 

Praca wykonana pod kierunkiem:  
**Dr hab. prof. PK Joanna Kołodziej**

**Praca inżynierska   
 na kierunku INFORMATYKA**

**Kraków, 2024**

**Imię i nazwisko autora pracy**

Numer albumu:

**Politechnika Krakowska  
im. Tadeusza Kościuszki**

Wydział Informatyki i Telekomunikacji

**Tytuł w języku polskim**

**Tytuł w języku angielskim**

# Spis treści

Najlepiej wygenerowany automatycznie

# Wstęp

Każdy pełny rozdział musi zaczynać się od nowej strony.

We wstępie należy napisać jaka była motywacja/ główny powód zajęcia się tematem pracy i stworzeniem aplikacji. Czy jest zapotrzebowanie na tego typu aplikacje? Nie trzeba mocno się rozpisywać

## **1.1 Cel i zakres pracy**

W tym podrozdziale trzeba napisać, co było głównym celem pracy. Proszę zwrócić uwagę na to, że w głównym założeniu, praca pisemna ma być pisana jak już wszystkie prace nad aplikacją i wszystkie dane, materiały testy zostały (przynajmniej teoretycznie) wykonane. W praktyce robi się to równolegle, ale w pracy pisemnej używamy w tekście czasu przeszłego – opisuje się coś, co już w domyśle zostało zrobione. Pisze o tym, bo studenci często piszą w tym miejscu w czasie przyszłym, a nie powinni.

W ramach pracy inżynierskiej tworzy się software - aplikację. Dobrze jest wymyślić dla niej nazwę własną (musi być cenzuralna). W rozdziale tym można będzie napisać mniej więcej tak: W ramach niniejszej pracy dyplomowej stworzono autorską aplikację NAZWA – do..... etc. Potem można odwoływać się do aplikacji używając jej nazwy.

## 1.2 Przewodnik po pracy (lub Zawartość pracy)

Tutaj trzeba napisać po kolei, co jest w kolejnych rozdziałach pracy. Zaczynamy rozdział zwykle od..

Praca składa się z XXXX rozdziałów. W rozdziale 2 …....

# Przegląd istniejących rozwiązań

W tym rozdziale należy opisać i porównać kilka wybranych aplikacji podobnych to tego, co tworzone jest w pracy. Mogą to być aplikacje komercyjne i mniejsze projekty. Poszczególne podrozdziały powinny być zatytułowane jak nazwy tych aplikacji. Nie ma konkretnych (formalnych) wymagań co do liczby tych podrozdziałów, ich długości i szczegółowości opisu, ale przydałoby się przeanalizować przynajmniej 3 wybrane podobne aplikacje i opisać ich podstawowe funkcjonalności. Ważne jest, aby w tych opisach wyodrębnić, jeśli jest to możliwe, wspólne funkcjonalności dla opisywanych aplikacji i różne. Można wypisać wady i zalety. Trzeba podać odnośnik (referencja) do pozycji w bibliografii dla każdej z opisywanych aplikacji/metod. Każda z nich ma dokumentację lub mniej lub bardziej dokładna prezentację na stronie internetowej, wystarczy więc link do tej strony.

## **2.X Podsumowanie**

Numer tego podrozdziału zależy od tego, ile podrozdziałów będzie wcześniej. Ma to być ostatni podrozdział w Rozdziale 2. W tym podrozdziale trzeba porównać krótko opisywane aplikacje z tą stworzoną w ramach pracy inżynierskiej. Najlepiej wybrać kilka funkcjonalności i/lub cech (można też podkreślić, czy jest to aplikacja płatna, czy nie), utworzyć tabelkę i zaznaczyć czy dana aplikacja ma taka funkcjonalność, czy nie. Na końcu w tabeli powinna być stworzona aplikacja (tu właśnie np. przyda się jej nazwa), dla której też pozaznacza się odpowiednie cechy. Pod tabelką krótkie podsumowanie, porównanie. Jednocześnie komentarz na temat tego, jak na tym tle wypada stworzona aplikacja. Takie porównanie ma też na celu wytłumaczenie motywacji do podjęcia tematu pracy, czyli dlaczego tworzenie nowej aplikacji ma sens.

# Projekt aplikacji NAZWA

Tutaj trzeba napisać o tym, jaka była ogólna koncepcja aplikacji (pomysł na strukturę/model architekturowy), czyli abstrakcyjny model architektury aplikacji – z jakich modułów się składa, jak są ze sobą powiązane. Można zamieścić jakiś abstrakcyjny rysunek, zamieścić model powiązanej bazy lub baz danych diagramy klas, etc. Ważne aby wstawiane rysunki – diagramy były czytelne (można je powiększać na całą stronę i zmieniać orientację dla tej strony – jeśli jest to zasadne)

Tworzona aplikacja jest aplikacja jest przykładem prostego projektu informatycznego, o czym trzeba napisać w pracy. Dlatego w dalszej części rozdziału powinny być wymagania funkcjonalne i niefunkcjonalne, przypadki użycia.

UWAGA: Każdy rysunek musi być podpisany (czcionka 9) i ponumerowany tak, aby można było się do niego odwoływać w tekście (trzeba takie odwołanie zrobić do każdego rysunku, czy tabeli).

Przykładowy tekst rozpoczynający ten rozdział (z odnośnikiem do rysunku):

W rozdziale tym przedstawiony został abstrakcyjny model aplikacji NAZWA. Składa się ona z następujących elementów (modułów):

* Moduł 1
* Moduł 2
* Moduł 3

Ogólna koncepcja aplikacji NAZWA przedstawiona jest na Rys. XX

## **3.X Wymagania stawiane aplikacji**

Tworzenie aplikacji NAZWA jest przykładem prostego projektu informatycznego. W poniższych tabelach zdefiniowane zostały wymagania funkcjonalne i niefunkcjonalne stawiane aplikacji oraz najważniejsze przypadki użycia.

### **3.X.1 Wymagania funkcjonalne**

### **3.X.2 Wymagania niefunkcjonalne**

### **3.X 3 Przypadki użycia**

# 4 Implementacja aplikacji NAZWA

W tym rozdziale należy opisać szczegóły implementacji

Na początku - narzędzia, biblioteki, frameworki wykorzystane w implementacji, potem nie za długo o implementacji poszczególnych modułów (bazując na modelu abstrakcyjnym przedstawionym w poprzednim rozdziale). Przy opisie narzędzi i frameworków należy umieścić odnośniki do materiałów źródłowych (pozycje w bibliografii) – mogą to być dokumentacje, książki, artykuły w czasopismach, strony internetowe.

Nie należy umieszczać zbyt wielu kodów, można zamieścić kody najważniejszych procedur, a resztę – jeśli jest taka potrzeba – zamieścić w dodatku na końcu pracy.

Jeśli aplikacja ma interfejs użytkownika, trzeba go opisać.

# 5 Testowanie aplikacji

W tym rozdziale trzeba opisać przeprowadzone testy aplikacji.

W pierwszej kolejności należy sprawdzić, czy aplikacja została zaimplementowana i działa poprawnie (testy jednostkowe oprogramowania).

Testowany jest fragment aplikacji (np. procedura) i porównywane są wyniki (np. zwrócone wartości, stan obiektu, zgłoszone wyjątki) z oczekiwanymi wynikami – tak pozytywnymi, jak i negatywnymi.

Po wykonaniu testów jednostkowych, wykonuje się testy wydajnościowe aplikacji.

Testowanie wydajności definiuje się jako typ testów oprogramowania weryfikujący czy aplikacja będzie działać poprawnie w ramach zdefiniowanego obciążenia. Na wydajność oprogramowania składają się m.in. następujące czynniki: czas reakcji, niezawodność, wykorzystanie zasobów czy skalowalność.

Głównym celem testów wydajnościowych jest sprawdzenie:

* szybkości działania aplikacji (ang. performance) - określa, czy aplikacja szybko reaguje na interakcje użytkownika,
* skalowalności (ang. scalability) - określa maksymalne obciążenie użytkownika, jakie może obsłużyć aplikacja,
* stabilność (ang. stability) - określa, czy aplikacja jest stabilna w obliczu różnych obciążeń.

Ostatni rodzaj testów to testy „jakościowe” - eksperymenty – trzeba pokazać na przykładowych danych albo w opracowanych scenariuszach w jakich sytuacjach aplikacja działa dobrze – w jakich gorzej sobie radzi. Ważne jest aby takich sytuacji nie unikać. Takie przykłady są cenne, bo nie ma aplikacji, które działają świetnie wszędzie. Na początku trzeba krótko opisać plan testów (scenariusze testów, można oprzeć się na przypadkach użytkowania).

# 6 Podsumowanie

# Bibliografia

# Spis rysunków

# Spis Tabel

# Dodatek - Listingi najważniejszych procedur (opcjonalnie)