



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

Projekt: System wsparcia decyzyjnego

Yurii-Volodymyr Shcheliuk

Dr inż. Arkadiusz Lewicki

Nr albumu studenta: 58913

Rzeszów 2022

Charakterystyka systemu informatycznego wsparcia decyzyjnego

Podstawowym celem korzystania z DSS (*DSS – ang. Decision Support System*) jest przedstawienie klientowi informacji w łatwy do zrozumienia sposób. System DSS jest korzystny, ponieważ można go zaprogramować do generowania wielu rodzajów raportów, a wszystko to w oparciu o specyfikacje użytkownika. Na przykład DSS może generować informacje i przedstawiać je w formie graficznej, na przykład na wykresie słupkowym przedstawiającym przewidywany dochód lub w formie pisemnego raportu.

Wraz z ciągłym rozwojem technologii analiza danych nie ogranicza się już do dużych, nieporęcznych komputerów typu mainframe. Ponieważ DSS jest zasadniczo aplikacją, można ją załadować na większości systemów komputerowych, zarówno na komputerach stacjonarnych, jak i laptopach. Niektóre aplikacje DSS są również dostępne za pośrednictwem urządzeń mobilnych.

Elastyczność DSS jest niezwykle korzystna dla użytkowników, którzy często podróżują. Daje im to możliwość bycia dobrze poinformowanymi przez cały czas, dając im możliwość podejmowania najlepszych decyzji dla swojej firmy i klientów w podróży lub nawet na miejscu.

Typowe informacje, które aplikacja wspomagająca podejmowanie decyzji może gromadzić i prezentować, obejmują:

- porównawcze dane dotyczące sprzedaży w okresie od tygodnia do tygodnia;
- prognozowane dane o przychodach oparte na założeniach sprzedaży nowych produktów;
- konsekwencje różnych decyzji.

Składnia DSS

Typowy DSS składa się z trzech różnych części:

1. bazy wiedzy
2. oprogramowania
3. interfejsu użytkownika.

Użytkownicy mogą również wkleić sztuczną inteligencję (AI) do systemów wspomagania decyzji. Nazywane inteligentnymi systemami wspomagania decyzji (IDSS), sztuczna inteligencja wydobywa i przetwarza duże ilości danych, aby uzyskać wgląd i przedstawić zalecenia dotyczące lepszego podejmowania decyzji. Czyni to, analizując wiele źródeł danych i identyfikując wzorce, trendy i powiązania w celu naśladowania ludzkich zdolności decyzyjnych.

Rodzaje systemów wspomagania decyzji

Systemy wspomagania decyzji można podzielić na kategorie, z których każda opiera się na swoich podstawowych źródłach informacji.

DSS oparty na danych

Podaje decyzje na podstawie danych z wewnętrznych lub zewnętrznych baz danych, co umożliwia przewidywanie przyszłych zdarzeń. Np. przewidywania prawdopodobieństwa przyszłych zachowań przestępczych.

DSS oparty na modelu

Oparte na podstawowym modelu decyzyjnym systemy wspomagania decyzji oparte na modelach są dostosowywane zgodnie z predefiniowanym zestawem wymagań użytkownika, aby pomóc w analizie różnych scenariuszy spełniających te wymagania. Na przykład DSS oparty na modelu może pomóc w planowaniu lub opracowywaniu sprawozdań finansowych.

Oparte na komunikacji i grupowe DSS

Oparte na komunikacji i grupowy system wspomagania decyzji wykorzystuje różnorodne narzędzia komunikacyjne — takie jak poczta e-mail, komunikatory lub czat głosowy — aby umożliwić więcej niż jednej osobie pracę nad tym samym zadaniem. Celem tego typu DSS jest zwiększenie współpracy pomiędzy użytkownikami a systemem oraz poprawa ogólnej wydajności i skuteczności systemu.

DSS oparte na wiedzy

W tego rodzaju systemie wspomagania decyzji dane sterujące systemem znajdują się w bazie wiedzy, która jest stale aktualizowana i utrzymywana przez system zarządzania wiedzą. DSS oparty na wiedzy dostarcza użytkownikom informacje spójne z procesami biznesowymi i wiedzą firmy.

DSS oparty na dokumentach

Umożliwiają użytkownikom przeszukiwanie stron internetowych lub baz danych albo znajdowanie określonych terminów wyszukiwania. Przykłady dokumentów, do których uzyskuje dostęp system DSS oparty na dokumentach, obejmują zasady i procedury, protokoły posiedzeń i akta korporacyjne.

Przykłady systemów wspomagania decyzji

Organizacje używają systemów wspomagania decyzji w kilku różnych kontekstach, w tym:

Trasowanie GPS.

Planowanie trasy GPS jest przykładem typowego DSS. Porównuje różne trasy, biorąc pod uwagę takie czynniki jak odległość, czas jazdy i koszt. System nawigacji GPS umożliwia również wybór alternatywnych tras, wyświetlanie ich na mapie i dostarczanie instrukcji krok po kroku.

Dashboardy ERP

Mogą wykorzystywać system wspomagania decyzji do wizualizacji zmian w procesach produkcyjnych i biznesowych, monitorowania bieżącej wydajności biznesowej względem wyznaczonych celów oraz identyfikowania obszarów wymagających poprawy. Pulpity nawigacyjne ERP pozwalają właścicielom firm zobaczyć migawkę najważniejszych liczb i wskaźników ich firmy.

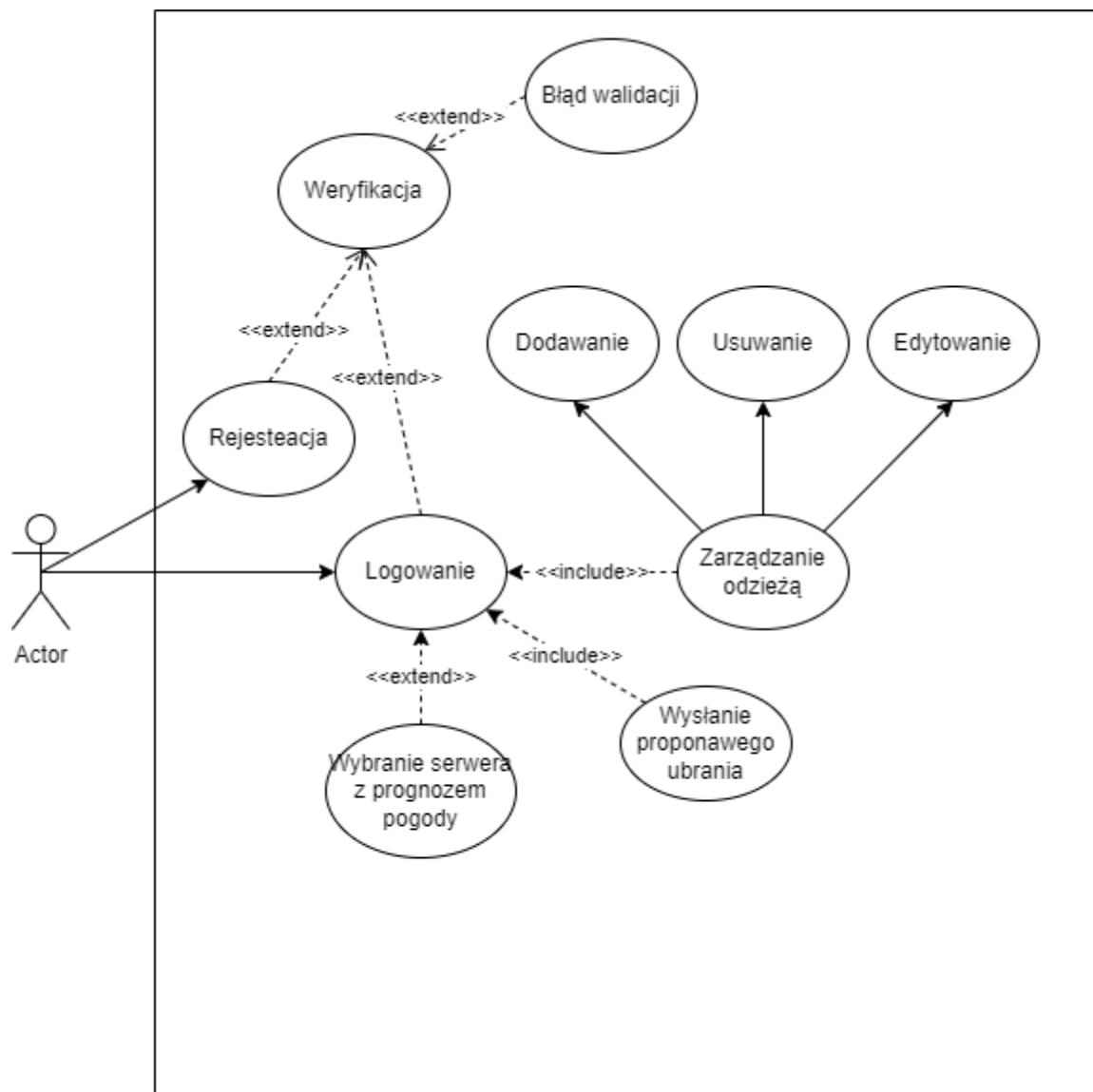
System wspomagania decyzji klinicznych

To oprogramowanie, które wykorzystuje zaawansowane algorytmy podejmowania decyzji, aby pomóc lekarzom w podejmowaniu najlepszych decyzji medycznych. Pracownicy służby zdrowia często używają ich do interpretacji dokumentacji pacjentów i wyników badań oraz do obliczania najlepszego planu leczenia. W opiece zdrowotnej może pomóc świadczeniodawcom zidentyfikować nieprawidłowości podczas określonych testów, a także monitorować pacjentów po pewnych procedurach w celu ustalenia, czy mają jakiegokolwiek działania niepożądane.

Projektowanie systemu decyzyjnego

Najbardziej powszechnym pytaniem jest pytanie co do pogody. Jak nie jest ona codziennie taka sama, a zmienia się z gorącego dnia na deszczowe tygodnie, to podejmować decyzję co ubrać by nie było ani za gorąco ani za chłodno pomógłby system decyzyjny który by uwzględniał w czasie rzeczywistym prognozowaną pogodę i na podstawie zdjęć garderoby by proponował w co się ubrać w dniu dzisiejszym lub wybranym okresie kilku tygodni.

Celem danego projektu jest stworzenie rozwiązania danego problemu (w postaci systemu o nazwie „Fit for the weather”, składającego się z części serwerowej, aplikacji mobilnej oraz portalu internetowego). Dla lepszego zrozumienia dostępnej użytkownikowi funkcjonalności danego rozwiązania został stworzony diagram przypadków użycia danego systemu.



Rys. 1 Diagram przypadków użycia aplikacji
Źródło: Opracowanie własne

Korzystanie z funkcjonalności systemu

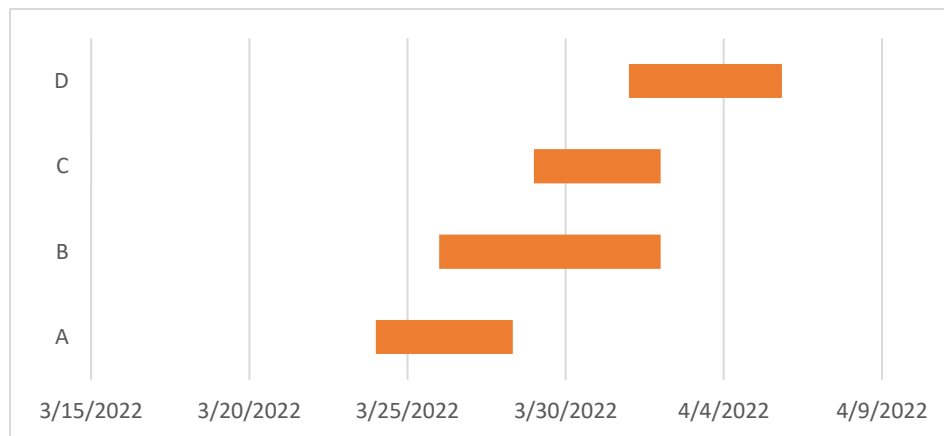
1. Dodawanie własnego serwera do pobrania prognozy pogody
2. Dodawanie odzieży do systemu
 - a. Wybór kategorii
 - i. Dodanie własnych kategorii
 - b. Dodawanie priorytetu

Otóż z tego wynikają następujące zadania dla aplikacji mobilnej:

Czasy trwania zadań

Zadanie	Nazwa	Poprzednik	Optymistyczny	Naj. prawd.	Pesymistyczny	Oczekiwany	wariancja czasu oczekiwanego
A	Logowanie	-	3.00	4.00	7.00	4.33	0.44444444
B	Inegracja API wraz z wyświetlaniem danych	A	5.00	7.00	9.00	7.00	0.44444444
C	Zarządzanie danymi użytkownika	B	3.00	4.00	5.00	4.00	0.11111111
D	Poprawianie bugów a publikowanie	C	2.00	5.00	7.00	4.83	0.69444444

Z sumowania wyników tabeli można wnioskować, że minimalny czas realizacji projektu stanowi 20 dni roboczych.



Rys. 1 Diagram Ganta
Źródło: Opracowanie własne

Ścieżka krytyczna będzie wyglądać w postaci szeregowych zadań, czyli: A-B-C-D

Szacowanie złożoności oprogramowania metodą punktów funkcyjnych:

Rodzaje atrybutów FP:

1. Liczba wejść zewnętrznych (EI)
2. Liczba wyjść zewnętrznych (EO)
3. Liczba zapytań zewnętrznych (EQ).
4. Liczba interfejsów zewnętrznych (EIF).
5. Liczba plików wewnętrznych, baz danych i katalogów (ILF)

Typ funkcji	Proste	Średnie	Złożone
EI	3	4	6
EO	4	5	7
EQ	3	4	6
EIF	7	10	15
ILF	5	7	10

Szacowane dane z programu:

Liczba wejść użytkownika (średnie) = 24

Liczba wyjść użytkownika (średnie) = 46

Liczba zapytań = 8

Liczba plików = 4

Liczba interfejsów zewnętrznych = 2

Cena = 5000zł/mieś

Wysilek = 37 (osób miesięcznie)

FP – Punkty funkcyjne

UT – Wstępne oszacowanie

CM – Mnożnik złożoności $CM = 0.65 + 0.01 * \Sigma$ Współczynniki wpływu

14 współczynników wpływu, 0 - 5 punktów każdy:

1. Czy jest wymagane przesyłanie danych?
2. Czy są funkcje przetwarzania rozproszonego?
3. Czy wydajność ma kluczowe znaczenie?
4. Czy system ma działać w mocno obciążonym środowisku operacyjnym?
5. Czy system wymaga wprowadzenia danych online?
6. Czy wewnętrzne przetwarzanie jest złożone?
7. Czy kod ma być re-używalny?
8. Czy wejścia, wyjścia, pliki i zapytania są złożone?
9. Czy wprowadzanie danych online wymaga transakcji obejmujących wiele ekranów lub operacji?
10. Czy pliki główne są aktualizowane on-line?
11. Czy system ma mieć automatyczne konwersje i instalacje?
12. Czy system wymaga mechanizmu kopii zapasowych i odtwarzania?
13. Czy system jest projektowany dla wielu instalacji w różnych organizacjach?
14. Czy aplikacja jest projektowana, aby wspomagać zmiany i być łatwą w użyciu przez użytkownika

Parametr pomiaru	Ilość		Współczynnik ważenia
1. Liczba wejść zewnętrznych (EI)	24	*	4 = 96
2. Liczba wyjść zewnętrznych (EO)	46	*	4 = 184
3. Liczba zapytań zewnętrznych (EQ)	8	*	6 = 48
4. Liczba akt wewnętrznych (ILF)	4	*	10 = 40
5. Liczba interfejsów zewnętrznych (EIF) Suma zliczania	2	*	5 = 10
Suma			378

$CM = [0,65 + 0,01 * \sum (Xi)] = 0,65 + 0,01 * (4 + 1 + 0 + 3 + 3 + 5 + 4 + 4 + 3 + 3 + 2 + 2 + 4 + 5) = 0,65 + 0,01 * 43 = 0,65 + 0,43 = 1,08$

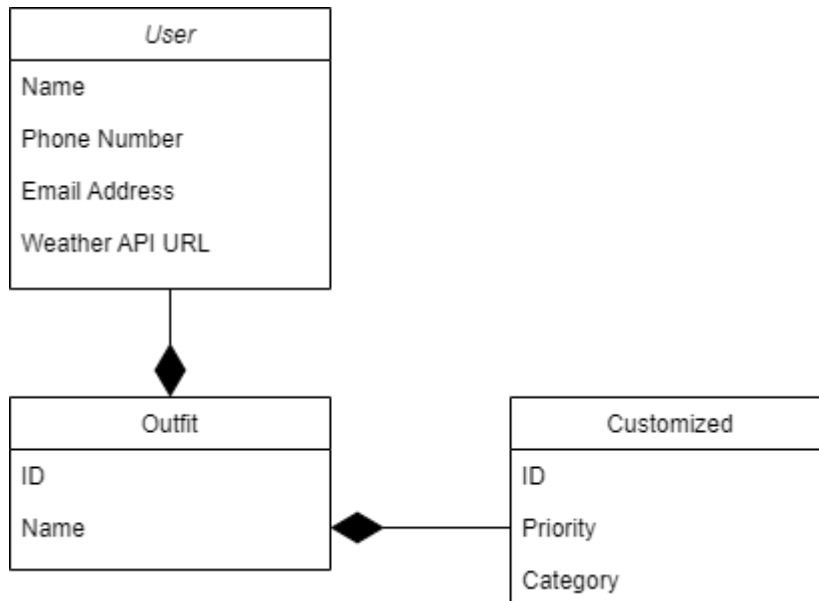
Punkty Funkcyjne (FP) = UT * CM = 378 * 1,08 = 408

Otrzymaną liczbę punktów funkcyjnych (FP) możemy następnie przemnożyć przez pewien współczynnik, aby uzyskać liczbę linii kodu (LOC) tej aplikacji, dla Java stanowi to 53 pkt, czyli wynik wynosi :
 $408 * 53 = 21\ 624$.

Produktywność = FP/Wysiłek = $408/37 = 11$, z czego wynika
 cena jednej funkcji która się oblicza formułą:
 cena/produktywność = $5000/11 = 454\text{zł}$

Diagram klas:

Diagram klas w tym przypadku by wyglądał jako połączenie „User to many Outfit” oraz „Outfits to many Customized”



Rys. 2 Diagram klas
Źródło: Opracowanie własne