



# POLSKO-JAPOŃSKA WYŻSZA SZKOŁA TECHNIK KOMPUTEROWYCH

## **Katedra inżynierii oprogramowania**

Inżynieria Oprogramowania i Baz Danych

**Andrzej Pajdziński**

Nr albumu 4008

**Krzysztof Paprota**

Nr albumu 4091

## **Model elektronicznej dystrybucji oprogramowania**

Praca magisterska

napisana pod kierunkiem

dr inż. Mariusz Trzaska

Warszawa, wrzesień, 2010

## **Streszczenie**

Niniejsza praca skupia się na problemie elektronicznej dystrybucji oprogramowania w sieci internet. Autorzy za bezpośredni cel pracy postanowiliśmy sobie zebranie i opracowanie cech i zasad dystrybucji treści w sieci internet, którymi powinien odznaczać się system dystrybucji elektronicznej oraz dobór odpowiednich do zrealizowania tego zadania technologii i bibliotek programistycznych pozwalających na ich implementację.

Został przedstawiony problem oraz dodatkowo przedstawiono istniejące już rozwiązania, razem z ich zaletami bądź wadami. Zaproponowano również podstawową funkcjonalność proponowanego rozwiązania.

Pod koniec pracy opisane zostały problemy, które napotkali autorzy podczas prac implementacyjnych oraz możliwe dalsze kierunki rozwoju.

# Spis treści

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 1. Wstęp.....                                                   | 5  |
| 1.1. Cel pracy.....                                             | 5  |
| 1.2. Rozwiązania przyjęte w pracy.....                          | 6  |
| 1.3. Rezultaty pracy.....                                       | 6  |
| 1.4. Organizacja pracy.....                                     | 7  |
| 2. Elektroniczna dystrybucja oprogramowania.....                | 8  |
| 2.1. Czym jest elektroniczna dystrybucja oprogramowania.....    | 8  |
| 2.2. Istniejące rozwiązania.....                                | 10 |
| 2.2.1. Strony z programami.....                                 | 10 |
| 2.2.1.1. „dobreprogramy.pl”.....                                | 10 |
| 2.2.2. Platformy dystrybucyjne.....                             | 13 |
| 2.2.2.1. STEAM.....                                             | 13 |
| 2.2.2.2. App Store.....                                         | 16 |
| 2.3. Aspekty prawne.....                                        | 18 |
| 2.4. Licencje.....                                              | 21 |
| 3. Proponowane rozwiązanie.....                                 | 24 |
| 3.1. Zabezpieczenie oprogramowania:.....                        | 24 |
| 3.1.1. Stosowane podejścia.....                                 | 25 |
| 3.1.1.1. Rygorystyczne.....                                     | 25 |
| 3.1.1.2. Nierygorystyczne.....                                  | 27 |
| 3.2. Aktualizowanie i powiadamianie o nowszych wersjach:.....   | 29 |
| 3.3. Finansowanie zakupów oprogramowania.....                   | 31 |
| 3.3.1. Rozliczanie.....                                         | 32 |
| 3.4. Zagadnienia powiązana z prawem.....                        | 33 |
| 3.4.1. Sprawa licencji.....                                     | 33 |
| 3.4.2. Sprawdzanie oprogramowania i nasza odpowiedzialność..... | 34 |
| 3.4.3. Nasza odpowiedzialność prawna.....                       | 35 |
| 3.4.4. Regulamin.....                                           | 36 |
| 3.4.4.1. Definicje.....                                         | 36 |
| 3.4.4.2. Preambuła.....                                         | 37 |
| 3.4.4.3. Treść.....                                             | 37 |

|                                                                     |    |
|---------------------------------------------------------------------|----|
| 4. Opis narzędzi zastosowanych w pracy.....                         | 39 |
| 4.1. Środowisko programistyczne – Microsoft Visual Studio 2010..... | 39 |
| 4.2. VISUAL STUDIO TEAM FOUNDATION SERVER 2010.....                 | 41 |
| 4.3. Microsoft Sql Server 2008.....                                 | 42 |
| 4.4. .NET Framework 4.....                                          | 44 |
| 4.5. ASP .NET MVC2.....                                             | 47 |
| 4.6. Windows Presentation Foundation.....                           | 49 |
| 4.7. Windows Communication Foundation (WCF).....                    | 50 |
| 4.8. NHibernate.....                                                | 52 |
| 5. Prototyp aplikacji AppShare.....                                 | 54 |
| 5.1. Opis architektury systemu.....                                 | 54 |
| 5.2. Aplikacja internetowa.....                                     | 58 |
| 5.3. Aplikacja kliencka.....                                        | 67 |
| 6. Zalety, wady oraz plany rozwoju.....                             | 81 |
| 6.1. Zalety oraz wady przyjętych rozwiązań.....                     | 81 |
| 6.2. Plany rozwojowe.....                                           | 82 |
| 7. Podsumowanie.....                                                | 84 |
| 8. Bibliografia.....                                                | 85 |

# 1. Wstęp

Elektroniczna dystrybucja oprogramowania staje się coraz bardziej popularną formą dostarczania i rozpowszechniania oprogramowania. W dzisiejszych czasach tradycyjne sposoby dystrybucji treści cyfrowych („pudełkowe”) coraz bardziej ustępują na rzecz nowych metod. Użytkownicy oczekują głównie ograniczenia kosztu zakupu oraz łatwego dostępu. Nikt nie lubi jazdy po mieście i stania w kolejkach w celu zdobycia potrzebnego programu. Natomiast twórcy oprogramowania decydujące się na nowe sposoby dystrybucji oczekują głównie maksymalizacji zysku przy jak najmniejszych kosztach.

Aktualnie istnieje szereg rozwiązań bazujących na tych oczekiwaniach, jednak nie są one pozbawione wad, niektóre z nich zostaną wymienione oraz zostanie zaproponowane autorskie podejście do tego problemu.

## 1.1. Cel pracy

Za bezpośredni cel pracy postanowiliśmy sobie zebranie i opracowanie cech i zasad dystrybucji treści w sieci internet, którymi powinien odznaczać się system dystrybucji elektronicznej oraz dobór odpowiednich do zrealizowania tego zadania technologii i bibliotek programistycznych pozwalających na ich implementację. Celem pośrednim pracy będzie stworzenie prototypu obrazującego proponowaną koncepcję.

W szczególności prototyp będzie pozwalał twórcom oprogramowania na publikację swoich programów a użytkownikom nam dostęp do nich. Prototyp powinien charakteryzować się następującymi cechami:

- Twórca oprogramowania powinien publikować swoje programy bez konieczności specjalnego ingerowania w kod
- Łatwy w obsłudze interfejs
- Powinna być napisana w sposób umożliwiający rozbudowę i rozszerzenie podstawowej funkcjonalności
- Nie powinien wymagać posiadania przez docelowych użytkowników dodatkowego oprogramowania
- Rozwiązać problem biznesowy – dystrybucji oprogramowania

- Powinien przewidywać zwiększenie liczby użytkowników

## 1.2. Rozwiązania przyjęte w pracy

Opracowany prototyp w ramach pracy opiera się o język C# oraz platformę Microsoft .NET (4.0). Na potrzeby zapewnienia dostępu do danych użyty zostanie SQL Server 2008 express oraz biblioteka NHibernate. Do stworzenia interfejsu użytkownika programu klienckiego zostanie wykorzystana technologia Windows Presentation Foundation (WPF) oraz asp.net MVC2 do wytworzenia aplikacji internetowej - portalu. Praca będzie wykonywana z użyciem Team Foundation Server 2010 oraz Visual Studio 2010. Wybór taki został dokonany z powodu dość dużej oraz stale rosnącej popularności wśród programistów ([Rysunek 16: Wykres popularności c#](#)).

Naszym zdaniem język wraz z platformą umożliwiają dość szybką i sprawną implementację prototypu. Dodatkowo za przewagę platformy .NET należy uznać dużą liczbę dodatkowych bibliotek oraz aktywnej społeczności zapewniającej wsparcie.

## 1.3. Rezultaty pracy

Bezpośrednim wynikiem pracy będzie opracowanie najważniejszych założeń i reguł dystrybucji oprogramowania za pomocą Internetu, w tym także określenie odpowiednich technologii i bibliotek programistycznych umożliwiających i ułatwiających pracę nad prototypem.

Pośrednim wynikiem pracy będzie prototyp obrazujący przedstawioną koncepcję systemu elektronicznej dystrybucji. Powstanie aplikacja internetowa oraz program kliencki (tzw. cienki klient), z których korzystać będą przyszli użytkownicy systemu.

## 1.4. Organizacja pracy

Praca została napisana w następującej kolejności:

- Początek rozpoczyna się od wyjaśnienia pojęcia elektronicznej dystrybucji oprogramowania oraz zestawienia go z tradycyjną metodą dystrybucji. Opisuje główne oczekiwania zarówno ze strony użytkowników jak i twórców oprogramowania.
- W kolejnym rozdziale, dalszej części przedstawiono aktualnie stosowane rozwiązania wraz z ich zaletami i wadami. Ponadto przedstawiono zagadnienia licencji i innych aspektów prawnych dotyczących zagadnienia dystrybucji i świadczenia usług drogą internetową.
- Rozdział trzeci prezentuje i omawia naszą koncepcję proponowanego rozwiązania.
- W rozdziale czwartym wymieniono i opisano narzędzia zastosowane w pracy nad prototypem.
- Rozdział piąty przedstawia prototyp będący pośrednim rezultatem pracy.
- Kolejny rozdział przedstawia wszystkie wady oraz zalety omawianego rozwiązania oraz plany na przyszłość.
- Na zakończenie przedstawione jest krótkie podsumowanie pracy oraz bibliografia użyta do opracowania pracy

## 2. Elektroniczna dystrybucja oprogramowania

Poniższy rozdział ma za zadanie wyjaśnić czym jest elektroniczna dystrybucja oprogramowania oraz czym różni się od wcześniej stosowanych metod dystrybucji.

W dalszej części przedstawiamy obecnie używane systemy które na własny sposób rozwiązują ten problem. Na rynku istnieje kilka powszechnie stosowanych rozwiązań jednak istniejące dość spore różnice sprawiają, że jedne są bardziej przyjazne dla użytkownika od innych.

Dodatkowo w rozdziale [2.3 Aspekty prawne](#) przedstawiamy zagadnienia prawne dotyczące dystrybucji i świadczenia usług w internecie w świetle polskich przepisów. Warto znać swoje prawa i obowiązki przy świadczeniu usług w internecie zwłaszcza przy zagadnieniu dystrybucji treści chronionych licencjami i prawami autorskimi.

### 2.1. Czym jest elektroniczna dystrybucja oprogramowania

Definicja – Według Wikipedii, elektroniczna dystrybucja oprogramowania jest rozwiązaniem mającym na celu umożliwić użytkownikom nabywanie i pobieranie oprogramowania przez internet (np. drogą elektroniczną). Takie podejście stanowi alternatywę dla dostarczania oprogramowania na nośnikach fizycznych taki jak CD czy DVD.

Zazwyczaj dystrybucje w formie elektronicznej mają tą przewagę, że kosztują mniej niż oprogramowanie dystrybuowane na fizycznych nośnikach. Składa się na to brak kosztu składowania i produkowania nośników oraz znacząco zmniejszony koszt dostarczenia. Dlatego powszechną praktyką jest, że twórcy oprogramowania oferują zniżki na oprogramowanie zakupione i pobrane z internetu. Ponadto elektroniczna dystrybucja oprogramowania stanowi szansę dla indywidualnych osób, dla których jest czasami jedyna możliwość na rozpowszechnienie swojego produktu. Jedną z głównych zalet i chyba najważniejszych, takiego modelu dystrybucji jest to, że oferuje użytkownikom stały dostęp do oprogramowania 24/7, bez względu na czas i miejsce. Wystarczy tylko dostęp do internetu. Jest to czynnik, który znacznie wpływa na zadowolenie użytkowników. Oczywiście użytkownicy muszą mieć pewność, że będą mieć takie same prawa i korzyści jak przy nabyciu oprogramowania w klasyczny sposób.



## **Zalety**

Elektroniczna dystrybucja oprogramowania jest szybsza, prostsza i mniej kosztowna niż „klasyczny” model dystrybucji oprogramowania. Biorąc pod uwagę, że użytkownicy z reguły chcą mieć jak najszybszy dostęp do oprogramowania, niezależnie od miejsca w którym się znajdują, elektroniczny model dystrybucji jest idealnym rozwiązaniem spełniającym te wymagania. Prawdopodobnie minie wiele lat nim całkowicie zastąpi klasyczny „pudełkowy” model dystrybucji, jednak do tego czasu będzie stanowić jego uzupełnienie. Już dziś niektóre pudełkowe produkty po zakupie zapewniają możliwość korzystania z ich elektronicznych wersji.

## **Wady**

„Tradycyjne” wersje pudełkowe dość często można później odsprzedać. Tymczasem elektronicznej dystrybucji zazwyczaj przyświeca zasada „nieodstępności nabytych praw”, inaczej mówiąc program zazwyczaj przywiązany jest do konta co uniemożliwia pożyczanie bądź odsprzedaż. Z drugiej strony odsprzedaż jest zazwyczaj niezgodna z licencją użytkownika.

Inną sprawą są ceny, które niejednokrotnie nie odbiegają znacząco od pudełkowych wersji. Poza tym sklepy rywalizują ze sobą, oferują promocje i wyprzedaże. W przypadku elektronicznych wersji mamy do czynienia z monopolem. Większość dystrybucji jest dostępna z pomocą tylko jednego systemu i w jednej globalnej cenie, nie ma mowy o konkurencji a promocje i wyprzedaże nie są jeszcze popularne.

Zostaje jeszcze sfera sentymentalna do rzeczy namacalnych jak pudełko, płyta i papierowa instrukcja. Wersje namacalne zawsze dają większy komfort psychiczny. Podobnie jak z książkami mamy do czynienia z zapachem oraz dotykiem. Poza tym do wersji pudełkowych mogą być dołączone różnego rodzaju gadżety lub urządzenia.

## **2.2. Istniejące rozwiązania**

Celem niniejszego podrozdziału jest przedstawienie obecnie popularnych i używanych rozwiązań zajmujących się dystrybucją programów w sieci. Skupimy się na trzech różnych rozwiązaniach. Pierwszym z nich będzie rodzima strona z programami „dobreprogramy.pl”. Serwis ten jest idealnym przedstawicielem wszelkich dostępnych w sieci stron skupionych na programach oraz umożliwiający ich pobieranie.

### **2.2.1. Strony z programami**

Strony z programami są jednym z bardziej popularnych sposobów dystrybucji oprogramowania, użytkownicy poszukujący konkretnego programu zazwyczaj w pierwszej kolejności sięgają do tego typu stron lub wyszukiwarek.

#### **2.2.1.1. „dobreprogramy.pl”**

Jeden z największych portali internetowy w całości poświęcony legalnemu oprogramowaniu. Jego nadrzędnym celem jest propagowanie legalnego oprogramowania zarówno darmowego jak i komercyjnego.

Portal posiada bardzo pokaźną bazę oprogramowania od kompresorów plików po przeglądarki internetowe. Wspierane platformy to Windows, Mac i Linux. Programy zgrupowane są w następujących kategoriach:

- Akcesoria
- Bezpieczeństwo
- Biuro
- Dom
- Edukacja i Nauka
- Firma
- Grafika
- Internet i Komunikacja
- Multimedia

- Narzędzia
- Obsługa urządzeń mobilnych
- Programowanie
- Dodatki

Na samym początku był to tylko serwis udostępniający programy. Obecnie posiada on znacznie więcej działów m.in.:

- Aktualności
- Artykuły i recenzje
- Blogi użytkowników portalu
- Forum (znacznie szersza tematyka, od sprzętu po zagadnienia związane z programowaniem)
- Materiały wideo (głównie wywiady z twórcami popularnych aplikacji)
- Gry (darmowe wersje gier)

### **Zalety**

- Szeroki zakres oferowanego oprogramowania
- Oprogramowanie na trzy najpopularniejsze platformy (Windows, Linux, Mac os)
- Programy na licencjach płatnych jak i darmowych
- Umożliwia znalezienie potrzebnego programu nawet jeżeli nie wiemy czego dokładnie potrzebujemy
- Możliwość stworzenia własnej listy ulubionych programów

### **Wady**

- Programy dodawane tylko i wyłącznie przez moderatorów serwisu
- Brak możliwości dodania własnych programów
- Bardzo ciężko zgłosić swoje własne sugestie co do oferty programów. Jedyne sposoby to droga mailowa [redakcja@dobreprogramy.pl](mailto:redakcja@dobreprogramy.pl)

## Wnioski

„dobreprogramy.pl” to bardzo dobra strona dla użytkowników programów jednak nie spełnia ona oczekiwań twórców oprogramowania. Brakuje jej przede wszystkim możliwości umieszczania swoich własnych programów.

## Przykładowe zrzuty ekranu

The screenshot shows the 'dobreprogramy.pl' website interface. The top navigation bar includes links for 'opracowanie dla firm', 'Analizy finansowe', 'Raporty', and 'Bazy danych'. The main content area is titled 'Kategoria: Kalkulatory' and lists various calculator programs. A sidebar on the left provides filters for 'Systemy operacyjne' and 'Licencje'. The main listing table includes columns for 'Nazwa programu', 'Data aktualizacji', 'Liczba pobrań', 'Ocena użytkowników', and 'Licencja'.

| Nazwa programu                                                                                                                                                                                                                                                                                        | Data aktualizacji | Liczba pobrań | Ocena użytkowników | Licencja                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---------------|--------------------|--------------------------------------------------|
| AllerCalc 2.11<br>AllerCalc to (wydobyty się z programu VersaCalc) zaawansowany kalkulator w którym zaimplementowano ponad 100 różnych funkcji algebraicznych, trygonometrycznych, inżynierskich, statystycznych i finansowych. Umożliwia operowanie nie tylko na...                                  | 26.09.2005        | 39 234        | 7,5                | bezpłatna                                        |
| Areo Kalkulator 1.0.0.2006<br>Program Areo Kalkulator to zaawansowany, całkowicie darmowy kalkulator, który potrafi rysować wykresy wszelkich funkcji oraz obliczać skomplikowane działania matematyczne. Wystarczy wpisać działania takie jak np. $(2+3.5)^{68/2} \cdot 2$ , a program...            | 23.06.2007        | 32 323        | 6,8                | bezpłatna                                        |
| Calculator Prompter 2.55<br>Bezpłatny kalkulator umożliwiający nie tylko przeprowadzanie prostych obliczeń takich jak dodawanie, odejmowanie, mnożenie, dzielenie itd., ale też obliczanie całych wyrażeń np. $\sin(\cos(\tan(p))) + e^{-3}$ czy $\text{floor}((19+34/3)^{-3} \cdot 2) \cdot 15$ z... | 27.10.2009        | 14 252        | 7,9                | bezpłatna                                        |
| Chemistry Assistant 2.4<br>Chemistry Assistant to bezpłatny kalkulator stworzony przede wszystkim z myślą o obliczeniach chemicznych. Głównym zadaniem programu jest wykonywanie obliczeń masy molekularnej składników chemicznych podanych w postaci klasycznego wzoru...                            | 05.02.2006        | 7 727         | 7,0                | bezpłatna                                        |
| <b>Kalkulator Hipoteczny</b><br>Zaoszczędź nawet 20% na kredycie! Złóż wniosek przez internet<br>PlaceMniej.pl/KredytyHipoteczne                                                                                                                                                                      |                   |               |                    |                                                  |
| EqualMath 1.2.5.0<br>EqualMath to program wykonujący różnorodne operacje matematyczne. Wyposażony jest w kalkulator umożliwiający wykonywanie działań, obliczanie wartości funkcji trygonometrycznych oraz logarytmów o dowolnej podstawie. Na uwagę zasługuje...                                     | 07.07.2008        | 9 891         | 7,7                | bezpłatna                                        |
| Euklides 2.1<br>Euklides to aplikacja matematyczna umożliwiająca rozwiązywanie działań z zachowaniem kolejności ich wykonywania. Dodatkowo program posiada wiele użytecznych funkcji, takich jak: rozwiązywanie równań, konwersje systemów liczbowych, rysowanie...                                   | 29.07.2007        | 6 548         | 7,3                | testowa (demo)                                   |
| HEXelon MAX 6.07<br>Najnowsza wersja zaawansowanego i, co ważne, całkowicie bezpłatnego kalkulatora matematycznego autorstwa Jerzego Znamierowskiego. Program pozwala na rozwiązywanie ciągów działań w postaci np.: $\sin(Pi) - \text{Pierwiastek}(-8/3)$ itp., posiada...                           | 29.06.2007        | 111 034       | 8,8                | bezpłatna                                        |
| Kalkulator Plus 1.0<br>Kalkulator Plus to znacznie rozbudowany wobec systemowego, bezpłatny kalkulator firmy Microsoft umożliwiający nie tylko przeprowadzanie różnego rodzaju obliczeń ale także konwersję pomiędzy różnymi jednostkami miary i walutami. W stosunku do...                           | 25.06.2005        | 23 986        | 7,1                | bezpłatna                                        |
| Karen's Calculator 1.2<br>Bezpłatny kalkulator wyróżniający się możliwością operowania na bardzo długich liczbach. Oprócz standardowych klawiszy numerycznych oraz klawiszy do podstawowych operacji matematycznych (dodawanie, odejmowanie, mnożenie, dzielenie, wynik, zapis...                     | 19.06.2007        | 9 364         | 6,8                | bezpłatna, z wyłączeniem zastosowań komercyjnych |

Rysunek 1: Dobreprogramy.pl Strona kategorii kalkulatory

## **2.2.2. Platformy dystrybucyjne**

Oddzielną kategorię stanowią platformy dystrybucyjne. W tej części zostanie przedstawiony oraz omówiony „Steam” oraz „App Store”. W odróżnieniu od zwykłych stron z programami, w tym podejściu niezbędne jest posiadanie specjalnego oprogramowania klienckiego umożliwiającego dostęp do systemu.

### **2.2.2.1. STEAM**

Jest systemem dystrybucji, zarządzania prawami cyfrowymi, platformą do gry sieciowej i komunikacji opracowaną przez firmę Valve Corporation. Nas będą interesowały dwie pierwsze funkcjonalności. Jego podstawową funkcją jest dystrybucja gier przez internet.

Obecnie jest dostępnych ponad 700 tytułów a liczba aktywnych użytkowników w 2009 roku przekroczyła 20 milionów. Obecnie kontroluje ok 70% rynku cyfrowej dystrybucji gier. Wielu wydawców współpracuje ze Steam, są tą m.in. Electronic Arts, Activision, 2K Games, Ubisoft, THQ, Sega, Codemasters, LucasArts, id Software.

Steam pozwala na zakup całkowicie cyfrowo. Wszystkie zakupione produkty są wiecześnie przywiązane do konta. Użytkownicy, którzy kupili pudełkowe wersje produktów sprzedawane w sklepach detalicznych mają możliwość zarejestrować go za pomocą klucza w ten sam sposób co przy cyfrowym zakupie. Jest bardzo rozwinięty od strony społecznościowej dzięki funkcjonalności komunikatora oraz możliwości tworzenia grup użytkowników. Steam akceptuje większość kart kredytowych jak również PayPal.

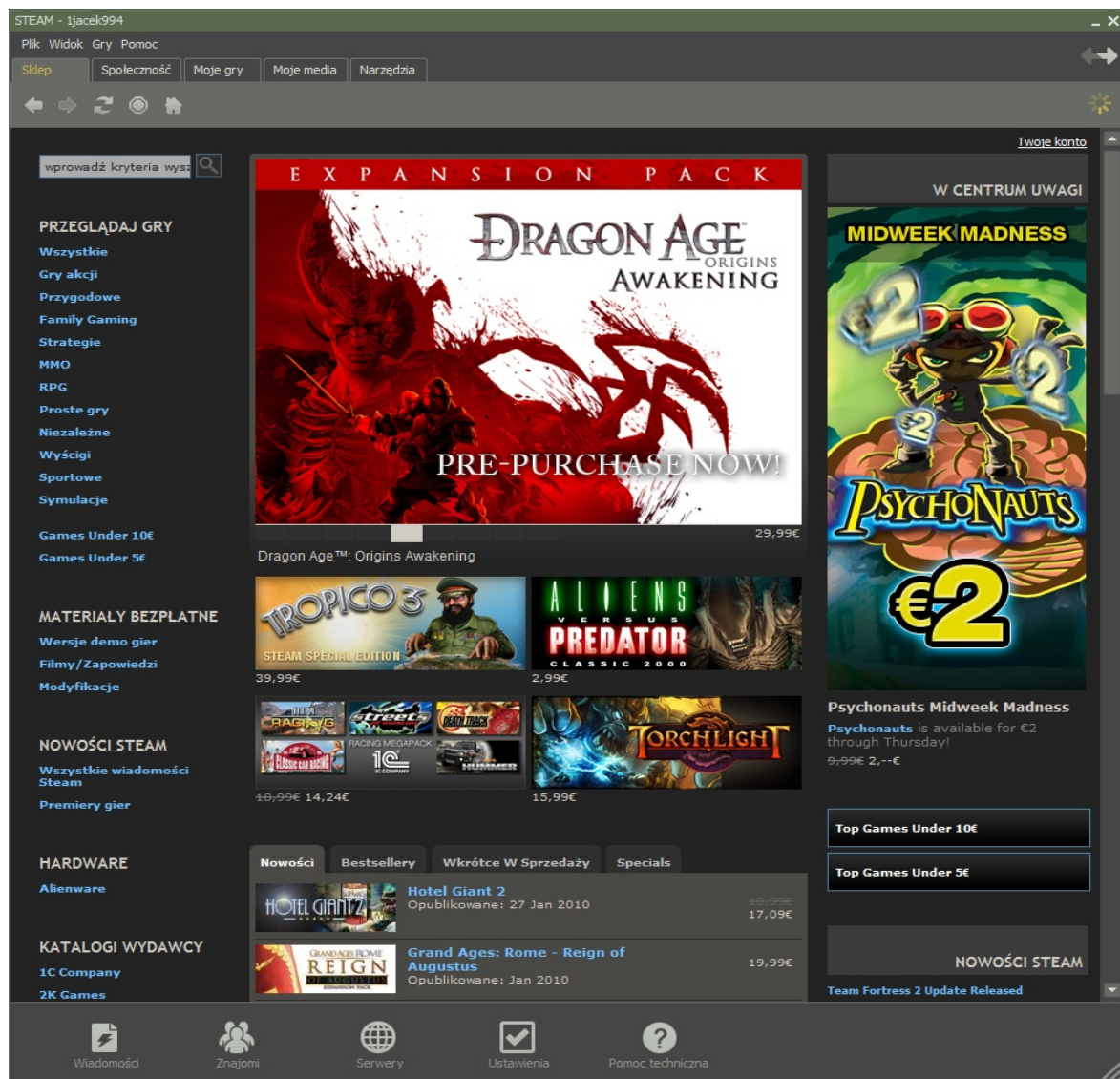
#### **Zalety**

- zautomatyzowany proces pobierania i aktualizowania
- system powiadamiania o aktualizacjach
- sieć społecznościowa
- rozproszony system
- ogromna skalowalność

## Wady

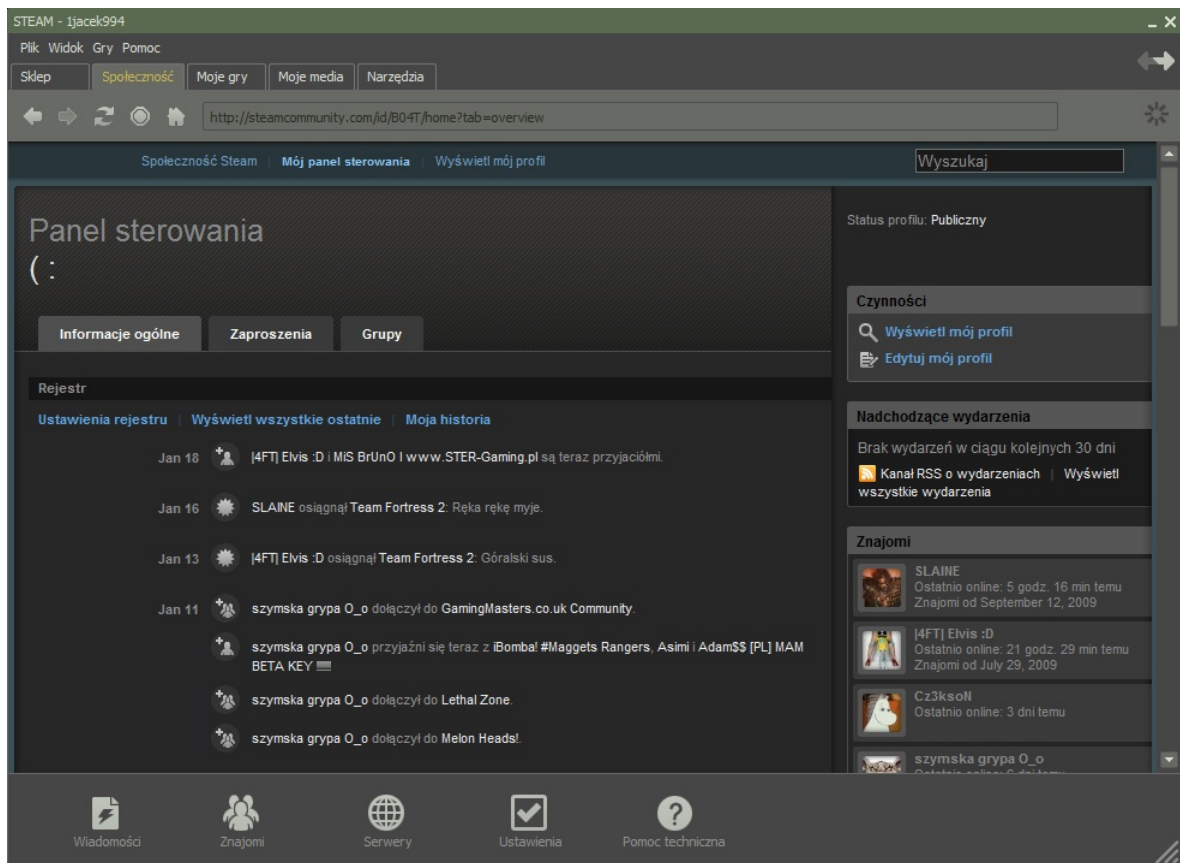
- aplikacje pisane specjalnie pod steam
- brak możliwości współpracy
- istnieje możliwość nielegalnego pobrania i użytkowania produktów

## Przykładowy zrzut ekranu

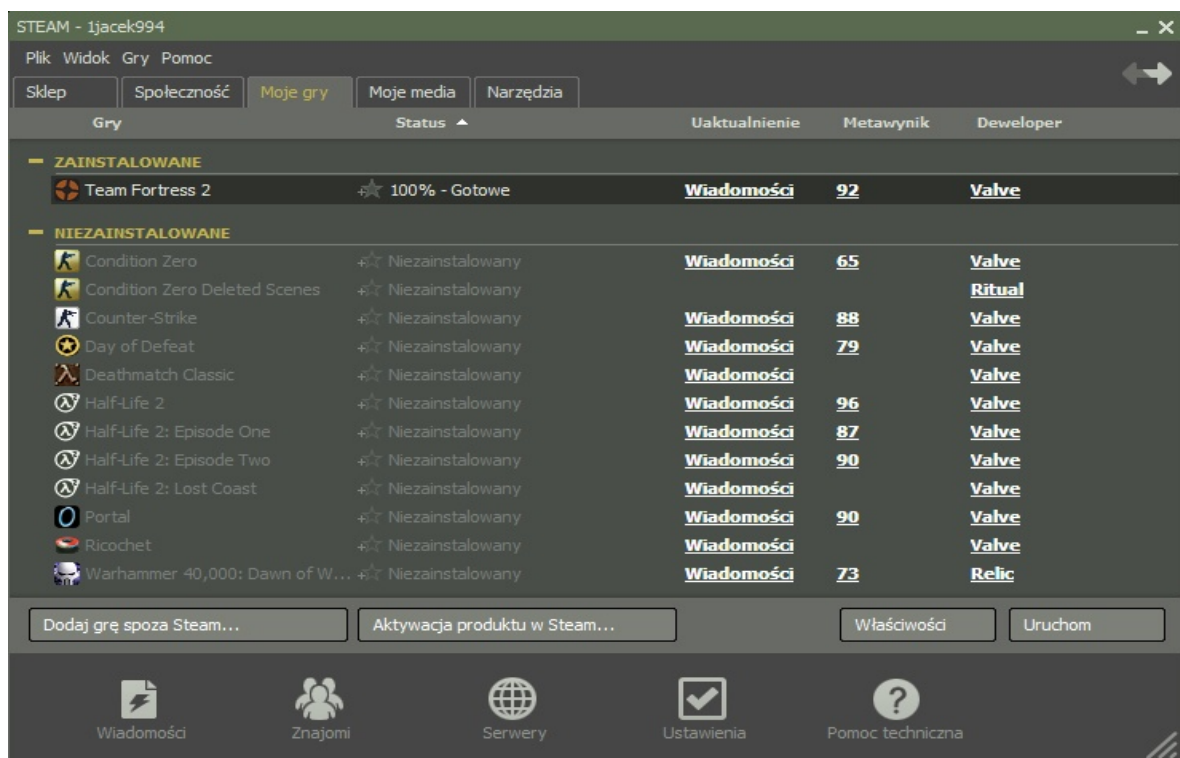


Rysunek 2: Steam





Rysunek 3: Steam - sieć społecznościowa



Rysunek 4: Steam - zakupione produkty

### **2.2.2.2. App Store**

Jest usługą Apple, która pozwala przeglądać i pobierać aplikacje dla produktów takich jak „iPhone”, „iPod touch” oraz „iPad”. Zadebiutował w sieci 10tego lipca 2008. Wszystkie publikowane programy pochodzą od samego Apple lub od osób posiadających pakiet iPhone SDK. Niestety wiąże się to z koniecznością posiadania systemu operacyjnego Mac OS czyli w praktyce komputer od Apple.

Wspierane jest zarówno oprogramowanie płatne jak i darmowe. Wszelkie płatności dokonywane są za pomocą karty kredytowej w jednej z „globalnych walut” takich jak dolar, euro lub funt szterling. Jeśli chodzi o polski region to transakcje rozliczane są w euro.

Jeśli publikujesz swój płatny program w App Store to 30% z jej ceny stanowi marża serwisu. Zanim program zostanie opublikowany, musi przejść weryfikację. Zgodnie z regulaminem Apple może bez podania przyczyny odmówić publikacji programu. Istnieje ryzyko, że nasz program nie zostanie opublikowany. App Store posiada również mechanizm aktualizacji programów. Użytkownicy są informowani o pojawieniu się nowszych wersji programów, które mają zainstalowane.

#### **Zalety**

- możliwość udostępniania własnych programów
- wygodny system płatności
- programy są weryfikowane przed opublikowaniem
- ogromny rynek (ok 120 tysięcy aplikacji i ponad 50 milionów użytkowników)

#### **Wady**

- ograniczenie do jednej platformy
- konieczność posiadania Mac'a (dla developera)
- ograniczone prawa (Apple może bez podania przyczyny wycofać nasz produkt)



## Przykładowe zrzuty ekranu



Rysunek 5: AppStore - kolekcja aplikacji



Rysunek 6: AppStore - mobilny klient

## **2.3. Aspekty prawne**

Zanim dokładnie omówimy opracowane przez nas rozwiązanie, chcemy w tym rozdziale przedstawić zebrane przez nas wszystkie aspekty prawne mające wpływ na przyjmowane rozwiązania.

### ***Definicja usługodawcy***

Jest osobą, która stale lub incydentalnie świadczy usługi drogą elektroniczną.

### ***Definicja usługobiorcy***

Jest osobą, która korzysta z usługi świadczonej drogą elektroniczną.

### ***Zawarcie umowy***

Umowa jest czynnością prawną, polegającą na zgodnym oświadczeniu woli dwóch stron zmierzających do wywołania określonych w jej treści skutków prawnych. Wola osoby dokonującej czynności prawnej może być wyrażona przez każde zachowanie się tej osoby, które ujawnia jej wolę w sposób dostateczny, w tym również przez ujawnienie tej woli w postaci elektronicznej. Niezbędne jest zawarcie umowy między usługodawcą a użytkownikami końcowymi (usługobiorcą), regulującej zasady korzystania z usługi.

Ustalony przez jedną ze stron wzorzec umowy, w szczególności regulamin wiąże drugą stronę, jeżeli zostały jej przedstawione przed realizacją usługi. Udostępnienie jej w postaci elektronicznej należy rozumieć jako możliwość zapoznania się z jej treścią a nie koniecznie jego fizycznego przekazywania. Za miejsce zawarcia umowy w postaci elektronicznej uznawana jest siedziba usługodawcy a nie pochodzenie usługobiorcy.

### ***Usługa świadczona drogą elektroniczną***

Wykonanie usługi polegającej na wysyłaniu i odbieraniu danych za pomocą systemów informatycznych, wykonywane na indywidualne żądanie usługobiorcy, bez ich obecności. Dane te oczywiście są przesyłane z użyciem sieci a nie np. firmy kurierskiej. Pojęcie indywidualne żądanie należy rozumieć jako każdą czynność, przez którą konkretny usługobiorca wyraża wolę świadczenia mu usługi, która może mieć zarówno postać odpłatną lub nieodpłatną.

Elektroniczna dystrybucja oprogramowania jest niewątpliwie przykładem usługi świadczonej drogą elektroniczną.

## **Obowiązki informacyjne nałożone na usługodawców**

Usługodawca zobowiązany jest do podania w sposób wyraźny i jednoznaczny następujących informacji:

- kontakt elektroniczny – np. adres e-mail, nr komunikatora itd.
- imię i nazwisko lub nazwę firmy oraz adres siedziby
- NIP i informacje o organie, który wydał zezwolenie jeżeli usługodawca jest przedsiębiorcą
- szczególnych zagrożeniach związanych z korzystaniem z usługi
- funkcji i celu oprogramowania lub danych niebędących częścią usługi

## **Regulamin**

Usługodawca określa regulamin świadczenia usług. Nieodpłatnie udostępnia go usługobiorcy przed świadczeniem usług. Usługobiorca nie jest związany tymi postanowieniami regulaminu, które nie zostały mu udostępnione we właściwy sposób. Usługodawca zobowiązany jest do świadczenia usług zgodnie z regulaminem.

Regulamin określa w szczególności rodzaje i zakres usług świadczonych drogą elektroniczną, warunki na których jest świadczona usługa, wymagania techniczne oraz pozatechniczne jak np. zakaz dostarczania przez usługobiorcę treści o charakterze bezprawnym.

## **Definicja informacji handlowej**

Każda informacja przeznaczona bezpośrednio lub pośrednio do promowania towarów lub usług.

## **Ochrona przez niezamówioną informacją handlową**

Informację handlową uważa się za zamówioną, jeżeli odbiorca wyraził zgodę na otrzymanie takiej informacji, w szczególności udostępnił w tym celu identyfikujący go adres elektroniczny. Samo udostępnienie adresu nie oznacza zgody na otrzymywanie w przyszłości informacji handlowych. Kto przesyła za pomocą środków komunikacji elektronicznej niezamówione informacje handlowe, podlega karze grzywny do 5 tys. zł.

## **Wyłączenie odpowiedzialności**

Nie ponosi odpowiedzialności za przechowywane dane ten, kto udostępniając zasoby systemu nie wie o bezprawnych charakterze danych, a w razie otrzymania informacji urzędowego

zawiadomienia lub uzyskania wiarygodnej wiadomości o bezprawnym charakterze danych niezwłocznie uniemożliwi dostęp do tych danych.

Usługodawca, który otrzymał urzędowe zawiadomienie i uniemożliwił dostęp do tych danych, nie ponosi odpowiedzialności względem tego usługobiorcy za szkody powstałe w wyniku blokady dostępu do tych danych..

### ***Brak konieczności monitoringu danych***

Podmiot, który świadczy usługi określone jako hosting, nie jest obowiązany do sprawdzania przekazywanych, przechowywanych lub udostępnianych przez niego danych.

### ***Ochrona danych osobowych***

Usługodawca świadczący usługi drogą elektroniczną zobowiązany jest do ochrony danych osobowych usługobiorców.

### ***Zasady ochrony danych osobowych***

Nie można zestawiać danych osobowych usługobiorcy z przybranym przez niego loginem (pseudonimem).

### ***Zakres ochrony programów komputerowych***

Programy komputerowe podlegają ochronie jak utwory literackie. Wyróżniamy Autorskie prawa osobiste i majątkowe.

Osobiste chronią więc twórcy z utworem natomiast majątkowe obejmują prawo twórcy do korzystania z utworu i rozporządzania nim oraz do wynagrodzenia za korzystanie z utworu.

Oprogramowanie jest dość specyficznym produktem. Nigdy nie kupujemy programów lecz prawo do użytkowania ich zgodnie z licencją. W tym rozdziale zostanie wyjaśnione czym jest licencja oraz dlaczego jest ważna.

## **2.4. Licencje**

### ***Czym jest licencja***

W przeciwieństwie do zwykłych towarów materialnych oprogramowanie, muzyka czy film nie jest nabywana na własność. W przypadku wyżej wymienionych dzieł nabywamy jedynie prawo do używania.

Licencja jest umową zawartą między właścicielem dzieła a użytkownikiem. Jest więc zezwoleniem na używanie. Oznacza to, że twórca decyduje o zakresie i warunkach udostępniania swojego dzieła. Umowa ta ma postać dokumentu. Bardzo często jest zintegrowana np. w postaci części procesu instalacyjnego. Określa prawa i ograniczenia nałożone na użytkownika. Umowa ta ma na celu ochronę praw autorskich twórców i praw własności intelektualnych przez nałożenie ograniczeń na użytkownika np. instalacja na określonej liczbie stanowisk, edytowania kodu, stosowania reverse engineering itp.

Większość umów zawiera również klauzulę wyłączającą z odpowiedzialności za skutki użytkowania.

### ***Dlaczego jest ważna***

Wybór licencji jest krytyczny z punktu widzenia wydawcy ze względu na ochronę praw intelektualnych oraz potencjalną możliwość generowania dochodów. Ważne jest również, aby uchronić się od różnych kwestii prawnych, które wynikają z upublicznienia dzieła.

### ***Przykłady licencji***

EULA (End User License Agreement). Termin ten dotyczy oprogramowania komercyjnego. Licencje te są bardzo zróżnicowane jednak posiadają pewne cechy wspólne. Każda licencja typu EULA zastrzega wszelkie prawa dla producenta oprogramowania. Użytkownik otrzymuje tylko ściśle zdefiniowany, wąski zakres uprawnień.

Adware. Licencji oprogramowania zamkniętego, bezpłatnego. Licencja, które w zamian za darmową wersję programu w większości przypadków wyświetla reklamy. Reklamy są kluczowym

elementem licencji. Blokowanie reklam jest jednoznaczne z niespełnieniem warunków licencji.

Freeware. Licencja dla programów udostępnianych bezpłatnie natomiast bez możliwości wglądu w kod źródłowy. Nie zawsze zezwala na wykorzystanie w celach komercyjnych. Nie wolno ich sprzedawać, ani dokonywać w nich zmian.

Shareware. Licencja która umożliwia bezpłatne używanie i za zgodą autora rozpowszechnianie. Korzystanie z niego jest już jednak w pewnym stopniu ograniczone. Jeżeli po upływie okresu testowego użytkownik zdecyduje się dalej używać lub przechowywać program na, powinien go kupić. Niekiedy pewne funkcje programów shareware są zablokowane i dopiero po rejestracji użytkownik otrzymuje hasło, które pozwala je uaktywnić.

GNU General Public License (GNU GPL lub po prostu GPL). Jest często stosowaną licencją otwartego oprogramowania. Pozwala na bezpłatne korzystanie w dowolnym celu oraz zezwala na modyfikowanie. Programy na tej licencji można również dowolnie rozpowszechniać oraz poprawiać (jednak należy te poprawki udostępnić publicznie). Zakazuje redystrybucji oprogramowania w formie czysto binarnej. Jeżeli ktoś wprowadza do obiegu oprogramowanie zawierające jakkolwiek część podlegającą licencji GPL, to musi udostępnić wraz z każdą dystrybucją binarną jej postać źródłową.

GNU Lesser General Public License (LGPL), Jest zmodyfikowaną, bardziej liberalną wersją GPL. Różni się od GPL głównie tym, że pozwala na łączenie z programami nieobjętymi licencjami GPL czy LGPL, niezależnie czy będą one wolnym czy własnościowym oprogramowaniem.

Public Domain. Licencja czyniąca z dzieła własność ogółu, w myśl której autor lub autorzy zrzekają się praw do upowszechniania na rzecz ogółu użytkowników. Czyli wszelkie oprogramowanie, zdjęcia oraz inne dobra dla których wygasły prawa autorskie, bądź takie, które od samego powstania nie były objęte prawem autorskim. Dzięki czemu są one dostępne dla każdego, do wszystkich zastosowań.

## **Wnioski**

Wymienione wcześniej licencje są tylko częścią najczęściej stosowanych. Każda licencja posiada swoje aspekty m.in.:

1. Ekonomiczne:

- licencje płatne
- licencje bezpłatne

2. Czasowe:

- licencja wieczysta
- licencja ograniczona czasowo

3. Zastosowanie:

- licencja komercyjna
- licencja do użytku domowego

4. Formy dystrybucji:

- licencje zamknięte (bez kodu źródłowego)
- licencje otwarte (z kodem źródłowym)

Każdy aspekt determinuje sposób dystrybucji oraz warunki użytkowania. Tworząc system do dystrybucji treści objętych prawami autorskimi i intelektualnymi należy dostosować proces dystrybucji tak aby spełniał on warunki licencji.

### 3. Proponowane rozwiązanie

W tym rozdziale chcemy przedstawić nasze rozwiązania i koncepcję podejścia do zagadnienia elektronicznej dystrybucji oprogramowania. Każdy rozdział zawiera nasze rozważania na temat problemów i możliwości ich rozwiązania. Tematem naszych rozważań między innymi są:

- Problem piractwa oraz możliwe metody walki z nim
- Potrzeba aktualizacji oraz sprawa nowych wersji programów
- Aspekty finansowe na które składają się między innymi:
  - ceny i dostępne waluty w tym bardziej sprawiedliwa polityka cenowa
  - problem naliczania prowizji
  - sposoby rozliczania
- Zagadnienia prawne takie jak:
  - problem licencji
  - weryfikacja oprogramowania i nasza odpowiedzialność za nie
  - regulamin

Na końcu rozdziału zawarty jest opis i specyfikacja całego systemu.

#### 3.1. Zabezpieczenie oprogramowania:



*Rysunek 7: Pirat*

Piractwo komputerowe stanowi poważny problem dla całego przemysłu oprogramowania. Twórcy oraz wydawcy oprogramowania ponoszą straty z każdą nielegalną kopią oprogramowania w użyciu. Polityka deweloperów zmierza do ponoszenia coraz większych nakładów czasu i pieniędzy na rzecz ograniczenia piractwa komputerowego. Wielu z nich skupia się na problemie piractwa z perspektywy użytkownika końcowego. Tymczasem użytkownicy końcowi mogą rozpowszechniać i korzystać z nielegalnego

oprogramowania, dopiero po uprzednim zdjęciu zabezpieczeń przed nielegalnym



kopiowania i rozpowszechniania przez „crackera”.

Skoro zdjęcie zabezpieczeń jest tak trudne i czasochłonne to dlaczego znajdują się ludzie, którzy angażują się w tę działalność? Niejednokrotnie są to ludzie młodzi i wykształceni. Czy chodzi o wyzwanie czy może sławę, uznanie i wyższy status społeczny?

Zdaniem doktora Sigi Goode w pracy *“What Motivates Software Crackers?”*, *Journal of Business Ethics*, Vol. 65, No. 2, 2006, 173-201 Prof Sigi Goode and Sam Cruise znacząca większość crackerów uważa, że za oprogramowanie powinno się płacić, jednak nie podoba im się polityka cenowa firm zajmujących się dystrybucją oprogramowania. Ich zdaniem firmy te zarabiają na tyle dużo, i że obecne ceny są znacznie zawyżone. Kolejnym ważnym czynnikiem jest stopień trudności, im bardziej skomplikowane zabezpieczenia tym większa pokusa dla „crackerów”.

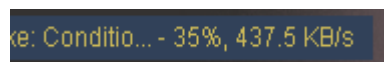
### **3.1.1. Stosowane podejścia**

#### **3.1.1.1. Rygorystyczne**

Przykładem rygorystycznego podejścia do sprawy charakteryzują się takie platformy jak między innymi „Steam” czy „App Store”. W tej części skupimy się tylko na „Steam” ponieważ posiada on cechy wystarczające do zademonstrowania idei takiego podejścia. Należy wspomnieć, że jest ono coraz bardziej popularne.

„Steam” jak platforma do elektronicznej dystrybucji oprogramowania charakteryzuje się bardzo silną integracją ze swoim klientem *Rysunek 2: Steam*. Nie inaczej jest w kwestii bezpieczeństwa.

Przed wszystkim jako użytkownicy końcowi jesteśmy pozbawieni instalatora w formie do jakiego wszyscy jesteśmy przyzwyczajeni. Proces instalacji odbywa się poprzez klienta „Steam”, zaczyna się i kończy na wybraniu opcji „zainstaluj” na požądanej przez nas aplikacji. Od tego momentu nie jest potrzebna już żadna nasza ingerencja.



*Rysunek 8: Pobieranie - instalacja aplikacji w Steam*

Nie doczekamy się chociażby pytania o docelowe miejsce instalacji programu. Spowodowane jest to tym, że „Steam” sam pobiera wszystkie niezbędne pliki i umieszcza je w swoich katalogach.

---

|                 |                                               |
|-----------------|-----------------------------------------------|
| Adres URL:      | <input type="text" value="steam://runid/80"/> |
| Klawisz skrótu: | <input type="text" value="Brak"/>             |
| Liczba wizyt:   | Nieznany                                      |

---

Zmień ikonę...

*Rysunek 9: Przykład skrótu aplikacji w Steam*

Po udanej instalacji jedyne co może rzucać się w oczy to brak katalogu dla zainstalowanej aplikacji w pasku „menu start”. Jedyńm śladem w pasku jest skrót umieszczony w katalogu „Steam” [Rysunek 9: Przykład skrótu aplikacji w Steam](#) oraz ewentualnie również na pulpicie. Sam skrót też działa na innej zasadzie niż moglibyśmy się tego spodziewać. Podobnie wygląda sytuacja z plikami „.exe” zainstalowanego programu. Otóż każde uruchomienie aplikacji sprowadza się do uruchomienia klienta „Steam” z odpowiednimi parametrami wskazującymi o jaką aplikację nam chodzi. Przed uruchomieniem odbywa się nasza autoryzacja z użyciem „Steam”. Sprawdzane jest czy taki użytkownik nie jest już zalogowany w innym miejscu, czy w ogóle taki użytkownik istnieje i posiada taką aplikację oraz czy jego konto nie zostało zablokowane. Dopiero po uruchomieniu klienta rozpoczyna się uruchamianie naszej aplikacji. Oczywiście w trakcie użytkowania jej, klient musi cały czas być włączony oraz musi mieć zapewniony kontakt z serwerem autentykacji. W przypadku wyłączenia klienta celowo czy w wyniku błędu lub utraty połączenia nasza aplikacja zostanie zamknięta. Oczywiście można uruchomić klienta w trybie „offline”, jednak w takim wypadku nie wszystkie programy mogą zostać uruchomione a pozostałe mają ograniczoną funkcjonalność.

### **Cechy:**

- Silna integracja aplikacji z platformą
- Brak typowych instalatorów oraz zautomatyzowany z klientem proces instalacji
- Brak typowego katalogu instalacyjnego
- Uruchamianie aplikacji poprzez klienta
- Ciągły nadzór nad użytkownikami

## **Wnioski**

Podejście rygorystyczne w pewnym stopniu ogranicza „piractwo” a właściwie jego skutki. Zastosowane rozwiązania skutecznie powstrzymują przeciętnych użytkowników przed próbą niedozwolonego użytku lub rozpowszechniania. Jednak nie jest to rozwiązanie idealne i nie stanowi żadnego problemu dla crackerów. Mimo zastosowanych mechanizmów w sieci można bez większego problemu znaleźć pełno funkcjonalne programy, które pochodzą z tych elektronicznych dystrybucji. Zastosowane podejście nie bierze pod uwagę przyczyn piractwa czyli crackerów i motywacji jakimi się kierują.

Każde zabezpieczenie przed piractwem, niesie ze sobą pewne niedogodności i czasami nawet frustracje dla użytkowników legalnie nabytych programów. Nie inaczej jest w tym przypadku. Do głównych niedogodności należy zaliczyć konieczność posiadania dostępu do internetu w trakcie użytkowania aplikacji. Nie każdą aplikację da się uruchomić w trybie offline, który w niesie ze sobą ograniczenie funkcjonalności. Istnieje jeszcze ryzyko, że nawet jeśli mamy dostęp do internetu to w przypadku awarii „Steam” a dokładniej serwera odpowiedzialnego z autentykację użytkowników, nie można uruchomić aplikacji. Jeśli awaria wystąpiła w trakcie użytkowania aplikacji to istnieje dość spore ryzyko, że nasz program zostanie zamknięty.

Kolejną wadą jest wydłużony czas uruchamiania aplikacji średnio o około minuty. Spowodowane jest to koniecznością uruchomienia klienta przed docelową aplikacją. W skrajnych przypadkach czas ten wydłuża się nawet o kilkanaście minut. Skrajny przypadek to oczywiście dość częsta aktualizacja klienta „Steam”.

Obecnie ilość zużywanych zasobów systemowych ma coraz mniejsze znaczenie dla użytkowników. Jednak konieczność posiadania uruchomionego w tle klienta wiąże się z dodatkowy zużyciem pamięci i w małym stopniu procesora.

### **3.1.1.2. Nierygorystyczne**

Dobrym przykładem takiego podejścia charakteryzują się serwis „dobreprogramy.pl”. W przeciwieństwie do rygorystycznego, nie oferuje ani nie narzuca swojego mechanizmu zabezpieczenia oprogramowania. Jest to zbliżone podejście do tego stosowanego przez dystrybutorów pudełkowych gdzie dystrybutor jest jedynie pośrednikiem między twórcą a użytkownikiem programu.

## **Wnioski**

Podstawowym założeniem takiego podejścia jest to, że twórcy oprogramowania jeśli chcą się zabezpieczyć przed nielegalnym użytkowaniem to robią to na własną rękę. Podobnie jest w przypadku dystrybucji pudełkowych. Oczywiście nie podnosi to poziomu bezpieczeństwa (wątpliwego biorąc pod uwagę łatwość w dostępie do pirackiego oprogramowania). Jako zaletę nierygorystycznego podejścia należy wymienić brak konieczność ingerencji w kod publikowanych aplikacji w celu dostosowania ich specjalnie do platformy dystrybucyjnej. Ponadto jest znacznie bardziej elastyczne ponieważ nie ogranicza możliwych aplikacji ani pod kątem języka w jakim zostały napisane, ani docelowego systemu.

## **Podsumowanie**

Niestety mimo coraz większych nakładów czasu i pieniędzy, nikomu jak dotąd nie udało się w 100% zabezpieczyć swoich programów przed nielegalnym kopiowaniem i użytkowaniem. Nawet w przypadku aplikacji pisanych pod konkretne urządzenia jak „iPhone”, różnego rodzaju urządzeń do nawigacji GPS z mapami czy konsol w których występują zabezpieczenia na poziomie sprzętowym.

Wszystkie restrykcyjne rozwiązania mają jedną wspólną cechę, będącą zarazem ich wadą. Mianowicie nie uwzględniają przyczyn piractwa, koncentrując się jedynie na skutkach piractwa. Przytłaczająca większość „piratów” nie łamie zabezpieczeń czy luk w systemie dystrybucji lecz szuka w internecie już gotowych i opracowanych nielegalnych wersji. Wersje te są przygotowywane przez grupy „crackerów”, którzy nie są motywowani chęcią zysku. Większość z nich robi to z samego faktu wyzwania. Im mocniejsze zabezpieczenie tym większe wyzwanie, satysfakcja z faktu złamania go, oraz uznanie. Robią to także z pobudek ideologicznych. Nie podoba im się polityka cenowa firm zajmujących się tworzeniem i dystrybucją oprogramowania.

Ponadto osoby które nie chcą płacić za oprogramowanie zawsze znajdą powód i sposób by tego nie robić. Dowodzi tego obecna sytuacja, w której niejednokrotnie łatwiej jest zdobyć i uruchomić nielegalną wersję oprogramowania od legalnej i zabezpieczonej wersji.

Naszym zdaniem nie ma powodów dla których uczciwi użytkownicy powinni odczuwać niedogodności związane z zastosowaniem dodatkowych mechanizmów zabezpieczenia oprogramowania. Zabezpieczenia te są zwykle bardziej dotkliwe dla zwykłych użytkowników niż ich nieuczciwych kolegów.

Dlatego wybieramy rozwiązanie nierestrykcyjne, które nie oznacza że twórcy nie mogą zabezpieczyć swoje programy np. poprzez system kluczy i dodatkowych aktywacji.

Przyczyny piractwa nie należy doszukiwać się w niskim poziomie zabezpieczeń lecz w problemie społecznym polegającym na złych relacjach użytkownik-twórca a twórca-użytkownik oraz na bardzo niskim poziomie egzekwowania kar za przestępstwo jakim jest nielegalne użytkowanie oprogramowania. Znacznie większe korzyści przyniosłaby kampania społeczna o szkodliwych skutkach używania nielegalnego oprogramowania połączona z bardziej sprawiedliwą polityką cenową.

### **3.2. Aktualizowanie i powiadamianie o nowszych wersjach:**

Programy mają to do siebie, że prędzej czy później pojawiają się ich nowsze wersje. Aktualizacje są konieczne zarówno w przypadku potrzeby naprawy błędów występujących w programie czy choćby rozbudowy o nową funkcjonalność bądź dostosowania do zaszłych zmian jak np. zmiana podatku VAT. Sam proces aktualizacji polega na zastąpieniu poprzedniej wersji oprogramowania na nowszą.

W tym rozdziale będziemy rozważać następujące rozwiązania:

- czy zastosować mechanizm aktualizacji czy nie?
- czy zautomatyzowane pobieranie czy może potrzebna ingerencja użytkownik?
- czy archiwizować poprzednie wersje czy udostępniać tylko najnowszą wersję?
- czy nowsze wersje programów powinny być darmowe, płatne czy może powinna decydować o tym twórca programu?
- jaki zastosować system powiadamiania?

Większe i bardziej skomplikowane programy posiadają własne mechanizmy aktualizacji, natomiast mniejsze i mniej skomplikowane wydają co jakiś czas nowsze wersje. Przy obecnie dostępnych zasobach i szybkości łącz, brak mechanizmu aktualizacji będzie miał mały wpływ na cały system.

Dodatkowo zastosowanie własnego mechanizmu aktualizacji wymuszałoby konieczność ingerencji w kod publikowanych programów.

Kolejnym aspektem jest to czy aktualizacje powinny być pobierane automatycznie, czy może powinien być w nią zaangażowany użytkownik. Z jednej strony wygoda, z drugiej większa kontrola nad tym co się dzieje w naszym komputerze. Ilu z nas nie przytrafiła się aktualizacja w najmniej oczekiwanym i odpowiednim momencie. Zautomatyzowane aktualizacje niosą ze sobą szereg niedogodności. W przypadku systemów operacyjnych są to np. nieplanowany reset komputera. Może się okazać że nowsza wersja programu ma zmienioną w stosunku do poprzedniej wersji funkcjonalność, która nie spełnia już naszych wymagań. Naszym zdaniem ręczna aktualizacja nie jest na tyle uciążliwa żeby była konieczne zautomatyzowanie jej.

Wszystkie omawiane w stanie sztuki rozwiązania nie archiwizowały poprzednich wersji programów lecz zawsze oferują tylko najnowszą wersję. Inaczej jest w przypadku stron autorów programów gdzie zwykle oprócz najnowszej stabilnej wersji oferowana jest nowsza wersja jeszcze w fazie bety. Naszym zdaniem najlepszym rozwiązaniem jest udostępnianie tylko najnowszych wersji i zapewnienie informacji dla użytkowników na temat strony domowej programu (jeśli takowa istnieje i autor umieści o niej informację w systemie) gdzie użytkownicy w ramach potrzeby będą mogli ściągnąć program w innej wersji niż aktualnie opublikowana wersja.

Kwestią sporną jest to czy nowsze wersje programów powinny być płatne. Z jednej strony mamy do czynienia z interesem użytkowników, którzy już raz zapłacili za program (jeśli jest płatny), z drugiej strony interes autorów programów, którzy poświęcają swój czas i środki w rozwój programu. Prawda jest taka, że jako pośrednik w dystrybucji nie mamy możliwości odgórnie wyegzekwowania ustalonej przez nas polityki. Dlatego uważamy, że jako pośrednik musimy zapewnić autorom możliwość zadecydowania czy nowe odsłony ich programów będą płatne dla użytkowników poprzednich wersji. Autorzy zawsze mogą opublikować nowszą wersję programu jako nową publikację (i pobierać nowe opłaty) lub jako aktualizacje do poprzedniej wersji (i udostępniać ją bez opłat dla użytkowników którzy już ją posiadają).

Na sam koniec pozostanie kwestia powiadomienia użytkowników o aktualizacjach i zmianach w programach które użytkują. Użytkownicy po zalogowaniu powinni mieć zapewniony dostęp do wiadomości w których zawarta, będzie informacja o pojawieniu się nowszych wersji programów, które posiadają oraz o zmianach jakie w nich zaszły w stosunku do poprzedniej wersji.

### **3.3. Finansowanie zakupów oprogramowania**

#### ***Problem***

Wszystkie obecnie używane systemy dystrybucji elektronicznej wspierające odpłatną dystrybucję stosuje najczęściej jedną lub dwie waluty. Najpopularniejsze to oczywiście dolar i euro. Stanowi to problem dla użytkowników nie posiadających żadnej z tych walut. Zarówno przy wykonywaniu płatności jak i ciągłych zmianach kursów walut (np. nieopłacalność zakupu przy obecnym kursie i konieczność czekania na jego obniżenie lub rezygnację).

Innym zagadnieniem jest polityka cenowa. Przy zastosowaniu jednej czy dwóch walut mamy sytuację w której jest ustalona jedna zglobalizowana cena nie uwzględniająca różnic między odmiennymi rynkami. Nawet w tak specyficznym rejonie jak Europa gdzie większość krajów posiada wspólną walutę takie rozwiązanie nie wydaje się sprawiedliwe. Płace i koszty wewnątrz strefy euro, mimo że ponoszone w jednej walucie są od siebie diametralnie różne. Przykładowa kwota dwudziestu euro jest nieporównywalnie większym kosztem dla Słowaka niż dla przeciętnego Niemca.

Jest to jedna z głównych i zarazem podstawowych przyczyna dlaczego oprogramowanie z elektronicznych dystrybucji nie odbiega cenowo od ich pudełkowych wersji a niekiedy nawet przy wysokim kursie waluty nawet droższe. Dodatkowo lokalni dystrybutorzy wersji pudełkowych mają możliwość wynegocjowania ceny dostosowanej do realiów rynkowych tymczasem dystrybutorzy elektroniczni posiadają jedną zglobalizowaną cenę.

#### ***Rozwiązanie***

Istnieje rozwiązanie pozbawione powyższych wad. Rozwiązanie zupełnie odmienne od stosowanych obecnie rozwiązań.

Zamiast stosowania popularnych walut proponujemy system wirtualnych punktów. Punkty te są przypisane do konta użytkownika na stałe. Konto jest doładowywane za pomocą wspieranych form płatności. Technicznie przy takim założeniu jest możliwe wsparcie znacznie większej liczby walut niż przy tradycyjny zglobalizowanym podejściu. np. wystarczy zaimplementować płatności SMS dla każdego kraju z osobna. Na tym etapie możliwe jest również dostosowanie polityki cenowej do warunków ekonomicznych rejonu z którego pochodzi użytkownik przez zastosowanie zindywidualizowanego przelicznika na wirtualne punkty. Trzeba również uwzględnić koszty związane z obsługą płatności, podatkami i oczywiście prowizją.

Uważamy, że przedstawione podejście jest bardziej praktyczne i sprawiedliwe dla użytkowników. Oczywiście proponowane rozwiązanie nie jest pozbawione wad. Jedną z nich jest mała przejrzystość cen. Użytkownik widząc cenę w wirtualnych punktach ma trudność w ustaleniu rzeczywistej ceny programu. Kolejną niedogodnością jest brak możliwości wypłacania przez zwykłych użytkowników nadplanowych punktów np. w wyniku tego, że doładowanie jest większe niż cena aplikacji, którą kupujemy.

### **3.3.1. Rozliczanie**

#### ***Czas realizacji na wypłatę***

Najlepszym rozwiązaniem byłoby zapewnienie natychmiastowego dostępu do środków zebranych przez osoby publikujące płatne aplikacje w serwisie. Niestety istnieją pewne ograniczenia uniemożliwiające realizację takiego podejścia.

Przede wszystkim chodzi o specyficzne cechy i uwarunkowania niektórych sposobów płatności. Przykładowo chodzi o ograniczony dostęp do środków z naszej strony. Istnieją uwarunkowania zarówno czasowe jak i co do minimalnej ilości zebranych środków po spełnieniu których możemy wypłacić zebrane środki. Istnieje spore ryzyko, że mimo iż w systemie właściciel publikacji zbierał określoną kwotę, to my nie mamy jeszcze dostępu do tych środków.

Kolejnym aspektem jest kwestia bezpieczeństwa. Wydłużony czas realizacji wypłaty zwiększa szanse wykrycia potencjalnych nieprawidłowości. Przykładowo w trakcie oczekiwania może się okazać, że osoba publikująca aplikację nie miała do niej praw. Istnieje również ryzyko, że opublikowana aplikacja była próbą wyłudzenia pieniędzy od użytkowników bądź po prostu nie spełnia zapewnianych funkcji. Opóźnienie to daje niezbędny czas dla użytkowników na zgłoszenie nieprawidłowości oraz dla nas na sprawdzenie go i podjęcie adekwatnych działań.

#### ***Prowizja i opłaty***

Oczywiście serwis musi pobierać opłaty żeby się utrzymać i rozwijać. Istnieje sporo możliwości naliczania opłat.

Najprostszą z nich byłoby naliczanie jednorazowych opłat z każde pobranie plików. Wielkość opłaty może być uzależniona od wielkości programu lub jego ceny. Ponieważ naliczanie opłat nie jest zgodne z filozofią darmowego oprogramowania, opłaty takie mogłyby być naliczane tylko za programy, które zostały opublikowane jako płatne. Takie podejście mogłoby wywrzeć wrażenie, że użytkownicy płatnych programów są gorzej traktowani od użytkowników darmowych



ponieważ muszą płacić za każdorazowe ściągnięcie swoich programów.

Istnieje też możliwość naliczania opłat podczas zakupu programu. W takim przypadku do ceny programu naliczana byłaby dodatkowa kwota. Jej wysokość byłaby uzależniona od ceny. Im droższa aplikacja tym większa prowizja. W przypadku bardzo tanich aplikacji warto by było rozważyć możliwość zastosowania mechanizmu minimalnej kwoty prowizji stosowanej w przypadku kiedy procent od ceny byłby mniejszy od danej kwoty.

Możliwe jest również zastosowanie techniki kursu kupna i sprzedaży punktów czyli mechanizmu stosowanego między innymi w kantorach i handlu walutami. Naliczanie prowizji w takim przypadku odbywałoby się poprzez zróżnicowanie kosztu nabycia punktów przez użytkownika oraz rzeczywistej wartości tych punktów dla właścicieli publikowanych płatnych aplikacji.

Naszym zdaniem ostatnie podejście ze zróżnicowaniem ceny kupna i sprzedaży punktów jest najlepszym rozwiązaniem. Jest to najbardziej elastyczne podejście oraz najmniej odczuwalne dla użytkowników.

### **3.4. Zagadnienia powiązana z prawem**

#### **3.4.1. Sprawa licencji**

Czy jest licencja i jak ważną rolę dla twórców i użytkowników programów pełni opisaliśmy w rozdziale [2.4. Licencje](#). System elektronicznej dystrybucji musi posiadać mechanizm umożliwiający autorom programu określenie na jakiej licencji program jest publikowany. Z perspektywy użytkownika ważne jest, żeby wiedzieć na jakich zasadach i w jaki sposób będą mogli korzystać z programu zgodnie z wolą autorów.

Najprostszym sposobem byłoby stworzenie bazy najpopularniejszych licencji, z których publikujący swoje aplikacje autor musiałby wybierać najbardziej mu odpowiadającą. W podrozdziale [2.4. Licencje](#) „Przykłady licencji” wymieniono i opisano najważniejsze z nich. Jednak jak się okazuje jest to tylko kropla w morzu wszystkich stosowanych obecnie licencji. Okazuje się, że nie jest możliwe ręczne zdefiniowanie bazy licencji, które spełniłoby wszystkie oczekiwania autorów. Bardzo często zdarza się, że autorzy publikują i udostępniają swoje programy na własnych licencjach lub modyfikują gotowe licencje do swoich potrzeb.

Jedynym praktycznym rozwiązaniem jest umożliwienie zdefiniowania treści przez autora

publikacji na etapie jej tworzenia. Autor sam decyduje jaka treść i warunki znajdują się w jej treści. Serwis zapewnia, że użytkownik jeśli będzie chciał skorzystać z publikacji, będzie musiał zapoznać się z treścią dołączonej do niej licencji oraz będzie musiał zgodzić się z jej warunkami.

Zgodnie z wcześniejszym rozdziałem o naszej odpowiedzialności, my jako pośrednicy nie ponosimy odpowiedzialności za publikowane programy oraz za działania naszych użytkowników.

### **3.4.2. Sprawdzanie oprogramowania i nasza odpowiedzialność**

#### **Zagrożenie**

Istnieje ryzyko pojawienia się nieuczciwych osób, które będą chciały wykorzystać system dystrybucji w celu innym niż upublicznienie swojego programu. Nie trudno sobie wyobrazić, że osoba publikując program nie ma do tego prawa. W takim wypadku publikacja taka nie różni się niczym od tak popularnych w internecie pirackich wersji. Może się okazać, że opublikowany program zawiera szkodliwe oprogramowanie, bądź ma na celu wyłudzenie od nas cennych informacji. Innym razem, że przedstawiony przez autora opis programu oraz jego funkcjonalność obiegają od rzeczywistego stanu.

Wszystko to wymusza na nas konieczność posiadania mechanizmu, który weryfikowałby publikowane u nas oprogramowanie oraz minimalizowała potencjalne skutki tych zagrożeń zarówno dla nas jak i naszych użytkowników.

#### **Weryfikacja programów - definicja**

Weryfikacja oprogramowania jest jedną z bardziej złożonych dziedzin inżynierii oprogramowania, którego celem jest zapewnienie, tego że oprogramowanie w pełni spełnia wszystkie stawiane przed nim wymagania.

#### **Możliwe rozwiązania**

Najbardziej oczywistym rozwiązaniem jest wprowadzenie fazy przejściowej między zgłoszenie programu do publikacji a samą publikacją. Podczas tej fazy program przechodzi proces akceptacji przez administratora, który decyduje czy dany program nadaje się do publikacji czy też nie. Aplikacja jest sprawdzana pod kątem poprawności z opisem, czy nie stanowi zagrożenia oraz czy nie jest to piracka wersja popularnego oprogramowania.

Niestety takie rozwiązanie nie jest pozbawione wad. Przede wszystkim niesie ze sobą dość

duży koszt czasu i pracy administratora, który musiałby sprawdzać i rozpatrywać każde zgłoszenie publikacji. Istnieją też ograniczenia techniczne związane z możliwościami sprawdzenia czy dany program może okazać się zagrożeniem dla użytkowników. Administrator, może przeskanować je jednym z programów antywirusowych albo kosztem wydłużenia czasu przeprowadzić dłuższy polegający na użytkowaniu jej. Dość sporym problemem może okazać się kwestia sprawdzenia czy osoba publikująca program ma do tego prawo. Administrator w razie wątpliwości musiałby na własną rękę szukać kontaktu z potencjalnymi dysponentami praw do programu. Podczas kontaktu może dojść do trudności polegających na różnicach językowych np. w przypadku kontaktów z obcokrajowcami.

Skoro nie jesteśmy w stanie sprawdzić wszystkiego, to czy istnieje inna skuteczniejsza metoda weryfikacji oprogramowania? Za przykład takiego rozwiązania może posłużyć serwis „Allegro”. Takie rozwiązanie nie wymaga dużego nakładu czasu i pracy administratorów systemu a przy tym zapewnia wystarczający poziom bezpieczeństwa. Nie będzie konieczne ręczne sprawdzanie wszystkich publikowanych aplikacji. Każdy program, będzie posiadał średnią ocenę oraz komentarz wystawiony przez użytkowników, którzy go posiadają i używają. Ocena wraz z komentarzami daje innym potencjalnym użytkownikom znacznie pełniejszy obraz aplikacji niż sam jej opis umieszczany przez autora. Użytkownik ponadto może przejrzeć inne publikacje danego autora, żeby sprawdzić jego rzetelność. W razie napotkania jakichkolwiek nieprawidłowości mogą zgłosić swoje zastrzeżenia administratorowi, który rozpatrzy je i zastosuje stosowne środki. W połączeniu z wydłużonym czasem realizacji wypłat stanowi to wystarczającą obronę przed potencjalnymi próbami oszustw i wyłudzeń.

## **Wnioski**

Naszym zdaniem jest to najpraktyczniejsze i zarazem sprawdzone w praktyce rozwiązanie. Oczywiście nie jest ono pozbawione wad. Podobnie jak w „Allegro” istnieje problem nowych użytkowników, w naszym przypadku publikujących do których inni użytkownicy nie mają na początku zaufania. Nikt z nas nie lubi być królikiem doświadczalnym.

### **3.4.3. Nasza odpowiedzialność prawna**

Czy jako właściciele systemu dystrybucji ponosimy odpowiedzialność prawną za działania naszych użytkowników? Zarówno publikujących jak i korzystających z aplikacji. Do tego problemu przydatne okażą się zebrane w rozdziale 2.3. **Aspekty prawne** najważniejsze zasady obowiązujące w polskim systemie prawnym, dotyczące świadczenia usług w internecie.

Przede wszystkim chroni nas wyłączenie odpowiedzialności (podrozdział o wyłączeniu odpowiedzialności) za przechowywane dane jeśli nie udostępniając je nic nie wiemy o ich bezprawnym charakterze. Zapis ten chroni nas przed skutkami prawnymi za udostępnianie aplikacji osób, które nie mają do nich praw oraz za skutki wywołane np. przez szkodliwe oprogramowanie. Oczywiście przepis ten nakłada na nas obowiązek niezwłocznego usunięcia aplikacji w razie otrzymania urzędowego zawiadomienia lub uzyskania wiarygodnej wiadomości. Niestety pojęcie niezwłoczny i wiarygodny istnieje w polskim prawie jako pojęcie nieostre. Oznacza to, że każdy może je interpretować po swojemu.

Polskie prawo nie wymusza na nas obowiązku monitorowania naszych publikacji. Zgodnie z podrozdziałem „Brak konieczności monitoringu danych”, my jako podmiot świadczący usługi kwalifikujące się jako usługi hostingowe, nie jesteśmy zobowiązani do sprawdzania przechowywanych, przekazywanych lub udostępnianych danych.

Dodatkową ochronę zapewnia nam regulamin, w którym zastrzegamy sobie, że nie ponosimy odpowiedzialności za publikowane u nas oprogramowanie.

### **3.4.4. Regulamin**

Regulamin jest tym dla naszego serwisu czym licencja dla oprogramowania. Podobnie jak licencje, regulamin określa jak i na jakich warunkach można z niego korzystać. Definicje regulaminu została zawarta w podrozdziale 2.3. Aspekty prawne. W tej części przedstawimy zapisy, które naszym zdaniem powinny znaleźć się w treści regulaminu.

#### **3.4.4.1. Definicje**

Publikacja - udostępniany w AppShare program.

Użytkownik – osoba spełniająca warunki regulaminu i dokonała rejestracji w AppShare.

Ukończenie rejestracji warunkuje utworzenie konta za pomocą, którego uzyskał dostęp do usług świadczonych przez AppShare na zasadach określonych w regulaminie.

Publikujący – użytkownik, który jest autorem publikacji w AppShare.

Konto – prowadzone przez AppShare pod unikalną nazwą „loginem” jest zbiorem zasobów, w

których gromadzone są dane użytkowników oraz informacje o ich działalności

Rejestracja – procedura zakładania konta.

#### **3.4.4.2. Preambuła**

Treść niniejszego regulaminu stanowi porozumienie pomiędzy tobą a AppShare, określającą warunki, które określają sposób korzystania z usług AppShare. Porozumienie to, wraz ze wszystkimi zmianami stanowi umowę pomiędzy tobą a AppShare. Jeśli zgadzasz się z jej treścią, wciśnij przycisk „zgadzam się”. Jeśli nie zgadzasz się, nie klikaj go i nie używaj serwisu. Musisz zaakceptować i przestrzegać warunki zawarte w niniejszym regulaminie.

#### **3.4.4.3. Treść**

##### **Ogólne**

*AppShare zastrzega sobie prawo do usuwania i blokowania kont, publikacji co do których istnieje podejrzenie łamania warunków regulaminu oraz ogólnie obowiązującego prawa.*

##### **Prawa autorskie**

*Kopiowanie i reprodukcja programów bez wcześniejszej, pisemnej zgody jest zabronione.*

*Zakazuje się rozpowszechnianie wszelkiego piractwa w każdej formie.*

##### **Wyłączenie z odpowiedzialności**

*AppShare nie ponosi żadnej odpowiedzialności za publikowane programy oraz za skutki ich użytkowania.*

##### **Znaki zastrzeżone**

*Wszystkie nazwy, znaki i loga użyte w publikacjach AppShare mają charakter wyłącznie informacyjny*

##### **Ochrona danych osobowych**

*AppShare przestrzega prawa do prywatności wszystkich użytkowników, nie ujawnia oraz nie przekazuje żadnych danych osobowych osobom trzecim.*

## **Komentarze i oceny**

*Każdy użytkownik ma prawo do komentowania i oceniania opublikowanych w AppShare programów.*

*Zakazuje się umieszczania treści sprzecznych z polskim prawem oraz naruszających dobra innych osób, w szczególności: treści wulgarnych, obraźliwych i nieprawdziwych, dyskryminujących wszelkiego rodzaju grupy rasowe i inne, faszystowskich czy propagujących przemoc etc.*

## 4. Opis narzędzi zastosowanych w pracy

Prototyp został zaimplementowany przy użyciu platformy Microsoft .NET Framework 4.0 wykorzystując język C#.

Autorzy podczas tworzenia niniejszej pracy posłużyli się wieloma „narzędziami” oraz technologiami (technologies), których większość chcieliby przedstawić w kolejnych podrozdziałach.

### 4.1. Środowisko programistyczne – Microsoft Visual Studio 2010

Jako środowisko programistyczne zostało wybrane Microsoft Visual Studio 2010 ([Rysunek 10: Visual Studio 2010 - Edytor kodu](#)) z powodu najlepszej integracji z rozwiązaniami Microsoftu.

Można znaleźć wiele wersji Microsoft Visual Studio 2010:

- Professional
- Premium
- Ultimate
- Test Professional

Wersja Professional i Premium jest przeznaczona dla osób pracujących samodzielnie lub w małych zespołach. Najtańsza wersja Professional została pozbawiona takich elementów jak profiler, statyczna analiza kodu, tworzenia schematu bazy danych z poziomu IDE.

Zdaniem autora jej odpowiednikiem może być wersja Standard z poprzedniej wersji.

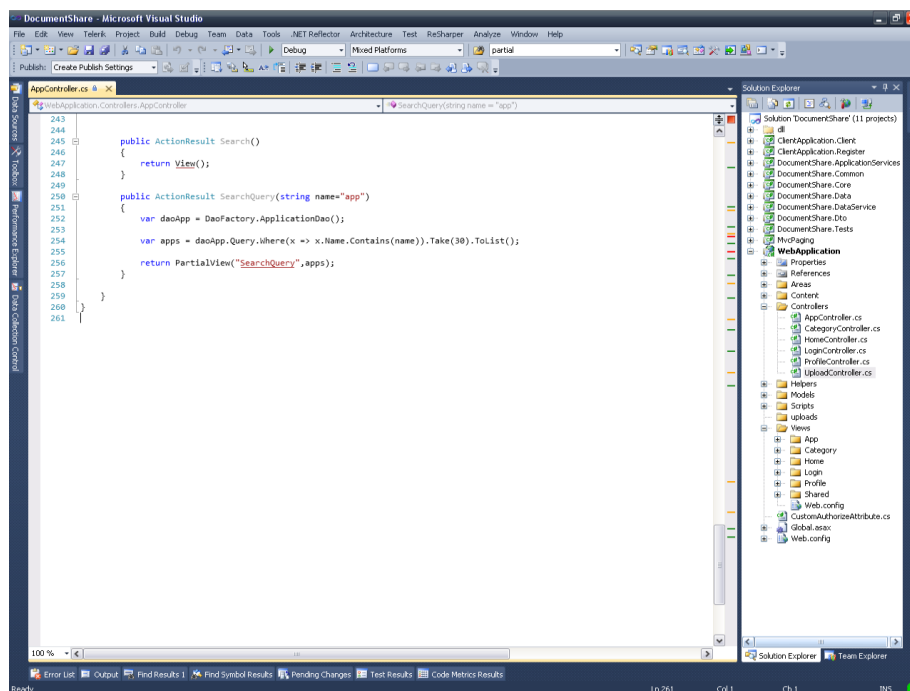
Wersja Ultimate zawiera wszystkie cechy uboższych wersji oraz wersji Test Professional, która jest przeznaczona głównie dla testowania aplikacji, wspierania automatyzacji testów, w tym testów interfejsu użytkownika etc.

Porównując „niższe” wersje z wersją Ultimate można zauważyć przede wszystkim brak IntelliTrace, wsparcie dla diagramów UML, moduł do tworzenia oraz przeprowadzania testów obciążeniowych oraz wydajności (Web Performance Test)

Inną cechą wersji Ultimate jest inaczej jak to było w poprzednich wersjach Visual Studio (poza wersją Team Edition) pełne wsparcie dla Team Foundation Server, który był również wykorzystywany podczas trwania projektu.

Warte wspomnienia jest również to, że podobnie jak w poprzednich wersjach została opublikowana darmowa wersja Visual Studio 2010 Express.

Autorzy pracy długi czas wykorzystywali wersję beta2 oraz RC, wraz z wprowadzeniem wydania finalnego Visual Studio 2010 (zdaniem autorów) wprowadziło wiele usprawnień w porównaniu z poprzednią wersją.



Rysunek 10: Visual Studio 2010 - Edytor kodu



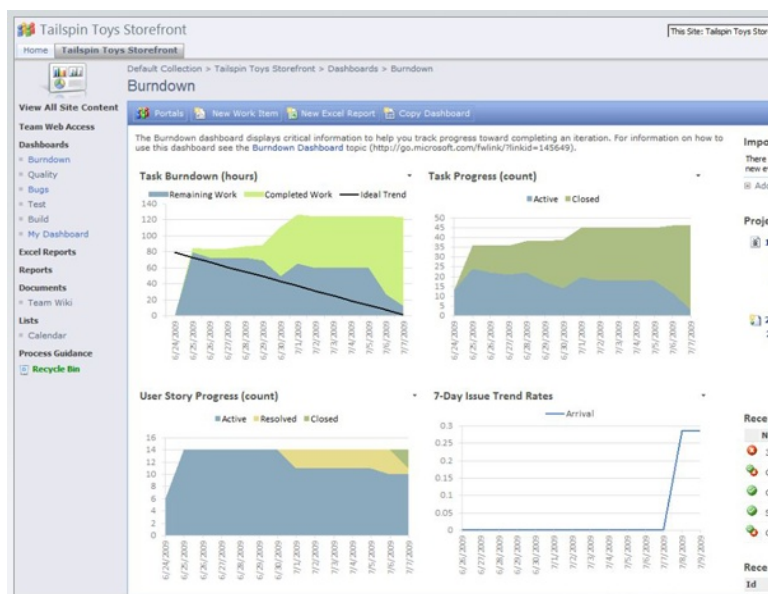
## 4.2. VISUAL STUDIO TEAM FOUNDATION SERVER 2010

Podczas procesu tworzenia autorzy mieli dostęp do Team Foundation Server 2010.

Team Foundation Server docelowo poza samą funkcją repozytorium kodu ma służyć jako centralny system do zarządzania projektem tzn. podziałem prac i monitorowania ich postępu.

Począwszy od wersji 2010 dla ułatwienia zarządzanie projektem zostały wprowadzone poprawione szablony przystosowane do zwinnych metodyk (SCRUM).

Można teraz stworzyć opowieści użytkownika (user story) a następnie konkretnie zadania, które następnie można przydzielić konkretnej osobie w konkretnej iteracji. Dodatkowo jest możliwość zarządzania większością kluczowych spraw dla projektu takich jak iteracje w Excelu. Bardzo interesująca funkcją jest generowanie wykresów wypalania ([Rysunek 11: Team Foundation Server 2010 Burndown chart](#)), które przy wykorzystaniu zwinnych metodyk pozwalają jako tako mieć wyobrażenie o przewidywanym końcu projektu oraz bieżących postępach prac.



*Rysunek 11: Team Foundation Server 2010 Burndown chart*

Podsumowując zdaniem autorów dzięki takim udoskonaleniom korzystanie z TFS 2010 stało się prostsze niż w poprzedniej wersji.

Oczywiście TFS to również możliwość stworzenia strony projektowej przy użyciu technologii Windows SharePoint Services oraz raportów przy użyciu SQL Server Reporting Services.

Niestety wadą jest ogromne zapotrzebowanie na zasoby komputera (a przynajmniej w używanej przez autorów wersji beta2), użycie wszystkich funkcji takich jak raportowanie postępów prac wymaga użycia „pełnej” wersji Sql Servera co przekłada się na ogromne zapotrzebowanie na pamięć serwera.

Z tych względów autorom mającym dostęp do serwera mającego jedynie 1GB Ram nie udało się wykorzystać całej funkcjonalności serwera.

### **4.3. Microsoft Sql Server 2008**

Microsoft Sql Server jest systemem zarządzania (relacyjną) bazą danych, komercyjnie rozpowszechniany przez Microsoft.

MS SQL jest głównym produktem bazodanowym Microsoft. Jest to relacyjna baza danych typu klient-serwer, jego główną cechą jest to, że jako język zapytań wykorzystywany jest T-SQL (Transact-SQL), który jest rozwinięciem ANSI SQL.

Sql Server od czasu premiery wersji 2005 opublikowanej w 2005 r posiada wbudowany język T-SQL, który podobnie jak PL/SQL (Oracle) umożliwia pisanie złożonych procedur składowanych, triggerów etc. dzięki którym jest możliwość wykonywania bardziej skomplikowanych operacji niż SELECT, INSERT, UPDATE, DELETE oraz (w przypadku procedur składowanych) jest wyższa wydajność.

Sql Server 2008 ([Rysunek 12: Komponenty Sql Server 2008](#)) dostarcza skutecznego oraz skalowalnego rozwiązania dostępu do danych w każdym miejscu i czasie, dostarczając również bogaty zestaw usług pozwalający na nowe sposoby dostępu i wykorzystania danych bez względu gdzie są przechowywane.



*Rysunek 12: Komponenty Sql Server 2008*

To również doskonała integracja z resztą produktów Microsoft, włączając w to Microsoft Office, z tych właśnie powodów autorzy przy wyborze relacyjnej bazy danych wybrali Sql Server 2008.

Poza wersjami Enterprise i Standard Microsoft opublikował również darmową wersję MS SQL Express.

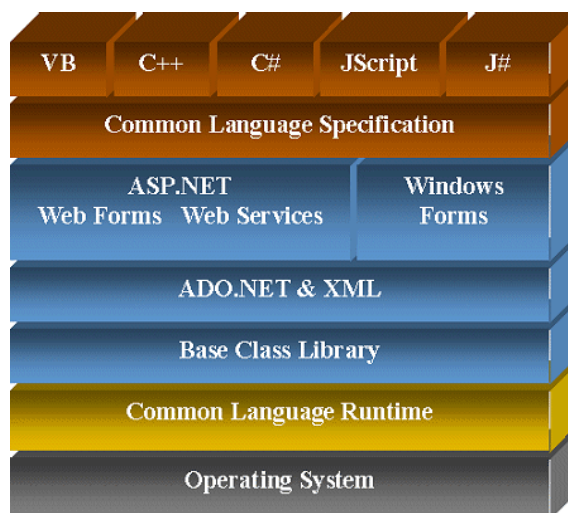
Jest ona przeznaczona dla mniejszych rozwiązań oraz cytując dosłownie broszurę Sql Server 2008 dla programistów w celach edukacyjnych dla lepszego poznania produktów Microsoft.

W odróżnieniu od pozostałych „korporacyjnych” wersji, edycja Express została pozbawiona niektórych zaawansowanych funkcjonalności np. Analysis Services, jednak wersja Express występuje w kilku rodzajach różniących się między innymi obecnością Server 2008 Management Studio Express oraz Reporting Services.

## 4.4. .NET Framework 4

.Net Framework jest platformą programistyczną stworzoną przez firmę Microsoft.

W jego skład wchodzi biblioteki i środowisko uruchomieniowe (wirtualna maszyna), która nadzoruje wykonywanie kodu.



*Rysunek 13: Komponenty .Net Framework*

Można tutaj dostrzec pewne podobieństwa do Javy, jednak .Net Framework obsługuje wiele języków programowania w tym C#, Visual Basic.Net, C++, J#.

Przy pomocy .Net Framework można stworzyć aplikacje konsolowe, desktopowe, webowe czy usługi Windows.

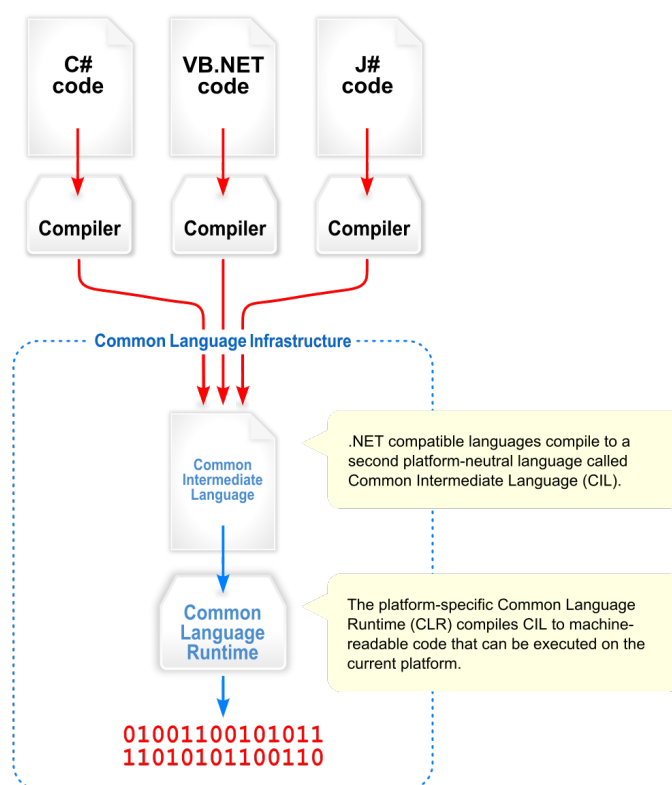
Na rysunku 13 ([Komponenty .Net Framework](#)) zostały przedstawione główne „komponenty” wchodzące w skład platformy .Net.

.Net Framework powstał jak powszechnie uważa się jako odpowiedź na rosnącą popularność Javy, starając jednocześnie wyeliminować wszelkie "niedociągnięcia" powstałe przy rozwoju pierwszych wersji Javy, tak aby platforma .Net była jak najbardziej przyjazna oraz oferowała jak najwyższą wydajność.

Podobnie jak w przypadku Javy kod źródłowy jest kompilowany do kody pośredniego CIL (Common Language Infrastructure lub dawniej MSIL).

Następnie kod pośredni przez CLR (Common Language Runtime) jest kompilowany przed wykonaniem do kodu maszynowego (Rysunek 14: Poglądowy diagram funkcjonalny Common Language Infrastructure (CLI)), w czasie wykonania wykorzystując JIT lub wcześniej wykorzystując Native Image Generator (NGEN).

Główną zaletą NGEN jest przyspieszenie pierwszego wykonania poprzez wyeliminowanie dodatkowego narzutu kompilatora JIT.



Rysunek 14: Poglądowy diagram funkcjonalny  
Common Language Infrastructure (CLI)

Kod uruchamiany w zarządzalnym środowisku uruchomieniowym zyskuje dodatkowe zalety takie jak: „wielu-języczna” integracja (np. C# z VB.NET), obsługa wyjątków, zarządzanie pamięcią czy bezpieczeństwo Code Access Security ( w najnowszej wersji 4.0 polityki bezpieczeństwa CAS zostały uznany za zbyt skomplikowany i zostały oznaczone jako „przestarzałe”).

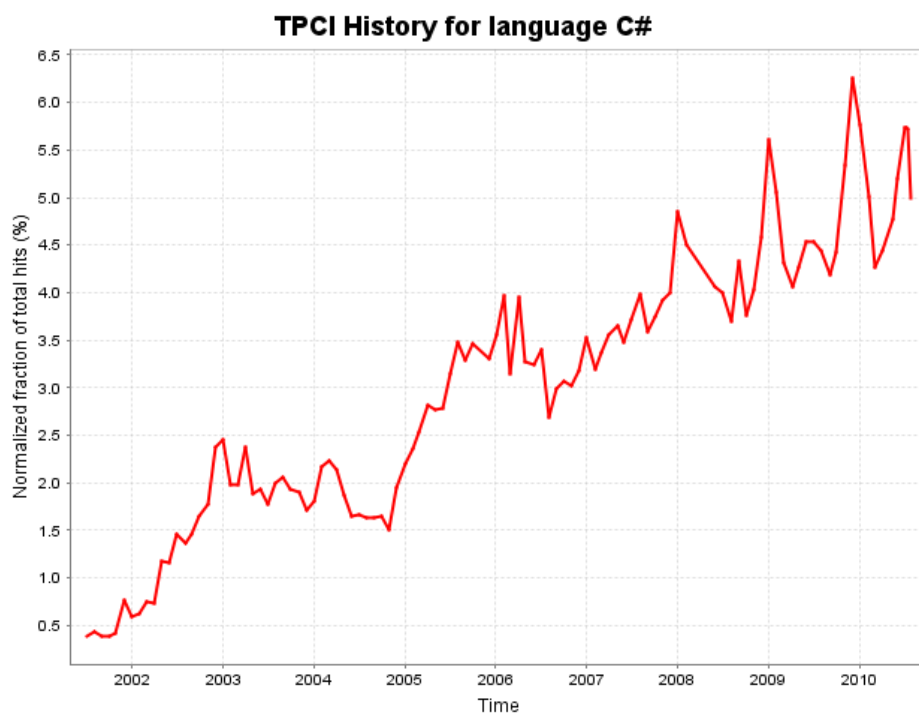
Następnym elementem .Net Framework jest Base Class Library (BCL) ([Rysunek 15: Główne przestrzenie nazw Base Class Library](#)).

Bcl zawiera zestaw bibliotek do obsługi wejścia/wyjścia, komunikacji sieciowej, dostępu do danych, łączności z bazami danych, globalizacji itd.



*Rysunek 15: Główne przestrzenie nazw Base Class Library*

Język C# jest dosyć popularny, a jego udział wśród innych języków statystycznie ciągle wzrasta ([Rysunek 16: Wykres popularności c#](#))



*Rysunek 16: Wykres popularności c#*

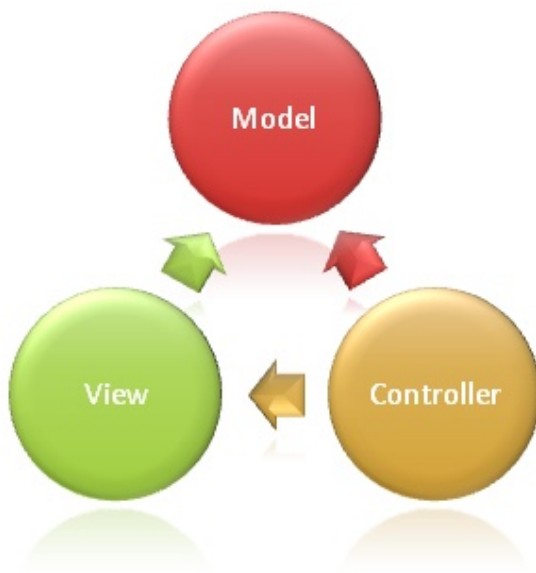
## 4.5. ASP .NET MVC2

Asp .Net MVC 2 jest platformą Microsoft do tworzenia aplikacji Web. implementującą wzorzec Model View Controller ([Rysunek 17: Model View Controller](#)).

Aplikacje web stanowią dzisiaj (jak można zaobserwować) przewagę aplikacji, ten typ aplikacji ma jedną zaletę – uruchamiany jest na serwerze, gdzie znajdują się cała logika aplikacji. Od strony klienta niezbędna jest jedynie przeglądarka internetowa (na której wyświetlany jest interfejs użytkownika – Cienki klient).

Podejście to (poza oczywistym brakiem potrzeby instalowania aplikacji lokalnie) ma wiele zalet od łatwości aktualizacji (wystarczy zaktualizować jedną lokalizację na serwerze) do ochrony przed nieuprawnioną modyfikacją (trudno sobie wyobrazić taki przypadek jak np. wytworzenie cracka do aplikacji internetowej lub rozpowszechnianie takiej aplikacji w sieciach p2p).

Ze względu na konieczność przesłania za każdym razem interfejsu użytkownika aplikacje web nie nadają się do tworzenia dynamicznych i multimedialnych aplikacji jak na przykład dynamiczne gry (choć istnieją przykłady gier strategicznych).



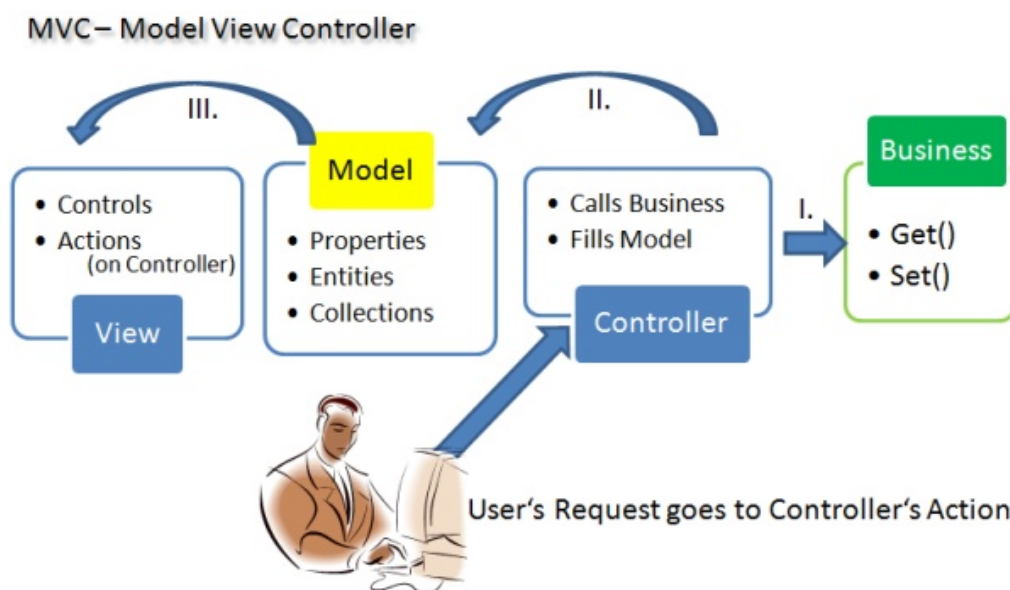
*Rysunek 17: Model View Controller*

Asp.Net Mvc wywodzi się poniekąd z „klasycznego” Asp.Net Web Forms, którego jednym

z głównych założeń było umożliwienie tworzenia aplikacji web osobom które nie znają dokładnie zagadnień związanym z protokołem HTTP, językiem HTML, Javascript, problemem stanowości itp.

Rozwiązanie było adresowane dla osób mających dotychczas styczność takimi językami jak Visual Basic (gdzie wszystkie kontrolki można było przeciągnąć na formę i w paru kliknięciach przy minimalnej ilości kodu stworzyć szybko prostą aplikację) gdzie tworzenie aplikacji odbywało się w dużej mierze w wizualny sposób.

Z różnych powodów pośród „społeczności” koncepcja Web Forms zaczęła się rozwijać, zaczęto stosować wzorzec MVP w końcu na początku 2007 roku została zaprezentowana wersja CTP ASP.NET MVC.



*Rysunek 18: MVC Koncepcja*

Stosowanie Mvc zmusza do zmiany myślenia o aplikacji web, jednak wbrew pozorom sama koncepcja jest dużo prostsza w porównaniu z Web Forms.

Podstawową koncepcją jest odmienny sposób tworzenia adresów (Url routing [Rysunek 18: MVC Koncepcja](#)).



Zapytania przekierowywane są do kontrolera (przy dopasowaniu przez „listę routingu”), który to następnie po wykonaniu operacji na modelu danych przekazuje odpowiedni model do widoku, który renderuje ostatecznie odpowiedź (html) zwracaną do użytkownika.

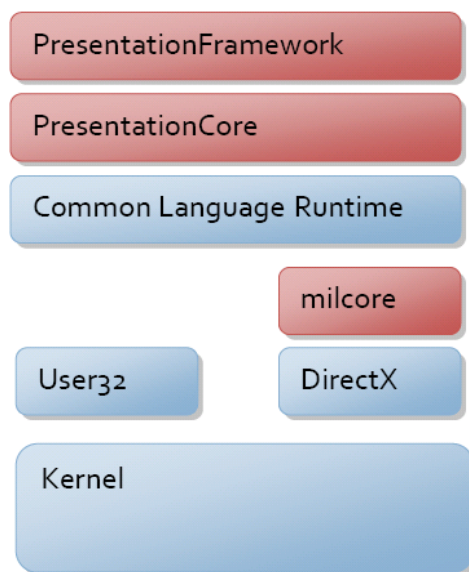
Zdaniem autorów Mvc jako alternatywa dla Web Forms jest lepszą platformą, umożliwia całkowitą kontrolę nad generowanym Html, umożliwia tworzenie przyjaznych użytkownikom adresów bez konieczności stosowania „ręcznie” technik Url Rewriting, w niektórych momentach upraszcza tworzenie aplikacji przez brak ViewState oraz całego cyklu życia typowego dla Web Forms i istniejących ograniczeń z tym związanych (np.inny cykl życia dla stron inny dla kontroltek użytkownika, istniejące problemy z ViewState dla dynamicznie tworzonych kontroltek przy bardziej rozbudowanych stronach zawierających dużą liczbę zagnieżdżonych kontroltek użytkownika ), wyraźne rozdzielanie aspektów funkcjonalności pomiędzy poszczególne komponenty aplikacji (Separation of Concerns).

## **4.6. Windows Presentation Foundation**

Windows Presentation Foundation (WPF, nazwa kodowa Avalon) jest platformą (silnik graficzny i API bazujące o .NET 3.0) umożliwiającą tworzenie bogatych i interaktywnych interfejsów użytkownika. WPF stara się zapewnić rozdzielenie logiki biznesowej od interfejsu użytkownika.

WPF w odróżnieniu od np. Windows Forms do definicji elementów interfejsu użytkownika wykorzystuje XML (XAML), silnik graficzny wykorzystuje grafikę wektorową, tworzoną z użyciem akceleracji 3D (Direct3D). WPF posiada oczywiście więcej zalet w porównaniu z Windows Forms, jednak podstawowa to wspomniany już XAML oraz narzędzia takie jak pakiet Expression ułatwiające prace nad interfejsem użytkownika innym osobom niż programiści – np. grafikom.

Windows Presentation Foundation jest dostępny we wszystkich systemach Microsoft Windows począwszy od Windows XP (SP2)



*Rysunek 19: Główne komponenty WPF*

Główne komponenty WPF zostały przedstawione na diagramie w kolorze czerwonym ([Rysunek 19: Główne komponenty WPF](#)).

Jak widać całe "renderowanie" odbywa się poprzez silnik DirectX, pozwalając na częściowe obciążenie procesora.

## 4.7. Windows Communication Foundation (WCF)

WCF (nazwa kodowa Indigo) jest platformą pozwalającą na budowanie aplikacji zorientowanych aspektowo. Została ona wprowadzona wraz z wydaniem .Net Framework 3.0

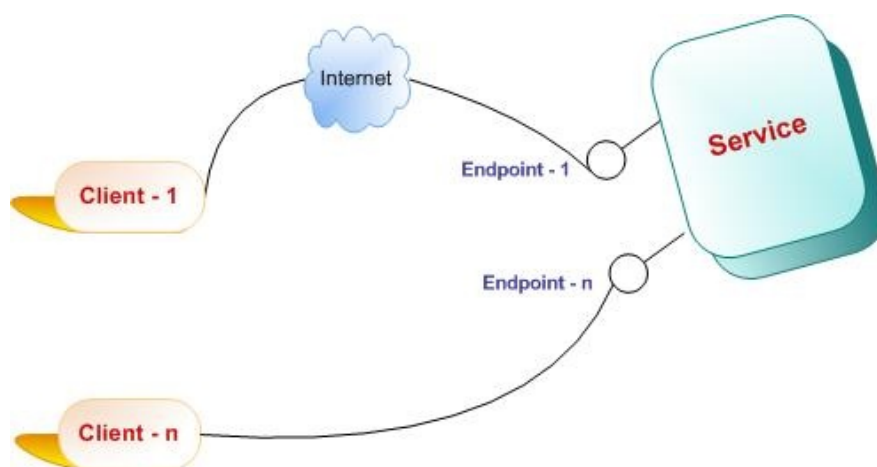
WCF upraszcza tworzenie aplikacji, integrując wszelkie dotychczasowe technologie Microsoft ([Rysunek 20: Poglądowe spojrzenie na WCF](#)) służące do komunikacji takie jak: .NET Remoting, MSMQ, Webserwisy (w tym standardy takie jak WS-Security) etc.



*Rysunek 20: Poglądowe spojrzenie na WCF*

Komunikacja pomiędzy usługami może odbywać się tradycyjnie poprzez HTTP lub Tcp a nawet z wykorzystaniem komunikacji międzyprocesowej lub nawet potoki (named pipes) z wykorzystaniem SOAP, co pozwala zachować pewien stopień interoperacyjności (interoperability) lub w przypadku gdy komunikacja odbywa się w obrębie tej samej technologii może być wykorzystana postać binarną.

Komunikacja w WCF odbywa się w modelu klient-serwer. Aplikacja kliencka ma bezpośredni dostęp do usługi przez punkt końcowy (end point) udostępniony przez usługę. Sam punkt końcowy jest niczym innym jak określoną lokalizacją, która może odbierać i wysyłać wiadomości, każda usługa może posiadać wiele punktów końcowych (Rysunek 21: WCF komunikacja klient-serwer).



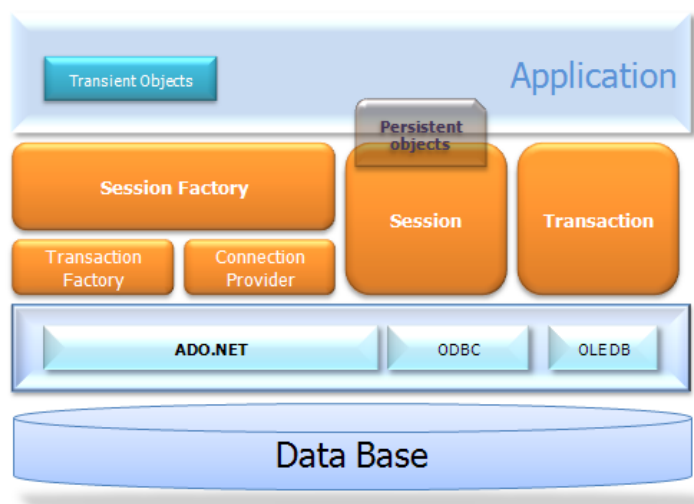
*Rysunek 21: WCF komunikacja klient-serwer*

## 4.8. NHibernate

NHibernate jest biblioteką ORM (Object Relational Mapper) dla platformy .NET.

Jak nazwa wskazuje umożliwia mapowanie obiektowego modelu domenowego do tradycyjnego modelu relacyjnego. NHibernate jest portem popularnego ORM Hibernate, oba projekty są open source udostępniane na GNU LGPL.

Jak twierdzą autorzy projektu NHibernate jest intensywnie rozwijany (szczególnie ostatnio) oraz z powodzeniem wykorzystywany w tysiącach projektów.



Rysunek 22: Architektura NHibernate

Podstawowym celem (zaletą) jest odciążenie programisty od „ręcznej” implementacji metod dostępu do relacyjnych danych (Create, Retrieve, Update, Delete – Rysunek 22: Architektura NHibernate).

Dużą zaletą Nhibernate jest jego dosyć proste użytkowanie, jedyne co musi on zrobić to określić sposób mapowania poszczególnych kolumn z tabeli na odpowiednie pola klas oraz na zdefiniowanie mapowania relacji (relacje klucza obcego pomiędzy tabelami w bazie).

Nhibernate do funkcjonowania używa obiektów POCO (Plain Old CLR Objects), które służą jedynie przechowywaniu danych pobranych z modelu relacyjnego, a nie jego przetwarzaniu -nie zawierają w sobie żadnej funkcjonalności ani nie odwołują się w żaden sposób do źródeł danych.

Takie podejście jest główną zaletą wyróżniającą NHibernate na tle konkurencji.

Sama definicja mapowania może zostać przeprowadzona standardowo za pomocą konfiguracji w plikach XML, lub za pomocą Fluent NHibernate który pozwala uwolnić się od plików XML.

NHibernate umożliwia wiele różnych sposobów dostępu do danych, posiada wiele providerów do różnych baz danych (autor wykorzystywał go z powodzeniem zarówno korzystając z MySQL jak i MS SQL) jak również wiele sposobów tworzenia zapytań.

Poza możliwością tworzenia zapytań w czystym SQLu w NHibernate został wbudowany język HQL (Hibernate Query Language) jest to język obiektowych zapytań mający (zdaniem autorów) ułatwić prace przy skomplikowanych zapytaniach.

Jednak najbardziej istotne zdaniem autorów jest począwszy od wersji 2.1 częściowe implementacja providera LINQ, w najnowszej rozwojowej wersji 3 implementacja dla LINQ została usprawniona.

Autorzy są zdania, że NHibernate jest najlepszym wyborem jeśli chodzi o ORM umożliwiającym w miarę prostą bez zbędnych komplikacji obsługę większości scenariuszy dostępu do danych takich jak: pobieranie kolekcji i pojedynczych wartości, aktualizacja, usuwanie, wyszukiwanie. W celu poprawienia wydajności poza samym stosowaniem tzw. lazy-loading jest możliwość stosowania cachowania (standardowo istnieją 2 poziomy cache) oraz dowolna i elastyczna jego konfiguracja (np. można wybrać standardowy provider Asp.Net lub np. Velocity)

## 5. Prototyp aplikacji AppShare

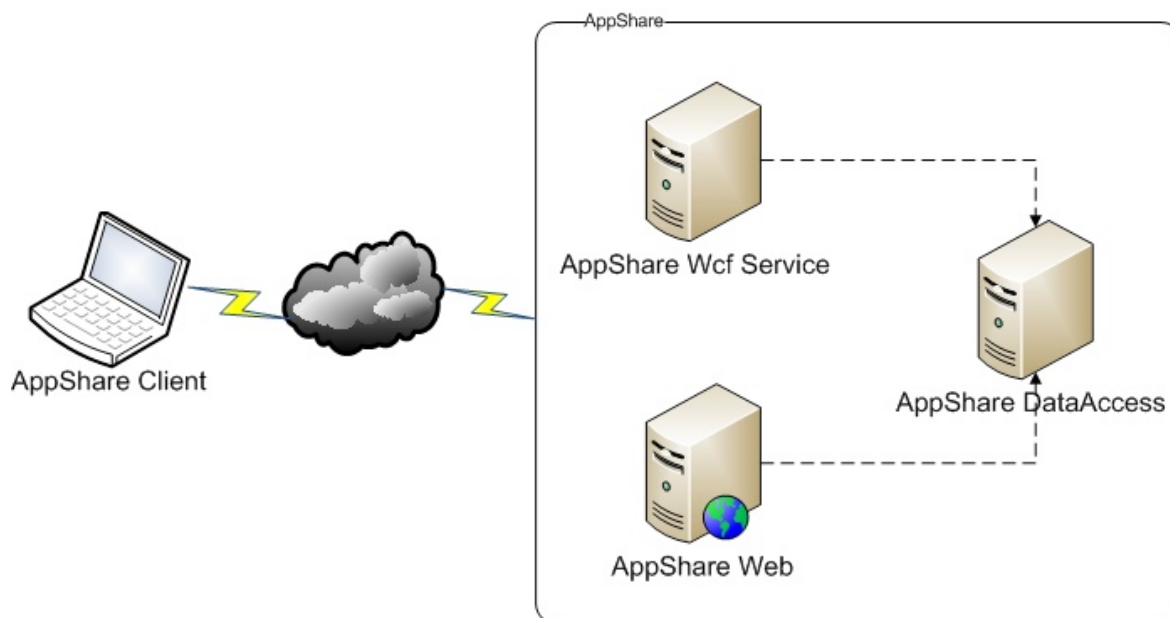
W niniejszym rozdziale zostanie krótko przedstawiony prototyp proponowanego systemu.

Całość została zaimplementowana w języku C# (platforma .NET 4.0) z wykorzystaniem Visual Studio 2010 oraz Sql Server 2008 Express.

### 5.1. Opis architektury systemu

System Elektronicznej dystrybucji oprogramowania składa się z 2 głównych części:

- Części klienckiej – „cienkiego klienta”
- Części serwerowej – wliczając w to:
  - Aplikację internetową
  - Usługę WCF (umożliwiająca dostęp aplikacji klienckiej do systemu)
  - „Infrastrukturę” zapewniającą m.in. dostęp do danych, model domenowy, mapowanie obiektowo-relacyjne



Rysunek 23: Uproszczona struktura prototypu

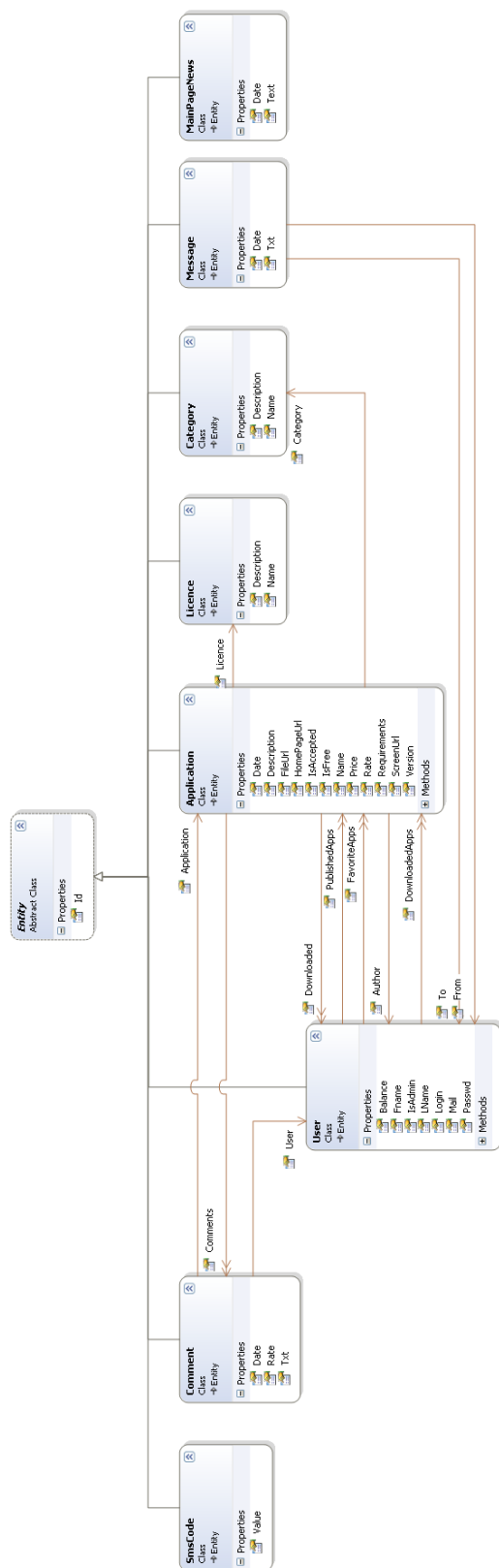
Na diagramie powyżej (Rysunek 23: Uproszczona struktura prototypu) została przedstawiona w uproszczeniu główna struktura systemu.

W jej skład wchodzi:

- AppShare Client jest aplikacją kliencką (cienki klient), który do swojego funkcjonowania wymaga dostępu do usługi WCF
- AppShare Wcf Service jest usługą umożliwiającą dostęp aplikacji klienckiej do systemu, implementuje ona zestaw „funkcjonalności” zdefiniowany w interfejsie.
- Komponent DataAccess poza samą funkcją dostępu do danych, zawiera również warstwę logiki
- AppShare Web to aplikacja internetowa, jest ona zależna od „Podsystemu” DataAccess (w szczególności wymaga modułu zarządzania sesją Nhibernate)

Architektura aplikacji znajdująca się po stronie serwera została stworzona w oparciu o postulaty zamieszczone w artykule „NHibernate Best Practices with ASP.NET”.

[Rysunek 24](#) przedstawia model domenowy stworzony na potrzeby prototypu, umożliwia on spełnienie niemal wszystkich wymagań biznesowych istotnych z punktu widzenia prototypu.



Rysunek 24: Model domenowy



Na potrzeby projektu do mapowania obiektowo-relacyjnego została użyta biblioteka Nhibernate, natomiast mapowania są określone w plikach xml (Rysunek 25: Przykład pliku xml z mapowaniem).

```
<?xml version="1.0" encoding="utf-8" ?>
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2" >

  <class name="DocumentShare.Core.Domain.Comment, DocumentShare.Core" table="[Comment]">

    <id name="Id" column="CommentId" type="int">
      <generator class="identity" />
    </id>

    <property name="Txt" type="String">
      <column name="Txt"/>
    </property>

    <property name="Date" type="DateTime">
      <column name="Date"/>
    </property>

    <property name="Rate" type="int">
      <column name="Rate"/>
    </property>

    <many-to-one name="User" class="DocumentShare.Core.Domain.User" column="UserId"/>
    <many-to-one name="Application" class="DocumentShare.Core.Domain.Application"
      column="ApplicationId"/>

  </class>
</hibernate-mapping>
```

*Rysunek 25: Przykład pliku xml z mapowaniem*

Pomimo „ręcznego” sposobu tworzenia mapowań, schemat bazy danych generowany jest automatycznie (Rysunek 26: Generowanie schematu bazy danych).

```
Configuration cfg = new Configuration().Configure();

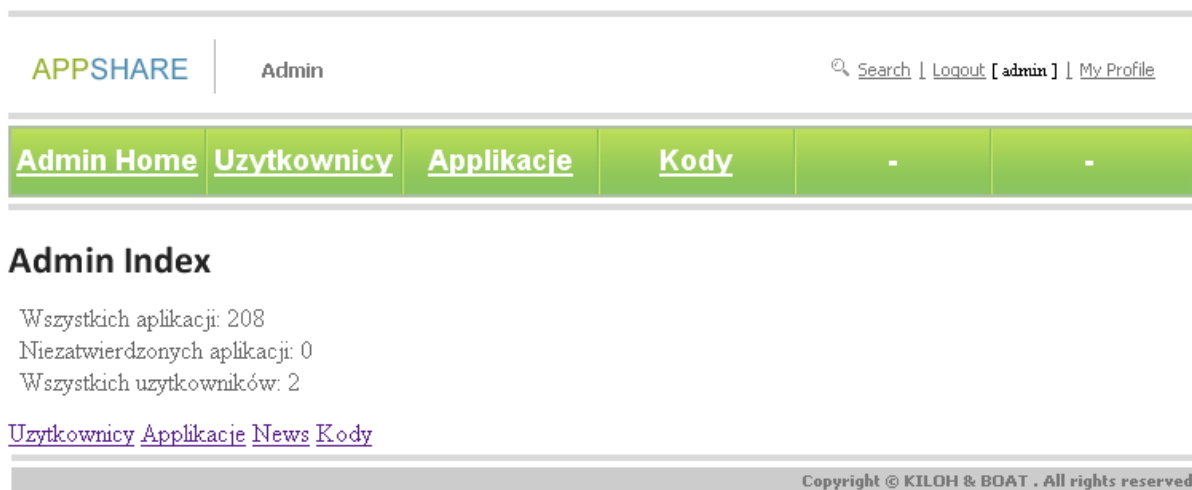
SessionFactory sessionFactory = cfg.BuildSessionFactory();
session = sessionFactory.OpenSession();
new SchemaExport(cfg).Execute(false, true, false);
```

*Rysunek 26: Generowanie schematu bazy danych*

## 5.2. Aplikacja internetowa

Aplikacja internetowa jest jednym ze sposobów dostępu do systemu. Składa się ona z dwóch głównych obszarów: przeznaczonego dla użytkowników oraz administracyjnego.

### **część administracyjna**



*Rysunek 27: Aplikacja internetowa - strona administracyjna*

Na rysunku 27 przedstawiony został panel administracyjny portalu, umożliwia on zarządzanie

- Użytkownikami
- Aplikacjami
- Najnowszymi wiadomościami wyświetlanymi na stronie głównej
- Systemem płatności sms (w tym przypadku kodami)

Rysunki 28, 29, 30 przedstawiają przykłady zarządzania kolejno: aplikacjami, użytkownikami oraz wiadomościami głównymi w części administracyjnej.

APPSHARE

Admin

Search | Logout: [admin] | My Profile

Admin Home

Uzytkownicy

Applikacje

Kody

-

-

### Edit

EDIT APPLICATION DETAILS

|              |                                                   |
|--------------|---------------------------------------------------|
| Name         | <input type="text" value="IrfanView"/>            |
| Description  | <input type="text" value="small graphic viewer"/> |
| Date         | <input type="text" value="2010-08-10 14:38:11"/>  |
| Version      | <input type="text" value="4"/>                    |
| Category     | <input type="text" value="Programy"/>             |
| Price        | <input type="text" value="0"/>                    |
| ScreenUrl    | <input type="text"/>                              |
| Home Page    | <input type="text" value="www"/>                  |
| File         | <input type="text"/>                              |
| Requirements | <input type="text"/>                              |
| IsFree?      | <input checked="" type="checkbox"/>               |
| IsAccepted?  | <input checked="" type="checkbox"/>               |

Save

[Back to List](#)

Rysunek 28: Przykład zarządzania aplikacjami

APPSHARE
Admin
Search | Logout [ admin ] | My Profile

Admin Home
Użytkownicy
Applikacje
Kody
-
-

### Create User

|           |                                |
|-----------|--------------------------------|
| Login     | <input type="text"/>           |
| Password  | <input type="password"/>       |
| Name      | <input type="text"/>           |
| Last Name | <input type="text"/>           |
| Mail      | <input type="text"/>           |
| Admin?    | <input type="checkbox"/>       |
| Balance   | <input type="text" value="0"/> |

Create

[Back to List](#)

Copyright © KILOH & BOAT . All rights reserved.

Rysunek 29: Przykład zarządzania użytkownikami

APPSHARE
Admin
Search | Logout [ admin ] | My Profile

Admin Home
Użytkownicy
Applikacje
Kody
-
-

### Delete News

Are you sure you want to delete this?

DELETE?

|         |                                                            |
|---------|------------------------------------------------------------|
| Id      | 3                                                          |
| Message | Miedzy godzinami 1-3 w nocy płatności sms mogą nie działać |

Yes
No

Copyright © KILOH & BOAT . All rights reserved.

Rysunek 30: Przykład zarządzania wiadomościami

Rysunek 31 przedstawia zarządzanie kodami sms.

Integracja z płatnościami smsowymi została przeprowadzona w dosyć prosty ale elastyczny sposób, podczas rejestracji u dostawcy płatności generowana jest lista kodów (zazwyczaj kilkaset kodów), którą można wprost skopiować do systemu (Rysunek 32: Zarządzanie kodami sms -kontynuacja).

Zarządzanie kodami sms wykorzystuje Ajax (użyta została biblioteka jQuery z ewentualnymi dodatkami).

Rysunek 31: Zarządzanie kodami sms

Rysunek 32: Zarządzanie kodami sms -kontynuacja

### ***część dostępna dla użytkownika***

Zarządzanie własnym profilem (Rysunek 33), w tym lista pobranych oraz opublikowanych aplikacji, publikacja własnej aplikacji, zmiana hasła (Rysunek 36), edycja swoich danych (Rysunek 34) oraz uzupełnienie punktów (Rysunek 35).

Oczywiście w przypadku wprowadzenia nieprawidłowego kodu, lub innych danych (np. hasła) pojawia się stosowna informacja o błędnie wprowadzonych danych.

The screenshot shows the user profile page for 'stefan' on the APPSHARE website. The page has a green navigation bar with links: Home, Aplikacje, Kategorie, Moje aplikacje, O nas, and Kontakt. The main content area is titled 'Twój profil, stefan'. It features a table with user statistics and a sidebar with links to manage the profile.

| Login stefan           |   |
|------------------------|---|
| Dostępne punkty        | 0 |
| Pobrane aplikacje      | 0 |
| Opublikowane aplikacje | 0 |

[Więcej](#) [Edytuj](#)

[MOJE APLIKACJE](#)  
[MOJE PUBLIKACJE](#)  
[MÓJ PROFIL](#)  
[OPUBLIKUJ APLIKACJE](#)  
[ZMIEN HASŁO](#)  
[UZUPEŁNIJ PUNKTY](#)

Copyright © KILOH & BOAT . All rights reserved.

*Rysunek 33: Zarządzaniem profilem użytkownika*

**APPSHARE**

Leading sharing solutions

Szukaj | Logout [ stefan ] | Mój Profil

[Home](#)
[Aplikacje](#)
[Kategorie](#)
[Moje aplikacje](#)
[O nas](#)
[Kontakt](#)

### Twój profil, stefan

|                        |        |
|------------------------|--------|
| Login                  | stefan |
| Dostępne punkty        | 0      |
| Pobrane aplikacje      | 0      |
| Opublikowane aplikacje | 0      |

EDYCJA DANYCH UŻYTKOWNIKA

|          |                                           |
|----------|-------------------------------------------|
| Imie     | <input type="text" value="Stefan"/>       |
| Nazwisko | <input type="text" value="Stefanski"/>    |
| Mail     | <input type="text" value="stefan@o2.pl"/> |

[MOJE APLIKACJE](#)  
[MOJE PUBLIKACJE](#)  
[MÓJ PROFIL](#)  
[OPUBLIKUJ APLIKACJE](#)  
[ZMIEN HASŁO](#)  
[UZUPEŁNIJ PUNKTY](#)

OK

Rysunek 34: Zarządzaniem profilem – edycja danych

**APPSHARE**

Leading sharing solutions

Szukaj | Logout [ stefan ] | Mój Profil

[Home](#)
[Aplikacje](#)
[Kategorie](#)
[Moje aplikacje](#)
[O nas](#)
[Kontakt](#)

### Doładuj swoje konto

|                 |        |
|-----------------|--------|
| Login           | stefan |
| Dostępne punkty | 2      |

Zamień smsa na punkty, doładuj swoje konto

Wyślij sms o treści XXX na numer XXXX oraz wpisz otrzymany kod do formularza.

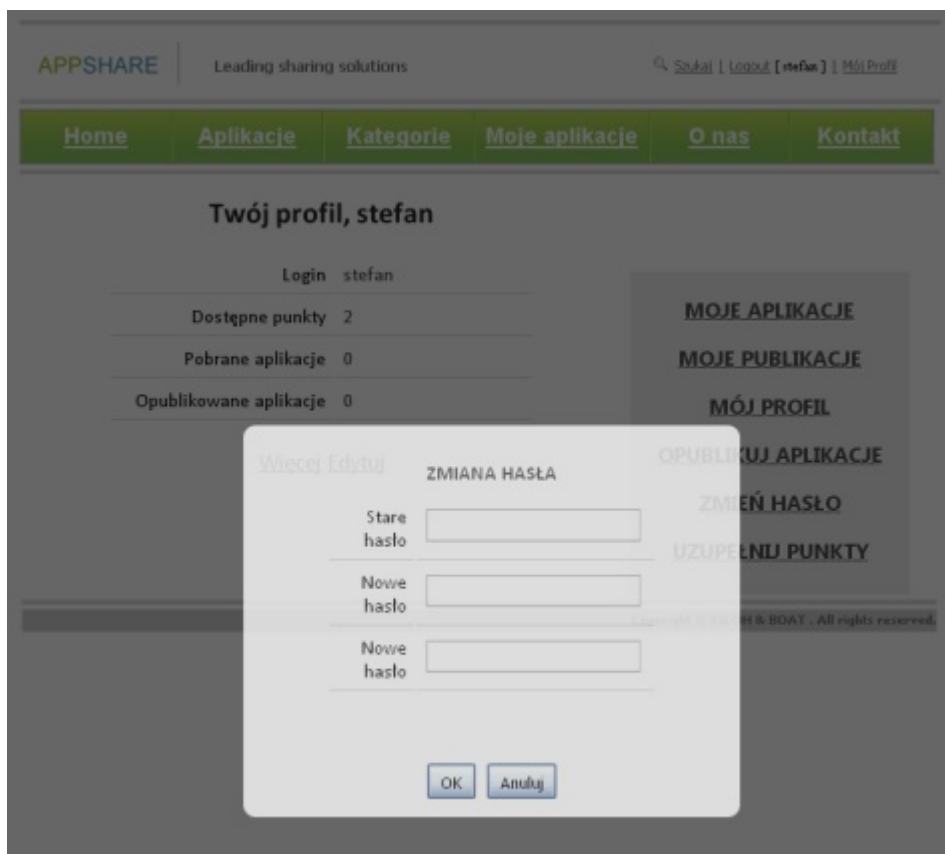
Twoje konto zostanie od razu zasilone nowymi punktami

Otrzymany kod

Ok

Copyright © KILOH & BOAT . All rights reserved.

Rysunek 35: Zarządzaniem profilem - doładowanie punktów

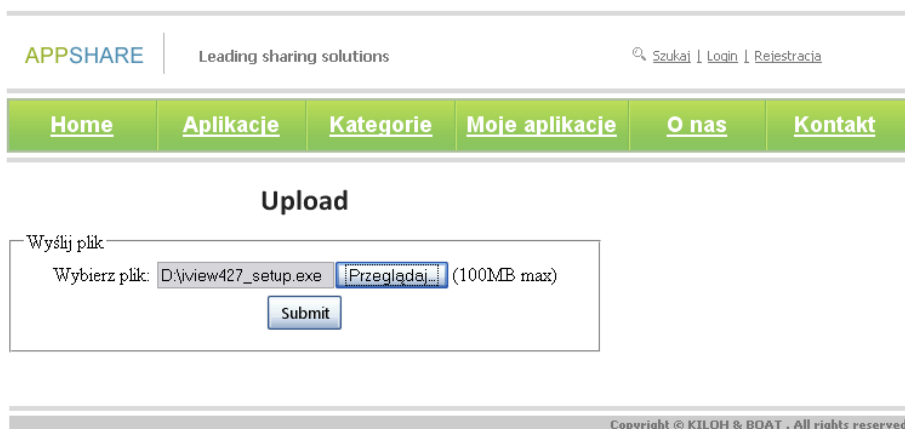


Rysunek 36: Zmiana hasła

### ***Publikacja własnej aplikacji***

Użytkownik ma możliwość opublikowania własnej aplikacji oraz w przyszłości wgrania nowszej wersji, Rysunek 37 przedstawia etap upload pliku.

Aplikacja będzie publicznie widoczna dopiero po jej akceptacji przez administratora.



Rysunek 37: Dodawanie własnej aplikacji



Rysunek 38 przedstawia stronę konkretnej aplikacji, znajdują się tutaj takie szczegóły jak nazwa, kategoria, wersja czy cena aplikacji.

Strona ta umożliwia również komentarz aplikacji przez użytkownika oraz pobranie aplikacji. Użytkownik może skomentować aplikacje tylko raz.

**APPSHARE** | Leading sharing solutions

Szukaj | Logout [ ppp ] | Mój Profil

[Home](#) | [Aplikacje](#) | [Kategorie](#) | [Moje aplikacje](#) | [O nas](#) | [Kontakt](#)

### IrfanView

|                 |                                     |
|-----------------|-------------------------------------|
| Autor           | stefan                              |
| Wersja          | 4                                   |
| Darmowe         | <input checked="" type="checkbox"/> |
| Cena            | 0                                   |
| Kategoria       | Programy                            |
| Data publikacji | 2010-08-12 19:48:08                 |
| Strona domowa   | http://www.irfanview.com/           |

  
**POBIERZ**

★★★★★

Oceń ( 2 )

[ admin ]  
2010-08-12 20:16

;)

[ ppp ]  
2010-08-12 20:17

fajny program

Już skomentowałeś, nie możesz ponownie komentować

Copyright © KILOH & BOAT . All rights reserved.

Rysunek 38: Strona aplikacji

Najnowsze (jak również popularne) aplikacje (Rysunek 39) oraz strona z wyborem kategorii (Rysunek 40)



Rysunek 39: Strona główna aplikacji



Rysunek 40: Kategorie

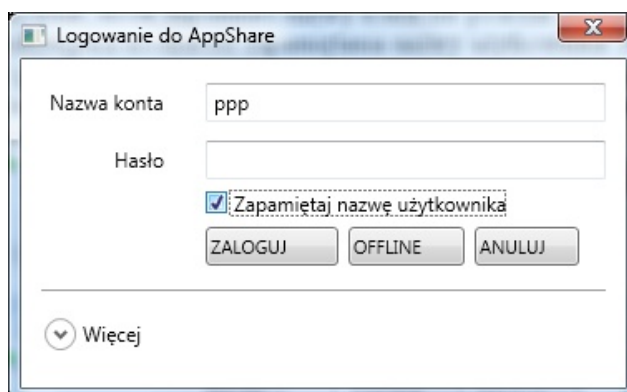
### 5.3. Aplikacja kliencka

Aplikacja kliencka stanowi dodatkową metodę dostępu do systemu. Posiada podstawową funkcjonalność jak np. zarządzanie swoim kontem czy dostęp do publikowanych programów. Ponadto w celu przekonania potencjalnych użytkowników do używania klienta desktopowego, rozszerzono funkcjonalność o możliwość tworzenia i archiwizowania list swoich ulubionych programów. Dodatkowo klient ułatwia proces instalacji poprzez automatyczne pobranie i uruchomienie instalatora. Użytkownik nie musi już szukać miejsca do którego ściągał program. Posiadając zainstalowane programy można je również odinstalować z poziomu klienta.

#### ***Logowanie czyli zanim uruchomimy klienta***

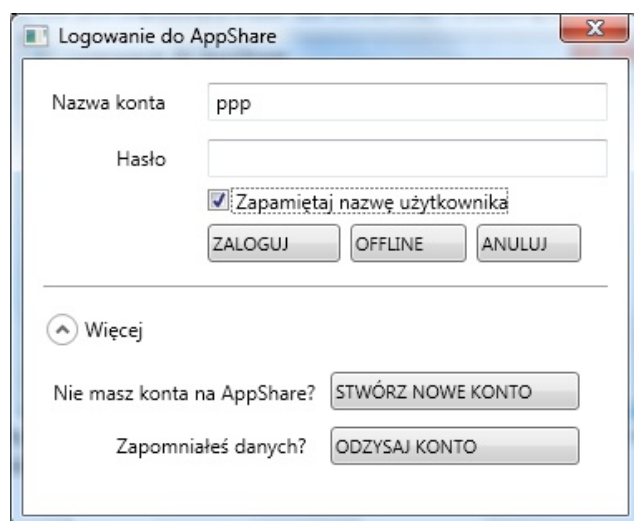
Za każdym razem przed uruchomieniem klienta zmuszeni jesteśmy się zalogować. W celu tym musimy posiadać konto, do którego przypisane są między innymi nazwa konta oraz hasło.

Ciągłe wpisywanie nazwy konta i hasła może narażać wiele problemów w praktyce. Między innymi użytkownik, może zapomnieć nazwy konta lub podczas wpisywania może popełnić literówkę. Program udostępnia możliwość zapamiętania nazwy użytkownika. Jeśli użytkownik zaznaczy tę opcję, to po udanym logowaniu program zapamięta użytą nazwę konta i przy następnym logowaniu pole to będzie już uzupełnione zapamiętaną nazwą. Ze względów bezpieczeństwa hasło nie jest zapamiętywane ani zestawiane z nazwą konta.

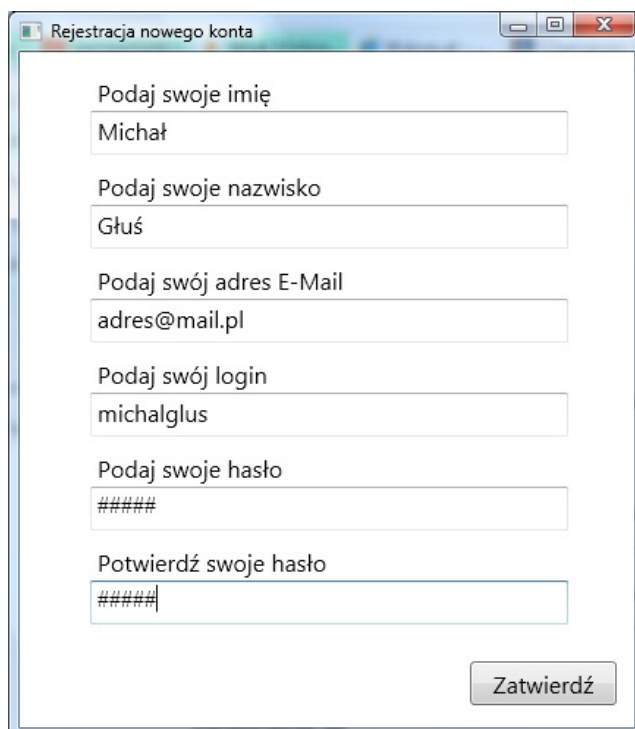


*Rysunek 41: Okno logowania*

## Rejestracja

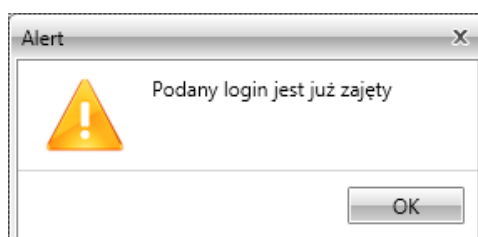


Rysunek 43: Rozszerzone okno logowania



Rysunek 42: Okno rejestracji

Panel logowania posiada jeszcze dodatkowe funkcje. Jedną z nich jest rejestracja nowego konta. Może się okazać, że użytkownik nie posiada jeszcze swojego konta. Podczas procesu rejestracji sprawdzane jest czy login lub adres mail nie jest już użyty w systemie. W takim przypadku użytkownik jest powiadamiany o konieczności podania innego loginu lub adresu mailowego.

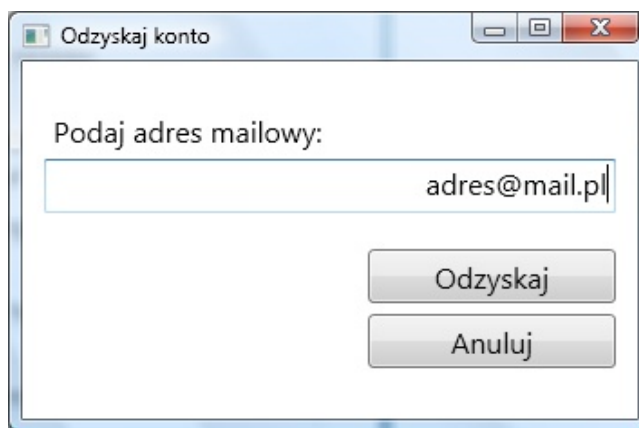


Rysunek 44: Komunikat o zajętych loginie

Dodatkowo poprzez potwierdzenie wprowadzonego hasła minimalizuje się prawdopodobieństwo popełnienia błędu podczas wprowadzania hasła.

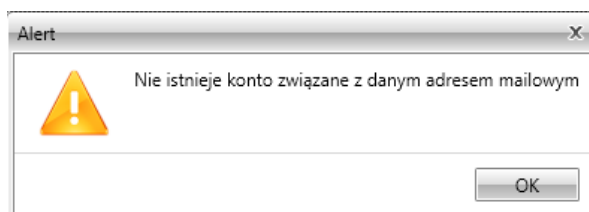
## Odzyskanie konta

Zawsze może się okazać, że użytkownik zapomni nazwy swojego konta lub użytego hasła. Z myślą o takich sytuacjach w oknie logowania umieszczono mechanizm odzyskiwania swojego konta.



Rysunek 45: Okno odzyskania konta

Sprawdzone jest czy istnieje konto z podanym adresem mailowym. Hasło wraz z nazwą konta jest wysłane na podany adres mailowy jeśli istnieje powiązane z tym adresem konto.



Rysunek 46: Komunikat o nieistniejącym adresie

Jeśli w systemie nie istnieje podany adres to użytkownik jest powiadamiany o tym fakcie.

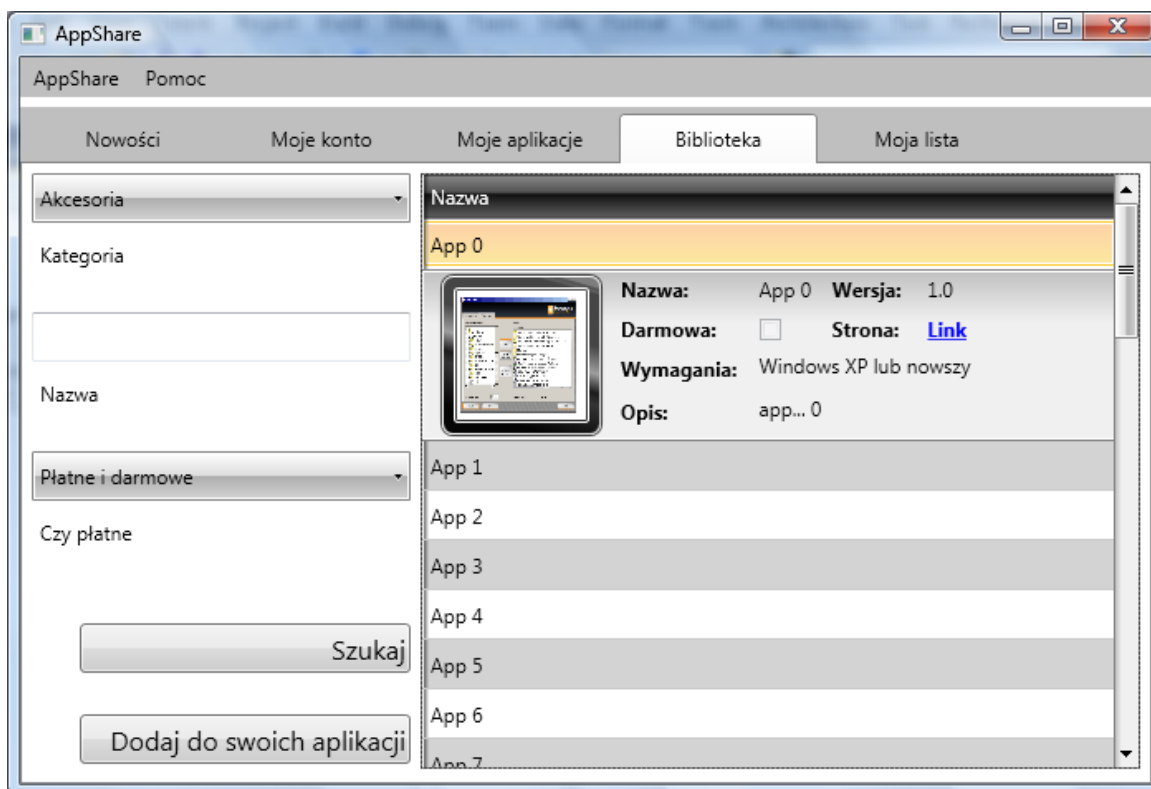
## Logowanie offline

Istnieje możliwość uruchomienia aplikacji w trybie offline. Możliwe jest to tylko i wyłącznie po co najmniej jednym logowaniu w normalnym trybie. Oczywiście w trybie tym funkcjonalność jest znacząco ograniczona. Nie mamy między innymi dostępu do panelu zarządzania swoim kontem oraz panelu publikowanych aplikacji.

## Główne okno aplikacji po zalogowaniu

Cała nawigacja odbywa się za pomocą głównego okna. Za jej pomocą można uzyskać dostęp między innymi do:

- nowości - można zobaczyć listę najnowszych płatnych oraz darmowych programów
- moje konto – udostępnia możliwość zarządzania swoim kontem oraz doładowania konta
- moje aplikacje – zawiera listę aplikacji jakie posiadamy (zarówno darmowe jak i płatne)
- biblioteka – umożliwia przeglądanie oraz wyszukiwanie publikacji w systemie
- moja lista – umożliwia tworzenie i archiwizowanie list swoich ulubionych programów



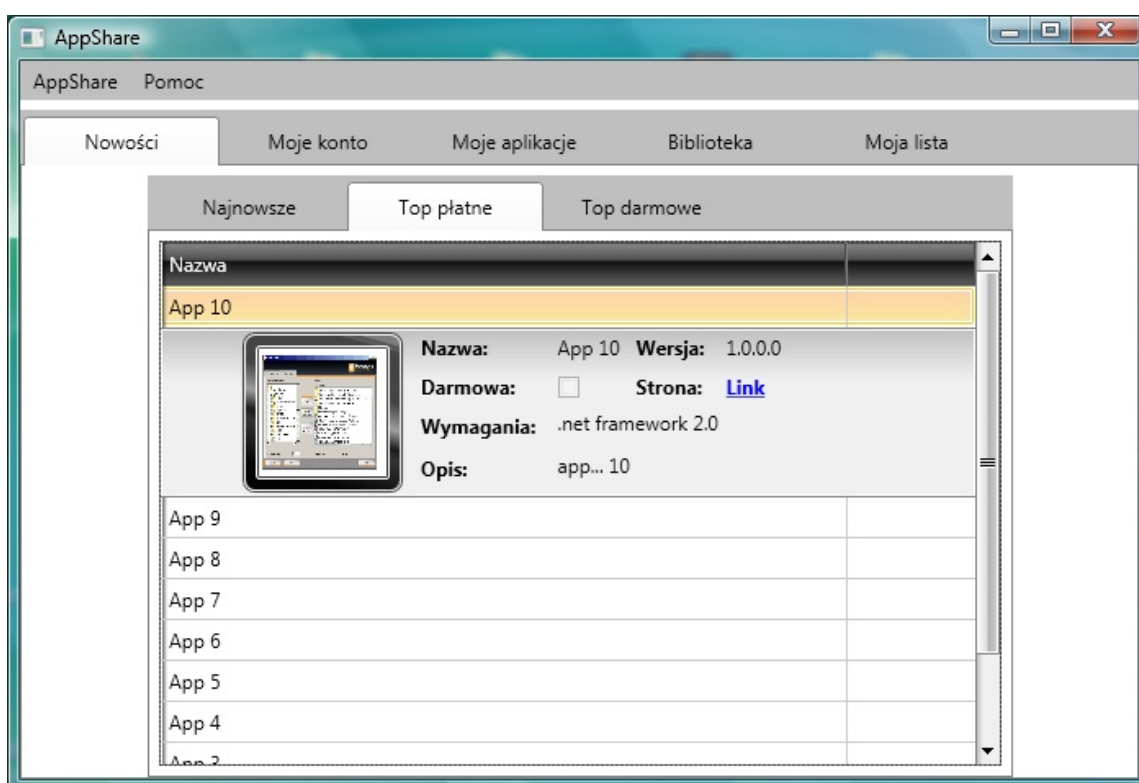
Rysunek 47: Główne okno

Mechanizm zakładek ogranicza liczbę i rozmiar otwartych okien, wnosząc przy tym jasne i czytelne rozmieszczenie poszczególnych funkcji w odpowiednie kategorie.

## Nowości

W dziale nowości użytkownik użytkownik ma przedstawione trzy listy aplikacji zgrupowane jako:

- najnowsze – grupa ostatnio opublikowanych aplikacji, zarówno płatne jak i darmowe
- top płatne – najlepiej oceniane przez użytkowników aplikacje płatne
- top darmowe – najlepiej oceniane darmowe aplikacje



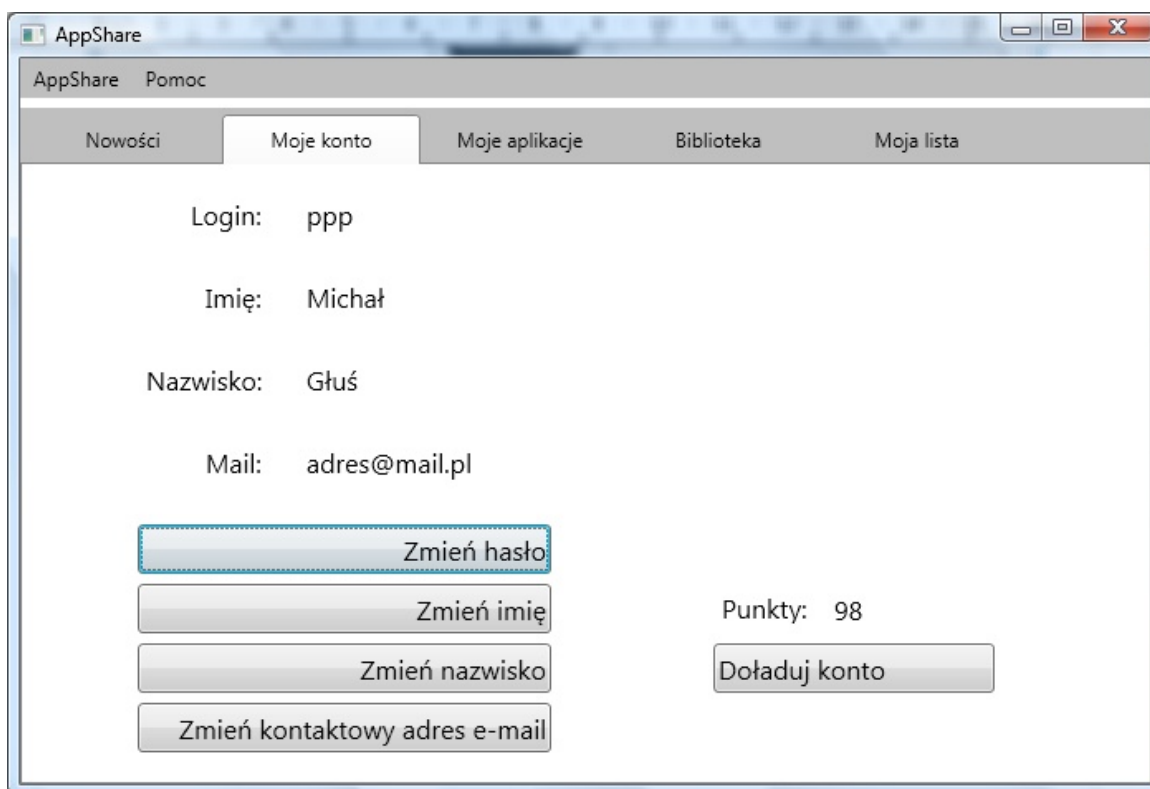
Rysunek 48: Zakładka: nowości

Lista programów zawiera podstawowe informacje jak nazwa programu, mini screen, wersja oraz wymagania. Dodatkowo użytkownik może w łatwy sposób odwiedzić stronę programu jeśli autor podał do niej link podczas publikacji programu. Jeśli któryś z programów zainteresuje użytkownika to po dwukrotnym kliknięciu ma dostęp do bardziej szczegółowego okna, gdzie przedstawione ma informacje o licencji oraz ewentualnej cenie w przypadku płatnego programu. W oknie tym użytkownik może dodać tą aplikację do swoich. Wiąże się to z koniecznością zaakceptowania dołączonej do programu licencji oraz w przypadku płatnego programu utraty określonej w cenie ilości punktów.

## Moje konto

Ta część aplikacji odpowiedzialna jest, za zarządzanie własnym profilem. Użytkownik za jego pomocą może wprowadzać zmiany do swoich danych, między innymi:

- dane osobowe jak imię i nazwisko
- dane kontaktowe – adres mailowy
- hasło do konta



Rysunek 49: Zakładka: moje konto

Oczywiście procedura zmiany hasła wymaga znajomości starego hasła. Użytkownik musi potwierdzić nowe hasło w celu uniknięcia pomyłki podczas podawania nowego hasła.

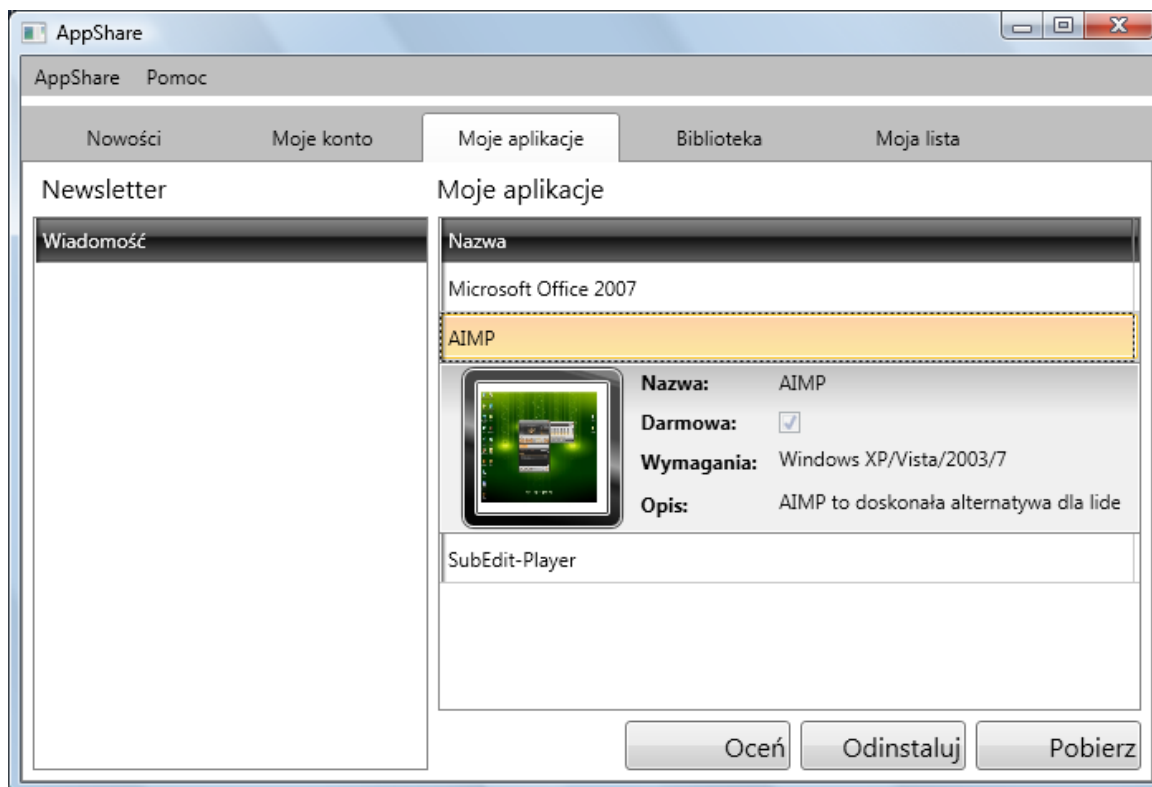
Z uwagi na wymóg unikalności adresów mailowych w systemie (jeden adres mailowy – jedno konto), podczas zmiany adresu sprawdzane jest czy nowy adres nie istnieje już w systemie.

Użytkownik może również doładować swoje konto. Po wysłaniu wiadomości SMS na odpowiedni numer otrzymuje kod zwrotny, który po wprowadzeniu zamieniany jest na punkty.

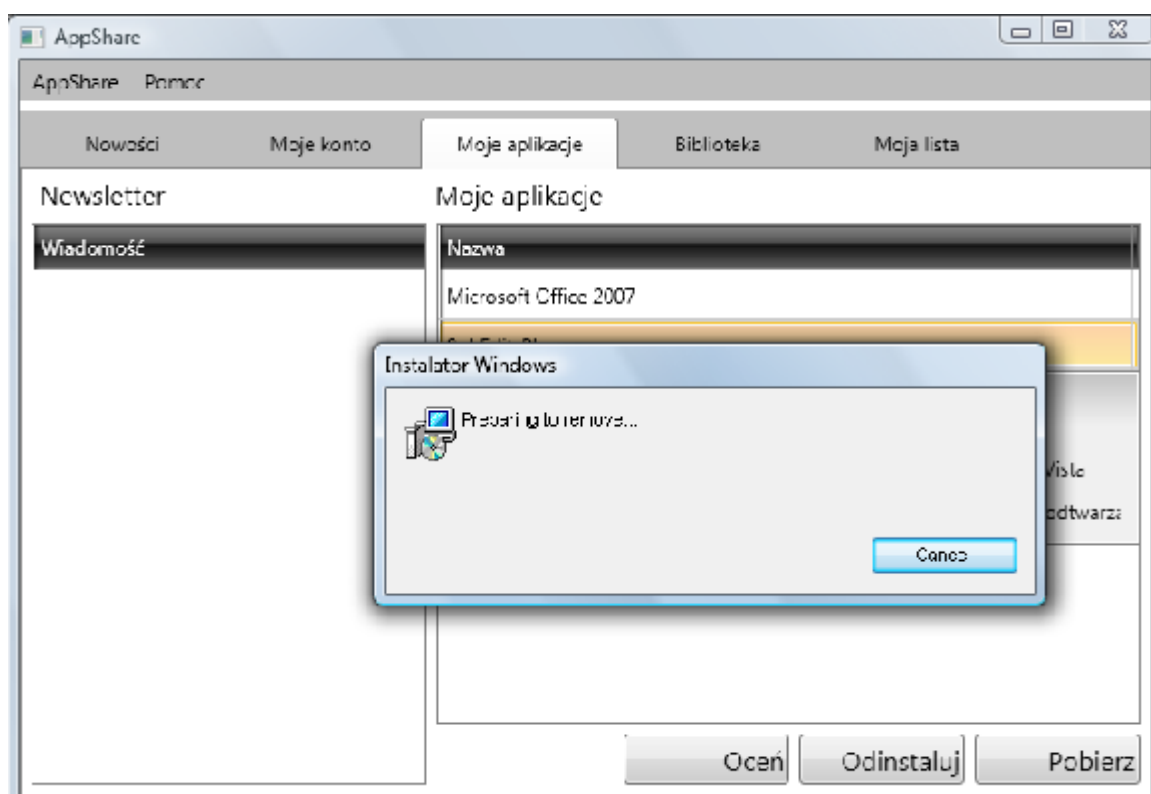


## Moje aplikacje

Ta część zawiera listę posiadanych przez użytkownika programów. Zarówno płatne i darmowe. Każdą z aplikacji może pobrać i zainstalować, wystawić swoją ocenę lub odinstalować.



Rysunek 50: Zakładka: moje aplikacje

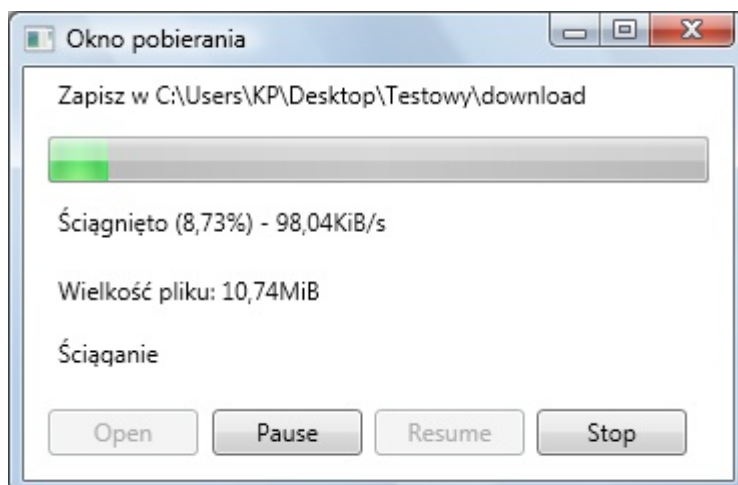


*Rysunek 51: Odinstalowanie programu z poziomu klienta*

Oprócz listy posiadanych programów, zakładka moje aplikacje posiada listę wiadomości z których może dowiedzieć się, że aplikacja, która posiada została opublikowana w nowej wersji. Klient przy uruchomieniu porównuje poprzednią listę aplikacji z tą która dostaje z serwera. Wychwytyjąc różnicę między nimi dedukuje, czy pojawiła się nowsza wersja programu, którego posiadamy.

Użytkownik może z poziomu klienta odinstalować swój program. Dzięki temu ma większą kontrolę nad oprogramowaniem pobranym z naszego serwisu. Na etapie prototypu wspierane jest odinstalowywanie jedynie programów opublikowanych w pakiecie instalacyjnym „MSI”.

## ***Pobieranie i instalacja***



*Rysunek 52: Okno pobierania*

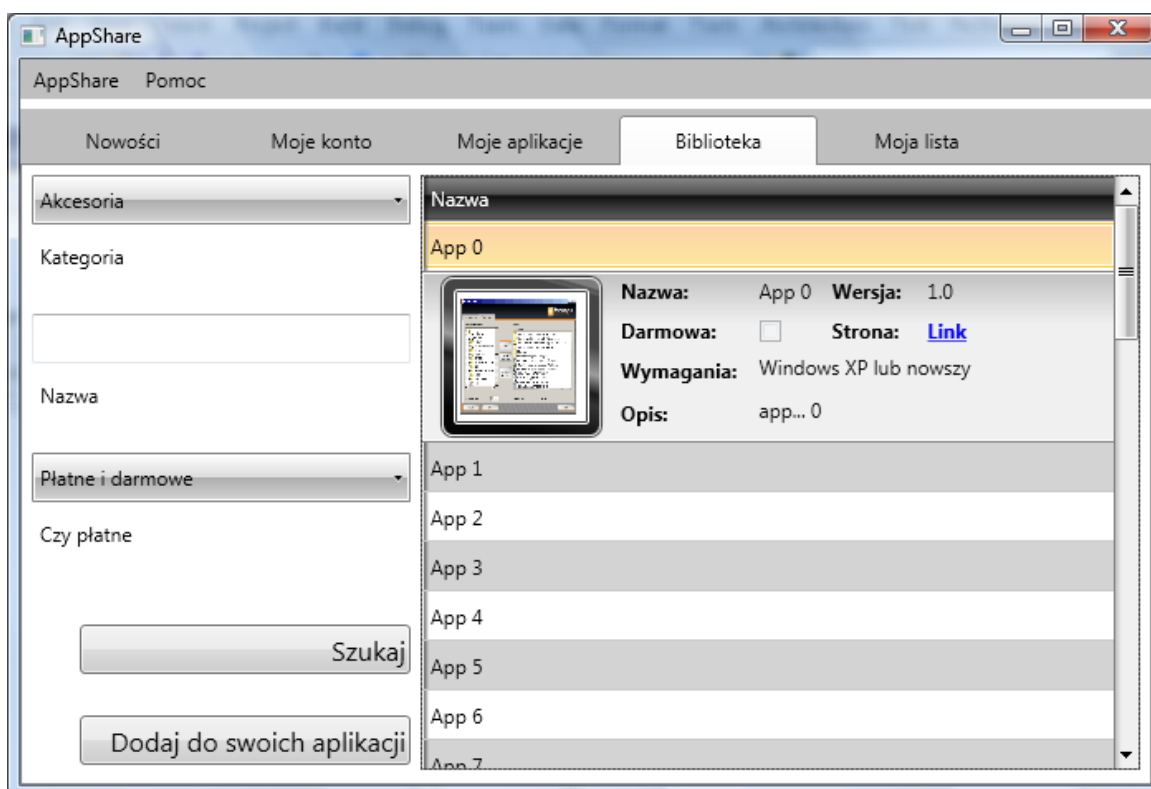
Proces pobierania i instalacji jest częściowo zautomatyzowany. Program pyta się o miejsce docelowe, w którym ma zapisać pobieraną aplikację po czym rozpoczyna pobieranie.

Po ukończonym pobieraniu klient w przypadku pliku instalacyjnego lub msi uruchamia pakiet instalacyjny. W każdym innym jak np. exe otwiera katalog do którego aplikacja została zapisana.

## Biblioteka

Zakładka biblioteka umożliwia dostęp oraz zapewnia podstawowe mechanizmy wyszukiwania publikowanych w systemie aplikacji. Użytkownik może wyszukiwać aplikacje na podstawie:

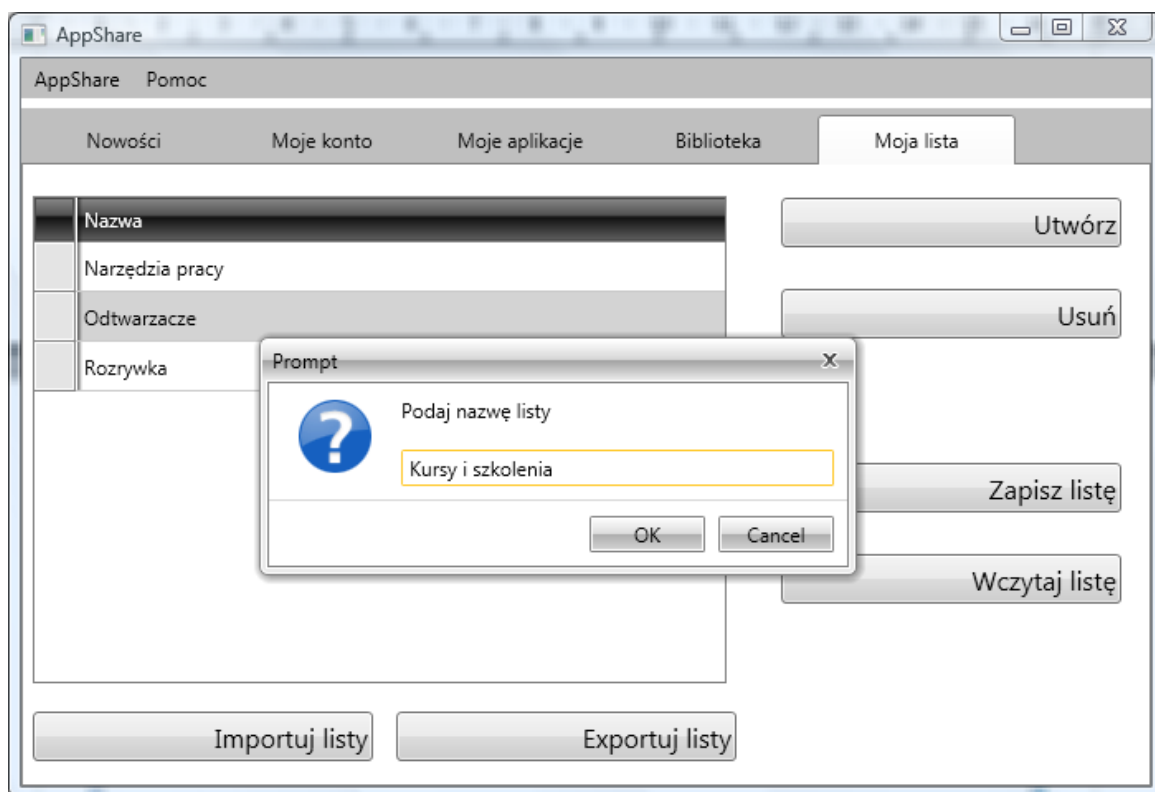
- fragmentu nazwy programu
- kategorii programów, które go interesują
- czy program jest płatny czy darmowy



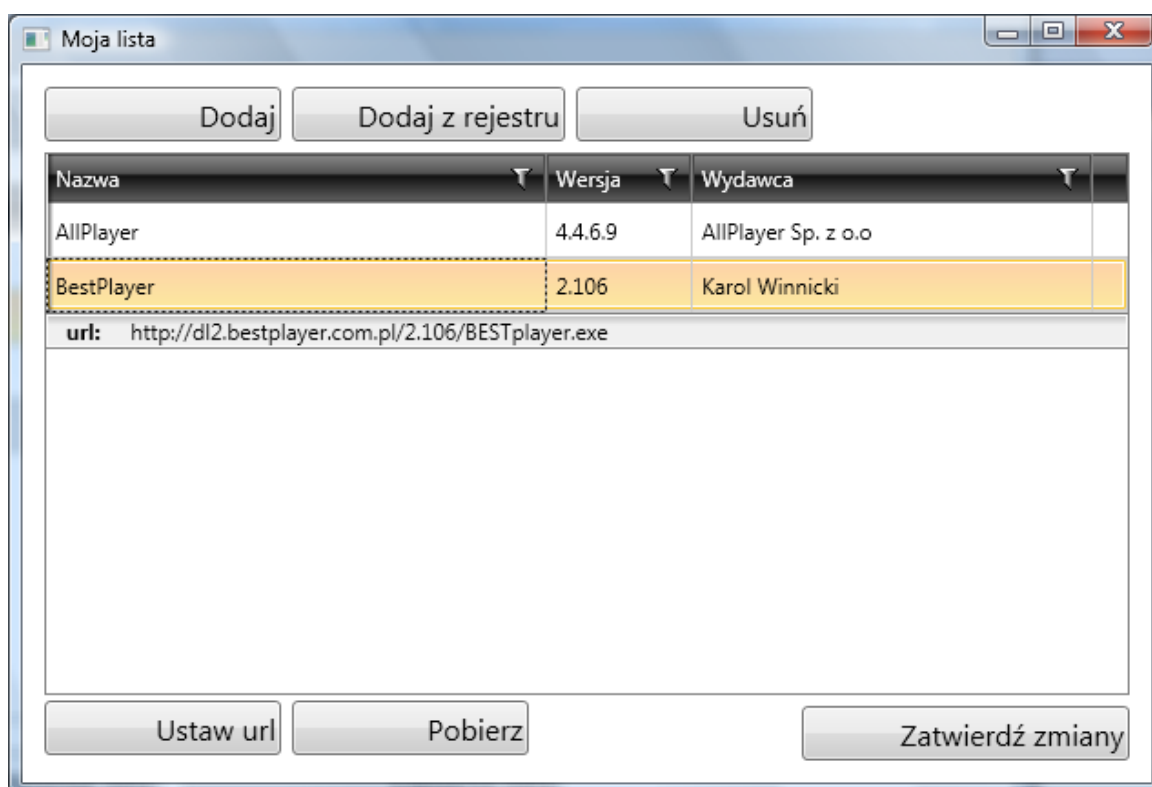
Rysunek 53: Zakładka: biblioteka

## Moja lista

Klient wnosi również pozasystemową funkcjonalność, która polega na możliwości tworzenia własnych list programów oraz archiwizowanie ich. Użytkownik może tworzyć np. własną listę programów służących do tworzenia i odtwarzania filmów. Listy może wyeksportować do pliku oraz wymieniać się nimi z innymi użytkownikami.



Rysunek 54: Zakładka: moja lista

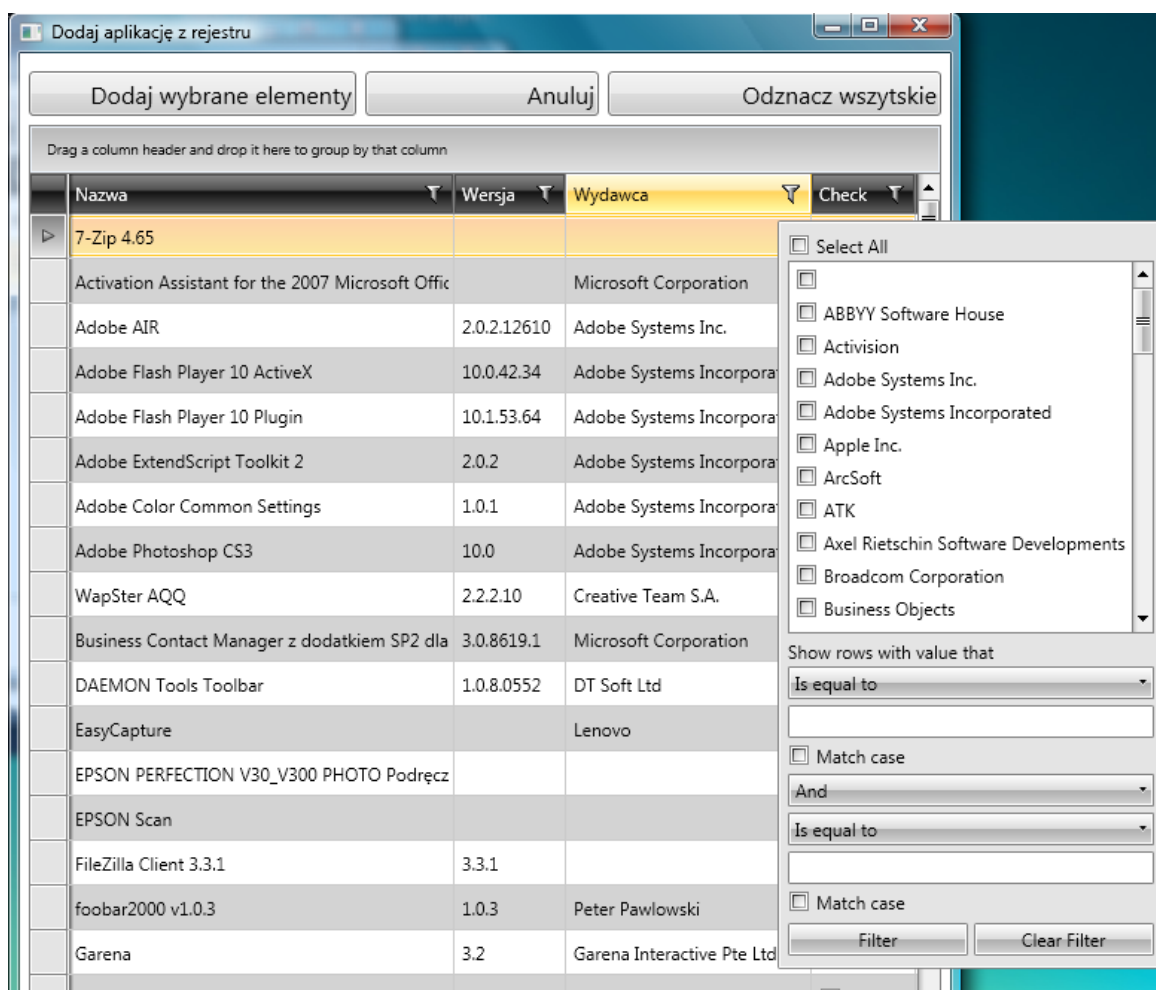


*Rysunek 55: Moja lista w szczegółach*

Listy zawierają takie informacje jak nazwa, wersja oraz wydawca programu. Dodatkowo posiadają informacje o miejscu z którego można je pobrać.

*Rysunek 56: Ręczne dodanie programu*

Jedną z metod dodania programu do listy jest ręczne wprowadzenie informacji o programie.



Rysunek 57: Aplikacje zainstalowane na komputerze

Znacznie wygodniej jest dodać program na podstawie informacji zawartych w rejestrze systemu. Rejestr mimo wcześniejszego przefiltrowania przez program może nadal posiadać bardzo dużą liczbę wpisów co może utrudnić wybór interesujących nas programów. Z tego powodu zastosowano mechanizm filtrowania oraz sortowania.

### ***Ikona programu w pasku systemowym (tray)***

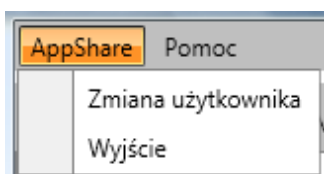
Wielkość pulpitu oraz paska menu jest ograniczona. Brak miejsca jest szczególnie dotkliwy dla osób, które mają dość dużo uruchomionych aplikacji w tle. Klient posiada możliwość minimalizacji do postaci ikony w tray, minimalizując w ten sposób ilość zajmowanego miejsca przestrzeni roboczej użytkownika.



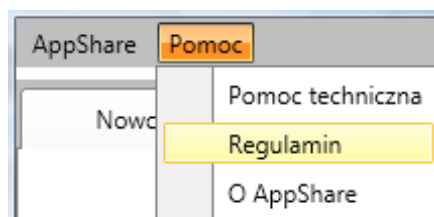
*Rysunek 58: Ikona w tray*

### ***Pasek menu***

Posiada dwie zakładki. Jedna z nich umożliwia zamknięcie program oraz zmianę zalogowanego użytkownika. Druga zapewnia dostęp do informacji o programie, regulaminie oraz pomocy technicznej.



*Rysunek 60: Pasek menu: AppShare*



*Rysunek 59: Pasek menu: Pomoc*



## 6. Zalety, wady oraz plany rozwoju

Przedstawiony w poprzednim rozdziale prototyp, realizuje podstawowe założenia (w pewnym stopniu uproszczone) w sposób wystarczający do przedstawienia sposobu dystrybucji treści elektronicznej w sieci internet.

Pomimo implementacji większości wymagań prototypu nie można uznać za skończony produkt gotowy do wdrożenia. Zdaniem autorów jest on dobrym punktem wyjścia dla dokładnej analizy potrzeb rynkowych, preferencji użytkowników, a po rozszerzeniu o brakującą funkcjonalność może być przydatny przy testach z użytkownikami końcowymi.

### 6.1. Zalety oraz wady przyjętych rozwiązań

Niewątpliwą zaletą omawianego systemu są różne metody dostępu do niego, użytkownik może zdecydować czy chce korzystać z klienta zainstalowanego na komputerze czy z aplikacji internetowej. Zapewnia to użytkownikom łatwy dostęp do systemu jak również nie ogranicza ich pod kątem używanych przez nich różnych komputerów, systemów i programów.

Publikowanie własnej aplikacji, a co wiąże się z tym sposób akceptacji jej w prototypie może być spornym elementem pod względem zakwalifikowania do zalet oraz wad.

Z jednej strony zastosowane podejście wymaga każdorazowej weryfikacji przez administrację, jednak jest rozwiązaniem zdecydowanie bezpiecznym w założeniu uniemożliwiającym dodania „złej” aplikacji do systemu. Z drugiej jednak strony autorzy zdają sobie sprawę z ogromnego obciążenia administratorów samym akceptowaniem aplikacji, przy dalszym rozwoju należy zastanowić się nad innymi sposobami weryfikacji aplikacji np. pobierania opłaty rejestracyjnej, wprowadzenia możliwości „zbiorowej-niejawnej” weryfikacji przez użytkowników.

Elementem którego nie udało się w pełni zaimplementować jest automatyczna aktualizacja oprogramowania. Biorąc pod uwagę specyfikę rejestru Windows, a dokładnie informacje o zainstalowanym oprogramowaniu dość często może zdarzyć się sytuacja w której w rejestrze są niepełne informacje (np. brak informacji o zainstalowanej wersji, wydawcy), dodatkowo nie ma żadnej gwarancji, że przy następnej wersji informacje w rejestrze będą takie same jak w poprzedniej. Praktycznie uniemożliwia to jednoznaczną identyfikację aplikacji (opierając się jedynie na informacjach pochodzących z rejestru). Pomimo, iż można to zakwalifikować do wad, w pewnym sensie kompromisowe może wydawać się przyjęte rozwiązanie, w którym to informacje o aktualizacji „pojawiają się” jedynie dla aplikacji które użytkownik wcześniej pobrał.

Niestety wadą tego jest to, że użytkownik może nigdy nie zainstalować aplikacji po jej pobraniu.

Interfejs graficzny aplikacji trudno zaliczyć do wad lub zalet (szczególnie przez autorów), jego główną zaletą jest czytelność, wyświetlane są tylko te informacje które są niezbędne, brak jest też wielu elementów mogących wprowadzić użytkownika w błąd lub dezorientację. Kolejną zaletą jest dostęp do wszystkich najważniejszych funkcjonalności systemu w maksymalnej odległości 4 kliknięć, co upraszcza nawigację po systemie jednoznacznie eliminując potrzebę stosowania skomplikowanych sposobów nawigacji. Następną zaletą interfejsu użytkownika w aplikacji internetowej jest jego budowa umożliwiająca poprawny dostęp dla urządzeń o małej rozdzielczości (np. netbooki)

## **6.2. Plany rozwojowe**

### ***Przyjazność użytkownika***

Wszelkie operacje powinny być jak najprostsze (najlepiej wzorowane na AppStore lub Android Market), zdaniem autorów przy wprowadzaniu jakichkolwiek modyfikacji, rozszerzeń lub udogodnień dla użytkownika konieczne jest sprawdzenie czy nowa funkcjonalność nie może być użyta inaczej niż przewidzieli twórcy, czy istnieje prosta i czytelna nawigacja i najważniejsze czy podobna funkcja nie jest już wykorzystywana w innej części. Zdaniem autorów zamiast „zwielokrotniania” istniejącej funkcjonalności lepszym rozwiązaniem jest modyfikacja i optymalizacja dotychczasowej.

### ***Wieloplatformowość***

Choć w obecnej wersji wieloplatformowość została uwzględniona poprzez stworzenie niezależnej od systemu metody dostępu (aplikacja internetowa), system został zaprojektowany tak, jak gdyby aplikacje na systemy inne niż Microsoft Windows nie zostały uwzględnione. Funkcjonalność mogła by być rozszerzona o inne platformy.

### ***Internacjonalizacja***

Kolejnym krokiem rozwoju aplikacji jest obsługa innych krajów. Nie dotyczy to samej zmiany języka w aplikacji. Jest to problem dość złożony obejmujący płatności, a co z tym wiąże się inne waluty, nowe sposoby płatności (inne niż obecnie DotPay).

## **Płatności**

Przed wdrożeniem aplikacji powinny zostać rozważone inne rodzaje płatności np. przelew bankowy, PayPal, płatność kartą płatniczą lub o nowe innowacyjne systemy płatności np. mPay (płatność przez telefon komórkowy –przykład wdrożenia ZTM).

Nie mniej ważne są sprawy rozliczeń finansowych. Osoby zarejestrowane jako programiści otrzymują prowizję od sprzedanych aplikacji, do rozliczenia potrzebują faktury VAT lub innego sposobu rozliczenia dla osób fizycznych np. umowa kupna-sprzedaży. W przyszłości system mógłby wspierać rozliczenia pomiędzy użytkownikami i urzędem skarbowym.

## 7. Podsumowanie

Celem niniejszej pracy było przetestowanie obecnie dostępnych rozwiązań wspomagających dystrybucję oprogramowania w sieci internet pod kątem opracowania modelu dystrybucji.. Obszar ten zawiera wiele bardzo ciekawych i oryginalnych rozwiązań. W naszej pracy skupiliśmy się na kilku z nich, prezentując ich wady i zalety. Zostały one poddane analizie w celu opracowanie cech i zasad systemu dystrybucji treści w sieci internet, którymi powinien odznaczać się dobry system dystrybucji elektronicznej. Dodatkowo zagadnienie to (dystrybucja elektroniczna w sieci) zostało zbadane pod kątem prawnym w celu ustalenia tego w jaki sposób polskie prawo reguluje i wpływa na relacje usługodawca – autor programu – użytkownik programu.

Po zdefiniowaniu powyższych cech powstał prototyp pod nazwą „AppShare”. Składa się on z aplikacji internetowej oraz klienta. Wszystkie niezbędne funkcje zostały w nim zaimplementowane. Za ich pomocą można m.in. opublikować swoją aplikację, jak również nabyć lub pobrać. Ponadto prototyp został napisany z użyciem sprawdzonych wzorców projektowych takich jak „Model-View-Controller” oraz dzięki zastosowaniu odpowiedniej architektury i interfejsów, umożliwia łatwą rozbudowę o nową funkcjonalność w przyszłości.

## 8. Bibliografia

1. "Ustawa o świadczeniu usług drogą elektroniczną" -  
<http://lex.pl/bap/student/Dz.U.2002.144.1204.html>
2. „Świadczenie usług drogą elektroniczną” Iwona Flis -  
<http://archiwum.parp.gov.pl/partnerinfo/przewodnik/06dz08.rtf>
3. „Licencje oprogramowania” Kowalski Jan -  
<http://ckzwm.edu.pl/doc/informatyka/licencja.doc>
4. „Analiza porównawcza modeli licencjonowania oprogramowania” Przemysław Kulczycki -  
[http://prawo.vagla.pl/files/mgr\\_p\\_kulczycki.pdf](http://prawo.vagla.pl/files/mgr_p_kulczycki.pdf)
5. Czynności prawne -  
[http://www.prawnik24.pl/kodeksywilny/prawnik\\_kodeks\\_cywilny\\_4.htm](http://www.prawnik24.pl/kodeksywilny/prawnik_kodeks_cywilny_4.htm)
6. “What Motivates Software Crackers?”, Journal of Business Ethics, Vol. 65, No. 2, 2006,  
173-201 Prof Sigi Goode and Sam Cruise
7. Badania popularności języków programowania -  
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
8. Architektura NHibernate -  
<http://nhforge.org/wikis/reference2-0en/architecture.aspx>
9. [http://en.wikipedia.org/wiki/.NET\\_Framework](http://en.wikipedia.org/wiki/.NET_Framework)
10. <http://msdn.microsoft.com/en-us/library/zw4w595w.aspx>
11. <http://www.microsoft.com/poland/sql/overview.aspx>
12. <http://www.codeproject.com/KB/WCF/WCFOverview.aspx>
13. <http://msdn.microsoft.com/en-us/library/dd394709.aspx>
14. <http://www.codeproject.com/KB/architecture/NHibernateBestPractices.aspx>