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 3](#_Toc421999011)

[1.1 Cele projektu 3](#_Toc421999012)

[1.1.1 Implementacja aplikacji 3](#_Toc421999013)

[1.1.2 Organizacja pracy 3](#_Toc421999014)

[1.2 Zakres projektu 3](#_Toc421999015)

[2. Analiza wymagań 4](#_Toc421999016)

[2.1 Założenia projektowe 4](#_Toc421999017)

[2.1.1 Słownik pojęć 4](#_Toc421999018)

[2.1.2 Założenia 4](#_Toc421999019)

[2.2 Opis projektu 5](#_Toc421999020)

[2.3 Wymagania niefunkcjonalne 5](#_Toc421999021)

[2.3.1 Wykorzystane technologie i narzędzia 5](#_Toc421999022)

[2.3.2 Wymagania efektywnościowe i niezawodnościowe 6](#_Toc421999023)

[2.3.3 Wymagania dotyczące bezpieczeństwa systemu 6](#_Toc421999024)

[2.4 Wymagania funkcjonalne 6](#_Toc421999025)

[2.4.1 Diagram i opisy przypadków użycia 6](#_Toc421999026)

[3. Projekt aplikacji 16](#_Toc421999027)

[3.1 Projekt bazy danych 16](#_Toc421999028)

[3.2 Projekt aplikacji 16](#_Toc421999029)

[3.2.1 Architektura aplikacji 16](#_Toc421999030)

[3.2.2 Interfejs graficzny i struktura okien 17](#_Toc421999031)

[3.2.3 Projekt funkcji systemu 24](#_Toc421999032)

[3.2.4 Metoda komunikacji z bazą danych 24](#_Toc421999033)

[3.2.5 Projekt zabezpieczeń na poziomie aplikacji 24](#_Toc421999034)

[4. Implementacja systemu 25](#_Toc421999035)

[4.1 Realizacja bazy danych 25](#_Toc421999036)

[4.2 Realizacja elementów aplikacji 25](#_Toc421999037)

[5. Testy systemu 26](#_Toc421999038)

[5.1 Testy zaimplementowanych funkcjonalności 26](#_Toc421999039)

[5.2 Testy mechanizmów bezpieczeństwa 26](#_Toc421999040)

[5.3 Wnioski z testów 26](#_Toc421999041)

[6. Podsumowanie 27](#_Toc421999042)

[7. Spis Rysunków 28](#_Toc421999043)

[8. Spis Listingów 29](#_Toc421999044)

[9. Literatura 30](#_Toc421999045)

# Wstęp

## Cele projektu

W ramach realizacji projektu zostały wyznaczone dwa główne cele. Jednym z nich jest 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. Wymogiem będzie też 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. Aby sprawdzać na bieżąco funkcjonalności programu wygenerowana zostanie przykładowa baza danych oraz przeprowadzone zostaną testy. Finalnym elementem będzie opracowanie dokumentacji projektu.

# Analiza wymagań

## 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. Nie dodaje się wydatków i przychodów z datą wcześniejszą niż data założenia budżetu, bo może to generować błędy.
* 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.
* Aktualne saldo rozpoczyna się od kwoty podanej przy tworzeniu nowego budżetu.

## Opis projektu

Program Budżet Domowy, to aplikacja desktopowa usprawniająca kontrolę wydatków i przychodów. Użytkownik może wprowadzać swoje wydatki i przychody (jednorazowe oraz okresowe), przeglądać historię już wprowadzonych włącznie z wartością salda po każdej operacji oraz personalizować wyświetlanie. Może wyznaczać sobie cele oszczędzania i ma wgląd w zbliżające się przychody lub wydatki okresowe (otrzymuje także powiadomienie o terminie płatności). Na podstawie wprowadzonych danych przedstawiane są wykresy ułatwiające określenie salda z danego miesiąca, włącznie z predykcją na dwa kolejne miesiące, wykresy wydatków i przychodów z poszczególnych kategorii za dany okres, czy wykres sumy przychodów i wydatków z kolejnych miesięcy. Użytkownik wyznaczając sobie cel oszczędzania widzi także prognozowany jest czas, w którym użytkownik będzie miał możliwość osiągnięcia wyznaczonych celów oszczędzania. Aplikacja oferuje także możliwość edycji oraz usuwania wprowadzonych danych.

## Wymagania niefunkcjonalne

### Wykorzystane technologie i narzędzia

Do realizacji aplikacji desktopowej zostaną wykorzystane różne technologie i narzędzia. W fazie projektu wstępnego będą 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 zostanie wykorzystana technologia programowania C# oraz technologia bazodanowa SQLite. Środowisko użyte do programowania to środowisko Microsoft Visual Studio 2013 oraz menadżer bazy danych SQLiteManager. Do rysowania wykresów zostały wykorzystane 2 darmowe biblioteki zapewniające możliwie największe możliwości edycji: Modern UI (Metro) Charting oraz WPF Toolkit.

W celu ułatwienia pracy zostanie wykorzystane repozytorium z kontrolą wersji GitHub. Natomiast do komunikacji oraz wyznaczania zadań posłuży portal społecznościowy Facebook oraz portal zarządzania zadaniami - Trello.

### Wymagania efektywnościowe i niezawodnościowe

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

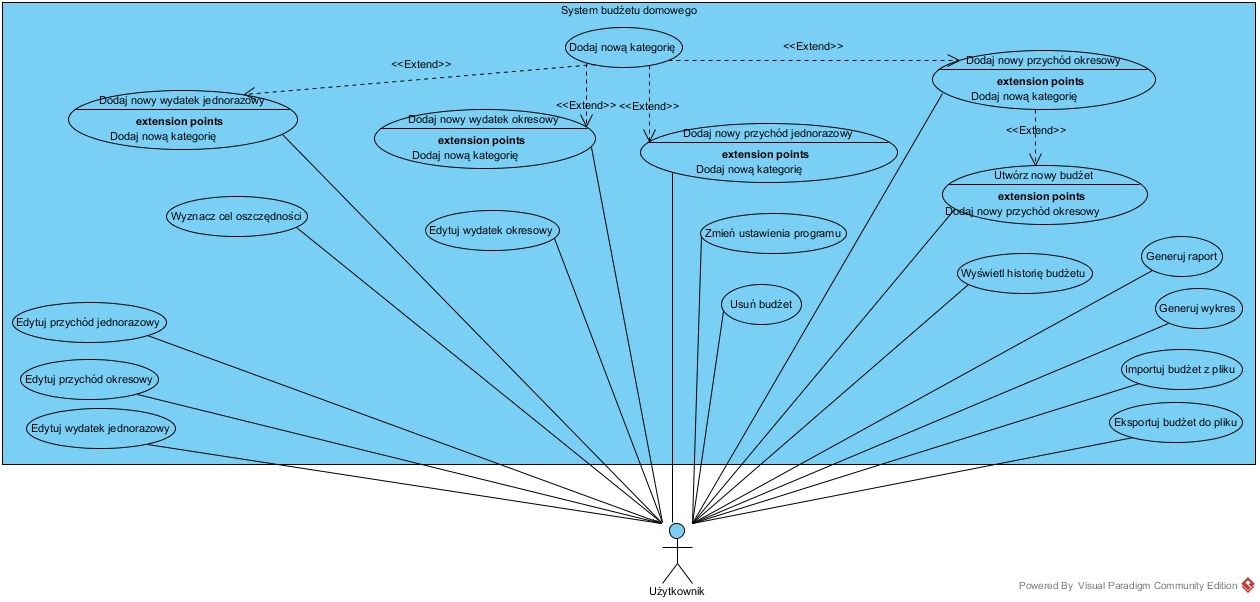
### Wymagania dotyczące bezpieczeństwa systemu

W celu zabezpieczenia programu przed nieuprawnionym dostępem do danych program będzie posiadał funkcję autoryzacji do każdego z budżetów. Poza tym baza będzie szyfrowana, żeby nie można było sprawdzić jej zawartości poprzez bezpośredni wgląd do pliku SQLite.

## Wymagania funkcjonalne

W tym punkcie zostaną przedstawione wymagania odnośnie funkcji realizowanych przez zaprojektowaną aplikację.

### Diagram i opisy przypadków użycia



1. Diagram przypadków użycia

|  |  |
| --- | --- |
| **Nazwa przypadku użycia** | **Utwórz nowy budżet.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć utworzenia nowego budżetu. |
| Warunki wstępne | Budżet nie został jeszcze stworzony. |
| Warunek pomyślnego zakończenia | Nowy budżet zostaje utworzony w systemie. |
| Warunek niepomyślnego zakończenia | Nowy budżet nie zostaje uwzględniony w systemie. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie sekwencji przycisków (kreator). |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o dodanie nowego budżetu lub, jeśli jest to pierwsze uruchomienie programu, akcja jest wykonywana samoczynnie. 2. Użytkownik wpisuje nazwę budżetu. 3. Baza danych zostaje utworzona 4. Opcjonalnie: Użytkownik ma możliwość dodania przychodów okresowych. 5. Użytkownik zatwierdza operację. 6. Użytkownik podaje hasło dostępu. 7. Budżet dodawany jest do systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Podczas tworzenie budżetu użytkownik ma możliwość wprowadzenia przychodów okresowych (**Extend::Dodaj nowy przychód okresowy)**  5.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Usuń istniejący budżet.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć usunięcia istniejącego budżetu. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. |
| Warunek pomyślnego zakończenia | Budżet zostaje usunięty z systemu. |
| Warunek niepomyślnego zakończenia | Budżet nie zostaje usunięty z systemu. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o usunięcie istniejącego budżetu. 2. Podanie hasła dostępu. 3. Użytkownik zatwierdza operację. 4. Budżet jest usuwany z systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  3.1 Użytkownik anuluje operację. |

|  |  |
| --- | --- |
| **Nazwa przypadku użycia** | **Dodaj nowy przychód okresowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć dodania nowego przychodu okresowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Przychód okresowy jest uwzględniony w systemie. |
| Warunek niepomyślnego zakończenia | Przychód okresowy nie jest uwzględniony w systemie. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o dodanie nowego przychodu okresowego. 2. Użytkownik wprowadza kwotę przychodu. 3. Użytkownik wprowadza nazwę identyfikującą przychód. 4. Użytkownik wybiera kategorie przychodu. 5. Użytkownik wybiera dzień miesiąca, w którym planowane jest pojawienie się przychodu. 6. Użytkownik zatwierdza operację. 7. Przychód okresowy dodawany jest do systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Użytkownik chce wprowadzić własną kategorię (**Extend::Utwórz nową kategorię**)  6.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Dodaj nowy wydatek okresowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć dodania nowego wydatku okresowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Wydatek okresowy jest uwzględniony w systemie. |
| Warunek niepomyślnego zakończenia | Wydatek okresowy nie jest uwzględniony w systemie. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o dodanie nowego wydatku okresowego. 2. Użytkownik wprowadza kwotę wydatku. 3. Użytkownik wprowadza nazwę identyfikującą wydatek. 4. Użytkownik wybiera kategorie wydatku. 5. Użytkownik wybiera dzień miesiąca, w którym planowane jest pojawienie się wydatku. 6. Użytkownik zatwierdza operację. 7. Wydatek okresowy dodawany jest do systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Użytkownik chce wprowadzić własną kategorię (**Extend::Utwórz nową kategorię**)  6.1 Użytkownik anuluje operację. |

|  |  |
| --- | --- |
|  |  |
| **Nazwa przypadku użycia** | **Dodaj wydatek jednorazowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć dodania wydatku jednorazowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Wydatek jednorazowy jest uwzględniony w systemie. |
| Warunek niepomyślnego zakończenia | Wydatek jednorazowy nie jest uwzględniony w systemie. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o dodanie nowego wydatku jednorazowego. 2. Użytkownik wprowadza kwotę wydatku. 3. Użytkownik wprowadza nazwę identyfikującą wydatek. 4. Użytkownik wybiera kategorie wydatku. 5. Użytkownik wybiera datę wydatku. 6. Użytkownik zatwierdza operację. 7. Wydatek jednorazowy dodawany jest do systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Użytkownik chce wprowadzić własną kategorię (**Extend::Utwórz nową kategorię**)  6.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Dodaj przychód jednorazowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć dodania przychodu jednorazowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Przychód jednorazowy jest uwzględniony w systemie. |
| Warunek niepomyślnego zakończenia | Przychód jednorazowy nie jest uwzględniony w systemie. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o dodanie nowego przychodu jednorazowego. 2. Użytkownik wprowadza kwotę przychodu. 3. Użytkownik wprowadza nazwę identyfikującą przychód. 4. Użytkownik wybiera kategorie przychodu. 5. Użytkownik wybiera datę przychodu. 6. Użytkownik zatwierdza operację. 7. Przychód jednorazowy dodawany jest do systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Użytkownik chce wprowadzić własną kategorię (**Extend::Utwórz nową kategorię**)  6.1 Użytkownik anuluje operację. |
|  |  |
| **Nazwa przypadku użycia** | **Edytuj przychód okresowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć edycji przychodu okresowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Przychód okresowy jest wyedytowany. |
| Warunek niepomyślnego zakończenia | Przychód okresowy pozostaje bez zmian. |
| Aktor | Użytkownik |
| Wyzwalacz | Wybranie przez użytkownika przychodu okresowego do edycji. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o edycję wybranego przychodu okresowego. 2. Użytkownik przeprowadza zmiany (kwota, dzień miesiąca, kategoria) 3. Użytkownik zatwierdza zmiany. 4. Zmiany uwzględniane są w systemie. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  2.1 Użytkownik wybiera opcję usunięcia przychodu okresowego.  2.2 Przychód jest usuwany z systemu.  3.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Edytuj wydatek okresowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć edycji wydatku okresowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Wydatek okresowy jest wyedytowany. |
| Warunek niepomyślnego zakończenia | Wydatek okresowy pozostaje bez zmian. |
| Aktor | Użytkownik |
| Wyzwalacz | Wybranie przez użytkownika wydatku okresowego do edycji. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o edycję wybranego wydatku okresowego. 2. Użytkownik przeprowadza zmiany (kwota, dzień miesiąca, kategoria) 3. Użytkownik zatwierdza zmiany. 4. Zmiany uwzględniane są w systemie. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  2.1 Użytkownik wybiera opcję usunięcia wydatku okresowego.  2.2 Wydatek jest usuwany z systemu.  3.1 Użytkownik anuluje operację. |

|  |  |
| --- | --- |
| **Nazwa przypadku użycia** | **Edytuj przychód jednorazowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć edycji przychodu jednorazowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Przychód jednorazowy jest wyedytowany. |
| Warunek niepomyślnego zakończenia | Przychód jednorazowy pozostaje bez zmian. |
| Aktor | Użytkownik |
| Wyzwalacz | Wybranie przez użytkownika przychodu jednorazowego do edycji. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o edycję wybranego przychodu jednorazowego. 2. Użytkownik przeprowadza zmiany (kwota, data, kategoria) 3. Użytkownik zatwierdza zmiany. 4. Zmiany uwzględniane są w systemie. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  2.1 Użytkownik wybiera opcję usunięcia przychodu jednorazowego  2.2 Przychód jest usuwany z systemu.  3.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Edytuj wydatek jednorazowy.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć edycji wydatku jednorazowego. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Wydatek jednorazowy jest wyedytowany. |
| Warunek niepomyślnego zakończenia | Wydatek jednorazowy pozostaje bez zmian. |
| Aktor | Użytkownik |
| Wyzwalacz | Wybranie przez użytkownika wydatku jednorazowego do edycji. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o edycję wybranego wydatku jednorazowego. 2. Użytkownik przeprowadza zmiany (kwota, data, kategoria) 3. Użytkownik zatwierdza zmiany. 4. Zmiany uwzględniane są w systemie. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  2.1 Użytkownik wybiera opcję usunięcia wydatku jednorazowego  2.2 Wydatek jest usuwany z systemu.  3.1 Użytkownik anuluje operację. |

|  |  |
| --- | --- |
| **Nazwa przypadku użycia** | **Utwórz nową kategorię.** |
| Kontekst zdaniowy | Użytkownik wyraża chęć dodania nowej kategorii przychodów/wydatków. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Nowa kategoria jest dodana do systemu. |
| Warunek niepomyślnego zakończenia | Nowa kategoria nie jest dodana do systemu. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku nowej kategorii. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o dodanie nowej kategorii. 2. Użytkownik wprowadza nazwę kategorii. 3. Użytkownik wybiera czy kategoria dotyczy przychodów lub wydatków lub obu. 4. Użytkownik zatwierdza operację. 5. Nowa kategoria dodawana jest do systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Wyświetl historię budżetu** |
| Kontekst zdaniowy | Użytkownik wyraża chęć wyświetlenia historii budżetu |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Historia budżetu zostaje wyświetlona. |
| Warunek niepomyślnego zakończenia | Historia budżetu nie zostaje wyświetlona, bądź jest wyświetlona błędnie. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku historii. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o wyświetlenie historii danego budżetu. 2. Użytkownik wybiera okres, jaki ma obejmować historia budżetu. 3. Użytkownik wybiera typy wydatków, jakie mają zostać uwzględnione. 4. Użytkownik zatwierdza operację. 5. Historia budżetu zostaje wyświetlona użytkownikowi. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Użytkownik anuluje operację. |

|  |  |
| --- | --- |
| **Nazwa przypadku użycia** | **Zmień ustawienia programu** |
| Kontekst zdaniowy | Użytkownik wyraża chęć zmiany ustawień programu. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Wybrane ustawienia programu zostają zmienione. |
| Warunek niepomyślnego zakończenia | Wybrane ustawienia programu nie zostają zmienione. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o zmianę ustawień. 2. Użytkownik wybiera opcje, które chce zmienić. 3. Opcjonalnie: Użytkownik zmienia wygląd programu. 4. Użytkownik zatwierdza zmiany. 5. Ustawienia zostają zapisane w systemie. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  4.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Eksportuj budżet do pliku** |
| Kontekst zdaniowy | Użytkownik wyraża chęć wyeksportowania budżetu do pliku. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Plik z budżetem zostaje zapisany na dysku. |
| Warunek niepomyślnego zakończenia | Plik z budżetem nie zostaje zapisany na dysku. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o zapisanie piku z obecnym budżetem 2. Użytkownik wybiera miejsce, w którym chce zapisać plik. 3. Użytkownik zatwierdza wybór. 4. Plik z budżetem zostaje zapisany na dysku. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  3.1 Użytkownik anuluje operację. |

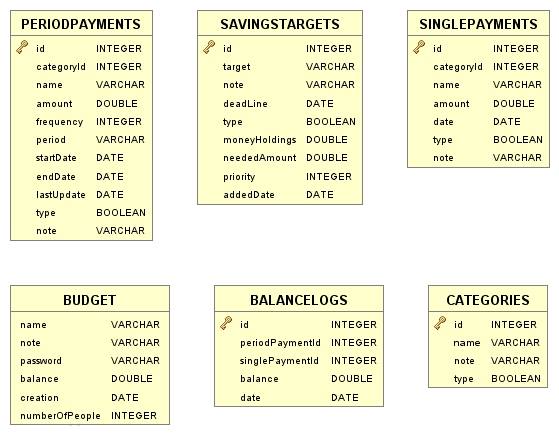
|  |  |
| --- | --- |
| **Nazwa przypadku użycia** | **Generuj raport** |
| Kontekst zdaniowy | Użytkownik wyraża chęć wygenerowania raportu. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Raport zostaje wygenerowany. |
| Warunek niepomyślnego zakończenia | Raport nie zostaje wygenerowany, bądź zostaje wygenerowany błędnie. |
| Aktor | Użytkownik. |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o wygenerowanie raportu. 2. Użytkownik wybiera zakres czasowy, jaki ma obejmować raport. 3. Użytkownik wybiera miejsce zapisu na dysku. 4. Użytkownik zatwierdza wybór. 5. Raport zostaje wygenerowany. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  3.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Generuj wykres** |
| Kontekst zdaniowy | Użytkownik wyraża chęć wygenerowania wykresu. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Wykres zostaje wygenerowany. |
| Warunek niepomyślnego zakończenia | Wykres nie zostaje wygenerowany, bądź zostaje wygenerowany błędnie. |
| Aktor | Użytkownik. |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o wygenerowanie wykresu. 2. Użytkownik wybiera zakres czasowy, typ wykresu oraz zakres danych (kategorie przychodów/wydatków). 3. Użytkownik zatwierdza wybór. 4. Wykres zostaje wygenerowany. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  3.1 Użytkownik anuluje operację. |

|  |  |
| --- | --- |
| **Nazwa przypadku użycia** | **Importuj budżet z pliku** |
| Kontekst zdaniowy | Użytkownik wyraża chęć zaimportowania budżetu z pliku. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. |
| Warunek pomyślnego zakończenia | Plik z budżetem zostaje wczytany z dysku. |
| Warunek niepomyślnego zakończenia | Plik z budżetem nie zostaje wczytany z dysku. |
| Aktor | Użytkownik |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o wczytanie pliku z istniejącym budżetem. 2. Użytkownik wybiera miejsce, z którego chce wczytać plik z budżetem. 3. Użytkownik zatwierdza wybór. 4. Plik z budżetem zostaje wczytany do systemu. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  3.1 Użytkownik anuluje operację. |
| **Nazwa przypadku użycia** | **Wyznacz cel oszczędzania** |
| Kontekst zdaniowy | Użytkownik wyraża chęć wyznaczenia celu oszczędzania. |
| Warunki wstępne | Użytkownik musi dysponować odpowiednim potwierdzeniem tożsamości. Budżet musi być stworzony. |
| Warunek pomyślnego zakończenia | Cel oszczędzania zostaje wyznaczony. |
| Warunek niepomyślnego zakończenia | Cel oszczędzania nie zostaje wyznaczony. |
| Aktor | Użytkownik. |
| Wyzwalacz | Naciśnięcie przycisku. |
| Główny przebieg | **Krok Akcja**   1. Użytkownik prosi system o wyznaczenie celu oszczędzania. 2. Użytkownik wpisuje termin, kwotę docelową oraz nazwę celu i opcjonalny opis. 3. Użytkownik zatwierdza wybór. 4. Cel oszczędzania zostaje wyznaczony. |
| Rozszerzenie | **Krok Rozgałęziona akcja**  3.1 Użytkownik anuluje operację. |

# Projekt aplikacji

## Projekt bazy danych

W wykorzystywanej bazie są potrzebne tabele przechowujące dane zawierające się w budżecie wydatki i przychody okresowe, wydatki i przychody jednorazowe, balance logi, dane budżetu, kategorie wydatków i przychodów oraz cele oszczędzania. Ze względu na niską złożoność bazy danych programu została ona zaimplementowana w technologii bazodanowej SQLite według poniższego diagramu.



1. Diagram bazy danych

## Projekt aplikacji

Ten podpunkt zawiera opis poszczególnych elementów aplikacji desktopowej budżet domowy. Kolejno zostaną przedstawione opisy wykorzystanej architektury, interfejsu graficznego, funkcji i zabezpieczeń.

### Architektura aplikacji

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

Do zaprojektowania i wygenerowania GUI zostało wykorzystane API z silnikiem graficznym - WPF, które ze względu na szerokie możliwości personalizacji, a także dostęp do designera ułatwiającego pracę, najlepiej nadawała się do naszych potrzeb.

Do rysowania wykresów zostały wykorzystane 2 darmowe biblioteki zapewniające możliwie największe możliwości edycji: Modern UI (Metro) Charting oraz WPF Toolkit.

### Interfejs graficzny i struktura okien

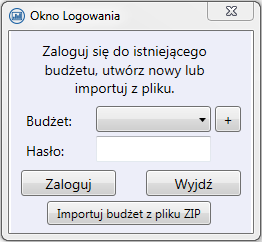
Interfejs graficzny składa się z sześciu głównych okien, które zapewniają większą wygodę korzystania z aplikacji. Wykorzystywane okna to:

* **Okno logowania** – obsługuje funkcje autoryzacji, rozpoczyna funkcję kreatora nowego budżetu oraz umożliwia import budżetu z archiwum zip;
* **Okno kreatora nowego budżetu** – zawiera strony umożliwiające stworzenie nowego budżetu oraz dodanie pierwszych przychodów i wydatków;
* **Okno główne** – zawiera strony obsługujące główne funkcje programu;
* **Okno nowego celu oszczędzania** – obsługuje funkcję dodawania nowego budżetu;
* **Okno zarządzania oszczędzaniem** – obsługuje funkcje dodawania lub odejmowania pieniędzy odkładanych na wybrany cel po jego wybraniu z listy rozwijanej;
* **Okno wylogowania** – obsługuje funkcję wylogowania i funkcję zamknięcia.

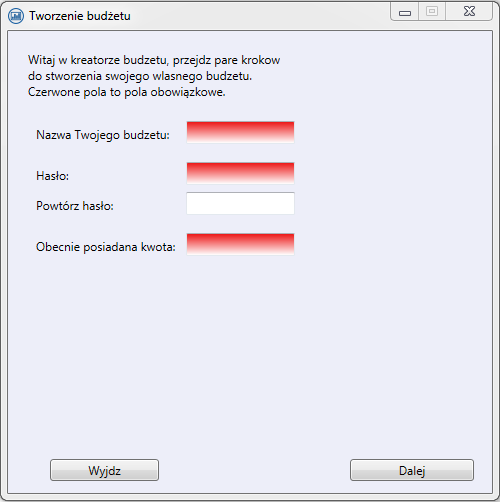
Okno główne i okno kreatora są oparte na technologii stron, które można przełączać nie zamykając okna. Wykorzystane są następujące strony:

* Dla okna kreatora budżetu:
  + **Strona do wprowadzenia danych nowego budżetu** – na tej stronie wprowadza się podstawowe dane budżetu jak nazwa, hasło i kwota początkowa;
  + **Strona do dodawania pierwszych wydatków** – na tej stronie można dodać pierwsze wydatki pojedyncze lub okresowe;
  + **Strona do dodawania pierwszych przychodów** – na tej stronie można wprowadzić pierwsze przychody pojedyncze lub okresowe.
* Dla okna głównego
  + **Strona Główna** – na tej stronie użytkownik ma wgląd w najbliższe wydatki i przychody okresowe, skróconą historię, cele oszczędzania i uproszczony bilans wydatków i przychodów w postaci słupków. Za pomocą odpowiednich przycisków użytkownik może wygenerować historię w formie pliku PDF (zaznaczając wybór okresu może sprecyzować z jakiego przedziału czasu ma zostać wygenerowana historia), otworzyć okno nowego celu oszczędzania lub otworzyć okno zarządzania oszczędzaniem;
  + **Strona Historia** – na tej stronie użytkownik ma wgląd w całą historię swoich przychodów i wydatków. Poprzez opcje personalizacji wyświetlania może zobaczyć przychody i wydatki określonego typu, określonej kategorii, z określonego przedziału kwotowego oraz z wyznaczonego przedziału czasu;
  + **Strona Analiza** – na tej stronie użytkownik ma możliwość zobaczenia analizy swoich przychodów i wydatków w czterech różnych formach dostępnych w osobnych zakładkach:
    - **Zakładka 1: Wykres kołowy** – na dwóch wykresach kołowych (jeden dla przychodów, a drugi dla wydatków) jest przedstawiona proporcja udziału poszczególnych kategorii w wybranym przez użytkownika przedziale czasu;
    - **Zakładka 2: Średnie wydatki i przychody** – użytkownik ma możliwość wyboru 3 miesięcy, dla których zostaną wyświetlone po dwa wykresy słupkowe, gdzie pierwszy oznacza sumę wydatków z miesiąca, a drugi sumę przychodów;
    - **Zakładka 3: Wykres salda** –
    - **Zakładka 4: Wykres kategorii płatności** -
  + **Strona Przychody i Wydatki**
  + **Strona Ustawienia**

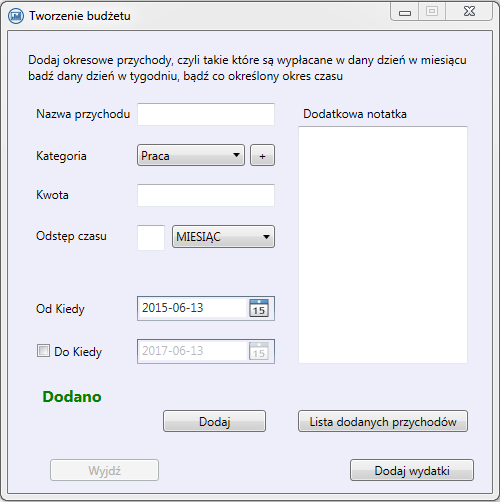
Poniżej umieszczam rysunki z poszczególnymi oknami o stronami.



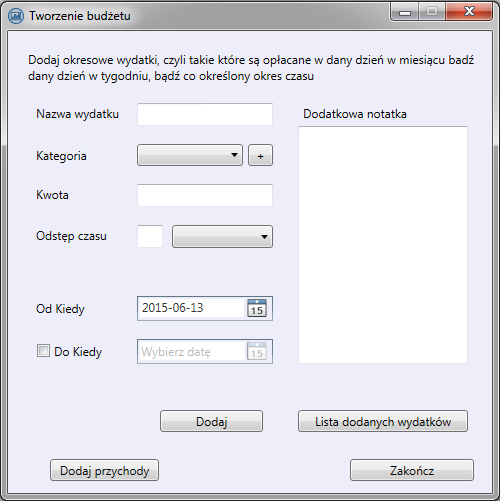
1. Okno Logowania



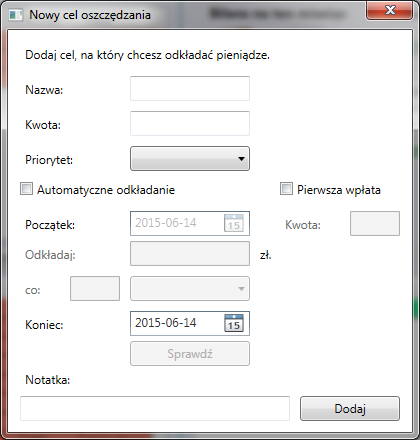
1. Okno kreatora budżetu – strona 1



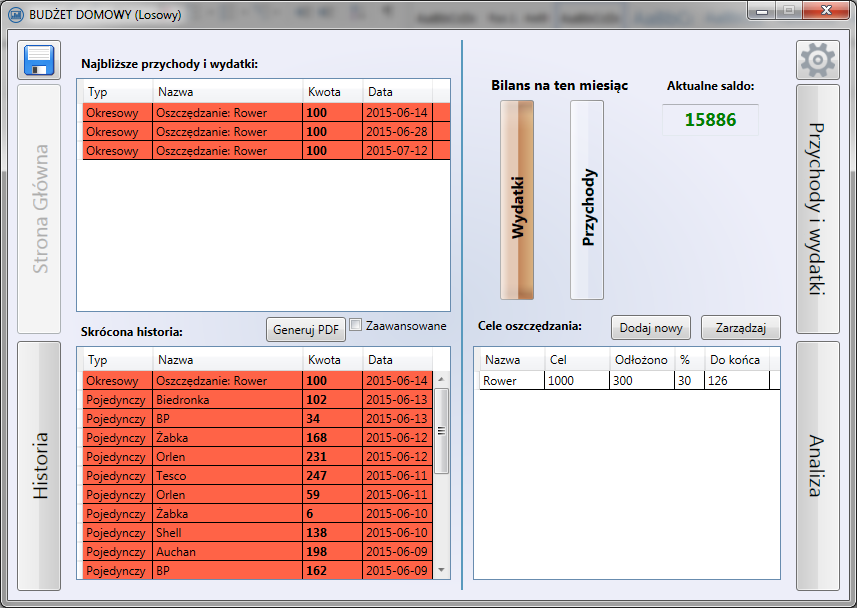
1. Okno kreatora budżetu – strona 2



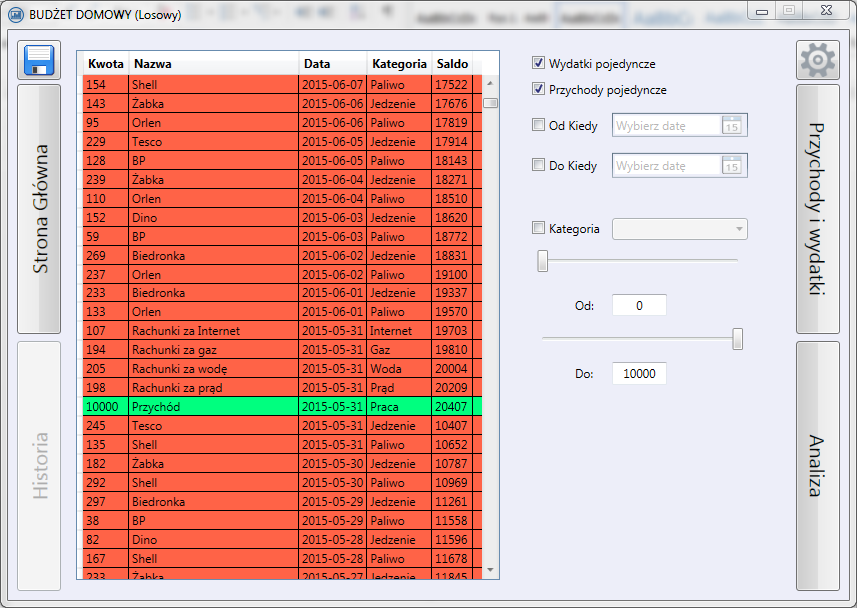
1. Okno kreatora budżetu – strona 3



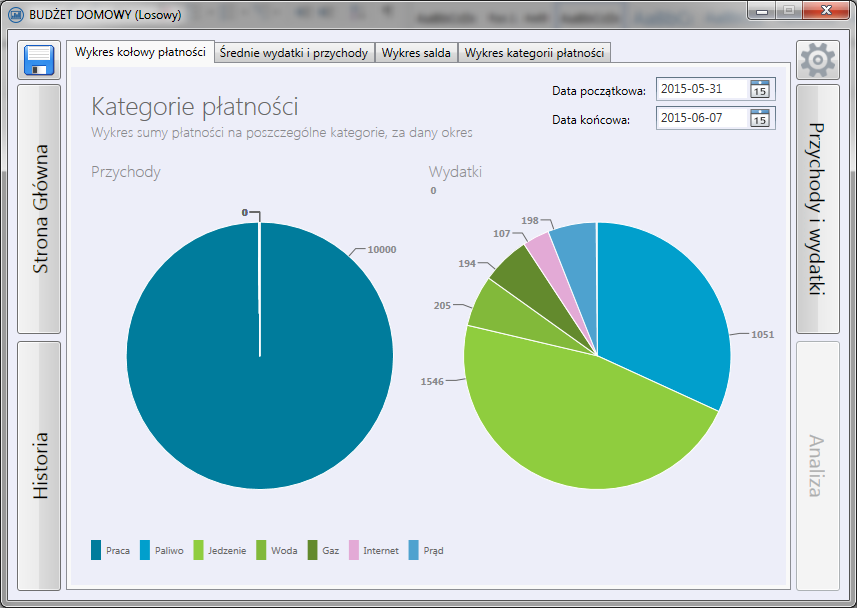
1. Okno nowego celu oszczędzania



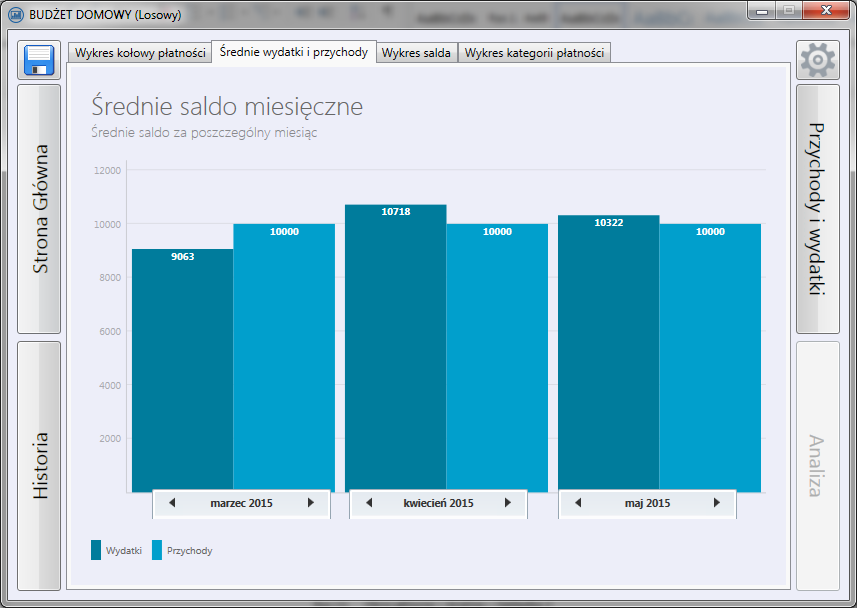
1. Okno główne – Strona Główna



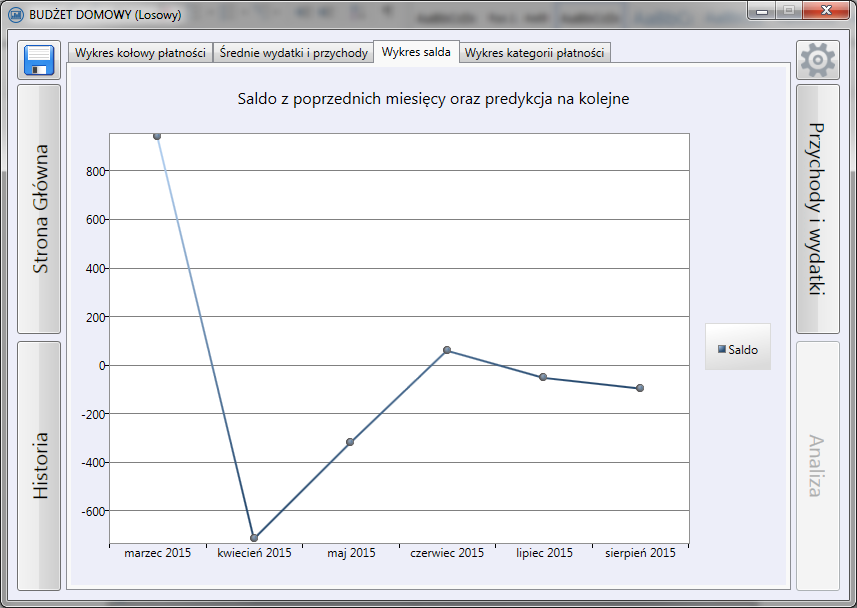
1. Okno główne – Historia



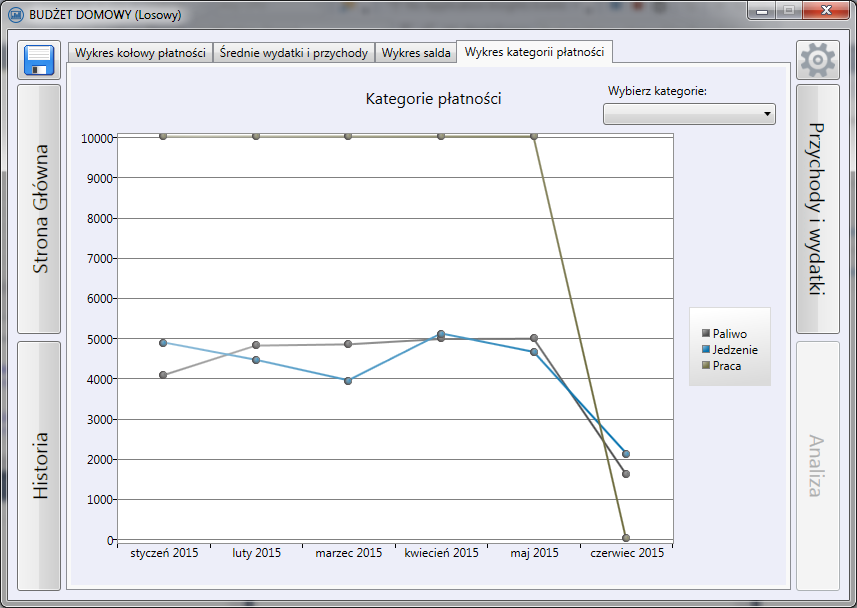
1. Okno główne – Analiza – Zakładka 1



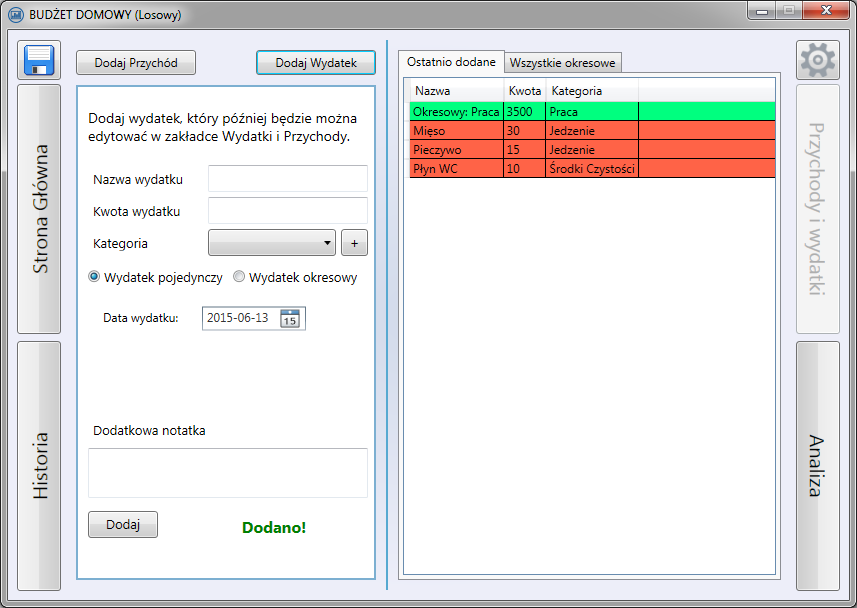
1. Okno główne – Analiza – Zakładka 2



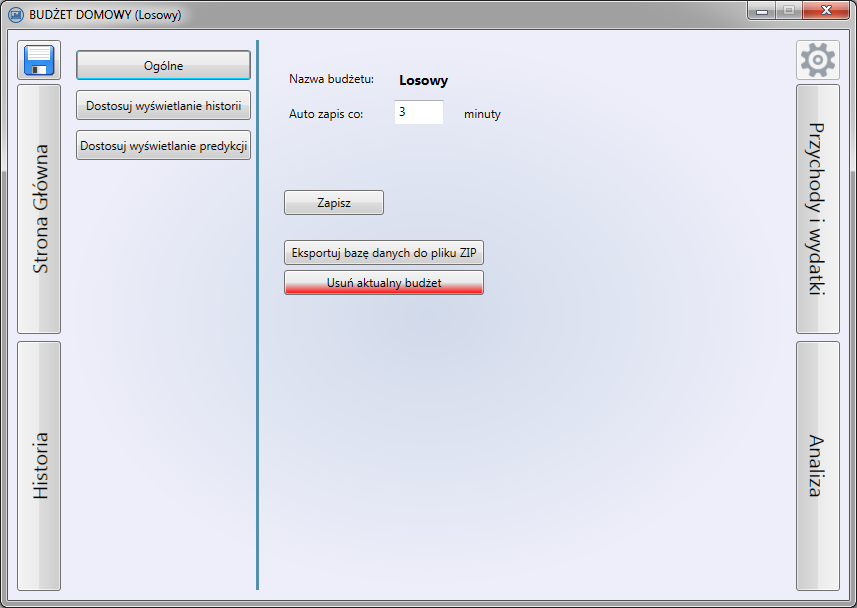
1. Okno główne – Analiza – Zakładka 3



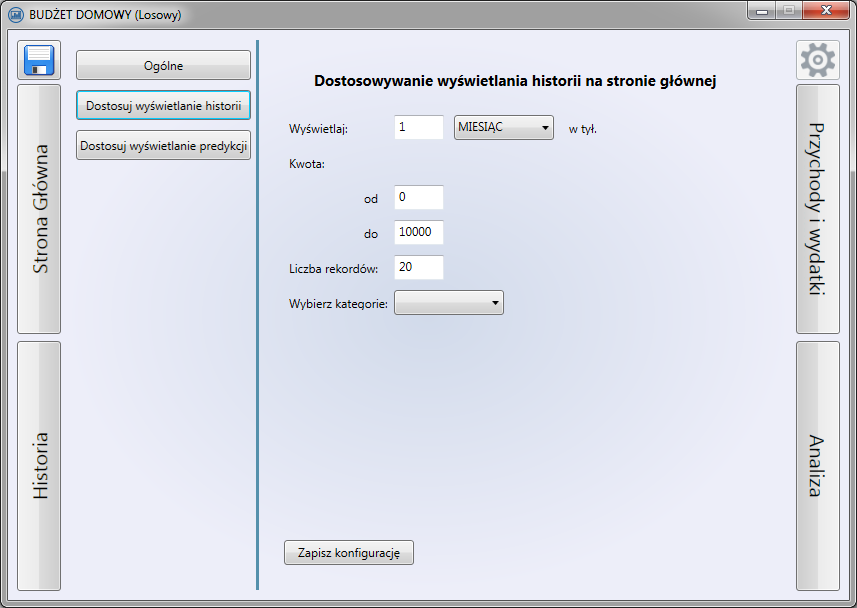
1. Okno główne – Analiza – Zakładka 4



1. Okno główne – Przychody i wydatki



1. Okno główne – Ustawienia - Ogólne



1. Okno główne – Ustawienia – Dostosuj wyświetlanie historii

### 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

# 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 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 skutecznie wypróbował zdobytą w czasie studiów wiedzę, a także pozwolił na naukę skutecznej organizacji pracy w zespole informatycznym. Udało się także zaprojektować i zaimplementować aplikację wspomagającą prowadzenie budżetu domowego. Jest to projekt, który ma szerokie możliwości rozwoju, jednak czas semestru jest ograniczony pozwolił na implementacji jedynie części funkcji. Biorąc pod uwagę ograniczenia czasowe 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/>

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