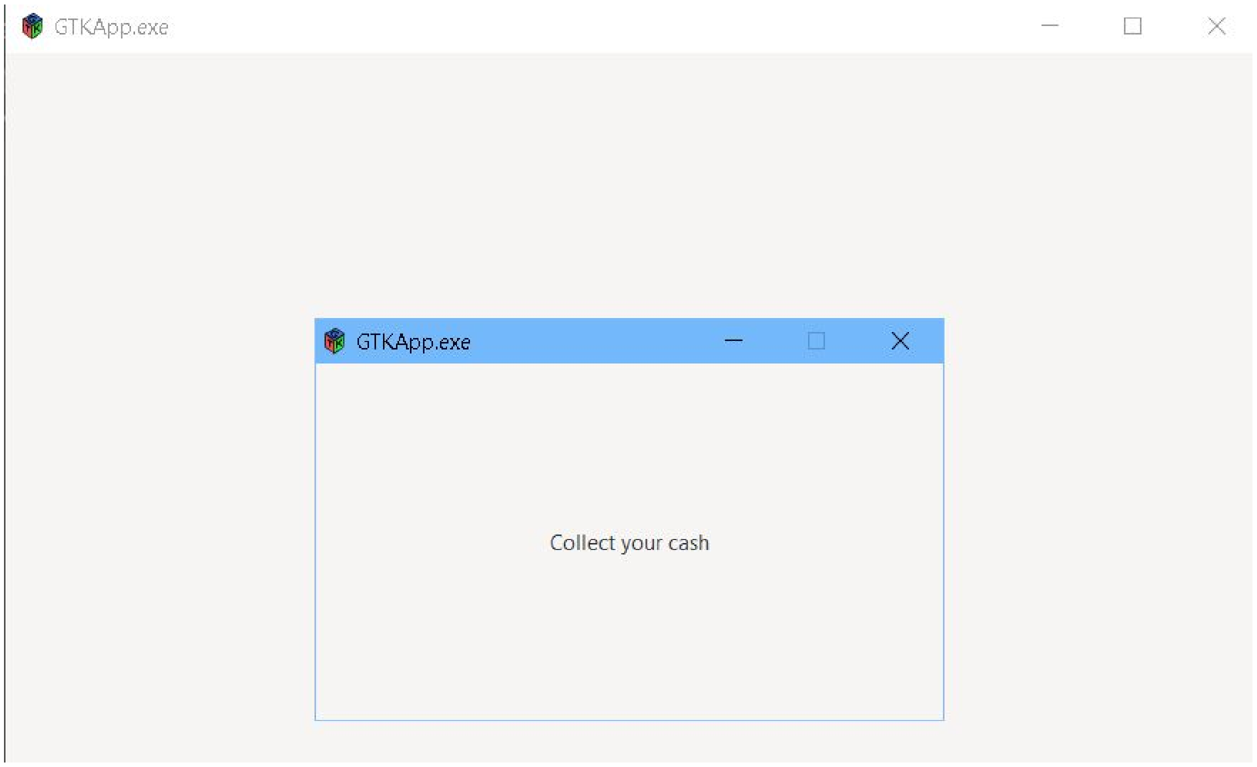


Withdrawal



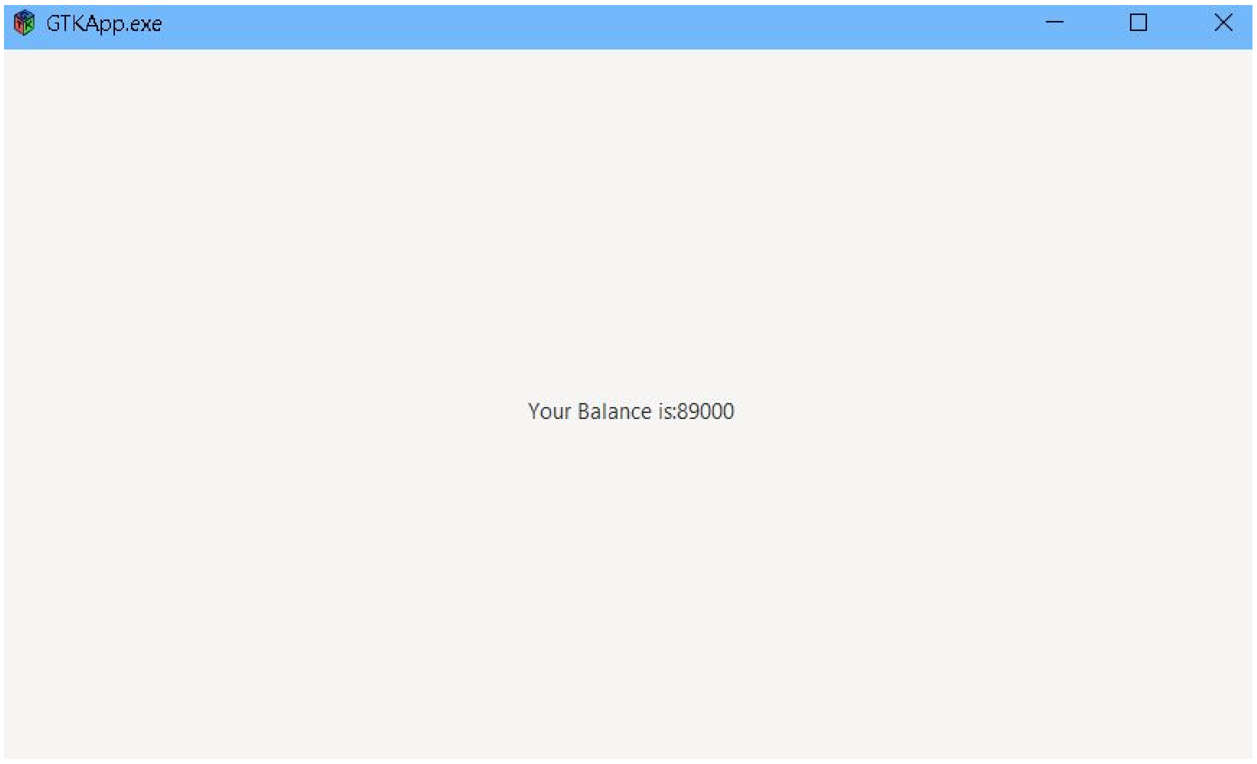
Enter amount to be withdrawn in multiple of 100

Collect Cash



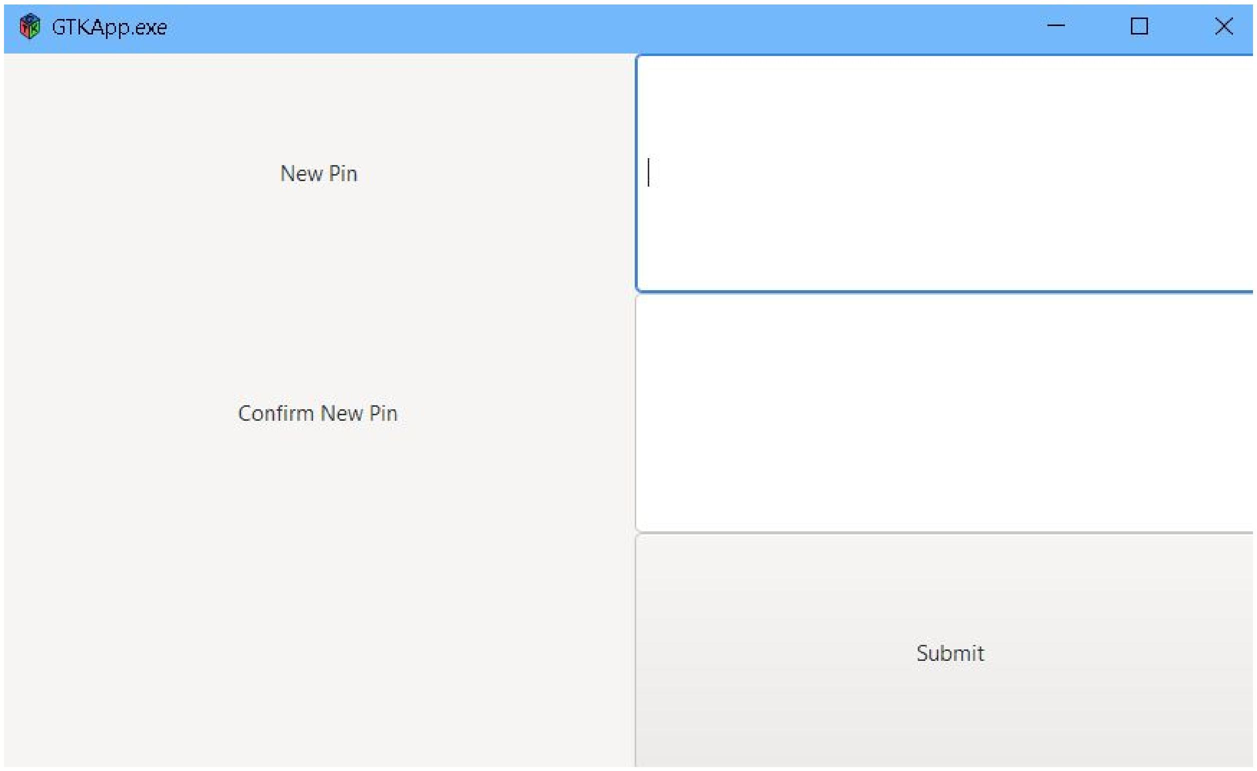
Collect your cash

Balance Enquiry

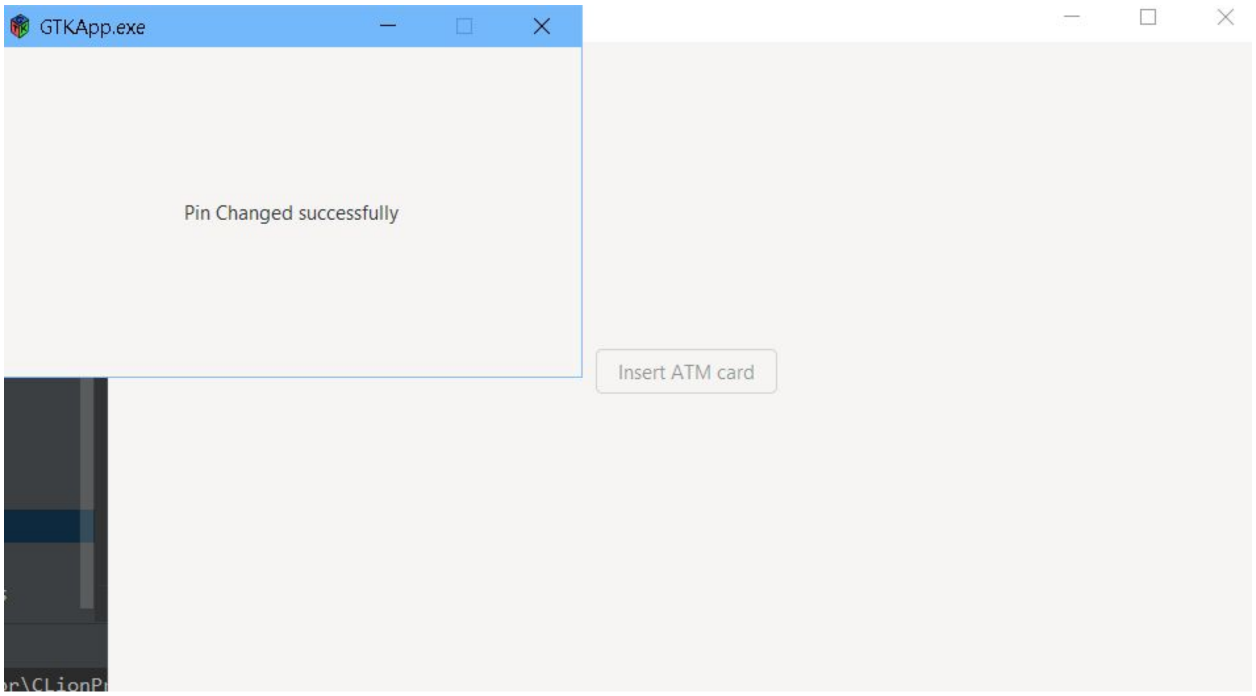


Your Balance is:89000

Change PIN



PIN Changed



Pin Changed successfully