POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK:

Informatyka

SPECJALNOŚĆ:

Inżynieria Systemów Komputerowych

Projekt Zespołowy

TEMAT PROJEKTU:

Budżet Domowy

PROJECT’S SUBJECT:

Household Budget

AUTORZY:

Piotr Danowski

Indeks: 200699

Maksymilian Knoski

Indeks: 200637

Wojciech Kuczyński

Indeks: 200739

Konrad Puchalski

Indeks: 200904

Jakub Zagrobelny

Indeks: 200660

PROWADZĄCY:

dr inż. Paweł Rogaliński

WE WSPÓŁPRACY Z FIRMĄ:

InsERT S.A.

OCENA:

Wrocław 2015

Spis treści

[1. Wstęp 4](#_Toc421875554)

[1.1 Cele projektu 4](#_Toc421875555)

[1.1.1 Realizacja aplikacji 4](#_Toc421875556)

[1.1.2 Organizacja pracy 4](#_Toc421875557)

[1.2 Zakres projektu 4](#_Toc421875558)

[2. Analiza wymagań 5](#_Toc421875559)

[2.1 Założenia projektowe 5](#_Toc421875560)

[2.1.1 Słownik pojęć 5](#_Toc421875561)

[2.1.2 Założenia 5](#_Toc421875562)

[2.2 Opis projektu 5](#_Toc421875563)

[2.3 Wymagania funkcjonalne 5](#_Toc421875564)

[2.3.1 Diagram przypadków użycia 6](#_Toc421875565)

[2.3.2 Scenariusze przypadków użycia 7](#_Toc421875566)

[2.4 Wymagania niefunkcjonalne 8](#_Toc421875567)

[2.4.1 Wykorzystane technologie i narzędzia 8](#_Toc421875568)

[2.4.2 Wymagania efektywnościowe i niezawodnościowe 8](#_Toc421875569)

[2.4.3 Wymagania dotyczące bezpieczeństwa systemu 9](#_Toc421875570)

[3. Projekt aplikacji 10](#_Toc421875571)

[3.1 Projekt bazy danych 10](#_Toc421875572)

[3.1.1 Analiza rzeczywistości i uproszczony model konceptualny 10](#_Toc421875573)

[3.1.2 Model logiczny 10](#_Toc421875574)

[3.1.3 Model fizyczny 10](#_Toc421875575)

[3.2 Projekt aplikacji 10](#_Toc421875576)

[3.2.1 Architektura aplikacji 10](#_Toc421875577)

[3.2.2 Interfejs graficzny i struktura okien 10](#_Toc421875578)

[3.2.3 Projekt funkcji systemu 11](#_Toc421875579)

[3.2.4 Metoda komunikacji z bazą danych 11](#_Toc421875580)

[3.2.5 Projekt zabezpieczeń na poziomie aplikacji 11](#_Toc421875581)

[4. Implementacja systemu 12](#_Toc421875582)

[4.1 Realizacja bazy danych 12](#_Toc421875583)

[4.2 Realizacja elementów aplikacji 12](#_Toc421875584)

[5. Testy systemu 13](#_Toc421875585)

[5.1 Testy zaimplementowanych funkcjonalności 13](#_Toc421875586)

[5.2 Testy mechanizmów bezpieczeństwa 13](#_Toc421875587)

[5.3 Wnioski z testów 13](#_Toc421875588)

[6. Podsumowanie 14](#_Toc421875589)

[7. Spis Rysunków 15](#_Toc421875590)

[8. Spis Listingów 16](#_Toc421875591)

[9. Literatura 17](#_Toc421875592)

# Wstęp

## Cele projektu

W ramach realizacji projektu zostały wyznaczone dwa główne cele. Jednym z nich jest zaprojektowanie i zaimplementowanie aplikacji desktopowej, a drugim nauka organizacji pracy w zespole.

### Implementacja aplikacji

Stworzenie aplikacji desktopowej budżet domowy będzie polegało na zaprojektowaniu poszczególnych jej elementów i ich implementacja. Wykorzystamy przy tym wiedzę zdobytą dotychczas podczas studiów i jeśli będzie to konieczne poznamy nowe technologie i techniki programowania.

### Organizacja pracy

Nauka organizacji pracy będzie polegała zapoznaniu się z zasadami panującymi w firmach branży IT i stosowaniu poznanych metod podczas realizacji projektu. Ważnymi elementami będą: podział pracy miedzy członków zespołu, wyznaczanie sobie kroków milowych i podział ich na mniejsze zadania do wykonania między spotkaniami, ustalenie wspólnych standardów kodu, ujednolicenie wykorzystywanego oprogramowania i określenie dróg komunikacji między sobą.

## Zakres projektu

Projekt aplikacji Budżet Domowy obejmie zamodelowanie i implementację programu. Wymagane będzie realizowanie projektu w grupie by możliwe było wykorzystanie zasad organizacji pracy w zespole informatycznym. Obowiązkiem grupy było cotygodniowe przedstawianie prowadzącemu postępów w pracy.

Kolejnymi etapami projektu będzie zaprojektowanie modelu bazy oraz programu w ramach wstępnego projektu. Następnie w fazie implementacji będzie stopniowo realizowany zarówno projekt bazy, jak i przypadki użycia aplikacji. Gdy kod będzie gotowy wygenerowana zostanie przykładowa baza danych oraz przeprowadzone zostaną testy. Finalnym elementem będzie opracowanie dokumentacji projektu.

# Analiza wymagań

Ten punkt przedstawia analizę wymagań dotyczących aplikacji realizowanej w ramach projektu. W kolejnych podpunktach są: założenia projektowe(w tym słownik pojęć), opis projektu, wymagania funkcjonalne (diagram przypadków użycia i ich opisy), wymagania niefunkcjonalne (używane oprogramowanie, wymogi efektywnościowe, niezawodnościowe i dotyczące bezpieczeństwa).

## Założenia projektowe

W tym podpunkcie zostanie przedstawiony słownik używanych pojęć oraz założenia z nimi związane, które były używane przy implementacji.

### Słownik pojęć

* Użytkownik – osoba korzystająca z programu; jeden budżet może być obsługiwany przez całą rodzinę, a także jedna osoba może prowadzić kilka budżetów.
* Wydatek okresowy – jest to forma wydawania pieniędzy, która następuje co pewien czas określony przez użytkownika np. czynsz mieszkaniowy, opłaty za zużycie wody.
* Wydatek pojedynczy – jest to forma wydawania pieniędzy, która w przeciwieństwie do wydatku okresowego nie ma powtarzalnego charakteru lub jest nieregularna np. zakup komputera, zakup jedzenia.
* Przychód okresowy – jest to forma zdobywania pieniędzy, która następuje co pewien czas określony przez użytkownika np. pensja, renta.
* Przychód pojedynczy – jest to forma zdobywania pieniędzy, która w przeciwieństwie do przychodu okresowego nie ma powtarzalnego charakteru lub jest nieregularna np. pożyczka z banku, premia.
* Budżet domowy – plan oraz zestawienie przychodów i wydatków gospodarstwa domowego w ustalonym przedziale czasowym.
* Cel oszczędzania – jest to forma wydania pieniędzy określona przez użytkownika, która jest zaplanowana do zrealizowania za pewien okres czasu i na którą użytkownik postanawia oszczędzać w wybranym trybie oszczędzania.
* Tryb oszczędzania – może być automatyczny(poprzez wybór opcji „Automatyczne oszczędzanie”) lub manualny. Automatyczne oszczędzanie dodaje co określony czas wyznaczoną kwotę, a użytkownik tylko zatwierdza jej odłożenie. Manualne oszczędzanie polega na tym, że użytkownik sam odkłada dowolną kwotę w dowolnym momencie.
* Aktualne saldo – kwota wyświetlana w oknie głównym w zakładce Strona główna. Reprezentuje obecnie posiadaną kwotę pieniędzy możliwą do rozdysponowania. Od salda są odejmowane także kwoty odkładane przy celach oszczędzania.

### Założenia

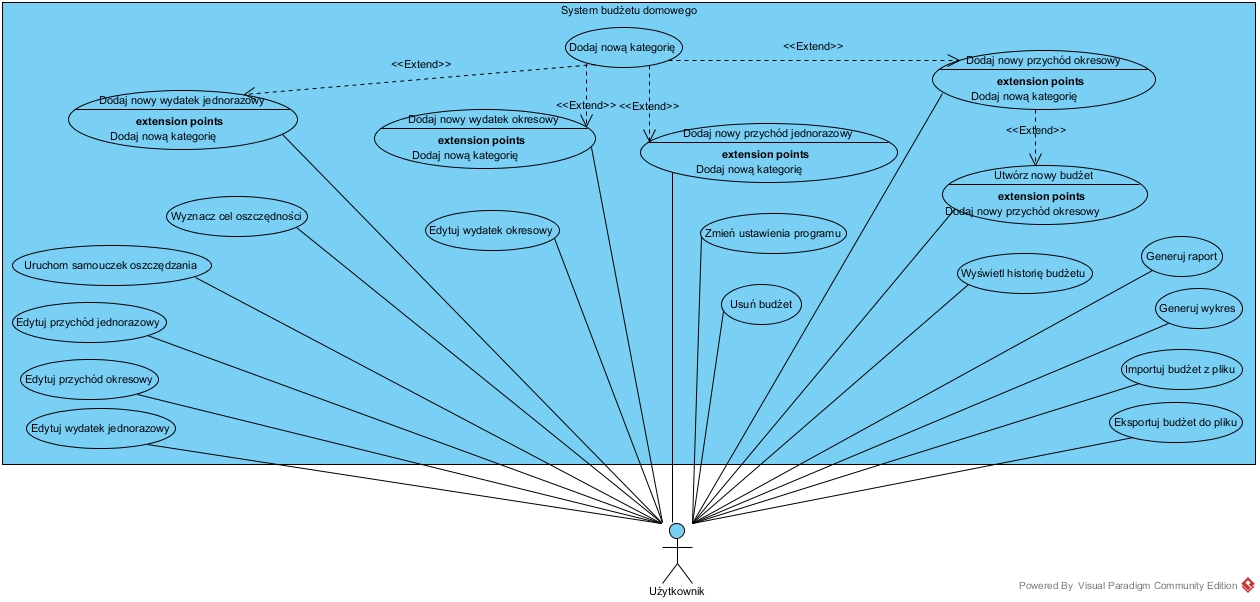
* Wydatki i przychody są zdefiniowane poprzez ich nazwę, kwotę, kategorię, typ(okresowy czy pojedynczy), datę, opcjonalnie notatkę i okres powtarzania w przypadku typu okresowego. Nie ma ograniczenia na liczbę wydatków i przychodów.
* Budżet jest zdefiniowany jako nazwa, hasło, saldo oraz baza wydatków, przychodów i celów oszczędzania. Nie ma ograniczenia na liczbę utworzonych budżetów.
* Cel oszczędzania jest definiowany jako nazwa, kwota docelowa, kwota odkładana, priorytet osiągnięcia celu, tryb oszczędzania(manualny lub automatyczny), data rozpoczęcia oszczędzania, data zakończenia oszczędzania, okres powtarzania odkładania(w przypadku trybu automatycznego) oraz notatka. Nie ma ograniczenia na liczbę wyznaczonych celów oszczędzania.

## Opis projektu

Aplikacja Budżet Domowy, to narzędzie desktopowe usprawniające kontrolę wydatków i przychodów. Użytkownik może wprowadzać swoje wydatki i przychody, przeglądać historię już wprowadzonych oraz personalizować wyświetlanie. Może wyznaczać sobie cele oszczędzania, a także ma wgląd w zbliżające się przychody lub wydatki okresowe (otrzymuje też powiadomienia o zbliżających się). Na podstawie wprowadzanych dane tworzone są wykresy ułatwiające określenie salda z danego okresu, czy obrazujące zależności np. pomiędzy wydatkami, a przychodami lub różnymi kategoriami płatności. Funkcjonalność celu oszczędzania pokazuje także prognozowany jest czas, w którym użytkownik będzie miał możliwość osiągnięcia wyznaczonych cel.

## Wymagania funkcjonalne

### Diagram przypadków użycia



* + - 1. Diagram przypadków użycia

### Scenariusze przypadków użycia

* Utwórz nowy budżet

W ramach przypadku użycia użytkownik klika przycisk „+” w oknie logowania, a następnie w oknie tworzenia nowego budżetu wpisuje w odpowiednie rubryki kolejno: nazwę budżetu, hasło, powtórnie hasło oraz obecnie posiadaną ilość pieniędzy. Następnie użytkownik klika przycisk „Dalej” i wyświetla się zapytanie o dodanie pierwszych wydatków/przychodów. Jeśli użytkownik kliknie „Tak” zostanie zainicjalizowany przypadek użycia Dodawania nowego przychodu. Jeśli natomiast kliknie przycisk „Nie” zostanie zakończone tworzenie nowego budżetu i otwarte okno główne.

* Usuń budżet

W ramach przypadku użycia użytkownik

* Dodaj nowy wydatek jednorazowy

W ramach przypadku użycia użytkownik

* Dodaj nowy przychód jednorazowy

W ramach przypadku użycia użytkownik

* Dodaj nowy wydatek okresowy

W ramach przypadku użycia użytkownik

* Dodaj nowy przychód okresowy

W ramach przypadku użycia użytkownik

* Dodaj nową kategorię

W ramach przypadku użycia użytkownik

* Edytuj wydatek jednorazowy

W ramach przypadku użycia użytkownik

* Edytuj przychód jednorazowy

W ramach przypadku użycia użytkownik

* Edytuj wydatek okresowy

W ramach przypadku użycia użytkownik

* Edytuj przychód okresowy

W ramach przypadku użycia użytkownik

* Wyznacz cel oszczędności

W ramach przypadku użycia użytkownik

* Wyświetl historię budżetu

W ramach przypadku użycia użytkownik

* Zmień ustawienia programu

W ramach przypadku użycia użytkownik

* Autoryzacja

W ramach przypadku użycia użytkownik

## Wymagania niefunkcjonalne

### Wykorzystane technologie i narzędzia

Do realizacji aplikacji desktopowej zostały wykorzystane różne technologie i narzędzia. W fazie projektu wstępnego zostały wykorzystane: Visual Paradigm Community Edition w celu zamodelowania bazy i aplikacji, a także program Microsoft PowerPoint 2013 w celu przedstawienia wstępnych założeń i modelu na konferencji początkowej.

W fazie implementacji została wykorzystana technologia programowania C# oraz technologia bazodanowa SQLite. Narzędzia użyte do programowania do środowisko Microsoft Visual Studio 2013 Ultimate oraz menadżer bazy danych SQLiteManager.

### Wymagania efektywnościowe i niezawodnościowe

Interfejs użytkownika powinien być możliwie prosty i zrozumiały, a także funkcjonalny. Program nie powinien się wieszać, a zapis i odczyt z bazy danych powinien być możliwie szybki tak, by nie był uciążliwy. Program nie powinien mieć błędów, więc nie powinien wyrzucać wyjątków podczas korzystania, a jeśli nawet takie wystąpią, to powinny być w odpowiedni sposób obsłużone.

### Wymagania dotyczące bezpieczeństwa systemu

# Projekt aplikacji

Ten punkt przedstawia ogólny projekt aplikacji desktopowej Budżet Domowy, czyli projekt bazy danych oraz projekt aplikacji w tym interfejsu graficznego użytkownika.

## Projekt bazy danych

### Analiza rzeczywistości i uproszczony model konceptualny

W wykorzystywanej bazie są potrzebne tabele przechowujące wydatki i przychody okresowe, wydatki i przychody jednorazowe, balance logi, dane budżetu, kategorie wydatków i przychodów oraz cele oszczędzania.

### Model logiczny

### Model fizyczny

## Projekt aplikacji

### Architektura aplikacji

Aplikacja została napisana w języku programowania C# w środowisku programistycznym Microsoft Visual Studio Ultimate 2013.

Do zaprojektowania i wygenerowania GUI została wykorzystane API oraz silnik graficzny WPF, która ze względu na szerokie możliwości, a także dostęp do designera ułatwiającego pracę, najbardziej nadawała się do naszych potrzeb.

Do rysowania wykresów zostały użyte dwie bezpłatne biblioteki zapewniające możliwie największą estetykę i możliwości: Modern UI (Metro) Charting oraz WPF Toolkit.

### Interfejs graficzny i struktura okien

Interfejs graficzny składa się z 4 głównych okien, co upraszcza interfejs i zapewnia większą wygodę korzystania z aplikacji. Wykorzystywane okna to:

* Okno logowania – obsługuje funkcje autoryzacji oraz rozpoczyna funkcję kreatora nowego budżetu
* Okno kreatora budżetu – zawiera strony umożliwiające stworzenie nowego budżetu oraz dodanie pierwszych wydatków i przychodów
* Okno główne – zawiera strony obsługujące główne funkcje programu
* Okno wylogowania – obsługuje funkcję wylogowania i funkcję zamknięcia.

Okno główne i okno kreatora są oparte na technologii stron, które można przełączać nie zamykając okna. Wykorzystane są następujące strony:

* Dla okna kreatora budżetu:
  + Strona do wprowadzenia danych nowego budżetu
  + Strona do wprowadzenia pierwszych wydatków
  + Strona do wprowadzenia pierwszych przychodów
* Dla okna głównego
  + Strona Główna
  + Strona Historia i Analiza
  + Strona Przychody i Wydatki
  + Strona Ustawienia

Dać do okien i stron dodatkowe opisy – jakie oferują funkcjonalności

Poniżej umieszczam rysunki z poszczególnymi oknami o stronami.

Tu będą okna i podpisy

### Projekt funkcji systemu

Funkcja autoryzacji jest obsługiwana przez pierwsze wyświetlane okno programu – Okno Logowania. Po poprawnej autoryzacji aplikacja przechodzi do głównego okna programu z domyślnie otwartą Stroną Główną. Logowania rozpoczyna się po wciśnięciu przycisku Zaloguj lub klawisza Enter na klawiaturze komputera i przebiega w następujący sposób:

* Z pól wyboru nazwy budżetu i pola hasła pobierane są dane
* Pobrane dane są porównywane z zapisanymi w bazie danych
* W zależności od wyniku porównania, program przechodzi dalej lub wyświetla komunikat o niezgodności danych.

Kolejne funkcje podobnie w tym funkcje analizy danych

### Metoda komunikacji z bazą danych

Łączenie z bazą danych odbywa się przy wykorzystaniu bibliotek do komunikacji z bazą SQLite. Po zalogowaniu dane z bazy są kopiowane do obiektu budżet i operacje na danych odbywają się na zmiennych. Wszelkie zmiany za zapisywane w przez czasowy autozapis, przez wciśnięcie przycisku Zapisz lub przy kończeniu pracy z programem, jeśli użytkownik zatwierdzi zmiany.

### Projekt zabezpieczeń na poziomie aplikacji

Na poziomie aplikacji zostało wprowadzone zabezpieczenie przed nieautoryzowanym dostępem do Budżetu poprzez wprowadzenie funkcji logowania.

# Implementacja systemu

## Realizacja bazy danych

Baza została zrealizowana i zaimplementowana przy użyciu bibliotek C# do obsługi bazy SQLite. Baza jest tworzona podczas tworzenia nowego budżetu. Odbywa się to poprzez wykonanie kodu z Listingu 1.

Listingi tylko fragmentów kodu, opis klas i ich działania

Listing kodu

## Realizacja elementów aplikacji

Aplikacja została napisana w języku C# i składa się z klas odpowiedzialnych za okna oraz z klas odpowiedzialnych z analizę danych komunikację z bazą. Tu następuję opis poszczególnych klas i listingi

# Testy systemu

## Testy zaimplementowanych funkcjonalności

Tutaj będą rysunki z oknami i wyświetlane komunikaty w przypadku np. błędnego logowania. Umieszczone będą też rysunki z bazy danych, aby potwierdzić, że wprowadzone zmiany poprawnie dodały się do bazy.

## Testy mechanizmów bezpieczeństwa

Tutaj te będą rysunki potwierdzające, poprawność działania funkcji bezpieczeństwa takich jak logowanie.

## Wnioski z testów

Po przetestowaniu funkcji programu można stwierdzić, że wszystkie działają poprawnie i zgodnie z założeniami. Wszelkie zabezpieczenia przed niepoprawną autoryzacją zostały przetestowanie uniemożliwiają dostęp do danych budżetu bez znajomości hasła. Wyniki testów aplikacji są zadowalające.

Dokumentacja – manual dla użytkownika (screeny, chronologia przejść między oknami, analiza, ustawienia)

Instrukcja instalacji

Wersja z kodem, wersja ze skompilowaną wersją (płyta)

Dodatkowy dokument z zawartością katalogów płyty

Płyta ma być podpisana

# Podsumowanie

Projekt zespołowy skutecznie wypróbował zdobytą w czasie studiów wiedzę, a także pozwolił na naukę skutecznej organizacji pracy w zespole informatycznym. Udało się także zaprojektować i zaimplementować praktyczną i w pełni funkcjonalną aplikację wspomagającą prowadzenie budżetu. Jest to projekt, który ma szerokie możliwości rozwoju, jednak czas semestru jest ograniczony pozwolił na implementacji jedynie podstawowych funkcji. Mimo ograniczeń czasowych efekt pracy jest zadowalający. Można uznać, że wszystkie cele projektowe zostały osiągnięte.

Co zostało zrobione, co można jeszcze zrobić, jak układała się współpraca, jak podobała się nam realizacja projektu

Na konferencji mówimy do ludzi, którzy nie wiedzą o co chodzi, więc musimy tak to przekazać, żeby zrozumieli, bo nie mają żadnych informacji o tym do czego służy ten projekt

# Spis Rysunków

# Spis Listingów

# Literatura

[1] <http://modernuicharts.codeplex.com/documentation#howto1>

[2] http://www.c-sharpcorner.com/uploadfile/mahesh/charting-in-wpf/