**PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA W NOWYM SĄCZU**

**INSTYTUT TECHNICZNY**

**PRACA DYPLOMOWA**

**Aplikacja obsługująca Service Desk wspomagająca komunikację między pracownikami przedsiębiorstwa**

**Autor: Marcin Ziemnik**

**Kierunek: Informatyka**

**Nr albumu: 28291**

**Promotor: dr inż. Witold Przygoda**

**NOWY SĄCZ 2021**

Spis treści

[Wstęp - 5 -](#_Toc92195772)

[1. Cel i zakres pracy - 6 -](#_Toc92195773)

[2. Środowisko pracy - 7 -](#_Toc92195774)

[2.1. Środowisko sprzętowe - 7 -](#_Toc92195775)

[Środowisko programowe - 9 -](#_Toc92195776)

[3. Bibliografia - 15 -](#_Toc92195777)

Wstęp

Technologie informatyczne wspomagają pracę prawie wszystkich przedsiębiorstw. Co raz to częściej podejmując próbę kontaktu z pomocą techniczną natrafiamy na zautomatyzowany system odpowiedzi czy infolinię przekierowującą połączenia po zakwalifikowaniu problemu do danej kategorii. Bardzo często maszyna nie jest w stanie zapewnić nam poziomu interakcji i zrozumienia problemu, które może nam okazać inny człowiek. W tym celu większość dużych korporacji korzysta z systemów obsługujących centra obsługi tzw. Service Desk, których zadaniem jest ułatwienie pracownikom zmagającym się z problemami natury technicznej uniemożliwiającymi im wykonywanie pracy, nawiązania kontaktu z bardziej wykwalifikowanym pracownikiem. Struktura podziału w takich systemach opiera się na zasadach opisanych w zbiorze publikacji ITIL opisującym najlepsze praktyki zarządzania usługami informatycznymi. Możemy wyróżnić cztery następujące typy:

1. Call Center którego zadaniem jest tylko i wyłącznie rejestrowanie zgłoszeń (nazywanych incydentami) otrzymywanych od klientów.
2. Niewykwalifikowany Service Desk (*First Level Support*) zajmujący się rejestracją incydentów, ich klasyfikacją ze względu na stan i typ problemu oraz przekazywaniem ich do kolejnych poziomów pomocy. Posiada również dostęp do zasobów pomagających rozwiązać najprostsze i najczęściej spotykane problemy.
3. Wykwalifikowany Service Desk (*Second Level Support*) – Przejmuje incydenty przekazane przez pomoc pierwszego poziomu. Posiadają oni wiedze i doświadczenie pozwalające na rozwiązanie średnio skomplikowanych problemów. Jeżeli problem jest związany z zewnętrznym podmiotem to korzystają z pomocy 3 poziomu.
4. Eksperci (*Third Level Support*) – Zwykle oznacza producentów sprzętu lub oprogramowania. Ich zadaniem jest naprawa najcięższych problemów i przywrócenie możliwości pracy w jak najkrótszym czasie.

Aby uzyskać zadowalającą wydajność przy wyżej wymienionym systemie pracy konieczne jest zwrócenie szczególnej uwagi na sposoby komunikacji między poszczególnymi warstwami Service Desk. Dodatkowym zadaniem pełnionym przez centrum obsługi jest zbieranie informacji na temat poszczególnych problemów i zapewnienie sposobu na ich monitorowanie i badanie aby zapewnić możliwość ciągłego rozwoju.

Z tego powodu w tej pracy podjąłem się stworzenia aplikacji, której zadaniem jest wspomaganie pracowników przedsiębiorstw poprzez stworzenie interfejsu dla komunikacji pomiędzy wyżej wymienionymi warstwami. Program działać będzie pod najpopularniejszym systemem do obsługi komputerów osobistych – Microsoft Windows. Dodatkowo wprowadzi funkcjonalność podstawowych raportów pozwalających na klasyfikację i analizę zgłoszeń.

# Cel i zakres pracy

Moim celem jest stworzenie oprogramowania wspomagającego komunikację między klientami i poszczególnymi poziomami pomocy technicznej. W tym celu zaprojektuję i stworzę aplikację, która w łatwy sposób pozwoli na tworzenie i zarządzanie incydentami oraz po zakwalifikowaniu przekaże je do odpowiedniego poziomu. Drugim celem będzie stworzenie systemu bazodanowego obsługującego autoryzację użytkowników systemu, pozwalającego na określenie poziomu uprawnień każdego z nich oraz zbierającego informację o incydentach zgłaszanych przez klientów. Dzięki informacjom znajdującym się w bazie danych zaprojektuje również system raportów opisujących między innymi najczęściej pojawiające się typy problemów. Dodatkowo podsumują one statystyki poszczególnych pracowników systemu *ServiceDesk*.

# Środowisko pracy

Celem tego rozdział jest przybliżenie środowiska sprzętowego i programowego koniecznego do wypełnienia celu pracy.

* 1. Środowisko sprzętowe

Ten podrozdział przedstawia urządzenia używane w niniejszym projekcie inżynierskim. Przybliża on pojęcie komputera osobistego wraz z jego alternatywnym nazewnictwem – „PC”. Dodatkowo przedstawia informację na temat zastosowania tego typu urządzeń w życiu codziennym.

**„Komputer osobisty** (ang. personal computer) to mikrokomputer przeznaczony do użytku osobistego w domu lub biurze. Może być to zarówno urządzenie typu desktop (inaczej komputer stacjonarny) lub notebook. Może służyć zarówno do uruchamiania oprogramowania biurowego, dostępu do zasobów Internetu jak i prezentacji treści multimedialnych takich jak teskty, obrazy, dźwięki, filmy i inne oraz gier.” (Definicja komputera osobistego:) Możemy zarówno spotkać się z gotowymi zestawami danych producentów jak i takimi samodzielnie składanymi przez użytkowników. Te drugie *do niedawna* były jeszcze rzadkością, ale atrakcyjność ceny uzyskiwana poprzez usunięcie marży powoduje, że stają się one co raz to bardziej popularne wśród użytkowników domowych. Wśród komputerów osobistych wyróżniamy dwie architektury – 32 i 64 bitową. Bardzo często możemy spotkać się z nazwą PC pochodzącą od jednego z prekursorów tego typu urządzeń – IBM PC (IBM 5150) wydanego w 1891 roku przez przedsiębiorstwo IBM. Idea tego typu komputerów pojawiła się na początku lat 70. Nie można jednoznacznie stwierdzić którą maszynę można uznać za pierwszą tego typu, ale wśród firm związanych z tym zagadnieniem możemy wyróżnić trzech ojców – Xerox, Hewlett-Packard oraz IBM. Pierwszym urządzeniem powszechnie uważanym za komputer osobisty przez świat zachodni była konstrukcja firmy MITS nazwana Altair 8800 zaprezentowana przez Eda Robertsa i Billa Gatesa w styczniowym wydaniu *Popular Electronics Magazine* w 1975 roku, natomiast pierwszym masowo produkowanym komputerem PC był Apple 1. Niedługo później rozpoczął się intensywny rozwój komputerów osobistych. Sprzedawany wtedy Apple II zawierał prosty edytor tekstowy i arkusz kalkulacyjny które bardzo szybko znalazły zastosowanie w prostych pracach biurowych. Za urządzeniami Apple poszedł IBM który dodał ideę otwartej architektury budowy komputera i wydał wcześniej wymieniony IBM 5150 zawierającego miedzy innymi następujące parametry:

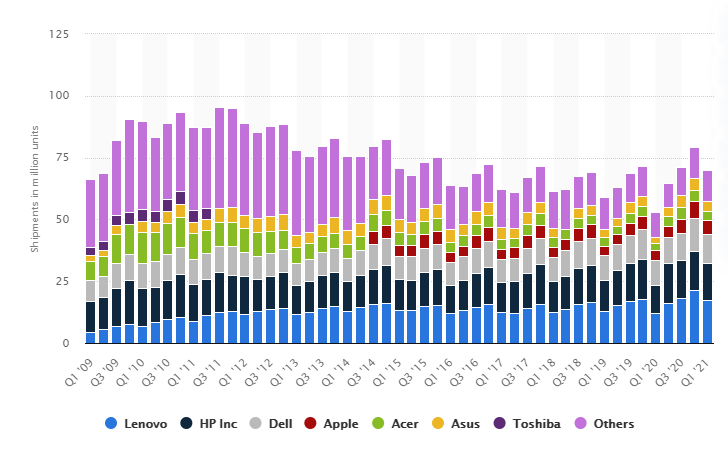
Tabela 1

Parametry techniczne IBM 5150

|  |  |
| --- | --- |
| Parametr | Wartość |
| Rozmiar: | Wysokość: 5,5 cala, szerokość 20 cali, długość 16 cali |
| Waga: | Od 21 do 28 funtów |
| Zasilanie: | 120 v. AC |
| Czas cyklu: | Główny magazyn – 410 nanosekund  Dostęp – 250 nanosekund |
| Pamięć: | 40 K pamięci ROM  16 do 256 K pamięci użytkownika |

*Opracowanie własne na podstawie:*

Oprócz tego zawierał wbudowaną klawiaturę oraz jednokolorowy wyświetlacz. Od tamtego czasu nastąpił ogromny wzrost technologiczny w dziedzinie komputerów osobistych i dzisiejsze maszyny oferują ogromną wydajność i niewielki rozmiar przez co stały się koniecznością przy pracach biurowych i nie tylko. W śród sprzedawców tego typu sprzętu możemy wyróżnić dwie największe korporacje cechujące się największym udziałem rynkowym– Lenovo, Dell, Hewelett-Packard(znany jako HP). Inne firmy o mniejszym, ale znaczącym udziale to Apple, Acer, Asus i Toshiba. W samym roku 2020 około 275,15 miliona komputerów osobistych zostało wypuszczonych na rynek. Jest to kontynuacja trwającego od nie dawna wzrostu zapotrzebowania na komputery domowe związanego z epidemią korona wirusa. Wcześniejszy spadek zainteresowania tego typu urządzeniami związany był z niedoborem jednostek CPU produkowanych przez Intel, zwiększeniem znaczenia systemów opartych na chmurze danych oraz częściowe zastępowanie komputerów osobistych przez smartfony.



Rysunek 1. Sprzedaż komputerów osobistych z podziałem na kwartwały i korporacje

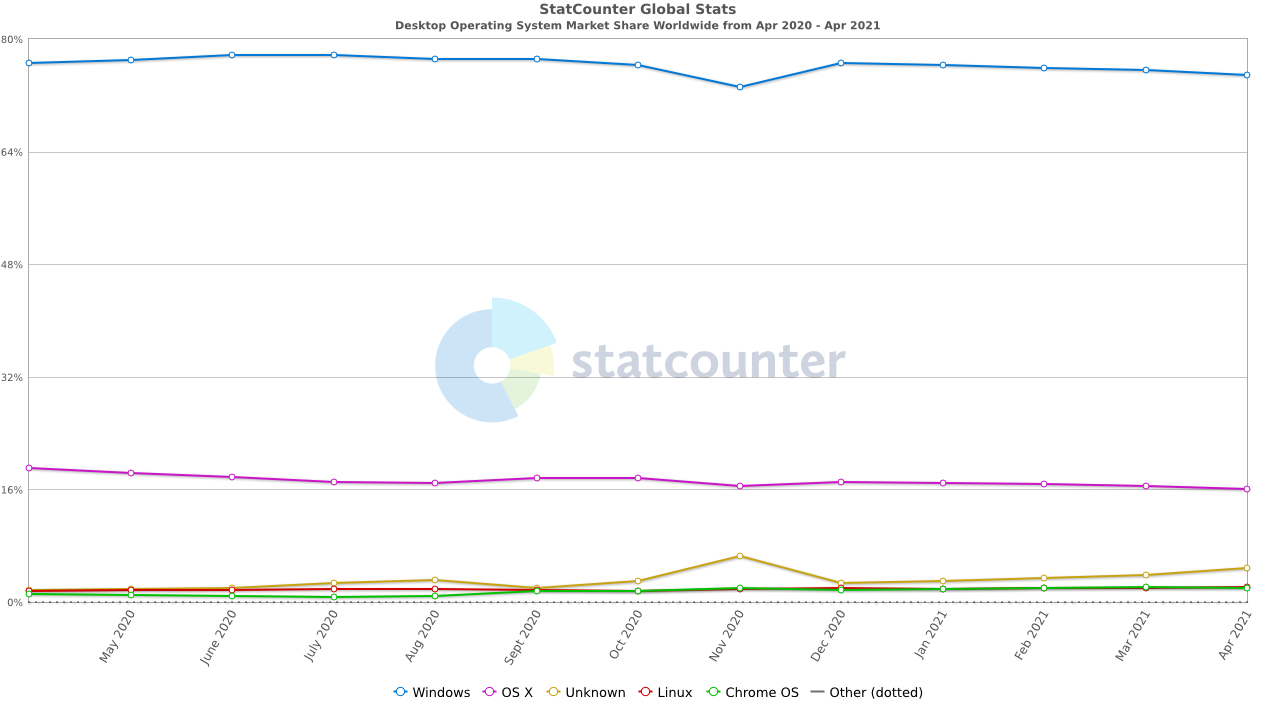
*(Źródło: )*

Środowisko programowe

Ten podrozdział przybliży system operacyjny który będzie zainstalowany na urządzeniach korzystających z oprogramowania. Dodatkowo przedstawi środowisko i platformę programistyczną, język programowania, Framework oraz najważniejsze biblioteki używane w programie.

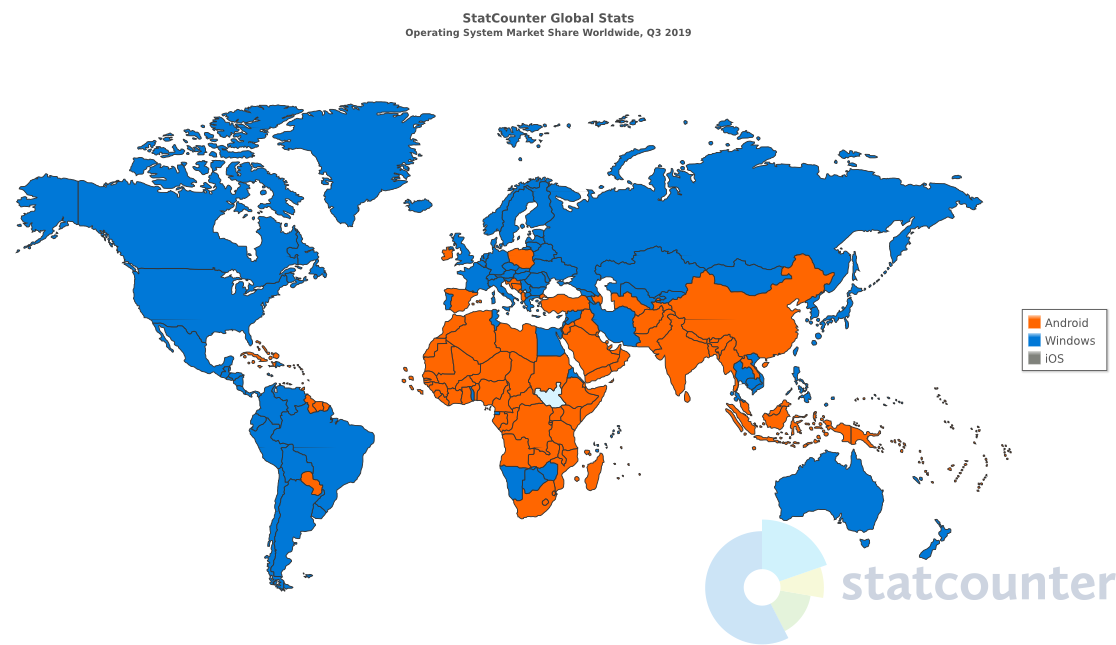
**Microsoft Windows** jest prawnie zastrzeżoną grupą rodzin graficznych systemów operacyjnych opracowywanych i sprzedawanych przez Microsoft. Każda z podgrup przeznaczona jest do użytku w innym sektorze przemysłu komputerowego. Aktualnie wspierane rodziny systemów to Windows NT i Windows IoT, które zawierają w sobie kolejne podrodziny takie jak *Windows Server* czy *Windows Embedded Compact* (Znany szerzej jako Windows CE). Warto tutaj też wspomnieć o nie wspieranych od kilku lat ale znanych przez wielu użytkowników grupach takich jak Windows 9x, Windows Mobile czy Windows Phone.

Pierwszą wersją systemu wypuszczoną na rynek w 1985 był Windows który rozbudował powłokę MS-DOS o graficzny interfejs rewolucjonizując tym rynek i bardzo szybko przewyższając MacOS zdobywając prawie 75% udziału rynkowego(Rysunek 2) wśród komputerów stacjonarnych i prawie 32% wśród wszystkich urządzeń(Rysunek 3). Warto też dodać, że przewaga androida nad systemem Windows pochodzi głównie z krajów azjatyckich, gdzie komputery stacjonarne nie są tak popularne jak w europie czy Ameryce.



Rysunek 2 udział poszczególnych systemów operacyjnych w rynku komputerów stacjonarnych

*(Źródło: )*



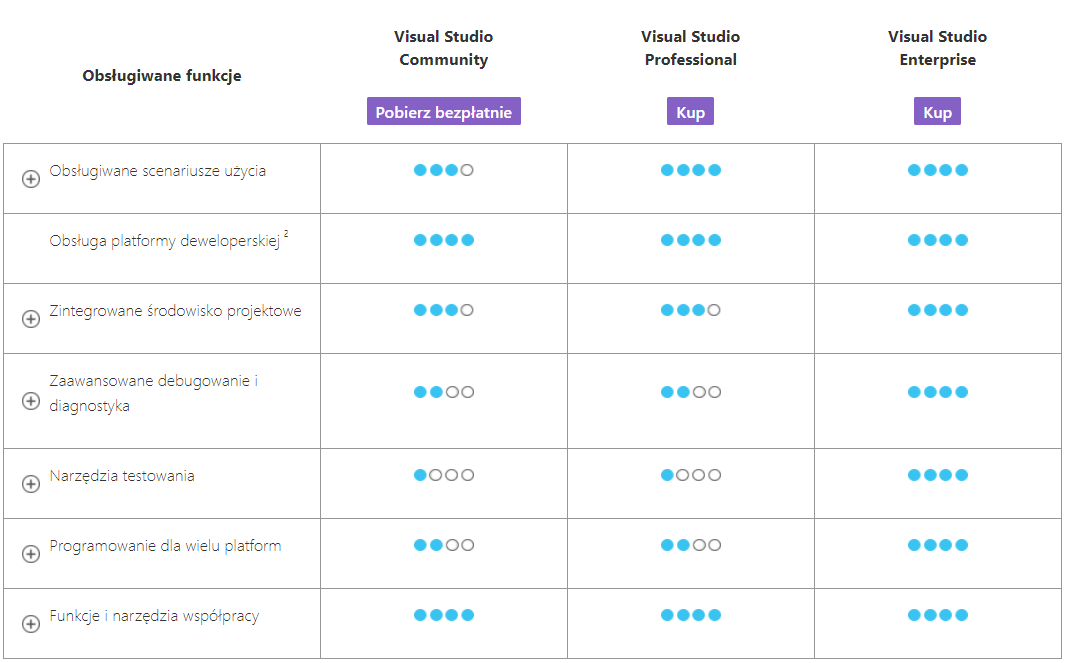
Rysunek 3 Udział rynkowy najpopularniejszych systemów operacyjnych na mapie geograficznej

*(Źródło: )*

Na dzień 11 maja 2021 roku najnowszą wersją systemu dla komputerów osobistych, tabletów i urządzeń wbudowanych jest Windows 10 wersja 21H1. Użytkownicy urządzeń serwerowych otrzymują Windows Server wersja 20H2. Dodatkowo istnieje specjalna wersja systemu dla urządzeń Xbox One i Xbox Series X/S.

Do pracy wybrałem właśnie ten system ze względu na jego popularność wśród użytkowników domowych. Dodatkowo jest on przystępny w obsłudze oraz wysoce intuicyjny co w wypadku mojego oprogramowania jest szczególnie mocnym atutem. Moją grupą docelową są nie tylko firmy związane z technologiami komputerowymi, ale również przedsiębiorstwa zajmujące się innymi dziedzinami rynku, więc program i jego otoczenie musi być możliwe w obsłudze bez wiedzy specjalistycznej i wymagać jedynie podstawowych umiejętności pracy z komputerem. W te założenia idealnie wpasowuje się Windows, który poprzez czytelny graficzny interfejs pozwala praktycznie każdemu cieszyć się z zasobów komputera.

**Visual Studio** to zintegrowane środowisko deweloperskie pozwalające na edytowanie, debugowanie oraz kompilację kodu programu oraz na publikację gotowych rozwiązań. Posiada ogromną ilość funkcjonalności wspomagających wiele aspektów procesu tworzenia oprogramowania, które nie są obecne w większości środowisk deweloperskich takich jak narzędzie do uzupełniania kodu i graficzny projektant pozwalający w prosty i intuicyjny sposób zarządzać graficznymi elementami naszego programu. Jest dostępny zarówno na platformę Windows jak i Mac. Możemy wyróżnić trzy główne edycje programu – *Community* dostępne za darmo posiadający podstawowe funkcjonalności programu, *Professional* dla profesjonalistów pracujących nad projektami o średniej złożoności oraz *Enterprise* posiadający pełen zestaw funkcjonalności przewyższający edycję *Professional* głównie pod względem narzędzi do testowania programowania i obsługi programowania dla wielu platform.

**

Rysunek 4 Funkcjonalności Visual Studio ze względu na edycję programu

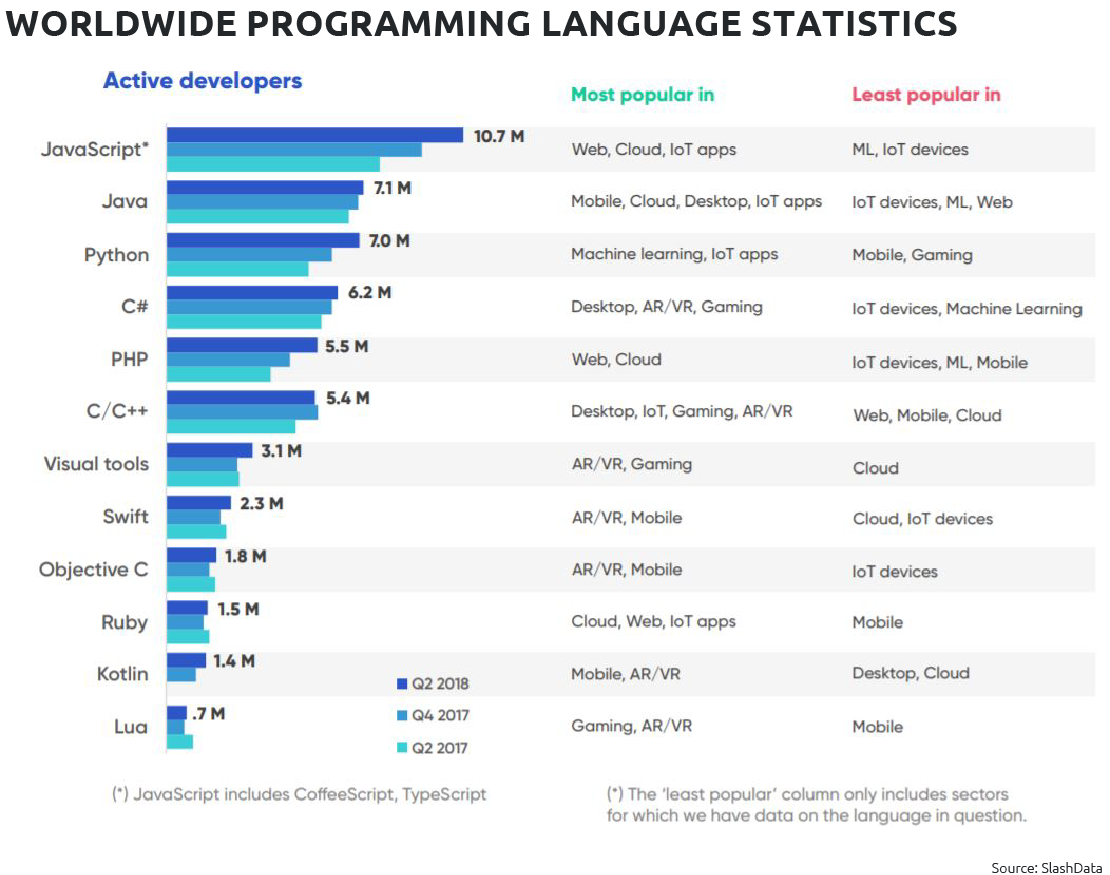
*(Źródło: )*

**Platforma programistyczna WPF (***ang. Windows Presentation Foundation***)** to framework interfejsu użytkownika służący do tworzenia aplikacji na urządzenia pracujące pod systemem Microsoft w oparciu o .NET framework. Poprzez wsparcie sporej ilości funkcji takich jak modele aplikacji, zasoby, kontrolki użytkownika, grafika, układy, powiązanie danych, dokumenty i bezpieczeństwo zapewnia programistom ogromną ilość możliwości konfiguracyjnych. Do tworzenia interfejsu użytkownika wykorzystuje deklaratywny język znaczników – XAML. Pozwala na oddzielenie procesu projektowania interfejsu od części logicznej programu dzieląc kontrolkę użytkownika na dwa elementy - definicję interfejsu i połączony z nim przy pomocy klas częściowych plik logiczny „*code-behind*”. Takie rozwiązanie dobrze sprawdzi się w sytuacji ,w której osobne zespoły pracują nad częścią interfejsu i elementami programowymi, a zwłaszcza w przypadku wzorca projektowego MVVM.

Zdecydowałem się na wybór tego środowiska ze względu na system operacyjny, który zapewnia wsparcie dla platformy .NET Framework. Dzięki temu uniknę problemów z kompatybilnością. Dodatkowo platforma zapewnia lepsze możliwości wizualne aplikacji niż jej alternatywa – Windows Forms.

**Język programowania C#** (ang. *C Sharp*) jest nowoczesnym i zorientowanym obiektowo językiem programowania należącym do grupy type-safe. Pozwala on na tworzenie bezpiecznych i solidnie wykonanych programów w oparciu o ekosystem .NET. Sporo czerpie on ze swoich poprzedników z rodziny języków programowania C. Dzięki zastosowaniu technologii komponentów programowych zapewnia struktury idealne do tworzenia i zarządzania komponentami programu. Od swojej premiery w 2000 roku jest regularnie rozwijany, a aktualna w pełni stabilna wersja to 9.0 wydana 20 maja 2020 roku. Z latami uzyskał ogromną ilość funkcjonalności pozwalających na tworzenie niezawodnych i cechujących się dużą żywotnością programów. Najważniejsze z tych funkcjonalności to między innymi Garbage Collector który automatycznie odzyskuje pamięć z obiektów nieużywane przez program obiekty, typy dopuszczające wartość null chroniące przed odwołaniem do nieistniejących lub nieokreślonych obiektów, obsługę wyjątków i wyrażenia lambda. Obsługuje również operacje asynchroniczne pozwalając programistom na rozplanowanie zadań w czasie i „przekazywanie” uwagi programu na dane zadanie wtedy kiedy tego potrzebuje. Kod napisany przy pomocy tego języka po kompilacji przekształcany jest na wspólny język pośredni IL określony przez CLI (Common Language Infrastructure). Kiedy program jest uruchamiany kod IL przekazywany jest do CLR(Common Language Runtime) gdzie przekształcany jest na język maszynowy.

Zdecydowałem się na wybór tego języka w mojej pracy ponieważ według danych statystycznych (Rysunek 5) jest on jednym z najpopularniejszych rozwiązań przy tworzeniu oprogramowania przeznaczonego do użytku na komputerach osobistych. Dodatkowym jego atutem jest fakt, że oparty on jest o platformę .NET, która wspaniale pracuje z systemem operacyjnym używanym przez większość potencjalnych użytkowników programu.



Rysunek 5 Statystyka popularności języków programowania wśród programistów

*(Źródło:*

**System.Security.Cryptography** jest biblioteką która zapewnia usługi kryptograficzne w tym bezpieczne kodowanie i dekodowanie danych. Obsługuje zarówno algorytmy symetryczne (DES,3DES,RC2,AES) jak i asymetryczne (RSA, DSA). Dodatkowo pozwala na tworzenie skrótów korzystając z następujących funkcji skrótu (MD5,SHA-1,SHA-256,SHA-483,  
SHA-512). Kolejną wprowadzaną przez bibliotekę funkcjonalnością jest możliwość generowania losowego i uwierzytelniania komunikatów. Zdecydowałem się na użycie tej biblioteki aby zapewnić bezpieczeństwo przy weryfikacji hasła w trakcie logowania użytkownika.

**MySQL Connector** to sterownik dla interfejsów JDBC, ODBC oraz .NET pozwalający na tworzenie aplikacji bazodanowych w wybranym przez nich języku. Wykorzystanie tego programu pozwoli nawiązać bezpieczne i szybkie połączenie z serwerem bazodanowym zaopatrującym system w dane. Pełna obsługa WPF idealnie wpasowuje się w moje zapotrzebowania.

**MySQL** jest najpopularniejszym systemem zarządzania bazą danych opartym o licencję Open Source. Jest tworzony, wspierany i dystrybuowany przez Oracle Corporation. Pozwala on użytkownikowi na tworzenie relacyjnych baz danych służących do przechowywania danych w strukturze podzielonej na tabele. Jej struktura pozwala na podobny podział w strukturze plików dzięki czemu zapewnia dużą wydajność. Logiczne modele takie jak baza danych, tabela, widok, wiersz i kolumna oferują wszechstronne środowisko dla programisty. MySQL pozwala również programiście na definicje relacji zachodzących między poszczególnymi encjami – obiektami w bazie danych. Zadaniem systemu jest egzekwowanie tych zasad aby zapobiec niespójnością, duplikacji, przedawnieniu lub utraceniu danych. Całość funkcjonalności pozwala systemowi na sprawne, niezawodne działanie zapewniając przy tym wysoką skalowalność i prostotę użycia. Zalety tego systemu bazodanowego idealnie wpasowały się w moje zapotrzebowania na potrzeby stworzenia aplikacji. Z tego powodu podjąłem decyzję, aby to właśnie tej technologii użyć w implementacji.

# Bibliografia

1. Definicja komputera osobistego:

https://kgfiks.oig.ug.edu.pl/ti/komp\_pc.pdf

Dane uzyskane w dniu: 04.01.2022

1. Parametry techniczne IBM 5150:

https://www.ibm.com/ibm/history/exhibits/pc25/pc25\_fact.html

Dane uzyskane w dniu: 04.01.2022

1. Statystyka dotycząca informacji o sprzedaży komputerów osobistych:

https://www.statista.com/statistics/263393/global-pc-shipments-since-1st-quarter-2009-by-vendor/

Dane uzyskane w dniu: 04.01.2022

https://www.statista.com/statistics/273495/global-shipments-of-personal-computers-since-2006/

Dane uzyskane w dniu: 04.01.2022

1. Statystyka dotycząca popularności systemów operacyjnych

https://gs.statcounter.com/os-market-share#quarterly-201903-201903-map

Dane uzyskane w dniu: 04.01.2022

https://gs.statcounter.com/os-market-share/desktop/worldwide

Dane uzyskane w dniu: 04.01.2022

1. Statystyka dotycząca popularności języków programowania

https://www.daxx.com/blog/development-trends/number-software-developers-world

Dane uzyskane w dniu: 04.01.2022

1. Dokumentacja Visual Studio:

https://docs.microsoft.com/pl-pl/visualstudio/get-started/visual-studio-ide?view=vs-2019

https://visualstudio.microsoft.com/pl/vs/compare/

Dane uzyskane w dniu: 04.01.2022

1. Dokumentacja WPF:

https://docs.microsoft.com/en-us/dotnet/desktop/wpf/getting-started/?view=netframeworkdesktop-4.8

https://docs.microsoft.com/en-us/dotnet/desktop/wpf/advanced/xaml-overview?view=netframeworkdesktop-4.8

Dane uzyskane w dniu: 04.01.2022

1. Wprowadzenie do języka C#

https://docs.microsoft.com/en-GB/dotnet/csharp/tour-of-csharp/

Dane uzyskane w dniu: 04.01.2022

1. Dokumentacja C# - System security cryptography:

https://docs.microsoft.com/pl-pl/dotnet/api/system.security.cryptography?view=net-5.0

1. Next

MySQL wyszukać literaturę!

MVVM wyszukać literaturę!

WPF wyszukać literaturę!

C# wyszukać literaturę!

Znaleźć literaturę związaną z ITIL Service Desk (najlepiej wersja 4),