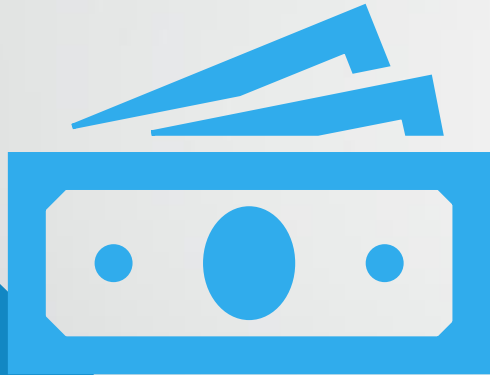


BUDGET MANAGEMENT APP



ΜΑΘΗΜΑ: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΔΙΔΑΣΚΩΝ: ΝΙΚΟΛΑΟΣ ΚΕΚΑΤΟΣ

ΦΟΙΤΗΤΕΣ: ΝΤΕΝΙΣΟΝ ΜΠΑΡΔΗ,
ΑΓΓΕΛΟΣ ΠΕΤΡΑΚΙΔΗΣ

ΕΙΣΑΓΩΓΗ & MOTIVATION

- Τι αφορά το project;
- Ποιο πρόβλημα λύνει;

Το παρόν project είναι μια full stack εφαρμογή υπολογισμού εξόδων (Budget Management App) η οποία επιτρέπει στους χρήστες να καταγράφουν και να παρακολουθούν τα έσοδα και τα έξοδά τους. Η εφαρμογή υποστηρίζει πολλαπλούς χρήστες, σύνδεση με βάση δεδομένων, και παρέχει μια απλή και φιλική προς τον χρήστη διεπαφή για καλύτερο οικονομικό έλεγχο.

- **Γιατί επιλέξαμε αυτό το αντικείμενο;**

Επιλέξαμε αυτό το αντικείμενο γιατί είναι πρακτικό και αφορά την καθημερινή οικονομική διαχείριση.

Objective & Scope

- Ποιοι είναι οι βασικοί στόχοι;
- Ποιες δυνατότητες περιλαμβάνονται;

Η δημιουργία μιας λειτουργικής εφαρμογής για την καταγραφή και παρακαλούθηση εξόδων και εσόδων στην οποία περιλαμβάνονται δυνατότητες αποθήκευσης στοιχείων ενός χρήστη εξυπηρετώντας έτσι την ιδιωτικότητά του. Επίσης παρουσιάζει το συνολικό υπόλοιπο του χρήστη και δίνει τέλος την δυνατότητα εκκαθάρισης δεδομένων.

System Architecture

Πώς λειτουργεί το σύστημα;

- Ο χρήστης κάνει login ή εγγραφή.
- Εισάγει budget/expenses.
- Τα δεδομένα αποθηκεύονται και υπολογίζεται το υπόλοιπο.

Ποιες είναι οι βασικές του συνιστώσες;

- User Authentication

LoginForm: Σύνδεση χρηστών (SQL queries).

RegisterForm: Εγγραφή νέων χρηστών.

- GUI & Business Logic

MainForm:

TextBoxes (εισαγωγή budget/εξόδων).

Buttons (υποβολή, εκκαθάριση).

Labels (εμφάνιση υπολοίπου).

- Data Management

SQL DATABASE: Αποθήκευση δεδομένων χρηστών(πίνακας users)

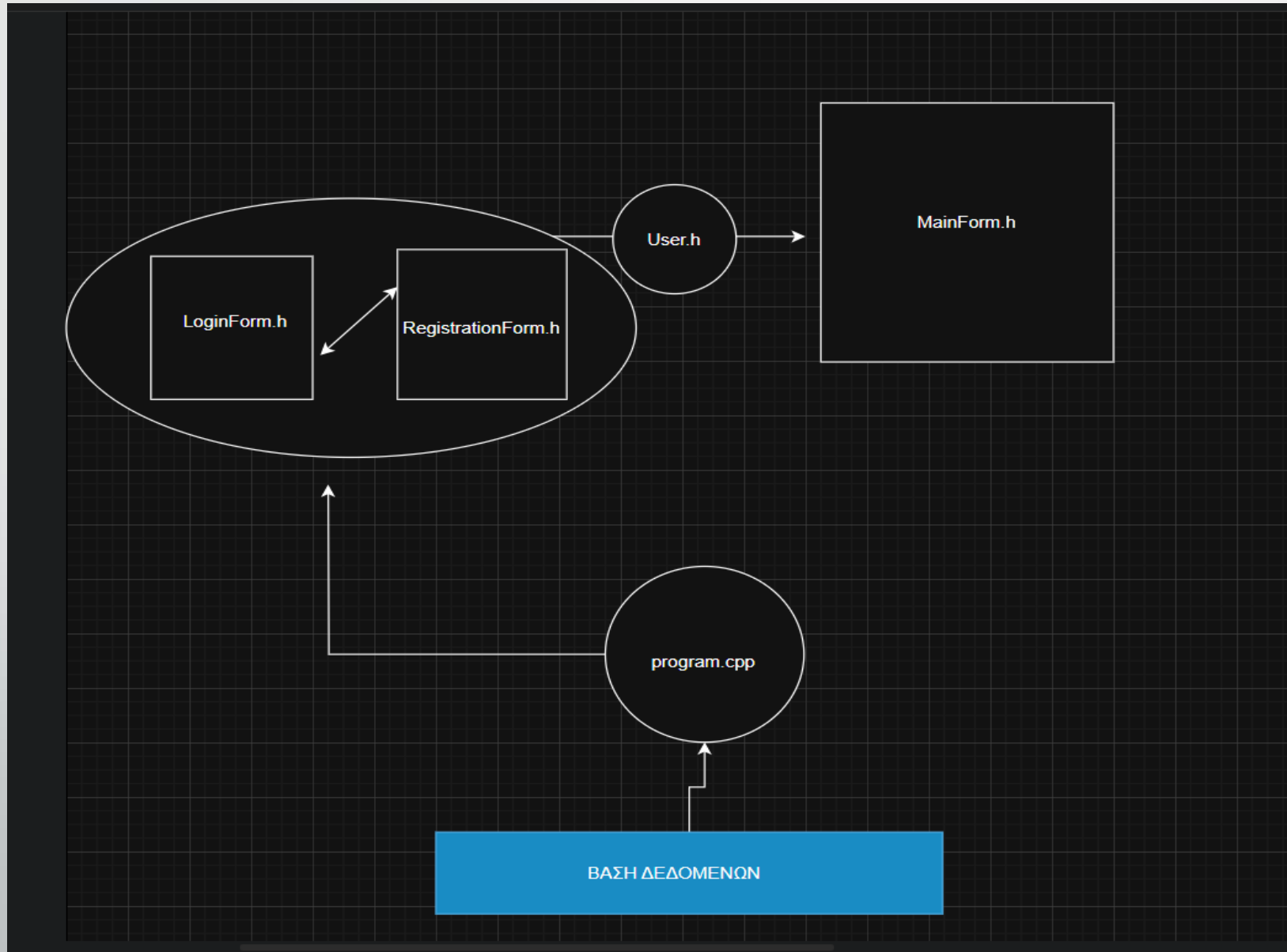
- Validation

Έλεγχοι:

TryParse για αριθμούς.

Έλεγχος υπολοίπου πριν την αφαίρεση.

Διάγραμμα Ροής



Τεχνολογίες που Χρησιμοποιήθηκαν:

Γλώσσα: C++

Περιβάλλον: Visual Studio 2022

GUI: Windows Forms

Βάση δεδομένων: SQL Server Management Studio

Includes : Pragma Once

Κώδικας και Υλοποίηση

- Πως υλοποιήθηκε η βασική λειτουργικότητα;

Η βασική λειτουργικότητα υλοποιήθηκε με γλώσσα c++. Ο κάθε χρήστης μπορεί να συνδεθεί με τους προσωπικούς του κωδικούς ή να δημιουργήσει λογαριασμό να συνδεθεί και να αρχίσει να καταχωρεί τον οικονομικό του προϋπολογισμό. Το πρόγραμμα παρουσιάζει σε μία απλή και ευανάγνωστη διεπαφή τα καταχωρημένα έξοδα το είδος και το ποσό τους όπως και το γενικό υπόλοιπο.

- Σημαντικοί αλγόριθμοι και δομές δεδομένων που χρησιμοποιήθηκαν.

2 Βασικοί αλγόριθμοι (λογικές) της εφαρμογής αφορούν τα κουμπιά Clear και AddExpense ως handlers των ClickEvents.

Με το clear καθαρίζονται όλα τα πεδία εισόδου (textboxes).

Με το AddExpense αποθηκεύεται η εκάστοτε τιμή (value) από το textbox δημιουργείται μία εγγραφή και παρουσιάζεται στο ανάλογο textbox. Αυτή η εγγραφή (αξία της μεταβλητής) αφαιρείται από το συνολικό budget το οποίο επίσης μεταβάλλει το Label του.



**Repository Link: [https://github.cub/trovaotrova/er
gakekatos](https://github.cub/trovaotrova/er
gakekatos)**

Αποτέλεσμα και Demo

LoginForm

Login

Email

Password

[register](#)

RegistrationForm

REGISTER

NAME:

EMAIL:

PHONE:

ADD:

PWD:

[login](#)

Welcome, Jack!

Enter Your Budget:

1000

Expense

laptop

Expense-Amount

100

add budget

add expense

CLEAR

Your Balance

Your Expenses

Welcome, Jack!

Enter Your Budget:

1000

Budget: 900,00 €

Expense

Expense-Amount

add budget

add expense

CLEAR

Your Expenses

laptop: -100,00 €

Welcome, Jack!

Enter Your Budget:

1000

Budget: 855,00 €

Expense

Expense-Amount

add budget

add expense

CLEAR

Your Expenses

laptop: -100,00 €
telephone: -45,00 €

Comparison with AI generated code

```
}  
private: System::Void expensebtn_Click(System::Object^ sender, System::EventArgs^ e) {  
    String^ expenseName = tbexpense->Text;  
    String^ amountText = tbAmount->Text;  
  
    double expenseAmount;  
    if (String::IsNullOrEmpty(expenseName)) {  
        MessageBox::Show("Please enter a valid expense name and amount.");  
        return;  
    }  
  
    if (expenseAmount > currentBalance) {  
        MessageBox::Show("Not enough balance.");  
        return;  
    }  
  
    if (Double::TryParse(amountText, expenseAmount)) {  
        currentBalance -= expenseAmount;  
  
        expenseList += expenseName + ": -" + expenseAmount.ToString("F2") + " €\n";  
        EXPENSES->Text = "Your Expenses\n\n" + expenseList;  
        balance->Text = "Budget: " + currentBalance.ToString("F2") + " €";  
    }  
  
    tbexpense->Clear();  
    tbAmount->Clear();  
}
```

c#

Ανταγραφή

Επεξεργασία

```
private: System::Void expensebtn_Click(System::Object^ sender, System::EventArgs^ e) {  
    String^ expenseName = tbexpense->Text->Trim();  
    String^ amountText = tbAmount->Text->Trim();  
  
    double expenseAmount;  
  
    // Έλεγχος αν το όνομα εξόδου ή το ποσό είναι κενά  
    if (String::IsNullOrEmpty(expenseName) || String::IsNullOrEmpty(amountText)) {  
        MessageBox::Show("Please enter both a valid expense name and amount.");  
        return;  
    }  
  
    // Έλεγχος αν το ποσό είναι έγκυρος αριθμός  
    if (!Double::TryParse(amountText, expenseAmount)) {  
        MessageBox::Show("Please enter a valid number for the amount.");  
        return;  
    }  
  
    // Έλεγχος αν υπάρχει αρκετό υπόλοιπο  
    if (expenseAmount > currentBalance) {  
        MessageBox::Show("Not enough balance to add this expense.");  
        return;  
    }  
  
    // Αφαίρεση του ποσού και ενημέρωση της λίστας εξόδων και του υπολοίπου  
    currentBalance -= expenseAmount;  
    expenseList += expenseName + ": -" + expenseAmount.ToString("F2") + " €\n";  
    EXPENSES->Text = "Your Expenses\n\n" + expenseList;  
    balance->Text = "Budget: " + currentBalance.ToString("F2") + " €";  
  
    // Καθαρισμός πεδίων  
    tbexpense->Clear();  
    tbAmount->Clear();  
}
```

Comparison with AI generated code

- Οι αλλαγές που παρατηρούνται στο κώδικα της τεχνητής νοημοσύνης είναι εμφανείς. Υπάρχει η συνάρτηση Trim() που αφαιρεί κενά στην αρχή και στο τέλος των πεδίων. Πιο ξεκάθαρα μηνύματα σφάλματος. Καλύτερη σειρά ελέγχων. Έλεγχος ποσού πριν την χρήση του.

Conclusions & Lessons learned

- Από αυτό το Project εξοικειωθήκαμε με τη γλώσσα C++, με το περιβάλλον ανάπτυξης κώδικα Visual Studio όπως και με το εργαλείο Microsoft SQL Server.
- Θα μπορούσε να βελτιωθεί η δομή του κώδικα και η διεπαφή της εφαρμογής.
- Μελλοντική προέκταση του Project πιθανόν να είναι η μόνιμη αποθήκευση των εισόδων των χρηστών ακόμα και μετά το τερματισμό του προγράμματος καθώς και η αυτόματη αποθήκευση της ημερομηνίας και ώρας που γίνονται οι καταχωρήσεις εξόδων.